

**Masteroppgave i geofag**

# **Utvikling av syndrombasert signalsystem og romlig analyse av influensadata**

**Torbjørn Dalløkken**



**UNIVERSITETET I OSLO**

**DET MATEMATISK-NATURVITENSKAPELIGE FAKULTET**

Blank side ved dobbelsidig papirkopi, fjernes ved  
enkelsidig kopi eller digital publisering

# **Utvikling av syndrombasert signalsystem og romlig analyse av influensadata**

**Torbjørn Dalløkken**



Masteroppgave i geofag

Studieretning: Naturgeografi, hydrologi og geomatikk

Institutt for geofag

Matematisk-naturvitenskaplig fakultet

**UNIVERSITETET I OSLO**

11.05.2009

© **Torbjørn Dalløkken, 2009**

Veileder(e): Bernd Etzelmüller, Universitetet i Oslo (hovedveileder). Berit Tafjord Heier,  
Nasjonalt Folkehelseinstitutt

Dette eksamensarbeidet er publisert elektronisk i DUO – Digitale Utgivelser ved UiO

<http://www.duo.uio.no>

Det er også katalogisert i BIBSYS (<http://www.bibsys.no/>)

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means,  
without permission.

*”...A new wave of technological innovation is allowing us to capture, process and display an unprecedented amount of information about our planet and a wide variety of environmental and cultural phenomena. Much of this information will be ‘geo-referenced’ – that is, it will refer to some specific place on the Earth’s surface. The hard part of taking advantage of this geospatial information will be making sense of it, turning raw data into understandable information...”*

- Former American Vice President Al Gore(Gore 1998)

# Forord

Når jeg nå sitter og skriver disse siste ordene i masteroppgaven, er våren i full gang utenfor lesesalen og det er på tide å tenke tilbake på arbeidet med masteroppgaven. Temaet for masteroppgaven ble gitt av Berit Heier, og arbeidet med oppgaven har vært interessant. Jeg synes det gjenspeiler de mange anvendelsene man kan bruke geografiske informasjonssystemer til. Hovedveileder har vært Bernd Etzelmüller ved Institutt for Geofag.

Jeg vil sende en stor takk til veilederne mine for gode råd og diskusjoner. Bernd har vært til god hjelp med spørsmålene jeg har hatt rundt romlig analyse og har kommet med konstruktive tilbakemeldinger når jeg har hatt behov for det. Berit har ført meg inn i et spennende fagfelt innen helseaspektet av oppgaven og har vært til stor hjelp med fremgangen av oppgaven gjennom diskusjoner. Nasjonalt Folkehelseinstitutt fortjener også en takk for at jeg har fått disponere et kontor som jeg har hatt nytte av under skriving av oppgaven.

Familie og venner fortjener en stor takk for hjelp og støtte gjennom en lang studietid. Miljøet på Institutt for Geofag og spesielt gjennom sosiale arrangement av GÆA Norvegica har bidratt til å gjøre studielivet bra. Mine medstudenter på lesesal 219 har gjort seg fortjent til en stor takk for alle diskusjoner og sosiale aktiviteter!

En meget stor takk rettes til foreldrene mine som har vært til god støtte og for korrekturlesing av oppgaven!

For de dere som ikke helt vet hva jeg har gjort det siste året: her er resultatet!

Torbjørn Dalløkken

Blindern, 11.05.2009

## Sammendrag

Denne studiens formål var å utvikle et syndrombasert signalsystem som sender e-post med et vedlagt kart dersom andel influensalignende symptomer i en kommune er større enn en gitt terskelverdi. Et annet formål var å analysere influensadata for å kunne fastslå om det fantes romlig autokorrelasjon mellom influensaregistreringene og sjekke om disse dataene var mulig å bruke som testdata i systemet. Influensadata har blitt samlet inn av Folkehelseinstituttet fra 201 legekantor hver uke gjennom 10 influensasesonger mellom 1998 og 2008.

Analysen av influensadata ble gjennomført ved hjelp av thiessenpolygoner, ordinær kriging og beregning av Morans I indeks. Ordinær kriging og Morans I indeks gir et inntrykk av hvordan dataene forholder seg til hverandre. Sammenligning av andel influensalignende symptomer basert på thiessenpolygoner, på kommuneflater og kontinuerlig overflate fra kriging ble gjennomført for å se om andel influensalignende symptomer ble det samme med de ulike metodene.

Store deler av det syndrombaserte signalsystemet ble utviklet i scriptspråket *Python* som har god integrasjon med ESRI's ArcGIS. Et program som automatisk genererer et kart over områdene med økt andel influensalignende symptomer ble utviklet i språket *C#* og ble inkorporert i signalsystemet. Systemets ytelse ble deretter testet.

Studien konkluderte med at influensadataene ikke er romlig autokorrelert, noe som enten kan komme av at det var for få og for spredte data eller at tidsoppløsningen var for høy. Det var imidlertid mulig å bruke dataene videre som testdata i signalsystemet. Sammenligningen av andel influensalignende symptomer ved de ulike metodene viste at det var stor forskjell mellom størrelsen på andel influensalignende symptomer etter beregning med de ulike metodene.

Kjøring av det syndrombaserte signalsystemet viste at alle steder med høyere forekomst av andel influensalignende symptomer enn terskelverdien ble detektert. Et oversiktskart over Norge hvor de berørte kommunene var symbolisert ble korrekt generert. Sending av e-post med vedlagt kart var også korrekt. Testen av programmets ytelse viste at kjøretiden til scriptet var sterkt avhengig av maskinressursene til pc-en som scriptet kjørt på.

# Innholdsfortegnelse

Forord .....	6
Sammendrag.....	7
1 Introduksjon .....	11
1.1 Problemstilling .....	11
1.2 Oppgavens struktur.....	12
2 Bakgrunn.....	13
2.1 Influenza .....	13
2.2 Epidemiologi.....	15
2.3 Tidligere studier .....	17
3 Teori .....	19
3.1 Overvåkningssystemer.....	19
3.1.1 Generelt .....	19
3.1.2 Overvåkning av sykdommer .....	19
3.2 Geografisk analyse .....	22
3.2.1 Generelt .....	22
3.2.2 Betydningen av romlig oppløsning .....	22
3.2.3 Anvendelse av mulitkriteriaanalyser i helse og beredskap.....	23
3.3 Interpolasjon .....	24
3.4 Romlig avhengighet og autokorrelasjon.....	24
3.5 Visualisering av geografiske data .....	25
4 Materiale .....	27
4.1 Geografiske datasett .....	27
4.2 Influensadata .....	28
4.3 Feilkilder.....	30
4.4 Bearbeiding og tilrettelegging av data.....	33
4.5 Programvare .....	33



5	Analyse av data .....	35
5.1	Utvikling av modell og system .....	35
5.1.1	Innhenting av eksterne data .....	35
5.2	Analyse av influensadata .....	36
5.2.1	Thiessenpolygoner .....	38
5.2.2	Geostatistikk og statistiske metoder .....	39
5.3	Resultater av analysene .....	41
5.4	Metoder for visualisering av data.....	49
5.5	Presentasjon av influensaspredning.....	50
6	Utvikling av signalsystem.....	53
6.1	Utvikling av signalsystem.....	53
6.2	Implementering av signalsystemet .....	56
6.3	Signalsystemets oppbygging.....	56
6.4	Vurdering av signalsystemets ytelse .....	61
7	Diskusjon .....	63
7.1	Anvendelse av varslingsystem/signalsystem.....	63
7.2	Utviklingsmetodikk og ytelse av signalsystem.....	64
7.3	Kvalitet og mulige forbedringer i data .....	66
7.4	Metoder for geografisk analyse av data.....	69
7.5	Presentasjonsmåter for influensadata .....	71
7.6	Videreutvikling av system og fremtidsutsikter .....	72
8	Konklusjon.....	74
9	Kilder .....	75
10	Appendiks.....	78
10.1	Figurliste .....	78
10.2	Tabelliste.....	80
10.3	Programkode for signalsystem.....	81

10.4	Programkode for generering av kart.....	85
10.5	Programkode for signalEditor – hovedfunksjoner .....	93
10.6	Autokorrelasjonsverdier .....	95

# 1 Introduksjon

Teknologien rundt geografiske informasjonssystemer har blitt mer og mer vanlig etter hvert som datateknologien har utviklet seg de siste tiårene. Geografiske informasjonssystemer muliggjør lagring, analyse og visualisering av geografiske data. I takt med den teknologiske utviklingen drar samfunnet mer og mer nytte av geografisk informasjon. Dette har påvirket at geografiske informasjonssystemer har kommet inn i ulike bransjer i samfunnet; det være seg naturforvaltning, forretningsanalyse, transport, sikkerhet og beredskap og innen helsevesenet. Helsevesenet kan bruke geografisk informasjon for å lagre data om utbrudd av ulike former for infeksjoner, for å analysere og finne sammenhenger i utbrudd eller informere presse og befolkningen gjennom å benytte ulike visualiseringsløsninger. Bruk av GIS innen helsevesenet kan deles inn i ulike retninger; det er de som bruker GIS for analyse av innsamlede data for å få oversikt over forekomst av sykdommer, og det er bruk av GIS til planlegging av virksomheten. Informasjon om demografi og innbyggere i en kommune kan brukes til å undersøke hvor lang tid det tar for en utrykning å komme seg fra sykehuset til ønsket sted. Det er også mulig å bruke GIS i beredskapssammenheng, ved at man for eksempel kan se hvordan vindforholdene bidrar til å spre utslipp fra en atomulykke, eller legionellabakterier fra et kjøletårn. Innen kreftforskning er det også blitt vanligere med bruk av GIS for å finne romlige forhold som kan ha en innvirkning på antallet krefttilfeller, analyse av dødelighet, samt kombinasjon mellom krefttilfeller og andre statistiske variable (Bell og Dallas 2007, Nygard et al. 2008). Geografisk informasjon kan visualiseres på flere ulike måter, som varierer mellom papirkart til kart på web.

## 1.1 Problemstilling

Folkehelseinstituttet (FHI) arbeider med et system som skal samle inn data om syndromer som influensaliknende symptomer og diaré fra allmennlegetjenesten. Målet er å overføre disse sykdomsdataene elektronisk hver natt til en database ved FHI og hensikten er å kunne oppdage utbrudd av smittsomme sykdommer tidligere enn det som er mulig i dag. Det overordnede målet for masteroppgaven er å lage et signalsystem knyttet til denne syndrombaserte overvåkingen som skal gi signal om at forekomst av et eller flere syndromer er over en viss terskel i et geografisk område. Signalene må følges opp og vurderes ved FHI.

En annen del av oppgaven blir å analysere influensadata som skal brukes som grunnlag for testing av systemet. Disse dataene må renses, og det skal undersøkes om det er mulig å finne noen romlig sammenheng i influensaregistreringene. Det må avgjøres om det er mulig å benytte dataene til testing av systemet.

Det overordnede målet med denne oppgaven er å lage et system for overvåkning av symptomer, og å se på nytteverdien av et slikt system. Data som viser informasjon om influensa er registrert for influensasesongene mellom 1998 og 2008 for 201 legekantor, og disse dataene skal kvalitetssikres for å kunne brukes som testdata i systemet. Dersom det er mulig, vil det også bli laget en spredningsmodell for disse influensadataene som viser spredningen av influensa hvert år.

## 1.2 Oppgavens struktur

Oppgaven er strukturert ved at kapittel to gir en introduksjon til influensa og generell epidemiologi, samt hvilke tidligere studier som er gjennomført innen relevante fagfelt innen geografiske informasjonssystemer anvendt innen helse, beredskap og sikkerhet. Deretter skal kapittel tre gi leseren en bakgrunn for overvåkningssystemer, analyse av geografisk informasjon og hvordan geografiske data presenteres. Kapittel fire presenterer de ulike datasettene som er brukt for analysene og hvilke programvarer som er benyttet i studien. Kapittel fem omhandler analysen som foretas av influensadataene og presenterer resultatene som kommer frem av analysen. Prosessen og metodene rundt utviklingen av signalsystemet blir redegjort for i kapittel seks og slutter med testresultater av systemets ytelse. Resultatene fra de foregående kapitlene blir diskutert i kapittel sju og det blir diskutert fordeler og ulemper med de ulike metodene og datasettet. Mulige videreutviklingsscenarioer blir også diskutert og vurdert. Resultatene og diskusjonen danner grunnlaget til oppgavens konklusjon som kommer i kapittel ni.

## 2 Bakgrunn

### 2.1 Influenza

I følge definisjonen til Folkehelseinstituttet (2005) er influensa er en akutt respiratorisk sykdom som påvirker øvre og nedre deler av de respiratoriske organene, spesielt nese, hals, bronkier og en sjelden gang lungene. Infeksjonen skyldes et influensavirus, og viruset kan kategoriseres som et A- eller B-virus. Genmaterialet som finnes i influensavirusene gjennomgår stadig små genetiske forandringer (mutasjoner), slik at det er nødvendig å utvikle nye vaksiner hvert år.

Influenza er ikke en spesiell sykdom, men i følge Folkehelseinstituttet (2005) finnes det totalt tre typer influensa; sesonginfluensa, pandemisk influensa og fugleinfluensa.

1. Sesonginfluensa eller vanlig influensa, forekommer hver vinter. Infeksjonen skyldes et virus som er litt forandret fra fjorårets virus, slik at ikke alle blir smittet, og sykdommen er relativt mild.
2. Pandemisk influensa vil si at det oppstår et helt influensavirus som ingen er immun mot. Sykdommen vil i dette tilfellet spre seg raskt over hele verden, og vil kunne medføre en mer alvorlig sykdom enn sesonginfluensa.
3. Fugleinfluensa er en sykdom som i hovedsak rammer fugler. Viruset kan ved enkelte tilfeller smitte over til menneske, men mennesker er ikke særlig mottagelig for slike virus.

Folkehelseinstituttet (2005) forklarer at influensaviruset forandres stadig gjennom endringer i arvematerialet og utveksling av arvestoff med andre influensavirus. Dette er årsaken til at mange frykter at fugleinfluensaviruset vil kunne få evnen til å smitte mellom mennesker. Utbredelsen av sesonginfluensa varierer i følge Folkehelseinstituttet (2005) fra år til år, men ligger gjennomsnittlig på mellom 5 – 10 % av befolkningen. Ved en pandemisk influensa vil utbredelsen kunne komme opp i halvparten av landets befolkning, men som regel en tredjedel i løpet av de første bølge influensasykdommen rammer i (Folkehelseinstituttet 2005).

Influensaviruset smitter ved kontakt- og dråpesmitte. WHO (2003) informerer om at nærkontaktsmitte vil si at viruset smitter dersom man puster inn dråper med influensavirus fra smittede personer som hoster eller nyser; man kan også bli smittet gjennom direkte kontakt

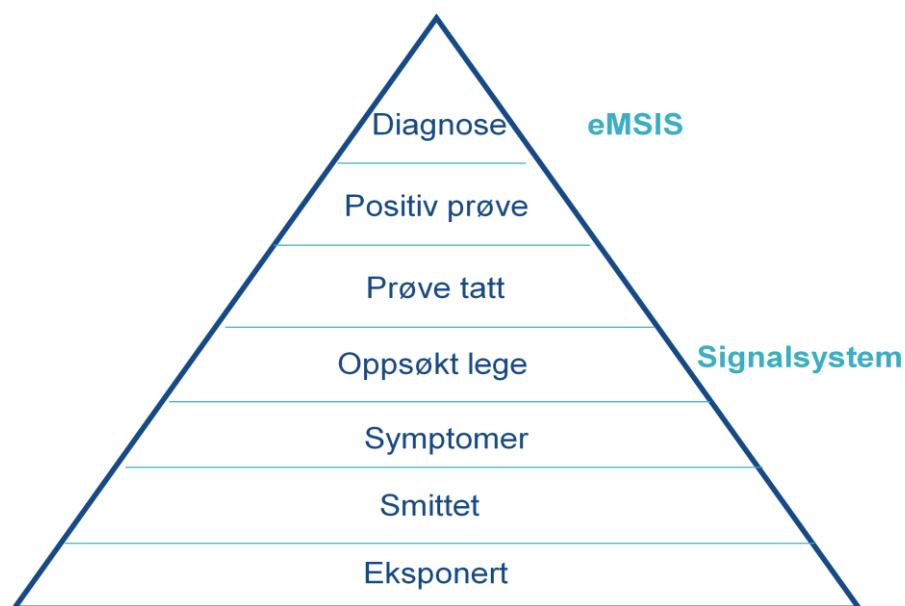
med smittede personer. Tiden det tar fra man er smittet og til sykdommen slår ut kalles inkubasjonstid, og for influensavirus er denne tida vanligvis to dager (WHO 2008). Dette kan imidlertid variere, fra én til tre dager. Smittede personer er smitteførende fra dagen symptomene oppstår og i tre til fem dager etterpå. Etter å ha gjennomgått sykdom fra et influensavirus, vil man være immun mot denne virusstammen i mange år fremover.

Det blir utviklet influensavaksiner hvert år for å forebygge sykdom hos risikogruppene. I forbindelse med influensa er det barn og unge, samt eldre som er i risikogruppene. Vaksinen gir beskyttelse hos 70-80 % av de vaksinerte.

En terskelverdi blir beregnet i starten av hver sesong for å finne en verdi på hvor mange prosent som er syke ved et utbrudd. Denne verdien blir beregnet på grunnlag av baseline. Baseline er en verdi som blir beregnet ut fra gjennomsnittlig andel ILS i de første ukene av en influensasessong. Baselineverdien tilsvarer på denne måten andelen av befolkningen som har influensa gjennom de delene av året hvor det ikke er influensasessong (EISS 2006).

Baselineverdien ligger vanligvis et sted mellom 0,5 og 0,6 % andel ILS. Terskelverdien som angir et utbrudd, blir registrert som den verdien som ligger to standardavvik høyere enn baseline. Det er ønskelig å oppdage potensielle utbrudd før de faktisk forekommer. Det er derfor laget en signalverdi som ligger noen verdier under terskelverdien. Dersom en finner at det er høyere forekomst av infeksjonen enn signalverdien, vil dette bli kontrollert og monitorert av den personen som har ansvaret for infeksjonen.

Som **Feil! Fant ikke referanse kilden.** viser er det stor forskjell på hvor mange som er smittet og hvor mange som får diagnostisert influensasykdommen. Dette skyldes at mange ikke oppsøker lege ved influensalignenede symptomer, men bare lar det gå over av seg selv. Datasettet som er benyttet i analysen og signalsystemet består av registreringer av de pasientene som har hatt en legekonsultasjon og kommet inn med influensalignenede symptomer. Av disse pasientene er det igjen en liten del som det blir tatt videre prøver av som blir sendt til mikrobiologisk analyse. Når prøveresultatene er klare, vil antallet som har en positiv prøve være betydelig mindre igjen. Disse får da konstatert diagnosen influensa. Når diagnosen har blitt endelig, registreres det i databasesystemet eMSIS.



**Figur 2.1:** Figuren viser hvordan antallet pasienter reduseres frem til diagnose

En av de verste pandemier verden har sett fant sted i året 1918 til 1919, og ble kalt Spanskesyken. Til sammen forårsaket denne influensapandemien anslagsvis 50 – 100 millioner dødsfall over hele verden. I følge Mamelund (2004), viser det seg at dette tallet er omtrent fem ganger så høyt som antall falne mellom 1914-1918 under første verdenskrig, og kostet flere liv på dette ene året enn HIV/AIDS har gjort gjennom de siste 25 årene. Det er usikkert hvor denne sykdommen først oppstod, men den fikk navnet etter at det var Spania som først rapporterte om epidemien. Studien til Mamelund (2004) forklarer videre at influensaepidemien bestod av tre bølger, hvor av den andre kostet flest mennesker livet; dette gjenspeiles i at 2/3 av antall dødsfall skjedde mellom oktober og desember 1918. De siste rapporterte tilfellene av sykdommen kom vinteren 1920. Spanskesyken hadde en dødelighet på mellom to til tre prosent på verdensbasis, mens den i Norge hadde en dødelighet på en prosent (Mamelund 2004). Dette er opptil ti ganger så høy dødelighet som vanlig influensa.

## 2.2 Epidemiologi

Epidemiologi er i følge Foldsprang et al. (1986) definert som den vitenskapen som studerer helsefenomeners forekomst og endringer i humane populasjoner. Helsefenomener kan spenne fra optimal sunnhet til dødstilstanden, og enkeltstående biologiske og psykologiske trekk. Rothman (2002) skriver at studiene foregår ved at fenomenet er en avhengig variabel, og man kan studere det sammen med en rekke uavhengige variabler; som blant annet biologiske-, psykologiske-, sosiale- og kjemiske forhold. Epidemiologi kan deles opp i en vitenskapelig

disiplin og praktisk disiplin; der hvor den vitenskaplige disiplinen har som mål å finne generell viten om en befolknings helse, så har den praktiske disiplinen til formål å fremskaffe og anvende den viten som er nødvendig for å påvirke konkrete helseforhold. Studiet av hyppighet og fordeling av sykkelighet og dødelighet i en befolkning kalles hører til beskrivende epidemiologi, mens studiet av årsakssammenhenger og effekter hører inn under den analytiske delen av epidemiologi (Rothman 2002).

Foldspang et al. (2002) definerer at en human populasjon karakteriserer en gruppe personer, og ofte vil den humane populasjonen i en epidemiologisk undersøkelse bestå av opptil flere tusen personer, eller til og med hele nasjoners og kontinenters befolkning. Sammensetningen av gruppen vil endre seg ut fra hva som ønskes undersøkt i studien.

Historisk sett har man innen epidemiologi vært opptatt med å studere epidemisk forekomst av smittsomme sykdommer. Som egen vitenskap har epidemiologi eksistert i ca 100 år, men tankegangen kan spores helt tilbake til de eldste medisinske skrifter. I følge Foldspang et al. (1986) har fokuset etter hvert endret seg til også å gjelde forekomst av kroniske sykdommer og ulike typer helsefenomener. I midten av 1900-tallet ble det i følge Foldspang et al. (1986) stadig mer fokus på forskning innen epidemiologi. Etersom det finnes sykdommer som i stor grad er påvirket av mange faktorer og utvikles etter flere års påvirkning, er det blitt stilt stadig større krav til epidemiologiske metoder og redskaper. Informasjonsteknologien med blant annet geografisk informasjon spiller nå en avgjørende rolle innen fagfeltet. Fagområdet har derved blitt bredere etter hvert som utviklingen har gått fremover, men fortsatt er det de sosiale- og miljømessige forholdene som er det sentrale. Epidemiologi og toksikologi utgjør en stor del av det vitenskaplige grunnlaget innen forebyggende medisin, som studeres i samfunnsmedisin (Foldspang et al. 1986).

Ved *rate* menes en forandring i én størrelse, som står i relasjon til forandring i en annen størrelse, gjerne representert ved tid . Foldspang et al. (1986) definerer også insidens som nye sykdomstilfeller. Insidensen av en gitt sykdom kan uttrykkes som et gitt antall tilfeller, som en rate. I denne studien blir det brukt en variabel som kalles andel ILS, hvor ILS står for influensalignende symptomer.

$$\text{andel ILS} = \frac{\text{antall sykdomstilfeller}}{\text{antall konsultasjoner}}$$



## 2.3 Tidligere studier

Innen epidemiologi er det blitt stadig vanligere å analysere geografisk informasjon, ettersom man da kan komme frem til romlige betingelser og årsaker til sykdommer man ellers ville hatt problemer med å finne. Det tidligste eksemplet på bruken av kartografi innen helse er et verdenskart som viser forekomst av sykdommer, som ble tegnet av Finke i 1792, og en kartlegging av forekomster av gulfeber i New York fra 1798.

Det har vært en rekke forsøk og studier hvor det er brukt GIS for å overvåke infeksjonssykdommer. Det er forsket på både hvordan GIS kan brukes for å få oversikt over sykdomstilfellene, men også hvordan man kan publisere karttjenester på nett med informasjon om forekomst av forskjellige sykdommer; blant annet beskrevet i Rolfhamre et al. (2004). Ved Smittskyddsinstituttet i Sverige ble det i 2004 utviklet og implementert en karttjeneste for å gi allmennheten mulighet til å visualisere de ulike forekomstene, og resultatene ble publisert i en artikkel skrevet av Rolfhamre et al. (2004). Dette systemet baserte seg på informasjon om 54 meldepliktige sykdommer som har blitt registrert i en nasjonal overvåkningsdatabase som igjen er basert på rapporter om smitte fra hele landet. Dataene ble i følge Rolfhamre et al. (2004) presentert på en webside hvor man kan se informasjon i tabeller og ta ut dataene i xls-format, samt at man kan lage et interaktivt kart basert på tabellene som enten viser alle forekomstene av sykdommen eller de som har blitt infisert i Sverige. Her er det meget viktig å ta hensyn til datasikkerhet og personvern.

I likhet med undersøkelsen til Rolfhamre et al. (2004), har det vært gjort studier på bruken av webbaserte løsninger for kartlegging og publisering i to delstater i Canada og USA. Denne studien ble gjennomført av Gao et al. (2007) for å undersøke effektiviteten av et internasjonalt overvåkningssystem I studien var det New Brunswick i Canada og delstaten Maine i USA som ble brukt som testområder på grunn av nærheten til hverandre og det store antallet reisende mellom de to områdene. Ettersom det er litt ulike data som er innhentet i de to testområdene ble de integrert ved å bli lagt inn i et felles skjema, for eksempel postnummer og zip-kode blir lagret som ett attributt postnummer. Ulike statistiske metoder ble brukt for å standardisere en tegnforklaring for de ulike representasjonene av datalagene.

Thiessenpolygoner har blitt brukt til blant annet visualisering av diarétilfeller i en studie gjennomført av Pande et al. i 2007. Studien inkluderte ulike variable som for eksempel tilgang til god vannkvalitet, belastningen på vannakviferen, belastning på vannkvaliteten og gjennomsnittlig regnfall og data hentet fra USAID. Resultatene viste at det ikke ville gi noen

betydelig nedgang i andel ILS med diaré dersom friskt og rent vann ble gjort tilgjengelig (Pande et al. 2008). Thiessenpolygoner har også blitt vurdert brukt i en analyse av dekningsområde til sykehus basert på reisetid (Schuurman et al. 2006). Problemet ved bruken av thiessenpolygoner i denne sammenhengen oppstår ved at thiessenpolygonene ikke tar hensyn til størrelsen og kapasiteten til sykehuset, og heller ikke høydeforskjeller eller veikvalitet. Det er et generelt problem med thiessenpolygoner at man ikke kan inkorporere flere variable til metoden.

Overvåkningssystemer basert på geografisk informasjon har blitt vanligere etter hvert som teknologien innenfor GIS og fjernanalyse har blitt bedre. Data om sykdommer og sykdomstilfeller kan bli analysert for å finne spredningsmønstre; enten enkeltvis eller kombinert med ulike typer datasett. I 2003 ble det publisert en artikkel i *International Journal of Health Geography* skrevet av Rogers og Randolph hvor mulighetene for bruk av fjernanalyse og GIS innen overvåkning av infeksjonssykdommer.

I en studie viser Martin et al. (2007) at GIS-baserte overvåkningssystemer har blitt brukt blant annet ved utbruddet av fugleinfluensa H5N1 epidemien i 2003-2004. Dette overvåkningssystemet ble brukt for å kontrollere hvor influensaen spredde seg og hvor smitten befant seg ved et gitt tidspunkt. Overvåkningssystemet er beskrevet i studien og består av en Oracle-database inneholdende informasjon om sykdomstilfeller; både på ville- og tamme fugler. Databasen ble oppdatert med informasjon hentet fra henholdsvis WHO og OIE. For å få muligheten til geografisk analyse og datavisualisering, ble denne databasen koblet til et GIS.

I 2004 ble en studie av romlig analyse av influensalignende symptomer med data fra Nederland og Tyskland utført av Uphoff et al. (2004). For å ta hensyn til denne faktoren, ble det utviklet en harmoniseringsindeks mellom de to landene basert på det normale nivået ved en influensaepidemi. Romlig analyse ble utført ved å bruke ordinær kriging for å interpolere manglende verdier. Resultatene viste at det er mulig å gjøre influensakartlegging og analyse på tvers av landegrensene.

## 3 Teori

### 3.1 Overvåkningssystemer

#### 3.1.1 Generelt

Overvåkningssystemer, også kalt *early warning systems*, baserer seg ofte på en eller annen form for GIS teknologi og har blitt tatt i bruk i ulike sammenhenger. Overvåkningssystemet har som funksjon å varsle om en forestående hendelse som kan ha farlige konsekvenser for mennesker, infrastruktur eller land. En anvendelse av slike overvåkningssystemer er geohazards, hvor man for eksempel overvåker et område med muligheter for potensielle skred eller innenfor jordskjelvovervåkning. I følge Einstein og Sousa (2007) kan en sammenligne overvåkningssystemer for en naturfare med overvåkningssystemene som brukes innen helsevesenet. Dette fører frem til konklusjonen på artikkelen, hvor det er oppsummert seks forhold som må være på plass for at et overvåkningssystem skal kunne fungere optimalt (Einstein og Sousa 2007):

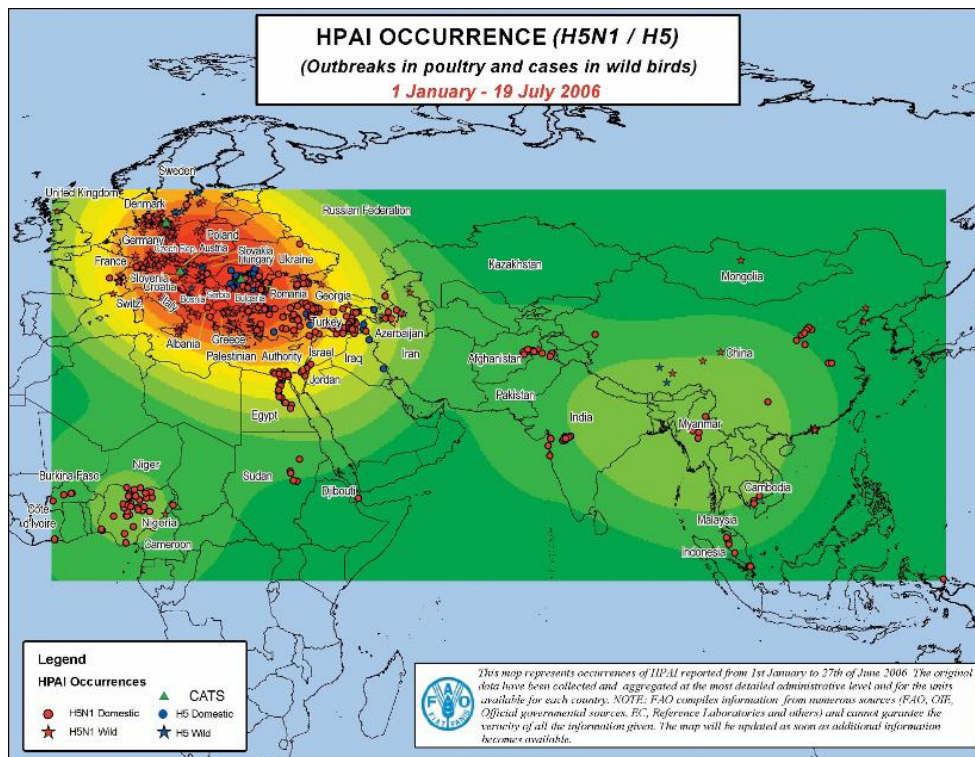
1. Situasjonen eller trusselen blir registrert av et instrument som varsler om faren
2. Informasjonen som instrumentet fanger opp blir sendt videre til et beredskapssenter og analyseres av eksperter, med hensyn til tidligere erfaringer og modeller.
3. Ut ifra utfallet av ekspertanalysen, bestemmes det om det skal sendes ut varsel til de berørte.
4. Faren er nødt til å bli videre observert av eksperter sammen med teknisk utstyr og observasjonene blir brukt for oppdatering av informasjonen.
5. Informasjonen om trusselen blir oppdatert; og trusselsnivået blir enten forhøyet eller senket ut fra informasjonen.
6. Lokale og regionale myndigheter beslutter ut ifra punkt 3 og 5 om det skal igangsettes tiltak som beskytter mot trusselen.

#### 3.1.2 Overvåkning av sykdommer

I oppgaven har jeg fått tilgang til data om forekomst av influensa, og dette er en sykdom det er gjort mange studier på. Det er imidlertid ikke så mange studier som går direkte på bruken av GIS til å kartlegge influensa. Uphoff et al. (2004) er en av de studiene som har blitt utført viser at det er mulig å bruke data fra influensaovervåkning i forskjellige land til å kartlegge

situasjonen. Tyskland og Nederland er nabolandene som i studien til Uphoff et al. (2004) har blitt undersøkt om bruken av influensadata. Dette er mulig selv om det er forskjellige data som blir registrert i de ulike landene. Et av problemene med en slik studie er at man ikke nøyaktig vet hvilket areal som legekantorene opererer på (Uphoff et al. 2004).

I forbindelse med utbruddene av fugleinfluensa (H5N1) hos villfugl og kommersielt fjørfe i Europa har det blitt økt fokus på spredningen og beregning av risikoen for potensielle pandemier hos mennesket (Martin et al. 2007). I denne artikkelen nevnes det at mange organisasjoner innen helse, både menneske- og dyrehelse, bruker et system kalt EMPRES (Emergency Prevention System for Transboundary Animal and Plant Pests and Diseases program) og EMPRES-i for overvåkning og analyse av situasjonen. Ut fra datagrunnlaget som ligger i systemet, blir det generert et raster som forteller om tettheten av utbrudd (Figur 3.1). Dette kan gi innsikt i hvilke områder som er mest utsatt og hvilke romlige faktorer som kan spille en rolle i et evt. utbrudd. Tre-dimensjonale representasjoner av dataene er også brukt for å kunne kartlegge spesielle romlige mønstre i forekomst av utbrudd. Andre informative beregninger som kan brukes innen epidemiologi er hot-spot analyser og romlig-temporale data.



Figur 3.1: Utbruddstetthet av høypatogenisk fugleinfluensa (Martin et al. 2007)

Ettersom det er ønskelig å lære mer om hvordan influensa sprer seg i en befolkning har det vært flere forsøk på å modellere og simulere spredning. Timpka et al. publiserte i 2005 en rapport hvor det ble utviklet en metode for å simulere influensatilfeller i Sverige. Denne simuleringen ble delt opp i to deler, en del for modellen og en del for output fra modellen. Resultatene fra simuleringen viser at det er viktig å balansere kriteriene for systemeffektivitet og detaljeringsgraden av modellen når det skal være et så reelt helsesystem som mulig (Timpka et al. 2005). Ettersom influensa har relativt kort inkubasjonstid, er det også viktig å få bekreftet diagnosen raskt; slik at man får skilt ut de som bare har influensalignende symptomer. Det er også vanskelig å programmere en samfunnsmodell som tar hensyn til de mellommenneskelige faktorene som; interfamiliær kontakt, sosiale nettverk, og om en har ønske og mulighet for å ta seg av syke.

Influensasesongene mellom 2003 og 2007 ble kartlagt gjennom EISS, European Influenza Surveillance Scheme, og resultatene fra studien er publisert i tre artikler av Arkema et al. (2009). Influen্সadata ble innhentet for en rekke europeiske land, og det ble foretatt analyser av virustypene i de ulike landene, og romlig analyse ble foretatt. Metoden som ble brukt i den romlige analysen var regresjonsanalyse hvor man plottet landets midtpunkt mot uken det var mest influensaaktivitet i hvert land. EISS rapporten fra Meijer et al. (2007) for sesongen 2005 og 2006 viste at influensaepidemien startet relativt sent for store deler av Europa, og det syntes som at det var virustype B som stod for mesteparten av smitten første del av sesongen. Influen্সatype A fikk en oppsving mot slutten av registreringene. Regresjonstesting på den romlige delen av dataene viste at det for denne sesongen ikke var noen tydelig vest-øst trend, slik som ved de foregående sesongene (Meijer et al. 2007b). For sesongen mellom 2006 og 2007 var influensasmitten relativt moderat i forhold til tidligere år, og det var en homogen spredning av virustype A. Retningen som influensaen sprer seg i, er uberegnelig i denne sesongen, men en liten nordlig trend er synlig på slutten av sesongen (Arkema et al. 2009).

Geografisk informasjon til bekjempelse og analyse av sykdomsutbrudd er også tatt i bruk i Norge. Her er det spesielt utbruddet av legionella i Østfold 2005 som skiller seg ut med avansert bruk av slik analyse. I følge Nygard et al. (2008) ble 55 personer smittet av legionella under utbruddet i Fredrikstad/Sarpsborg, og det var påkrevet å finne smittekilden for å stoppe spredningen. Legionella smitter via små vanndråper i lufta, og det ble etter hvert fastslått at smittekilden var kjøletårn. Artikkelen til Nygard et al. (2008) forteller at analysen for å finne hvor smittekilden befant seg innebar å kartlegge alle installasjoner av kjøletårn i området og andre smittekilder, plote bostedet til de smittede og alle bevegelsene de har hatt i

området i smitteperioden, hvor de smittede hadde beveget seg, samt å koble dette opp mot vindhastighet og vindretning. Fra denne analysen kom man frem til ett mulig smittested, som viste seg å være et kjøletårn (Nygard et al. 2008).

## 3.2 Geografisk analyse

### 3.2.1 Generelt

Geografiske analyser utføres ved å bruke en funksjon eller en samling funksjoner, på ett eller flere datasett, og man får et nytt datasett som resultat. Dersom analysen består av flere funksjoner og flere datasett, kaller man det en kartografisk modell.

### 3.2.2 Betydningen av romlig oppløsning

Når man bruker data i GIS analyser spiller den romlige oppløsningen en betydelig rolle.

Romlig oppløsning er definert som det minste mellomrommet mellom to visualiserte eller prosesserte elementer; samt den minste størrelsen av et objekt som kan bli kartlagt. (Burrough et al. 1998) Det er viktig at den romlige oppløsningen passer til det fenomenet man skal studere; det er ikke nødvendig med detaljer på meternivå dersom man vil analysere hele terrenget på vestlandet. Hvis man derimot skal studere terrenget innenfor et lite område, må man ha mer detaljerte data.

Dataene som er tilgjengelig har en god distribusjon som spenner over hele Norge. Dette gjør at den romlige oppløsningen blir relativt stor, ettersom vi ikke trenger å vise små detaljer i variasjonen i Norge. Det har vært gjort flere studier av hvordan romlig oppløsning spiller inn på deteksjon av sykdomsutbrudd, blant annet av Ozonoff et al. i 2007. Forskjellen på å bruke eksakte posisjoner på sykdomstilfellene og å gå ned til en dårlig oppløsning, er markant. Tre simuleringer viste at sannsynligheten for å oppdage ansamlinger med tilfeller falt fra 100 % til rundt 40 % ved skifte fra eksakte posisjoner til grov oppløsning. Dersom man bruker høyere oppløsning, øker også risikoen for å detektere falske utbrudd (Ozonoff et al. 2007). For å gjøre analyser i Norge har man flere ulike nivåer med romlig oppløsning ettersom hvor detaljerte analyser som skal utføres. Hvert fylke er gitt et fylkesnummer og består av flere kommuner, som igjen har et kommunenummer knyttet til seg. De ulike kommunene har minst en postsoner, med et entydig postnummer. Større kommuner er gjerne bygget opp av flere postsoner.

Et kritisk punkt når det gjelder studier og analyser av helseinformasjon er personvernet til pasientene som studeres. I likhet med influensadataene i denne studien er det vanlig å bruke punktdata for visualisering og plotting av de enkelte sykdomstilfellene. Dersom man bruker pasientenes bosted som lokaliseringpunkt, kan dette bryte med personvernet. Muligheten for etablering av et rammeverk for presentasjon av konfidensielle punktdata ble undersøkt av Leitner og Curtis i en studie fra 2006. I denne studien ble adressen til alle drapsofre i en gitt bydel brukt som grunnlagsdata. Forsøket gikk ut på å gi hvert adressepunkt en forskyvning ut fra et rutenettverk med sider på henholdsvis 100 m, 200 m og 350 m. Forskyvningen av punktene ble utført ved blant annet rotering, tilting og tilfeldig plassering innen cellen. Resultatene kan summeres på følgende måte:

1. Det ser ut til å finnes et inverst forhold mellom nivået av konfidensialitet og nivået av gjenkjenbarhet.
2. Umaskerte punkt mønstre blir gjenkjent forskjellig fra det geografisk maskerte mønsteret, dersom cellestørrelsen er større enn en terskelverdi.
3. Kartmålestokk, symbolstørrelse, datatypen er blant andre faktorer som kan påvirke strukturen.
4. Målgruppen og hvor lang tid datasettet studeres er viktige faktorer som påvirker konfidensialiteten. (Leitner og Curtis 2006).

### 3.2.3 Anvendelse av multikriteriaanalyser i helse og beredskap

Geografiske informasjonssystemer har mulighet for å kunne utføre multikriterieanalyser ettersom systemet har mulighet for innhenting, håndtering og analyse av romlige data (Chakhar og Martel 2003). Multikriteriaanalyser er ofte anvendt i beredskaps- og epidemiologisk sammenheng. I forbindelse med beredskap brukes analyser i GIS både når det gjelder naturfarer, som skred og flom, men også med tanke på eventuelle terroranslag og andre hendelser. Bell og Dallas utførte i 2007 en GIS-studie i forbindelse med beredskap ved terroranslag. Studien gikk på sårbarheten til befolkning og infrastrukturen til helsevesenet ved angrep med atomvåpen. Til denne analysen brukte Bell og Dallas (2007) data om befolkningstallet i byene om natten, informasjon om sykehusene i området og vær og klimatiske data. I tillegg ble det lagt inn variabler med hensyn på størrelsen av våpen og funksjoner som kalkulerer radioaktiv- og termiskstråling fra detonasjonene, og andre igjen som beregner atmosfæriske effekter (Bell og Dallas 2007).

### 3.3 Interpolasjon

Når man skal bruke data for å gjøre analyser, er det ofte ønskelig å ha kontinuerlige flater som viser endringer i dataene, for eksempel hvordan nedbøren er i et område mellom to værstasjoner. Det er imidlertid ikke mulig, eller særlig kostnadseffektivt, å registrere data fra absolutt alle punkter i området man ønsker å studere. For å løse dette er det utviklet flere metoder for å beregne verdien i et gitt nytt punkt, ut fra allerede eksisterende data. Denne beregningen kalles interpolasjon. De ulike metodene som brukes for interpolering av verdier kan gjerne deles opp i undergrupper basert på prinsippene som ligger bak metodene. I sin lærebok fra 1986 introduserte Bourrogh begrepene diskret og kontinuerlig interpolasjon. Dette er bare en av mange ulike inndelinger (Burrough et al. 1998).

Diskret interpolasjon vil si at dataene som blir interpolert blir ansett som å være diskrete, altså at de endrer seg brått i en gitt grense.

I motsetning til diskret interpolasjon, har vi kontinuerlig interpolasjon. I følge Burrough et al (1998) betyr dette at dataene endrer seg gradvis over en distanse, og at det ikke forekommer plutselige endringer. Slike interpolasjonsmetoder deles igjen opp i lokale- og globale metoder etter hvilke kjente punkter som blir tatt med i betraktningen. Lokale metoder bruker kun de kjente punktene som ligger i umiddelbar nærhet til det nye punktet, mens globale metoder tar hensyn til alle de kjente punktene når en ny verdi skal beregnes (Burrough et al. 1998).

### 3.4 Romlig avhengighet og autokorrelasjon

Et av de grunnleggende elementene innen geografi er romlig avhengighet; og tilsier at objekter som ligger i nærheten av hverandre har flere likheter enn de objektene som er lengre borte, og måles i følge Lee og Wong (2001) med hensyn på styrken av sammenheng. Denne antagelsen kan være av betydning for hvordan influensaepidemien utvikler seg i Norge i løpet av en gitt sesong. I følge Lee og Wong (2001) finnes det ulike metoder å analysere romlig avhengighet på. For å oppdage spesielle mønstre i et punktdatasett kan man bruke ulike metoder; hvor noen tar hensyn kun til punktenes plassering, og igjen andre metoder som tar hensyn til både punktenes posisjon og punktenes attributter (Lee og Wong 2001).

En av metodene som brukes for å finne autokorrelasjon i et datasett er Morans I indeks, og det er i følge Lee og Wong (2001) når dataene har informasjon på interval- eller ratioform at Morans I er best å benytte. Dette er en geostatistisk indeks som forteller om den romlige



sammenhengen i datasettet. I følge Basara og Yuan (2008) er verdiene som blir beregnet ut ifra metoden i størrelsesorden fra -1 til +1, og skal tolkes som korrelasjonskoeffisienter. Dette medfører at en verdi på +1 tilsier at dataene er samlet, mens ved -1 er dataene spredt (Lee og Wong 2001, Basara og Yuan 2008). For å utføre tolkninger av indeksverdiene, er det i følge Lee og Wong (2001) vanlig å sette opp en hypotese og teste denne ved hjelp av en Z-variabel. Z-verdien forteller oss hvor mange standardavvik observasjonen ligger fra gjennomsnittet, og i hvilken retning; den brukes også til å fastslå om man hypotesen er korrekt eller om den skal forkastes. De observasjonene som er større enn gjennomsnittet er positive, og observasjoner mindre enn gjennomsnittet er negative.

### 3.5 Visualisering av geografiske data

For å presentere geografiske data er det bedre å bruke grafiske og kartografiske virkemidler enn i tabell eller tekstform. Rene geografiske objekter som har en gitt posisjon i rommet, blir gjerne presenter ved objektets X- og Y-koordinat og et beskrivende symbol. Det er når objektene har mer informasjon det kan diskuteres hvordan kartografien skal være for å visualisere objektene på en best mulig måte. I følge Krak og Ormeling (1996) vil fargene som blir brukt for å representere elementer på et kart bli oppfattet på forskjellig måte, og både psykologiske og fysiologiske faktorer spiller inn. Fremgangsmåten for å lage en kartografisk korrekt presentasjon kan deles opp i tre hoveddeler og består av disse stegene (Kraak og Ormeling 1996):

1. Bestem hvilket type kart som skal lages. Mulige karttyper er blant annet kolorpleth, punktkart, isobarkart og statistisk overflate.
2. Begrens antall klasser av verdiene. Ved å bruke mange klasser på informasjonen som skal presenteres, kan det bli vanskelig å skille mellom de ulike klassene ved bruk av fargegradienter.
3. Bestem klassenes minimum- og maksimumsverdier. Det er nødvendig å velge hvilke verdier som skal presenteres i hver klasse, dette foregår ofte automatisk.

Geografiske data endrer seg ofte gjennom et tidsintervall. En av fordelene ved å bruke et geografisk informasjonssystem er at en har muligheten til å animere endringene i datasettet. I følge Kraak og Ormeling (1996) kan animasjoner brukes både når det forekommer en direkte

kobling mellom visningstiden i animasjonen og reell tid, og det forekommer bruk av animasjon hvor det ikke er noen sammenheng mellom visningstid og virkelig tid. De animasjonene som har temporale data i seg, kan i følge Midtbø et al. (2007) igjen deles opp i sykliske- og lineære animasjoner. Disse klassifiseringene gjenspeiler om datasettet som animeres går gjennom samme tidsperiode en eller flere ganger.

En måte å presentere data på som har blitt mer vanlig etter at datamaskinene har blitt mer grafiske, er tredimensjonale representasjoner. Tredimensjonale kartrepresentasjoner kan være med på å uttrykke viktige verdier i dataene, og man kan uttrykke flere variable på en gang ved for eksempel å bruke en variabel for høyde på en søyle og fargen til søyla for temperatur. Problemet med å kunne representere flere variable samtidig er at brukeren av kartet kan for mye informasjon til å kunne tolke det riktig (Krisp og Fronzek 2003).

Ved bruk av tredimensjonal visualisering er det mulig å representere både enkeltpunkt og overflater. Curtis (2008) publiserte en artikkel i *International Journal of Health Geographics* som omhandler studier av utbruddet med gulfeber i New Orleans i 1878. Ut fra sykdomsdata ble det dannet overflater som viser områder med høye forekomster av syke. Dette ble kombinert med demografisk informasjon om bydelene i New Orleans, og analyser ble utført. Det viser seg at sykdommen spredte seg med en sammenheng mellom kulturell avstand og geografisk avstand (Curtis 2008).



Figur 3.2: Illustrasjon av en sirkulær animasjon av temperatur i amerikanske byer gjennom ett år (Midtbø et al. 2007).

## 4 Materiale

### 4.1 Geografiske datasett

I et geografisk informasjonssystem ligger det mange datasett med ulik informasjon som kan kobles sammen og på denne måten visualisere eller analysere det ønskede fenomenet. I Norge er det flere ulike private selskap og offentlige organisasjoner som står for kartleggingen og oppdateringen av ulike geografiske data. Disse selskapene og organisasjonene samarbeider om Norge Digitalt, hvor dataene publiseres for at de andre aktørene kan benytte seg av dataene. Firmaer og organisasjoner som ikke er medlem i Norge Digitalt, har mulighet til å kjøpe disse dataene via kommersielle leverandører.

Denne oppgaven skal baseres på analyse av influensadata på stor skala, ettersom det eksisterer data fra store deler av Norge. Det er derfor valgt å benytte data fra N2000-serien til Statens Kartverk. Datasettene som benyttes på denne skalaen er fylkesflater og kommuneflater. Disse datasettene er klippet slik at de tar med detaljer om kystlinjen, det finnes også data som representerer hele utstrekningen til kommunen og fylket. For visualiseringer og kartografisk presentasjon vil det i tillegg bli brukt ulike data som blant annet symboliserer vann og veier. Datasettet som inneholder informasjon om vannflatene i Norge er hentet fra N1000-settet. For å knytte influensadataene sammen med de relevante kommunene er det valgt å representere dataene som punkter, ettersom dette gjør det enklere med senere analyser. For å representere kommunene som punkter tar studien utgangspunkt i et datalag hvor det er plassert punkt i det geometriske midtpunktet til kommunen. Hvis det skulle være ønskelig å gå videre inn i dataene og se på spredning innenfor en gitt kommune, er det tilgjengelig et punktdatasett som inneholder midtpunktet til hver postzone.

Gjennom analysen som skal gjøres er N2000-dataene brukt på grunn av at de allerede er klippet til kystlinjen og derfor gir et bedre bilde av Norge. Fylkes- og kommunegrensene som ligger i N1000-settet er mer detaljert enn N2000. Det hadde derfor vært en fordel å bruke N1000 til analysene dersom dataene hadde vært klippet til kystlinjen. Det er imidlertid ikke stor forskjell i avstand mellom grensene i de to målestokkene.

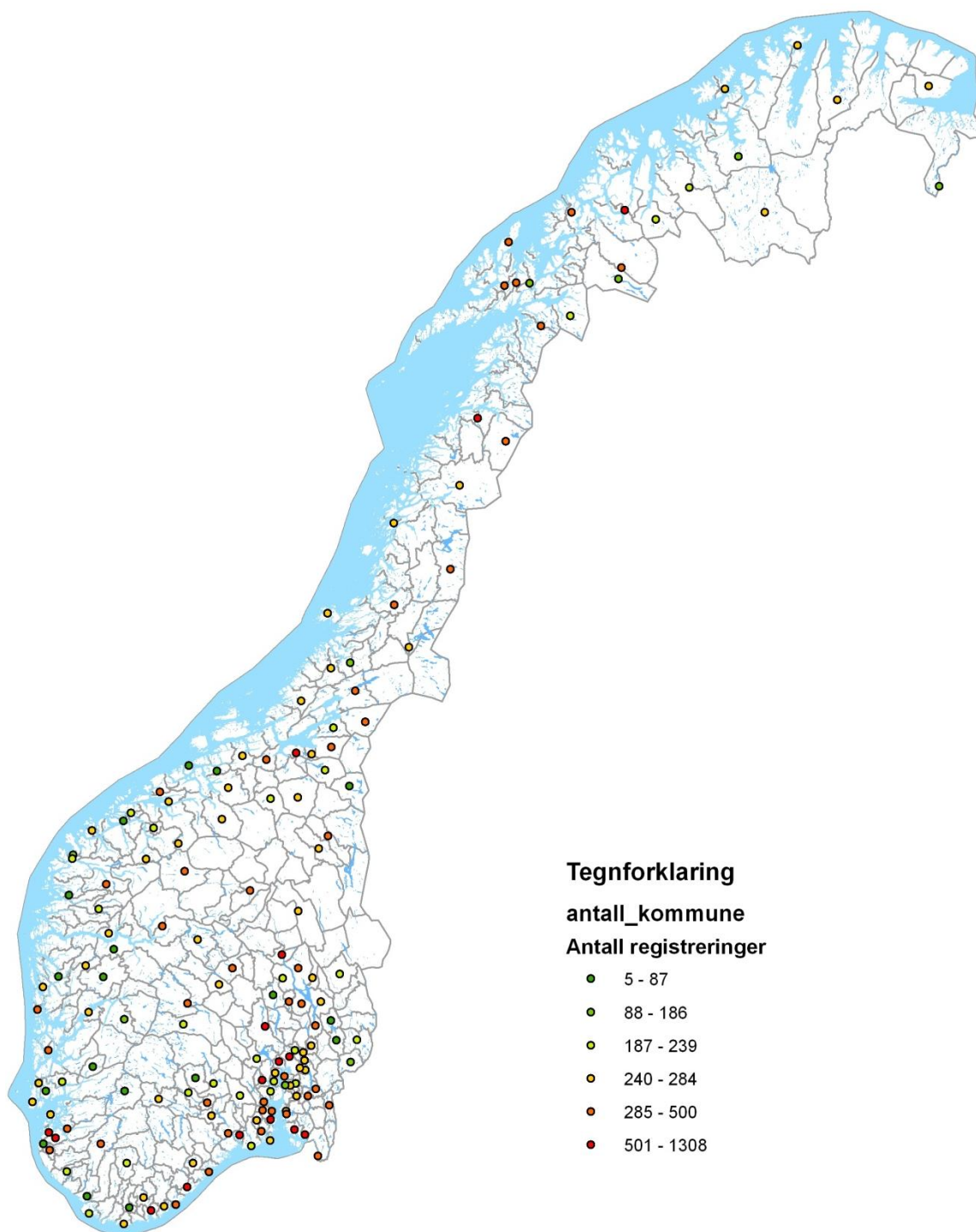
Alle kartlagene som brukes i analysene er oppdaterte med dagens kommunenavn og kommunenummer. Dette er viktig ettersom kommunene i Norge ikke er statiske.

## 4.2 Influensadata

FHI får i vintersesongen tilsendt data fra 201 legekontorer om antall pasienter med influensaliknende symptomer som har oppsøkt kontorene ukentlig. Dataene blir lagt inn i en database, og resultater blir publisert hver 2. uke i MSIS-rapporter. Influensadataene er samlet inn i perioden 1998 til 2007, og representeres ved kommunenummer. Influensautbruddene kommer spesielt i høst- og vintersesongen, så dataene i studien gjelder fra og med uke 40 inneværende år til uke 20 påfølgende år. Den totale mengden av influensadata er på om lag 48000 registreringer.

Alle registreringer har et unikt løpenummer som består av en kobling mellom år, uke og meldernummer. Meldernummer er en identifikator for den enkelte lege som har fylt ut kortene og hatt pasientene. I tillegg består dataene av; en kolonne med antall tilfeller av sesonginfluensa, en med antall konsultasjoner legen har hatt den uken det gjelder, fylkesnummer som viser hvilket fylke legekantoret ligger i, og dato for registrering av kortet.

En utfordring når det gjelder influensadataene er kommunenummeret. For kombinasjon mellom influensadata og GIS, skal det benyttes kommunenivå, og vi har ett fullstendig oppdatert datasett med dagens kommuner. Utfordringen ligger i at kommunene i Norge ikke er statiske, slik at en del av kommunenumrene fra 1998 og fremover ikke eksisterer lenger. Det er derfor nødvendig å lage en oversikt over hvilke kommuner som er slått sammen, og hvilke gamle kommunenummer de nye kommunenumrene representerer. De influensaregistreringene som er registrert med dagens kommunenummer er blitt plottet på kart for å få oversikt over den geografiske distribusjonen (Figur 4.1). Som man kan se ut fra kartbildet er det en god spredning på distribusjonen av data, noe som gjør at modellen blir mer nøyaktig for store deler av landet. Etter å ha studert dataene kom det frem at det i løpet av tidsserien har kommet til flere kontor, samt at det ikke er registrert data for alle kontor i alle ukene. For å få et overblikk over hvor mange data hver kommune har registrert, har det blitt dannet et slikt oversiktsbilde:



Figur 4.1: Oversiktskart over kommuner med influensaregistreringer og antallet registreringer som finnes i den enkelte kommune.

### 4.3 Feilkilder

Når man arbeider og analyserer data, er det nødvendig å ta hensyn til eventuelle feilkilder og feil som kan befinne seg i dataene. Influensaregistreringene som er tatt i bruk, er registrert manuelt i en database, og derfor vil det også blant disse kunne finnes ulike feil. Dette kan gjelde blant annet registreringsfeil (to tall byttes om), dobbeltlagring dersom databasen ikke er optimalisert og hvor vidt registreringene er gyldige for å representere andel ILS.

Registreringene er som allerede beskrevet i kapittel 3.1 lagret på denne måten:

Lpnr	År	Uke	Meldenr	KNR	FNR	INFLU	KONS	Reg.dato
200442/010305	2004	42	010305	0103	12	0	150	06.12.2004
200513/250001	2005	13	250001	2500	14	2	100	05.05.2005

Tabell 4.1: Eksempel på influensaregistreringer

Ettersom det kommer inn data fra relativt mange kontor hver uke, vil det være en stor jobb å registrere disse manuelt. Det er ikke uvanlig at man ved registrering av store datamengder kommer i skade for å taste for eksempel 42 i ukenr når det egentlig skulle være 43. Slike feil burde naturligvis bli fanget opp av databasesystemet dersom det allerede ligger inne en registrering på uke 42. Dette viser seg at ikke er tilfellet med databasesystemet som er brukt for behandling av disse data. Ved søking etter dobbellagring gjennom en SQL-spørring som omfatter år, uke, meldenr, kommer det frem at det finnes rundt 150 like registreringer (300 duplikate). Ved å gå gjennom disse duplikatene og korrigere de på en god måte, kan man forbedre den originale kvaliteten på datasettet. Det reiser seg derimot flere usikkerheter rundt korrigeringen, ettersom dataene kan variere ganske mye.

Det første eksempelet på feil i datasettet kan vises som en slik tabell

Lpnr	År	Uke	Meldenr	KNR	FNR	INFLU	KONS	Reg.dato
200313/182001	2003	13	182001	1820	8	0	164	04.05.2003
200313/182001	2003	13	182001	1820	8	0	164	05.05.2003

Tabell 4.2: Eksempel på dobbellagring med hensyn på reg.dato

I dette tilfellet er de to registreringene fullstendig like i alle attributter bortsett fra registreringsdato. Dette tyder på at den samme registreringen er slått inn to ganger, kanskje av ulike personer som ikke vet hva den andre har registrert.

Lpnr	År	Uke	Meldenr	KNR	FNR	INFLU	KONS	Reg.dato
200313/182001	2003	13	182001	1820	8	0	164	04.05.2003
200313/182001	2003	13	182001	1820	8	4	243	12.12.2003

**Tabell 4.3: Eksempel på dobbellagring, med hensyn på influ, kons, regdato**

I tabellen over er det gitt et eksempel på feil som er litt verre å forklare. Denne registreringen har unik registreringsdato, antall sesonginfluensa og antall konsultasjoner, men de andre attributtene er like. Det er her nærliggende å anta at det dreier seg om en annen registrering enn uke 13, ettersom datoen for registrering er 12. desember i motsetning til 4. mai. En er i dette tilfellet nødt til å gå inn på det nevnte år og meldnr og se om det kan stemme at det er en uke som er utelatt fra datasettet. Enten så finner vi at det er mangler en uke i november/desember, i motsatt fall er alle registreringene riktig plassert. Dersom det siste er tilfellet, står vi igjen med et problem. Hvor skal egentlig denne siste registreringen? Er det ny informasjon om den opprinnelige uka, eller er det en tastefeil i for eksempel år eller meldnr? Sjekker man dataene i relasjonen og finner ut at man står overfor en manglende uke, har jeg da valgt å endre ukenummer og løpenummer på den doble registreringen med ”best” dato. Når denne situasjonen har oppstått, har jeg sett på det nærmeste året for å se om uken mangler. Dette gjør den sjelden, så jeg har valgt å slette registreringen i de tilfellene jeg har vært usikker på. Slike feil som dobbeltlagring og andre anomalier i en database, er det relativt lett å unngå dersom man lager databasen i henhold til normaliseringsregler. Disse går ut på å dele opp databasens tabeller ut i fra gitte kriterier, slik at data som er avhengig av hverandre kommer i samme tabell uten å virke inn på andre avhengigheter. Dette bør gjennomføres dersom det fortsatt skal lagres sesonginfluensa data i denne databasen, slik at eventuelle brukere slipper å feilsøke alle data før bruk.

Ved å sortere dataene etter de ulike variablene ble det klart at flere legekantor var registrert med veldig få konsultasjoner i løpet av ei uke. For å få mest mulig korrekte data var det nødvendig å analysere hvor mange kommuner dette gjaldt og få plukket vekk disse registreringene. Det var da nødvendig å fastslå en terskelverdi for hva som er minste naturlige antall konsultasjoner en lege har i løpet av en arbeidsuke. En del registreringer er oppført med 1 konsultasjon, noe som tydelig er feil. En grense på 5 konsultasjoner ble valgt for å trekke ut mulige feilregistreringer. Etter å ha trukket ut registreringene kommer det frem at det er noen kommuner som går igjen med lavt antall konsultasjoner i forskjellige uker i løpet av perioden dataene spenner over. Oversikt over disse kommunene ble satt opp i Tabell 4.4 og registreringene ble fjernet fra databasen og filene som skulle brukes i GIS-analysene.

Kommune	1 Kons	2 Kons	3 Kons	4 Kons	5 Kons
1939	1				
1811	1				
1418				1	
0912	1		1	1	
1871		1			
1714					1
0807	1				1
1630				1	
0819			1	1	
0501				1	
0105	1		1	1	
1725		1			
0104				1	
0904					1
2022	1				
0419				1	
0219	2				

Tabell 4.4: Oversikt over kommunene som har 5 eller færre antall konsultasjoner registrert en uke i løpet av perioden.

En tredje og ikke minst like stor usikkerhet som er knyttet til dataene er befolkningen og synet på sesonginfluensa. Det er urealistisk å tro at alle som har sesonginfluensa går til legen og får tatt de bestemte prøver for å konstatere sykdommen, og at det er dette dataene viser. I mange tilfeller går symptomene over av seg selv, eller forveksles med forkjølelse; og fører til at mange unnlater å dra til legen og få sjekket det. Blant de av befolkningen som går til lege når de oppdager symptomer på sesonginfluensa, er det heller ikke sikker at dette blir konstatert eller stemmer overens med registreringene. Kraftig forkjølelse og sesonginfluensa har de samme symptomene, så det kan skje at legen registrerer disse typene symptomer som influensalignende symptomer uten å ta prøver. Dette kan igjen føre til at antallet influensatilfeller er høyere enn det som det egentlig er registrert, ettersom pasientene egentlig bare har en kraftig forkjølelse.



## 4.4 Bearbeiding og tilrettelegging av data

Når en arbeider med geografiske analyser og romlige data, er det gjerne behov for å bearbeide og tilrettelegge dataene for analysen som skal gjennomføres. Influensadataene som ble gitt som testdata var opprinnelig i tabellform i Excel-format. For å gjøre disse mer tilgjengelig for senere GIS-analyser ble det opprettet en database i Microsoft Access hvor influensadataene ble lastet inn, sammen med informasjon om landets kommuner. Bakgrunnsdataene som ligger i bunnen av analysen; fylkesgrenser og kommunepunkter var opprinnelig i shape-format som brukes av ArcGIS-programvaren, slik at disse ikke trengte videre konvertering.

Etter feilsøking og korrigerings av influensadata ble dataene deretter tatt inn i ArcGIS gjennom databasen og konvertert til shapeformat for lettere å kunne arbeide med dataene i et romlig format. Datafilen bestod da av alle de 50000 registreringene, noe som førte til at det ikke var hensiktsmessig å jobbe med denne enkelte filen. Det ble derfor utviklet to script som deler denne filen opp i flere mindre filer for videre analyser. Først ble filen splittet opp i en fil for hver influensasesong. Script nummer to ble kjørt med katalogen hvor sesongfilene lå, for automatisk kunne dele opp disse i mindre filer igjen basert på ukenummeret for de forskjellige influensadataene. Disse filene ble til slutt stående igjen:

Filnavn	Beskrivelse
<b>Registreringer.shp</b>	Samlet fil med alle de 50000 influensaregistreringene som er samlet inn.
<b>Sesong_#.shp</b>	Til sammen 9 filer med ulikt sesongnummer
<b>Uke_sesong#_uke#.shp</b>	Til sammen X filer som inneholder influensaregistreringene som er foretatt i den aktuelle uken for den aktuelle sesongen.

Tabell 4.5: Oversikt over produserte shape-filer fra influensadata

## 4.5 Programvare

Innen geografiske informasjonssystemer finnes det en rekke ulike programvare for analyser og visualisering av geografisk informasjon. Det finnes programvare for ulike operativsystemer, inkludert både Windows og Linux, samt både kommersiell programvare og fri programvare. Den markedsledende leverandøren for GIS programvare er det amerikanske ESRI (Environmental Systems Research Institute) som utvikler ArcGIS. For visualisering og bearbeiding av materialet er det ArcDesktop som er benyttet. Dette programmet inneholder en hel rekke funksjoner som er viktig for analysen. For å få tatt i bruk flere typer analyser, som

for eksempel interpolering og geostatistikk, finnes det ulike moduler man kan installere i tillegg til Desktop-versjonen. Av disse modulene er det brukt SpatialAnalyst og GeoStatistics.

## 5 Analyse av data

### 5.1 Utvikling av modell og system

I dette kapitlet skal datasettet analyseres for å undersøke om det er egnet til å se sammenhenger i spredning, samt å kvalitetssikre dataene for bruk i signalsystemet.

Utviklingen og implementeringen av signalsystemet ble gjennomført ved at det ble utviklet en egen applikasjon som implementerte de nødvendige GIS- funksjonene for å detektere utbrudd. Selve systemet for deteksjon av signaler er ment kjørt som en bakgrunnsprosess og har derved ingen grafiske brukergrensesnitt. Det ble utviklet en støtteapplikasjon for å håndtere interaksjon med bruker ved endring av infeksjonsinformasjon. Utviklingen og implementeringen har blitt utført i programmeringsspråket *Python* med utviklingsmiljøet Eclipse for grafisk debugging av programmert kode. Programvaren som ble benyttet for å analysere datasettet med hensyn på romlig sammenheng er i all hovedsak GIS-applikasjonen ArcMAP og tilleggsmodulene GeoStatistics og Spatial Analyst.

#### 5.1.1 Innhenting av eksterne data

Signalsystemet som ble utviklet i studien er basert på en database som skal implementeres ved Folkehelseinstituttet. Denne databasen skal inneholde informasjon om diagnoser på infeksjonssykdommene de enkelte leger stiller hver dag. Det er tenkt at oppdateringen av databasen skal foregå automatisk en gang i døgnet, hvor nye registreringer blir lagt inn i databasen.

De fleste av landets legekantor og spesialistkantor er med i Norsk Helsenet. Dette er en virksomhet som ble opprettet i 2007 og ble etablert for å sikre en felles standardisert plattform for elektronisk samhandling mellom de ulike helseaktørene i Norge (Helsenett 2009). Denne virksomheten leverer ulike tjenester; en sikker sone for samhandling mellom de enkelte aktører, samband og IT-sikkerhet, og et nasjonalt system for rekvirering av pasienttransport. Håndtering av pasientdata og journalføring ved dagens legekantor gjøres elektronisk, og det er spesielt to leverandører av IT-løsninger til helsetjenestene som er markedsledende. Det norskebaserte firmaet Profdoc utvikler programmet WinMed som er et omfattende journalsystem med mulighet for å motta epikriser, svar fra laboratorier, elektroniske henvisninger og bestilling av laboratorietjenester, samt røntgen- og epikrisebestillinger. Brukermassen til programmet WinMed innbefatter rundt 6000 brukere på norske allmennlegekantor, 120 spesialister, 265 helsestasjoner (Profdoc 2008). I tillegg til journalsystemet som utvikles av Profdoc, finnes Hove Medical Systems, som har rundt 1200

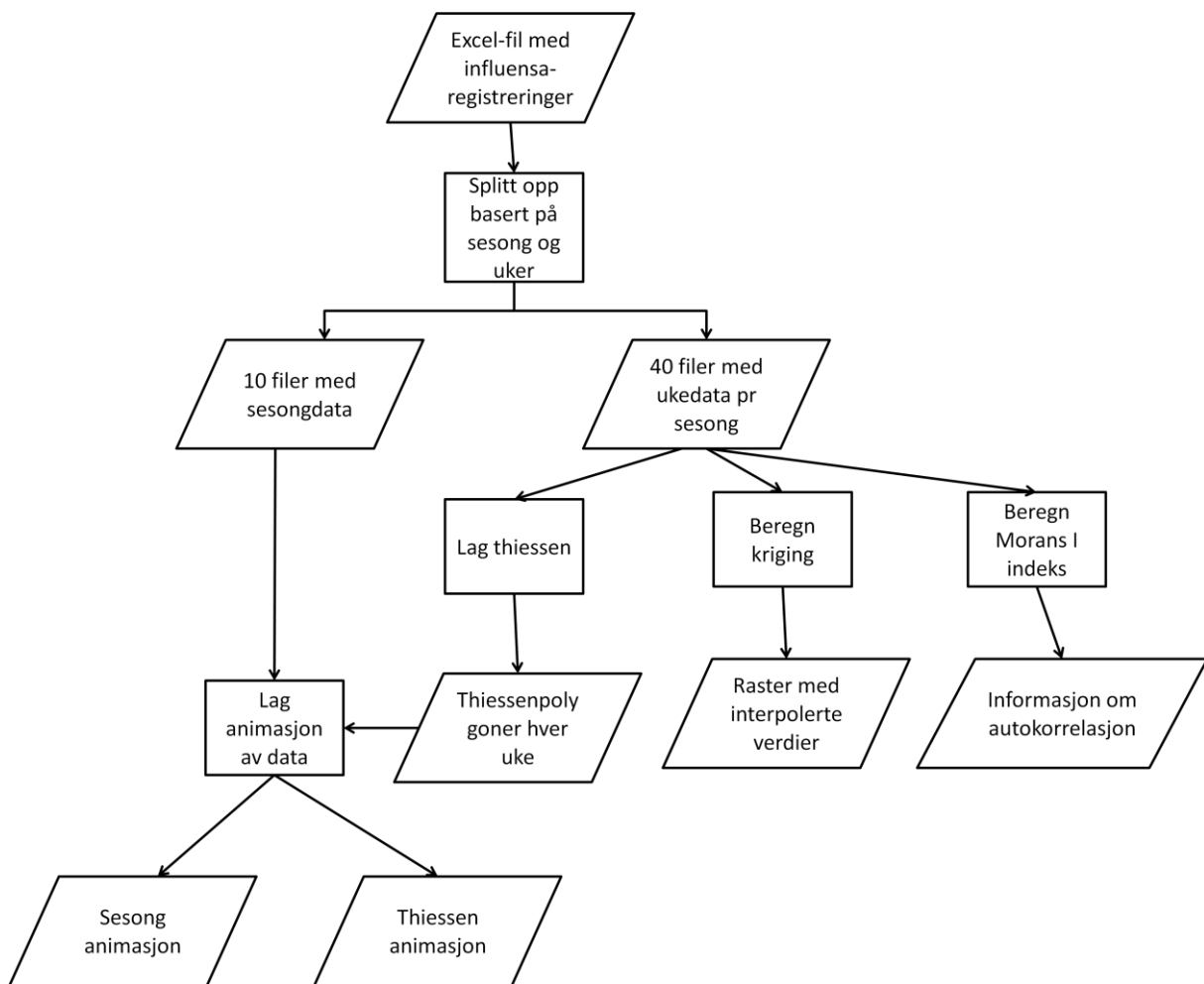
brukere i Norge. Dette systemet innehar også en funksjon som gjør det mulig med sending og mottak av SMS-meldinger i avtaleboka (Systems 2008).

Pr dags dato er ikke databasen som skal innhente denne informasjonen implementert, så det vil bli brukt data fra MSIS-databasen som testdata i tillegg til influensadataene. For at systemet som utvikles skal kunne tas i bruk ved Folkehelseinstituttet må databasen utvikles, samt at det blir laget en modul i de allerede eksisterende pasientjournaløsningene som brukes rundt om i landets legekantor. Dette er en langvarig og dyr prosess.

## 5.2 Analyse av influensadata

Et av de overordnede formålene for å analysere influensadataene er å se om det er mulig å fastslå om det finnes noen romlig korrelasjon mellom registreringene, samt å kvalitetssikre influensadatane for å se om disse kan brukes til signalsystemet. Når man ser på spredning er det ønskelig å studere informasjonen på så kort tidsoppløsning som mulig. Den minste tidsoppløsningen for disse dataene er uker.

Ut fra en fil med absolutt alle registreringer ble det valgt å utvikle et script som delte opp dataene i tidsintervaller for videre studier. Intervallene som ble valgt er sesong og uke. For å dele opp ei fil i ulike deler er det noen operasjoner som må gjøres gjentatte ganger, og i den forbindelse er det hensiktsmessig å benytte scripting. Scriptet starter med å lese inn filen med alle registreringene, og tabellen i shapefilen får opprettet og beregnet et attributt for andel ILS. Det neste som skjer er at alle unike verdier for kolonnen med sesong blir fastslått. Det blir da laget en liste med unike verdier, og scriptet løper gjennom listen og skriver alle tuplene med den aktuelle sesongen til en egen fil. Når alle unike sesonger er ferdige, leser scriptet inn hver enkelt sesongfil. For hver sesong blir det valgt ut unike verdier i attributtet ukenummer, og det blir dannet en fil for alle tupler med dette attributtet. Etter at scriptet har kjørt, står vi igjen med filer som har velegnet oppløsning for videre studier.



**Figur 5.1:** Flyttdiagram for analyse av influensadata. Starter med rådata, før man splitter denne opp i flere ulike filer og begynner med ulike analyser som blant annet thiessenpolygoner og kriging. Det blir laget animasjon av datasettene.

For å finne eventuelle trender og romlige mønstre i datasettet ble det valgt å benytte to ulike interpolasjonsmetoder for å beregne en overflate med denne ukes andel ILS for hele Norge. Interpolasjonsmetodene som ble valgt var den diskrete metoden Thiessenpolygoner, og den geostatistiske metoden Ordinær Kriging. Beregningen av thiessenpolygonene ble utført via en batch-funksjon i ArcGIS, noe som vil si at man kan gjøre en beregning på flere datasett samtidig. Minimumsverdi og maksimumsverdi ble nå trukket ut fra hver periode for å kunne gi thiessenpolygonene en farge som kan sammenlignes innen hver sesong. Deretter ble thiessenpolygonene animert ved hjelp av en funksjon i ArcGIS (se animasjoner på vedlagt CD).

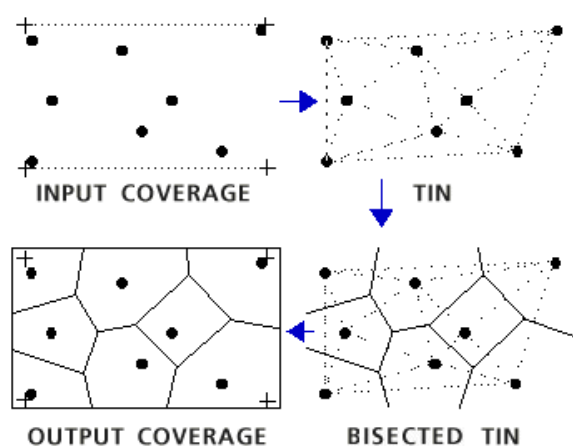
Beregning av thiessenpolygoner kan altså gjøres automatisk ettersom det ikke er noen spesielle variabler som må tas hensyn til for de ulike filene. Kriging derimot er en metode som er lite egnet å automatisere. Dette er på grunn av at det blir beregnet spesielle variabler for hver enkelt fil, og som det kan være hensiktsmessig å studere nærmere for å kvalitetssikre dataene. Derfor har jeg valgt ut enkelt uker gjennom hver sesong og kjørt kriging funksjonen på disse for å studere datasettet.

For å beregne ulike typer statistikk om dataene ble det brukt funksjoner i henholdsvis ArcGIS og Microsoft Excel. Det ble også implementert ett script som automatisk beregnet Morans I indeks og z-verdier for alle ukene. Dette scriptet er en noe modifisert versjon av et eksempelscript som ligger i hjelpdokumentet til ArcGIS.

Som vist i figur 4b, så vil det være diskrete verdier for andel ILS i kartet. Her dukker spørsmålet ”*hva skjer med andel ILS i grenseområdene?*” opp.

### 5.2.1 Thiessenpolygoner

Beregning av nye verdier ut fra thiessenpolygoner er et eksempel på en diskret interpolasjonsmetode. Denne metoden foregår ved at et område i bildet blir delt opp slik at verdien i området er basert på det nærmeste datapunktet. Algoritmen som blir brukt ved generering av thiessenpolygoner foregår ved at for ethvert punkt i bildet blir tilordnet en *voronoi-region*. Denne flaten består av alle punkter som er nærmere det gitte punktet enn alle andre punkter i bildet.



Figur 5.2: Algoritme for dannelse av thiessenpolygoner, her vist ved et coverageformat til ArcGIS (ESRI 2009).

Algoritmen som brukes for å danne thiessenpolygonene er vist i Figur 5.2. Det som først skjer er at punktene blir skannet fra venstre topp til høyre bunn. Dersom man har angitt en terskelverdi blir punktene utenfor denne verdien ignorert. Deretter blir punktene triangulert på en slik måte at de er i samsvar med Delaunay kriteriene. Normalene for hver kant i trianguleringen blir fastslått, og danner utgangspunktet for kantene til thiessenpolygonene. Til slutt blir posisjonen til de opprinnelige punktene omgjort til posisjonen som skal brukes for etikettene til polygonene (ESRI 2009).

### 5.2.2 Geostatistikk og statistiske metoder

Interpolasjonsmetoden kriging baserer seg på geostatistikk og hører til under kategorien lokal kontinuerlig interpolasjon. Metoden ble laget av den sørafrikanske geologen D.G. Krige som ønsket å estimere en overflate på bakgrunn av enkelte målepunkter så nøyaktig som mulig. Kriging baserer seg på teorien om at den romlige variabelen baseres på de tre komponentene; strukturell komponent, en stokastisk og tilfeldig komponent, og hvit støy (Burrough et al. 1998).

Når en arbeider med romlige datasett av naturlige fenomener, er det ofte en underliggende trend i dataene. Det at det forekommer en trend i informasjonen vil si at verdiene har en tendens til å utvikle seg på en bestemt måte og i en gitt retning. Denne påvirkningen av dataverdiene ønsker man gjerne å få visualisert for å få ut mer informasjon om datasettet, samt fjerne denne påvirkningen for videre analyser på de grunnleggende dataene. Denne analysen kalles en trendoverflateanalyse, ettersom eventuelle trender i datasettet blir visualisert som en kontinuerlig overflate. Resultatet fra denne analysen brukes videre i krigingen, og er nødt til å ta stilling til om det kan finnes trender i dataene og hvordan denne trenden ser ut. Hvis man antar at det finnes en trend i datasettet, og at denne trenden er konstant, kan det være lurt å benytte seg av metoden som kalles ordinær kriging. Trenden kan også være uttrykt som en lineær funksjon av de kjente punktene. For å få en mer korrekt beregning av datasettet er det nødvendig å fjerne denne trenden før man lager variogrammet.

Dersom man plotter semivariansen mot distansen, får man et såkalt eksperimentelt variogram. Variogrammet gir verdifull informasjon for interpoleringen, og gjenkjenning av romlige mønstre. I variogrammet fremkommer det elementer som er vesentlig for generering av modellen for interpolasjonen. Det punktet hvor verdiene flater ut kalles *sill*, og tilsier at fra denne verdien finnes det ikke noe romlig avhengighet mellom datapunktene. Intervallet

mellom starten av kurven og frem til punktet hvor sill forekommer, kalles *range*. Dette elementet beskriver hvordan differansene mellom punktene er romlig avhengig. Jo nærmere hverandre punktene er innenfor range, jo større romlig avhengighet er det mellom punktene. Det tredje elementet i semivariogrammet er nugget, som er variansen av målefeilene kombinert med feilene fra den romlige variasjonen på kortere distanser enn avstanden mellom datapunktene (Webster og Oliver 2001).

De fleste undersøkelser hvor variogram benyttes for å studere et datasett baserer seg på nugget-verdien på grunn av at forskjellen mellom nugget- og sillverdien representerer den delen av datasettets varians som kan bli modellert med tanke på romlig variabilitet. Dersom funksjonen i et semivariogram er fullstendig flat, vil det si at det ikke finnes romlig sammenheng mellom dataene i datasettet på den benyttede målestokken (Liebhold og Sharov 1998).

Ettersom andel ILS i de ulike kommunene endrer seg fra uke til uke, kan det være at dataene er autokorrelerte. For å teste om dette stemmer med datasettet i studien, brukes en funksjon i ArcMap for å få beregnet Morans I indeks. I dette tilfellet antar vi at den romlige sammenhengen gradvis blir mindre med avstanden, altså at dataene er omvendt proporsjonale. Resultatet fra denne funksjonen blir en z-verdi og en verdi som representerer indeksen. Z verdien er den verdien en Morans I indeks må overstige for at hypotesen kan forkastes. Hypotesen som skal testes fremsettes som en nullhypotese, og vil i denne sammenhengen være: *andel ILS er tilfeldig fordelt utover landet og det finnes ingen romlig sammenheng.*

For å undersøke om datasettet har noen spesiell romlig autokorrelasjon ble det utviklet et script som gikk gjennom alle ukene i hver sesong og beregnet de nødvendige variablene. For å kunne sammenligne disse variablene med informasjonen fra den ordinære krigingen, ble det valgt å plukke ut informasjon fra 5 uker; ukene 40, 50, 1, 10, 20.

Interpolasjonsmetoden kriging gir oss også en indikasjon på hvordan dataene forholder seg romlig til hverandre. Ved hjelp av den grafiske interpoleringsmetoden som er tilgjengelig i modulen GeoStatistics fra ESRI, får man plottet et semivariogram. Semivariogrammet brukes for å studere data med tanke på romlig sammenheng og man får beregnet verdier i områdene som ligger mellom de originale punktene. For å få best mulig resultat av interpolasjonen er det mulig å velge om man skal fjerne trender i dataene. Trenden til et datasett vil si at det forekommer en romlig fordeling over hvor det er høye og lave verdier. Trender i datasettet



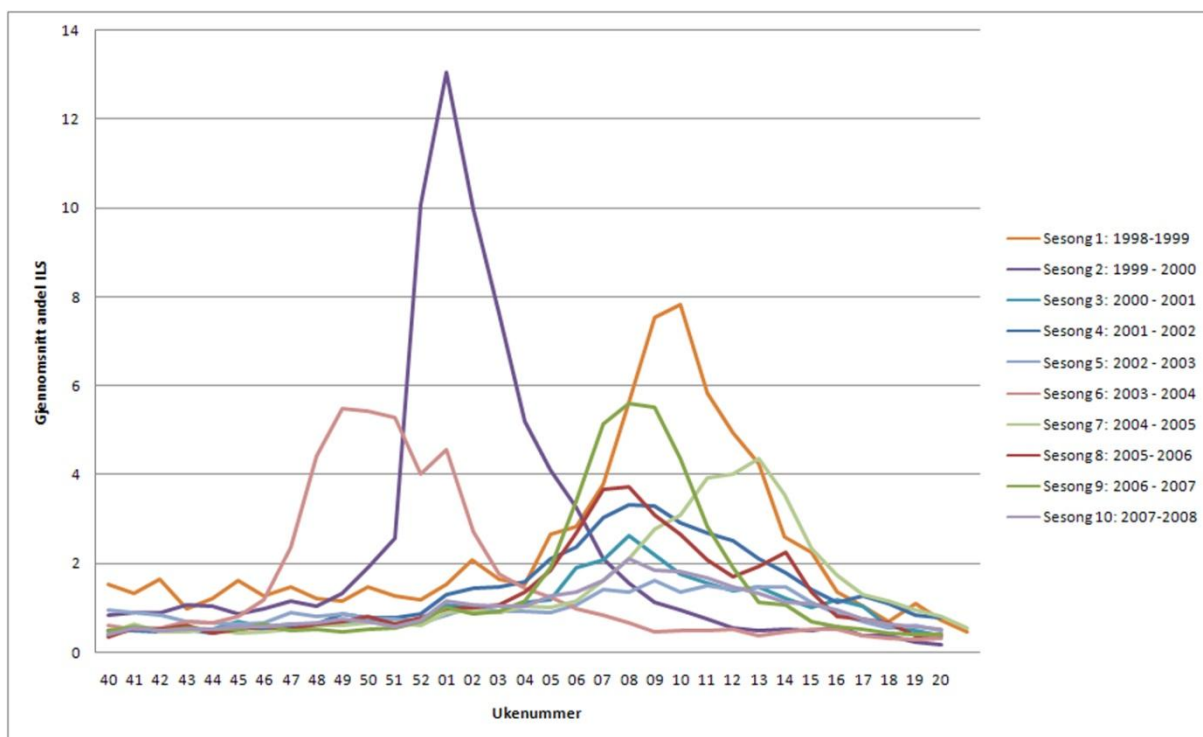
blir fremstilt som en matematisk funksjon med varierende potens etter hvor stor effekten av trenden er. En trendanalyse ble kjørt for å gi informasjon om andel ILS i de forskjellige ukene er påvirket av trender. Denne analysen ble gjort ved et pythonscript og det ble testet på de 5 tilsvarende ukene som nevnt ovenfor. Det ble fastslått at fjerning av førsteordens trend var det beste for å behandle andel ILS.

For å få en god oversikt over hvordan dataene korrelerer og hvordan utviklingen av andel ILS er gjennom de ulike sesongene ble det gjort noen statistiske beregninger. Ved å beregne den gjennomsnittlige andel ILS for landet i hver uke kan det være mulig å undersøke om det finnes noen likheter i hver sesong. Vil det være noen spesielle uker hvor andel ILS blir markant høyere, eller er det tilfeldige uker hver sesong? En annen type informasjon som er interessant å vite i denne forbindelsen er forholdet de gjennomsnittlige minimums- og maksimumsverdiene i hver sesong hvordan seg til hverandre. Er det noen sesonger som skiller seg ut, eller ligger minimums- og maksimumsverdien stort sett på samme nivå i hele perioden? Ved bearbeiding av datamaterialet, ble det laget filer bestående av alle uker i en sesong, samt filer for hver uke. Ved å bruke data fra filene med informasjon om gjeldende uke ble det beregnet gjennomsnittsandel ILS; mens minimums- og maksimumsverdiene ble hentet ut fra filen med informasjon om alle ukene i en sesong.

I de 10 sesongene som dataene strekker seg over kan det være slik at det er ulike kommuner som har sendt inn informasjon regelmessig. For å få klarhet i hvor mange kommuner det er registrert data fra blir dette talt opp og satt i et diagram. Er antallet kommuner jevnt fordelt over årene eller vil det komme til flere kommuner etter hvert som prosjektet har vart lenger?

### 5.3 Resultater av analysene

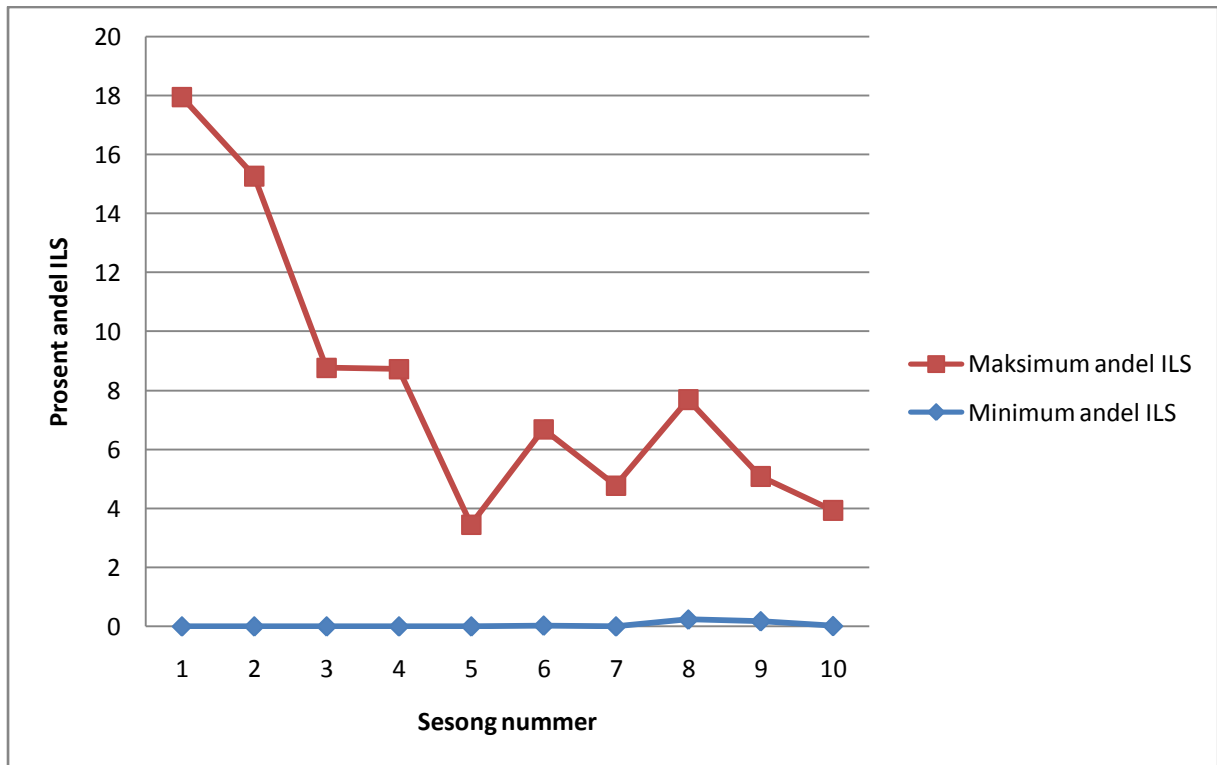
For å få en god oversikt over hvordan dataene forholder seg til hverandre og hvordan influensaen utvikler seg, ble det laget ulike statistikker. Andel ILS på utbruddene av sesonginfluensa vil naturlig nok variere gjennom en sesong, og det er ofte noen uker som har høyere forekomst av sykdommen. I denne forbindelsen dukket følgende spørsmål opp: ”*finnes det noen sammenheng mellom andel ILS for hver sesong?*” For å få testet og kanskje finne svar på denne hypotesen ble den gjennomsnittlige andel ILS for landet hver uke i sesongen trukket ut og modellert i et linjediagram, se Figur 5.3.



**Figur 5.3: Endring i gjennomsnittlig andel ILS for Norge for hver uke i de ulike sesongene datasettet rekker over. Sesongene 1998 til 1999 (Sesong 1) og 2004 til 2005 (sesong 7) er registrert med både uke 1 og uke 53, derfor er disse linjene lengre enn de andre.**

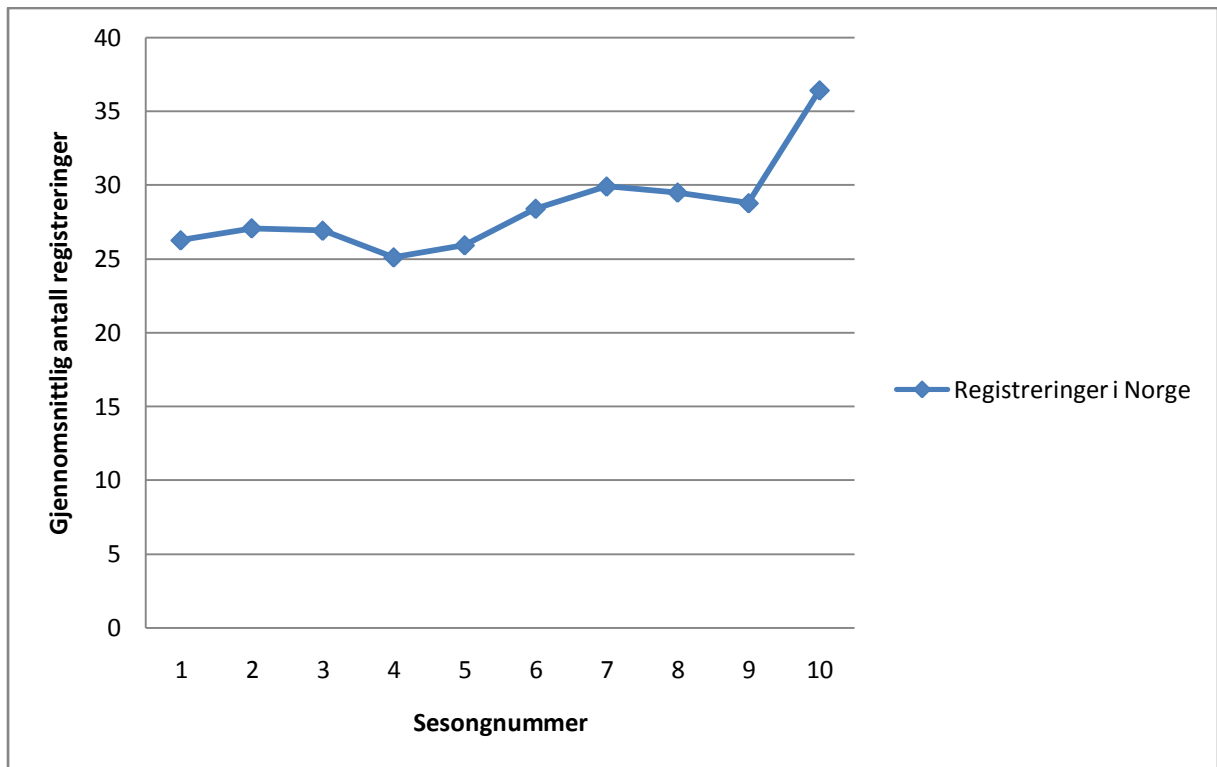
Hver sesong ble symbolisert med en enkelt farge for lettere å kunne skille de ulike sesongene fra hverandre. Utviklingen til kurvene i figuren tyder på at det finnes en sammenheng mellom de ulike sesongene, ved at det er en periode med mye smitte mellom uke 4 og uke 14 i året. Det er imidlertid tre sesonger som skiller seg markant ut med henholdsvis høy andel ILS og tidligere forekomst av maksimumsperioden. En mulig forklaring på den høye andel ILS er at denne sesongens virus var mer virulent enn vanlig, slik at flere ble syke. Figur 5.4 viser et diagram over minimums- og maksimumsverdier for gjennomsnittlig andel ILS gjennom sesongene. Ut i fra denne kan man se at minimumsverdien ligger stabilt rundt 0 bortsett fra en liten økning i sesongene mellom 2005 til 2007 (sesong 8 og 9). En faktor som påvirker at minimumsverdien er så lav er det faktum at ikke alle kommuner har meldt om registreringer hver uke, og noen av de kommunene med få registreringer kan ha 0 tilfeller av sesonginfluensa de få ukene de har meldt inn. Maksimumsverdien for andel ILS finner man her i 1998-1999 (sesong 1) hvor man har et maksimum på 18 % og maksimumsraten går jevnt nedover frem til 2002-2003 (sesong 5) hvor det deretter blir litt variasjon mellom rundt 4 % til 8 %. Noen av de samme feilkildene som med minimumsverdien kan også finnes for maksimumsverdier, altså at kommuner med få registreringer kan ha stor innvirkning på

gjennomsnittet. Dette er spesielt gjeldende for enkelte kommuner hvor det er veldig lavt antall konsultasjoner hver uke. I disse kommunene vil hver diagnostisert pasient ha stor innvirkning.



**Figur 5.4:** Diagrammet viser den maksimale- og minimale gjennomsnittsverdien for andel ILS for hver sesong.

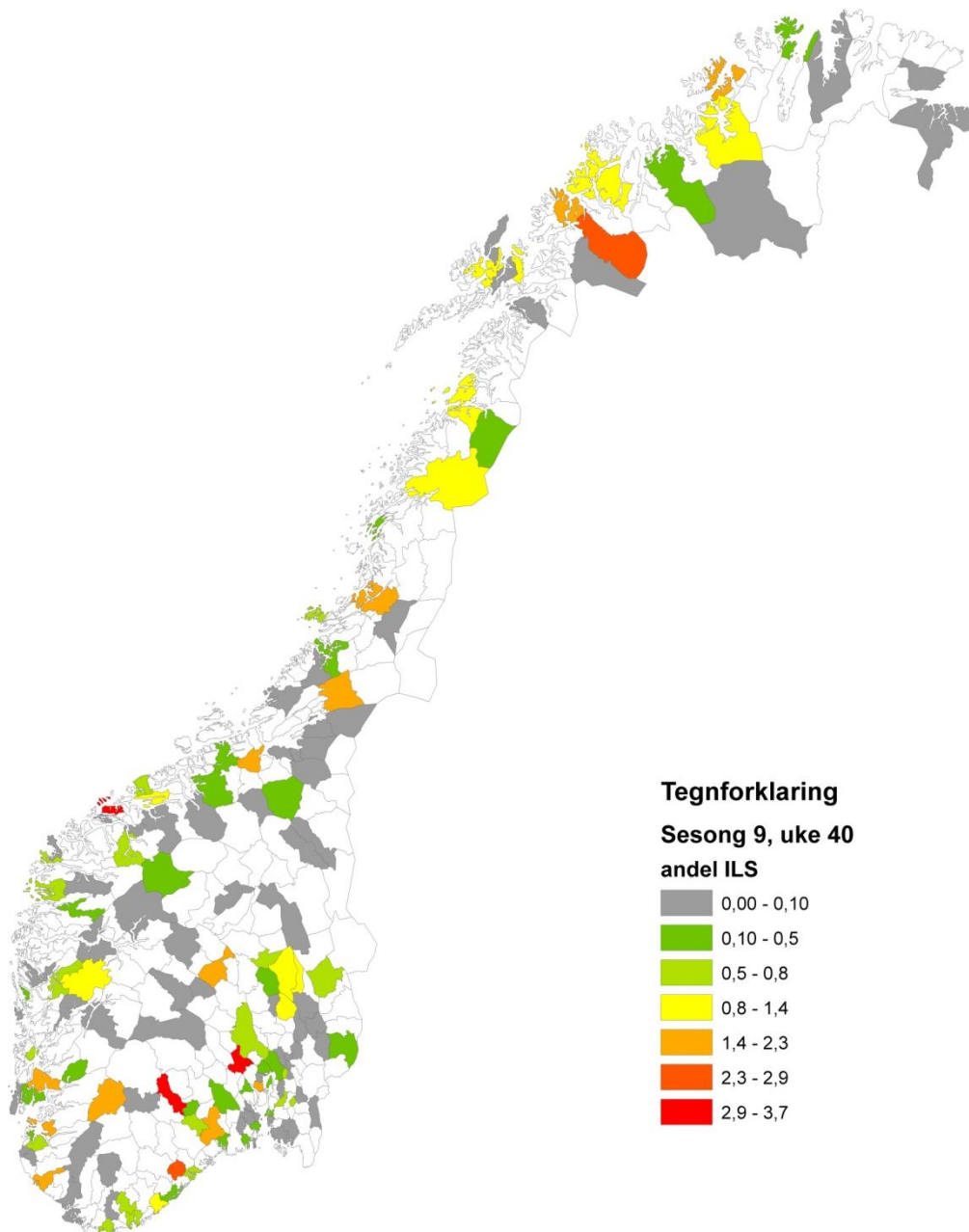
De høye verdiene for 1998 til 2000 (sesong 1 og 2) gjenspeiler godt resultatet fra Figur 5.3, hvor gjennomsnittet for de samme sesongene skiller seg markant ut med en meget høy maksimumsverdi for andel ILS. Perioden mellom 2003 og 2004 (sesong 6) skiller seg ut i figur 5.3 ved å ha relativt små verdier i gjennomsnittsverdiene, både for maksimum og minimum. Denne sesongen er imidlertid den første som registrerer minimumsverdier større enn 0.



**Figur 5.5: Diagrammet viser det gjennomsnittlige antallet registreringer pr kommune i hver sesong.**

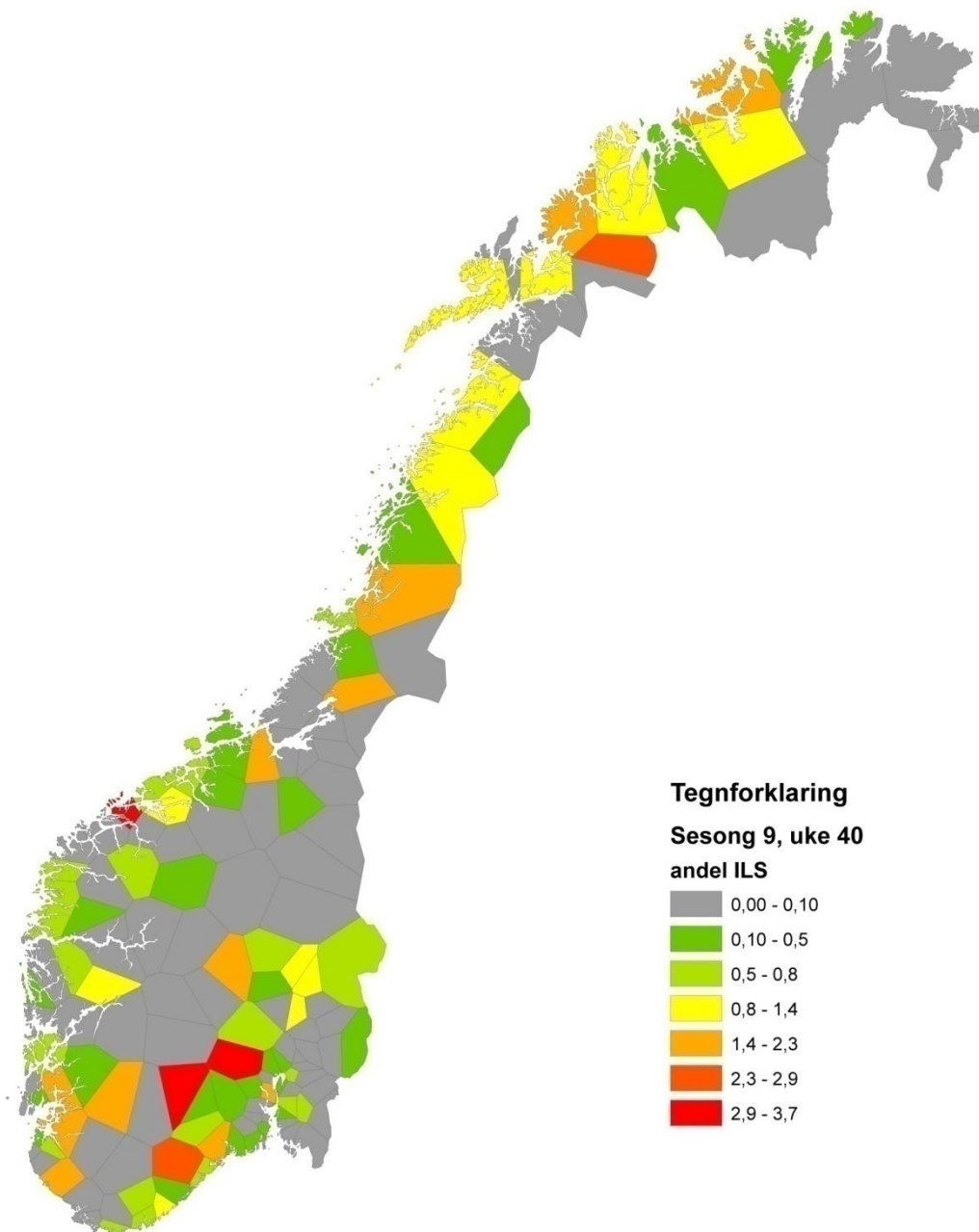
Figur 5.5 viser antallet registreringer i hver sesong. Man kan se at fra 2003-2004 (sesong 6) øker antallet registreringer, og det blir akkumulert en større informasjonsmengde for de siste sesongene av testperioden. Antallet kommuner som har vært med på registreringen har ligget stabilt rundt 157 alle sesongene dataene er registrert, med et unntak av sesong 1 og 2 hvor det eksisterer data for henholdsvis 150 og 148 kommuner. Det er til enhver tid ganske mange av legekantorene som ikke har sendt inn data.

Et kart ble produsert for å få en oversikt over hvordan fordelingen av andel ILS var i uken med det maksimale andel ILS i sesongen mellom 1999 og 2000 (sesong 2).



**Figur 5.6:** Figuren viser hvordan andel ILS fordeler seg i Norge fordelt på kommuneflatene ved starten av 2006 – 2007 sesongen. Datasettet som er brukt for kommuneflatene er N2000.

For å kunne studere spredningen over hele landet med de dataene som er innhentet, ble det laget thiessenpolygoner som representerer andel ILS. Det ble konstruert polygoner for hver uke gjennom et par sesonger for å se utviklingen og utbredelsen av influensaen. Et slikt kart er vist under, og det er produsert animasjoner som viser hvordan verdiene rundt i landet endrer seg gjennom influensasesongen.



**Figur 5.7:** Figuren viser hvordan andel ILS fordeler seg i Norge når det er beregnet thiessenpolygoner rundt registreringene ved starten av 2006 – 2007 sesongen. Datasettet som er brukt for kommuneflatene er N2000.

Ut fra beregningen av Morans I indeks kommer en frem til om verdiene er tilfeldige eller ikke. Det ble kjørt beregning på utvalgte uker gjennom alle sesongene (se vedlegg). Det viser seg at det er relativt stor forskjell mellom de ulike ukene som er testet i hver sesong. Enkelte sesonger er det tydelig at det finnes uker hvor det er sammenheng mellom de ulike verdiene. Denne sammenhengen ser man tydelig i ukene midt i sesongen, ved uke 50 og utover. I starten og slutten av sesongen finnes det ikke tegn på at det er sammenheng mellom verdiene. For å illustrere hvordan verdiene er presentert og ser ut, er det i Tabell 5.1 satt inn informasjon fra de utvalgte ukene i sesongen mellom 2006 og 2007 (sesong 9).

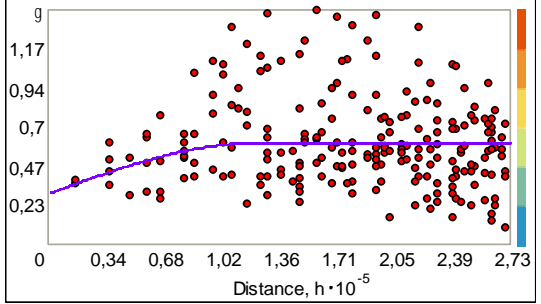
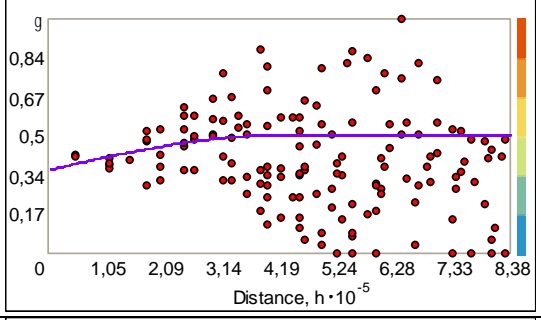
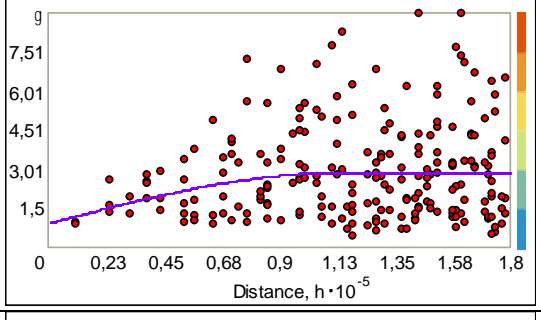
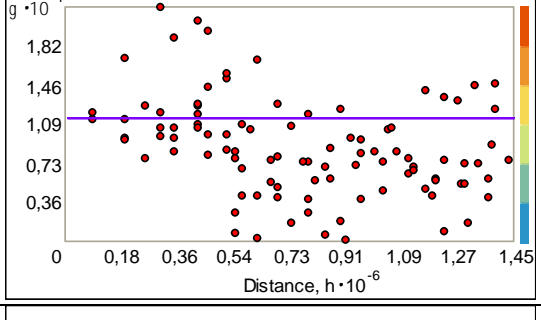
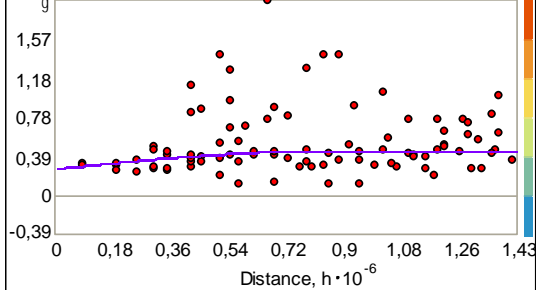
Uke	Z-Score	Morans I indeks
40	0,73	0
50	-0,92	-0,02
1	0,51	0
10	0,95	0,01
20	0,27	0

Tabell 5.1: Verdier for romlig autokorrelasjon for sesong 9 (2006-2007)

Som man kan se ut fra verdiene, ligger Morans I indeks i intervallet -0,02 til 0,01, og z-verdien er relativt lav både i positiv og negativ retning. Verdiene som fremkommer i Morans I indeks er relativt små og stabilt rundt 0, slik at det kan virke som verdiene for andel ILS er tilfeldig fordelt i området. Dette viser at i de ukene som er undersøkt er det ikke noen klare sammenhenger mellom andel ILS. Ser man på resultatet for de andre sesongene, ser man at spesielt sesongene mellom 2000 og 2002 (sesong 3 og 4) har store Z-verdier; noe som igjen kan tyde på at like andel ILS ligger tett samlet og at det kan være en sammenheng mellom dem. I begge sesongene er det uke 50 og uke 1 som skiller seg ut fra resten. I følge dokumentasjonen av metoden, vil det være rundt 95 % sjans for at dataene er autokorrelert i disse ukene. Dette kan ha en sammenheng med at det er på dette tidspunktet influensaepidemien begynner å ta form, men det er ikke noe konkret samsvar mellom Morans I indeks og prosent syke som vist i Figur 5.3. Når det gjelder 1999 til 2000 (sesong 2), som skiller seg markant ut fra gjennomsnittlig prosentantall syke, så ser vi at uke 1 har en høyere Z-verdi enn de andre ukene, men ikke nok til at det er en sikker sammenheng.

I de samme ukene som har blitt analysert med tanke på autokorrelasjon, har det blitt foretatt beregning av semivariogram og diverse geostatistikk. For å få semivariogrammene og krigingsfunksjonen mest mulig korrekt, ble det dannet trendoverflater over de ulike ukene som skulle testes. Trendoverflatene var overflater av henholdsvis første og andre grad. Ut fra disse trendoverflatene og eksperimentering i krigingsfunksjonen, ved å teste datasettene med trendfjerning av de to ulike gradene, viser det seg at verdiene som viser standardfeilen blir minst ved fjerning av førsteordenstrendoverflate. Dette tyder på at det sannsynligvis forekommer en førsteordenstrend i datasettet. Trendoverflaten endrer seg for hver uke, slik at det er ikke noen entydig retning på trenden for alle influensadataene; det er øst-vest og vest-øst som dominerer, med noe grad av rotering. Semivariogrammene og verdiene endrer seg kraftig mellom hver uke, noe som tilsier at det kan være store forskjeller på om dataene er

romlig tilfeldige. I Tabell 5.1 er det valgt ut data fra 2006 til 2007 (sesong 9), disse ukene er presentert som semivariogram sammen med geostatistiske verdier i Tabell 5.2 under.

<p><b>Uke 40 - Sesong 9</b></p>		<p>Range: 123 124            Nugget: 0,3086            Sill: 0,30579            Lagsize: 22 742            Model: Spherical</p> <p>RMS: 0,7984            ASE: 0,7107</p>
<p><b>Uke 50 - Sesong 9</b></p>		<p>Range: 417 542            Nugget: 0,36221            Sill: 0,15014            Lagsize: 69 829            Model: Spherical</p> <p>RMS: 0,7277            ASE: 0,6487</p>
<p><b>Uke 1 - Sesong 9</b></p>		<p>Range: 112 886            Nugget: 0,93963            Sill: 1,9521            Lagsize: 15 000            Model: Spherical</p> <p>RMS: 1,445            ASE: 1,44</p>
<p><b>Uke 10 - Sesong 9</b></p>		<p>Range: 1 372 990            Nugget: 11,589            Sill: 0            Lagsize: 120 950            Model: Spherical</p> <p>RMS: 3,575            ASE: 3,506</p>
<p><b>Uke 20 - Sesong 9</b></p>		<p>Range: 844 909            Nugget: 0,27068            Sill: 0,1805            Lagsize: 119 550</p> <p>RMS: 0,6596            AVS: 0,5608</p>

**Tabell 5.2: Semivariogram og geostatistiske verdier fra beregning av ordinær kriging. RMS står for *root mean squared* og beskriver virkningen av en varierende verdi., ASE er en forkortelse fra *Average Standard Error* som betegner den gjennomsnittlige standardfeilen.**



Ser vi på de enkelte ukene fra sesong 9, ser vi at et par uker skiller seg spesielt ut med tanke på semivariogrammene og de medfølgende verdiene. Uke 1 har en sill-verdi som er mye høyere enn nugget, mens uke 10 skiller seg ut i motsatt retning. Informasjonen om uke 1 viser klare tegn på at dataverdiene som er registrert er autokorrelerte og således at verdiene avhenger av hverandre. Det motsatte er tilfelle for uke 10, hvor funksjonen som vises i semivariogrammet er fullstendig flat innenfor avstanden som vises og vi har derfor en sill-verdi på 0. Dette tilsier at det ikke finnes noe form for autokorrelasjon og sammenheng mellom dataene på de ulike punktene. Når et datasett er autokorrelert er det ikke gitt at kriging er den mest optimale interpoleringsmetoden. For videre å kunne undersøke om modellen passer til datasettet, er det gitt en RMS-verdi. RMS er en forkortelse for *root mean squared* og denne verdien beskriver virkningen av en varierende verdi. ASE (*Average Standard Error*) blir også oppgitt i beregningen. Dette er en verdi som betegner den gjennomsnittlige standardfeilen i beregningen. Dette kan man blant annet se på RMS og ASE verdien som angir hvor stor gjennomsnittsfeil som finnes i det interpolerte datasettet. De samme ukene, 1 og 10, skiller seg spesielt ut med høye RMS og ASE verdier.

En sammenligning av verdiene beregnet fra de ulike visualiseringsmetodene ble utført ved å summere andel ILS på fylkesnivå. Denne sammenligningen viser at det er en generell stor forskjell mellom de ulike metodene som ble brukt. Den største forskjellen kommer mellom summering av verdiene fra de kontinuerlige verdiene fra krigingmetoden og de diskrete verdiene i de ulike punktene og kommunene. De oppsummerte verdiene på fylkesnivå fra kriging har en maksimalverdi på rundt 3 000, mens maksimalverdien av de andre metodene ligger på rundt 11. Årsaken til denne store forskjellen er at kriging lager en kontinuerlig overflate, slik at det vil være en verdi for andel ILS for hvert piksel innenfor fylket.

## 5.4 Metoder for visualisering av data

I tillegg til å ha informasjon om geografisk posisjon og blant annet andel ILS, har de tilgjengelige influensadataene temporal informasjon knyttet til seg. For hver influensaregistrering er det oppgitt i hvilken uke den ble foretatt og i hvilket år. Denne informasjonen kan brukes videre for å kunne presentere resultatene fra analysen og visuelt vurdere om det er noen spesiell trend på spredningen. Registreringen er gjort for samme

periode hvert år, slik at influensaen kan sies å være et syklisk fenomen; slik som definert av Midtbø et al.

For å få et korrekt inntrykk av endringene i andel ILS gjennom en sesong er det nødvendig å dele opp symboliseringen i like kategorier hver uke gjennom sesongen. Dette gjøres ved å se på den maksimale andel ILS hver sesong og fordele kategorier ut fra denne. Dersom alle sesongene skal vises innad i samme animasjon, vil det være nødvendig å klassifisere symbologien tilsvarende. Symboliseringsmulighetene som finnes i GIS-programvaren er store, slik at en kan bruke ulike symboliseringsformer for å se endringer. Det kan være endring av selve symbolet etter hvert som andel ILS øker, endring i farge eller endring av størrelse på symbolet. Det vil bli prøvd ut ulike symboliseringsmetoder, og de ulike symbolene vil bli diskutert.

Ved hjelp av tilleggsmodulen Tracking Analyst er det mulig å lage animasjoner av data med temporal informasjon. For å kunne ta i bruk datalagene i Tracking Analyst må man ha et felt som inneholder dato. Ettersom registreringene allerede inneholder ukenummer, vil det være naturlig å bruke denne for å finne en dato å representere registreringen på. I studien er det valgt å bruke den første dagen i den nevnte uka som dato for registreringene.

Ved bruk av thiessenpolygoner og kriging blir det dannet en overflate som også kan presenteres på forskjellige måter. Når en skal vise data på en overflate er man i likhet med punktdata nødt til å ha samme verdikategorier på de forskjellige datasettene.

## 5.5 Presentasjon av influensaspredning

For å presentere hvordan influensaen sprer seg gjennom en sesong ble det valgt å anvende ulike symboliseringsmetoder, og animasjoner ble laget på grunnlag av de ulike metodene. Animasjonene som ble laget er som følger

- Andel ILS symbolisert ved punkter for hver sesong
- Andel ILS symbolisert ved punkter for hele perioden

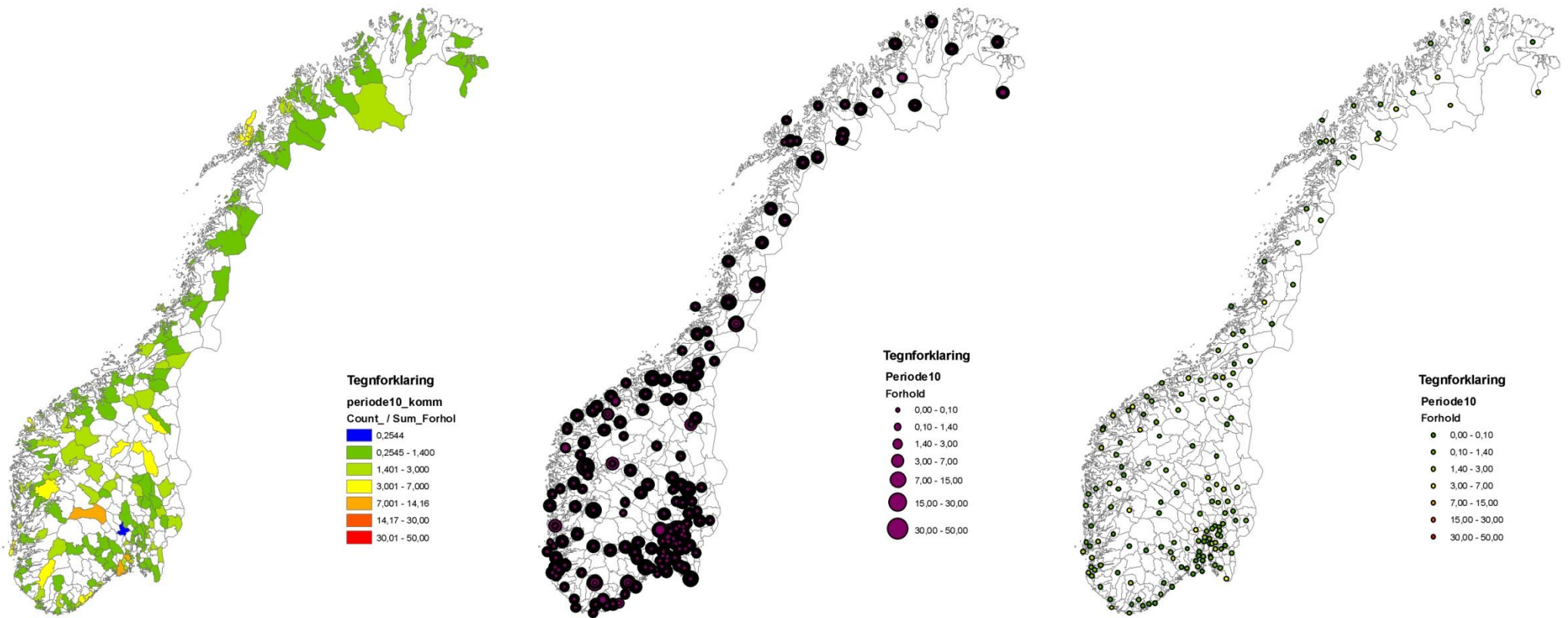
Det er i ArcGIS ikke noen enkel funksjon for å flytte posisjonen til teksten som viser tidsangivelsen. Derfor vil denne være statisk plassert øverst til venstre i bildene. Animering av thiessenpolygoner ble forsøkt gjennomført, men dette er ikke mulig på grunn av at polygoner

kun kan symboliseres med verdier som finnes i attributt Tabellen. Animasjonene er vedlagt på CD.

Symboliseringen av andel ILS kan endres for å presentere informasjonen på forskjellige måter, men det er nødvendig å klassifisere symbolene til faste verdier. Under arbeidet med klassifiseringen av de ulike sesongene ble det bestemt at datalagens verdier skulle deles opp i 7 kategorier, hvor man skilte ut andel ILS på 0 % og lot disse punktene få en egen symbologi.

Selve symboliseringen ble utført på to måter; den ene hvor endring i andel ILS ble symbolisert med endring i farge, mens den andre symboliserte andel ILS i punktets størrelse. Fargenyansene som andel ILS symboliseres på, ble valgt å være en fargeovergang mellom grønn og rød. Når verdiene for andel ILS er små og det er lite syke vil punktene være grønn, mens de blir sterkere rød jo større verdi andel ILS har. I det andre tilfellet hvor endring i andel ILS skal representeres ved hjelp av størrelse på punkt er det tatt utgangspunkt i kategorien med en rate på 0 %, denne ble satt til størrelse 4 og den høyeste kategorien fikk størrelsen 18.

Ved presentasjon av de dataene hvor andel ILS er representert som polygoner og flater er det kun mulig å symbolisere endring basert på farge. I likhet med symboliseringen av punktdata ble det brukt fargegradienter fra grønn til rød.

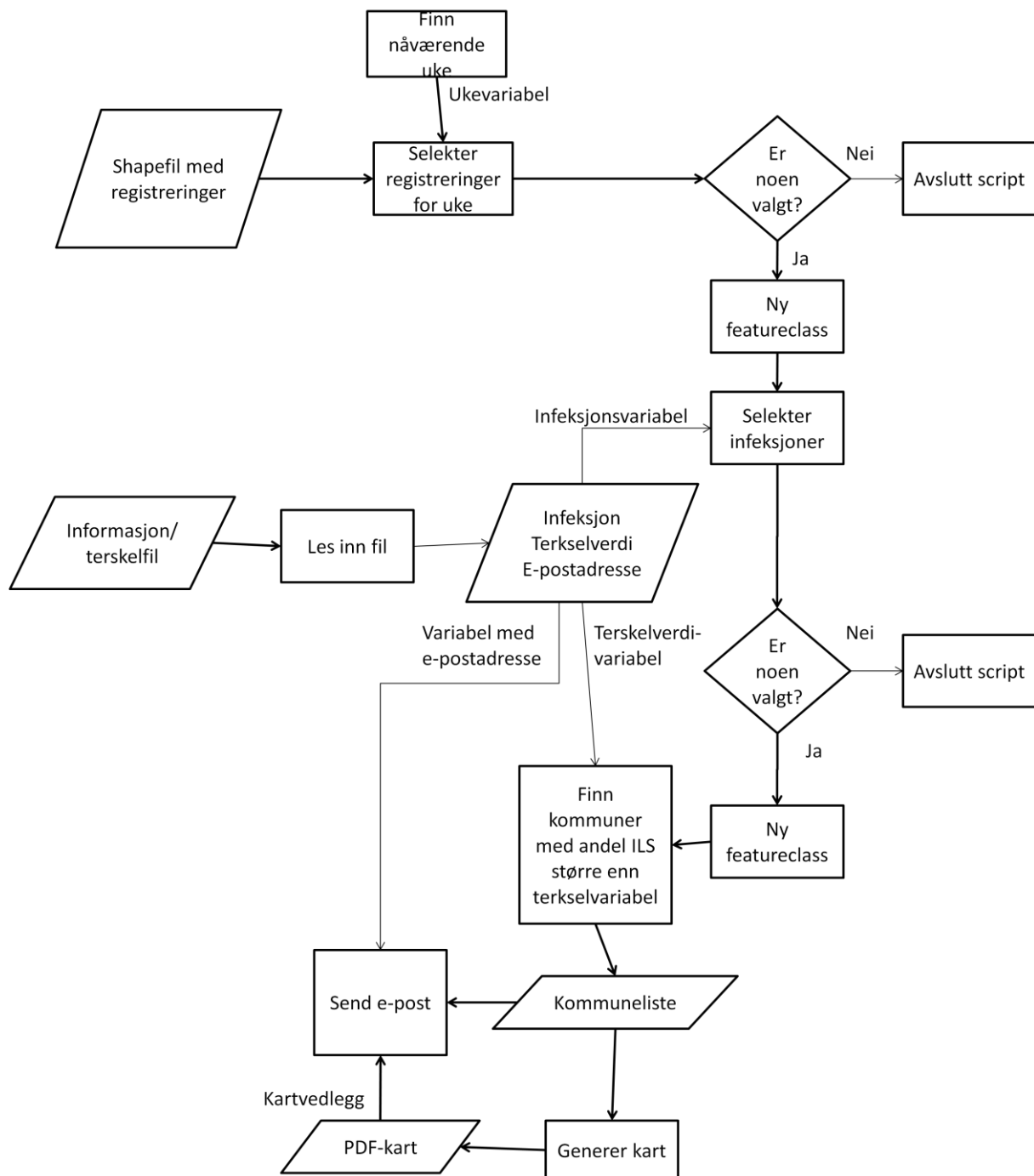


Figur 5.8: Kartene viser andel ILS i sesong 10 (2007-2008) på ulike presentasjonsmåter. Fra venstre: summert i kommuneflater, gradvis endret størrelse på punkt, og endring i farge på punkt.

## 6 Utvikling av signalsystem

### 6.1 Utvikling av signalsystem

Signalsystemet som ble utviklet i oppgaven er basert på å gjennomføre samme operasjon på flere datasett. Programvaren ArcGIS har utviklet et bibliotek av GIS-funksjonalitet til flere ulike programmeringsverktøy, for at det skal bli lettere å utvikle programmer som bruker GIS-data. Når en utvikler et system er det alltid hensiktsmessig å sette opp en oversikt over systemet som en UML-modell for å vise og holde oversikt over hvordan de ulike delene av systemet forholder seg til hverandre. Figur 6.1 viser et slikt diagram for hvordan signalsystemet er bygget opp.



**Figur 6.1:** Flytdiagrammet viser hvordan det syndrombaserte signalsystemet fungerer ved å bruke influensaregistreringene og en informasjonsfil som input, for så å gjøre noen tester og til slutt generere et kart og sende dette på e-post.

Det første scriptet gjør er å koble seg mot en Microsoft SQL Serverdatabase hvor det er lagret informasjon om dagens konsultasjoner og infeksjonsdiagnoser. Ettersom dette blir store datamengder gjøres en seleksjon av data som har blitt innhentet siste år. Denne informasjonen blir trukket ut og lagret som et featurelayer. Deretter blir det utført enda en seleksjon for å

motta de dataene som er registrert med ønsket ukenummer. Et nytt featurelayer blir dannet for videre bruk.

Hver infeksjon har en gitt terskelverdi som det skal testes for overskridelser, og hver infeksjon har en ansvarlig person som skal få tilbakemelding dersom denne terskelen blir overskredet. For å få tilgang til denne informasjonen, leser scriptet inn en tekstfil hvor informasjonen er lagret. Det vil stadig være ønskelig og nødvendig med oppdatering av informasjonen i denne; både lagring, sletting og endring av informasjon vil forekomme. For å få en enklere og mer oversiktlig måte å gjøre disse endringene på enn å gå inn i filen manuelt, ble det utviklet en egen applikasjon.

Applikasjonen ble implementert i utviklingsmiljøet Microsoft Visual Studio og er utviklet i språket *C#*. Denne applikasjonen er den eneste delen av systemet som har et brukergrensesnitt. Brukergrensesnittet gjør det betydelig enklere å legge til og endre informasjon om de ulike infeksjonene som signalsystemet skal overvåke.

Applikasjonsutviklingen ble gjennomført i henhold til en lagdelt arkitektur (Larman 2004). Det at et program er utviklet med lagdelt arkitektur vil si at brukergrensesnittet og selve applikasjon er fordelt på ulike lag i programmet. Brukergrensesnittet inneholder alle komponenter som er med på kommunikasjon med brukeren, mens applikasjonslaget består av kodene som utfører programmets hovedhensikt. En av fordelene med å utvikle på denne måten er at dersom man ønsker å endre brukergrensesnittet, gjør man disse endringene uten at man trenger å forstyrre applikasjonslaget; og omvendt.

Scriptet som er utviklet for å detektere utbrudd er foreslått kjørt som en planlagt prosess. Det at scriptet blir kjørt som en planlagt prosess vil si at datamaskinen ved et bestemt tidspunkt vil sette i gang kjøring av scriptet automatisk, slik at man slipper å sette scriptet i gang manuelt. Fordelen med dette er at man kan sette opp funksjonen slik at scriptet kjøres ved et tidspunkt man vet at det ikke kommer inn flere data fra legekantor, og store deler av systemressursene vil være ledig slik at programmet kjøres raskere.

## 6.2 Implementering av signalsystemet

Systemet som skal utvikles er basert på ArcGIS og tilgang til en database via et nettverk. Dersom tilkoblingen til nettverket ikke lykkes, vil det bli brukt en fil inneholdende data fra eventuelt forrige kjøring; eller manuell kopiering av data fra database via *Model Builder*.

Ved kjøring av signalsystemet vil hastigheten og ytelsen bli påvirket av ulike faktorer som blant annet maskinvaren, databasetilgangen og nettverkskapasiteten. Dersom databasen med daglige infeksjonsdata hadde vært implementert og mulig å kjøre scriptet mot, ville det være hensiktsmessig å gjøre tester på når det er optimalt at scriptet kjøres. Scriptet ville da kunne bli kjørt ved ulike tidspunkt, dag og natt, for å undersøke forskjell i hastighet på grunn av de variablene som er nevnt over. Ettersom dette scriptet bruker influensadata fra en lokalt lagret fil, vil nettverkskapasitet ikke ha noe innvirkning.

## 6.3 Signalsystemets oppbygging

Programvaren ArcGIS fra ESRI har mange gode muligheter for å utvikle egne moduler og tillegg til programmet, samt automatisering av funksjoner. Et av de mest brukte høynivåspråkene for utvikling av systemer basert på ArcGIS er scriptspråket *Python*. ESRI har implementert en rekke geoprosesseringsverktøyer i et bibliotek som enkelt kan inkorporeres i et ønsket script. For utvikling i *Python* er det flere mulige verktøy som er tilgjengelig, med ulik grad av funksjonalitet. Utviklingsmiljøet Eclipse ble valgt på grunn av de gode mulighetene for debugging av scriptet, samt god visualisering av filer og syntaks.

Databasen som skal brukes for å lagre infeksjonsdata hver dag er ikke implementert. På grunn av dette er signalsystemet blitt forenklet i bruk av datamateriale. Hvis signalsystemet skulle bruke databasen, er det ved oppstart nødvendig å sjekke om programmet har tilgang til nettverksbanen hvor databasen ligger, og om det er tilgang til geodatabasen eller katalogen som skal brukes for lagring av mellomdataene. I denne forenklete versjonen er det valgt å bruke influensadatasettet som ble gjort tilgjengelig for analyse. Denne ligger lagret lokalt på pc og er lagret som en shapefil.

```
Function lesFiler():
    try:
        - oppretter filobjekt og klargjør for lesing
        - oppretter filobjekt og klargjør for skriving
```



```
except:
    - gir tilbakemelding om feilen
for (hver linje i tekstfil):
    - leser inn linje
    - legger linje inn i liste
```

Etter å ha opprettet tilgang til katalogen hvor datagrunnlaget er lagret, blir det testet om filen med informasjon om infeksjonene finnes på korrekt sted. Dersom denne testen er vellykket vil innholdet i filen bli lest inn i programmet for å kunne sjekke terskelverdiene. Formatet på filen er slik at all informasjon om en infeksjon ligger lagret på en linje, og det er derfor nødvendig å dele opp hver linje i ulike segmenter. Dette gjøres senere i scriptet. Hvis denne filen ikke finnes på det angitte stedet, vil programmet avslutte med en feilmelding til brukeren hvor det opplyses om at det må finnes ei slik fil. I samme del av programkoden blir det opprettet en ny tekstfil som skal inneholde en rapport angående hvordan kjøringen har gått. Denne filen blir overskrevet for hver kjøring av programmet ettersom det for hver dag kan være ulike utbrudd av sykdommer og det er ikke så interessant å spare på tekstfilen som opplyser om dette.

Når programmet har opprettet rapportfilen og lest inn alle verdiene fra terskelfilen, starter selve analysene av datasettet. Det første som gjøres er at datasettet som er importert som utvalg i en tabell blir omgjort til et featurelayer. Dette featurelayeret blir deretter gjenstand for en spørring som finner de dataene som ble registrert med forrige ukes ukenummer, ettersom scriptet regnes kjørt ved midnatt på mandag. Denne spørringen er enkel å endre dersom databasen skulle bli implementert og det er ønskelig med deteksjon på døgn. Dersom seleksjonen viser at det ikke er registrert data siste uke, vil programmet avslutte uten at det gjøres noen analyser. Hvis det derimot er registrert infeksjoner siste uke, vil de valgte dataene bli kopiert til en ny featureclass for videre analyser. En for-løkke kjører gjennom elementene som finnes i listen med infeksjonsinformasjonen som ble opprettet tidligere i scriptet. Hvert element i listen blir delt opp i 4 deler; terskelverdi, e-postadresse til kontaktperson, katalog til ArcGIS-prosjektfil, katalog til eksportert kart og navn på den eksporterte filen.

En ny featurelayer blir dannet med elementene som omhandler den angitte infeksjonen og disse blir også kopiert til en ny featureclass. Det er ønskelig å se på infeksjonenes andel av ILS, og det vil derfor bli lagt til et nytt felt i featureclass-tabellen for denne verdien. Andel ILS beregnes, og disse sjekkes opp mot terskelverdien. Hvis noen registreringer med infeksjoner har større andel ILS enn terskelverdien skal scriptet sende en e-post med

informasjon om hvilke kommuner det har blitt registrert høy andel ILS i. E-posten med varsel om utbrudd inneholder en meldingstekst som forklarer hvilken infeksjon som er registrert og i hvilke kommuner registreringene har funnet sted. For å kunne få et bedre innblikk i hvor utbruddene er, kjøres det et program kodet i Microsoft Studio for å lage et oversiktskart over sykdommen. Dette oversiktskartet brukes som et vedlegg med e-posten. Programmet som genererer kart kan beskrives med disse to hovedfunksjonene, i tillegg til flere mindre funksjoner:

```
function lagKart(prosjektkatalog):
```

- Åpner ArcGIS prosjektet
- aktiverer alle lagene og kartlayout
- eksporterer kart
- lukker ArcGIS

```
function eksporterKart():
```

- definerer hvilket output-format (standard er pdf)
- definerer output oppløsning
- henter informasjon om skjermbilde, etc.
- starter eksportering

Programmet som lager kart blir kjørt gjennom en pythonkommando og får med seg parametere angående hvilken prosjektfil som brukes for den aktuelle infeksjonen og hva filnavnet på det eksporterte kartet skal være. Det første som skjer i programmet er at det blir opprettet et objekt som inneholder kartdokumentet, og denne blir åpnet med prosjektfilen som parameter. For at de data som finnes i prosjektet skal kunne vises er det nødvendig å kalle en funksjon som aktiverer lagene og den aktive layouten i prosjektet. Etter aktivering kalles en metode for å eksportere kartet til ønsket filformat. Eksporteringsmulighetene i ArcGIS er mange, og ettersom formålet i dette tilfellet er å lage en lett tilgang til kartet vil PDF formatet bli brukt som standard. I eksporteringsfunksjonen blir det også definert katalog og navn for det eksporterte kartet, samt kartets oppløsning og ulike eksporteringsregler. Standardverdiene som brukes er:

Variabel	Standardverdi
Filformat	PDF
Kart oppløsning	300 dpi
Clip graphics to extent	False

**Tabell 6.1: Standardverdier for automatisert eksportering av infeksjonskart**

Sending av e-post gjøres ved å kalle en metode hvor man sender med en liste inneholdende infeksjonsnavnet, kommunenavn, og e-postadresse. Denne funksjonen er beskrevet med følgende pseudokode:

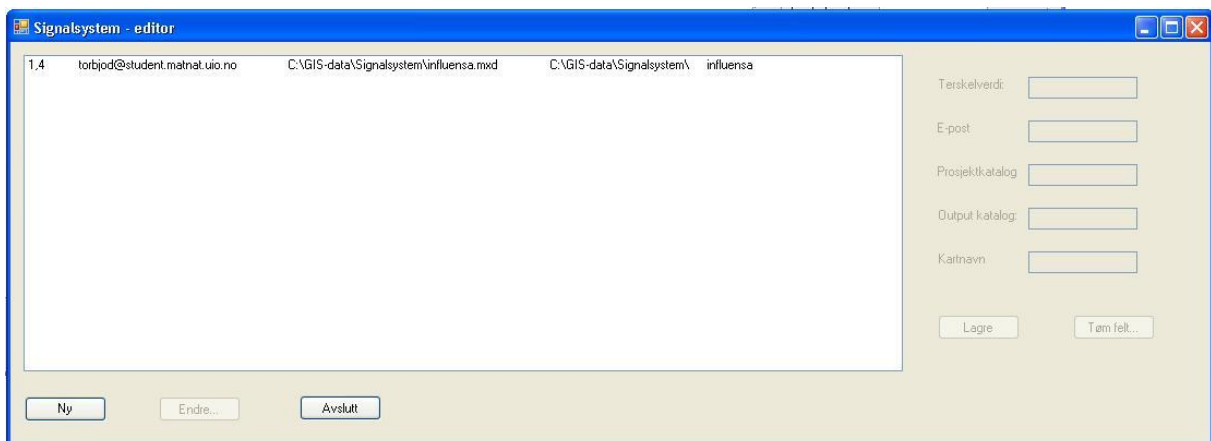
```
function sendMail (infeksjonsnavn, kommuner, epostadresse, infeksjonskart):  
    - oppretter melding fra MIMEMultipart  
    - registrerer meldingens mottaker  
    - registrerer meldingens avsender  
    - oppretterer MIMEText-objekt med tekstinnholdet i meldingen  
    - oppretter MIMEBase-objekt for behandling av vedlegg  
    - leser inn vedlegget og koder det som base64  
    - legger inn vedlegg i meldingsobjektet  
  
    - oppretter en variabel med avsenderens server  
    - oppretter variabel med avsenders brukernavn  
    - oppretter variabel med avsenders passord  
  
    try:  
        - oppretter et smtp-objekt med avsenders server og serverport  
        - angir grad av debugging (enten 1 eller 0)  
        - hvis det kreves autorisasjon:  
            - logger inn med brukernavn og passord  
            - sender e-posten  
            - lukker smtp-objektet  
  
    except:  
        - dersom noe går feil, gir en god tilbakemelding om feilen.
```

For å bruke funksjonen som sender e-post er det nødvendig å importere noen eksterne pythonmoduler for å få tilgang til de ulike funksjonene. Modulen `smtplib` inneholder funksjonene som omhandler serverfunksjonaliteten, mens ulike MIME-objekter fra e-post modulen beskriver selve e-postobjektet. Funksjonen mottar kommunenavn, e-postadresse og det eksporterte kartets katalog. Meldingen blir opprettet som et `MIMEMultipart`-objekt som inneholder både meldingstekst og meldingsvedlegg. Meldingsteksten blir opprettet og lagt inn i meldingsobjektet i tekstformat. Meldingsteksten inneholder blant annet navn på kommunene hvor det har blitt oppdaget forhøyet forekomst av smitte. Meldingsobjektet må også knyttes til informasjon om mottaker og avsenders e-postadresse. Delen av meldingen som inneholder

vedlegget blir opprettet som et MIMEBase objekt. Filen blir lest inn som en del av objektet, kodet i base64 format og lagt inn i meldingsobjektet. Det blir også dannet variabler som inneholder informasjon om blant annet avsenderens serveradresse, brukernavn og passord. Sikkerheten rundt avsenders brukernavn og passord kan diskuteres, og det kan vurderes om disse skal ligge i selve scriptet eller en annen ekstern fil. Koblingen med serveren gjøres ved å lage et smtpobjekt og tilordne serverinformasjonen til denne. Som oftest kreves det autorisasjon og innlogging for å sende mail, slik at brukernavn og passord må bli benyttet her. Selve sendinga av e-posten skjer med metodekallet `sendmail` i `smtplib`modulen.

Dersom det skulle skje noe feil i løpet av funksjonen for sending av e-post, blir disse fanget opp av Pythons innebygde exception-fanger, og det blir skrevet ut en feilmelding som forteller hva som har gått galt under sending. Dersom dette skulle inntreffe, vil scriptet stoppe prosesseringen av infeksjonen og gå videre til en eventuell neste infeksjon.

Filen som leses inn for å få informasjon om blant annet terskelverdier og e-postadresser, er en tekstfil som i dette tilfellet består av ei linje. Dersom man ønsker at scriptet skal undersøke flere forskjellige infeksjoner, må det gjøres noen endringer i scriptet samt legge inn ei linje for hver ny infeksjon i fila. Etter hvert er det sannsynligvis nødvendig å endre e-postadresser på kontaktpersoner, terskelverdi eller sti til prosjekt og det eksporterte kartet. For å gjøre dette enkelt er det utviklet et program i Microsoft Visual Studio, som gir disse endringene et grafisk brukergrensesnitt.



**Figur 6.2:** Brukergrensesnitt for editering av fil med informasjon om infeksjon.

Dette programmet er utviklet med en lagdelt arkitektur. Koden som åpner og lagrer endringene i terskelfilen er utviklet i en egen klasse, mens elementene som er vist i brukergrensesnittet er kodet i en annen klasse. På denne måten er det mulig å gjøre endringer i

underliggende kode eller i brukergrensesnittet uten å være nødt til å gjøre endringer i resten av programmet.

## 6.4 Vurdering av signalsystemets ytelse

Databasen som scriptet er tenkt kjørt sammen med er ikke implementert, noe som fører til at testingen av scriptet foregikk lokalt på en shapefil med influensaregistreringer mellom år 1998 og 2008. Det at scriptet kjøres lokalt og ikke opp mot en database kan påvirke tidsforbruket i positiv retning, dvs. det kan ta lengre tid å logge seg på og hente ferske data fra databasen, avhengig av brukermengde på nettverket og antall innloggede brukere i databasen. For å se om hastigheten endrer seg etter maskinvaren som brukes, er scriptet kjørt på datamaskiner med spesifikasjonene presentert i Tabell 6.2.

Navn	Prosesor	RAM	Harddisk	Operativsystem
<b>Toshiba Satellite A10</b>	2,50 Ghz	512Mb	30Gb	Windows XP, Service Pack 3
<b>Lenovo</b>	2.0 Ghz x 2	2 Gb	143Gb	Windows XP, Service Pack 3

Tabell 6.2: Informasjon om pc-er brukt under testing av signalsystemscriptet

Ved kjøring av profileringsfunksjonen i *Python* fremkommer det en liste med alle metodekall som scriptet foretar. Dette innbefatter både egenproduserte metoder og metoder innen Pythonkjernen, som blant annet utskrift til skjerm etc. For å minske informasjonsmengden er det valgt å kun fokusere på de metodene som er egenlagde og de som har en tydelig funksjon i scriptet. Eksempelet som er vist under er kjørt på Lenovo-maskinen, uten andre programmer enn scriptet oppe.

```
53046 function calls (52977 primitive calls) in 49.665 CPU seconds
```

```
Ordered by: standard name
```

```
ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
      1   0,089    0,089    0,089    0,089   :0(AddField)
      1   0,364    0,364    0,364    0,364   :0(CalculateField_management)
      2   1,054    0,527    1,054    0,527   :0(CopyFeatures)
      2   0,039    0,019    0,039    0,019   :0(GetCount_management)
      3   8,316    2,772    8,316    2,772   :0(MakeFeatureLayer)
      2   0,261    0,131    0,261    0,131   :0(SelectLayerByAttribute)
      1   3,119    3,119    3,119    3,119   :0(system)
      1   0,000    0,000    0,000    0,000
prosjekt_fhi.py:100(lesTerskelFil)
      1   0,058    0,058   49,656   49,656
prosjekt_fhi.py:128(prosessering)
      1   0,001    0,001   36,334   36,334  prosjekt_fhi.py:14(sendMail)
```

Den første linjen forteller hvor mange funksjonskall som blir utført ved kjøring av scriptet, 53 046, hvor av 52 977 funksjoner er ikke-rekursive, og kjøretiden av scriptet og profileringen er registrert på totalt 50 sekunder. Listen er sortert alfabetisk etter navnet på funksjonen. Dersom en ser på kolonnen `ncalls` som forteller hvor mange kall som gjøres på hver metode, ser man at det er fire funksjoner som blir gjentatt flere ganger. Dette er funksjonen som finner antall selekterte elementer i shapefilen, kopierer de selekterte elementene til en ny fil, metoden som selekterer elementer med en gitt verdi, samt metoden som lager featurelayer av de selekterte elementene. Pythonfunksjonen som starter genereringen av kartet tar 3 sekunder, noe som er det høyeste tidsforbruket dersom man ser bort fra sending av e-post og selve prosesseringsdelen. Metoden som sender e-post til gitte adresser bruker 36 sekunder, og er således den metoden som scriptet bruker mest tid på å gjennomføre. Årsaken til at det tar såpass lang tid å opprette mailobjektet og få sendt dette er at *Python* bruker mange interne kommandoer for å få koblet seg til mailserveren, kode om vedlegget til gyldig format og så sende mailen. Det store antallet funksjonskall som blir nevnt skyldes i stor grad denne funksjonen.

Sammenligner vi med kjøringen av scriptet på den andre pc'en som har mindre hardware-ressurser, kan en se at det er forskjell på kjøretid. Den totale kjøretiden til scriptet er 144 sekunder, mot 50 sekunder på Lenovo. Dette er en ganske merkbar hastighetsforkjell, men ettersom scriptet er tenkt kjørt som en windowsprosess har ikke dette så mye å si ettersom programmet kjører i bakgrunnen og man kan fortsette å jobbe som normalt.

## 7 Diskusjon

### 7.1 Anvendelse av varslingsystem/signalsystem

Signalsystemet som er utviklet i oppgaven gir tilbakemelding med et kart dersom det er registrert økt forekomst av andel ILS i en eller flere av kommunene. Hvis GIS-analysen viser økt forekomst, må resultatet fra analysen vurderes av en kvalifisert beslutningstaker. Dette fremkommer i Chakhar og Martel (2003) hvor det hevdes at et geografisk informasjonssystem ikke har evnen til å vurdere flere alternativer eller scenarioer. Dersom beslutningstaker kommer frem til at det er et utbrudd eller en epidemi og at det skal gjøres tiltak; blir det nødvendig å gi befolkningen informasjon om at sesonginfluensaen er på vei. I forhold til de forholdene som er påkrevd i følge Einstein og Sousa (2007) er det i dette signalsystemet og analysen tatt mest hensyn til krav 1 og 2 (kapittel 4.1.1, side 25).

I signalsystemet som er utviklet for sesonginfluensa er det brukt en terskelverdi på 1,4 % for å finne mulige utbrudd av influensa. Denne verdien for andel ILS brukes som definisjon for utbrudd av influensa. Det kan være ønskelig å få varsel om økt forekomst av smitte før det er et reelt utbrudd. Ved å sette terskelverdien til 1, vil systemet kunne fange opp utbruddene på et tidlig tidspunkt. Baselineverdien som ligger til grunn for beregningen av terskel/utbruddsverdien er beregnet ut fra de første ukene i influensasезongen og definisjonen brukes blant annet i influensaovervåkingen som gjennomføres i EISS. Effekten som fremkommer dersom man setter inn 1 % i stedet for 1,4 % i signalsystemet, blir at det blir gitt tidlige, men også flere signaler enn ved utbruddsnivået. Fordelen med dette er at man får en oversikt over de stedene hvor det er mulig at det vil bli et utbrudd, men det vil også bidra til å øke informasjonsmengden. Når signalsystemet skal lagre diagnoseinformasjon hver dag, vil datamengdene etter kort tid bli store. Det er derfor viktig å sette inn kriterier i signalsystemet som luker ut alle ikke-relevante signaler. Dersom man legger inn informasjon om infeksjonstype i fila som brukes av programmet, vil man kunne begrense søket til kun å gjelde den aktuelle infeksjonen, på samme måte som man avgrenser søket ved å velge data fra et gitt ukenummer.

Ut ifra hvilken infeksjon det blir gitt signal om økt forekomst av, settes det i gang ulike tiltak. Ved et signal om økt forekomst av diaré vil den ansvarlige ved Folkehelseinstituttet kontakte den personen som er ansvarlig for smittevern i den berørte kommunen; dette er ofte kommuneoverlegen. Det er viktig å undersøke om den økte forekomsten er reell eller om

signalet har blitt gitt ved en feil. Dersom det skulle vise seg at det er en reell økning i forekomst av diaré, vil det bli innledet videre etterforskning. Etterforskningen foretas ved å ta prøver av pasienten for å analysere og finne hvilken smittestoff som er gjeldende i det enkelte tilfellet. Hvis det blir gitt et signal for influensa, er det ikke noen større tiltak som gjøres for å begrense spredning. Det vil imidlertid bli informert om vaksinasjonsmuligheter for risikogrupper.

For å varsle publikum om et mulig utbrudd av en sykdom, kan det tas i bruk flere kommunikasjonsmetoder. Det ene er å informere via media, mens en annen mulighet er å legge ut informasjon på web. En slik webløsning kan tenkes å være både som artikler på hjemmesidene og som interaktive kart som viser hvor sykdommen har økt forekomst og hvor høy forekomsten er.

## 7.2 Utviklingsmetodikk og ytelse av signalsystem

Et script som kjøres som signalsystem for influensalignende symptomer ble utviklet i blant annet utviklingsmiljøene Eclipse og Microsoft Visual Studio i henholdsvis *Python* og *C# .NET*. Utviklingen av signalsystemet ble gjennomført med en smidig utviklingsmetodikk. Fordelen med å bruke en smidig utviklingsmetodikk er at utvikleren har mulighet for å endre programmet fortløpende dersom det skulle dukke opp nye ønskede funksjoner ved programmet. I en annen utviklingsmetodikk vil programmet ofte bli utviklet i sin helhet før utvikler eventuelt går inn og gjør endringer og legger til ny funksjonalitet. En annen fordel med denne metodikken er at man får testet programmet underveis og at man da kan jobbe med hver enkelt del av programmet separat. Brukergrensesnittet som ble utviklet for å enkelt kunne endre filen med informasjon om influensaens terskelverdi, e-postadresse etc. ble utviklet med en lagdelt arkitektur. Fordelen med denne type arkitektur i forbindelse med grafiske brukergrensesnitt er at man skiller mellom programmets grunnleggende funksjoner, kommunikasjon med brukeren og det grafiske grensesnittet. Dette gjøres for å lettere kunne endre og legge til ny funksjonalitet uten å måtte endre andre deler av scriptet eller gjøre endringene i samme fil.

En av de delene av programmet som kan kodes på flere måter og som berører sikkerhetshensyn er funksjonen som tar for seg sending av e-post. For å kunne sende e-post er brukeren nødt til å ha tilgang til og koble seg opp mot en SMTP-server, og i den forbindelse



trengs det brukernavn og passord. Det å oppgi brukernavn og passord til en tjeneste er alltid knyttet opp mot risiko. Dersom programmet og scriptet skulle bli brukt av andre, så er det nødvendig å tenke på sikkerhet rundt brukernavnet og passordet til brukeren. Det er flere løsninger på dette problemet; de to måtene som blir diskutert er å ha informasjon direkte i programmet eller i en utenforstående fil. Dersom utvikleren går for å legge inn brukerinformasjon i programkoden, oppstår det problem når brukeren skifter brukernavn eller passord. Ved et slikt tilfelle er utvikleren nødt til å gå inn i programkoden å endre informasjonen i selve programmet. Dette krever at programmereren er tilgjengelig eller at man har programmeringskompetanse for å gjøre endringer i koden. Fordelen med denne metoden er at det for utenforstående er vanskelig å finne brukerinformasjonen dersom programmet kjøres som en eksekverbar fil og programkoden oppbevares på en annen disk eller katalog enn den kjørbare filen. Den andre måten å behandle denne problematikken på er å la brukernavn og passord ligge i en egen fil. Brukeren av programmet kan nå enkelt gå inn og endre informasjonen når det måtte trenge uten å gå via programkode. Ulempen med dette er at dersom det ikke gjøres noen kryptering av informasjonen vil brukerinformasjonen ligge tilgjengelig for alle som har tilgang til området hvor programmet ligger. Dette kan løses ved å lage et grafisk brukergrensesnitt for endring av brukerinformasjon, mens dataene som blir skrevet inn i grensesnittet blir kryptert og lagret i filen. På denne måten har man kontroll på at brukerinformasjon ikke blir liggende åpenlyst.

Signalsystemet ble utviklet i *Python* på grunn av den enkle koblingen mot ArcGIS og dets biblioteker. Programmeringsspråket *Python* er et høynivåspråk, noe som gjør at programmerer behøver mindre kode enn ved lavnivåspråk. I utviklingen av GIS-programvare og utvidelser av disse, er det ofte brukt *Python* eller .NET-rammeverk. Fordelen med å kunne utvikle i GIS utvidelser i *Python* er at programmeringsmiljøet og kompilatoren til *Python* er gratis og dermed tilgjengelig for alle, i motsetning til .NET som gjerne blir utviklet i Microsoft Visual Studio. Etersom *Python* er plattformuavhengig, finnes det ikke støtte for å generere kjørbare filer i .exe-format; noe som fører til at brukeren som skal kjøre programmet er helt avhengig av å ha pythonkompilatoren og alle de benyttede bibliotekene installert på datamaskinen. Dette slipper man dersom man utvikler i .NET-rammeverket som er windowsbasert og er beregnet for å kunne generere kjørbare filer. I alle tilfeller er man nødt til å ha tilgang til ArcGIS biblioteket.

ArcObjects og .NET-rammeverket er tatt i bruk for utviklingen av programmet som gjør den automatiske genereringen av influensakartet. ArcObjects har et veldig omfattende klassebibliotek, noe som bidrar til å gjøre terskelen for å ta i bruk mulighetene det gir vesentlig høyere enn det er med biblioteket i *Python*. På den andre siden er kombinasjonen av disse veldig godt egnet for å utvikle tilleggsprogrammer med grafisk brukergrensesnitt, automatiserte rutiner og ulike analyser.

Når det gjelder ytelsen til scriptet som ble utviklet kan man se at det er store forskjeller mellom de to ulike maskinene som scriptet ble testet på. Dette kan både være på grunn av ulik maskinressurser, men også hvor mange programmer som var oppe og tilstanden til harddisk og minne. For å få en optimal test på ytelsen til programmet burde maskinene være helt rene, noe de ikke var. Ettersom det blir brukt mange metoder fra biblioteket til ArcGIS er det ikke mulig å gjøre endringer i disse, og på den måten få scriptet til å kjøre raskere.

Når det gjelder resultatet av scriptet, så blir det dannet en shapefil som inneholder influensaregistreringen koblet sammen med midtpunkt til kommunen det er registrert i. Det er ikke sikkert det er mest optimalt å presentere resultatet på denne måten. Dersom man som i dette tilfellet bruker punkter, er det lettere å gjøre geografiske analyser på dataene, men det kan være vanskelig å se hvordan andel ILS i de ulike kommunene utvikler seg. Dersom man skal bruke resultatet som kun en presentasjonsmåte, så kan det lønne seg å koble registreringene sammen med kommuneflatene.

### 7.3 Kvalitet og mulige forbedringer i data

Datasettet som ble benyttet i studien er basert på innsendt informasjon fra utvalgte legekantor, hvor det er registrert antall pasienter med influensalignende symptomer og antall konsultasjoner legen har hatt inneværende uke. Legekantorene som er brukt for innsamling av data ligger spredt rundt om i landet, men de fleste legekantorene er fra Øst- og Sørlandet. Det at det er samlet inn informasjon fra få legekantor i Nord-Norge vil bidra til å svekke muligheten for å få en optimal geografisk analyse av dataene. Når influensaregistreringene er lagret som punkter, vil den geografiske analysen bli påvirket av punktenes tetthet. Ved en geografisk analyse som for eksempel interpolasjon, vil de nye beregnede verdiene være mest nøyaktig når det finnes naboverdier. I dette tilfellet, antar vi at informasjonen i datasettet er omvendt proporsjonal, og det vil for de beregnede verdiene i Nord-Norge være større usikkerhet knyttet til verdien. Det finnes to måter å gjøre analysene mer optimale på. Den

første måten analysen kan gjennomføres på er å dele opp analysene i landsdeler, slik at Sør- og Nord-Norge blir interpolert for seg. Konsekvensen av dette er at man får en veldig nøyaktig interpolering av området med mange punkter, og man kan i tillegg øke antallet nabopunkter som skal brukes i de områdene med få punkt. Den andre metoden som kan brukes for å optimalisere analysene er at det blir startet innsamling av data fra alle landets legekantor.

En gang i uken gjennom influensasesongen blir det kjørt et uttrekk fra legekantorenes elektroniske journalsystem som teller opp antallet diagnoser med influensalignende symptomer. Informasjonen som blir trukket ut blir så overført til et spesielt kort og oversendt Folkehelseinstituttet; ved noen legekantor skrives tallene ut automatisk, mens de hos andre blir skrevet på kortet for hånd. I denne prosessen kan det oppstå skrivefeil som forplanter seg videre i systemet når kortene blir registrert ved Folkehelseinstituttet. Folkehelseinstituttet kan ikke gå inn i datasystemet til det enkelte legekantor og sikre at korrekte data er mottatt. Konsekvensene av dette vil være at databasen inneholder feilaktige verdier, og det vil igjen svekke validiteten til dataene for videre bruk. Dersom databasen ikke er tilstrekkelig optimalisert eller at man ikke har gode nok sjekkerutiner som fanger opp feil, vil det kunne oppstå problemer med dobbeltlagring av data. Databasesystemet bør derfor ha funksjonalitet for å luke ut slike feil. For å forbedre dette systemet og dermed få ned antallet feilregistreringer, er det flere aktuelle tiltak som kan benyttes. Disse tiltakene kan blant annet være å forbedre datakontrollen i databasen slik at feil oppdages ved manuell registrering i databasen, eller at det gjøres et automatisk uttrekk fra legekantorenes datasystemer. Hele denne prosessen fra uttrekk av data, håndskrivning av tall til registrering i databasen kan gjøres automatisk. Programvareleverandørene kan videreutvikle opptellingsfunksjonen, slik at de oppsummerte tallene blir sendt elektronisk til databasen hos Folkehelseinstituttet.

Under importeringen av Excel-dataene til Access-databasen som ble brukt til feilrettingene oppstod det et problem med dataformatet. Ved importering av kolonner i Excel med datoformat til Access, er det nødvendig at cellen er formatert på en bestemt måte. Det var ikke tilfellet her, slik at det en del datoer ble lagt inn i Access som tall. For å korrigere dette er det nødvendig å gå gjennom regnearket og finne de cellene med feil format. Dette eksempelet viser at ved konvertering mellom ulike format er dataene nødt til å være nøyaktig. Dette er et problem som oppstår spesielt ved geografisk analyse på den måten som er gjort i denne

studien. Ved en eventuell presentasjon av data på webløsning eller senere geografiske analyser, vil dette sannsynligvis ikke være noe stort problem.

For å kunne presentere influensadataene på mest mulig korrekt måte er det nødvendig med korrekt bakgrunnsinformasjon. Ettersom kommunene i Norge ikke er statiske, vil det være nødvendig å ta hensyn til endringer i disse dataene. Dette kan løses ved å ha en rutine som automatisk sjekker kommunenummer mot en tjeneste hos for eksempel SSB. Dersom det skal foretas en analyse av datasettet slik det ligger i dag, er det nødvendig å innhente informasjon om endringer i kommunenummer, legge disse inn i en tabell og kjøre en SQL-spørring for å bytte ut gamle kommunenummer med nye.

Kvaliteten på dataene som er registrert i databasen blir betydelig påvirket av om det er riktig antall influensatilfeller som er registrert. Betegnelsen influensalignende symptomer gjør at det kan være usikkerhet om en pasient virkelig har influensa eller kanskje bare en luftveisinfeksjon eller forkjølelse. Det er heller ikke alle personer med influensasymptomer som oppsøker lege og graden av dette kan muligens variere geografisk. Figur 1 illustrerer at av en stor mengde pasienter er det en mindre andel som det blir tatt prøver av, og at det er få av disse igjen som til slutt får en bekreftet influensadiagnose. Databasen som influensadataene i denne studien er hentet fra inneholder antall pasienter med influensalignende symptomer, så det vil være en stor usikkerhet rundt disse tallene. Kvaliteten på informasjonen vil også bli påvirket av hvordan pasientene forholder seg til sykdommen. Ved utvikling av et nytt databasesystem for innhenting av denne informasjonen, bør det være et skille mellom disse to kategoriene; influensalignende symptomer og diagnostisert influensa. Strengere kriterier for hvordan diagnosen influensalignende symptomer skal brukes, kan også bidra til at datakvaliteten blir bedre.

Registreringene i denne studien er summert opp og blir presentert ved midtpunktet til postnummerets flate. Dersom plassering av punktene er basert på pasientens bostedsadresse er den beste løsningen å maskere den eksakte adressen innenfor et 200 x 200 m rutenett (Leitner og Curtis 2006). Ut i fra informasjonen som er koblet til hver enkelt registrering i datamaterialet, er det ikke mulig å finne ut noe om de enkelte personene som det er registrert informasjon om. Personvernet er i så måte ivaretatt på en tilstrekkelig måte i studien. Innenfor en kommune er det ofte flere legekantor, og det er flere områder som sokner til kontorene. Dersom det fantes et datasett med de ulike legekantorenes koordinater, så er det mulig at det kunne sees en forskjell i andel ILS innad i kommunene med flere legekantor. Med tanke på

personvernet, så ville det bli ivaretatt på tilstrekkelig måte selv ved bruk av legekantorenes eksakte plassering i stedet for kommunens midtpunkt.

## 7.4 Metoder for geografisk analyse av data

En av årsakene til at det ikke er funnet noen tydelig autokorrelasjon i datasettet kan være at den romlige oppløsningen er for stor. Dette vil si at det er samlet inn data fra få legekantor og at disse ikke er spredt på en god måte. Som en kan se ut fra figuren, befinner de fleste legekantorene som det er registrert data fra i Sør-Norge, og det er få nordover i landet. En jevn fordeling av punkter er nødvendig for at kriging og andre interpolasjonsmetoder skal klare å beregne verdiene optimalt. Ved å ha en jevn fordeling av punkter vil hvert punkt ha like mange nabopunkter og en kan beregne mellomliggende verdier bedre. Dersom man leter etter clusterer i et datasett med flere forskjellige romlige oppløsninger, kan mulighetene for å finne clusterene minke fra 100 % ved eksakte posisjoner til 40 % ved den groveste oppløsningen (Ozonoff et al. 2007). I studien som er utført her, er data presentert ved eksakte posisjoner av midtpunktet i kommunen. Dersom man hadde hatt koordinat på det enkelte legekantor, kunne resultatet bli endret. En annen mulig årsak til at det ikke fremkommer noen tydelig autokorrelasjon og romlig sammenheng i datasettet kan skyldes den temporale oppløsningen. Dataene er registrert med en temporal oppløsning på 1 uke. For influensaovervåking kan dette være for høy oppløsning ettersom influensasjukdommen varer ca en uke med en inkubasjonstid på 2 dager. Når man har fått implementert databasen med døgnlige oppdateringer, vil man muligens få sett flere detaljer om hvordan sykdommen sprer seg. Influensa er en smittsom sykdom som smitter ved dråpesmitte, og det gir en ekstra utfordring når en skal kartlegge smitte i en så stor skala som i dette tilfellet.

Dagens situasjon med transportnettverket i landet er også av stor betydning for romlig sammenheng i influensasmitten. I dagens samfunn beveger mennesker seg mer mellom ulike steder i landet og på flere måter enn det de har gjort før. Dette medfører utfordringer i forhold til å studere sammenheng mellom ulike transportmetoder og smitte av influensa, slik det ble gjort av Skog et al. (2008) hvor man fokuserte på influensasmitte langs jernbanen i Sverige. I den studien ble det fastslått at det ikke fantes tilstrekkelig informasjon i datasettet for med sikkerhet å kunne si at de første influensatilfellene oppstod i bebyggelse ved jernbanestasjoner. Datasettet var innsamlet fra 398 leger og inkluderte ca 32 600 pasienter. Disse dataene er relativt godt dekkende for Sverige, men også her forekommer det flest registreringer i sør og i befolkningstette områder (Skog et al. 2008). Dataene er samlet inn for

sesongen 1889-1890, og det er imponerende at det var mulig å samle inn så mye informasjon på den tiden og at det er mulig å benytte dette materialet i dag. Grunnlaget for datainnsamling av influensalignende symptomer er tydelig bedre i Norge enn i Sverige. Den siste sesongen som er registrert dekker totalt 15973 personer med influensalignende symptomer, i forhold til Sverige som årlig registrerer gjennomsnittlig 500 – 1000 personer med influensa. I likhet med denne studien brukte Skog et al. (2008) thiessenpolygoner for å visualisere hvordan spredningen utviklet seg gjennom influensasesongen. Det er imidlertid forskjell på hvilken informasjon som er brukt i disse to studiene; Skog et al. brukte et forholdstall mellom antall syke og populasjonen i området, mens i denne studien ble det brukt antall syke mot antall konsultasjoner.

Det ble i denne studien valgt å se på datasettet ved hjelp av blant annet thiessenpolygoner og kriging. Fordelen med å bruke en kontinuerlig interpolert flate, her beregnet ved kriging, i forhold til en diskret interpolert flate er at det er beregnet verdier over alt i området. På denne måten får man et inntrykk av at andel ILS endrer seg gradvis ut fra et gitt punkt i studieområdet, noe man ikke gjør ved diskrete flater. De diskrete flatene fra thiessenpolygonene gir en brå overgang mellom hver enkelt region. Det er ikke sannsynlig at andel ILS er lik over alt innen området hvert polygon representerer. Problemet med grenseverdiene i slike tilfeller hører til et emne som kalles *fuzzy logikk* og er ikke behandlet i denne studien.

Når en har data fra ulike land og skal sammenligne dataene er det ikke mulig å studere dataene direkte. Årsaken til dette er at det er ulikt befolkningsgrunnlag, og ulike metoder som brukes for å beskrive influensaepidemiene. Metodene som brukes er blant annet konsultasjonsrater, som det er brukt i denne studien, antall konsultasjoner pr 100 og 100 000 personer eller eksakt befolkningstall. Dette medfører at en ofte er nødt til å beregne et normalisert forholdstall for å kunne sammenligne data fra de ulike landene (Uphoff et al. 2004, Arkema et al. 2009). Når det gjelder spredningsmodellen som skulle dannes ut fra influensadataene som var gjort tilgjengelig, ble det i resultater konkludert med at det ikke var noen konkret sammenheng mellom andel ILS og at det ikke kunne sees noen tydelig sammenheng mellom spredningen i de ulike sesongene. Det har blitt funnet at retningen på økningene i influensaaktiviteten gjennom forskjellige sesonger er uberegnelig. Vinteren 2005 – 2006 skilte seg markant ut fra de foregående sesongene i forhold til spredningsretningen til influensa; ved at sesongens maksimale verdier oppstod på ulike steder i Europa og ikke i vest-

øst som tidligere sesonger viser. Sesongen 2006 – 2007 hadde en sør-nord retning i perioden med maksimal aktivitet, men bevegelsen i nordlig retning ble markant først sent i sesongen.(Meijer et al. 2007a, Arkema et al. 2009). I disse to studiene ble det gjennomført en geografisk analyse over hvilke uker som hadde den største verdien av andel ILS i de ulike landene. Analysen som ble foretatt var ordinær kriging. Dette ble forsøkt i denne studien også, men det er stor feilmargin på krigingen, så dette ble forkastet.

Bruken av kriging i denne oppgaven bestod i å beregne en kontinuerlig overflate for å se hvordan sykdommen er fordelt rundt i landet og om det er noen sammenheng i datasettet. I de ukene som ordinær kriging ble testet på var det en relativt høy RMS-verdi, noe som tyder på at modellen som ble brukt ikke var tilstrekkelig for å få optimale resultater. De semivariogrammene som viser at sill er større enn nugget tyder på at det ikke eksisterer noen romlig sammenheng i datasettet for den gitte uken. Dette er i samsvar med verdiene fra beregningen av Morans I indeks. En av faktorene som kan spille inn på hvordan krigingsfunksjonen blir beregnet og hvordan semivariogrammet blir seende, ut er den geografiske spredningen til dataregistreringene.

## 7.5 Presentasjonsmåter for influensadata

Utviklingen av en influensaepidemi er presentert på to forskjellige måter; symbolisering av punkter og flater. Det er presentert forskningsresultater som viser at det er mulig å undersøke sykdommer med blant annet en tredimensjonal symbolisering. Ved hjelp av tredimensjonal visualisering, kan man presentere flere variable i den samme symbologien, men dette kan gi et for detaljert kart slik at det blir vanskeligere å se helheten i informasjonen (Krisp og Fronzek 2003, Curtis 2008). I denne studien ble det valgt å ikke presentere andel ILS som 3D-representasjoner. Årsaken til dette er at andel ILS er forholdsvis lav og det er derfor nødvendig å forstørre representasjonene kraftig for å kunne se noe forskjell i dataene. I tillegg er det kun andel ILS som er ønskelig å presentere i studien; variablene for antall syke og antall konsultasjoner bør ikke presenteres sammen på grunn av at befolkningsgrunnet og derav antall konsultasjoner er forskjellig for hver kommune.

Ved bruk av polygonsymbolisering av andel ILS ble flatene som representerer postnummersonene brukt. For å skille mellom de ulike gradene av andel ILS ble det brukt en gradient fra grønn til rød. Dette gir en god oversikt over hvordan sykdomsbildet er i de ulike

kommunene som er representert. Ved å bruke en stor målestokk for å kunne se hele landet under ett, oppstår det problemer med at det er mange små kommuner og disse blir ikke tydelige. Jo større areal kommunen har, jo bedre vil fargegradienten bli.

Punktsymboliseringen ble gjennomført med både fargegradient og størrelsesgradient. De ulike symboliseringene gjør det enklere å oppdage ulike elementer ved datasettet og utviklingen over tid. Ved størrelsesgradienten ser man tydelig hvordan andel ILS øker utover i sesongen, og videre synker etter den maksimale uken. Fargegraderingen gjør det imidlertid lettere å se forskjell på de ulike ukene og hvor stor andel ILS som forekom i kommunen på et gitt tidspunkt.

Fenomener med ulik tidsperiode bør animeres som en lineær- eller en syklisk animasjon ut fra tidsaspektet til fenomenet (Midtbø et al. 2007). Influensasesongene går gjennom flere uker og gjentar seg ikke på samme måte. Det er derfor hensiktsmessig å bruke et lineært system for å animere informasjonen som fremkommer for hver enkelt sesong. Dersom man ønsker å studere hvordan influensasesongene er i forhold til hverandre i samme animasjon, vil det være på sin plass å bruke en syklisk animasjon; dette på grunn av at datasettet dekker de samme ukene hvert år. Ettersom det kan være ønskelig å se på animasjonene på flere pc-er som ikke har tilgang på ArcGIS ble animasjonene eksportert til filformatet avi; noe som gjør at videoene kan kjøres på de fleste operativsystemer. Det er imidlertid begrenset oppløsning på animasjonene som blir dannet ved hjelp av animeringsfunksjonen i ArcGIS. Det er i all hovedsak størrelsen på kartlayouten som påvirker oppløsningen på animasjonen.

## 7.6 Videreutvikling av system og fremtidsutsikter

Når det gjelder presentasjon av influensadataene, er det mulig å presentere disse dataene på web. På grunn av ønsker fra brukere av websidene til SMI i Sverige ble det utviklet en tilsvarende tjeneste som viser ukentlig registrert informasjon for utvalgte infeksjonssykdommer. I løpet av perioden mellom lansering i mai 2003 og februar 2004, ble det registrert 1 000 000 besøkende på siden, hvorav 12 600 med unik IP-adresse (Rolfhamre et al. 2004). Dersom brukere og allmennheten i Norge ønsker det, vil det være mulig å utvikle et slikt system hvor den som vil kan gå inn og hente ut informasjon fra en nettside og se kart over stedene med registrerte infeksjoner; blant annet influensa. Før man eventuelt går i gang med å lage et slikt system er det nødvendig å se på brukergrunnlaget for tjenesten. En enkel



måte å sette opp en slik webtjeneste på er å bruke ESRI's ArcGIS Server-teknologi. Ved å anvende dette serversystemet kan man bruke det samme datasettet som er benyttet i oppgaven. For å få tilgang til de mest oppdaterte influensadataene når et nytt system blir implementert, er det fornuftig å bruke et åpent format som for eksempel XML. Dette vil gjøre kartpubliseringen lettere, og det vil være enklere for utviklere å lage systemer som bruker dataene.

Dersom det blir aktuelt med innsamling av data hver dag og systemet blir implementert, vil det være interessant å utføre analyser for å se om det kan være en forskjell på spredningstilfeldighet i ulike deler av landet. Det kan for eksempel forekomme forskjeller på spredning i kystområder og innlandsområder. Det er også mulig å legge inn andre romlige variabler for å se om det kan være noen sammenheng med influensaspredningen. Legger man inn gjennomsnittstemperatur for de ulike deler av landet i foregående uke, kan man se om dette også har betydning for spredning av influensa.

## 8 Konklusjon

Denne oppgaven presenterer metoder for geografisk analyse av influensadata og bruk av dataene innenfor et overvåkningssystem basert på geografisk informasjon. Dataene og overvåkningssystemet skal brukes på et makronivå, hele Norge, og det vil muliggjøre identifisering av økt forekomst av smitte. Resultatene fra denne studien vil kunne brukes både for å videreutvikle overvåkningssystemet, og for å gjøre videre analyser på data

Geografiske analyser av influensadata innhentet fra Folkehelseinstituttet ble gjennomført ved ulike metoder. Målet med de geografiske analysene var å vurdere om det er mulig å se noen romlig korrelasjon mellom influensaregistreringene og om det var mulig å bruke dataene videre i et syndrombasert signalsystem. Metodene som ble brukt for den geografiske analysen er danning av thiessenpolygoner, interpolasjon ved kriging og beregning av romlig autokorrelasjon. Influensadataene ble presentert på ulike måter for å avgjøre hvilken presentasjonsmåte som egner seg best til å vise forekomstene av økt andel ILS. Resultatene fra dataanalysen viser at det ikke er noe sikker romlig korrelasjon mellom registreringene. Det viste seg imidlertid at det er grunnlag for å bruke dataene videre som grunnlagsdata for signalsystemet. Presentasjonen av influensadataene viste at det for andel ILS er punkter med fargegradering som er mest hensiktsmessig.

Kravene som det syndrombaserte signalsystemet skulle oppfylle ble fastsatt, og utviklingsmetodikken ble diskutert. Signalsystemet ble utviklet i scriptspråket *Python* og et program ble utviklet i *C#* for automatisk å generere kart ved et gitt signal. Signalsystemet ble implementert på to ulike datamaskiner og testet på influensadataene. Kjøring av programmet viser at det er mulig å lage et signalsystem som sender e-post med et automatisk generert kart vedlagt e-posten dersom det skulle bli registrert andel ILS over en gitt verdi. Terskelverdien som ble brukt for å detektere signal ble satt til 1,4 %, noe som indikerer utbrudd. Resultatene av kjøring viser at de stedene hvor det er registrert en høyere andel ILS kommer klart frem. Ytelsen til programmet ble deretter testet på de to maskinene for å se hvordan programmet yter under varierende forhold. Resultatet av ytelsestesten tilsier at maskinressurser påvirker signalsystemet, og viser at den raskeste maskinen hadde en kjøretid som var ca 1/3 av kjøretiden til den andre maskinen.

## 9 Kilder

- Arkema, J.M., Meijer, A., Meerhoff, T.J., Velden, J.V.D., Paget, W.J. og (EISS), E.I.S.S. 2009. Epidemiological and Virological Assessment of Influenza Activity in Europe, During the 2006-2007 Winter. *Euro Surveillace* 13 (7-9).
- Basara, H. og Yuan, M. 2008. Community Health Assessment Using Self-Organizing Maps and Geographic Information Systems. *International Journal of Health Geographics* 7 (1), 67.
- Bell, W. og Dallas, C. 2007. Vulnerability of Populations and the Urban Health Care Systems to Nuclear Weapon Attack - Examples from Four American Cities. *International Journal of Health Geographics*.
- Burrough, P.A., McDonnell, R. og Burrough, P.A. 1998. *Principles of Geographical Information Systems*, Oxford: Oxford University Press. XIII, 333 s., 4 pl. s.
- Chakhar, S. og Martel, J.-M. 2003. Enhancing Geographical Information Systems Capabilities with Multi-Criteria Evaluation Functions. *Journal of Geographic Information and Decision Analysis* 7 (2), 47 - 71.
- Curtis, A. 2008. *Three-Dimensional Visualization of Cultural Clusters in the 1878 Yellow Fever Epidemic of New Orleans*.
- Einstein, H.H. og Sousa, R. 2007. *Warning Systems for Natural Threats*: Taylor & Francis.
- EISS. 2009. *Weekly Electronic Bulletin*. European Influenza Surveillance Scheme 2006 [nedlastet 02.03 2009]. Tilgjengelig fra [http://www.eiss.org/cgi-files/bulletin\\_v2.cgi?season=2008](http://www.eiss.org/cgi-files/bulletin_v2.cgi?season=2008).
- ESRI. 2009. *Create Thiessen Polygons (Analysis)* 2009 [nedlastet 22.01 2009]. Tilgjengelig fra [http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?id=1274&pid=1272&topicname=Create Thiessen Polygons \(Analysis\)](http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?id=1274&pid=1272&topicname=Create_Thiessen_Polygons_(Analysis)).
- Foldspang, A., Juul, S., Olsen, J. og Sabroe, S. 1986. *Epidemiologi: Sygdom Og Befolkning*, København: Munksgaard. 312 s. s.
- Folkehelseinstituttet. 2008. *Influenza*. Folkehelse Instituttet, 29.01.2008 2005 [nedlastet 13.05.2008 2008]. Tilgjengelig fra [http://www.fhi.no/eway/default.aspx?pid=233&trg=MainLeft\\_5799&MainArea\\_5661=5799:0:15,1904:1:0:0:::0:0&MainLeft\\_5799=5544:55787::1:5800:59:::0:0](http://www.fhi.no/eway/default.aspx?pid=233&trg=MainLeft_5799&MainArea_5661=5799:0:15,1904:1:0:0:::0:0&MainLeft_5799=5544:55787::1:5800:59:::0:0).
- Gore, A. 1998. *The Digital Earth: Understanding Our Planet in the 21st Century*, 31. January 1998.
- Helsenett, N. 2009. *Norsk Helsenett*, 08.09.2008 2009 [nedlastet 28.04.2009 2009]. Tilgjengelig fra <http://www.norsk-helsenett.no>.
- Krisp, J.M. og Fronzek, S. 2003. Visualising Thematical Spatial Data by Using the Third Dimension. Paper read at ScanGIS, at Espoo, Finland.

- Kraak, M.-J. og Ormeling, F.J. 1996. *Cartography: Visualization of Spatial Data*, Harlow: Longman. VIII, 222 s., [4] pl. s.
- Larman, C. 2004. *Applying Uml and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, Upper Saddle River, N.J.: Prentice Hall PTR. XXV, 703 s. s.
- Lee, J. og Wong, D.W.-S. 2001. *Statistical Analysis with Arcview Gis*, New York: John Wiley. XI, 192 s. s.
- Leitner, M. og Curtis, A. 2006. *A First Step Towards a Framework for Presenting the Location of Confidential Point Data on Maps-Results of an Empirical Perceptual Study*: Taylor & Francis.
- Liebhold, A.M. og Sharov, A.A. 1998. *Testing for Correlation in the Presence of Spatial Autocorrelation in Insect Count Data*. Redigert av Baumgartner, J., Brandmayr, P. og Manly, B.F.J. Population and Community Ecology for Insect Management and Conservation, Balkema, Rotterdam. 111-117 s.
- Mamelund, S.-E. 2004. *Spanish Influenza and Beyond: The Case of Norway*, Department of Economics, University of Oslo, [Oslo]. 1 b. (flere pag.) s.
- Martin, V., von Dobschuetz, S., Lemenach, A., Rass, N., Schoustra, W. og DeSimone, L. 2007. *Early Warning, Database, and Information Systems for Avian Influenza Surveillance*.
- Meijer, A., Meerhoff, T., Meuwissen, L., Velden, J.v.d., Paget, W. og (EISS), E.I.S.S. 2007a. Epidemiological and Virological Assessment of Influenza Activity in Europe During the Winter 2005-2006. *Euro Surveillance* 12 (9).
- Meijer, A., Meerhoff, T., Meuwissen, L., Velden, J.v.d., Paget, W. og Scheme, E.I.S. 2007b. Epidemiological and Virological Assessment of Influenza Activity in Europe During the Winter 2005-2006. *Euro Surveillance* 12 (9).
- Midtbø, T., Clarke, K.C. og Fabrikant, S.I. 2007. Human Interaction with Animated Maps: The Portrayal of the Passage of Time. *ScanGIS*, 5.-7. September 2007, Ås, Norway, 45-60.
- Nygaard, K., Werner-Johansen, O., Ronsen, S., Caugant, D.A., Simonsen, O., Kanestrom, A., Ask, E., Ringstad, J., Odegard, R., Jensen, T., Krogh, T., Hoiby, E.A., Ragnhildstveit, E., Aaberge, I.S. og Aavitsland, P. 2008. *An Outbreak of Legionnaires Disease Caused by Long-Distance Spread from an Industrial Air Scrubber in Sarpsborg, Norway*.
- Ozonoff, A., Jeffery, C., Manjourides, J., White, L. og Pagano, M. 2007. Effect of Spatial Resolution on Cluster Detection: A Simulation Study. *International Journal of Health Geographics* 6, 52.
- Pande, S., Keyzer, M.A., Arouna, A. og Sonneveld, B.G.J.S. 2008. Addressing Diarrhea Prevalence in the West African Middle Belt: Social and Geographic Dimensions in a Case Study for Benin. *International Journal of Health Geographics* 7 (17).
- Profdoc. 2008. *Profdoc Norge As 2008* [nedlastet 10.10 2008]. Tilgjengelig fra <http://www.profdoc.no>.

- Rolfhamre, P., Grabowska, K. og Ekdahl, K. 2004. Implementing a Public Web Based Gis Service for Feedback of Surveillance Data on Communicable Diseases in Sweden. *BMC Infectious Diseases* 4 (1), 17.
- Rothman, K.J. 2002. *Epidemiology: An Introduction*, New York: Oxford University Press.
- Schuurman, N., Fiedler, R.S., Grzybowski, S.C. og Gund, D. 2006. Defining Rational Hospital Catchments for Non-Urban Areas Based on Travel-Time. *International Journal of Health Geographics*.
- Skog, L., Hauska, H. og Linde, A. 2008. The Russian Influenza in Sweden in 1889-90: An Example of Geographic Information Systems Analysis. *Euro Surveillance*.
- Systems, H.M. 2008. *System X | Hove Medical Systems As 2008* [nedlastet 08.08 2008]. Tilgjengelig fra [http://www.systemx.no/?page\\_id=18](http://www.systemx.no/?page_id=18).
- Timpka, T., Morin, M., Jenvald, J., Eriksson, H. og Gursky, E.A. 2005. Towards a Simulation Environment for Modeling of Local Influenza Outbreaks. Paper read at AMIA 2005 Symposium.
- Uphoff, H., Stalleicken, I., Bartelds, A., Phiesel, B. og Kistemann, B.T. 2004. Are Influenza Surveillance Data Useful for Mapping Presentations? *Virus Research* 103 (1-2), 35-46.
- Webster, R. og Oliver, M.A. 2001. *Geostatistics for Environmental Scientists*, Chichester: Wiley. XI, 271 s. s.
- WHO. 2008. *Influenza*. World Health Organization, March 2003 2008 [nedlastet 13.05 2008]. Tilgjengelig fra <http://www.who.int/mediacentre/factsheets/2003/fs211/en/>.

# 10 Appendiks

## 10.1 Figurliste

Figur 2.1: Figuren viser hvordan antallet pasienter reduseres frem til diagnose .....	15
Figur 3.1: Utbruddstetthet av høypatogenisk fugleinfluensa (Martin et al. 2007).....	20
Figur 3.2: Illustrasjon av en sirkulær animasjon av temperatur i amerikanske byer gjennom ett år (Midtbø et al. 2007).....	26
Figur 4.1: Oversiktskart over kommuner med influensaregistreringer og antallet registreringer som finnes i den enkelte kommune. ....	29
Figur 5.1: Flytdiagram for analyse av influensadata. Starter med rådata, før man splitter denne opp i flere ulike filer og begynner med ulike analyser som blant annet thiessenpolygoner og kriging. Det blir laget animasjon av datasettene. ....	37
Figur 5.2: Algoritme for dannelse av thiessenpolygoner, her vist ved et coverageformat til ArcGIS (ESRI 2009). ....	38
Figur 5.3: Endring i gjennomsnittlig andel ILS for Norge for hver uke i de ulike sesongene datasettet rekker over. Sesongene 1998 til 1999 (Sesong 1) og 2004 til 2005 (sesong 7) er registrert med både uke 1 og uke 53, derfor er disse linjene lengre enn de andre. ....	42
Figur 5.4: Diagrammet viser den maksimale- og minimale gjennomsnittsverdien for andel ILS for hver sesong. ....	43
Figur 5.5: Diagrammet viser det gjennomsnittlige antallet registreringer pr kommune i hver sesong. ....	44
Figur 5.6: Figuren viser hvordan andel ILS fordeler seg i Norge fordelt på kommuneflatene ved starten av 2006 – 2007 sesongen. Datasettet som er brukt for kommuneflatene er N2000. ....	45
Figur 5.7: Figuren viser hvordan andel ILS fordeler seg i Norge når det er beregnet thiessenpolygoner rundt registreringene ved starten av 2006 – 2007 sesongen. Datasettet som er brukt for kommuneflatene er N2000. ....	46
Figur 5.8: Kartene viser andel ILS i sesong 10 (2007-2008) på ulike presentasjonsmåter. Fra venstre: summert i kommuneflater, gradvis endret størrelse på punkt, og endring i farge på punkt. ....	52

Figur 6.1: Flytdiagrammet viser hvordan det syndrombaserte signalystemet fungerer ved å bruke influensaregistreringene og en informasjonsfil som input, for så å gjøre noen tester og til slutt generere et kart og sende dette på e-post. ....	54
Figur 6.2: Brukergrensesnitt for editering av fil med informasjon om infeksjon. ....	60

## 10.2 Tabelliste

Tabell 4.1: Eksempel på influensaregistreringer.....	30
Tabell 4.2: Eksempel på dobbellagring med hensyn på reg.dato.....	30
Tabell 4.3: Eksempel på dobbellagring, med hensyn på influ, kons, regdato .....	31
Tabell 4.4: Oversikt over kommunene som har 5 eller færre antall konsultasjoner registrert en uke i løpet av perioden.....	32
Tabell 4.5: Oversikt over produserte shape-filer fra influensadata.....	33
Tabell 5.1: Verdier for romlig autokorrelasjon for sesong 9 (2006-2007).....	47
Tabell 5.2: Semivariogram og geostatistiske verdier fra beregning av ordinær kriging. RMS står for <i>root mean squared</i> og beskriver virkningen av en varierende verdi,. ASE er en forkortelse fra <i>Average Standard Error</i> som betegner den gjennomsnittlige standardfeilen. ...	48
Tabell 6.1: Standardverdier for automatisert eksportering av infeksjonskart.....	59
Tabell 6.2: Informasjon om pc-er brukt under testing av signalsystemscriptet.....	61



## 10.3 Programkode for signalsystem

```
#!/usr/bin/env python
""" prosjekt_fhi.py er et script laget i forbindelse med masteroppgave ved
    Institutt for Geofag, som gar ut paa utvikling av et signalsystem for
    deteksjon av forhoyede verdier av andel influensalignende symptomer
    (andel ILS). Masteroppgaven blir skrevet med veiledning fra både Institutt
    for Geofag og Nasjonalt Folkehelseinstitutt. """
import sys, os, arcgisscripting, poplib, smtplib, time, profile
from email.MIMEText import MIMEText
from email.MIMEBase import MIMEBase
from email.MIMEMultipart import MIMEMultipart
from email import encoders

# Dersom det er registrert økt andel ILS, opprett mail og send til ansvarshavende.
def sendMail(din, steder, kartet):
    """
    Oppretter et mailobjekt, legger ved kart og sender mail til angitt person
    og adresse ved hjelp av smtplib.

    * din - tekststreng som inneholder mottakers e-postadresse

    * steder - liste med navn paa de rammede kommunene

    * kartet - tekststreng som stien til det genererte kartet

    """
    #Oppretter informasjon om mail
    me = "torbjorn@dallokken.com"
    you = din
    msg = MIMEMultipart()
    msg.attach(MIMEText("Det er registrert forhoyet forekomst av influensa i kommunene %s" % steder))
    msg['Subject'] = 'Forhoyet forekomst av influensa'
    msg['From'] = me
    msg['To'] = you

    # Oppretter vedleggsdel av mail og koder filen i henhold til base64.
    part = MIMEBase('application','octet-stream')
    part.set_payload(open(kartet,'rb').read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition', 'attachment; filename = "%s"' % os.path.basename(kartet))
    msg.attach(part)

    # Oppretter tilkobling til SMTP server, men inkluderer ikke konvolutthodet
    smtpserver = 'mail.dallokken.com'
    AUTHREQUIRED = 1
    smtpuser = 'torbjorn@dallokken.com'
    smtppass = 'pwd4torbis'

    try: # Forsoeker aa koble til server
        s = smtplib.SMTP(smtpserver,587)
        s.set_debuglevel(1)
        if AUTHREQUIRED:
            s.login(smtpuser, smtppass)
            s.sendmail(me, you, msg.as_string())
        s.close()

    # mottar exception dersom sendmail feiler
```

```

except(smtplib.SMTPSenderRefused, smtplib.SMTPHeloError, smtplib.SMTPRecipientsRefused):
    print "Kunne ikke sende mailen!"
    type, message, traceback = sys.exc_info()
    print 'exception type:', type
    print 'exception message:', message
# mottar exception dersom autentisering feiler
except(smtplib.SMTPAuthenticationError):
    print "Kunne ikke godkjenne brukernavn eller passord"
    type, message, traceback = sys.exc_info()
    print 'exception type:', type
    print 'exception message:', message

# mottar exception dersom kobling mot server feiler
except(smtplib.SMTPConnectError):
    print "Kunne ikke koble til nettverket!\n"
    type, message, traceback = sys.exc_info()
    print 'exception type:', type
    print 'exception message:', message

gp = arcgisscripting.create() # Oppretter objektet for geoprosessering
gp.overwriteoutput = 1      # Programmet skriver over eksisterende filer

usage = "Usage: prosjekt_fhi.py workspaceFolder terskelfil infeksjonsdata"
if len(sys.argv) < 4:
    print usage
    sys.exit(1)

# Oppretter lokale variable
workspace = sys.argv[1]
terskelfil = sys.argv[2]
infeksjoner = sys.argv[3]
input = workspace+infeksjoner
ukeNr = time.strftime("%U", time.gmtime());
aar = 2005
collect_output = "collect_output.shp"
where_clause = "\"UKE\" LIKE '"+str(ukeNr)+"' AND \"AAR\" LIKE '"+str(aar)+"'"
list_gaar = []
tmp_terksel = 'tom'
kartgeneratorDir = "C:\\GIS-data\\Master\\"
prosjektDir = kartgeneratorDir+"Influensa.mxd"
exportName = "influensaKart.pdf"

def lesTerskelFil(terskelfil, workspace, tmp_terksel):
    """
    Metode som henter inn filen med terskelinformasjon om ulike infeksjoner, og
    skriver innholdet til en liste: list_terskler.

    * terskelfil - Fil som inneholder informasjon om infeksjon, terskelverdi, e-post

    * workspace - Katalogen som brukes til aa lagre output og hente input fra

    * tmp_terksel - Variabel som brukes for aa lese inn terskelverdier

    """
    try:
        # Aapner og leser filen, legger inn informasjon i struktur
        fp = open(workspace + terskelfil, 'r')
        while not tmp_terksel == "":
            tmp_terksel = fp.readline()

```

```

        if tmp_terksel == "": break
        tmp_terksel = tmp_terksel.split('\t')
        list_gaar.append(tmp_terksel)
    fp.close()
    return list_gaar
except:
    print "Fant ikke terskel-filen."
    print usage
    sys.exit(0)

def prosessering():
    """
    Metoden prosessering brukes for aa gjøre beregningene og prosesseringen av data
    som ligger i filen som er oppgitt. Mye av prosesseringen skjer via det ESRI
    utviklede modulen arcgisscripting; noe som gjør at det er nødvendig med ArcGIS
    lisens tilgjengelig.
    """

    list_gaar = lesTerskelFil(terskelFil, workspace, tmp_terksel)

    gp.MakeFeatureLayer(input,"inputFC")
    # Aapner tilgang til database og henter ut informasjon fra gaardagen
    try:
        gp.SelectLayerByAttribute("inputFC","NEW_SELECTION",where_clause)
    except:
        print gp.getmessages()

    numValgt = gp.GetCount_management("inputFC") # Finner antall valgte

    # Sjekker om det er registrert noen data fra gitt dato
    if numValgt > 0:
        gp.CopyFeatures("inputFC", workspace + "ukens.shp") # Kopierer utvalg til ny fil
        i = 0
        # Loeper gjennom liste med informasjon om infeksjoner
        for var in list_gaar:
            i = i + 1
            dblTerskel = float(var[0]) # smitteterskel til infeksjon
            txtKontaktavn = var[1] # kontaktperson for infeksjon
            txtEpost = var[2] # e-postadresse til kontaktperson

            # Lag layer av inputFC
            gp.MakeFeatureLayer(workspace + "ukens.shp", "ukens_alle_fl"+str(i))

            nyFC = workspace+"ny_FC"+str(i)

            gp.CopyFeatures("ukens_alle_fl"+str(i), nyFC)
            nyFC2 = nyFC+".shp"

            # Beregner andel ILS
            gp.AddField(workspace+"ny_FC"+str(i)+".shp","ILS","DOUBLE")
            try:
                gp.CalculateField_management(workspace+"ny_FC"+str(i)+".shp", "ILS", "([INFLU] /
[KONS])*100")
            except:
                print gp.getmessages()

            gp.MakeFeatureLayer(workspace+"ny_FC"+str(i)+".shp","FC_ins")

```

```

# Sjekker antall tilfeller med terkselveri i array basert paa infeksjon
gp.SelectLayerByAttribute("FC_ins", "new_selection", "ILS >= "+str(dbITerskel))

# Sjekker antall valgte tupler
numSelected = gp.GetCount_management("FC_ins")

# Selekerer rader i tabellen og henter kommunenavnet.
rows = gp.SearchCursor("FC_ins", "ILS >= "+str(dbITerskel))
row = rows.next()
kommuner = []
while row:
    kommuner.append(row.Kommune)
    row = rows.next()

# Sjekker om det er noen hoeyere ILS enn terskelverdien i datasettet
if numSelected > 0:
    # Starter kartgeneratoren utviklet i Microsoft Visual Studio.
    os.system(kartgeneratorDir+"Kartgenerering.exe"+" "+prosjektDir+" "+kartgeneratorDir+"
"+exportName)

    # Sender e-postadressa, kommuneliste og filen med kartet til sendmail funksjonen
    sendMail(txtEpost, kommuner, kartgeneratorDir+exportName)

prosessering()

```

## 10.4 Programkode for generering av kart

```
// Importerer de nødvendige modulene for bruk til generering
using ESRI.ArcGIS.ADF;
using ESRI.ArcGIS.ArcMapUI;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.CartoUI;
using ESRI.ArcGIS.Controls;
using ESRI.ArcGIS.DataSourcesFile;
using ESRI.ArcGIS.Display;
using ESRI.ArcGIS.Framework;
using ESRI.ArcGIS.Geodatabase;
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.Output;
using ESRI.ArcGIS.esriSystem;
using ESRI.ArcGIS.SystemUI;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace Kartgenerering
{
    class Program
    {
        // Oppretter et LicenseInitializer objekt for å sjekke om pc har tilgang til ArcGIS ArcView
        private static LicenseInitializer m_AOLicenseInitializer = new Kartgenerering.LicenseInitializer();

        /* GDI delegate to GetDeviceCaps function */
        [DllImport("GDI32.dll")]
        public static extern int GetDeviceCaps(int hdc, int nIndex);

        /* User32 delegates to getDC and ReleaseDC */
        [DllImport("User32.dll")]
        public static extern int GetDC(int hWnd);

        [DllImport("User32.dll")]
        public static extern int ReleaseDC(int hWnd, int hdc);

        [DllImport("user32.dll", SetLastError = true)]
        static extern bool SystemParametersInfo(uint uiAction, uint uiParam, ref int pvParam, uint fWinIni);

        [System.Runtime.InteropServices.DllImport("user32.dll")]
        public extern static int GetDesktopWindow();

        /* constants used for user32 calls */
        const uint SPI_GETFONTSMOOTHING = 74;
        const uint SPI_SETFONTSMOOTHING = 75;
        const uint SPIF_UPDATEINIFILE = 0x1;

        /* Oppretter globale variable som blir brukt for å bearbeide kartet*/
        static IMapDocument pMapDocument = new MapDocumentClass();
        static IPage pPage = new PageClass();
        static IActiveView aView;
        static String outputFile;
    }
}
```

```

static String outputPath;
static void Main(string[] args)
{
    // Tester om det finnes tilgjengelig ArcGIS-ArcView lisens. ESRI License Initializer generated code.
    if (!m_AOLicenseInitializer.InitializeApplication(new esriLicenseProductCode[] {
esriLicenseProductCode.esriLicenseProductCodeArcView },
    new esriLicenseExtensionCode[] { }))
    {
        System.Console.WriteLine(m_AOLicenseInitializer.LicenseMessage());
        System.Console.WriteLine("This application could not initialize with the correct ArcGIS license and
will shutdown.");
        m_AOLicenseInitializer.ShutdownApplication();
        return;
    }

    /* Sjekker at programmet er startet med korrekt input.*/
    if (args.Length == 0) // Ikke noe input er gitt, gir feilmelding og avslutter
programmet
        System.Console.WriteLine("Feil...");
    else
    {
        pMapDocument.Open(args[0], null); // Input er korrekt angitt, bruker argument 1 som
kartprosjekt
        outputPath = args[1].Replace("\\", "\\\\"); // Bruker input 2 som katalog hvor kart plasseres,
erstatte \ med \\.
        outputFile = args[2]; // Bruker input 3 som filnavn på kart som skal genereres.
    }

    aView = pMapDocument.ActiveView; // Assigns the documents active view to new
activeview
    aView.Activate(GetDesktopWindow()); // Aktiverer den aktive view'et i ArcMapactivates the
active view in ArcMap
    exportActiveView(); // Kaller metoden exportActiveView() for å eksportere den
aktive layouten.
    System.Console.WriteLine("Ferdig med eksportering"); // Skriver en melding til konsoll når
eksportering er ferdig.
    pMapDocument.Close(); // Lukker kartdokumentet.

    //Avslutter programmet. ESRI License Initializer generated code.
    //Do not make any call to ArcObjects after ShutdownApplication()
    m_AOLicenseInitializer.ShutdownApplication();
}
/* Exports the active view */
public static void exportActiveView()
{
    long iOutputResolution = 300; // Angir oppløsningen av eksportert kart, standard er
300dpi
    long lResampleRatio = 1;
    string ExportType = outputFile.Substring(outputFile.Length-3).ToUpper(); // Finner ut hvilket filformat
ut fra filnavnet i input
    string sOutputDir = outputPath; // Angir katalogen hvor kart skal plasseres
    Boolean bClipToGraphicsExtent = false; // Angir om kartet skal klippes

    IActiveView docActiveView = aView;
    IExport docExport;
    long iPrevOutputImageQuality;
    IOutputRasterSettings docOutputRasterSettings;
    IEnvelope PixelBoundsEnv;
    tagRECT exportRECT;
    tagRECT DisplayBounds;

```

```

IDisplayTransformation docDisplayTransformation;
IPageLayout docPageLayout;
IEnvelope docMapExtEnv;
long hdc;
long tmpDC;
string sNameRoot;
long iScreenResolution;
bool bReenable = false;

IEnvelope docGraphicsExtentEnv;
IUnitConverter pUnitConvertor;

if (GetFontSmoothing())
{
    /* glatting av skrifttype er slått på, slå av denne og slå på igjen senere */
    bReenable = true;
    DisableFontSmoothing();
    if (GetFontSmoothing())
    {
        // glatting av skrift kunne ikke slås av.
        return;
    }
    // glatting av skrift ble slått av.
}

// Initiserer en export-klasse ut fra hvilket format som er angitt.

if (ExportType == "PDF")
{
    docExport = new ExportPDFClass();
}
else if (ExportType == "EPS")
{
    docExport = new ExportPSClass();
}
else if (ExportType == "AI")
{
    docExport = new ExportAIClass();
}
else if (ExportType == "BMP")
{
    docExport = new ExportBMPClass();
}
else if (ExportType == "TIFF")
{
    docExport = new ExportTIFFClass();
}
else if (ExportType == "SVG")
{
    docExport = new ExportSVGClass();
}
else if (ExportType == "PNG")
{
    docExport = new ExportPNGClass();
}
else if (ExportType == "GIF")
{
    docExport = new ExportGIFClass();
}

```

```

    }
    else if (ExportType == "EMF")
    {
        docExport = new ExportEMFClass();
    }
    else if (ExportType == "JPEG")
    {
        docExport = new ExportJPEGClass();
    }
    else
    {
        System.Console.WriteLine("Unsupported export type " + ExportType + ", defaulting to EMF.");
        ExportType = "EMF";
        docExport = new ExportEMFClass();
    }

    // lagre det forrige valgte utkvaliteten, slik at den resettes når eksporteringen er ferdig.
    docOutputRasterSettings = docActiveView.ScreenDisplay.DisplayTransformation as
IOOutputRasterSettings;
    iPrevOutputImageQuality = docOutputRasterSettings.ResampleRatio;

    if (docExport is IExportImage)
    {
        // for bildeformater, så settes kvaliteten til 1
        SetOutputQuality(docActiveView, 1);
    }
    else
    {
        // for vektorformat, settes ønsket ResampleRatio for å kontrollere tegning av rastere under eksportering
        SetOutputQuality(docActiveView, iResampleRatio);
    }

    // setter korrekt grunnlagsnavn på output-kartfil
    sNameRoot = outputFile.Substring(0,outputFile.Length-4);

    // angir navnet til den eksporterte kartfilen (grunnlagsnavn + filtype)
    docExport.ExportFileName = sOutputDir + sNameRoot + "." +
docExport.Filter.Split('.')[1].Split('(')[0].Split(')')[0];

    /* mottar innholdet på skjermen */
    tmpDC = GetDC(0);
    /* mottar skjermens oppløsning */
    iScreenResolution = GetDeviceCaps((int)tmpDC, 88); //88 er win32 sin konstanfor logiske piksler/inch i
X)
    ReleaseDC(0, (int)tmpDC);
    docExport.Resolution = iOutputResolution;

    if (docActiveView is IPageLayout)
    {
        // henter grensene til eksporteringskartet på det aktivelaget.
        DisplayBounds = docActiveView.ExportFrame;
        // setter opp pGraphicsExtent, som blir brukt hvis det skal klippes til grafikkomraadet.
        docGraphicsExtentEnv = GetGraphicsExtent(docActiveView);
    }
    else
    {

```



```

// henter grensene for skjermen
docDisplayTransformation = docActiveView.ScreenDisplay.DisplayTransformation;
DisplayBounds = docDisplayTransformation.get_DeviceFrame();
}

PixelBoundsEnv = new Envelope() as IEnvelope;

if (bClipToGraphicsExtent && (docActiveView is IPageLayout))
{
docGraphicsExtentEnv = GetGraphicsExtent(docActiveView);
docPageLayout = docActiveView as PageLayout;
pUnitConvertor = new UnitConverter();

// angir x og y verdier som representerer det klippede omraadet til PixelBounds.
PixelBoundsEnv.XMin = 0;
PixelBoundsEnv.YMin = 0;
PixelBoundsEnv.XMax = pUnitConvertor.ConvertUnits(docGraphicsExtentEnv.XMax,
docPageLayout.Page.Units, esriUnits.esriInches) * docExport.Resolution -
pUnitConvertor.ConvertUnits(docGraphicsExtentEnv.XMin, docPageLayout.Page.Units, esriUnits.esriInches) *
docExport.Resolution;
PixelBoundsEnv.YMax = pUnitConvertor.ConvertUnits(docGraphicsExtentEnv.YMax,
docPageLayout.Page.Units, esriUnits.esriInches) * docExport.Resolution -
pUnitConvertor.ConvertUnits(docGraphicsExtentEnv.YMin, docPageLayout.Page.Units, esriUnits.esriInches) *
docExport.Resolution;

// angir x og y verdier som representerer det klippede omraadet til exportRECT.
exportRECT.bottom = (int)(PixelBoundsEnv.YMax) + 1;
exportRECT.left = (int)(PixelBoundsEnv.XMin);
exportRECT.top = (int)(PixelBoundsEnv.YMin);
exportRECT.right = (int)(PixelBoundsEnv.XMax) + 1;

// setter synlige grenser.
docMapExtEnv = docGraphicsExtentEnv;
}
else
{
double tempratio = iOutputResolution / iScreenResolution;
double tempbottom = DisplayBounds.bottom * tempratio;
double tempright = DisplayBounds.right * tempratio;
// verdiene i exportRECT tagRECT korresponderer med bredde
// og hoedye for eksportering, maalt i piksler med start i oeverste venstre hjoerne.
exportRECT.bottom = (int)Math.Truncate(tempbottom);
exportRECT.left = 0;
exportRECT.top = 0;
exportRECT.right = (int)Math.Truncate(tempright);

// fyller PixelBounds med verdier fra exportRECT.
PixelBoundsEnv.PutCoords(exportRECT.left, exportRECT.top, exportRECT.right,
exportRECT.bottom);

// trenger ikke docMapExtEnv ettersom det er et kartview.
docMapExtEnv = null;
}

// tilordner konvoluttobjektet til eksporteringsobjektets PixelBounds egenskap.
docExport.PixelBounds = PixelBoundsEnv;

// kaller metoden StartExporting
hdc = docExport.StartExporting();

```

```

// Tegner aktivt view på nytt, sender det til eksporteringsobjektet i stedet for til program GUI.
docActiveView.Output((int)hdc, (int)docExport.Resolution, ref exportRECT, docMapExtEnv, null);

// ferdig med eksportering og rydder opp.
docExport.FinishExporting();
docExport.Cleanup();

System.Console.WriteLine("Finished exporting " + sOutputDir + sNameRoot + "." +
docExport.Filter.Split('.')[1].Split('.')[0].Split('.')[0] + ".", "Export Active View Sample");

// setter opploesningen tilbake til den originale verdien i kartprosjektet
SetOutputQuality(docActiveView, iPrevOutputImageQuality);
if (bReenable)
{
    /* slaa paa glatting av skrift dersom det var slaaet av */
    EnableFontSmoothing();
    bReenable = false;
    if (!GetFontSmoothing())
    {
        //error: kan ikke slaa paa glatting.
        System.Console.WriteLine("Unable to reenable Font Smoothing", "Font Smoothing error");
    }
}

docMapExtEnv = null;
PixelBoundsEnv = null;

}

private static void SetOutputQuality(IActiveView docActiveView, long iResampleRatio)
{
    /* Denne funksjonen setter ut-kvaliteten til det aktive viewet. Dersom det aktive view er layout,
    * maa metoden ogsaa sette bildekvalitet for hvert kart i layout'en.
    */
    IGraphicsContainer oiqGraphicsContainer;
    IElement oiqElement;
    IOutputRasterSettings docOutputRasterSettings;
    IMapFrame docMapFrame;
    IActiveView TmpActiveView;

    if (docActiveView is IMap)
    {
        docOutputRasterSettings = docActiveView.ScreenDisplay.DisplayTransformation as
IOutputRasterSettings;
        docOutputRasterSettings.ResampleRatio = (int)iResampleRatio;
    }
    else if (docActiveView is IPageLayout)
    {
        // tilordner ResampleRatio for PageLayout
        docOutputRasterSettings = docActiveView.ScreenDisplay.DisplayTransformation as
IOutputRasterSettings;
        docOutputRasterSettings.ResampleRatio = (int)iResampleRatio;
        // tilordner ResampleRatio for kartene i layout
        oiqGraphicsContainer = docActiveView as IGraphicsContainer;
        oiqGraphicsContainer.Reset();

        oiqElement = oiqGraphicsContainer.Next();
        while (oiqElement != null)

```

```

    {
        if (oiqElement is IMapFrame)
        {
            docMapFrame = oiqElement as IMapFrame;
            TmpActiveView = docMapFrame.Map as IActiveView;
            docOutputRasterSettings = TmpActiveView.ScreenDisplay.DisplayTransformation as
IOutputRasterSettings;
            docOutputRasterSettings.ResampleRatio = (int)iResampleRatio;
        }
        oiqElement = oiqGraphicsContainer.Next();
    }

    docMapFrame = null;
    oiqGraphicsContainer = null;
    TmpActiveView = null;
}
docOutputRasterSettings = null;

}

private static IEnvelope GetGraphicsExtent(IActiveView docActiveView)
{
    /* Henter det kombinerte omraadet av alle objektene i kartet. */
    IEnvelope GraphicsBounds;
    IEnvelope GraphicsEnvelope;
    IGraphicsContainer oiqGraphicsContainer;
    IPageLayout docPageLayout;
    IDisplay GraphicsDisplay;
    IElement oiqElement;

    GraphicsBounds = new EnvelopeClass();
    GraphicsEnvelope = new EnvelopeClass();
    docPageLayout = docActiveView as IPageLayout;
    GraphicsDisplay = docActiveView.ScreenDisplay;
    oiqGraphicsContainer = docActiveView as IGraphicsContainer;
    oiqGraphicsContainer.Reset();

    oiqElement = oiqGraphicsContainer.Next();
    while (oiqElement != null)
    {
        oiqElement.QueryBounds(GraphicsDisplay, GraphicsEnvelope);
        GraphicsBounds.Union(GraphicsEnvelope);
        oiqElement = oiqGraphicsContainer.Next();
    }

    return GraphicsBounds;
}

private static void DisableFontSmoothing()
{
    bool iResult;
    int pv = 0;

    /* kaller metoden systemparametersinfo for aa tilordne verdien som skrift skal glattes med */
    iResult = SystemParametersInfo(SPI_SETFONTSMOOTHING, 0, ref pv, SPIF_UPDATEINIFILE);
}

private static void EnableFontSmoothing()
{

```

```

bool iResult;
int pv = 0;

/* kaller metoden systemparametersinfo for aa tilordne verdien som skrift skal gattes med */
iResult = SystemParametersInfo(SPI_SETFONTSMOOTHING, 1, ref pv, SPIF_UPDATEINIFILE);

}

private static Boolean GetFontSmoothing()
{
    bool iResult;
    int pv = 0;

    /* kaller metoden systemparametersinfo for aa hente skriftenes glatteverdi */
    iResult = SystemParametersInfo(SPI_GETFONTSMOOTHING, 0, ref pv, 0);

    if (pv > 0)
    {
        //pv > 0 betyr at glatting er paaslaat.
        return true;
    }
    else
    {
        //pv == 0 betyr at glatting er avslaet.
        return false;
    }
}
}
}

```

## 10.5 Programkode for signalEditor – hovedfunksjoner

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using System.IO;
using System.Collections;

namespace SignalEditor
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            modell m = new modell();           // Kaller klassen modell
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            ArrayList infeksjonsListe = new ArrayList(); // Oppretter arraylist

            try
            {
                infeksjonsListe = m.lesFraFil();    // Leser fra fil
            }
            catch
            {
                Console.WriteLine("...");
                infeksjonsListe = null;
            }
            Application.Run(new frmInfeksjonregistrering(m, infeksjonsListe)); // Starter program
        }
    }

    public class modell
    {
        ArrayList infeksjoner = new ArrayList();
        Infeksjoner infeksjonen;

        // Metode som leser informasjon fra filen
        public ArrayList lesFraFil()
        {
            // Oppretter en StreamReaderobjekt for å lese inn data fra fil
            StreamReader inFile = new StreamReader("terskel.dat");
            String linje;

            while ((linje = inFile.ReadLine()) != null) // Leser inn linje fra fil
            {
                String[] element = linje.Split('\t'); // Splitter linjen med hensyn til tab
                infeksjonen = new Infeksjoner(double.Parse(element[0]),element[1],element[2],
                    element[3],element[4]); // Oppretter nytt infeksjonsobjekt med info
                infeksjoner.Add(infeksjonen); // Lagrer informasjonen i arraylisten
            }
            inFile.Close(); // Lukker filen
        }
    }
}
```

```

    return infeksjoner;
}

// Metoden skriver informasjonen som finnes i arraylisten til filen
public void skrivTilFil()
{
    StreamWriter outFile = new StreamWriter("terskel.dat", false);
    for (int i = 0; i < infeksjoner.Count; i++)
    {
        infeksjonen = (Infeksjoner)infeksjoner[i];
        outFile.WriteLine(infeksjonen.terksel + "\t" +
infeksjonen.epost+"\t"+infeksjonen.prosjekt+"\t"+infeksjonen.kartMappe+"\t"
        +infeksjonen.kartNavn);
    }
    outFile.Close();
}

// Lagrer ny infeksjon i arraylisten
public void lagreNy(String terskel,String epost, String prosjekt, String kartMappe, String kartNavn )
{
    infeksjonen = new Infeksjoner(double.Parse(terskel),epost,prosjekt,kartMappe,kartNavn);
    infeksjoner.Add(infeksjonen);
}

}

/** Klassen representerer Infeksjoner
 * Brukes for å opprette infeksjonsobjekter
 *
 * **/
class Infeksjoner
{
    // Variabler som inneholder informasjon om infeksjonen
    public double terksel = 0.0;
    public String epost = null;
    public String prosjekt = null;
    public String kartMappe = null;
    public String kartNavn = null;

    // Konstruktør som oppretter infeksjonsobjekt
    public Infeksjoner(double terskel, String epost,String prosjekt, String kartMappe, String kartNavn)
    {
        this.terksel = terskel;
        this.epost = epost;
        this.prosjekt = prosjekt;
        this.kartMappe = kartMappe;
        this.kartNavn = kartNavn;
    }
}
}

```

## 10.6 Autokorrelasjonsverdier

Sesong 1:

Uke	Z-Score	Moran's I index
40	-0,49	-0.02
50	1,5	0.01
1	-0,45	-0.02
10	1.08	0.01
20	0,8	0,01

Sesong 2:

Uke	Z-Score	Moran's I index
40	-0,32	-0,02
50	0,83	0
1	1,56	0,02
10	-0,3	-0,01
20	-0,18	-0,01

Sesong 3:

Uke	Z-Score	Moran's I index
40	-0,52	-0,02
50	4,95	0,06
1	3,4	0,04
10	0,26	0
20	1,94	0,02

Sesong 4:

Uke	Z-Score	Moran's I index
40	-0,66	-0,02
50	3,07	0,04
1	3,42	0,05
10	1,18	0,01
20	1,28	0,02

Sesong 5:

Uke	Z-Score	Moran's I index
40	-0,02	-0,01
50	0,41	0
1	0,76	0
10	-0,53	-0,02
20	0,2	-0,01

Sesong 6:

Uke	Z-Score	Moran's I index
40	0,45	0
50	3,16	0,04
1	-0,07	-0,01
10	1,04	0,01
20	0,97	0,01

Sesong 7:

Uke	Z-Score	Moran's I index
40	-0,23	-0,01
50	-0,45	-0,01
1	0,62	0
10	2,92	0,03
20	1,4	0,01



Sesong 8:

<b>Uke</b>	<b>Z-Score</b>	<b>Moran's I index</b>
<b>40</b>	0,34	0
<b>50</b>	-0,55	-0,01
<b>1</b>	-0,91	-0,02
<b>10</b>	-0,45	-0,01
<b>20</b>	-0,35	-0,01

Sesong 9:

<b>Uke</b>	<b>Z-Score</b>	<b>Moran's I index</b>
<b>40</b>	0.73	0.003454
<b>50</b>	-0.92	-0.02
<b>1</b>	0.51	0
<b>10</b>	0.95	0.01
<b>20</b>	0.27	0