

A26649 - Åpen

Rapport

Data og metoder for modellering av biltrafikkens fart i transportmodeller

Forfattere:

Tomas Levin, Trude Tørset, Olav Kåre Malmin og Ola Rennemo



Rapport

Data og metoder for modellering av biltrafikkens fart i transportmodeller

EMNEORD:Fart
Transportmodell**VERSJON**

1.00

DATO

2015-01-06

FORFATTERE:Tomas Levin
Trude Tørset, Olav Kåre Malmin og Ola Rennemo**OPPDRAGSGIVER**

NTP Transportanalyser

OPPDRAGSGIVERS REF.

Oskar Kleven

PROSJEKTNR

102005718

ANTALL SIDER OG VEDLEGG:

32 + vedlegg

SAMMENDRAG

Denne rapporten dokumenterer arbeid gjennomført med videreutvikling av fartsrepresentasjonen i de regionale transportmodellene. I dette arbeidet er fartsreduksjoner delt inn i fire kategorier etter årsak: *i)* vegens geometriske utforming på strekninger *ii)* sammenhengen mellom volum/hastighet på strekninger *iii)* geometrisk forsinkelse i kryss *iv)* trafikkrelaterte forsinkelser i kryss.

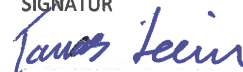
Studiene knyttet til fart er hovedsakelig basert på empiriske studier, men inneholder også pragmatiske avveininger slik at utviklet metodikk kan benyttes i dagens utgave av den regionale transportmodellen.

Fartsmodellen som er utviklet i dette prosjektet er en forbedring av dagens metode, men prosjektet har også avdekket et potensiale for å utvikle modellen videre, etter hvert som datagrunnlaget utvides.

UTARBEIDET AV

Tomas Levin

SIGNATUR

**KONTROLLERT AV**

Unn Karin Thorenfeldt

SIGNATUR

**GODKJENT AV**

Roar Norvik

SIGNATUR

**RAPPORTNR**

A26649

ISBN

9788214058000

GRADERING

Åpen

GRADERING DENNE SIDE

Åpen

Historikk

VERSJON	DATO	VERSJONSBESKRIVELSE
1.0	2015-02-02	Rapport

Forord

Den regionale transportmodellen utvikles med stadig mer funksjonalitet samtidig som det arbeides for at den skal gi raske beregninger og realistiske resultater. Ofte har utviklingsarbeidet i modellen vært nært knyttet til et konkret problem som man har søkt en løsning for. Dette arbeidet har hatt en litt annen innretning ved at vi har tatt utgangspunkt i muligheten for å gjøre beregningene bedre ved introduksjon av nytt datagrunnlag. Initiativet kom etter at vi fikk tilgang til geometrisk utforming av vegnettet og GPS-data fra Statens vegvesens tjenestebiler. Da så vi muligheten for å etablere en beregning av friflytfart som kunne erstatte bruken av fartsgrensene til dette. Prosjektet søkte også relativt vidt etter data og metoder som kunne forbedre andre deler av fartsberegningen i transportmodellen, og vi har avdekket at det fortsatt er et stort potensiale for utvikling av bedre beregning av fartsnivå i modellen.

Fra SINTEF har flere av medarbeiderne vært involvert. Tomas Levin har hatt et hovedansvar for bearbeiding av data, analyser og dokumentasjon, et arbeid som ble fullført av Trude Tørset. Olav Kåre Malmin og Ola Rennemo har samarbeidet om implementering av metodikk i TNExt og i den Regionale transportmodellen for persontransport, implementeringen er dokumentert i denne rapporten.

Arbeidet er finansiert av NTP Transportanalyser og Statens vegvesen har stilt data til rådighet for prosjektet. En arbeidsgruppe ledet av Oskar Kleven fra Statens vegvesen Vegdirektoratet har fulgt prosjektet, og har bidratt med viktige erfaringer og synspunkt underveis. Erfaringene til Tore Moan fra Statens vegvesen Region midt og Stig Nyland Andersen Statens vegvesen Region vest var også viktige bidrag, spesielt for den delen som handler om forsinkelse på strekninger forårsaket av trafikk tetthet.

Trondheim, januar 2015



Roar Norvik
Forskningsjef

Sammendrag

Denne rapporten dokumenterer arbeid gjennomført med videreutvikling av fartsrepresentasjonen i de regionale transportmodellene. I dette arbeidet er fartsreduksjoner delt inn i fire kategorier etter årsak:

- i) Vegens geometriske utforming på strekninger
- ii) Sammenhengen mellom volum/hastighet på strekninger
- iii) Geometrisk forsinkelse i kryss
- iv) Trafikkrelaterte forsinkelser i kryss

Studiene knyttet til fart er hovedsakelig basert på empiriske studier, men inneholder også pragmatiske avveininger slik at utviklet metodikk kan benyttes i dagens utgave av den regionale transportmodellen.

Fartsmodellen som er utviklet i dette prosjektet er en forbedring av dagens metode, men prosjektet har også avdekket et potensiale for å utvikle modellen videre, etter hvert som datagrunnlaget utvides.

Summary

SINTEF has developed an improved speed representation for passenger cars in the Regional Transport Model for passenger transport. The speed reduction, compared to a speed limit, is calculated in four sequential steps by its cause.

- v) The geometric design of the road sections
- vi) The link between traffic flow and speed on road sections
- vii) The geometric design of the intersections
- viii) Traffic flows and interactions in intersections

This work is based on empirical studies, but includes pragmatic considerations, to allow usage in the current version of the transport model.

The speed model developed in this project is an improvement from the former methods, but more importantly the project has detected a potential for further development and improvement as the available data increases.

Innholdsfortegnelse

1	Bakgrunn	7
2	Geometrisk utforming	9
2.1	Etablering av radius på et vilkårlig vegelement?	9
2.2	Fartsmodeller for horisontal kurvatur	12
2.3	Fartsmodell for stigninger og fall	13
2.4	Testing av modellene	13
2.5	Visuelle inspeksjoner av hvilke veger som får reduksjoner i hastigheter	16
2.6	Oppsummering fart som følge av veggeometri	16
3	Forsinkelse på strekning som følge av trafikk	17
3.1	Datsett og metode	17
3.2	Resultater	19
4	Registrerte hastigheter gjennom kryss	21
4.1	Datsett og metode	21
4.2	Valg av statistisk parameter lavtrafikk	21
4.3	Valg av statistisk parameter rushtrafikk	22
4.4	Registrerte forsinkelse i ulike krysstyper	23
4.4.1	Forsinkelser i lavtrafikk	23
4.4.2	Forsinkelser i rushtidstimen	24
4.5	Tilpasning til bruk	24
5	Implementering i TNEExt og i transportmodellen	26
5.1	Implementering av fartsmodell i TNEExt.	26
5.2	Bruk av fartsmodelldata i Regionale Persontransportmodeller (RTM)	26
6	Videre arbeid	27
6.1	Geometrisk forsinkelse	27
6.2	Trafikkbelastning og forsinkelser på strekninger	27
6.3	Geometrisk forsinkelse gjennom kryss	28
6.4	Trafikkbelastning og forsinkelser gjennom kryss	28
7	Referanser	29

1 Bakgrunn

Reisetider er en viktig faktor i transportanalyser, og reisetiden langs bestemte ruter er bestemt av den farten trafikken holder. Kjørearten påvirkes av en rekke faktorer som geometrisk utforming, værforhold og trafikk (Sakshaug og Toftegård, 2011).

Nye vegtiltak utenfor byområder fører ofte reelt sett til endringer i kjøretid som følge av redusert reiselengde (veginnkorting), men også som følge av forbedret geometri, fordi nye standarder for vegutformingen anvendes. Dagens metodikk i Regional TransportModell for persontransport (Tørset m fl, 2013, Malmin, 2013) bruker volum-hastighetskurver for å avgjøre fartsnivået på veglenkene, og disse tar utgangspunkt i en fartsutvikling hvor fartsgrensen holdes ved lite trafikk og farten nedskrives gradvis i forhold til økning trafikkvolumet på lenkene. Geometrien vil kun redusere lenkefarten i modellen hvis det er gjort manuelle korreksjoner. Fartsgrensen blir ikke gitt ut fra geometriforhold eller hvilken fart som anbefales, derfor kan vegnett med relativt utfordrende geometri ha mye høyere fartsgrense enn trafikantene kan holde eller hva som er anbefalt ut fra trafiksikkerhetshensyn. Generelt kan dette føre til at farten på eksisterende vegnett er for høy. Dette kan gi bil en urealistisk fordel fremfor andre reisemidler. Innsparing i reisetid inngår i nyttekostnadsanalyser, og utgjør ofte en betydelig andel av nyttesiden ved analyser av vegtiltak. Denne gevinsten blir undervurdert dersom fartsnivået i sammenligningsalternativet allerede er like høyt som i det planlagte tiltaket.

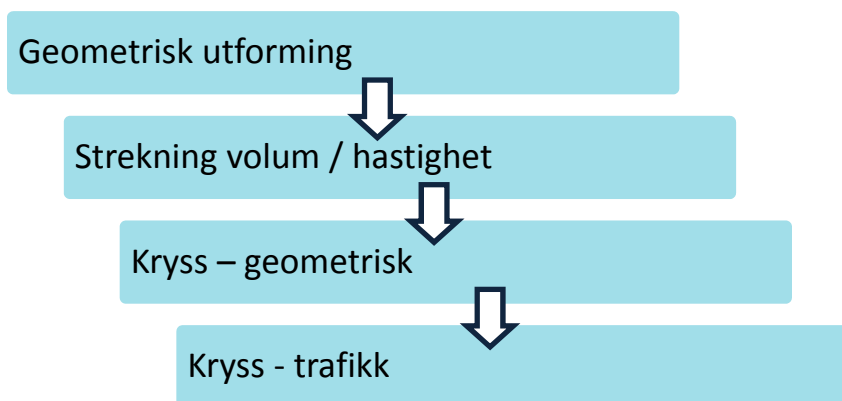
Denne rapporten dokumenterer prosessen med å ta fram kodingsprinsipper og nye volum/hastighets-kurver som skiller mellom reisetidsforsinkelser skapt av veggeometri og trafikk. Dette gjøres ved å dele inn reisetidsberegninger i flere separate trinn som er tilpasset strategiske transportmodeller.

Utviklingen av ITS og fremveksten av nøyaktige posisjoneringsenheter og kraftige IT-verktøy gir flere nye muligheter enn de som var tilgjengelig når RTM-modellsystemet først ble bygget. I dette prosjektet har vi tatt i bruk nye datakilder for å kunne vurdere beregningsmodeller for reisetider i forhold til observerte data.

I dette prosjektet har vi valgt å bruke en sekvensiell, firedelt modell for beregning av kjørehastighet. Delene er ikke uavhengige av hverandre, men kan ses som et hierarki av faktorer som påvirker farten, se Figur 1. I prinsippet vil lavere trinn overstyre fartsreduksjoner fra trinnene over. Denne måten å tenke fartsbestemmelse på er inspirert av metoden utviklet i Fartsmodellprosjektet for næringslivets transporter (Tørset mfl. 2011).

Vi forutsetter at på en vilkårlig rett og flat vegstrekning, uten annen trafikk eller kryss, vil trafikantene velge selv hvilket fartsnivå de skal holde. Da ligger det gjennomsnittlige fartsnivået rundt fartsgrensen. Vi ser foreløpig bort fra situasjoner hvor værforholdene reduserer fartsnivået. Sånn sett kunne fartsgrensen representert et nivå over det første trinnet. Her har vi konsentrert modelloppbyggingen om forhold som gjør at trafikantene må holde lavere fart enn fartsgrensen. Geometriske forhold består i hovedsak av krappe svinger, stigninger eller unnabakke som naturlig fører til at trafikantene velger lavere fart. Her vil selvsagt glatte veier kunne gi enda lavere fart enn på sommerføre, men vi har konsentrert innsatsen om reduksjoner kun pga. geometri. Hvis de geometriske forholdene på en bestemt strekning ikke gir lavere fart enn fartsgrensen, så er det fortsatt den som gjelder. Kryss vil som regel kunne påvirke farten ytterligere nedover. Selv om trafikken er lav, vil vinklene i kryss, kryssutformingen og kryssreguleringen være med å bestemme farten gjennom krysset. Slike forhold har vi bevisst valgt å beregne for seg, og kalt det for geometrisk forsinkelse gjennom kryss. Det er fordi kryssforsinkelse som kommer av interaksjon mellom kryssende trafikk er mye mer uforutsigbar og er derfor noe mer krevende å lage en beregningsmodell for.

Rapporten presenterer metodikk valgt og benyttet for å kunne gi bedre estimat på reisetiden i vegnettet, basert på nye datakilder, blant annet data om vegens geometri hentet fra digitale vegbeskrivelser.



Figur 1: Sekvensiell beregning av hastighet

Tematisk følger rapporten innledningsvis rekkefølgen i den sekvensielle beregningen. Så følger et kapittel som dokumenterer implementering av fartsberegning i TNext og RTM. Avslutningsvis gir rapporten anbefalinger om videre arbeid, noe som kan bli mulig allerede i løpet av 2015 når flere datakilder er tilgjengelige.

2 Geometrisk utforming

Vegens geometriske utforming betyr mye for fartsvalg, da tilgjengelig friksjon og fart vil avgjøre om kjøretøy klarer å fullføre en krapp kurve med ønsket komforten i kupéen, når sideveiskreftene merkes. Sammenhengen mellom sidefriksjon, fart og radius kan beskrives ved formelen¹:

$$Radius = \frac{fart^2}{g \left(Sidefriksjon + \frac{tverrfall}{100} \right)}$$

g	står for gravitasjonskonstanten på 9.81 m/s ²
Sidefriksjonen	er et enhetsløsttall som uttrykker «hvor godt veigrepet er»
Tverrfall	beskriver doseringen i kurven i %
Farten	er farten som man ønsker å holde gjennom kurven

Ved bruk av denne formelen kan man regne seg fram til en trygg radius, gitt at man ønsker et bestemt fartsnivå. Man kan videre snu på formelen for å vurdere hvilken hastighet man trygt kan holde gjennom en kurve. Ønsket sidefriksjon er ikke et enkelt tall å finne, da dette beskriver forholdet mellom to flater (vegen og bildekket). I tillegg kan noen sjåførere bruke deler av motgående kjørefelt og «kutte svinger», noe som gjør at radiusen på den fulgte kurven øker eller at vegen er bygget med breddeutvidelse som tillater sjåføren et linjevalg som vil oppleves som en «radiusøkning».

En ytterligere kompliserende faktor er sjåførenes individuelle fartsvalg. Forskjellige sjåførere vil velge forskjellig fart gjennom kurver basert på egne preferanser i forhold til komfort i kupéen eller andre vurderinger. I trafikksikkerhetsstudier benyttes ofte 85 prosentilet som mål på hastighet; det vil si den hastigheten som 85 % av sjåførene holder seg under. I prosjektet har vi derfor valgt å bruke samme definisjon på fart for å gjøre koblingen til andre fartsstudier så enkel som mulig. Med andre ord er ikke farten som beregnes gjennomsnittshastigheten, men farten 85 % av sjåførene vil holde seg under. Dette passer også bruksområdet til den beregnede farten fordi vi ved å bruke 85 prosentilet, sikrer at det ikke er andre faktorer, for eksempel annen trafikk, som har vært med på å overstyre fartsvalget. Resultatet blir en fart som kan regnes som friflytfart på vegstreknings uten kryss.

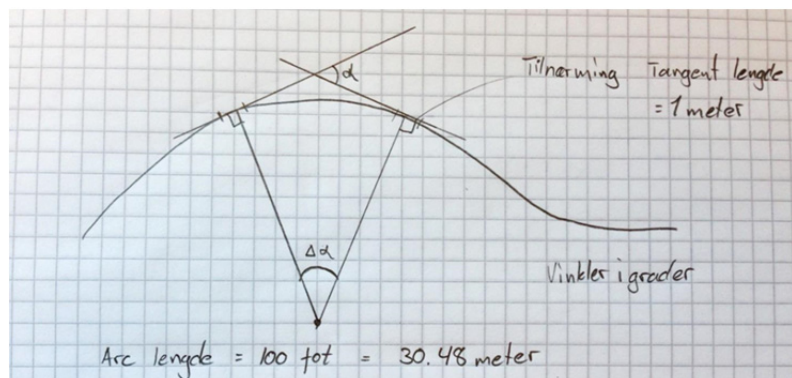
Prosjektet har også vurdert hvorvidt fartsmodellen skal være ensidig eller tosidig i forhold til fartsgrensen. I prosjektet ble det bestemt at fartsmodellen skal være ensidig. Det betyr at modellen kun vil beregne reduserte hastigheter og bruke skiltet fartsgrense der geometrien ikke er utfordrende. Et nøkkelargument for dette valget er at vi ikke ønsker å planlegge for at sjåfører skal bryte fartsgrensen. Rent teknisk er det enklest å fjerne dette valget, men da må man gjøre en større supplerende studie av fartsvalg på veger med god geometri i Norge. Hovedfokus i dette prosjektet har også vært at vi ønsker å representere dårlig geometri i dagens vegnett bedre, slik at utbedringer i vegnettet kommer ut med mer realistiske endringer i reisetidsverdier.

2.1 Etablering av radius på et vilkårlig vegelement?

Den grunnleggende formelen for forholdet mellom radius, fart og friksjon krever at man finner en radius før man kan beregne hastigheten. I prosjektet «Fartsmodell for næringslivets transport» ble det etablert rutiner for å hente ut den radien som er lagt inn i NVDB. Å kode opp uttaksrutiner mot NVDB er tidkrevende og

¹ Hb V120 Statens vegvesen

krever spesialkompetanse i proprietært Application Programming Interface (API) som har en begrenset brukermasse utenfor Statens vegvesen. Et alternativ til å bruke SVV sitt proprietære API var å utvikle en løsning som baserer seg på mer åpne standarder. Når prosjektet startet var arbeidet med ny publikumsadgang til NVDB i gang. Løsningen benytter åpne standarder for interaksjon mot NVDB for eksterne². Den nye løsningen kunne ikke levere ut veggeometrier da den var rettighetsbelagt og hadde en restriktiv lisens. I prosjektet fikk vi overlevert et Elveg datasett som beskriver det norske vegnettet. Datasettet har et begrenset utvalg av fagdata, men har geometrier med Z-koordinater i alle knekkpunkter langs veglinjen. Dette innebærer at prosjektet fikk en svært detaljert beskrivelse av veggeometrien.



Figur 2: Illustrasjon av forholdet mellom degree of curvature og kurveradius

I amerikansk undervisningsmateriell innenfor vegutforming er «degree of curve» en kjent begrep. Dette kan beskrives som den *vinkelendringen man observerer etter å ha kjørt en fast distanse langs en kurve*. Vinkelendringen er proporsjonal med radiusen til kurven. Figur 2 viser sammenhengen mellom «degree of curvature» og radius i en kurve. Denne formelen og metodikken for å etablere radius langs en kurve kan operasjonaliseres i kode. I prosjektet ble Python valgt som programmeringsspråk da moduler for å håndtere dataformatet Elveg blir levert på er fritt tilgjengelig etter at ESRI frigav formatbeskrivelsen i 1998 (ESRI 1998). Koden deler et vilkårlig vegstykke inn i delsegmenter på 30.48 meter (100 fot) og beregner tangenten ved starten av delsegmentet og tangenten ved slutten av delsegmentet. Forskjellen mellom de to vinklene er «degree of curvature» og er direkte relatert til radien på delsegmentet.

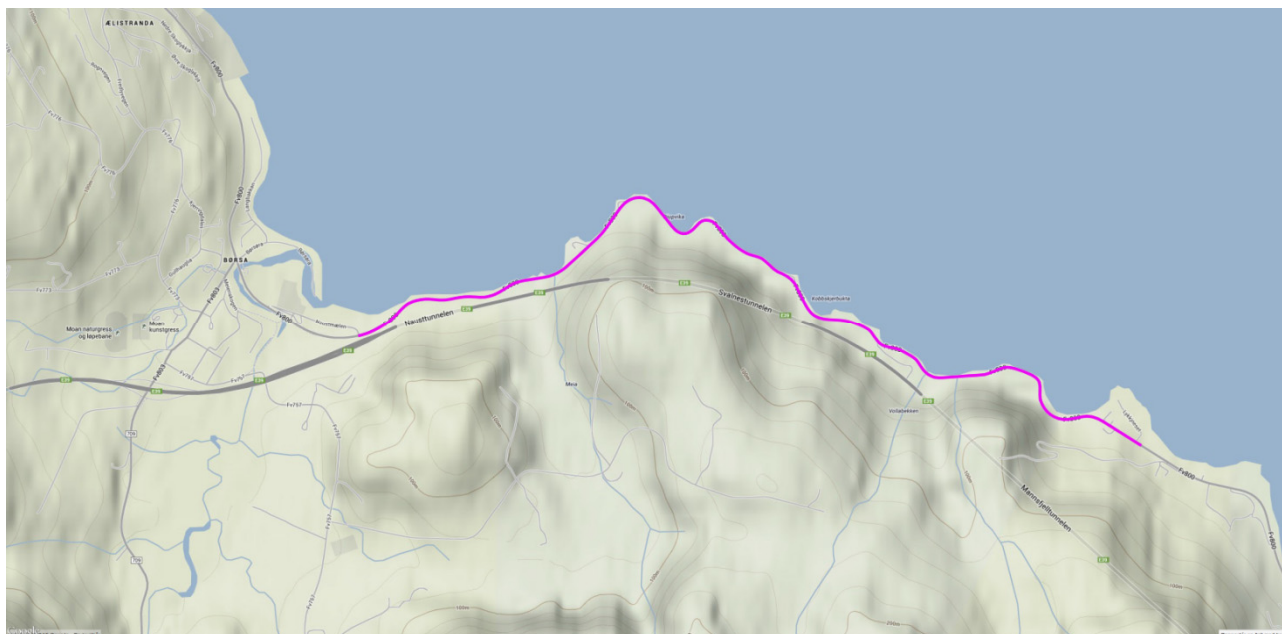
$$Radius = \frac{1746.38}{\Delta\alpha} \text{ (meter)}$$

Med de tilgjengelige data og metoder var det mulig å beregne seg fram til et sett med radier langs veglinjen. Det ideelle ville ha vært å ta utgangspunkt i de radier som er kodet inn i NVDB, men denne operasjonen er for kostbar å kode. Praktisk erfaring har vist at det er utfordrende å ta ut data fra NVDB for hele Norge. Beregningsmetodikken er testet og gir gode svar fra 15 meters radier og oppover. Nedre radius settes derfor til 15 meter.

Ved å bruke en metodikk som ikke tar høyde for den vanlige inndelingen av veglenker i sirkelbue, klotoider og rettstrekninger var det viktig å teste konsekvensen av forskjellige startpunkt på veglinja. Dvs. med forskjellige startpunkt for inndeling i delsegmenter. Det ble derfor valgt ut et vegelement på fylkesveg 800 der startpunkt for beregningen ble flyttet fra 0 til 30 meter. Gjennomsnittlig hastighet ble beregnet til 55,58 km/t med standard avvik på 0,31 som følge av de 30 forskyvningene. Figur 3 viser strekningen som ble benyttet til strekningstesten. Strekningen er på folkemunne kjent som «Børsabergan», har en svært

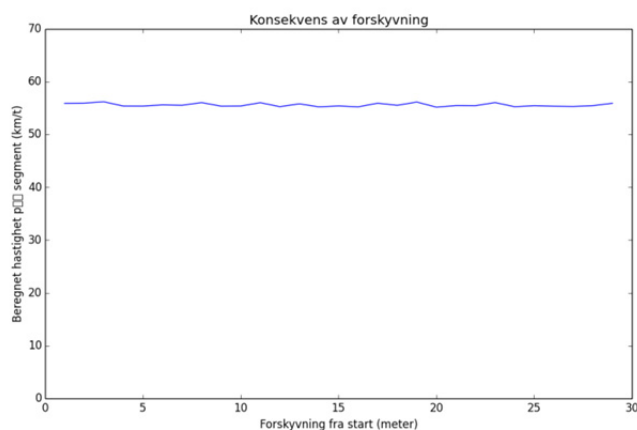
² <https://www.vegvesen.no/nvdb/api/dokumentasjon>

utfordrende veggeometri og er senere erstattet av ny E39 som går i tunnel innenfor. Strekningen er 3700 meter lang med skiltet fartsgrense på 60 km/t.



Figur 3: Strekning benyttet til forskyvningstest

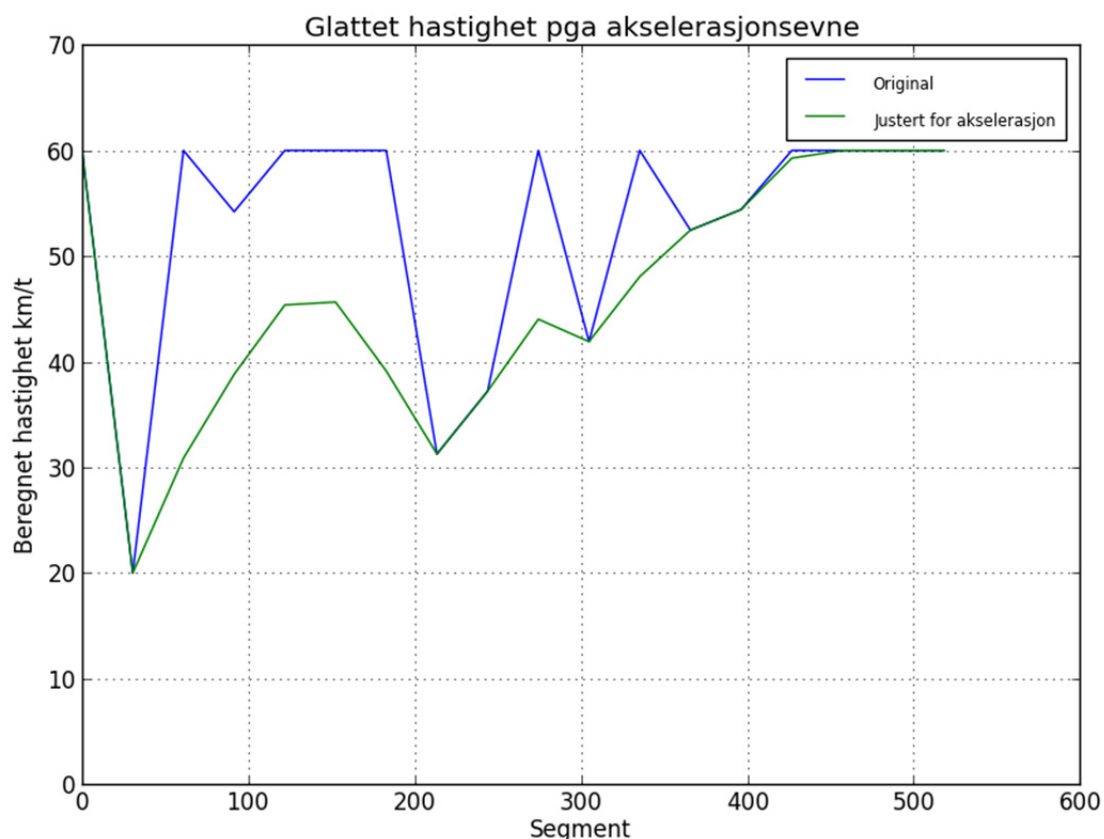
Figur 4 viser at konsekvensen av forskjøvet startpunkt er minimal og det er ingen systematisk endring som følge av at startpunktet blir flyttet. På bakgrunn av denne testen har vi konkludert med at radiusestimeringsteknikken er stabil og god nok som grunnlag for fartsberegninger. Plottet visere resultatet når en modell utviklet av Islam og Seneviratne i 1994 (en av flere modellforslag opplistet i Laurel and Sayed, 2005) legges til grunn. Til sammen 7 modeller ble testet.



Figur 4: Konsekvensen av forskjøvet startpunkt

En utfordring ved å beregne hastighet for hver enkelt kurve er at man får beregnet veldig forskjellig hastighet for to elementer som ligger ved siden av hverandre. Man må derfor ta hensyn til inngangshastigheten til hver enkelt kurve og muligheten for akselerasjon, ellers kan hastighetssprangene bli store i forhold til normale

kjøretøy. For å ta høyde for dette i beregningen ble det laget en akselerasjonsmodell som sørger for at akselerasjonsevnen holdes på et normalt nivå. Figur 5 viser hvordan akselerasjonsmodulen fungerer i praksis på en 540 meter lang veglenke med svært utfordrende veggeometri. Den blå linjen viser hastigheter beregnet av fartsmodellen, mens den grønne linjen er justert for et akselerasjonsnivå på 1 m/s^2 . Akselerasjon på 1 m/s^2 er regnet som en relativt kraftig akselerasjon.



Figur 5: Resultat av akselerasjonsmodul

2.2 Fartsmodeller for horisontal kurvatur

Ønsket var å finne en så enkel som mulig modell som benyttet et minimum av data om vegen og kjøretøyene. Som en del av prosjektet ble litteratur tatt fram i «Fartsmodell for næringslivets transporter» vurdert (Sakshaug and Toftegård 2011). I tillegg ble en nyere rapport fra Transportation Research Board gjennomgått (Dimaiuta, Donnell et al. 2011). Fartsmodellen i Effekt ble også vurdert, men denne ble valgt bort da den krever flere inndata deriblant data om vegens bredde for beregning av basisfart, før korreksjon for horisontal og vertikalkurvatur gjennomføres. Vegbredde er en parameter som krever programmering mot NVDB sitt API. Vi ble derfor sittende igjen med 7 fartsmodeller som kun bruker radius som inngangsdata.

- Fransk 2 felt
- Islam & Seneviratne 1994
- Kanellaidis Et.al.1990
- Lam et.al 1999
- Lamm & Choueiri 1986
- Ottesen & Krammes 2000
- Sveitsisk basis

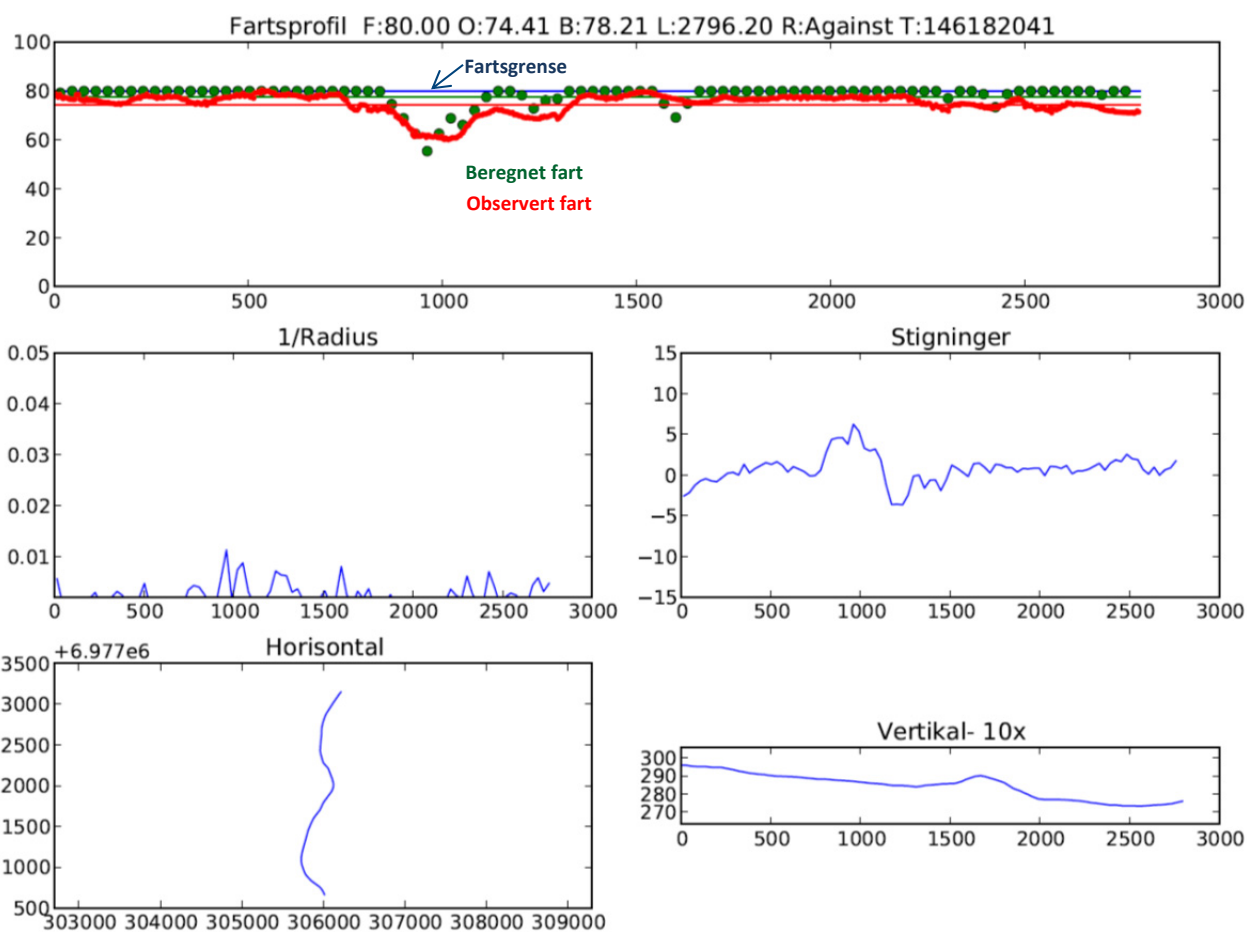
Det er stor forskjell i antall kurver og spredninger i kurveradier som er brukt til å estimere modellene over. Å gjøre antagelser om modellenes overførbarhet til norske forhold basert på beskrivelsen av modellene lar seg ikke gjøre innenfor rammen av dette prosjektet. Det ble gjennomført en test ved å kode opp alle modellene og teste dem på norske veier. I tillegg ble det gjennomført et forsøk der to studenter som hadde sommerjobb hos Statens vegvesen Region midt kjørte spesielle ruter med loggeutstyr fra SINTEF. De ble instruert til å kjøre under fartsgrensen og ha en normal kjøreadferd med tanke på akselerasjoner og retardasjoner. Data fra disse kjøringene ble senere brukt til å vurdere hvilken modell som synes å være best egnet for veier med utfordrende kurvatur.

2.3 Fartsmodell for stigninger og fall

På hvert delsegment beregnes en gradient for stigning eller fall. Blant fartsmodellene fra litteratursøket tok få hensyn til både vertikal og horisontal geometri. Det ble gjort tester av noen modeller som brukte både vertikal og horisontal kurvatur, men disse stammet stort sett fra USA og ga ulogiske svar ved radier mindre enn 250 meter; de indikerte at farten skulle øke. Dette skyldes trolig at modellene ikke var estimert for radier under 250 meter. Vi valgte derfor bort disse modellene og så på andre mulige løsninger for hvordan horisontal og vertikal kurvatur kan tas inn. Det ble valgt å benytte seg av separate vertikal- og horisontalmodeller. For å finne resulterende hastighet av de to modellene velges resultatet fra modellen som beregner lavest hastighet. Relativt få vertikalmodeller med inngangsdata var tilgjengelige, og den som syntes å passe best var en fransk modell for 2 feltveier (Dimaiuta, Donnell et al. 2011).

2.4 Testing av modellene

5Hz GPS data samlet fra studentene ble koblet til vegnettet og sammenliknet med modellberegninger. For å få en bedre følelse med hvordan hver enkelt modell fungerte ble 256 spesielt utfordrende veglenker plukket ut. Ingen av lenkene hadde kryss eller andre egenskaper som skulle føre til redusert hastighet. Noen av lenkene har blitt kjørt flere ganger, noe som betyr at det totale datasettet har 328 lenker der vi kan sammenlikne observert hastighet og beregnet hastighet. Figur 6 viser et diagnoseplott for en av de 328 veglenkene. Diagnoseplottet består av 5 grafer. Den øverste grafen viser fartsgrense (blå strek), beregnet hastighet (grønne prikker) og observert hastighet (røde prikker). Y-aksen er hastigheten og x-aksen viser avstand fra der kjøretøyet entret veglenken. Den heltrukne grønne linjen viser beregnet gjennomsnittshastighet på lenken, og den røde heltrukne linjen viser gjennomsnittshastigheten for det kjøretøyet som ga observert fart. På figuren kan man se at det er delvis sammenfall mellom beregnet kjørehastighet og observert kjørehastighet. I de påfølgende plottene vises kurvatur som $1/\text{radius}$, som uttrykker krumningen på kurven. Høyere verdier betyr en krappere kurve. Grafen til høyre, stigninger, viser gradienten langs vegen i retningen som turen ble kjørt. De to siste grafene viser horisontal- og vertikalplott av strekningen som er analysert. I figuren benevnt *horisontal* ser vi en projeksjon av horisontalkurvaturen der verdiene på aksene er koordinater i UTM 33 Nord. Figuren som er benevnt *vertikal* viser vertikalprofilen for strekningen i den retningen strekningen ble digitalisert når den ble lagt inn i NVDB. I det konkrete plottet har kjøretøyet kjørt i motsatt retning, noe som vises av overskriften på figuren R: *Against*. Det vil si at man må snu metringene for å få den til å stemme med stigningsplottet over som er tegnet i kjørt retning. Overskriften på figuren gir også informasjon om observert og beregnet strekningshastighet (O, B). L beskriver lengden på strekningen og T beskriver en unik koblingsnøkkel mot NVDB.



Figur 6: Diagnoseplott for en enkelt strekning

Plottene ble brukt til to formål:

- i) Til å luke ut strekninger der kjøretøyet hadde gjort et stopp (sjåfør har sannsynligvis kjørt til side og tatt en pause) eller at bilen kom inn i en veiarbeidssone
- ii) Til å bedre forstå fartsendringer langs forskjellige strekninger, da vertikal og horisontal kurvatur samt observert fartsprofil er sammenstilt på en og samme figur

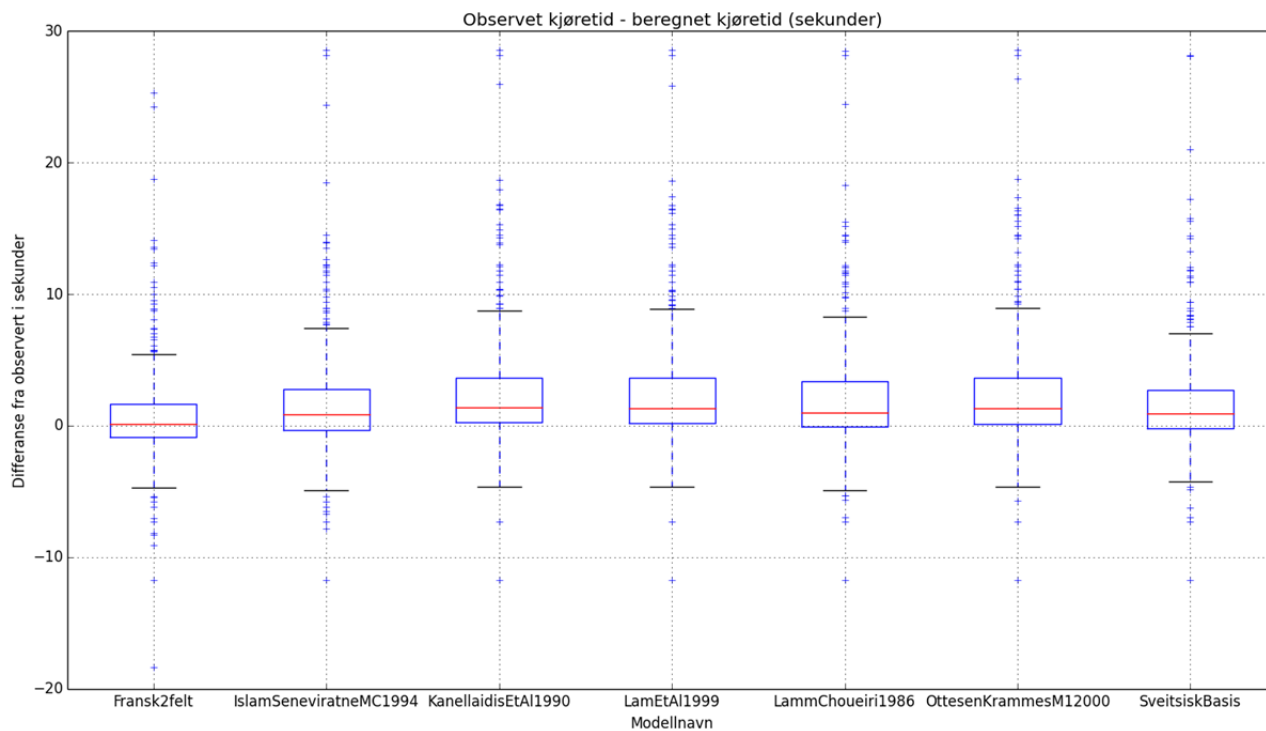
Kombinasjonen av lenken som ble benyttet til beregningene har en total lengde på 238 km. Det tok 3 timer og 43 minutter å kjøre denne turen. Ulike modeller ga forskjellig beregnet kjøretid som vist i Tabell 1.

Tabell 1: Kjøretid langs en teststrekning, observert og beregnet gitt ulike modeller

Kjøretid	Kjøretid
Obsvert kjøretid	3 t 43 min
Fransk 2 felt	3 t 38 min
Islam & Seneviratne 1994	3 t 34 min
Sveitsisk Basis	3 t 32 min
Lamm & Choueiri 1986	3 t 31 min
Ottesen & Krammes 2000	3 t 28 min
Lam et.al. 1999	3 t 28 min
Kanellaidis et.al. 1990	3 t 28 min
Fartsgrense	3 t 23 min

I dette tilfellet er kun navnet på horisontalmodellene med, da alle er beregnet med en enkel fransk stigningsmodell. Det vi ser er at den observerte kjøretiden er 3 timer og 43 minutter. Mens dersom vi hadde brukt fartsgrensen som estimat på kjørehastighet ville kjøretiden ha vært 3 timer og 23 minutter. Dersom vi brukte fartsgrensen som estimat ville vi få 9 % for kort kjøretid på veger med moderat dårlig standard. Strekningene er hentet fra fylkesveg 800 (Børsabergan), riksveg 30 (Røros – Støren), fylkesveg 705 (Stjørdal – Brekken) og 715 og 717 (Fosen). Alle vegene har fast dekke.

En inspeksjon av enkeltavvikene på hver lenke bekrefter at den franske 2-feltsmodellen synes å gjenspeile fartsvalet på teststrekningene best.

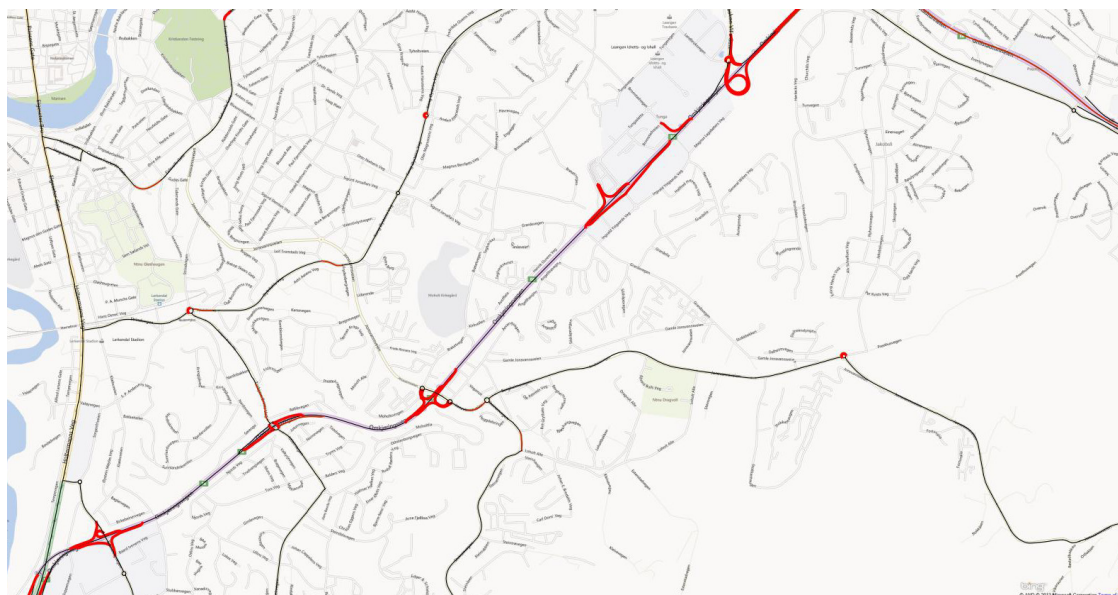

Figur 7: Boxplott av differanser mellom beregnet og observert reisetid

Figur 7 viser et boxplott av differansen mellom beregnet kjøretid og observert kjøretid på enkelt strekninger. Denne figuren bekrefter også at den franske 2-feltsmodellen er godt egnet til å beskrive kjørehastighet på

veglenker med utfordrende veggeometri. For lesbarhetens skyld er 2 observasjoner utelatt fra alle modellene: én der differansen var rundt + 60 sekunder og én der differansen var – 30 sekunder. Begge analyseplottene tyder på at sjåføren har ikke sett opp eller nedskilting av hastighet eller at det er kodet feil hastighet i NVDB. Forskjellene i beregning er lik for alle modellene, så avvikene skyldes ikke modellberegningene.

2.5 Visuelle inspeksjoner av hvilke vegger som får reduksjoner i hastigheter

En annen måte å vurdere effekten av fartsmodellen på er å bruke geografiske informasjonssystem til å analysere hvilke veglenker som får reduserte hastigheter. Ramper på og av motorveger er kjente områder der man ikke klarer å gjenskape fornuftige fartsreduksjoner. Geometriske fartsmodeller kan imidlertid bidra til å skape mer realistiske kjørehastigheter. På Figur 8 er lenker med en fartsreduksjon større enn 5 kilometer i timen kodet med rødt. Kartutsnittet er hentet fra Trondheim og viser Omkjøringsvegen. Figuren viser tydelig hvordan rampene i vegnettet får en fartsreduksjon.



Figur 8: Eksempler på veglenker hvor friflytfarten er lavere enn fartsgrensen ifølge fartsmodellen

Små rundkjøringer kan være et problem, fordi lenkelengdene inni rundkjøringene er for korte til at metodikken for å finne geometriske forsinkelser fungerer. Men sirkulasjonsarealet i rundkjøringer kodes med egne hovedparsell nummer. Lenkene i sirkulasjonsarealet kan derfor tilordnes en redusert hastighet basert på at de blir funnet ut fra vegidenten.

2.6 Oppsummering fart som følge av veggeometri

Som en del av prosjektet ble det funnet fram til 7 fartsmodeller for horisontal kurvatur og én modell for vertikal kurvatur. Alle modellene ble implementert og testet på forskjellige radier som man forventer å møte på vegger med utfordrende geometri. I tillegg ble en vertikal fartsmodell implementert. For å vurdere egnetheten til modellene ble modellenes beregningsresultater sammenliknet med observasjoner fra et kjøretøy på over 250 forskjellige veglenker. Resultatene viser at bruk av skiltet hastighet gir ca. 9 % for lave reisetider, mens bruk av den franske 2-feltsmodellen gir ca. 2 % for lave reisetider. Bruk av den franske 2-feltsmodellen vil gi en klar forbedring på tidsestimatene på vegger med utfordrende veggeometri.

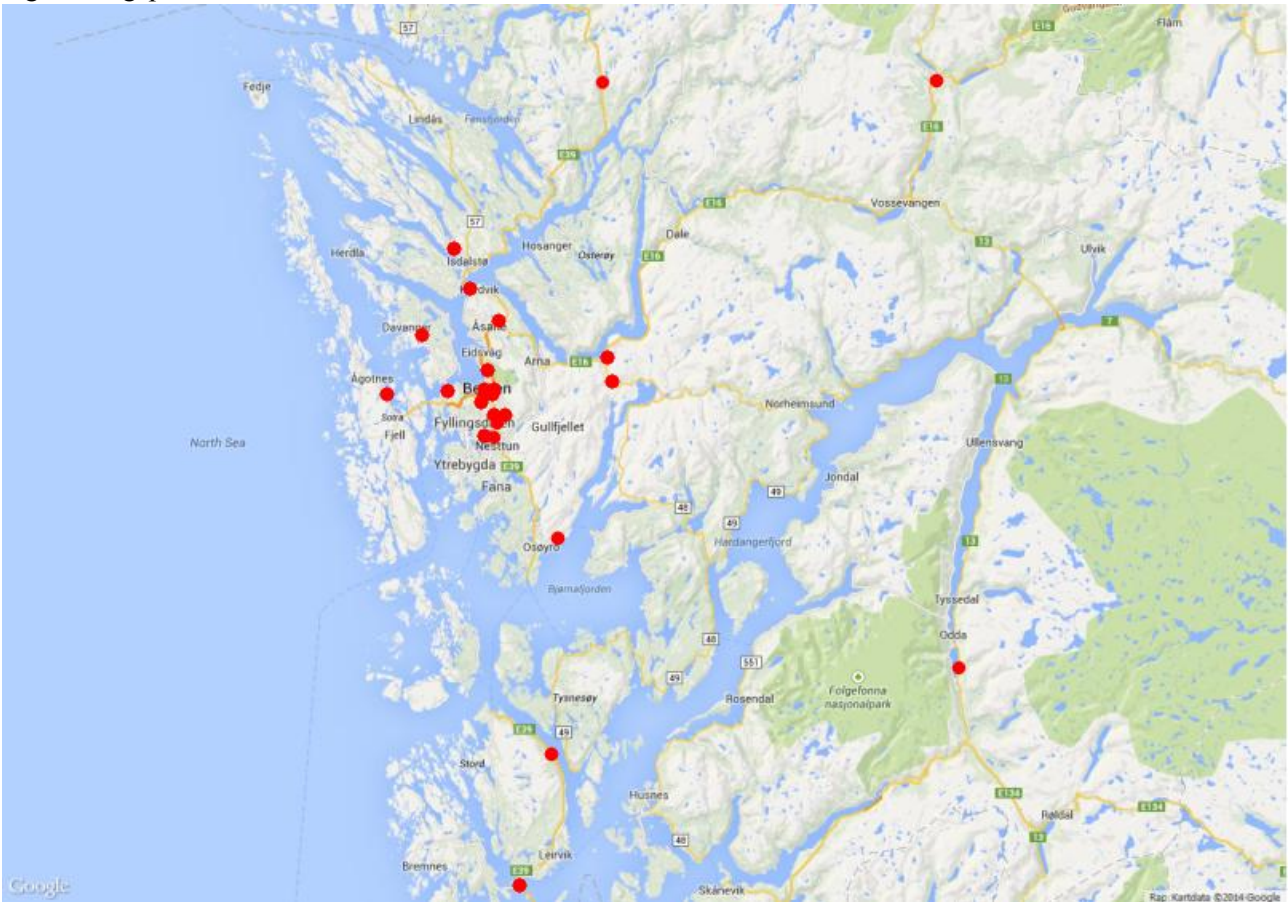
3 Forsinkelse på strekning som følge av trafikk

Også på rette strekninger som har god horisontal- og vertikalutforming vil det kunne oppstå forsinkelser. Disse forsinkelsene er primært knyttet til interaksjonen mellom kjøretøyene på en aktuell strekning. En vanlig måte å inkludere disse forsinkelsene på er å bruke volumhastighetskurver. Som navnet tilsier beskriver volumhastighetskurver reduksjoner i hastighet som følge av økt trafikkvolum. I denne deloppgaven ble det undersøkt hvorvidt Statens vegvesen har data som kan brukes til å estimere volumhastighetskurver. I tillegg ble det studert om det var mulig å benytte seg av kompetanse hos modellbrukere som har kalibrert modellene sine for å få mer realistiske kjøretider for bil.

Målet for denne oppgaven var å vurdere eksisterende kurver og skaffe et kunnskapsgrunnlag dersom det skal estimeres nye volumhastighetskurver basert på faktiske data fra Statens vegvesens tellepunkter. Oppgaven skulle også vurdere om det var mulig å benytte maskinlæringsteknikker for å hjelpe brukere å velge riktige volumhastighetskurver.

3.1 Datasett og metode

Første deloppgave var å få tak i datasett som inneholder observasjoner for enkeltkjøretøy. Mye av utstyret som brukes i de kontinuerlige tellepunktene kan lagre data om enkeltkjøretøy, men utstyret må settes opp for denne oppgaven da dette ikke er en standard datatype som Statens vegvesen lager. Etter et søk i Statens vegvesen fikk vi overlevert data for 51 tellepunkter i Hordaland, se Figur 9 for lokaliseringen av registreringspunkter.



Figur 9: Oversikt over tellepunkter som har lagret enkeltkjøretøydata i Hordaland

Overlevert datasett bestod av 1487 filer med totalt 59,6 millioner rader med data. Hver rad inneholder data om kjøretøyets lengde, fart og passeringstidspunkt. Data ble lastet inn i en PostgreSQL database og det ble

laget databaserutiner for å aggregere data til 5 minutters nivå. Etter aggregering har man data for 5 minutters intervaller med en oversikt over volum og hastigheter for de aktuelle tellepunktene fordelt på retning og felt. I datasettet er registreringspunktet i Fløyfjellstunnelen det desidert største, med 31 millioner observasjoner.

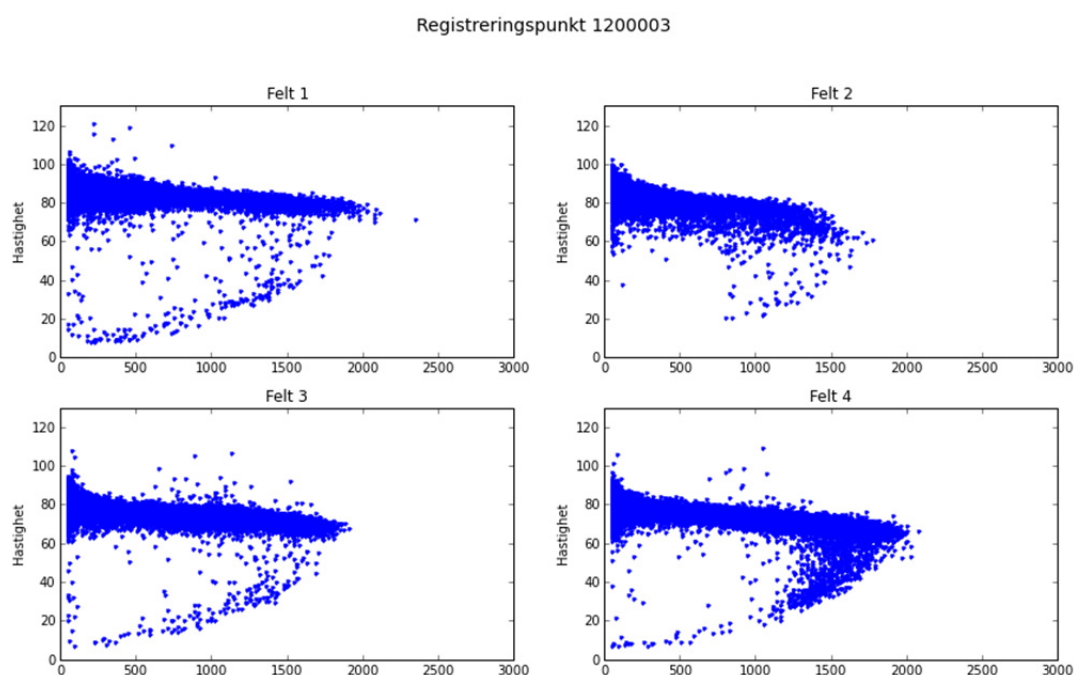
I tillegg til databasefunksjonen ble det utviklet metoder for å plote data fra observasjonene og kurvene som ligger inne i de regionale transportmodellene. I tillegg ble rutiner for kurvetilpasning etablert for å kunne tilpasse vilkårlige kurver til observasjonene.

En stor svakhet med datasettet er at over 50 % av datamaterialet stammer fra ett registreringspunkt, i tillegg til at alle data er fra ett fylke. For å kunne estimere volumhastighetskurver vil det være nødvendig med mer data fra et større spekter av tellepunkter. Statens vegvesen er også i ferd med å endre rutine for hvordan registreringsdata lagres, noe som vil gjøre at man i fremtiden vil kunne få enkeltkjøretøyobservasjoner fra kontinuerlige registreringspunkter. På bakgrunn av dette ble datamengden og variasjonen i datamaterialet for liten til at det var fornuftig å gjennomføre en større estimering av volumhastighetskurver. Datamaterialet ble derfor prosessert for å gi et grunnlag til å vurdere avvikling på feltnivå. Plott av volum hastighetsdata er lagt ved som vedlegg i denne rapporten.

Det andre ønsket i prosjektet var å lære av hvordan erfarne brukere av modellen har kalibrert sine modeller på lenkenivå. Maskinlæringsrutinen kaldt K-Nearest Neighbor (KNN), skulle benyttes til å lære datamaskinen til hvordan den erfarne modellbrukeren prioriterer/tenkte ved valg av data i som skal inngå i modellen. Ved å sammenligne datatyper fra vegdatabanken (via det nye åpne grensesnittet) med de valg den erfarne har tatt ved kalibrering av volumhastighetskurver i sitt prosjektområde ønsket en å gi mer uerfarne brukere et bedre utgangspunkt for valg i sine modeller. Fotgjengerfelt, vikeplikt og parkeringsrestriksjoner er eksempler på datatyper det var aktuelle å se nærmere på. Denne delen av prosjektet ble ikke gjennomført.

3.2 Resultater

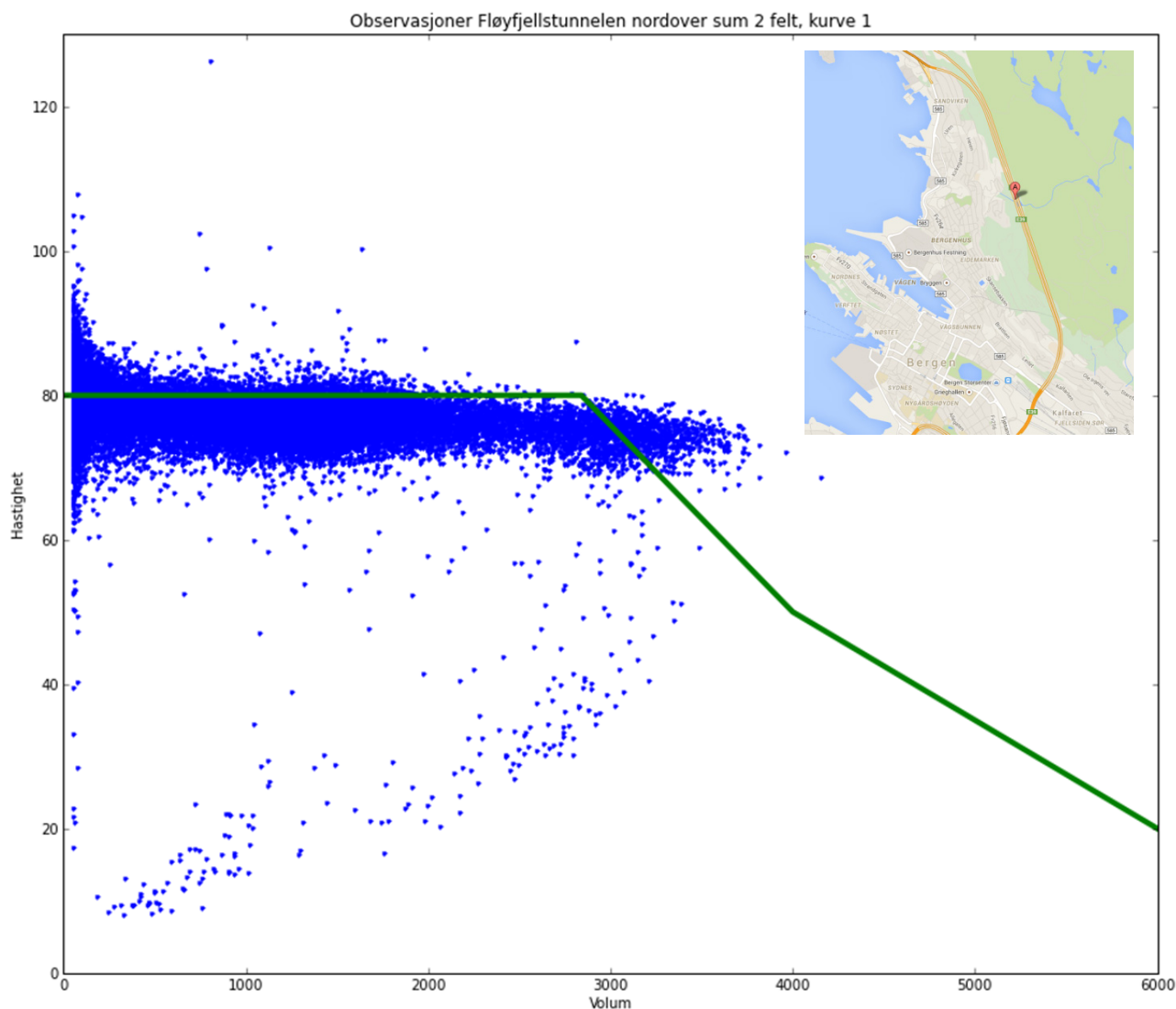
Å nå målet for denne deloppgaven ble vanskelig da datasettet var for begrenset. Men metodikk for å kunne estimere nye kurver for volum hastighetskurver ble utviklet. I samråd med oppdragsgiver ble fokus flyttet mot å generere et kunnskapsgrunnlag for de data som var tilgjengelig. For 38 registreringspunkter ble det produsert volumhastighetsplott fordelt på felt, disse plottene er vedlagt som appendix. Kun i 6 av de 38 tellepunktene ser vi plott som minner om klassiske sammenbruddssituasjoner. Figur 10 viser et plott der man ser en klassisk sammenbruddssituasjon, felt 1 og 3 er Nordover i tunnelen fra Bergen sentrum, felt 2 og 4 er inn mot Bergen sentrum.



Figur 10: Tellepunkt 1200003 Fløyfjellstunnelen

I den regionale transportmodellen kodes ikke feltene separat, men som en sum av felt i samme retning. Figur 11 viser volum/hastighetsplott for felt 1 og 3 summert og den grønne streken viser kurven som i RTM representerer denne volum hastighetskurven.

Diskusjoner rundt hvordan modellbrukere velger volumhastighetskurver viste at de stort sett bruker metodikken som ligger inne i eksisterende transportmodell der kurver velges automatisk. Det var få som justerte på formen på kurvene. Den mest vanlige justeringen var å endre på hastigheten for halen på volum hastighetskurvene (det siste innslagspunktet). Det var også vanlig å redusere friflythastigheten (skiltet hastighet) for å få en mer troverdig kjørehastighet ved lav trafikkbelastning (behandlet i forrige kapittel). Disse kalibreringsteknikkene gav ikke grunnlag for å kjøre maskinlæring på det arbeidet som var gjort. Det ble derfor ikke utviklet ny metodikk for å hjelpe brukere til å velge riktig kurve.



Figur 11: Fløyfjellstunnelen felt 1 og 3 Nordover fra Bergen sentrum. Kart fra Google.

På nåværende tidspunkt var det for lite spredning i datakilder i eksisterende datamateriale til at prosjektet kunne levere grunnlag til å endre VDF kurvene som benyttes i dagens transportmodeller. Det er forventet at dette vil endre seg når det nye systemet for lagring av trafikkdata kommer på plass. Da vil de fleste tellepunkt rapportere på enkeltkjøretøynivå slik at vi kan få den nødvendige spredningen over vegkategorier og vegfunksjoner. I påvente av nødvendig grunnlag til å endre VDF kurvene i dagens transportmodell er det lagt til en kodingshjelp i TNext, som hjelper bruker med valg av tilgjengelig VDF-kurve basert på hastighet og antall felt.

4 Registrerte hastigheter gjennom kryss

I kryss har vi to hovedårsaker til forsinkelser: trafikk og geometrisk utforming av krysset. I dette kapittelet dokumenteres arbeidet som ble gjennomført i prosjektet knyttet til registrert forsinkelse gjennom kryss, basert på et datasett generert av ISA-enheter³ utplassert i Statens vegvesens tjenestebiler. Datasettet består av litt over 220 millioner observasjoner og gir posisjon for kjøretøyene hvert sekund. Ved å bruke data fra typiske lavtrafikkperioder, vil vi kunne få reisetider gjennom kryssene som hovedsaklig er påvirket av kryssets geometriske utforming. Fra dette datasettet er det utviklet tall for forsinkelsen gjennom kryss knyttet til geometrisk utforming sammenlignet med å holde en fart tilsvarende fartsgrensen.

I og med at datasettet inneholder reisetider gjennom kryss også i rushtidene, er det oppgitt forsinkelsesverdier for disse periodene. Rushtidstimmene er definert som perioden fra klokken 07.00-9.55 og 15.00-17.59. Forsinkelsesverdiene for rushtimene er oppgitt i denne rapporten, men de kan ikke uten videre representere forsinkelser gjennom kryss i rushperioder i transportmodellene. Årsaken er at reisetid gjennom kryss vil være veldig avhengig av trafikkvolumet fordelt på de ulike armene i krysset. Slik informasjon var ikke tilgjengelig i datagrunnlaget.

4.1 Datasett og metode

Å håndtere et datasett med 220 millioner observasjoner har noen datatekniske utfordringer som gjorde det nødvendig å redusere datamengden tidlig i analysen. Derfor ble det først kodet opp en rekke kryssområder der det ble hentet ut data. Dette reduserte datamengden betydelig, og man satt i praksis igjen med fartsobservasjoner i en radius på 200 meter rundt de utvalgte kryssene. Det ble plukket ut kryss i Trondheim, Bergen, Drammen, Lillehammer og Bodø. Totalt inkluderte datasettet 130 forskjellige kryss. Det ble skrevet skript (Python) for å analysere kjøreturer gjennom kryss, og totalt 8141 turer gjennom kryss ble funnet. I datasettet er det noen turer der mannskapet fra Statens vegvesen har stoppet inni kryssområdene, noe som har gitt forsinkelser på over 120 sekunder. Disse dataene ble ekskludert, fordi vi har grunn til å tro at dette er tjenestebiler som har stoppet i kryssområdet for sjekk av styreskap eller vask av lyshoder. Etter kvalitetskontroll av datasettet satt vi igjen med 7114 turer med ulike bevegelser gjennom kryss, og disse er benyttet til analysene.

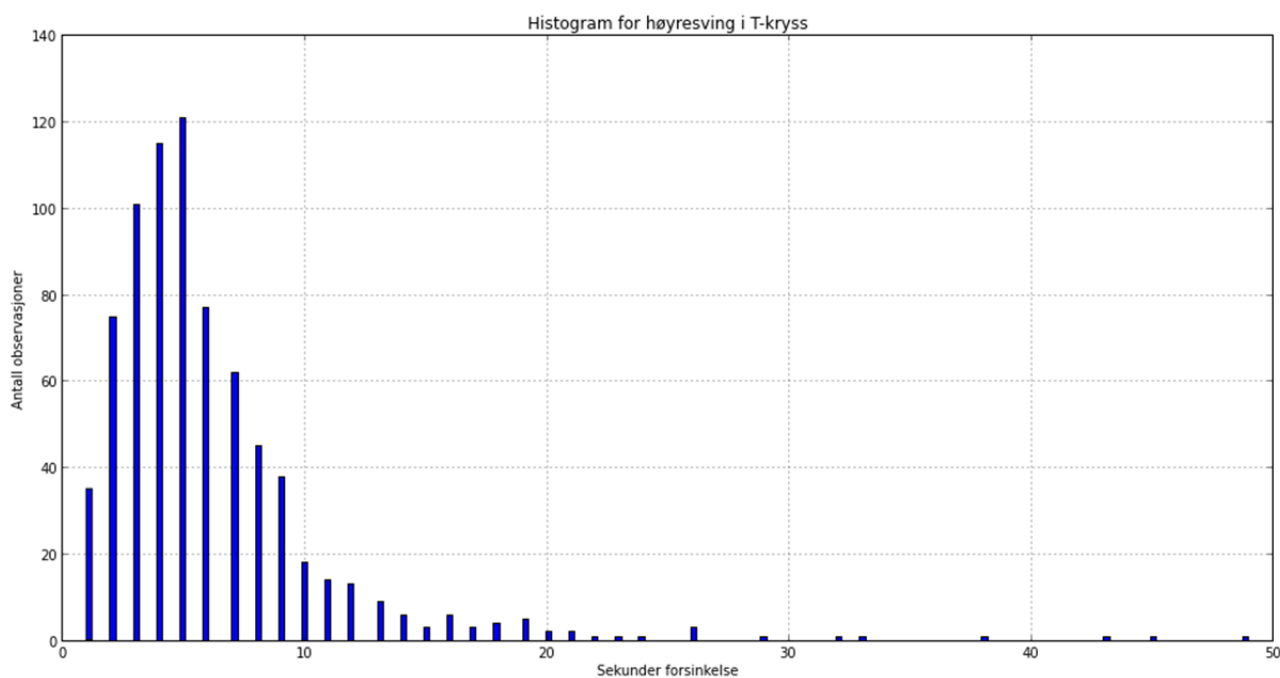
For å beregne forsinkelse gjennom kryssene ble det manuelt etablert soner som beskrev når kjøretøy ankommer og forlater et kryss. Disse sonene ble kodet med unike IDer som gjorde det mulig å programmatisk finne ut om kjøretøy kjørte rett fram, til høyre eller til venstre i krysset. Så ble tiden mellom ankomststed og avgangssted beregnet og sammenholdt med tiden det ville ta å kjøre gjennom krysset ved skiltet hastighet. Differansen mellom disse to verdiene er definert som forsinkelsen gjennom krysset. For den delen av datasettet som er fra lavtrafikkperiodene, forutsetter vi at de fleste kjøretøyene kan kjøre uhindret gjennom krysset og dette gir den geometriske forsinkelsen. For rushtidsperiodene forutsetter vi at forsinkelsen kan forklares av interaksjon med andre trafikanter som kommer i tillegg til den geometriske forsinkelsen.

4.2 Valg av statistisk parameter lavtrafikk

Datamaterialet inkluderer ikke trafikkvolum, derfor har vi ikke kontroll med om bevegelsene gjennom kryssene var hindret. Vi har benyttet tidsperioder utenfor rush for å unngå tider med rushtrafikk, men bruk av gjennomsnittsverdier tilsier at det trolig er noen trafikale hindringer som skaper forsinkelser i våre observasjoner. Vi ser også at datasettet inneholder enkelte høye verdier som gir betydelig høyere gjennomsnittsverdier i forhold til medianverdien. Medianverdien synes å være høy, ettersom denne også vil inkludere både geometrisk forsinkelse og trafikal forsinkelse. Tanken med estimeringen av kryssforsinkelse var å få fram den forsinkelsen som er relatert til kurvaturen og som følge av risikooppfattelse. Etter å ha sett

³ ISA = Intelligent Speed Adaption

på histogram for forskjellige svingebevegelser gjennom kryss kom vi fram til at vi måtte finne en robust statistisk parameter for å uttrykke denne forsinkelsen. NTNU ved Trude Tørset gjennomførte også et sett med kontrollerte forsøk på høyre- og venstresving i T-kryss, 34 observasjoner fordelt på 3 kryss og høyre og venstre sving. Disse forsøkene indikerte at man skal se på nedre del av datasettet, men ikke de nedre ekstremverdiene. Figur 12 viser som et eksempel forsinkelse for en høyresving gjennom T-kryss. Her ser vi at det er ganske mange observasjoner med høye forsinkelsesverdier, mens medianen ligger på rundt 5 sekunder. På bakgrunn av dette og det kontrollerte eksperimentet ble nedre 25 % av datasettet valgt som måltall for geometrisk fartsreduksjon.



Figur 12: Histogram over forsinkelser for høyresving i T-kryss

4.3 Valg av statistisk parameter rushtrafikk

I rushperiodene vil trafikkstrømmene og interaksjonene mellom trafikanter bestemme størrelsen på forsinkelsene. Her er det valgt et relativt langt tidsrom som skal representere rushtrafikk, mye lenger enn vi vanligvis regner som rushperioder i de byene som inngår i datasettet. Vi har også bare ett sett av forsinkelsesverdier, selv om rushtrafikken vanligvis er svært retningskjev. Det betyr at selv om vi benytter en middelvei for å representere forsinkelser gjennom kryss, så kan faktiske reisetider gjennom kryss være mye høyere enn det vi finner her. Gjennomsnittsverdiene ble i stor grad overstyrt av ekstremverdier, og disse kan skyldes nettverkseffekter og hendelser som vi ikke har kontroll på. Det ble derfor valgt å benytte medianverdien for forsinkelsene i rushtidstimerne.

4.4 Registrerte forsinkelse i ulike krysstyper

4.4.1 Forsinkelser i lavtrafikk

Man må skille på bevegelser som har vikeplikt og de som ikke trenger å vike for annen trafikk. Hvis vi inkluderer alle observasjoner uten tanke på vikepliktsforhold, får vi forsinkelsesverdier som vist i Tabell 2. Dersom man kan skille mellom bevegelser med og uten vikeplikt, vil verdiene for T-kryss og X-kryss bli som i Tabell 3.

Tabell 2 Forsinkelse i sekunder gjennom kryss uavhengig av vikepliktsforhold i lavtrafikk (grått felt = anbefalt brukt)

Kryss type	Svingebevegelse	Gjennomsnitt	Median	25 % nedre	Antall observasjoner
Lyskryss	Høyre	13,5	12	5	37
	Rett fram	15,7	12	4	267
	Venstre	20,3	17,5	7	34
Rundkjøring	Høyre	7,7	6	4	559
	Rett fram	7,9	6	5	463
	Venstre	13,0	11	8	552
T-kryss	Høyre	6,1	5	3	767
	Rett fram	4,1	2	1	540
	Venstre	8,9	7	5	762
X-kryss	Høyre	7,3	5	4	87
	Rett fram	5,0	4	1	226
	Venstre	8,4	6	4	79

I utgangspunktet har X-kryss som er lysregulert samme geometriske avbøyningsforsinkelse som vanlig X-kryss, men de skiftende fasene fører til at man også får forsinkelser selv i lavtrafikkperioder som følge av faseskiftene. Lyskryss ble plukket ut fra datamaterialet for å se om det var større tidsbruk gjennom lyskryss. Økt forsinkelse i lyskryss er veldig synlig for rett fram bevegelsen, 4 sekunder i nedre 25 % kvartil for lyskryss mot 1 sekund i nedre 25 % kvartil for X-kryss.

Tabell 3: Forsinkelse i sekunder gjennom kryss i lavtrafikk avhengig av vikepliktsforhold (grått felt = anbefalt brukt)

Kryss type	Svingebevegelse	Vikepliktig	Gjennomsnitt	Median	25 % nedre	Antall observasjoner
T-kryss	Høyre	Nei	5,5	5	3	631
		Ja	8,9	8	6	136
	Rett fram Venstre	Nei	4,1	2	1	540
		Nei	8,6	7	5	693
		Ja	11,5	9	6	69
X-kryss	Høyre	Nei	6,3	4,5	3	68
		Ja	11,0	8	7	19
	Rett fram Venstre	Nei	5,0	4	1	226
		Nei	7,6	6	4	73
		Ja	17,6	15,5	11,3	6

4.4.2 Forsinkelser i rushtidstimene

Tabell 4: Forsinkelse i sekunder gjennom kryss uavhengig av vikepliktsforhold i rushtid (grått felt = anbefalt brukt)

Kryss type	Svingebevegelse	Gjennomsnitt	Median	25 % nedre	Antall observasjoner
Lyskryss	Høyre	18,0	17	6,5	15
	Rett fram	18,9	10	4	139
	Venstre	30,6	23	13	19
Rundkjøring	Høyre	8,7	7	4	452
	Rett fram	9,5	7,5	5	280
	Venstre	18,0	14	9	270
T-kryss	Høyre	7,0	5	3	490
	Rett fram	4,4	3	1	363
	Venstre	9,4	7	5	509
X-kryss	Høyre	8,2	4	4	60
	Rett fram	7,0	6	2	138
	Venstre	11,5	8	5	36

Tabell 5: Forsinkelse i sekunder gjennom kryss i rushtidstimene avhengig av vikepliktsforhold (grått felt = anbefalt brukt)

Kryss type	Svingebevegelse	Vikepliktig	Gjennomsnitt	Median	25 % nedre	Antall observasjoner	
T-kryss	Høyre	Nei	6,7	5	3	417	
		Ja	9,1	8	5	73	
	Rett fram	Nei	4,4	3	1	363	
		Venstre	Nei	9,0	7	5	478
			Ja	16,1	11	7,5	31
X-kryss	Høyre	Nei	7,5	5	4	40	
		Ja	9,7	7,5	5,75	20	
	Rett fram	Nei	7,0	4	2	138	
		Venstre	Nei	11,5	8	5	35
			Ja	8	8	8	1

4.5 Tilpasning til bruk

For å kunne legge inn forsinkelser i transportmodellene må det kodes inn krysstype i modellgrunnlaget. Grunnkrysstypene er X-kryss og T-kryss, men vi har to undervarianter av disse: lyskryss som variant av X-kryss og rundkjøring som en variant av T-kryss.

Det er betydelige forskjeller mellom tilfarter som har vikeplikt eller ikke. Det er derfor anbefalt at det kodes inn vikeplikter på lenkene i modellen som går inn mot vikepliktsregulerte kryss. Det kan være noen utfordringer til en slik koding ettersom det ikke finnes noen entydige og komplette databaser over denne reguleringsformen. I analysefasen ble flyfoto fra flere tjenester benyttet (Google, norgeskart.no, Bing og Finn benyttet) til å visuelt identifisere vikelinjen. Vegkart fra Statens vegvesen (www.vegvesen.no/vegkart/vegkart/) ble benyttet til å finne skiltplater med vikeplikt på vegnett som Statens

vegvesen har ansvar for. I praksis betyr dette at det er behov for at brukeren som skal kode opp en transportmodell har en stor grad av lokalkunnskap for å gjøre denne jobben på en effektiv måte.

Kjøring gjennom rundkjøringer ble i utgangspunktet målt på samme måte som de andre krysstypene, mens man i transportmodellene koder rundkjøringene som påfølgende T-kryss med kort avstand. NTNUs testing viste at man kan forutsette at trafikantene holder ca. 20 km/time i sirkulasjonsarealet. For å finne forsinkelsesverdier må de registrerte verdiene beregnes i forhold til at farten er 20 km/t i sirkulasjonsarealet. Da ble observerte forsinkelsesverdier som vist i Tabell 6.

Tabell 6: Forsinkelse i sekunder gjennom rundkjøring observert av SVV's tjenestebiler

Svingebevegelse	Observert i Rundkjøring
Høyre	4
Rett fram	5
Venstre	8

Denne forsinkelsen må nå dekomponeres i verdier for rett fram bevegelsen eller høyresving i T-kryss slik at vi kommer fram til tilsvarende forsinkelsesverdier.

Tabell 7: Tidsforbruk i sekunder for rundkjøring i lavtrafikk

Bevegelse	Forsinkelse (sekunder)
Rett fram	1,5
Høyre	2,1

Tabell 8: Tidsforbruk i sekunder for rundkjøring i rushtrafikk

Bevegelse	Forsinkelse (sekunder)
Rett fram	2,8
Høyre	3,9

Beregnet forsinkelse gjennom kryss hvis vi ser bort fra kjøretid på sirkulasjonslenkene, vil bli som i Tabell 9.

Tabell 9: Forsinkelse i sekunder gjennom rundkjøring beregnet

Svingebevegelse	Observert se Tabell 6	Beregnet (s)
Høyre	4	$2 * \text{Høyre} = 2,1+2,1 = 4,2$
Rett fram	5	$\text{Høyre} + \text{rett fram} + \text{høyre} = 2,1+1,5+2,1 = 5,7$
Venstre	8	$\text{Høyre} + \text{rett fram} + \text{rett fram} + \text{høyre} = 7,2$

5 Implementering i TNext og i transportmodellen

5.1 Implementering av fartsmodell i TNext.

Fartsmodellen blir brukt i TNext forbindelse med eksport av transportlenker til CUBE. Det er kun veggeometrien som blir hensyntatt. Effekten av stigning og fall blir beregnet i h.h.t. fransk 2-feltsmodell. Effekten av horisontalkurvatur blir beregnet i h.h.t. Lam mfl (1999). Den laveste resulterende farten av disse brukes, og nabosegenter glettes v.hj av en maksimal akselerasjon på 1 m/s², begge veier. Følgende tilpasninger er gjort:

- Strekningslengde på 30,48 m brukes, men kortere lenker får hastighet lik fartsgrensen.
- Beregning av radius er basert på den aktuelle segmentlengden: $\text{Radius[m]} = \frac{\text{Segmentlengde[m]}}{\text{vinkelendring[radianer]}}$.
- Det er bare lenker som kan kjøres med buss eller bil som beregnes.
- I rundkjøringer settes farten alltid til 20 km/t.

5.2 Bruk av fartsmodelldata i Regionale Persontransportmodeller (RTM)

Geodatabasen som blir produsert av TNext-funksjonen "Eksport til Cube" inneholder et datafelt **FMFART** som angir gjennomsnittshastighet på en lenke beregnet med fartsmodellen. Ut fra denne hastigheten og lenkelengden blir det beregnet en tidsbruk på lenken. Hvis hastigheten fra fartsmodellen skulle være lik 0 blir hastigheten tilordnet kodet hastighet på lenken.

Gjennomsnittshastighet fra fartsmodellen blir benyttet til å beregne avstander og tidsbruk (LoS-data) mellom alle soner i modellområdet. Fartsmodellhastighet benyttes også som utgangshastighet i nettfordeling av bilturer. I en kapasitetsavhengig nettfordeling vil volum/hastighetskurven bli kuttet av ved fartsmodellhastighet slik at selv om en kurve har høyere utgangshastighet enn den som blir beregnet av fartsmodellen, vil det ikke være mulig å kjøre fortere enn fartsmodellhastigheten på lenken.

Forsinkelser gjennom kryss er lagt inn i transportmodellen slik som Tabell 10 viser. I transportmodellen defineres riktignok rushperioden for morgenen fra klokken 06.00 til 09.00.

Tabell 10: Forsinkelser for ulike krysstyper i transportmodellen (RTM)

Krysstype	Lavtrafikk					Rushtrafikk				
	Rett fram	Venstre	Høyre	Vikeplikt		Rett fram	Venstre	Høyre	Vikeplikt	
				Venstre	Høyre				Venstre	Høyre
T-kryss	1,0	5,0	3,0	6,0	6,0	3,0	7,0	5,0	11,0	8,0
X-kryss	1,0	4,0	3,0	11,3	7,0	4,0	8,0	5,0	8,0	7,5
Rundkjøring	1,5		2,1			2,8		3,9		
Lyskryss	4,0	7,0	5,0			10,0	23,0	17,0		

6 Videre arbeid

Det blir stadig mer data som kan anvendes til å videreutvikle metodikken i transportmodellene, og dette prosjektet har både avdekket mulig bruk av allerede tilgjengelige data, for eksempel bruk av data fra tjenestebilene til Statens vegvesen, og dessuten et databehov som kan dekkes uten store investeringer, slik som for eksempel data fra trafikkregistreringer på strekninger som blir tilgjengelig i løpet av 2015.

6.1 Geometrisk forsinkelse

Data om hastighet på eksisterende vegnett er viktig for at forbedringer i kurvatur og kjøreforhold ellers vurderes riktig i nyttekostnadsanalyser. I dette arbeidet ble hastighetsnivå basert på vegens horisontalkurvatur og vertikalkurvatur beregnet med fartsmodeller funnet via litteraturstudier. Hastighetene ble verifisert gjennom kjøring av enkeltstrekninger med utfordrende kurvatur. Data om vegens kurvatur kommer fra NVDB.

Behov utover dette er å utvikle en fartsmodell hvor det inngår flere forklaringsfaktorer. Dette kan for vanlige personbiler være dekketype, dekketilstand, sikt lengde og vegbredde. For vegbredde er sannsynligvis også bredden på vegskulder viktig ved smal og svingete veg. I en slik utvikling blir det viktig å samkjøre transportmodellen og TNMs fartsmodell med EFFEKTs beregning av fart slik at det er den samme fart som legges til grunn i hele analysen.

Ytterligere testing kunne også vært gjort for å verifisere fartsmodellen. Den kunne også vært utvidet til å inkludere mer dynamiske forhold knyttet til for eksempel vinterføre, regn og mørke.

Dette prosjektet har fokusert på fart for personbiler. Det er også behov for friflytfart for andre typer reisemidler, både for tungtrafikk og myke trafikanter. Fartsmodellen for næringslivets transporter utviklet en fartsmodell som kunne vært tilrettelagt for bruk i transportmodellen, både den regionale transportmodellen for persontransport, men også for godsmodellen. Fartsnivå for gående og syklende inngår foreløpig ikke i den regionale transportmodellen for persontransport. I stedet er det avstand som forklarer valg eller ikke valg av gange eller sykkel. Dette burde vært studert nærmere. Det virker naturlig at for eksempel kurvaturforhold og infrastruktur som tilrettelegger for syklende vil påvirke antall syklistene på en strekning.

Beregninger som ga et riktig estimat på friflythastighet i byområder ville vært et stort steg framover for å kunne beskrive fartsnivået der. En slik verdi ville vært nyttig til bruk i meso- og mikromodeller i tillegg til de strategiske transportmodellene. Designelementer som er viktig for friflythastigheten i byområder kan være fotgjengerkryssinger, fartsdumper, veginnsnevninger osv. Med en sterk økning i sykkeltrafikken vil også sannsynligvis infrastruktur for sykkeltrafikk og kunne virke inn på farten.

6.2 Trafikkbelastning og forsinkelser på strekninger

I dag bruker vi volumhastighetskurver for å bestemme fartsnivået gitt trafikkbelastningen på veglenkene. Det er satt i gang et arbeid med å hente ut flere data fra tellepunkter som kan brukes til å bestemme kurveforløpet for sammenhengen mellom volum og fart. Dette arbeidet hadde ved utgangen av 2014 ikke kommet så langt at det kunne benyttes enda, men dette er data som bør kunne tas i bruk i løpet av 2015.

Med et større datasett samlet inn fra tellepunkter over hele landet, bør det også være mulig å vurdere volumhastighetskurver for ulike tidsoppløsninger. I dag brukes kurver utarbeidet for timetraffikk som utgangspunkt for beregninger også på døgnnivå, og det medfører en feilkilde i det beregnede hastighetsnivået og dette innvirker også på beregningen av etterspørsel og rutevalg. Det ligger også et potensiale for å studere hastighetsnivå på flerfeltsveier og hvordan motgående trafikk påvirker hastighetsnivået.

Data om rutevalg i rushtrafikk er dessverre ikke tilgjengelig så langt, ettersom reisevaneundersøkelsene ikke dekker dette. Det må dermed skaffes på andre måter, og det kunne vært ønskelig for å forbedre beregningen av rutevalg i transportmodellene.

6.3 Geometrisk forsinkelse gjennom kryss

Datagrunnlaget for forsinkelser gjennom kryss forklart av den geometriske utformingen av krysset er egnet for formålet, og de estimerte verdiene ser ut til å ligge på et fornuftig nivå, gitt bruksområdet i strategiske transportmodeller. Standardverdiene er likevel sjablonmessige, og det kan være behov for en gjennomgang av verdiene ved lokal kalibrering av modellene.

6.4 Trafikkbelastning og forsinkelser gjennom kryss

Det er beregnet forsinkelser for kryss i rushsituasjoner ut fra samme datagrunnlaget som ga geometrisk forsinkelse gjennom kryss. Disse er implementert i transportmodellen, men dette må regnes som en midlertidig løsning til det settes i gang med et arbeid som utarbeider en metodikk som tar hensyn til trafikk mønsteret gjennom kryssene.

7 Referanser

Dimaiuta, M., E. Donnell, et al. (2011). Modeling Operating Speed. Transportation Research CIRCULAR E-C151. Washington DC, Transportation Research board.

ESRI (1998) ESRI Shapefile Technical Description. An ESRI White Paper

Richl, Laurel og Tarek Sayed (2005): Effect of speed prediction models on design consistency. 3rd International Symposium on Highway Geometric Design 2005

Sakshaug, K. og H. Toftegård (2011). Fartsmodell for næringslivets transporter Litteraturstudium. SINTEF Rapport A17522. Trondheim, SINTEF.

Tørset, Trude, Arvid Aakre, Vilhelm Børnes og Ola Rennemo (2011): Fartsmodell for næringslivets transporter. Datagrunnlag og dokumentasjon av modell. SINTEF-rapport 17524. SINTEF. Trondheim.

Statens vegvesen Vegdirektoratet Oslo (2013). Hb V120 Premisser for geometrisk utforming av veger.

8 Vedlegg

- Volum hastighetskurver
- Fartsmodell TNext kode

Volum/hastighetskruver

De forskjellige grafene i vedlegget viser volum/hastighetskruver for diverse tellepunkt i Norge. Statens vegvesen sine offisielle registreringspunkt ID'er er brukt til å identifisere punktene. Volum er beregnet som et gjennomsnitt over 300 sekunder og farten er beregnet som gjennomsnittsfart for alle kjøretøy innen dette intervallet.

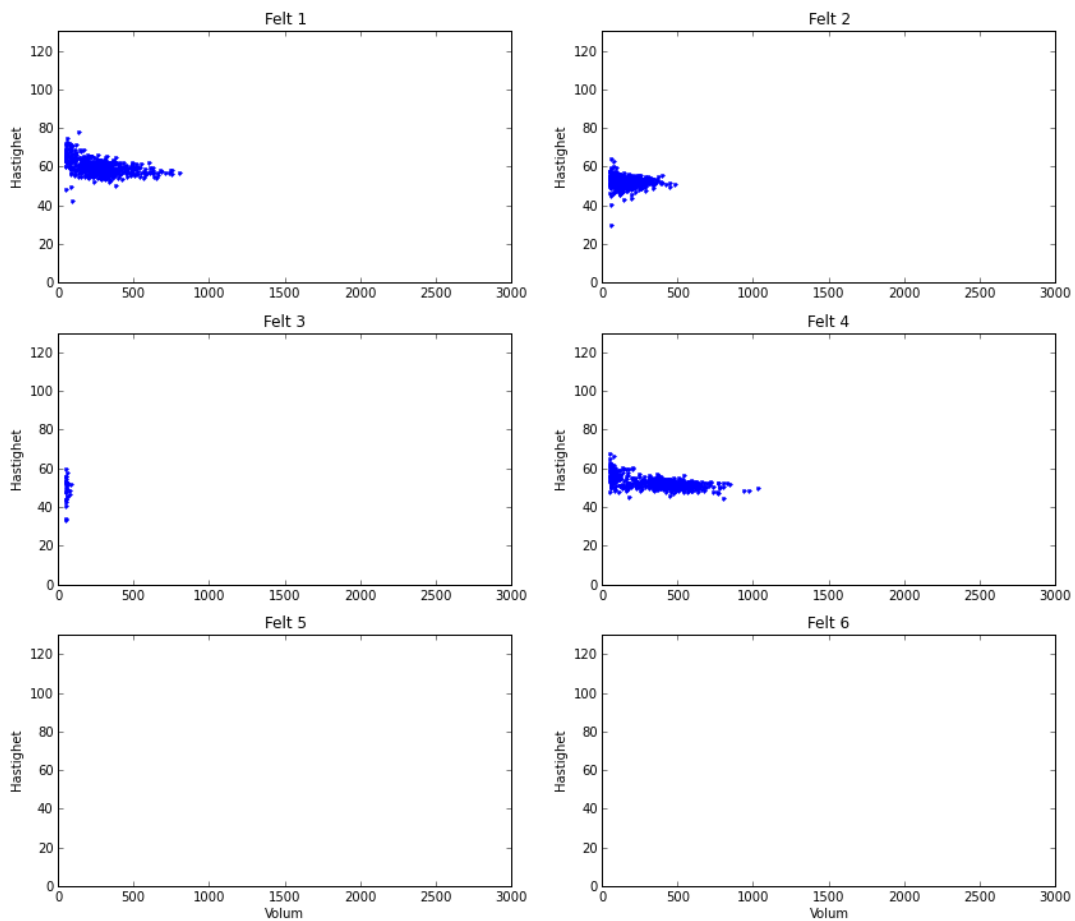
Tabell 1: Id for volumhastighetsplott fordelt på felt for 38 registreringspunkter er vedlagt som appendix.

Tellepunktnummer	Fylkesnavn	Kommunenavn	Fylke	Kom	Vk	Vs	Vnr	Hp	Fra m
1200001	HORDALAND	BERGEN	12	0	F	V	585	4	2700
1200002									
1200003	HORDALAND	BERGEN	12	0	E	V	39	19	836
1200011	HORDALAND	BERGEN	12	0	F	V	556	3	2600
1200013									
1200016	HORDALAND	BERGEN	12	0	F	V	580	1	6700
1200033									
1200041									
1200042									
1200052	HORDALAND	SAMNANGER	12	0	F	V	7	19	11320
1200058									
1200059	HORDALAND	BERGEN	12	0	R	V	555	4	10
1200069									
1200071									
1200072	HORDALAND	FJELL	12	0	F	V	561	1	4650
1200073	HORDALAND	ASKØY	12	0	F	V	562	3	90
1200074	HORDALAND	VOSS	12	0	E	V	16	3	650
1200075									
1200078									
1200081	HORDALAND	LINDÅS	12	63	K	V	9002	1	1070
1200083									
1200085									
1200086									
1200088									
1200090	HORDALAND	OS	12	0	F	V	552	2	1200
1200092	HORDALAND	BERGEN	12	0	F	V	582	4	3422
1200096									
1200100	HORDALAND	STORD	12	0	E	V	39	4	8495
1200112	HORDALAND	BERGEN	12	0	E	V	39	74	13190
1200113									
1200127	HORDALAND	LINDÅS	12	0	F	V	565	1	2626
1200029									
1200234	HORDALAND	BERGEN	12	0	F	V	562	1	1500
1200277	HORDALAND	BERGEN	12	0	F	V	540	2	1420
1200381	HORDALAND	BERGEN	12	0	E	V	39	18	500
1200413	HORDALAND	BERGEN	12	0	E	V	39	22	500
1201151	HORDALAND	BERGEN	12	0	E	V	16	11	10466

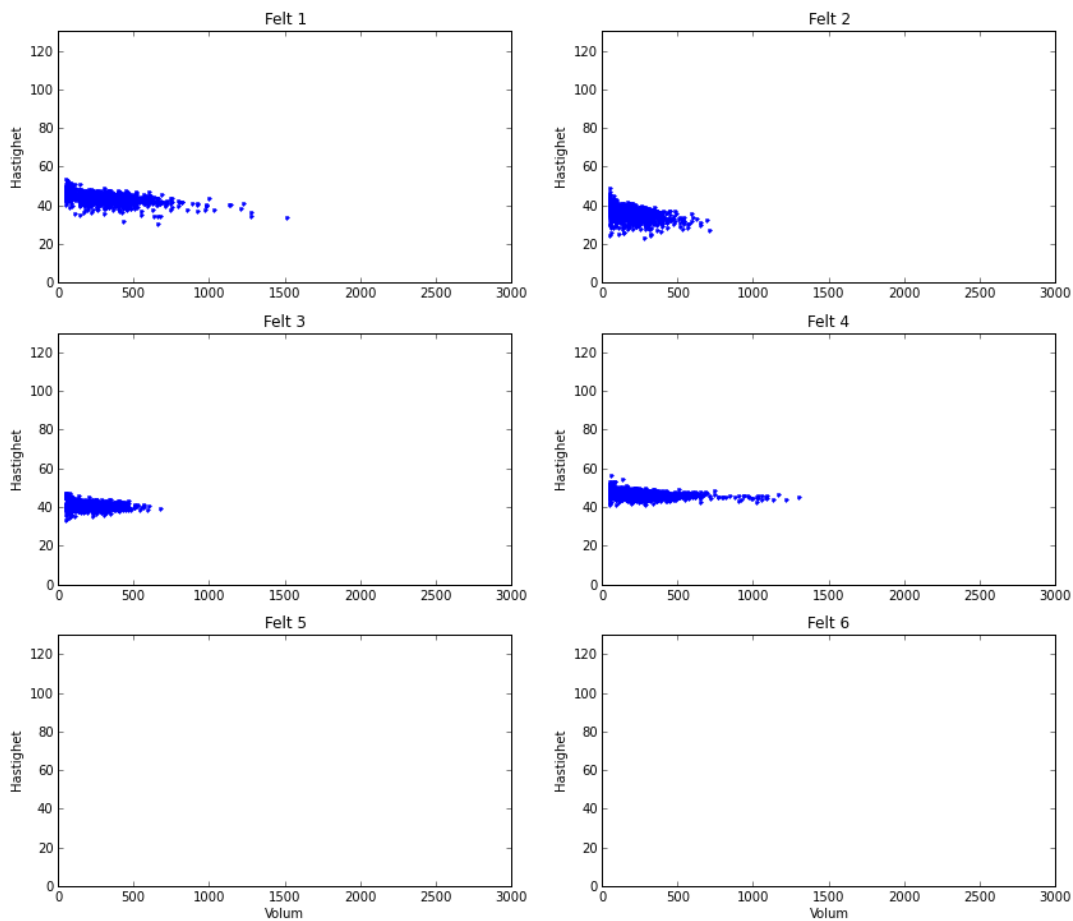
Tellepunktnummer	Fylkesnavn	Kommunenavn	Fylke	Kom	Vk	Vs	Vnr	Hp	Fra m
1201622	HORDALAND	ODDA	12	0	R	V	13	3	12600
1201632	HORDALAND	ULLENSVANG	12	0	R	V	13	7	130
1204365									
1200001	HORDALAND	BERGEN	12	0	F	V	585	4	2700
1200002									
1200003	HORDALAND	BERGEN	12	0	E	V	39	19	836
1200011	HORDALAND	BERGEN	12	0	F	V	556	3	2600
1200013									
1200016	HORDALAND	BERGEN	12	0	F	V	580	1	6700
1200033									
1200041									
1200042									
1200052	HORDALAND	SAMNANGER	12	0	F	V	7	19	11320
1200058									
1200059	HORDALAND	BERGEN	12	0	R	V	555	4	10
1200069									
1200071									
1200072	HORDALAND	FJELL	12	0	F	V	561	1	4650
1200073	HORDALAND	ASKØY	12	0	F	V	562	3	90
1200074	HORDALAND	VOSS	12	0	E	V	16	3	650
1200075									
1200078									
1200081	HORDALAND	LINDÅS	12	63	K	V	9002	1	1070
1200083									
1200085									
1200086									
1200088									
1200090	HORDALAND	OS	12	0	F	V	552	2	1200
1200092	HORDALAND	BERGEN	12	0	F	V	582	4	3422
1200096									
1200100	HORDALAND	STORD	12	0	E	V	39	4	8495
1200112	HORDALAND	BERGEN	12	0	E	V	39	74	13190
1200113									
1200127	HORDALAND	LINDÅS	12	0	F	V	565	1	2626
1200029									
1200234	HORDALAND	BERGEN	12	0	F	V	562	1	1500
1200277	HORDALAND	BERGEN	12	0	F	V	540	2	1420
1200381	HORDALAND	BERGEN	12	0	E	V	39	18	500
1200413	HORDALAND	BERGEN	12	0	E	V	39	22	500
1201151	HORDALAND	BERGEN	12	0	E	V	16	11	10466
1201622	HORDALAND	ODDA	12	0	R	V	13	3	12600
1204365									

A.1 Plott fra registeringspunkter

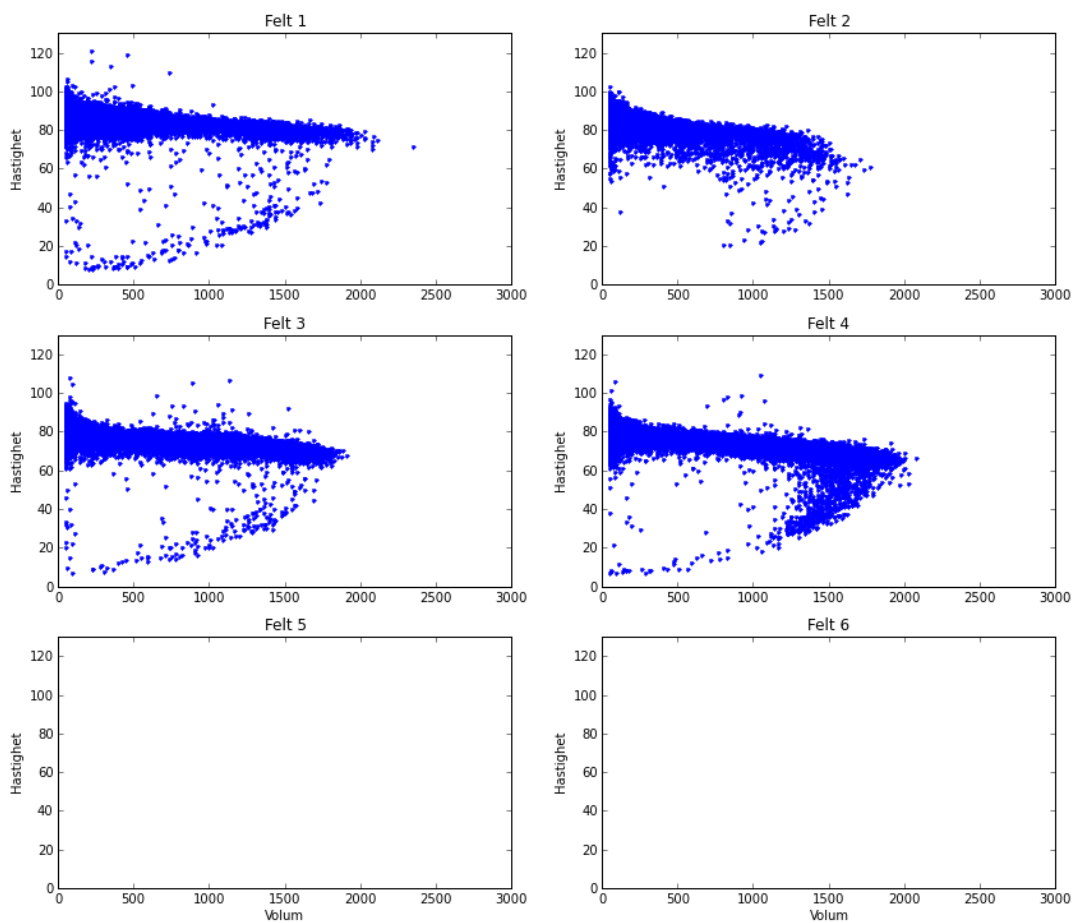
Registreringspunkt 1200001



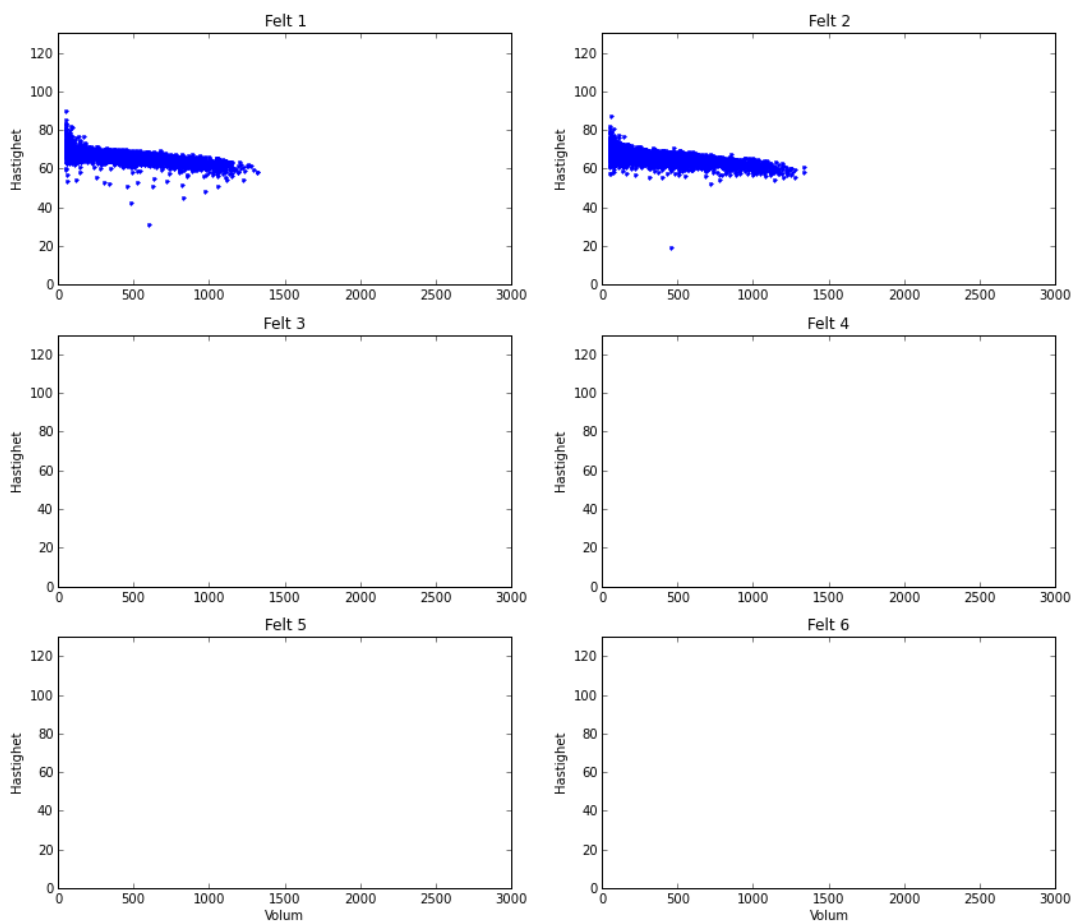
Registreringspunkt 1200002



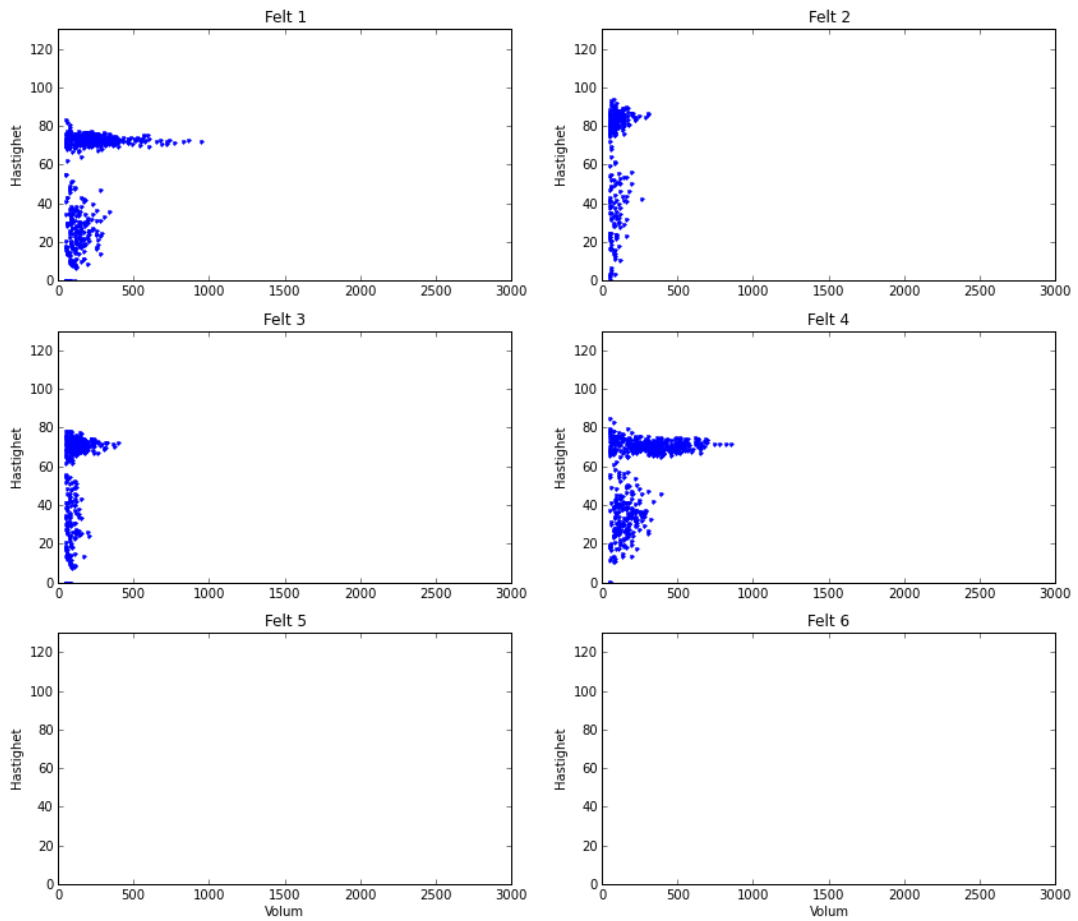
Registreringspunkt 1200003



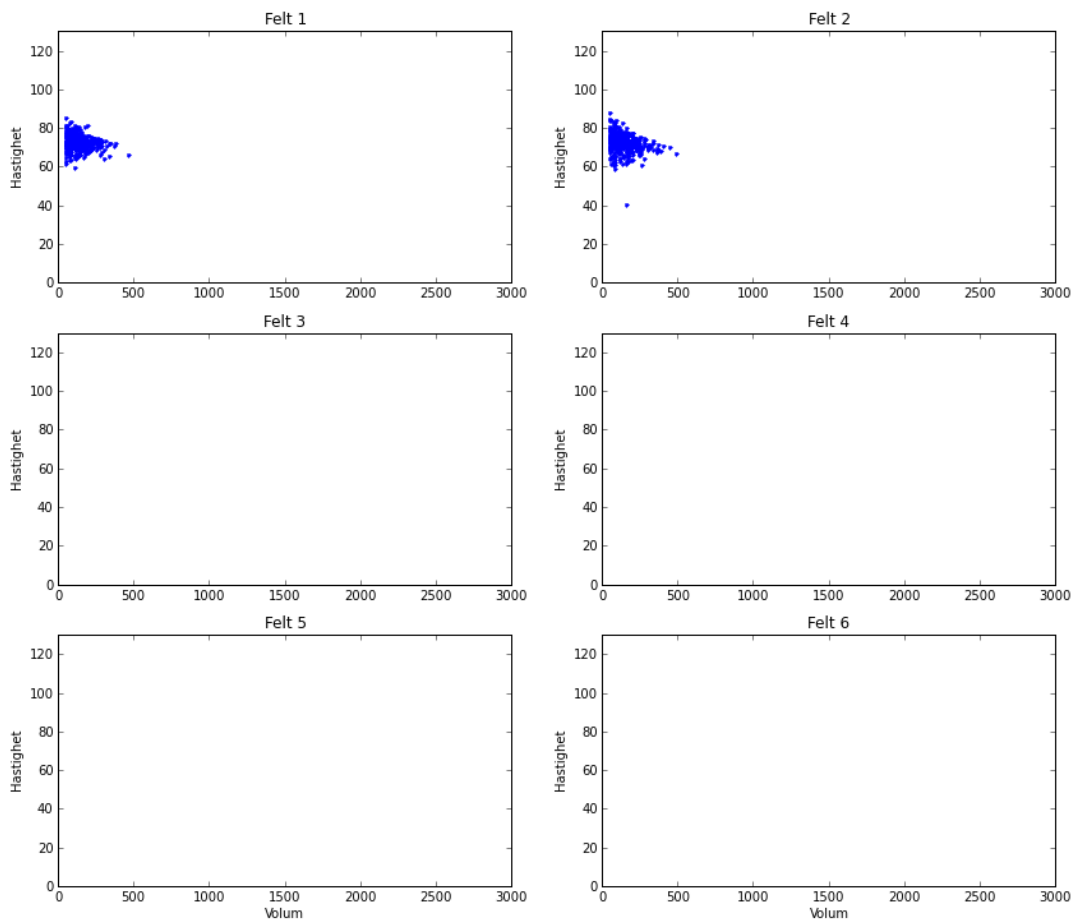
Registreringspunkt 1200011



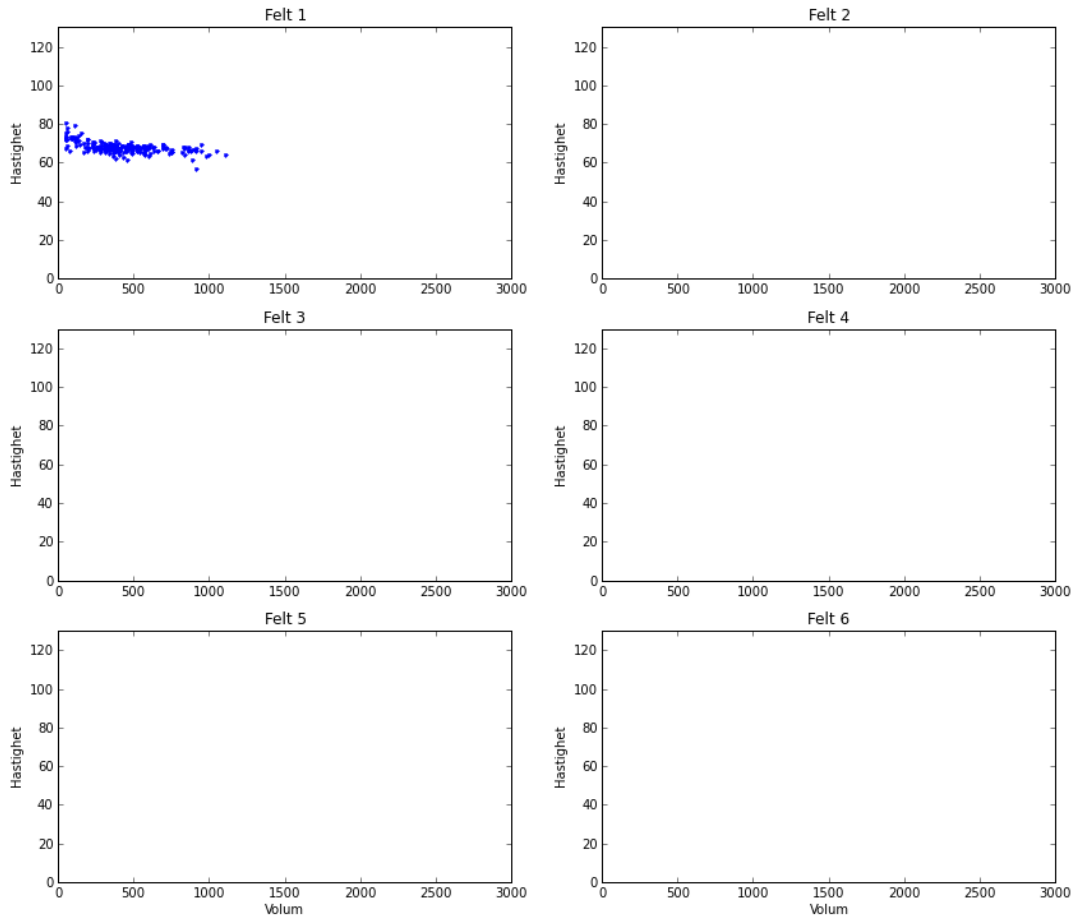
Registreringspunkt 1200013



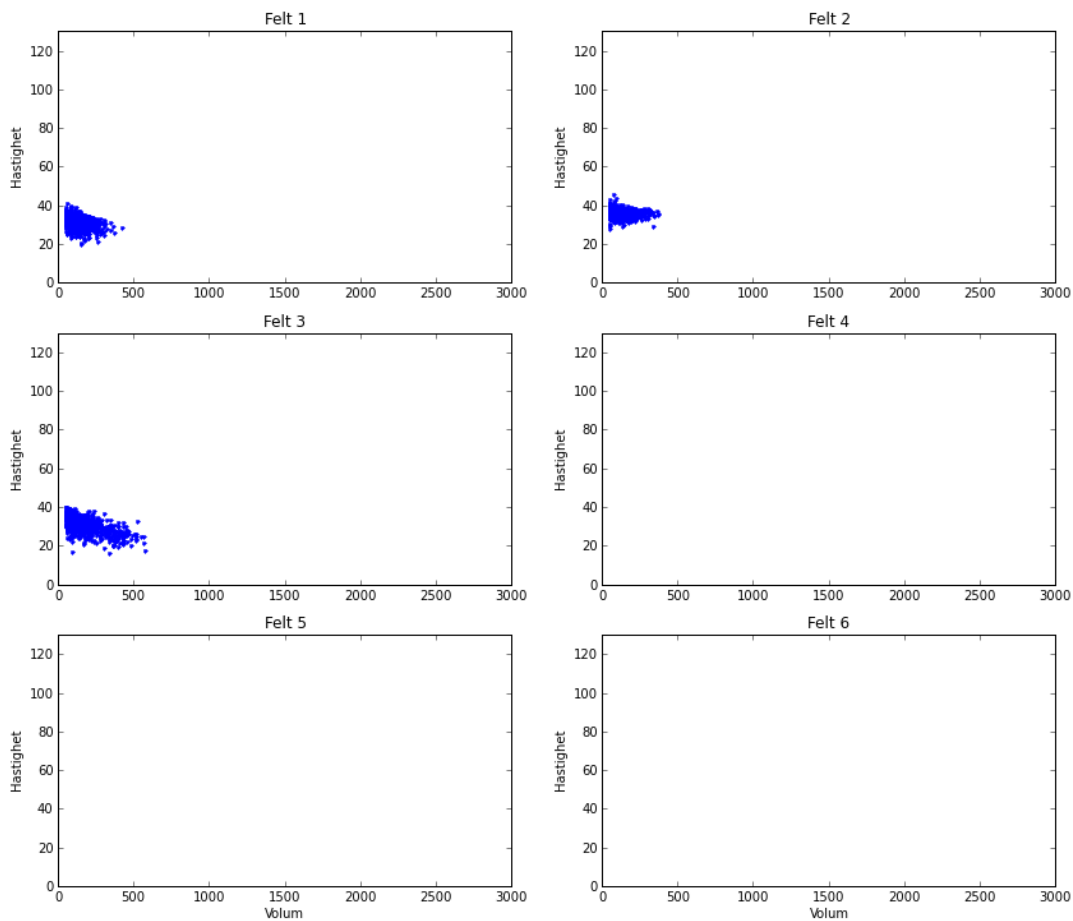
Registreringspunkt 1200018



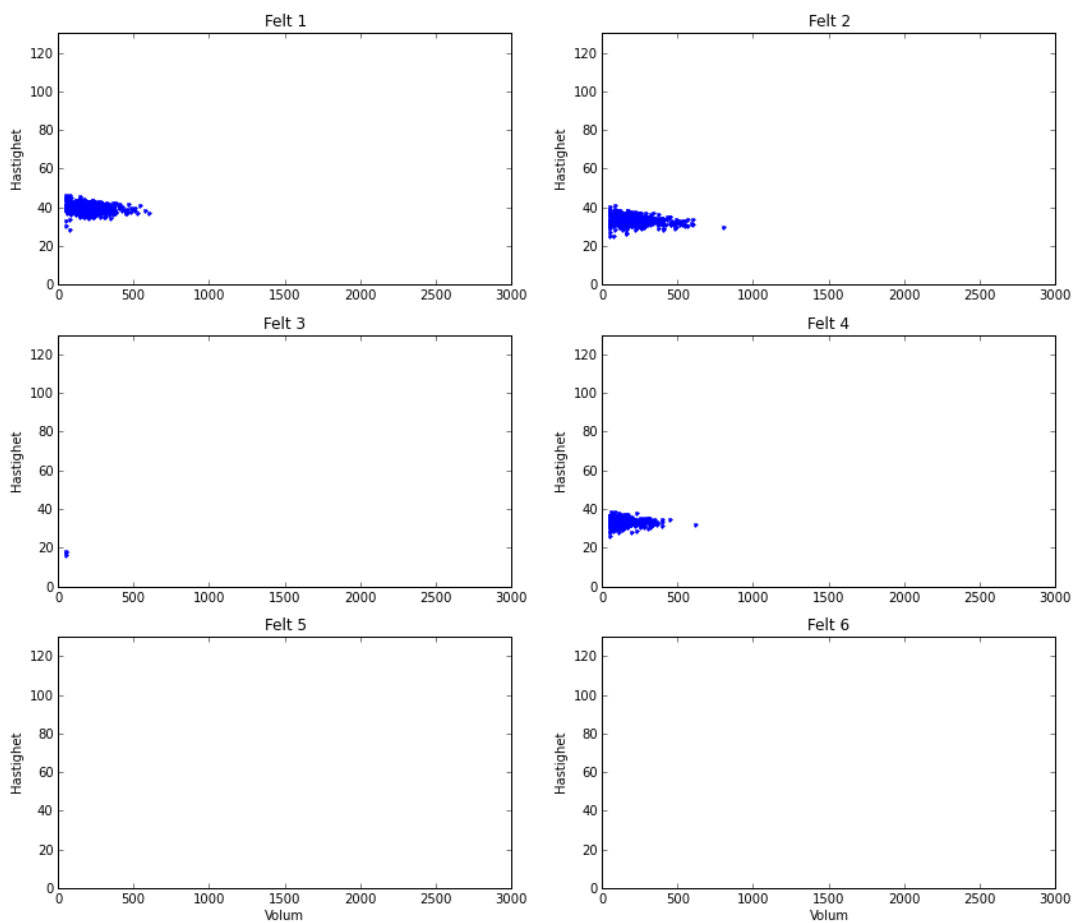
Registreringspunkt 1200033



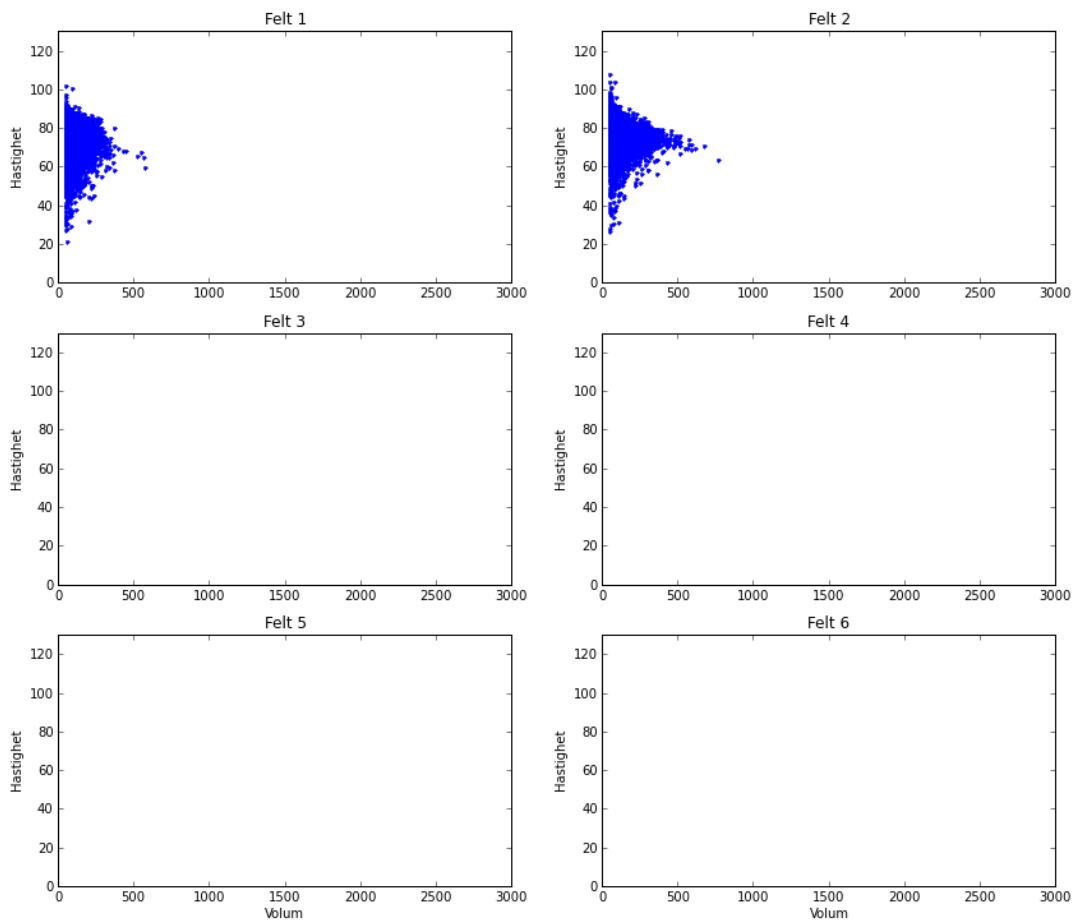
Registreringspunkt 1200041



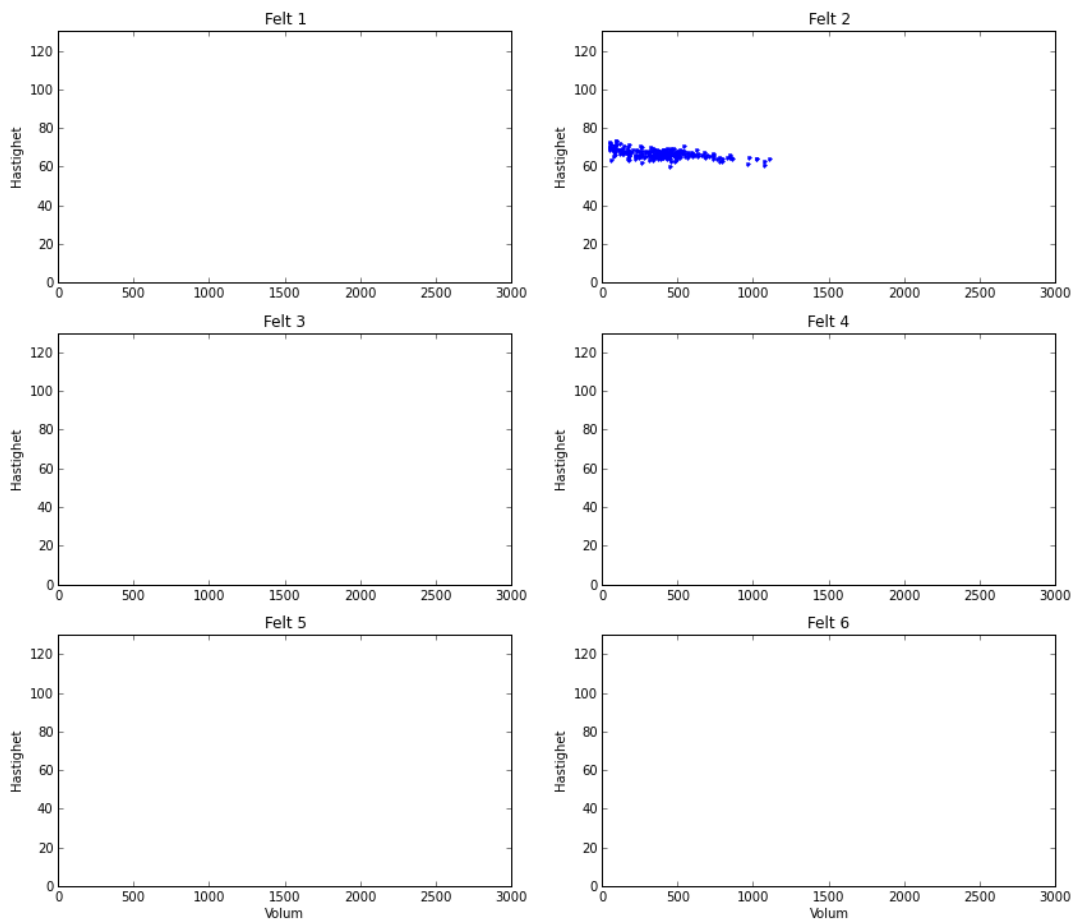
Registreringspunkt 1200042



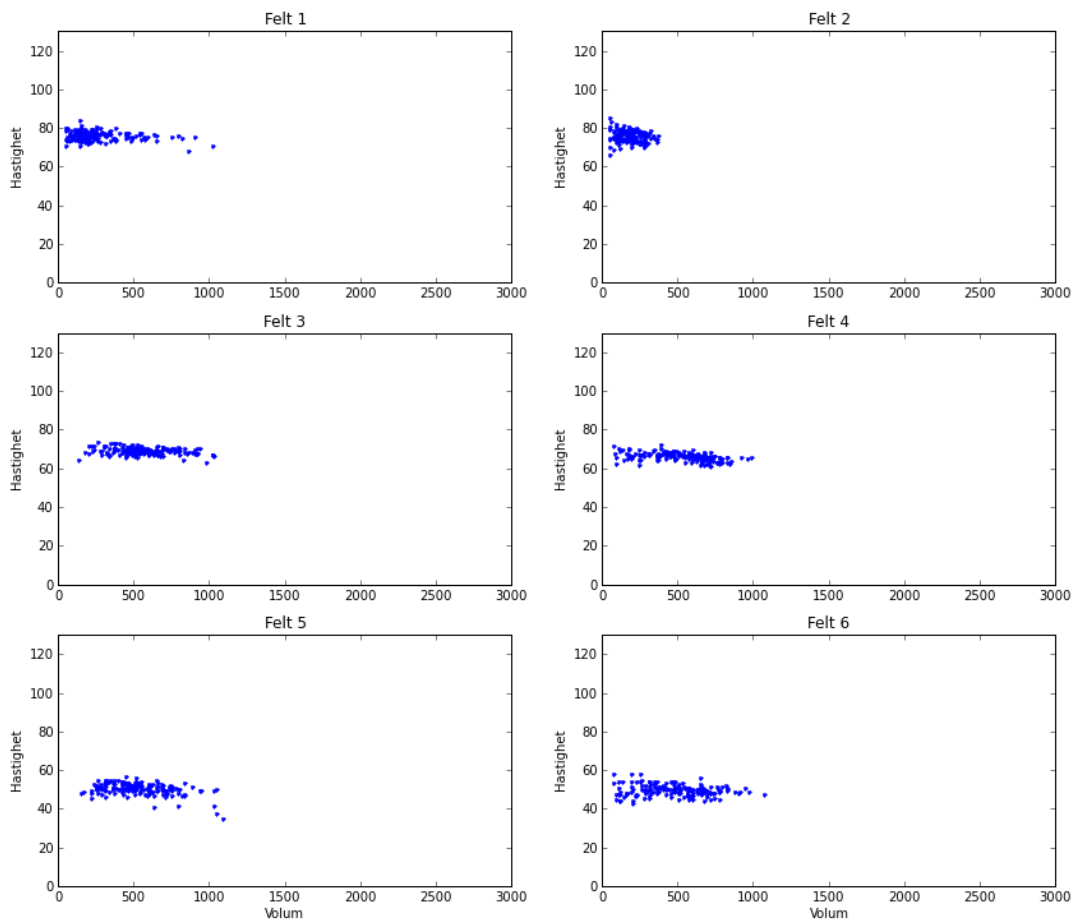
Registreringspunkt 1200052



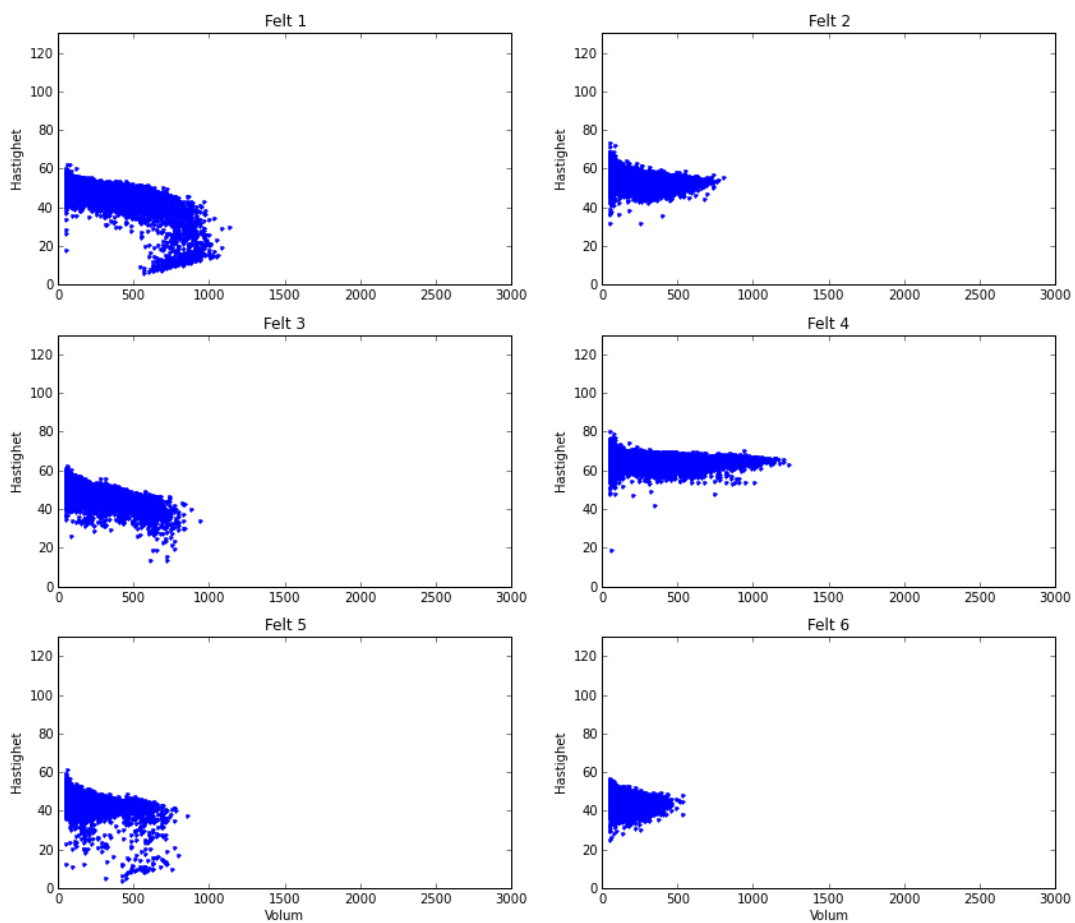
Registreringspunkt 1200058



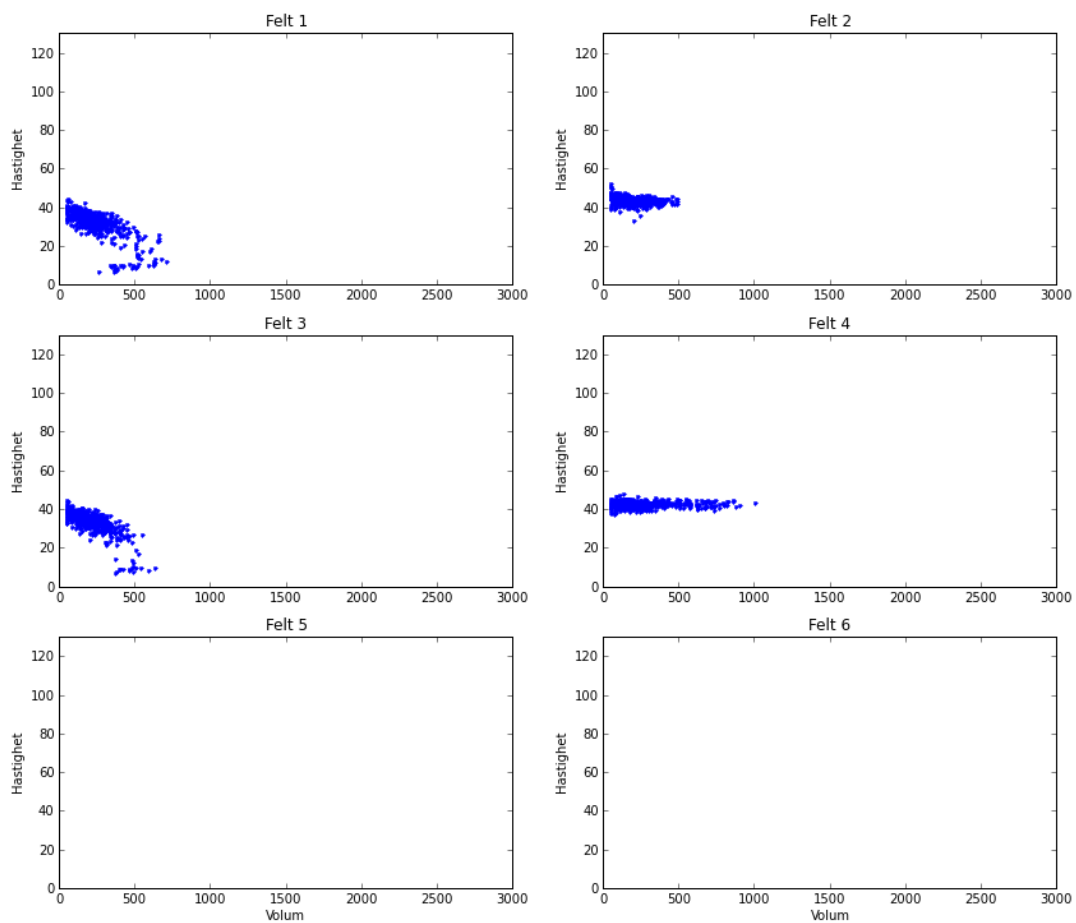
Registreringspunkt 1200059



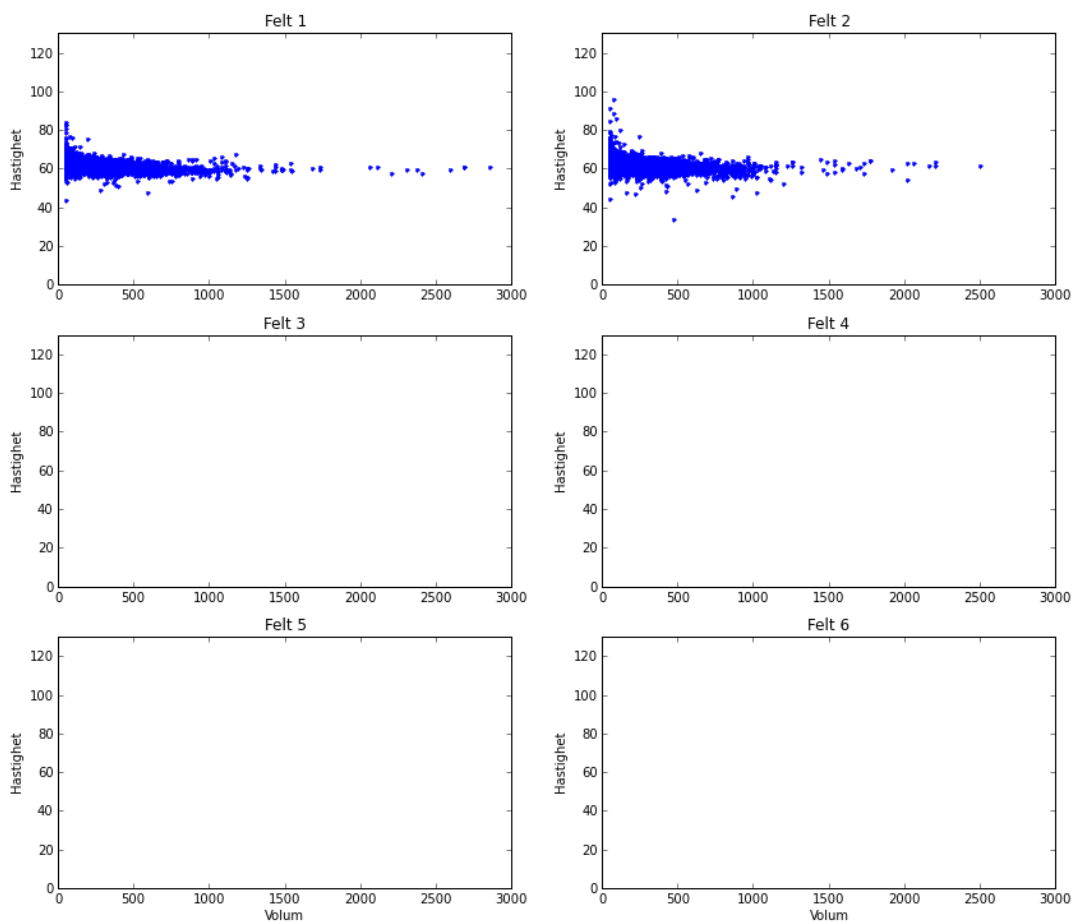
Registreringspunkt 1200069



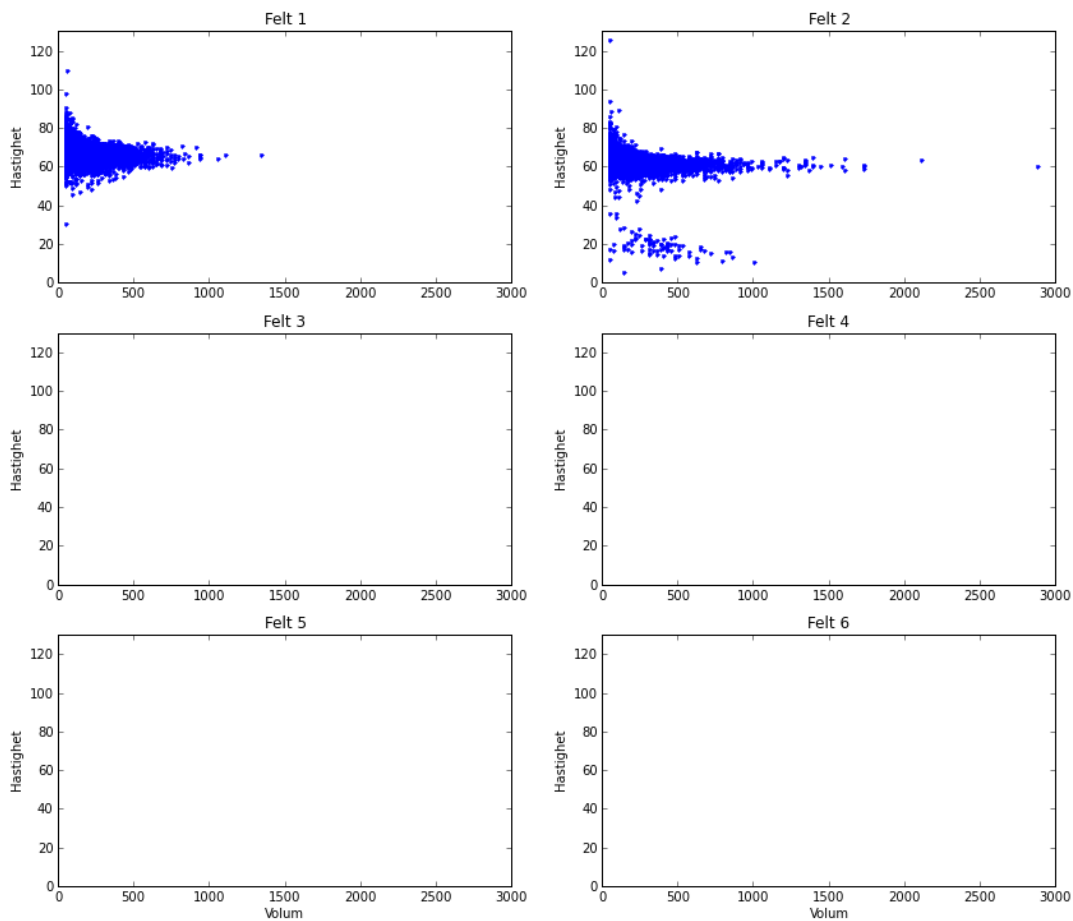
Registreringspunkt 1200071



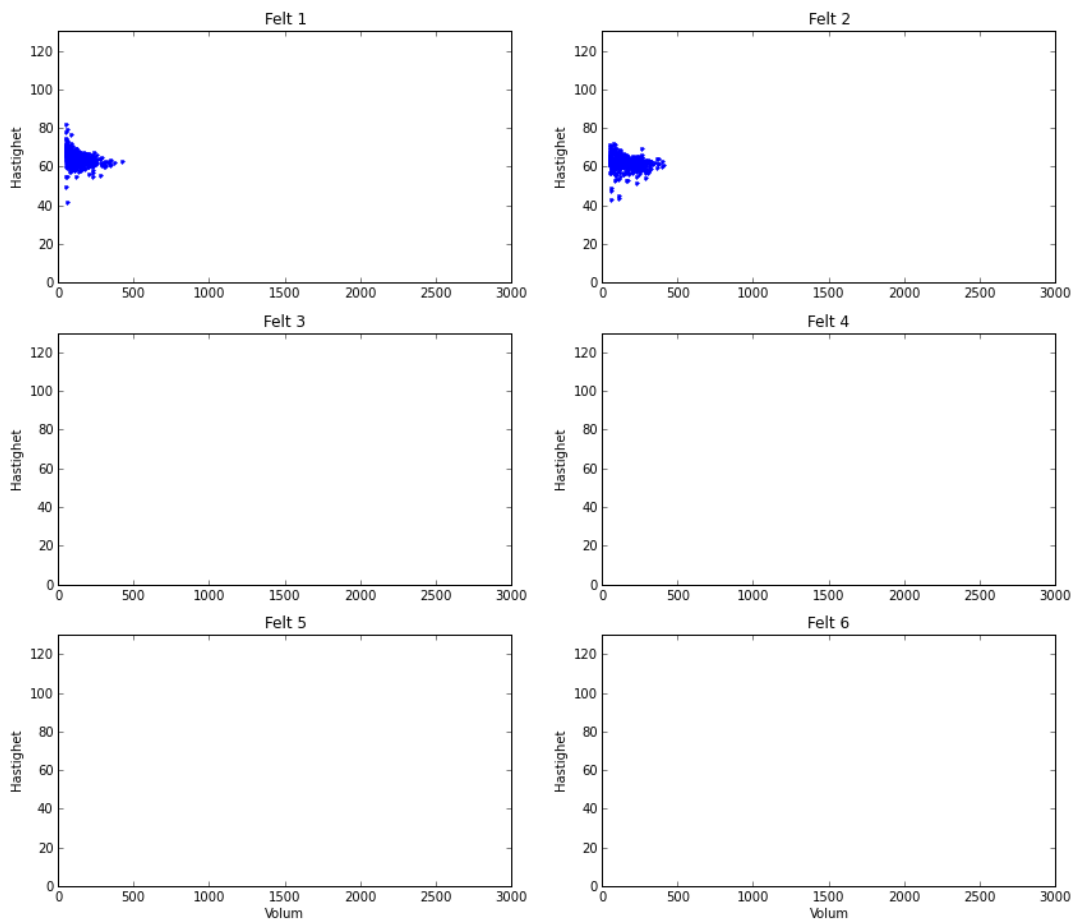
Registreringspunkt 1200072



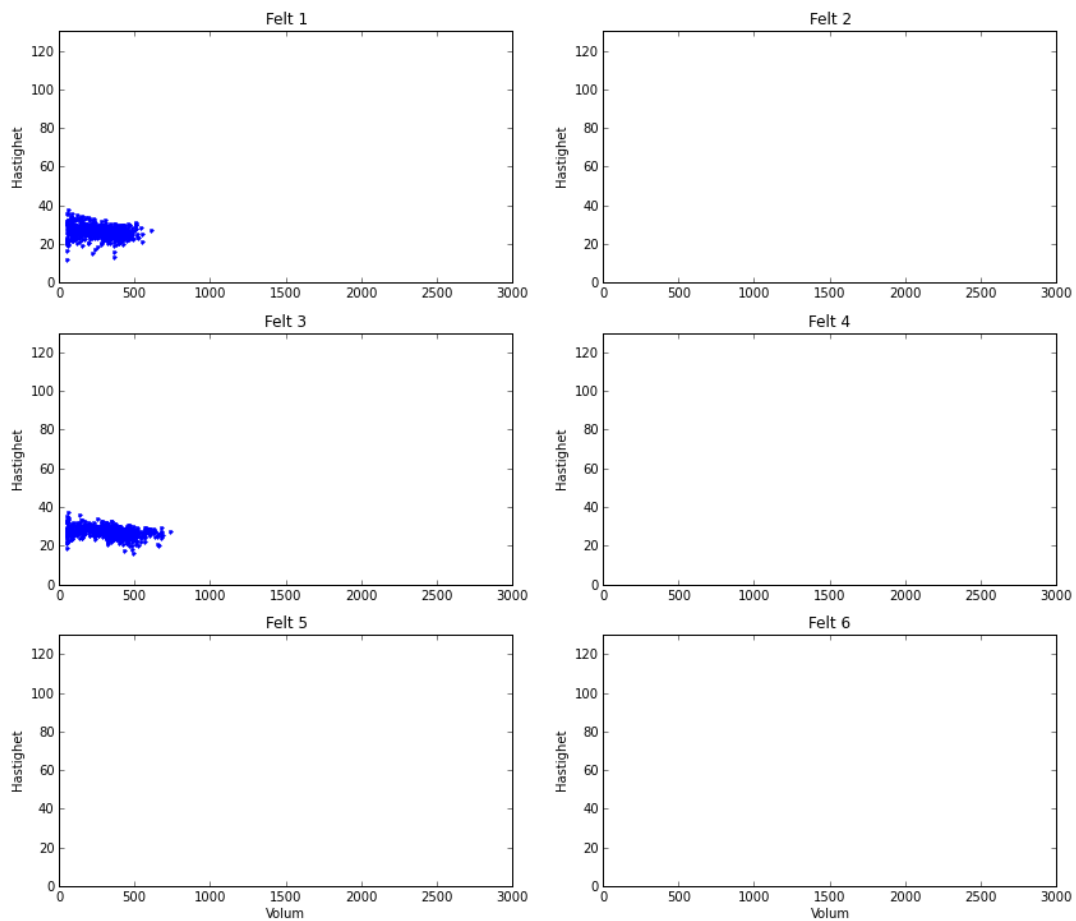
Registreringspunkt 1200073



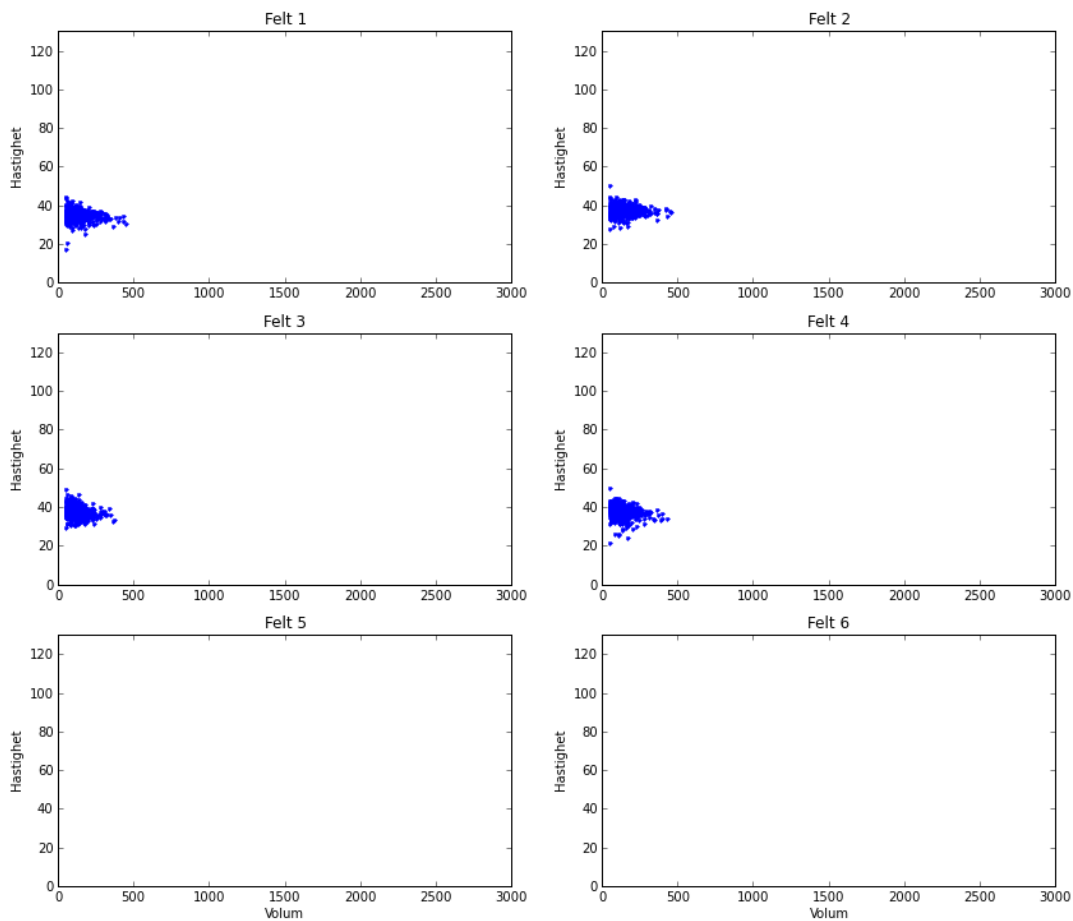
Registreringspunkt 1200074



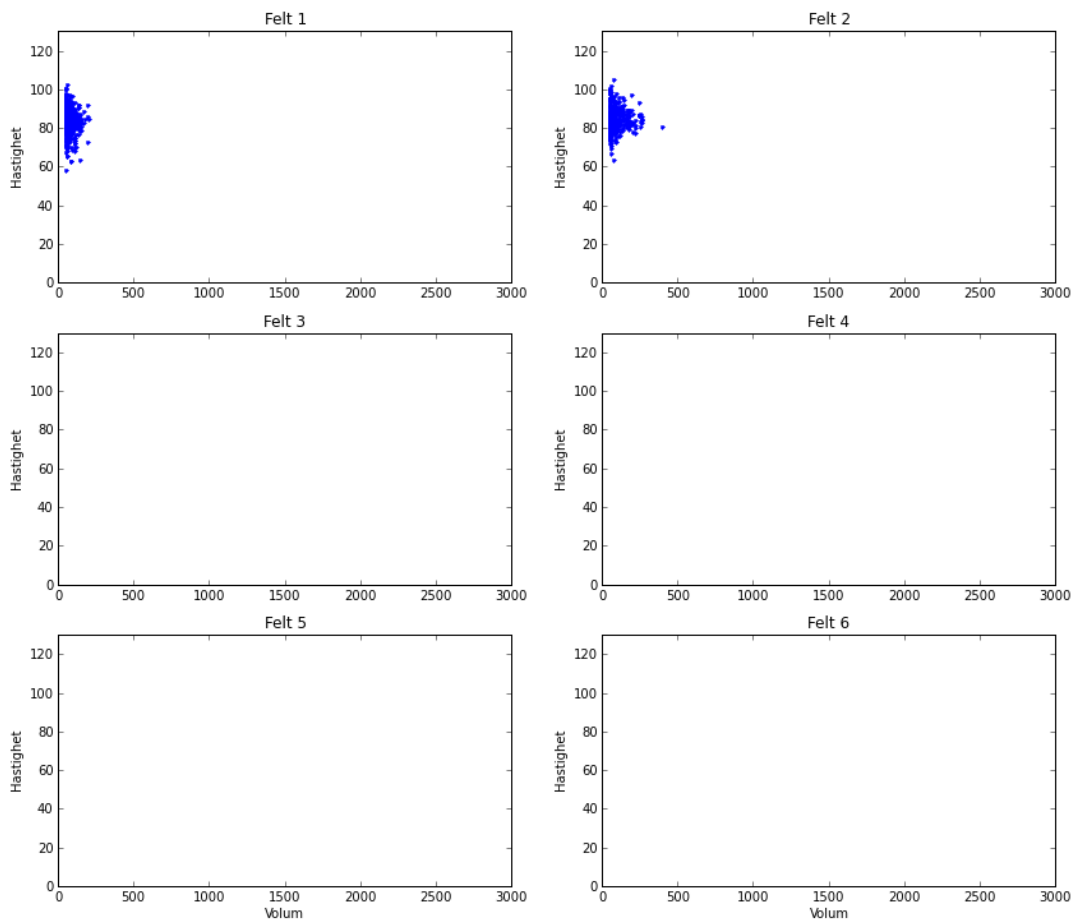
Registreringspunkt 1200075



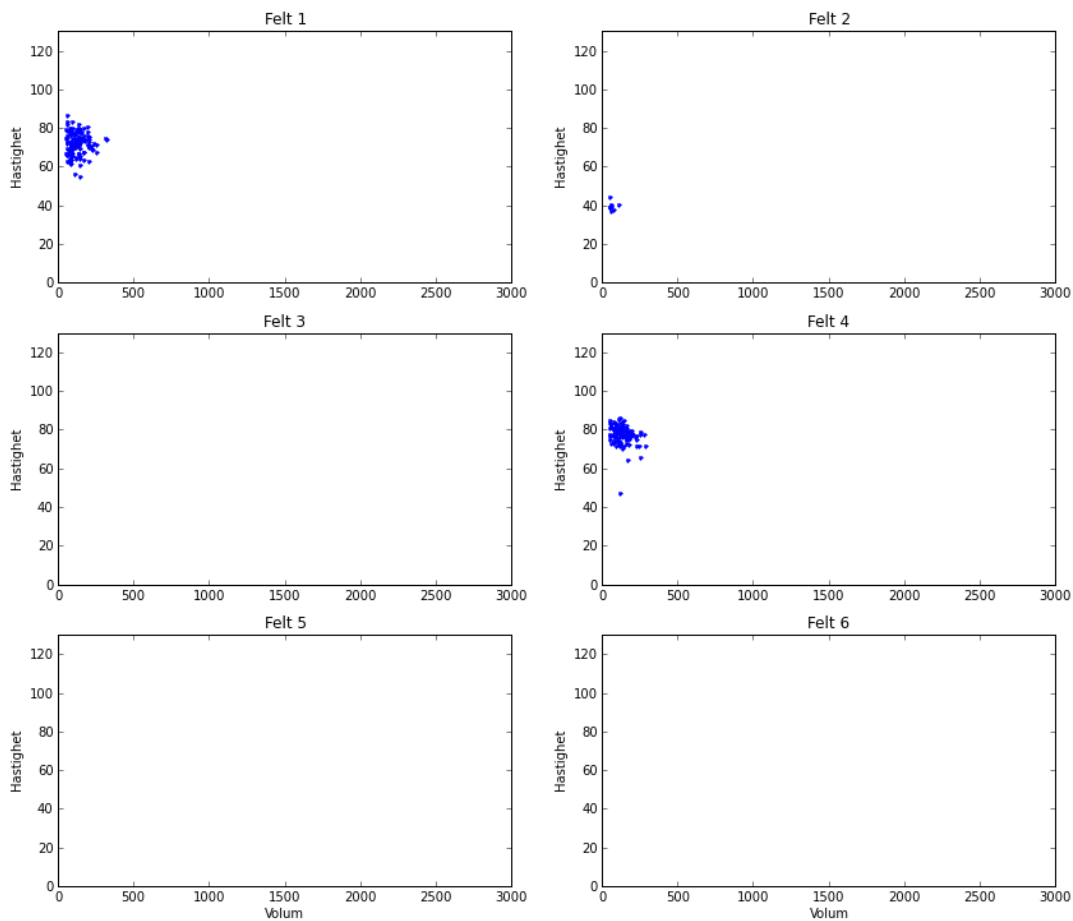
Registreringspunkt 1200078



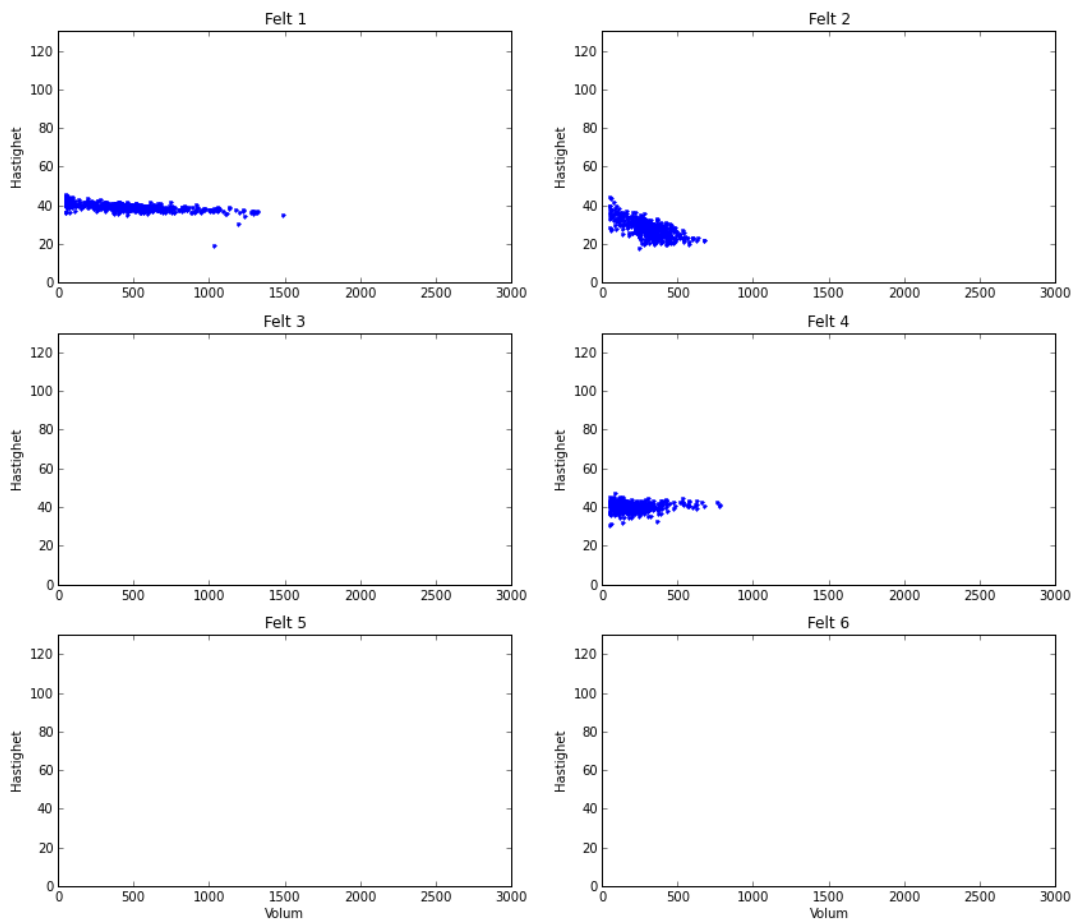
Registreringspunkt 1200081



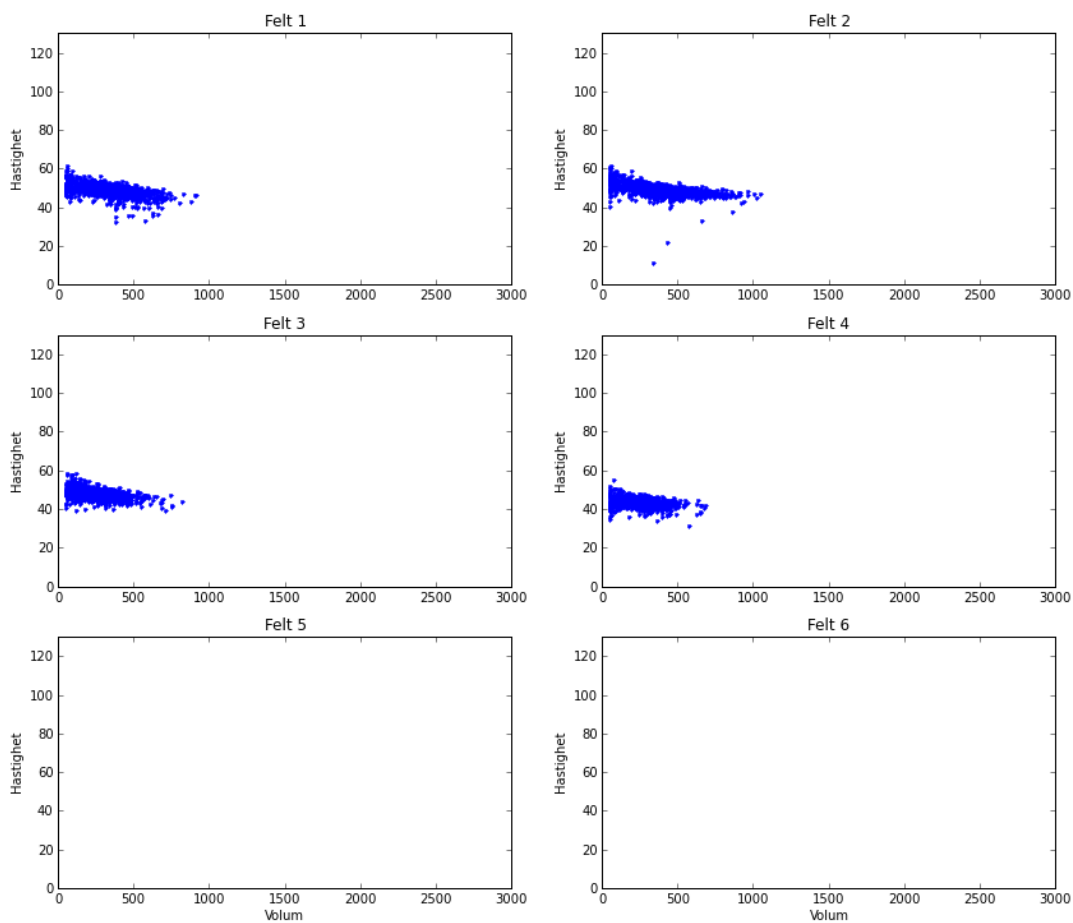
Registreringspunkt 1200083



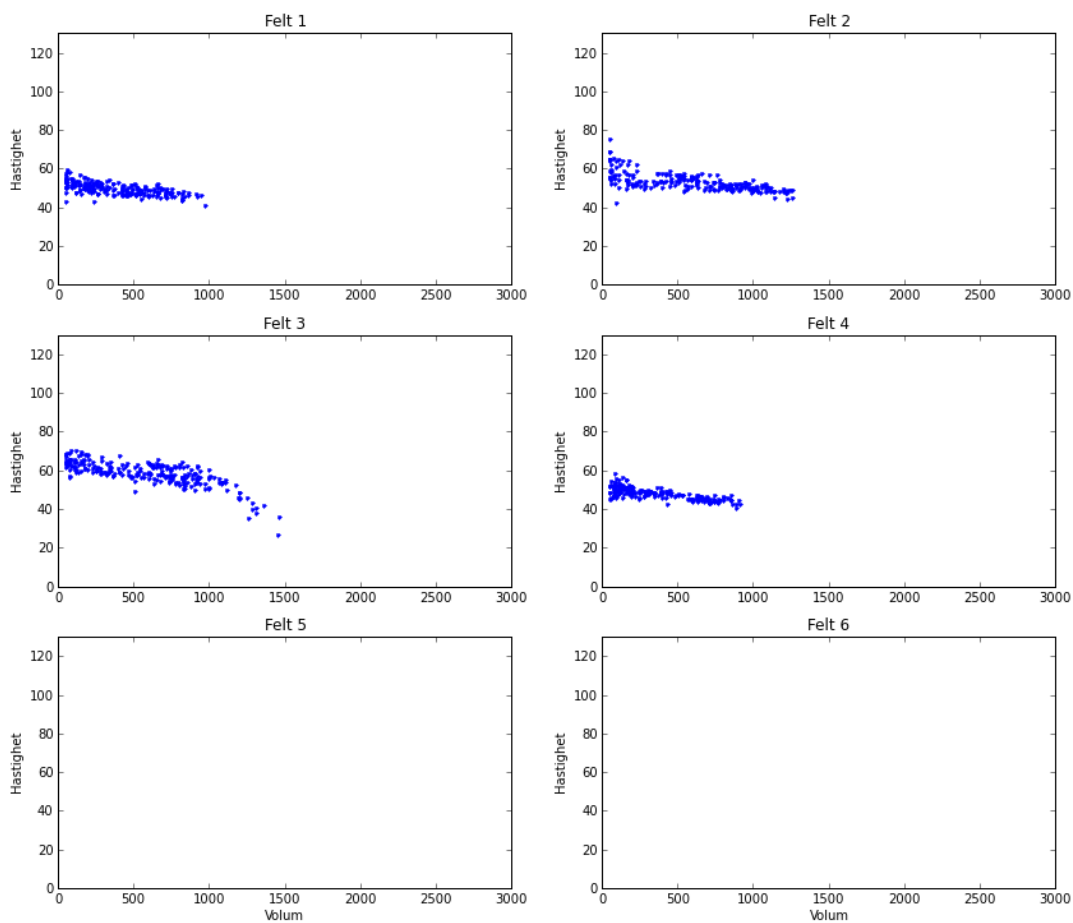
Registreringspunkt 1200085



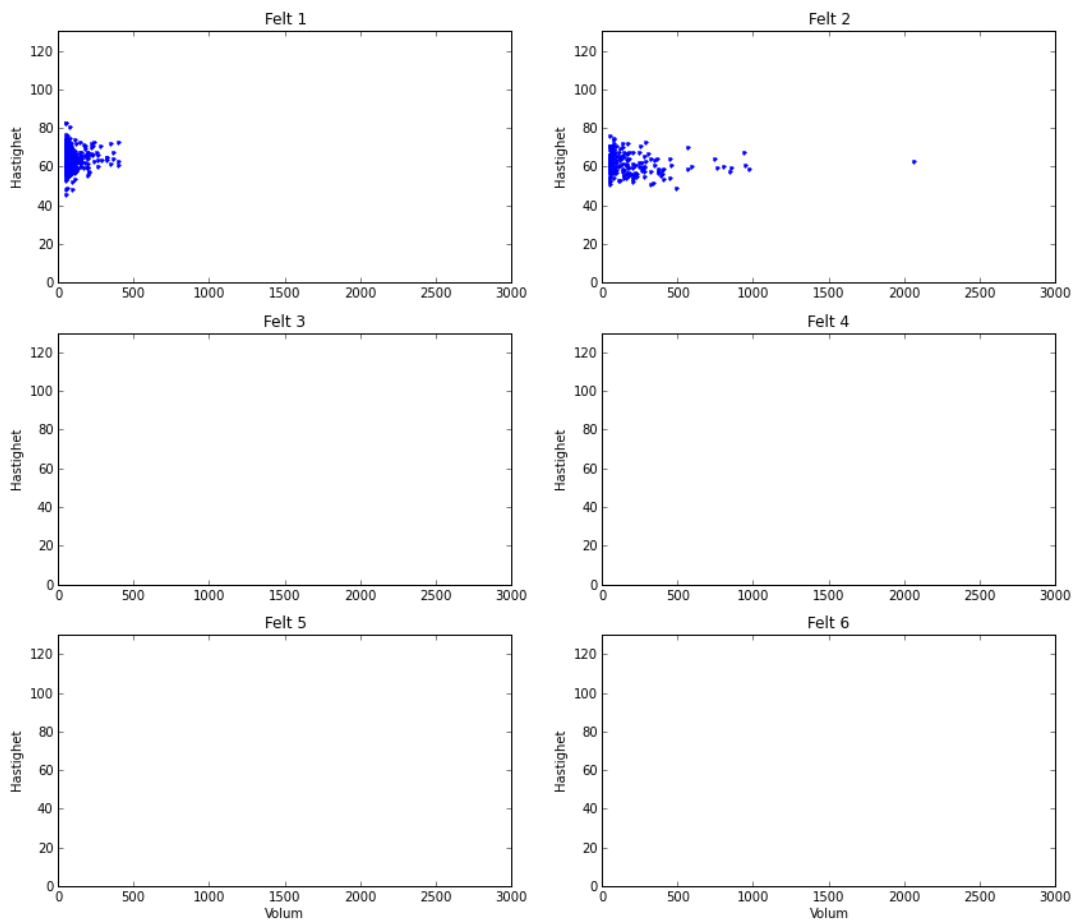
Registreringspunkt 1200086



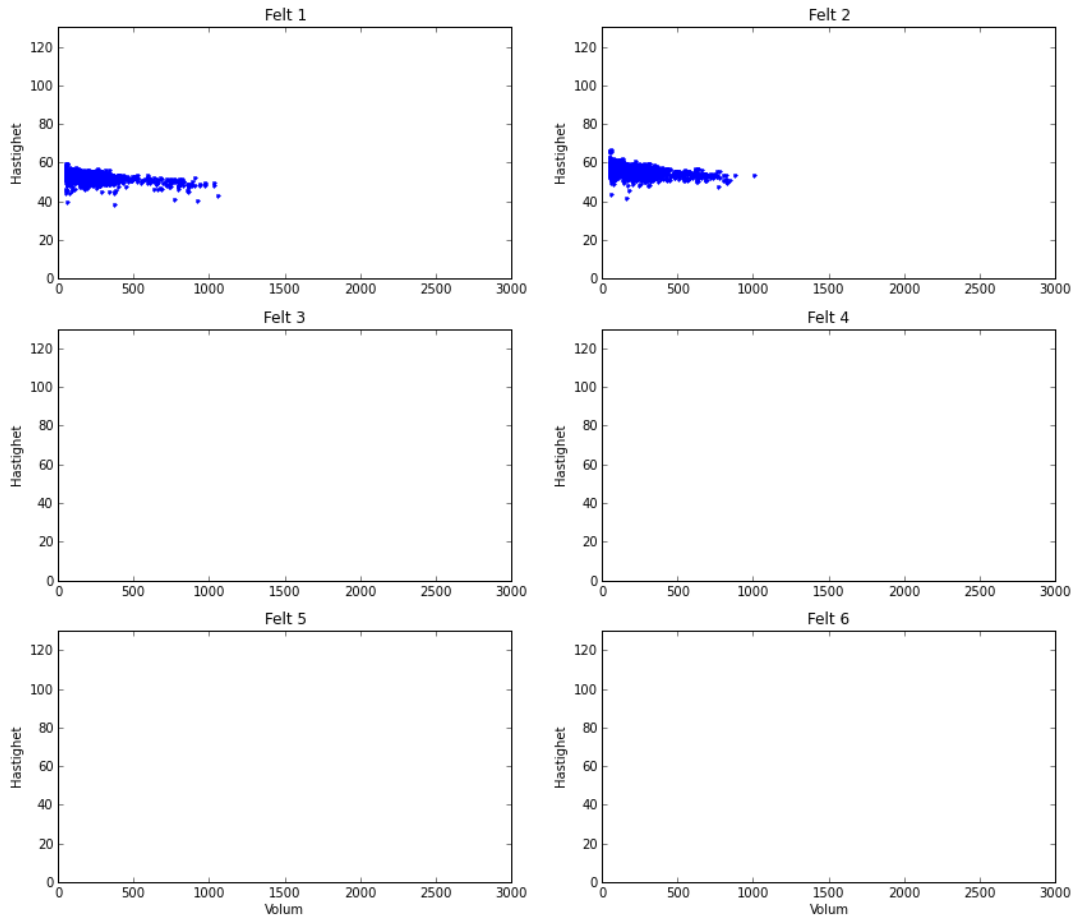
Registreringspunkt 1200088



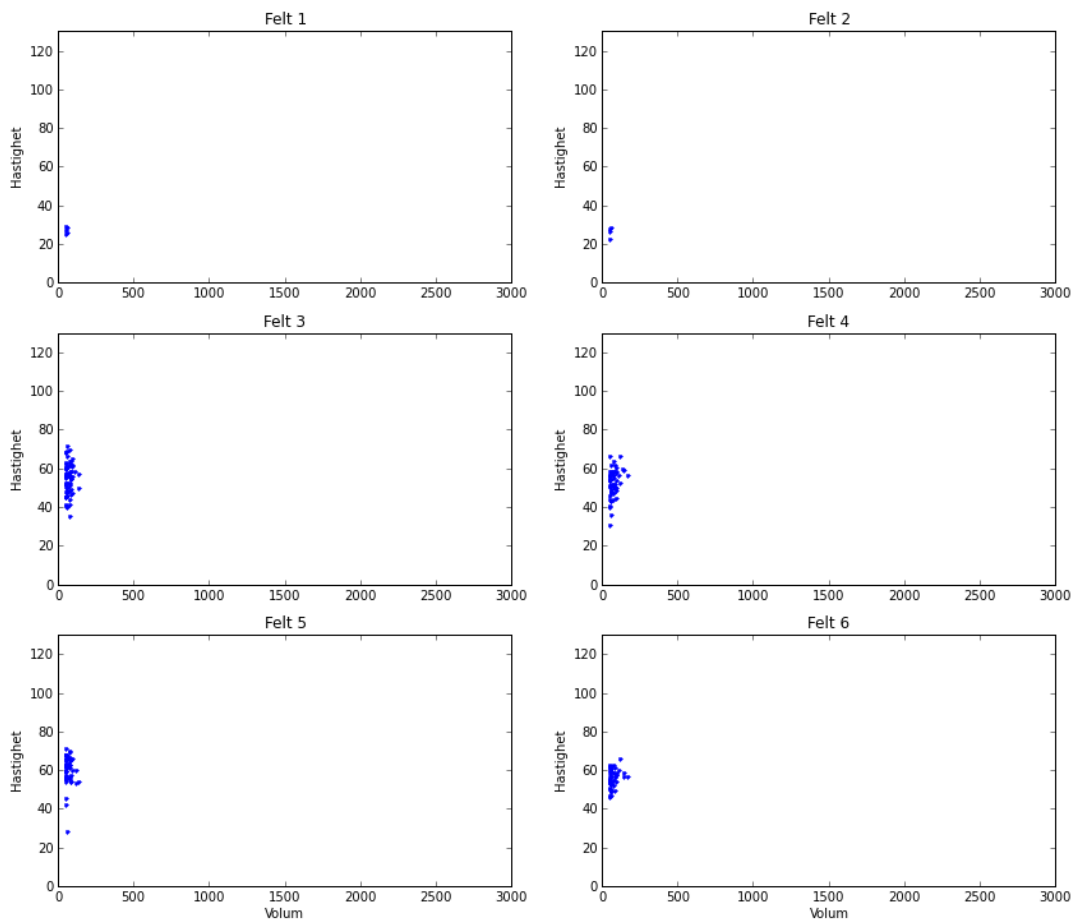
Registreringspunkt 1200090



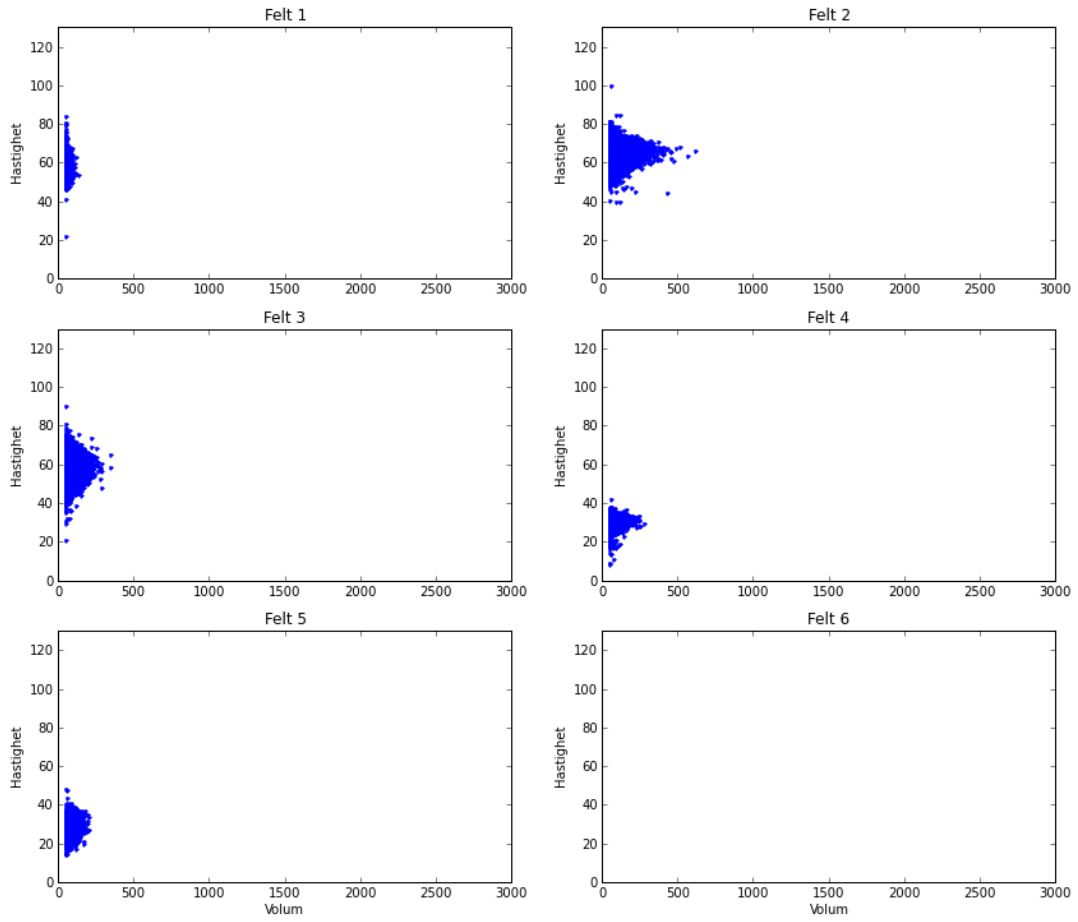
Registreringspunkt 1200092



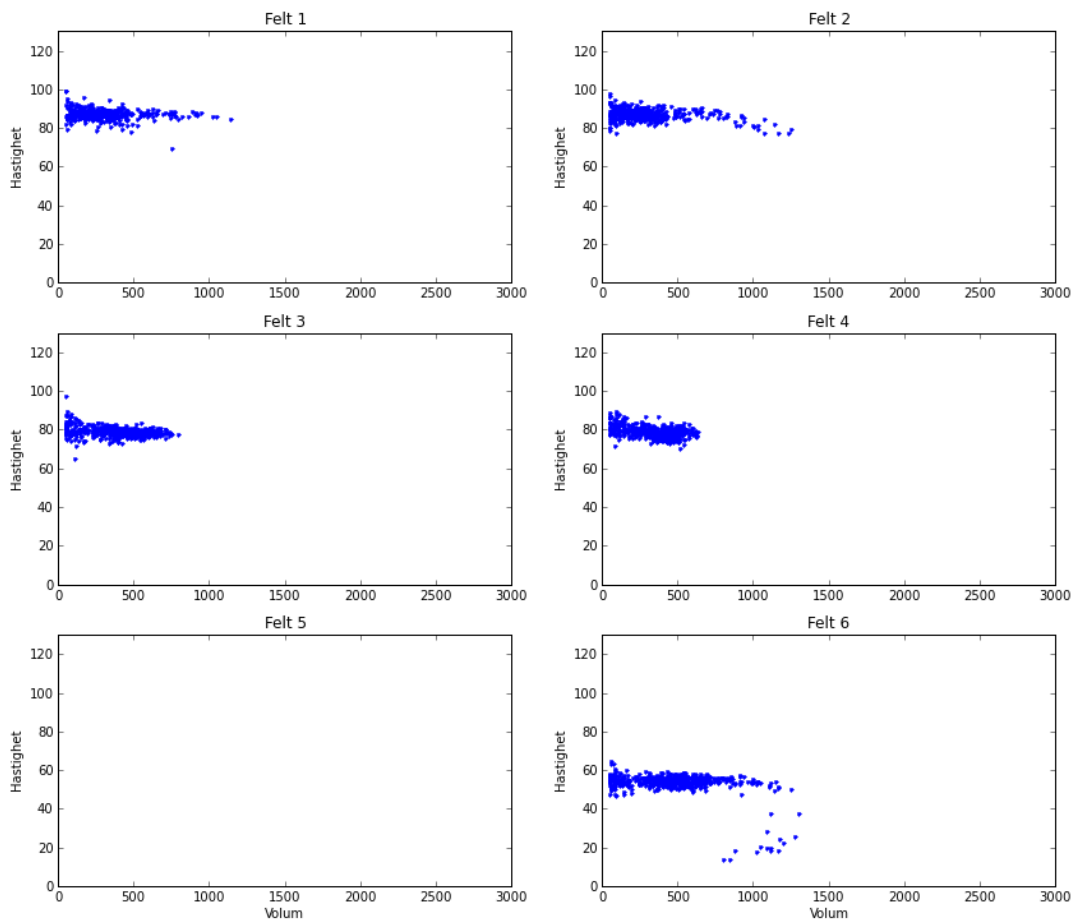
Registreringspunkt 1200096



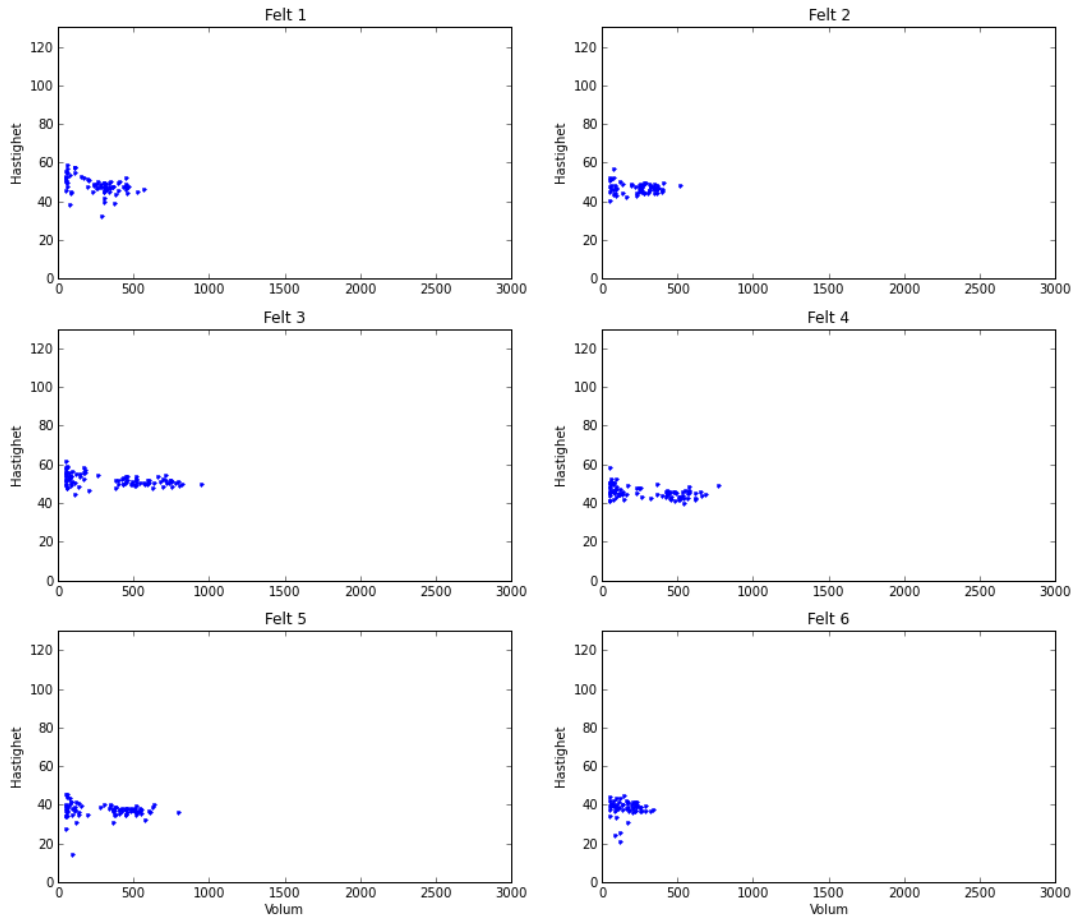
Registreringspunkt 1200100



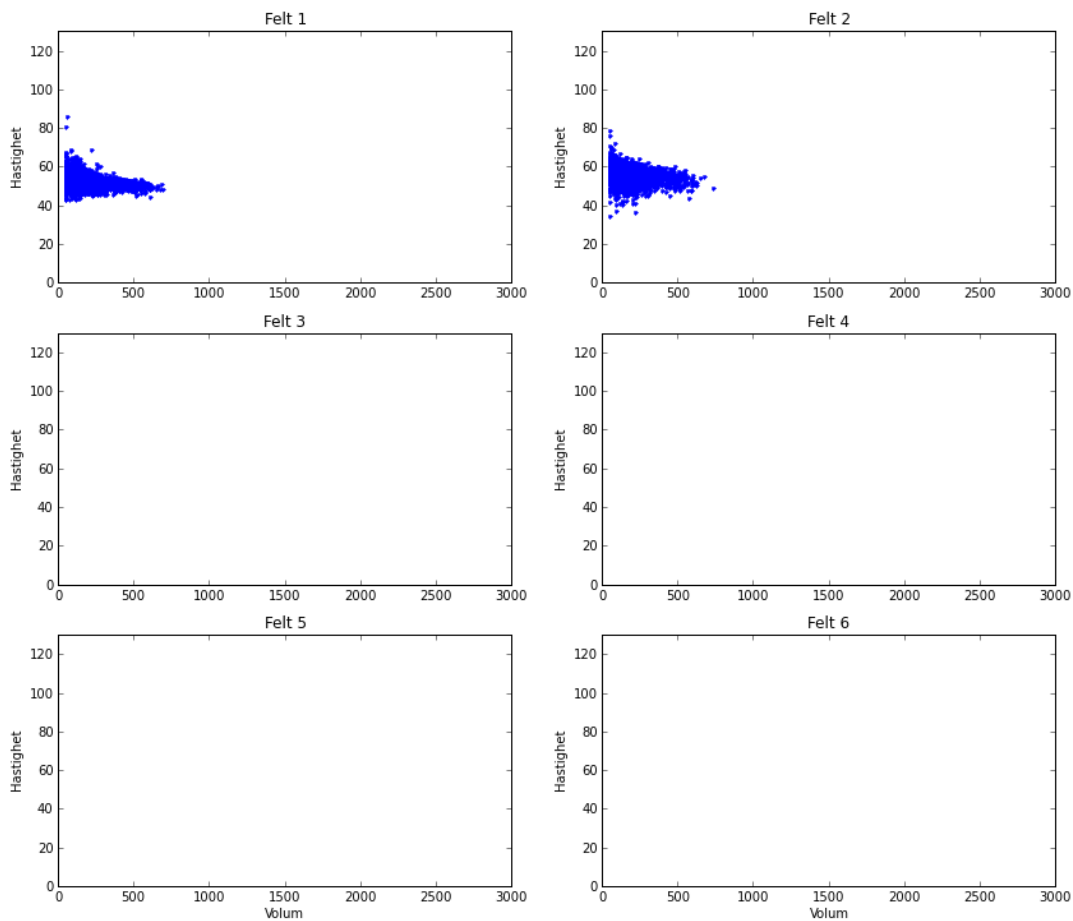
Registreringspunkt 1200112



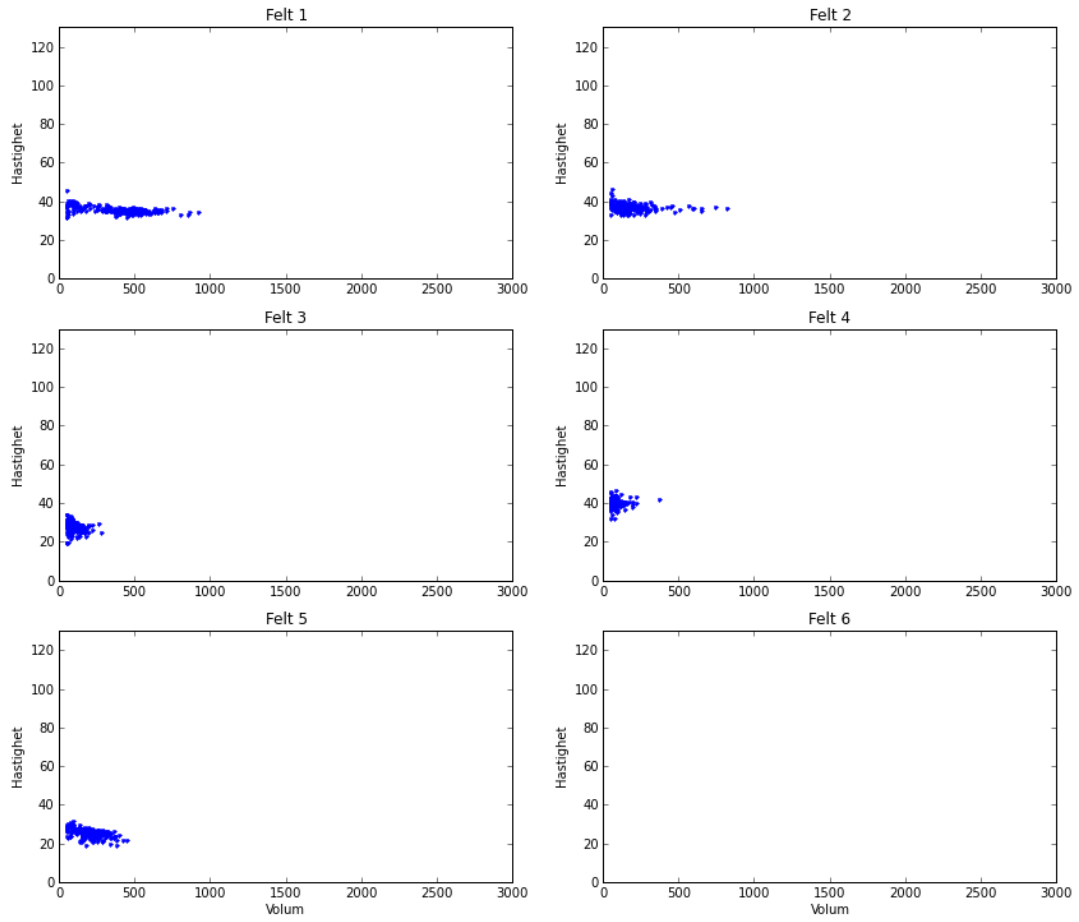
Registreringspunkt 1200113



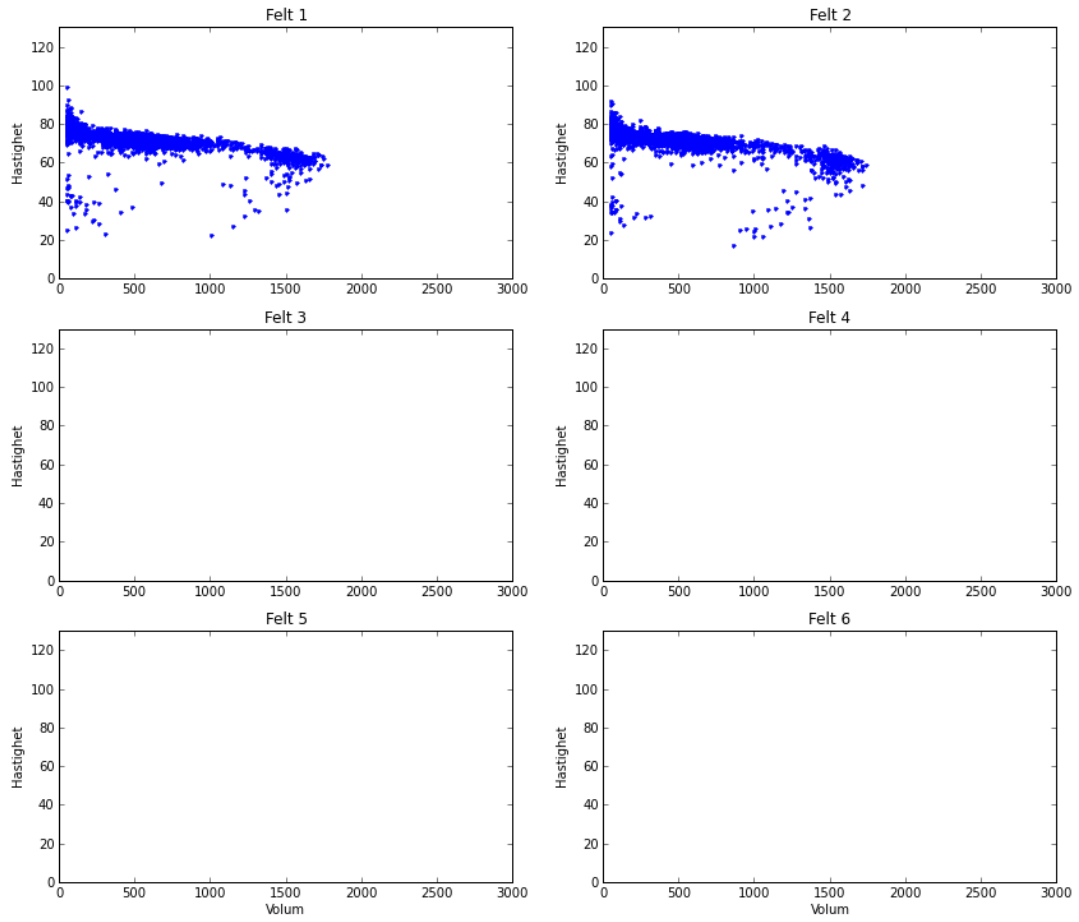
Registreringspunkt 1200127



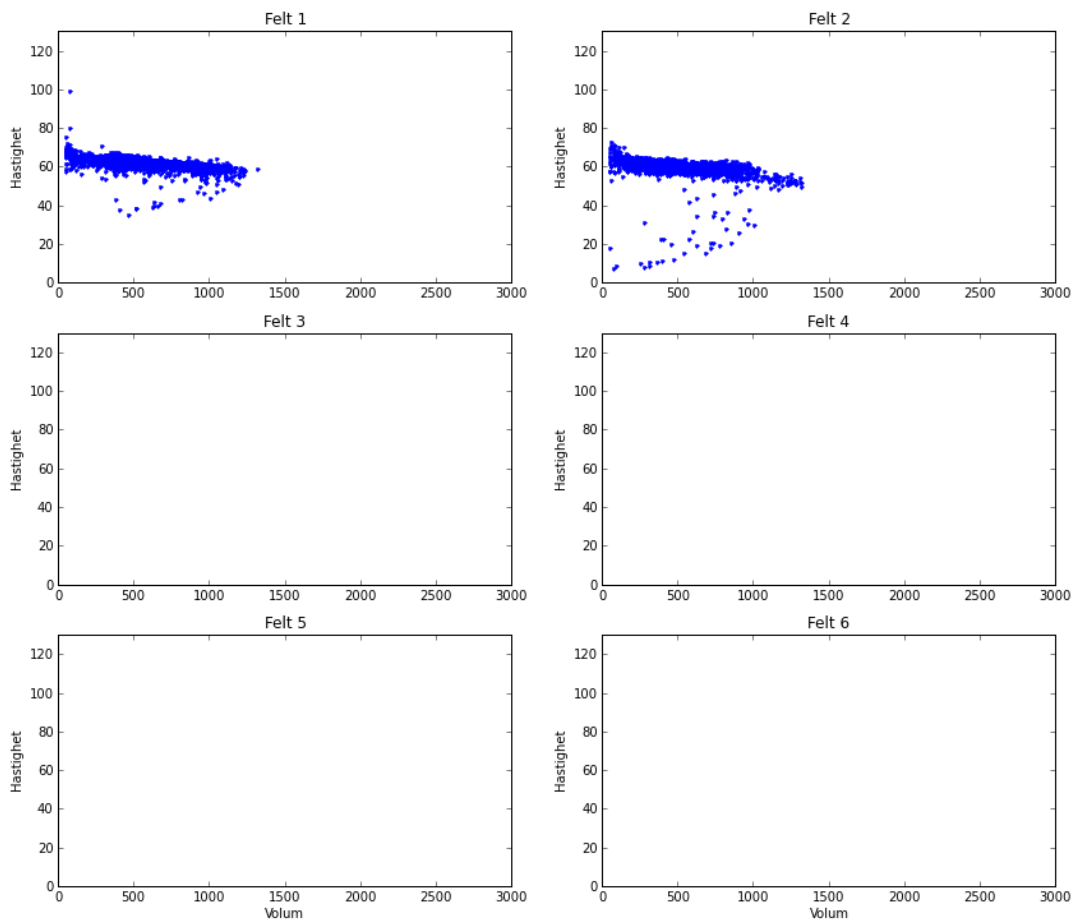
Registreringspunkt 1200129



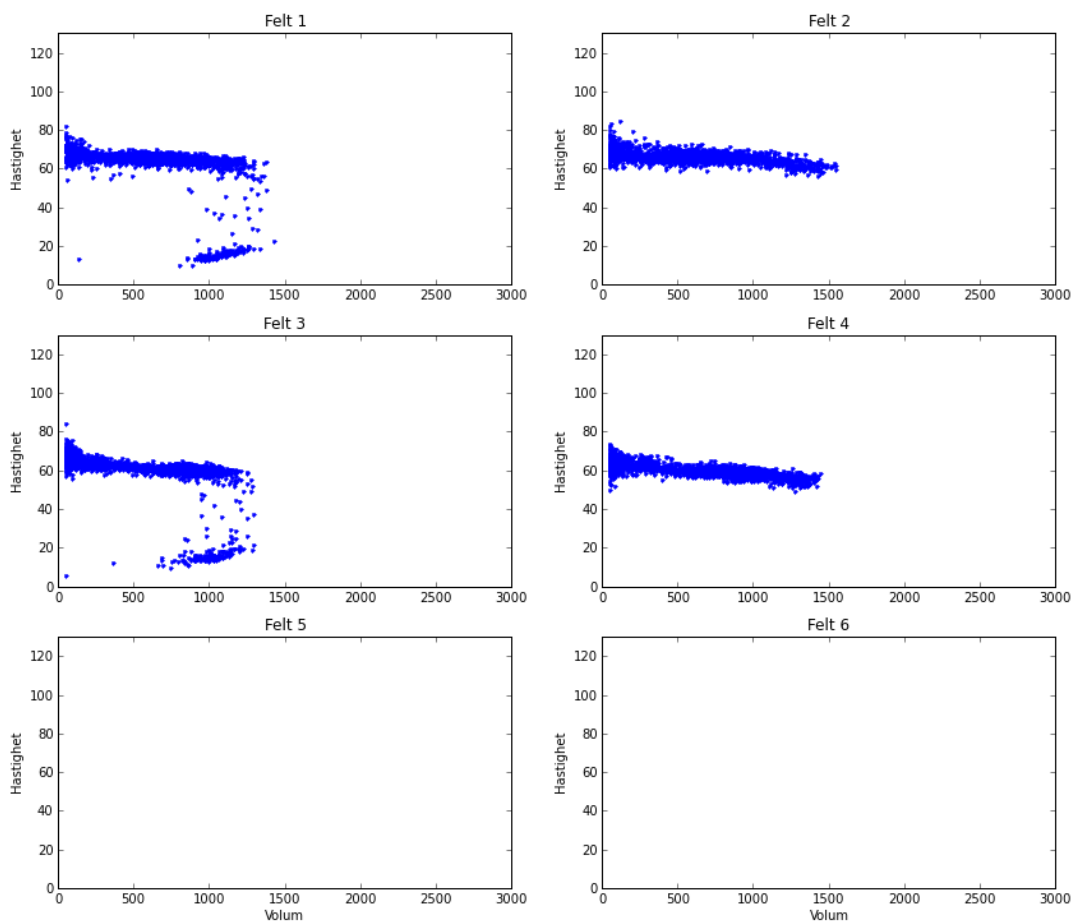
Registreringspunkt 1200234



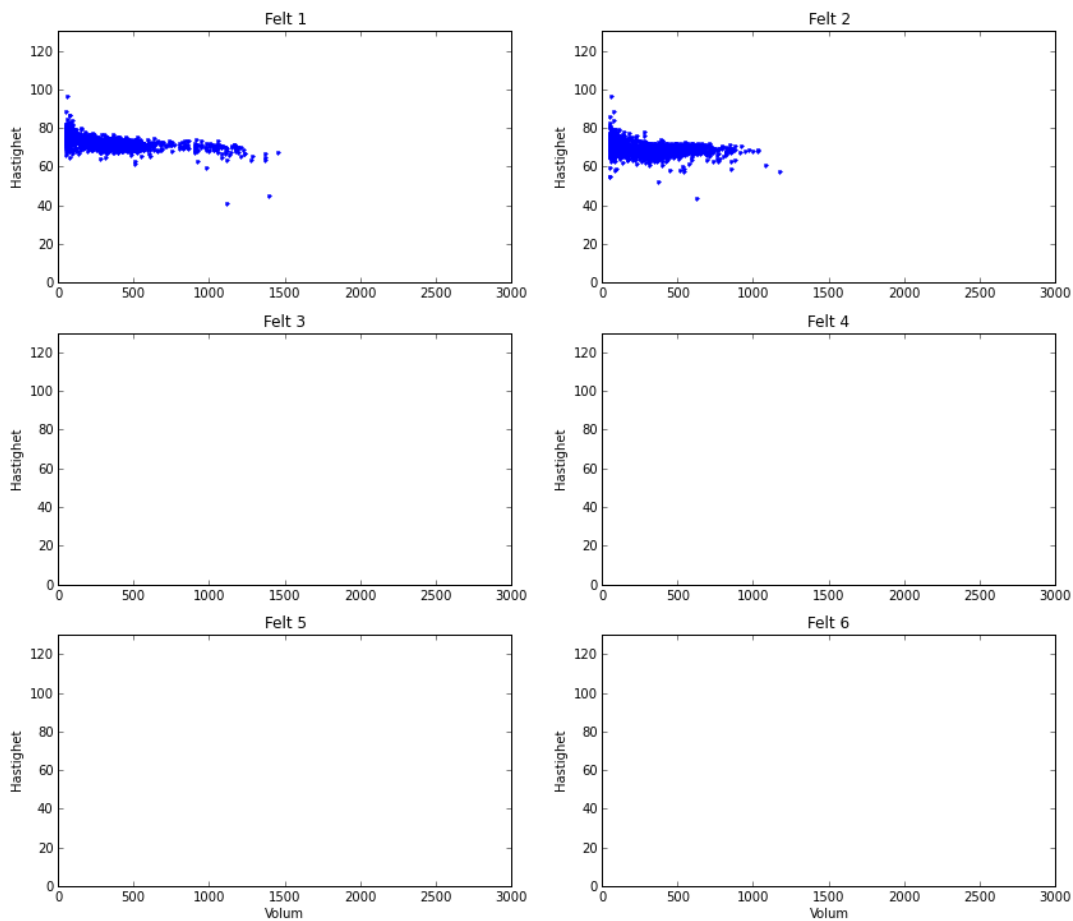
Registreringspunkt 1200277



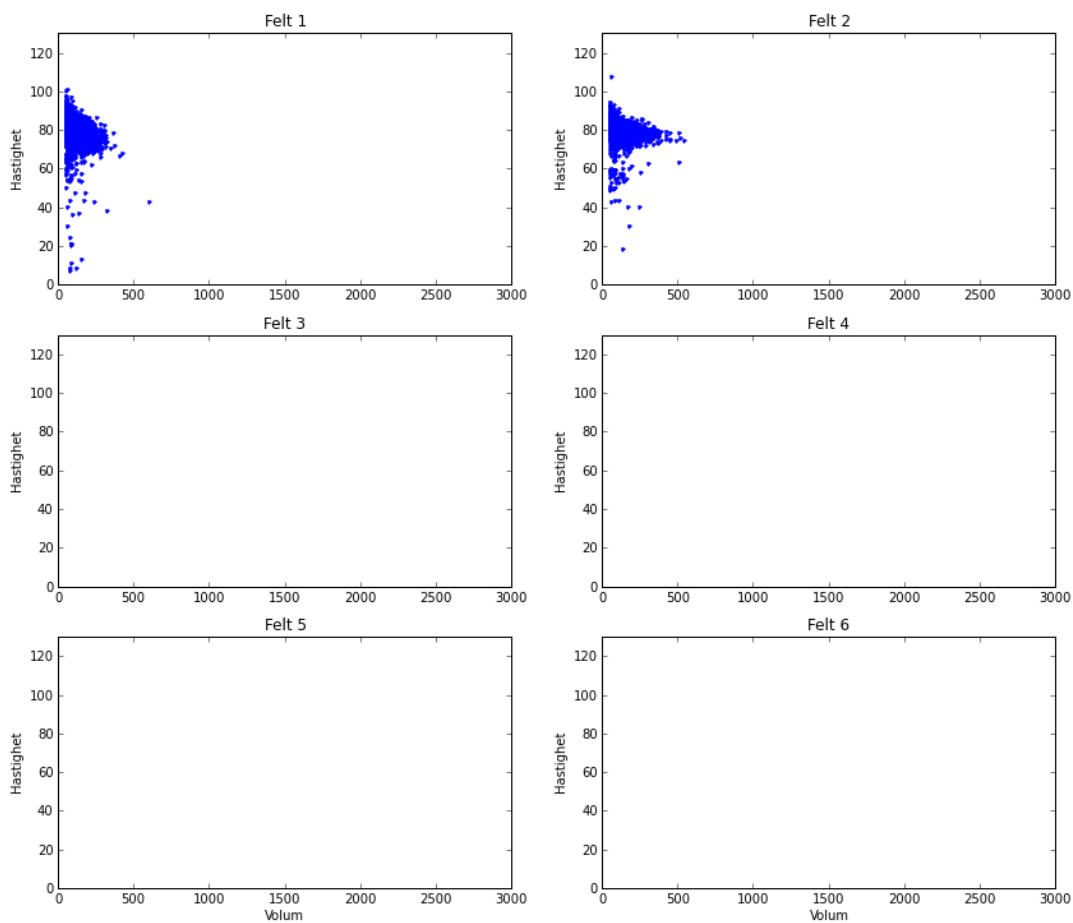
Registreringspunkt 1200381



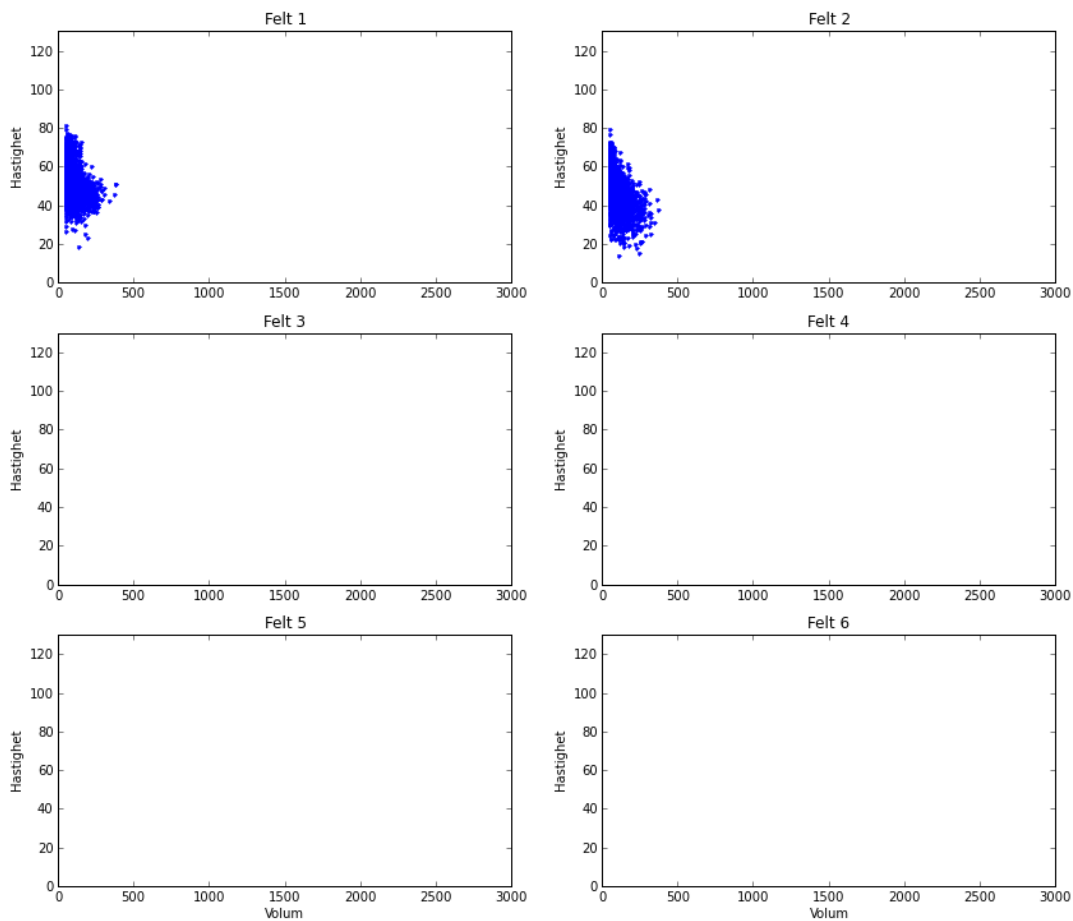
Registreringspunkt 1200413



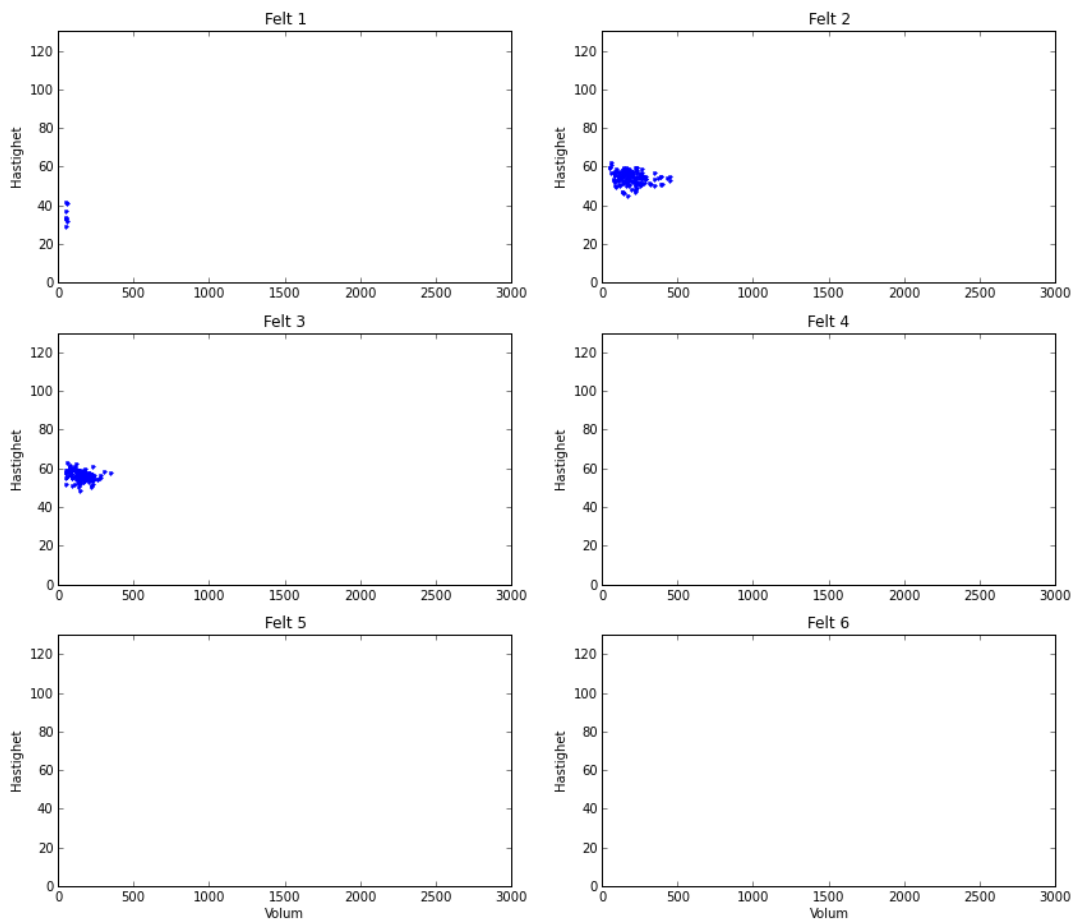
Registreringspunkt 1201151



Registreringspunkt 1201622



Registreringspunkt 1204365



Bruk av fartsmodell i TNext ved eksport til CUBE

Utdraget under viser hvordan eksportfunksjonen benytter seg av fartsmodellen på CUBE-lenker.

Parameter "fCl" representerer en ferdig CUBE-lenke.

```

    /// <summary>
    /// Finner og setter beregnet hastighet på en CUBE-lenke. Forutsetter at redigering allerede er aktiv.
    /// Tak ikke hensyn til inngangsfart/kryss eller utgangsfart/kryss/vikeplikt
    /// </summary>
    /// <param name="fCl"></param>
    /// <param name="rundkjoering"></param>
    /// <param name="fartsgrense"></param>
    /// <param name="lenkeType"></param>
    /// <param name="vegKat"></param>
    public virtual void finnOgSettLenkeFart(IFeature fCl, bool rundkjoering, int fartsgrense, int lenkeType,
String vegKat)
    {
        String errMsg = null;
        int fart = 0;
        // Fart i rundkjøring settes til til 20 km/t konstant for disse:
        if (rundkjoering)
        {
            fart = 20;
        }
        // Ellers er det bare hvis fartsgrense er 90 km/t eller lavere, at fart kan beregnes:
        else if (fartsgrense <= 90)
        {
            if (fartsgrense <= 0)
            {
                fartsgrense = TransportFunc.getDefaultSpeed(lenkeType, vegKat);
                TmAppFunc.cubeLenkeAnt_stdVerdier++;
            }
            // Kjøre fart beregnes med fartsmodell bare for kjørbare lenker:
            if ((lenkeType < VegLenkeTM.lenketype_bomlenke) || (lenkeType == VegLenkeTM.lenketype_buss))
            {
                double speedAB = Fartsmodell.beregn_lenke_hastighet(fCl.Shape as IPolyline, fartsgrense, ref
errMsg);

                fart = System.Convert.ToInt32(speedAB);
            }
            else
            {
                fart = fartsgrense;
            }
        }
        if (errMsg != null)
        {
            fCl.Store();
            GisApp.log(" Feil i beregning av hastighet på CUBE-lenke " + fCl.OID.ToString() + " : " + errMsg);
        }
        //
        // Lagre beregnet fart, hvis den finnes:
        if (fart > 0)
        {
            fCl.set_Value(fnrFmFart, fart);
            fCl.Store();
        }
    }
}

```

Utdrag fra modulen Fartsmodell.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using no.sintef.omr.util;
using ESRI.ArcGIS.Geometry;

namespace no.sintef.pro.t1
{
    /// <summary>
    /// Fartsmodell for personbil basert på Python-kode fra Tomas Levin 2013-01-11
    /// </summary>
    public class Fartsmodell
    {
        private static double MINIMUM_HASTIGHET = 5.0;
        private static double SEGMENT LENGDE = 30.48;
        private static double MINSTE_HOYDE = -100;
        private static double STORSTE_HOYDE = 5000;
    }
}
```

```

/// <summary>
/// Returnerer hastighet med geometrisk definisjonsretning.
/// Eventuell feil gir ikke avbrudd, men setter parameter errMsg.
/// </summary>
/// <param name="linje"></param>
/// <param name="fartsgrense"></param>
/// <returns></returns>
public static double beregn_lenke_hastighet(IPolyline linje, int fartsgrense, ref String errMsg)
{
    errMsg = null;
    if (fartsgrense <= 0)
    {
        errMsg = "Fartsgrense er ikke oppgitt";
        return -1;
    }
    //
    //Beregner hastighet på lenke:
    List<double> segmentHastigheter = new List<double>();
    try
    {
        if ((linje != null) && (!linje.IsEmpty))
        {
            {
                List<double[]> radiusStigningMeter = beregnRadiusStigningSegment(linje);
                for (int i = 0; i < radiusStigningMeter.Count; i++)
                {
                    double[] data = radiusStigningMeter[i];
                    double radius = data[0];
                    double stigning = data[1];
                    double meter = data[2];
                    double radius_hastighet = Fartsmodell.lamEtAl1999(radius, fartsgrense);
                    double gradient_hastighet = Fartsmodell.enkelStigningsmodell(stigning, fartsgrense);
                    double hastighet = Math.Min(gradient_hastighet, radius_hastighet);
                    //
                    // Undersøk minimumssvar. Feil i datagrunnlaget, valigvis udefinert z-verdi, gir slike
svar.
                    // Isåfall brukes det eventuelle delsvaret som er større enn minimum, eller eventuelt
fartsgrensen.
                    if ( Math.Abs(hastighet - Fartsmodell.MINIMUM_HASTIGHET) < 0.001)
                    {
                        //
                        // Hvis minst en av høydeverdiene være 0
                        if ((Math.Abs(linje.FromPoint.Z) < 0.001) || (Math.Abs(linje.ToPoint.Z) < 0.001))
                        {
                            if (radius_hastighet > Fartsmodell.MINIMUM_HASTIGHET + 0.001)
                            {
                                hastighet = radius_hastighet;
                            }
                            else if (gradient_hastighet > Fartsmodell.MINIMUM_HASTIGHET + 0.001)
                            {
                                hastighet = gradient_hastighet;
                            }
                            else
                            {
                                hastighet = fartsgrense;
                            }
                        }
                    }
                    segmentHastigheter.Add(hastighet);
                }
            }
        }
    }
    catch (Exception e)
    {
        errMsg = e.Message;
    }
    //
    // Hvis lenken er for kort til å inneholde noen segmenter, brukes fartsgrensen:
    if (segmentHastigheter.Count == 0)
    {
        segmentHastigheter.Add(fartsgrense);
    }
    //
    // Jevner ut fartsprofilen ved å beholde laveste hastigheter, og justerer ned omkringliggende hastigheter
    // med oppgitt akselerasjon/retardasjon:
    Fartsmodell.reduser_akselerasjon(segmentHastigheter, 1.0);
}

```

```

//
// Returner svaret som et gjennomsnitt:
return segmentHastigheter.Average();
}

/// <summary>
/// Beregner radius stigning og meterverdi for hele segmenter av linjstrengen.
/// Returnerer liste med [radius, stigning, meterverdi] for hvert lenkesegment.
/// </summary>
/// <param name="p1"></param>
/// <returns></returns>
private static List<double[]> beregnRadiusStigningSegment(IPolyline p1)
{
    List<double[]> resultat = new List<double[]>();
    double posTangent1 = 0.0;
    ILine tangent1 = null;
    do
    {
        //
        // Finn tangent for start segment:
        if (posTangent1 < 0.001)
        {
            tangent1 = new LineClass();
            p1.QueryTangent(esriSegmentExtension.esriNoExtension, posTangent1, false, 1, tangent1);
        }
        //
        // Finn tangent slutt segment / start neste segment:
        // Regner bare på hele segmenter!
        double posTangent2 = posTangent1 + Fartsmodell.SEGMENT LENGDE;
        if (posTangent2 > p1.Length)
        {
            break;
        }
        ILine tangent2 = new LineClass();
        p1.QueryTangent(esriSegmentExtension.esriNoExtension, posTangent2, false, 1.0, tangent2);
        //
        // Finn radius:
        double radius = 5000;
        double kursDiffRad = Math.Abs(MathFunc.angleDiffRadians(tangent1.Angle, tangent2.Angle));
        if (kursDiffRad > 0.0)
        {
            double r = SEGMENT LENGDE / kursDiffRad; // D.v.s. vinkel[rad] = buelengde / radius
            radius = Math.Min(radius, r);
        }
        //
        // Stigning:
        // Hvis urimelige verdier dukker opp, avbrytes hele beregningen.
        double z1 = tangent1.FromPoint.Z;
        if ((z1 < Fartsmodell.MINSTE_HOYDE) || (z1 > Fartsmodell.STORSTE_HOYDE))
        {
            resultat.Clear();
            break;
        }
        double z2 = tangent2.FromPoint.Z;
        if ((z2 < Fartsmodell.MINSTE_HOYDE) || (z2 > Fartsmodell.STORSTE_HOYDE))
        {
            resultat.Clear();
            break;
        }
        double gradient = 100 * (z2 - z1) / (posTangent2 - posTangent1);
        //
        // Middelpos:
        double meterSegmentSenter = (posTangent1 + posTangent2)/2;
        //
        // Oppdater liste:
        resultat.Add(new double[] { radius, gradient, meterSegmentSenter });
        //
        // Neste segment:
        //angleXY1 = angleXY2;
        tangent1 = tangent2;
        posTangent1 = posTangent2;
    } while (true);
    return resultat;
}

```



```

/// <summary>
/// Lam et al. fra 1999. Side 24 i farmodell studium
/// </summary>
/// <param name="radius"></param>
/// <param name="fartsgrense"></param>
/// <returns></returns>
private static double lamEtA11999(double radius, int fartsgrense)
{
    double fart85 = 95.594 - ((1.597 * 1746.38)/radius);
    if (fart85 > fartsgrense)
    {
        fart85 = fartsgrense;
    }
    if (fart85 < MINIMUM_HASTIGHET)
    {
        fart85 = MINIMUM_HASTIGHET;
    }
    return fart85;
}

/// <summary>
/// Enkel stigningsmodell basert på Franske modell, gjelder for tofelt med 5 meter bred veibane.
/// Hentet fra side 72 i Modelling Operating Speed TRB E-C151 Juli 2011.
/// Merk at positiv og negativ stigning gir samme resultat.
/// </summary>
/// <param name="gradient"></param>
/// <param name="fartsgrense"></param>
/// <returns></returns>
private static double enkelStigningsmodell(double gradient, int fartsgrense)
{
    double fart85 = 92 - (0.31 * Math.Pow(gradient,2) );
    if (fart85 > fartsgrense)
    {
        fart85 = fartsgrense;
    }
    if (fart85 < MINIMUM_HASTIGHET)
    {
        fart85 = MINIMUM_HASTIGHET;
    }
    return fart85;
}

/// <summary>
/// Return minimum angle difference,
/// i.e. return difference as a positive value between 0 and PI
/// </summary>
/// <param name="radAngle1"></param>
/// <param name="radAngle2"></param>
/// <returns></returns>
public static double angleDiffRadians(double radAngle1, double radAngle2)
{
    double diff = 0.0;
    if (radAngle1 > radAngle2)
    {
        swapDouble(ref radAngle1, ref radAngle2);
    }
    diff = radAngle2 - radAngle1;
    if (diff > Math.PI)
    {
        diff = (2 * Math.PI) - (radAngle2 - radAngle1);
    }
    return diff;
}

```



Teknologi for et bedre samfunn

www.sintef.no