

Estimerte ekvipasjepasseringer fra 74T prøveordningen

Anders V. Stubberud

21. desember 2024

Innhold

Figurer	i
Tabeller	i
1 Intensjon	1
2 Premisser	1
2.1 Nedre grense for radius av registreringer	1
2.2 Øvre grense for radius av registreringer	3
2.3 Unngå duplikate registreringer	4
3 Fremgangsmåte	5
4 Resultater	7

Figurer

1	Illustrasjon brukt for å begrunne valg av radius for registreringer	2
2	Histogram over loggførte hastigheter	2
3	Histogram og kumulativ fordeling av tidsintervallene mellom etterfølgende logginger . . .	3
4	Veienes definerte koordinatpunkter sammen med en radius på 4.375km	4
5	Implementasjonen som lager resultat-tabellene	6

Tabeller

1	Deskriptiv statistikk over loggførte hastigheter	5
2	Resultater ved å benytte loggingene fra lastebilene	7
3	Resultater ved å benytte loggingene fra tilhengerne	7

1 Intensjon

Dette dokumentet har til hensikt å presentere en oversikt over antallet passeringer ekvipasjene fra 74T prøveordningen har gjort på et utvalg veger der bæreevnedata for både vår og sommer 2023-24 er tilgjengelig, mer spesifikt: Fv24 S7, Fv24 S8, Fv227 S2, Fv1796 S2, Fv1798 S2, Fv1820 S1, Fv1844 S1, og Fv1900 S1. Ettersom en veg kan strekke langt, så er det her tatt utgangspunkt i antall passeringer av forhåndsdefinerte koordinatpunkter som representativer for vegene. Koordinatpunktene er hentet fra Vegkart.

2 Premisser

Datagrunnlag

Grunnlaget for denne besvarelsen er posisjonsdata innhentet i forbindelse med prøveordningen for økt tillatt totalvekt i tømmer/tungtransportindustrien. Denne dataen består av regelmessige loggninger fra prøveordningens kjøretøy, hvor verdier for VIN, Organisasjonsnummer, Dato, Breddegrad, Lengdegrad, Høyde, Retning, Hastighet, Temperatur, Aktuell vekt, Akselvekt foran, Akselvekt bak, Drivstoffs nivå (%), Catalystrnivå (%), Dekkmåling, Hard akselerasjon, Hard bremsing, og Hard sving er registrert. Til denne oppgaven er enkelte verdier mer relevante enn andre.

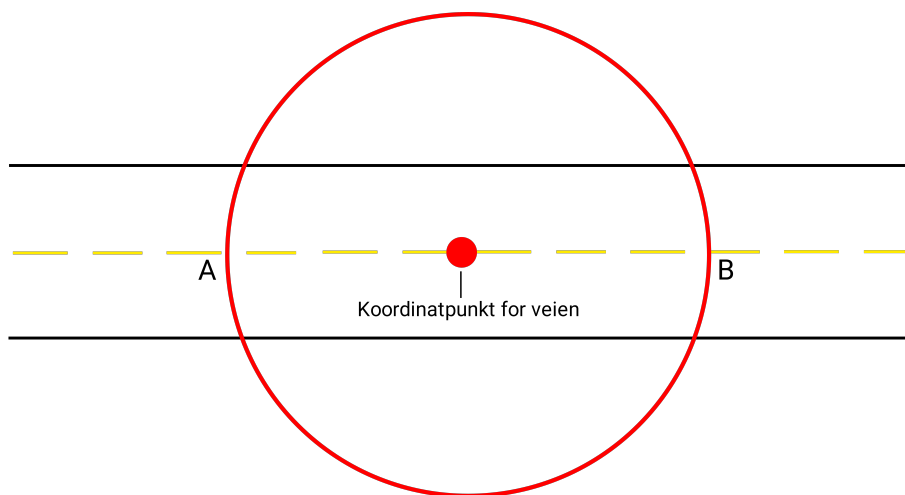
Begrensninger i dataen og tilnærminger for å overkomme disse

Intensjonen er som nevnt å sette opp en oversikt over antall passeringer prøveordningens kjøretøy har gjort på punktene som representerer veiene. Ettersom loggningen av data fra prøveordningen ikke har noen tilknytning til disse punktene og dermed ikke direkte er representativ for antallet passeringer, så er en mulig tilnærming å anse loggninger innen en viss radius fra veienes definerte punkter som passeringer av veiene. Dette reiser naturligvis spørsmålet av hvor stor denne radiusen skal være. Ved denne tilnærmingen står man imidlertid i fare for å telle den samme passeringen flere ganger; dersom kjøretøyet på den samme turen foretar flere loggninger innen området dekket av radiusen, vil samtlige av disse kunne tatt med i tellingen, til tross for at det kun forekommer én passering. For å unngå dette, kan man basere seg på å kun telle loggninger fra en gitt kombinasjon av kjøretøy og vei dersom det har gått en viss tid siden forrige logging for denne kombinasjonen. Effekten av denne tidsperioden er dermed at sekvensielle loggninger ikke telles med flere ganger, og dermed at neste registrering trolig tilhører en annen passering. Dette gir grunnlag for enda et spørsmål, nemlig hvor lang perioden skal være.

Dataen består av individuelle loggninger på bilene og tilhengerne. Videre er det noe avvik i loggningen fra bilene og deres respektive tilhengere, eksempelvis at enkelte biler har logget data i lengre perioder der deres tilhørende tilhenger ikke har noen loggninger i det hele tatt. Dette skaper en komplikasjon rundt hvilke datapunkter man skal ta hensyn til: kun de som stammer fra bilene, kun tilhengerne, eller der det er registrert for både bil og tilhenger? Ettersom bilene har mulighet til å kjøre uten tilhengerne, men ikke motsatt, så er det her hovedsaklig tatt utgangspunkt i datapunktene tilhørende tilhengerne. Dermed er plots, statistikk, og argumentasjon for premisser basert på dataen fra tilhengerne. Resultat-seksjonen inneholder dog også en tabell basert på dataen logget utelukkende på bilene, som er beregnet på samme måte som for tilhengerne.

2.1 Nedre grense for radius av registreringer

Radiusen rundt det konkrete koordinatpunktet en vei er definert på som benyttes til å registrere passeringer må være stor nok til at tidsintervallene mellom loggingene ikke fører til at et kjøretøy som kjører på en av veiene av interesse har mulighet til å passere gjennom hele dette området uten at det forekommer minst én logging. Figur 1 forsøker å illustrere dette: Et kjøretøy som passerer koordinatpunktet en vei er definert på ender med å logge data på punkter A og B, som et resultat av et gitt tidsintervall mellom loggingene og en gitt hastighet. Dersom radiusen rundt koordinatpunktet til veien er for liten til å fange opp noen av disse registreringene, vil passeringen ikke bli registrert.

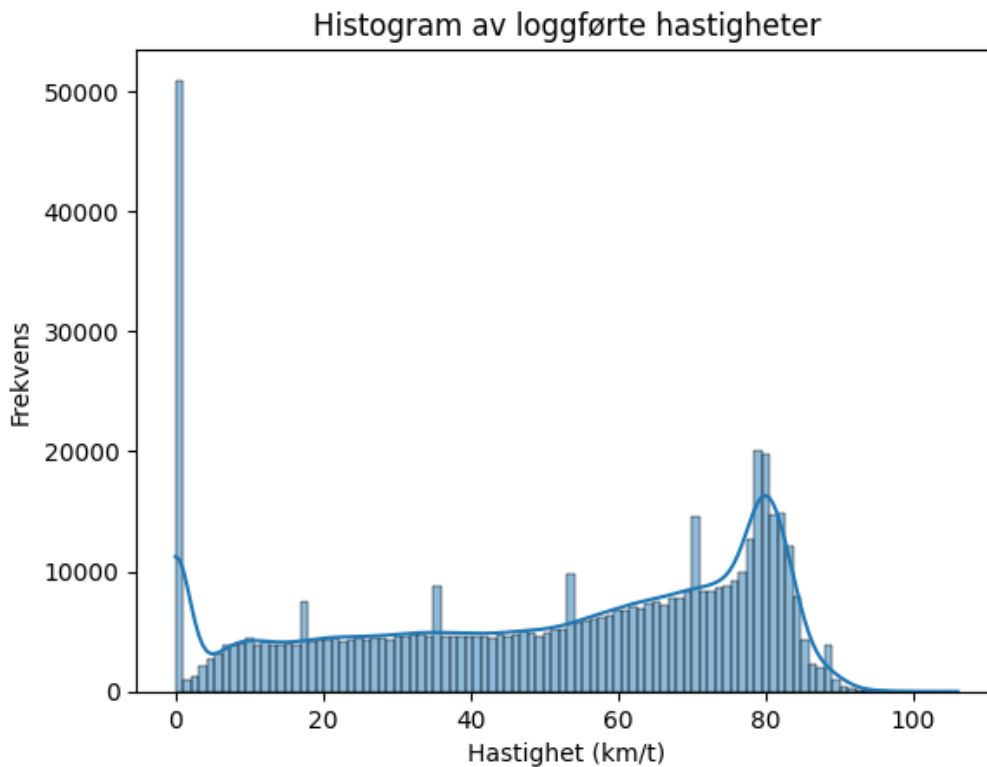


Figur 1: Illustrasjon brukt for å begrunne valg av radius for registreringer

Forholdet mellom strekning s , hastighet v , og tid t , $s = v \times t$, tilsier dermed at radiusen r som en funksjon av kjøretøyets hastighet v og tidsintervallet Δt mellom to påfølgende logginger må (eller hvertfall burde) være $r \geq (v \times \Delta t)/2$.

Hastighet

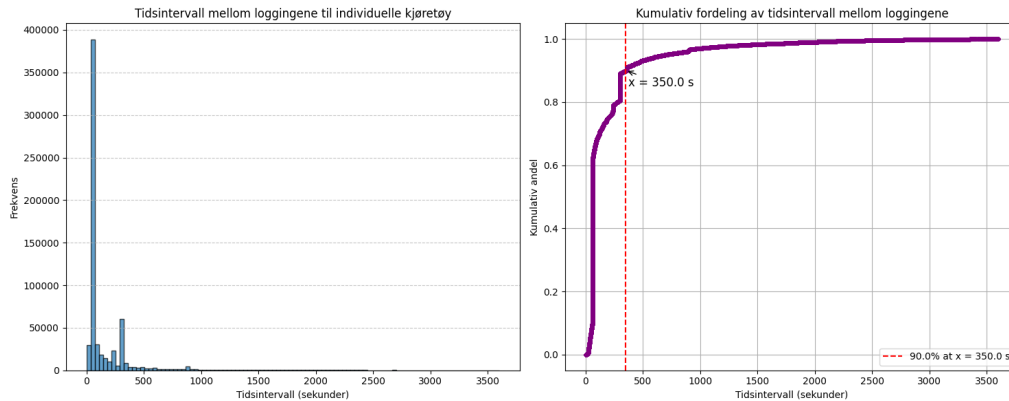
Desto større verdi for hastigheten v man plusser inn i formelen, desto større radius får man, og desto flere logginger kan bli registrert som en passering av en vei. Ved å sette inn en hastighet som representerer det øvre sjiktet av hastighetene som er loggført, legger man dermed til rette for at de fleste reelle passeringer av koordinatpunktet til en vei kan bli registrert. Som man ser i Figur 2 (som viser fordelingen av alle loggførte hastigheter i datasettet) er det tilnærmet ingen registrerte hastigheter over **90 km/t**, så det er denne verdien som har blitt brukt for hastighet.



Figur 2: Histogram over loggførte hastigheter

Tidsintervall

Tidsintervallene mellom loggingene spiller også en sentral rolle i å bestemme radiusen. Et større tidsintervall fører til en større distanse mellom loggingene, hvilket igjen forutsetter en romsligere radius for å fange opp passeringer. I likhet med verdien for hastighet, så vil en verdi fra det øvre sjiktet av tidsintervall mellom logginger føre til en radius som fanger opp de fleste reelle passeringer. Figur 3 viser i venstre subplot et histogram over fordelingen av tidsintervallene mellom etterfølgende logginger innad i samme kjøretur for individuelle kjøretøy. Høyre subplot viser den samme dataen som en kumulativ fordeling, hvor det også kommer tydelig frem man kan ta hensyn til majoriteten av tidsintervallene (90%) ved å basere seg på å benytte en verdi av **rundt 6 minutter** (350 sekunder). Dermed har denne verdien blitt brukt som input for tidsintervallene.



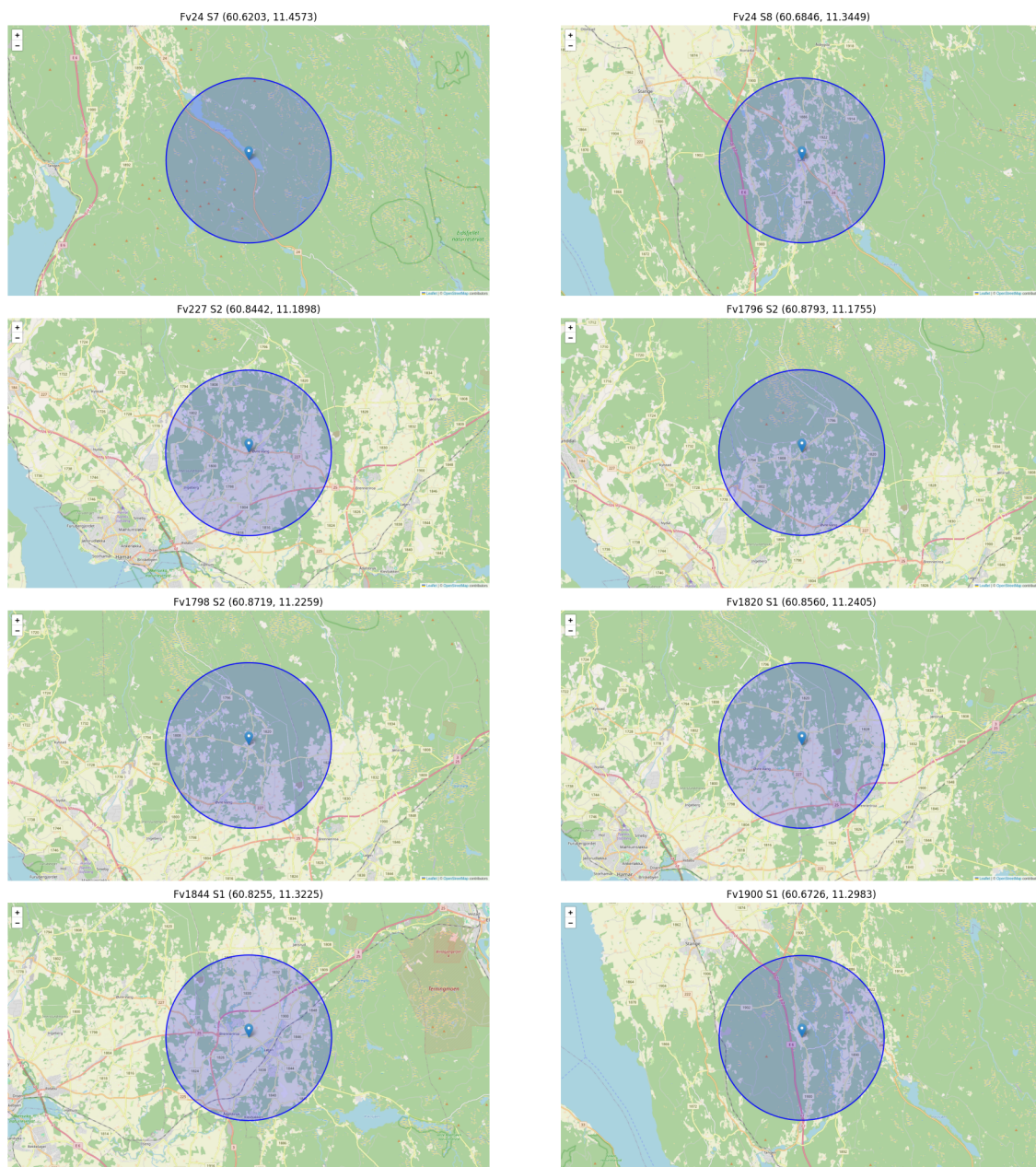
Figur 3: Histogram og kumulativ fordeling av tidsintervallene mellom etterfølgende logginger

Konklusjon nedre grense for radius av registreringer

Med verdier tilhørende det øvre sjiktet av hastighet og tidsintervaller, så burde den resulterende radiusen være romslig nok til å fange opp de aller fleste reelle passeringer. Plugger man inn verdiene for hastighet og tidsintervall, ender man dermed med en nedre grense for radiusen på $r = (v \times \Delta t)/2 = (90 \text{ km/h} \times 350 \text{ s}/60^2)/2 = 4.375 \text{ km}$

2.2 Øvre grense for radius av registreringer

Mens den nedre grensen for radiusen sørger for at de fleste reelle passeringer blir plukket opp, så forsøker den øvre grensen å unngå at passeringer på nærliggende veier blir registrert som en passering av en av de aktuelle veiene, som et resultat av at radiusen også berører disse. Figur 4 viser de aktuelle vei- enes definerte koordinatpunkter med en radius på 4.375km rundt seg - her ser man allerede tendenser til at også andre veier berøres av radiusen, så en særlig romsligere radius vil trolig føre til mindre presise resultater.



Figur 4: Veienes definerte koordinatpunkter sammen med en radius på 4.375km

2.3 Unngå duplikate registreringer

Ettersom logger innenfor radiusen registreres som passeringer av veien, så står man i fare for at flere logger fra samme tur kan bli telt som flere individuelle passeringer. For å unngå dette, så har kun én logging for hvert VIN-nummer på en spesifikk vei innenfor et gitt tidsintervall blitt telt med. Tidsintervallet burde følgelig være langt nok til å sørge for at selv saktekjørende kjøretøy rekker å passere gjennom hele området dekket av radiusen. Tabell 1 viser deskriptiv statistikk over de loggførte hastighetene, hvor det kommer frem at den nedre kvartilen har en hastighet på 26 km/t . Egen intuisjon tilsier også at lavere hastigheter enn dette under reell kjøring på de aktuelle veiene er lite sannsynlig. Ettersom det også er lite sannsynlig at kjøretøyene regelmessig endrer kjøreretning rundt koordinatpunktene for de aktuelle veiene, så virker det ikke som at man introduserer en særlig risiko ved å benytte det forholdsvis lange tidsintervallet som følger av en såpass lav hastighet. Dermed er dette tidsintervallet satt til $t = (2 \times r) / v = (2 \times 4.375 \text{ km}) / 26 \text{ km/t} \approx 20 \text{ min}$

	Hastighet
count	561468.000000
mean	49.377107
std	27.794024
min	0.000000
25%	26.000000
50%	56.000000
75%	75.000000
max	106.000000

Tabell 1: Deskriptiv statistikk over loggførte hastigheter

3 Fremgangsmåte

Grovt oppsummert ble dermed fremgangsmåten for å telle antallet passeringer fra prøveordningen som følger:

1. For hver av de aktuelle veiene går man over hvert loggførte datapunkt
2. Dersom distansen mellom veien og datapunktet er mindre eller lik radiusen på 4.375km og ikke fører til en duplikat telling, så inkrementeres antall passeringer for ekvipasjen kjøretøyet som logget datapunktet tilhører.

Mer konkret er dette koden som har implementert de nevnte forutsetningene, og som følgelig produserer resultatene:

```

1 def table(
2     df: pd.DataFrame,
3     road_coordinates: dict[str, tuple[float, float]],
4     threshold_radius_km: float,
5     threshold_time_hours: float
6 ) -> pd.DataFrame:
7     """
8     Generates a table with counts of registrations per road, tonnage, and year
9     based on proximity and time criteria.
10
11     Args:
12         df (pd.DataFrame): DataFrame containing VIN, Latitude, Longitude, Date, and
13             Tonnage columns. Sorted by date, ascending.
14         road_coordinates (dict): A dictionary mapping road names to (latitude,
15             longitude) tuples.
16         threshold_radius_km (float): Maximum distance in kilometers to consider a
17             registration near a road.
18         threshold_time_hours (float): Minimum time in hours to avoid counting
19             duplicate registrations.
20
21     Returns:
22         pd.DataFrame: A summary DataFrame with counts of registrations per road,
23             year, and tonnage.
24     """
25
26     df['Dato'] = pd.to_datetime(df['Dato'])
27
28     columns = [
29         "Vei",
30         *[f"{year} {tonnage}t" for year in years for tonnage in tonnages]
31     ]
32
33     result = {col: [] for col in columns}
34
35     VINs = df["VIN"].unique().tolist()
36
37     for road, (road_lat, road_lon) in road_coordinates.items():
38         most_recent_entry = {VIN: None for VIN in VINs}
39         road_counts = {f"{year} {tonnage}t": 0 for year in years for tonnage in
40             tonnages}
41
42         for _, entry in df.iterrows():
43             entry_coords = (entry["Latitude"], entry["Longitude"])
44             distance = geodesic(entry_coords, (road_lat, road_lon)).km
45
46             if distance <= threshold_radius_km:
47                 VIN = entry["VIN"]
48                 entry_time = entry["Dato"]
49
50                 if most_recent_entry[VIN] is None or (entry_time -
51                     most_recent_entry[VIN]).total_seconds() / 3600 >
52                     threshold_time_hours:
53                     most_recent_entry[VIN] = entry_time
54
55                 year = entry["Dato"].year
56                 tonnage = entry["Tonnage"]
57                 key = f"{year} {tonnage}t"
58                 if key in road_counts:
59                     road_counts[key] += 1
60
61         result["Vei"].append(road)
62         for key, value in road_counts.items():
63             result[key].append(value)
64
65     return pd.DataFrame(result)

```

Figur 5: Implementasjonen som lager resultat-tabellene

4 Resultater

Under premissene og ved fremgangsmåten beskrevet ovenfor, er dette resultatene:

	Vei	2021 60t	2021 65t	2021 68t	2021 74t	2022 60t	2022 65t	2022 68t	2022 74t	2023 60t	2023 65t	2023 68t	2023 74t	2024 60t	2024 65t	2024 68t	2024 74t	Total
0	Fv24 S7	6	10	0	205	7	22	27	655	7	67	442	931	5	83	114	384	2965
1	Fv24 S8	52	21	0	806	130	143	621	1818	117	149	1603	2681	29	132	597	1378	10277
2	Fv227 S2	196	34	0	140	414	69	748	305	401	115	295	726	106	47	138	290	4024
3	Fv1796 S2	111	0	0	104	231	0	652	266	123	2	258	654	16	2	117	286	2822
4	Fv1798 S2	105	0	0	188	217	1	648	350	122	2	302	758	14	2	126	300	3135
5	Fv1820 S1	263	39	0	453	490	93	876	1042	442	163	520	1284	112	65	205	513	6560
6	Fv1844 S1	384	70	0	702	743	271	672	1400	625	382	634	1962	194	185	369	1208	9801
7	Fv1900 S1	83	22	0	914	188	202	685	1942	132	178	1775	3015	45	146	712	1648	11687
Total	NaN	1200	196	0	3512	2420	801	4929	7778	1969	1058	5829	12011	521	662	2378	6007	51271

Tabell 2: Resultater ved å benytte loggingene fra lastebilene

	Vei	2021 60t	2021 65t	2021 68t	2021 74t	2022 60t	2022 65t	2022 68t	2022 74t	2023 60t	2023 65t	2023 68t	2023 74t	2024 60t	2024 65t	2024 68t	2024 74t	Total
0	Fv24 S7	0	0	0	166	0	0	0	549	0	35	21	577	9	18	57	163	1595
1	Fv24 S8	0	0	0	439	0	0	0	1262	9	47	89	1484	100	90	519	486	4525
2	Fv227 S2	0	0	0	107	1	0	0	228	3	25	1	338	237	45	123	124	1232
3	Fv1796 S2	0	0	0	87	0	0	0	217	0	1	0	322	45	0	98	117	887
4	Fv1798 S2	0	0	0	129	0	0	0	286	0	1	0	373	51	0	113	145	1098
5	Fv1820 S1	0	0	0	238	1	0	0	538	13	35	11	535	295	99	214	245	2224
6	Fv1844 S1	0	0	0	462	1	0	0	1115	21	77	21	1046	387	155	286	453	4024
7	Fv1900 S1	0	0	0	555	0	0	0	1482	9	47	105	1641	105	102	544	619	5209
Total	NaN	0	0	0	2183	3	0	0	5677	55	268	248	6316	1229	509	1954	2352	20794

Tabell 3: Resultater ved å benytte loggingene fra tilhengerne