

# Estimerte ekvipasjepasseringer fra 74T prøveordningen

Anders V. Stubberud

16. januar 2025

## Innhold

<b>Figurer</b>	i
<b>Tabeller</b>	i
<b>1 Intensjon</b>	1
<b>2 Premisser</b>	1
2.1 Lengde og bredde på området for registrering . . . . .	1
2.2 Unngå duplike registreringer . . . . .	8
<b>3 Fremgangsmåte</b>	8
<b>4 Resultater</b>	10

## Figurer

1 Illustrasjon brukt for å begrunne valg av lengde av område for registreringer . . . . .	2
2 Histogram over loggførte hastigheter . . . . .	2
3 Histogram og kumulativ fordeling av tidsintervallene mellom etterfølgende logginger . . . . .	3
4 Implementasjonen som lager resultat-tabellene . . . . .	9

## Tabeller

1 Deskriptiv statistikk over loggførte hastigheter . . . . .	8
2 Resultater ved å benytte loggingene fra lastebilene . . . . .	10
3 Resultater ved å benytte loggingene fra tilhengerne . . . . .	10

## 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, Akselvikt foran, Akselvikt bak, Drivstoffsnivå (%), Catalystnivå (%), Dekkmåling, Hard akselerasjon, Hard bremsing, og Hard sving er registrert. Til denne oppgaven er enkelte verdier mer relevante enn andre.

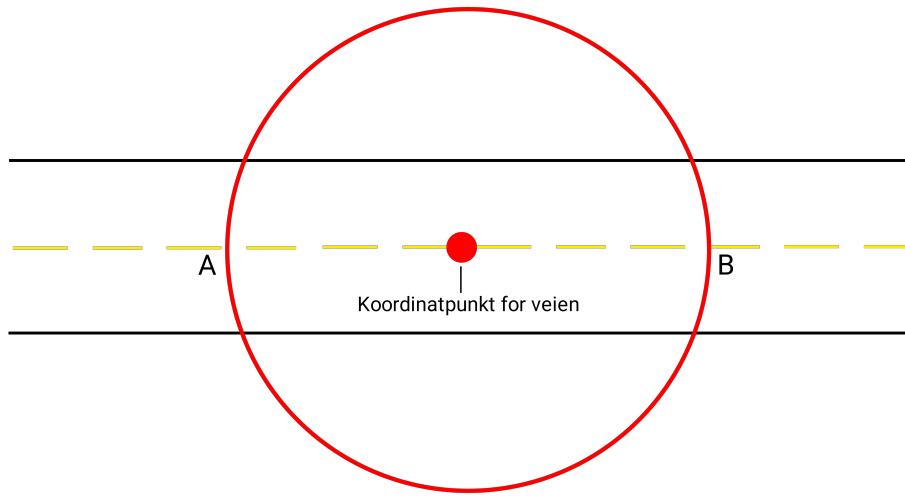
### Begrensninger i dataen og tilnærming for å overkomme disse

Intensjonen er som nevnt å sette opp en oversikt over antall passeringer prøveordningens kjøretøy har gjort på punktene som representerer veiene. Ettersom loggingen 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 sette opp et område rundt vegstrekningen hvor punktene befinner seg, for deretter å sjekke hvor mange loggninger som er foretatt innenfor disse områdene. Ved denne tilnærmingen står man imidlertid i fare for å telle den samme passeringen flere ganger; dersom kjøretoyet på den samme turen foretar flere loggninger innen området, 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 logginger 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. Intensjonen her er dermed å avgrense lengden på området til at et kjøretøy som passerer gjennom det etter all sannsynlighet har foretatt minst én logging i det. Et lengre område vil øke risikoen for at flere loggninger innenfor området tilhørende samme passering kan bli telt som individuelle passeringer.

Dataen består av individuelle loggninger på bilene og tilhengerne. Videre er det noe avvik i loggingen 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 Lengde og bredde på området for registrering

Området 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 området rundt koordinatpunktet til veien er for liten til å fange opp noen av disse registreringene, vil passeringen ikke bli registrert. Presisjonen til posisjonsloggingene er av såpass god kvalitet at bredde på området ikke er et problem, og bredde er dermed satt til 50 meter.

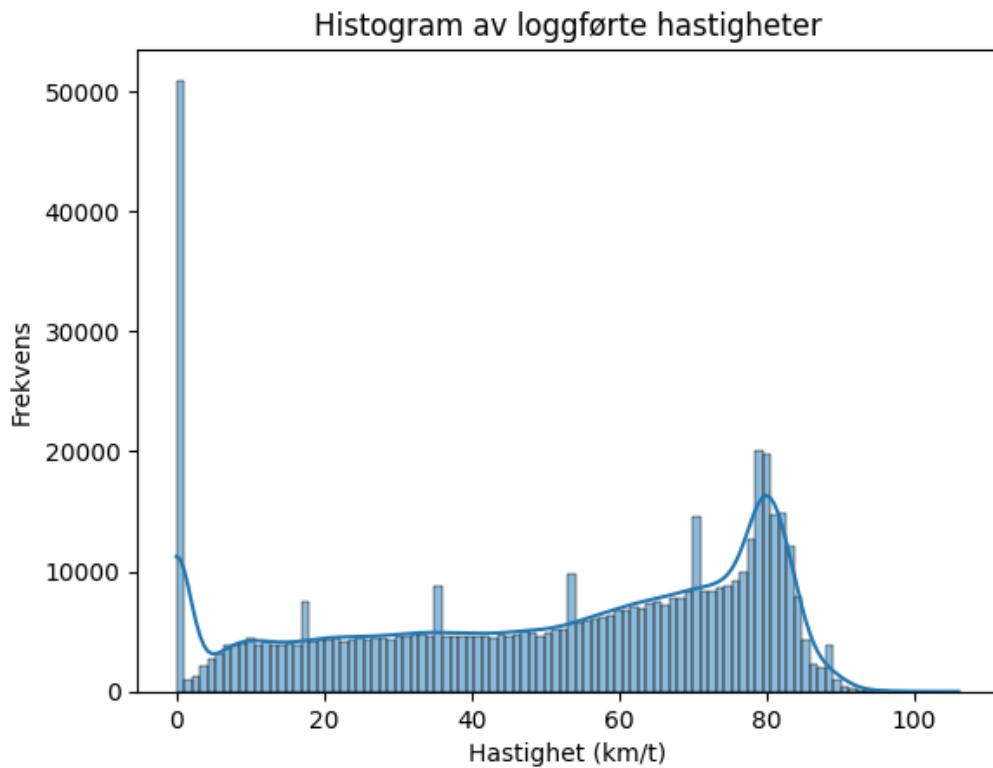


**Figur 1:** Illustrasjon brukt for å begrunne valg av lengde av område for registreringer

Forholdet mellom strekning  $s$ , hastighet  $v$ , og tid  $t$ ,  $s = v \times t$ , tilslør dermed at lengden  $s$  som en funksjon av kjøretøyets hastighet  $v$  og tidsintervallet  $\Delta t$  mellom to påfølgende logger må (eller hvertfall burde) være  $s \geq (v \times \Delta t)/2$  i hver retning fra punktet.

### Hastighet

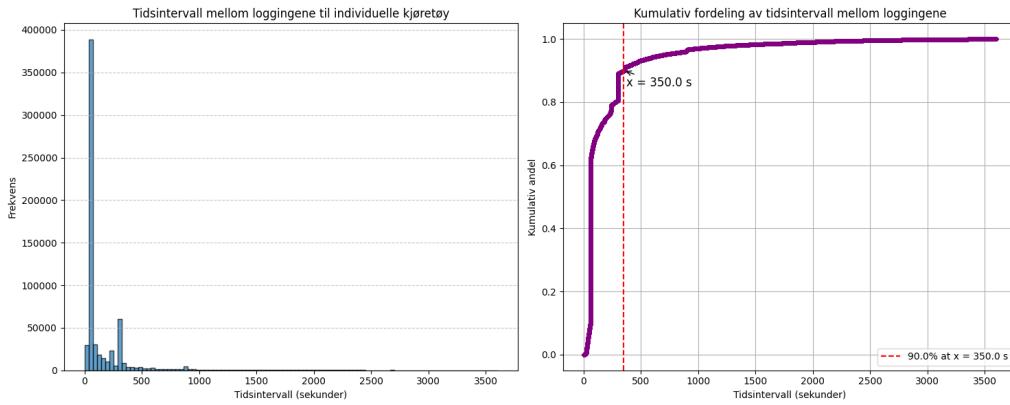
Desto større verdi for hastigheten  $v$  man pluggar inn i formelen, desto lengre område får man, og desto flere logger kan bli registrert som en passering av en vei. Ved å sette inn en hastighet som representerer det øvre sjiktet av hastighetene som er loggført, legger man dermed til rette for at de 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

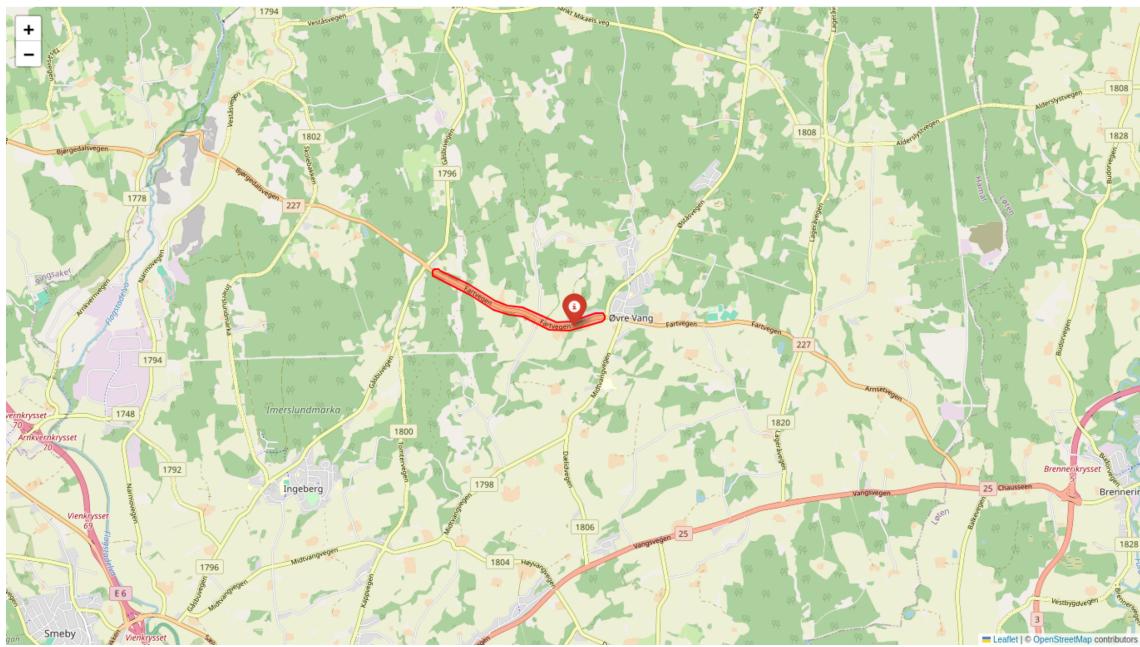
Tidsintervallene mellom loggingene spiller også en sentral rolle i å bestemme lengden av området. Et større tidsintervall fører til en større distanse mellom loggingene, hvilket igjen forutsetter et lengre område for å fange opp passeringer. I likhet med verdien for hastighet, så vil en verdi fra det øvre sjiktet av tidsintervall mellom loggingene føre til et område 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 kjøreretur 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.



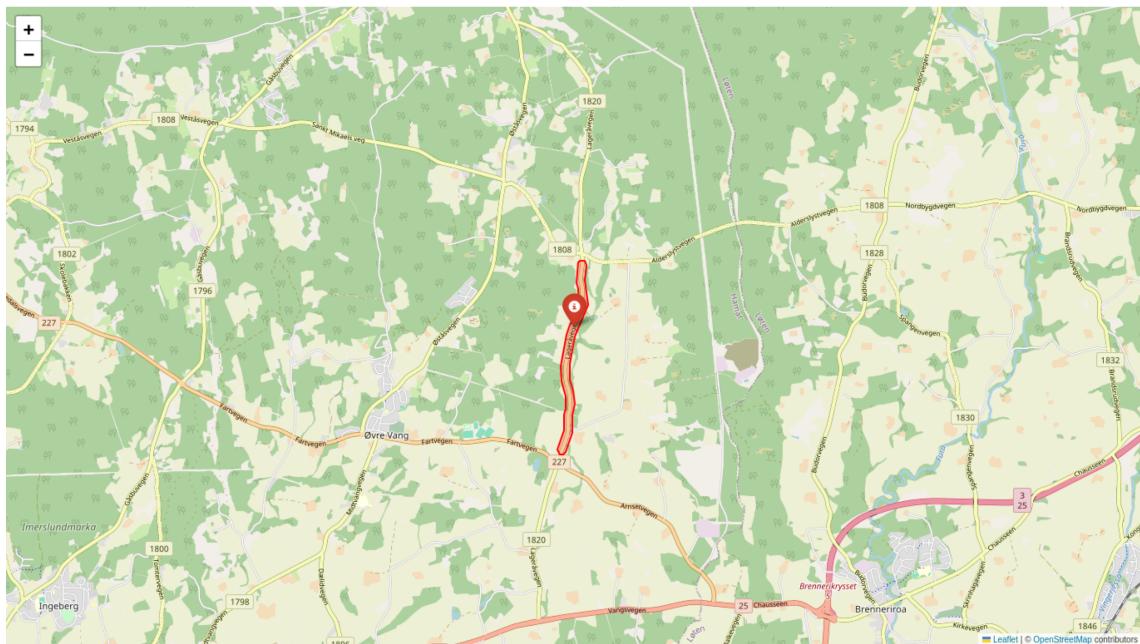
**Figur 3: Histogram og kumulativ fordeling av tidsintervallene mellom etterfølgende logginger**

Med verdier tilhørende det øvre sjiktet av hastighet og tidsintervaller, så burde det resulterende området være romslig nok til å fange opp de aller fleste reelle passeringer. Pluggar man inn verdiene for hastighet og tidsintervall, ender man dermed med en lengde på  $s = (v \times \Delta t)/2 = (90 \text{ km}/\text{t} \times (350 \text{ s}/60^2))/2 = 4.375 \text{ km}$  i hver retning fra punktet. Imidlertid er det flere av punktene som ligger nærmere enn dette til vegkryss, hvor det følgelig ikke lenger er mulig å vite hvor kjøretøyet har kjørt. Dermed stopper området opp ved det første inntreffende av veikryss og 4.375km i hver retning av punktet.

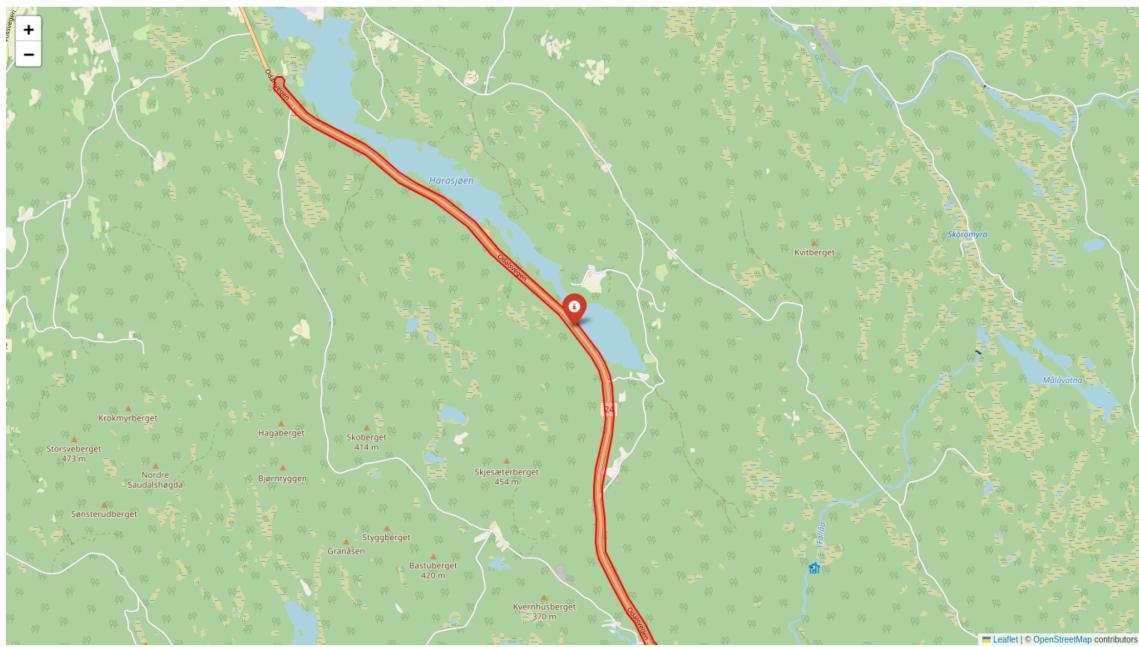
Fv227 S2 (60.84417394856876, 11.189843853440689)



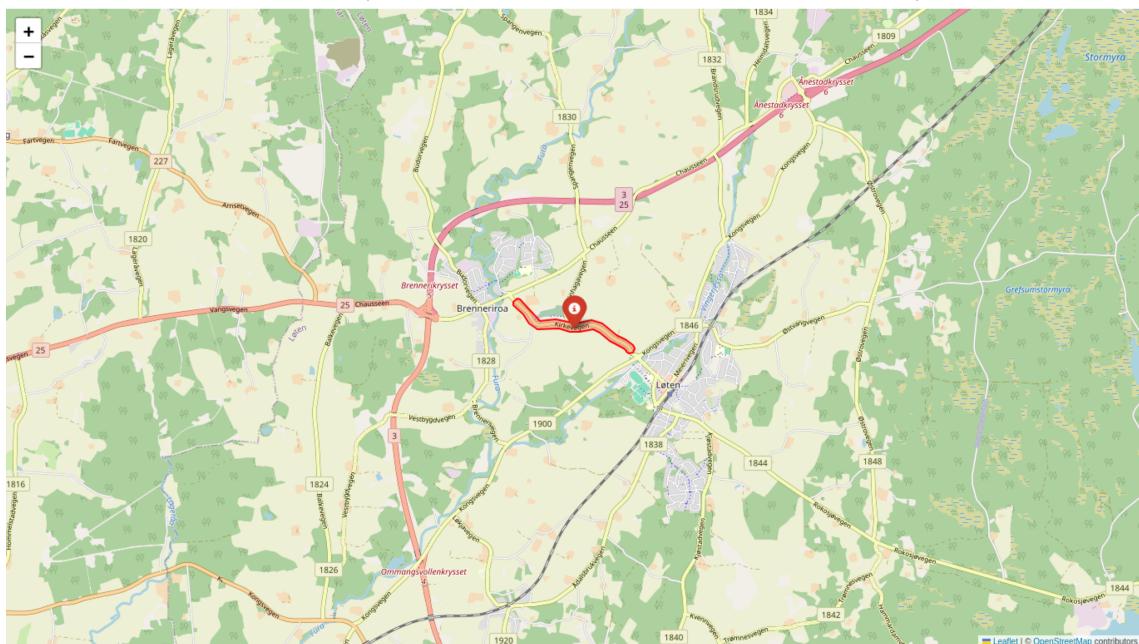
Fv1820 S1 (60.85597322615382, 11.240456370570067)



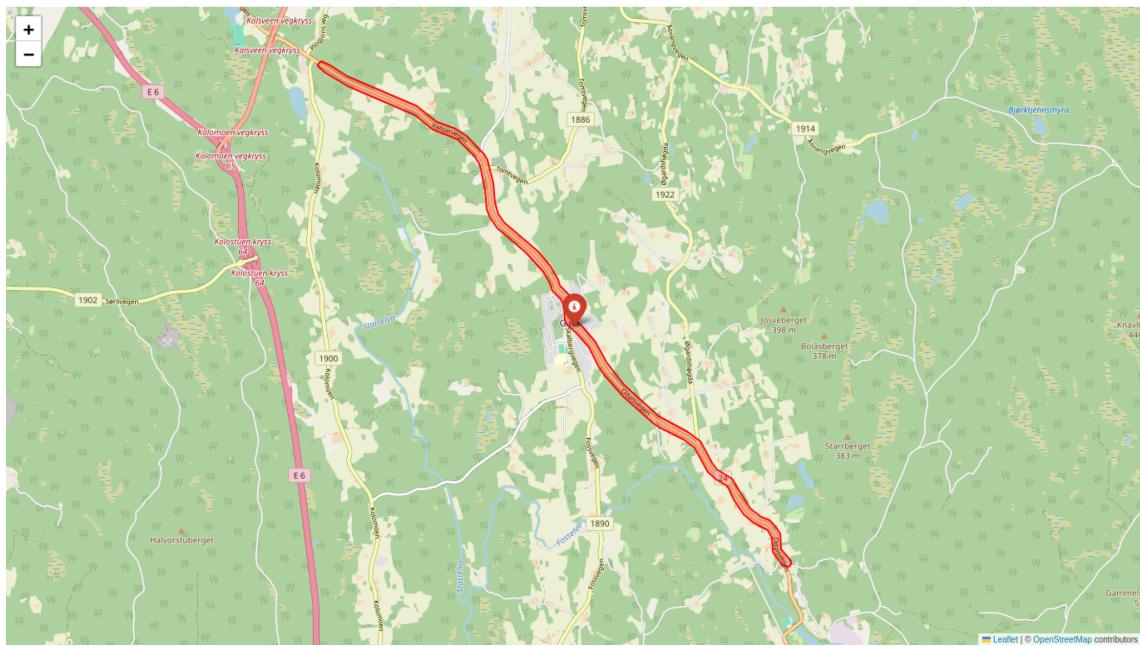
Fv24 S7 (60.620318692686524, 11.45732868472923)



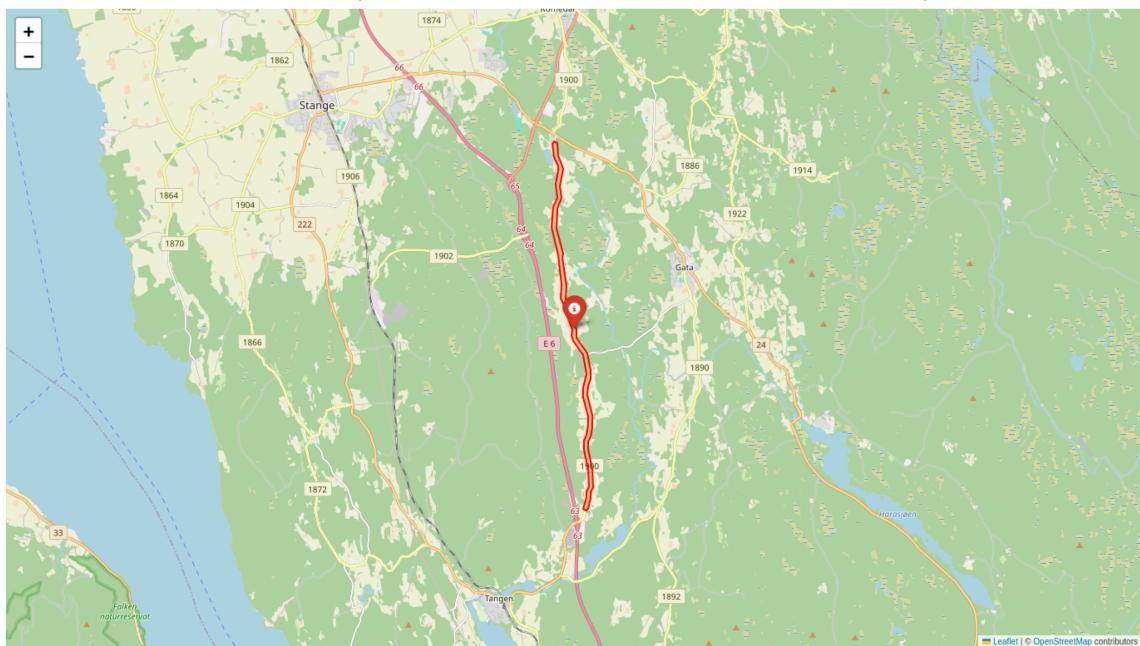
Fv1844 S1 (60.82551243698731, 11.322549441663224)



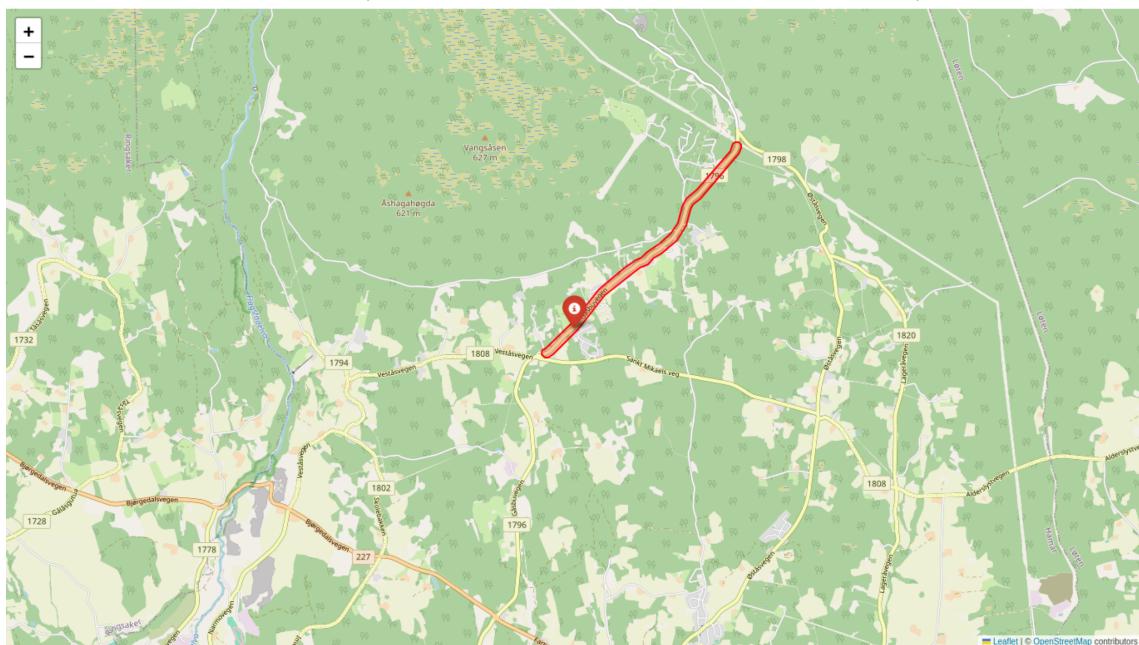
Fv24 S8 (60.68464568378821, 11.34488939729827)



Fv1900 S1 (60.672633328131525, 11.298307776875701)



Fv1796 S2 (60.87926155709279, 11.175477138285103)



## 2.2 Unngå duplike registreringer

Ettersom logger innenfor området 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 passe gjennom hele området. Tabell 1 viser deskriptiv statistikk over de loggførte hastighetene, hvor det kommer frem at den nedre kvartilen har en hastighet på  $26 \text{ km}/\text{t}$ . 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  $t = (2 \times s)/v = (2 \times 4.375 \text{ km})/26 \text{ km}/\text{t} \approx 20 \text{ min}$

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 datapunktet befinner seg innenfor et område som representerer et av veggpunktene av interesse 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 nevne forutsetningene, og som følgelig produserer resultatene:

```

1  def table(
2      df: pd.DataFrame,
3      road_coordinates: dict[str, tuple[float, float]],
4      threshold_length_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_length_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 = ["Vei", *[f"{year} {tonnage}t" for year in years for tonnage in
29                 tonnages]]
30
31     result = {col: [] for col in columns}
32
33     VINs = df["VIN"].unique().tolist()
34
35     for road, (road_lon, road_lat) in road_coordinates.items():
36         most_recent_entry = {VIN: None for VIN in VINs}
37         road_counts = {f"{year} {tonnage}t": 0 for year in years for tonnage in
38                         tonnages}
39         polygon_boundary = load_polygon_boundary_from_file(road)
40
41         for _, entry in df.iterrows():
42             cur_point = Point(entry["Longitude"], entry["Latitude"])
43             if polygon_boundary.contains(cur_point):
44                 VIN = entry["VIN"]
45                 entry_time = entry["Dato"]
46                 if (
47                     most_recent_entry[VIN] is None
48                     or (entry_time - most_recent_entry[VIN]).total_seconds() / 3600
49                     > threshold_time_hours
50                 ):
51                     most_recent_entry[VIN] = entry_time
52
53                     year = entry["Dato"].year
54                     tonnage = entry["Tonnage"]
55                     key = f"{year} {tonnage}t"
56                     if key in road_counts:
57                         road_counts[key] += 1
58
59             result["Vei"].append(road)
60
61             for key, value in road_counts.items():
62                 result[key].append(value)
63
64     return pd.DataFrame(result)

```

Figur 4: 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	4	10	0	154	4	22	20	316	0	67	109	325	5	80	58	98	1272	
1 Fv24 S8	16	10	0	203	15	25	49	724	4	68	499	964	7	74	255	471	3384	
2 Fv227 S2	13	0	0	11	44	0	219	72	39	2	49	93	0	0	13	13	568	
3 Fv1796 S2	29	0	0	4	15	0	92	14	13	0	36	110	2	0	12	5	332	
4 Fv1798 S2	15	0	0	18	36	0	87	133	22	0	115	341	2	0	32	70	871	
5 Fv1820 S1	17	0	0	21	35	0	162	128	36	0	118	296	0	0	19	59	891	
6 Fv1844 S1	14	17	0	19	28	3	3	27	31	12	11	107	5	3	8	10	298	
7 Fv1900 S1	1	0	0	1	3	0	7	88	23	0	77	146	0	0	6	54	406	
Total	NaN	109	37	0	431	180	50	639	1502	168	149	1014	2382	21	157	403	780	8022

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	71	0	0	0	198	0	35	11	198	9	18	27	71	638
1 Fv24 S8	0	0	0	147	0	0	0	553	0	35	41	631	18	18	210	179	1832
2 Fv227 S2	0	0	0	4	0	0	0	40	0	1	0	27	8	0	11	9	100
3 Fv1796 S2	0	0	0	3	0	0	0	12	0	0	0	26	12	0	10	1	64
4 Fv1798 S2	0	0	0	12	0	0	0	51	0	0	0	93	17	0	65	23	261
5 Fv1820 S1	0	0	0	8	0	0	0	60	0	0	0	71	13	0	49	15	216
6 Fv1844 S1	0	0	0	9	0	0	0	7	0	2	0	20	34	0	20	4	96
7 Fv1900 S1	0	0	0	1	0	0	0	67	0	0	13	80	2	0	4	10	177
Total	NaN	0	0	255	0	0	0	988	0	73	65	1146	113	36	396	312	3384

Tabell 3: Resultater ved å benytte loggingene fra tilhengerne