



# HJEMMEEKSAMEN MET4

Vår, 2020

**Start:** 27. april 2020, kl 09:00

**Slutt:** 29. april 2020, kl 14:00

BESVARELSEN SKAL LEVERES I WISEFLOW

På våre nettsider finner du informasjon om hvordan du leverer din besvarelse:  
<https://www.nhh.no/for-studenter/eksamen/innlevering-individuelt-og-i-gruppe/>

Kandidatnummer blir oppgitt på StudentWeb i god tid før innlevering.  
Kandidatnummer skal være påført på alle sider øverst i høyre hjørne (ikke navn eller studentnummer). Ved gruppeinnlevering skal alle gruppemedlemmers kandidatnummer påføres.

UTFYLLENDE BESTEMMELSER FOR HJEMMEEKSAMEN

<https://www.nhh.no/for-studenter/forskrifter/forskrift-om-eksamen-ved-nhh/utfyllende-bestemmelser-til-eksamensforskriften/>

**Antall sider, inkludert forside:** 6

**Antall vedlegg:** 2. avinor.Rdata og avinor\_cleaned.Rdata

## KORONAVIRUS OG FLYAVGANGER I NORGE

### INSTITUTT FOR FORETAKSØKONOMI

NHH

FRA 27. APRIL 2020 KL. 9.00 TIL 29. APRIL 2020 KL. 14.00

#### INNLEDNING

Den pågående koronapandemien har vidtrekkende konsekvenser også på andre områder enn helse. Særlig har luftfartsnæringen blitt hardt rammet. I skrivende stund har prisen på aksjer i SAS og Norwegian falt med henholdsvis 40 og 80 prosent siden slutten av februar. Det hagler med artikler om kansellerte flyavganger og permitteringer i luftfartsnæringen og om mulige tiltak regjeringen kan sette inn for å hjelpe næringen gjennom krisen.

En måte å få en oversikt over koronaepidemiens virkning på luftfartsnæringen, er å bruke avgangsdata. Flyavganger lar oss måle servicenivået i den norske luftfartsnæringen – både samlet og for hvert flyselskap – og kan således være en enklere måte å studere virkningen av krisen på enn å prøve å holde oversikt over alle artiklene i nyhetene. Imidlertid måler avgangsdata, i motsetning til f.eks. aksjepriser, bare aktiviteten i nåtid og fortid, og er ikke i seg selv framtidsorienterte. En annen begrensning ligger i at dataene ikke måler inntekter eller passasjerantall, noe som innebærer at de bare fanger opp en del av luftfartsnæringen.

Dataene er innsamlet fra Avinor gjennom et åpent programmeringsgrensesnitt (API).<sup>1</sup> Se fotnoten for forklaringer på de ulike variablene i datasettet. Data har blitt innsamlet gjennom dette programmeringsgrensesnittet fra 10. januar 2020 fram denne hjemmeeksamenen ble ferdigstilt, og dekker alle kommersielle flyavganger fra flyplasser drevet av Avinor.

Det store spørsmålet i denne hjemmeeksamenen er å kvantifisere koronapandemiens påvirkning på flyavganger. Den generelle ideen er å bruke data fra tiden før krisen til å estimere "typiske" nivåer for flytrafikken og så sammenligne tiden siden krisen satte inn med hva vi ville forvente dersom flytrafikken hadde fortsatt etter samme mønster som før krisen.

Oppgave 1–3 gjelder alle datarammen (eng: *data frame*) ved navn `df_total` – dvs. avganger totalt per dag. Denne datarammen ble laget ved å bearbeide rådata fra Avinor. Se vedlegget for forklaring av variablene i datarammen og et skript som viser hvordan datasettet ble til. I oppgave 4 kan du få behov for et annet datasett til analysen din. Datarammen `df_airline` er ett eksempel på et datasett som kan brukes i oppgave 4.

Eksamensoppgaven inneholder kanskje ikke alle opplysningene du trenger for å svare på alle spørsmålene. Dersom det er tilfelle, må du selv vurdere hvordan du best kan gjennomføre oppgaven. I så fall må det gå tydelig fram i eksamensbesvarelsen hvilke antagelser du har arbeidet ut fra, og bakgrunnen for valgene dine. Skulle noen av oppgavene eller deler av oppgavene være for vanskelige, gjør du ditt beste med de oppgavene du kan svare på.

#### EKSAMENSOPPGAVER

1. Presenter deskriptiv statistikk av dataene. Visualiser både det totale antallet flygninger per dag og kumulative daglige flygninger. Finn ut fra hvilken dato krisen hadde en synlig virkning på avgangene, og forklar hvordan du kom fram til denne datoen.
2. Estimer seks lineære regresjonsmodeller på data fra før krisen. Bruk følgende avhengige variabler:

---

<sup>1</sup> <https://avinor.no/konsern/tjenester/flydata/flydata-i-xml-format>

I: Flygninger totalt per dag

II: Kumulative flygninger totalt

III: Innenlandsflygninger per dag

IV: Kumulative innenlandsflygninger per dag

V: Utenlandsflygninger per dag

IV: Kumulative utenlandsflygninger per dag

I alle regresjoner skal du bruke indikatorvariabler for ukedag og inkludere en lineær trend om nødvendig. Svar på følgende spørsmål:

- a. Tolk regresjonsresultatene. Svaret skal inneholde en drøfting av validiteten til modellene du har estimert.
  - b. Vurder i hvilken grad modellene kan brukes til å predikere kommersiell flyaktivitet i Norge før krisen.
3. Bruk regresjonsmodellene du har estimert, og sammenlign deres prediksjoner med flydata for perioden etter at krisen satte inn. Kvantifiser koronakrisens virkning på den kommersielle flytrafikken i Norge. Diskuter troverdigheten til estimatet ditt og hvordan den kan bedres.
  4. Avgangsdataene inneholder mye informasjon utover det som har blitt drøftet i oppgave 1–3, og denne informasjonen kan brukes til å analysere krisens virkninger på f.eks. regioner og de enkelte flyselskap. Bruk dataene sammen med en egnet statistisk metode etter eget valg og presenter virkninger av koronakrisen som ikke allerede er dekket i oppgave 1–3.

#### ADMINISTRATIVE BESTEMMELSER

- Hjemmeeksamen i Met4 skal leveres i grupper på 2, 3 eller 4 studenter.
- Studenter får ikke diskutere eksamen med studenter utenfor sin egen gruppe etter at eksamensoppgaven har blitt lagt ut.
- Besvarelsene vil bli rettet iht rubrikk postet på Canvas.
- Du kan besvare eksamen på norsk eller engelsk.
- Send en e-post til både Ole-Petter Moe Hansen (s9705@nhh.no) og Håkon Otneim (hakon.otneim@nhh.no) dersom du har spørsmål om eksamenen. Eventuell viktig tilleggsinformasjon blir lagt ut på kursets hjemmeside på Canvas. NB! Det vil ikke bli gitt tilleggsinformasjon som supplerer oppgaveteksten. Kun forespørsler som gjelder eventuelle feil i oppgaveteksten besvares.
- Rapporten kan maksimalt være 10 sider lang, inkludert tabeller, figurer og kildeliste. Dersom rapporten har en forside som ikke inneholder noen form for besvarelse på eksamensoppgavene, kan forsiden komme i tillegg til de 10 sidene. En eventuell innholdsliste vil telle med i sidetallet, men er ikke obligatorisk. Prioriter hva som inkluderes i rapporten!
- Rapporten skal være i Times New Roman med fontstørrelse 12 og linjeavstand 1,5. Teksten i figurer og tabeller kan være så liten som fontstørrelse 9.
- Eksamenen blir gjennomført via Wiseflow. Eksamensbesvarelsen skal leveres som én enkelt PDF-fil. Andre filformater (f.eks. .doc, .docx og .R) blir ikke godtatt.

#### DATASETT OG RÅD OM R

Rådataene er lagret i en fil som heter "avinor.Rdata". Filen inneholder datarammen "df", som inneholder rådataene fra Avinor. Filen "avinor\_cleaned.Rdata" inneholder to bearbejdede datasett:

- "df\_total", som er en dataramme med antall flygninger per dag, både sammenlagt og fordelt på innenlands- og utenlandsflygninger. Se koden på neste side som viser hvordan "df\_total" ble til på grunnlag av rådataene. I tillegg til data om flygninger inneholder "df\_total" variabler for dato, ukedag og en trend.
- "df\_airline". Dette er et *eksempel* på en dataramme som kan brukes til å besvare oppgave 4. "df\_airline" er svært lik "df\_total", men datasettet inneholder en rad som gir totalt antall flygninger

*for ett bestemt flyselskap.* Variablene i “df\_airline” er de samme som i “df\_total”, bortsett fra at “df\_total” i tillegg inneholder en variabel som viser hvilket flyselskap observasjonen gjelder.

Skriptet på de neste sidene viser hvordan “df\_airline” og “df\_total” ble laget på grunnlag av rådataene. Det er ikke påkrevet å forstå dette skriptet for å kunne besvare eksamenen, men det kan tjene som utgangspunkt om du skulle ønske å lage ditt eget datasett i oppgave 4.

```

# Load necessary libraries. If you do not have these installed,
# you may need to run "install.library("libraryname")" where you
# replace libraryname with the library you want to install:
library(tidyverse)
library(lubridate)

# Load the avinor-data.
load("avinor.Rdata")

df_total <-
df %>% # The raw data frame from Avinor
  filter(
    # Drop all flights that have not yet departed
    code == "D") %>%
  mutate(
    # Create an indicator variable, equal to 1 if the flight
    # is international
    international = case_when(dom_int == "D" ~ 0, TRUE ~ 1),
    # Recode the departure time into a date
    dep_date = as.Date(schedule_time)
  ) %>%
  filter(
    # Drop all obs. before 19th of Jan, due to a data hiccup
    # at this date.
    dep_date > as.Date('2020-01-19')) %>%
  group_by(
    # Group the data frame by each date
    dep_date) %>%
  summarise(
    # Count the total number of flights each day
    flights = n(),
    # Count the number of international flights each day
    flights_int = n() * sum(international) / n(),
    # Count the number of domestic flights each day
    flights_dom = n() * sum(1-international) / n()) %>%
  arrange(
    # Ensure the data frame is sorted by date
    dep_date) %>%
  mutate(
    # Cumulative number of daily flights
    cumflights = cumsum(flights),
    # Cumulative number of daily international flights
    cumflights_int = cumsum(flights_int),
    # Cumulative number of daily domestic flights
    cumflights_dom = cumsum(flights_dom),
    # Create a factor variable, indicating the day of week
    day_of_week = factor(lubridate::wday(dep_date)),
    # Calculate a trend variable, increasing by 1 each day.
    trend = 1,
    trend = cumsum(trend))

df_airline <-
df %>% # The original data set
  filter(
    # Drop all flights that have not yet departed
    code == "D") %>%
  mutate(
    # Create an indicator variable, equal to 1 if the flight
    # is international
    international = case_when(dom_int == "D" ~ 0, TRUE ~ 1),
    # Recode the departure time into a date
    dep_date = as.Date(schedule_time)
  ) %>%
  filter(
    # Drop all obs. before 19th of Jan, due to a data hiccup on the
    # 19th.
    dep_date > as.Date('2020-01-19')) %>%

```

```

group_by(
  # Group the data frame by each date and airline
  dep_date, airline) %>%
summarise(
  # Count the total number of flights each day
  flights = n(),
  # Count the number of international flights each day
  flights_int = n() * sum(international) / n(),
  # Count the number of domestic flights each day
  flights_dom = n() * sum(1-international) / n()) %>%
ungroup(
  # Remove the grouping from the data set
) %>%
full_join(
  # Some airlines might not have any flights every day. If so, they
  # will be missing entirely from the data frame. I would prefer to have
  # such days as a row in the data frame, with the number of flights
  # equal to zero. So, expand the data set with all the missing
  # date/airline combinations
  expand.grid(
    dep_date = unique(.$dep_date),
    airline = unique(.$airline))) %>%
mutate(
  # Set the flights data to zero for all those we have
  # just filled in
  flights = replace_na(flights, 0),
  flights_int = replace_na(flights_int, 0),
  flights_dom = replace_na(flights_dom, 0)) %>%
arrange(
  # Sort the data frame by airline and dep. date
  airline, dep_date) %>%
group_by(
  # Group by airline
  airline) %>%
mutate(
  # Create cumulative variables, day of week and a trend.
  cumflights = cumsum(flights),
  cumflights_int = cumsum(flights_int),
  cumflights_dom = cumsum(flights_dom),
  day_of_week = factor(lubridate::wday(dep_date)),
  trend = 1,
  trend = cumsum(trend)) %>%
mutate(
  # There are many airlines with very few flights.
  # Find the total number of flights for each airline.
  max_cum_flights = max(cumflights)) %>%
filter(
  # Remove all airlines with less than 100 flights.
  max_cum_flights > 100) %>%
ungroup()

# Save the cleaned data frames in a new file
save(
  df_total,
  df_airline,
  file = "avinor_cleaned.Rdata")

```