STK-IN1050 Statistikk for informatikere, V26

Obligatorisk oppgave 4 av 4

Oppgave 1: Døgntemperatur på Blindern i julimåned

Meteorologisk institutt har sitt hovedkontor på Blindern i Oslo. I hagen til hovedbygget står en værstasjon hvor det blant annet måles temperatur hver time. Vi skal her bruke døgntemperatur, definert som gjennomsnittet av de 24 timesverdiene for et døgn. Dataene er de samme som i oblig 2. De er blitt lastet ned fra seklima.met.no og kan leses inn direkte fra GitHub:

https://github.uio.no/milenpa/IN-STK1050/tree/main/oblig/oblig4

Vi ser på data for juli i to tiårsperioder:

- 1986–1995 (fil:`blindern_daily1986-1995.txt`)
- 2016–2025 (fil:`blindern_daily2016-2025.txt`)

Hver fil inneholder én linje per dag i juli (31 dager per år, totalt 310 dager per periode) med kolonnene `day`, `month`, `year` og `temp`, der `temp` er døgntemperaturen i grader Celsius.

Vi ønsker å undersøke om gjennomsnittlig døgntemperatur i juli på Blindern har endret seg fra perioden 1986–1995 til perioden 2016–2025.

Velg to ulike metoder for hypotesetesting (for eksempel en to-utvalgs t-test og en permutasjonstest) for å teste om gjennomsnittlig døgntemperatur i juli er forskjellig mellom de to periodene. For hver metode skal du gjennomføre følgende:

1. **Beskrivelse av metode**

   - Gi en kort beskrivelse av metoden.
   - Forklar kort hvorfor den er egnet i denne situasjonen.
   - Formuler nullhypotesen H_0, den alternative hypotesen H_a og velg et signifikansnivå α.

Vi skal bruke en to-utvalgs t-test og en permutasjonstest.

To-utvalgs t-test: Vi har $X_1, \dots, X_{310}$, døgntemperatur i juli på Blindern i perioden 1986-1995 og $Y_1, \dots, Y_{310}$, døgntemperatur i juli på Blindern i perioden 2016-2025. Vi er interessert i å undersøke om den gjennomsnittlige døgntemperaturen har endret seg fra den ene perioden til den andre. Dette er presis den type situasjon to-utvalgs t-test er laget for: å sammenligne forventninger av to utvalg. Betegn med μ_X forventningen av X og med μ_Y forventningen av Y. Vi velger signifikansnivå 0.05 og setter opp hypoteser:

$$H_0 : \mu_X - \mu_Y = 0$$
$$H_a : \mu_X - \mu_Y \neq 0.$$

Da er testobservatoren gitt ved

$$T = \frac{\bar{X} - \bar{Y}}{\mathrm{SE}(\bar{X} - \bar{Y})} \sim t(\mathrm{df}),$$

hvor standardfeilen $\text{SE}(\bar{X} - \bar{Y})$ og antall frihetsgrader, df, avhenger av om vi antar lik eller forskjellig varians i de to gruppene. Vi vurderer begge situasjoner nedenfor.

Vi har her satt opp en to-sidig test, og p-verdien er derfor gitt ved $2P(T > |t|)$, hvor t er den observerte testobservatoren.

Permutasjonstest: Alternativt vurderer vi også en permutasjonstest med samme signifikansnivå og hypoteser

$$H_0 : \text{De to gruppene kommer fra samme fordeling,}$$
$$H_a : \text{De to gruppene kommer fra forskjellige fordelinger.}$$

Vi er her hovedsaklig interessert i gjennomsnittlig døgntemperatur per periode og vi velger derfor en testobservator gitt ved forskjell i gjennomsnitt, $\bar{X} - \bar{Y}$. Under nullhypotesen er etiketten tildelt hver observasjon vilkårlig. Vi gjennomfører testen ved å gjentatt tildele observasjonene en tilfeldig gruppeetikett og regne ut testobservatoren $\bar{x} - \bar{y}$ for hvert tilfelle. Deretter regner vi ut p-verdien gitt ved sannsynligheten for å observere noe like eller mer ekstremt enn den observerte testobservatoren (med opprinnelige etiketter) ved tilfeldig utdeling av gruppeetiketter.

2. **Gjennomføring av testen**

   - Beregn testobservator.
   - Beregn eller estimer p-verdien.
   - Gi en kort og tydelig konklusjon formulert i ord (bruk både testresultat og faglig kontekst).

Koden nedenfor bruker tilgjengelige funksjoner i Python til å gjennomføre testene. Vi gjennomfører tre tester, to-utvalgs t-test med og uten antakelse om lik varians, samt en permutasjonstest basert på forskjell i gjennomsnitt.

```
import numpy as np
import pandas as pd
from scipy import stats
from scipy.stats import permutation_test

file_old = "blindern_daily1986-1995.txt"
file_new = "blindern_daily2016-2025.txt"

def read_temp_file(path):
    df = pd.read_csv(path, sep=';', decimal='.', header=0)
    df.columns = [c.strip('"') for c in df.columns]
    df['month'] = pd.to_numeric(df['month'], errors='coerce')
    df['temp'] = pd.to_numeric(df['temp'], errors='coerce')
    return df

df_old = read_temp_file(file_old)
df_new = read_temp_file(file_new)

july_old = df_old['temp'].dropna().to_numpy()
july_new = df_new['temp'].dropna().to_numpy()

t_stat, p_value_t = stats.ttest_ind(july_new, july_old, equal_var=True)
```

```
print("To-utvalgs-test (lik varians)")
print("t observator:", t_stat.round(3))
print("p-verdi:", p_value_t.round(7))

t_stat, p_value_t = stats.ttest_ind(july_new, july_old, equal_var=False)

print("To-utvalgs t-test (ulik varians)")
print("t observator:", t_stat.round(3))
print("p-verdi:", p_value_t.round(7))

def mean_diff(x, y, axis):
    return np.mean(x, axis=axis) - np.mean(y, axis=axis)

res = permutation_test(
    (july_new, july_old),
    mean_diff,
    n_resamples=1000,       # number of permutations
    alternative='two-sided',
    vectorized=True,
    random_state=42
)

print("Test observator:", res.statistic.round(3))
print("Permutasjon p-verdi:", res.pvalue.round(7))
```

```
To-utvalgs-test (lik varians)
t observator: 5.085
p-verdi: 5e-07
To-utvalgs t-test (ulik varians)
t observator: 5.085
p-verdi: 5e-07
Test observator: 1.18
Permutasjon p-verdi: 0.001998
```

Vi observerer at det er ingen forskjell om man antar lik/ulik varians i de to gruppene. Konklusjonen for t-test er at vi forkaster null hypotesen om lik forventning, dvs. vi konkluderer at det er forskjellig forventet døgntemperatur i de to periodene. For permutasjonstest forkaster vi null hypotesen og konkluderer at fordelingen av døgntemperatur er forskjellig for de to periodene.

3. **Antakelser**

    - Beskriv hvilke antakelser metoden bygger på.
    - Diskuter om det virker rimelig at disse antakelsene er oppfylt i dette tilfellet, basert på dataene (for eksempel ved hjelp av grafer eller oppsummerende statistikk).

t-testen antar uavhengige observasjoner innenfor og mellom utvalgene, at begge populasjoner er tilnærmet normalfordelte (hvor noen avvik er greit pga antall datapunkter), og enten lik eller ulik varians.

Permutasjonstest har minimale antakelser om form på fordelingen til testobservator, men den kan gi feil resultater hvis dataene er sterkt korrelerte.

For å vurdere om antakelse er oppfylt vurderer vi først om det er korrelasjon i dataene. Spesifikt, så vurderer vi om døgntemperatur på en dag er korrelert med temperaturen dagen før, noe som også kalles lag-1 korrelasjon:

```
def lag1_autocorr(x):
    x = np.asarray(x, dtype=float)
    x = x[~np.isnan(x)]
    return np.corrcoef(x[:-1], x[1:])[0, 1]

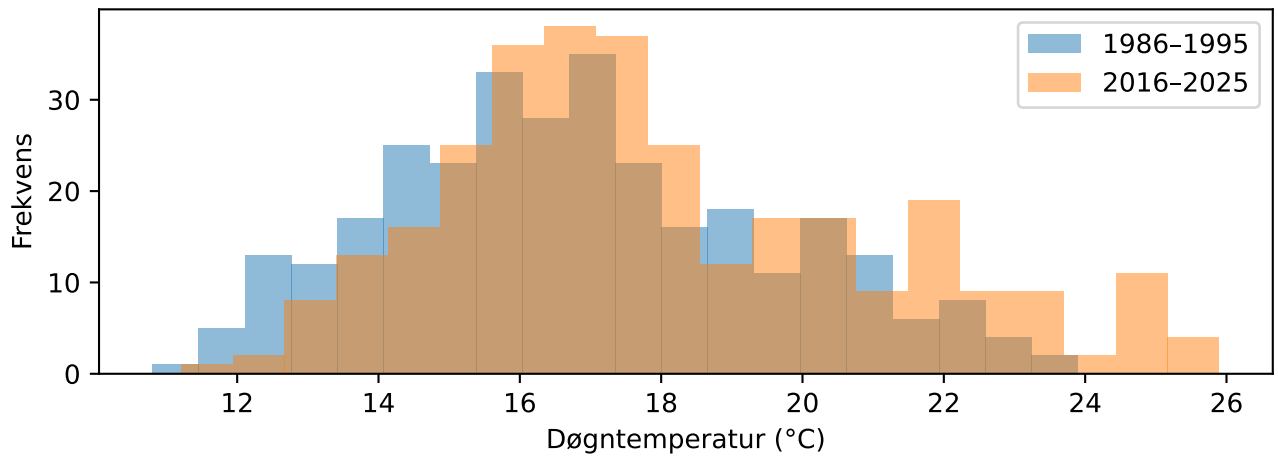
rho1 = lag1_autocorr(july_old)
rho2 = lag1_autocorr(july_new)
print("Lag-1 correlation 1986-1995:", round(rho1,3))
print("Lag-1 correlation 2016-2025:", round(rho2,3))
```

```
Lag-1 correlation 1986-1995: 0.741
Lag-1 correlation 2016-2025: 0.817
```

Vi ser at det er sterk korrelasjon mellom temperaturen på to påfølgende dager, noe som indikerer at antakelsene bak testene ikke er oppfylt.

I tillegg vurderer vi formen på fordelingene, for å se om antakelse i t-testen om normalfordeling er oppfylt:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 2.5))
plt.hist(july_old, bins=20, alpha=0.5, label="1986-1995")
plt.hist(july_new, bins=20, alpha=0.5, label="2016-2025")
plt.legend()
plt.xlabel("Døgntemperatur (°C)")
plt.ylabel("Frekvens")
plt.show()
```



Histogrammene er litt skjeve, men det er mange observasjoner og ikke noen åpenbare outliers.

4. **Sammenligning av metodene**

    - Sammenlign kort de to metodene du har brukt (fordeler/ulemper, robusthet, tolkning).

Det observeres ikke stor forskjell mellom de to metodene, men permutasjonstest er oft litt mer robust mot outliers og har færre antakelser. Mens t-test kan virke bedre på små datasett hvor antakelser om normalfordeling er godt oppfylt. De tolkes her på en ligende måte, dersom vi har brukt forskjell i gjennomsnitt som testobservator for permutasjonstesten. Permutasjonstesten kan også brukes med andre testobservatorer, noe som gjør den mer fleksibel.

Oppgave 2: Månedsgjennomsnitt for juli

Vi bruker de samme dataene som i oppgave 1, men nå konstruerer vi månedstemperatur for juli for hvert år.

For hvert år i perioden 1986–1995 og 2016–2025 skal du beregne månedstemperaturen i juli som gjennomsnittet av alle døgntemperaturene i juli det aktuelle året. Dermed får du to datasett:

- Ett datasett med 10 observasjoner (1986–1995), én per år.
- Ett datasett med 10 observasjoner (2016–2025), én per år.

```
mean_old_year = df_old.groupby("year")["temp"].mean()
mean_new_year = df_new.groupby("year")["temp"].mean()
```

Bruk de samme to metodene som i oppgave 1 til å undersøke om gjennomsnittlig månedstemperatur i juli har endret seg fra perioden 1986–1995 til perioden 2016–2025.

For hver metode skal du:

1. Formulere H_0, H_a og signifikansnivå α.

Her bruker vi helt tilsvarende hypoteser og signifikansnivå som før, bare på månedsnivå i stedet for døgnnivå.

2. Beregne testobservator og p-verdi og gi en kort konklusjon.

Vi kjører de samme funksjonene som før på de nye dataene:

```
t_stat, p_value_t = stats.ttest_ind(mean_new_year, mean_old_year, equal_var=True)

print("To-utvalgs-test (lik varians)")
print("t observator:", t_stat.round(3))
print("p-verdi:", p_value_t.round(7))

t_stat, p_value_t = stats.ttest_ind(mean_new_year, mean_old_year, equal_var=False)

print("To-utvalgs t-test (ulik varians)")
print("t observator:", t_stat.round(3))
print("p-verdi:", p_value_t.round(7))

def mean_diff(x, y, axis):
```

```
    return np.mean(x, axis=axis) - np.mean(y, axis=axis)

res = permutation_test(
    (mean_new_year, mean_old_year),
    mean_diff,
    n_resamples=1000,       # number of permutations
    alternative='two-sided',
    vectorized=True,
    random_state=42
)

print("Test observator:", res.statistic.round(3))
print("Permutasjon p-verdi:", res.pvalue.round(7))
```

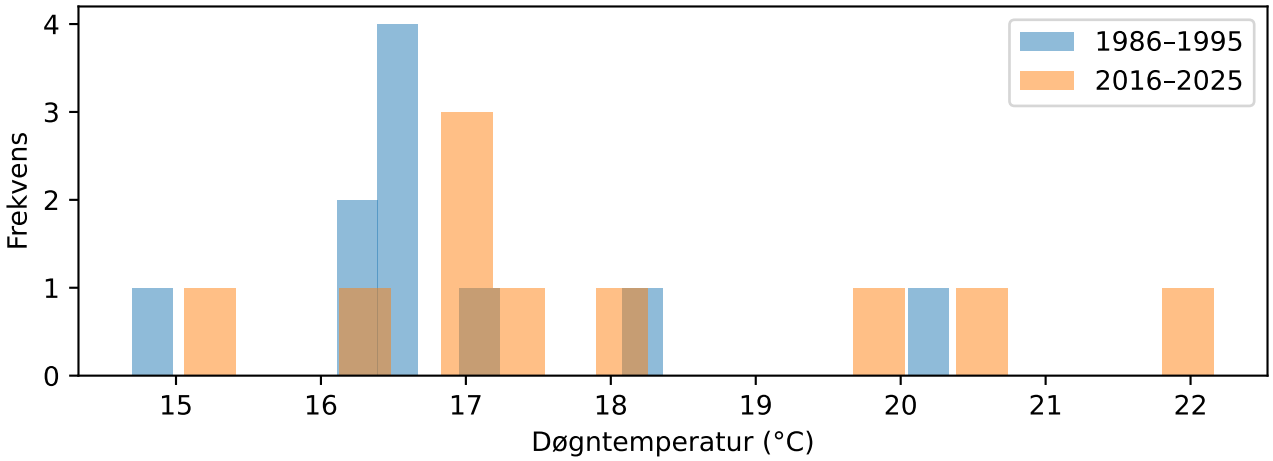
```
To-utvalgs-test (lik varians)
t observator: 1.411
p-verdi: 0.175297
To-utvalgs t-test (ulik varians)
t observator: 1.411
p-verdi: 0.1775661
Test observator: 1.18
Permutasjon p-verdi: 0.1678322
```

Her kan vi ikke forkaste nullhypotesene om lik forventning, eller lik fordeling. Vi kunne eventuelt bruke flere versjoner med permutasjonstesten, med andre testobservatorer. Men, dersom vi er hovedsaklige interessert i forskjell i gjennomsnittlig temperatur, så stopper vi her.

3. Diskutere antakelsene for metoden, og om de virker rimelig oppfylt for disse dataene.

Vi bruker de samme testene som før, så vi har samme antakelser. Men vi har fått nye data. Her har vi bare en observasjon per måned og vi vurderer bare den ene måneden hvert år. Det er derfor rimelig å anta at dataene er uavhengige. Dermed mangler bare å sjekke normalfordeling for t-test:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 2.5))
plt.hist(mean_old_year, bins=20, alpha=0.5, label="1986-1995")
plt.hist(mean_new_year, bins=20, alpha=0.5, label="2016-2025")
plt.legend()
plt.xlabel("Døgntemperatur (°C)")
plt.ylabel("Frekvens")
plt.show()
```



Vurdering av form på fordeling er vanskelig med så få observasjoner, men vi forventer at månedstemperatur er rimelig nærme normalfordeling pga det store antallet gjennomsnitter som tas for å komme fram til månedstemperatur.

Til slutt skal du:

- Sammenligne situasjonen og konklusjonene i oppgave 1 (døgndata) og oppgave 2 (månedsgjennomsnitt).
- Diskutere hvilken analyse du ville lagt mest vekt på for å si noe om endring i sommertemperatur på Blindern, og begrunne valget.

De to analysene gir to helt forskjellige konklusjoner. I den første oppgaven forkaster vi nullhypotese om lik forventning eller lik fordeling på en overbevisende måte med p-verdier nærme null. I oppgave to har vi ikke nok bevis til å forkaste hypotese om lik forventning/fordeling, med p-verdier på rundt 0.17. På grunn av at vi forventer at månedstemperatur er nærme normalfordelt og at vi kan unngå avhengighet ved å gå over til månedsskala, så foretrekkes analysen i oppgave to, selv om det er lite data.