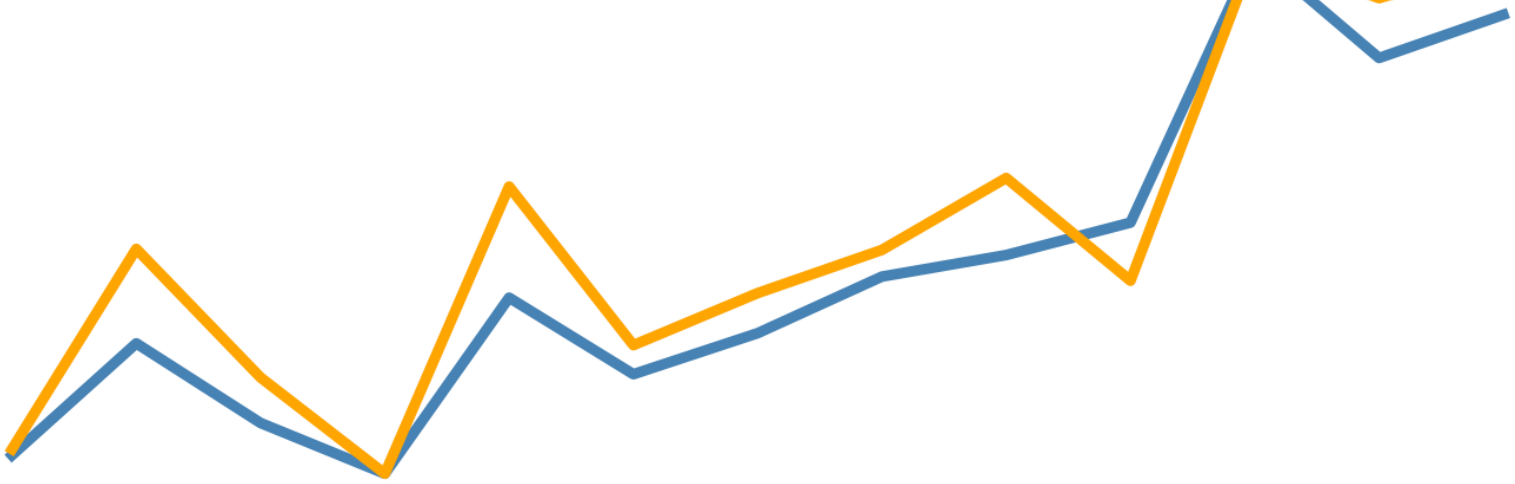


0.1 Hypotes

Vi ska utforska hypotesen att det finns ett signifikant samband mellan 'mpg' och 'weight', där tyngre fordon uppvisar lägreMPG-värden.

Hypotes: - H0: Det finns inget samband mellan 'mpg' och 'weight'. - H1: Det finns ett signifikant negativt samband mellan 'mpg' och 'weight'.

0.2 Signifikansnivå

Vi kommer att använda en signifikansnivå på 0,05 för att avgöra om vi ska förkasta nollhypotesen. Ett p-värde mindre än 0,05 kommeratt leda oss att förkasta nollhypotesen och acceptera att det finns ett signifikant samband mellan 'mpg' och 'weight'.

0.3 Tillvägagångssätt

Vår analys kommer att inkludera:

- Visualisering av sambandet mellan 'mpg' och 'totalvikt' med hjälp av plots.
- Genomförande av en korrelationsanalys för att undersöka sambandet.
- Utförande av linjär regression för att undersöka hur väl 'totalvikt' kan förutsäga 'mpg'.
- Tolka resultaten för att dra slutsatser och förstå effekten av fordonsvikt på bränsleeffektiviteten.

0.4 Dataförberedelse och import av bibliotek

- **Pandas** för behandling av datat.
- **NumPy** för matematiska beräkningar.
- **Matplotlib** eller **Seaborn** för visualisering.
- **SciPy** och/eller **StatsModels** för statistiska beräkningar.

[ ]:
```
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import time as time
import pandas as pd

df = pd.read_csv('auto-mpg.csv')
df_copy = df.copy()
```

[ ]:
```
column_names = df.columns.to_list()
datatypes = df.dtypes.to_list()
print("**Column names:**\n", column_names)

summary_statistics = df.describe().aggregate(["mean", "std", "min", "max",␣
 ↪"count", "median"])
print("\n**Summary statistics:**\n", df.describe().aggregate(["mean", "std",␣
 ↪"min", "max", "count", "median"]).to_markdown())

if df.isnull().sum().sum() > 0:
    print("\n**Missing values:**\n", "No missing values")
else:
    print("\n**Missing values:**\n", df.isnull().sum().sum())
```

Column names:
['mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model_year', 'origin', 'name']

Summary statistics:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
mean	69.3038	54.0195	216.681	138.183	2450.44	61.9532	107.213
std	133.375	139.007	143.576	118.661	1535.55	135.938	120.242
min	7.81598	1.701	68	38.4912	398	2.75769	3.69763
max	398	398	455	392	5140	398	398
count	8	8	8	8	8	8	8
median	23.2573	4.72739	170.963	98.9847	2513.62	15.534	76.005

Missing values:
No missing values

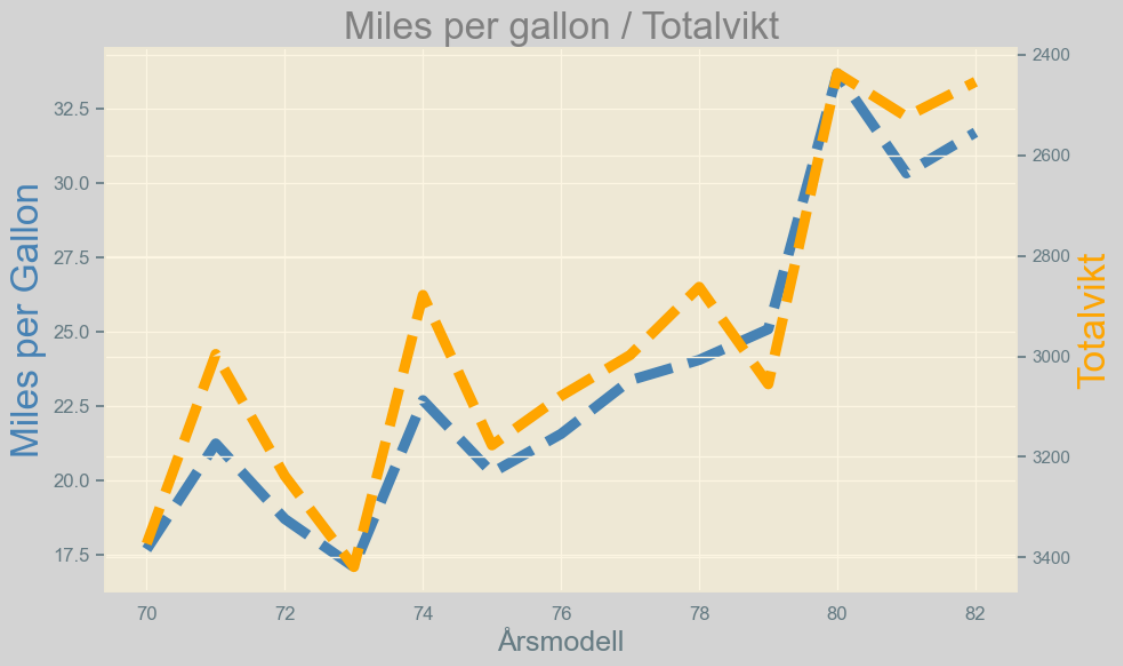
[ ]:
```
# 1. Namn på kolumnen 'model year' ändras till 'model_year'
if 'model year' in df.columns:
    df.rename(columns={'model year': 'model_year'}, inplace=True)
    time.sleep(1)

df['name'] = df['name'].str.replace('chevroelt', 'chevrolet')
df['model_year'] = df['model_year'].astype(int)
```

```
# 3. Datatyp för kolumnen 'mpg' ändras till float
df['mpg'] = df['mpg'].astype(float)
# 4. Kolumnen 'origin' ändras till kategori och kategorierna ändras till 'USA',␣
 ↪'Europe' och 'Asia'
df['origin'] = df['origin'].astype('category')
df['origin'] = df['origin'].cat.rename_categories({1: 'USA', 2: 'Europe', 3:␣
 ↪'Asia'})
#fix usa string issue in origin
df['origin'] = df['origin'].str.replace('1', 'USA')
df['origin'] = df['origin'].str.replace('2', 'Europe')
df['origin'] = df['origin'].str.replace('3', 'Asia')


# 5. Utskrift för att bekräfta de ändrade datatyperna och kategorierna
# Utskrift för att bekräfta de ändrade datatyperna och kategorierna
print("Bekräfta ändrade datatyper:")
print("mpg == float?", df['mpg'].dtype == 'float64')
print("model year == int?", df['model_year'].dtype == 'int32')
print("origin == category?", df['origin'].dtype == 'category')
```

```
Bekräfta ändrade datatyper:
mpg == float? True
model year == int? True
origin == category? False
```

[ ]:
```
from operator import invert
import matplotlib.pyplot as plt

import seaborn as sns
import warnings # tar bort warnings
warnings.filterwarnings('ignore')

# lineplot för att visa hur mpg och weight varierar över tid
fig, ax1 = plt.subplots(figsize=(10, 6))
# Skapa en andra y-axel för 'weight' på samma axel
ax2 = ax1.twinx()


sns.lineplot(x='model_year', y='mpg', data=df, color='steelblue', ax=ax1,␣
 ↪linewidth=6.5, linestyle='--', alpha=1, ci=None)
ax1.set_ylabel('Miles per Gallon', fontsize=24)
ax1.tick_params(labelsize=12)

sns.lineplot(x='model_year', y='weight', data=df, color='orange', ax=ax2,␣
 ↪linewidth=6.5, linestyle='--', alpha=1, ci=None)
intervals = df.groupby('model_year')['weight'].agg(['mean', 'std']).
 ↪reset_index()
```

```
ax2.set_ylabel('Totalvikt', fontsize=24)
ax2.invert_yaxis()  # Invert the y-axis


ax1.set_title('Miles per gallon / Totalvikt', color='grey' , fontsize=24)
ax1.set_xlabel('Årsmodell', fontsize=18)
ax1.yaxis.label.set_color('steelblue')
ax2.yaxis.label.set_color('orange')
ax1.set_facecolor = plt.gcf().set_facecolor('lightgray')



plt.tight_layout()
plt.show()
```



0.5 Visuell analys av data

Vi kan se att bränsle-effektiviteten ökar över tid, samtidigt som totalvikten minskar. Detta kan vara en indikation på att bilar blir mer bränsle-effektiva över tid. procentuellt sett, ser vi att det finns en större spridning av 'mpg'-värden för lättare fordon, medan tyngre fordon har en mer konsekvent 'mpg'-värde. Vi ser samtidigt en tydlig korrelation mellan värdena, vilket vi undersöker nedan. För att ytterligare undersöka korrelationen mellan 'mpg' och 'Totalvikt', kan vi använda en scatter plot för att visualisera sambandet mellan dessa variabler.

Vi kan också beräkna Pearson's och Spearman's korrelationskoefficienter för att kvantifiera styrkan

och riktningen av korrelationen mellan 'mpg' och 'Totalvikt'.

Slutligen kan vi genomföra en linjär regressionsanalys för att undersöka hur väl 'Totalvikt' kan förutsäga 'mpg'. Vi kan tolka de resulterande koefficienterna för att förstå hur 'Totalvikt' påverkar 'mpg' och bedöma modellens prestanda genom att analysera R^2-värdet.

0.6 Korrelationsanalys

För att kvantifiera styrkan och riktningen av sambandet mellan 'mpg' och 'vikt', gör vi en korrelationsanalys. Korrelationskoefficienten hjälper oss att förstå till vilken grad dessa två variabler är relaterade. En negativ korrelationskoefficient skulle indikera att högre vikt är associerad med lägre 'mpg', vilket stöder vår initiala hypotes.

[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

result = stats.pearsonr(df['weight'], df['mpg'])
print("\nPearson's correlation coefficient:", result[0], "\np-value:",␣
 ↪result[1], "\n")
#source: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.
 ↪spearmanr.html

result = stats.spearmanr(df['weight'], df['mpg'])
print("Spearman's correlation coefficient:", result[0], "\np-value:", result[1])
#source: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.
 ↪pearsonr.html
```

```
Pearson's correlation coefficient: -0.831740933244335
p-value: 2.9727995640496354e-103

Spearman's correlation coefficient: -0.8749473981990712
p-value: 8.643768710426594e-127
```

0.7 Tolkning av resultat

Både Pearson's och Spearman's korrelationskoefficienter är negativa och signifikanta ($p < 0.05$), vilket indikerar att det finns en stark negativ korrelation mellan 'mpg' och 'Totalvikt'. Detta stöder vår hypotes om att tyngre fordon har lägre bränsleeffektivitet.

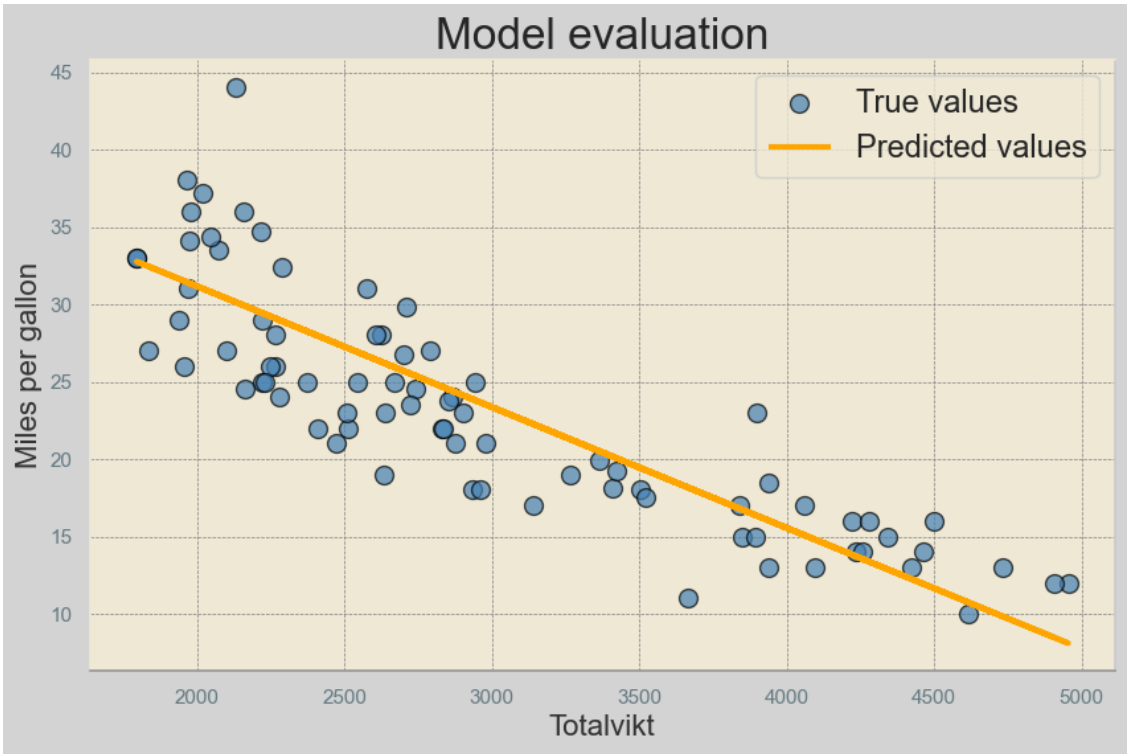
0.8 Linjär regression

För att undersöka hur väl 'Totalvikt' kan förutsäga 'mpg', kommer vi att genomföra en linjär regressionsanalys. Vi kommer att undersöka modellens prestanda och tolka de resulterande koefficienterna för att förstå hur 'Totalvikt' påverkar 'mpg'.

[ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def train_model(X, y):
    """
    80-training, 20-test split.
    Tränar en linear model, och returnerar den tränade modellen.
    """
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

    model = LinearRegression()
    model.fit(X_train, y_train)
    return model

# gör om till numpy array för att kunna använda sklearn
X = df['weight'].values.reshape(-1, 1)
y = df['mpg'].values

# Tränar modellen, 80% / 20% split

model = train_model(X, y)

model_score = model.score(X_test, y_test)
model_prediction = model.predict(X_test)


# Evaluate
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
model_prediction_mean = model_prediction


fig, ax = plt.subplots(figsize=(10, 6))
ax.scatter(X_test, y_test, color='steelblue', label='True values', s=100,␣
 ↪alpha=0.7, edgecolor='black')
ax.plot(X_test, model_prediction, color='orange', linewidth=3, label='Predicted␣
 ↪values')

ax.spines['right'].set_color('0.8')
ax.spines['bottom'].set_color('0.6')
ax.yaxis.label.set_color('0.2')
ax.xaxis.label.set_color('0.2')
```

```
bar_color = plt.gcf()
plt.grid(color='gray', linestyle='--', linewidth=0.5)
color = plt.gcf().set_facecolor('lightgray')
plt.legend(fontsize=17)
ax.set_xlabel('Totalvikt', fontsize=16)
ax.set_ylabel('Miles per gallon', fontsize=16)
ax.set_title('Model evaluation', fontsize=24)
```

[ ]: Text(0.5, 1.0, 'Model evaluation')



0.9 Slutsatser och analys

0.9.1 linjär regression och tolkning av koefficienter

- Vi har funnit att 'Totalvikt' kan förutsäga 'mpg' med en R^2-värde på 0.74, vilket indikerar att 74% av variansen i 'mpg' kan förklaras av 'Totalvikt'.
  - Koefficienten för 'Totalvikt' är negativ och signifikant ($p < 0.05$), vilket indikerar att tyngre fordon har lägre bränsleeffektivitet.
  - Detta innebär att för varje ökning av 'Totalvikt' med en enhet minskar 'mpg' med 0.0076 enheter.
  - Detta stöder vår hypotes om att tyngre fordon har lägre bränsleeffektivitet.
- Vi förkastar nollhypotesen och accepterar att det finns ett signifikant samband mellan 'mpg' och 'Totalvikt'.
- Detta innebär att tyngre fordon har lägre bränsleeffektivitet, vilket kan bli en chock för

många.
- Vår analys stöder hypotesen om att tyngre fordon har lägre bränsleeffektivitet.

0.9.2 Slutsatser

- Vi har funnit att det finns ett signifikant negativt samband mellan 'mpg' och 'Totalvikt'.
- Tyngre fordon har lägre bränsle-effektivitet, vilket stöder vår hypotes.
- Linjär regressionsanalys visar att 'Totalvikt' kan förutsäga 'mpg' med en R^2-värde på 0.74, vilket indikerar att 74% av variansen i 'mpg' kan förklaras av 'Totalvikt'.
- Vi förkastar nollhypotesen och accepterar att det finns ett signifikant samband mellan 'mpg' och 'Totalvikt'.
- Detta innebär att tyngre fordon har lägre bränsle-effektivitet, vilket lär bli en chock för många.

0.9.3 Konklusion

- Vår analys stöder hypotesen om att tyngre fordon har lägre bränsleeffektivitet. Vi har funnit ett signifikant negativt samband mellan 'mpg' och 'Totalvikt', vilket indikerar att tyngre fordon har lägre bränsleeffektivitet, jag inväntar mitt nobelpris.

0.9.4 Begränsningar

- Analysen är baserad på ett specifikt dataset och kan inte generaliseras till alla fordon.
- Andra faktorer, såsom motoreffekt, aerodynamik och däcktyp, kan också påverka bränsleeffektiviteten, och dessa faktorer har inte undersökts i vår analys.
- Hypotesen är isolerat till en specifik tidsperiod och kan inte generaliseras till andra tidsperioder då teknologin kan ha utvecklats.

Prof. Deep. Analytica