Oppgave 1 kode

November 18, 2024

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from scipy import stats
import statsmodels.formula.api as smf
import statsmodels.api as sm
```

[2]:
```
df = pd.read_csv("lego.population.csv", sep = ",", encoding = "latin1")

df
```

[2]:

	Item_Number	Set_Name	Theme	Pieces	\
0	41916	Extra Dots - Series 2	DOTS	109.0	
1	41908	Extra Dots - Series 1	DOTS	109.0	
2	11006	Creative Blue Bricks	Classic	52.0	
3	11007	Creative Green Bricks	Classic	60.0	
4	41901	Funky Animals Bracelet	DOTS	33.0	
…	…	…	…	…	
1299	45678	SPIKE Prime Set	LEGO® Education	528.0	
1300	71367	Mario's House & Yoshi	LEGO® Super Mario	205.0	
1301	71368	Toad's Treasure Hunt	LEGO® Super Mario	464.0	
1302	71369	Bowser's Castle Boss Battle	LEGO® Super Mario	1010.0	
1303	71371	Propeller Mario Power-Up Pack	LEGO® Super Mario	13.0	

	Price	Amazon_Price	Year	Ages	Pages	Minifigures	Packaging	\
0	$3.99	$3.44	2020	Ages_6+	NaN	NaN	Foil pack	
1	$3.99	$3.99	2020	Ages_6+	NaN	NaN	Foil pack	
2	$4.99	$4.93	2020	Ages_4+	37.0	NaN	Box	
3	$4.99	$4.93	2020	Ages_4+	37.0	NaN	Box	
4	$4.99	$4.99	2020	Ages_6+	NaN	NaN	Foil pack	
…	…	…	…	…	…	…		
1299	$329.95	NaN	2020	Ages_10+	NaN	2.0	NaN	
1300	$29.99	NaN	2020	Ages_6+	NaN	2.0	Box	
1301	$69.99	NaN	2020	Ages_8+	NaN	4.0	Box	
1302	$99.99	NaN	2020	Ages_8+	NaN	NaN	Box	

```
1303    $9.99           NaN  2020    Ages_6+    NaN          NaN        Box

     Weight  Unique_Pieces Availability   Size
0       NaN            6.0       Retail  Small
1       NaN            6.0       Retail  Small
2       NaN           28.0       Retail  Small
3       NaN           36.0       Retail  Small
4       NaN           10.0       Retail  Small
…     …             …          …     …
1299    NaN          108.0          NaN  Small
1300    NaN          114.0       Retail  Small
1301    NaN          195.0       Retail  Small
1302    NaN          346.0       Retail  Small
1303    NaN           11.0       Retail  Small

[1304 rows x 15 columns]
```

[3]:
```
# fjerner forklaringsvariabler vi ikke trenger
df2 = df[['Set_Name', 'Theme', 'Pieces', 'Price', 'Pages',  'Unique_Pieces']]

# fjerner observasjoner med manglende datapunkter
df2 = df2.dropna()

# gjør themes om til string og fjern alle tegn vi ikke vil ha med
df2['Theme'] = df2['Theme'].astype(str)
df2['Theme'] = df2['Theme'].str.replace(r'[^a-zA-Z0-9\s-]', '', regex = True)

# fjerner dollartegn og trademark-tegn fra datasettet
df2['Price'] = df2['Price'].str.replace('\$', '', regex = True)

# og gjør så prisen om til float
df2['Price'] = df2['Price'].astype(float)

# det er dataset dere skal bruke!
df2
```

[3]:
```
                          Set_Name            Theme  Pieces   Price  Pages  \
2              Creative Blue Bricks          Classic    52.0    4.99   37.0
3             Creative Green Bricks          Classic    60.0    4.99   37.0
11                       Fire Truck            DUPLO     6.0    6.99    3.0
12                       Tow Truck            DUPLO     7.0    6.99    3.0
13     Stephanie's Summer Heart Box          Friends    95.0    7.99   40.0
…                              …             …     …     …      …
1173     Welcome to Apocalypseburg!  THE LEGO MOVIE 2  3178.0  299.99  452.0
1174  Jurassic Park: T. rex Rampage     Jurassic World  3120.0  249.99  464.0
1175    Monkie Kid's Team Secret HQ         Monkie Kid  1105.0  169.99  556.0
1176                   Grand Piano              Ideas  3662.0  349.99  564.0
```

```
1177        Lamborghini Sián FKP 37           Technic  3696.0  379.99  657.0

      Unique_Pieces
2              28.0
3              36.0
11              6.0
12              7.0
13             52.0
…               …
1173           692.0
1174           525.0
1175           622.0
1176           345.0
1177           293.0

[922 rows x 6 columns]
```

[4]:
```
#Definerer Temaer med spillassosiasjon
df2['HasGames'] = np.where(df2['Theme'].isin(['The Lego Movie 2', 'Friends',␣
 ↪'Marvel', 'Creator 3-in-1',
                                               'Harry Potter', 'Batman',␣
 ↪'Creator Expert', 'DC',
                                               'NINJAGO', 'Star Wars', 'City',␣
 ↪'Jurassic World',
                                               'Minifigures']), 'yes', 'no')
df2.groupby(['HasGames']).size().reset_index(name='Count')
```

[4]:
```
  HasGames  Count
0       no    367
1      yes    555
```

[5]:
```
# Definerer lisensierte temaer
licensed_themes = [
    "Star Wars", "Marvel", "Disney", "The Lego Movie 2", "Minecraft",
    "Harry Potter", "Jurassic World", "Speed Champions", "Batman", "DC",␣
 ↪"Trolls World Tour",
    "Overwatch", "LEGO Frozen 2", "Spider-Man", "Powerpuff Girls", "Minions",
    "Stranger things"
]

# Oppretter en ny kolonne 'Licensed' i datasettet
df2['Licensed'] = np.where(df2['Theme'].isin(licensed_themes), 'yes', 'no')

# Sjekker resultatet
df2[['Theme', 'Licensed']].head()
```

[5]:
```
      Theme Licensed
2   Classic       no
3   Classic       no
11    DUPLO       no
12    DUPLO       no
13  Friends       no
```

[6]:
```
# Lager dummy-variabler for 'HasGames' og 'Licensed', og fjerner en kategori␣
 ↪for å unngå multikollinearitet
df2 = pd.get_dummies(df2, columns=['HasGames', 'Licensed'], drop_first=False)

if 'HasGames_no' in df2.columns:
    df2 = df2.drop(columns=['HasGames_no'])

if 'Licensed_no' in df2.columns:
    df2 = df2.drop(columns=['Licensed_no'])

# Sjekker at kun de ønskede dummy-variablene er igjen
print(df2.columns)
```

```
Index(['Set_Name', 'Theme', 'Pieces', 'Price', 'Pages', 'Unique_Pieces',
       'HasGames_yes', 'Licensed_yes'],
      dtype='object')
```

[7]:
```
# Legger til interaksjonsvariabel
df2['HasGames_Licensed'] = df2['HasGames_yes'] * df2['Licensed_yes']

X = df2[['Pieces', 'HasGames_yes', 'Unique_Pieces', 'Pages', 'Licensed_yes',␣
 ↪'HasGames_Licensed']]
y = df2['Price']

# Konverterer boolske kolonner til float
X['HasGames_yes'] = X['HasGames_yes'].astype(float)
X['Licensed_yes'] = X['Licensed_yes'].astype(float)
X['HasGames_Licensed'] = X['HasGames_Licensed'].astype(float)

# Legger til konstantleddet i modellen
X = sm.add_constant(X)

# Kjører regresjonsanalysen
model = sm.OLS(y, X).fit()

# Viser oppsummering av regresjonsresultater
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
```

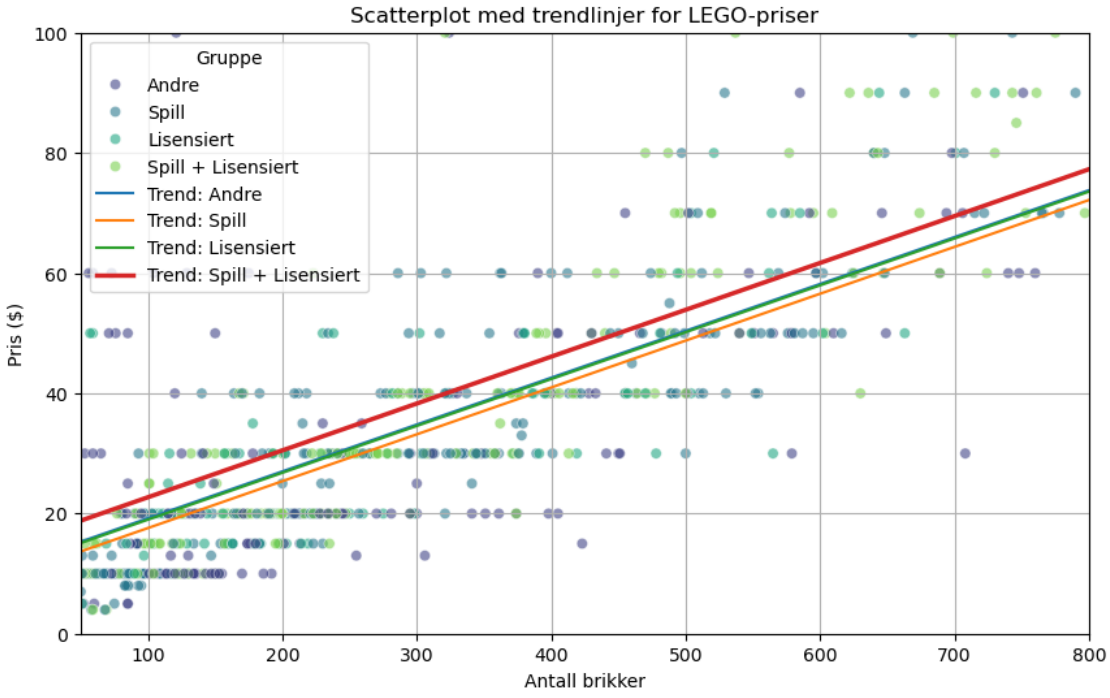
```
Dep. Variable:                  Price   R-squared:                       0.859
Model:                            OLS   Adj. R-squared:                  0.858
Method:                 Least Squares   F-statistic:                     925.9
Date:                Sun, 17 Nov 2024   Prob (F-statistic):               0.00
Time:                        23:55:05   Log-Likelihood:                -4149.4
No. Observations:                 922   AIC:                             8313.
Df Residuals:                     915   BIC:                             8347.
Df Model:                           6
Covariance Type:            nonrobust
================================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-----
const                5.8737      1.647      3.567      0.000       2.642
9.105
Pieces               0.0780      0.002     31.898      0.000       0.073
0.083
HasGames_yes        -2.9468      1.897     -1.554      0.121      -6.669
0.776
Unique_Pieces        0.0274      0.011      2.411      0.016       0.005
0.050
Pages                0.0345      0.011      3.147      0.002       0.013
0.056
Licensed_yes        -0.3868      2.492     -0.155      0.877      -5.278
4.504
HasGames_Licensed    6.3719      3.151      2.022      0.043       0.188
12.556
==============================================================================
Omnibus:                      831.387   Durbin-Watson:                   1.763
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            70115.004
Skew:                           3.693   Prob(JB):                         0.00
Kurtosis:                      45.078   Cond. No.                     4.42e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.42e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
/tmp/ipykernel_455/2686860798.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
  X['HasGames_yes'] = X['HasGames_yes'].astype(float)
/tmp/ipykernel_455/2686860798.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['Licensed_yes'] = X['Licensed_yes'].astype(float)
/tmp/ipykernel_455/2686860798.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['HasGames_Licensed'] = X['HasGames_Licensed'].astype(float)
```

[19]:
```python
# Denne koden er generert med hjelp fra ChatGPT 4o

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Filtrer data for å fokusere på relevante verdier
filtered_df = df2[(df2['Pieces'] >= 50) & (df2['Pieces'] <= 800)]

# Opprett en scatterplot
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x='Pieces', y='Price', hue='Group', data=filtered_df, alpha=0.6,␣
 ↪palette='viridis'
)

# Legg til trendlinjer basert på modellen
for group in filtered_df['Group'].unique():
    subset = filtered_df[filtered_df['Group'] == group]
    avg_unique_pieces = subset['Unique_Pieces'].mean()
    avg_pages = subset['Pages'].mean()

    # Simuler linje for Pieces
    pieces = np.linspace(50, 2000, 100)
    has_games = 1 if group in ['Spill', 'Spill + Lisensiert'] else 0
    licensed = 1 if group in ['Lisensiert', 'Spill + Lisensiert'] else 0
    interaction = has_games * licensed

    predicted_prices = (
        model.params['const'] +
        model.params['Pieces'] * pieces +
```

```
        model.params['HasGames_yes'] * has_games +
        model.params['Licensed_yes'] * licensed +
        model.params['HasGames_Licensed'] * interaction +
        model.params['Unique_Pieces'] * avg_unique_pieces +
        model.params['Pages'] * avg_pages
    )

    # Legg til trendlinje med tydelig farge og linjestil
    plt.plot(
        pieces, predicted_prices, label=f"Trend: {group}",
        linewidth=2.5 if group == "Spill + Lisensiert" else 1.5,
    )

# Juster aksene for zoom og klarhet
plt.xlim(50, 800)
plt.ylim(0, 100)

# Tilpass grafen
plt.title("Scatterplot med trendlinjer for LEGO-priser")
plt.xlabel("Antall brikker")
plt.ylabel("Pris ($)")
plt.legend(title="Gruppe", loc="upper left")
plt.grid(True)
plt.show()
```



[ ]:

[ ]: