Projektna naloga za OVS

Projektna naloga

Šolsko leto 2019/20

Ime in priimek: Jan Karlovčec
Vpisna številka: 04190015
Podatkovni niz: avcena.csv

Izjava o avtorstvu

Izjavljam, da sem nalogo samostojno izdelal.

1 Opis podatkov in raziskovalne domneve

Izmerili smo prodajno ceno 57 avtomobilov. Pri tem smo izvedeli 3 lastnosti. Model avtomobila ter njegovo prodajno in nabavno ceno. Glavna raziskovalna domneva je, da obstaja med nabavno in prodajno ceno linearna funkcijska zveza.

2 Opisna statistika

Zdaj bomo izračunali opisno statistiko – povzetek s petimi števili (minimum, maksimum, prvi in tretji kvartil, mediano) in povprečno vrednost mase in porabe goriva. Za prikaz opisne statistike lahko uporabimo funkcijo summary.

Naprej bom importiral knjižnico "pandas", da bom lahko prebral podatke.

In [9]:

```
import pandas
```

In [10]:

```
data = pandas.read_csv(r"C:\Users\Jan\Desktop\podatki_avcena.csv")
```

In [11]:

```
tabela = pandas.DataFrame(data, columns =["model", "nabcena", "prodcena"])
tabela
```

Out[11]:

	model	nabcena	prodcena
0	Chevrolet Aveo	10.965	11.690
1	Chevrolet Cavalier	13.884	14.810
2	Dodge Neon SE	12.849	13.670
3	Dodge Neon SXT	14.086	15.040
4	Ford Focus LX	12.906	13.730
5	Ford Focus SE	14.496	15.460
6	Ford Focus ZX5	14.607	15.580
7	Honda Civic LX	14.531	15.850
8	Hyundai Accent GL	11.116	11.839
9	Hyundai Elantra GLS	12.781	13.839
10	Hyundai Elantra GT	14.207	15.389
11	Kia Optima LX	14.910	16.040
12	Kia Rio	9.875	10.280
13	Kia Spectra	11.630	12.360
14	Mini Cooper	15.437	16.999
15	Nissan Sentra 1.8	12.205	12.740
16	Saturn Ion1	10.319	10.995
17	Saturn Ion2	13.393	14.300
18	Saturn Ion3	14.811	15.825
19	Suzuki Aeno S	12.719	12.884
20	Suzuki Aerio LX	14.317	14.500
21	Suzuki Forenza S	12.116	12.269
22	Suzuki Forenza EX	15.378	15.568
23	Toyota Corolla CE	13.065	14.085
24	Toyota Corolla S	13.650	15.030
25	Toyota Corolla LE	13.889	15.295
26	Toyota Echo	10.642	11.290
27	Chevrolet Impala	20.095	21.900
28	Chevrolet Malibu	17.434	18.995
29	Chevrolet Malibu LS	18.639	20.370
30	Chrysler PT Cruiser	16.919	17.985
31	Chrysler Sebring	17.805	19.090
32	Dodge Intrepid SE	20.502	22.035
33	Dodge Stratus SXT	17.512	18.820
34	Dodge Stratus SE	18.821	20.220
35	Ford Focus SVT	17.878	19.135
36	Ford Taurus LX	18.881	20.320

	model	nabcena	prodcena
37	Ford Taurus SES Duratec	20.857	22.735
38	Honda Civic EX	16.265	17.750
39	Honda Civic Hybrid	18.451	20.140
40	Hyundai Sonata GLS	17.574	19.339
41	Hyundai Sonata LX	18.380	20.339
42	Kia Optima LX V6	16.850	18.435
43	Mazda6 I	17.817	19.270
44	Mini Cooper S	18.137	19.999
45	Nissan Altima S	18.030	19.240
46	Nissan Sentra SE-R	16.444	17.640
47	Pontiac Grand Prix GT1	20.545	22.395
48	Saturn L300-2	19.801	21.410
49	Subaru Impreza 2.5 RS	18.399	19.945
50	Subaru Legacy L	18.713	20.445
51	Suzuki Verona LX	17.053	17.262
52	Toyota Camry LE	17.558	19.560
53	Toyota Camry LE V6	20.325	22.775
54	Toyota Prius	18.926	20.510
55	Volkswagen Golf GLS	17.478	18.715
56	Volkswagen Jetta GLS TDI	19.638	21.055

Ko sem podatke spravil v tabelo sem lahko začel računati opisno statistiko. Za izračun opisne statistike sem uporabil funkcijo "describe()", ki je del knjižnice pandas.

In [12]:

```
data.nabcena.describe()
```

Out[12]:

```
count    57.000000
mean     15.903702
std       2.997093
min       9.875000
25%      13.650000
50%      16.444000
75%      18.380000
max      20.857000
Name: nabcena, dtype: float64
```

Pri nabavni ceni vidimo da je minumum 9.875 maksimum pa 20.857. Prvi kvartil je pri 13.65 tretji pa pri 18.38. Mediana znaša 16.44 povprečje pa 15.904. Standardni odklon pa 3.000.

In [56]:

```
data.prodcena.describe()
```

Out[56]:

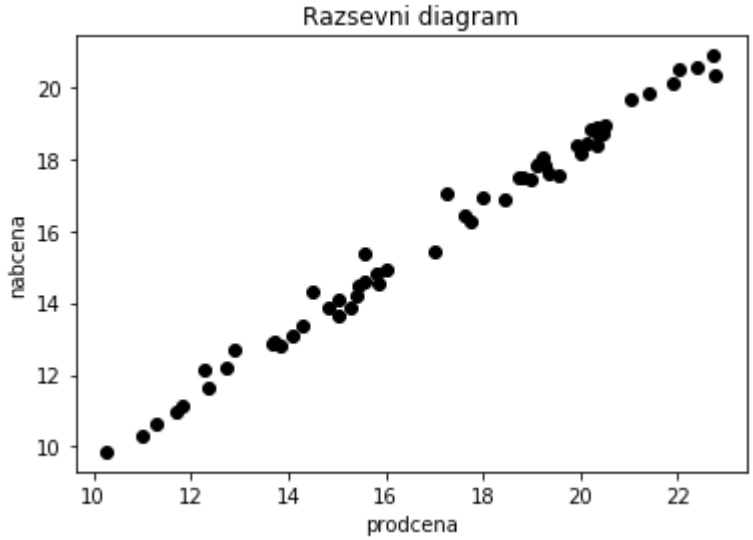
```
count    57.000000
mean     17.108526
std       3.408276
min      10.280000
25%      14.500000
50%      17.640000
75%      19.999000
max      22.775000
Name: prodcena, dtype: float64
```

Pri prodajni ceni vidimo da je minumum 10.28 maksimum pa 22.775. Prvi kvartil je pri 14.5 tretji pa pri 19.999. Mediana znaša 17.64 povprečje pa 17.109. Standardni odklon pa 3.408.

3 Razsevni diagram in vzorčni koeficient korelacije

In [17]:

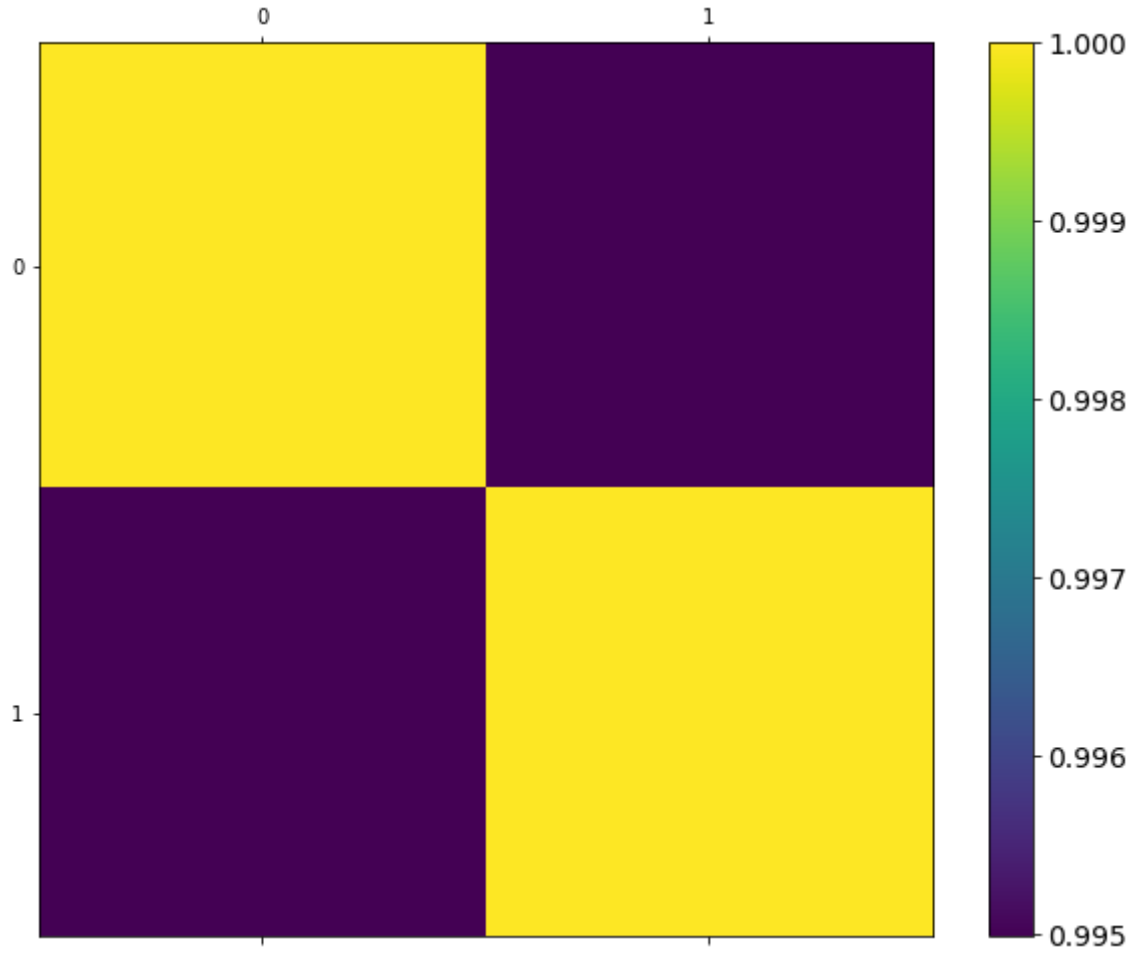
```
import matplotlib.pyplot as mpl
prodcena=data["prodcena"]
nabcena=data["nabcena"]
mpl.scatter(prodcena, nabcena, color='black')
mpl.title('Razsevni diagram')
mpl.xlabel('prodcena')
mpl.ylabel('nabcena')
mpl.show()
```



Točke na razsevnem diagramu se nahajajo okoli namišljene premice, tako da je linearni model primeren.

In [19]:

```
f = mpl.figure(figsize=(10, 8),facecolor="white")
mpl.matshow(tabela.corr(), fignum=f.number)
cb = mpl.colorbar()
cb.ax.tick_params(labelsize=14)
mpl.show()
```



In [20]:

```
tabela.corr()
```

Out[20]:

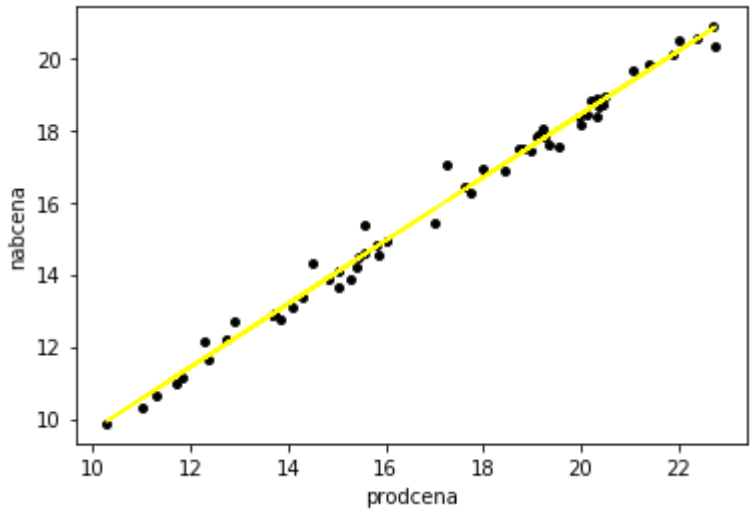
	nabcena	prodcena
nabcena	1.000000	0.994988
prodcena	0.994988	1.000000

Kot vidimo je koeficient korelacije blizu 1 kar pomeni, da sta spremeljivki zelo odvisni ena od druge.

4 Formiranje linearnegaregresijskega modela in preverjanje njegovih predpostavk

In [59]:

```
import numpy as np
import matplotlib.pyplot as plt

def lin(x, y):
    dolzina = np.size(x)

    mx, my = np.mean(x), np.mean(y)

    xy = np.sum(y*x) - dolzina*my*mx
    xx = np.sum(x*x) - dolzina*mx*mx

    b1 = xy / xx
    b0 = my - b1*mx

    return(b0, b1)

def linija(x, y, b):
    plt.scatter(x, y, color = "black",
                marker = "o", s = 15)

    y_pred = b[0] + b[1]*x

    plt.plot(x, y_pred, color = "yellow")

    plt.xlabel('prodcena')
    plt.ylabel('nabcena')

    plt.show()

def main():
    x = np.array(prodcena)
    y = np.array(nabcena)

    b = lin(x, y)

    linija(x, y, b)

if __name__ == "__main__":
    main()
```



Za računanje linearnegaregresijskega modela sem uporabil funkcijo numpy in matplotlib. Začel sem z izračunom dolžine in povprečja obeh sestavin. Nato izračunal oba odklona ter regresijski koeficient. Nato je bilo potrebno še use podatke izrisati z premico. Vsi podatki, ki odstopajo od rumene premice naj bi bili kot nepravilni.

5 Testiranje linearnosti regresijskega modela in koeficient determinacije

In [51]:

```
import scipy as sc
slope, intercept, r_value, p_value, std_err = sc.stats.linregress(nabcena, prodcena)
print(r_value**2)
```

```
0.9900006597050255
```

In [55]:

```
from scipy import stats
stats.ttest_ind(nabcena, prodcena)
```

Out[55]:

```
Ttest_indResult(statistic=-2.0041911207769667, pvalue=0.04746288893803342)
```

Na osnovi rezultatov t-testa smo potrdili, da linearni model ustreza podatkom. Koeficient determinacije je enak R2 = 0.990, kar pomeni, da 99% podatkov potrjuje linearni regresijski model.

6 Intervala zaupanja za naklon in odsek regresijske premice

In [50]:

```
from scipy.stats import t

zaupanje = 0.99
x = np.mean(nabcena)
delta = 2.997093 * t(df = len(nabcena)-1).ppf((1+zaupanje)/2) / (len(nabcena) ** (1/2))

spodnameja = x - delta
zgornameja = x + delta
print(x*1000 , "$")
print(spodnameja*1000, "$")
print(zgornameja*1000, "$")
```

```
15903.701754385962 $
14845.163887145713 $
16962.23962162621 $
```

In [41]:

```
from scipy.stats import t

zaupanje = 0.99
x = np.mean(prodcena)
delta = 2.997093 * t(df = len(prodcena)-1).ppf((1+zaupanje)/2) / (len(prodcena) ** (1/2
))

spodnameja = x - delta
zgornameja = x + delta
print(x*1000 , "$")
print(spodnameja*1000, "$")
print(zgornameja*1000, "$")
```

```
17108.526315789477 $
16049.988448549226 $
18167.064183029728 $
```

Izračunal sem 99% interval zaupanja s pomočjo funkcije scipy.

7 Interval predikcije za vrednost Y pri izbrani vrednosti X

In [32]:

```
povp1 = np.mean(data["nabcena"])
povp2 = np.mean(data["prodcena"])

s = (data["nabcena"] - povp1) * (data["prodcena"] - povp2)
i = (data["nabcena"] - povp1)**2

beta = s.sum() / i.sum()
alpha = povp2 - (beta * povp1)
```

In [34]:

```
cena = 15000

predvidenacena = alpha + beta * cena
print(predvidenacena,"$")
```

```
16971.524730299585 $
```

Sprva sem izračunal alfo in beto, ki mi bosta pomagali pri izračunu predikcije. Za izbrano vrednost X sem si izbral 15000 ameriških dolarjev. Pri tej ceni je po moji predikciji vrednost prodajne cene 16971.52 ameriških dolarjev.