Mappeoppgave 5 a)

April 1, 2022

1 Mappeoppgave 5 sok-1005

Denne oppgaven har vi (Kenneth Andreassen, Falk Falkum og Christian Pettersen) valgt å skrape skattelistene fra E24 for Oslo kommune (2020) for å se om det er noen sammenheng mellom inntekt og skatt.

[1]:
```
from bs4 import BeautifulSoup
import requests

def fetch_html_tables(url):
    "Returns a list of tables in the html of url"
    page = requests.get(url)
    bs=BeautifulSoup(page.content)
    tables=bs.find_all('table')
    return tables

tables=fetch_html_tables('https://e24.no/spesial/skattelister/2020/0301/')
table_html=tables[0]

#printing top
print(str(table_html)[:1000])
```

```
<table class="table table-sm"><thead><tr><th class="w-5"></th><th
class="w-30">Navn</th><th class="text-right clickable">Inntekt<i
class="material-icons md-14 middle"> </i></th><th class="text-right
clickable">Formue</th><th class="text-right
clickable">Skatt</th></tr></thead><tbody><tr><td>1<!-- -->.</td><td><div
class="name">ØYSTEIN STRAY<!-- --> <!-- -->SPETALEN</div><div class="text-
muted"><a class="text-muted" href="/spesial/skattelister/2020/0301/">Oslo</a>,
f.<!-- --> <!-- -->1962</div></td><td class="text-right">258 709 731</td><td
class="text-right">2 586 723 237</td><td class="text-
right">105 131 374</td></tr><tr><td>2<!-- -->.</td><td><div
class="name">ØYSTEIN<!-- --> <!-- -->MOAN</div><div class="text-muted"><a
class="text-muted" href="/spesial/skattelister/2020/0301/">Oslo</a>, f.<!--
--> <!-- -->1959</div></td><td class="text-right">181 404 420</td><td
class="text-right">961 391 825</td><td class="text-
right">52 228 271</td></tr><tr><td>3<!-- -->.</td><td><div class="name
```

[2]:
```
def html_to_table(html):
    "Returns the table defined in html as a list"
    #defining the table:
    table=[]
    #iterating over all rows
    for row in html.find_all('tr'):
        r=[]
        #finding all cells in each row:
        cells=row.find_all('td')

        #if no cells are found, look for headings
        if len(cells)==0:
            cells=row.find_all('th')

        #iterate over cells:
        for cell in cells:
            cell=format(cell)
            r.append(cell)

        #append the row to t:
        table.append(r)
    return table

def format(cell):
    "Returns a string after converting bs4 object cell to clean text"
    if cell.content is None:
        s=cell.text
    elif len(cell.content)==0:
        return ''
    else:
        s=' '.join([str(c) for c in cell.content])

    #here you can add additional characters/strings you want to
    #remove, change punctuations or format the string in other
    #ways:
    s=s.replace("\xa0","")
    s=s.replace("\n","")
    s=s.replace("Oslo", "")
    s=s.replace("\ue5cf", "")
    s=s.replace("Navn", "")
    return s

table=html_to_table(table_html)

#printing top
print(str(table)[:1000])
```

```
[['', '', 'Inntekt', 'Formue', 'Skatt'], ['1.', 'ØYSTEIN STRAY SPETALEN,
f.1962', '258709731', '2586723237', '105131374'], ['2.', 'ØYSTEIN MOAN, f.1959',
'181404420', '961391825', '52228271'], ['3.', 'MAGNUS REITAN, f.1975',
'166542768', '4620795695', '92706050'], ['4.', 'OLE ROBERT REITAN, f.1971',
'162833343', '4546957558', '90924296'], ['5.', 'MARGARET BOEL GARMANN, f.1955',
'154930624', '2562173467', '70855059'], ['6.', 'JØRGEN DAHL, f.1969',
'131447948', '2790325759', '65814357'], ['7.', 'EDGAR HAUGEN, f.1965',
'112271516', '2777922578', '58914860'], ['8.', 'ERIK WILSON LANDGRAFF, f.1984',
'109748661', '100233911', '1459975'], ['9.', 'TOR ØIVIND FJELD, f.1979',
'103320599', '2743633213', '55434237'], ['10.', 'EILERT GIERTSEN HANOA, f.1970',
'101504287', '78400112', '33184365'], ['11.', 'ERIK KRISTOFFER ARTHUR, f.1962',
'90443514', '194726696', '30669028'], ['12.', 'ODD JOHNNY WINGE, f.1975',
'89197038', '441465476', '31968678'], ['13.', 'RAKESH PATEL, f.1975',
'73778983', '1342678
```

[3]:
```
f=open('skattoslo.csv','w')
";".join(table[0])
```

[3]:
```
';;Inntekt;Formue;Skatt'
```

[4]:
```
def save_data(file_name,table):
    "Saves table to file_name"
    f=open(file_name,'w')
    for row in table:
        f.write(';'.join(row)+'\n')
    f.close()

save_data('skattoslo.csv',table)
```

[5]:
```
import pandas as pd
pd.read_csv('skattoslo.csv', delimiter=';', encoding='latin1')
```

[5]:

	Unnamed: 0	Unnamed: 1	Inntekt	Formue	\
0	1.0	Ã YSTEIN STRAY SPETALEN, f.1962	258709731	2586723237	
1	2.0	Ã YSTEIN MOAN, f.1959	181404420	961391825	
2	3.0	MAGNUS REITAN, f.1975	166542768	4620795695	
3	4.0	OLE ROBERT REITAN, f.1971	162833343	4546957558	
4	5.0	MARGARET BOEL GARMANN, f.1955	154930624	2562173467	
5	6.0	JÃ RGEN DAHL, f.1969	131447948	2790325759	
6	7.0	EDGAR HAUGEN, f.1965	112271516	2777922578	
7	8.0	ERIK WILSON LANDGRAFF, f.1984	109748661	100233911	
8	9.0	TOR Ã IVIND FJELD, f.1979	103320599	2743633213	
9	10.0	EILERT GIERTSEN HANOA, f.1970	101504287	78400112	
10	11.0	ERIK KRISTOFFER ARTHUR, f.1962	90443514	194726696	
11	12.0	ODD JOHNNY WINGE, f.1975	89197038	441465476	
12	13.0	RAKESH PATEL, f.1975	73778983	13426780	
13	14.0	PÃ			

```
L ERIK SJÃ
TIL, f.1972   72857445   188862505
14        15.0             STEIN ERIK HAGEN, f.1956   72780634   433286845
15        16.0              KENNETH LÃ VOLD, f.1970   71412906           0
16        17.0  KRISTIAN GJERDRUM ROSENBERG, f.1962   67976007    53784556
17        18.0  NICOLAI HARALD LÃ VENSKIOLD, f.1962   66690101   103552557
18        19.0                 PETTER SOLUM, f.1954   64567647    28793539
19        20.0             KARL JOHAN SUNDE, f.1949   62920580   542148645

        Skatt
0   105131374
1    52228271
2    92706050
3    90924296
4    70855059
5    65814357
6    58914860
7     1459975
8    55434237
9    33184365
10   30669028
11   31968678
12   34149789
13   13496248
14   27864669
15   23024867
16   31891947
17   11340325
18   20667701
19   24335298
```

Her ser jeg at jeg ikke får opp rikrige bokster til navnene (Æ,Ø,Å) dette er da noe vi kunne forbedret i koden vår.

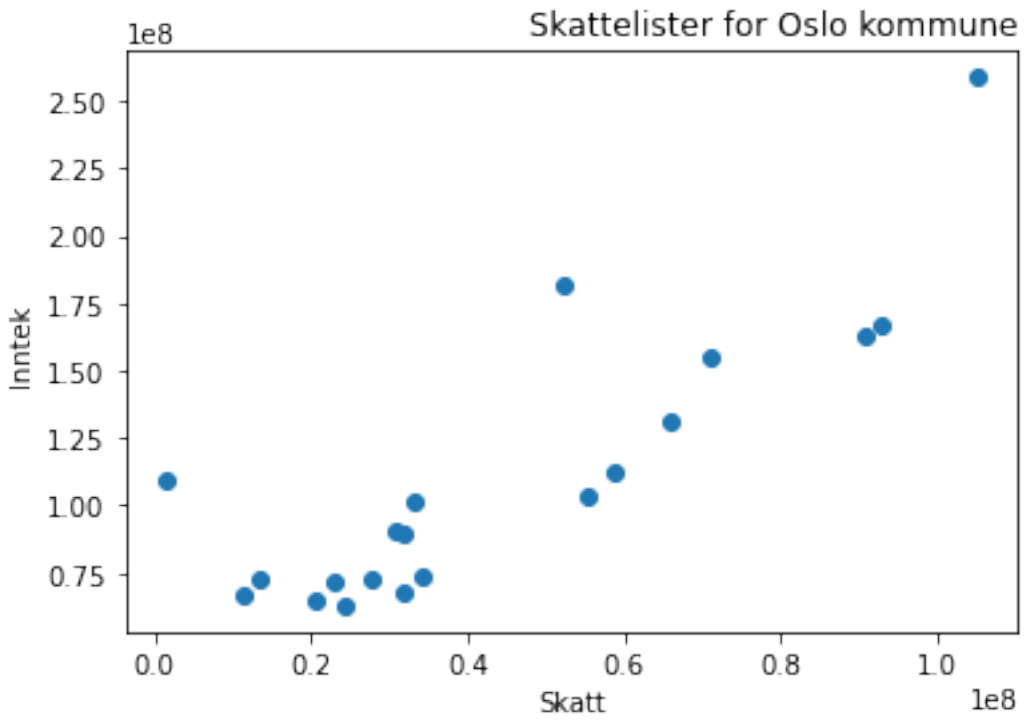
[6]:
```
from matplotlib import pyplot as plt
import numpy as np

data = pd.read_csv('skattoslo.csv', delimiter=';', encoding='UTF-8')
```

[7]:
```
fig,ax=plt.subplots()

ax.scatter(data["Skatt"], data["Inntekt"], label = "Observasjoner")
ax.set_ylabel("Inntek")
ax.set_xlabel('Skatt')
plt.title(label="Skattelister for Oslo kommune", loc="right")
```

[7]:
```
Text(1.0, 1.0, 'Skattelister for Oslo kommune')
```



Her får vi opp et plott som viser skattetoppen fra Oslo.

[8]:
```
y=data["Inntekt"] #Henter ut data til regresjon
pd.DataFrame(y)
```

[8]:

	Inntekt
0	258709731
1	181404420
2	166542768
3	162833343
4	154930624
5	131447948
6	112271516
7	109748661
8	103320599
9	101504287
10	90443514
11	89197038
12	73778983
13	72857445
14	72780634
15	71412906
16	67976007
17	66690101

```
18   64567647
19   62920580
```

Førstser vi på inntekten.

[9]:
```
x=pd.DataFrame(data["Skatt"])  #Henter ut data til regresjon og legger til en intercept
x["intercept"]=1
x
```

[9]:

	Skatt	intercept
0	105131374	1
1	52228271	1
2	92706050	1
3	90924296	1
4	70855059	1
5	65814357	1
6	58914860	1
7	1459975	1
8	55434237	1
9	33184365	1
10	30669028	1
11	31968678	1
12	34149789	1
13	13496248	1
14	27864669	1
15	23024867	1
16	31891947	1
17	11340325	1
18	20667701	1
19	24335298	1

Så ser vi på skattebeløpet vært individ måtte betale.

[10]:
```
import numpy as np
from matplotlib import pyplot as plt

from statsmodels.regression.linear_model import OLS

res=OLS(y,x).fit()

print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Inntekt   R-squared:                       0.715
Model:                            OLS   Adj. R-squared:                  0.699
Method:                 Least Squares   F-statistic:                     45.06
```

```
Date:                Fri, 01 Apr 2022   Prob (F-statistic):           2.71e-06
Time:                        16:43:45   Log-Likelihood:                -370.41
No. Observations:                  20   AIC:                             744.8
Df Residuals:                      18   BIC:                             746.8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Skatt          1.4934      0.222      6.713      0.000       1.026       1.961
intercept   4.535e+07   1.16e+07      3.908      0.001     2.1e+07    6.97e+07
==============================================================================
Omnibus:                        9.296   Durbin-Watson:                   1.766
Prob(Omnibus):                  0.010   Jarque-Bera (JB):                7.083
Skew:                           1.412   Prob(JB):                       0.0290
Kurtosis:                       3.725   Cond. No.                     9.61e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 9.61e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[11]:
```
res.params
```

[11]:
```
Skatt        1.493423e+00
intercept    4.535041e+07
dtype: float64
```

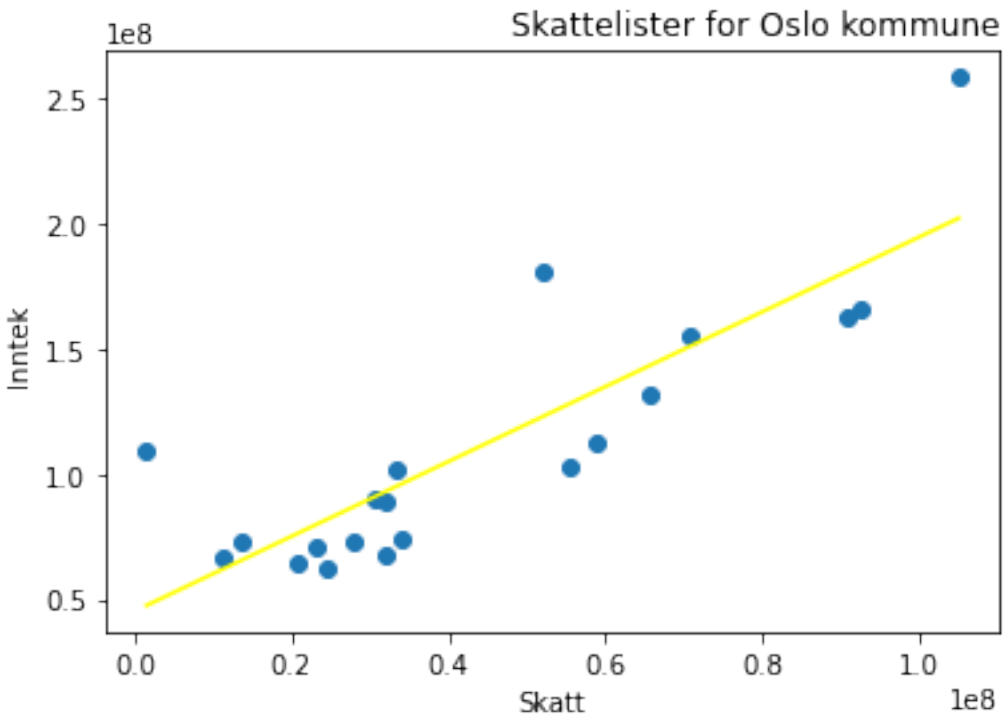
[12]:
```
x=np.linspace(min(data['Skatt']), max(data['Skatt']), 100)

regression_line=res.params['intercept']+res.params['Skatt']*x

ax.plot(x, regression_line , color='yellow')

fig
```

[12]:



Vi kan se at regresjonslinja følger godt med de meste punktene, men at noen av disse står litt utenfor. Vi kan dermed si at det er en sammenheng mellom inntekt, formue og skatt. Ettersom at det er det regresjonslinjen vår indikerer, men ettersom at det er noen prikker som står utenfor, så vil det si at man betaler forskjellig i skatt, fordi at det er forskjell på formueskatt, inntektskatt osv, denne grafen har da tattt utgangspunkt i skatt som noe generelt. Det er da grunnen til at prikkene ikke står i form som en linær graf. Derfor er det vansklig å se på. f.eks. sammenhengen mellom inntekt og skatt.

1.0.1 Kilder

Vi har hentet mye inspirasjonen på kodene fra https://github.com/espensirnes/notebooks.

Hentet koder fra:

https://github.com/espensirnes/notebooks/blob/main/10%20-%20statsmodels.ipynb

https://github.com/espensirnes/notebooks/blob/main/9%20-%20webskraping%20med%20python.ipynb

[ ]: