In [ ]:

```python
import pandas as pd
import numpy as np


data1 = pd.read_csv("Booli_sold.csv")
```

In [ ]:

```python
data1.head()
```

Apartment Prices

the ppsqm is **soldPrice / livingArea**
We want then to add the new **ppsqm** column to the original table .

In [ ]:

```python
squareMeters = data1.loc[:, "livingArea"] #select all rows : and a specific column "livi

soldPrice = data1.loc[:, "listPrice"] #select all rows : and a specific column "soldPric

ppsqm = round(soldPrice / squareMeters) #Round the number to the nearest integer

data1["ppsqm"] = ppsqm #Adding the new columnt to the original data set
```

In [ ]:

```python
data1.head()
```

Now the goal is to rank top 5 most expensive apartments w.r.t ppsqm. We want to sort the **ppsqm** by price and then see the 5 top candidates. Lets do that with the **sort** function and then print the head of our dataframe so that we can see top 5 candidates wrt ppsqm.

In [ ]:

```python
table_sorted_by_ppsqm = data1.sort_values("ppsqm", ascending=False)
table_sorted_by_ppsqm.head()
```

Now let us calculate the average ppsqm in Ekhagen. For this purpose we can just calculate the mean of the **ppsqw**. Note that the np.mean - function automatically ignores the missing values of ppsqw vector.

In [ ]:

```python
ekhagen_mean = round(np.mean(ppsqm))

print("The average ppsqm in Ekhagen is: " , ekhagen_mean )
```

What i find interesting about the data is that some values for the living area are missing which will naturally lead to that some values of the ppsqw will be missing (by a trivial reasons). However when calculating the mean of ppsqw we were able to preserve the sample size.

The Swedish Election of 2018

Calculate the total number of legitimate votes (Giltiga Röster) in Stockholm during the election. That is, sum upp the number of legitimate votes for all municipalities (kommun) in Stockholm.

In [ ]:

```python
data2 = pd.read_csv("2018_R_per_kommun.csv", sep = ";", decimal =",")

data2.head()
```

Deriving stockholm and legit votes

In [ ]:

```python
votes = data2.loc[:, ["LÄNSNAMN", "RÖSTER GILTIGA"]]
```

```
filtered_votes = votes[votes["LÄNSNAMN"].str.contains("Stockholms län", case=False, na=F

legit_votes_in_stockholm = filtered_votes.loc[:,"RÖSTER GILTIGA"]

legit_votes_in_stockholm_sum = sum(legit_votes_in_stockholm)

legit_votes_in_stockholm_sum
```

In which municipality did the social democratic party (Social demokraterna, S) garner the hightest voting percentage?

We can approach this by simply filtering the data in such a way that we only se manucipalitys and precentage that S got. BY later filtering the data some more so that we rearang rows so that the highest ranks first etc we will be able to clearly see the first one. And we can also use almost the exact same approach for calculating the top 3 municipalitys and presenting it in a table after.

In [ ]:

```
s = data2.loc[:, ["KOMMUNNAMN","S"]].sort_values(by = "S", ascending = False) #sortering

s.head()
```

In [ ]:

```
valdeltagande = data2.loc[:, ["KOMMUNNAMN", "VALDELTAGANDE"]].sort_values(by ="VALDELTAG

valdeltagande.head(3)
```

In [ ]:

```
#!jupyter nbconvert --to markdown HW2_new.ipynb
```

In [ ]:

```
```