ITMAL Øvelser - Uge 5

Øvelse 1

For this subexercise the group is to perform data analysis of the given dataset, namely "california housing prices". The subexercise consists of 5 parts which will be described herafter in chronological order. Before starting, the group needs to get the data read in the python workspace, to which is has been decided to store it as a pandas dataframe. This is done as pandas is build on numpy, and as such, has the same features as numpy with more.

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm
import pandas as pd
import math

# Load data - vægt data (kvinder/mænd)
data = pd.read_csv('.../Uge5_files/housing.csv', sep=',',header=0)
```

a)

For the first part we are to plot the distribution of the median_income data for the districts. This is done using matplotlibs *hist()* function. Futhermore, mean, variance, std. deviatien and median can all be found using numpy's build in methods.

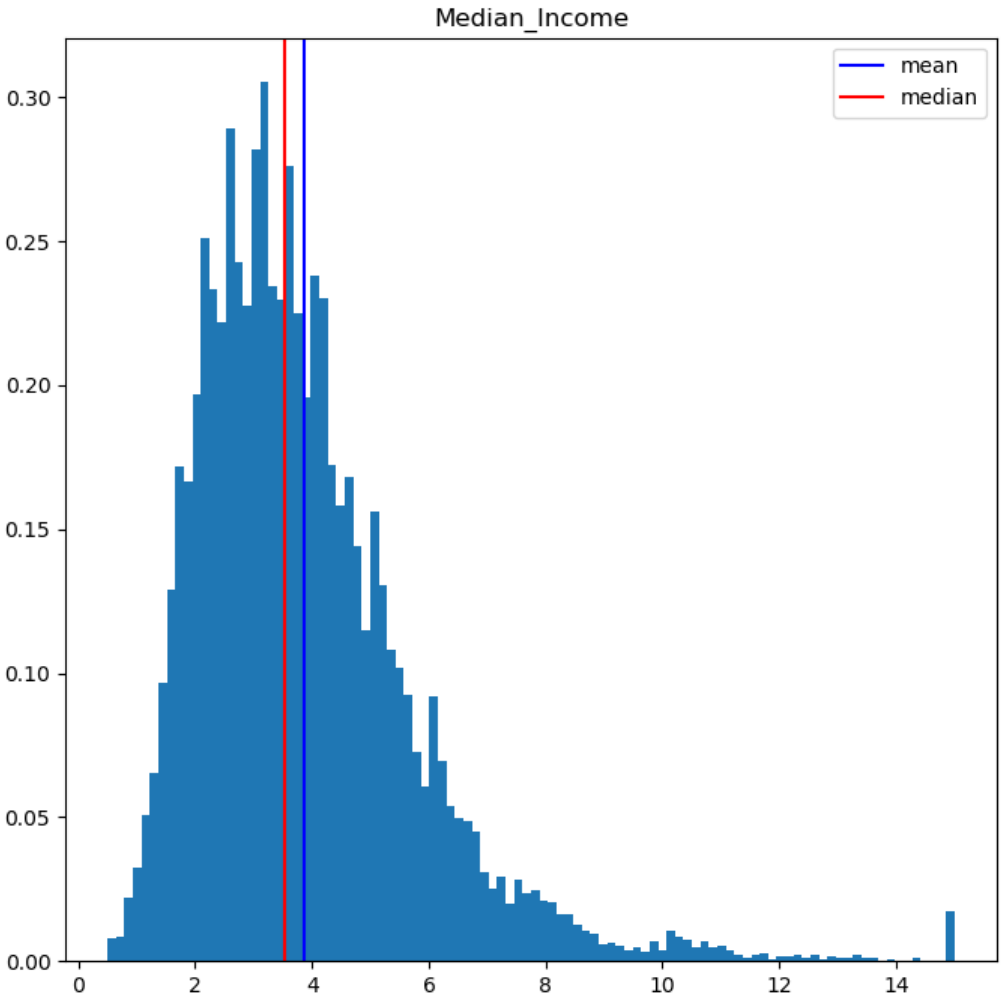
```
median_income = data['median_income'] # Extract column of df

mu = np.mean(median_income)         # Mean
sigma = np.std(median_income)       # Var
sigma2 = np.var(median_income)      # std deviatien
median = np.median(median_income)   # Median
print(f"Mean: {mu}, Varians {sigma2}, Std Deviatien {sigma}, Median {median}")
```

Yields:

```
Mean: 3.8706710029070246, Varians 3.60914768969746, Std Deviatien
1.899775694574878, Median 3.5347999999999997
```

With the plot using matplotlib:

```
fig, ax = plt.subplots(1, 1, figsize=[8,8])   # Create the plot object
ax.hist(median_income,bins=100, density=True) # Normalises the histogram
```



Median income distribution from california housing prices

From the above it is seen that the histogram somewhat resembles a normal distribution, although it must be noted that is rather tail heavy towards the higher median income values. Futhermore, it is noted that there is a rather large amount of districts towards the right with a extremely high median income - this and the heavy tail could explain the skewedness in the mean value compared to the median.

b)

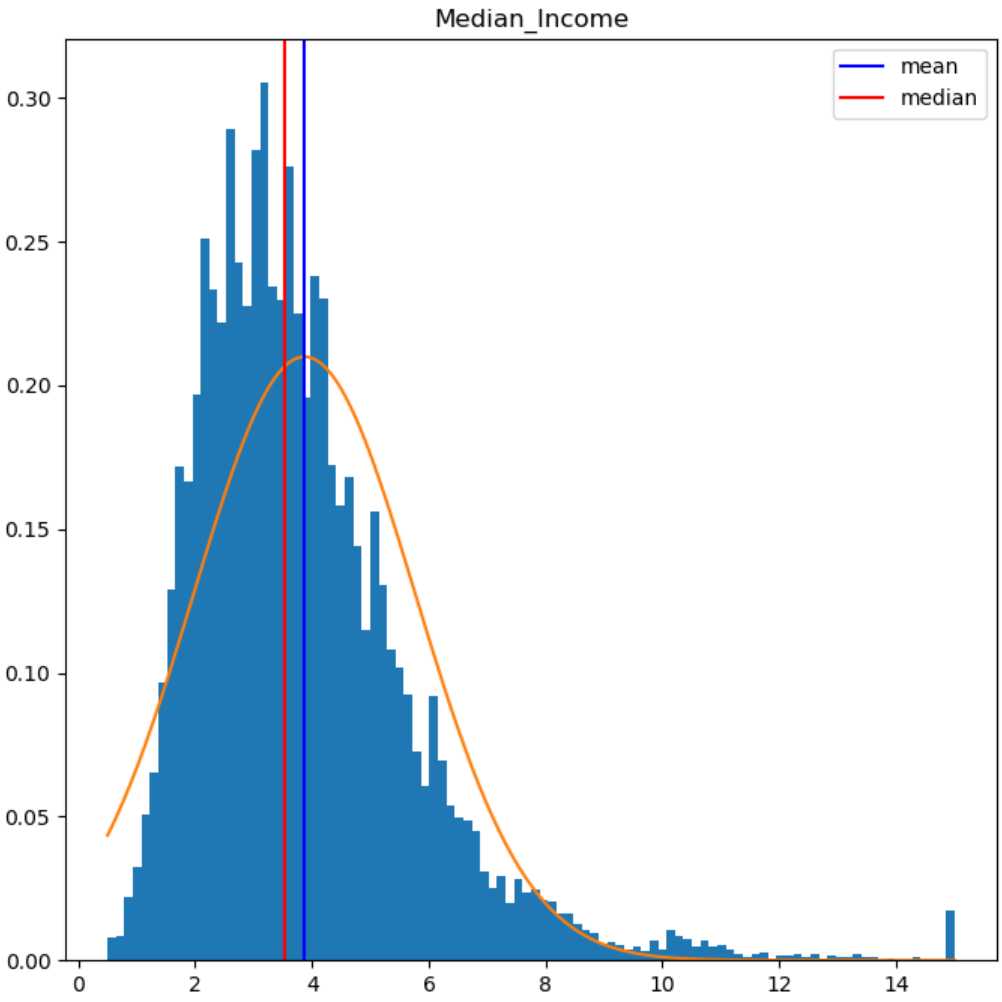
Now, looking at the mean and medianb, it is seen these are not identical. This is easily explained if "median_income" is seen as a column vector, where the mean is exactly the sum of all it's values, divided by it's length. The median however, is the value of the center index in the vector. So, which is the most telling? In the case of median income, the mean can be quite biased, as this is more easily influenced by outliers - for example the mean income of the population of a country might be heavily influenced by the top 1%, and therefore not very telling of the average citizen. The same goes for our example.

c)

For this part, a true normal distribution is fitted on top of the histogram - since we already normalised the histogram, this should be directly comparable:

```
myMax = np.max(median_income)
myMin = np.min(median_income)
xarr = np.linspace(myMax, myMin, 500)
ax.plot(xarr, norm.pdf(xarr, mu, sigma)) # Plot
```

Which yields the plot:



Median income distribution from california housin prices, now with a true normal distribution on top

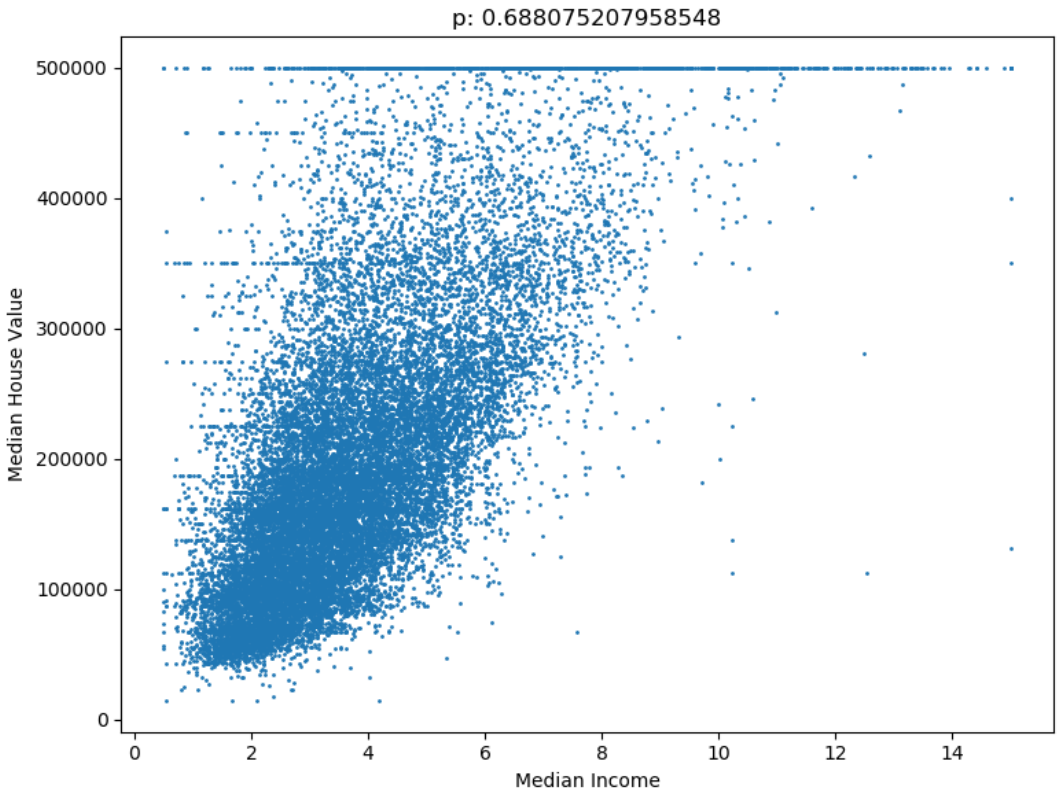
From which it is seen that our distribution does not follow that of a normal distribution, as it is too tail heavy, and as such, has shifted it's mean too far to the right. Furthermore, it would seem that our variance is also affected by the tail heavyness, as the true gaussian model does not encapsulate(reach) our top points around the median or mean.

d)

In order to determine if there is a correlation between median house value, and median income, this is computed and plotted:

```
# Check if there is a corelation between median income and median house value
medIncAndHVal = data[['median_income','median_house_value']]
corrcoef = np.corrcoef(medIncAndHVal.T) # obs: rækker=variable, kolonner=samples
(modsat normalt..)

plt.scatter(medIncAndHVal['median_income'], medIncAndHVal['median_house_value'],
s=1)

plt.title(f"p: {corrcoef[1,0]}")
plt.xlabel('Median Income')
plt.ylabel('Median House Value')
```

Which yields:



Correlation plot of median house value and income

From the above, a trend can be seen along the axis' as they seem to be increasing together(a linear upward trend, or positive correlation). The correlation coefficient(title of plot) verifies this. At the same time, some vague horisontal lines can be seen. These seem to limitations in the house value, which must stem from an unknown bias(or prerendition) of the data we are processing. The upmost horizontol line of the data plots seem to stem from an upper limit on the measurement collecting the data.

e)

For this part, the 5% and 95% percentile is to be found. This is done using numpy's build in functions like before:

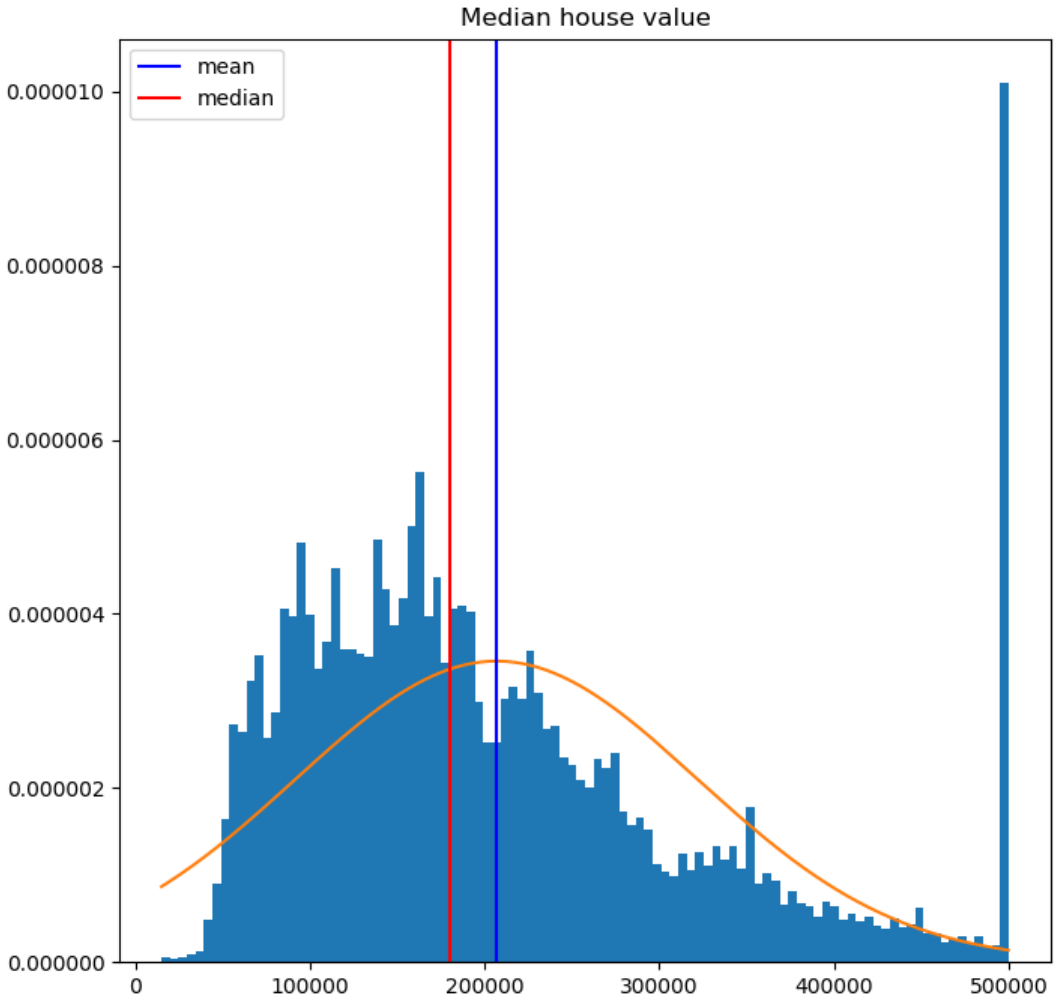
```
print("Fifth Percentile of med H val: ", np.percentile(data['median_house_value'],
5))

print("Ninety-fith Percentile of H val: ",
np.percentile(data['median_house_value'], 95))

print("With max: ", np.max(data['median_house_value']), " and Min: ",
np.min(data['median_house_value']))
```

Which yields:

```
Fifth Percentile of med H val:  66200.0
Ninety-fith Percentile of H val:  489809.9999999998
With max:  500001.0  and Min:  14999.0
```

Telling us that 5% of the districts has a median income of 66200 or less, and 95% has a median income of 489809 or less.

Using the same code as before, but with another feature from the dataset, median house value, a distribution can be found:



Median house value histogram

From the above, an obvious fault can be seen(the reason for the horizontol line before) at a median house value of 500000. Like said, this must stem from the data not being able to measure past this value, and as such, this value cannot be seen as representative, as these values can range from 500000 to ∞. A solution to this could be to remove all districts with this value, or simply assign them the median instead. Looking at the representation, e.g the numbers of samples with this value, the first seems the best, as the other would assign an artificial bias to the set. Also, disregarding the top value, the distribution does not seem to follow that of a gausian one, as this is heavily tail heavy towards the upper values - it seems more to resemble that of a Reyleigh distribution if one were to guess.