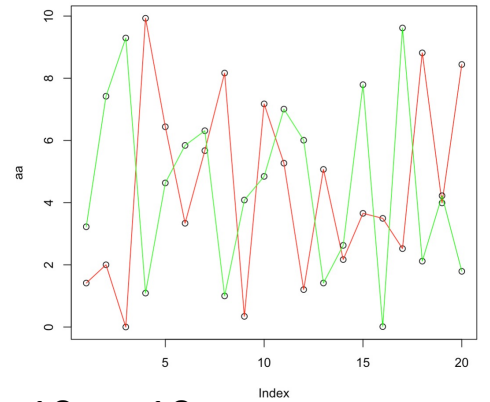
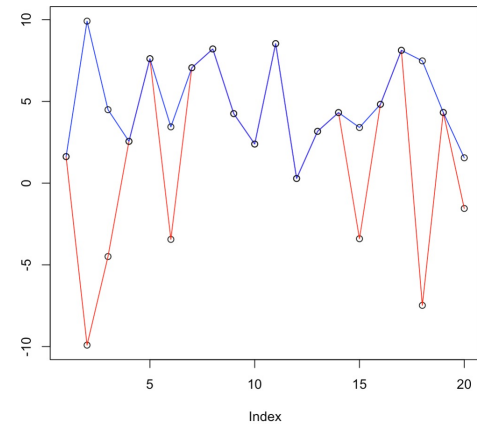


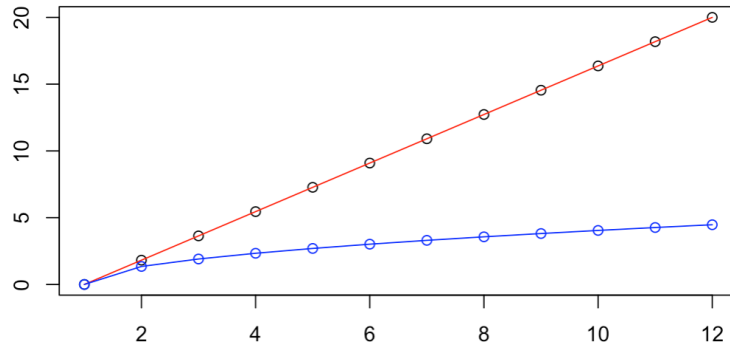
- 1) Brug runif til at lave to vektorer på 20 tilfældige tal mellem 0 og 20
- 2) Illustrer i en graf vha base R's graphics



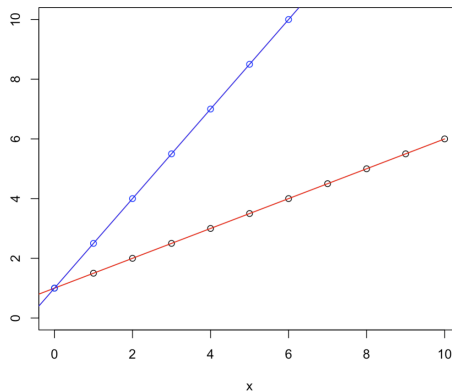
- 3) Lav en ny vektor som 1 men nu mellem -10 og 10.
- 4) Brug en R-funktion på vektoren så du får flg graf (igen vha R's graphics)



- Brug seq-funktionen til at lave en række af tal mellem 0 og 20 med 12 punkter
- Kombiner med en matematisk funktion så I får flg graf:



- Lav en funktion med flg signatur: `give_graf <- function(slope,inter)` som kan lave denne graf:



- Brug rep-funktionen til at lave en række af tal mellem 1 og 6 hvor hvert tal gentages 3 gange
- Samme opgave men hvor sekvensen gentages 3 gange
- Brug rep-funktionen til at lave en række med 24 bogstaver ud fra A,B,C,D hvor hvert bogstav gentages
- Lav en vektor af byer i DK og en funktion som kan returnere en tilfældig valgt by
- Modifier roll side 32 så prob sættes lig en vektor pv som beregnes af en funktion med følgende
signatur: `comp_ss <- function(s) {}` hvor s er det tal den sidste skal veje tungere således at `comp_ss(2)` skal give vektoren fra bogen.

Kap 3 til 5

- Hent støj og fugt fra <https://www.opendata.dk/city-of-aarhus/sensordata>
- Verficier grafisk hvad der er fugt og hvad der er støj
- Skriv jeres egen normaliserings-funktion
- Plot relevante udsnit af begge observationer i samme graf (DSTApi)

Øvelser:

Lav en matrix med tre kolonner. Den første med en række observationer der begynder ved 80, ender med 180 og springer med 10 mellem hver observation. De to andre skal fyldes med NA (Rintro)

Fyld den næste kolonne med værdier som udregnes vha Dolbears formel

$$T_F = 50 + \left(\frac{N_{60} - 40}{4} \right).$$

Kap 3 til 5

øvelse:

- indlæs filen test2.csv og ret den til så y ikke er en karakter men numerisk
- Hvor mange na'er?

(Rintro)

```
8 # lister
9 v1=1:10
10 v2=1:20
11 v3=letters[3:9]
12 mylist <- list("tal"=v1,"udgift"=v2,"prefix"=v3)
```

Øvelse:

1. tilføj to tal til v1
2. lav en liste og føj den til mylist
3. indlæs filen "data/evt_5360191_1_020223_1521.json" og beskriv datastrukturen

Øvelse:

1. Lav en dataframe på 5 observationer som simulerer alder og kropsvægt
2. Udvid så den består af 50 ved hjælp af runif.
3. Lav et plot af vægt-data for at se om den er normal-fordelt
4. Find et værktøj i R som kan lave normal-fordelte data
5. Lav et histogram og find et værktøj som kan afgøre om det er normal fordelt

(Rintro)

Øvelse:

1. Hente indholdet fra bit.ly/3OAtdQI og kopiér det lille R-script ind i en fil i Rstudio
1. Hvad er problemet med wq.csv?
2. Beskriv datasættet

Kap 3 til 5

Øvelse:

1. Hente indholdet fra bit.ly/3OAtdQI og kopiér det lille R-script ind i en fil i Rstudio
1. Hvad er problemet med wq.csv?
2. Beskriv datasættet

Øvelser

1. indlæs children.csv fra <https://github.com/cphstud/RIntroData>
2. Hvor gammelt er element 340?
3. Lav en dataframe med rækkerne 200 til 300
4. Lav en alternerede boolean vektor på 100 elementer (check rep-funktionen)
5. Brug denne vektor til at lave en delmængde af de 100 udvalgte

Øvelse:

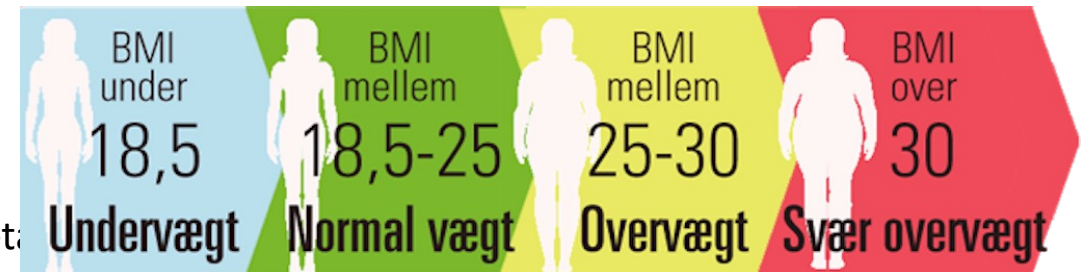
- Vis hvordan man kan få schild vha sample
- Brug == og sum til at tælle antal 15-årige
- Lav en dataframe med preteens fra children
- Lav en df med dem over 15 og under 10
- Lav en df med piger over 15

Kap 7

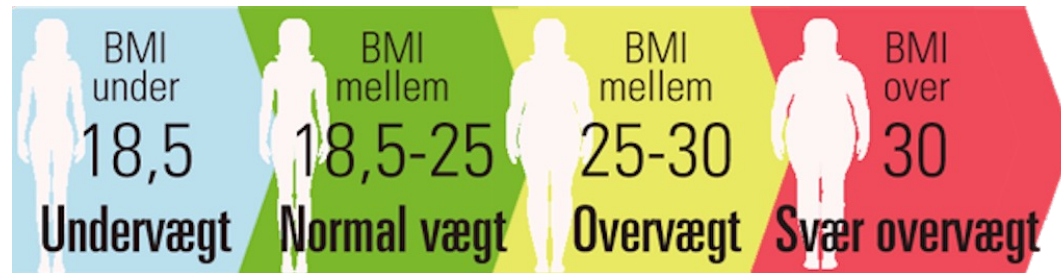
Øvelse:

```
res = dfchildren$height > 170
if (res) {
  print("OK")
} else {
  print("BAD")
}
```

- Få den til at printe "OK"
- tjek datasættet i winequality-red-2.csv for kvalitetskategorier
- tjek mtcars for antal cylindre
- lav en named vector til pokerjetoner
- åben lifeexpatbirth.csv og skriv en funktion som vælger et tilfældigt land og returnerer om det er nordisk.
- åben filen estonia-passenger-list.csv og lav en vektor med lande
- åben hwg.csv. Lav en tom kolonne til bmi. Loop igennem vha index-tilgængeligheden og indsæt bmi ud fra formlen vægt / højde^2 og figuren til højre. Start med den mindste
- Sys.time() give tiden nu. Sys.sleep(t) stopper programmet i t sekunder. Lav et while-loop som som kører i 30 sekunder og som printer "Hej Kurt" 6 gange.



åben hwg.csv. Lav en tom kolonne til bmi. Loop igennem vha index-tæller og indsæt bmi ud fra formelen vægt / højde^2 og figuren til højre. Start med den mindste



ggplot i R for Data Science

åben children.csv.

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

OBESITY

Attributes related with eating habits are:

Frequent consumption of high caloric food (FAVC),

Frequency of consumption of vegetables (FCVC),

Number of main meals (NCP),

Consumption of food between meals (CAEC),

Consumption of water daily (CH20),

and Consumption of alcohol (CALC).

Attributes related with the physical condition are:

Calories consumption monitoring (SCC),

Physical activity frequency (FAF),

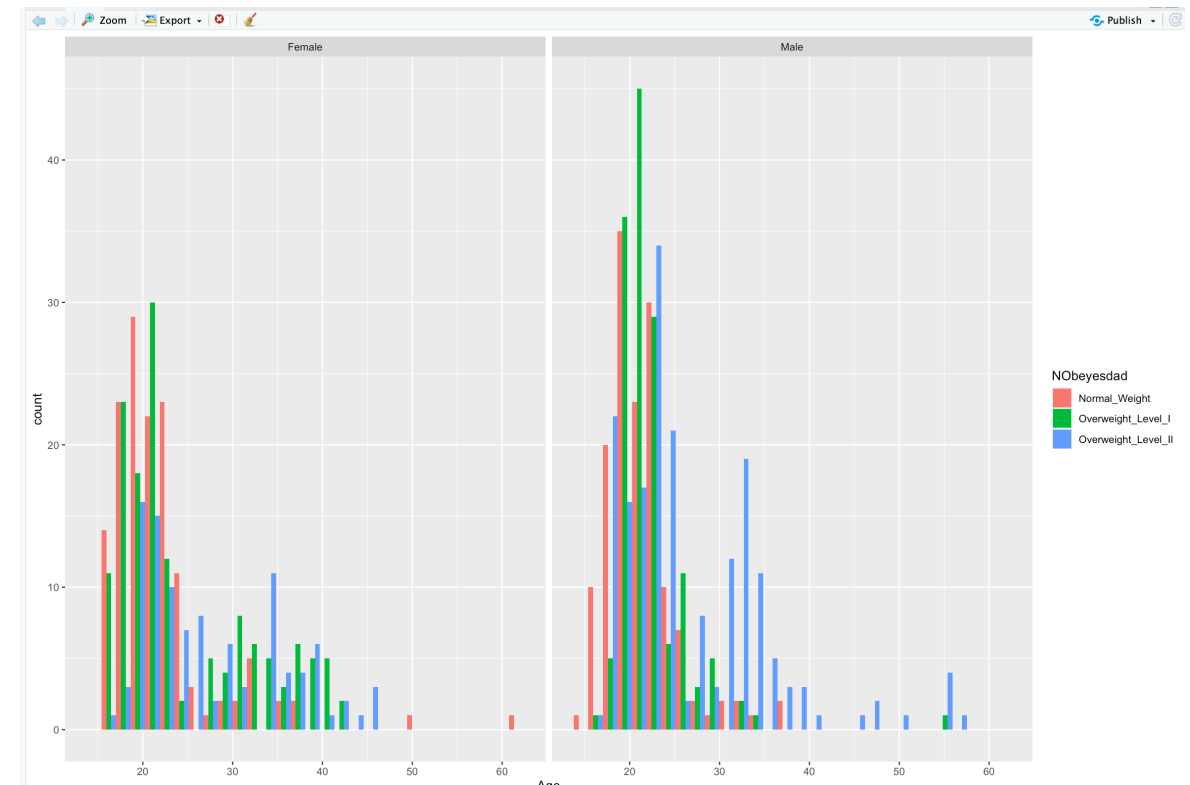
Time using technology devices (TUE),

Transportation used (MTRANS),

Other variables were:

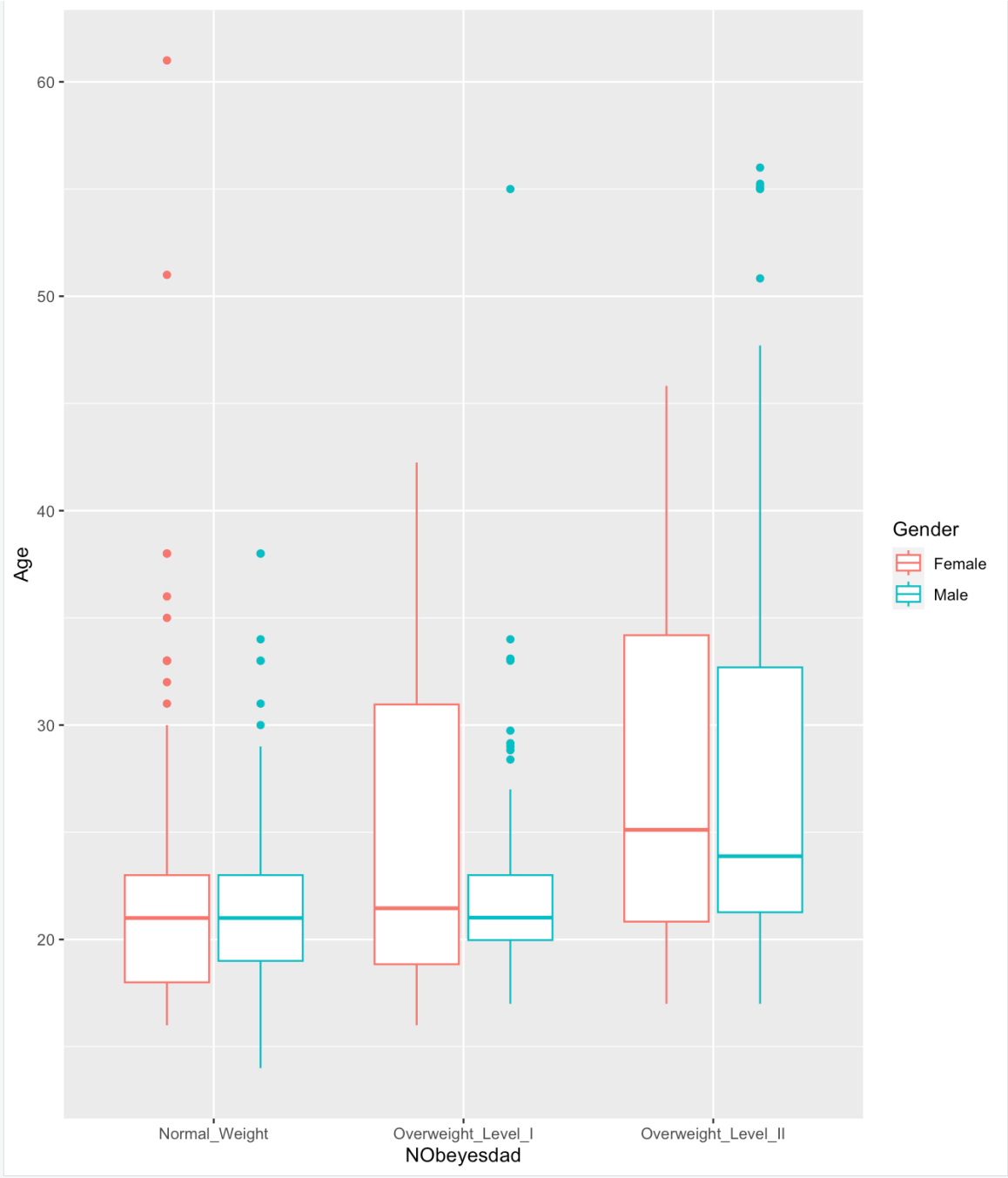
Gender, Age, Height and Weight.

- Hvordan er fordelingen på transportform?
- Hvordan er vægten fordelt i forhold til transportform?
- Lav et subset med normal vægt og overvægt I og II
- Lav en graf som flg:



OBESITY

Lav et boxplot på obesity som flg:

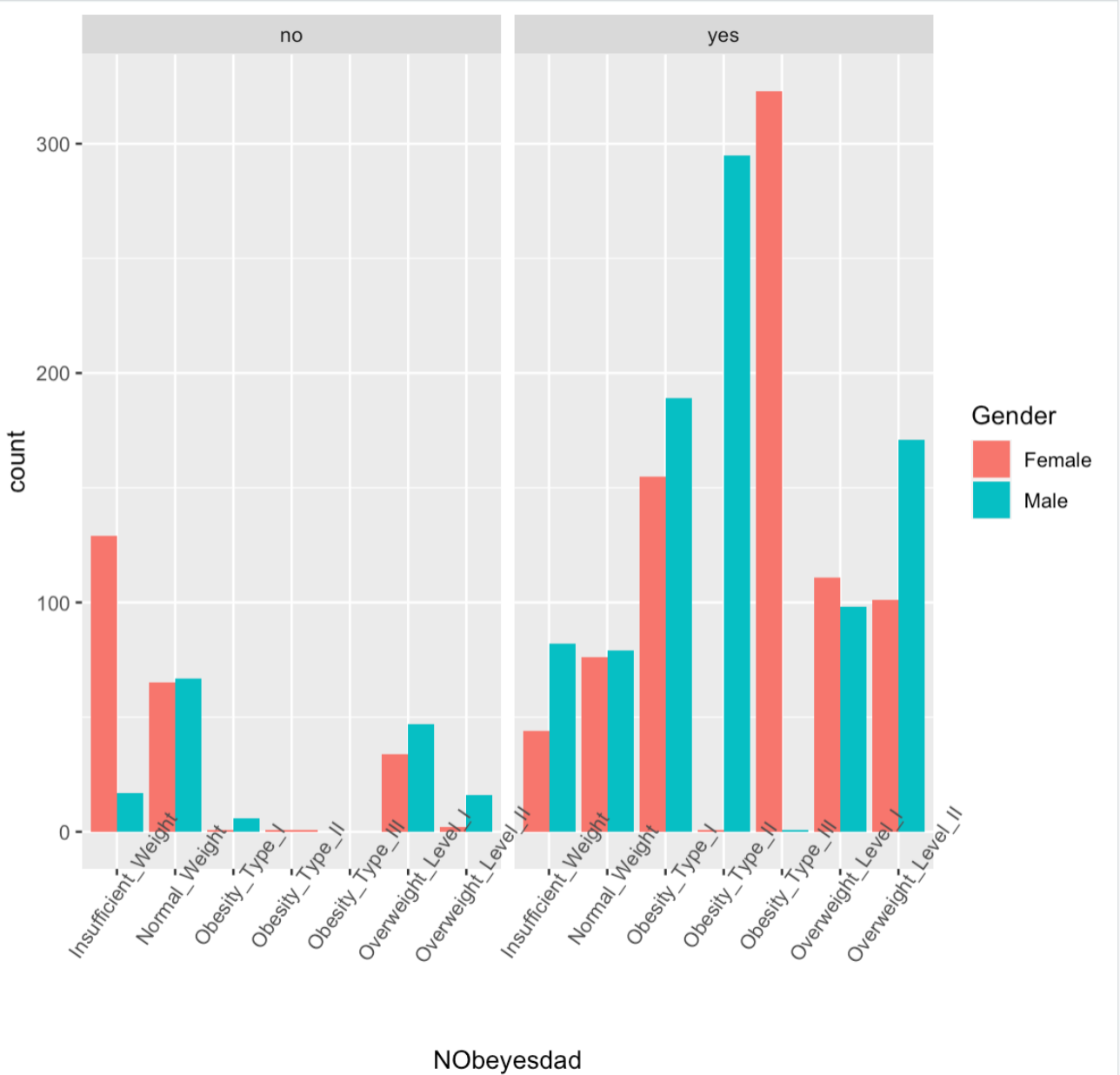


OBESITY

Lav dette subset på obesity:

| Gender | Famhist | NObeyesdad | count |
|--------|---------|---------------------|-------|
| Female | no | Insufficient_Weight | 129 |
| Male | no | Insufficient_Weight | 17 |
| Female | yes | Insufficient_Weight | 44 |
| Male | yes | Insufficient_Weight | 82 |
| Female | no | Normal_Weight | 65 |
| Male | no | Normal_Weight | 67 |
| Female | yes | Normal_Weight | 76 |
| Male | yes | Normal_Weight | 79 |
| Female | no | Obesity_Type_I | 1 |
| Male | no | Obesity_Type_I | 6 |
| Female | yes | Obesity_Type_I | 155 |
| Male | yes | Obesity_Type_I | 189 |

Tjek den første linje vha manuel subsetting
Lav denne graf



Boligsiden

- Gå på DST og find en tabel som kan give en liste over byer med indbyggertal
- Lav en kategorivariabel i R ud fra følgende kriterier:
`bycat=(c("landsby"=250,"lille by"=1000, "almindelig by"=2500, "større by"=10000, "storby"=50000))`
- Indlæs filen med boliger og tilpas de to dataframes så du kan udføre flg:
`total <- merge(boliger,byer,by="by", all.x = T)` således at du får kategorien `bycat` med i dit bolig-datasæt.
- Producer en dataframe og et plot som vist nedenfor

| | | | | |
|----|-------------|----------------|--------|---------------|
| 14 | åbenrå | 2.145.000 kr. | 12.883 | større by |
| 15 | åbenrå | 2.495.000 kr. | 12.600 | større by |
| 16 | åbybro | 1.595.000 kr. | 8.886 | almindelig by |
| 17 | åbybro | 23.500.000 kr. | 14.424 | almindelig by |
| 24 | agerskov | 3.995.000 kr. | 5.022 | lille by |
| 25 | åkirkeby | 1.395.000 kr. | 12.935 | lille by |
| 26 | åkirkeby | 1.095.000 kr. | 9.144 | lille by |
| 27 | åkirkeby | 1.495.000 kr. | 6.689 | lille by |
| 28 | ålbæk | 895.000 kr. | 8.544 | lille by |
| 29 | albertslund | 3.695.000 kr. | 28.314 | større by |

