# Homework 1

## Machine Learning in Economics

Prof. Dr. Hüseyin Taştan

Due: next week April 8th, 2021

NAME & SURNAME:BURAK UZUN NO:18035028

**Arrange your answers in an `.Rmd` file (you can take `HW01_MLinEcon.Rmd` as a template) and produce a html file containing your answers. You need to upload your files to the Google Classroom by the deadline.**

---

# Part 1

## Question

In this exercise you will simulate a data set and run a simple regression. To ensure reproducible results, make sure you use `set.seed(1)`. This exercise is similar to the one we saw in class (see regression lab).

a. Using the `rnorm()` function create vector $X$ that contains 200 observations from `N(0,1)` distribution.

b. Similarly, create a 200 element vector, `epsilon` ($\epsilon$), drawn from `N(0,0.25)` distribution. This is the irreducible error.

c. Create the response data using the following relationship:

$$Y = -1 + 0.5X + \epsilon$$

d. Now using your simulated data set, fit a linear regression of Y on X using `lm()` function. Display the summary statistics and interpret the diagnostic plots. In a single graph, draw a scatter diagram of Y and X values, and the fitted line.

e. Now, fit a quadratic model by adding the $X^2$ into the model. Discuss whether there is improvement in the fit or not. Draw scatter diagram and fitted line similar to the previous part. Interpret the diagnostic plots.

f. Using the `sample()` function create train and test sets just like we did in class. Obtain predicted values (from the test set) for the linear and quadratic fits. Compare their MSEs. Which model is better in terms of predictions?

## Answer

Put your answers, explanations, interpretations here.

You can fill in the following code chunk.

```
library(ggplot2)
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.0.5
```

```
library(tidyr)
library(tidyverse)
```

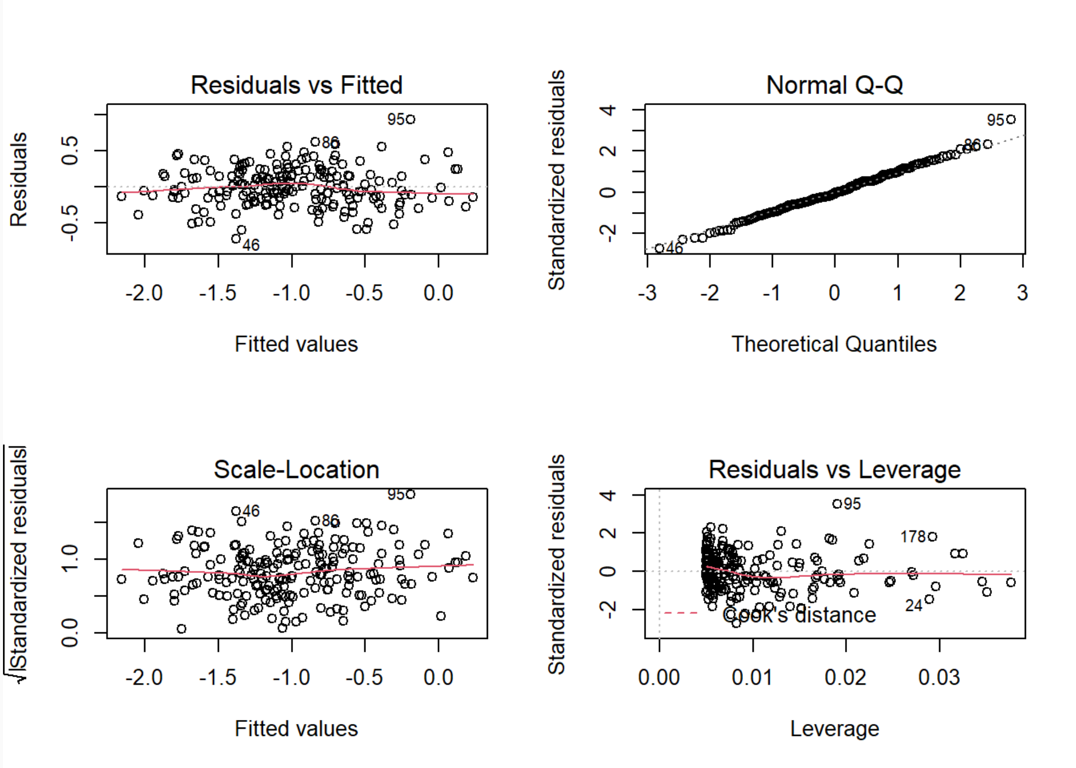
```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.0     v dplyr   1.0.5
## v readr   1.4.0     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1
```

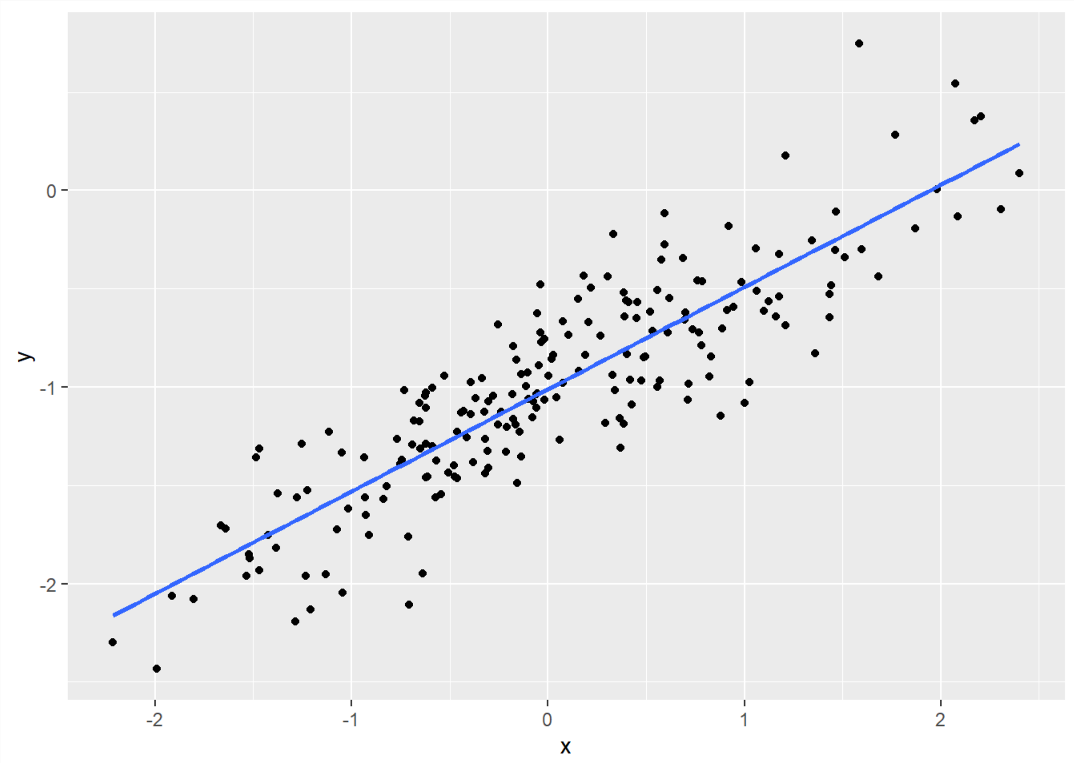
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
set.seed(1)
n       <- 200
set.seed(1)
x <- rnorm(n, mean = 0, sd = 1)
epsilon <- rnorm(n, mean = 0, sd = 0.25)
y = -1 + 0.5*x + rnorm(n, mean = 0, sd = 0.25)
reg1 <- lm(formula = y ~ x)
tidy(reg1)
par(mfrow=c(2,2))
plot(reg1)
```



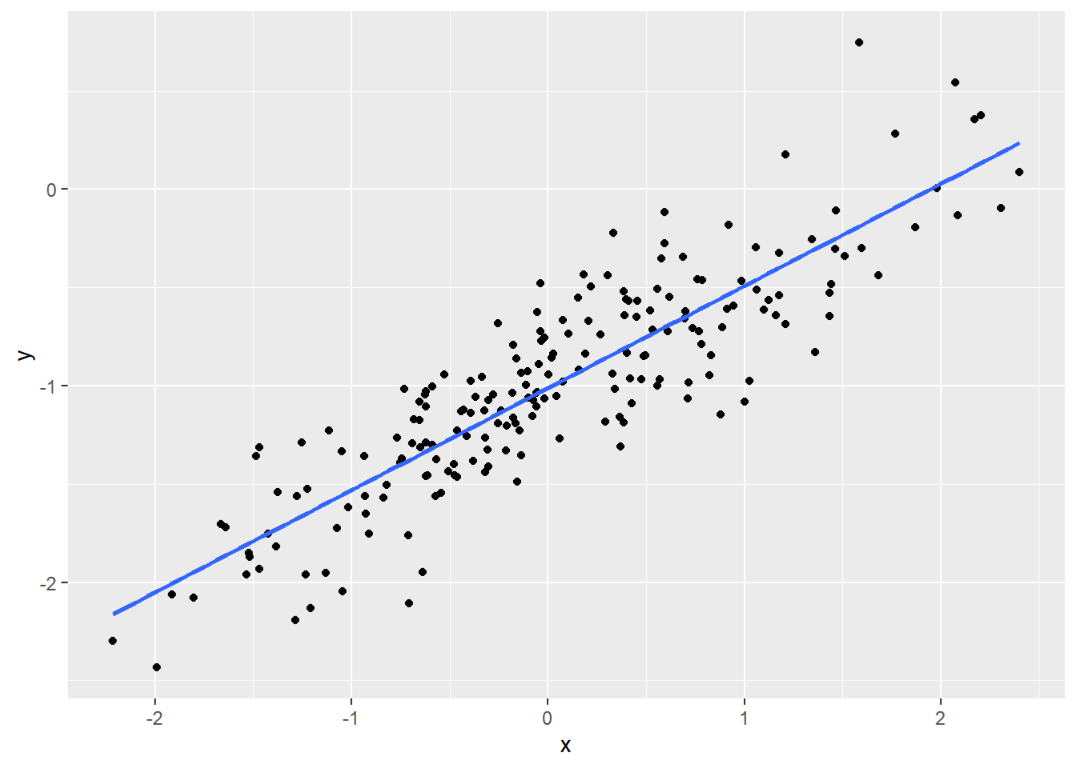

```
ggplot(reg1, aes(x = x, y = y)) + geom_point() + geom_smooth(method = "lm", se = FALS
```

```
## `geom_smooth()` using formula 'y ~ x'
```

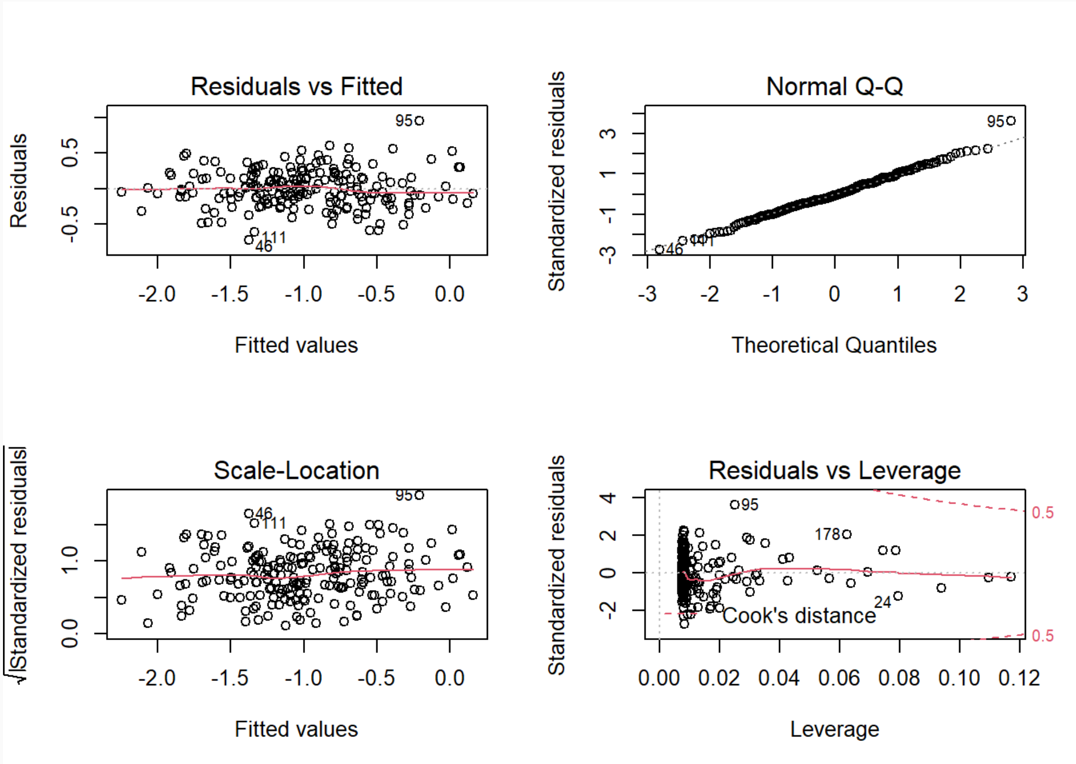



```
reg2 <- lm(formula = y~x + I(x^2))
anova(reg1, reg2)
ggplot(reg2, aes(x = x, y = y)) +geom_point() + geom_smooth(method = "lm", se = FALSE
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
par(mfrow=c(2,2))
plot(reg2)
```



```
set.seed(1)
df1 <- tibble(id=1:n, y, x)
train <- sample(n, 100)
train_data <- df1[train, ]
test_data <- df1[-train, ]
lfit <- lm(y ~ x, data = train_data)
qfit <- lm(y ~ x, I(x^2), data = train_data)
lin_predict <- predict(lfit, test_data)
lin_error <- test_data$y - lin_predict
MSE_lfit <- mean(lin_error^2)
MSE_lfit
qupredict <- predict(qfit, test_data)
querror <- test_data$y - qupredict
MSE_qfit <- mean(querror^2)
MSE_qfit
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   -1.01     0.0189     -53.4 1.54e-119
## 2 x              0.520    0.0204      25.5 2.37e- 64
## Analysis of Variance Table
##
## Model 1: y ~ x
## Model 2: y ~ x + I(x^2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    198 14.174
## 2    197 14.087  1  0.087729 1.2269 0.2694
## [1] 0.08039026
## [1] 0.08635954
```

e.Quadratic model is not better according to F test from anova function.

f. It looks like test MSE = 0.0598797 from the linear model and MSE = 0.059424 for the quadratic model. The difference is very small, 0.0004560702. It looks like we can use both models in our predictions.

# Part 2

## Question

Boston house data set (part of `MASS` package) that we saw in class has a variable that measures per capita crime rate by town (`crim`). Now suppose that the `crim` is the response variable and all the remaining variables are features. Fit a multiple linear regression model and interpret the results.

## Answer

```
# put your code here
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
regboston <- lm(formula = crim ~ . , data = Boston)
summary(regboston)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
names(regboston)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

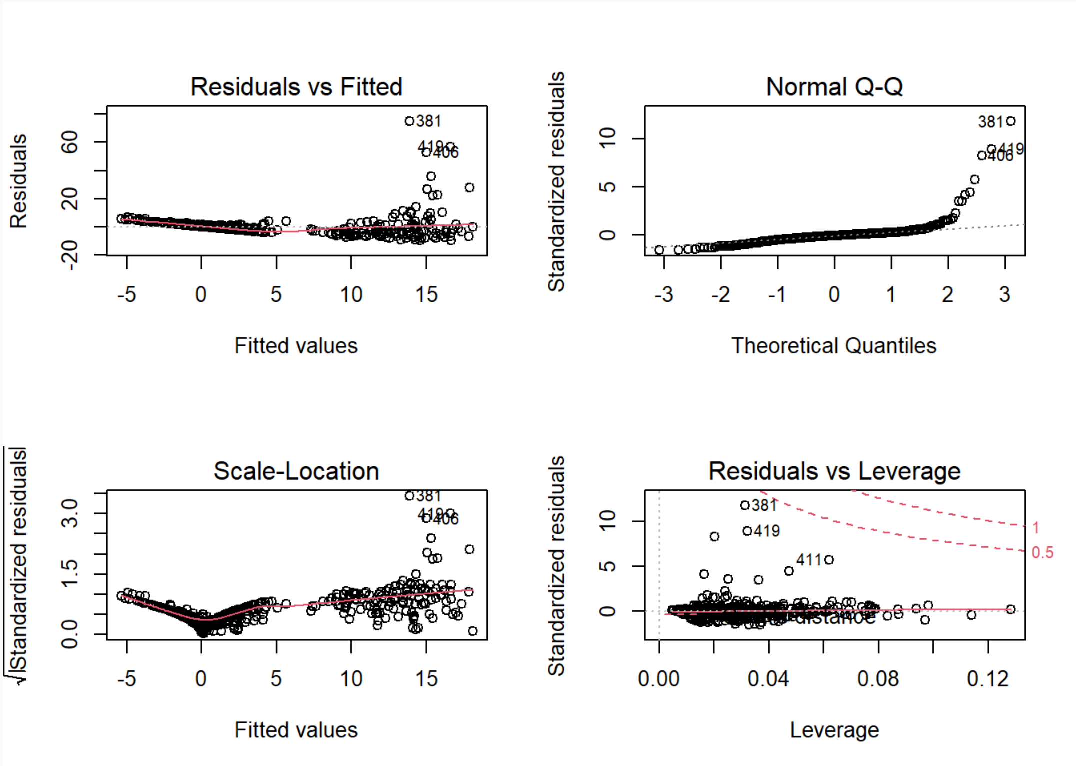
```
coef(regboston)
```

```
##   (Intercept)            zn         indus          chas           nox
##  17.033227523   0.044855215  -0.063854824  -0.749133611 -10.313534912
##            rm           age           dis           rad           tax
##   0.430130506   0.001451643  -0.987175726   0.588208591  -0.003780016
##       ptratio         black         lstat          medv
##  -0.271080558  -0.007537505   0.126211376  -0.198886821
```

```
confint(regboston)
```

```
##                      2.5 %        97.5 %
## (Intercept)   2.818109179 31.2483458660
## zn            0.008046562  0.0816638671
## indus        -0.227733150  0.1000235023
## chas         -3.067882868  1.5696156471
## nox         -20.678894713  0.0518248891
## rm           -0.773956866  1.6342178774
## age          -0.033767600  0.0366708869
## dis          -1.540889544 -0.4334619069
## rad           0.415209611  0.7612075719
## tax          -0.013909700  0.0063496670
## ptratio      -0.637417996  0.0952568794
## black        -0.014754837 -0.0003201725
## lstat        -0.022572584  0.2749953365
## medv         -0.317788478 -0.0799851646
```

```
par(mfrow=c(2,2))
plot(regboston)
```




# Part 3

## Question

Prepare a simple table that shows the most popular women's names in Turkey in 2018 (see the example we discussed in class). Prepare a similar table for the year 2017 and compare the two tables. Are there any differences?

## Answer

```
# put your code here
# the data set is already available in the HW project folder
# you can read it into R using:
library(tidyverse)
library(readxl)
womennames2018 <- read_excel("womennames2018.xls", range = "A4:AQ338")
```

```
## New names:
## * `` -> ...1
```

```
head(womennames2018)
```

```
## # A tibble: 6 x 43
##   ...1   `1950` `1960` `1970` `1980` `1981` `1982` `1983` `1984` `1985` `1986`
##   <chr>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Ada        NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 2 Ahsen      NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 3 Aleyna     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 4 Aliye      98     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 5 Alya       NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 6 Amine      NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## # ... with 32 more variables: 1987 <dbl>, 1988 <dbl>, 1989 <dbl>, 1990 <dbl>,
## #   1991 <dbl>, 1992 <dbl>, 1993 <dbl>, 1994 <dbl>, 1995 <dbl>, 1996 <dbl>,
## #   1997 <dbl>, 1998 <dbl>, 1999 <dbl>, 2000 <dbl>, 2001 <dbl>, 2002 <dbl>,
## #   2003 <dbl>, 2004 <dbl>, 2005 <dbl>, 2006 <dbl>, 2007 <dbl>, 2008 <dbl>,
## #   2009 <dbl>, 2010 <dbl>, 2011 <dbl>, 2012 <dbl>, 2013 <dbl>, 2014 <dbl>,
## #   2015 <dbl>, 2016 <dbl>, 2017 <dbl>, 2018 <dbl>
```

```
womennames2018 <- rename(womennames2018, "name"="...1")
head(womennames2018)
```

```
## # A tibble: 6 x 43
##   name   `1950` `1960` `1970` `1980` `1981` `1982` `1983` `1984` `1985` `1986`
##   <chr>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Ada        NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 2 Ahsen      NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 3 Aleyna     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 4 Aliye      98     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 5 Alya       NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 6 Amine      NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## # ... with 32 more variables: 1987 <dbl>, 1988 <dbl>, 1989 <dbl>, 1990 <dbl>,
## #   1991 <dbl>, 1992 <dbl>, 1993 <dbl>, 1994 <dbl>, 1995 <dbl>, 1996 <dbl>,
## #   1997 <dbl>, 1998 <dbl>, 1999 <dbl>, 2000 <dbl>, 2001 <dbl>, 2002 <dbl>,
## #   2003 <dbl>, 2004 <dbl>, 2005 <dbl>, 2006 <dbl>, 2007 <dbl>, 2008 <dbl>,
## #   2009 <dbl>, 2010 <dbl>, 2011 <dbl>, 2012 <dbl>, 2013 <dbl>, 2014 <dbl>,
## #   2015 <dbl>, 2016 <dbl>, 2017 <dbl>, 2018 <dbl>
```

```
womennames <- womennames2018 %>%
 pivot_longer(-name, names_to="year", values_to = "rank")
womennames %>%
  arrange(year, name) %>%
  head(10)
```

```
## # A tibble: 10 x 3
##    name   year   rank
##    <chr>  <chr> <dbl>
##  1 Ada    1950     NA
##  2 Ahsen  1950     NA
##  3 Aleyna 1950     NA
##  4 Aliye  1950     98
##  5 Alya   1950     NA
##  6 Amine  1950     NA
##  7 Arife  1950     80
##  8 Arin   1950     NA
##  9 Arya   1950     NA
## 10 Arzu   1950     NA
```

```
top10_2018 <- womennames %>%
  filter(year==2018, rank<11) %>%
  arrange(rank)
top10_2018
```

```
## # A tibble: 10 x 3
##    name    year   rank
##    <chr>   <chr> <dbl>
##  1 Zeynep  2018      1
##  2 Elif    2018      2
##  3 Defne   2018      3
##  4 Ebrar   2018      4
##  5 Eylül   2018      5
##  6 Hiranur 2018      6
##  7 Asel    2018      7
##  8 Meryem  2018      8
##  9 Zehra   2018      9
## 10 Miray   2018     10
```

```
top10_2017 <- womennames %>%
  filter(year==2017,rank<11) %>%
  arrange(rank)
top10_2017
```

```
## # A tibble: 10 x 3
##    name    year   rank
##    <chr>   <chr> <dbl>
##  1 Zeynep  2017      1
##  2 Elif    2017      2
##  3 Defne   2017      3
##  4 Hiranur 2017      4
##  5 Ebrar   2017      5
##  6 Eylül   2017      6
##  7 Miray   2017      7
##  8 Zehra   2017      8
##  9 Ecrin   2017      9
## 10 Azra    2017     10
```

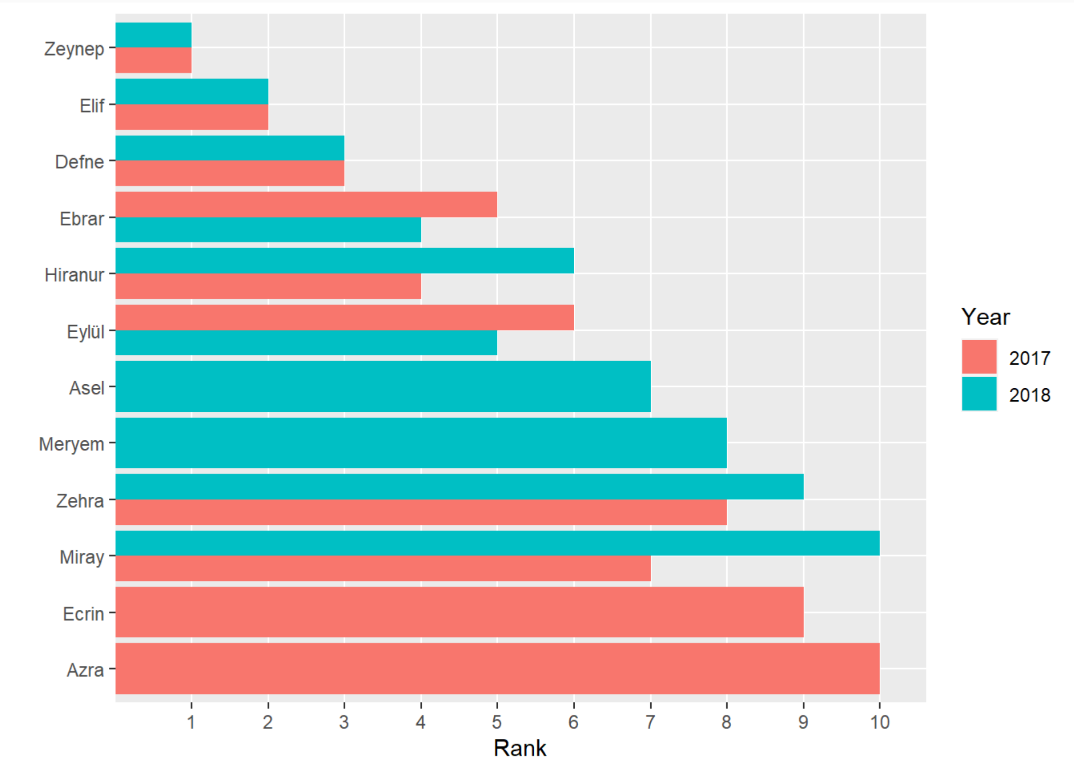
```
top10 <- full_join(top10_2017, top10_2018)
```

```
## Joining, by = c("name", "year", "rank")
```

```
top10
```

```
## # A tibble: 20 x 3
##    name    year  rank
##    <chr>   <chr> <dbl>
##  1 Zeynep  2017      1
##  2 Elif    2017      2
##  3 Defne   2017      3
##  4 Hiranur 2017      4
##  5 Ebrar   2017      5
##  6 Eylül   2017      6
##  7 Miray   2017      7
##  8 Zehra   2017      8
##  9 Ecrin   2017      9
## 10 Azra    2017     10
## 11 Zeynep  2018      1
## 12 Elif    2018      2
## 13 Defne   2018      3
## 14 Ebrar   2018      4
## 15 Eylül   2018      5
## 16 Hiranur 2018      6
## 17 Asel    2018      7
## 18 Meryem  2018      8
## 19 Zehra   2018      9
## 20 Miray   2018     10
```

```
top10 %>%
  mutate(Name = factor(name), Rank = factor(rank), Year=factor(year)) %>%
  ggplot(aes(x=reorder(name,-rank), Rank, fill=Year)) +
  geom_bar(position=position_dodge(), stat = "identity") +
  xlab("") +
  coord_flip()
```




See the example in Introduction to the R Tidyverse

There are a total of 12 names in top 10 in years 2017 and 2018. Zeynep, Elif, and Defne are in the top 3, respectively, in both years. Ecrin and Azra were in the top 10 in 2017 but not in 2018. We see that Asel and Meryem entered the top 10 in 2018 at 7th and 8th places, respectively. Miray and Zehra became less popular as they went from the 8th to 9th and 7th to 10th places, respectively. Ebrar and Eylül, on  e more popular.