HW10

108048110

4/20/2022

BACS HW - Week 9

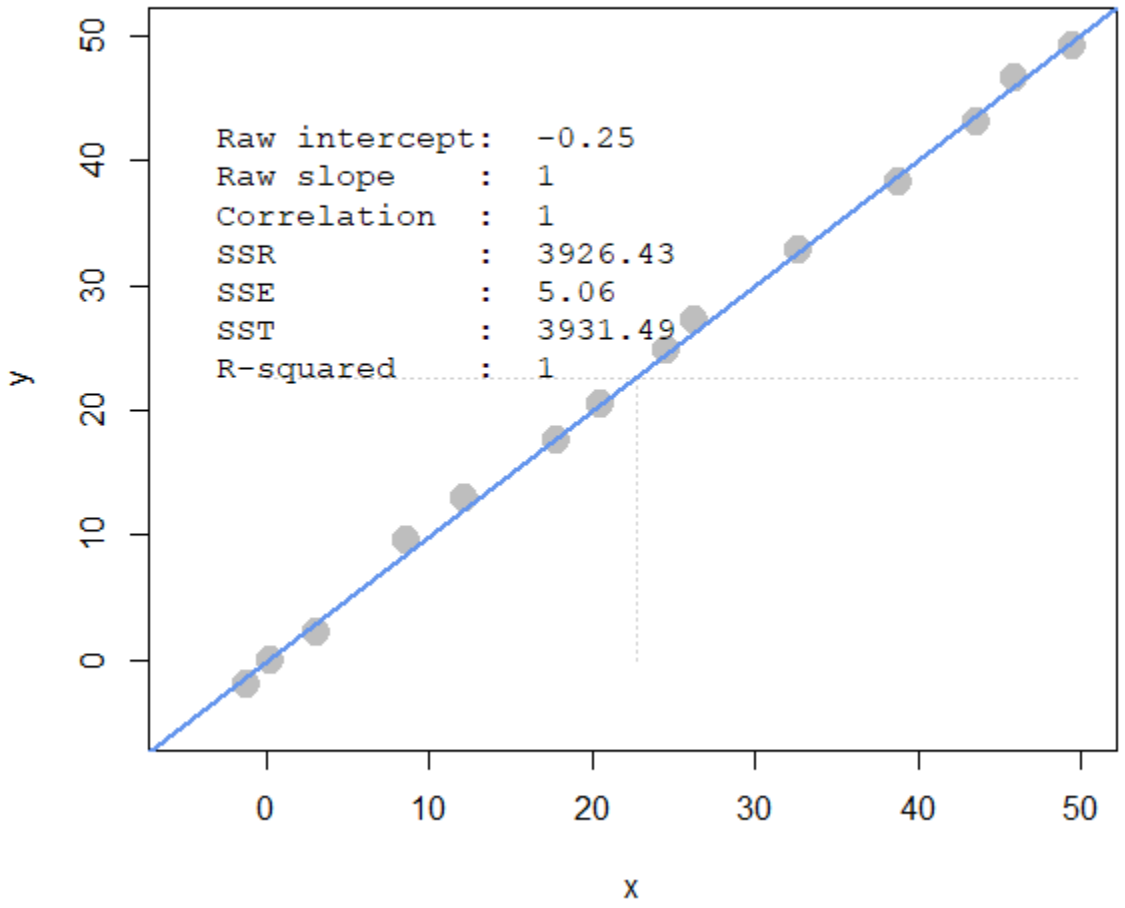
Prerequisite

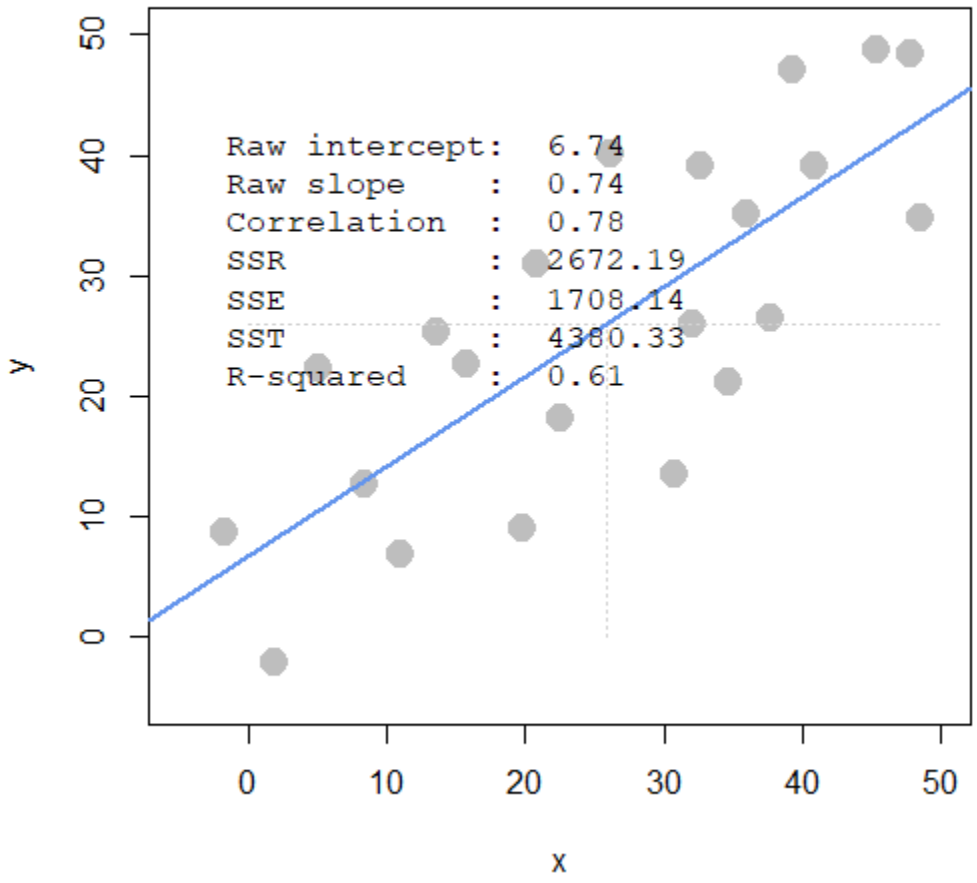
```
library(dplyr)
library(ggplot2)
library(reshape2)
library(ggcorrplot)
library(highcharter)
library(webshot)
```

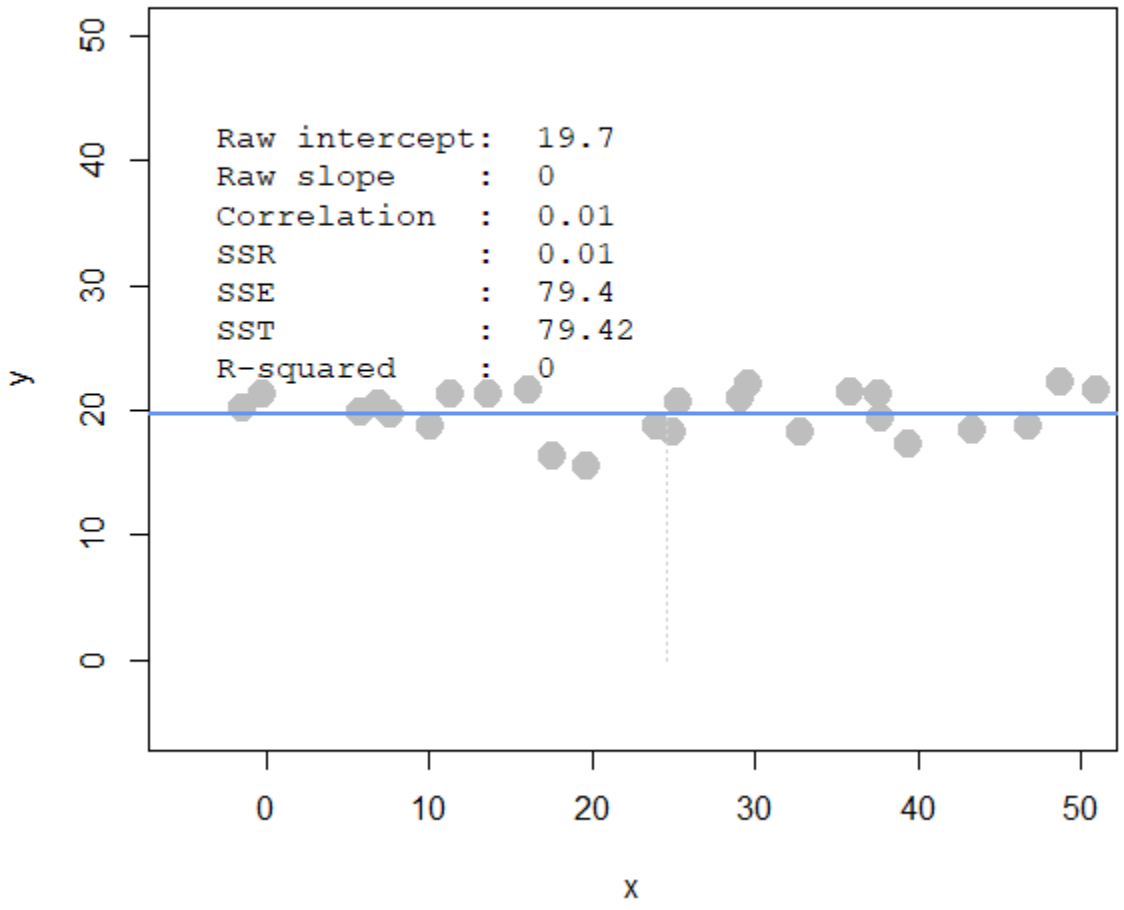
Question 1) Simulate each scenarios.

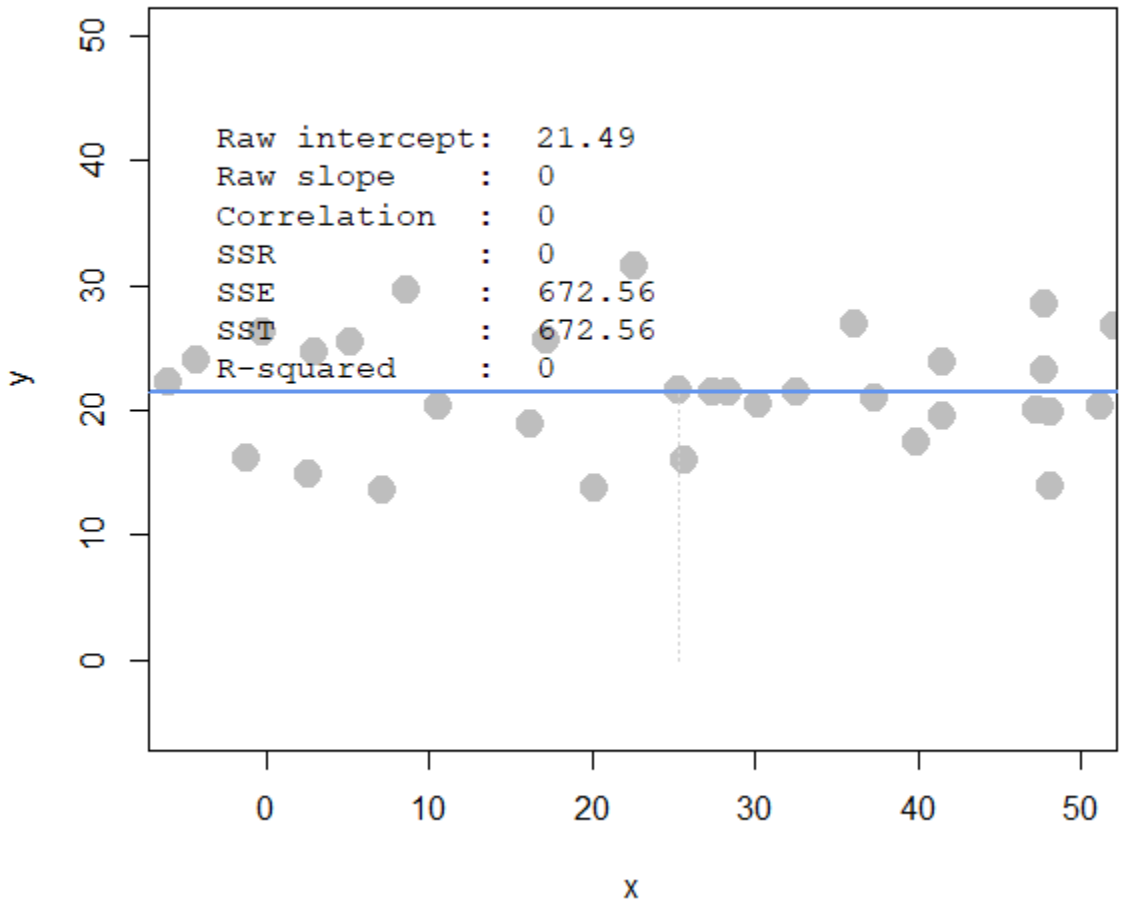
Table 1: Four Scenarios

Scenarios	Explaination
Scenario 1	***a very narrowly dispersed set of points (negative / positive steep slope)***
Scenario 2	***a widely dispersed set of points (negative / positive steep slope)***
Scenario 3	***a very narrowly dispersed set of points (negative / positive shallow slope)***
Scenario 4	***a widely dispersed set of points (negative or positive shallow slope)***


Raw intercept: -0.25
Raw slope : 1
Correlation : 1
SSR : 3926.43
SSE : 5.06
SST : 3931.49
R-squared : 1
y
x
0
10
20
30
40
50


Raw intercept: 6.74
Raw slope : 0.74
Correlation : 0.78
SSR : 2672.19
SSE : 1708.14
SST : 4380.33
R-squared : 0.61
y
x
0
10
20
30
40
50


Raw intercept: 19.7
Raw slope : 0
Correlation : 0.01
SSR : 0.01
SSE : 79.4
SST : 79.42
R-squared : 0
y
x
0
10
20
30
40
50



a. Comparing scenarios 1 and 2, which do we expect to have a stronger R2 ?

Ans. I expect the first scenario (1) to have a stronger R^2 value.

b. Comparing scenarios 3 and 4, which do we expect to have a stronger R2 ?

Ans. I expect the former scenario (3) to have a stronger R^2 value.

c. Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST?

Table 2: Variations

SSE	sum of squares' error	y, $\bar{y}$
SSR	sum of squares' regression	$\hat{y}$, $\bar{y}$
SST	sum of squares' total.	y, $\hat{y}$

Ans. I expect the first scenario to have a smaller SSE; while plot1 has a steeper slope than plot2, I would also assume that the former plot has a larger SSR value. Last but not least, points on plot1 falls almost exactly on the estimated regression line, as a result I would expect plot1 to have a smaller value of SST.

d. Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

Ans. As we can observe from the plots, both of their estimated regression line lie on the means. As we can observe from the above table, we can easily conclude that SSR values on both plots are expected to be 0. Therefore, in this case, the SSE value is equal to SST value, which is observed to be larger in plot4.

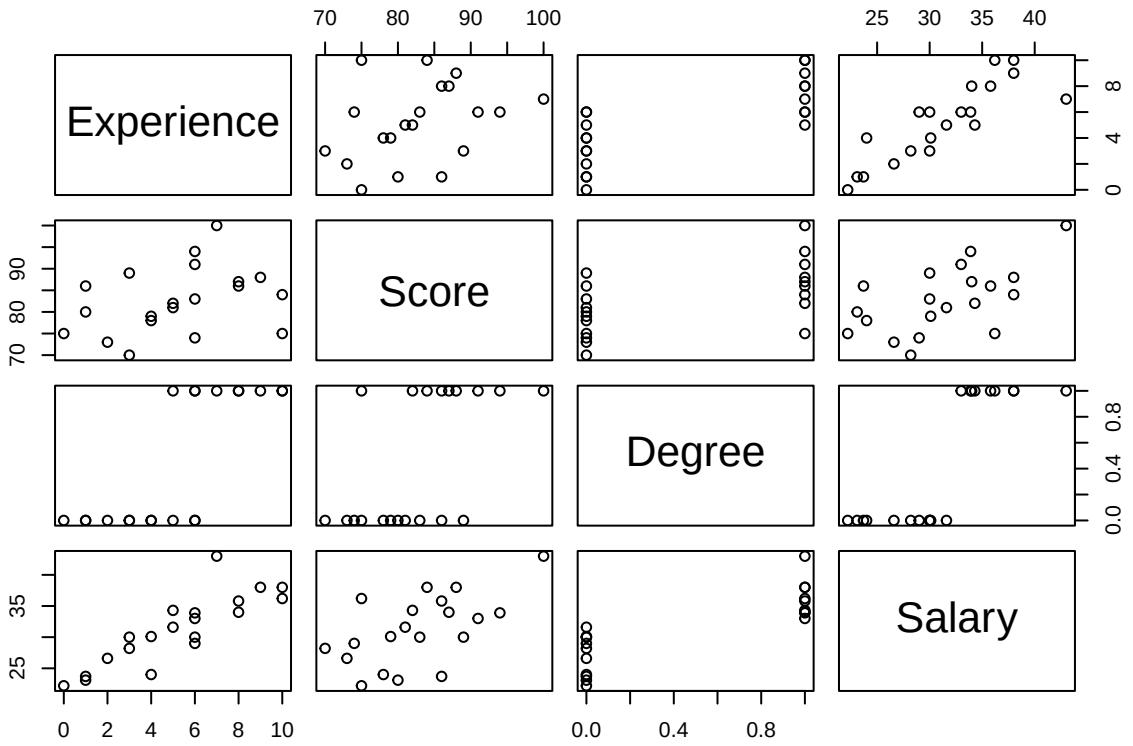
Question 2) Regression

Prerequisite

```
dataset = read.csv('programmer_salaries.txt', sep='\t')
knitr::kable(head(dataset))
```

Experience	Score	Degree	Salary
4	78	0	24.0
7	100	1	43.0
1	86	0	23.7
5	82	1	34.3
8	86	1	35.8
10	84	1	38.0

```
plot(dataset)
```



a. Estimate the model `Salary ~ Experience + Score + Degree`

- β
- R^2
- Top 5 values of $\hat{y}$ and ϵ

```
# Raw data regression
summary(lm(Salary~Experience+Score+Degree, data=dataset))
```

```
##
## Call:
## lm(formula = Salary ~ Experience + Score + Degree, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8963 -1.7290 -0.3375  1.9699  5.0480
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9448     7.3808   1.076   0.2977
## Experience    1.1476     0.2976   3.856   0.0014 **
## Score         0.1969     0.0899   2.191   0.0436 *
## Degree        2.2804     1.9866   1.148   0.2679
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.396 on 16 degrees of freedom
## Multiple R-squared:  0.8468,   Adjusted R-squared:  0.8181
## F-statistic: 29.48 on 3 and 16 DF,  p-value: 9.417e-07
```

- β_0: If the first variable has no value, how much of variable2 will it be? (intercept)
- β_1: How much of the covariance of v1 and v2 is explained by the variance of v1? (slope)
- In this case, $R^2 = 0.8468$

```
summary(lm(Salary~Experience+Score+Degree, data=data.frame(scale(dataset))))
```

```
##
## Call:
## lm(formula = Salary ~ Experience + Score + Degree, data = data.frame(scale(dataset)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69347 -0.30773 -0.06007  0.35061  0.89845
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.689e-17  9.538e-02   0.000   1.0000
## Experience  6.059e-01  1.571e-01   3.856   0.0014 **
## Score       2.669e-01  1.218e-01   2.191   0.0436 *
## Degree      2.072e-01  1.805e-01   1.148   0.2679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4265 on 16 degrees of freedom
## Multiple R-squared:  0.8468,   Adjusted R-squared:  0.8181
## F-statistic: 29.48 on 3 and 16 DF,  p-value: 9.417e-07
```

- After standardizing the data, R^2 remains the same.
- $R^2 = 0.8468$

```
lm(Salary~Experience+Score+Degree, data=dataset)$fitted.values[1:5]
```

```
##        1        2        3        4        5
## 27.89626 37.95204 26.02901 32.11201 36.34251
```

```
lm(Salary~Experience+Score+Degree, data=dataset)$residuals[1:5]
```

```
##          1          2          3          4          5
## -3.8962605  5.0479568 -2.3290112  2.1879860 -0.5425072
```

Note. R is the coefficient of multiple correlation between dependent variable and all independent variables, and R^2 is the coefficient of determination.

b. Use `linear algebra` and the `geometric view of regression` to estimate the regression.

- ***i.*** Create an $\mathbf{X}$ matrix that has a first column of `1s` followed by columns of the independent variables.

```
ones <- rep(1, length(dataset$Experience))
X <- matrix(c(ones, dataset$Experience, dataset$Score, dataset$Degree),
            ncol=4,
            nrow=length(ones),
            byrow=FALSE)

colnames(X) <- c("1","exp","score","degree")
dim(X)
```

```
## [1] 20  4
```

```
knitr::kable(head(X))
```

1	exp	score	degree
1	4	78	0
1	7	100	1
1	1	86	0
1	5	82	1
1	8	86	1
1	10	84	1

- ***ii.*** Create a `y` vector with the Salary values.

```
y <- as.matrix(dataset$Salary)
dim(y)
```

```
## [1] 20  1
```

- ***iii.*** Compute the $\hat{\beta}$ vector of estimated regression coefficients.

```
beta_hat <- solve(t(X)%*%X)%*%t(X)%*%y
dim(beta_hat)
```

```
## [1] 4 1
```

```
print(beta_hat)
```

```
##             [,1]
## 1      7.944849
## exp    1.147582
## score  0.196937
## degree 2.280424
```

- ***iv.*** Compute a $\hat{y}$ vector of estimated y values, and a ϵ vector of residuals

```r
y_hat <- X %*% beta_hat
dim(y_hat)
```

```
## [1] 20  1
```

```r
print(y_hat[1:5,])
```

```
## [1] 27.89626 37.95204 26.02901 32.11201 36.34251
```

```r
residual <- y - y_hat
print(residual[1:5,])
```

```
## [1] -3.8962605  5.0479568 -2.3290112  2.1879860 -0.5425072
```

- ***v.*** Using only the results from ***(i)*** – ***(iv)***, compute `SSR`, `SSE` and `SST`.

```r
variations <- function(y, y_hat){
  y_mean = mean(y)

  stat <- list()
  stat$R_squared <- cor(y, y_hat)^2
  stat$SSR <- sum((y_hat-y_mean)^2)
  stat$SSE <- sum((y-y_hat)^2)
  stat$SST <- sum((y-y_mean)^2)

  return(stat)
}

model_fit <- variations(y, y_hat)
round(model_fit$SSR+model_fit$SSE, 2) == round(model_fit$SST, 2)
```

```
## [1] TRUE
```

```r
print(model_fit)
```

```
## $R_squared
##           [,1]
## [1,] 0.8467961
##
## $SSR
## [1] 507.896
##
## $SSE
## [1] 91.88949
##
## $SST
## [1] 599.7855
```

c. Compute R^2 for in two ways, and confirm you get the same results.

- ***i.***

```
R_squared_i <- 1-model_fit$SSE/model_fit$SST
```

- ***ii.***

```
R_squared_ii <- model_fit$R_squared
```

```
# Comparison
round(R_squared_i, 2) == round(R_squared_ii, 2)
```

```
##      [,1]
## [1,] TRUE
```

```
print(c(R_squared_i, R_squared_ii))
```

```
## [1] 0.8467961 0.8467961
```

Question 3) Early heady days of global car manufacturing, we are interested in explaining what kind of cars have higher fuel efficiency (mpg).

Table 5: auto-data.txt

1	mpg	miles-per-gallon
2	cylinders	cylinders in engine
3	displacement	size of engine
4	horsepower	power of engine
5	weight	weight of car
6	acceleration	acceleration ability of car
7	model_year	year model was released
8	origin	place car was designed (1: USA, 2: EU, 3. Japan)
9	car_name	make and model names

Prerequisite

```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
auto = auto[complete.cases(auto),]
```

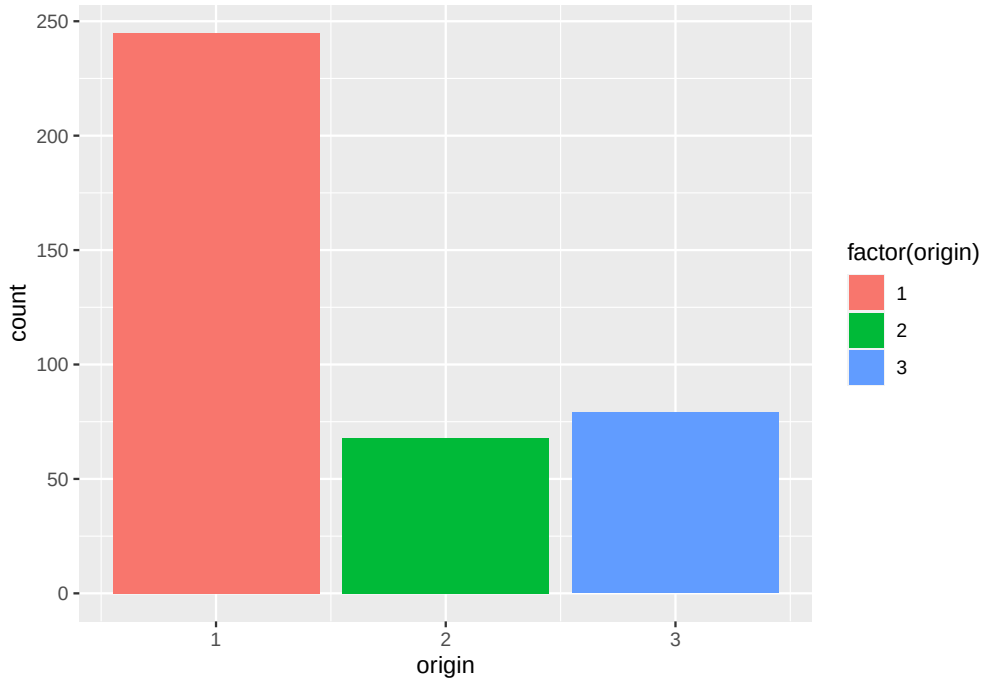
a. Exploring the data.

- ***i.*** Visualize the data.
  I create a `type` variable using `car_names`.

```
type=""
for (name in auto$car_name){
  type = c(type, strsplit(name, split = " ")[[1]][1])
}
auto$type = type[-1]

# Trying to plot linear relationship between variabes
plt <- function(a, b){
ggplot(auto, aes(x=a, y=b))+
  geom_point()+
  stat_smooth(method='lm')+
    labs(x='', y='')
}
```

- – Origin counts

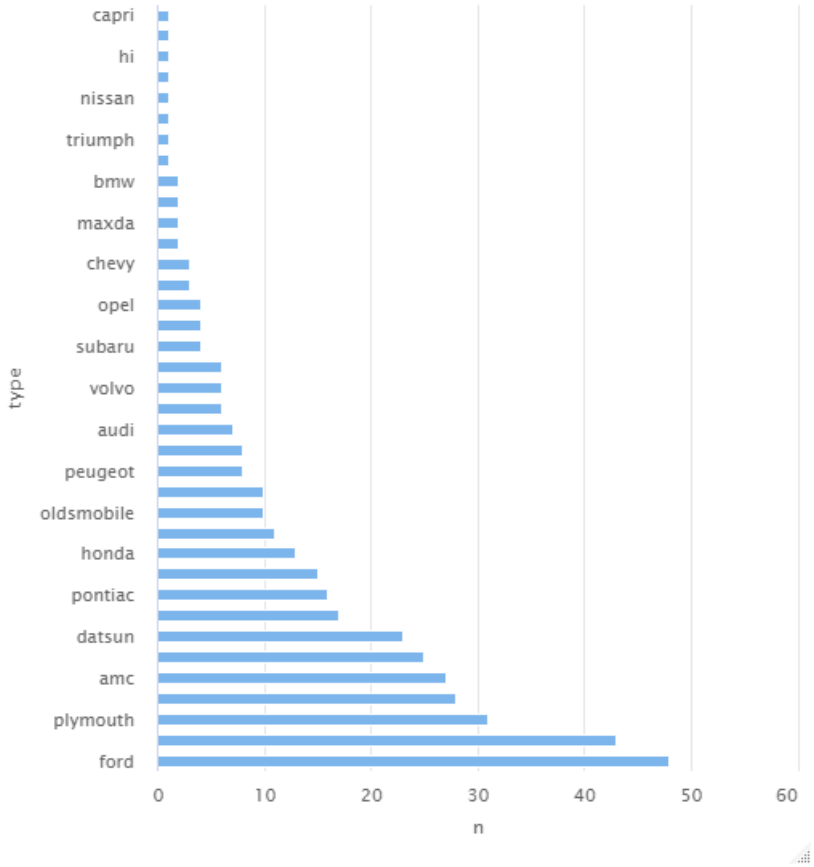
```
ggplot(data=auto) +
  geom_bar(mapping=aes(x=origin,fill=factor(origin)))
```



Most of the cars are made from the United States.

– Type counts

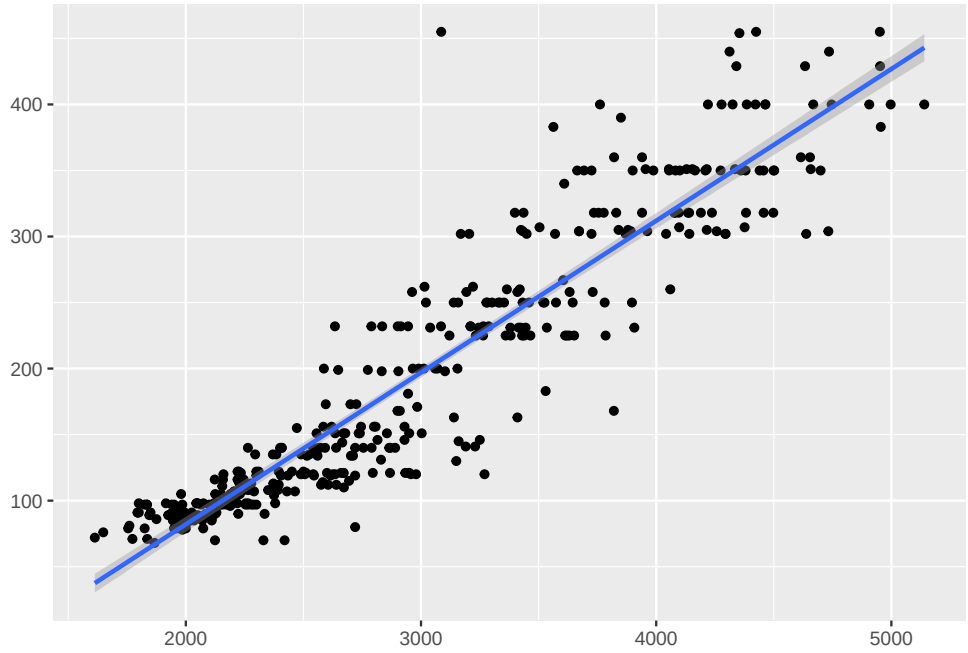
```
#auto %>%
  #count(type) %>%
  #arrange(n) %>%
  #hchart(type="bar", hcaes(x=type, y=n))
```



Most of the vehicles are produced by "Ford."

– Weight vs. Displacement

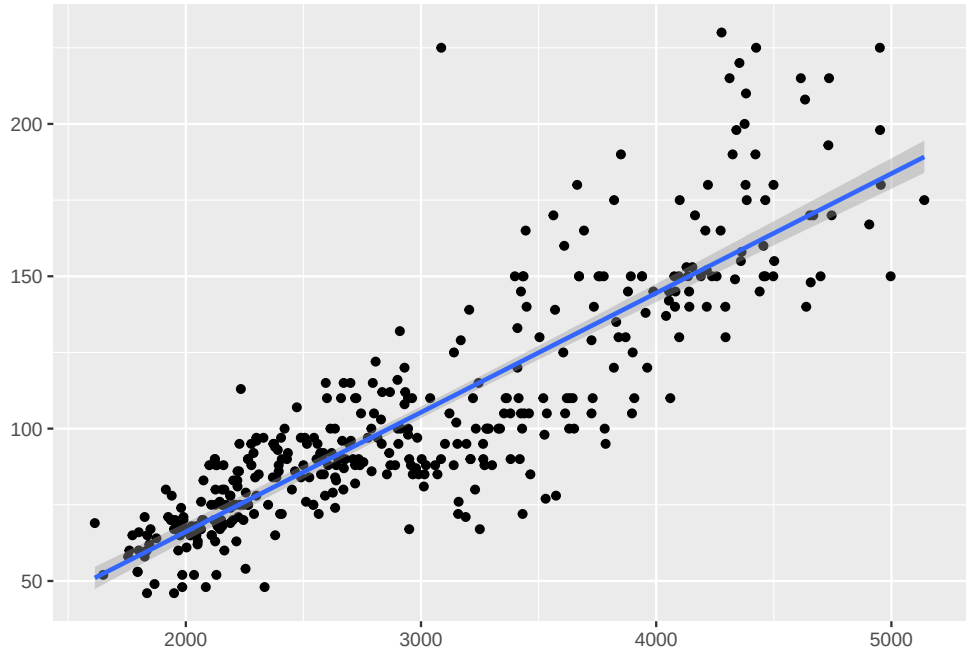
```
#highchart()%>%
  #hc_add_series(auto, "scatter", hcaes(x=weight, y=displacement))

plt(auto$weight, auto$displacement)
```

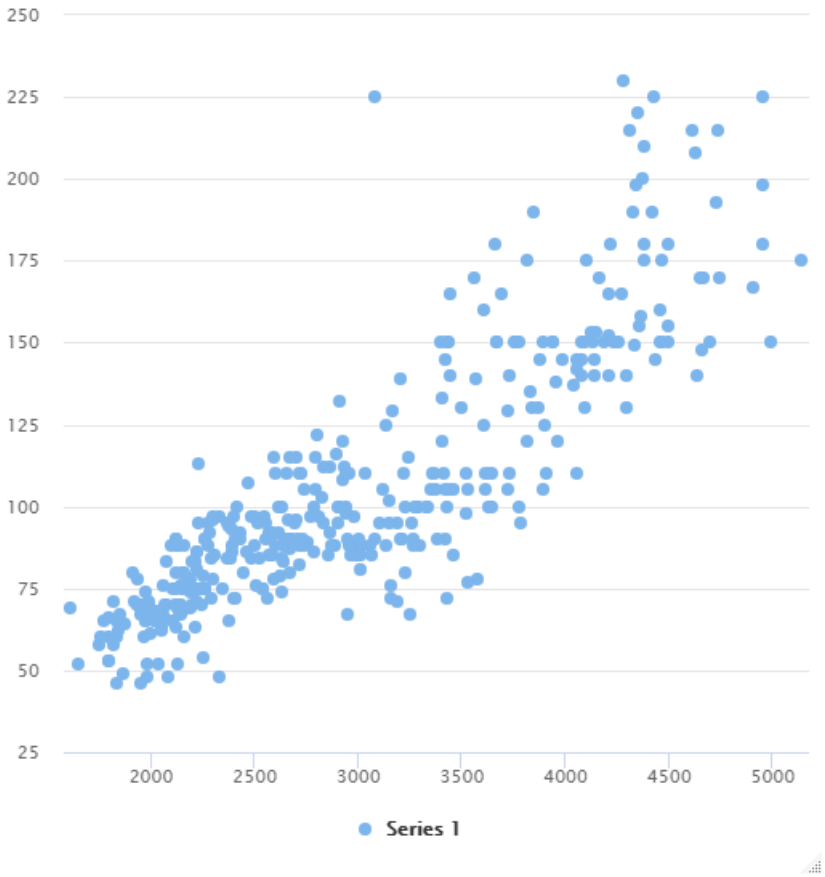


The heavier engine tends to have a larger size.

– Weight vs. Horsepower

```
#highchart()%>%
  #hc_add_series(auto, "scatter", hcaes(x=weight, y=horsepower))

plt(auto$weight, auto$horsepower)
```

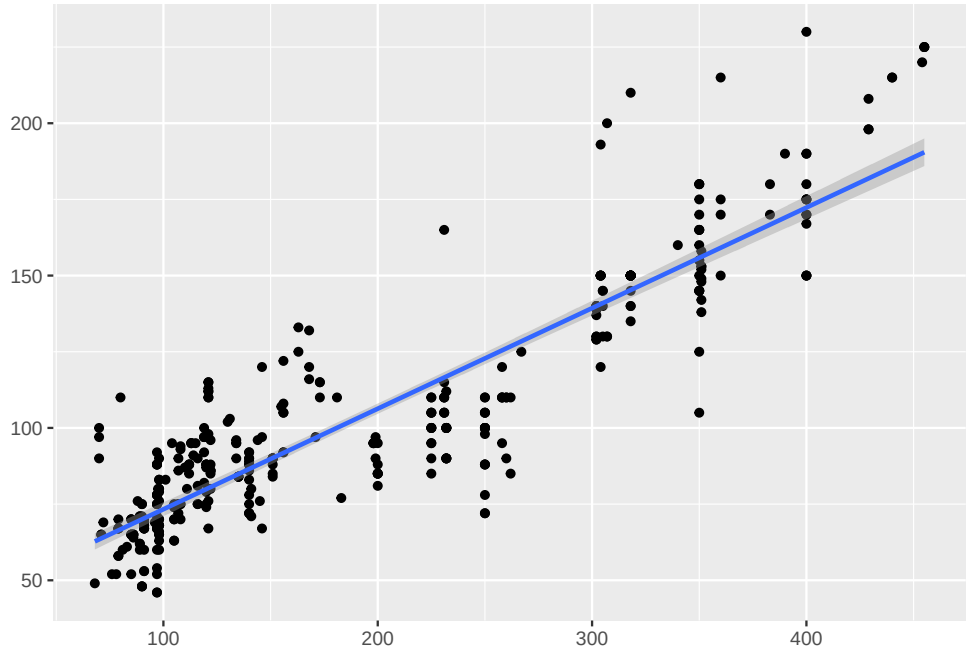


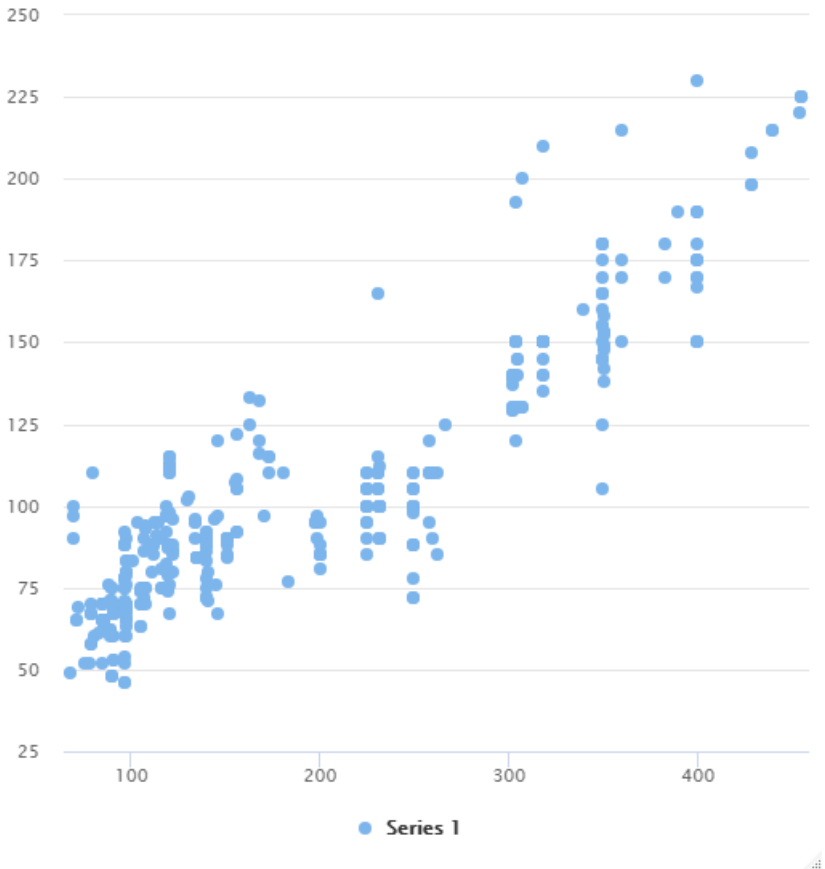


The heavier the engine, the more power it produced. (COOL!)

– Displacement vs. Horsepower

```
#highchart()%>%
  #hc_add_series(auto, "scatter", hcaes(x=displacement, y=horsepower))

plt(auto$displacement, auto$horsepower)
```

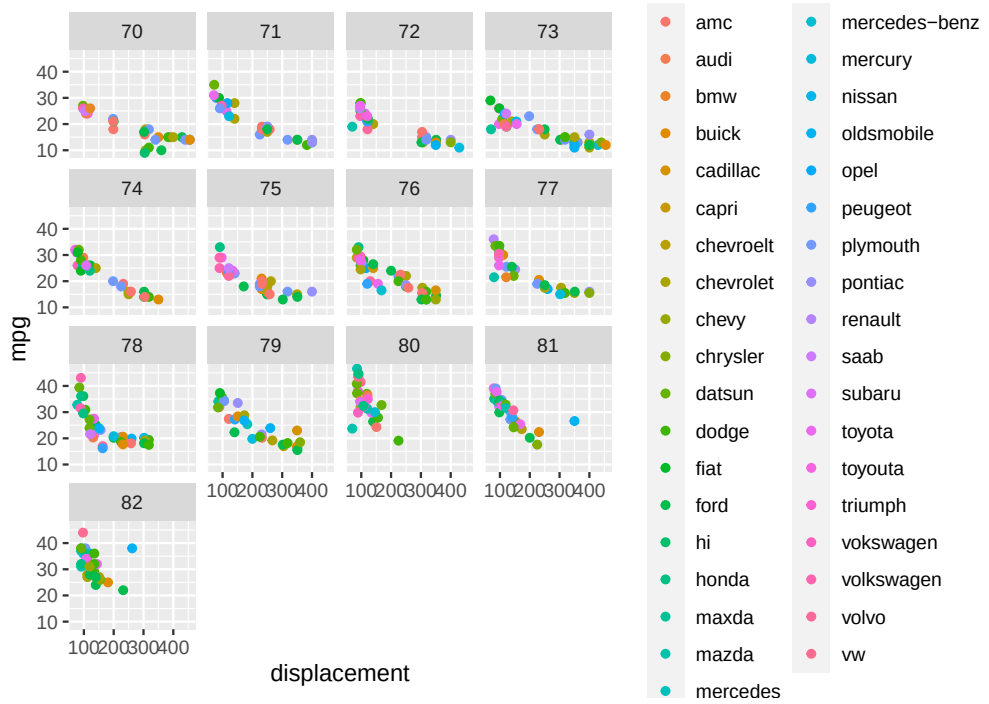




The larger the size of the car engine, the powerful it is. (COOL!)

– Displacement vs. mpg vs. Model_year

```
ggplot(data=auto)+
  aes(x = displacement, y=mpg, color=type)+
  geom_point()+
  facet_wrap(.~model_year)
```



The larger the engine size, the higher fuel efficiency the car.

- ***ii.*** Report a correlation table of all variables, rounding to two decimal places.

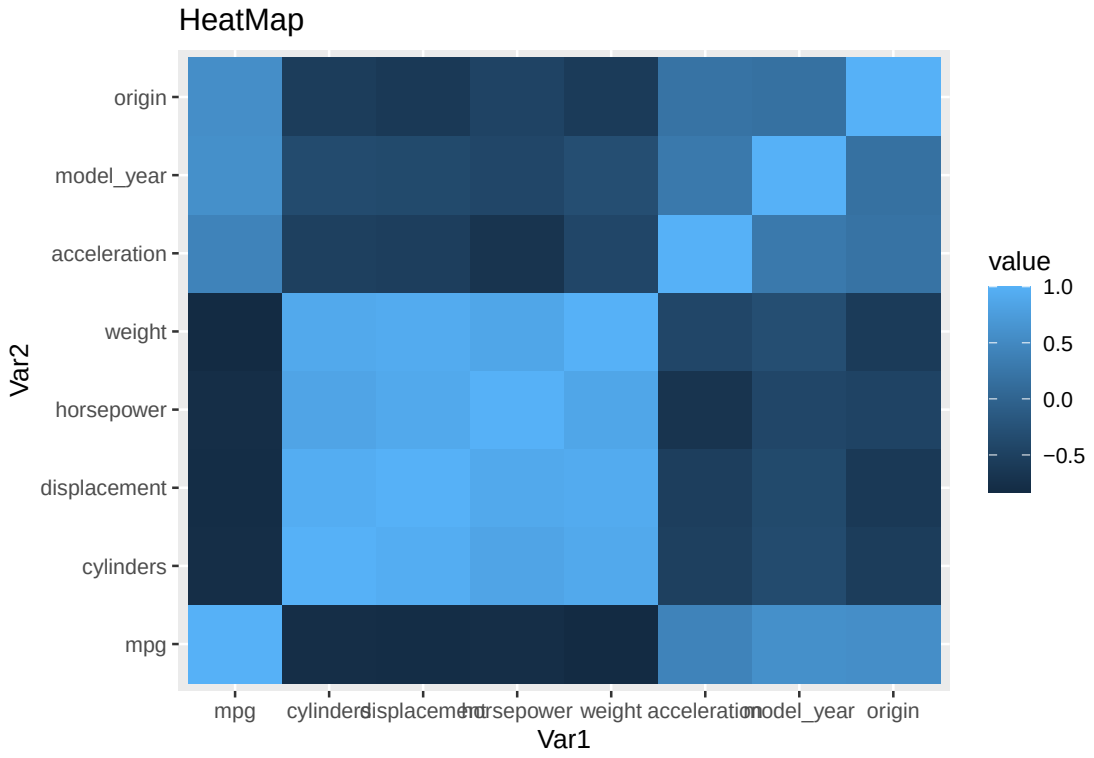
```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")

names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")

cormat <- round(cor(auto[,1:8], use="pairwise.complete.obs"), 2)
#View(cormat)
knitr::kable(cormat)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
mpg	1.00	-0.78	-0.80	-0.78	-0.83	0.42	0.58	0.56
cylinders	-0.78	1.00	0.95	0.84	0.90	-0.51	-0.35	-0.56
displacement	-0.80	0.95	1.00	0.90	0.93	-0.54	-0.37	-0.61
horsepower	-0.78	0.84	0.90	1.00	0.86	-0.69	-0.42	-0.46
weight	-0.83	0.90	0.93	0.86	1.00	-0.42	-0.31	-0.58
acceleration	0.42	-0.51	-0.54	-0.69	-0.42	1.00	0.29	0.21
model_year	0.58	-0.35	-0.37	-0.42	-0.31	0.29	1.00	0.18
origin	0.56	-0.56	-0.61	-0.46	-0.58	0.21	0.18	1.00

```
melted_cormat <- melt(cormat)
knitr::kable(head(melted_cormat))
```

Var1	Var2	value
mpg	mpg	1.00
cylinders	mpg	-0.78
displacement	mpg	-0.80
horsepower	mpg	-0.78
weight	mpg	-0.83
acceleration	mpg	0.42

```
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()+ggtitle('HeatMap')
```



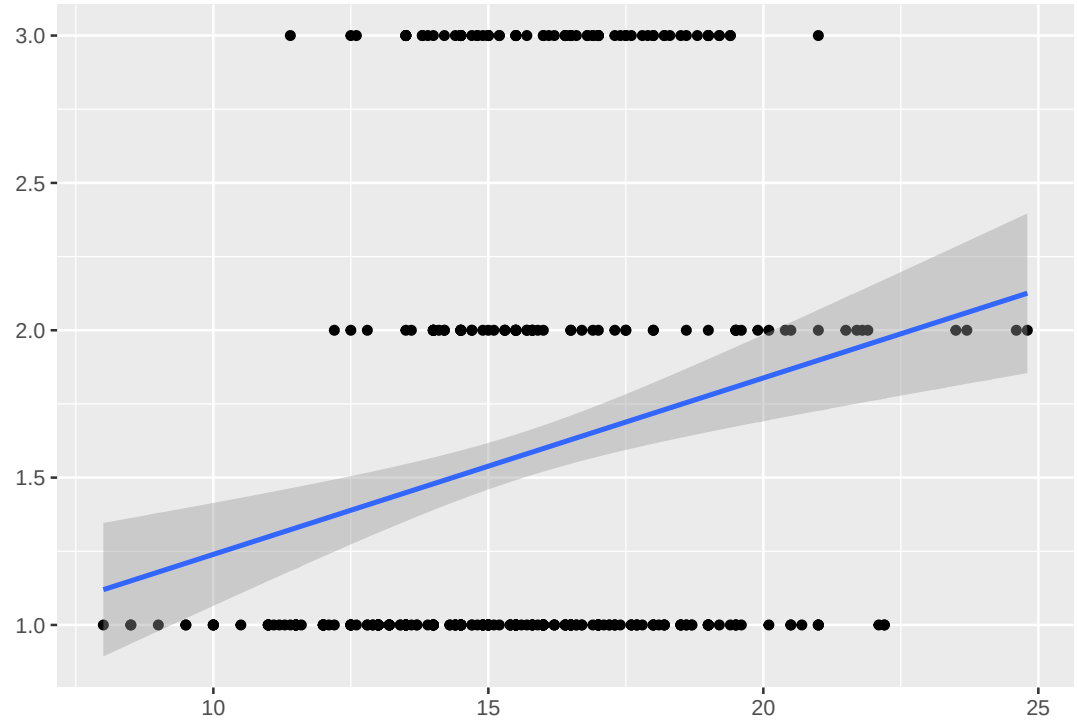
- ***iii.*** Which variables seem to relate to mpg?

Ans. According to the visualization and the correlation matrix, I think cylinders, displacement, horsepower and weight have a negative correlation respectively with mpg.

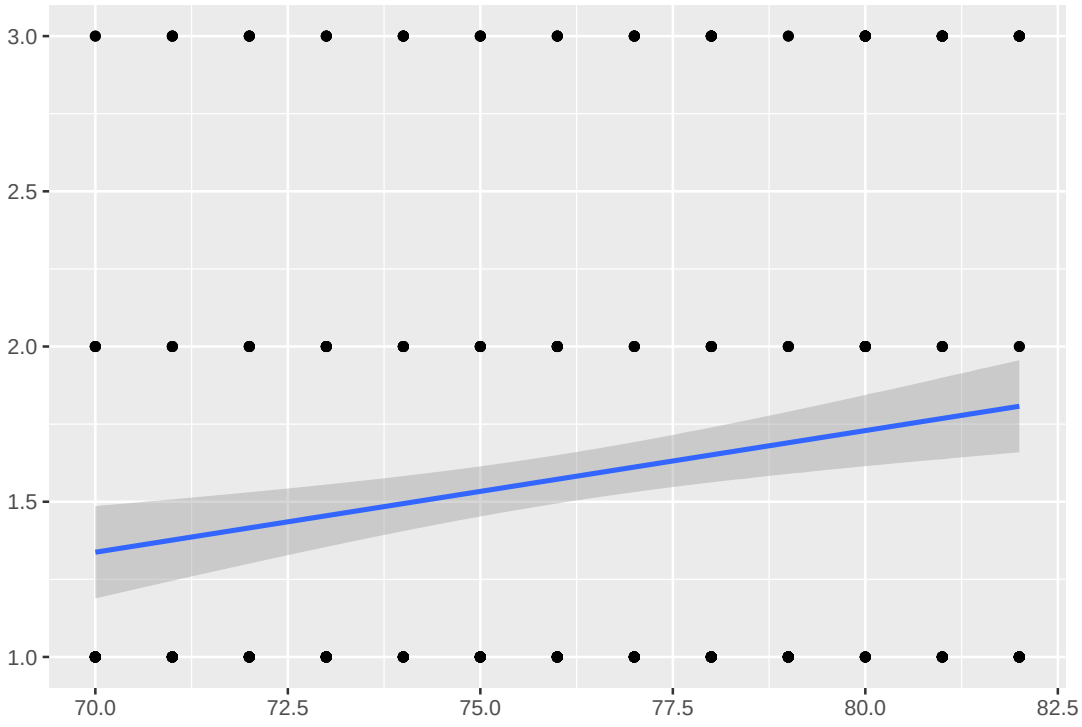
- ***iv.*** Which relationships might not be linear?

Ans. By observing the heatmap presented above, I think **`origin`** & **`acceleration`**, **`origin`** & **`model_year`**, **`acceleration`** & **`model_year`** are the least likely pairs of variables to have a linear relationship due to their correlation values (0.21, 0.18, 0.29)

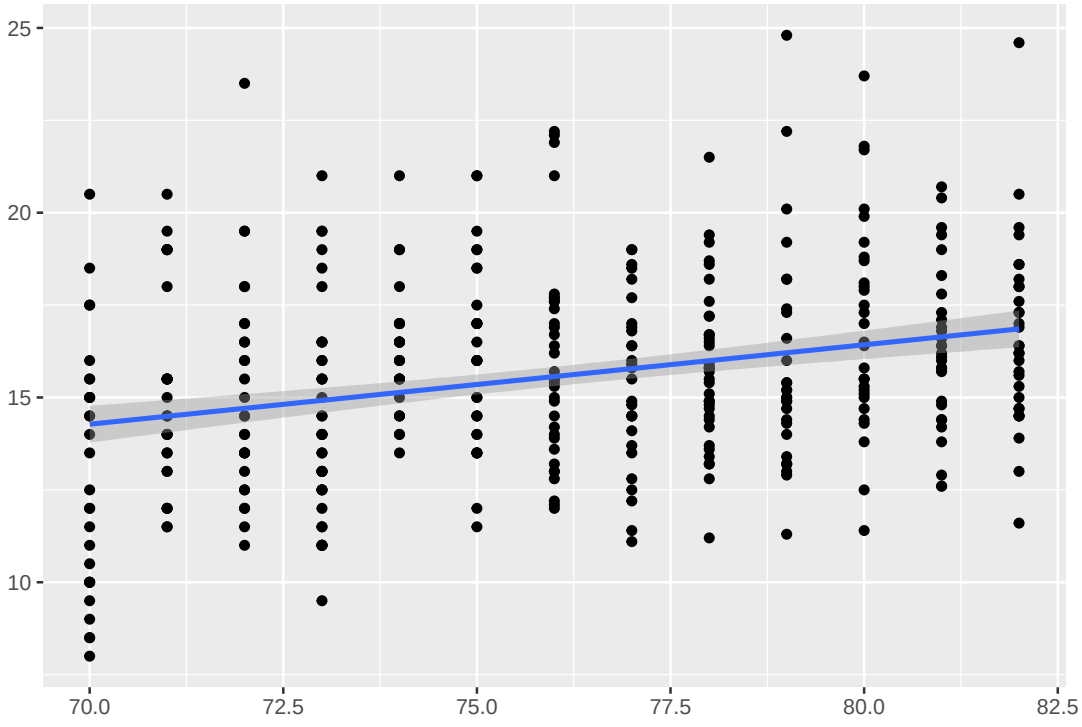
```
# Verify
plt(auto$acceleration, auto$origin)
```




```
plt(auto$model_year, auto$origin)
```




```
plt(auto$model_year, auto$acceleration)
```



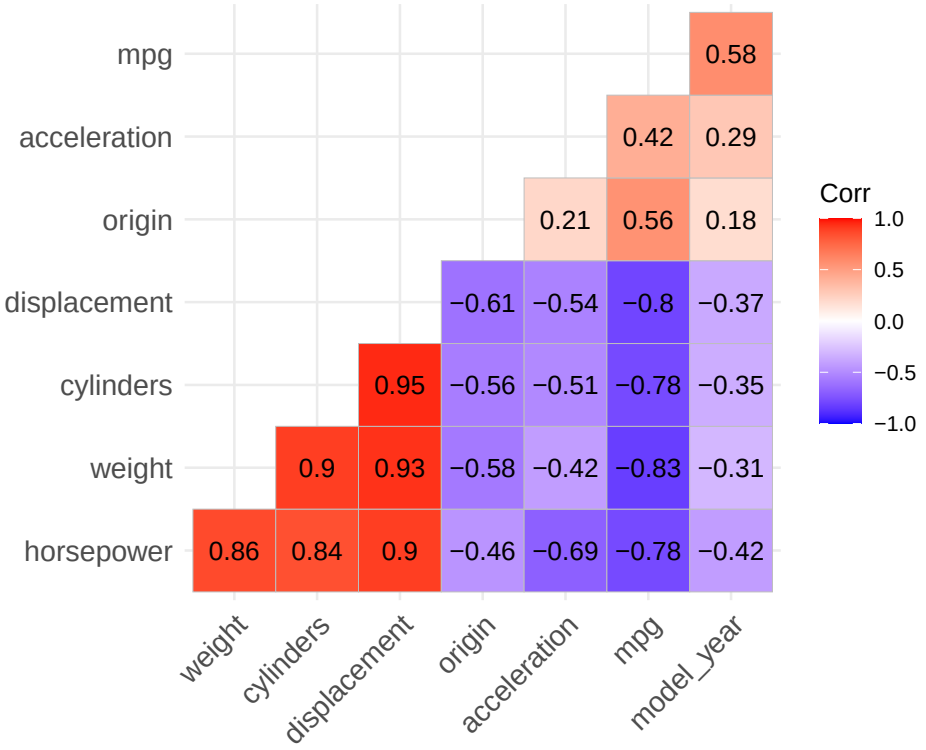
- ***v.*** Are there any pairs of independent variables that are highly correlated $r > 0.7$?

Ans. Yes.

```
cordf <- as.data.frame(cormat)
cormat[cormat<=0.7|cormat==1] <- ''
knitr::kable(head(cormat))
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
mpg								
cylinders			0.95	0.84	0.9			
displacement		0.95		0.9	0.93			
horsepower		0.84	0.9		0.86			
weight		0.9	0.93	0.86				
acceleration								

```
ggcorrplot(cordf, hc.order = TRUE, type = "lower",lab = TRUE)
```



As we can observe from the plot above, `displacement` has a highly positive correlation with `cylinders`, `weight` and `horsepower`; `cylinders` has a highly positive correlation with `weight`; while weight has a highly positive correlation with `horsepower`.

Note. mpg has a highly negative correlation with displacement, cylinders, weight and horsepower.

b. Create a linear regression model where `mpg` is dependent upon all other suitable variables.

- ***i.*** Which independent variables have a ***significant*** relationship with mpg at 1% significance?

```
summary(lm(mpg~cylinders+
             displacement+
             horsepower+
             weight+
             acceleration+
             model_year+
             factor(origin), data=auto[1:8]))
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year + factor(origin), data = auto[1:8])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders       -4.897e-01  3.212e-01  -1.524 0.128215
## displacement     2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower      -1.818e-02  1.371e-02  -1.326 0.185488
## weight          -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration     7.910e-02  9.822e-02   0.805 0.421101
## model_year       7.770e-01  5.178e-02  15.005  < 2e-16 ***
## factor(origin)2  2.630e+00  5.664e-01   4.643 4.72e-06 ***
## factor(origin)3  2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8242,   Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

Ans. cylinders, horsepower and acceleration.

- ***ii.*** Is it possible to determine which independent variables are the most effective at increasing mpg? If so, which ones, and if not, why not?

Ans. We can not rely on the above results to draw conclusion about which variables have the most power to make an effect at `mpg`. Hence, no, before standardizing the data, we are getting results that measure slopes, intercepts by different units. Comparing values based on different units is meaningless.

c. Resolve some of the issues with our regression model above.

- ***i.*** Create fully standardized regression results: are these slopes easier to compare?

```
summary(lm(mpg~cylinders+
             displacement+
             horsepower+
```

```
            weight+
            acceleration+
            model_year, data=as.data.frame(scale(auto[1:7]))))
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year, data = as.data.frame(scale(auto[1:7])))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11217 -0.30532 -0.01025  0.25961  1.83734
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0004236  0.0222112   0.019    0.985
## cylinders    -0.0717877  0.0722763  -0.993    0.321
## displacement  0.1024348  0.0981565   1.044    0.297
## horsepower   -0.0019273  0.0681403  -0.028    0.977
## weight       -0.7361794  0.0725952 -10.141   <2e-16 ***
## acceleration  0.0300867  0.0360009   0.836    0.404
## model_year    0.3564069  0.0248929  14.318   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4395 on 385 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8093,   Adjusted R-squared:  0.8063
## F-statistic: 272.2 on 6 and 385 DF,  p-value: < 2.2e-16
```

Ans. Yes. Standardizing data allows data to be compared to each others in the same units. As we can observe from the results, by reducing `weights` of a vehicle, `mpg` would be able to increase effectively.

- ***ii.*** Regress `mpg` over each **nonsignificant** independent variable, individually. Which ones become significant when we regress mpg over them individually?

```
# weight
summary(lm(mpg~weight, data=as.data.frame(auto[c('mpg','weight')])))
```

```
##
## Call:
## lm(formula = mpg ~ weight, data = as.data.frame(auto[c("mpg",
##     "weight")]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.012  -2.801  -0.351   2.114  16.480
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.3173644  0.7952452   58.24   <2e-16 ***
## weight      -0.0076766  0.0002575  -29.81   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.345 on 396 degrees of freedom
## Multiple R-squared:  0.6918,   Adjusted R-squared:  0.691
## F-statistic: 888.9 on 1 and 396 DF,  p-value: < 2.2e-16
```

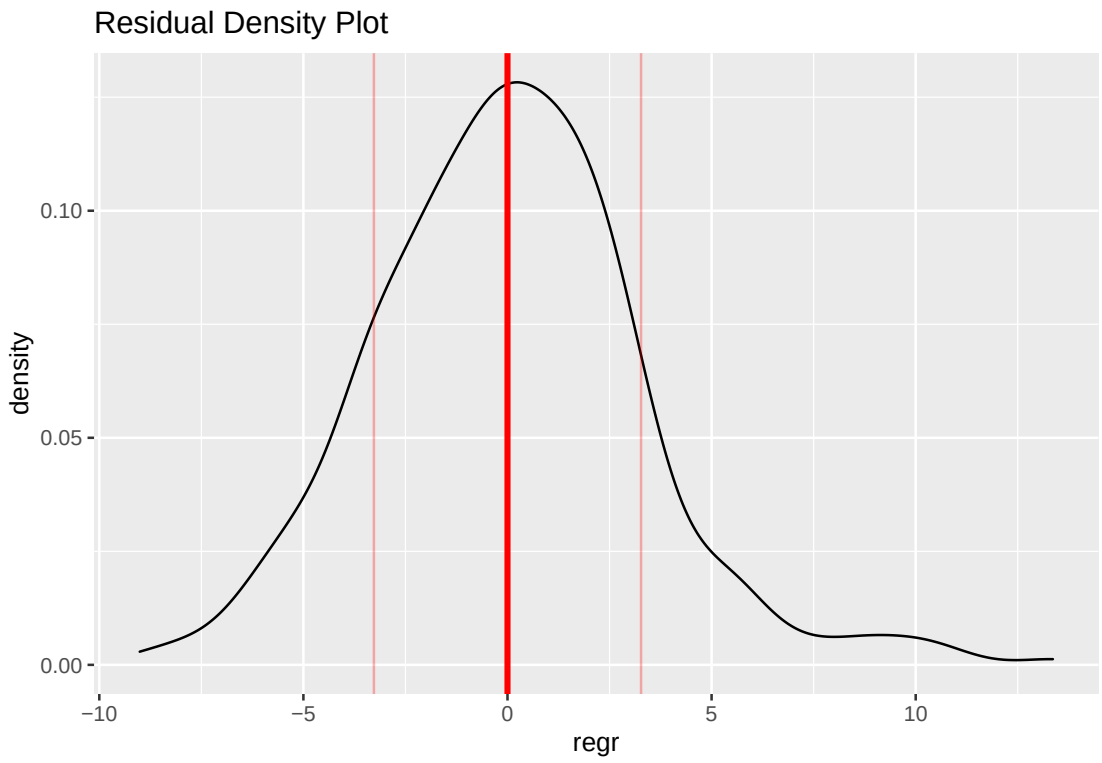
```
# model_year
summary(lm(mpg~model_year, data=as.data.frame(auto[c('mpg','model_year')])))
```

```
##
## Call:
## lm(formula = mpg ~ model_year, data = as.data.frame(auto[c("mpg",
##     "model_year")]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.024  -5.451  -0.390   4.947  18.200
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -69.55560    6.58911  -10.56   <2e-16 ***
## model_year    1.22445    0.08659   14.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.379 on 396 degrees of freedom
## Multiple R-squared:  0.3356,   Adjusted R-squared:  0.3339
## F-statistic:   200 on 1 and 396 DF,  p-value: < 2.2e-16
```

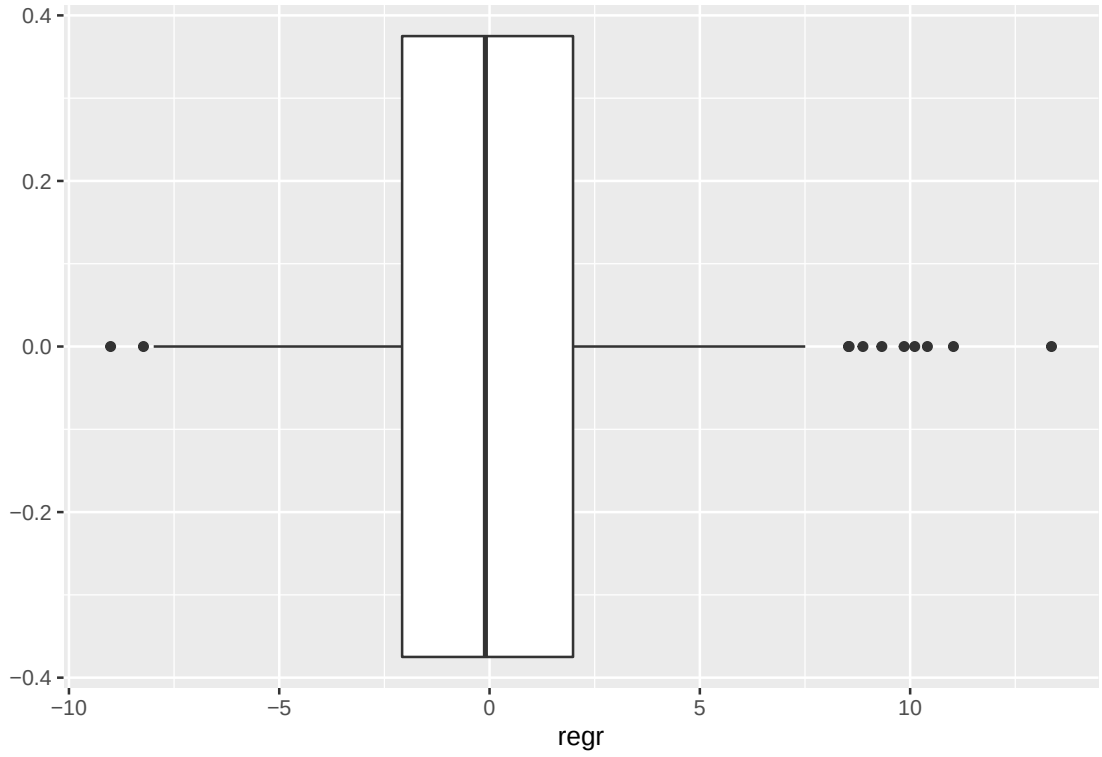
Ans. Both of them become significant after regressing `mpg` over them.

- ***iii.*** Plot the density of the residuals: are they normally distributed and centered around zero?

```
regr <- lm(mpg~cylinders+
             displacement+
             horsepower+
             weight+
             acceleration+
             model_year+
             factor(origin), data=auto[1:8])$residuals
ggplot()+
  aes(regr)+
  geom_density()+
  ggtitle('Residual Density Plot')+
  geom_vline(xintercept = 0, colour='red', lwd=1.2)+
  geom_vline(xintercept = c(sd(regr), -sd(regr)), colour='red', alpha=0.3)
```




```
ggplot()+
  aes(regr)+
  geom_boxplot()
```



Ans. Yes~