BACS HW11

109062710

5/5/2021

Question 1

a. Let's dig into what regression is doing to compute model fit

```
pts <- data.frame(
  x = c(
    -4.704724, 3.966620, 3.448928, 11.861426, 11.473157,
    20.662193, 18.462001, 27.909883, 25.709691, 35.416420,
    32.957382, 41.887572, 41.369880, 49.652955, 7.331619
  ),
  y = c(
    4.682789, -1.593332, 14.096971, 7.472177, 22.465133,
    16.537685, 31.879314, 25.951867, 41.293496, 29.089927,
    45.826250, 39.550129, 50.010331, 48.615637, 5.728810
  )
)
```

i. Plot Scenario 2, storing the returned points

```
regr <- lm(y ~ x, data = pts)
summary(regr)
```

ii. Run a linear model of x and y points to confirm the R2 value reported by the simulation

```
##
## Call:
## lm(formula = y ~ x, data = pts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.299  -6.970  -2.898   6.492  12.215
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.9885     3.6651   1.361    0.197
```

```
## x             0.9370     0.1362   6.878 1.12e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.298 on 13 degrees of freedom
## Multiple R-squared:  0.7844, Adjusted R-squared:  0.7678
## F-statistic:  47.3 on 1 and 13 DF,  p-value: 1.123e-05
```

iii. Add line segments to the plot

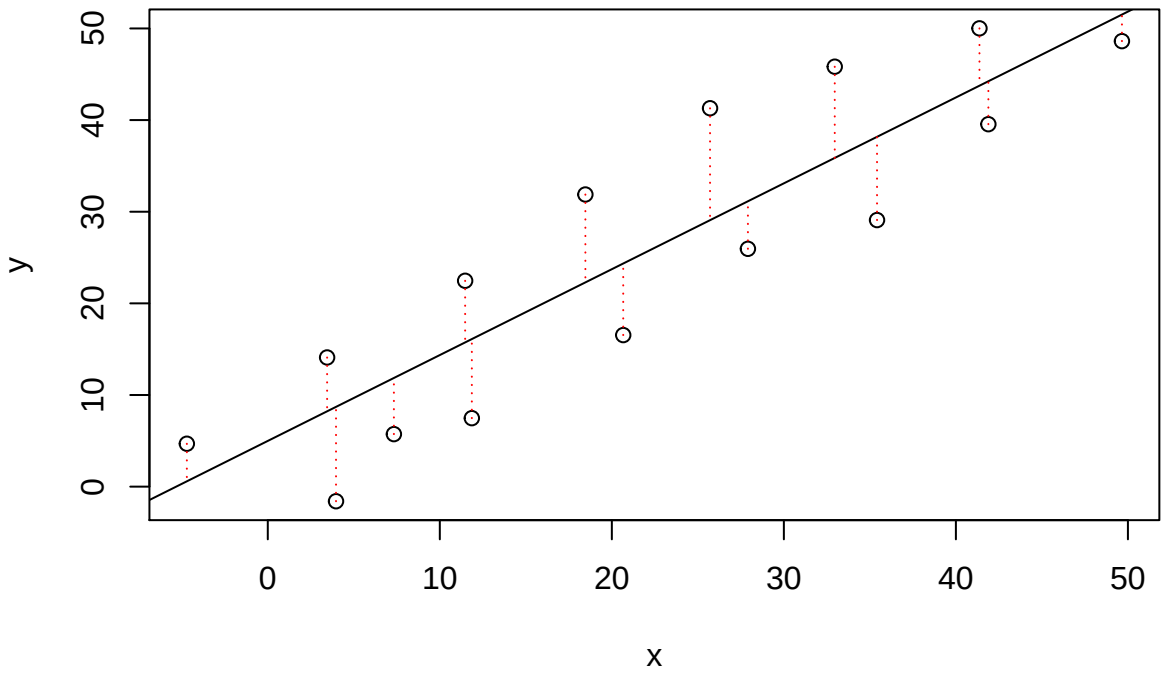
1. Get values of $\hat{y}$ (estimated values)

```
y_hat <- regr$fitted.values
y_hat
```

```
##          1          2          3          4          5          6          7
##  0.5801637  8.7052424  8.2201632 16.1027022 15.7388929 24.3490507 22.2874633
##          8          9         10         11         12         13         14
## 31.1401607 29.0785733 38.1738112 35.8696843 44.2373025 43.7522233 51.5134926
##         15
## 11.8582578
```

2. Add segments

```
plot(pts)
abline(lm(pts$y ~ pts$x))
segments(pts$x, pts$y, pts$x, y_hat, col="red", lty="dotted")
```




```
sse <- sum((fitted(regr) - mean(pts$y))^2)
ssr <- sum((fitted(regr) - pts$y)^2)
sst <- sse + ssr
r2 <- 1 - (ssr / sst)
```

iv. Use only `pts$x`, `pts$y`, `y_hat` and `mean(pts$y)` to compute SSE, SSR and SST, and verify R^2

b. Comparing scenarios 1 and 2, which do we expect to have a stronger R^2? For the first scenario, R^2 will be very close to $+1$ since most of the data points are sitting at or close to the increasing regression line. However, the second scenario's R^2 value won't be as high as the first scenario's, but it still will be near 1.

In this case, the first scenario will have a stronger R^2.

c. Comparing scenarios 3 and 4, which do we expect to have a stronger R^2? In the third scenario, the R^2 will be close to -1 since most of the data points are sitting on or close to the decreasing regression line. However, the fourth scenario's R^2 value won't be as high as the third scenario's, but it still will be near -1.

In this case, the third scenario will have a stronger R^2, but in a decreasing manner.

d. Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST?
In first scenario, we expect smaller SSE & SST. Clearly, SST value for the first scenario will be also small in this case.

In the second scenario, SSE will be bigger since all the data points are quite far from the regression line. However, the distance from the mean value might or might not be close to the regression line. Clearly, the second scenario will have bigger SST because of SSE, regardless SSR.

e. Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST?
The third scenario is quite similar to the first scenario since most data points are sitting quite close to the regression line. Thus, SST will be smaller compared to the fourth scenario due to small SSE and SSR values.

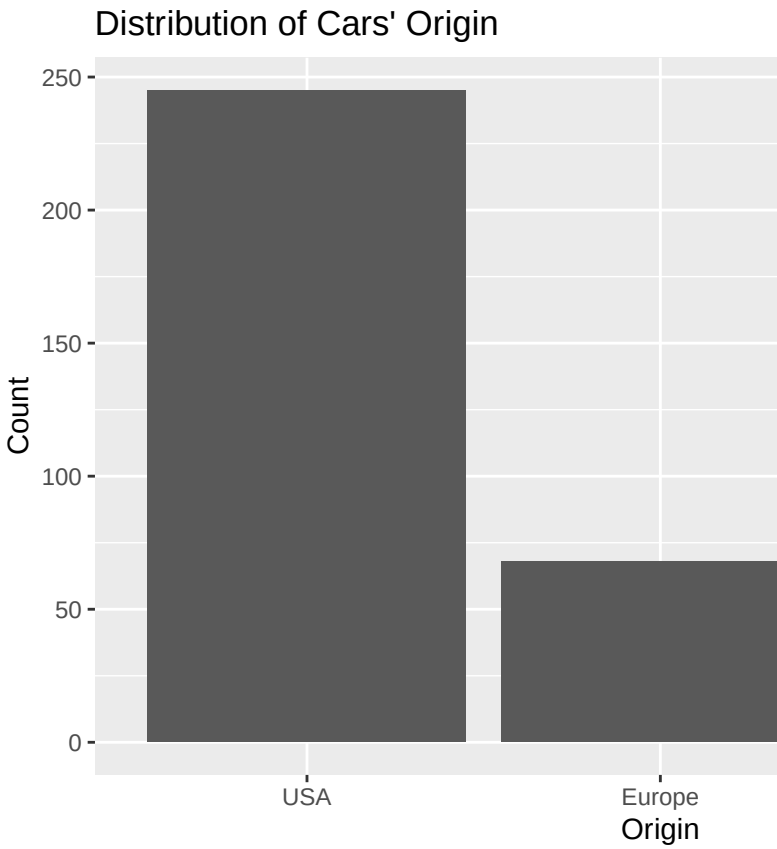
The fourth scenario is also similar to the second scenario where SSE is big since most data points are sitting far from the regression line. Clearly, SST value will be bigger compared to the third scenario due to SSE, regardless SSR value.

Question 2

```
auto <- Auto
origins <- c("USA", "Europe", "Japan")
auto$origin <- factor(auto$origin, labels = origins)
```

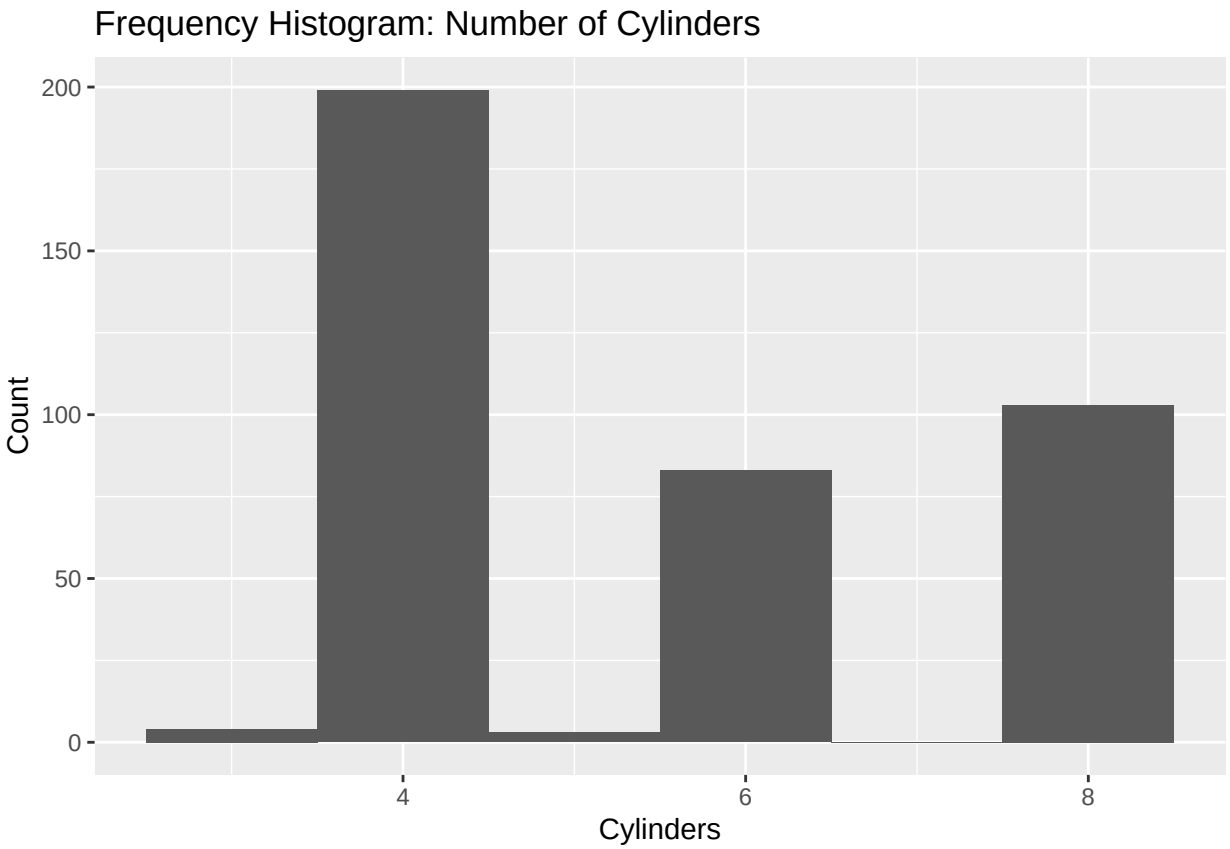
a. Let's first try exploring this data and problem

```
qplot(
  auto$origin,
  xlab = "Origin",
  ylab = "Count",
  main = "Distribution of Cars' Origin"
)
```

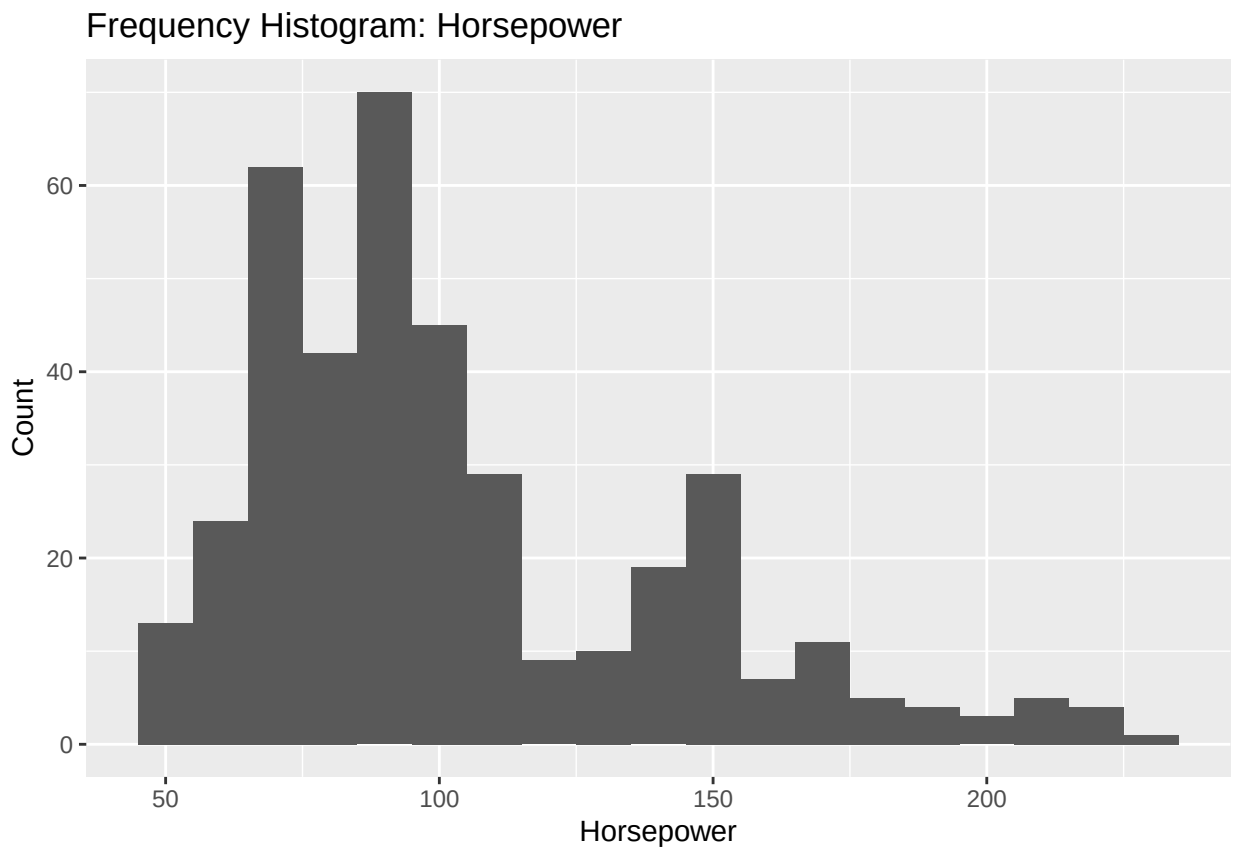


i. Visualize the data in any way you feel relevant

```
qplot(auto$cylinders, xlab = 'Cylinders', ylab = 'Count',
      main='Frequency Histogram: Number of Cylinders', binwidth = 1)
```

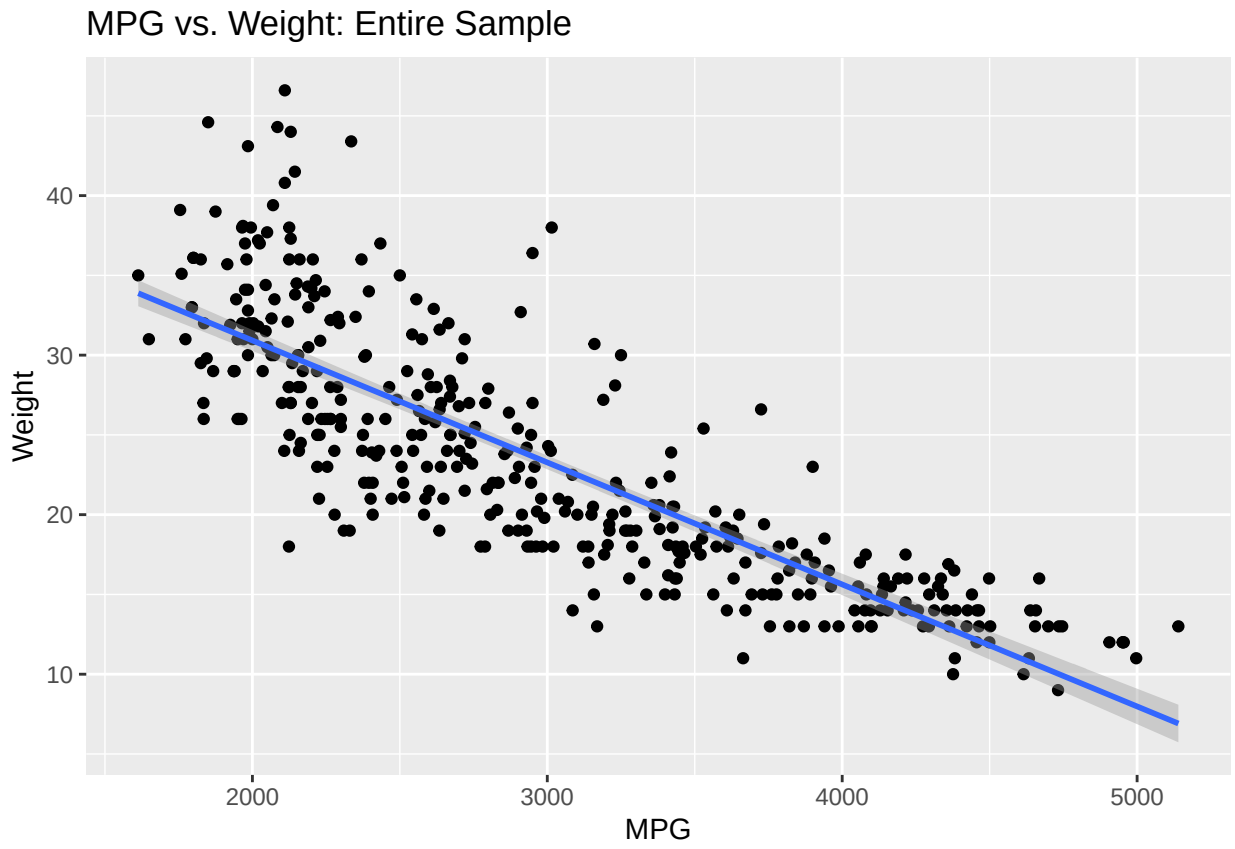



```
qplot(auto$horsepower, xlab = 'Horsepower', ylab = 'Count', binwidth = 10,
      main='Frequency Histogram: Horsepower')
```




```
ggplot(data = auto, aes(x = weight, y = mpg)) +
  geom_point() +
  geom_smooth(method = lm) +
  xlab('MPG') +
  ylab('Weight') +
  ggtitle('MPG vs. Weight: Entire Sample')
```

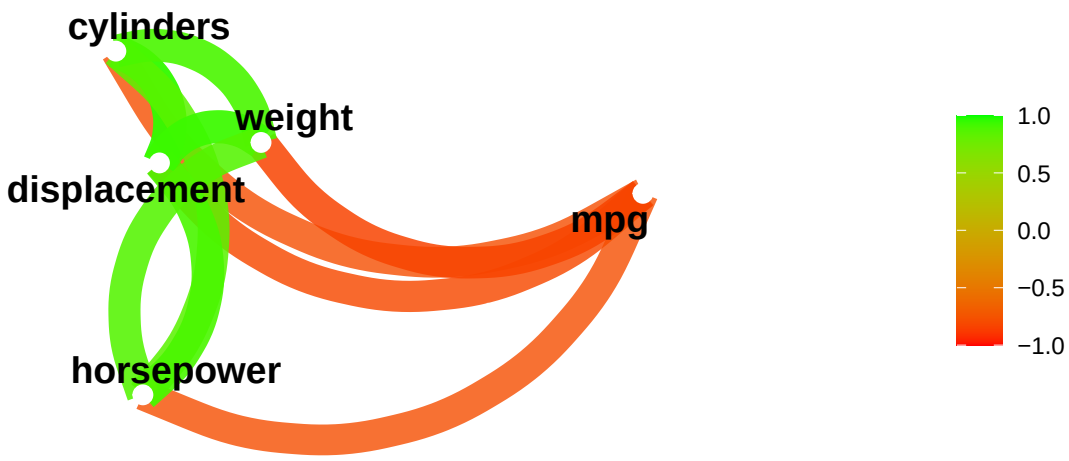
```
## `geom_smooth()` using formula 'y ~ x'
```



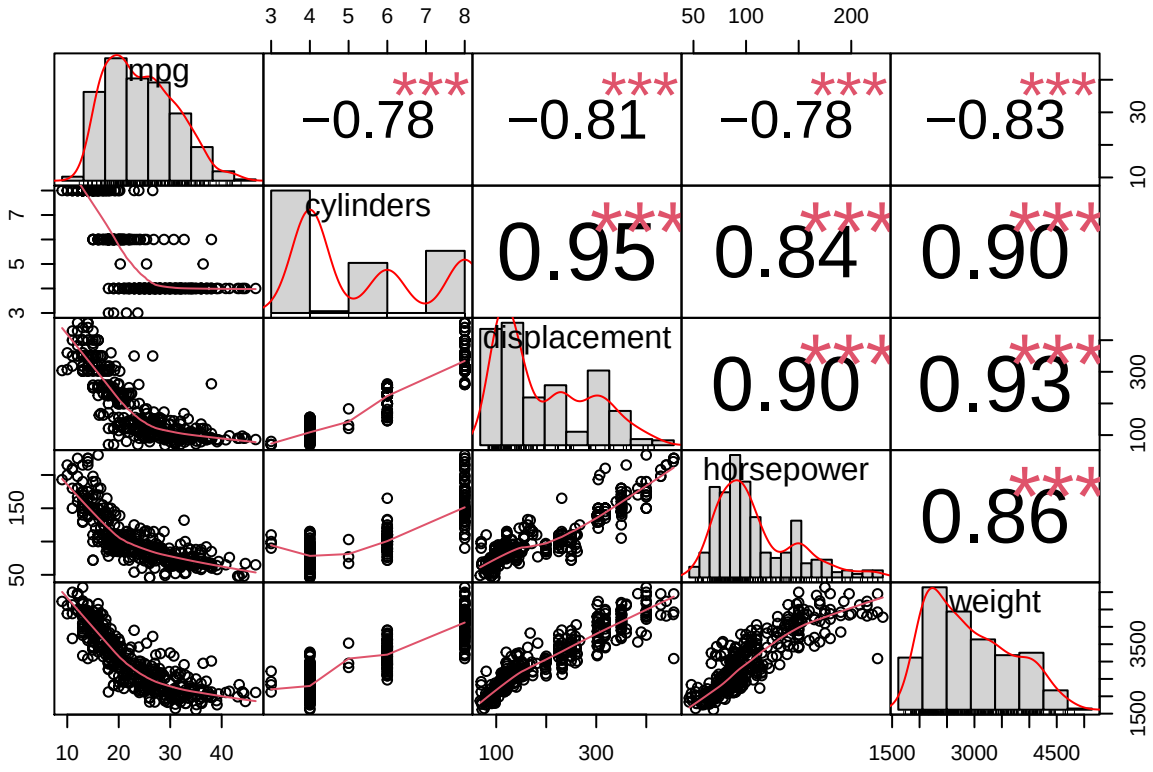

```
variables <-
  auto[,
       c("mpg",
         "cylinders",
         "displacement",
         "horsepower",
         "weight")
  ]
variables %>%
  correlate() %>%
  network_plot(min_cor = 0.7, colors = c("red", "green"), legend = TRUE)
```

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

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```




```
chart.Correlation(variables, histogram=TRUE, pch=19)
```



iii. From the visualizations and correlations, which variables seem to relate to mpg? From the correlation graph, seems like `cylinders`, `displacement`, `horsepower`, and `weight` relare to `mpg`. However, they are negatively correlated. Besides, `weight` is the only variable with the correlation value closest -1 meaning it's the most negatively correlated variable against `mpg`.

iv. Which relationships might not be linear? The relationships between `mpg` with `displacement`, `horsepower` and `weight` are not linear, indicated by the moon-shaped curve that the data points follow. Refer to the graph in (iii)

v. Are there any pairs of independent variables that are highly correlated r > 0.7? Yes, there are `cylinders`, `displacement`, `horsepower` and `weight`. The values can be seen in (ii) and (iii) answers.

```
simple_regr <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + factor(o
summary(simple_regr)
```

b. Let's create a linear regression model where mpg is dependent upon all other suitable variables

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
```

```
##     acceleration + year + factor(origin), data = auto)
##
## 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
## year                  7.770e-01  5.178e-02  15.005  < 2e-16 ***
## factor(origin)Europe  2.630e+00  5.664e-01   4.643 4.72e-06 ***
## factor(origin)Japan   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
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

i. Which independent variables have a 'significant' relationship with mpg at 1% significance? There are five independent variables that have significant relationships with mpg at 1% significance, namely `displacement`, `weight`, `year`, `factor(origin)Europe`, and `factor(origin)Japan`.

ii. Looking at the coefficients, is it possible to determine which independent variables are the most effective at increasing mpg? If so, which ones, and if not, why not? We can't determine which and which are effective since all the independent variables are not standardized.

c. Let's try to resolve some of the issues with our regression model above.

```
auto_sd <- cbind(scale(auto[1:7]), auto$origin)
colnames(auto_sd)[8] <- "origin"
auto_sd <- as.data.frame(auto_sd)
auto_sd$origin <- factor(auto$origin, labels = origins)
simple_regr_std <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + facto
summary(simple_regr_std)
```

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

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + factor(origin), data = auto_sd)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.15432 -0.26630 -0.01259  0.25440  1.71182
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.13213    0.03155  -4.187 3.50e-05 ***
## cylinders            -0.10703    0.07020  -1.524  0.12821
## displacement          0.32149    0.10261   3.133  0.00186 **
## horsepower           -0.08967    0.06761  -1.326  0.18549
## weight               -0.73028    0.07130 -10.243  < 2e-16 ***
## acceleration          0.02796    0.03472   0.805  0.42110
## year                  0.36673    0.02444  15.005  < 2e-16 ***
## factor(origin)Europe  0.33696    0.07257   4.643 4.72e-06 ***
## factor(origin)Japan   0.36556    0.07082   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4236 on 383 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
mpg_cyl.lm <- lm(mpg ~ cylinders, data = auto_sd)
summary(mpg_cyl.lm)
```

ii. Regress mpg over each non significant independent variable, individually. Which ones become significant when we regress mpg over them individually?

```
##
## Call:
## lm(formula = mpg ~ cylinders, data = auto_sd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82463 -0.40784 -0.08113  0.32660  2.29555
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.731e-16  3.180e-02    0.00        1
## cylinders   -7.776e-01  3.184e-02  -24.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6295 on 390 degrees of freedom
## Multiple R-squared:  0.6047, Adjusted R-squared:  0.6037
## F-statistic: 596.6 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
mpg_hp.lm <- lm(mpg ~ horsepower, data = auto_sd)
summary(mpg_hp.lm)
```

```
##
```

```
## Call:
## lm(formula = mpg ~ horsepower, data = auto_sd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73876 -0.41757 -0.04402  0.35401  2.16836
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.213e-16  3.175e-02    0.00        1
## horsepower  -7.784e-01  3.179e-02  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6285 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

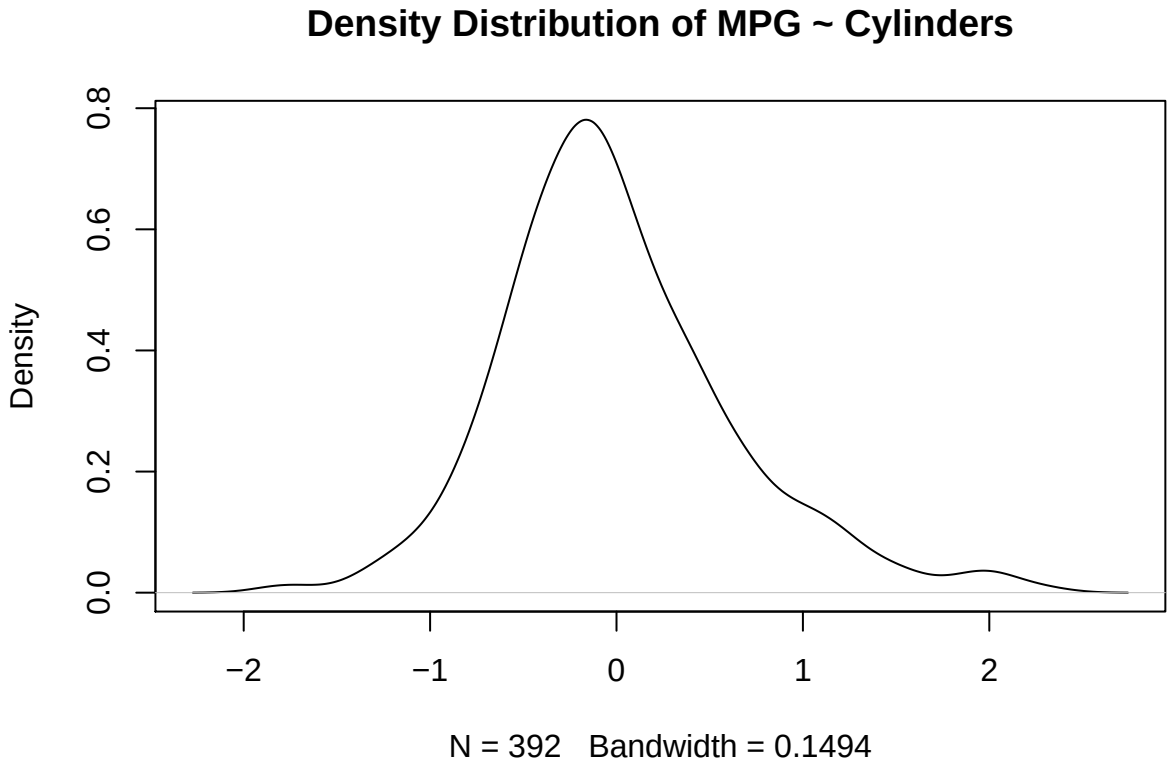
```
mpg_acc.lm <- lm(mpg ~ acceleration, data = auto_sd)
summary(mpg_acc.lm)
```

```
##
## Call:
## lm(formula = mpg ~ acceleration, data = auto_sd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3048 -0.7195 -0.1536  0.6151  2.9775
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.427e-16  4.582e-02   0.000        1
## acceleration 4.233e-01  4.588e-02   9.228   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9071 on 390 degrees of freedom
## Multiple R-squared:  0.1792, Adjusted R-squared:  0.1771
## F-statistic: 85.15 on 1 and 390 DF,  p-value: < 2.2e-16
```

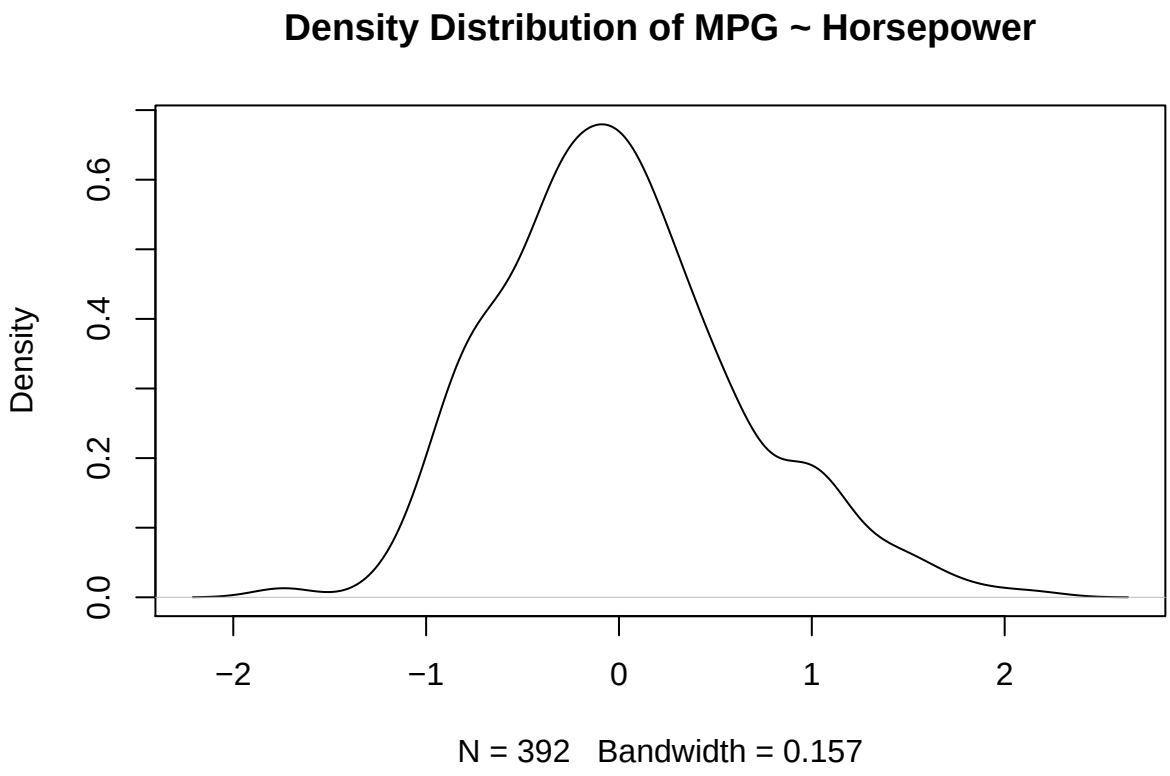
All of them became significant.

```
mpg_cyl.res <- resid(mpg_cyl.lm)
plot(density(mpg_cyl.res), main = "Density Distribution of MPG ~ Cylinders")
```

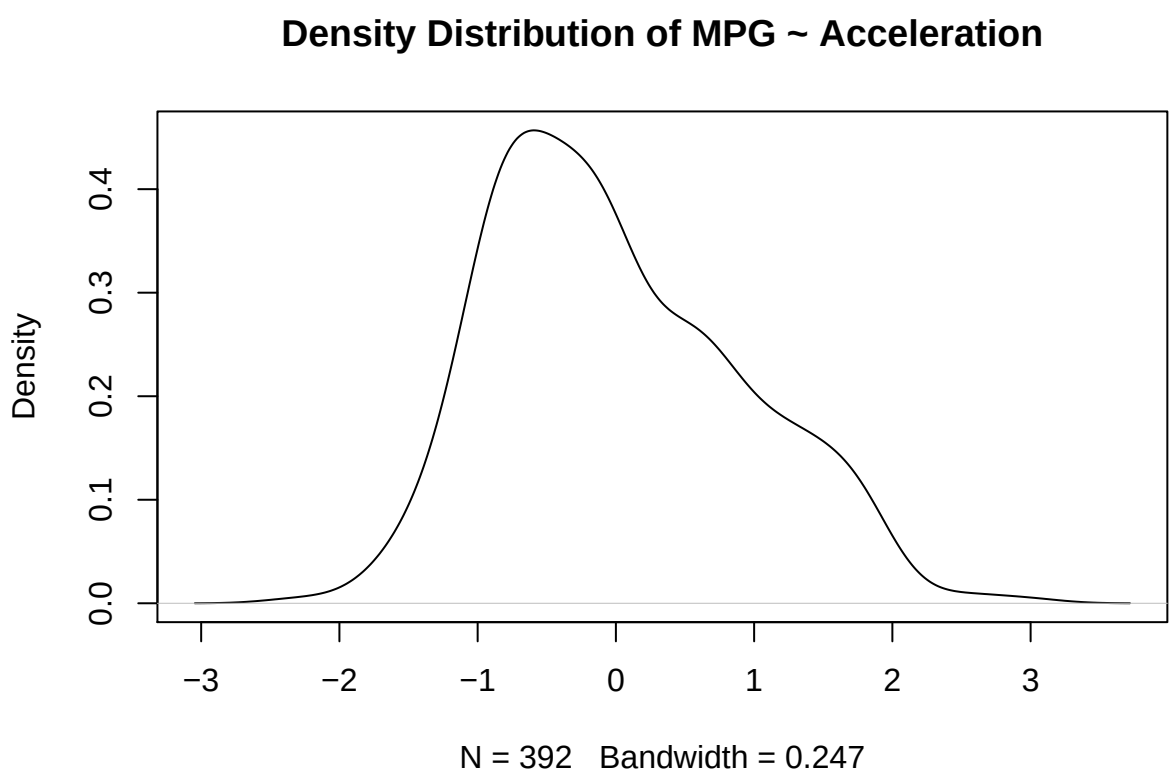
iii. Plot the density of the residuals: are they normally distributed and centered around zero?




```
mpg_hp.res <- resid(mpg_hp.lm)
plot(density(mpg_hp.res), main = "Density Distribution of MPG ~ Horsepower")
```




```
mpg_acc.res <- resid(mpg_acc.lm)
plot(density(mpg_acc.res), main = "Density Distribution of MPG ~ Acceleration")
```



They are all normally distributed and centered around zero.

Reference:

R SQUARED: SST, SSE AND SSR

The Correlation Coefficient (r)

Significance Codes in R