Z604_HW1_Nishant_Shah_nishshah

Preprocessing data:

```
mpg_data <- read.table("auto-mpg.data",na.strings = "?")
colnames(mpg_data) <- c("mpg","cylinders","displacement","horsepower","weight","accelerat
ion","model_year","origin","car_name")
mpg_data <- na.omit(mpg_data)
```

Top five rows for data:

```
head(mpg_data)
```

```
##   mpg cylinders displacement horsepower weight acceleration model_year
## 1  18         8          307        130   3504         12.0         70
## 2  15         8          350        165   3693         11.5         70
## 3  18         8          318        150   3436         11.0         70
## 4  16         8          304        150   3433         12.0         70
## 5  17         8          302        140   3449         10.5         70
## 6  15         8          429        198   4341         10.0         70
##   origin                  car_name
## 1      1 chevrolet chevelle malibu
## 2      1         buick skylark 320
## 3      1        plymouth satellite
## 4      1             amc rebel sst
## 5      1               ford torino
## 6      1          ford galaxie 500
```

Structure of Data:

```
str(mpg_data)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model_year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ car_name    : Factor w/ 305 levels "amc ambassador brougham",..: 50 37 232 15 162 1
42 55 224 242 2 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:6] 33 127 331 337 355 375
##   .. ..- attr(*, "names")= chr [1:6] "33" "127" "331" "337" ...
```

We can see that R has co-erced “cylinders”,“model_year”,“origin” to descrete variables instead of categorical.

```
is.factor(mpg_data$cylinders)
```

```
## [1] FALSE
```

```
is.factor(mpg_data$model_year)
```

```
## [1] FALSE
```

```
is.factor(mpg_data$origin)
```

```
## [1] FALSE
```

So we need to process that as well:

```
mpg_data$cylinders <- factor(mpg_data$cylinders)
mpg_data$model_year <- factor(mpg_data$model_year)
mpg_data$origin <- factor(mpg_data$origin)
```

New structure of Data:

```
str(mpg_data)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : Factor w/ 5 levels "3","4","5","6",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model_year  : Factor w/ 13 levels "70","71","72",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ origin      : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ car_name    : Factor w/ 305 levels "amc ambassador brougham",..: 50 37 232 15 162 1
42 55 224 242 2 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:6] 33 127 331 337 355 375
##   .. ..- attr(*, "names")= chr [1:6] "33" "127" "331" "337" ...
```

Answers to homework questions:

1). Dependent variable:

```
## [1] "mpg"
```

Depdent variables:

```
## [1] "cylinders"    "displacement" "horsepower"   "weight"
## [5] "acceleration" "model_year"   "origin"       "car_name"
```

2). Summary statistics and Standard Deviation each numerical variable:

```
summary(mpg_data)
```

```
##       mpg        cylinders  displacement     horsepower        weight
##  Min.   : 9.00   3:  4     Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.00   4:199     1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.75   5:  3     Median :151.0   Median : 93.5   Median :2804
##  Mean   :23.45   6: 83     Mean   :194.4   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.00   8:103     3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.60             Max.   :455.0   Max.   :230.0   Max.   :5140
##
##   acceleration     model_year  origin                car_name
##  Min.   : 8.00   73     : 40   1:245   amc matador       :  5
##  1st Qu.:13.78   78     : 36   2: 68   ford pinto        :  5
##  Median :15.50   76     : 34   3: 79   toyota corolla    :  5
##  Mean   :15.54   75     : 30           amc gremlin       :  4
##  3rd Qu.:17.02   82     : 30           amc hornet        :  4
##  Max.   :24.80   70     : 29           chevrolet chevette:  4
##                  (Other):193           (Other)           :365
```

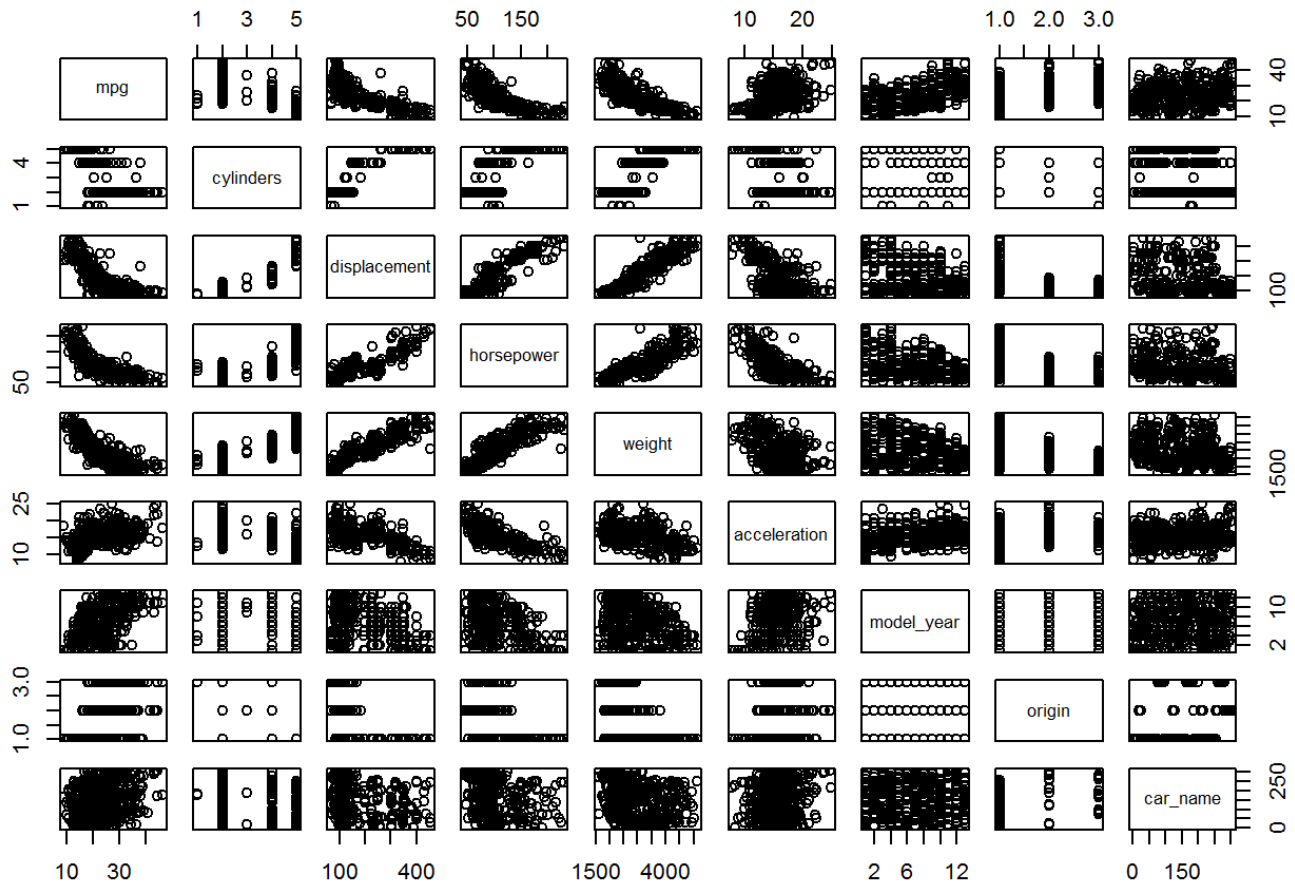
Standard deviation

```
apply(mpg_data[1:6],2,FUN=sd)
```

```
##          mpg    cylinders displacement   horsepower       weight
##     7.805007     1.705783   104.644004    38.491160   849.402560
## acceleration
##     2.758864
```

Plot pairs of variables:

```
pairs(mpg_data)
```



From plot pairs we can see that, mpg is not highly co-related to cylinders, car_name and origin but very loosely co-related with model_year and acceleration

##3.) Build a linear regression model: ###First model containing all independent variables but car_name:

```
lm_model <- lm(mpg~.-car_name,data=mpg_data)
summary(lm_model)
```

```
##
## Call:
## lm(formula = mpg ~ . - car_name, data = mpg_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9267 -1.6678 -0.0506  1.4493 11.6002
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.9168415  2.3608985  13.095  < 2e-16 ***
## cylinders4    6.9399216  1.5365961   4.516 8.48e-06 ***
## cylinders5    6.6377310  2.3372687   2.840 0.004762 **
## cylinders6    4.2973139  1.7057848   2.519 0.012182 *
## cylinders8    6.3668129  1.9687277   3.234 0.001331 **
## displacement  0.0118246  0.0067755   1.745 0.081785 .
## horsepower   -0.0392323  0.0130356  -3.010 0.002795 **
## weight       -0.0051802  0.0006241  -8.300 1.99e-15 ***
## acceleration  0.0036080  0.0868925   0.042 0.966902
## model_year71  0.9104285  0.8155744   1.116 0.265019
## model_year72 -0.4903062  0.8038193  -0.610 0.542257
## model_year73 -0.5528934  0.7214463  -0.766 0.443947
## model_year74  1.2419976  0.8547434   1.453 0.147056
## model_year75  0.8704016  0.8374036   1.039 0.299297
## model_year76  1.4966598  0.8019080   1.866 0.062782 .
## model_year77  2.9986967  0.8198949   3.657 0.000292 ***
## model_year78  2.9737783  0.7792185   3.816 0.000159 ***
## model_year79  4.8961763  0.8248124   5.936 6.74e-09 ***
## model_year80  9.0589316  0.8751948  10.351  < 2e-16 ***
## model_year81  6.4581580  0.8637018   7.477 5.58e-13 ***
## model_year82  7.8375850  0.8493560   9.228  < 2e-16 ***
## origin2       1.6932853  0.5162117   3.280 0.001136 **
## origin3       2.2929268  0.4967645   4.616 5.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.848 on 369 degrees of freedom
## Multiple R-squared:  0.8744, Adjusted R-squared:  0.8669
## F-statistic: 116.8 on 22 and 369 DF,  p-value: < 2.2e-16
```

F-statistics, R-squared error, p-value:

```
summary(lm_model)$fstatistic
```

```
##    value    numdf    dendf
## 116.7504  22.0000 369.0000
```

```
summary(lm_model)$r.squared
```

```
## [1] 0.8743834
```

Removing car_name, cylinders, origin from linear model

```
lm_model <- lm(mpg~displacement+horsepower+weight+acceleration+model_year,data=mpg_data)
summary(lm_model)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + horsepower + weight + acceleration +
##     model_year, data = mpg_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6993 -2.1482  0.0061  1.9405 13.0119
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  38.4405431  2.0371982  18.869  < 2e-16 ***
## displacement  0.0019671  0.0051575   0.381 0.703120
## horsepower   -0.0074477  0.0136458  -0.546 0.585537
## weight       -0.0062285  0.0006564  -9.489  < 2e-16 ***
## acceleration  0.0620858  0.0950781   0.653 0.514158
## model_year71  0.9917153  0.8982323   1.104 0.270269
## model_year72 -0.0337674  0.8791339  -0.038 0.969381
## model_year73 -0.4673662  0.7971620  -0.586 0.558035
## model_year74  1.6164929  0.9400415   1.720 0.086331 .
## model_year75  0.9647018  0.9215532   1.047 0.295856
## model_year76  1.6831300  0.8851237   1.902 0.057993 .
## model_year77  3.0511586  0.8997363   3.391 0.000770 ***
## model_year78  2.8567085  0.8554664   3.339 0.000924 ***
## model_year79  5.0480920  0.9049701   5.578 4.66e-08 ***
## model_year80  9.8627003  0.9508384  10.373  < 2e-16 ***
## model_year81  6.8256754  0.9371649   7.283 1.93e-12 ***
## model_year82  8.0502748  0.9099990   8.846  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.159 on 375 degrees of freedom
## Multiple R-squared:  0.8429, Adjusted R-squared:  0.8362
## F-statistic: 125.7 on 16 and 375 DF,  p-value: < 2.2e-16
```

F-statistics, R-squared error, p-value:

```
summary(lm_model)$fstatistic
```

```
##    value    numdf    dendf
## 125.7446  16.0000 375.0000
```

```
summary(lm_model)$r.squared
```

```
## [1] 0.8428933
```

Removing acceleration from model:

```
lm_model <- lm(mpg~displacement+horsepower+weight+model_year,data=mpg_data)
summary(lm_model)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + horsepower + weight + model_year,
##     data = mpg_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8777 -2.1166  0.0489  1.9555 13.1972
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.5638302  1.0905364  36.279  < 2e-16 ***
## displacement  0.0013051  0.0050530   0.258 0.796332
## horsepower   -0.0131377  0.0104934  -1.252 0.211350
## weight       -0.0060186  0.0005719 -10.524  < 2e-16 ***
## model_year71  0.9141366  0.8896616   1.028 0.304841
## model_year72 -0.0693213  0.8767767  -0.079 0.937024
## model_year73 -0.5073537  0.7942000  -0.639 0.523327
## model_year74  1.5422821  0.9324346   1.654 0.098954 .
## model_year75  0.8821169  0.9121373   0.967 0.334122
## model_year76  1.6095423  0.8772507   1.835 0.067332 .
## model_year77  2.9815320  0.8927142   3.340 0.000922 ***
## model_year78  2.7989103  0.8502258   3.292 0.001089 **
## model_year79  4.9779025  0.8978783   5.544 5.57e-08 ***
## model_year80  9.8011674  0.9454356  10.367  < 2e-16 ***
## model_year81  6.7362807  0.9264046   7.271 2.08e-12 ***
## model_year82  7.9893880  0.9045187   8.833  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.157 on 376 degrees of freedom
## Multiple R-squared:  0.8427, Adjusted R-squared:  0.8364
## F-statistic: 134.3 on 15 and 376 DF,  p-value: < 2.2e-16
```

F-statistics, R-squared error, p-value:

```
summary(lm_model)$fstatistic
```

```
##   value   numdf   dendf
## 134.304  15.000 376.000
```

```
summary(lm_model)$r.squared
```

```
## [1] 0.8427147
```

Removing horsepower and displacement from model:

```
lm_model <- lm(mpg~weight+model_year,data=mpg_data)
summary(lm_model)
```

```
##
## Call:
## lm(formula = mpg ~ weight + model_year, data = mpg_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2409  -2.0409   0.0044   1.9897  13.4664
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.1251645  0.8994586  43.499  < 2e-16 ***
## weight       -0.0063554  0.0002023 -31.409  < 2e-16 ***
## model_year71  1.2466287  0.8465989   1.473 0.141714
## model_year72  0.1661482  0.8363701   0.199 0.842640
## model_year73 -0.2958322  0.7695462  -0.384 0.700880
## model_year74  1.9352031  0.8579845   2.256 0.024672 *
## model_year75  1.3313934  0.8225812   1.619 0.106377
## model_year76  2.0150140  0.7997316   2.520 0.012160 *
## model_year77  3.2992924  0.8393682   3.931 0.000101 ***
## model_year78  3.1239166  0.7940124   3.934 9.93e-05 ***
## model_year79  5.3859319  0.8310444   6.481 2.85e-10 ***
## model_year80 10.1958802  0.8645457  11.793  < 2e-16 ***
## model_year81  7.1408917  0.8531344   8.370 1.13e-15 ***
## model_year82  8.3449817  0.8432910   9.896  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.155 on 378 degrees of freedom
## Multiple R-squared:  0.842,  Adjusted R-squared:  0.8366
## F-statistic:   155 on 13 and 378 DF,  p-value: < 2.2e-16
```

F-statistics, R-squared error, p-value:

```
summary(lm_model)$fstatistic
```

```
##    value    numdf    dendf
## 154.9858  13.0000 378.0000
```

```
summary(lm_model)$r.squared
```

```
## [1] 0.8420271
```

What does the hypothesis testing (i.e. t-test results) tell you about the linear model coefficients?

What does R square of this model tell you?

Ans. We can see that f-statistics is getting increased significantly with removal of less significant independent variables. And R-squared is remaining almost constant. So model is getting better and better. Also p-value is much less than 0.05 in every model, so we reject the null hypothesis. Hence there is a significant relationship between the variables in the linear regression model.