151AHW1

Jiyoon Clover Jeong

9/5/2017

Question 4

```
auto <- read.table("/Users/cloverjiyoon/2017Fall/Stat 151A/HW/HW1/auto-mpg.data.txt")

colnames(auto) <- c("mpg","cylinders","displacement",
                    "horsepower","weight","acceleration","modelyear", "origin","carname")

auto$horsepower <- as.numeric(levels(auto$horsepower))[auto$horsepower]
```

```
## Warning: NAs introduced by coercion
```

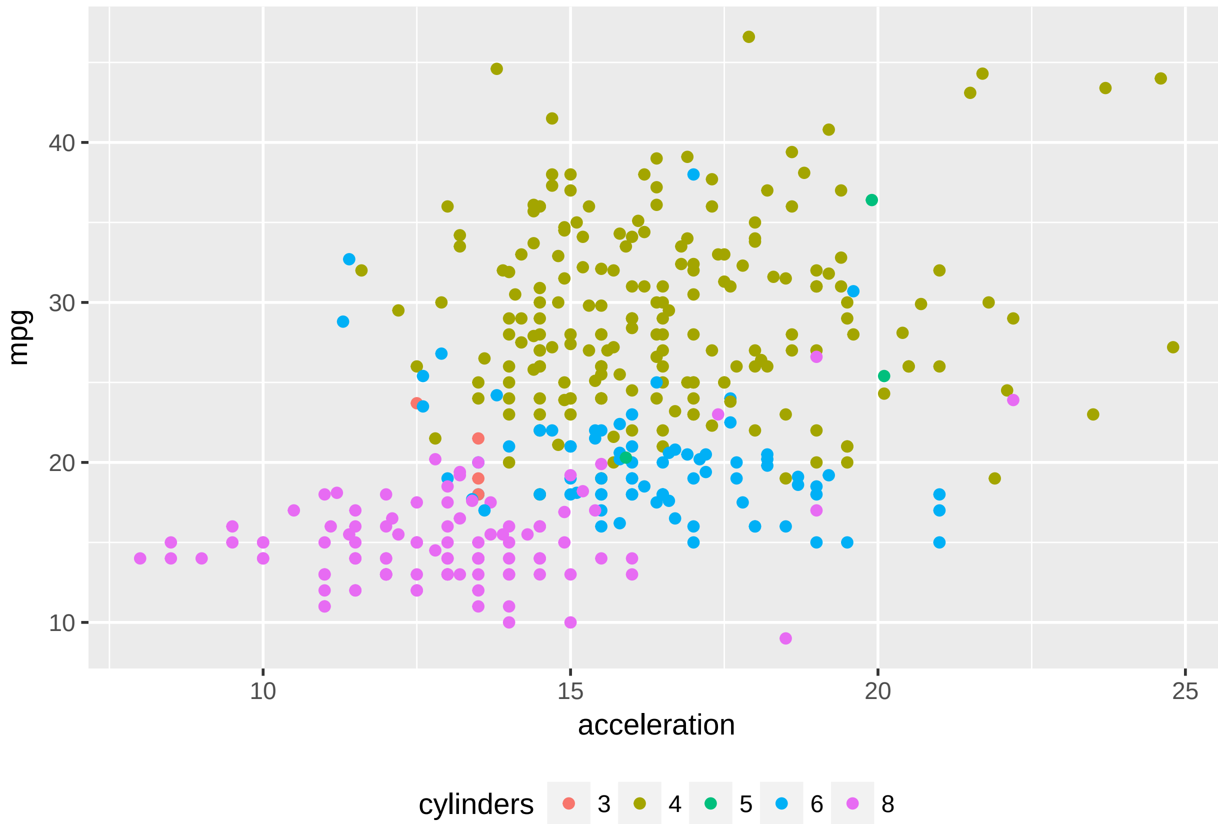
```
auto$cylinders <- factor(auto$cylinders)
auto <- na.omit(auto)

head(auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration modelyear
## 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                   carname
## 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
```

Part (a) - EDA

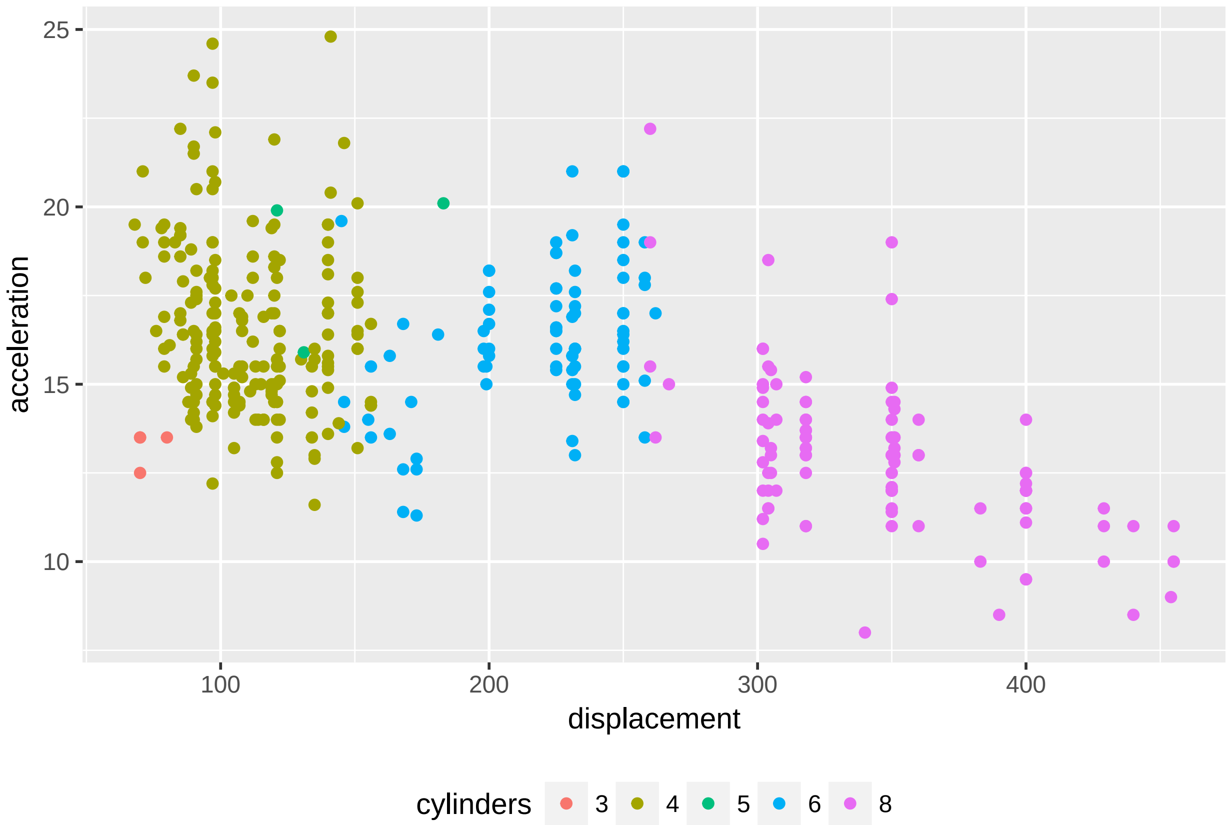
Scatterplot

```
ggplot(data=auto, aes(x=acceleration, y=mpg)) +
  geom_point(aes(color=cylinders)) + theme(legend.position="bottom")
```



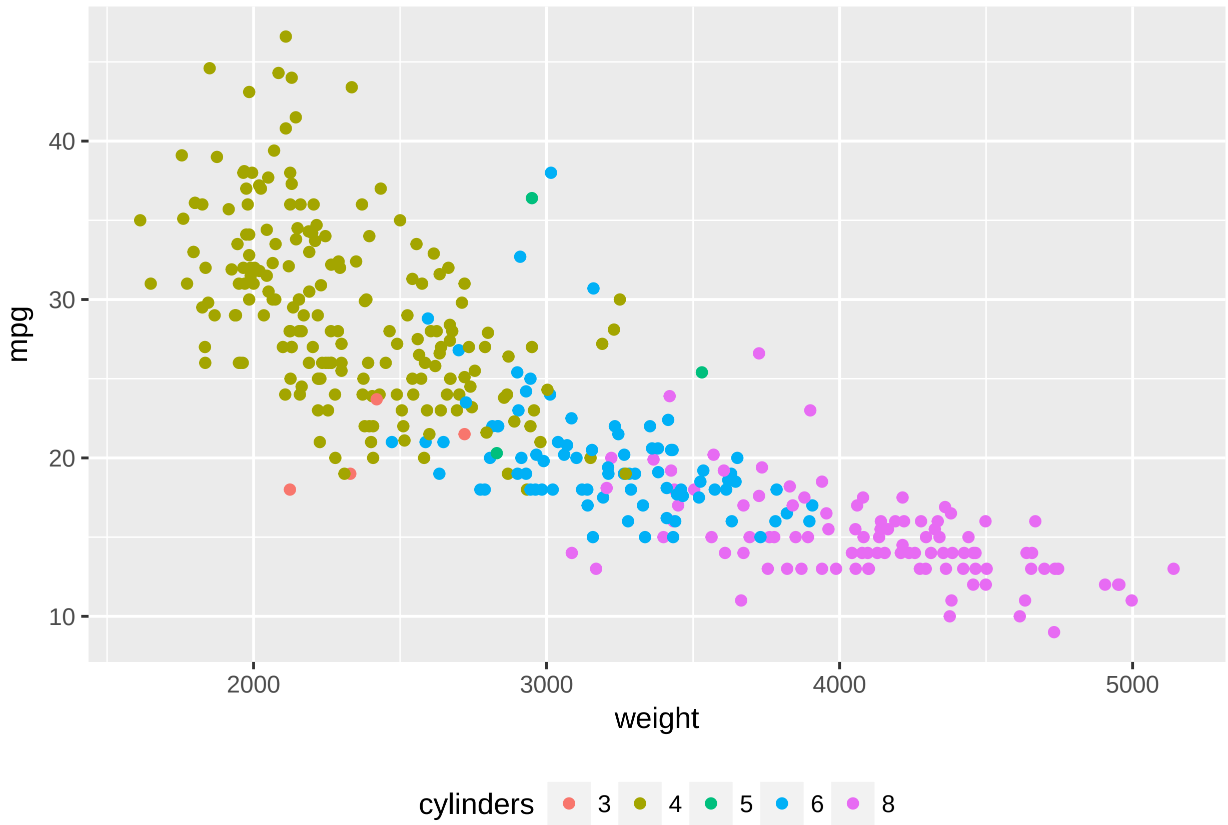
This graph shows that the acceleration variable and mpg variable are roughly proportional to each other. Also, the number of cylinders are high when mpg and acceleration are small and low when acceleration and mpg are high.

```
ggplot(data=auto, aes(x=displacement, y=acceleration)) +
  geom_point(aes(color=cylinders)) + theme(legend.position="bottom")
```



We can see that acceleration and displacement are inversely proportional to each other. Also, cylinder of car increases as displacement increases.

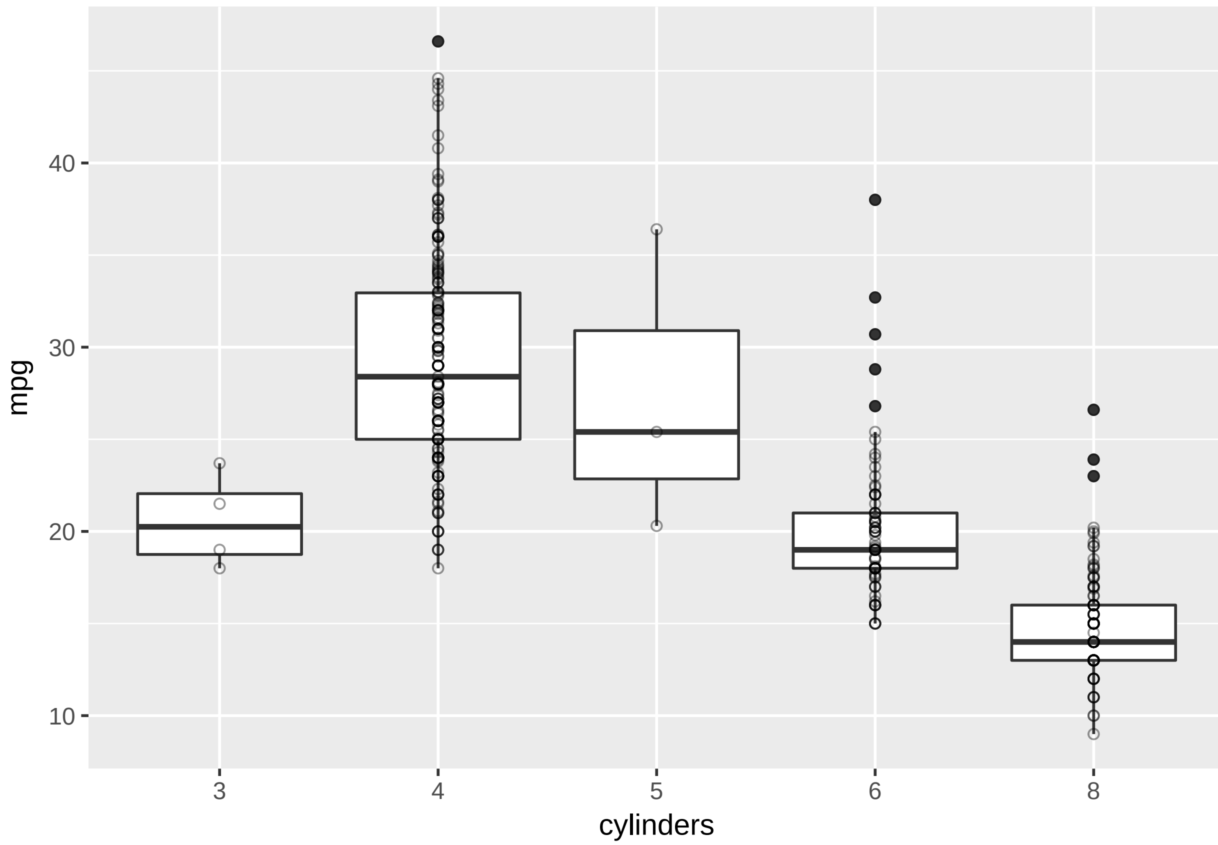
```
ggplot(data=auto, aes(x=weight, y=mpg, na.rm = T)) +
  geom_point(aes(color=cylinders)) + theme(legend.position="bottom")
```



Mpg and weight are inversely proportional to each other. Also, cylinders of cars increase as the weight of car increases

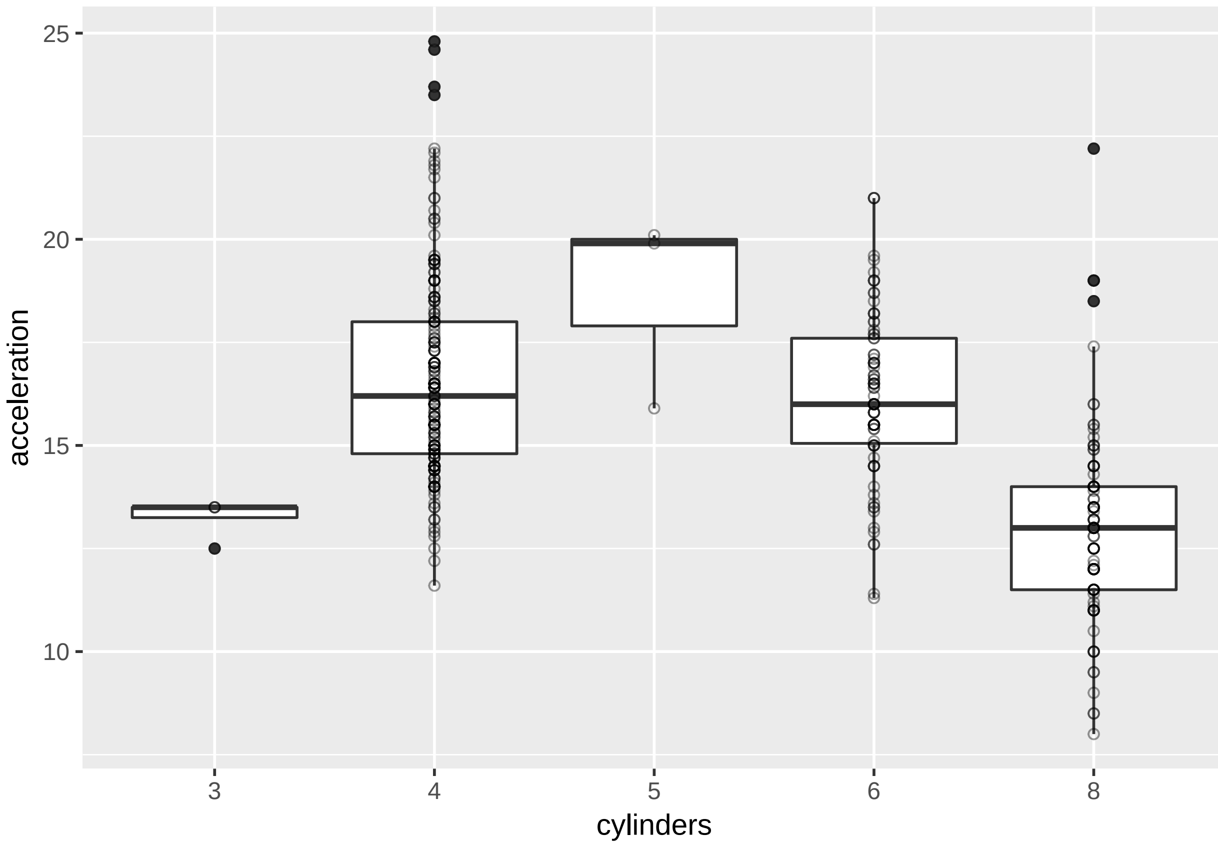
Boxplots

```
ggplot(auto, aes(x=factor(cylinders), y=mpg)) + geom_boxplot() +
  geom_point(shape=1, alpha = 0.4) + labs(x="cylinders", y="mpg")
```



The cars which have 4 cylinders more likely to have higher mpg than other cars which have 3,5,6,and 8 cylinders.

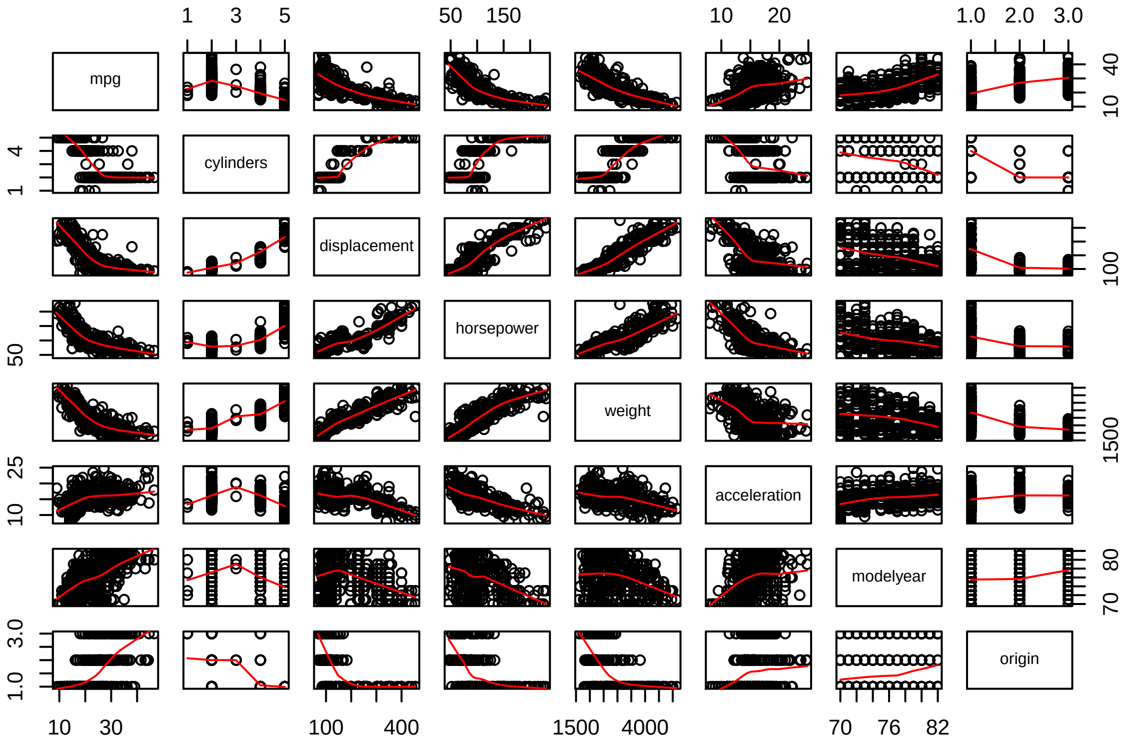
```r
ggplot(auto, aes(x=factor(cylinders), y=acceleration)) + geom_boxplot()+
  geom_point(shape=1, alpha = 0.4) + labs(x="cylinders", y="acceleration")
```



The cars which have 5 cylinders more likely to have higher acceleration than other cars which have 3,4,6,and 8 cylinders. Interestingly, acceleration doesn't always increases as the number of cylinders in car increases, but it decreases when the number of cylinders in car exceeds 5.

Pairs plot with smoothing lines

```
pairs(auto[,c(1:8),], panel = panel.smooth)
```

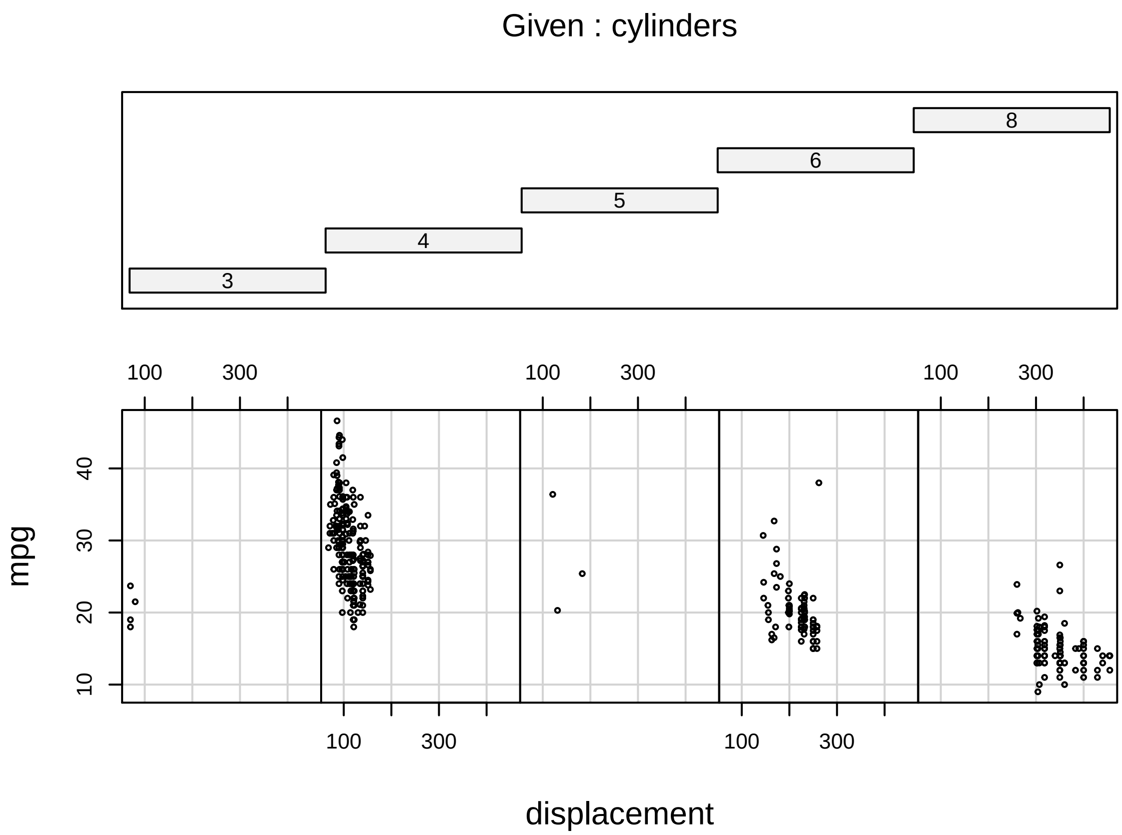



```
#ommit car_name variable since they don't have valuable information
```

We can see that mpg is proportional to acceleration and model year but inversely proportional to displacement, horsepower, and weight from the first row. Origin is weekly proportional to mpg also. Also, the number of cylinders is proportional to displacement, horsepower, and weight but inversely proportional to acceleration, model year, and origin. The smoothing lines in graph let us to see general trends and relationships between two variables.

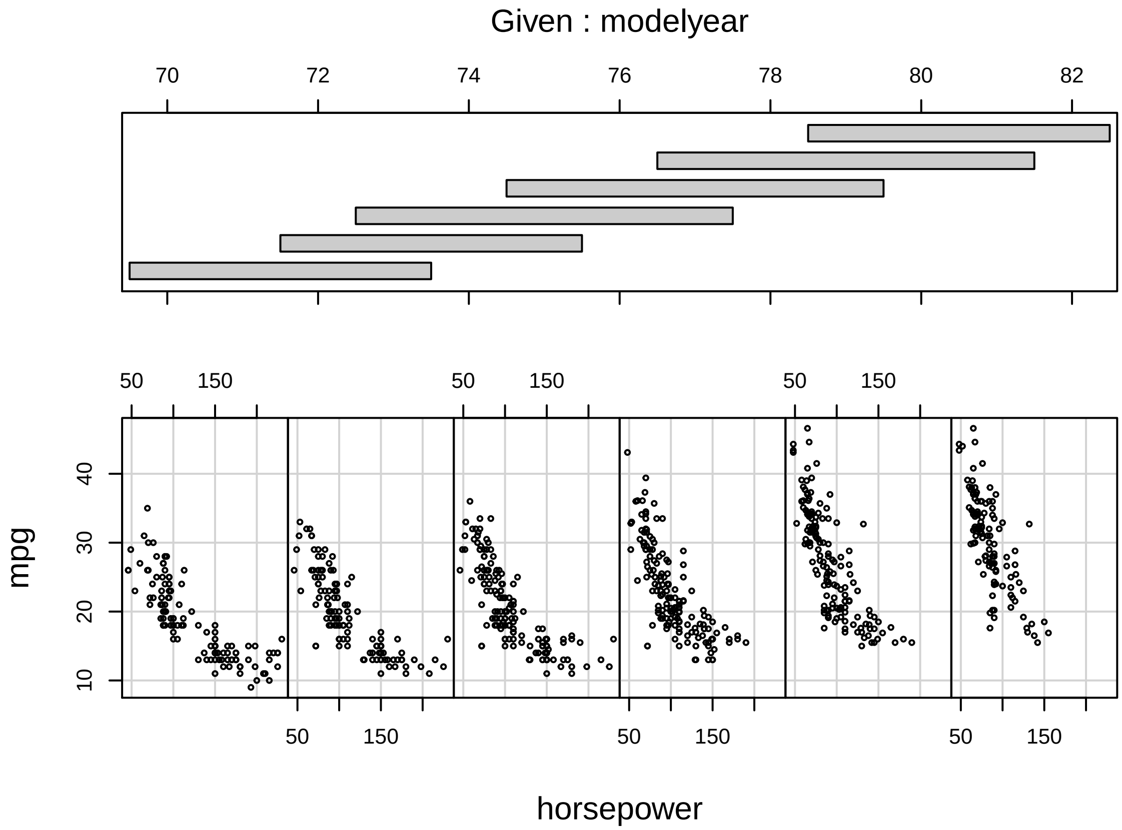
Co-plot

```
coplot(mpg ~ displacement | cylinders, data = auto, cex = 0.5, columns = 5)
```



As the number of cylinders increases, mpg decreases but displacement increases.

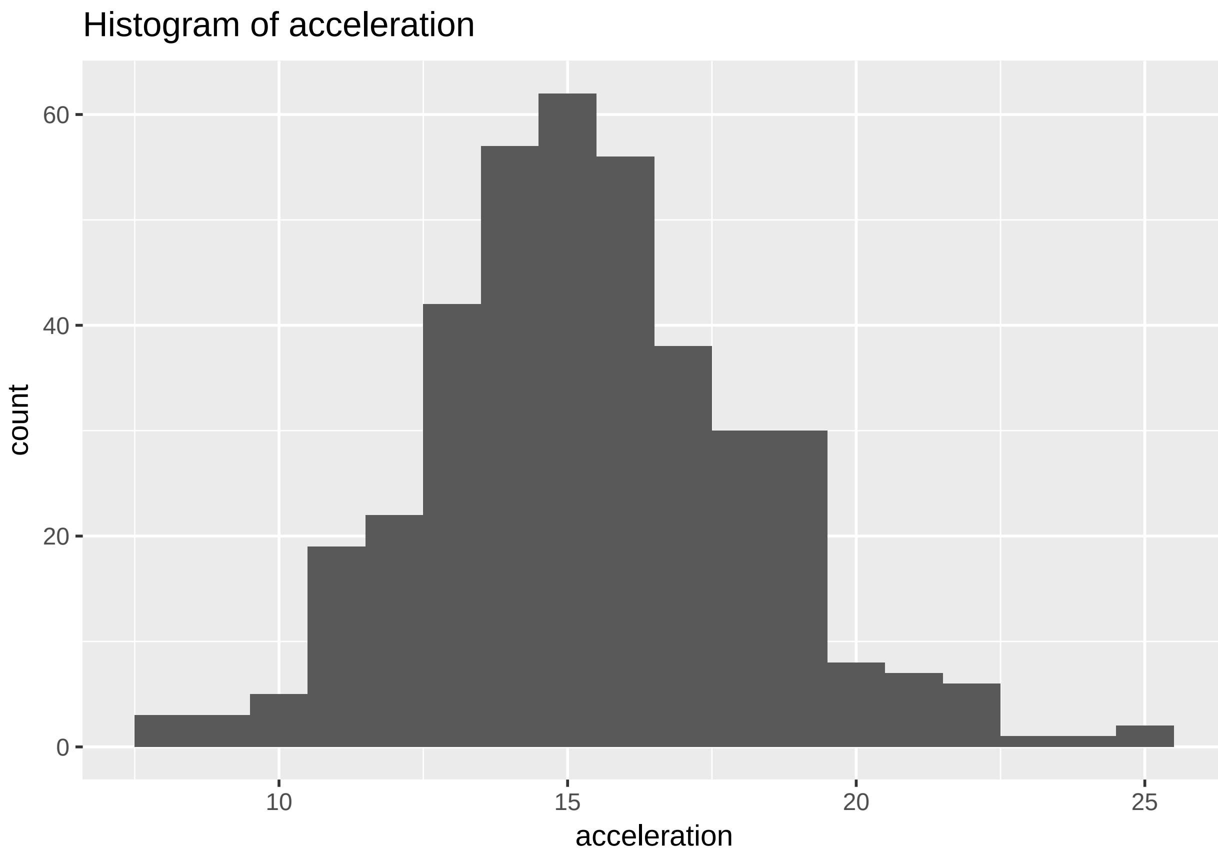
```
coplot(mpg ~ horsepower | modelyear, data = auto, cex = 0.5, xlab = "horsepower", column = 6)
```



As modelyear increases, mpg increases and horsepower decreases.

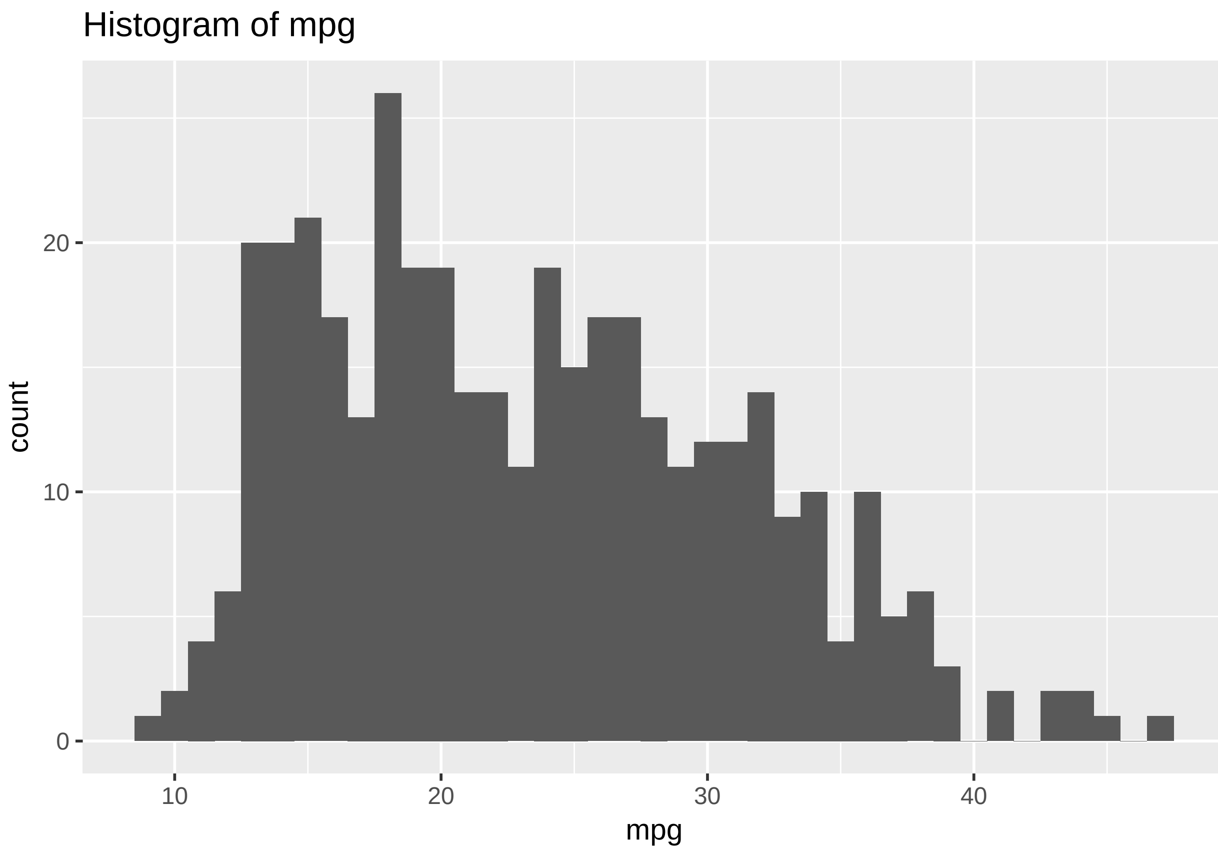
Density estimators plot (Histogram)

```
ggplot(data=auto, aes(x= acceleration)) + geom_histogram(binwidth = 1)+
  labs(x="acceleration", title="Histogram of acceleration")
```



The histrogram(density plot) of acceleration variable is similar to bell shape as normal distribution. The most frequent acceleration of the cars in the given dataset is approximately 15.

```
ggplot(data=auto, aes(x= mpg)) + geom_histogram(binwidth = 1) +
  labs(x="mpg", title="Histogram of mpg")
```



Most of the car's mpg is between 13 to 25. The histogram(density plot) of mpg variable is right-skewed.

Part(b)

```
fac.cyl <- as.factor(auto$cylinders)
cyl.mat <- sapply(levels(fac.cyl), function(x) as.integer(x == auto$cylinders))

cyl.mat <- cyl.mat[,-1]

fac.modelyear <- as.factor(auto$modelyear)
year.mat <- sapply(levels(fac.modelyear), function(x) as.integer(x == auto$modelyear))

year.mat <- year.mat[,-1]

fac.origin <- as.factor(auto$origin)
origin.mat <- sapply(levels(fac.origin), function(x) as.integer(x == auto$origin))

origin.mat <- origin.mat[,-1]


X <- cbind(rep(1,392), cyl.mat, auto[3:6], year.mat, origin.mat)
colnames(X)[1] <- c("intercept")

head(X,3)
```

```
##   intercept 4 5 6 8 displacement horsepower weight acceleration 71 72 73
```

```
## 1          1 0 0 0 1          307        130   3504          12.0  0   0   0
## 2          1 0 0 0 1          350        165   3693          11.5  0   0   0
## 3          1 0 0 0 1          318        150   3436          11.0  0   0   0
##   74 75 76 77 78 79 80 81 82 2 3
## 1  0  0  0  0  0  0  0  0  0 0 0
## 2  0  0  0  0  0  0  0  0  0 0 0
## 3  0  0  0  0  0  0  0  0  0 0 0
```

```r
y = as.matrix(auto$mpg)

X <- as.matrix(X)
betahat <- solve((t(X) %*% X)) %*% t(X) %*% as.matrix(auto$mpg)


ols <- function(X, y, betahat){

  X <- as.matrix(X)

  #inverse for solve
  #solve(A, b)  Returns vector x in the equation b = Ax (i.e., A-1b)

  SSres <-sum((y - (X %*% betahat))^2)    # (X %*% betahat) is y^hat

  SSreg <- sum(((X %*% betahat) - mean(y))^2)

  SStotal <- sum((y-mean(y))^2)

  Rsq <- SSreg/SStotal

  output <- list("coefficients" = betahat, "SSres" = SSres, "SSreg" = SSreg,
                "Rsq" = Rsq, "SStotal" = SStotal)
  return(output)

}




list <- ols(X, y, betahat)

auto$cylinders <- as.factor(auto$cylinders)
auto$origin <- as.factor(auto$origin)
auto$modelyear <- as.factor(auto$modelyear)

fit2 <- lm(mpg~ . -carname, data = auto)
summary(fit2)
```

```
##
## Call:
## lm(formula = mpg ~ . - carname, data = auto)
##
## 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
## modelyear71   0.9104285  0.8155744   1.116 0.265019
## modelyear72  -0.4903062  0.8038193  -0.610 0.542257
## modelyear73  -0.5528934  0.7214463  -0.766 0.443947
## modelyear74   1.2419976  0.8547434   1.453 0.147056
## modelyear75   0.8704016  0.8374036   1.039 0.299297
## modelyear76   1.4966598  0.8019080   1.866 0.062782 .
## modelyear77   2.9986967  0.8198949   3.657 0.000292 ***
## modelyear78   2.9737783  0.7792185   3.816 0.000159 ***
## modelyear79   4.8961763  0.8248124   5.936 6.74e-09 ***
## modelyear80   9.0589316  0.8751948  10.351  < 2e-16 ***
## modelyear81   6.4581580  0.8637018   7.477 5.58e-13 ***
## modelyear82   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
```

```
cat("The coefficient estimates (betahat) is ")
```

```
## The coefficient estimates (betahat) is
```

```
list$coefficient
```

```
##                      [,1]
## intercept    30.916841489
## 4             6.939921560
## 5             6.637730992
## 6             4.297313906
## 8             6.366812930
## displacement  0.011824592
## horsepower   -0.039232282
## weight       -0.005180179
## acceleration  0.003607983
## 71            0.910428513
## 72           -0.490306154
## 73           -0.552893391
## 74            1.241997594
## 75            0.870401578
## 76            1.496659785
## 77            2.998696745
## 78            2.973778349
```

```
## 79            4.896176328
## 80            9.058931568
## 81            6.458158033
## 82            7.837584958
## 2             1.693285334
## 3             2.292926778
```

```r
cat("residual sum of squares is")
```

```
## residual sum of squares is
```

```r
list$SSres
```

```
## [1] 2992.061
```

```r
cat("SSreg is")
```

```
## SSreg is
```

```r
list$SSreg
```

```
## [1] 20826.93
```

```r
cat("SStotal is")
```

```
## SStotal is
```

```r
list$SStotal
```

```
## [1] 23818.99
```

```r
cat("R^2 is")
```

```
## R^2 is
```

```r
list$Rsq
```

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

Part (c)

```r
fitted <- as.matrix(X) %*% betahat
head(fitted)
```

```
##       [,1]
## 1 17.70555
## 2 15.86002
## 3 17.39962
## 4 17.25323
## 5 17.53361
## 6 12.13733
```

```r
residuals <- y - fitted
head(residuals)
```

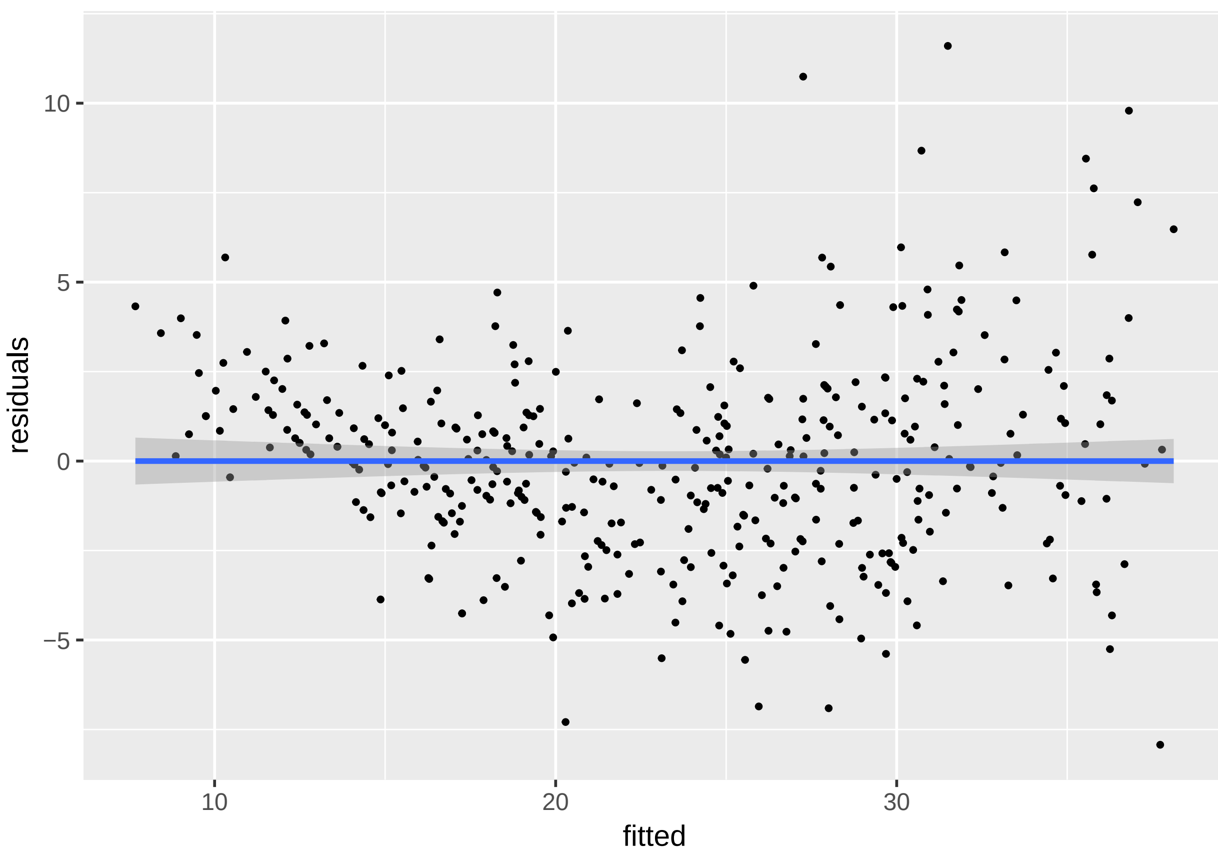
```
##         [,1]
## 1  0.2944452
## 2 -0.8600245
## 3  0.6003761
## 4 -1.2532282
```

```
## 5 -0.5336069
## 6  2.8626662
```

```r
data <- as.data.frame(cbind(fitted, residuals))
head(data)
```

```
##         V1         V2
## 1 17.70555  0.2944452
## 2 15.86002 -0.8600245
## 3 17.39962  0.6003761
## 4 17.25323 -1.2532282
## 5 17.53361 -0.5336069
## 6 12.13733  2.8626662
```

```r
ggplot(data, aes(y = residuals, x= fitted)) +
  geom_point(size = 0.7) +  geom_smooth(method='lm', formula=y~x)
```



since R^2 is 0.8743834, it is close to 1. Since R^2 is close to 1, it means that the regresssion model that I chose is more accurate than the small model.

In residual versus pitted plot from regression, there are no obvious outliers. However, we can see that points in the residuals versus fitted plot have slight quadratic/linear patterns.

Part (d) What can you conclude from your overall analysis?

As I stated in part(c), the residuals exhibit slight quadratic/linear shape, and this possibly means that the there is a better model than linear model for the relationship between reponse variables and explanatory variables. This fact might also suggests that transformations of the variables and/or interaction terms may be a more appropriate fit.

In addition to the previous statement, another trend of residuals is that they are vertically more spread out as fitted value increases. This might suggest that the model which fits better than linear model(true model) reveals more variability when fitted value is larger.

After I did some research about residuals versus fitted plot, I found that the residuals from this plot are heteroscedastically distributed. In another words, the ϵ in y = $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$ will have variance depending on the x_i, rather than having some constant variance σ^2.