HW3

Khanin

2/18/2022

0)

```
library(MASS)
library(ISLR2)
```

```
##
## Attaching package: 'ISLR2'
```

```
## The following object is masked from 'package:MASS':
##
##     Boston
```

```
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

```
# Single Linear regression
lm.fit =lm(medv~lstat, data=Boston) # fit lstat(X) with medv(Y)
lm.fit # Show basic information
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

```
summary(lm.fit) # Show all statistical information
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
names(lm.fit) # We can use the names() function in order to find out what other pieces of information a
```

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

```
coef(lm.fit) # Access coefficient
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

```
confint(lm.fit) # a confidence interval for the coefficient estimates
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

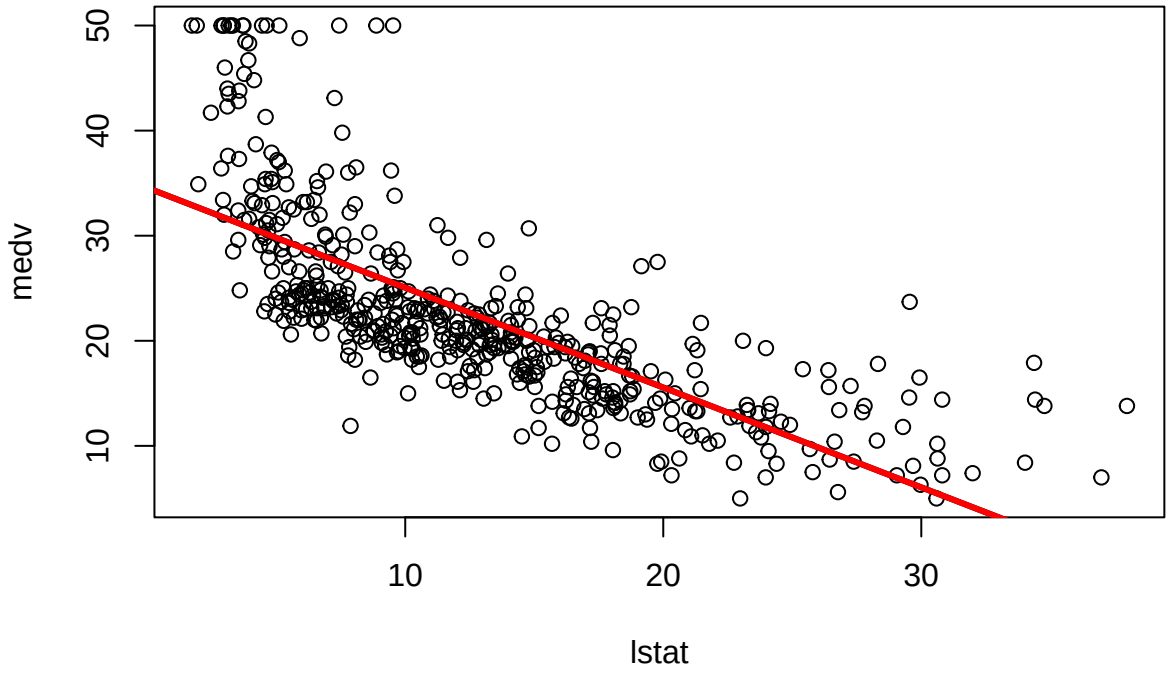
```
# Prediction intervals for the prediction of medv for a given value of lstat.
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

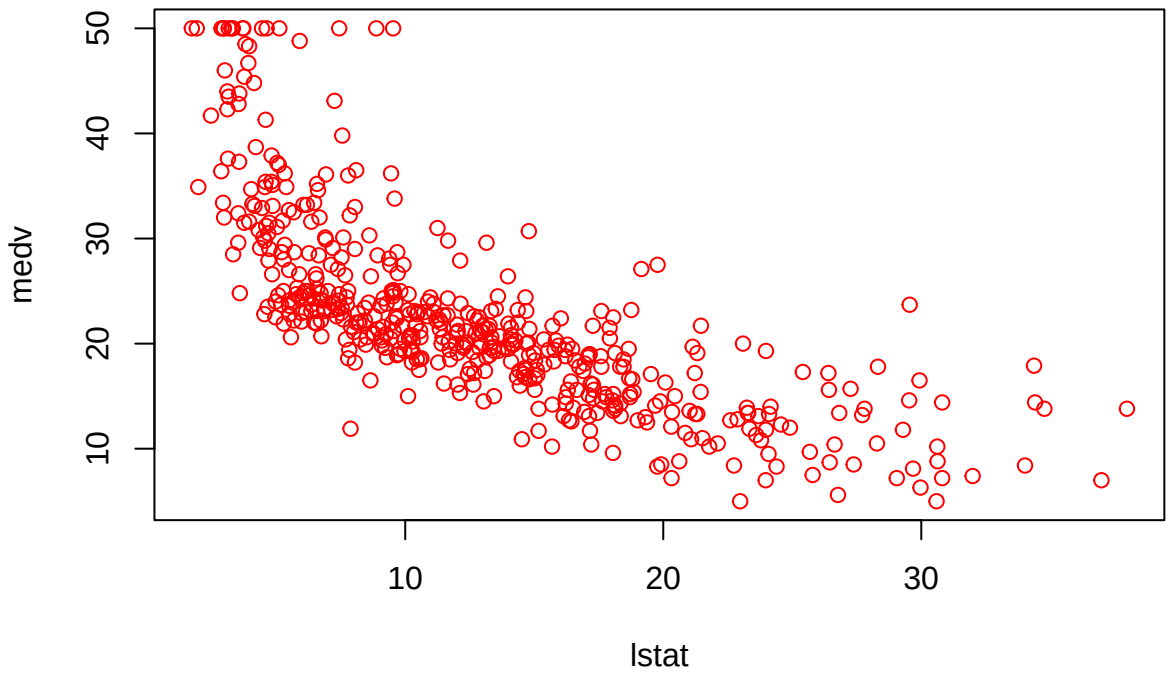
```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))),
        interval = "prediction")
```

```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

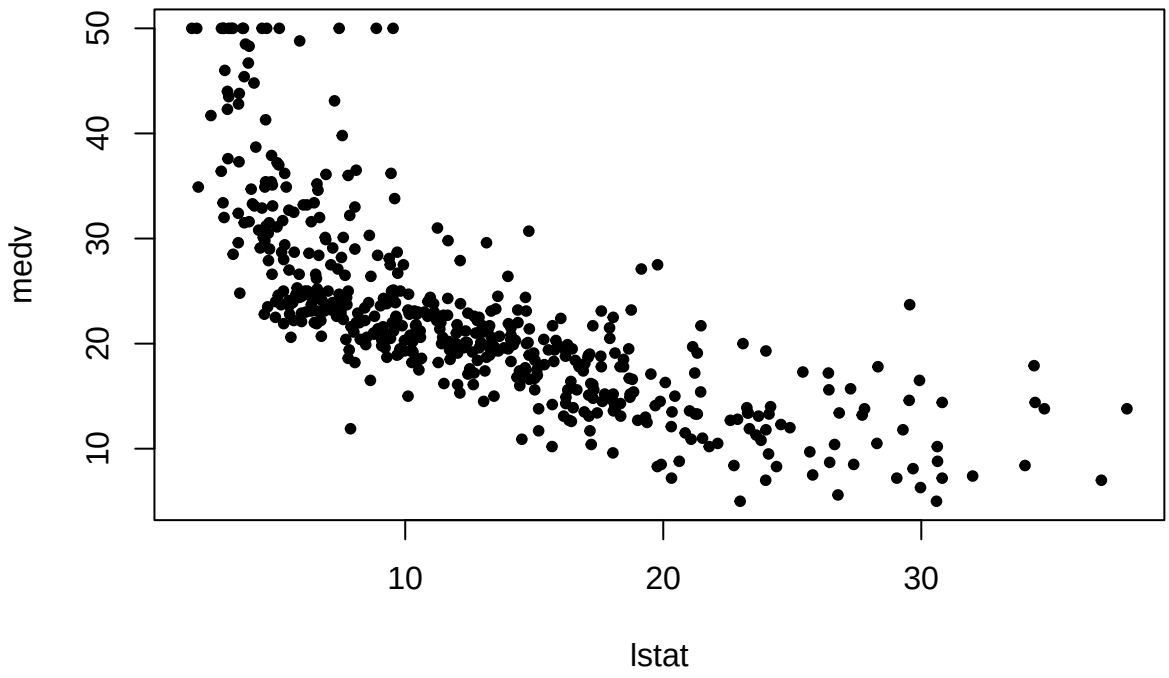
```
attach(Boston)

plot(lstat, medv)
abline(lm.fit) # Add a straight line
abline(lm.fit, lwd = 3) # Change the size of the straight line
abline(lm.fit, lwd = 3,col = 'red') # Change the color of the straight line to be red
```



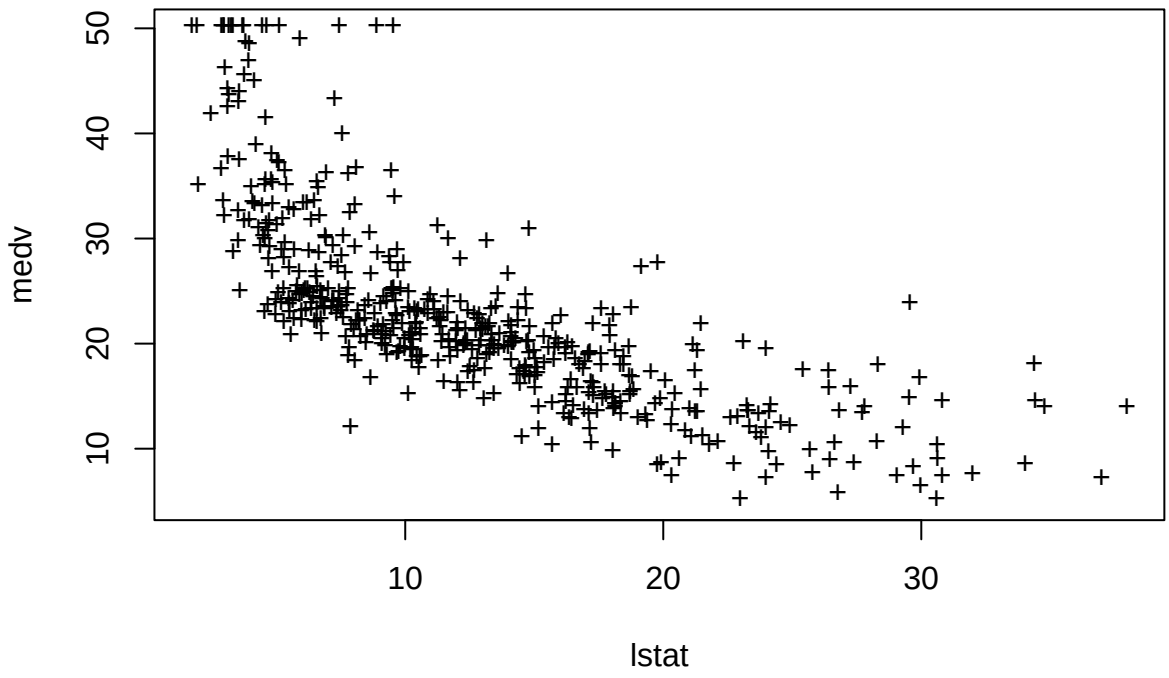

```
plot(lstat, medv, col = 'red') # Scatter plot with red color
```



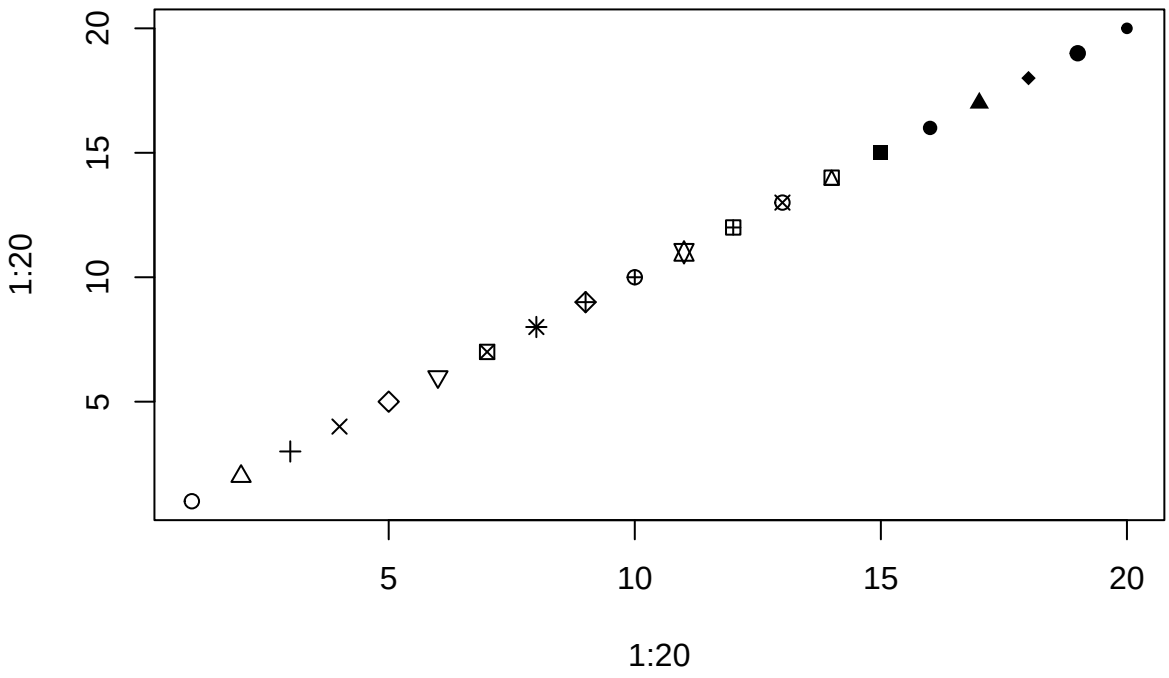

```
plot(lstat, medv, pch = 20) # Scatter plot with circles
```



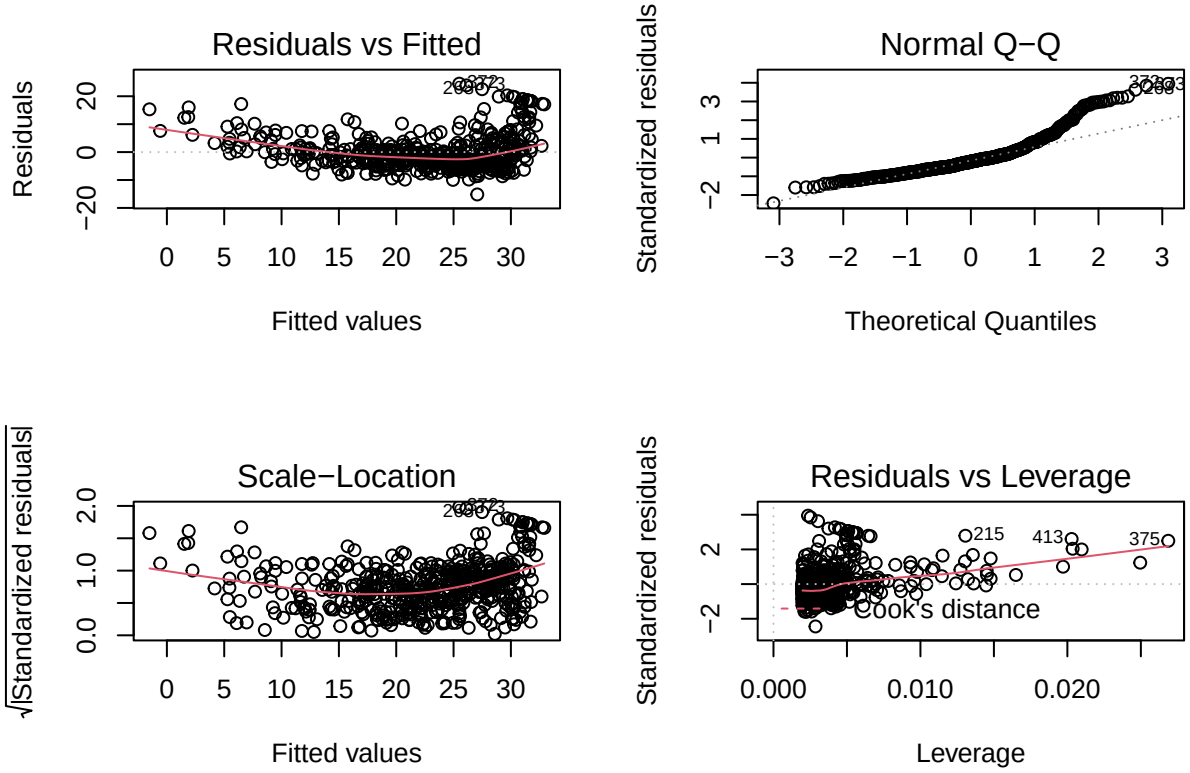

```
plot(lstat, medv, pch = '+') # Plot a '+' symbol
```




```
plot(1:20, 1:20, pch = 1:20) # Plot 20 symbols for 20 points
```




```
par(mfrow = c(2, 2)) # Divide plotting region into 2x2
plot(lm.fit)
```




```
plot(predict(lm.fit), residuals(lm.fit)) # compute the residuals from a linear
# regression fit using the residuals() function
plot(predict(lm.fit), rstudent(lm.fit)) #  The function rstudent() will
# return the studentized residuals.
plot(hatvalues(lm.fit)) # Leverage statistics can be computed for any number
# of predictors using the hatvalues() function.
which.max(hatvalues(lm.fit)) # it tells us which observation has the
```

```
## 375
## 375
```

```
# largest leverage statistic.


## Multiple linear regression
lm.fit =lm(medv~lstat + age, data=Boston) # Only lstat and age
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

```
lm.fit =lm(medv~., data=Boston) # Use all 12 variables
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1304  -2.7673  -0.5814   1.9414  26.2526
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
## crim         -0.121389   0.033000  -3.678 0.000261 ***
## zn            0.046963   0.013879   3.384 0.000772 ***
## indus         0.013468   0.062145   0.217 0.828520
## chas          2.839993   0.870007   3.264 0.001173 **
## nox         -18.758022   3.851355  -4.870 1.50e-06 ***
## rm            3.658119   0.420246   8.705  < 2e-16 ***
## age           0.003611   0.013329   0.271 0.786595
## dis          -1.490754   0.201623  -7.394 6.17e-13 ***
## rad           0.289405   0.066908   4.325 1.84e-05 ***
## tax          -0.012682   0.003801  -3.337 0.000912 ***
## ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
## lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.798 on 493 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7278
## F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
library(car)
```

```
## Loading required package: carData
```

```
vif(lm.fit)  ## Calculate VIF
```

```
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.767486 2.298459 3.987181 1.071168 4.369093 1.912532 3.088232 3.954037
##      rad      tax  ptratio    lstat
## 7.445301 9.002158 1.797060 2.870777
```

```
lm.fit1 <- lm(medv ~. - age, data = Boston) ## Exclude age from the model
summary(lm.fit1)
```

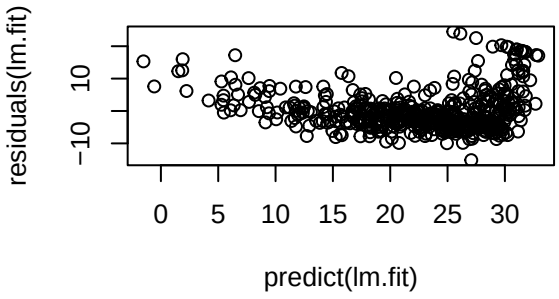
```
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1851  -2.7330  -0.6116   1.8555  26.3838
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.525128   4.919684   8.441 3.52e-16 ***
## crim         -0.121426   0.032969  -3.683 0.000256 ***
## zn            0.046512   0.013766   3.379 0.000785 ***
## indus         0.013451   0.062086   0.217 0.828577
## chas          2.852773   0.867912   3.287 0.001085 **
## nox         -18.485070   3.713714  -4.978 8.91e-07 ***
## rm            3.681070   0.411230   8.951  < 2e-16 ***
## dis          -1.506777   0.192570  -7.825 3.12e-14 ***
## rad           0.287940   0.066627   4.322 1.87e-05 ***
## tax          -0.012653   0.003796  -3.333 0.000923 ***
## ptratio      -0.934649   0.131653  -7.099 4.39e-12 ***
## lstat        -0.547409   0.047669 -11.483  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.794 on 494 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7284
## F-statistic: 124.1 on 11 and 494 DF,  p-value: < 2.2e-16
```

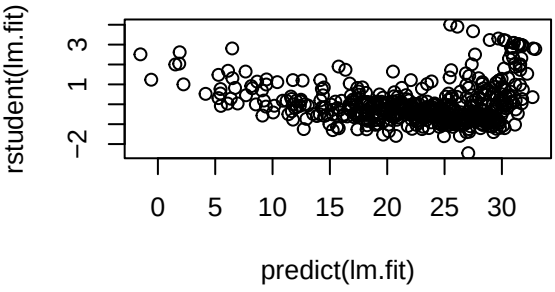
```
lm.fit1 <- update(lm.fit, ~. - age) # Alternatively, we can use update with
# the new variables.

summary(lm(medv ~ lstat * age, data = Boston)) ## Add an interaction term
```

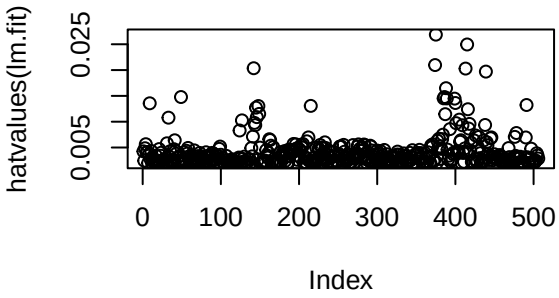
```
##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
```

```
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
# and summarize
```








```
## 3.6.7 Writing Functions
LoadLibraries = function() {
  library(ISLR2)
  library(MASS)
  print("The libraries have been loaded.")
}

LoadLibraries()
```

```
## [1] "The libraries have been loaded."
```

1)

a) Preliminary steps before creating any model

```
class(cars) # Show the class
```

```
## [1] "data.frame"
```

```
head(cars)  # Show top 6 rows
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

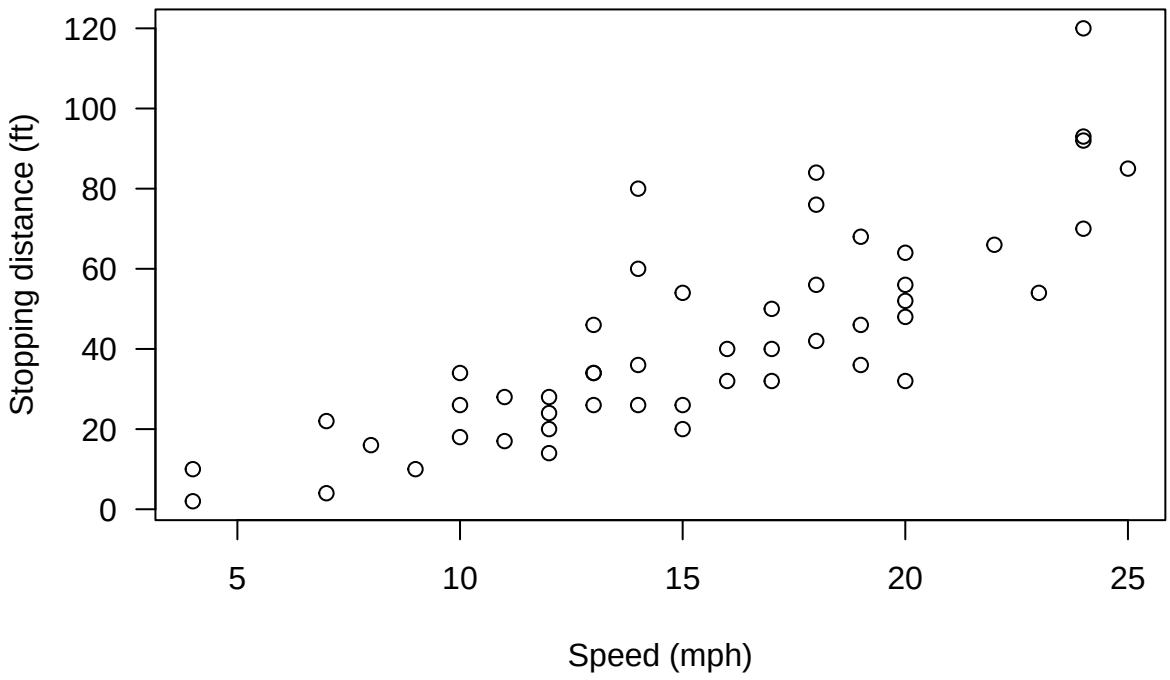
```
dim(cars)  # Dimensions
```

```
## [1] 50  2
```

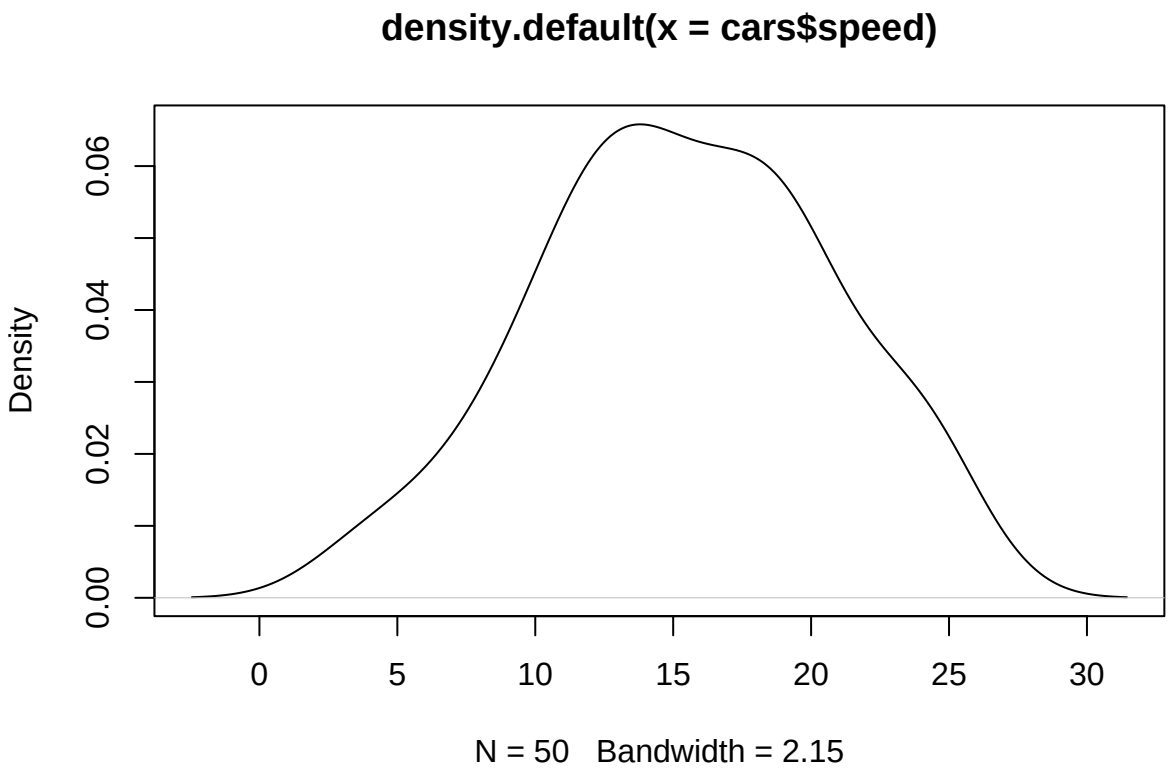
```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

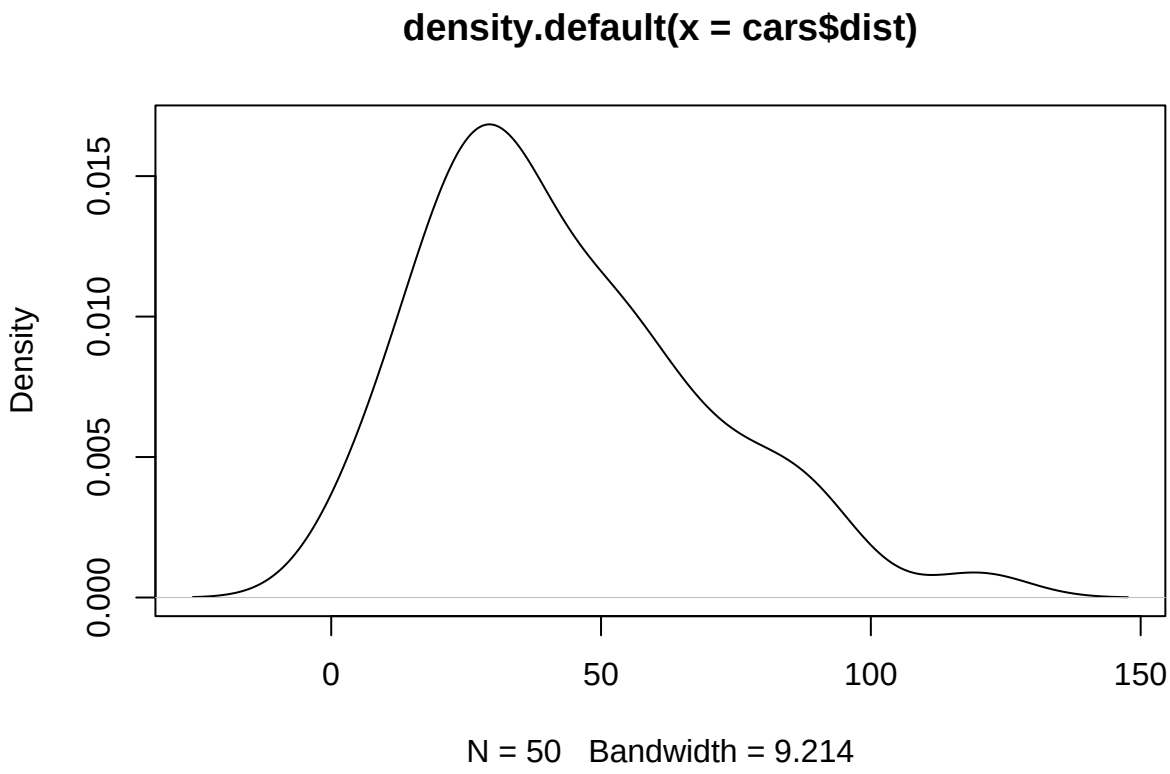
```
# Scatter plot
plot(cars, xlab = "Speed (mph)", ylab = "Stopping distance (ft)",
     las = 1)
```




```
# Speed (mph) and Stopping distance (ft) seem to have a linear relationship

## Plot histogram to understand distribution of the data
plot(density(cars$speed)) # Speed has a normal distribution with mean around 15 mph.
```




```
plot(density(cars$dist)) # Distance has a normal distribution with positive skewness.
```




```
## Calculate correlation for all variables in R to understand linear relationship
cor(cars)
```

```
##           speed      dist
## speed 1.0000000 0.8068949
## dist  0.8068949 1.0000000
```

```
# Speed and distance have a very high correlation with around 0.8, which
#indicates a strong linear relationship.
```

b)

```
knitr::include_graphics("/Users/khaninsi/Documents/USC Master/Spring 2022/ISE 529 Predictive Analytics/
```

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

Given that X_1 is speed in mph and Y is distance in ft

All variables in the above linear model live in unknowable world because these variables are population

parameters, which we cannot find explicitly. However, we want to estimate these variables based on the variables in the knowable world in the following equation. According to a), it is promising to get the accurate prediction of stopping distance because speed has a strong linear correlation with the dependent variable. Thus p-value should be much lower than 0.05, indicating that the linear relationship between independent and dependent variables are not by chance.

c)

```
knitr::include_graphics("/Users/khaninsi/Documents/USC Master/Spring 2022/ISE 529 Predictive Analytics/
```

$$\hat{y} = \widehat{\beta_0} + \widehat{\beta_1} x_1$$

Given that x_1 is speed in mph and y is distance in ft

The estimated model for linear regression that we will implement in the R programming. These variables live in the knowable world because we already have the value of dependent and independent variables, and thus we can estimate the beta hat.

```
lm.fit =lm(dist~speed, data=cars) # fit linear regression on speed(X) with dist(Y)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

According to the model, the intercept term does not make sense to interpret because it means the average stopping distance is -17.5 when the speed is equal to zero, which is not viable, but we need this term in order to make predictions.
The coefficients of speed is 3.9324, which would mean with every speed increases, the stopping distance would increase 3.9324 feet. Moreover, the p-value is strongly significant as it is much lower than 0.05 and

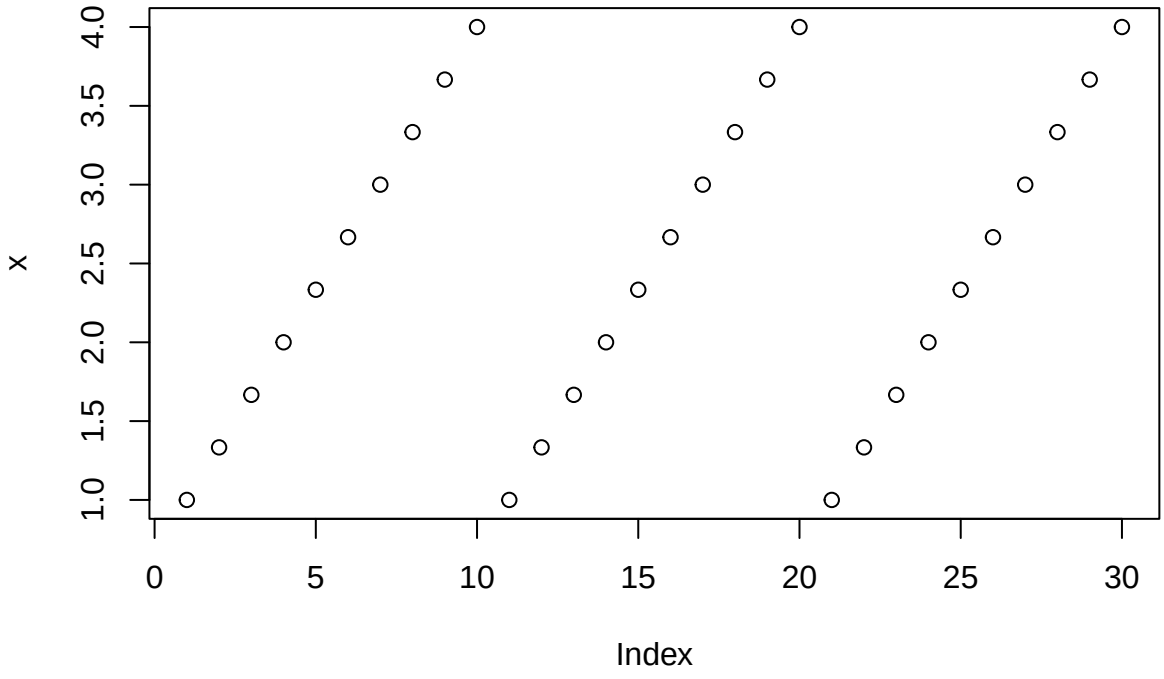
aligns with the high correlation value showed in section (a).
Finally, the F-statistics is significantly larger than 1 and its p-value is a lot less than 0.05. This indicates that the as least one beta is non-zero.

d)

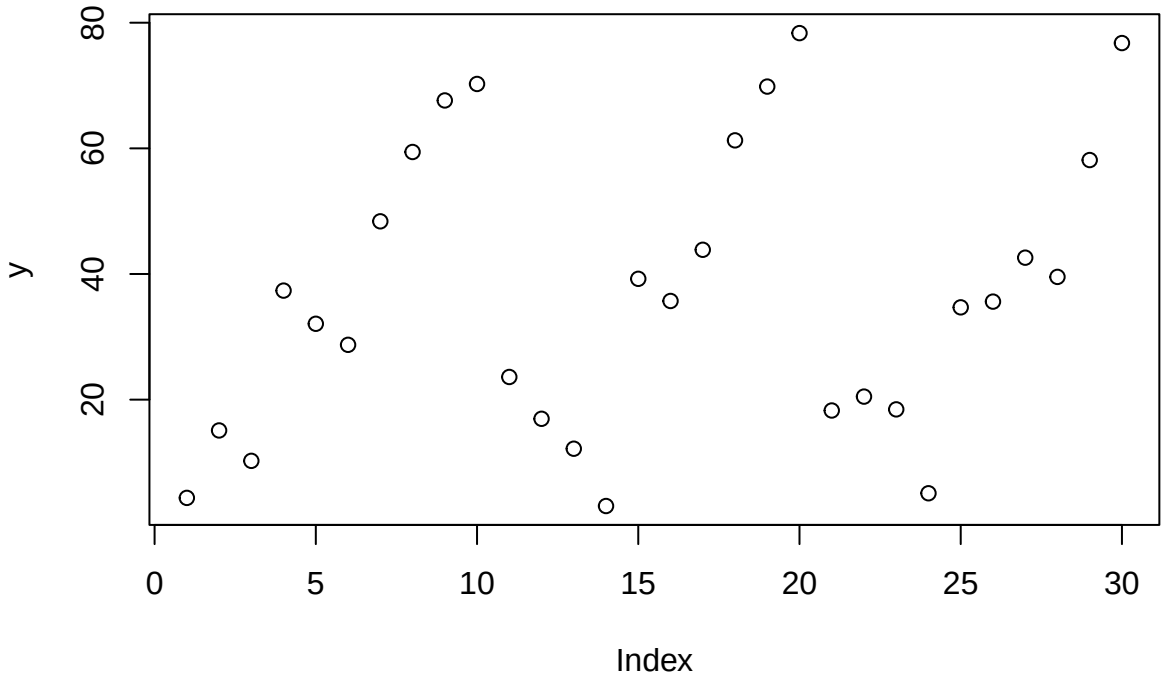
To assess the fit of the model, we will use two methods. The first method is R-squared. To interpret the R-squared in this model, the R-squared of 0.6511 means that the speed can explain 65.11% of total variance in the stopping distance, which is considerably high.
Another measurement is residual standard error (RSE), which estimates standard error of irreducible error. It means that even if the model were correct and the true values of the unknown coefficients beta 0 and beta 1 were known exactly, any prediction of stopping distance on the basis of speed would still be off by about 15.38 ft on average, which is still acceptable if comparing with the average stopping distance of 42.98.

2)

```
set.seed(1)
# Create a vector x in R that has 30 values going from 1 to 4 by 1/3 three times
x = rep(seq(1,length=10,by=1/3), times=3)
y = 5 + x + 4* x^2 # create a variable y as follows: 5 + x + 4*x^2
y = y + rnorm(30, sd = 9) # a vector of noise drawn from N(0,9) to y
plot(x)
```



```
plot(y)
```



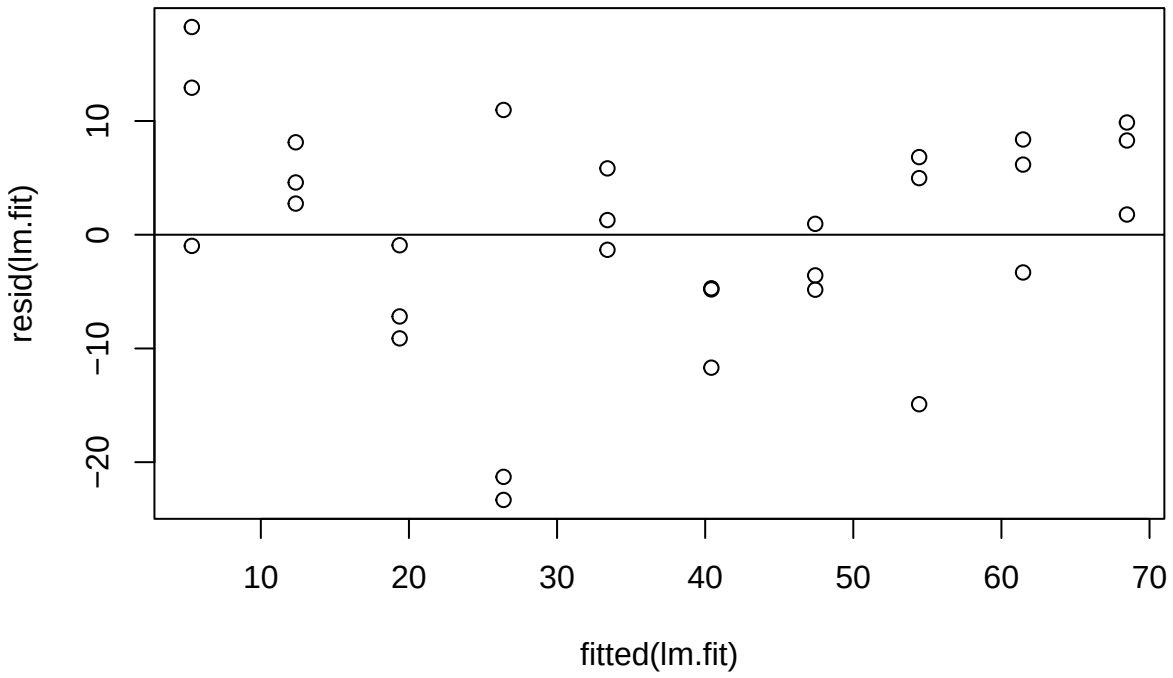
a) Fit a linear model predicting y from x. Interpret the coefficients and plot the residuals.

```
lm.fit =lm(y~x) # fit linear regression on y with x
summary(lm.fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.319  -4.785   1.122   6.662  18.263
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -15.701      5.002  -3.139  0.00397 **
## x             21.044      1.868  11.264 6.54e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.797 on 28 degrees of freedom
## Multiple R-squared:  0.8192, Adjusted R-squared:  0.8127
## F-statistic: 126.9 on 1 and 28 DF,  p-value: 6.535e-12
```

Based on the coefficients of x in the linear model, if x increases by 1, y would increase by 21.044, and the linear relationship between x and y does not occur by chance indicating by extremely low p-value. Moreover, the F-statistics also reinforces the t-value.

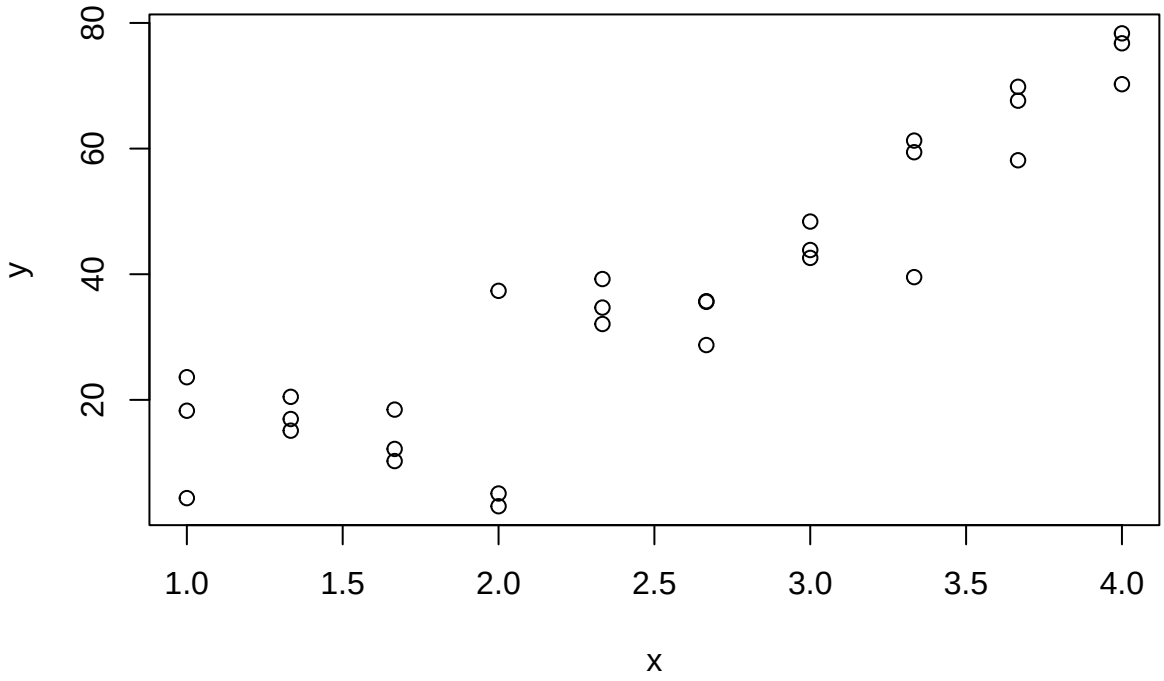
```
# Plot residuals
plot(fitted(lm.fit), resid(lm.fit))
abline(0,0)
```



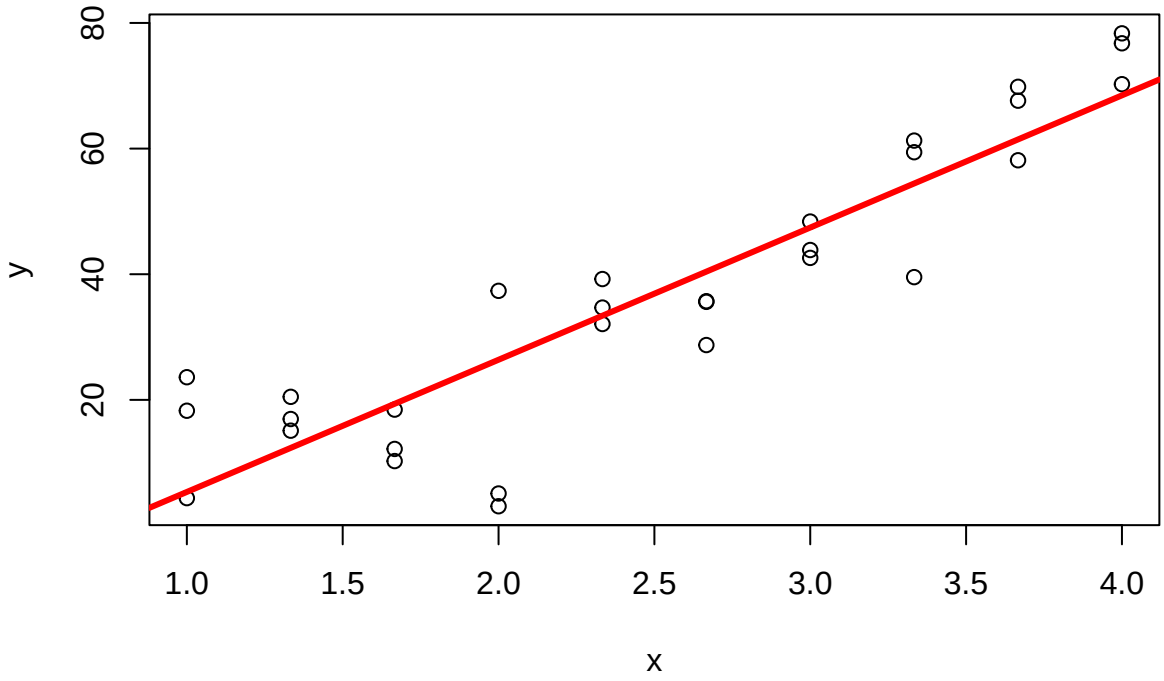
There is little pattern in the residuals, and the variance of residuals is constant. However, there is a potential outlier when the residual is 20.

b) Create a scatterplot of the data. Add the points predicted by your model in red

```
plot(x, y)
```




```
abline(lm.fit, lwd = 3,col = 'red')
```



We can see that the linear model can reasonably explain the dependent variable using an independent variable, and doing a good job explaining the variance, which follows the high r-squared of 0.81.

c) Given that you know the true underlying model in this case (in the unknowable world), how do you assess the linear model?

Given that I know the model in unknowable world, I would compare beta and beta hat, and measure the irreducible error term using MSE or RSE to assess the accuracy of model.

d) Now fit the same model as in a) but include a squared term e.g. x^2 as well as x. Interpret the coefficients and plot the residuals.

```
x2 = x^2
lm.fit2 =lm(y ~ x + x2) # fit linear regression on y with x and x^2
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = y ~ x + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.0806  -5.9716   0.8915   5.2741  15.2092
```

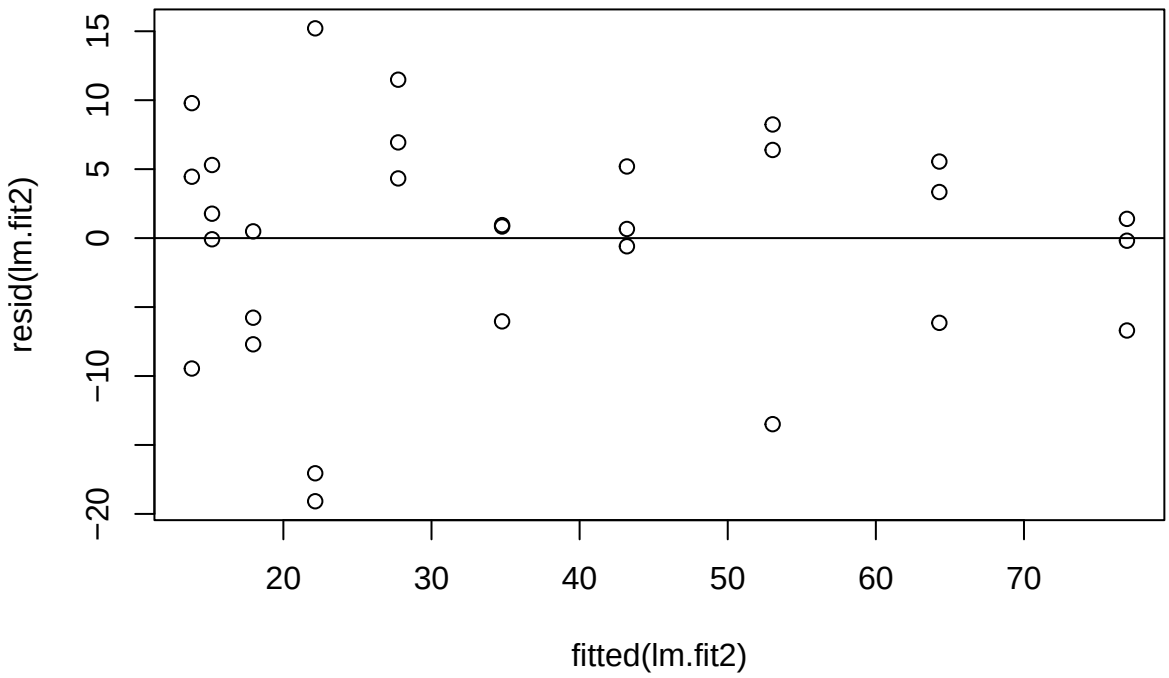
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.207     10.978   1.658  0.10881
## x            -10.745      9.612  -1.118  0.27348
## x2             6.358      1.896   3.354  0.00237 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.383 on 27 degrees of freedom
## Multiple R-squared:  0.8724, Adjusted R-squared:  0.8629
## F-statistic: 92.28 on 2 and 27 DF,  p-value: 8.513e-13
```

Based on the coefficients of x in the new linear model, if x increases by 1, y would decrease by 10.745. Similarly, if x2 increases by 1, y would increase by 6.358. However, the p-value of coefficients of intercept and x are all greater than 0.05, indicating that the linear relations of all variables with y occur by chance even though the F-statistics does not align with the t-statistics. With this information, I am of the opinion that the colinearity between x and x2 is the main reason of the inconsistent results because the correlation of x and x2 is extremely high (0.986).

```
# Calculate correlation
cor(data.frame(x, x2, y))
```

```
##            x        x2         y
## x  1.0000000 0.9860742 0.9050991
## x2 0.9860742 1.0000000 0.9308432
## y  0.9050991 0.9308432 1.0000000
```

```
# Plot residuals
plot(fitted(lm.fit2), resid(lm.fit2))
abline(0,0)
```



The residual is more likely to be heteroscedasticity, which violates the constant variance rule.

e) Discuss each of the Potential Problems (section 3.3.3) and determine whether they apply to this model. How about for your model from part a)?

The potential problem for model in part (a)

- Non-linearity of the Data: there is little pattern in the residuals.
- Non-constant Variance of Error Terms: the variance of residuals is constant.
- Outliers: one potential outlier when the residual is 20

The potential problem for model in part (b)

- Non-linearity of the Data: the relationship between the residuals and y is non-linear.
- Non-constant Variance of Error Terms: the residual tends to be heteroscedasticity, which violate the constant variance rule. I can apply log Y or square root of Y to reduce the effect.
- Outliers: there is no obvious outlier.
- Collinearity: x and x2 have a strong correlation, which is 0.986. I can either select one of the predictor variables or combine into one variable.

3)

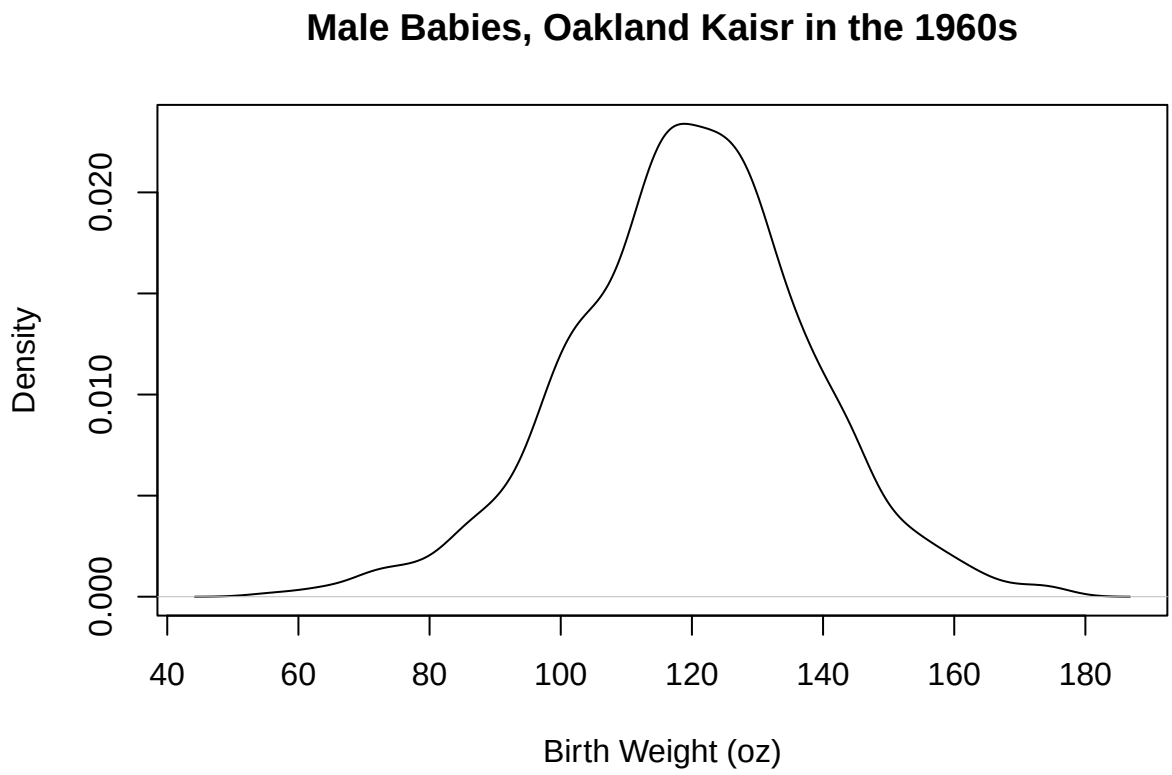
```
# Load the "infants" dataset
load(url("http://www.stodden.net/StatData/KaiserBabies.rda"))

# Check the data
names(infants)
```

```
##  [1] "gestation" "bwt"       "parity"    "age"       "ed"        "ht"
##  [7] "wt"        "dage"      "ded"       "dht"       "dwt"       "marital"
## [13] "inc"       "smoke"     "number"
```

```
dim(infants)
```

```
## [1] 1236   15
```

```
# y = bwt (birth weight)
# Density plot to check distribution of birth weight
plot(density(infants$bwt), xlab = "Birth Weight (oz)",
    main = "Male Babies, Oakland Kaisr in the 1960s")
```



The birth weight tends to be a normal distribution with mean = 120.

```r
## Understand the data
summary(infants)
```

```
##    gestation          bwt            parity            age
##  Min.   :148.0   Min.   : 55.0   Min.   : 0.000   Min.   :15.00
##  1st Qu.:272.0   1st Qu.:108.8   1st Qu.: 0.000   1st Qu.:23.00
##  Median :280.0   Median :120.0   Median : 1.000   Median :26.00
##  Mean   :279.3   Mean   :119.6   Mean   : 1.932   Mean   :27.26
##  3rd Qu.:288.0   3rd Qu.:131.0   3rd Qu.: 3.000   3rd Qu.:31.00
##  Max.   :353.0   Max.   :176.0   Max.   :13.000   Max.   :45.00
##  NA's   :13                                       NA's   :2
##                 ed            ht              wt             dage
##  No High School  : 19   Min.   :53.00   Min.   : 87.0   Min.   :18.00
##  Some High School:183   1st Qu.:62.00   1st Qu.:114.8   1st Qu.:25.00
##  High School     :444   Median :64.00   Median :125.0   Median :29.00
##  Trade           : 65   Mean   :64.05   Mean   :128.6   Mean   :30.35
##  Some College    :298   3rd Qu.:66.00   3rd Qu.:139.0   3rd Qu.:34.00
##  College         :219   Max.   :72.00   Max.   :250.0   Max.   :62.00
##  Unknown         :  8   NA's   :22      NA's   :36      NA's   :7
##                ded           dht            dwt           marital
##  No High School  : 33   Min.   :60.0   Min.   :110.0   Married:1208
##  Some High School:193   1st Qu.:68.0   1st Qu.:155.0   Once   :  20
##  High School     :342   Median :71.0   Median :170.0   Never  :   6
##  Trade           : 37   Mean   :70.2   Mean   :171.2   NA's   :   2
##  Some College    :265   3rd Qu.:72.0   3rd Qu.:185.0
##  College         :347   Max.   :78.0   Max.   :260.0
##  Unknown         : 19   NA's   :492    NA's   :499
##              inc                  smoke         number
##  [2500, 5000)  :195   Never         :544   Never  :544
##  [6000, 7000)  :180   Now           :484   20-29  :195
##  [5000, 6000)  :179   Until Pregnant: 95   5-9    :167
##  [10000, 12500):143   Once, Not Now :103   1-4    :155
##  [7000, 8000)  :138   Unknown       : 10   10-14  : 75
##  [8000, 9000)  :126                        30-39  : 32
##  (Other)       :275                        (Other): 68
```

Forward Selection

```r
# Change from categorical data to numerical data
infants.tranf = infants
infants.tranf$ed    = as.integer(infants$ed)
infants.tranf$smoke = as.integer(infants$smoke)
infants.tranf$ded   = as.integer(infants$ded)
infants.tranf$marital = as.integer(infants$marital)
infants.tranf$number = as.integer(infants$number)
infants.tranf$inc = as.integer(infants$inc)

## Fill missing values for dht and dwt with mean of its value
infants.tranf$dht[is.na(infants.tranf$dht)] <- mean(infants.tranf$dht, na.rm = T)
infants.tranf$dwt[is.na(infants.tranf$dwt)] <- mean(infants.tranf$dwt, na.rm = T)
## I fill the missing value because we will loose about 40% of data if we remove rows
```

```
## that are null and I choose mean because it is a reasonable representation of data.
## Additionally, we can neglect the columns that have small nulls.

# Recheck the manipulated data again
summary(infants.tranf)
```

```
##    gestation          bwt            parity            age
##  Min.   :148.0   Min.   : 55.0   Min.   : 0.000   Min.   :15.00
##  1st Qu.:272.0   1st Qu.:108.8   1st Qu.: 0.000   1st Qu.:23.00
##  Median :280.0   Median :120.0   Median : 1.000   Median :26.00
##  Mean   :279.3   Mean   :119.6   Mean   : 1.932   Mean   :27.26
##  3rd Qu.:288.0   3rd Qu.:131.0   3rd Qu.: 3.000   3rd Qu.:31.00
##  Max.   :353.0   Max.   :176.0   Max.   :13.000   Max.   :45.00
##  NA's   :13                                       NA's   :2
##        ed              ht              wt             dage
##  Min.   :1.000   Min.   :53.00   Min.   : 87.0   Min.   :18.00
##  1st Qu.:3.000   1st Qu.:62.00   1st Qu.:114.8   1st Qu.:25.00
##  Median :3.000   Median :64.00   Median :125.0   Median :29.00
##  Mean   :3.913   Mean   :64.05   Mean   :128.6   Mean   :30.35
##  3rd Qu.:5.000   3rd Qu.:66.00   3rd Qu.:139.0   3rd Qu.:34.00
##  Max.   :7.000   Max.   :72.00   Max.   :250.0   Max.   :62.00
##                  NA's   :22      NA's   :36      NA's   :7
##       ded             dht             dwt           marital
##  Min.   :1.000   Min.   :60.0   Min.   :110.0   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:70.0   1st Qu.:165.0   1st Qu.:1.000
##  Median :5.000   Median :70.2   Median :171.2   Median :1.000
##  Mean   :4.153   Mean   :70.2   Mean   :171.2   Mean   :1.026
##  3rd Qu.:6.000   3rd Qu.:71.0   3rd Qu.:175.0   3rd Qu.:1.000
##  Max.   :7.000   Max.   :78.0   Max.   :260.0   Max.   :3.000
##                                                 NA's   :2
##       inc             smoke           number
##  Min.   : 1.000   Min.   :1.000   Min.   : 1.000
##  1st Qu.: 3.000   1st Qu.:1.000   1st Qu.: 1.000
##  Median : 5.000   Median :2.000   Median : 2.000
##  Mean   : 5.333   Mean   :1.828   Mean   : 2.883
##  3rd Qu.: 8.000   3rd Qu.:2.000   3rd Qu.: 4.000
##  Max.   :11.000   Max.   :5.000   Max.   :10.000
##
```

Find the miminum RSE of these model and pick the lowest one I choose RSE because RSE stems from RSS, and RSE is easier to compute from the model.

```
lm.fitInf =lm(bwt~gestation, data=infants.tranf) # fit linear regression on bwt with gestation
sigma(lm.fitInf)
```

```
## [1] 16.66484
```

```
lm.fitInf =lm(bwt~parity, data=infants.tranf) # fit linear regression on bwt with parity
sigma(lm.fitInf)
```

```
## [1] 18.23583
```

```
lm.fitInf =lm(bwt~age, data=infants.tranf) # fit linear regression on bwt with age
sigma(lm.fitInf)
```

```
## [1] 18.24797
```

```
lm.fitInf =lm(bwt~ed, data=infants.tranf) # fit linear regression on bwt with ed
sigma(lm.fitInf)
```

```
## [1] 18.2303
```

```
lm.fitInf =lm(bwt~ht, data=infants.tranf) # fit linear regression on bwt with ht
sigma(lm.fitInf)
```

```
## [1] 17.93619
```

```
lm.fitInf =lm(bwt~wt, data=infants.tranf) # fit linear regression on bwt with wt
sigma(lm.fitInf)
```

```
## [1] 18.14825
```

```
lm.fitInf =lm(bwt~dage, data=infants.tranf) # fit linear regression on bwt with dage
sigma(lm.fitInf)
```

```
## [1] 18.22415
```

```
lm.fitInf =lm(bwt~ded, data=infants.tranf) # fit linear regression on bwt with ded
sigma(lm.fitInf)
```

```
## [1] 18.23667
```

```
lm.fitInf =lm(bwt~dht, data=infants.tranf) # fit linear regression on bwt with dht
sigma(lm.fitInf)
```

```
## [1] 18.17955
```

```
lm.fitInf =lm(bwt~dwt, data=infants.tranf) # fit linear regression on bwt with dwt
sigma(lm.fitInf)
```

```
## [1] 18.13125
```

```
lm.fitInf =lm(bwt~marital, data=infants.tranf) # fit linear regression on bwt with marital
sigma(lm.fitInf)
```

```
## [1] 18.24023
```

```r
lm.fitInf =lm(bwt~inc, data=infants.tranf) # fit linear regression on bwt with inc
sigma(lm.fitInf)
```

```
## [1] 18.23395
```

```r
lm.fitInf =lm(bwt~smoke, data=infants.tranf) # fit linear regression on bwt with smoke
sigma(lm.fitInf)
```

```
## [1] 18.24309
```

```r
lm.fitInf =lm(bwt~number, data=infants.tranf) # fit linear regression on bwt with number
sigma(lm.fitInf)
```

```
## [1] 18.05422
```

```r
# Pick gestation as a first variable because it gives the lowest RSE
lm.fitInf1 =lm(bwt~gestation, data=infants.tranf) # fit linear regression on bwt with gestation
summary(lm.fitInf1)
```

```
##
## Call:
## lm(formula = bwt ~ gestation, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.394 -11.125   0.071  10.106  57.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.06418    8.32220  -1.209    0.227
## gestation     0.46426    0.02974  15.609   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.66 on 1221 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.1663, Adjusted R-squared:  0.1657
## F-statistic: 243.6 on 1 and 1221 DF,  p-value: < 2.2e-16
```

```r
## Iterate over next loop to find another variable
lm.fitInf =lm(bwt~gestation + parity, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.61891
```

```r
lm.fitInf =lm(bwt~gestation + age, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.65793
```

```
lm.fitInf =lm(bwt~gestation + ed, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.66894
```

```
lm.fitInf =lm(bwt~gestation + ht, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.3747
```

```
lm.fitInf =lm(bwt~gestation + wt, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.56324
```

```
lm.fitInf =lm(bwt~gestation + dage, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.63689
```

```
lm.fitInf =lm(bwt~gestation + ded, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.67145
```

```
lm.fitInf =lm(bwt~gestation + dht, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.60876
```

```
lm.fitInf =lm(bwt~gestation + dwt, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.55806
```

```
lm.fitInf =lm(bwt~gestation + marital, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.58193
```

```
lm.fitInf =lm(bwt~gestation + inc, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.66753
```

```
lm.fitInf =lm(bwt~gestation + smoke, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.66619
```

```
lm.fitInf =lm(bwt~gestation + number, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.46566
```

```
# Pick ht as a second variable because it gives the lowest RSE
lm.fitInf2 =lm(bwt~gestation + ht, data=infants.tranf)

## Check for p-value
summary(lm.fitInf2)
```

```
##
## Call:
## lm(formula = bwt ~ gestation + ht, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.817 -10.629   0.344  10.232  54.289
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -89.76084   14.12619  -6.354 2.97e-10 ***
## gestation     0.45859    0.02961  15.486  < 2e-16 ***
## ht            1.26883    0.18710   6.782 1.86e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.37 on 1199 degrees of freedom
##   (34 observations deleted due to missingness)
## Multiple R-squared:  0.2003, Adjusted R-squared:  0.199
## F-statistic: 150.2 on 2 and 1199 DF,  p-value: < 2.2e-16
```

Since p-values for both variables are significantly less than 0.05 and adjusted R-squared , I will keep adding another variable.

```
## Find next variable
lm.fitInf =lm(bwt~gestation + ht + parity, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.32043
```

```
lm.fitInf =lm(bwt~gestation + ht + age, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.36322
```

```
lm.fitInf =lm(bwt~gestation + ht + ed, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.38129
```

```
lm.fitInf =lm(bwt~gestation + ht + wt, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.38965
```

```
lm.fitInf =lm(bwt~gestation + ht + dage, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.34297
```

```
lm.fitInf =lm(bwt~gestation + ht + ded, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.37762
```

```
lm.fitInf =lm(bwt~gestation + ht + dht, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.37102
```

```
lm.fitInf =lm(bwt~gestation + ht + dwt, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.32833
```

```
lm.fitInf =lm(bwt~gestation + ht + marital, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.29628
```

```
lm.fitInf =lm(bwt~gestation + ht + inc, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.37966
```

```
lm.fitInf =lm(bwt~gestation + ht + smoke, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.36645
```

```r
lm.fitInf =lm(bwt~gestation + ht + number, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.10435
```

```r
# Pick number as a second variable because it's model gives the lowest RSE
lm.fit3 =lm(bwt~gestation + ht + number, data=infants.tranf)
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = bwt ~ gestation + ht + number, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.860 -10.337  -0.095   9.836  52.168
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -92.10531   13.89773  -6.627 5.15e-11 ***
## gestation     0.45525    0.02913  15.629  < 2e-16 ***
## ht            1.37957    0.18481   7.465 1.60e-13 ***
## number       -1.32813    0.20594  -6.449 1.63e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.1 on 1198 degrees of freedom
##   (34 observations deleted due to missingness)
## Multiple R-squared:  0.2272, Adjusted R-squared:  0.2252
## F-statistic: 117.4 on 3 and 1198 DF,  p-value: < 2.2e-16
```

I will keep this variable because p-values of each variable is less than 0.05.

```r
## Start next iteration
lm.fitInf =lm(bwt~gestation + ht + number + parity, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.04888
```

```r
lm.fitInf =lm(bwt~gestation + ht + number + age, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.09483
```

```r
lm.fitInf =lm(bwt~gestation + ht + number + ed, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.10805
```

```
lm.fitInf =lm(bwt~gestation + ht + number + wt, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.13192
```

```
lm.fitInf =lm(bwt~gestation + ht + number + dage, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.06807
```

```
lm.fitInf =lm(bwt~gestation + ht + number + ded, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.10691
```

```
lm.fitInf =lm(bwt~gestation + ht + number + dht, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.09461
```

```
lm.fitInf =lm(bwt~gestation + ht + number + dwt, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.05379
```

```
lm.fitInf =lm(bwt~gestation + ht + number + marital, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.03623
```

```
lm.fitInf =lm(bwt~gestation + ht + number + inc, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.10667
```

```
lm.fitInf =lm(bwt~gestation + ht + number + smoke, data=infants.tranf)
sigma(lm.fitInf)
```

```
## [1] 16.05988
```

```
# Pick marital and check p-values
lm.fit4 =lm(bwt~gestation + ht + number + marital, data=infants.tranf)
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = bwt ~ gestation + ht + number + marital, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.700 -10.263  -0.133   9.874  52.151
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -93.34257   14.13923  -6.602 6.10e-11 ***
## gestation     0.47844    0.02988  16.012  < 2e-16 ***
## ht            1.35638    0.18424   7.362 3.36e-13 ***
## number       -1.29999    0.20528  -6.333 3.40e-10 ***
## marital      -3.77467    2.44243  -1.545    0.123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.04 on 1195 degrees of freedom
##   (36 observations deleted due to missingness)
## Multiple R-squared:  0.2356, Adjusted R-squared:  0.233
## F-statistic: 92.08 on 4 and 1195 DF,  p-value: < 2.2e-16
```

Since the p-value of marital is more than 0.05, the null hypothesis has been rejected. Moreover, adjusted R-squared increases insignificantly from lm.fit3 (0.199) to lm.fit4 (0.233). For these reasons, I will stop iterate over independent variables because the stop condition has been met.

```
## Finaly model
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = bwt ~ gestation + ht + number, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.860 -10.337  -0.095   9.836  52.168
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -92.10531   13.89773  -6.627 5.15e-11 ***
## gestation     0.45525    0.02913  15.629  < 2e-16 ***
## ht            1.37957    0.18481   7.465 1.60e-13 ***
## number       -1.32813    0.20594  -6.449 1.63e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.1 on 1198 degrees of freedom
##   (34 observations deleted due to missingness)
## Multiple R-squared:  0.2272, Adjusted R-squared:  0.2252
## F-statistic: 117.4 on 3 and 1198 DF,  p-value: < 2.2e-16
```

With the mull model, the birthweight of a baby is -92.10 oz. With a 100 of gestation, height, and number, the birthweight would increase 45.5 oz, 138 oz, and decrease 133 oz respectively.

The relationships of each variable do not occur with birthweight by chance because low extremely low p-values.
At least one variable has significant relationship with birthweight indicating with p-value of F-statistics less than 0.05.
Finally, only about 23% of total variance of baby's birthweight can be explained by this model.

Backward selection

```
## Start with a model with all variables
lm.backward =lm(bwt~., data=infants.tranf)
summary(lm.backward)
```

```
##
## Call:
## lm(formula = bwt ~ ., data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.400  -9.909  -0.066   9.661  49.663
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.091e+02  2.001e+01  -5.455 5.99e-08 ***
## gestation    4.806e-01  3.036e-02  15.832  < 2e-16 ***
## parity       5.357e-01  3.126e-01   1.713  0.08689 .
## age         -6.927e-03  1.520e-01  -0.046  0.96365
## ed           7.512e-02  4.194e-01   0.179  0.85788
## ht           1.011e+00  2.177e-01   4.646 3.78e-06 ***
## wt           5.699e-02  2.625e-02   2.171  0.03010 *
## dage         4.167e-02  1.231e-01   0.338  0.73510
## ded          3.087e-02  3.669e-01   0.084  0.93297
## dht          1.766e-01  2.546e-01   0.694  0.48797
## dwt          6.594e-02  3.203e-02   2.059  0.03976 *
## marital     -2.401e+00  2.695e+00  -0.891  0.37315
## inc          9.838e-02  1.664e-01   0.591  0.55451
## smoke        1.696e+00  6.074e-01   2.792  0.00533 **
## number      -1.724e+00  2.564e-01  -6.726 2.74e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.97 on 1160 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.2541, Adjusted R-squared:  0.2451
## F-statistic: 28.22 on 14 and 1160 DF,  p-value: < 2.2e-16
```

Since age has the highest p-value (0.96365), I will remove this variable.

```
lm.backward =lm(bwt~. -age, data=infants.tranf)
summary(lm.backward)
```

```
##
```

```
## Call:
## lm(formula = bwt ~ . - age, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.430  -9.897  -0.057   9.700  49.662
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -109.12530   19.99644  -5.457 5.91e-08 ***
## gestation      0.48062    0.03034  15.843  < 2e-16 ***
## parity         0.53098    0.29490   1.801  0.07203 .
## ed             0.07207    0.41385   0.174  0.86178
## ht             1.01129    0.21759   4.648 3.74e-06 ***
## wt             0.05702    0.02623   2.174  0.02990 *
## dage           0.03747    0.08181   0.458  0.64700
## ded            0.03028    0.36656   0.083  0.93418
## dht            0.17580    0.25384   0.693  0.48872
## dwt            0.06597    0.03201   2.061  0.03954 *
## marital       -2.39446    2.68993  -0.890  0.37357
## inc            0.09777    0.16580   0.590  0.55552
## smoke          1.69687    0.60674   2.797  0.00525 **
## number        -1.72509    0.25586  -6.742 2.45e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.96 on 1161 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.2541, Adjusted R-squared:  0.2457
## F-statistic: 30.42 on 13 and 1161 DF,  p-value: < 2.2e-16
```

Since ded has the highest p-value (0.93418), I will remove this variable.

```
lm.backward =lm(bwt~. -age - ded, data=infants.tranf)
summary(lm.backward)
```

```
##
## Call:
## lm(formula = bwt ~ . - age - ded, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.424  -9.925  -0.076   9.735  49.598
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -109.17436   19.97908  -5.464 5.68e-08 ***
## gestation      0.48070    0.03031  15.861  < 2e-16 ***
## parity         0.53017    0.29460   1.800  0.07219 .
## ed             0.09104    0.34414   0.265  0.79141
## ht             1.01297    0.21654   4.678 3.24e-06 ***
## wt             0.05675    0.02601   2.182  0.02934 *
## dage           0.03759    0.08176   0.460  0.64578
```

```
## dht            0.17581    0.25373   0.693  0.48851
## dwt            0.06594    0.03199   2.061  0.03953 *
## marital       -2.39615    2.68870  -0.891  0.37301
## inc            0.09858    0.16544   0.596  0.55137
## smoke          1.69867    0.60609   2.803  0.00515 **
## number        -1.72482    0.25573  -6.745 2.41e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.95 on 1162 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.2541, Adjusted R-squared:  0.2464
## F-statistic: 32.98 on 12 and 1162 DF,  p-value: < 2.2e-16
```

Since ed has the highest p-value (0.79141), I will remove this variable.

```
lm.backward =lm(bwt~. -age - ded -ed, data=infants.tranf)
summary(lm.backward)
```

```
##
## Call:
## lm(formula = bwt ~ . - age - ded - ed, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.357  -9.975  -0.038   9.710  49.752
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -109.66709   19.88411  -5.515 4.29e-08 ***
## gestation      0.48079    0.03029  15.871  < 2e-16 ***
## parity         0.51180    0.28619   1.788  0.07399 .
## ht             1.02177    0.21389   4.777 2.00e-06 ***
## wt             0.05597    0.02583   2.166  0.03048 *
## dage           0.04155    0.08034   0.517  0.60510
## dht            0.18186    0.25260   0.720  0.47170
## dwt            0.06513    0.03183   2.046  0.04100 *
## marital       -2.41144    2.68700  -0.897  0.36967
## inc            0.10093    0.16514   0.611  0.54122
## smoke          1.70342    0.60558   2.813  0.00499 **
## number        -1.72998    0.25488  -6.787 1.82e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.95 on 1163 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.254,  Adjusted R-squared:  0.247
## F-statistic:    36 on 11 and 1163 DF,  p-value: < 2.2e-16
```

Since dage has the highest p-value (0.60510), I will remove this variable.

```
lm.backward =lm(bwt~. -age - ded -ed -dage, data=infants.tranf)
summary(lm.backward)
```

```
##
## Call:
## lm(formula = bwt ~ . - age - ded - ed - dage, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.834  -9.919   0.033   9.606  49.482
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -107.81881   19.55421  -5.514 4.32e-08 ***
## gestation      0.48092    0.03028  15.881  < 2e-16 ***
## parity         0.57961    0.25431   2.279   0.0228 *
## ht             1.01677    0.21360   4.760 2.18e-06 ***
## wt             0.05773    0.02560   2.255   0.0243 *
## dht            0.17166    0.25175   0.682   0.4955
## dwt            0.06473    0.03181   2.035   0.0421 *
## marital       -2.37417    2.68519  -0.884   0.3768
## inc            0.11669    0.16225   0.719   0.4721
## smoke          1.72068    0.60447   2.847   0.0045 **
## number        -1.74036    0.25401  -6.852 1.18e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.94 on 1164 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.2538, Adjusted R-squared:  0.2474
## F-statistic:  39.6 on 10 and 1164 DF,  p-value: < 2.2e-16
```

Since dht has the highest p-value (0.4955), I will remove this variable.

```
lm.backward =lm(bwt~. -age - ded -ed -dage -dht, data=infants.tranf)
summary(lm.backward)
```

```
##
## Call:
## lm(formula = bwt ~ . - age - ded - ed - dage - dht, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.976  -9.919  -0.041   9.576  50.145
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -99.15566   14.86080  -6.672 3.88e-11 ***
## gestation     0.48050    0.03027  15.874  < 2e-16 ***
## parity        0.56418    0.25324   2.228  0.02608 *
## ht            1.04319    0.21001   4.967 7.80e-07 ***
## wt            0.05716    0.02558   2.234  0.02564 *
```

```
## dwt           0.07591    0.02726   2.785  0.00544 **
## marital      -2.37035    2.68457  -0.883  0.37744
## inc           0.11709    0.16221   0.722  0.47056
## smoke         1.70515    0.60390   2.824  0.00483 **
## number       -1.72996    0.25349  -6.824 1.42e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.94 on 1165 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.2535, Adjusted R-squared:  0.2478
## F-statistic: 43.97 on 9 and 1165 DF,  p-value: < 2.2e-16
```

Since inc has the highest p-value (0.47056), I will remove this variable.

```
lm.backward =lm(bwt~. -age - ded -ed -dage -dht -inc, data=infants.tranf)
summary(lm.backward)
```

```
##
## Call:
## lm(formula = bwt ~ . - age - ded - ed - dage - dht - inc, data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.430  -9.975  -0.159   9.419  50.043
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -99.01886   14.85654  -6.665 4.07e-11 ***
## gestation     0.48106    0.03025  15.901  < 2e-16 ***
## parity        0.56326    0.25318   2.225  0.02629 *
## ht            1.04546    0.20994   4.980 7.33e-07 ***
## wt            0.05680    0.02557   2.221  0.02653 *
## dwt           0.07704    0.02721   2.831  0.00471 **
## marital      -2.34292    2.68375  -0.873  0.38284
## smoke         1.70416    0.60378   2.823  0.00485 **
## number       -1.72397    0.25331  -6.806 1.60e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.93 on 1166 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.2532, Adjusted R-squared:  0.2481
## F-statistic: 49.42 on 8 and 1166 DF,  p-value: < 2.2e-16
```

Since marital has the highest p-value (0.38284), I will remove this variable.

```
lm.backward =lm(bwt~. -age - ded -ed -dage -dht -inc - marital
                , data=infants.tranf)
summary(lm.backward)
```

```
##
```

```
## Call:
## lm(formula = bwt ~ . - age - ded - ed - dage - dht - inc - marital,
##     data = infants.tranf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.386 -10.092  -0.178   9.415  50.121
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -101.62362   14.55237  -6.983 4.83e-12 ***
## gestation      0.48151    0.03025  15.920  < 2e-16 ***
## parity         0.56842    0.25309   2.246  0.02490 *
## ht             1.04714    0.20991   4.988 7.01e-07 ***
## wt             0.05625    0.02556   2.201  0.02796 *
## dwt            0.07727    0.02720   2.841  0.00458 **
## smoke          1.72367    0.60330   2.857  0.00435 **
## number        -1.73737    0.25282  -6.872 1.03e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.93 on 1167 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.2527, Adjusted R-squared:  0.2482
## F-statistic: 56.38 on 7 and 1167 DF,  p-value: < 2.2e-16
```

As all variables have p-values less thab 0.05, I will stop backward selection.

Interpretation of the model
With the null model, the birthweight of a baby is -101.6 oz. With a 100 of gestation, parity, ht, wt,dwt, smoke, and number, the birthweight would increase 48.2 oz, 56.8 oz, 104.7 oz, 5.6 oz, 7.7 oz, 172.4 oz, and decrease 173.7 oz respectively.
The relationships of each variable do not occur with birthweight by chance because low extremely low p-values.
At least one variable has significant relationship with birthweight indicating with p-value of F-statistics less than 0.05.
Finally, only about 25% of total variance of baby's birthweight can be explained by this backward selection model.

To compare models forward and backward selection, they both have gestation, height, and number, but the backward selection model has parity, wt, dwt, and smoke as additional features. This is because parity, wt, dwt, and smoke have extremely low p-values, and thus we cannot remove them.
If I would have to choose between these two models, I would pick choose based on lower RSE and higher R-squared because low RSE means the predictor(y hat) is closed to the actual y and high R-squared indicates how the model explains the total variance of the predictor. Therefore, I would choose the backward selection model because it has lower RSE and higher R-squared.