a) One of the first steps in statistical modeling is to decide on the model structure. What do you have to take into account to balance the trade-off between variance and bias?

In most cases we decompose the model's expected error (although we learned it as MSE) on an unseen sample x as follows:

$$\mathrm{E}\Big[\big(y-\hat{f}(x)\big)^2\Big] = \mathrm{Bias}\big[\hat{f}(x)\big]^2 + \mathrm{Var}\big[\hat{f}(x)\big] + \sigma^2$$

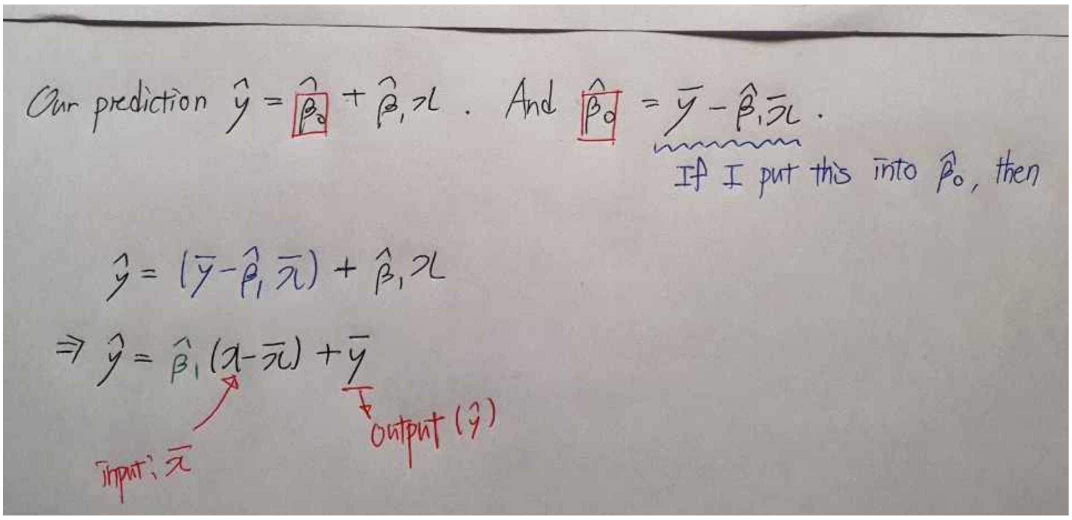
, our goal is to minimize both bias and variance, who are reducible errors, in statistical modeling. High Bias is likely to cause underfitting problem. In contrast, High variance would cause overfitting problem. But generally we fall into the dilemma that it is almost impossible to minimize them simultaneously, which is called 'bias-variance tradeoff'. So we should consider and tune the model flexibility (complexity) (and degree) to find the point which marks optimum model complexity, the lowest Total Error.

b) in the case of simple linear regression, these equations define the coefficients:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x},$$

argue that the regression line always passes through the point (mean(x) ,mean(y)).



Our prediction $\hat{y} = \boxed{\hat{\beta}_0} + \hat{\beta}_1 x$. And $\boxed{\hat{\beta}_0} = \underline{\bar{y} - \hat{\beta}_1 \bar{x}}$.

If I put this into $\hat{\beta}_0$, then

$$\hat{y} = (\bar{y} - \hat{\beta}_1 \bar{x}) + \hat{\beta}_1 x$$

$$\Rightarrow \hat{y} = \hat{\beta}_1 (x - \bar{x}) + \bar{y}$$

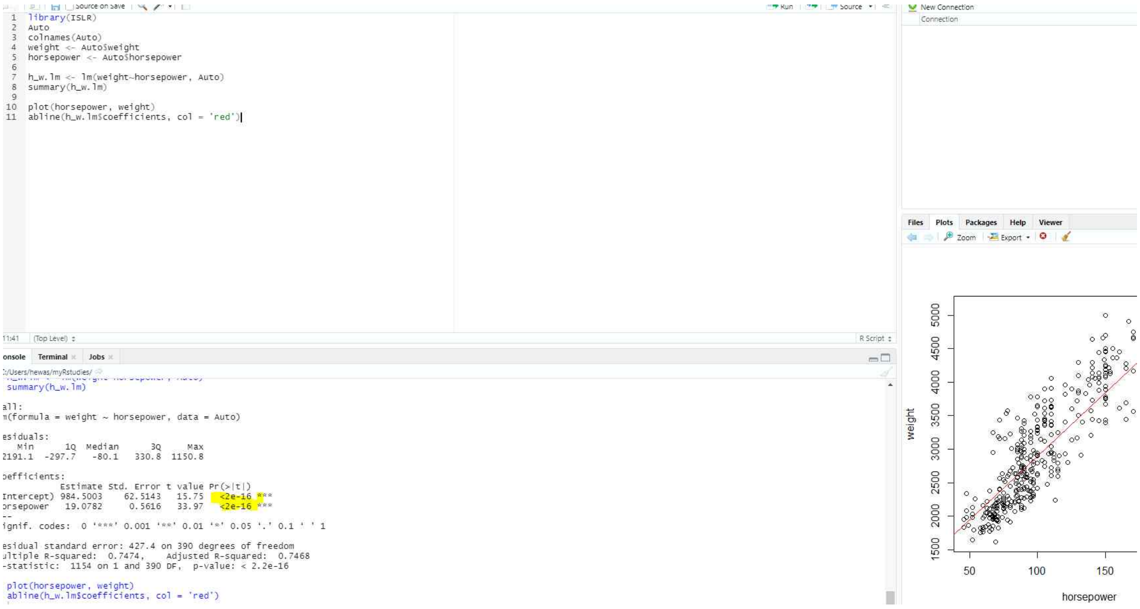
input: $\bar{x}$

output ($\hat{y}$)

Therefore, whatever x bar and y bar are, the regression line always passes through the point (mean(x), mean(y)).

c) Use the `Auto' data with library(ISLR) and lm() to perform a simple linear regression with weight as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output.

Is there a relationship between the predictor and the response?

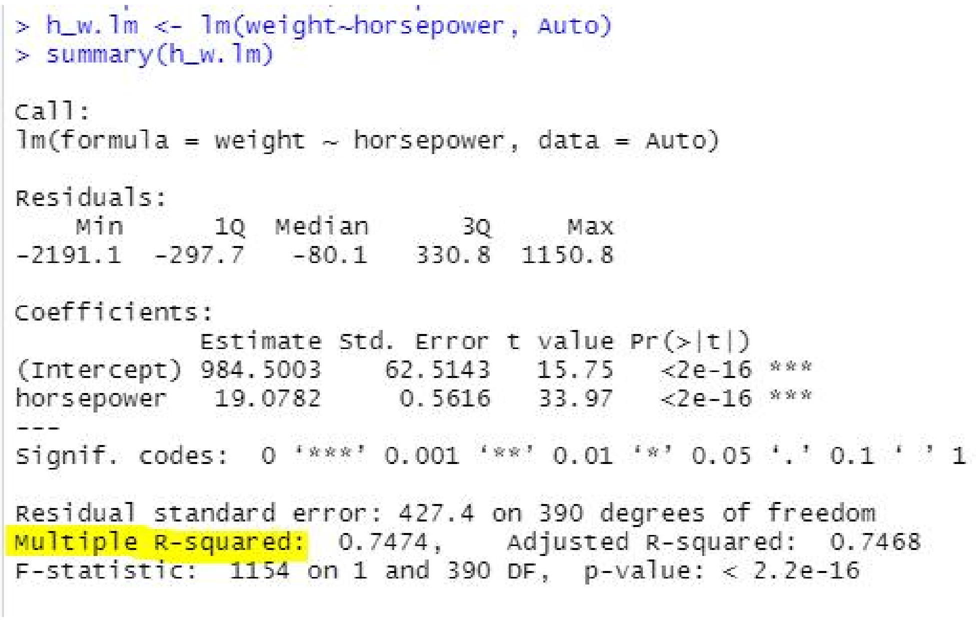


```
library(ISLR)
Auto
colnames(Auto)
weight <- Auto$weight
horsepower <- Auto$horsepower

h_w.lm <- lm(weight~horsepower, Auto)
summary(h_w.lm)

plot(horsepower, weight)
abline(h_w.lm$coefficients, col = 'red')
```

```
:/Users/hewas/myRstudies/
 summary(h_w.lm)

all:
n(formula = weight ~ horsepower, data = Auto)

esiduals:
   Min      1Q  Median      3Q     Max
2191.1  -297.7   -80.1   330.8  1150.8

oefficients:
            Estimate Std. Error t value Pr(>|t|)
Intercept) 984.5003    62.5143   15.75   <2e-16 ***
orsepower   19.0782     0.5616   33.97   <2e-16 ***
--
ignif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

esidual standard error: 427.4 on 390 degrees of freedom
ultiple R-squared:  0.7474,    Adjusted R-squared:  0.7468
-statistic:  1154 on 1 and 390 DF,  p-value: < 2.2e-16

 plot(horsepower, weight)
 abline(h_w.lm$coefficients, col = 'red')
```

Yes there is. Because the p-values for the regression coefficients are extremely small, which means that there is a relationship between the response and the predictor.

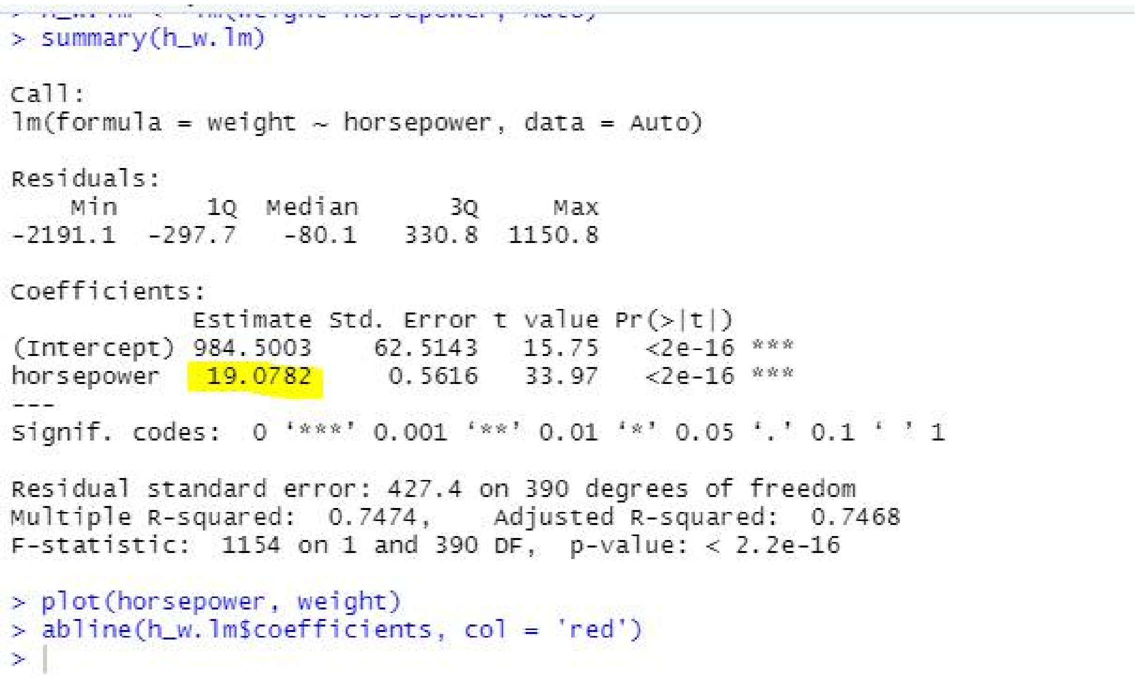
How strong is the relationship between the predictor and the response?



```
> h_w.lm <- lm(weight~horsepower, Auto)
> summary(h_w.lm)

Call:
lm(formula = weight ~ horsepower, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max 
-2191.1  -297.7   -80.1   330.8  1150.8 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 984.5003    62.5143   15.75   <2e-16 ***
horsepower   19.0782     0.5616   33.97   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 427.4 on 390 degrees of freedom
Multiple R-squared:  0.7474,	Adjusted R-squared:  0.7468 
F-statistic:  1154 on 1 and 390 DF,  p-value: < 2.2e-16
```

As we can see 0.7474 which is suggested by the indicator ‘Multiple R-squared’, there’s about 75% association between them.

Is the relationship between the predictor and the response positive or negative?

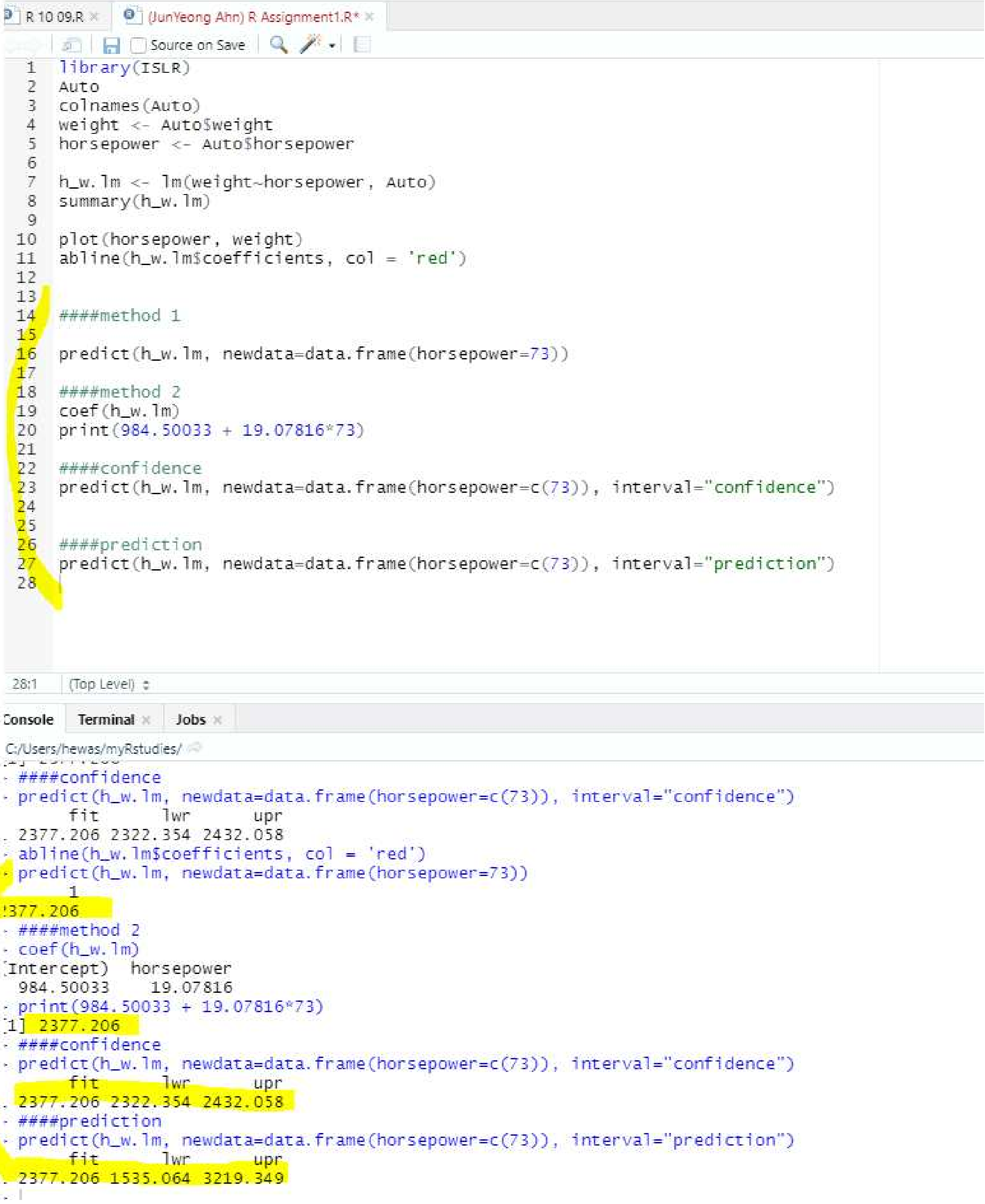


```
> summary(h_w.lm)

Call:
lm(formula = weight ~ horsepower, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max 
-2191.1  -297.7   -80.1   330.8  1150.8 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 984.5003    62.5143   15.75   <2e-16 ***
horsepower   19.0782     0.5616   33.97   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 427.4 on 390 degrees of freedom
Multiple R-squared:  0.7474,	Adjusted R-squared:  0.7468 
F-statistic:  1154 on 1 and 390 DF,  p-value: < 2.2e-16

> plot(horsepower, weight)
> abline(h_w.lm$coefficients, col = 'red')
> 
```

It is positive since the coefficient for the horsepower is bigger than 0.

What is the 1) predicted weight associated with a horsepower of 73? What are the associated 95% 2) confidence and 3) prediction intervals?

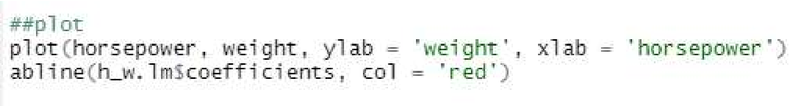
1) 2377.206 / 2) [2322.354, 2432.058] / 3) [1535.064, 3219.349]



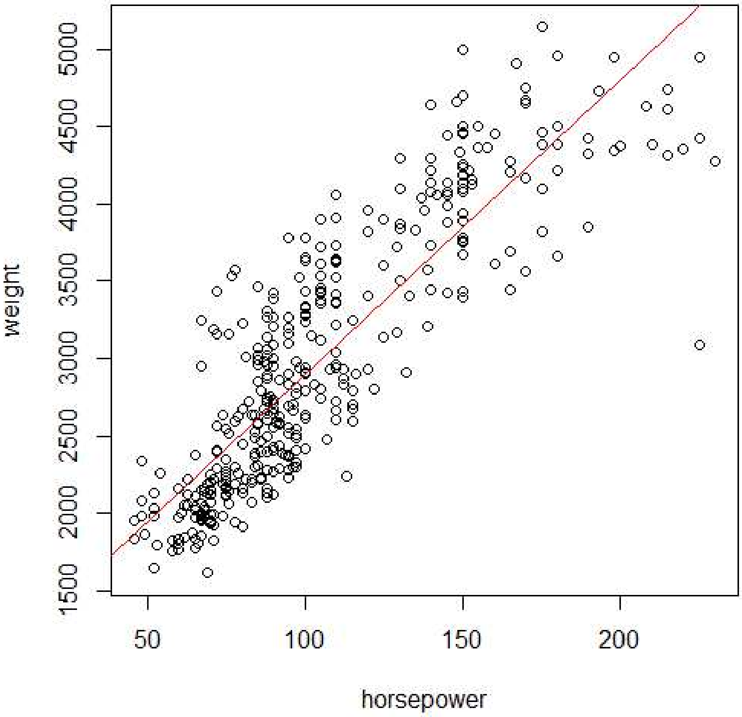
```r
library(ISLR)
Auto
colnames(Auto)
weight <- Auto$weight
horsepower <- Auto$horsepower

h_w.lm <- lm(weight~horsepower, Auto)
summary(h_w.lm)

plot(horsepower, weight)
abline(h_w.lm$coefficients, col = 'red')


####method 1

predict(h_w.lm, newdata=data.frame(horsepower=73))

####method 2
coef(h_w.lm)
print(984.50033 + 19.07816*73)

####confidence
predict(h_w.lm, newdata=data.frame(horsepower=c(73)), interval="confidence")


####prediction
predict(h_w.lm, newdata=data.frame(horsepower=c(73)), interval="prediction")

```

```
 ####confidence
 predict(h_w.lm, newdata=data.frame(horsepower=c(73)), interval="confidence")
       fit      lwr      upr
 2377.206 2322.354 2432.058
 abline(h_w.lm$coefficients, col = 'red')
 predict(h_w.lm, newdata=data.frame(horsepower=73))
       1
377.206
 ####method 2
 coef(h_w.lm)
Intercept)  horsepower
 984.50033   19.07816
 print(984.50033 + 19.07816*73)
1] 2377.206
 ####confidence
 predict(h_w.lm, newdata=data.frame(horsepower=c(73)), interval="confidence")
       fit      lwr      upr
 2377.206 2322.354 2432.058
 ####prediction
 predict(h_w.lm, newdata=data.frame(horsepower=c(73)), interval="prediction")
       fit      lwr      upr
 2377.206 1535.064 3219.349
```

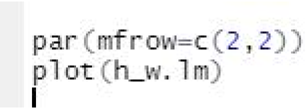
Plot the response and the predictor. Label the axis meaningful. Use the abline() function to display the least squares regression line.



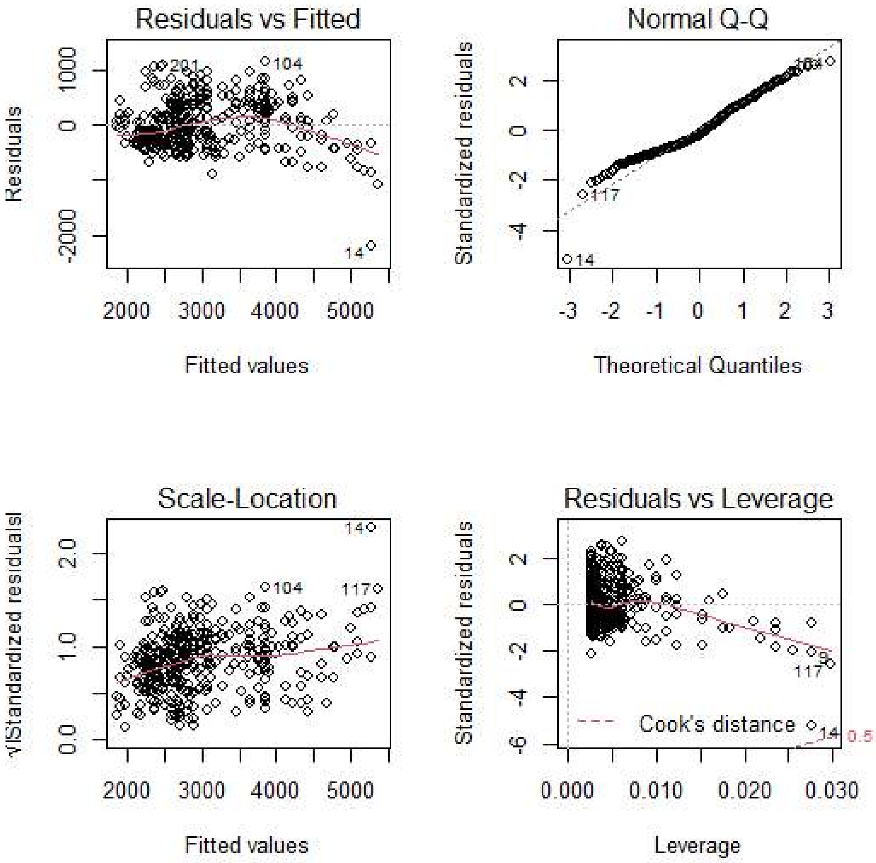
```
##plot
plot(horsepower, weight, ylab = 'weight', xlab = 'horsepower')
abline(h_w.lm$coefficients, col = 'red')
```



Use the plot() function to produce diagnostic plots of the least squares regression fit and explain for each plot what kind of information are displayed and what they mean.



```
par(mfrow=c(2,2))
plot(h_w.lm)
```



Residuals vs Fitted)

This shows a pattern between the residuals and the fitted values. It seems like a hat-shaped one, so we could infer that there is a non-linear relationship between the predictor and response variables.

Normal Q-Q)

This plot shows that residuals are normally distributed because residuals are lined well on the straight dashed line.

Scale-Location)

This plot indicates that the variance of the errors is almost constant because the red line is nearly horizontal.

Residuals vs Leverage)

This graph helps us to detect outliers. Since Cook's distance (red dashed line) has barely appeared and our red line has never gone outside it, we can say that there are no leverage points in the data.