Lab4.R

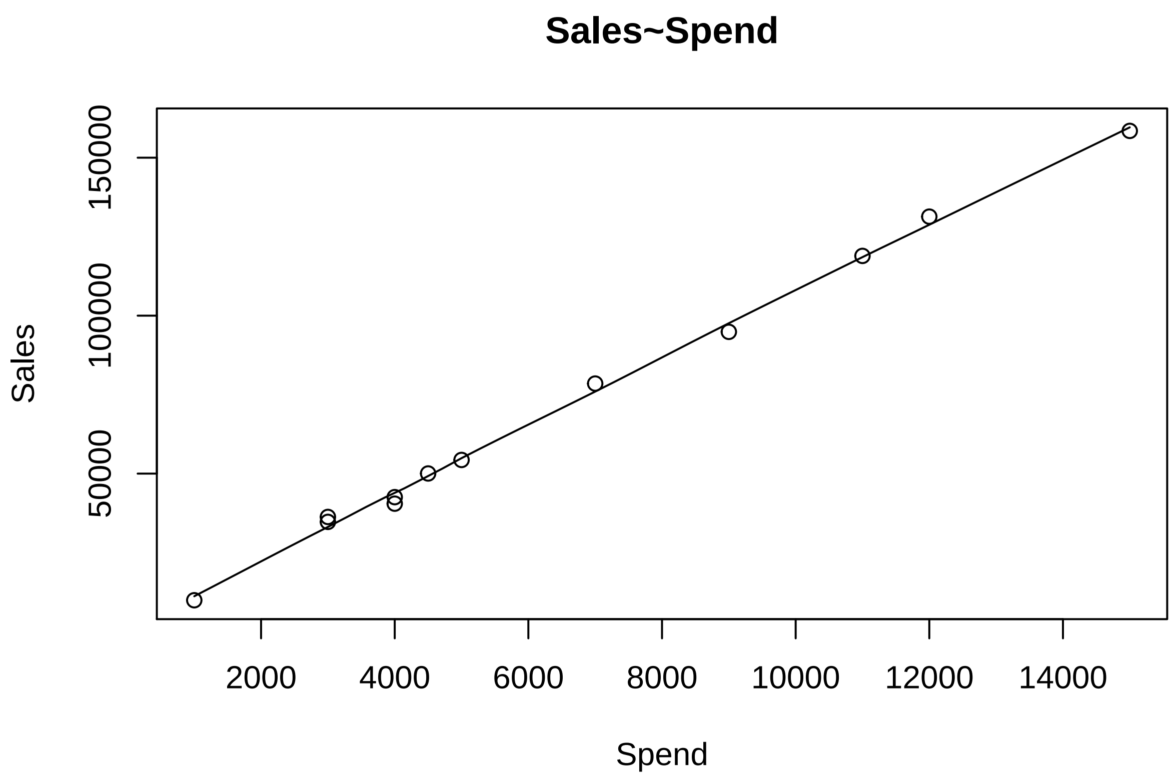
rstudio-user

2021-02-08

```r
#1. Read dataset
data <- read.csv('marketing.csv')
print(data)
```

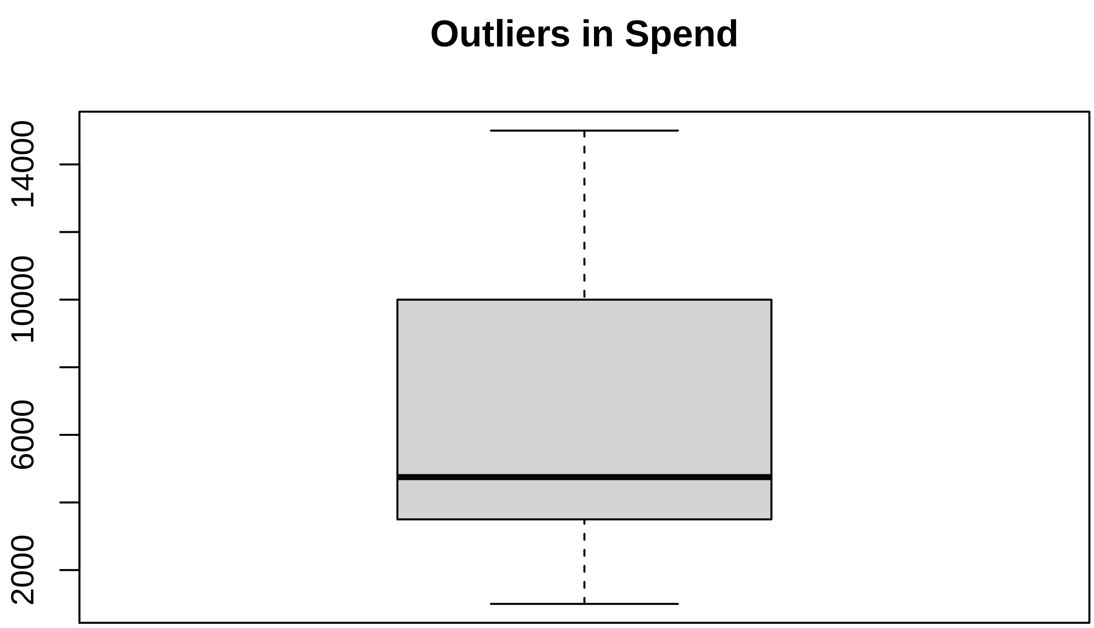
```
##    Month Spend  Sales
## 1      1  1000   9914
## 2      2  4000  40487
## 3      3  5000  54324
## 4      4  4500  50044
## 5      5  3000  34719
## 6      6  4000  42551
## 7      7  9000  94871
## 8      8 11000 118914
## 9      9 15000 158484
## 10    10 12000 131348
## 11    11  7000  78504
## 12    12  3000  36284
```

```r
#2. Use Scatter Plot to visualize the Relationship
scatter.smooth(x=data$Spend, y=data$Sales, main="Sales~Spend", xlab="Spend", ylab="Sales")
```

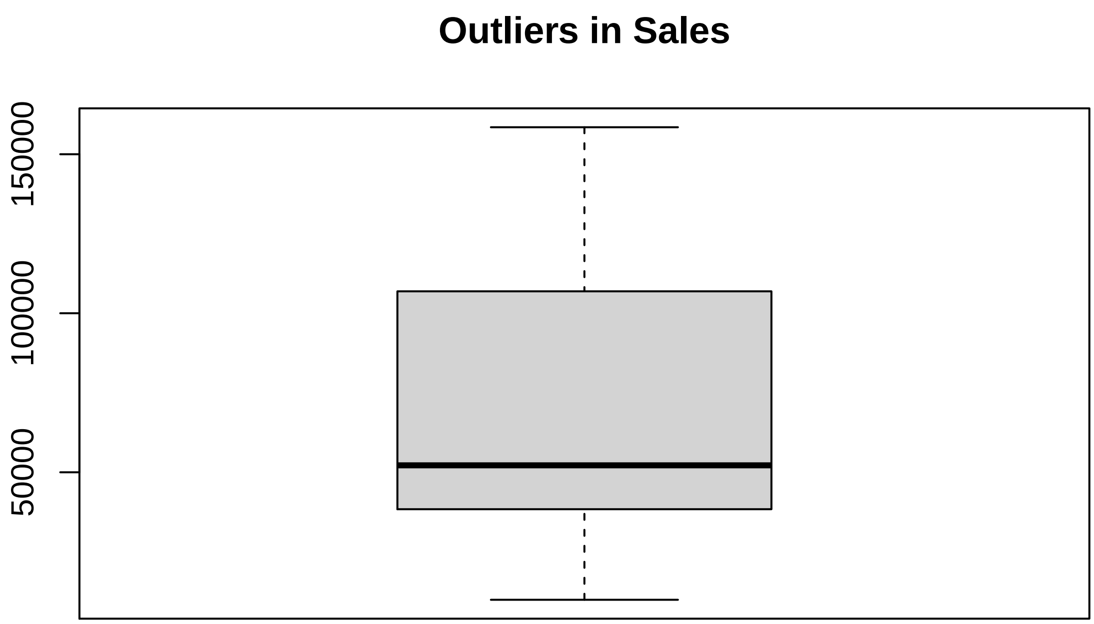



```
#With the scatter plot, we can easily see that the data has a positive strong correlation.


#3. Using BoxPlot to check for Outliers
boxplot(data$Spend, main="Outliers in Spend")
```

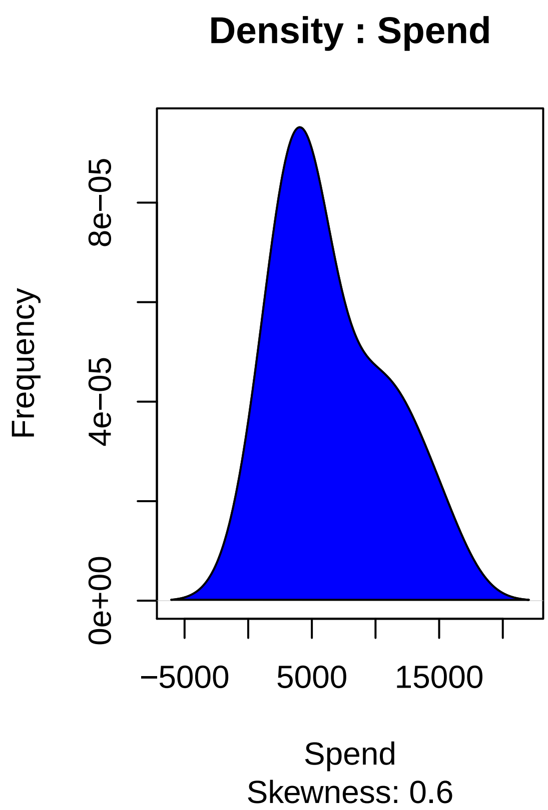


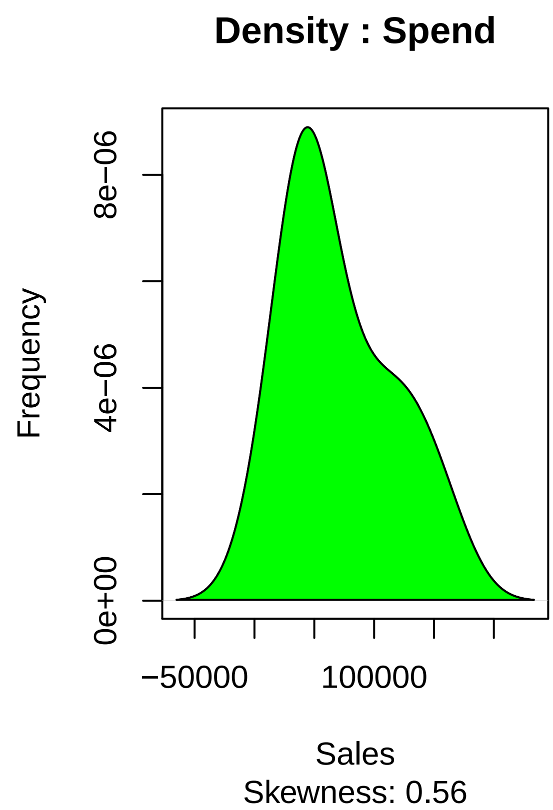

```
boxplot(data$Sales, main="Outliers in Sales")
```




```
#There are no outliers in either of the columns as there are no data point outside the plot range.


#4. Using Density Plot To Check If Response Variable Is Close To Normal
library(e1071)
par(mfrow=c(1, 2))
plot(density(data$Spend), main="Density : Spend", xlab="Spend", ylab="Frequency", sub=paste("Skewness:"
polygon(density(data$Spend), col="blue")
plot(density(data$Sales), main="Density : Spend", xlab="Sales", ylab="Frequency", sub=paste("Skewness:"
polygon(density(data$Sales), col="green")
```






```
#5. Check the Correlation Analysis
cor(data$Spend, data$Sales)
```

```
## [1] 0.9988322
```

```
#Since the correlation of the two variables is above 0.99, approx 1, we can say that they have a strong

#6. Build the Linear Regression Model
model <- lm(Sales~Spend, data=data)

#7. Using p-value Check For Statistical Significance
modelsummary <- summary(model)
modelcoeffs <- modelsummary$coefficients
est <- modelcoeffs['Spend', 'Estimate']
stderr <- modelcoeffs['Spend', 'Std. Error']
tval <- est/stderr
pval <- 2*pt(-abs(tval), df=length(data$Sales)-1)
print(paste("p-value:", pval))
```

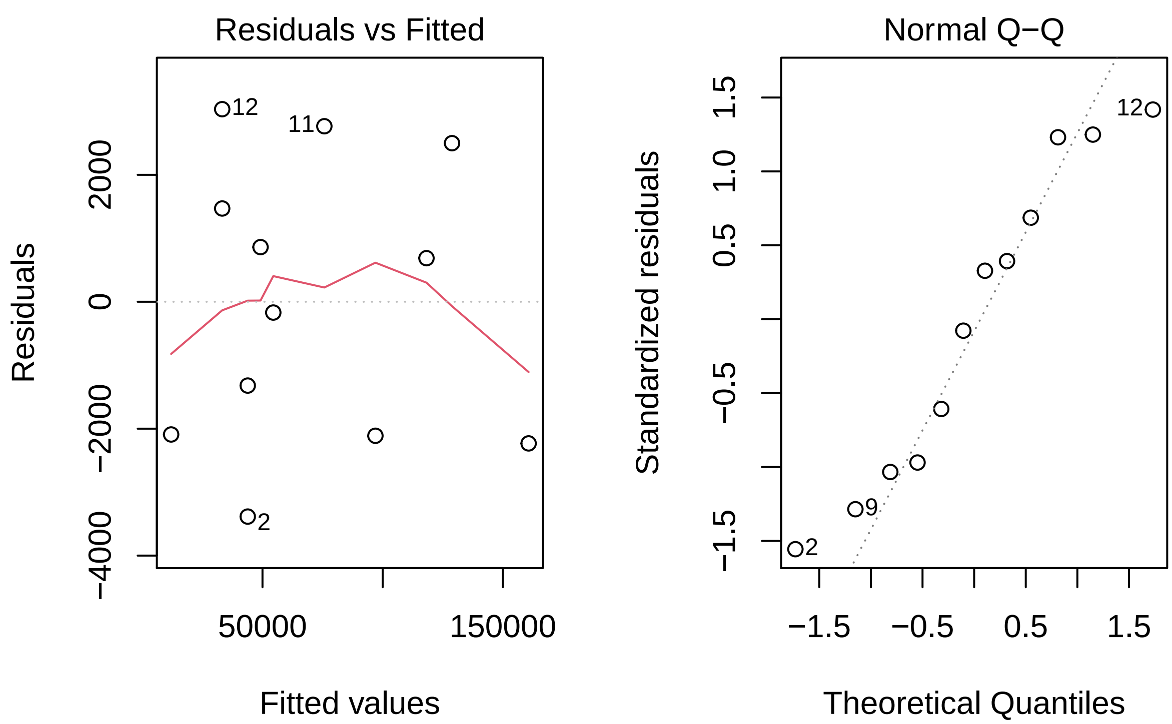
```
## [1] "p-value: 1.32941244094728e-15"
```

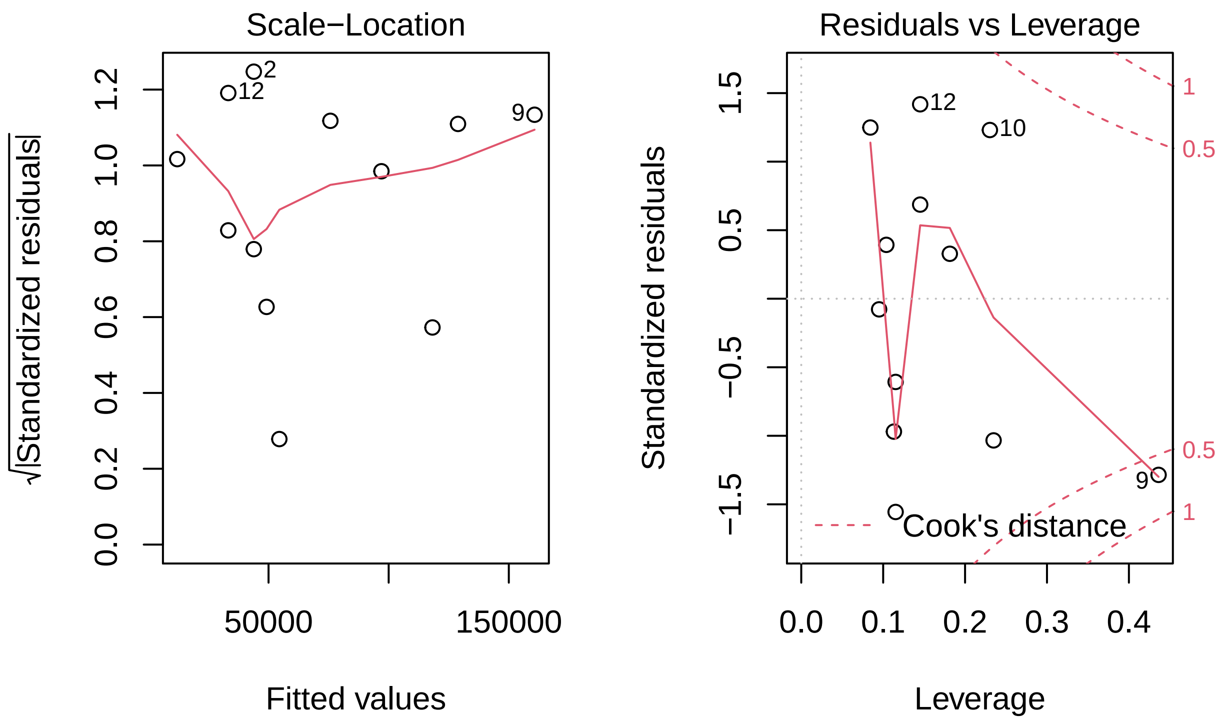
```
#8. Capture the summary of the linear model
modelsummary
```

```
##
## Call:
## lm(formula = Sales ~ Spend, data = data)
##
## Residuals:
##    Min      1Q Median      3Q     Max
```

```
##   -3385  -2097    258   1726   3034
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1383.4714  1255.2404   1.102    0.296
## Spend         10.6222     0.1625  65.378 1.71e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2313 on 10 degrees of freedom
## Multiple R-squared:  0.9977, Adjusted R-squared:  0.9974
## F-statistic:  4274 on 1 and 10 DF,  p-value: 1.707e-14
```

```
#9. Also perform the Linear Diagnostics for the given data set(Hint: plot(lmmodel))
plot(model)
```





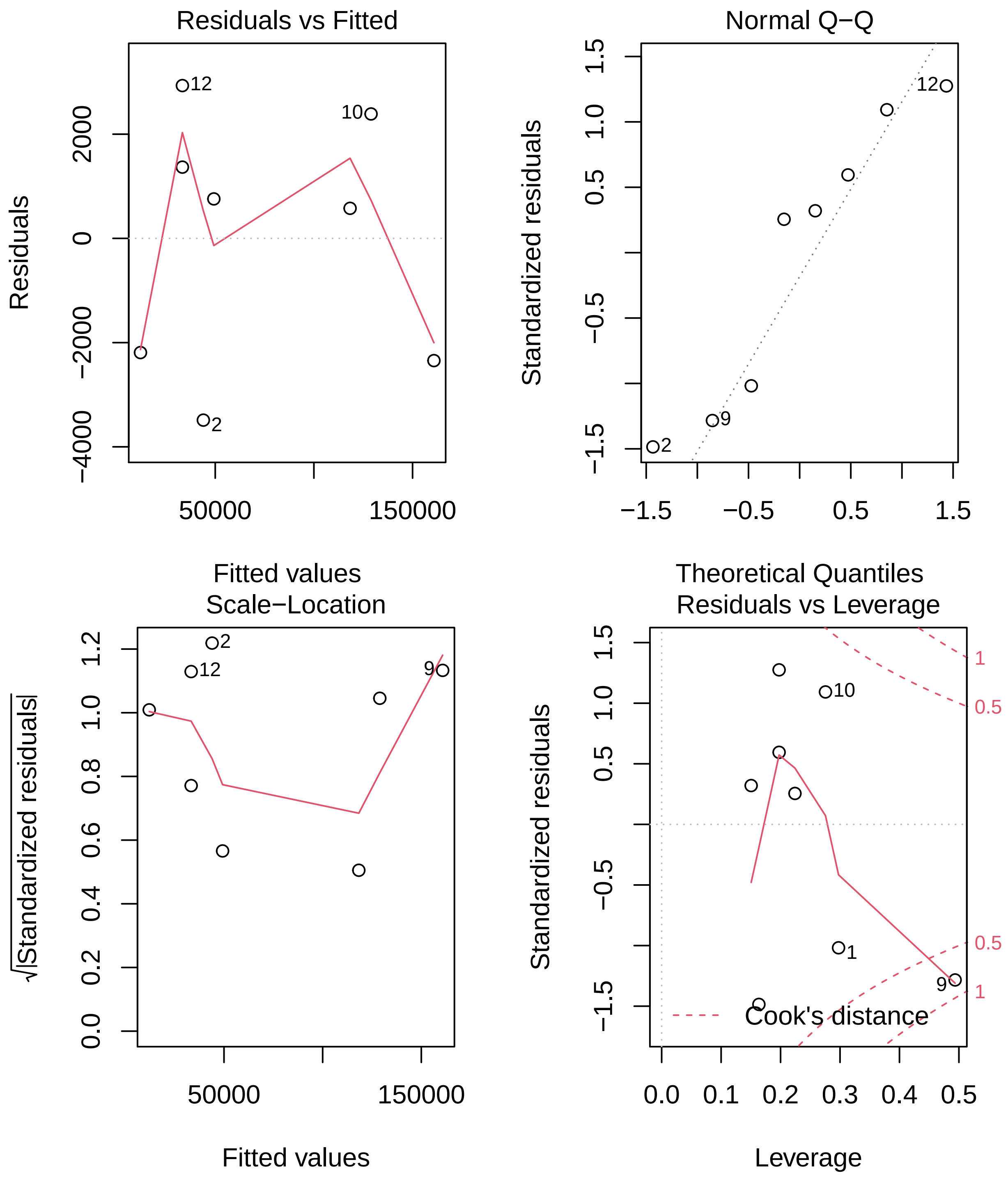

```
#10. Create the training and test data (70:30)
set.seed(11)
indexes = sample(nrow(data), nrow(data)*.7)
train <- data[indexes,]
test <- data[-indexes,]

#11. Fit the model on training data and predict sales on test data
model <- lm(Sales~Spend, data=train)
predict(model)
```

```
##        10         2         8         9        12         1         5         4
## 128959.82  43974.49 118336.65 160829.32  33351.33  12104.99  33351.33  49286.07
```

```
#We can see the values predicted by the model. On comparing a value at an index with the observed value


#12. Review the diagnostic measures
plot(model)
```



Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance