Homework Chapter 4

Jinhong Du 15338039

4.4 Refer to Airfreight breakage Problem 1.21.

(b) Obtain Bonferroni joint confidence intervals for β_0 and β_1, using a 99 percent family confidence coefficient. Interpret your confidence intervals.

The $100(1-\alpha)\%$ Bonferroni joint confidence intervals for joint estimation of β_0, β_1 is given by

$$(b_0 - Bs\{b_0\}, b_0 + Bs\{b_0\})$$
$$(b_1 - Bs\{b_1\}, b_1 + Bs\{b_1\})$$

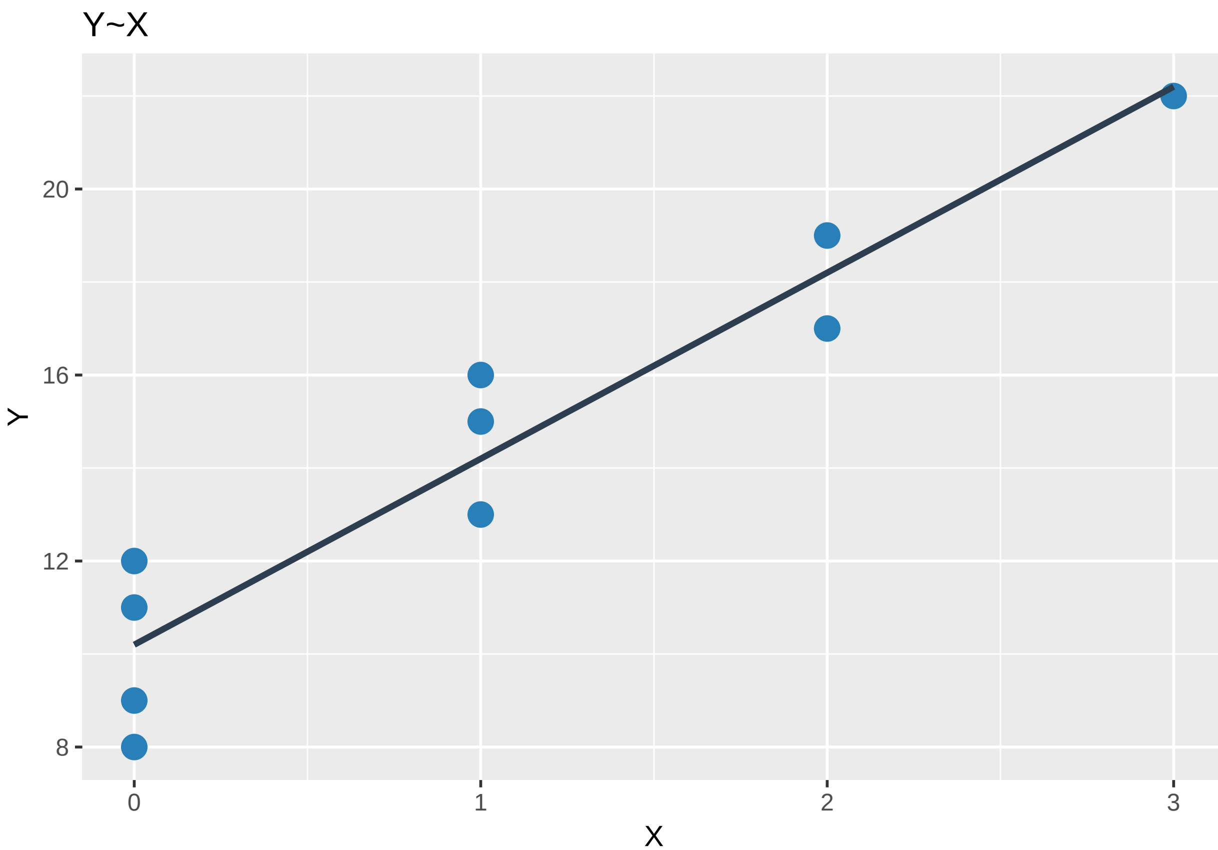
where

$$B = t\left(1 - \frac{\alpha}{4}; n-2\right)$$

```
library(ggplot2)
library(gridExtra)
data1 <- read.table("CH01PR21.txt",head=FALSE,col.names = c('Y','X'))
X = data1$X
Y = data1$Y
fit <- lm('Y~X',data1)
summary(fit)
```

```
##
## Call:
## lm(formula = "Y~X", data = data1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -2.2   -1.2    0.3    0.8    1.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.2000     0.6633  15.377 3.18e-07 ***
## X             4.0000     0.4690   8.528 2.75e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.483 on 8 degrees of freedom
## Multiple R-squared:  0.9009, Adjusted R-squared:  0.8885
## F-statistic: 72.73 on 1 and 8 DF,  p-value: 2.749e-05
```

```
lm.scatter <- ggplot(data1, aes(x=X, y=Y)) +
  geom_point(color='#2980B9', size = 4) + xlim(c(0, 3)) +
  geom_smooth(method = lm, se=FALSE, fullrange=TRUE, color='#2C3E50', size=1.1) +
  labs(title='Y~X')
grid.arrange(lm.scatter)
```




```r
df = fit$df.residual
n = length(X)
mse <- sum((Y - fit$fitted.values)^2) / (n - 2)
sb1 <- sqrt(mse/sum((X-mean(X))^2))
sb0 <-sqrt(mse*(1/n+mean(X)^2/sum((X-mean(X))^2)))
b0 <- fit$coefficients[1]
b1 <- fit$coefficients[2]
B <- qt(1-0.01/(2 * 2), df)
print(sprintf("B = %f",B))
```

```
## [1] "B = 3.832519"
```

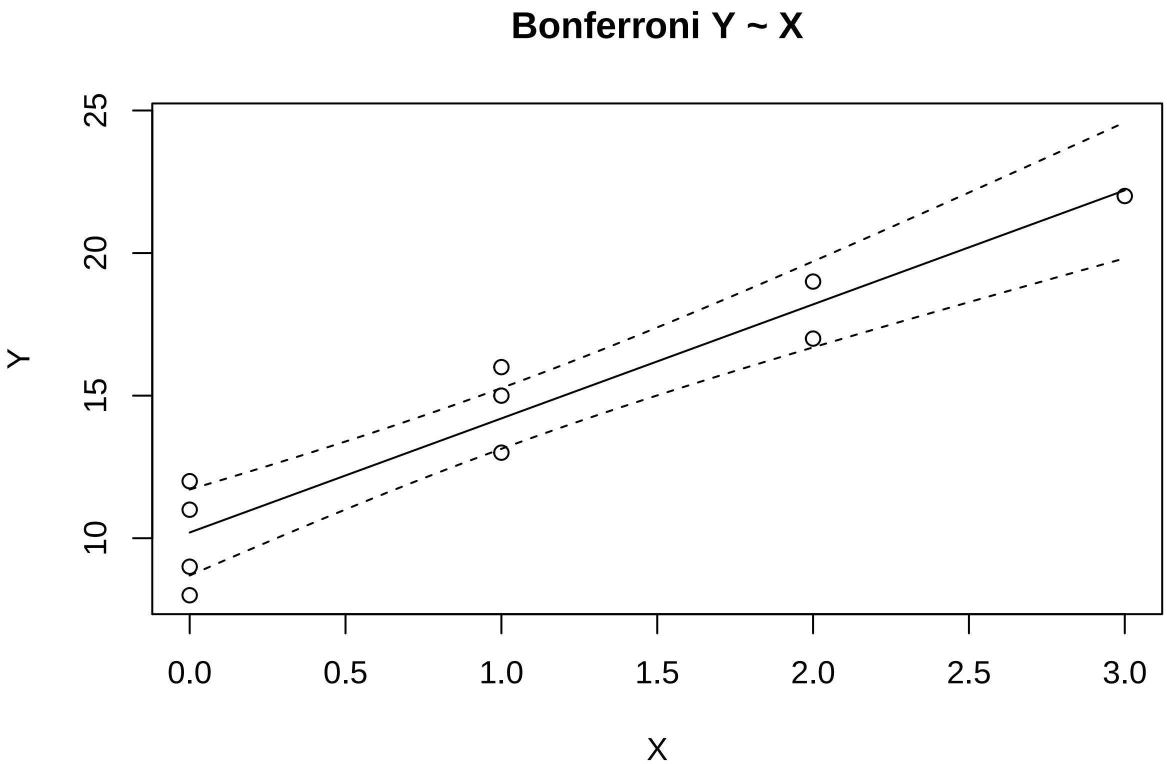
```r
print(sprintf("The confidence interval for beta0 is (%f,%f)",b0-B*sb0,b0+B*sb0))
```

```
## [1] "The confidence interval for beta0 is (7.657795,12.742205)"
```

```r
print(sprintf("The confidence interval for beta1 is (%f,%f)",b1-B*sb1,b1+B*sb1))
```

```
## [1] "The confidence interval for beta1 is (2.202389,5.797611)"
```

```r
library(investr)
plotFit(fit, interval = 'confidence', k = 0.95, adjust = 'Bonferroni',
        main = 'Bonferroni Y ~ X')
```



4.8 Refer to Airfreight breakage Problem 1.21.

(a) It is desired to obtain interval estimates of the mean number of broken ampules when there are 0,1 and 2 transfers for a shipment, using a 95 percent family confidence coefficient. Obtain the desired confidence intervals, using the Working-Hotelling procedure.

The $100(1-\alpha)\%$ Working-Hotelling confidence intervals for mean response $\mathbb{E}Y_h$ is given by

$$(\hat{Y}_h - Ws\{\hat{Y}_h\}, \hat{Y}_h + Ws\{\hat{Y}_h\})$$

where

$$W = \sqrt{2F(1-\alpha; 2, n-2)}$$

```
W <- sqrt(2 * qf(p = 0.95, df1 = 2, df2 = n - 2))
print(sprintf("W = %f",W))
```

```
## [1] "W = 2.986292"
```

```
for (i in c(0:2)){
  Xh <- data.frame(X=i)
  Yh <- predict(fit,Xh)
  sYh <- sqrt(mse*(1/n+(Xh-mean(X))^2/sum((X-mean(X))^2)))
  print(
    sprintf(
      "The Working-Hotelling confidence interval for Xh=%d is (%f,%f)",
      i,Yh-W*sYh,Yh+W*sYh))
}
```

```
## [1] "The Working-Hotelling confidence interval for Xh=0 is (8.219118,12.180882)"
## [1] "The Working-Hotelling confidence interval for Xh=1 is (12.799305,15.600695)"
## [1] "The Working-Hotelling confidence interval for Xh=2 is (16.219118,20.180882)"
```

(b) Are the confidence intervals obtained in part (a) more efficient than Bonferroni intervals here? Explain.

The $100(1-\alpha)\%$ Bonferroni confidence intervals for mean response $\mathbb{E}Y_h$ is given by

$$(\hat{Y}_h - Bs\{\hat{Y}_h\}, \hat{Y}_h + Bs\{\hat{Y}_h\})$$

where

$$B = t\left(1 - \frac{\alpha}{4}; n-2\right)$$

```
B <- qt(1-0.05/(2 * 3), df)
print(sprintf("B = %f",B))
```

```
## [1] "B = 3.015762"
```

```
for (i in c(0:2)){
  Xh <- data.frame(X=i)
  Yh <- predict(fit,Xh)
  sYh <- sqrt(mse*(1/n+(Xh-mean(X))^2/sum((X-mean(X))^2)))
  print(
    sprintf(
      "The Working-Hotelling confidence interval for Xh=%d is (%f,%f)",
      i,Yh-B*sYh,Yh+B*sYh))
}
```

```
## [1] "The Working-Hotelling confidence interval for Xh=0 is (8.199570,12.200430)"
## [1] "The Working-Hotelling confidence interval for Xh=1 is (12.785482,15.614518)"
## [1] "The Working-Hotelling confidence interval for Xh=2 is (16.199570,20.200430)"
```

The confidence intervals obtained in part (a) are more efficient than Bonferroni intervals here since $B > W$.

4.16 Refer to Copier maintenance Problem 1.20.

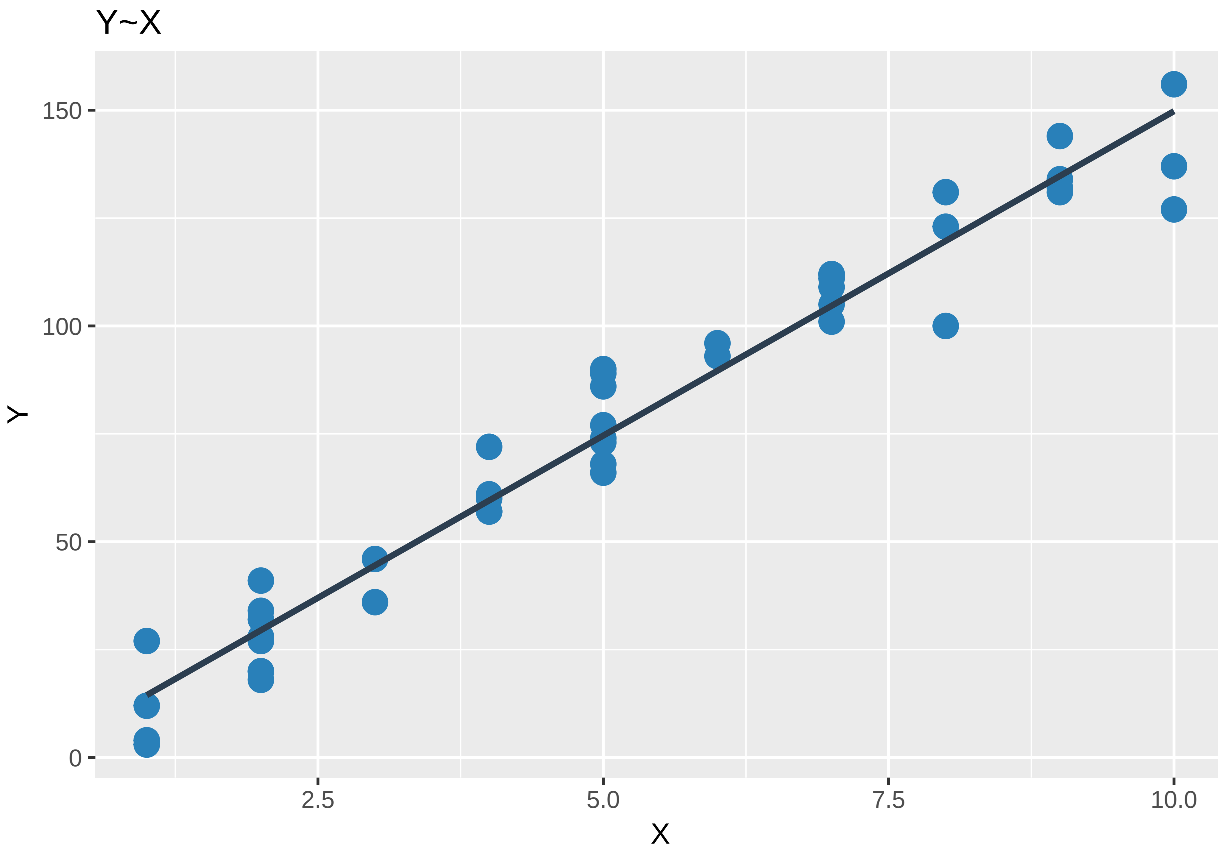
(a) Obtain the estimated regression function.

```
data2 <- read.table("CH01PR20.txt",head=FALSE,col.names = c('Y','X'))
X2 <- data2$X
Y2 <- data2$Y
fit2 <- lm('Y~X',data2)
summary(fit2)
```

```
##
## Call:
## lm(formula = "Y~X", data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.7723  -3.7371   0.3334   6.3334  15.4039
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5802     2.8039  -0.207    0.837
## X            15.0352     0.4831  31.123   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.914 on 43 degrees of freedom
```

```
## Multiple R-squared:  0.9575, Adjusted R-squared:  0.9565
## F-statistic: 968.7 on 1 and 43 DF,  p-value: < 2.2e-16
```

```r
lm.scatter <- ggplot(data2, aes(x=X, y=Y)) +
  geom_point(color='#2980B9', size = 4)+
  geom_smooth(method = lm, se=FALSE, fullrange=TRUE, color='#2C3E50', size=1.1) +
  labs(title='Y~X')
grid.arrange(lm.scatter)
```



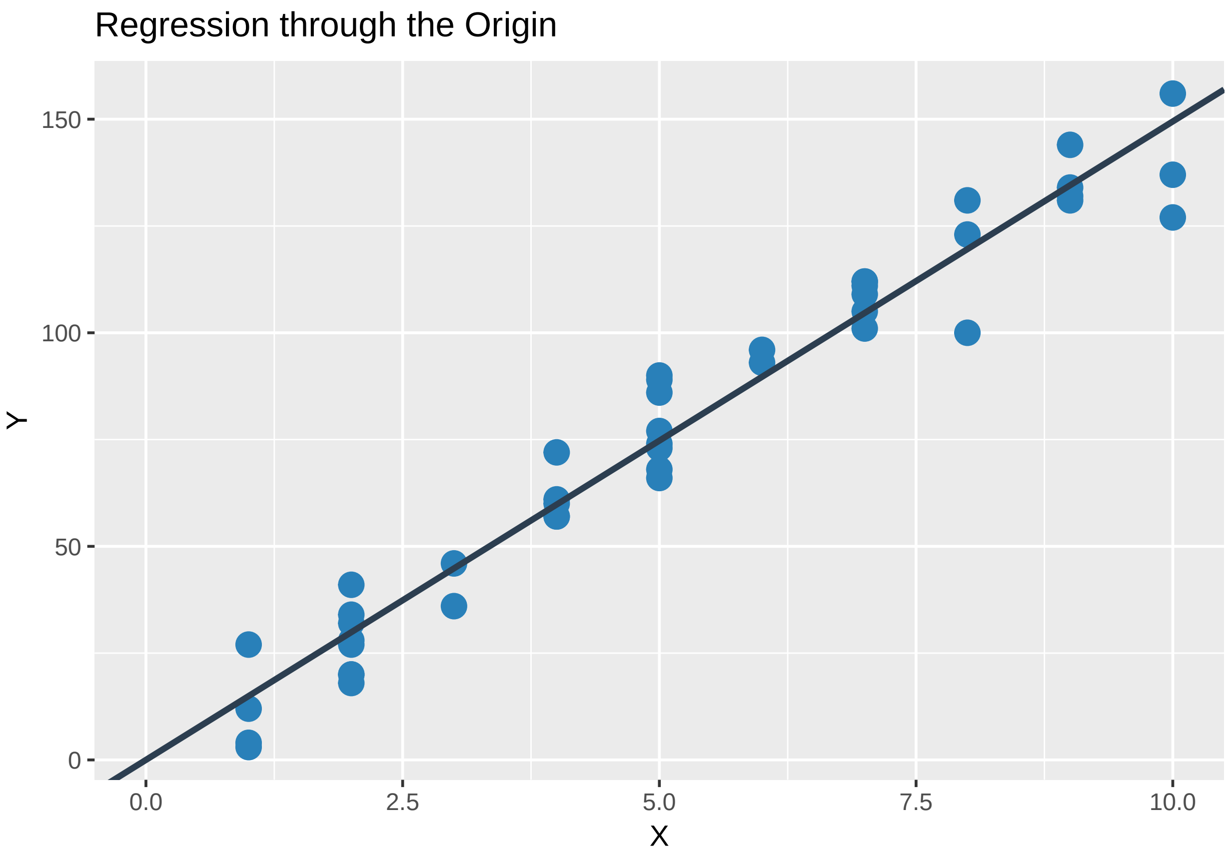
Since P-value of testing $H_0 : \beta_0 = 0 \quad H_a : \beta_0 \neq 0$ is 0.837, accept H_0. Therefore, we should do the regression through origin.

```r
fit2 <- lm('Y~0+X',data2)
summary(fit2)
```

```
##
## Call:
## lm(formula = "Y~0+X", data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.4723  -3.6306   0.2111   6.3694  15.2639
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## X  14.9472     0.2264   66.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 8.816 on 44 degrees of freedom
## Multiple R-squared:   0.99,  Adjusted R-squared:  0.9898
## F-statistic:  4358 on 1 and 44 DF,  p-value: < 2.2e-16
```

```
lm.scatter <- ggplot(data2, aes(x=X, y=Y)) +
  geom_point(color='#2980B9', size = 4) + xlim(c(0, 10)) +
  geom_abline(intercept=0, slope=fit2$coefficients[1], color='#2C3E50', size=1.1) +
  labs(title='Regression through the Origin')
grid.arrange(lm.scatter)
```



the regression function is

$$y = 15.0352x$$

(b) Estimate β_1, with a 90 percent confidence interval. Interpret your interval estrmate.

The $1 - \alpha$ confidence interval of β_1 is given by

$$\left(b_1 - t\left(1 - \frac{\alpha}{2}\right) s\{b_0\}, b_0 + t\left(1 - \frac{\alpha}{2}\right) s\{b_0\}\right)$$

where

$$s\{b_1\} = \sqrt{\frac{MSE}{SS_{XX}}}$$

```
df2 = fit2$df.residual
n2 = length(X2)
mse2 <- sum((Y2 - fit2$fitted.values)^2) / df2
sb12 <- sqrt(mse2/sum(X2^2))
b1 <- fit2$coefficients[1]
```

```r
t = qt(p = 0.95, df = df2)
print(sprintf("The confidence interval for beta1 is (%f,%f)",b1-t*sb12,b1+t*sb12))
```

```
## [1] "The confidence interval for beta1 is (14.566785,15.327674)"
```

(c) Predict the service time on a new call in which six copiers are to be serviced. Use a 90 percent prediction interval.

The $1-\alpha$ prediction interval of $Y_{h(new)}$ is given by

$$\left(\hat{Y}_h - t\left(1-\frac{\alpha}{2};n-1\right)s\{pred\}, \hat{Y}_h + t\left(1-\frac{\alpha}{2};n-1\right)s\{pred\}\right)$$

where

$$s\{pred\} = \sqrt{MSE\left(1+\frac{X_h^2}{SS_{XX}}\right)}$$

```r
Xh <- data.frame(X=6)
pred_Yh <- predict(fit2,Xh)
s_pred <- sqrt(mse2*(1+Xh^2/sum(X2^2)))
print(sprintf("The prediction interval for Yh(new) is (%f,%f)",pred_Yh-t*s_pred,pred_Yh+t*s_pred))
```

```
## [1] "The prediction interval for Yh(new) is (74.695586,104.671168)"
```

4.17 Refer to Copier maintenance Problem 4.16.

(c) Conduct a formal test for lack of fit of linear regression through the origin; use $\alpha = .01$. State the alternatives, decision rule, and conclusion. What is the P-value of the test?

$$H_0 : \mathbb{E}Y = \beta_1 X \qquad H_a : \mathbb{E}Y \neq \beta_1 X$$

The decision rule is: If $F^* \leqslant 2.963012$, then conclude H_0, otherwise conclude H_a.

Here, $F^* = 0.864779$, conclude H_0.

$$\begin{aligned} Pvalue &= \mathbb{P}\{H_a \text{ holds}\} \\ &= \mathbb{P}\{F(9,35) > F^*\} \\ &= 1 - \mathbb{P}\{F(9,35) \leqslant F^*\} \\ &= 0.564434 \end{aligned}$$

```r
level = length(unique(data2$X))
n2 = length(data2$X)
SSER = sum(fit2$residuals^2)
SSEF = 0
for (i in unique(data2$X)) {
  SSEF <- SSEF + sum((data2[data2$X==i,]$Y-mean(data2[data2$X==i,]$Y))^2)
}
Fvalue = (SSER-SSEF)/(level-n2+df2)/(SSEF/(n2-level))
Pvalue = 1-pf(Fvalue,df1=level-n2+df2,df2=n2-level)
print(sprintf('SSE of Reduced Model :%f',SSER))
```

```
## [1] "SSE of Reduced Model :3419.778364"
```

```r
print(sprintf('SSE of Full Model    :%f',SSEF))
```

```
## [1] "SSE of Full Model    :2797.658333"
```

```r
print(sprintf('F-value              :%f',Fvalue))
```

```
## [1] "F-value              :0.864779"
```

```r
print(sprintf('0.99 Quantile F(%d,%d) value:%f',
              level-n2+df2,n2-level,qf(0.99,level-n2+df2,n2-level)))
```

```
## [1] "0.99 Quantile F(9,35) value:2.963012"
```

```r
print(sprintf('P-value              :%f',Pvalue))
```

```
## [1] "P-value              :0.564434"
```