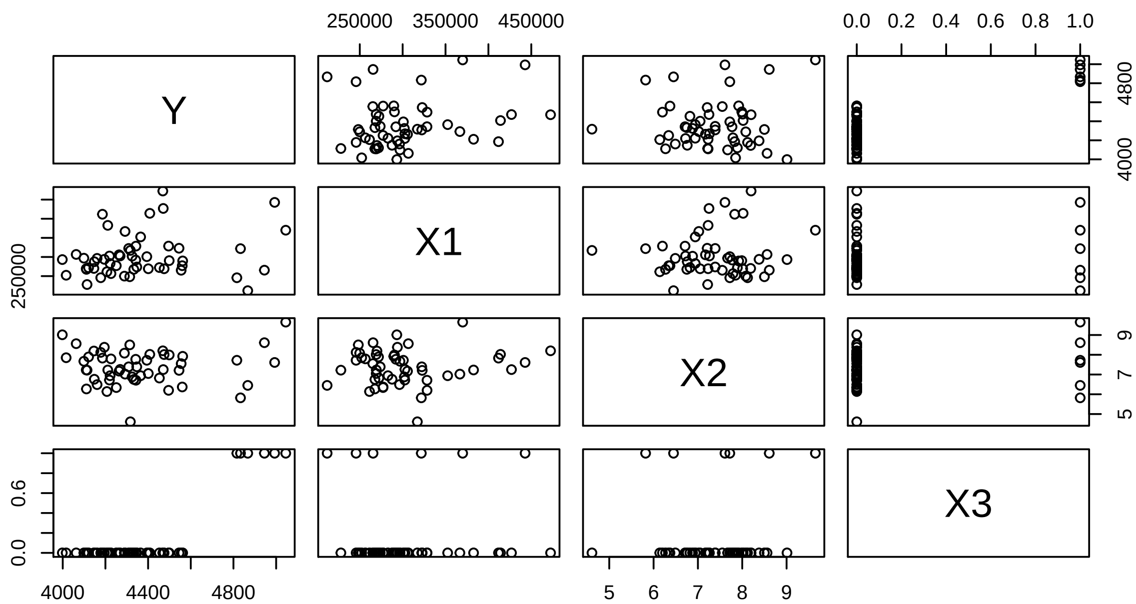
Appendix

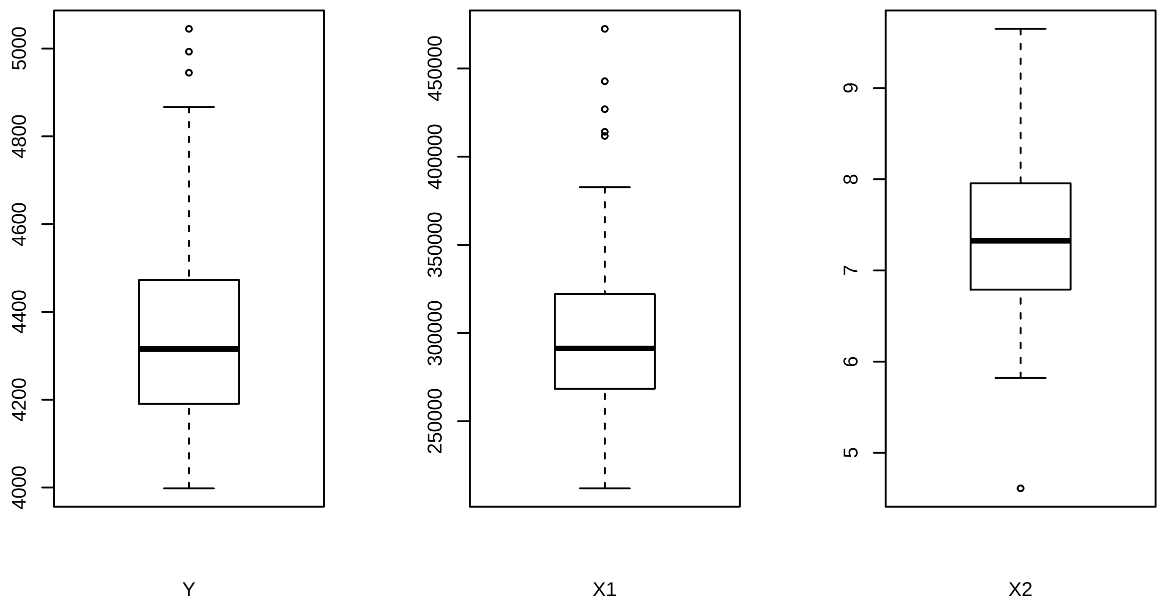
(Includes R codes used in the paper)

Problem 1:

```
# Importing the Data
datahw4 <- read.csv("D:/Vtech/Regression and ANOVA/PS4/datahw4.txt",
    sep = "")
# Scatterplot Matrix and Boxplot
pairs(datahw4)
```



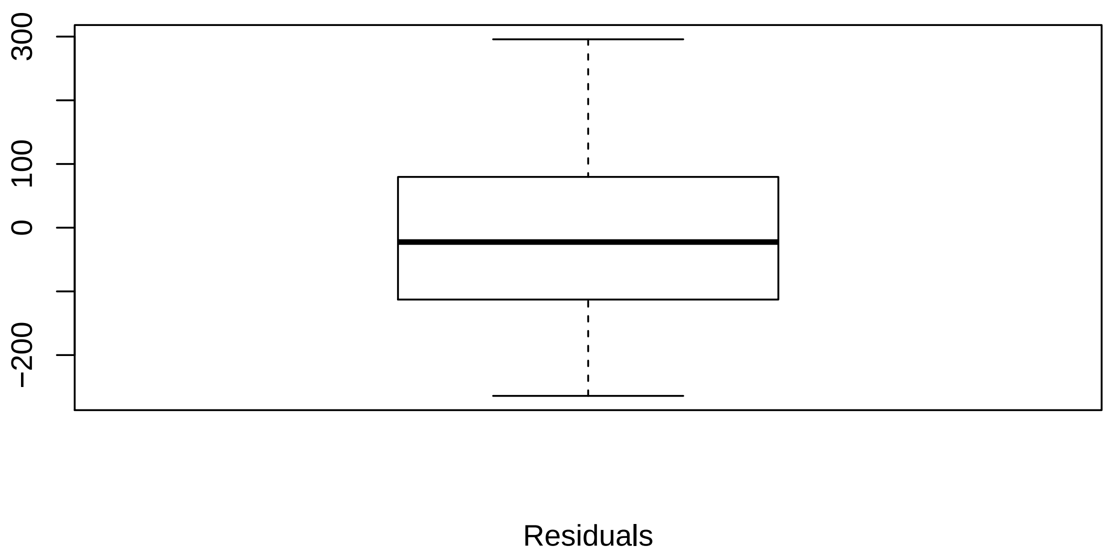

```
par(mfrow = c(1, 3))
boxplot(datahw4$Y, xlab = "Y")
boxplot(datahw4$X1, xlab = "X1")
boxplot(datahw4$X2, xlab = "X2")
```



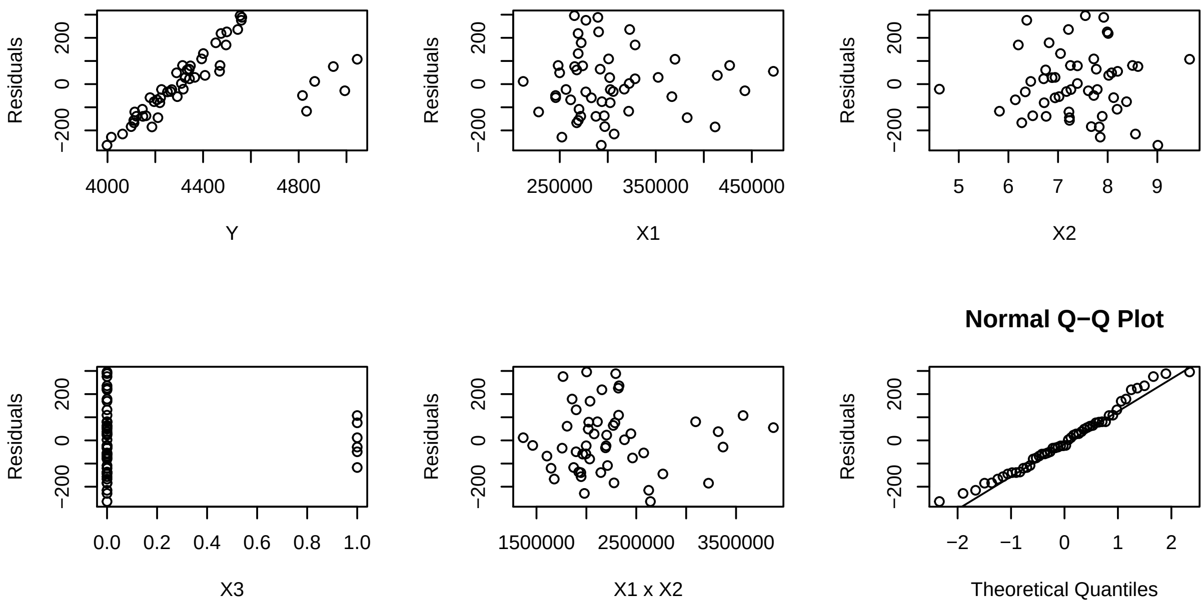

```
# setting up the linear model
X <- cbind(rep(x = 1, 52), datahw4$X1, datahw4$X2, datahw4$X3)
y <- datahw4$Y
n <- length(y)
xtxi <- solve(t(X) %*% X)
beta <- xtxi %*% t(X) %*% y
vhat <- (t(y - X %*% beta) %*% (y - X %*% beta))/(n - length(xtxi[,
    1]))
Vbeta <- as.numeric(vhat) * xtxi
SEbeta <- c(sqrt(Vbeta[1, 1]), sqrt(Vbeta[2, 2]), sqrt(Vbeta[3, 3]),
    sqrt(Vbeta[4, 4]))
y_hat <- beta[1] * X[, 1] + X[, 2] * beta[2] + X[, 3] * beta[3] +
    X[, 4] * beta[4]
res_vec <- y - y_hat
# ANOVA table for linear regression
anv_mat <- data.frame(matrix(NA, nrow = 2, ncol = 6))
names(anv_mat) <- c("", "Df", "Sum Sq", "Mean Sq", "F-Value", "Pr(>F)")
anv_mat[1, 1] <- "x"
anv_mat[2, 1] <- "Residuals"
anv_mat[1, 2] <- length(xtxi[, 1]) - 1
anv_mat[2, 2] <- n - length(xtxi[, 1])
anv_mat[1, 3] <- t(y - mean(y)) %*% (y - mean(y)) - t(y - X %*% beta) %*%
    (y - X %*% beta)
anv_mat[2, 3] <- t(y - X %*% beta) %*% (y - X %*% beta)
anv_mat[1, 4] <- anv_mat[1, 3]/anv_mat[1, 2]
anv_mat[2, 4] <- anv_mat[2, 3]/anv_mat[2, 2]
anv_mat[1, 5] <- round(anv_mat[1, 4]/anv_mat[2, 4], 3)
anv_mat[2, 5] <- ""
anv_mat[1, 6] <- round(1 - pf(as.numeric(anv_mat[1, 5]), anv_mat[1,
    2], anv_mat[2, 2]), 4)
anv_mat[2, 6] <- ""
```

```r
rsq <- anv_mat[1, 3]/(t(y - mean(y)) %*% (y - mean(y)))
tbl_anv <- (kable(anv_mat))
tbl_anv <- kable_styling(tbl_anv, position = "center")
tbl_anv <- column_spec(tbl_anv, 1, border_left = T)
tbl_anv <- column_spec(tbl_anv, 6, border_right = T)
```

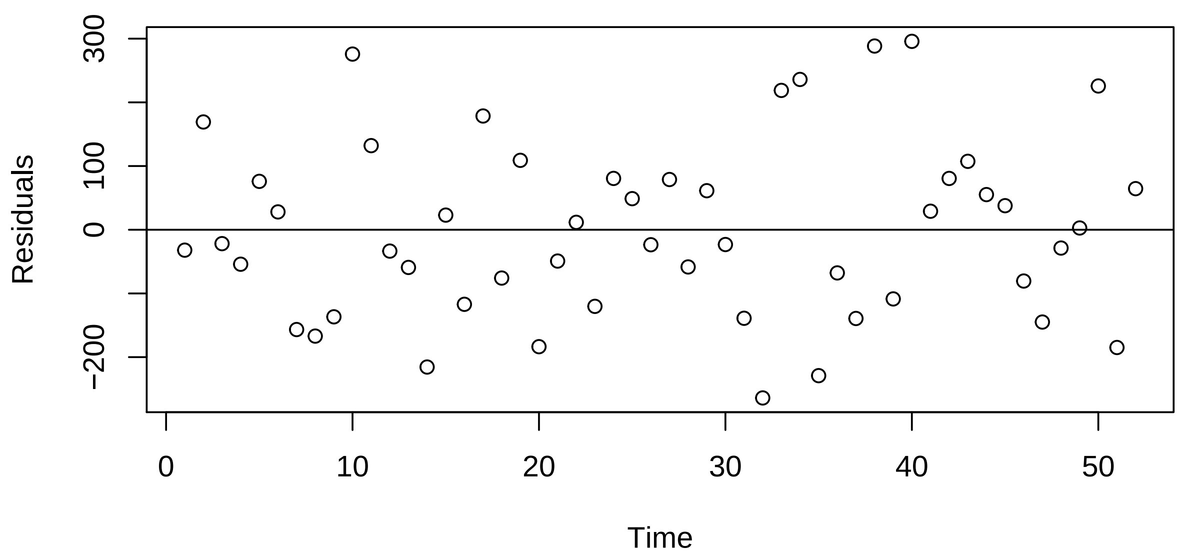
```r
# Residual boxplot
boxplot(res_vec, xlab = "Residuals")
```




```r
par(mfrow = c(2, 3))
# Residual plots
plot(datahw4$Y, res_vec, xlab = "Y", ylab = "Residuals")
plot(datahw4$X1, res_vec, xlab = "X1", ylab = "Residuals")
plot(datahw4$X2, res_vec, xlab = "X2", ylab = "Residuals")
plot(datahw4$X3, res_vec, xlab = "X3", ylab = "Residuals")
plot(datahw4$X1 * datahw4$X2, res_vec, xlab = "X1 x X2", ylab = "Residuals")
# Normal QQ plot for the residuals
qqnorm(res_vec, ylab = "Residuals")
qqline(res_vec)
```




```r
# Residual Time Plot
plot(c(1:n), res_vec, xlab = "Time", ylab = "Residuals")
abline(h = 0)
```




```r
# The Brown-forsythe test for constant variance
res_mat <- data.frame(cbind(y, X, y_hat, res_vec))
res_mat <- arrange(res_mat, y_hat)
```

```r
res_vec_low <- res_mat$res_vec[1:26]
res_vec_high <- res_mat$res_vec[27:52]
rvl_mat <- data.frame(cbind(res_vec_low, 1))
names(rvl_mat) <- c("residuals", "group")
rvh_mat <- data.frame(cbind(res_vec_high, 2))
names(rvh_mat) <- c("residuals", "group")
rv_mat <- rbind(rvl_mat, rvh_mat)
rv_mat$group <- as.factor(rv_mat$group)
ng1 <- length(res_vec_low)
ng2 <- length(res_vec_high)
mrl <- median(res_vec_low)
mrh <- median(res_vec_high)
dg1 <- abs(res_vec_low - mrl)
dg2 <- abs(res_vec_high - mrh)
sbf <- (sum((dg1 - mean(dg1))^2) + sum((dg2 - mean(dg2))^2))/(n -
    length(xtxi[, 1]))
tbf <- (mean(dg1) - mean(dg2))/(sqrt(sbf) * sqrt(1/ng1 + 1/ng2))
bftstatp <- pt(abs(tbf), n - length(xtxi[, 1]), lower.tail = F) *
    2
```

```r
# t-test for linear model slope coefficient for Beta1
t <- as.numeric(beta[2])/(SEbeta[2])
tvcrit <- qt(0.975, n - length(xtxi[, 1]))
tstatp <- round(pt(abs(t), df = n - length(xtxi[, 1]), lower.tail = F) *
    2, 5)
```

```r
# t-test for linear model slope coefficient for Beta2
t <- as.numeric(beta[3])/(SEbeta[3])
tvcrit <- qt(0.975, n - length(xtxi[, 1]))
tstatp <- round(pt(abs(t), df = n - length(xtxi[, 1]), lower.tail = F) *
    2, 5)
```

```r
# t-test for linear model slope coefficient for Beta3
t <- as.numeric(beta[4])/(SEbeta[4])
tvcrit <- qt(0.975, n - length(xtxi[, 1]))
tstatp <- round(pt(abs(t), df = n - length(xtxi[, 1]), lower.tail = F) *
    2, 5)
```

```r
# ANOVA table for linear regression
anv_mat <- data.frame(matrix(NA, nrow = 2, ncol = 6))
names(anv_mat) <- c("", "Df", "Sum Sq", "Mean Sq", "F-Value", "Pr(>F)")
anv_mat[1, 1] <- "x"
anv_mat[2, 1] <- "Residuals"
anv_mat[1, 2] <- length(xtxi[, 1]) - 1
anv_mat[2, 2] <- n - length(xtxi[, 1])
anv_mat[1, 3] <- t(y - mean(y)) %*% (y - mean(y)) - t(y - X %*% beta) %*%
    (y - X %*% beta)
anv_mat[2, 3] <- t(y - X %*% beta) %*% (y - X %*% beta)
anv_mat[1, 4] <- anv_mat[1, 3]/anv_mat[1, 2]
anv_mat[2, 4] <- anv_mat[2, 3]/anv_mat[2, 2]
anv_mat[1, 5] <- round(anv_mat[1, 4]/anv_mat[2, 4], 3)
anv_mat[2, 5] <- ""
anv_mat[1, 6] <- round(1 - pf(as.numeric(anv_mat[1, 5]), anv_mat[1,
    2], anv_mat[2, 2]), 4)
anv_mat[2, 6] <- ""
```

```r
tbl_anv <- (kable(anv_mat))
tbl_anv <- kable_styling(tbl_anv, position = "center")
tbl_anv <- column_spec(tbl_anv, 1, border_left = T)
tbl_anv <- column_spec(tbl_anv, 6, border_right = T)
```

```r
# family confidence intervals, Bonferroni adjusted
tv <- qt((1 - (0.05/(2 * (length(xtxi[, 1]) - 1)))), n - length(xtxi[,
    1]))
cilow.beta0 <- beta[1] - tv * SEbeta[1]
ciupp.beta0 <- beta[1] + tv * SEbeta[1]
cilow.beta1 <- beta[2] - tv * SEbeta[2]
ciupp.beta1 <- beta[2] + tv * SEbeta[2]
cilow.beta2 <- beta[3] - tv * SEbeta[3]
ciupp.beta2 <- beta[3] + tv * SEbeta[3]
cilow.beta3 <- beta[4] - tv * SEbeta[4]
ciupp.beta3 <- beta[4] + tv * SEbeta[4]
cnfint <- data.frame()
cnfint[1, 1] <- cilow.beta0
cnfint[2, 1] <- ciupp.beta0
cnfint[1, 2] <- cilow.beta1
cnfint[2, 2] <- ciupp.beta1
cnfint[1, 3] <- cilow.beta2
cnfint[2, 3] <- ciupp.beta2
cnfint[1, 4] <- cilow.beta3
cnfint[2, 4] <- ciupp.beta3
colnames(cnfint) <- c("b0", "b1", "b2", "b3")
tbl_anv <- (kable(cnfint))
tbl_anv <- kable_styling(tbl_anv, position = "center")
tbl_anv <- column_spec(tbl_anv, 1, border_left = T)
tbl_anv <- column_spec(tbl_anv, 4, border_right = T)
```

```r
# ANOVA table for extra sum of squares
extss_mat <- data.frame(matrix(NA, nrow = 2, ncol = 5))
names(extss_mat) <- c("", "df.R - df.F", "Extra Sum Sq", "F-Value",
    "Pr(>F)")
extss_mat[1, 1] <- "Total Model"
extss_mat[1, 2] <- length(xtxi[, 1]) - 1
# RSS for total model
rss_b0b1b2b3 <- t(y - X %*% beta) %*% (y - X %*% beta)
dfb0b1b2b3 <- n - length(xtxi[, 1])

# RSS for b0 model
xtxi <- solve(t(X[, 1]) %*% X[, 1])
beta <- xtxi %*% t(X[, 1]) %*% y
rss_b0 <- t(y - X[, 1] %*% beta) %*% (y - X[, 1] %*% beta)
dfb0 <- n - length(xtxi[, 1])

# RSS for b1 model
xtxi <- solve(t(X[, 1:2]) %*% X[, 1:2])
beta <- xtxi %*% t(X[, 1:2]) %*% y
rss_b0b1 <- t(y - X[, 1:2] %*% beta) %*% (y - X[, 1:2] %*% beta)
dfb0b1 <- n - length(xtxi[, 1])

# RSS for b2 model
```

```r
xtxi <- solve(t(X[, c(1, 3)]) %*% X[, c(1, 3)])
beta <- xtxi %*% t(X[, c(1, 3)]) %*% y
rss_b0b2 <- t(y - X[, c(1, 3)] %*% beta) %*% (y - X[, c(1, 3)] %*%
    beta)
dfb0b2 <- n - length(xtxi[, 1])

# RSS for b1,b3 model
xtxi <- solve(t(X[, c(1, 2, 4)]) %*% X[, c(1, 2, 4)])
beta <- xtxi %*% t(X[, c(1, 2, 4)]) %*% y
rss_b0b1b3 <- t(y - X[, c(1, 2, 4)] %*% beta) %*% (y - X[, c(1, 2,
    4)] %*% beta)
dfb0b1b3 <- n - length(xtxi[, 1])

# RSS for b1,b2 model
xtxi <- solve(t(X[, c(1, 2, 3)]) %*% X[, c(1, 2, 3)])
beta <- xtxi %*% t(X[, c(1, 2, 3)]) %*% y
rss_b0b1b2 <- t(y - X[, c(1, 2, 3)] %*% beta) %*% (y - X[, c(1, 2,
    3)] %*% beta)
dfb0b1b2 <- n - length(xtxi[, 1])

# RSS for b1,b3,b2 model
xtxi <- solve(t(X) %*% X)
beta <- xtxi %*% t(X) %*% y
rss_b0b1b2b3 <- t(y - X %*% beta) %*% (y - X %*% beta)
dfb0b1b2b3 <- n - length(xtxi[, 1])

extss_mat[2, 1] <- "X1"
extss_mat[3, 1] <- "X3|X1"
extss_mat[4, 1] <- "X2|X1,X3"
extss_mat[5, 1] <- "Total"

extss_mat[1, 2] <- dfb0 - dfb0b1b2b3
extss_mat[2, 2] <- dfb0 - dfb0b1
extss_mat[3, 2] <- dfb0b1 - dfb0b1b3
extss_mat[4, 2] <- dfb0b1b3 - dfb0b1b2b3
extss_mat[5, 2] <- n - 1

extss_mat[1, 3] <- round(rss_b0 - rss_b0b1b2b3, 0)
extss_mat[2, 3] <- round(rss_b0 - rss_b0b1, 0)
extss_mat[3, 3] <- round(rss_b0b1 - rss_b0b1b3, 0)
extss_mat[4, 3] <- round(rss_b0b1b3 - rss_b0b1b2b3, 0)
extss_mat[5, 3] <- round(t(y - mean(y)) %*% (y - mean(y)), 0)

extss_mat[1, 4] <- round((extss_mat[1, 3]/extss_mat[1, 2])/(rss_b0b1b2b3/dfb0b1b2b3),
    2)
extss_mat[2, 4] <- round((extss_mat[2, 3]/extss_mat[2, 2])/(rss_b0b1/dfb0b1),
    2)
extss_mat[3, 4] <- round((extss_mat[3, 3]/extss_mat[3, 2])/(rss_b0b1b3/dfb0b1b3),
    2)
extss_mat[4, 4] <- round((extss_mat[4, 3]/extss_mat[4, 2])/(rss_b0b1b2b3/dfb0b1b2b3),
    2)
extss_mat[5, 4] <- ""
```

```r
extss_mat[1, 5] <- round(1 - pf(as.numeric(extss_mat[1, 4]), extss_mat[1,
    2], dfb0b1b2b3), 4)
extss_mat[2, 5] <- round(1 - pf(as.numeric(extss_mat[2, 4]), extss_mat[2,
    2], dfb0b1), 4)
extss_mat[3, 5] <- round(1 - pf(as.numeric(extss_mat[3, 4]), extss_mat[3,
    2], dfb0b1b3), 4)
extss_mat[4, 5] <- round(1 - pf(as.numeric(extss_mat[4, 4]), extss_mat[4,
    2], dfb0b1b2b3), 4)
extss_mat[5, 5] <- ""

tbl_extss <- (kable(extss_mat))
tbl_extss <- kable_styling(tbl_extss, position = "center")
tbl_extss <- column_spec(tbl_extss, 1, border_left = T)
tbl_extss <- column_spec(tbl_extss, 5, border_right = T)
```

```r
# Calculating Coefficients of Determination

# Rsquared
rsq_mat <- data.frame(matrix(NA, nrow = 6, ncol = 1))
rownames(rsq_mat) <- c("X1", "X2", "X1,X2", "X1|X2", "X2|X1", "X1,X2,X3")
colnames(rsq_mat) <- c("R.sq")

# Rsq for b1 model
xtxi <- solve(t(X[, 1:2]) %*% X[, 1:2])
beta <- xtxi %*% t(X[, 1:2]) %*% y
rsq_b0b1 <- (t(y - X[, 1:2] %*% beta) %*% (y - X[, 1:2] %*% beta))/(t(y -
    mean(y)) %*% (y - mean(y)))

# Rsq for b2 model
xtxi <- solve(t(X[, c(1, 3)]) %*% X[, c(1, 3)])
beta <- xtxi %*% t(X[, c(1, 3)]) %*% y
rsq_b0b2 <- (t(y - X[, c(1, 3)] %*% beta) %*% (y - X[, c(1, 3)] %*%
    beta))/(t(y - mean(y)) %*% (y - mean(y)))

# Rsq for b1,b2 model
xtxi <- solve(t(X[, 1:3]) %*% X[, 1:3])
beta <- xtxi %*% t(X[, 1:3]) %*% y
rsq_b0b1b2 <- (t(y - X[, 1:3] %*% beta) %*% (y - X[, 1:3] %*% beta))/(t(y -
    mean(y)) %*% (y - mean(y)))

# Rsq for b1/b2 model
xtxi <- solve(t(X[, c(1, 2, 3)]) %*% X[, c(1, 2, 3)])
beta <- xtxi %*% t(X[, c(1, 2, 3)]) %*% y
rss_b0b1b2 <- t(y - X[, c(1, 2, 3)] %*% beta) %*% (y - X[, c(1, 2,
    3)] %*% beta)
rsq_b1cb2 <- (rss_b0b2 - rss_b0b1b2)/rss_b0b2

# Rsq for b2/b1 model
rsq_b2cb1 <- (rss_b0b1 - rss_b0b1b2)/rss_b0b1

rsq_mat[1, 1] <- rsq_b0b1
rsq_mat[2, 1] <- rsq_b0b2
rsq_mat[3, 1] <- rsq_b0b1b2
rsq_mat[4, 1] <- rsq_b1cb2
```

```r
rsq_mat[5, 1] <- rsq_b2cb1
rsq_mat[6, 1] <- rsq

tbl_rsq <- (kable(rsq_mat))
tbl_rsq <- kable_styling(tbl_rsq, position = "center")
tbl_rsq <- column_spec(tbl_rsq, 1, border_left = T)
tbl_rsq <- column_spec(tbl_rsq, 2, border_right = T)
```

```r
# setting up the standardized linear model
X <- cbind(((1/sqrt(n - 1)) * (datahw4$X1 - mean(datahw4$X1))/sd(datahw4$X1)),
    ((1/sqrt(n - 1)) * (datahw4$X2 - mean(datahw4$X2))/sd(datahw4$X2)),
    ((1/sqrt(n - 1)) * (datahw4$X3 - mean(datahw4$X3))/sd(datahw4$X3)))
y <- ((1/sqrt(n - 1)) * (datahw4$Y - mean(datahw4$Y))/sd(datahw4$Y))
xtxi <- solve(t(X) %*% X)
beta <- xtxi %*% t(X) %*% y
vhat <- (t(y - X %*% beta) %*% (y - X %*% beta))/(n - length(xtxi[,
    1]))
Vbeta <- as.numeric(vhat) * xtxi
SEbeta <- c(sqrt(Vbeta[1, 1]), sqrt(Vbeta[2, 2]), sqrt(Vbeta[3, 3]))
y_hat <- beta[1] * X[, 1] + X[, 2] * beta[2] + X[, 3] * beta[3]
res_vec <- y - y_hat
# ANOVA table for standardized linear regression
anv_mat <- data.frame(matrix(NA, nrow = 2, ncol = 6))
names(anv_mat) <- c("", "Df", "Sum Sq", "Mean Sq", "F-Value", "Pr(>F)")
anv_mat[1, 1] <- "x"
anv_mat[2, 1] <- "Residuals"
anv_mat[1, 2] <- length(xtxi[, 1]) - 1
anv_mat[2, 2] <- n - length(xtxi[, 1])
anv_mat[1, 3] <- t(y - mean(y)) %*% (y - mean(y)) - t(y - X %*% beta) %*%
    (y - X %*% beta)
anv_mat[2, 3] <- t(y - X %*% beta) %*% (y - X %*% beta)
anv_mat[1, 4] <- anv_mat[1, 3]/anv_mat[1, 2]
anv_mat[2, 4] <- anv_mat[2, 3]/anv_mat[2, 2]
anv_mat[1, 5] <- round(anv_mat[1, 4]/anv_mat[2, 4], 3)
anv_mat[2, 5] <- ""
anv_mat[1, 6] <- round(1 - pf(as.numeric(anv_mat[1, 5]), anv_mat[1,
    2], anv_mat[2, 2]), 4)
anv_mat[2, 6] <- ""
rsq <- anv_mat[1, 3]/(t(y - mean(y)) %*% (y - mean(y)))
tbl_anv <- (kable(anv_mat))
tbl_anv <- kable_styling(tbl_anv, position = "center")
tbl_anv <- column_spec(tbl_anv, 1, border_left = T)
tbl_anv <- column_spec(tbl_anv, 6, border_right = T)
```

```r
# Computing coefficients of partial determination between all
# pairs of predictor variables

# Rsquared
rssp_mat <- data.frame(matrix(NA, nrow = 3, ncol = 1))
rownames(rssp_mat) <- c("X1|X2,X3", "X2|X1,X3", "X3|X1,X2")
colnames(rssp_mat) <- c("R.sq")

# rssp for b2,b3 model
xtxi <- solve(t(X[, c(2, 3)]) %*% X[, c(2, 3)])
```

```r
beta <- xtxi %*% t(X[, c(2, 3)]) %*% y
rssp_b2b3 <- (t(y - X[, c(2, 3)] %*% beta) %*% (y - X[, c(2, 3)] %*%
    beta))/(t(y - mean(y)) %*% (y - mean(y)))

# rssp for b1,b3 model
xtxi <- solve(t(X[, c(1, 3)]) %*% X[, c(1, 3)])
beta <- xtxi %*% t(X[, c(1, 3)]) %*% y
rssp_b1b3 <- (t(y - X[, c(1, 3)] %*% beta) %*% (y - X[, c(1, 3)] %*%
    beta))/(t(y - mean(y)) %*% (y - mean(y)))

# rssp for b1,b2 model
xtxi <- solve(t(X[, c(1, 2)]) %*% X[, c(1, 2)])
beta <- xtxi %*% t(X[, c(1, 2)]) %*% y
rssp_b1b2 <- (t(y - X[, c(1, 2)] %*% beta) %*% (y - X[, c(1, 2)] %*%
    beta))/(t(y - mean(y)) %*% (y - mean(y)))

# rssp for b1,b2,b3 model
xtxi <- solve(t(X) %*% X)
beta <- xtxi %*% t(X) %*% y
rssp_b1b2b3 <- (t(y - X %*% beta) %*% (y - X %*% beta))/(t(y - mean(y)) %*%
    (y - mean(y)))

# rsqp
rsqp_b1cb2b3 <- (rssp_b2b3 - rssp_b1b2b3)/rssp_b2b3
rsqp_b2cb1b3 <- (rssp_b1b3 - rssp_b1b2b3)/rssp_b1b3
rsqp_b3cb1b2 <- (rssp_b1b2 - rssp_b1b2b3)/rssp_b1b2

rssp_mat[1, 1] <- rsqp_b1cb2b3
rssp_mat[2, 1] <- rsqp_b2cb1b3
rssp_mat[3, 1] <- rsqp_b3cb1b2

tbl_rssp <- (kable(rssp_mat))
tbl_rssp <- kable_styling(tbl_rssp, position = "center")
tbl_rssp <- column_spec(tbl_rssp, 1, border_left = T)
tbl_rssp <- column_spec(tbl_rssp, 2, border_right = T)
```

```r
# Transforming the standardized coefficients back to their
# original form
X <- cbind(rep(x = 1, 52), datahw4$X1, datahw4$X2, datahw4$X3)
y <- datahw4$Y
xtxi <- solve(t(X) %*% X)
beta <- xtxi %*% t(X) %*% y
beta_comp <- data.frame(matrix(NA, nrow = 3, ncol = 2))
beta_comp[1, 2] <- beta[2]
beta_comp[2, 2] <- beta[3]
beta_comp[3, 2] <- beta[4]

X <- cbind(((1/sqrt(n - 1)) * (datahw4$X1 - mean(datahw4$X1))/sd(datahw4$X1)),
    ((1/sqrt(n - 1)) * (datahw4$X2 - mean(datahw4$X2))/sd(datahw4$X2)),
    ((1/sqrt(n - 1)) * (datahw4$X3 - mean(datahw4$X3))/sd(datahw4$X3)))
y <- ((1/sqrt(n - 1)) * (datahw4$Y - mean(datahw4$Y))/sd(datahw4$Y))
xtxi <- solve(t(X) %*% X)
beta <- xtxi %*% t(X) %*% y
X <- cbind(rep(x = 1, 52), datahw4$X1, datahw4$X2, datahw4$X3)
```

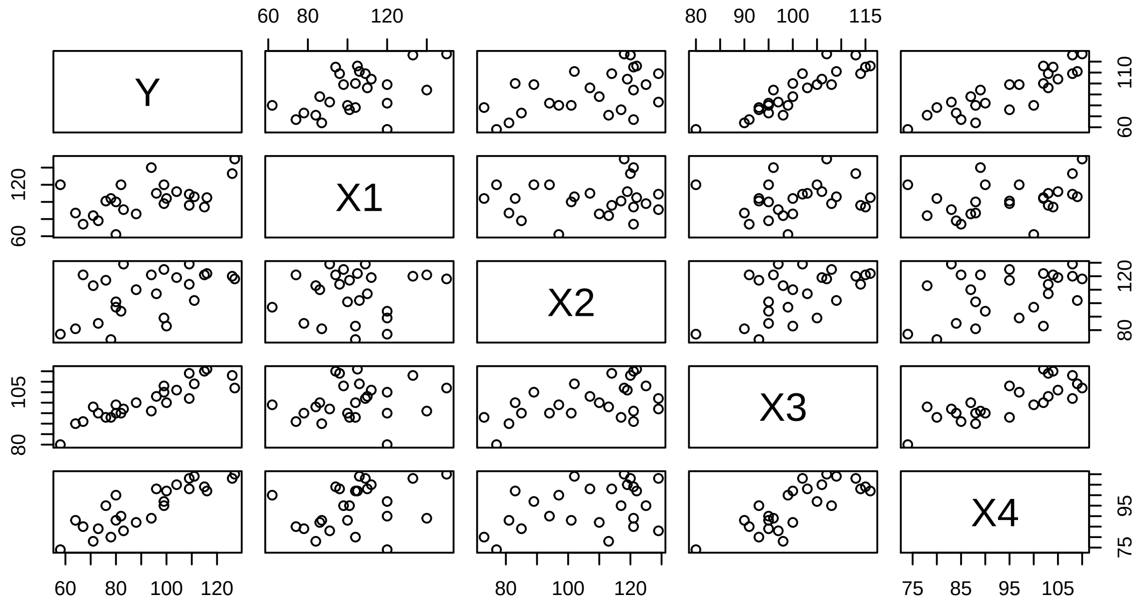
```r
y <- datahw4$Y
beta_comp[1, 1] <- beta[1] * (sd(y)/sd(X[, 2]))
beta_comp[2, 1] <- beta[2] * (sd(y)/sd(X[, 3]))
beta_comp[3, 1] <- beta[3] * (sd(y)/sd(X[, 4]))

names(beta_comp) <- c("Beta Transformed", "Beta Original")

tbl_beta_comp <- (kable(beta_comp))
tbl_beta_comp <- kable_styling(tbl_beta_comp, position = "center")
tbl_beta_comp <- column_spec(tbl_beta_comp, 1, border_left = T)
tbl_beta_comp <- column_spec(tbl_beta_comp, 2, border_right = T)
```

```r
# SSR Comparison
ssr_comp <- data.frame(matrix(NA, nrow = 2, ncol = 1))
rownames(ssr_comp) <- c("SSR X1", "SSR X1|X2")
ssr_comp[2, 1] <- (rss_b0b2 - rss_b0b1b2)
ssr_comp[1, 1] <- (rss_b0 - rss_b0b1)
names(ssr_comp) <- c("")
tbl_ssr_comp <- (kable(ssr_comp))
tbl_ssr_comp <- kable_styling(tbl_ssr_comp, position = "center")
tbl_ssr_comp <- column_spec(tbl_ssr_comp, 1, border_left = T)
tbl_ssr_comp <- column_spec(tbl_ssr_comp, 2, border_right = T)
```

Problem 2:

(a)

```r
# Importing the Data
data3hw4 <- read.csv("D:/Vtech/Regression and ANOVA/PS4/data3hw4.txt",
    sep = "")
# Scatterplot Matrix
pairs(data3hw4)
```




```r
# Correlation Matric
corr_mat <- cor(data3hw4[, 2:5])
tbl_corr <- (kable(corr_mat))
tbl_corr <- kable_styling(tbl_corr, position = "center")
tbl_corr <- column_spec(tbl_corr, 1, border_left = T)
tbl_corr <- column_spec(tbl_corr, 5, border_right = T)
```

```r
# Estimating the Variance Inflation Factors
lmfitx1 <- lm(X1 ~ X4 + X2 + X3, data = data3hw4)
lmfitx2 <- lm(X2 ~ X1 + X4 + X3, data = data3hw4)
lmfitx3 <- lm(X3 ~ X1 + X2 + X4, data = data3hw4)
lmfitx4 <- lm(X4 ~ X1 + X2 + X3, data = data3hw4)

VIF_X1 <- 1/(1 - summary(lmfitx1)$r.squared)
VIF_X2 <- 1/(1 - summary(lmfitx2)$r.squared)
VIF_X3 <- 1/(1 - summary(lmfitx3)$r.squared)
VIF_X4 <- 1/(1 - summary(lmfitx4)$r.squared)

vif_mat <- data.frame()
vif_mat[1, 1] <- VIF_X1
vif_mat[2, 1] <- VIF_X2
vif_mat[3, 1] <- VIF_X3
vif_mat[4, 1] <- VIF_X4
rownames(vif_mat) <- c("VIF X1", "VIF X2", "VIF X3", "VIF X4")
colnames(vif_mat) <- c("")
tbl_vif <- (kable(vif_mat))
tbl_vif <- kable_styling(tbl_vif, position = "center")
tbl_vif <- column_spec(tbl_vif, 1, border_left = T)
tbl_vif <- column_spec(tbl_vif, 2, border_right = T)
```

```r
# Fitting the full multiple linear model
lmfit2 <- lm(Y ~ ., data = data3hw4)
summary(lmfit2)
```

```
##
## Call:
## lm(formula = Y ~ ., data = data3hw4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9779 -3.4506  0.0941  2.4749  5.9959
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -124.38182    9.94106 -12.512 6.48e-11 ***
## X1             0.29573    0.04397   6.725 1.52e-06 ***
## X2             0.04829    0.05662   0.853  0.40383
## X3             1.30601    0.16409   7.959 1.26e-07 ***
## X4             0.51982    0.13194   3.940  0.00081 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.099 on 20 degrees of freedom
## Multiple R-squared:  0.9629, Adjusted R-squared:  0.9555
## F-statistic: 129.7 on 4 and 20 DF,  p-value: 5.262e-14
```

```r
# Doing variable selection using the all possible subsets approach
# based on adjusted R-sqaured
library(leaps)
b <- regsubsets(Y ~ ., data = data3hw4, nbest = 16)
rs <- summary((b))
rsort <- arrange(data.frame(cbind(rs$which, rs$adjr)), desc(rs$adjr2))
names(rsort) <- c("Intercept", "X1", "X2", "X3", "X4", "R.sq.adj")
subs_mat <- head(rsort, 4)
tbl_subs <- (kable(subs_mat))
tbl_subs <- kable_styling(tbl_subs, position = "center")
tbl_subs <- column_spec(tbl_subs, 1, border_left = T)
tbl_subs <- column_spec(tbl_subs, 6, border_right = T)
```

```r
# Computing Adjusted R-squared, Mallow's Cp, AIC, BIC and
# displaying results in a table
library(leaps)
b <- regsubsets(Y ~ ., data = data3hw4, nbest = 16)
rs <- summary((b))
rsnew <- data.frame(rs$which)
rsnew$rsum <- rowSums(rs$which) - 1
rsort <- arrange(data.frame(cbind(rs$which, rs$adjr, rs$cp, (rs$bic -
    (rsnew$rsum) * log(length(data3hw4[, 1])) + (rsnew$rsum) * 2),
    rs$bic)), desc(rs$adjr2))
rsort$rsum <- NULL
names(rsort) <- c("Intercept", "X1", "X2", "X3", "X4", "R.sq.adj",
    "Cp", "AIC", "BIC")
subs_mat <- head(rsort, 4)
tbl_subs <- (kable(subs_mat))
```

```r
tbl_subs <- kable_styling(tbl_subs, position = "center")
tbl_subs <- column_spec(tbl_subs, 1, border_left = T)
tbl_subs <- column_spec(tbl_subs, 9, border_right = T)
```

```r
# Applying Backward elimination and saving results to a table
b <- regsubsets(Y ~ ., data = data3hw4, nbest = 16, method = "backward")
rs <- summary((b))
rsort <- arrange(data.frame(cbind(rs$which, rs$adjr)), desc(rs$adjr2))
names(rsort) <- c("Intercept", "X1", "X2", "X3", "X4", "R.sq.adj")
subs_mat <- head(rsort, 1)
tbl_subs <- (kable(subs_mat))
tbl_subs <- kable_styling(tbl_subs, position = "center")
tbl_subs <- column_spec(tbl_subs, 1, border_left = T)
tbl_subs <- column_spec(tbl_subs, 6, border_right = T)
```

```r
# Applying Forward Selection and saving results to a table
b <- regsubsets(Y ~ ., data = data3hw4, nbest = 16, method = "forward")
rs <- summary((b))
rsort <- arrange(data.frame(cbind(rs$which, rs$adjr)), desc(rs$adjr2))
names(rsort) <- c("Intercept", "X1", "X2", "X3", "X4", "R.sq.adj")
subs_mat <- head(rsort, 1)
tbl_subs <- (kable(subs_mat))
tbl_subs <- kable_styling(tbl_subs, position = "center")
tbl_subs <- column_spec(tbl_subs, 1, border_left = T)
tbl_subs <- column_spec(tbl_subs, 6, border_right = T)
```

```r
# Applying Stepwise regression and saving results to a table
b <- regsubsets(Y ~ ., data = data3hw4, nbest = 16, method = "seqrep")
rs <- summary((b))
rsort <- arrange(data.frame(cbind(rs$which, rs$adjr)), desc(rs$adjr2))
names(rsort) <- c("Intercept", "X1", "X2", "X3", "X4", "R.sq.adj")
subs_mat <- head(rsort, 1)
tbl_subs <- (kable(subs_mat))
tbl_subs <- kable_styling(tbl_subs, position = "center")
tbl_subs <- column_spec(tbl_subs, 1, border_left = T)
tbl_subs <- column_spec(tbl_subs, 6, border_right = T)
```