7.11

Problem 7.15

R Program (Direct Calculations)

```
prob7.15 <- read.table("http://www.stat.ufl.edu/~winner/sta4702/data/wichern/T7-1.DAT",
     header=F,col.names=c("dwsz","asval","sellpr"))
attach(prob7.15)

n <- length(sellpr)
Z <- cbind(rep(1,n),dwsz,asval)
y <- sellpr

ZZI <- solve(t(Z) %*% Z)
beta_hat <- ZZI %*% t(Z) %*% y
SSErr <- t(y - (Z %*% beta_hat)) %*% (y - (Z %*% beta_hat))
MSErr <- SSErr / (n-ncol(Z))
SE_beta_hat <- sqrt(diag(MSErr[1,1] * ZZI))
t_beta_hat <- beta_hat / SE_beta_hat
P_beta_hat <- 2*(1-pt(abs(t_beta_hat),n-ncol(Z)))

reg.out <- cbind(beta_hat,SE_beta_hat,t_beta_hat,P_beta_hat)
colnames(reg.out) <- c("Estimate","Std Err","t-stat","P-value")
round(reg.out,4)

ybar <- rep(mean(y),n)
SSReg <- t((Z %*% beta_hat) - ybar) %*% ((Z %*% beta_hat) - ybar)
dfReg <- ncol(Z) - 1
MSReg <- SSReg / dfReg

(R_square <- SSReg / (SSReg + SSErr))
(F_stat <- MSReg / MSErr)
(F_CV <- qf(.95,ncol(Z)-1,n-ncol(Z)))
(F_PV <- 1 - pf(F_stat,ncol(Z)-1,n-ncol(Z)))

resids <- y - Z %*% beta_hat
qqnorm(resids); qqline(resids)
shapiro.test(resids)

### Prediction Interval
z0 <- matrix(c(1,17,46),ncol=1)

yhat0 <- t(z0) %*% beta_hat
SE.pred.z0 <- sqrt(MSErr[1,1] * (1 + (t(z0) %*% ZZI %*% z0)))
t.crit <- qt(.975,n-ncol(Z))

pred.int <- cbind(yhat0, yhat0-t.crit*SE.pred.z0, yhat0+t.crit*SE.pred.z0)
colnames(pred.int) <- c("Prediction","Lower","Upper")
round(pred.int,4)

### Likelihood-Ratio Test
C.LR <- matrix(c(0,0,1),ncol=3)
F.LR1 <- t(C.LR %*% beta_hat) %*% solve(C.LR %*% ZZI %*% t(C.LR)) %*% (C.LR %*% beta_hat)
F.LR2 <- MSErr[1,1] * 1
F.LR <- F.LR1 / F.LR2
F.CV <- qf(.95,1,n-ncol(Z))
F.PV <- 1 - pf(F.LR,1,n-ncol(Z))

F.out <- cbind(F.LR, F.CV, F.PV)
colnames(F.out) <- c("F", "F(.05)", "Pr>F")
round(F.out, 4)
```

7.12

R Program (lm function)

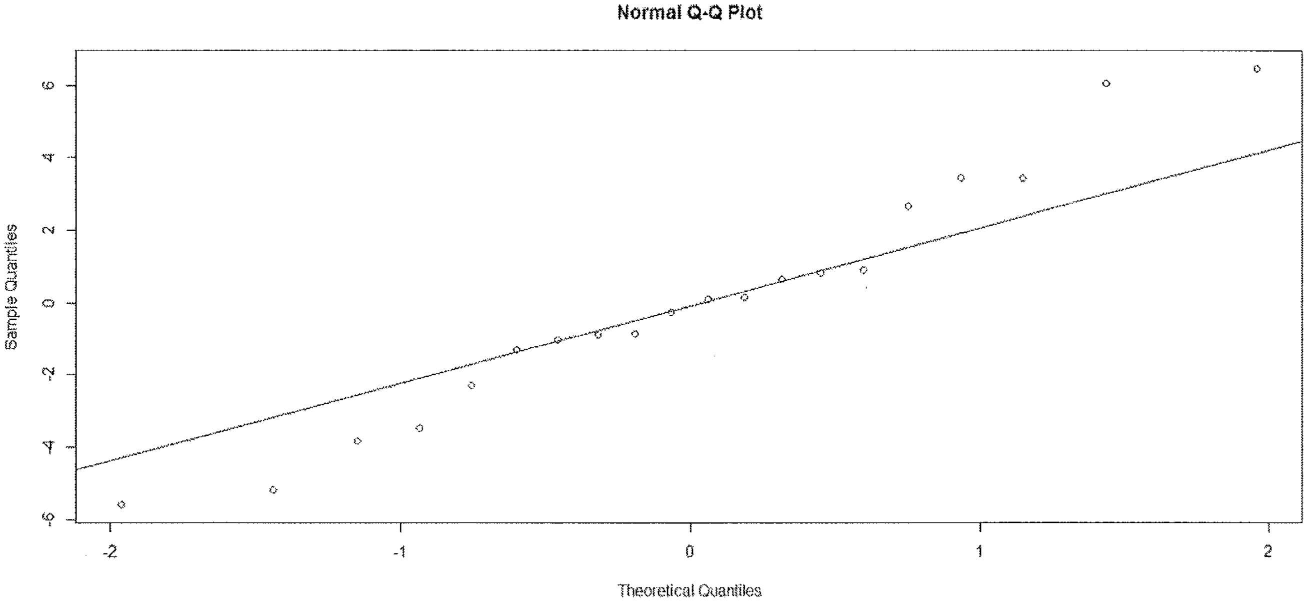
```
#### Using lm function

mod1 <- lm(sellpr ~ dwsz + asval)
summary(mod1)

newdata <- data.frame(dwsz=17,asval=46)
predict(mod1,newdata,int="predict",se=T)

mod2 <- lm(sellpr ~ dwsz)
anova(mod2,mod1)
```

R Output

```
> round(reg.out,4)
      Estimate Std Err t-stat P-value
       30.9666  7.8822 3.9287  0.0011
dwsz    2.6344  0.7856 3.3534  0.0038
asval   0.0452  0.2852 0.1584  0.8760

> (R_square <- SSReg / (SSReg + SSErr))
 [1,] 0.834397
> (F_stat <- MSReg / MSErr)
 [1,] 42.82758
> (F_CV <- qf(.95,ncol(Z)-1,n-ncol(Z)))
[1] 3.591531
> (F_PV <- 1 - pf(F_stat,ncol(Z)-1,n-ncol(Z)))
 [1,] 2.30186e-07

> shapiro.test(resids)
        Shapiro-Wilk normality test
data:  resids
W = 0.9666, p-value = 0.6814

> round(pred.int,4)
     Prediction   Lower   Upper
[1,]    77.8298 64.1192 91.5404

> round(F.out, 4)
          F F(.05)  Pr>F
[1,] 0.0251 4.4513 0.876
```

7.13

```
> summary(mod1)

Call:
lm(formula = sellpr ~ dwsz + asval)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.96657    7.88221   3.929  0.00108 **
dwsz         2.63440    0.78560   3.353  0.00377 **
asval        0.04518    0.28518   0.158  0.87598

Residual standard error: 3.473 on 17 degrees of freedom
Multiple R-squared:  0.8344,    Adjusted R-squared:  0.8149
F-statistic: 42.83 on 2 and 17 DF,  p-value: 2.302e-07

> predict(mod1,newdata,int="predict",se=T)
$fit
       fit      lwr      upr
1 77.82982 64.11922 91.54041

> anova(mod2,mod1)
Analysis of Variance Table

Model 1: sellpr ~ dwsz
Model 2: sellpr ~ dwsz + asval
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     18 205.3
2     17 205.0  1    0.3027 0.0251  0.876
```


Normal Q-Q Plot

Problem 7.19

7.14

R Program

```
prob7_19 <- read.table("http://www.stat.ufl.edu/~winner/sta4702/data/wichern/T7-5.DAT",
     header=F,col.names=c("z1","z2","z3","z4","z5","y"))
attach(prob7_19)

lny <- log(y)

battery <- data.frame(lny, prob7_19)

install.packages("leaps")
library(leaps)

all_battery <- regsubsets(lny ~ z1+z2+z3+z4+z5,
 nbest=4,data=battery)
aprout <- summary(all_battery)
n <- length(battery$lny)
p <- apply(aprout$which, 1, sum)
aprout$aic <- aprout$bic - log(n) * p + 2 * p
with(aprout,round(cbind(which,rsq,adjr2,cp,bic,aic),3))     ## Prints "readable" results

### Based on Cp(Z2,Z4)=2.604 <= (2+1) and BIC(z2,z4) minimized, use z2,z4

best.mod <- lm(lny ~ z2+z4)
summary(best.mod)
shapiro.test(resid(best.mod))
qqnorm(resid(best.mod)); qqline(resid(best.mod))
```

R Text Output

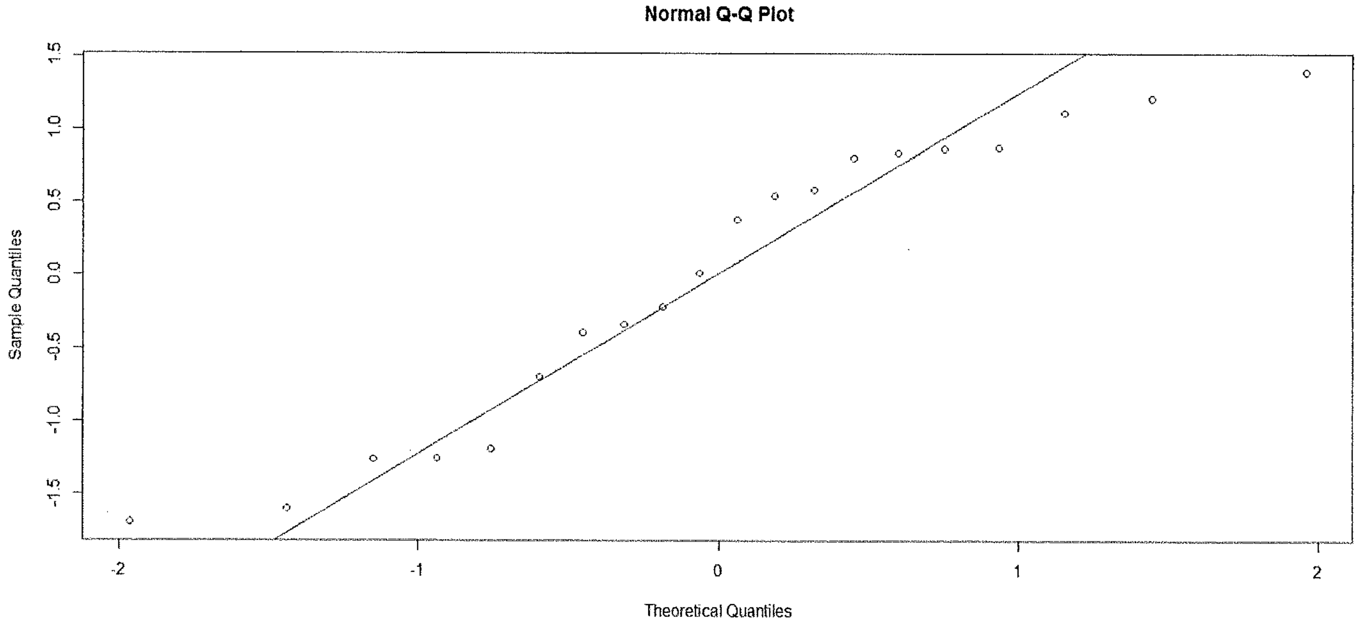
```
> with(aprout,round(cbind(which,rsq,adjr2,cp,bic,aic),3))     ## Prints
"readable" results
  (Intercept) z1 z2 z3 z4 z5   rsq  adjr2     cp    bic     aic
1           1  0  0  0  1  0 0.517  0.490  4.087 -8.558 -10.549
1           1  0  1  0  0  0 0.085  0.034 22.044  4.216   2.225
1           1  1  0  0  0  0 0.031 -0.023 24.297  5.367   3.375
1           1  0  0  0  0  1 0.010 -0.045 25.150  5.786   3.794
2           1  0  1  0  1  0 0.601  0.554  2.604 -9.370 -12.357
2           1  0  0  0  1  1 0.551  0.498  4.671 -7.024 -10.011
2           1  0  0  1  1  0 0.531  0.475  5.515 -6.140  -9.127
2           1  1  0  0  1  0 0.523  0.467  5.829 -5.820  -8.808
3           1  0  1  0  1  1 0.642  0.575  2.880 -8.567 -12.550
3           1  1  1  0  1  0 0.612  0.539  4.149 -6.930 -10.913
3           1  0  1  1  1  0 0.605  0.530  4.441 -6.572 -10.555
3           1  0  0  1  1  1 0.566  0.485  6.047 -4.708  -8.691
4           1  1  1  0  1  1 0.653  0.560  4.429 -6.186 -11.165
4           1  0  1  1  1  1 0.646  0.552  4.699 -5.816 -10.795
4           1  1  1  1  1  0 0.621  0.520  5.746 -4.440  -9.419
4           1  1  0  1  1  1 0.581  0.469  7.440 -2.397  -7.375
5           1  1  1  1  1  1 0.663  0.543  6.000 -3.795  -9.769
```

Continued

7.15

```
> best.mod <- lm(lny ~ z2+z4)
> summary(best.mod)

Call:
lm(formula = lny ~ z2 + z4)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.6870 -0.8171  0.1999  0.8448  1.3938 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.75648    0.74098   3.720 0.001702 ** 
z2          -0.32182    0.17043  -1.888 0.076180 .  
z4           0.11382    0.02429   4.685 0.000213 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.058 on 17 degrees of freedom
Multiple R-squared:  0.6006,    Adjusted R-squared:  0.5536 
F-statistic: 12.78 on 2 and 17 DF,  p-value: 0.000409

> shapiro.test(resid(best.mod))

        Shapiro-Wilk normality test

data:  resid(best.mod)
W = 0.92029, p-value = 0.1004
```

Graphics Output



7.16

Problem 7.21

R Program:

```
prob7_21 <- read.table("http://www.stat.ufl.edu/~winner/sta4702/data/wichern/T1-5.dat",
  header=F,col.names=c("wind","solar","CO","NO","NO2","O3","HC"))
attach(prob7_21)

Y1 <- as.matrix(NO2,ncol=1)
n <- nrow(Y1)
Z <- cbind(rep(1,n),wind,solar)
r <- ncol(Z)-1

beta_hat1 <- solve(t(Z) %*% Z) %*% t(Z) %*% Y1
Yhat1 <- Z %*% beta_hat1
E1 <- Y1 - Yhat1
MSE1 <- (1/(n-(r+1))) * (t(E1) %*% E1)
V.beta_hat1 <- MSE1[1,1] * solve(t(Z) %*% Z)
SE.beta_hat1 <- sqrt(diag(V.beta_hat1))
t.beta_hat1 <- beta_hat1/SE.beta_hat1
pv.beta_hat1 <- 2*(1-pt(abs(t.beta_hat1),n-(r+1)))

beta1.out <- cbind(beta_hat1,SE.beta_hat1,t.beta_hat1,pv.beta_hat1)
colnames(beta1.out) <- c("Estimate","Std. Err.","t","p-value")
round(beta1.out,3)

z01 <- matrix(c(1,10,80),ncol=1)
yhat01 <- t(z01) %*% beta_hat1
se.pred <- sqrt(MSE1[1,1] * (1 + (t(z01) %*% solve(t(Z) %*% Z) %*% z01)))
t.crit <- qt(.975,n-(r+1))
pi.out1 <- cbind(yhat01,yhat01-t.crit*se.pred,yhat01+t.crit*se.pred)
colnames(pi.out1) <- c("pred","Power","Upper")
round(pi.out1,3)

mod1 <- lm(NO2~wind+solar)
newdata <- data.frame(wind=10,solar=80)
summary(mod1)
predict(mod1,newdata,int="pred")

Y <- cbind(NO2,O3)
n <- nrow(Y); m <- ncol(Y)
Z <- cbind(rep(1,n),wind,solar)
r <- ncol(Z)-1

(beta_hat <- solve(t(Z) %*% Z) %*% t(Z) %*% Y)
Yhat <- Z %*% beta_hat
E <- Y - Yhat

(Sigma.hat <- (1/n) * (t(E) %*% E))

### Prediction Interval based on bivariate regression

yhat01.2 <- t(z01) %*% beta_hat[,1]
pi_cv.2 <- sqrt(((m*(n-r-1))/(n-r-m))*qf(0.95,m,n-r-m))
pi_se.2 <- sqrt((1 + (t(z01) %*% solve(t(Z) %*% Z) %*% z01)) *
       Sigma.hat[1,1]*n/(n-r-1))
pi_lb.2 <- yhat01.2 - pi_cv.2 * pi_se.2
pi_ub.2 <- yhat01.2 + pi_cv.2 * pi_se.2
pi_out.2 <- cbind(yhat01.2, pi_lb.2, pi_ub.2)
colnames(pi_out.2) <- c("Predict", "Lower", "Upper")
round(pi_out.2,3)
```

7.17

```
yhat01.2 <- t(z01) %*% beta_hat[,2]
pi_cv.2 <- sqrt(((m*(n-r-1))/(n-r-m))*qf(0.95,m,n-r-m))
pi_se.2 <- sqrt((1 + (t(z01) %*% solve(t(Z) %*% Z) %*% z01)) *
      Sigma.hat[2,2]*n/(n-r-1))
pi_lb.2 <- yhat01.2 - pi_cv.2 * pi_se.2
pi_ub.2 <- yhat01.2 + pi_cv.2 * pi_se.2
pi_out.2 <- cbind(yhat01.2, pi_lb.2, pi_ub.2)
colnames(pi_out.2) <- c("Predict", "Lower", "Upper")
round(pi_out.2,3)

#### Points a constant distance from the origin: drive distance/fairway pct

alpha <- 0.05
crit.dist1 <- (1 + t(z01) %*% solve(t(Z) %*% Z) %*% z01)
#crit.dist2 <- ((m*(n-r-1))/(n-r-m))*qf(1-alpha,m,n-r-m)
crit.dist2 <- (m/(n-r-m))*qf(1-alpha,m,n-r-m)
crit.dist <- crit.dist1*crit.dist2
crit.dist <- sqrt(crit.dist)
S <- (n / (n-r-1)) * Sigma.hat
A <- n * Sigma.hat
# mu.test <- c(0.562,0.589)
ctr1 <- t(z01) %*% beta_hat[,1]
ctr2 <- t(z01) %*% beta_hat[,2]
ctr <- c(ctr1,ctr2)
angles <- seq(0, 2*pi, length.out=200)


eigVal  <- eigen(A)$values
eigVec  <- eigen(A)$vectors
eigScl  <- eigVec %*% diag(sqrt(eigVal))  # scale eigenvectors to length = square-root
xMat    <- rbind(ctr[1] + eigScl[1, ]*crit.dist, ctr[1] - eigScl[1, ]*crit.dist)
yMat    <- rbind(ctr[2] + eigScl[2, ]*crit.dist, ctr[2] - eigScl[2, ]*crit.dist)
ellBase <- cbind(sqrt(eigVal[1])*crit.dist*cos(angles),
sqrt(eigVal[2])*crit.dist*sin(angles)) # normal ellipse
ellRot  <- eigVec %*% t(ellBase)                                  # rotated ellipse

plot((ellRot+ctr)[1, ], (ellRot+ctr)[2, ], asp=1, type="l", lwd=2,
main="100(1-a)% Confidence Ellipsoid",
xlab="NO2", ylab="O3")
matlines(xMat, yMat, lty=1, lwd=2, col="blue")
points(ctr[1], ctr[2], pch=4, col="orange", lwd=3)
```

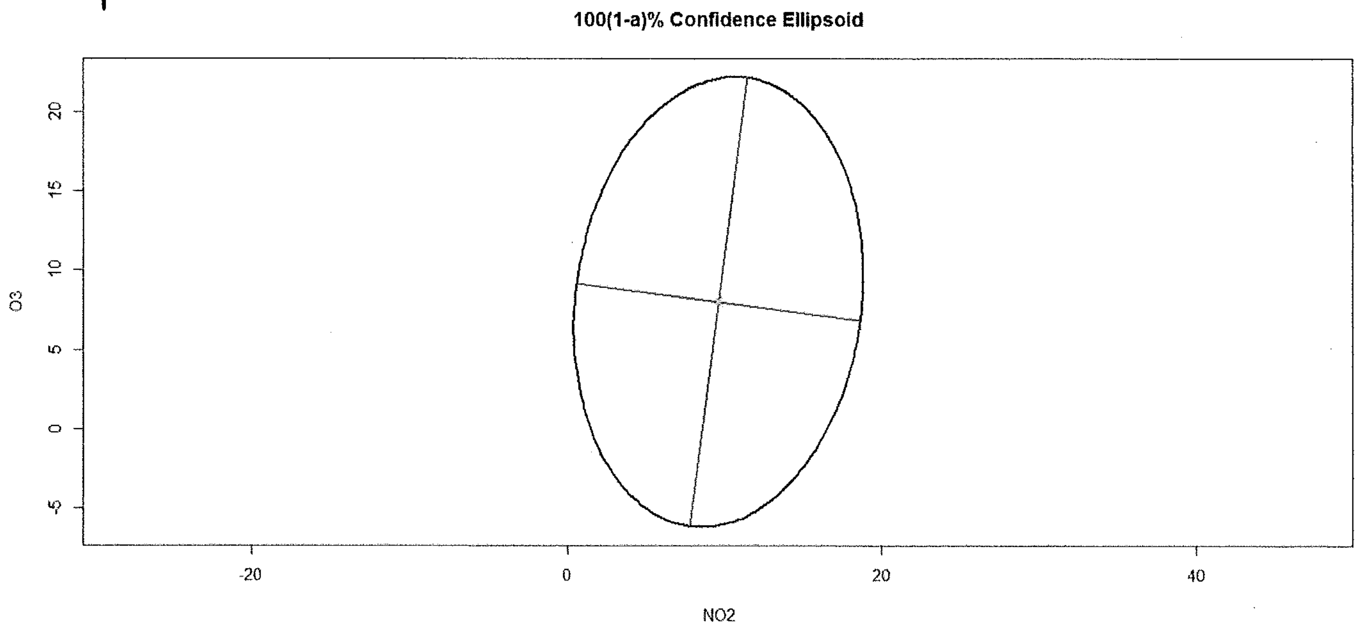
7.18

R Text Output

```
> round(beta1.out,3)
      Estimate Std. Err.      t p-value
        10.115     3.626  2.789   0.008
wind    -0.211     0.339 -0.623   0.537
solar    0.021     0.031  0.664   0.510
>
> round(pi.out1,3)
      pred Power  Upper
[1,] 9.646 2.427 16.864
>
> summary(mod1)

Call:
lm(formula = NO2 ~ wind + solar)

Residuals:
    Min      1Q  Median      3Q     Max
-5.7521 -2.2053 -0.5917  1.6852 10.4623

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.11454    3.62607   2.789  0.00813 **
wind        -0.21129    0.33917  -0.623  0.53694
solar        0.02055    0.03094   0.664  0.51042
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.416 on 39 degrees of freedom
Multiple R-squared:  0.02311,   Adjusted R-squared:  -0.02698
F-statistic: 0.4614 on 2 and 39 DF,  p-value: 0.6338

> predict(mod1,newdata,int="pred")
       fit      lwr      upr
1 9.645627 2.427199 16.86406

> (beta_hat <- solve(t(Z) %*% Z) %*% t(Z) %*% Y)
              NO2          O3
      10.11454142  8.27619196
wind  -0.21129082 -0.78682381
solar  0.02054992  0.09518035
> Yhat <- Z %*% beta_hat
> E <- Y - Yhat
>
> (Sigma.hat <- (1/n) * (t(E) %*% E))
          NO2        O3
NO2 10.836561  1.974156
O3   1.974156 25.665864
>
> ### Prediction Interval based on bivariate regression

> round(pi_out.2,3)
     Predict Lower  Upper
[1,]   9.646 0.436 18.856
>
>
> round(pi_out.2,3)
     Predict  Lower  Upper
[1,]   8.022 -6.152 22.196
```

Problem 7.21

7.19



7.20

Problem 7.25

R Program

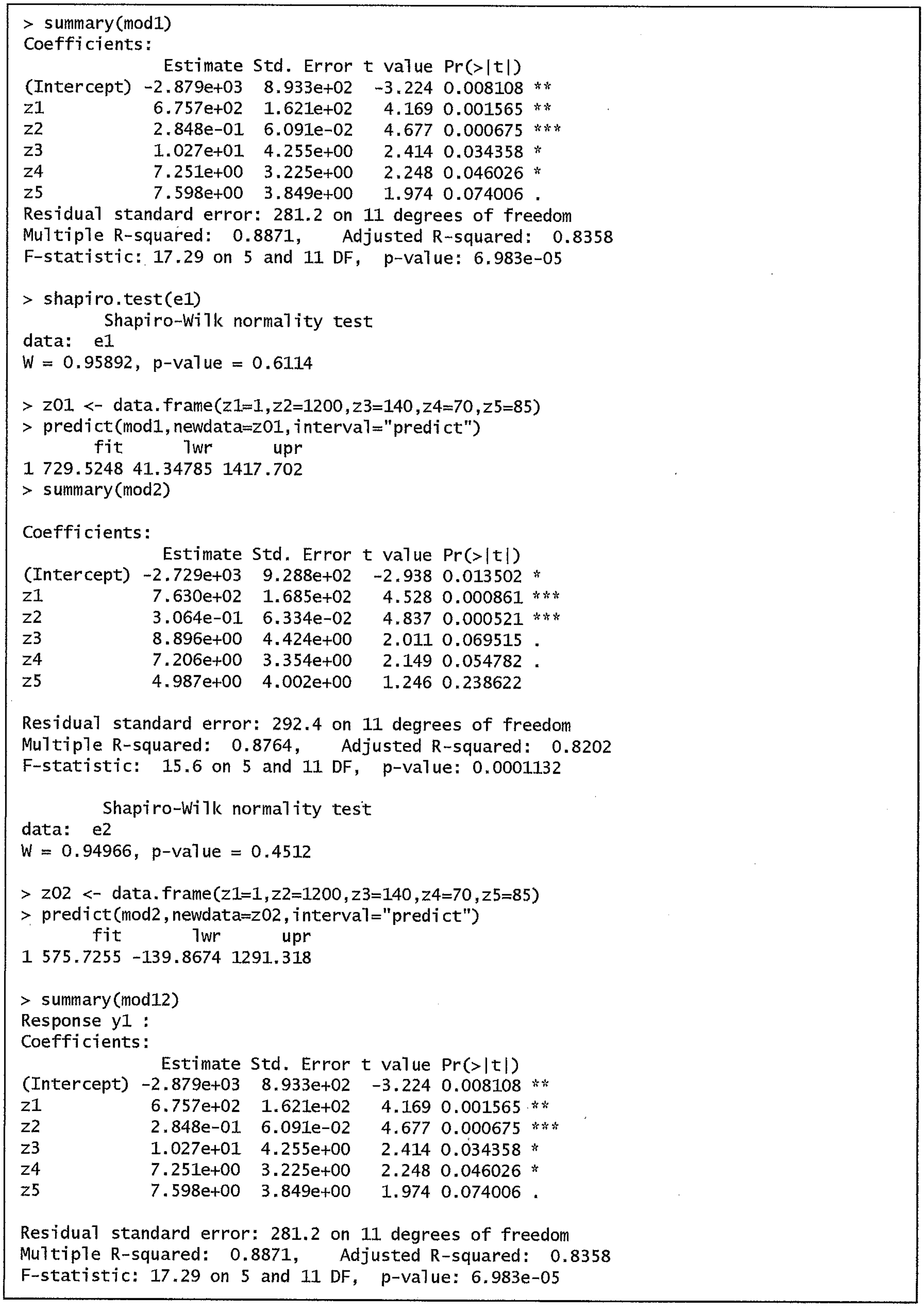
```
prob7.25 <-
read.table("http://www.stat.ufl.edu/~winner/sta4702/data/wichern/T7-6.DAT",
     header=F,col.names=c("y1","y2","z1","z2","z3","z4","z5"))
attach(prob7.25)

n <- length(y1)
mod1 <- lm(y1 ~ z1 + z2 + z3 + z4 + z5)
summary(mod1)
e1 <- resid(mod1)

shapiro.test(e1)
qqnorm(e1); qqline(e1)

z01 <- data.frame(z1=1,z2=1200,z3=140,z4=70,z5=85)
predict(mod1,newdata=z01,interval="predict")

mod2 <- lm(y2 ~ z1 + z2 + z3 + z4 + z5)
summary(mod2)
e2 <- resid(mod2)

shapiro.test(e2)
qqnorm(e2); qqline(e2)

z02 <- data.frame(z1=1,z2=1200,z3=140,z4=70,z5=85)
predict(mod2,newdata=z02,interval="predict")

m <- 2; r <- 5
mod12 <- lm(cbind(y1,y2) ~ z1 + z2 + z3 + z4 + z5)
summary(mod12)
e12 <- resid(mod12)

shapiro.test(e12)
qqnorm(e12); qqline(e12)

z01 <- data.frame(z1=1,z2=1200,z3=140,z4=70,z5=85)
predict(mod12,newdata=z01,interval="predict")

Sigma.hat <- (1/n) * (t(e12) %*% e12)
z0 <- matrix(c(1,1,1200,140,70,85), ncol=1)
beta_hat <- coef(mod12)
Z <- model.matrix(mod12)

### Prediction Interval based on bivariate regression

yhat01.2 <- t(z0) %*% beta_hat[,1]
pi_cv.2 <- sqrt(((m*(n-r-1))/(n-r-m))*qf(0.95,m,n-r-m))
pi_se.2 <- sqrt((1 + (t(z0) %*% solve(t(Z) %*% Z) %*% z0)) *
       Sigma.hat[1,1]*n/(n-r-1))
pi_lb.2 <- yhat01.2 - pi_cv.2 * pi_se.2
pi_ub.2 <- yhat01.2 + pi_cv.2 * pi_se.2
pi_out.2 <- cbind(yhat01.2, pi_lb.2, pi_ub.2)
colnames(pi_out.2) <- c("Predict", "Lower", "Upper")
round(pi_out.2,3)
```

7.21

```
yhat01.2 <- t(z0) %*% beta_hat[,2]
pi_cv.2 <- sqrt(((m*(n-r-1))/(n-r-m))*qf(0.95,m,n-r-m))
pi_se.2 <- sqrt((1 + (t(z0) %*% solve(t(Z) %*% Z) %*% z0)) *
       Sigma.hat[2,2]*n/(n-r-1))
pi_lb.2 <- yhat01.2 - pi_cv.2 * pi_se.2
pi_ub.2 <- yhat01.2 + pi_cv.2 * pi_se.2
pi_out.2 <- cbind(yhat01.2, pi_lb.2, pi_ub.2)
colnames(pi_out.2) <- c("Predict", "Lower", "Upper")
round(pi_out.2,3)

#### Points a constant distance from the origin: drive distance/fairway pct

alpha <- 0.05
crit.dist1 <- (1 + t(z0) %*% solve(t(Z) %*% Z) %*% z0)
#crit.dist2 <- ((m*(n-r-1))/(n-r-m))*qf(1-alpha,m,n-r-m)
crit.dist2 <- (m/(n-r-m))*qf(1-alpha,m,n-r-m)
crit.dist <- crit.dist1*crit.dist2
crit.dist <- sqrt(crit.dist)
S <- (n / (n-r-1)) * Sigma.hat
A <- n * Sigma.hat
# mu.test <- c(0.562,0.589)
ctr1 <- t(z0) %*% beta_hat[,1]
ctr2 <- t(z0) %*% beta_hat[,2]
ctr <- c(ctr1,ctr2)
angles <- seq(0, 2*pi, length.out=200)


eigVal  <- eigen(A)$values
eigVec  <- eigen(A)$vectors
eigScl  <- eigVec %*% diag(sqrt(eigVal))  # scale eigenvectors to length = square-root
xMat    <- rbind(ctr[1] + eigScl[1, ]*crit.dist, ctr[1] - eigScl[1, ]*crit.dist)
yMat    <- rbind(ctr[2] + eigScl[2, ]*crit.dist, ctr[2] - eigScl[2, ]*crit.dist)
ellBase <- cbind(sqrt(eigVal[1])*crit.dist*cos(angles),
sqrt(eigVal[2])*crit.dist*sin(angles)) # normal ellipse
ellRot  <- eigVec %*% t(ellBase)                                 # rotated ellipse

plot((ellRot+ctr)[1, ], (ellRot+ctr)[2, ], asp=1, type="l", lwd=2,
main="100(1-a)% Confidence Ellipsoid",
xlab="y1", ylab="y2")
matlines(xMat, yMat, lty=1, lwd=2, col="blue")
points(ctr[1], ctr[2], pch=4, col="orange", lwd=3)
```

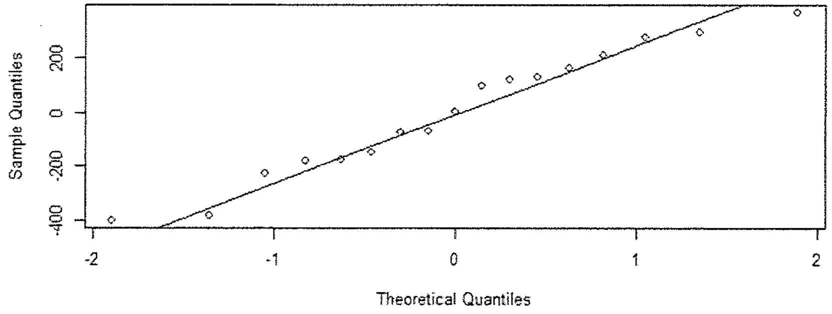
Text Output

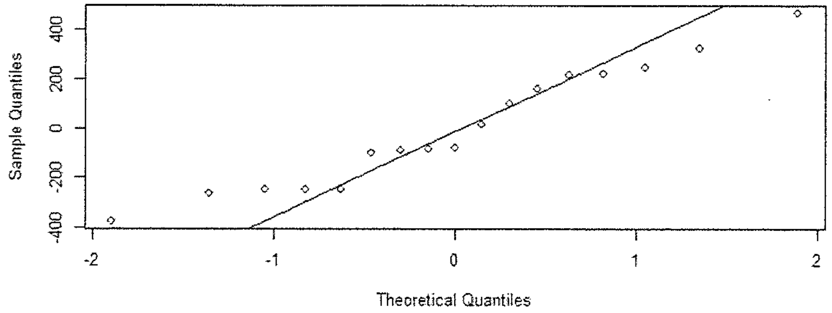
7.22

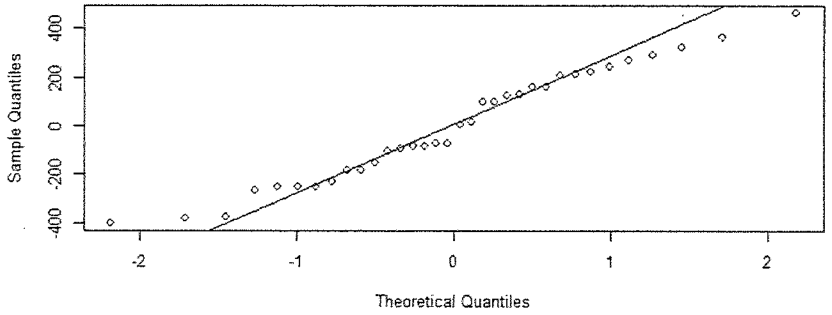


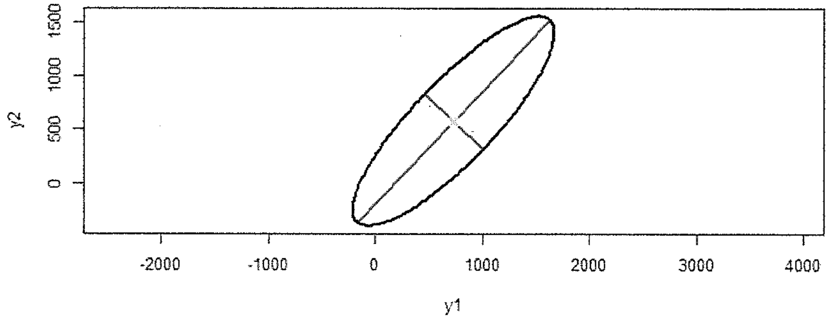
```
> summary(mod1)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.879e+03  8.933e+02  -3.224 0.008108 **
z1           6.757e+02  1.621e+02   4.169 0.001565 **
z2           2.848e-01  6.091e-02   4.677 0.000675 ***
z3           1.027e+01  4.255e+00   2.414 0.034358 *
z4           7.251e+00  3.225e+00   2.248 0.046026 *
z5           7.598e+00  3.849e+00   1.974 0.074006 .
Residual standard error: 281.2 on 11 degrees of freedom
Multiple R-squared:  0.8871,    Adjusted R-squared:  0.8358
F-statistic: 17.29 on 5 and 11 DF,  p-value: 6.983e-05

> shapiro.test(e1)
        Shapiro-Wilk normality test
data:  e1
W = 0.95892, p-value = 0.6114

> z01 <- data.frame(z1=1,z2=1200,z3=140,z4=70,z5=85)
> predict(mod1,newdata=z01,interval="predict")
       fit      lwr      upr
1 729.5248 41.34785 1417.702
> summary(mod2)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.729e+03  9.288e+02  -2.938 0.013502 *
z1           7.630e+02  1.685e+02   4.528 0.000861 ***
z2           3.064e-01  6.334e-02   4.837 0.000521 ***
z3           8.896e+00  4.424e+00   2.011 0.069515 .
z4           7.206e+00  3.354e+00   2.149 0.054782 .
z5           4.987e+00  4.002e+00   1.246 0.238622

Residual standard error: 292.4 on 11 degrees of freedom
Multiple R-squared:  0.8764,    Adjusted R-squared:  0.8202
F-statistic:  15.6 on 5 and 11 DF,  p-value: 0.0001132

        Shapiro-Wilk normality test
data:  e2
W = 0.94966, p-value = 0.4512

> z02 <- data.frame(z1=1,z2=1200,z3=140,z4=70,z5=85)
> predict(mod2,newdata=z02,interval="predict")
       fit       lwr      upr
1 575.7255 -139.8674 1291.318

> summary(mod12)
Response y1 :
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.879e+03  8.933e+02  -3.224 0.008108 **
z1           6.757e+02  1.621e+02   4.169 0.001565 **
z2           2.848e-01  6.091e-02   4.677 0.000675 ***
z3           1.027e+01  4.255e+00   2.414 0.034358 *
z4           7.251e+00  3.225e+00   2.248 0.046026 *
z5           7.598e+00  3.849e+00   1.974 0.074006 .

Residual standard error: 281.2 on 11 degrees of freedom
Multiple R-squared:  0.8871,    Adjusted R-squared:  0.8358
F-statistic: 17.29 on 5 and 11 DF,  p-value: 6.983e-05
```

7.23


Normal Q-Q Plot


Normal Q-Q Plot


Normal Q-Q Plot


100(1-a)% Confidence Ellipsoid