Ngoc Ha

HW 4 - ST 557

Problem 1

In [1]:
```
reading_data <- read.csv('ReadingTest.csv')
head(reading_data)
```

Subject	PRE1	PRE2	POST1	POST2
1	4	3	5	4
2	6	5	9	5
3	9	4	5	3
4	12	6	8	5
5	16	5	10	9
6	15	13	9	8

In [2]:
```
pre  <- cbind(reading_data$PRE1,  reading_data$PRE2)
post <- cbind(reading_data$POST1, reading_data$POST2)
```

In [3]:
```
preMean <- c(mean(pre[,1]),mean(pre[,2]))
postMean <- c(mean(post[,1]),mean(post[,2]))
cat("Before:", preMean, "\n")
cat("After: ", postMean)
```

```
Before: 9.787879 5.106061
After:  8.075758 6.712121
```

(1a)

Paired test is appropriate to test the hypothesis $H_0 : \mu_1 = \mu_2$ vs $H_1 : \mu_1 \neq \mu_2$, because test scores before and after come from the same sample, under 2 different conditions.

(1b)

In [4]:
```
n = length(pre)
p = 2
alpha = 0.05
diff <- pre-post
mu1 <- mean(diff[,1])
mu2 <- mean(diff[,2])
sampDiffMean <- c(mu1,mu2)
sampDiffCov  <- var(diff)
```

In [5]:
```
Tstat <- n*t(sampDiffMean)%*%solve(sampDiffCov)%*%sampDiffMean
crit <- p*(n-1)/(n-p)*qf(1-alpha,p,n-p)
decision = Tstat>crit
cat("Test statistic:", Tstat, "Critical level:", crit, "\n")
cat("Reject Null:", decision)
```

```
Test statistic: 74.94971 Critical level: 6.178845
Reject Null: TRUE
```

(1c)

In [6]:
```
alpha_Bon = alpha/p
test1_bon <- t.test(diff[,1],alternative=c("two.sided"),conf.level=1-alpha_Bon
)
test2_bon <- t.test(diff[,2],alternative=c("two.sided"),conf.level=1-alpha_Bon
)
cat("95% simultaneous Bonferroni confidence intervals:\n")
cat("Test 1:", test1_bon$conf, "Test 2:", test2_bon$conf)
```

```
95% simultaneous Bonferroni confidence intervals:
Test 1: 0.8629862 2.561256 Test 2: -2.473941 -0.7381799
```

(1d)

In [7]:
```
eiDec <- eigen(sampDiffCov)
eiVec1 <- eiDec$vectors[,1]
eiVec2 <- eiDec$vectors[,2]
eiVal1 <- eiDec$values[1]
eiVal2 <- eiDec$values[2]
eiDec
```

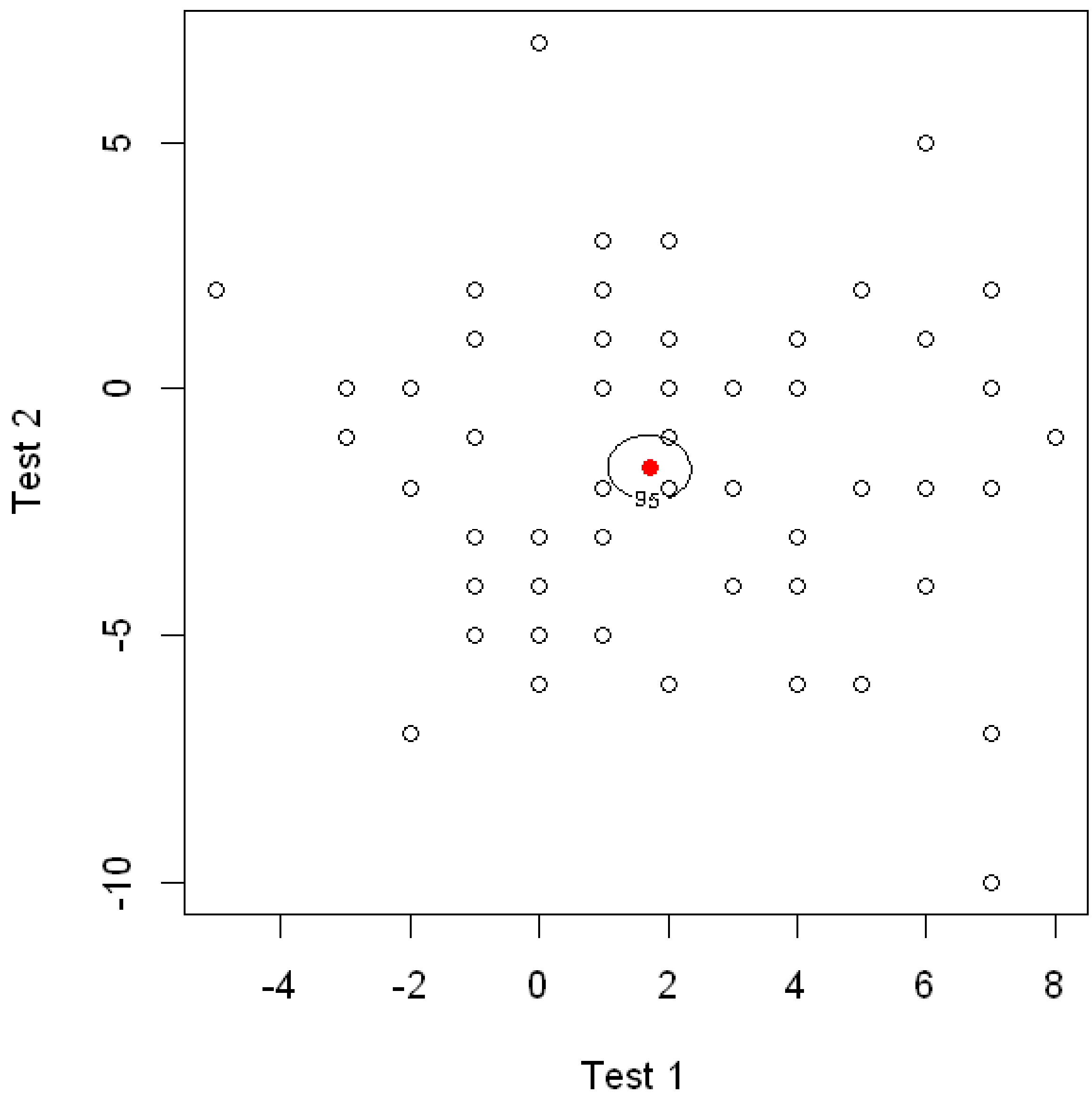
```
eigen() decomposition
$values
[1] 9.765824 8.715528

$vectors
           [,1]       [,2]
[1,] -0.5548990 -0.8319177
[2,]  0.8319177 -0.5548990
```

In [8]:

```
scale1 <- sqrt(eiDec$values[1]*p*(n-1)/(n*(n-p))*df(1-alpha, df1=p, df2=n-p))
scale2 <- sqrt(eiDec$values[2]*p*(n-1)/(n*(n-p))*df(1-alpha, df1=p, df2=n-p))
```

In [9]:

```r
options(repr.plot.width=5, repr.plot.height=5)
muTest.1 <- seq(min(diff[,1]), max(diff[,1]), 0.1)
muTest.2 <- seq(min(diff[,2]), max(diff[,2]), 0.1)
Tstats <- matrix(0, nrow=length(muTest.1), ncol=length(muTest.2))
for(i in 1:length(muTest.1)){
    for(j in 1:length(muTest.2)){
        muTest <- c(muTest.1[i], muTest.2[j])
        Tstats[i,j] <- n*t(sampDiffMean - muTest) %*% solve(sampDiffCov) %*% (
sampDiffMean - muTest)
    }
}
par(mar=c(4,4,1,1))
# Plot the data, and superimpose the confidence ellipsoids
# using the contour() function.
plot(diff, xlab="Test 1", ylab="Test 2")
points(mu1, mu2, pch=16, col=2)
contour(muTest.1, muTest.2, Tstats, levels=(n-1)*p/(n-p)*qf(1-alpha, p, n-p),
drawlabels=T, add=T, labels=95)
```



Problem 2

(2a)

In [10]:
```
tempData <- read.csv('TempData.csv')
head(tempData)
```

X1950s	X1960s	X1970s	X1980s	X1990s	X2000s
10.903	10.999	11.006	11.220	11.661	11.466
11.319	11.405	11.360	11.462	11.980	11.825
10.927	10.873	10.854	10.980	11.330	11.086
11.303	11.271	11.108	11.316	11.869	11.599
11.138	11.239	11.196	11.602	11.945	11.767
9.448	9.620	9.321	9.532	9.774	9.734

In [11]:
```
n <- dim(tempData)[1]
q <- dim(tempData)[2]
```

In [12]:
```
alpha = 0.05
sampMean <- apply(tempData,2,mean)
covar <- var(tempData)
C <- rbind(c(1, -1, 0, 0, 0, 0), c(1, 0, -1, 0, 0, 0), c(1, 0, 0, -1, 0, 0), c
(1, 0, 0, 0, -1, 0), c(1, -1, 0, 0, 0, -1))
```

In [13]:
```
Tstat <- n*t(sampMean)%*%t(C)%*%solve(C%*%covar%*%t(C))%*%C%*%sampMean
critLevel <- (q-1)*(n-1)/(n-q+1)*qf(1-alpha, df1=q-1, df2=n-q+1)
decision = Tstat>crit
cat("Test statistic:", Tstat, "Critical level:", crit, "\n")
cat("Reject Null:", decision)
```

```
Test statistic: 2612.187 Critical level: 6.178845
Reject Null: TRUE
```

(2b)

```
In [14]: alpha_Bon <- alpha/q
test1_bon <- t.test(tempData[,2]-tempData[,1],alternative=c("two.sided"),conf.
level=1-alpha_Bon)
test2_bon <- t.test(tempData[,3]-tempData[,1],alternative=c("two.sided"),conf.
level=1-alpha_Bon)
test3_bon <- t.test(tempData[,4]-tempData[,1],alternative=c("two.sided"),conf.
level=1-alpha_Bon)
test4_bon <- t.test(tempData[,5]-tempData[,1],alternative=c("two.sided"),conf.
level=1-alpha_Bon)
test5_bon <- t.test(tempData[,6]-tempData[,1],alternative=c("two.sided"),conf.
level=1-alpha_Bon)
cat("Bonferroni simultaneous interval 1:", test1_bon$conf, "\n")
cat("Bonferroni simultaneous interval 2:", test2_bon$conf, "\n")
cat("Bonferroni simultaneous interval 3:", test3_bon$conf, "\n")
cat("Bonferroni simultaneous interval 4:", test4_bon$conf, "\n")
cat("Bonferroni simultaneous interval 5:", test5_bon$conf, "\n")
```

```
Bonferroni simultaneous interval 1: 0.08955848 0.2316415
Bonferroni simultaneous interval 2: -0.07257532 0.1199753
Bonferroni simultaneous interval 3: 0.1652247 0.4388753
Bonferroni simultaneous interval 4: 0.5795466 0.8806534
Bonferroni simultaneous interval 5: 0.4245939 0.7960061
```

95% CI for $H_0 : \mu_3 - \mu_1 = 0$ includes 0. No other CI contains 0.

Conclusion: average temperature in Corvallis does not stay constant over the past 60 years.

Problem 3

(3a)

```
In [15]: baseball <- read.csv('BaseballData.csv')
head(baseball)
```

FreeAgent	BatAvg	OBP	Runs	Hits	Doubles	Triples	HRs	RBI	Walks	StrikeOuts	SB	Errc
0	0.260	0.292	59	128	22	7	12	50	23	64	21	
0	0.273	0.346	87	169	28	5	8	58	70	53	3	
0	0.228	0.279	16	38	7	2	3	21	11	32	2	
0	0.250	0.327	40	61	11	0	1	18	24	26	14	
0	0.203	0.240	39	64	10	1	10	33	14	96	13	
0	0.262	0.283	7	38	5	0	0	10	5	18	2	

A paired test is not appropriate because the data comes from different, independent players (i.e. they don't come in pairs). A two-sample test would be more appropriate.

(3b)

In [16]:
```
baseball0 <- baseball[baseball$FreeAgent == 0, 2:ncol(baseball)]
baseball1 <- baseball[baseball$FreeAgent == 1, 2:ncol(baseball)]
```

In [17]:
```
covar0 <- var(baseball0)
covar1 <- var(baseball1)
covar0
covar1
```

	BatAvg	OBP	Runs	Hits	Doubles	Triples	
BatAvg	0.001681337	0.001593887	0.4020978	0.871891	0.1622746	0.02232954	0
OBP	0.001593887	0.002272110	0.5567157	0.927463	0.1762044	0.02075187	0
Runs	0.402097766	0.556715725	776.7007267	1361.935327	250.7841779	44.96768766	144
Hits	0.871890967	0.927463005	1361.9353265	2741.478808	494.2825684	84.29434717	250
Doubles	0.162274570	0.176204409	250.7841779	494.282568	108.1292982	13.12963956	52
Triples	0.022329537	0.020751866	44.9676877	84.294347	13.1296396	6.78447057	3
HRs	0.064828196	0.106463737	144.1820221	250.066185	52.9903673	3.33912110	60
RBI	0.379077867	0.471553456	661.0559186	1279.440155	249.4052822	31.46059113	185
Walks	0.238480832	0.559351778	530.4561040	919.795347	169.0643077	23.40008779	113
StrikeOuts	0.129073989	0.270392869	676.0398966	1269.080110	240.5384822	37.71192021	190
SB	0.070165854	0.095360630	183.0638931	301.258060	46.8771887	19.54482271	11
Errors	0.027357948	0.026698337	80.7989806	170.895454	29.8011023	4.40455055	10

	BatAvg	OBP	Runs	Hits	Doubles	Triples	
BatAvg	0.001384682	0.001328429	0.6062737	1.222733	0.2119166	0.03226899	0.
OBP	0.001328429	0.001994757	0.7503599	1.101477	0.1869919	0.02553877	0.
Runs	0.606273651	0.750359948	755.9731231	1109.447368	205.1219280	30.81405005	183.
Hits	1.222733083	1.101477444	1109.4473684	2057.424812	377.8157895	47.71428571	261.
Doubles	0.211916564	0.186991864	205.1219280	377.815789	97.8272360	7.14151049	58.
Triples	0.032268993	0.025538772	30.8140501	47.714286	7.1415105	5.98069801	1.
HRs	0.083334586	0.125793233	183.9135338	261.823308	58.8984962	1.32330827	106.
RBI	0.464127146	0.502980922	601.7707328	1007.864662	211.4084839	11.67197845	262.
Walks	0.283131859	0.718061834	510.8568062	622.917293	111.4513523	10.27639996	130.
StrikeOuts	0.018600213	0.272056167	504.1301201	646.537594	126.0598698	11.34810908	252.
SB	0.124090226	0.162157895	158.3909774	175.037594	15.3007519	9.71428571	-1.
Errors	0.045063405	0.028763887	28.0742902	67.233083	8.3514757	0.88744249	6.

In [18]:
```
det(covar0)
det(covar1)
sum(diag(covar0)) #trace
sum(diag(covar1))
```

536625347621.708

1236170570008.94

6212.72009254831

5738.6350144968

The covariance matrices do *not* appear to be similar.

(3c)

In [19]:
```
p = dim(baseball0)[2]
n = dim(baseball)[1]
n0 = dim(baseball0)[1]
n1 = dim(baseball1)[1]
mu0 <- apply(baseball0,2,mean)
mu1 <- apply(baseball1,2,mean)
covar_pool <- ((n0-1)*covar0 + (n1-1)*covar1)/(n0+n1-2)
```

In [20]:
```
alpha = 0.05
Tstat <- t(mu0-mu1)%*%solve((1/n0+1/n1)*covar_pool)%*%(mu0-mu1)
crit_level_c <- p*(n0+n1-2)/(n0+n1-p-1)*qf(1-alpha, p, n0+n1-p-1)
reject = Tstat>crit_level_c
cat("T2 statistic:", Tstat, "Critical level:", crit_level_c, "\n")
cat("Reject Null:", reject)
```

```
T2 statistic: 82.92731 Critical level: 22.11154
Reject Null: TRUE
```

(3d)

In [21]:
```
Tstat <- t(mu0-mu1)%*%solve((1/n0)*covar0+(1/n1)*covar1)%*%(mu0-mu1)
crit_level_d <- qchisq(1-alpha,p)
reject = Tstat>crit_level_d
cat("T2 statistic:", Tstat, "Critical level:", crit_level_d, "\n")
cat("Reject Null:", reject)
```

```
T2 statistic: 85.04061 Critical level: 21.02607
Reject Null: TRUE
```

The decision did not change.

(3e)

Not reasonable to always perform both tests for equal and unequal covariance matrices, since that would double the probability of making a type I error.

(3f)

In [22]:
```
v = (p+p^2)/((1/n0*(sum(diag((1/n0*covar0%*%solve(1/n0*covar0+1/n1*covar1))^2
))+(sum(diag(1/n0*covar0%*%solve(1/n0*covar0+1/n1*covar1))))^2))+(1/n1*(sum(di
ag((1/n1*covar1%*%solve(1/n0*covar0+1/n1*covar1))^2))+(sum(diag(1/n1*covar1%*%
solve(1/n0*covar0+1/n1*covar1))))^2)))
```

In [23]:
```
crit_level_f <- (v*p)/(v-p+1)*qf(1-alpha,p,v-p+1)
cat("Critical levels:\n", "Equal variance:", crit_level_c, "\n Unequal varianc
e (large sample):", crit_level_d, "\n Unequal variance (moderate sample):", cr
it_level_f)
```

```
Critical levels:
 Equal variance: 22.11154
 Unequal variance (large sample): 21.02607
 Unequal variance (moderate sample): 22.32848
```

For this problem, it doesn't matter which critical level is used.

Problem 4

(4a)

In [24]:
```
skull <- read.csv("SkullData.csv")
head(skull)
```

Year	MB	BH	BL	NH
-4000	131	138	89	49
-4000	125	131	92	48
-4000	131	132	99	50
-4000	119	132	96	44
-4000	136	143	100	54
-4000	138	137	89	56

In [25]:
```
BC4000 <- skull[skull$Year==-4000,2:5]
BC3300 <- skull[skull$Year==-3300,2:5]
BC1850 <- skull[skull$Year==-1850,2:5]
BC200 <- skull[skull$Year==-200,2:5]
AD150 <- skull[skull$Year==150,2:5]
```

In [26]:
```
covar1 <- var(BC4000)
covar2 <- var(BC3300)
covar3 <- var(BC1850)
covar4 <- var(BC200)
covar5 <- var(AD150)
```

In [27]:
```
det(covar1)
det(covar2)
det(covar3)
det(covar4)
det(covar5)
```

96431.2959853298

61827.9461650085

74729.0725238389

35211.6575797549

189666.581818259

The covariance matrices do not seem to be similar.

(4b)

In [28]:
```
p = 4; k = 5; n = dim(skull)[1];
n1 = dim(BC4000)[1]; n2 = dim(BC3300)[1]; n3 = dim(BC1850)[1]; n4 = dim(BC200)
[1]; n5 = dim(AD150)[1]
mu = apply(skull[,2:ncol(skull)],2,mean); mu1 = apply(BC4000[,2:ncol(BC4000)],
2,mean);
mu2 = apply(BC3300[,2:ncol(BC3300)],2,mean); mu3 = apply(BC1850[,2:ncol(BC1850
)],2,mean);
mu4 = apply(BC200[,2:ncol(BC200)],2,mean); mu5 = apply(AD150[,2:ncol(AD150)],2
,mean);
```

In [29]:
```
alpha = 0.05
W = (n1-1)*covar1 + (n2-1)*covar2 + (n3-1)*covar3 + (n4-1)*covar4 + (n5-1)*cov
ar5
T = (n-1)*var(skull[2:5])
gamma <- det(W)/det(T)
tstat <- -(n-1-(p+k)/2)*log(gamma)
crit_level <- qchisq(1-0.05,p*(k-1))
reject <- tstat > crit_level
cat("Test statistic:", tstat, "\nCritical level:", crit_level, "\nReject Nul
l:", reject)
```

```
Test statistic: 59.25903
Critical level: 26.29623
Reject Null: TRUE
```

Based on the test, we can conclude that skull size changed over this period of time. Therefore, **there is evidence** of interbreeding.

(4c)

In [30]:
```
alpha_Bon <- alpha/p
ssw <- matrix(0, 4, 1)
for (i in c(1:4)){
    ssw[i] <- (n1-1)*var(BC4000[,i])+(n2-1)*var(BC3300[,i])+(n3-1)*var(BC1850[
,i])+(n4-1)*var(BC200[,i])+(n5-1)*var(AD150[,i])
}
sst <- matrix(0,4,1)
for (j in c(1:4)){
    sst[j] <- (n-1)*var(skull[,j+1])
}
ssb <- sst-ssw
cbind(ssw,ssb,sst)
```

3061.067	502.8267	3563.893
3405.267	229.9067	3635.173
3505.967	803.2933	4309.260
1472.133	61.2000	1533.333

In [31]:
```
Tstats <- (ssb/(k-1))/(ssw/(n-k))
crit_level <- qf(1-alpha_Bon,k-1,n-k)
reject <- Tstats > crit_level
reject
```

TRUE
FALSE
TRUE
FALSE

The Null is rejected (reject = TRUE) in 2 of the 4 univariate ANOVAs.

$\alpha* = \frac{\alpha}{p}$

Probability of Type I error is:

$$\begin{aligned} P_{H_0}(Reject H_0) &= P_{H_0}(Reject\ at\ least\ one\ of\ H_{0j}) \\ &\leq \sum_{j=1}^{p} P_{H_0}(Reject H_{0j}) \\ &= \sum_{j=1}^{p} \frac{\alpha}{p} \\ &= \alpha \end{aligned}$$

Probability of type I error is controlled at level α

Problem 5

In [32]:
```
pollution <- as.matrix(read.csv('PollutionData.csv'))
head(pollution)
```

Wind	SolarRad	NO2	O3
8	98	12	8
7	107	9	5
7	103	5	6
10	88	8	15
6	91	8	10
8	90	12	12

(5a)

In [33]:
```
n = nrow(pollution)
x <- cbind(1,pollution[,1:2])
y <- pollution[,3:4]
p <- ncol(x)-1
m <- ncol(y)
mlr <- lm(y~x)
sum <- summary(mlr)
sum
```

```
Response NO2 :

Call:
lm(formula = NO2 ~ x)

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

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)   
(Intercept) 10.11454    3.62607   2.789  0.00813 **
x                 NA         NA      NA       NA   
xWind       -0.21129    0.33917  -0.623  0.53694   
xSolarRad    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


Response O3 :

Call:
lm(formula = O3 ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.9527 -3.5053 -0.2998  1.4703 14.7123 

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  8.27619    5.58044   1.483   0.1461  
x                 NA         NA      NA       NA  
xWind       -0.78682    0.52198  -1.507   0.1398  
xSolarRad    0.09518    0.04761   1.999   0.0526 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.257 on 39 degrees of freedom
Multiple R-squared:  0.1513,	Adjusted R-squared:  0.1078 
F-statistic: 3.476 on 2 and 39 DF,  p-value: 0.04082
```

In [34]:
```
betaMat <- solve(t(x)%*%x) %*% t(x) %*% y
round(betaMat, 3)
```

	NO2	O3
	10.115	8.276
Wind	-0.211	-0.787
SolarRad	0.021	0.095

In [35]:
```
sigHat <- t(y-x%*%betaMat) %*% (y-x%*%betaMat)
round(sigHat, 1)
```

	NO2	O3
NO2	455.1	82.9
O3	82.9	1078.0

In [36]:
```
x1 <- x[,2]
x2 <- x[,3]
```

In [37]:
```
q <- 1
alpha = 0.05
betaMat1 <- solve(t(x1)%*%x1) %*% t(x1) %*% y
sigHat1 <- t(y-x1%*%betaMat1) %*% (y-x1%*%betaMat1)
Lambda1 <- det(sigHat)/det(sigHat1)
stat1 <- -(n-p-1-0.5*(m-p+q+1))*log(Lambda1)
crit <- qchisq(1-alpha,m*(p-q))
reject1 <- stat1 > crit
cat("Reject Null:", reject1)
```

```
Reject Null: TRUE
```

Conclusion: $\beta_2 = 0$ is not plausible

(5b)

In [38]:
```
q <- 1
betaMat2 <- solve(t(x2)%*%x2) %*% t(x2) %*% y
sigHat2 <- t(y-x2%*%betaMat2) %*% (y-x2%*%betaMat2)
Lambda2 <- det(sigHat)/det(sigHat2)
stat2 <- -(n-p-1-0.5*(m-p+q+1))*log(Lambda2)
crit <- qchisq(1-alpha,m*(p-q))
reject2 <- stat2 > crit
cat("Reject Null:", reject1)
```

```
Reject Null: TRUE
```

Conclusion: $\beta_1 = 0$ is not plausible

(5c)

In [39]:
```
q <- 2
betaMat3 <- lm(y~1)$coef
sigHat3 <- t(y-x[,1]%*%betaMat2) %*% (y-x[,1]%*%betaMat2)
Lambda3 <- det(sigHat)/det(sigHat3)
stat3 <- -(n-p-1-0.5*(m-p+q+1))*log(Lambda3)
crit <- qchisq(1-alpha,m*(p-q))
reject3 <- stat3 > crit
cat("Reject Null:", reject1)
```

```
Reject Null: TRUE
```

Conclusion: $\beta_1 = \beta_2 = 0$ is not plausible. This is consistent with (a) and (b).

In [ ]: