CODE APPENDIX

1. LDA and Two Sample Test

~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(pacman)
p_load(dplyr,table1,ggplot2, GGally, MASS, kableExtra, grid, gridExtra, klaR)
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
bankruptcy <- read.table("T11-4.DAT")
colnames(bankruptcy) <- c("x1","x2", "x3", "x4","population")
label(bankruptcy$x1) = "CF/TD"
label(bankruptcy$x2) = "NI/TA"
label(bankruptcy$x3) <- "CA/CL"
label(bankruptcy$x4) <- "CA/NS"
label(bankruptcy$population) <- "Population"

attach(bankruptcy)
bankruptcy$population = factor(bankruptcy$population)
bankruptcy$x1 = as.numeric(bankruptcy$x1)
bankruptcy$x2 = as.numeric(bankruptcy$x2)
bankruptcy$x3 = as.numeric(bankruptcy$x3)
bankruptcy$x4 = as.numeric(bankruptcy$x4)
## Plot (x1,x2)



ggpairs(bankruptcy[,1:4], aes(color = factor(population)), lower = list(continuous = wrap("smooth", alpha = 0.4, size = 0.3), discrete = "blank", combo="blank"), diag = list(discrete="barDiag", continuous = wrap("densityDiag", alpha=0.5 )), upper = list(combo = wrap("box_no_facet", alpha=0.5), continuous = wrap("cor", size=4, alignPercent=0.8))) + theme(panel.grid.major = element_blank()) + ggtitle("Summary plot for Bankruptcy data")


g2 <- ggplot(bankruptcy, aes(x = x1, y = x2, color = population)) + geom_point() +
  stat_ellipse(aes(x=x1, y= x2, color= x1),type = "norm") +
  theme(legend.position='none') +scale_color_manual(values = c("#00AFBB", "#E7B800"))
# ## Plot (x1,x3)
~~~

~~~
g3 <- ggplot(bankruptcy, aes(x = x1, y = x3, color = population)) + geom_point() +
  stat_ellipse(aes(x=x1, y= x3, color= x1),type = "norm") +
  theme(legend.position='none') +scale_color_manual(values = c("#00AFBB", "#E7B800"))
# ## Plot (x1,x4)
g4 <- ggplot(bankruptcy, aes(x = x1, y = x4, color = population)) + geom_point() +
  stat_ellipse(aes(x=x1, y= x4, color= x1),type = "norm") +
  theme(legend.position='none') +scale_color_manual(values = c("#00AFBB", "#E7B800"))
#
grid.arrange(g2,g3,g4, nrow = 2, ncol=2)
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE, include=FALSE}
lda.obj <- lda(population ~ x1+ x2,data=bankruptcy,prior=c(1,1)/2)

plda <- predict(object=lda.obj,newdata=bankruptcy)

# # Confusion matrix
table(population,plda$class)
#
# #plot the decision line
gmean <- lda.obj$prior %*% lda.obj$means

const <- as.numeric(gmean %*%lda.obj$scaling)

slope <- - lda.obj$scaling[1] / lda.obj$scaling[2]

intercept <- const / lda.obj$scaling[2]
#


par(mfrow = c(2,1))
# #Plot decision boundary
plot(bankruptcy[,1:2],pch=rep(c(18,20),each=50),col=rep(c(2,4),each=50))
abline(intercept, slope)
#legend("topright",legend=c("Alaskan","Canadian"),pch=c(18,20),col=c(2,4))
partimat(population~.,data = bankruptcy,method="lda",  main = "LDA Partition Plot")
~~~

~~~
```

```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
x1_col<- c(-0.069,0.235,3.271)
x2_col <- c(-0.081,0.055,3.367)
ldamat = data.frame(matrix(cbind(x1_col, x2_col), nrow = 3, ncol = 2))
colnames(ldamat) = c("x1","x2")
rownames(ldamat) = c("Group 1 mean", "Group 2 mean","LDA coefficients")
kable(ldamat, digits = 4, format = "pandoc", caption = "Table 1: LDA for x1 and x2")
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
x1_col<- c(-0.069,0.235,2.664)
x3_col <- c(1.3661,2.5936,0.8156)
ldamat = data.frame(matrix(cbind(x1_col, x2_col), nrow = 3, ncol = 2))
colnames(ldamat) = c("x1","x3")
rownames(ldamat) = c("Group 1 mean", "Group 2 mean","LDA coefficients")
kable(ldamat, digits = 4, format = "pandoc", caption = "Table 2: LDA for x1 and x3 ")
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
x1_col<- c(-0.069,0.235,4.6773)
x4_col <- c(0.4376,0.4268,0.01965)
ldamat = data.frame(matrix(cbind(x1_col, x2_col), nrow = 3, ncol = 2))
colnames(ldamat) = c("x1","x4")
rownames(ldamat) = c("Group 1 mean", "Group 2 mean","LDA coefficients")
kable(ldamat, digits = 4, format = "pandoc", caption = "Table 3: LDA for x1 and x4")
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
x2_col <- c(-0.081,0.055,5.496)
x3_col <- c(1.3661,2.5936,0.8896)
ldamat = data.frame(matrix(cbind(x1_col, x2_col), nrow = 3, ncol = 2))
colnames(ldamat) = c("x1","x4")
rownames(ldamat) = c("Group 1 mean", "Group 2 mean","LDA coefficients")
kable(ldamat, digits = 4, format = "pandoc", caption = "Table 4: LDA for x2 and x3")
```
~~~

~~~~
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
x2_col <- c(-0.081,0.055,9.62999)
x4_col <- c(0.4376,0.4268,-0.6980)
ldamat = data.frame(matrix(cbind(x1_col, x2_col), nrow = 3, ncol = 2))
colnames(ldamat) = c("x2","x4")
rownames(ldamat) = c("Group 1 mean", "Group 2 mean","LDA coefficients")
kable(ldamat, digits = 4, format = "pandoc", caption = "Table 5: LDA for x2 and x4")
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
partimat(population~.,data = bankruptcy,method="lda",  main = "LDA Partition Plot")
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
x1_col<- c(-0.069,0.235,0.66124)
x2_col <- c(-0.081,0.055,4.3935)
x3_col <- c(1.3661,2.5936,0.887250)
x4_col <- c(0.4376,0.4268,-1.178500)
ldamat = data.frame(matrix(cbind(x1_col,x2_col,x3_col, x4_col), nrow = 3, ncol = 4))
colnames(ldamat) = c("x1","x2","x3","x4")
rownames(ldamat) = c("Group 1 mean", "Group 2 mean","LDA coefficients")
kable(ldamat, digits = 4, format = "pandoc", caption = "Table 6: LDA for x1, x2, x3 and x4")
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
~~~~

Two sample

~~~~
bankrupt<- bankruptcy[bankruptcy$population == "0",-c(1,2,5)]
nonbankrupt <-bankruptcy[bankruptcy$population == "1",-c(1,2,5)]


# bankrupt <- iris[iris$Species == "setosa",-c(3,4,5)]
# nonbankrupt <- iris[iris$Species == "versicolor",-c(3,4,5)]
~~~~

~~~
# now we perform the two-sample Hotelling T^2-test
n<-c(dim(bankrupt)[1],dim(nonbankrupt)[1])
p<- dim(bankruptcy)[2] - 1
xmean1<-colMeans(bankrupt)
xmean2<-colMeans(nonbankrupt)
d<-xmean1-xmean2
S1<-var(bankrupt)
S2<-var(nonbankrupt)
Sp<-((n[1]-1)*S1+(n[2]-1)*S2)/(sum(n)-2)
t2 <- t(d)%*%solve(sum(1/n)*Sp)%*%d


alpha<-0.05
cval <- (sum(n)-2)*p/(sum(n)-p-1)*qf(1-alpha,p,sum(n)-p-1)

```

```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
# Confidence Region
es<-eigen(sum(1/n)*Sp)
e1<-es$vec %*% diag(sqrt(es$val))
r1<-sqrt(cval)
theta<-seq(0,2*pi,len=250)
v1<-cbind(r1*cos(theta), r1*sin(theta))
pts<-t(d-(e1%*%t(v1)))
plot(pts,type="l",main="Confidence Region for Bivariate Normal",xlab="CF/TD",ylab="NI/TA",asp=1)
segments(0,d[2],d[1],d[2],lty=2) # highlight the center
segments(d[1],0,d[1],d[2],lty=2)

th2<-c(0,pi/2,pi,3*pi/2,2*pi)   #adding the axis
v2<-cbind(r1*cos(th2), r1*sin(th2))
pts2<-t(d-(e1%*%t(v2)))
segments(pts2[3,1],pts2[3,2],pts2[1,1],pts2[1,2],lty=3)
segments(pts2[2,1],pts2[2,2],pts2[4,1],pts2[4,2],lty=3)
~~~

~~~
```

```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}

# since we reject the null, we use the simultaneous confidence intervals

# to check the significant components


# simultaneous confidence intervals

wd<-sqrt(cval*diag(Sp)*sum(1/n))

Cis<-cbind(d-wd,d+wd)

#Bonferroni simultaneous confidence intervals

wd.b<- qt(1-alpha/(2*p),n[1]+n[2]-2) *sqrt(diag(Sp)*sum(1/n))

Cis.b<-cbind(d-wd.b,d+wd.b)

# both component-wise simultaneous confidence intervals do not contain 0, so they have significant differences.

```
~~~

2. Principal Component Analysis

~~~
```{r setup, include=FALSE}

knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)

library(GGally)

library(kableExtra)

```

```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}

stock <-  read.table("T8-4.DAT")

colnames(stock) <- c("JP Morgan","Citibank", "Wells Fargo","Royal Dutch Shell","Exxon")

ggpairs(stock, lower = list(continuous = wrap("smooth", alpha = 0.4, size = 0.3), discrete = "blank", combo="blank"), diag = list(discrete="barDiag", continuous = wrap("densityDiag", alpha=0.5 )), upper = list(combo = wrap("box_no_facet", alpha=0.5), continuous = wrap("cor", size=4, alignPercent=0.8))) + theme(panel.grid.major = element_blank())

```

```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}

# Sample covariance matrix

scov <- cov(stock)
~~~

~~~
 scov_frame <- data.frame(scov)
kable(scov_frame, format = "pandoc", caption = "Covariance matrix Forbes Dataset")
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
#Principal components
stock.pca <- princomp(stock, cor=TRUE)
stock.pce <- data.frame(matrix(rbind(c(1.561, 1.18, 0.707, 0.632, 0.505 ), c(0.4874546, 0.2814025, 0.1001025, 0.08000632, 0.05103398), c(0.4874546, 0.7688572, 0.8689597, 0.94896602, 1.00000000)), nrow = 3, ncol = 5))
colnames(stock.pce) = c("comp 1", "comp 2", "comp 3","comp 4","comp 5")
rownames(stock.pce) = c("Standard deviation", "Proportion of Variance", "Cumulative Proportion ")
kable(stock.pce, digits = 4, format = "pandoc", caption = "Table 1: Principal components")
```


```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
#Scree plot to determine number of PC to use


# A scree plot:
plot(1:(length(stock.pca$sdev)),  (stock.pca$sdev)^2, type='b',
    main="Scree Plot", xlab="Number of Components", ylab="Eigenvalue Size")
day = factor(rep(c("M", "Tu","W","Th","F"),21))
day = day[1:dim(stock)[1]]
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
plot(stock.pca$scores[,1], stock.pca$scores[,2],
    xlab="PC 1", ylab="PC 2",  lwd=2, col=day)
legend("topright",legend=levels(day),pch=1,col=1:3,cex=0.7)
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
biplot(stock.pca,xlabs=day)
```
```{r echo=FALSE, message=FALSE, warning= FALSE, result=FALSE}
biplot(stock.pca, choices=3:4,xlabs=day)
```
~~~

3. Multiple Linear Regression

~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(pacman)
p_load(caret, GGally, kableExtra, dplyr, ellipse, gridExtra, grid)
```
```{r echo=FALSE, message=FALSE, result=FALSE,warning= FALSE}


company <- c("Citigroup", "General Electric", "American Intl Group", "Bank of America", "HSBC Group", "Exxon
Mobil", "Royal Dutch/ Shell", "BP", "ING Group", " Toyota")


sales <- c(108.28,152.36,95.04,65.45,62.97,263.99,265.19,285.06,92.01,165.68)


profits <- c(17.05,16.59,10.91,14.14,9.52,25.33,18.54,15.73,8.10,11.13)


assets <- c(1484.10,750.33,766.42,1110.46,1031.29,195.26,193.83,191.11,1175.16,211.15)


forbes <-  as.data.frame(matrix(cbind( sales, profits, assets), nrow = 10, ncol = 3))
colnames(forbes) <- c( "sales", "profits", "assets")
forbes$company <- company
forbes$sales <- as.numeric(forbes$sales)
forbes$profits <- as.numeric(forbes$profits)
forbes$assets <- as.numeric(forbes$assets)
```
```{r echo=FALSE, message=FALSE, result=FALSE,warning= FALSE}
forbes_summary <- summary(forbes[,1:3])
sales_sum <- c(62.97,92.77,130.32,155.60,239.41,
285.06)
profits_sum <- c(8.10,10.96,14.94,14.70,16.93,25.33)
assets_sum <- c(191.1,199.2,758.4,710.9,1090.7,1484.1)
five_forbes_summary <- as.data.frame(cbind(sales_sum,profits_sum,assets_sum ))
colnames(five_forbes_summary) = c("Sales","Profits","Assets")
rownames(five_forbes_summary) = c("Minimum", "1st Quartile", "Median", "Mean", "3rd Quartile", "Max")
kable(five_forbes_summary, format = "pandoc", caption = "Summary Statistics for Forbes Dataset")


layout(matrix(c(1,1, 2, 3), nrow = 2, ncol = 2, byrow = TRUE))
~~~

~~~
ggpairs(forbes[,1:3], lower = list(continuous = wrap("smooth", alpha = 0.4, size = 0.3), discrete = "blank", combo="blank"), diag = list(discrete="barDiag", continuous = wrap("densityDiag", alpha=0.5 )), upper = list(combo = wrap("box_no_facet", alpha=0.5), continuous = wrap("cor", size=4, alignPercent=0.8))) + theme(panel.grid.major = element_blank()) + ggtitle("Figure 1")


ggplot(forbes, aes( company, sales, color = company)) + geom_point() + theme( axis.text.x = element_blank(), axis.ticks = element_blank()) +ggtitle("Figure 2")

ggplot(forbes, aes( company, profits, color = company)) + geom_point() + theme( axis.text.x = element_blank(), axis.ticks = element_blank()) + ggtitle("Figure 3")


# lay <- c(1,2)

# grid.arrange(grobs=lapply(list(g2,g3),grobTree), layout_matrix = lay)

```


```{r echo=FALSE,include=FALSE, message=FALSE, result=FALSE,warning= FALSE}

model <- lm(profits~assets+sales)

par(mfrow = c(1,2))

plot(model, which = c(2,1), main = "Figure 4")


Z = matrix(cbind(rep(1, 10), assets,  sales), nrow = 10, ncol = 3)

Y =  matrix(profits, nrow = 10, ncol = 1)

n <- length(Y)

r <- dim(Z)[2]-1

```


```{r echo=FALSE, include=FALSE, message=FALSE, result=FALSE,warning= FALSE}

# least square estimates

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

```
~~~

explained by the model.

~~~
```{r echo=FALSE, include=FALSE, message=FALSE, result=FALSE}

# R^2 statistic

R_square <- 1 - sum((Y - Z%*%beta_hat)^2)/sum((Y-mean(Y))^2)

```

```{r echo=FALSE,include=FALSE, message=FALSE, result=FALSE,warning= FALSE}

# sigma_hat_square

sigma_hat_square <- sum((Y - Z%*%beta_hat)^2)/(n-r-1)

sigma_hat_square
~~~

~~~~
```
```{r echo=FALSE,include=FALSE, message=FALSE, result=FALSE}
# estimated covariance of hat{beta}
sigma_hat_square * solve(t(Z)%*%Z)
```

```{r echo=FALSE,include=FALSE, message=FALSE, result=FALSE,warning= FALSE}
# t-test for single coefficient
# H_0: beta_j = 0, H_a: beta_j != 0

j <- 1
t_stat <- (beta_hat[j+1] - 0)/sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1])
t_stat

alpha <- 0.05
cval_t <- qt(1-alpha/2, n-r-1)
cval_t
```

```{r echo=FALSE,include=FALSE, message=FALSE, result=FALSE,warning= FALSE}
# One-at-a-time confidence interval for beta_j

j <- 1
cat('[',
   beta_hat[j+1] - qt(1-alpha/2, n-r-1)*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1]),
   ',',
   beta_hat[j+1] + qt(1-alpha/2, n-r-1)*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1]),
   ']')
```

The 95% confidence intervals for $\beta_{j}$, $j$= 0,1,2 based on confidence region are $$[\beta_{j} -
\hat{\sigma^2}\sqrt{\omega_{11}}\sqrt{(r+1)F_{r+1,n-r-1}(0.05)}), \beta_{j} +
\hat{\sigma^2}\sqrt{\omega_{11}}\sqrt{(r+1)F_{r+1,n-r-1}(0.05)})]$$

$\beta_{0} \in [ -27.5812 , 27.60785\\]$
$\beta_{1} \in [ -0.0121 , 0.0236 ]\\$
$\beta_{2} \in [ -0.03251 , 0.1686 ]\\$'
~~~~

~~~
```{r echo=FALSE,include=FALSE, message=FALSE, result=FALSE,warning= FALSE}

# confidence region based simultaneous confidence intervals


j <- 0

cat('[',

   beta_hat[j+1] - sqrt((r+1)*qf(1-alpha, r+1, n-r-1))*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1]),

   ',',

   beta_hat[j+1] + sqrt((r+1)*qf(1-alpha, r+1, n-r-1))*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1]),

   ']')

```
~~~