Assignment #3

Janelle Morano

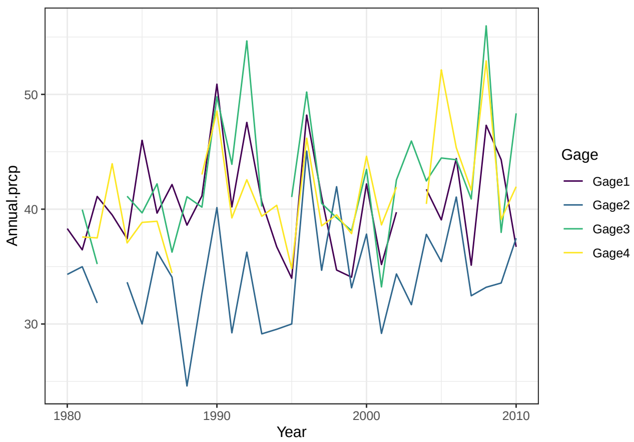
3/17/2022

Assess Normality

a)

```
#### a) Create a matrix (annual.prcp) only containing the precipitation data at
#### the four gages (not the years). Convert this matrix from a data.frame
#### object (what read.table() returns) into adata.matrix object. Data matrices
#### are better suited for matrix multiplication.
annual.prcp.df <- read.table("/Users/janellemorano/Git/Reference-R-scripts/Envtl-Multivariate-Stats/ass
    header = TRUE)
annual.prcp <- as.matrix(annual.prcp.df[, 2:5])

# Plot annual precipitation over time for understanding of trends
library(tidyverse)
annual.prcp.long <- gather(annual.prcp.df, Gage, Annual.prcp, Gage1:Gage4, factor_key = TRUE)

library(viridis)
ggplot(annual.prcp.long, aes(x = Year, y = Annual.prcp, colour = Gage, group = Gage)) +
    geom_line() + theme_bw() + scale_color_viridis(discrete = TRUE)
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```

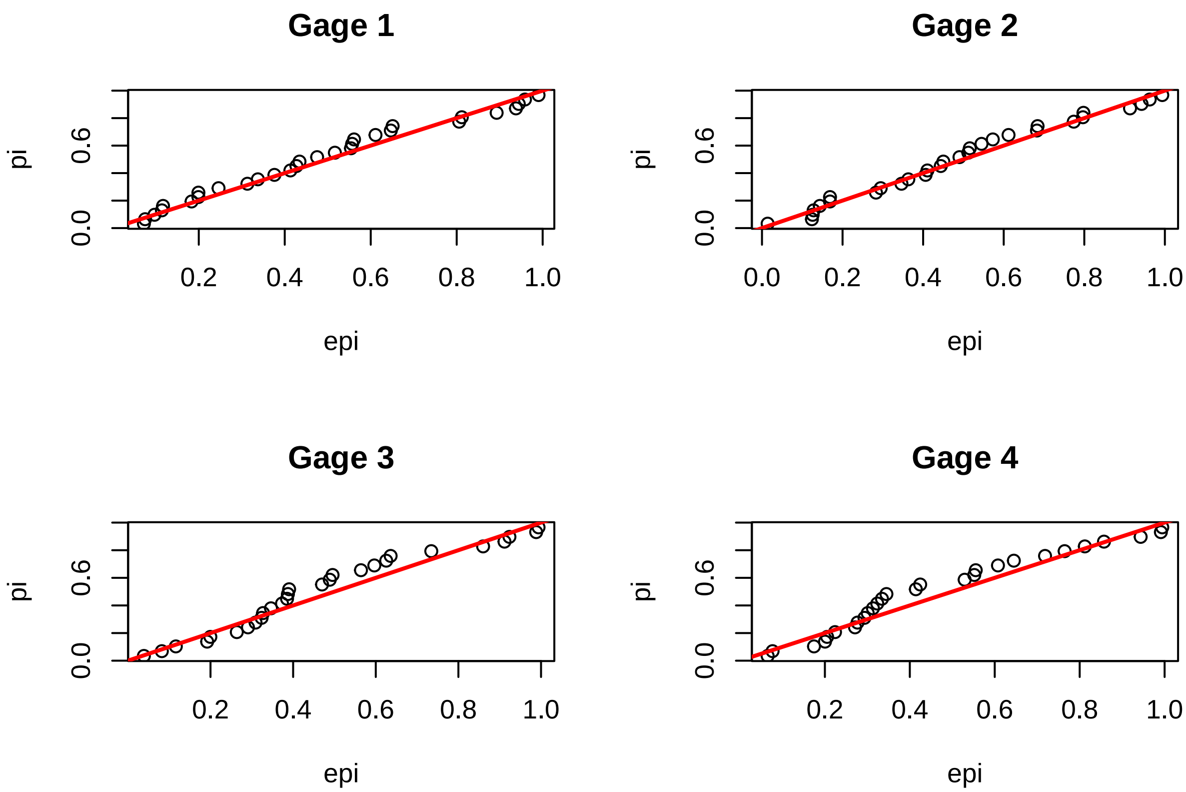


Check Marginal Distributions

b)

```
##### b) Check marginal distributions
par(mfrow = c(2, 2))
for (i in 1:4) {
    # Sort data
```

```
    x <- sort(annual.prcp[, i], na.last = NA)
    # Calculate pi=1/(n+1)
    pi <- c((1:length(x))/(length(x) + 1))
    # Find mean and standard deviation of data with pnorm()
    epi <- pnorm(x, mean(x), sd(x))
    # Plot probability plots (P-P) of the data for all four gages
    plot(epi, pi, main = paste0("Gage ", i))
    abline(a = 0, b = 1, col = "red", lwd = 2)
}
```




```
par(mfrow = c(1, 1))
```

The marginal distributions (i.e. the distribution of the variables: annual precipitation at each gage) for appear normally distributed, although there is more variance from the mean on gages 3 & 4.

Check Multivariate Structure

c)

```
#### c) Calculate the covariance matrix of the precipitation data matrix using
#### the cov() function
S <- cov(annual.prcp, use = "pairwise.complete")
# Calculate mean vector mu
mu <- apply(annual.prcp, MARGIN = 2, FUN = mean, na.rm = TRUE)
```

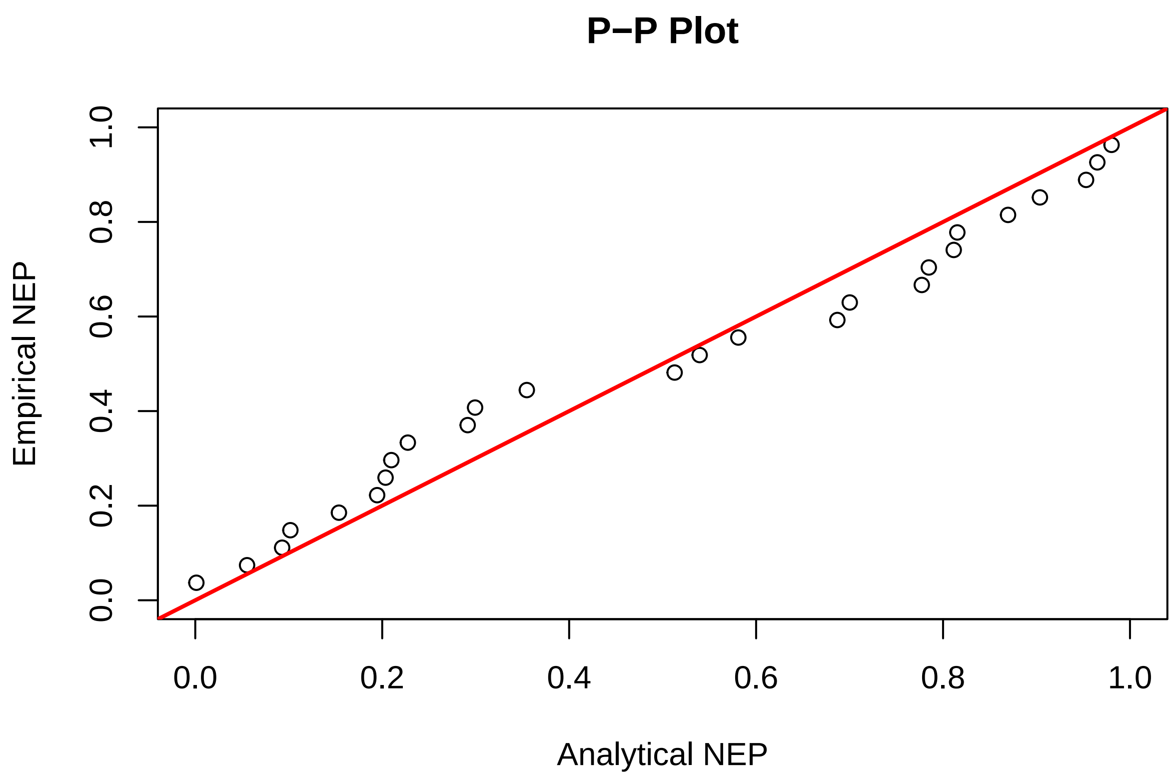
d)

```
#### d) Manahalobis distance between each of 31 observations and the mean D2 =
#### ((xi-mu)^T) * S^-1 * (x-mu) D2 <- t(xi - mu) %*% solve(S) %*% (xi - mu) xi
#### is a row of observations across each of 4 gages
```

```r
rows = nrow(annual.prcp)  #number of observations = number of rows
D.sq <- c()

for (i in 1:rows) {
    # subset data for each row
    xi <- unname(annual.prcp[i, ])
    # calculate Dsq for each row, where mu is defined above
    D2 <- t(xi - mu) %*% solve(S) %*% (xi - mu)
    # Export data
    D.sq[i] = D2
}
```

e)

```r
#### e) P-P plot for D.sq (empirical/model-free non-exceedence probabilities
#### (NEP) vs. analytical/model-based NEP.

# 1) Fit normal dist to data, ie find mean and sd
D.sq.mean = mean(D.sq)
D.sq.sd = sd(D.sq)
df = ncol(annual.prcp)  #should be 4

# 2) Sort the data, use sort() function to order a vector of data from smallest
# to largest, and set the argument na.last=NA to drop the NAs
D.sq = sort(D.sq, na.last = NA)

# 3) Calculate analytical/model-based NEP
NEP.analytical.D.sq <- pchisq(D.sq, df)

# 4) Calculate empirical/model-free NEP
NEP.empirical.D.sq <- (1:length(D.sq))/(length(D.sq) + 1)

# 5) Plot empirical v analytical
plot(NEP.analytical.D.sq, NEP.empirical.D.sq, xlim = c(0, 1), ylim = c(0, 1), xlab = "Analytical NEP",
    ylab = "Empirical NEP", main = "P-P Plot")
abline(a = 0, b = 1, col = "red", lwd = 2)
```



f) Based on the marginal distributions and the P-P plot, a multivariate normal distributionn (MVN) is appropriate to model the stochastic behavior of annual precipitation at the 4 gages. There is some deviation, but the MVN distribution is appropriate.

2) Gap-Filling Missing Data (Imputation)

g)

```
#### g) Use conditional normality for MVN variable to estimate precip data for
#### each year and gage with missing data using all available obs that year.

# Steps to find Expected Value E = mu.na + S12 * (S22)^-1 * (x.obs - mu.obs) 1)
# Go row by row, find the rows with the NA, then find the columns in the row
# with NA 2) For each column with NA in the row...  3) ...Get the mu for the
# column with the NA, X^T = [mu1, mu2, mu3, mu4] 4) ...Then get the mu for the
# other columns with observations, X^T = [mu1, mu2, mu3, mu4] 5) ...Then get
# the observed values in the columns that have data 6) ???Calculate and split
# the covariance matrix based on the NAs 7) Find the expected value of the NAs
# and replace the NA

# Run loop on data to find NAs and calculate expected values
i <- 1
# make copy of data
annual.prcp.fill <- annual.prcp
# create empty vectors for output
i.id.all <- c()
j.na.all <- c()
means.na.all <- c()
var.na.all <- c()
CI.upper.all <- c()
```

```r
CI.lower.all <- c()

for (i in 1:length(annual.prcp[, 1])) {
    # 1) Go row by row, find the rows with the NA, then find the columns in the
    # row with NA
    if (sum(is.na(annual.prcp[i, ])) > 0) {
        # ID columns in row with NA and put in vectors
        i.id <- c(i)
        i.id.all <- c(i.id)  #append to vector
        j.na <- which(is.na(annual.prcp[i, ]))
        j.obs <- which(!is.na(annual.prcp[i, ]))
        j.na.all <- c(j.na)  #append to vector

        # 2) For each column with NA in the row...  3) ...Get the mu for the
        # column with the NA, X^T = [mu1, mu2, mu3, mu4]
        mu.na <- mu[j.na]
        # 4) ...Then get the mu for the other columns with observations, X^T =
        # [mu1, mu2, mu3, mu4]
        mu.obs <- mu[j.obs]  #also could use setdiff(mu, mu.na)
        # 5) ...Then get the observed values in the row in the columns that
        # have data
        x.obs <- annual.prcp[i, j.obs]

        # 6) Calculate and split the covariance matrix, getting the S12 (upper
        # right quadrant) that represents the row with no data and the column
        # with data and the S22 (lower right quadrant) that represents the row
        # and column with data
        S11 <- S[j.na, j.na]  #no data for row and column
        S12 <- S[j.na, j.obs]  #row index without data, col index with data
        S21 <- S[j.obs, j.na]  #row index with data, col index no data
        S22 <- S[j.obs, j.obs]  #row index with data, col index of data

        # 7) Find the expected value of the NAs (need to deal with the NAs in
        # the calculation)
        x.na <- mu.na + S12 %*% solve(S22) %*% (x.obs - mu.obs)
        # replace this vector back into the dataset with i and j index
        annual.prcp.fill[i, j.na] <- x.na

        # 8) Find the variance
        var.na <- S11 - S12 %*% solve(S22) %*% S21
        var.na <- diag(var.na)

        # 10) Report the estimates and 95% confidence bounds of each missing
        # data point Calculate 95% confidence bounds of each missing data point
        # CI = 1.96 * sqrt(conditional variance or SD)
        margin <- 1.96 * sqrt(var.na)
        lower <- x.na - margin
        upper <- x.na + margin

        # Append estimates, variances, and CI values
        means.na.all <- c(means.na.all, x.na)
        var.na.all <- c(var.na.all, var.na)
        CI.lower.all <- c(CI.lower.all, lower)
```

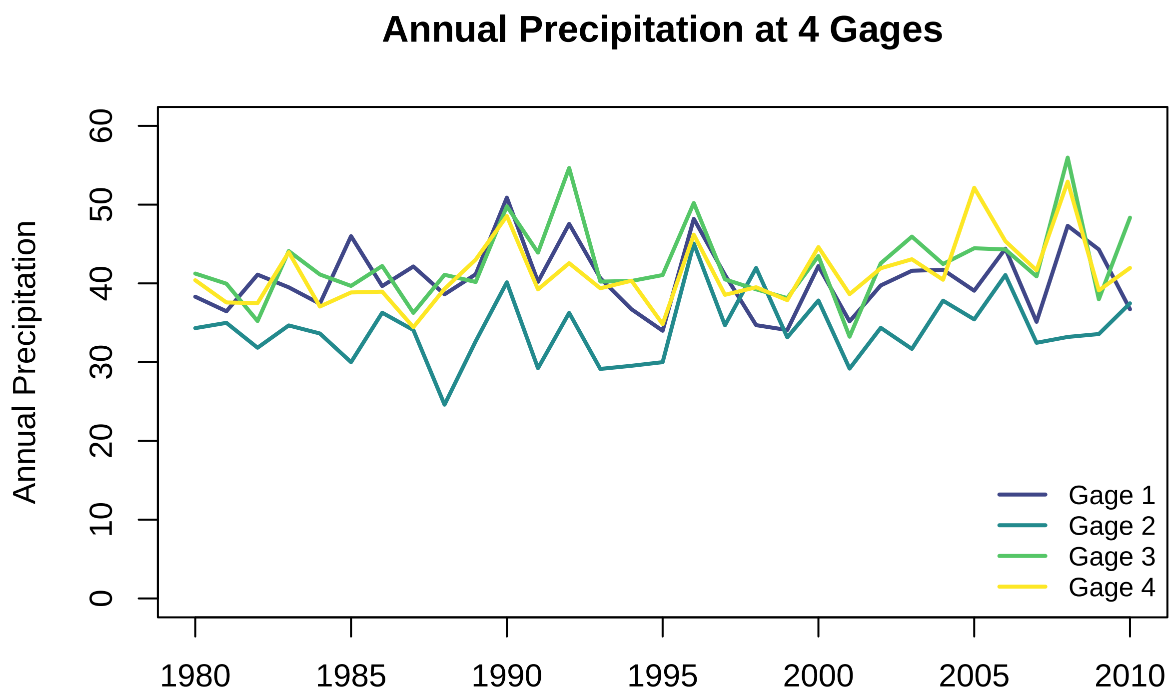
```
        CI.upper.all <- c(CI.upper.all, upper)

    }
}
NA.means <- data.frame()
NA.means <- rbind(NA.means, data.frame(means.na.all, var.na.all, CI.lower.all, CI.upper.all))
print(NA.means)
```

```
##   means.na.all var.na.all CI.lower.all CI.upper.all
## 1     41.25017  16.658406     33.25049     49.24985
## 2     40.40134  14.105202     33.04019     47.76249
## 3     34.66076  15.263370     27.00336     42.31816
## 4     44.11050  10.173973     37.85876     50.36225
## 5     39.30848   8.541471     33.58023     45.03674
## 6     40.30645  10.087575     34.08131     46.53160
## 7     41.60449  11.915943     34.83867     48.37030
## 8     43.06810   8.601530     37.31974     48.81646
```

3) Multivariate Inference

i)

```
# i) Plot gape-filled data as annual precipitation over time for each gage

annual.prcp.fill <- cbind(annual.prcp.fill, as.matrix(annual.prcp.df[, 1]))
plot(annual.prcp.fill[, 5], annual.prcp.fill[, 1], type = "l", col = "#404788FF",
    lwd = 2, ylim = c(0, 60), xlab = "", ylab = "Annual Precipitation", main = "Annual Precipitation at
lines(annual.prcp.fill[, 5], annual.prcp.fill[, 2], type = "l", lwd = 2, col = "#238A8DFF")
lines(annual.prcp.fill[, 5], annual.prcp.fill[, 3], type = "l", lwd = 2, col = "#55C667FF")
lines(annual.prcp.fill[, 5], annual.prcp.fill[, 4], type = "l", lwd = 2, col = "#FDE725FF")
legend("bottomright", box.lty = 0, lty = 1, cex = 0.8, lwd = 2, legend = c("Gage 1",
    "Gage 2", "Gage 3", "Gage 4"), col = c("#404788FF", "#238A8DFF", "#55C667FF",
    "#FDE725FF"))
```



```r
# Linear regression of Gage X against years
lm.gage1 <- lm(annual.prcp.fill[, 1] ~ annual.prcp.fill[, 5])
summary(lm.gage1)
```

```
##
## Call:
## lm(formula = annual.prcp.fill[, 1] ~ annual.prcp.fill[, 5])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5214 -3.4081 -0.2892  1.6719 10.4064
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.942e+01  1.783e+02   0.165    0.870
## annual.prcp.fill[, 5] 5.559e-03  8.938e-02   0.062    0.951
##
## Residual standard error: 4.451 on 29 degrees of freedom
## Multiple R-squared:  0.0001333,  Adjusted R-squared:  -0.03434
## F-statistic: 0.003868 on 1 and 29 DF,  p-value: 0.9508
```

```r
lm.gage2 <- lm(annual.prcp.fill[, 2] ~ annual.prcp.fill[, 5])
summary(lm.gage2)
```

```
##
## Call:
## lm(formula = annual.prcp.fill[, 2] ~ annual.prcp.fill[, 5])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9121 -3.0709  0.2417  2.2608 10.7591
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -162.50707  171.15430  -0.949     0.35
## annual.prcp.fill[, 5]    0.09860    0.08579   1.149     0.26
##
## Residual standard error: 4.272 on 29 degrees of freedom
## Multiple R-squared:  0.04357,    Adjusted R-squared:  0.01058
## F-statistic: 1.321 on 1 and 29 DF,  p-value: 0.2598
```

```r
lm.gage3 <- lm(annual.prcp.fill[, 3] ~ annual.prcp.fill[, 5])
summary(lm.gage3)
```

```
##
## Call:
## lm(formula = annual.prcp.fill[, 3] ~ annual.prcp.fill[, 5])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1583  -2.2030  -0.6026   1.4378  12.5433
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -239.3215   200.0127  -1.197    0.241
```

```
## annual.prcp.fill[, 5]    0.1413     0.1003   1.409    0.169
##
## Residual standard error: 4.993 on 29 degrees of freedom
## Multiple R-squared:  0.06409,    Adjusted R-squared:  0.03182
## F-statistic: 1.986 on 1 and 29 DF,  p-value: 0.1694
```

```
lm.gage4 <- lm(annual.prcp.fill[, 4] ~ annual.prcp.fill[, 5])
summary(lm.gage4)
```

```
##
## Call:
## lm(formula = annual.prcp.fill[, 4] ~ annual.prcp.fill[, 5])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4583 -2.3179 -0.7523  1.9845  9.0828
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -349.86464  161.67333  -2.164   0.0388 *
## annual.prcp.fill[, 5]    0.19607    0.08104   2.419   0.0220 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.036 on 29 degrees of freedom
## Multiple R-squared:  0.1679, Adjusted R-squared:  0.1393
## F-statistic: 5.854 on 1 and 29 DF,  p-value: 0.02204
```

```
# Function to grab p-values from lm summaries
pval <- function(modelobject) {
    if (class(modelobject) != "lm")
        stop("Not an object of class 'lm' ")
    f <- summary(modelobject)$fstatistic
    p <- pf(f[1], f[2], f[3], lower.tail = F)
    attributes(p) <- NULL
    return(p)
}

# P-value for gages 1-4
pval(lm.gage1)
```

```
## [1] 0.9508379
```

```
pval(lm.gage2)
```

```
## [1] 0.2598144
```

```
pval(lm.gage3)
```

```
## [1] 0.1693994
```

```
pval(lm.gage4)
```

```
## [1] 0.02204457
```

Gage 4 has a significant positive relationship between annual precipitation and year ($p < 0.05$), where the annual precipitation is increasing over time. Gages 1-3 do not have a significant relationship with years.

j)

```r
# j) Partition the data into two segments: first 15 years of data, and the
# other 16 years of data
annual.prcp.fill.1980.1994 <- annual.prcp.fill[1:15, ]
annual.prcp.fill.1995.2010 <- annual.prcp.fill[16:31, ]
```

k)

```r
# k) Calculate the multivariate mean for each of these subsets.  delta.x =
# xbar.2 - xbar.1 = 0?
xbar.1 <- apply(annual.prcp.fill.1980.1994[, 1:4], MARGIN = 2, FUN = mean)
xbar.2 <- apply(annual.prcp.fill.1995.2010[, 1:4], MARGIN = 2, FUN = mean)
delta.x <- xbar.2 - xbar.1
print(delta.x)
```

```
##     Gage1     Gage2     Gage3     Gage4
## -1.128720  2.798200  1.048983  2.347601
```

l)

```r
# l) Calculate the covariance matrix for each of these subsets.
cov.delta.x.1 <- cov(annual.prcp.fill.1980.1994[, 1:4])
print(cov.delta.x.1)
```

```
##           Gage1     Gage2     Gage3     Gage4
## Gage1 16.897797  6.846465 11.449933  7.592181
## Gage2  6.846465 14.546623  8.423151  5.421608
## Gage3 11.449933  8.423151 23.501912 11.443409
## Gage4  7.592181  5.421608 11.443409 11.444767
```

```r
cov.delta.x.2 <- cov(annual.prcp.fill.1995.2010[, 1:4])
print(cov.delta.x.2)
```

```
##           Gage1     Gage2     Gage3     Gage4
## Gage1 21.883259  8.789708 15.946155 13.813922
## Gage2  8.789708 19.278318  8.245653  6.318688
## Gage3 15.946155  8.245653 28.989683 19.722876
## Gage4 13.813922  6.318688 19.722876 24.317156
```

m)

```r
# m) Estimate the pooled covariance matrix based on the two covariance matrices
# above.  Var(delta.x) = ( (n1 - 1)*cov.n1 + (n2-1)*cov.n2 ) / (n1 + n2 -2)
n1 <- length(annual.prcp.fill.1980.1994[, 1])
n2 <- length(annual.prcp.fill.1995.2010[, 1])
var.pooled <- (1/n1 + 1/n2) * (((n1 - 1)/(n1 + n2 - 2)) * cov.delta.x.1 + ((n2 -
    1)/(n1 + n2 - 2)) * cov.delta.x.2)
print(var.pooled)
```

```
##          Gage1     Gage2    Gage3     Gage4
## Gage1 2.515713 1.0141638 1.779344 1.3963334
## Gage2 1.014164 2.1950650 1.076132 0.7602253
## Gage3 1.779344 1.0761316 3.402303 2.0312610
## Gage4 1.396333 0.7602253 2.031261 2.3382911
```

n)

```r
# n) Calculate and report the Hoteling-T statistic that compares the two means
# of these different subsets.
```

```
T.sq = t(delta.x) %*% solve(var.pooled) %*% t(t(delta.x))  #t(t(delta.x)) to get the dimensions correct
print(T.sq)
```

```
##          [,1]
## [1,] 11.04961
```

```
# Then calculate and report the F-statistic used in the F-test to compare these
# mean vectors (a scaled version the Hoteling-T statistic).  K = degrees of
# freedom = 1? or 3?  F = (n1+n2-k-1)/((n1+n2-2)*k) * T.sq
k <- 4
F.test <- ((n1 + n1 - k - 1)/((n1 + n2 - 2) * k)) * T.sq
print(F.test)
```

```
##          [,1]
## [1,] 2.381381
```

o)

```
# o) Compare this statistic to a 95% critical value of the F-distribution with
# appropriate degrees of freedom.
df <- n1 + n2 - k - 1
qf(0.95, k, df)
```

```
## [1] 2.742594
```

The null hypothesis (the difference in precipitation means between the 2 time periods is equal to 0, or there is no difference) can be rejected because the adjusted T^2 value is NOT approximately equal to the F-distribution. This tells us that there is a change in precipitation over time, but it does not tell us if there is a specific gage or gages that are driving this difference. The individual linear models tells us that it is Gage 4 that is driving this difference. Both approaches tell us something important about the system, but the approaches are asking different questions and testing different hypotheses. The linear model addresses the relationship between annual precipitation and time, and the hypothesis testing approach addresses if there is a difference between 2 time periods.