MATH 537 EXAM II

Gustavo Esparza

7/23/2019

1

A

Before performing a statistical test for a difference in mean vectors for the 6 Apple Tree groups, we will review the mean vectors first.

Group	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_4$
1	1.1375	2.977125	3.73875	0.871125
2	1.1575	3.109125	4.515	1.2805
3	1.1075	2.81525	4.455	1.391375
4	1.0975	2.87975	3.90625	1.039
5	1.08	2.55725	4.3125	1.181
6	1.036250	2.214625	3.596250	0.735000

We can see that there are some differences across the several mean vectors, so there is some reason to believe that there is a statistically significant difference between the means for our six rootstock groups.

Now, we will begin our statistical test, using the MANOVA test with Wilk's Lambda as our test statistic. Here is the hypothesis statement for our test:

$H_o : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6$ VS H_a : At least one μ is different from the other μ_i
$\alpha = .05$
Test Statistic: Wilk's Λ that is defined as follows : $\frac{|SS_w|}{|SS_t|}$, where SS_w is the Sum of Squares within the data and SS_t is the Sum of Squares total. This Wilk's Λ is then transformed into an F statistic for computating a p-value.

Here is our MANOVA test result with Wilk's Lambda as our test of choice:

```
##           Df   Wilks approx F num Df den Df    Pr(>F)
## Rootstock  5 0.15401   4.9369     20  130.3 7.714e-09 ***
## Residuals 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that our Λ value is .15401, which is then computed to be an F value of 4.9369. This corresponds to a p-value that is ≈ 0. Thus, we proceed to reject our null hypothesis.

When considering the 4 qualities (Trunk Girth, Extension Growth, Trunk Girth at 15 years, Weight) of our 6 distinct tree groups, we have strong statistical evidence to suggest that the mean vectors four our groups are not all the same as one another.

B

Now, we will be using Pillai's Trace instead of Λ for the same hypothesis test. We will first make some distinctions between Wilk's Lambda and Pillai's Trace:

* Pillai's Trace is computed using the Sum of Squares in a similar fashion to Wilk's Lambda. In particular we have $V = Trace((SS_T)^{-1} * (SS_T - SS_W))$

* Pillai's Trace is a positive valued statistic, while Wilk's Lambda is restricted between 0 and 1.

* A higher Pillai's Trace value indicates statistical significance, while a smaller Wilk's Lambda value indicates statistical signficance.
This fact implies that Wilk's Lambda is similar to a Likelihood Test, and Pillai's Trace is interpreted like a Z-score distance.

In addition to these key differences, it also important to note that Pillai's Trace is more effecient when the assumptions of Multivariate Normality and Equal Variance/covariance are violated and Wilk's Lambda is not as reliable.

Now, we will begin our statistical test, using the MANOVA test with Pillai's Trace as our test statistic. Here is the hypothesis statement for our test:

$H_o : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6$ VS H_a : At least one μ is different from the other μ_i
$\alpha = .05$
Test Statistic: Pillai's Trace that is defined as follows : $Trace((SS_T)^{-1} * (SS_T - SS_W))$, where SS_w is the Sum of Squares within the data and SS_t is the Sum of Squares total. This Trace value is then transformed into an F statistic for computating a p-value.

Here is our MANOVA test result with Pillai's Trace as our test of choice:

```
##           Df Pillai approx F num Df den Df    Pr(>F)
## Rootstock  5 1.3055   4.0697     20    168 1.983e-07 ***
## Residuals 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that our trace value is 1.3055, which is then computed to be an F value of 4.0697. This corresponds to a p-value that is ≈ 0. Thus, we proceed to reject our null hypothesis.

Again, when considering the 4 qualities (Trunk Girth, Extension Growth, Trunk Girth at 15 years, Weight) of our 6 distinct tree groups, we have strong statistical evidence to suggest that the mean vectors for our rootstocks are not all the same as one another.

C

We have just seen two similar results for our hypothesis test using Wilk's Lambda and Pillai's Trace, with similar results. We still need to review our assumptions for MANOVA before completely relying on these results. We will proceed to do so, assumption by assumption.

Independence

We will not be doing any formal test for independence, but we know that the data for each group of trees is coming from a different rootstock. Although we know the rootstocks are different, we are not given much information regarding other factors for the region where the trees were selected in terms of other variates that may be affecting trees within the rootstocks.
Although we are assuming independence for the MANOVA computations, we can not say with great certainty that these trees were independently and randomly selected due to possible underlying covariates within the trees and the region in which they were located.

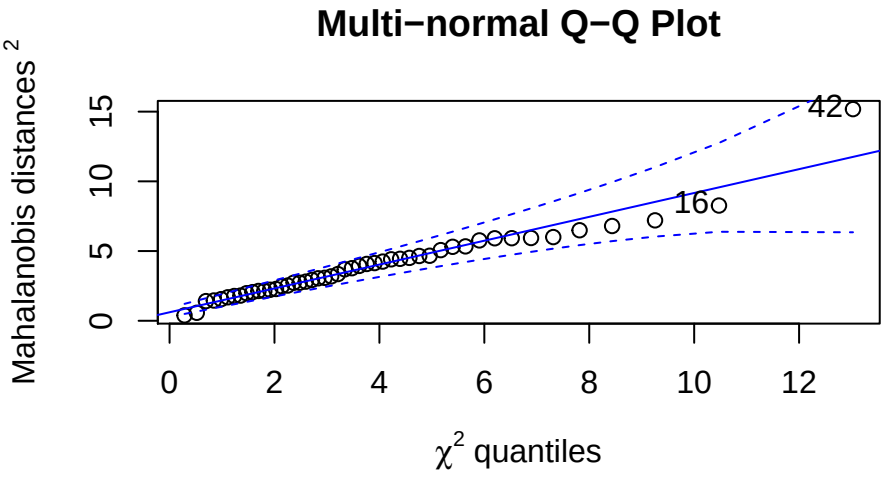
Multivariate Normality

We need multivariate normality for our data before performing Manova, so we will conduct a shapiro test first and also provide a graphical QQ plot.

Shapiro Test For Normality:
Here, we are using the null hypothesis that the data is multivariate normal and any statistical significance would prove otherwise. Here is our result:

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.92118, p-value = 0.003272
```

We can see that our p-value is quite small, which implies that there is evidence to suggest our data is not actually multivariate normal. From research, this test can tend to imply non-normality very easily, so we will also provide a qq plot to judge normality in another manner:



Using mqqorm, we are able to provide a qq plot of our data. We do see most points lying on our qq line at first, but there is eventually some consistent deviation that results in two extreme values at the tail end,

which does imply non-normality. Ultimately, we can say that our distribution looks somewhat normal, but there is still some doubt to our assumption.

Homogeneity of Covariances

We can use the BoxM function to test for Homegeneity of Variance/Covariance, which is essential to our MANOVA test. Since our sample is not extremely large, we should make sure this assumption is met.

The numm hypothesis for this test is that we have homogeneity, while statistical significance would prove otherwise.

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  Trees[, -1]
## Chi-Sq (approx.) = 44.018, df = 50, p-value = 0.711
```

We can see that our Box M-test does provide a large p value, that leads us to conclude that the data does in fact have homogeneity of covariance matrices.

Thus, we can see there is some evidence to suggest that the usual MANOVA assumptions are not clearly met. Although our qq plot did not have extreme violations, there was some deviation (in addition to the shapiro test) that leads us to believe that non-normality is a factor to be seriously considered before taking the MANOVA results at face value.

D

Wilk's Lambda Bootstrap

For the Wilk's Lambda bootstrapping process, we will need a function to compute the Λ value for each of our bootstrapped datasets. The details of the function can be found in the Appendix.

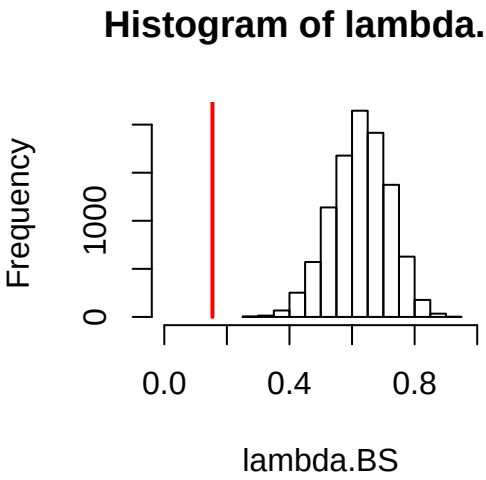
We will verify our function by computing the observed Wilk's Lambds value of our original Tree data:

```
## [1] 0.1540077
```

We can note that this computed Λ value is nearly identical to the value provided in our MANOVA result, thus we have verified our Lambda function.

Now, we will bootstrap our Lambda values 10,000 times and compare them to our observed value.

Here is a histogram of our bootstrapped Lambdas, along with a line indicating our observed Lambda:



We can see that nearly all bootstrapped values are above our observed Lambda, which implies that we have statistical evidence to suggest there is a difference between our 6 rootstock mean vectors.

```
## P value =  0
```

This p-value further establishes our claim of difference in means.

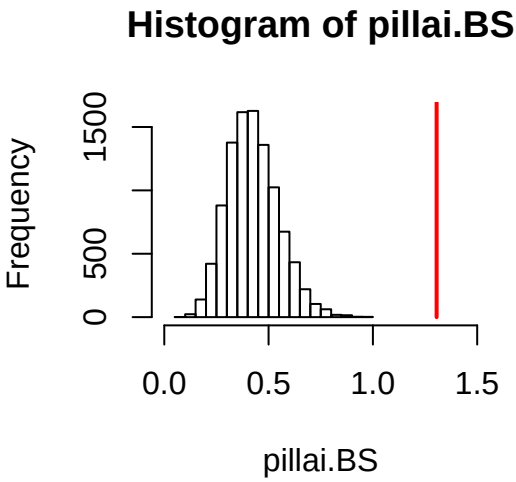
Pillai's Trace Bootstrap

For the Pillai's Trace bootstrapping process, we will need a function to compute the *trace* value for each of our bootstrapped datasets. The required function can be found in our Appendix.

We will verify our function by computing the observed Wilk's Lambds value of our original Tree data:

```
## [1] 1.305472
```

We can note that this computed *trace* value is nearly identical to the value provided in our MANOVA result, thus we have verified our Trace function.
Now, we will bootstrap our Trace values 10,000 times and compare them to our observed value.



We can see that nearly all bootstrapped values are below our observed trace value, which implies that we have statistical evidence to suggest there is a difference between our 6 rootstock mean vectors.

```
## P value =  0
```

This p-value further establishes our claim of difference in means.

Discussion

We have seen the same result for all three of our methods. We can see that the MANOVA method is extremely simple to calculate and make conclusions from. However, we are risking statistical significance when faced with violations of the MANOVA test. Although Pillai's Trace is more reliable when sosme of these assumptions are violated, we know that Bootstrapping will always be benificial regardless of the violations occuring. Bootstrapping in this case always provided extremely small p-values regardless of how many times our process was computed.

2

A

Original Model

Here is the summary of our full Multinomial Logistic Regression model

```
## Call:
## multinom(formula = FaceResponse ~ ., data = fullmodel)
##
## Coefficients:
##   (Intercept)        xF       yF        wF       hF       xRE       yRE
## 2    10.38748 -89.65076 78.98405 -47.15394 20.09614 -65.27159 -89.18336
## 3   -86.29002 -90.48307 78.41401 -47.12165 20.18279 -64.03784 -88.64829
##         xLE       yLE       xN        yN       xRM      yRM      xLM
## 2 -61.69195 -32.73307 234.6328 -107.9025 -86.10134 81.98117 51.81217
## 3 -62.56950 -32.92945 235.1140 -107.5433 -85.21284 80.99595 51.19664
##        yLM
## 2 68.52441
## 3 69.25024
##
## Std. Errors:
##   (Intercept)       xF        yF        wF        hF       xRE       yRE
## 2 0.001193671 0.177941 0.2638379 0.1342265 0.1022427 0.3074845 0.2302183
## 3 0.001193671 0.177941 0.2638379 0.1342265 0.1022427 0.3074845 0.2302183
##         xLE       yLE        xN        yN       xRM       yRM       xLM
## 2 0.2418206 0.1623031 0.1747753 0.2080282 0.1933215 0.3195582 0.2394856
## 3 0.2418206 0.1623031 0.1747753 0.2080282 0.1933215 0.3195582 0.2394856
##         yLM
## 2 0.2453609
## 3 0.2453609
##
## Residual Deviance: 19.56918
## AIC: 79.56918
```

```
## BIC =  211.7756
```

From our model summary, we can see the log odds for responses 2 and 3 for each predictor provided. In addition, we can see our AIC value is 79.56. Rather than only use this value to compare to our eventually reduced Principal Component model, we will also be using BIC (211.7756) as an additional measure of capturing information from our response. This is due to the fact models with a large amount of variables tend to provide more accurate information when using the Likelihood calculation found in BIC as opposed to AIC. We will be comparing both AIC and BIC when reducing our dimensions of predictors after Principal Component Analysis.

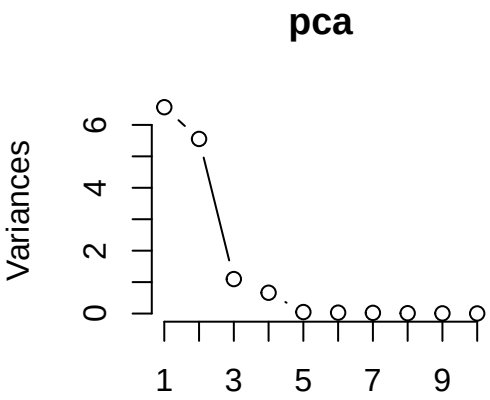
PCA

Now, we will be performing a full PCA analysis of our predictor variables to predict our response of the gaze direction. First and foremost, we will take all of our 14 predictor variables and compute our PCA analysis:

Here is a general summary of our PCA:

```
## Importance of components:
##                            PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.5618 2.3566 1.04734 0.81383 0.21737 0.17005
## Proportion of Variance 0.4688 0.3967 0.07835 0.04731 0.00337 0.00207
## Cumulative Proportion  0.4688 0.8655 0.94381 0.99112 0.99449 0.99656
##                            PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.14855 0.10291 0.07865 0.05919 0.04784 0.04182
## Proportion of Variance 0.00158 0.00076 0.00044 0.00025 0.00016 0.00012
## Cumulative Proportion  0.99814 0.99889 0.99933 0.99958 0.99975 0.99987
##                           PC13    PC14
## Standard deviation     0.03265 0.02662
## Proportion of Variance 0.00008 0.00005
## Cumulative Proportion  0.99995 1.00000
```

Immediatly, we can see that the first two components provide 47 and 40 percent of the variance in our data. This is then followed by component 3, which drops dramatically to a contribution of about 8 percent of the variance in data.
We can model this contribution trend in a linear plot:



The dramatic drop is even more apparent when displayed in this linear fashion. At this point, I would only keep the first three components, as the fourth component is very low and does not appear to contribute much to the overal model.

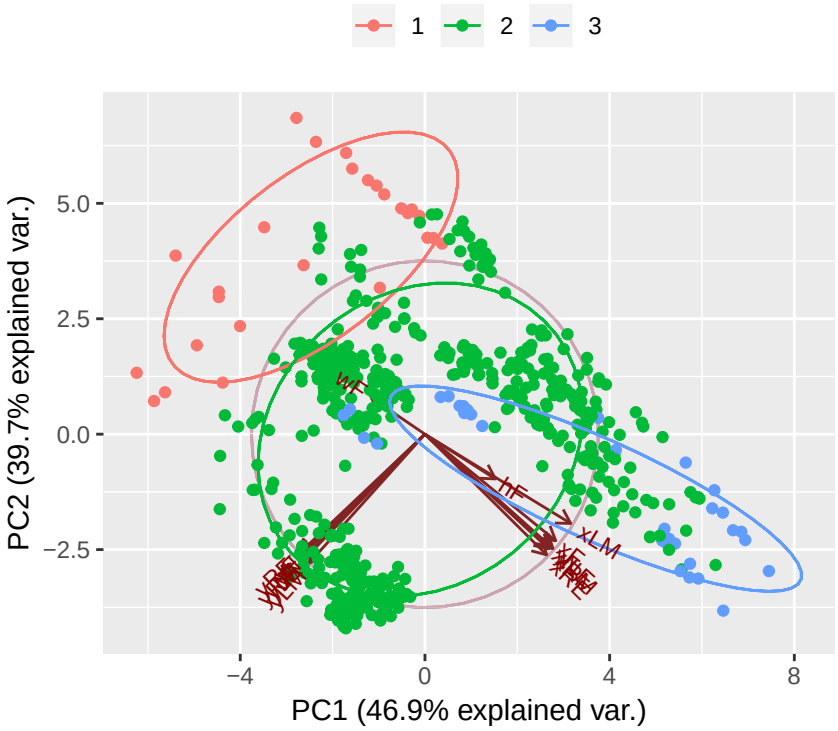
We can also analyze which variables are being loaded onto our components, which can be viewed using the rotation values from our PCA:

```
##             PC1         PC2          PC3
## xF    0.2942435 -0.26070636  0.078744995
## yF   -0.2794330 -0.29479659  0.046130612
## wF   -0.1206052  0.08913905 -0.766422460
## hF    0.1585780 -0.10946051 -0.628050184
## xRE   0.2738470 -0.29522750  0.022563377
## yRE  -0.2831645 -0.29171738  0.001006468
## xLE   0.2881683 -0.28016465 -0.034533147
## yLE  -0.2705769 -0.30511584  0.016749591
## xN    0.2832853 -0.27525736  0.020665047
## yN   -0.2768979 -0.29728337 -0.025268870
## xRM   0.2788090 -0.28061881 -0.002109732
## yRM  -0.2722651 -0.30239743 -0.037238484
## xLM   0.3280473 -0.21838995 -0.069718427
## yLM  -0.2604178 -0.31515751 -0.022807691
```

1.) For our first component, we can see the following loadings: It appears that all x coordinate variables are loaded in a positive direction, while the y coordinate variables are loaded in a negative direction.
2.) For our second component, we can see the following loadings: It appears that all x coordinate variables are loaded in a negative direction, while the y coordinate variables are also loaded in a negative direction.
3.) From our third component, we can see that we are primarily loading from the wF and hF variables, which determine the width and height of the image.

Bi-Plot

Next, we will develop a much more involved plot that shows the interaction of our three facial position responses and our components/predictors:



We can now see three clusters representing our response groups for position.

* Our "1" direction response cluster appears to have contributions from negative Component 1 values and positive Component 2 values.
* Our "2" direction response cluster appears to have contributions from generally low Component 1 and Component 2 values in both positive and negative directions.
* Our "3" position response appears to have contributions from positive Component values and negative component 2 values.

PCA Logistic Prediction Model

Now that we have our top three principal components, we can build another Multinomial Logistic Regression and compare our results to the original full model.

Here are the results from our Reduced Multinomial Logistic Regression:

```
## Call:
## multinom(formula = FaceResponse ~ ., data = pc3model)
##
## Coefficients:
##   (Intercept)      PC1       PC2       PC3
## 2    9.942927 1.443920 -1.999357 -0.223195
## 3    5.597659 2.060796 -2.313425 -1.272849
##
## Std. Errors:
##   (Intercept)       PC1       PC2       PC3
## 2    1.534336 0.2748871 0.3784010 0.2810547
## 3    1.583551 0.2890738 0.3963798 0.3748737
##
## Residual Deviance: 235.9345
## AIC: 251.9345
```

Once again, we have the log odds coeffecients for each of our predictors in the context of a response of 2 and 3. Now, we are able to compare our AIC and BIC values to see how effecient our reduced model is at explaining our response:

```
## AIC for Complete Model = 79.56918
## BIC for Complete Model = 211.7756
```

```
##
## AIC for 3 component Model = 251.9345
## BIC for 3 component Model = 287.1895
```

We can see that the AIC values in our model show that our original full model is doing much better at explaining the data, and our BIC is a bit larger for the reduced model which also implies the full model is working better. Given the large amount of predictors in the original model, we would expect AIC and BIC to penalize this model enough so that our Principal Component model would be working almost as effeciently as the complete model.
We are not seeing this strong similarity in explaining our response, so perhaps we have not added as many principal component to our reduced model as we should have in order to compete with our full model. Although the first three components accounted for 94% of the variance within our predictors, we will double our component size to six and compare those AIC and BIC values for a hopeful adjustment in information explained.

Here are the results when a top six component model is created:

```
## AIC for Complete Model = 79.56918
## BIC for Complete Model = 211.7756
```

```
##
## AIC for 3 component Model = 251.9345
## BIC for 3 component Model = 287.1895
```

```
##
## AIC for 6 component Model = 99.04275
## BIC for 6 component Model = 160.7391
```

With our six component model, we have successfuly reduced our AIC value to be almost as small as the full model and have an even lower BIC value.
This result provides evidence for us to suggest that the principal component model is quite effective in reducing the dimensions of our original prediction model, while still being able to retain a significant amount of variance explanation of our response variable. Although we originally proposed a 3 component model that ended up not being as effecient as expected, our 6 component model was effective and still reduced our dimensionality from 14 to 6 predictors.

B

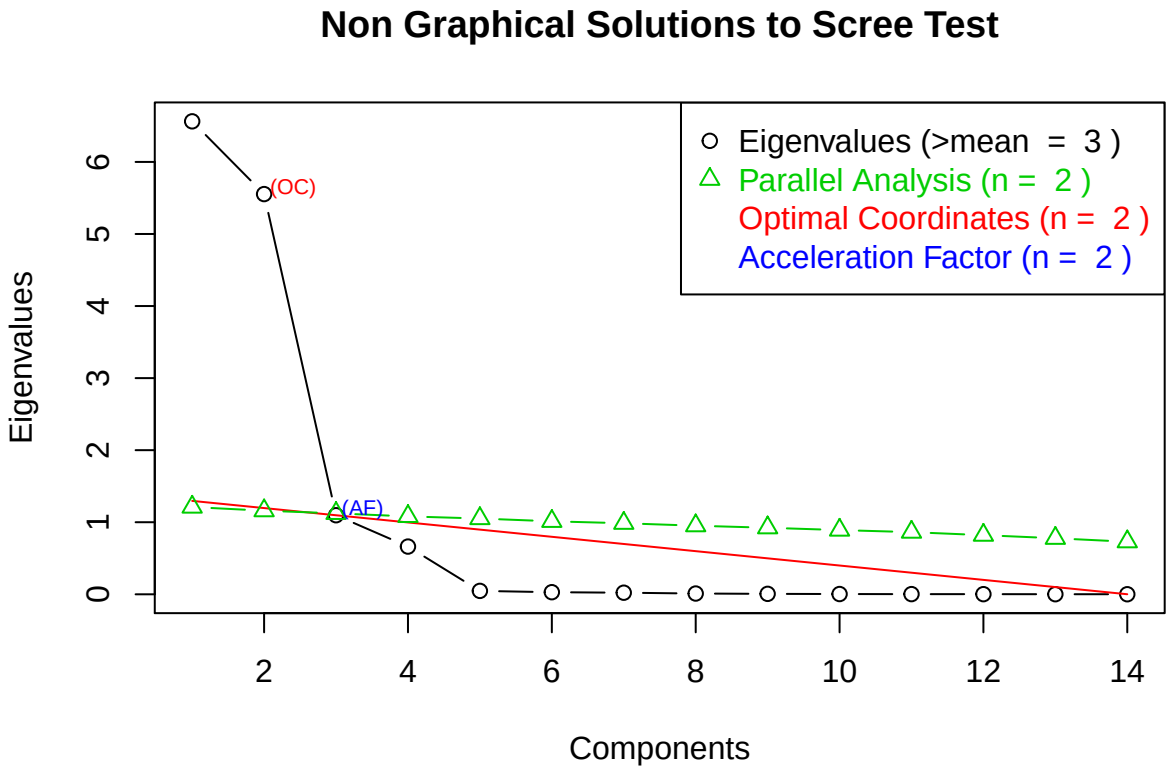
We will be performing a Factor Analysis for our provided variables in order to determine the underlying features accounting for our variables, as well as the amount of factors.

Using our Factor Analysis function, we were able to test various amounts of factors to decide which amount appropriatly contained all of our observed variables. Here is the output from a trial of 5 Factors:

```
##
## Loadings:
##     Factor1 Factor2 Factor3 Factor4 Factor5
## xF           0.900  -0.336   0.262
## yF   0.994
## wF          -0.257   0.849   0.105
## hF           0.344   0.116   0.840
## xRE          0.989
## yRE  0.996
## xLE          0.975           0.159
## yLE  0.997
## xN           0.996
## yN   0.996
## xRM          0.984                  -0.147
## yRM  0.997
## xLM -0.185   0.949           0.190
## yLM  0.998
##
##                Factor1 Factor2 Factor3 Factor4 Factor5
## SS loadings      5.999   5.795   0.874   0.866   0.031
## Proportion Var   0.429   0.414   0.062   0.062   0.002
## Cumulative Var   0.429   0.842   0.905   0.967   0.969
```

*We can see that our first factor loads many of the variables containing information regarding our y coordinates of various facial features.
*We can also see that our second factor loads the variables containing information regarding our x coordinates of various facial features.
*We can observe that our third factor loads the variable h, which pertains to the height of the image.
*We can observe that our fourth factor loads the variable w, which pertains to the width of the image.
*Finally, we can clearly observe that our fifth factor does not appear to load a great deal of information from any of our predictors.

Thus we can conclude that we have 4 underlying features/factors: X coordinate, Y coordinate, Face Height and Face Width. We can see that all x coordinate variables (6 in total) are loaded on to the X coordinate factor(Factor 1), and likewise for the Y coordinate variables(Factor 2). The face width(Factor 3) and face height(Factor 4) variables are also loaded onto their respective factors.



From our plot, we can see that our factor importance levels out after the fourth factor, so we can conclude that the four factors and their respective underlying features captures essentially all of our original predictors.

Differences Between Factor Analysis And PCA

Unlike PCA, Factor Analysis is unable to make nay predictions using our created factors. This is due to the fact that we do not have any information regarding to the actual factors, but rather only the variables loaded onto each factor. Thus, there are no variables to make from these factors and no coeffecients for any sort of prediction model are available. Factor Analysis is not exactly a tool used to create forecast/prediction models, but rather a method to analyze how our predictors can be grouped together.

Appendix

1

A

Before performing a statistical test for a difference in mean vectors for the 6 Apple Trees, we will review the mean vectors first.

```
Trees=read.table("Apple.txt",h=T)
Trees$Rootstock = as.factor(Trees$Rootstock)
```

```
RS1 = Trees[1:8,]
xbar1 = c(mean(RS1[,2]),mean(RS1[,3]),mean(RS1[,4]),mean(RS1[,5]))

RS2 = Trees[9:16,]
xbar2 = c(mean(RS2[,2]),mean(RS2[,3]),mean(RS2[,4]),mean(RS2[,5]))

RS3 = Trees[17:24,]
xbar3 = c(mean(RS3[,2]),mean(RS3[,3]),mean(RS3[,4]),mean(RS3[,5]))

RS4 = Trees[25:32,]
xbar4 = c(mean(RS4[,2]),mean(RS4[,3]),mean(RS4[,4]),mean(RS4[,5]))

RS5 = Trees[33:40,]
xbar5 = c(mean(RS5[,2]),mean(RS5[,3]),mean(RS5[,4]),mean(RS5[,5]))

RS6 = Trees[41:48,]
xbar6 = c(mean(RS6[,2]),mean(RS6[,3]),mean(RS6[,4]),mean(RS6[,5]))
```

Here is our MANOVA test result with Wilk's Lambda as our test of choice:

```
model = manova(cbind(y1,y2,y3,y4)~Rootstock,data=Trees)
summary(model,test = "Wilks")
```

B

Here is our MANOVA test result with Pillai's Trace as our test of choice:

```
summary(model,test = "Pillai")
```

C

Here, we are using the null hypothesis that the data is multivariate normal and any statistical significance would prove otherwise. Here is our result:

```
Trees.norm = Trees[,-1]
Trees.mat = matrix(c(0),48,4)
for(i in 1:48){
  for(j in 1:4){
    Trees.mat[i,j] = Trees.norm[i,j]
  }
}
Trees.mat = t(Trees.mat)
```

```r
mshapiro.test(Trees.mat)
```

```r
x = as.matrix(Trees[,-1])
center = colMeans(x)
n = nrow(x)
p <- ncol(x)
cov <- cov(x)
d = mahalanobis(x,center,cov)
```

```r
mqqnorm <- function(x,main="Multi-normal Q-Q Plot") {
  if (!is.matrix(x)) {x <- as.matrix(x)}
  distances <- mahalanobis(x,colMeans(x),cov(x))
  car::qqPlot(distances,distribution="chisq",df=mean(distances),lwd=1,grid=FALSE,
    main=main,xlab=expression(chi^2 * " quantiles"),
    ylab=expression("Mahalanobis distances "^2))
}
```

Homogeneity of Covariances

The numm hypothesis for this test is that we have homogeneity, while statistical significance would prove otherwise.

```r
boxM(Trees[,-1],Trees$Rootstock)
```

D

Wilk's Lambda Bootstrap

For the Wilk's Lambda bootstrapping process, we will need a function to compute the Λ value for each of our bootstrapped datasets. Here is that required function:

```r
Wlam = function(df){
    unique.groups = unique(df$Rootstock)
    g = length(unique.groups)
    ni = rep(0,g)
    Si = list()
    Sw = matrix(0,4,4)
    for(i in 1:g){
        ni[i] = length(df$y1[df$Rootstock==unique.groups[i]])
        Si[[i]] = var(df[df$Rootstock==unique.groups[i],-1])
        Sw = Sw+(Si[[i]]*(ni[i]-1))
    }
    n = sum(ni)
    S = var(df[,-1])
    St = S*(n-1)
    lambda = det(Sw)/det(St)
    lambda
}
```

We will verify our function by computing the observed Wilk's Lambds value of our original Tree data:

```r
lambda.obs = Wlam(Trees)
lambda.obs
```

Now, we will bootstrap our Lambda values 10,000 times and compare them to our observed value.

```
lambda.BS = rep(0,10000)
temp.data = Trees
for(i in 1:10000){
    temp.data[,-1] = Trees[sample(1:48,48,replace=T),-1]
    lambda.BS[i] = Wlam(temp.data)
}
```

Here is a histogram of our bootstrapped Lambdas, along with a line indicating our observed Lambda:

```
hist(lambda.BS,xlim=c(0,1))
lines(c(lambda.obs,lambda.obs),c(0,100000),lwd=2,col=2)
```

Pillai's Trace Bootstrap

For the Pillai's Trace bootstrapping process, we will need a function to compute the *trace* value for each of our bootstrapped datasets. The required function can be found in our Appendix.

```
Pillai = function(df){
  unique.groups = unique(df$Rootstock)
  g = length(unique.groups)
  ni = rep(0,g)
  Si = list()
  Sw = matrix(0,ncol(df)-1,ncol(df)-1)
  for(i in 1:g){
    ni[i] = length(df$y1[df$Rootstock==unique.groups[i]])
    Si[[i]] = var(df[df$Rootstock==unique.groups[i],-1])
    Sw = Sw+(Si[[i]]*(ni[i]-1))
  }
  n = sum(ni)
  S = var(df[,-1])
  St = S*(n-1)
  H = St -Sw
  lambda = sum(diag(H %*% solve(St) ))
  lambda
}
```

We will verify our function by computing the observed Wilk's Lambds value of our original Tree data:

```
pillai.obs = Pillai(Trees)
pillai.obs
```

Now, we will bootstrap our Trace values 10,000 times and compare them to our observed value.

```
pillai.BS = rep(0,10000)
temp.data = Trees
for(i in 1:10000){
  temp.data[,-1] = Trees[sample(1:48,48,replace=T),-1]
  pillai.BS[i] = Pillai(temp.data)
}
```

```
hist(pillai.BS,xlim=c(0,1.5))
lines(c(pillai.obs,pillai.obs),c(0,100000),lwd=2,col=2)
```

2

```
Face=read.csv("drivPoints.txt")
FaceResponse = as.factor(Face[,4])
FaceResp = as.numeric(Face[,4])
FaceAngle = as.numeric(Face[,5])
Face2 = Face[,6:19]
```

A

Original Model

Here is the summary of our full Multinomial Logistic Regression model

```
fullmodel = cbind(Face2,FaceResponse)
multi1 =  multinom(FaceResponse~.,data = fullmodel)
```

```
summary(multi1)
cat("BIC = ",BIC(multi1),"")
```

PCA

Now, we will be performing a full PCA analysis of our predictor variables to predict our response of either the gaze direction or the angle of the direction. First and foremost, we will take all of our 14 predictor variables and compute our PCA analysis:

```
pca = prcomp(Face2,center=T,scale=T)
```

Here is a general summary of our PCA:

```
summary(pca)
```

We can model this contribution trend in a linear plot:

```
plot(pca,type="l")
```

We can also analyze which variables are being loaded onto our components, which can be viewed using the rotation values from our PCA:

```
pca$rotation[,1:3]
```

Bi-Plot

Next, we will develop a much more involved plot that shows the interaction of our three facial position responses and our components/predictors:

```
source("ggbiplot.R")
```

```
g = ggbiplot(pca, obs.scale = 1, var.scale = 1, groups = FaceResponse, ellipse = TRUE,circle = TRUE)
g = g + scale_color_discrete(name = '')
g = g + theme(legend.direction = 'horizontal',legend.position = 'top')
g
```

PCA Logistic Prediction Model

Now that we have our top three principal components, we can build another Multinomial Logistic Regression and compare our results to the original full model.

```
pc3 = as.data.frame(pca$x[,1:3])
pc6 = as.data.frame(pca$x[,1:6])

pc3model = cbind(pc3,FaceResponse)
pc6model = cbind(pc6,FaceResponse)

multi3 = multinom(FaceResponse~.,data = pc3model)
multi6 = multinom(FaceResponse~.,data = pc6model)
```

Here are the results from our Reduced Multinomial Logistic Regression:

```
summary(multi3)
```

Once again, we have the log odds coeffecients for each of our predictors in the context of a response of 2 and 3. Now, we are able to compare our AIC and BIC values to see how effecient our reduced model is at explaining our response:

```
cat("AIC for Complete Model =",AIC(multi1),
    "\nBIC for Complete Model =",BIC(multi1),"\n")

cat("\nAIC for 3 component Model =",AIC(multi3),
    "\nBIC for 3 component Model =",BIC(multi3),"\n")
```

Here are the results when a top six component model is created:

```
cat("AIC for Complete Model =",AIC(multi1),
    "\nBIC for Complete Model =",BIC(multi1),"\n")

cat("\nAIC for 3 component Model =",AIC(multi3),
    "\nBIC for 3 component Model =",BIC(multi3),"\n")

cat("\nAIC for 6 component Model =",AIC(multi6),
    "\nBIC for 6 component Model =",BIC(multi6),"\n")
```

B

```
X = Face2
fact.model = factanal(X,factors=5,rotation="varimax")
fact.model$loadings
```

```
ev = eigen(cor(X))
ap = parallel(subject=nrow(X),var=ncol(X),rep=100,cent=.05)
nS = nScree(x=ev$values, aparallel=ap$eigen$qevpea)
plotnScree(nS)
```