Final Project

Keith Mitchell

3/7/2020

https://www.chegg.com/homework-help/Applied-Multivariate-Statistical-Analysis-6th-edition-chapter-11-problem-32E-solution-9780131877153 https://www.chegg.com/homework-help/Applied-Multivariate-Statistical-Analysis-6th-edition-chapter-8-problem-11E-solution-9780131877153

STA135 Final Project (UC Davis)

Instructor: Professor Li

TA: Cong Xu

Dataset 1: Conduct multiple linear regression;

Dataset 2: Conduct two-sample test and LDA;

Dataset 3: Conduct PCA.

For each data analysis, you should write in full sentences, and have thefollowing sections for the body of your report.

1) Introduction: Briefly summarize the goal of the analysis in your own words;
2) Summary: Summarize your data by plots or sample estimates;
3) Analysis: Implement the analysis based on what you have done in homework;
4) Conclusion: Describe and interpret your findings.

Details:

1) A title page including your name, the name of the class, and the name of your instructor.
2) Do not include R code in the body of your report. R code used to produce the results should all go to the appendix.(echo=FALSE)
3) Typed.
4) Double-sided pages.

DATASET 2: Two Sample Test and LDA

INTRODUCTION:

Data has been gather on hemophilia A carriers.

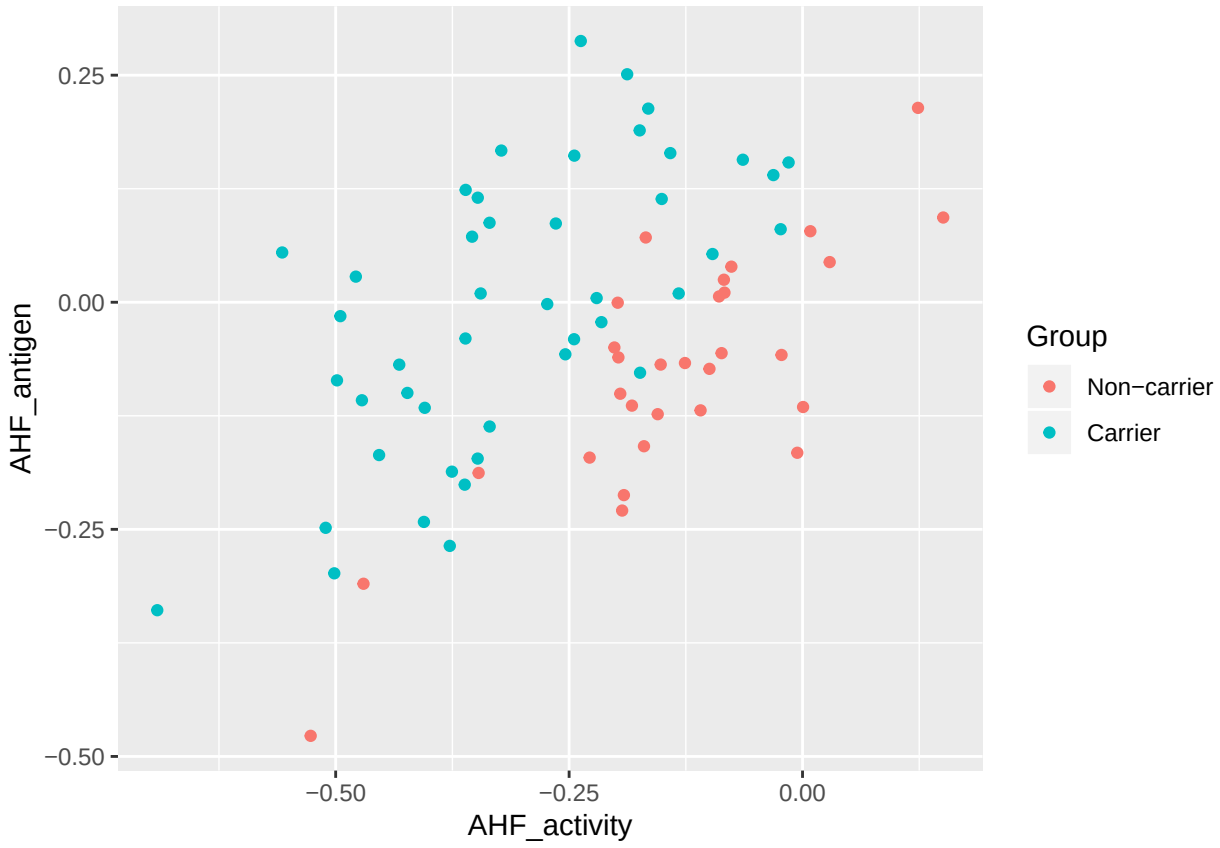
1) Predicting disease based on other characteristics is a common technique that doctors and healthcare workers use to produce a prediction for a given person.
2) This is a common method that if not used solely for diagnosing a disease is used in conjunction with other tests to increase the statistical power of the diagnoses. ### The first goal of this analysis will be to perform two sample test on the data in order to see if there is a significant difference between individuals that are carriers and noncarriers with respect to AHF activity (log_{10} scale) and AHF antigen (log_{10} scale). Group 1 is considered to be the non carrier group while group 2 is considered to be the obligatory carrier group. ### The second goal of this analysis will be to perform linear discriminant analysis to try and predict the best possible distinguishable boundary between carriers and non carriers with respect to AHF activity (log_{10} scale) and AHF antigen (log_{10} scale).

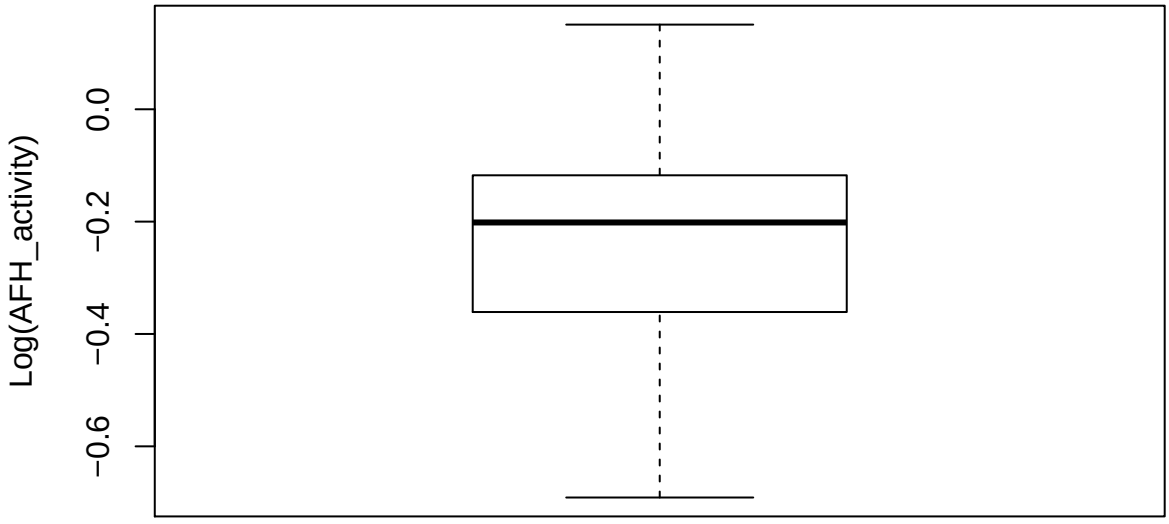
SUMMARIZE DATA:

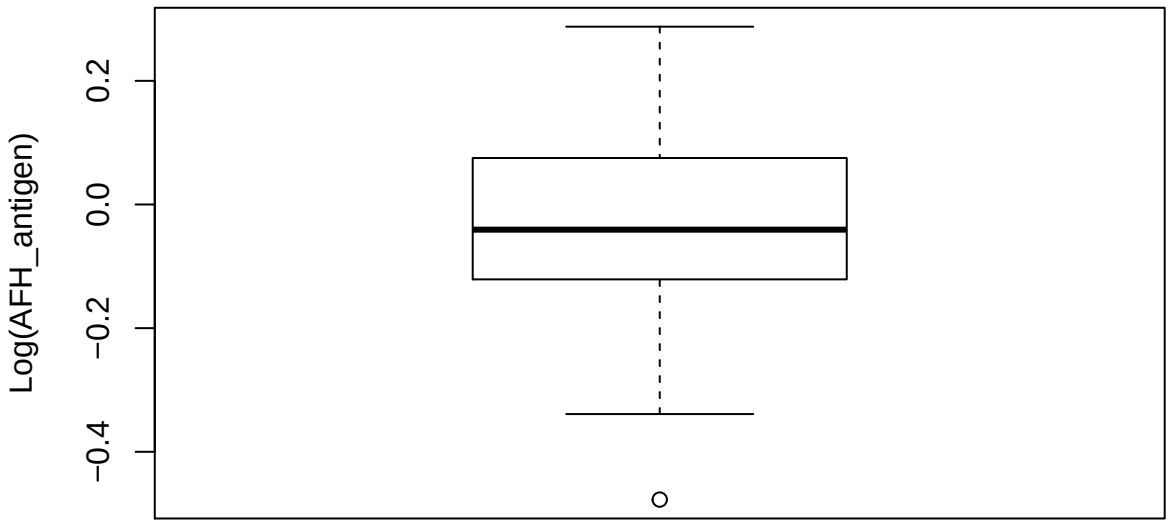
Lets look at the data as a whole (with the two groups together (carriers and non carriers))

```
## [1] 75  2
```

```
##          Group     AHF_activity      AHF_antigen
##  Non-carrier:30   Min.   :-0.6911   Min.   :-0.47730
##  Carrier    :45   1st Qu.:-0.3609   1st Qu.:-0.12110
##                   Median :-0.2015   Median :-0.04070
##                   Mean   :-0.2387   Mean   :-0.03474
##                   3rd Qu.:-0.1176   3rd Qu.: 0.07520
##                   Max.   : 0.1507   Max.   : 0.28760
```


0.25
0.00
−0.25
−0.50
AHF_antigen
−0.50
−0.25
0.00
AHF_activity
Group
Non−carrier
Carrier


Log(AFH_activity)
0.0
−0.2
−0.4
−0.6




```
## [1] "Summary and dimensions of the first group of individuals."
```

```
##          Group     AHF_activity       AHF_antigen
##  Non-carrier:30   Min.   :-0.52680   Min.   :-0.47730
##  Carrier    : 0   1st Qu.:-0.19470   1st Qu.:-0.14968
##                   Median :-0.13890   Median :-0.06775
##                   Mean   :-0.13487   Mean   :-0.07786
##                   3rd Qu.:-0.07805   3rd Qu.: 0.00955
##                   Max.   : 0.15070   Max.   : 0.21400
```

```
## [1] 30  3
```

```
## [1] "Summary and dimensions of the second group of individuals."
```

```
##          Group     AHF_activity      AHF_antigen
##  Non-carrier: 0   Min.   :-0.6911   Min.   :-0.339000
##  Carrier    :45   1st Qu.:-0.4055   1st Qu.:-0.107900
##                   Median :-0.3352   Median : 0.004600
##                   Mean   :-0.3079   Mean   :-0.005991
##                   3rd Qu.:-0.1878   3rd Qu.: 0.115100
##                   Max.   :-0.0149   Max.   : 0.287600
```

```
## [1] 45  3
```

DATA ANALYSIS:

Computing the LDA

```
group_1 <- group_1[2:3]
group_2 <- group_2[2:3]

x1 <- group_1
x2 <- group_2


# compute sample mean vectors:

x1.mean <- colMeans(x1)
x2.mean <- colMeans(x2)
x1.mean
```

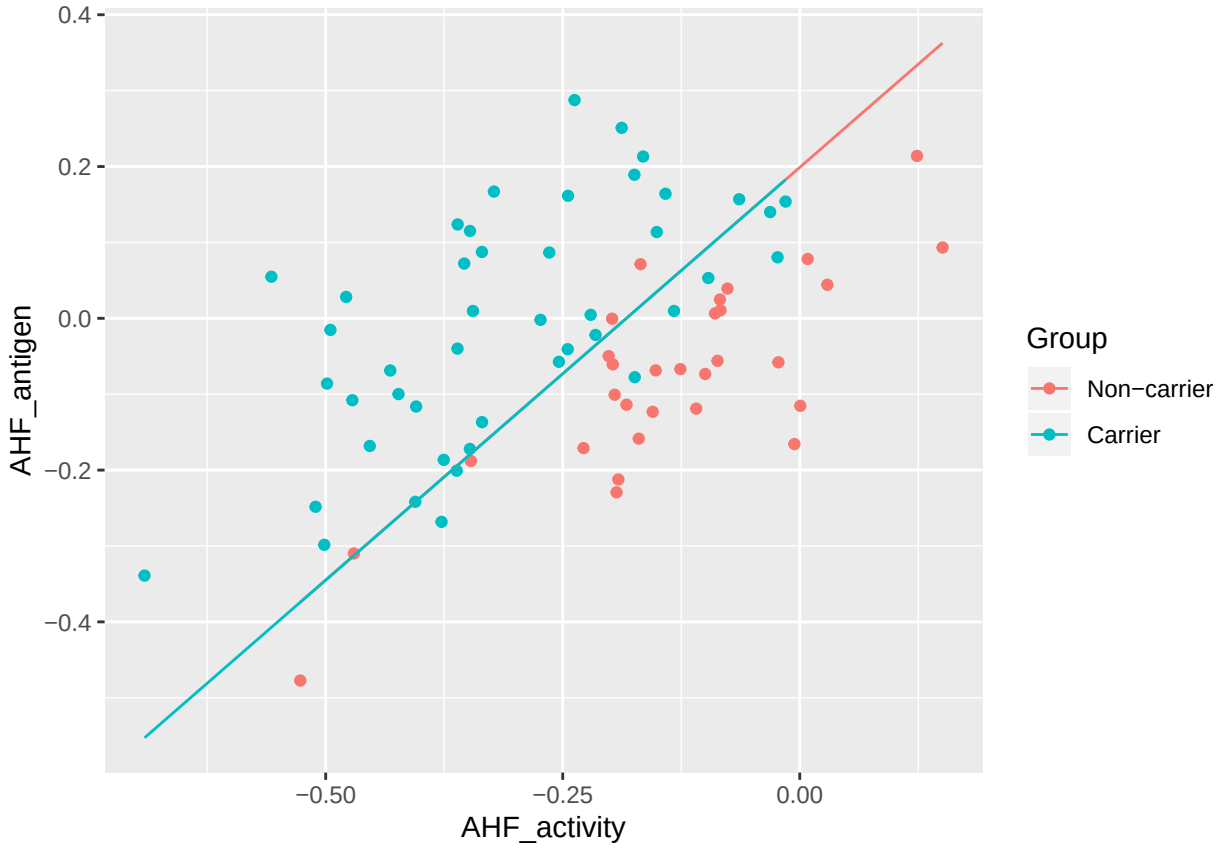
```
## AHF_activity  AHF_antigen
##  -0.13487000  -0.07785667
```

```
x2.mean
```

```
## AHF_activity  AHF_antigen
## -0.307946667 -0.005991111
```

```
# compute pooled estimate for the covariance matrix:
#TODO check this formula again
S_carrier <- var(x2)
S_noncarrier <- var(x1)
S.u <- (44*(var(x1))+29*(var(x2)))/73
w <- solve(S.u)%*%(x1.mean-x2.mean)
w0 <- -(x1.mean+x2.mean)%*%w/2
ggplot(data, aes(x=AHF_activity, y=AHF_antigen, color=Group)) + geom_point() + geom_line(aes(X[,1], -(w
```

```
## Warning in w[1] * X[, 1] + w0: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```



Partial code credit to Prof Li (UC Davis), Weiping Zhang(USTC)

Two sample test with

$H_0 : \overrightarrow{\mu}_{noncarrier} = \overrightarrow{\mu}_{carrier}$ at level of $\alpha = 0.05$ with the Hotelling's T^2 test.

```
# now we perform the two-sample Hotelling T^2-test
n<-c(45,30)
p<-2
```

```r
xmean1<-x1.mean
xmean2<-x2.mean
d<-xmean1-xmean2
d
```

```
## AHF_activity  AHF_antigen
##   0.17307667  -0.07186556
```

```r
Sp<-S.u
t2 <- t(d)%*%solve(sum(1/n)*Sp)%*%d
t2
```

```
##          [,1]
## [1,] 95.16565
```

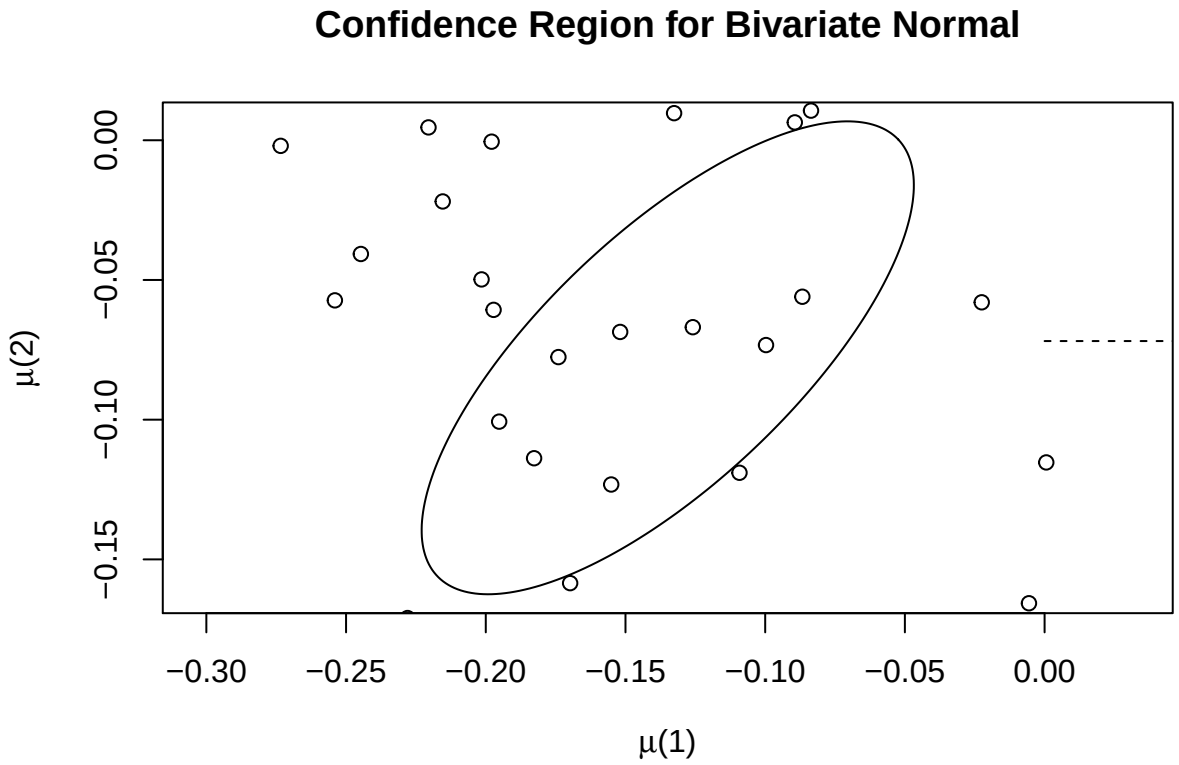
```r
alpha<-0.05
cval <- (sum(n)-2)*p/(sum(n)-p-1)*qf(1-alpha,p,sum(n)-p-1)
cval
```

```
## [1] 6.33459
```

Since $T^2 = 95.16 > 6.33$ the null hypothesis is rejected at 5% level of significance.

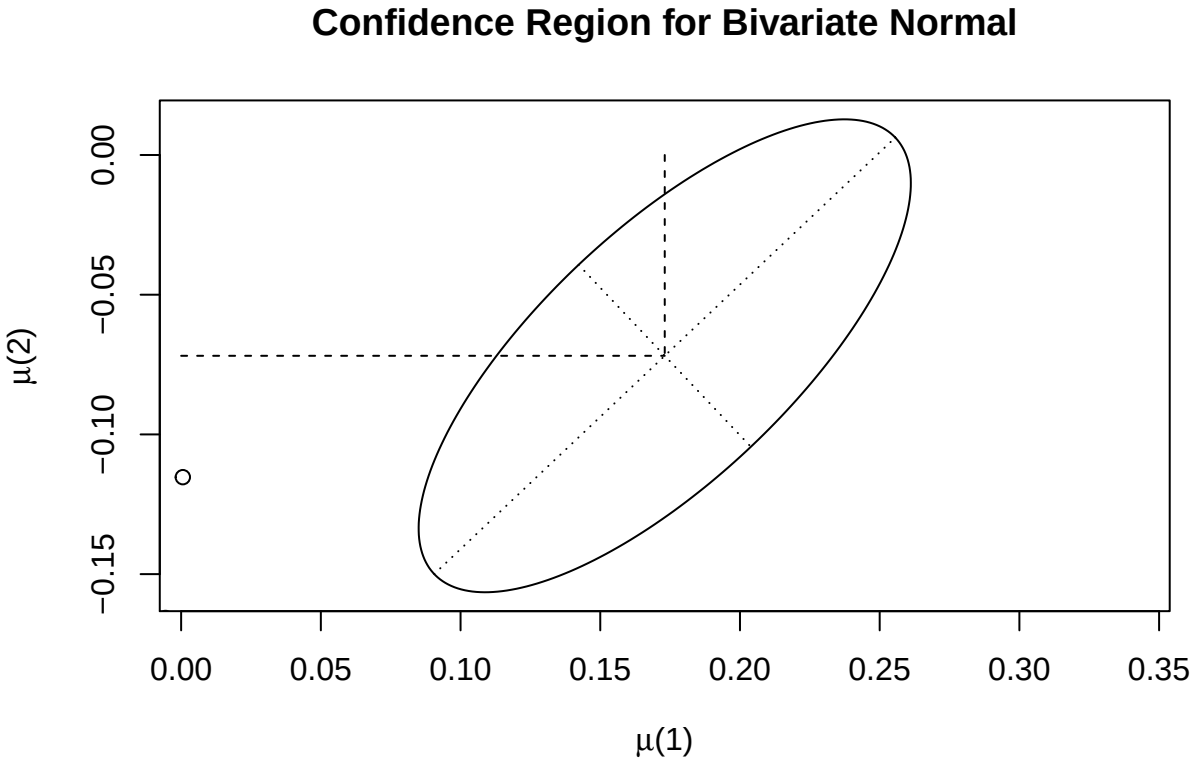
Confidence Region for Non Carriers

```r
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(xmean1-(e1%*%t(v1)))
plot(pts,type="l",main="Confidence Region for Bivariate Normal",xlab=expression(paste(mu, "(1)")), ylab=
segments(0,d[2],d[1],d[2],lty=2) # highlight the center
segments(d[1],0,d[1],d[2],lty=2)
#TODO check why these bars in the middle are weird and what are the values here corresponding to becaus
points(X)

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)
```




```
# since we reject the null, we use the simultaneous confidence intervals
# to check the significant components
```

```
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=expression(paste(mu, "(1)")), ylab=
segments(0,d[2],d[1],d[2],lty=2) # highlight the center
segments(d[1],0,d[1],d[2],lty=2)
#TODO check why these bars in the middle are weird and what are the values here corresponding to becaus
points(X)

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)
```




```
# 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)

# 95% simultaneous confidence interval
Cis
```

```
##                      [,1]       [,2]
## AHF_activity  0.08499858 0.26115476
## AHF_antigen  -0.15649036 0.01275925
```

```
#plot(Cis[1][1]:0, 1:10, type="l", lty=2)
```

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)
# 95% Bonferroni simultaneous confidence interval
Cis.b
```

```
##                      [,1]        [,2]
## AHF_activity  0.09298751 0.253165822
## AHF_antigen  -0.14881465 0.005083538
```

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

CONCLUSION

The company can use the following function to predict the amount of CPU time that they will need for a new set of parameters:

- y = 1.07898250130454z_1 + 0.419888473166963z_2 + 8.42368896741667
  - z_1 is the value of the companies number of orders (in thousands) that they need to know CPU time requirements for
  - z_2 is the companies add-delete items (in thousands) that they need to know CPU time requirements for
  - y is the predicted CPU time needed.

Then, as generated in the last portion of the data analysis, we can see that if we want to be XX% (where an alpha value of 0.05 corresponds to 95%, alpha of 0.001 corresponds to 99.9% etc.) sure that the true value of CPU time will fall within the lower and upper bound we will add and subtract the values generated to the value of y predicted above for some value of z_{01} being the orders in thousands and z_{02} being the add-delete items in thousands.

- Alpha value: 0.05 Interval: 3.91241721099891
- Alpha value: 0.01 Interval: 6.48784302704889
- Alpha value: 0.005 Interval: 7.88779247898333
- Alpha value: 0.001 Interval: 12.1331741930768

In addtion, these values can automatically be generated with the following script by just altering the line z_0 <- as.matrix(c(1,130,7.5)) by replacing "130 and 7.5 in the following manner. The general line would be z_0 <- as.matrix(c(1,z_{01},z_{02})) where the value of z_{01} is the orders in thousands and z_{02} is the add-delete items in thousands.

```
for (val in c(0.05,0.01,0.005,0.001))
{
  z_0 <- as.matrix(c(1,130,7.5))
  z_0_beta_hat <- t(z_0)%*%beta_hat
  statistic<-qt(1-(val/2),n-r-1)
  ans<-t(z_0)%*%(solve(t(Z)%*%Z))%*%z_0
  interval <- sqrt(sigma_sq)*statistic*sqrt(1+ans)
  int_low <- z_0_beta_hat - interval
  int_up <- z_0_beta_hat + interval
  print (sprintf("Alpha value: %s  Interval: %s  Lower bound: %s  Upper bound: %s",  val, interval, int
}
```

As a precaution we do suggest gather more data on this situation if possible. This will increase the strength of the analysis and account for other variables in the situation. This would be achieved by increasing the value of n, or observed situaitons (company CPU time requirements), and r, the number of observed varaibles upon which to make the prediction of CPU time (orders, add-delete items, etc.). Please contact if this is the case and a more robust script will be made to factor in variations in new potential data gathering.

DATASET 3: PCA

INTRODUCTION:

Data has been gatherd on populations which is considered "Census-tract data". The variables that have been gathered are "Total Population (thousands)", "Professional Degree (%)", "Employed age over 16 (%)", "Government Employment (%)", "Median home value ($100,000s)".

1) Census data can be very important for a variety of reasons, one of the most important/common ones is predicting voting outcomes.
2) Politicians may try to predict their popularity to certain populations by find the more common types of districts and try to gain popularity with one of those districts which would then hopefully have a similar effect on those other common areas. ### PCA (principle component analysis) is a popular way to explore datasets with multiple dimensions due to the fact that it is a diminsional reduction technique which allows exploration of the data in 2 dimensions (depending on how much of the variance can be summarized in those two dimensions). This is a great high level method to explore which groups have the most variation and potentially cluster together; therefore, this data can help those interested, such as politicians, in understanding their demographic.

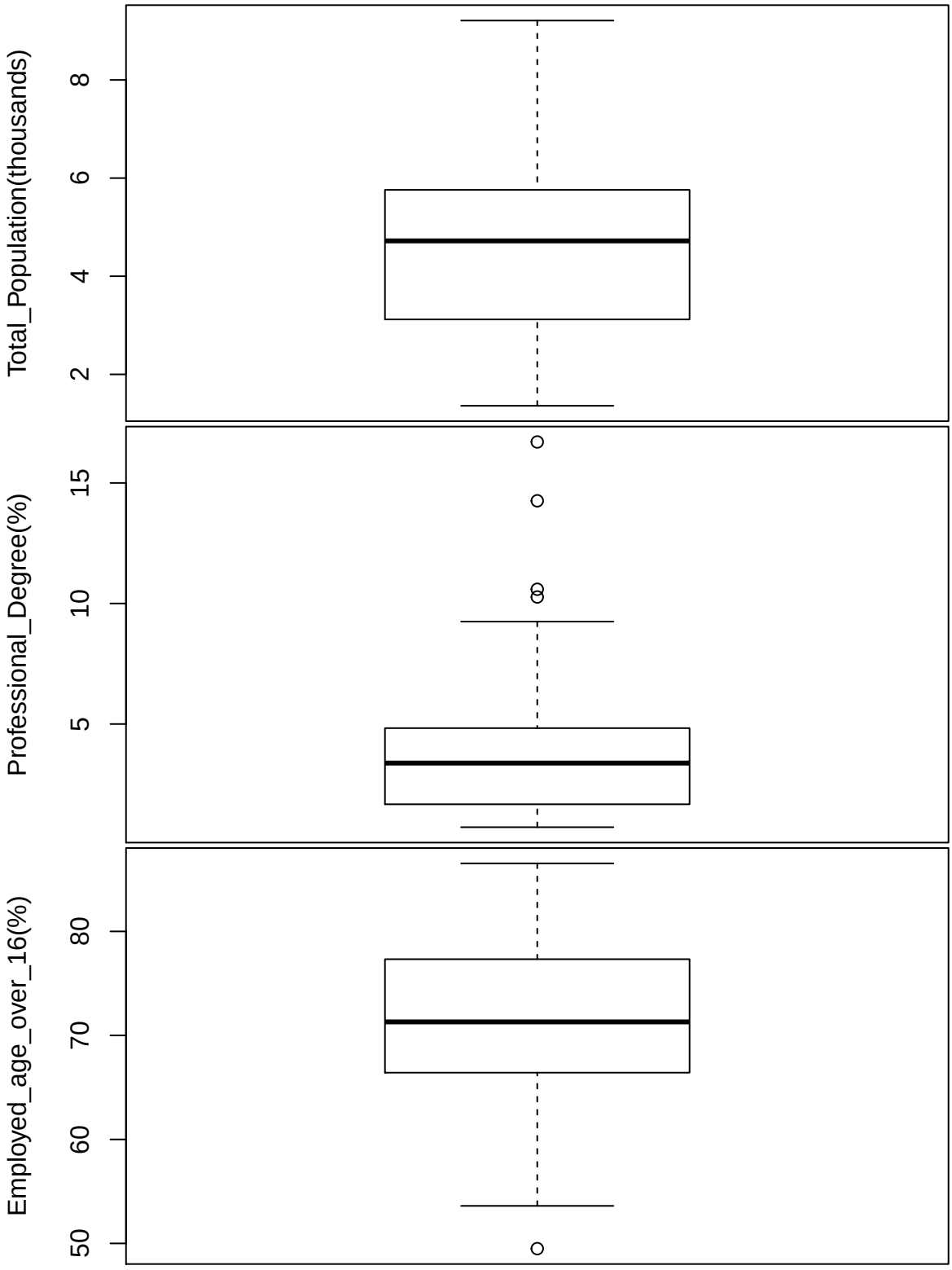
SUMMARIZE DATA:

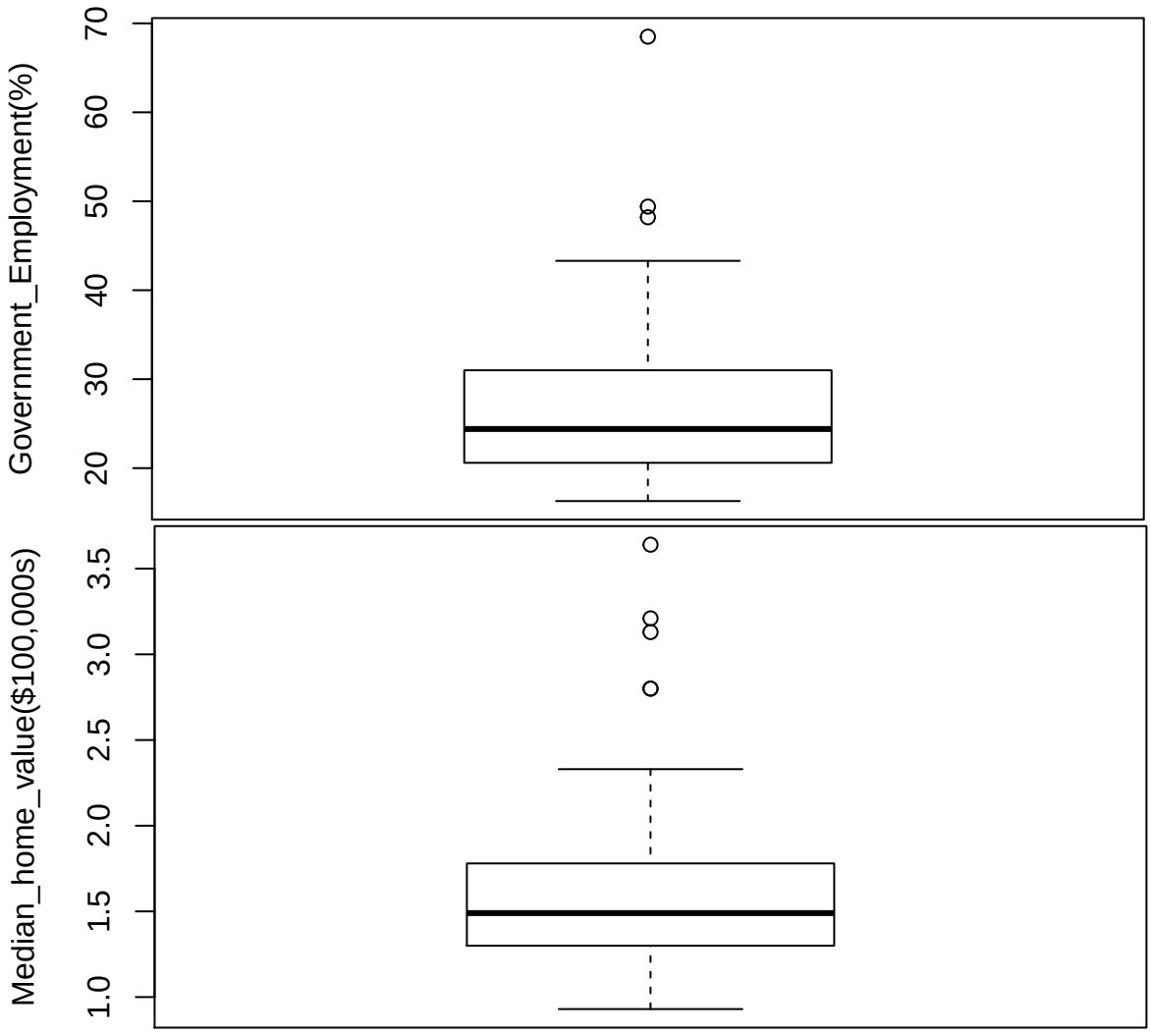
Lets summarize the dataframe

```
##  Total_Population(thousands) Professional_Degree(%) Employed_age_over_16(%)
##  Min.   :1.360               Min.   : 0.720         Min.   :49.50
##  1st Qu.:3.120               1st Qu.: 1.670         1st Qu.:66.42
##  Median :4.720               Median : 3.380         Median :71.30
##  Mean   :4.469               Mean   : 3.962         Mean   :71.42
##  3rd Qu.:5.760               3rd Qu.: 4.830         3rd Qu.:77.33
##  Max.   :9.210               Max.   :16.700         Max.   :86.54
##  Government_Employment(%) Median_home_value($100,000s)
##  Min.   :16.30            Min.   :0.930
##  1st Qu.:20.60            1st Qu.:1.300
##  Median :24.40            Median :1.490
##  Mean   :26.91            Mean   :1.636
##  3rd Qu.:31.00            3rd Qu.:1.780
##  Max.   :68.50            Max.   :3.640
```

```
## [1] 61  5
```

Lets also look at some box plots to help visualize the distribution of our varibles





DATA ANALYSIS:

First lets take a look at the eigen values and eigen vectors of the covariance matrix.

```
## [1] 107.0152535  39.6721358   8.3708660   2.8678740   0.1546931
```

```
##              [,1]        [,2]        [,3]        [,4]         [,5]
## [1,]  0.038887287 -0.07114494 -0.18789258  0.97713524 -0.057699864
## [2,] -0.105321969 -0.12975236  0.96099580  0.17135181 -0.138554092
## [3,]  0.492363944 -0.86438807 -0.04579737 -0.09104368  0.004966048
## [4,] -0.863069865 -0.48033178 -0.15318538 -0.02968577  0.006691800
## [5,] -0.009122262 -0.01474342  0.12498114  0.08170118  0.988637470
```

Also, lets take a look at the eigen values and eigen vectors of the correlation matrix which is also sometimes a good way to summarize our data (can be preferrable if the primary eigen values are a larger proportion than that of the covariance matrix).

```
## [1] 1.9919183 1.3675266 0.8641573 0.5350610 0.2413367
```

```
##            [,1]       [,2]        [,3]       [,4]       [,5]
## [1,]  0.2625829  0.4629936  0.78390268  0.2169291  0.2347882
## [2,] -0.5933541  0.3256442 -0.16407255 -0.1446471  0.7028828
```

```
## [3,]  0.3256978  0.6051419 -0.22487455 -0.6628689 -0.1943206
## [4,] -0.4792022 -0.2524850  0.55070086 -0.5716730 -0.2766497
## [5,] -0.4932213  0.4996473 -0.06882436  0.4072024 -0.5801162
```

Lets see if the covariance matrix or the correlation matrix summarizes the data in the first two components better.

```
## [1] "Variance summarized in first two components of the covariance matrix:"
```

```
## [1] 0.9279265
```

```
## [1] "Variance summarized in first two components of the corrlation matrix:"
```

```
## [1] 0.671889
```

COMPARE TO THE STANDARDIZED MATRIX

```
##              [,1]        [,2]         [,3]        [,4]        [,5]
##  [1,] -0.97609885  0.56194340 -0.321768114  0.35870114 -0.27561091
##  [2,] -1.20397978  0.13109026  0.209185541  1.73618299 -0.34647398
##  [3,] -0.73194072  2.02813021 -0.868811274  0.53883338  0.84048257
##  [4,]  0.36405801  1.11819410 -0.017408317 -0.25586769  0.37987256
##  [5,]  0.58108746  1.70016737  0.471980785  0.43287324  1.05307180
##  [6,]  0.30980065  0.28221039 -2.387928677  2.25538769 -0.06302167
##  [7,] -0.72108925  0.27577974 -0.592608110  1.13221018 -0.20474783
##  [8,] -1.10631652 -0.50232817 -0.565792269  1.04744206 -0.41733706
##  [9,]  0.49427568  0.10858300  1.556681561 -0.76447637  0.76961949
## [10,]  1.55771999 -0.39622254  0.158235443 -0.25586769 -0.38190552
## [11,]  0.25554328  0.22433459 -0.951940382  0.08320477 -0.38190552
## [12,]  0.19043445  0.09572172  1.504390671 -0.70090029 -0.31104244
## [13,]  0.29894918  0.06678382  1.720258192 -0.66911224 -0.38190552
## [14,] -0.59629731 -0.95247324 -0.199756037 -1.11414484 -0.82479976
## [15,] -0.45522817  0.78380061 -0.148805938  0.22095295  0.64560911
## [16,]  1.60655162  0.65518773 -0.119308513  1.14280619 -0.34647398
## [17,] -1.23110846  2.13102051 -0.210482373  1.56664676  0.66332488
## [18,]  1.46005674  0.24041120  1.075337212  0.64479352 -0.15160052
## [19,]  1.00972063 -0.34799271 -0.694508306  1.18519025  0.16728333
## [20,] -1.03035622 -0.67917087 -0.559088308  0.68717758 -0.80708399
## [21,]  1.03684931 -0.77241521 -1.128924934 -0.92341659 -1.24997823
## [22,]  0.47257274 -0.17758065  0.154213067 -0.72209232  0.04327295
## [23,]  0.21756313  0.39796198  0.497455834 -1.10354883  3.55099530
## [24,] -1.31792024  0.27899507 -0.488026329 -1.00818470 -0.25789514
## [25,]  0.54853304 -0.84315229  0.805838008 -0.56315210 -0.55906322
## [26,]  1.79102666  0.43654584 -1.722895816  0.45406527  2.78921721
## [27,]  0.98801768 -0.43802173 -0.951940382  0.05141672  0.25586218
## [28,]  1.03684931 -0.08112099  0.964051471  0.76134968 -0.59449476
## [29,]  0.70045366  0.03141528  1.655900173  0.47525729 -0.20474783
## [30,]  0.84694854 -0.27725563  0.619467912 -0.20288762 -0.98424168
## [31,]  0.33692933 -0.67917087  0.433097815 -0.29825175 -1.17911515
## [32,] -0.05914942 -0.73704667 -0.803112463 -0.34063580 -1.00195745
## [33,] -0.75906940 -0.63094104 -0.454506528 -0.07573545 -0.78936822
## [34,] -1.43728644 -0.91067405 -2.938994214 -0.53136406 -0.02759013
## [35,] -0.62885173 -0.97176517  0.446505736 -0.04394740 -0.91337861
## [36,] -0.55289142 -1.04250225 -0.728028108 -0.52076804 -0.77165245
## [37,] -1.48069233 -0.96211921 -1.498983542 -0.52076804 -0.82479976
## [38,] -1.44271217 -0.77884585 -0.183666532 -0.26646370 -1.12596784
```

```
## [39,]  0.60821614 -0.74026199  0.876899987 -1.03997274 -0.59449476
## [40,] -0.40639654 -0.73061603  1.472211661 -1.12474086 -0.20474783
## [41,] -0.58544584 -0.87530551 -0.569814645  0.08320477 -1.07282053
## [42,] -1.20397978 -0.37371529 -0.081766335 -0.37242385 -0.87794707
## [43,] -0.62885173 -0.85601358 -0.016067525 -0.81745644 -0.75393668
## [44,]  0.43459258 -0.88816680  0.222593462  0.35870114 -0.50591591
## [45,] -0.65598041 -0.94604260  0.394214846 -1.10354883 -0.71850514
## [46,]  1.24302729 -0.78206118  0.931872461 -0.45719196 -0.54134745
## [47,] -0.84045545  0.14716687 -1.732281360  4.40637858  1.08850334
## [48,] -1.53494969  4.09558222 -0.913057412  2.38253986  2.64749105
## [49,] -1.68687031  3.31104366 -0.670374049 -0.46778797  2.06287066
## [50,] -0.48235685 -0.18722662 -0.784341374 -0.08633146 -0.57677899
## [51,] -0.59087158 -0.57628057 -0.713279395 -0.45719196 -0.34647398
## [52,]  1.50888837 -0.90102809  0.951984342 -0.35123182 -0.24017937
## [53,]  0.52683010 -0.33191610  0.290973857 -0.48898000  0.02555718
## [54,]  0.73843381  0.16324348  0.792430087 -0.07573545  0.92906142
## [55,] -0.39554507 -0.54734267  1.110197805 -0.71149630 -0.09845321
## [56,]  2.57233268 -0.51518945  0.423712271 -0.54196007  0.14956756
## [57,] -1.26366288  0.75164739  2.027299574 -1.00818470  2.06287066
## [58,]  1.16706698  0.26613378  0.994889688 -0.73268833  1.23022950
## [59,] -0.12425826  0.59731194 -0.004000396  0.01962868  0.09642025
## [60,]  0.13617709  0.24041120  0.883603947 -0.66911224 -0.15160052
## [61,]  1.09110667  0.31114829  0.376784549 -0.63732420  0.61017757
## attr(,"scaled:center")
## [1]  4.469016  3.962295 71.419836 26.914754  1.635574
## attr(,"scaled:scale")
## [1] 1.8430678 3.1101085 7.4582781 9.4375109 0.5644688
```

```
## [1] 1.9919183 1.3675266 0.8641573 0.5350610 0.2413367
```

```
##            [,1]       [,2]        [,3]       [,4]       [,5]
## [1,]  0.2625829  0.4629936 -0.78390268  0.2169291  0.2347882
## [2,] -0.5933541  0.3256442  0.16407255 -0.1446471  0.7028828
## [3,]  0.3256978  0.6051419  0.22487455 -0.6628689 -0.1943206
## [4,] -0.4792022 -0.2524850 -0.55070086 -0.5716730 -0.2766497
## [5,] -0.4932213  0.4996473  0.06882436  0.4072024 -0.5801162
```

Also, lets take a look at the eigen values and eigen vectors of the correlation matrix which is also sometimes a good way to summarize our data (can be preferrable if the primary eigen values are a larger proportion than that of the covariance matrix).

```
## [1] 1.9919183 1.3675266 0.8641573 0.5350610 0.2413367
```

```
##            [,1]       [,2]        [,3]       [,4]       [,5]
## [1,]  0.2625829  0.4629936  0.78390268  0.2169291  0.2347882
## [2,] -0.5933541  0.3256442 -0.16407255 -0.1446471  0.7028828
## [3,]  0.3256978  0.6051419 -0.22487455 -0.6628689 -0.1943206
## [4,] -0.4792022 -0.2524850  0.55070086 -0.5716730 -0.2766497
## [5,] -0.4932213  0.4996473 -0.06882436  0.4072024 -0.5801162
```

Lets see if the covariance matrix or the correlation matrix summarizes the data in the first two components better.

```
## [1] "Variance summarized in first two components of the covariance matrix:"
```

```
## [1] 0.9279265
```

```
## [1] "Variance summarized in first two components of the corrlation matrix:"
## [1] 0.671889
```

So we see that using the covariance matrix is preferred here.

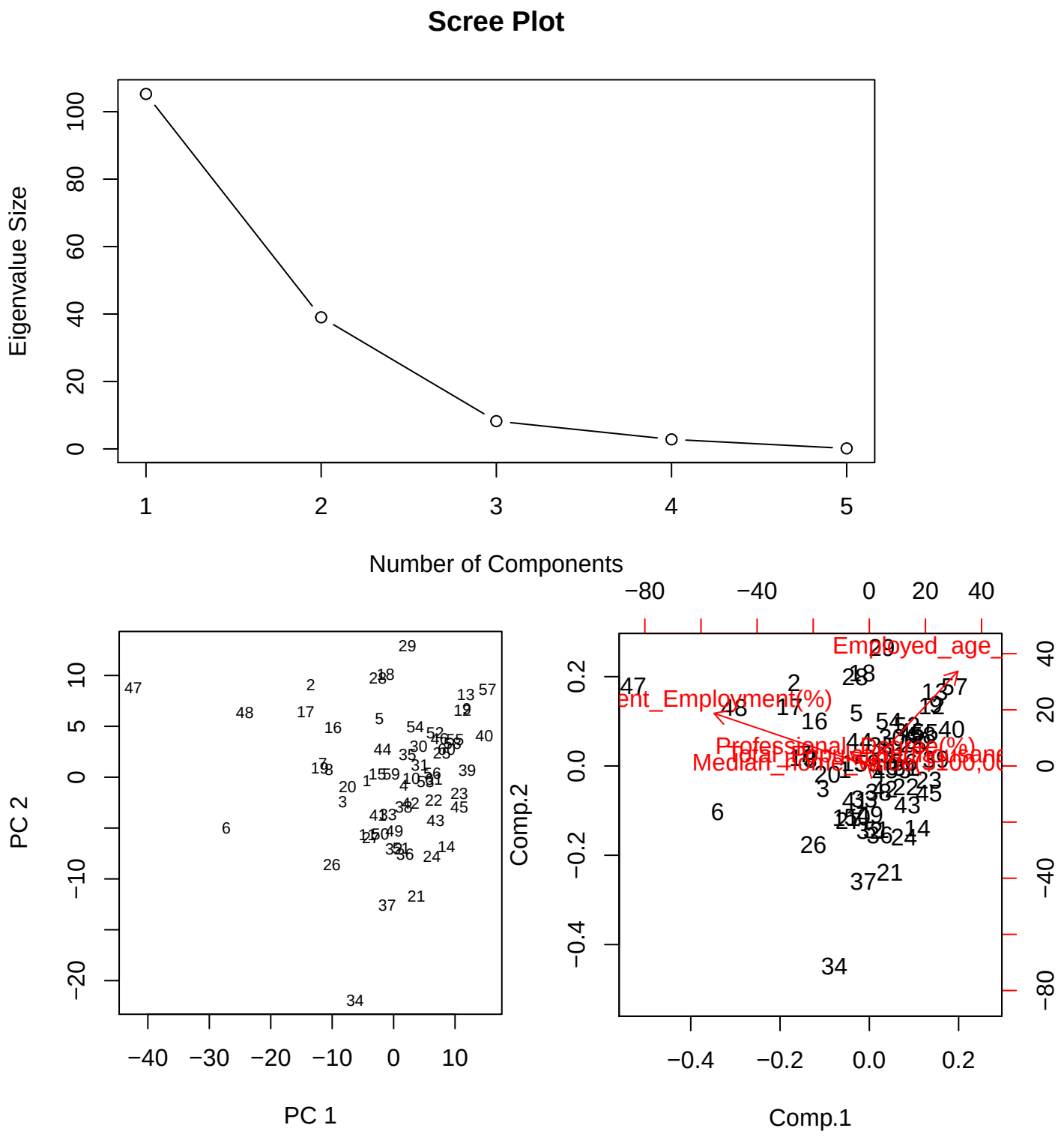
So since we can summarize 92% of the variance in our data using the first two principal components it would fair to graph our analysis using these two PCs. Lets also looks the the eigen value size graphed over the five components. Finally, we will overlay the PC scores for the sample data in the space of the first two principal components so we can visualize which of the sample data is most contributing to the the principal components which we see Government employment percentage and employed age over 16 (%) are the main two contributors to the

TODO, how do we do this with stuff we have learned in the course.

TODO, Do we need to do this for the standardized matrix instead?

```
## Importance of components:
##                             Comp.1    Comp.2     Comp.3     Comp.4       Comp.5
## Standard deviation     10.2596737 6.2467410 2.86943177 1.67954149 0.3900732431
## Proportion of Variance  0.6769654 0.2509611 0.05295308 0.01814182 0.0009785696
## Cumulative Proportion   0.6769654 0.9279265 0.98087961 0.99902143 1.0000000000
##
## Loadings:
##                               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## Total_Population(thousands)                  0.188  0.977
## Professional_Degree(%)        -0.105  0.130 -0.961  0.171  0.139
## Employed_age_over_16(%)        0.492  0.864
## Government_Employment(%)      -0.863  0.480  0.153
## Median_home_value($100,000s)                -0.125        -0.989
```

```
##      Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
## 105.2609051  39.0217730   8.2336387   2.8208596   0.1521571
```


Scree Plot
Eigenvalue Size
0
20
40
60
80
100
1
2
3
4
5
Number of Components
PC 2
PC 1
−20
−10
0
10
−40
−30
−20
−10
0
10
Comp.2
Comp.1
−0.4
−0.2
0.0
0.2
−80
−40
0
20
40
Employed_age_
ent_Employment(%)
Professional

CONCLUSION

As a precaution we do suggest gather more data on this situation if possible. This will increase the strength of the analysis and account for other variables in the situation. This would be achieved by increasing the value of n, or observed situaitons (census data in new areas), and r, the number of observed varaibles upon which to build PCs (such as "Total_Population(thousands)", "Professional_Degree(%)", and "Employed_age_over_16(%)" etc). Please contact if this is the case and a more robust script will be made to factor in variations in new potential data gathering.