00D7×

Booklet of Code and Output
for
STAD29/STA 1007 Final Exam

List of Figures in this document by page:

List of Figures

```r
library(MASS)
library(ggbiplot)
```

```
## Warning: package 'ggbiplot' was built under R version 3.5.1
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.5.3
## Loading required package: plyr
## Warning: package 'plyr' was built under R version 3.5.1
## Loading required package: scales
## Warning: package 'scales' was built under R version 3.5.1
## Loading required package: grid
```

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------
tidyverse 1.2.1 --
## v tibble  2.1.1         v purrr  0.3.2
## v tidyr  0.8.3.9000    v dplyr  0.8.0.1
## v readr  1.3.1         v stringr 1.4.0
## v tibble  2.1.1         v forcats 0.3.0
## Warning: package 'tibble' was built under R version 3.5.3
## Warning: package 'tidyr' was built under R version 3.5.3
## Warning: package 'readr' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.3
## Warning: package 'dplyr' was built under R version 3.5.2
## Warning: package 'stringr' was built under R version 3.5.2
## Warning: package 'forcats' was built under R version 3.5.1
## -- Conflicts ------------------------------------
tidyverse_conflicts() --
## x dplyr::arrange()    masks plyr::arrange()
## x readr::col_factor() masks scales::col_factor()
## x purrr::compact()    masks plyr::compact()
## x dplyr::count()      masks plyr::count()
## x purrr::discard()    masks scales::discard()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()        masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::select()    masks MASS::select()
## x dplyr::summarise()  masks plyr::summarise()
## x dplyr::summarize()  masks plyr::summarize()
```

```r
library(nnet)
library(car)
```

```
## Warning: package 'car' was built under R version 3.5.1
## Loading required package: carData
## Warning: package 'carData' was built under R version 3.5.1
##
## Attaching package: 'car'
## The following object is masked from 'package:dplyr':
##
##    recode
## The following object is masked from 'package:purrr':
##
```

```
football=read.csv("football.csv",header=T)
football %>% sample_n(20)
```

```
##    group wdim circum fbeye eyehd earhd  jaw
## 1      1 15.5  57.15  19.0  13.0  15.5 12.5
## 2      3 15.1  56.00  19.4  10.0  13.1 10.9
## 3      1 15.5  59.69  20.5  13.0  15.0 13.0
## 4      2 16.5  59.80  20.2   9.4  14.3 12.2
## 5      3 15.8  60.30  20.8  12.4  13.4 12.1
## 6      1 15.5  57.15  19.5  13.5  15.0 12.0
## 7      2 15.5  57.00  19.6  10.5  13.9 11.8
## 8      2 14.3  56.90  18.9  11.0  13.4 11.0
## 9      2 16.5  58.00  19.5   9.0  13.9 13.3
## 10     3 15.3  55.40  19.2   9.7  13.3 11.5
## 11     3 14.6  58.00  19.9  13.0  13.4 11.5
## 12     3 16.6  59.30  19.9  12.1  14.6 12.1
## 13     1 15.5  60.96  20.5  12.0  13.0 12.5
## 14     1 15.5  56.90  20.0  13.5  14.0 12.0
## 15     3 15.5  58.40  19.8  13.1  14.5 11.7
## 16     1 15.0  56.90  19.0  13.0  14.0 12.5
## 17     1 15.0  58.42  19.5  13.5  15.5 13.0
## 18     2 15.3  56.50  19.3   9.1  12.8 11.7
## 19     3 16.0  57.20  19.8  10.8  13.9 12.0
## 20     2 15.5  57.20  20.0  11.2  13.4 12.4
```

Figure 2: Football data (20 randomly-chosen rows)

```r
football.1=multinom(factor(group)~wdim+circum+fbeye+eyehd+earhd+jaw,
  data=football)
```

```
## # weights:  24 (14 variable)
## initial  value 98.875106
## iter  10 value 53.052168
## iter  20 value 51.037137
## iter  30 value 50.193419
## iter  40 value 50.102582
## iter  50 value 50.086496
## final  value 50.072216
## converged
```

```r
football.2=update(football.1,.~.-circum-fbeye)
```

```
## # weights:  18 (10 variable)
## initial  value 98.875106
## iter  10 value 54.475541
## iter  20 value 52.560238
## iter  30 value 51.745551
## iter  40 value 51.392312
## iter  50 value 51.217845
## iter  60 value 51.216798
## final  value 51.216069
## converged
```

```r
anova(football.2,football.1)
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: factor(group)
##                                         Model Resid. df Resid. Dev   Test
## 1                   wdim + eyehd + earhd + jaw       170   102.4321
## 2 wdim + circum + fbeye + eyehd + earhd + jaw       166   100.1444 1 vs 2
##      Df LR stat.   Pr(Chi)
## 1
## 2     4 2.287705 0.6830084
```

Figure 3: Football data modelling

```
wdims=c(15,16)
eyehds=c(10,13)
earhds=c(13.5,14.5)
jaws=c(11.5,12.5)
football.new=expand.grid(wdim=wdims,eyehd=eyehds,
  earhd=earhds,jaw=jaws)
pp=predict(football.2,football.new,type="probs")
cbind(football.new,pp)
```

```
##    wdim eyehd earhd  jaw            1           2           3
## 1    15    10  13.5 11.5 0.0046998061 0.606316601 0.388983593
## 2    16    10  13.5 11.5 0.0001243585 0.357557304 0.642318337
## 3    15    13  13.5 11.5 0.4903450615 0.093190874 0.416464065
## 4    16    13  13.5 11.5 0.0171707621 0.072729717 0.910099521
## 5    15    10  14.5 11.5 0.0201074693 0.619416127 0.360476404
## 6    16    10  14.5 11.5 0.0005536082 0.380082952 0.619363440
## 7    15    13  14.5 11.5 0.8134380174 0.036914893 0.149647089
## 8    16    13  14.5 11.5 0.0741175244 0.074963318 0.850919158
## 9    15    10  13.5 12.5 0.1211085566 0.722519265 0.156372179
## 10   16    10  13.5 12.5 0.0046611832 0.619757119 0.375581698
## 11   15    13  13.5 12.5 0.9784366914 0.008599227 0.012964082
## 12   16    13  13.5 12.5 0.4943800103 0.096836327 0.408783663
## 13   15    10  14.5 12.5 0.3697906836 0.526788323 0.103420993
## 14   16    10  14.5 12.5 0.0199194004 0.632422310 0.347658290
## 15   15    13  14.5 12.5 0.9950559839 0.002088237 0.002855779
## 16   16    13  14.5 12.5 0.8157446592 0.038153749 0.146101592
```

Figure 4: Football data predictions

```r
qq=read.csv("qq.csv",header=T)
qq
```

```
##    colour response_rate size
## 1    blue            28  300
## 2    blue            26  381
## 3    blue            31  226
## 4    blue            27  350
## 5    blue            35  100
## 6   green            34  153
## 7   green            29  334
## 8   green            25  473
## 9   green            31  264
## 10  green            29  325
## 11 orange            31  144
## 12 orange            25  359
## 13 orange            27  296
## 14 orange            29  243
## 15 orange            28  252
```

Figure 5: Questionnaire data

```r
qq.1=lm(response_rate~size*colour,data=qq)
qq.2=update(qq.1,.~.-size:colour)
anova(qq.2,qq.1)
```

```
## Analysis of Variance Table
##
## Model 1: response_rate ~ size + colour
## Model 2: response_rate ~ size * colour
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1     11 1.31619
## 2      9 0.76817  2   0.54801 3.2103 0.08864 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 6: Questionnaire analysis part 1

```
summary(qq.2)

##
## Call:
## lm(formula = response_rate ~ size + colour, data = qq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54735 -0.17479 -0.01275  0.18398  0.52896
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.4912250  0.3033120 123.606  < 2e-16 ***
## size         -0.0298129  0.0009613 -31.013 4.64e-12 ***
## colourgreen   1.3448159  0.2218649   6.061 8.17e-05 ***
## colourorange -1.7756427  0.2191075  -8.104 5.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3459 on 11 degrees of freedom
## Multiple R-squared:  0.9894,Adjusted R-squared:  0.9865
## F-statistic: 341.8 on 3 and 11 DF,  p-value: 3.9e-11
```

Figure 7: Questionnaire analysis part 2

```
qq.3=aov(response_rate~colour,data=qq)
summary(qq.3)

##             Df Sum Sq Mean Sq F value Pr(>F)
## colour       2    7.6     3.8   0.392  0.684
## Residuals   12  116.4     9.7
```

Figure 8: Questionnaire analysis part 3

```
ptsd=read.csv("ptsd.csv",header=T)
ptsd

##   patient trt pre post followup
## 1       1   A  21   15       15
## 2       2   A  24   15        8
## 3       3   A  21   17       22
## 4       4   A  26   20       15
## 5       5   B  32   17       16
## 6       6   B  27   20       17
## 7       7   B  21    8        8
## 8       8   B  25   19       15
## 9       9   B  18   10       13
```

Figure 9: PTSD data

```
response=with(ptsd,cbind(pre,post,followup))
ptsd.1=lm(response~trt,data=ptsd)
times=colnames(response)
times.df=data.frame(times)
ptsd.2=Manova(ptsd.1,idata=times.df,idesign=~times)
ptsd.2

##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##             Df test stat approx F num Df den Df   Pr(>F)
## (Intercept)  1   0.96881  217.400      1      7 1.58e-06 ***
## trt          1   0.00630    0.044      1      7 0.839098
## times        1   0.88973   24.206      2      6 0.001341 **
## trt:times    1   0.29361    1.247      2      6 0.352486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 10: PTSD repeated measures analysis

```
y.1=glm(y~x,data=mydata,family="binomial")
```

Figure 11: Code example

```
x=data.frame(id=1:2,t1=c(10,11),t2=c(12,14),t3=c(13,16))
x

##   id t1 t2 t3
## 1  1 10 12 13
## 2  2 11 14 16

x %>% gather(time,resp,t1:t3) -> y
```

Figure 12: Code example: “gather”

```
x

##   id t1 t2 t3
## 1  1 10 12 13
## 2  2 11 14 16

w=map_dbl(x,mean)
```

Figure 13: Code example: “map_dbl”

```
xx=1:4
xx

## [1] 1 2 3 4

yy=map_dbl(xx,sqrt)
```

Figure 14: Code example: “map_dbl” again

```r
f=function(mydata) {
  q1=quantile(mydata,0.25)
  q3=quantile(mydata,0.75)
  return(c(q1,q3))
}
data

##              x1          x2          x3
## 1   1.621867352 -1.65547514  1.16077867
## 2  -0.746347365 -1.20687430  0.47187652
## 3  -0.268930797  1.26874912  0.94460805
## 4  -0.699535090  0.83839323 -0.80725768
## 5   0.213237930 -0.74610634  0.27918883
## 6   0.708968535  0.05275361  0.68644436
## 7  -1.078329045  1.51487539  0.60764160
## 8   0.791310415 -0.11230871  0.07134409
## 9   0.004046959  0.26653521 -0.15448600
## 10  1.095879569 -1.72037830 -1.17761202

data %>% map(f) %>% bind_rows() -> res
```

Figure 15: Code example: “map”

```r
summit=read.csv("sumcr.csv",header=T)
head(summit)

##   Location Reach HU CumLen Length DepthWS WidthWS WidthBF HUAreaWS
## 1    HUA-1     A  R    9.2    9.2    0.12    4.10    9.00    37.72
## 2    HUA-2     A  G   29.7   20.5    0.21    3.98    9.63    81.66
## 3    HUA-3     A  R   51.2   21.5    0.10    4.46   11.43    95.83
## 4    HUA-4     A  R   61.2   10.0    0.10    3.57    8.80    35.67
## 5    HUA-5     A  G   72.4   11.2    0.19    2.90    5.27    32.48
## 6    HUA-6     A  P   91.4   19.0    0.10    5.04    8.72    95.76
##   HUAreaBF    wsgrad
## 1    82.80  0.008696
## 2   197.48  0.002927
## 3   245.75  0.001395
## 4    88.00  0.071000
## 5    58.99 -0.000893
## 6   165.68  0.006316
```

Figure 16: Summit Creek data (some)

```r
response=with(summit,cbind(DepthWS,WidthWS,WidthBF,
  HUAreaWS,HUAreaBF,wsgrad))
summit.1=manova(response~Reach,data=summit)
summary(summit.1)
```

```
##           Df Pillai approx F num Df den Df    Pr(>F)
## Reach      2 1.0472   14.838     12    162 < 2.2e-16 ***
## Residuals 85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 17: Summit Creek MANOVA

```r
summit.s=data.frame(scale(summit[,6:11]),Reach=summit$Reach)
head(summit.s)
```

```
##      DepthWS    WidthWS     WidthBF   HUAreaWS
## 1 -1.1140023  1.0464475  0.02624171 0.28303272
## 2 -0.1193125  0.9129552  0.25827368 2.41575861
## 3 -1.3350444  1.4469241  0.92122217 3.10353115
## 4 -1.3350444  0.4568569 -0.04741923 0.18353140
## 5 -0.3403547 -0.2884746 -1.34753489 0.02869764
## 6 -1.3350444  2.0921364 -0.07688361 3.10013355
##      HUAreaBF       wsgrad Reach
## 1 -0.11705133 -0.004015163     A
## 2  1.90642201 -0.320731457     A
## 3  2.75812290 -0.404837778     A
## 4 -0.02529983  3.416454930     A
## 5 -0.53716730 -0.530448261     A
## 6  1.34532634 -0.134676418     A
```

Figure 18: Standardizing the variables

```r
summit.2=lda(Reach~DepthWS+WidthWS+WidthBF+
  HUAreaWS+HUAreaBF+wsgrad,data=summit.s)
summit.2

## Call:
## lda(Reach ~ DepthWS + WidthWS + WidthBF + HUAreaWS + HUAreaBF +
##     wsgrad, data = summit.s)
##
## Prior probabilities of groups:
##         A         B         C
## 0.2272727 0.5227273 0.2500000
##
## Group means:
##       DepthWS    WidthWS    WidthBF   HUAreaWS
## A -0.57797502  0.9140677 -0.1372856  0.6621813
## B  0.29153758 -0.6836309 -0.5222121 -0.4504809
## C -0.08414673  0.5984395  1.2167030  0.3399316
##     HUAreaBF      wsgrad
## A  0.1780621  0.08637168
## B -0.3921102 -0.02679376
## C  0.6579922 -0.02249639
##
## Coefficients of linear discriminants:
##                  LD1        LD2
## DepthWS  -0.14441198 -0.5561652
## WidthWS   0.92700211  0.4230815
## WidthBF   0.65181363 -0.9518740
## HUAreaWS  0.16814959  1.2646110
## HUAreaBF  0.05114102 -1.0227454
## wsgrad    0.06106117 -0.0350707
##
## Proportion of trace:
##    LD1    LD2
## 0.6295 0.3705
```

Figure 19: Summit Creek discriminant analysis

```
library(cluster)

## Warning: package 'cluster' was built under R version 3.5.2

flower0=flower
names(flower0)=c("winters","shadow","tubers","colour","soil",
  "preference","height","distance")
flower0

##    winters shadow tubers colour soil preference height distance
## 1        0      1      1      4    3         15     25       15
## 2        1      0      0      2    1          3    150       50
## 3        0      1      0      3    3          1    150       50
## 4        0      0      1      4    2         16    125       50
## 5        0      1      0      5    2          2     20       15
## 6        0      1      0      4    3         12     50       40
## 7        0      0      0      4    3         13     40       20
## 8        0      0      1      2    2          7    100       15
## 9        1      1      0      3    1          4     25       15
## 10       1      1      0      5    2         14    100       60
## 11       1      1      1      5    3          8     45       10
## 12       1      1      1      1    2          9     90       25
## 13       1      1      0      1    2          6     20       10
## 14       1      1      1      4    2         11     80       30
## 15       1      0      0      3    2         10     40       20
## 16       1      0      0      4    2         18    200       60
## 17       1      0      0      2    2         17    150       60
## 18       0      0      1      2    1          5     25       10
```

Figure 20: Flower data. Data set `flower` comes with the package `cluster` and does not need to be read in separately. `flower0` is a copy of `flower`, which I modify by adding names to it.

```
flower.1=hclust(flower.d,method="ward.D")
```

Figure 21: Flower data, Ward cluster analysis

```
flower.2=cmdscale(flower.d,eig=T)
flower.2$GOF

## [1] 0.4467244 0.5394018
```

Figure 22: Flower data, multidimensional scaling

```
data.frame(flower.2$points) %>%
  mutate(id=row_number()) %>%
  ggplot(aes(x=X1,y=X2,label=id))+
    geom_point()+geom_text_repel()
```

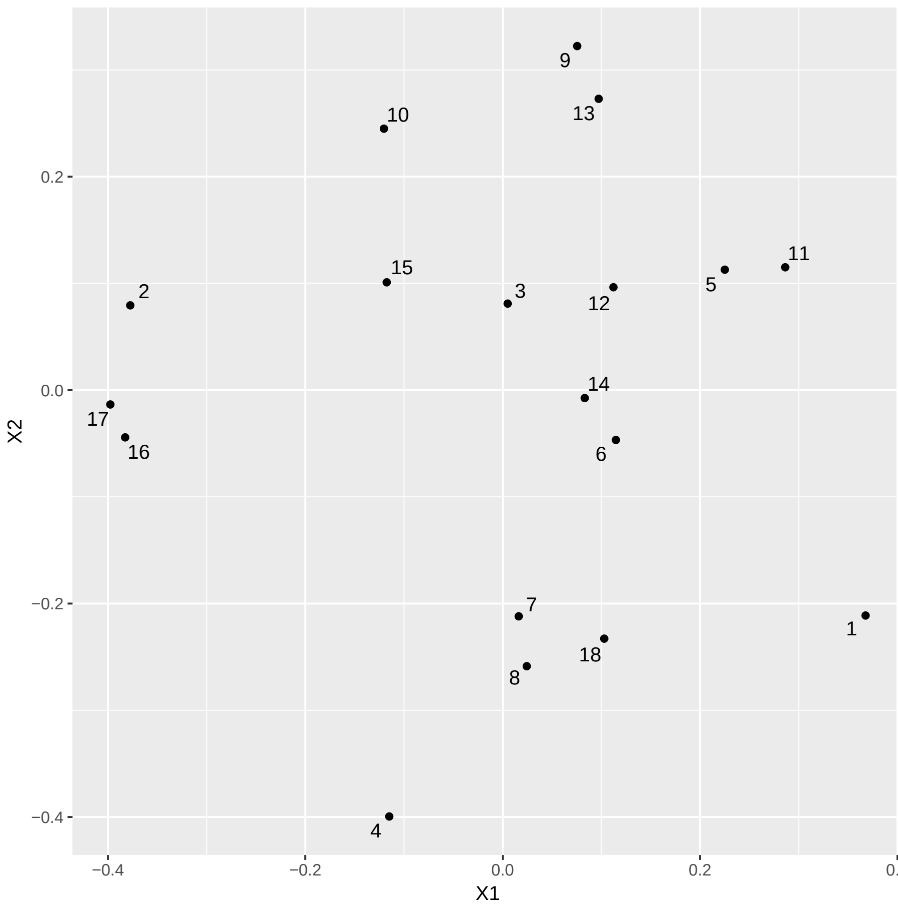

Figure 23: Flower data, multidimensional scaling map

```
la=read.table("la.txt",header=T, stringsAsFactors=F)
la

##    district population school employment services housevalue
## 1         A       5700   12.8       2500      270      25000
## 2         B       1000   10.9        600       10      10000
## 3         C       3400    8.8       1000       10       9000
## 4         D       3800   13.6       1700      140      25000
## 5         E       4000   12.8       1600      140      25000
## 6         F       8200    8.3       2600       60      12000
## 7         G       1200   11.4        400       10      16000
## 8         H       9100   11.5       3300       60      14000
## 9         J       9900   12.5       3400      180      18000
## 10        K       9600   13.7       3600      390      25000
## 11        L       9600    9.6       3300       80      12000
## 12        M       9400   11.4       4000      100      13000
```

Figure 24: LA census district data

```
la.1=princomp(la[,-1],cor=T)
summary(la.1)

## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4      Comp.5
## Standard deviation     1.6950851 1.3403955 0.46350500 0.31612348 0.123512644
## Proportion of Variance 0.5746627 0.3593320 0.04296738 0.01998681 0.003051075
## Cumulative Proportion  0.5746627 0.9339947 0.97696211 0.99694893 1.000000000
```

Figure 25: LA data principal components analysis

```r
ggscreeplot(la.1)
```

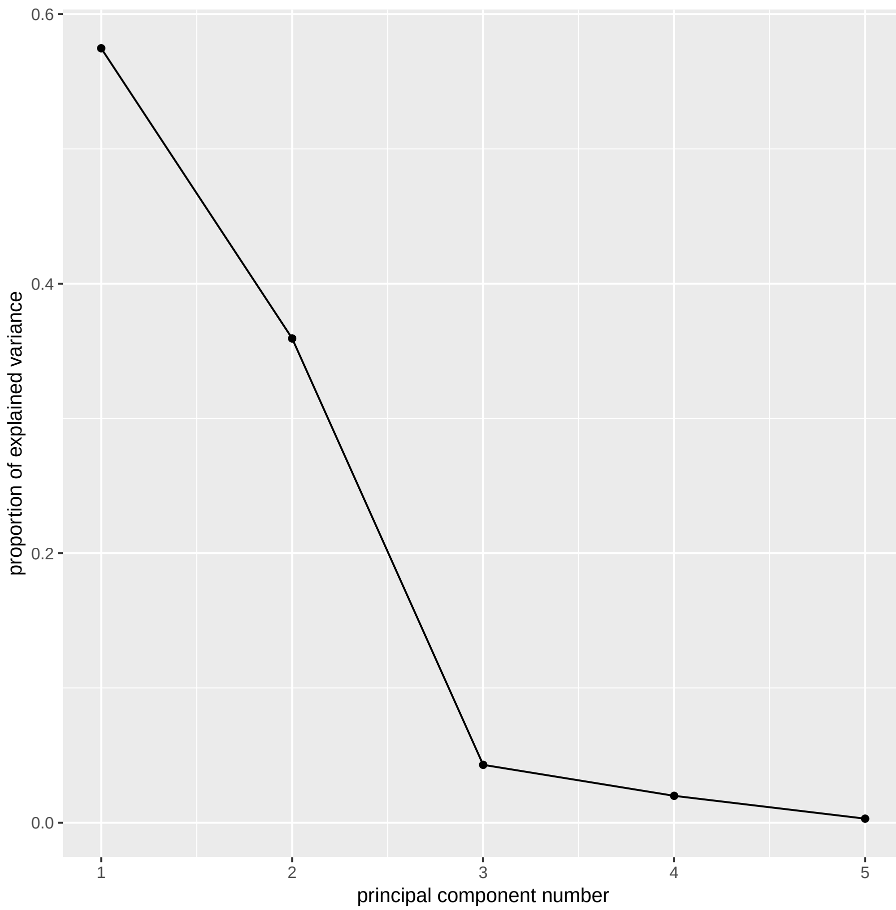

Figure 26: LA data scree plot

```
la.1$loadings

##
## Loadings:
##            Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## population  0.343  0.602         0.204  0.689
## school      0.453 -0.406  0.689 -0.354  0.175
## employment  0.397  0.542  0.248        -0.698
## services    0.550        -0.664 -0.500
## housevalue  0.467 -0.416 -0.140  0.763
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## SS loadings       1.0    1.0    1.0    1.0    1.0
## Proportion Var    0.2    0.2    0.2    0.2    0.2
## Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

Figure 27: LA data component loadings

```r
data.frame(la.1$scores) %>%
  ggplot(aes(x=Comp.1,y=Comp.2,label=la[,1]))+
    geom_point()+geom_text_repel()
```

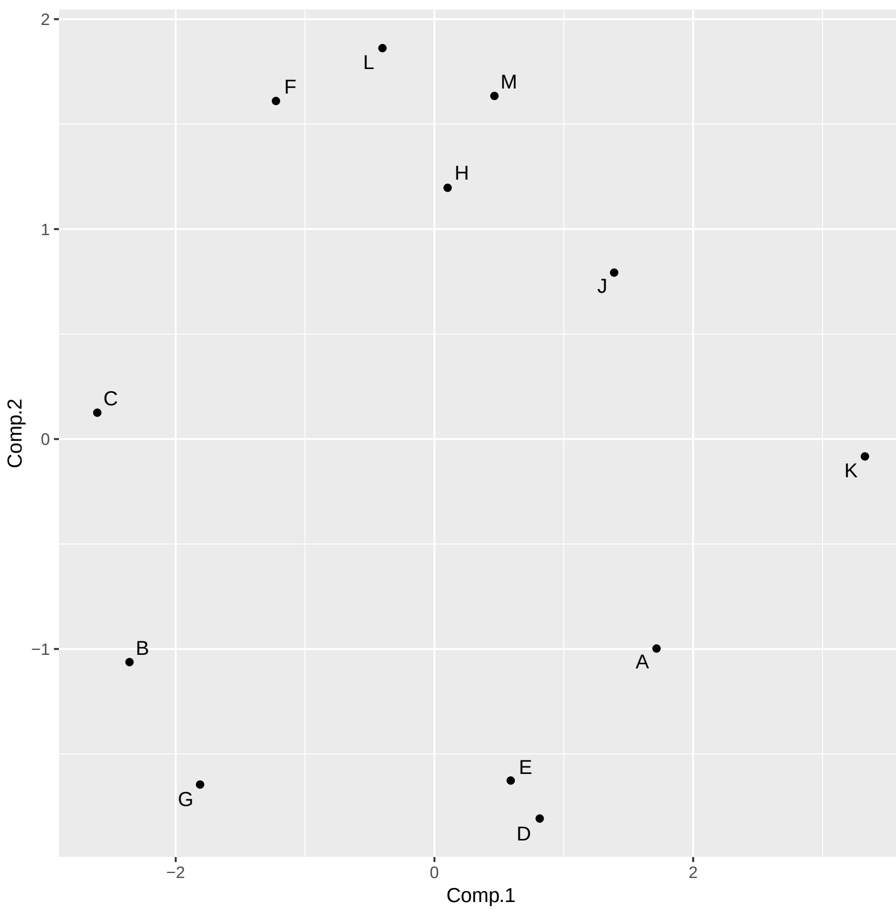

Figure 28: LA data principal component scores

```
classroom=read.csv("classroom.csv",header=T)
classroom

##      behaviour        risk adversity freq
## 1  non_deviant not_at_risk       low   16
## 2      deviant not_at_risk       low    1
## 3  non_deviant     at_risk       low    7
## 4      deviant     at_risk       low    1
## 5  non_deviant not_at_risk    medium   15
## 6      deviant not_at_risk    medium    3
## 7  non_deviant     at_risk    medium   34
## 8      deviant     at_risk    medium    8
## 9  non_deviant not_at_risk      high    5
## 10     deviant not_at_risk      high    1
## 11 non_deviant     at_risk      high    3
## 12     deviant     at_risk      high    3
```

Figure 29: Classroom behaviour data

```
tab.1=xtabs(freq~behaviour+risk+adversity,data=classroom)
ftable(tab.1)

##                          adversity high low medium
## behaviour   risk
## deviant     at_risk                   3   1      8
##             not_at_risk               1   1      3
## non_deviant at_risk                   3   7     34
##             not_at_risk               5  16     15
```

Figure 30: Classroom behaviour contingency table

```
classroom.1=glm(freq~behaviour*risk*adversity,family="poisson",
  data=classroom)
anova(classroom.1,test="Chisq")

## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: freq
##
## Terms added sequentially (first to last)
##
##
##                          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                       11    100.718
## behaviour                 1   44.430        10     56.288 2.636e-
11 ***
## risk                      1    2.329         9     53.959  0.126990
## adversity                 2   37.540         7     16.419 7.053e-
09 ***
## behaviour:risk            1    1.442         6     14.977  0.229767
## behaviour:adversity       2    3.656         4     11.321  0.160707
## risk:adversity            2   10.378         2      0.943  0.005578 **
## behaviour:risk:adversity  2    0.943         0      0.000  0.624114
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 31: Classroom behaviour analysis step 1

```
classroom.2=update(classroom.1,.~.-behaviour:risk:adversity)
drop1(classroom.2,test="Chisq")

## Single term deletions
##
## Model:
## freq ~ behaviour + risk + adversity + behaviour:risk + behaviour:adversity +
##     risk:adversity
##                     Df Deviance    AIC     LRT Pr(>Chi)
## <none>                   0.9428 61.692
## behaviour:risk       1   1.9040 60.653  0.9611 0.326906
## behaviour:adversity  2   4.1180 60.867  3.1752 0.204420
## risk:adversity       2  11.3206 68.069 10.3777 0.005578 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

classroom.3=update(classroom.2,.~.-behaviour:risk)
drop1(classroom.3,test="Chisq")

## Single term deletions
##
## Model:
## freq ~ behaviour + risk + adversity + behaviour:adversity + risk:adversity
##                     Df Deviance    AIC     LRT Pr(>Chi)
## <none>                   1.9040 60.653
## behaviour:adversity  2   5.5603 60.309  3.6563 0.160707
## risk:adversity       2  12.7629 67.512 10.8589 0.004385 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 32: Classroom behaviour analysis step 2

```
classroom.4=update(classroom.3,.~.-behaviour:adversity)
drop1(classroom.4,test="Chisq")

## Single term deletions
##
## Model:
## freq ~ behaviour + risk + adversity + risk:adversity
##                Df Deviance     AIC    LRT  Pr(>Chi)
## <none>               5.560  60.309
## behaviour       1   49.990 102.739 44.430 2.636e-11 ***
## risk:adversity  2   16.419  67.168 10.859  0.004385 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

classroom.5=update(classroom.4,.~.-risk:adversity)
drop1(classroom.5,test="Chisq")

## Single term deletions
##
## Model:
## freq ~ behaviour + risk + adversity
##           Df Deviance     AIC    LRT  Pr(>Chi)
## <none>         16.419  67.168
## behaviour  1   60.849 109.598 44.430 2.636e-11 ***
## risk       1   18.748  67.497  2.329     0.127
## adversity  2   53.959 100.708 37.540 7.053e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

classroom.6=update(classroom.5,.~.-risk)
drop1(classroom.6,test="Chisq")

## Single term deletions
##
## Model:
## freq ~ behaviour + adversity
##           Df Deviance     AIC   LRT  Pr(>Chi)
## <none>         18.748  67.497
## behaviour  1   63.178 109.927 44.43 2.636e-11 ***
## adversity  2   56.288 101.037 37.54 7.053e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 33: Classroom behaviour analysis step 3

```
tab.2=xtabs(freq~behaviour,data=classroom)
tab.2

## behaviour
##     deviant non_deviant
##          17          80

tab.3=xtabs(freq~adversity,data=classroom)
tab.3

## adversity
##   high    low medium
##     12     25     60

tab.4=xtabs(freq~behaviour,data=classroom)
tab.4

## behaviour
##     deviant non_deviant
##          17          80

tab.5=xtabs(freq~risk+adversity,data=classroom)
tab.5

##              adversity
## risk          high low medium
##   at_risk        6   8     42
##   not_at_risk    6  17     18

prop.table(tab.5,1)

##              adversity
## risk               high       low    medium
##   at_risk     0.1071429 0.1428571 0.7500000
##   not_at_risk 0.1463415 0.4146341 0.4390244
```

Figure 34: Classroom sub-tables

```r
ggplot(qq,aes(x=size,y=response_rate,colour=colour))+
  geom_point()+geom_smooth(method="lm",se=F)+
  scale_colour_manual(values=c("blue","green","orange"))
```

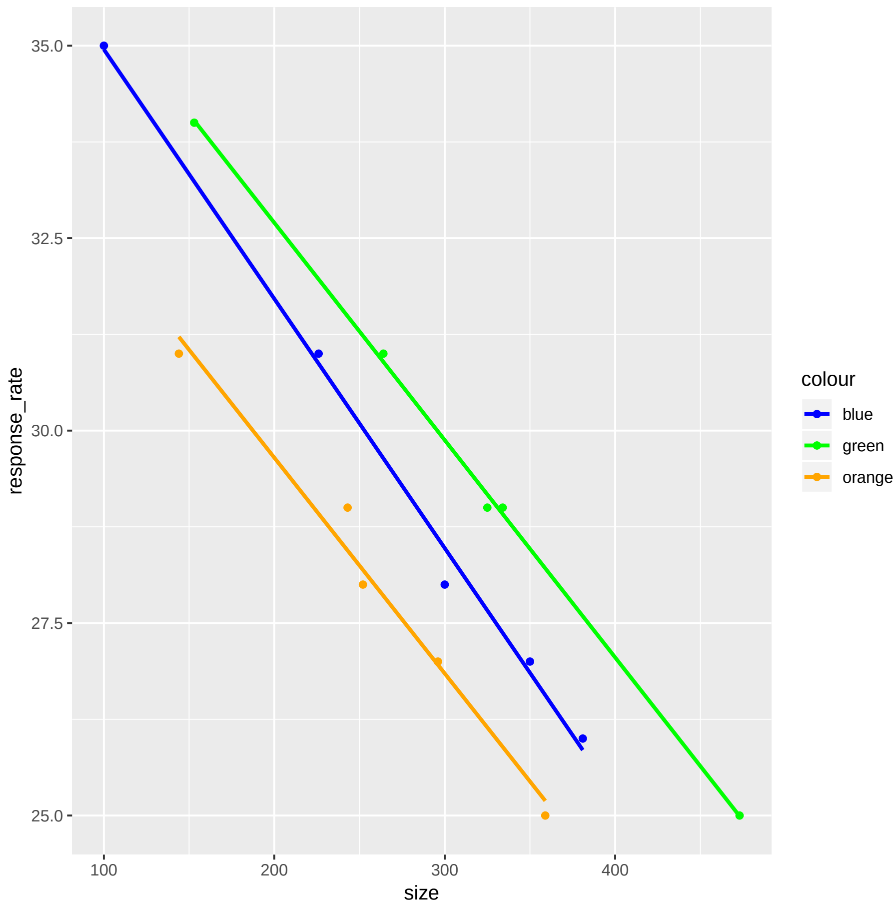

Figure 35: Questionnaire scatterplot

```
ptsd %>%
  gather(time,symptoms,pre:followup) %>%
  mutate(realtime=ordered(time,c("pre","post","followup"))) %>%
  ggplot(aes(x=realtime,y=symptoms,group=patient,colour=trt))+
    geom_point()+geom_line()
```

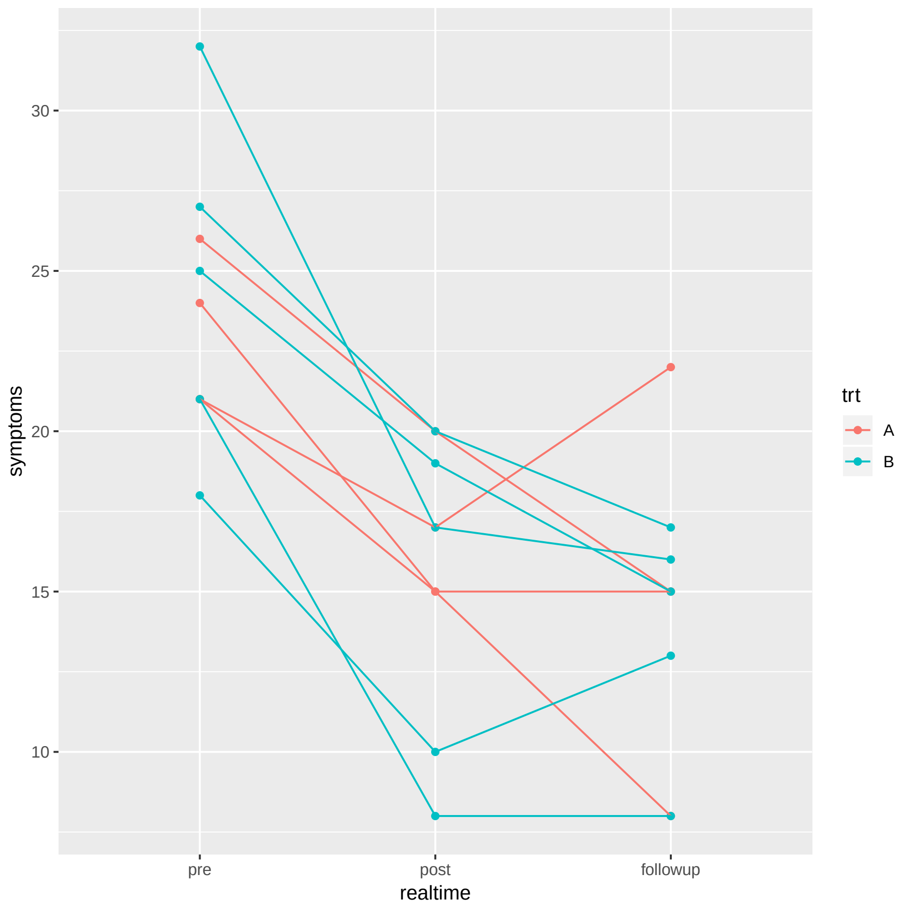

Figure 36: PTSD spaghetti plot

```r
ggbiplot(summit.2,groups=summit.s$Reach)
```

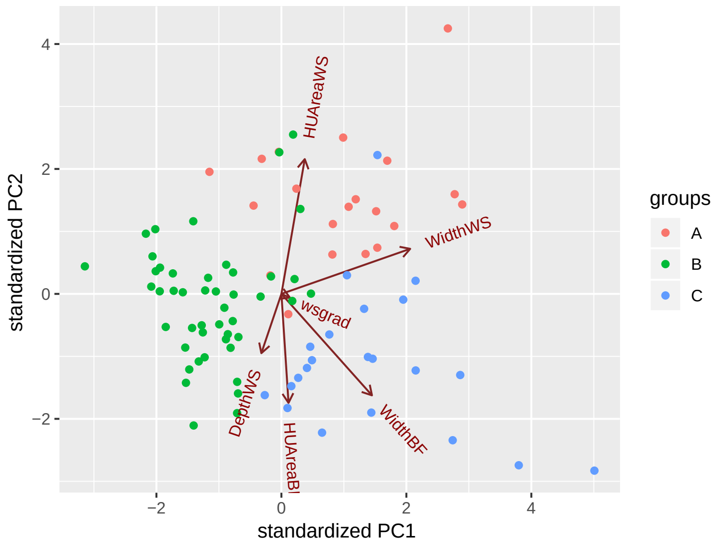

Figure 37: Summit Creek: biplot

```r
pp=predict(summit.2)
data.frame(Reach=summit.s$Reach,pp$x) %>%
  ggplot(aes(x=LD1,y=LD2,colour=Reach))+geom_point()
```

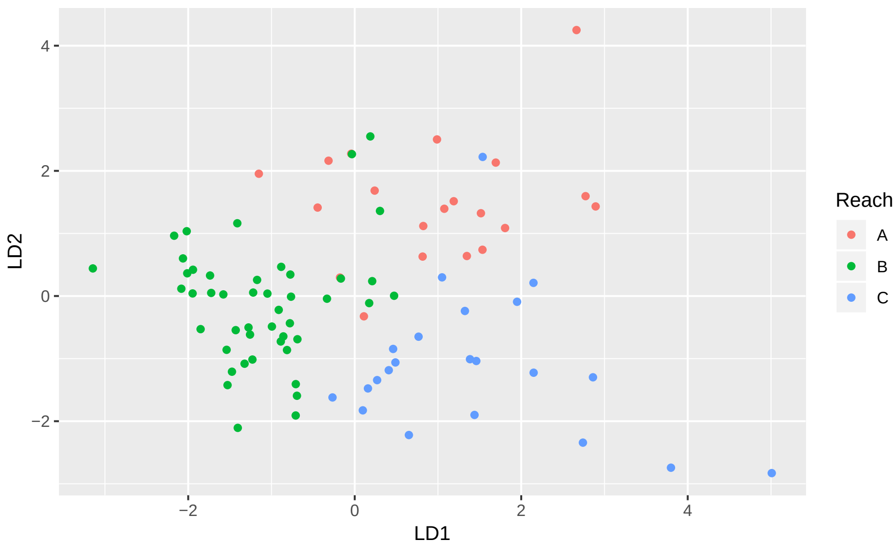

Figure 38: Summit Creek: plot of discriminant scores