Booklet of Figures
for
STAD29/STA 1007 Final Exam

List of Figures in this document by page:

List of Figures

```r
library(ggbiplot)
library(MASS)
library(lubridate)
library(tidyverse)
library(broom)
library(survival)
library(survminer)
library(nnet)
library(car)
library(tmaptools)
```

Figure 1: Packages

```
##    Glass Temp Light
## 1      A  100   580
## 2      A  100   568
## 3      A  100   570
## 4      B  100   550
## 5      B  100   530
## 6      B  100   579
## 7      C  100   546
## 8      C  100   575
## 9      C  100   599
## 10     A  125  1090
## 11     A  125  1087
## 12     A  125  1085
## 13     B  125  1070
## 14     B  125  1035
## 15     B  125  1000
## 16     C  125  1045
## 17     C  125  1053
## 18     C  125  1066
## 19     A  150  1392
## 20     A  150  1380
## 21     A  150  1386
## 22     B  150  1328
## 23     B  150  1312
## 24     B  150  1299
## 25     C  150   867
## 26     C  150   904
## 27     C  150   889
```

Figure 2: GTL data

```r
gtl %>%
  group_by(Glass, Temp) %>%
  summarize(mean_light = mean(Light)) -> gtl_means
```

```
## `summarise()` has grouped output by 'Glass'. You can override
using the `.groups`
## argument.
```

```r
ggplot(gtl_means, aes(x = Temp, y = mean_light, colour = Glass, group = Glass)) +
  geom_point() + geom_line()
```

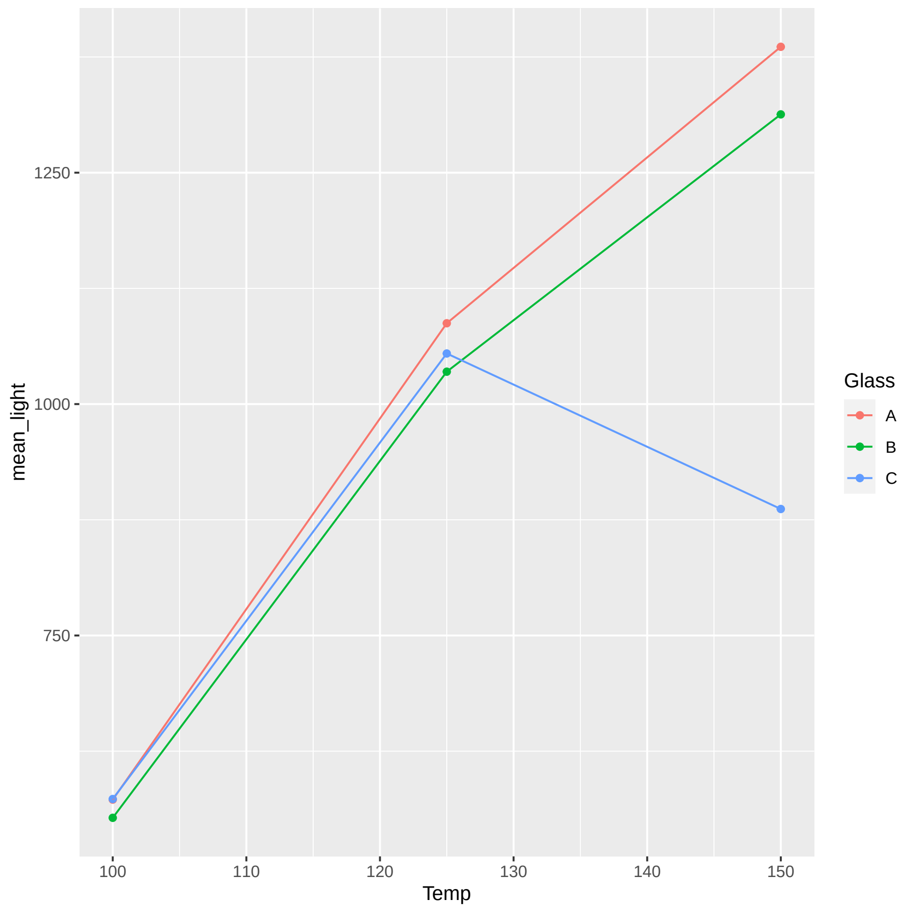

Figure 3: Plot of GTL data

```
gtl.1 <- aov(Light ~ Glass * factor(Temp), data = gtl)
summary(gtl.1)

##                     Df  Sum Sq Mean Sq F value   Pr(>F)
## Glass                2  150865   75432   206.4 3.89e-13 ***
## factor(Temp)         2 1970335  985167  2695.3  < 2e-16 ***
## Glass:factor(Temp)  4  290552   72638   198.7 1.25e-14 ***
## Residuals          18    6579     366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 4: ANOVA for GTL data

```
gtl %>% filter(Temp == 100) %>%
  aov(Light ~ Glass, data = .) -> temp100
summary(temp100)

##             Df Sum Sq Mean Sq F value Pr(>F)
## Glass        2  800.7   400.3   0.888  0.459
## Residuals    6 2705.3   450.9
```

Figure 5: More analysis for GTL data

```r
gtl %>% filter(Temp == 150) %>%
  aov(Light ~ Glass, data = .) -> temp150
summary(temp150)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Glass        2 436423  218211    1103 1.99e-08 ***
## Residuals    6   1187     198                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(temp150)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = Light ~ Glass, data = .)
## 
## $Glass
##          diff       lwr       upr     p adj
## B-A  -73.0000 -108.2320  -37.7680 0.0017263
## C-A -499.3333 -534.5653 -464.1013 0.0000000
## C-B -426.3333 -461.5653 -391.1013 0.0000001
```

Figure 6: Yet more analysis for the GTL data

```r
wine <- read_rds("wine_data.rds")
glimpse(wine)

## Rows: 178
## Columns: 14
## $ cultivar             <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ alcohol              <dbl> 14.23, 13.20, 13.16, 14.37, 13.24, 14.20, 14.39, ~
## $ malic_acid           <dbl> 1.71, 1.78, 2.36, 1.95, 2.59, 1.76, 1.87, 2.15, 1~
## $ ash                  <dbl> 2.43, 2.14, 2.67, 2.50, 2.87, 2.45, 2.45, 2.61, 2~
## $ ash_alkalinity       <dbl> 15.6, 11.2, 18.6, 16.8, 21.0, 15.2, 14.6, 17.6, 1~
## $ magnesium            <dbl> 127, 100, 101, 113, 118, 112, 96, 121, 97, 98, 10~
## $ phenols_total        <dbl> 2.80, 2.65, 2.80, 3.85, 2.80, 3.27, 2.50, 2.60, 2~
## $ flavonoids           <dbl> 3.06, 2.76, 3.24, 3.49, 2.69, 3.39, 2.52, 2.51, 2~
## $ phenols_nonflavonoid <dbl> 0.28, 0.26, 0.30, 0.24, 0.39, 0.34, 0.30, 0.31, 0~
## $ proanthocyanins      <dbl> 2.29, 1.28, 2.81, 2.18, 1.82, 1.97, 1.98, 1.25, 1~
## $ colour_intensity     <dbl> 5.64, 4.38, 5.68, 7.80, 4.32, 6.75, 5.25, 5.05, 5~
## $ hue                  <dbl> 1.04, 1.05, 1.03, 0.86, 1.04, 1.05, 1.02, 1.06, 1~
## $ od280_315            <dbl> 3.92, 3.40, 3.17, 3.45, 2.93, 2.85, 3.58, 3.58, 2~
## $ proline              <dbl> 1065, 1050, 1185, 1480, 735, 1450, 1290, 1295, 10~
```

Figure 7: Italian wine data (some)

```r
wine %>% select(-cultivar) %>%
  as.matrix() -> response
```

```r
wines.1 <- manova(response~factor(cultivar), data = wine)
summary(wines.1)

##                   Df Pillai approx F num Df den Df    Pr(>F)
## factor(cultivar)   2 1.7058   73.151     26    328 < 2.2e-16 ***
## Residuals        175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 8: Wine data MANOVA

Note that the `.` in the `lda` line means "all the other variables".

```
wine.2 <- lda(factor(cultivar) ~ .,  data = wine)
wine.2

## Call:
## lda(factor(cultivar) ~ ., data = wine)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3314607 0.3988764 0.2696629
##
## Group means:
##    alcohol malic_acid      ash ash_alkalinity magnesium phenols_total
## 1 13.74475   2.010678 2.455593       17.03729  106.3390      2.840169
## 2 12.27873   1.932676 2.244789       20.23803   94.5493      2.258873
## 3 13.15375   3.333750 2.437083       21.41667   99.3125      1.678750
##   flavonoids phenols_nonflavonoid proanthocyanins colour_intensity       hue
## 1  2.9823729             0.290000        1.899322         5.528305 1.0620339
## 2  2.0808451             0.363662        1.630282         3.086620 1.0562817
## 3  0.7814583             0.447500        1.153542         7.396250 0.6827083
##   od280_315   proline
## 1  3.157797 1115.7119
## 2  2.785352  519.5070
## 3  1.683542  629.8958
##
## Coefficients of linear discriminants:
##                               LD1           LD2
## alcohol              -0.403399781  0.8717930699
## malic_acid            0.165254596  0.3053797325
## ash                  -0.369075256  2.3458497486
## ash_alkalinity        0.154797889 -0.1463807654
## magnesium            -0.002163496 -0.0004627565
## phenols_total         0.618052068 -0.0322128171
## flavonoids           -1.661191235 -0.4919980543
## phenols_nonflavonoid -1.495818440 -1.6309537953
## proanthocyanins       0.134092628 -0.3070875776
## colour_intensity      0.355055710  0.2532306865
## hue                  -0.818036073 -1.5156344987
## od280_315            -1.157559376  0.0511839665
## proline              -0.002691206  0.0028529846
##
## Proportion of trace:
##    LD1    LD2
## 0.6875 0.3125
```

Figure 9: Wine discriminant analysis

```
wine.3 <- predict(wine.2)
d <- data.frame(cultivar = factor(wine$cultivar), wine.3$x)
ggplot(d, aes(x=LD1, y = LD2, colour = cultivar)) + geom_point()
```

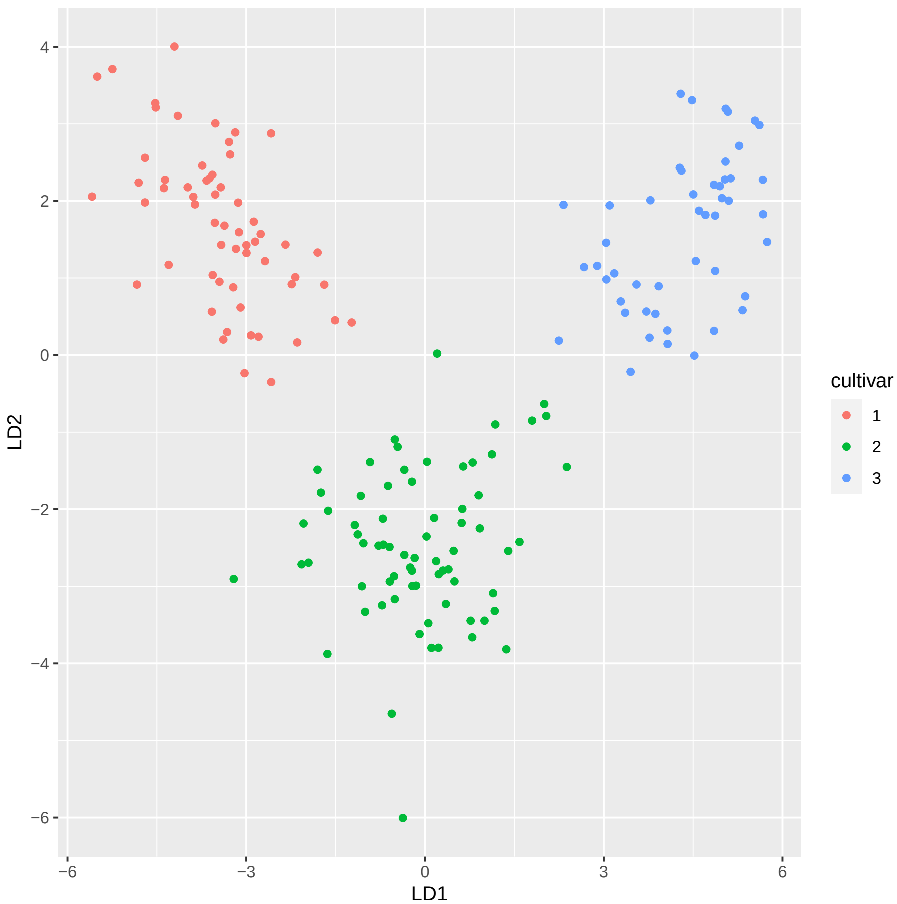

Figure 10: Wine data plot of discriminant scores

```
wine.4 <- lda(factor(cultivar) ~ ., data = wine, CV = TRUE)
table(cultivar = wine$cultivar, pred = wine.4$class)
```

```
##         pred
## cultivar  1  2  3
##        1 59  0  0
##        2  1 69  1
##        3  0  0 48
```

```
d <- data.frame(cultivar = wine$cultivar, pred = wine.4$class, round(wine.4$posterior, 3) )
d %>% rowwise() %>%
  filter(cultivar != pred)
```

```
## # A tibble: 2 x 5
## # Rowwise:
##   cultivar pred     X1    X2    X3
##      <dbl> <fct> <dbl> <dbl> <dbl>
## 1        2 3     0     0.156 0.844
## 2        2 1     0.658 0.342 0
```

Figure 11: Wine data misclassifications

```
## # A tibble: 1,575 x 6
##    hvltt hvltt2 hvltt3 hvltt4 treatment    id
##    <dbl>  <dbl>  <dbl>  <dbl> <fct>     <int>
##  1    28     28     17     22 control       1
##  2    24     22     20     27 control       2
##  3    24     24     28     27 reasoning     3
##  4    35     34     32     34 control       4
##  5    35     29     34     34 speed         5
##  6    29     27     26     29 control       6
##  7    18     16     27     30 control       7
##  8    25     26     25     29 speed         8
##  9    24     17     20     11 speed         9
## 10    22     19     21     26 speed        10
## # ... with 1,565 more rows
```

Figure 12: ACTIVE data

```
active %>%
  pivot_longer(starts_with("hvl"), names_to = "time", values_to = "score") %>%
  group_by(treatment, time) %>%
  summarize(n = n(), mean_score = mean(score), sd_score = sd(score)) -> active_summary

## `summarise()` has grouped output by 'treatment'. You can
override using the `.groups`
## argument.

active_summary

## # A tibble: 16 x 5
## # Groups:   treatment [4]
##    treatment time       n mean_score sd_score
##    <fct>     <chr>  <int>      <dbl>    <dbl>
##  1 control   hvltt    392       27.1     4.95
##  2 control   hvltt2   392       26.1     5.29
##  3 control   hvltt3   392       27.6     4.85
##  4 control   hvltt4   392       28.6     5.41
##  5 memory    hvltt    387       26.8     5.14
##  6 memory    hvltt2   387       24.5     5.31
##  7 memory    hvltt3   387       26.7     4.97
##  8 memory    hvltt4   387       26.4     6.16
##  9 reasoning hvltt    407       27.1     4.58
## 10 reasoning hvltt2   407       24.9     5.12
## 11 reasoning hvltt3   407       26.9     4.80
## 12 reasoning hvltt4   407       27.0     5.71
## 13 speed     hvltt    389       26.4     5.23
## 14 speed     hvltt2   389       24.1     5.63
## 15 speed     hvltt3   389       26.4     5.05
## 16 speed     hvltt4   389       26.2     6.04
```

Figure 13: ACTIVE data summary

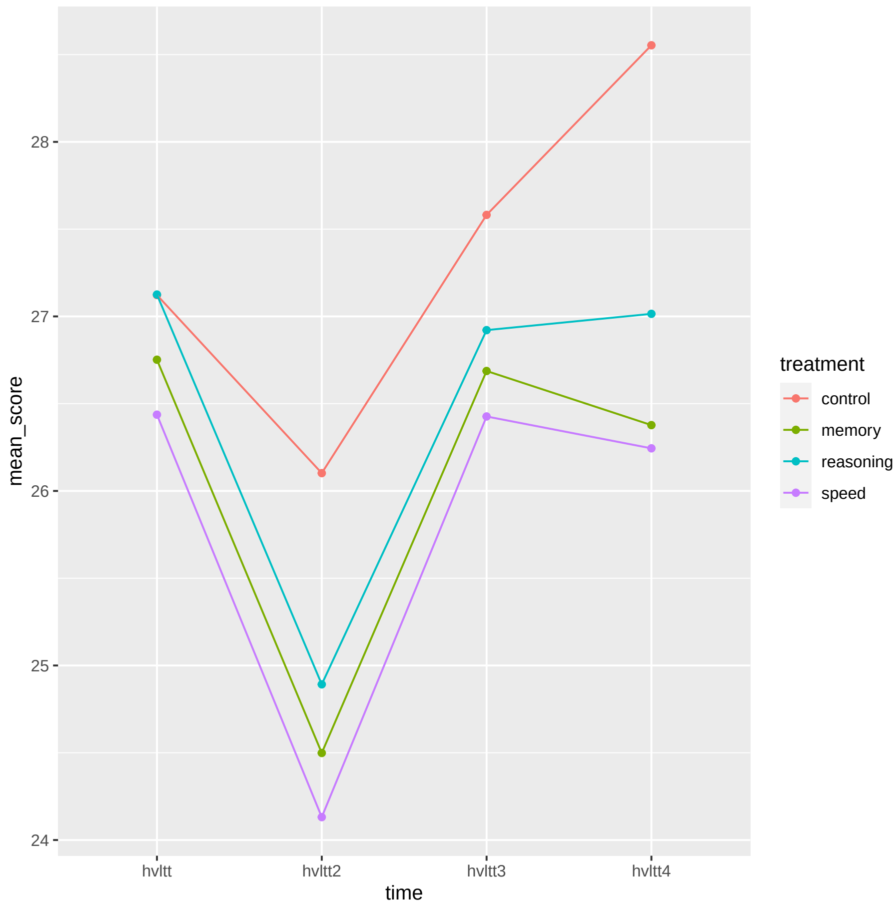

Figure 14: ACTIVE data interaction plot

```r
active %>%
  select(starts_with("hvl")) %>%
  as.matrix() -> response
active.1 <- lm(response ~ treatment, data = active)
times <- colnames(response)
times.df <- data.frame(times = factor(times))
Manova(active.1, idata = times.df, idesign = ~times)

##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##                 Df test stat approx F num Df den Df    Pr(>F)
## (Intercept)      1   0.97132    53209      1   1571 < 2.2e-16 ***
## treatment        3   0.01585        8      3   1571 1.464e-05 ***
## times            1   0.24053      166      3   1569 < 2.2e-16 ***
## treatment:times  3   0.03349        6      9   4713 2.984e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 15: ACTIVE study MANOVA

```
##              Atlanta Chicago Denver Houston LosAngeles Miami NewYork SanFrancisco Seattle
## Chicago          587
## Denver          1212     920
## Houston          701     940    879
## LosAngeles      1936    1745    831    1374
## Miami            604    1188   1726     968       2339
## NewYork          748     713   1631    1420       2451  1092
## SanFrancisco    2139    1858    949    1645        347  2594    2571
## Seattle         2182    1737   1021    1891        959  2734    2408          678
## WashingtonDC     543     597   1494    1220       2300   923     205         2442    2329
```

Figure 16: US city air distances

```r
cities.1 <- hclust(distance_grid, method = "complete")
plot(cities.1)
```

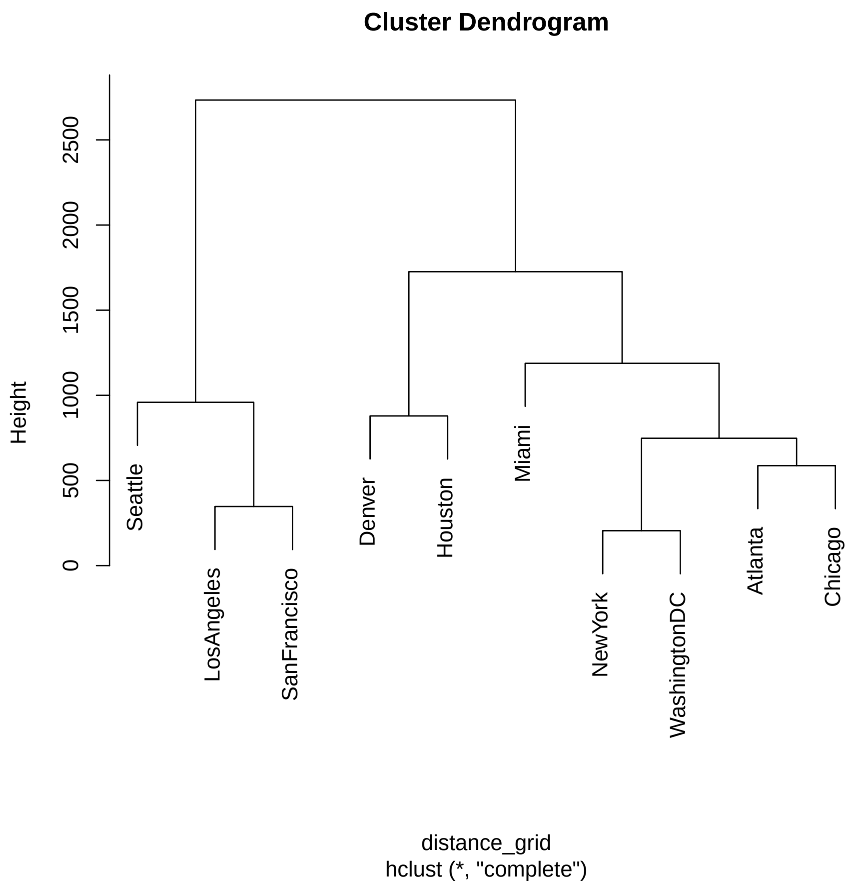

Figure 17: US city dendrogram

```
## Joining, by = "city"

## # A tibble: 10 x 3
##    city            lat    lon
##    <chr>         <dbl>  <dbl>
##  1 Atlanta        33.7  -84.4
##  2 Chicago        41.9  -87.6
##  3 Denver         39.7 -105.
##  4 Houston        29.8  -95.4
##  5 Los Angeles    34.1 -118.
##  6 Miami          25.8  -80.2
##  7 New York       40.7  -74.0
##  8 San Francisco  37.8 -122.
##  9 Seattle        47.6 -122.
## 10 Washington DC  38.9  -77.0
```

Figure 18: Latitudes and longitudes of US cities

The game of basketball is played between two teams of five players each. The aim is to shoot a ball through a “basket” consisting of a metal rim with a net below. (The net has a hole in the bottom so that the ball falls through, but the net slows it down so that you can see that the ball actually did pass through). A successful shot is usually worth two points. There are detailed rules about how players are allowed to compete; a player who breaks these rules commits a foul, and sometimes the player who is fouled gets to attempt one or two “free throws” (shots) from a marked line without any other players in the way. A successful free throw is worth one point. In addition, there is a line on the court some distance away from the basket; a successful shot from behind this line is worth three points rather than two (but of course is less likely to succeed than a shot taken from close to the basket).

If a player takes a shot that does not go through the basket, it will usually hit the metal rim and bounce out. The player that catches the ball after it has bounced off the rim is credited with a “rebound”. In this dataset we distinguish between offensive and defensive rebounds. If team A shoots the ball, misses, and another player from team A catches the ball after it rebounds from the rim, the player gets an “offensive rebound”. If, on the other hand, a player from the other team B catches the ball, that is a “defensive rebound”.

A player that passes the ball to a teammate who then makes a successful shot can be credited with an “assist”. A player who (within the rules) takes the ball away from an opponent, or who intercepts a pass made by an opponent, is credited with a “steal”. If a defending player gets in the way of a shot by an opponent so that the shot is then missed, that is a “block”. A player who causes his team to lose the ball before taking a shot commits a “turnover” (so that a high number of turnovers is bad). None of these score a team any points, but they can result in the player’s team scoring (or losing) points later, so they are valuable information about how well a player is playing.

Figure 19: Basketball information

```
## # A tibble: 1,002 x 10
##    player_name         fg_pct fg3_pct ft_pct  oreb  dreb   ast   stl   blk   tov
##    <chr>                <dbl>   <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Michael Jordan       0.497   0.327  0.835 1.56   4.67  5.25 2.35  0.833  2.73
##  2 Kevin Durant         0.488   0.379  0.882 0.787  6.37  3.79 1.19  1.05   3.16
##  3 LeBron James         0.501   0.342  0.74  1.21   6.05  7.03 1.65  0.770  3.41
##  4 Allen Iverson        0.425   0.313  0.78  0.815  2.90  6.15 2.17  0.179  3.57
##  5 George Gervin        0.511   0.297  0.844 1.50   3.06  2.80 1.19  0.847  3.01
##  6 Karl Malone          0.516   0.274  0.742 2.41   7.73  3.56 1.41  0.776  3.07
##  7 Kobe Bryant          0.447   0.329  0.837 1.11   4.12  4.68 1.44  0.475  2.98
##  8 Dominique Wilkins    0.461   0.319  0.811 2.75   3.93  2.49 1.28  0.598  2.49
##  9 Carmelo Anthony      0.452   0.346  0.813 1.78   4.80  3.13 1.06  0.483  2.79
## 10 Kareem Abdul-Jabbar  0.559   0.056  0.721 2.40   7.58  3.63 0.936 2.57   2.72
## # ... with 992 more rows
```

Figure 20: Basketball data (some)

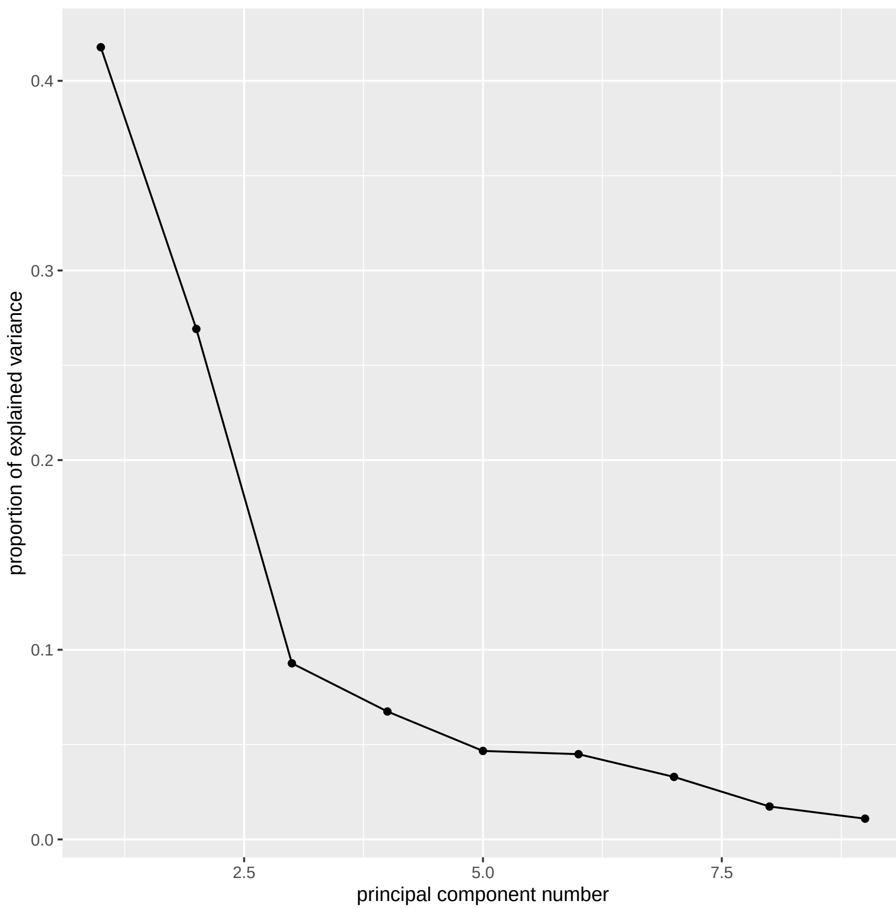

Figure 21: Basketball scree plot

```
##
## Loadings:
##         Factor1 Factor2
## fg_pct   0.605
## fg3_pct -0.495   0.229
## ft_pct  -0.478   0.346
## oreb     0.957
## dreb     0.862   0.197
## ast     -0.312   0.880
## stl     -0.107   0.779
## blk      0.692
## tov      0.223   0.867
##
##                Factor1 Factor2
## SS loadings      3.135   2.354
## Proportion Var   0.348   0.262
## Cumulative Var   0.348   0.610
```

Figure 22: Basketball factor analysis, showing factor loadings

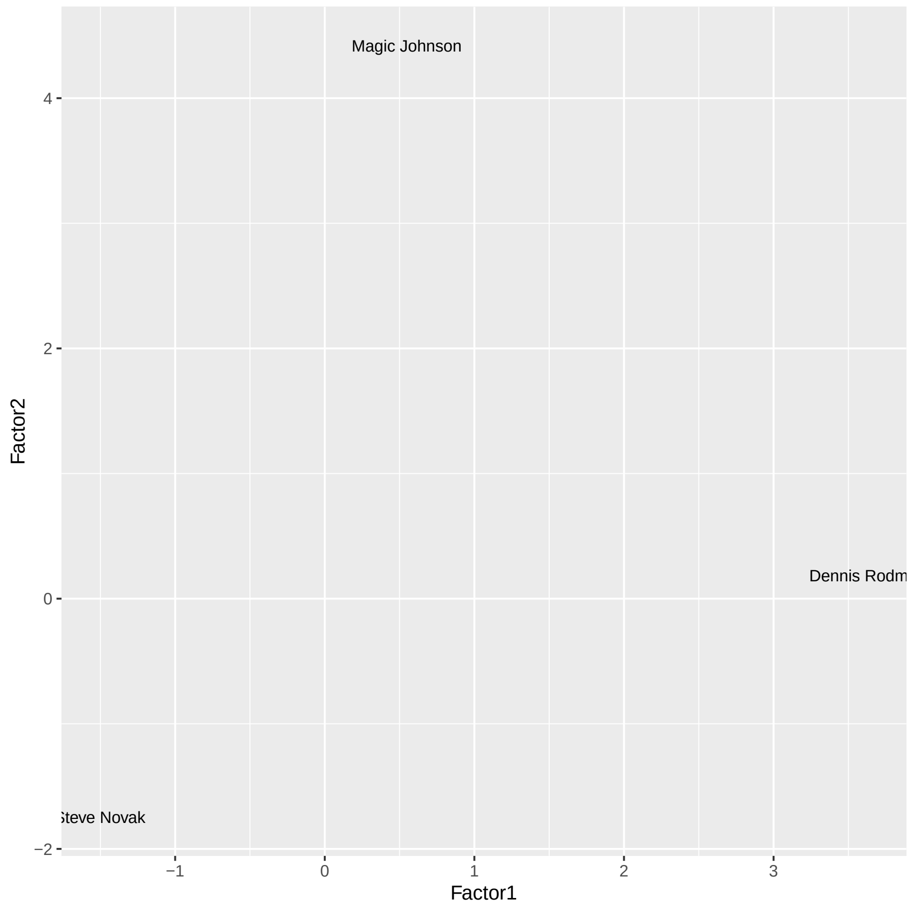

Figure 23: Factor score plot for three players

```
## # A tibble: 3 x 10
##   player_name   fg_pct fg3_pct ft_pct  oreb  dreb    ast   stl    blk   tov
##   <chr>          <dbl>   <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>
## 1 Magic Johnson  0.52    0.303  0.848 1.77   5.47 11.2   1.90  0.413  3.87
## 2 Dennis Rodman  0.521   0.231  0.584 4.75   8.37  1.76  0.671 0.583  1.63
## 3 Steve Novak    0.437   0.43   0.877 0.146  1.12  0.283 0.195 0.0814 0.173
```

Figure 24: Original data for three players

```
## # A tibble: 3 x 10
##   player_name   fg_pct fg3_pct ft_pct  oreb   dreb     ast     stl    blk   tov
##   <chr>          <dbl>   <dbl>  <dbl> <dbl>  <dbl>   <dbl>   <dbl>  <dbl> <dbl>
## 1 Magic Johnson  0.918   0.529 0.916  0.721 0.912  1       0.983   0.565  0.999
## 2 Dennis Rodman  0.920   0.375 0.0460 0.999 0.996  0.445   0.368   0.690  0.576
## 3 Steve Novak    0.246   0.989 0.981  0     0.0280 0.00400 0.00500 0.0719 0
```

Figure 25: Percentile ranks for three players

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   socioeconomic = col_character(),
##   boy_scout = col_character(),
##   Yes = col_double(),
##   No = col_double()
## )

## # A tibble: 12 x 4
##    socioeconomic boy_scout delinquent frequency
##    <fct>         <chr>     <chr>          <dbl>
##  1 Low           Yes       Yes               11
##  2 Low           Yes       No                43
##  3 Low           No        Yes               42
##  4 Low           No        No               169
##  5 Medium        Yes       Yes               14
##  6 Medium        Yes       No               104
##  7 Medium        No        Yes               20
##  8 Medium        No        No               132
##  9 High          Yes       Yes                8
## 10 High          Yes       No               196
## 11 High          No        Yes                2
## 12 High          No        No                59
```

Figure 26: Boy Scouts data

```r
xt <- xtabs(frequency ~ boy_scout + delinquent, data = scouts)
xt

##          delinquent
## boy_scout  No Yes
##       No  360  64
##       Yes 343  33

prop.table(xt, margin = 1)

##          delinquent
## boy_scout         No        Yes
##       No  0.84905660 0.15094340
##       Yes 0.91223404 0.08776596
```

Figure 27: Boy Scouts table

```
scouts.1 <- glm(frequency ~ socioeconomic*boy_scout*delinquent,
                data = scouts, family = "poisson")
drop1(scouts.1, test = "Chisq")

## Single term deletions
##
## Model:
## frequency ~ socioeconomic * boy_scout * delinquent
##                                     Df Deviance    AIC     LRT Pr(>Chi)
## <none>                                  0.00000 88.526
## socioeconomic:boy_scout:delinquent  2   0.15429 84.680 0.15429   0.9258

scouts.2 <- update(scouts.1, .~. - socioeconomic:boy_scout:delinquent)
drop1(scouts.2, test = "Chisq")

## Single term deletions
##
## Model:
## frequency ~ socioeconomic + boy_scout + delinquent + socioeconomic:boy_scout +
##     socioeconomic:delinquent + boy_scout:delinquent
##                          Df Deviance     AIC     LRT  Pr(>Chi)
## <none>                         0.154  84.680
## socioeconomic:boy_scout   2  174.797 255.323 174.643 < 2.2e-16 ***
## socioeconomic:delinquent  2   28.802 109.328  28.648 6.015e-07 ***
## boy_scout:delinquent      1    0.162  82.688   0.008    0.9285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

scouts.3 <- update(scouts.2, .~. - boy_scout:delinquent)
drop1(scouts.3, test = "Chisq")

## Single term deletions
##
## Model:
## frequency ~ socioeconomic + boy_scout + delinquent + socioeconomic:boy_scout +
##     socioeconomic:delinquent
##                          Df Deviance     AIC     LRT  Pr(>Chi)
## <none>                         0.162  82.688
## socioeconomic:boy_scout   2  182.410 260.936 182.248 < 2.2e-16 ***
## socioeconomic:delinquent  2   36.415 114.940  36.252 1.342e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 28: Boy Scouts analysis

```r
xt <- xtabs(frequency ~ socioeconomic + boy_scout, data = scouts)
xt

##              boy_scout
## socioeconomic  No Yes
##        Low    211  54
##        Medium 152 118
##        High    61 204

prop.table(xt, margin = 1)

##              boy_scout
## socioeconomic        No       Yes
##        Low    0.7962264 0.2037736
##        Medium 0.5629630 0.4370370
##        High   0.2301887 0.7698113
```

```r
xt <- xtabs(frequency ~ socioeconomic + delinquent, data = scouts)
xt

##              delinquent
## socioeconomic  No Yes
##        Low    212  53
##        Medium 236  34
##        High   255  10

prop.table(xt, margin = 1)

##              delinquent
## socioeconomic         No        Yes
##        Low    0.80000000 0.20000000
##        Medium 0.87407407 0.12592593
##        High   0.96226415 0.03773585
```

Figure 29: Boy Scouts more tables