Booklet of Code and Output

for

STAD29/STA 1007 Final Exam

List of Figures in this document by page:

List of Figures

Figure captions are *below* the Figure they refer to.

```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

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()

library(broom)

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

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':
##
##    some

library(ggrepel)
```

Dose	SampSize	Deaths
1	250	28
2	250	53
3	250	93
4	250	126
5	250	172
6	250	197

Figure 2: Toxicity data

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -2.64     0.156      -16.9 2.47e-64
## 2 Dose           0.674    0.0391      17.2 1.48e-66
```

Figure 3: Logistic regression for insects data

```
puzzle=read_delim("puzzle.txt"," ")

## Parsed with column specification:
## cols(
##   reward = col_character(),
##   attempts = col_double()
## )

puzzle

## # A tibble: 20 x 2
##    reward     attempts
##    <chr>         <dbl>
##  1 Constant         12
##  2 Constant         13
##  3 Constant         11
##  4 Constant         12
##  5 Constant         12
##  6 Frequent          9
##  7 Frequent         10
##  8 Frequent          9
##  9 Frequent         13
## 10 Frequent         14
## 11 Infrequent       15
## 12 Infrequent       16
## 13 Infrequent       17
## 14 Infrequent       16
## 15 Infrequent       16
## 16 Never            17
## 17 Never            18
## 18 Never            12
## 19 Never            18
## 20 Never            20
```

Figure 4: Puzzle data

```
my_levels=c("Constant","Frequent","Infrequent","Never")
puzzle = puzzle %>% mutate(rewardf=ordered(reward,levels=my_levels))
```

Figure 5: Creating a factor

```
attempts.1=lm(attempts~rewardf,data=puzzle)
summary(attempts.1)

##
## Call:
## lm(formula = attempts ~ rewardf, data = puzzle)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##     -5     -1      0      1      3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.0000     0.4402  31.806 6.83e-16 ***
## rewardfC1    -3.0000     0.7624  -3.935 0.001183 **
## rewardfC2    -1.0000     0.7188  -1.391 0.183204
## rewardfC3    -2.5000     0.6225  -4.016 0.000998 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.969 on 16 degrees of freedom
## Multiple R-squared:  0.6771,Adjusted R-squared:  0.6165
## F-statistic: 11.18 on 3 and 16 DF,  p-value: 0.0003332
```

Figure 6: Analysis of puzzle data

```r
diabetes=read_csv("diabetes.csv")

## Parsed with column specification:
## cols(
##   Treatment = col_character(),
##   Age = col_double(),
##   FBS_change = col_double()
## )

diabetes

## # A tibble: 20 x 3
##    Treatment   Age FBS_change
##    <chr>     <dbl>      <dbl>
##  1 diet         30         10
##  2 diet         50          5
##  3 diet         45          0
##  4 diet         60          5
##  5 diet         55         10
##  6 diet         40          5
##  7 diet         35          0
##  8 diet         45         10
##  9 diet         50          5
## 10 diet         55         10
## 11 insulin      55         10
## 12 insulin      60         20
## 13 insulin      55         10
## 14 insulin      70          5
## 15 insulin      50         10
## 16 insulin      60         15
## 17 insulin      50          5
## 18 insulin      45          0
## 19 insulin      65         10
## 20 insulin      50         15
```

Figure 7: Diabetes data

```
fbs.1=lm(FBS_change~Age*Treatment,data=diabetes)
anova(fbs.1)

## Analysis of Variance Table
##
## Response: FBS_change
##               Df Sum Sq Mean Sq F value Pr(>F)
## Age            1  68.29  68.293  2.6204 0.1250
## Treatment      1  30.78  30.776  1.1809 0.2933
## Age:Treatment  1   3.94   3.936  0.1510 0.7027
## Residuals     16 417.00  26.062
```

Figure 8: Analysis of covariance part 1

```
fbs.2=lm(FBS_change~Age+Treatment,data=diabetes)
anova(fbs.2)

## Analysis of Variance Table
##
## Response: FBS_change
##           Df Sum Sq Mean Sq F value Pr(>F)
## Age        1  68.29  68.293  2.7581 0.1151
## Treatment  1  30.78  30.776  1.2429 0.2804
## Residuals 17 420.93  24.761
```

Figure 9: Analysis of covariance part 2

```
  tidy(fbs.2) %>% select(term, estimate)

## # A tibble: 3 x 2
##   term             estimate
##   <chr>               <dbl>
## 1 (Intercept)         0.458
## 2 Age                 0.119
## 3 Treatmentinsulin    2.87
```

(Note: `tidy` comes from package `broom`.)

Figure 10: Analysis of covariance part 3

```
words=read_csv("vocal.csv")

## Parsed with column specification:
## cols(
##   id = col_double(),
##   unrelated = col_double(),
##   semantic = col_double(),
##   phonological = col_double()
## )

words

## # A tibble: 16 x 4
##        id unrelated semantic phonological
##     <dbl>     <dbl>    <dbl>        <dbl>
##  1      1        12       10           11
##  2      2        11        9            8
##  3      3         5        6            4
##  4      4         8        7            3
##  5      5        11        9           10
##  6      6         7        6            7
##  7      7         9        7            9
##  8      8        11        9            8
##  9      9         9        8            6
## 10     12         4        4            9
## 11     13        10        8            7
## 12     14         9       11            9
## 13     15        13       10           10
## 14     16         7        6            6
## 15     17         9        8            9
## 16     18         6       10            4
```

Figure 11: Word memory data

```r
response=with(words,cbind(unrelated, semantic, phonological))
words.1=lm(response~1,data=words)
list_types=colnames(response)
list_df=data.frame(list_types)
words.2=Manova(words.1,idata=list_df,idesign=~list_types)
words.2

##
## Type III Repeated Measures MANOVA Tests: Pillai test statistic
##             Df test stat approx F num Df den Df    Pr(>F)
## (Intercept)  1   0.95137  293.447      1     15 2.938e-11 ***
## list_types   1   0.32374    3.351      2     14   0.06468 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 12: Word memory analysis

```
words %>% gather(relatedness,recall,-id) %>%
    ggplot(aes(x=relatedness, y=recall, group=id)) +
      geom_point()+geom_line()
```

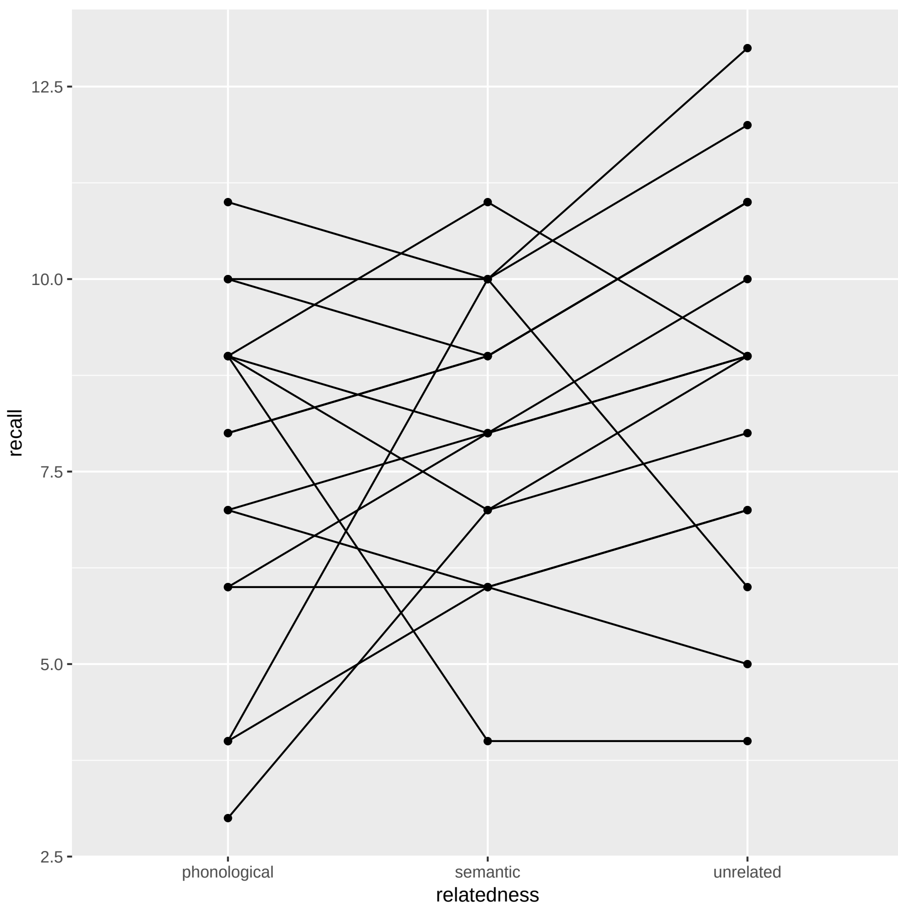


Figure 13: Word memory spaghetti plot

```r
rootstocks=read_csv("rootstocks.csv")

## Parsed with column specification:
## cols(
##   rootstock = col_double(),
##   girth4 = col_double(),
##   extension = col_double(),
##   girth15 = col_double(),
##   weight = col_double(),
##   row = col_double()
## )

rootstocks %>% print(n=Inf)

## # A tibble: 48 x 6
##    rootstock girth4 extension girth15 weight   row
##        <dbl>  <dbl>     <dbl>   <dbl>  <dbl> <dbl>
##  1         1  1.11      2.57     3.58  0.760     1
##  2         1  1.19      2.93     3.75  0.821     2
##  3         1  1.09      2.87     3.93  0.928     3
##  4         1  1.25      3.84     3.94  1.01      4
##  5         1  1.11      3.03     3.60  0.766     5
##  6         1  1.08      2.34     3.51  0.726     6
##  7         1  1.11      3.21     3.98  1.21      7
##  8         1  1.16      3.04     3.62  0.75      8
##  9         2  1.05      2.07     4.09  1.04      9
## 10         2  1.17      2.88     4.06  1.09     10
## 11         2  1.11      3.38     4.87  1.63     11
## 12         2  1.25      3.91     4.98  1.52     12
## 13         2  1.17      2.78     4.38  1.20     13
## 14         2  1.15      3.02     4.65  1.24     14
## 15         2  1.17      3.38     4.69  1.50     15
## 16         2  1.19      3.45     4.40  1.03     16
## 17         3  1.07      2.51     3.76  0.912    17
## 18         3  0.990     2.32     4.44  1.40     18
## 19         3  1.06      2.67     4.38  1.20     19
## 20         3  1.02      2.39     4.67  1.61     20
## 21         3  1.15      3.02     4.48  1.48     21
## 22         3  1.20      3.09     4.78  1.57     22
## 23         3  1.20      3.31     4.57  1.51     23
## 24         3  1.17      3.23     4.56  1.46     24
## 25         4  1.22      2.84     3.89  0.944    25
## 26         4  1.03      2.35     4.05  1.24     26
## 27         4  1.14      3.00     4.05  1.02     27
## 28         4  1.01      2.44     3.92  1.07     28
## 29         4  0.990     2.20     3.27  0.693    29
## 30         4  1.11      3.32     3.95  1.09     30
## 31         4  1.20      3.60     4.27  1.24     31
## 32         4  1.08      3.29     3.85  1.02     32
## 33         5  0.910     1.53     4.04  1.08     33
## 34         5  1.15      2.55     4.16  1.15     34
## 35         5  1.14      3.08     4.79  1.38     35
## 36         5  1.05      2.33     4.42  1.24     36
## 37         5  0.990     2.08     3.47  0.673    37
## 38         5  1.22      3.37     4.41  1.14     38
## 39         5  1.05      2.42     4.64  1.46     39
## 40         5  1.13      3.10     4.57  1.33     40
## 41         6  1.11      2.81     3.76  0.800    41
## 42         6  0.75      0.840    3.14  0.606    42
## 43         6  1.05      2.20     3.75  0.790    43
## 44         6  1.02      2.13     3.99  0.853    44
## 45         6  1.05      1.95     3.34  0.610    45
## 46         6  1.07      2.25     3.21  0.562    46
## 47         6  1.13      3.06     3.63  0.707    47
## 48         6  1.11      2.47     3.95  0.952    48
```


Figure 14: Rootstock data

```
##                   Df Pillai approx F num Df den Df    Pr(>F)
## factor(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
```

Figure 15: Rootstock MANOVA

```
rootstocks.2 = rootstocks %>%
    mutate(froot=factor(rootstock)) %>%
    lda(froot~girth4+extension+girth15+weight,data=.)
rootstocks.2$svd

## [1] 3.9693370 2.5771749 1.3870876 0.4669155

rootstocks.2$scaling

##                  LD1        LD2       LD3       LD4
## girth4     3.0479969  -1.140083 -1.002452 23.419065
## extension -1.7025951  -1.215889  1.672714 -3.076805
## girth15    4.2332621   7.166402  3.045555 -2.011415
## weight    -0.4785109 -11.520300 -5.506194  3.101660
```

Figure 16: Rootstock discriminant analysis

```
rootstocks.3=predict(rootstocks.2)
rootstocks.3$posterior %>% as_tibble() %>%
    mutate(obs=rootstocks$rootstock,
           pred=rootstocks.3$class,
           row=rootstocks$row) %>%
    print(n=Inf)
```

```
## # A tibble: 48 x 9
##          `1`      `2`      `3`      `4`      `5`      `6`   obs pred    row
##        <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <dbl> <fct> <dbl>
##  1 0.471     0.00548  0.00316  0.172    0.00939   3.39e-1     1 1         1
##  2 0.584     0.0188   0.00528  0.139    0.0161    2.37e-1     1 1         2
##  3 0.285     0.0787   0.0113   0.228    0.0775    3.19e-1     1 6         3
##  4 0.674     0.00151  0.00121  0.319    0.000227  4.38e-3     1 1         4
##  5 0.651     0.00127  0.000375 0.245    0.000934  1.01e-1     1 1         5
##  6 0.350     0.00442  0.00317  0.139    0.0118    4.91e-1     1 6         6
##  7 0.145     0.000876 0.0263   0.826    0.000428  1.35e-3     1 4         7
##  8 0.701     0.00257  0.000586 0.155    0.00183   1.39e-1     1 1         8
##  9 0.00309   0.161    0.0578   0.00546  0.685     8.78e-2     2 5         9
## 10 0.223     0.114    0.210    0.281    0.103     6.94e-2     2 4        10
## 11 0.000810  0.255    0.633    0.0224   0.0885    1.67e-4     2 3        11
## 12 0.000701  0.836    0.0533   0.00227  0.107     2.40e-4     2 2        12
## 13 0.00431   0.409    0.121    0.00642  0.444     1.51e-2     2 5        13
## 14 0.000215  0.577    0.00891  0.000318 0.410     3.41e-3     2 2        14
## 15 0.00474   0.299    0.553    0.0403   0.102     7.98e-4     2 3        15
## 16 0.0140    0.693    0.00186  0.00472  0.227     5.88e-2     2 2        16
## 17 0.305     0.0213   0.0267   0.344    0.0407    2.62e-1     3 4        17
## 18 0.000703  0.109    0.583    0.0165   0.288     2.70e-3     3 3        18
## 19 0.00310   0.378    0.0566   0.0103   0.528     2.39e-2     3 5        19
## 20 0.0000363 0.0262   0.914    0.00233  0.0575    5.58e-5     3 3        20
## 21 0.00433   0.0343   0.884    0.0546   0.0229    3.79e-4     3 3        21
## 22 0.000242  0.142    0.772    0.00247  0.0828    9.94e-5     3 3        22
## 23 0.00598   0.0544   0.862    0.0559   0.0210    2.73e-4     3 3        23
## 24 0.00779   0.135    0.732    0.0636   0.0611    9.27e-4     3 3        24
## 25 0.437     0.0747   0.0709   0.164    0.0735    1.79e-1     4 1        25
## 26 0.0264    0.0168   0.614    0.292    0.0412    1.00e-2     4 3        26
## 27 0.282     0.147    0.0444   0.260    0.110     1.57e-1     4 1        27
## 28 0.127     0.0340   0.112    0.545    0.0735    1.08e-1     4 4        28
## 29 0.433     0.000139 0.000688 0.412    0.000489  1.54e-1     4 1        29
## 30 0.297     0.00261  0.00650  0.685    0.00107   7.89e-3     4 4        30
## 31 0.266     0.0503   0.0839   0.581    0.0117    6.71e-3     4 4        31
## 32 0.323     0.00102  0.00173  0.665    0.000448  9.20e-3     4 4        32
## 33 0.000654  0.0651   0.0875   0.00507  0.780     6.19e-2     5 5        33
## 34 0.0248    0.183    0.406    0.0455   0.304     3.68e-2     5 3        34
## 35 0.000110  0.600    0.0264   0.000391 0.372     9.41e-4     5 2        35
## 36 0.000363  0.253    0.0834   0.00169  0.654     8.07e-3     5 5        36
## 37 0.120     0.00342  0.000882 0.0656   0.0153    7.95e-1     5 6        37
## 38 0.0254    0.661    0.0254   0.0159   0.243     2.96e-2     5 2        38
## 39 0.0000824 0.199    0.372    0.00131  0.427     7.18e-4     5 5        39
## 40 0.00342   0.539    0.114    0.0130   0.325     5.60e-3     5 2        40
## 41 0.368     0.0251   0.00244  0.131    0.0274    4.47e-1     6 6        41
## 42 0.0102    0.000216 0.000705 0.0350   0.0103    9.44e-1     6 6        42
## 43 0.0500    0.0451   0.00525  0.0254   0.157     7.18e-1     6 6        43
## 44 0.00283   0.134    0.00228  0.00182  0.533     3.27e-1     6 5        44
## 45 0.154     0.00171  0.00110  0.0425   0.00953   7.91e-1     6 6        45
## 46 0.507     0.000231 0.000233 0.108    0.000735  3.84e-1     6 1        46
## 47 0.619     0.00367  0.000218 0.120    0.00254   2.55e-1     6 1        47
## 48 0.0815    0.159    0.0551   0.0642   0.321     3.20e-1     6 5        48
```

Figure 17: Rootstock predictions

1. Alcohol
2. Malic acid
3. Ash
4. Alkalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Colour intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

If you don't know what these are, I probably don't know either!

Figure 18: Variables in wines data

```
wines=read_csv("wine.csv")

## Parsed with column specification:
## cols(
##   id = col_character(),
##   alcohol = col_double(),
##   malic_acid = col_double(),
##   ash = col_double(),
##   ash_alkalinity = col_double(),
##   magnesium = col_double(),
##   phenols = col_double(),
##   flavonoids = col_double(),
##   nonf_phenols = col_double(),
##   proanthocyanins = col_double(),
##   colour = col_double(),
##   hue = col_double(),
##   od280 = col_double(),
##   proline = col_double()
## )

wines

## # A tibble: 178 x 14
##    id    alcohol malic_acid   ash ash_alkalinity magnesium phenols
##    <chr>   <dbl>      <dbl> <dbl>          <dbl>     <dbl>   <dbl>
##  1 V001     14.2       1.71  2.43           15.6       127    2.8
##  2 V002     13.2       1.78  2.14           11.2       100    2.65
##  3 V003     13.2       2.36  2.67           18.6       101    2.8
##  4 V004     14.4       1.95  2.5            16.8       113    3.85
##  5 V005     13.2       2.59  2.87           21         118    2.8
##  6 V006     14.2       1.76  2.45           15.2       112    3.27
##  7 V007     14.4       1.87  2.45           14.6        96    2.5
##  8 V008     14.1       2.15  2.61           17.6       121    2.6
##  9 V009     14.8       1.64  2.17           14          97    2.8
## 10 V010     13.9       1.35  2.27           16          98    2.98
## # ... with 168 more rows, and 7 more variables: flavonoids <dbl>,
## #   nonf_phenols <dbl>, proanthocyanins <dbl>, colour <dbl>, hue <dbl>,
## #   od280 <dbl>, proline <dbl>
```

Figure 19: Wines data (some)

```
wines2 = wines %>% mutate_if(is.numeric,scale)
```

Figure 20: Some calculation with the wines data

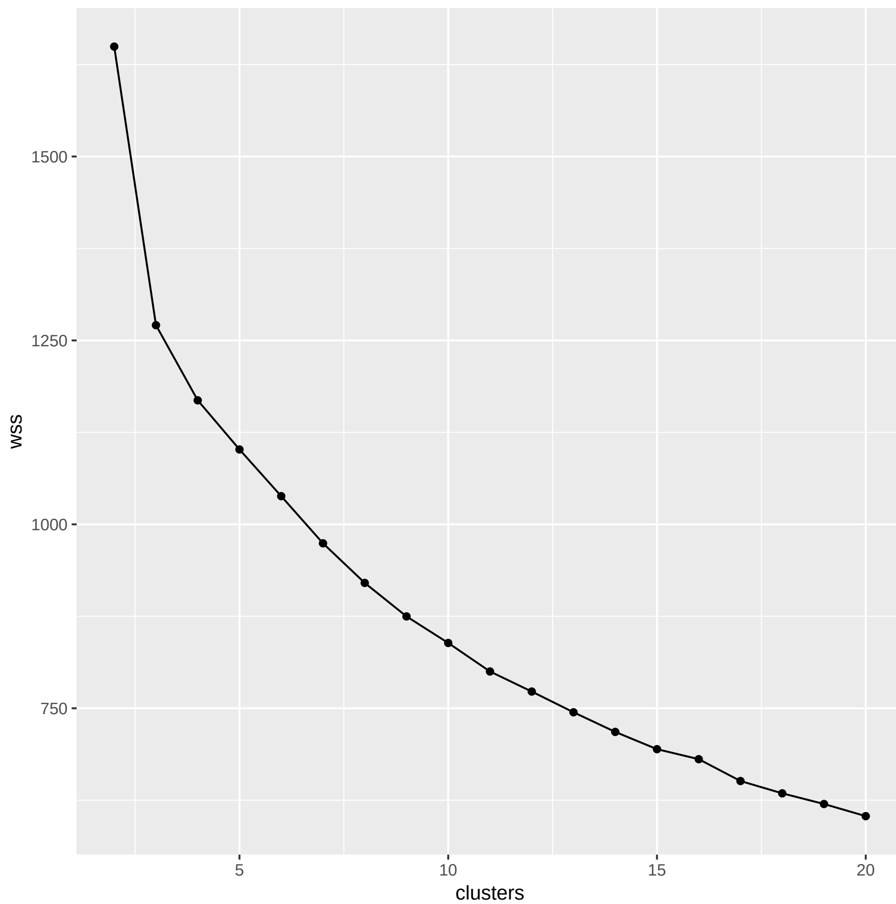


Figure 21: Wine data scree plot

```
skiers = read_delim("ski.txt", " ")

## Parsed with column specification:
## cols(
##   skier = col_character(),
##   cost = col_double(),
##   lift = col_double(),
##   depth = col_double(),
##   powder = col_double()
## )

skiers

## # A tibble: 5 x 5
##   skier  cost  lift depth powder
##   <chr> <dbl> <dbl> <dbl>  <dbl>
## 1 s1       32    64    65     67
## 2 s2       61    37    62     65
## 3 s3       59    40    45     43
## 4 s4       36    62    34     35
## 5 s5       62    46    43     40
```

Figure 22: Skiers data

```
skiers.1 = skiers %>% select(-skier) %>%
    princomp(cor=T)
summary(skiers.1)

## Importance of components:
##                           Comp.1    Comp.2      Comp.3      Comp.4
## Standard deviation     1.4199666 1.3933821 0.194453866 0.066096719
## Proportion of Variance 0.5040763 0.4853785 0.009453076 0.001092194
## Cumulative Proportion  0.5040763 0.9894547 0.998907806 1.000000000
```

Figure 23: Skiers principal components

```
ggscreeplot(skiers.1)
```

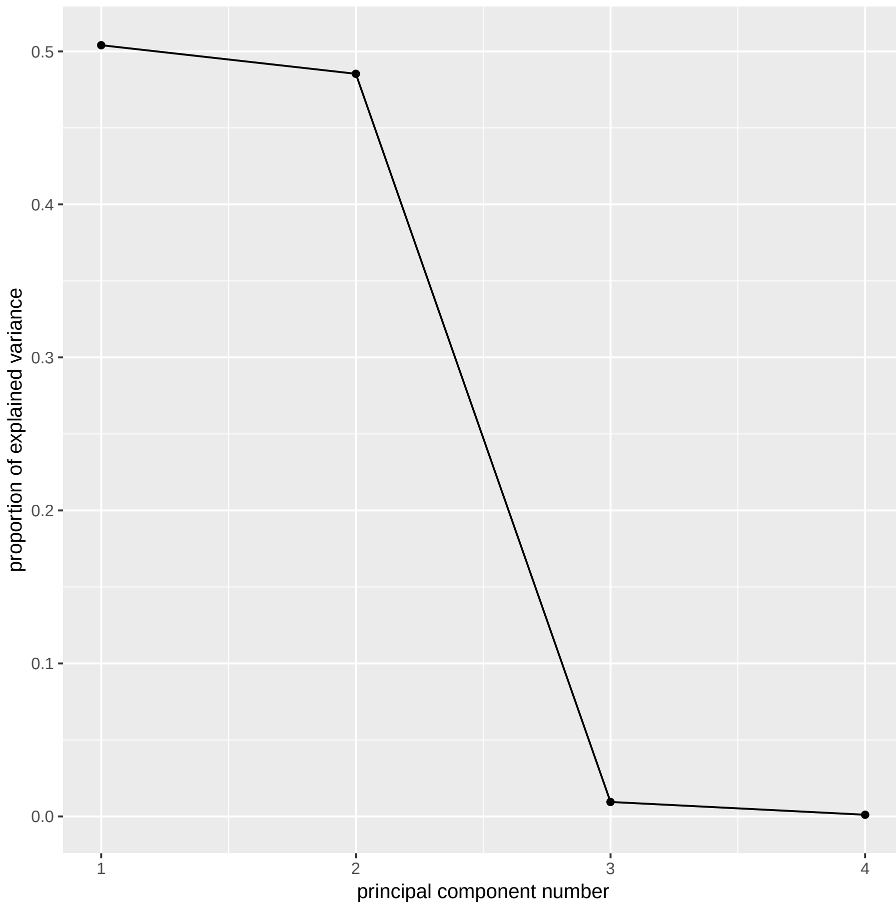


Figure 24: Skiers scree plot

```
skiers.1$loadings

##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4
## cost    0.352  0.614  0.662  0.244
## lift   -0.251 -0.664  0.676  0.199
## depth  -0.627  0.322  0.275 -0.653
## powder -0.647  0.280 -0.169  0.689
##
##                Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.00   1.00   1.00   1.00
## Proportion Var   0.25   0.25   0.25   0.25
## Cumulative Var   0.25   0.50   0.75   1.00
```

Figure 25: Skiers component loadings

```r
ggbiplot(skiers.1,labels=skiers$skier)
```

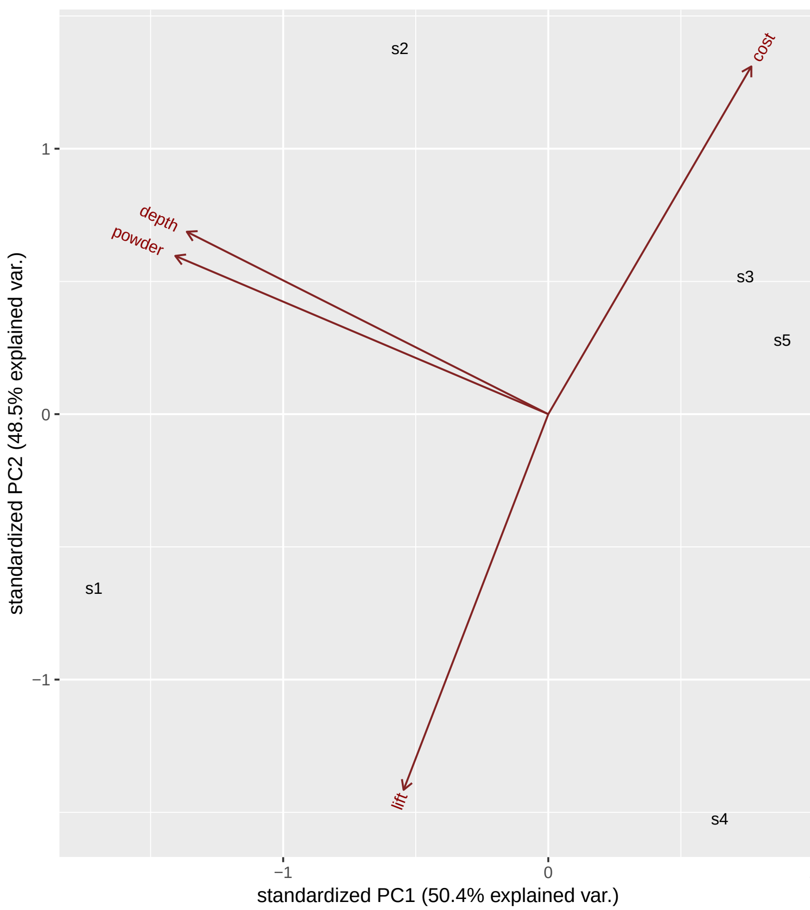


Figure 26: Skiers biplot

```r
housing %>% as_tibble()

## # A tibble: 72 x 5
##    Sat    Infl   Type      Cont   Freq
##    <ord>  <fct>  <fct>     <fct> <int>
##  1 Low    Low    Tower     Low      21
##  2 Medium Low    Tower     Low      21
##  3 High   Low    Tower     Low      28
##  4 Low    Medium Tower     Low      34
##  5 Medium Medium Tower     Low      22
##  6 High   Medium Tower     Low      36
##  7 Low    High   Tower     Low      10
##  8 Medium High   Tower     Low      11
##  9 High   High   Tower     Low      36
## 10 Low    Low    Apartment Low      61
## # ... with 62 more rows
```

Figure 27: Housing data

```
## Single term deletions
##
## Model:
## Freq ~ Sat * Infl * Type * Cont
##                    Df Deviance    AIC    LRT Pr(>Chi)
## <none>                  0.0000 484.97
## Sat:Infl:Type:Cont 12   5.9443 466.91 5.9443   0.9189
```

(intervening steps not shown)

```
housing.5=update(housing.4,.~.-Sat:Type:Cont)
drop1(housing.5,test="Chisq")
```

```
## Single term deletions
##
## Model:
## Freq ~ Sat + Infl + Type + Cont + Sat:Infl + Sat:Type + Infl:Type +
##     Sat:Cont + Infl:Cont + Type:Cont + Sat:Infl:Type
##               Df Deviance    AIC    LRT  Pr(>Chi)
## <none>              22.132 451.10
## Sat:Cont       2   38.119 463.09 15.987 0.0003376 ***
## Infl:Cont      2   45.811 470.78 23.679 7.215e-06 ***
## Type:Cont      3   66.144 489.11 44.012 1.500e-09 ***
## Sat:Infl:Type 12   43.952 448.92 21.820 0.0395878 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 28: Housing analysis

```
xt=xtabs(Freq~Sat+Cont,data=housing)
prop.table(xt,margin=2)
```

```
##         Cont
## Sat             Low      High
##   Low    0.3674614 0.3150826
##   Medium 0.2496494 0.2768595
##   High   0.3828892 0.4080579
```

Figure 29: Housing: contact by satisfaction cross-table

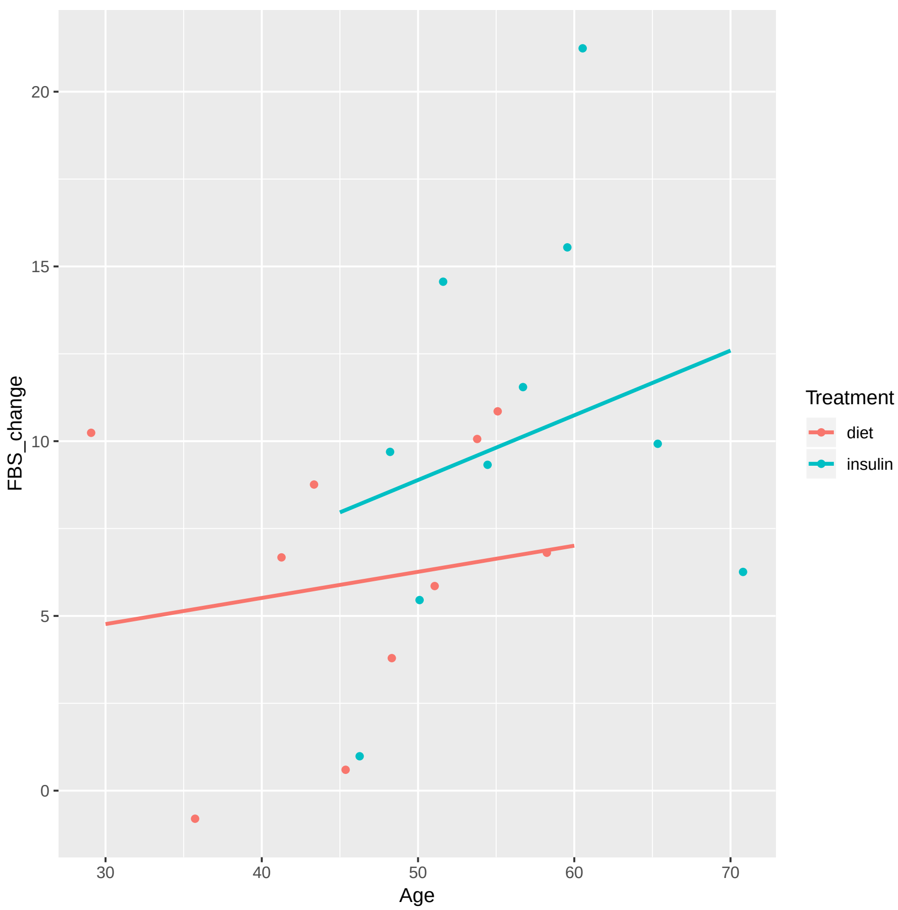
```
xt2=xtabs(Freq~Infl+Sat+Type,data=housing)
ftable(prop.table(xt2,margin=c(1,3)))

##              Type     Tower Apartment    Atrium   Terrace
## Infl   Sat
## Low    Low        0.2500000 0.5186567 0.3473684 0.6048387
##        Medium     0.2857143 0.2574627 0.3368421 0.2338710
##        High       0.4642857 0.2238806 0.3157895 0.1612903
## Medium Low        0.2965116 0.3063973 0.2142857 0.4339623
##        Medium     0.2616279 0.2693603 0.3571429 0.3207547
##        High       0.4418605 0.4242424 0.4285714 0.2452830
## High   Low        0.1477273 0.2050000 0.2166667 0.2553191
##        Medium     0.1818182 0.2150000 0.2833333 0.2340426
##        High       0.6704545 0.5800000 0.5000000 0.5106383
```

Note that `ftable` produces a compact representation of the table, without changing any of the numbers in it.

Figure 30: Housing: Satisfaction by influence by type cross-table

```
ggplot(diabetes,aes(x=Age,y=FBS_change,colour=Treatment))+
    geom_jitter()+geom_smooth(method="lm",se=F)
```



Note: I have "jittered" the points so that they don't overplot each other and you can see them all.

Figure 31: Plot of diabetes data

```
wines.3 = wines2 %>% select(-id) %>%
    kmeans(3,nstart=20)
cf=factor(wines.3$cluster)
wines.4=lda(cf~alcohol+malic_acid+ash+ash_alkalinity+
                magnesium+phenols+flavonoids+nonf_phenols+
                proanthocyanins+colour+hue+od280+proline,
            data=wines2)
wines.5=predict(wines.4)
ggbiplot(wines.4,groups=cf)
```

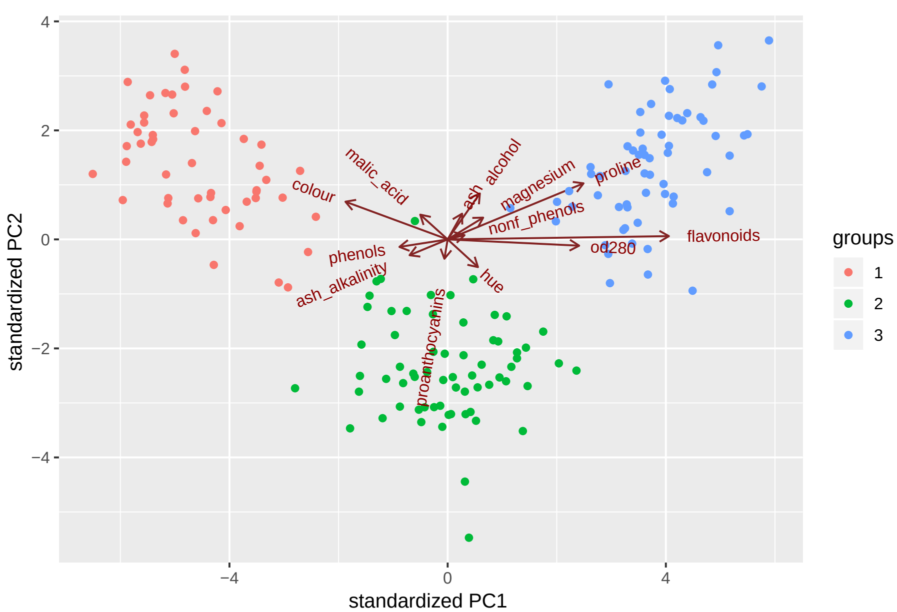


Figure 32: Biplot of wines

```
rootstocks.3$x %>% as_tibble() %>%
    mutate(obs=factor(rootstocks$rootstock),
           row=rootstocks$row) %>%
    ggplot(aes(x=LD1,y=LD2,colour=obs,label=row))+
    geom_point()+geom_text_repel()
```

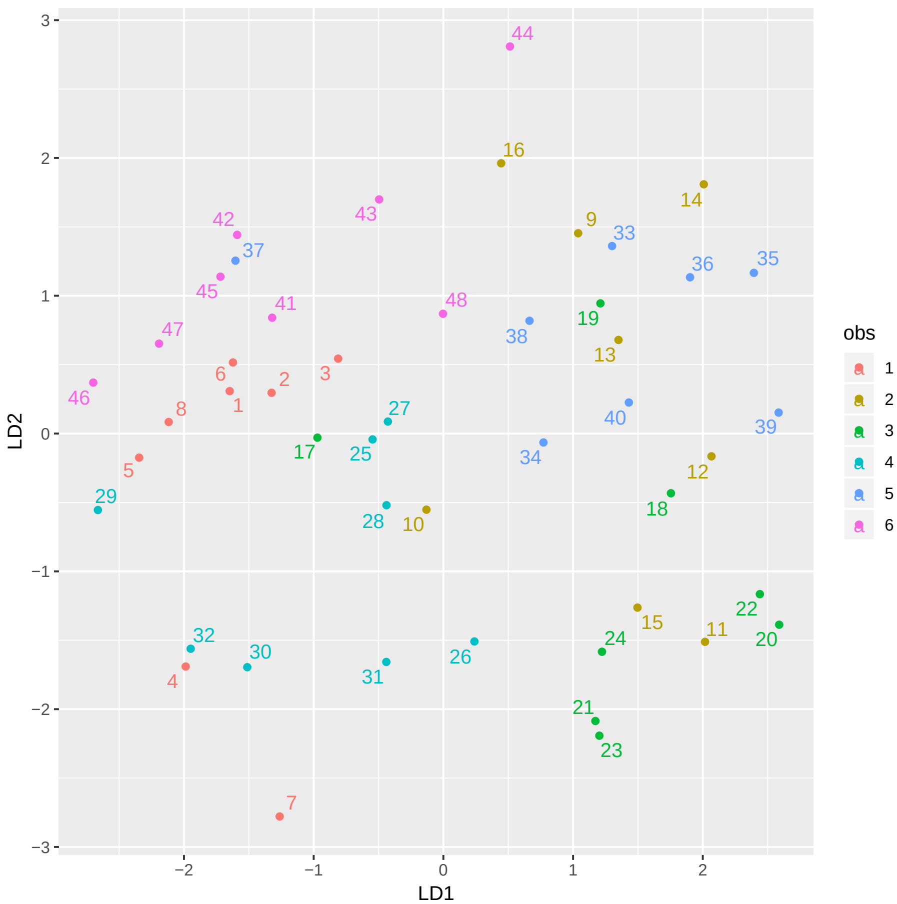


Figure 33: Rootstock discriminant scores plot