Unsupervised learning: PCA & CA

Contents

Introduction

In this practical, we will learn how to use principal components analysis and correspondence analysis.

We will use the package `ca`. For this, you will probably need to `install.packages("ca")` before running the `library()` functions.

```
library(ISLR)
library(tidyverse)
library(ca)
```

Principal components analysis

---

1. **Load the questionnaire dataset and explore it.**

```
ques_df <- read_csv("data/questionnaire.csv")

ques_df
```

```
## # A tibble: 529 x 21
##    BIGF1 BIGF3 BIGF4 BIGFe5 BIGF7 BIGF10 BIGF11 BIGF13 BIGF14 BIGF15
##    <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1     5     4     5      4     5      5      4      5      5      4
##  2     4     4     1      4     4      5      4      5      4      2
##  3     1     5     1      5     5      5      5      5      5      5
##  4     2     4     4      2     4      3      3      4      4      2
##  5     4     2     2      2     4      4      3      2      4      2
```

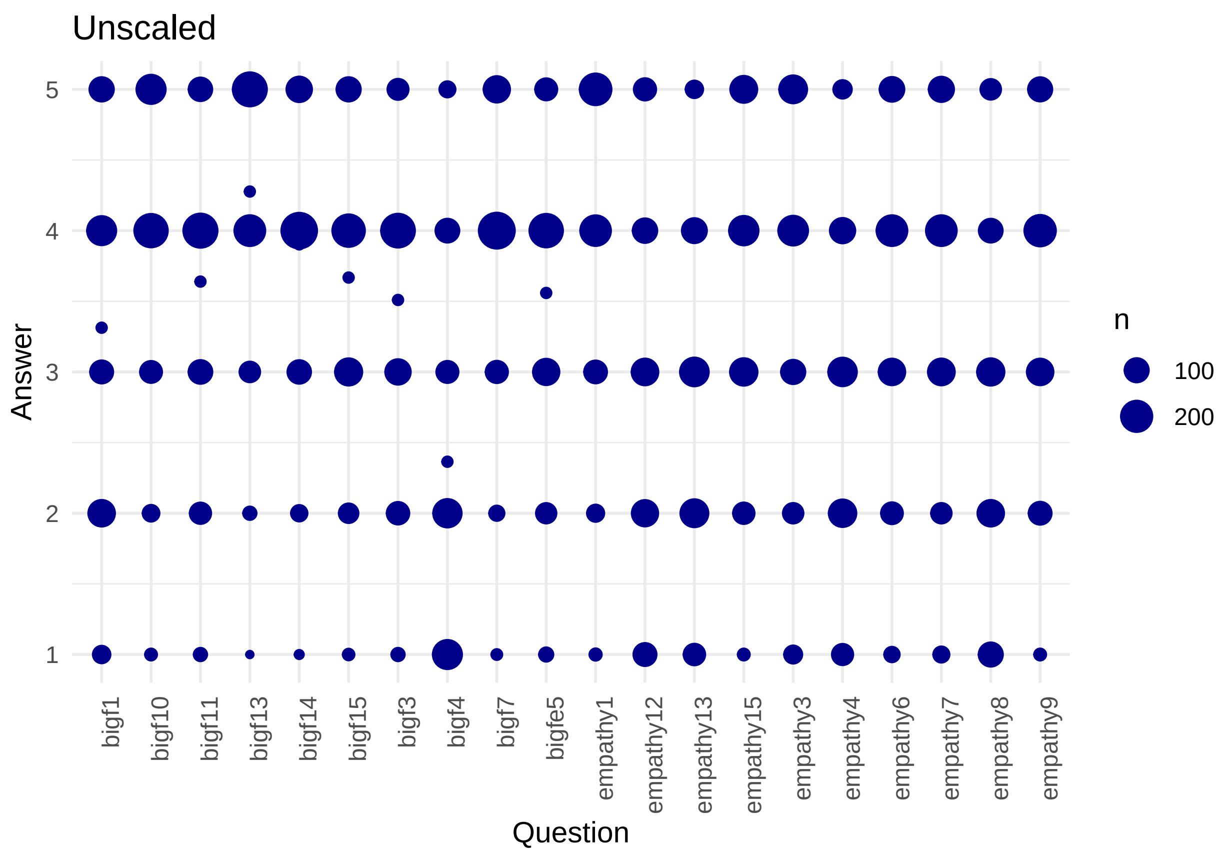
```
## 6     2    4    4    3    5    3    3    5    3    3
## 7     4    2    2    4    4    5    4    4    4    4
## 8     2    2    3    4    4    4    4    5    5    5
## 9     2    2    2    2    4    4    4    5    4    3
## 10    2    2    3    2    4    2    3    4    4    3
## # ... with 519 more rows, and 11 more variables: EMPATHY1 <int>,
## #   EMPATHY3 <int>, EMPATHY4 <int>, EMPATHY6 <int>, EMPATHY7 <int>,
## #   EMPATHY8 <int>, EMPATHY9 <int>, EMPATHY12 <int>, EMPATHY13 <int>,
## #   EMPATHY15 <int>, sex <chr>
```

```r
# a lot of likert scale variables and then one sex variable
```

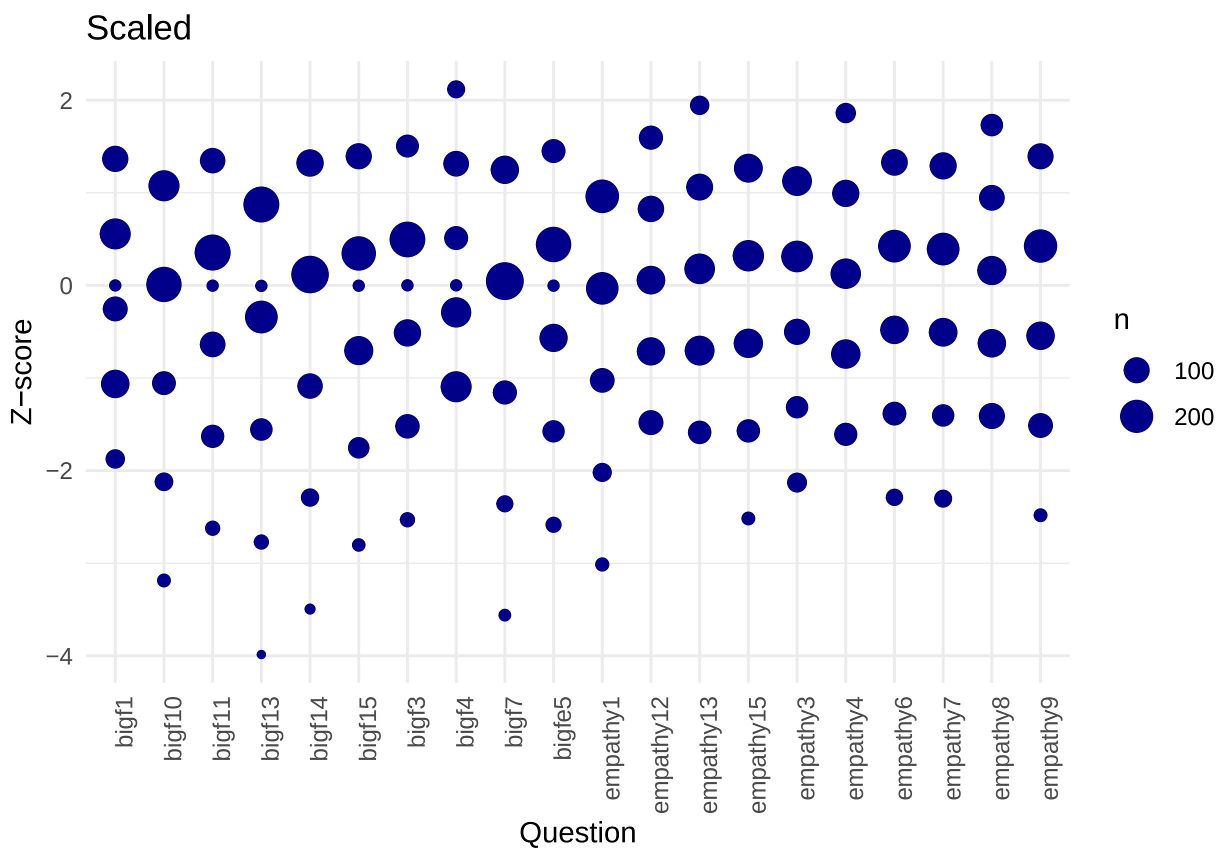
---

2. **Create a data frame with only the questionnaire columns, and standardise the dataset**

```r
ques_scaled <-
  ques_df %>%
  select(-sex) %>%
  scale() %>%
  as_tibble()

# optionally, we can also compare datasets to see what happened.
bubble_plot <- function(df) {
  df %>%
    gather(key = Question, value = Answer) %>%
    mutate(Question = str_to_lower(Question)) %>%
    ggplot(aes(x = Question, y = Answer)) +
    geom_count(colour = "#00008B") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
}

# Unscaled
ques_df %>%
  select(-sex) %>%
  bubble_plot() +
  ggtitle("Unscaled")
```




```r
# Scaled
ques_scaled %>%
  bubble_plot() +
  labs(y = "Z-score",
       title = "Scaled")
```



---

3. **Use the `prcomp()` function to create a principal components analysis for the scaled dataset. Save the result as `pca_mod`.**

```
# use prcomp here?
pca_mod <- prcomp(ques_scaled)
```

---

4. **Are the first two principal components successful in explaining variance in the dataset? How many components do we need to explain 50% of the variation in the dataset?**

```
summary(pca_mod)
```

```
## Importance of components:
##                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.990 1.7203 1.2231 1.13377 1.08634 1.01829 0.93155
## Proportion of Variance 0.198 0.1480 0.0748 0.06427 0.05901 0.05185 0.04339
## Cumulative Proportion  0.198 0.3459 0.4207 0.48501 0.54402 0.59586 0.63925
##                            PC8     PC9    PC10    PC11    PC12    PC13
```

```
## Standard deviation     0.88987 0.8843 0.87713 0.80285 0.80101 0.74269
## Proportion of Variance 0.03959 0.0391 0.03847 0.03223 0.03208 0.02758
## Cumulative Proportion  0.67884 0.7179 0.75641 0.78863 0.82072 0.84829
##                           PC14    PC15    PC16    PC17    PC18    PC19
## Standard deviation     0.72226 0.69182 0.6885 0.66436 0.6418 0.59656
## Proportion of Variance 0.02608 0.02393 0.0237 0.02207 0.0206 0.01779
## Cumulative Proportion  0.87438 0.89831 0.9220 0.94408 0.9647 0.98247
##                           PC20
## Standard deviation     0.59211
## Proportion of Variance 0.01753
## Cumulative Proportion  1.00000
```

```
# Together, the first two components explain 35% of the variance in the dataset.
# We would need 5 components to explain 50% of the variance in the dataset.
```

---

5. **Which original variable is most related to the first principal component? Which is the least relevant for the first principal component?**

```
loadings_pc1 <- pca_mod$rotation[, 1]

loadings_pc1[which.max(abs(loadings_pc1))]
```

```
##    BIGF13
## 0.3182789
```

```
# BIGF13 is the most related.

loadings_pc1[which.min(abs(loadings_pc1))]
```

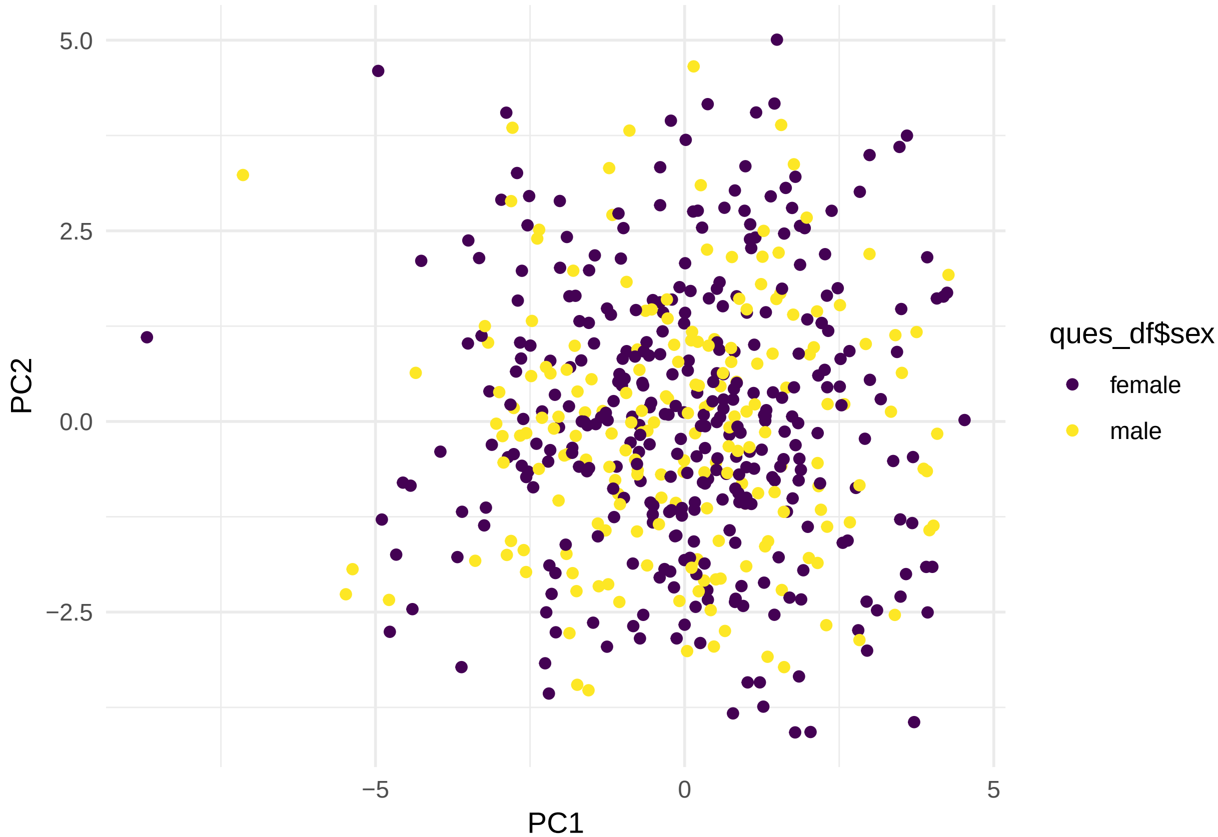
```
##    EMPATHY4
## -0.02768959
```

```
# EMPATHY4 is the least related.
```

---

6. **Create a scatter plot of the first two principal components. Map the sex of the respondents to the `colour` aesthetic. Is there a sex difference?**

```
as_tibble(pca_mod$x) %>%
  ggplot(aes(x = PC1, y = PC2, colour = ques_df$sex)) +
  geom_point() +
  theme_minimal() +
  scale_colour_viridis_d()
```




```
# There does not seem to be a sex difference in the first two components
```

Correspondence analysis

We've preprocessed a dataset from kaggle on song lyrics for the purpose of this practical. You can find the original dataset here or in the `data/` directory. If you want to know which preprocessing steps have been used and how it has been saved, you can take a look at the file `data/song_data_preproc.R`.

The `songs_ca` dataset is stored as a `.RData` file, a native file format from `R` which efficiently stores any `R` object. The `load()` function immediately loads the dataset `songs_ca` into your environment.

---

7. **Load the preprocessed `songs_ca` dataset into the environment from the `data/songs_ca.RData` file.**

```
load("data/songs_ca.RData")
```

---

8. **Use the `ca()` function from the `ca` package to create a correspondence analysis object.**

```r
ca_mod <- ca(songs_ca)
```

9. **Use the `summary()` function on this object. What can you conclude about the first two inertias? What can you say about the word "love" in this dataset?**

```r
summary(ca_mod)
```

```
##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.050916  34.6  34.6  *********
##  2      0.037005  25.2  59.8  ******
##  3      0.021217  14.4  74.2  ****
##  4      0.013919   9.5  83.7  **
##  5      0.008329   5.7  89.3  *
##  6      0.006925   4.7  94.1  *
##  7      0.003570   2.4  96.5  *
##  8      0.002149   1.5  97.9
##  9      0.001293   0.9  98.8
##  10     0.001119   0.8  99.6
##  11     0.000614   0.4 100.0
##         -------- -----
##  Total: 0.147056 100.0
##
##
## Rows:
##      name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
## 1  | CtSt |   76  175   37 |   -4   0   0 | -113 175  26 |
## 2  |  Exo |   49  972  288 | -820 775 645 |  414 198 226 |
## 3  | SnpD |  155  252   71 |   61  55  11 | -115 198  56 |
## 4  | Vngl |   44  385  123 | -237 136  48 | -320 249 122 |
## 5  | TmMG |   78  290   24 |    8   1   0 | -114 289  27 |
## 6  | IggP |   61  156   27 |   63  62   5 |  -77  94  10 |
## 7  | BryW |   73  410   40 | -107 141  16 | -148 269  43 |
## 8  | GrtD |   57   55   35 |   68  51   5 |  -20   4   1 |
## 9  | FlRd |  149  397   84 |  171 352  86 |   62  46  15 |
## 10 | JhnM |   95  188   49 | -118 185  26 |  -17   4   1 |
## 11 | JnsJ |  109  845  200 |  268 266 154 |  395 579 461 |
## 12 | Extr |   53  200   20 |  -53  49   3 |  -92 151  12 |
##
```

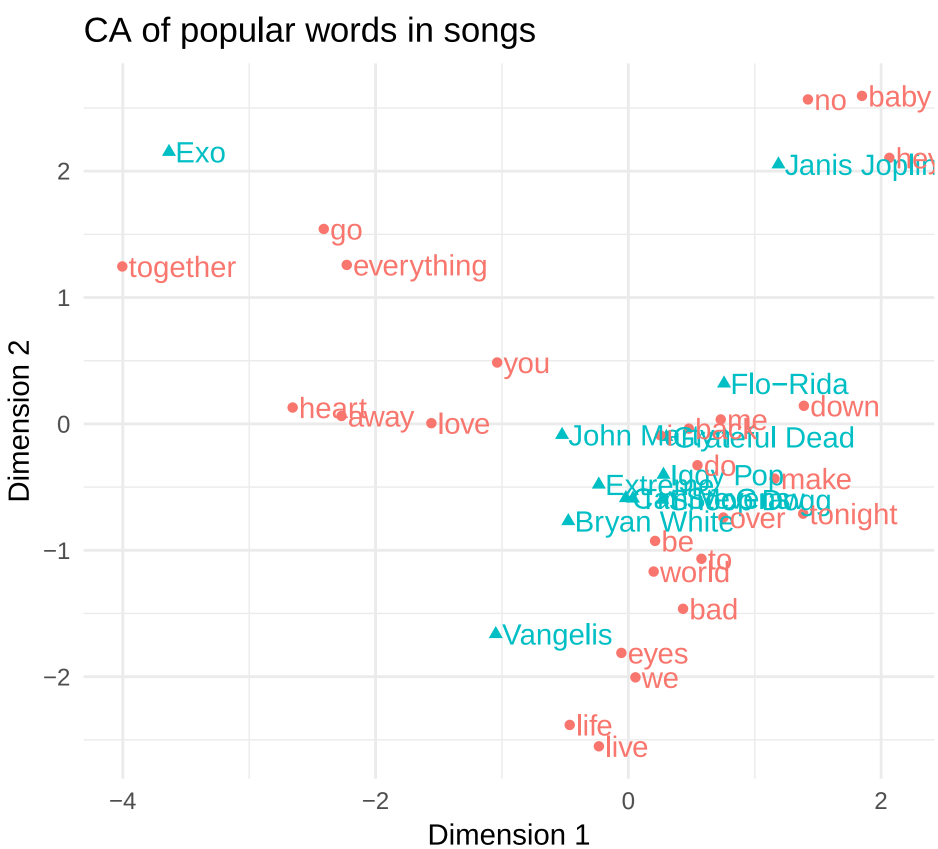
```
## Columns:
##        name   mass  qlt  inr    k=1  cor ctr    k=2  cor ctr
## 1   |     i |  258  203   31 |   57  186  16 |  -17   17   2 |
## 2   | love  |   39  475   69 | -352  475  94 |    1    0   0 |
## 3   |  you  |  206  956   93 | -234  825 223 |   93  131  49 |
## 4   |   me  |   84  494   32 |  165  493  45 |    7    1   0 |
## 5   |   we  |   33  508   67 |   13    1   0 | -386  507 135 |
## 6   |   to  |  113  667   68 |  131  192  38 | -205  475 129 |
## 7   |   be  |   36  529   16 |   48   35   2 | -178  494  31 |
## 8   |   do  |   22  250   11 |  123  198   7 |  -63   52   2 |
## 9   |   go  |   22  830   70 | -544  640 130 |  297  190  53 |
## 10  |   no  |   35  852   97 |  321  253  71 |  494  599 231 |
## 11  | baby  |   23  799   84 |  417  329  80 |  499  471 157 |
## 12  | hert  |    8  668   31 | -599  666  59 |   25    1   0 |
## 13  | life  |   12  350   52 | -105   17   3 | -458  333  69 |
## 14  | down  |   21  395   36 |  313  392  41 |   28    3   0 |
## 15  | wrld  |    8  245   11 |   45   10   0 | -225  235  11 |
## 16  | over  |    5  159   11 |  170   93   3 | -143   66   3 |
## 17  | eyes  |    5  205   21 |  -12    0   0 | -349  205  17 |
## 18  |  bad  |    3  218    9 |   98   23   1 | -281  195   7 |
## 19  | away  |   10  590   29 | -512  590  49 |   12    0   0 |
## 20  | tgth  |    2  803   18 | -904  750  38 |  240   53   4 |
## 21  | tngh  |    3  132   21 |  312  111   7 | -137   21   2 |
## 22  | evry  |    3  623   12 | -503  506  17 |  242  117   5 |
## 23  | live  |    5  348   24 |  -53    4   0 | -491  344  32 |
## 24  | make  |   13  334   20 |  261  303  17 |  -83   31   2 |
## 25  | back  |   15   62   19 |  108   61   3 |   -7    0   0 |
## 26  |  hey  |   13  686   50 |  466  391  57 |  405  295  59 |
```

---

10. **Recreate using `ggplot` the biplot that results from the `plot()` method on this object. Hint: for this, you can use the `rowcoord` and `colcoord` elements of the object.**

```r
gg_ca <-
  rbind(ca_mod$rowcoord[, 1:2], ca_mod$colcoord[, 1:2]) %>%
  as_tibble() %>%
  mutate(name = c(rownames(songs_ca), colnames(songs_ca)),
         type = c(rep("row", 12), rep("col", 26)))

gg_ca %>%
  ggplot(aes(x = Dim1, y = Dim2, shape = type, colour = type, label = name)) +
  geom_point() +
  geom_text(hjust = 0, nudge_x = 0.05) +
  coord_fixed() +
```

```r
  theme_minimal() +
  theme(legend.position = "none") +
  labs(x = "Dimension 1", y = "Dimension 2",
       title = "CA of popular words in songs")
```



---

11. **What can you conclude about Exo and Janis Joplin? Can you come up with reasonable explanations for this?**

```
# Exo is a k-pop band. Janis Joplin was born in 1943 (and a member of the
# infamous 27 club), so her lyrics are from a different period than the other
# artists.
```

---

12. **In which ways would the plot be different if we would use different artists?**

```
# It would look completely different: correspondence analysis maps
# rows and columns together. So the words will be in different places.
# For example, if the artist "high school musical" would be in the sample,
```

```
# it would separate itself from the other artists because the word "together"
# is used a lot. Consequently, the word "together" would appear near the
# artist. The same will happen with Cole Porter and "love".
```

Final assignment: High-dimensional PCA using SVD

This is an advanced assignment. You will have to figure out how to create PC scores from the output of a singular value decomposition.

Principal components analysis can also be used to generate a low-dimensional number of features from a high-dimensional (p > n) dataset. One area where high-dimensional data frequently occurs is in chemometrics, assessing the properties of materials using spectroscopy (Wikipedia link).

---

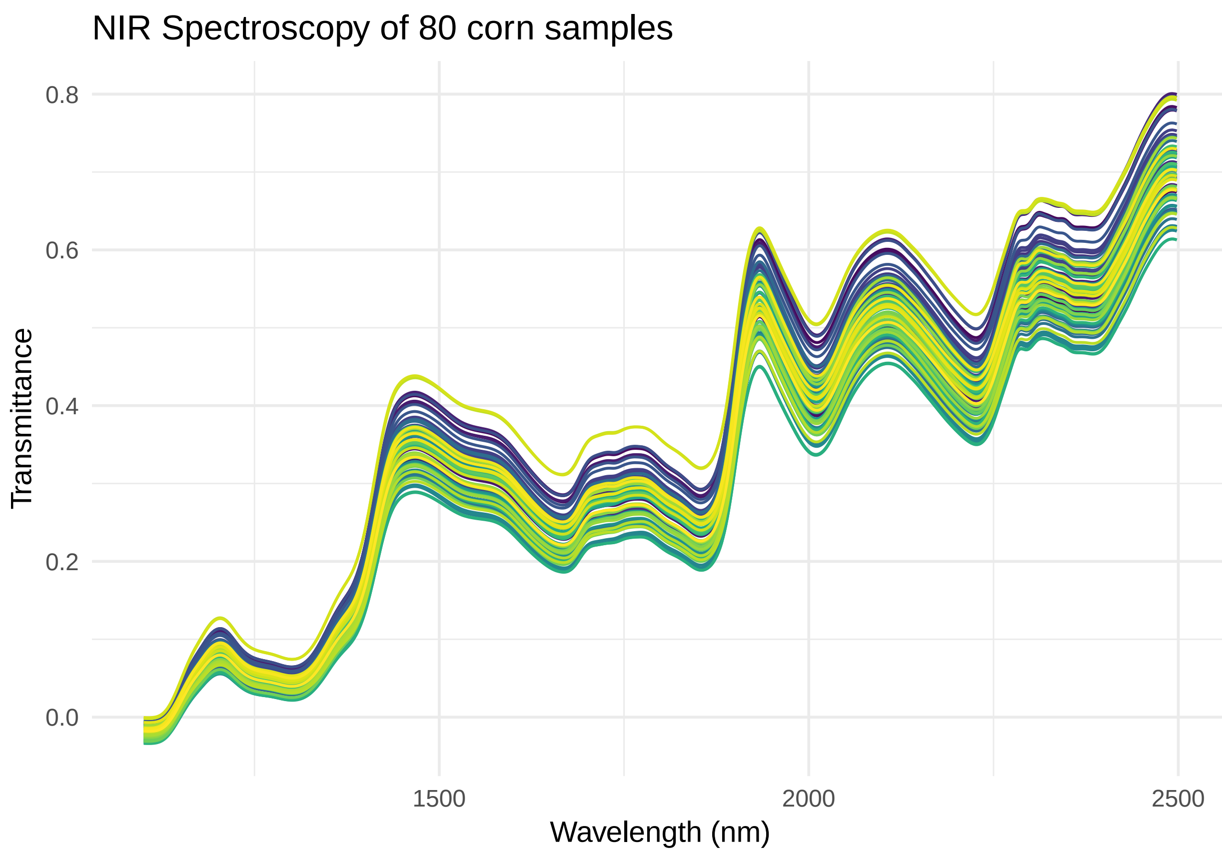
13. **Load the dataset "data/corn.RData" using the function `load()`.**

```
# Load the data. Source: http://www.eigenvector.com/data/Corn/index.html
# converted from matlab to an R data object.
load("data/corn.RData")
```

The first four columns contain properties of the corn samples (80 corn samples were analysed) and the remaining 700 columns indicate the measured transmittance at different near-infrared wavelengths. You can find more information about this dataset at the source website.

Here is a plot of wavelength versus transmittance, with one line for each of the 80 corn samples:

```
t(corn[, -c(1:4)]) %>%
  as_tibble %>%
  gather(key = corn, value = signal) %>%
  mutate(wavelength = rep(seq(1100, 2498, 2), 80)) %>%
  ggplot(aes(x = wavelength, y = signal, colour = corn)) +
  geom_line() +
  theme_minimal() +
  scale_colour_viridis_d(guide = "none") +
  labs(x = "Wavelength (nm)",
       y = "Transmittance",
       title = "NIR Spectroscopy of 80 corn samples")
```

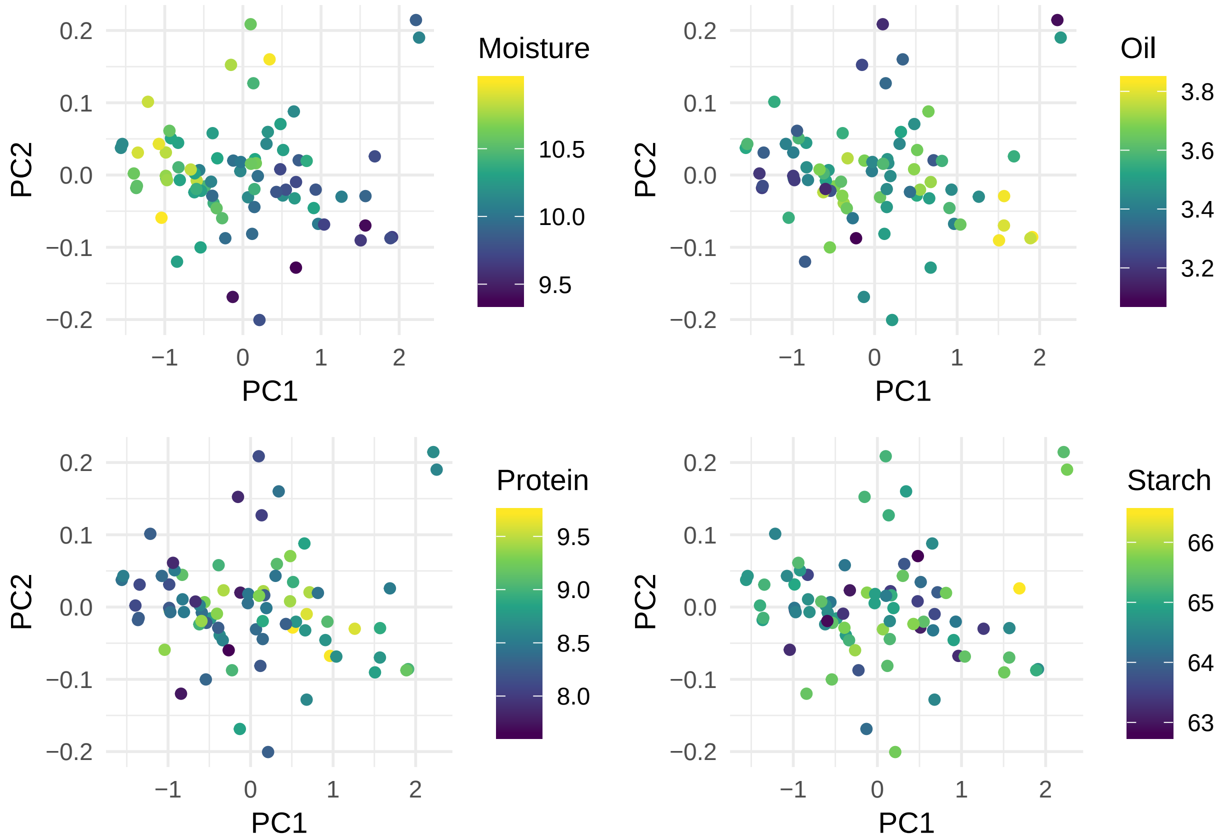


---

14. **Use the `svd()` function to run a principal components analysis on the spectroscopy part of the corn dataset. Save the PC scores and plot the first two principal components. Create four plots, each mapping one of the four properties to the `colour` aesthetic. Which of the properties relate most to the first two principal components? Base your answer on the plots only. Then, do the same thing for PCs 5 and 6.**

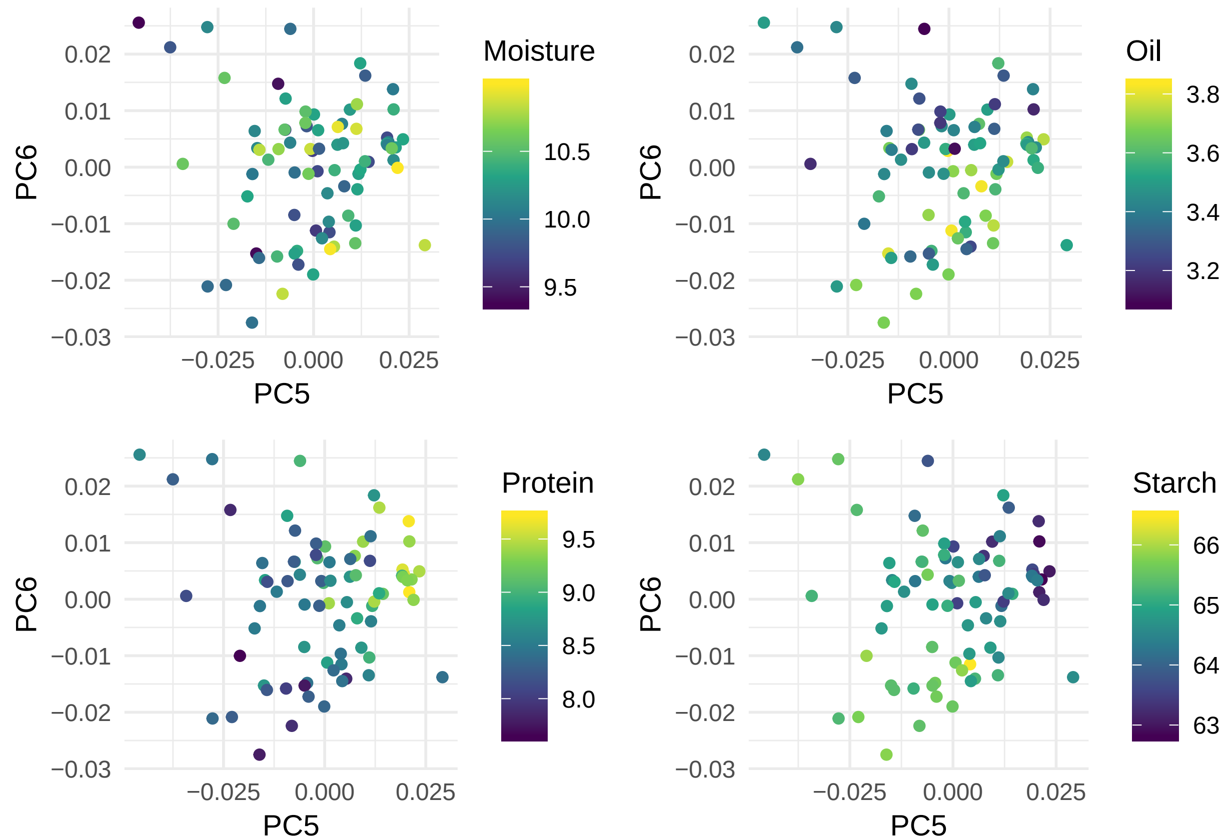
```r
# see https://stats.stackexchange.com/a/134283 for information.

# Perform svd-pca
x_scaled  <- scale(corn[, -c(1:4)], scale = FALSE)
svd_corn  <- svd(x_scaled)
pc_scores <- svd_corn$u %*% diag(svd_corn$d)

# Plot first two principal components versus the corn properties
ggcorn <-
  corn[, 1:4] %>%
  bind_cols(tibble(PC1 = pc_scores[, 1],
                   PC2 = pc_scores[, 2],
                   PC5 = pc_scores[, 5],
```

```
                   PC6 = pc_scores[, 6]))

cowplot::plot_grid(
  ggcorn %>%
    ggplot(aes(x = PC1, y = PC2, colour = Moisture)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c(),
  ggcorn %>%
    ggplot(aes(x = PC1, y = PC2, colour = Oil)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c(),
  ggcorn %>%
    ggplot(aes(x = PC1, y = PC2, colour = Protein)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c(),
  ggcorn %>%
    ggplot(aes(x = PC1, y = PC2, colour = Starch)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c()
)
```




```
# Moisture and oil seem to have a strong relation with the first two PCs,
# and their high values are on opposite sides.
# The high/low values for protein and starch seem more randomly
# ordered in these plots.

cowplot::plot_grid(
  ggcorn %>%
    ggplot(aes(x = PC5, y = PC6, colour = Moisture)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c(),
  ggcorn %>%
    ggplot(aes(x = PC5, y = PC6, colour = Oil)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c(),
  ggcorn %>%
    ggplot(aes(x = PC5, y = PC6, colour = Protein)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c(),
```

```r
  ggcorn %>%
    ggplot(aes(x = PC5, y = PC6, colour = Starch)) +
    geom_point() +
    theme_minimal() +
    scale_colour_viridis_c()
)
```




```r
# Here, it's protein and starch that relate more to the PCs. They too are
# opposites in these samples.
```