Logboek

Kasper Notebomer

9/14/2021

Setup of the libraries.

```
library("ggplot2")
library("kableExtra")
library("factoextra")
library("cowplot")
library("gridExtra")
library("dplyr")
library(ggpubr)
library(ggfortify)
library(tidyr)
library(purrr)
library(ggcorrplot)
library(reshape2)
```

Introduction

The data was provided by Tsjerk Wassenaar of the RuG. The data set contains data on membrane composition and characteristics. The data set is not publicly available. The data set also does not have a publication linked to it.

Seeing as there is no paper linked to the data set I can only guess using the information that I did get that the data was gathered by making a membrane using specific variables and the measuring the resulting membranes to get variables like the thickness and compressibility. The data set contains 14 variables and 2843 different measurement. The goal is to use measurement to predict the composition of the membrane. Typically you would use independent variables to predict one dependent variable. In this data set this is not the case, seeing as the variables that are given as parameters are seen as class variables, this means that these would be considered the labels. The parameters are: Temperature, Sterol type, Sterol concentration, Other (phospho)lipids in membrane, Aliphatic tails, Saturation index, Phosphatidyl choline concentration and Ethanol concentration. So the biggest question that the EDA should anwser is which of these variables is most interesting to use as the label, or could we even predict multiple of them using the save variables.

Data exploration

Data reading & codebook

To start off the data is read in using read.csv(). The data frame that is created by this step is turned into a tibble and then used to create a codebook. The codebook is a csv file that contains the abbreviation, type and the description of every variable in the data set.

```
# Read the data and transform it to a tibble
data <- read.csv("data/dataFrame_all_sims.csv", sep = ",", header=TRUE, stringsAsFactors=FALSE)
data <- as_tibble(data)
```

Here, the codebook is made of all of the abbreviations in the data frame, their meaning and their type.

```
# Create the codebook
codebook <- data.frame("Abbreviation" = colnames(data), "Type" = sapply(data, typeof),
                       "Class" = sapply(data, class),
                       "Desciption" = c("Temperature (Kelvin)",
                                        "Sterol type",
                                        "Sterol concentration (%)",
                                        "Other (phospho)lipids in membrane (headgroup)",
                                        "Aliphatic tails",
                                        "Saturation index (double bonds per tail)",
                                        paste("Phosphatidyl choline concentration",
                                              "(% of non-sterol lipids)"),
                                        "Ethanol concentration (% of solvent)",
                                        "Area per lipid (nm^2)",
                                        "Thickness (nm)", "Bending rigidity (kB T)",
                                        "Tilt angle (degrees)",
                                        "Z-order", "Compressibility (cN / m)"))

codebook <- as_tibble(codebook)
kable(codebook, caption = "Codebook") %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Table 1: Codebook

Abbreviation	Type	Class	Desciption
temperature	integer	integer	Temperature (Kelvin)
sterol.type	character	character	Sterol type
sterol.conc	integer	integer	Sterol concentration (%)
other.phosph	character	character	Other (phospho)lipids in membrane (headgroup)
tails	character	character	Aliphatic tails
satur.index	double	numeric	Saturation index (double bonds per tail)
PC.conc	integer	integer	Phosphatidyl choline concentration (% of non-sterol lipids)
ethanol.conc	integer	integer	Ethanol concentration (% of solvent)
APL	double	numeric	Area per lipid (nm^2)
thickness	double	numeric	Thickness (nm)
bending	double	numeric	Bending rigidity (kB T)
tilt	double	numeric	Tilt angle (degrees)
zorder	double	numeric	Z-order
compress	double	numeric	Compressibility (cN / m)

```
# Write codebook to .csv file
write.csv(codebook, "data/codebook.csv", row.names = FALSE)
```

To easily access the description based on the abbreviation the following function is used.

```r
# Get the description by the abbreviation from the codebook
get_des_by_ab <- function(df, abbreviation){
  description <- df[df$Abbreviation == abbreviation,]$Desciption
  return(description)
}
```

Structure & summary

Here I'll take a look at the structure and the five number summary of the data to look for any problems or discrepancies.

```r
# Print the structure of the data
#str(data)
data.frame(variable = names(data),
           classe = sapply(data, class),
           first_values = sapply(data, function(x) paste0(head(x),  collapse = ", ")),
           row.names = NULL) %>%
  kable(caption="Data structure")  %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Table 2: Data structure

variable	classe	first_values
temperature	integer	298, 298, 298, 298, 298, 298
sterol.type	character	chole, chole, chole, chole, chole, chole
sterol.conc	integer	20, 20, 20, 20, 20, 20
other.phosph	character	PE, PE, PE, PE, PE, PE
tails	character	PI, PI, PI, PI, PI, PI
satur.index	numeric	1, 1, 1, 1, 1, 1
PC.conc	integer	33, 50, 25, 100, 0, 75
ethanol.conc	integer	20, 20, 20, 20, 20, 20
APL	numeric	0.736911, 0.746226, 0.731945, 0.76963, 0.719509, 0.760091
thickness	numeric	3.62897268631964, 3.59699140829278, 3.6526367497855, 3.49605162681907, 3.71001820243157, 3.53890823219068
bending	numeric	10.0802, 10.1403, 10.2547, 10.0232, 10.4654, 9.86163
tilt	numeric	14.7578, 14.3075, 14.9777, 13.2029, 15.753, 13.7148
zorder	numeric	0.175499, 0.172084, 0.178494, 0.163349, 0.182191, 0.166807
compress	numeric	24.1821842679417, 24.3249489613324, 23.766069963793, 22.9607685634682, 23.1050322963544, 23.8604998172132

Looking at the structure of the data, it seems like all of the columns have been read and have the right datatype.

```r
# Get the five number summary of the selected columns
sum <- summary(data[c("temperature","sterol.conc","satur.index",
                      "PC.conc","ethanol.conc","APL", "thickness", "bending",
                      "tilt", "zorder","compress")])

sum <- sub(".*:", "", sum)
sum[is.na(sum)] <- 0
rownames(sum) <- c("Minimum", "Q1", "Median", "Mean", "Q3", "Maximum", "Number of NA's")

# Print five number summary using kable
kable(sum, caption="Five number summary") %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Looking at the structure there doesn't seem to be immediate problem with the data, like the data being read as a wrong type of data. When looking at the five number summary there do seem to be some rows that are

Table 3: Five number summary

	temperature	sterol.conc	satur.index	PC.conc	ethanol.conc	APL	thickness	bending	tilt	zorder	compress
Minimum	298.0	0.00	0.0000	0.00	0	0.4514	3.003	0.2025	6.774	0.02736	1.286
Q1	298.0	10.00	0.0000	25.00	5	0.6350	3.627	9.8579	14.490	0.18604	23.097
Median	298.0	20.00	0.5000	50.00	15	0.6885	3.812	12.1000	18.157	0.25860	31.346
Mean	305.2	16.55	0.6208	50.02	15	0.6807	3.934	22.0256	35.732	0.33269	48.714
Q3	298.0	30.00	1.0000	75.00	25	0.7394	4.238	25.5712	35.933	0.45559	43.348
Maximum	328.0	30.00	2.0000	100.00	30	0.8924	5.101	125.3820	227.813	0.93013	538.446
Number of NA's	0	0	0	0	0	104	103	104	104	153	103

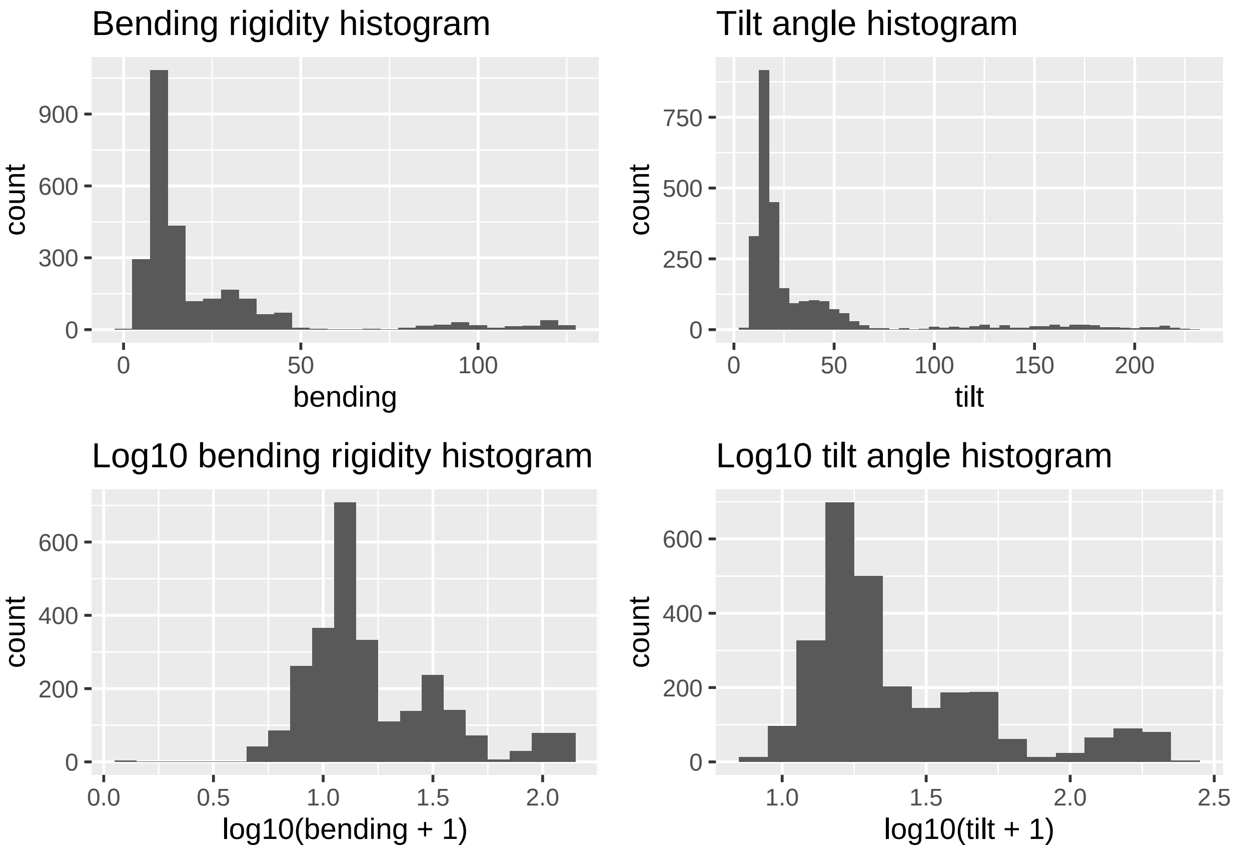
missing values like the APL, thickness, bending and tilt values. These rows will be omitted for plotting and other research seeing as they can't be used. The maximum of the bending rigidity and tilt angle seem to be too high, this probably means that these are outliers seeing as the rest of the values in the summary seem to be believable. To check this we'll make a histogram of these variables.

```
# Remove NA's
row.has.na <- apply(data, 1, function(x){any(is.na(x))})
nona_data <- na.omit(data)
row.has.na.2 <- apply(nona_data, 1, function(x){any(is.na(x))})
paste("Number of NA's before: ", sum(row.has.na),
", Number of NA's after:", sum(row.has.na.2), sep="")
```

```
## [1] "Number of NA's before: 153, Number of NA's after:0"
```

153 lines with NA's where omitted from the data set.

```
# Create 4 histograms
p <- ggplot(nona_data, aes(x= bending)) +
  geom_histogram(binwidth=5) +
  ggtitle("Bending rigidity histogram")
p2 <- ggplot(nona_data, aes(x=tilt)) +
  geom_histogram(binwidth=5) +
  ggtitle("Tilt angle histogram")
p3 <- ggplot(nona_data, aes(x= log10(bending + 1))) +
  geom_histogram(binwidth=0.1) +
  ggtitle("Log10 bending rigidity histogram")
p4 <- ggplot(nona_data, aes(x=log10(tilt + 1))) +
  geom_histogram(binwidth=0.1) +
  ggtitle("Log10 tilt angle histogram")
ggarrange(p, p2, p3, p4)
```



Both the bending and the tilt variable seem to have a lot of outliers. Even when transformed using log10 they still seem to be quite skewed. This is something we need to keep in mind in future research.

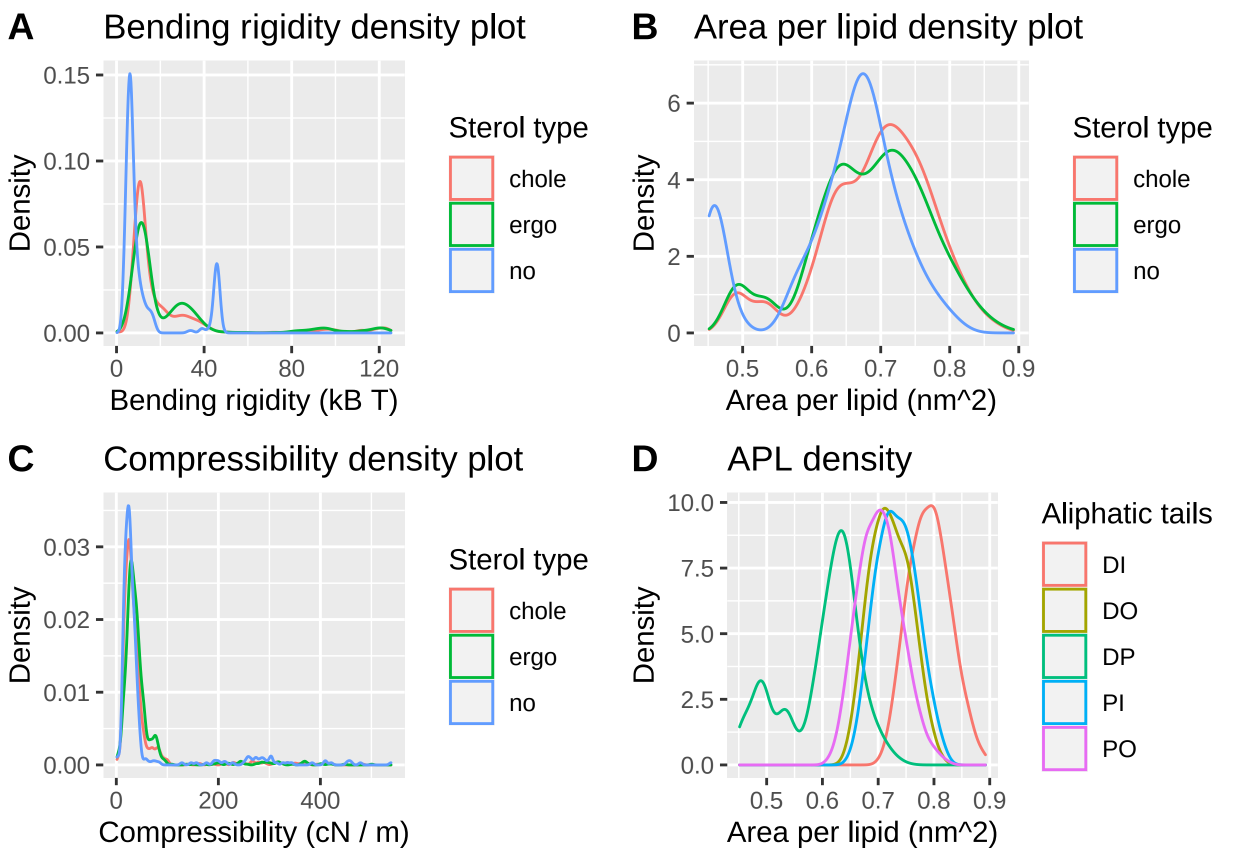
Exploring relations between variables

Density plot

I will now make density plots of some of the variables to see whether or not they look promising in finding class distinctions. These density plots will also make any skewing in the data apparent.

```
p <- ggplot(nona_data, aes(x=bending, colour = factor(sterol.type))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "bending")) +
  ylab("Density") +
  ggtitle("Bending rigidity density plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))

p2 <- ggplot(nona_data, aes(x=APL, colour = factor(sterol.type))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab("Density") +
  ggtitle("Area per lipid density plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))

p3 <- ggplot(nona_data, aes(x=compress, colour = factor(sterol.type))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "compress")) +
  ylab("Density") +
  ggtitle("Compressibility density plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))

p4 <- ggplot(nona_data, aes(x=APL, colour = factor(tails))) +
  geom_density() +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab("Density") +
  ggtitle("APL density") +
  labs(colour = get_des_by_ab(codebook, "tails"))
```

```
ggarrange(p, p2, p3, p4,
  labels = c("A", "B", "C", "D"),
  ncol = 2, nrow = 2)
```



Plot A show that there is basically no distinction between cholesterol and ergosterol based on the bending rigidity. However, distinction between no sterol or a sterol does seem to be possible based on the bending rigidity seeing as the peaks of there classes only overlaps a small bit. There does seem to be an odd peak in the no sterol class at around 50 kB/t bending rigidity.

In plot B it looks like there is absolutely no way of distinguishing different classes based on the area per lipid, seeing as the peaks are basically in the same place. But just to be certain we'll still plot in against some other variables to make sure, seeing as it might be a very useful variable when paired with something like bending rigidity. The results depicted in plot C also don't look very promising seeing as every peak is around the same place again. Plot D on the other hand seems a lot more interesting seeing as all of the peaks are in slightly different locations, the DO, PO and the PI seem to overlap a lot but the DP and DI tails seem to have little overlap with the rest. It seems like APL could possibly be used as a variable to distinguish between aliphatic tails when paired with another variable.

When looking at all of the plots it looks like most of the data is skewed in one way or another, even though when we looked at the structure there didn't seem to be much. This can probably be explained by the extra dimension that was added in these plots, the sterol type and the aliphatic tails.

Scatter plot

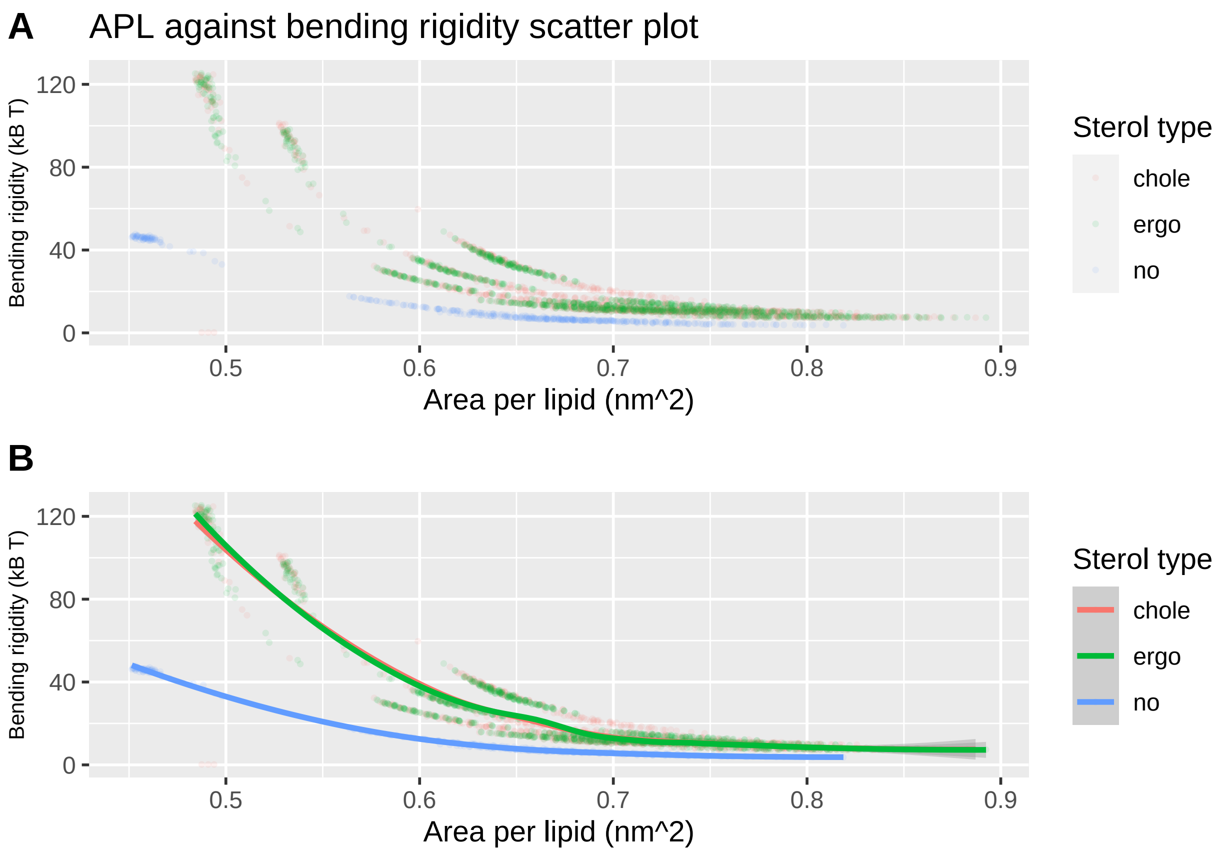
```
pairs(nona_data,
      panel = "panel.smooth",
      pch = 20,
      cex = 0.35,
      col = rgb(0.1, 0.4, 1, alpha = 0.4))
```

This code is for a paired scatter plot. The eval is currently set to false seeing as there are a lot of variables that are compared to each other in this data set. Running this code will result in a big plot that isn't very well interpret able in a pdf file.

First we'll plot the bending area per lipid against the rigidity of the membrane, coloring the points based on the sterol type, to look for any kind of clustering or correlation. To make it even cleared we'll also make a second plot with a loess regression. The reason we use loess regression instead of linear regression is because loess will show us a better trend line, seeing as it isn't trying to fit a straight line like linear regression.

```
#Plot APL against bending rigidity
chart <- ggplot(nona_data, aes(APL, bending, colour = factor(sterol.type) )) +
  geom_point(alpha=1/10, size=0.5) +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab(get_des_by_ab(codebook, "bending")) +
  theme(axis.title.y = element_text(size=8)) +
  ggtitle("APL against bending rigidity scatter plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))

ggarrange(chart,
          chart +
            geom_smooth(method = "loess") + ggtitle(""),
          labels = c("A", "B"),
          ncol = 1, nrow = 2)
```

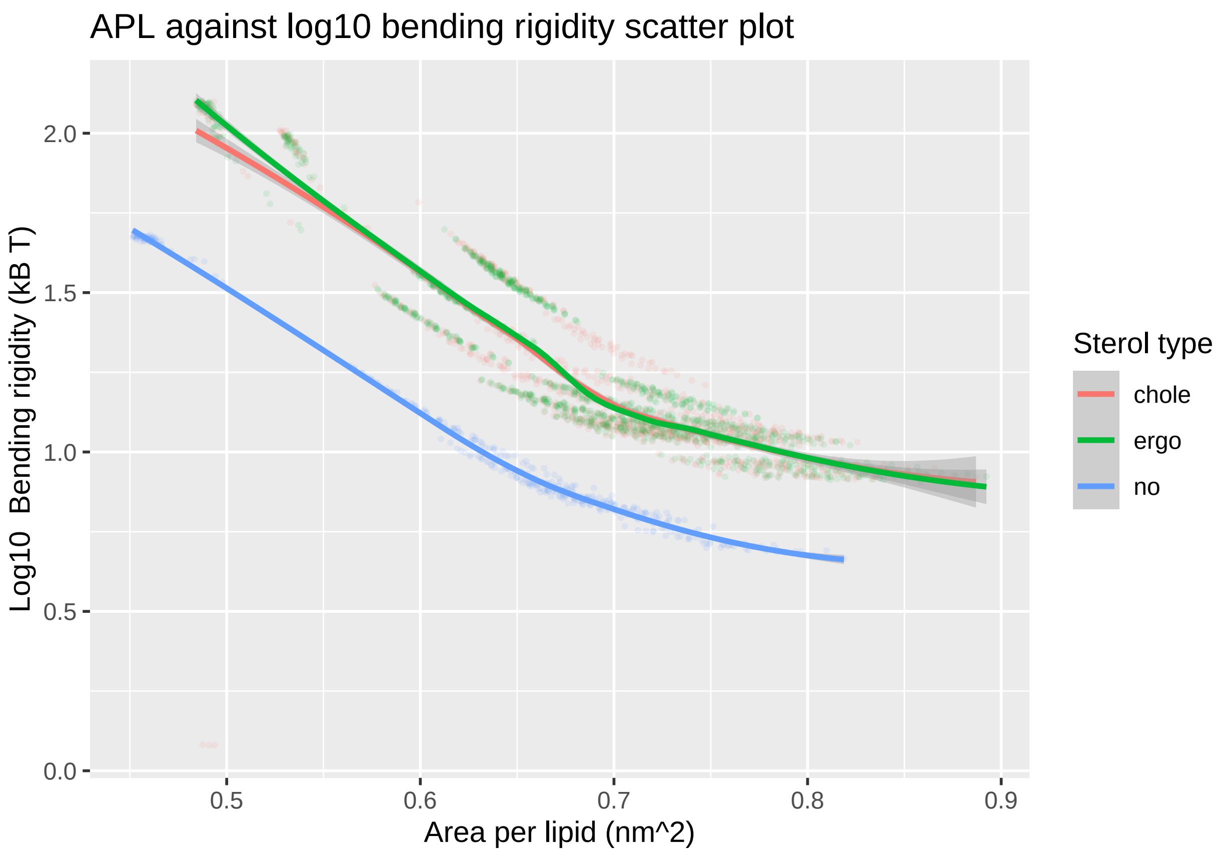
```
## `geom_smooth()` using formula 'y ~ x'
```



Looking at the plot we can see that there is a clear separation in the area per lipid to bending rigidity ratio when comparing no sterol present to both cholesterol and ergosterol. When comparing cholesterol and ergosterol we can't see such a clear separation, there seems to be a lot of overlap. There are also a few outliers that are barely visible at area per lipid 0.4 and bending rigidity 0. The correlation seem to inverse logarithmic, so to check this I'll make a plot where the bending rigidity is log transformed.

```
chart <- ggplot(nona_data, aes(APL, log10(bending + 1), colour = factor(sterol.type) )) +
  geom_point(alpha=1/10, size=0.5) +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab(paste( "Log10 ", get_des_by_ab(codebook, "bending"))) +
  geom_smooth(method = "loess") +
  ggtitle("APL against log10 bending rigidity scatter plot") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))
chart
```

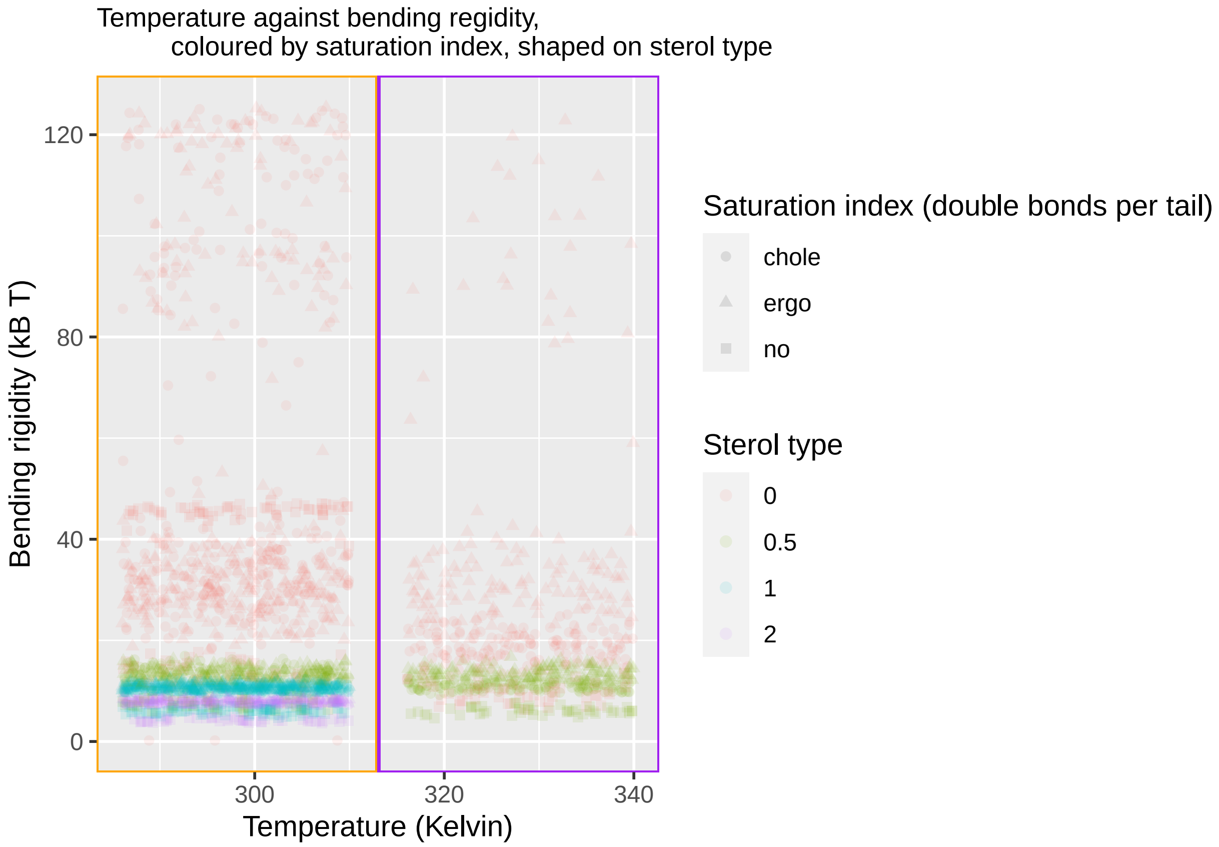
```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Looking at this plot we can see that there indeed seem to be an almost linear correlation between the log10 transformed bending rigidity and the area per lipid. There does still seem to be a sleight curve in all of the trend lines when the area per lipid increases.

Here I'll plot the temperature against the bending rigidity while coloring based on saturation index, and deciding the shape based on the sterol type.

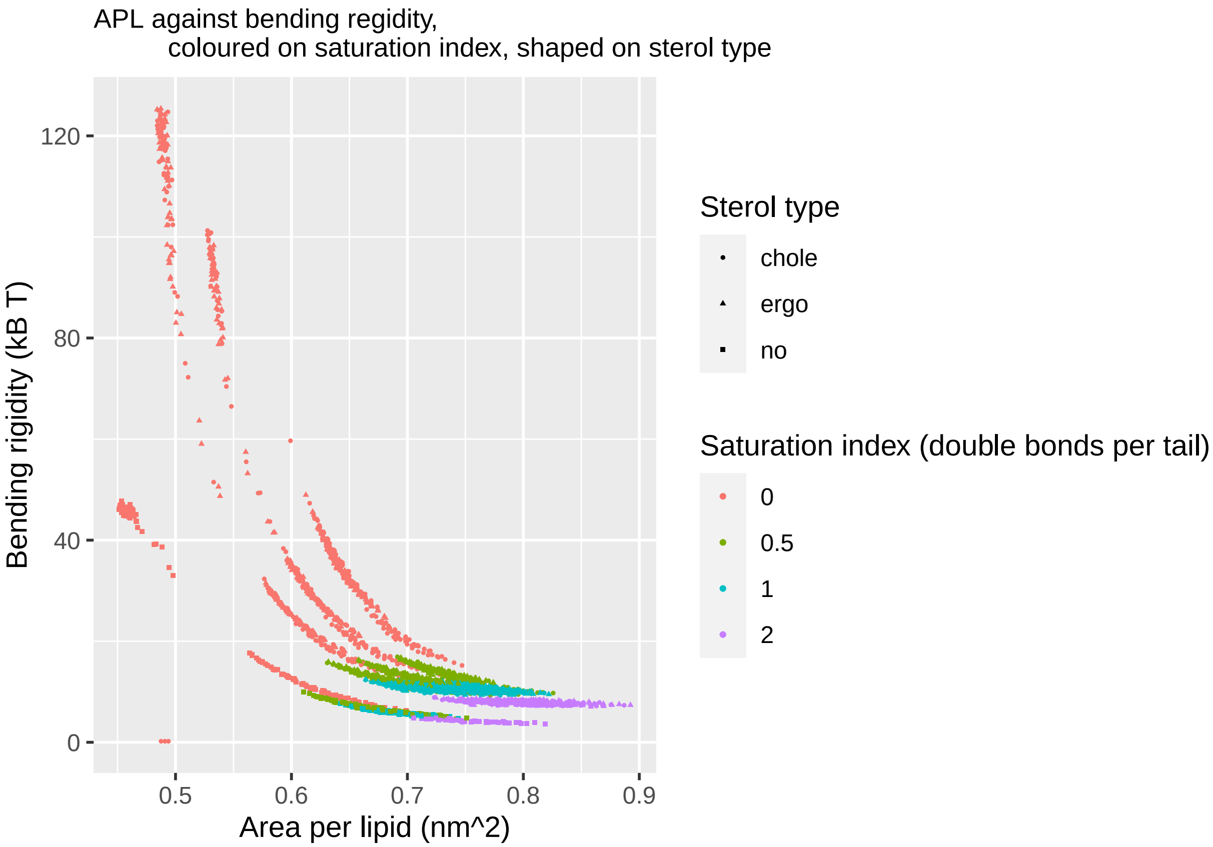
```
rects <- data.frame(xstart = seq(0,600,300), xend = seq(300,900,300), col = letters[1:3])

chart <- ggplot(nona_data, aes(temperature, bending, colour = factor(satur.index),
                               shape = factor(sterol.type) )) +
  geom_jitter(alpha=1/10) +
  xlab(get_des_by_ab(codebook, "temperature")) +
  ylab(get_des_by_ab(codebook, "bending"))  +
  ggtitle("Temperature against bending regidity,
          coloured by saturation index, shaped on sterol type") +
  theme(plot.title = element_text(size=10)) +
  labs(colour = get_des_by_ab(codebook, "sterol.type"),
      shape = get_des_by_ab(codebook, "satur.index"))


chart +
  geom_rect(aes(xmin = -Inf, xmax = 312.9, ymin = -Inf, ymax = Inf),
            color="orange",
            fill = "green", alpha = 0.0001) +
  geom_rect(aes(xmin = 313.1, xmax = Inf, ymin = -Inf, ymax = Inf),
            color="purple",
          fill = "yellow", alpha = 0.0001)
```



When looking at this plot there seem to be a clear separation between a saturation index of 0 while the rest of the values seem to be clustered quite close together. There does seem to be a small separation between 0.5, 1 and 2 values. An interesting observation is that there don't seem to be any data point with a saturation index of more than 0.5 in the 328 kelvin group (the purple box).

Here the APL is plot against the bending rigidity again, but here the colour is decided based on the saturation index and the shape on the sterol type.

```
chart2 <- ggplot(nona_data, aes(APL, bending, colour = factor(satur.index), shape = factor(sterol.type)
  geom_jitter(alpha=1, size=0.5) +
  xlab(get_des_by_ab(codebook, "APL")) +
  ylab(get_des_by_ab(codebook, "bending"))  +
  ggtitle("APL against bending regidity,
          coloured on saturation index, shaped on sterol type") +
  theme(plot.title = element_text(size=10))
chart2 + labs(colour = get_des_by_ab(codebook, "satur.index"), shape = get_des_by_ab(codebook, "sterol.
```

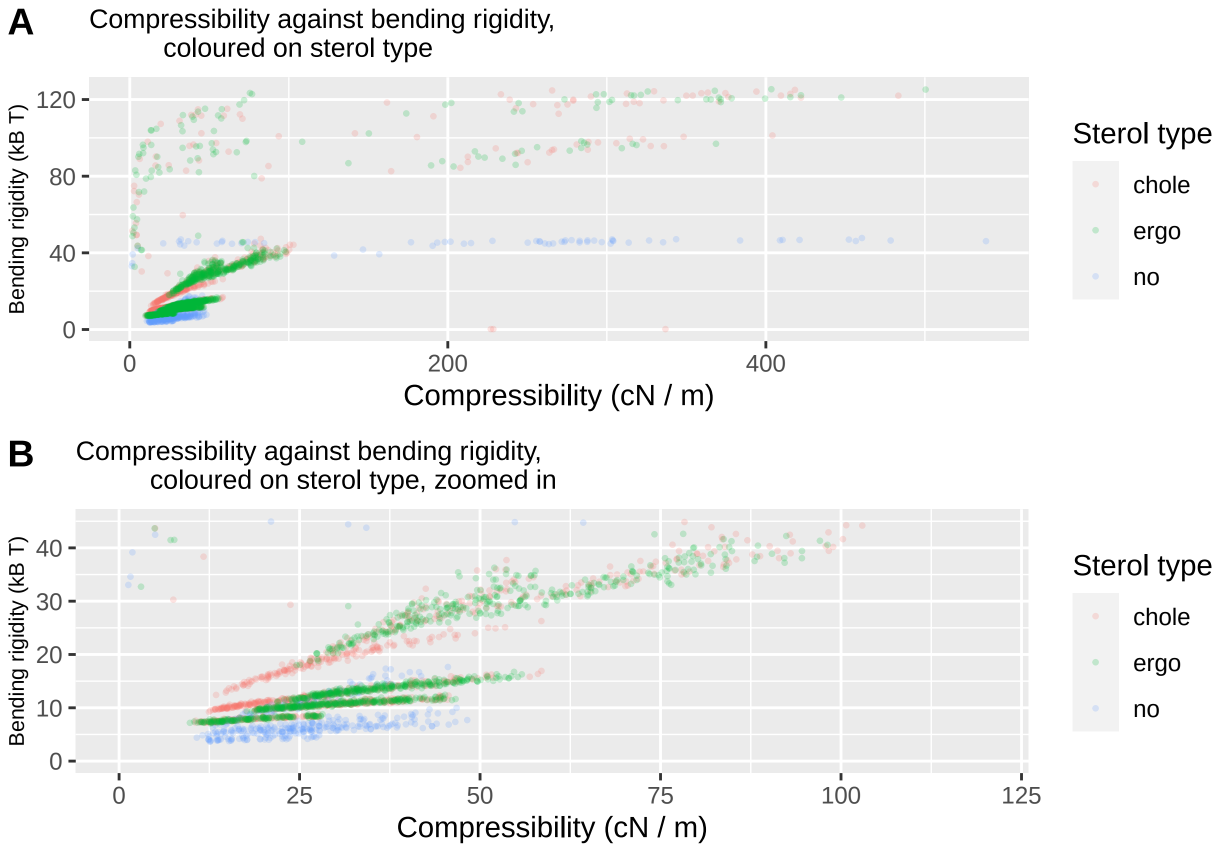


In the plot we can see the same kind of correlation as before, but now we can see that there also seem to be some kind of clustering based on the saturation index.

Here is a plot that shows the Compressibility plotted against the bending rigidity.

```
chart <- ggplot(nona_data, aes(compress, bending, colour = factor(sterol.type) )) +
  geom_point(alpha=2/10, size=0.5) +
  xlab(get_des_by_ab(codebook, "compress")) +
  ylab(get_des_by_ab(codebook, "bending")) +
  theme(axis.title.y = element_text(size=8)) +
  ggtitle("Compressibility against bending rigidity,
          coloured on sterol type") +
  labs(colour = get_des_by_ab(codebook, "sterol.type"))  +
  theme(plot.title = element_text(size=10))

chart2 <- ggplot(nona_data, aes(compress, bending, colour = factor(sterol.type) )) +
  geom_point(alpha=2/10, size=0.5) +
  xlab(get_des_by_ab(codebook, "compress")) +
  ylab(get_des_by_ab(codebook, "bending")) +
  theme(axis.title.y = element_text(size=8)) +
  ggtitle("Compressibility against bending rigidity,
          coloured on sterol type, zoomed in") +
  labs(colour = get_des_by_ab(codebook, "sterol.type")) +
  scale_x_continuous(limits=c(0,120)) +
  scale_y_continuous(limits=c(0, 45))  +
  theme(plot.title = element_text(size=10))
```

```r
ggarrange(chart,
          chart2,
          labels = c("A", "B"),
          ncol = 1, nrow = 2)
```

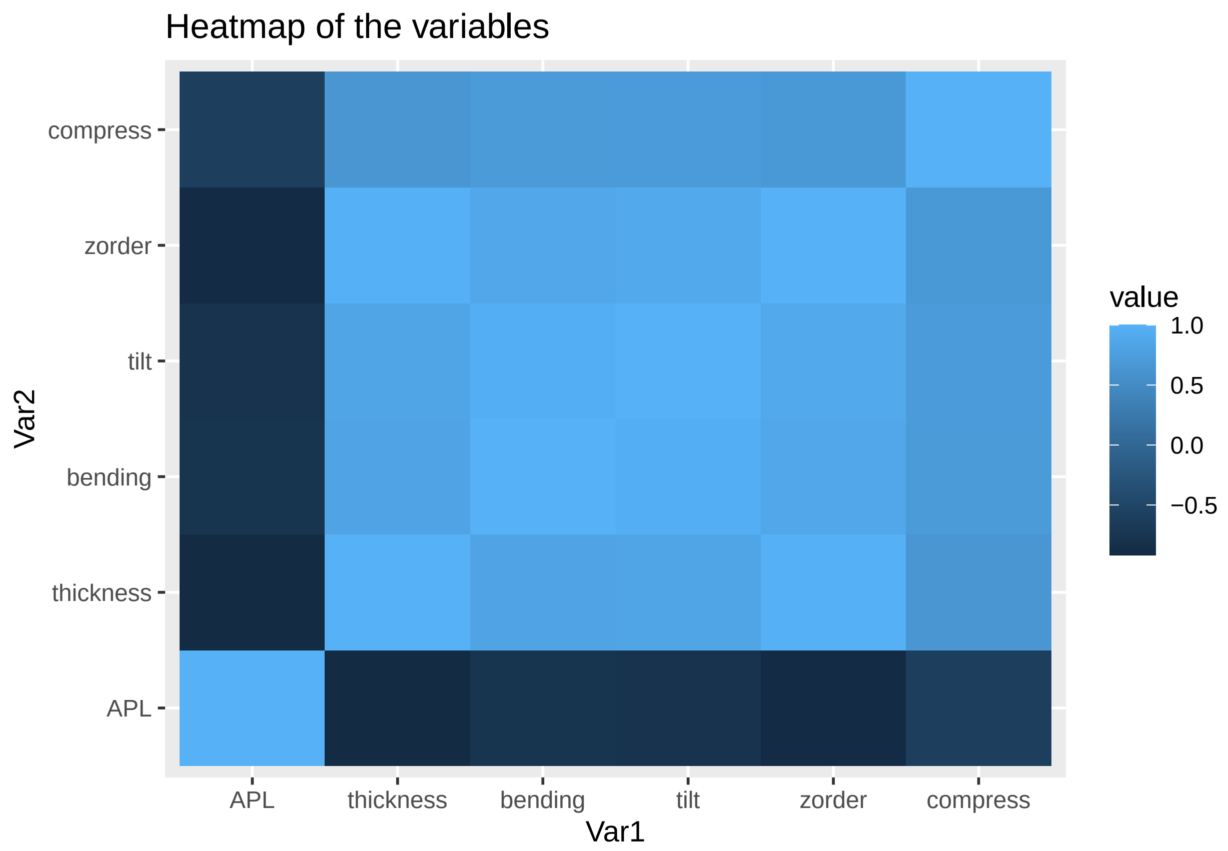


In plot A we can't really make out much of a pattern in the dense clout to the left, but when looking at the rest of the data points there does seem to be some grouping based on whether or not no sterol or a sterol is present. When zooming in in plot B we can now see the same kind of grouping happen again based on either a sterol present or no sterol present.

Heatmap

Here a heatmap is made to look for correlation between variables.

```r
corrolation <- nona_data[,9:14]
#cor2 <- na.omit(cor2)
#cormat <- round(cor(corrolation),5)
#head(cormat)

melted_cormat <- melt(cor(corrolation))
#head(melted_cormat)

ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  ggtitle("Heatmap of the variables")
```



This is a very interesting result. It seems like all of the variables seem to be quite heavily correlated, except for APL, which is negatively corrolated. It also seem like the comprehensibility is slightly less correlated then the rest of the variables. But thickness, bending, tilt and z order all seem to have a heavy correlation.

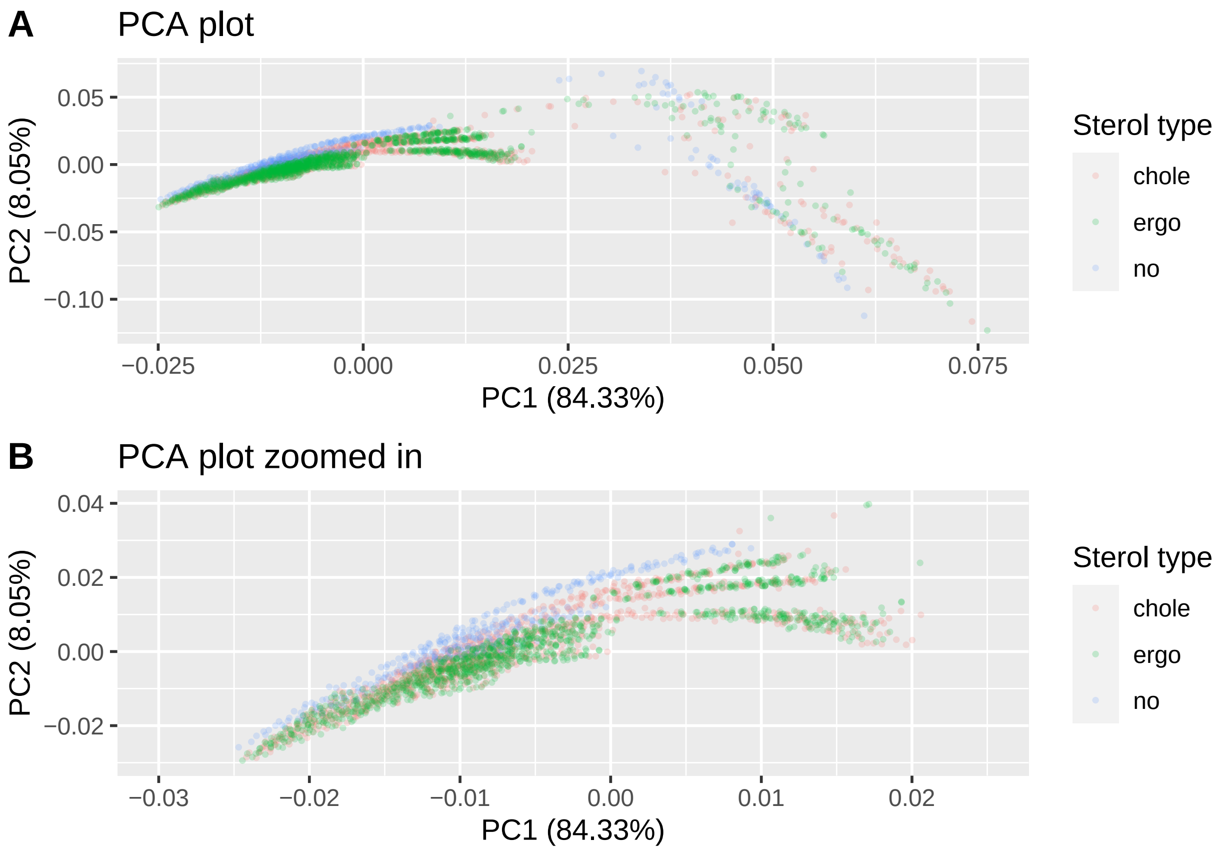
PCA

Here we'll make a PCA plot to look for possible clusters.

```
# PCA plot results
pca_res <- prcomp(corrolation, scale. = TRUE, center = TRUE)
print(pca_res)
```

```
## Standard deviations (1, .., p=6):
## [1] 2.24944886 0.69482659 0.55396674 0.31774787 0.20132585 0.09391948
##
## Rotation (n x k) = (6 x 6):
##                  PC1         PC2        PC3          PC4         PC5
## APL       -0.4039636 -0.38324528 -0.4127355 -0.705249607  0.14566451
## thickness  0.4238589  0.31048244  0.2156260 -0.511891894 -0.01429709
## bending    0.4152014 -0.14463396 -0.5644389  0.026515944 -0.69807879
## tilt       0.4196997 -0.08623976 -0.5085343  0.252592319  0.68632375
## zorder     0.4334754  0.22183399  0.0793969 -0.419528906  0.14134816
## compress   0.3474421 -0.82410989  0.4468250 -0.008899588  0.01561576
##                    PC6
## APL        0.031475796
## thickness  0.644369420
## bending   -0.008970044
## tilt       0.151540515
## zorder    -0.748732960
## compress   0.012300032
```

```r
# Create PCA plot
ggarrange(autoplot(pca_res, data = nona_data,
                   colour = 'sterol.type', alpha=2/10,
                   size = 5/10) +
            ggtitle("PCA plot") +
            labs(colour = get_des_by_ab(codebook, "sterol.type")),
          autoplot(pca_res, data = nona_data,
                   colour = 'sterol.type', alpha=1/5,
                   size = 1/2) +
            ggtitle("PCA plot zoomed in") +
            labs(colour = get_des_by_ab(codebook, "sterol.type")) +
            scale_x_continuous(limits=c(-0.03,0.025)) +
            scale_y_continuous(limits=c(-0.03,0.04)),
          labels = c("A", "B"),
          ncol = 1, nrow = 2)
```



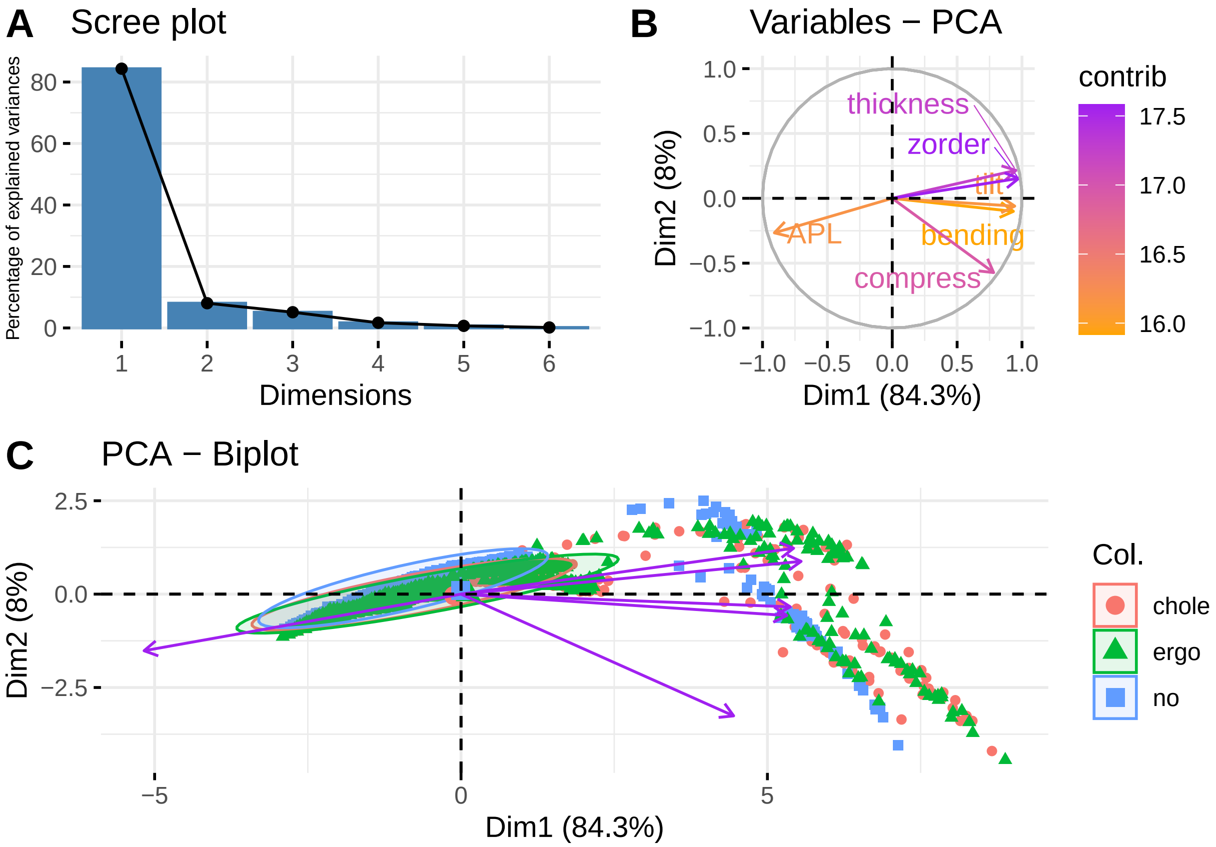
In plot A we can hardly see any clusters at all, it does seem to have a dense cloud to the right. Plot B cuts off the data point on the side and zooms in on the dense cloud of dots visible in plot A, a total of 271 dots aren't visible because of this. In this plot we can't really make out much of a pattern or cluster. It does, however, seem like no sterol cluster is sleightly seperated from the other sterol types.

Here is another way to make a PCA plot that is a bit more advanced seeing as it will also give us a way to identify correlation based on variables in the clustering.

```
# Create PCA plot
res.pca <- prcomp(corrolation, scale = TRUE)
plot1 <- fviz_eig(res.pca) +
  theme(axis.title.y = element_text(size=7))
plot2 <- fviz_pca_var(res.pca,
                      col.var = "contrib", # Color by contributions to the PC
                      gradient.cols = c("orange", "purple"),
                      repel = TRUE     # Avoid text overlapping
                      )
plot3 <- fviz_pca_biplot(res.pca, repel = FALSE,
                         col.var = "purple", # Variables color
                         col.ind = nona_data$sterol.type,  # Individuals color
                         label = FALSE,
                         addEllipses =  TRUE,
                         ellipse.type = "t"
                         )

# ggarrange(plot1, plot2, plot3,
#  labels = c("A", "B", "C"),
#  ncol = 2, nrow = 2)
plot <- arrangeGrob(plot1,  plot2,
                    plot3,
                    ncol = 2, nrow = 2,
                    layout_matrix = rbind(c(1,2), c(3,3)))
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```

```
plot <- as_ggplot(plot) +
  draw_plot_label(label = c("A", "B", "C"), size = 15,
                  x = c(0, 0.5, 0), y = c(1, 1, 0.5))
plot
```



This PCA plot confirms what we could already see in the heat map, that is that most of the variables seem to be heavily correlated, seeing as they are pointing in the same direction, except for APL and compress. We can also see that thickness and z order are correlated just like tilt and bending. The PCA plot also still looks the same, only now it has as ellipse for clusters that was calculated using a t test. The ellipses all seem to overlap so there doesn't seem to be much clustering.

Research question

Is it possible to use machine learning to reliably predicts the sterol type in a membrane with a higher than 80% accuracy, given the area per lipid, the bending rigidity and the compressibility?