

Cheat sheet for Classification

Libraries

```
library(tidyverse)
library(dplyr)
library(car)
library(caret)
library(ggfortify)
```

In order to carry out PCA. Scaling and centering can be don here as well!

```
model_pca <- prcomp(subset(dataset, select = -c(`00000Shrt_Desc`,
clusters_assigned_kmeans, clusters_assigned_hc)),
                    center = TRUE,
                    scale. = TRUE)
```

To see how the variance in the dataset is explained by the PC-s:

```
summary(model_pca)
```

To calculate the scores and loadings you can use:

```
dataset_pca <- as_tibble(predict(model_pca))
loadings_pca <- as_tibble(model_pca$rotation)
```

Now you can also add other data (e.g. categories from previous lab) to the same tibble.

To see a nice 3D graph (NB! See line "groups"):

```
car::scatter3d(x = dataset_pca$PC1,
              y = dataset_pca$PC2,
              z = dataset_pca$PC3,
              main="3D PCA",
              xlab = "PC1",
              ylab = "PC2",
              zlab = "PC3",
              surface = FALSE,
              groups = dataset_pca$clusters_assigned_hc,
              axis.col = c("black", "black", "black"),
              axis.size = 2,
              sphere.size = 3)
```

On 2D ggplot is the best:

```
ggplot() +
  geom_point(mapping = aes(x = dataset_pca$PC1,
                          y = dataset_pca$PC2,
                          color =
dataset_pca$clusters_assigned_hc),
            size = 2,
            alpha = 0.75) +
  geom_segment(mapping = aes(x = 0,
                            xend = 10*loadings_pca$PC1,
                            y = 0,
                            yend = 10*loadings_pca$PC2),
              arrow = arrow(),
              color = "gray31") +
  geom_text(mapping = aes(x = 11*loadings_pca$PC1,
                          y = 11*loadings_pca$PC2,
```

```
        label = colnames(dataset[4:14])),
      color = "gray31") +
labs(x = "PC1", y = "PC2") +
theme(panel.background = element_blank(),
      axis.ticks = element_blank(),
      axis.text = element_text(size=14, face = "plain", colour =
"gray"),
      axis.title = element_text(size=14, face = "plain", colour =
"gray"),
      legend.position = "bottom"
)
```

ggplot can also be easely saved as .svg, .pdf, .jpg, .tiff, ...

```
ggsave("name.svg")
```