Assignment 2

Caolan McDonagh

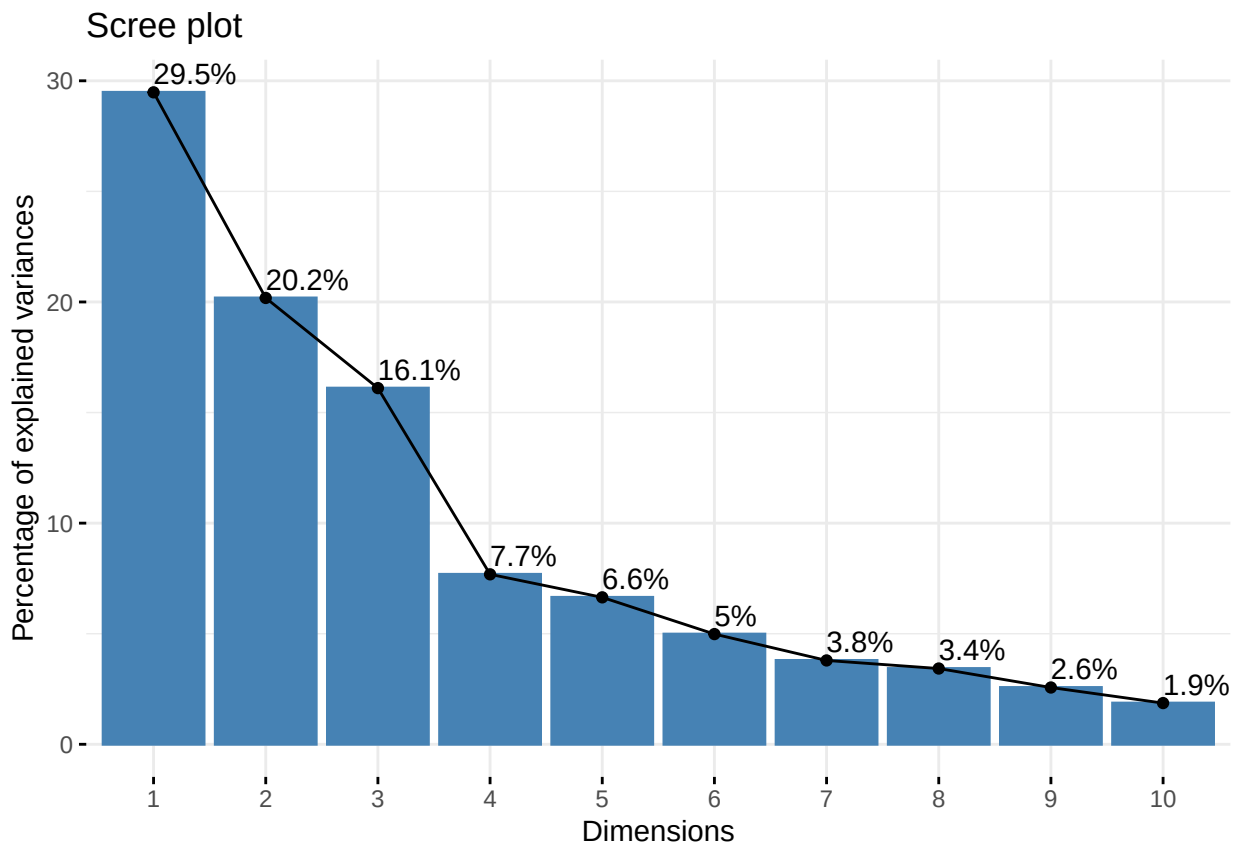
31/01/2022

Data Visualisation - Assignment 2

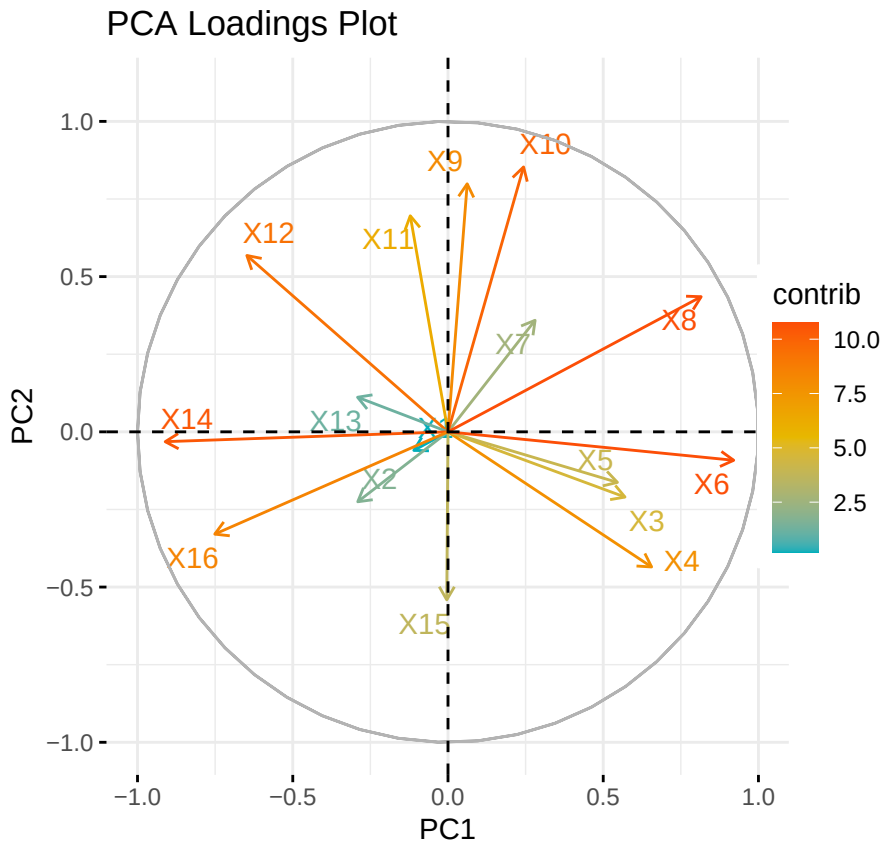
By Caolan McDonaghw

*Skeleton r code adapted from PCA_Scatterplot_Worksheet.pdf

```
library(colorspace)
library(dplyr)
library(factoextra)
library(FactoMineR)
library(ggplot2)
library(ggrepel)
library(kableExtra)
library(patchwork)
library(readr)
library(RColorBrewer)
library(scales)
library(shiny)
library(shinyjs)
library(colorblindr)
library(ggrepel)

#Section 1 - PCA:

#Loading Data
Pendigits<-read_csv(file = "pendigits.csv", col_names = FALSE)

Pen_digits <- Pendigits


#PCA
dplyr::select(Pen_digits, -17) %>%
  PCA(graph = FALSE) -> pca


#Scree
fviz_screeplot(pca, choice="variance", addlabels = TRUE,)
```

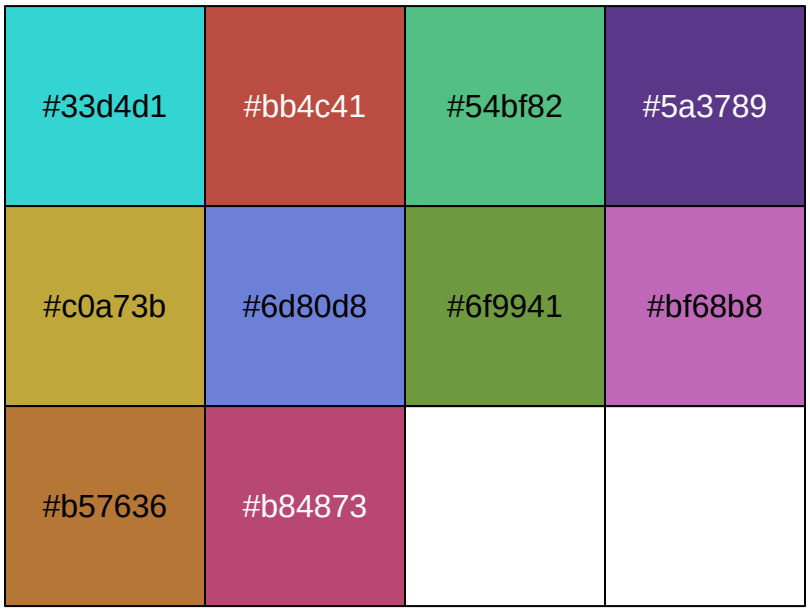



```
#Loadings
loadings_plot <- fviz_pca_var(pca,
                              col.var = "contrib",
                              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                              repel = TRUE
) +
  xlab("PC1") +
  ylab("PC2") +

  ylim(c(-1,1.1))+
  ggtitle("PCA Loadings Plot") +

  theme(
    legend.box.background = element_rect(fill = "white", color = "white"),
    legend.position = c(1.055, 0.5),
    legend.direction = "vertical",

    legend.box.margin = margin(0.05, 0.05, 0.05, 0.05),
    legend.key = element_rect(fill = "white"),
  )

loadings_plot
```

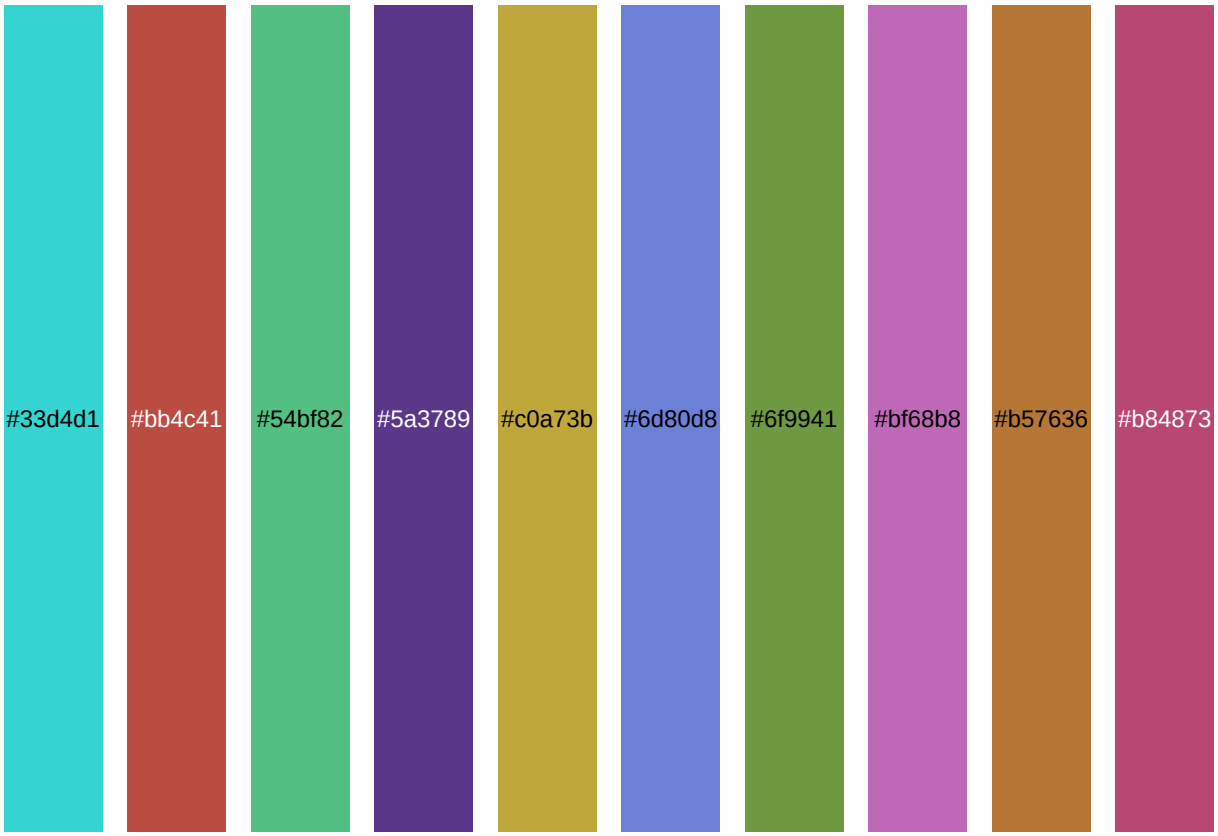



```
data_pca_ind <- get_pca_ind(pca)

data_pca <- data_pca_ind$coord[,c(1,2)]
data_pca <- as.data.frame(data_pca)

names(data_pca)[1] <- "PC1"
names(data_pca)[2] <- "PC2"

Digits <- as.factor(Pen_digits[[17]])

data_pca<- cbind(data_pca, Digit = Pen_digits[[17]])



#Section 2 - Colour:



colors <-  c("#33d4d1",
             "#bb4c41",
             "#54bf82",
             "#5a3789",
             "#c0a73b",
             "#6d80d8",
             "#6f9941",
             "#bf68b8",
```

```
              "#b57636",
              "#b84873")

colorElip <-  c("#33d4d1",
              "#bb4c41",
              "#54bf82",
              "#5a3789",
              "#c0a73b",
              "#6d80d8",
              "#6f9941",
              "#bf68b8",
              "#b57636",
              "#b84873")

show_col(colors)
```




```
palette_plot(colors, label_size = 3)
```




```
print("See the below for a list of the above HEX values: ")
```

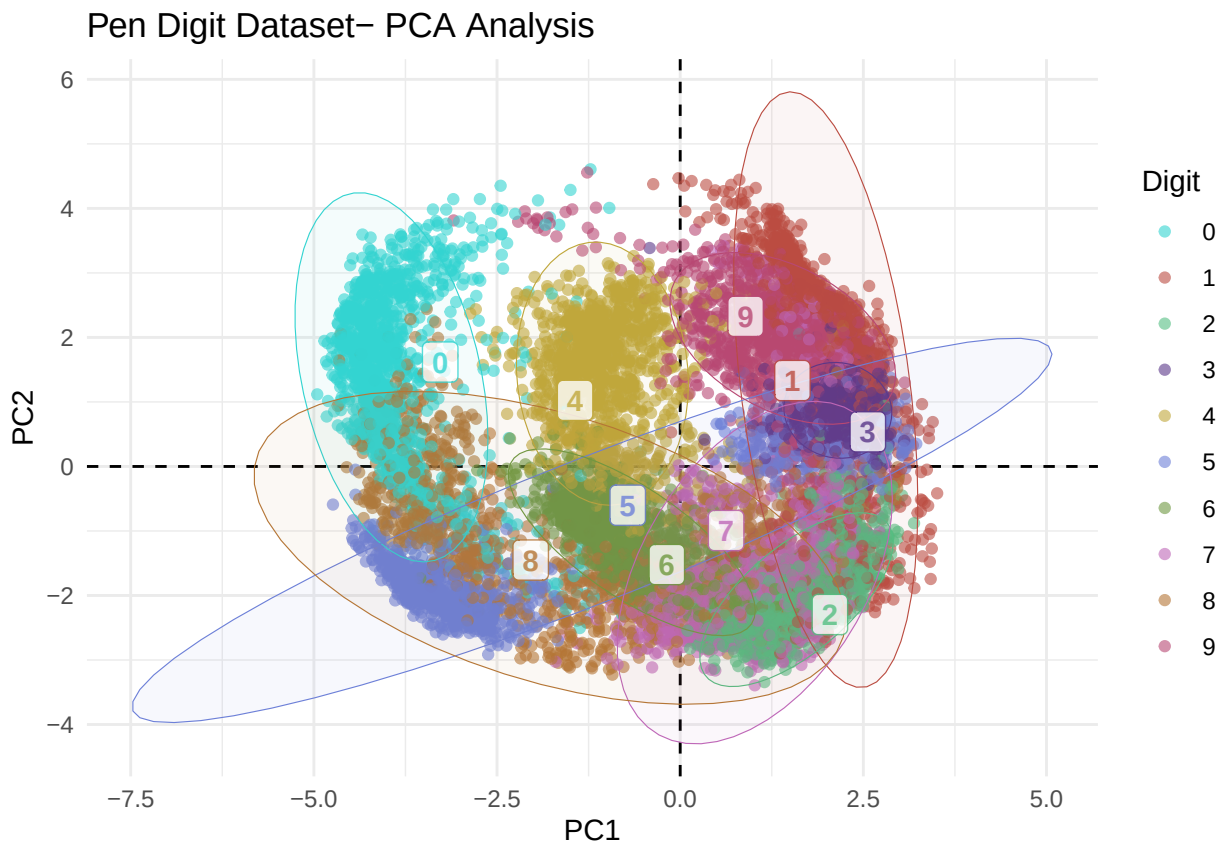
```
## [1] "See the below for a list of the above HEX values: "
```

```
print(paste(colors))
```

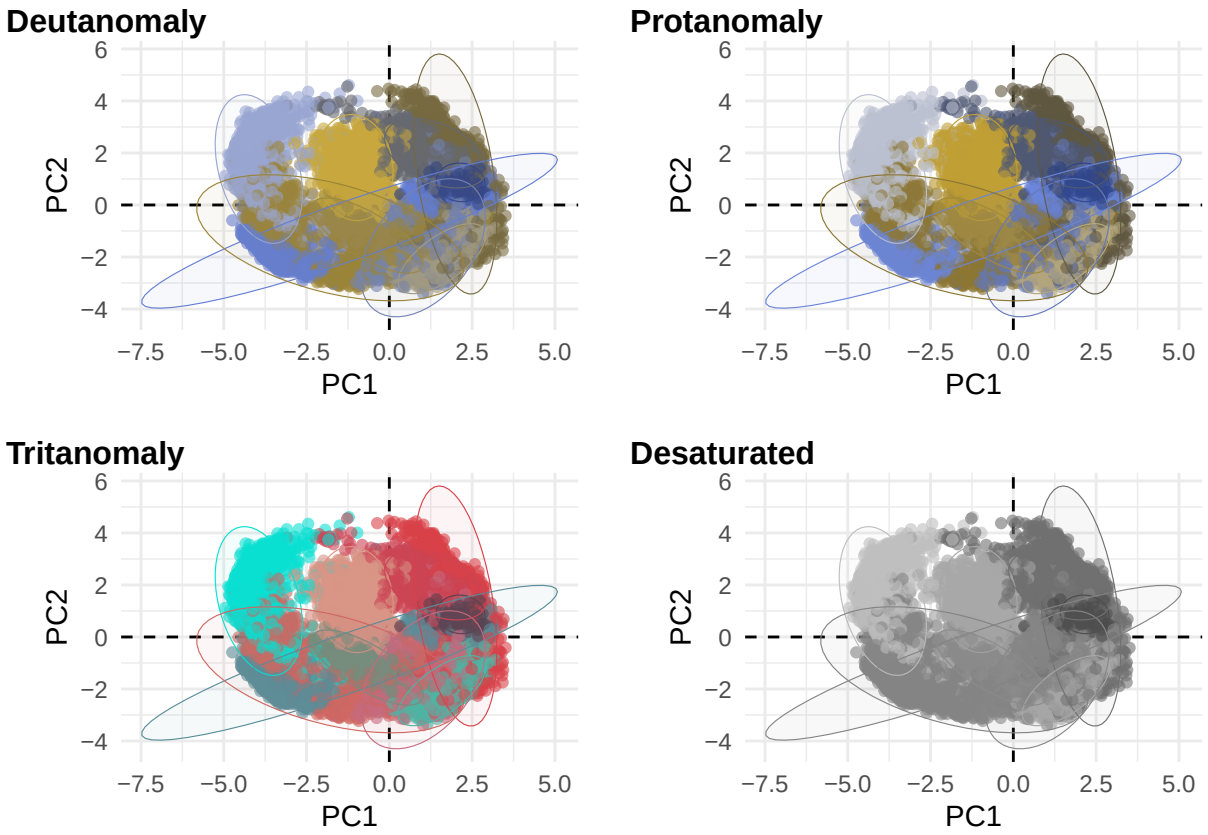
```
##  [1] "#33d4d1" "#bb4c41" "#54bf82" "#5a3789" "#c0a73b" "#6d80d8" "#6f9941"
##  [8] "#bf68b8" "#b57636" "#b84873"
```

```
#Section 3: ggplot

means <- data_pca %>%
  group_by(Digit) %>%
  summarise(PC1 = mean(PC1),
            PC2 = mean(PC2))

par(mar=c(1,3,1,1))
fig <- ggplot(data_pca,aes(x=PC1, y=PC2, color=as.factor(Pen_digits[[17]]))) +

  geom_vline(xintercept = 0, linetype="dashed") +
  geom_hline(yintercept = 0, linetype="dashed") +

  geom_point(size=1.5, alpha = 0.6) +
```

```
  stat_ellipse(geom = "polygon",type = "t",size = 0.2,
               aes(fill = as.factor(Pen_digits[[17]])),
               alpha = 0.05, show.legend = FALSE) +

  scale_fill_manual(values=colors)+


  scale_color_manual(values = colors, name = "Digit")  +

  geom_label_repel(data = means, size = 4, aes(x = PC1, y = PC2, label = Digit,
                   color = as.factor(Digit)),
                   segment.colour = "black",
                   fontface = 'bold',
                   show_guide = F,
                   alpha=0.8) +



  theme_minimal() +
  ggtitle("Pen Digit Dataset- PCA Analysis")

fig
```



```
#Duplicate of previous scatterplot minus the label, for visiblity.
cvdfig <- ggplot(data_pca,aes(x=PC1, y=PC2, color=as.factor(Pen_digits[[17]]))) +

  geom_vline(xintercept = 0, linetype="dashed") +
  geom_hline(yintercept = 0, linetype="dashed") +

  geom_point(size=1.5, alpha = 0.6) +



  stat_ellipse(geom = "polygon",type = "t",size = 0.2,
               aes(fill = as.factor(Pen_digits[[17]])),
               alpha = 0.05, show.legend = FALSE) +

  scale_fill_manual(values=colors)+


  scale_color_manual(values = colors, name = "Digit")  +

    theme_minimal() +
    theme(legend.position = "none")

cvd_grid(cvdfig)
```



Colour

When thinking on a colour palette, there was no real "importance" as to what color represented what digit as far as I thought. With this I went with a more distributed approach to alter each color to be as seperate as possible, to distingush each digit as its own.

This was accomplished by setting a standard chroma (45 in hcl color picker) and standard luminance (60 in hcl). This left me with a complete hue palette to pick from, e.g there was options 0-324 (as 0==360) in my Hue-Chroma plane.

Here I simple divided out my 10 colors equally into the hue, 10 colors of increasing hue intervals of 36, starting at 0 and ending in 324.

This landed me my base palette, I then went in and tweaked certain colours that I considered to be visually similar to other colors via the HCL Color picker and the luminance-chroma plane to eye ball more appealing/distinguishable colours.

I then realized I was oblivious to CVD in this, so I made use of the "iwanthue" package and k-Means to generate a 10 colour palette, which was colourblind friendly.

Colour Retention in Scatter Plot

Inside of this scatter plot I decided it was best to retain colour of the data points and ellipses as the amount of digits and mutual overlap in places would cause ambiguity when studying the scatter plot. Without this the data visualization would mean very little.

Overlap

We see quite a bit of overlap in this scatter plot due to the simple fact there is only so much differences between digits when they're written. For example 6 and 8 are very similar when written, this would cause coordinate data from the tablet to show similarities in the dataset. The same can be said for 1 and 4. This would cause overlap in our PCA.

CVD

We can see by the CVD simulation, showing Deutanomaly, Protanomaly, Tritanomaly and a Desaturated it isn't the easiest thing to interpret, there are a couple of colours that are too indistinguishable in my opinion. I think for the most part someone with CVD could interpret most of the plot but could struggle with certain digit colours that turned out too close, especially where there are large clusters of data points overlapping, such as the bottom right of our scatter plot. The type of CVD would also be a factor in the difficulty in this, as it is hard to cater to one perception of colour, where types of CVD can alter the perception of different colours uniquely.