Assignment 2: Visualisation of PCA

Kalyani Prashant Kawale

Student ID: 21237189

Section 1: PCA

Section 1 Solution

Following the worksheet [1], PCA was performed on the given data-set of handwritten digits and visualizations were created as follows:

```
# Loading libraries
library(readr)
library(dplyr)
library(FactoMineR)
library(factoextra)
library(ggplot2)
library(colorblindr)
```

The data-set given in *pendigits.csv* file was read into a data frame as follows:

```
# Reading data from csv file
DIGITS <- read_csv(file = 'pendigits.csv', col_names=FALSE)

# Renaming target column to digit
names(DIGITS)[17] <- "digit"

# Displaying first 5 rows of DIGITS
head(DIGITS)
```

```
## # A tibble: 6 x 17
##      X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12   X13
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    88    92     2    99    16    66    94    37    70     0     0    24    42
## 2    80   100    18    98    60    66   100    29    42     0     0    23    42
## 3     0    94     9    57    20    19     7     0    20    36    70    68   100
## 4    95    82    71   100    27    77    77    73   100    80    93    42    56
## 5    68   100     6    88    47    75    87    82    85    56   100    29    75
## 6    70   100   100    97    70    81    45    65    30    49    20    33     0
## # ... with 4 more variables: X14 <dbl>, X15 <dbl>, X16 <dbl>, digit <dbl>
```

- Performing PCA on the data set with 16 features to reduce the dimensionality from 16 to 2.

- No feature processing was performed on DIGITS as all 16 features are numeric factors labeled from X1 to X16.
- The factors are not scaled either as each factor has levels on the same scale (values between 0 to 100).

```
# Using dplyr select method to select all dependent factors from data set,
# except for the target factor "digit".
# The selected factors are piped into PCA method of "FactoMineR" library
# Applying PCA to DIGITS data-set and saving the components in "pca" object
select(DIGITS, -digit) %>% PCA(graph = FALSE) -> pca
```

- Plotting the Scree plot to analyse variance captured by each principle component.
- The first two principle components cover around 50% of data variance of the data as depicted by the plot below,

```
# Plotting scree plot
fviz_screeplot(pca, choice="variance", addlabels = TRUE)
```

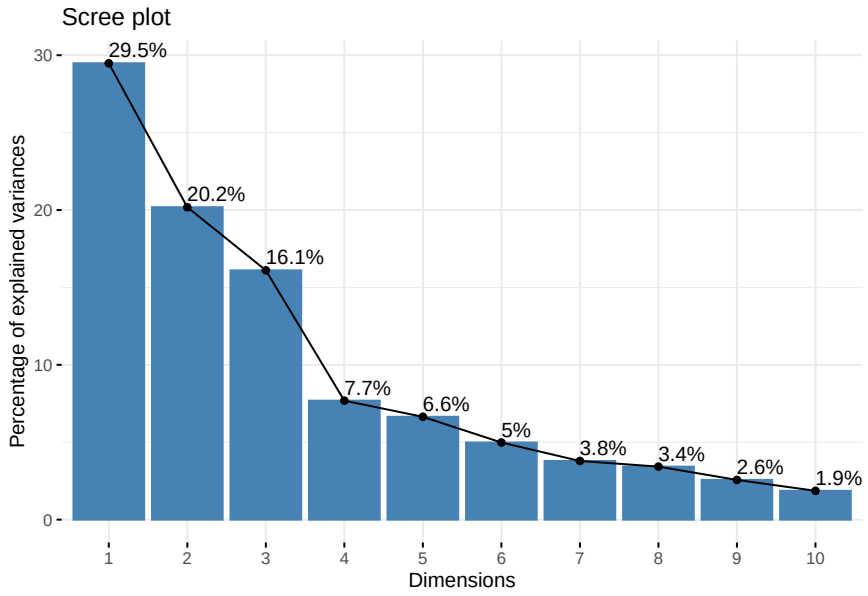

Figure 1: Data Variance Distribution of Principle Components for DIGITS data set

Plotting the loadings plot,

```
# Plotting loadings_plot
loadings_plot <- fviz_pca_var(pca, col.var = "contrib",
                 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                 repel = TRUE) +  xlab("PC1") +
                 ylab("PC2") +
                 # removing the custom title, plot fig is captioned below
                 ggtitle("") +
                 theme(
                    legend.position = c(1.10, 0.85),
```

```
                    legend.box.margin = margin(0, 0.05, 0.05, 0.05)
                )
```

```
loadings_plot
```

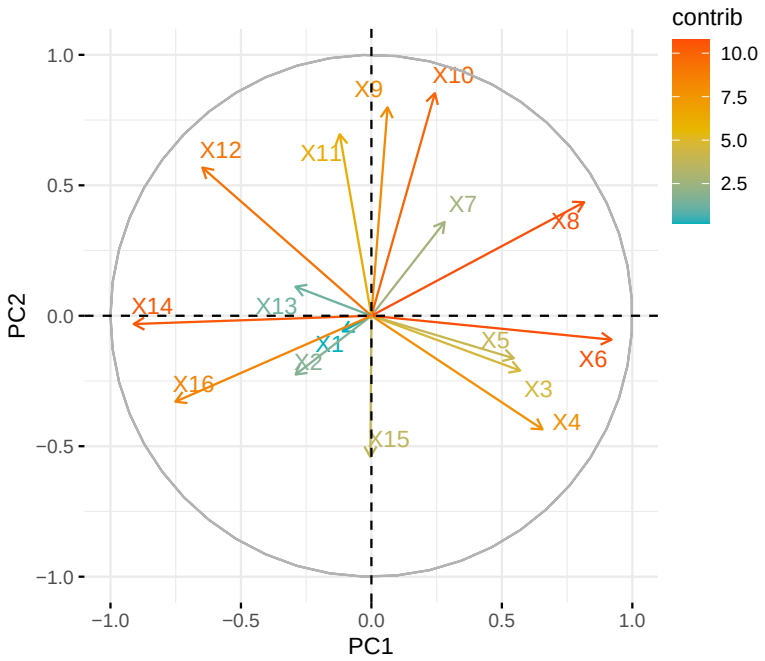

Figure 2: PCA Loadings Plot for DIGITS Dataset

- From Figure 1, it can be concluded that factors X6, X8, X14 have the highest impact among all features.

- X6 and X8 have a strong positive correlation with principle component 1 (PC1), and X14 has a negative correlation with PC1, however, while X8 also has a positive correlation with PC2, X6 and X14 do not have any significant relation with PC2.

- Other features that may have a significant impact on the principle components are,

    1. X4 has a positive correlation with PC1 and a negative correlation with PC2.
    2. X10 and X9 have a positive correlation with PC2, but little relation with PC1.
    3. X16 has a negative correlation with both the principle components.
    4. X12 has a positive correlation with PC2, but is negatively correlated to PC1.

- The remaining features have a combination of positive and negative correlation with PC1 and PC2, but their impact is smaller.

Section 2: Custom Palette

Section 2 Solution

The custom palette for the 10 qualitative classes of digits from 0 to 9 was created as follows,

1. The HSL(A) functionality of colorizer.org tool was used to create the palette.
2. The saturation and lightness were set to fixed values and the hue was changed to get 10 distinct colors.
3. Further, the hex values obtained from colorizer.org were fed into the hcl_color _picker() tool, where the Chroma and Luminance for each hex value was adjusted to match each other and the palette was exported as hex values.
4. Due to the large size of classes, red and green colors were selected together despite the problem they present for people with color vision deficiency, however a darker red and a lighter green was created using hcl_color_picker() to make the visualization more perceptible.
5. The palette was also tested for different classes using scatter plot, and colors were assigned to classes to provide as clear distinction as possible.

```
# Saving the hex values obtained from colorizer.org and
# adjusted with hcl_color_picker in "custom_palette"
custom_palette <- c('#9F69E1', '#59C7DB',  '#F1954D', '#E562A5','#6C7AEA',
                    '#59DCB2', '#CB464E',  '#EFD456', '#D065DF', '#4D8FD3')
```

```
## [1] "The hex values for the colors in the custom palette are:"

##  [1] "#9F69E1" "#59C7DB" "#F1954D" "#E562A5" "#6C7AEA" "#59DCB2" "#CB464E"
##  [8] "#EFD456" "#D065DF" "#4D8FD3"
```

```
# Plotting the color bar using palette_plot method
palette_plot(custom_palette, label_size=3)
```

Figure 3: Custom Palette of 10 Colors to Represent Digits 0 to 9

Section 3: ggplot2 Scatterplot

Section 3 Solution

The scatterplot of data using first two principle components is as follows:

```
# Getting the projection of samples in terms of principle components
data_pca_ind <- get_pca_ind(pca)
# Selecting all rows in data-set with first two principle component values
data_pca <- data_pca_ind$coord[,c(1,2)]
# Converting data into a data frame
data_pca <- as.data.frame(data_pca)
# Naming the columns
names(data_pca)[1] <- "Principle_Component_1"
names(data_pca)[2] <- "Principle_Component_2"
# Combing the PC values with target column for each data point
data_pca<- cbind(data_pca, DIGITS$digit)
# Setting the target column name to digit
names(data_pca)[3] <- "digit"
# Setting digit to factor
data_pca$digit <- as.factor(data_pca$digit)
```

Plotting Scatterplot for *data_pca* with data points plotted using **geom_point** and colored using the custom palette created in section 2:

```
# ggplot for creating scatterplot for data_pca
digits_scatter <- ggplot(data_pca, aes(x=Principle_Component_1, y=Principle_Component_2,
                    color=digit)) +
  geom_point(alpha=0.4) +
  geom_vline(xintercept = 0, linetype="dashed") +
  geom_hline(yintercept = 0, linetype="dashed") +
  scale_colour_manual(values = custom_palette) +
  scale_fill_manual(values = custom_palette) +
  stat_ellipse(geom = "polygon",type = "t", size = 0.2,
               aes(fill = digit),
               alpha = 0.08) +
  theme_minimal()
digits_scatter
```

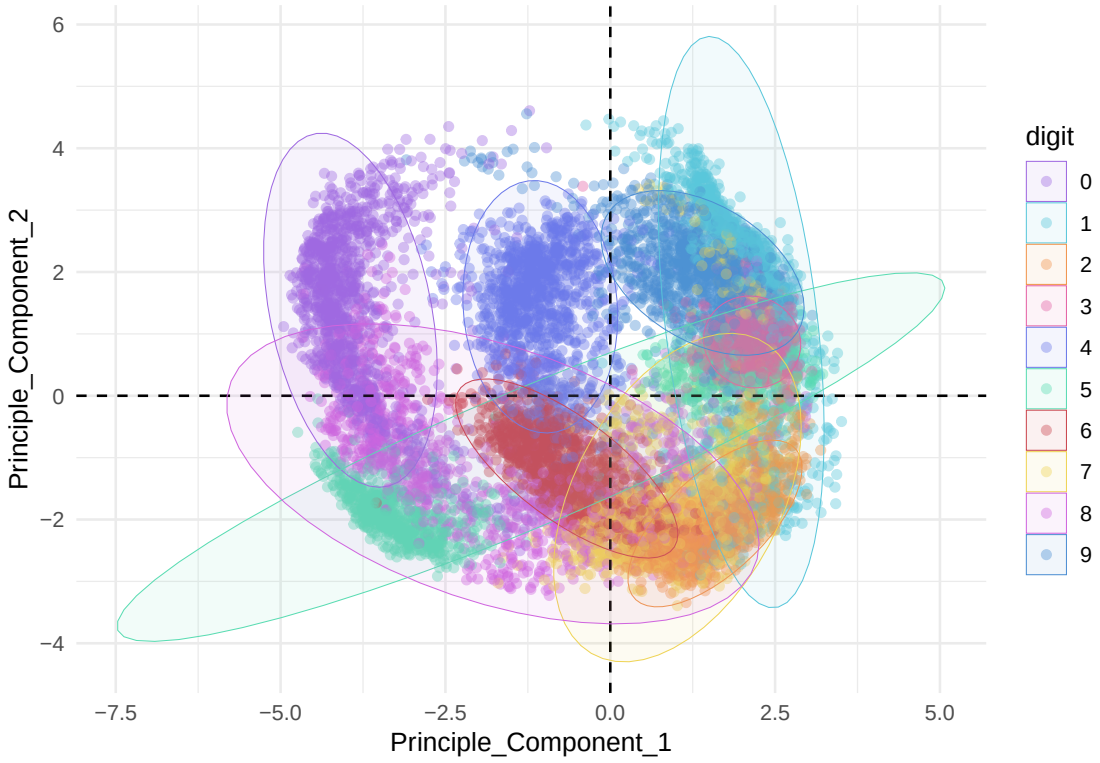

Figure 4: PCA Scatterplot for DIGITS Dataset

The following code was used to represent the scatter plot without colors,

```
# ggplot for creating scatterplot for data_pca without colored data points
ggplot(data_pca, aes(x=Principle_Component_1, y=Principle_Component_2, color=digit)) +
  geom_point(alpha = 0.4, colour="black") +
  stat_ellipse(geom = "polygon",type = "t",size = 0.2,
               aes(fill = digit),
               alpha = 0.08) +
  geom_vline(xintercept = 0, linetype="dashed") +
  geom_hline(yintercept = 0, linetype="dashed") +
  theme_minimal()
```

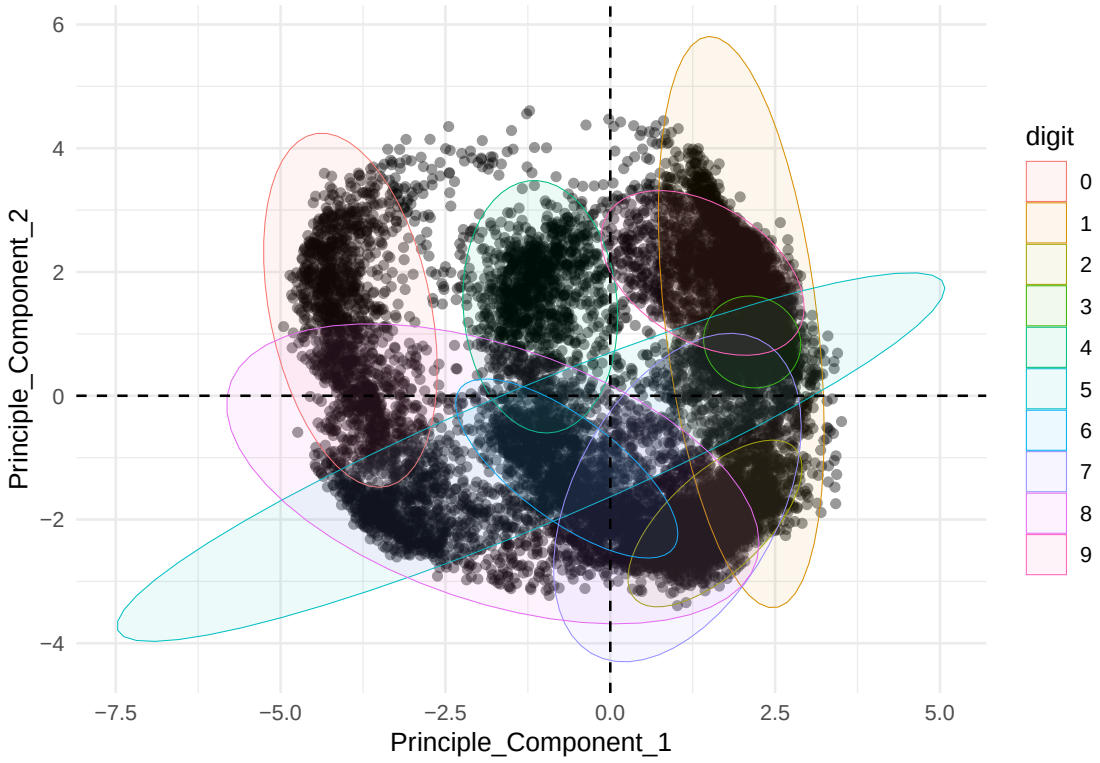

Figure 5: PCA Scatterplot for DIGITS Dataset, without colored data points

- As can be seen from Figure 5, removing the color of data points creates ambiguous visualization due to the multiple overlapping data points and number of classes.
- Compared to Figure 5, Figure 4 visualization, with colored data points gives better insight into which classes have overlapping features.
- To further understand which classes have similar features and overlap, removing the data points from the Scatterplot and retaining colored ellipses provides a clearer visualization as follows,

```
# ggplot for creating scatterplot for data_pca without data points
# and containing only colored ellipses
ggplot(data_pca, aes(x=Principle_Component_1, y=Principle_Component_2,
                     color=digit)) +
  geom_vline(xintercept = 0, linetype="dashed") +
  geom_hline(yintercept = 0, linetype="dashed") +
  scale_colour_manual(values = custom_palette) +
  scale_fill_manual(values = custom_palette) +
  stat_ellipse(geom = "polygon",type = "t", size = 0.2,
               aes(fill = digit),
               alpha = 0.3) +
  theme_minimal()
```

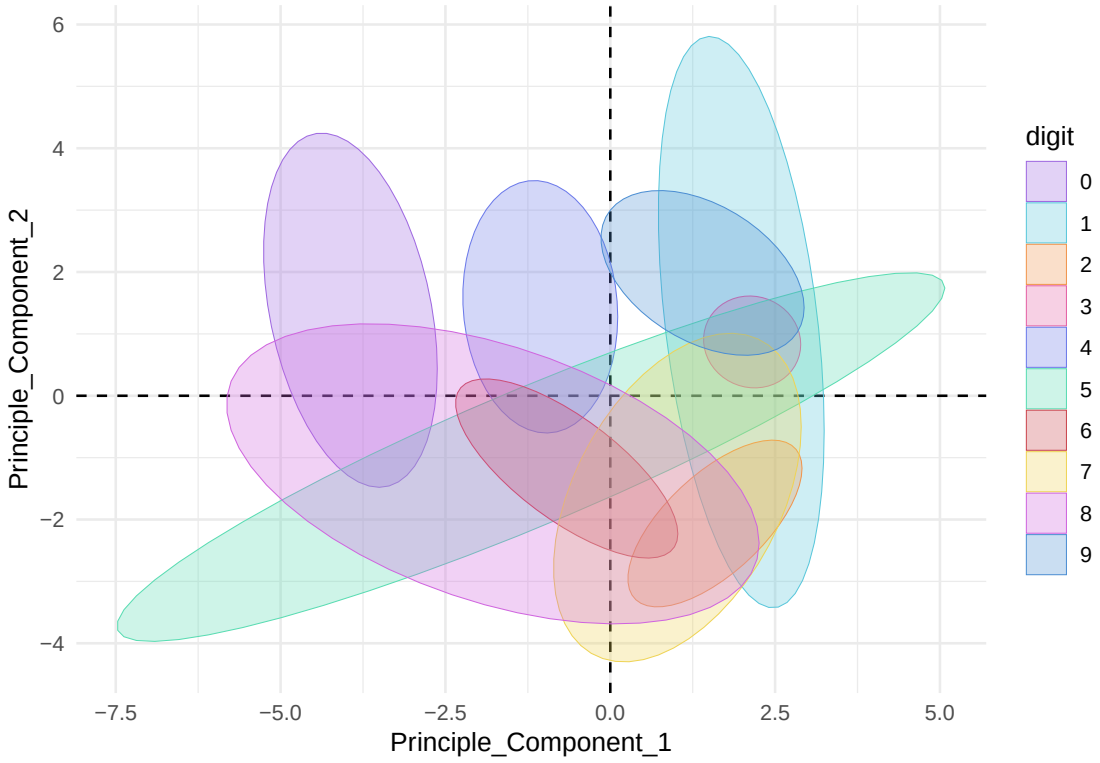

Figure 6: PCA Scatterplot for DIGITS Dataset without Data Points

- In all three visualizations (Figure 4, 5 and 6) there is an overlapping between the groups of data points.
- This is due to the fact that the numeric features representing the digits may have close values for the digits with similar shapes and curves.
- For example, the digits 3 and 5 have similar shapes, which implies that they must share certain features with same values, these features are projected onto the principle components as well. The pink colored ellipse representing digit 3 is almost contained within the group for digit 5 represented as the green ellipse.

CVD Simulation:

cvd_grid(digits_scatter) plots the CVD simulation of Figure 4,

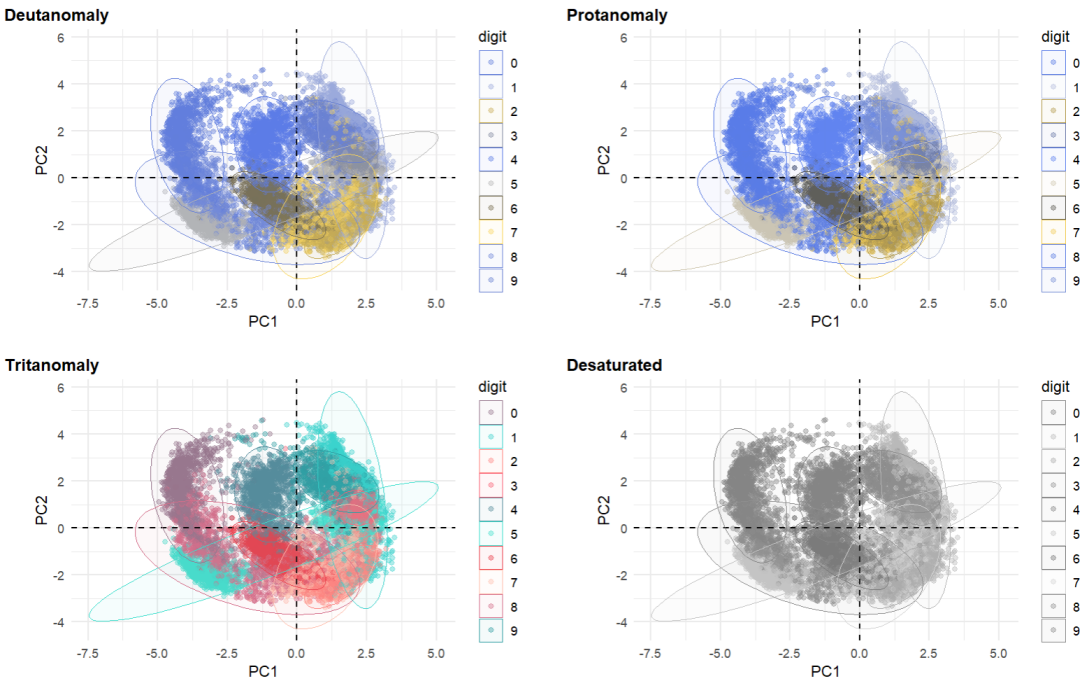

Figure 7: CVD Simulation of Scatterplot for Digits Dataset

- From the simulation displayed in Figure 7, it can be deduced that the plot might not be easily readable for someone with cvd.
- While the plot depicted in Figure 8 below, might provide a general overview of the classes and their relation with the principle components and each other, the large number of target classes make it difficult to represent each of the 10 classes with a distinct hue, leading to the use of related colors in the palette.
- It can be seen that atleast 3 to 4 classes end up having the same color in the CVD simulation even when the colors are distinct in plot without CVD simulation.

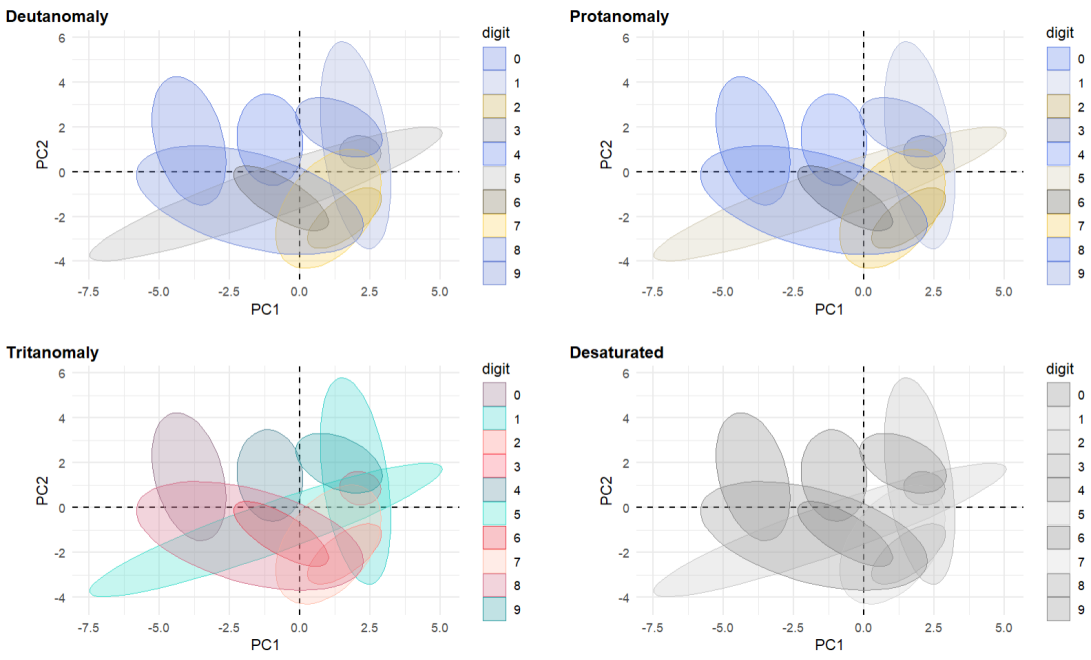

Figure 8: CVD Simulation of Scatterplot for Digits Dataset, Without Data Points

Acknowledgments

Following resources were referred to perform above tasks,

[1] Dr. Conor Hayes. (2022). Visualising a Principal Component Analysis Using A Scatterplot.

[2] Pulagam, S. (2020) All you need to know about PCA technique in Machine Learning. Available at: https://towardsdatascience.com/all-you-need-to-know-about-pca-technique-in-machine-learning-443b0c2be9a1 (Accessed on: 25/01/2022)

[3] Dr. Conor Hayes. (2022). Week 3 Lecture: Introducing Color Principle.

[4] Dr. Conor Hayes. (2022). CVD simulation using the colorblindr package.

[5] barplot: Bar Plots. Available at: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/barplot (Accessed on: 25/01/2022)

[6] Setting a height in Barplots for R. Available at: https://stackoverflow.com/questions/9287903/setting-a-height-in-barplots-for-r (Accessed on: 25/01/2022)