PCA_Tutorial

Michelle Zong, Daniia Newman, Jasmin Mendoza

2024-05-25

Codebook

Breast Cancer Wisconsin (Diagnostic) Data Set For this demo, we used a subset of the Breast Cancer Kaggle data set.

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Attribute Information:

- ID - int - unique identifier for each patient
- Diagnosis - str - breast cancer diagnosis (B = Benign, M = Malignant)

Real-valued features (10) are computed for each cell nucleus. All feature values are re-coded with four significant digits:

- Radius_mean - float - mean of distances from center to points on the perimeter
- Texture_mean - float - standard deviation of gray-scale values
- Perimeter_mean - float - mean size of the core tumor
- Area_mean - float - mean area of the core tumor
- Smoothness_mean - float - mean of local variation in radius lengths
- Compactness_mean - float - mean of perimeter^2 / area - 1.0
- Concavity_mean - float - mean of severity of concave portions of the contour
- Concave points_mean - float - mean for number of concave portions of the contour
- Symmetry_mean - float - mean of symmetry
- Fractal_dimension_mean - float - mean for 'coastline approximation' - 1

Introduction

In this tutorial, we will perform **Principal Component Analysis (PCA)** on the Breast Cancer data set. Principal Component Analysis (PCA) is an unsupervised statistical technique used to reduce the dimensionality of a data set while retaining most of the variance present in the data. PCA does what the name describes, it finds 'principal components' of the data.

- It is a tool often used for data visualization or in data pre-processing before supervised techniques are applied. PCA can also be used to *impute* missing data values through a process called *matrix completion*, which is where all of our Netflix recommendations come from, but we will not go into that today.

- The process if PCA involves transforming the original variables into a new set of uncorrelated variables, known as principal components. Being uncorrelated is equivalent to being orthogonal or perpendicular to each other.
- These components are ordered so that the first few retain most of the variation present in the original data set. The first principal component (PC1) is defined as the direction that maximizes the variance of the projected data. We want maximal variation because more variation equals more information (If all of our values were all equal to 1, there would be no variation in the data set and thus, no information).
- The second principal component (PC2) is uncorrelated to PC1, and is the direction that maximizes the variance of the projected data with this constraint of orthogonality to PC1. This continues with every subsequent principal component, all needing to be orthogonal to the principal components before them.

The main steps in PCA include:

1) Standardize the data: Centering the data by subtracting every observation in column by the mean of that column, and scaling by dividing every observation in a column by the standard deviation of that column.

2) Calculate the correlation matrix: The goal of this step is to understand if there is any relationship between the variables. The matrix summarizes the correlations between all possible pairs of variables.

3) Compute the the principal components: Use the function prcomp() to compute the principal components. This function uses Singular Value Decomposition.

4) Visualize and interpret PCA results: There are many ways to do this, in our tutorial we will look at the rotation matrix, Scree plot, Biplot, bar graphs, and a scatter plot with concentration ellipses.

Load the Data

In this section, we load the Breast Cancer data set and take an initial look at its structure.

```
# Load the data
data <- read.csv("breast-cancer_small.csv")

# View the first few rows of the data
head(data)
```

```
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         M       17.99        10.38         122.80    1001.0
## 2   842517         M       20.57        17.77         132.90    1326.0
## 3 84300903         M       19.69        21.25         130.00    1203.0
## 4 84348301         M       11.42        20.38          77.58     386.1
## 5 84358402         M       20.29        14.34         135.10    1297.0
## 6   843786         M       12.45        15.70          82.57     477.1
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
```

```
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
## 4         0.14250          0.28390         0.2414             0.10520
## 5         0.10030          0.13280         0.1980             0.10430
## 6         0.12780          0.17000         0.1578             0.08089
##   symmetry_mean fractal_dimension_mean
## 1        0.2419                0.07871
## 2        0.1812                0.05667
## 3        0.2069                0.05999
## 4        0.2597                0.09744
## 5        0.1809                0.05883
## 6        0.2087                0.07613
```

```
# View the dimensions of the table (number of rows, number of columns)
dim(data)
```

```
## [1] 569  12
```

Data Preprocessing

Before performing PCA, it's important to preprocess the data. This involves handling any missing values, removing non-numeric columns, and the dependent variable if you have one. **PCA requires your data to only be predictors and be continuous numeric variables.** Below, we remove 5 missing values and the 2 columns with non-numeric/non-continuous values.

```
# Check for missing values
sum(is.na(data))
```

```
## [1] 5
```

```
# If there are any missing values, we need to remove them.
data <- na.omit(data)

# Remove non-numeric columns and IDs
data_numeric <- data %>% select_if(is.numeric) %>% select(-id)

# View the first few rows of the numeric data
head(data_numeric)
```

```
##   radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 1       17.99        10.38         122.80    1001.0         0.11840
## 2       20.57        17.77         132.90    1326.0         0.08474
## 3       19.69        21.25         130.00    1203.0         0.10960
## 4       11.42        20.38          77.58     386.1         0.14250
## 5       20.29        14.34         135.10    1297.0         0.10030
## 6       12.45        15.70          82.57     477.1         0.12780
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## 1          0.27760         0.3001              0.14710        0.2419
## 2          0.07864         0.0869              0.07017        0.1812
## 3          0.15990         0.1974              0.12790        0.2069
```

```
## 4          0.28390         0.2414             0.10520        0.2597
## 5          0.13280         0.1980             0.10430        0.1809
## 6          0.17000         0.1578             0.08089        0.2087
##   fractal_dimension_mean
## 1                0.07871
## 2                0.05667
## 3                0.05999
## 4                0.09744
## 5                0.05883
## 6                0.07613

# View the dimensions of the final table (number of rows, number of columns)
dim(data_numeric)

## [1] 564  10
```

Descriptive Exploration and Visualization

We explore the data set descriptively and create a professional-quality visual aid to better understand the distribution of key variables.

```
# Summary statistics
summary(data_numeric)

##   radius_mean      texture_mean   perimeter_mean     area_mean
##  Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  1st Qu.:11.697   1st Qu.:16.17   1st Qu.: 75.14   1st Qu.: 420.2
##  Median :13.375   Median :18.84   Median : 86.29   Median : 551.4
##  Mean   :14.138   Mean   :19.28   Mean   : 92.03   Mean   : 656.1
##  3rd Qu.:15.893   3rd Qu.:21.79   3rd Qu.:104.40   3rd Qu.: 789.7
##  Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##  smoothness_mean   compactness_mean  concavity_mean    concave.points_mean
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08604   1st Qu.:0.06431   1st Qu.:0.02940   1st Qu.:0.02030
##  Median :0.09582   Median :0.09235   Median :0.06140   Median :0.03338
##  Mean   :0.09627   Mean   :0.10411   Mean   :0.08869   Mean   :0.04885
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13100   3rd Qu.:0.07401
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
##  symmetry_mean    fractal_dimension_mean
##  Min.   :0.1060   Min.   :0.04996
##  1st Qu.:0.1619   1st Qu.:0.05769
##  Median :0.1792   Median :0.06152
##  Mean   :0.1811   Mean   :0.06277
##  3rd Qu.:0.1956   3rd Qu.:0.06613
##  Max.   :0.3040   Max.   :0.09744
```

We display a correlation matrix of all the dimensions. A correlation matrix shows the correlation coefficients between pairs of variables. These coefficients range from -1 to 1, where:

- 1 indicates a perfect positive correlation,

- -1 indicates a perfect negative correlation,
- 0 indicates no correlation.

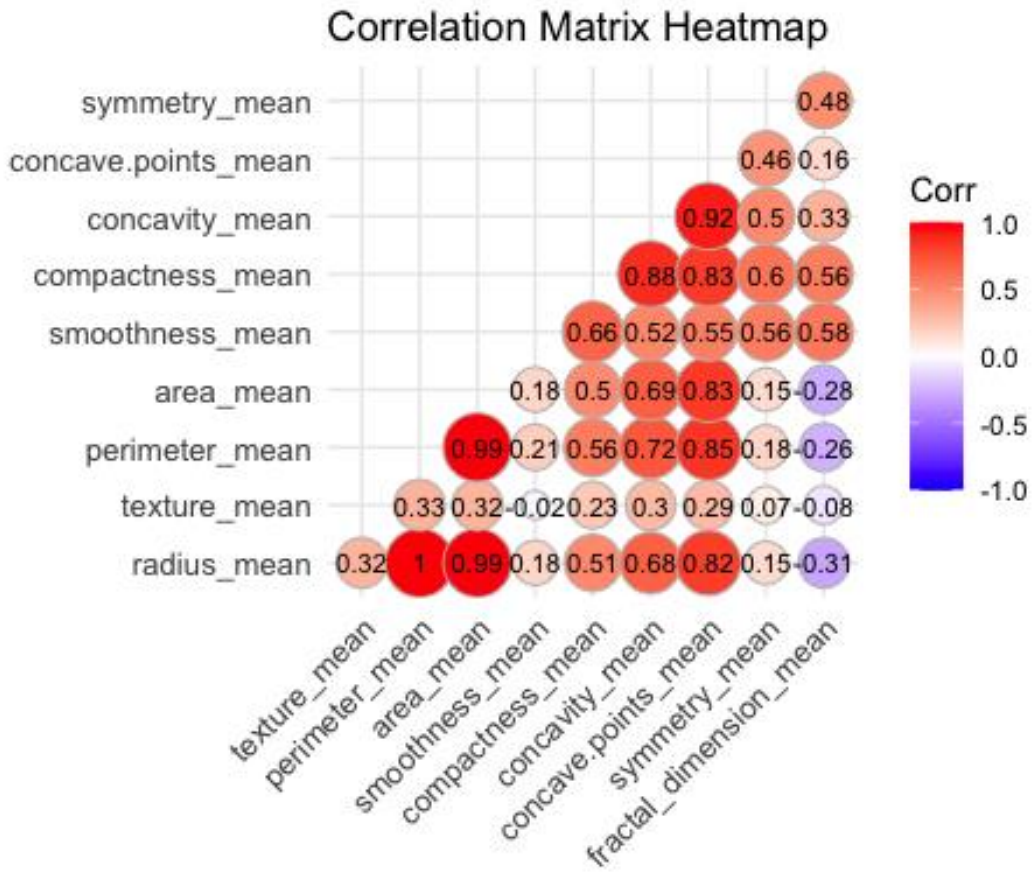
```r
# Creating the correlation matrix:
cor_matrix <- cor(data_numeric, use = "complete.obs")

# Print the first 5 rows and columns of the correlation matrix. The
correlation matrix has dimensions 10 by 10
print(cor_matrix[1:5, 1:5])
```

```
##                  radius_mean texture_mean perimeter_mean area_mean
## radius_mean        1.0000000   0.32389977      0.9978719 0.9873554
## texture_mean       0.3238998   1.00000000      0.3293948 0.3216360
## perimeter_mean     0.9978719   0.32939475      1.0000000 0.9865328
## area_mean          0.9873554   0.32163597      0.9865328 1.0000000
## smoothness_mean    0.1753074  -0.02274314      0.2117090 0.1817920
##                  smoothness_mean
## radius_mean           0.17530736
## texture_mean         -0.02274314
## perimeter_mean        0.21170904
## area_mean             0.18179200
## smoothness_mean       1.00000000
```

```r
# To see the whole correlation matrix, you may run:
# view(cor_matrix)


# Heat map of the correlation matrix:
ggcorrplot(cor_matrix,
           method = "circle", # Use circles to represent the correlations
           type = "lower", # Only show the lower triangle of the correlation
matrix
           lab = TRUE, # Display correlation coefficients on the heat map
           lab_size = 3,
           colors = c( "blue","white","red"),
           title = "Correlation Matrix Heatmap",
           tl.cex = 10, # Set the size of the axis text labels
           tl.srt = 45, # Rotate the axis text labels by 45 degrees
           ggtheme = theme_minimal())
```



Red areas show strong positive correlations:

- Size-related features: *radius_mean*, *perimeter_mean*, and *area_mean* are highly correlated (~0.99), indicating they increase together proportionally.
- Shape-related features: *concavity_mean*, *compactness_mean*, *concave.points_mean* are highly correlations as well (0.88-0.92), suggesting that these shape-related features tend to increase together.
- Size and shape related features: *concave.points_mean* is highly correlated with size-related features *area_mean*, *perimeter_mean*, and *radius_mean* (~0.80-0.87) suggesting that there may be a relationship between *concave.points_mean* and the size-related features.

The data set shows strong correlations among *size-related* and *shape-related* features, indicating that they can be grouped into fewer dimensions for analysis. This insight is valuable for techniques like Principal Component Analysis (PCA).

Note: Since *radius_mean*, *perimeter_mean*, and *area_mean* are so highly correlated, you would normally just choose 1 of these features to include in your analysis. This is a situation where you would likely consult the oncology expert you are working with to ask which is most important in this context. For the purposes of this tutorial, we will keep all features.

Perform PCA

We perform PCA using the `prcomp` function, which automatically centers and scales the data if specified.

```
# Perform PCA
pca <- prcomp(data_numeric, center = TRUE, scale. = TRUE)
# Center: Centers the data by subtracting each variable by the mean.
# Scale: Scales the data by dividing each variable by its standard deviation.
```

Interpreting the PCA Results

Summary of PCA

This gives us the standard deviations, proportion of variance, and cumulative proportion of variance for each principal component.

```
# Summary of PCA
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
PC7
## Standard deviation     2.3432 1.5848 0.93714 0.70490 0.61010 0.35249
0.2828
## Proportion of Variance 0.5491 0.2512 0.08782 0.04969 0.03722 0.01243
0.0080
## Cumulative Proportion  0.5491 0.8002 0.88805 0.93774 0.97497 0.98739
0.9954
##                            PC8    PC9    PC10
## Standard deviation     0.18659 0.1051 0.01671
## Proportion of Variance 0.00348 0.0011 0.00003
## Cumulative Proportion  0.99887 1.0000 1.00000
```

Interpretation of first two principal components:

- *PC1* has a standard deviation of 2.3432 and explains 54.91% of the variance in the data.
- *PC2* explains an additional 25.12% of the variance, bringing the cumulative proportion to 80.02%.

The first two components together explain about 80.02% of the total variance, suggesting that they capture the majority of the information in the data set. We can reduce the dimensionality of the data set while retaining most of the information by deciding which principal components to keep based on the cumulative proportion.

Principal Components (Scores) in PCA

In Principal Component Analysis (PCA), the principal components (scores) are the coordinates of the original data points in the new coordinate system defined by the

principal components. These scores represent the projection of the original data points onto the principal component axes - meaning they are the distance between each observation and the principal component direction, for each principal component.

In the mathematical representation of PC1 below, these scores are represented by their respective variable name such as $\left(\text{radius\_mean}_i - \overline{\text{radius\_mean}}\right)$, and multiplied by their respective loadings, $\phi_{1,1}$. We will talk about loadings in the next section.

Mathematical representation of PC1:

$$\begin{aligned} Z_{i,1} &= \phi_{1,1} \cdot \left(\text{radius\_mean}_i - \overline{\text{radius\_mean}}\right) + \phi_{2,1} \\ &\cdot \left(\text{texture\_mean}_i - \overline{\text{texture\_mean}}\right) + \ldots + \phi_{10,1} \\ &\cdot \left(\text{fractal\_dimension\_mean}_i - \overline{\text{fractal\_dimension\_mean}}\right) \end{aligned}$$

- $Z_{i,1}$ refers to the i^{th} observation for PC1.
- For PC1, R calculates $Z_{1,1}$ for the first observation, $Z_{2,1}$ for the 2nd observation, $Z_{3,1}$ for the 3rd observation and so forth.
- You can think of PC1 Z_1 as a single number summary for all observations of all variables in your data set related to the component direction.

The scores (pca$x) are typically stored in a matrix with the following structure:

- Rows: Each row corresponds to an observation (data point) in the original data set. Here, only 10 rows are displayed out of the 564 in the data set.
- Columns: Each column corresponds to a principal component. The number of columns is equal to the number of principal components retained in the analysis.
- These are often difficult to interpret with a number for every observation, so they are typically used for visualization.

Magnitude of Scores:

- High Scores (positive or negative) indicate significant deviation from the mean along that principal component.
- Low Scores (close to zero) indicate little deviation from the mean along that principal component.

Interpretation of Sign:

- Positive Scores mean that when the original observation's value increases, the principal component's value also increases.
- Negative Scores mean that when the original observation's value increases, the principal component's value decreases.

Significance of Principal Component Scores:

- Patterns: Similar scores on principal components indicate similar characteristics.
- Outliers/Clusters: Extreme scores highlight outliers or clusters.

- Dimensionality Reduction: Focus on a few principal components to simplify the data set.

The rotation matrix (or loadings)

The **rotation matrix (or loadings)** provides insights into how different features of the breast cancer data set contribute to the new dimensions (principal components) created by PCA. The columns of this matrix correspond to the principal components, and the rows correspond to each original variable.

The values in this matrix indicate how much each original variable contributes to the corresponding principal component - they define the direction (or rotation) of each principal component.

Mathematically:

$$\begin{aligned} Z_{i,1} \\ = \phi_{1,1} \cdot \left(\text{radius\_mean}_i - \overline{\text{radius\_mean}}\right) + \phi_{2,1} \\ \cdot \left(\text{texture\_mean}_i - \overline{\text{texture\_mean}}\right) + \ldots + \phi_{10,1} \\ \cdot \left(\text{fractal\_dimension\_mean}_i - \overline{\text{fractal\_dimension\_mean}}\right) \end{aligned}$$

- For radius_mean, PC1, $\phi_{1,1} = -0.36440813$
- For texture_mean, PC1, $\phi_{2,1} = -0.15323027$

PC1: High negative loadings for radius_mean, perimeter_mean, area_mean, compactness_mean, concavity_mean, and concave.points_mean, which indicates that PC1 likely represents a combination of tumor **size** and **shape** characteristics, as these variables significantly contribute to this component.

PC2: PC2 shows positive loadings for radius_mean, texture_mean, perimeter_mean, and area_mean, and negative loadings for smoothness_mean, compactness_mean, symmetry_mean, and fractal_dimension_mean. This suggests that PC2 captures variance related to the **texture** and **smoothness** of the tumor, highlighting an inverse relationship between size-related features and texture/smoothness.

The PCA rotation matrix helps determine which features are most important in explaining the variance in the data since it is the weight (ϕ) we are multiplying each feature's variance by. For instance, size and shape related features play a key role in PC1.

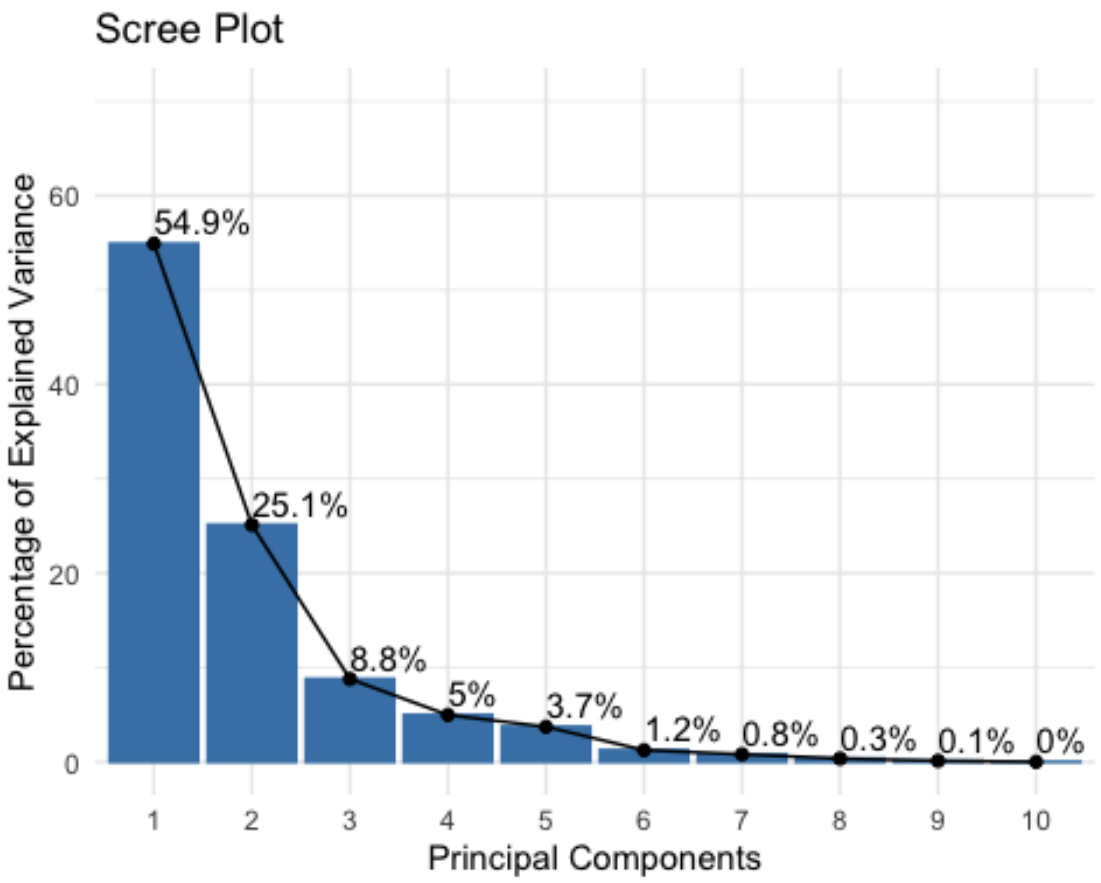
Visualizing PCA Results

Scree Plot

A scree plot helps visualize the proportion of total variance explained by each principal component. It assists in determining the number of components to retain.

```
# Scree plot (option 1)
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 70)) +
  labs(title = "Scree Plot",
```

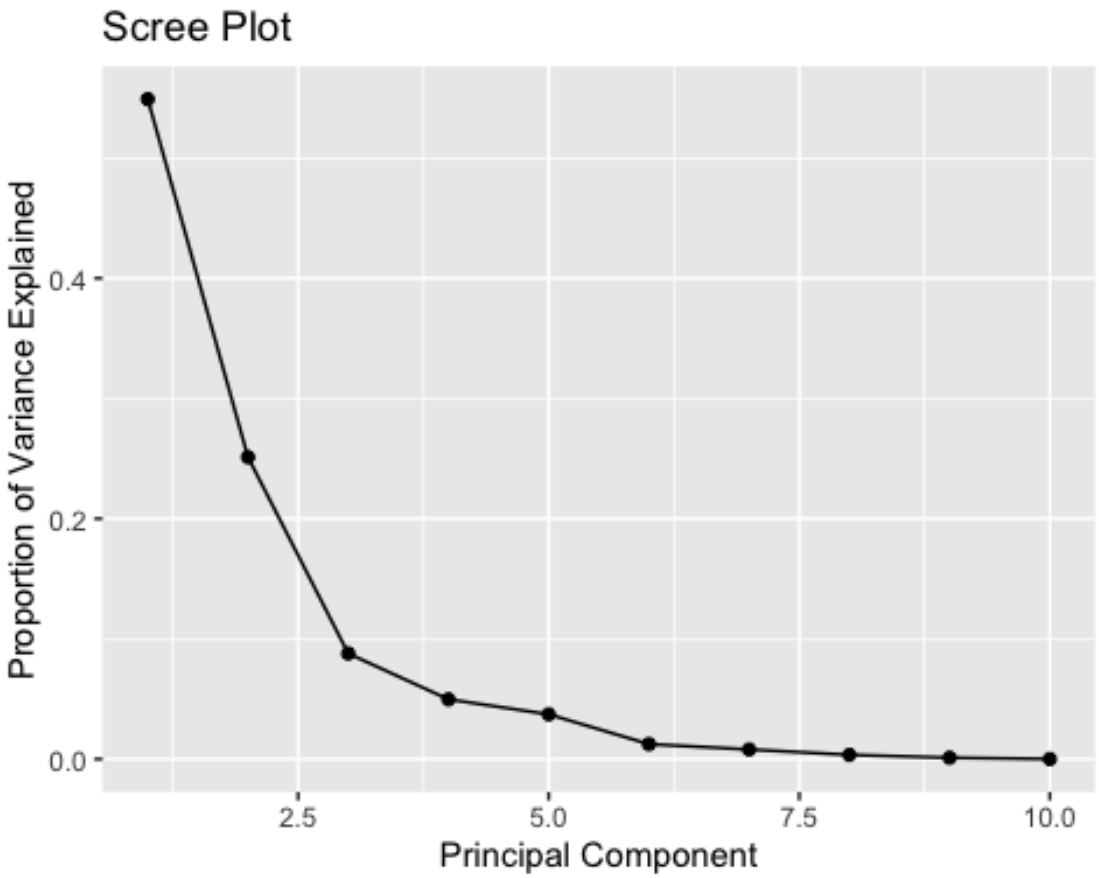
```
       x = "Principal Components",
       y = "Percentage of Explained Variance") +
  theme_minimal()
```




```
# Scree plot (option 2)

# First calculate the proportion of the total variance in the data that is
explained by each principal component in PCA:
# pca_SVD$sdev - the standard deviations of the principal components
variance_explained <- pca$sdev^2 / sum(pca$sdev^2)

qplot(y = variance_explained, x = 1:length(variance_explained), geom =
"line") +
  geom_point() +
  xlab("Principal Component") +
  ylab("Proportion of Variance Explained") +
  ggtitle("Scree Plot")
```



In the scree plot, we look for an "elbow" point where the explained variance starts to level off. We can decide to keep all the principal components including the one where the elbow point is, or just the principal components before the elbow point. The goal is to keep enough components to explain a large enough portion of the data's variance. This is a very subjective assessment and generally based on the context of the data.

In our scree plots, the first component explains about 55% of the variance and the second component explains about 25%. Together, they explain a significant portion (about 80%) of the total variance in the data set.

Visualization of variable contributions to components

Bar plots of variable contributions

These bar plots display the percentage each original variable contributes to PC1 and PC2.

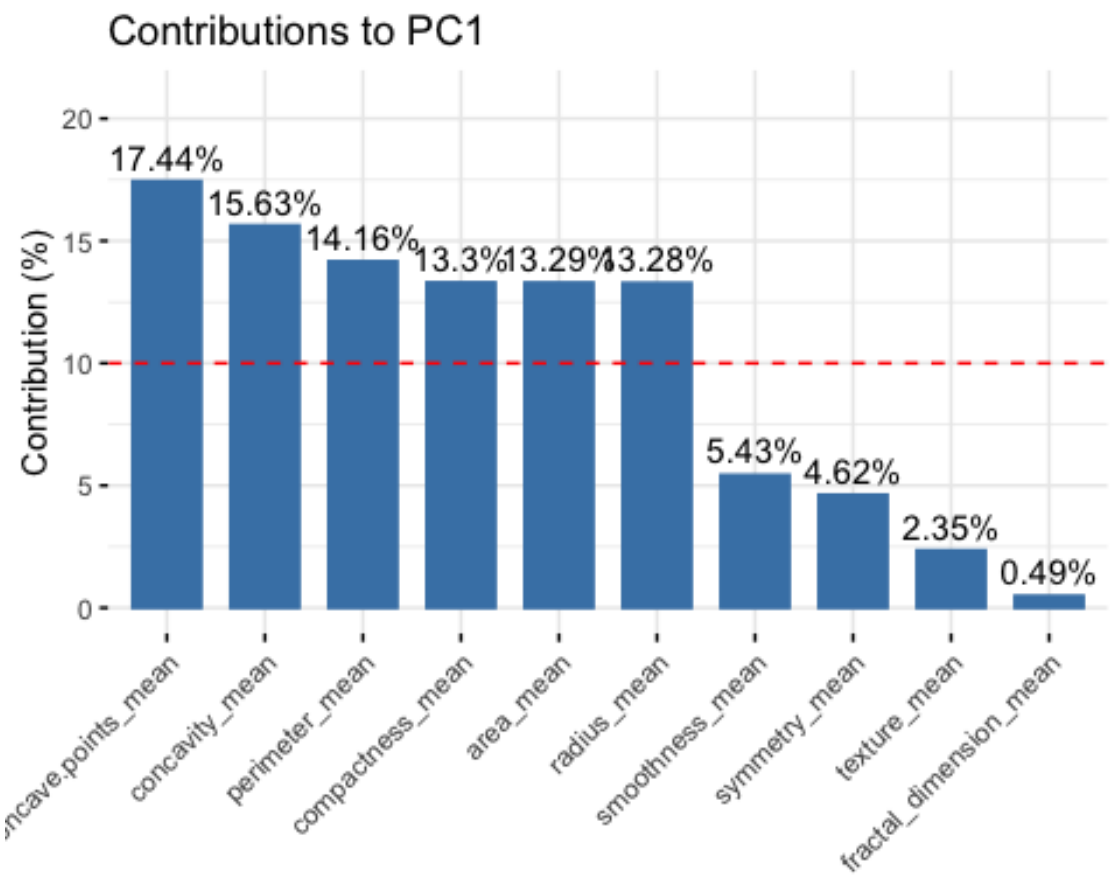
```
# Visualize contributions of variables to PC1 using fviz_contrib
fviz_contrib_pc1 <- fviz_contrib(pca, choice = "var", axes = 1)

# Extract data from the fviz_contrib plot
pc1_data <- fviz_contrib_pc1$data

# Calculate percentages
```

```r
pc1_data$percentage <- round(pc1_data$contrib / sum(pc1_data$contrib) * 100,
2)

# Add percentage labels to the bars
fviz_contrib_pc1 +
  geom_text(aes(label = paste0(pc1_data$percentage, "%")), vjust = -0.5) +
  ylim(0, max(pc1_data$contrib) * 1.2) +  # Increase y-axis limit to 120% of
the max value
  labs(title = "Contributions to PC1", y = "Contribution (%)")
```



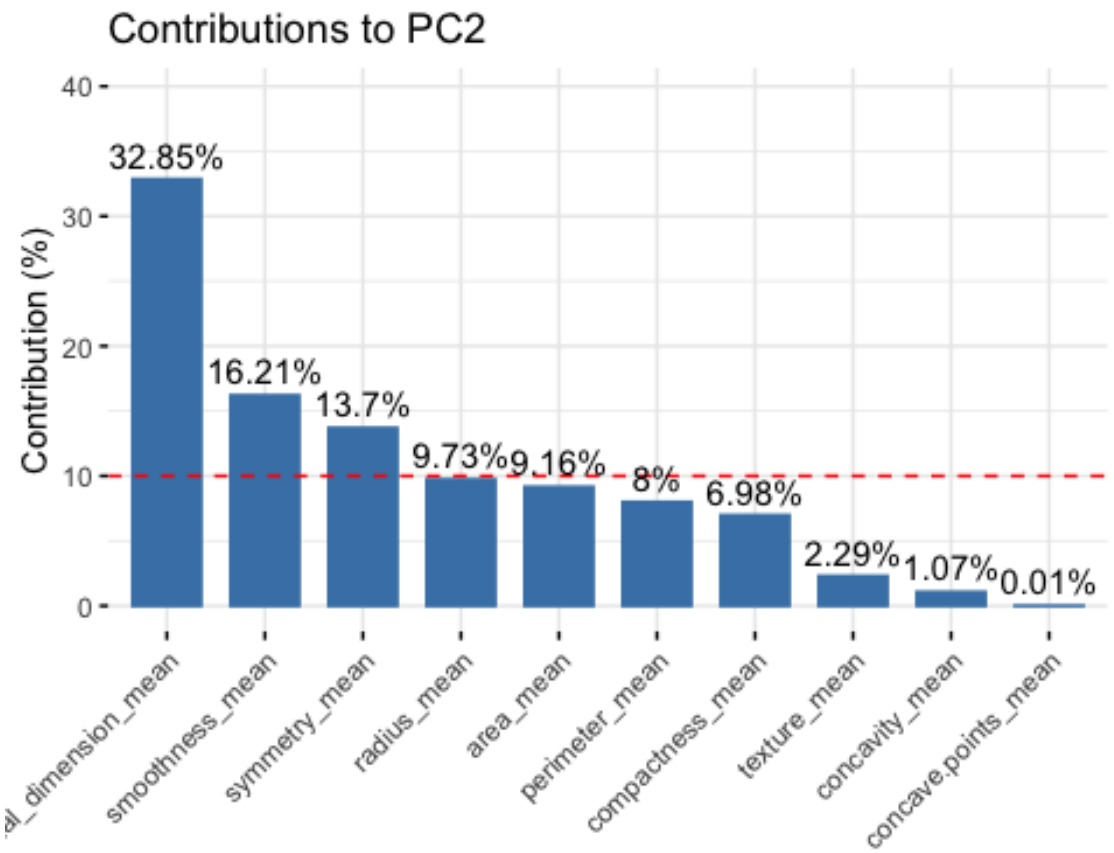

```r
# Visualize contributions of variables to PC2 using fviz_contrib
fviz_contrib_pc2 <- fviz_contrib(pca, choice = "var", axes = 2)

# Extract data from the fviz_contrib plot
pc2_data <- fviz_contrib_pc2$data

# Calculate percentages
pc2_data$percentage <- round(pc2_data$contrib / sum(pc2_data$contrib) * 100,
2)

# Add percentage labels to the bars
fviz_contrib_pc2 +
  geom_text(aes(label = paste0(pc2_data$percentage, "%")), vjust = -0.5) +
```

```
  ylim(0, max(pc2_data$contrib) * 1.2) +  # Increase y-axis limit to 120% of
the max value
  labs(title = "Contributions to PC2", y = "Contribution (%)")
```




```
# Bar plots if you don't want percentages above each bar:

# Bar plot for PC1
# fviz_contrib(pca, choice = 'var', axes = 1)

# Bar plot for PC2
# fviz_contrib(pca, choice = 'var', axes = 2)
```

Interpretation:

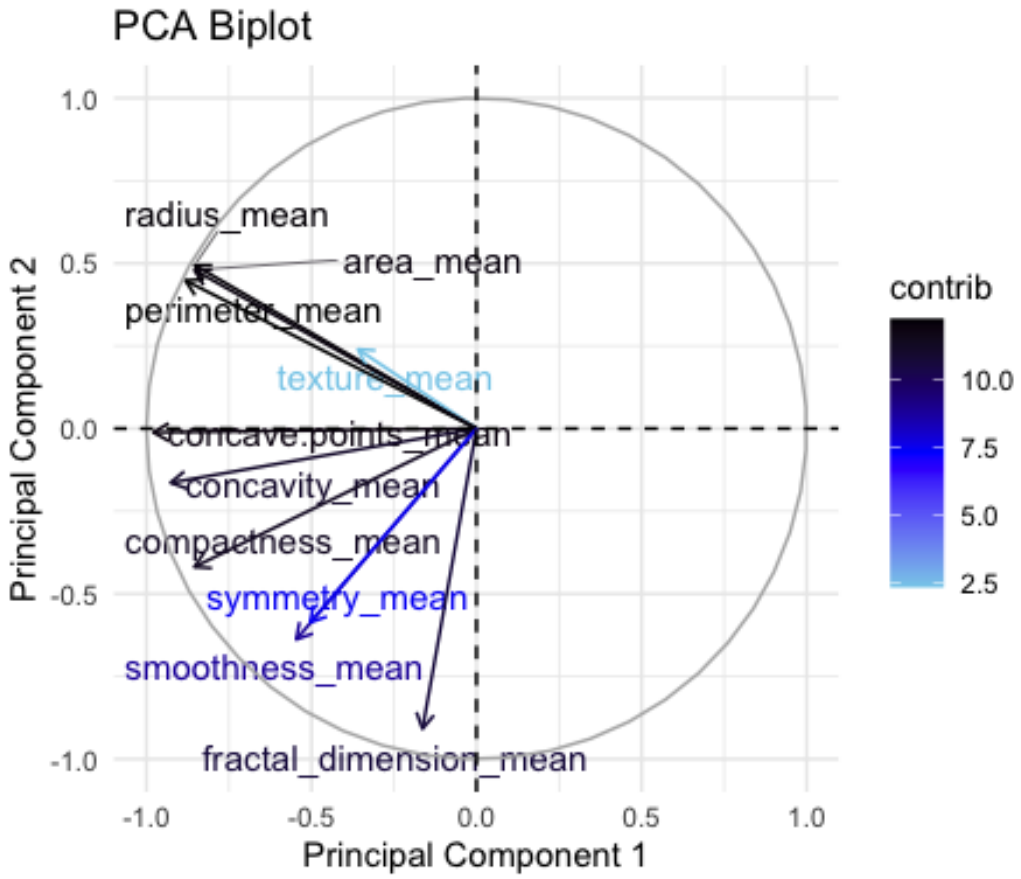
- The horizontal red line here shows the expected level of contribution if the contributions were uniform. This line is 10% in our plot because we have 10 variables. We will use this line to interpret which variables contribute more than average.
- concave.points_mean, concavity_mean, perimeter_mean, compactness_mean, area_mean, and radius_mean contribute the most to PC1.
- fractal_dimension_mean, smoothness_mean, and symmetry_mean contribute the most to PC1.

Biplot

A Biplot combines the **principal component scores** and the **loadings**, providing a visual representation of both the observations and variables. A Biplot is able to display all of the variables at once providing a more complete multidimensional comparison of the variables in our data set compared to the correlation matrix that was only able to compare pairs of variables.

A Biplot visualizes the similarities and dissimilarities between the variables and also shows the impact of each variable on each of the PCs.

```
# Biplot
fviz_pca_var(pca,
             col.var = "contrib", # Color the variables by their contribution
to the PCs
             gradient.cols = c("skyblue", "blue", "black"),  # Customize the
color gradient
             repel = TRUE) +  # Avoid text overlapping
  labs(title = "PCA Biplot",
       x = "Principal Component 1",
       y = "Principal Component 2") +
  theme_minimal()
```



This Biplot shows how the original variables relate to the first two principal components. The

Biplot has two main axes, each representing one of the principal components. The direction and length of the arrows indicate how each variable contributes to the components. Variables with longer arrows have a stronger influence on the principal components. Vectors that are close to each other have similar values.

The color gradient of the loading vectors is a joint measurement of how much each original variable contributes to both of the PCs.

In the breast cancer data set:

- **radius_mean, perimeter_mean, area_mean** (the size-related variables) have long arrows pointing in the same direction, indicating they contribute significantly to PC1 and are highly correlated with each other.
- **concave.points_mean, concavity_mean, and compactness_mean** have similar length arrows as the shape-related variables which indicate they contribute significantly to PC1.
- **fractal_dimension_mean** shows a significant contribution to PC2 and a negative correlation with size-related variables on PC2.
- *smoothness_mean* and *symmetry_mean* have shorter length arrows on the PC1 axis and indicating a moderate to low contribution to PC1, but a relatively longer arrow on the PC2 axis indicating a stronger contribution to PC2.

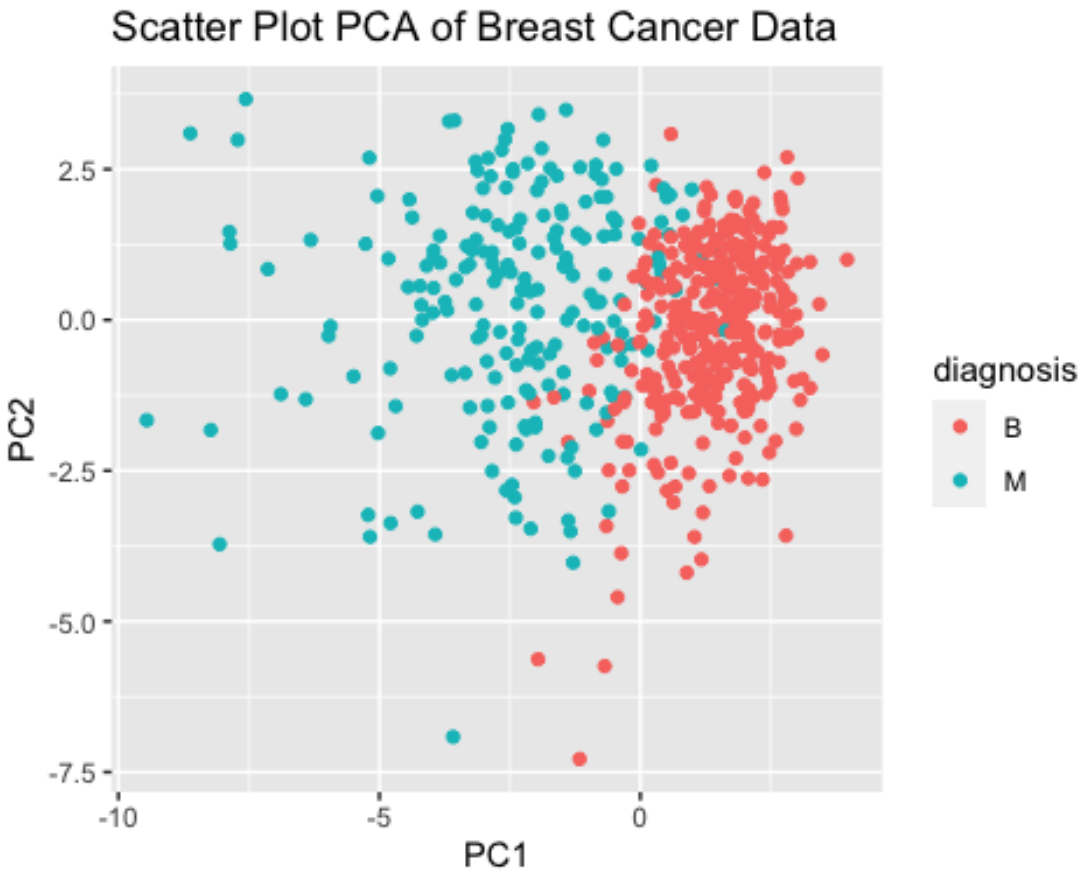
These observations are all consistent with the bar plots of variable contributions.

Scatter Plot of PC1 vs. PC2

We visualize the first two principal components and color the points by "diagnosis" variable to see how the first two principal components (PC1 and PC2) can separate the observations based on their diagnosis.

```
# Convert PCA results to Data Frame
data_pca <- as.data.frame(pca$x)
# Add "diagnosis" column (e.g., B for benign, M for malignant)
data_pca$diagnosis <- data$diagnosis

# Create a Scatter Plot for the first two principal components
ggplot(data_pca, aes(x = PC1, y = PC2, color=diagnosis)) +
  geom_point() +
  labs(title = "Scatter Plot PCA of Breast Cancer Data", x = "PC1", y =
"PC2")
```



Each point represents an observation from the breast cancer data set, plotted according to its scores on the first two principal components. Points are colored based on their diagnosis, allowing you to visually assess whether there is a clear separation between benign and malignant cases in the reduced dimensionality space. On this plot, red points represent benign cases, and blue points represent malignant cases.

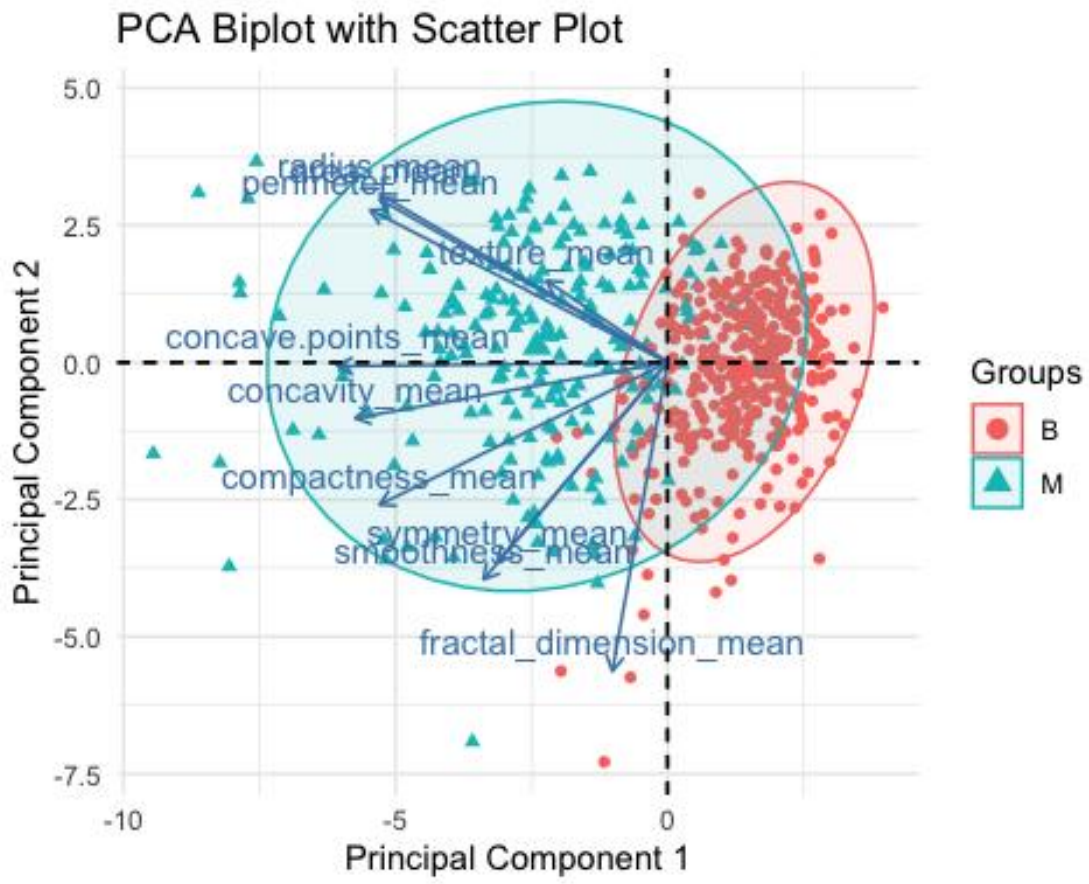
The scatter plot shows that the first two principal components provide a good separation between benign and malignant diagnoses (malignant and benign cases form separate clusters), indicating that PCA has effectively captured the variance related to the diagnosis in the data set.

PCA Biplot with Scatter Plot

This PCA Biplot includes both the variable vectors and a scatter plot of the individual observations, with points colored by diagnosis.

```
fviz_pca_biplot(pca,
                geom = "point",  # Use points to represent the individual
observations
                label = "var", # Label the variable vectors
                habillage = data$diagnosis, # Color the points by the
'diagnosis' column
                addEllipses = TRUE, # Add confidence ellipses for each group
```

```
                ellipse.level = 0.95) + # Set the confidence level for the
ellipses to 95%
  labs(title = "PCA Biplot with Scatter Plot",
       x = "Principal Component 1",
       y = "Principal Component 2") +
  theme_minimal()
```



Red circles (B) represent benign tumors. Blue triangles (M) represent malignant tumors. Ellipses show the 95% confidence intervals for each group (benign and malignant).

Conclusion

In this tutorial, we performed Principal Component Analysis (PCA) on the Breast Cancer data set. We went through data pre-processing, performed PCA, and visualized and interpreted the results.

The analysis provided the following results:

- Dimensionality Reduction: PCA reduced the data set to two principal components, capturing 80% of the total variance, simplifying the data without significant loss of information.

- Scree Plot Insights: The scree plot (fviz_eig or qplot) suggests retaining the first two components to capture most of the variance.
- Importance of Variables: Size and concavity related features dominate the first principal component, while texture and smoothness play significant roles in the second component, providing a comprehensive understanding of tumor characteristics.
- Separation of Tumor Types: The scatter plot (ggplot) of the first two principal components clearly separates benign (red circles) and malignant tumors (blue triangles). Benign tumors cluster on the left of PC1, while malignant tumors spread towards the right. However, this divide between benign and malignant tumors is still more of a horizontal line indicating that PC2 still aids in separating the tumor diagnoses.

Next steps:

- Since we have an outcome variable (diagnosis), a natural next step would be to perform Principal Component Regression (PCR) where we simply use the principal components as predictors in a regression model in place of the original larger set of variables. However, note that since the outcome variable was not involved in the unsupervised method of PCA to identify the principal components, a drawback to PCR is that there is no guarantee the principal components will best predict the response.
- Alternatively, since we explored both PC1 and PC2 and found interesting patterns in the data, then typically, we would continue to look at subsequent principal components until no further interesting patterns are found. This very subjective approach is reflective of the fact that PCA is generally used in the EDA step of data analysis and next steps are decided based on the patterns found from PCA.

Recommended Reading List

Resources for Learning PCA

1. An Introduction to Statistical Learning with applications in R by James, Witten, Hastie, Tibshirani. P. 498-510. E-text provided by Dr. Wendy.
2. A Tutorial on PCA Paper
3. StatQuest PCA Step-by-Step YouTube
4. VCU PCA and Optimization: A tutorial
5. PubMed PCA: a review and recent developments
6. Visually Explained - PCA YouTube
7. Statistics by Jim PCA

Resources for Conducting PCA in R

8. Datacamp
9. STHDA
10. Kaggle Nutrition PCA