Costa Stavrianidis Homework 3

In this homework, the objectives are to

1. Use R to examine and preprocess a dataset
2. Implement Unsupervised learning methods in a real-world scenario, including: Principal Component Analysis, Hierarchical Clustering, and K-means Clustering in R
3. Visualize and understand how to employ Principal Components, Hierarchical Clustering Dendrograms, and K-means Clustering in R

Please make sure to **print your knitted .html file into a pdf before you submit it to the Gradescope, and you may only submit your .rmd file to Sakai**.(Since Gradescope only allow you to upload pdf file, while sometimes students have problems in knitting pdf directly, hence please knit your rmd files as a html and print the html file as pdf.) **5 points will be deducted for every assignment submission that does not include either the RMarkdown file or the knitted html file.** Your code should be adequately commented to clearly explain the steps you used to produce the analyses. RMarkdown homework files should be uploaded to Sakai with the naming convention date_lastname_firstname_HW[X].Rmd. For example, my first homework assignment would be named 20220830_Dunn_Jessilyn_HW1.Rmd. **It is important to note that 5 points will be deducted for every assignment that is named improperly.** Please add your answer to each question directly after the question prompt in the homework .Rmd file template provided below.

```
library(tidyverse)
library(ggplot2)
library(lubridate)
library(patchwork)
library(gridExtra)
library(psych)
library(corrplot)
library(ggfortify)
library(factoextra)
```

Dataset

Breast Cancer Prediction from Cytopathology Data https://www.kaggle.com/code/gpreda/breast-cancer-prediction-from-cytopathology-data/data (https://www.kaggle.com/code/gpreda/breast-cancer-prediction-from-cytopathology-data/data)

Data Preparation (30 points)

1. Download the cancer data titled “Breast_Cytopatholgy.csv” from Sakai and import it into R. Look at the first 5 lines of the data to learn about the dataset. The “diagnosis” field shows whether the patient was diagnosed with a benign or malignant tumor. Please read additional information about each column online with the link above.

```
cancer <- read_csv("Breast_Cytopatholgy.csv")
```

```
## Rows: 569 Columns: 32
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): diagnosis
## dbl (31): id, radius_mean, texture_mean, perimeter_mean, area_mean, smoothne...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(cancer, n=5)
```

```
## # A tibble: 5 × 32
##       id diagn…¹ radiu…² textu…³ perim…⁴ area_…⁵ smoot…⁶ compa…⁷ conca…⁸ conca…⁹
##    <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 8.42e5 M          18.0    10.4   123.    1001   0.118   0.278   0.300   0.147
## 2 8.43e5 M          20.6    17.8   133.    1326   0.0847  0.0786  0.0869  0.0702
## 3 8.43e7 M          19.7    21.2   130     1203   0.110   0.160   0.197   0.128
## 4 8.43e7 M          11.4    20.4    77.6    386.  0.142   0.284   0.241   0.105
## 5 8.44e7 M          20.3    14.3   135.    1297   0.100   0.133   0.198   0.104
## # … with 22 more variables: symmetry_mean <dbl>, fractal_dimension_mean <dbl>,
## #   radius_se <dbl>, texture_se <dbl>, perimeter_se <dbl>, area_se <dbl>,
## #   smoothness_se <dbl>, compactness_se <dbl>, concavity_se <dbl>,
## #   `concave points_se` <dbl>, symmetry_se <dbl>, fractal_dimension_se <dbl>,
## #   radius_worst <dbl>, texture_worst <dbl>, perimeter_worst <dbl>,
## #   area_worst <dbl>, smoothness_worst <dbl>, compactness_worst <dbl>,
## #   concavity_worst <dbl>, `concave points_worst` <dbl>, …
```

2. Answer the following questions by using the summary function or other methods of your choice:

a. How many observations are there in total?

```
nrow(cancer)
```

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

There are 569.

b. How many independent variables are there?

```
ncol(cancer) - 2
```

```
## [1] 30
```

There are 30, because in the context of the problem, we are considering the “Diagnosis” variable to be dependent.

c. Is there any column with missing values? If yes, how many values are missing?

```
colSums(is.na(cancer))
```

```
##                      id               diagnosis             radius_mean
##                       0                       0                       0
##            texture_mean          perimeter_mean               area_mean
##                       0                       0                       0
##         smoothness_mean        compactness_mean          concavity_mean
##                       0                       0                       0
##     concave points_mean           symmetry_mean  fractal_dimension_mean
##                       0                       0                       6
##               radius_se              texture_se            perimeter_se
##                       0                       0                       0
##                 area_se           smoothness_se          compactness_se
##                       0                       0                       0
##            concavity_se       concave points_se             symmetry_se
##                       0                       0                       0
##    fractal_dimension_se             radius_worst           texture_worst
##                       0                       0                       0
##         perimeter_worst              area_worst        smoothness_worst
##                       0                       0                       0
##       compactness_worst         concavity_worst    concave points_worst
##                       0                       0                       0
##          symmetry_worst fractal_dimension_worst
##                       0                       0
```

Yes, there are 6 missing values in the ‘fractal_dimension_mean’ column.

d. How many observations are there with a malignant diagnosis and how many are there with a benign diagnosis?

```
sum(cancer$diagnosis == "M")
```

```
## [1] 212
```

```
sum(cancer$diagnosis == "B")
```

```
## [1] 357
```

212 malignant, 357 benign.

For this question, please type your answers in full sentences outside of R chunks. Do not just show the output of running your code.

3. Change the “id” column into the index column (i.e. turn the ID values into row names) and delete the “id” column. Use str() to display the resulting dataframe. (5 points)

```
rownames(cancer) <- cancer$id
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
cancer <- cancer %>% subset(select=-id)
```

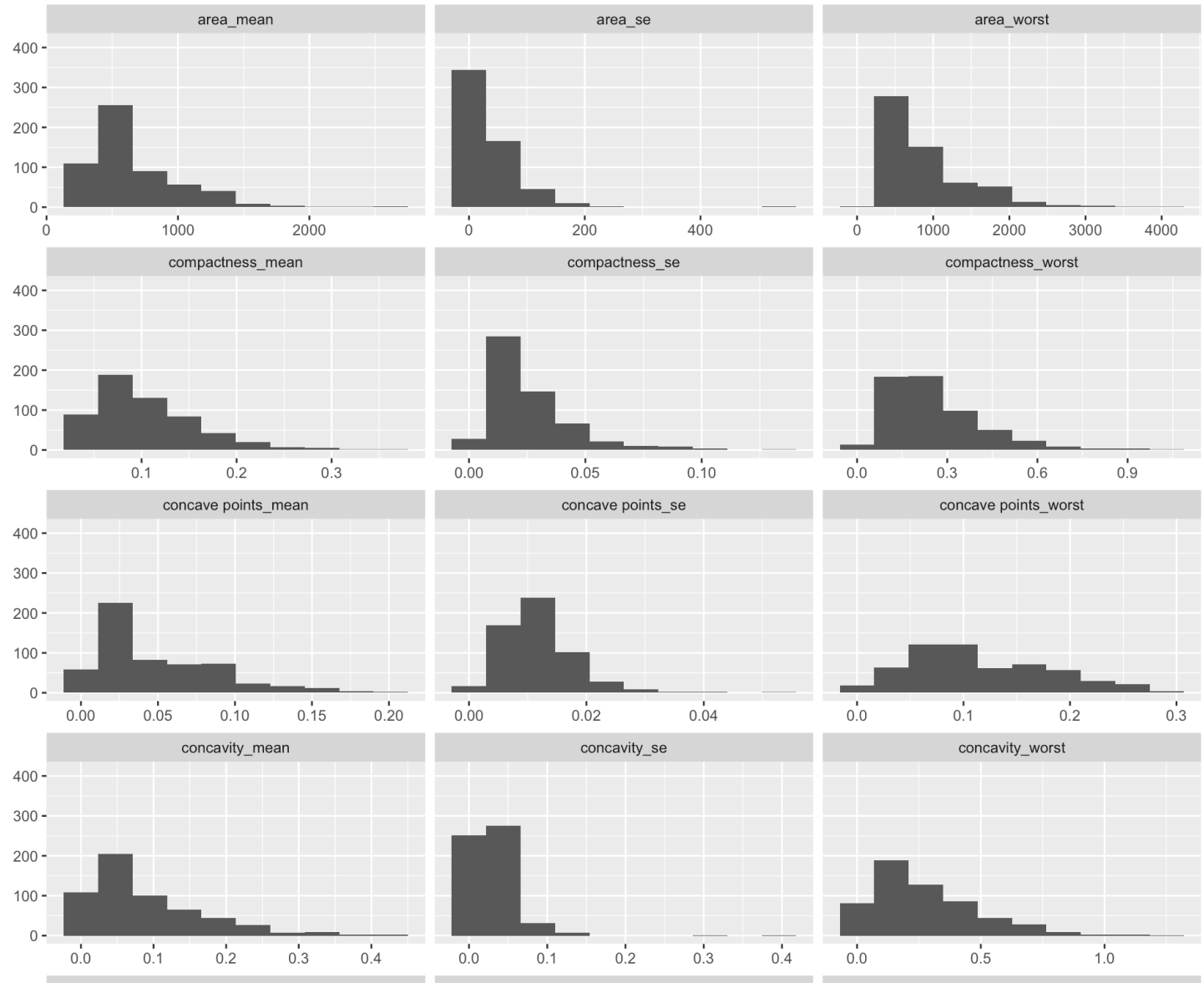
4. In this dataset, there isn’t any column with a very large number of missing values. For the column(s) with some missing values, let’s impute these missing values by mean substitution. Keep in mind that if it is reasonable to assume that the observations with missing values could have different distributions and characteristics for the two different diagnosis groups, imputation must be performed separately for the two different diagnosis groups.

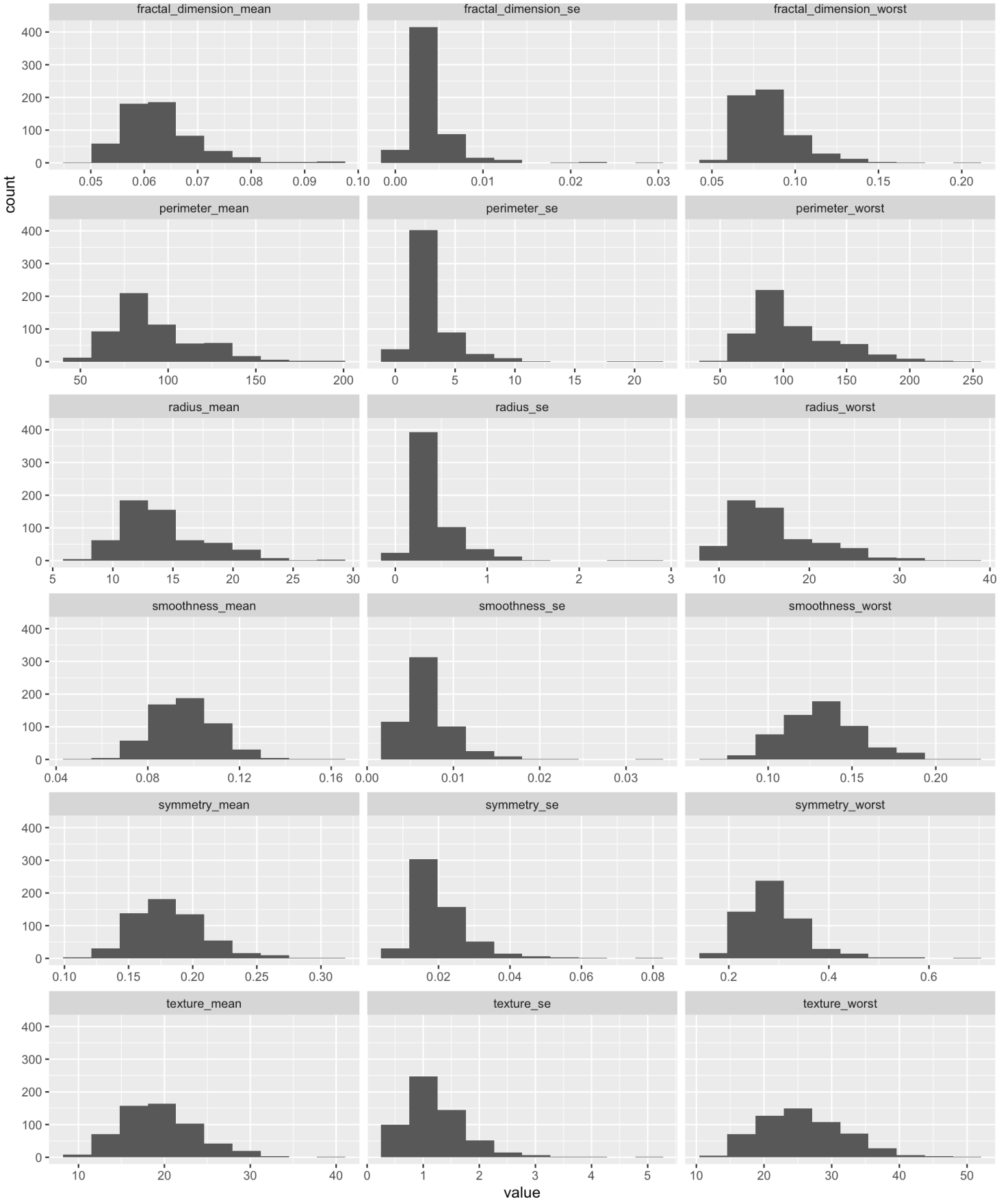
```
# Calculate means for different outcome groups
mean_m <- cancer %>% filter(diagnosis == "M" & !is.na(fractal_dimension_mean)) %>%
  pull(fractal_dimension_mean) %>%
  mean
mean_b <- cancer %>% filter(diagnosis == "B" & !is.na(fractal_dimension_mean)) %>%
  pull(fractal_dimension_mean) %>%
  mean

# Set empty values in variable to mean
cancer <- cancer %>%
  mutate(fractal_dimension_mean = ifelse(diagnosis == "M" & is.na(fractal_dimension_m
ean),
                                         mean_m, fractal_dimension_mean)) %>%
  mutate(fractal_dimension_mean = ifelse(diagnosis == "B" & is.na(fractal_dimension_m
ean),
                                         mean_b, fractal_dimension_mean))

sum(is.na(cancer$fractal_dimension_mean))
```

```
## [1] 0
```

5. After imputation, use “ggplot” and “facet_wrap” to plot a 10 x 3 grid of histograms to explore the data shape and distribution of all the independent variables in this dataset. The dataset has 10 sets of independent variables, and each set consists of the mean, standard error and worst value of a particular cell measurement. For example, “area_se” is the standard error of area measurements from a particular patient in this study. Remember to select a reasonable number of bins when plotting and add legends and labels when appropriate. Adjust the size of the plot display so that you can see all the facets clearly when you knit.

```
cancer_plot <- cancer %>% subset(select=-diagnosis) %>% gather()
plot1 <- ggplot(cancer_plot, aes(value)) + geom_histogram(bins = 10) +
  facet_wrap(~key, scales = 'free_x', ncol = 3)
plot1
```





6. If you observe the independent variable distributions closely, groups of variables that start with “area”,

"compactness" and "concavity" are consistently strongly skewed to the right. Apply log transform using formula $log(x + 1)$ to these 9 variables.

```
# Create function for log transformation
scale1 <- function(x) (log(x+1))

# Apply function to 9 variables
cancer <- cancer %>% mutate_at(c("area_mean", "compactness_mean", "concavity_mean",
                                 "area_se", "compactness_se", "concavity_se",
                                 "area_worst", "compactness_worst", "concavity_worst"
),
                               scale1)
```

7. The pre-processed dataset needs to be scaled before performing PCA. Can you give a brief explanation as to why that is the case? Standardize the dataset. Use summary() again to show that your dataset has been properly standardized by checking the means and range of values of the variables.

```
# Create function for standardization
scale2 <- function(x) ((x - mean(x)) / sd(x))

# Apply function to every independent variable
cancer <- cancer %>% mutate_at(vars(-("diagnosis")), scale2)
summary(cancer)
```

```
##   diagnosis          radius_mean       texture_mean     perimeter_mean
##  Length:569         Min.   :-2.0279   Min.   :-2.2273   Min.   :-1.9828
##  Class :character   1st Qu.:-0.6888   1st Qu.:-0.7253   1st Qu.:-0.6913
##  Mode  :character   Median :-0.2149   Median :-0.1045   Median :-0.2358
##                     Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##                     3rd Qu.: 0.4690   3rd Qu.: 0.5837   3rd Qu.: 0.4992
##                     Max.   : 3.9678   Max.   : 4.6478   Max.   : 3.9726
##    area_mean       smoothness_mean    compactness_mean  concavity_mean
##  Min.   :-2.8860   Min.   :-3.10935   Min.   :-1.6925   Min.   :-1.1774
##  1st Qu.:-0.6672   1st Qu.:-0.71034   1st Qu.:-0.7556   1st Qu.:-0.7619
##  Median :-0.1065   Median :-0.03486   Median :-0.2049   Median :-0.3256
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.6198   3rd Qu.: 0.63564   3rd Qu.: 0.5237   3rd Qu.: 0.5746
##  Max.   : 3.0268   Max.   : 4.76672   Max.   : 4.2564   Max.   : 3.8920
##  concave points_mean symmetry_mean      fractal_dimension_mean
##  Min.   :-1.2607     Min.   :-2.74171   Min.   :-1.8259
##  1st Qu.:-0.7373     1st Qu.:-0.70262   1st Qu.:-0.7205
##  Median :-0.3974     Median :-0.07156   Median :-0.1620
##  Mean   : 0.0000     Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.6464     3rd Qu.: 0.53031   3rd Qu.: 0.4735
##  Max.   : 3.9245     Max.   : 4.48081   Max.   : 4.9554
```

```
##    radius_se         texture_se       perimeter_se        area_se
##  Min.   :-1.0590   Min.   :-1.5529   Min.   :-1.0431   Min.   :-1.9362
##  1st Qu.:-0.6230   1st Qu.:-0.6942   1st Qu.:-0.6232   1st Qu.:-0.6865
##  Median :-0.2920   Median :-0.1973   Median :-0.2864   Median :-0.2568
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2659   3rd Qu.: 0.4661   3rd Qu.: 0.2428   3rd Qu.: 0.5832
##  Max.   : 8.8991   Max.   : 6.6494   Max.   : 9.4537   Max.   : 4.0749
##  smoothness_se     compactness_se     concavity_se     concave points_se
##  Min.   :-1.7745   Min.   :-1.3241   Min.   :-1.1284   Min.   :-1.9118
##  1st Qu.:-0.6235   1st Qu.:-0.6989   1st Qu.:-0.5833   1st Qu.:-0.6739
##  Median :-0.2201   Median :-0.2773   Median :-0.1981   Median :-0.1404
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3680   3rd Qu.: 0.4028   3rd Qu.: 0.3708   3rd Qu.: 0.4722
##  Max.   : 8.0229   Max.   : 5.9323   Max.   :11.0139   Max.   : 6.6438
##   symmetry_se      fractal_dimension_se  radius_worst     texture_worst
##  Min.   :-1.5315   Min.   :-1.0960      Min.   :-1.7254   Min.   :-2.22204
##  1st Qu.:-0.6511   1st Qu.:-0.5846      1st Qu.:-0.6743   1st Qu.:-0.74797
##  Median :-0.2192   Median :-0.2297      Median :-0.2688   Median :-0.04348
##  Mean   : 0.0000   Mean   : 0.0000      Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.3554   3rd Qu.: 0.2884      3rd Qu.: 0.5216   3rd Qu.: 0.65776
##  Max.   : 7.0657   Max.   : 9.8429      Max.   : 4.0906   Max.   : 3.88249
##  perimeter_worst     area_worst      smoothness_worst  compactness_worst
##  Min.   :-1.6919   Min.   :-2.5092   Min.   :-2.6803   Min.   :-1.6394
##  1st Qu.:-0.6890   1st Qu.:-0.6689   1st Qu.:-0.6906   1st Qu.:-0.6990
##  Median :-0.2857   Median :-0.1521   Median :-0.0468   Median :-0.2316
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.5398   3rd Qu.: 0.6712   3rd Qu.: 0.5970   3rd Qu.: 0.6186
##  Max.   : 4.2836   Max.   : 3.1371   Max.   : 3.9519   Max.   : 4.2793
##  concavity_worst   concave points_worst symmetry_worst
##  Min.   :-1.4782   Min.   :-1.7435      Min.   :-2.1591
##  1st Qu.:-0.7766   1st Qu.:-0.7557      1st Qu.:-0.6413
##  Median :-0.1557   Median :-0.2233      Median :-0.1273
##  Mean   : 0.0000   Mean   : 0.0000      Mean   : 0.0000
##  3rd Qu.: 0.6201   3rd Qu.: 0.7119      3rd Qu.: 0.4497
##  Max.   : 3.7763   Max.   : 2.6835      Max.   : 6.0407
##  fractal_dimension_worst
##  Min.   :-1.6004
##  1st Qu.:-0.6913
##  Median :-0.2163
##  Mean   : 0.0000
##  3rd Qu.: 0.4504
##  Max.   : 6.8408
```

PCA is looking for the sequence of linear combinations of the variables that have maximal variance. Since it is trying to maximize variance, the variables will have different variances depending on their individual scales. If you change one variable's scale from kg to g, it will then have more variance. Since the scale clearly matters for PCA, we must standardize the different variables to put them on the same scale beforehand.

PCA (25 points)

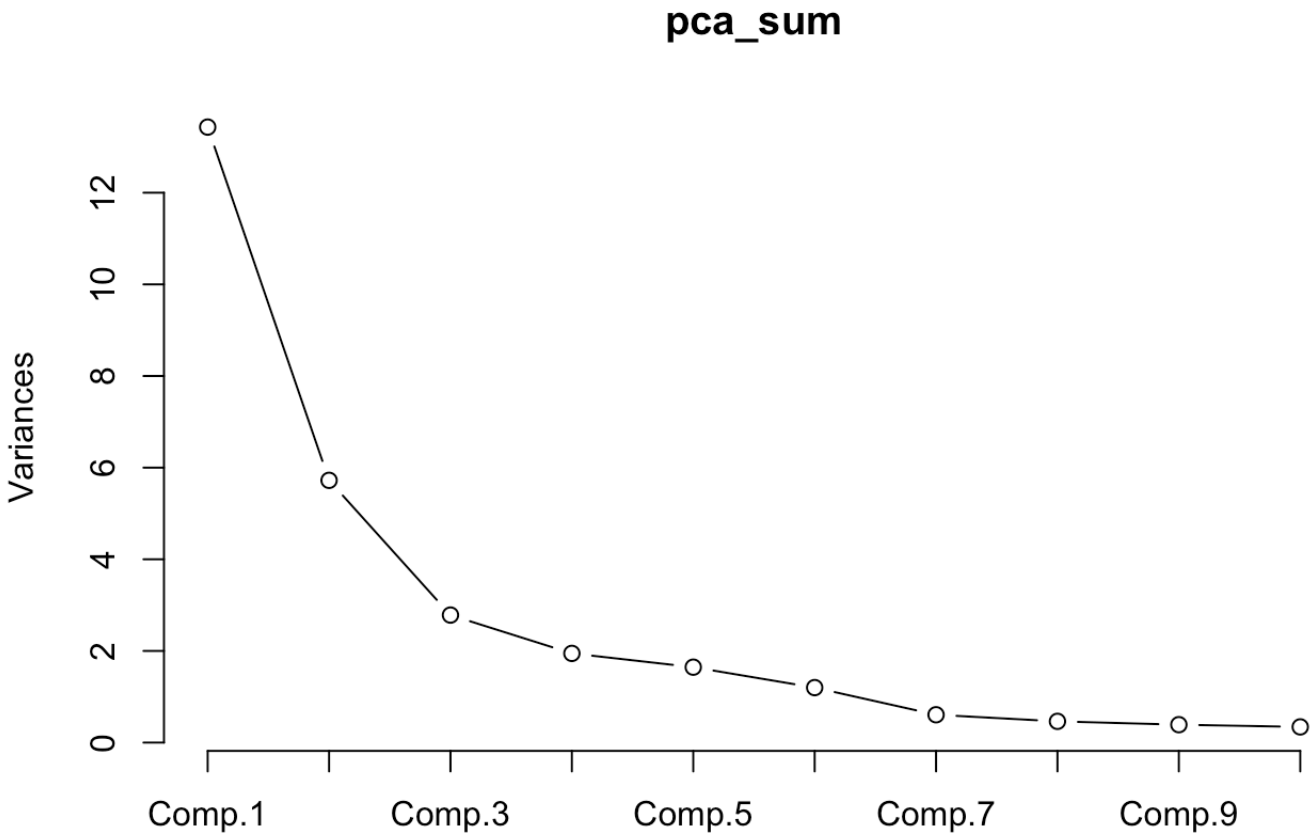
8. Calculate the principal components using the function princomp() and print the summary of the results.

```
pca_cancer <- princomp(cancer[,-1])
pca_sum <- summary(pca_cancer)
pca_sum
```

```
## Importance of components:
##                            Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     3.6649813 2.3922560 1.66811749 1.39504520 1.28282751
## Proportion of Variance 0.4485245 0.1910988 0.09291716 0.06498591 0.05495146
## Cumulative Proportion  0.4485245 0.6396234 0.73254052 0.79752643 0.85247789
##                            Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     1.09535614 0.77911399 0.68072240 0.62488147 0.58480539
## Proportion of Variance 0.04006391 0.02026958 0.01547329 0.01303881 0.01141998
## Cumulative Proportion  0.89254180 0.91281138 0.92828467 0.94132348 0.95274347
##                           Comp.11     Comp.12     Comp.13     Comp.14
## Standard deviation     0.527120745 0.496992966 0.480676398 0.401643553
## Proportion of Variance 0.009278182 0.008247896 0.007715219 0.005386718
## Cumulative Proportion  0.962021648 0.970269544 0.977984763 0.983371481
##                           Comp.15     Comp.16     Comp.17     Comp.18
## Standard deviation     0.300490558 0.289969850 0.268537964 0.235037574
## Proportion of Variance 0.003015118 0.002807685 0.002407987 0.001844664
## Cumulative Proportion  0.986386600 0.989194284 0.991602271 0.993446935
##                           Comp.19     Comp.20     Comp.21      Comp.22
## Standard deviation     0.193208881 0.179478677 0.174145056 0.1639589564
## Proportion of Variance 0.001246513 0.001075644 0.001012663 0.0008976623
## Cumulative Proportion  0.994693448 0.995769092 0.996781755 0.9976794170
##                            Comp.23      Comp.24      Comp.25      Comp.26
## Standard deviation     0.1475830310 0.1210596744 0.1114839259 0.1055347689
## Proportion of Variance 0.0007273032 0.0004893749 0.0004150182 0.0003719065
## Cumulative Proportion  0.9984067203 0.9988960952 0.9993111134 0.9996830199
##                            Comp.27      Comp.28      Comp.29      Comp.30
## Standard deviation     0.0840075545 4.101507e-02 2.477457e-02 1.180696e-02
## Proportion of Variance 0.0002356565 5.617327e-05 2.049533e-05 4.654990e-06
## Cumulative Proportion  0.9999186764 9.999748e-01 9.999953e-01 1.000000e+00
```

9. Plot a scree plot using the screeplot() function.

```
screeplot(pca_sum, type = 'lines')
```

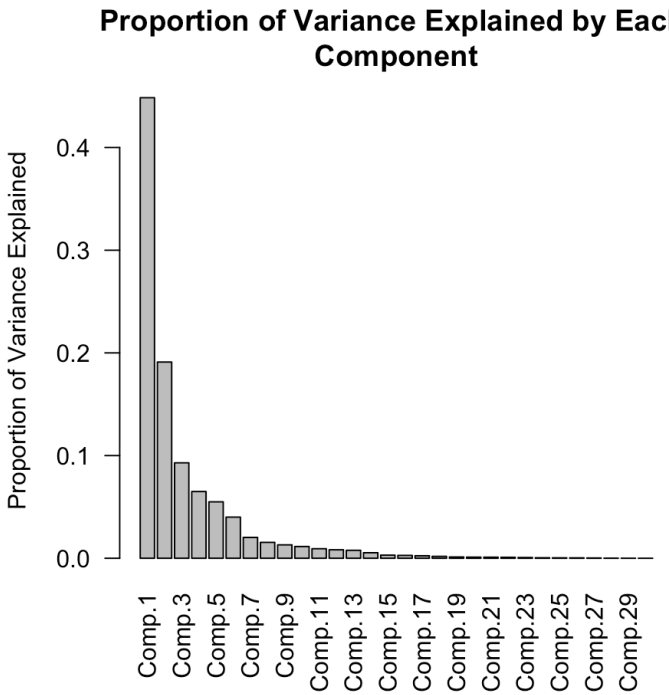


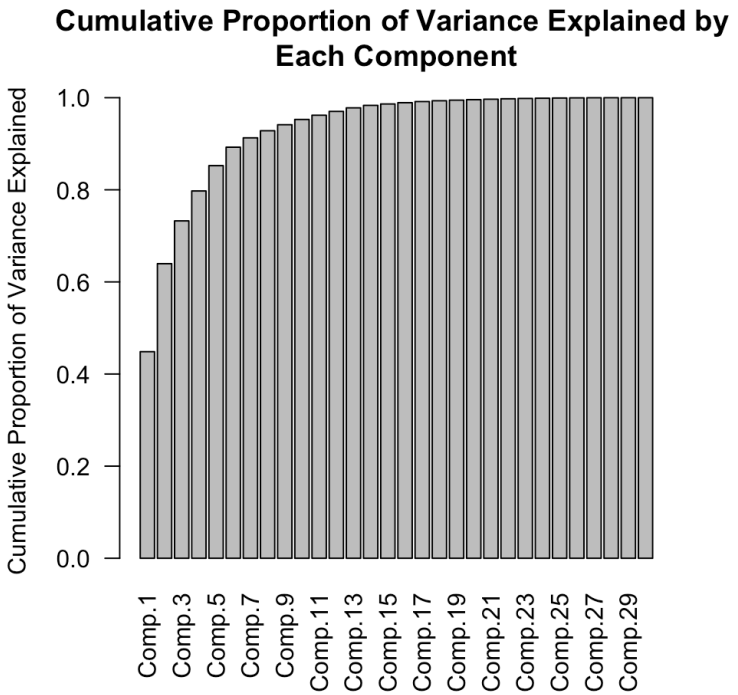
10. Plot the following two plots and use patchwork/gridExtra to position the two plots side by side:

    a. proportion of variance explained by the number of principal components
    b. cumulative proportion of variance explained by the number of principal components; draw horizontal lines at 88% of variance and 95% variance.

Note: please remember to clearly label your plots with titles, axis labels and legends when appropriate.

```
# Calculate proportion of variance for each component
pov <- pca_sum$sdev^2/sum(pca_sum$sdev^2)

# Calculate cumulative proportion of variance for each component
pov_cum <- cumsum(pov)

# Plot barplots of proportions for each component
par(mfrow=c(1,2))
barplot(pov, ylab = "Proportion of Variance Explained",
        main = "Proportion of Variance Explained by Each \nComponent", las = 2)
barplot(pov_cum, ylab = "Cumulative Proportion of Variance Explained",
        main = "Cumulative Proportion of Variance Explained by \nEach Component", las
= 2)
```





11. What proportions of variance are captured from the first, second and third principal components? How many principal components do you need to describe at least 88% and 95% of the variance, respectively?

```
print(c(pov[1], pov[2], pov[3]))
```

```
##     Comp.1     Comp.2     Comp.3
## 0.44852454 0.19109881 0.09291716
```

```
pov_cum[pov_cum>.88]
```

```
##    Comp.6    Comp.7    Comp.8    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13
## 0.8925418 0.9128114 0.9282847 0.9413235 0.9527435 0.9620216 0.9702695 0.9779848
##   Comp.14   Comp.15   Comp.16   Comp.17   Comp.18   Comp.19   Comp.20   Comp.21
## 0.9833715 0.9863866 0.9891943 0.9916023 0.9934469 0.9946934 0.9957691 0.9967818
##   Comp.22   Comp.23   Comp.24   Comp.25   Comp.26   Comp.27   Comp.28   Comp.29
## 0.9976794 0.9984067 0.9988961 0.9993111 0.9996830 0.9999187 0.9999748 0.9999953
##   Comp.30
## 1.0000000
```

```
pov_cum[pov_cum>.95]
```

```
##   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14   Comp.15   Comp.16   Comp.17
## 0.9527435 0.9620216 0.9702695 0.9779848 0.9833715 0.9863866 0.9891943 0.9916023
##   Comp.18   Comp.19   Comp.20   Comp.21   Comp.22   Comp.23   Comp.24   Comp.25
## 0.9934469 0.9946934 0.9957691 0.9967818 0.9976794 0.9984067 0.9988961 0.9993111
##   Comp.26   Comp.27   Comp.28   Comp.29   Comp.30
## 0.9996830 0.9999187 0.9999748 0.9999953 1.0000000
```

The first, second, and third components capture 0.44852454, 0.19109881, and 0.09291716 of the variance, respectively.

You need 6 principal components to describe at least 88% of the variance, and 10 to capture at least 95% of the variance.

12. Which are the top 2 variables that contribute the most to the variance captured from PC1, PC2, and PC3 respectively? (hint: look at the loadings information)

```
sort(abs(pca_sum$loadings[,1]), decreasing = T)[1:2]
```

```
## concave points_mean      concavity_mean
##           0.2594154           0.2570949
```

```
sort(abs(pca_sum$loadings[,2]), decreasing = T)[1:2]
```

```
## fractal_dimension_mean   fractal_dimension_se
##              0.3687653              0.2841809
```

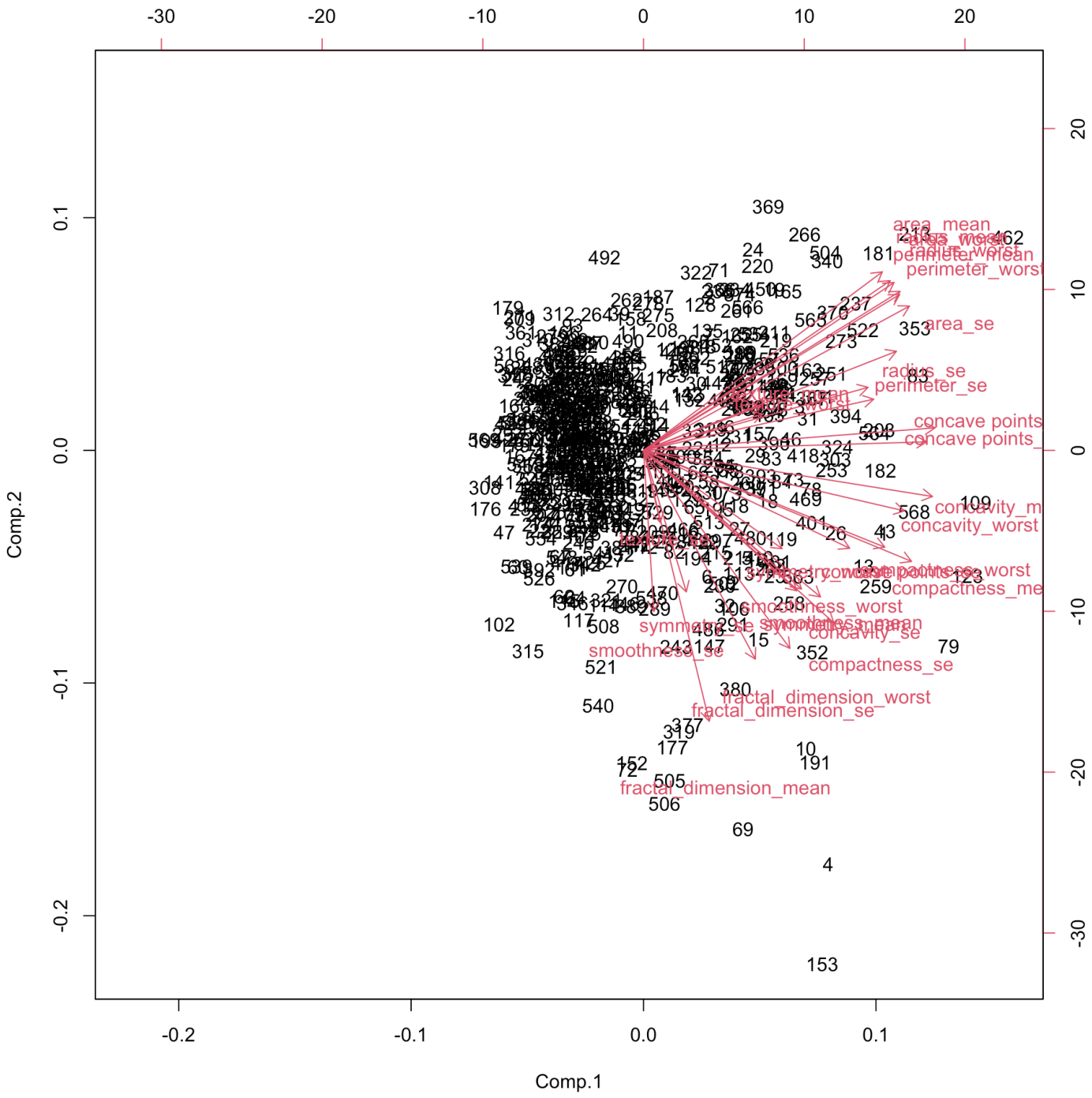
```
sort(abs(pca_sum$loadings[,3]), decreasing = T)[1:2]
```

```
##    texture_se smoothness_se
##     0.3936197     0.2929809
```

The top 2 variables for PC1 are concave points_mean and concavity_mean. The top 2 variables for PC2 are fractal_dimension_mean and fractal_dimension_se. The top 2 variables for PC3 are texture_se and smoothness_se.

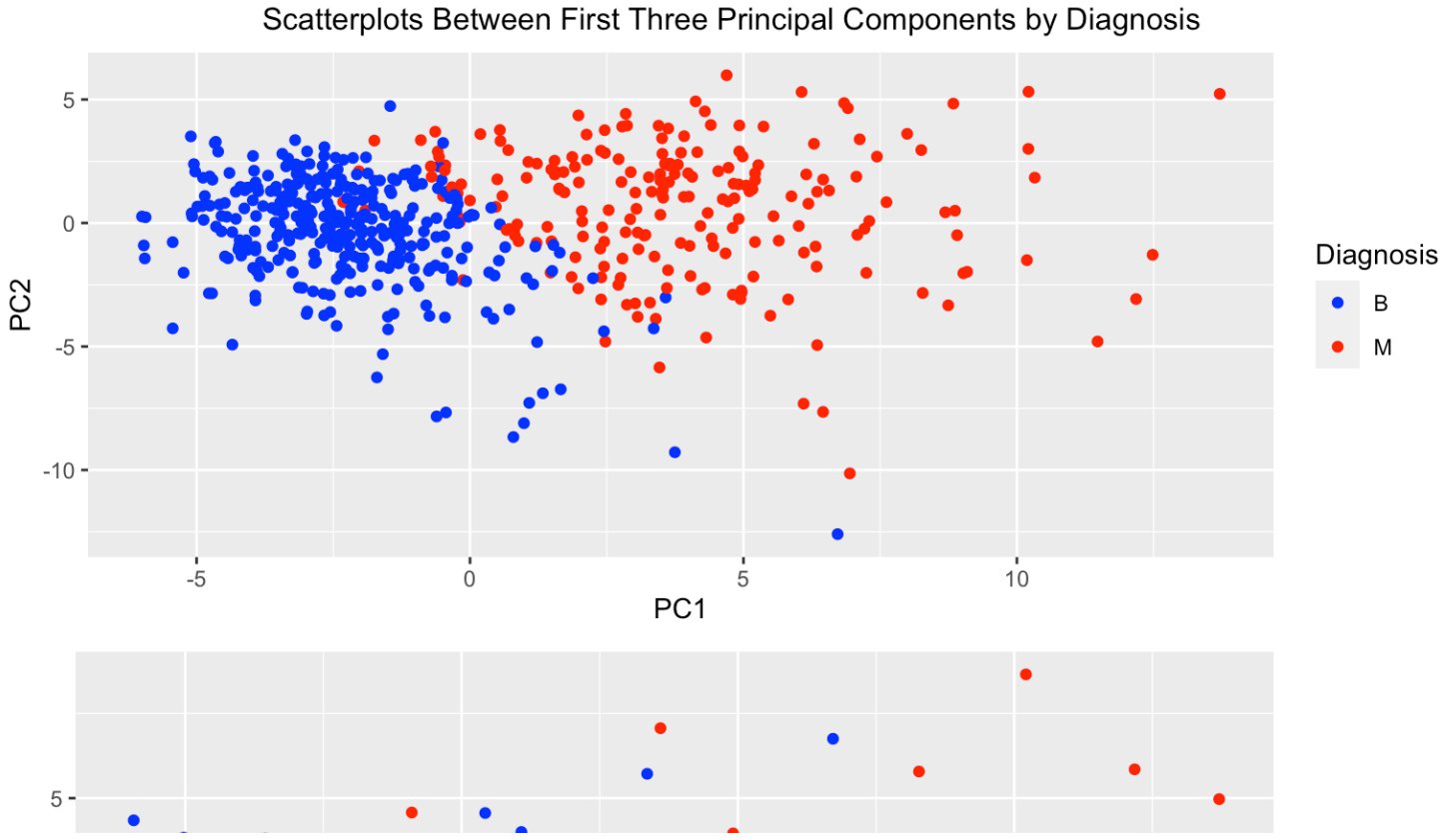
13. Plot a biplot using the biplot() function.

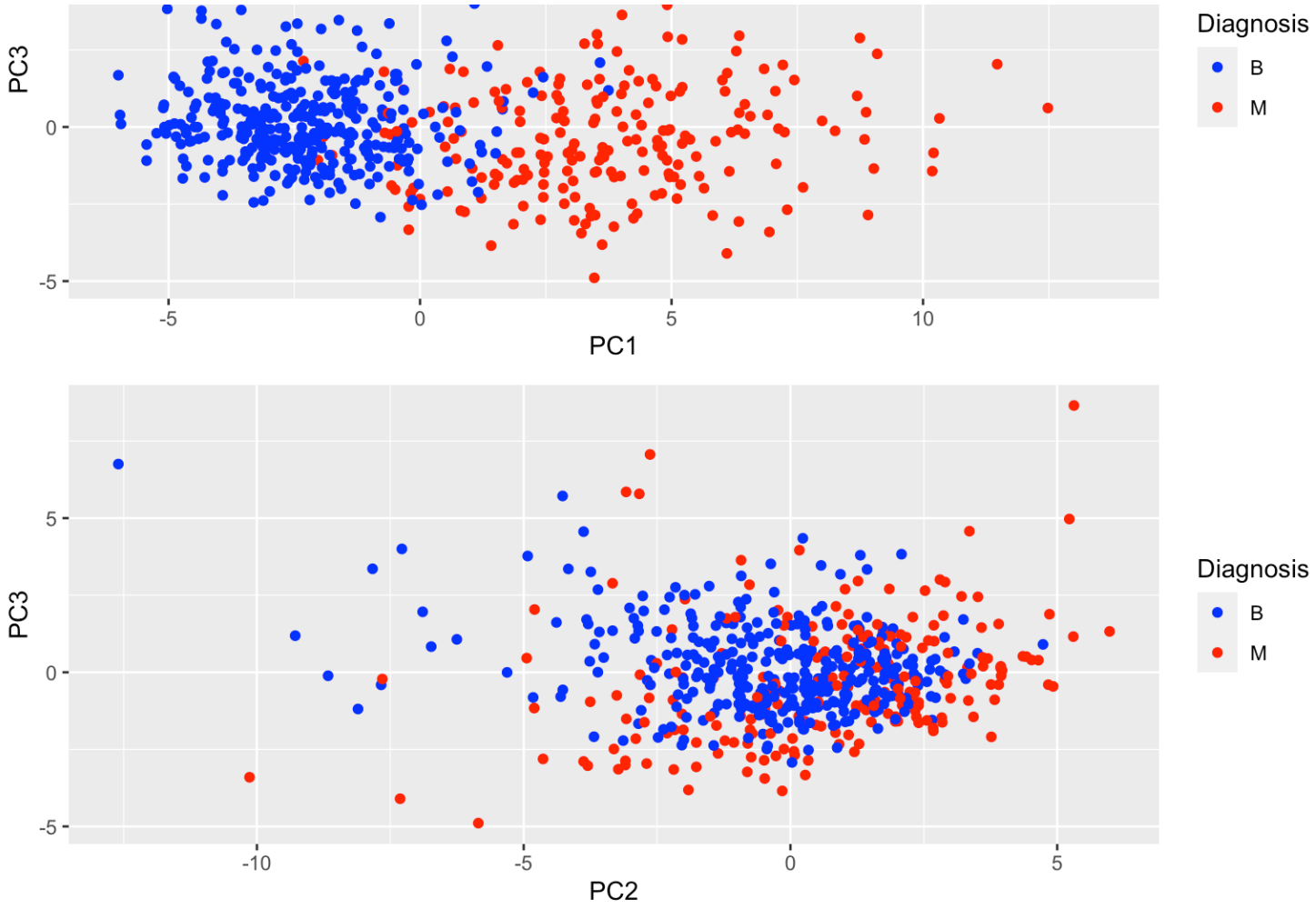
```
biplot(pca_cancer)
```



14. Plot a 3 x 1 grid of scatter plots, where each plot is a scatter plot between two of the first 3 principal components, with different colors for each diagnosis group. For example, in grid cell (1,1), you should plot a scatter plot where the x-axis is PC1 and the y-axis is PC2, where red observations correspond to malignant diagnosis and blue observations correspond to the benign diagnosis. Remember to adjust the plot display size so that you can see clearly. Add legends and labels when appropriate.

```
# Create dataframe of principle components with their scores and diagnosis for each
# observation
pc_3 <- pca_cancer$scores[,1:3]
pc_3_diag <- data.frame(pc_3, as.factor(cancer$diagnosis))
colnames(pc_3_diag) <- c("PC1", "PC2", "PC3", "Diagnosis")

# Plot scatterplots
plot1 <- ggplot(pc_3_diag, aes(x = PC1, y = PC2, color = Diagnosis)) + geom_point() +
  scale_color_manual(values=c("blue", "red"))
plot2 <- ggplot(pc_3_diag, aes(x = PC1, y = PC3, color = Diagnosis)) + geom_point() +
  scale_color_manual(values=c("blue", "red"))
plot3 <- ggplot(pc_3_diag, aes(x = PC2, y = PC3, color = Diagnosis)) + geom_point() +
  scale_color_manual(values=c("blue", "red"))
plotlist <- list(plot1, plot2, plot3)

grid.arrange(grobs = plotlist, ncol = 1,
             top = "Scatterplots Between First Three Principal Components by Diagnosi
s")
```

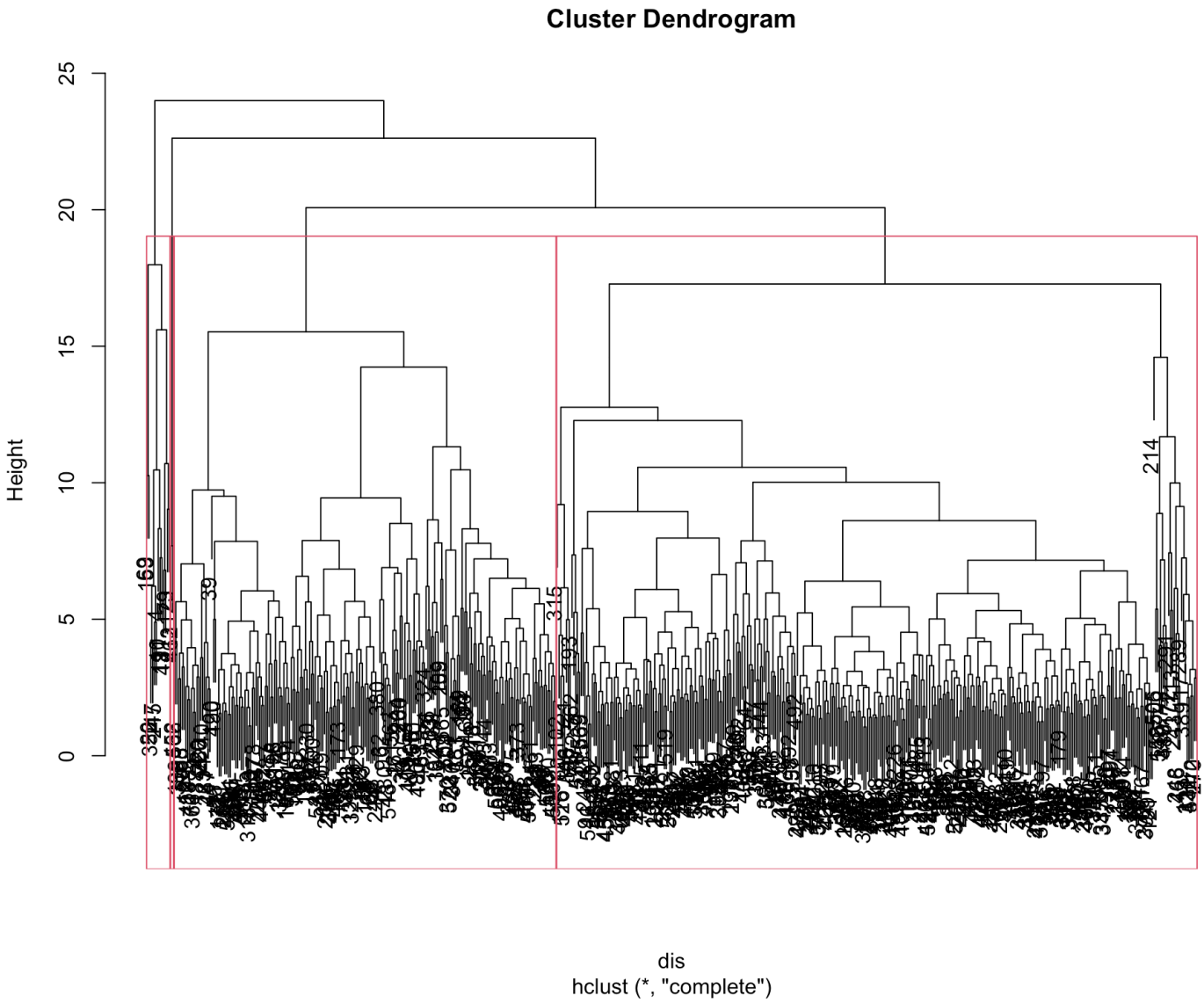




Hierarchical Clustering (15 points)

15. Calculate a dissimilarity matrix using Euclidean distance. Compute hierarchical clustering using the complete linkage method and plot the dendrogram. Use the rect.hclust() function to display dividing the dendrogram into 4 branches.

```
# Dissimilarity matrix
dis <- dist(cancer[,-1])

set.seed(20)
# Hierarchical clustering with complete linkage
hc <- hclust(dis, method = 'complete')

# Plotting
plot(hc)
rect.hclust(hc, 4)
```



16. Divide the dendrogram into 4 clusters using cutree() function. Then use the table() function and the diagnosis label to compare the diagnostic composition (benign vs. malignant) of each of the 4 clusters. If you had to choose diagnostic labels for each of the clusters, how would you label each(e.g. cluster 1 is benign or malignant, cluster 2 is …, etc.)?

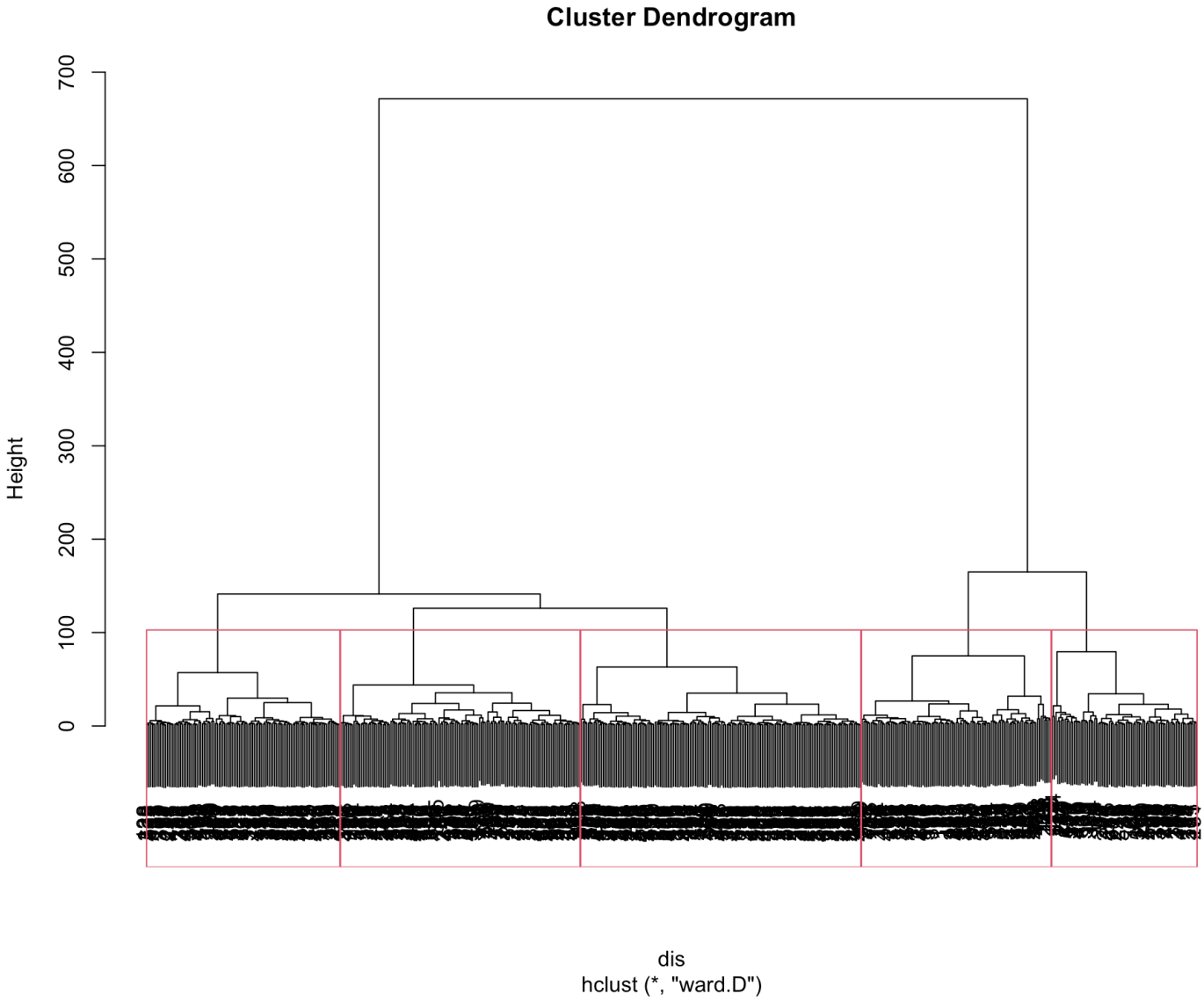
```
# Divide into 4 clusters
hc_4 <- cutree(hc, 4)

# Create dataframe with clusters and diagnosis for each observation
diagnosis <- cancer %>% pull(diagnosis)
comp <- data.frame(hc_4, diagnosis)

# Create table of amount of each diagnosis for each cluster
with(comp, table(hc_4, diagnosis))
```

```
##     diagnosis
## hc_4   B   M
##    1  18 189
##    2   2  11
##    3 337  10
##    4   0   2
```

Clusters 1, 2, and 4 I would label as malignant, and Cluster 3 as benign.

17. Now try 5 clusters with and plot dendrograms for hierarchical clustering using Ward's linkage. Then use the table() function to view the clustering result. As in the previous question, how would you label each of these 5 clusters?

```
set.seed(20)
# Hierarchical clustering with Ward's linkage
hc1 <- hclust(dis, method = 'ward.D')

# Plotting
plot(hc1)
rect.hclust(hc1, 5)
```




```
# Labeling 5 clusters
hc_5 <- cutree(hc1, 5)
comp1 <- data.frame(hc_5, diagnosis)
with(comp1, table(hc_5, diagnosis))
```

```
##     diagnosis
## hc_5   B   M
##    1   0 103
##    2  19  60
##    3  63  42
##    4 127   3
##    5 148   4
```

I would label Clusters 1 and 2 as malignant, and Clusters 3, 4, and 5 as benign.

K-Means Clustering (15 points)

18. Perform k-means clustering on this dataset using the kmeans() function with K=2. Then use the table() function and the diagnosis label to compare the diagnostic composition (benign vs. malignant) of each of the 2 clusters (hint: the cluster information from k-means is stored in the $cluster attribute of the k-means result.)

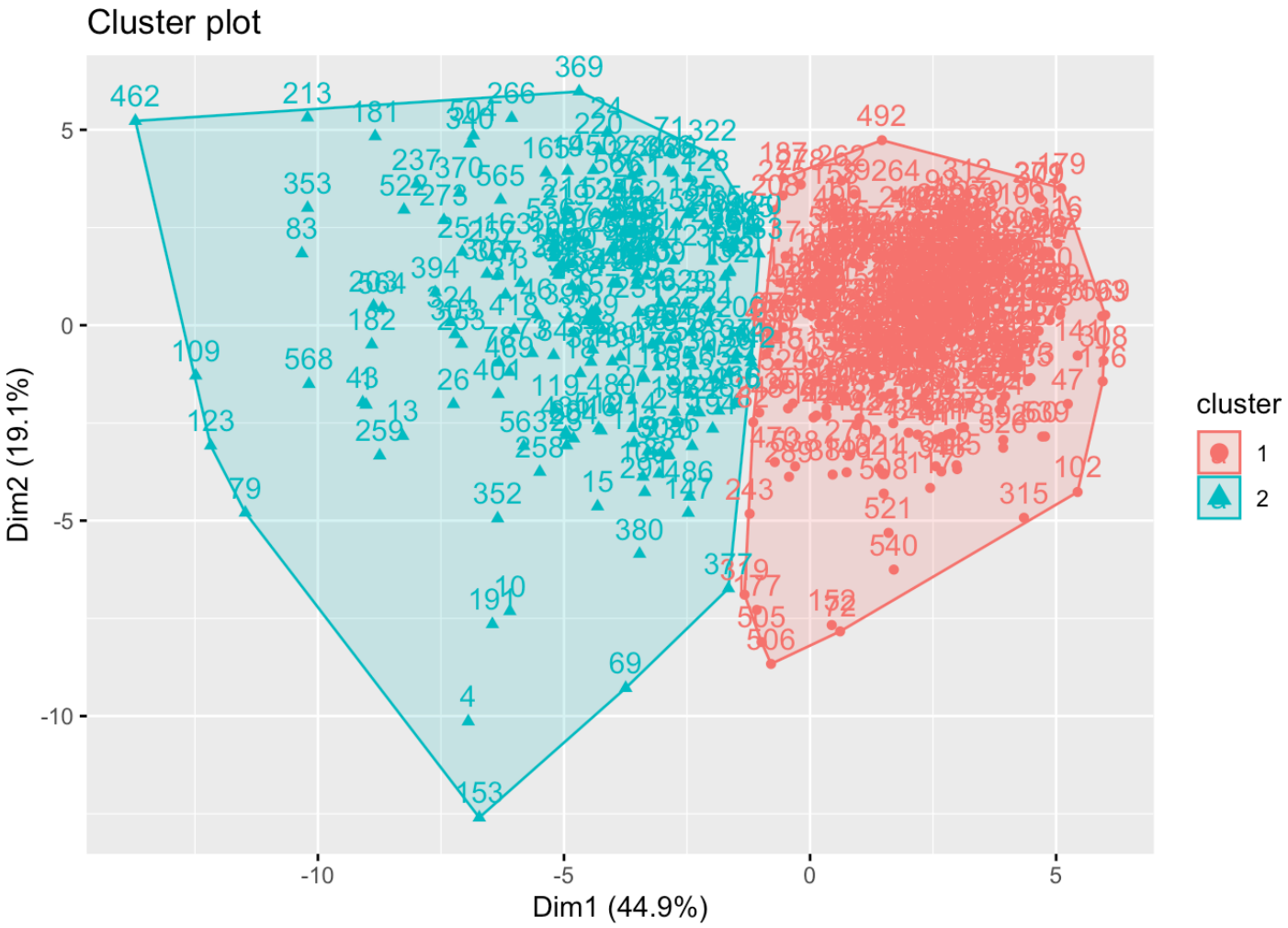
```
set.seed(20)
# Perform Kmeans clustering and create table of amount of diagnoses per cluster
km <- kmeans(cancer[,-1], 2, nstart = 20)
comp2 <- data.frame(km$cluster, diagnosis)
with(comp2, table(km$cluster, diagnosis))
```

```
##    diagnosis
##       B   M
##   1 346  33
##   2  11 179
```

Cluster 1 I would label as benign and Cluster 2 as malignant.

19. Visualize the clusters using the fviz_cluster() function from the factoextra package.

```
fviz_cluster(km, data = cancer[,-1])
```



20. What is the benefit of hierarchical clustering over k-means based on the example problem we have just explored? The benefit is that you do not have to pre-specify how many clusters you want as you do with K-means. Choosing the correct number of clusters can be difficult when using K-means. With Hierarchical, you take a bottom-up approach, and begin with each data point as its own cluster and you merge clusters together. You can then choose how many clusters you'd like after viewing the dendrogram.