HW3

Jackson Dial

2022-10-19

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

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.

```
dat0 <- read_csv("Data/Breast_Cytopathology.csv")
head(dat0)
```

```
## # A tibble: 6 x 32
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
##      <dbl> <chr>           <dbl>        <dbl>          <dbl>     <dbl>
## 1   842302 M                18.0         10.4          123.      1001
## 2   842517 M                20.6         17.8          133.      1326
## 3 84300903 M                19.7         21.2          130       1203
## 4 84348301 M                11.4         20.4           77.6      386.
## 5 84358402 M                20.3         14.3          135.      1297
## 6   843786 M                12.4         15.7           82.6      477.
## # ... with 26 more variables: smoothness_mean <dbl>, compactness_mean <dbl>,
## #   concavity_mean <dbl>, concave points_mean <dbl>, 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>, ...
```

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

a. How many observations are there in total?
b. How many independent variables are there?
c. Is there any column with missing values? If yes, how many values are missing?
d. How many observations are there with a malignant diagnosis and how many are there with a benign diagnosis?

```
nrow(dat0)
```

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

```
ncol(dat0)
```

```
## [1] 32
```

```
sum(is.na(dat0))
```

```
## [1] 6
```

```
summary(dat0) %>% pander()
```

Table continues below

id	diagnosis	radius_mean	texture_mean
Min. : 8670	Length:569	Min. : 6.981	Min. : 9.71
1st Qu.: 869218	Class :character	1st Qu.:11.700	1st Qu.:16.17
Median : 906024	Mode :character	Median :13.370	Median :18.84

id	diagnosis	radius_mean	texture_mean
Mean : 30371831	NA	Mean :14.127	Mean :19.29
3rd Qu.: 8813129	NA	3rd Qu.:15.780	3rd Qu.:21.80
Max. :911320502	NA	Max. :28.110	Max. :39.28
NA	NA	NA	NA

Table continues below

perimeter_mean	area_mean	smoothness_mean	compactness_mean
Min. : 43.79	Min. : 143.5	Min. :0.05263	Min. :0.01938
1st Qu.: 75.17	1st Qu.: 420.3	1st Qu.:0.08637	1st Qu.:0.06492
Median : 86.24	Median : 551.1	Median :0.09587	Median :0.09263
Mean : 91.97	Mean : 654.9	Mean :0.09636	Mean :0.10434
3rd Qu.:104.10	3rd Qu.: 782.7	3rd Qu.:0.10530	3rd Qu.:0.13040
Max. :188.50	Max. :2501.0	Max. :0.16340	Max. :0.34540
NA	NA	NA	NA

Table continues below

concavity_mean	concave points_mean	symmetry_mean
Min. :0.00000	Min. :0.00000	Min. :0.1060
1st Qu.:0.02956	1st Qu.:0.02031	1st Qu.:0.1619
Median :0.06154	Median :0.03350	Median :0.1792
Mean :0.08880	Mean :0.04892	Mean :0.1812
3rd Qu.:0.13070	3rd Qu.:0.07400	3rd Qu.:0.1957
Max. :0.42680	Max. :0.20120	Max. :0.3040
NA	NA	NA

Table continues below

fractal_dimension_mean	radius_se	texture_se	perimeter_se
Min. :0.04996	Min. :0.1115	Min. :0.3602	Min. : 0.757
1st Qu.:0.05769	1st Qu.:0.2324	1st Qu.:0.8339	1st Qu.: 1.606
Median :0.06149	Median :0.3242	Median :1.1080	Median : 2.287
Mean :0.06274	Mean :0.4052	Mean :1.2169	Mean : 2.866

fractal_dimension_mean	radius_se	texture_se	perimeter_se
3rd Qu.:0.06610	3rd Qu.:0.4789	3rd Qu.:1.4740	3rd Qu.: 3.357
Max. :0.09744	Max. :2.8730	Max. :4.8850	Max. :21.980
NA’s :6	NA	NA	NA

Table continues below

area_se	smoothness_se	compactness_se	concavity_se
Min. : 6.802	Min. :0.001713	Min. :0.002252	Min. :0.00000
1st Qu.: 17.850	1st Qu.:0.005169	1st Qu.:0.013080	1st Qu.:0.01509
Median : 24.530	Median :0.006380	Median :0.020450	Median :0.02589
Mean : 40.337	Mean :0.007041	Mean :0.025478	Mean :0.03189
3rd Qu.: 45.190	3rd Qu.:0.008146	3rd Qu.:0.032450	3rd Qu.:0.04205
Max. :542.200	Max. :0.031130	Max. :0.135400	Max. :0.39600
NA	NA	NA	NA

Table continues below

concave points_se	symmetry_se	fractal_dimension_se	radius_worst
Min. :0.000000	Min. :0.007882	Min. :0.0008948	Min. : 7.93
1st Qu.:0.007638	1st Qu.:0.015160	1st Qu.:0.0022480	1st Qu.:13.01
Median :0.010930	Median :0.018730	Median :0.0031870	Median :14.97
Mean :0.011796	Mean :0.020542	Mean :0.0037949	Mean :16.27
3rd Qu.:0.014710	3rd Qu.:0.023480	3rd Qu.:0.0045580	3rd Qu.:18.79
Max. :0.052790	Max. :0.078950	Max. :0.0298400	Max. :36.04
NA	NA	NA	NA

Table continues below

texture_worst	perimeter_worst	area_worst	smoothness_worst
Min. :12.02	Min. : 50.41	Min. : 185.2	Min. :0.07117
1st Qu.:21.08	1st Qu.: 84.11	1st Qu.: 515.3	1st Qu.:0.11660
Median :25.41	Median : 97.66	Median : 686.5	Median :0.13130
Mean :25.68	Mean :107.26	Mean : 880.6	Mean :0.13237
3rd Qu.:29.72	3rd Qu.:125.40	3rd Qu.:1084.0	3rd Qu.:0.14600

texture_worst	perimeter_worst	area_worst	smoothness_worst
Max. :49.54	Max. :251.20	Max. :4254.0	Max. :0.22260
NA	NA	NA	NA

Table continues below

compactness_worst	concavity_worst	concave points_worst	symmetry_worst
Min. :0.02729	Min. :0.0000	Min. :0.00000	Min. :0.1565
1st Qu.:0.14720	1st Qu.:0.1145	1st Qu.:0.06493	1st Qu.:0.2504
Median :0.21190	Median :0.2267	Median :0.09993	Median :0.2822
Mean :0.25427	Mean :0.2722	Mean :0.11461	Mean :0.2901
3rd Qu.:0.33910	3rd Qu.:0.3829	3rd Qu.:0.16140	3rd Qu.:0.3179
Max. :1.05800	Max. :1.2520	Max. :0.29100	Max. :0.6638
NA	NA	NA	NA

fractal_dimension_worst
Min. :0.05504
1st Qu.:0.07146
Median :0.08004
Mean :0.08395
3rd Qu.:0.09208
Max. :0.20750
NA

```
class(dat0$diagnosis)
```

```
## [1] "character"
```

```
dat0$diagnosis <- as.factor(dat0$diagnosis)
levels(dat0$diagnosis)
```

```
## [1] "B" "M"
```

```
table(dat0$diagnosis) %>% pander()
```

B	M
357	212

There are 569 observations in this dataset, and 10 independent variables. The 'fractal_dimension_mean' variable has 6 NA values, and is the only column with any missing values. There are 357 observations with benign diagnosis, and 212 with malignant diagnosis.

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)

```
dat1 <- dat0 %>% remove_rownames() %>% column_to_rownames(var = "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.

```
#check if different imputaiton should be used
dat1 %>% filter(is.na(dat1$fractal_dimension_mean) == TRUE) %>% select(diagnosis, fractal_dimens
ion_mean)
```

```
##         diagnosis fractal_dimension_mean
## 843786          M                     NA
## 852973          M                     NA
## 857373          B                     NA
## 8610175         B                     NA
## 862261          B                     NA
## 86409           B                     NA
```

```
dat1 %>% filter(diagnosis == "M") %>% summarise(avg_fdm = mean(fractal_dimension_mean, na.rm = T
RUE))
```

```
##     avg_fdm
## 1 0.0626031
```

```
dat1 %>% filter(diagnosis == "B") %>% summarise(avg_fdm = mean(fractal_dimension_mean, na.rm = T
RUE))
```

```
##      avg_fdm
## 1 0.06282833
```

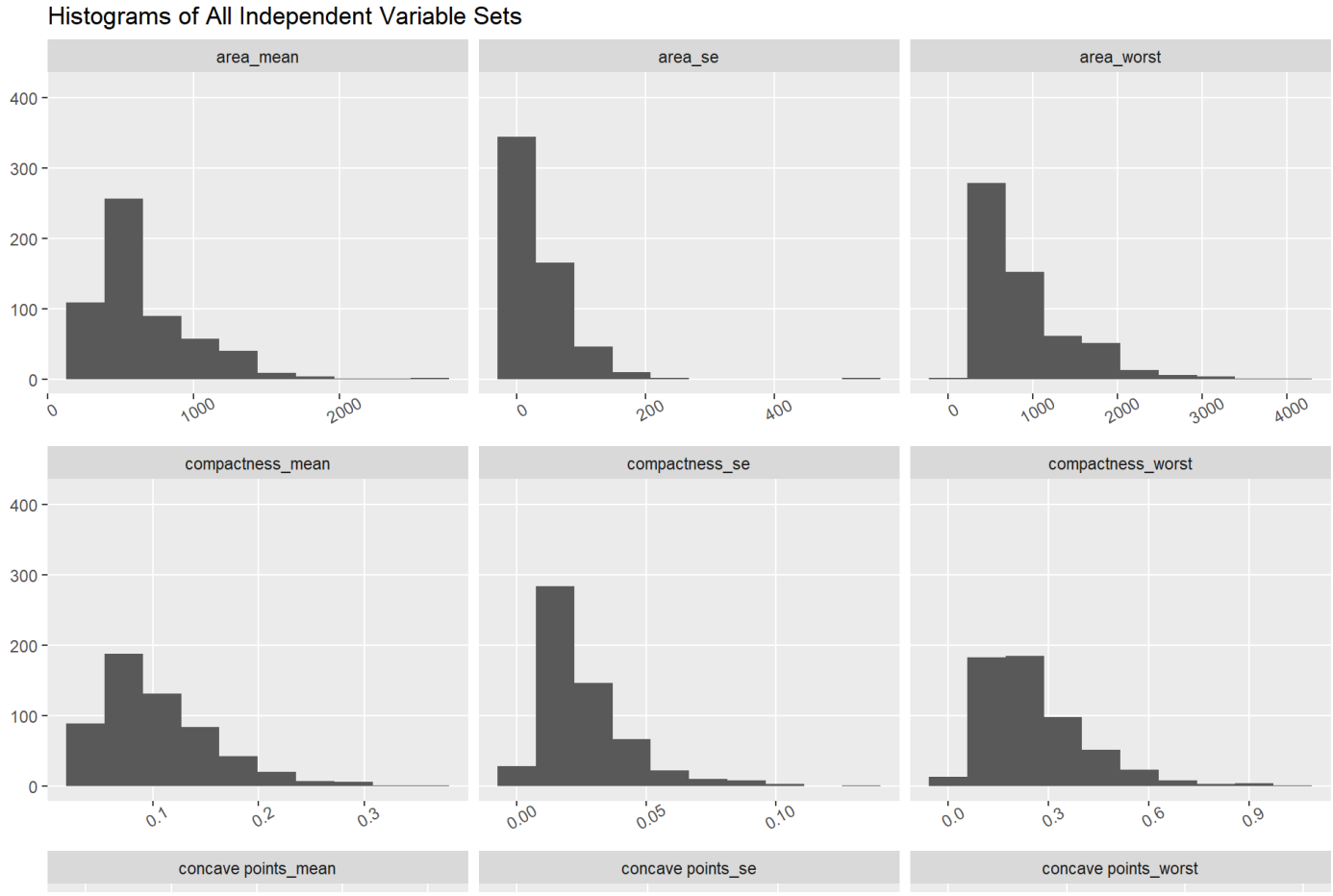
```
#since the two values are very similar it will not be necessary to use factor-level-wise mean im
putation.
```

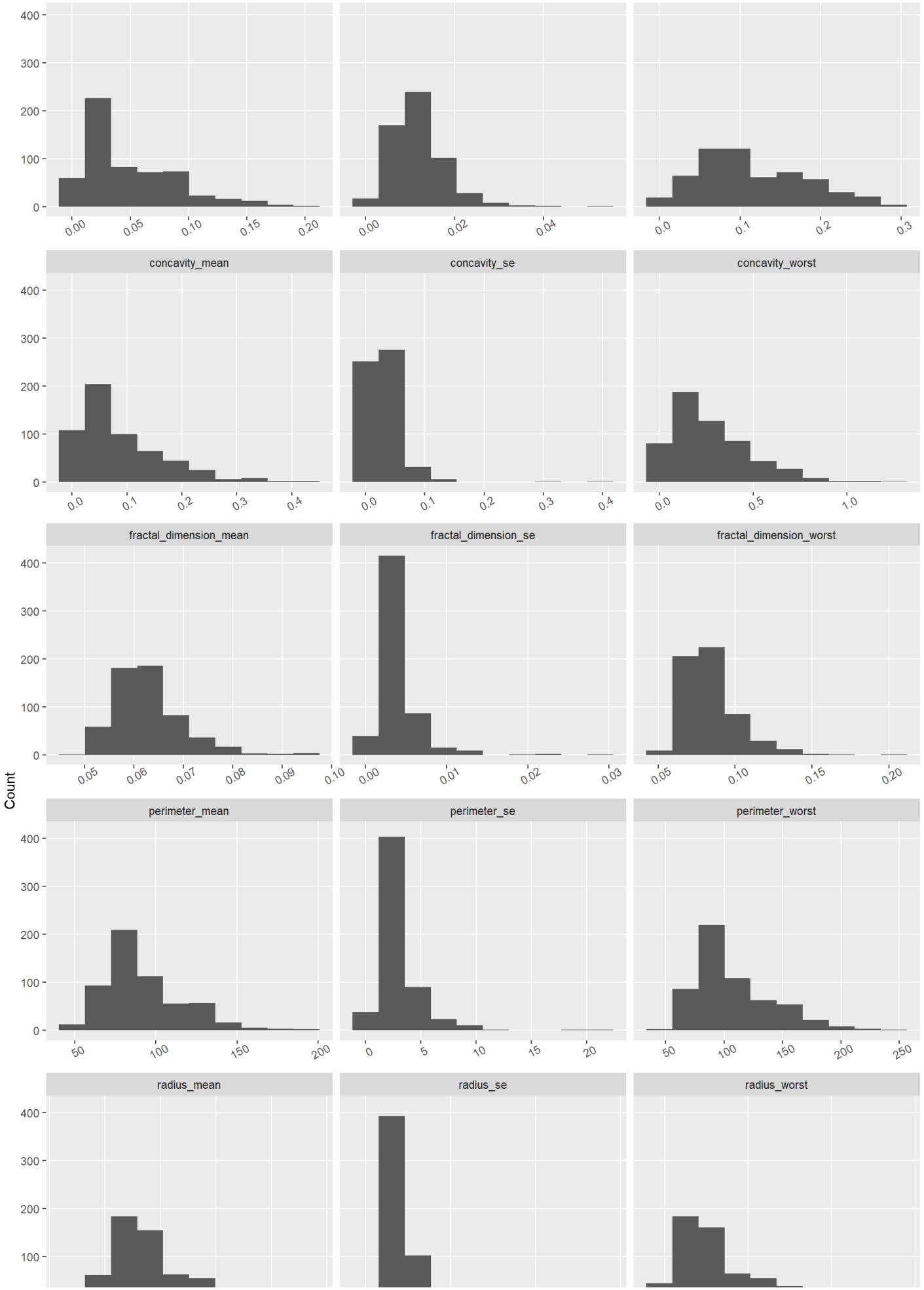
```
dat1$fractal_dimension_mean[is.na(dat1$fractal_dimension_mean)] <- mean(dat1$fractal_dimension_m
ean, na.rm = TRUE)
sum(is.na(dat1))
```

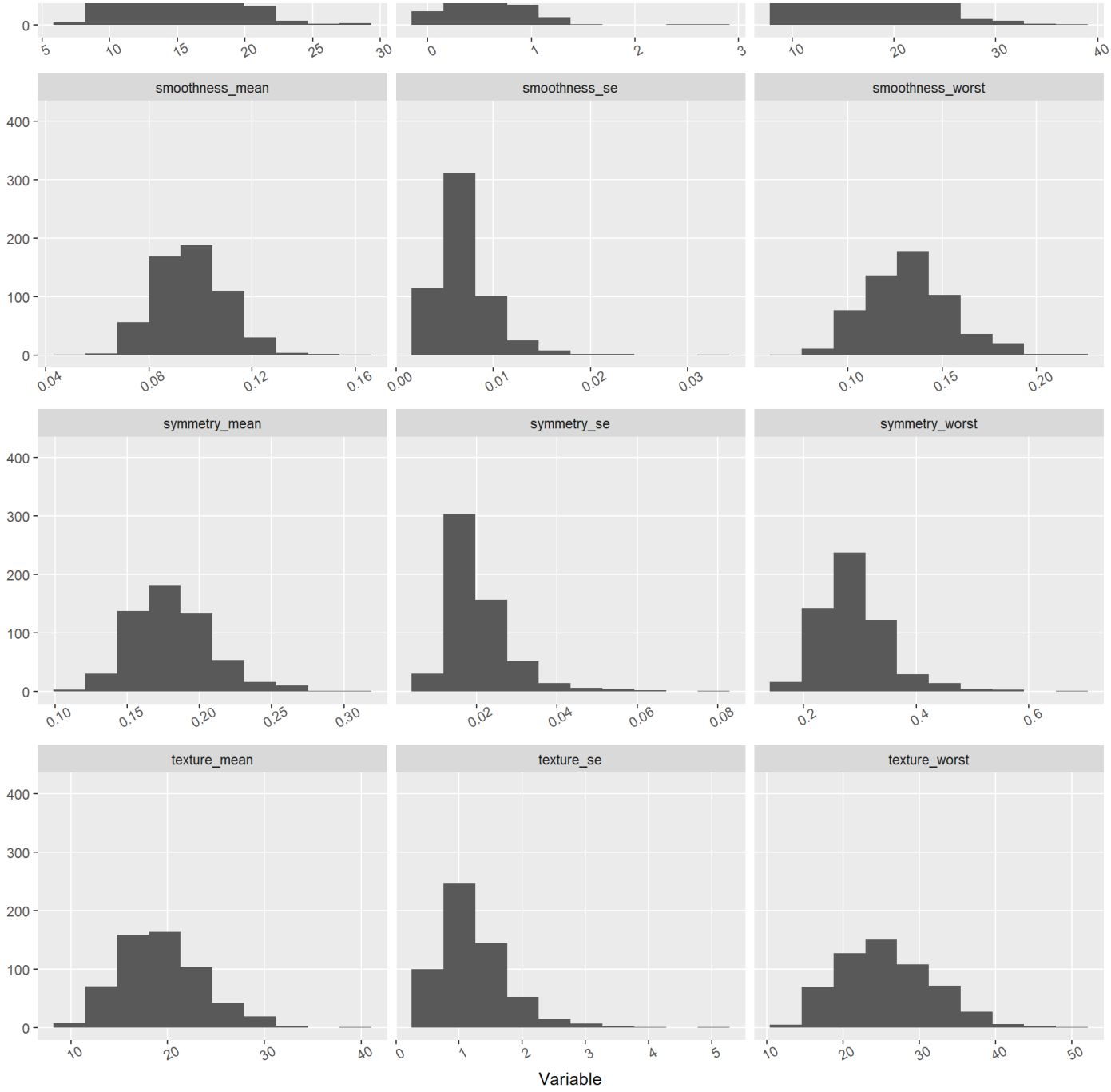
```
## [1] 0
```

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.

```
ggplot(gather(dat1 %>% subset(select = -c(diagnosis))), aes(value)) +
    geom_histogram(bins = 10) +
    facet_wrap(~key, scales = 'free_x', nrow = 10)+
  theme(axis.text.x = element_text(angle = 30),
        panel.grid.minor = element_blank())+
  labs(x = "Variable",
       y = "Count",
       title = "Histograms of All Independent Variable Sets")
```




concavity_mean
concavity_se
concavity_worst
fractal_dimension_mean
fractal_dimension_se
fractal_dimension_worst
perimeter_mean
perimeter_se
perimeter_worst
radius_mean
radius_se
radius_worst
Count



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.

```
# area_ <- dat1[grepl("area",colnames(dat1))] %>% colnames()
# compactness_ <- dat1[grepl("compactness",colnames(dat1))] %>% colnames()
# concavity_ <- dat1[grepl("concavity",colnames(dat1))] %>% colnames()
# my_vars <- c(area_, compactness_, concavity_)

# library(foreach)
# i = 1
# for (i in 1:length(my_vars)) {
#   current_var <- col[i]
#   dat1$current_val <- exp(dat1$current_val +1)
#   i = i + 1
# }

dat2 <- dat1 %>%
  mutate(area_mean = log(area_mean + 1),
         area_se = log(area_se + 1),
         area_worst = log(area_worst + 1),
         compactness_mean = log(compactness_mean + 1),
         compactness_se = log(compactness_se + 1),
         compactness_worst = log(compactness_worst + 1),
         concavity_mean = log(concavity_mean + 1),
         concavity_se = log(concavity_se +1),
         concavity_worst = log(concavity_worst + 1))
```

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.

```
diag_only <- dat2 %>% select(diagnosis)
dat3 <- dat2 %>% subset(select = -c(diagnosis)) %>% mutate_all(~(scale(.) %>% as.vector)) %>% cb
ind(diag_only)
summary(dat3) %>% pander()
```

Table continues below

radius_mean	texture_mean	perimeter_mean	area_mean
Min. :-2.0279	Min. :-2.2273	Min. :-1.9828	Min. :-2.8860
1st Qu.:-0.6888	1st Qu.:-0.7253	1st Qu.:-0.6913	1st Qu.:-0.6672
Median :-0.2149	Median :-0.1045	Median :-0.2358	Median :-0.1065
Mean : 0.0000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.4690	3rd Qu.: 0.5837	3rd Qu.: 0.4992	3rd Qu.: 0.6198
Max. : 3.9678	Max. : 4.6478	Max. : 3.9726	Max. : 3.0268

Table continues below

smoothness_mean	compactness_mean	concavity_mean	concave points_mean

smoothness_mean	compactness_mean	concavity_mean	concave points_mean
Min. :-3.10935	Min. :-1.6925	Min. :-1.1774	Min. :-1.2607
1st Qu.:-0.71034	1st Qu.:-0.7556	1st Qu.:-0.7619	1st Qu.:-0.7373
Median :-0.03486	Median :-0.2049	Median :-0.3256	Median :-0.3974
Mean : 0.00000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.63564	3rd Qu.: 0.5237	3rd Qu.: 0.5746	3rd Qu.: 0.6464
Max. : 4.76672	Max. : 4.2564	Max. : 3.8920	Max. : 3.9245

Table continues below

symmetry_mean	fractal_dimension_mean	radius_se	texture_se
Min. :-2.74171	Min. :-1.8259	Min. :-1.0590	Min. :-1.5529
1st Qu.:-0.70262	1st Qu.:-0.7205	1st Qu.:-0.6230	1st Qu.:-0.6942
Median :-0.07156	Median :-0.1620	Median :-0.2920	Median :-0.1973
Mean : 0.00000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.53031	3rd Qu.: 0.4736	3rd Qu.: 0.2659	3rd Qu.: 0.4661
Max. : 4.48081	Max. : 4.9554	Max. : 8.8991	Max. : 6.6494

Table continues below

perimeter_se	area_se	smoothness_se	compactness_se
Min. :-1.0431	Min. :-1.9362	Min. :-1.7745	Min. :-1.3241
1st Qu.:-0.6232	1st Qu.:-0.6865	1st Qu.:-0.6235	1st Qu.:-0.6989
Median :-0.2864	Median :-0.2568	Median :-0.2201	Median :-0.2773
Mean : 0.0000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.2428	3rd Qu.: 0.5832	3rd Qu.: 0.3680	3rd Qu.: 0.4028
Max. : 9.4537	Max. : 4.0749	Max. : 8.0229	Max. : 5.9323

Table continues below

concavity_se	concave points_se	symmetry_se	fractal_dimension_se
Min. :-1.1284	Min. :-1.9118	Min. :-1.5315	Min. :-1.0960
1st Qu.:-0.5833	1st Qu.:-0.6739	1st Qu.:-0.6511	1st Qu.:-0.5846
Median :-0.1981	Median :-0.1404	Median :-0.2192	Median :-0.2297
Mean : 0.0000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000

concavity_se	concave points_se	symmetry_se	fractal_dimension_se
3rd Qu.: 0.3708	3rd Qu.: 0.4722	3rd Qu.: 0.3554	3rd Qu.: 0.2884
Max. :11.0139	Max. : 6.6438	Max. : 7.0657	Max. : 9.8429

Table continues below

radius_worst	texture_worst	perimeter_worst	area_worst
Min. :-1.7254	Min. :-2.22204	Min. :-1.6919	Min. :-2.5092
1st Qu.:-0.6743	1st Qu.:-0.74797	1st Qu.:-0.6890	1st Qu.:-0.6689
Median :-0.2688	Median :-0.04348	Median :-0.2857	Median :-0.1521
Mean : 0.0000	Mean : 0.00000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.5216	3rd Qu.: 0.65776	3rd Qu.: 0.5398	3rd Qu.: 0.6712
Max. : 4.0906	Max. : 3.88249	Max. : 4.2836	Max. : 3.1371

Table continues below

smoothness_worst	compactness_worst	concavity_worst	concave points_worst
Min. :-2.6803	Min. :-1.6394	Min. :-1.4782	Min. :-1.7435
1st Qu.:-0.6906	1st Qu.:-0.6990	1st Qu.:-0.7766	1st Qu.:-0.7557
Median :-0.0468	Median :-0.2316	Median :-0.1557	Median :-0.2233
Mean : 0.0000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.5970	3rd Qu.: 0.6186	3rd Qu.: 0.6201	3rd Qu.: 0.7119
Max. : 3.9519	Max. : 4.2793	Max. : 3.7763	Max. : 2.6835

symmetry_worst	fractal_dimension_worst	diagnosis
Min. :-2.1591	Min. :-1.6004	B:357
1st Qu.:-0.6413	1st Qu.:-0.6913	M:212
Median :-0.1273	Median :-0.2163	NA
Mean : 0.0000	Mean : 0.0000	NA
3rd Qu.: 0.4497	3rd Qu.: 0.4504	NA
Max. : 6.0407	Max. : 6.8408	NA

PCA (25 points)

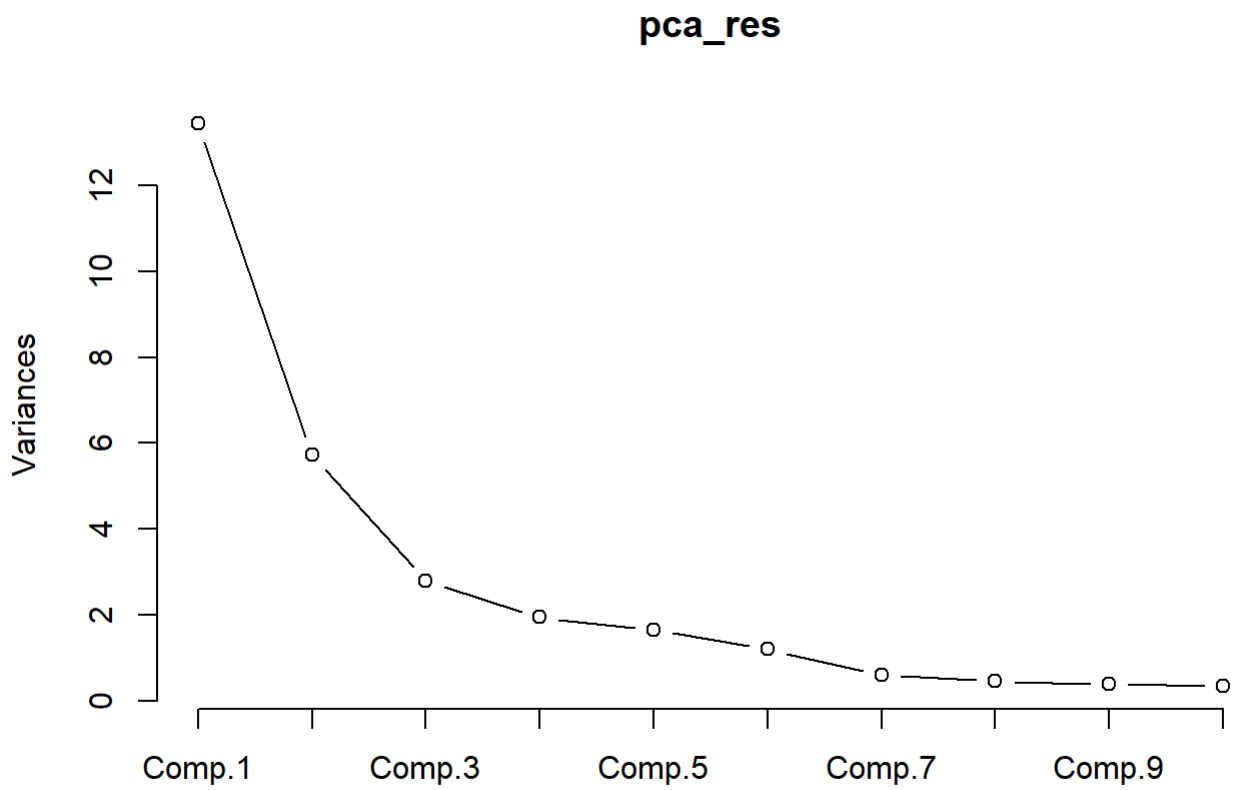
8. Calculate the principal components using the function princomp() and print the summary of the results.

```
pca_res <- princomp(dat3 %>% subset(select = -c(diagnosis)), scores = TRUE)
pca_res
```

```
## Call:
## princomp(x = dat3 %>% subset(select = -c(diagnosis)), scores = TRUE)
##
## Standard deviations:
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 3.66498724 2.39226784 1.66812505 1.39504425 1.28283303 1.09535346 0.77909553
##     Comp.8     Comp.9    Comp.10    Comp.11    Comp.12    Comp.13    Comp.14
## 0.68071443 0.62487552 0.58480234 0.52711827 0.49698962 0.48066857 0.40160221
##    Comp.15    Comp.16    Comp.17    Comp.18    Comp.19    Comp.20    Comp.21
## 0.30044290 0.28996368 0.26854025 0.23503444 0.19320842 0.17947906 0.17414979
##    Comp.22    Comp.23    Comp.24    Comp.25    Comp.26    Comp.27    Comp.28
## 0.16393421 0.14758231 0.12106563 0.11149618 0.10554321 0.08400599 0.04101629
##    Comp.29    Comp.30
## 0.02477533 0.01180667
##
##  30  variables and  569 observations.
```

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

```
screeplot(pca_res, type = "lines")
```

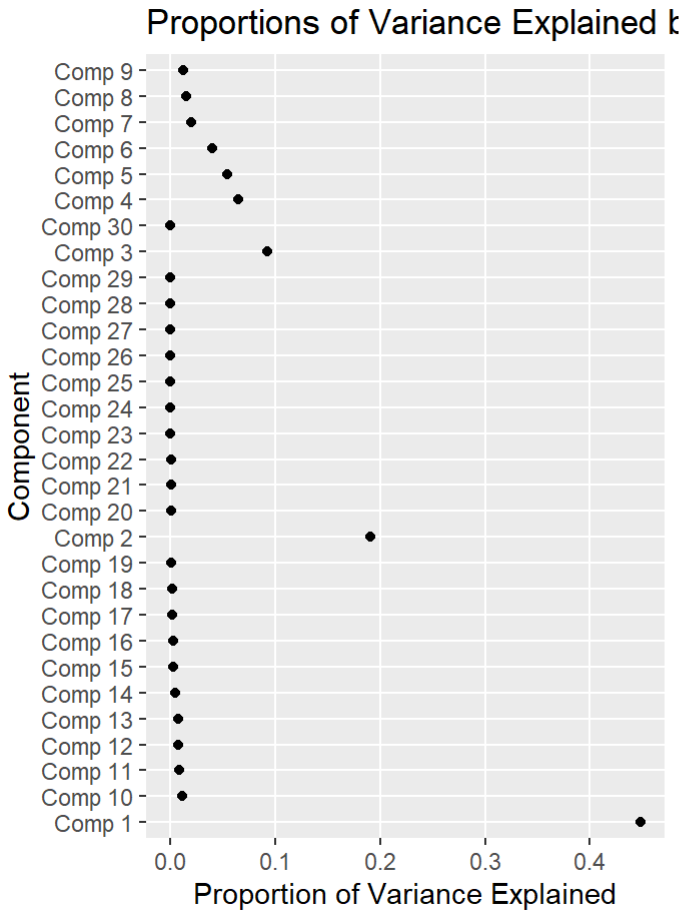


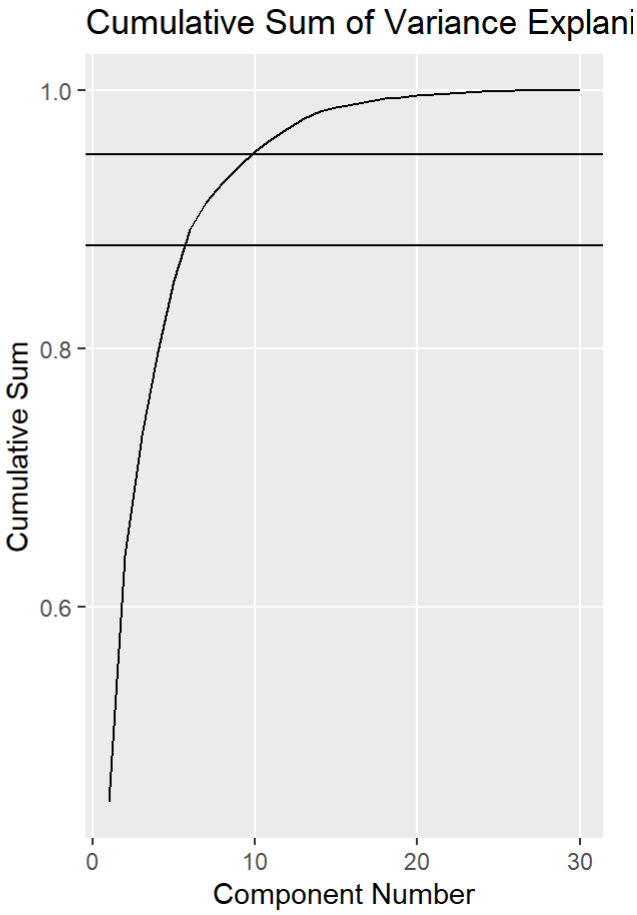
10. Plot the following two plots and use patchwork/gridExtra to position the two plots side by side:

    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.

```
pca_variances <- (pca_res$sdev)^2
pca_probs <- pca_variances / sum(pca_variances)
proportion_df <- cbind.data.frame(pca_probs, "Component" = paste("Comp", 1:30))

p1 <- ggplot(proportion_df, aes(x = as.factor(Component), y = pca_probs))+
  geom_point()+
  coord_flip()+
  labs(y = "Proportion of Variance Explained",
       x =  "Component",
       title = "Proportions of Variance Explained by Component Number")+
  theme(panel.grid.minor = element_blank())


proportion_df$cum_sum <- cumsum(proportion_df$pca_probs)

p2 <- ggplot(proportion_df, aes(y = cum_sum, x = 1:30))+
  geom_line()+
  labs(x = "Component Number",
       y = "Cumulative Sum",
       title = "Cumulative Sum of Variance Explanined by Component")+
  theme(panel.grid.minor = element_blank())+
  geom_hline(yintercept = c(.88, .95))

p1 + p2
```





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?

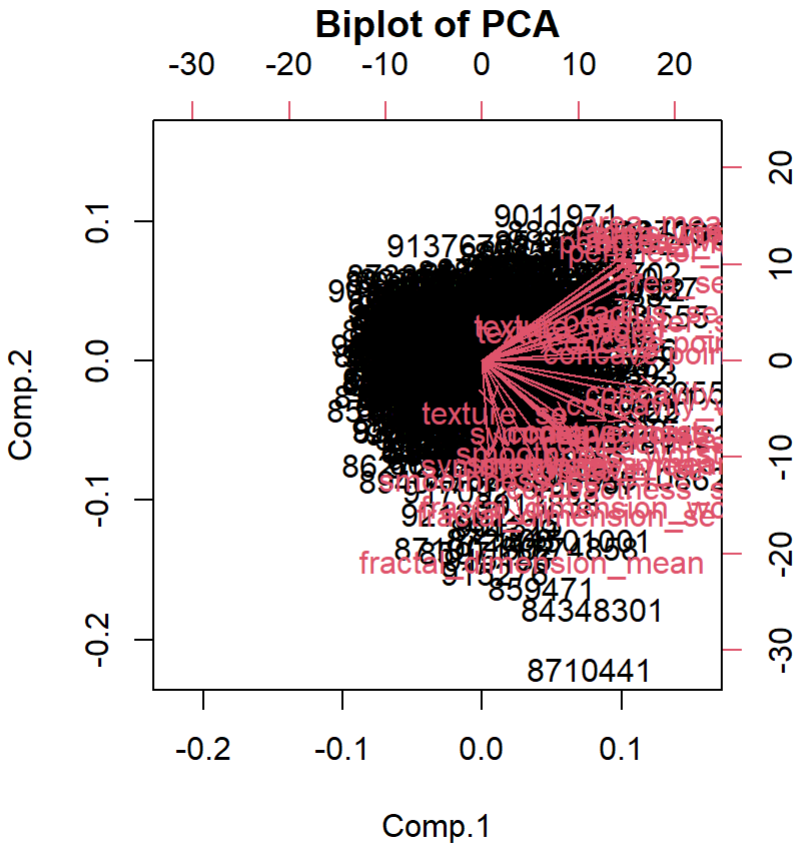
The first PC captures about 45% of the variance, the second captures about 19% additionally, and the third captures about 9.1%. To describe at least 88% of the variance, we need the first 6 components, and to describe 95% of the variance, we need the first 10.

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)

For component 1, concave points_mean and concavity_mean contribute most. For component 2, fractal_dimension_mean and fractal_dimension_se contribute most For component 3, texture_se and smoothness_se contribute most.

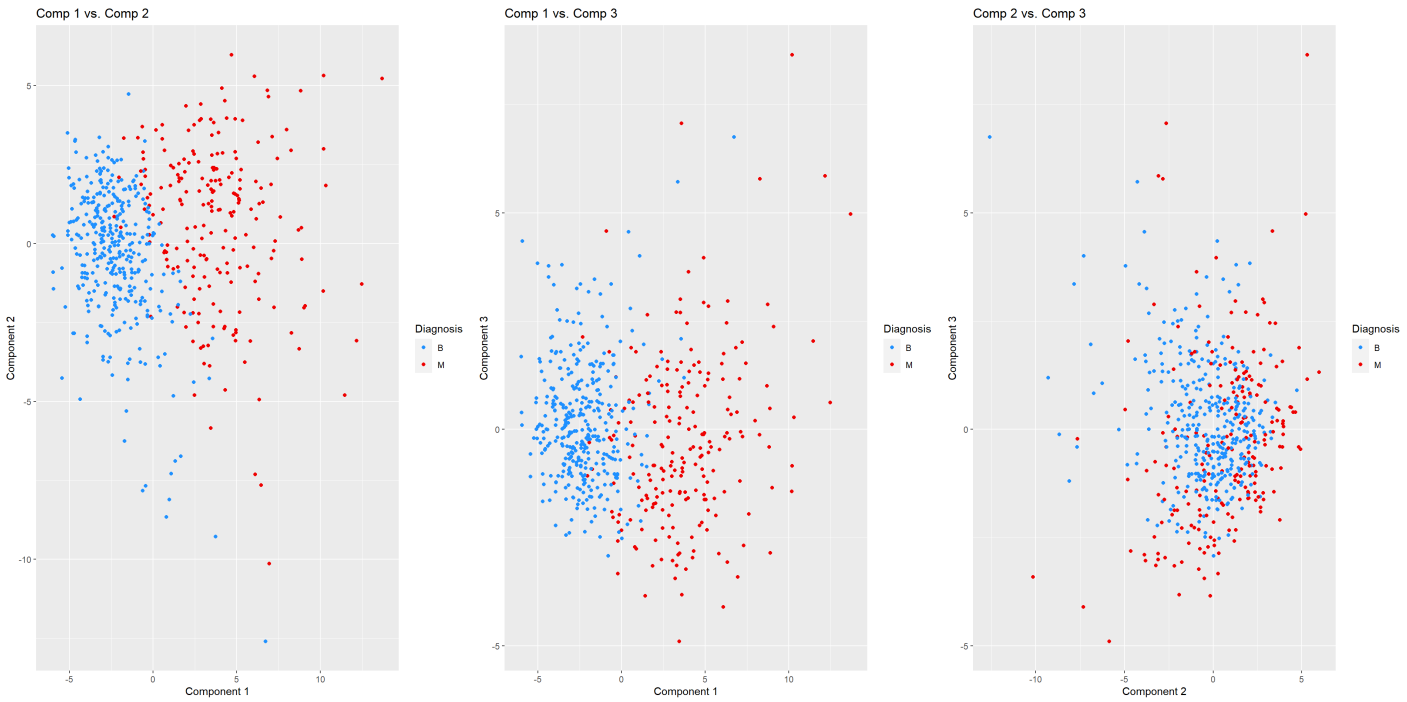
13. Plot a biplot using the biplot() function.

```
biplot(pca_res, main = "Biplot of PCA")
```



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.

```
pca_scores <- pca_res$scores %>% as.data.frame() %>% cbind(diag_only)
p1 <- ggplot(pca_scores, aes(x = Comp.1, y = Comp.2))+
  geom_point(aes(color = as.factor(diagnosis)))+
  labs(x = "Component 1",
       y = "Component 2",
       title = "Comp 1 vs. Comp 2",
       color = "Diagnosis")+
  scale_color_manual(values = c("dodgerblue", "red2"))

p2 <- ggplot(pca_scores, aes(x = Comp.1, y = Comp.3))+
  geom_point(aes(color = as.factor(diagnosis)))+
  labs(x = "Component 1",
       y = "Component 3",
       title = "Comp 1 vs. Comp 3",
       color = "Diagnosis")+
    scale_color_manual(values = c("dodgerblue", "red2"))


p3 <- ggplot(pca_scores, aes(x = Comp.2, y = Comp.3))+
  geom_point(aes(color = as.factor(diagnosis)))+
  labs(x = "Component 2",
       y = "Component 3",
       title = "Comp 2 vs. Comp 3",
       color = "Diagnosis")+
    scale_color_manual(values = c("dodgerblue", "red2"))


p1+p2+p3
```



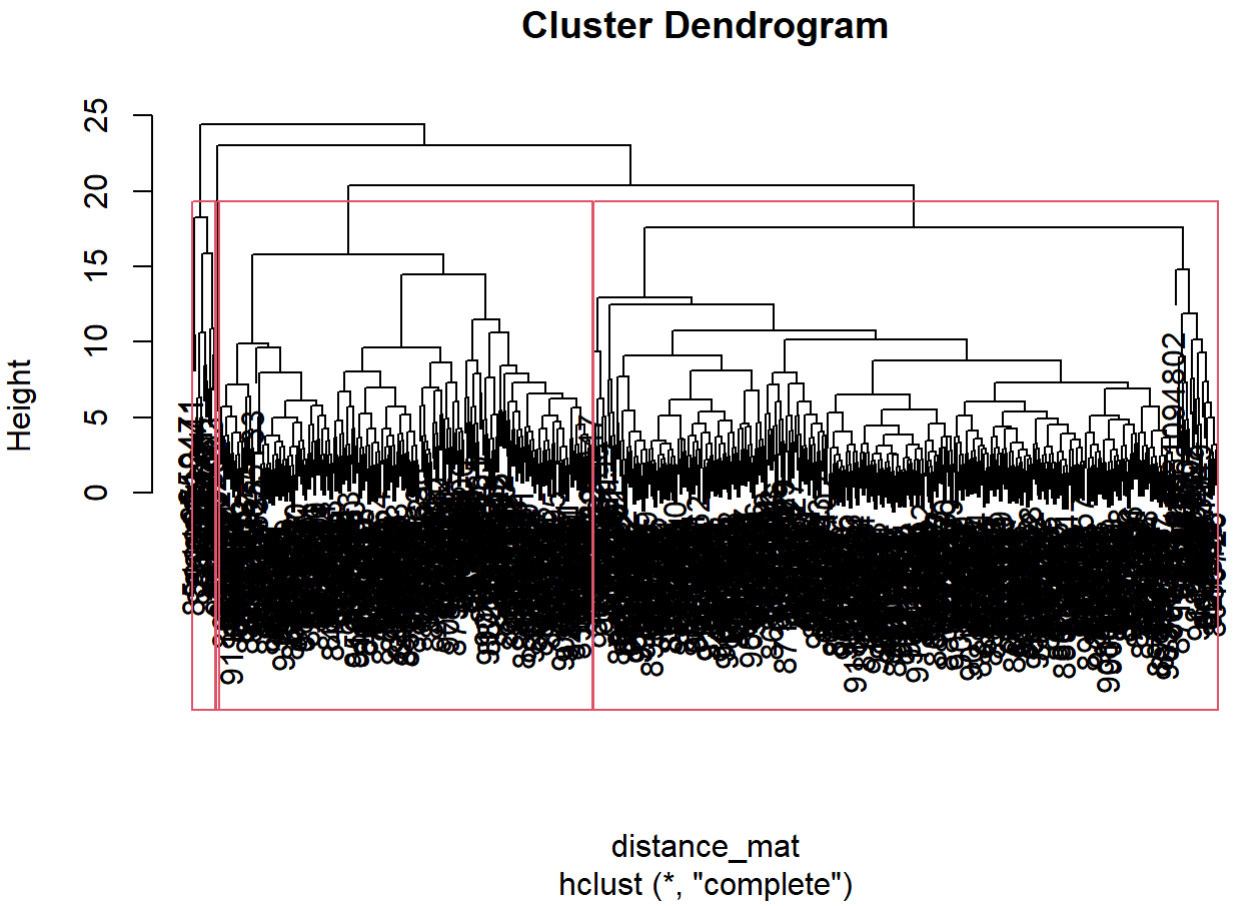
Hierarchical Clustering (15 points)

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.

```
distance_mat <- dist(dat3, method = 'euclidean')
```

```
## Warning in dist(dat3, method = "euclidean"): NAs introduced by coercion
```

```
Hierar_cl <- hclust(distance_mat, method = "complete")
plot(Hierar_cl)

hclust_obj <- rect.hclust(Hierar_cl, k = 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.)?

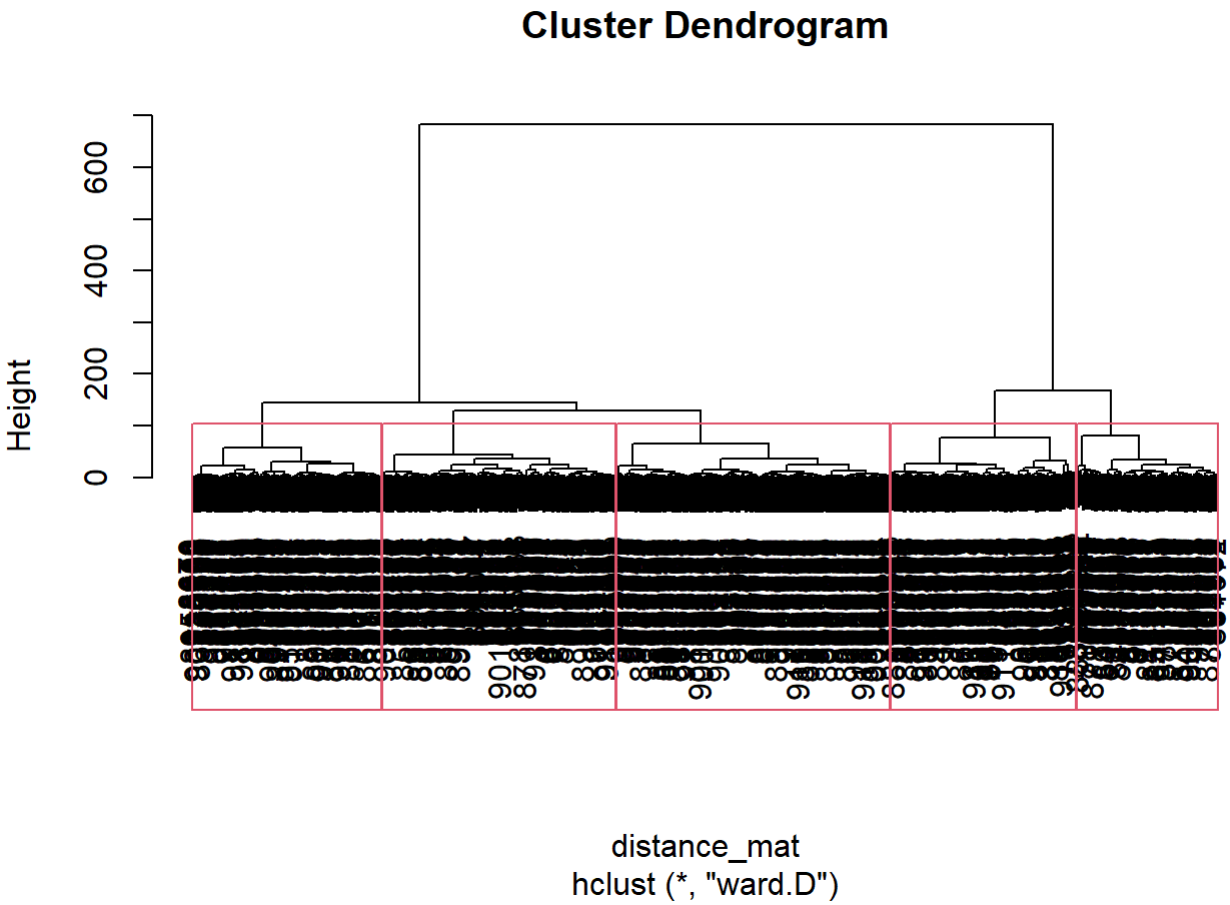
```
cut_tree <- cutree(Hierar_cl, k = 4)
table(cut_tree, diag_only$diagnosis)
```

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

Cluster 1 should be Malignant, cluster 2 should be malignant, cluster 3 should be benign, and cluster 4 should be malignant.

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?

```
Hierar_cl2 <- hclust(distance_mat, method = "ward.D")
plot(Hierar_cl2)

hclust_obj2 <- rect.hclust(Hierar_cl2, k = 5)
```




```
cut_tree2 <- cutree(Hierar_cl2, k = 5)
table(cut_tree2, diag_only$diagnosis)
```

```
##
## cut_tree2   B   M
##         1   0 103
##         2  19  60
##         3  63  42
##         4 127   3
##         5 148   4
```

Cluster 1 should be malignant, cluster 2 should be malignant, cluster 3 should be benign, cluster 4 should be benign, and cluster 5 should be 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.)

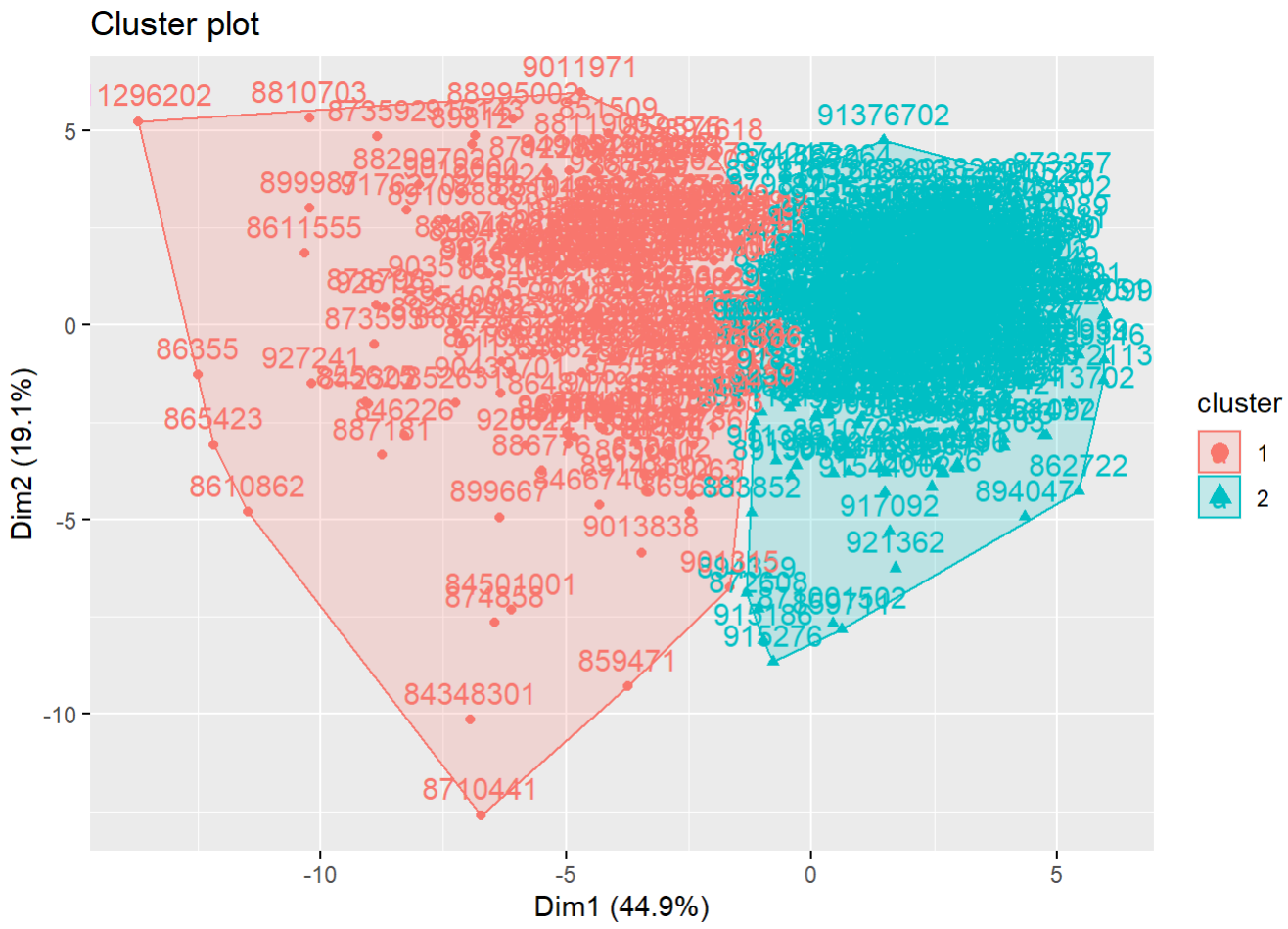
```
k_means <- kmeans(dat3 %>% subset(select = -c(diagnosis)), centers = 2)
table(k_means$cluster, diag_only$diagnosis)
```

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

The first cluster should be labelled as malignant, and the second cluster should be labelled as benign.

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

```
fviz_cluster(k_means, data = dat3 %>% subset(select = -c(diagnosis)))
```



20. What is the benefit of hierarchical clustering over k-means based on the example problem we have just explored?

With hierarchical , we don't have to specify the number of clusters and allow for the model to find the best number of clusters to fit the data.