NAIVE BAYES CLASSIFIER

```
install.packages("reshape2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("ggcorrplot")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("e1071")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(reshape2)
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyve
rse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ lubridate 1.9.3     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_co
nflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to fo
rce all conflicts to become errors
```

```
library(lattice)
library(dplyr)
library(reshape2)
library(ggcorrplot)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at htt
ps://goo.gl/ve3WBa
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
```

```
##
##     lift
```

```
library(e1071)
data=read_csv('data.csv')
```

```
## New names:
## • `` -> `...33`
```

```
## Warning: One or more parsing issues, call `problems()` on your data
frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 568 Columns: 33
## ── Column specification ────────────────────────────────────────────
────────────
## Delimiter: ","
## chr  (1): diagnosis
## dbl (31): id, radius_mean, texture_mean, perimeter_mean, area_mean,
smoothne...
## lgl  (1): ...33
##
## ℹ Use `spec()` to retrieve the full column specification for this d
ata.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
 this message.
```

```
print(data)
```

```
## # A tibble: 568 × 33
##          id diagnosis radius_mean texture_mean perimeter_mean area_m
ean
##       <dbl> <chr>           <dbl>        <dbl>          <dbl>     <d
bl>
##  1   842302 M                18.0         10.4          123.      10
01
##  2   842517 M                20.6         17.8          133.      13
26
##  3 84300903 M                19.7         21.2          130       12
03
##  4 84348301 M                11.4         20.4           77.6      3
86.
##  5 84358402 M                20.3         14.3          135.      12
97
##  6   843786 M                12.4         15.7           82.6      4
77.
##  7   844359 M                18.2         20.0          120.      10
40
##  8 84458202 M                13.7         20.8           90.2      5
```

```
78.
##  9   844981 M                13           21.8           87.5      5
20.
## 10 84501001 M                12.5         24.0           84.0      4
76.
## # ℹ 558 more rows
## # ℹ 27 more variables: smoothness_mean <dbl>, compactness_mean <db
l>,
## #   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` <d
bl>,
## #   symmetry_se <dbl>, fractal_dimension_se <dbl>, radius_worst <db
l>, …

#Structure
str(data)

## spc_tbl_ [568 × 33] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id                      : num [1:568] 842302 842517 84300903 84348
301 84358402 ...
##  $ diagnosis               : chr [1:568] "M" "M" "M" "M" ...
##  $ radius_mean             : num [1:568] 18 20.6 19.7 11.4 20.3 ...
##  $ texture_mean            : num [1:568] 10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean          : num [1:568] 122.8 132.9 130 77.6 135.1
 ...
##  $ area_mean               : num [1:568] 1001 1326 1203 386 1297 ...
##  $ smoothness_mean         : num [1:568] 0.1184 0.0847 0.1096 0.1425
0.1003 ...
##  $ compactness_mean        : num [1:568] 0.2776 0.0786 0.1599 0.2839
0.1328 ...
##  $ concavity_mean          : num [1:568] 0.3001 0.0869 0.1974 0.2414
0.198 ...
##  $ concave points_mean     : num [1:568] 0.1471 0.0702 0.1279 0.1052
0.1043 ...
##  $ symmetry_mean           : num [1:568] 0.242 0.181 0.207 0.26 0.181
 ...
##  $ fractal_dimension_mean : num [1:568] 0.0787 0.0567 0.06 0.0974 0.
0588 ...
##  $ radius_se               : num [1:568] 1.095 0.543 0.746 0.496 0.75
7 ...
##  $ texture_se              : num [1:568] 0.905 0.734 0.787 1.156 0.78
1 ...
##  $ perimeter_se            : num [1:568] 8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                 : num [1:568] 153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se           : num [1:568] 0.0064 0.00522 0.00615 0.009
11 0.01149 ...
##  $ compactness_se          : num [1:568] 0.049 0.0131 0.0401 0.0746 0.
```

```
0246 ...
##  $ concavity_se           : num [1:568] 0.0537 0.0186 0.0383 0.0566
0.0569 ...
##  $ concave points_se      : num [1:568] 0.0159 0.0134 0.0206 0.0187
0.0188 ...
##  $ symmetry_se            : num [1:568] 0.03 0.0139 0.0225 0.0596 0.
0176 ...
##  $ fractal_dimension_se   : num [1:568] 0.00619 0.00353 0.00457 0.00
921 0.00511 ...
##  $ radius_worst           : num [1:568] 25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst          : num [1:568] 17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst        : num [1:568] 184.6 158.8 152.5 98.9 152.2
 ...
##  $ area_worst             : num [1:568] 2019 1956 1709 568 1575 ...
##  $ smoothness_worst       : num [1:568] 0.162 0.124 0.144 0.21 0.137
 ...
##  $ compactness_worst      : num [1:568] 0.666 0.187 0.424 0.866 0.20
5 ...
##  $ concavity_worst        : num [1:568] 0.712 0.242 0.45 0.687 0.4
 ...
##  $ concave points_worst   : num [1:568] 0.265 0.186 0.243 0.258 0.16
3 ...
##  $ symmetry_worst         : num [1:568] 0.46 0.275 0.361 0.664 0.236
 ...
##  $ fractal_dimension_worst: num [1:568] 0.1189 0.089 0.0876 0.173 0.
0768 ...
##  $ ...33                  : logi [1:568] NA NA NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   diagnosis = col_character(),
##   ..   radius_mean = col_double(),
##   ..   texture_mean = col_double(),
##   ..   perimeter_mean = col_double(),
##   ..   area_mean = col_double(),
##   ..   smoothness_mean = col_double(),
##   ..   compactness_mean = col_double(),
##   ..   concavity_mean = col_double(),
##   ..   `concave points_mean` = col_double(),
##   ..   symmetry_mean = col_double(),
##   ..   fractal_dimension_mean = col_double(),
##   ..   radius_se = col_double(),
##   ..   texture_se = col_double(),
##   ..   perimeter_se = col_double(),
##   ..   area_se = col_double(),
##   ..   smoothness_se = col_double(),
##   ..   compactness_se = col_double(),
##   ..   concavity_se = col_double(),
##   ..   `concave points_se` = col_double(),
##   ..   symmetry_se = col_double(),
```

```
##   ..   fractal_dimension_se = col_double(),
##   ..   radius_worst = col_double(),
##   ..   texture_worst = col_double(),
##   ..   perimeter_worst = col_double(),
##   ..   area_worst = col_double(),
##   ..   smoothness_worst = col_double(),
##   ..   compactness_worst = col_double(),
##   ..   concavity_worst = col_double(),
##   ..   `concave points_worst` = col_double(),
##   ..   symmetry_worst = col_double(),
##   ..   fractal_dimension_worst = col_double(),
##   ..   ...33 = col_logical()
##   .. )
##  - attr(*, "problems")=<externalptr>

#Dimension
dim(data)

## [1] 568  33

#First few rows
head(data)

## # A tibble: 6 × 33
##         id diagnosis radius_mean texture_mean perimeter_mean area_me
an
##      <dbl> <chr>           <dbl>        <dbl>          <dbl>     <db
l>
## 1   842302 M                18.0         10.4          123.      100
1
## 2   842517 M                20.6         17.8          133.      132
6
## 3 84300903 M                19.7         21.2          130       120
3
## 4 84348301 M                11.4         20.4           77.6      38
6.
## 5 84358402 M                20.3         14.3          135.      129
7
## 6   843786 M                12.4         15.7           82.6      47
7.
## # ℹ 27 more variables: smoothness_mean <dbl>, compactness_mean <db
l>,
## #   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` <d
bl>,
## #   symmetry_se <dbl>, fractal_dimension_se <dbl>, radius_worst <db
l>,
## #   texture_worst <dbl>, perimeter_worst <dbl>, area_worst <dbl>, …
```

```r
#Summary
summary(data)
```

```
##        id              diagnosis          radius_mean      texture_me
an
##  Min.   :     8670   Length:568         Min.   : 6.981   Min.   : 9.
71
##  1st Qu.:   869222   Class :character   1st Qu.:11.707   1st Qu.:16.
17
##  Median :   906157   Mode  :character   Median :13.375   Median :18.
84
##  Mean   : 30425140                      Mean   :14.139   Mean   :19.
28
##  3rd Qu.:  8825022                      3rd Qu.:15.797   3rd Qu.:21.
79
##  Max.   :911320502                      Max.   :28.110   Max.   :39.
28
##  perimeter_mean     area_mean      smoothness_mean   compactness_mea
n
##  Min.   : 43.79   Min.   : 143.5   Min.   :0.06251   Min.   :0.01938

##  1st Qu.: 75.20   1st Qu.: 420.3   1st Qu.:0.08640   1st Qu.:0.06517

##  Median : 86.29   Median : 551.4   Median :0.09589   Median :0.09312

##  Mean   : 92.05   Mean   : 655.7   Mean   :0.09644   Mean   :0.10445

##  3rd Qu.:104.15   3rd Qu.: 784.1   3rd Qu.:0.10533   3rd Qu.:0.13043

##  Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.   :0.34540

##  concavity_mean    concave points_mean symmetry_mean    fractal_dime
nsion_mean
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04
996
##  1st Qu.:0.02958   1st Qu.:0.02035     1st Qu.:0.1620   1st Qu.:0.05
770
##  Median :0.06155   Median :0.03360     Median :0.1792   Median :0.06
155
##  Mean   :0.08896   Mean   :0.04901     Mean   :0.1812   Mean   :0.06
280
##  3rd Qu.:0.13100   3rd Qu.:0.07401     3rd Qu.:0.1957   3rd Qu.:0.06
613
##  Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09
744
##    radius_se        texture_se      perimeter_se       area_se

##  Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
```

```
##  1st Qu.:0.2324   1st Qu.:0.8331   1st Qu.: 1.605   1st Qu.: 17.850

##  Median :0.3240   Median :1.1080   Median : 2.285   Median : 24.565

##  Mean   :0.4052   Mean   :1.2165   Mean   : 2.867   Mean   : 40.374

##  3rd Qu.:0.4798   3rd Qu.:1.4743   3rd Qu.: 3.360   3rd Qu.: 45.237

##  Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200

##  smoothness_se      compactness_se      concavity_se     concave poi
nts_se
##  Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   Min.   :0.0
00000
##  1st Qu.:0.005166   1st Qu.:0.013133   1st Qu.:0.01510   1st Qu.:0.0
07663
##  Median :0.006374   Median :0.020460   Median :0.02592   Median :0.0
10950
##  Mean   :0.007041   Mean   :0.025515   Mean   :0.03195   Mean   :0.0
11817
##  3rd Qu.:0.008151   3rd Qu.:0.032455   3rd Qu.:0.04212   3rd Qu.:0.0
14730
##  Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   Max.   :0.0
52790
##   symmetry_se       fractal_dimension_se  radius_worst   texture_wor
st
##  Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.
02
##  1st Qu.:0.015128   1st Qu.:0.0022445    1st Qu.:13.03   1st Qu.:21.
07
##  Median :0.018725   Median :0.0031955    Median :14.97   Median :25.
41
##  Mean   :0.020531   Mean   :0.0037967    Mean   :16.28   Mean   :25.
67
##  3rd Qu.:0.023398   3rd Qu.:0.0045585    3rd Qu.:18.80   3rd Qu.:29.
68
##  Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.
54
##  perimeter_worst    area_worst     smoothness_worst  compactness_wor
st
##  Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729

##  1st Qu.: 84.15   1st Qu.: 515.7   1st Qu.:0.11660   1st Qu.:0.14758

##  Median : 97.67   Median : 686.5   Median :0.13135   Median :0.21300

##  Mean   :107.35   Mean   : 881.7   Mean   :0.13244   Mean   :0.25460

##  3rd Qu.:125.53   3rd Qu.:1085.0   3rd Qu.:0.14602   3rd Qu.:0.33930
```

```
##  Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800

##  concavity_worst  concave points_worst symmetry_worst   fractal_dime
nsion_worst
##  Min.   :0.0000   Min.   :0.00000      Min.   :0.1565   Min.   :0.05
504
##  1st Qu.:0.1159   1st Qu.:0.06497      1st Qu.:0.2504   1st Qu.:0.07
147
##  Median :0.2275   Median :0.10002      Median :0.2821   Median :0.08
005
##  Mean   :0.2727   Mean   :0.11481      Mean   :0.2901   Mean   :0.08
397
##  3rd Qu.:0.3835   3rd Qu.:0.16168      3rd Qu.:0.3180   3rd Qu.:0.09
208
##  Max.   :1.2520   Max.   :0.29100      Max.   :0.6638   Max.   :0.20
750
##   ...33
##  Mode:logical
##  NA's:568
##
##
##
##

#Column names
colnames(data)

##  [1] "id"                      "diagnosis"
##  [3] "radius_mean"             "texture_mean"
##  [5] "perimeter_mean"          "area_mean"
##  [7] "smoothness_mean"         "compactness_mean"
##  [9] "concavity_mean"          "concave points_mean"
## [11] "symmetry_mean"           "fractal_dimension_mean"
## [13] "radius_se"               "texture_se"
## [15] "perimeter_se"            "area_se"
## [17] "smoothness_se"           "compactness_se"
## [19] "concavity_se"            "concave points_se"
## [21] "symmetry_se"             "fractal_dimension_se"
## [23] "radius_worst"            "texture_worst"
## [25] "perimeter_worst"         "area_worst"
## [27] "smoothness_worst"        "compactness_worst"
## [29] "concavity_worst"         "concave points_worst"
## [31] "symmetry_worst"          "fractal_dimension_worst"
## [33] "...33"

#Checking mising values
is.null(data)

## [1] FALSE
```

```r
sum(is.na(data))
```

```
## [1] 568
```

```r
# Drop the X column
df1 <- subset(data, select = ...33)
df1
```

```
## # A tibble: 568 × 1
##    ...33
##    <lgl>
##  1 NA
##  2 NA
##  3 NA
##  4 NA
##  5 NA
##  6 NA
##  7 NA
##  8 NA
##  9 NA
## 10 NA
## # ℹ 558 more rows
```

```r
#drop id column
df2 = subset(data, select = -id)
df2
```

```
## # A tibble: 568 × 32
##    diagnosis radius_mean texture_mean perimeter_mean area_mean smoot
hness_mean
##    <chr>           <dbl>        <dbl>          <dbl>     <dbl>
     <dbl>
##  1 M                18.0         10.4          123.      1001
    0.118
##  2 M                20.6         17.8          133.      1326
    0.0847
##  3 M                19.7         21.2          130       1203
    0.110
##  4 M                11.4         20.4           77.6      386.
    0.142
##  5 M                20.3         14.3          135.      1297
    0.100
##  6 M                12.4         15.7           82.6      477.
    0.128
##  7 M                18.2         20.0          120.      1040
    0.0946
##  8 M                13.7         20.8           90.2      578.
    0.119
##  9 M                13           21.8           87.5      520.
    0.127
## 10 M                12.5         24.0           84.0      476.
```

```
    0.119
## # ℹ 558 more rows
## # ℹ 26 more variables: 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` <d
bl>,
## #   symmetry_se <dbl>, fractal_dimension_se <dbl>, radius_worst <db
l>, …
```

```
#Subsetting
unique_diagnosis <- unique(data$diagnosis)
unique_diagnosis
```

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

```
#split dataset as M & B
# Splitting data based on diagnosis
mdf <- subset(data, diagnosis == "M")
bdf<- subset(data, diagnosis == "B")
#Viewing benign
dim(bdf)
```

```
## [1] 356  33
```

```
#Viewing malignant
dim(mdf)
```

```
## [1] 212  33
```

```
#type of attributes
sapply(data, class)
```

```
##                      id               diagnosis             radius_m
ean
##               "numeric"             "character"               "numer
ic"
##            texture_mean          perimeter_mean               area_m
ean
##               "numeric"               "numeric"               "numer
ic"
##         smoothness_mean        compactness_mean          concavity_m
ean
##               "numeric"               "numeric"               "numer
ic"
##     concave points_mean           symmetry_mean  fractal_dimension_m
ean
##               "numeric"               "numeric"               "numer
ic"
##               radius_se              texture_se            perimeter
_se
```

```
##               "numeric"                "numeric"                "numer
ic"
##                 area_se           smoothness_se          compactness
_se
##               "numeric"                "numeric"                "numer
ic"
##            concavity_se       concave points_se             symmetry
_se
##               "numeric"                "numeric"                "numer
ic"
##    fractal_dimension_se            radius_worst           texture_wo
rst
##               "numeric"                "numeric"                "numer
ic"
##         perimeter_worst              area_worst        smoothness_wo
rst
##               "numeric"                "numeric"                "numer
ic"
##       compactness_worst         concavity_worst    concave points_wo
rst
##               "numeric"                "numeric"                "numer
ic"
##          symmetry_worst fractal_dimension_worst                    ...
33
##               "numeric"                "numeric"                "logic
al"
```

```
#EDA
table(data$diagnosis)
```

```
##
##   B   M
## 356 212
```

```
#Count of people affected with malignant and benign
ggplot(data, aes(x = diagnosis, fill = diagnosis)) +geom_bar() +
  labs(title = "Count of people affected with malignant and benign",
       x = "Diagnosis",
       y = "Count")
```

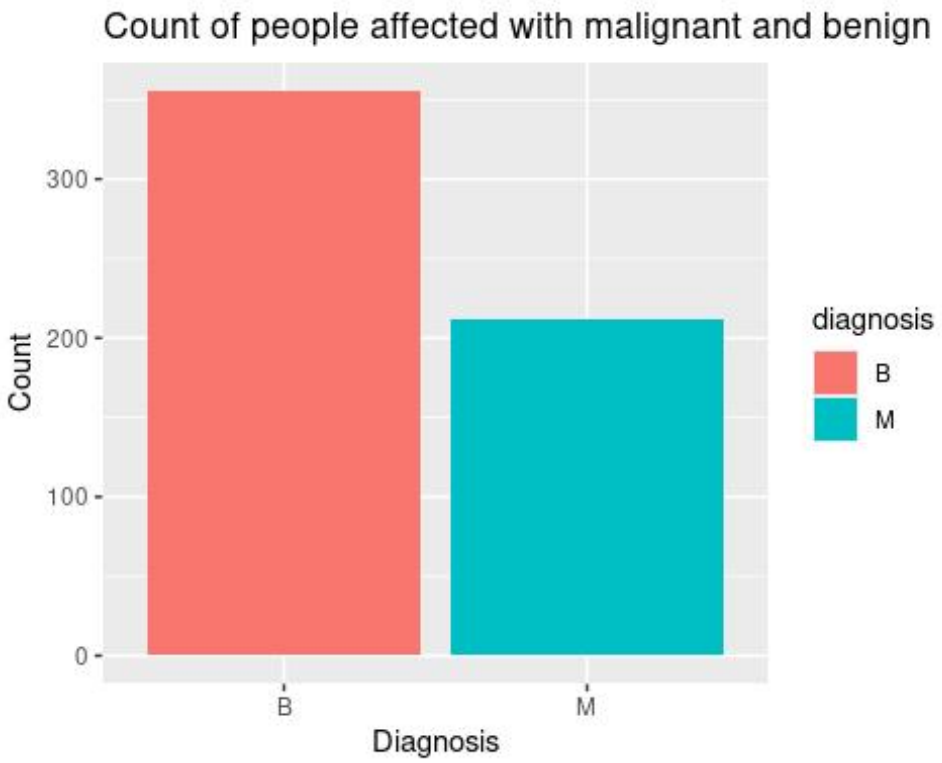
Count of people affected with malignant and benign
300
200
100
0
Count
B
M
Diagnosis
diagnosis
B
M

Fig 1.1

```r
#histogram of mean area
ggplot(data, aes(x = area_mean)) +geom_histogram(bins = 20, fill = "ste
elblue", color = "white") +
  labs(title = "Histogram of Mean Area",
       x = "Mean radius",
       y = "Count")
```

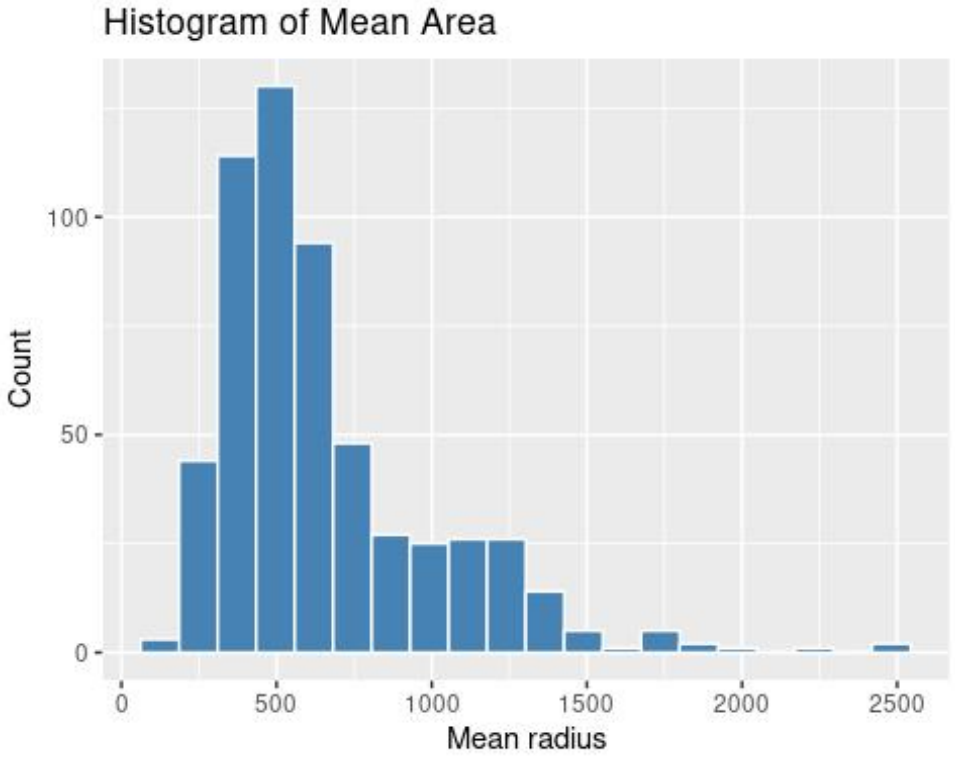

Fig 1.2

```r
#histogram of mean radius
ggplot(data, aes(x = radius_mean)) +geom_histogram(bins = 20, fill = "s
teelblue", color = "white") +
  labs(title = "Histogram of Mean Radius",
       x = "Mean Radius",
       y = "Count")
```

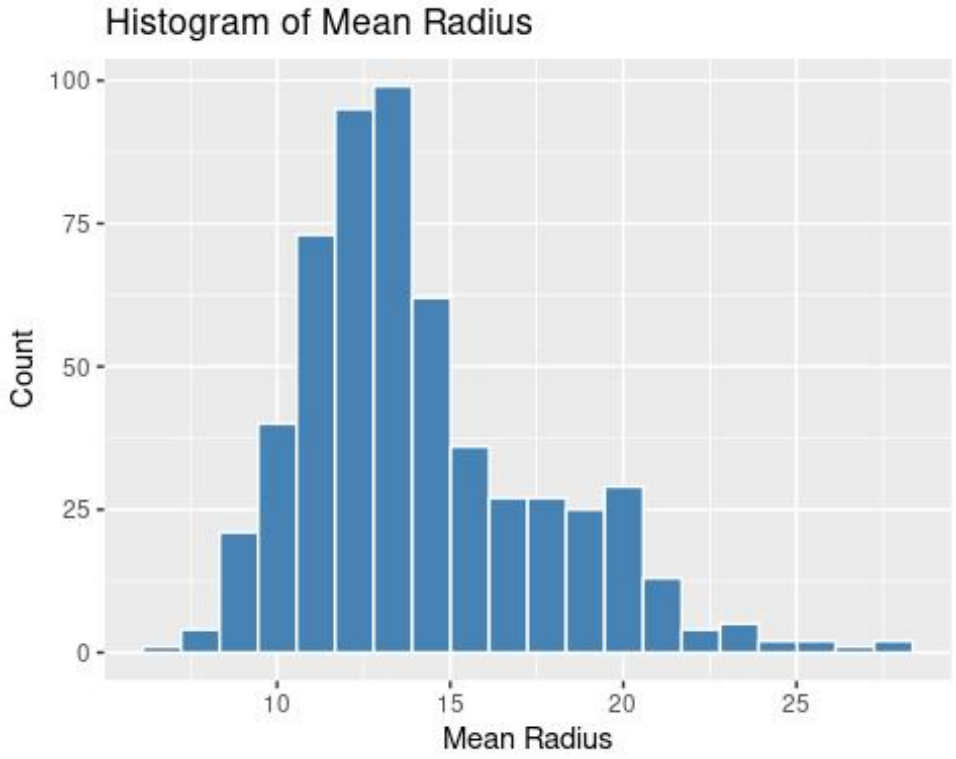

Fig 1.3

```r
#histogram of mean texture
ggplot(data, aes(x = texture_mean)) +geom_histogram(bins = 20, fill = "
steelblue", color = "white") +
  labs(title = "Histogram of Mean Texture",
       x = "Mean Texture",
       y = "Count")
```

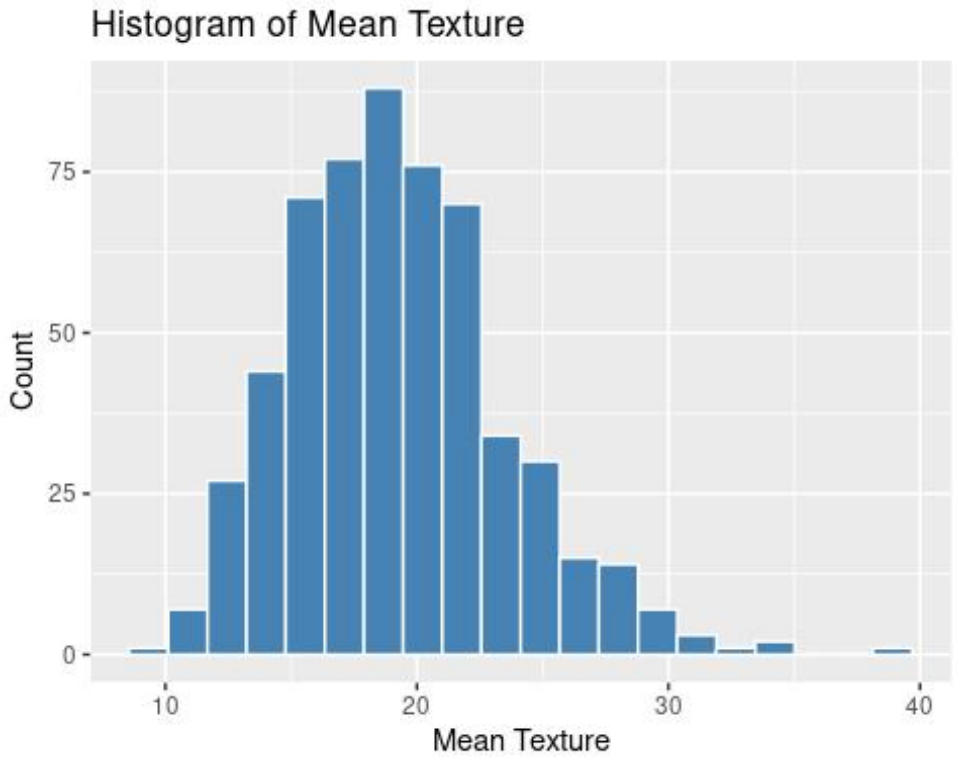

Fig 1.4

```r
#histogram of mean perimeter
ggplot(data, aes(x = perimeter_mean)) +geom_histogram(bins = 20, fill =
 "steelblue", color = "white") +
  labs(title = "Histogram of Mean Perimeter",
       x = "Mean Perimeter",
       y = "Count")
```

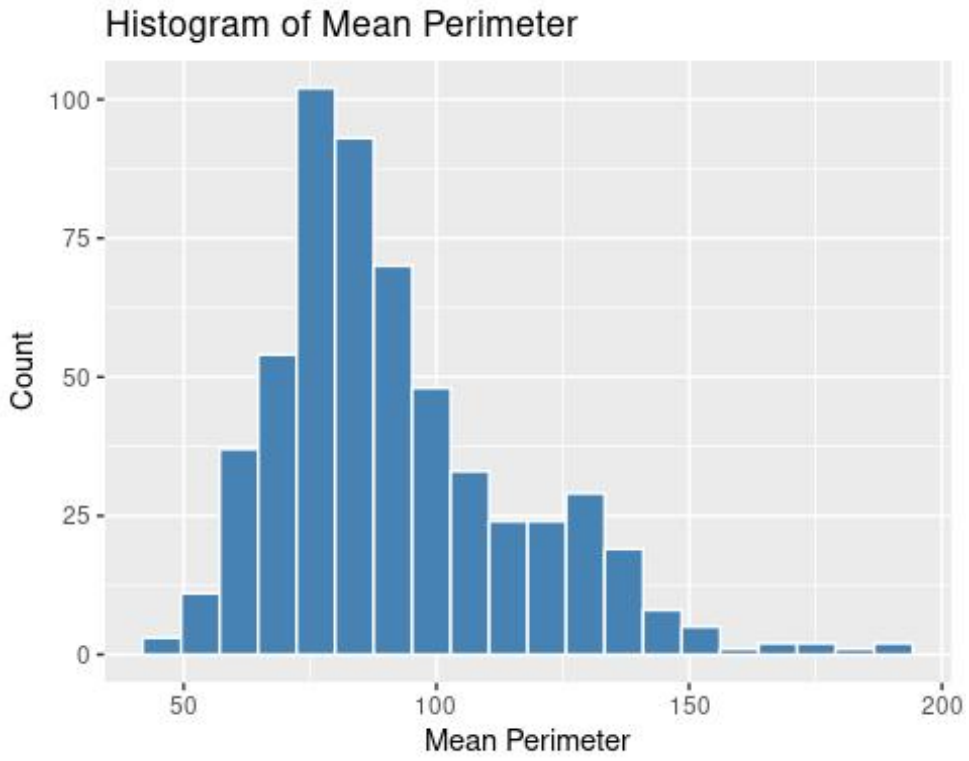

Fig 1.5

```r
#Boxplot of Mean Radius by Diagnosis
ggplot(data, aes(x = diagnosis, y = radius_mean, fill = diagnosis)) +ge
om_boxplot() +
  labs(title = "Boxplot of Mean Radius by Diagnosis",
       x = "Diagnosis",
       y = "Mean Radius")
```

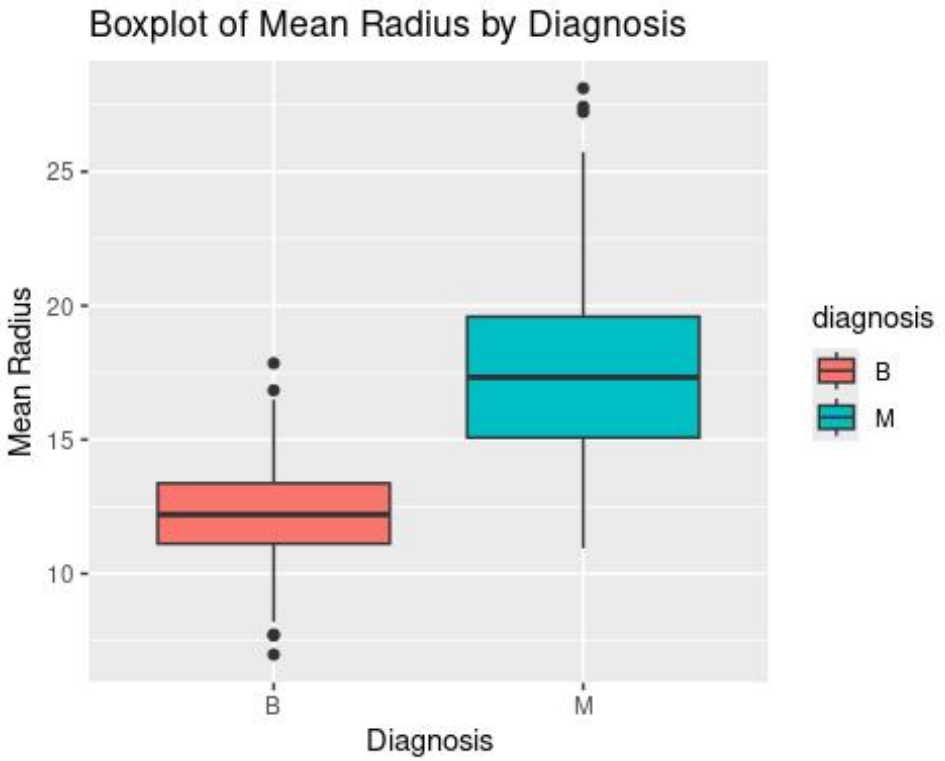

Fig 1.6

```r
#Boxplot of Mean Texture by Diagnosis
ggplot(data, aes(x = diagnosis, y =texture_mean, fill = diagnosis)) +ge
om_boxplot() +
  labs(title = "Boxplot of Mean Texture by Diagnosis",
       x = "Diagnosis",
       y = "Mean Texture")
```

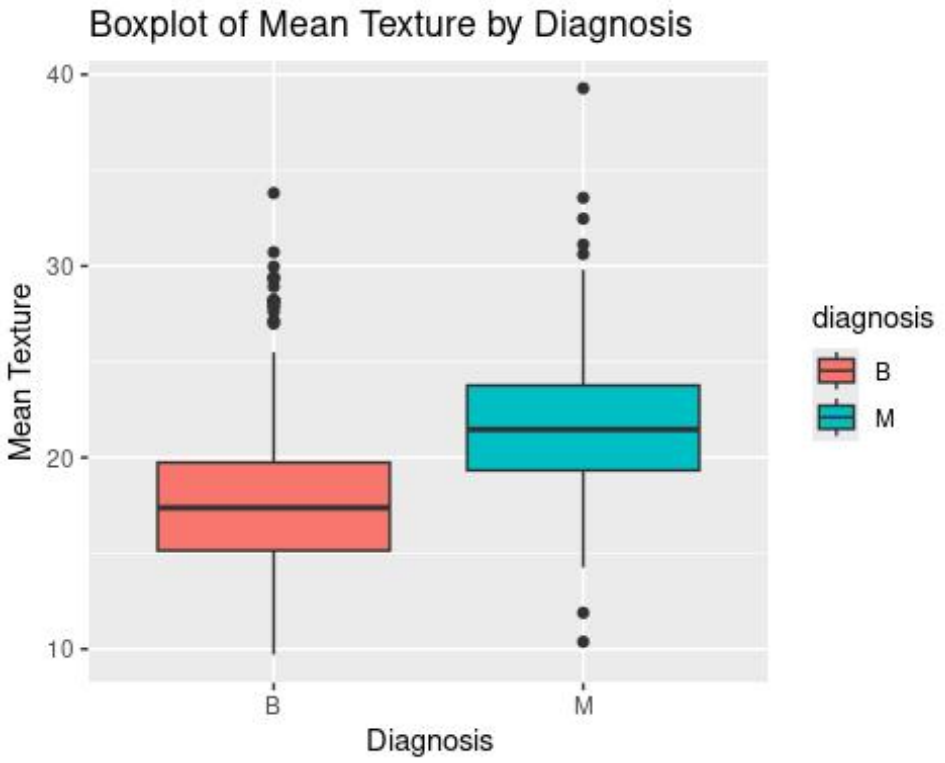

Fig 1.7

```r
# Boxplot of mean perimeter by diagnosis
ggplot(data, aes(x = diagnosis, y = perimeter_mean, fill = diagnosis))
+
  geom_boxplot() +
  labs(title = "Boxplot of Mean Perimeter by Diagnosis",
       x = "Diagnosis",
       y = "Mean Perimeter")
```

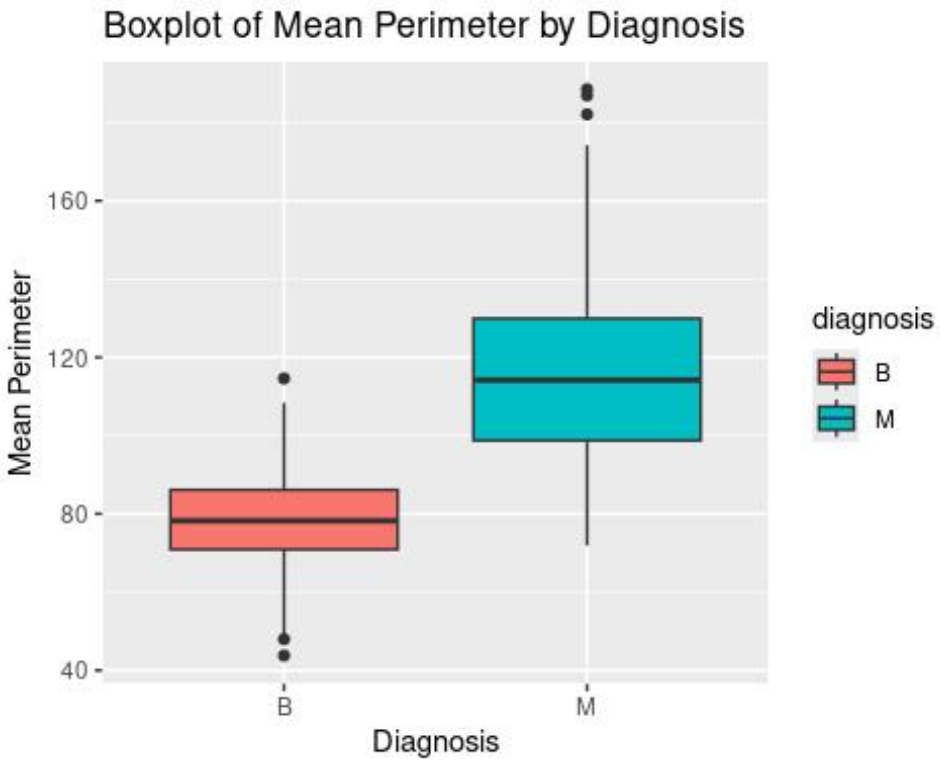

Fig 1.8

```r
# Identify numeric columns
numeric_cols <- sapply(data, is.numeric)
# Filter numeric columns
numeric_data <- data[, numeric_cols]
# Calculate the first and third quartiles
Q1 <- apply(numeric_data, 2, quantile, probs = 0.25)
Q3 <- apply(numeric_data, 2, quantile, probs = 0.75)
# Calculate the IQR (Interquartile Range)
IQR <- Q3 - Q1
# Define the lower and upper bounds for outliers detection
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
# Identify outliers indices
```

```r
outlier_indices <- sapply(numeric_data, function(x) which(x < lower_bou
nd | x > upper_bound))
```

```
## Warning in x < lower_bound: longer object length is not a multiple o
f shorter
## object length
```

```
## Warning in x > upper_bound: longer object length is not a multiple o
f shorter
## object length
```

```
## Warning in x < lower_bound: longer object length is not a multiple o
f shorter
## object length
```

```
## Warning in x > upper_bound: longer object length is not a multiple o
f shorter
## object length
```

```
## Warning in x < lower_bound: longer object length is not a multiple o
f shorter
## object length
```

```
## Warning in x > upper_bound: longer object length is not a multiple o
f shorter
## object length
```

```r
# Remove outliers
bc_no_outliers <-data[-unique(unlist(outlier_indices)), ]
# Check the dimensions of the original and new datasets
dim(data)
```

```
## [1] 568  33
```

```r
dim(bc_no_outliers)
```

```
## [1] 17 33
```

```r
#Scatter plot
ggplot(data, aes(x = radius_mean, y = texture_mean, color = diagnosis))
 +geom_point() +
  labs(title = "Scatter Plot of Radius Mean vs. Texture Mean",
       x = "Radius Mean",
       y = "Texture Mean",
       color = "Diagnosis")
```

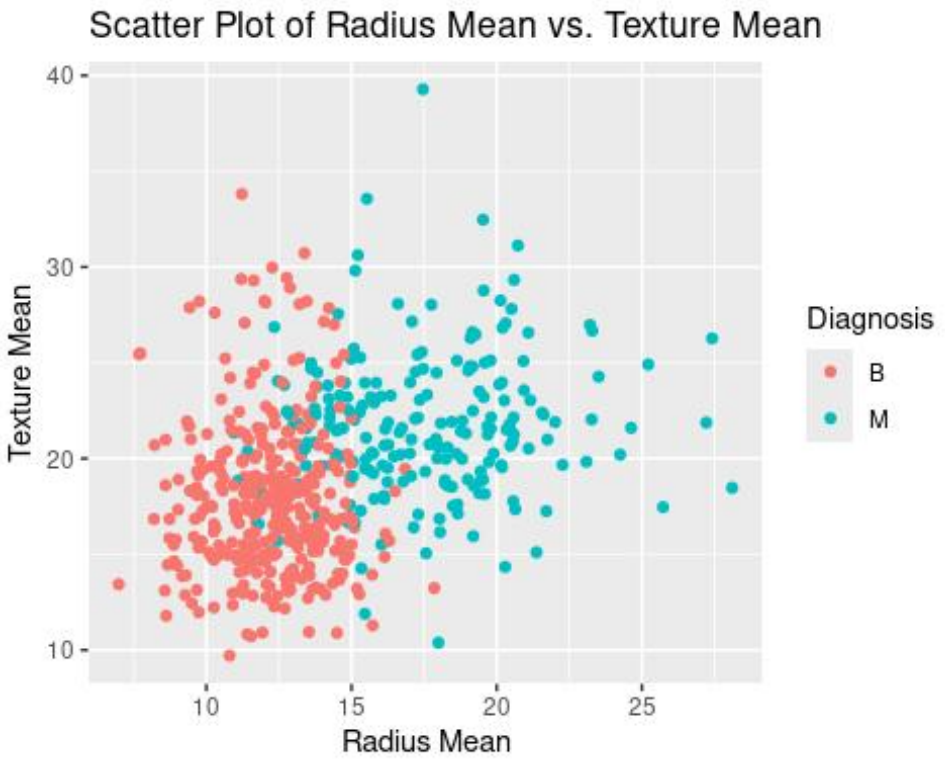

Fig 1.9

```r
ggplot(data, aes(x = radius_mean, y = area_mean, color = diagnosis)) +
  geom_point() +
  labs(title = "Scatter Plot of Radius Mean vs. Area Mean",
       x = "Radius Mean",
       y = "Area Mean",
       color = "Diagnosis")
```

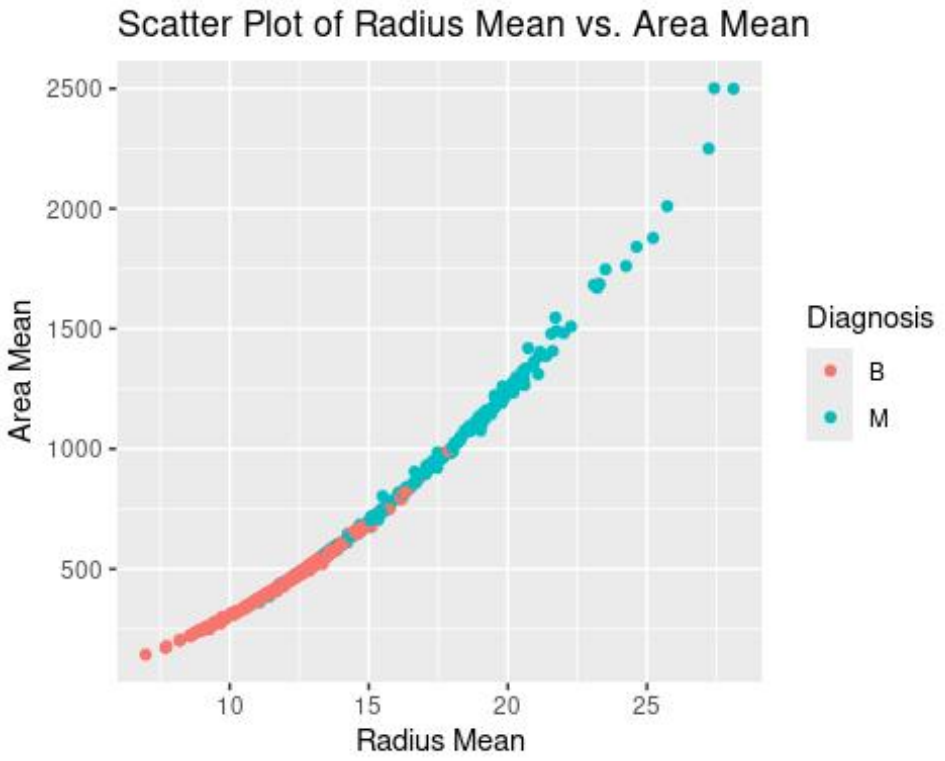

Fig 1.10

```
ggplot(data, aes(x = radius_mean, y = perimeter_mean, color = diagnosi
s)) +
  geom_point() +
  labs(title = "Scatter Plot of Concavity Mean vs. Concave Points Mean",
       x = "Concavity Mean",
       y = "Concave Points Mean",
       color = "Diagnosis")
```

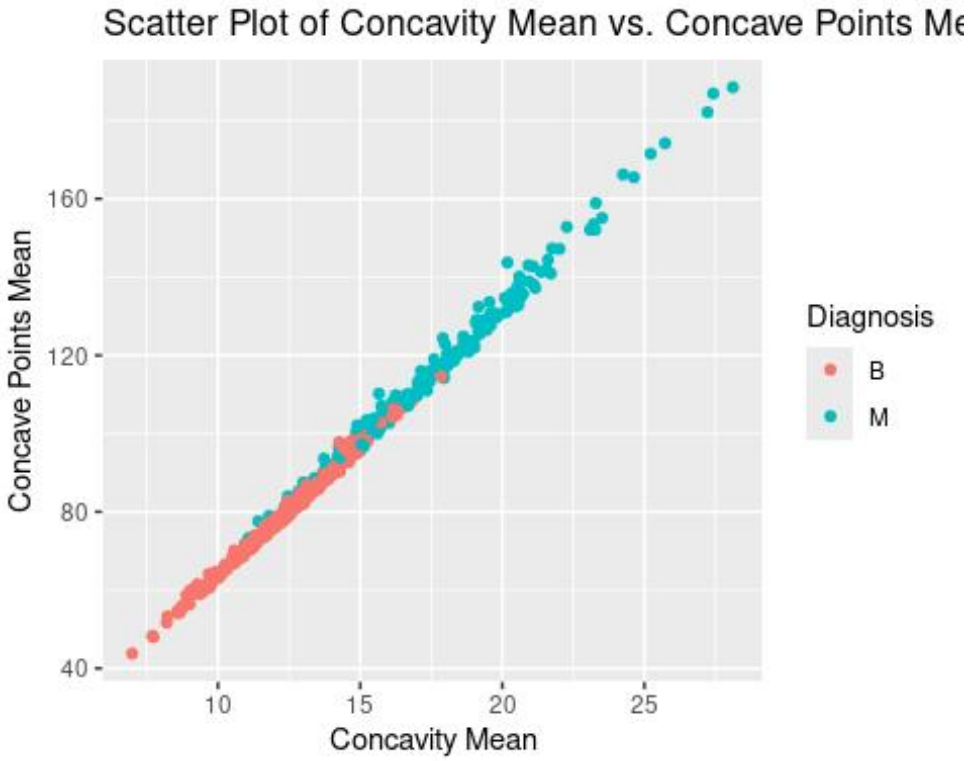

Fig 1.11

```r
#Heatmap
numeric_data <-data[, c("concavity_se","smoothness_se","perimeter_mean",
 "radius_mean", "area_mean","texture_mean", "smoothness_mean", "compact
ness_mean")]
correlation_matrix <- cor(numeric_data)
options(repr.plot.width = 8, repr.plot.height = 6)
ggcorrplot(correlation_matrix,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           method = "circle",
           colors = c("blue", "white", "red"),
           title = "Correlation Heatmap for Breast Cancer Dataset")
```

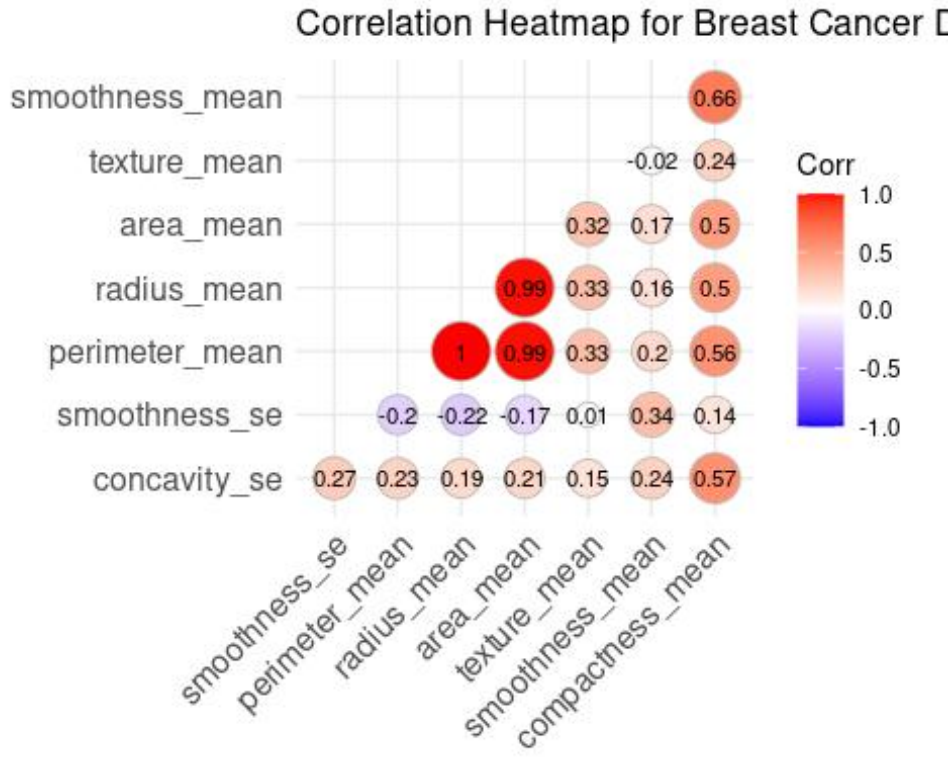

Fig 1.12

```r
# Define features and target variable
features <- c("radius_mean", "texture_mean", "perimeter_mean", "area_me
an",
              "smoothness_mean", "compactness_mean", "concavity_se","sm
oothness_se")
target <- "diagnosis"
# Create a dataframe containing both features and target variable
data <- data[, c(features, target)]
# Split the data into training and testing sets
```

```r
set.seed(123) # for reproducibility
split <- createDataPartition(data$diagnosis, p = 0.7, list = FALSE)
train_data <- data[split, ]
test_data <- data[-split, ]

# Separate features and target variable in training and testing sets
features_train <- train_data[, features]
target_train <- train_data$diagnosis
features_test <- test_data[, features]
target_test <- test_data$diagnosis

# Naive Bayes classifier
NB_model <- naiveBayes(x = features_train, y = target_train)

# Predictions
pred <- predict(NB_model, features_test)

# Accuracy calculation
accuracy <- sum(pred == target_test) / length(pred)
accuracy
```

```
## [1] 0.887574
```

```r
# Confusion Matrix
table(target_test, pred)
```

```
##             pred
## target_test  B  M
##           B 96 10
##           M  9 54
```

```r
# Multinomial Naive Bayes classifier
MNB_model <- naiveBayes(x = features_train, y = as.factor(target_train))

# Predictions
pred_mnb <- predict(MNB_model, features_test)

# Accuracy calculation
accuracy_mnb <- sum(pred_mnb == target_test) / length(pred_mnb)
accuracy_mnb
```

```
## [1] 0.887574
```