Project Code

```
library("tidyverse")
library("caret")
library("rpart")
library("partykit")
library("randomForest")
library("class")
```

```
cancer_data <-  data.table::fread("FNA_cancer.csv")
glimpse(cancer_data)
```

```
Rows: 569
Columns: 33
$ id                      <int> 842302, 842517, 84300903, 84348301, 84358402, ~
$ diagnosis               <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
$ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290, 12.450~
$ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15.70, 19.9~
$ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10, 82.57, ~
$ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0, 477.1, ~
$ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0.10030, 0~
$ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0.13280, 0~
$ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0.19800, 0~
$ `concave points_mean`   <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0.10430, 0~
$ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809, 0.2087~
$ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0.05883, 0~
$ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572, 0.3345~
$ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813, 0.8902~
$ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.217, 3.18~
$ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27.19, 53.~
$ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110, 0.0114~
$ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580, 0.0246~
$ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0~
$ `concave points_se`     <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.0188~
$ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0.01756, 0~
$ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.0051~
$ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.8~
$ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 27.6~
$ perimeter_worst         <dbl> 184.60, 158.80, 152.50, 98.87, 152.20, 103.40,~
$ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, ~
$ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374, 0.1791~
$ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050, 0.5249~
$ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0~
$ `concave points_worst`  <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0~
$ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985~
$ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0~
$ V33                     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

Process Data

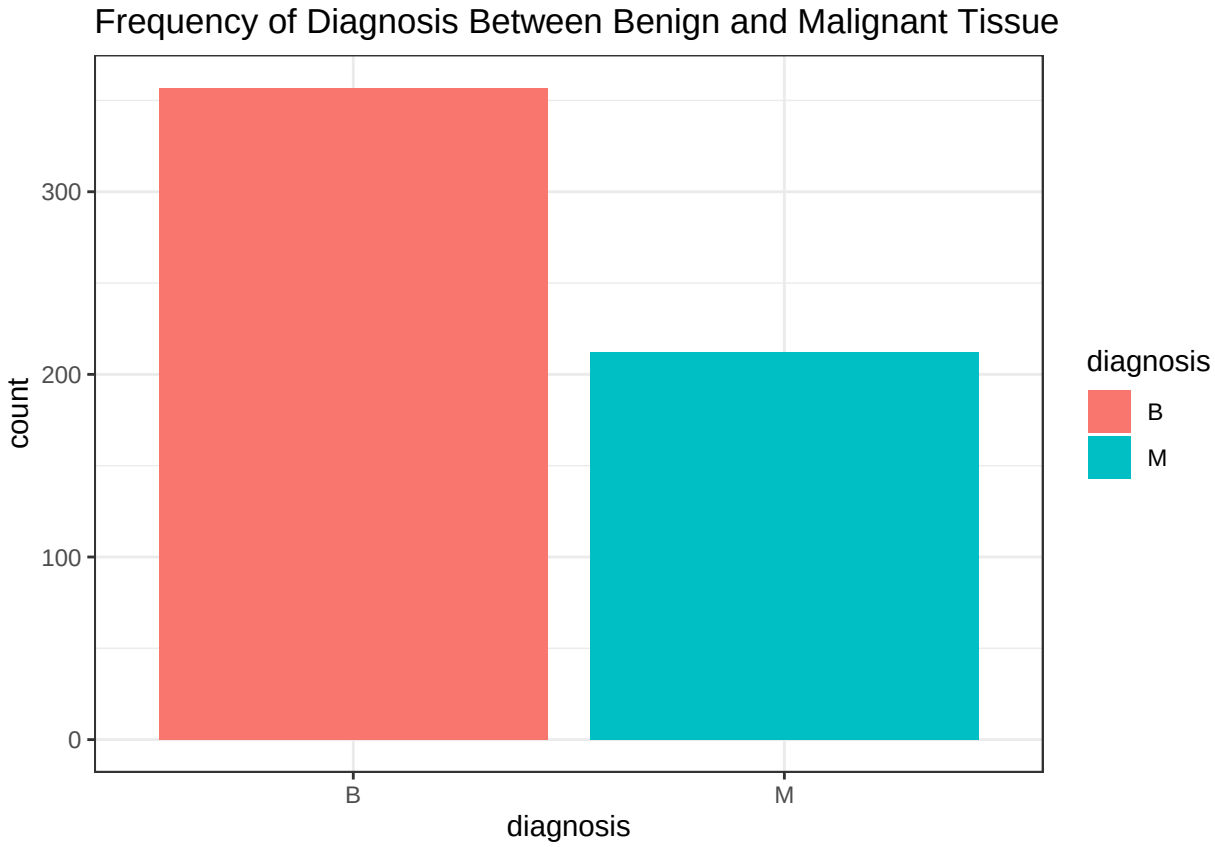
```
# Drop last column, clean certain column names, and drop all na's
cancer_data_clean <- cancer_data %>%
  select(-33) %>%
  rename("concave_points_mean" = "concave points_mean",
         "concave_points_se" = "concave points_se",
         "concave_points_worst" = "concave points_worst") %>%
  drop_na()

# Glimpse of processed data
glimpse(cancer_data_clean)
```

```
Rows: 569
Columns: 32
$ id                      <int> 842302, 842517, 84300903, 84348301, 84358402, ~
$ diagnosis               <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
$ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290, 12.450~
$ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15.70, 19.9~
$ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10, 82.57, ~
$ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0, 477.1, ~
$ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0.10030, 0~
$ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0.13280, 0~
$ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0.19800, 0~
$ concave_points_mean     <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0.10430, 0~
$ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809, 0.2087~
$ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0.05883, 0~
$ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572, 0.3345~
$ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813, 0.8902~
$ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.217, 3.18~
$ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27.19, 53.~
$ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110, 0.0114~
$ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580, 0.0246~
$ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0~
$ concave_points_se       <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.0188~
$ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0.01756, 0~
$ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.0051~
$ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.8~
$ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 27.6~
$ perimeter_worst         <dbl> 184.60, 158.80, 152.50, 98.87, 152.20, 103.40,~
$ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, ~
$ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374, 0.1791~
$ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050, 0.5249~
$ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0~
$ concave_points_worst    <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0~
$ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985~
$ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0~
```

Exploratory Data Analysis

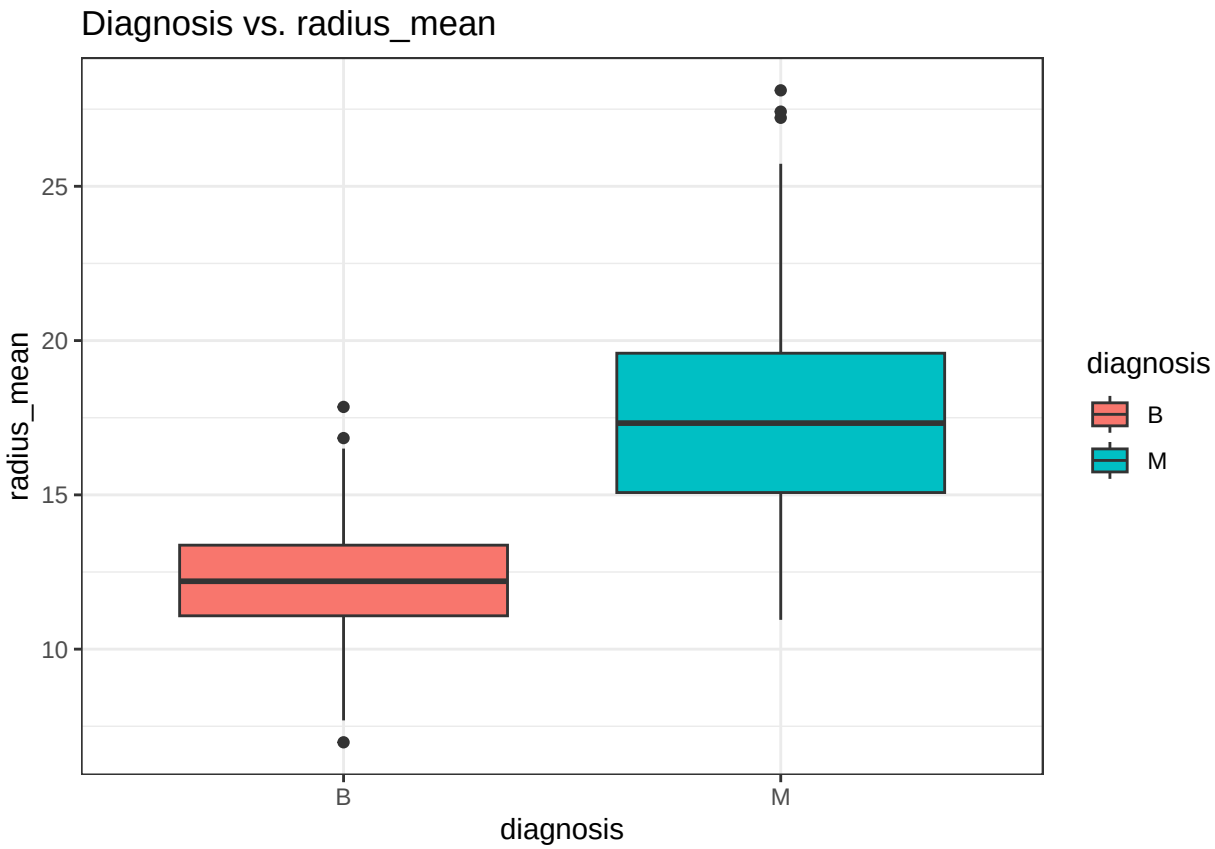
Frequency of diagnosis

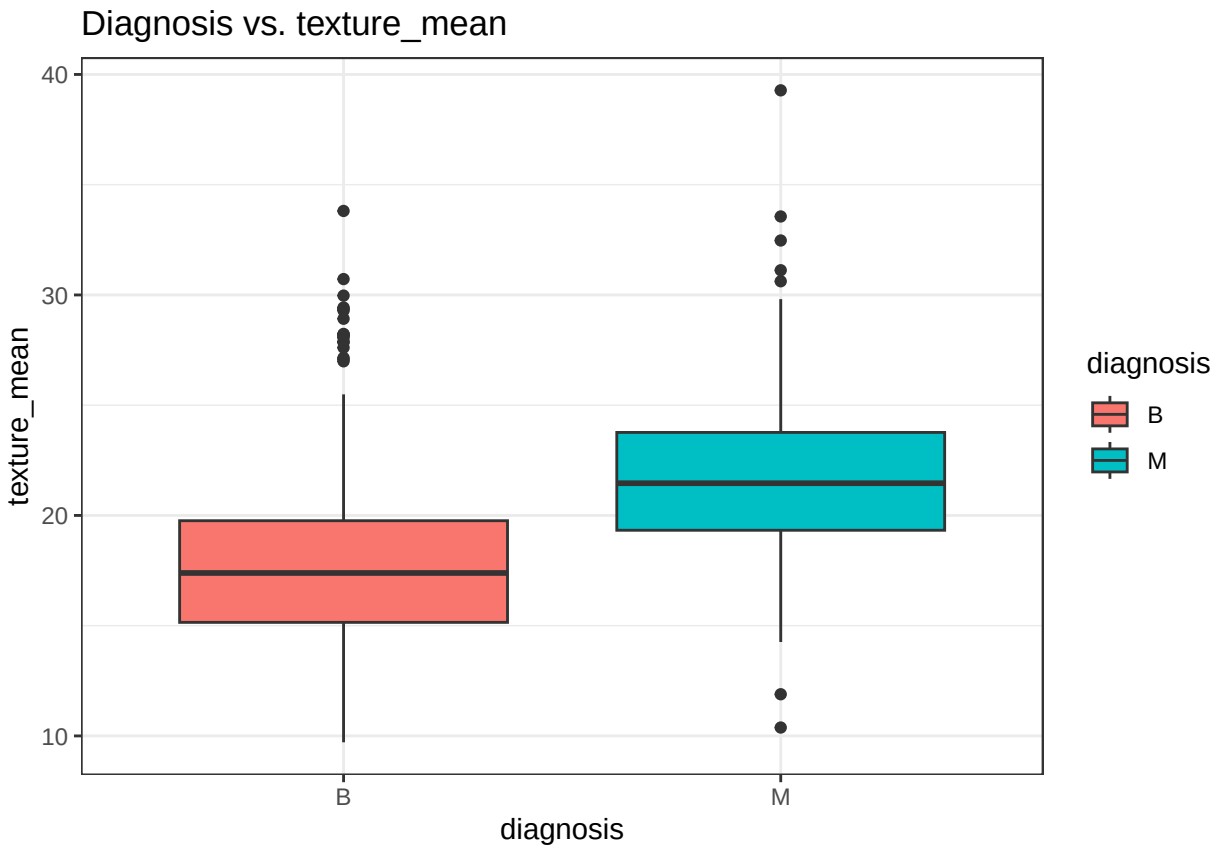
```
# Frquency of diagnosis
cancer_data_clean %>%
  ggplot(aes(x = diagnosis, fill = diagnosis)) +
  geom_bar() +
  theme_bw() +
  ggtitle(label = "Frequency of Diagnosis Between Benign and Malignant Tissue")
```

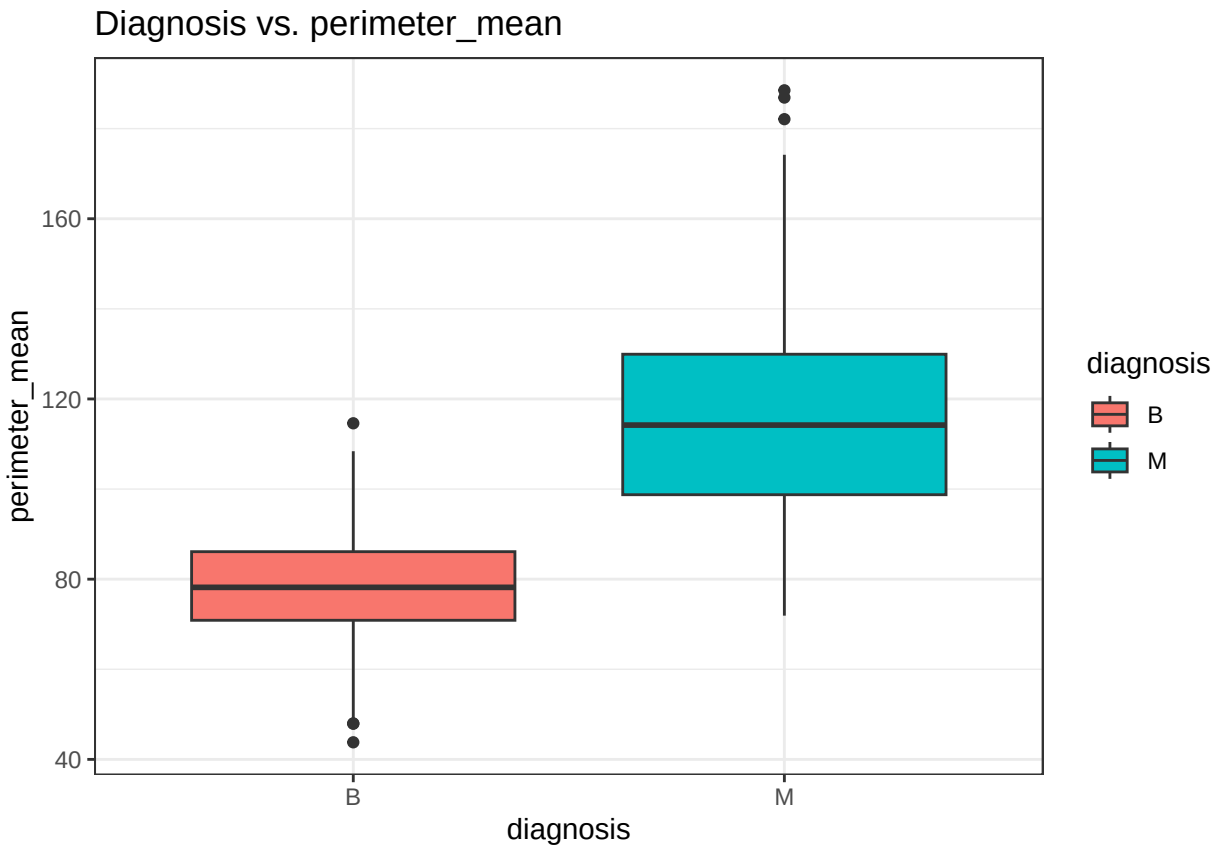


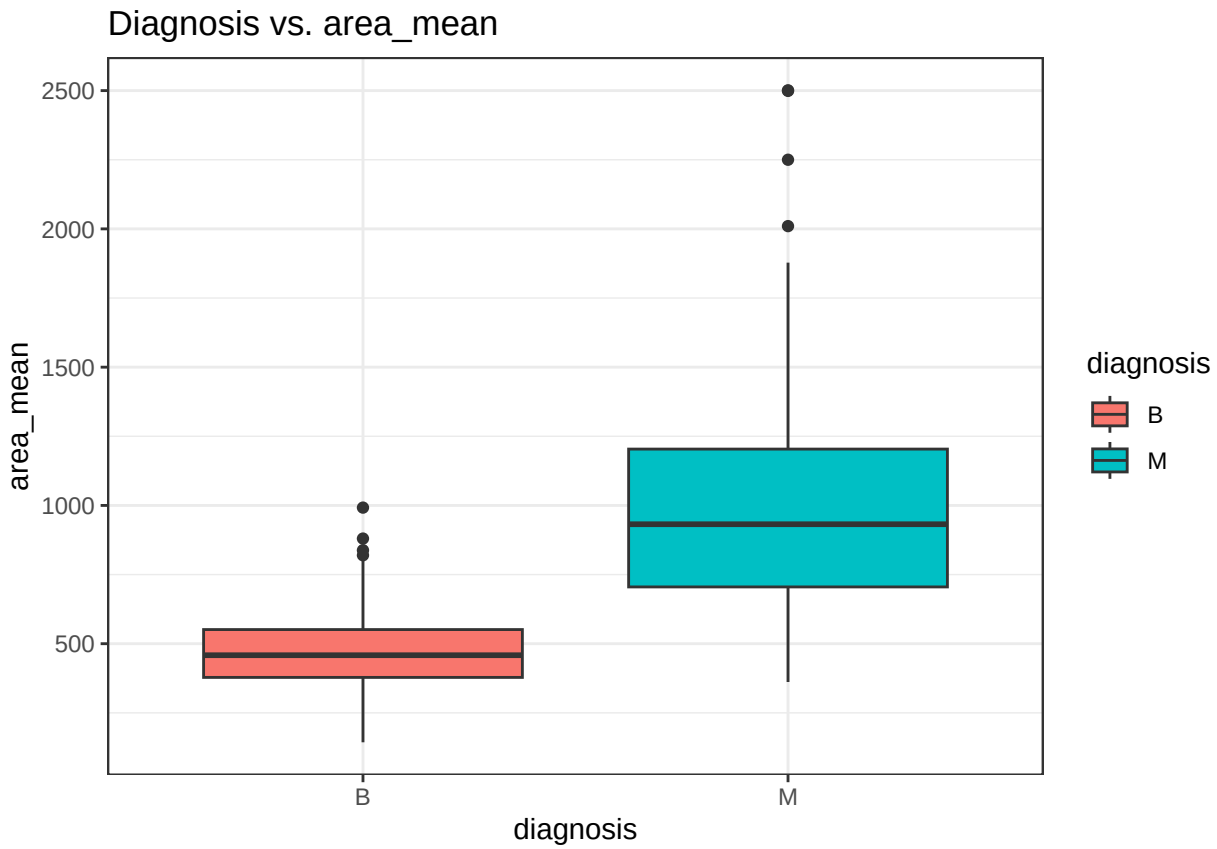
Univariate relationship between diagnosis and potential explanatory variables

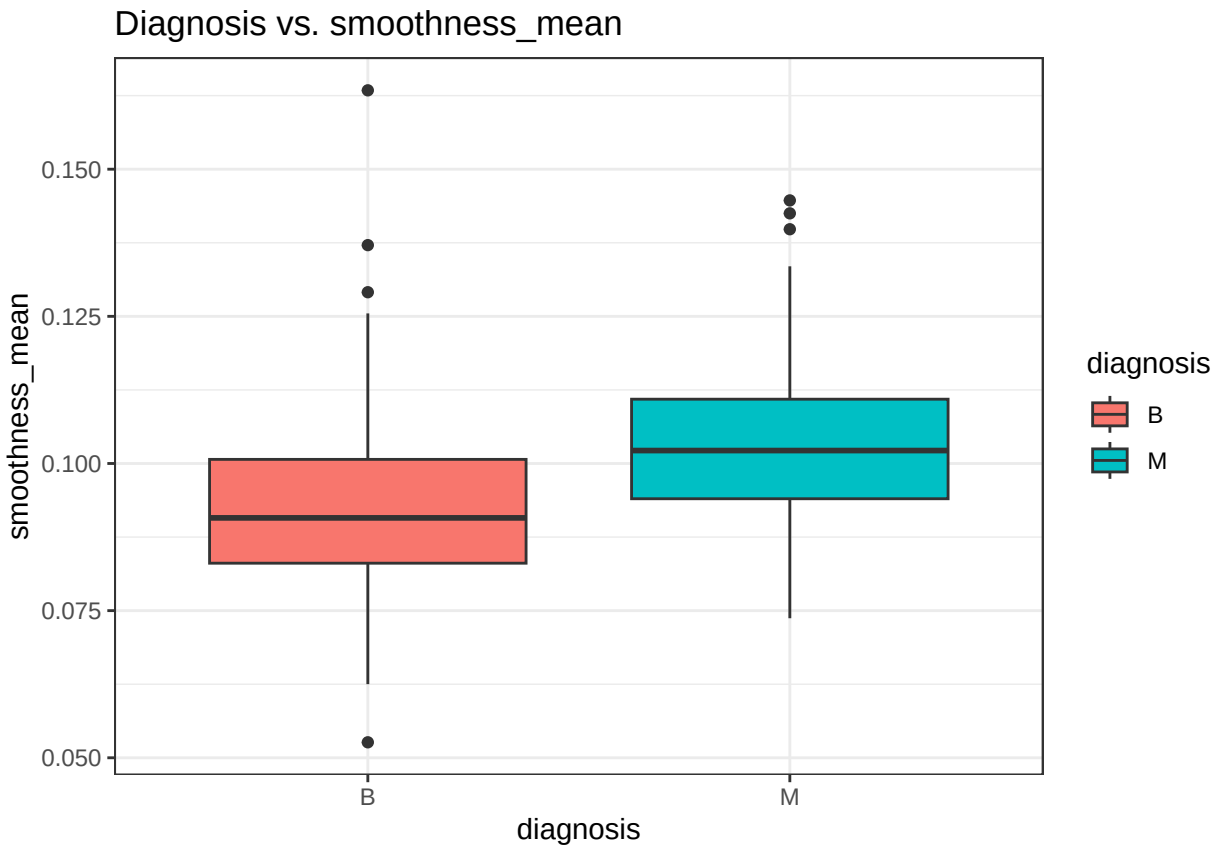
```
for (i in names(cancer_data_clean)[3:32]) {
  print(cancer_data_clean %>%
    ggplot(aes_string(x = "diagnosis", y = i, fill = "diagnosis")) +
    geom_boxplot() +
    theme_bw() +
    ggtitle(label = paste0("Diagnosis vs. ", i)))
}
```

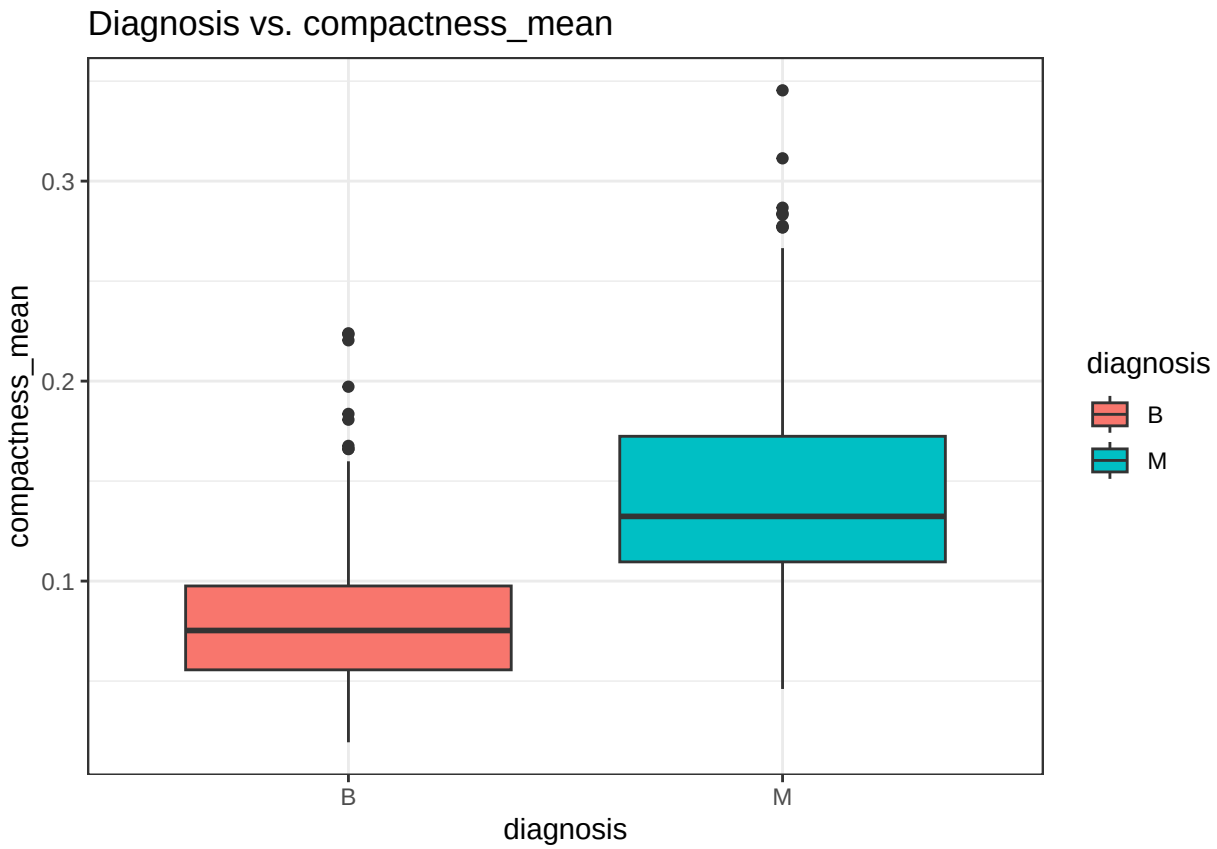

Diagnosis vs. radius_mean
radius_mean
25
20
15
10
B
M
diagnosis
diagnosis
B
M

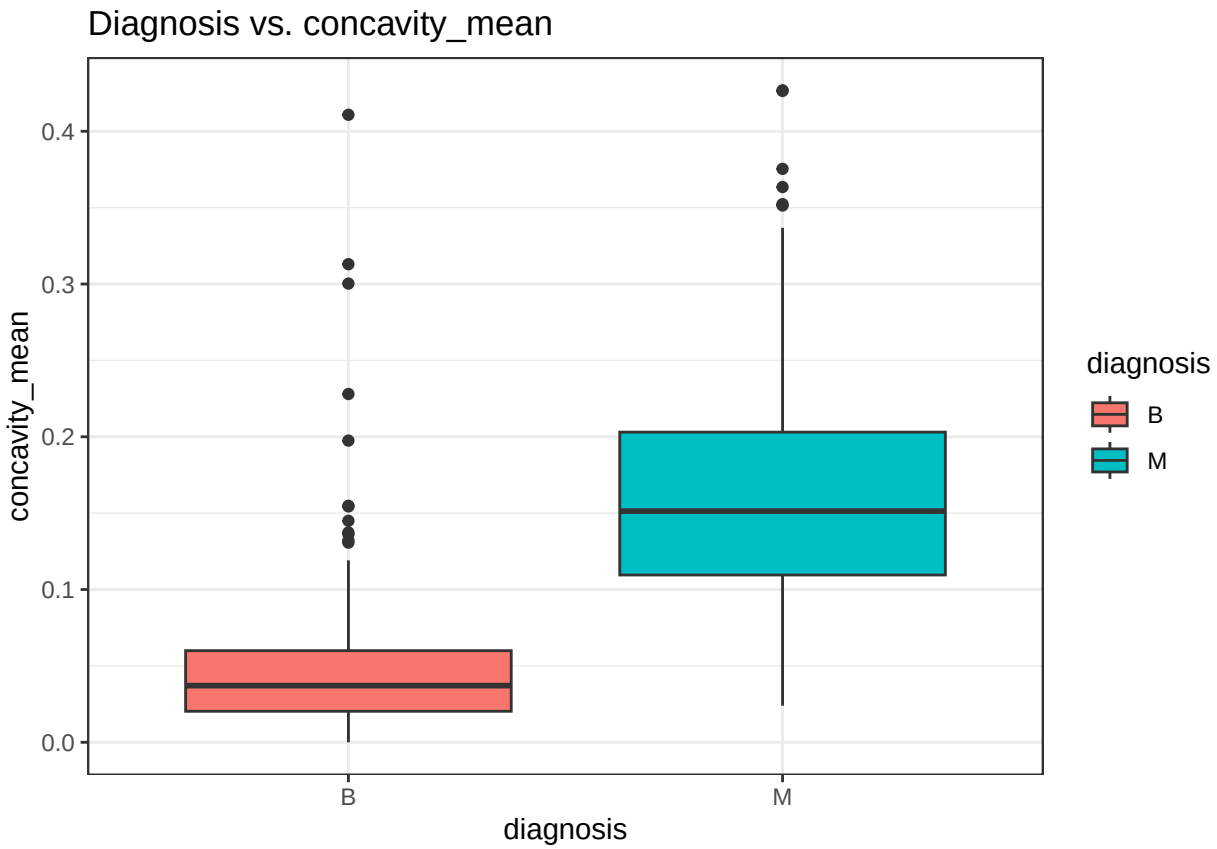

Diagnosis vs. texture_mean
texture_mean
40
30
20
10
B
M
diagnosis
diagnosis
B
M

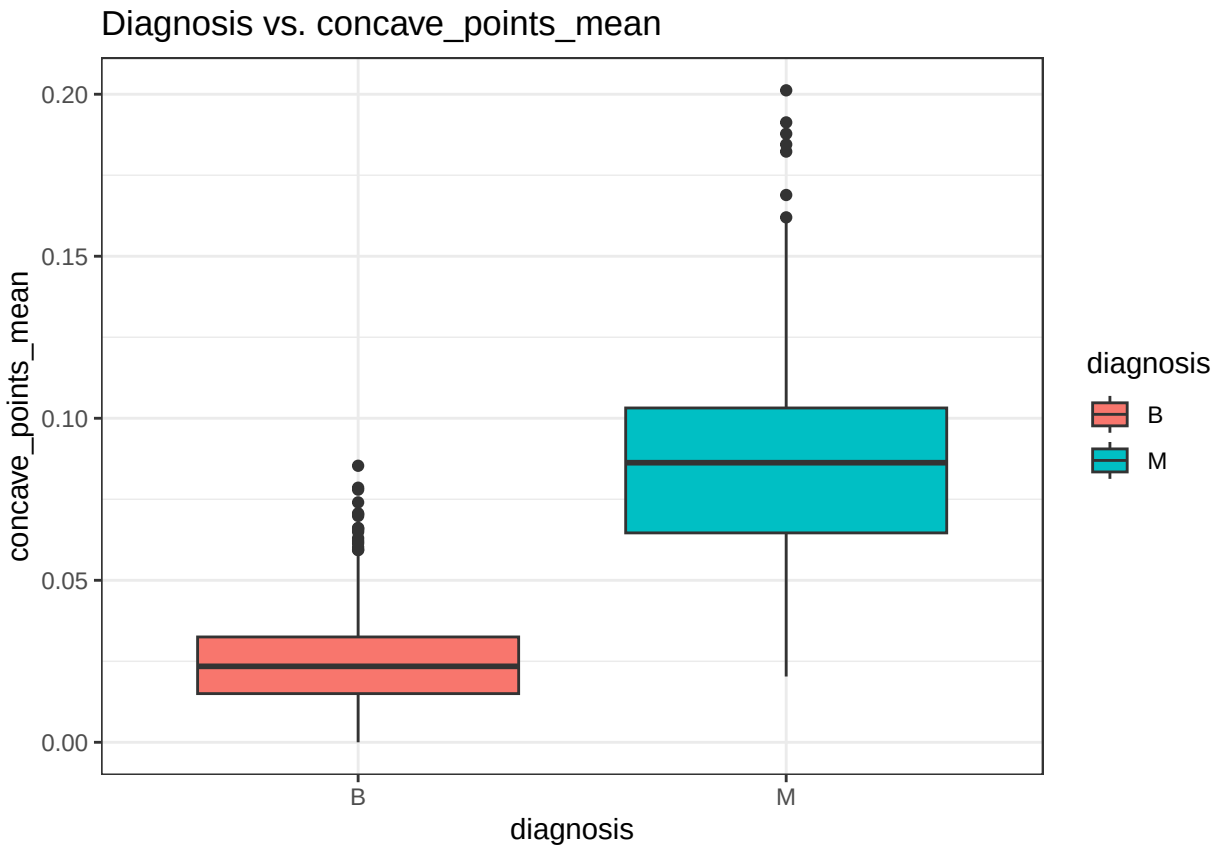

Diagnosis vs. perimeter_mean
perimeter_mean
160
120
80
40
B
M
diagnosis
diagnosis
B
M

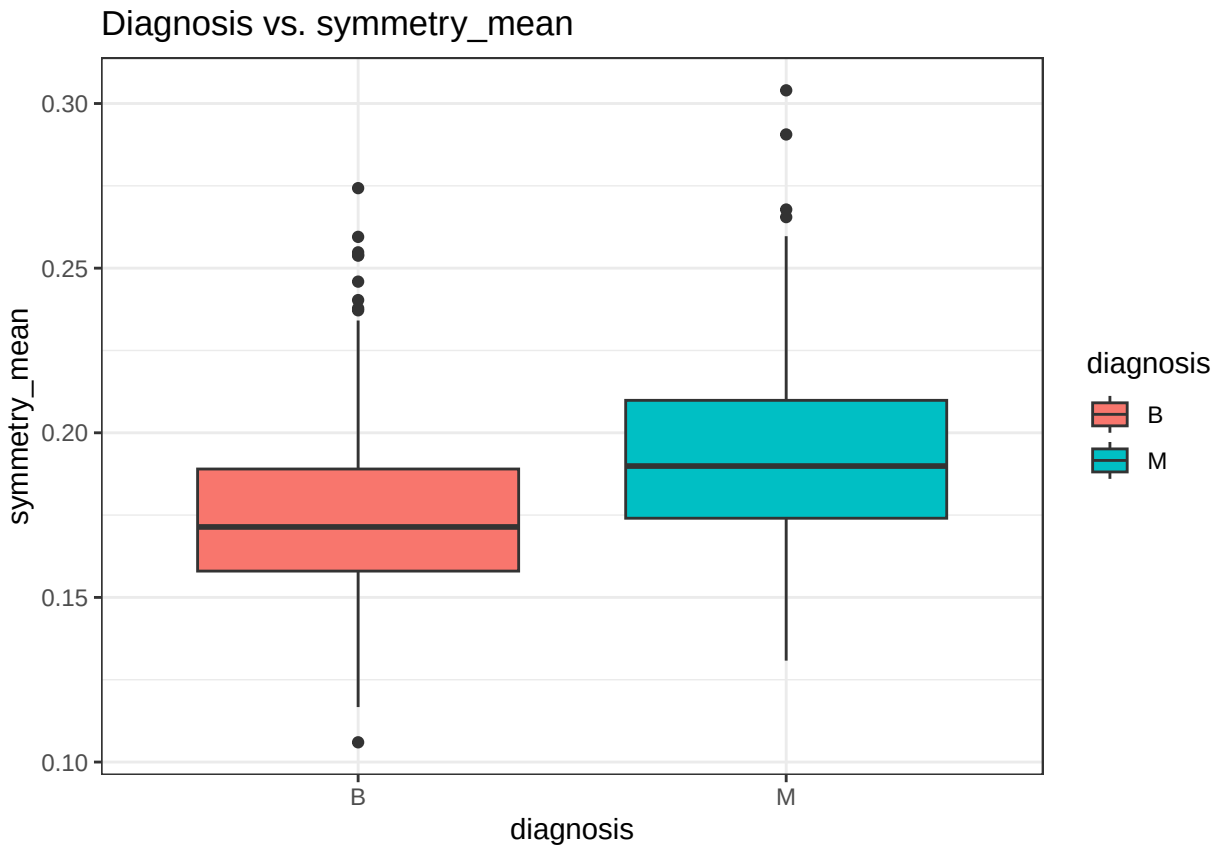

Diagnosis vs. area_mean
2500
2000
1500
1000
500
area_mean
B
M
diagnosis
diagnosis
B
M

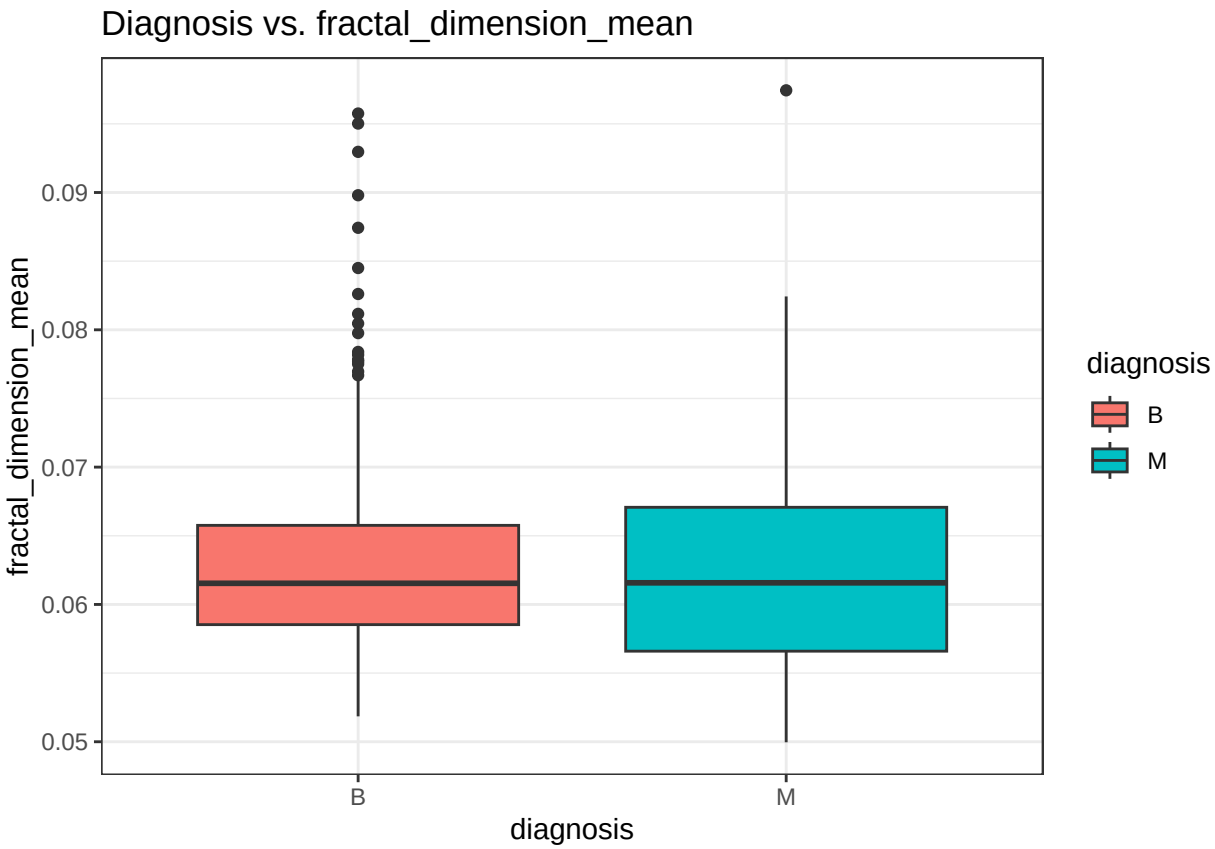

Diagnosis vs. smoothness_mean
smoothness_mean
0.150
0.125
0.100
0.075
0.050
B
M
diagnosis
diagnosis
B
M

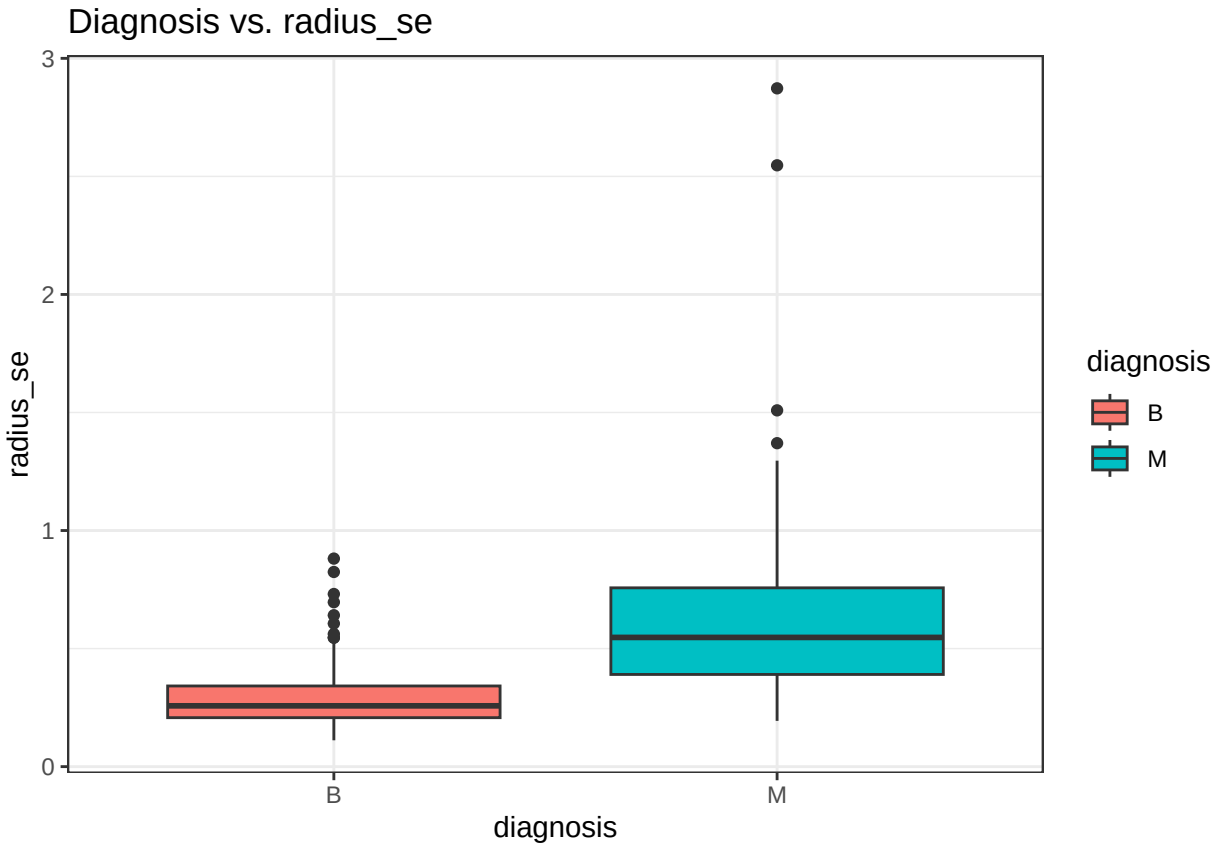

Diagnosis vs. compactness_mean
compactness_mean
0.3
0.2
0.1
B
M
diagnosis
diagnosis
B
M

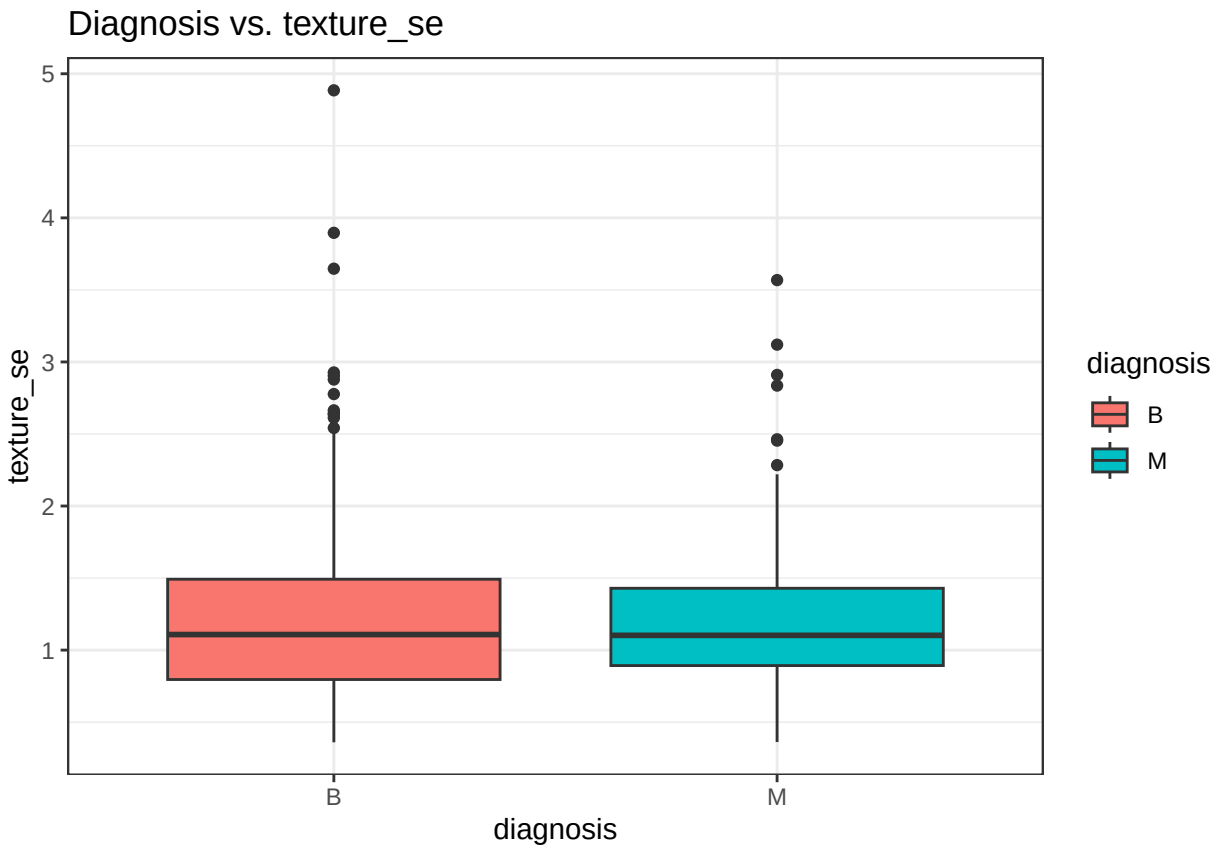

Diagnosis vs. concavity_mean
concavity_mean
0.4
0.3
0.2
0.1
0.0
B
M
diagnosis
diagnosis
B
M

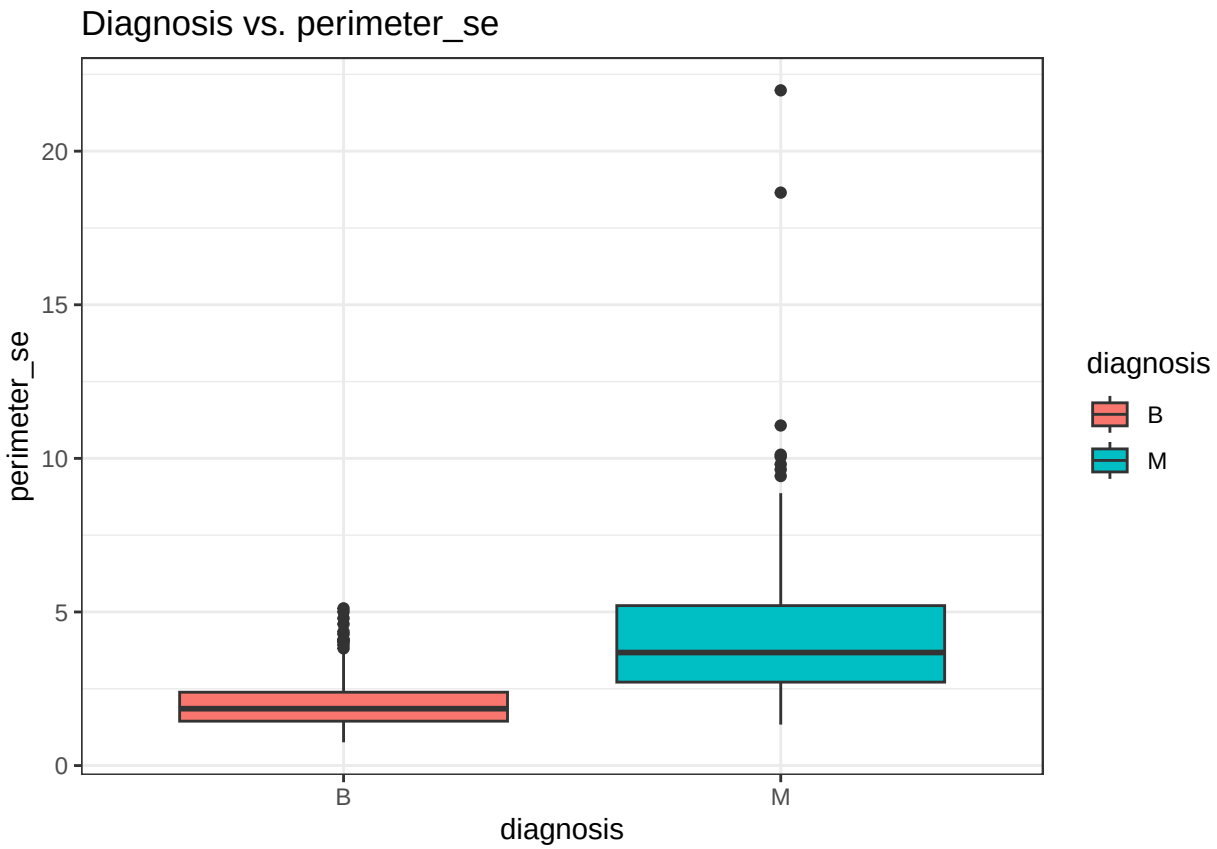

Diagnosis vs. concave_points_mean
concave_points_mean
0.20
0.15
0.10
0.05
0.00
B
M
diagnosis
diagnosis
B
M

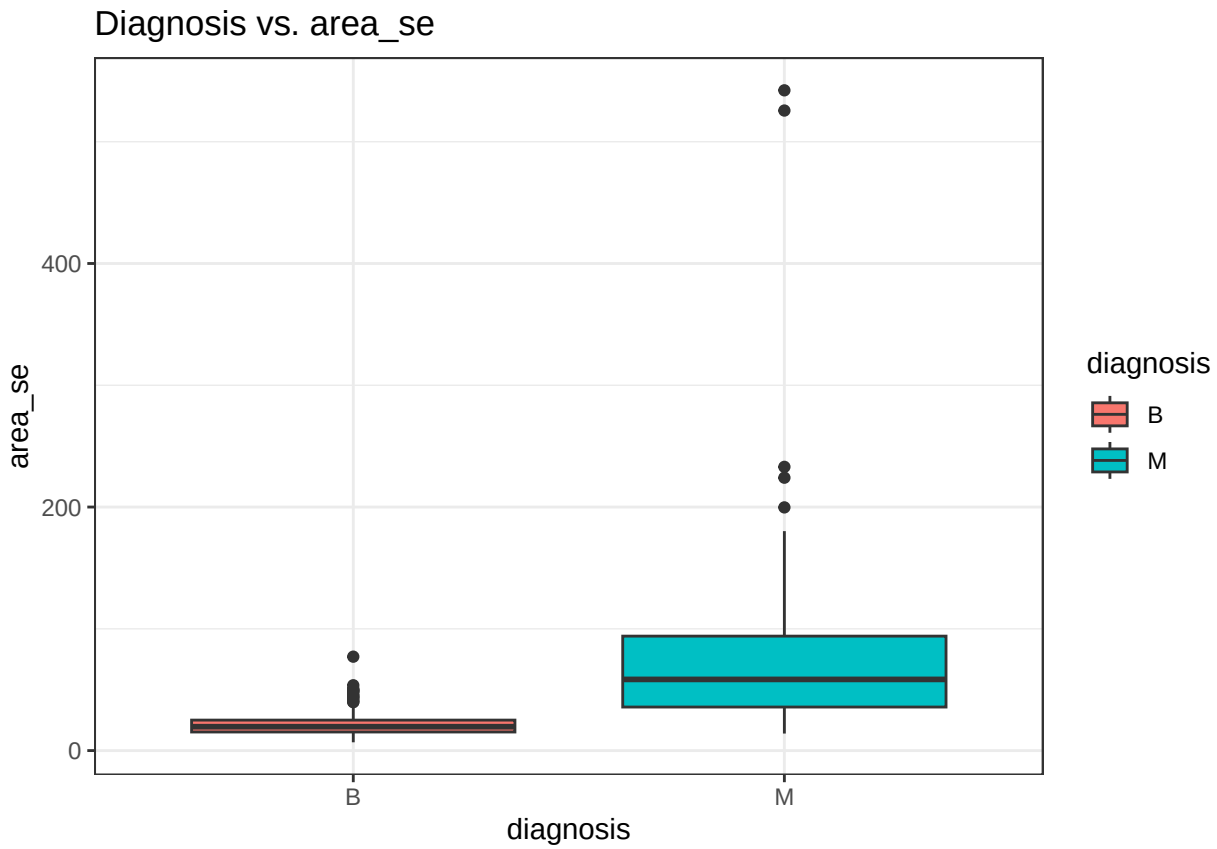

Diagnosis vs. symmetry_mean
0.30
0.25
0.20
0.15
0.10
symmetry_mean
B
M
diagnosis
diagnosis
B
M

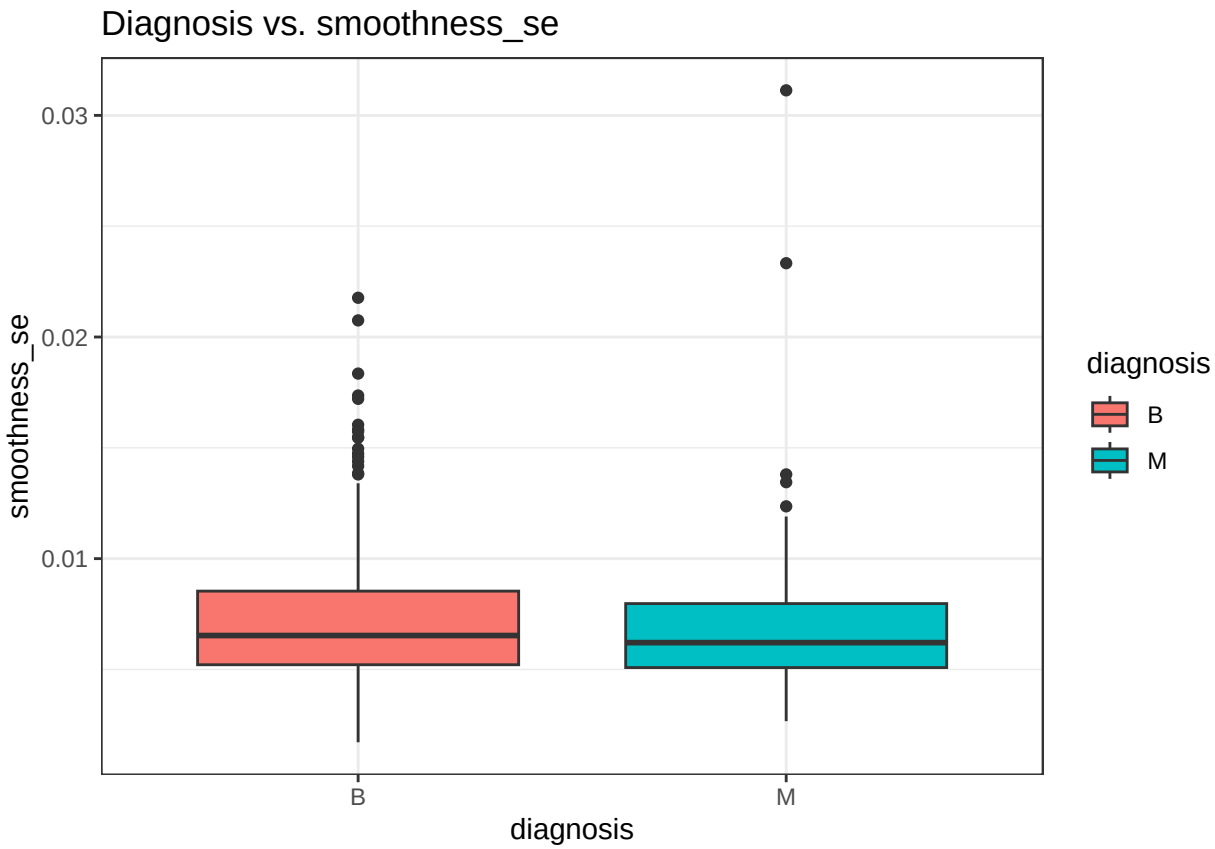

Diagnosis vs. fractal_dimension_mean
fractal_dimension_mean
0.09
0.08
0.07
0.06
0.05
B
M
diagnosis
diagnosis
B
M

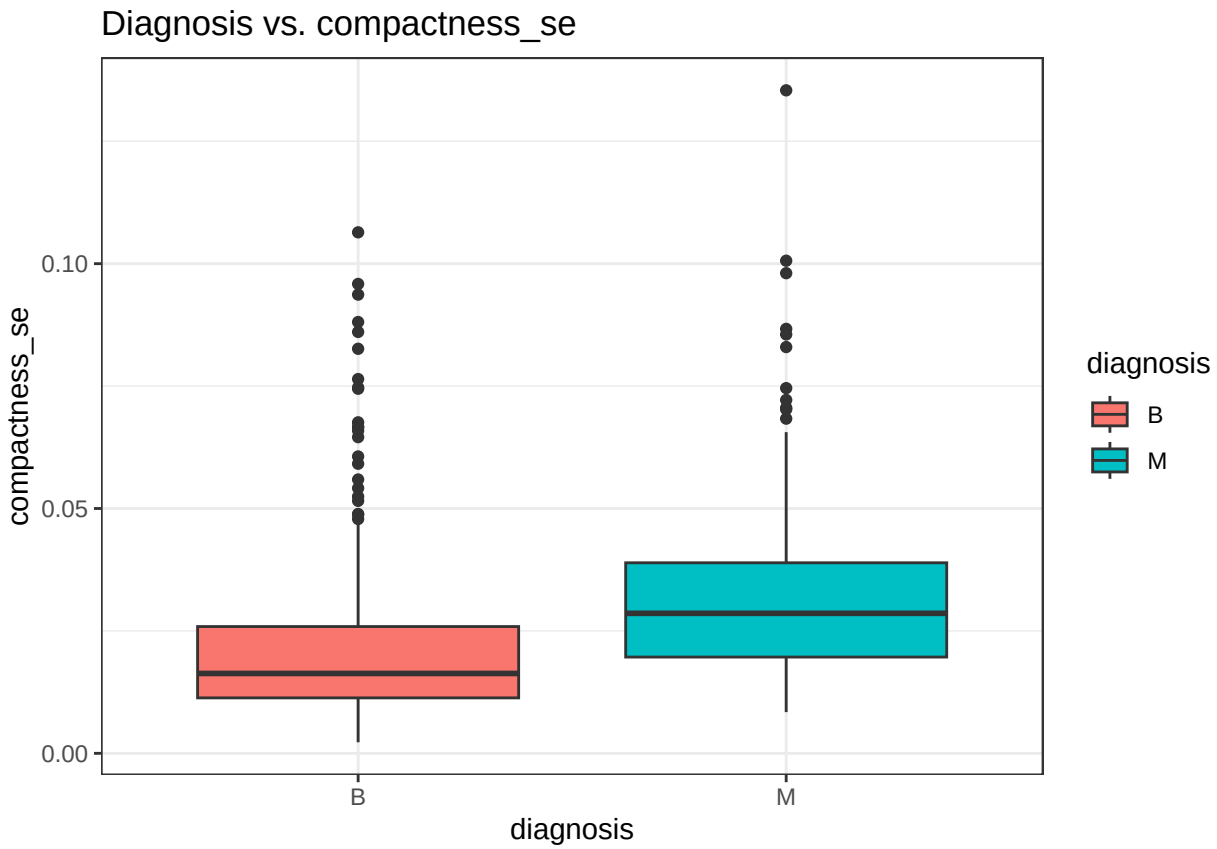

Diagnosis vs. radius_se
radius_se
3
2
1
0
B
M
diagnosis
diagnosis
B
M

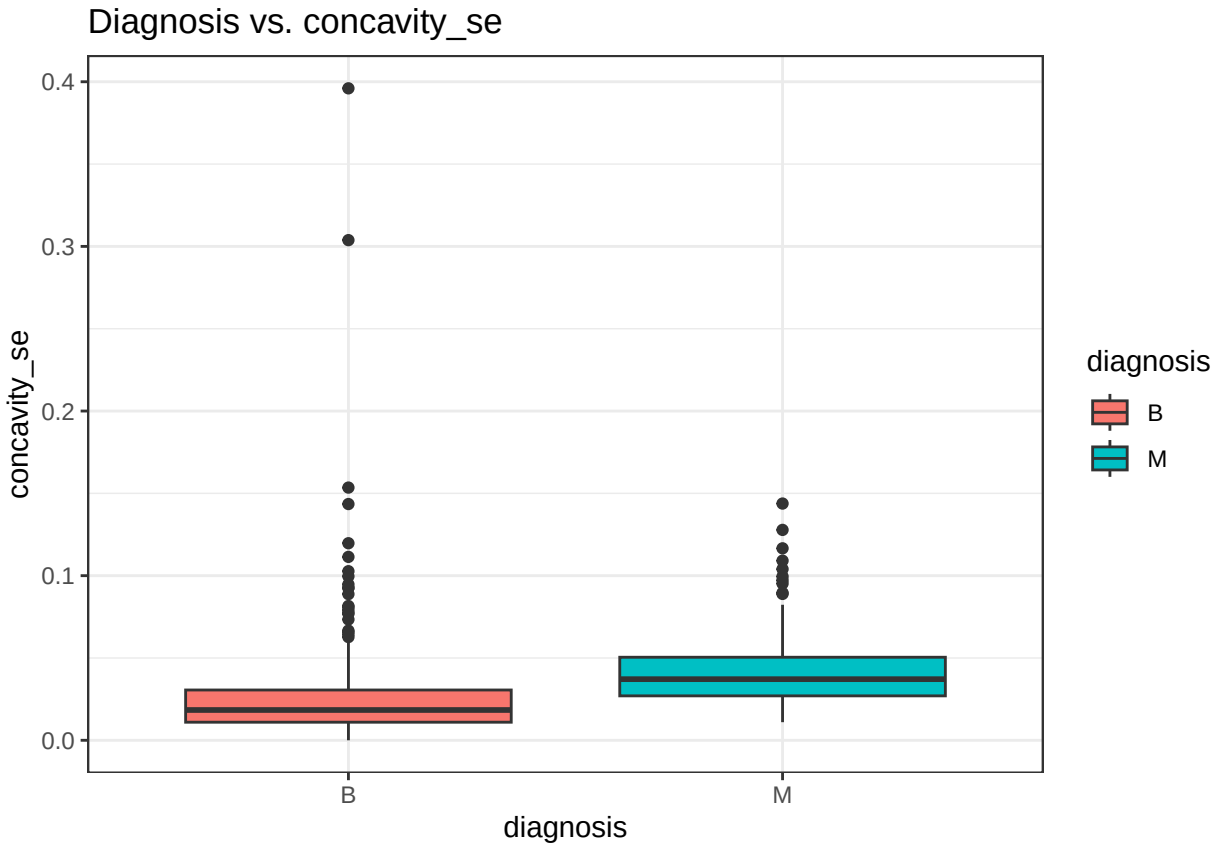

Diagnosis vs. texture_se
texture_se
5
4
3
2
1
B
M
diagnosis
diagnosis
B
M

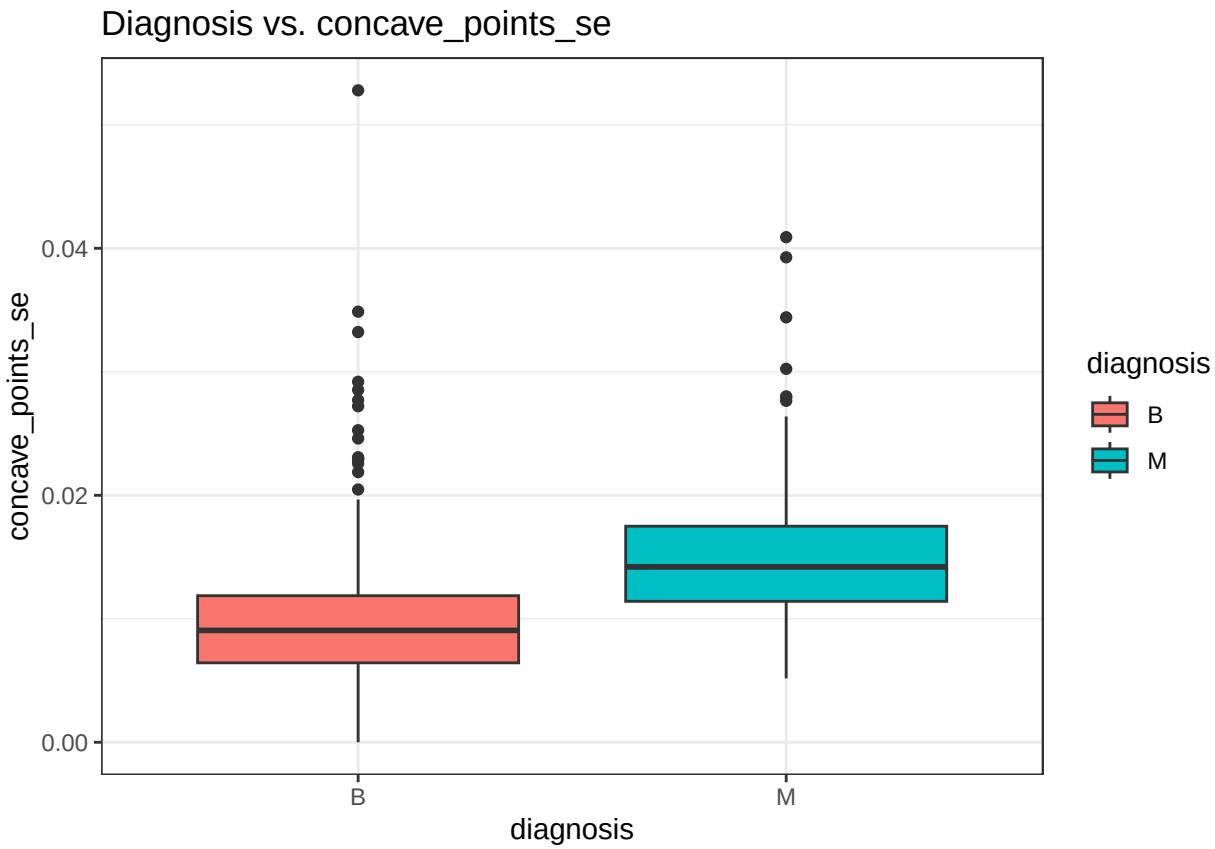

Diagnosis vs. perimeter_se
perimeter_se
20
15
10
5
0
B
M
diagnosis
diagnosis
B
M

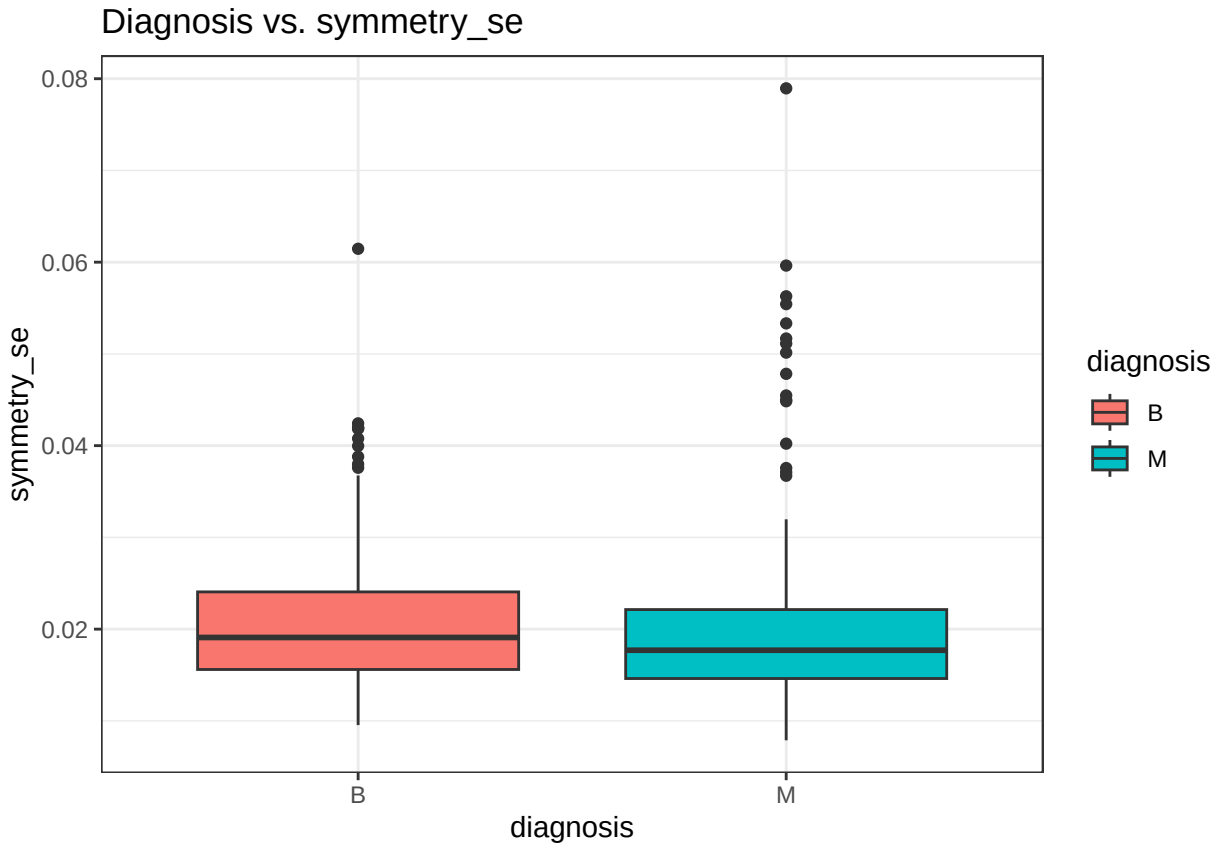

Diagnosis vs. area_se
area_se
400
200
0
B
M
diagnosis
diagnosis
B
M

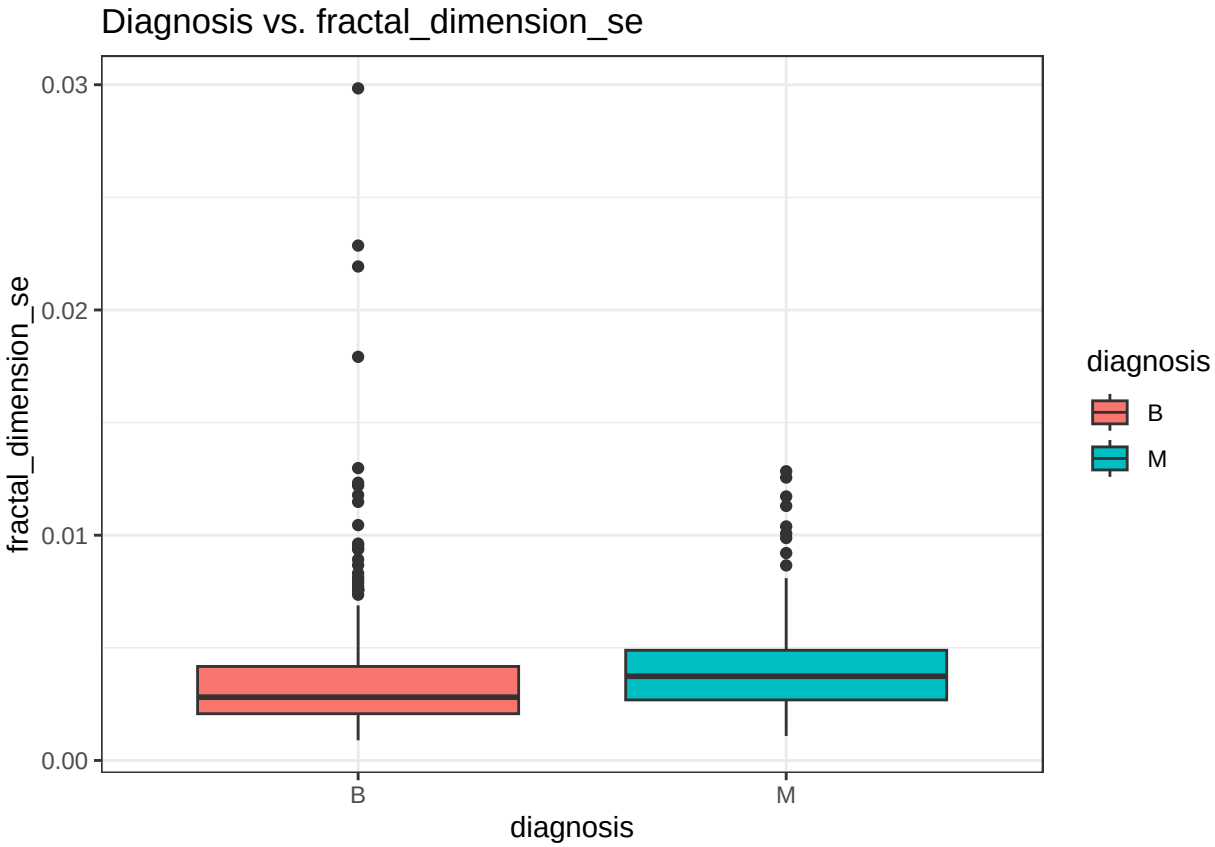

Diagnosis vs. smoothness_se
smoothness_se
0.03
0.02
0.01
B
M
diagnosis
diagnosis
B
M

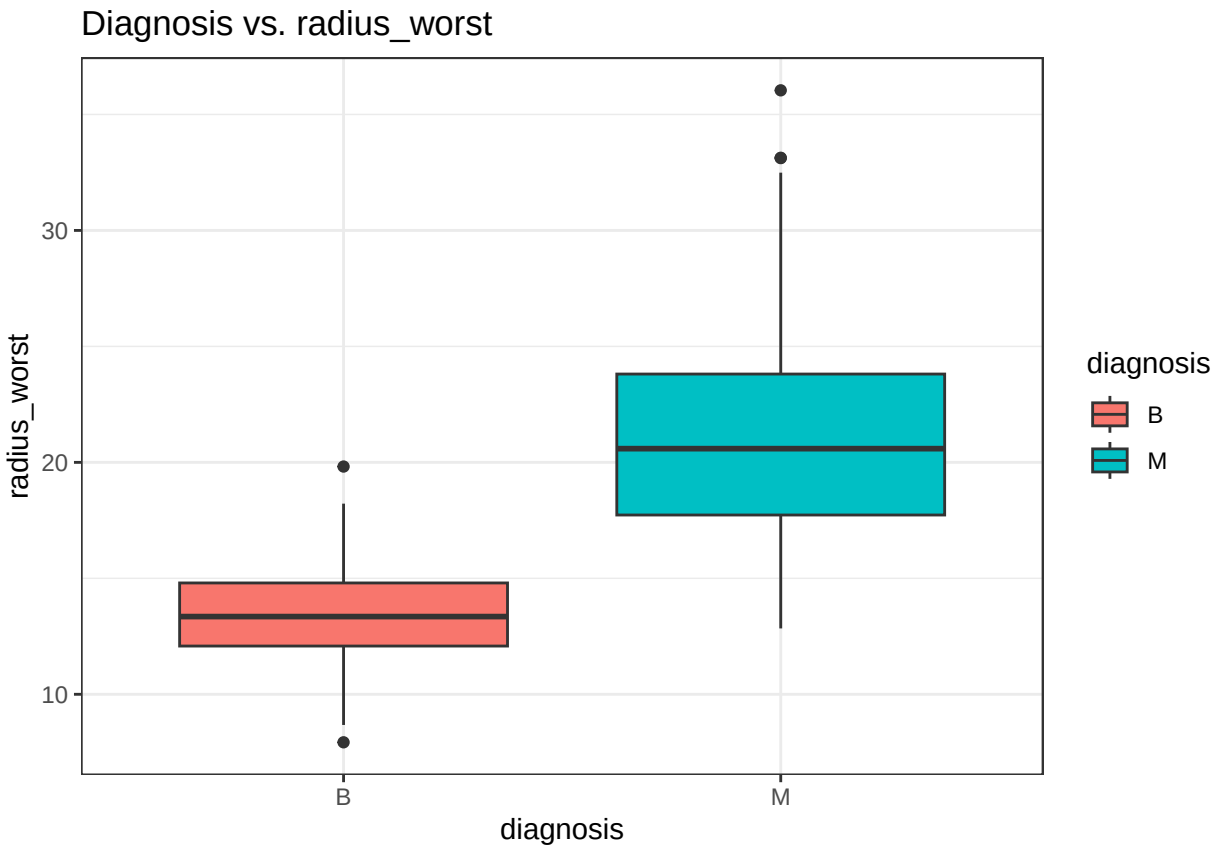

Diagnosis vs. compactness_se
compactness_se
0.10
0.05
0.00
B
M
diagnosis
diagnosis
B
M

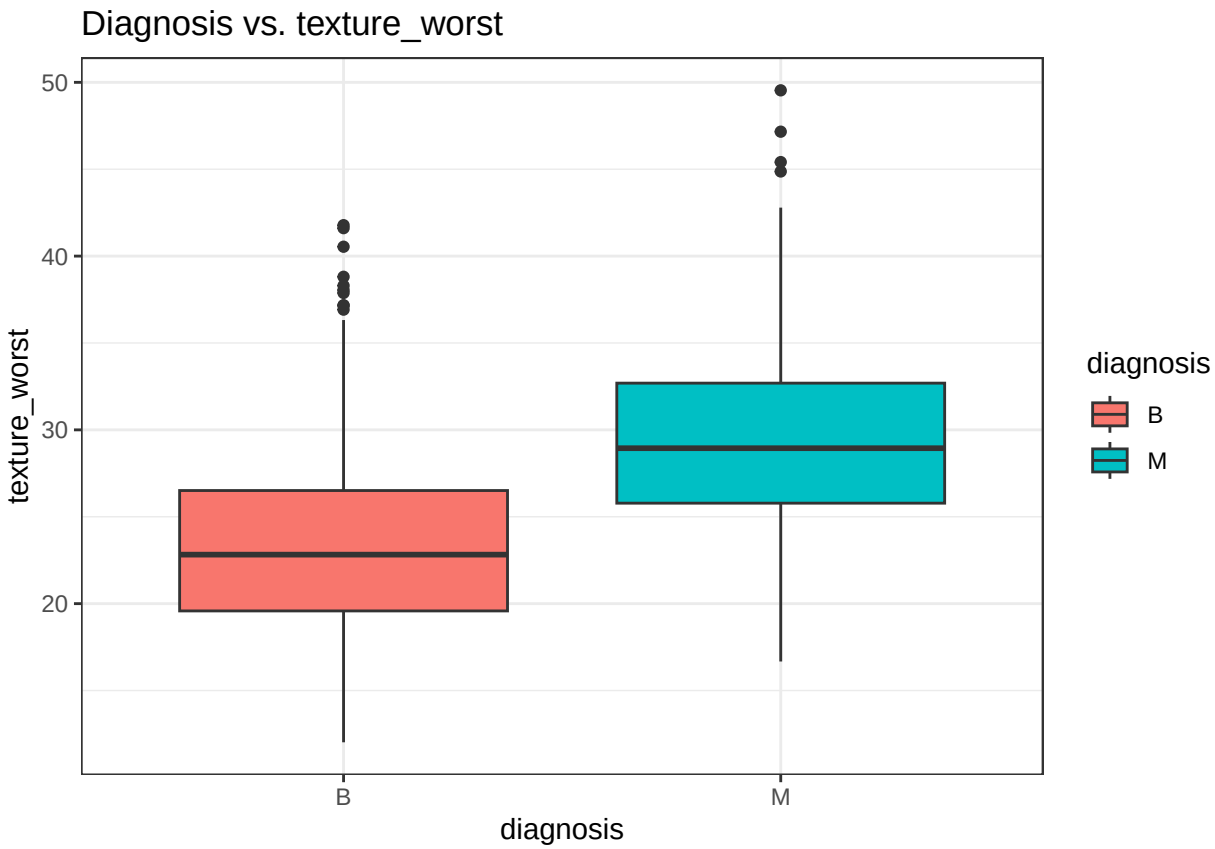

Diagnosis vs. concavity_se
concavity_se
0.4
0.3
0.2
0.1
0.0
B
M
diagnosis
diagnosis
B
M

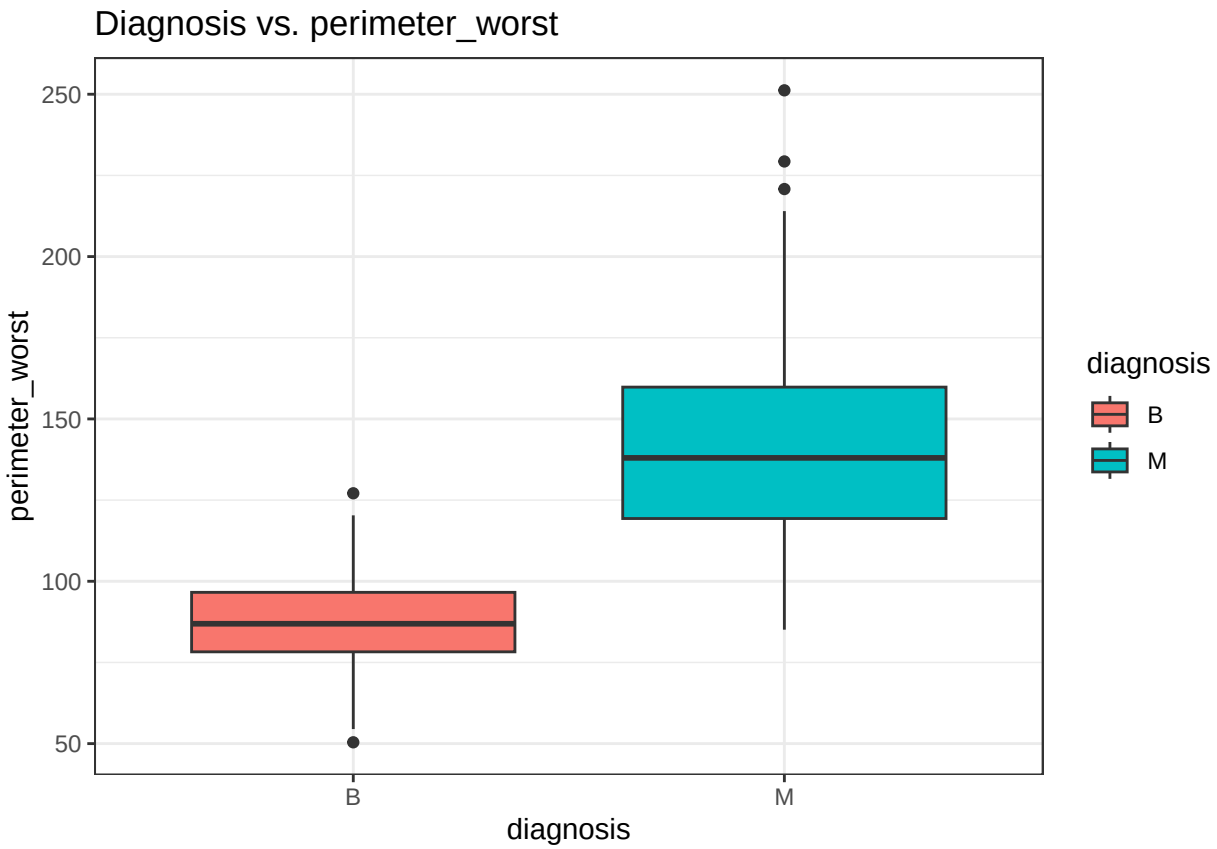

Diagnosis vs. concave_points_se
concave_points_se
0.04
0.02
0.00
B
M
diagnosis
diagnosis
B
M

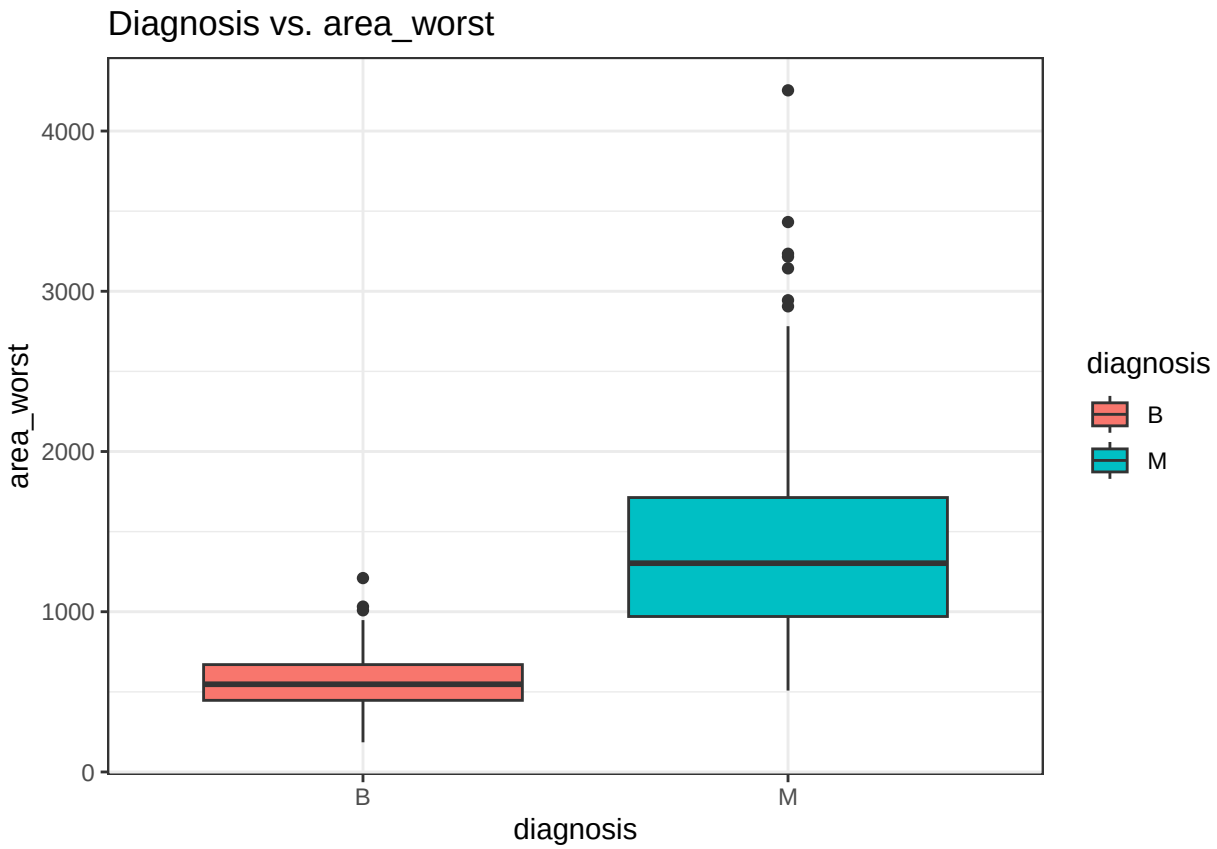

Diagnosis vs. symmetry_se
0.08
0.06
0.04
0.02
symmetry_se
B
M
diagnosis
diagnosis
B
M

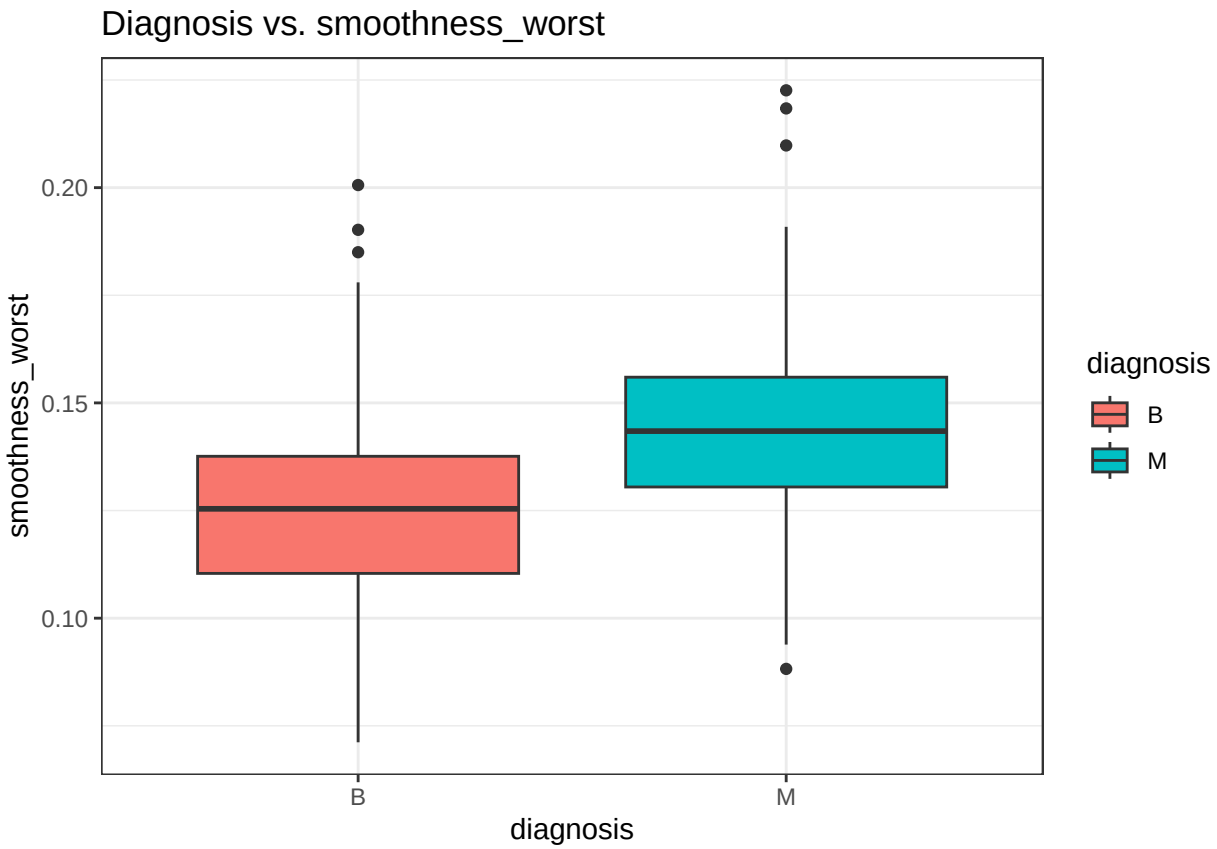

Diagnosis vs. fractal_dimension_se
fractal_dimension_se
0.03
0.02
0.01
0.00
B
M
diagnosis
diagnosis
B
M

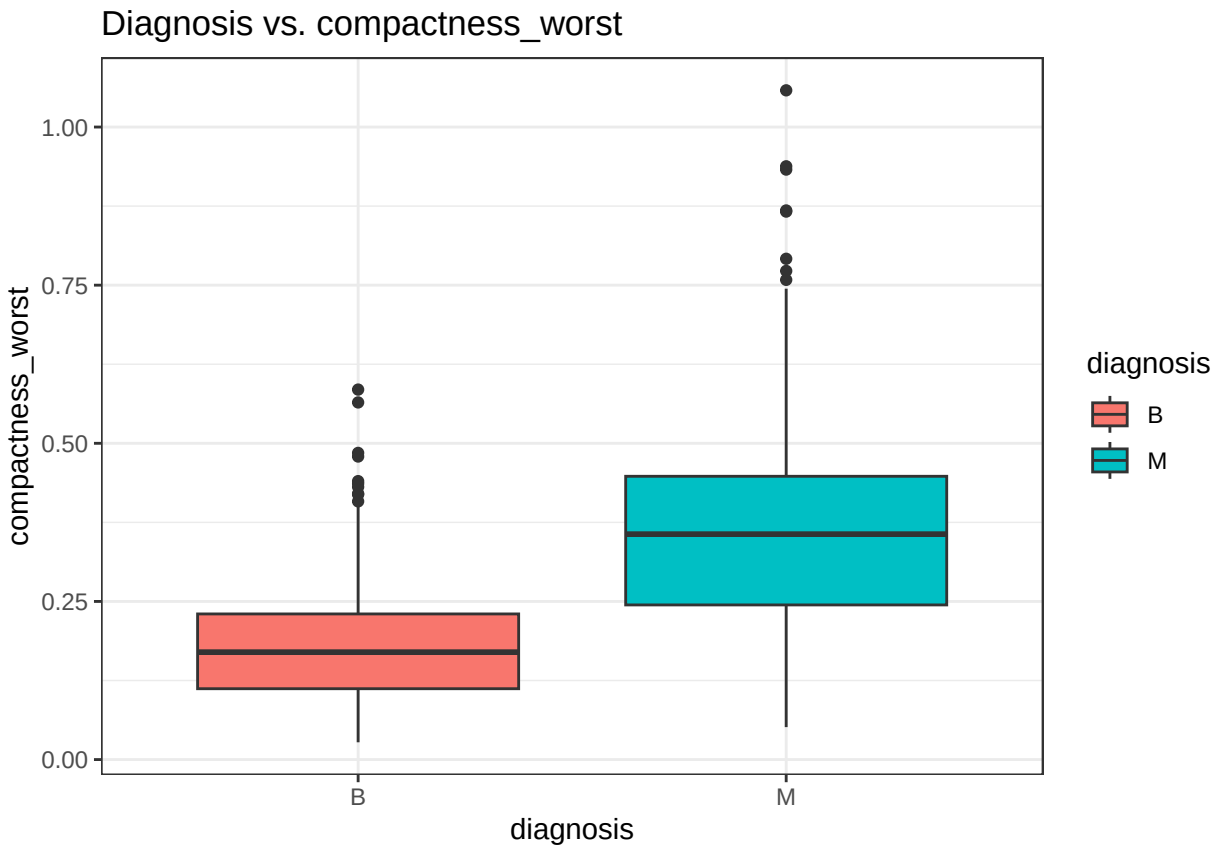

Diagnosis vs. radius_worst
radius_worst
30
20
10
B
M
diagnosis
diagnosis
B
M

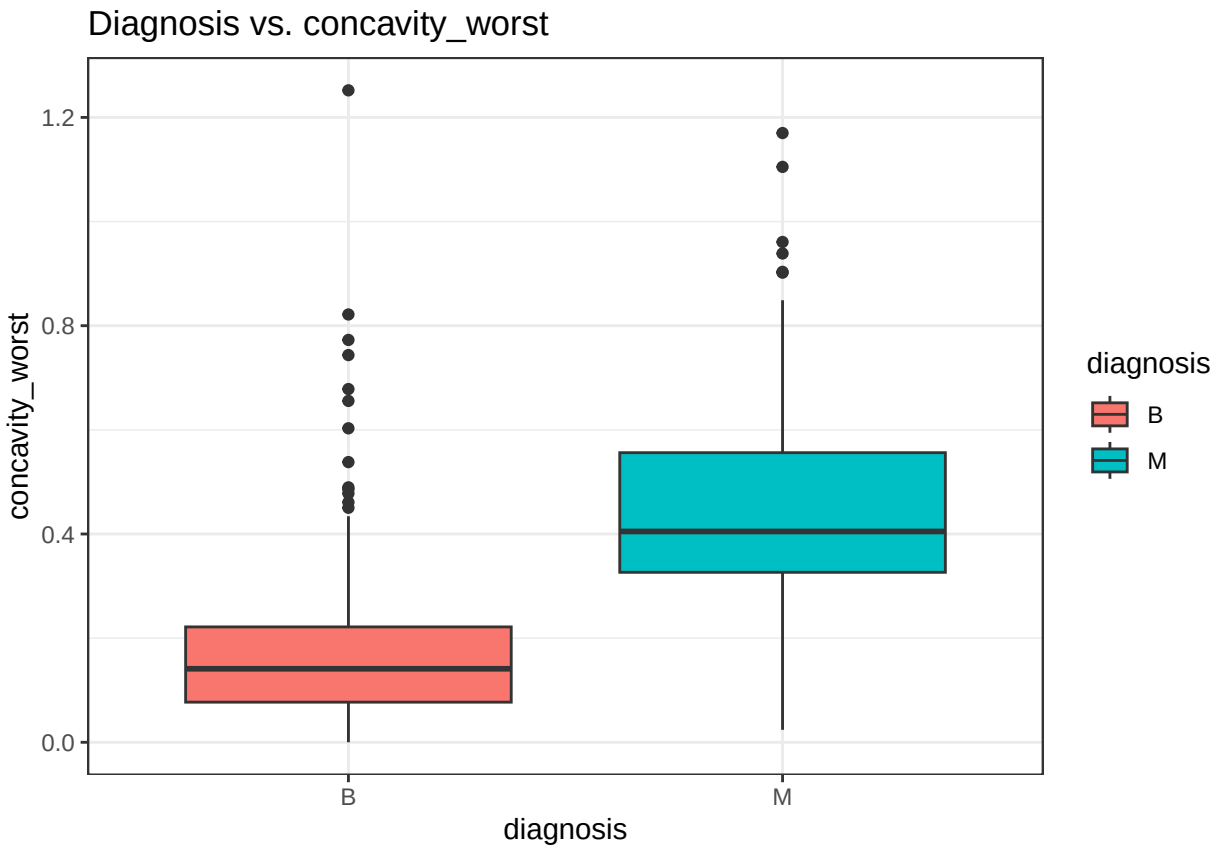

Diagnosis vs. texture_worst
texture_worst
50
40
30
20
B
M
diagnosis
diagnosis
B
M

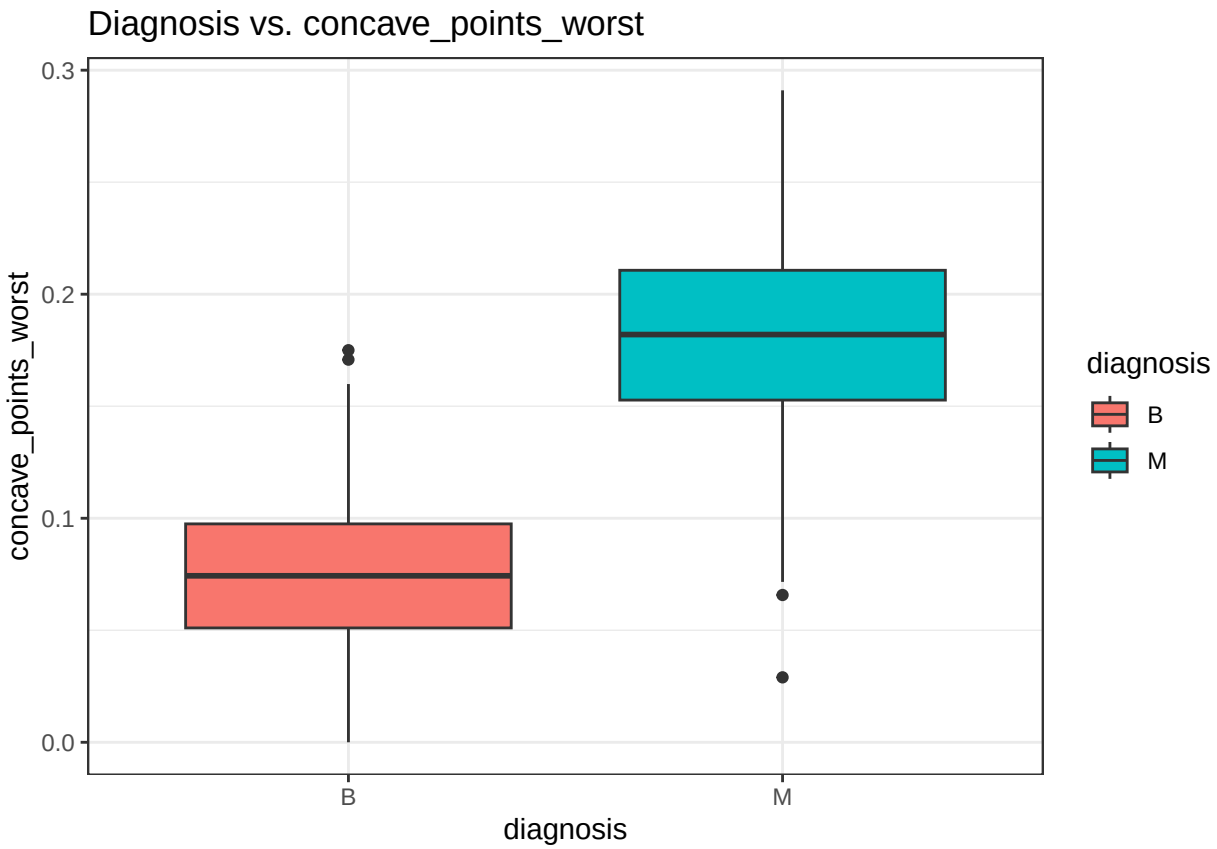

Diagnosis vs. perimeter_worst
perimeter_worst
250
200
150
100
50
B
M
diagnosis
diagnosis
B
M

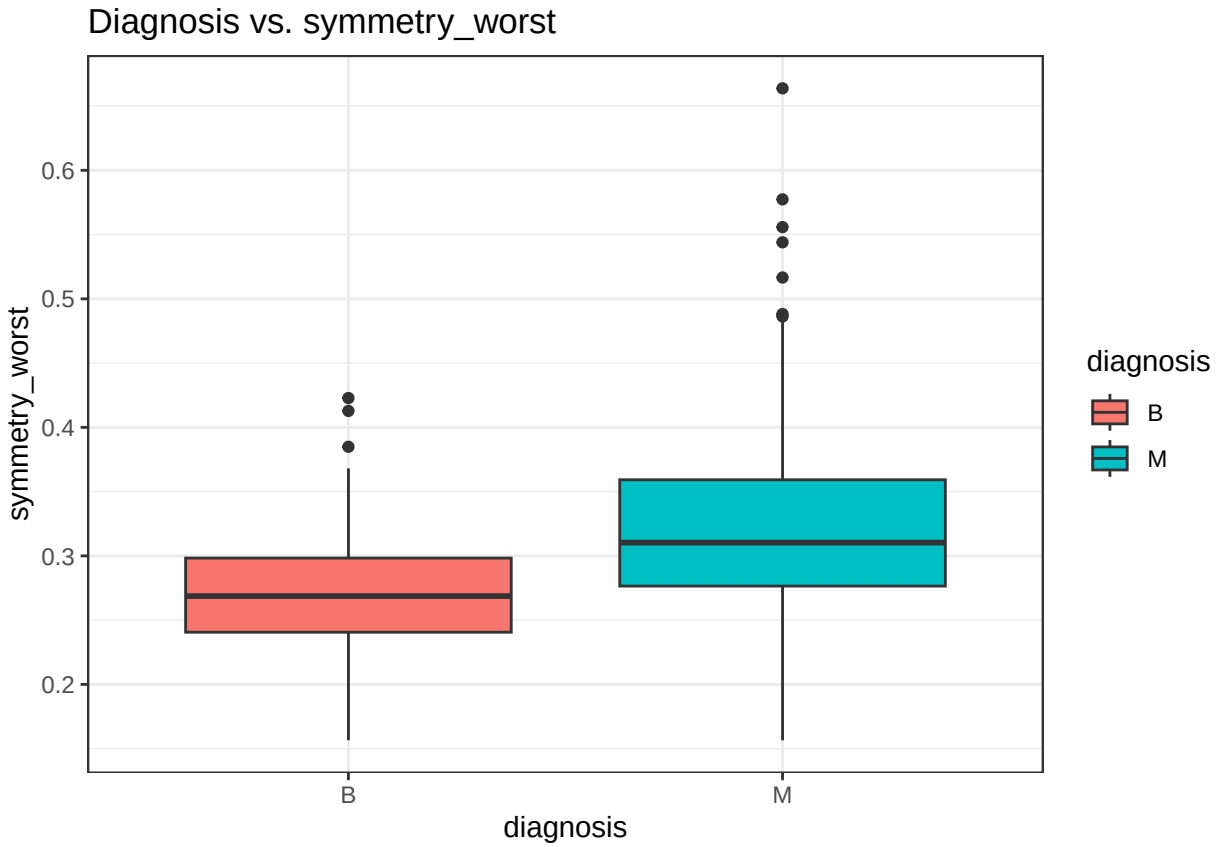

Diagnosis vs. area_worst
area_worst
4000
3000
2000
1000
0
B
M
diagnosis
diagnosis
B
M

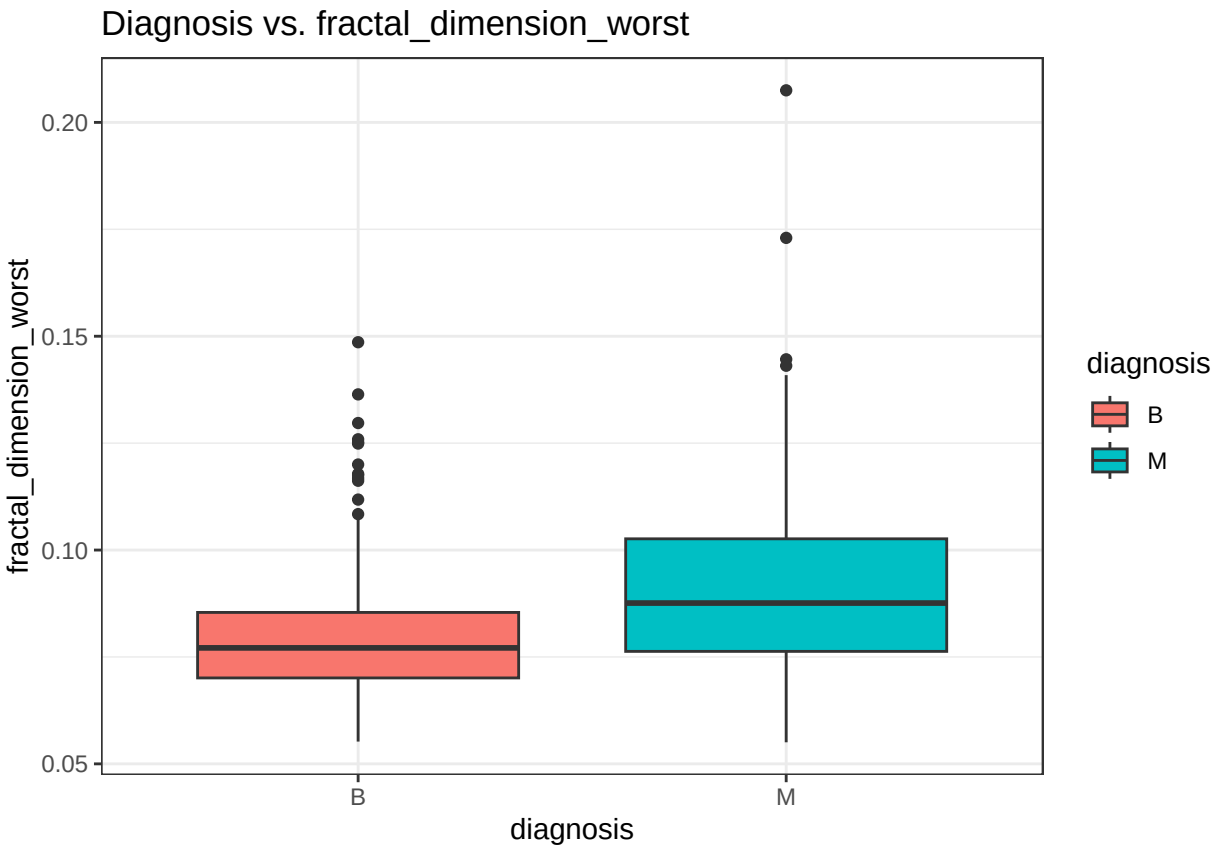

Diagnosis vs. smoothness_worst
smoothness_worst
0.20
0.15
0.10
B
M
diagnosis
diagnosis
B
M


Diagnosis vs. compactness_worst
compactness_worst
1.00
0.75
0.50
0.25
0.00
B
M
diagnosis
diagnosis
B
M


Diagnosis vs. concavity_worst
concavity_worst
1.2
0.8
0.4
0.0
B
M
diagnosis
diagnosis
B
M


Diagnosis vs. concave_points_worst
concave_points_worst
0.3
0.2
0.1
0.0
B
M
diagnosis
diagnosis
B
M

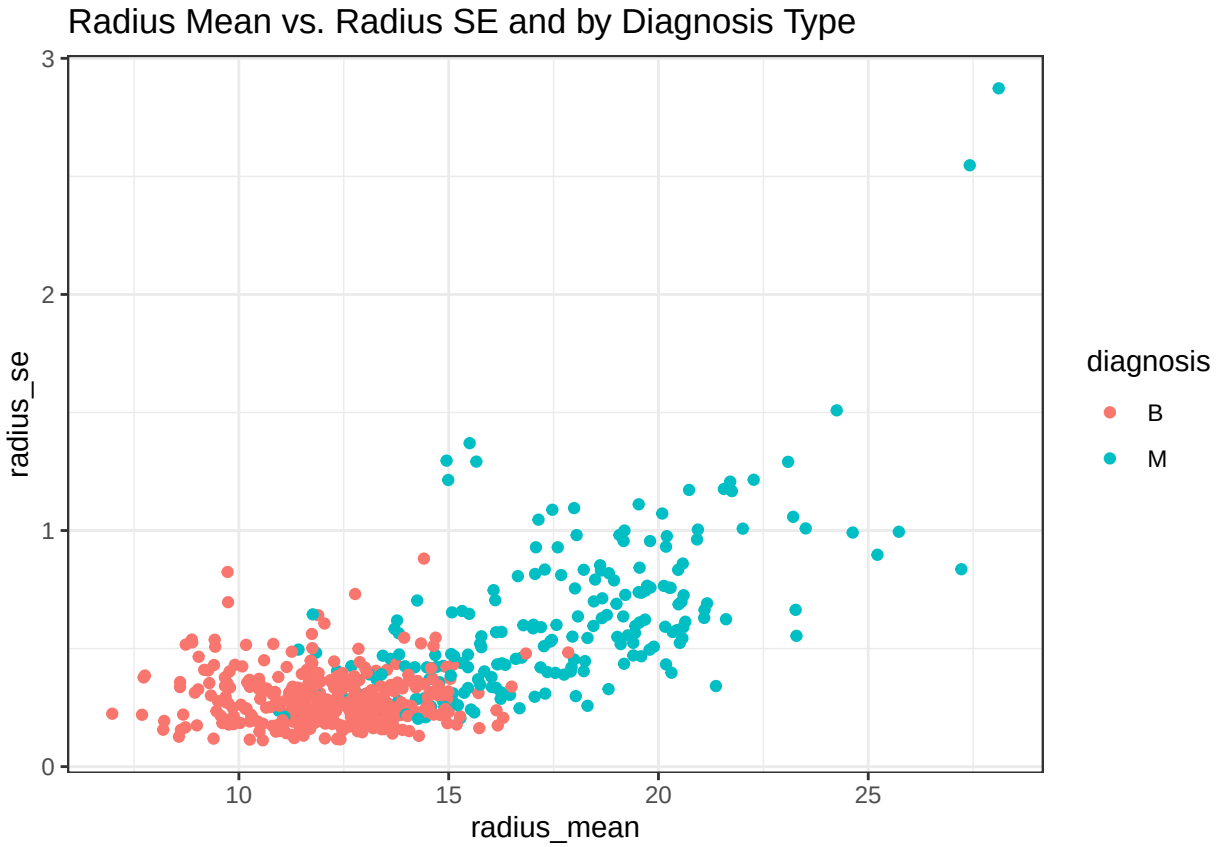

Diagnosis vs. symmetry_worst
symmetry_worst
0.6
0.5
0.4
0.3
0.2
B
M
diagnosis
diagnosis
B
M



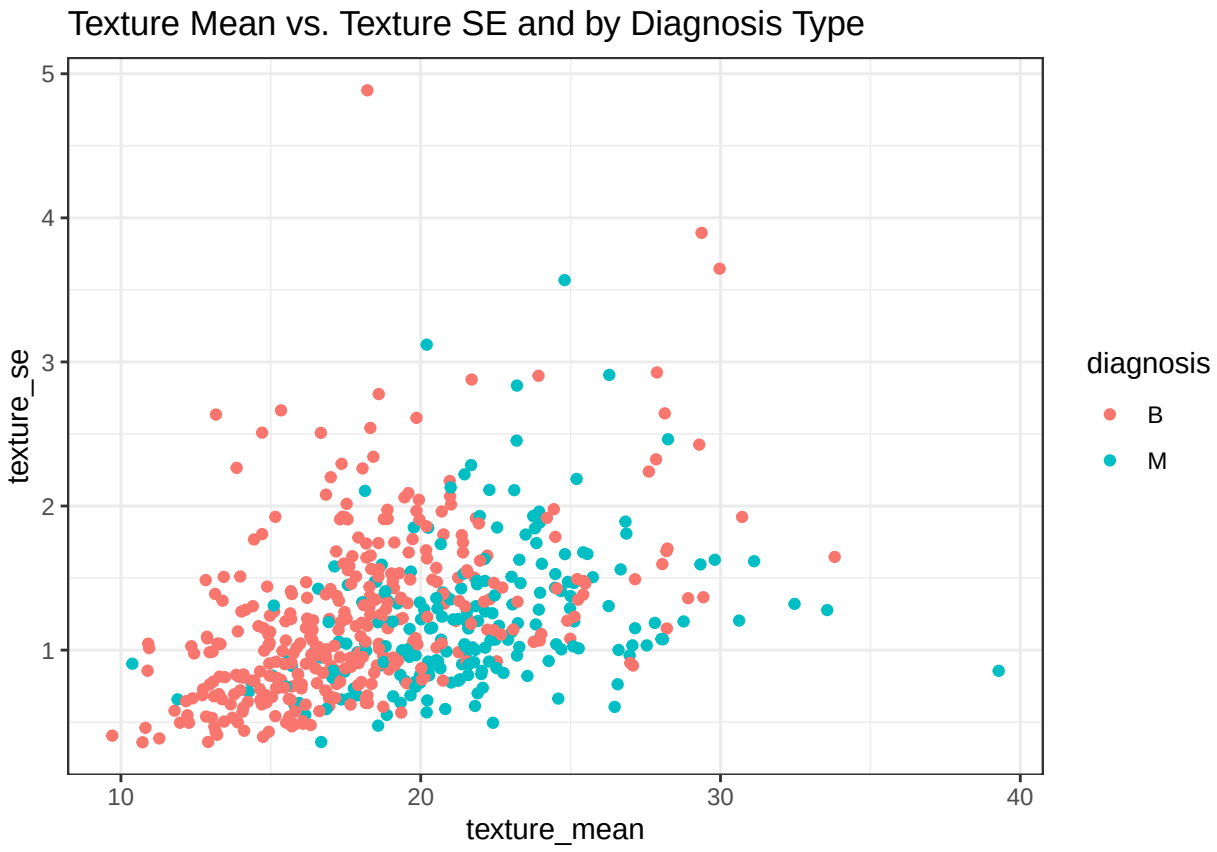
Conclusion: In general, higher values for any of the potential explanatory variables are found in malignant tissue masses compared to those classified as benign.

Relationship between mean and se of each explanatory variable

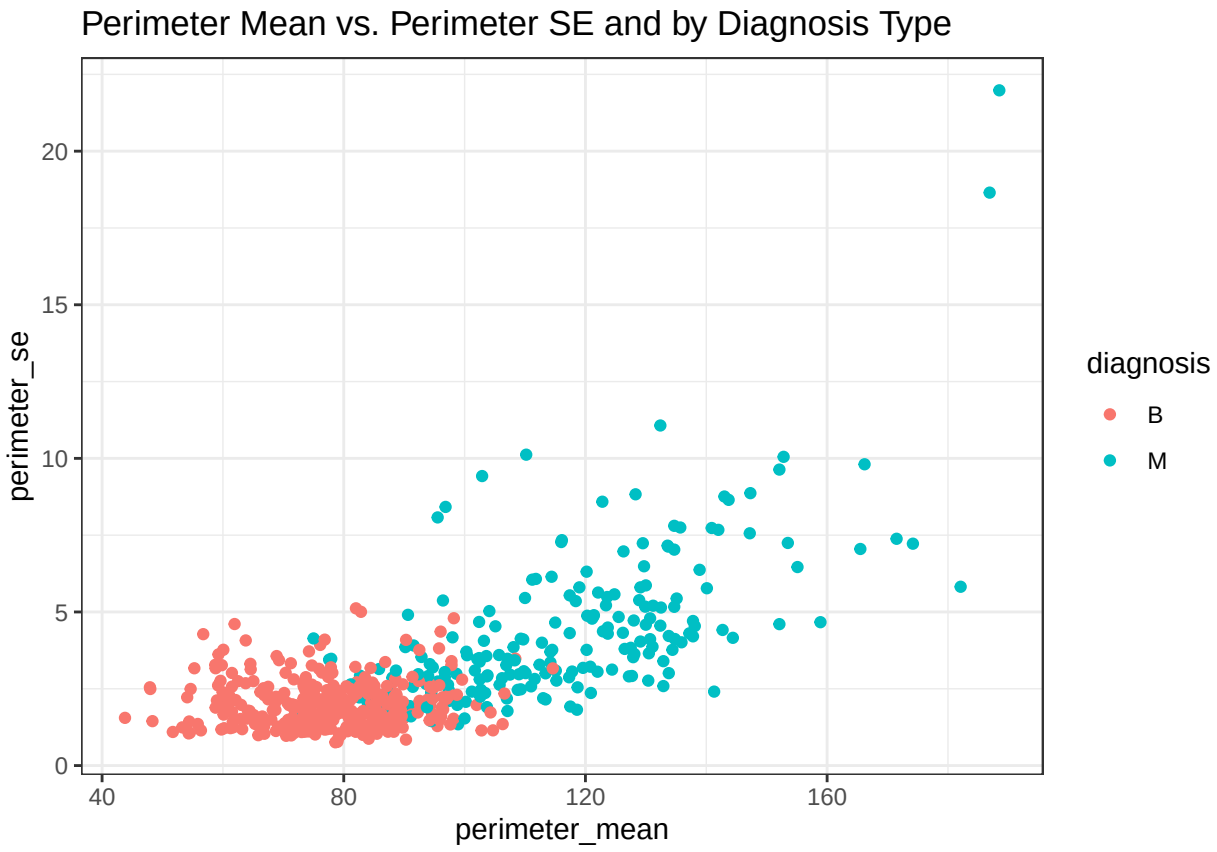
```
cancer_data_clean %>%
  ggplot(aes(x = radius_mean, y = radius_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Radius Mean vs. Radius SE and by Diagnosis Type") +
  theme_bw()
```



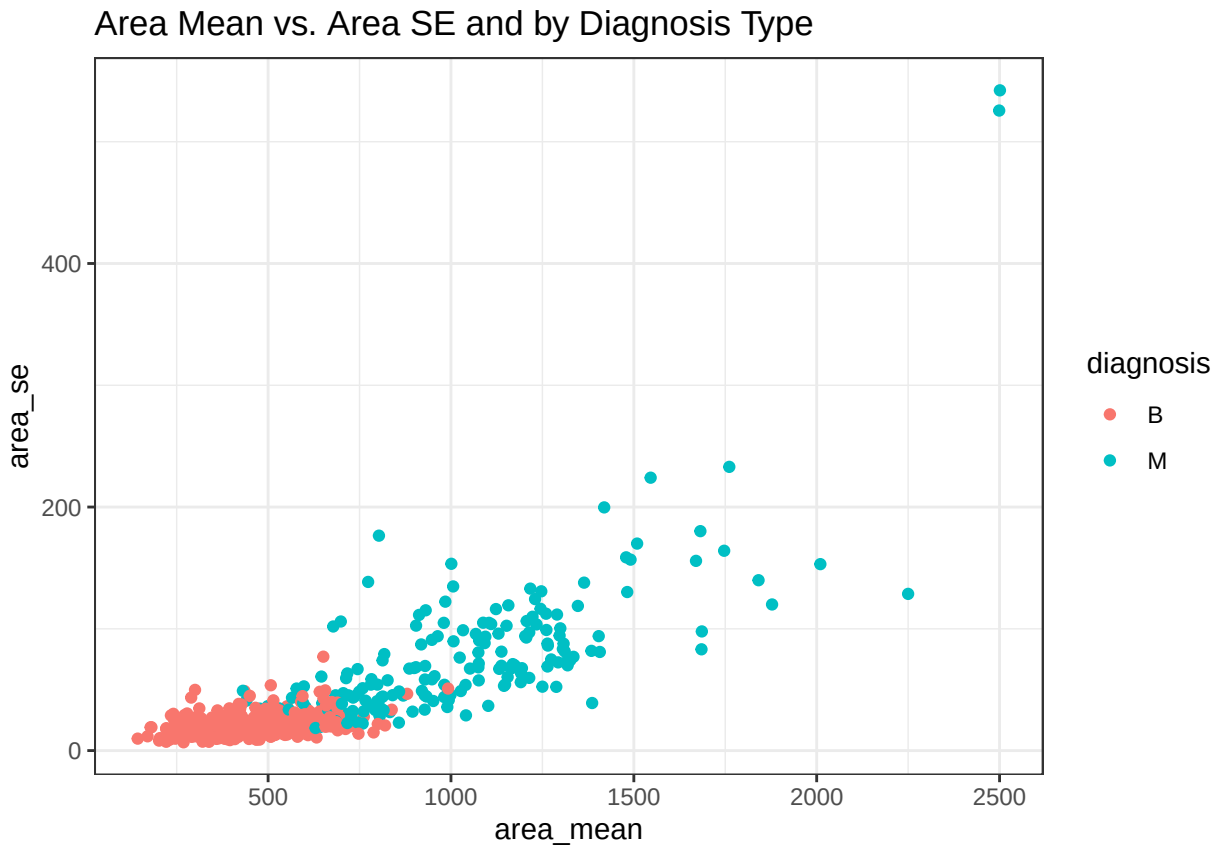

```
cancer_data_clean %>%
  ggplot(aes(x = texture_mean, y = texture_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Texture Mean vs. Texture SE and by Diagnosis Type") +
  theme_bw()
```



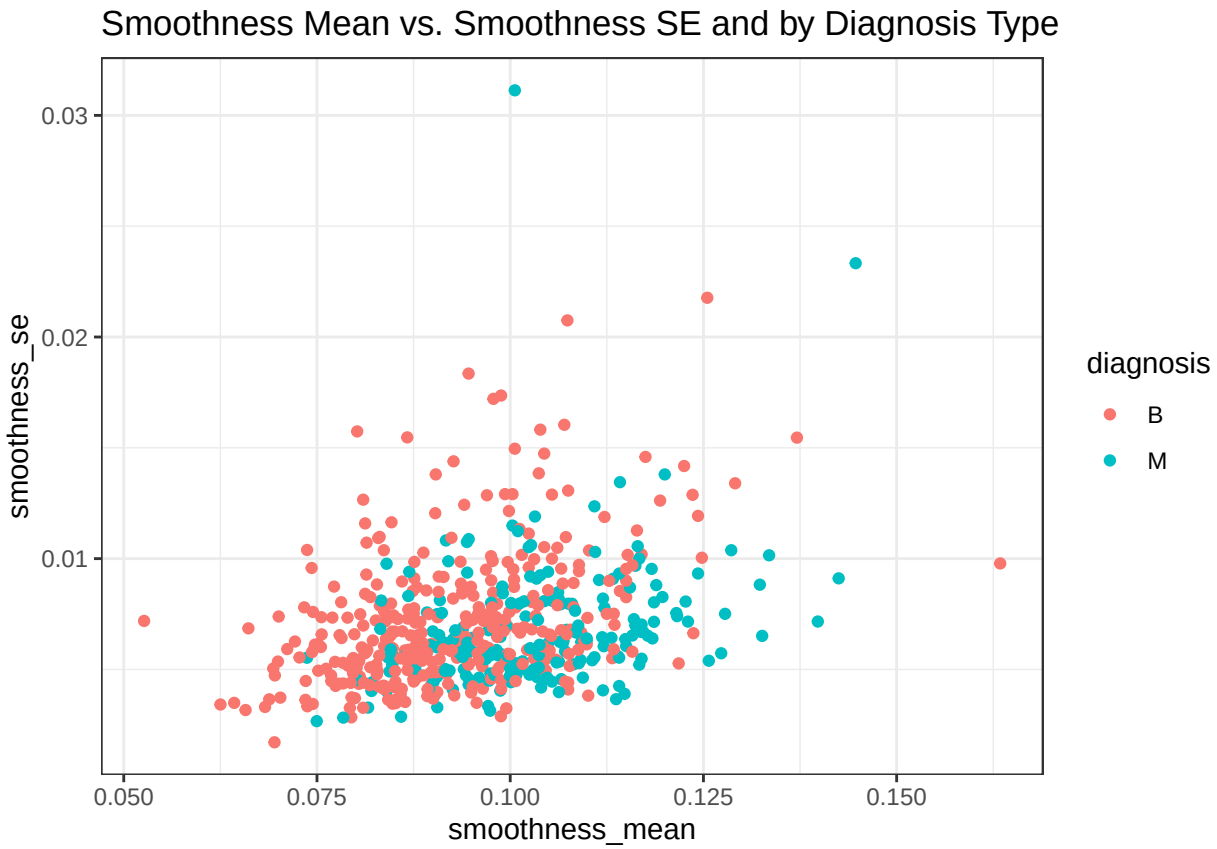

```
cancer_data_clean %>%
  ggplot(aes(x = perimeter_mean, y = perimeter_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Perimeter Mean vs. Perimeter SE and by Diagnosis Type") +
  theme_bw()
```



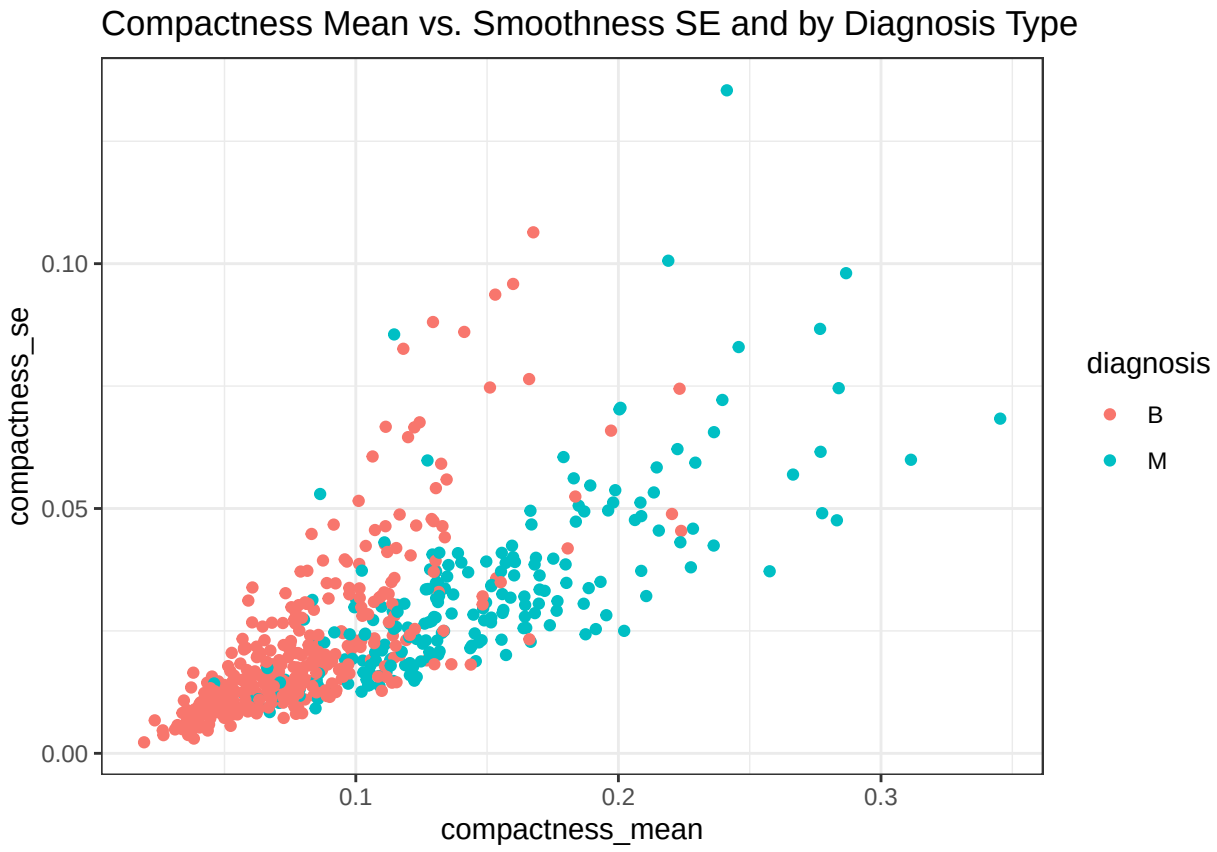

```
cancer_data_clean %>%
  ggplot(aes(x = area_mean, y = area_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Area Mean vs. Area SE and by Diagnosis Type") +
  theme_bw()
```



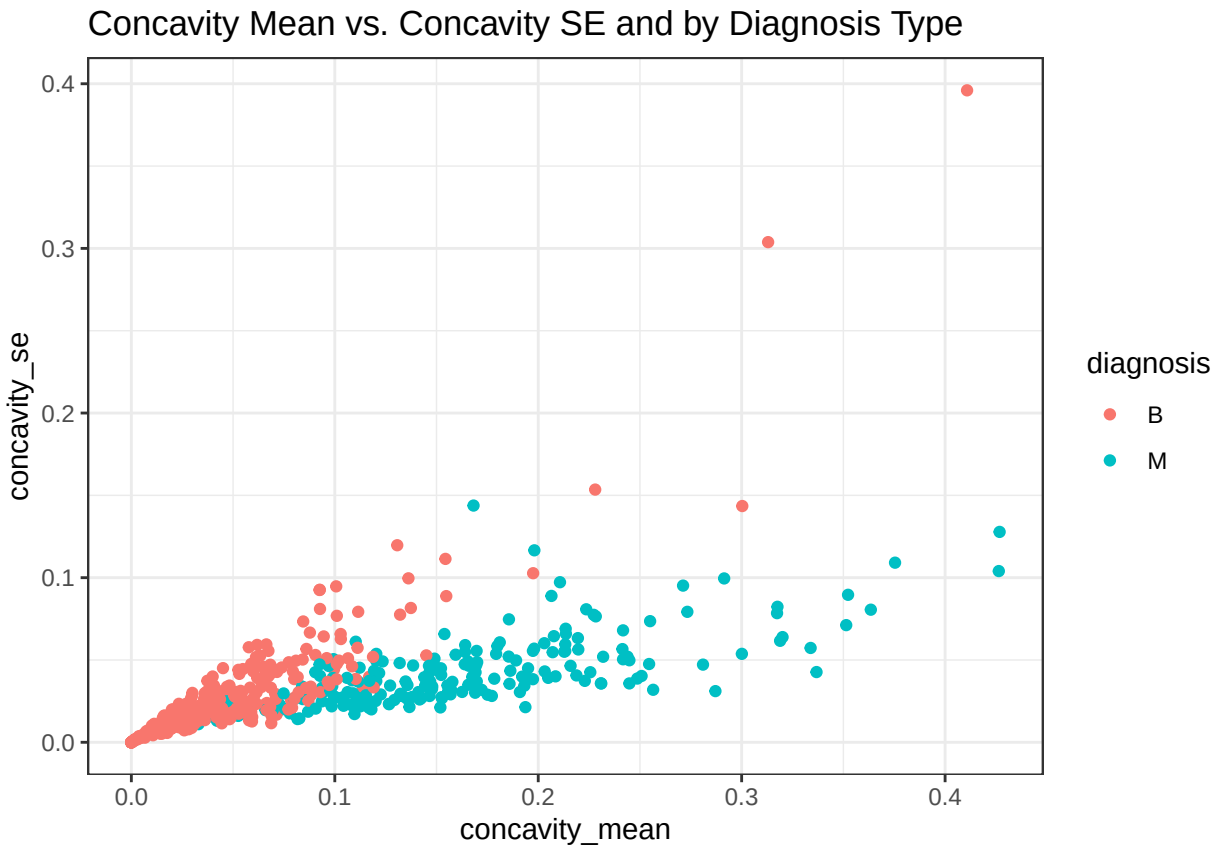

```
cancer_data_clean %>%
  ggplot(aes(x = smoothness_mean, y = smoothness_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Smoothness Mean vs. Smoothness SE and by Diagnosis Type") +
  theme_bw()
```



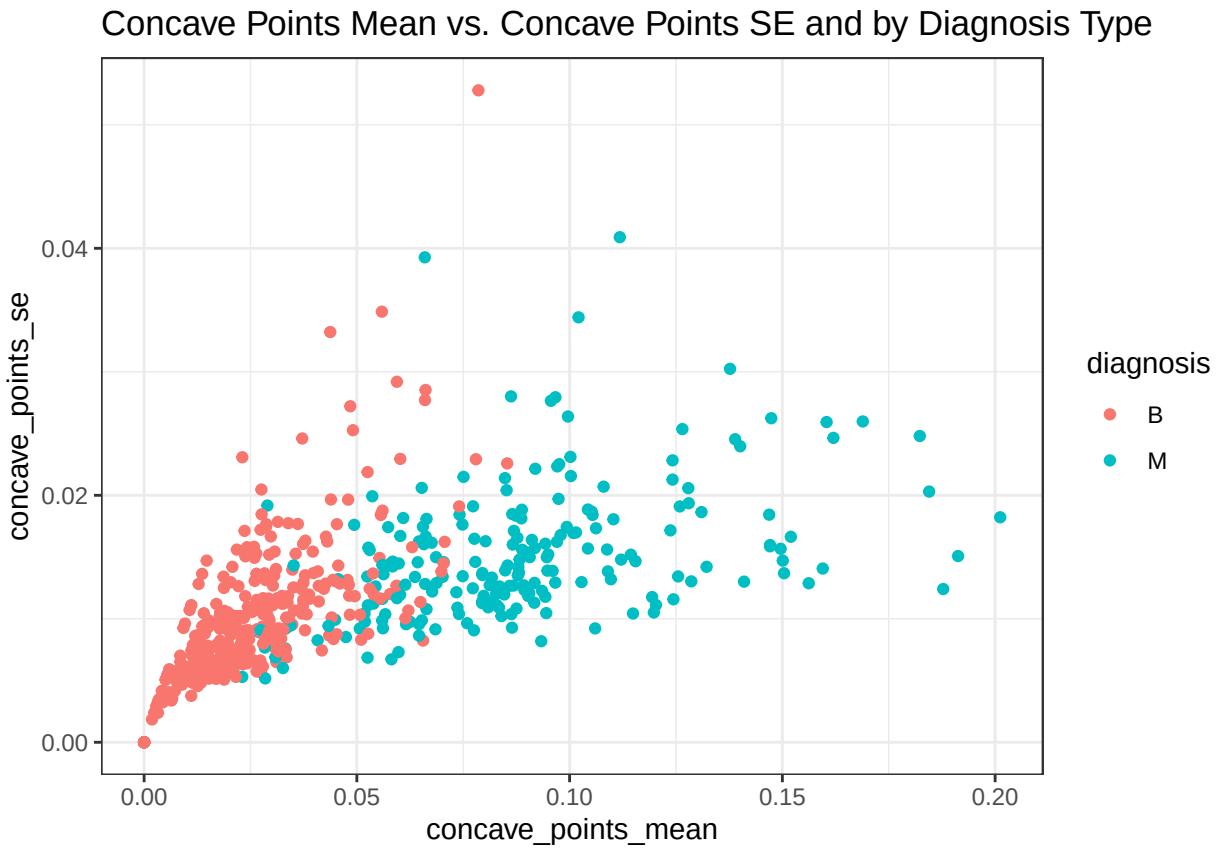

```
cancer_data_clean %>%
  ggplot(aes(x = compactness_mean, y = compactness_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Compactness Mean vs. Smoothness SE and by Diagnosis Type") +
  theme_bw()
```



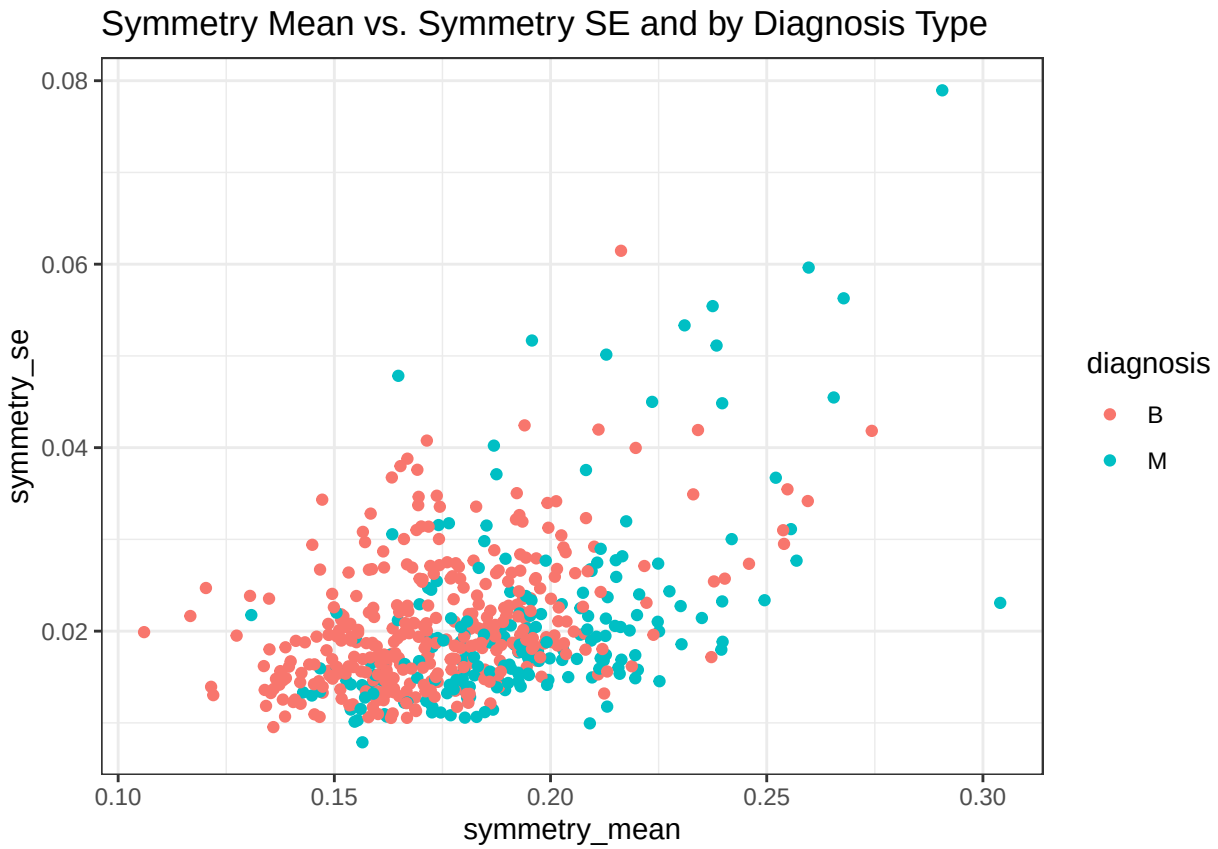

```
cancer_data_clean %>%
  ggplot(aes(x = concavity_mean, y = concavity_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Concavity Mean vs. Concavity SE and by Diagnosis Type") +
  theme_bw()
```



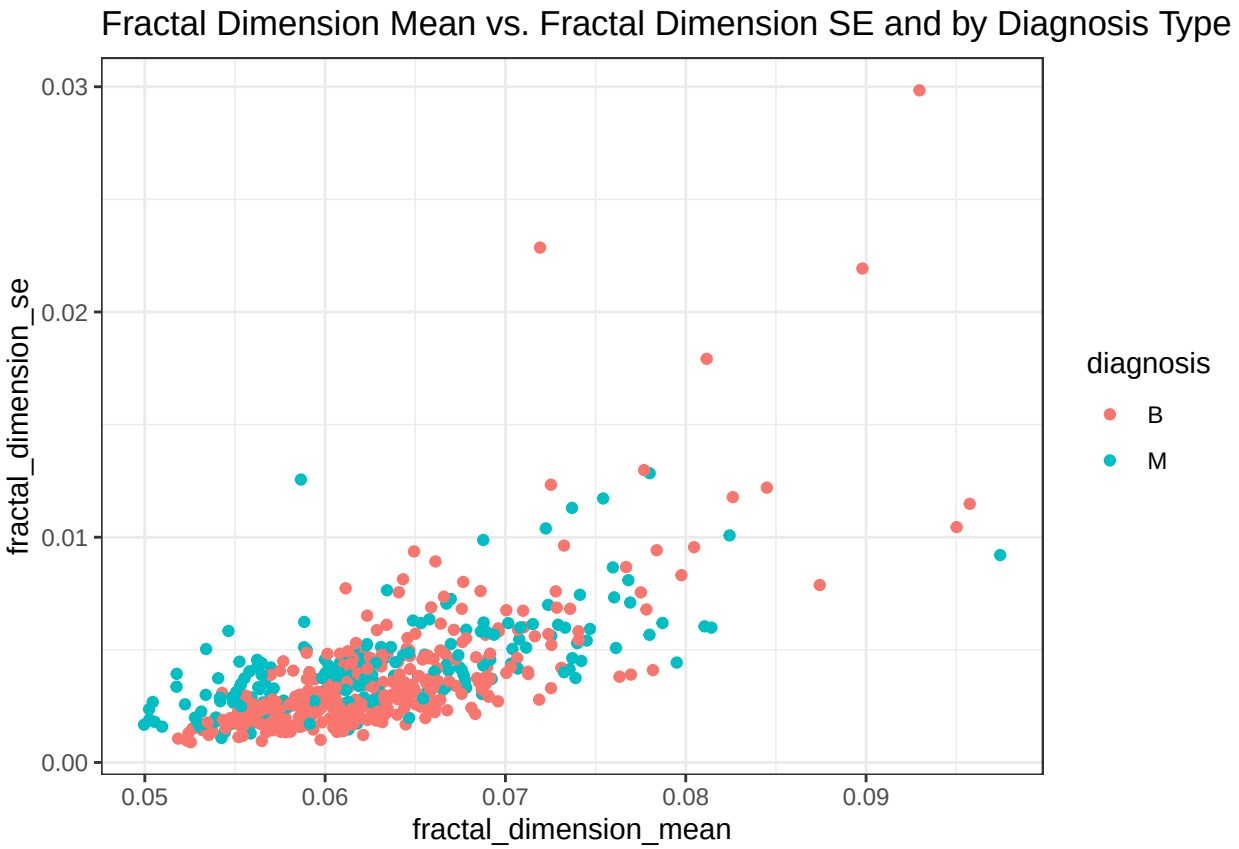

```
cancer_data_clean %>%
  ggplot(aes(x = concave_points_mean, y = concave_points_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Concave Points Mean vs. Concave Points SE and by Diagnosis Type") +
  theme_bw()
```




```
cancer_data_clean %>%
  ggplot(aes(x = symmetry_mean, y = symmetry_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Symmetry Mean vs. Symmetry SE and by Diagnosis Type") +
  theme_bw()
```




```
cancer_data_clean %>%
  ggplot(aes(x = fractal_dimension_mean, y = fractal_dimension_se, color = diagnosis)) +
  geom_point() +
  ggtitle(label = "Fractal Dimension Mean vs. Fractal Dimension SE and by Diagnosis Type") +
  theme_bw()
```



Conclusion: For most potential predictors, there seems to be a positive relationship between mean and se. Also, records with a high mean and se value are more likely malignant compared to records with a lower mean and se.

Classification Algorithms

Split the data into train and testing

```
set.seed(1899)
# Set an index for train and test dataset
train_index <- createDataPartition(1:nrow(cancer_data_clean), p = 0.8, list = FALSE, times = 1)

# Use index formed above to partition the data accordingly
train_data <- cancer_data_clean[train_index,]
test_data <- cancer_data_clean[-train_index,]
# Train dataset
glimpse(train_data)
```

```
Rows: 457
Columns: 32
$ id                      <int> 842302, 842517, 84300903, 84348301, 84358402, ~
$ diagnosis               <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
$ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290, 12.450~
```

```
$ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15.70, 19.9~
$ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10, 82.57, ~
$ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0, 477.1, ~
$ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0.10030, 0~
$ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0.13280, 0~
$ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0.19800, 0~
$ concave_points_mean     <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0.10430, 0~
$ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809, 0.2087~
$ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0.05883, 0~
$ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572, 0.3345~
$ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813, 0.8902~
$ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.217, 3.18~
$ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27.19, 53.~
$ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110, 0.0114~
$ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580, 0.0246~
$ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0~
$ concave_points_se       <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.0188~
$ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0.01756, 0~
$ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.0051~
$ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.8~
$ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 27.6~
$ perimeter_worst         <dbl> 184.60, 158.80, 152.50, 98.87, 152.20, 103.40,~
$ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, ~
$ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374, 0.1791~
$ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050, 0.5249~
$ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0~
$ concave_points_worst    <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0~
$ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985~
$ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0~
```

```
# Test dataset
glimpse(test_data)
```

```
Rows: 112
Columns: 32
$ id                      <int> 84458202, 84501001, 84667401, 8510653, 852763,~
$ diagnosis               <chr> "M", "M", "M", "B", "M", "M", "M", "M", "M", "~
$ radius_mean             <dbl> 13.710, 12.460, 13.730, 13.080, 14.580, 18.610~
$ texture_mean            <dbl> 20.83, 24.04, 22.61, 15.71, 21.53, 20.25, 25.2~
$ perimeter_mean          <dbl> 90.20, 83.97, 93.60, 85.63, 97.41, 122.10, 102~
$ area_mean               <dbl> 577.9, 475.9, 578.3, 520.0, 644.8, 1094.0, 732~
$ smoothness_mean         <dbl> 0.11890, 0.11860, 0.11310, 0.10750, 0.10540, 0~
$ compactness_mean        <dbl> 0.16450, 0.23960, 0.22930, 0.12700, 0.18680, 0~
$ concavity_mean          <dbl> 0.09366, 0.22730, 0.21280, 0.04568, 0.14250, 0~
$ concave_points_mean     <dbl> 0.059850, 0.085430, 0.080250, 0.031100, 0.0878~
$ symmetry_mean           <dbl> 0.2196, 0.2030, 0.2069, 0.1967, 0.2252, 0.1697~
$ fractal_dimension_mean  <dbl> 0.07451, 0.08243, 0.07682, 0.06811, 0.06924, 0~
$ radius_se               <dbl> 0.5835, 0.2976, 0.2121, 0.1852, 0.2545, 0.8529~
$ texture_se              <dbl> 1.3770, 1.5990, 1.1690, 0.7477, 0.9832, 1.8490~
$ perimeter_se            <dbl> 3.856, 2.039, 2.061, 1.383, 2.110, 5.632, 3.49~
$ area_se                 <dbl> 50.960, 23.940, 19.210, 14.670, 21.050, 93.540~
$ smoothness_se           <dbl> 0.008805, 0.007149, 0.006429, 0.004097, 0.0044~
$ compactness_se          <dbl> 0.030290, 0.072170, 0.059360, 0.018980, 0.0305~
$ concavity_se            <dbl> 0.024880, 0.077430, 0.055010, 0.016980, 0.0268~
```

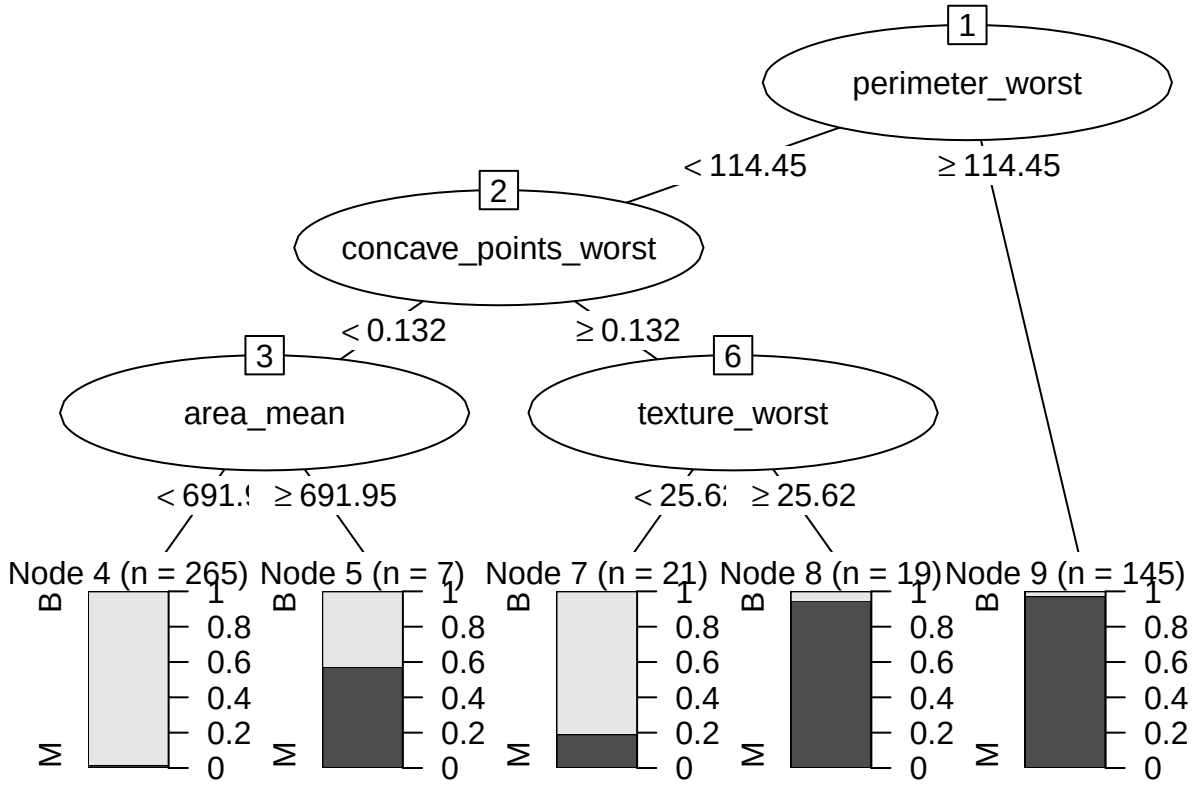
```
$ concave_points_se       <dbl> 0.014480, 0.014320, 0.016280, 0.006490, 0.0135~
$ symmetry_se             <dbl> 0.01486, 0.01789, 0.01961, 0.01678, 0.01454, 0~
$ fractal_dimension_se    <dbl> 0.005412, 0.010080, 0.008093, 0.002425, 0.0037~
$ radius_worst            <dbl> 17.060, 15.090, 15.030, 14.500, 17.620, 21.310~
$ texture_worst           <dbl> 28.14, 40.68, 32.01, 20.49, 33.21, 27.26, 36.7~
$ perimeter_worst         <dbl> 110.60, 97.65, 108.80, 96.09, 122.40, 139.90, ~
$ area_worst              <dbl> 897.0, 711.4, 697.7, 630.5, 896.9, 1403.0, 126~
$ smoothness_worst        <dbl> 0.16540, 0.18530, 0.16510, 0.13120, 0.15250, 0~
$ compactness_worst       <dbl> 0.36820, 1.05800, 0.77250, 0.27760, 0.66430, 0~
$ concavity_worst         <dbl> 0.26780, 1.10500, 0.69430, 0.18900, 0.55390, 0~
$ concave_points_worst    <dbl> 0.15560, 0.22100, 0.22080, 0.07283, 0.27010, 0~
$ symmetry_worst          <dbl> 0.3196, 0.4366, 0.3596, 0.3184, 0.4264, 0.2341~
$ fractal_dimension_worst <dbl> 0.11510, 0.20750, 0.14310, 0.08183, 0.12750, 0~
```

Classification algorithm using decision trees

```
# Build tree using all potential explanatory variables - start with most complex tree possible
cancer_tree <- rpart(diagnosis ~., data = train_data, cp = 0)
cancer_tree
```

```
n= 457

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 457 171 B (0.62582057 0.37417943)
   2) perimeter_worst< 114.45 312  30 B (0.90384615 0.09615385)
     4) concave_points_worst< 0.13235 272   8 B (0.97058824 0.02941176)
       8) area_mean< 691.95 265   4 B (0.98490566 0.01509434) *
       9) area_mean>=691.95 7   3 M (0.42857143 0.57142857) *
     5) concave_points_worst>=0.13235 40  18 M (0.45000000 0.55000000)
      10) texture_worst< 25.62 21   4 B (0.80952381 0.19047619) *
      11) texture_worst>=25.62 19   1 M (0.05263158 0.94736842) *
   3) perimeter_worst>=114.45 145   4 M (0.02758621 0.97241379) *
```

```
plot(as.party(cancer_tree))
```

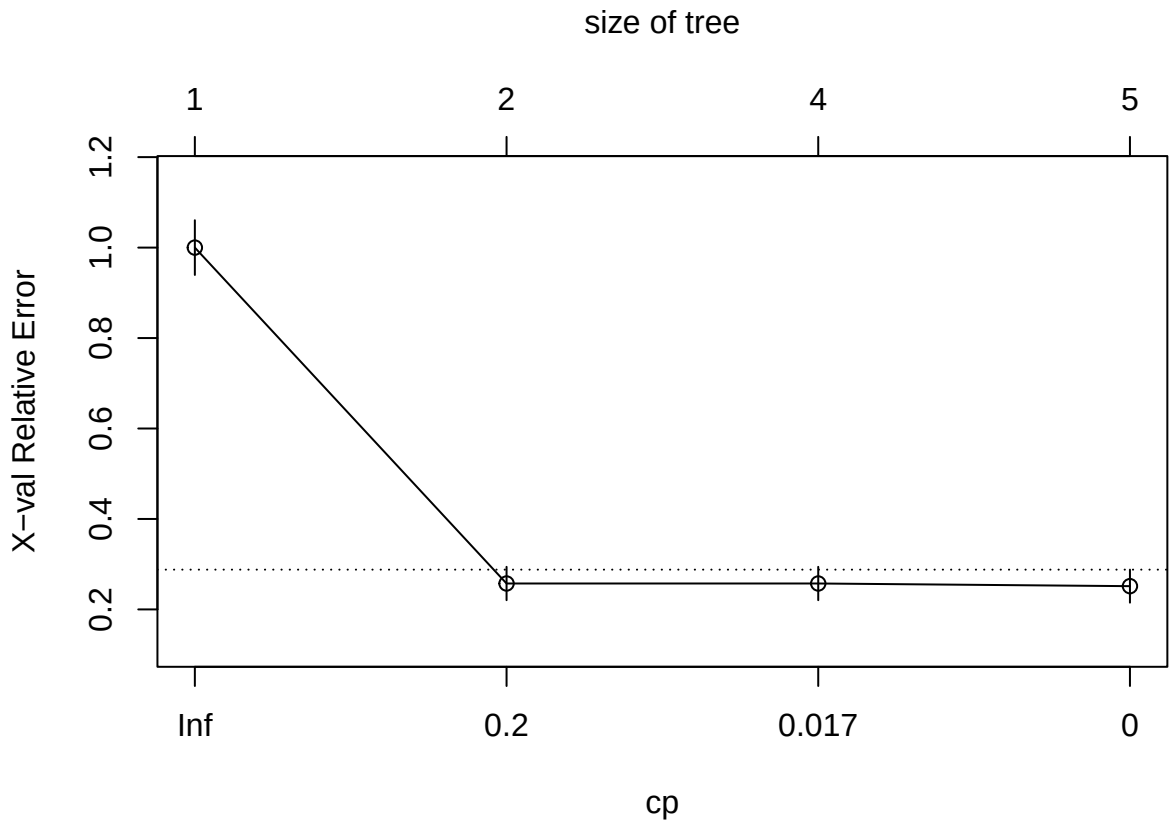



```
# Predict on test data using tree created in the training dataset
test_data$preds <- predict(cancer_tree, newdata = test_data, "class")

# Confusion matrix
confusionMatrix(table(test_data$diagnosis, test_data$preds))
```

```
Confusion Matrix and Statistics


    B  M
  B 66  5
  M  1 40

               Accuracy : 0.9464
                 95% CI : (0.887, 0.9801)
    No Information Rate : 0.5982
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8869

 Mcnemar's Test P-Value : 0.2207

            Sensitivity : 0.9851
            Specificity : 0.8889
         Pos Pred Value : 0.9296
         Neg Pred Value : 0.9756
```

```
            Prevalence : 0.5982
        Detection Rate : 0.5893
  Detection Prevalence : 0.6339
     Balanced Accuracy : 0.9370

      'Positive' Class : B
```

```
# What other levels of complexity would improve accuracy of the decision tree?
plotcp(cancer_tree)
```

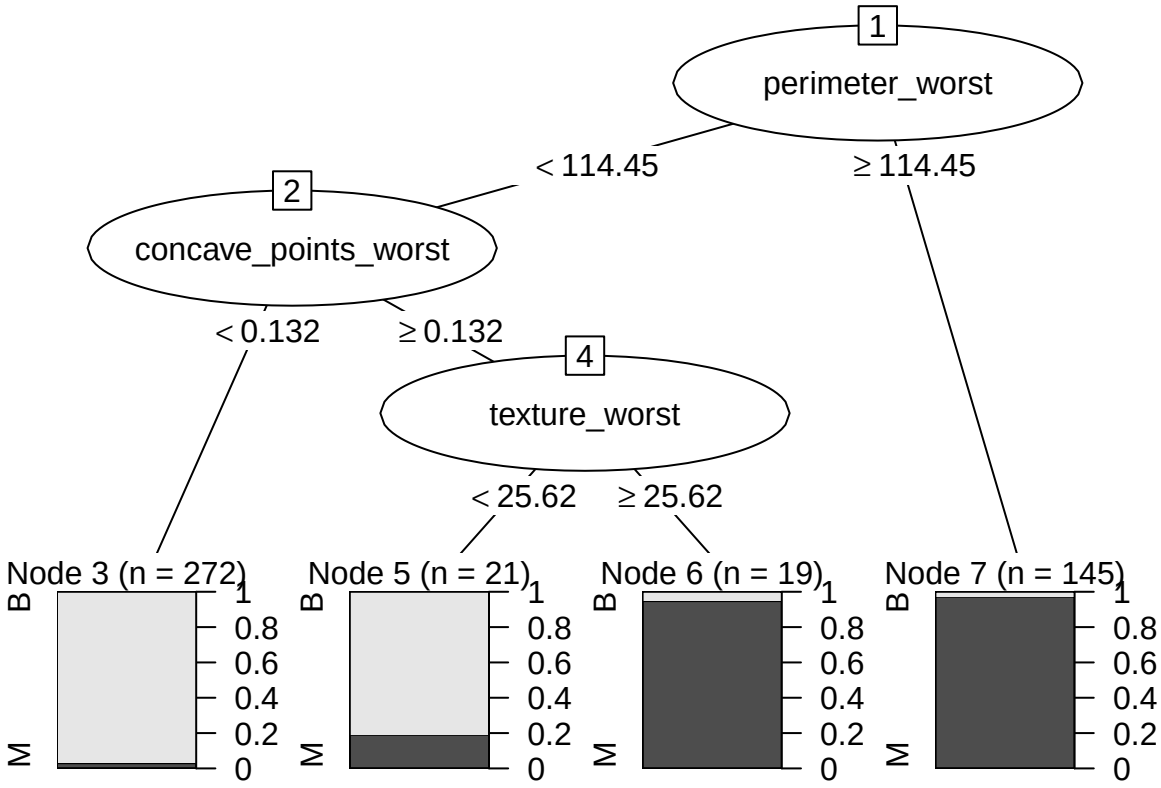



```
# Based on plot above, a complexity parameter of 0.017 may give us low error and high interpretability


# Prune original tree using a cp of 0.017
cancer_tree2 <- prune(cancer_tree, cp = 0.017)
cancer_tree2
```

```
n= 457

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 457 171 B (0.62582057 0.37417943)
```

```
   2) perimeter_worst< 114.45 312  30 B (0.90384615 0.09615385)
     4) concave_points_worst< 0.13235 272   8 B (0.97058824 0.02941176) *
     5) concave_points_worst>=0.13235 40  18 M (0.45000000 0.55000000)
      10) texture_worst< 25.62 21   4 B (0.80952381 0.19047619) *
      11) texture_worst>=25.62 19   1 M (0.05263158 0.94736842) *
   3) perimeter_worst>=114.45 145   4 M (0.02758621 0.97241379) *
```

```
plot(as.party(cancer_tree2))
```




```
# Predict on test data using tree created in the training dataset
test_data$preds2 <- predict(cancer_tree2, newdata = test_data, "class")

# Confusion matrix
confusionMatrix(table(test_data$diagnosis, test_data$preds2))
```

```
Confusion Matrix and Statistics


     B  M
  B 67  4
  M  3 38

               Accuracy : 0.9375
                 95% CI : (0.8755, 0.9745)
    No Information Rate : 0.625
```

```
    P-Value [Acc > NIR] : 1.564e-14

                  Kappa : 0.866

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9571
            Specificity : 0.9048
         Pos Pred Value : 0.9437
         Neg Pred Value : 0.9268
             Prevalence : 0.6250
         Detection Rate : 0.5982
   Detection Prevalence : 0.6339
      Balanced Accuracy : 0.9310

       'Positive' Class : B
```

Conclusion: Although a tree with a complexity parameter of 0.017 is slightly less accurate than the original tree, its lower overall complexity makes it easier to interpret and apply to other similar data.

Classification algorithm using bagging algorithm

```r
# First, turn our outcome variable into a factor variable
train_data$diagnosis <- factor(train_data$diagnosis, levels = c("B", "M"))
test_data$diagnosis <- factor(test_data$diagnosis, levels = c("B", "M"))

# Build random forest using bagging algorithm
formula <- as.formula(diagnosis ~.)
cancer_bagging <- randomForest(formula, data = train_data, mtry = 30, ntree = 500)
cancer_bagging
```

```
Call:
 randomForest(formula = formula, data = train_data, mtry = 30,      ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 30

        OOB estimate of  error rate: 5.03%
Confusion matrix:
    B   M class.error
B 277   9  0.03146853
M  14 157  0.08187135
```

```r
# Predict on test data using bagging algorithm created in the training dataset
test_data$bag_pred <- predict(cancer_bagging, test_data, type = "class")

# Confusion matrix
confusionMatrix(table(test_data$diagnosis, test_data$bag_pred))
```

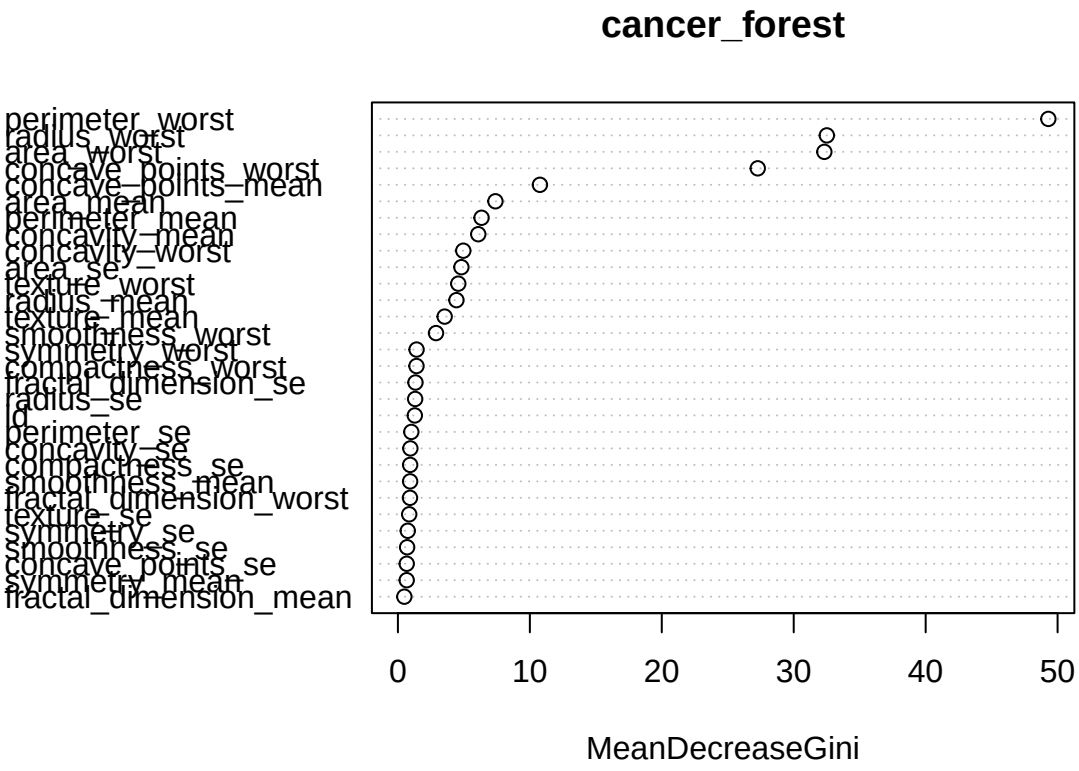
```
Confusion Matrix and Statistics


    B  M
  B 68  3
  M  1 40

               Accuracy : 0.9643
                 95% CI : (0.9111, 0.9902)
    No Information Rate : 0.6161
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9238

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9855
            Specificity : 0.9302
         Pos Pred Value : 0.9577
         Neg Pred Value : 0.9756
             Prevalence : 0.6161
         Detection Rate : 0.6071
   Detection Prevalence : 0.6339
      Balanced Accuracy : 0.9579

       'Positive' Class : B
```

Classification algorithm using random forest

```
# Build random forest using random forest - start with 10 predictors
cancer_forest <- randomForest(formula, data = train_data, mtry = 10, ntree = 500)
cancer_forest
```

```
Call:
 randomForest(formula = formula, data = train_data, mtry = 10,      ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 10

        OOB estimate of  error rate: 4.16%
Confusion matrix:
    B   M class.error
B 278   8  0.02797203
M  11 160  0.06432749
```

```
# How often is a variable being used to make a split?
varImpPlot(cancer_forest)
```




```
# Most important variables seem to be perimeter_worst, concave points worst, area_worst, and radius_wor
# This is very similar to the decision tree algorithm

# Predict on test data using random forest algorithm created in the training dataset
test_data$forest_pred <- predict(cancer_forest, test_data, type = "class")

# Confusion matrix
confusionMatrix(table(test_data$diagnosis, test_data$forest_pred))
```

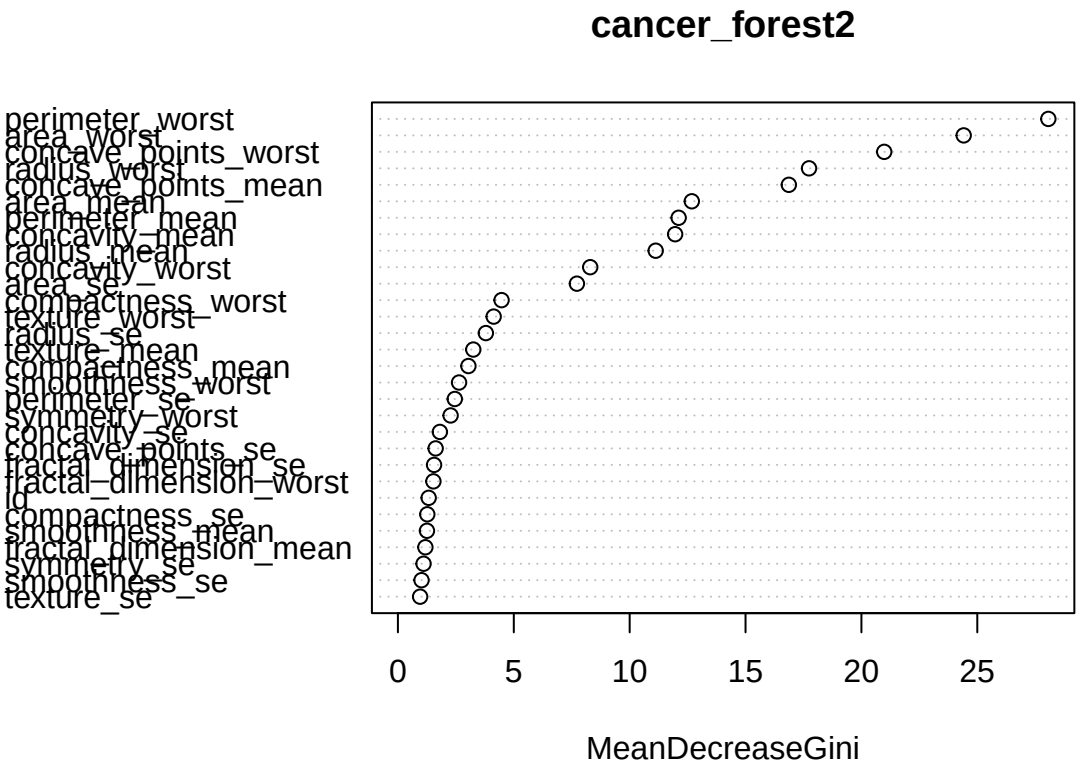
```
Confusion Matrix and Statistics


    B  M
  B 70  1
  M  1 40

               Accuracy : 0.9821
                 95% CI : (0.937, 0.9978)
    No Information Rate : 0.6339
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9615

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9859
```

```
            Specificity : 0.9756
         Pos Pred Value : 0.9859
         Neg Pred Value : 0.9756
             Prevalence : 0.6339
         Detection Rate : 0.6250
   Detection Prevalence : 0.6339
      Balanced Accuracy : 0.9808

       'Positive' Class : B
```

```
# Now try only 4 predictors
cancer_forest2 <- randomForest(formula, data = train_data, mtry = 4, ntree = 500)
cancer_forest2
```

```
Call:
 randomForest(formula = formula, data = train_data, mtry = 4,      ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 4.6%
Confusion matrix:
    B   M class.error
B 277   9  0.03146853
M  12 159  0.07017544
```

```
# How often is a variable being used to make a split?
varImpPlot(cancer_forest2)
```




```
# Same variables are important

# Predict on test data using random forest algorithm created in the training dataset
test_data$forest_pred2 <- predict(cancer_forest2, test_data, type = "class")

# Confusion matrix
confusionMatrix(table(test_data$diagnosis, test_data$forest_pred2))
```

```
Confusion Matrix and Statistics


    B  M
  B 70  1
  M  1 40

               Accuracy : 0.9821
                 95% CI : (0.937, 0.9978)
    No Information Rate : 0.6339
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9615

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9859
            Specificity : 0.9756
```

```
         Pos Pred Value : 0.9859
         Neg Pred Value : 0.9756
             Prevalence : 0.6339
         Detection Rate : 0.6250
   Detection Prevalence : 0.6339
      Balanced Accuracy : 0.9808

       'Positive' Class : B
```

Conclusion: A random forest using 10 predictors at every split, on average, yields the best accuracy.

Classification algorithm using KNN

```
# First, process data to get it ready for knn algorithm

# Turn our outcome variable to dummy variable
train_data$diagnosis <- ifelse(train_data$diagnosis == "M", 1, 0)
test_data$diagnosis <- ifelse(test_data$diagnosis == "M", 1, 0)

# Rescale predictor variables
rescale_x <- function(x){(x-min(x))/(max(x)-min(x))}

# Train data
for (i in names(train_data)[-1:-2]) {
  train_data[,i] <- rescale_x(train_data[,..i])
}

# Test data
for (i in names(test_data)[3:32]) {
  test_data[,i] <- rescale_x(test_data[,..i])
}

glimpse(train_data)
```

```
Rows: 457
Columns: 32
$ id                     <int> 842302, 842517, 84300903, 84348301, 84358402, ~
$ diagnosis              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ radius_mean            <dbl> 0.50438317, 0.63073608, 0.58763896, 0.18262403~
$ texture_mean           <dbl> 0.00000000, 0.25570934, 0.37612457, 0.34602076~
$ perimeter_mean         <dbl> 0.53265045, 0.60449566, 0.58386684, 0.21098307~
$ area_mean              <dbl> 0.35638891, 0.49583798, 0.44306187, 0.09255127~
$ smoothness_mean        <dbl> 0.5937528, 0.2898799, 0.5143089, 0.8113208, 0.~
$ compactness_mean       <dbl> 0.8842545, 0.2029313, 0.4811999, 0.9058284, 0.~
$ concavity_mean         <dbl> 0.70313964, 0.20360825, 0.46251172, 0.56560450~
$ concave_points_mean    <dbl> 0.7311133, 0.3487575, 0.6356859, 0.5228628, 0.~
$ symmetry_mean          <dbl> 0.6863636, 0.3797980, 0.5095960, 0.7762626, 0.~
$ fractal_dimension_mean <dbl> 0.60309388, 0.13604577, 0.20639966, 1.00000000~
$ radius_se              <dbl> 0.35614702, 0.15643672, 0.22962158, 0.13909107~
```

```
$ texture_se              <dbl> 0.15416596, 0.10569037, 0.12067990, 0.22506929~
$ perimeter_se            <dbl> 0.36903360, 0.12444047, 0.18037035, 0.12665504~
$ area_se                 <dbl> 0.27323299, 0.12496355, 0.16225522, 0.03738887~
$ smoothness_se           <dbl> 0.13111759, 0.08987106, 0.12236939, 0.22636405~
$ compactness_se          <dbl> 0.35139844, 0.08132304, 0.28395470, 0.54321507~
$ concavity_se            <dbl> 0.13568182, 0.04696970, 0.09676768, 0.14295455~
$ concave_points_se       <dbl> 0.3006251, 0.2538360, 0.3898466, 0.3536655, 0.~
$ symmetry_se             <dbl> 0.41337863, 0.11213558, 0.27283587, 0.96584419~
$ fractal_dimension_se    <dbl> 0.18096267, 0.08879630, 0.12478309, 0.28539019~
$ radius_worst            <dbl> 0.6104086, 0.5961553, 0.5442585, 0.2277611, 0.~
$ texture_worst           <dbl> 0.12162531, 0.28742842, 0.34524134, 0.37169348~
$ perimeter_worst         <dbl> 0.66143053, 0.53027299, 0.49824615, 0.22561131~
$ area_worst              <dbl> 0.44546447, 0.42983327, 0.36854903, 0.08537614~
$ smoothness_worst        <dbl> 0.5726919, 0.3010258, 0.4467634, 0.9094446, 0.~
$ compactness_worst       <dbl> 0.70268706, 0.16950511, 0.43431510, 0.92608918~
$ concavity_worst         <dbl> 0.56861022, 0.19297125, 0.35974441, 0.54864217~
$ concave_points_worst    <dbl> 0.9120275, 0.6391753, 0.8350515, 0.8848797, 0.~
$ symmetry_worst          <dbl> 0.5984624, 0.2335896, 0.4037059, 1.0000000, 0.~
$ fractal_dimension_worst <dbl> 0.54136996, 0.28806375, 0.27585622, 1.00000000~
```

```
glimpse(test_data)
```

```
Rows: 112
Columns: 37
$ id                      <int> 84458202, 84501001, 84667401, 8510653, 852763,~
$ diagnosis               <dbl> 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0~
$ radius_mean             <dbl> 0.33247690, 0.27071496, 0.33346509, 0.30134888~
$ texture_mean            <dbl> 0.51938347, 0.66931340, 0.60252219, 0.28024288~
$ perimeter_mean          <dbl> 0.33555057, 0.29050683, 0.36013303, 0.30250886~
$ area_mean               <dbl> 0.20621885, 0.15779729, 0.20640873, 0.17873249~
$ smoothness_mean         <dbl> 0.82885906, 0.82382550, 0.73154362, 0.63758389~
$ compactness_mean        <dbl> 0.43812896, 0.67138775, 0.63939620, 0.32165486~
$ concavity_mean          <dbl> 0.24949387, 0.60548748, 0.56686201, 0.12168354~
$ concave_points_mean     <dbl> 0.31869010, 0.45489883, 0.42731629, 0.16560170~
$ symmetry_mean           <dbl> 0.5917194, 0.4962622, 0.5186889, 0.4600345, 0.~
$ fractal_dimension_mean  <dbl> 0.65519082, 0.86655991, 0.71684014, 0.48438751~
$ radius_se               <dbl> 0.44133283, 0.16992595, 0.08876021, 0.06322385~
$ texture_se              <dbl> 0.21671951, 0.26628857, 0.17027643, 0.07620685~
$ perimeter_se            <dbl> 0.35533256, 0.12157468, 0.12440499, 0.03717998~
$ area_se                 <dbl> 0.228918911, 0.088844882, 0.064324151, 0.04078~
$ smoothness_se           <dbl> 0.4950094, 0.3794235, 0.3291687, 0.1663991, 0.~
$ compactness_se          <dbl> 0.37759151, 1.00000000, 0.80962147, 0.20950555~
$ concavity_se            <dbl> 0.22804766, 0.70971586, 0.50421632, 0.15563703~
$ concave_points_se       <dbl> 0.5236890, 0.5179024, 0.5887884, 0.2347197, 0.~
$ symmetry_se             <dbl> 0.066559860, 0.110690358, 0.135741334, 0.09452~
$ fractal_dimension_se    <dbl> 0.49179114, 1.00000000, 0.78367374, 0.16659409~
$ radius_worst            <dbl> 0.36244541, 0.28423978, 0.28185788, 0.26081778~
$ texture_worst           <dbl> 0.45873648, 0.81559476, 0.56886739, 0.24103586~
$ perimeter_worst         <dbl> 0.35324843, 0.27724632, 0.34268443, 0.26809085~
$ area_worst              <dbl> 0.21923124, 0.16206727, 0.15784773, 0.13715043~
$ smoothness_worst        <dbl> 0.6400190, 0.7751817, 0.6379814, 0.4077294, 0.~
$ compactness_worst       <dbl> 0.33075259, 1.00000000, 0.72300647, 0.24285201~
$ concavity_worst         <dbl> 0.24235294, 1.00000000, 0.62832579, 0.17104072~
$ concave_points_worst    <dbl> 0.53599724, 0.76128143, 0.76059249, 0.25087840~
```

```
$ symmetry_worst          <dbl> 0.38638228, 0.70631665, 0.49576155, 0.38310090~
$ fractal_dimension_worst <dbl> 0.38625042, 1.00000000, 0.57223514, 0.16526071~
$ preds                   <fct> M, M, M, B, M, M, M, M, M, M, B, B, M, B, B, B~
$ preds2                  <fct> M, M, M, B, M, M, M, M, M, M, B, B, M, B, B, B~
$ bag_pred                <fct> M, M, M, B, M, M, M, M, M, M, B, B, M, B, B, B~
$ forest_pred             <fct> M, M, M, B, M, M, M, M, M, M, B, B, M, B, B, B~
$ forest_pred2            <fct> M, M, M, B, M, M, M, M, M, M, B, B, M, B, B, B~
```

```
# Create function for knn algorithm
knn_fun <- function(k_value){
  cancer_knn <- knn(train = train_data[,3:32],
                    test = test_data[,3:32],
                       cl = train_data$diagnosis, k = k_value)
  cancer_knn_table <- table(test_data$diagnosis, cancer_knn)
  return(cancer_knn_table)
}

cancer_knn1 <- knn_fun(1) # K = 1
cancer_knn10 <- knn_fun(10) # K = 10
cancer_knn25 <- knn_fun(25) # K = 25

confusionMatrix(cancer_knn1)
```

```
Confusion Matrix and Statistics

   cancer_knn
     0  1
  0 62  9
  1  1 40

               Accuracy : 0.9107
                 95% CI : (0.8419, 0.9564)
    No Information Rate : 0.5625
    P-Value [Acc > NIR] : 5.395e-16

                  Kappa : 0.8152

 Mcnemar's Test P-Value : 0.02686

            Sensitivity : 0.9841
            Specificity : 0.8163
         Pos Pred Value : 0.8732
         Neg Pred Value : 0.9756
             Prevalence : 0.5625
         Detection Rate : 0.5536
   Detection Prevalence : 0.6339
      Balanced Accuracy : 0.9002

       'Positive' Class : 0
```

```
confusionMatrix(cancer_knn10)
```

```
Confusion Matrix and Statistics
```

```
   cancer_knn
     0  1
  0 63  8
  1  0 41

               Accuracy : 0.9286
                 95% CI : (0.8641, 0.9687)
    No Information Rate : 0.5625
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8522

 Mcnemar's Test P-Value : 0.01333

            Sensitivity : 1.0000
            Specificity : 0.8367
         Pos Pred Value : 0.8873
         Neg Pred Value : 1.0000
             Prevalence : 0.5625
         Detection Rate : 0.5625
   Detection Prevalence : 0.6339
      Balanced Accuracy : 0.9184

       'Positive' Class : 0
```

```
confusionMatrix(cancer_knn25)
```

```
Confusion Matrix and Statistics

   cancer_knn
     0  1
  0 64  7
  1  0 41

               Accuracy : 0.9375
                 95% CI : (0.8755, 0.9745)
    No Information Rate : 0.5714
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.87

 Mcnemar's Test P-Value : 0.02334

            Sensitivity : 1.0000
            Specificity : 0.8542
         Pos Pred Value : 0.9014
         Neg Pred Value : 1.0000
             Prevalence : 0.5714
         Detection Rate : 0.5714
   Detection Prevalence : 0.6339
      Balanced Accuracy : 0.9271
```

```
'Positive' Class : 0
```

Conclusion: Using a knn value of 25 yields the best accuracy.