

Análise de Câncer de Mama

Thiago Silva

26/04/2021

O câncer de mama é a doença maligna mais comum entre as mulheres, sendo responsável por quase 1 em cada 3 cânceres diagnosticados entre as mulheres nos Estados Unidos, e é a segunda principal causa de morte por câncer entre as mulheres. O câncer de mama ocorre como resultado do crescimento anormal de células no tecido mamário, comumente referido como tumor. Um tumor não significa câncer - os tumores podem ser benignos (não cancerosos), pré-malignos (pré-cancerosos) ou malignos (cancerosos). Testes como ressonância magnética, mamografia, ultrassom e biópsia são comumente usados para diagnosticar o câncer de mama.

Este conjunto de dados foi criado pelo Dr. William H. Wolberg, médico do Hospital da Universidade de Wisconsin em Madison, Wisconsin, EUA. Para criar o conjunto de dados, o Dr. Wolberg usou amostras de fluidos, retiradas de pacientes com massas mamárias sólidas e um programa gráfico de computador fácil de usar chamado Xcyt, que é capaz de realizar a análise de características citológicas com base em uma varredura digital. O programa usa um algoritmo de ajuste de curva, para calcular dez características de cada uma das células da amostra, depois calcula o valor médio, valor extremo e erro padrão de cada característica para a imagem, retornando um vetor de 30 reais.

Informação de Atributo:

ID número 2) Diagnóstico (M = maligno, B = benigno) 3-32) Dez recursos de valor real são calculados para cada núcleo da célula:

raio (média das distâncias do centro aos pontos do perímetro) textura (desvio padrão dos valores da escala de cinza) perímetro área suavidade (variação local em comprimentos de raio) compacidade ($\text{perímetro}^2 / \text{área} - 1,0$) concavidade (gravidade das porções côncavas do contorno) pontos côncavos (número de porções côncavas do contorno) simetria dimensão fractal ("aproximação do litoral" - 1)

Objetivo: Esta análise visa observar quais características são mais úteis na previsão do câncer maligno ou benigno e para ver as tendências gerais que podem nos ajudar na seleção de modelo e seleção de hiperparâmetro. O objetivo é classificar se o câncer de mama é benigno ou maligno. Para conseguir isso, usamos métodos de classificação de aprendizado de máquina para ajustar uma função que pode prever a classe discreta de novos dados de entrada.

Esta base de dados se encontra no pacote "mclust" e pode ser carregada no R através do código abaixo:

```
library(mclust)
```

```
## Package 'mclust' version 5.4.7  
## Type 'citation("mclust")' for citing this R package in publications.
```

```
BD <- wdbc
```

Demonstrando a tabela de dados

```
head(BD, 10)
```

##	ID	Diagnosis	Radius_mean	Texture_mean	Perimeter_mean	Area_mean
## 1	842302	M	17.99	10.38	122.80	1001.0
## 2	842517	M	20.57	17.77	132.90	1326.0
## 3	84300903	M	19.69	21.25	130.00	1203.0
## 4	84348301	M	11.42	20.38	77.58	386.1
## 5	84358402	M	20.29	14.34	135.10	1297.0
## 6	843786	M	12.45	15.70	82.57	477.1
## 7	844359	M	18.25	19.98	119.60	1040.0
## 8	84458202	M	13.71	20.83	90.20	577.9
## 9	844981	M	13.00	21.82	87.50	519.8
## 10	84501001	M	12.46	24.04	83.97	475.9
##	Smoothness_mean	Compactness_mean	Concavity_mean	Nconcave_mean	Symmetry_mean	
## 1	0.11840	0.27760	0.30010	0.14710	0.2419	
## 2	0.08474	0.07864	0.08690	0.07017	0.1812	
## 3	0.10960	0.15990	0.19740	0.12790	0.2069	
## 4	0.14250	0.28390	0.24140	0.10520	0.2597	
## 5	0.10030	0.13280	0.19800	0.10430	0.1809	
## 6	0.12780	0.17000	0.15780	0.08089	0.2087	
## 7	0.09463	0.10900	0.11270	0.07400	0.1794	
## 8	0.11890	0.16450	0.09366	0.05985	0.2196	
## 9	0.12730	0.19320	0.18590	0.09353	0.2350	
## 10	0.11860	0.23960	0.22730	0.08543	0.2030	
##	Fractaldim_mean	Radius_se	Texture_se	Perimeter_se	Area_se	Smoothness_se
## 1	0.07871	1.0950	0.9053	8.589	153.40	0.006399
## 2	0.05667	0.5435	0.7339	3.398	74.08	0.005225
## 3	0.05999	0.7456	0.7869	4.585	94.03	0.006150
## 4	0.09744	0.4956	1.1560	3.445	27.23	0.009110
## 5	0.05883	0.7572	0.7813	5.438	94.44	0.011490
## 6	0.07613	0.3345	0.8902	2.217	27.19	0.007510
## 7	0.05742	0.4467	0.7732	3.180	53.91	0.004314
## 8	0.07451	0.5835	1.3770	3.856	50.96	0.008805
## 9	0.07389	0.3063	1.0020	2.406	24.32	0.005731
## 10	0.08243	0.2976	1.5990	2.039	23.94	0.007149
##	Compactness_se	Concavity_se	Nconcave_se	Symmetry_se	Fractaldim_se	
## 1	0.04904	0.05373	0.01587	0.03003	0.006193	
## 2	0.01308	0.01860	0.01340	0.01389	0.003532	
## 3	0.04006	0.03832	0.02058	0.02250	0.004571	
## 4	0.07458	0.05661	0.01867	0.05963	0.009208	
## 5	0.02461	0.05688	0.01885	0.01756	0.005115	
## 6	0.03345	0.03672	0.01137	0.02165	0.005082	
## 7	0.01382	0.02254	0.01039	0.01369	0.002179	
## 8	0.03029	0.02488	0.01448	0.01486	0.005412	
## 9	0.03502	0.03553	0.01226	0.02143	0.003749	
## 10	0.07217	0.07743	0.01432	0.01789	0.010080	
##	Radius_extreme	Texture_extreme	Perimeter_extreme	Area_extreme		
## 1	25.38	17.33	184.60	2019.0		
## 2	24.99	23.41	158.80	1956.0		
## 3	23.57	25.53	152.50	1709.0		
## 4	14.91	26.50	98.87	567.7		
## 5	22.54	16.67	152.20	1575.0		
## 6	15.47	23.75	103.40	741.6		
## 7	22.88	27.66	153.20	1606.0		
## 8	17.06	28.14	110.60	897.0		
## 9	15.49	30.73	106.20	739.3		
## 10	15.09	40.68	97.65	711.4		
##	Smoothness_extreme	Compactness_extreme	Concavity_extreme	Nconcave_extreme		
## 1	0.1622		0.6656	0.7119	0.2654	

```
## 2      0.1238      0.1866      0.2416      0.1860
## 3      0.1444      0.4245      0.4504      0.2430
## 4      0.2098      0.8663      0.6869      0.2575
## 5      0.1374      0.2050      0.4000      0.1625
## 6      0.1791      0.5249      0.5355      0.1741
## 7      0.1442      0.2576      0.3784      0.1932
## 8      0.1654      0.3682      0.2678      0.1556
## 9      0.1703      0.5401      0.5390      0.2060
## 10     0.1853      1.0580      1.1050      0.2210
##      Symmetry_extreme Fractaldim_extreme
## 1      0.4601      0.11890
## 2      0.2750      0.08902
## 3      0.3613      0.08758
## 4      0.6638      0.17300
## 5      0.2364      0.07678
## 6      0.3985      0.12440
## 7      0.3063      0.08368
## 8      0.3196      0.11510
## 9      0.4378      0.10720
## 10     0.4366      0.20750
```

Verificando dimensoes do dataframe

```
dim(BD)
```

```
## [1] 569 32
```

Neste caso, temos 569 linhas e 32 colunas.

Verficando proporcao de dados benignos e malignos

```
table(BD$Diagnosis)
```

```
##
##  B   M
## 357 212
```

```
prop.table(table(BD$Diagnosis))
```

```
##
##      B      M
## 0.6274165 0.3725835
```

A distribuição de cancer de mama desta base de dados está dividida em 62% de casos benignos e 37% de casos malignos.

Para garantirmos a qualidade das análises, verificamos a porcentagem de valores ausentes em cada coluna.

```
anyNA(BD)
```

```
## [1] FALSE
```

```
NAS <- round(colSums(is.na(BD))*100/nrow(BD), 2)
NAS
```

```
##          ID          Diagnosis      Radius_mean      Texture_mean
##          0            0            0            0
##    Perimeter_mean      Area_mean    Smoothness_mean    Compactness_mean
##          0            0            0            0
##    Concavity_mean      Nconcave_mean    Symmetry_mean    Fractaldim_mean
##          0            0            0            0
##          Radius_se      Texture_se      Perimeter_se      Area_se
##          0            0            0            0
##    Smoothness_se      Compactness_se      Concavity_se      Nconcave_se
##          0            0            0            0
##    Symmetry_se      Fractaldim_se      Radius_extreme      Texture_extreme
##          0            0            0            0
##    Perimeter_extreme      Area_extreme    Smoothness_extreme    Compactness_extreme
##          0            0            0            0
##    Concavity_extreme      Nconcave_extreme    Symmetry_extreme    Fractaldim_extreme
##          0            0            0            0
```

Como não havia nenhum dado ausente em nossa demos continuidade a análise.

Em nossa base consta uma Variavel ID que não nos será útil, assim a removemos.

```
BD <- BD[, -1]
```

Para compreendermos um pouco mais da nossa base, visualizamos como estão distribuidos os diagnósticos quando cruzados com as demais variáveis através de boxplots.

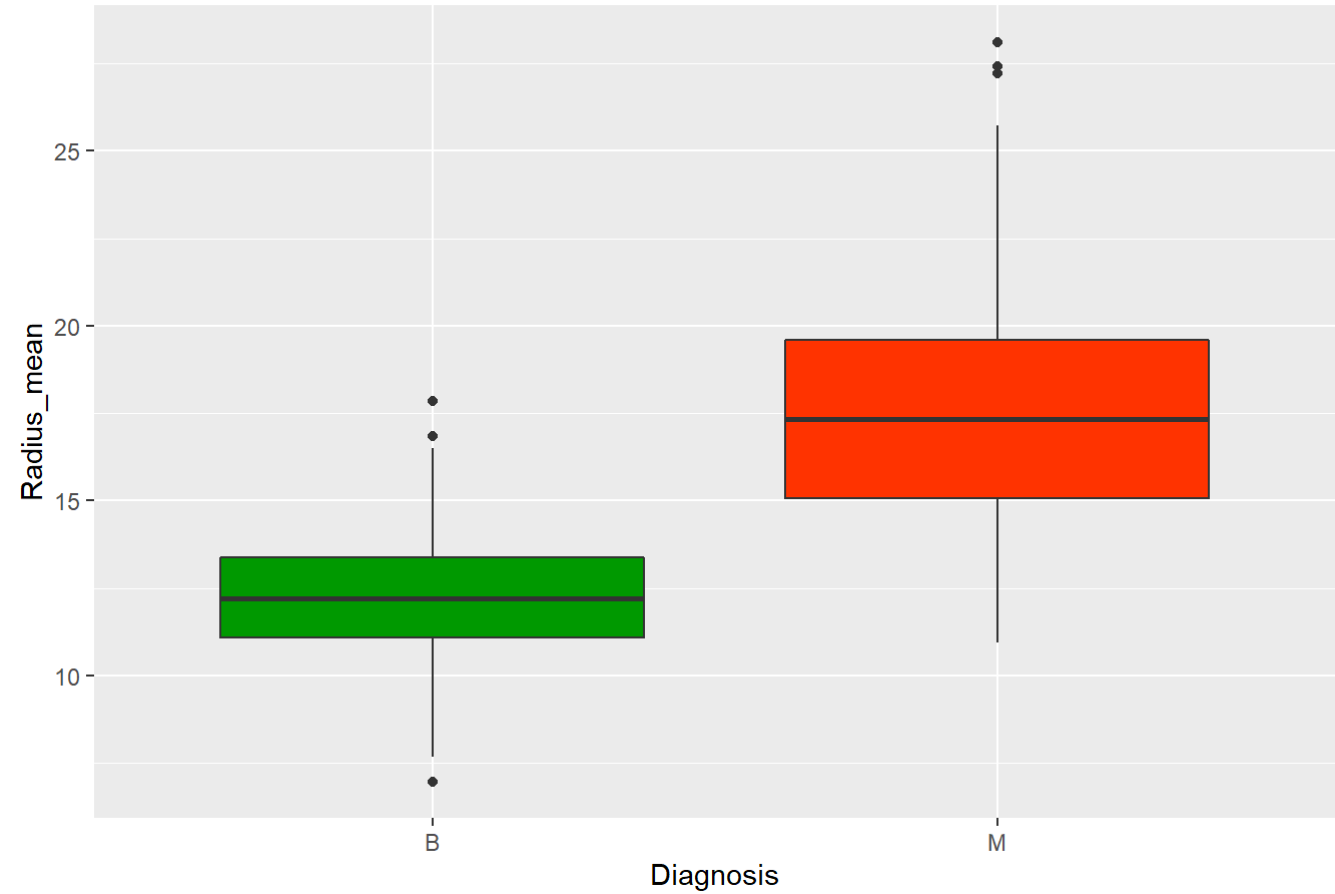
```
library(ggplot2)

box.plot <- function(x){
  if(is.numeric(BD[, x])){
    ggplot(BD, aes_string('Diagnosis', x)) +
      geom_boxplot( fill=c('#009900', '#FF3300')) +
      ggtitle(paste('Diagnósticos por ', x))
  }
}

col.names = names(BD)
sapply(col.names, box.plot)
```

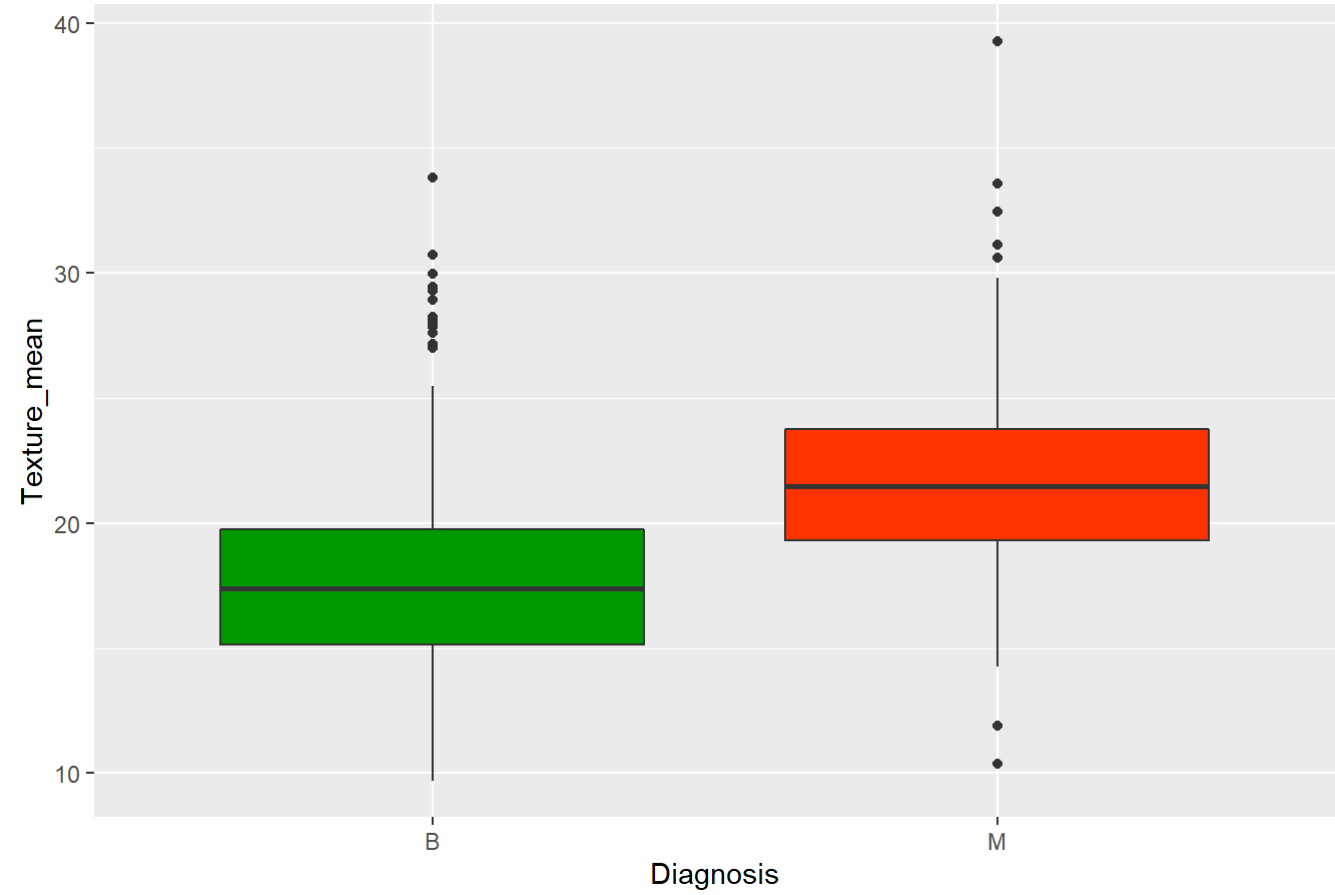
```
## $Diagnosis
## NULL
##
## $Radius_mean
```

Diagnósticos por Radius_mean



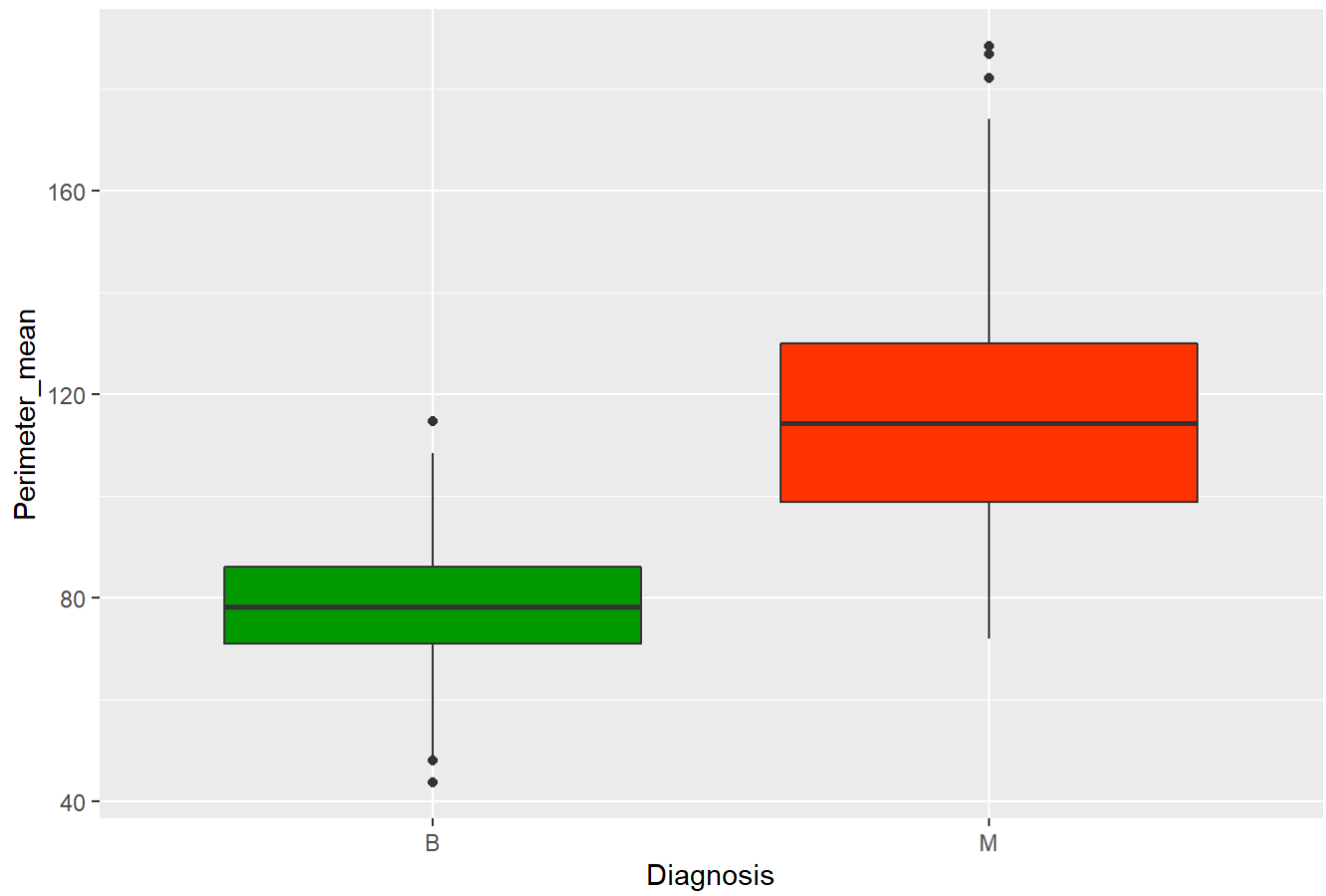
\$Texture_mean

Diagnósticos por Texture_mean



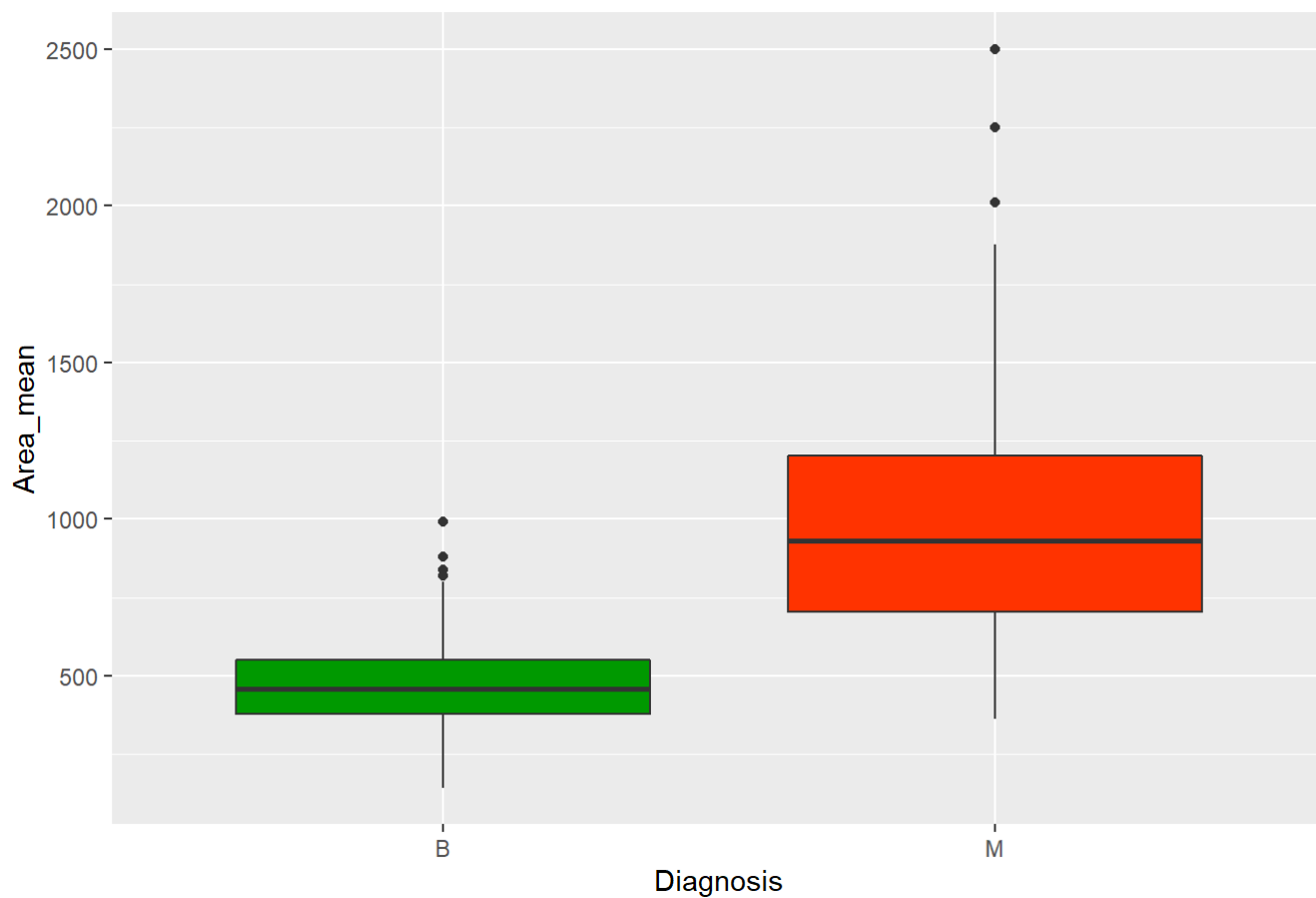
```
##  
## $Perimeter_mean
```

Diagnósticos por Perimeter_mean



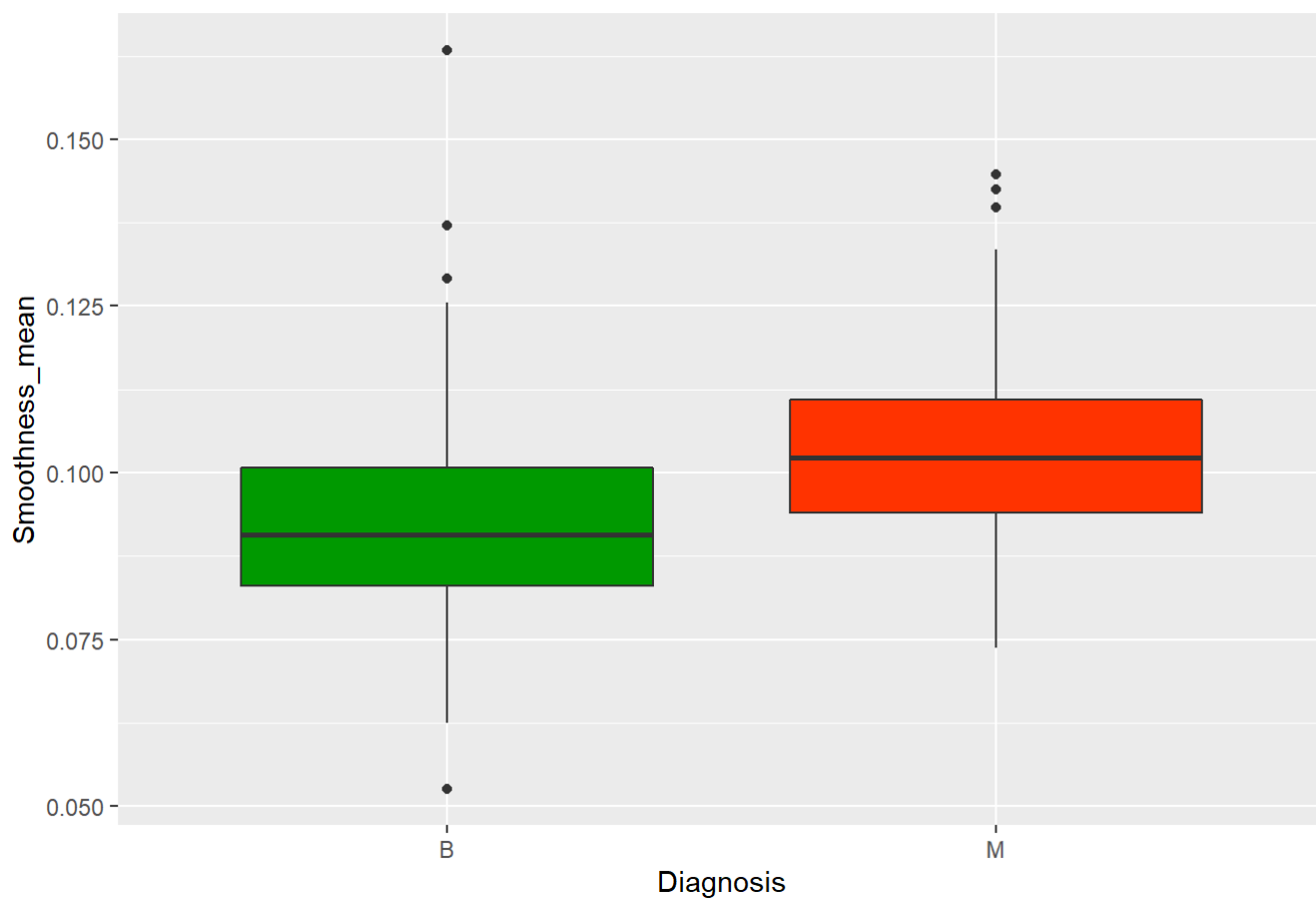
```
##  
## $Area_mean
```

Diagnósticos por Area_mean

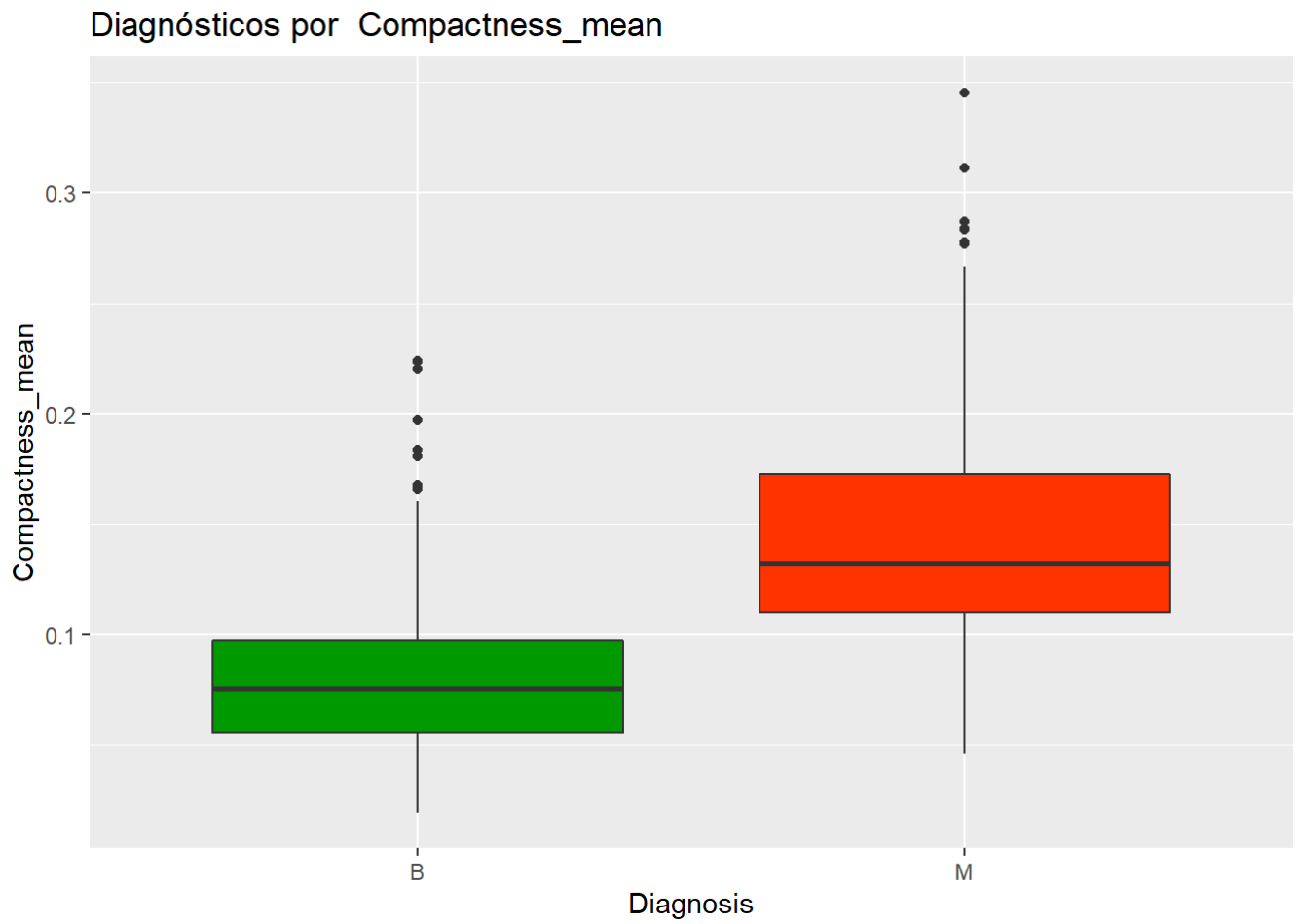


\$Smoothness_mean

Diagnósticos por Smoothness_mean

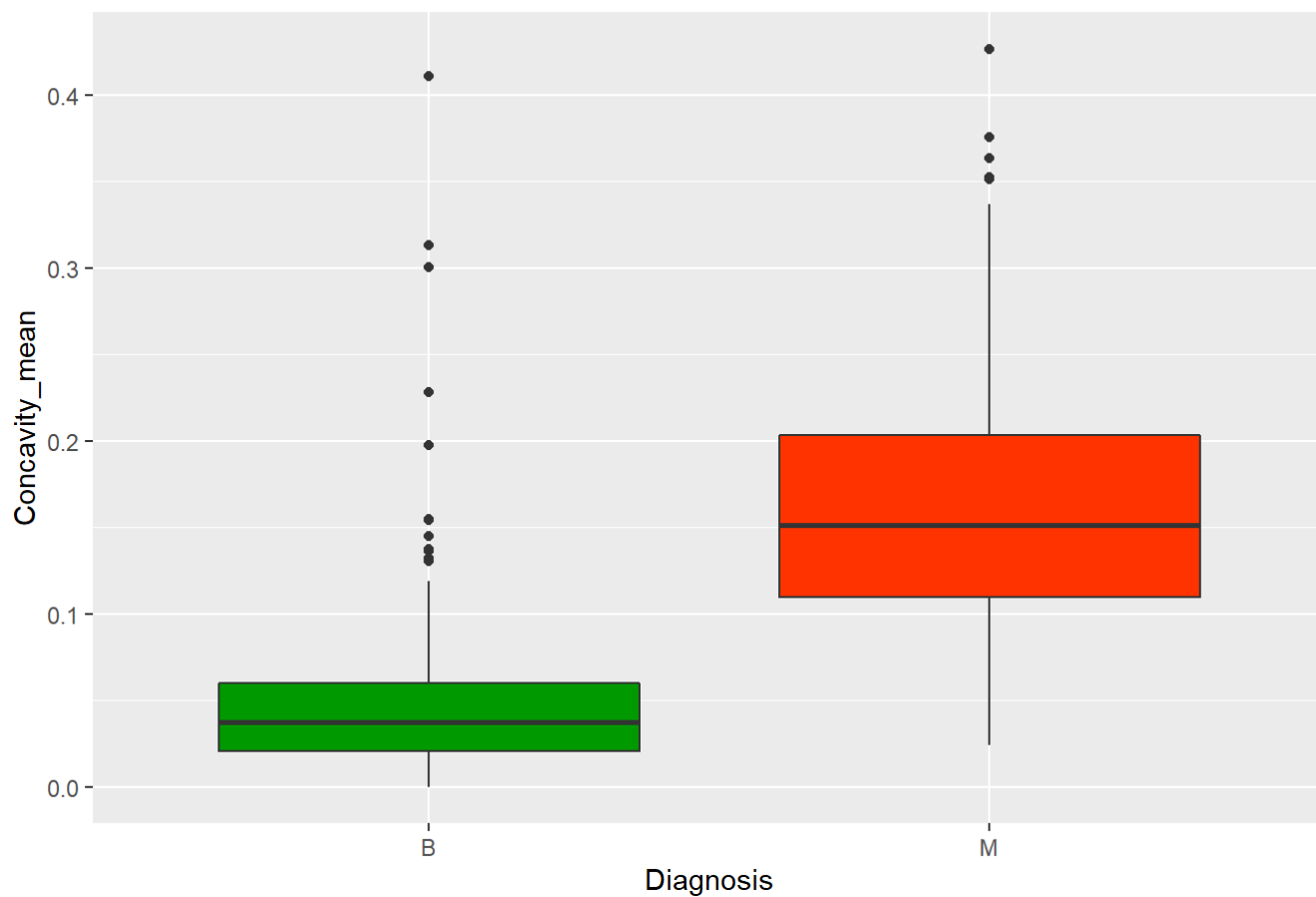


```
##  
## $Compactness_mean
```



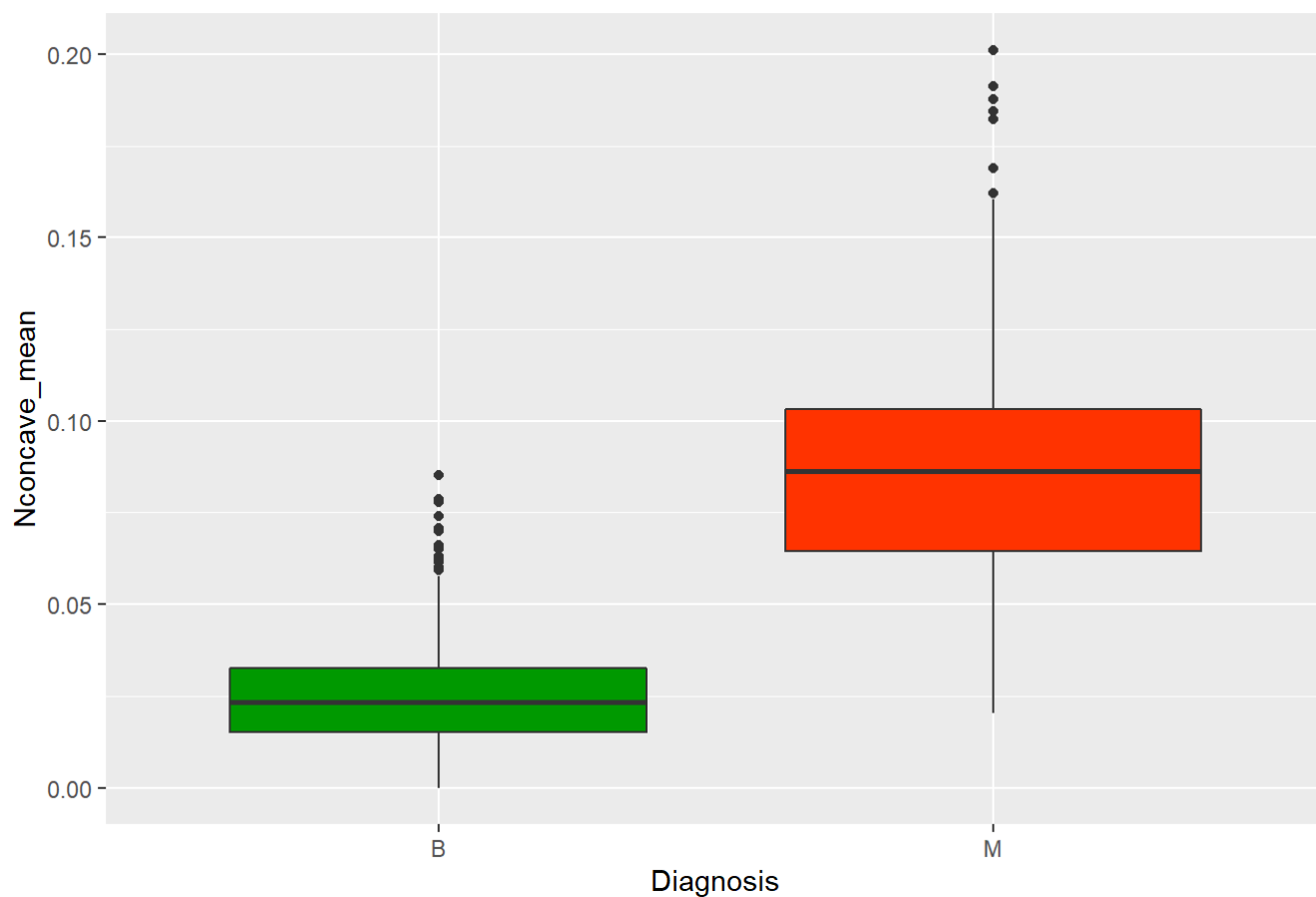
```
##  
## $Concavity_mean
```


Diagnósticos por Concavity_mean

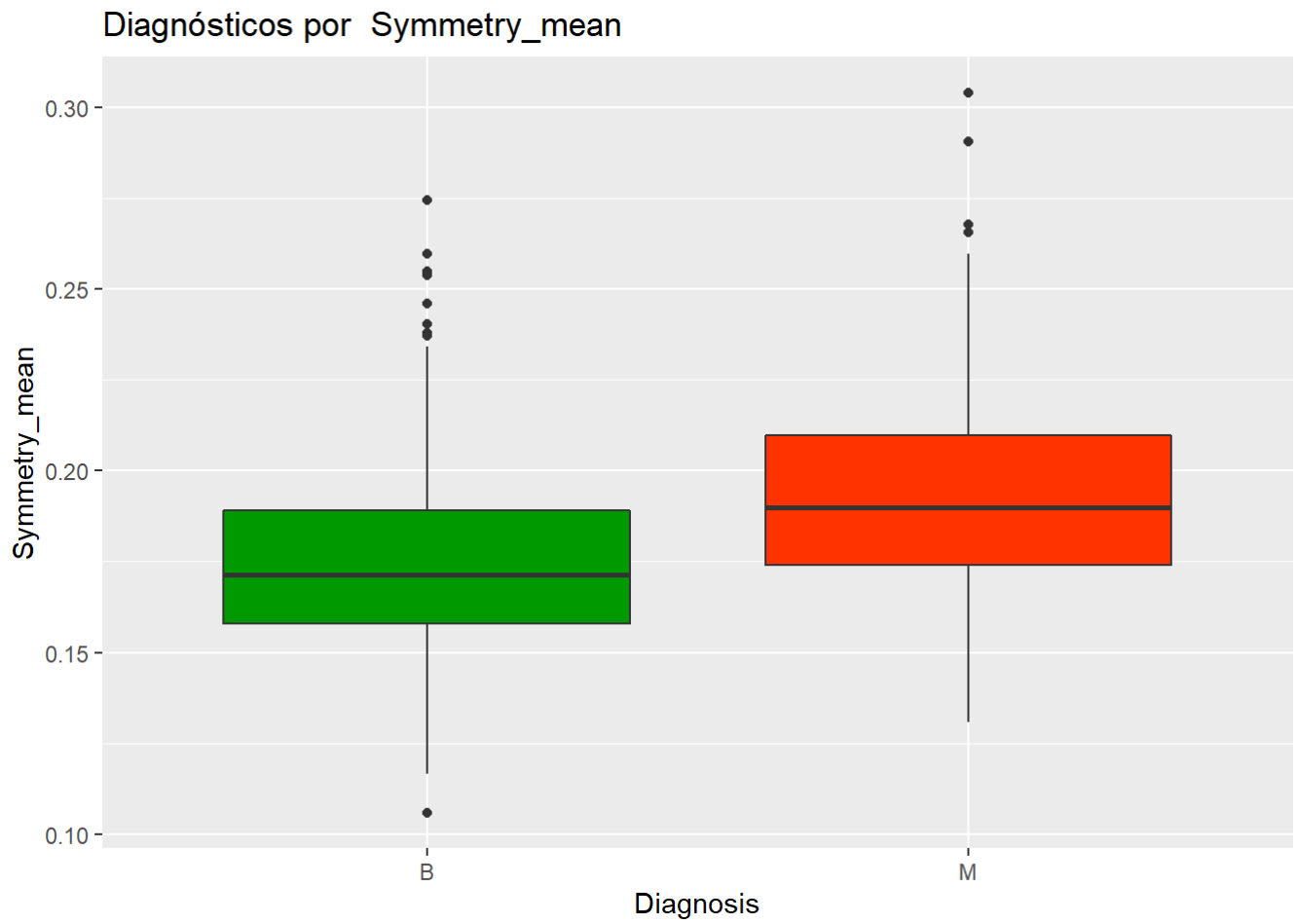


\$Nconcave_mean

Diagnósticos por Nconcave_mean

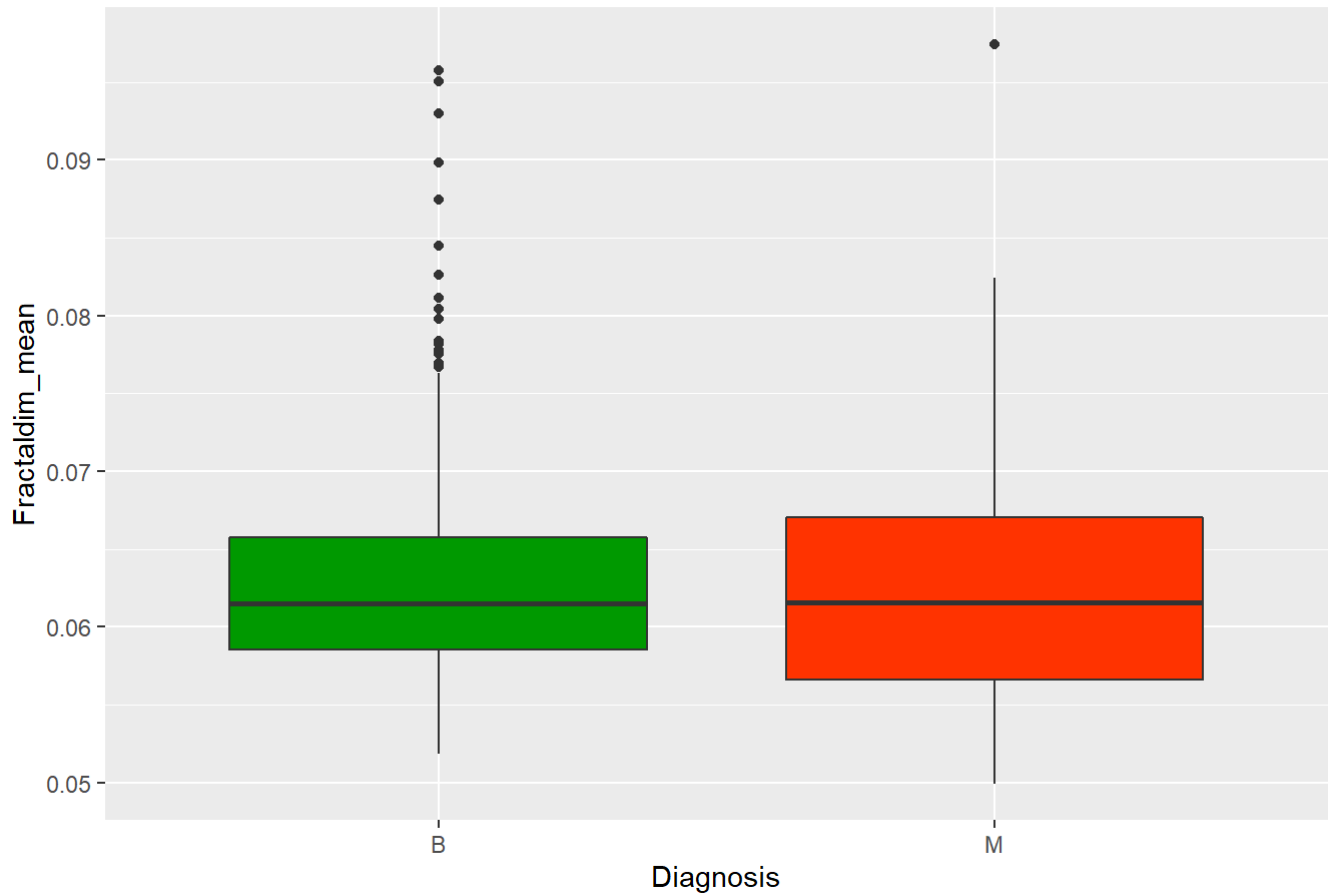


```
##  
## $Symmetry_mean
```



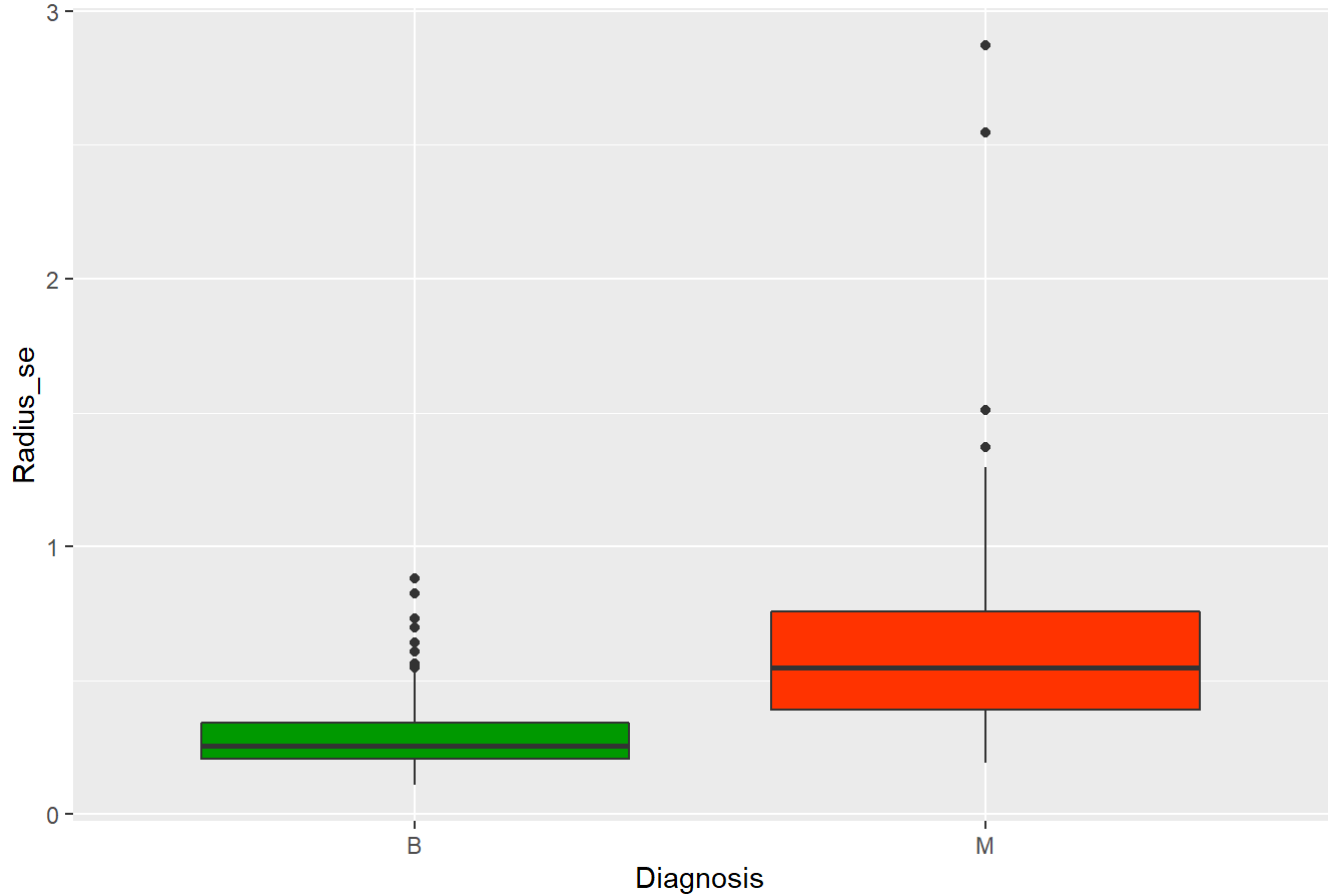
```
##  
## $Fractaldim_mean
```

Diagnósticos por Fractaldim_mean

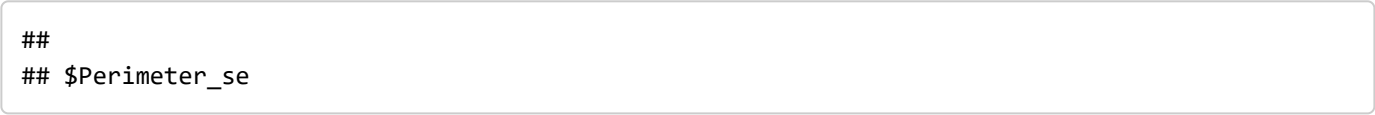


```
##  
## $Radius_se
```

Diagnósticos por Radius_se

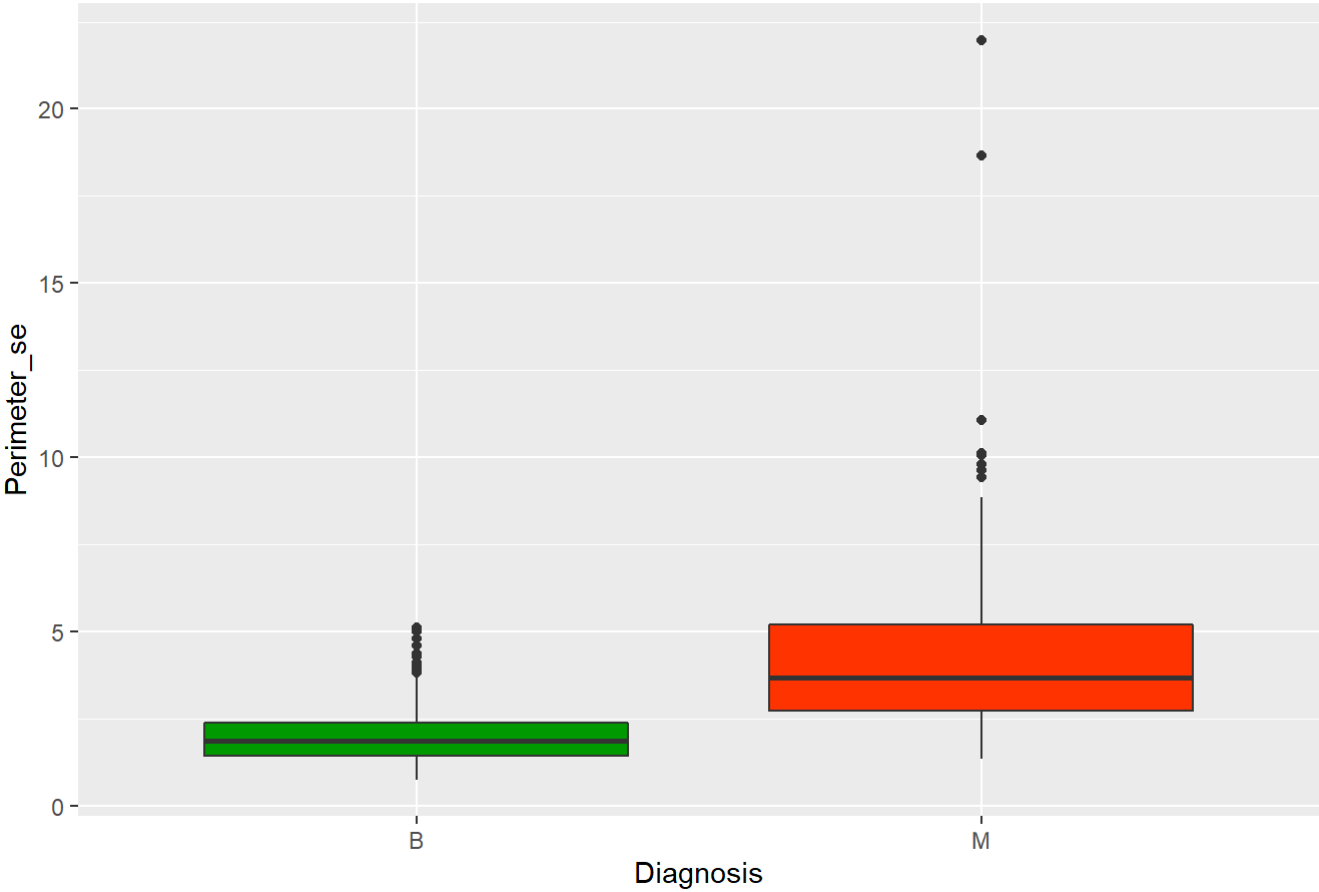


```
##
## $Texture_se
```



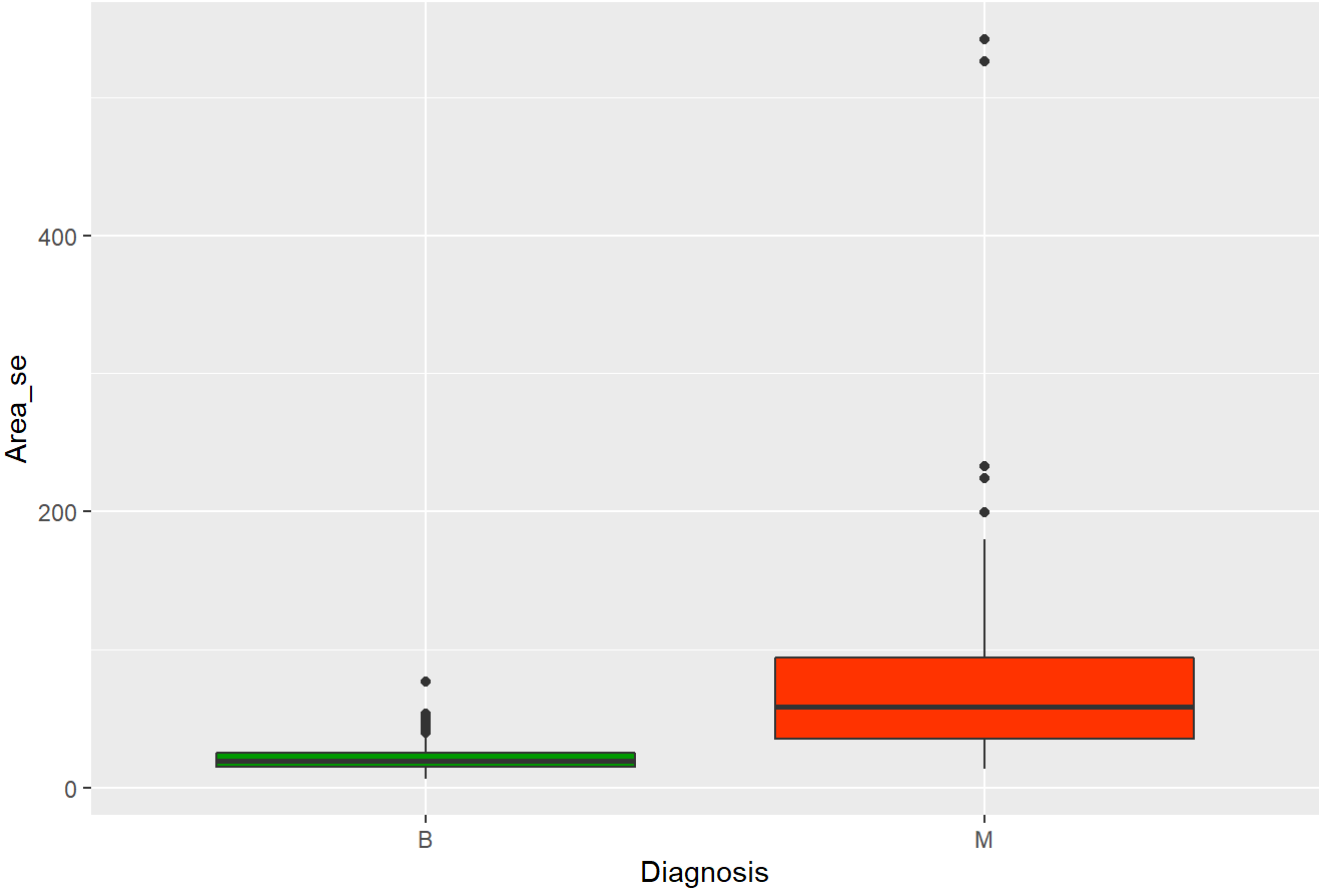
```
##
## $Perimeter_se
```

Diagnósticos por Perimeter_se

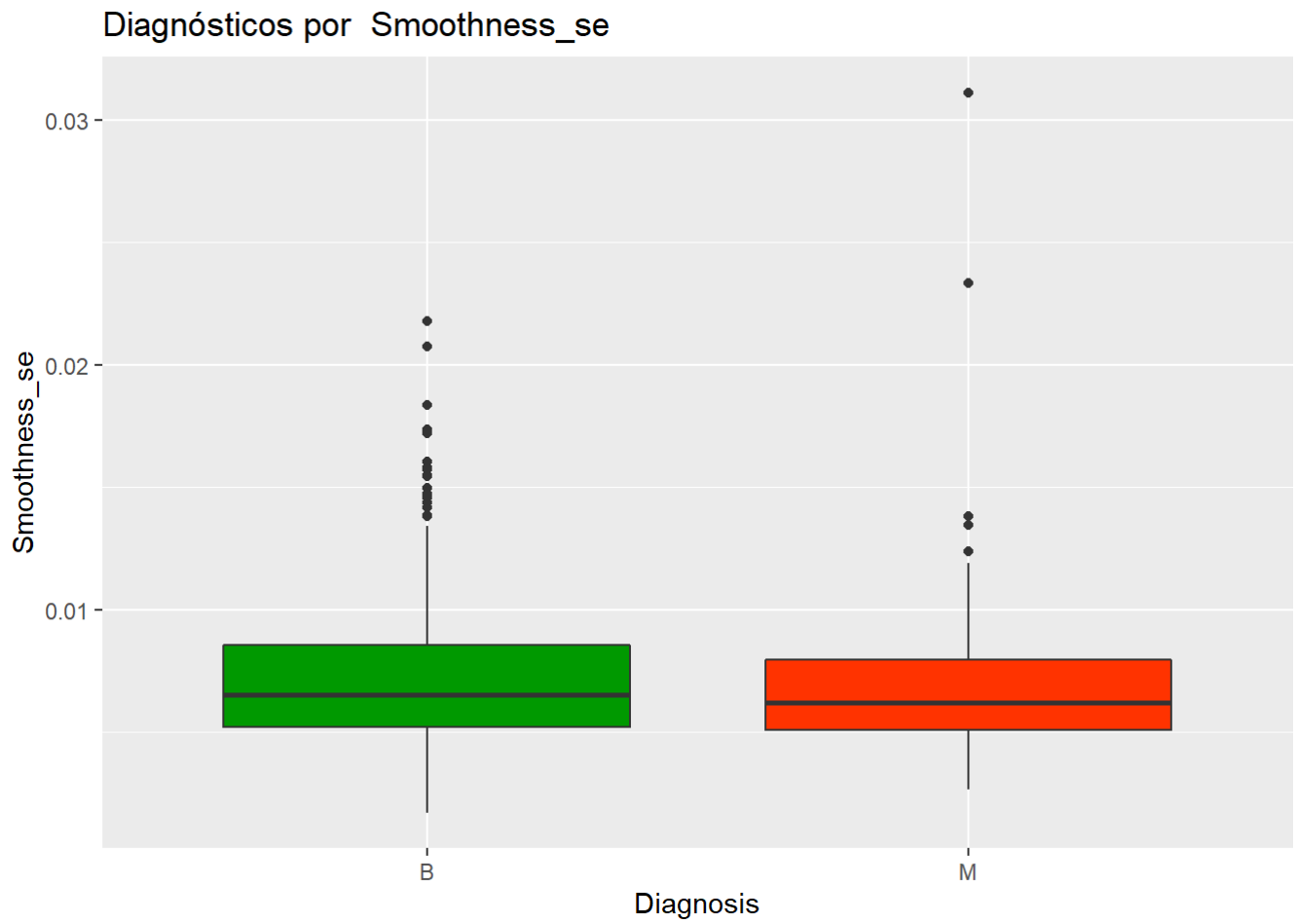


\$Area_se

Diagnósticos por Area_se

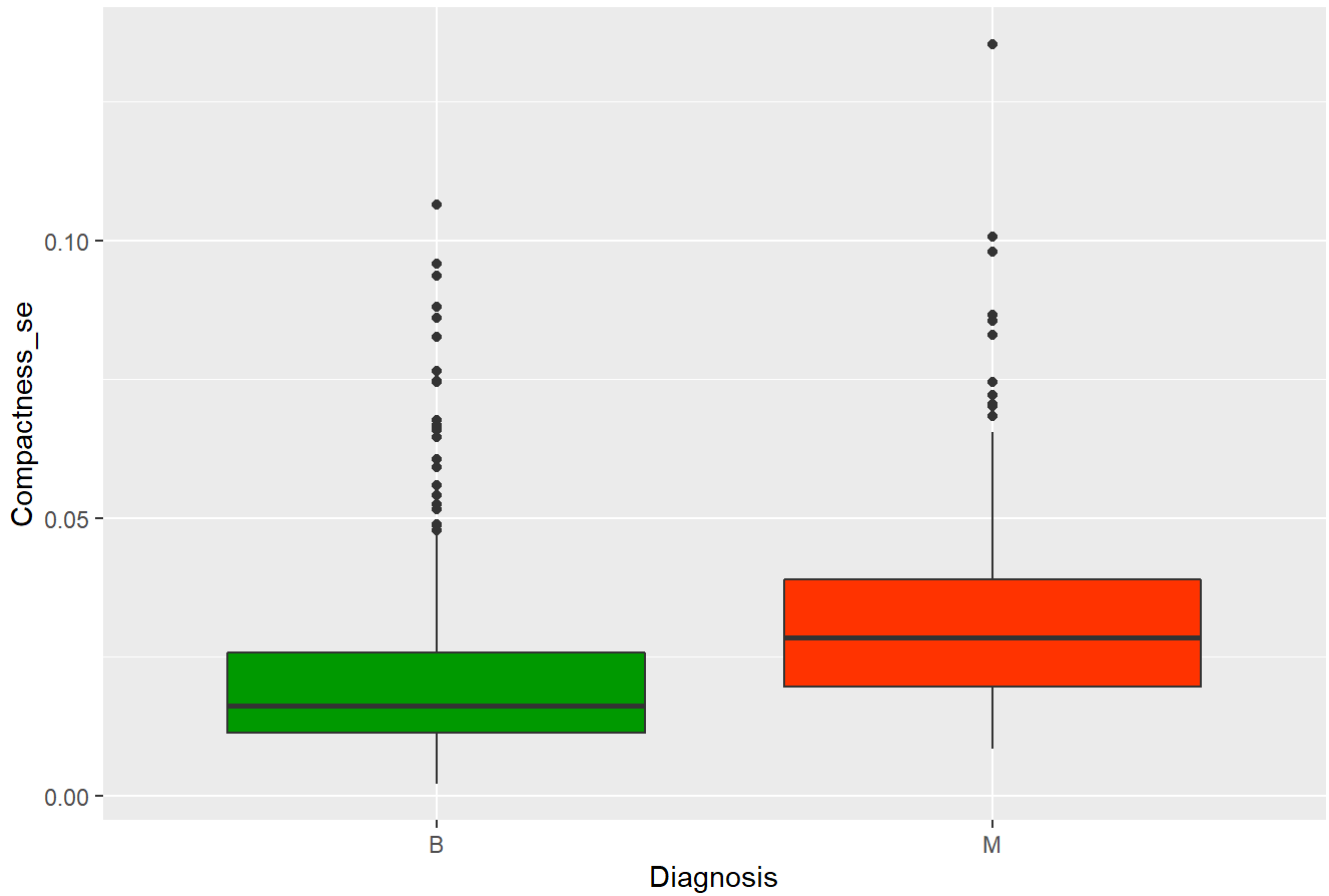


```
##  
## $Smoothness_se
```



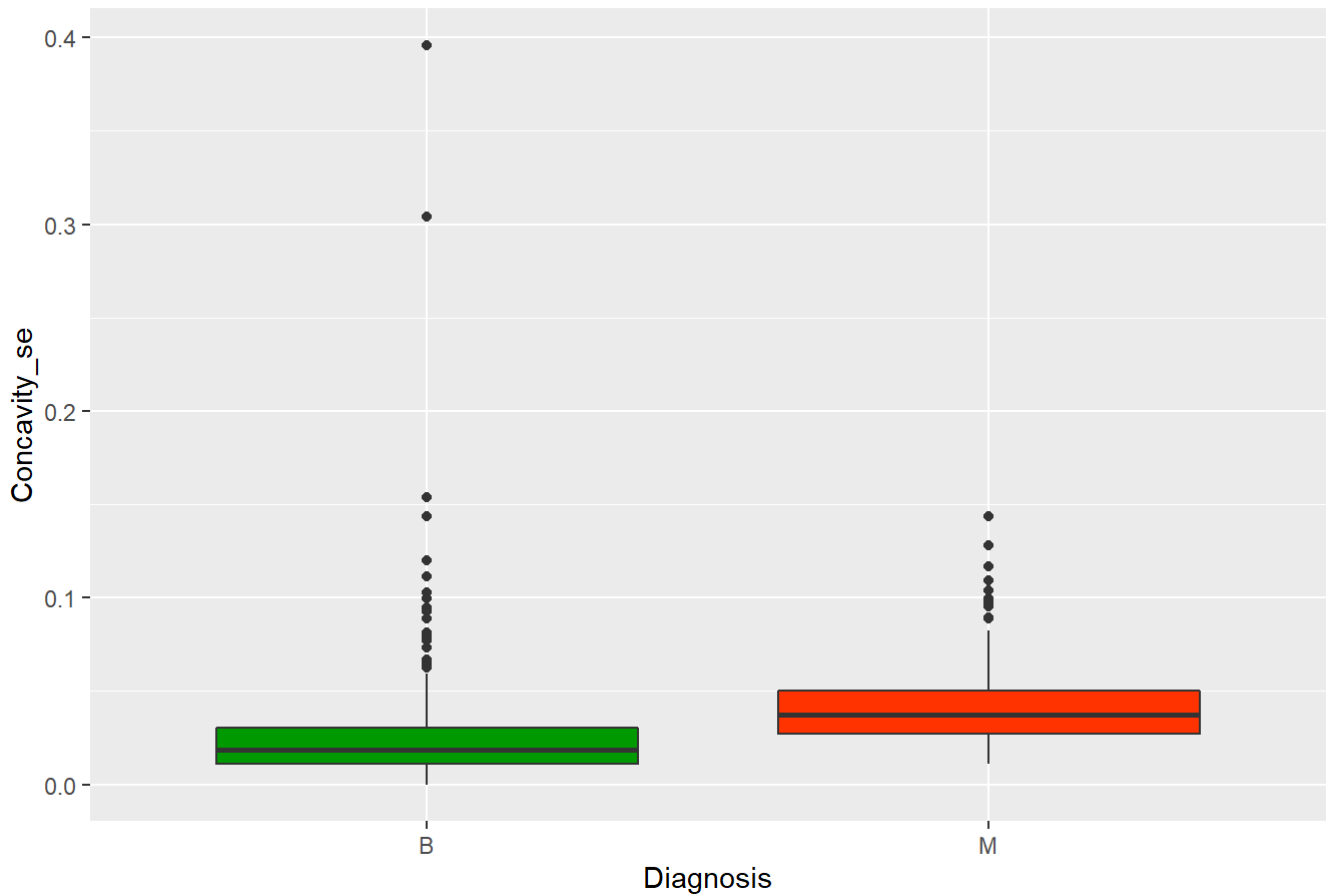
```
##  
## $Compactness_se
```

Diagnósticos por Compactness_se



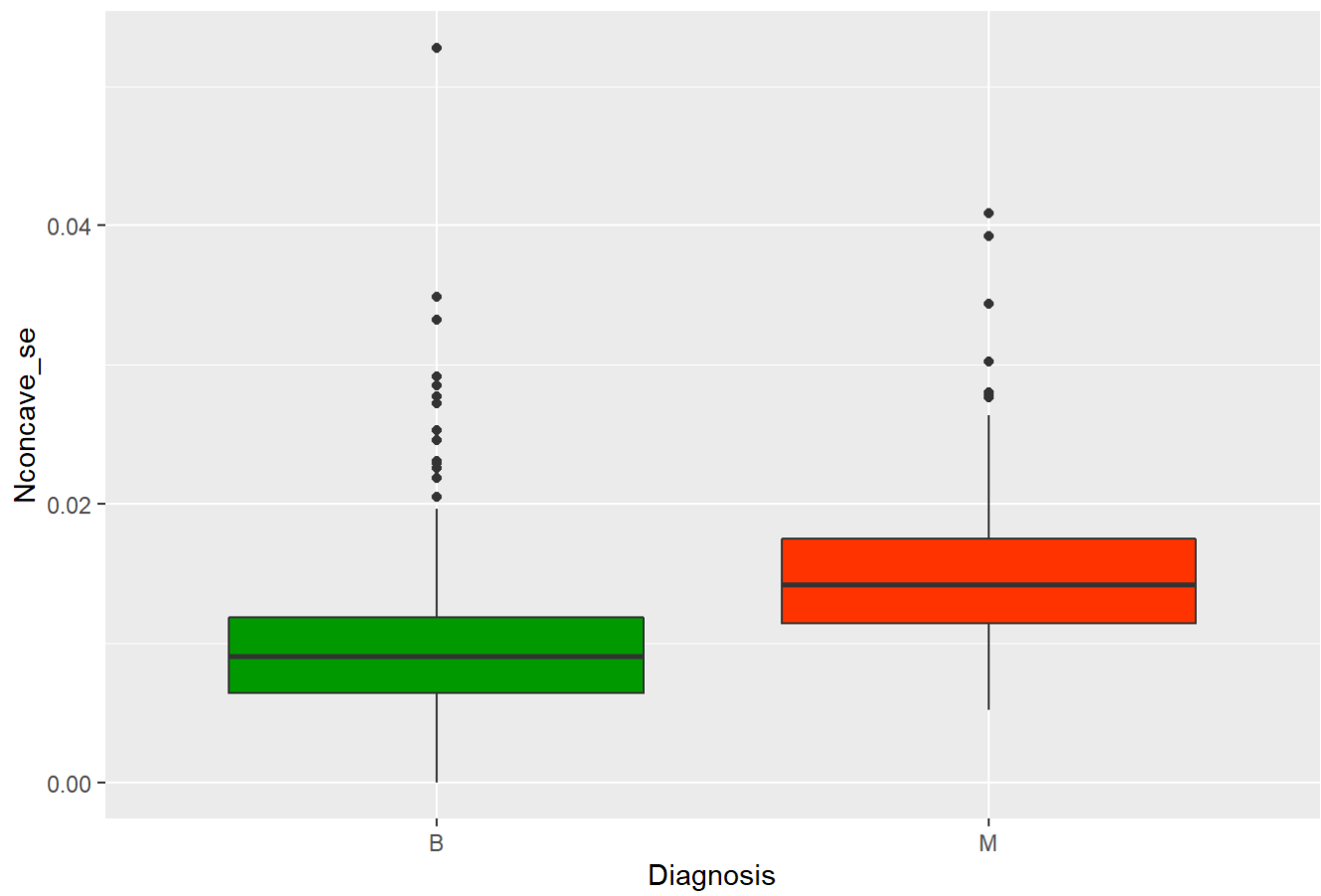
\$Concavity_se

Diagnósticos por Concavity_se



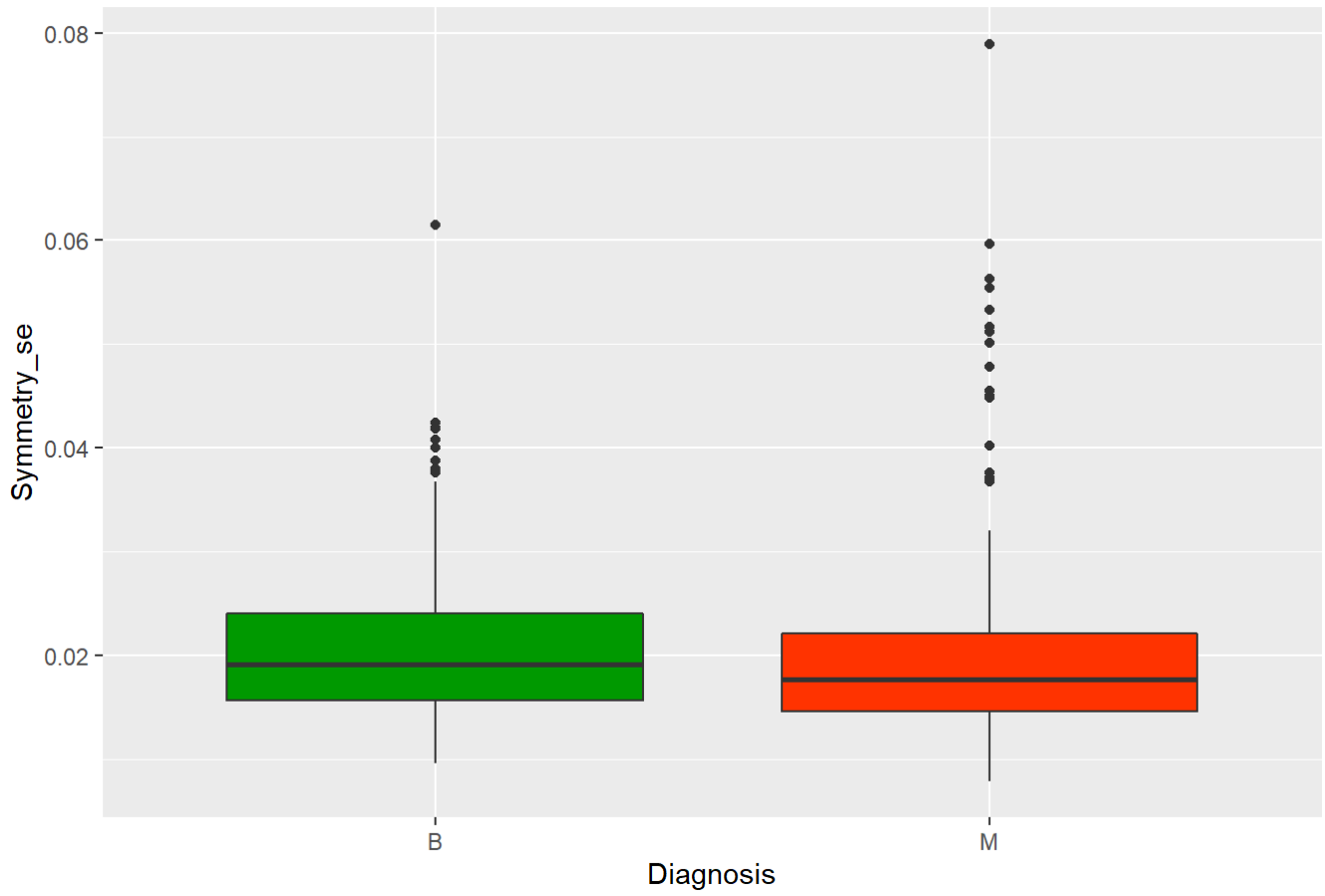
```
##  
## $Nconcave_se
```

Diagnósticos por Nconcave_se



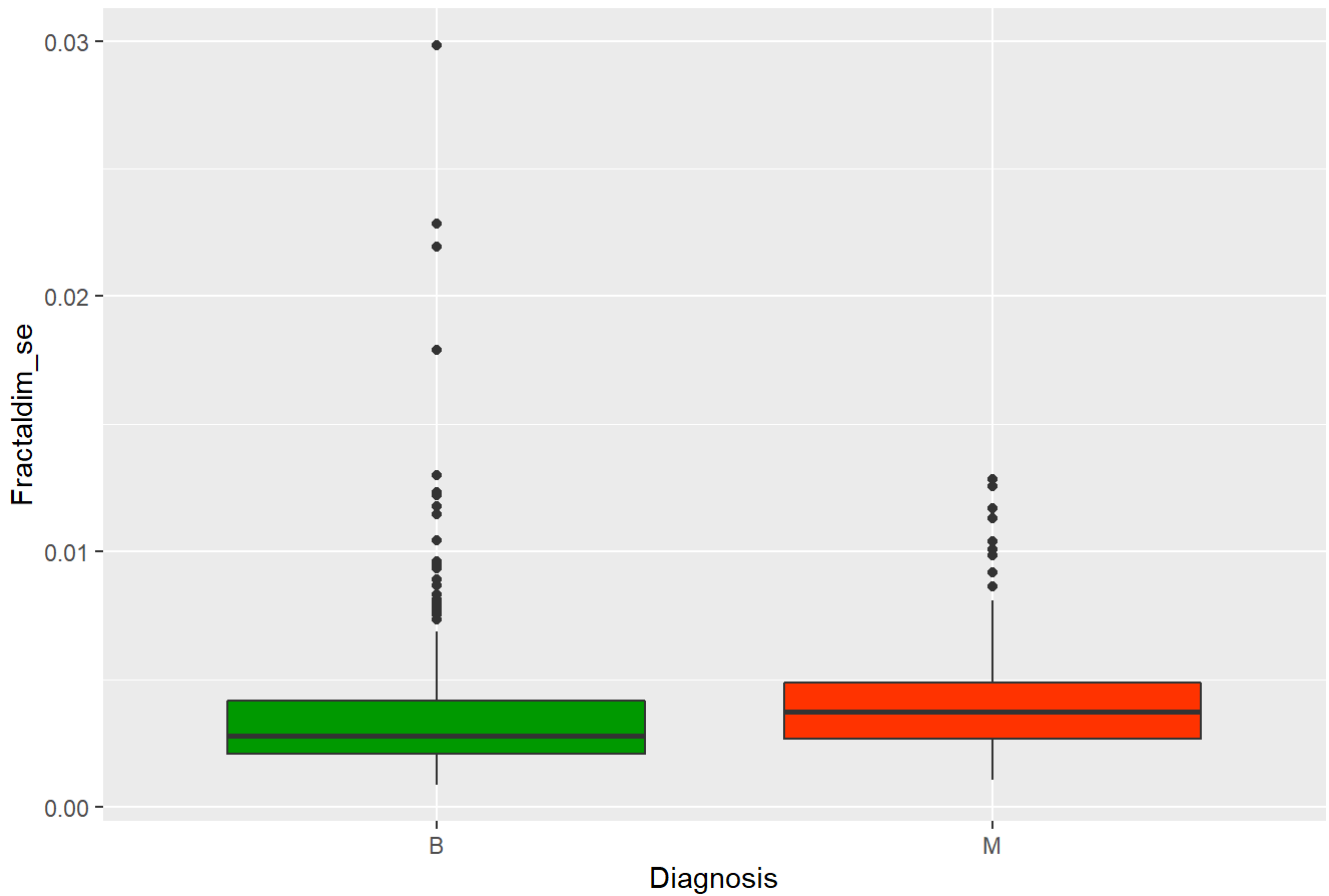
```
##  
## $Symmetry_se
```


Diagnósticos por Symmetry_se



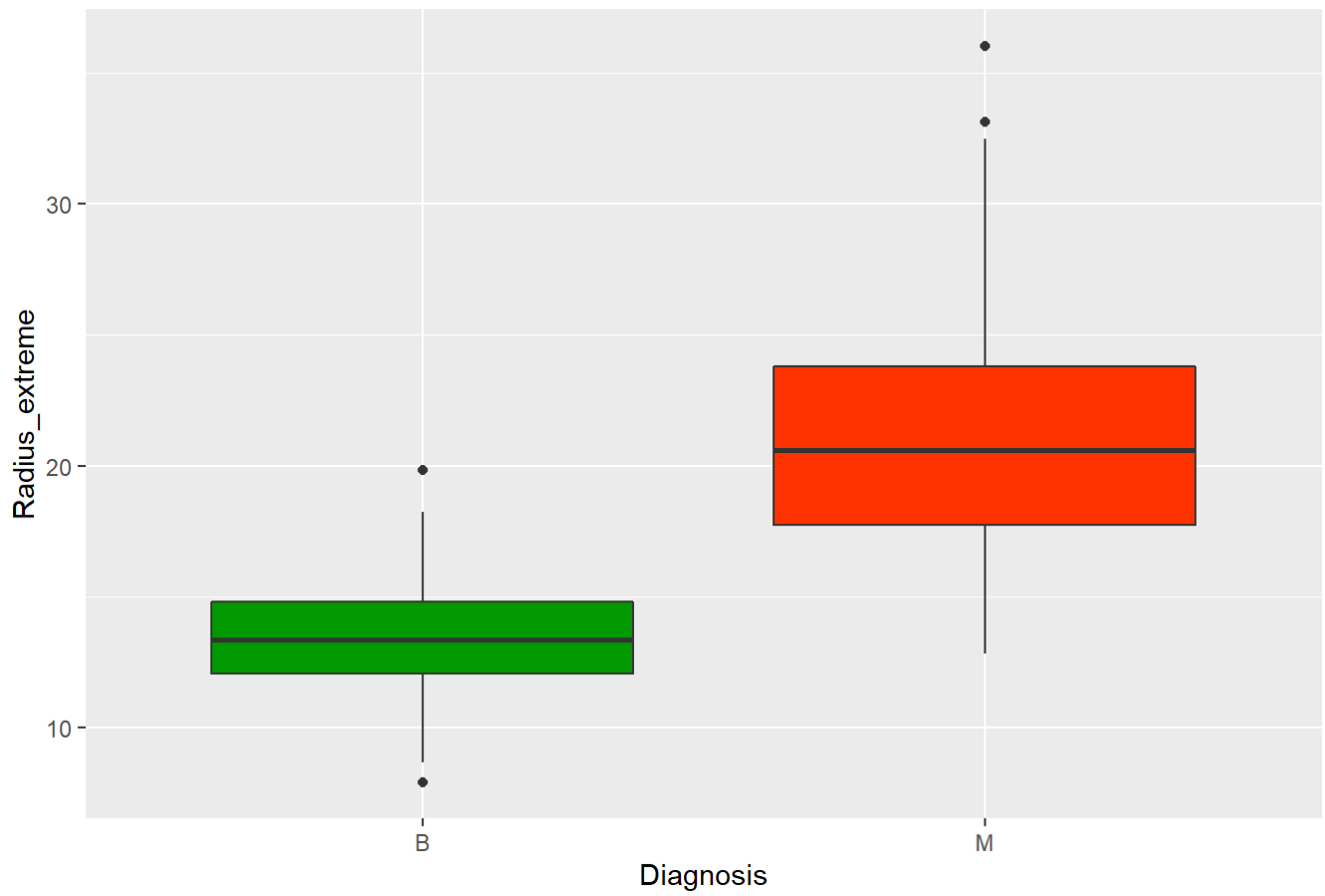
```
##  
## $Fractaldim_se
```

Diagnósticos por Fractaldim_se



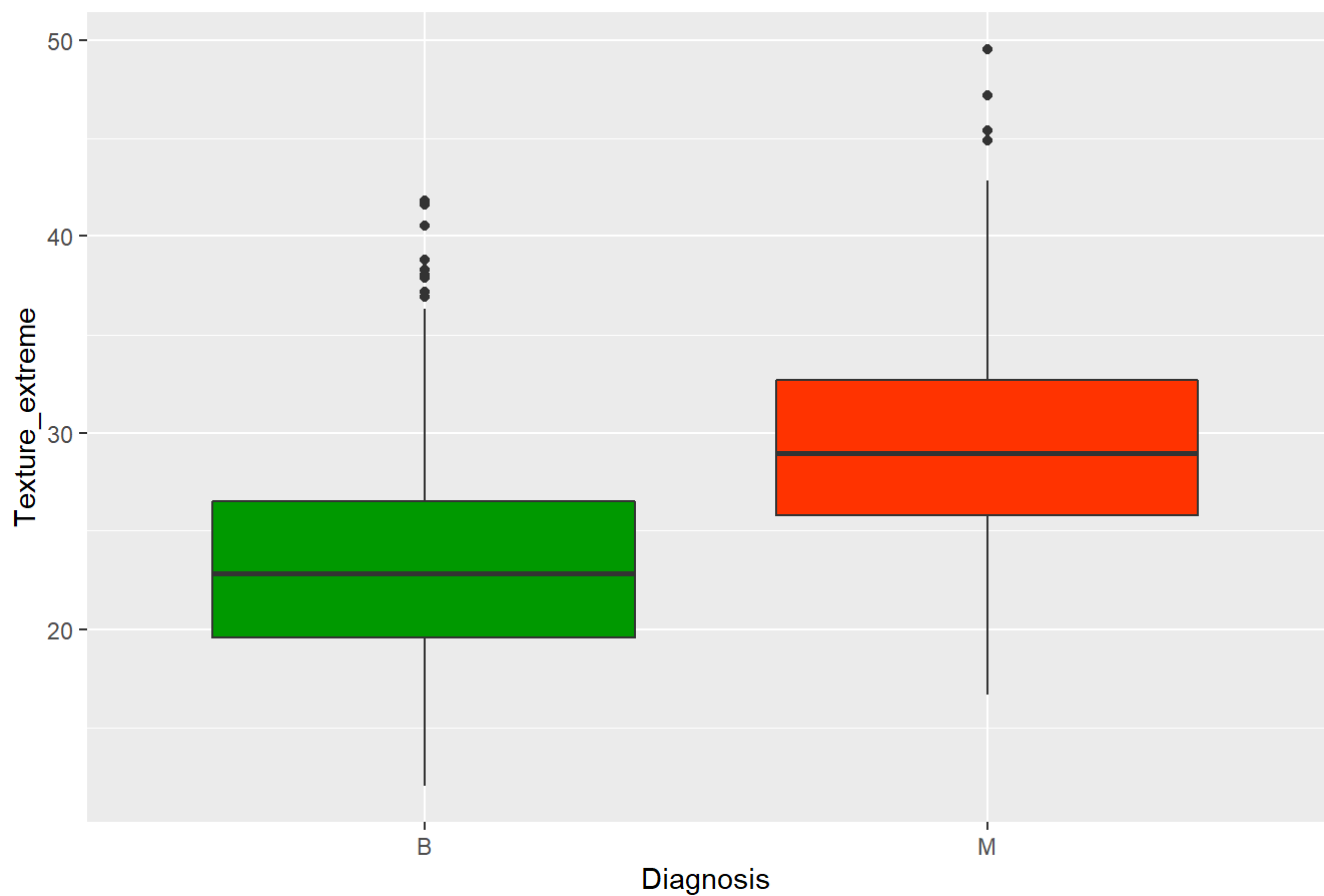
```
##  
## $Radius_extreme
```

Diagnósticos por Radius_extreme



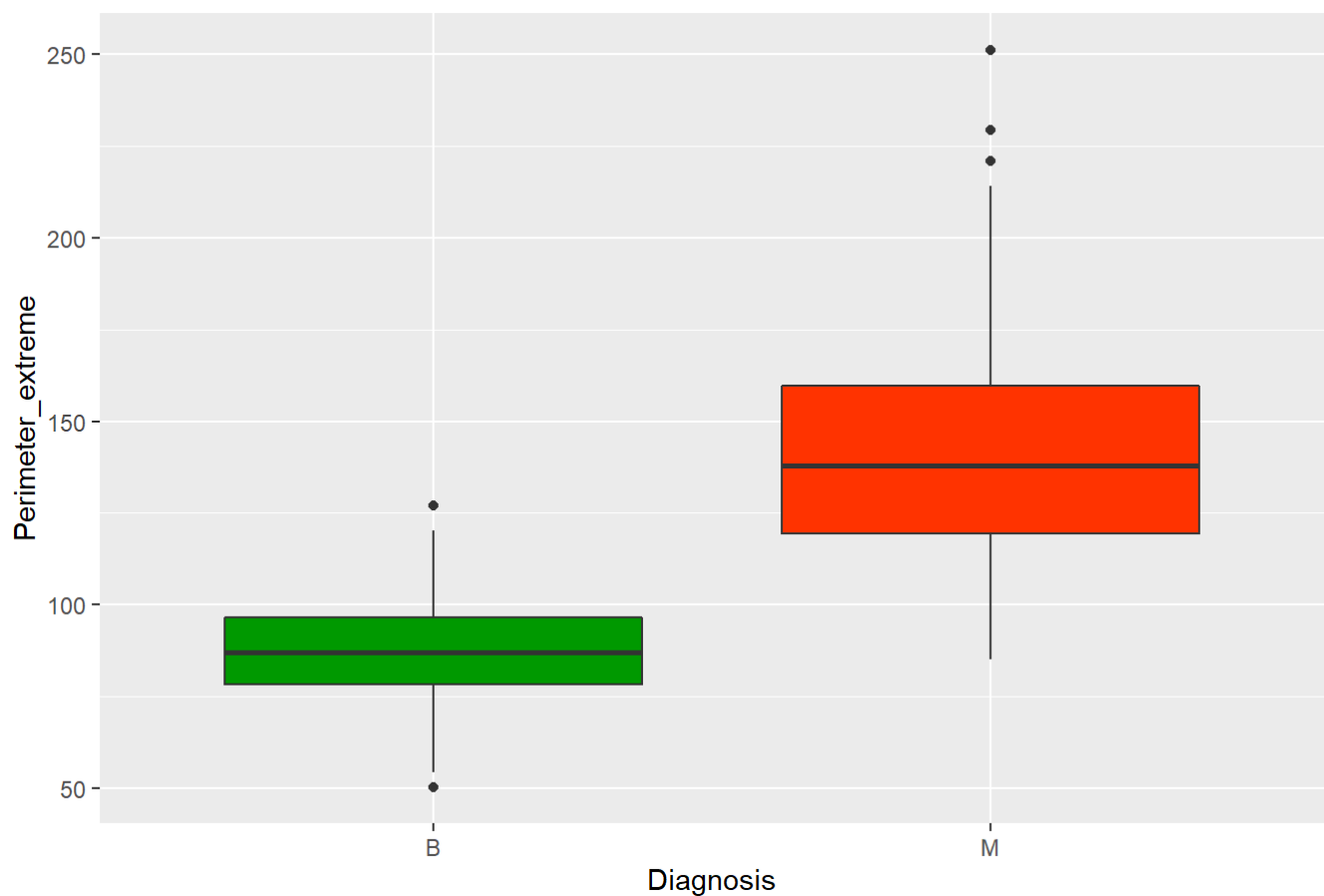
```
##  
## $Texture_extreme
```

Diagnósticos por Texture_extreme

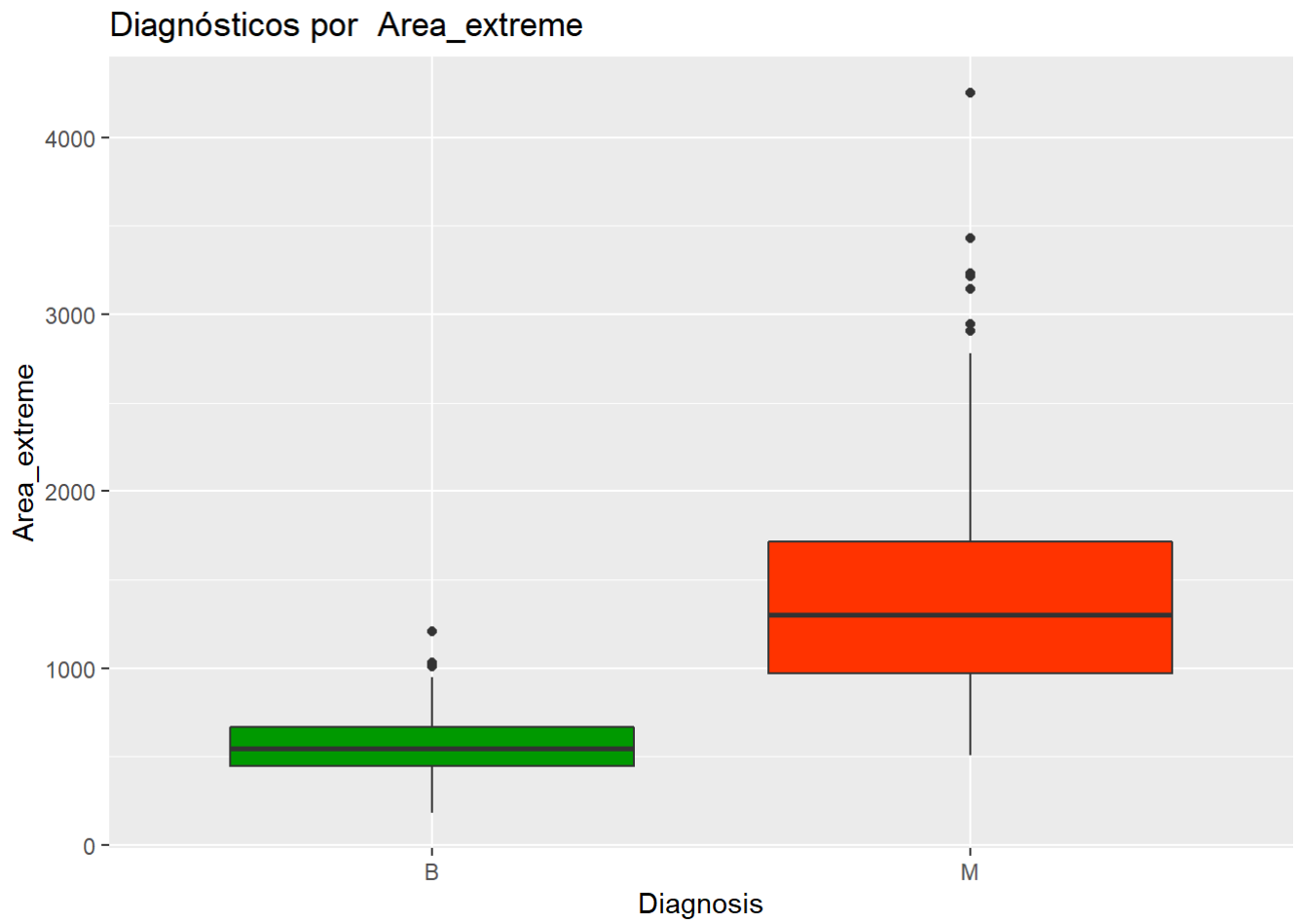


\$Perimeter_extreme

Diagnósticos por Perimeter_extreme

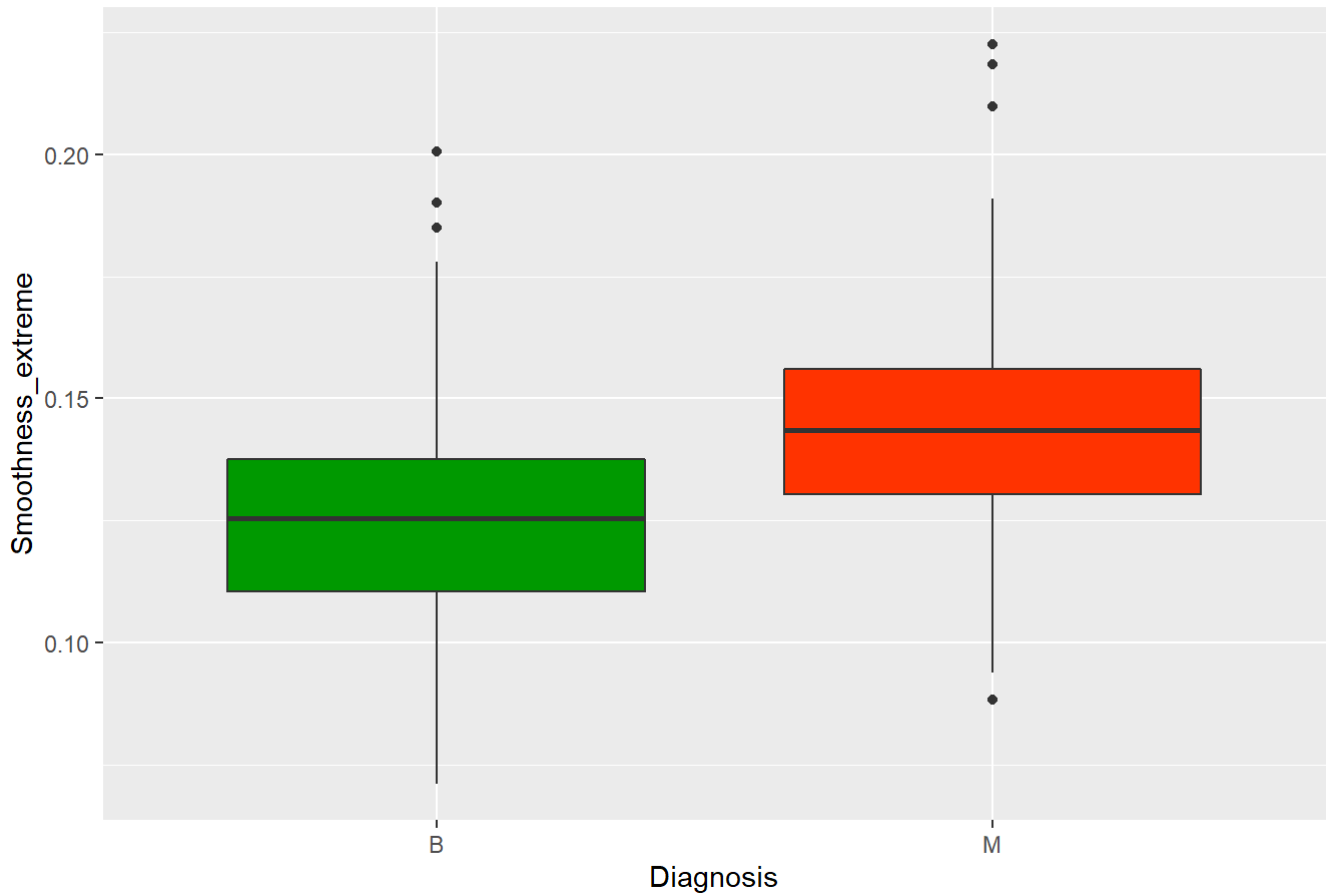


```
##  
## $Area_extreme
```



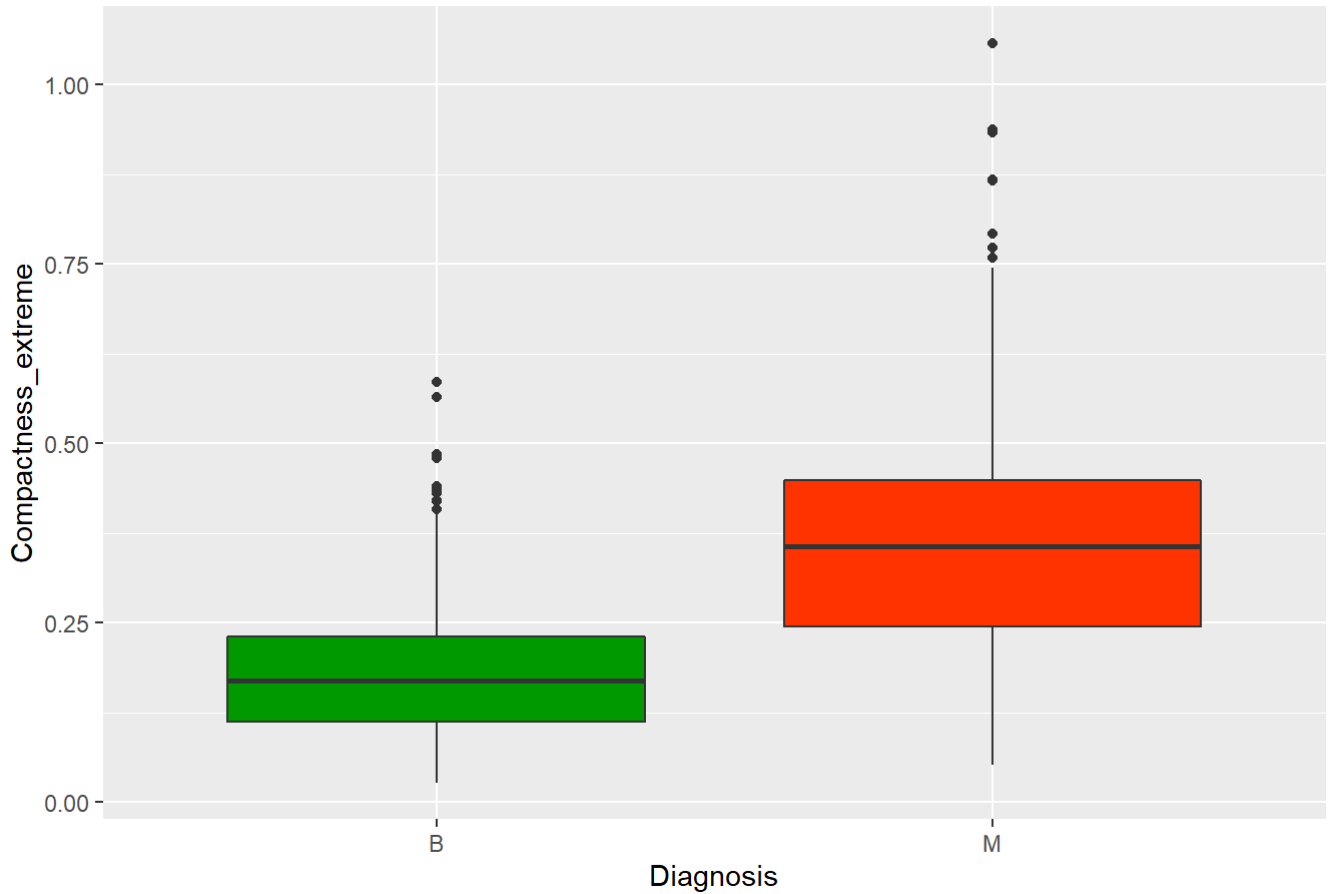
```
##  
## $Smoothness_extreme
```

Diagnósticos por Smoothness_extreme

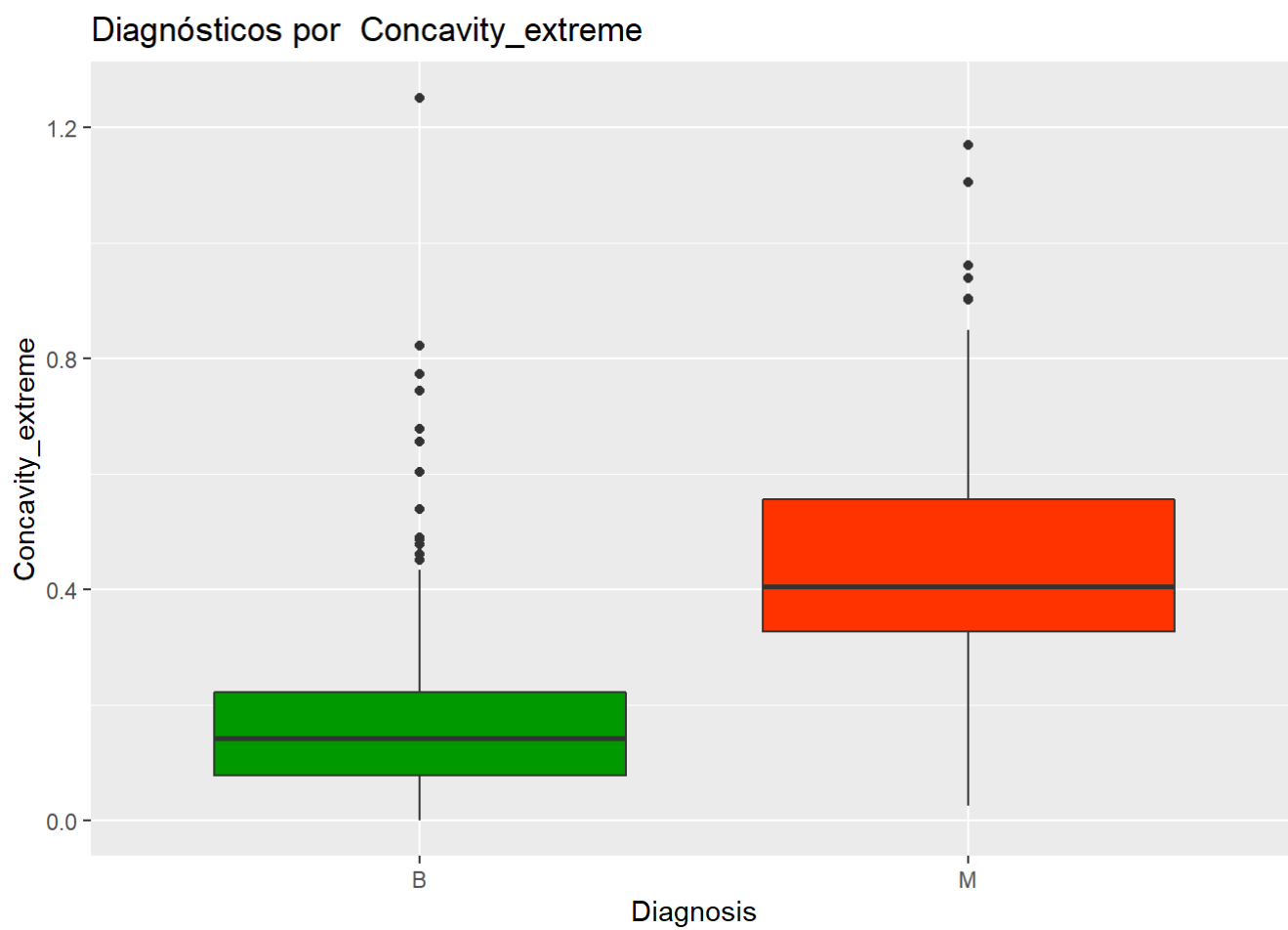


```
##  
## $Compactness_extreme
```

Diagnósticos por Compactness_extreme

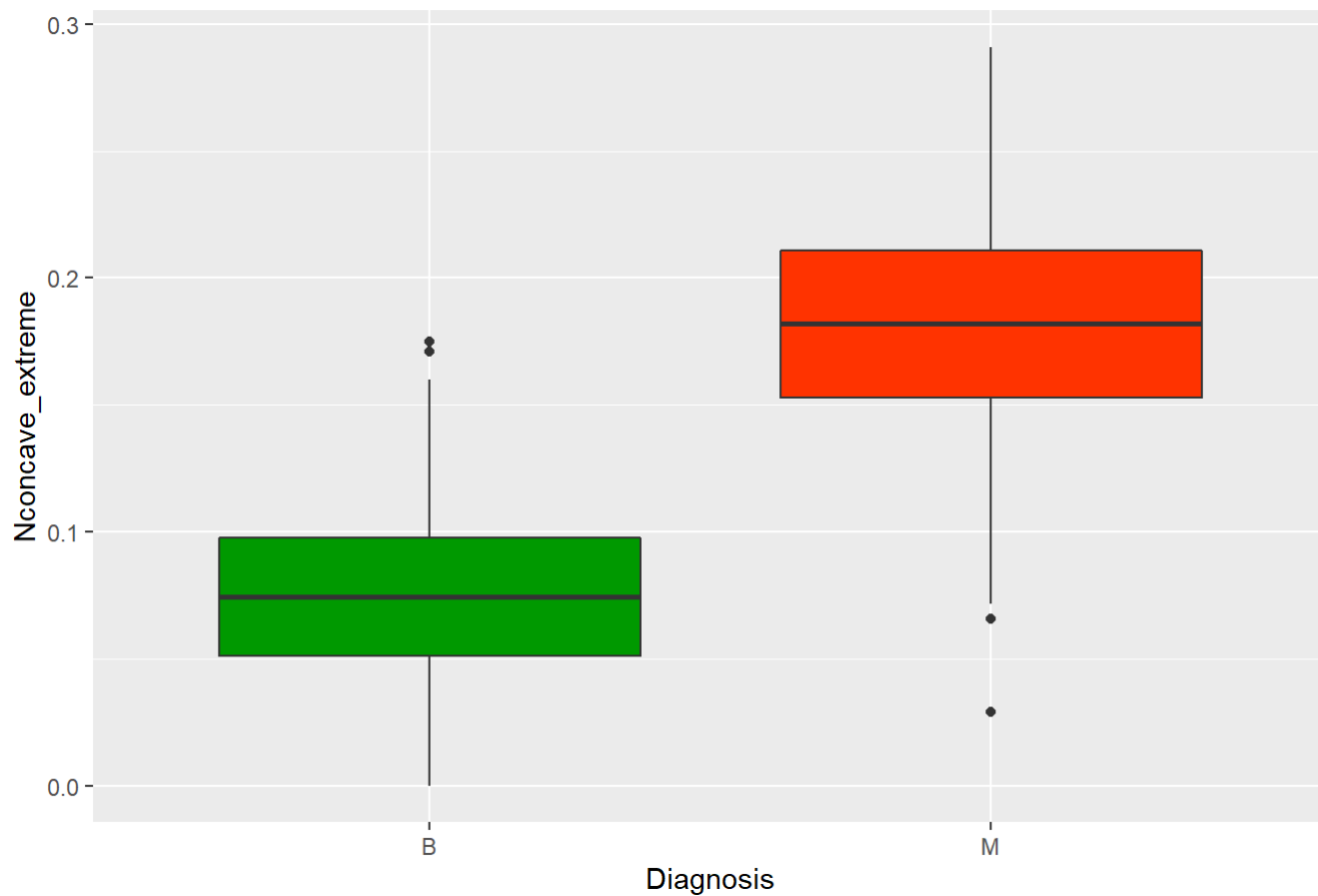


```
##  
## $Concavity_extreme
```



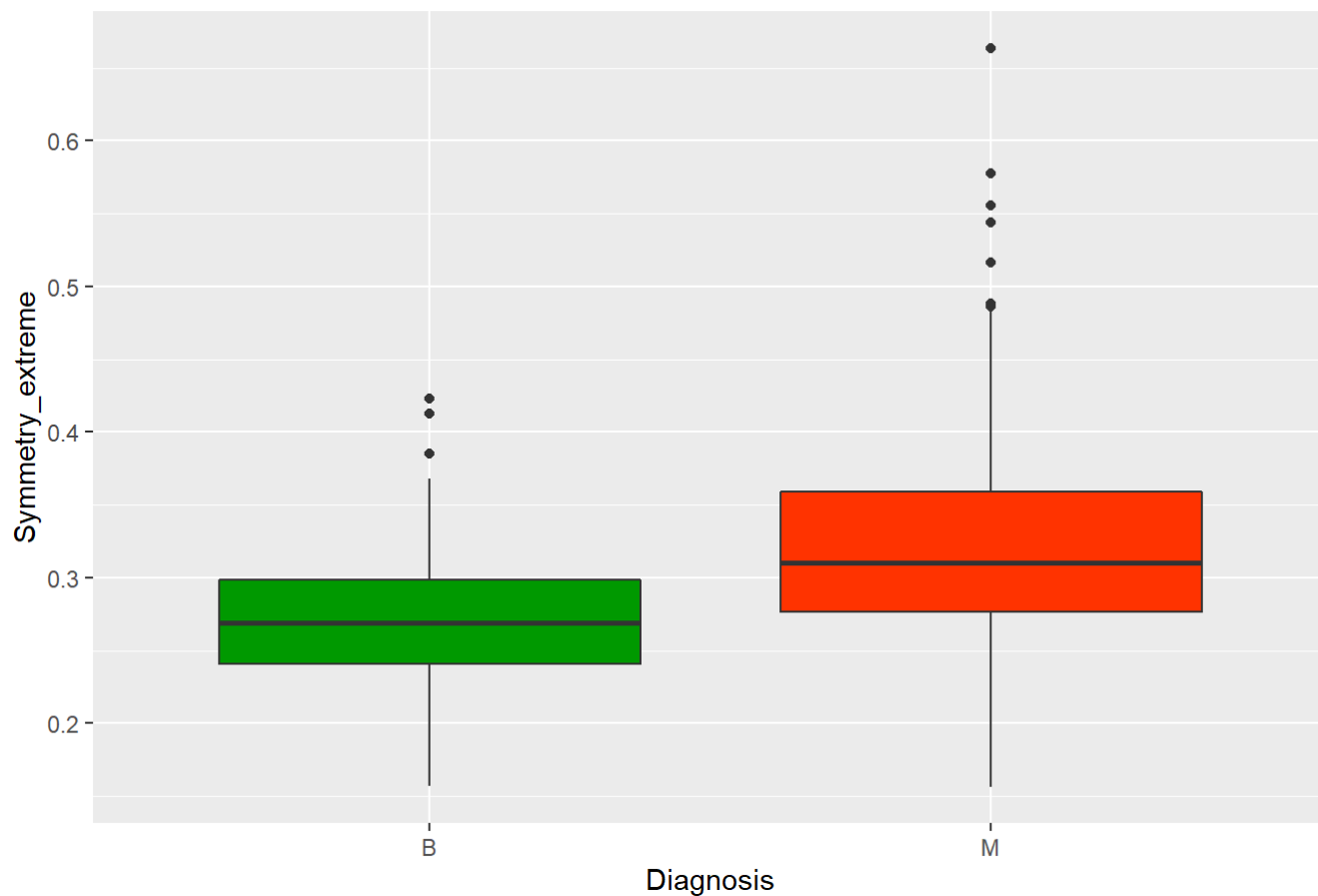
```
##  
## $Nconcave_extreme
```

Diagnósticos por Nconcave_extreme

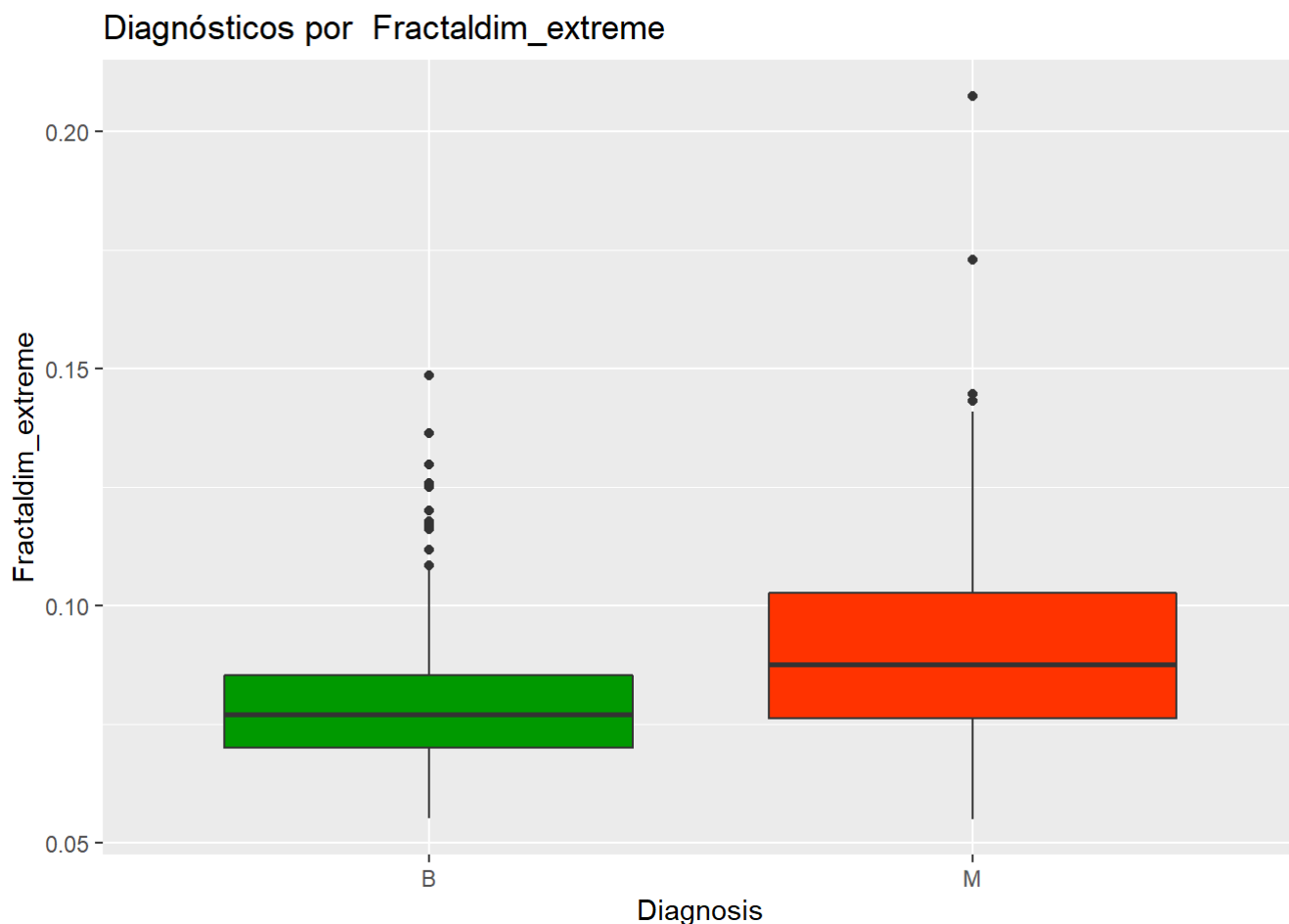


\$Symmetry_extreme

Diagnósticos por Symmetry_extreme



```
##  
## $Fractaldim_extreme
```



Separamos a nossa base de dados em 70% para treino e 30% para teste

```
library(caret)
```

```
## Loading required package: lattice
```

```
#Criando matriz com linhas dos dados de treino - 70%  
set.seed(1)  
filtro <- createDataPartition(y=BD$Diagnosis, p=0.7, list=FALSE)  
  
#dados de treino e teste  
treino <- BD[filtro,]  
teste <- BD[-filtro,]
```

Criamos um modelo de regressão logística para prevermos a variavel diagnostico com base em todas as outras variaveis, neste caso para validarmos o modelo usaremos a validação cruzada de 5 niveis, onde serão separadas 5 grupos de dados para que possamos garantir uma eficácia mínima do modelo.

```
#Criando o modelo  
set.seed(1)  
modelo <- train(Diagnosis ~ ., data=treino, method="glmnet", tuneLength=4, trControl = trainControl(method="cv", number = 5))
```


Vemos a seguir que após serem testados 5 grupos diferentes de dados de treino, possuímos uma acuracia de no minimo 97,46% e uma acurácia maxima de 98,75%

```
#Acuracia de cada uma das 5 separacoes de dados
modelo$resample$Accuracy
```

```
## [1] 0.9750000 0.9875000 0.9750000 0.9746835 0.9875000
```

```
#Precisao no modelo de treino
mean(modelo$resample$Accuracy)
```

```
## [1] 0.9799367
```

Realizando a média dos 5 grupos, temos a média de acurácia que o modelo entrega nos dados de treino.

```
#Precisao no modelo de treino
mean(modelo$resample$Accuracy)
```

```
## [1] 0.9799367
```

Assim, podemos aplicar o modelo nos dados de teste.

```
#Prevendo dados no modelo de teste
Prev <- predict(modelo, teste)
head(data.frame(teste$Diagnosis, Prev))
```

```
##  teste.Diagnosis Prev
## 1                M   M
## 2                M   M
## 3                M   M
## 4                M   M
## 5                M   M
## 6                M   M
```

Visualizando as 6 primeiras linhas da base de teste com seus dados previstos.

```
head(data.frame(teste$Diagnosis, Prev))
```

```
##  teste.Diagnosis Prev
## 1                M   M
## 2                M   M
## 3                M   M
## 4                M   M
## 5                M   M
## 6                M   M
```

Podemos visualizar melhor o resultado através de uma matriz de confusão.

```
library(gmodels)
```

```
#Visualizando matriz de confusao
```

```
CrossTable(Prev, teste$Diagnosis, dnn = c("Previsto", "Real"), prop.chisq = FALSE, prop.t = F  
ALSE, prop.r = FALSE, prop.c = FALSE)
```

```
##  
##  
##      Cell Contents  
## |-----|  
## |                      N |  
## |-----|  
##  
##  
## Total Observations in Table:  170  
##  
##  
##      Previsto | Real  
##      Previsto |      B |      M | Row Total |  
## -----|-----|-----|-----|  
##      B |      106 |      4 |      110 |  
## -----|-----|-----|-----|  
##      M |      1 |      59 |      60 |  
## -----|-----|-----|-----|  
## Column Total |      107 |      63 |      170 |  
## -----|-----|-----|-----|  
##  
##
```

Podemos visualizar as demais estatísticas como sensibilidade e especificidade também.

```
#Caret
```

```
#Verificando acuracia
```

```
confusionMatrix(Prev, teste$Diagnosis, dnn = c("Previsto", "Real"))
```

```
## Confusion Matrix and Statistics
##
##           Real
## Previsto   B    M
##           B 106   4
##           M   1  59
##
##           Accuracy : 0.9706
##           95% CI : (0.9327, 0.9904)
##           No Information Rate : 0.6294
##           P-Value [Acc > NIR] : <2e-16
##
##           Kappa : 0.9363
##
##           McNemar's Test P-Value : 0.3711
##
##           Sensitivity : 0.9907
##           Specificity : 0.9365
##           Pos Pred Value : 0.9636
##           Neg Pred Value : 0.9833
##           Prevalence : 0.6294
##           Detection Rate : 0.6235
##           Detection Prevalence : 0.6471
##           Balanced Accuracy : 0.9636
##
##           'Positive' Class : B
##
```

Neste caso, visualizamos um ponto de atenção, nosso modelo classificou 4 vezes tumores malignos como benignos, o que seria um grande equivoco para a área da saúde, não queremos que nenhum paciente seja informado que possui um cancer benigno quando na verdade possui um cancer maligno, entao trabalharemos para melhorar esse ponto mais adiante.

Usaremos a curva ROC para visualizarmos como se comportaram as classificações do modelo, bem como as melhores combinações de sensibilidade e especificidade.

Para isso, armazenamos as probabilidades de cada registro possuir cancer benigno ou maligno e arredondamos para 2 casas decimais.

```
#Calculando Probabilidades de ser benigno ou maligno
PrevProb <- predict(modelo, teste, type="prob")
#Visualizando probabilidades
head(round(PrevProb, 2))
```

```
##      B    M
## 1 0.00 1.00
## 2 0.00 1.00
## 3 0.00 1.00
## 4 0.15 0.85
## 5 0.00 1.00
## 6 0.00 1.00
```

Nota-se que a segunda coluna é apenas um complemento da probabilidade da coluna anterior, assim precisamos de apenas uma destas colunas para fazermos as substituições, assim deletamos uma das colunas.

```
#Precisa-se armazenar apenas uma das colunas de vetores, neste caso as probabilidades dos benignos  
PrevProb <- PrevProb$B
```

Usaremos a seguir a curva ROC para visualizarmos as melhores combinações de sensibilidade e especificidade que entregam a melhor acurácia do modelo.

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##  
## Attaching package: 'pROC'
```

```
## The following object is masked from 'package:gmodels':  
##  
##      ci
```

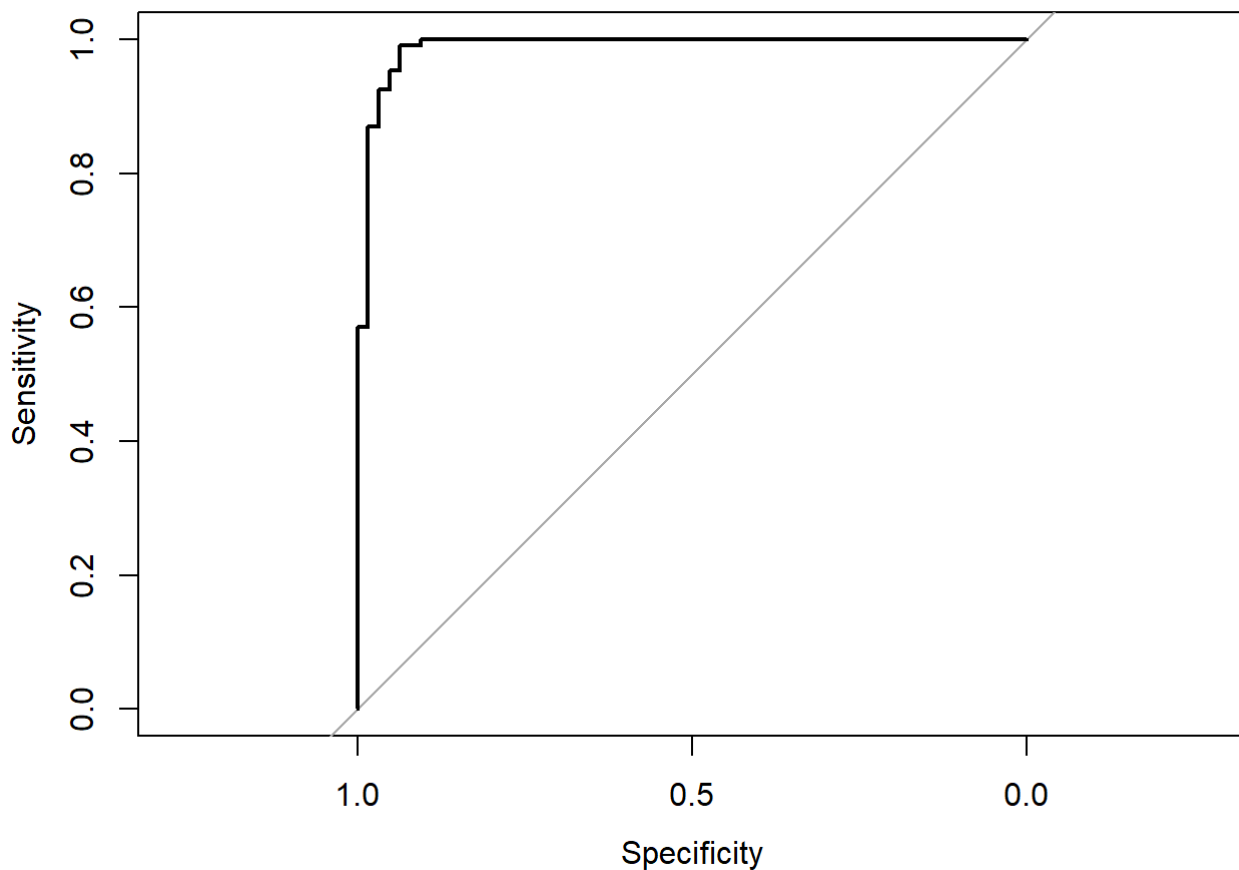
```
## The following objects are masked from 'package:stats':  
##  
##      cov, smooth, var
```

```
#Curva ROC  
#Calculando os valores com base nos dados de teste em função da probabilidade  
ROC <- roc(teste$Diagnosis ~ PrevProb, levels= c("M", "B"))
```

```
## Setting direction: controls < cases
```

Visualizando a curva.

```
plot(ROC)
```



Visualizando a área abaixo da curva.

```
ROC$auc
```

```
## Area under the curve: 0.9889
```

Neste caso, a área abaixo da curva é de 0.9889 ou seja muito próxima a 1, o que torna as classificações do modelo aceitáveis.

Como havíamos notado antes, precisamos corrigir ou amenizar a quantidade de casos malignos que foram classificados como benignos neste modelo, para isso exploraremos as combinações de sensibilidade e especificidade, afim de encontrarmos uma solução que diminua o efeito de uma pessoa com cancer maligno receber um diagnostico de cancer benigno. Para isso criamos uma tabela com as combinações de sensibilidade, especificidade e thresholds.

```
head(round(data.frame(ROC$sensitivities, ROC$specificities, ROC$thresholds),2))
```

```
## ROC.sensitivities ROC.specificities ROC.thresholds
## 1                1                0.00          -Inf
## 2                1                0.05           0
## 3                1                0.06           0
## 4                1                0.08           0
## 5                1                0.10           0
## 6                1                0.11           0
```

Ordenando por especificidade conseguimos visualizar a combinacao que zeraria os falsos positivos, o que estaria presente na linha 111, porem o modelo perderia muita sensibilidade ficando em torno de apenas 56%.

```
#Ordenando por especificidade conseguimos visualizar a combinacao que zeraria os falsos positivos,  
#porem o modelo perderia muita sensibilidade ficando em torno de apenas 56%  
ROC$thresholds[111]
```

```
## [1] 0.9998995
```

Armazenamos o vetor de probabilidades em outra variavel, pois está será manipulada de forma manual.

```
#Armazenando o vetor de probabilidades em outra variavel  
PrevT <- PrevProb
```

Assim classificamos como benigno apenas os registros que possuem mais de 99,98% de chances de serem cancer benigno, ou seja, classificaremos como benigno apenas quando tivermos praticamente certeza que este registro será.

```
#Substituindo as probabilidades acima de 99,45% por "B" (tumores benignos)  
#Dessa forma, um tumor sera considerado benigno apenas se apresentar probabilidade superior a 99,45%  
PrevT[PrevT>ROC$thresholds[111]] <- "B"  
#Todas as probabilidades que nao foram identificadas como "B" (benigno) serao "M" (maligno)  
PrevT[PrevT != "B"] <- "M"  
#Convertendo para fator  
PrevT <- as.factor(PrevT)
```

Podemos visualizar a matriz de confusão, após estas manipulações.

```
#Resultado do modelo com threshold ajustado  
confusionMatrix(PrevT, teste$Diagnosis, dnn = c("Previsto", "Real"))
```

```
## Confusion Matrix and Statistics
##
##           Real
## Previsto  B  M
##           B 58  0
##           M 49 63
##
##           Accuracy : 0.7118
##           95% CI : (0.6374, 0.7785)
## No Information Rate : 0.6294
## P-Value [Acc > NIR] : 0.01486
##
##           Kappa : 0.4673
##
## Mcnemar's Test P-Value : 7.025e-12
##
##           Sensitivity : 0.5421
##           Specificity : 1.0000
## Pos Pred Value : 1.0000
## Neg Pred Value : 0.5625
## Prevalence : 0.6294
## Detection Rate : 0.3412
## Detection Prevalence : 0.3412
## Balanced Accuracy : 0.7710
##
## 'Positive' Class : B
##
```

Nota-se que a precisao do modelo caiu 24%, porem reduzimos ao máximo o numero de acusações de cancer malignos que poderiam ser classificados como benignos. Neste caso, optaremos por realizar 1 acusação falsa de cancer benigno, em troca de um modelo com maior acurácia.

Voltamos a visualizar a tabela com as combinações de especificidade e sensibilidade, optamos desta vez por uma sensibilidade de no minimo 89% e selecionamos a especificidade de melhor combinação.

```
#Filtrando dados com sensibilidade acima de 0.89 e a maior combinacao possivel de especificidade
head(round(data.frame(ROC$sensitivities, ROC$specificities, ROC$thresholds),2))
```

```
## ROC.sensitivities ROC.specificities ROC.thresholds
## 1                1                0.00          -Inf
## 2                1                0.05           0
## 3                1                0.06           0
## 4                1                0.08           0
## 5                1                0.10           0
## 6                1                0.11           0
```

Esta se encontrava na linha 75, assim substituímos novamente na tabela de probabilidades.

```
#Encontramos o vetor 75, com sensibilidade de 0.89 e especificidade de 0.98
#Substituindo no modelo temos
ROC$thresholds[75]
```

```
## [1] 0.9869348
```

```

PrevT <- PrevProb
PrevT[PrevT>ROC$thresholds[75]] <- "B"
PrevT[PrevT != "B"] <- "M"
PrevT <- as.factor(PrevT)

```

E podemos ver o resultado deste novo modelo.

```

#Resultado do modelo com threshold ajustado
confusionMatrix(PrevT, teste$Diagnosis, dnn = c("Previsto", "Real"))

```

```

## Confusion Matrix and Statistics
##
##           Real
## Previsto  B  M
##           B 93  1
##           M 14 62
##
##               Accuracy : 0.9118
##               95% CI : (0.8586, 0.9498)
##       No Information Rate : 0.6294
##       P-Value [Acc > NIR] : < 2.2e-16
##
##               Kappa : 0.8186
##
##  Mcnemar's Test P-Value : 0.001946
##
##       Sensitivity : 0.8692
##       Specificity : 0.9841
##       Pos Pred Value : 0.9894
##       Neg Pred Value : 0.8158
##       Prevalence : 0.6294
##       Detection Rate : 0.5471
##       Detection Prevalence : 0.5529
##       Balanced Accuracy : 0.9266
##
##       'Positive' Class : B
##

```

Conseguimos melhorar a precisão do modelo em 20%, aceitando 1 caso de falso positivo para cancer benigno, aceitaremos este modelo com a acuracia de 91%.