breast-cancer

Shinjini

10 December 2017

import dataset

```
data=read.csv("C://Users//Administrator//Desktop//KAGGLE//breastcancer//data.csv")
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
glimpse(data)
```

```
## Observations: 569
## Variables: 33
## $ id                      <int> 842302, 842517, 84300903, 84348301, 84...
## $ diagnosis               <fctr> M, M, M, M, M, M, M, M, M, M, M, M, M...
## $ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290...
## $ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15....
## $ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10,...
## $ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0,...
## $ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0....
## $ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0....
## $ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0....
## $ concave.points_mean     <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0....
## $ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809...
## $ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0....
## $ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572...
## $ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813...
## $ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.2...
## $ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27...
## $ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110...
## $ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580...
## $ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0....
## $ concave.points_se       <dbl> 0.015870, 0.013400, 0.020580, 0.018670...
## $ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0....
## $ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208...
## $ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15....
## $ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23....
## $ perimeter_worst         <dbl> 184.60, 158.80, 152.50, 98.87, 152.20,...
## $ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0,...
## $ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374...
## $ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050...
## $ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0....
## $ concave.points_worst    <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0....
## $ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364...
## $ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0....
## $ X                       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
```

Before making anything like feature selection,feature extraction and classification, firstly we start with basic data analysis. Lets look at features of data.

```
head(data)
```

```
##         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
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
## 4         0.14250          0.28390         0.2414             0.10520
## 5         0.10030          0.13280         0.1980             0.10430
## 6         0.12780          0.17000         0.1578             0.08089
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.2419                0.07871    1.0950     0.9053        8.589
## 2        0.1812                0.05667    0.5435     0.7339        3.398
## 3        0.2069                0.05999    0.7456     0.7869        4.585
## 4        0.2597                0.09744    0.4956     1.1560        3.445
## 5        0.1809                0.05883    0.7572     0.7813        5.438
## 6        0.2087                0.07613    0.3345     0.8902        2.217
##   area_se smoothness_se compactness_se concavity_se concave.points_se
## 1  153.40      0.006399        0.04904      0.05373           0.01587
## 2   74.08      0.005225        0.01308      0.01860           0.01340
## 3   94.03      0.006150        0.04006      0.03832           0.02058
## 4   27.23      0.009110        0.07458      0.05661           0.01867
## 5   94.44      0.011490        0.02461      0.05688           0.01885
## 6   27.19      0.007510        0.03345      0.03672           0.01137
##   symmetry_se fractal_dimension_se radius_worst texture_worst
## 1     0.03003             0.006193        25.38         17.33
## 2     0.01389             0.003532        24.99         23.41
## 3     0.02250             0.004571        23.57         25.53
## 4     0.05963             0.009208        14.91         26.50
## 5     0.01756             0.005115        22.54         16.67
## 6     0.02165             0.005082        15.47         23.75
##   perimeter_worst area_worst smoothness_worst compactness_worst
## 1          184.60     2019.0           0.1622            0.6656
## 2          158.80     1956.0           0.1238            0.1866
## 3          152.50     1709.0           0.1444            0.4245
## 4           98.87      567.7           0.2098            0.8663
## 5          152.20     1575.0           0.1374            0.2050
## 6          103.40      741.6           0.1791            0.5249
##   concavity_worst concave.points_worst symmetry_worst
## 1          0.7119               0.2654         0.4601
## 2          0.2416               0.1860         0.2750
## 3          0.4504               0.2430         0.3613
## 4          0.6869               0.2575         0.6638
## 5          0.4000               0.1625         0.2364
## 6          0.5355               0.1741         0.3985
##   fractal_dimension_worst  X
## 1                 0.11890 NA
## 2                 0.08902 NA
## 3                 0.08758 NA
## 4                 0.17300 NA
## 5                 0.07678 NA
## 6                 0.12440 NA
```

- There are 4 things that take my attention
- 1. There is an id that cannot be used for classificaiton
- 2. Diagnosis is our class label
- 3. Unnamed: 32 feature includes NaN so we do not need it.

feature names as a list

```
list= colnames(data)
print(list)
```

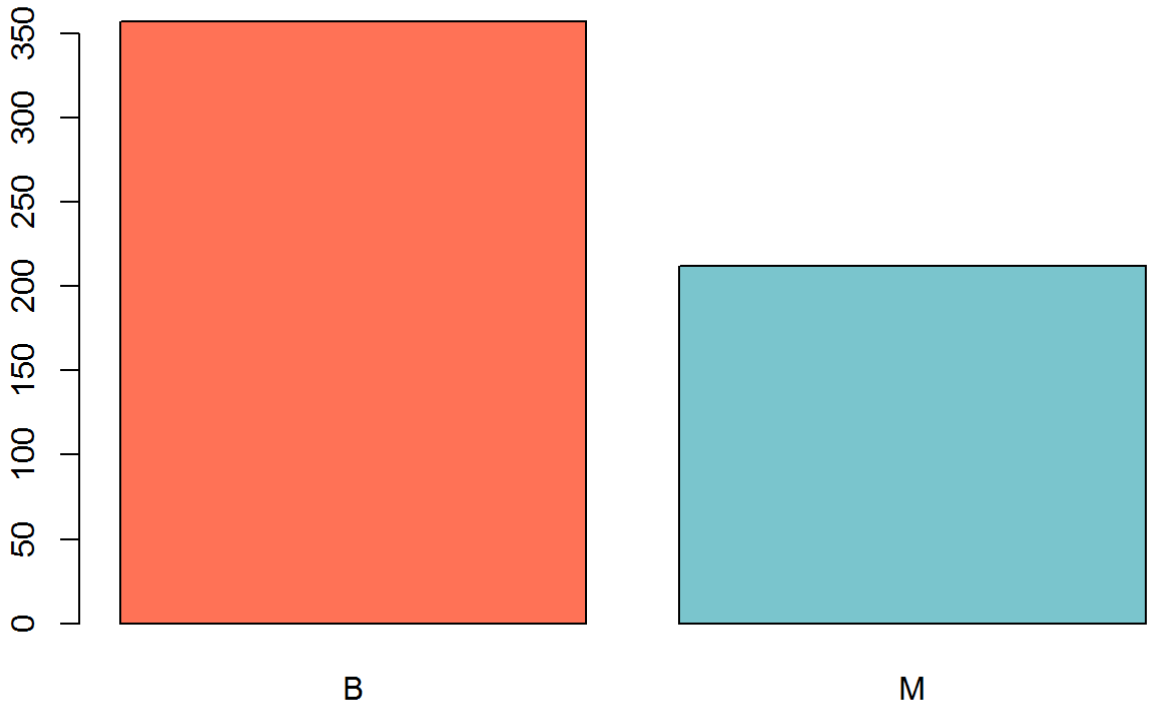
```
##  [1] "id"                     "diagnosis"
##  [3] "radius_mean"            "texture_mean"
##  [5] "perimeter_mean"         "area_mean"
##  [7] "smoothness_mean"        "compactness_mean"
##  [9] "concavity_mean"         "concave.points_mean"
## [11] "symmetry_mean"          "fractal_dimension_mean"
## [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] "X"
```

classify the data into feautures and labels

```
x= data[,c(-1,-2,-33)]
y= data[,2]
```

lets check the number of benign vs no of malignent

```
t=table(data$diagnosis)
barplot(t,col=c("coral1","cadetblue3"))
```



statistical summary of the data

```
library(fBasics)
```

```
## Warning: package 'fBasics' was built under R version 3.4.3
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
## Warning: package 'timeSeries' was built under R version 3.4.3
```

```
basicStats(x)
```

```
##             radius_mean texture_mean perimeter_mean    area_mean
## nobs         569.000000   569.000000     569.000000 5.690000e+02
## NAs            0.000000     0.000000       0.000000 0.000000e+00
## Minimum        6.981000     9.710000      43.790000 1.435000e+02
## Maximum       28.110000    39.280000     188.500000 2.501000e+03
## 1. Quartile   11.700000    16.170000      75.170000 4.203000e+02
## 3. Quartile   15.780000    21.800000     104.100000 7.827000e+02
## Mean          14.127292    19.289649      91.969033 6.548891e+02
## Median        13.370000    18.840000      86.240000 5.511000e+02
## Sum         8038.429000 10975.810000   52330.380000 3.726319e+05
## SE Mean        0.147736     0.180309       1.018666 1.475301e+01
## LCL Mean      13.837117    18.935495      89.968221 6.259120e+02
## UCL Mean      14.417467    19.643802      93.969846 6.838662e+02
## Variance      12.418920    18.498909     590.440480 1.238436e+05
## Stdev          3.524049     4.301036      24.298981 3.519141e+02
## Skewness       0.937417     0.647024       0.985433 1.637065e+00
## Kurtosis       0.814142     0.728007       0.939282 3.586549e+00
##             smoothness_mean compactness_mean concavity_mean
## nobs             569.000000       569.000000     569.000000
## NAs                0.000000         0.000000       0.000000
## Minimum            0.052630         0.019380       0.000000
## Maximum            0.163400         0.345400       0.426800
## 1. Quartile        0.086370         0.064920       0.029560
## 3. Quartile        0.105300         0.130400       0.130700
## Mean               0.096360         0.104341       0.088799
## Median             0.095870         0.092630       0.061540
## Sum               54.829000        59.370020      50.526811
## SE Mean            0.000590         0.002214       0.003342
## LCL Mean           0.095202         0.099992       0.082235
## UCL Mean           0.097518         0.108690       0.095364
## Variance           0.000198         0.002789       0.006355
## Stdev              0.014064         0.052813       0.079720
## Skewness           0.453921         1.183856       1.393801
## Kurtosis           0.824467         1.608897       1.953136
##             concave.points_mean symmetry_mean fractal_dimension_mean
## nobs                 569.000000    569.000000             569.000000
## NAs                    0.000000      0.000000               0.000000
## Minimum                0.000000      0.106000               0.049960
## Maximum                0.201200      0.304000               0.097440
## 1. Quartile            0.020310      0.161900               0.057700
## 3. Quartile            0.074000      0.195700               0.066120
## Mean                   0.048919      0.181162               0.062798
## Median                 0.033500      0.179200               0.061540
## Sum                   27.834994    103.081100              35.731840
## SE Mean                0.001627      0.001149               0.000296
## LCL Mean               0.045724      0.178905               0.062216
## UCL Mean               0.052114      0.183419               0.063379
## Variance               0.001506      0.000752               0.000050
## Stdev                  0.038803      0.027414               0.007060
## Skewness               1.165012      0.721788               1.297619
## Kurtosis               1.032469      1.251135               2.948055
##              radius_se texture_se perimeter_se      area_se smoothness_se
## nobs        569.000000 569.000000   569.000000   569.000000    569.000000
## NAs           0.000000   0.000000     0.000000     0.000000      0.000000
## Minimum       0.111500   0.360200     0.757000     6.802000      0.001713
## Maximum       2.873000   4.885000    21.980000   542.200000      0.031130
## 1. Quartile   0.232400   0.833900     1.606000    17.850000      0.005169
```

```
## 3. Quartile   0.478900   1.474000     3.357000    45.190000      0.008146
## Mean          0.405172   1.216853     2.866059    40.337079      0.007041
## Median        0.324200   1.108000     2.287000    24.530000      0.006380
## Sum         230.542900 692.389600  1630.787700 22951.798000      4.006317
## SE Mean       0.011626   0.023126     0.084761     1.907082      0.000126
## LCL Mean      0.382338   1.171430     2.699577    36.591285      0.006794
## UCL Mean      0.428006   1.262277     3.032542    44.082873      0.007288
## Variance      0.076902   0.304316     4.087896  2069.431583      0.000009
## Stdev         0.277313   0.551648     2.021855    45.491006      0.003003
## Skewness      3.072347   1.637773     3.425480     5.418500      2.302262
## Kurtosis     17.449095   5.262633    21.118775    48.585397     10.320592
##             compactness_se concavity_se concave.points_se symmetry_se
## nobs            569.000000   569.000000        569.000000  569.000000
## NAs               0.000000     0.000000          0.000000    0.000000
## Minimum           0.002252     0.000000          0.000000    0.007882
## Maximum           0.135400     0.396000          0.052790    0.078950
## 1. Quartile       0.013080     0.015090          0.007638    0.015160
## 3. Quartile       0.032450     0.042050          0.014710    0.023480
## Mean              0.025478     0.031894          0.011796    0.020542
## Median            0.020450     0.025890          0.010930    0.018730
## Sum              14.497061    18.147525          6.712002   11.688568
## SE Mean           0.000751     0.001265          0.000259    0.000347
## LCL Mean          0.024004     0.029408          0.011288    0.019862
## UCL Mean          0.026953     0.034379          0.012304    0.021223
## Variance          0.000321     0.000911          0.000038    0.000068
## Stdev             0.017908     0.030186          0.006170    0.008266
## Skewness          1.892203     5.083550          1.437070    2.183573
## Kurtosis          5.022692    48.241974          5.042496    7.778402
##             fractal_dimension_se radius_worst texture_worst
## nobs                  569.000000   569.000000    569.000000
## NAs                     0.000000     0.000000      0.000000
## Minimum                 0.000895     7.930000     12.020000
## Maximum                 0.029840    36.040000     49.540000
## 1. Quartile             0.002248    13.010000     21.080000
## 3. Quartile             0.004558    18.790000     29.720000
## Mean                    0.003795    16.269190     25.677223
## Median                  0.003187    14.970000     25.410000
## Sum                     2.159300  9257.169000  14610.340000
## SE Mean                 0.000111     0.202620      0.257665
## LCL Mean                0.003577    15.871214     25.171132
## UCL Mean                0.004013    16.667166     26.183315
## Variance                0.000007    23.360224     37.776483
## Stdev                   0.002646     4.833242      6.146258
## Skewness                3.903304     1.097306      0.495697
## Kurtosis               25.937966     0.911503      0.200530
##             perimeter_worst   area_worst smoothness_worst
## nobs             569.000000 5.690000e+02       569.000000
## NAs                0.000000 0.000000e+00         0.000000
## Minimum           50.410000 1.852000e+02         0.071170
## Maximum          251.200000 4.254000e+03         0.222600
## 1. Quartile       84.110000 5.153000e+02         0.116600
## 3. Quartile      125.400000 1.084000e+03         0.146000
## Mean             107.261213 8.805831e+02         0.132369
## Median            97.660000 6.865000e+02         0.131300
## Sum            61031.630000 5.010518e+05        75.317730
## SE Mean            1.408692 2.386869e+01         0.000957
## LCL Mean         104.494332 8.337015e+02         0.130489
## UCL Mean         110.028094 9.274648e+02         0.134249
```

```
## Variance        1129.130847 3.241674e+05         0.000521
## Stdev             33.602542 5.693570e+02         0.022832
## Skewness           1.122223 1.849581e+00         0.413238
## Kurtosis           1.036019 4.321528e+00         0.490459
##             compactness_worst concavity_worst concave.points_worst
## nobs               569.000000      569.000000           569.000000
## NAs                  0.000000        0.000000             0.000000
## Minimum              0.027290        0.000000             0.000000
## Maximum              1.058000        1.252000             0.291000
## 1. Quartile          0.147200        0.114500             0.064930
## 3. Quartile          0.339100        0.382900             0.161400
## Mean                 0.254265        0.272188             0.114606
## Median               0.211900        0.226700             0.099930
## Sum                144.676810      154.875247            65.210941
## SE Mean              0.006596        0.008746             0.002756
## LCL Mean             0.241310        0.255010             0.109194
## UCL Mean             0.267220        0.289367             0.120019
## Variance             0.024755        0.043524             0.004321
## Stdev                0.157336        0.208624             0.065732
## Skewness             1.465795        1.144179             0.490021
## Kurtosis             2.981042        1.574447            -0.550001
##             symmetry_worst fractal_dimension_worst
## nobs            569.000000              569.000000
## NAs               0.000000                0.000000
## Minimum           0.156500                0.055040
## Maximum           0.663800                0.207500
## 1. Quartile       0.250400                0.071460
## 3. Quartile       0.317900                0.092080
## Mean              0.290076                0.083946
## Median            0.282200                0.080040
## Sum             165.053000               47.765170
## SE Mean           0.002594                0.000757
## LCL Mean          0.284981                0.082459
## UCL Mean          0.295170                0.085433
## Variance          0.003828                0.000326
## Stdev             0.061867                0.018061
## Skewness          1.426376                1.653824
## Kurtosis          4.369103                5.159356
```

Visualization

- Before plotiong, we need to normalization or standirdization. Because differences between values of features are very high to observe on plot.

```
listx= colnames(x)
data_x=x
i=1
for(i in 1:30)
{
  data_x[,i]=(data_x[,i]-mean(x[,i]))/stdev(x[,i])
}
```

- I plot features in 3 group and each group includes 10 features to observe better.

VIOLIN PLOTS: first ten features

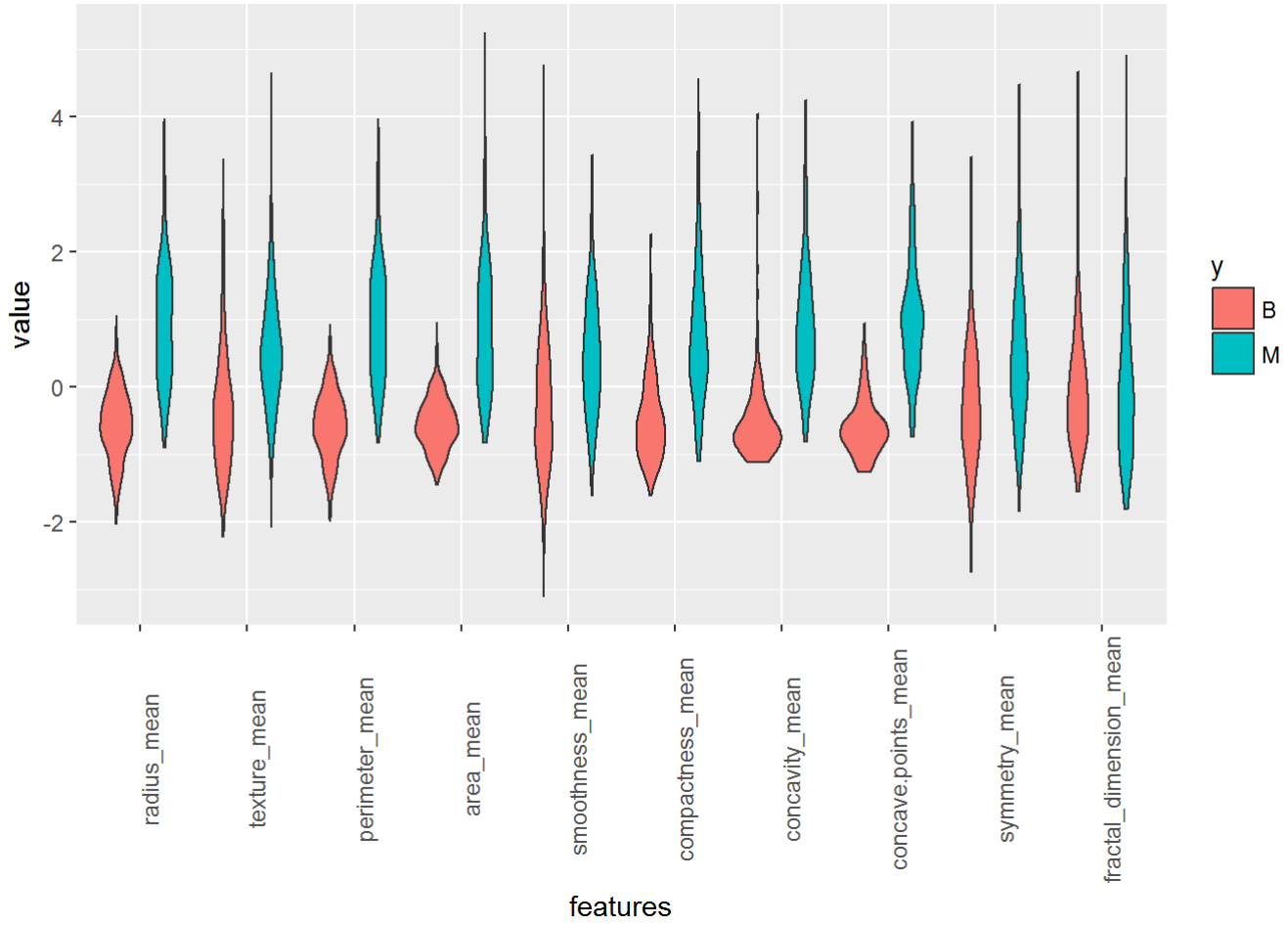
```
library(ggplot2)
data.normal=cbind(data_x,y)
#concatenate the feautures dataset with the labels
data.normal1=cbind(data_x[,1:10],y)

#reshape the data using the melt function
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 3.4.3
```

```
##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
df=melt(data.normal1,id.vars = "y",variable_name = "features")

ggplot(df,aes(x=features,y=value))+
    geom_violin(trim=T,aes(fill=y))+theme(axis.text.x = element_text(angle = 90))
```

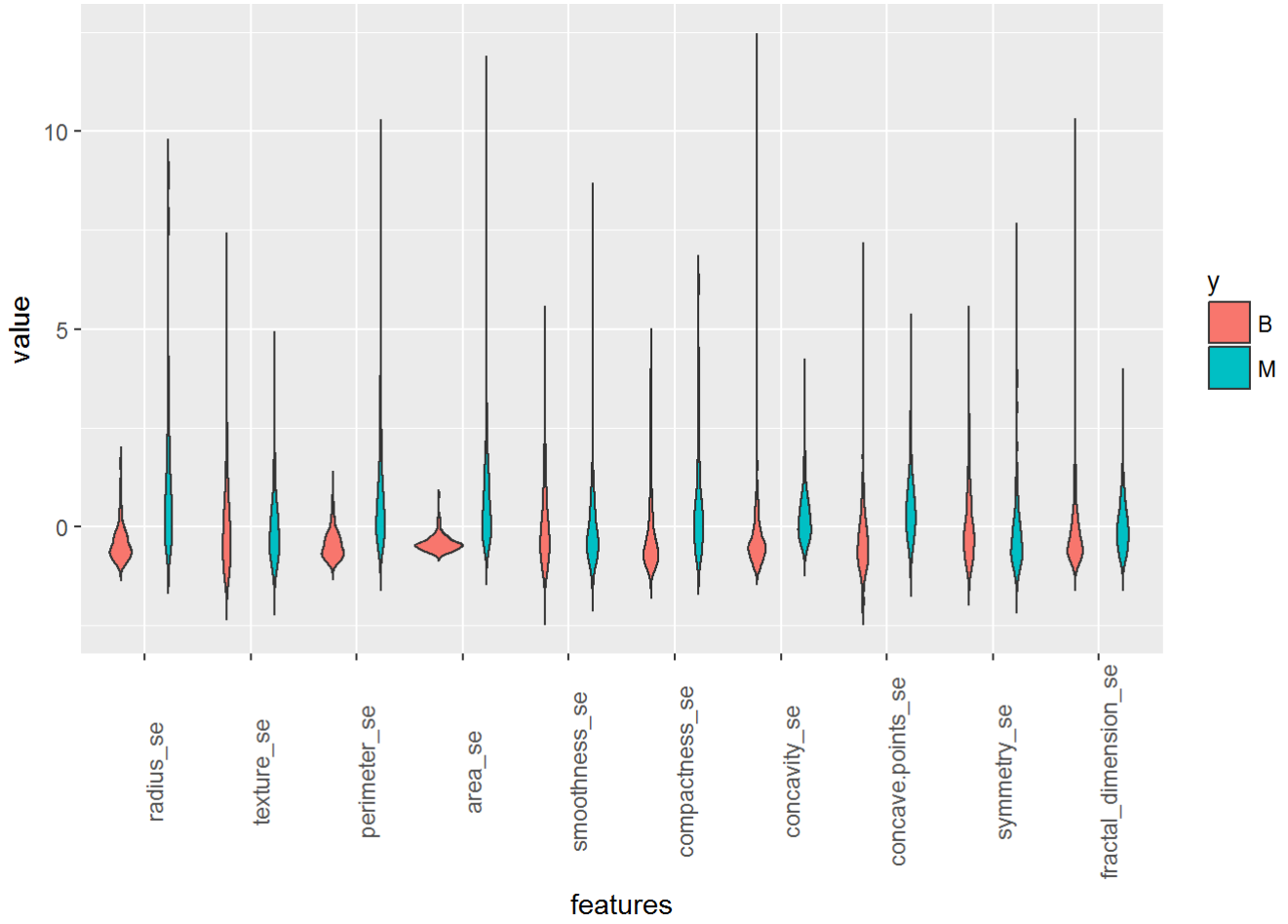


- Lets interpret the plot above together.
- In texture_mean feature, median of the Malignant and Benign looks like separated so it can be good for classification.

- However, in fractal_dimension_mean feature, median of the Malignant and Benign does not looks like separated so it does not gives good information for classification.

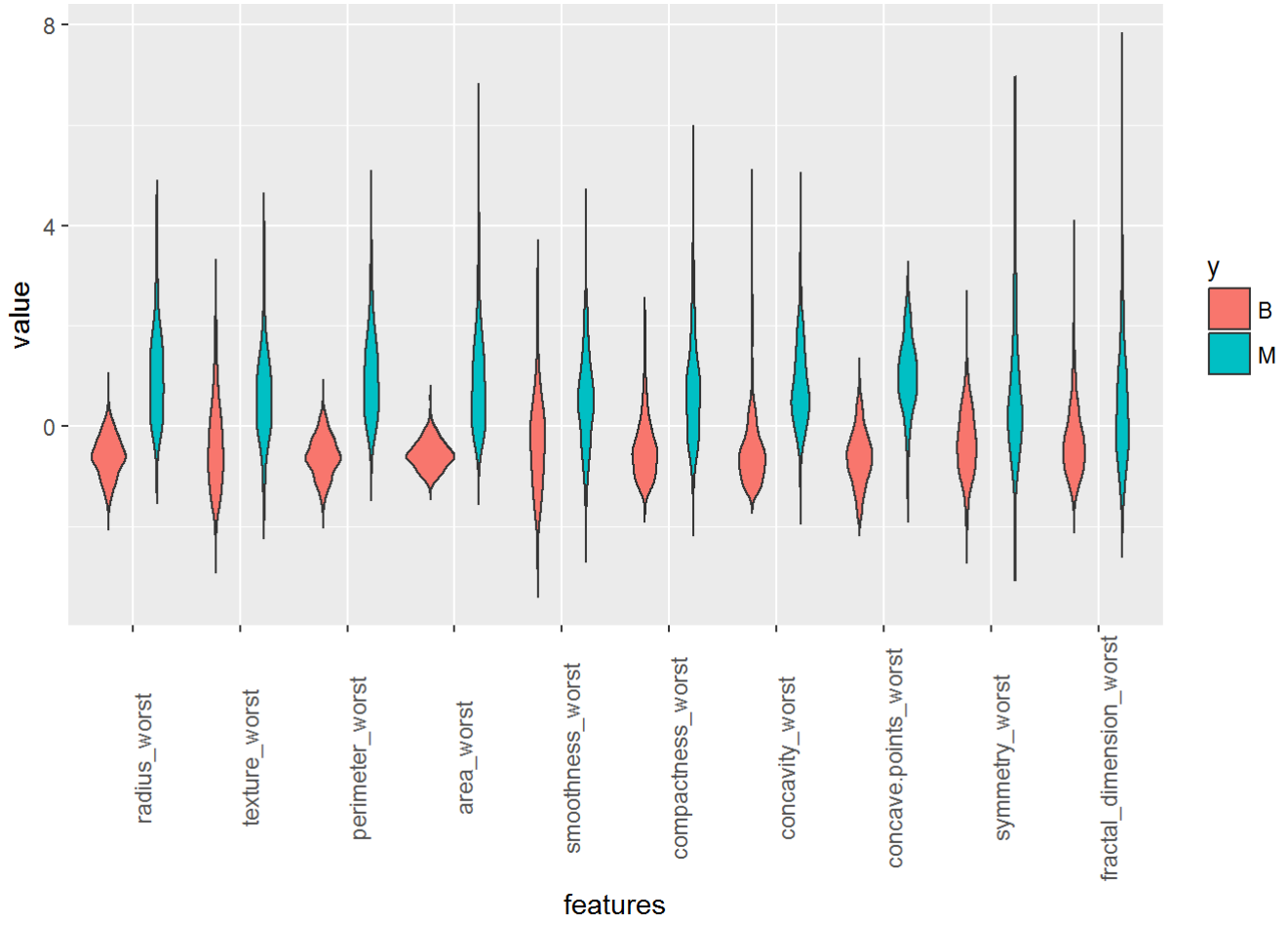
Second ten features

```
library(ggplot2)
#concatenate the feautures dataset with the labels
data.normal2=cbind(data_x[,11:20],y)

#reshape the data using the melt function
library(reshape)
df2=melt(data.normal2,id.vars = "y",variable_name = "features")

ggplot(df2,aes(x=features,y=value))+
    geom_violin(trim=F,aes(fill=y))+theme(axis.text.x = element_text(angle = 90))
```



Third ten features

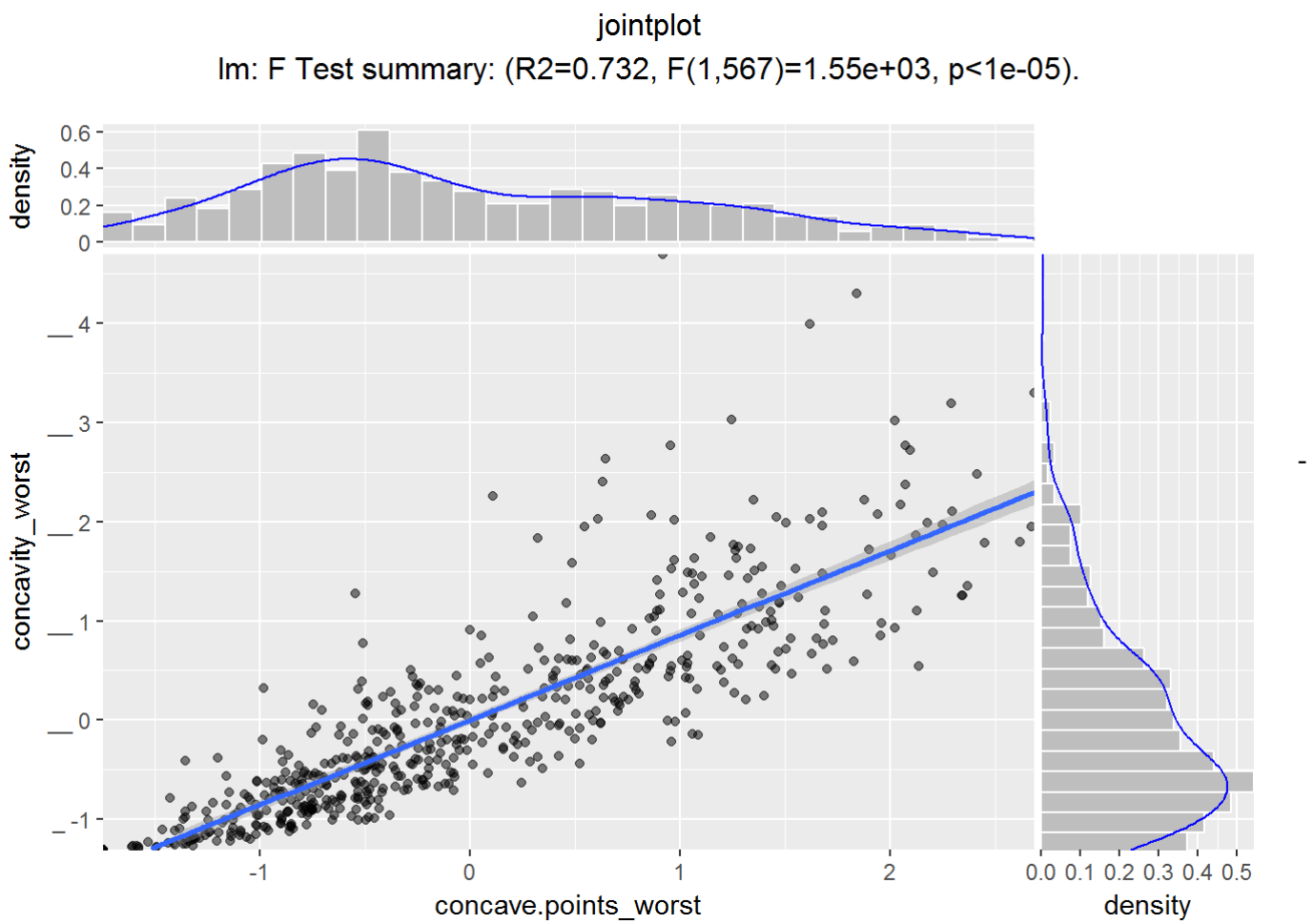
```
library(ggplot2)
#concatenate the feautures dataset with the labels
data.normal3=cbind(data_x[,21:30],y)

#reshape the data using the melt function
library(reshape)
df3=melt(data.normal3,id.vars = "y",variable_name = "features")

ggplot(df3,aes(x=features,y=value))+
    geom_violin(trim=F,aes(fill=y))+theme(axis.text.x = element_text(angle = 90))
```



- Lets interpret one more thing about plot above, variable of concavity_worst and concave point_worst looks like similar but how can we decide whether they are correlated with each other or not.
- (Not always true but, basically if the features are correlated with each other we can drop one of them)
- in order to compare two features deeper, lets use joint plot. Look at this in joint plot below, it is really correlated. Do not forget, we are not choosing features yet, we are just looking to have an idea about them.

```
library(WVPlots)
WVPlots::ScatterHist(frame=data_x,xvar="concave.points_worst",yvar="concavity_worst",title =
"jointplot",smoothmethod = "lm")
```

```
## Warning: Removed 4 rows containing missing values (geom_smooth).
```



What about three or more feauture comparision ? For this purpose we can use pair grid plot. Also it seems very cool :) And we discover one more thing radius_worst, perimeter_worst and area_worst are correlated as it can be seen pair grid plot. We definetely use these discoveries for feature selection.

```
require(GGally)
```

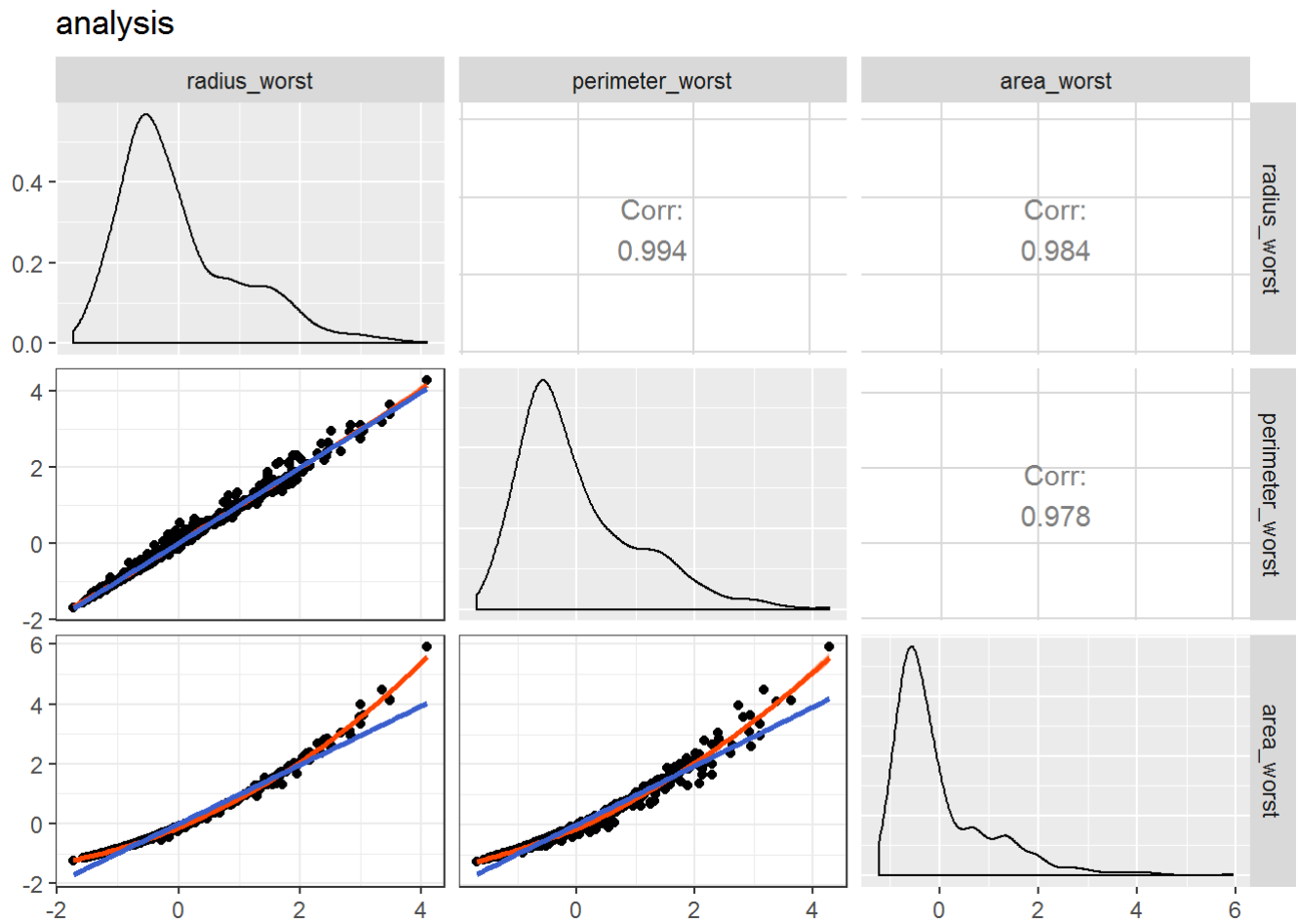
```
## Loading required package: GGally
```

```
## Warning: package 'GGally' was built under R version 3.4.3
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
require(ggplot2)

my_fn <- function(data, mapping, ...){
  p <- ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method=loess, fill="orangered1", color="orangered1", ...) +
    geom_smooth(method=lm, fill="palegreen3", color="royalblue3", ...)+theme_bw()
  p
}

g = ggpairs(data.normal,columns =c(21,23,24) , lower = list(continuous = my_fn),title="analysis")
g
```



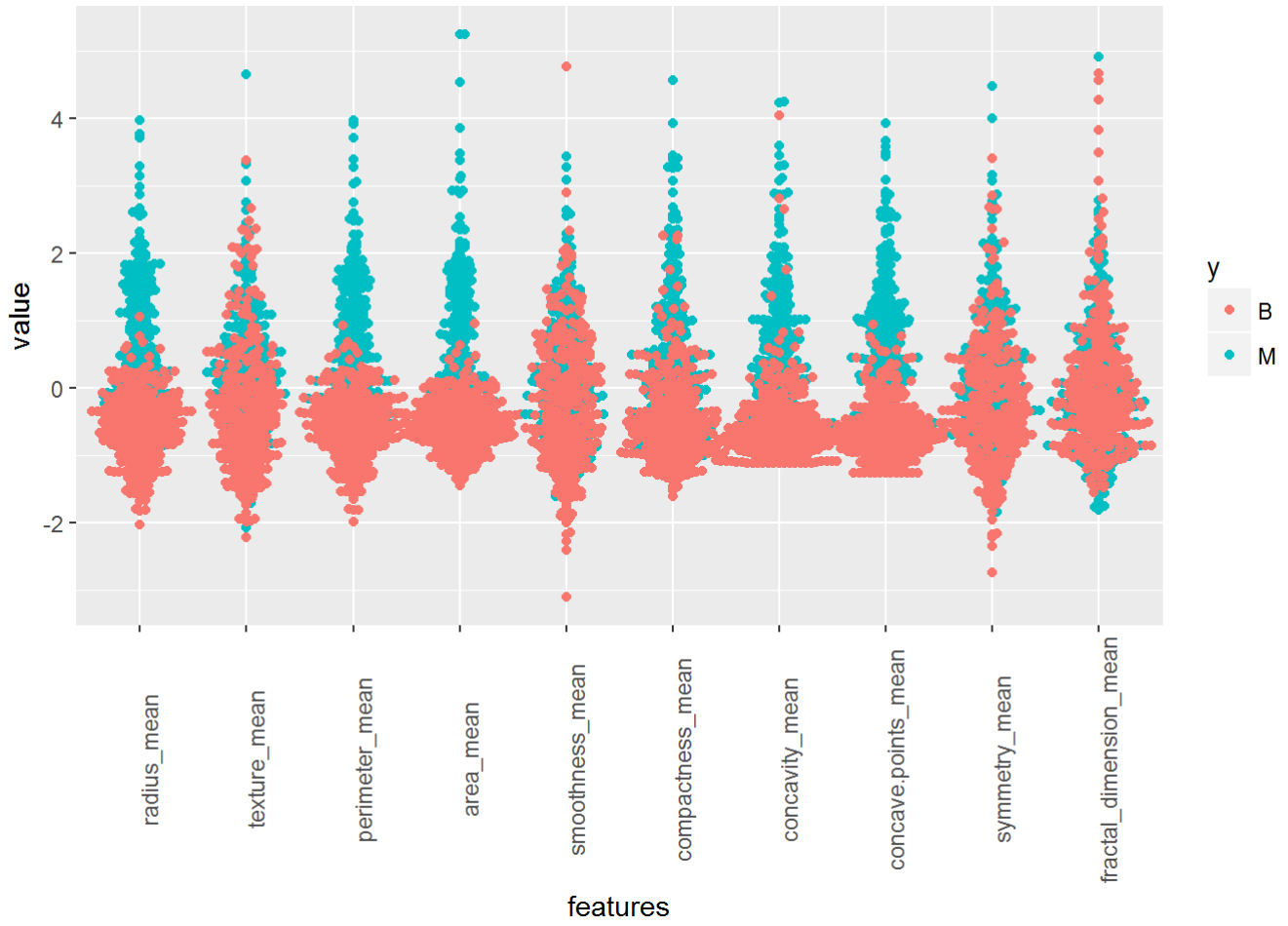
Swarm Plots!!

- first ten plots

```
library(ggbeeswarm)
```

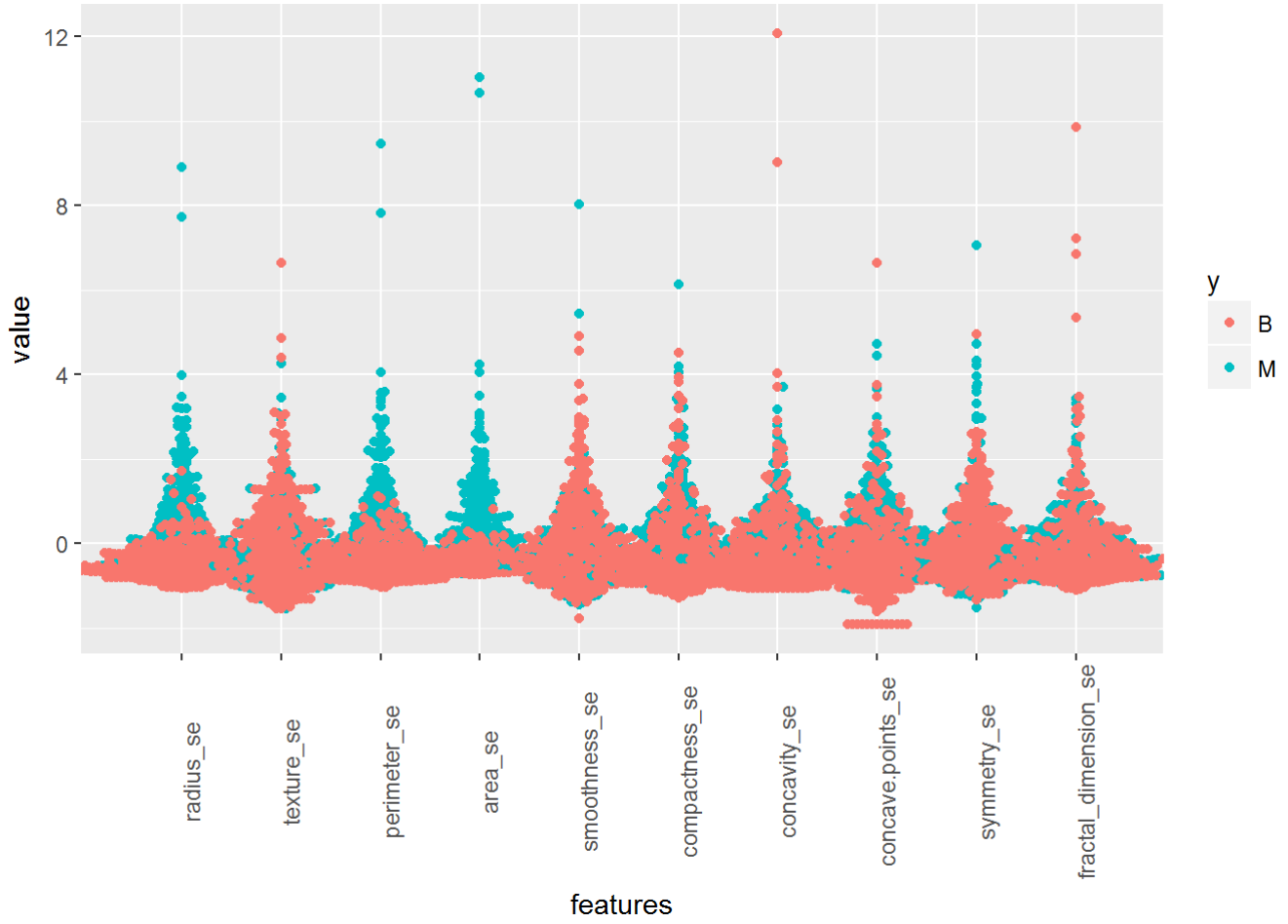
```
## Warning: package 'ggbeeswarm' was built under R version 3.4.3
```

```
ggplot(df,aes(features, value,col=y)) + geom_beeswarm(cex = 0.5)+ theme(axis.text.x = element_text(angle = 90))
```



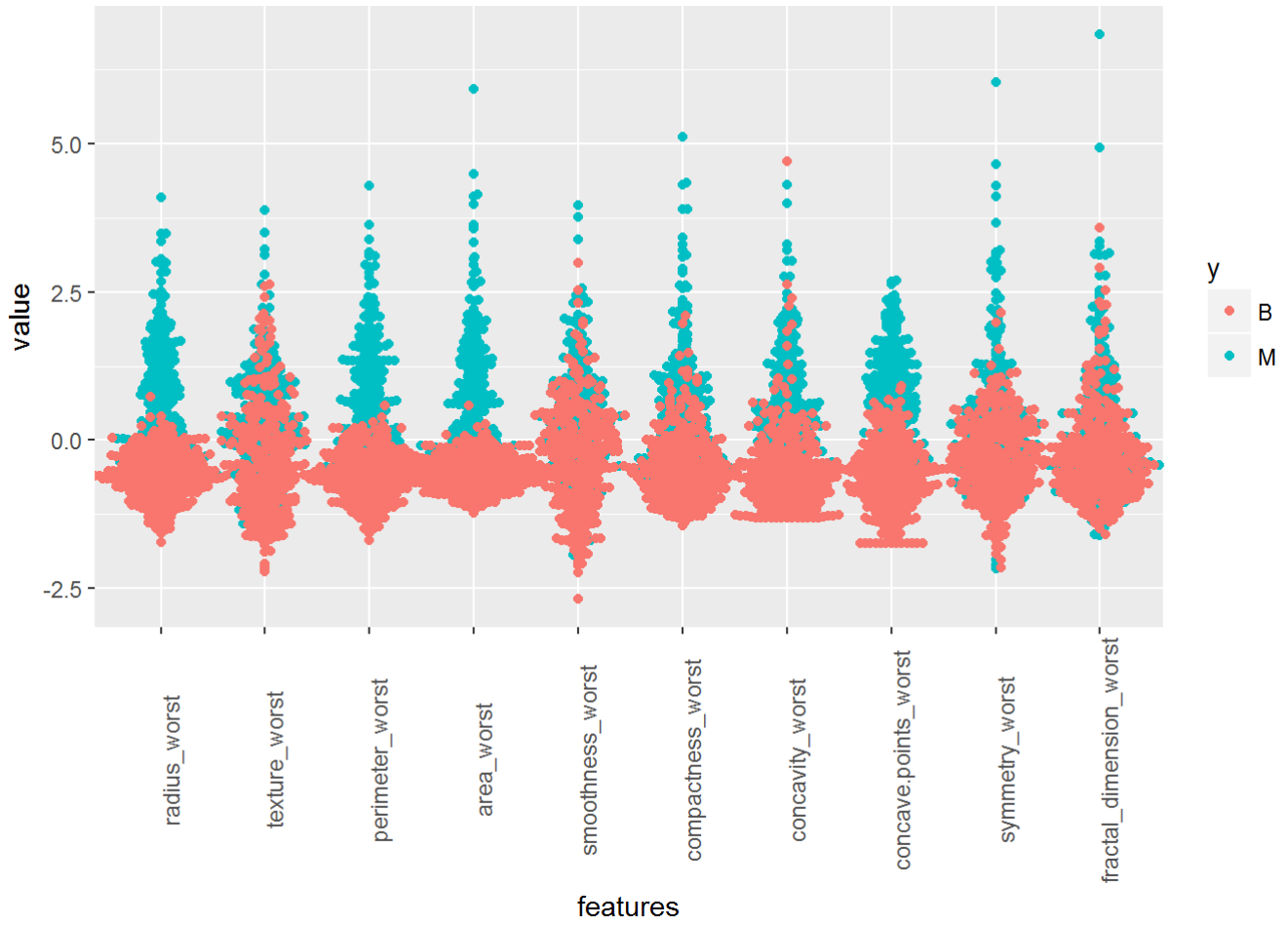
- Second set of Plots

```
library(ggbeeswarm)
ggplot(df2,aes(features, value,col=y)) + geom_beeswarm(cex = 0.5)+ theme(axis.text.x = elemen
t_text(angle = 90))
```



- Third Set of Plots

```
library(ggbeeswarm)
ggplot(df3,aes(features, value,col=y)) + geom_beeswarm(cex = 0.5)+ theme(axis.text.x = elemen
t_text(angle = 90))
```



Heat Map

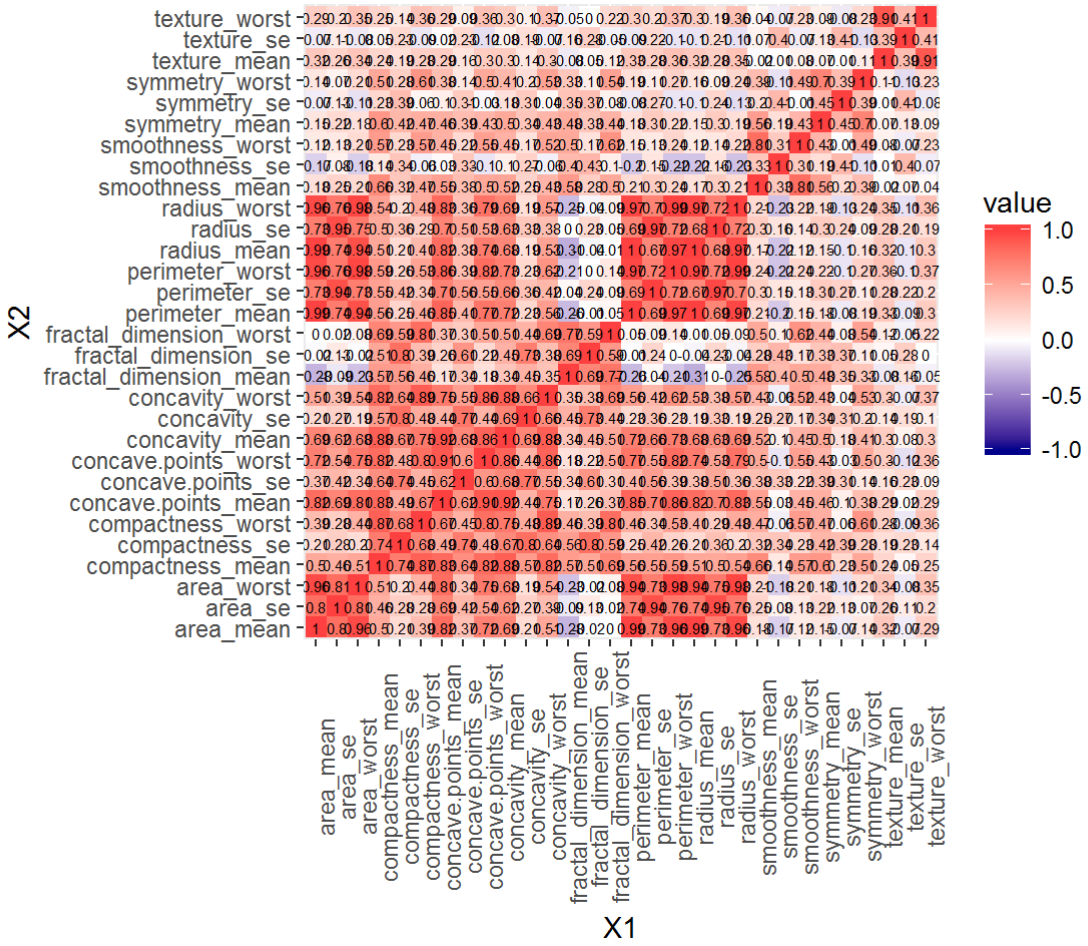
- What if we want to observe all correlation between features? Yes, you are right. The answer is heatmap that is old but powerful plot method.

```
cormat=round(cor(x),2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast
```

```
melted_cormat=melt(cormat)
g1=ggplot(data = melted_cormat,aes(X1,X2,fill=value))+geom_tile()+scale_fill_gradient2(low =
"blue4",mid = "white", high = "brown1",midpoint = 0,limit=c(-1,1))+ theme(axis.text.x = eleme
nt_text(angle = 90))+coord_fixed()
g1+geom_text(aes(X1,X2,label=value),size=2)
```



- the radius, perimeter and area are highly correlated as expected from their relation so from these we will use anyone of them
- compactness_mean, concavity_mean and concavepoint_mean are highly correlated so we will use compactness_mean from here

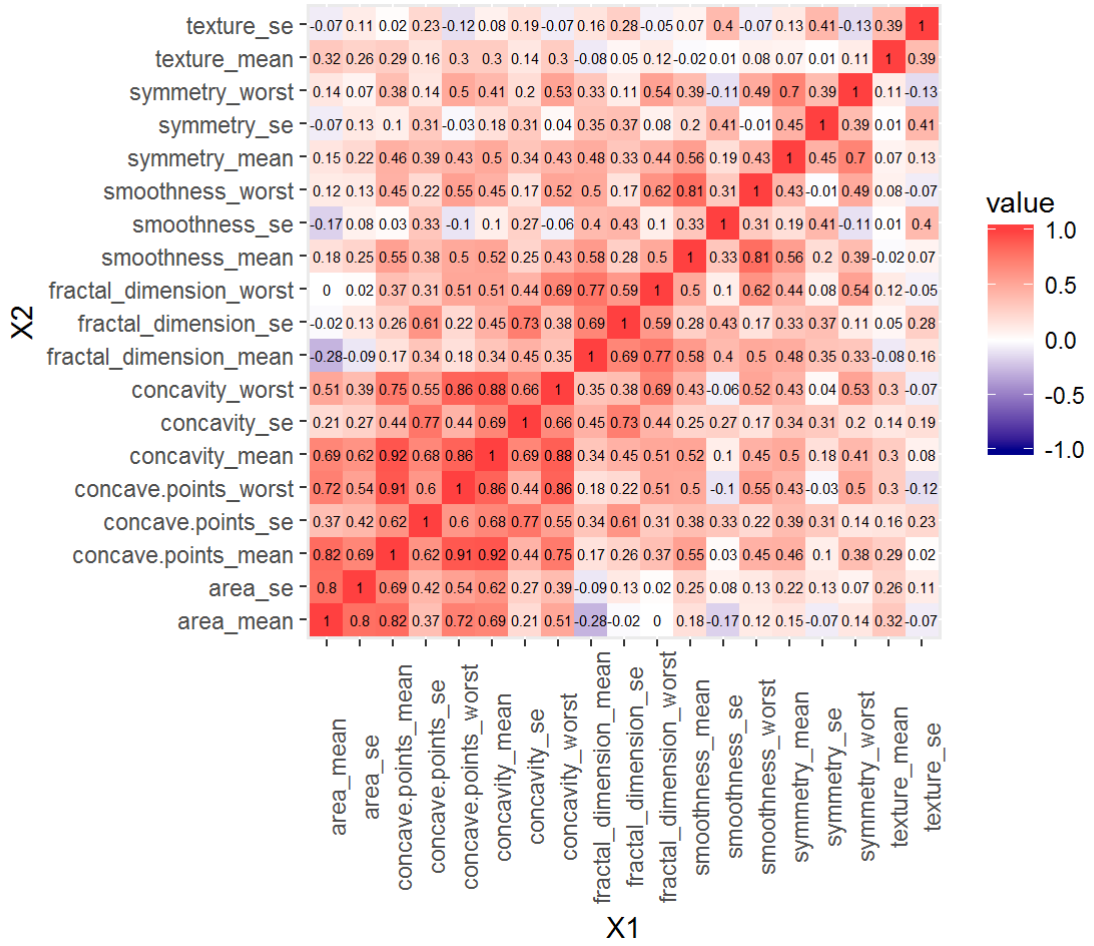
Dropping off Highly Correlated Parameters

```
drop_list1 = c('perimeter_mean','radius_mean','compactness_mean','concave points_mean','radiu
s_se','perimeter_se','radius_worst','perimeter_worst','compactness_worst','concave points_wor
st','compactness_se','concave points_se','texture_worst','area_worst')

x2<- x[, !colnames(x) %in% drop_list1]
#colnames(x2)
head(x2)
```

```
##   texture_mean area_mean smoothness_mean concavity_mean
## 1        10.38    1001.0         0.11840         0.3001
## 2        17.77    1326.0         0.08474         0.0869
## 3        21.25    1203.0         0.10960         0.1974
## 4        20.38     386.1         0.14250         0.2414
## 5        14.34    1297.0         0.10030         0.1980
## 6        15.70     477.1         0.12780         0.1578
##   concave.points_mean symmetry_mean fractal_dimension_mean texture_se
## 1             0.14710        0.2419                0.07871     0.9053
## 2             0.07017        0.1812                0.05667     0.7339
## 3             0.12790        0.2069                0.05999     0.7869
## 4             0.10520        0.2597                0.09744     1.1560
## 5             0.10430        0.1809                0.05883     0.7813
## 6             0.08089        0.2087                0.07613     0.8902
##   area_se smoothness_se concavity_se concave.points_se symmetry_se
## 1  153.40      0.006399      0.05373           0.01587     0.03003
## 2   74.08      0.005225      0.01860           0.01340     0.01389
## 3   94.03      0.006150      0.03832           0.02058     0.02250
## 4   27.23      0.009110      0.05661           0.01867     0.05963
## 5   94.44      0.011490      0.05688           0.01885     0.01756
## 6   27.19      0.007510      0.03672           0.01137     0.02165
##   fractal_dimension_se smoothness_worst concavity_worst
## 1             0.006193           0.1622          0.7119
## 2             0.003532           0.1238          0.2416
## 3             0.004571           0.1444          0.4504
## 4             0.009208           0.2098          0.6869
## 5             0.005115           0.1374          0.4000
## 6             0.005082           0.1791          0.5355
##   concave.points_worst symmetry_worst fractal_dimension_worst
## 1               0.2654         0.4601                 0.11890
## 2               0.1860         0.2750                 0.08902
## 3               0.2430         0.3613                 0.08758
## 4               0.2575         0.6638                 0.17300
## 5               0.1625         0.2364                 0.07678
## 6               0.1741         0.3985                 0.12440
```

correlation map

```
cormat=round(cor(x2),2)
library(reshape2)
melted_cormat=melt(cormat)
ggheatmap=ggplot(data = melted_cormat,aes(X1,X2,fill=value))+geom_tile()+scale_fill_gradient2
(low = "blue4",mid = "white", high = "brown1",midpoint = 0,limit=c(-1,1))+ theme(axis.text.x
 = element_text(angle = 90))+coord_fixed()

ggheatmap+geom_text(aes(X1,X2,label=value),size=2)
```



RANDOM FOREST

```
combi=cbind(x2,y)
idx=sample.int(nrow(x2),size = 0.75*nrow(x2))
set.seed(1203)

train = combi[idx,]
test= combi[-idx,]
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:timeSeries':
##
##     outlier
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
fit <-randomForest::randomForest(as.factor(y) ~.,
                    data=train,
                    importance=TRUE,
                    ntree=2000)
```

```
Prediction <- predict(fit, test)
library(caret)
```

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

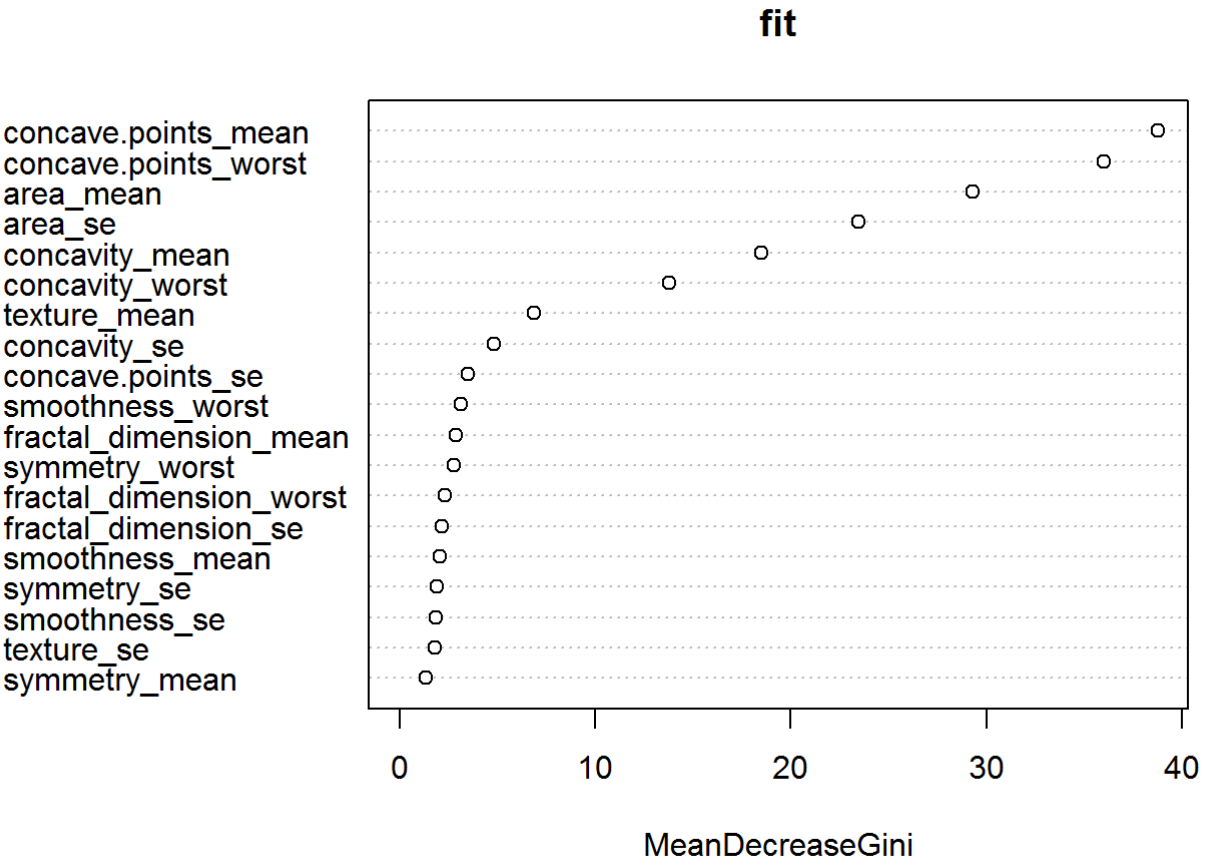
```
## Warning: package 'lattice' was built under R version 3.4.3
```

```
cm=caret::confusionMatrix(data=Prediction,reference = test$y)
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 83  1
##          M  4 55
##
##                Accuracy : 0.965
##                  95% CI : (0.9203, 0.9886)
##     No Information Rate : 0.6084
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9273
##  Mcnemar's Test P-Value : 0.3711
##
##             Sensitivity : 0.9540
##             Specificity : 0.9821
##          Pos Pred Value : 0.9881
##          Neg Pred Value : 0.9322
##              Prevalence : 0.6084
##          Detection Rate : 0.5804
##    Detection Prevalence : 0.5874
##       Balanced Accuracy : 0.9681
##
##        'Positive' Class : B
##
```

- The importance=TRUE argument allows us to inspect variable importance as we'll see,and the ntree argument specifies how many trees we want to grow.
- which variables where important

```
varImpPlot(fit,type=2)
```

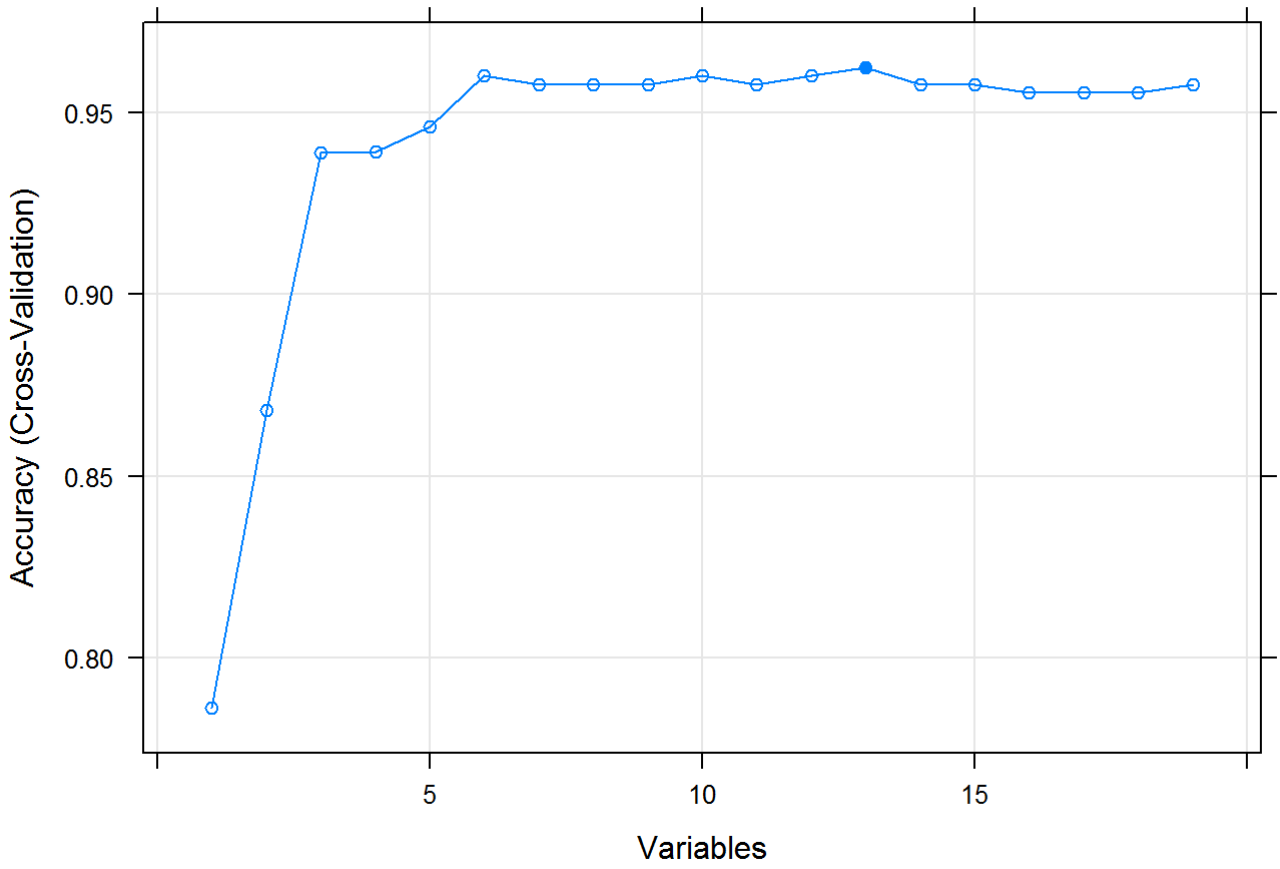


Feature selection on importance

```
# define the control using a random forest selection function
control <- rfeControl(functions=rfFuncs, method="cv", number=10)

results=caret::rfe(x = train[,1:19],train[,20],sizes=c(1:19),rfeControl=control)
predictors(results)
```

```
##  [1] "area_mean"               "area_se"
##  [3] "concave.points_worst"    "concave.points_mean"
##  [5] "texture_mean"            "concavity_worst"
##  [7] "concavity_mean"          "smoothness_worst"
##  [9] "fractal_dimension_mean"  "concavity_se"
## [11] "fractal_dimension_worst" "symmetry_worst"
## [13] "concave.points_se"
```

```
#Plot the results
plot(results, type=c('g','o'))
```



Model After Feature Selection

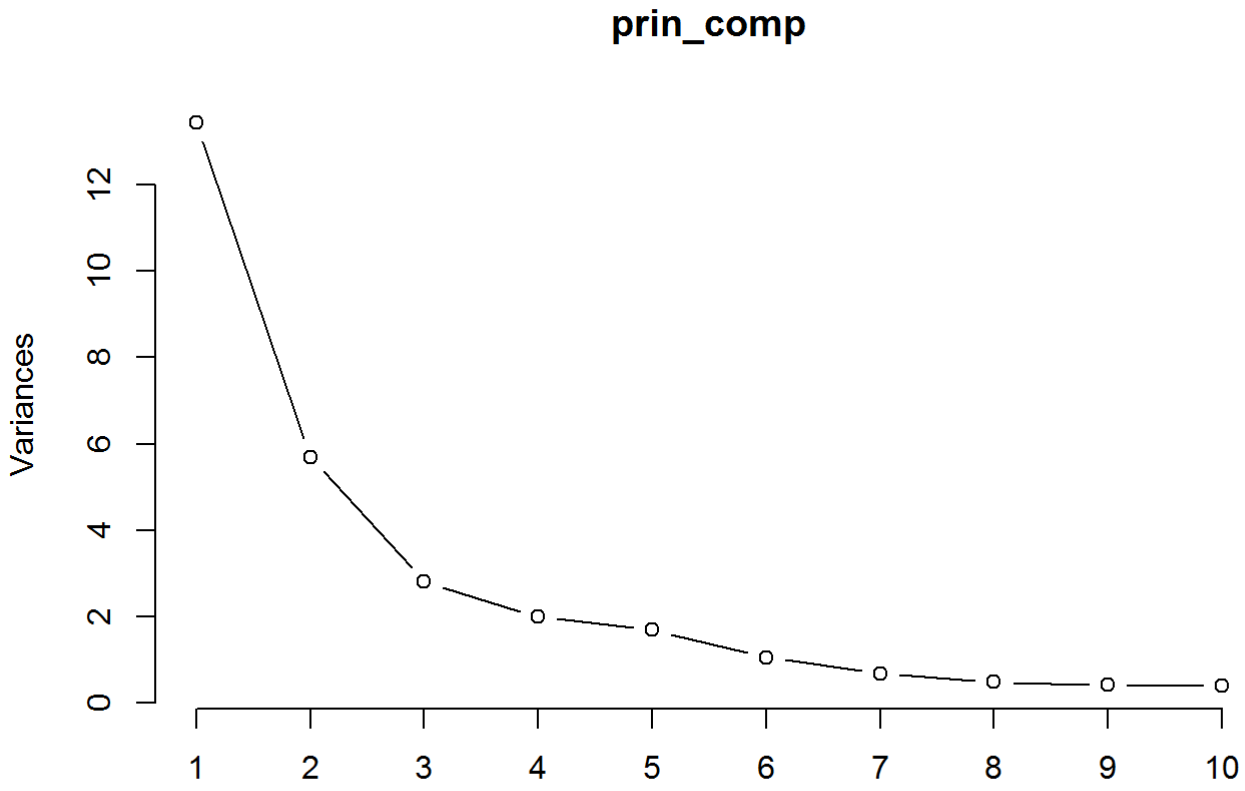
```
fit <-randomForest::randomForest(as.factor(y) ~concave.points_worst +area_se + area_mean + co
ncave.points_mean+texture_mean,
                    data=train,
                    importance=TRUE,
                    ntree=2000)

Prediction <- predict(fit, test)
cm=confusionMatrix(data=Prediction,reference = test$y)
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 84  4
##          M  3 52
##
##                Accuracy : 0.951
##                  95% CI : (0.9017, 0.9801)
##     No Information Rate : 0.6084
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8969
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9655
##             Specificity : 0.9286
##          Pos Pred Value : 0.9545
##          Neg Pred Value : 0.9455
##              Prevalence : 0.6084
##          Detection Rate : 0.5874
##    Detection Prevalence : 0.6154
##       Balanced Accuracy : 0.9470
##
##        'Positive' Class : B
##
```

- Accuracy is almost 96% and as it can be seen in confusion matrix, we make few wrong prediction. What we did up to now is that we choose features according to correlation matrix and according to rfe method.
- Now lets see other feature selection methods to find better results.

PRINCIPAL COMPONENT ANALYSIS

```
#normalized data
data=data.normal


#split the data in 70-30
idx=sample.int(nrow(data),size = 0.7*nrow(data))
set.seed(120)

pca.train = data[idx,-31]
pca.test= data[-idx,-31]

#Use PCA
prin_comp <- prcomp(pca.train, scale. = T,center = TRUE,)
plot(prin_comp, type = "l")
```



- According to variance ration, 3 component can be chosen.