AnalysisScript - Breast Cancer Wisconsin (Diagnostic)

Thanh La, Son Luong

10/17/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)
```

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

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(reshape)
```

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

## The following objects are masked from 'package:reshape2':
##
##     colsplit, melt, recast

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

## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```r
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(ISLR)
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

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

```r
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:reshape':
##
##     stamp
```

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```r
library(dplyr)
```

```r
workfile <- read.csv("data.csv")
numeric_workfile <- workfile[,3:32 ]
```

workfile Exploration

```r
colnames(workfile)
```

```
##  [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"
```

```r
summary(numeric_workfile)
```

```
##   radius_mean      texture_mean   perimeter_mean     area_mean
##  Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
##  Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
##  Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
##  3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
##  Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##  smoothness_mean   compactness_mean  concavity_mean    concave.points_mean
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
##  symmetry_mean    fractal_dimension_mean   radius_se        texture_se
##  Min.   :0.1060   Min.   :0.04996        Min.   :0.1115   Min.   :0.3602
##  1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324   1st Qu.:0.8339
##  Median :0.1792   Median :0.06154        Median :0.3242   Median :1.1080
##  Mean   :0.1812   Mean   :0.06280        Mean   :0.4052   Mean   :1.2169
##  3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789   3rd Qu.:1.4740
##  Max.   :0.3040   Max.   :0.09744        Max.   :2.8730   Max.   :4.8850
##   perimeter_se       area_se        smoothness_se      compactness_se
##  Min.   : 0.757   Min.   :  6.802   Min.   :0.001713   Min.   :0.002252
##  1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080
##  Median : 2.287   Median : 24.530   Median :0.006380   Median :0.020450
##  Mean   : 2.866   Mean   : 40.337   Mean   :0.007041   Mean   :0.025478
##  3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450
##  Max.   :21.980   Max.   :542.200   Max.   :0.031130   Max.   :0.135400
##   concavity_se     concave.points_se   symmetry_se       fractal_dimension_se
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948
##  1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480
```

```
##  Median :0.02589   Median :0.010930   Median :0.018730   Median :0.0031870
##  Mean   :0.03189   Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949
##  3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580
##  Max.   :0.39600   Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400
##   radius_worst   texture_worst   perimeter_worst    area_worst
##  Min.   : 7.93   Min.   :12.02   Min.   : 50.41   Min.   : 185.2
##  1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11   1st Qu.: 515.3
##  Median :14.97   Median :25.41   Median : 97.66   Median : 686.5
##  Mean   :16.27   Mean   :25.68   Mean   :107.26   Mean   : 880.6
##  3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40   3rd Qu.:1084.0
##  Max.   :36.04   Max.   :49.54   Max.   :251.20   Max.   :4254.0
##  smoothness_worst  compactness_worst concavity_worst  concave.points_worst
##  Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493
##  Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993
##  Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461
##  3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140
##  Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100
##  symmetry_worst   fractal_dimension_worst
##  Min.   :0.1565   Min.   :0.05504
##  1st Qu.:0.2504   1st Qu.:0.07146
##  Median :0.2822   Median :0.08004
##  Mean   :0.2901   Mean   :0.08395
##  3rd Qu.:0.3179   3rd Qu.:0.09208
##  Max.   :0.6638   Max.   :0.20750
```

```
all(is.na(workfile))
```

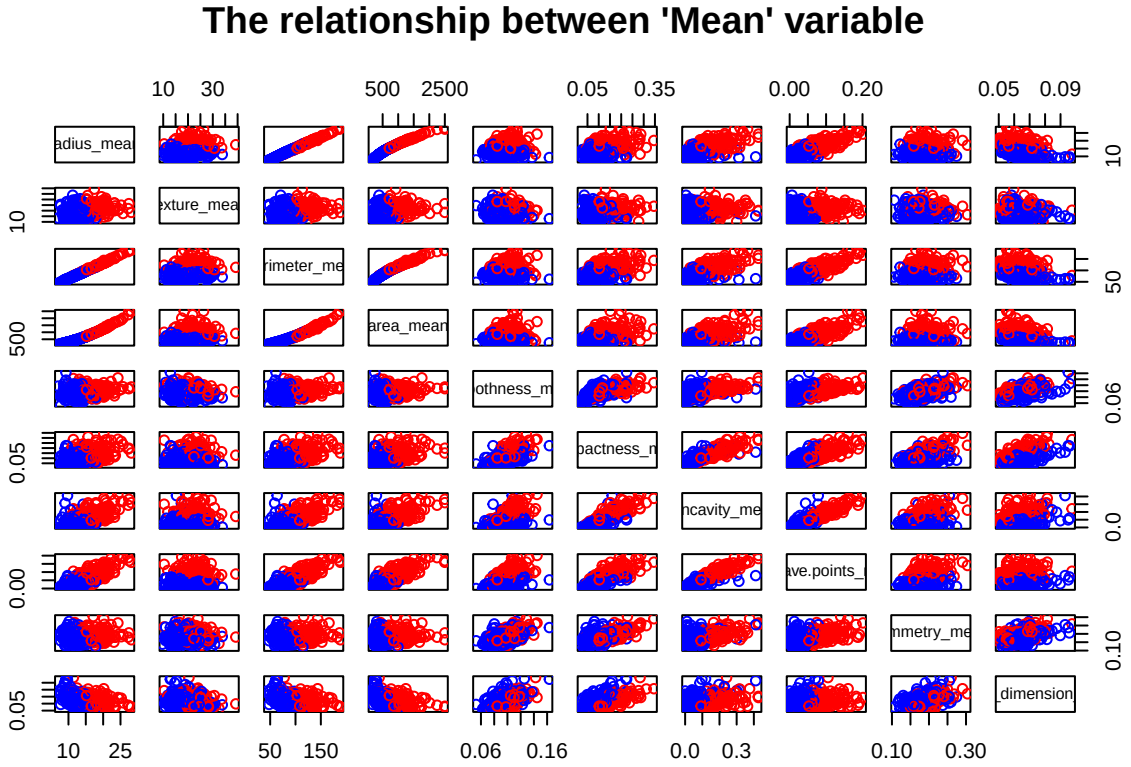
```
## [1] FALSE
```

workfile Visualization

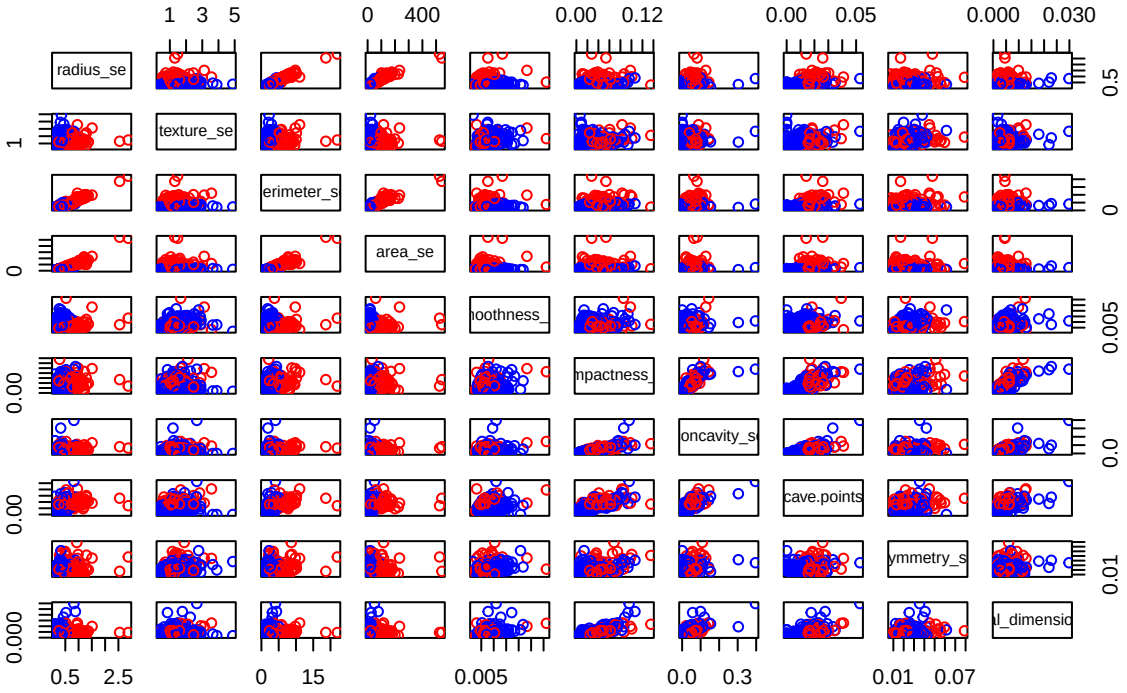
I. Relationship Between Variable

```
group = NA
group[workfile$diagnosis == "B"] = 1
group[workfile$diagnosis == "M"] = 2


workfile %>%
  select_if(is.numeric) %>%
  dplyr::select(contains("mean")) %>%
  pairs(col = c("blue", "red")[group], main = "The relationship between 'Mean' variable")
```

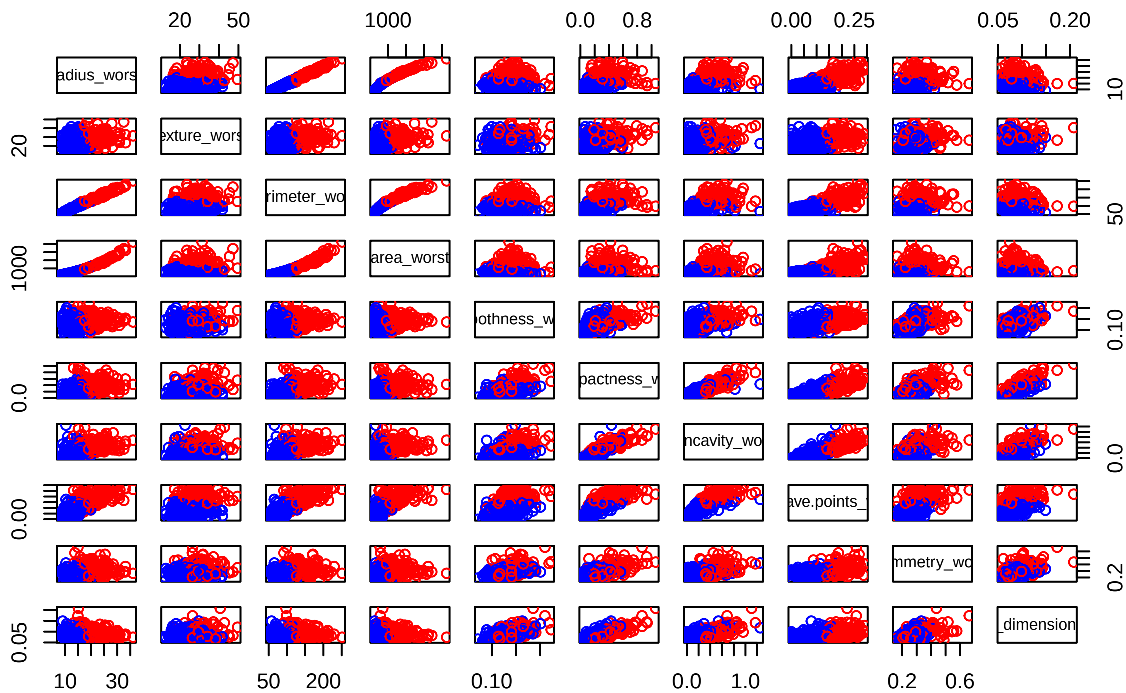



```
workfile %>%
  select_if(is.numeric) %>%
  dplyr::select(contains("SE")) %>%
  pairs(col = c("blue", "red")[group], main = "The relationship between 'Standard Error' variable")
```




```
workfile %>%
  select_if(is.numeric) %>%
  dplyr::select(contains("worst")) %>%
  pairs(col = c("blue", "red")[group], main = "The relationship between 'Worst' variable")
```

The relationship between 'Worst' variable



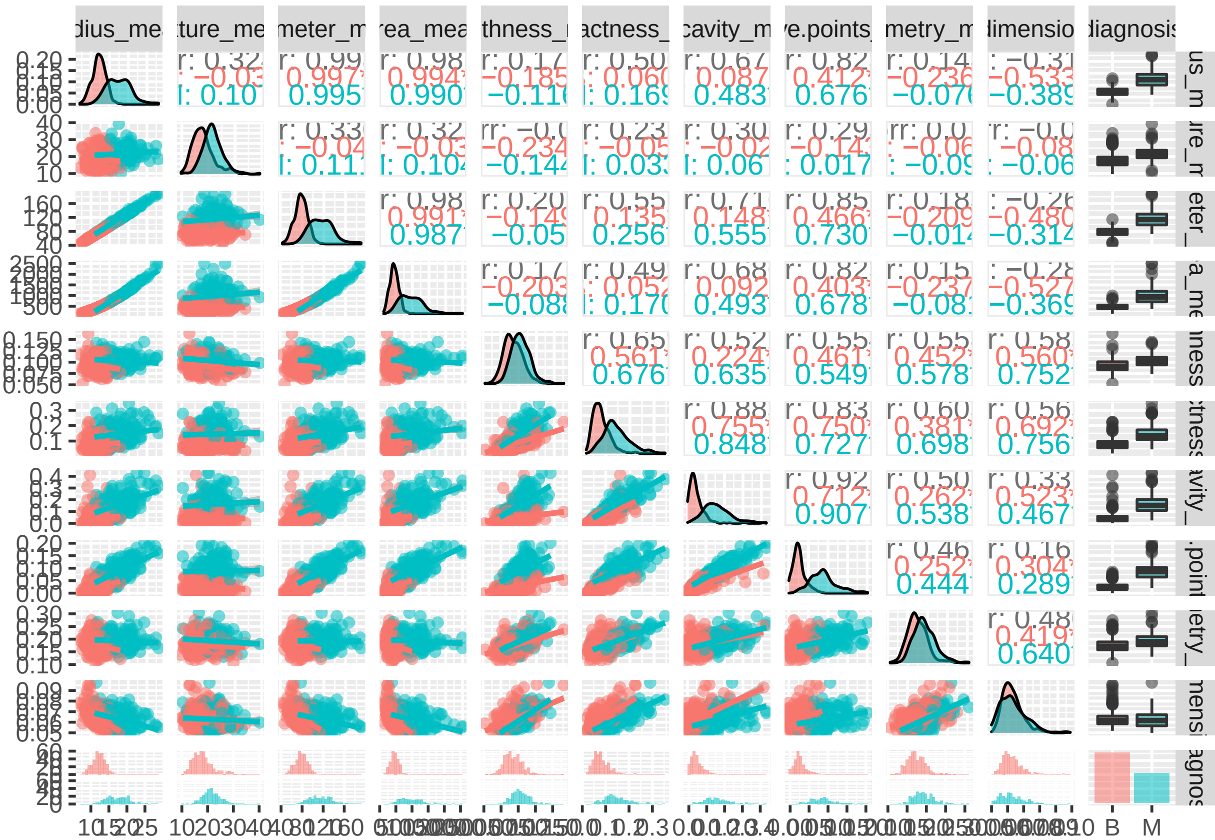
II. Differently Distributed Between Two Group of Diagnosis

look at how the variables are differently distributed between the two groups

these variables below are greatly different among 2 group of diagnosis

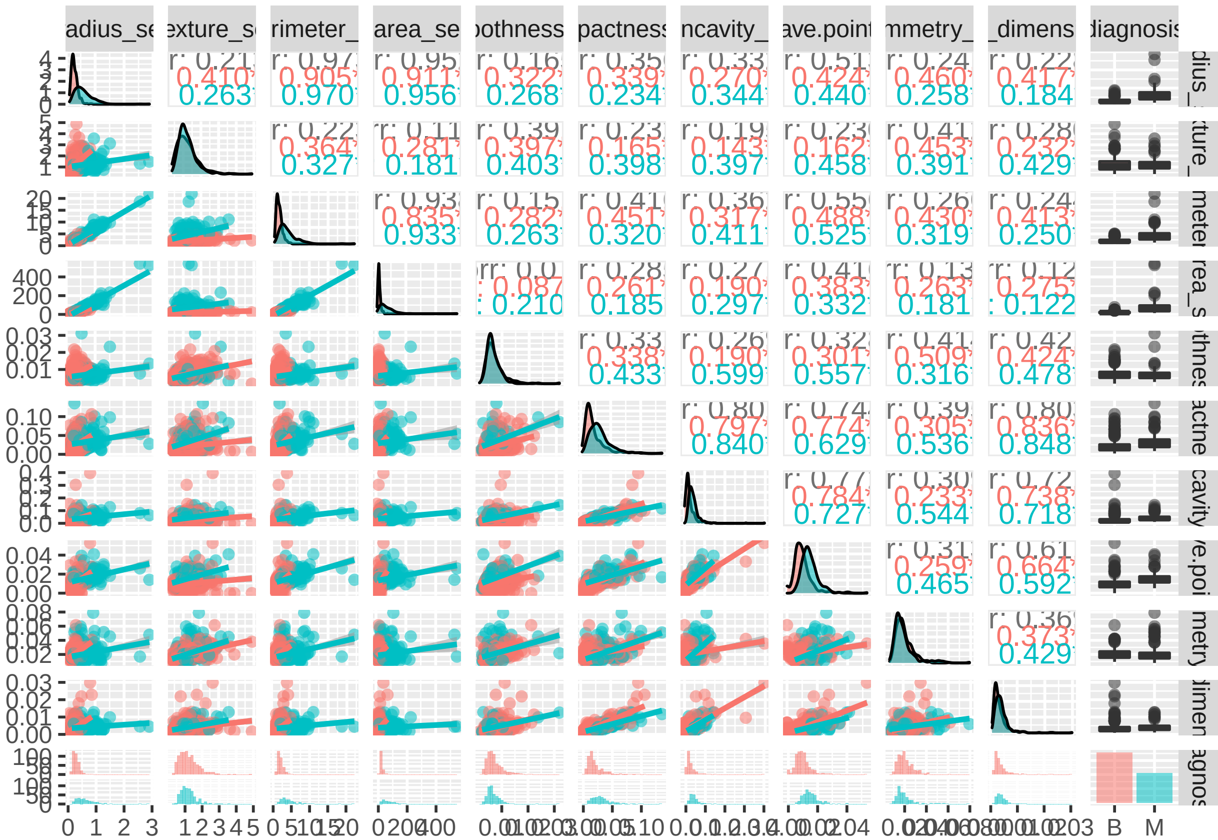
```
ggpairs(workfile[,c(3:12,2)], aes(color=diagnosis, alpha=0.5), lower=list(continuous="smooth"))
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



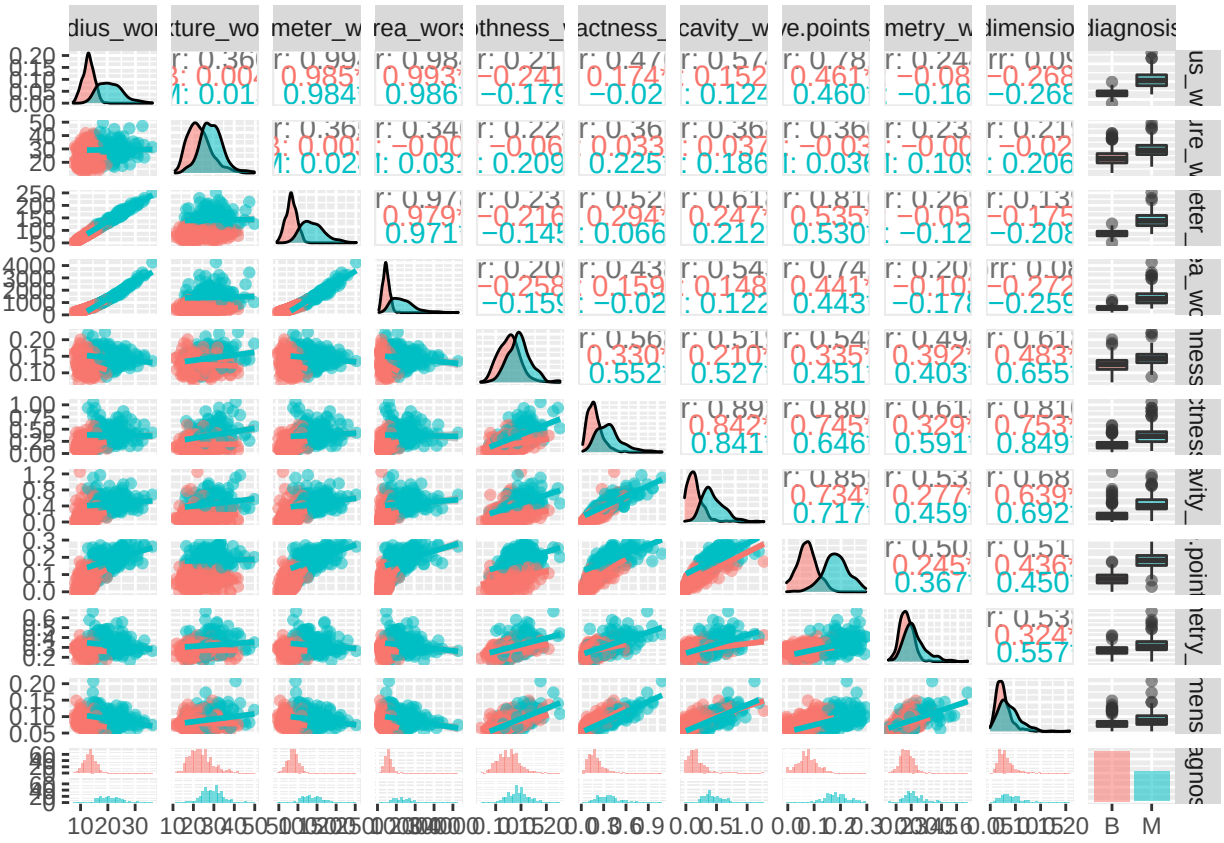

```
ggpairs(workfile[,c(13:22,2)], aes(color=diagnosis, alpha=0.5), lower=list(continuous="smooth"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggpairs(workfile[,c(23:32,2)], aes(color=diagnosis, alpha=0.5), lower=list(continuous="smooth"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



cor()# cov()# https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/cor

III. Logistic Regression Model

```
workfile$diagnosis = as.factor(workfile$diagnosis)
workfile = subset(workfile, select = -c(id))
```

```
head(workfile)
```

```
##   diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 1         M       17.99        10.38          122.80    1001.0         0.11840
## 2         M       20.57        17.77          132.90    1326.0         0.08474
## 3         M       19.69        21.25          130.00    1203.0         0.10960
## 4         M       11.42        20.38           77.58     386.1         0.14250
## 5         M       20.29        14.34          135.10    1297.0         0.10030
## 6         M       12.45        15.70           82.57     477.1         0.12780
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## 1          0.27760         0.3001             0.14710         0.2419
## 2          0.07864         0.0869             0.07017         0.1812
## 3          0.15990         0.1974             0.12790         0.2069
## 4          0.28390         0.2414             0.10520         0.2597
## 5          0.13280         0.1980             0.10430         0.1809
## 6          0.17000         0.1578             0.08089         0.2087
```

```
##   fractal_dimension_mean radius_se texture_se perimeter_se area_se
## 1                0.07871    1.0950     0.9053        8.589  153.40
## 2                0.05667    0.5435     0.7339        3.398   74.08
## 3                0.05999    0.7456     0.7869        4.585   94.03
## 4                0.09744    0.4956     1.1560        3.445   27.23
## 5                0.05883    0.7572     0.7813        5.438   94.44
## 6                0.07613    0.3345     0.8902        2.217   27.19
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## 1      0.006399        0.04904      0.05373           0.01587     0.03003
## 2      0.005225        0.01308      0.01860           0.01340     0.01389
## 3      0.006150        0.04006      0.03832           0.02058     0.02250
## 4      0.009110        0.07458      0.05661           0.01867     0.05963
## 5      0.011490        0.02461      0.05688           0.01885     0.01756
## 6      0.007510        0.03345      0.03672           0.01137     0.02165
##   fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
## 1             0.006193        25.38         17.33          184.60     2019.0
## 2             0.003532        24.99         23.41          158.80     1956.0
## 3             0.004571        23.57         25.53          152.50     1709.0
## 4             0.009208        14.91         26.50           98.87      567.7
## 5             0.005115        22.54         16.67          152.20     1575.0
## 6             0.005082        15.47         23.75          103.40      741.6
##   smoothness_worst compactness_worst concavity_worst concave.points_worst
## 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
##   symmetry_worst fractal_dimension_worst
## 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
```

Apply logistic regression to data

To make the logistic regression model more efficient (or improve the probability of the prediction), I will separate the characteristic variable to different part such as: size, shape, surface + Size: Radius, perimeter, area, compactness, fractal dimension

```
by_size = function(workfile){
  workfile_glm = glm(diagnosis~ radius_mean + radius_se + radius_worst + perimeter_mean + perimeter_se +
  return(workfile_glm)
}
```

+shape, surface: texture, smoothness, concavity, concave.points, symmetry, fractal dimension

```
by_shape = function(workfile){
  workfile_glm = glm(diagnosis~ texture_mean + texture_se + texture_worst + smoothness_mean + smoothness
  return(workfile_glm)
}
```

Build a Train - Test of Logistic Regression

Train - test by size

```
workfile_glm = by_size(train_workfile)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(workfile_glm)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius_mean + radius_se + radius_worst +
##     perimeter_mean + perimeter_se + perimeter_worst + area_mean +
##     area_se + area_worst + compactness_mean + compactness_se +
##     compactness_worst + fractal_dimension_mean + fractal_dimension_se +
##     fractal_dimension_worst, family = "binomial", data = workfile)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.65581  -0.13118  -0.04021   0.00024   2.74211
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              2.369e+01  1.610e+01   1.471  0.14127
## radius_mean             -1.829e+01  6.426e+00  -2.846  0.00443 **
## radius_se                1.151e+01  1.865e+01   0.617  0.53723
## radius_worst             1.158e+00  2.892e+00   0.401  0.68876
## perimeter_mean           1.944e+00  8.639e-01   2.250  0.02445 *
## perimeter_se            -2.471e+00  1.629e+00  -1.517  0.12938
## perimeter_worst          1.157e-01  2.356e-01   0.491  0.62343
## area_mean                5.227e-02  3.207e-02   1.630  0.10312
## area_se                  1.732e-01  1.748e-01   0.991  0.32164
## area_worst              -5.558e-03  2.647e-02  -0.210  0.83371
## compactness_mean        -1.930e+01  3.642e+01  -0.530  0.59613
## compactness_se           1.799e+01  5.500e+01   0.327  0.74360
## compactness_worst        1.591e+00  8.796e+00   0.181  0.85648
## fractal_dimension_mean  -2.202e+02  1.577e+02  -1.396  0.16265
## fractal_dimension_se    -7.132e+02  3.859e+02  -1.848  0.06456 .
## fractal_dimension_worst  1.282e+02  6.808e+01   1.883  0.05973 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 689.984  on 520  degrees of freedom
## Residual deviance:  98.825  on 505  degrees of freedom
## AIC: 130.82
##
## Number of Fisher Scoring iterations: 10
```

```r
# now ewe predict on the test data
y_hat = predict(workfile_glm, newdata = test_workfile, type = "response")
predicted_class <- vector(length = length(y_hat))
predicted_class[y_hat > 0.5] <- "M"
predicted_class[y_hat <= 0.5] <- "B"
predicted_class <- as.factor(predicted_class)

confusionMatrix(predicted_class, test_workfile$diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 28  0
##          M  4 16
##
##                Accuracy : 0.9167
##                  95% CI : (0.8002, 0.9768)
##     No Information Rate : 0.6667
##     P-Value [Acc > NIR] : 5.163e-05
##
##                   Kappa : 0.8235
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 0.8750
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8000
##              Prevalence : 0.6667
##          Detection Rate : 0.5833
##    Detection Prevalence : 0.5833
##       Balanced Accuracy : 0.9375
##
##        'Positive' Class : B
##
```

```r
length(predicted_class)
```

```
## [1] 48
```