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  3.1.0     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     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
```

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

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

```
library(ISLR)
```

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

workfile Exploration

```
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"
```

```
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
```

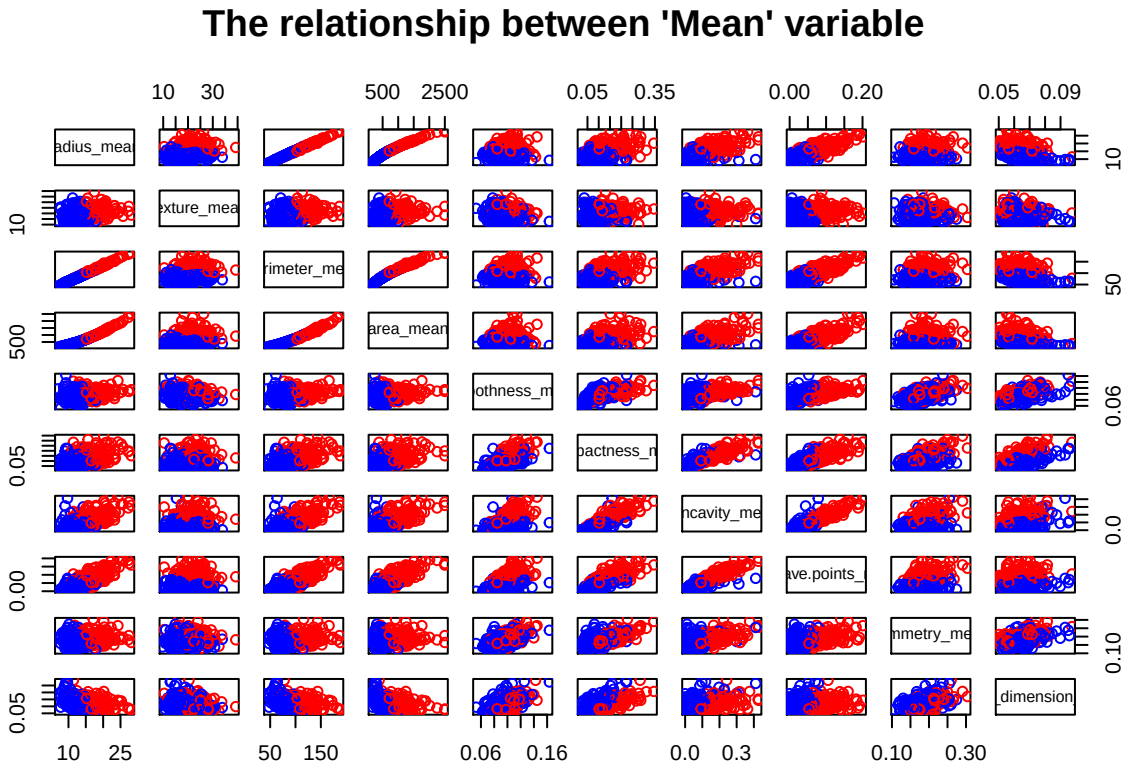
```
#mean of each varible, group by diagnosis
workfile_mean <- workfile %>% group_by(diagnosis)%>%summarise_at(vars(-id),funs(mean(., na.rm=TRUE)))
```

workfile Visualization

I. Relationship Between Variable

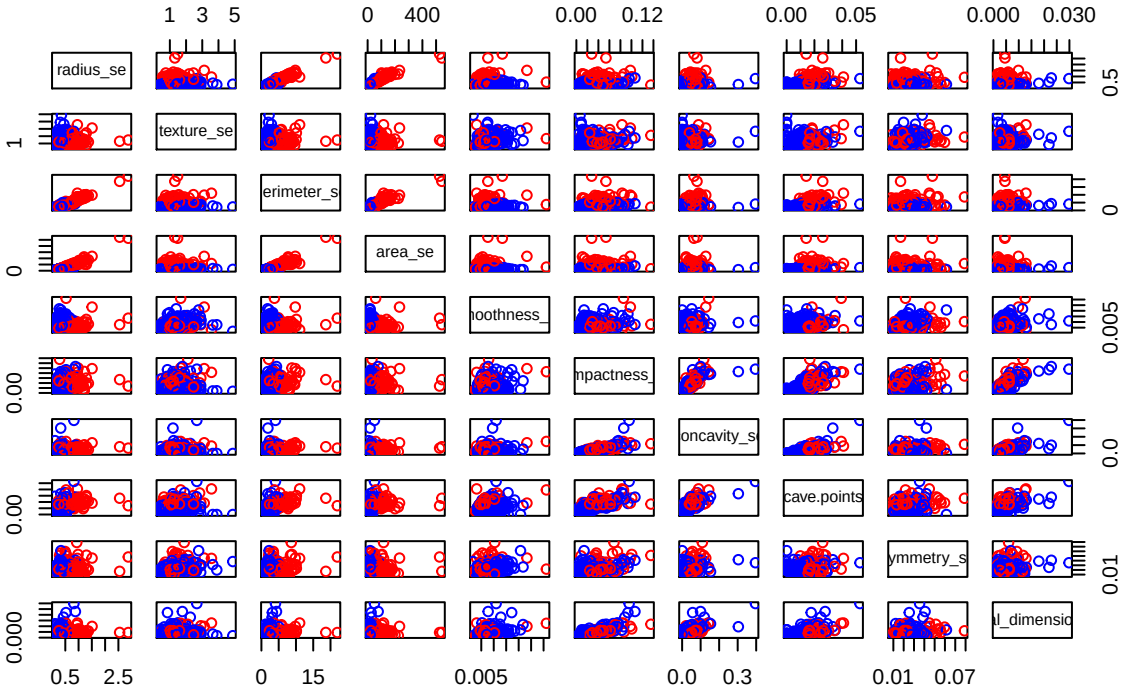
```
group = NA
group[workfile$diagnosis == "B"] = 1
group[workfile$diagnosis == "M"] = 2


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

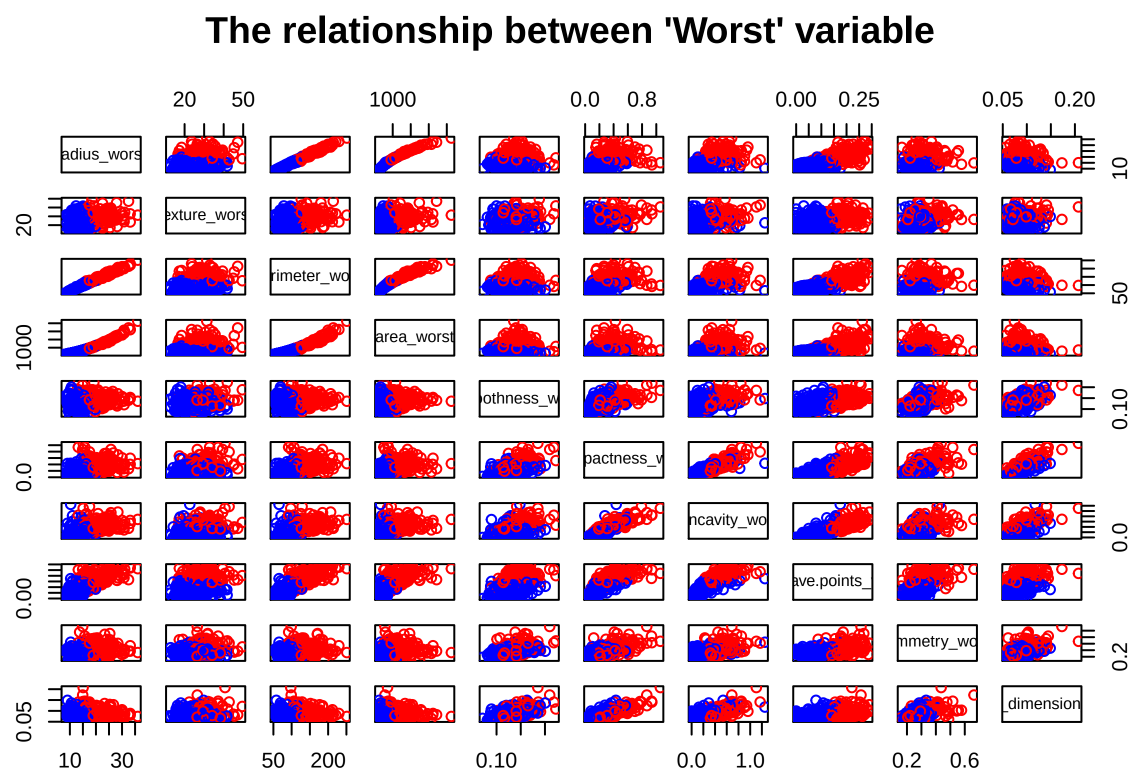



```
workfile %>%
  select_if(is.numeric) %>%
  select(contains("SE")) %>%
  pairs(col = c("blue", "red")[group], main = "The relationship between 'Standard Error' variable")
```

The relationship between 'Standard Error' variable




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



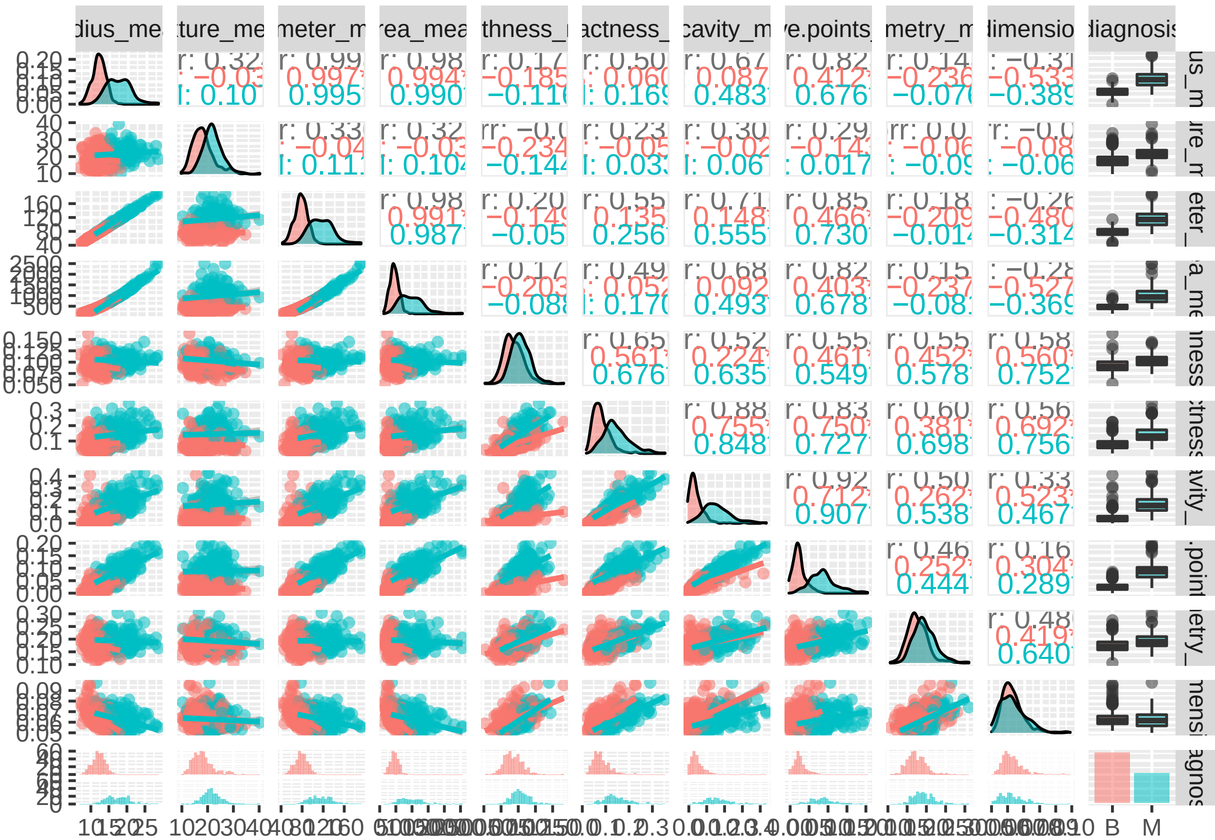
II. Differently Distributed Between Two Group of Diagnosis

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

these variables below are greatly different among 2 group of diagnosis

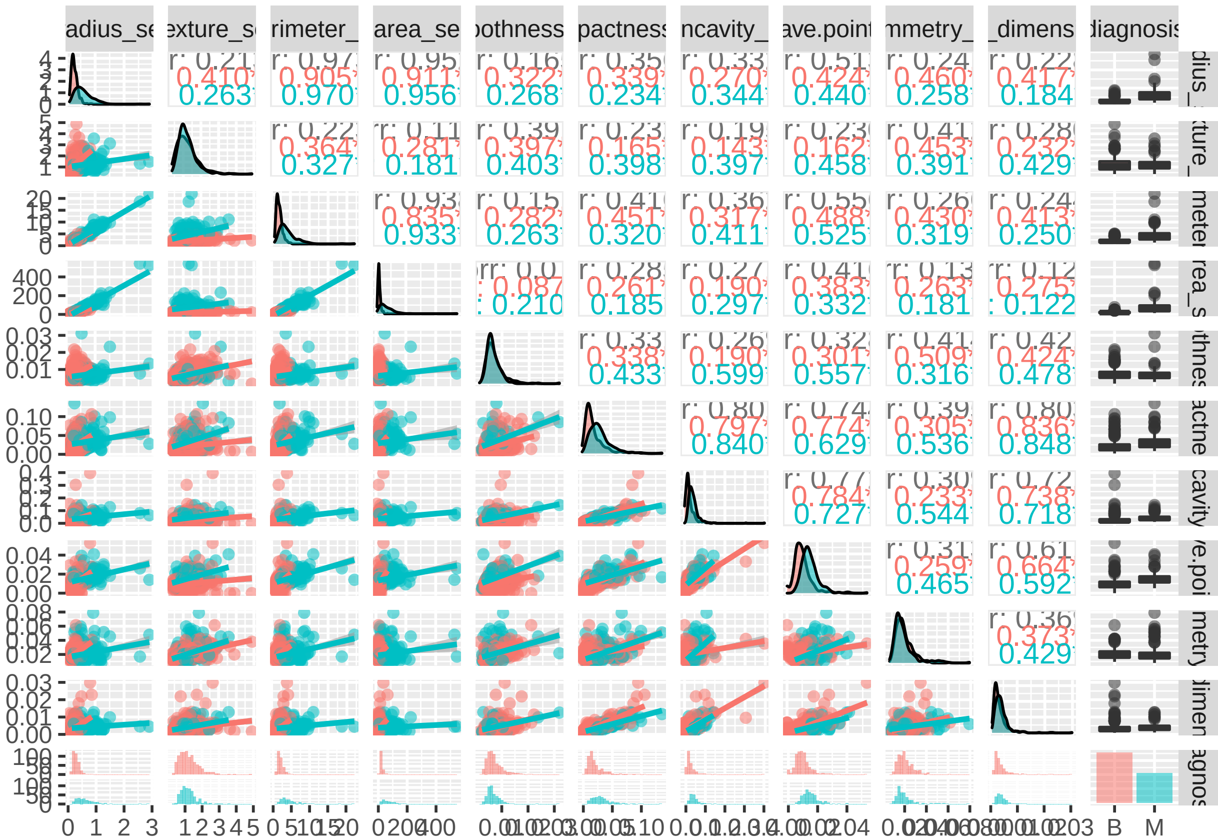
```
ggpairs(workfile[,c(3:12,2)], aes(color=diagnosis, alpha=0.5), lower=list(continuous="smooth"))
```

```
## `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`.
```



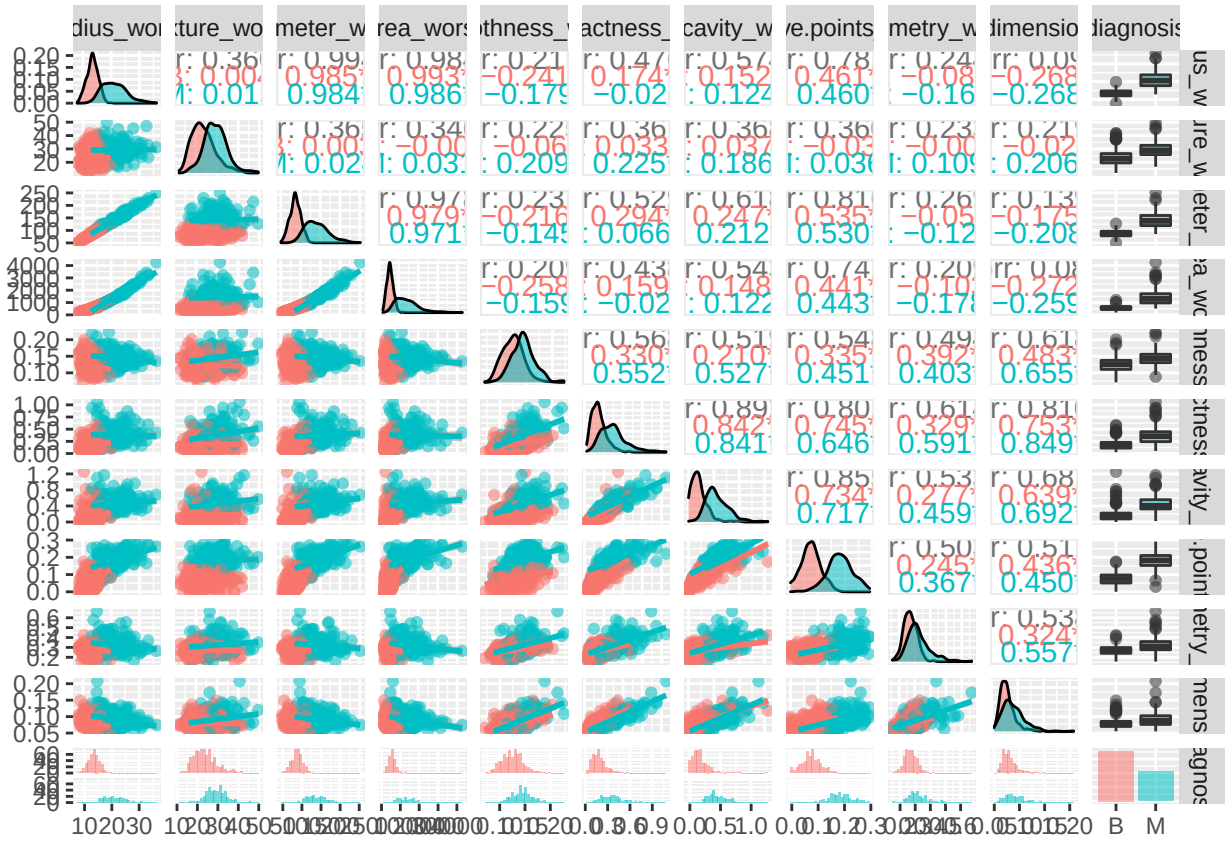
```r
ggpairs(workfile[,c(13:22,2)], aes(color=diagnosis, alpha=0.5), lower=list(continuous="smooth"))
```

```
## `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‘.
```



III. Logistic Regression Model1

```
workfile$diagnosis = as.factor(workfile$diagnosis)
temp_workfile = subset(workfile, select = -c(id))
temp_workfile = temp_workfile %>% select(diagnosis, contains("mean"))
workfile_glm = glm(diagnosis ~ ., family = "binomial", data = temp_workfile)
```

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

```
summary(workfile_glm)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = "binomial", data = temp_workfile)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.95590  -0.14839  -0.03943   0.00429   2.91690
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -7.35952   12.85259  -0.573   0.5669
```

```
## radius_mean             -2.04930    3.71588  -0.551   0.5813
## texture_mean             0.38473    0.06454   5.961  2.5e-09 ***
## perimeter_mean          -0.07151    0.50516  -0.142   0.8874
## area_mean                0.03980    0.01674   2.377   0.0174 *
## smoothness_mean         76.43227   31.95492   2.392   0.0168 *
## compactness_mean        -1.46242   20.34249  -0.072   0.9427
## concavity_mean           8.46870    8.12003   1.043   0.2970
## concave.points_mean     66.82176   28.52910   2.342   0.0192 *
## symmetry_mean           16.27824   10.63059   1.531   0.1257
## fractal_dimension_mean -68.33703   85.55666  -0.799   0.4244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 146.13  on 558  degrees of freedom
## AIC: 168.13
##
## Number of Fisher Scoring iterations: 9
```

```
y_hat = predict(workfile_glm, data = temp_workfile, type = "response")
predicted_class <- vector(length = length(y_hat))
predicted_class[y_hat > 0.5] <- "B"
predicted_class[y_hat <= 0.5] <- "M"
predicted_class <- as.factor(predicted_class)

table(predicted_class, temp_workfile$diagnosis)
```

```
##
## predicted_class   B   M
##               B  10 193
##               M 347  19
```

```
## how do we know 1 is no and 2 is yes:
levels(temp_workfile$diagnosis)
```

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

```
levels(predicted_class)
```

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

```
x <- cbind(predicted_class, temp_workfile$diagnosis)
## we sum up the logical/Boolean vector here to see how many times our model was right
sum(predicted_class == temp_workfile$diagnosis)
```

```
## [1] 29
```

```r
## divide that by the number of observations to see our accuracy
sum(predicted_class == temp_workfile$diagnosis) / length(predicted_class) * 100
```

```
## [1] 5.096661
```

```r
## Note: this is the accuracy on data used to build/create the model.
## we're going to be interested in how well the model works on NEW data.
```