Team Project

Donovan Nguyen

2/27/2021

### Initial Data Load

# Load library  
library(fastDummies)  
library(janitor)  
library('caTools')  
  
# Load data and clean  
df <- read.csv('data.csv')  
  
# Data values overview  
head(df)

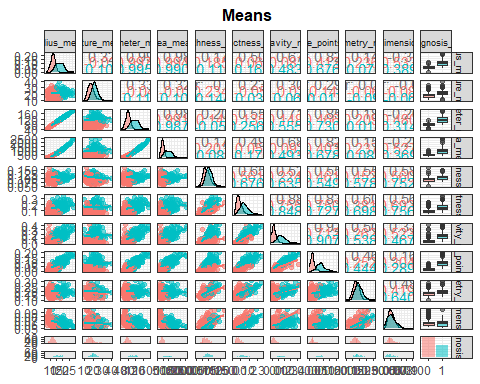
## id diagnosis radius\_mean texture\_mean perimeter\_mean area\_mean  
## 1 842302 M 17.99 10.38 122.80 1001.0  
## 2 842517 M 20.57 17.77 132.90 1326.0  
## 3 84300903 M 19.69 21.25 130.00 1203.0  
## 4 84348301 M 11.42 20.38 77.58 386.1  
## 5 84358402 M 20.29 14.34 135.10 1297.0  
## 6 843786 M 12.45 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 perimeter\_worst  
## 1 0.03003 0.006193 25.38 17.33 184.60  
## 2 0.01389 0.003532 24.99 23.41 158.80  
## 3 0.02250 0.004571 23.57 25.53 152.50  
## 4 0.05963 0.009208 14.91 26.50 98.87  
## 5 0.01756 0.005115 22.54 16.67 152.20  
## 6 0.02165 0.005082 15.47 23.75 103.40  
## area\_worst smoothness\_worst compactness\_worst concavity\_worst  
## 1 2019.0 0.1622 0.6656 0.7119  
## 2 1956.0 0.1238 0.1866 0.2416  
## 3 1709.0 0.1444 0.4245 0.4504  
## 4 567.7 0.2098 0.8663 0.6869  
## 5 1575.0 0.1374 0.2050 0.4000  
## 6 741.6 0.1791 0.5249 0.5355  
## concave.points\_worst symmetry\_worst fractal\_dimension\_worst X  
## 1 0.2654 0.4601 0.11890 NA  
## 2 0.1860 0.2750 0.08902 NA  
## 3 0.2430 0.3613 0.08758 NA  
## 4 0.2575 0.6638 0.17300 NA  
## 5 0.1625 0.2364 0.07678 NA  
## 6 0.1741 0.3985 0.12440 NA

# Dummy diagnosis variable  
df <- dummy\_cols(df, select\_columns = 'diagnosis')  
  
# Remove unneeded columns cannot be used for modeling  
df$X <- NULL  
df$id <- NULL  
df$diagnosis <- NULL  
df$diagnosis\_B <- NULL  
  
# Clean header names  
df <- clean\_names(df)  
df$diagnosis\_m <- as.factor(df$diagnosis\_m)  
head(df)

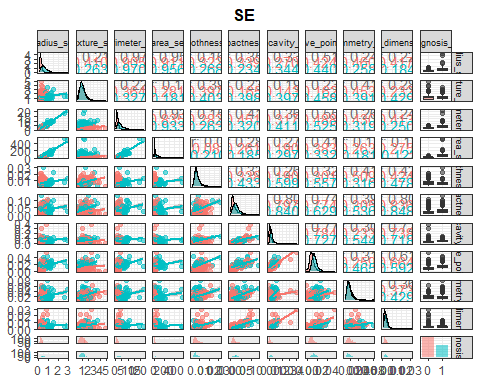
## radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean  
## 1 17.99 10.38 122.80 1001.0 0.11840  
## 2 20.57 17.77 132.90 1326.0 0.08474  
## 3 19.69 21.25 130.00 1203.0 0.10960  
## 4 11.42 20.38 77.58 386.1 0.14250  
## 5 20.29 14.34 135.10 1297.0 0.10030  
## 6 12.45 15.70 82.57 477.1 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 diagnosis\_m  
## 1 0.4601 0.11890 1  
## 2 0.2750 0.08902 1  
## 3 0.3613 0.08758 1  
## 4 0.6638 0.17300 1  
## 5 0.2364 0.07678 1  
## 6 0.3985 0.12440 1

#### Summary statistics

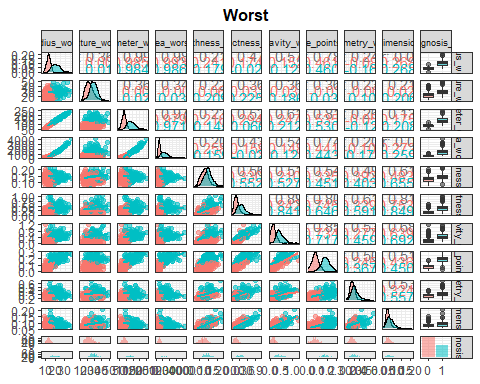
# Load library  
library(GGally)  
library(ggplot2)  
  
# Correlation matrix, density curve, and scatterplot of means  
ggpairs(df[,c(1:10,31)],   
 aes(color=diagnosis\_m, alpha = 0.5),   
 lower=list(continuous="smooth")) +   
 theme\_bw() +   
 labs(title="Means") +   
 theme(plot.title = element\_text(face='bold', color='black', hjust=0.5, size=12),  
 legend.position = "bottom",  
 strip.text = element\_text(size=6.5))



ggpairs(df[,c(11:20,31)],   
 aes(color=diagnosis\_m, alpha = 0.5),   
 lower=list(continuous="smooth")) +   
 theme\_bw() +   
 labs(title="SE") +   
 theme(plot.title = element\_text(face='bold', color='black', hjust=0.5, size=12),  
 legend.position = "bottom",  
 strip.text = element\_text(size=6.5))



ggpairs(df[,c(21:30,31)],   
 aes(color=diagnosis\_m, alpha = 0.5),   
 lower=list(continuous="smooth")) +   
 theme\_bw() +   
 labs(title="Worst") +   
 theme(plot.title = element\_text(face='bold', color='black', hjust=0.5, size=12),  
 legend.position = "bottom",  
 strip.text = element\_text(size=6.5))



#### Split data

# Split data into train and test  
split <- sample.split(df$diagnosis\_m, SplitRatio = .70)  
df.train <- subset(df, split == TRUE)  
df.test <- subset(df, split == FALSE)

### Training Model