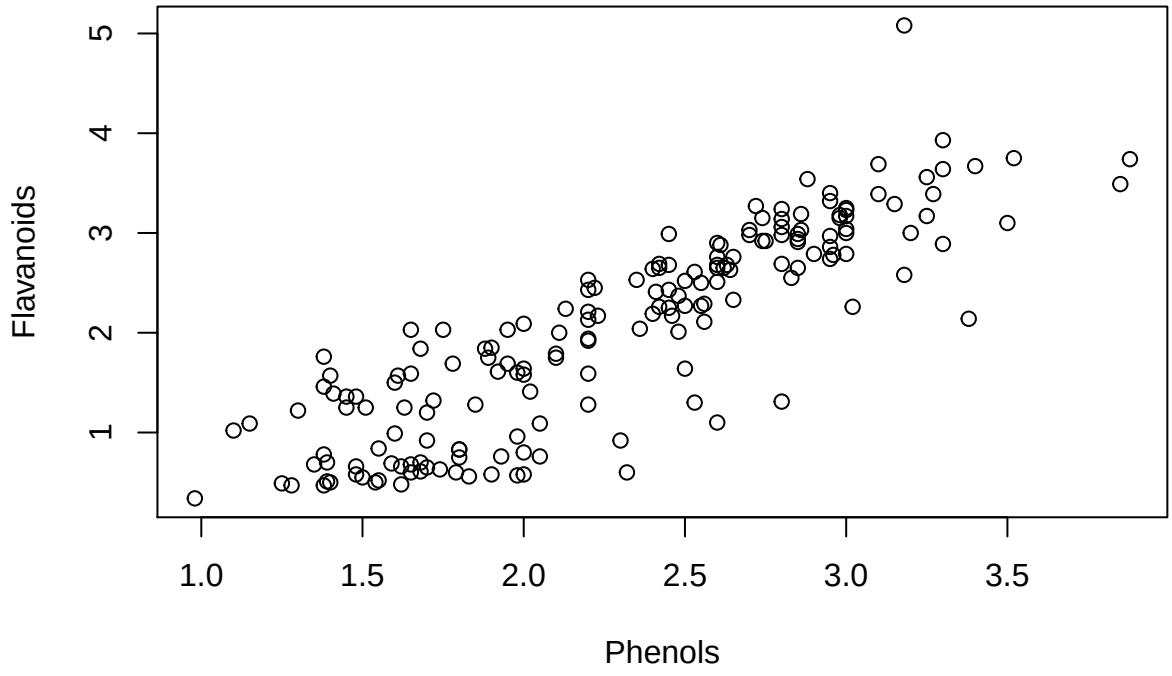
Data Science - Assignment #2

Daniel Glants I.D:203267182 / Omri Ben Menahem I.D:204048771

Question 1:

part a:

```
library(rattle.data)
data("wine")
plot(wine$Phenols,wine$Flavanoids,xlab="Phenols",ylab ="Flavanoids" )
```



We can assume from the plot that the relation between the Flavanoids and the Phenols is linear. As observed when the Phenols increase so as the Flavanoids.

part b:

We can assume that the appropriate linear model will be:
$Flavanoids_i = \beta_0 + \beta_1 \cdot Phenols_i + \varepsilon_i$

We assume that:
1. **Independence:** we assume ε are independent of everything else.
2. **Centered:** we assume that $E[\varepsilon] = 0$ meaning there is no systematic error.
3. **Normality:** we assume that $\varepsilon \sim N(0, \sigma^2)$

part c:

All the data we need is given in the wine dataset.
in order to compute $\overline{x}$ we simply sum all the Phenols value we have (Sum(column)) and then we divide the result by the number of observations we've got. Same goes for computing the $\overline{y}$ we sum the Flavanoids values and dividing by the amount od y observations.

Afterwards we can compute $\hat{\beta}_1$ by simply preforming the formula that is given: $\hat{\beta}_1 = (\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})^2}) =$

At last after computing $\overline{x}$, $\overline{y}$ and $\hat{\beta}_1$ we will compute the intercept $\hat{\beta}_0$ with the:$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \cdot \overline{x}$

In order for those computations to be Valid important assumptions should be made about the data we're given.

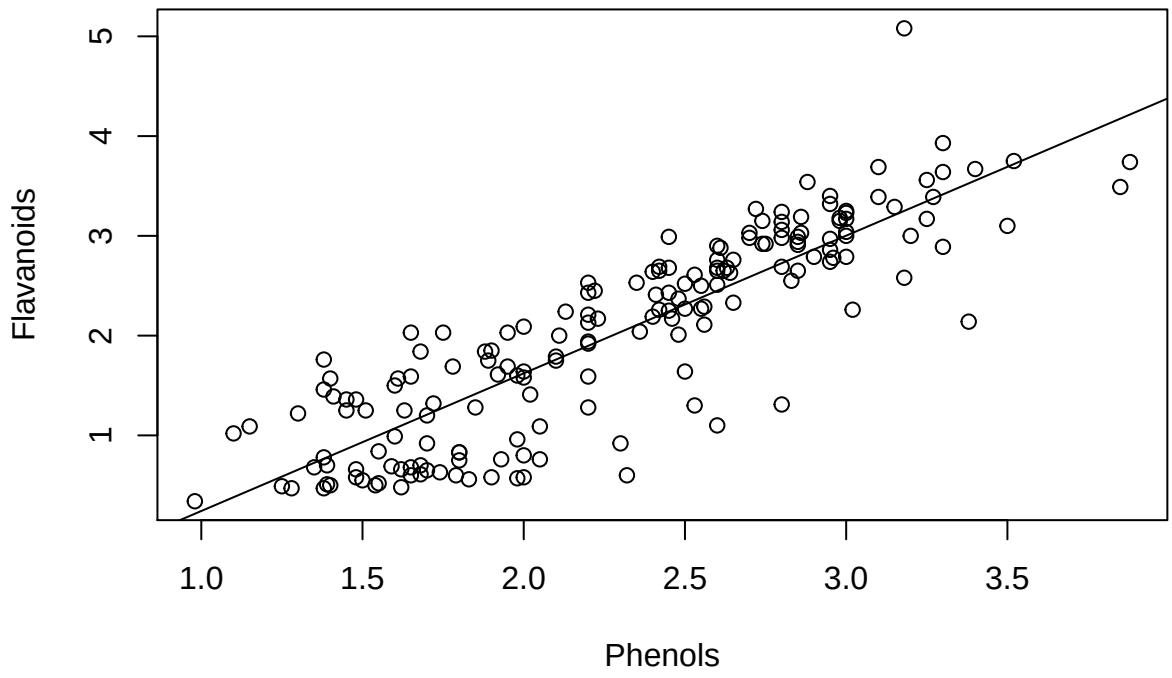
The assumtion of **Normality**, that $\varepsilon \sim N(0, \sigma^2)$, if the error isn't distributes normally across the dataset we cannot exclude it from the $\hat{\beta}_1$ equation and thus rendering all our computations as faulty.

Another assumption that is derivative from the first is about the correlation (ρ) between ε_i and x_i. if there is a correlation between the two of them, the above computations are false.

part d:

```
lm_wine <- lm(Flavanoids~Phenols, data = wine)

plot(wine$Phenols,wine$Flavanoids,xlab="Phenols",ylab ="Flavanoids" )
abline(lm_wine)
```



The estimation results are:

```
coefficients(summary(lm_wine))
```

```
##              Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) -1.137627 0.14378958 -7.91175 2.712103e-13
## Phenols      1.379844 0.06045513 22.82426 1.755839e-54
```

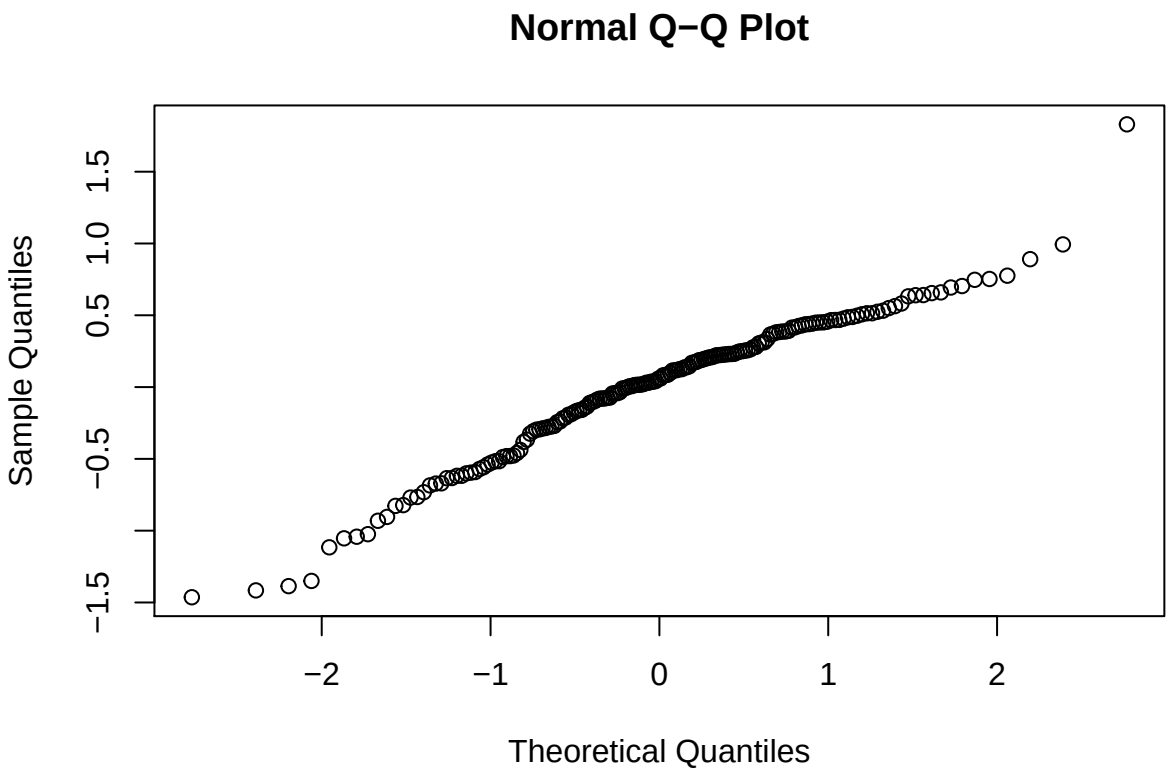
part e:

what is the meaning of the slope's coefficient? it means that **On Avarege** the "Flavanoids" units will increase by 1.379844 as a result of increasing the "Phenols" unit by 1.

Is the Estimate result is significant? we can see that the t values of the hypothesis that $\hat{\beta}_1 = 0$ is very high and the P-value of making a mistake in this hypothesis is very low 1.755839e-54, thus we can infer that the estimators result is significant.

part f:

```
qqnorm(resid(lm_wine))
```



we can see from the qqplot that most of the errors main centered around the 0 as one can expect from anormal distribution. thus, we can conclude that the assumption of Normality of the errors is correct.

part g:

lets compute $\hat{\beta_1}$: for us to do that we initially need to compute $\overline{x}$ and $\overline{y}$

```
X_roof <- sum(wine$Phenols)/nrow(wine)
Y_roof <- sum(wine$Flavanoids)/nrow(wine)
paste("X_roof= ",X_roof)
```

```
## [1] "X_roof=  2.29511235955056"
```

```
paste("Y_roof= ",Y_roof)
```

```
## [1] "Y_roof=  2.02926966292135"
```

```
numerator_b1 <- sum((wine$Phenols-X_roof)*(wine$Flavanoids-Y_roof))
denominator_b1 <- sum((wine$Phenols-X_roof)^2)
beta1_hat <- numerator_b1/denominator_b1
paste("beta1_hat=",beta1_hat)
```

```
## [1] "beta1_hat= 1.37984391402326"
```

lets compute $\hat{\beta}_0$:

```
beta0_hat <- Y_roof- beta1_hat*X_roof
paste("beta0_hat=",beta0_hat)
```

```
## [1] "beta0_hat= -1.13762715840406"
```

lets compute RSS $\rightarrow \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e^2$

```
e_sqr <- residuals(lm_wine)^2
print(sum_e_sqr <- sum(e_sqr))
```

```
## [1] 44.59583
```

lets compute the Coefficient of determination R^2 using the function introduced in the Course's Notebook

```
R2 <- function(y, y.hat){
  numerator_R_sqr <- (y-y.hat)^2%>%sum()
  denominator_R_sqr <- (y-mean(y))^2%>%sum()
  return(1-numerator_R_sqr/denominator_R_sqr)
}

R2(y=wine$Flavanoids, y.hat=predict(lm_wine))
```

```
## [1] 0.74747
```

Now we can compare our results to the result we get from the lm() object:

```
paste("beta0_hat: ",coefficients(lm_wine)[1])
```

```
## [1] "beta0_hat:  -1.13762715840406"
```

```
paste("beta1_hat: ",coefficients(lm_wine)[2])
```

```
## [1] "beta1_hat:  1.37984391402326"
```

Also, when Inspecting the rest of the lm's summary we can see that all of the other estimators we computed are identical (Except for the RSS wich the lm object do not provide)

```
summary(lm_wine)
```

```
##
## Call:
## lm(formula = Flavanoids ~ Phenols, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.46361 -0.28305  0.05922  0.37011  1.82972
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.13763    0.14379  -7.912 2.71e-13 ***
## Phenols      1.37984    0.06046  22.824  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5034 on 176 degrees of freedom
## Multiple R-squared:  0.7475, Adjusted R-squared:  0.746
## F-statistic: 520.9 on 1 and 176 DF,  p-value: < 2.2e-16
```

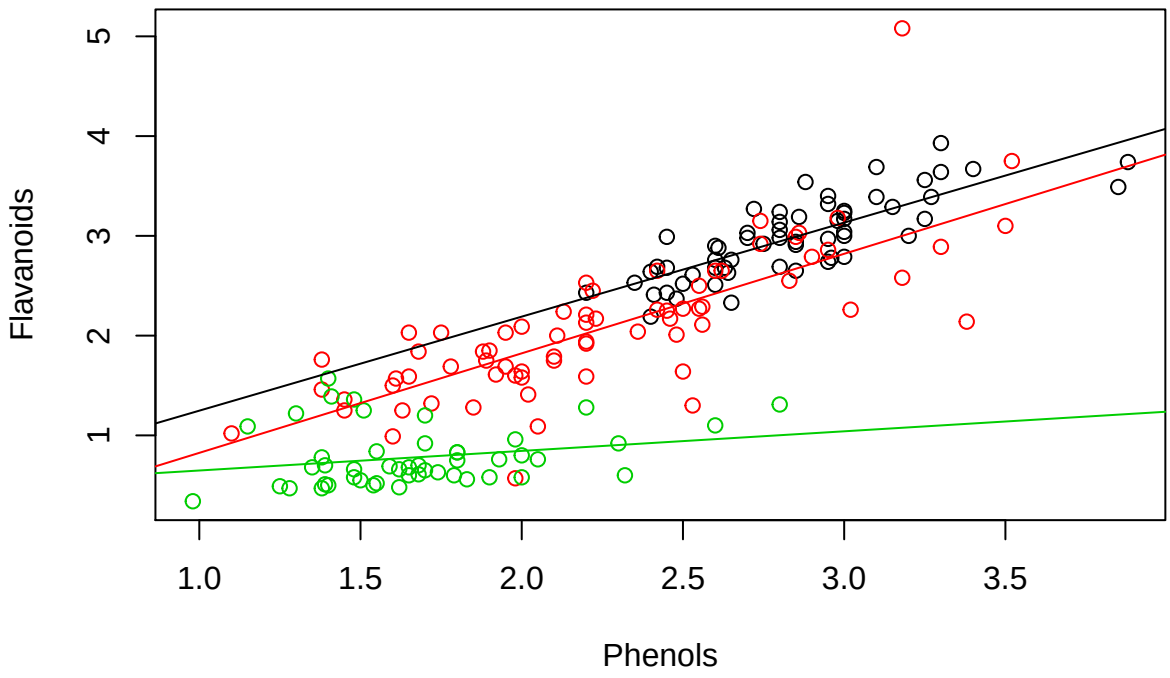
Part h+i:

```
wine$Type <- as.factor(wine$Type)
levels(wine$Type)
```

```
## [1] "1" "2" "3"
```

```
lm_wine_Type <- lm(Flavanoids~Phenols*Type, data=wine)


plot(Flavanoids~Phenols,xlab="Phenols",ylab ="Flavanoids",col = Type,data = wine)
abline(lm_wine_Type$coefficients[1:2],col=1)
abline(lm_wine_Type$coefficients[1]+lm_wine_Type$coefficients[3],
       lm_wine_Type$coefficients[2]+lm_wine_Type$coefficients[5],col=2)
abline(lm_wine_Type$coefficients[1]+lm_wine_Type$coefficients[4],
       lm_wine_Type$coefficients[2]+lm_wine_Type$coefficients[6],col=3)
```



Part j: The estimators coefficients are:

```
lm_wine_Type$coefficients
```

```
##   (Intercept)       Phenols         Type2         Type3 Phenols:Type2
##    0.30527782    0.94258285   -0.47806091    0.14641141    0.05509516
## Phenols:Type3
##   -0.74614556
```

That means that the "(Intercept)" and the "Phenols" are the $\hat{\beta}_0$ and $\hat{\beta}_1$ for wines from type 1.

when we calculate: (Intercept)+Type2 we get the Interceptor ($\hat{\beta}_0$) for type 2. same when we calculate: "Phenols" + "Phenols:Type2" we get the slope ($\hat{\beta}_1$) for type 2.

when we calculate: (Intercept)+Type3 we get the Interceptor ($\hat{\beta}_0$) for type 3. same when we calculate: "Phenols" + "Phenols:Type3" we get the slope ($\hat{\beta}_1$) for type 3.

That means that for every type of wine an increase in 1 unit of phenols, is correlated in increase in ($\hat{\beta}_1$) units of Flavanoids, **in average**.

Question 3:

part a:

```r
wine$is_1 <- ifelse(wine$Type==1,1,0)

glm_wine <- glm(is_1~Alcohol+Ash+Magnesium+Phenols+Flavanoids,data = wine,family = binomial)
```

Model description: We have chosen to perform a logistic regression upon "is_1" variable which determines either the wine is Type 1 or not, and we chose the predictors to be the: Alcohol, Ash, Magnesium, Phenols and Flavanoids levels. when everyone except magnesium are continuous variables, and magnesium is discrete variable. luckily for us the logistic regression can handle both kinds.

the logistic regression retrieves the contribution of any variable to the odds of the result being either 0 or 1.

We assume that the variable is_1 distributes bionomically when the *Odd* for getting 1 is P and the *Odd* of getting 0 is 1-p, we use the *Odds Ratio* to measure the two Bernoulli instances because the *Odds Ratio* have better mathematical properties than other candidate distance measures.

There for the link function for odds ratio is $\frac{p}{1-p}$ thus allowing us to use the logistic model assumptions and conclude that the is_1 variable distribution is $is\_1_i|x' \sim Binom(1, p = \frac{e^{x'\cdot\beta}}{1+e^{x'\cdot\beta}})$ thus, the *Odds Ratio* function allows us to interpret β as a measure of change of binary random variables due to a unit increase in x'

part b:

An increase of 1 unit in one of the x'_i veriabls (Alcohol, Ash, Magnesium, Phenols or Flavanoids) is correlated with change of β_i in the log *Odds Ratio* of the wine to be of type 1.

part c:

```r
yhat_glm <- predict(glm_wine,type = "response")
yhat_glm_binar <- ifelse(yhat_glm>0.5, 1, 0)*1
print("The confusion matrix for the classification:")
```

```
## [1] "The confusion matrix for the classification:"
```

```r
(CM <- table(true= wine$is_1, predicted = yhat_glm_binar))
```

```
##     predicted
## true   0   1
##    0 114   5
##    1   4  55
```

```r
Precision <- CM[4] / sum(CM[,2])
Recall <- CM[4] / sum(CM[2,])

paste("the Precision rate is: ",Precision)
```

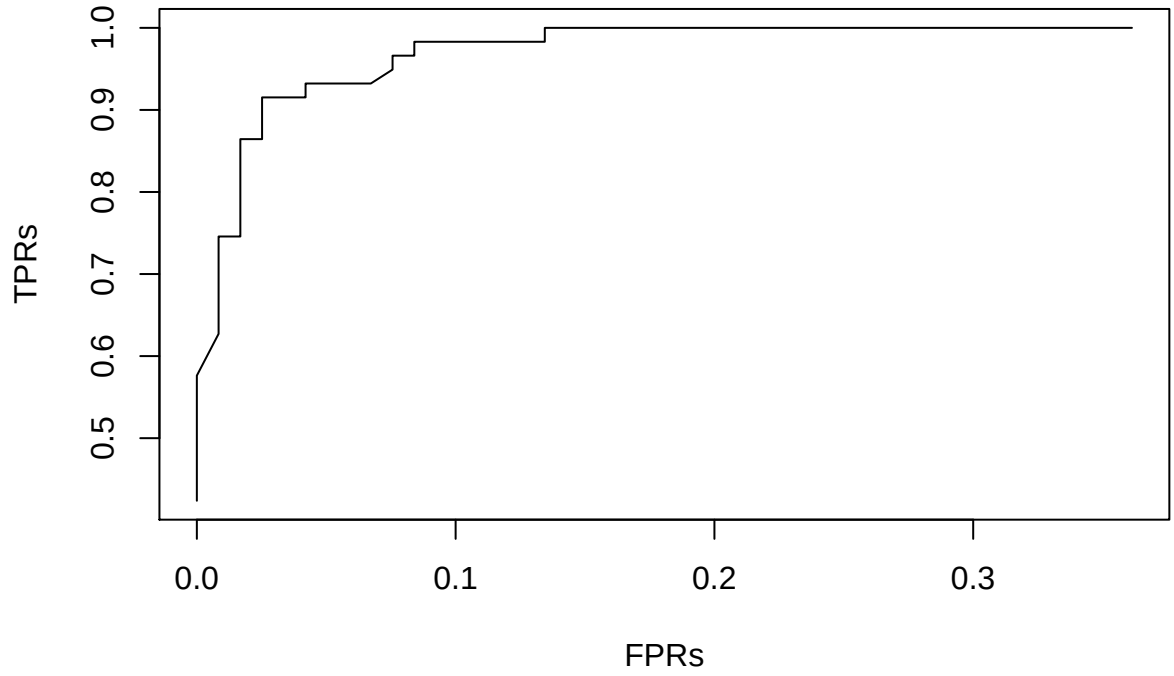
```
## [1] "the Precision rate is:  0.916666666666667"
```

```r
paste("the Recall rate is: ",Recall)
```

```
## [1] "the Recall rate is:  0.932203389830508"
```

part d:

```r
alphas <- seq(0,1,0.01)
TPRs <- numeric(length(alphas))
FPRs <- numeric(length(alphas))
for (i in seq_along(alphas)) {
  pr_i <- ifelse(yhat_glm>alphas[i],1,0)
  CM_i <- table(wine$is_1, pr_i)
  TPRs[i] <- CM_i[4] / sum(CM_i[2,])
  FPRs[i] <- CM_i[3] / sum(CM_i[1,])
}
plot(TPRs~FPRs, type = "l")
```



The definition of "best" threshold depends on the subject of our test. if we give significance value to "Positive" result we'd allow our model to give **more** False Positives, as long that we would not miss many true positives (as explained in class for the hospital patients example, we would prefer dispatch a doctor for a False positive case, rather than miss an actual emergency.)

In our exercise we think it would be sufficient for a 0.95 TPR. meaning athreashold should be around $\alpha = 28\%$

```
rn_TPRs <- round(TPRs,2)
threshold <- (which(0.95 == rn_TPRs)[1]*0.01)%>%print
```

```
## [1] 0.28
```

part e:

A common technich for multi-class classification is using the One Vs Rest/(All) technique.
Basically what we have to do is first to classify our y_i values into separate types (i.e. Type 1, Type 2, Type 3 . . . Type n).
Then, using our training data we will run **n** logistic regression with binary distributions when each time a different Type will be set as TRUE and the rest as False.

After we will get all the regression models and would want to predict from new data, we will run all the models on the new data ,which in turn, each of them will generate a different **Odd** for each y_i to be any of the Types.

For each instance of y_i we will determine his type by the regression result with the largest Odd.

Mathematically speaking the model looks pretty much like this: $h_\theta^{(i)}(x) = P(y = i|x;\theta)$ $(i = 1, 2, 3..n)$