1) Dataset illustration and goal of the analysis

The data regards Celiac Disease, an autoimmune disease that occurs in genetically predisposed people where the ingestion of gluten leads to damage in the small intestine. The dataset is available on Kaggle and is originally taken from the Wageningen University & Research Biotechnology Department.

The goal of this assignment is to understand how IgA, a crucial antibody in the diagnosis of Celiac Disease, is affected by the most-known symptoms of the disease and by the fact that a person has been diagnosed as Celiac or not, in order to be able to make predictions regarding possible IgA values. Although only 1 in 400 to 1 in 800 people in the general population have IgA deficiency, 2% to 3% of people with Celiac Disease have IgA deficiency. In my personal case, this antibody among all the symptoms has been crucial in my diagnosis, therefore I decided to perform this analysis.

Furthermore, I decided to perform data cleaning on the original dataset, removing some of the variables, since based on clinical evidence, they were not highly related with the analysis. In order to pursue my goal, the dataset contains the following variables:

- *Age*: it indicates the age of the people who participated to the research, measured in years.
- *Gender*: it indicates the gender of the people who participated to the research (i.e. Male or Female).
- *Diabetes*: people affected by Diabetes (i.e Yes) or not (i.e. No).
- *Diarrhoea*: it indicates the different types of diarrhoea (i.e. fatty, watery, inflammatory).
- *Abdominal*: presence of abdominal pain (i.e Yes) or absence (i.e No).
- *Short_Stature*: people affected by different types of short stature problems (i.e Variant, PSS, DSS).
- *IgA*: antibody that protects the mucosal surfaces (e.g. digestive tracts), measured in (g/L).
- *IgG*: most abundant antibody, providing long-term immunity, measured in (g/L).
- *IgM*: first antibody produced in response to infections, measured in (g/L)
- *Marsh*: it is a system for grading intestinal damage in Celiac Disease, from normal (i.e Marsh type 0) to severe condition (i.e Marsh type 3)
- *Disease_Diagnose*: it indicates if to a person has been diagnosed the disease (i.e Yes) or not (i.e No)

```
data = read.csv("cleaned_data.csv")
head(data)
```

```
##   Age Gender Diabetes    Diarrhoea Abdominal Short_Stature  IgA  IgG  IgM
## 1  10   Male      Yes inflammatory       yes           PSS 1.30 10.0 1.00
## 2   9   Male      Yes        fatty       yes           PSS 1.50 12.5 1.30
## 3   8 Female      Yes       watery       yes       Variant 0.40  8.0 0.50
## 4  10   Male      Yes       watery       yes           PSS 0.98  9.0 0.66
## 5   9   Male      Yes        fatty       yes           PSS 1.00 10.5 1.10
## 6   8 Female      Yes        fatty       yes       Variant 1.10  9.5 1.00
##           Marsh Disease_Diagnose
## 1  marsh type 0              yes
## 2 marsh type 3a              yes
## 3  marsh type 1              yes
## 4 marsh type 3a              yes
## 5  marsh type 1              yes
## 6 marsh type 3a              yes
```

1.2) Exploratory data analysis

I decided to begin the exploratory data analysis with a graphical representation of the response variable IgA, for the purpose of understand the distribution of the IgA level:

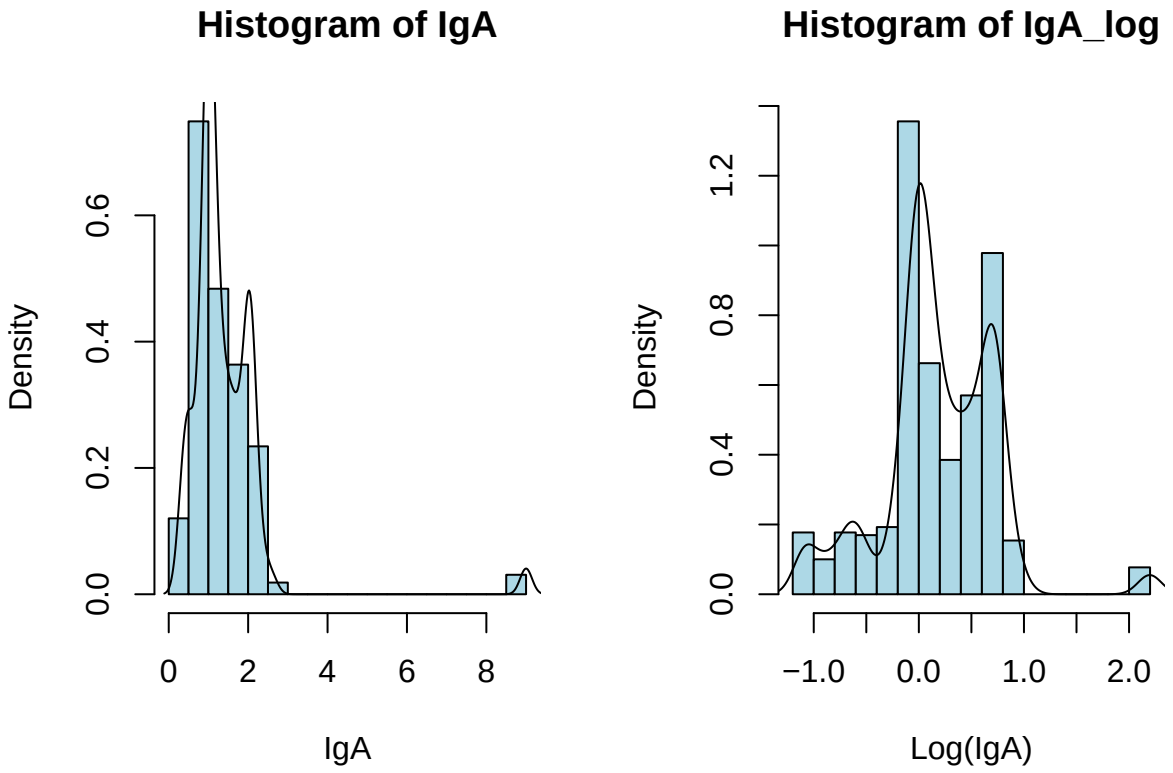


Figure 1: Histogram of IgA and IgA_log

Looking at the histogram of raw IgA (plot on the left, Figure 1), the distribution is right-skewed, which is expected since IgA levels range between $[0, +\infty)$. To reduce skewness and improve interpretability, a logarithmic transformation was applied.

The transformation reduced right-skewness, making the distribution more symmetric. However, the resulting distribution now presents multimodality, consequently further investigation is needed to determine whether this pattern is due to underlying categorical variables or other data characteristics.

Afterwards, it is crucial to consider the quantitative variables in order to test for their correlation.

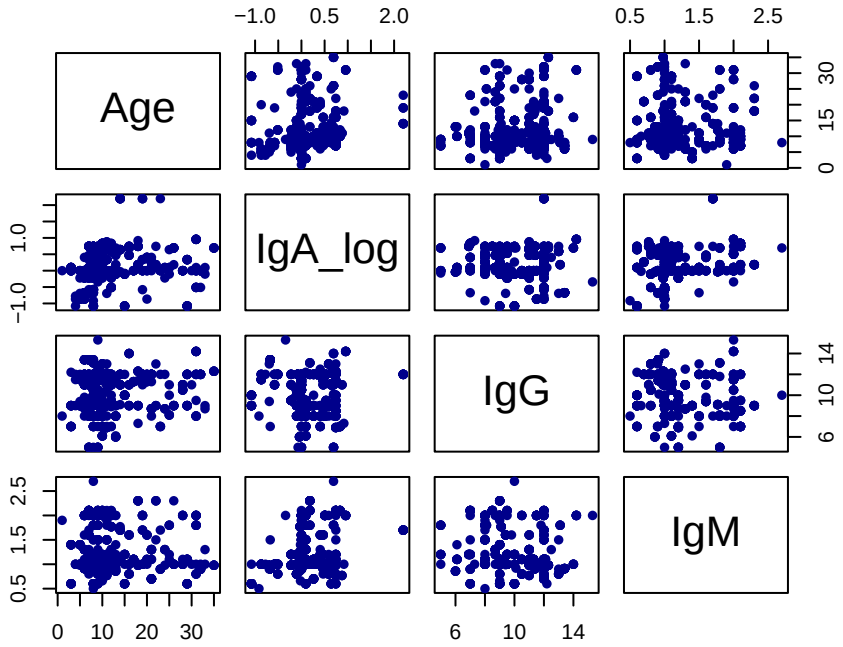


Figure 2: Correlation between variables

Looking at Figure 2, it is possible to denote a weak correlation between IgA and the continuous variables and this is reasonable since, based on clinical evidence, IgA levels do not change substantially with age and IgG, IgM and IgA_log are different classes of immunoglobulins that serves distinct immune functions. It is always a best practice to check for high correlation since, as it will be shown in the following sections, it can cause problems of collinearity.

Let's now consider qualitative predictors: some of them are clinically crucial in affecting Celiac Disease, but they need to be evaluated in terms of IgA levels, so a graphical representation of the variables categorical variables is needed.

Looking at Figure 3:

- Overall, there is no substantial difference in IgA_log levels across most categorical variables, except for Marsh classification.
- In Marsh case a general increasing trend in IgA_log is observed as the Marsh classification progresses, with Marsh type 0 (no damage) showing the lowest median and Marsh 3b reaching the highest levels. However, Marsh 3c exhibits a slight decline, suggesting variability in severe cases and lastly the None category displays a regular range of IgA values compared to the other levels.
- Extreme values (red dots) can be denoted in all the plots, indicating potential outliers or influential points. Further investigation will be done during the diagnostic.

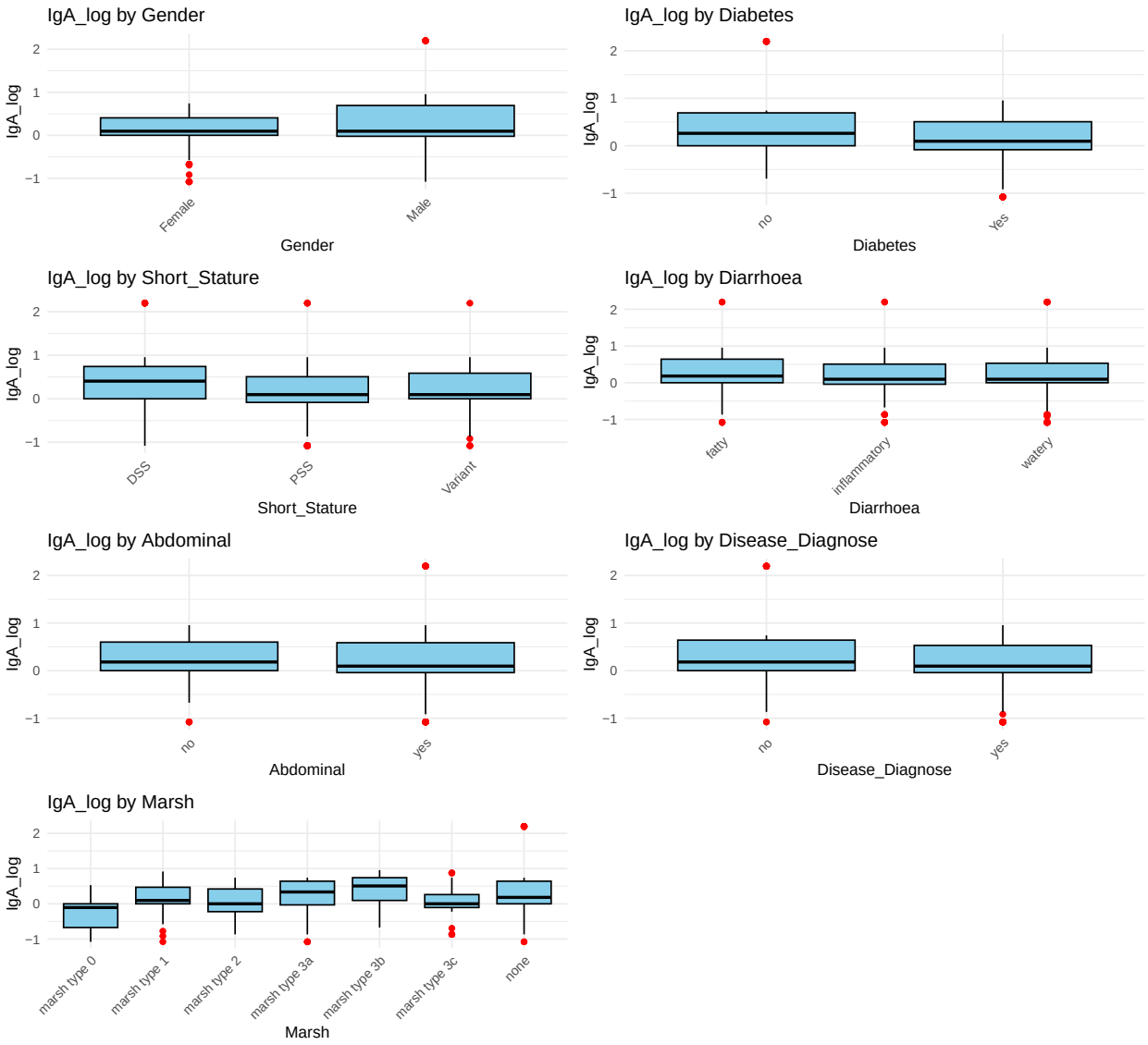


Figure 3: Boxplots of the categorical variables

2) Variable selection

In order to perform variable selection, we can use *regsubsets()* function that is included in the *leaps* library. Using this function it is possible to compare various models with different numbers of predictors. I decided to use the best subset selection and since by default *regsubsets()* function allows a maximum of 8 predictors in the selected models, I set *nvmax* equal 17 to address the issue.

```
best_sub = regsubsets(IgA_log ~., data = data, nvmax = 17)
summ=summary(best_sub)
```

3) Best model according to AIC, BIC, Adjusted R squared and Mallow's Cp

The best models according to the different criteria can be seen in Figure 4:

```
par(pty="s",mfrow=c(1,4),mar=c(2,1,2,1))
# AIC
p = 17
```

```r
n = nrow(data)
aic = matrix(NA,p,1)
for(i in 1:p){
  aic[i] = summ$bic[i]-(i+2)*log(n)+2*(i+2)}
plot(aic, type = "b", pch = 19, xlab = "Number of predictors", ylab = "", main = "Drop in AIC");
abline(v=which.min(aic), col=2, lty=2)
# BIC
plot(summ$bic, type = "b", pch = 19, xlab = "Number of predictors", ylab = "", main = "Drop in BIC");
abline(v=which.min(summ$bic), col=2, lty=2)
# Adjusted R squared
plot(summ$adjr2, type = "b", pch = 19, xlab = "Number of predictors", ylab = "", main = "Adjusted R^2")
abline(v=which.max(summ$adjr2), col=2,lty=2)
# Mallow's Cp
plot(summ$cp, type = "b", pch = 19, xlab = "Number of predictors", ylab = "", main = "Mallow's Cp");
abline(v=which.min(summ$cp), col=2, lty=2)
```

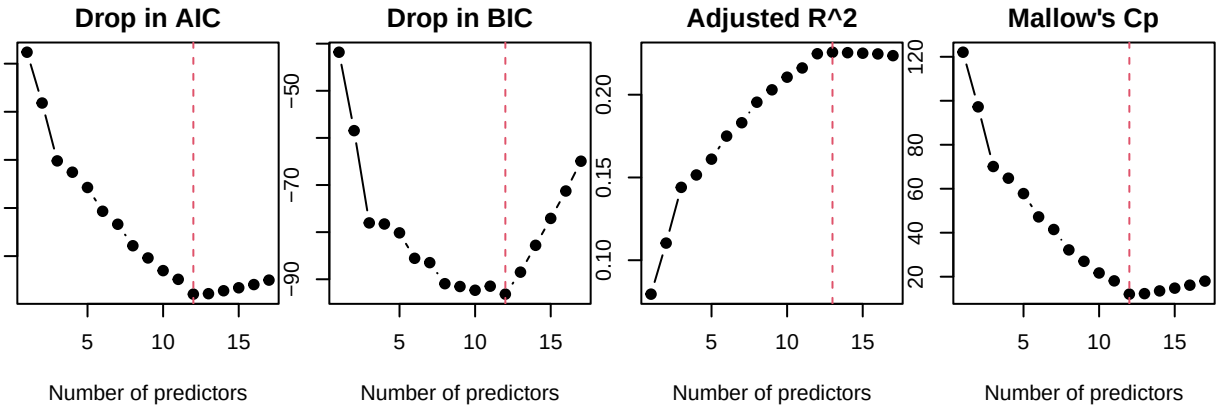


Figure 4: Best model according to AIC, BIC, Adjusted R^2 and Cp

3.1) Best model according to Cross validation

This approach applies 10-fold cross-validation to evaluate models of different sizes using best subset selection in *regsubsets()*. Each observation is randomly assigned to one of k = 10 folds. For each fold, subset selection is performed on the training set (k-1 folds), and test errors are computed on the left-out fold.

```r
p = 17
k = 10
set.seed(456)
folds = sample(1:k, nrow(data), replace = TRUE)
cv.errors = matrix(NA, k, p, dimnames=list(NULL, paste(1:p)))
```

The cross-validation error for each model size is stored in the 10x17 matrix *cv.errors*, allowing to determine the optimal number of predictors based on mean squared error (MSE).

```r
for(j in 1:k){
  best.fit =regsubsets (IgA_log ~ ., data=data[folds!=j,], nvmax = 17)
  for(i in 1:p) {
    mat = model.matrix(as.formula(best.fit$call[[2]]), data[folds==j,])
    coefi = coef(best.fit ,id = i)
    xvars = names(coefi)
    pred = mat[,xvars ]%*% coefi
```

```
    cv.errors[j,i] = mean( (data$IgA[folds==j] - pred)^2)
  }
}
```

Furthermore, let's use the *colMean()* function to compute the average cross-validation error across all folds for each model size. This results in a vector where the j-th element represents the cross-validation error for the model with j predictors. Figure 5 shows the cross-validation error for the best model at each size, helping identify the optimal number of predictors.

```
cv.mean = colMeans(cv.errors)
plot(cv.mean ,type="b",pch=19, xlab="Number of predictors", ylab="CV error");
abline(v=which.min(cv.mean), col=2, lty=2)
```

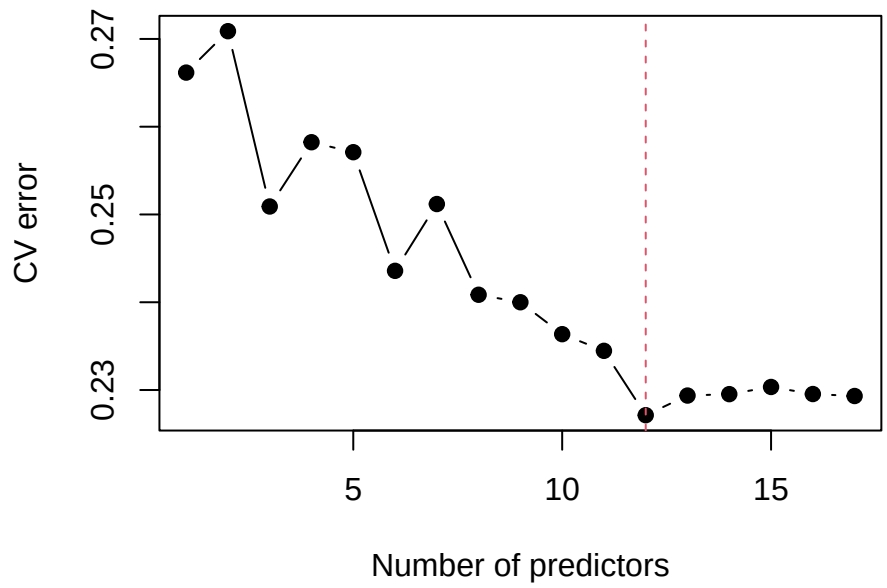


Figure 5: Best model according to Cross-validation

3.2) Best overall model

All model selection procedures identified the model with 12 predictors as the optimal choice. This agreement across multiple methods provides strong evidence that this model achieves the best trade-off between predictive accuracy and complexity.

Therefore, the model that will be used is the following:

```
ols=lm(IgA_log ~ Age + Gender + Diabetes + Short_Stature + IgM + Marsh, data = data)
```

4) Potential collinearity issues

Identify potential collinearity issues is crucial in the regression context, since it can be difficult to separate out the individual effects of collinear variables on the response. Furthermore, it leads to imprecise estimate of β.

Collinearity can be detected through the examination of the *Correlation Matrix* of the predictors, but unfortunately, not all collinearity problems can be detected by inspection of it: it is possible for collinearity

to exist between three or more variables even if no pair of variables has a particularly high correlation. This is the so called *multicollinearity*

Hence, a better way to discover collinearity is to compute the *Variance Inflation Factor* (VIF). The VIF is the ratio of the variance of $\hat{\beta}_j$ when fitting the full model divided by the variance of $\hat{\beta}_j$ if fit on its own and it can be computed in the following way:

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j}|X - j}$$

Where $R^2_{X_j}|X - j$ is the R^2 obtained from a regression of X_j of all the other predictors. If $R^2_{X_j}|X - j$ is close to 1, then collinearity is present and so the VIF will be large. The smallest possible value for VIF is 1, which indicates the complete absence of collinearity. As a rule of thumb, a VIF>10 indicates a problematic amount of collinearity.

```
vif(ols)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## Age            1.120692  1        1.058627
## Gender         1.077833  1        1.038187
## Diabetes       1.965616  1        1.402004
## Short_Stature 1.166331  2        1.039215
## IgM            1.152363  1        1.073482
## Marsh          2.331797  6        1.073102
```

Due to the presence of at least one categorical variables with more than 2 levels (in this case *Short_Stature* and *Marsh*), the *Generalized Inflation Factor* (GVIF) is adopted, that is an extension of VIF. Furthermore raw GVIF values tend to be inflated compared to standard VIF due to the additional degrees of freedom introduced by categorical variables. To adjust for this inflation and allow for direct comparison with standard VIF, Fox & Weisberg (2011) suggest using the transformation:

$$GVIF^{\frac{1}{2Df}} = \left(\frac{1}{1 - R^2_{X_j|X_{-j}}}\right)^{\frac{1}{2Df}}$$

where Df represents the degrees of freedom (i.e. the number of levels minus one). This adjustment makes this quantity analogous to $\sqrt{VIF}$, ensuring that the usual VIF threshold remain applicable when checking for collinearity.

Since none of the variables present high values of the GVIF, no problems of collinearity are detected.

5) Diagnostics

Since the erros are not observable, the assumptions will be discussed on the residuals, that are computable.

5.1) Constant variance and linearity

In order to check for these assumptions, we can look at the residuals (y-axis) versus fitted values (x-axis) plot shown in Figure 7.

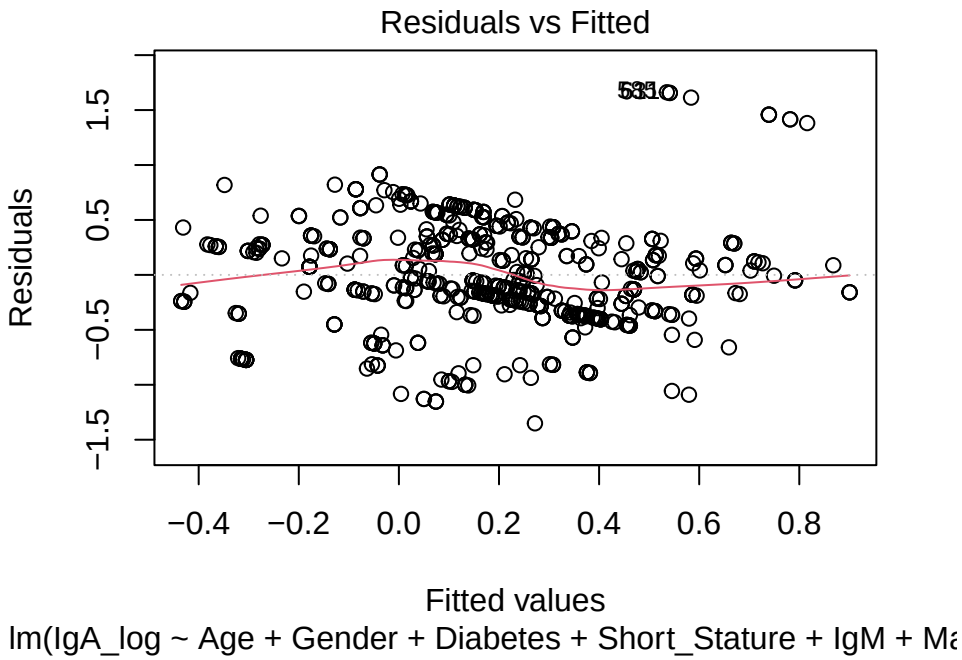


Figure 6: Residuals versus fitted for ols

The ideal situation should be the following:

- In case of *homoschedasticity*, the points should be randomly located around zero, resulting into no clear pattern.
- In case of *linearity*, the red smooth line that is automatically generated by the R commander function, should be mostly horizontal around 0 (i.e. dashed line at the residuals with zero mean), without systematic trends.

Looking at Figure 6 it is possible to denote signs of both non-constant variance and non-linearity since residuals tend to spread slightly more for larger fitted values. Furthermore, the red line is not flat since it surpassed zero.

5.2) Remedies for non-constant variance and non-linearity

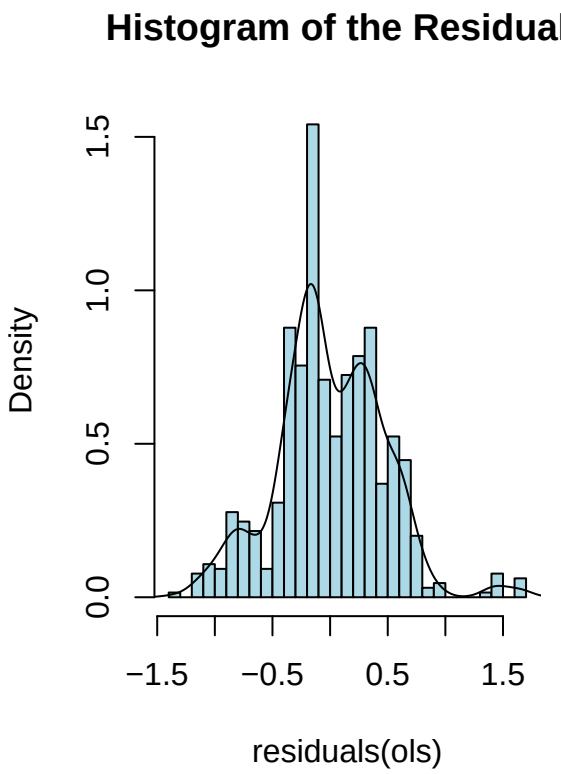
From theoretical notions:

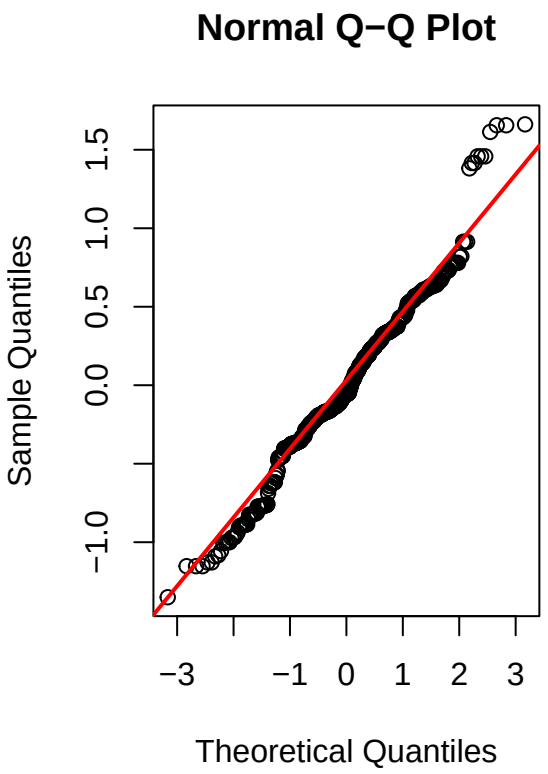
1. In order to address non-constant variance issue, transformations of the response variable (i.e. variance stabilizing transformations) are crucial. However, a logarithmic transformation has been already adopted.
2. In order to address non-linearity issue, it is possible to perform transformations on the predictors.

Although, after testing various transformations on continuous predictors, no significant improvement in non-linearity was observed. Given this, I proceeded the diagnostic using *ols*.

5.3) Normality

Normality assumption can be checked firstly though a histogram of the residuals, in order to understand how they are distributing and consequently through the QQ-plot, that is a crucial tool to assess normality.






```
shapiro_test = shapiro.test(residuals(ols))
print(shapiro_test)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ols)
## W = 0.98133, p-value = 2.313e-07
```

Furthermore, the graphical analysis can be accompanied by an useful test, that is the Shapiro-Wilk normality test, which helps to address the following hypothesis:

$$\begin{cases} H_0 : \text{The residuals follow a normal distribution} \\ H_1 : \text{The residuals do not follow a normal distribution} \end{cases}$$

The p-value associated to test is less than 0.05, so the result actually support what displayed in the graphical analysis and consequently the normality assumption is violated. The Q-Q plot shows deviations in the upper tail, indicating potential outliers or influential points, therefore the next step is to check for their presence.

5.4) High-leverage points

High-leverage points can be defined as extreme values in the predictors' space. In order to identify them, we need to measure the distance of each point with respect to the joint center of the observations (considering all the covariates).

To identify high-leverage points, we calculate the leverage values h_{ii} using the *hatvalues()* function, which measures how much each observation influences its own fitted value. Furthermore, from theoretical notions, we know by rule of thumb that we have a high-leverage point if h_{ii} is bigger than $\frac{2(p+1)}{n}$.

```r
hat = hatvalues(ols)
head(hat)
```

```
##          1          2          3          4          5          6
## 0.01516492 0.01756154 0.01961036 0.01952691 0.01464405 0.01810408
```

```r
hat[which(hat>=(2*15/nrow(data)))]
```

```
## named numeric(0)
```

This result indicates that zero high-leverage points have been identified, suggesting that no observations have an unusually large influence on their own fitted values. However, further investigation on potential outliers and influential poits is needed.

5.5) Outliers

Outliers are data points that do not fit the model with respect to $\hat{y}$. Outliers can be identified through the concept of the standardized residuals:

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

This procedure can be implemented thorugh the *rstandard()* function:

```r
stand_res = rstandard(ols)
range(stand_res)
```

```
## [1] -2.912713  3.558698
```

By rule of thumb, we say to have an outlier when $r_i > |3|$. Since it is possible to denote that *range(stand_res)* exceeds the value of 3, this means that we have some outliers:

```r
stand_res[which(stand_res>3)]
```

```
##       64      154      285      443      501      531      585      611
## 3.127054 3.127054 3.127054 3.040226 3.040226 3.547499 3.558698 3.547499
##      633
## 3.461000
```

The observations corresponding to rows 64, 154, 285, 442 and 501, 531, 585, 611 and 633 can be considered outliers.

5.6) Influential points

We say that a point is influential if its removal causes a large change in the fit of the model. To identify influential points, we compute Cook's Distance using the *cooks.distance()* function:

```r
Di = cooks.distance(ols)
Di[which.max(Di)]
```

```
##        396
## 0.02027144
```

By rule of thumb, we have an influential point if $D_i > 0.5$, but since the maximum value resulting from the computation of Cook's Distance is less than 0.5, then no influential points are identified. This result suggests that while some points may be outliers, they do not cause major changes to the overall fit of the model.

6) Best model obtained

Since the model remained unchanged after diagnostics, *ols* is the best model obtained.

6.1) Interpretation of the coefficients of the best model

The coefficients of the best model are the following:

```
summary(ols)$coefficients
```

```
##                            Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)          -0.259964173 0.112710793 -2.306471 2.140488e-02
## Age                   0.008494836 0.002664088  3.188647 1.499577e-03
## GenderMale            0.109017055 0.038421087  2.837428 4.692787e-03
## DiabetesYes          -0.216103699 0.070916469 -3.047299 2.404788e-03
## Short_StaturePSS     -0.197913868 0.057676566 -3.431443 6.392780e-04
## Short_StatureVariant -0.203883730 0.058669911 -3.475099 5.453887e-04
## IgM                   0.203854760 0.045097273  4.520334 7.363965e-06
## Marshmarsh type 1     0.446310074 0.071669131  6.227368 8.620857e-10
## Marshmarsh type 2     0.279191913 0.078483619  3.557327 4.025119e-04
## Marshmarsh type 3a    0.448628051 0.078595531  5.708061 1.753527e-08
## Marshmarsh type 3b    0.563920102 0.070746369  7.971011 7.310144e-15
## Marshmarsh type 3c    0.282128215 0.081359195  3.467687 5.603635e-04
## Marshnone             0.424681723 0.077637199  5.470080 6.473056e-08
```

- $\beta_0 = -0.2599 \Rightarrow e^{-0.260} = 0.771$. This corresponds to the expected IgA values for a female newborn with IgM immunodeficiency, no diabetes, Marsh type 0 (i.e., no damage to the small intestine), and classified as Short Stature DSS. Even if still possible, this result represents an atypical scenario.
- $\beta_1 = 0.0084 \Rightarrow e^{0.00849} = 1.0085$. A unitary increase in Age implies a multiplicative increase of 1.0085 in the expected IgA value.
- $\beta_2 = 0.1090 \Rightarrow e^{0.109} = 1.115$. Being male increases the expected IgA level by a factor of 1.115 compared to females.
- $\beta_3 = -0.2161 \Rightarrow e^{-0.216} = 0.805$. Having diabetes decreases the expected IgA value by a factor of 0.805, compared to non-diabetic individuals.
- $\beta_4 = -0.1979 \Rightarrow e^{-0.198} = 0.820$. Being classified as Short Stature PSS reduces the expected IgA value by a factor of 0.820, compared to the reference Short Stature category (Short_Stature DSS).
- $\beta_5 = -0.2038 \Rightarrow e^{-0.204} = 0.816$. Being classified as Short Stature Variant reduces the expected IgA value by a factor of 0.816, compared to the reference Short Stature category.
- $\beta_6 = 0.2038 \Rightarrow e^{0.204} = 1.226$. A unitary increase in IgM implies a multiplicative increase in the expected IgA value by a factor of 1.226.
- $\beta_7 = 0.4463 \Rightarrow e^{0.446} = 1.563$. Having Marsh type 1 increases the expected IgA value by a factor of 1.563, compared to the reference Marsh category (Marsh type 0).
- $\beta_8 = 0.2791 \Rightarrow e^{0.279} = 1.322$. Having Marsh type 2 increases the expected IgA value by a factor of 1.322, compared to the reference Marsh category.
- $\beta_9 = 0.4486 \Rightarrow e^{0.448} = 1.565$. Having Marsh type 3a increases the expected IgA value by a factor of 1.565, compared to the reference Marsh category.
- $\beta_{10} = 0.5639 \Rightarrow e^{0.564} = 1.758$. Having Marsh type 3b increases the expected IgA value by a factor of 1.758, compared to the reference Marsh category.
- $\beta_{11} = 0.2821 \Rightarrow e^{0.282} = 1.326$. Having Marsh type 3c increases the expected IgA value by a factor of 1.326, compared to the reference Marsh category.
- $\beta_{12} = 0.4246 \Rightarrow e^{0.425} = 1.530$. Having no Marsh classification (Marsh none) increases the expected IgA value by a factor of 1.530, compared to the reference Marsh category.

6.2) Individual t-test and uncertainties

In order to check for the significance of each of the $\hat{\beta}_j$, we need to perform the following hypothesis testing and then check the related p-values from the *summary(ols)*:

$$\begin{cases} H_0 : \hat{\beta}_j = 0 \\ H_1 : \hat{\beta}_j \neq 0 \end{cases}$$

If the resulting p-value is less than 0.05, then there is evidence against the null hypothesis and we reject it, meaning that $\hat{\beta}_j$ is significant. Otherwise, if the p-value it is greater than 0.05, then we accept the null hypothesis, meaning that there is not evidence to affirm that $\hat{\beta}_j$ is significant.

Looking at the *summary(ols)*, it is possible to denote that all the coefficients have small p-values, since they are all below the significance level of 0.05; consequently, all the $\hat{\beta}_j$ are significant.

Moreover, since all predictors are significant, their 95% confidence intervals do not include zero, reinforcing their effect on the response variable and the direction of the effect,whether positive or negative, remains the same. This conclusion is confirmed by the resulting C.I obtained through the *confint()* function:

```
confint(ols)
```

```
##                              2.5 %      97.5 %
## (Intercept)          -0.481294465 -0.03863388
## Age                   0.003263363  0.01372631
## GenderMale            0.033569529  0.18446458
## DiabetesYes          -0.355362437 -0.07684496
## Short_StaturePSS     -0.311173395 -0.08465434
## Short_StatureVariant -0.319093891 -0.08867357
## IgM                   0.115297201  0.29241232
## Marshmarsh type 1     0.305573334  0.58704682
## Marshmarsh type 2     0.125073554  0.43331027
## Marshmarsh type 3a    0.294289931  0.60296617
## Marshmarsh type 3b    0.424995389  0.70284481
## Marshmarsh type 3c    0.122363085  0.44189334
## Marshnone             0.272225481  0.57713797
```

6.3) Measures of goodnees of fit

In linear regression, one of the most important measures of goodness of fit is the *coefficient of determination*, R^2, that represents the proportion of variability of the response variable explained by the linear regression model using the predictors.

Furthermore, as a measure of goodness of fit, I decided to adopt the Adjusted R^2, that is a refined version of the coefficient of determination that adjusts for the number of predictors in the model.

```
summary(ols)$adj.r.squared
```

```
## [1] 0.2247432
```

```
summary(ols)$r.squared
```

```
## [1] 0.2390998
```

The obtained value of 0.2247 suggests that only about 22.47% of the variability in IgA levels is explained by the predictors in the model. The low Adjusted R^2 does not necessarily invalidate the model, but it indicates that IgA levels are influenced by additional factors not captured here (e.g. genetic predisposition). Given the clinical complexity of Celiac Disease and IgA antibody, this result is reasonable given the intricate area of medical research.

7) Model comparison

I decided to perform a model comparison in order to understand if it's possible to remove some of the predictors without loosing substantially important informations.

In the sub-model that will be checked, the predictors *Diabetes* and *Short_Stature* have been dropped. This

procedure can be done through the ANOVA test, that perform the following hypothesis testing:

$$\begin{cases} H_0: \text{ The full model and the submodel are equivalent} \\ H_1: \text{ The full model and the submodel are not equivalent} \end{cases}$$

I will use the *anova()* function in order to implement it on R:

```
ols2 = lm(IgA_log ~ Age + Gender + Marsh + IgM, data = data)
anova(ols,ols2)
```

```
## Analysis of Variance Table
##
## Model 1: IgA_log ~ Age + Gender + Diabetes + Short_Stature + IgM + Marsh
## Model 2: IgA_log ~ Age + Gender + Marsh + IgM
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    636 141.01
## 2    639 147.57 -3   -6.5685 9.8756 2.261e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the resulting p-value from the ANOVA test is less than 0.05, we reject the null hypothesis H_0 which, states that the full model and the sub-model are equivalent. This indicates that removing Diabetes and Short_Stature leads to a substantial loss of information, meaning the reduced model cannot be considered equivalent to the full model *ols*. Consequently, the full model should be preferred for the analysis.

8) Prediction

Supposing now that we have informations about a new observation, we can create a new data frame in order to store this information and use the *predict()* function to estimate the expected IgA_log value, along with the related 95% prediction interval:

```
new_data = data.frame(Age = 35, Gender = "Male", Marsh = "marsh type 3a",
                      Short_Stature = "Variant", Diabetes = "Yes",  IgM = 2.5)
pred = predict(ols, newdata = new_data, interval = "confidence", level = 0.95)
pred
```

```
##         fit       lwr     upr
## 1 0.6846497 0.4747193 0.89458
```

The model predicts a log_IgA level of 0.684, with a 95% confidence interval ranging from 0.474 to 0.895. Transforming back to the original scale, the expected IgA level is approximately:

$$e^{0.684} \approx 1.98$$

with the prediction interval given by:

$$\left[e^{0.474}, e^{0.895}\right] \approx [1.61, 2.45]$$

The predicted IgA level of 1.98 falls within the observed range of values in the dataset but is positioned in the upper quartile, as the mean value of the IgA antibody in the dataset is 1.709. This relatively high predicted value suggests a strong immune response, which may be clinically relevant.

Given that higher IgA levels are often associated with mucosal inflammation and immune activation, this result is consistent with the individual's characteristics (in the new observation), particularly the Marsh Type 3a classification and presence of diabetes, which are both linked to immune system dysregulation.

9) Simulation

The Observed versus Simulated IgA plot evaluates how well the simulated values align with the observed values on the logarithmic scale. The red diagonal dashed line represents the ideal scenario where simulated values perfectly match the observed ones (y = x).

```
#set.seed(567)

#beta = coefficients(ols)
#X = model.matrix(ols)
#y_hat = X %*% beta + rnorm(n, 0, sigma(ols))
#y_obs = ols$model[, 1]
#plot(y_obs, y_hat, xlab=Observed Response, ylab = Simulated Response, pch=16, col=blue);
#abline(a=0, b=1, col=red, lwd=2)


set.seed(567)

beta = coefficients(ols)
X = model.matrix(ols)
y_hat = X %*% beta + rnorm(n, 0, sigma(ols))  # Simulated values
y_obs = ols$model[, 1]  # Observed values

# Assuming 'Category' is a factor variable in your dataset
category = ols$model$Marsh  # Replace 'Marsh' with the actual categorical variable

# Scatterplot with color-coded categories
plot(y_obs, y_hat, col = as.numeric(as.factor(category)), pch = 16,
     xlab = "Observed Response", ylab = "Simulated Response"); legend("bottomright", legend = levels(as
```

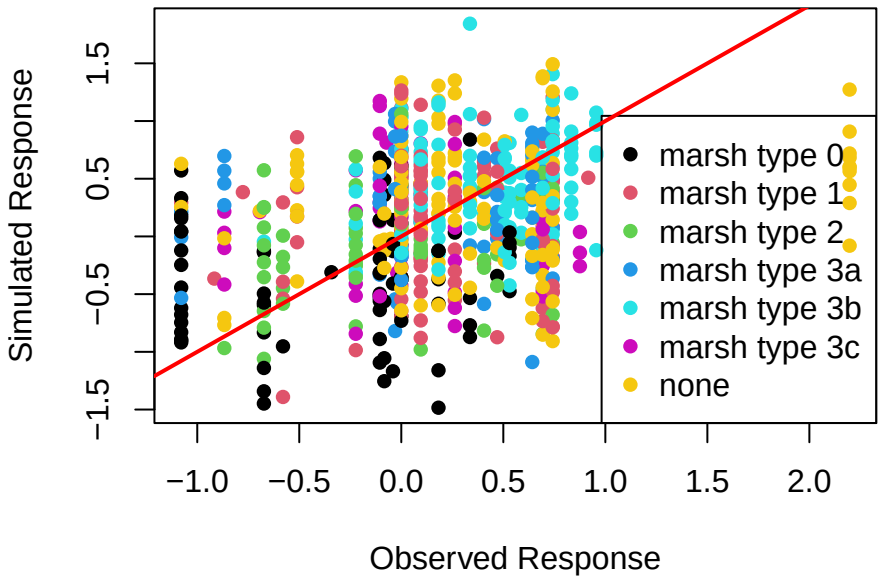


Figure 7: Observed IgA values versus simulated

```r
residuals_train <- residuals(ols)

# Calculate Training MSE
mse_train <- mean(residuals_train^2)
print(mse_train)
```

```
## [1] 0.2172667
```

```r
mse_difference <- cv.mean - mse_train
print(mse_difference)
```

```
##           1           2           3           4           5           6
## 0.048898734 0.053616091 0.033641305 0.040964540 0.039830034 0.026312119
##           7           8           9          10          11          12
## 0.033921655 0.023586072 0.022732694 0.019096998 0.017196992 0.009865414
##          13          14          15          16          17
## 0.012101654 0.012262609 0.013080852 0.012274101 0.012049352
```

Looking at Figure 7, it is possible to denote a deviation between the observed and simulated values, especially for higher IgA levels. Most of the simulated values are located between -1 and 1, aligning with the majority of the observed data, but there are notable discrepancies for higher IgA values, where the model fails to capture extreme values.

This result is not surprising, since as previously discussed in *Section 6.3*, where the adjusted R^2 was 0.2247, implying that important influencing factors may be missing. The observed deviation in extreme values further confirms this limitation, as the model appears to underperform in capturing higher IgA levels. This suggests that adding relevant predictors or exploring nonlinear modeling approaches could enhance predictive accuracy.

10) Conclusions

Considering the goal of the analysis (i.e. prediction), the final model showed moderate predictive power with an Adjusted R^2 of 0.224, indicating that additional factors may influence IgA levels. The prediction analysis yielded an IgA estimate of 1.98 positioned in the upper range but within the observed data distribution, aligning with clinical findings that higher IgA levels indicate immune activation.

To conclude, despite limitations, the model provides useful insights regarding IgA antibody and since Celiac Disease it's something that affects personally me and my family, I found extremely interesting performing this analysis.

References

- http://www.kaggle.com/
- Weisberg S. (2014), Applied Linear Regression (4th edition)
- Fox & Weisberg (2011), An R Companion to Applied Regression (2nd edition)