Machine Learning with R: Linear Regression

Armand Tossou

9/6/2021

Resources

In one of the upcoming lectures, we'll encounter the free ISLR book download link, its been updated, you can now just find the link on the homepage for the latest edition (on the right you see a picture).

Here is that link: www-bcf.usc.edu/~gareth/ISL/

Here is a link to the book directly: http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Sixth%20Printing.pdf

Get our Data

We will use the Student Performance Data Set from UC Irvine's Machine Learning Repository! Download this data our just use the supplied csv files in the notebook repository. We'll specifically look at the math class (student-mat.csv). Make sure to take note that the delimiter is a semi-colon.

```
# check our working directory
getwd()
```

```
## [1] "C:/Users/adtos/Dropbox/Data_Science/R programming/Data Science and Machine Learning Bootcamp wi
```

```
#  "C:\Users\adtos\Dropbox\Data_Science\R programming\Data Science and Machine Learning Bootcamp with R

# load the dataset
df <- read.csv('C:/Users/adtos/Dropbox/Data_Science/R programming/Data Science and Machine Learning Boo
```

Preview the data:

```
head(df)
```

```
##   school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob     reason
## 1     GP   F  18       U     GT3       A    4    4  at_home  teacher     course
## 2     GP   F  17       U     GT3       T    1    1  at_home    other     course
## 3     GP   F  15       U     LE3       T    1    1  at_home    other      other
## 4     GP   F  15       U     GT3       T    4    2   health services       home
## 5     GP   F  16       U     GT3       T    3    3    other    other       home
## 6     GP   M  16       U     LE3       T    4    3 services    other reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2         2        0       yes     no   no         no
## 2   father          1         2        0        no    yes   no         no
## 3   mother          1         2        3       yes     no  yes         no
## 4   mother          1         3        0        no    yes  yes        yes
## 5   father          1         2        0        no    yes  yes         no
## 6   mother          1         2        0        no    yes  yes        yes
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1     yes    yes       no       no      4        3     4    1    1      3
## 2      no    yes      yes       no      5        3     3    1    1      3
## 3     yes    yes      yes       no      4        3     2    2    3      3
## 4     yes    yes      yes      yes      3        2     2    1    1      5
## 5     yes    yes       no       no      4        3     2    1    2      5
## 6     yes    yes      yes       no      5        4     2    1    2      5
##   absences G1 G2 G3
## 1        6  5  6  6
## 2        4  5  5  6
## 3       10  7  8 10
## 4        2 15 14 15
## 5        4  6 10 10
## 6       10 15 15 15
```

Get a summary of the data:

```
summary(df)
```

```
##     school              sex                 age          address
##  Length:395         Length:395         Min.   :15.0   Length:395
##  Class :character   Class :character   1st Qu.:16.0   Class :character
##  Mode  :character   Mode  :character   Median :17.0   Mode  :character
##                                        Mean   :16.7
##                                        3rd Qu.:18.0
##                                        Max.   :22.0
##    famsize            Pstatus               Medu            Fedu
##  Length:395         Length:395         Min.   :0.000   Min.   :0.000
##  Class :character   Class :character   1st Qu.:2.000   1st Qu.:2.000
##  Mode  :character   Mode  :character   Median :3.000   Median :2.000
##                                        Mean   :2.749   Mean   :2.522
##                                        3rd Qu.:4.000   3rd Qu.:3.000
##                                        Max.   :4.000   Max.   :4.000
##      Mjob               Fjob              reason            guardian
##  Length:395         Length:395         Length:395         Length:395
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    traveltime      studytime        failures       schoolsup
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   Length:395
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   Class :character
##  Median :1.000   Median :2.000   Median :0.0000   Mode  :character
##  Mean   :1.448   Mean   :2.035   Mean   :0.3342
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000
##     famsup              paid            activities          nursery
##  Length:395         Length:395         Length:395         Length:395
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     higher            internet          romantic            famrel
##  Length:395         Length:395         Length:395         Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:4.000
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
##                                                            Mean   :3.944
##                                                            3rd Qu.:5.000
##                                                            Max.   :5.000
##     freetime         goout            Dalc            Walc
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :3.000   Median :3.000   Median :1.000   Median :2.000
##  Mean   :3.235   Mean   :3.109   Mean   :1.481   Mean   :2.291
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##      health         absences            G1              G2
##  Min.   :1.000   Min.   : 0.000   Min.   : 3.00   Min.   : 0.00
##  1st Qu.:3.000   1st Qu.: 0.000   1st Qu.: 8.00   1st Qu.: 9.00
##  Median :4.000   Median : 4.000   Median :11.00   Median :11.00
##  Mean   :3.554   Mean   : 5.709   Mean   :10.91   Mean   :10.71
##  3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:13.00   3rd Qu.:13.00
##  Max.   :5.000   Max.   :75.000   Max.   :19.00   Max.   :19.00
##        G3
##  Min.   : 0.00
##  1st Qu.: 8.00
##  Median :11.00
##  Mean   :10.42
##  3rd Qu.:14.00
##  Max.   :20.00
```

Attribute Information

Here is the attribute information for our data set: Attribute Information:

Attributes for both student-mat.csv (Math course) and student-por.csv (Portuguese language course) datasets:

- 1 school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
- 2 sex - student's sex (binary: 'F' - female or 'M' - male)
- 3 age - student's age (numeric: from 15 to 22)
- 4 address - student's home address type (binary: 'U' - urban or 'R' - rural)
- 5 famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
- 6 Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
- 7 Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
- 8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
- 9 Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')

- 10 Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
- 11 reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
- 12 guardian - student's guardian (nominal: 'mother', 'father' or 'other')
- 13 traveltime - home to school travel time (numeric: 1 - less than 15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - more than 1 hour)
- 14 studytime - weekly study time (numeric: 1 - less than 2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - more than 10 hours)
- 15 failures - number of past class failures (numeric: n if between 1 and 3 , else 4)
- 16 schoolsup - extra educational support (binary: yes or no)
- 17 famsup - family educational support (binary: yes or no)
- 18 paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- 19 activities - extra-curricular activities (binary: yes or no)
- 20 nursery - attended nursery school (binary: yes or no)
- 21 higher - wants to take higher education (binary: yes or no)
- 22 internet - Internet access at home (binary: yes or no)
- 23 romantic - with a romantic relationship (binary: yes or no)
- 24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
- 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)
- 26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)
- 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 29 health - current health status (numeric: from 1 - very bad to 5 - very good)
- 30 absences - number of school absences (numeric: from 0 to 93)

These grades are related with the course subject, Math or Portuguese:

- 31 G1 - first period grade (numeric: from 0 to 20)
- 31 G2 - second period grade (numeric: from 0 to 20)
- 32 G3 - final grade (numeric: from 0 to 20, output target)

Clean the Data

Next we have to clean this data. This data is actually already cleaned for you, But here are some things you may want to consider doing for other data sets:

Check for NA values

Let's see if we have any NA values:

```
any(is.na(df))
```

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

Great! Most real data sets will probably have NA or Null values, so its always good to check! Its up to you how to deal with them, either dropping them if they aren't too many, or imputing other values, like the mean value.

Categorical Features

Moving on, let's make sure that categorical variables have a `factor` set to them. For example, the MJob column refers to categories of Job Types, not some numeric value from 1 to 5. R is actually really good at detecting these sort of values and will take care of this work for you a lot of the time, but always keep in mind the use of `factor()` as a possible. Luckily this is basically already, we can check this using the `str()` function:

```
str(df)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : chr  "at_home" "at_home" "at_home" "health" ...
##  $ Fjob      : chr  "teacher" "other" "other" "services" ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : chr  "yes" "no" "yes" "no" ...
##  $ famsup    : chr  "no" "yes" "no" "yes" ...
##  $ paid      : chr  "no" "no" "yes" "yes" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : chr  "yes" "no" "yes" "yes" ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

Exploratory Data Analysis

Let's use `ggplot2` to explore the data a bit. Feel free to expand on this section:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggthemes)
```

Correlation and CorrPlots

From Wikipedia, correlation is defined as:

- In statistics, dependence or association is any statistical relationship, whether causal or not, between two random variables or two sets of data. Correlation is any of a broad class of statistical relationships involving dependence, though in common usage it most often refers to the extent to which two variables have a linear relationship with each other. Familiar examples of dependent phenomena include the correlation between the physical statures of parents and their offspring, and the correlation between the demand for a product and its price.

Correlation plots are a great way of exploring data and seeing if there are any interaction terms. Let's start off by just grabbing the numeric data (we can't see correlation for categorical data):

```
# Grab only numeric columns. This is a mask: a vector of booleans
num.cols <- sapply(df, is.numeric)

# Filter to numeric columns for correlation
cor.data <- cor(df[,num.cols])

cor.data
```

```
##                     age         Medu         Fedu   traveltime    studytime
## age         1.000000000 -0.163658419 -0.163438069  0.070640721 -0.004140037
## Medu       -0.163658419  1.000000000  0.623455112 -0.171639305  0.064944137
## Fedu       -0.163438069  0.623455112  1.000000000 -0.158194054 -0.009174639
## traveltime  0.070640721 -0.171639305 -0.158194054  1.000000000 -0.100909119
## studytime  -0.004140037  0.064944137 -0.009174639 -0.100909119  1.000000000
## failures    0.243665377 -0.236679963 -0.250408444  0.092238746 -0.173563031
## famrel      0.053940096 -0.003914458 -0.001369727 -0.016807986  0.039730704
## freetime    0.016434389  0.030890867 -0.012845528 -0.017024944 -0.143198407
## goout       0.126963880  0.064094438  0.043104668  0.028539674 -0.063903675
## Dalc        0.131124605  0.019834099  0.002386429  0.138325309 -0.196019263
## Walc        0.117276052 -0.047123460 -0.012631018  0.134115752 -0.253784731
## health     -0.062187369 -0.046877829  0.014741537  0.007500606 -0.075615863
## absences    0.175230079  0.100284818  0.024472887 -0.012943775 -0.062700175
## G1         -0.064081497  0.205340997  0.190269936 -0.093039992  0.160611915
## G2         -0.143474049  0.215527168  0.164893393 -0.153197963  0.135879999
## G3         -0.161579438  0.217147496  0.152456939 -0.117142053  0.097819690
##               failures       famrel     freetime        goout         Dalc
## age         0.24366538  0.053940096  0.01643439  0.126963880  0.131124605
## Medu       -0.23667996 -0.003914458  0.03089087  0.064094438  0.019834099
## Fedu       -0.25040844 -0.001369727 -0.01284553  0.043104668  0.002386429
## traveltime  0.09223875 -0.016807986 -0.01702494  0.028539674  0.138325309
## studytime  -0.17356303  0.039730704 -0.14319841 -0.063903675 -0.196019263
## failures    1.00000000 -0.044336626  0.09198747  0.124560922  0.136046931
## famrel     -0.04433663  1.000000000  0.15070144  0.064568411 -0.077594357
## freetime    0.09198747  0.150701444  1.00000000  0.285018715  0.209000848
## goout       0.12456092  0.064568411  0.28501871  1.000000000  0.266993848
## Dalc        0.13604693 -0.077594357  0.20900085  0.266993848  1.000000000
## Walc        0.14196203 -0.113397308  0.14782181  0.420385745  0.647544230
## health      0.06582728  0.094055728  0.07573336 -0.009577254  0.077179582
## absences    0.06372583 -0.044354095 -0.05807792  0.044302220  0.111908026
## G1         -0.35471761  0.022168316  0.01261293 -0.149103967 -0.094158792
## G2         -0.35589563 -0.018281347 -0.01377714 -0.162250034 -0.064120183
## G3         -0.36041494  0.051363429  0.01130724 -0.132791474 -0.054660041
##                   Walc       health    absences          G1          G2
## age         0.11727605 -0.062187369  0.17523008 -0.06408150 -0.14347405
## Medu       -0.04712346 -0.046877829  0.10028482  0.20534100  0.21552717
## Fedu       -0.01263102  0.014741537  0.02447289  0.19026994  0.16489339
## traveltime  0.13411575  0.007500606 -0.01294378 -0.09303999 -0.15319796
## studytime  -0.25378473 -0.075615863 -0.06270018  0.16061192  0.13588000
## failures    0.14196203  0.065827282  0.06372583 -0.35471761 -0.35589563
## famrel     -0.11339731  0.094055728 -0.04435409  0.02216832 -0.01828135
## freetime    0.14782181  0.075733357 -0.05807792  0.01261293 -0.01377714
## goout       0.42038575 -0.009577254  0.04430222 -0.14910397 -0.16225003
## Dalc        0.64754423  0.077179582  0.11190803 -0.09415879 -0.06412018
## Walc        1.00000000  0.092476317  0.13629110 -0.12617921 -0.08492735
## health      0.09247632  1.000000000 -0.02993671 -0.07317207 -0.09771987
## absences    0.13629110 -0.029936711  1.00000000 -0.03100290 -0.03177670
## G1         -0.12617921 -0.073172073 -0.03100290  1.00000000  0.85211807
## G2         -0.08492735 -0.097719866 -0.03177670  0.85211807  1.00000000
## G3         -0.05193932 -0.061334605  0.03424732  0.80146793  0.90486799
##                     G3
## age        -0.16157944
## Medu        0.21714750
## Fedu        0.15245694
## traveltime -0.11714205
## studytime   0.09781969
## failures   -0.36041494
## famrel      0.05136343
## freetime    0.01130724
## goout      -0.13279147
## Dalc       -0.05466004
## Walc       -0.05193932
## health     -0.06133460
## absences    0.03424732
## G1          0.80146793
## G2          0.90486799
## G3          1.00000000
```

While this is fantastic information, it's hard to take it all in. Let's visualize all this data. There are lots of amazing 3rd party packages to do this, let's use and install the `corrgram` package and the `corrplot` package. This will also install a bunch of dependencies for the package.

```
## install packages

#install.packages('corrgram',repos = 'http://cran.us.r-project.org')
#install.packages('corrplot',repos = 'http://cran.us.r-project.org')

# load the packages

library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(corrgram)
```

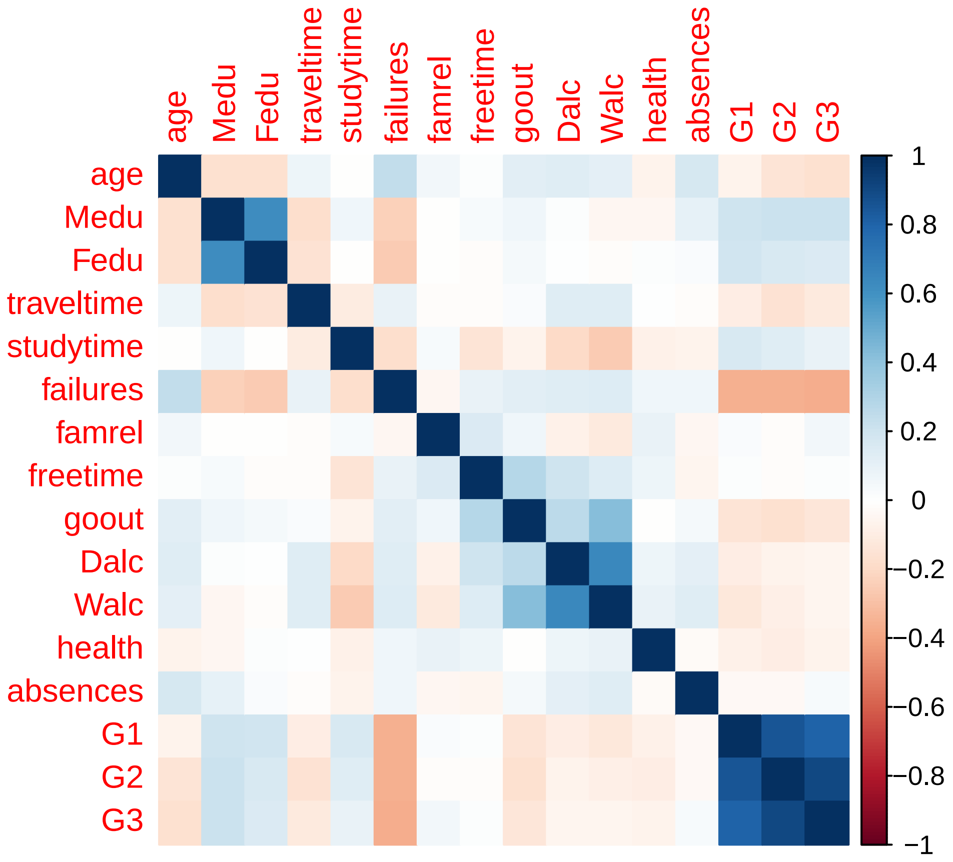
Let's start by using corrplot, the most common one. Here's a really nice documentation page on the package. I encourage you to play around with it.

```
# access the help page for this package
help(corrplot)
```

```
## starting httpd help server ... done
```

Create a correlation matrix for all the numeric variables in the dataset:

```
corrplot(cor.data,method='color')
```



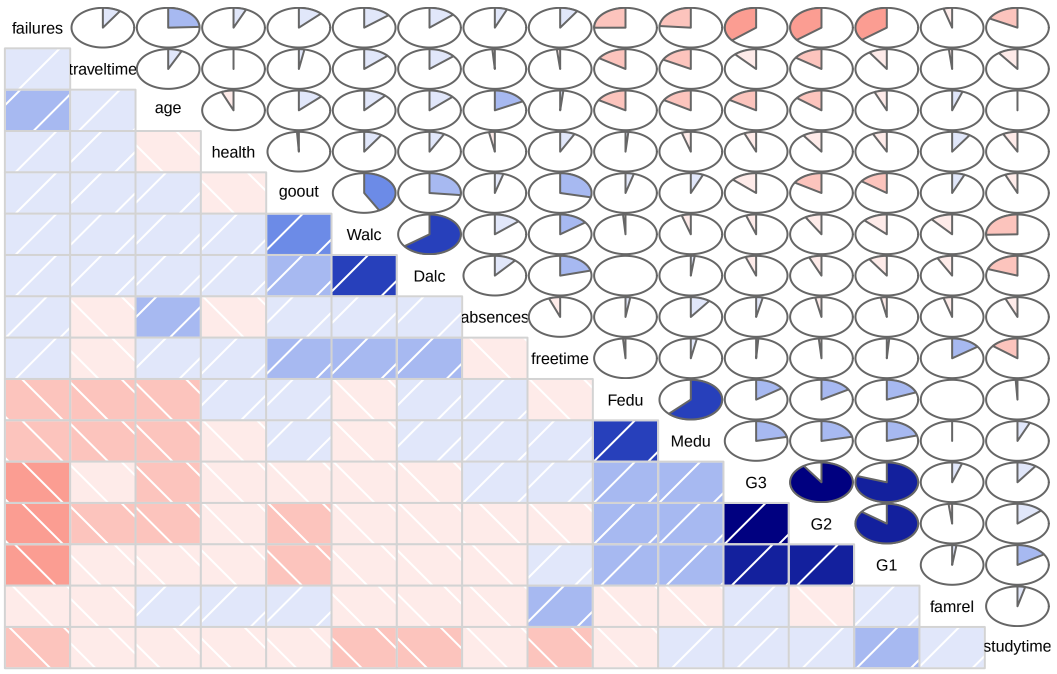
Clearly, we have very high correlation between G1, G2, and G3; which makes sense since, those are grades: - 31 G1 - first period grade (numeric: from 0 to 20) - 31 G2 - second period grade (numeric: from 0 to 20) - 32 G3 - final grade (numeric: from 0 to 20, output target)

Meaning good students do well each period, and poor students do poorly each period, etc. Also a high G1,G2, or G3 value has a negative correlation with failure (number of past class failures).

Also Mother and Father education levels are correlated, which also makes sense.

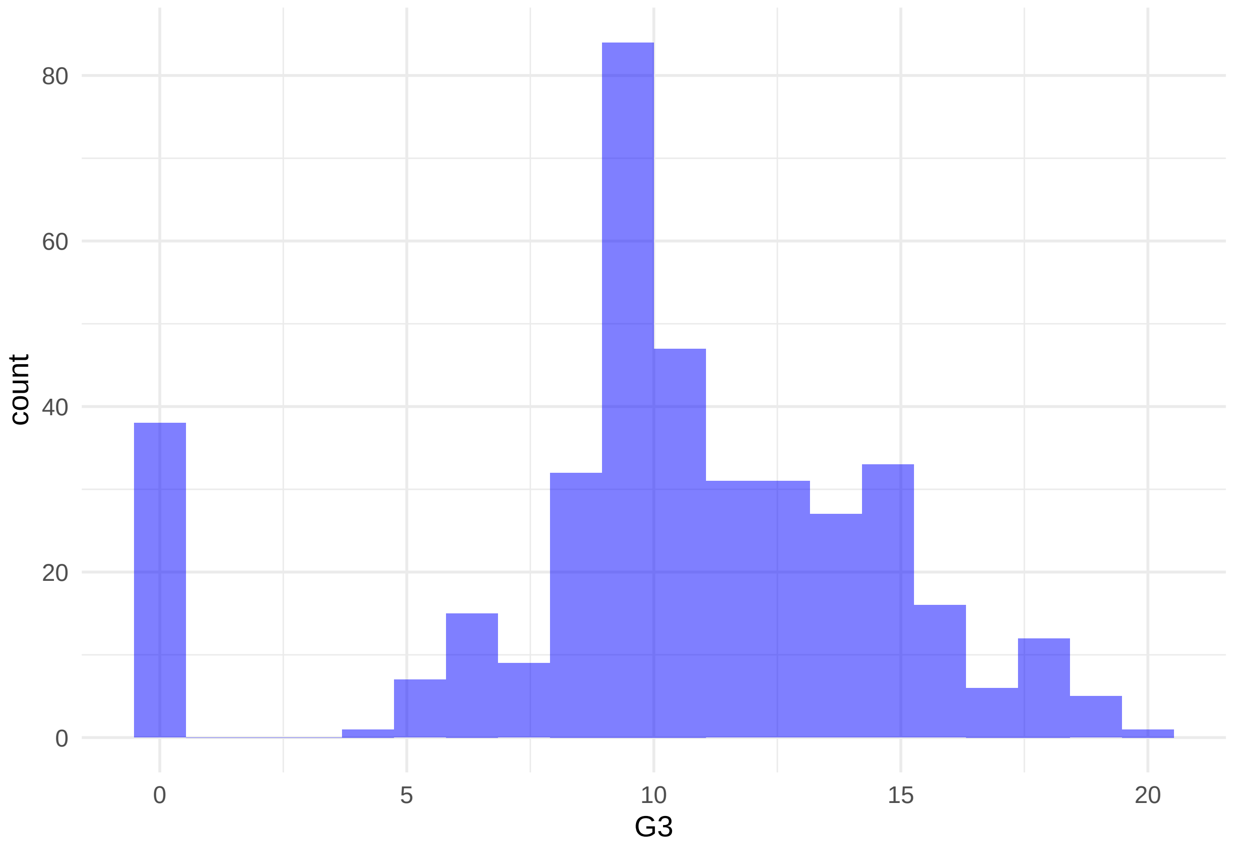
We can also use the `corrgram`, which allows to just automatically do these type of figures by just passing in the dataframe directly. There's a lot going on here, so reference the documentation of corrgram for more info.

```
corrgram(df,order=TRUE, lower.panel=panel.shade,
  upper.panel=panel.pie, text.panel=panel.txt)
```



Since we're going to eventually try to predict the G3 score, let's see a histogram of these scores:

```
ggplot(df,aes(x=G3)) + geom_histogram(bins=20,alpha=0.5,fill='blue') + theme_minimal()
```



Looks like quite a few students get a zero. This is a good place to ask questions, like are students missing the test? Also why is the mean occurrence so high? Is this test curved?

Let's continue by building a model.

Building a Model

General Form:

The general model of building a linear regression model in R looks like this:

`model <- lm(y ~ x1 + x2,data)`

or to use all the features in your data

`model <- lm(y ~. , data)` # Uses all features

Train and Test Data

We'll need to split our data into a training set and a testing set in order to test our accuracy. We can do this easily using the `caTools` library:

```
# install the package
#install.packages("caTools")

# Import Library
library(caTools)
```

```r
# Set a random seed so your "random" results are the same as this notebook
set.seed(101)

# Split up the sample, basically randomly assigns a booleansto a new column "sample"
sample <- sample.split(df$age, SplitRatio = 0.70) # SplitRatio = percent of sample==TRUE

# Training Data
train = subset(df, sample == TRUE)

# Testing Data
test = subset(df, sample == FALSE)
```

Training our Model

Let's train out model on our training data, then ask for a summary of that model:

```r
model <- lm(G3 ~ .,train)

summary(model)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7681 -0.6423  0.2294  1.0691  4.5942
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.329568   2.474569  -0.537 0.591574
## schoolMS          0.838581   0.470545   1.782 0.076016 .
## sexM              0.034883   0.275586   0.127 0.899382
## age              -0.214994   0.119579  -1.798 0.073472 .
## addressU          0.067190   0.326035   0.206 0.836905
## famsizeLE3       -0.111068   0.283228  -0.392 0.695302
## PstatusT         -0.153653   0.401679  -0.383 0.702417
## Medu              0.279949   0.171111   1.636 0.103164
## Fedu             -0.221275   0.151103  -1.464 0.144422
## Mjobhealth        0.002065   0.610532   0.003 0.997304
## Mjobother         0.509947   0.403195   1.265 0.207209
## Mjobservices      0.475476   0.435332   1.092 0.275857
## Mjobteacher       0.285345   0.550640   0.518 0.604802
## Fjobhealth        0.433172   0.774191   0.560 0.576343
## Fjobother        -0.296792   0.577217  -0.514 0.607611
## Fjobservices     -0.311595   0.593628  -0.525 0.600148
## Fjobteacher      -0.321205   0.712695  -0.451 0.652628
## reasonhome       -0.431435   0.319907  -1.349 0.178755
## reasonother       0.159612   0.454480   0.351 0.725755
## reasonreputation -0.051845   0.317894  -0.163 0.870589
## guardianmother    0.267462   0.311371   0.859 0.391226
## guardianother    -0.157335   0.554872  -0.284 0.777003
## traveltime        0.274301   0.197865   1.386 0.166968    
## studytime        -0.140650   0.155149  -0.907 0.365577    
## failures         -0.185333   0.211040  -0.878 0.380739    
## schoolsupyes      0.562716   0.379303   1.484 0.139268    
## famsupyes         0.369402   0.268848   1.374 0.170745    
## paidyes           0.060643   0.270971   0.224 0.823107    
## activitiesyes    -0.286006   0.247519  -1.155 0.249063    
## nurseryyes       -0.426858   0.315064  -1.355 0.176774    
## higheryes         0.503353   0.677346   0.743 0.458148    
## internetyes      -0.097405   0.331040  -0.294 0.768834    
## romanticyes      -0.243837   0.268414  -0.908 0.364577    
## famrel            0.494479   0.136663   3.618 0.000363 ***
## freetime         -0.139869   0.138297  -1.011 0.312879    
## goout             0.078871   0.128794   0.612 0.540879    
## Dalc             -0.248633   0.178366  -1.394 0.164651    
## Walc              0.221434   0.139178   1.591 0.112950    
## health            0.027495   0.095100   0.289 0.772748    
## absences          0.060830   0.017915   3.396 0.000804 ***
## G1                0.162966   0.076956   2.118 0.035255 *  
## G2                0.994677   0.066450  14.969  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.868 on 235 degrees of freedom
## Multiple R-squared:  0.8616, Adjusted R-squared:  0.8374 
## F-statistic: 35.68 on 41 and 235 DF,  p-value: < 2.2e-16
```

Model Interpretation

Understanding requires a general understanding of statistics, check out Wikipedia for overviews on some of these topics, as well as the ISLR book. Here's a quick guide on understanding the model summary:

1. Residuals: The residuals are the difference between the actual values of the variable you're predicting and predicted values from your regression–y - ŷ. For most regressions you want your residuals to look like a normal distribution when plotted. If our residuals are normally distributed, this indicates the mean of the difference between our predictions and the actual values is close to 0 (good) and that when we miss, we're missing both short and long of the actual value, and the likelihood of a miss being far from the actual value gets smaller as the distance from the actual value gets larger.

Think of it like a dartboard. A good model is going to hit the bullseye some of the time (but not everytime). When it doesn't hit the bullseye, it's missing in all of the other buckets evenly (i.e. not just missing in the 16 bin) and it also misses closer to the bullseye as opposed to on the outer edges of the dartboard.

2. Significance Stars The stars are shorthand for significance levels, with the number of asterisks displayed according to the p-value computed. *** for high significance and * for low significance. In this case, *** indicates that it's unlikely that no relationship exists b/w absences and G3 scores.

3. Estimated Coeffecient The estimated coefficient is the value of slope calculated by the regression. It might seem a little confusing that the Intercept also has a value, but just think of it as a slope that is always multiplied by 1. This number will obviously vary based on the magnitude of the variable you're inputting into the regression, but it's always good to spot check this number to make sure it seems reasonable.

4. Standard Error of the Coefficient Estimate Measure of the variability in the estimate for the coefficient. Lower means better but this number is relative to the value of the coefficient. As a rule of thumb, you'd like this value to be at least an order of magnitude less than the coefficient estimate.

5. t-value of the Coefficient Estimate Score that measures whether or not the coefficient for this variable is meaningful for the model. You probably won't use this value itself, but know that it is used to calculate the p-value and the significance levels.

6. Variable p-value Probability the variable is NOT relevant. You want this number to be as small as possible. If the number is really small, R will display it in scientific notation.

7. Significance Legend The more punctuation there is next to your variables, the better.

Blank=bad, Dots=pretty good, Stars=good, More Stars=very good

8. Residual Std Error / Degrees of Freedom The Residual Std Error is just the standard deviation of your residuals. You'd like this number to be proportional to the quantiles of the residuals in #1. For a normal distribution, the 1st and 3rd quantiles should be 1.5 +/- the std error.

The Degrees of Freedom is the difference between the number of observations included in your training sample and the number of variables used in your model (intercept counts as a variable).

9. R-squared Metric for evaluating the goodness of fit of your model. Higher is better with 1 being the best. Corresponds with the amount of variability in what you're predicting that is explained by the model. WARNING: While a high R-squared indicates good correlation, correlation does not always imply causation.

10. F-statistic & resulting p-value Performs an F-test on the model. This takes the parameters of our model (in our case we only have 1) and compares it to a model that has fewer parameters. In theory the model with more parameters should fit better. If the model with more parameters (your model) doesn't perform better than the model with fewer parameters, the F-test will have a high p-value (probability NOT significant boost). If the model with more parameters is better than the model with fewer parameters, you will have a lower p-value.

The DF, or degrees of freedom, pertains to how many variables are in the model. In our case there is one variable so there is one degree of freedom.

Looks like Absences, G1, and G2 scores are good predictors. With age and activities also possibly contributing to a good model.

Visualize our Model

We can visualize our linear regression model by plotting out the residuals, the residuals are basically a measure of how off we are for each point in the plot versus our model (the error).

```r
# Grab residuals
res <- residuals(model)

# Convert to DataFrame for gglpot
res <- as.data.frame(res)

head(res)
```

```
##           res
## 1   0.9678451
## 5   1.1829980
## 6  -1.4096050
## 7   0.1125706
## 9   0.3814670
## 10  0.3221204
```

Why Plot Residuals?

We want a histogram of our residuals to be normally distributed, something with a strong bimodal distribution may be a warning that our data was not a good fit for lienar regression. However, this can also be hidden from out model. A famous example is Anscombe's Quartet.

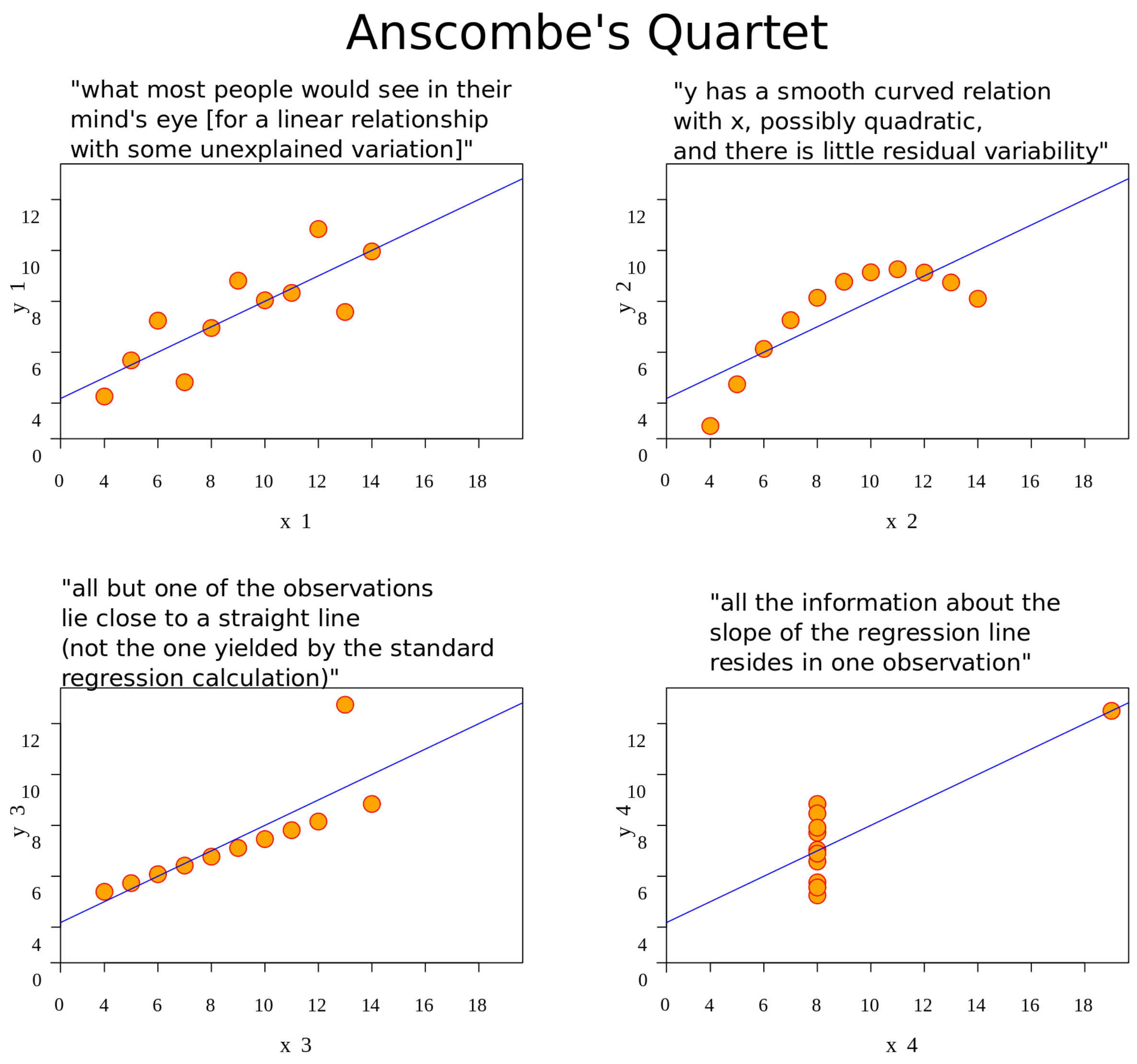
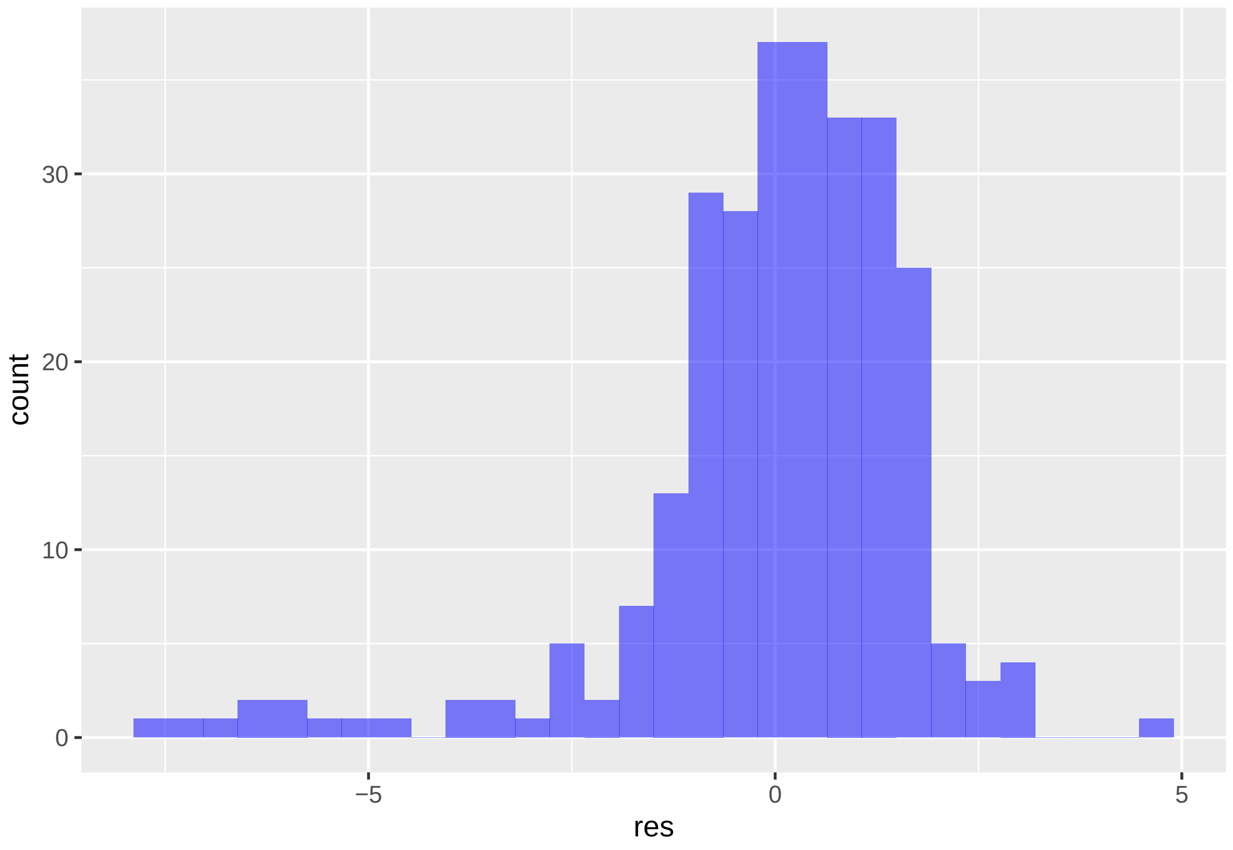


Figure 1: Anscombe's quartet

Using ggplot

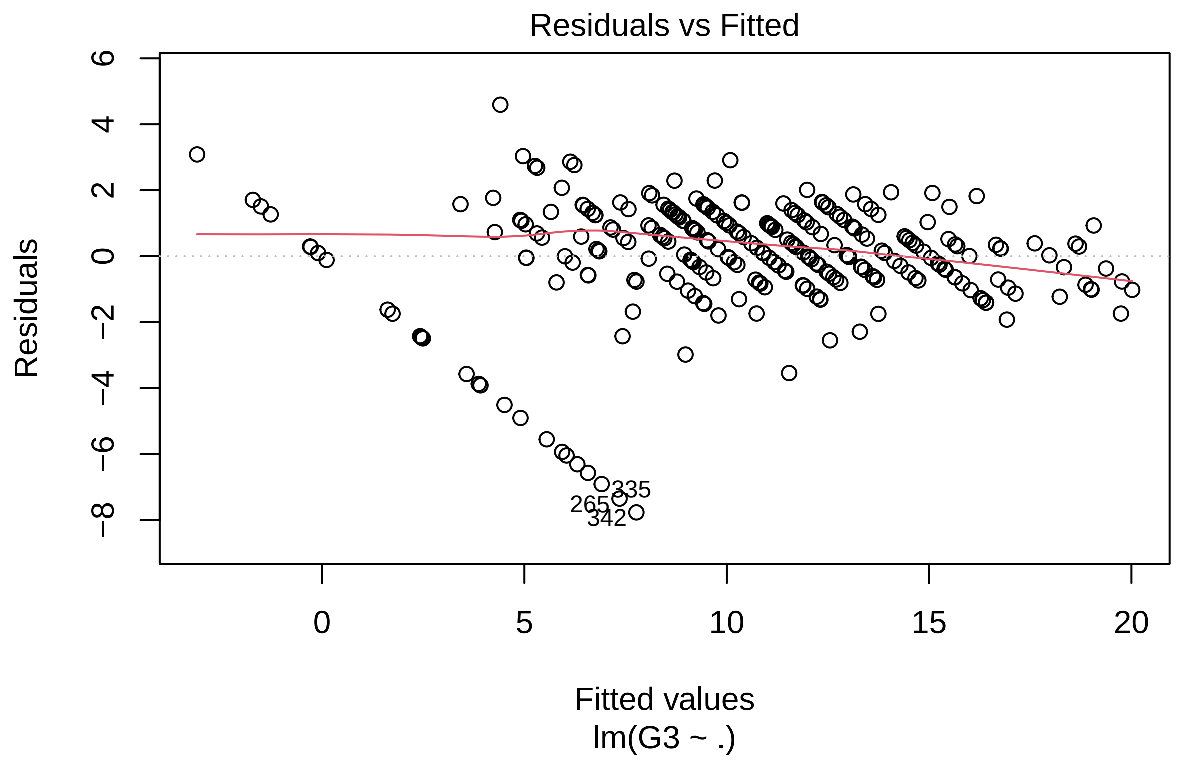
```r
# Histogram of residuals
ggplot(res,aes(res)) +  geom_histogram(fill='blue',alpha=0.5)
```

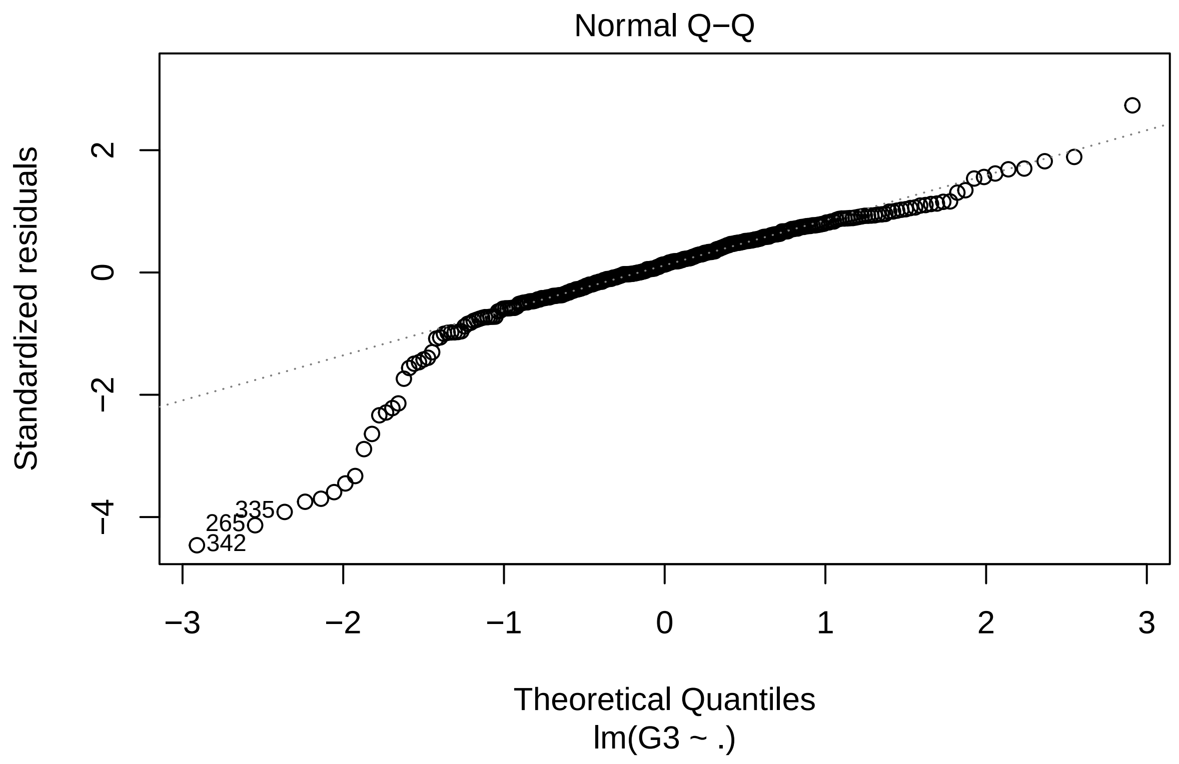
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

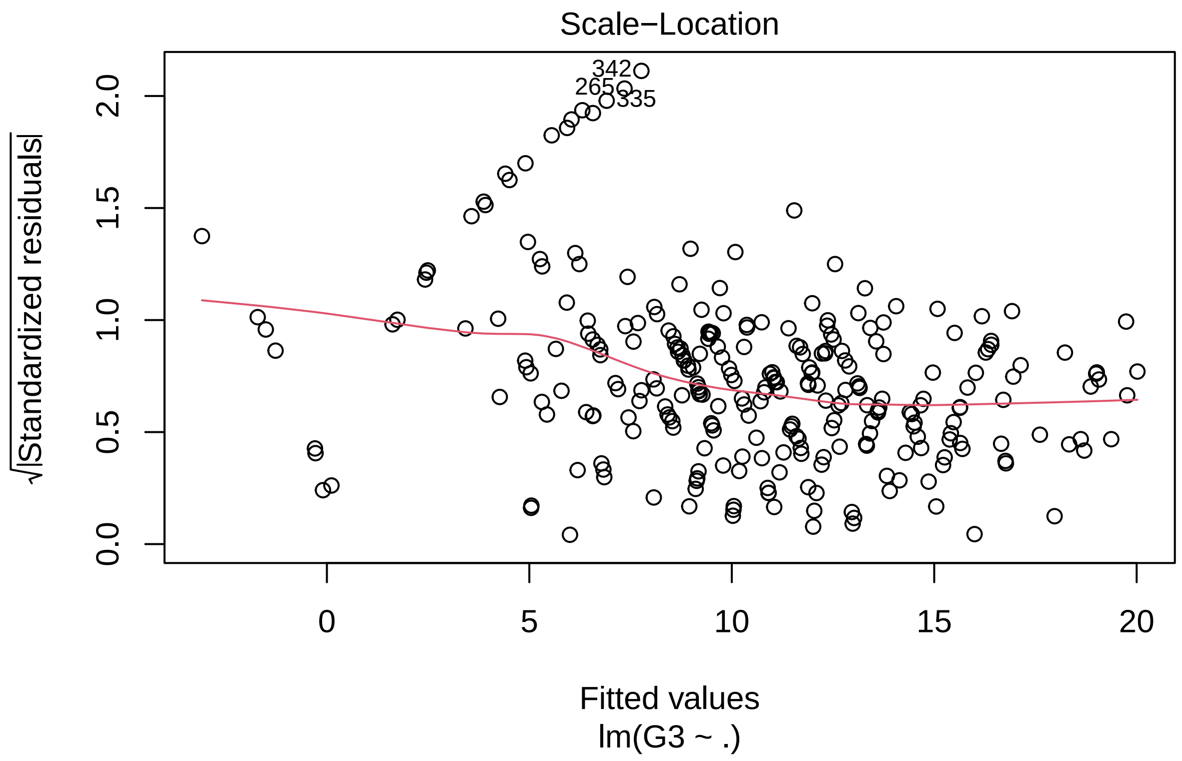


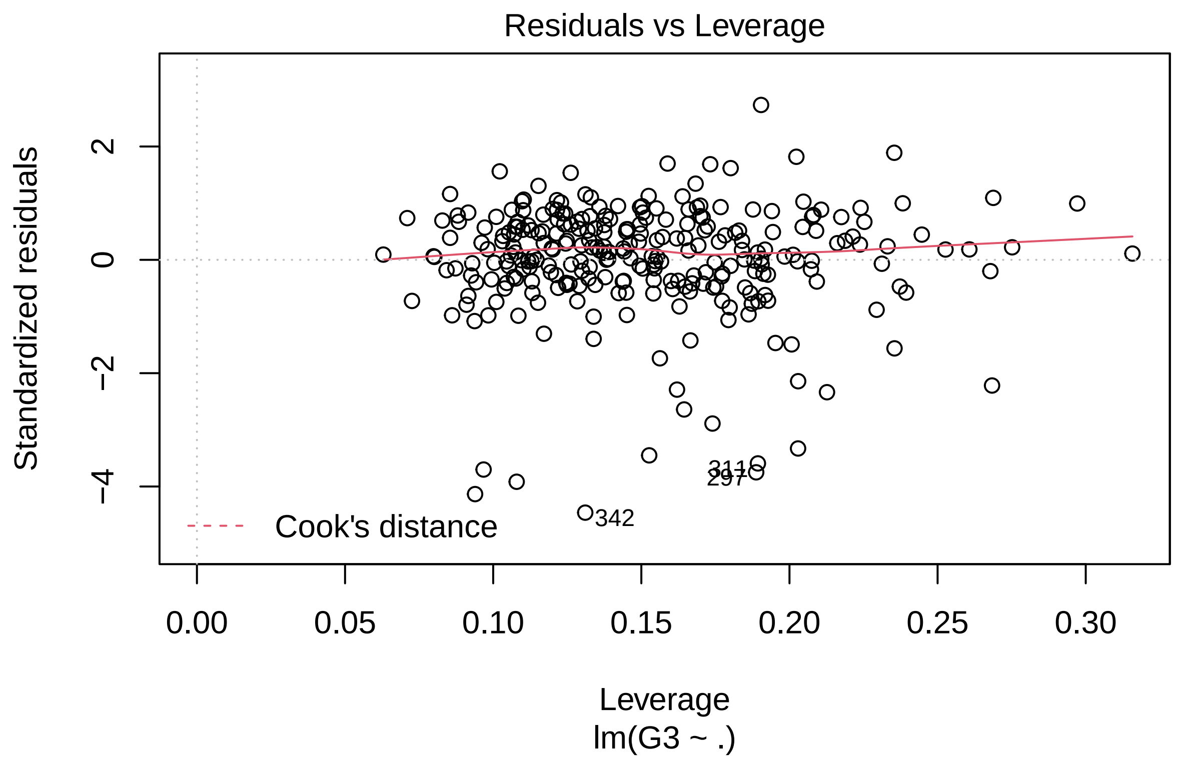
Looks like there are some suspicious residual values that have a value less than -5. We can further explore this by just calling plot on our model. What these plots represent is outside the course of this lecture, but it’s covered in ISLR, as well as the Wikipedia page on Regression Validation.

```r
plot(model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(G3 ~ .)
335
265
342
6
4
2
0
−2
−4
−6
−8
0
5
10
15
20


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(G3 ~ .)
335
265
342
2
0
−2
−4
−3
−2
−1
0
1
2
3


Scale−Location
342
265
335
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
5
10
15
20
Fitted values
lm(G3 ~ .)



Basically after looking at these plots, what you will realize is that our model (behaving as a continuous line, predicted students would get negative scores on their test! Let's make these all zeros when running our results against our predictions.

Predictions

Let's test our model by predicting on our testing set:

```
G3.predictions <- predict(model,test)
```

Now we can get the root mean squared error (RMSE), a standardized measure of how off we were with our predicted values:

```
results <- cbind(G3.predictions,test$G3)
colnames(results) <- c('pred','real')
results <- as.data.frame(results)
```

Now let's take care of negative predictions! Lots of ways to this, here's a more complicated way, but its a good example of creating a custom function for a custom problem:

So, let's define a function that converts negative values to zero:

```
to_zero <- function(x){
    if (x < 0){
        return(0)
    }else{
        return(x)
    }
}
```

And let's apply our function to the predictions series:

```
results$pred <- sapply(results$pred,to_zero)
```

There's lots of ways to evaluate the prediction values, for example the MSE (mean squared error):

```
mse <- mean((results$real-results$pred)^2)
print(mse)
```

```
## [1] 4.411405
```

Or the root mean squared error (RMSE):

```
mse^0.5
```

```
## [1] 2.100335
```

Or just the R-Squared Value for our model (just for the predictions):

```
SSE = sum((results$pred - results$real)^2)
SST = sum( (mean(df$G3) - results$real)^2)

R2 = 1 - SSE/SST
R2
```

```
## [1] 0.7779023
```

Conclusion

You should now feel comfortable with the R syntax for a Linear Regression. If some of the plots or math did not make sense to you, make sure to review ISLR and the relevant Wikipedia pages. There is no real substitute for taking the time to read about this material

Up next is an exercise to test your knowledge!