#Data Analysis and Visualisation using R

The purpose of this documenent is to provide data analysis and visualisation using R of students performance in exams. The dataset used in this overview was taken from: https://www.kaggle.com/spscientist/students-performance-in-exams

##Import libraries

##Load the dataset

```r
data<-read.csv("StudentsPerformance.csv")
# Checking for missing values:
cat("There are", sum(is.na(data)), "missing values.")
```

```
## There are 0 missing values.
```

Data Manipulation Before working with the data

```r
# Converting raw data into a tibble
spdata <- as_tibble(data)
# Converting appropriate categorical data to ordinal data
paredu <- ordered(spdata$parental.level.of.education, levels = c("some high school", "high school", "some college", "associate's degree", "bachelor's degree", "master's degree"))
```

Grading Scale

The grading scale are as follows:

A-> 90-100 B-> 80-89 C-> 70-79 D-> 60-69 F-> 0-59

```r
# New grade columns were created based on corresponding scores:
spdata_with_grades <- spdata %>%
  mutate(math.grade = case_when(math.score < 60 ~ "F",
                                math.score >= 60 & math.score <= 69 ~ "D",
                                math.score >= 70 & math.score <= 79 ~ "C",
                                math.score >= 80 & math.score <= 89 ~ "B",
                                math.score >= 90 & math.score <= 100 ~ "A"),
         reading.grade = case_when(reading.score < 60 ~ "F",
                                   reading.score >= 60 & reading.score <= 69 ~ "D",
                                   reading.score >= 70 & reading.score <= 79 ~ "C",
                                   reading.score >= 80 & reading.score <= 89 ~ "B",
                                   reading.score >= 90 & reading.score <= 100 ~ "A"),
         writing.grade = case_when(writing.score < 60 ~ "F",
                                   writing.score >= 60 & writing.score <= 69
~ "D",
                                   writing.score >= 70 & writing.score <= 79
~ "C",
                                   writing.score >= 80 & writing.score <= 89
~ "B",
                                   writing.score >= 90 & writing.score <= 100
~ "A"))
# The new columns were converted to factors with levels using lapply:
grades <- c("math.grade", "reading.grade", "writing.grade")
spdata_with_grades[grades] <- lapply(spdata_with_grades[grades], factor)
str(spdata_with_grades)
```

```
## tibble [1,000 x 11] (S3: tbl_df/tbl/data.frame)
##  $ gender                     : chr [1:1000] "female" "female" "female"
"male" ...
##  $ race.ethnicity             : chr [1:1000] "group B" "group C" "group B"
"group A" ...
##  $ parental.level.of.education: chr [1:1000] "bachelor's degree" "some
college" "master's degree" "associate's degree" ...
##  $ lunch                      : chr [1:1000] "standard" "standard"
"standard" "free/reduced" ...
##  $ test.preparation.course    : chr [1:1000] "none" "completed" "none"
"none" ...
##  $ math.score                 : int [1:1000] 72 69 90 47 76 71 88 40 64 38
...
##  $ reading.score              : int [1:1000] 72 90 95 57 78 83 95 43 64 60
...
##  $ writing.score              : int [1:1000] 74 88 93 44 75 78 92 39 67 50
...
##  $ math.grade                 : Factor w/ 5 levels "A","B","C","D",..: 3 4
1 5 3 3 2 5 4 5 ...
##  $ reading.grade              : Factor w/ 5 levels "A","B","C","D",..: 3 1
1 5 3 2 1 5 4 4 ...
##  $ writing.grade              : Factor w/ 5 levels "A","B","C","D",..: 3 2
1 5 3 3 1 5 4 5 ...
```

The new data had to be written to a new file inorder to keep my original data intact.

```
# Writing to a new file:
write.csv(spdata_with_grades, file = "C:/Users/Romeo/Desktop/University of
Guyana/4th year 2nd
Semester/CSE4202/DataAnalysisOfStudentsPerformance/StudentsPerformance_man.cs
v", row.names = FALSE, col.names = TRUE)
```

```
## Warning in write.csv(spdata_with_grades, file = "C:/Users/Romeo/
## Desktop/University of Guyana/4th year 2nd Semester/CSE4202/

## DataAnalysisOfStudentsPerformance/StudentsPerformance_man.csv", : attempt
to set

## 'col.names' ignored
```

Q.1 What does the dataset involve?

```
str(data)
```

```
## 'data.frame':    1000 obs. of  8 variables:

##  $ gender                     : chr  "female" "female" "female" "male" ...

##  $ race.ethnicity             : chr  "group B" "group C" "group B" "group
A" ...

##  $ parental.level.of.education: chr  "bachelor's degree" "some college"
"master's degree" "associate's degree" ...

##  $ lunch                      : chr  "standard" "standard" "standard"
"free/reduced" ...

##  $ test.preparation.course    : chr  "none" "completed" "none" "none" ...

##  $ math.score                 : int  72 69 90 47 76 71 88 40 64 38 ...

##  $ reading.score              : int  72 90 95 57 78 83 95 43 64 60 ...

##  $ writing.score              : int  74 88 93 44 75 78 92 39 67 50 ...
```

Interpretations/Conlusion: As shown in the output there are 1000 obs. of 8 variables.

Q2. What is the general statistical description of this dataset?

```
summary(data)
```

```
##     gender          race.ethnicity     parental.level.of.education

##  Length:1000        Length:1000        Length:1000

##  Class :character   Class :character   Class :character

##  Mode  :character   Mode  :character   Mode  :character

##

##

##

##     lunch           test.preparation.course   math.score      reading.score

##  Length:1000        Length:1000             Min.   :  0.00   Min.   :
17.00

##  Class :character   Class :character        1st Qu.: 57.00   1st Qu.:
59.00

##  Mode  :character   Mode  :character        Median : 66.00   Median :
70.00
```

```
##                                              Mean   : 66.09   Mean   :
69.17
##                                              3rd Qu.: 77.00   3rd Qu.:
79.00
##                                              Max.   :100.00   Max.
:100.00
##  writing.score
##  Min.   : 10.00
##  1st Qu.: 57.75
##  Median : 69.00
##  Mean   : 68.05
##  3rd Qu.: 79.00
##  Max.   :100.00
```

Interpretations/Conlusion:

Q3. What is the number of occurance for students that pass math with a score of 65 ?

```
with(data, table(math.score))
```

```
## math.score
##   0   8  18  19  22  23  24  26  27  28  29  30  32  33  34  35  36  37
38  39
##   1   1   1   1   1   1   1   1   2   1   3   2   3   1   2   5   2   4
3   4
##  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57
58  59
##  10   6   6   5   9   9  11  11  11  17  15  11  18  24  18  18   9  18
25  32
##  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77
78  79
##  16  27  35  26  20  36  24  26  26  32  18  26  18  27  25  21  21  24
14  22
##  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97
98  99
##  17  22  18   8  11  14   8  16  15   6   8   9   6   4   7   2   3   6
3   3
## 100
##   7
```

Interpretations/Conlusion: We can see 37 occurrence of students who pass math with a score of 37.

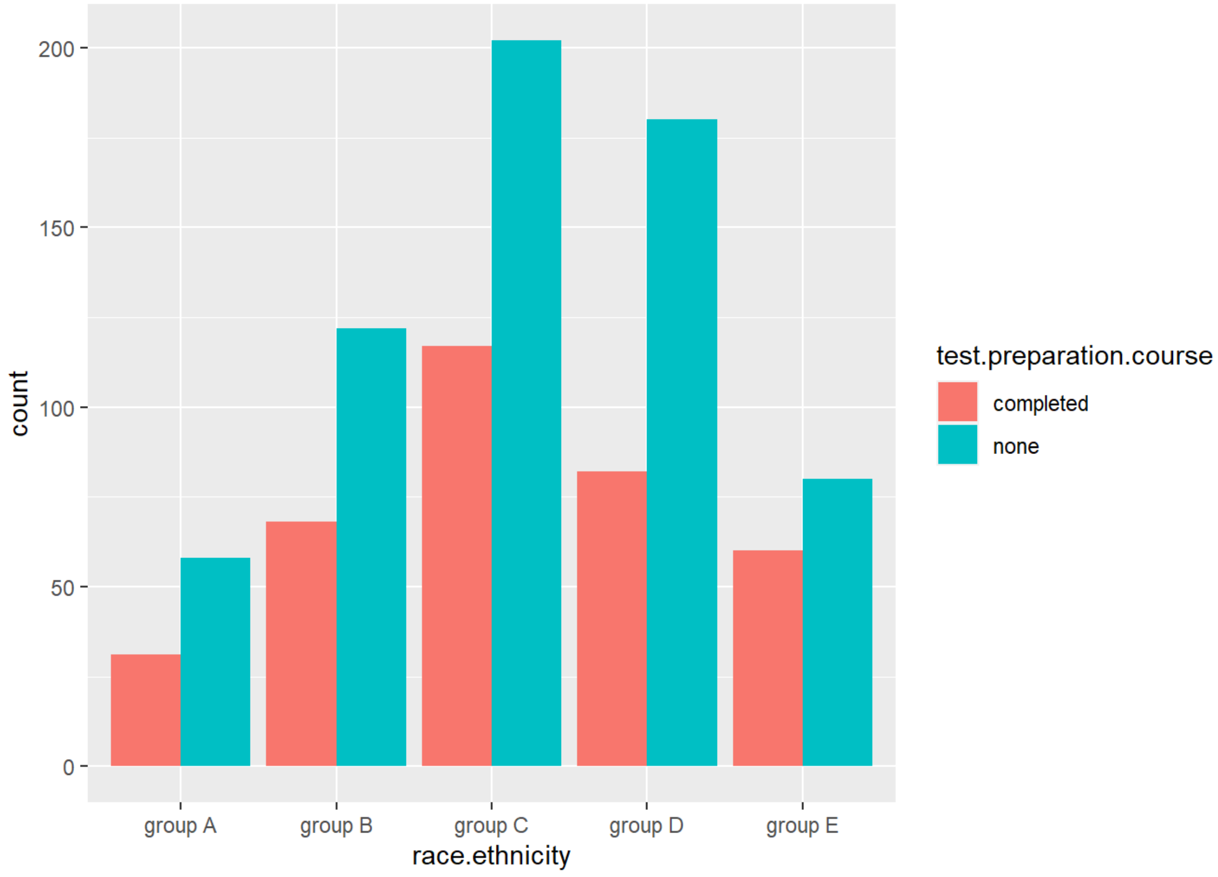
Q4. Which gender are most prepare for exams?

```
table(data$test.preparation.course, data$gender)
##
##              female male
##   completed     184  174
##   none          334  308
```

Interpretations/Conlusion: From the table shown there it clearly shows that female are most prepare for exams.

Q5. Does preparation make students perform better?

```
# A side-by-side barchart of race.ethnicity by test.preparation.course



ggplot(data, aes(x = race.ethnicity, fill = test.preparation.course)) +
  geom_bar(position = "dodge")
```



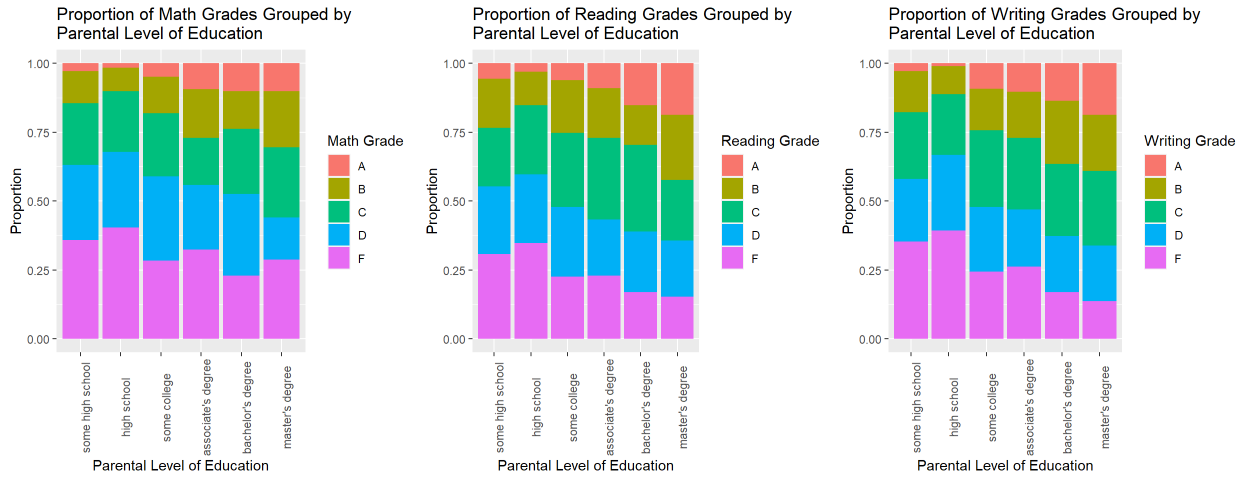
Interpretations/Conlusion:

Q6. What are the three highest proportion of parentel level of Education are ?

Interpretations/Conlusion: Highest proportion of parentel level of Education is ‘Some college’, ‘associate’s degreee’ and ‘high school’

Q7. Does parent’s education background influenced student’s performance in exam?

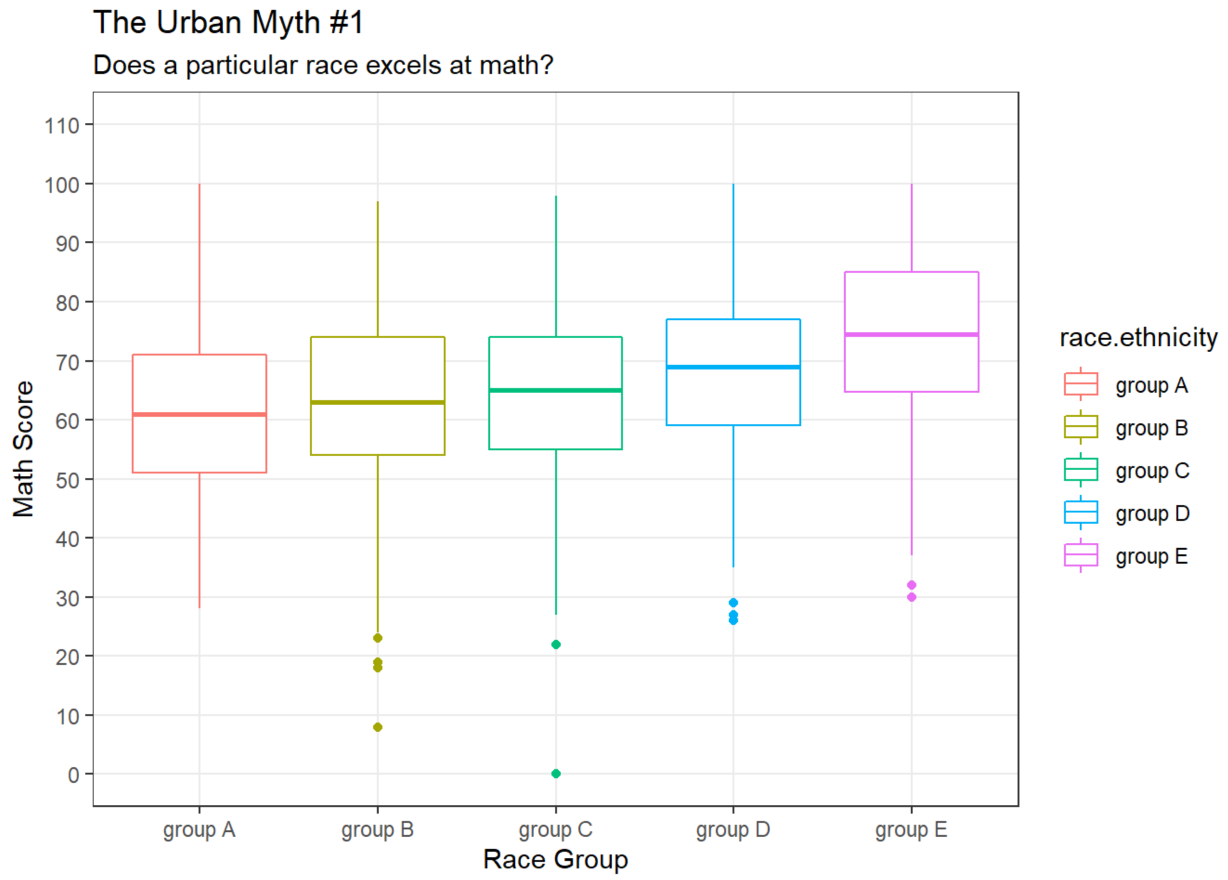
```
# Three proportional graphs where created where the students scores were
compared with the parent level of education



# Long Title Wrap function:
wrapper <- function(x, ...)
{
  paste(strwrap(x, ...), collapse = "\n")
}


# Proportional graph of math grades vs. parental level of education
math_grades_vs_paredu_prop <-ggplot(spdata_with_grades, aes(x = paredu, fill
= math.grade)) +
  geom_bar(position = "fill") +
  ggtitle(wrapper("Proportion of Math Grades Grouped by Parental Level of
Education", width = 40)) +
  xlab("Parental Level of Education") +
  ylab("Proportion") +
  labs(fill = "Math Grade") +
  theme(axis.text.x = element_text(angle = 90))
# Proportional graph of reading grades vs. parental level of education
read_grades_vs_paredu_prop <- ggplot(spdata_with_grades, aes(x = paredu, fill
= reading.grade)) +
  geom_bar(position = "fill") +
  ggtitle(wrapper("Proportion of Reading Grades Grouped by Parental Level of
Education", width = 40)) +
  xlab("Parental Level of Education") +
  ylab("Proportion") +
  labs(fill = "Reading Grade") +
  theme(axis.text.x = element_text(angle = 90))

# Proportional graph of writing grades vs. parental level of education

writ_grades_vs_paredu_prop <- ggplot(spdata_with_grades, aes(x = paredu, fill
= writing.grade)) +

  geom_bar(position = "fill") +

  ggtitle(wrapper("Proportion of Writing Grades Grouped by Parental Level of
Education", width = 40)) +

  xlab("Parental Level of Education") +

  ylab("Proportion") +

  labs(fill = "Writing Grade") +

  theme(axis.text.x = element_text(angle = 90))

grid.arrange(math_grades_vs_paredu_prop, read_grades_vs_paredu_prop,
writ_grades_vs_paredu_prop, ncol = 3)
```



Interpretations/Conlusion: From this view, we can see that the higher three levels of parental education (master’s degree, bachelor’s degree, and associate’s degree) tend to have a higher proportion of students with As, Bs, and Cs compared to the lower three levels (some college, high school, and some high school).

Q8. Does a particular race excels at math?

```
# Box plot base on score for math with colours


ggplot(data, mapping=aes(x=race.ethnicity, y=math.score, col=race.ethnicity
))+

  theme_bw() +

  geom_boxplot()+

  scale_y_continuous(limits=c(0,110),breaks = seq(0,110,10))+

  labs(title="The Urban Myth #1", subtitle="Does a particular race excels at
math?", x="Race Group",        y="Math Score")+
  theme(panel.grid.minor = element_blank())
```


The Urban Myth #1

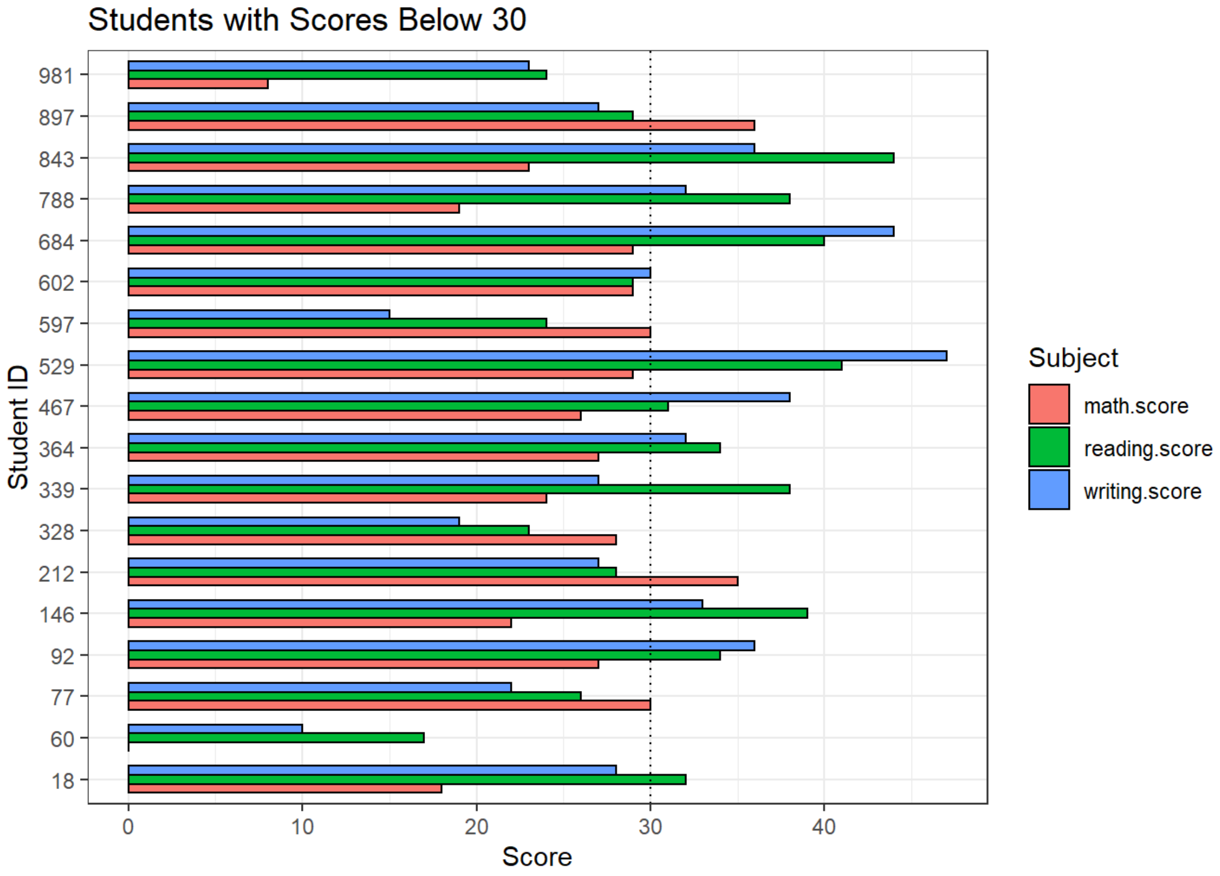
Does a particular race excels at math?

Interpretations/Conlusion: By looking at the graph, group E may excels from the rest.

Q9. Are students struggling in all 3 areas or just 1 or 2 ?

```
# A side-by-side barchart of race.ethnicity by test.preparation.course

# Create a Student ID field for a unique identifier:
spdata_with_grades_ID <- tibble::rowid_to_column(spdata_with_grades, "ID")

# Convert new ID variable to factor:
spdata_with_grades_ID$ID <- as.factor(spdata_with_grades_ID$ID)

# Filter for the <= 30 score students that need help:
math_below30 <- spdata_with_grades_ID %>%
  filter(math.score <= 30)
reading_below30 <- spdata_with_grades_ID %>%
  filter(reading.score <= 30)
```

```r
writing_below30 <- spdata_with_grades_ID %>%
  filter(writing.score <= 30)

# Create a union for these 3 sets of data:
students_below30 <- list(math_below30, reading_below30, writing_below30) %>%
  reduce(union, by = "ID")

# Visualize the data:
students_below30.long <- gather(students_below30, key = "Subject", value =
"Score", -ID, -gender, -race.ethnicity, -parental.level.of.education, -lunch,
-test.preparation.course, -math.grade, -reading.grade, -writing.grade)
ggplot(students_below30.long, aes(x = ID, y = Score, fill = Subject)) +
  geom_col(position = "dodge", color = "black", width = 0.65) +
  geom_hline(yintercept = 30, linetype = "dotted") +
  ggtitle("Students with Scores Below 30") +
  xlab("Student ID") +
  theme_bw() +
  coord_flip()
```

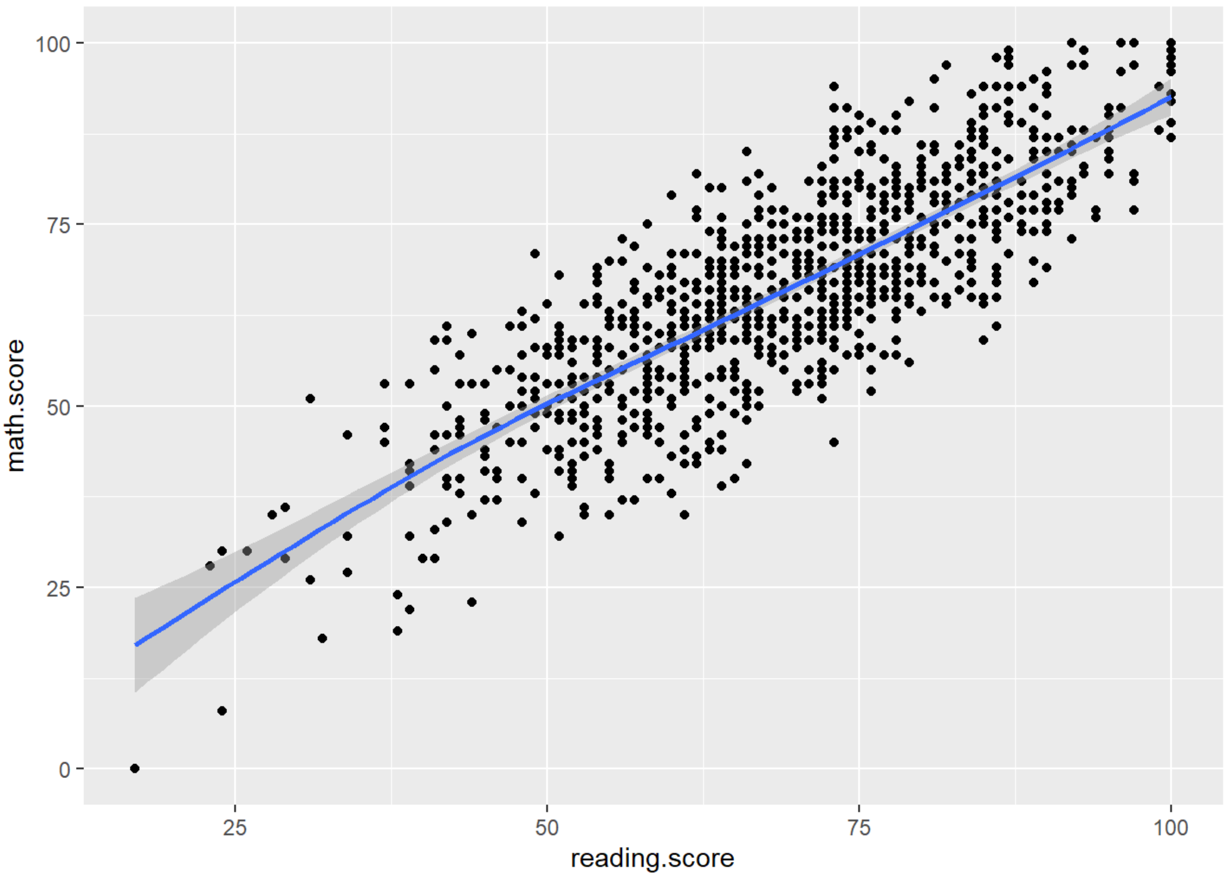


Interpretations/Conlusion: There are 18 students that have a score of 30 or below in at least 1 subject. As we can see, there are some students who score 30 or below in all 3 subjects and some who score higher in one or two other subjects. One student, #60, scored 0 in math and also has the lowest reading and writing scores out of any other student.

Q.10 what is the relationship between maths and reading scores?

```
ggplot(data, aes(reading.score, math.score)) + geom_point() +
        geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Interpretations/Conlusion: At the math, the correlation is very strong.

##Prepaing for modeling

```
data$gender <- as.factor(data$gender)
data$math.score <- as.factor(data$math.score)
data$reading.score <- as.factor(data$reading.score)
data$writing.score <- as.factor(data$writing.score)


set.seed(134)
sampleSize <- floor(.75*nrow(data))
```

```
trainIndexes <- sample(seq_len(nrow(data)), sampleSize, replace = FALSE)
train <- data[trainIndexes, ]
test <- data[-trainIndexes, ]
```

Modeling

```
linear_mod1<-lm(math.score~reading.score, data = data)
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a
factor
## response will be ignored
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
```

```
linear_mod1
```

```
##
## Call:
## lm(formula = math.score ~ reading.score, data = data)
##
## Coefficients:
##      (Intercept)   reading.score23   reading.score24   reading.score26
##             1.00              9.00              6.00             11.00
##  reading.score28   reading.score29   reading.score31   reading.score32
##            15.00             13.00             19.00              2.00
##  reading.score34   reading.score37   reading.score38   reading.score39
##            13.50             28.33              4.50             18.50
##  reading.score40   reading.score41   reading.score42   reading.score43
##            10.00             24.50             27.00             26.78
##  reading.score44   reading.score45   reading.score46   reading.score47
##            23.25             25.00             25.83             32.75
##  reading.score48   reading.score49   reading.score50   reading.score51
##            30.60             32.90             33.57             31.06
##  reading.score52   reading.score53   reading.score54   reading.score55
##            29.44             28.69             36.56             34.13
##  reading.score56   reading.score57   reading.score58   reading.score59
##            35.31             38.35             35.43             34.18
##  reading.score60   reading.score61   reading.score62   reading.score63
##            40.29             36.50             41.82             40.80
##  reading.score64   reading.score65   reading.score66   reading.score67
##            41.97             39.05             44.52             45.23
##  reading.score68   reading.score69   reading.score70   reading.score71
##           46.74           45.73           44.92           47.80
##  reading.score72  reading.score73  reading.score74  reading.score75
##           46.03           50.43           51.73           52.35
##  reading.score76  reading.score77  reading.score78  reading.score79
##           51.12           52.50           52.81           52.74
##  reading.score80  reading.score81  reading.score82  reading.score83
##           55.43           56.56           57.65           56.21
##  reading.score84  reading.score85  reading.score86  reading.score87
##           59.35           60.28           57.16           67.85
##  reading.score88  reading.score89  reading.score90  reading.score91
##           60.11           61.33           62.59           61.50
##  reading.score92  reading.score93  reading.score94  reading.score95
##           65.40           68.50           60.00           66.88
##  reading.score96  reading.score97  reading.score99 reading.score100
##           74.50           67.40           70.00           74.53
```

```
#Simple linear model
linear_mod2<-lm(math.score~reading.score, data = train)
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a
factor
## response will be ignored
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
```

```
linear_mod2
```

```
##
## Call:
## lm(formula = math.score ~ reading.score, data = train)
##
## Coefficients:
##      (Intercept)  reading.score24  reading.score28  reading.score29
##            6.00            -1.50            6.00            7.00
##  reading.score31  reading.score32  reading.score34  reading.score37
##           10.00            -4.00           -1.00           24.00
##  reading.score39  reading.score40  reading.score41  reading.score42
##            9.00            1.00           15.25           19.17
##  reading.score43  reading.score44  reading.score45  reading.score46
##           17.71           27.50           13.80           16.83
##  reading.score47  reading.score48  reading.score49  reading.score50
##           26.33           21.60           24.25           24.67
##  reading.score51   reading.score52   reading.score53   reading.score54
##            19.67             19.86             18.45             27.37
##  reading.score55   reading.score56   reading.score57   reading.score58
##            26.33             28.58             30.29             25.10
##  reading.score59   reading.score60   reading.score61   reading.score62
##            24.58             32.57             26.71             33.76
##  reading.score63   reading.score64   reading.score65   reading.score66
##            31.22             33.89             31.31             35.57
##  reading.score67   reading.score68   reading.score69   reading.score70
##            36.50             37.93             36.75             36.39
##  reading.score71   reading.score72   reading.score73   reading.score74
##            36.31             37.52             41.14             42.21
##  reading.score75   reading.score76   reading.score77   reading.score78
##            43.85             41.50             44.94             45.47
##  reading.score79   reading.score80   reading.score81   reading.score82
##            44.33             45.90             48.24             48.50
##  reading.score83   reading.score84   reading.score85   reading.score86
##            47.90             48.94             48.50             51.70
##  reading.score87   reading.score88   reading.score89   reading.score90
##            57.82             51.43             52.33             52.62
##  reading.score91   reading.score92   reading.score93   reading.score94
##            52.25             54.78             54.75             52.50
##  reading.score95   reading.score96   reading.score97   reading.score99
##            57.87             64.50             64.00             59.00
## reading.score100
##            65.83
```

```
linear_mod3<-lm(writing.score~reading.score, data = train)
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a factor
## response will be ignored
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
```

```
linear_mod3
```

```
##
## Call:
## lm(formula = writing.score ~ reading.score, data = train)
##
## Coefficients:
```

```
##      (Intercept)   reading.score24   reading.score28   reading.score29
##        2.000e+00         1.524e-12         2.000e+00         2.000e+00
##  reading.score31   reading.score32   reading.score34   reading.score37
##        8.000e+00         3.000e+00         5.500e+00         1.100e+01
##  reading.score39   reading.score40   reading.score41   reading.score42
##        6.000e+00         1.500e+01         1.600e+01         1.317e+01
##  reading.score43   reading.score44   reading.score45   reading.score46
##        1.357e+01         1.550e+01         1.460e+01         1.567e+01
##  reading.score47   reading.score48   reading.score49   reading.score50
##        2.233e+01         1.870e+01         1.825e+01         1.967e+01
##  reading.score51   reading.score52   reading.score53   reading.score54
##        2.058e+01         2.021e+01         2.345e+01         2.300e+01
##  reading.score55   reading.score56   reading.score57   reading.score58
##        2.467e+01         2.500e+01         2.543e+01         2.805e+01
##  reading.score59   reading.score60   reading.score61   reading.score62
##        2.983e+01         2.886e+01         3.000e+01         3.206e+01
##  reading.score63   reading.score64   reading.score65   reading.score66
##        3.278e+01         3.530e+01         3.492e+01         3.652e+01
##  reading.score67   reading.score68   reading.score69   reading.score70
##        3.705e+01         3.780e+01         3.975e+01         4.083e+01
##  reading.score71   reading.score72   reading.score73   reading.score74
##        4.323e+01         4.226e+01         4.227e+01         4.367e+01
##  reading.score75   reading.score76   reading.score77   reading.score78
##        4.530e+01         4.580e+01         4.675e+01         4.774e+01
##  reading.score79   reading.score80   reading.score81   reading.score82
##        4.980e+01         4.680e+01         4.948e+01         5.221e+01
##  reading.score83   reading.score84   reading.score85   reading.score86
##        5.250e+01         5.471e+01         5.542e+01         5.690e+01
##  reading.score87   reading.score88   reading.score89   reading.score90
##        5.600e+01         5.657e+01         6.067e+01         5.823e+01
##  reading.score91   reading.score92   reading.score93   reading.score94
##        6.125e+01         5.967e+01         6.450e+01         6.200e+01
##  reading.score95   reading.score96   reading.score97   reading.score99
##        6.350e+01         6.650e+01         6.800e+01         6.850e+01
## reading.score100
##        6.958e+01
```