R Notebook

Import Libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(dplyr)
```

Load our Data

```
data <- read.csv("assets/StudentsPerformance.csv")
data%>%head()
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group B           bachelor's degree     standard
## 2 female        group C                some college     standard
## 3 female        group B             master's degree     standard
## 4   male        group A          associate's degree free/reduced
## 5   male        group C                some college     standard
## 6 female        group B          associate's degree     standard
##   test.preparation.course math.score reading.score writing.score
```

```
## 1                    none         72            72            74
## 2               completed         69            90            88
## 3                    none         90            95            93
## 4                    none         47            57            44
## 5                    none         76            78            75
## 6                    none         71            83            78
```

```
ndata<-mutate(data,mean=(`math.score`+`reading.score`+`writing.score`)/3) #mean of three score
table(data$parental.level.of.education)
```

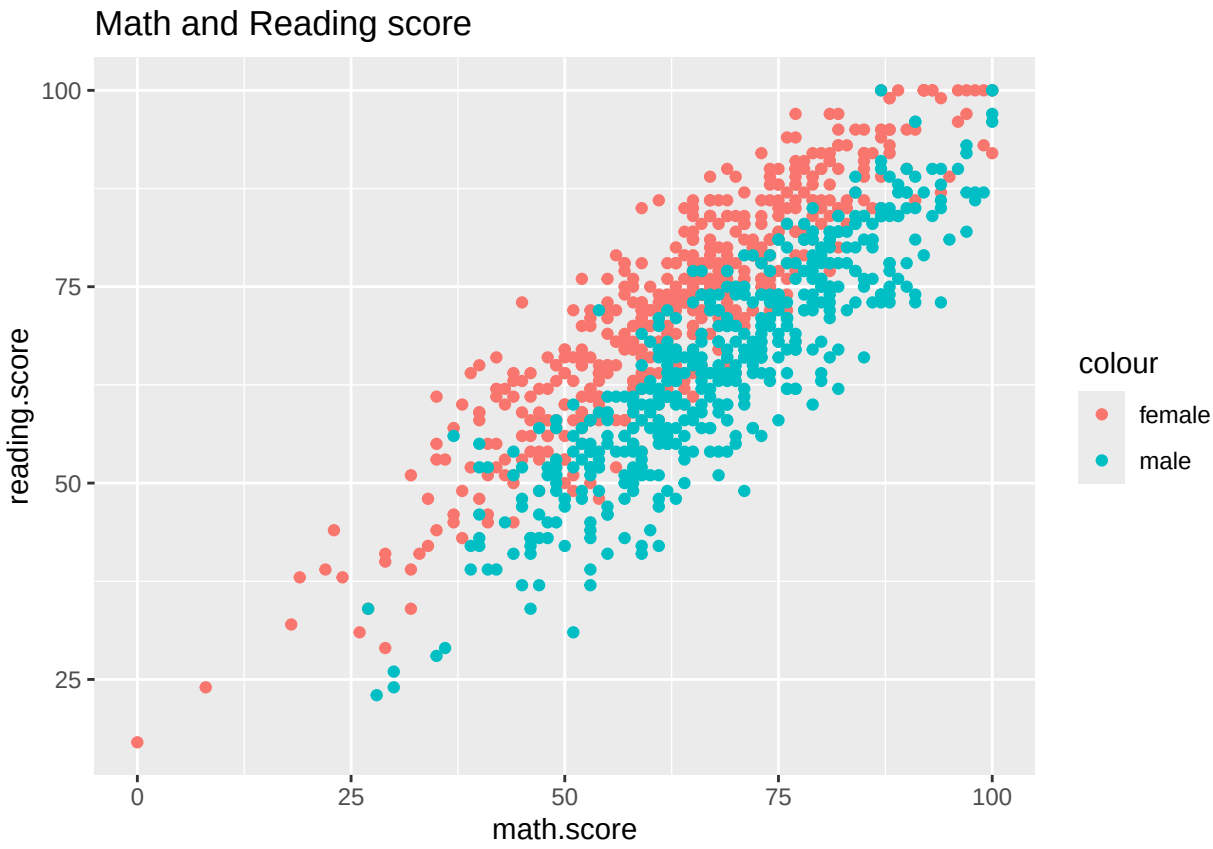
```
##
## associate's degree  bachelor's degree        high school    master's degree
##                222                118                196                 59
##       some college   some high school
##                226                179
```

```
table(data$lunch)
```

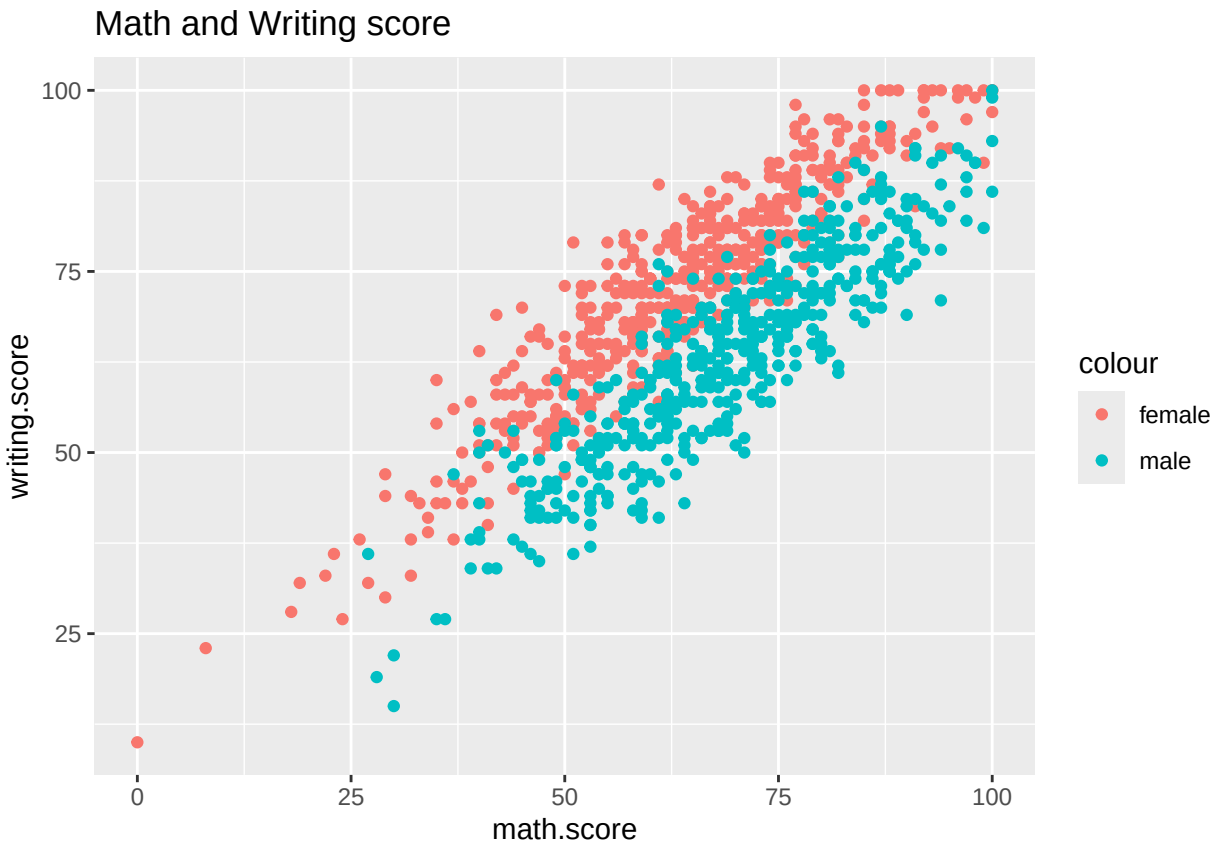
```
##
## free/reduced     standard
##          355          645
```

Comparing the Reading and Writing Scores again Math Scores

```
girl_data<-data%>%filter(gender=='female')
boy_data<-data%>%filter(gender=='male')
ggplot()+
  geom_point(girl_data,mapping=aes(`math.score`,`reading.score`,color='female'))+
  geom_point(boy_data,mapping=aes(`math.score`,`reading.score`,color='male'))+labs(title='Math and Readi
```

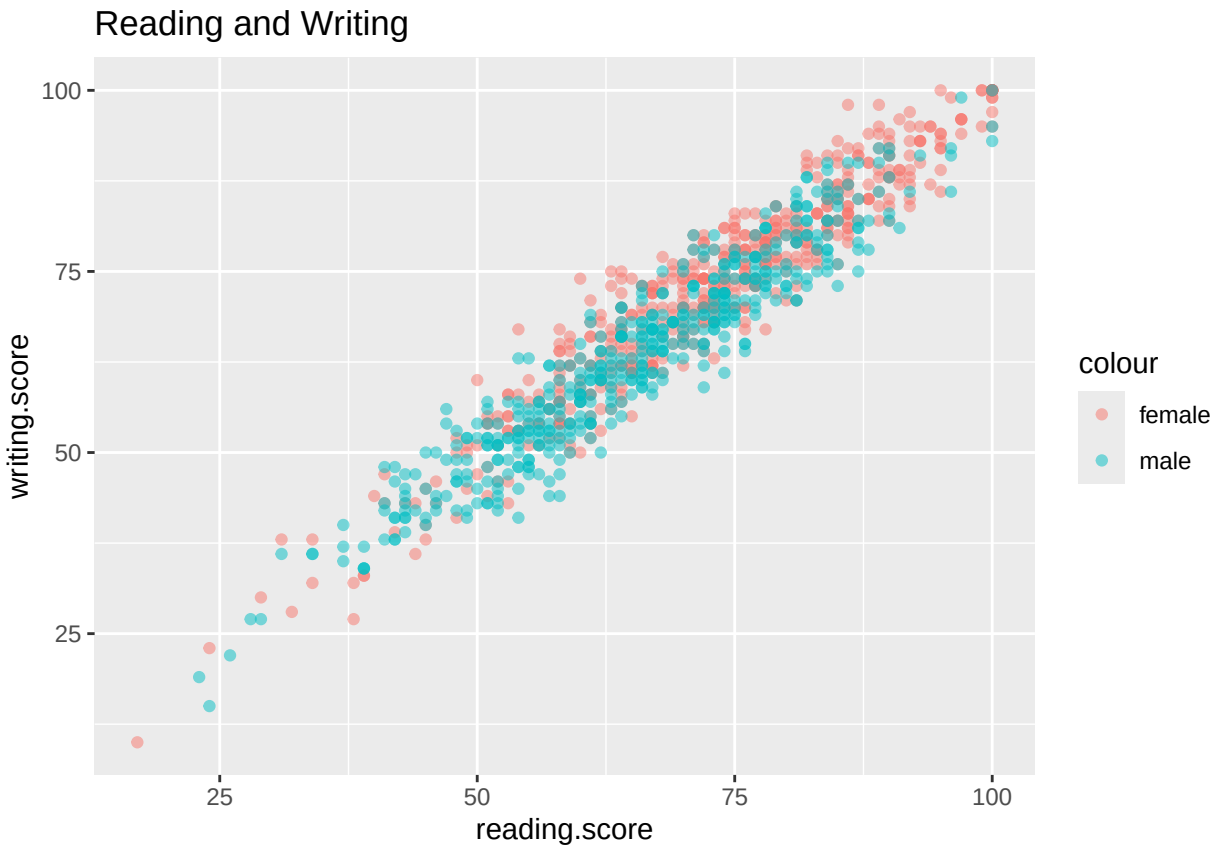



```
ggplot()+
  geom_point(girl_data,mapping=aes(`math.score`,`writing.score`,color='female'))+
  geom_point(boy_data,mapping = aes(`math.score`,`writing.score`,color='male'))+labs(title='Math and Wr
```

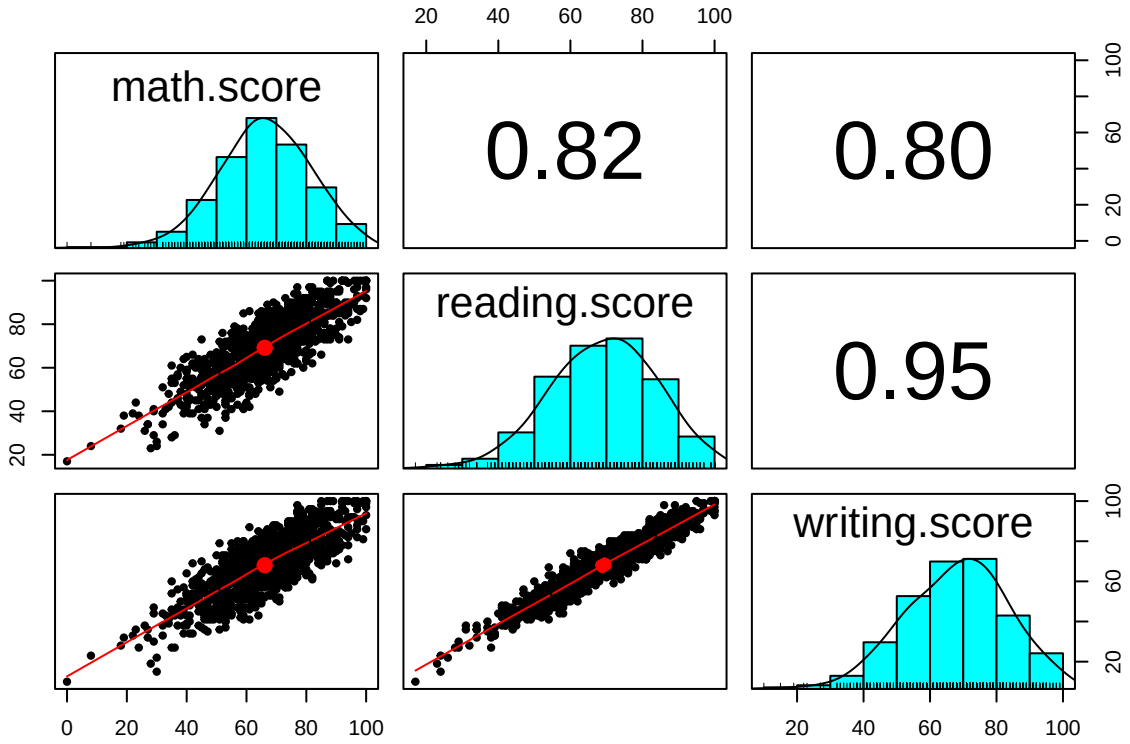


Now we comparing Reading and Writing Scores

```
ggplot()+
  geom_point(girl_data,mapping=aes(`reading.score`,`writing.score`,color='female'),alpha=1/2)+
  geom_point(boy_data,mapping = aes(`reading.score`,`writing.score`,color='male'),alpha=1/2)+labs(title=
```




```
pairs.panels(data[6:8])
```

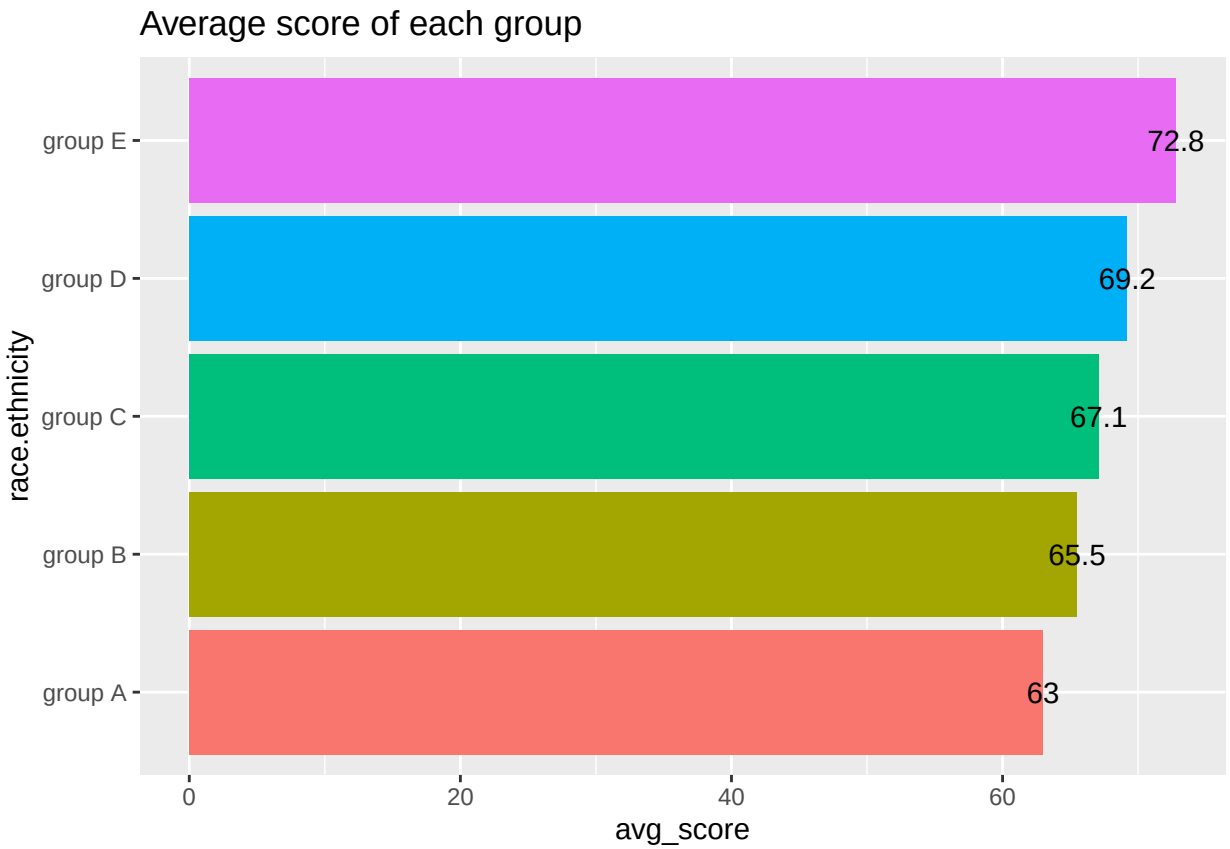


The plot above shows that females have a higher reading/wriitng score and males tend to have higher math scores

Averages Scores of Each Group

```
ndata%>%
group_by(race.ethnicity)%>%
summarize(avg_score=round(sum(mean)/n(),1))%>%
ggplot(aes(race.ethnicity,avg_score,fill=race.ethnicity))+geom_bar(stat='identity')+
geom_text(aes(label = avg_score))+coord_flip()+labs(title='Average score of each group')+guides(fill=F)
```

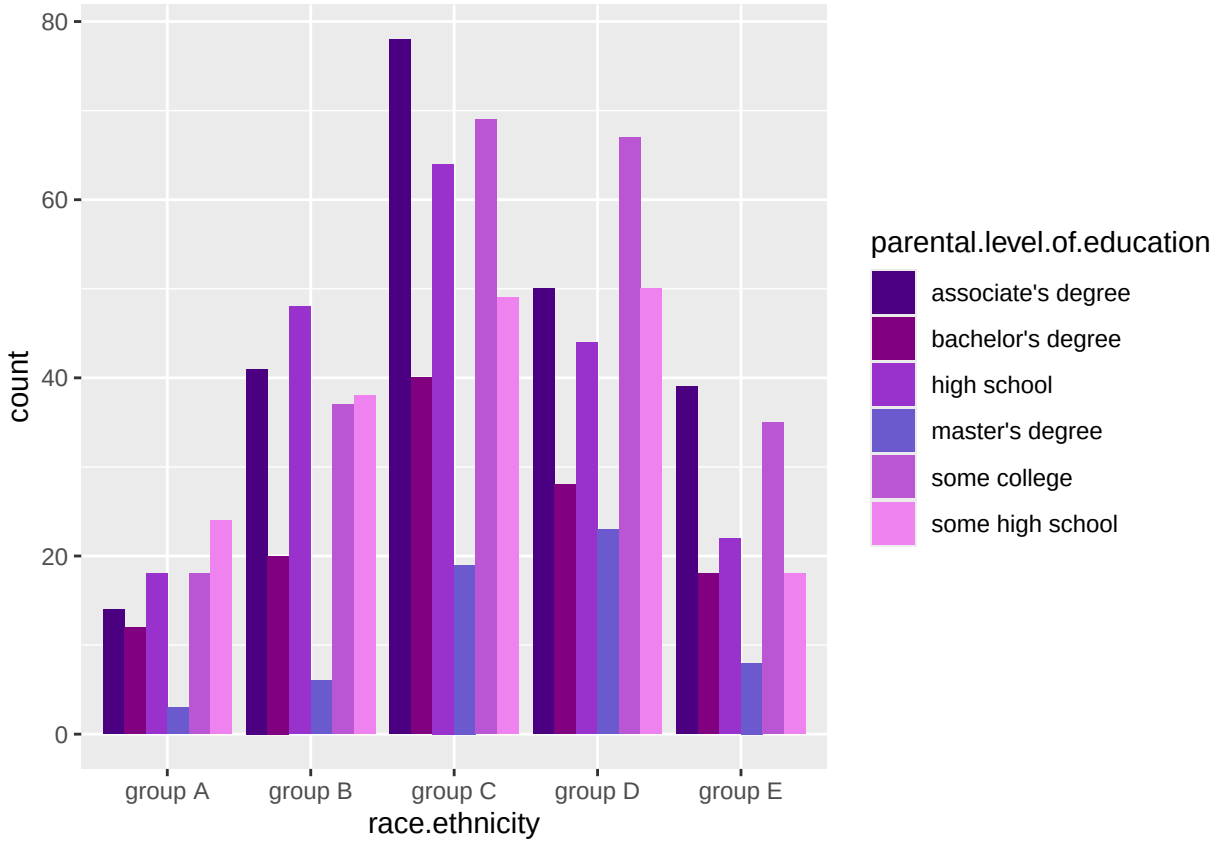
```
## Warning: The ‘<scale>‘ argument of ‘guides()‘ cannot be ‘FALSE‘. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Level of Education of Parents

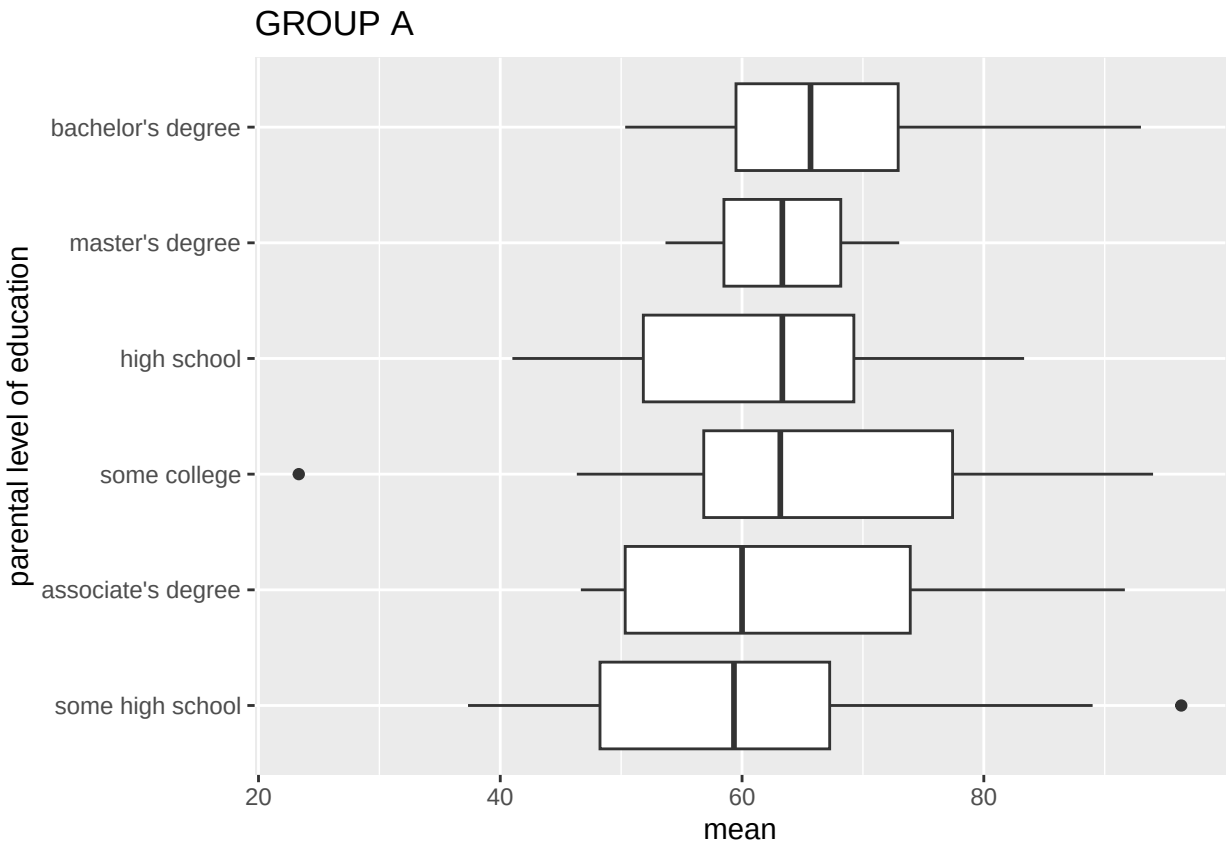
```
data%>%group_by(race.ethnicity,parental.level.of.education)%>%summarize(count=n())%>%
ggplot()+geom_col(aes(race.ethnicity,count,fill=parental.level.of.education),position='dodge')+scale_fi
```

```
## `summarise()` has grouped output by 'race.ethnicity'. You can override using
## the `.groups` argument.
```

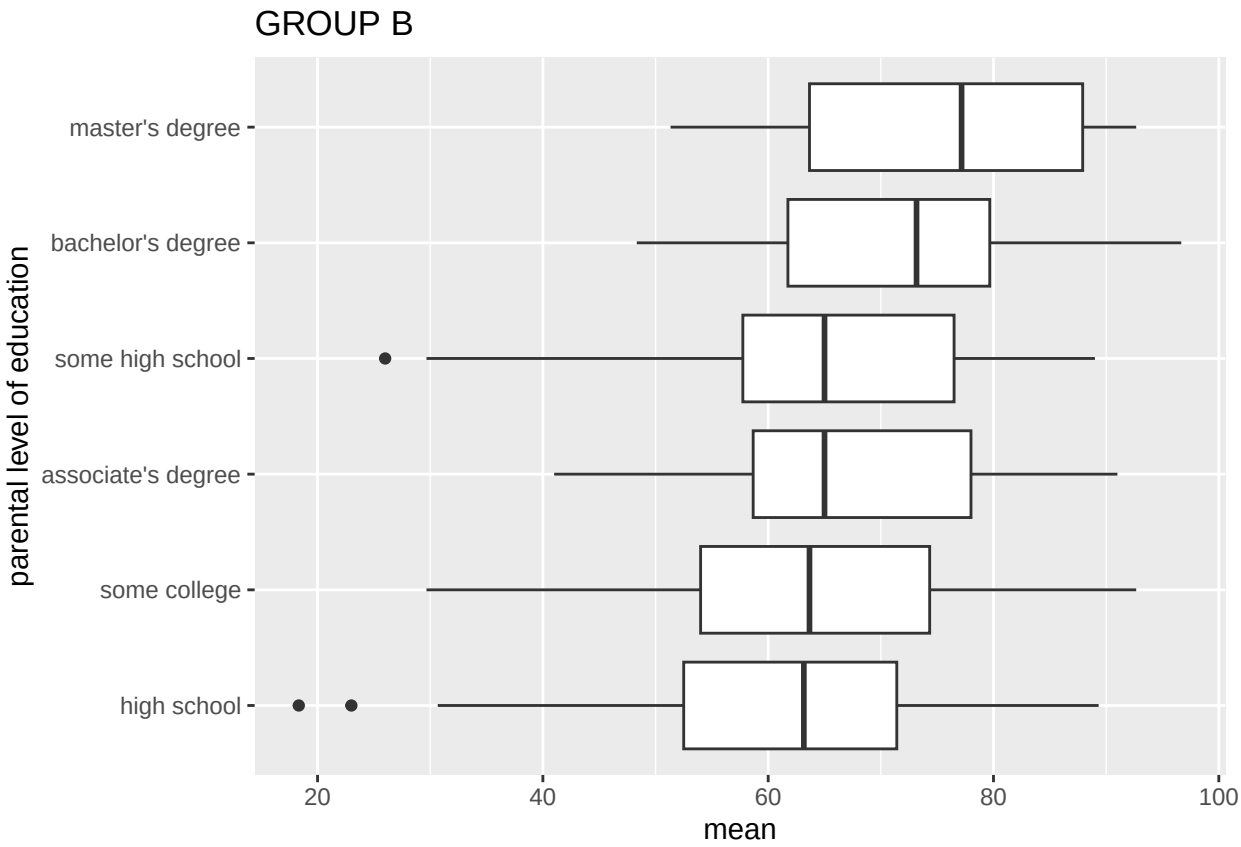


Education of Parents and Mean Score

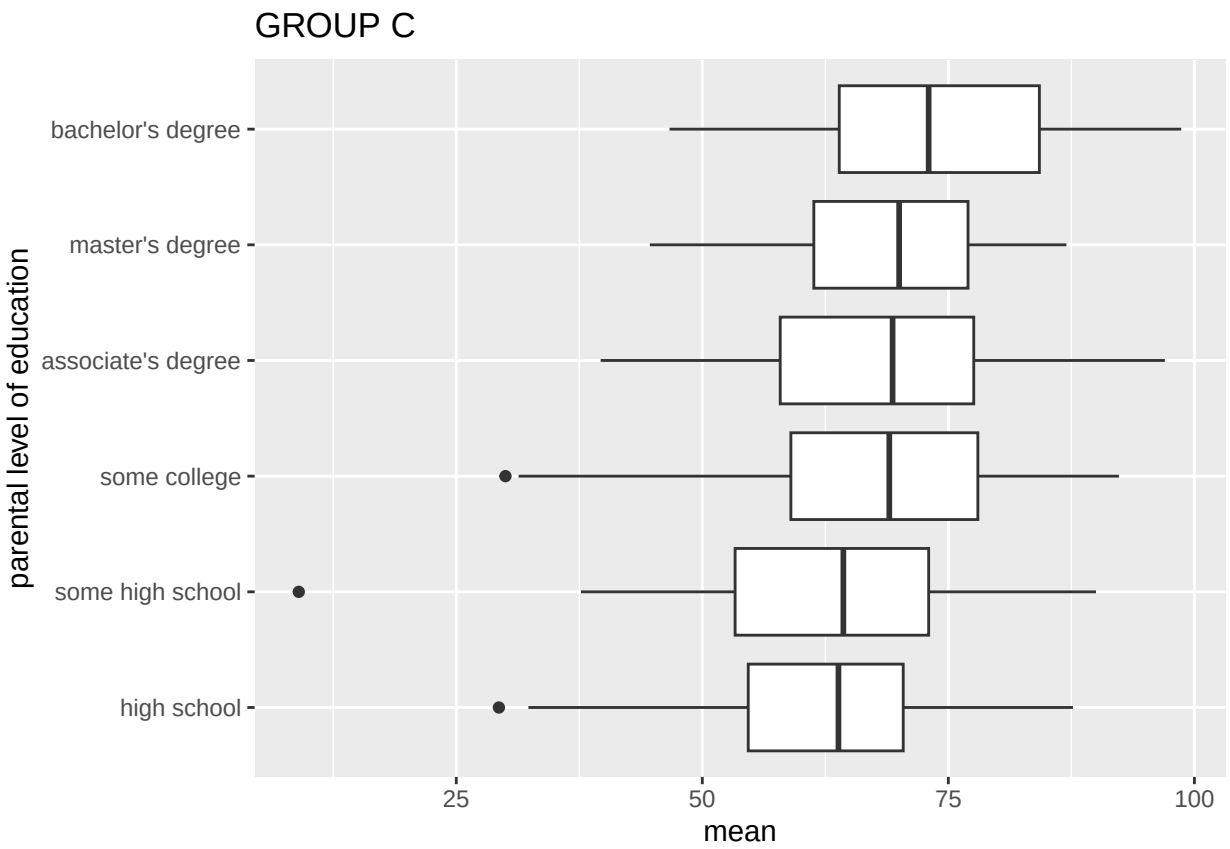
```
groupa<-ndata%>%filter(`race.ethnicity`=='group A')
groupb<-ndata%>%filter(`race.ethnicity`=='group B')
groupc<-ndata%>%filter(`race.ethnicity`=='group C')
groupd<-ndata%>%filter(`race.ethnicity`=='group D')
groupe<-ndata%>%filter(`race.ethnicity`=='group E')
ggplot(groupa)+geom_boxplot(mapping=aes(x=reorder(`parental.level.of.education`,mean,median),mean))+
  ggtitle('GROUP A')+xlab('parental level of education')+coord_flip()
```



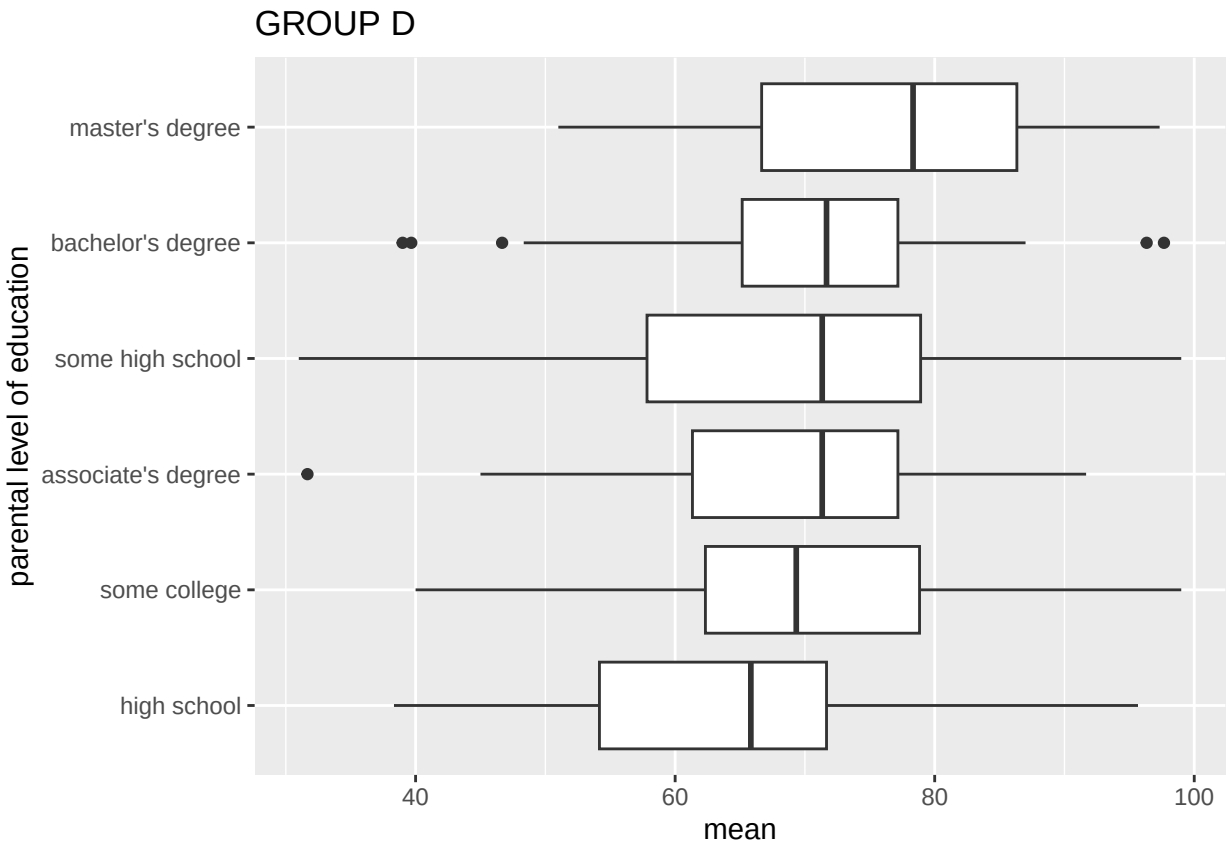

```
ggplot(groupb)+geom_boxplot(mapping=aes(x=reorder(`parental.level.of.education`,mean,median),mean))+
  ggtitle('GROUP B')+xlab('parental level of education')+coord_flip()
```



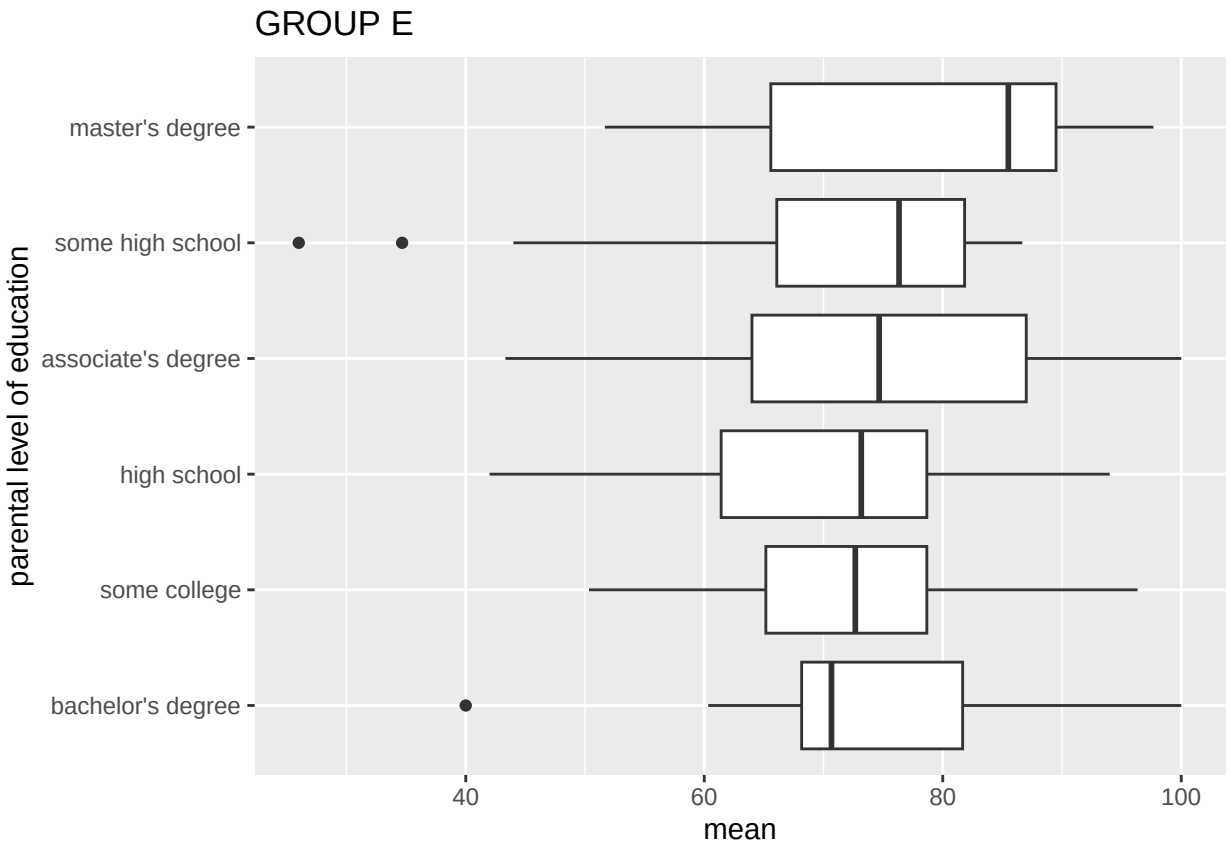

```
ggplot(groupc)+geom_boxplot(mapping=aes(x=reorder(`parental.level.of.education`,mean,median),mean))+
  ggtitle('GROUP C')+xlab('parental level of education')+coord_flip()
```




```
ggplot(groupd)+geom_boxplot(mapping=aes(x=reorder(`parental.level.of.education`,mean,median),mean))+
  ggtitle('GROUP D')+xlab('parental level of education')+coord_flip()
```



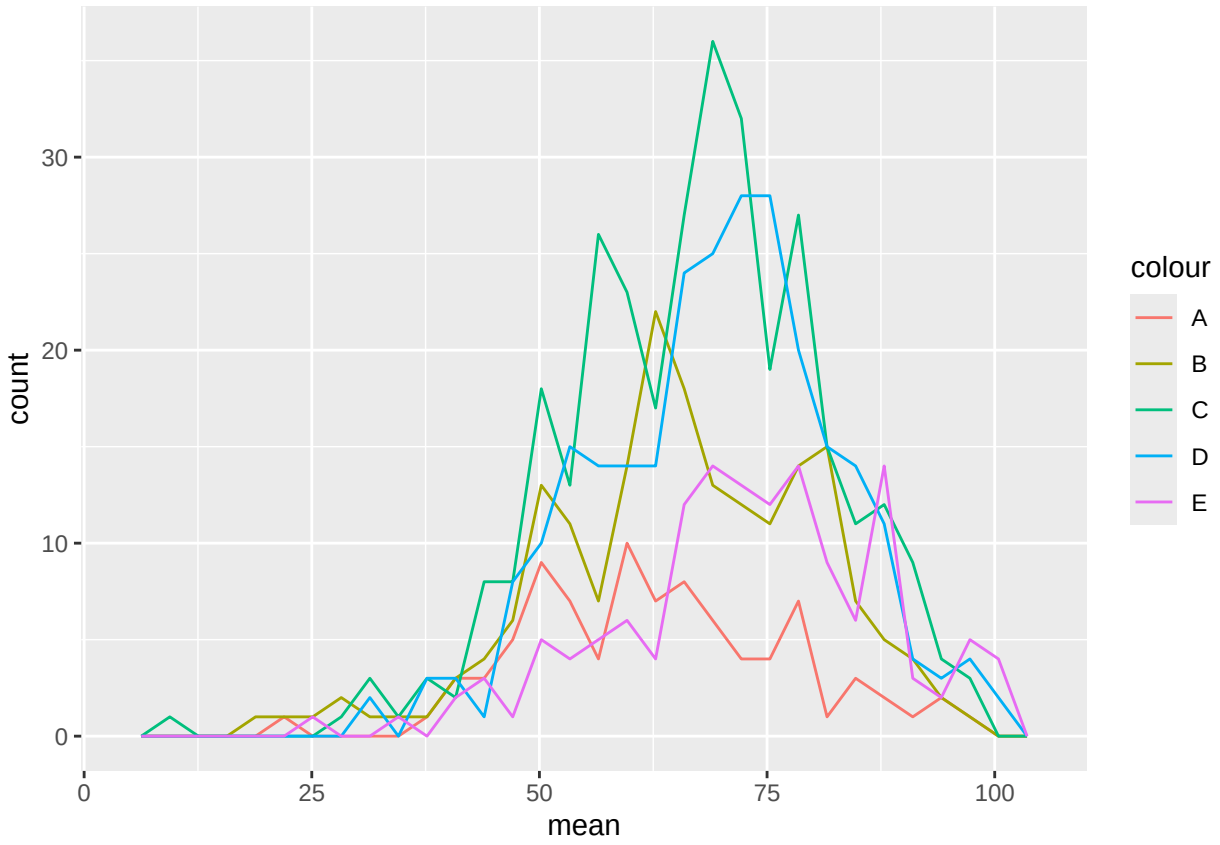

```
ggplot(groupe)+geom_boxplot(mapping=aes(x=reorder(`parental.level.of.education`,mean,median),mean))+
  ggtitle('GROUP E')+xlab('parental level of education')+coord_flip()
```



Mean Score of Each Plot

```
ggplot()+geom_freqpoly(groupa,mapping=aes(mean,color='A'))+geom_freqpoly(groupb,mapping=aes(mean,color=
  geom_freqpoly(groupc,mapping=aes(mean,color='C'))+geom_freqpoly(groupd,mapping=aes(mean,color='D'))+
  geom_freqpoly(groupe,mapping=aes(mean,color='E'))
```

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


count
30
20
10
0
0
25
50
75
100
mean
colour
A
B
C
D
E