Big Data Creation in R

ASSIGNMENT USING R: BIG DATA

The data created contains 10000 rows and 7 columns.

Data creation with a Vector

The important function used here is the c() which is used when combining more than one numbers together in R.

GENDER

```
### reproducibility of random numbers
set.seed(1001)

gender <- sample(x=c("Male","Female"), 10000, replace = T, prob = c(0.35,0.65) )
# The first six roles of the newly created gender variable.
head(gender)
```

```
## [1] "Male"   "Female" "Female" "Female" "Female" "Male"
```

```
# table showing the frequencies of the genders
prop.table(table(gender))
```

```
## gender
## Female   Male
## 0.6561 0.3439
```

```
# number of observations in the gender variable
length(gender)
```

```
## [1] 10000
```

RACE

```
race <- c(rep("Africa", 0.1*10000), rep("S_America", 0.2*10000), rep("Europe", 0.35*10000),
          rep("Asia", 0.25*10000), rep("Australia", 0.1*10000))

race <- sample(race,10000)
head(race)
```

```
## [1] "Asia"      "Europe"    "Europe"    "S_America" "Europe"    "Europe"
```

```
# number of obseravtions in the race variable.
length(race)
```

```
## [1] 10000
```

```
# table with the proportion of each unique levels in the race variable
prop.table(summary(as.factor(race)))
```

```
##    Africa      Asia Australia    Europe S_America 
##      0.10      0.25      0.10      0.35      0.20
```

```
# Alternative to creating the race variable.
race <- rep(c("Africa","S_America","Europe","Asia","Australia"), 10000*c(0.1,0.2,0.35,0.25,0.1))
head(race)
```

```
## [1] "Africa" "Africa" "Africa" "Africa" "Africa" "Africa"
```

```
length(race)
```

```
## [1] 10000
```

```
prop.table(summary(as.factor(race)))
```

```
##    Africa      Asia Australia    Europe S_America 
##      0.10      0.25      0.10      0.35      0.20
```

LUNCH

```
n <- 10000
food <- character(n)
u <- runif(n)
food[u<=0.1] <- "Free"
food[u>0.1 & u<=0.3] <- "Reduced"
food[u>0.3 & u<=0.7] <- "normal"
food[u>0.7] <- "Standard"
table(food)
```

```
## food
##     Free   normal  Reduced Standard 
##     1026     3997     2041     2936
```

```
prop.table(summary(as.factor(food)))
```

```
##     Free   normal  Reduced Standard 
##   0.1026   0.3997   0.2041   0.2936
```

```
Lunch <- food
```

MATHEMATICS SCORE

```
mathScore <- rnorm(10000, mean = 55, sd = 10)
head(mathScore)
```

```
## [1] 44.89654 51.19266 64.72695 56.90852 50.12859 69.37889
```

```
# rounding up to the nearest whole number
mathScore <- round(mathScore,0)
summary(mathScore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   16.00   49.00   55.00   55.18   62.00   91.00
```

CHEMISTRY SCORE

```
chemScore <- rnorm(10000, mean = 60, sd = 5)
chemScore <- round(chemScore,0)
summary(chemScore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   39.00   57.00   60.00   60.01   63.00   79.00
```

BIOLOGY SCORE

```
BioScore <- rnorm(10000, mean = 70, sd = 5)
BioScore <- round(BioScore,0)
summary(BioScore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   51.00   67.00   70.00   69.98   73.00   88.00
```

CLUB MEMBERSHIP

```
n <- 10000
member <- character(n)
u <- runif(n)
member[u<=0.6] <- "Yes"
member[u>0.6] <- "No"
table(member)
```

```
## member
##   No  Yes
## 4021 5979
```

```
prop.table(summary(as.factor(member)))
```

```
##     No    Yes
## 0.4021 0.5979
```

```
membership <- member
```

MATRIX

```
classPerformance <- cbind(gender, race, Lunch, mathScore, chemScore, BioScore, membership)
head(classPerformance)
```

```
##      gender   race     Lunch      mathScore chemScore BioScore membership
## [1,] "Male"   "Africa" "Reduced"  "45"      "59"      "69"     "Yes"
## [2,] "Female" "Africa" "normal"   "51"      "71"      "86"     "Yes"
## [3,] "Female" "Africa" "Reduced"  "65"      "64"      "73"     "No"
## [4,] "Female" "Africa" "normal"   "57"      "58"      "64"     "No"
## [5,] "Female" "Africa" "Standard" "50"      "55"      "74"     "No"
## [6,] "Male"   "Africa" "normal"   "69"      "60"      "75"     "No"
```

```
# The cbind function is also called the column binding. It binds each vectors by column
# into a resulting matrix object.

class(classPerformance)
```

```
## [1] "matrix" "array"
```

```
# summary statistics
summary(classPerformance)
```

```
##     gender              race               Lunch            mathScore        
##  Length:10000       Length:10000       Length:10000       Length:10000      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##   chemScore           BioScore          membership       
##  Length:10000       Length:10000       Length:10000      
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
```

DATA FRAME

```
classPerformance <- as.data.frame(classPerformance)

# ALternative
classPerformance <- data.frame(gender, race, Lunch, mathScore, chemScore, BioScore, membership)
class(classPerformance)
```

```
## [1] "data.frame"
```

```
head(classPerformance)
```

	gender	race	Lunch	mathScore	chemScore	BioScore	membership
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<chr>
1	Male	Africa	Reduced	45	59	69	Yes
2	Female	Africa	normal	51	71	86	Yes
3	Female	Africa	Reduced	65	64	73	No
4	Female	Africa	normal	57	58	64	No
5	Female	Africa	Standard	50	55	74	No
6	Male	Africa	normal	69	60	75	No

6 rows

```
summary(classPerformance)
```

```
##     gender              race               Lunch              mathScore       chemScore    
##  Length:10000       Length:10000       Length:10000       Min.   :16.00   Min.   :39.00  
##  Class :character   Class :character   Class :character   1st Qu.:49.00   1st Qu.:57.00  
##  Mode  :character   Mode  :character   Mode  :character   Median :55.00   Median :60.00  
##                                                           Mean   :55.18   Mean   :60.01  
##                                                           3rd Qu.:62.00   3rd Qu.:63.00  
##                                                           Max.   :91.00   Max.   :79.00  
##     BioScore      membership       
##  Min.   :51.00   Length:10000      
##  1st Qu.:67.00   Class :character  
##  Median :70.00   Mode  :character  
##  Mean   :69.98                     
##  3rd Qu.:73.00                     
##  Max.   :88.00                     
```

Show Data Table

```
library(DT)

datatable(classPerformance, extensions = "Buttons", options = list(
  dom ="Bfrtip",
  buttons = c("copy", "csv", "excel", "pdf", "print")
))
```

Copy CSV Excel PDF Print

Search:

	gender	race	Lunch	mathScore	chemScore	BioScore	membership
1	Male	Africa	Reduced	45	59	69	Yes
6	Male	Africa	normal	69	60	75	No
10	Male	Africa	Standard	56	59	67	Yes
13	Male	Africa	normal	60	70	68	No
16	Male	Africa	normal	53	69	68	Yes
20	Male	Africa	Standard	41	60	67	No
23	Male	Africa	Reduced	57	65	72	No
24	Male	Africa	Standard	63	71	68	Yes
26	Male	Africa	Standard	48	61	72	Yes
27	Male	Africa	Standard	60	55	69	Yes

Showing 1 to 10 of 10,000 entries

Previous 1 2 3 4 5 … 1000 Next

DATA MANIPULATION WITH THE DPLYR PACKAGE

install.packages("dplyr") for Installing the dplyr package from CRAN

library(dplyr)

SELECT AND SUBSET

```
class(classPerformance)
```

```
## [1] "data.frame"
```

```
str(classPerformance)
```

```
## 'data.frame':    10000 obs. of  7 variables:
##  $ gender    : chr  "Male" "Female" "Female" "Female" ...
##  $ race      : chr  "Africa" "Africa" "Africa" "Africa" ...
##  $ Lunch     : chr  "Reduced" "normal" "Reduced" "normal" ...
##  $ mathScore : num  45 51 65 57 50 69 47 49 63 56 ...
##  $ chemScore : num  59 71 64 58 55 60 59 59 56 59 ...
##  $ BioScore  : num  69 86 73 64 74 75 70 65 70 67 ...
##  $ membership: chr  "Yes" "Yes" "No" "No" ...
```

FACTOR: CATEGORICAL VARIABLE

```
classPerformance$gender <- as.factor(classPerformance$gender)
classPerformance$race <- as.factor(classPerformance$race)
classPerformance$Lunch <- as.factor(classPerformance$Lunch)
classPerformance$membership <- as.factor(classPerformance$membership)
levels(classPerformance$Lunch)
```

```
## [1] "Free"     "normal"   "Reduced"  "Standard"
```

```
select(classPerformance, mathScore, chemScore, BioScore)
```

mathScore	chemScore	BioScore
<dbl>	<dbl>	<dbl>
45	59	69
51	71	86

	mathScore	chemScore	BioScore
	<dbl>	<dbl>	<dbl>
	65	64	73
	57	58	64
	50	55	74
	69	60	75
	47	59	70
	49	59	65
	63	56	70
	56	59	67

1-10 of 1,000 rows Previous **1** 2 3 4 5 6 ... 100 Next

```
scores <- select(classPerformance, mathScore, chemScore, BioScore)
head(scores)
```

	mathScore	chemScore	BioScore
	<dbl>	<dbl>	<dbl>
1	45	59	69
2	51	71	86
3	65	64	73
4	57	58	64
5	50	55	74
6	69	60	75

6 rows

Students that scored above 50 in all the courses

```
above50 <- subset(classPerformance, mathScore>50 & chemScore>50 & BioScore>50)
select(above50, mathScore, chemScore, BioScore)
```

	mathScore	chemScore	BioScore
	<dbl>	<dbl>	<dbl>
2	51	71	86
3	65	64	73

	mathScore <dbl>	chemScore <dbl>	BioScore <dbl>
4	57	58	64
6	69	60	75
9	63	56	70
10	56	59	67
12	71	64	70
13	60	70	68
14	56	55	63
16	53	69	68

1-10 of 1,000 rows Previous **1** 2 3 4 5 6 ... 100 Next

```
scores_above50 <- select(above50, mathScore, chemScore, BioScore)
min(scores_above50$mathScore)
```

```
## [1] 51
```

```
min(scores_above50$chemScore)
```

```
## [1] 51
```

```
min(scores_above50$BioScore)
```

```
## [1] 52
```

```
max(scores_above50$mathScore)
```

```
## [1] 91
```

```
max(scores_above50$chemScore)
```

```
## [1] 79
```

```
max(scores_above50$BioScore)
```

```
## [1] 88
```

No student have 50 all through in all the courses

```
equal50 <- subset(classPerformance, mathScore==50 & chemScore==50 & BioScore==50)
```

ARRANGE

```
arranged_cp <- arrange(classPerformance, mathScore, chemScore, BioScore)
head(arranged_cp)
```

	gender <fct>	**race** <fct>	**Lunch** <fct>	**mathScore** <dbl>	**chemScore** <dbl>	**BioScore** <dbl>	**membership** <fct>
1	Female	Africa	Standard	16	61	69	No
2	Male	Asia	Standard	17	59	68	Yes
3	Female	Europe	Reduced	19	61	64	No
4	Male	S_America	Free	19	63	71	Yes
5	Female	Asia	Standard	20	53	70	No
6	Female	Asia	normal	20	60	68	Yes

6 rows

FILTER

Filter out only Africans who score above 80 in maths score and are also a member of

an association and give them a scholarship

```
dplyr::filter(classPerformance, mathScore>80, race=="Africa", membership=="Yes")
```

gender <fct>	**race** <fct>	**Lunch** <fct>	**mathScore** <dbl>	**chemScore** <dbl>	**BioScore** <dbl>	**membership** <fct>
Male	Africa	Reduced	81	62	75	Yes
Female	Africa	Reduced	83	61	68	Yes
Male	Africa	Standard	83	68	76	Yes

3 rows

```
head(dplyr::filter(classPerformance, mathScore>80, race=="Africa", membership=="Yes"))
```

	gender <fct>	race <fct>	Lunch <fct>	mathScore <dbl>	chemScore <dbl>	BioScore <dbl>	membership <fct>
1	Male	Africa	Reduced	81	62	75	Yes
2	Female	Africa	Reduced	83	61	68	Yes
3	Male	Africa	Standard	83	68	76	Yes

3 rows

Filter out Female Africans who are on free lunch. How many are they?

```
filter(classPerformance, race=="Africa", gender=="Female", Lunch=="Free")
```

gender <fct>	race <fct>	Lunch <fct>	mathScore <dbl>	chemScore <dbl>	BioScore <dbl>	membership <fct>
Female	Africa	Free	44	51	74	No
Female	Africa	Free	71	64	70	Yes
Female	Africa	Free	42	56	68	Yes
Female	Africa	Free	60	63	70	Yes
Female	Africa	Free	55	68	76	Yes
Female	Africa	Free	44	70	74	Yes
Female	Africa	Free	37	63	86	Yes
Female	Africa	Free	56	60	73	Yes
Female	Africa	Free	49	56	68	Yes
Female	Africa	Free	50	71	70	Yes

1-10 of 62 rows Previous **1** 2 3 4 5 6 7 Next

```
a <- filter(classPerformance, race=="Africa", gender=="Female", Lunch=="Free")
head(filter(classPerformance, race=="Africa", gender=="Female", Lunch=="Free"))
```

	gender <fct>	race <fct>	Lunch <fct>	mathScore <dbl>	chemScore <dbl>	BioScore <dbl>	membership <fct>
1	Female	Africa	Free	44	51	74	No
2	Female	Africa	Free	71	64	70	Yes
3	Female	Africa	Free	42	56	68	Yes
4	Female	Africa	Free	60	63	70	Yes

	gender <fct>	race <fct>	Lunch <fct>	mathScore <dbl>	chemScore <dbl>	BioScore <dbl>	membership <fct>
5	Female	Africa	Free	55	68	76	Yes
6	Female	Africa	Free	44	70	74	Yes

6 rows

```
nrow(filter(classPerformance, race=="Africa", gender=="Female", Lunch=="Free"))
```

```
## [1] 62
```

```
# They are 54 in number
```

PIPE OPERATOR %>%

create a variable with information on the scores of the male Europeans with a standard

lunch who are not a member of any organization. Store the first 6 rows

```
df <- classPerformance %>%
  filter(race=="Europe", gender=="Male", Lunch=="Standard", membership=="No") %>%
  arrange(desc(mathScore)) %>%
  select(mathScore, chemScore, BioScore) %>%
  head()
```

MUTATE

Creating a new column using the existing columns.

Find the average of all the scores

```
mutate(classPerformance, AvgScores = (mathScore+chemScore+BioScore)/3)
```

gender <fct>	race <fct>	Lunch <fct>	mathScore <dbl>	chemScore <dbl>	BioScore <dbl>	membership <fct>	AvgScores <dbl>
Male	Africa	Reduced	45	59	69	Yes	57.66667
Female	Africa	normal	51	71	86	Yes	69.33333
Female	Africa	Reduced	65	64	73	No	67.33333
Female	Africa	normal	57	58	64	No	59.66667
Female	Africa	Standard	50	55	74	No	59.66667
Male	Africa	normal	69	60	75	No	68.00000
Female	Africa	Standard	47	59	70	No	58.66667
Female	Africa	normal	49	59	65	No	57.66667
Female	Africa	Reduced	63	56	70	Yes	63.00000
Male	Africa	Standard	56	59	67	Yes	60.66667

1-10 of 1,000 rows Previous **1** 2 3 4 5 6 ... 100 Next

```
head(mutate(classPerformance, AvgScores = (mathScore+chemScore+BioScore)/3))
```

	gender <fct>	race <fct>	Lunch <fct>	mathScore <dbl>	chemScore <dbl>	BioScore <dbl>	membership <fct>	AvgScores <dbl>
1	Male	Africa	Reduced	45	59	69	Yes	57.66667
2	Female	Africa	normal	51	71	86	Yes	69.33333
3	Female	Africa	Reduced	65	64	73	No	67.33333
4	Female	Africa	normal	57	58	64	No	59.66667
5	Female	Africa	Standard	50	55	74	No	59.66667
6	Male	Africa	normal	69	60	75	No	68.00000

6 rows

Rank

```
classPerformance %>%
  group_by(gender,race) %>%
  summarise(total_cnt = n(), totalsc = sum(mathScore,chemScore,BioScore)) %>%
  arrange(gender, race, desc(total_cnt), desc(totalsc)) %>%
  mutate(rank = dense_rank(desc(total_cnt))) %>%
  arrange(rank) %>%
  head()
```

```
## `summarise()` regrouping output by 'gender' (override with `.groups` argument)
```

gender <fct>	race <fct>	total_cnt <int>	totalsc <dbl>	rank <int>
Female	Europe	2299	425315	1
Male	Europe	1201	222401	1
Female	Asia	1623	300094	2
Male	Asia	877	162412	2
Female	S_America	1311	242235	3
Male	S_America	689	128389	3

6 rows

DATA VISUALIZATION

install.packages("ggplot2") for installing the ggplot2 package

library(ggplot2)

install.packages("ggthemes") for installing the ggthemes package

library(ggthemes)

PIE OR BARCHART

PIE CHART

Chart 1

```
classPerformance %>%
  ggplot(aes(x= "", fill = factor(race))) +
  geom_bar(stat= "count", width = 1, color = "white") +
  coord_polar("y", start = 0, direction = -1) +
  scale_fill_manual(values = c(rgb(1,0,.5),rgb(.5,.5,1),rgb(.7,.2,.1),rgb(0,.2,.9),rgb(.7,.5,0
))) +
  theme_void()
```

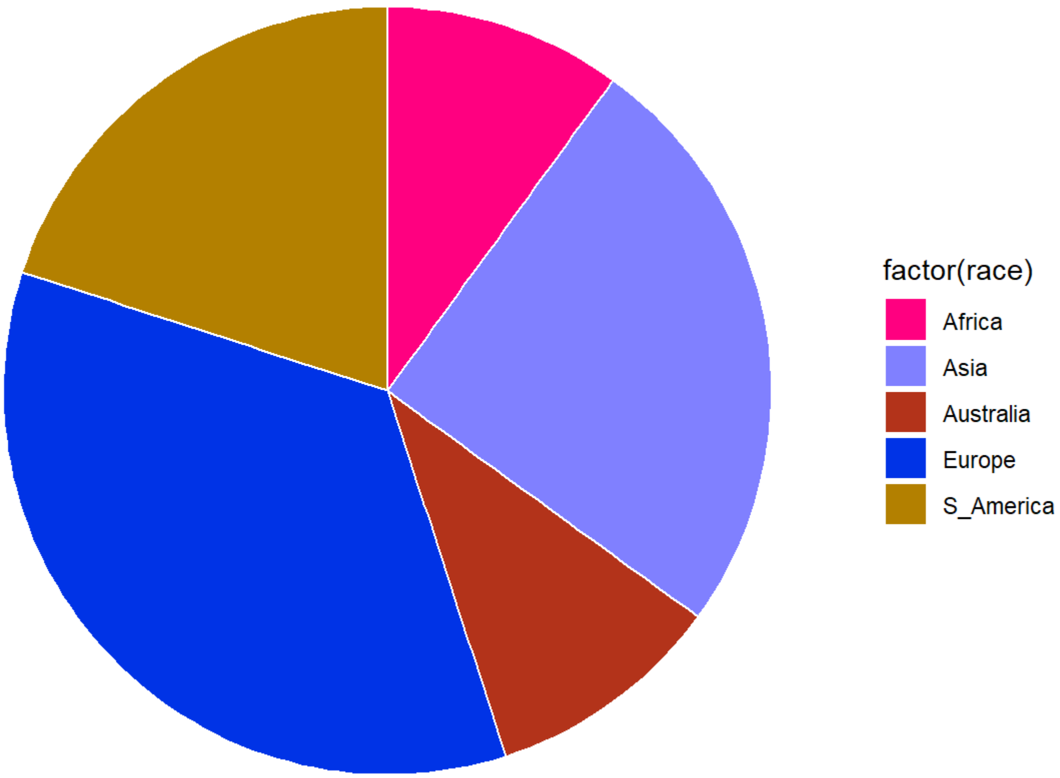
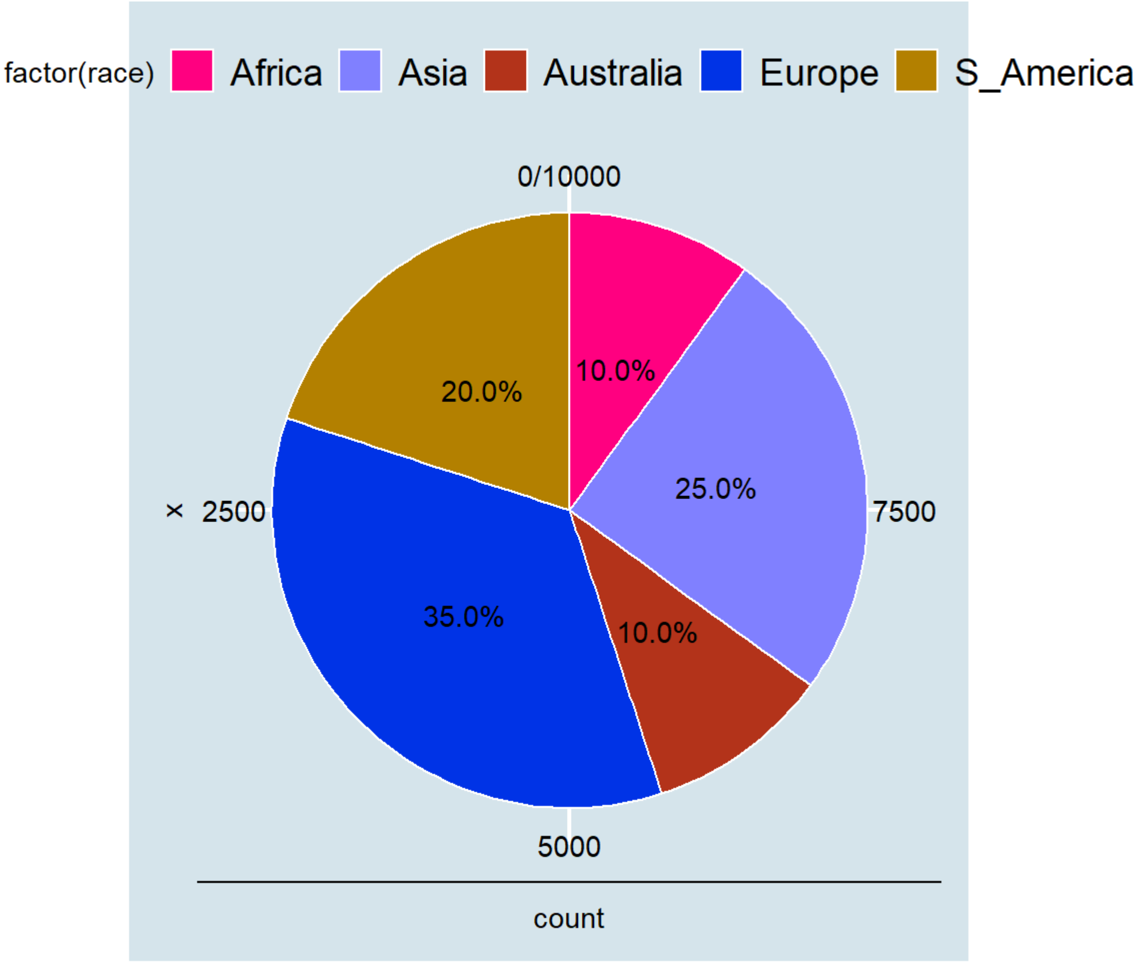


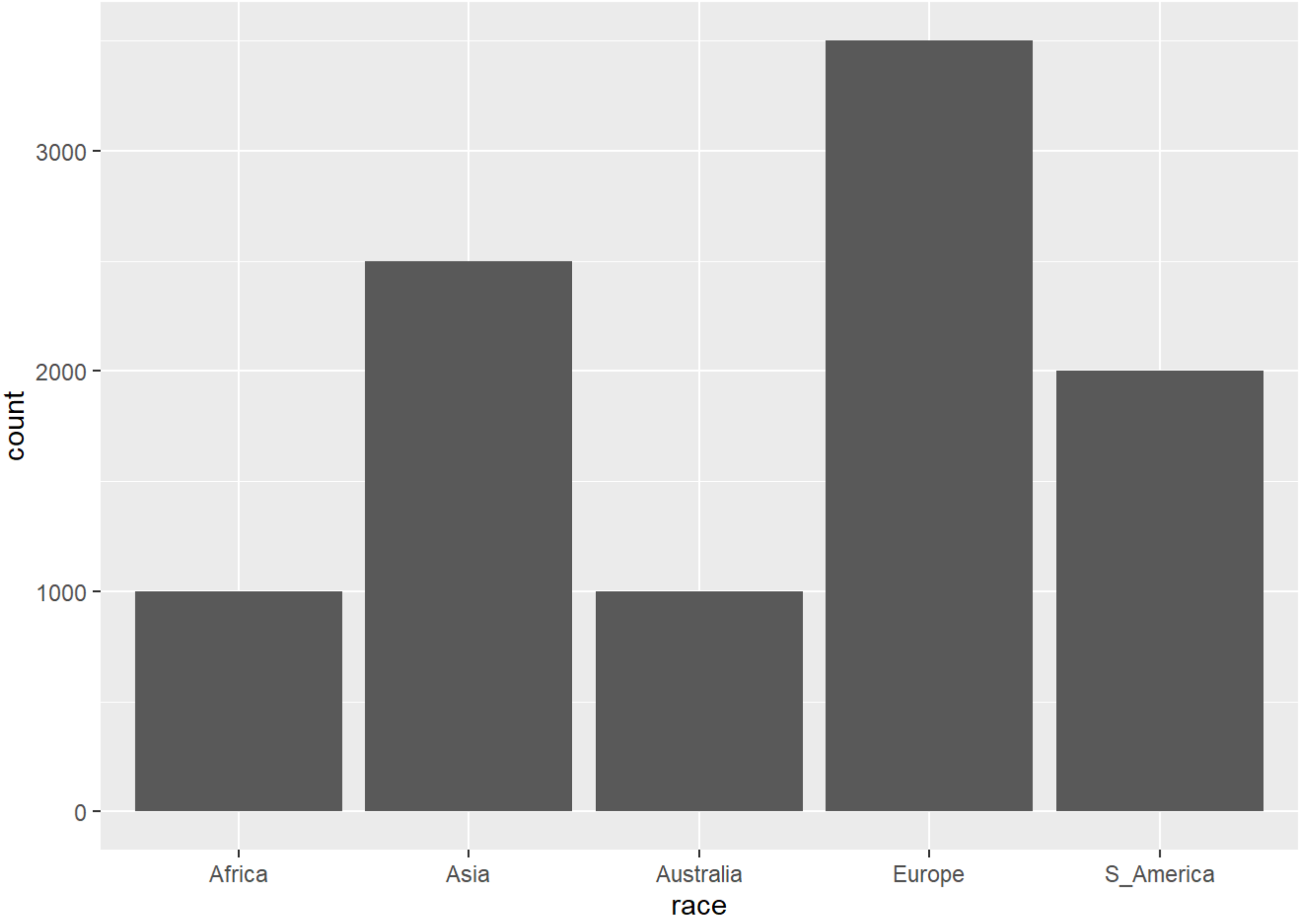
Chart 2

```
ggplot(classPerformance, aes(x = "", fill = factor(race))) +
  geom_bar(stat= "count", width = 1, color = "white") +
  geom_text(aes(label = scales::percent(..count.. / sum(..count..))),
            stat = "count", position = position_stack(vjust = .5)) +
  coord_polar("y", start = 0, direction = -1) +
  scale_fill_manual(values = c(rgb(1,0,.5),rgb(.5,.5,1),rgb(.7,.2,.1),rgb(0,.2,.9),rgb(.7,.5,0
))) +
  theme_economist()
```

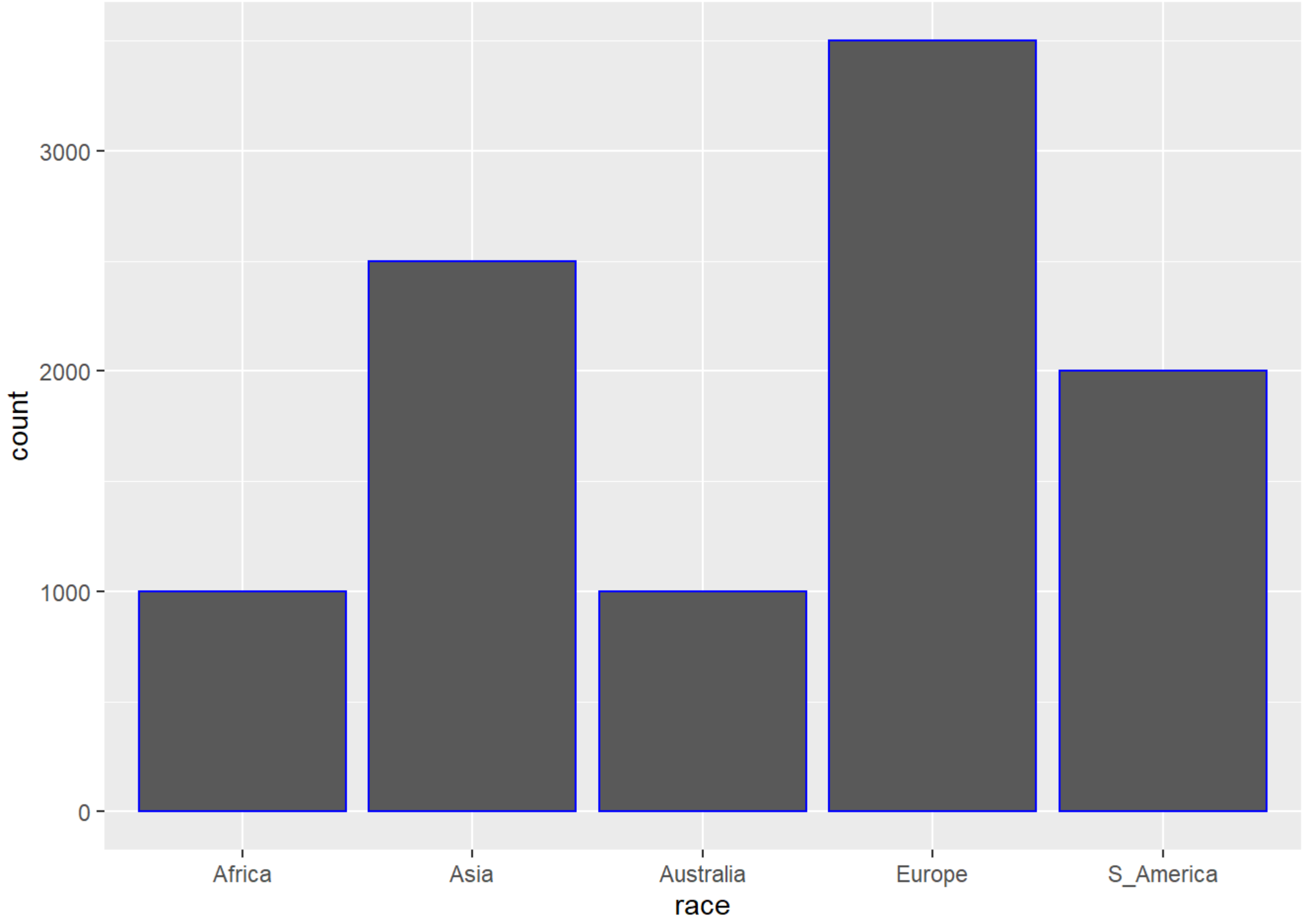


BAR CHART

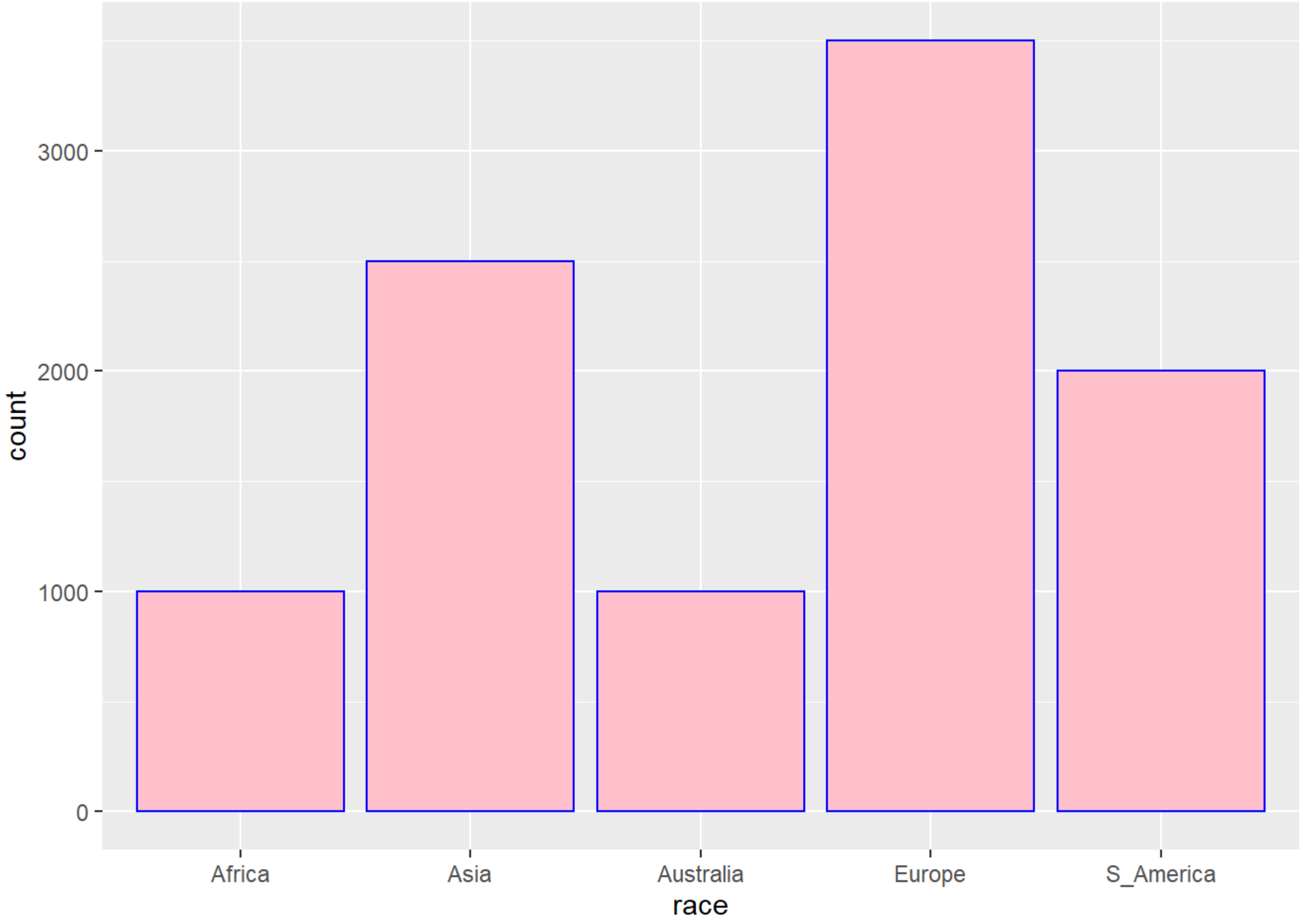
```
pl <- ggplot(classPerformance, aes(x=race))
print(pl + geom_bar())
```



```
print(pl + geom_bar(color="blue")) ## for outline colour
```

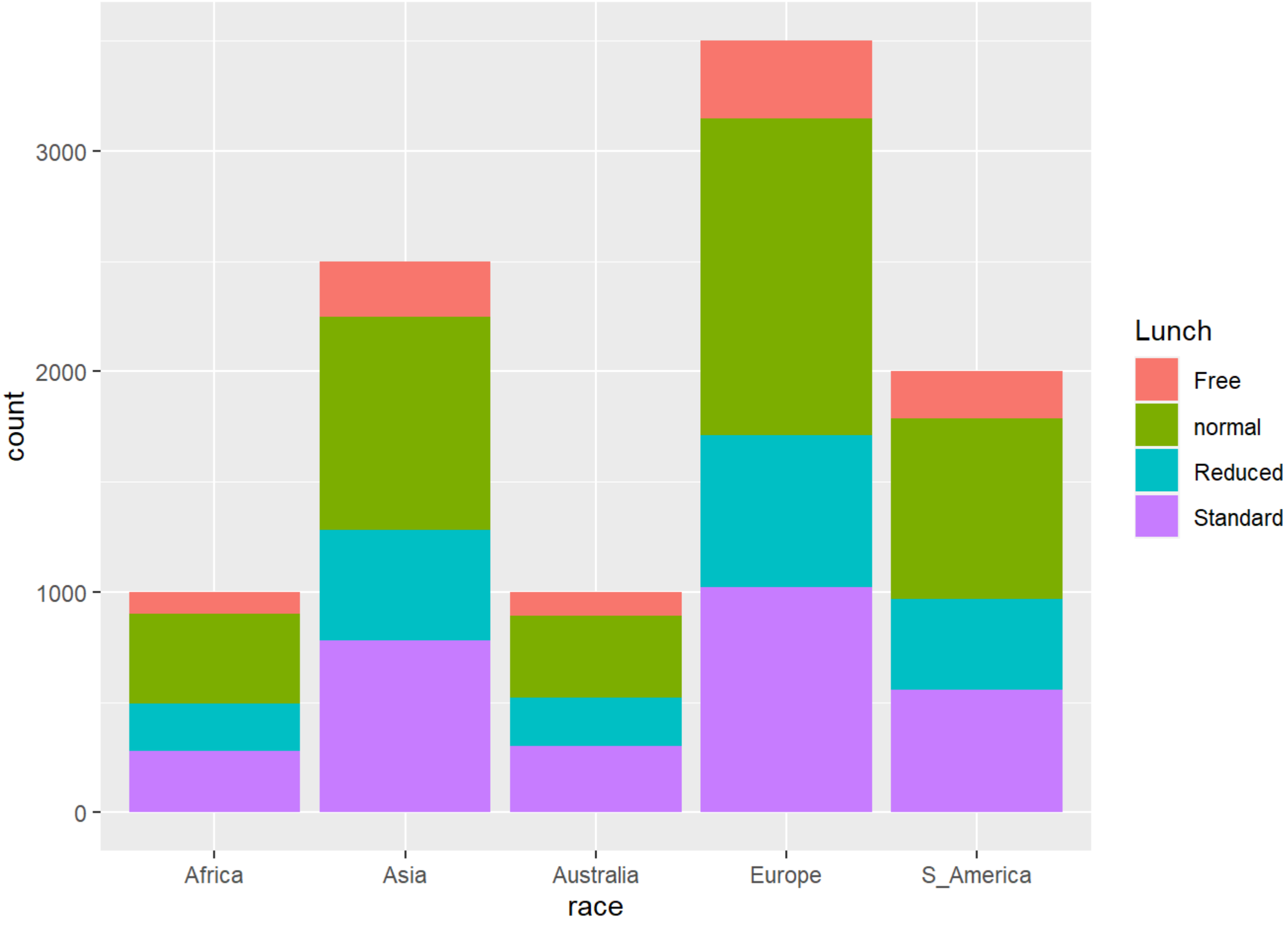


```
print(pl + geom_bar(color="blue", fill="pink"))
```



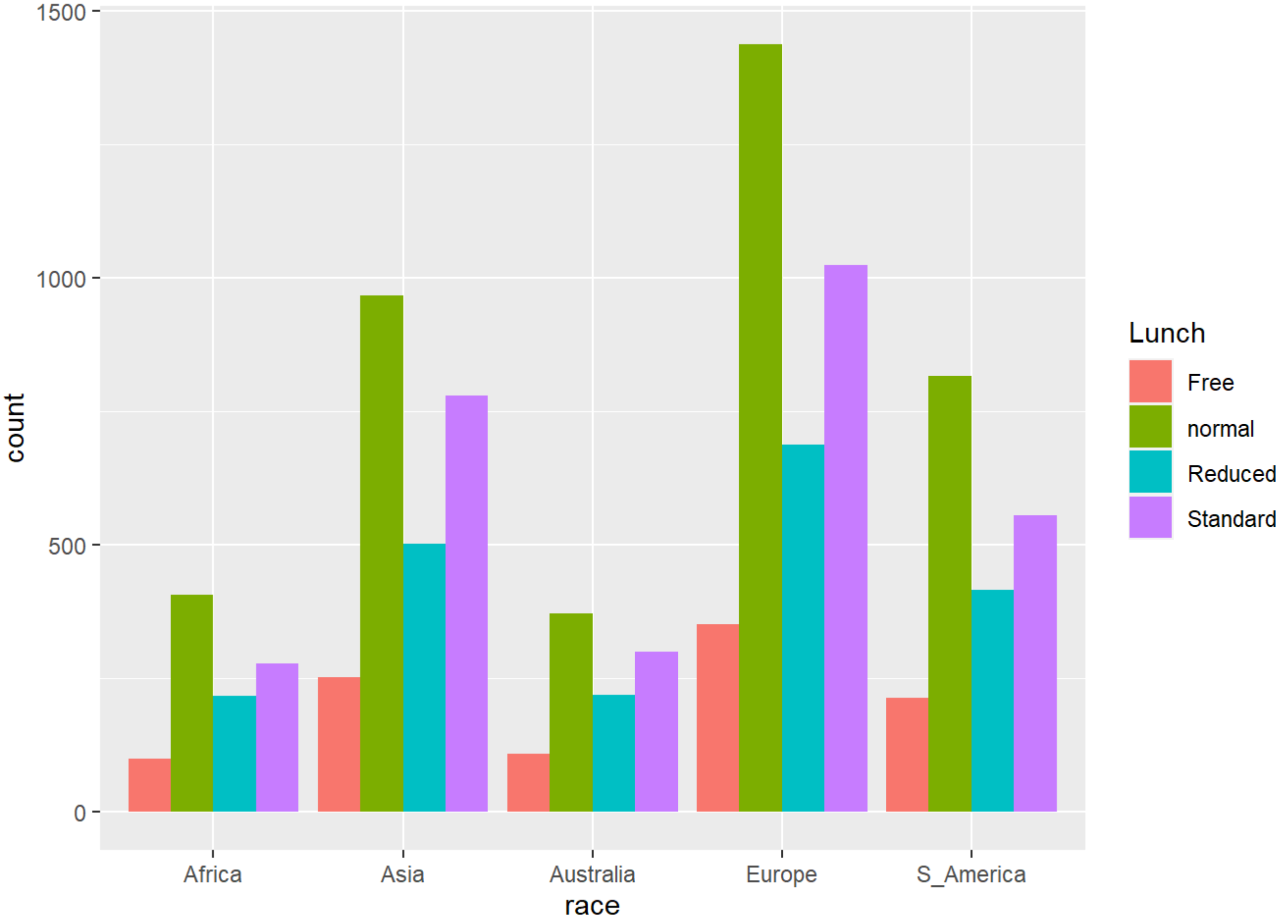
To automatically create a stacked bar.

```
print(pl + geom_bar(aes(fill=Lunch)))
```



For comparing

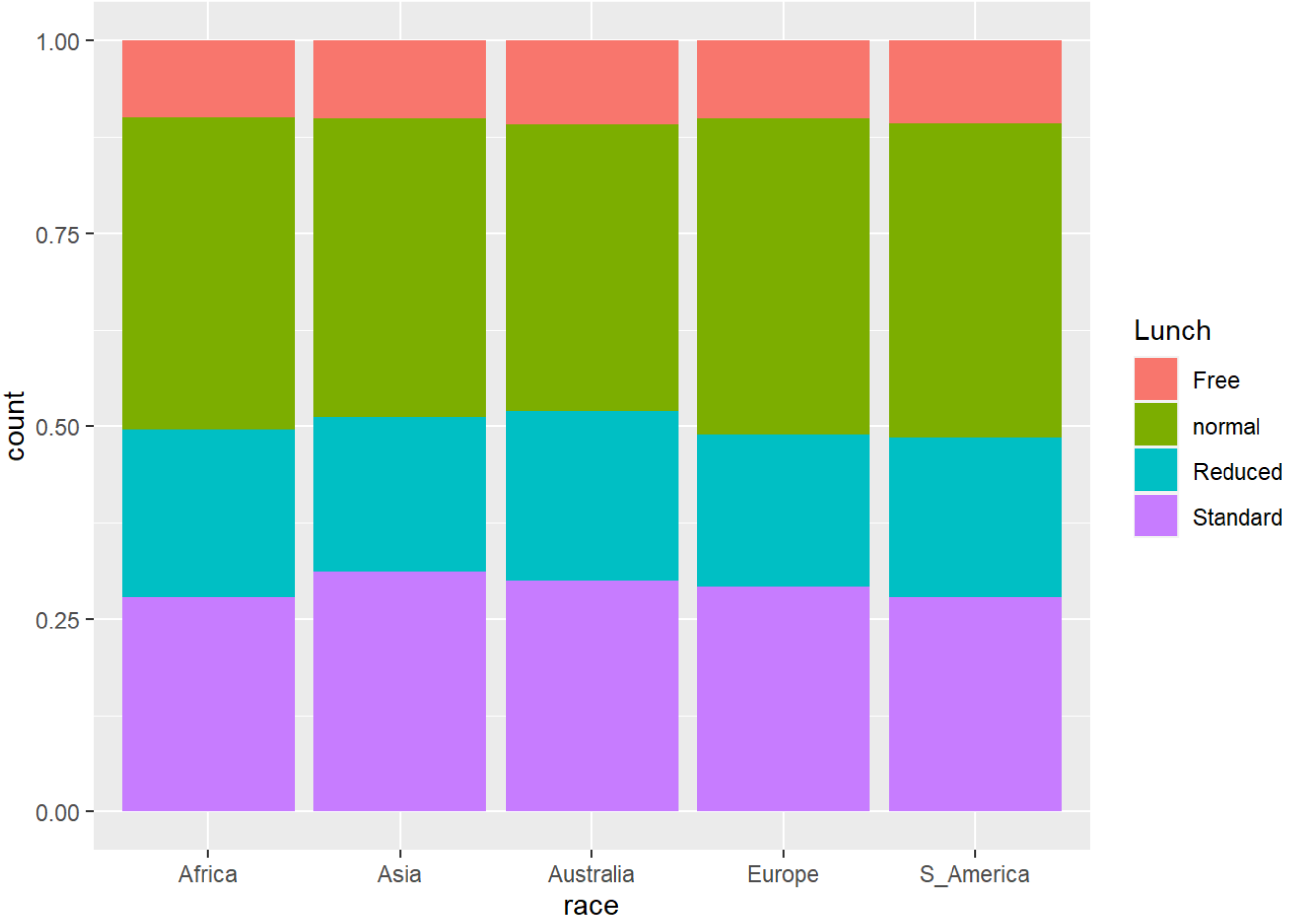
```
print(pl + geom_bar(aes(fill=Lunch ), position = "dodge"))
```



#

This shows the percentage instead of count

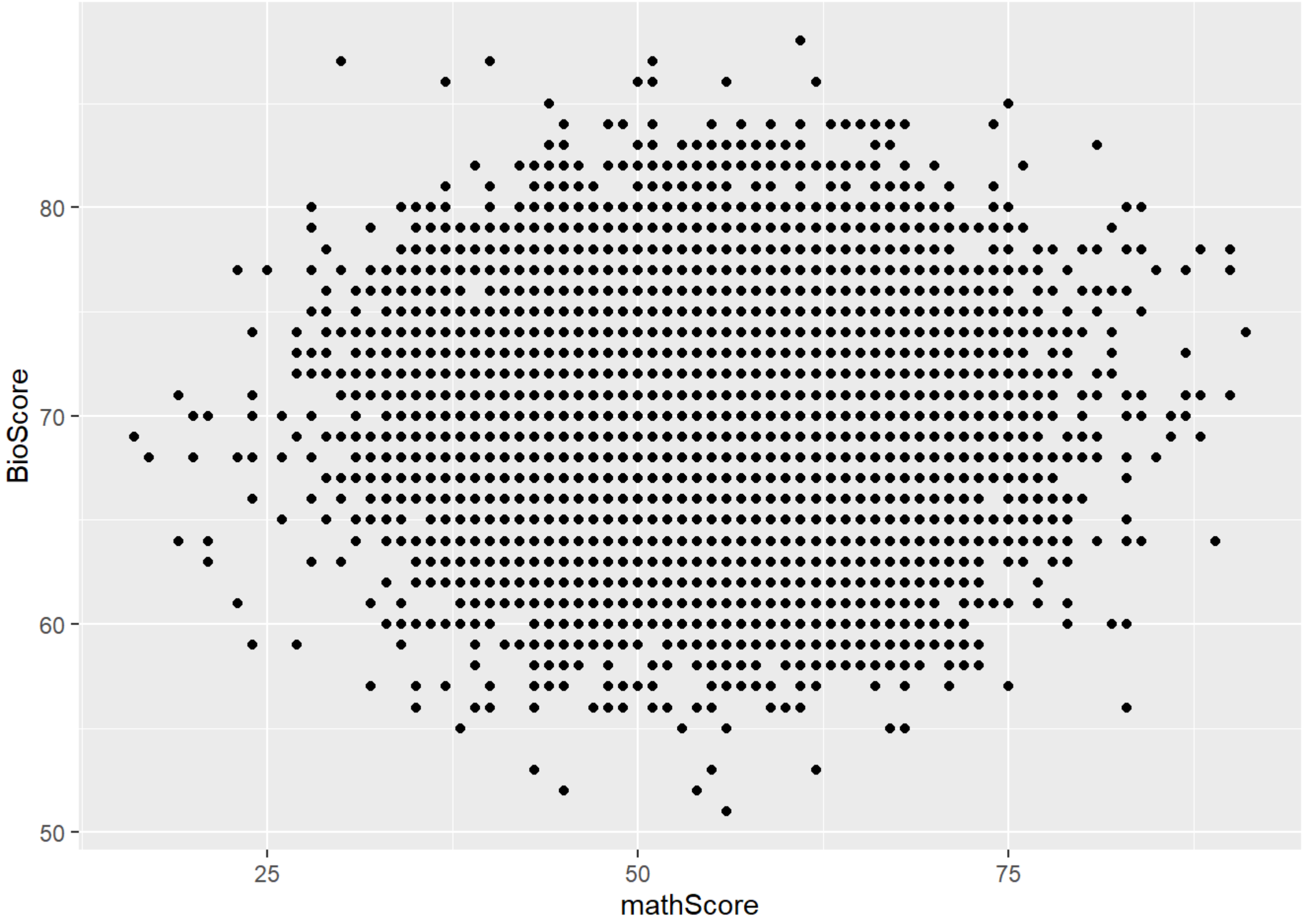
```
print(pl + geom_bar(aes(fill=Lunch ), position = "fill"))
```



PLOTS OR HISTOGRAMS

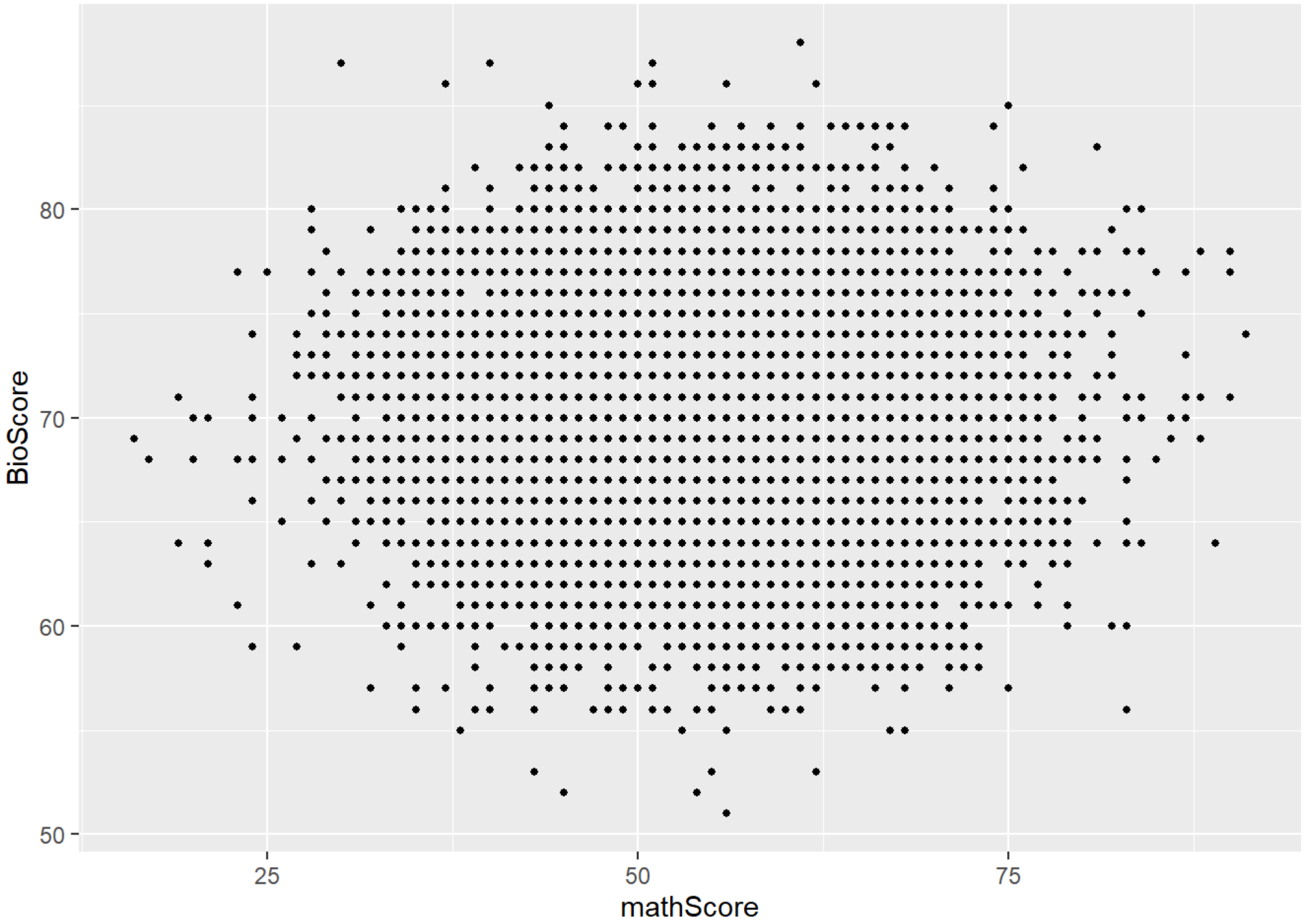
PLOTS

```
pl <- ggplot(classPerformance, aes(x=mathScore, y=BioScore))
pl + geom_point()
```



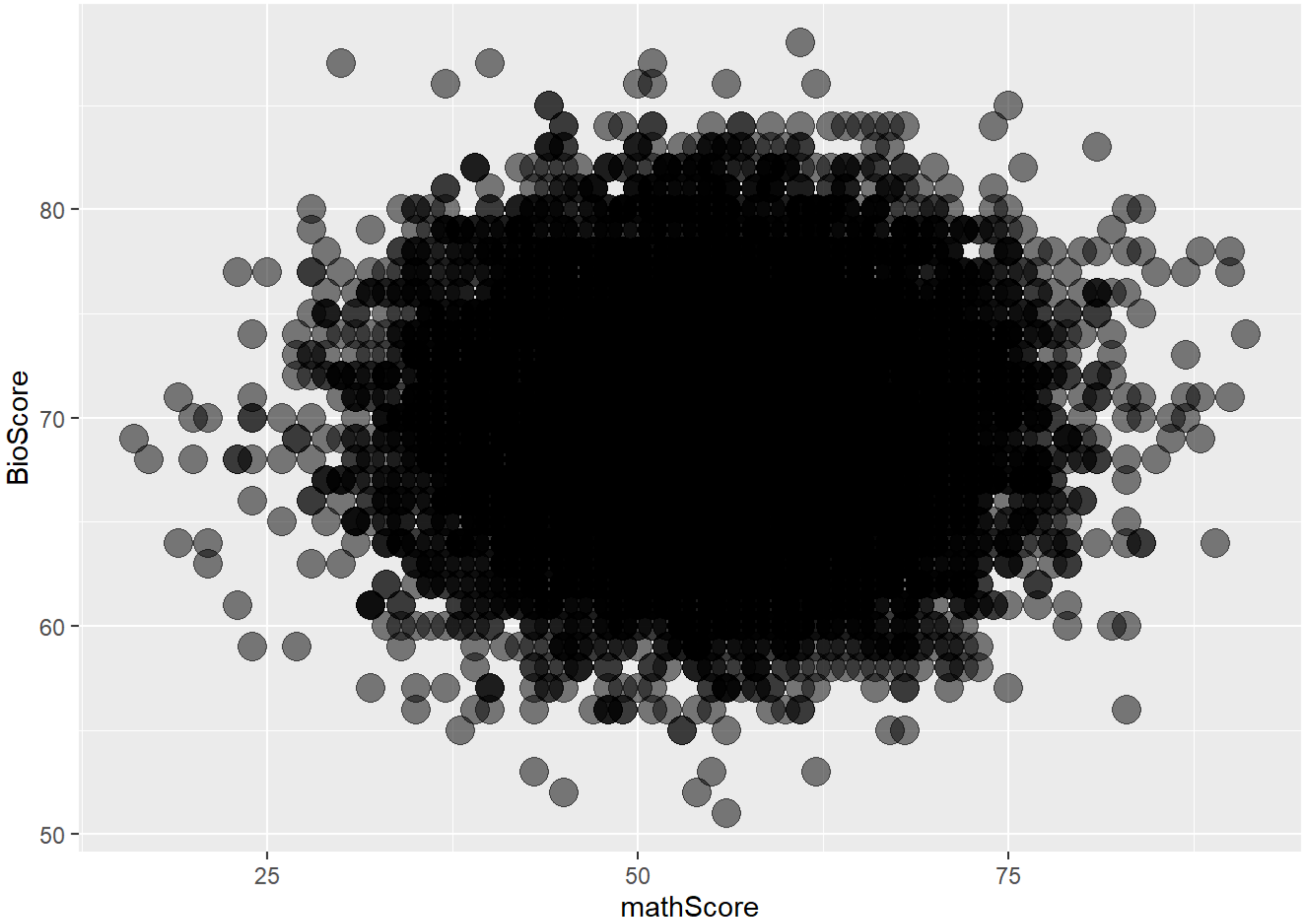
changing the size of data points

```
pl + geom_point(size=1)
```



overlapping points gets darker

```
pl + geom_point(alpha=0.5,size=5)
```



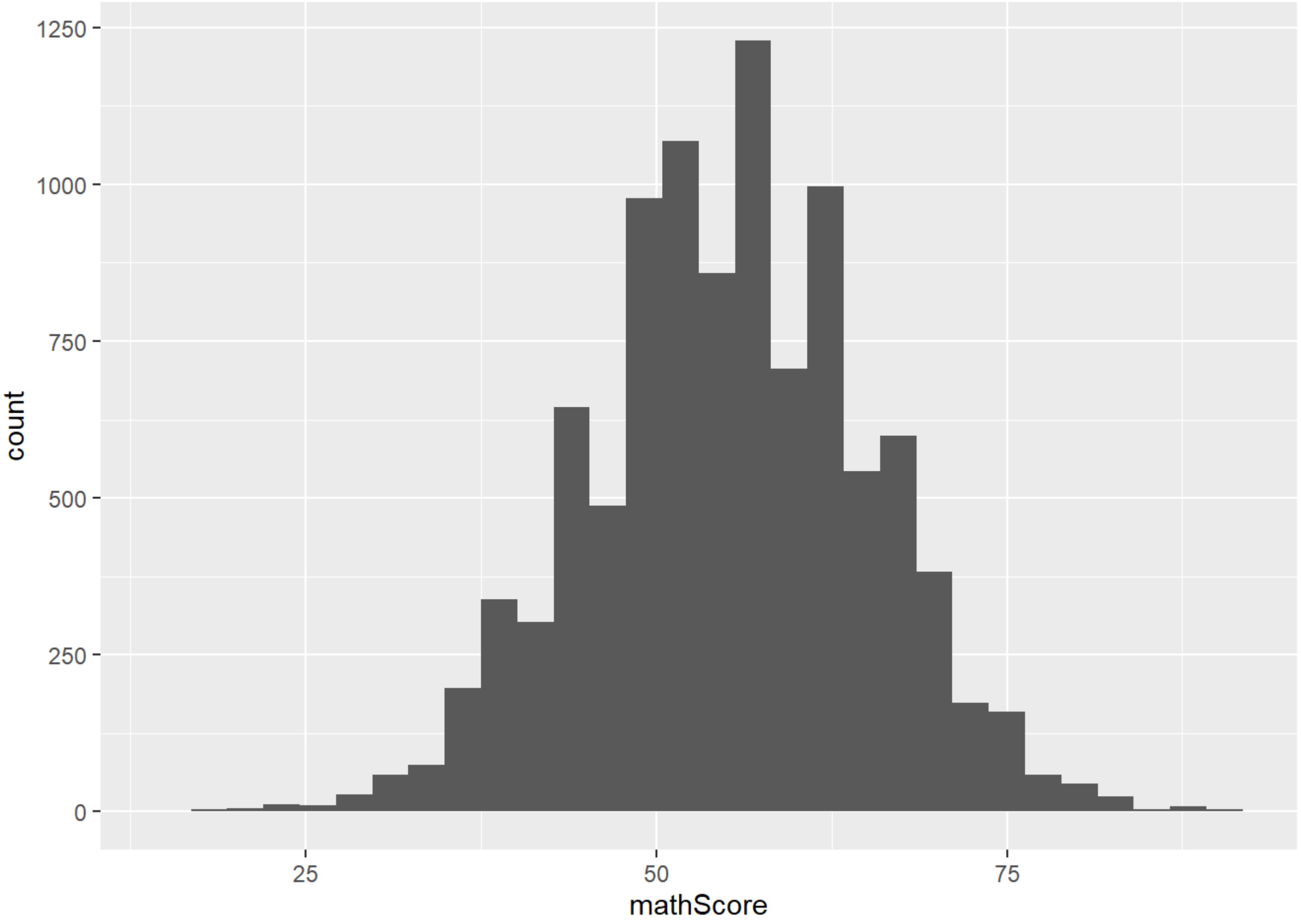
HISTOGRAM

```
pl <- ggplot(classPerformance, aes(x=mathScore))
```

The year axis is not compulsory when plotting an histogram

```
pl + geom_histogram()
```

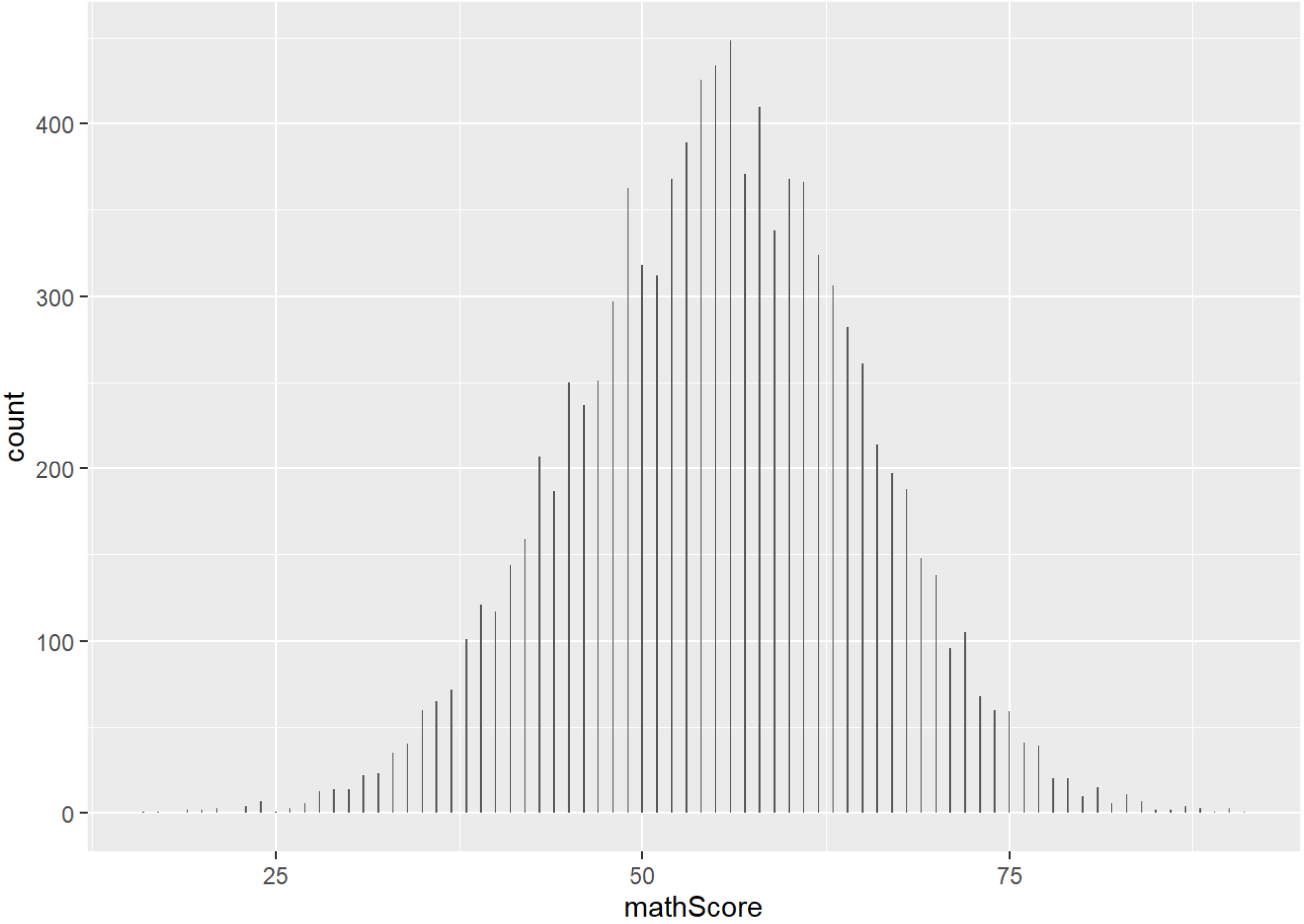
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



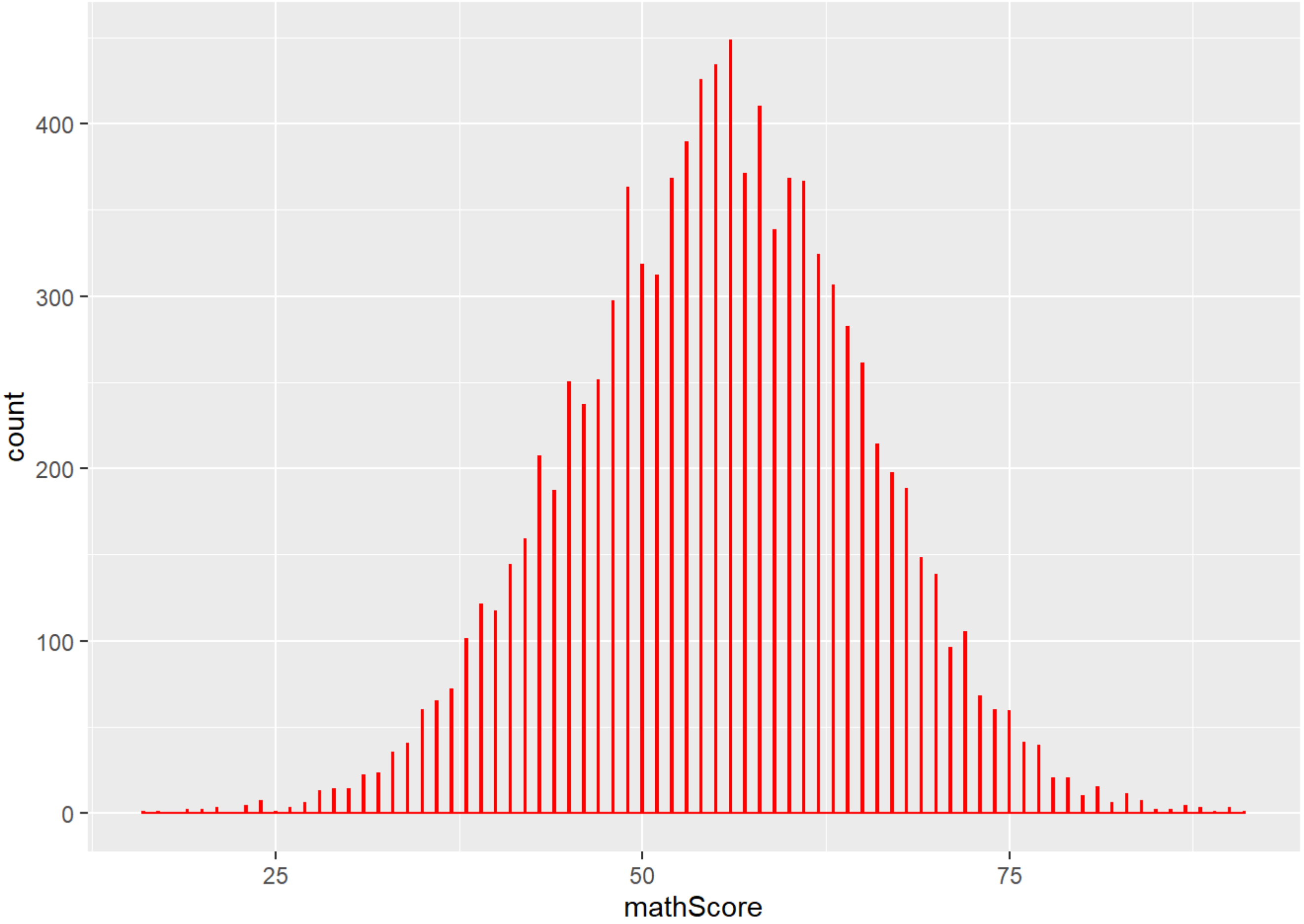
#

Addditional arguments to the geometric component.

```
pl + geom_histogram(binwidth = 0.1)
```



```
pl + geom_histogram(binwidth = 0.1, color="red", fill='pink', alpha=0)
```

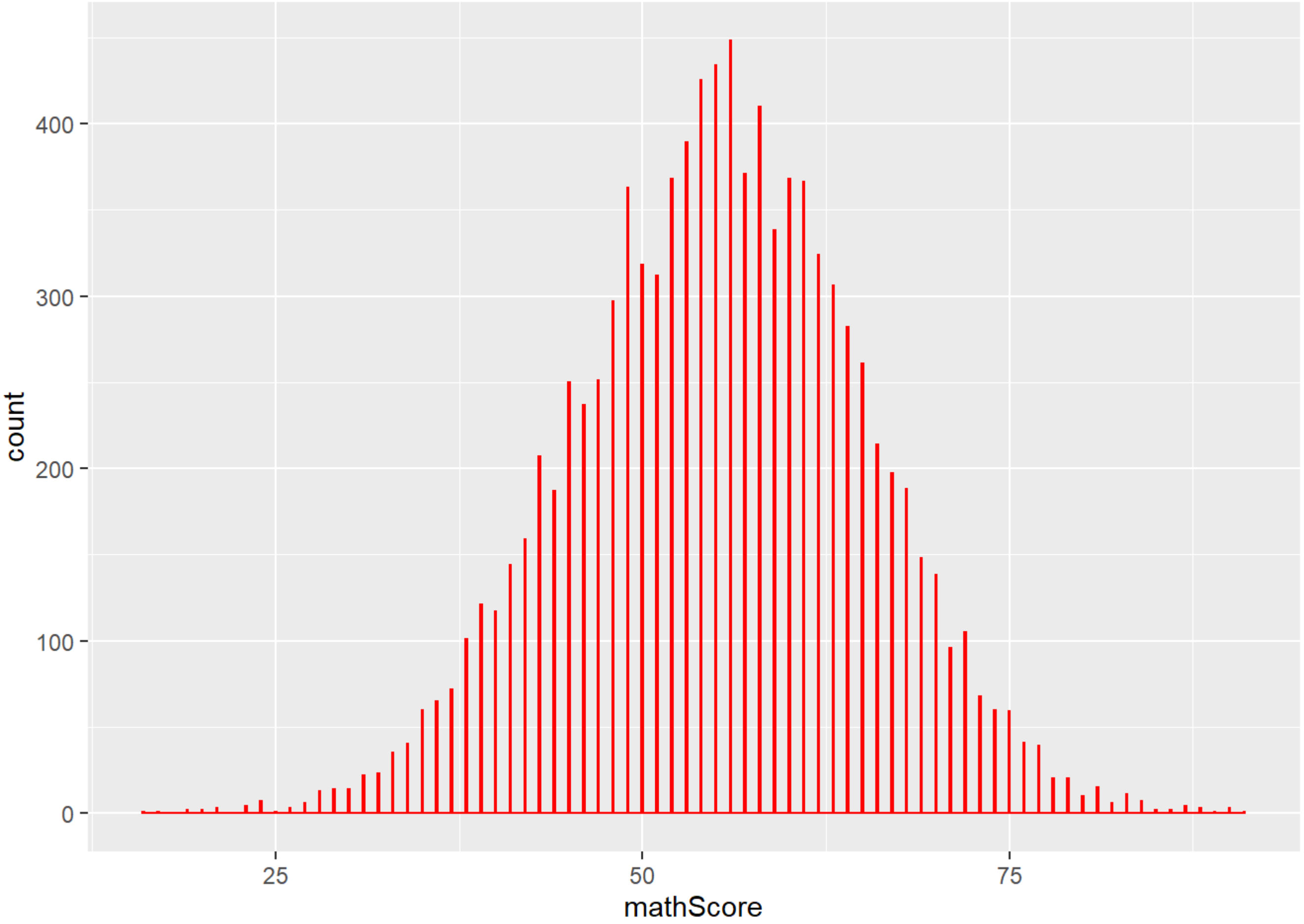


#

The alpha is for setting transparency. The default is 1

This shows the bar with the gridlines

```
pl + geom_histogram(binwidth = 0.1, color="red", fill='pink', alpha=0.4)
```

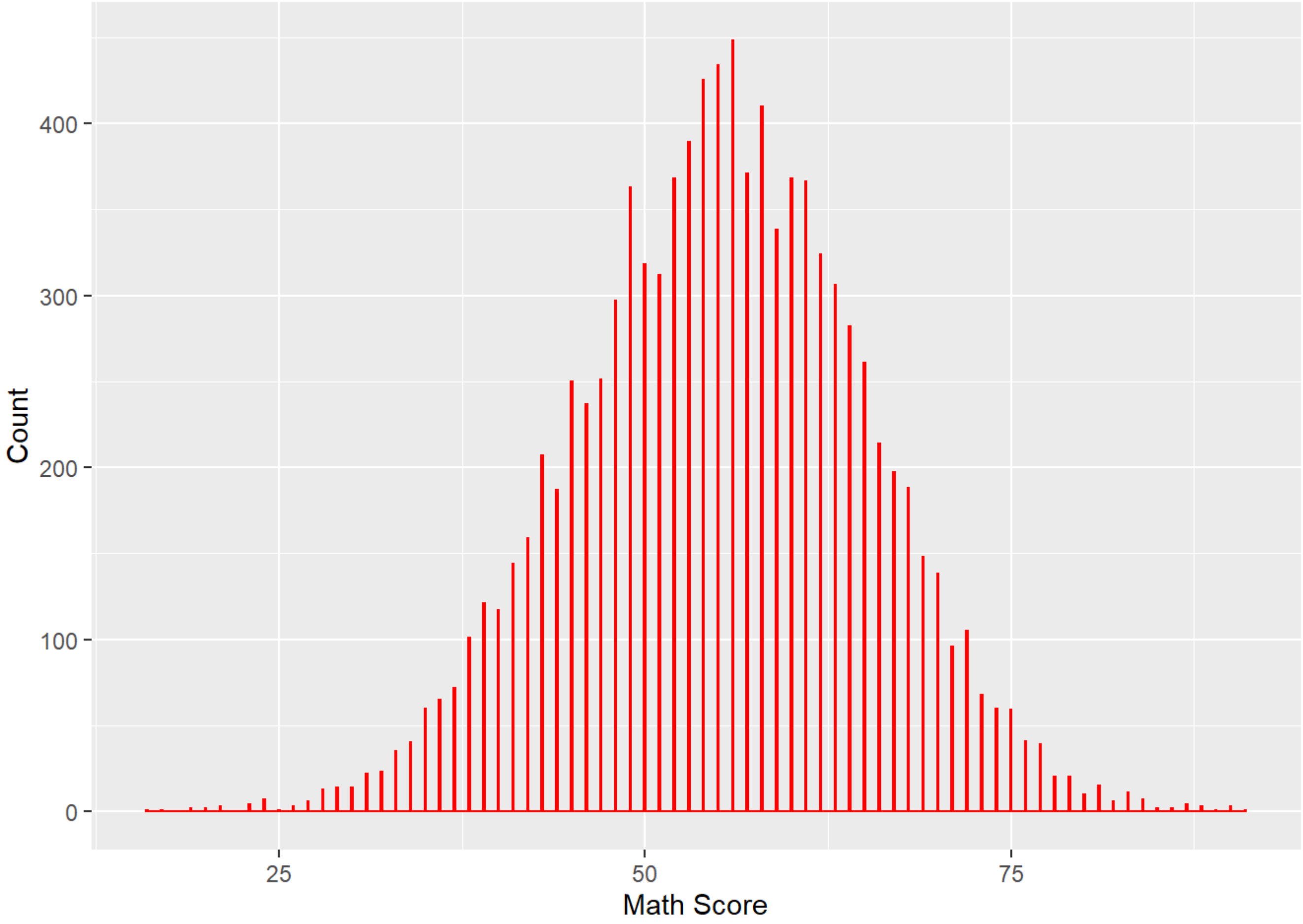


#

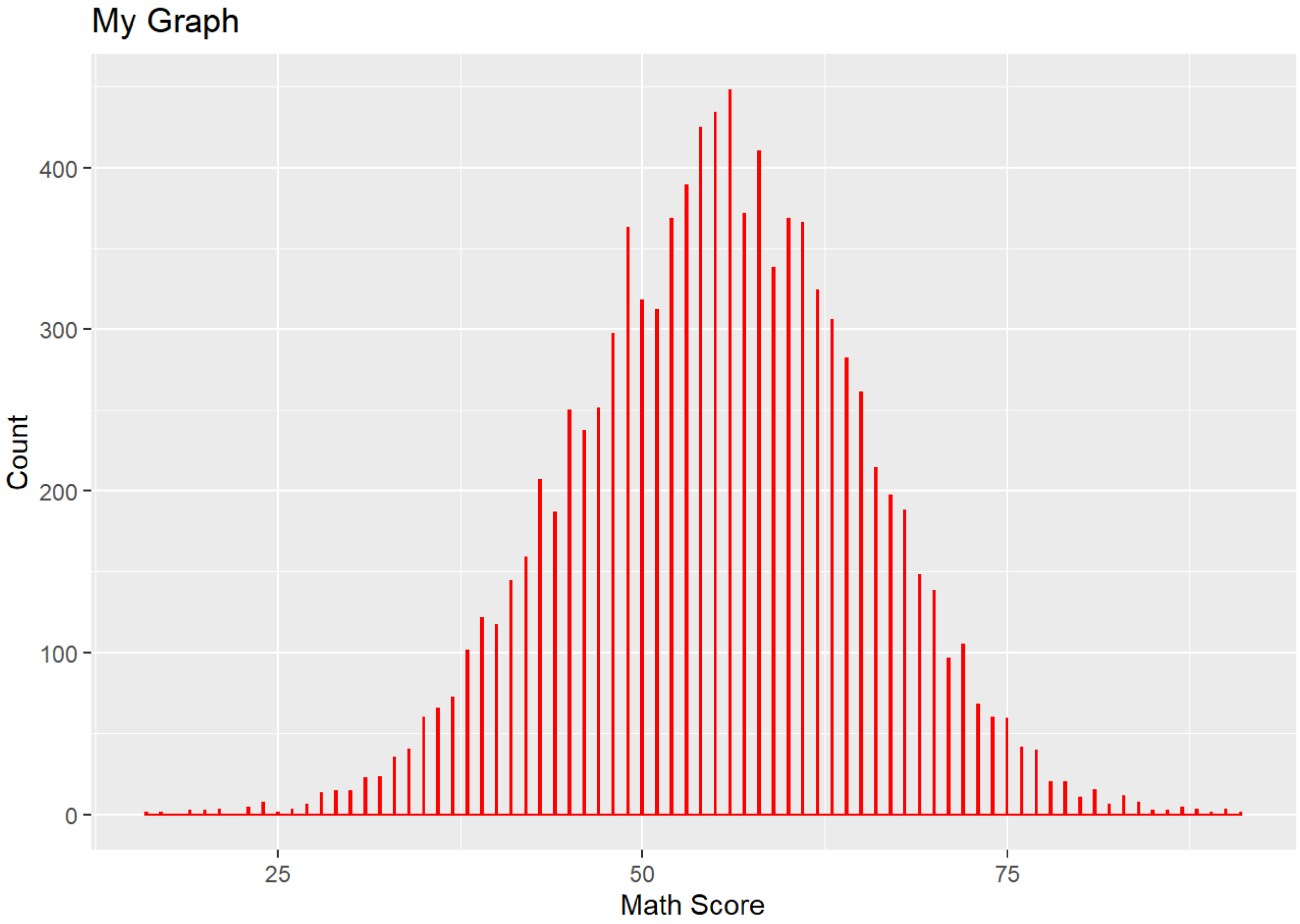
The line of code is getting too long. This is why I have to store it in pl1

```
pl1 <- pl + geom_histogram(binwidth = 0.1, color="red", fill='pink', alpha=0.4)
```

```
pl2 <- pl1 + xlab('Math Score') + ylab('Count')
print(pl2)
```



```
pl2 + ggtitle("My Graph")
```


My Graph
Count
400
300
200
100
0
25
50
75
Math Score