Assignment 3

Author: Amanda Everitt
Original Date: April 21, 2016
ID number: 998974934

Exercise 1:
Use sum() to determine the sum of numbers from 2000 to 20000.

```
sum(2000:20000)
```

```
## [1] 198011000
```

Exercise 2:
In one or two sentences, describe what the above code snippet it did.

```
a <- 5
b <- 2:20
```

It assinged variable 'a' as 5 and 'b' as a vector of every number 2 through 20.

Exercise 3:
Add the contents of a and b together and place the results in a new object. Try using both sum() and +; do you get different results? If so, why?

```
new <- sum(a,b)
new
```

```
## [1] 214
```

```
new1 <- a+b
new1
```

```
##  [1]  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

These are different results because a + b is iterative addition and sum is not.

Exercise 4:
What is the sum of the 5th through 10th element of object b?

```
sum(b[c(5:10)])
```

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

Exercise 5:
What is the sum of the 3rd, 8th, and 10th element of b?

```r
sum(b[c(3, 8, 10)])
```

```
## [1] 24
```

Exercise 6:
When extracting from a 2-dimensional object, which number specifies rows and which specifies columns? What does m[3,] do? How can you extract the 3rd, 4th and 5th columns of m together as one object?

m[row,column]
m[3,] would return row 3

```r
m <- matrix(data=1:25,ncol=5,byrow=T)
c(m[,3],m[,4],m[,5]) # will return list object
```

```
##  [1]  3  8 13 18 23  4  9 14 19 24  5 10 15 20 25
```

```r
subset(data.frame(m), select=c(X3,X4,X5)) # will return matrix object
```

```
##   X3 X4 X5
## 1  3  4  5
## 2  8  9 10
## 3 13 14 15
## 4 18 19 20
## 5 23 24 25
```

Exercise 7:
What does the cbind command do? How about rbind? Create a new object "n" where the first row is a new row of numbersand the following rows are the followed by the m matrix. Extra credit: do the same but reverse the order of the rows from m.

cbind and rbind combines two elements into the same matrix. cbind adds to columns where rbind adds rows.

```r
n <- rbind(100,m)     # will add new row
n2 <- rbind(100, apply(m,2,rev))  # will add new row with m reversed
n
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  100  100  100  100  100
## [2,]    1    2    3    4    5
## [3,]    6    7    8    9   10
## [4,]   11   12   13   14   15
## [5,]   16   17   18   19   20
## [6,]   21   22   23   24   25
```

```r
n2
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  100  100  100  100  100
## [2,]   21   22   23   24   25
## [3,]   16   17   18   19   20
## [4,]   11   12   13   14   15
## [5,]    6    7    8    9   10
## [6,]    1    2    3    4    5
```

Exercise 8:
How many hits have an e-value of 0? * How many have hits have a percent identity > 50? * Recalculate the above values but in percentage of hits rather than absolute values. * How many hits have an e-value of 0 and have a percent identity less than 50? * What is the minimum percent identity of the hits with an E-value equal to 0?

> 7531 hits have an e-value of 0 which is 1.5%
> 20928 hits have a percent identity greater than 50% which is 4.3%.
> 3351 hits have an e-value of 0 and percent identity less than 50%.
> 24.07% is the minimum percent identity with an E-value of 0. I just viewed the fly.worm.subE.pct dataset in the window and sorted by decreasing pct values.

```
fly.worm <- read.delim("~/Labwork/Rstudio/fly2worm.blastp.gz",header=F)
colnames(fly.worm) <- c("qid", "sid", "pct", "len", "mis", "gaps", "qb", "qe", "sb", "se", "E", "S")
table(fly.worm $E == 0)
```

```
##
##  FALSE   TRUE
## 496570   7531
```

```
table(fly.worm $pct > 50)
```

```
##
##  FALSE   TRUE
## 483173  20928
```

```
table(fly.worm $E == 0) / table(fly.worm $E !=0)
```

```
##
##       FALSE        TRUE
## 65.93679458  0.01516604
```

```
table(fly.worm $pct > 50) / table(fly.worm $pct <= 50)
```

```
##
##       FALSE        TRUE
## 23.08739488  0.04331368
```

```
fly.worm.subE.pct <- subset(fly.worm, fly.worm$E == 0 & fly.worm$pct < 50)    # gives 3351  objects in
```

Exercise 9: Are you surprised that sequences with relatively low percent identity can still have an E-value of 0? * State a hypothesis about what alignment properties might produce a zero E-value even when the percent identity is less than 50%. * Test your hypothesis

Yes, because generally low E-values indicate good alignments which would have high percent identities. However, if the sequence alignment is long it can still produce a zero E-value even with a low percent identity because it will map to more places than a short alignment with a high percent identity.

Within a dataset with the same E value of 0, on average lower percent identity lengths are 1611bp whereas higher percent identity lengths are 730bp. This means having a long alignment with low percent identity is comparable to a short alignment with high percent identity. This supports my hypothesis that length is important in determining E value along with percent identity.

```
summary(fly.worm.subE.pct$len) #len of everything with percent <50 and E =0
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     552     930    1267    1611    1897    5795
```

```
fly.worm.subE.pct50 <- subset(fly.worm, fly.worm$E == 0 & fly.worm$pct >= 50)
summary(fly.worm.subE.pct50$len) #len of everything with percent >= 50 and E =0
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   277.0   462.0   591.0   730.3   883.0  4625.0
```

Excercise 10:
It sometimes is useful to create a random subset of a large data set like the fly.worm results. Create a new dataframe fly.worm.small that has 10,000 rows taken at random from the larger data set.

```
fly.worm.small <- fly.worm[sample(nrow(fly.worm),10000),]
```

Exercise 11:
Use the summary() and head() functions to examine the stereotypes object.

```
stereotypes <- read.csv("~/Labwork/Rstudio/stereotypes.csv",header=T)
#summary(stereotypes)
#head(stereotypes)
```

Activity Break:
1.Subset the data to only nerds and metalheads.

```
stereo.sub.nerd.metal <- subset(stereotypes, population == 'nerd' | population == 'metalhead')
```

2.Are there more males who binge drink (over 25 a week) and eat over 20 tacos a week in the metal heads or hipster population?

There are more metalheads (68) that eat a lot of tacos and drink beer than hipsters(4).

```
hipster.sub <- subset(stereotypes, gender == 'male' & population == 'hipster' & tacos > 20 & beer > 25,s
metal.sub <- subset(stereotypes, gender == 'male' & population == 'metalhead' & tacos > 20 & beer > 25,s
summary(hipster.sub)
```

```
##     gender      population      beer          tacos
##  female:0   hippie   :0    Min.   :26.0   Min.   :21.0
##  male  :4   hipster  :4    1st Qu.:27.5   1st Qu.:21.0
##             metalhead:0    Median :29.0   Median :21.0
##             nerd     :0    Mean   :30.0   Mean   :21.5
##                            3rd Qu.:31.5   3rd Qu.:21.5
##                            Max.   :36.0   Max.   :23.0
```

```
summary(metal.sub)
```

```
##     gender       population      beer           tacos
##  female: 0   hippie   : 0   Min.   :26.00   Min.   :21.00
##  male  :68   hipster  : 0   1st Qu.:37.00   1st Qu.:23.00
##              metalhead:68   Median :43.00   Median :25.50
##              nerd     : 0   Mean   :46.38   Mean   :26.43
##                             3rd Qu.:56.50   3rd Qu.:28.25
##                             Max.   :80.00   Max.   :42.00
```

3.Ask one more question about the dataset that you are interested in. Subset the data accordingly.

The maximum beers any female drank was 73.

```
female.sub <- subset(stereotypes, gender == 'female',select = c(gender,beer))
summary(female.sub)
```

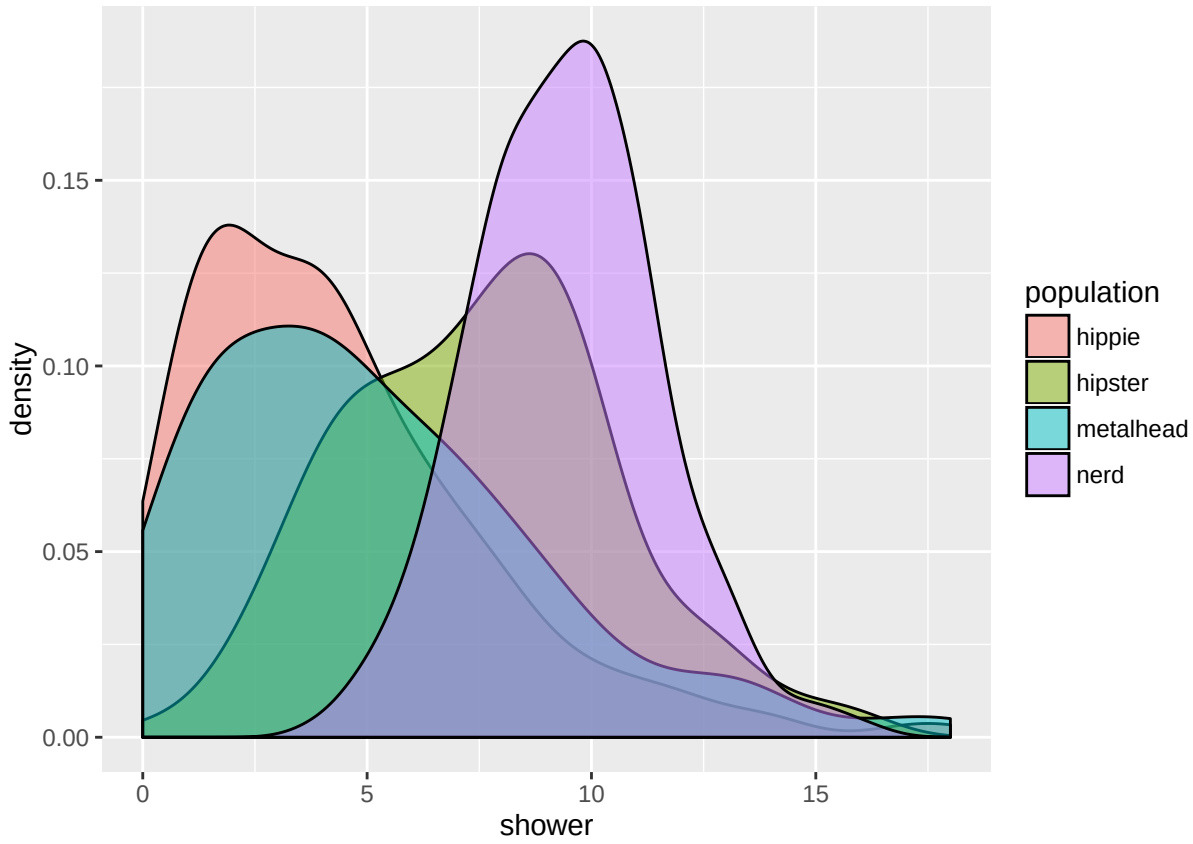
```
##     gender         beer
##  female:400   Min.   : 0.00
##  male  :  0   1st Qu.: 3.00
##               Median :15.50
##               Mean   :19.24
##               3rd Qu.:33.00
##               Max.   :73.00
```

4.In your own words explain what "==" means?

is

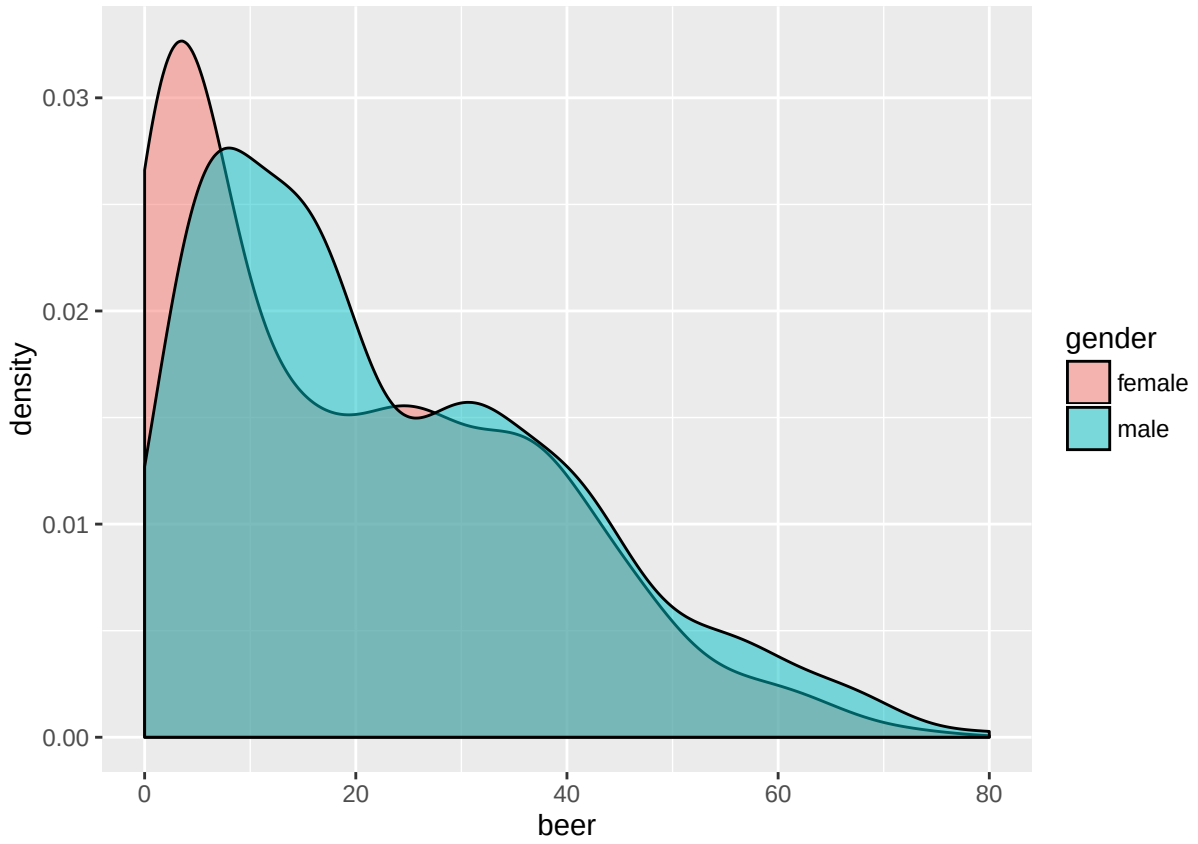
Activity Break:
1.View the distribution of showers. Color by population.

```
library(ggplot2)
qplot(shower, data = stereotypes, geom = "density", alpha = I(0.5), fill = population)
```



2.View the distribution of beer consumption and color by gender.

```
qplot(beer, data = stereotypes, geom = "density", alpha = I(0.5), fill = gender)
```

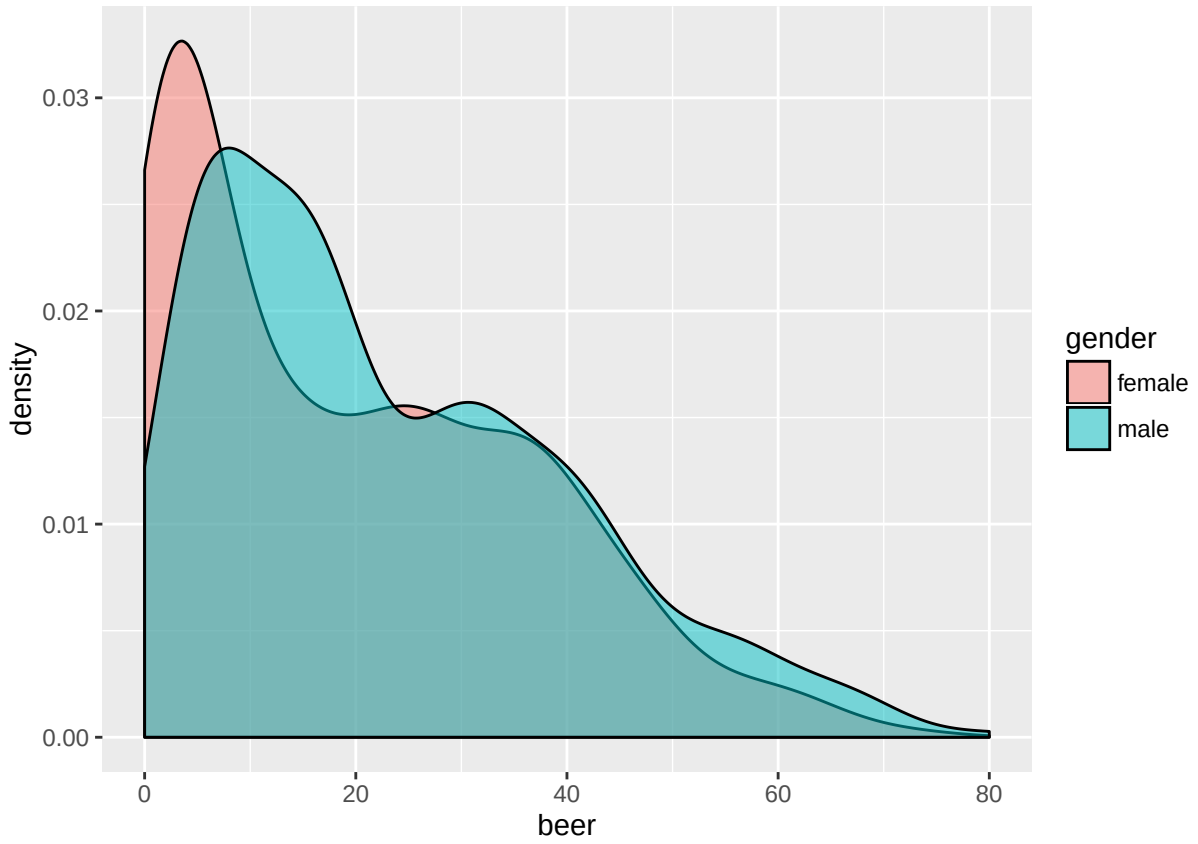


3.Find a way to create a plot using the the geom violin with the stereotype data set.

```
qplot(age, beer, data = stereotypes, geom = "violin")
```

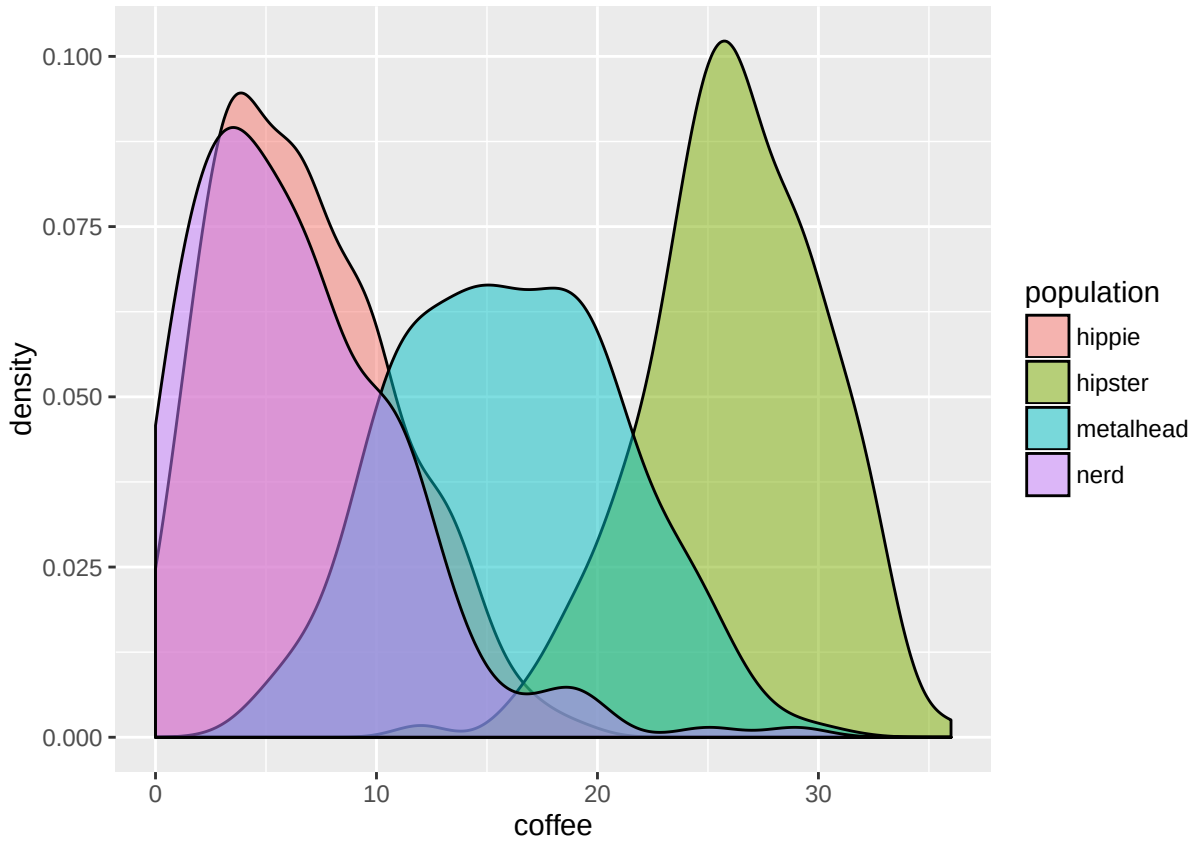
Activity Break:
1.Make a graph that visually describes the difference between beer consumption between gender in the nerd category.

```
nerds <- subset(stereotypes, population = 'nerds')
qplot(beer, data = nerds, geom = "density", alpha = I(0.5), fill = gender)
```



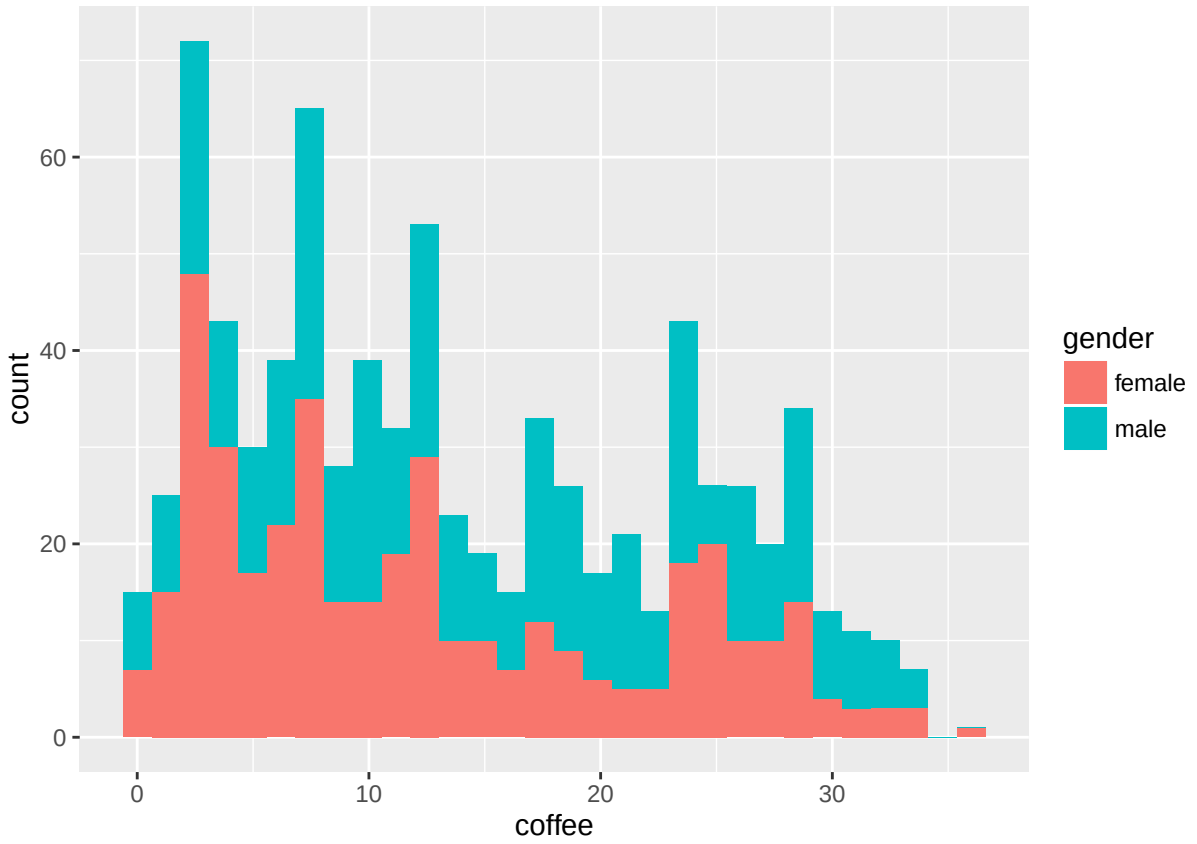
2.Using theggplot2 documentation, explore three different geoms, using either the full data set or a subset of the data.

```
qplot(coffee, data = stereotypes, geom = "density", alpha = I(0.5), fill = population)
```

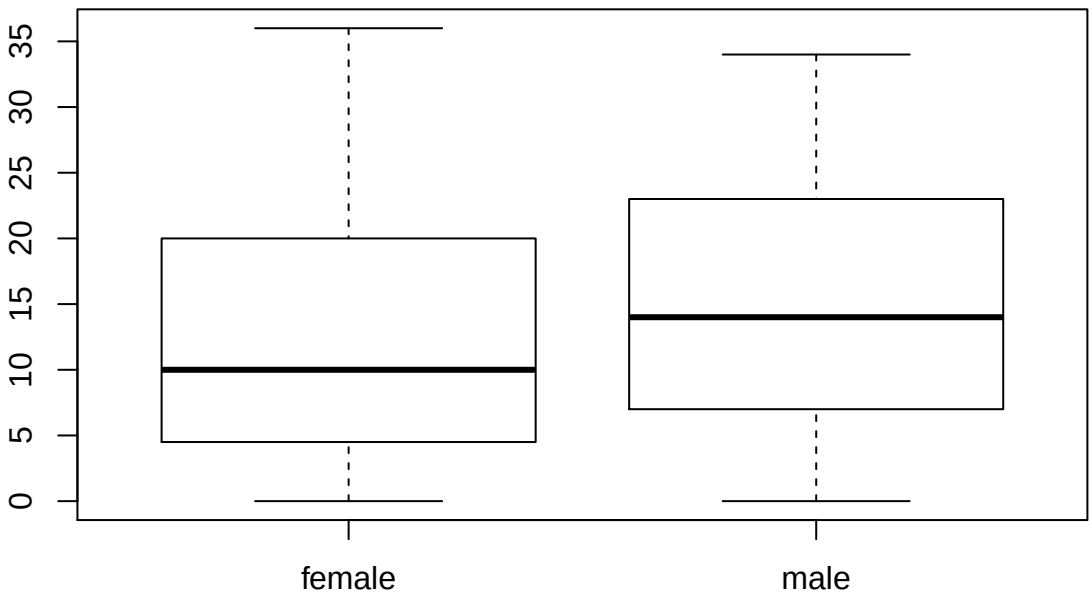



```
qplot(coffee, data = stereotypes, geom = "histogram", fill = gender)
```

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




```
hipsters <- subset(stereotypes, population = 'hipsters')
boxplot(hipsters$coffee ~ hipsters$gender)
```



3.From your exploration of the stereotypes data make three hypotheses about the data.

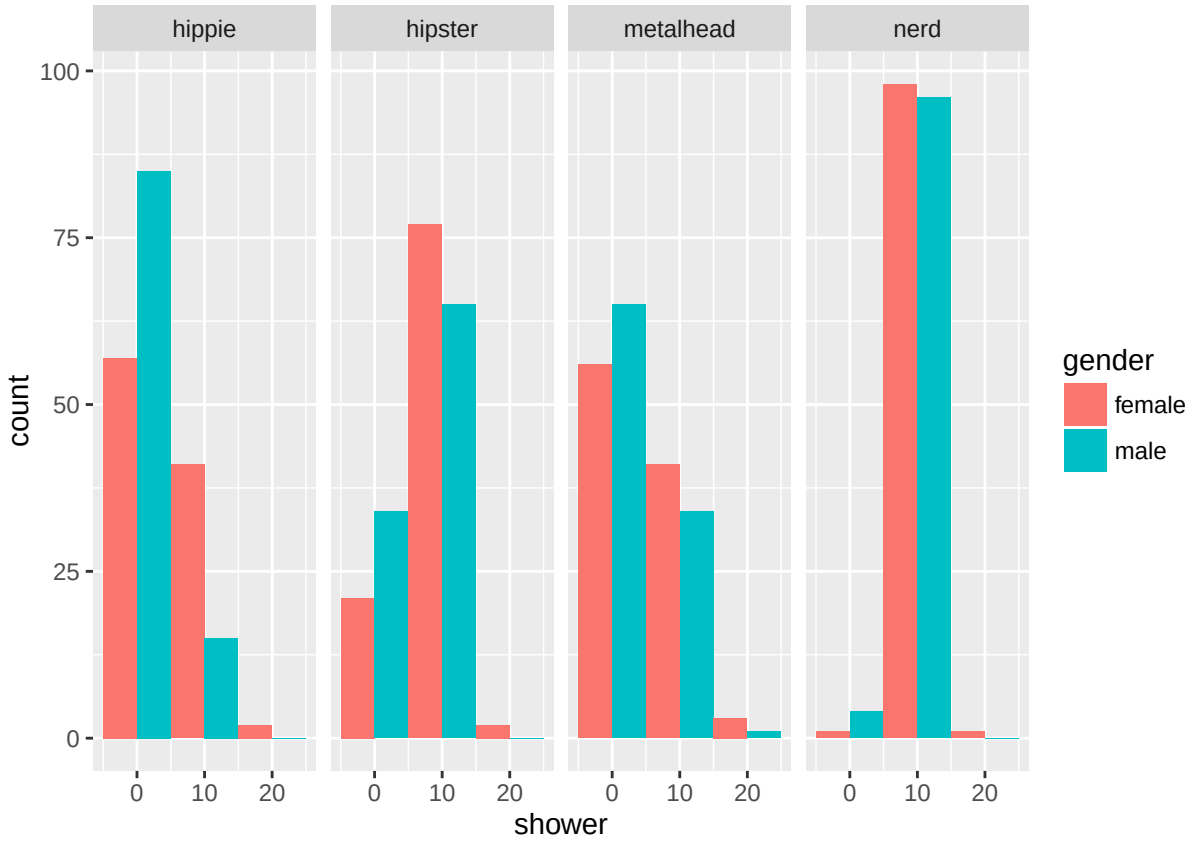
On average, hipsters drink more coffee than the other populations.

Generally, males drink more coffee than females.

Within the hipster population, females and males relatively drink the same amount of coffee which could be why the hipster population has a disproportionately high proportion of coffee consumption.

Activity Break:
1.With the p base layer, make a unique plot by adding additional layers.

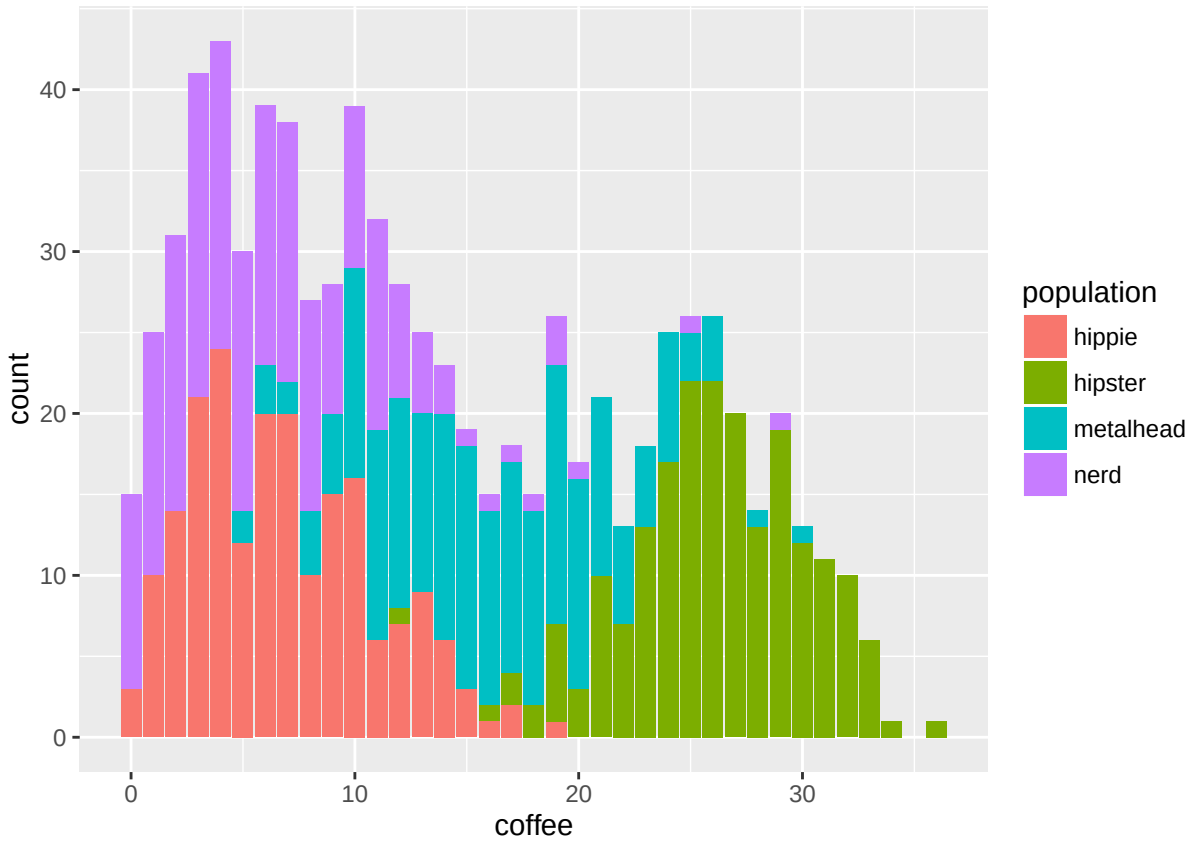
```
p <- ggplot(stereotypes, aes(shower, fill = gender))
p + geom_histogram(binwidth = 10, position = 'dodge') + facet_grid(~population)
```



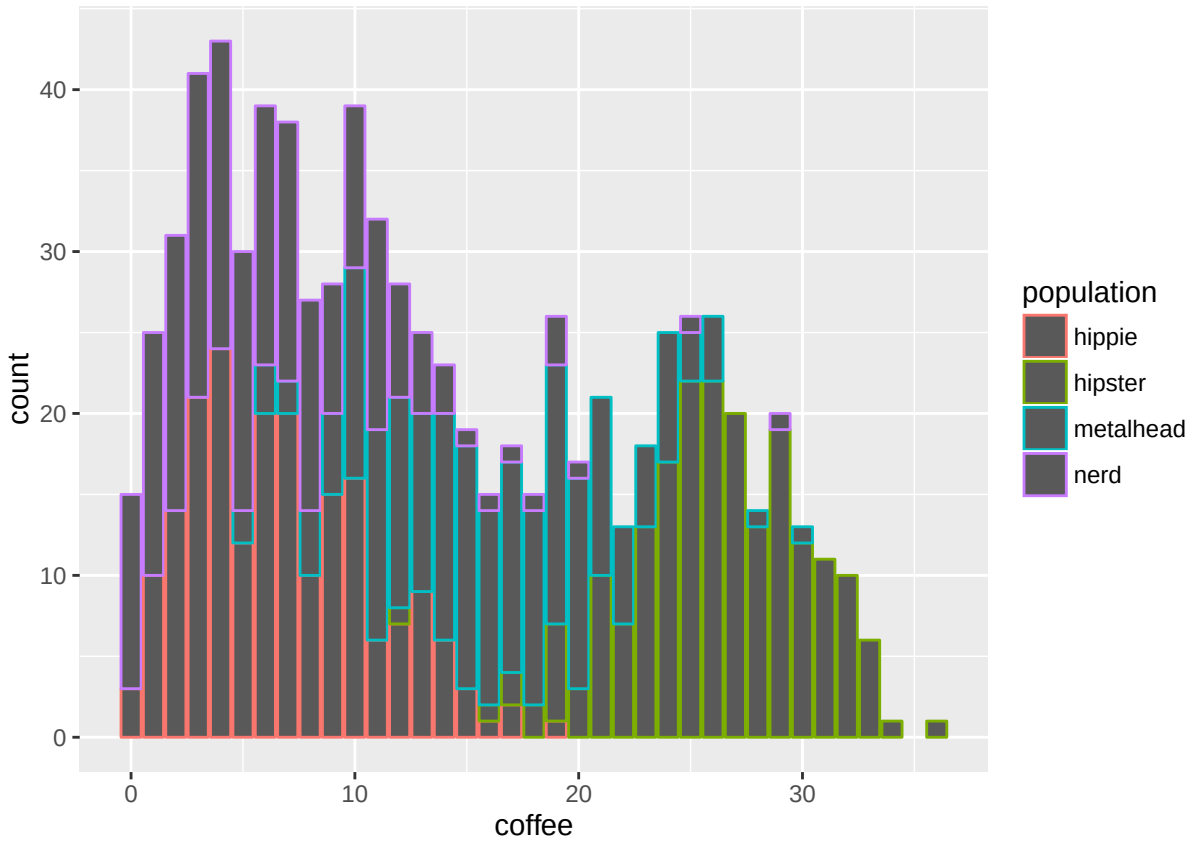
2.Make new base layer. Explore what the difference is between color = and fill =. Use your own words to discern the difference.

'Fill' colors in the graphs completely, whereas 'color' just outlines them.

```
q <- ggplot(stereotypes, aes(coffee, fill = population))
q + geom_bar()
```

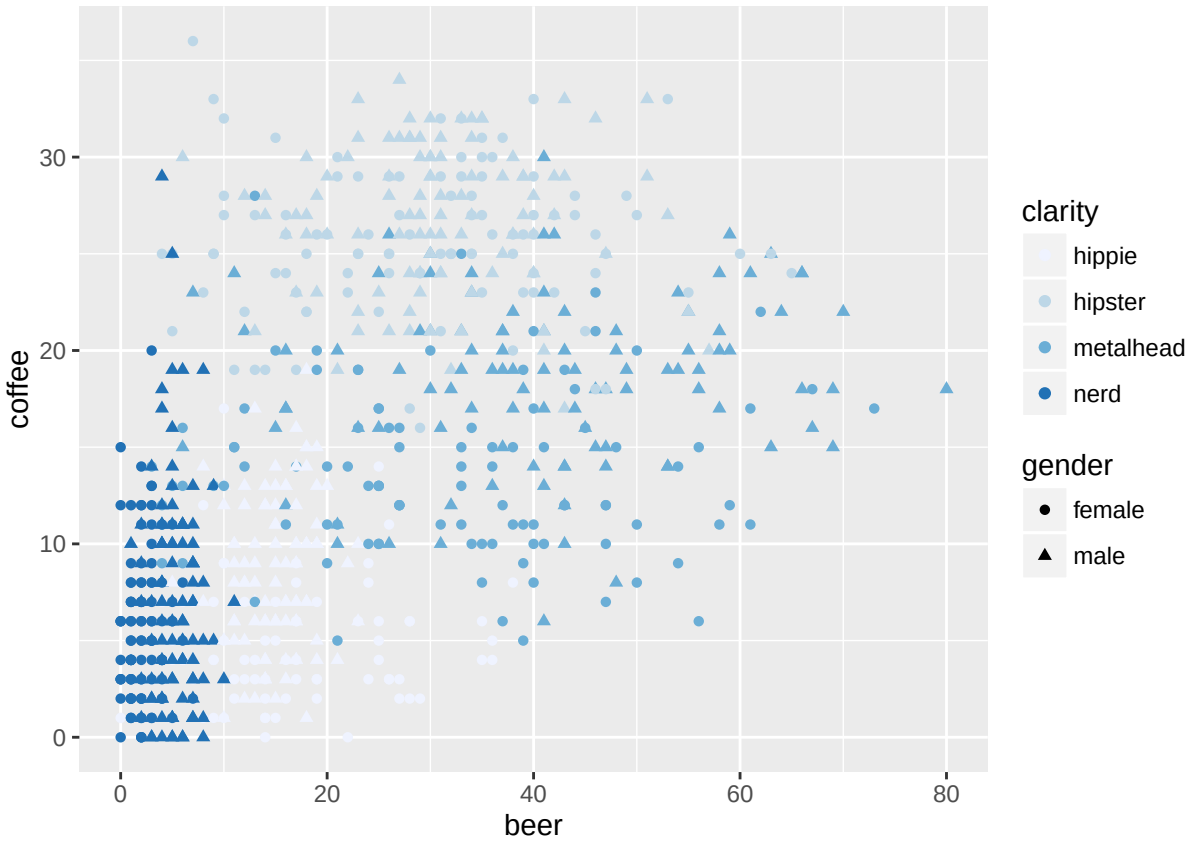



```
q1 <- ggplot(stereotypes, aes(coffee, color = population))
q1 + geom_bar()
```



Activity Break:
1.Make a tricked out box plot using the stereotypes data. Use at least 3 of the aesthetic options available for geom_boxplot. Use ggplot2 boxplot documentation for guidance.

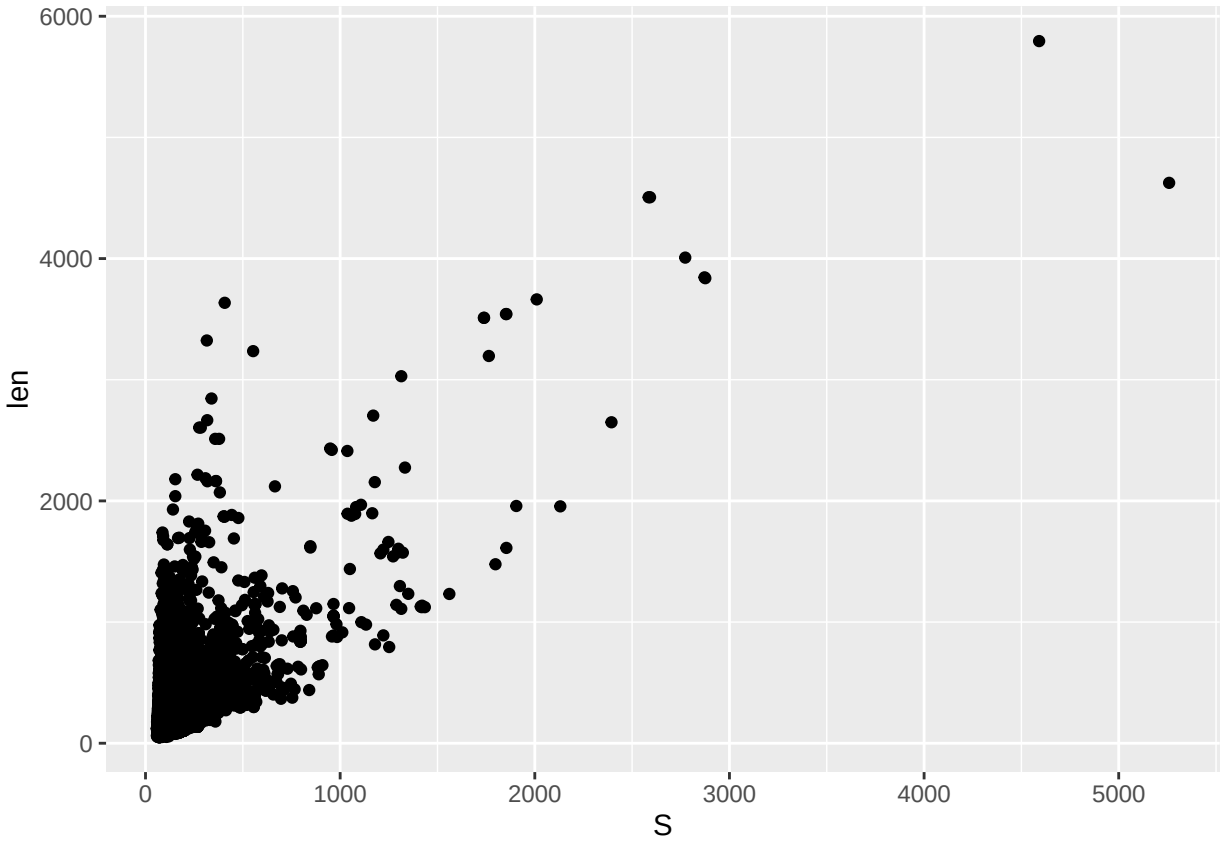
```
ggplot(stereotypes, aes(beer, coffee, color = population, shape =gender)) + geom_point() + scale_colour
```



Exercise 12:
Use ggplot to explore the relationship between score ("S") and alignment length("align_length"). Provide a plot that illustrates the relationship and describe the relationship in words.

As the length of the alignment decreases, the score also decreases.

```
r <- ggplot(fly.worm.small, aes(S, len))
r + geom_point()
```



Excercise 13:
While you might expect that BLAST results with long alignments would have high scores, this is not always the case. Form a hypothesis as to what might influence the relationship between alignment length and score. Use ggplot to make a new plot to explore this hypothesis. Does the plot support your hypothesis or not? State your hypothesis, provide the code for your plot, and state your conclusion.

> My hypothesis is that the long alignments will have more gaps, which may be penalized heavily in the BLAST parameters resulting in low scores.

> First, I determine an arbitrary cut off for what is a "long" alignment.

```
summary(fly.worm$len) #Will use 387.0 as cut off for what is "long"
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    32.0   203.0   268.0   343.1   387.0  5795.0
```

> Next, I establish that longer alignments do in fact have more gaps on average. Long alignments average 21 gaps versus 6 gaps in comparison.

```
long <- subset(fly.worm, len > 387)
notlong <- subset(fly.worm, len <= 387)
summary(long$gaps)
```

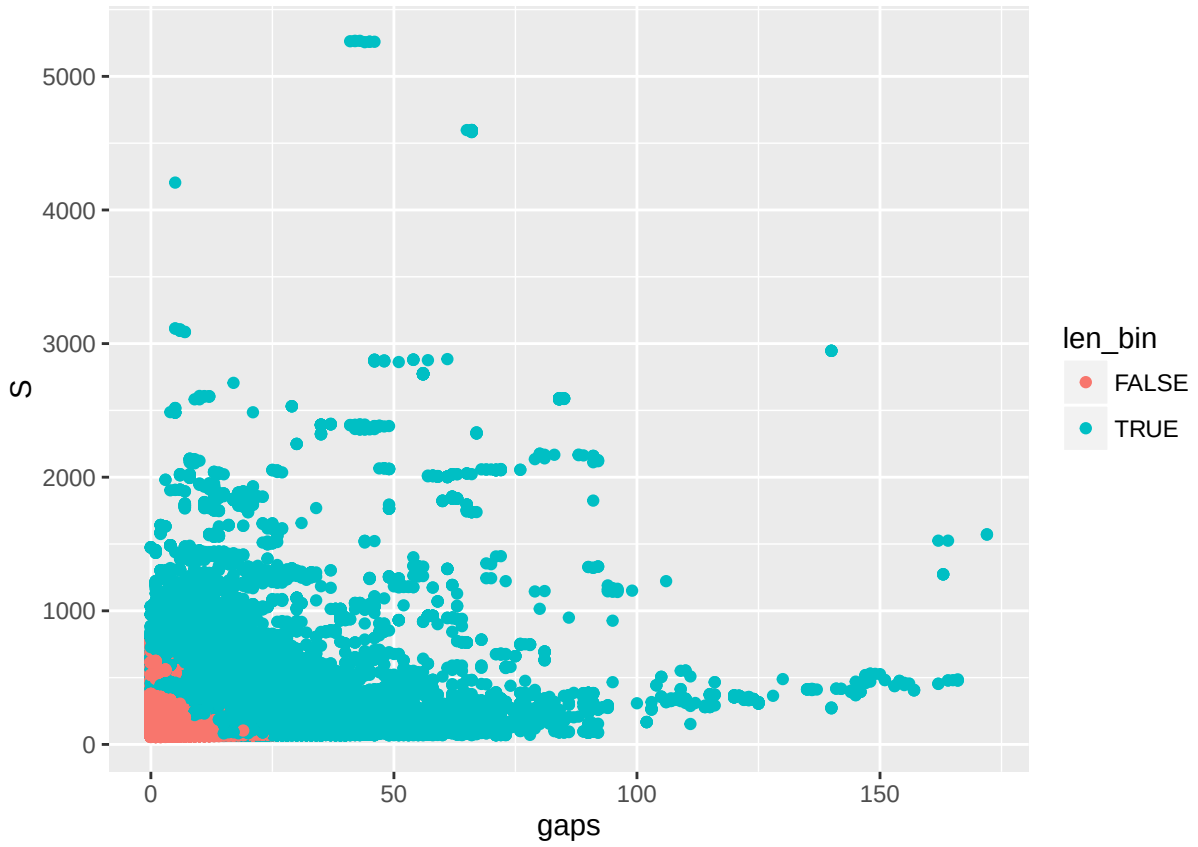
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   13.00   19.00   21.48   27.00  172.00
```

```
summary(notlong$gaps)
```

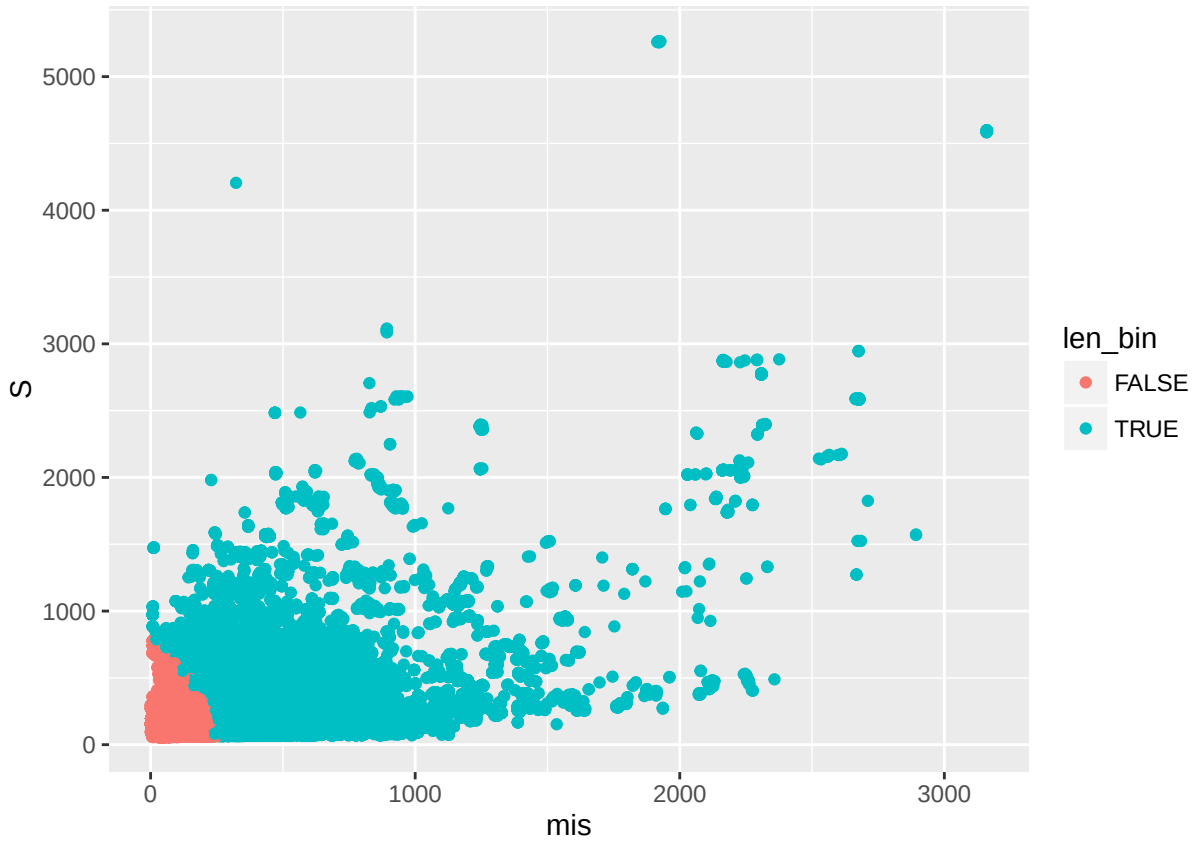
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   6.000   5.956   8.000  27.000
```

Next, I establish the penalization of gaps is signficant in our BLAST. In the following graphs blue represents long alignments. The first graph shows that even with long alignments, the larger number of gaps results in a lower scores. The second graph shows that gaps appear to be penalized more stringently than mismatches.

```
fly.worm$len_bin <- fly.worm$len>387
ggplot(fly.worm, aes(gaps, S, color = len_bin)) + geom_point()  # where TRUE =long seq
```




```
ggplot(fly.worm, aes(mis, S, color = len_bin)) + geom_point()  # where TRUE =long seq
```


5000
4000
3000
2000
1000
0
s
0
1000
2000
3000
mis
len_bin
FALSE
TRUE