Analyzing Genome-scale data in (Non)Brood Parasitic Birds

Introduction

The 2020 Harvard Research for undergraduates Program was funded by the National Science Foundation. The proposed project revolved around analyzing the genome-scaled data of brood parasitic birds. Although the project focused on bird genomes, the skills that were learned are broadly applicable in many science-realted fields. It allowed me to understand that the methods and science behind evolutionary biology are central to the future of personalized medicine for humans as well. I learned how computational biology is relevant to the environment and boosted my skills in computer programming, which will be very useful in a variety of careers. This project characterized patterns of genomic change in both parasitic and non-parasitic birds to test whether changes in the same specific genes and/or parallel patterns of genomic change have occurred in independent brood parasitic lineages. I was able to successfully identify unknown pieces of chromosomes and categorize them as Z-linked, W-linked, or autosomal. The project also examined the unique evolutionary dynamics of the sex-determination chromosomes. The results provided insight into fundamental questions about genome evolution and will provide other reseearchers with mapped genomes of speicies and reliable references that will further their research. Below you will find a breif summary of the codes and steps I used to analyze the data given to me.

To install the tidyverse package, use `install.packages("tidyverse")`

##In order to use the package you must load the package every time you start a new session

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(knitr)
```

Made a reference genome based on a female reference genome obtained from NCBI.

```
module load bwa/0.7.17-fasrc01
bwa index -p hetAtr 00_genome/GCA_011075105.1_BPBGC_Hatr_1.0_genomic.fna.gz
```

The sample genes were mapped. A sam file was created for each sample.

```
module load bwa/0.7.17-fasrc01

R1=$1
name=`echo $R1 | sed 's/_1.fastq.gz\+//'`
R2=${name}_2.fastq.gz
#run mapping
bwa mem -t 8 -R '@RG\tID:${name}\tSM:${name}' 00_genome/hetAtr $R1 $R2 > $name.sam
```

Once the same files were converted into bam files, one large bam file was made for each of the 14 indiviuals that contained all the information pertaining to the individual.

```
module load samtools/1.10-fasrc01
samtools merge -f -b indv1.txt indv1.final.bam
samtools merge -f -b indv2.txt indv2.final.bam
samtools merge -f -b indv3.txt indv3.final.bam
```

The coverage of each indivdual was calculated in order to determine the scaffolds associated with sex chromosomes by comparing between males and females.

```
module load samtools/1.10-fasrc01
samtools coverage indv1.final.bam > indv1.out
samtools coverage indv2.final.bam > indv2.out
```

Identified three possible autosomal scaffolds to explore coverage and compare other scaffolds against and make sure sex chromosomes were actually sex chromosomes in the males and females:

CM021752.1 1 5558834 CM021757.1 1 5765391 CM021758.1 1 5737139

#COMBINING AUTOSOMAL AND SEX CHROMOSOMES TO PLOT FOR EVERY INDIVIDUAL TO INDENTIFY GENDER #Of the 14 individuals, Males and Females were identified since, compared to autosomes, males should have an overlapping rate and females should have a different rate.

#Example: FEMALE (indiviual 8)

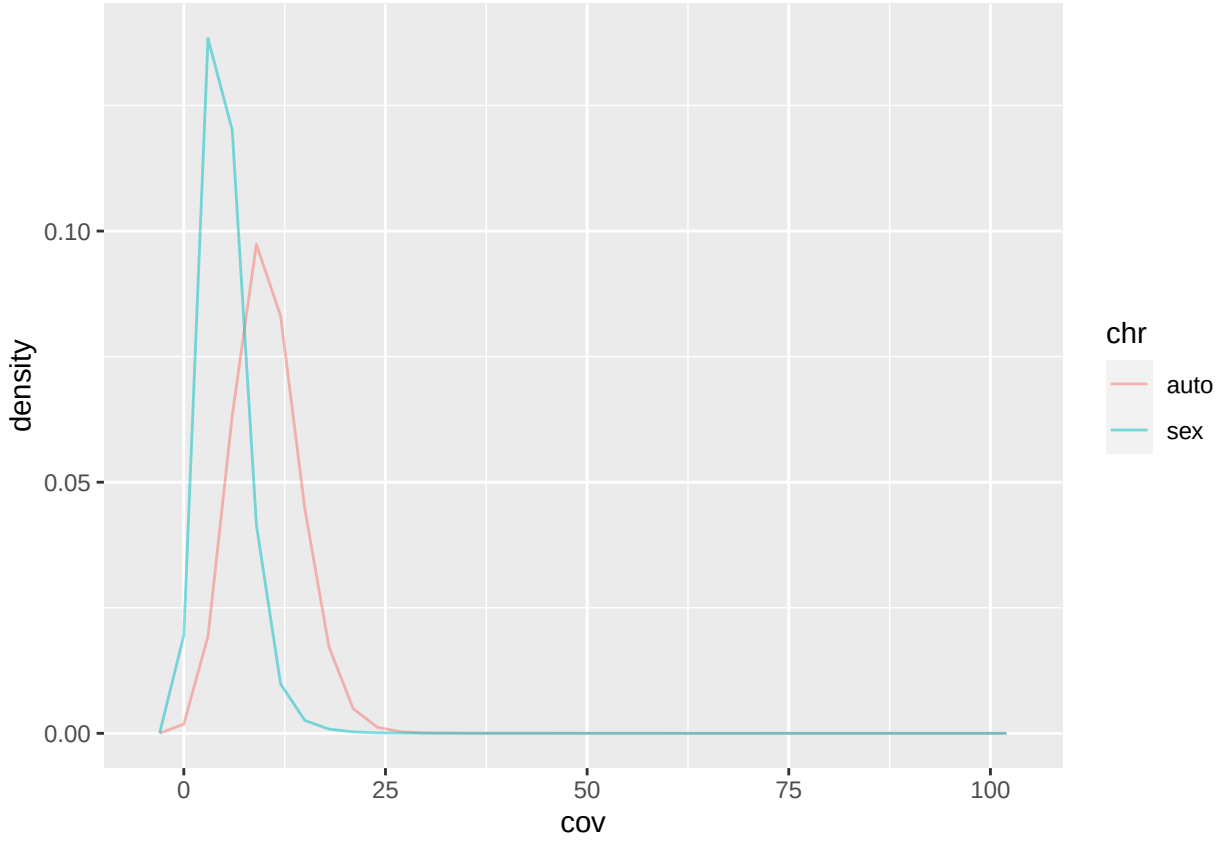
```
auto <- read_delim('auto_cov8.txt', delim = '\t', col_names = F) %>%
  rename(scaffold = X1, pos = X2, cov = X3) %>%
  mutate(chr = 'auto') %>%
  select(chr, cov)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_double(),
##   X3 = col_double()
## )
```

```
sex <- read_delim('i8.txt', delim = '\t', col_names = F) %>%
  rename(scaffold = X1, pos = X2, cov = X3) %>%
  mutate(chr = 'sex')
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_double(),
##   X3 = col_double()
## )
```

```
as <- bind_rows(auto, sex) %>% filter(cov < 100)
ggplot(as, aes(cov, color = chr,y = ..density..)) + geom_freqpoly(alpha = 0.5, position = 'identity', bi
```



#x-axis: coverage, y-axis: density, The autosomes and sex chromosomes of individual 8 shows that the rates of each set of chromosomes differ, this holds to be true in females because the autosomes have half the data that females hold as they have heterozygous sex chromosomes (ZW).

#Example: MALE (individual 3)

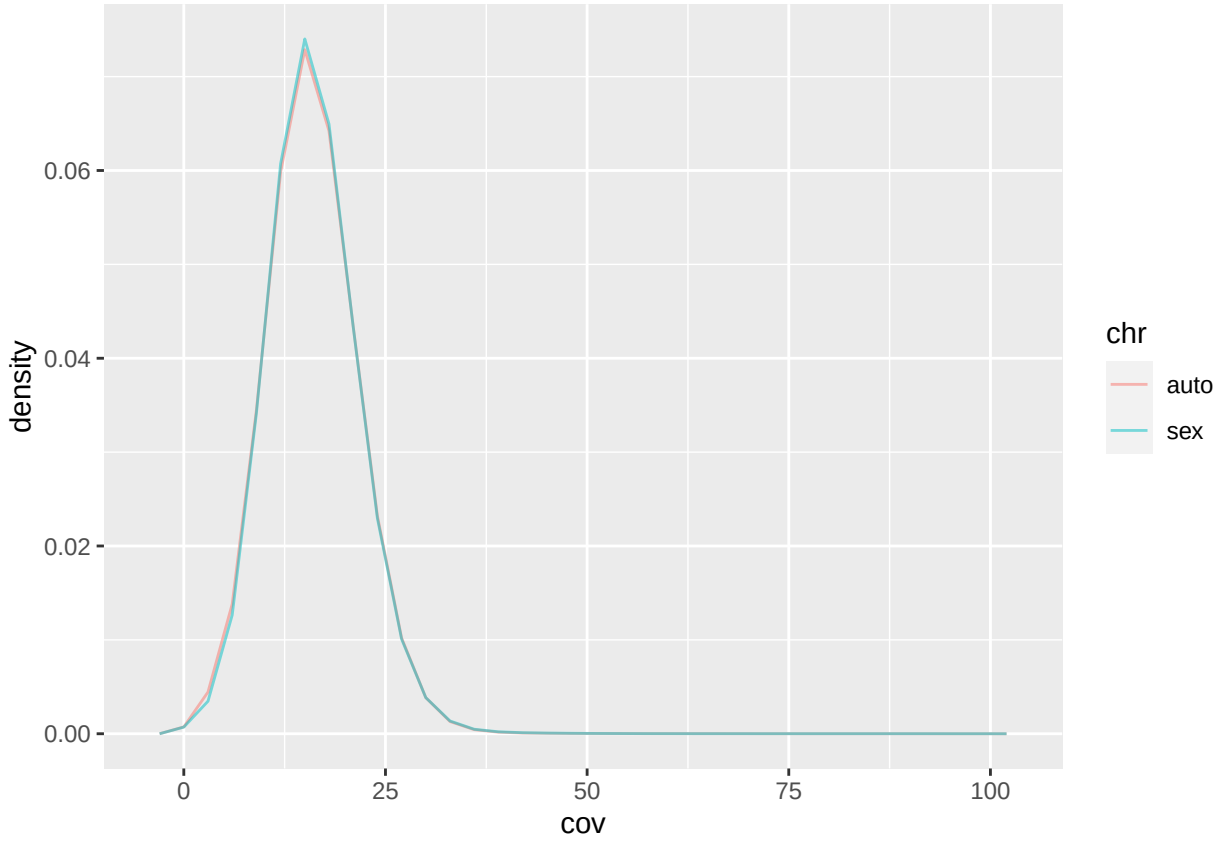
```
auto <- read_delim('auto_cov3.txt', delim = '\t', col_names = F) %>%
  rename(scaffold = X1, pos = X2, cov = X3) %>%
  mutate(chr = 'auto') %>%
  select(chr, cov)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_double(),
##   X3 = col_double()
## )
```

```
sex <- read_delim('i3.txt', delim = '\t', col_names = F) %>%
  rename(scaffold = X1, pos = X2, cov = X3) %>%
  mutate(chr = 'sex')
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_double(),
##   X3 = col_double()
## )
```

```
as <- bind_rows(auto, sex) %>% filter(cov < 100)
ggplot(as, aes(cov, color = chr,y = ..density..)) + geom_freqpoly(alpha = 0.5, position = 'identity', bi
```



#x-axis: coverage, y-axis: density, The autosomes and sex chromosomes of individual 3 shows that the rates of each set of chromosomes overlap each other. This holds to be true in males because the autosomes have the same the data that males hold as they have homozygous sex chromosomes (WW).

Merged all females into one bam file and all males into another bam file

```
samtools merge fem.final.bam indv4.final.bam indv5.final.bam indv6.final.bam indv7.final.bam indv8.final
```

```
samtools merge male.final.bam indv1.final.bam indv2.final.bam indv3.final.bam indv11.final.bam indv12.fi
```

Identified scaffolds associated with sex chromosomes by calculating covergae between the male and female larger combined files

```
samtools coverage fem.final.bam > female.out
samtools coverage male.final.bam > male.out
```

FINDING THE TOTAL COVERAGE OF MALES AND FEMALES

MALES

```
data <- read_delim('male_depth.txt', delim = '\t', col_names = F) %>%
  rename(scaffold = X1, pos = X2, cov = X3)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_double(),
##   X3 = col_double()
## )
```

```
mean(data$cov)
```

```
## [1] 116.4228
```

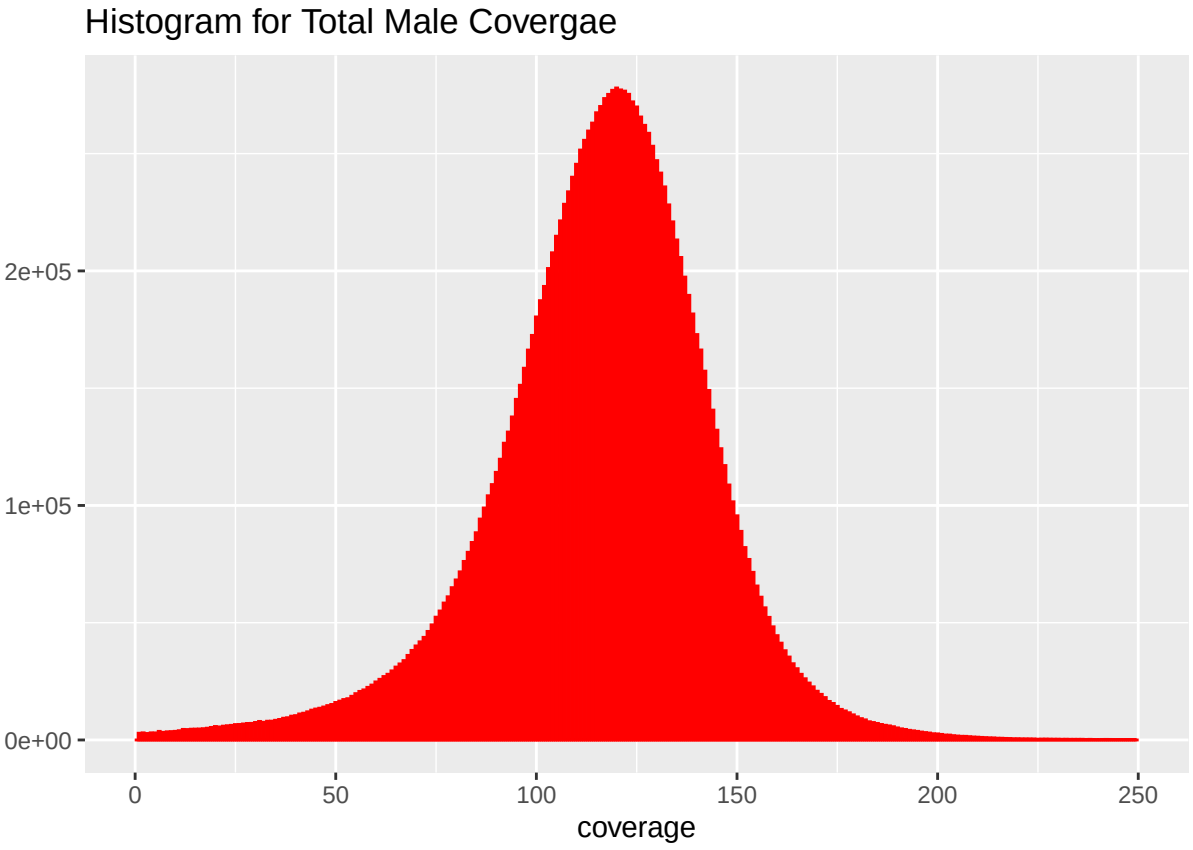
```
median(data$cov)
```

```
## [1] 118
```

```
Totalcovg <- qplot(data$cov,
      geom="histogram",
      binwidth = 0.5,
      main = "Histogram for Total Male Covergae ",
      xlab = "coverage",
      fill=I("blue"),
      col=I("red"),
      alpha=I(.2),
      xlim=c(0,250))
Totalcovg
```

```
## Warning: Removed 17004 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



#x-axis: coverage, y-axis: number of scaffolds. The total coverage was found for all males to ensure that important scaffolds were included.

Calculated the mean and median of the 3 autosomal scaffolds in males to ensure the numbers used for coverage and results obtained were reasonable and not largely skewed

```
Scaf123mean <- data %>% group_by(scaffold) %>% summarize(Avgcov = mean(cov))
```

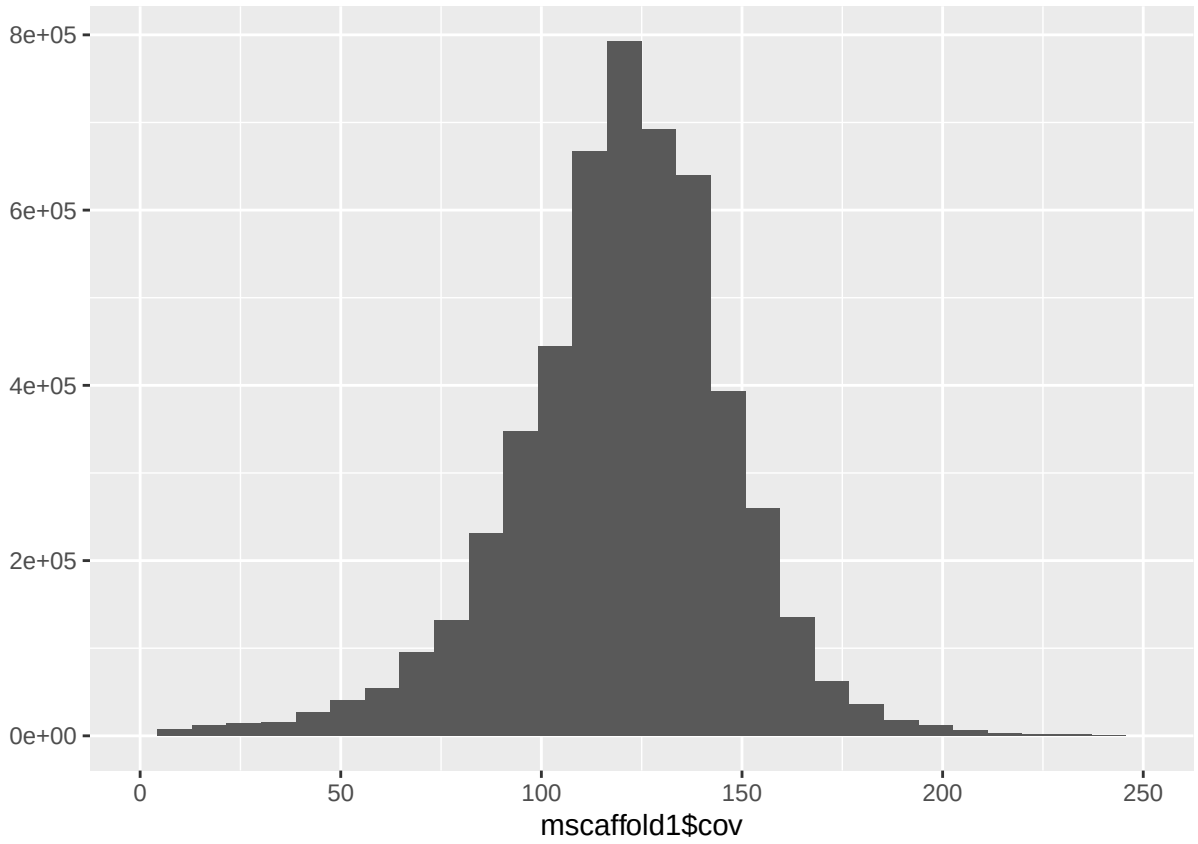
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Scaf123med <- data %>% group_by(scaffold) %>% summarize(Med = median(cov))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
mscaffold1 <- data %>% filter(scaffold == "CM021752.1")
qplot(mscaffold1$cov, xlim=c(0,250))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 5927 rows containing non-finite values (stat_bin).
## Warning: Removed 2 rows containing missing values (geom_bar).
```

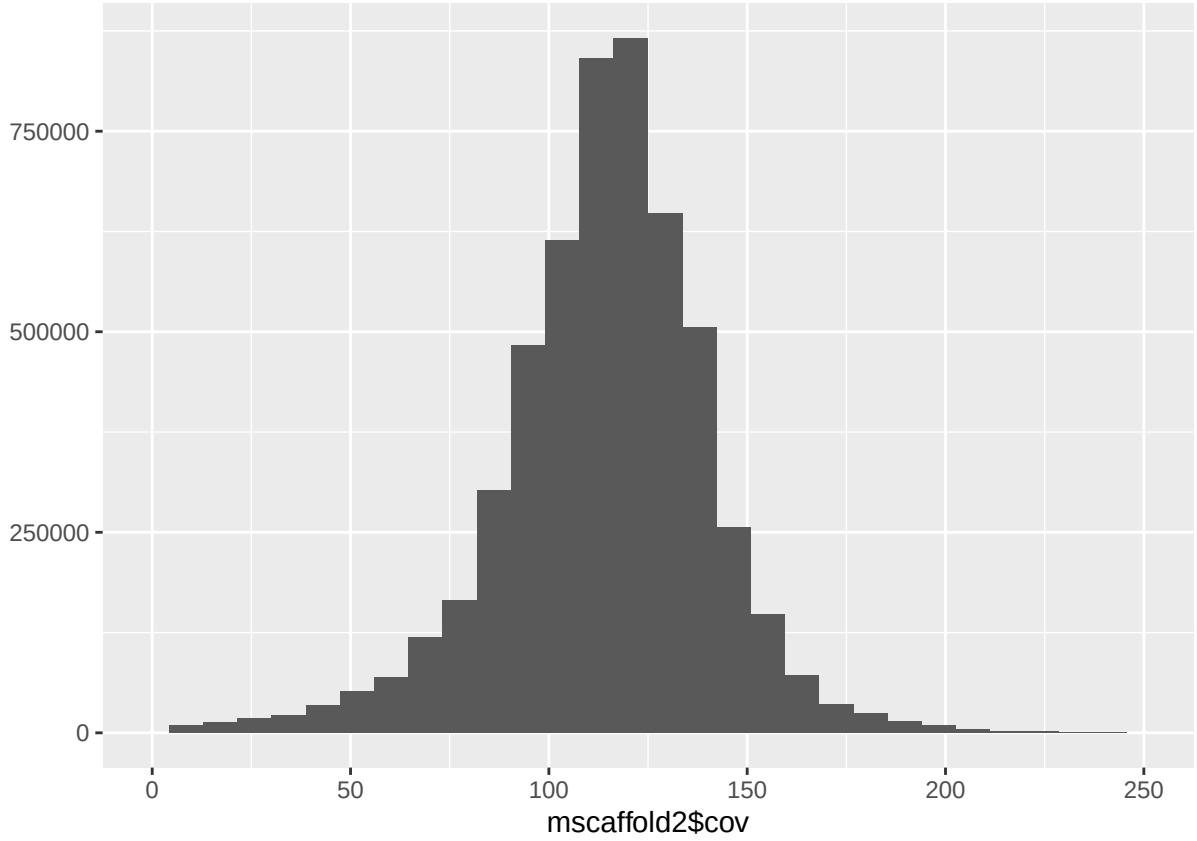



```
mscaffold2 <- data %>% filter(scaffold == "CM021757.1")
qplot(mscaffold2$cov, xlim=c(0,250))
```

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

```
## Warning: Removed 5702 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

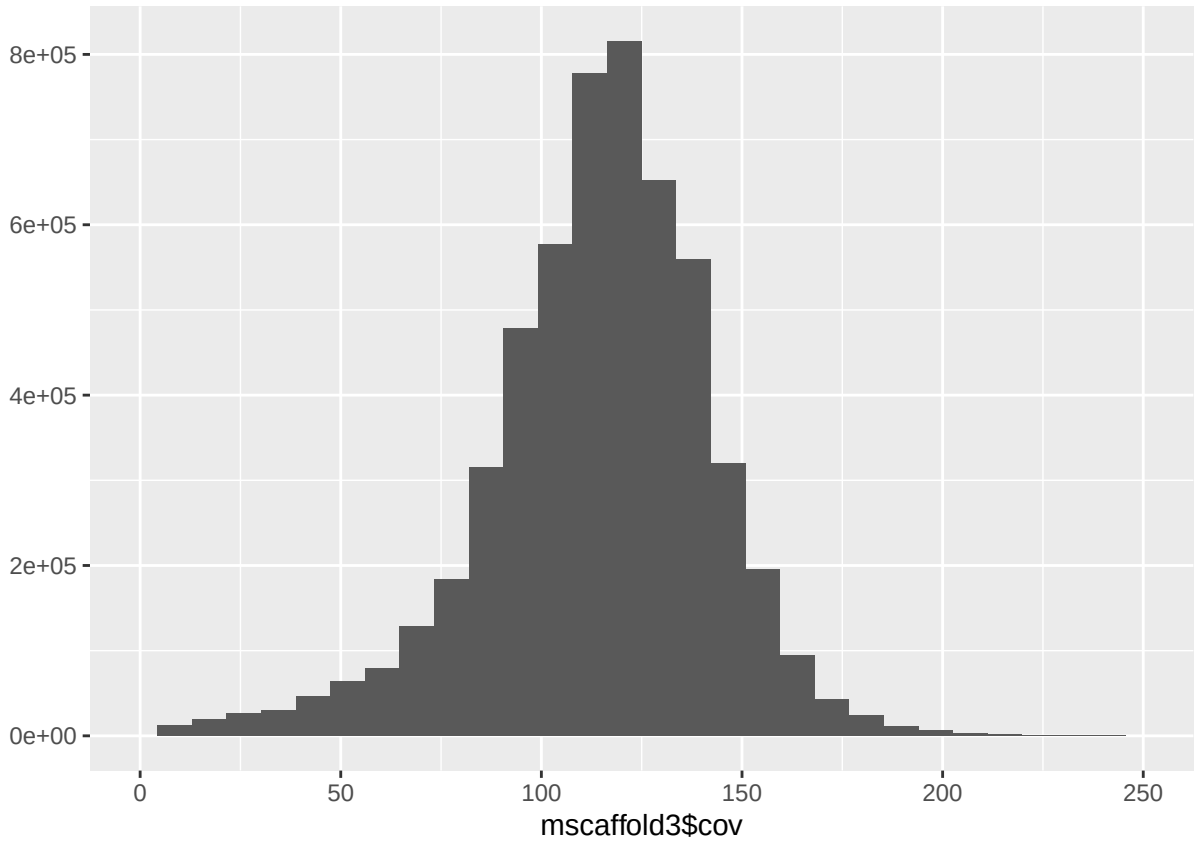



```
mscaffold3 <- data %>% filter(scaffold == "CM021758.1")
qplot(mscaffold3$cov, xlim=c(0,250))
```

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

```
## Warning: Removed 5375 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



#x-axis: coverage, y-axis: number of scaffolds. The mean and median coverage of each scaffold (mscaffold1$cov = CM021752.1, mscaffold2$cov= CM021757.1, mscaffold3$cov= CM021758.1) was found in males to see if numbers would differ, both mean and median values were similar.

FEMALES

```
data2 <- read_delim('fem_depth.txt', delim = '\t', col_names = F) %>%
  rename(scaffold = X1, pos = X2, cov = X3)
```

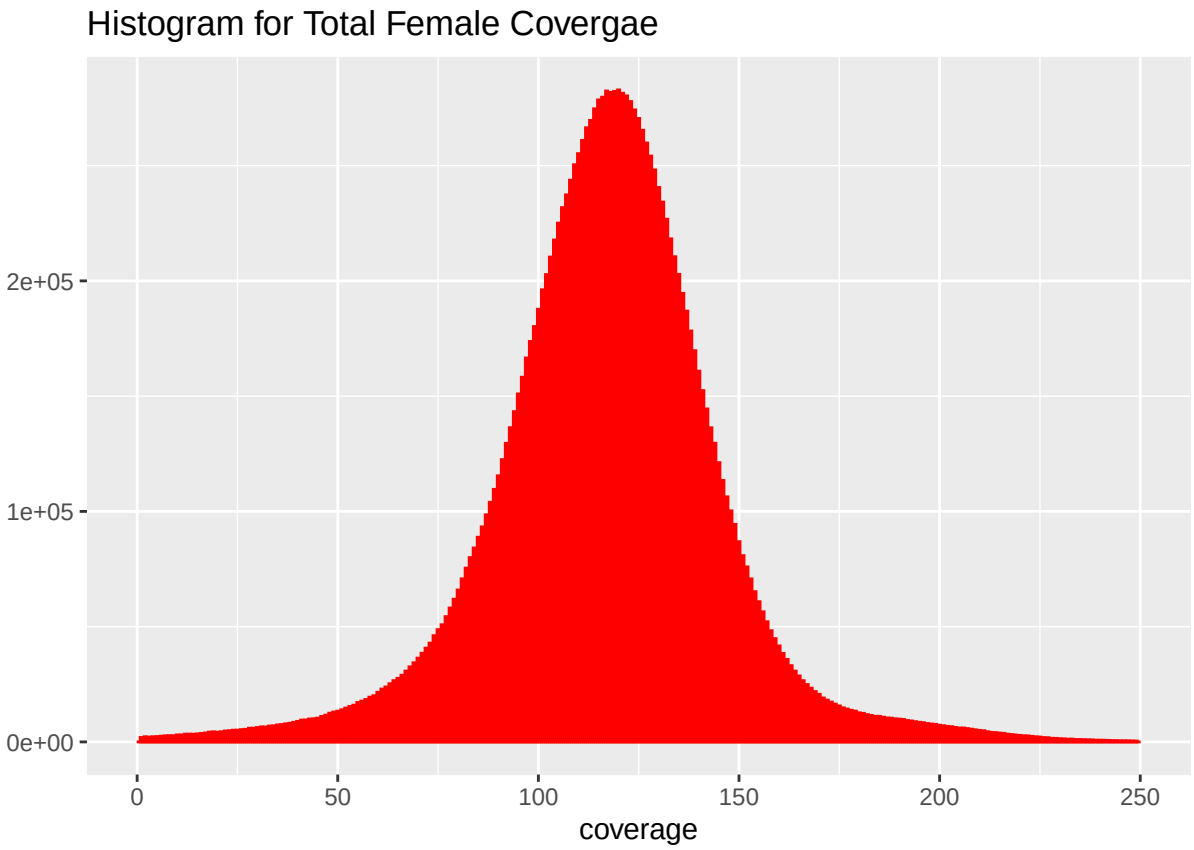
```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_double(),
##   X3 = col_double()
## )
```

```
mean.female <- mean(data2$cov)
median.femle <- median(data2$cov)
Totalcovgfem <- qplot(data2$cov,
                      geom="histogram",
                      binwidth = 0.5,
                      main = "Histogram for Total Female Covergae ",
                      xlab = "coverage",
                      fill=I("blue"),
                      col=I("red"),
                      alpha=I(.2),
```

```
                    xlim=c(0,250))
Totalcovgfem
```

```
## Warning: Removed 19719 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



#x-axis: coverage, y-axis: number of scaffolds. The total coverage was found for all females to ensure that important scaffolds were included.

Calculated the mean and median of the 3 autosomal scaffolds in females to ensure the numbers used for coverage and results obtained were reasonable and not largely skewed.

```
Scaf123fean <- data2 %>% group_by(scaffold) %>% summarize(Avgcov = mean(cov))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

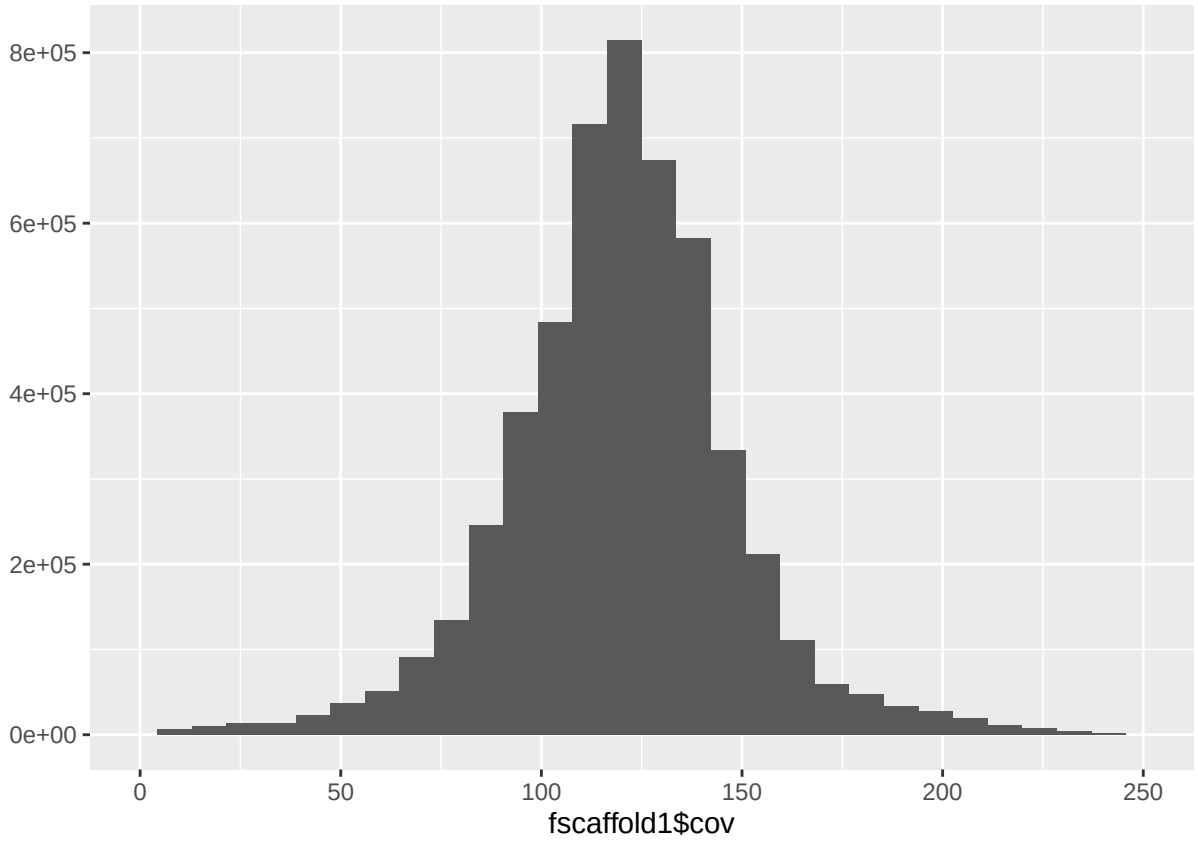
```
Scaf123fed <- data2 %>% group_by(scaffold) %>% summarize(Med = median(cov))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
fscaffold1 <- data2 %>% filter(scaffold == "CM021752.1")
qplot(fscaffold1$cov, xlim=c(0,250))
```

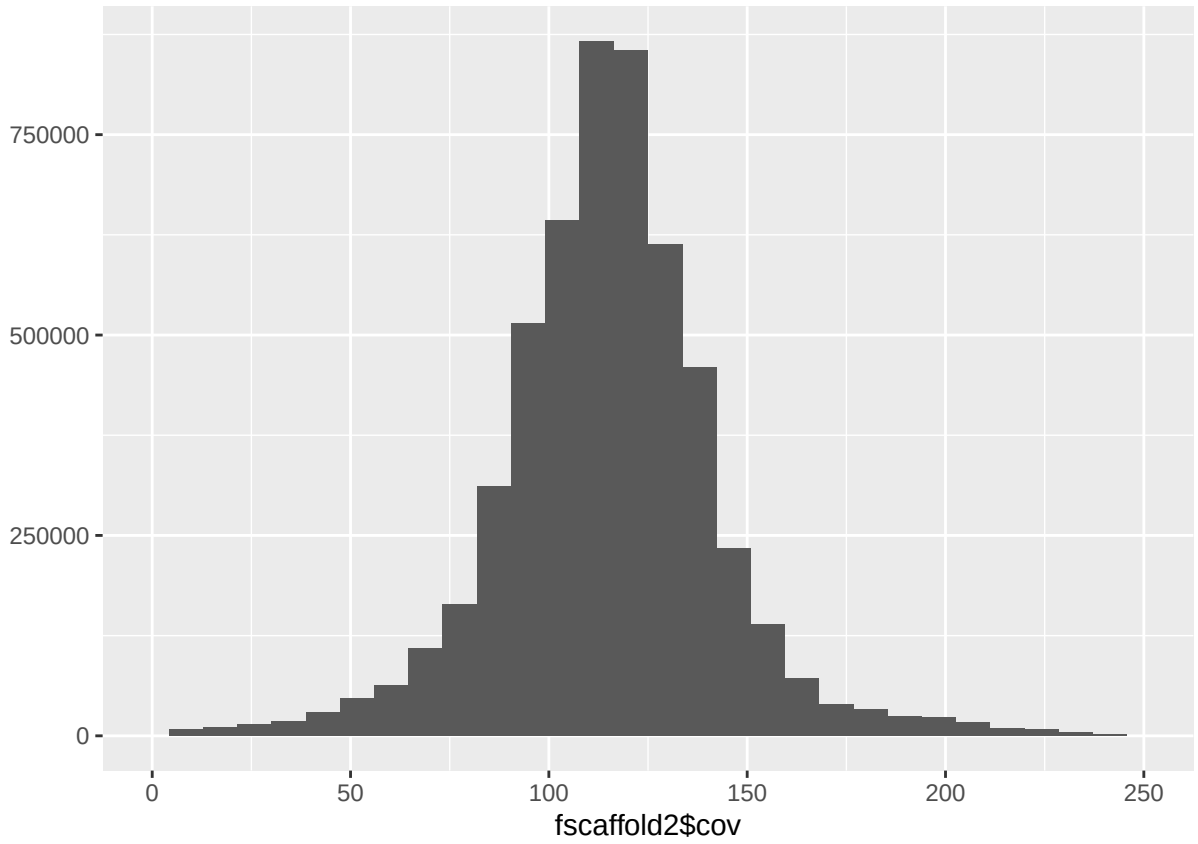
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 7424 rows containing non-finite values (stat_bin).
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
fscaffold2 <- data2 %>% filter(scaffold == "CM021757.1")
qplot(fscaffold2$cov, xlim=c(0,250))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 6434 rows containing non-finite values (stat_bin).
## Warning: Removed 2 rows containing missing values (geom_bar).
```

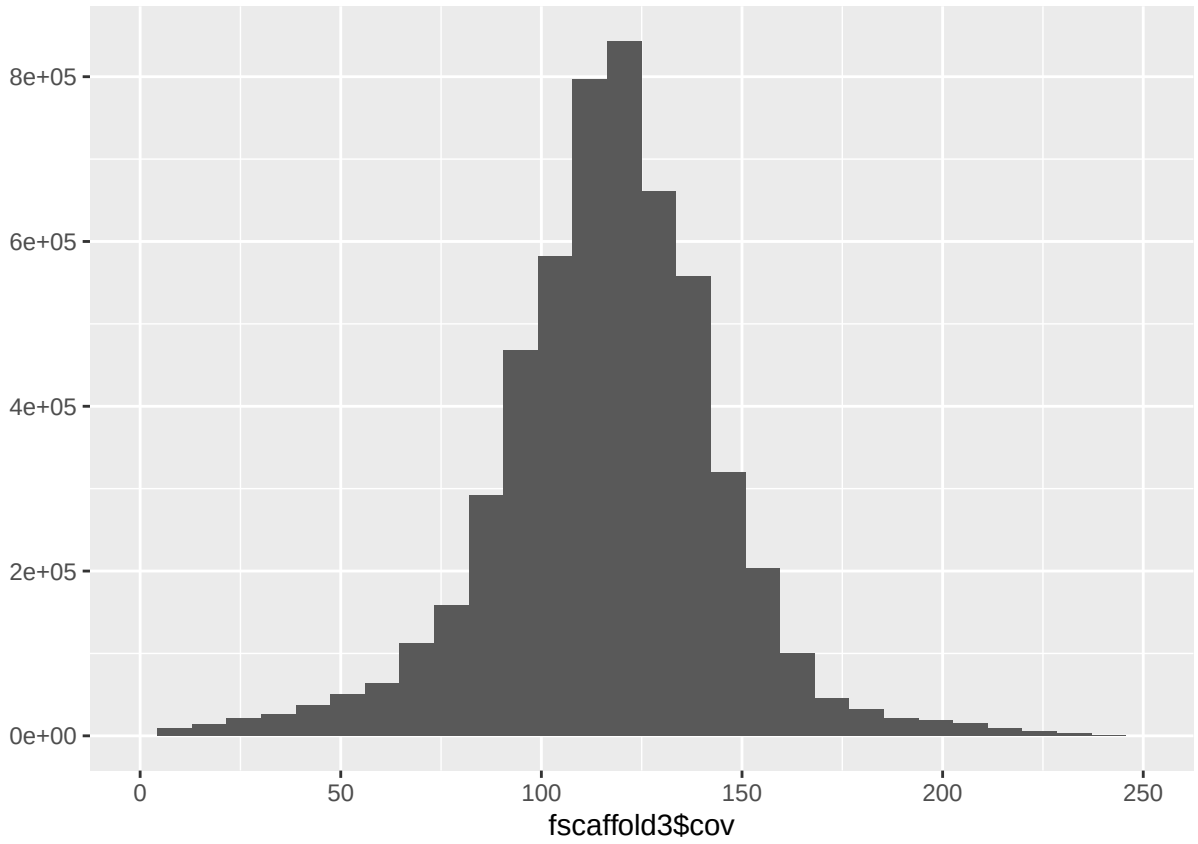



```
fscaffold3 <- data2 %>% filter(scaffold == "CM021758.1")
qplot(fscaffold3$cov, xlim=c(0,250))
```

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

```
## Warning: Removed 5861 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



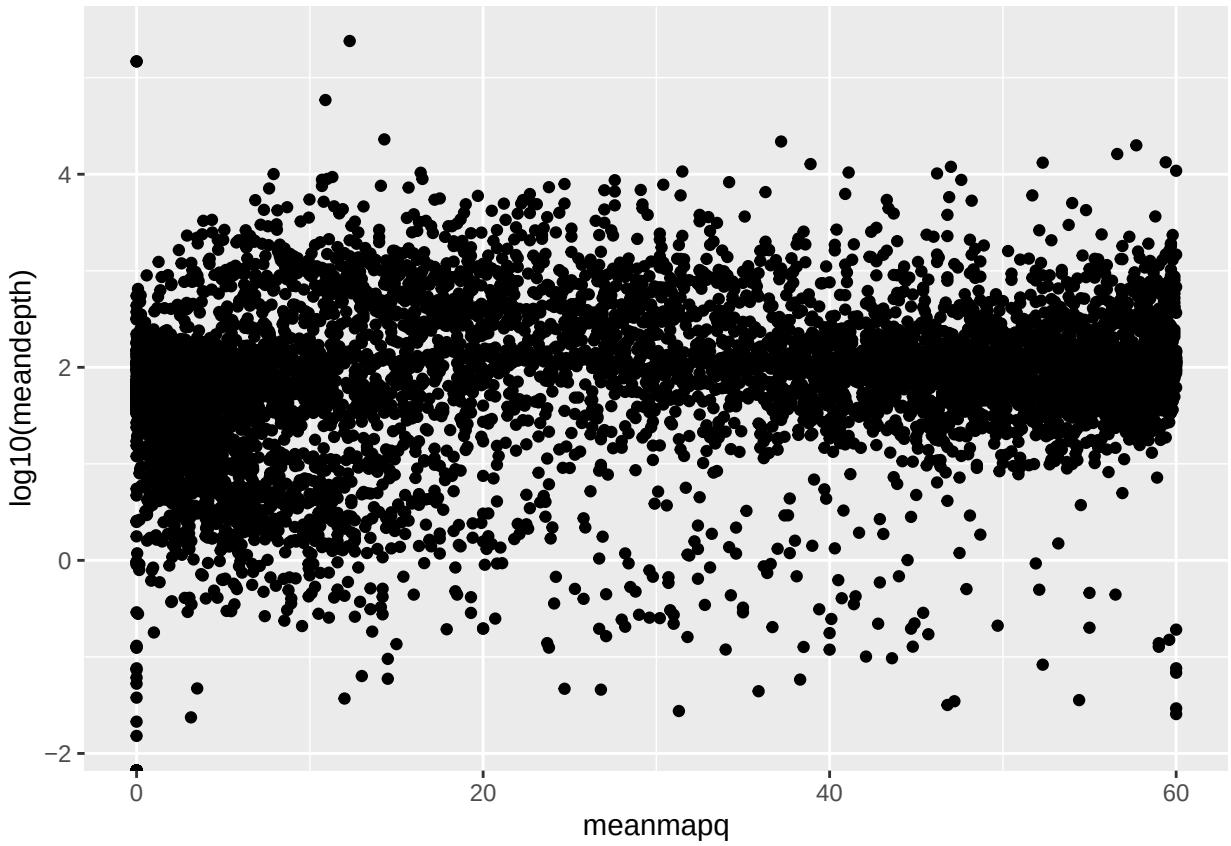
#x-axis: coverage, y-axis: number of scaffolds. The mean and median coverage of each scaffold (fscaffold1$cov = CM021752.1, fscaffold2$cov= CM021757.1, fscaffold3$cov= CM021758.1) was found in males to see if numbers would differ, both mean and median values were similar.

Comparing Columns in MALES

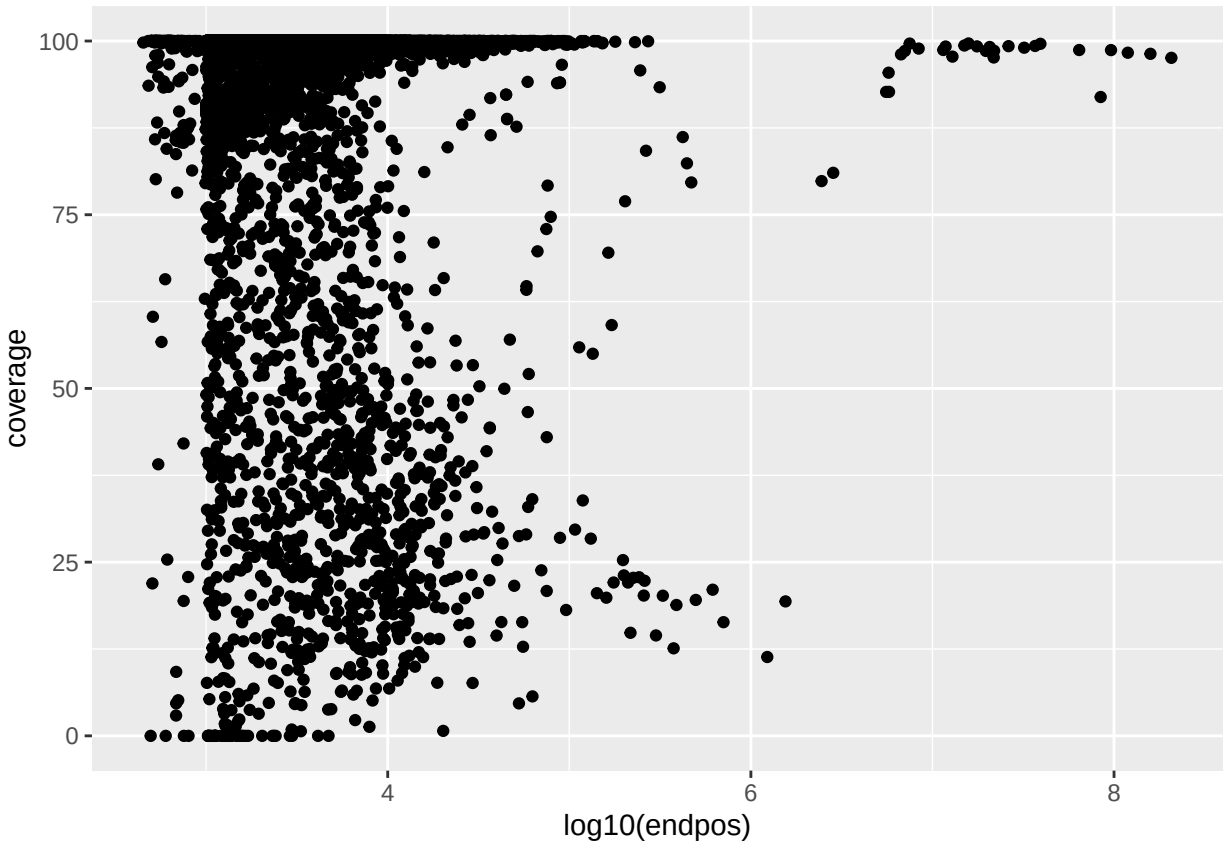
```
male.out <- read_delim('male.out', delim = '\t', col_names = T)
```

```
## Parsed with column specification:
## cols(
##   `#rname` = col_character(),
##   startpos = col_double(),
##   endpos = col_double(),
##   numreads = col_double(),
##   covbases = col_double(),
##   coverage = col_double(),
##   meandepth = col_double(),
##   meanbaseq = col_double(),
##   meanmapq = col_double()
## )
```

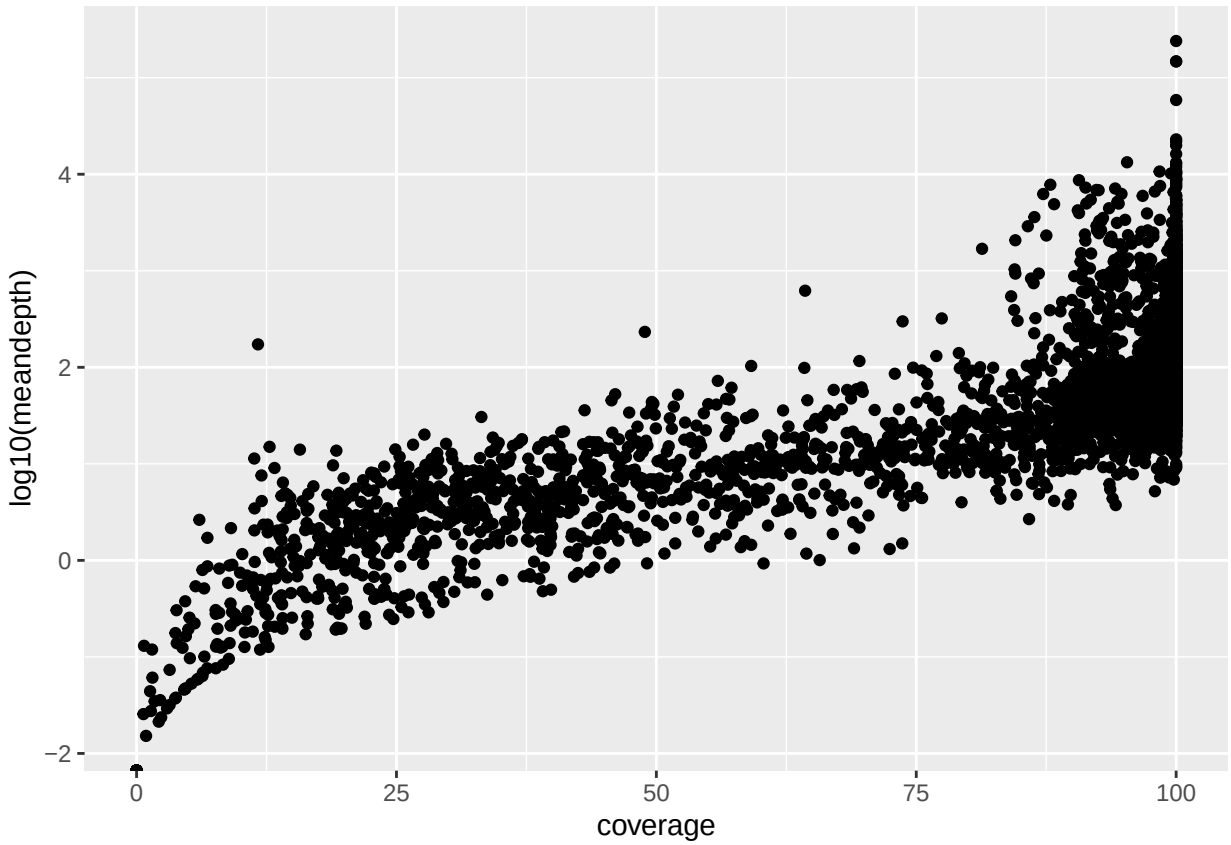
```
ggplot(male.out, aes(x=meanmapq, y=log10(meandepth))) + geom_point()
```



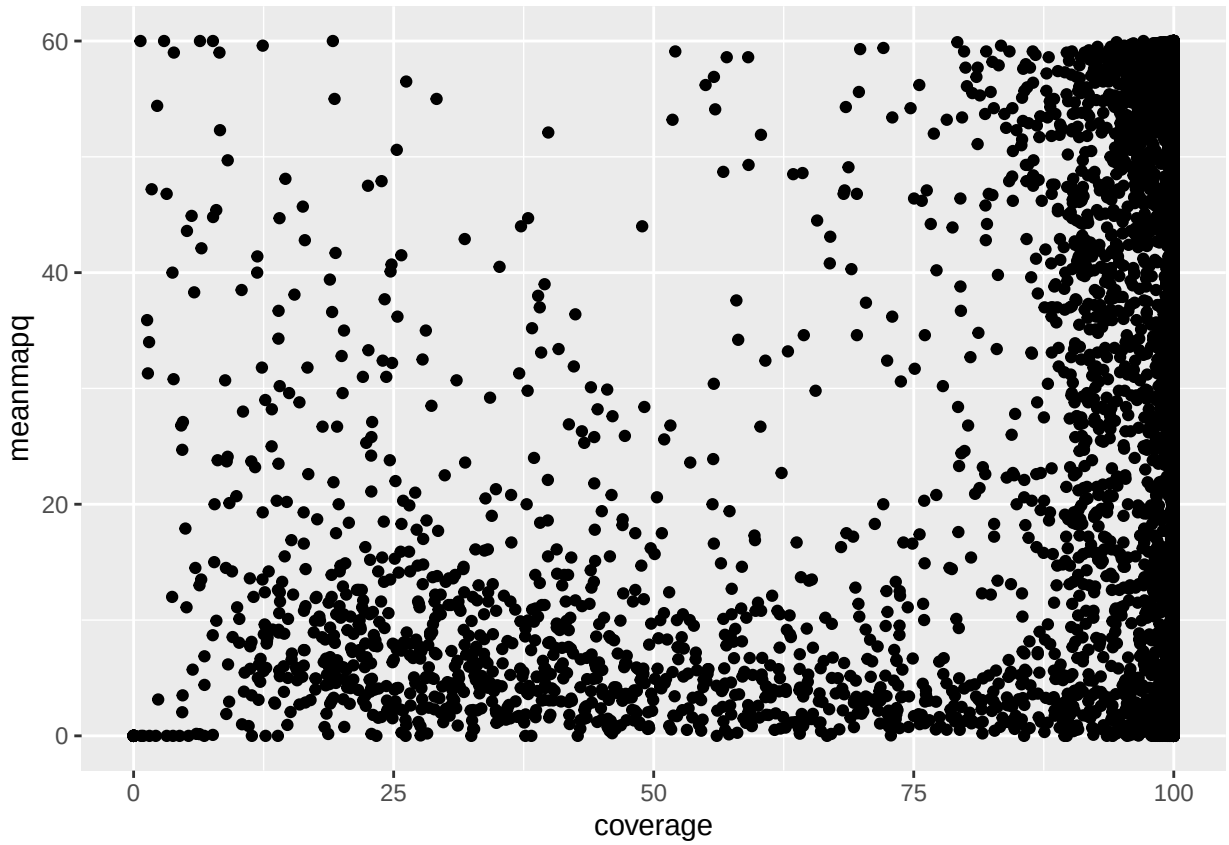

```
ggplot(male.out, aes(x=log10(endpos), y=coverage)) + geom_point()
```



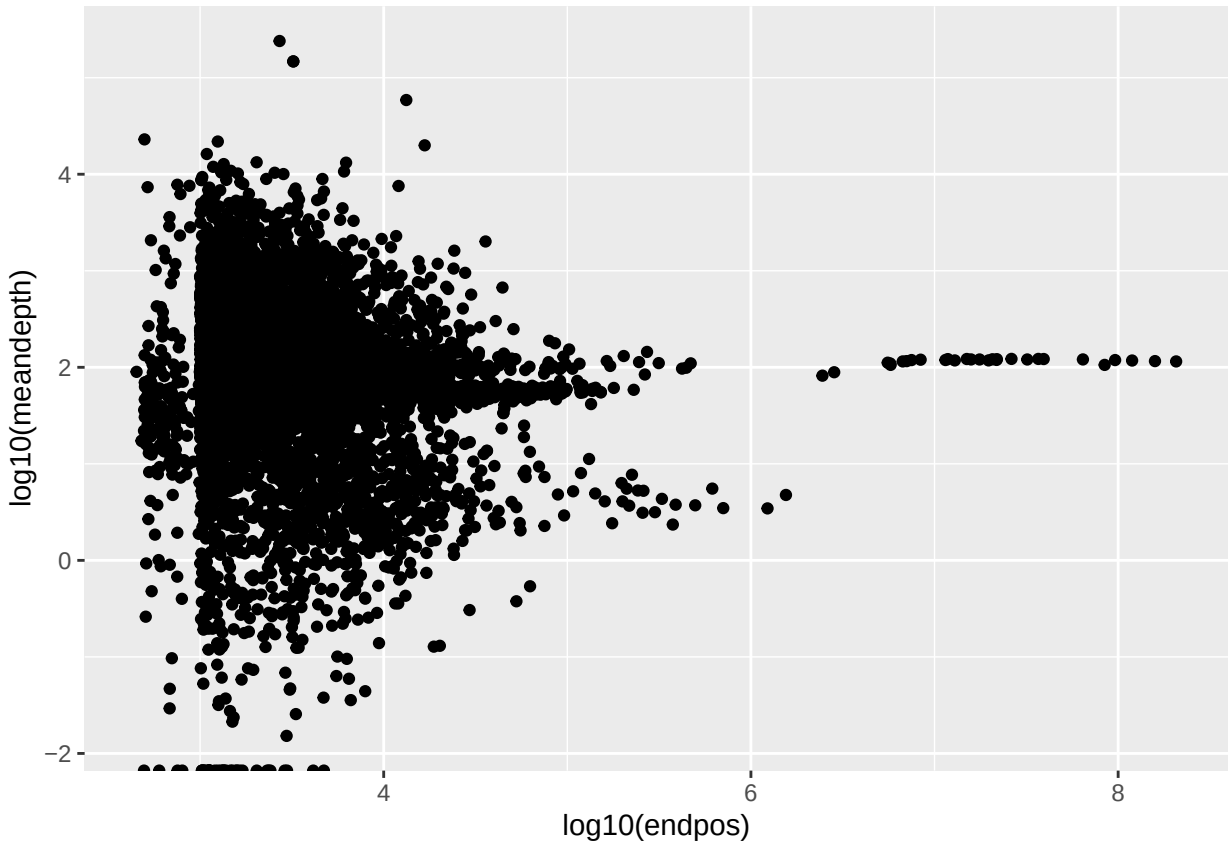

```
ggplot(male.out, aes(x=coverage, y=log10(meandepth))) + geom_point()
```




```
ggplot(male.out, aes(x=coverage, y=meanmapq)) + geom_point()
```




```
ggplot(male.out, aes(x=log10(endpos), y=log10(meandepth))) + geom_point()
```



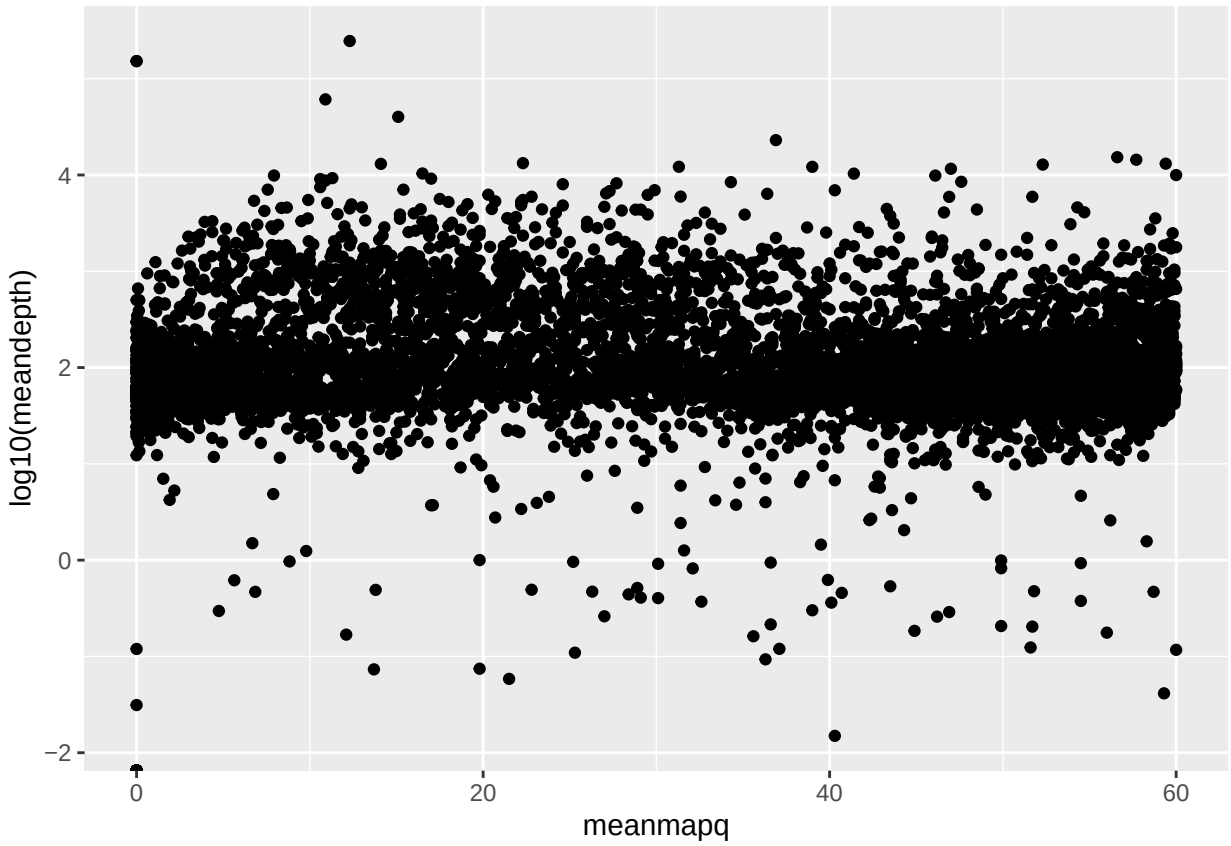
#X & Y axis (V3= end position, V6=coverage, V7= mean depth, V9=mean mapq) # Compared comlums to each other within males to see if correlation to each parameter matched hypotheses

Comparing Columns in FEMALES

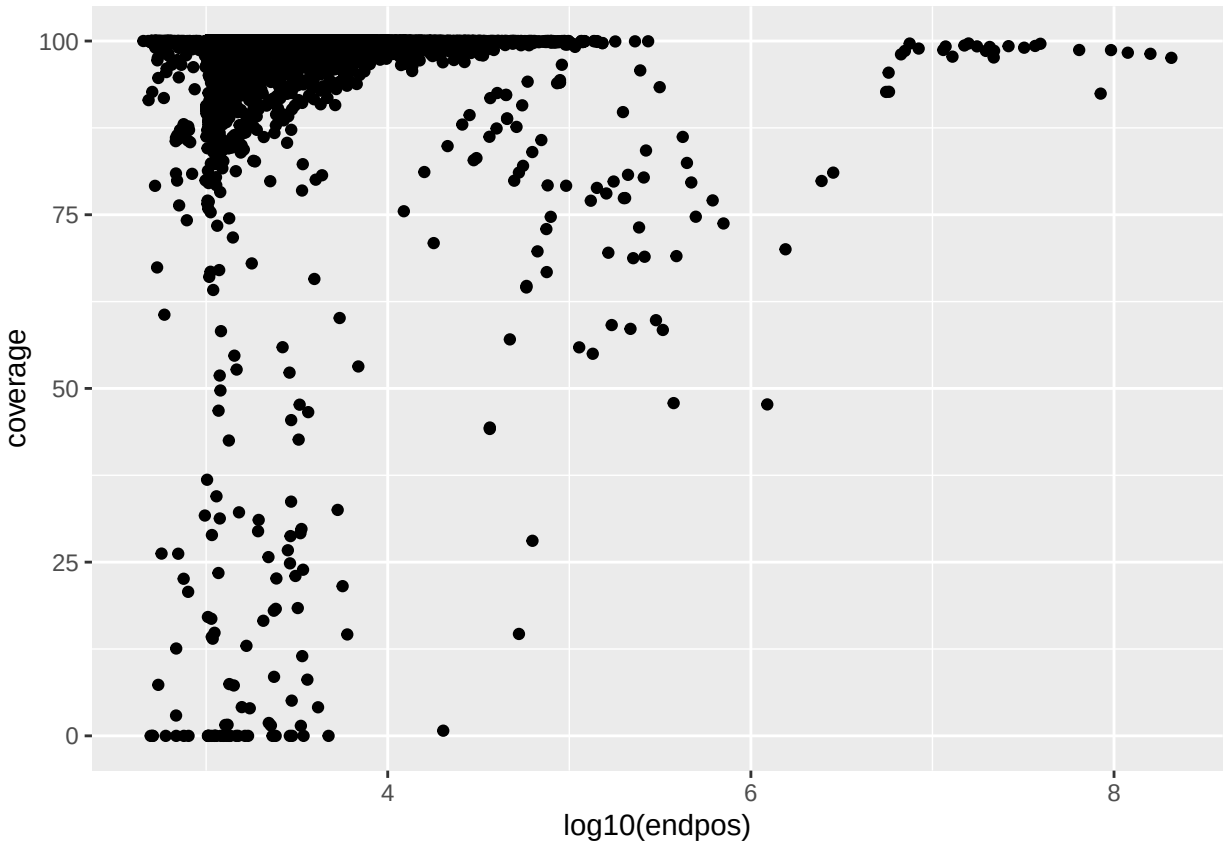
```
female.out <- read_delim('female.out', delim = '\t', col_names = T)
```

```
## Parsed with column specification:
## cols(
##   `#rname` = col_character(),
##   startpos = col_double(),
##   endpos = col_double(),
##   numreads = col_double(),
##   covbases = col_double(),
##   coverage = col_double(),
##   meandepth = col_double(),
##   meanbaseq = col_double(),
##   meanmapq = col_double()
## )
```

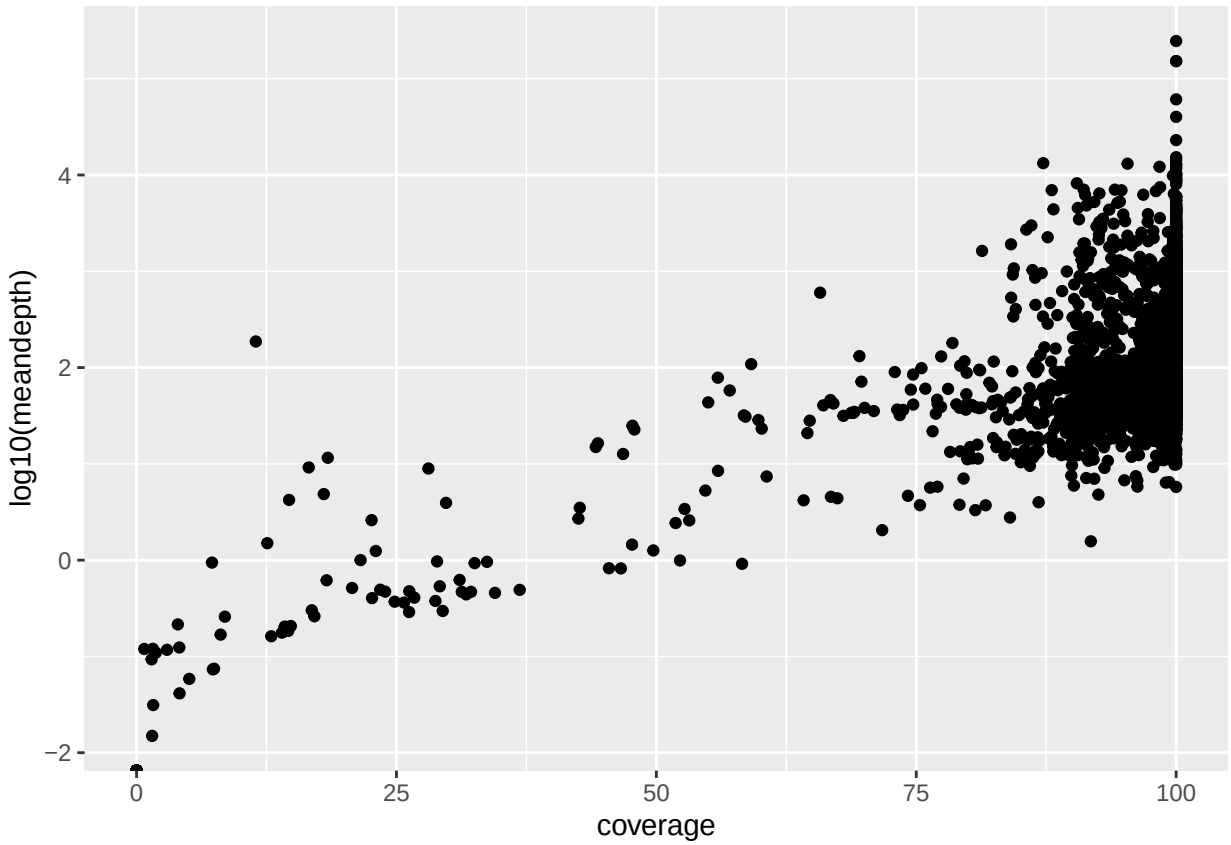
```
ggplot(female.out, aes(x=meanmapq, y=log10(meandepth))) + geom_point()
```



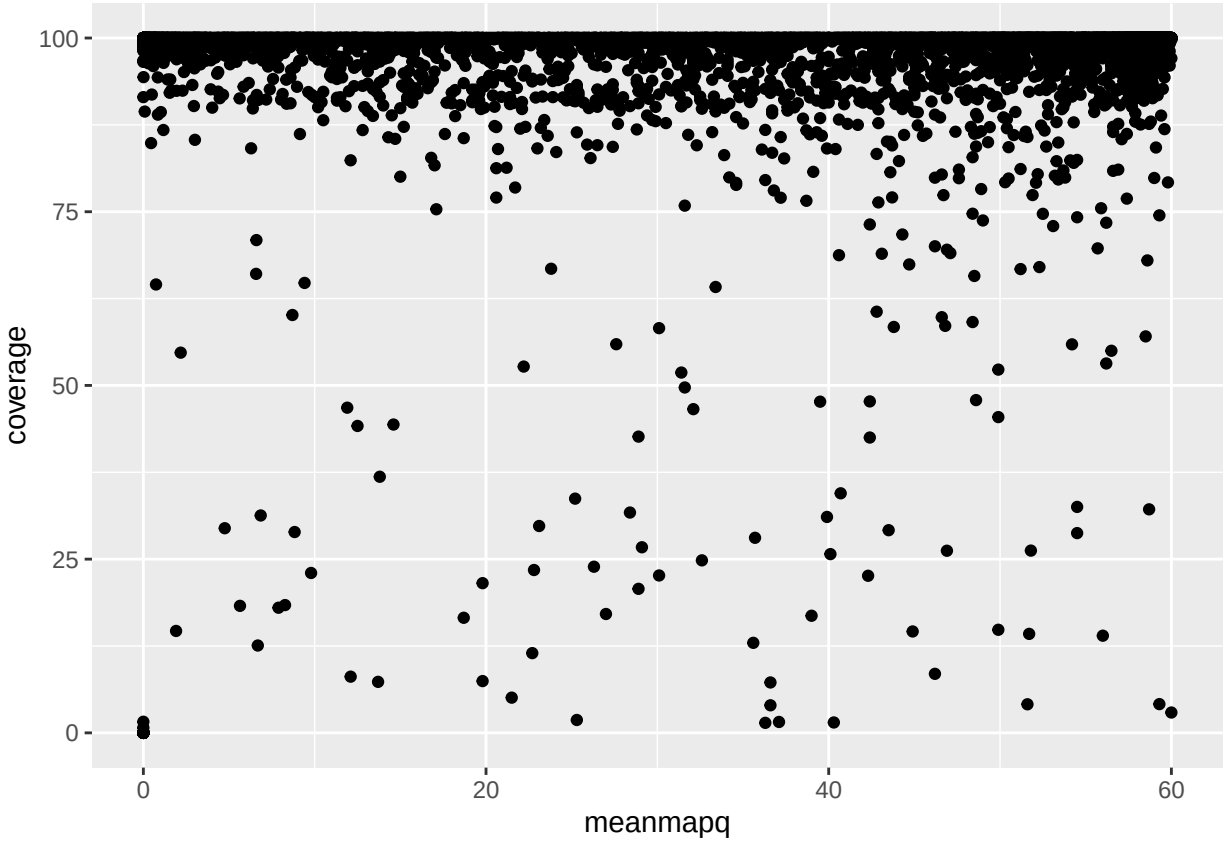

```
ggplot(female.out, aes(x=log10(endpos), y=coverage)) + geom_point()
```



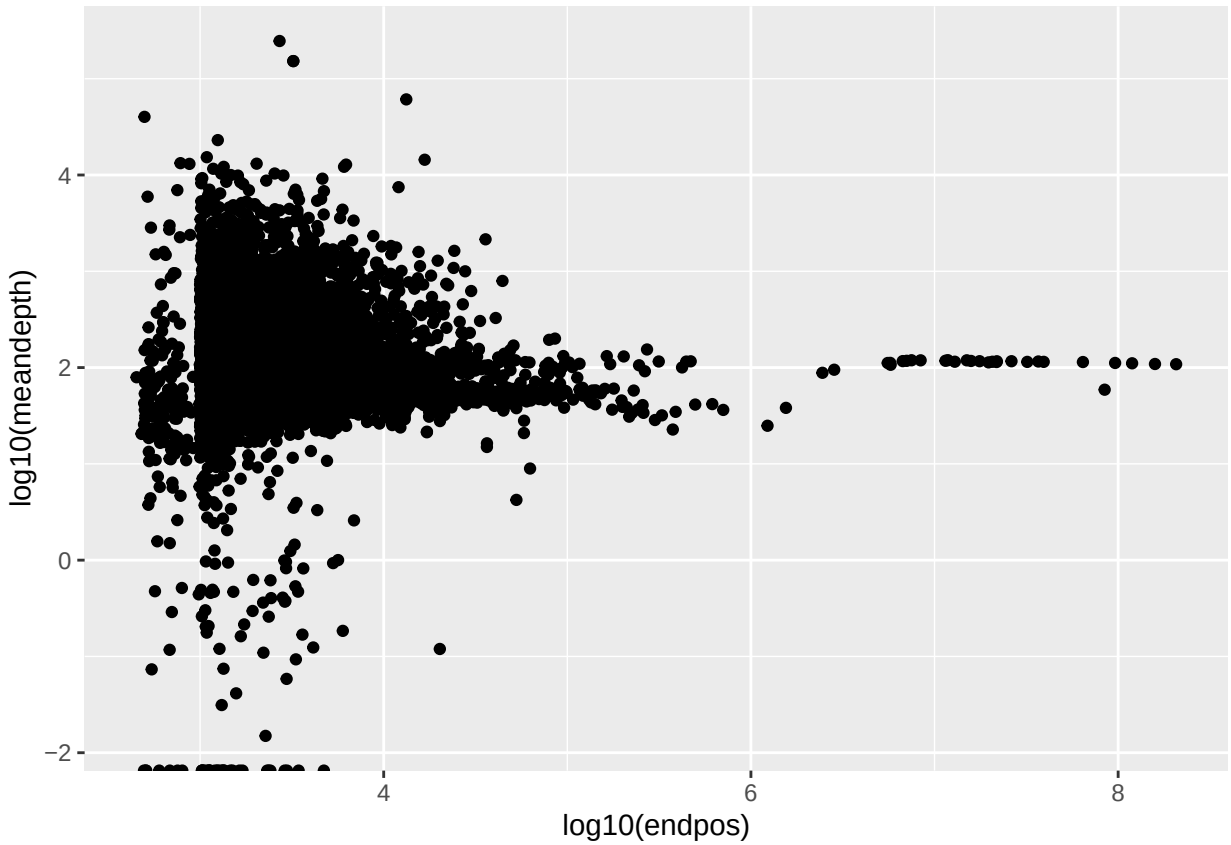

```
ggplot(female.out, aes(x=coverage, y=log10(meandepth))) + geom_point()
```




```
ggplot(female.out, aes(x=meanmapq, y=coverage)) + geom_point()
```




```
ggplot(female.out, aes(x=log10(endpos), y=log10(meandepth))) + geom_point()
```



#X & Y axis (V3= end position, V6=coverage, V7= mean depth, V9=mean mapq) # Compared comlums to each other within males to see if correlation to each parameter matched hypotheses

#Calculating Normalized Depth for males and females which is a necessary step to compare data between two samples

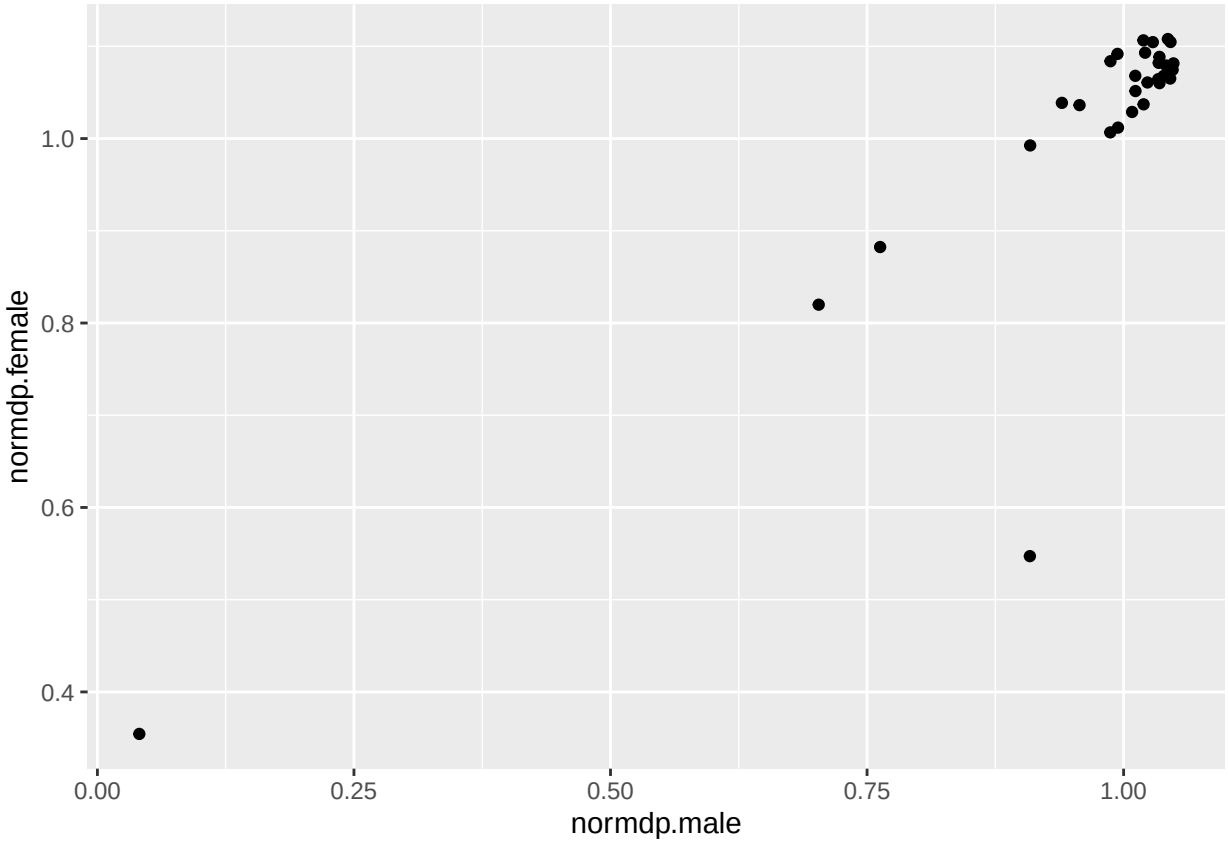
```
male_aut_cov <- male.out %>% filter(endpos > 1e6) %>% mutate(weight = endpos * meandepth) %>% summarize
male_norm <- male.out %>% mutate(normdp = meandepth/male_aut_cov)

female.out %>% filter(endpos > 1e6) %>% arrange(meandepth)
```

```
## # A tibble: 31 x 9
##    `#rname` startpos endpos numreads covbases coverage meandepth meanbaseq
##    <chr>       <dbl>  <dbl>    <dbl>    <dbl>    <dbl>     <dbl>     <dbl>
##  1 MU01470~        1 1.23e6   3.10e5   5.87e5     47.7      24.9      34.9
##  2 CM02176~        1 1.55e6   6.02e5   1.09e6     70.0      38.2      34.9
##  3 CM02176~        1 8.44e7   5.01e7   7.80e7     92.4      58.9      35
##  4 CM02176~        1 2.45e6   2.20e6   1.96e6     79.9      88.3      34.3
##  5 CM02175~        1 2.84e6   2.74e6   2.30e6     81.1      95.0      34.4
##  6 CM02175~        1 5.77e6   6.23e6   5.34e6     92.7     107.       34.7
##  7 CM02173~        1 2.07e8   2.26e8   2.02e8     97.6     108.       35
##  8 CM02173~        1 1.59e8   1.74e8   1.56e8     98.2     109.       35
##  9 CM02173~        1 1.19e8   1.33e8   1.17e8     98.3     111.       35
## 10 CM02175~        1 5.56e6   6.25e6   5.15e6     92.7     112.       34.8
## # ... with 21 more rows, and 1 more variable: meanmapq <dbl>
```

```
female_aut_cov <- female.out %>% filter(endpos > 1e6, meandepth > 30) %>% mutate(weight = endpos * meanc
female_norm <- female.out %>% mutate(normdp = meandepth/female_aut_cov)
```

#MERGED normdp of males and females in order to sort the scaffolds into appropriate ranges of chromosome Z, W, or autosome

```
merged_norm <- full_join(male_norm, female_norm, by=c("#rname" = "#rname", "endpos" = "endpos"), suffix=

merged_norm %>% filter(endpos > 1e6, coverage.female > 50, meanmapq.female > 20) %>% ggplot(aes(x=normdp
```

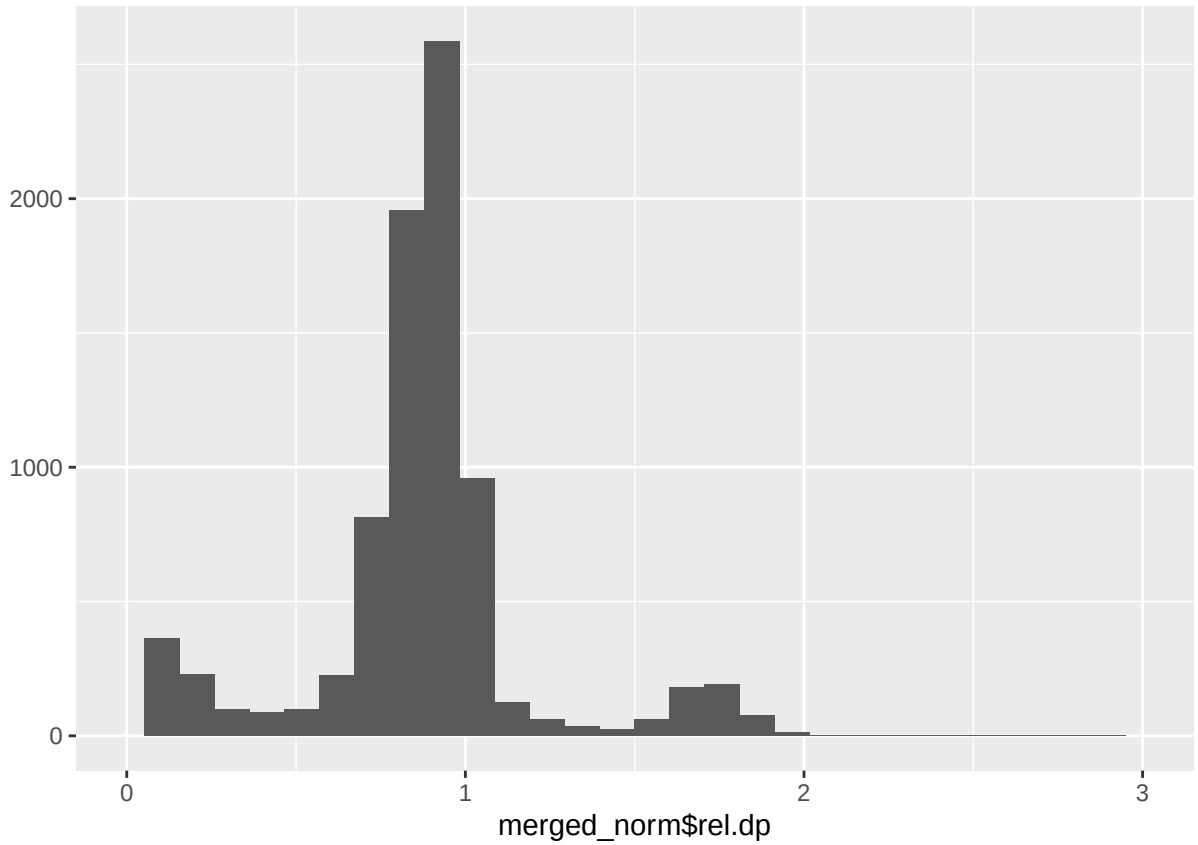


x-axis: normalized depth of males, y-axis: normalized depth of females. This graph shows that the normalized depths are directly correlated with eahc other.

Created a relative depth column for ease of distribution and graphs

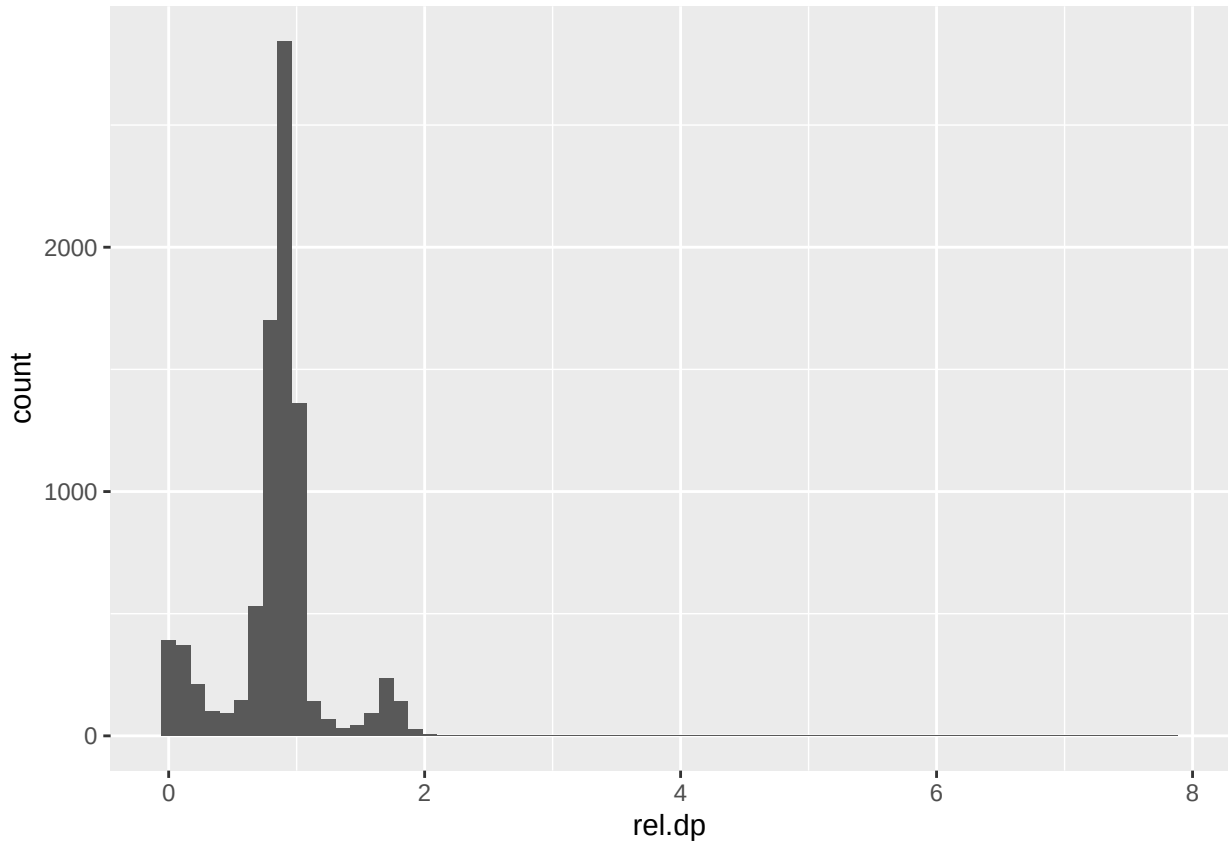
```
qplot(merged_norm$rel.dp, xlim=c(0,3))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 43 rows containing non-finite values (stat_bin).
## Warning: Removed 2 rows containing missing values (geom_bar).
```

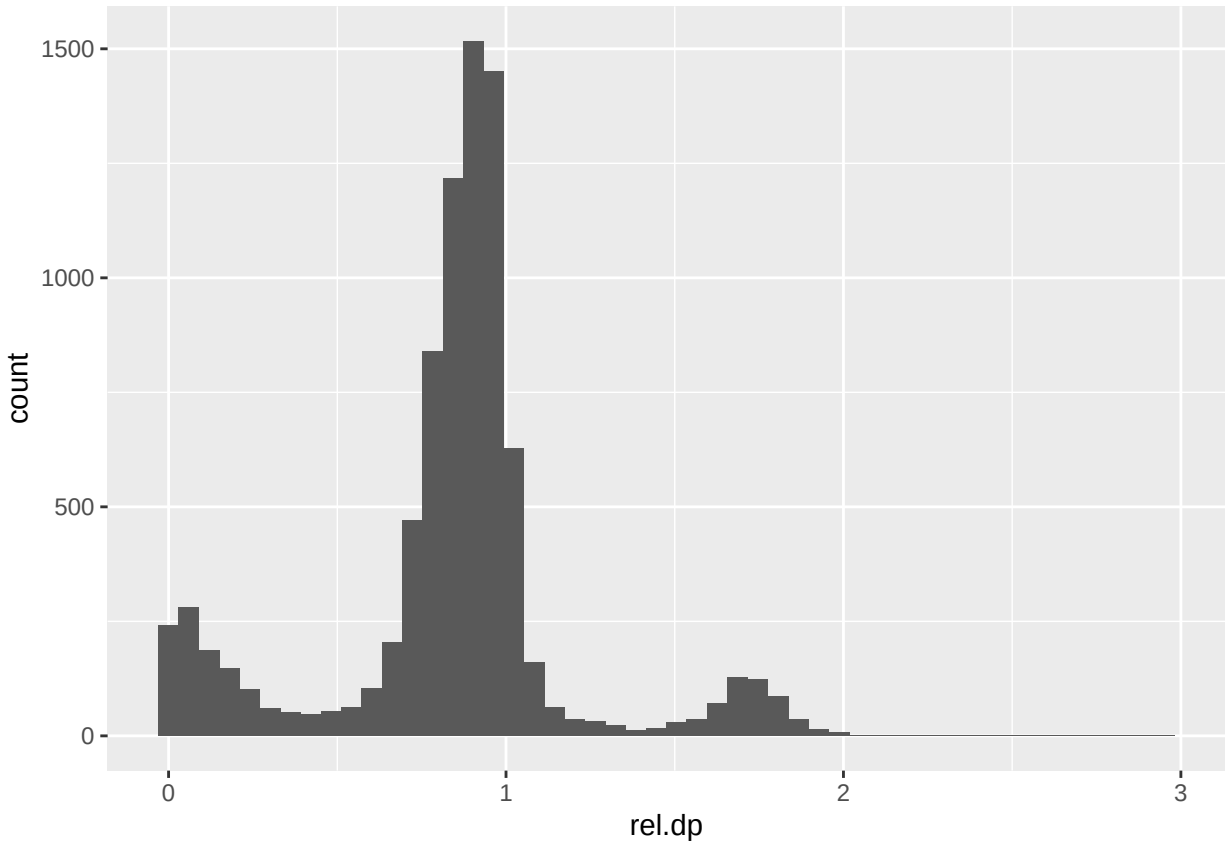



```
ggplot(merged_norm, aes(x=rel.dp)) + geom_histogram(bins=70)
```

```
## Warning: Removed 34 rows containing non-finite values (stat_bin).
```




```
merged_norm %>% filter(rel.dp < 3) %>% ggplot(aes(x=rel.dp)) + geom_histogram(bins=50)
```



x-axis: relative depth, y-axis: number of scaffolds. This graph shows a graphcal representation of the distribution of scaffolds that are sorted by a relative depth value.

Filtering Relative Depth Plot by sorting out scaffolds by their relative depth values to label them as W, Z, or autosomal.

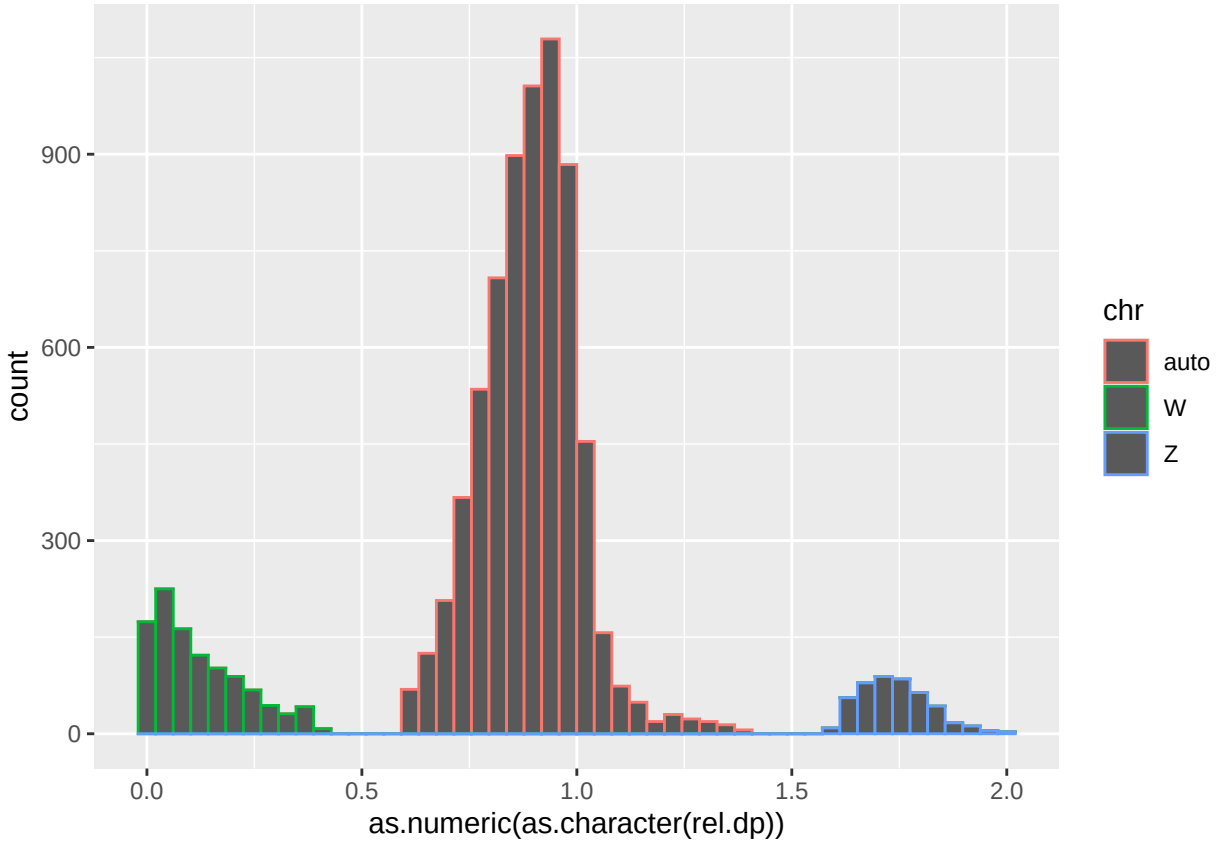
W chromosomes = 0 - .4, autosomes = .6 - 1.4, Z chromosomes = 1.6 - 2

```
scaffold_W <- merged_norm %>% select('#rname', rel.dp, coverage.female, coverage.male) %>% filter((rel.

scaffold_auto <- merged_norm %>% select('#rname', rel.dp, coverage.female, coverage.male) %>% filter((r

scaffold_Z <- merged_norm %>% select('#rname', rel.dp, coverage.female, coverage.male) %>% filter((rel.

scaff_undetermined <- merged_norm %>% select('#rname', rel.dp, coverage.female, coverage.male) %>% filt
```

Combining Z, W, and Auto chromosomes

```
all_rel_dp <- rbind(scaffold_Z, scaffold_W, scaffold_auto)
W <- scaffold_W %>% mutate(chr = "W")
auto <- scaffold_auto %>% mutate(chr = "auto")
Z <- scaffold_Z %>% mutate(chr = "Z")
all_rel_dp <- rbind(Z, W, auto)
```

```
ggplot(all_rel_dp, aes(x=as.numeric(as.character(rel.dp)), col=chr)) + geom_histogram(bins=50)
```

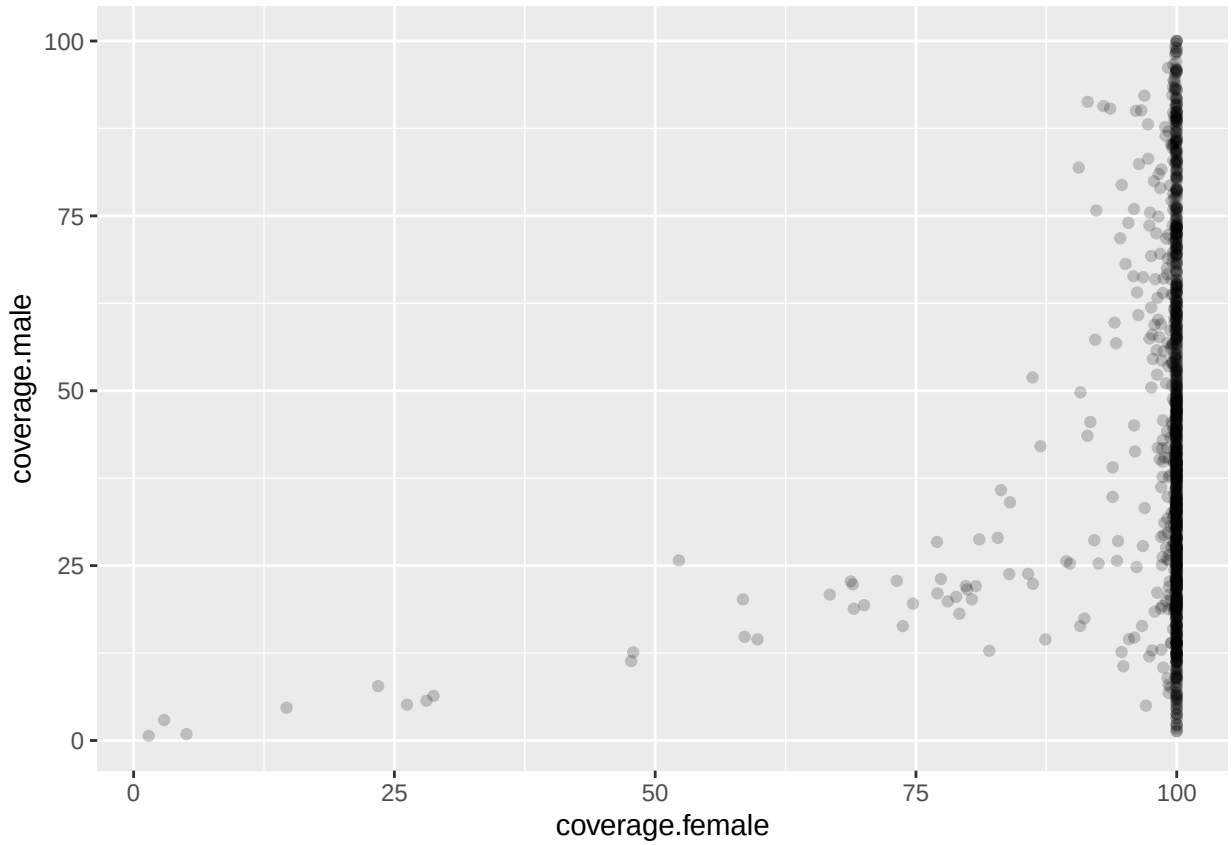


#x-axis: relative depth, y-axis: number of scaffolds. This graph shows a more defined representation of the scaffolds sorted by a relative depth value and lables them respectively as W-linked, autosomal, and Z-linked.

Challenging Hypotheses

#Examining the coverage of females versus males in relation to the W chromosome

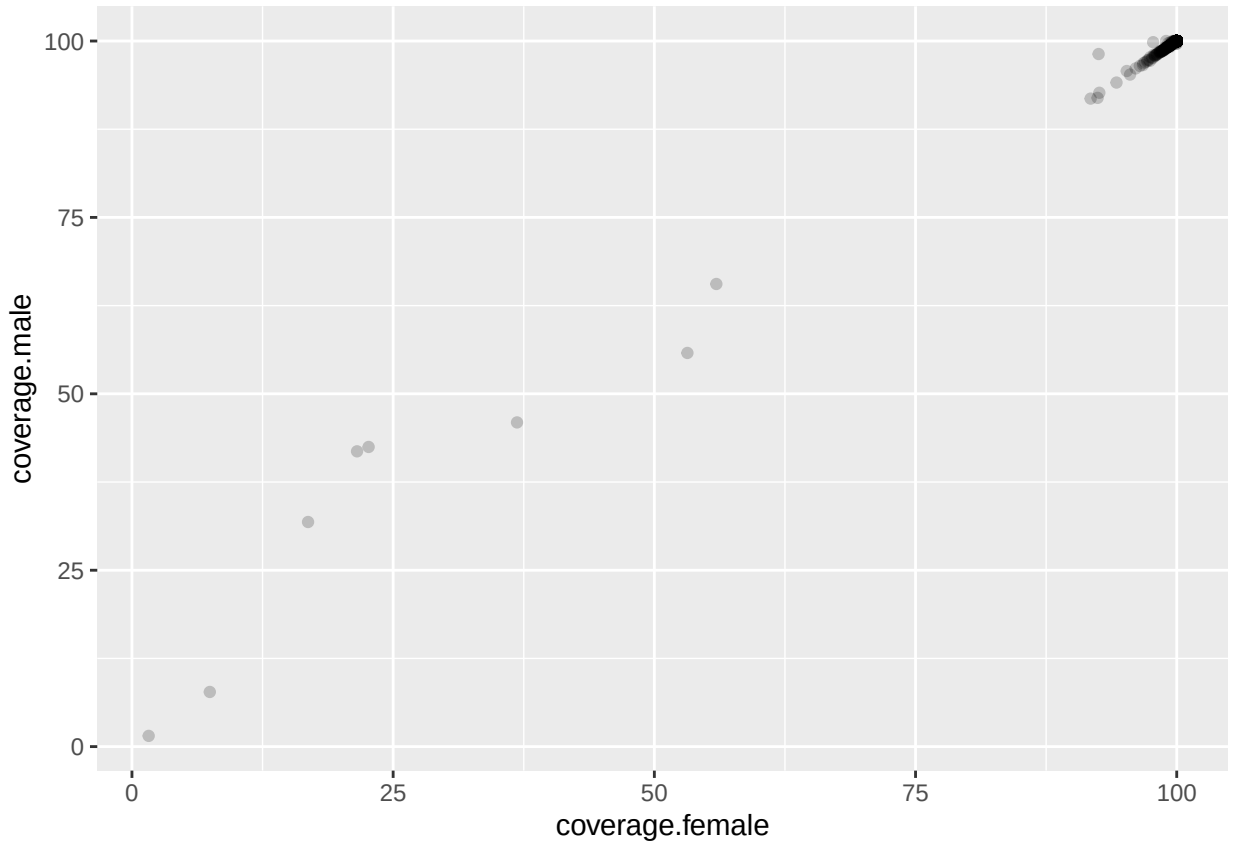
```
femcov_v_malecovW <- all_rel_dp %>% filter(chr == "W") %>% ggplot(aes(x=coverage.female, y=coverage.mal
femcov_v_malecovW
```



x-axis: coverage of females, y-axis: coverage of males # This graph shows that the females have very high coverage, which entails that maybe some W chromosomes may be classified in error since they have the same trend as autosomes and have higher than expected male coverage, so they could share some homologous characteristics with the Z chromosome.

#Examining the coverage of females versus males in relation to the Z chromosome

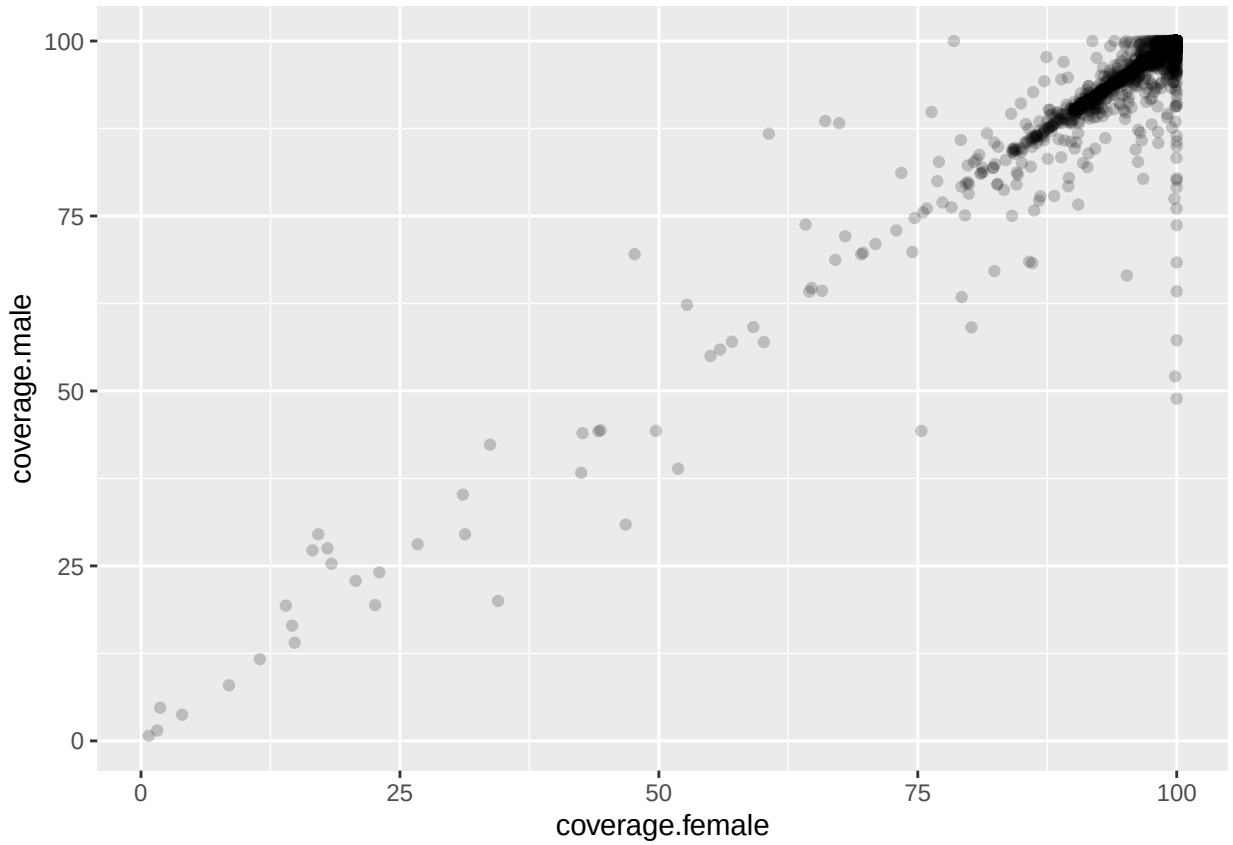
```
femcov_v_malecovZ <- all_rel_dp %>% filter(chr == "Z") %>% ggplot(aes(x=coverage.female, y=coverage.mal
femcov_v_malecovZ
```



x-axis: coverage of females, y-axis: coverage of males #This graph shows that the Z chromosome has a very high amount of male coverage, which is expected for a Z linked chromosome.

#Examining the coverage of females versus males in relation to the autosomes

```
femcov_v_malecov_auto <- all_rel_dp %>% filter(chr == "auto") %>% ggplot(aes(x=coverage.female, y=covera
femcov_v_malecov_auto
```



x-axis: coverage of females, y-axis: coverage of males # This graph shows exactly what we expected, that the coverage of autosomes have a very high male coverage.