R_assignment

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
## -- 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   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- 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(tidyr)
```

Data Inspection

```
fang<-read.delim("fang_et_al_genotypes.txt")
fangdim=dim(fang) #return number of rows and columns
fang_info=(file.info('fang_et_al_genotypes.txt'))
```

Number of rows=2782 Number of columns=986 Size in Bytes=11054722

```
snp<-read.delim("snp_position.txt")
snpdim=dim(snp) #return number of rows and columns
snp_info=(file.info('snp_position.txt'))
```

Number of rows=983 Number of columns=15 Size in Bytes=83747

Data Processing

Use the transposed data before joining

```
fang_t<-read.delim("transposed_genotypes.txt")
```

From the genotype data, we remove the rows containing Sample_ID and JG_OTU, and arrange the table based on the GROUP row as header to facilitate merging and sorting

```
fang_t <- as.data.frame(fang_t)
new_fang<-fang_t[-c(0,1),]
colnames(new_fang)<-as.character(new_fang[1,])
new_fang<-new_fang[-c(1),]
```

Joining the genotype data with the SNP data

```
merged<-merge(snp,new_fang, by.x="SNP_ID",by.y="Group", all=TRUE  )
```

Removing columns other than SNP_ID, Chromosome and Position

```
final <-merged[-c(2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15)]
```

Maize Dataset

Find columns containing "ZMMIL", "ZMMLR", and "ZMMMR" and remove the rest

```
maize<-final[c(1,2,3,1213:2468, 2469:2495, 2496:2785)]
maize<-as.data.frame(maize)
```

We have all maize data now.

To generate 10 files with SNPs ordered on increasing position values, we shall use the order() command First replace all missing data with ?,and split the large file based on each chromosome

```
maize_inc=maize
maize_inc[maize_inc=="?/?"]<-"?"
```

```
inc_chr <- split(maize_inc, maize_inc$Chromosome)
```

Now sorting each list based on increasing position values

To generate 10 files with SNPs ordered on decreasing position values, we shall use the order() command First replace all missing data with - symbol,and split the large file based on each chromosome.

```
maize_dec=maize
maize_dec[maize_dec=="?/?"]<-"-"
```

```
dec_chr<- split(maize_dec, maize_dec$Chromosome)
```

Thus we have the required 20 files. A similar procedure is repeated for the teosinte data. ## Teosinte Dataset

```
teosinte=final[c(1, 2, 3, 77:976, 977:1010, 1166:1206)]
teosinte<-as.data.frame(teosinte)
```

Let's generate 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?

Next step is 10 files (1 for each chromosome) with SNPs ordered based on decreasing position values and with missing data encoded by this symbol: -

Thus we have all required files.

Part II Visualization

Task-1: Plotting total number of SNPs per chromosome

```
library(dplyr)
```

```
maize_snp_count <- aggregate(SNP_ID ~ Chromosome, data = maize, FUN = length)
colnames(maize_snp_count)[2] <- "SNP_Count"
maize_snp_count$Group <- "Maize"
```

```
teosinte_snp_count <- aggregate(SNP_ID ~ Chromosome, data = teosinte, FUN = length)
colnames(teosinte_snp_count)[2] <- "SNP_Count"
teosinte_snp_count$Group <- "Teosinte"
```

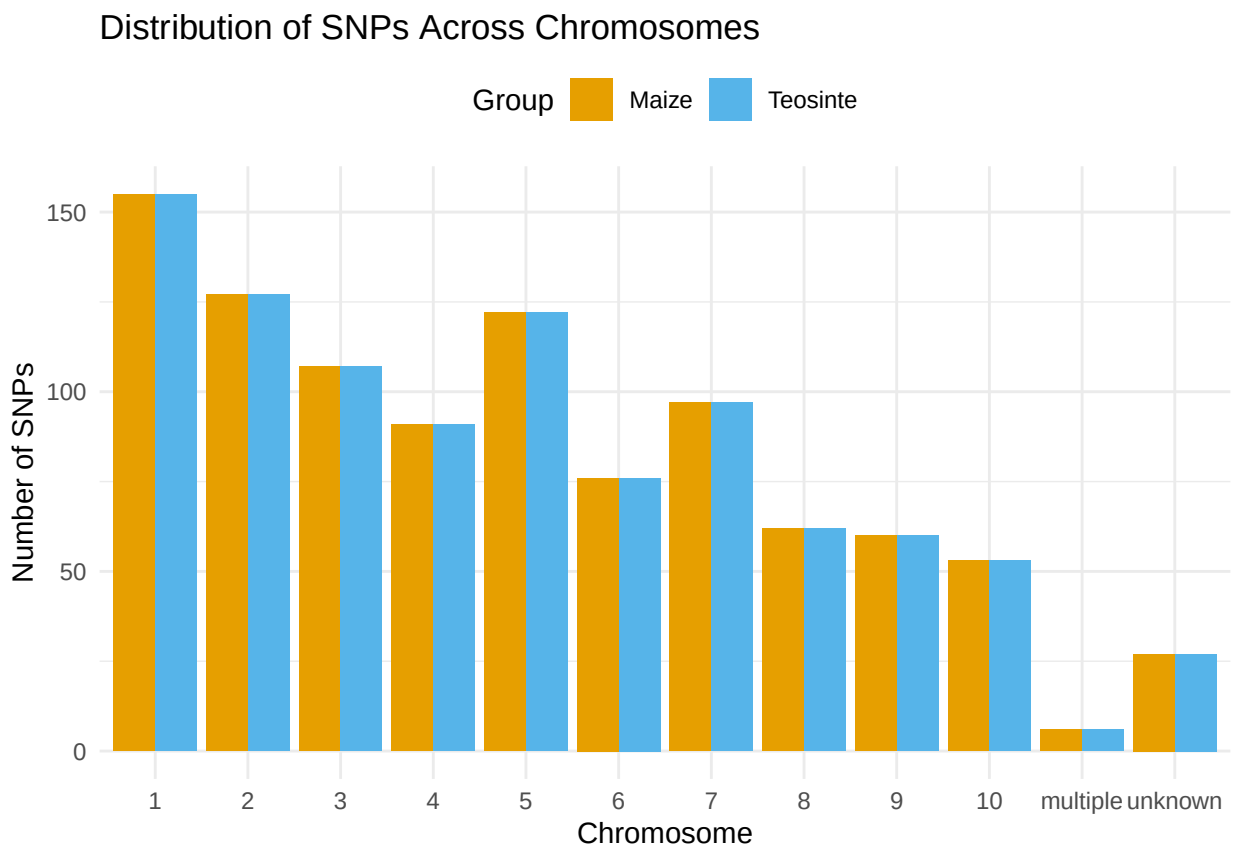
```
snp_counts <- rbind(maize_snp_count, teosinte_snp_count)
```

The chromosomes need to be sorted to be plotted.

```
snp_counts$Chromosome_Numeric <- as.numeric(snp_counts$Chromosome)
```

```
## Warning: NAs introduced by coercion
```

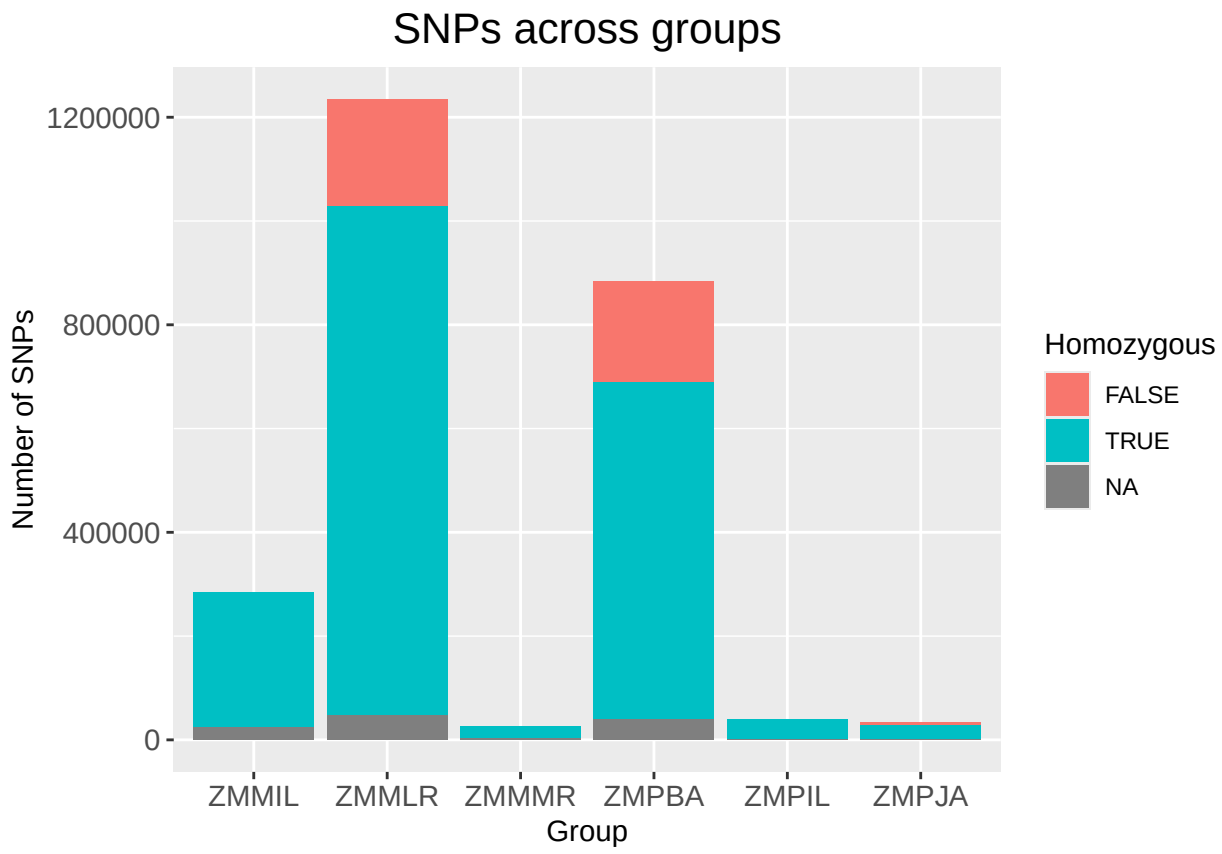
```
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "multiple"] <- 11
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "unknown"] <- 12

unique_chromosomes <- unique(snp_counts[, c("Chromosome", "Chromosome_Numeric")])
unique_chromosomes <- unique_chromosomes[order(unique_chromosomes$Chromosome_Numeric), ]
sorted_chromosome_levels <- unique_chromosomes$Chromosome

snp_counts$Chromosome <- factor(snp_counts$Chromosome, levels = sorted_chromosome_levels)

ggplot(snp_counts, aes(x = Chromosome, y = SNP_Count, fill = Group)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of SNPs Across Chromosomes",
       x = "Chromosome",
       y = "Number of SNPs") +
  scale_fill_manual(values = c("Maize" = "#E69F00", "Teosinte" = "#56B4E9")) +
  theme_minimal() +
  theme(legend.position = "top")
```



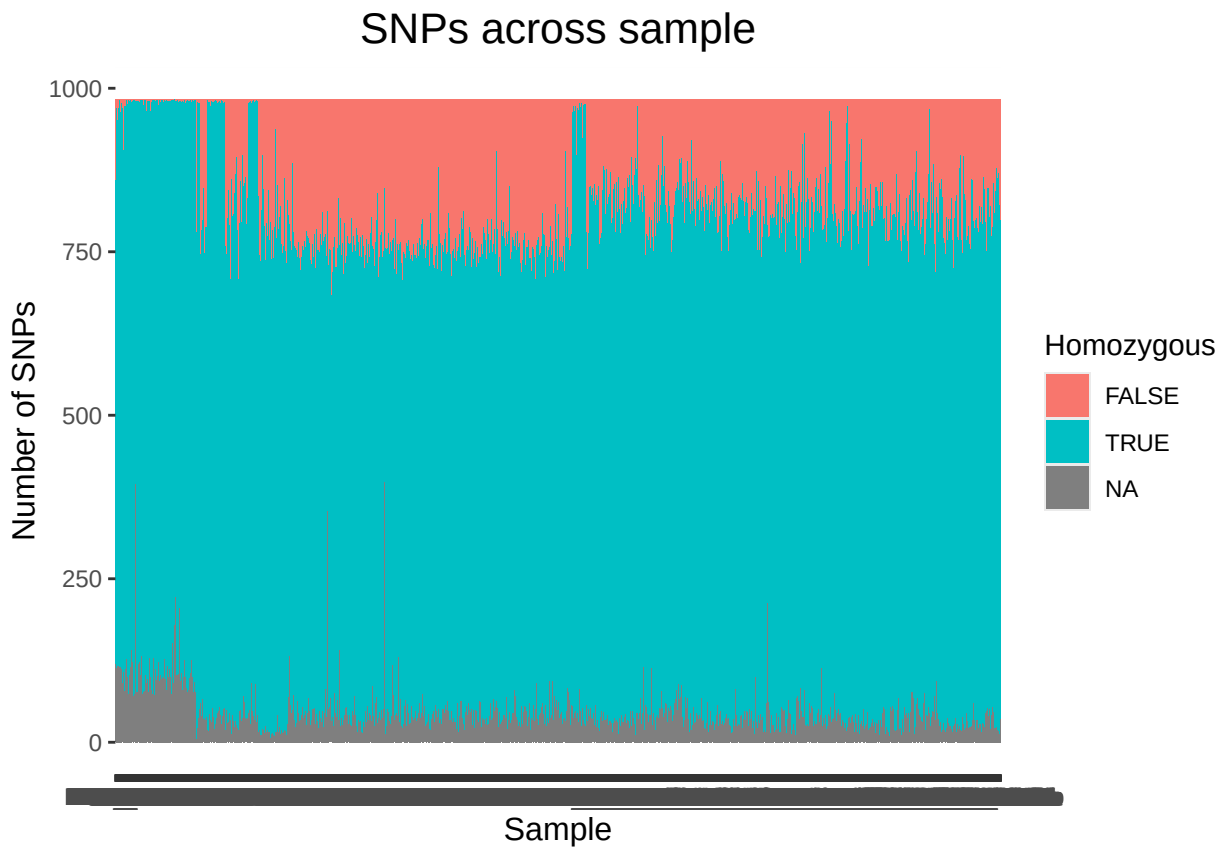
Task-2: Identifying homozygous and heterozygous sites

```
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/A", "C/C", "G/G", "T/T"), TRUE, Homozygous
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/C", "A/G", "A/T", "C/G", "C/T", "G/T"), FA
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("?/?"), NA, Homozygous))
both <- arrange(both, Sample_ID, Group)
```

```
ggplot(data = both) +
  geom_bar(mapping = aes(x = Group, fill = Homozygous), stat = "count") +
  ggtitle(label = "SNPs by  groups") +
  ylab(label = "Number of SNPs") +
  ggtitle(label = "SNPs across groups") +
  xlab(label = "Group") +
  ylab(label = "Number of SNPs") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title
    axis.text = element_text(size = 11),
    axis.title = element_text(size = 11)
  )
```




```
ggplot(data = both) +
  geom_bar(mapping = aes(x = Sample_ID, fill = Homozygous), stat = "count") +
  ggtitle(label = "SNPs by Ordered Sample_ID") +
  ylab(label = "Number of SNPs") +
  ggtitle(label = "SNPs across sample") +
  xlab(label = "Sample") +
  ylab(label = "Number of SNPs") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title
    axis.title = element_text(size = 12)
  )
```



Thus we see that the proportion of homozygous sites are higher compared to heterozygous sites.

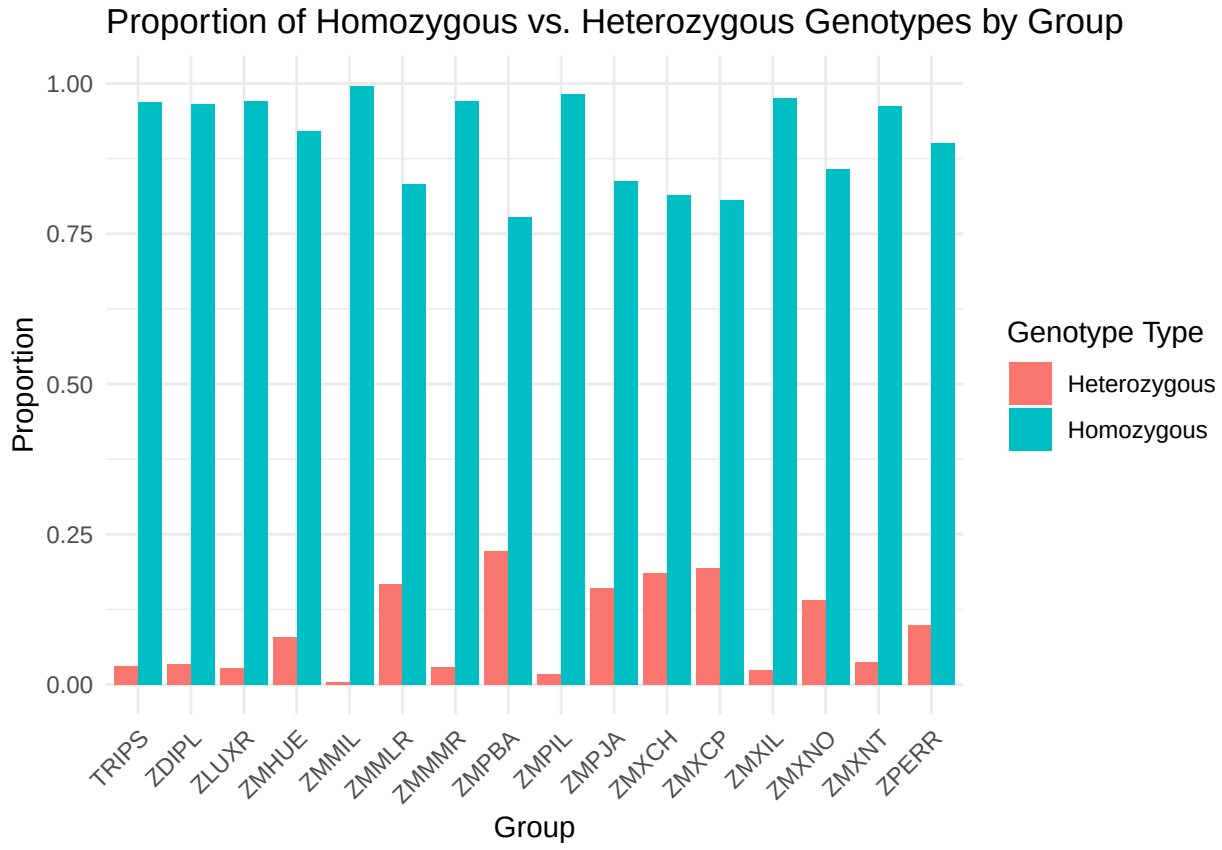
Task3: Own Analysis Reshaping the original data:

```
fang_long <- pivot_longer(fang,
                          cols = -c(Sample_ID, JG_OTU, Group),
                          names_to = "SNP",
                          values_to = "Genotype")
```

Let us analyse the proportion of homozygous and heterozygous sites in all of the groups

```
fang_long <- mutate(fang_long,Genotype_Type = case_when( Genotype == "?" ~ "Missing", str_detect(Genoty
```

```
ggplot(summary_data, aes(x = Group, y = Proportion, fill = Genotype_Type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Proportion of Homozygous vs. Heterozygous Genotypes by Group",
       x = "Group",
       y = "Proportion",
       fill = "Genotype Type") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Proportion of Homozygous vs. Heterozygous Genotypes by Group
Proportion
1.00
0.75
0.50
0.25
0.00
TRIPS
ZDIPL
ZLUXR
ZMHUE
ZMMIL
ZMMLR
ZMMMR
ZMPBA
ZMPIL
ZMPJA
ZMXCH
ZMXCP
ZMXIL
ZMXNO
ZMXNT
ZPERR
Group
Genotype Type
Heterozygous
Homozygous