R project work through:

1. View both files and their structure
   1. You can test if an object is a matrix or array using is.matrix() and is.array(), or by looking at the length of the dim(). as.matrix() and as.array() make it easy to turn an existing vector into a matrix or array.
   2. typeof(df)
   3. [1] "list"
   4. class(df)
   5. [1] "data.frame"
   6. is.data.frame(df)
   7. [1] TRUE
      1. Str(d)
   8. head(d, n=3)
   9. nrow(d)
   10. [1] 59140
   11. ncol(d)
   12. [1] 16
   13. dim(d)
   14. [1] 59140 16
   15. colnames(d)
   16. mean(d$depth)
   17. [1] 8.183938
   18. summary(d$depth)
2. Extract teosinte and maize groups out and put them in their own data set
   1. We can create a logical vector containing whether each observation (row) has 85 or more SNPs using the following:
   2. d$total.SNPs >= 85
   3. We can use this logical vector to extract the rows of our dataframe that have a TRUE value for d$total.SNPs >= 85:
   4. d[d$total.SNPs >= 85, ]
   5. https://eeob-biodata.github.io/R-Data-Skills/04-data-frames/
   6. After all this work, if you want to save your dataframe back to a file use:

write.csv(cats, file = "data/new\_cats.csv")

1. Transpose the data fang et al data set using t(), there will be suprises? Joy..
2. Join the genotypes with the snps
3. Separate the data by chromosomes and create files for each with snps in increasing/decreasing positions
   1. mtfs\_df %>% group\_by(chr)
   2. arrange(d\_df, desc(total.SNPs), desc(depth))
4. Change symbol – to ? in the decreasing ones

Part 2

Add new columns with mutate()

Using dplyr’s mutate() function, we can add new columns to our dataframe: For example, we added a rescaled version of the Pi column as d$diversity—let’s drop d $diversity using select() and then recalculate it:

d\_df <- select(d\_df, -diversity) # remove our earlier diversity column

d\_df <- mutate(d\_df, diversity = Pi/(10\*1000))

d\_df

#pipes

d\_df %>% mutate(GC.scaled = scale(percent.GC)) %>%

filter(GC.scaled > 4, depth > 4) %>%

select(start, end, depth, GC.scaled, percent.GC) %>%

arrange(desc(depth))