Tidyverse Practice (Optional)

NOT DUE, NOT GRADED, OPTIONAL PRACTICE

Problem 1 (Optional)

This problem is **OPTIONAL** and designed to help you learn to use `dplyr`. This problem will NOT be graded.

Read in the example VCF data from GitHub with the following command:

```
file_url <- "https://github.com/katiesevans/IGP_biostatistics/blob/main/data/sample_vcf.tsv.gz?raw=true"
download.file(url = file_url, destfile = "sample_vcf.tsv.gz")
vcf <- readr::read_tsv("sample_vcf.tsv.gz")
```

VCF stands for "variant call format". It is a standardized way to show at which genomic positions individuals differ from the "reference". If you are interested, you can find more information about VCFs here or here. However, biological knowledge like this will not be necessary for answering this problem.

a. Using the `str()` and `unique()` functions (and any others you feel), look at the data. What type of data is it? How many rows and columns? What type of values are in each column?

The next few steps are broken down into different steps, but in practicality can be written together as one "chunk" of code (using pipes to connect each step) if you prefer (recommended)

b. Subset to keep only chromosomes IV and V
c. Remove all quality calls of "high_heterozygosity"
d. Keep only bi-allelic sites (remove any rows with a comma in the "ALT" column). *Hint: check out the useful `grepl(pattern, vector)` function*
e. Remove missing genotype calls (a missing genotype is represented as "./."). After this step, the only possible genotypes left should be "0/0" or "1/1" – check this with `unique()`
f. Make a new column called "genotype" that is either "REF" for a GT of "0/0" or "ALT" for a GT of "1/1" (*Hint: try using `ifelse` combined with a `mutate`*)
g. Remove the "QUAL", "AF", and "DP" columns
h. Group the data by variant (chrom, pos, and genotype)
i. Make a new column that counts the number of strains with REF or ALT calls for each variant *Hint, the `n()` function counts the number of observations in each group. Use this in combination with `summarize()`*
j. Spread (or `pivot_wider`) the genotype data so that you end up with a REF column and an ALT column with number of strains containing each type of allele in each (*this function is in the `tidyr` package, not `dplyr`!!*) (*Hint: you might want to use `values_fill = 0` to avoid adding NAs at this step*)
k. Calculate the "percent reference" (i.e. REF / (REF + ALT)) for each variant
l. Plot the "percent reference" column by genomic position! (*Hint: if using ggplot, try to `facet_grid` by chromosome. Alternatively, make one plot for chrIV and one plot for chrV*) **For reference, here is the plot you are aiming to produce:**