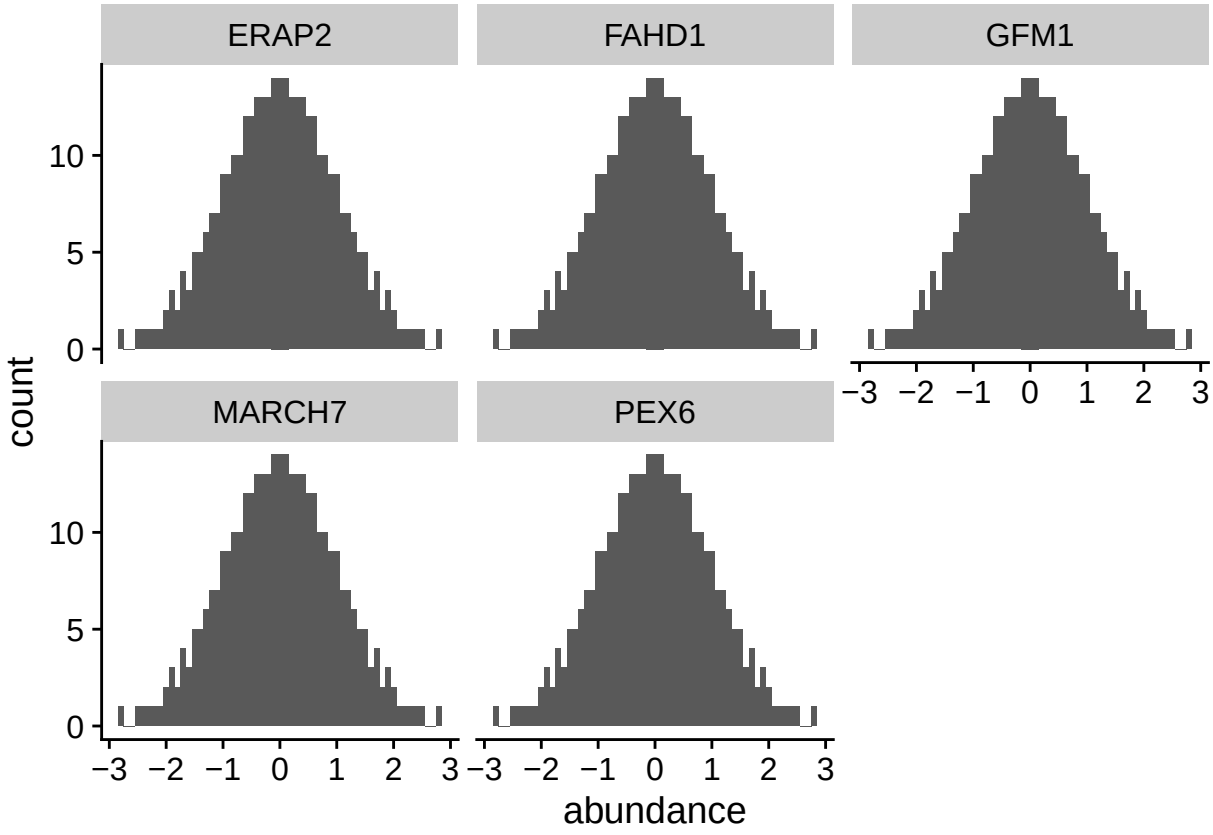
quant_gen_project

Darya Akimova

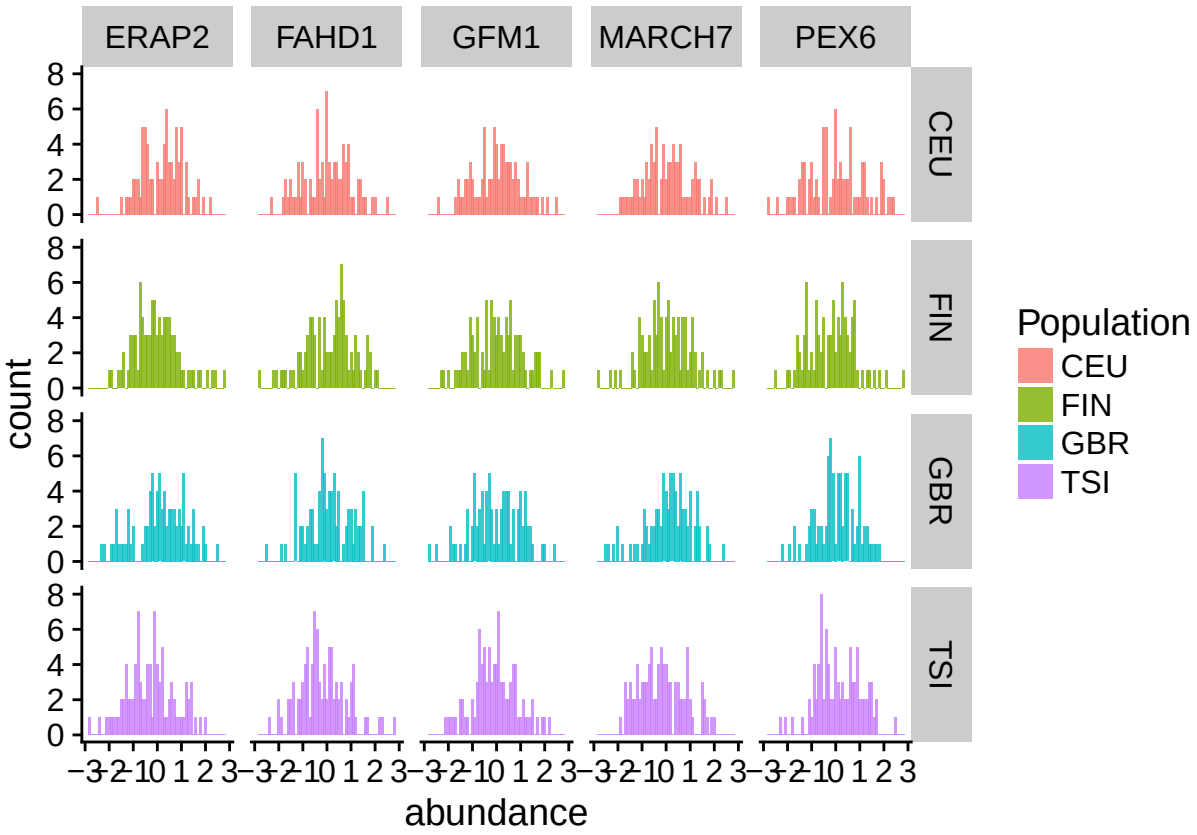
April 20, 2018

Phenotype plots:

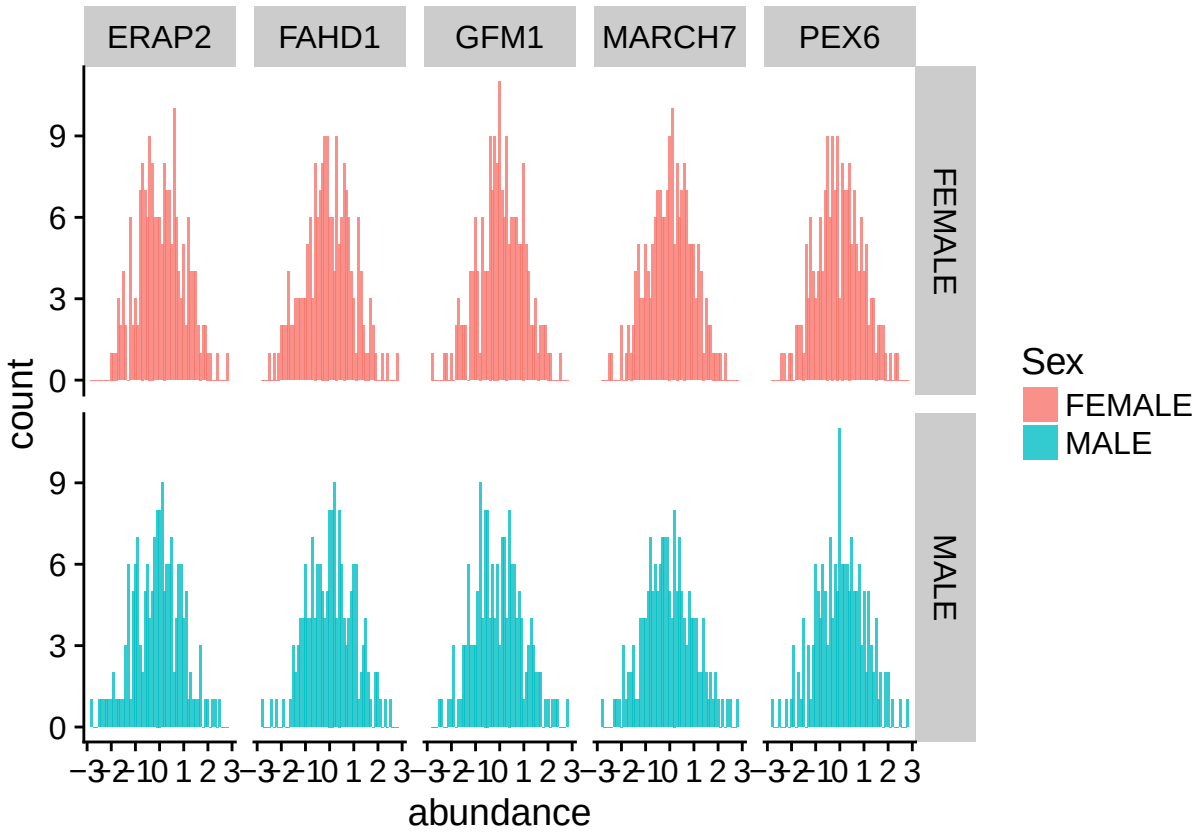
```
pheno %>%
  rownames_to_column("sample") %>%
  gather("probe", "abundance", 2:6) %>%
  left_join(gene_info, by = "probe") %>%
  ggplot(aes(x = abundance)) +
  geom_histogram(binwidth = 0.1) +
  facet_wrap(~ symbol)
```



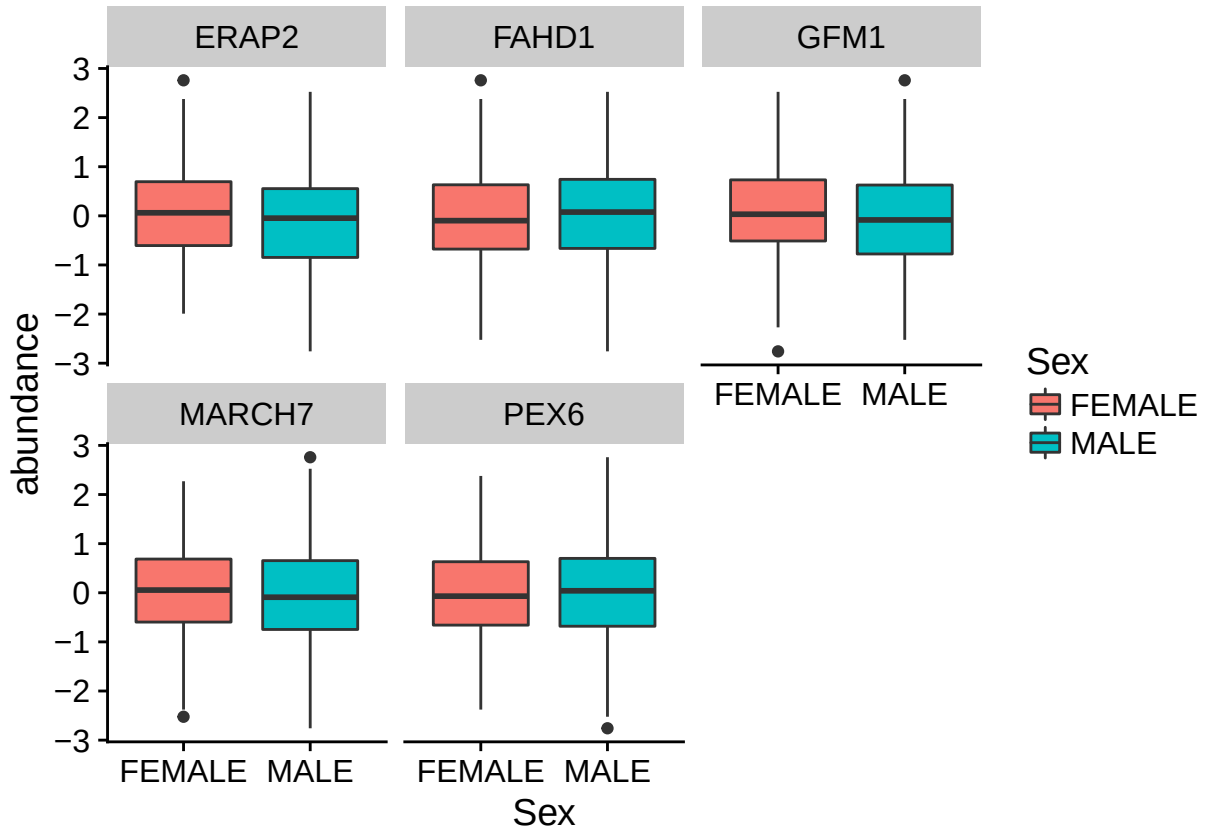

```
pheno %>%
  rownames_to_column("sample") %>%
  gather("probe", "abundance", 2:6) %>%
  left_join(gene_info, by = "probe") %>%
  left_join(
    covars %>% rownames_to_column("sample"),
    by = "sample") %>%
  ggplot(aes(x = abundance, fill = Population)) +
  geom_histogram(binwidth = 0.1, alpha = 0.8, position = "identity") +
  facet_grid(Population ~ symbol)
```



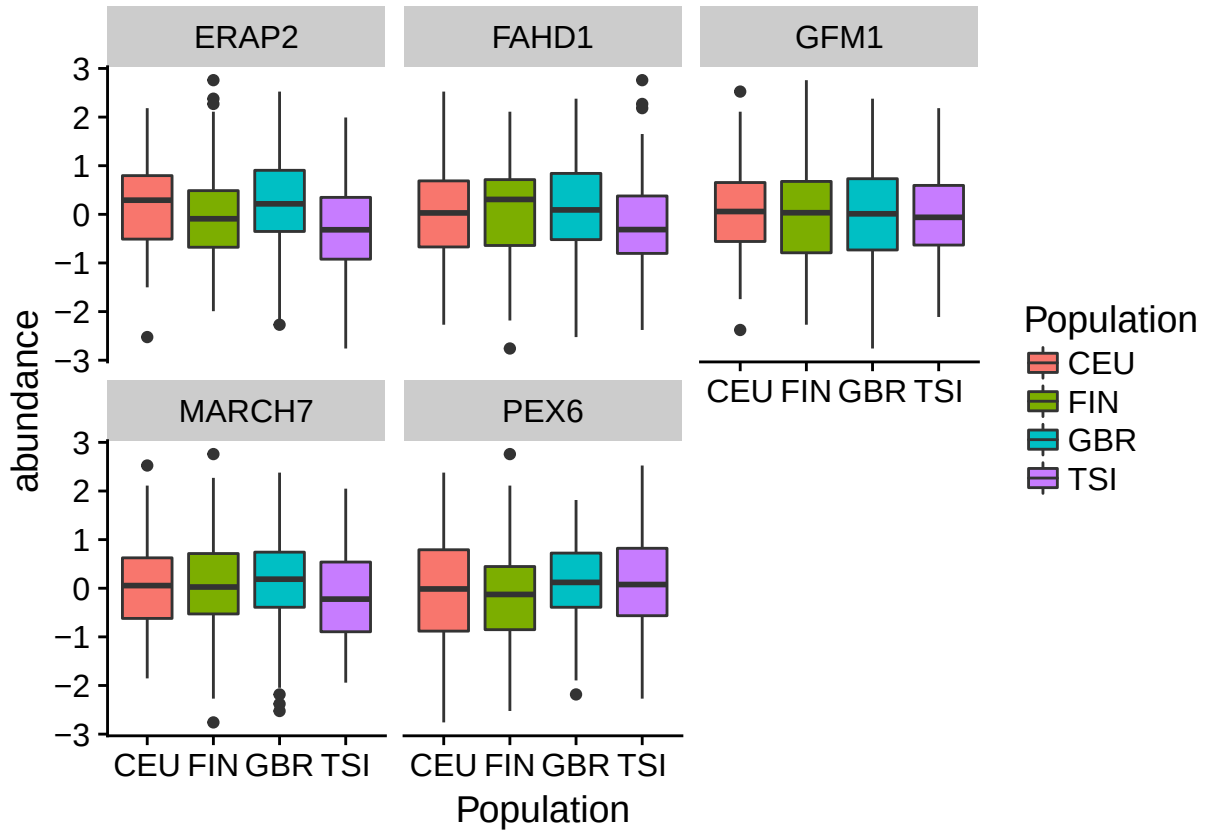

```
pheno %>%
  rownames_to_column("sample") %>%
  gather("probe", "abundance", 2:6) %>%
  left_join(gene_info, by = "probe") %>%
  left_join(
    covars %>% rownames_to_column("sample"),
    by = "sample") %>%
  ggplot(aes(x = abundance, fill = Sex)) +
  geom_histogram(binwidth = 0.1, alpha = 0.8, position = "identity") +
  facet_grid(Sex ~ symbol)
```




```
pheno %>%
  rownames_to_column("sample") %>%
  gather("probe", "abundance", 2:6) %>%
  left_join(gene_info, by = "probe") %>%
  left_join(
    covars %>% rownames_to_column("sample"),
    by = "sample") %>%
  ggplot(aes(x = Sex, y = abundance, fill = Sex)) +
  geom_boxplot() +
  facet_wrap(~ symbol)
```




```
pheno %>%
  rownames_to_column("sample") %>%
  gather("probe", "abundance", 2:6) %>%
  left_join(gene_info, by = "probe") %>%
  left_join(
    covars %>% rownames_to_column("sample"),
    by = "sample") %>%
  ggplot(aes(x = Population, y = abundance, fill = Population)) +
  geom_boxplot() +
  facet_wrap(~ symbol)
```




```
table(covars$Population)
```

```
CEU FIN GBR TSI
 78  89  85  92
```

```
table(covars$Sex)
```

```
FEMALE   MALE
   181    163
```

```
# does the coding of the genotypes makes sense across all the data?
max(as.matrix(geno))
```

```
[1] 2
```

```
min(as.matrix(geno))
```

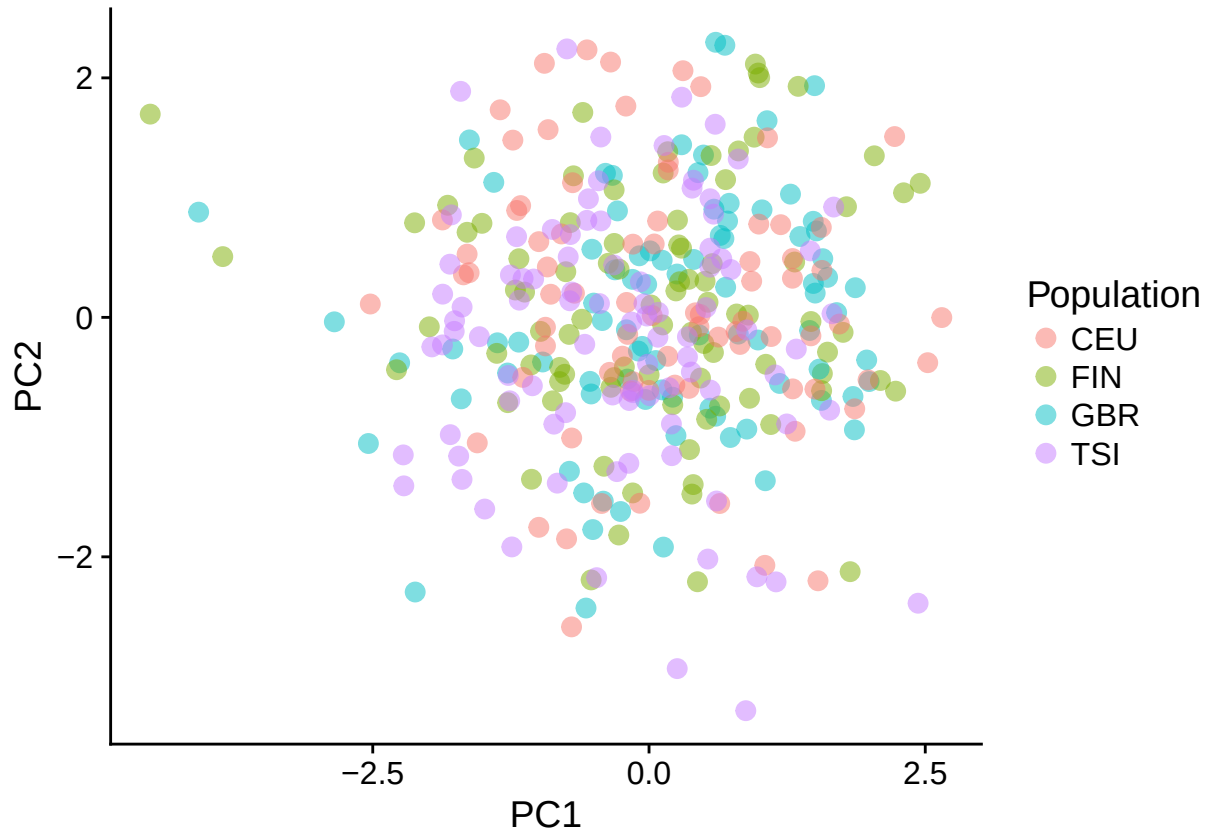
```
[1] 0
```

```
# are there any genotypes without associated phenotypes, or vice versa?
all.equal(rownames(geno), rownames(pheno))
```

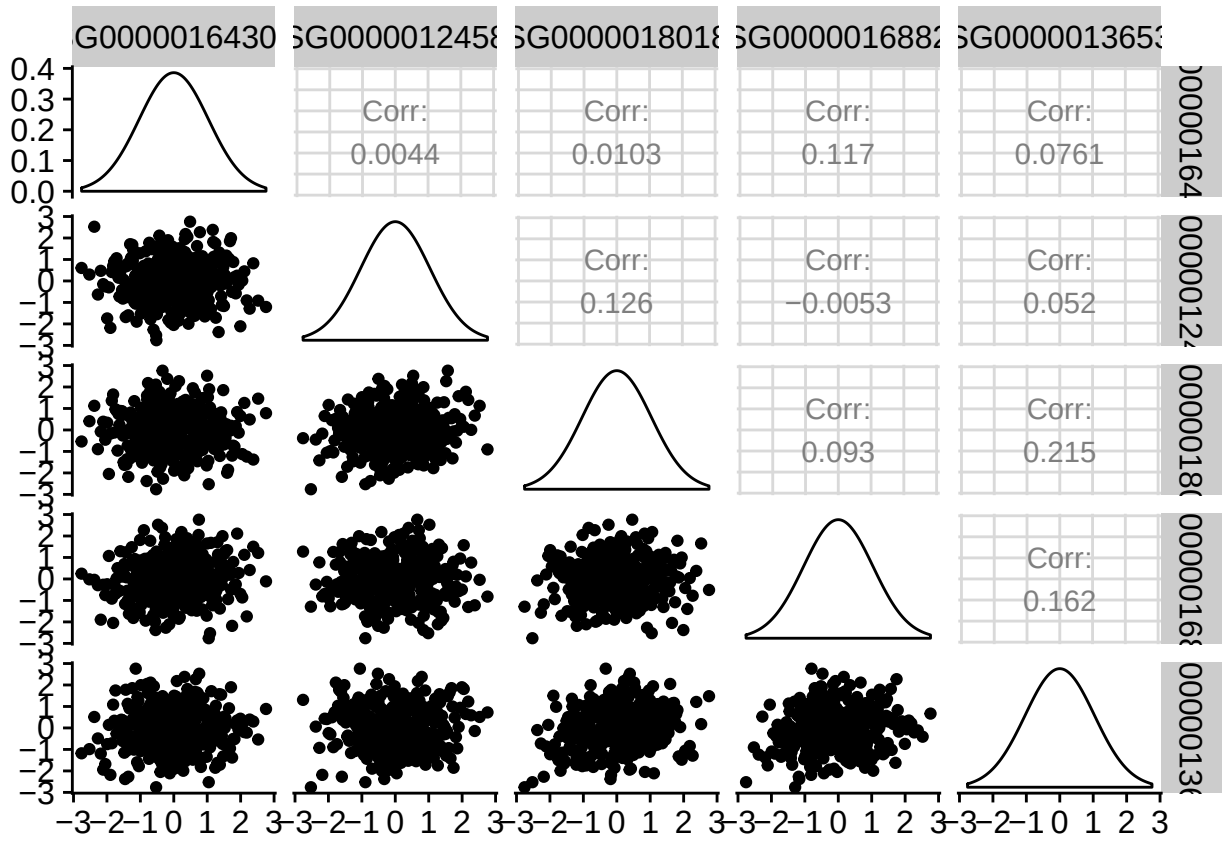
```
[1] TRUE
```

```
pheno_pca <- prcomp(pheno)
pheno_pca_x <- as.data.frame(pheno_pca$x)
pheno_pca_x %>%
  rownames_to_column("sample") %>%
```

```
  left_join(covars %>% rownames_to_column("sample"), by = "sample") %>%
  ggplot(aes(x = PC1, y = PC2, color = Population)) +
  geom_point(size = 3, alpha = 0.5)
```




```
ggpairs(pheno)
```

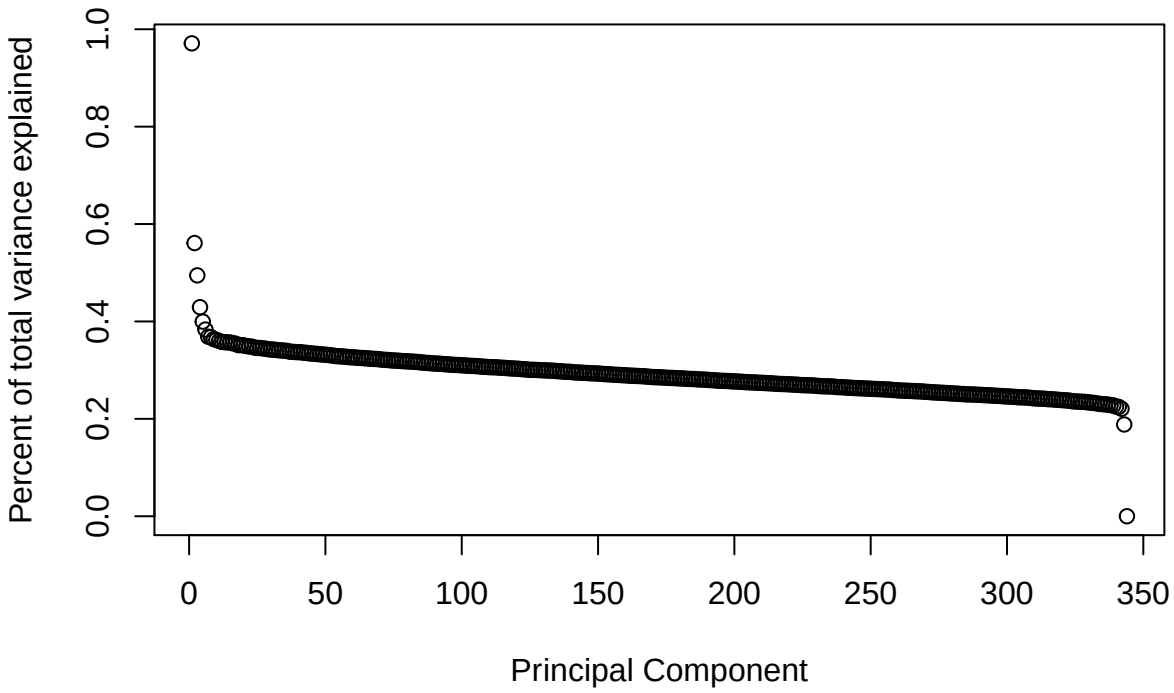


Genotype plots:

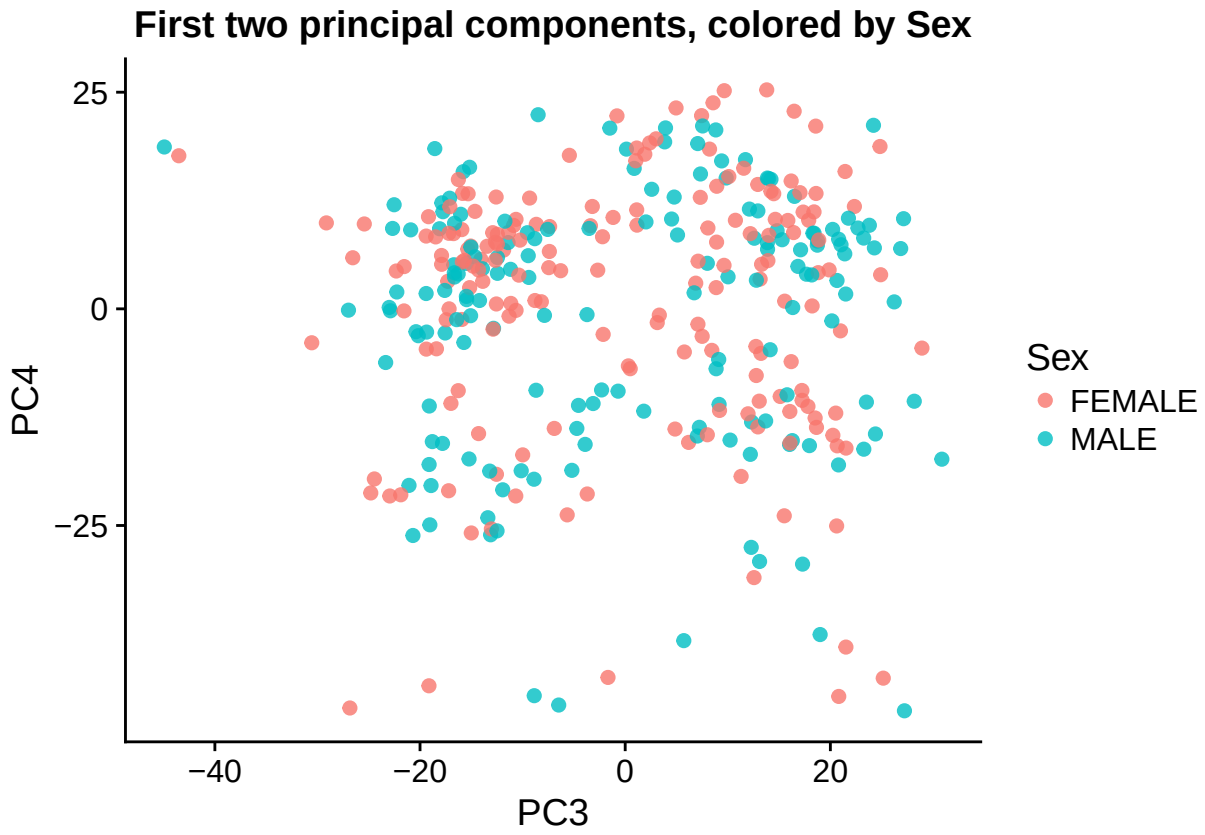
```
# calculate allele frequency
geno_sums <- sapply(geno, function(x) sum(x) / (nrow(geno) * 2))
# any genotypes need to be removed because of MAF < 5%?
which(geno_sums < 0.05 | geno_sums > 0.95)
```

```
named integer(0)
```

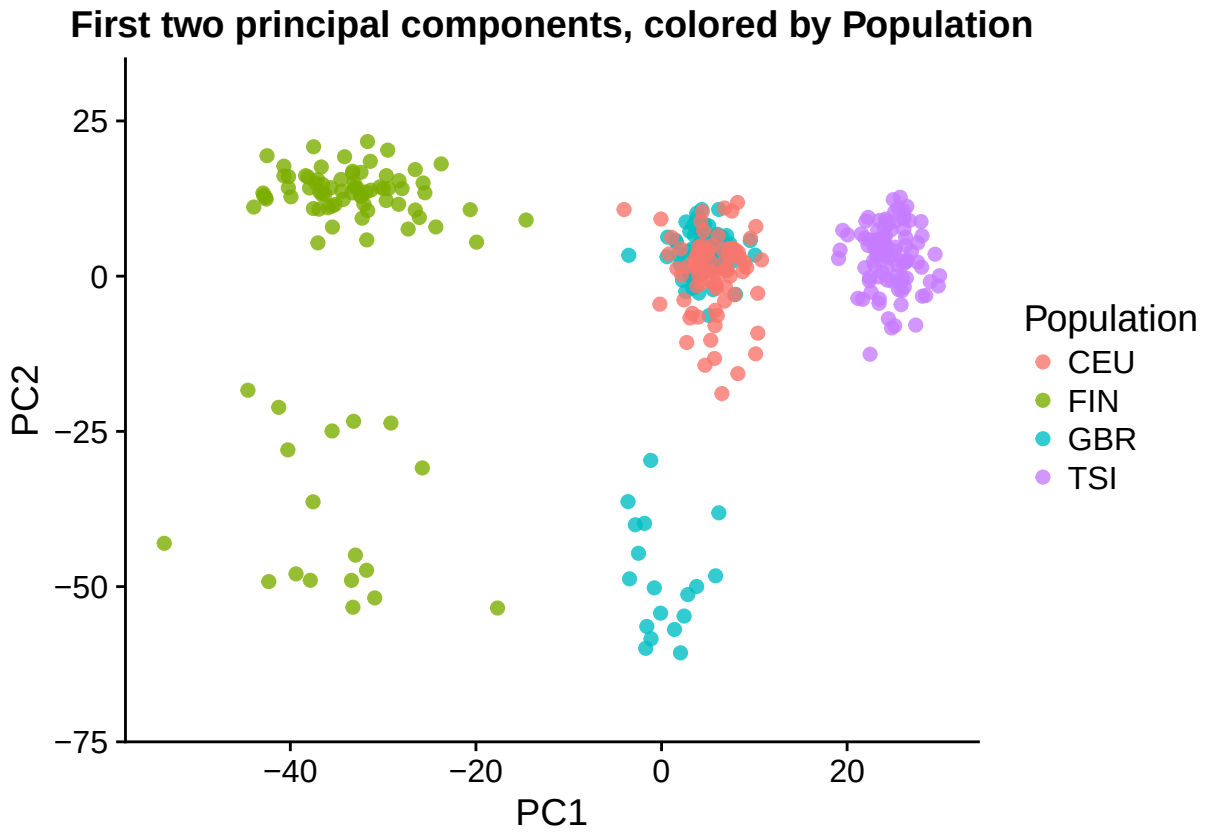
```
# no genotypes need to be removed because of MAF < 5%
# PCA analysis of genetic data
geno_pca <- prcomp(geno, center = TRUE, scale. = TRUE)
plot((geno_pca$sdev^2/ sum(geno_pca$sdev^2)) * 100, xlab = "Principal Component", ylab = "Percent of to
```




```
geno_pca_x <- data.frame(geno_pca$x)
geno_pca_x %>%
  rownames_to_column("sample") %>%
  left_join(covars %>% rownames_to_column("sample"), by = "sample") %>%
  ggplot(aes(x = PC3, y = PC4, color = Sex)) +
    geom_point(size = 2, alpha = 0.8) +
    ggtitle("First two principal components, colored by Sex")
```




```
geno_pca_x %>%
  rownames_to_column("sample") %>%
  left_join(covars %>% rownames_to_column("sample"), by = "sample") %>%
  ggplot(aes(x = PC1, y = PC2, color = Population)) +
    geom_point(size = 2, alpha = 0.8) +
    ggtitle("First two principal components, colored by Population") +
  ylim(-70, 30)
```



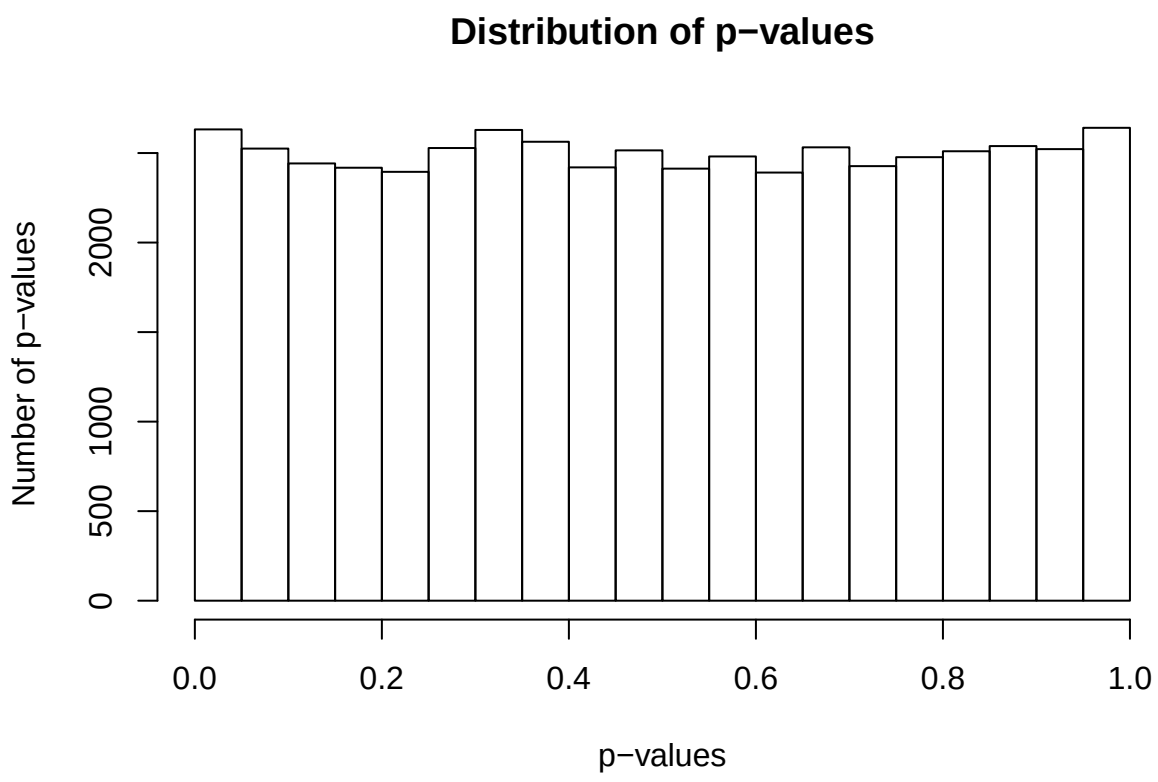

```
indv_pca <- prcomp(t(geno), center = TRUE, scale. = TRUE)
indv_pca_x <- indv_pca$x
```

There's clearly population structure.

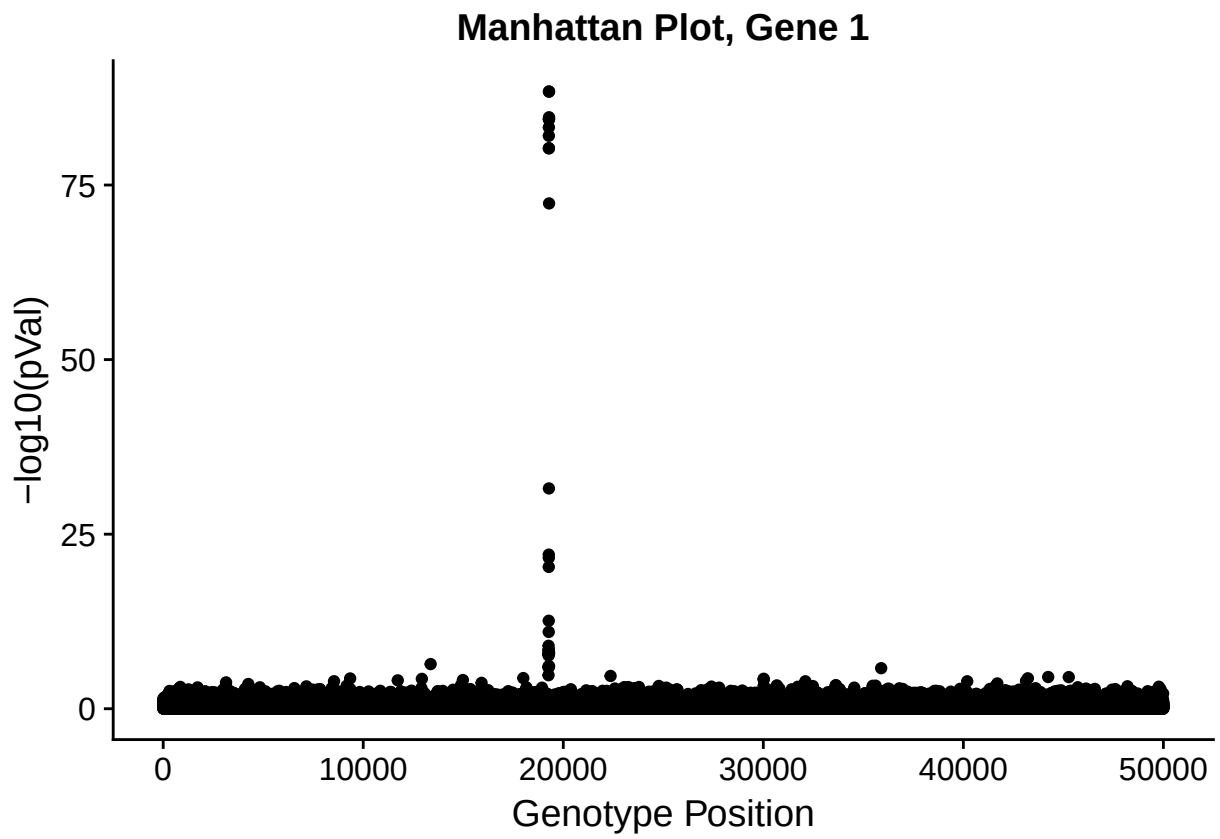
Test each covariate individually.

```
x_a <- as.matrix(geno - 1)
x_d <- replace(as.matrix(geno), which(as.matrix(geno) == 2 | as.matrix(geno) == 0), -1)
MLE <- function(y_mat, xa, xd) {
  X_mat <- cbind(1, xa, xd)
  beta_hat <- ginv(t(X_mat) %*% X_mat) %*% t(X_mat) %*% y_mat
}
MLE_result_col1 <- matrix(NA, nrow = 3, ncol = ncol(geno))
for(i in 1:ncol(geno)){
  MLE_result_col1[,i] <- MLE(as.numeric(as.matrix(pheno[, 2])), x_a[,i], x_d[,i])
}
fstat_calc <- function(y_mat, xa, xd, MLE) {
  X_mat <- cbind(1, xa, xd)
  y_hat <- X_mat %*% MLE
  SSM <- sum((y_hat - mean(y_mat)) ^ 2)
  SSE <- sum((y_mat - y_hat) ^ 2)
  df_M <- 2
  df_E <- length(xa) - 3
  Fstat <- (SSM / df_M) / (SSE / df_E)
  return(Fstat)
}
```

```r
fstat_result_col1 <- matrix(NA, nrow = 1, ncol = ncol(geno))
for(i in 1:ncol(geno)) {
  fstat_result_col1[i] <- fstat_calc(pheno[, 2], x_a[, i], x_d[, i], MLE_result_col1[, i])
}
# check the degrees of freedom numbers
pval_result_col1 <- pf(fstat_result_col1, 2, 344 - 3, lower.tail = FALSE)
hist(pval_result_col1, breaks = 20,
     xlab = "p-values",
     ylab = "Number of p-values",
     main = "Distribution of p-values")
```



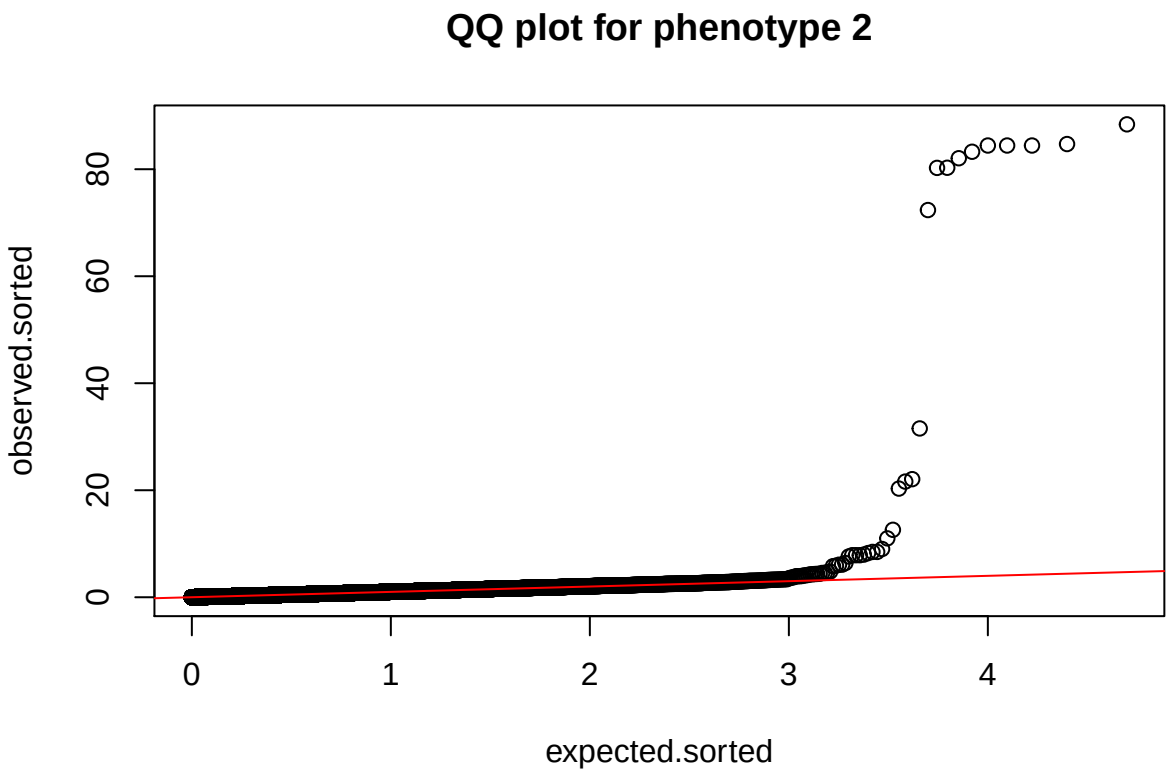

```r
man_plot_data_col1 <- as.data.frame(cbind(1:ncol(geno), t(pval_result_col1)))
colnames(man_plot_data_col1) <- c("x", "pval")
ggplot(man_plot_data_col1, aes(x = x, y = -log(pval, base = 10))) +
  geom_point() +
  ggtitle("Manhattan Plot, Gene 1") +
  xlab("Genotype Position") +
  ylab("-log10(pVal)")
```




```
summary(which(pval_result_col1 < 0.05 / ncol(geno)))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13371   19271   19278   19075   19285   19293
```

```
expected.pvals <- -log10(seq(from=0,to=1, length.out = length(man_plot_data_col1$pval)))
observed.pvals <- -log10(man_plot_data_col1$pval)
expected.sorted <- sort(expected.pvals)
observed.sorted <- sort(observed.pvals)
plot(expected.sorted, observed.sorted, main = "QQ plot for phenotype 2")
abline(a = 0,b = 1, col = "red", lwd = 1)
```




```
lm_test <- lm(pheno$ENSG00000164308.12 ~ x_a[, 1000] + x_d[, 1000] + factor(covars$Population))
lm_tidy <- tidy(lm_test)
fstat <- summary(lm_test)$fstatistic
fstat_2 <- glance(lm_test)$statistic
pf(fstat[1], fstat[2], fstat[3], lower.tail = FALSE)
```

```
    value
0.0190177
```