

2023 Cleaned Graphics

Michelle Bang

2023-03-13

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

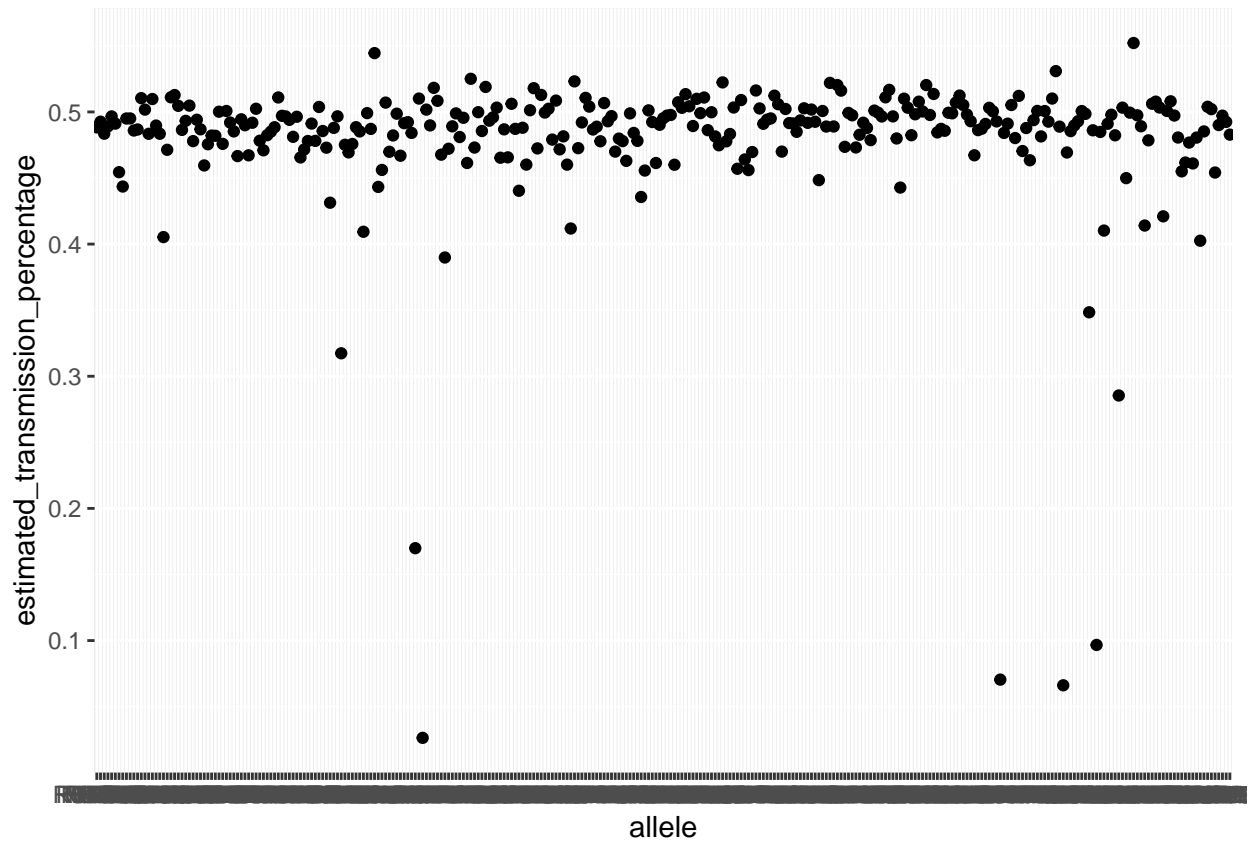
## v ggplot2 3.3.5    v purrr  0.3.4
## v tibble  3.1.6    v dplyr  1.0.7
## v tidyr   1.1.4    v stringr 1.4.0
## v readr   2.1.1    v forcats 0.5.1

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

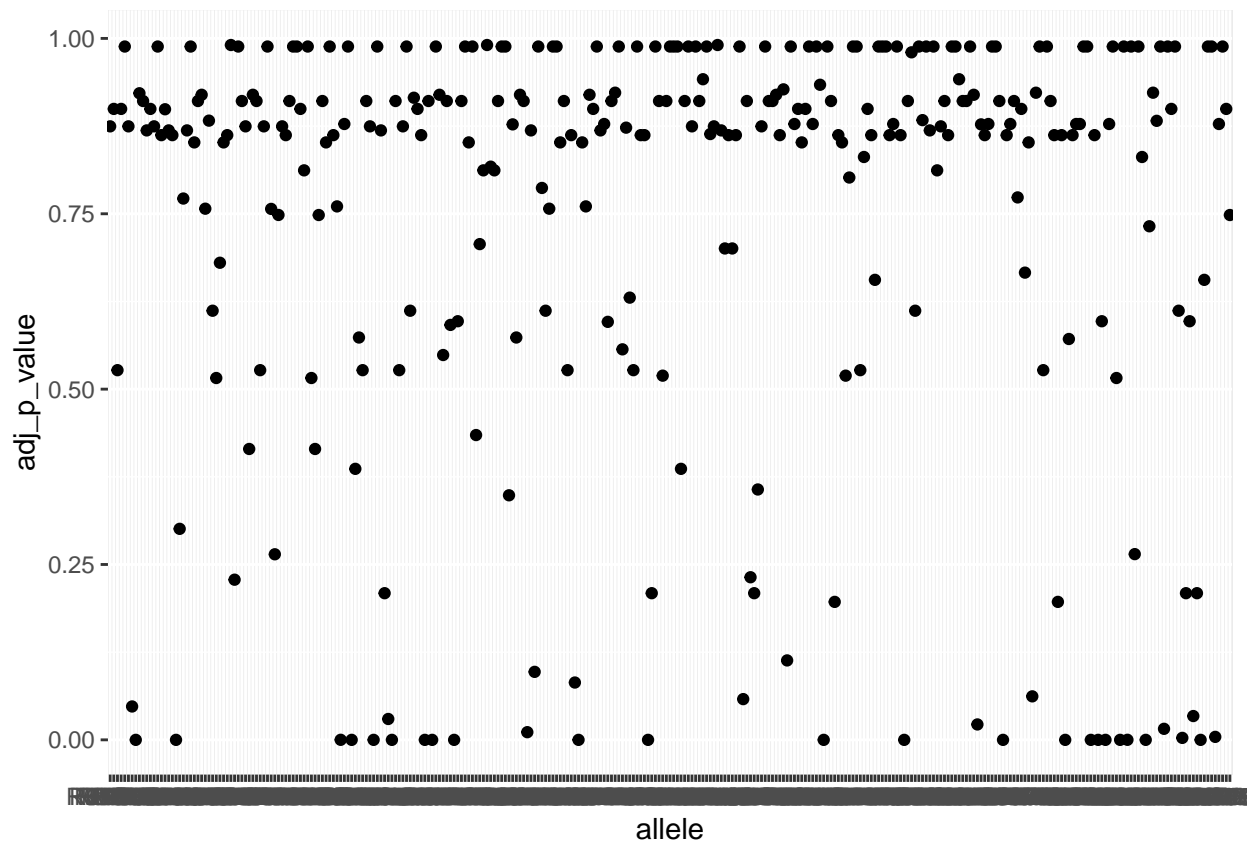
asn <- read.table(file = "2023_cleaned_analysis_sum.tsv",
                  header = TRUE,
                  sep = '\t')

asn_pcr <- read.table(file = "2023_cleanPCR_analysis_sum.tsv",
                     header = TRUE,
                     sep = '\t')

ggplot(asn, mapping = aes(x = allele,
                          y = estimated_transmission_percentage)) + geom_point()
```



```
ggplot(asn, mapping = aes(x = allele,  
                           y = adj_p_value)) + geom_point()
```



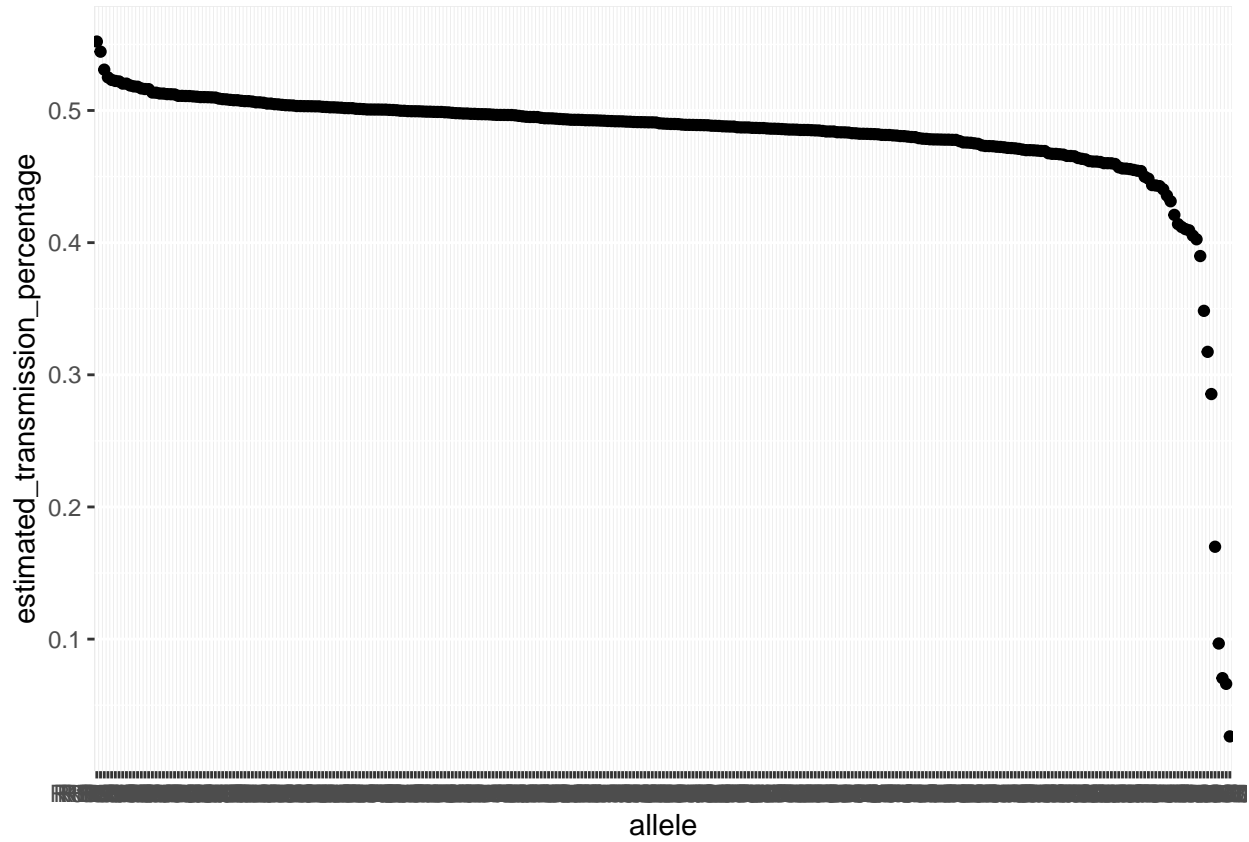
#ADDITIONAL REQUESTS FROM JOHN (02/28)

```
ordered_df <- asn %>%  
  arrange(desc(estimated_transmission_percentage))
```

```
write.table(ordered_df, file = "2023_cleaned_analysis_ordered.tsv", sep = "\t", row.names=FALSE)
```

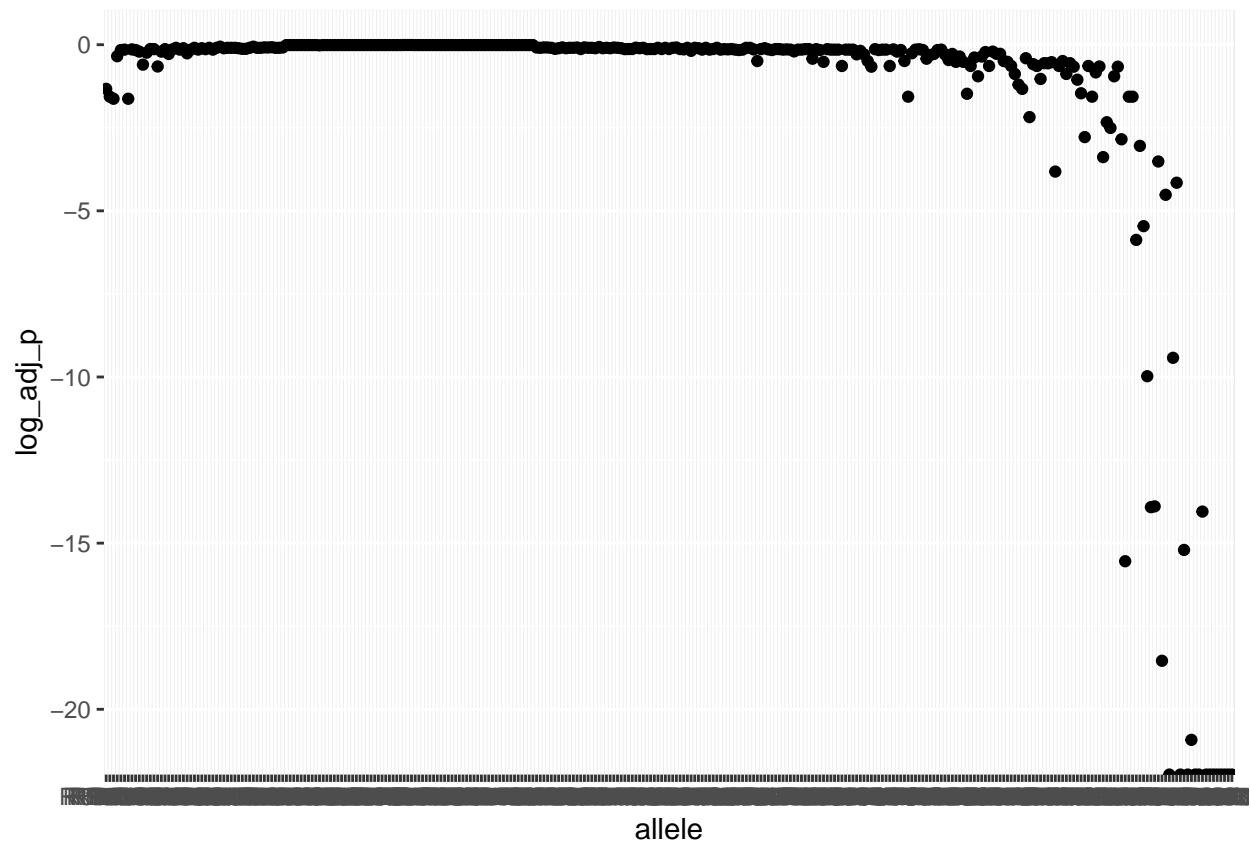
GRAPHS

```
ordered_plot <- ggplot(ordered_df, mapping = aes(x = allele,  
                                                  y = estimated_transmission_percentage )) + geom_point()  
  
ordered_plot + scale_x_discrete(limits = ordered_df[["allele"]])
```

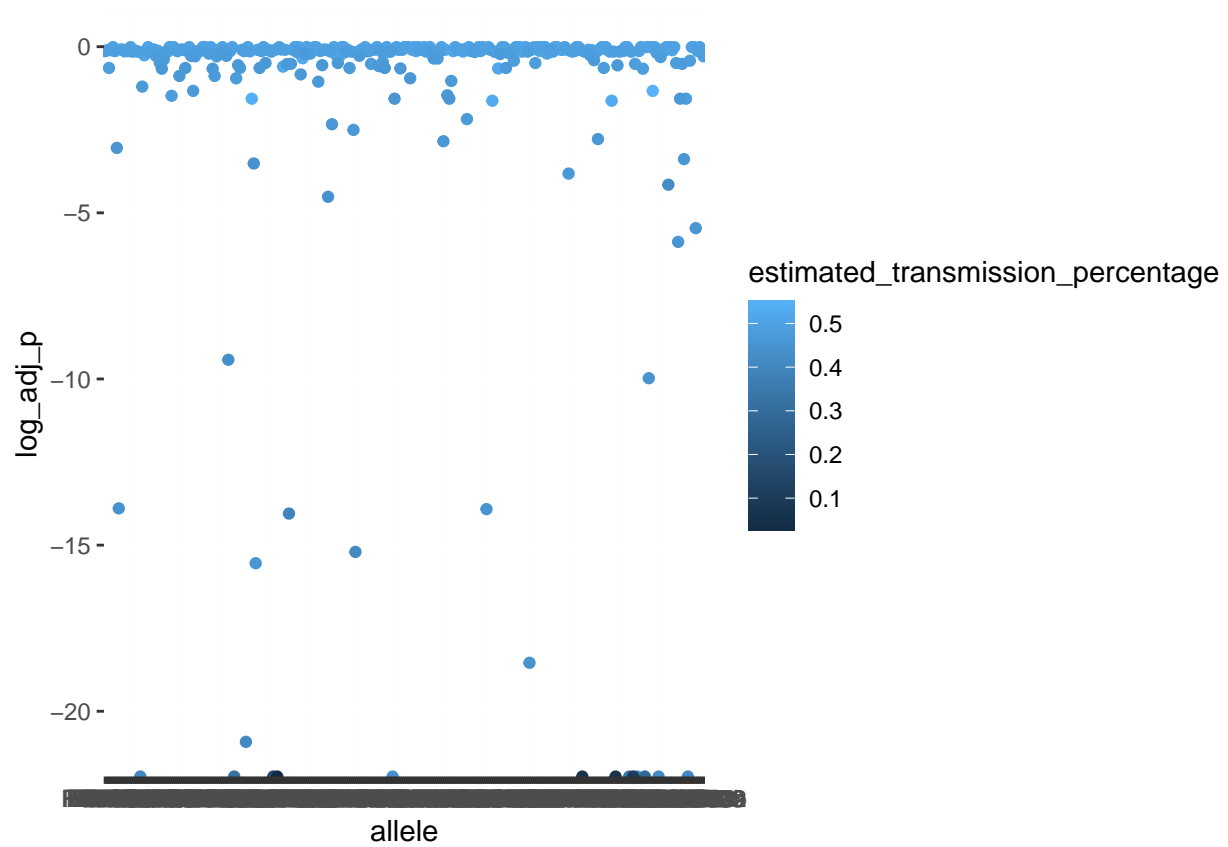


```
with_log_df <- ordered_df %>%  
  mutate(log_adj_p = log(adj_p_value)) %>%  
  #only R187B08 had log(adj_p_value) of 0  
  mutate(log_adj_p = if_else(log_adj_p == 0, 1.0*10(-15), log_adj_p))
```

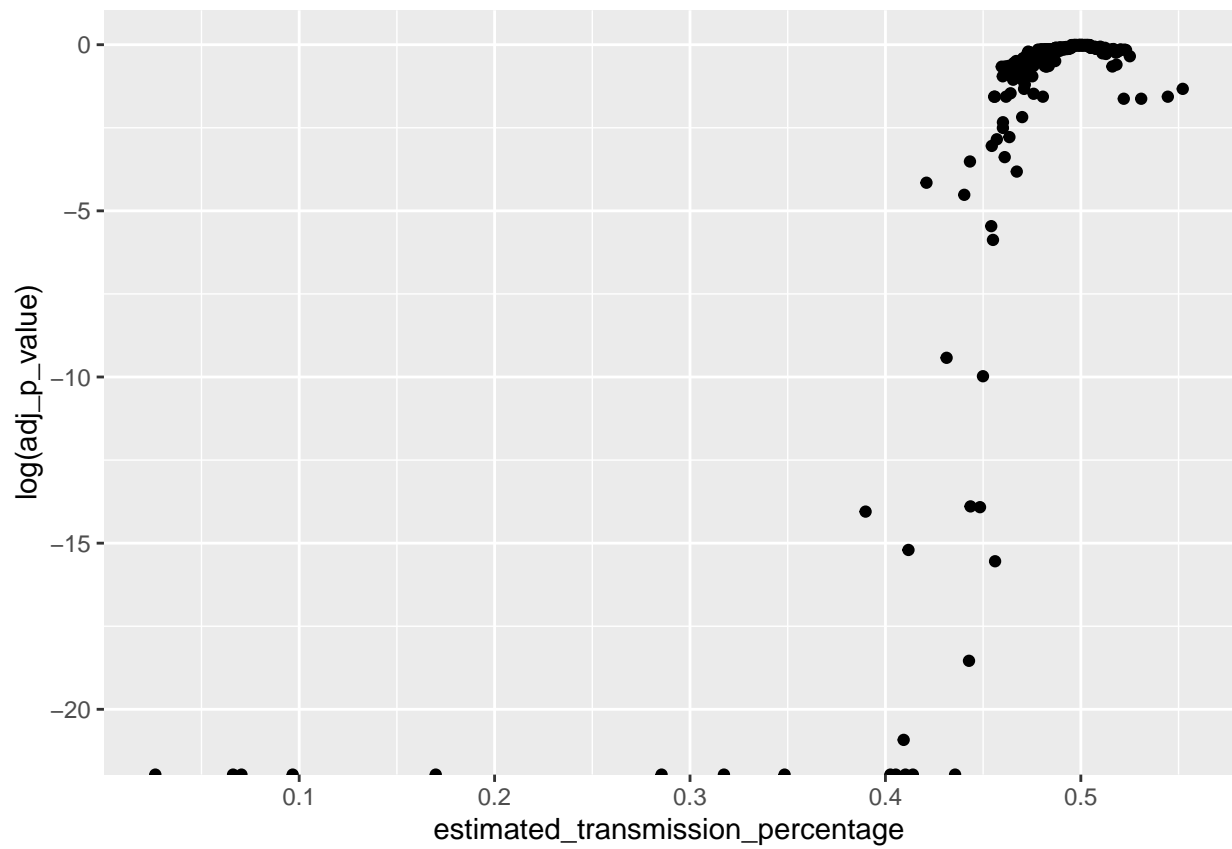
```
ordered_log <- ggplot(with_log_df, mapping = aes(x = allele,  
                                                  y = log_adj_p)) + geom_point()  
  
ordered_log + scale_x_discrete(limits = ordered_df[["allele"]])
```



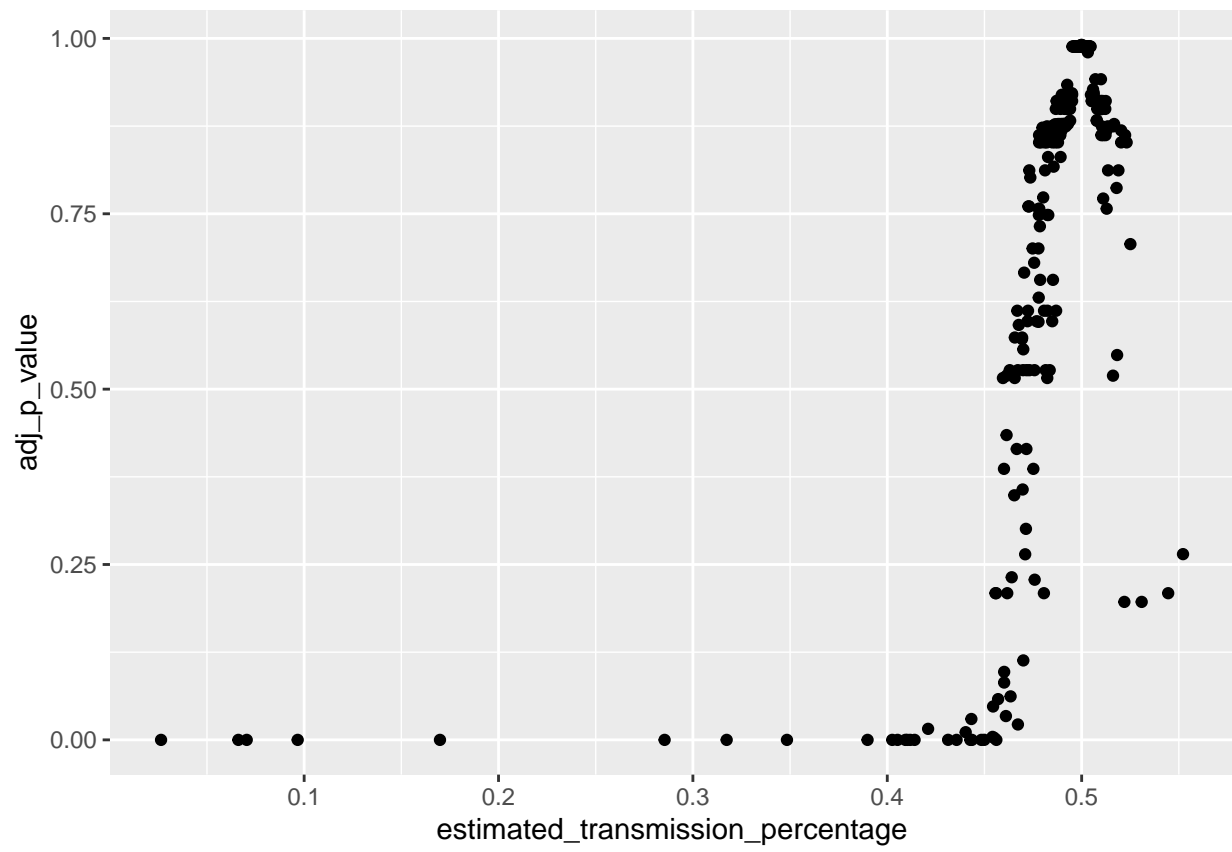
```
ggplot(with_log_df, mapping = aes(x = allele,  
  y = log_adj_p,  
  col = estimated_transmission_percentage)) + geom_point()
```



```
ggplot(with_log_df, mapping = aes(x = estimated_transmission_percentage,  
y = log(adj_p_value))) + geom_point()
```



```
ggplot(asn, mapping = aes(x = estimated_transmission_percentage,  
                           y = adj_p_value)) + geom_point()
```

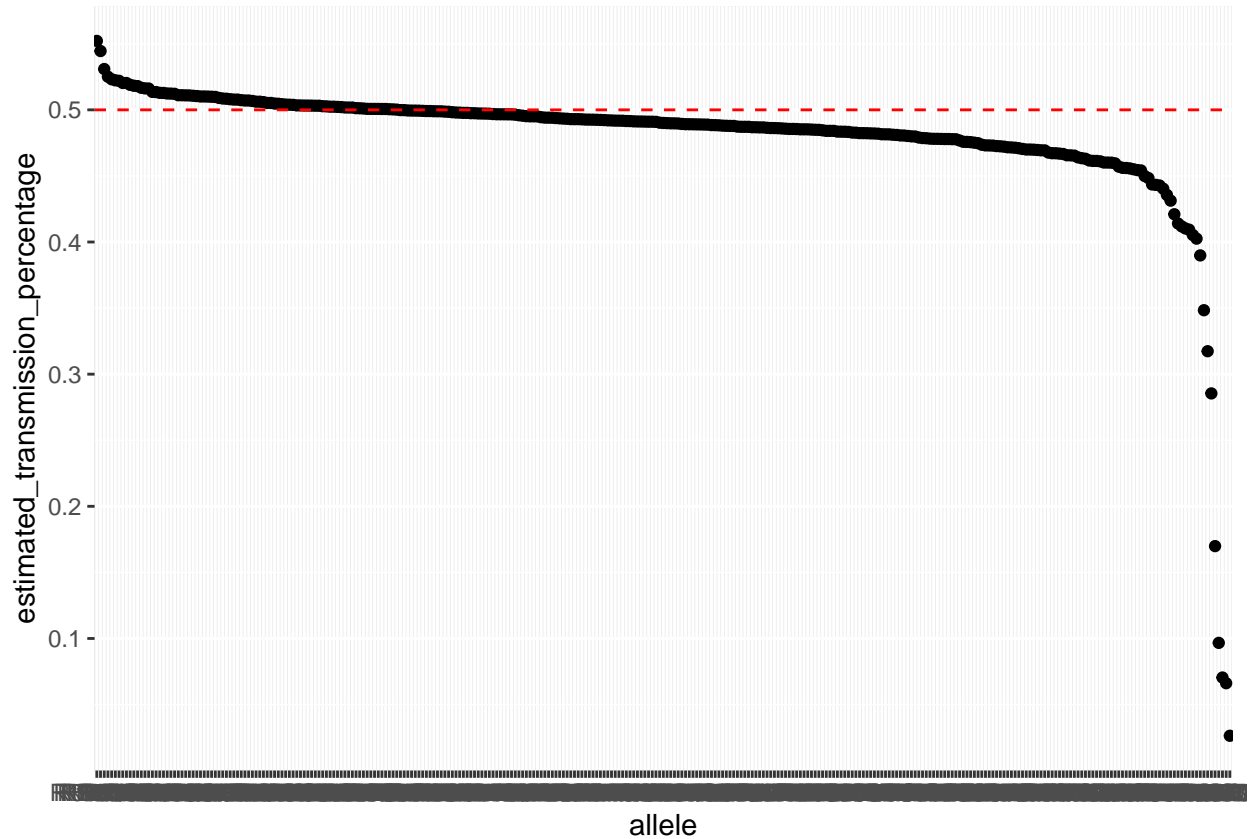



#NEW GRAPHICS...

```
ordered_new <- ordered_df %>%  
  arrange(desc(estimated_transmission_percentage))
```

```
ordered_plot_new <- ggplot(ordered_new, mapping = aes(x = allele,  
  y = estimated_transmission_percentage )) + geom_point()
```

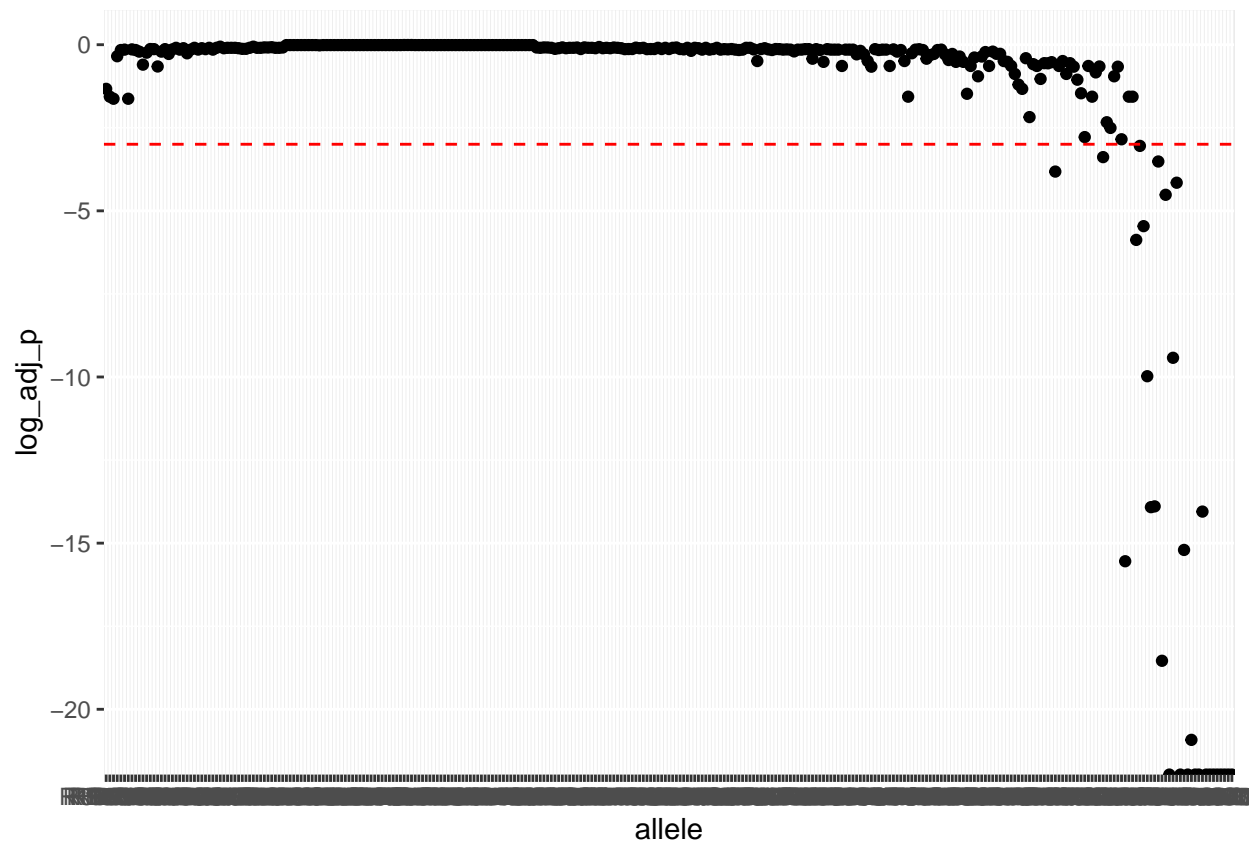
```
ordered_plot_new + scale_x_discrete(limits = ordered_new[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed", color = "red")
```



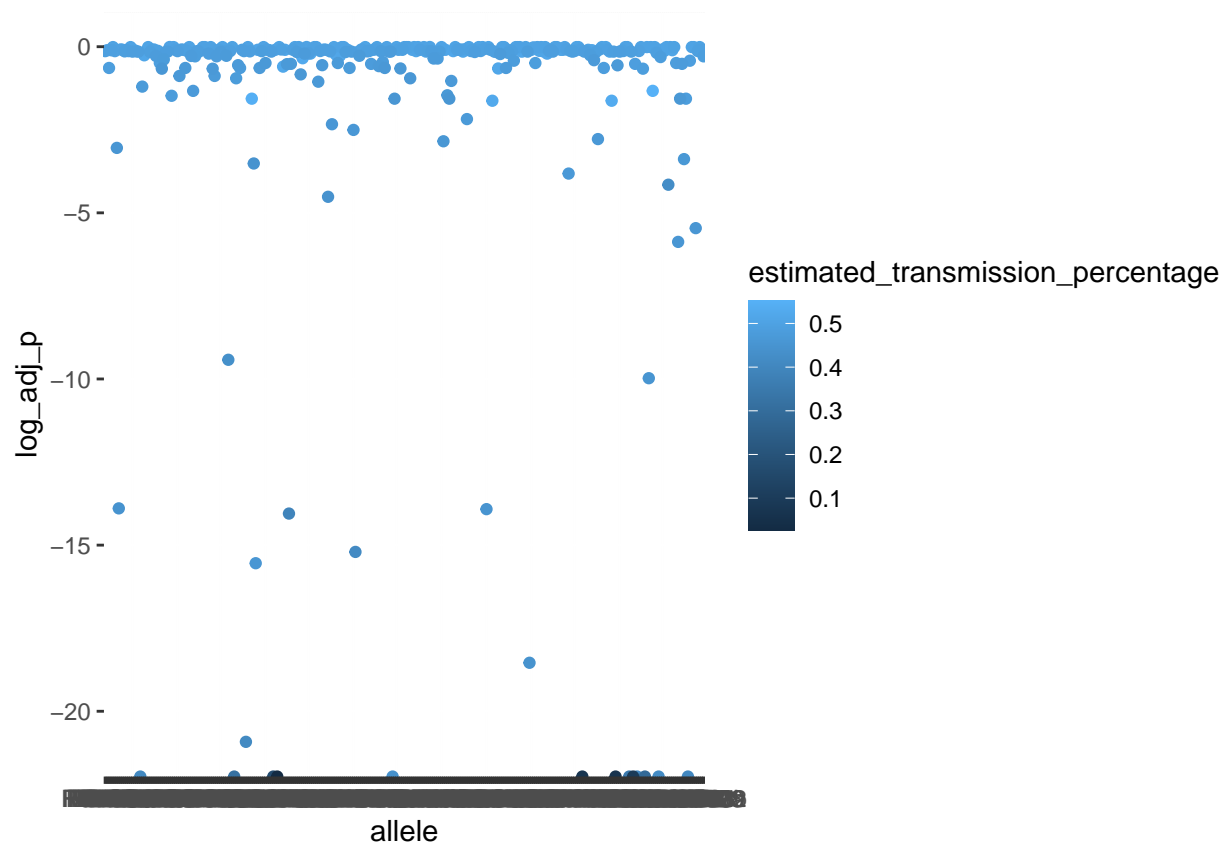
```
with_log_df_new <- ordered_new %>%  
  mutate(log_adj_p = log(adj_p_value)) %>%  
  #only R187B08 had log(adj_p_value) of 0  
  mutate(log_adj_p = if_else(log_adj_p == 0, 1.0*10-15, log_adj_p))
```

```
ordered_log_new <- ggplot(with_log_df_new, mapping = aes(x = allele,  
  y = log_adj_p)) + geom_point()
```

```
ordered_log_new + scale_x_discrete(limits = ordered_new[["allele"]]) + geom_hline(  
  yintercept = log(.05),  
  linetype = "dashed",  
  color = "red")
```

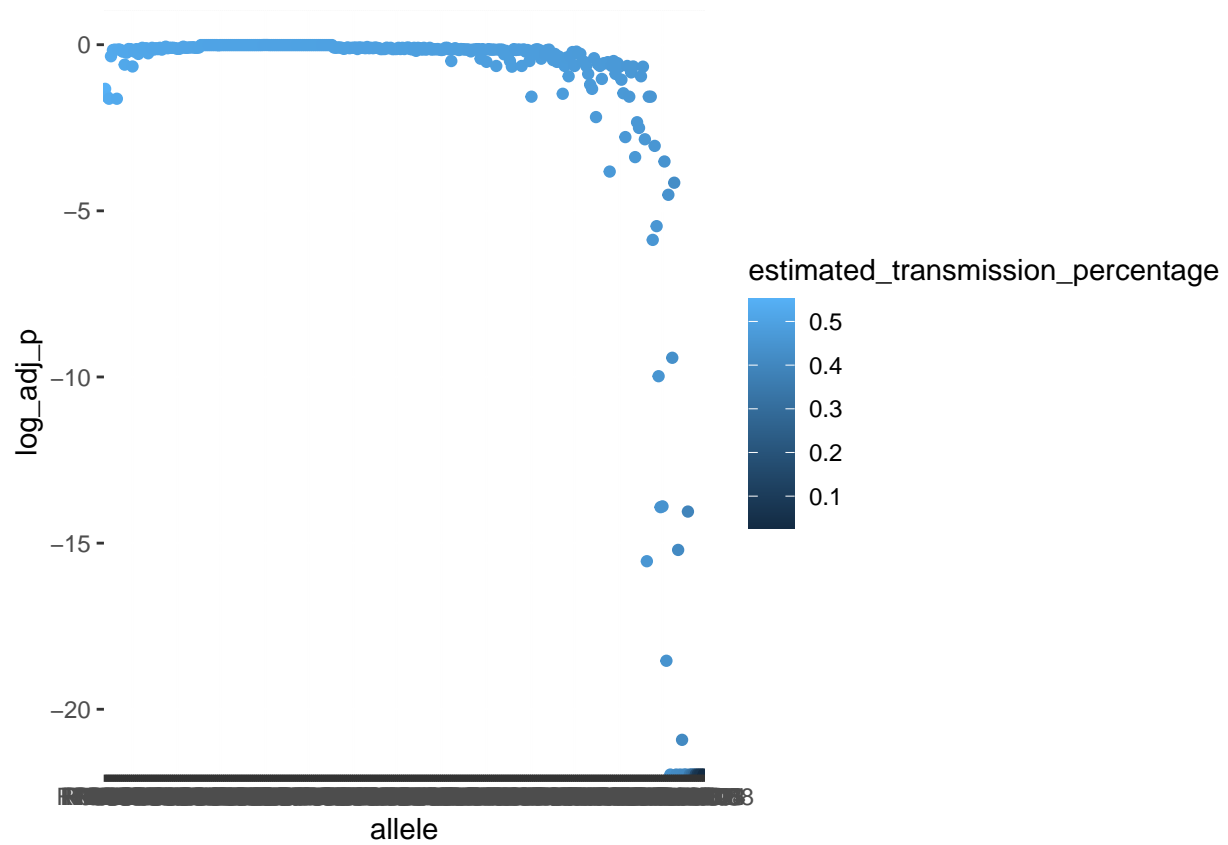


```
ggplot(with_log_df_new, mapping = aes(x = allele,  
  y = log_adj_p,  
  col = estimated_transmission_percentage)) + geom_point()
```



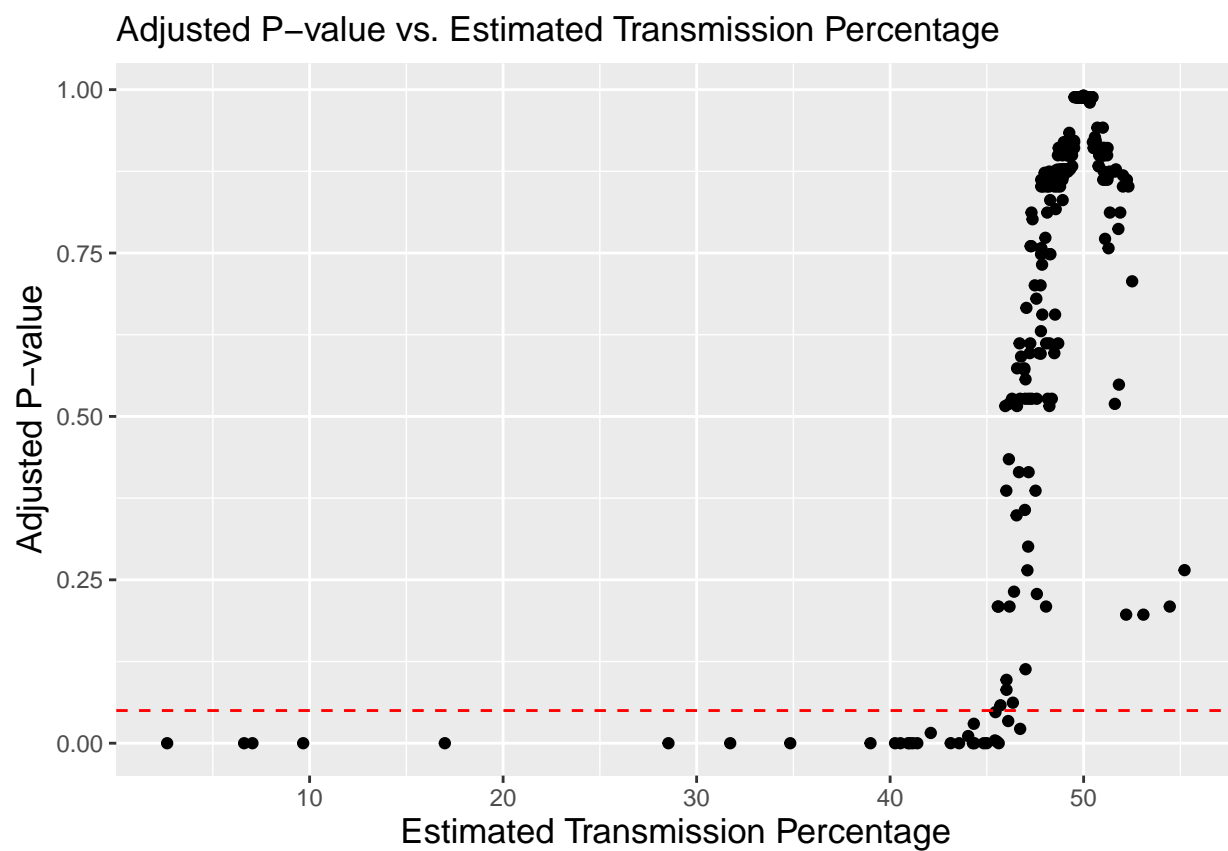
```
logcolor <- ggplot(with_log_df_new, mapping = aes(x = allele,
  y = log_adj_p,
  col = estimated_transmission_percentage)) + geom_point()

logcolor + scale_x_discrete(limits = ordered_new[["allele"]])
```



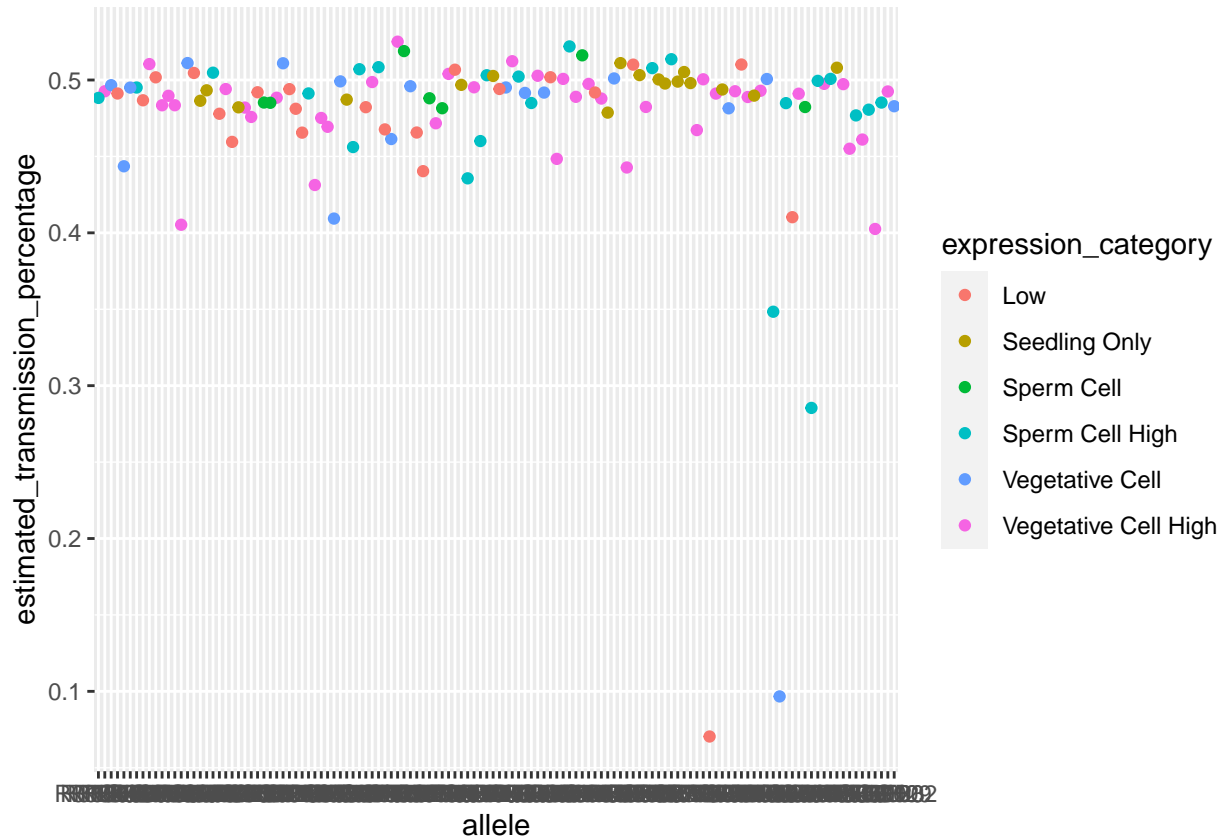
```
ordered_new <- ordered_new %>%
  mutate(perc = estimated_transmission_percentage*100)
```

```
ggplot(ordered_new, mapping = aes(x = perc,
  y = adj_p_value)) + geom_point() + geom_hline(yintercept = 0.05, linetype = "dashed")
```

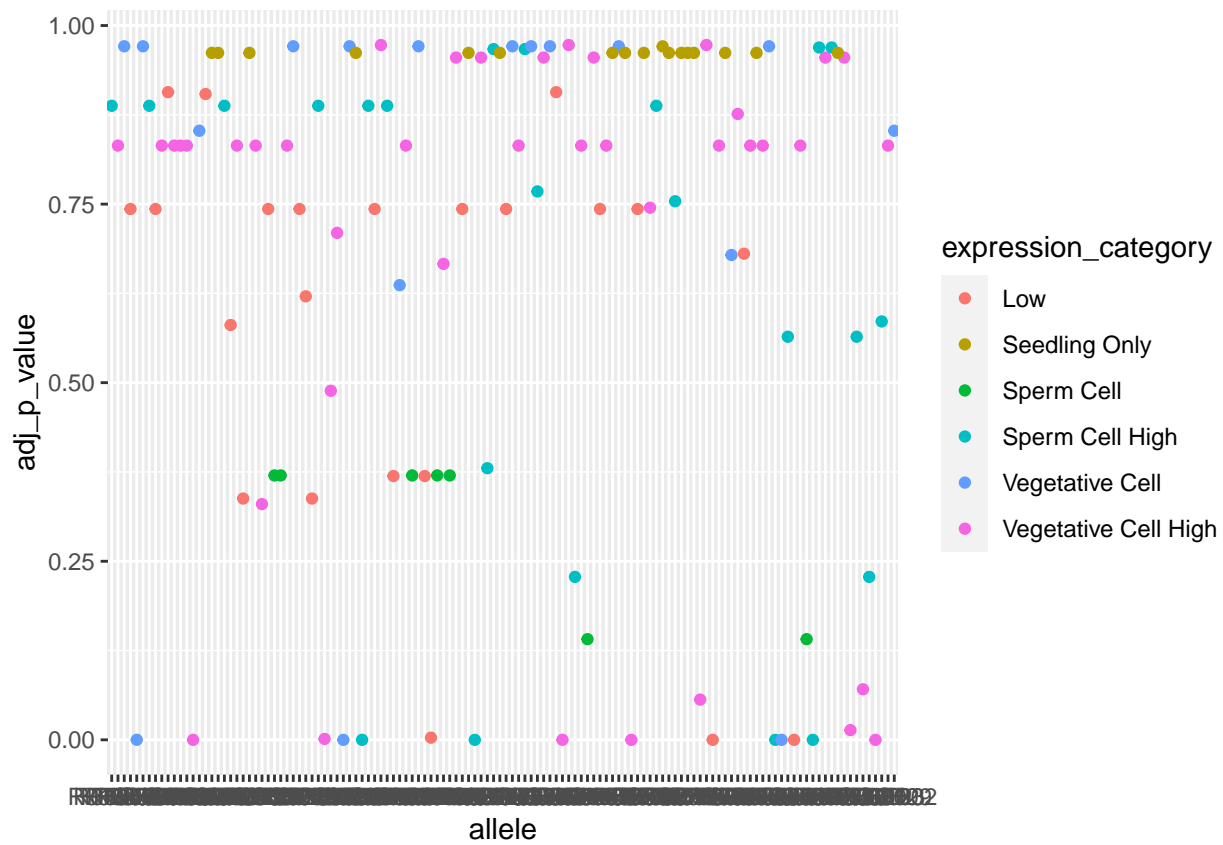


```
#This is for the smaller dataset with PCR_valid
```

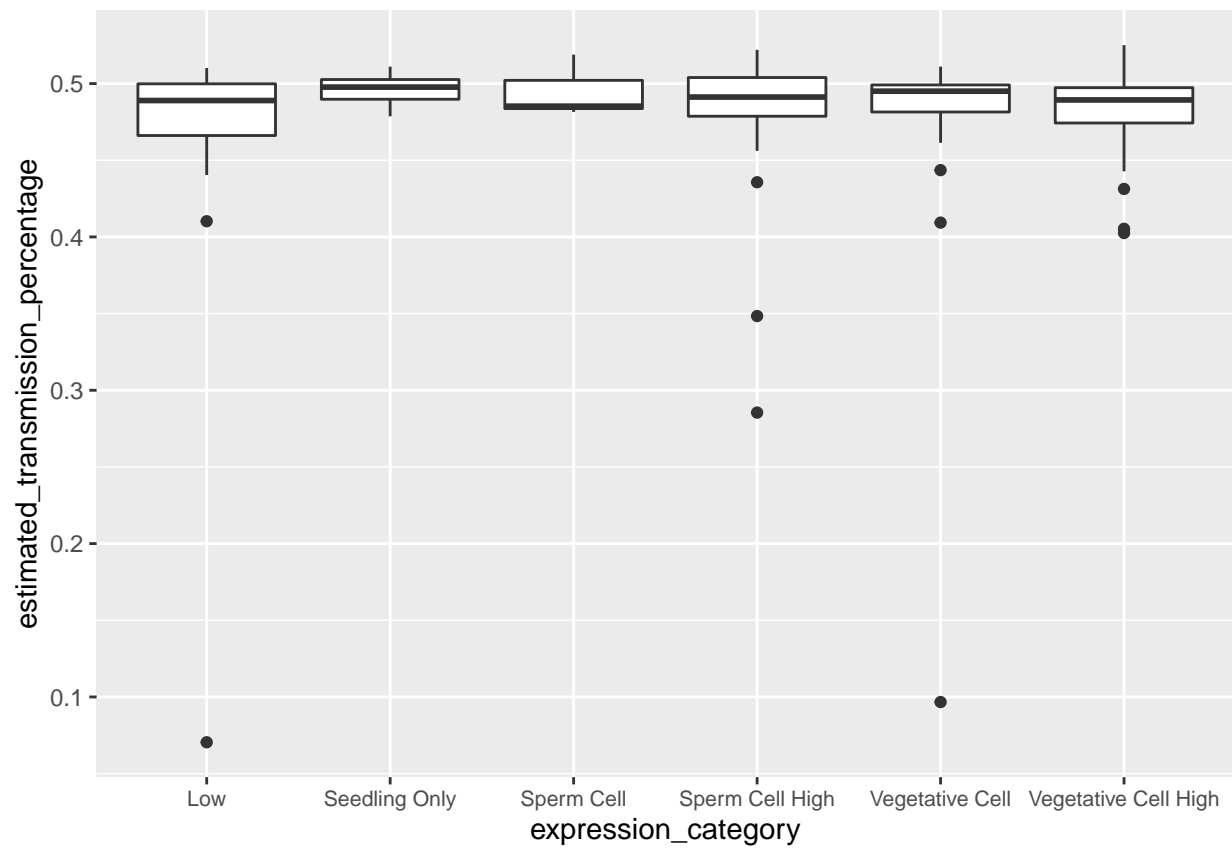
```
ggplot(asn_pcr, mapping = aes(x = allele,  
                              y = estimated_transmission_percentage,  
                              col = expression_category)) + geom_point()
```



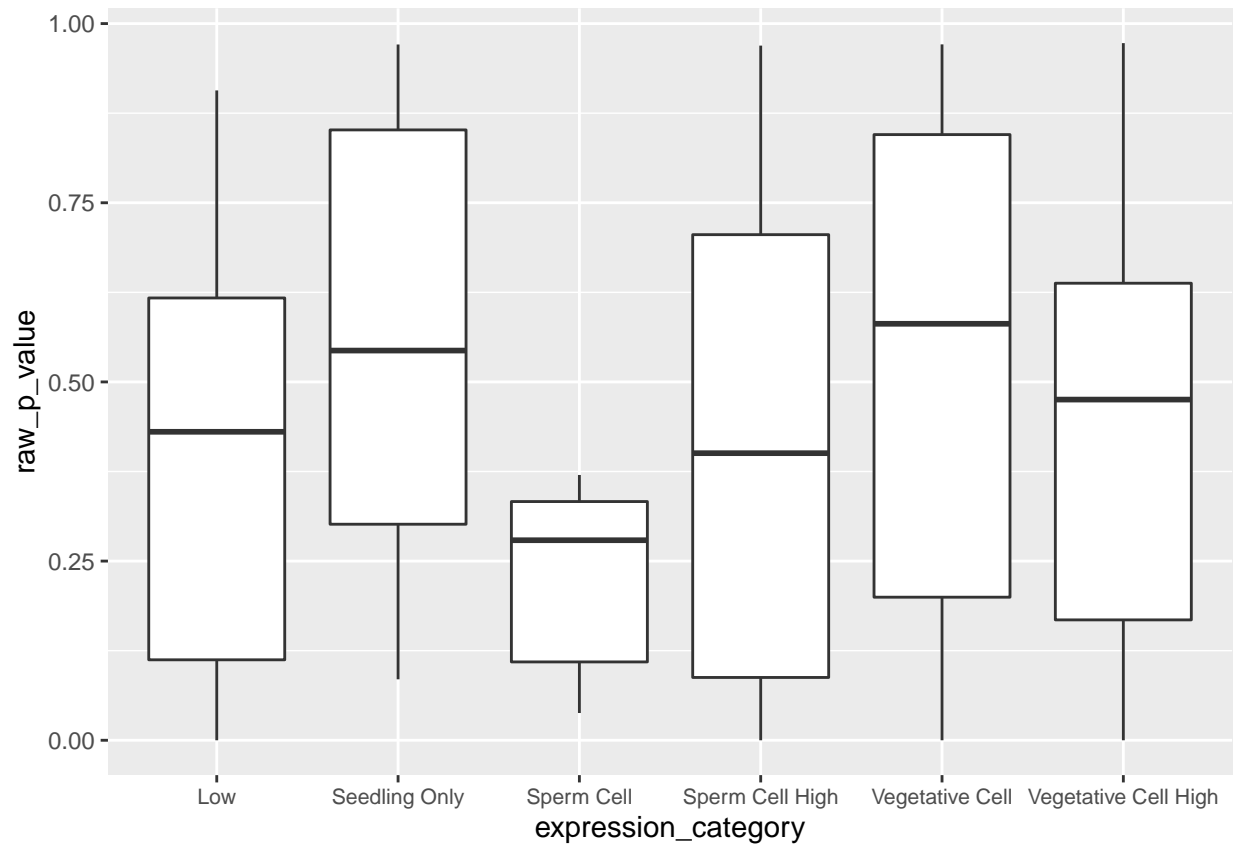
```
ggplot(asn_pcr, mapping = aes(x = allele,  
                              y = adj_p_value,  
                              col = expression_category)) + geom_point()
```



```
ggplot(asn_pcr, mapping = aes(x = expression_category,
                              y = estimated_transmission_percentage)) + geom_boxplot() + theme(axis.text.x = "none")
```

```
ggplot(asn_pcr, mapping = aes(x = expression_category,
                              y = raw_p_value)) + geom_boxplot() + theme(axis.text.x = element_text(size = 8))
```



#ADDITIONAL REQUESTS FROM JOHN (02/28)

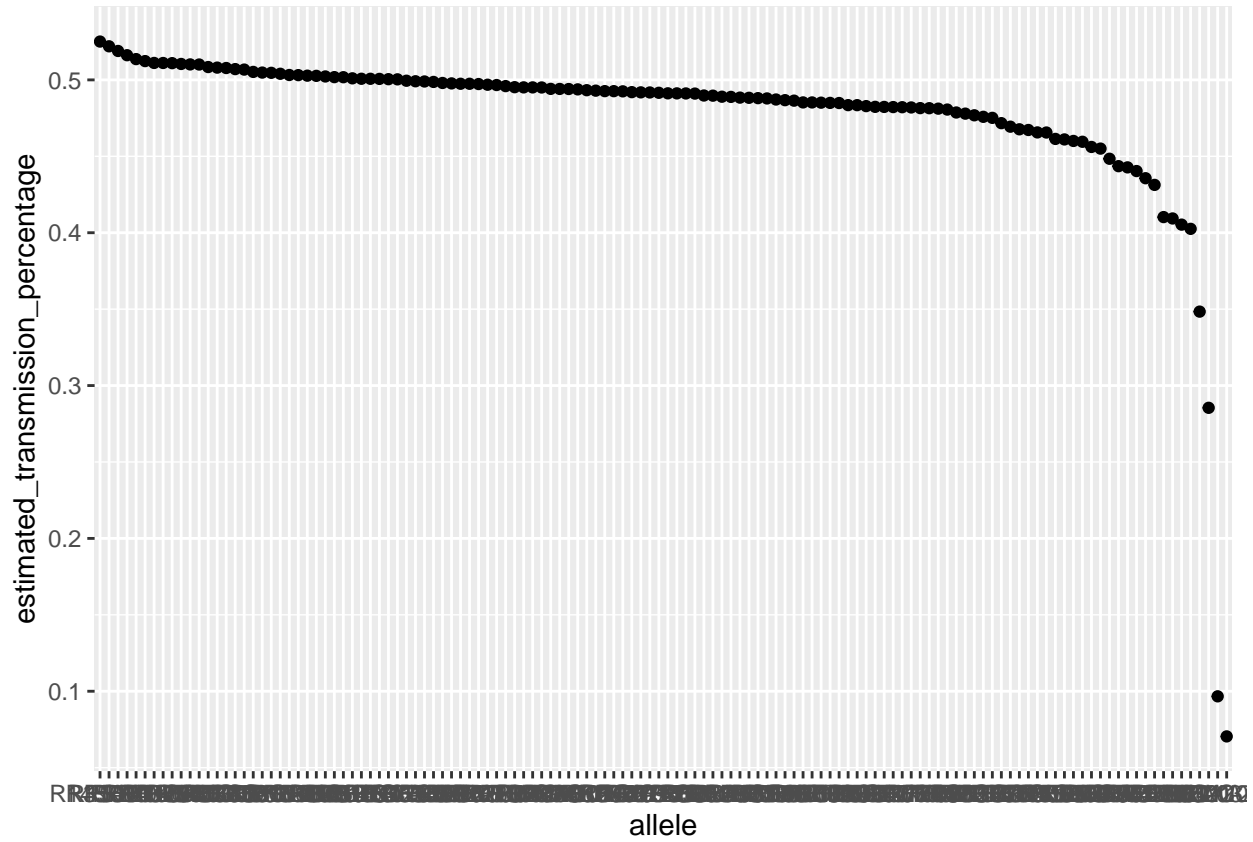
```
ordered_df <- asn_pcr %>%  
  arrange(desc(estimated_transmission_percentage))
```

```
write.table(ordered_df, file = "2023_cleanPCR_analysis_ordered.tsv", sep = "\t", row.names=FALSE)
```

GRAPHS

```
ordered_plot <- ggplot(ordered_df, mapping = aes(x = allele,
                                                  y = estimated_transmission_percentage)) + geom_point()

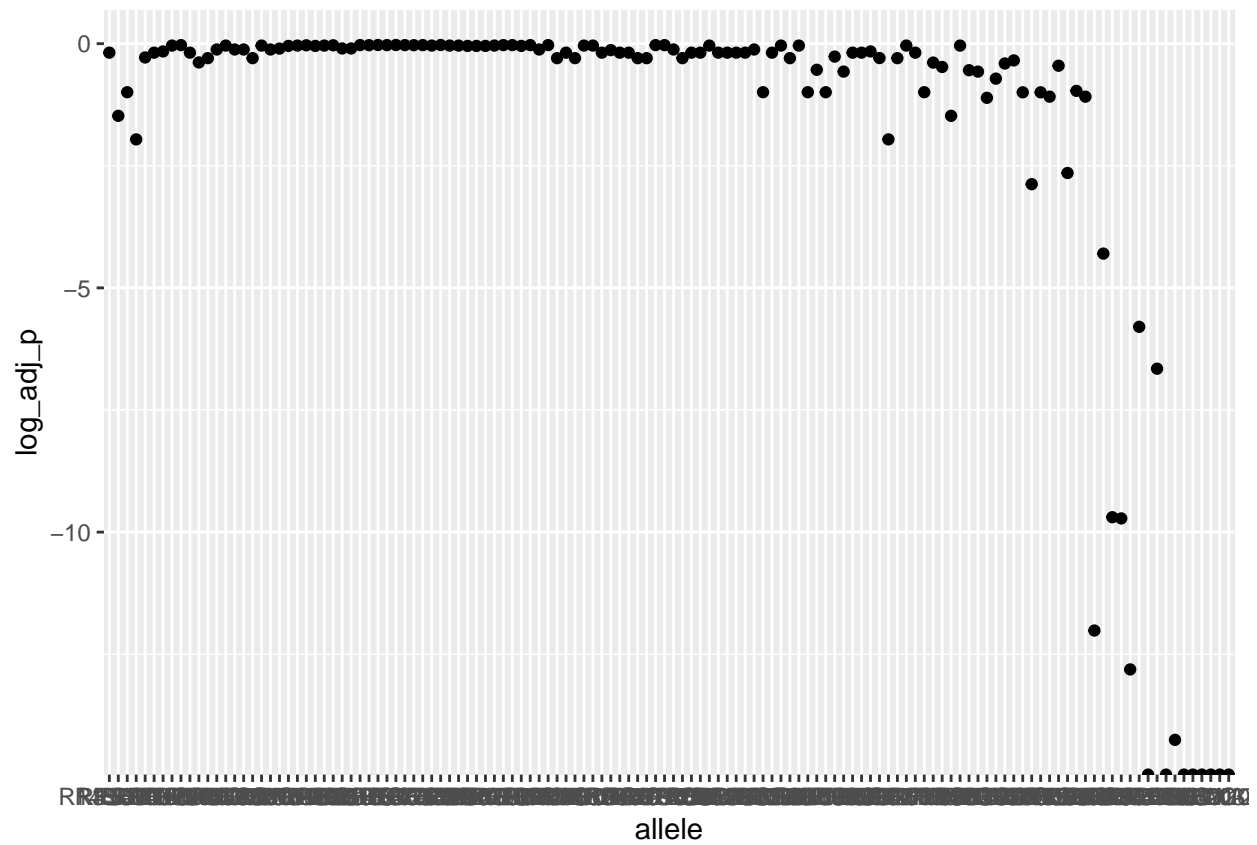
ordered_plot + scale_x_discrete(limits = ordered_df[["allele"]])
```



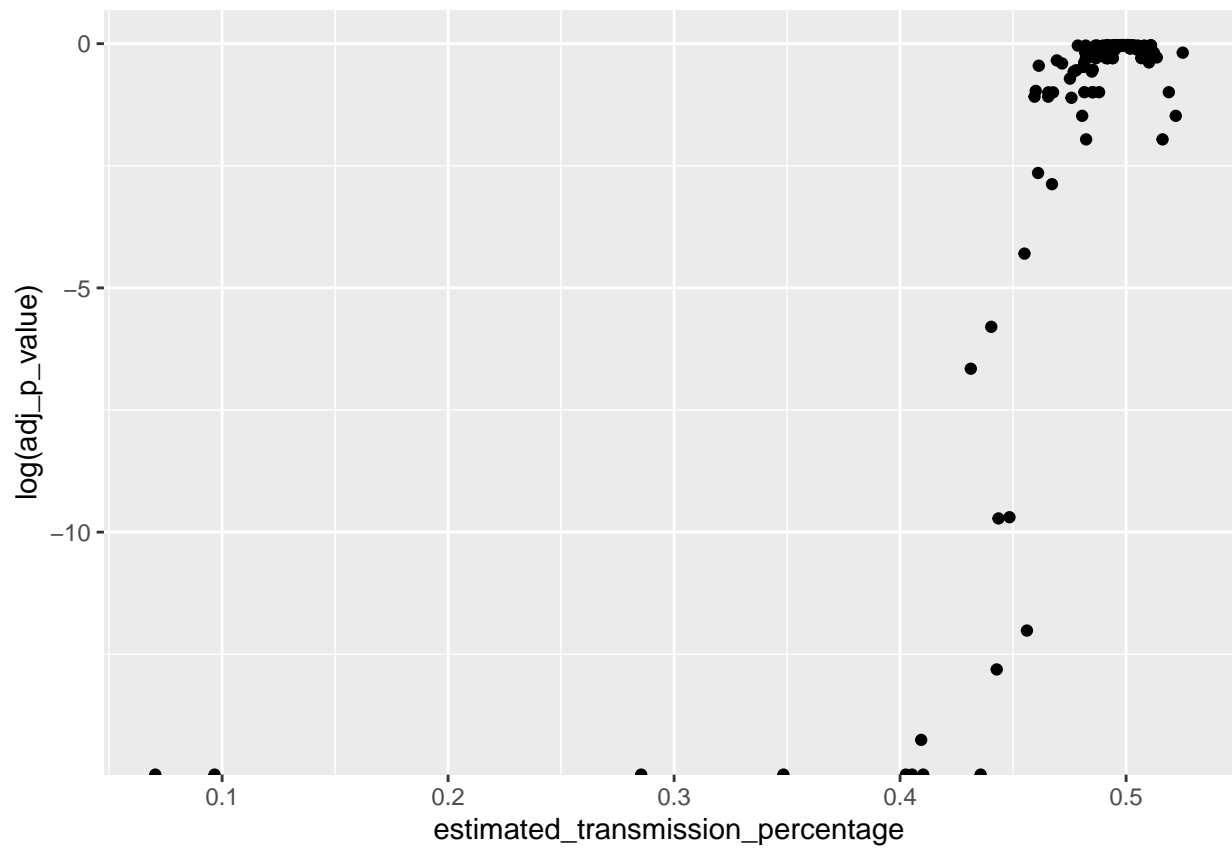
```
with_log_df <- ordered_df %>%
  mutate(log_adj_p = log(adj_p_value)) %>%
  #only R187B08 had log(adj_p_value) of 0
  mutate(log_adj_p = if_else(log_adj_p == 0, 1.0*10(-15), log_adj_p))
```

```
ordered_log <- ggplot(with_log_df, mapping = aes(x = allele,
                                                  y = log_adj_p)) + geom_point()

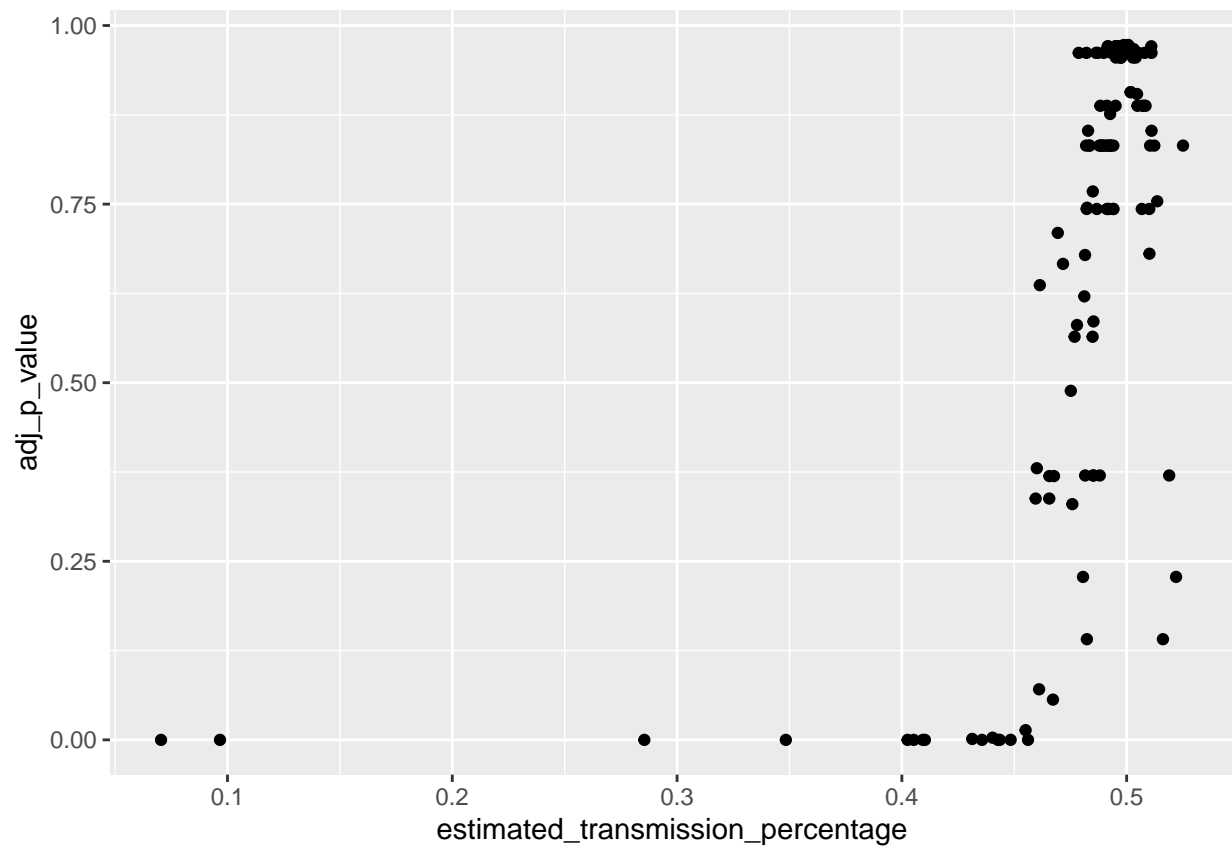
ordered_log + scale_x_discrete(limits = ordered_df[["allele"]])
```



```
ggplot(with_log_df, mapping = aes(x = allele,
                                  y = log_adj_p,
                                  col = estimated_transmission_percentage)) + geom_point()
```

```
ggplot(asn_pcr, mapping = aes(x = estimated_transmission_percentage,  
                               y = adj_p_value)) + geom_point()
```

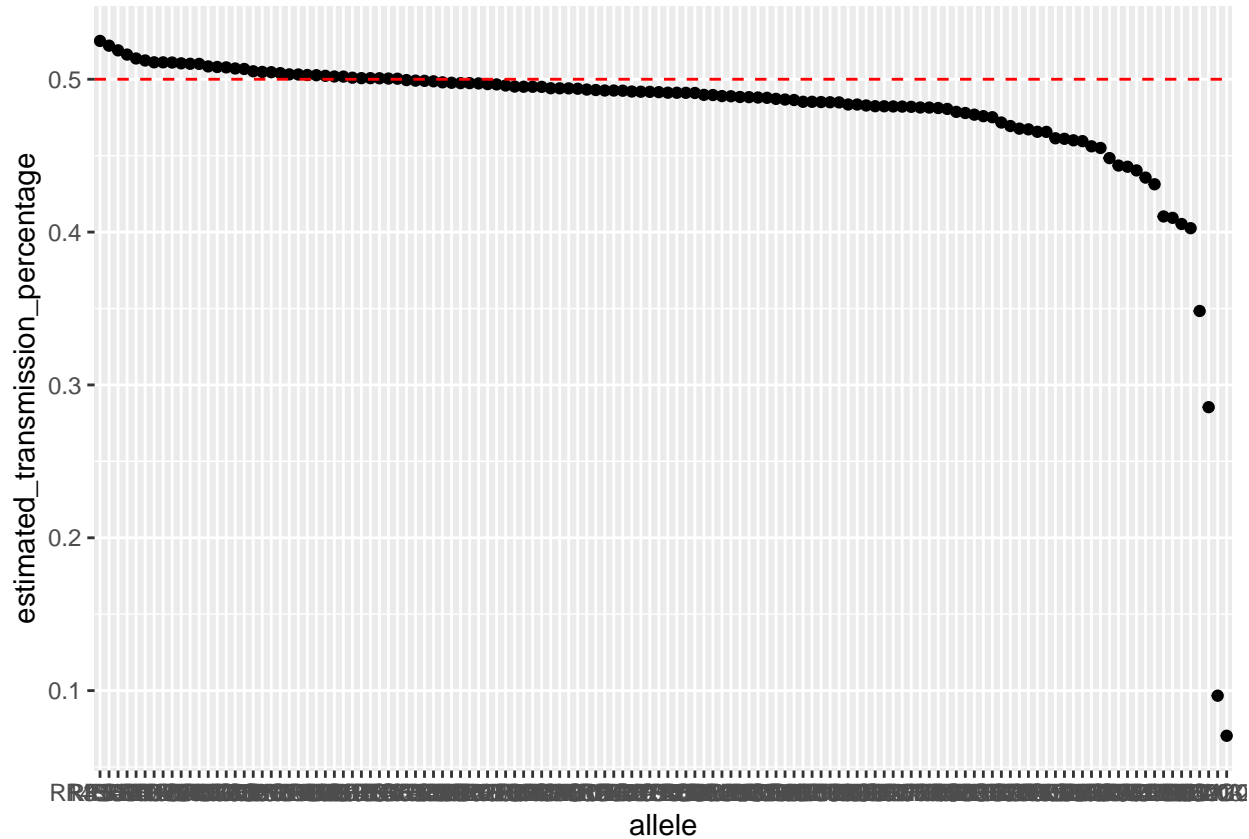


#NEW GRAPHICS...

```
ordered_new <- ordered_df %>%  
  arrange(desc(estimated_transmission_percentage))
```

```
ordered_plot_new <- ggplot(ordered_new, mapping = aes(x = allele,  
  y = estimated_transmission_percentage )) + geom_point()
```

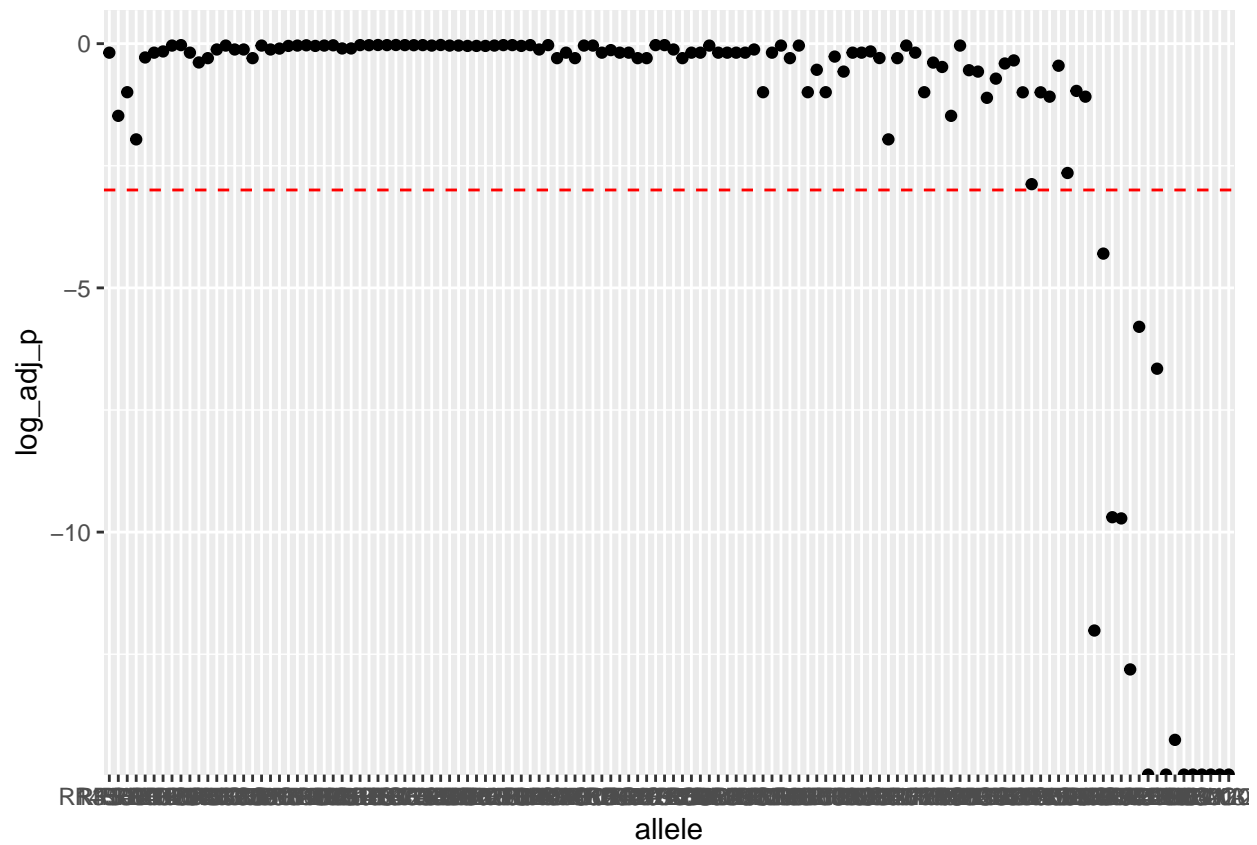
```
ordered_plot_new + scale_x_discrete(limits = ordered_new[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed", color = "red")
```



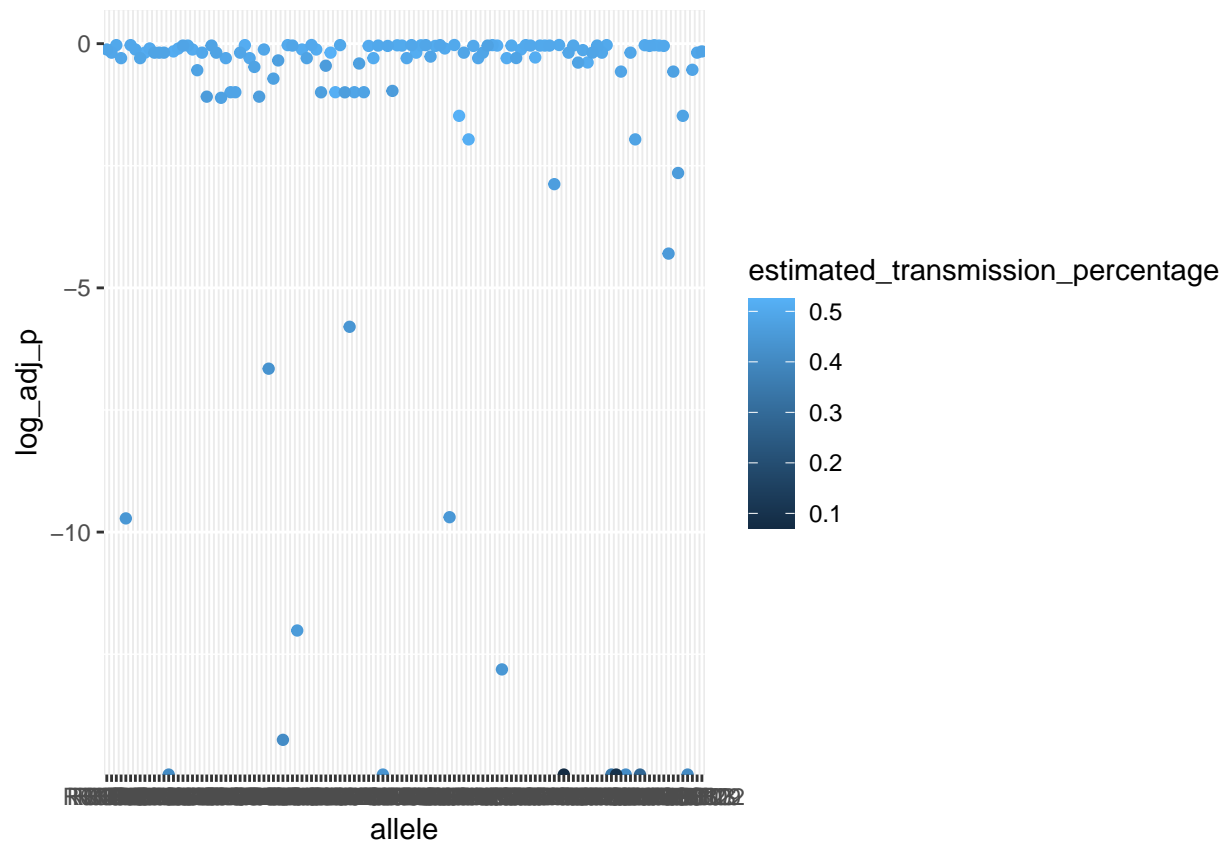
```
with_log_df_new <- ordered_new %>%  
  mutate(log_adj_p = log(adj_p_value)) %>%  
  #only R187B08 had log(adj_p_value) of 0  
  mutate(log_adj_p = if_else(log_adj_p == 0, 1.0*10-15, log_adj_p))
```

```
ordered_log_new <- ggplot(with_log_df_new, mapping = aes(x = allele,  
  y = log_adj_p)) + geom_point()
```

```
ordered_log_new + scale_x_discrete(limits = ordered_new[["allele"]]) + geom_hline(  
  yintercept = log(.05),  
  linetype = "dashed",  
  color = "red")
```

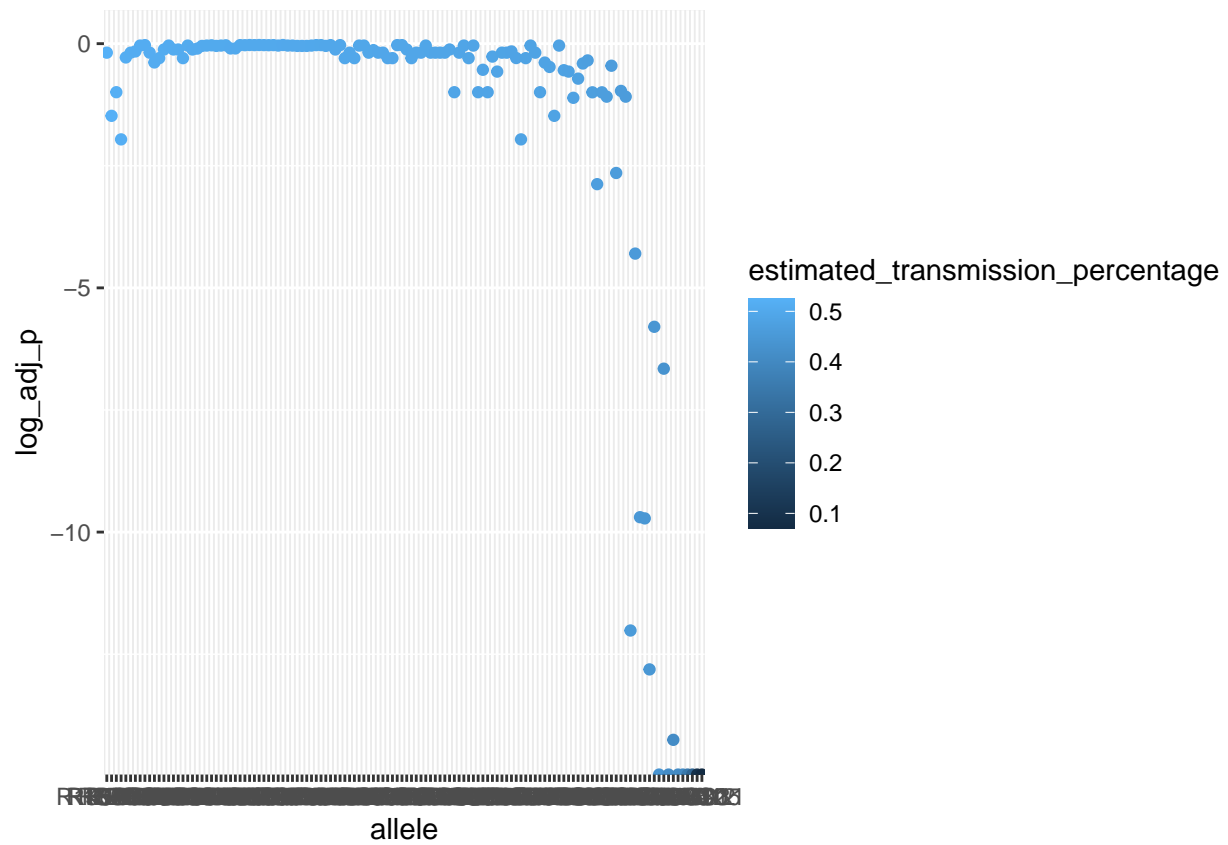


```
ggplot(with_log_df_new, mapping = aes(x = allele,
  y = log_adj_p,
  col = estimated_transmission_percentage)) + geom_point()
```

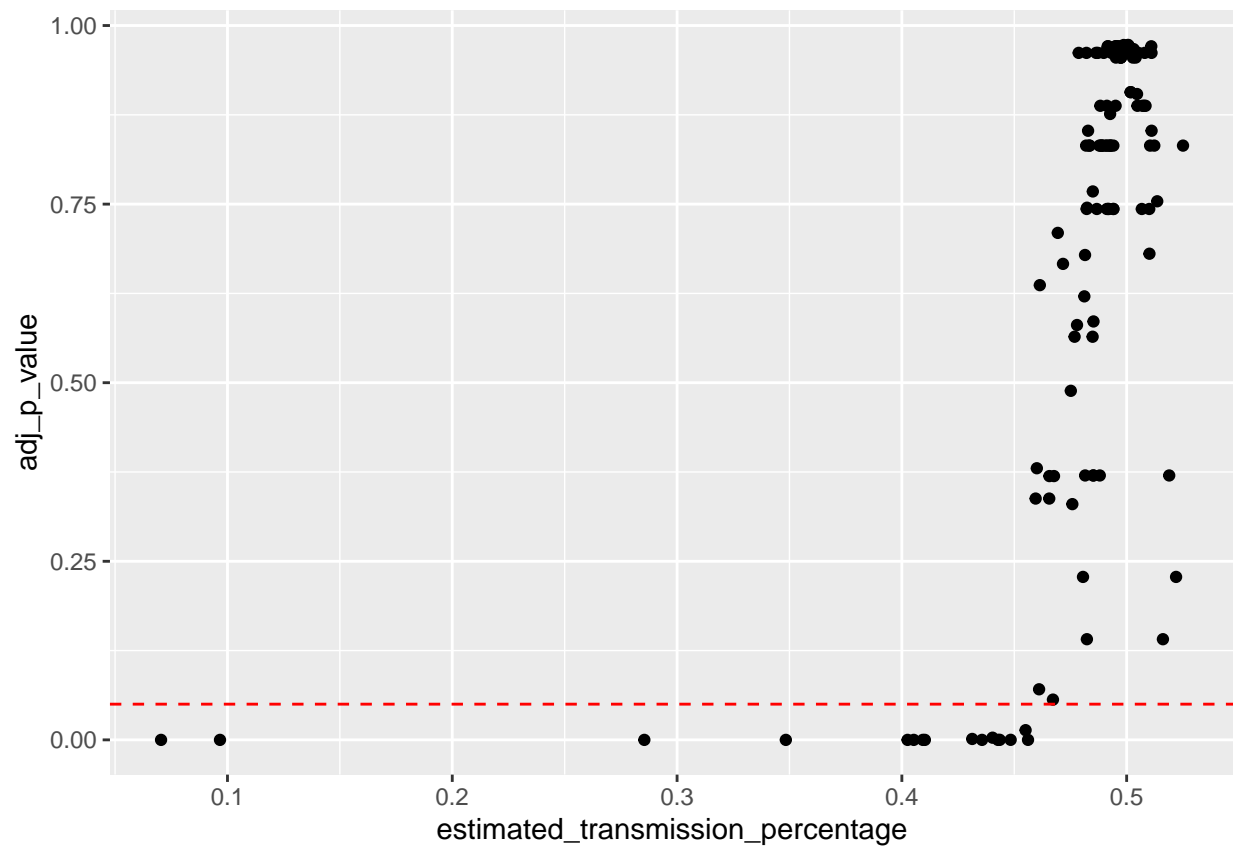


```
logcolor <- ggplot(with_log_df_new, mapping = aes(x = allele,
  y = log_adj_p,
  col = estimated_transmission_percentage)) + geom_point()

logcolor + scale_x_discrete(limits = ordered_new[["allele"]])
```

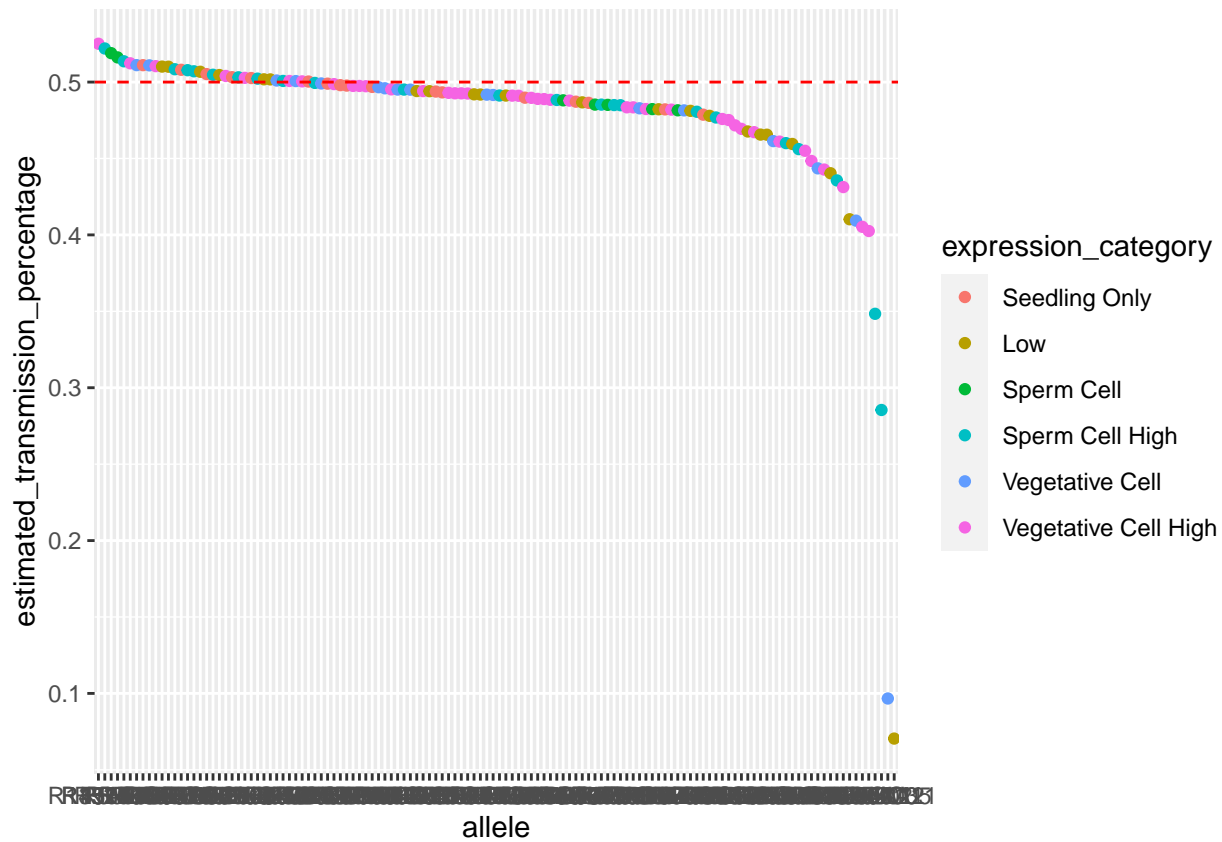


```
ggplot(ordered_new, mapping = aes(x = estimated_transmission_percentage,
                                   y = adj_p_value)) + geom_point() + geom_hline(yintercept = 0.05, linetype = "dashed")
```



#Additional Request from John 03/09

```
ordered_new$expression_category <- factor(ordered_new$expression_category, levels = c("Seedling Only", "Low", "Sperm Cell", "Sperm Cell High", "Vegetative Cell", "Vegetative Cell High"))
ordered_plot_new <- ggplot(ordered_new, mapping = aes(x = allele,
  y = estimated_transmission_percentage, color = expression_category )) + geom_point()
ordered_plot_new + scale_x_discrete(limits = ordered_new[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed")
```



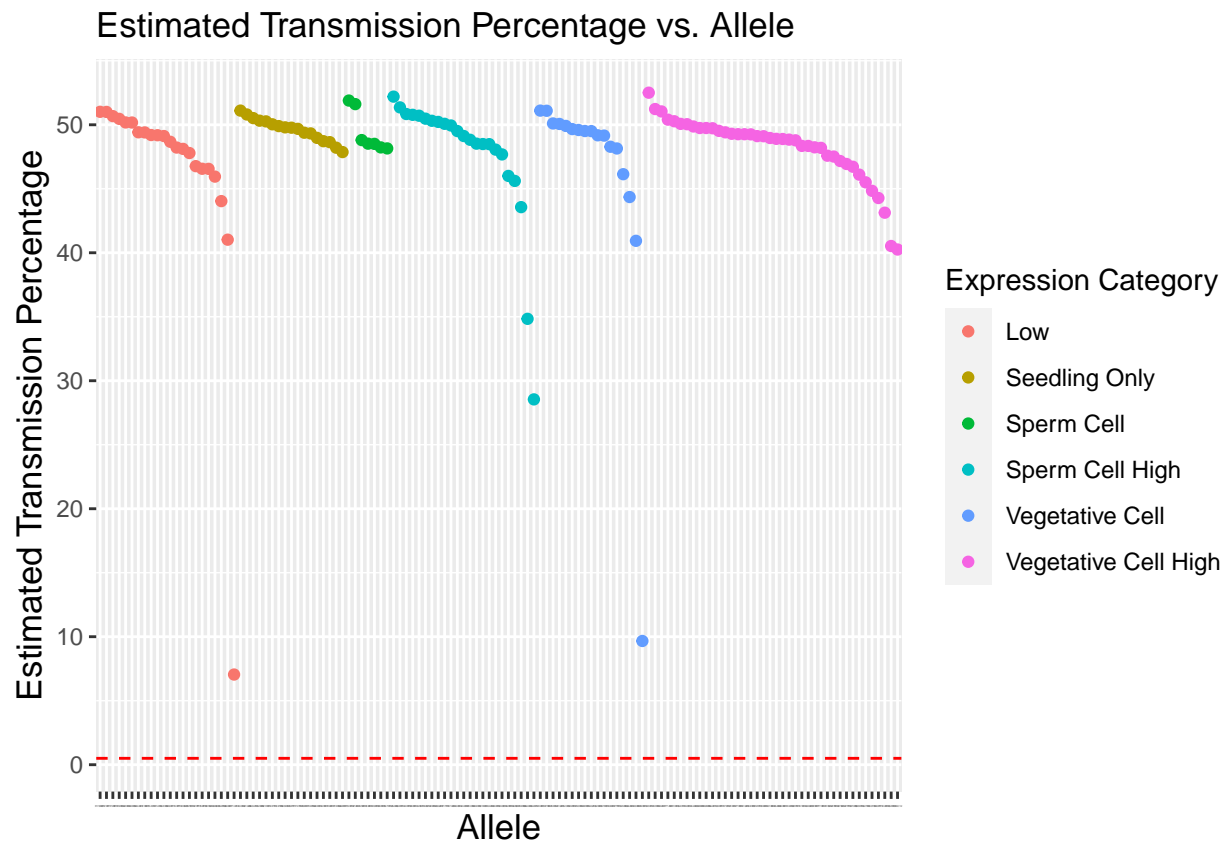
```
grouped_new <- ordered_df %>%
  arrange(expression_category)

#below does not work...?
#grouped_new <- ordered_df %>%
  #arrange(factor(grouped_new$expression_category, levels = c("Seedling Only", "Low", "Sperm Cell", "Sperm Cell High", "Vegetative Cell", "Vegetative Cell High")))
```

```
grouped_new <- grouped_new %>%
  mutate(perc = estimated_transmission_percentage*100)
```

```
#grouped_new$expression_category <- factor(grouped_new$expression_category, levels = c("Seedling Only", "Low", "Sperm Cell", "Sperm Cell High", "Vegetative Cell", "Vegetative Cell High"))
grouped_plot_new <- ggplot(grouped_new, mapping = aes(x = allele,
  y = perc, color = expression_category )) + geom_point()

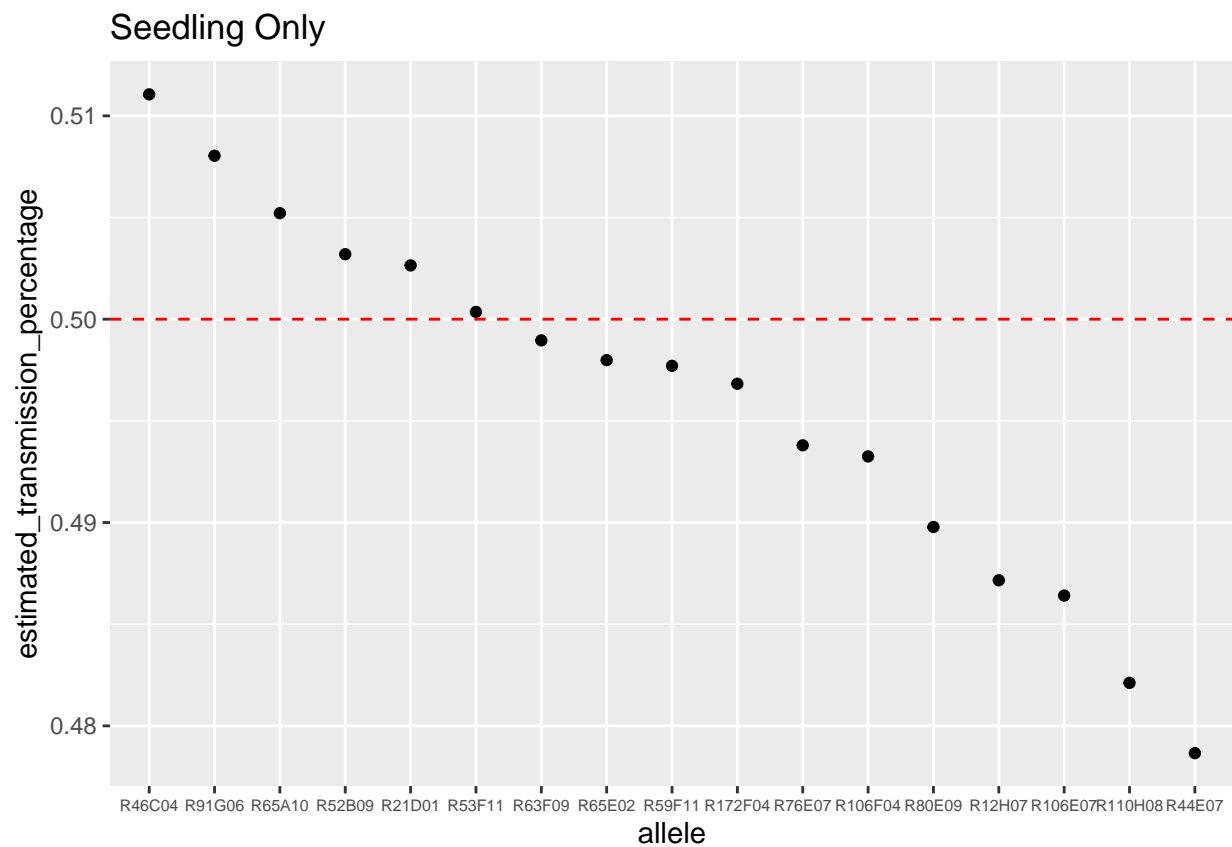
#grouped_plot_new
grouped_plot_new + scale_x_discrete(limits = grouped_new[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed")
```



```
seedling_df <- ordered_new %>%
  filter(expression_category == "Seedling Only")

seedling_plot <- ggplot(seedling_df, mapping = aes(x = allele,
  y = estimated_transmission_percentage)) + geom_point()

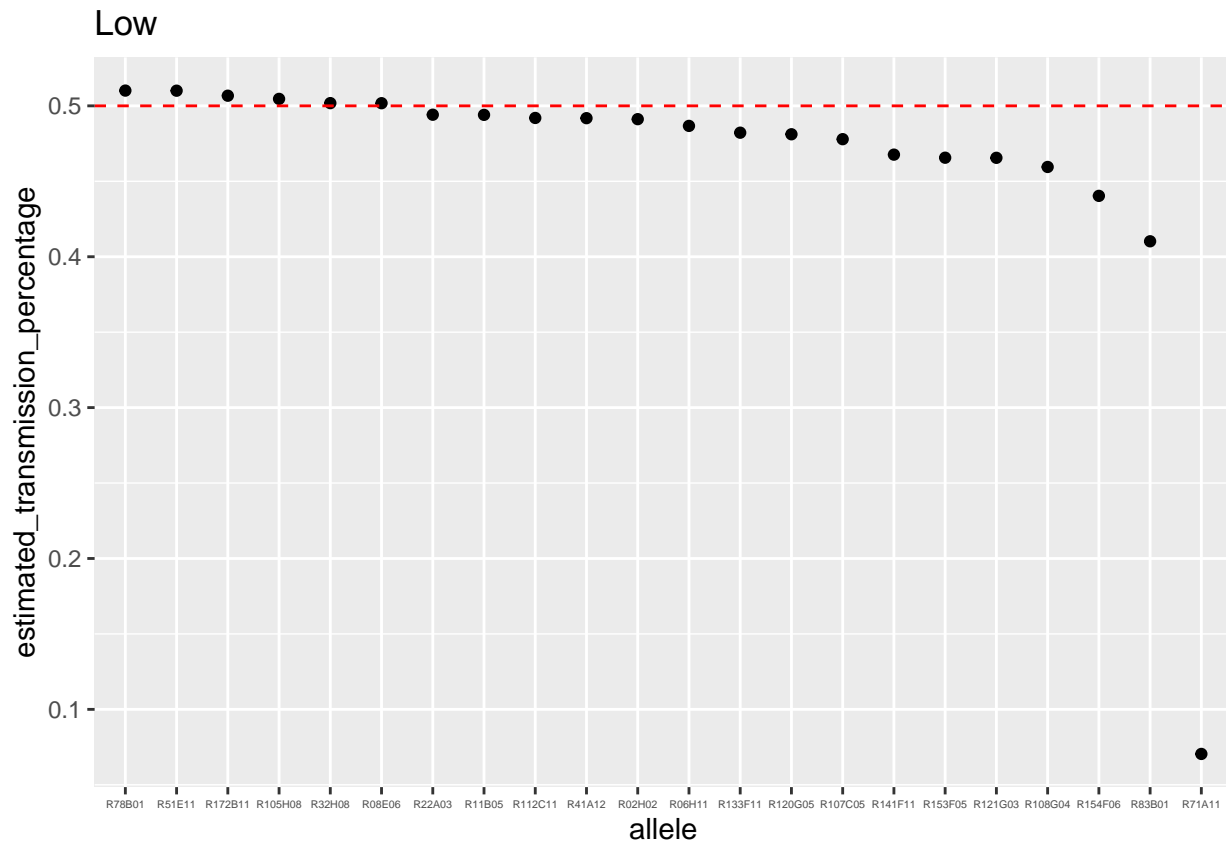
seedling_plot + scale_x_discrete(limits = seedling_df[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed")
```



```
low_df <- ordered_new %>%
  filter(expression_category == "Low")

low_plot <- ggplot(low_df, mapping = aes(x = allele,
                                          y = estimated_transmission_percentage)) + geom_point()

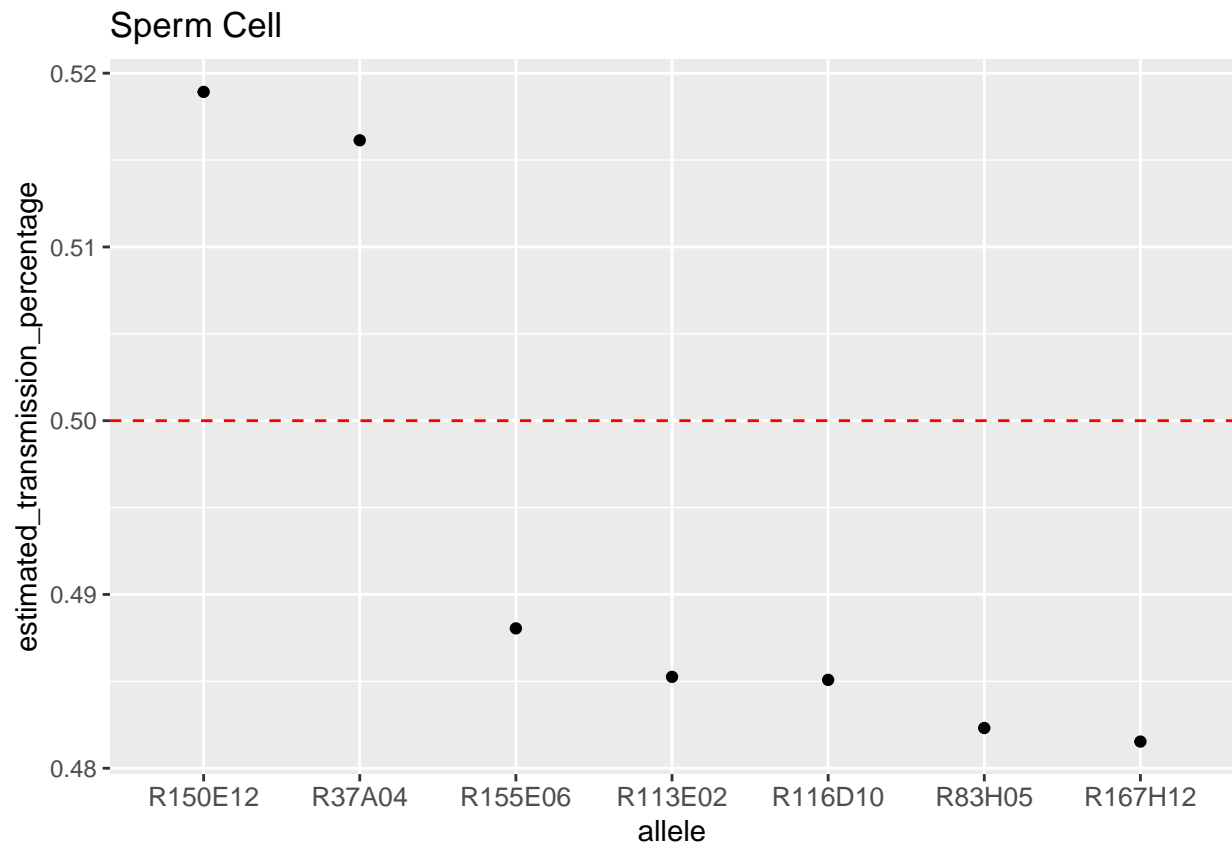
low_plot + scale_x_discrete(limits = low_df[["allele"]]) + geom_hline(yintercept = .5, linetype = "dash")
```

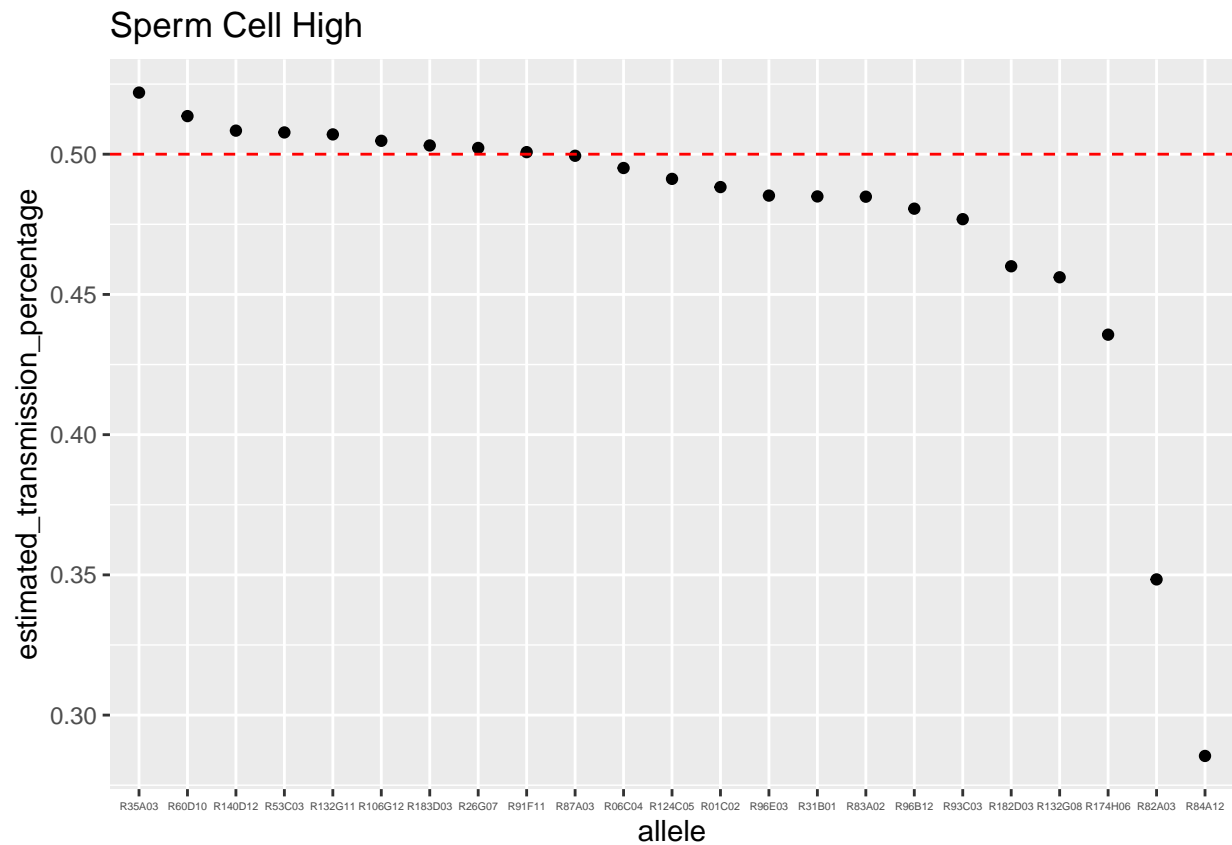
```
sperm_df <- ordered_new %>%
  filter(expression_category == "Sperm Cell")

sperm_plot <- ggplot(sperm_df, mapping = aes(x = allele,
  y = estimated_transmission_percentage)) + geom_point()

sperm_plot + scale_x_discrete(limits = sperm_df[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed")
```



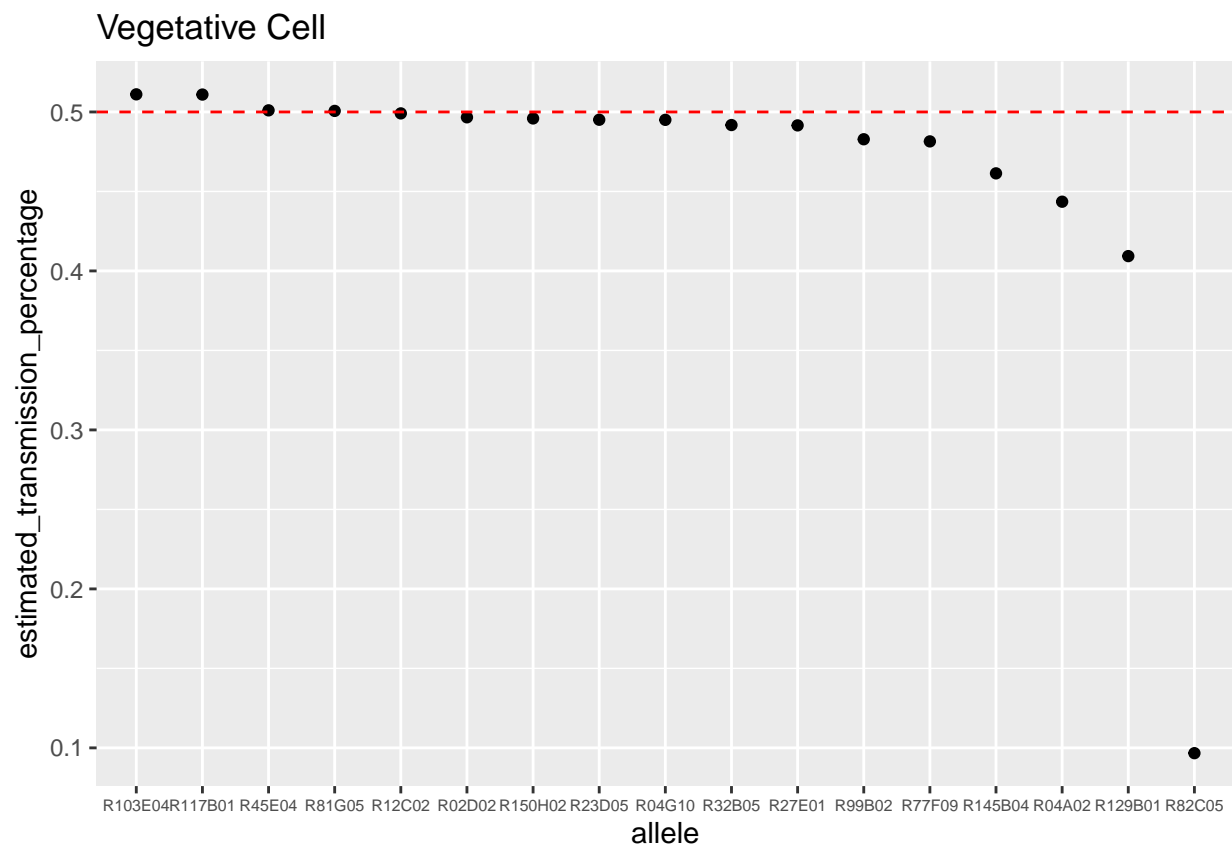
```
shigh_df <- ordered_new %>%  
  filter(expression_category == "Sperm Cell High")  
  
shigh_plot <- ggplot(shigh_df, mapping = aes(x = allele,  
                                              y = estimated_transmission_percentage)) + geom_point()  
  
shigh_plot + scale_x_discrete(limits = shigh_df[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed")
```



```
veg_df <- ordered_new %>%
  filter(expression_category == "Vegetative Cell")

veg_plot <- ggplot(veg_df, mapping = aes(x = allele,
                                          y = estimated_transmission_percentage)) + geom_point()

veg_plot + scale_x_discrete(limits = veg_df[["allele"]]) + geom_hline(yintercept = .5, linetype = "dashed")
```



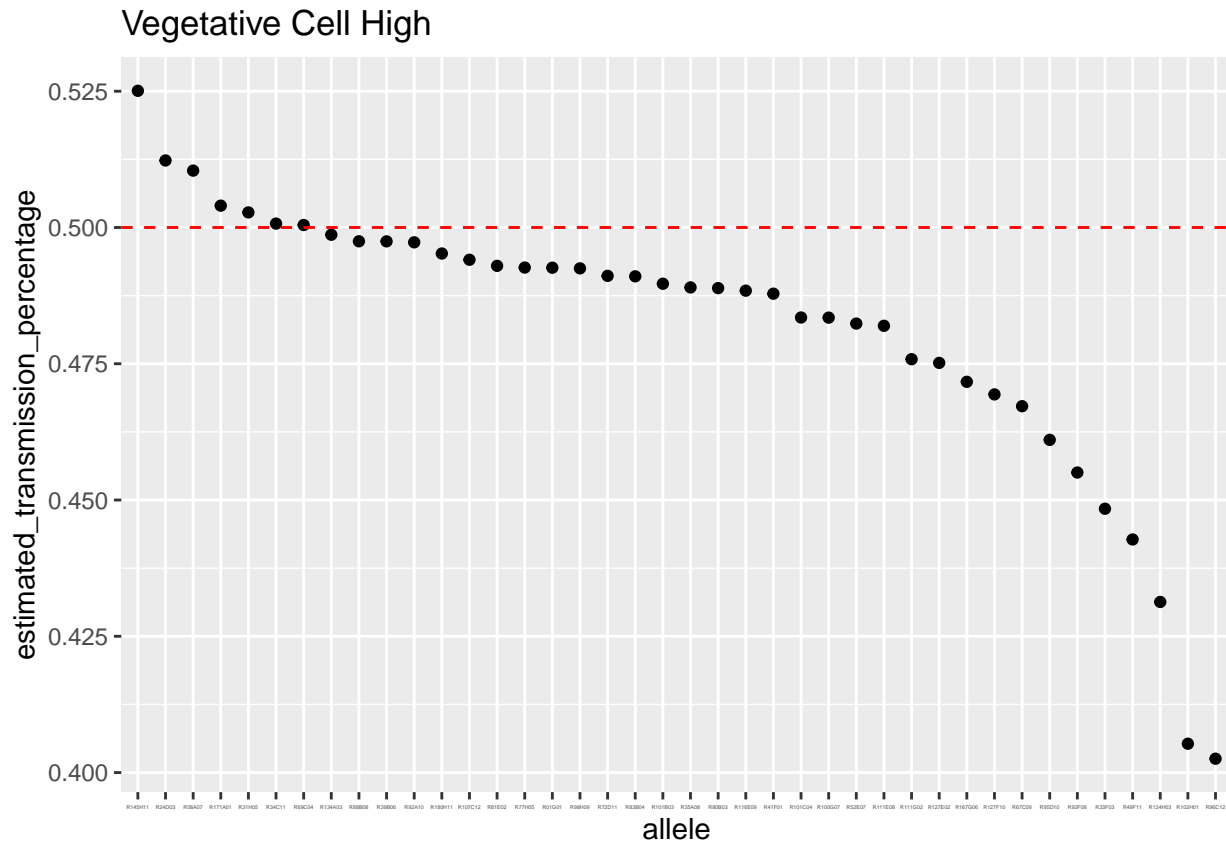
```

vhigh_df <- ordered_new %>%
  filter(expression_category == "Vegetative Cell High")

vhigh_plot <- ggplot(vhigh_df, mapping = aes(x = allele,
                                              y = estimated_transmission_percentage)) + geom_point()

vhigh_plot + scale_x_discrete(limits = vhigh_df[["allele"]]) + geom_hline(yintercept = .5, linetype = "

```



```
ordered_new %>%
  group_by(expression_category) %>%
  summarize(count = n())
```

```
## # A tibble: 6 x 2
##   expression_category count
##   <fct>              <int>
## 1 Seedling Only      17
## 2 Low                22
## 3 Sperm Cell         7
## 4 Sperm Cell High    23
## 5 Vegetative Cell    17
## 6 Vegetative Cell High 40
```

```
count_df <- ordered_df %>%
  mutate(expression_category = if_else(expression_category == "Seedling Only", "Seedling Only (17)", expression_category))
  mutate(expression_category = if_else(expression_category == "Low", "Low (22)", expression_category))
  mutate(expression_category = if_else(expression_category == "Sperm Cell", "Sperm Cell (7)", expression_category))
  mutate(expression_category = if_else(expression_category == "Sperm Cell High", "Sperm Cell High (23)", expression_category))
  mutate(expression_category = if_else(expression_category == "Vegetative Cell", "Vegetative Cell (17)", expression_category))
  mutate(expression_category = if_else(expression_category == "Vegetative Cell High", "Vegetative Cell High (40)", expression_category))
```

```
ggplot(count_df, mapping = aes(x = expression_category,
                               y = estimated_transmission_percentage)) + geom_boxplot() + theme(axis.text.x = element_text(angle = 45))
```

