Computing Serostatus with External Cutoffs

2025-05-22

Introduction

The purpose of this document is to show how specific samples are determined to seropositive or seronegative using external information about a particular pathogen and antigen (e.g., an internationally recognized threshold of protective immunity). This document then describes how to aggregate serostatus information about each sample to calculate a population level seroprevalence. Upon completing this lab you should be able to:

- Read in control data if available
- Visualize MFI distributions on the appropriate scale
- Calculate serostatus for each sample using a predetermined cutoff
- Calculate a population seroprevalence.

General housekeeping

Before we start, let's navigate to the appropriate working directory. You can accomplish this by navigating to the "Session" tab of Rstudio, and choosing "Set Working Directory" –> "Choose Directory" and using your file browser to navigate to the Data folder within the seroanalytics_workshop folder. Alternatively, you can modify the code below as appropriate for your files to get to the Data folder in the seroanalytics_workshop folder.

```
#setwd("~seroanalytics_workshop/Data/)
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message=FALSE)
source("/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofFlorida/Desktop/Research/Bead_serology
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.
```

Reading in data

Read in the example data using the code below. You will notice these data have a slightly different structure than the csv files we have previously been working with. This is because oftentimes serial dilutions (standard curves), and control samples with knwon prior exposure status are often run on different plates than the cohort samples. This can often be the case even when there are a small number of control wells on the plates with cohort samples. This can be because a larger volume of control samples are required to establish a cutoff therefore it is not feasible to run these two sets of samples on the same plates.

```
#standard curve data
standards_data <- read.csv("/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofFlorida/Desktop/R
head(standards_data)
```

```
##   Plate Sample     Antigen cutoff_iu sim_dilution dilution_iu_ml  SNAP   WNV
## 1     1     P1 pan control        NA          100             NA 21548 27697
## 2     1     P2 pan control        NA          200             NA 15331 22863
## 3     1     P3 pan control        NA          400             NA  9583 22630
## 4     1     P4 pan control        NA          800             NA  5084 16681
## 5     1     P5 pan control        NA         1600             NA  2383  8402
## 6     1     P6 pan control        NA         3200             NA  1133  6145
##       YF   JE3  ZIKA  DENV CHIKV GLURPR2   CSP PfAMA1 PfMSP119 WRUV WMEV  X
## 1 17621 20810 26005 18080 21657   26175 18499  21727    16077   NA   NA NA
## 2 13256 15570 23214 12473 14633   22562 12395  14309    14156   NA   NA NA
## 3  9328  9745 22922  8663  9928   21772  9071   9610    13917   NA   NA NA
## 4  4849  5580 17512  4690  5553   17433  4708   5252     9837   NA   NA NA
## 5  2131  2451  8564  2011  2482    8787  2108   2607     5064   NA   NA NA
## 6   930  1156  6042   972  1209    6179   989   1193     3824   NA   NA NA
```

```
#data on control samples with known prior exposure (positive or negative)
control_data <- read.csv("/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofFlorida/Desktop/Rese
head(control_data)
```

```
##   antigen  pos_neg mfi
## 1     CSP negative  19
## 2     CSP negative  41
## 3     CSP negative  18
## 4     CSP negative  76
## 5     CSP negative  15
## 6     CSP negative 128
```

```
#cohort sample data with some demographic variables
sample_data <- read.csv("/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofFlorida/Desktop/Resea
head(sample_data)
```

```
##   id age sex SNAP  WNV  YF JE3 ZIKA     DENV    CHIKV GLURPR2  CSP PfAMA1
## 1  1   0   2   80   90 105 116  100 310.4210 154.6159    1503 5429  52324
## 2  2   0   2   74   71  82  94   87 486.4240 387.3742     115   13   2683
## 3  3   0   2  214  234 235 278  235 713.3341 349.4496       5    4      7
## 4  4   0   2 1495 1475 153 186  160 215.3154 205.5457       2    3      8
```

```
## 5  5   0   1  226 2095 245 247 2195 395.7054 146.3569       9   11     24
## 6  6   0   2  189 2415 232 286  216 440.5869 253.1668      35  342  42924
##   PfMSP119 WRUV WMEV
## 1      661  803  721
## 2     7007 7976  103
## 3       16    5    7
## 4       10  619   23
## 5    10475 5858 1302
## 6      108   32   69
```

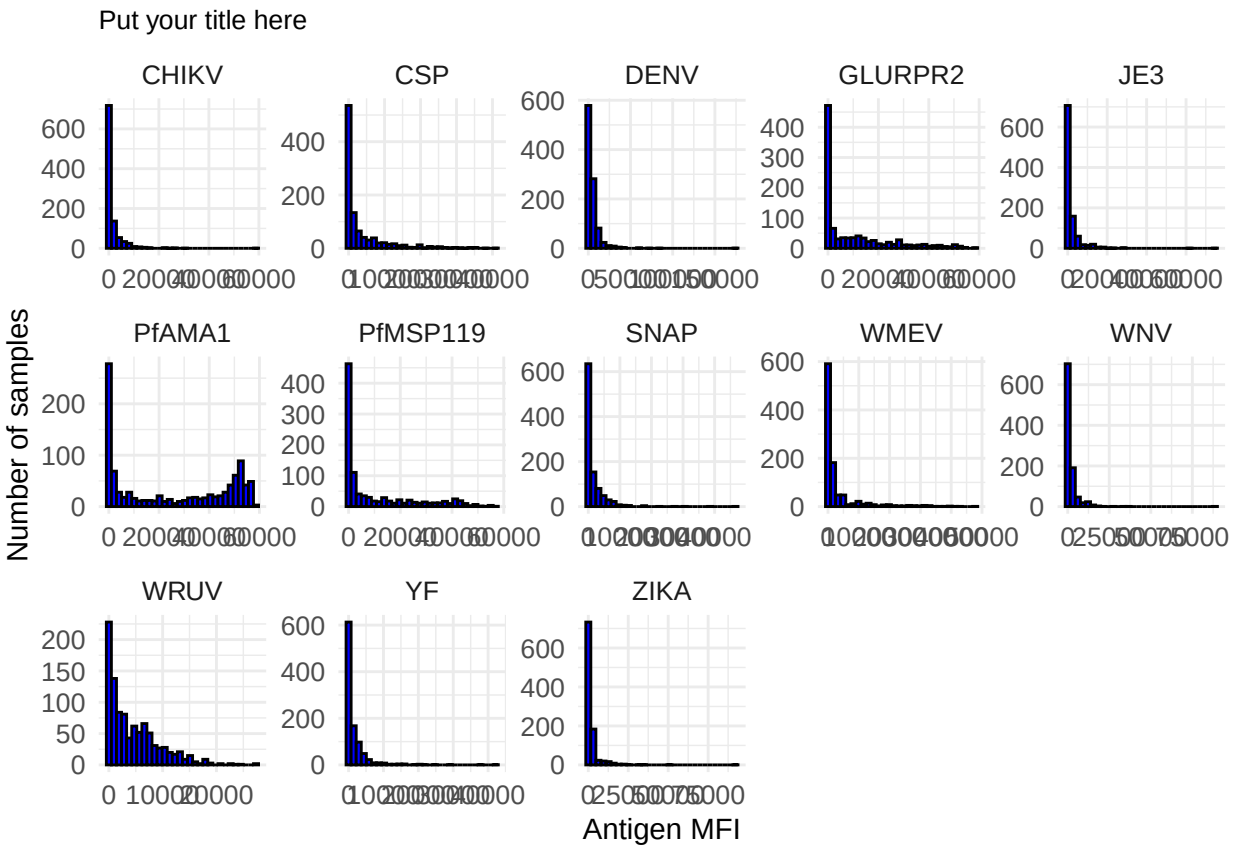
```r
#this converts sample_data from a wide to a long dataframe.
sample_long <- reshape(
  sample_data,
  varying = setdiff(names(sample_data), c("id", "age", "sex")),
  v.names = "mfi",
  timevar = "antigen",
  times = setdiff(names(sample_data), c("id", "age", "sex")),
  idvar = "id",
  direction = "long"
)
rownames(sample_long) <- NULL
```

#General visalization

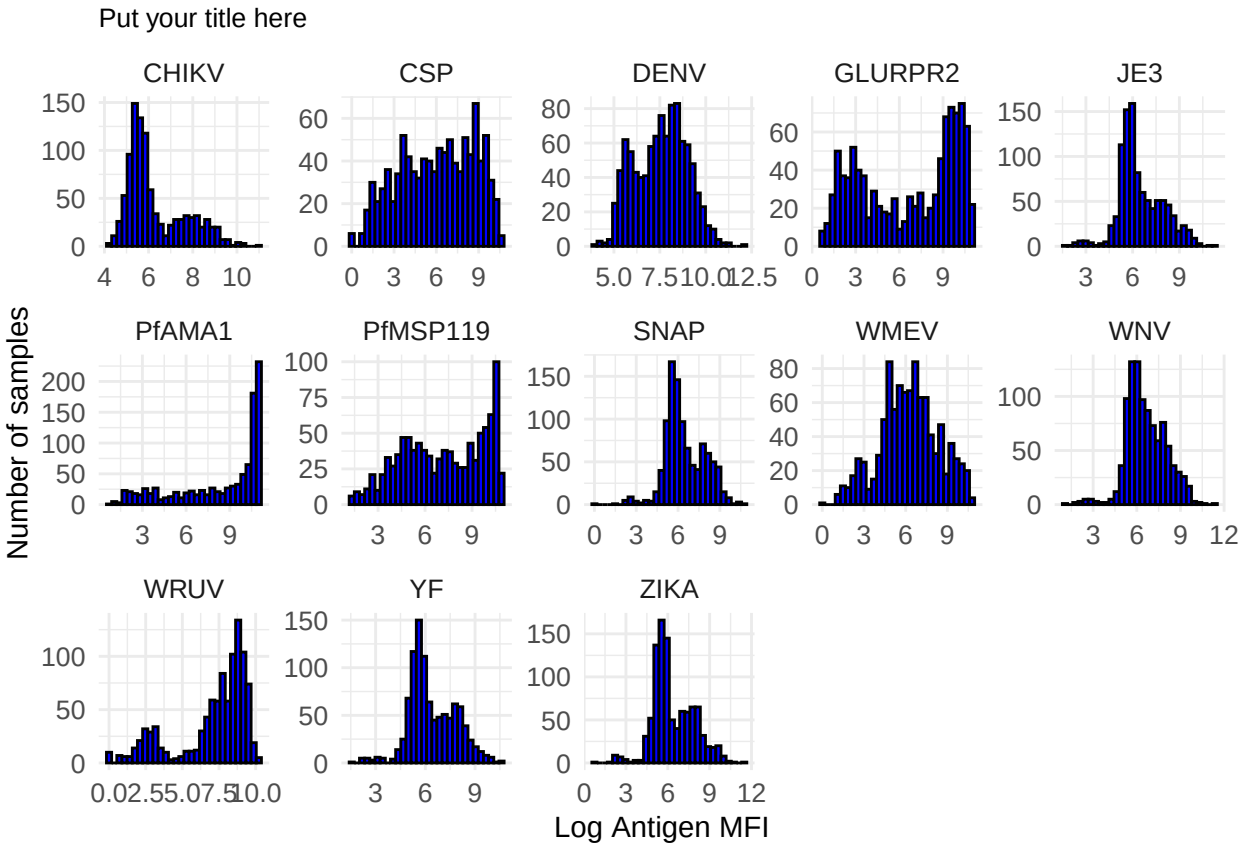
1. Adjust the code below to make a histogram of sample MFI values in an untransformed and log scale.

a. Consider how many bins to use (edit bins = 30 to see what data looks like with different numbers of bins).
b. Describe the distribution (untransformed and log scale).
c. Are there any outliers or anything unusual about your data?

```r
#natural scale
faceted_natural_scale <- ggplot(sample_long, aes(x = mfi)) +
  geom_histogram(bins = 30, color = "black", fill = "blue") +
  facet_wrap(~ antigen, scales = "free", ncol = 5) +  # <- Set 5 columns per row
  labs(
    title = "Put your title here",
    x = "Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),  # Smaller font for facet labels
    axis.text = element_text(size = 10),  # Smaller font for axis text
    plot.title = element_text(size = 10)  # Smaller font for the plot title
  )
faceted_natural_scale
```

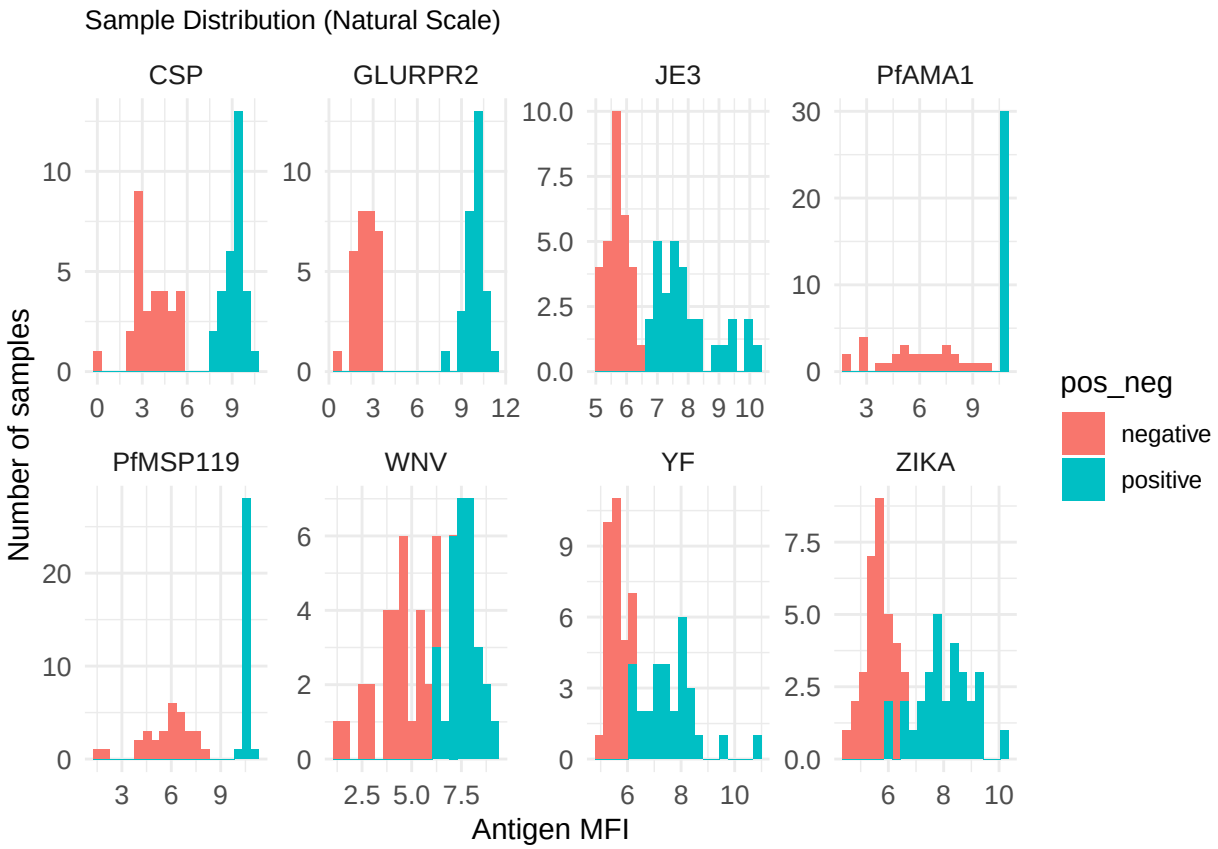



```
#log scale
faceted_log_scale <- ggplot(sample_long, aes(x = log(mfi))) +
  geom_histogram(bins = 30, color = "black", fill = "blue") +
  facet_wrap(~ antigen, scales = "free", ncol = 5) +  # <- Set 5 columns per row
  labs(
    title = "Put your title here",
    x = "Log Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),  # Smaller font for facet labels
    axis.text = element_text(size = 10),  # Smaller font for axis text
    plot.title = element_text(size = 10)  # Smaller font for the plot title
  )
faceted_log_scale
```

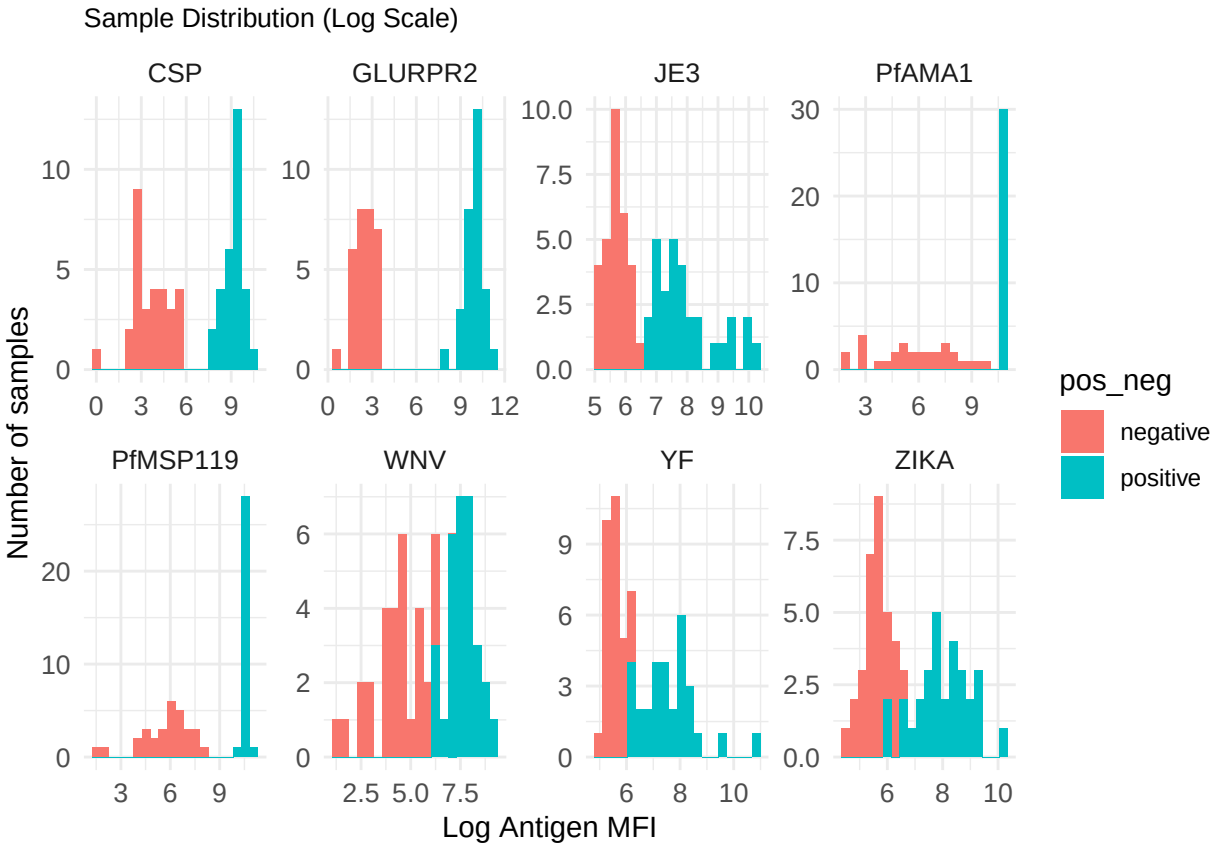


2. Make a histogram of control MFI values. Color the histogram by positive and negative controls. Is there overlap between your positive and negative controls?

```
neg_controls_natural_scale <- ggplot(control_data, aes(x = log(mfi), fill = pos_neg)) +
  geom_histogram(bins = 20) +
  facet_wrap(~ antigen, scales = "free", ncol = 4) +
  labs(
    title = "Sample Distribution (Natural Scale)",
    x = "Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),
    axis.text = element_text(size = 10),
    plot.title = element_text(size = 10)
  )
neg_controls_natural_scale
```




```r
neg_controls_log_scale <- ggplot(control_data, aes(x = log(mfi), fill = pos_neg)) +
  geom_histogram(bins = 20) +
  facet_wrap(~ antigen, scales = "free", ncol = 4) +  # <- Set 5 columns per row
  labs(
    title = "Sample Distribution (Log Scale)",
    x = "Log Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),  # Smaller font for facet labels
    axis.text = element_text(size = 10),  # Smaller font for axis text
    plot.title = element_text(size = 10)  # Smaller font for the plot title
  )
neg_controls_log_scale
```



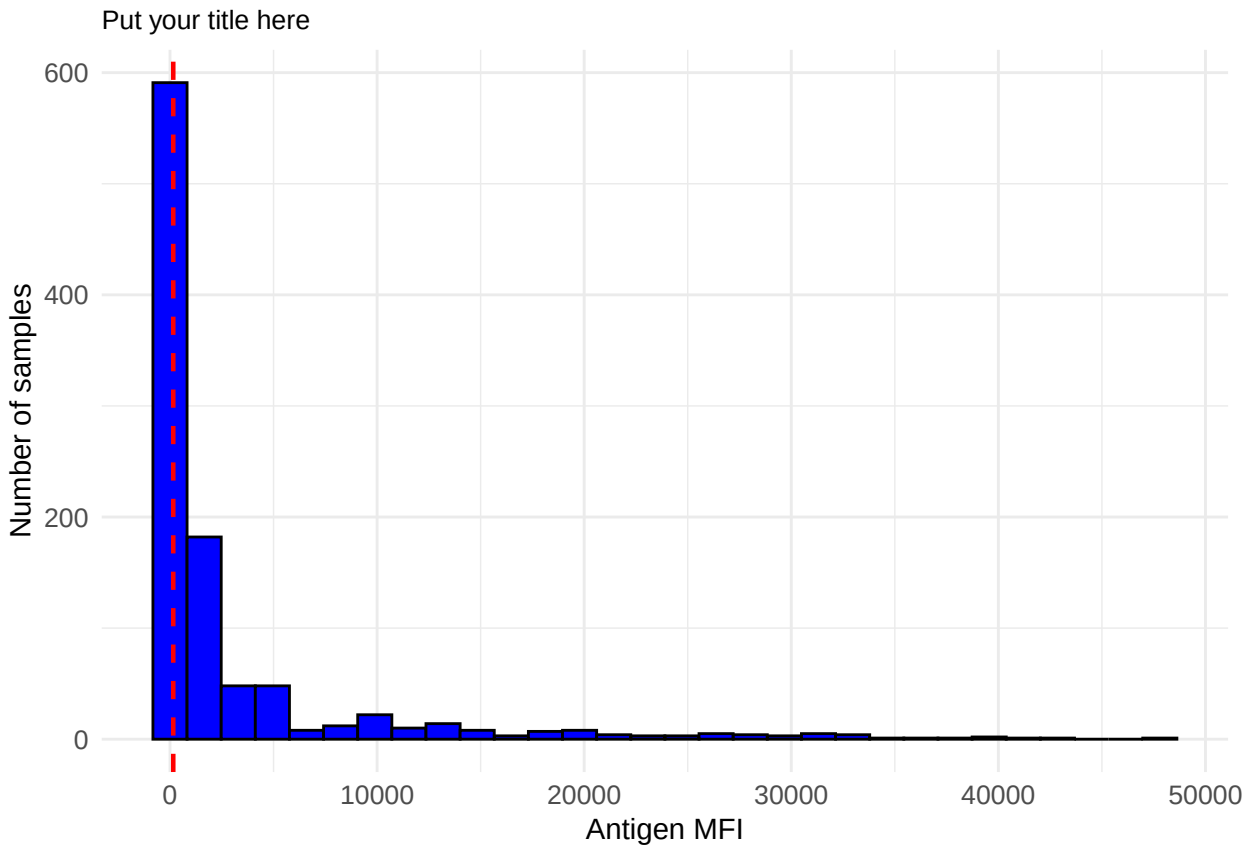
Establishing and applying a cutoff

3. For an antigen with an established standard cutoff (eg. based on a correlate of protection) like Measles (wmev antigen) the cutoff value of 153 mIU/mL is equivalent to an MFI of 156.94 based on a standard curve. Therefore, those samples with MFI above 156.94 are positive and protected from future measles infection, and those samples with MFI below 156.94 are negative and not protected from future measles infection.

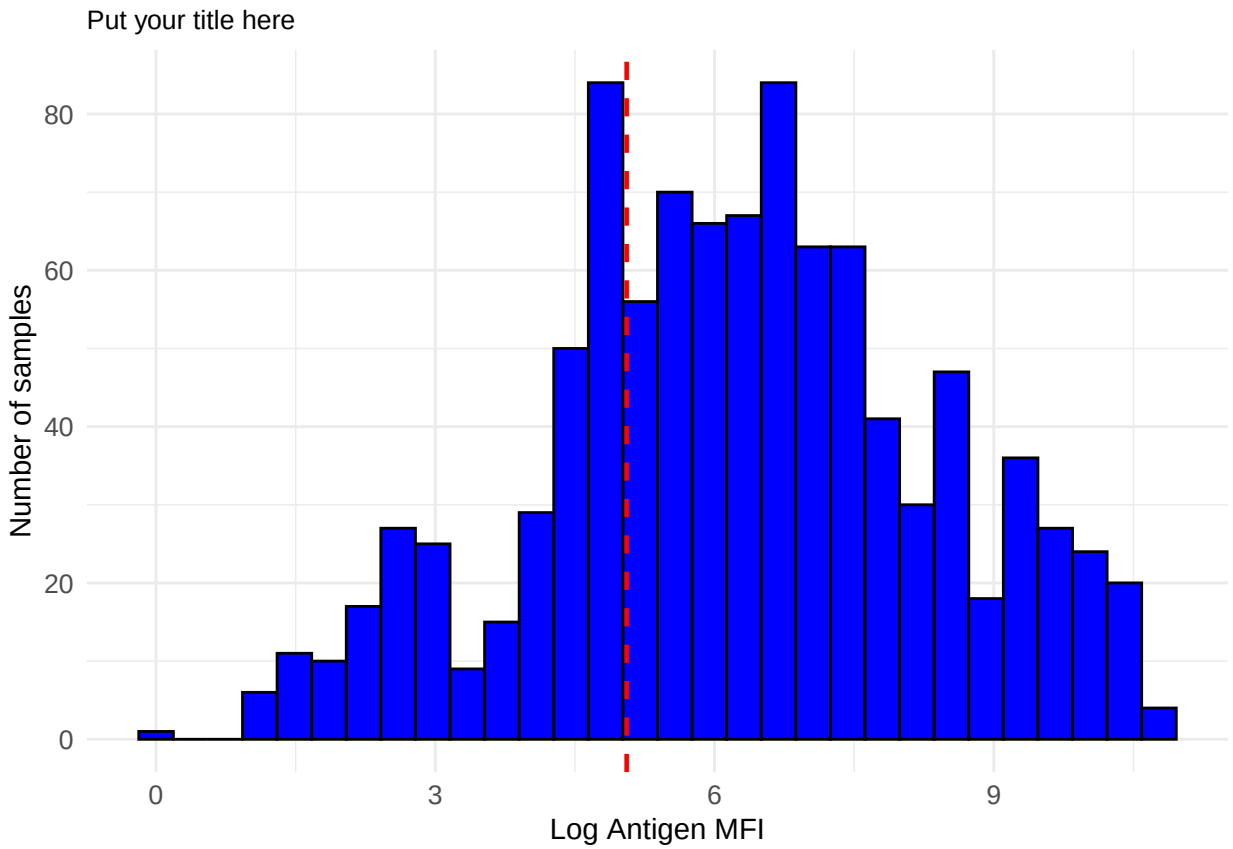
a. For this antigen, remake the histogram showing sample MFI values and add a vertical line showing the cutoff. Do this on an untransformed and a log scale. Be sure to adjsut the code to include an appropriate title.

```
cutoff<- 156.94
#natural scale
faceted_natural_scale <- ggplot(sample_data, aes(x = WMEV)) +
  geom_histogram(bins = 30, color = "black", fill = "blue") +
  geom_vline(xintercept = cutoff, linetype = "dashed", color = "red", size = 0.8) +  # Add vertical das
  labs(
    title = "Put your title here",
    x = "Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),  # Smaller font for facet labels
    axis.text = element_text(size = 10),  # Smaller font for axis text
    plot.title = element_text(size = 10)  # Smaller font for the plot title
```

```
  )
faceted_natural_scale
```




```
#log scale
faceted_log_scale <- ggplot(sample_data, aes(x = log(WMEV))) +
  geom_histogram(bins = 30, color = "black", fill = "blue") +
  geom_vline(xintercept = log(cutoff), linetype = "dashed", color = "red", size = 0.8) +  # Add vertica
  labs(
    title = "Put your title here",
    x = "Log Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),
    axis.text = element_text(size = 10),
    plot.title = element_text(size = 10)
  )
faceted_log_scale
```



b. Apply cutoff. How many people are seropositive according to this cutoff, and what proportion of people are seropositive?

```
#Applying cutoff:
  #ifelse function, if first statement is true, then outcome is set to 1, and if first statemnet is fal
  #since the first statement is a vector, then outcome will be a vector of 1's and 0's.
seropositivity <- ifelse(sample_data$WMEV > cutoff, 1, 0) #1 indicates seropositive, and 0 indicates se

#number of people seropositve and seronegative
cat("Table of number seronegative and seronegative", "\n")
```

```
## Table of number seronegative and seronegative
```

```
table(seropositivity,useNA="always")
```

```
## seropositivity
##    0    1 <NA>
##  288  712    0
```

```
#percent of people seropositive and negative
cat("Table of percent seronegative and seronegative", "\n")
```

```
## Table of percent seronegative and seronegative
```

```
round(prop.table(table(seropositivity,useNA="always")),3)*100
```

```
## seropositivity
##    0    1 <NA>
## 28.8 71.2  0.0
```

c. Calculate the confidence interval for this seroprevalence

```r
# Set your parameters
x <- 712 #number seropositive. Get the number seropositive in your sample from the seropositive table
n <- nrow(sample_data) #total number of samples in your data, note in these data we saw above there w
conf <- 0.95 #confidence interval. 95% is a standard CI but you can adjust this if you want

# Exact interval
ci <- binom.exact(x, n, conf.level = conf) #epitools function
#
cat("CI lower", round(ci$lower,4)*100, "%, CI upper", round(ci$upper,4)*100, "%")
```

```
## CI lower 68.28 %, CI upper 73.99 %
```

d. For this specific antigen, how would you interpret this seropositivity and confidence interval?

e. What do you think about using this cutoff method for this antigen? What are the assumptions that went into this cutoff method?