ST117 Final Written Report - Part Moths

Report Pod 017

Today's date in the format 2024-05-06

- Please read the document "WR Assessment, Style, and Submission Guidance".
- Please also read the FAQs about the WR on Moodle.
- Please adhere to the upper limits for pages, words, number of figures and tables.
- We put together some generic examples of code chunks for initial steps and data visualisation (see in the bat section below). The references to online sources may be worth exploring further if you are looking for ideas about how to visually represent data.
- Please remove the generic examples before you submit and replace them by your own work. Include your own work in other sections, too.
- Please replace 099 and 1999-12-31 in the header by the appropriate numbers.
- Please list below all members of the report pod who contributed to the work on this assignment (including the author of this Rmd file).
- Please also list the specific species and locations assigned to your report pod to be used in some of the questions

This final Written Report submission is for Report Pod 099 with contributions from:

1. Archie Bevan - 5538165
2. Nicolas Antoniou - 5538893
3. Chenxi Zhang - 5508580
4. Jack Mahon - 5538889
5. Ewan Abbottspooner - 5536967

My Report Pod's species (FIELDNAME)

Moths (302;180;382;648;313):

Birds (MP;WR;S;C):

Bats (Pp;M;P):

My Report Pod's locations (SITECODE)

Locations (T01;T02;T03;T04):

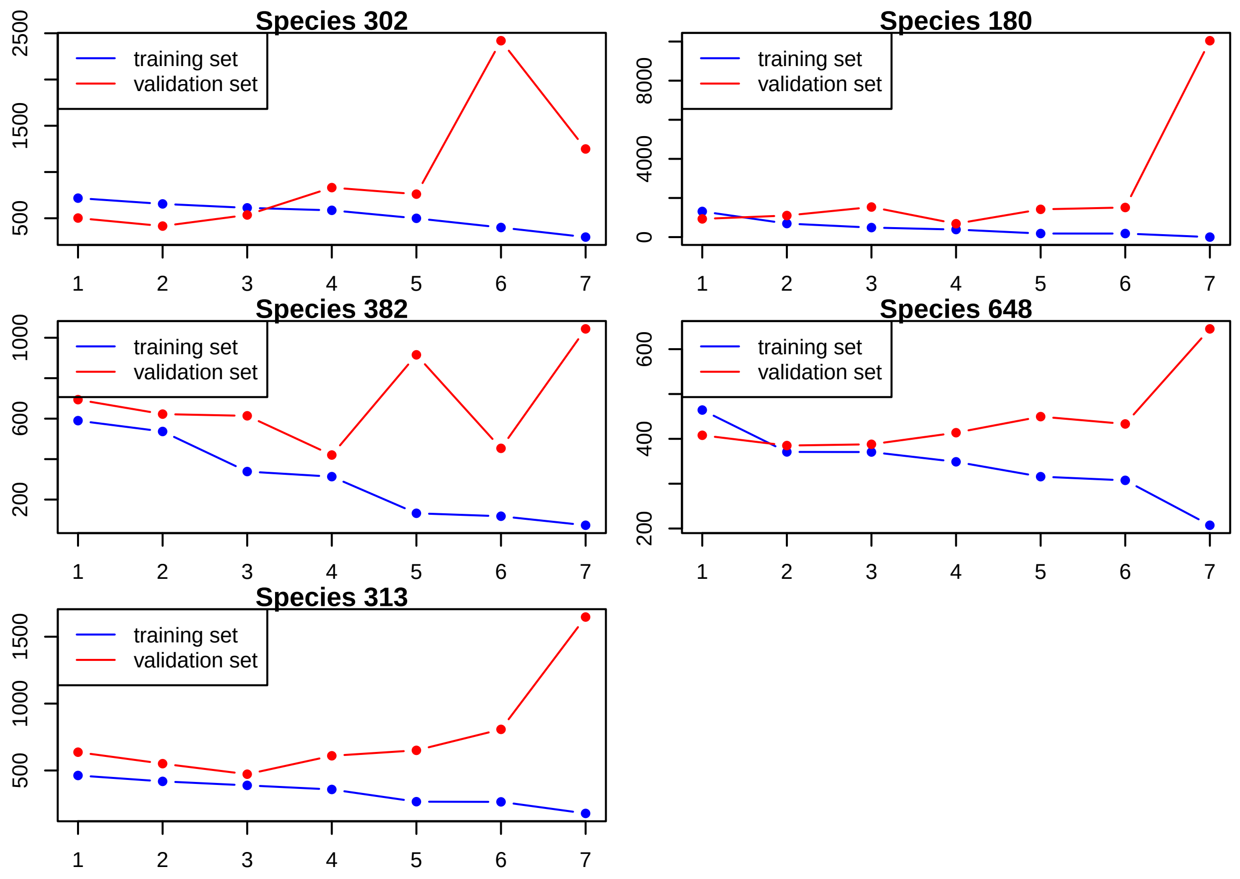
Question 1 (Polynomial regression for seasonal patterns)

Citation: For the following questions involving polynomial regression we used the R example code from the week 9 lecture notes- “Simulation Example R code” “Wages Example R code”.

a) Aggregated locations, 5 species

We start by finding the best order of polynomial for each of the 5 moth species by finding the order that gives the least RMSE for both the training and validation sets.

```r
setwd('../../ST117 Project/00_raw_data/')
set.seed(69)

# Create a list of years for which we have moth data for
yearlist = 1992:2015
# Create a list for species given to our report pod
specieslist = c(302,180,382,648,313)
# Create a list of the sitecodes given to our group
sitelist = c("T01","T02","T03","T04")

#Create data frames from csvs
Moths = read.csv("Moth Raw Data.csv")

#Change Sampling Date Column to a different format
Moths$SDATE <- as.Date(Moths$SDATE, format = "%d-%b-%y")

#Create a year column
Moths$YEAR = year(Moths$SDATE)

#Create a week column
Moths$WEEK = week(Moths$SDATE)

# Create a new dataframe, keeping only the species in our specieslist
# and aggregating counts by week for each species

observed_counts <- Moths %>%
  filter(FIELDNAME %in% specieslist) %>%
  group_by(WEEK,FIELDNAME) %>%
  summarise(total_value = sum(VALUE))


# Plotting setup
par(mfrow=c(3,2), mar=c(2,2,1,1)) # Set up 6 subplots

for (species in specieslist) { #for each of the five species in the list

#Create a new dataframe with just that particular species
Moth_data <- observed_counts %>%
  filter(FIELDNAME == species)

# define split vector
split <- sample.split(Moth_data$total_value, SplitRatio = 0.65)
# split data set
train_set <- Moth_data[split == TRUE, ]
val_set <- Moth_data[split == FALSE, ]

# setup

# empty data frame to store RMSE
RMSE <- data.frame("kth_order" = NA, "RMSE_train" = NA, "RMSE_val" = NA)

# Remove rows with missing values in 'Week' variable
train_set <- train_set[complete.cases(train_set$WEEK), ]
val_set <- val_set[complete.cases(val_set$WEEK), ]

# set up vector used for prediction
vals <- list("x" <- seq(min(train_set$WEEK,na.rm=TRUE),
                        max(train_set$WEEK,na.rm=TRUE), by = 1))

# run  loop
k <- seq(1, 7) # k-th order

for (i in k) {
  # build models
  model <- lm(total_value ~ poly(WEEK, k[i]), data = train_set)

  # calculate RMSE and store it for further usage
  RMSE[i, 1] <- k[i] # store k-th order
  # calculate RMSE of the training set
  RMSE[i, 2] <- sqrt(sum((fitted(model) - train_set$total_value)^2) /
                       length(train_set$total_value))

  # predict
  predictions <- predict(model, newdata = val_set)
  # calculate RMSE of the validation set
  RMSE[i, 3] <- sqrt(sum((predictions - val_set$total_value)^2) /
                       length(val_set$total_value))
}

# plot RMSE for training and validation set
plot(RMSE[, 1], RMSE[, 2],
  xlab = "k-th order", ylab = "RMSE",
  main=paste0("Species ", species),
  ylim = c(min(RMSE[, c(2, 3)]), max(RMSE[, c(2, 3)])),
  type = "b", col = "blue", pch = 16
)
lines(RMSE[, 3],
  type = "b", col = "red", pch = 16
)
legend("topleft",
  legend = c("training set", "validation set"),
  lty = c(1, 1),
  col = c("blue", "red")
)
}
```



As a result, we record the appropriate order for each species.

```
Species_Table = data.frame(
  Species = specieslist,
  Order = c(3,4,6,2,3)
)

print(Species_Table)
```

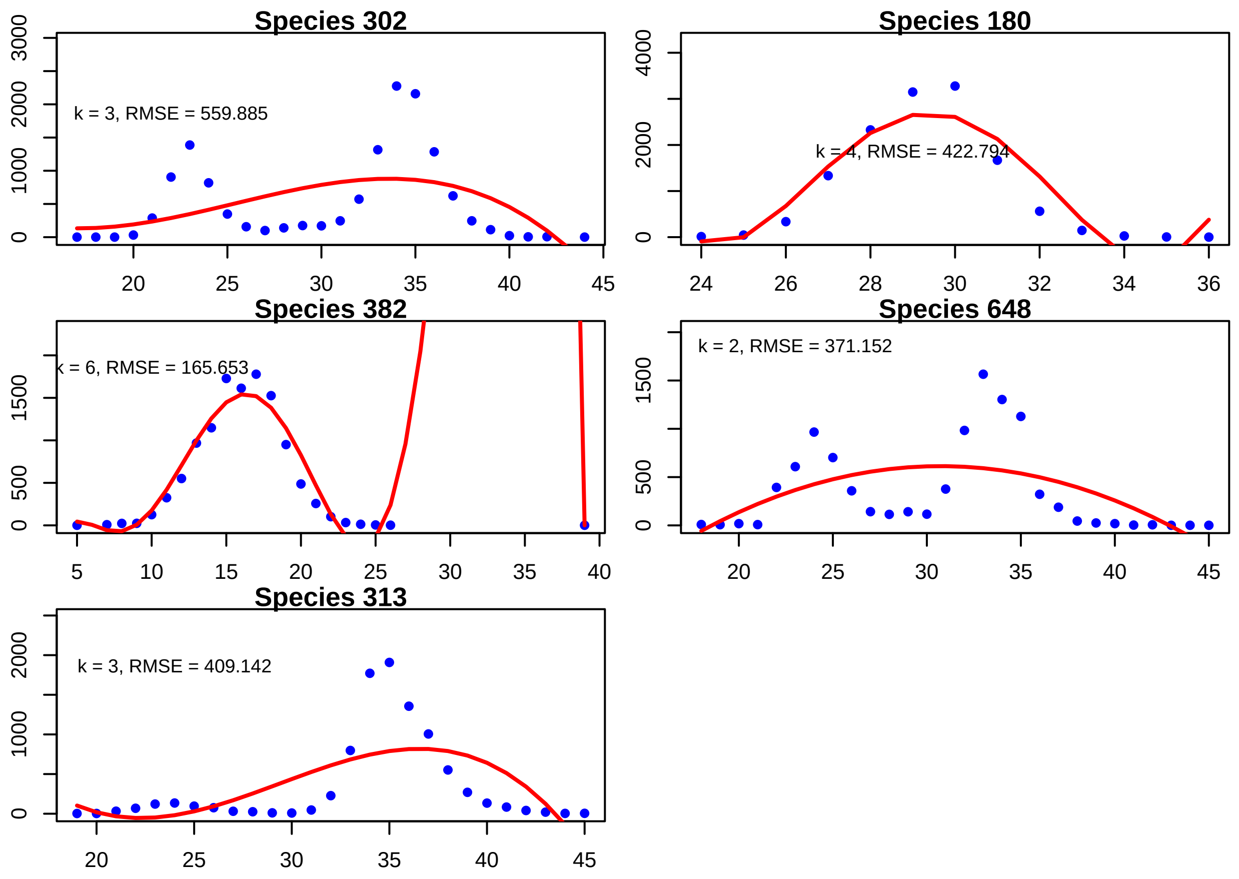
```
  Species Order
1     302     3
2     180     4
3     382     6
4     648     2
5     313     3
```

We conduct polynomial regression for the orders of polynomials we found for each species.

```
# Plotting setup
par(mfrow=c(3,2), mar=c(2,2,1,1)) # Set up 6 subplots

# Get appropriate orders from table above
k = Species_Table$Order

# Set up data frame to store RMSE
```

```r
RMSE <- data.frame("kth_order" = k, "RMSE" = NA)


# For each of the 5 species
for (i in 1:5){

  #Make a new dataframe for each species
  species_df = observed_counts %>%
    filter(FIELDNAME == specieslist[i])

  # Set up vector used for prediction
  vals <- list("x" = seq(min(species_df$WEEK), max(species_df$WEEK), by = 1))

  # Build models
  model <- lm(total_value ~ poly(WEEK, k[i]), data = species_df)

  # Calculate RMSE
  RMSE[i, 2] <- sqrt(mean((predict(model) - species_df$total_value)^2))

  # Predict
  predictions <- predict(model, newdata = data.frame(WEEK = vals$x))

  # Plot
  plot(species_df$WEEK, species_df$total_value, pch = 16, col = "blue",
       main=paste0("Species ", specieslist[i]),
       ylim = c(min(species_df$total_value) * 1.3, max(species_df$total_value)
                * 1.3),
       ylab ="Moth Counts", xlab= "Week")
  lines(vals$x, predictions, lwd = 2, col = "red")
  # Annotate the plot
  text(x = min(species_df$WEEK)+5, y = 1850, paste0("k = ", k[i], ", RMSE = ",
                                                    round(RMSE[i, 2], 3)), cex=0.9)




}
print("Below are the plots that minimise RMSE for both validation and training set")
```

```
[1] "Below are the plots that minimise RMSE for both validation and training set"
```



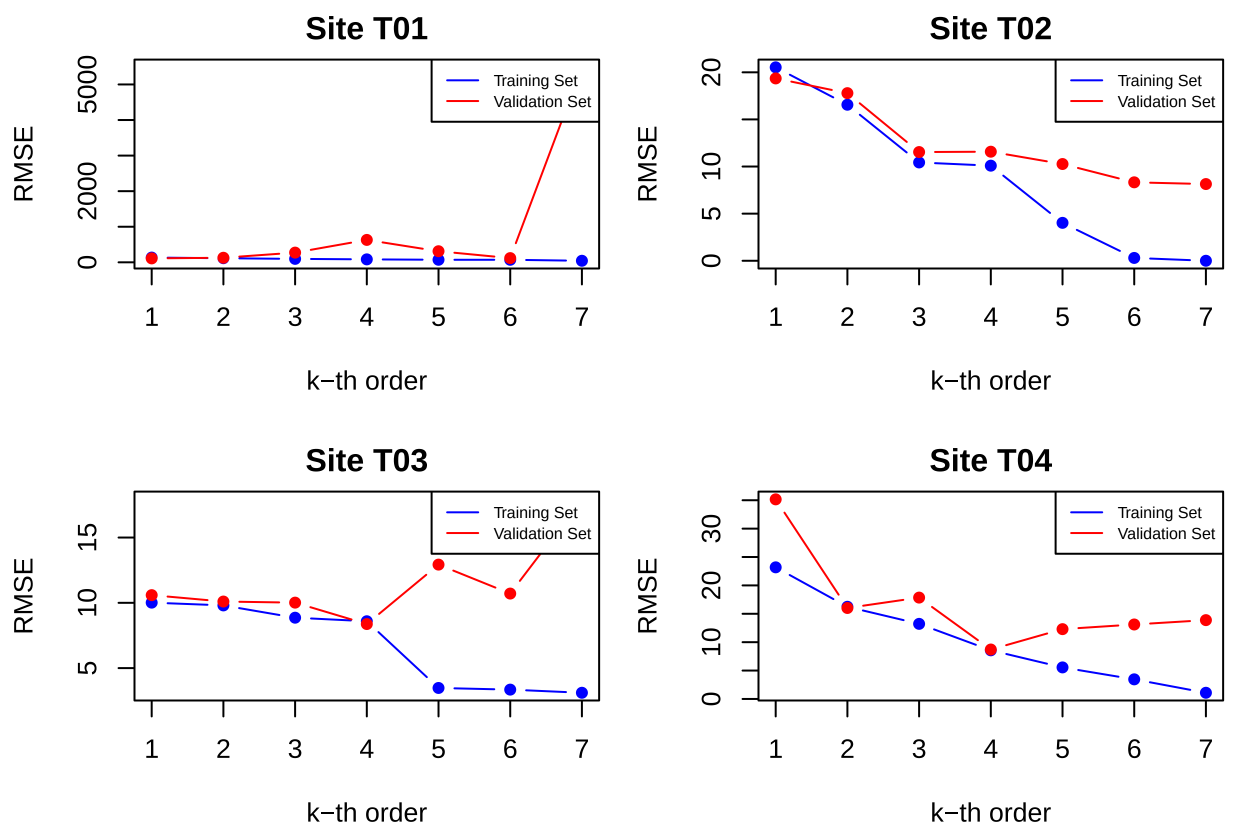
We can see for the species (180) our regression fit seems to fit well so our degree might not need to be changed however for species (302), the peaks are much steeper and the regression fit needs to have a higher degree polynomial. We note that polynomials of a higher order would result in a better fitting graph with less RMSE, however from the investigation above, we see that this may result in over-fitting and thus is a bad fit for the seasonality of moth species.

b) Locations separate, 1 species

We conduct polynomial regression for species 302. We investigate to find out the polynomial order for each site that minimises the RMSE for both training and validation sets.

```r
# Create a list of the sitecodes given to our group
sitelist <- c("T01", "T02", "T03", "T04")

# Plotting setup
par(mfrow=c(2,2), mar=c(5,4,2,1)) # Set up 4 subplots

for (site in sitelist) {
  # Create a new dataframe with just that particular site
  Moth_data <- Moths %>%
  filter(SITECODE == site, FIELDNAME == specieslist[1]) %>%
  group_by(WEEK,SITECODE) %>%
  summarise(total_value = sum(VALUE))

  # Define the SplitRatio based on the length of data
```

```r
split_ratio <- 0.65
if (nrow(Moth_data) > 1) {
  split_ratio <- 0.65 * nrow(Moth_data)
}

# Split data set
split <- sample.split(seq(nrow(Moth_data)), SplitRatio = split_ratio)
train_set <- Moth_data[split, ]
val_set <- Moth_data[!split, ]

# Remove rows with missing values in 'Week' variable
train_set <- train_set[complete.cases(train_set$WEEK), ]
val_set <- val_set[complete.cases(val_set$WEEK), ]

# Empty data frame to store RMSE
RMSE <- data.frame("kth_order" = NA, "RMSE_train" = NA, "RMSE_val" = NA)

# Set up vector used for prediction
vals <- list("x" <- seq(min(train_set$WEEK, na.rm = TRUE),
                        max(train_set$WEEK, na.rm = TRUE), by = 1))

# Run loop
k <- seq(1, 7) # k-th order

for (i in k) {
  # Build models
  model <- lm(total_value ~ poly(WEEK, k[i]), data = train_set)

  # Calculate RMSE and store it for further usage
  RMSE[i, 1] <- k[i] # Store k-th order
  # Calculate RMSE of the training set
  RMSE[i, 2] <- sqrt(sum((fitted(model) - train_set$total_value)^2) /
                       length(train_set$total_value))

  # Predict
  predictions <- predict(model, newdata = val_set)
  # Calculate RMSE of the validation set
  RMSE[i, 3] <- sqrt(sum((predictions - val_set$total_value)^2) /
                       length(val_set$total_value))
}

# Plot RMSE for training and validation set
plot(RMSE[, 1], RMSE[, 2],
     xlab = "k-th order", ylab = "RMSE",
     main = paste0("Site ", site),
     ylim = c(min(RMSE[, c(2, 3)]), max(RMSE[, c(2, 3)])),
     type = "b", col = "blue", pch = 16,
     xlim = c(1, max(k))
)
lines(RMSE[, 1], RMSE[, 3],
      type = "b", col = "red", pch = 16
)
legend("topright",
          legend = c("Training Set", "Validation Set"),
          lty = c(1, 1),
          col = c("blue", "red"),
          bg = "white",
          cex = 0.65
  )
}
```



As a result, we record the appropriate order for each site.

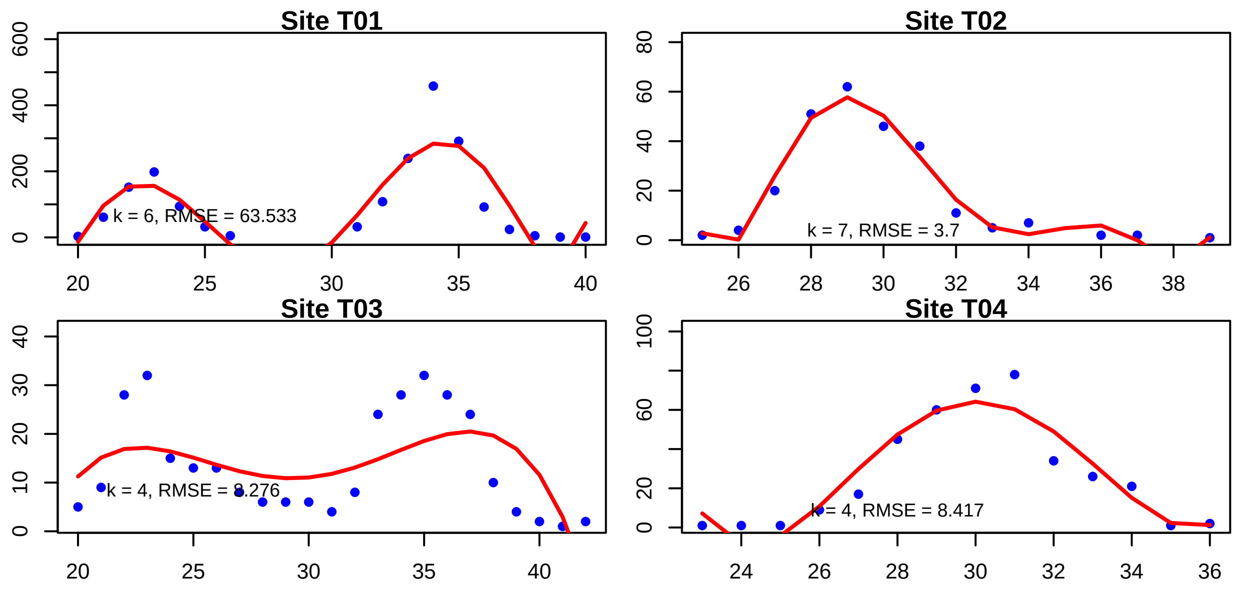
```
Site_Table = data.frame(
  Site = sitelist,
  Order = c(6,7,4,4)
)

print(Site_Table)
```

```
  Site Order
1  T01     6
2  T02     7
3  T03     4
4  T04     4
```

We conduct polynomial regression for the orders of polynomials we found for each site.

```r
# Plotting setup
par(mfrow=c(3,2), mar=c(2,2,1,1)) # Set up 6 subplots

# Get appropriate orders from table above
k = Site_Table$Order

# Set up data frame to store RMSE
RMSE <- data.frame("kth_order" = k, "RMSE" = NA)

# For each of the 4 sites
for (i in 1:4){

  #Make a new dataframe for each site
  site_df = Moths %>%
    filter(SITECODE == sitelist[i],FIELDNAME == specieslist[1]) %>%
    group_by(WEEK,SITECODE) %>%
    summarise(total_value = sum(VALUE))

  # Set up vector used for prediction
  vals <- list("x" = seq(min(site_df$WEEK), max(site_df$WEEK), by = 1))

  # Build models
  model <- lm(total_value ~ poly(WEEK, k[i]), data = site_df)

  # Calculate RMSE
  RMSE[i, 2] <- sqrt(mean((predict(model) - site_df$total_value)^2))

  # Predict
  predictions <- predict(model, newdata = data.frame(WEEK = vals$x))

  # Plot
  plot(site_df$WEEK, site_df$total_value, pch = 16, col = "blue",
       main=paste0("Site ", sitelist[i]),
       ylim = c(min(site_df$total_value) * 1.3, max(site_df$total_value)
                * 1.3),
       ylab ="Moth Counts", xlab= "Week")
  lines(vals$x, predictions, lwd = 2, col = "red")
  # Annotate the plot
  text(x = min(site_df$WEEK)+5, y = max(RMSE[i,2]), paste0("k = ", k[i], ", RMSE = ",
                                                     round(RMSE[i, 2], 3)), cex=0.9)
}
print("Below are the plots that minimise RMSE for both validation and training set")
```

```
[1] "Below are the plots that minimise RMSE for both validation and training set"
```



Conclusion

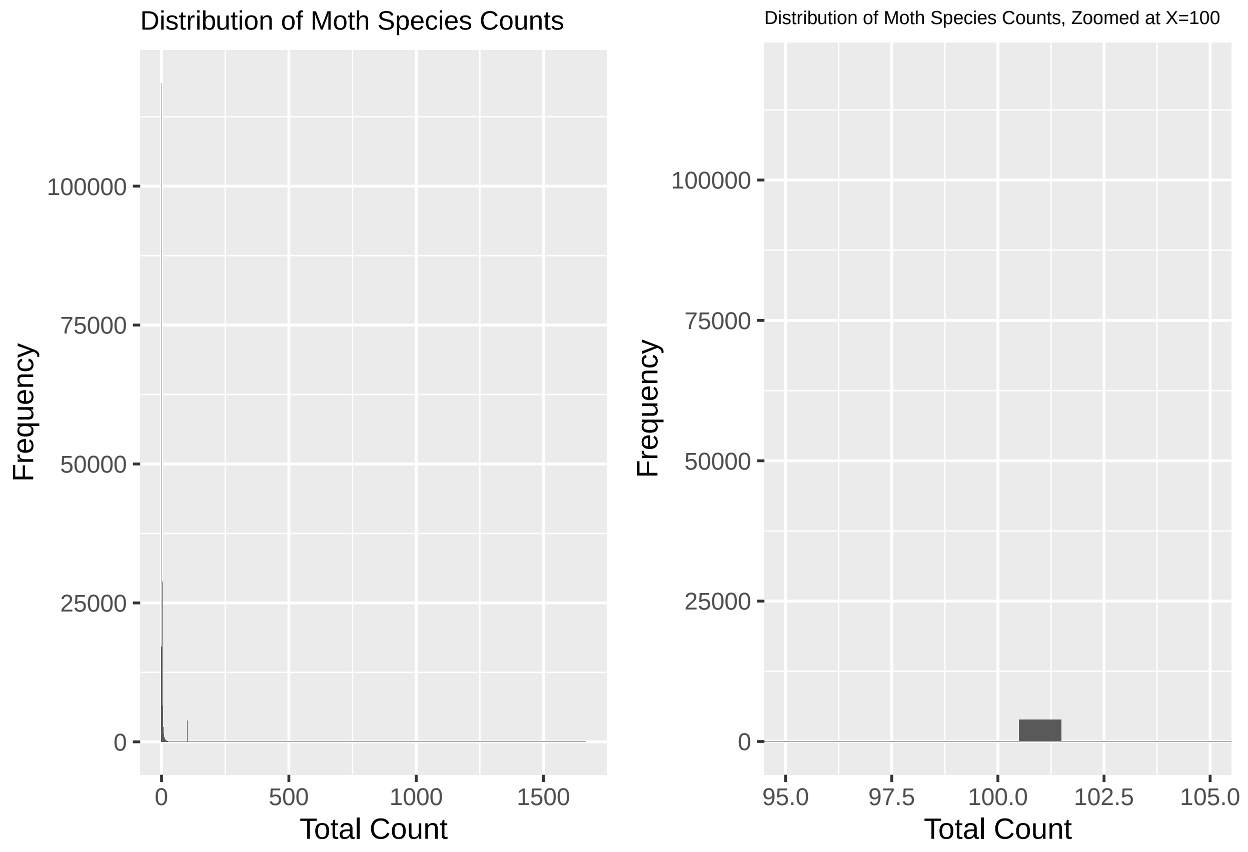
Summarise what the models suggest about the seasonality of the moths counts

These polynomial regression models for species 302 show that seasonality for that species depends on the site. For example, Site T02 shows that moth species 302 has a peak around weeks 28-30 and then declines in sample count for all other weeks of the year. This is in contrast to Site T03 where the sample counts are shown to have two peaks, from weeks 20-25, then declines at weeks 25-32 and increases in sample count at weeks 33-36. This is the exact opposite from the seasonality shown at site T02. A similarity between the models is that by week 40 the sample count is very low or at zero for all locations.

Question 2 (Distributions of counts)

a) Overall

```r
# Disable scientific notation
options(scipen=10000)

# Plot Histogram
plot1 <- ggplot(Moths, aes(x = VALUE)) +
  geom_histogram(binwidth = 1) +
  labs(title = "Distribution of Moth Species Counts",
       x = "Total Count",
       y = "Frequency") +
  theme(plot.title = element_text(size = 10))

plot2 <- ggplot(Moths, aes(x = VALUE)) +
  geom_histogram(binwidth = 1) +
  labs(title = "Distribution of Moth Species Counts, Zoomed at X=100",
       x = "Total Count",
       y = "Frequency") +
  coord_cartesian(xlim = c(95,105)) +
  theme(plot.title = element_text(size = 7))

grid.arrange(plot1,plot2 , ncol = 2)
```



We plot the distribution of all counts of all moth species in Figure 1. We show the outliers presented in this distribution in Figure 2.

We note that this distribution is very positively skewed and also consists of many outliers in the 100-102.5 region as indicated by the second plot above. As a result, the mean of the distribution would be value larger than 1, while the median and mode are equal to 1, due to the large number of outliers at 100-102.5.

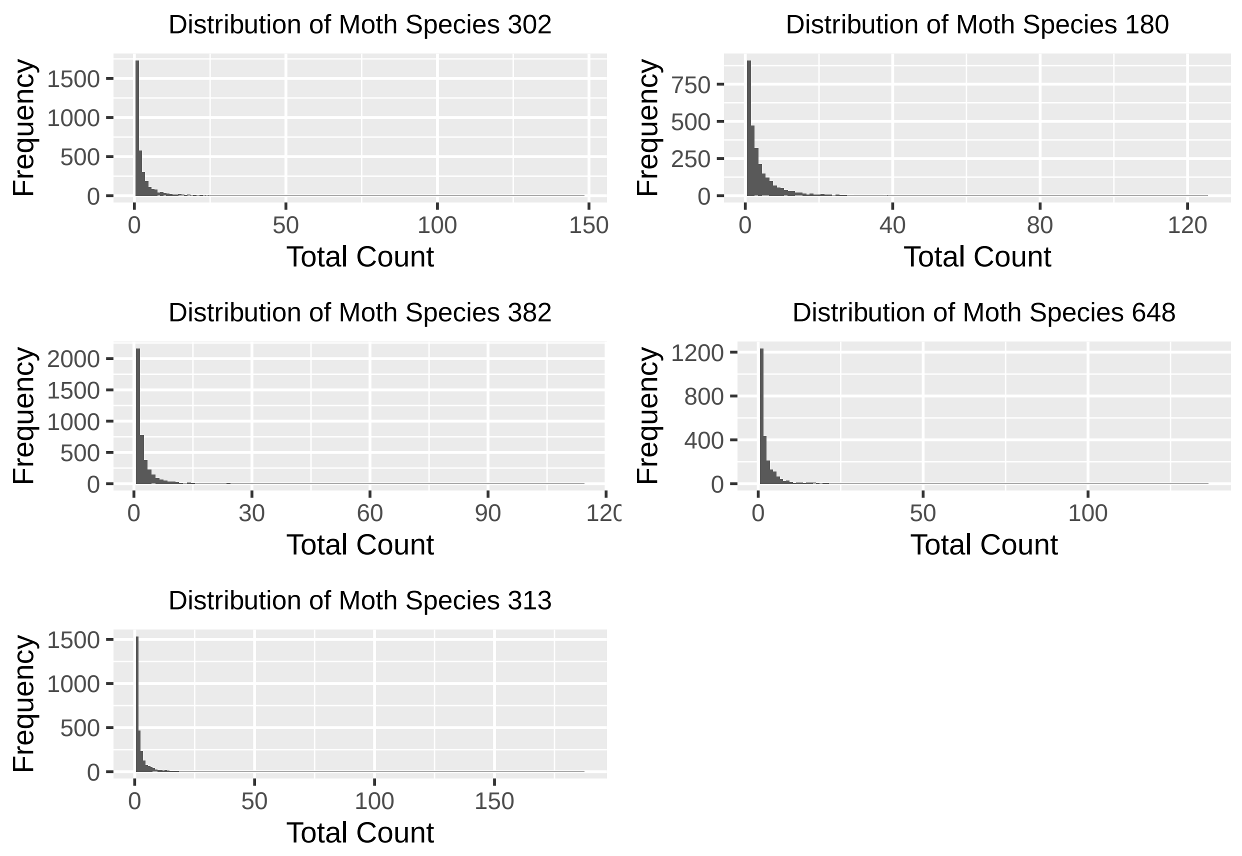
```
  Characteristics   Values
1            Mean 4.148842
2          Median 1.000000
3            Mode 1.000000
```

As a result, the value of the mean is very far apart to the mode and median which are both 1. Therefore, a parametric distribution family would be a difficult fit for this distribution.

b) 5 species

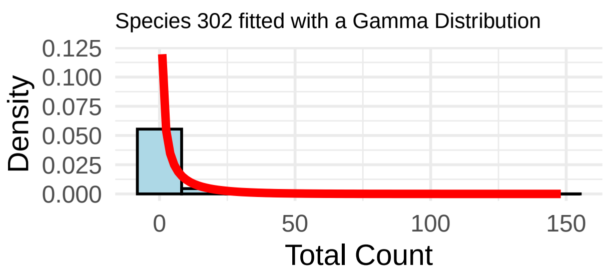
```r
# Define log-likelihood functions for each distribution
log_likelihood_geometric <- function(data, p) {
  sum(dgeom(data$VALUE, prob = p, log = TRUE))
}

log_likelihood_binomial <- function(data, size, prob) {
  sum(dbinom(data$VALUE, size = size, prob = prob, log = TRUE))
}

log_likelihood_gamma <- function(data, shape, rate) {
  sum(dgamma(data$VALUE, shape = shape, rate = rate, log = TRUE))
}

log_likelihood_beta <- function(data, shape1, shape2) {
  sum(dbeta(data$VALUE, shape1 = shape1, shape2 = shape2, log = TRUE))
}


log_likelihood_normal <- function(data, mean, sd) {
  sum(dnorm(data$VALUE, mean = mean, sd = sd, log = TRUE))
}

log_likelihood_poisson <- function(data, lambda) {
  sum(dpois(data$VALUE, lambda = lambda, log = TRUE))
}

log_likelihood_negbin <- function(data, p, k) {
  sum(dnbinom(data$VALUE, size = k , prob = p , log = TRUE))
}

log_likelihood_exponential <- function(data, r) {
  sum(dexp(data$VALUE, rate = r, log = TRUE))
}

# Filter data for each species in the specieslist
filtered_data <- lapply(specieslist, function(species){
  Moths %>%
    filter(FIELDNAME == species)
})

# Plot histograms for each species in a grid
plots <- lapply(filtered_data, function(df){
  ggplot(df, aes(x = VALUE)) +
    geom_histogram(binwidth = 1) +
    labs(title = paste("Distribution of Moth Species", unique(df$FIELDNAME)),
         x = "Total Count",
         y = "Frequency") +
    theme(plot.title = element_text(size = 10,hjust =.5))
})
```

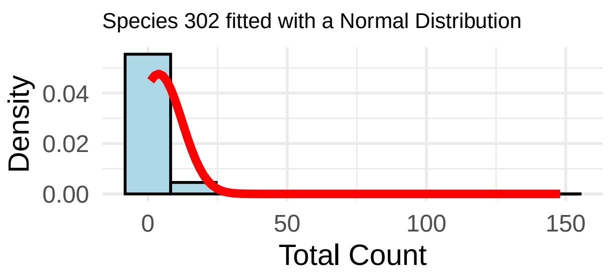
```r
grid.arrange(grobs = plots, ncol = 2)
```

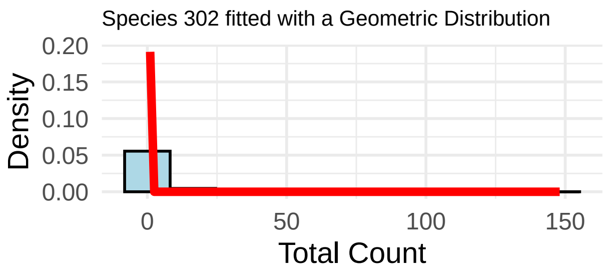


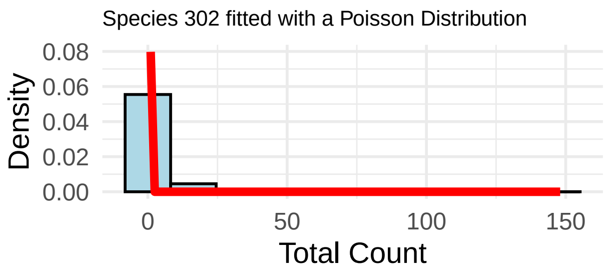
```r
#Create a table for the log likelihood of each distribution
Table_Df = data.frame(
  Distribution = c("Exponential","Gamma","Geometric","Normal","Poisson", "Neg Bin")
)
# for each species in the specieslist
for (species in specieslist){

# make a new dataframe of just that species
df_species <- Moths %>%
  filter(FIELDNAME == species, )

# calculate the parameters of each distribution using the sample data

first_moment = mean(df_species$VALUE)
second_moment = mean((df_species$VALUE)^2)
normal_sd = sd(df_species$VALUE)

geometric_p = 1/first_moment

betadist_alpha = (first_moment^2-first_moment*second_moment)/(second_moment-first_moment^2)
betadist_beta = betadist_alpha*((1-first_moment)/first_moment)

gammadist_alpha = first_moment^2/(second_moment-first_moment^2)
```

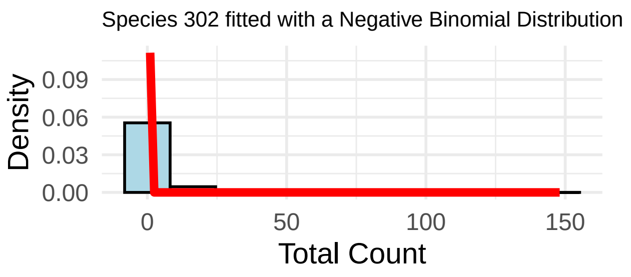
```r
gammadist_beta = first_moment/(second_moment-first_moment^2)

binomial_p = 1 - var(df_species$VALUE)/first_moment

negbin_p = first_moment/normal_sd^2
negbin_k = first_moment^2/(normal_sd^2-first_moment)

parameters = c(first_moment,second_moment,normal_sd,geometric_p,betadist_alpha,
               betadist_alpha,gammadist_alpha,gammadist_beta,negbin_p,negbin_k)

gammaplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dgamma,
args = list(shape = parameters[7],
rate = parameters[8]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Species", species, "fitted with a Gamma Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 8))

poissonplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dpois,
args = list(lambda = parameters[1]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Species", species, "fitted with a Poisson Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 8))

normalplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dnorm,
args = list(mean = parameters[1],
sd = parameters[3]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Species", species, "fitted with a Normal Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 8))

geomplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dgeom,
args = list(parameters[4]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Species", species, "fitted with a Geometric Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 8))

negbinplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dnbinom,
args = list(size = parameters[10], prob = parameters[9]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Species", species, "fitted with a Negative Binomial Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 8))

exponplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dexp,
args = list(rate = 1/parameters[1]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Species", species, "fitted with an Exponential Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 8))

grid.arrange(gammaplot,normalplot,geomplot ,poissonplot, negbinplot, exponplot, ncol = 2)

# create a vector of the log likelihoods of each of the distributions

newvector = c(log_likelihood_exponential(df_species,geometric_p),
              log_likelihood_gamma(df_species,gammadist_alpha,gammadist_beta),
              log_likelihood_geometric(df_species,geometric_p),
              log_likelihood_normal(df_species,first_moment,normal_sd),
              log_likelihood_poisson(df_species,first_moment),
              log_likelihood_negbin(df_species,negbin_p,negbin_k))

# add this vector to the dataframe
Table_Df = cbind(Table_Df,newvector)
}
```

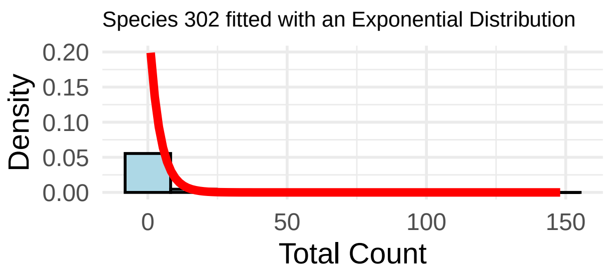

Species 302 fitted with a Gamma Distribution
Density
0.125
0.100
0.075
0.050
0.025
0.000
0
50
100
150
Total Count

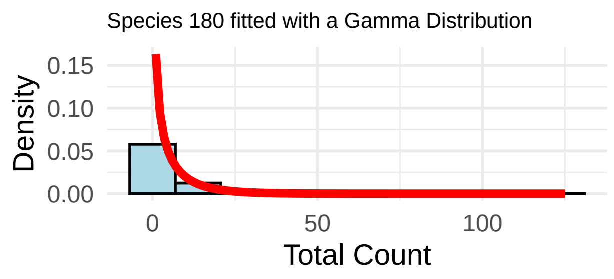

Species 302 fitted with a Normal Distribution
Density
0.04
0.02
0.00
0
50
100
150
Total Count

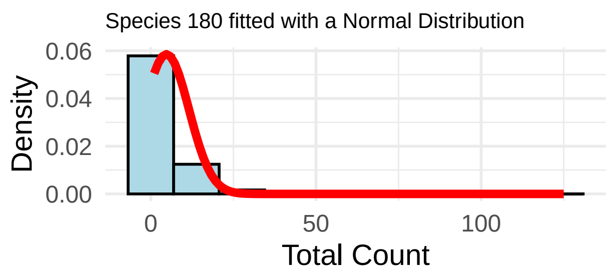

Species 302 fitted with a Geometric Distribution
Density
0.20
0.15
0.10
0.05
0.00
0
50
100
150
Total Count

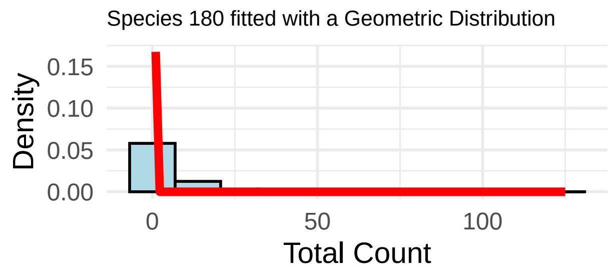

Species 302 fitted with a Poisson Distribution
Density
0.08
0.06
0.04
0.02
0.00
0
50
100
150
Total Count

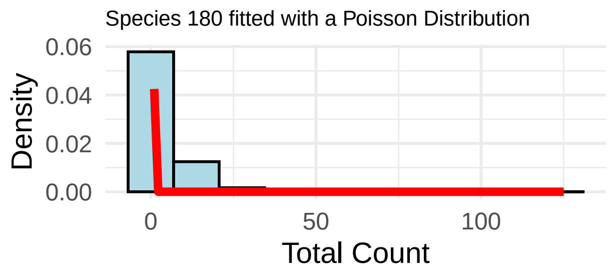

Species 302 fitted with a Negative Binomial Distribution
Density
0.09
0.06
0.03
0.00
0
50
100
150
Total Count

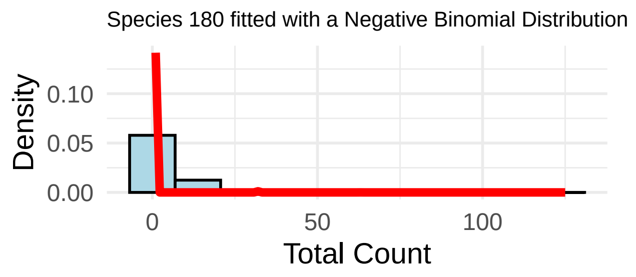

Species 302 fitted with an Exponential Distribution
Density
0.20
0.15
0.10
0.05
0.00
0
50
100
150
Total Count

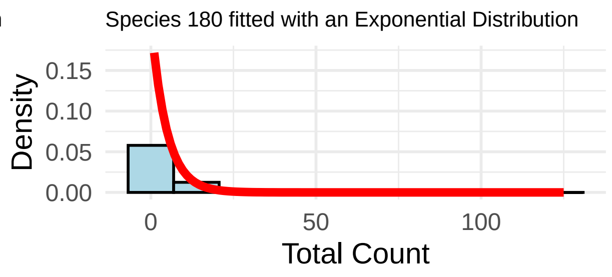

Species 180 fitted with a Gamma Distribution
Density
0.15
0.10
0.05
0.00
0
50
100
Total Count

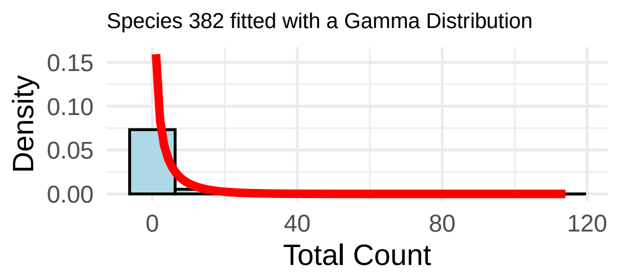

Species 180 fitted with a Normal Distribution
Density
0.06
0.04
0.02
0.00
0
50
100
Total Count

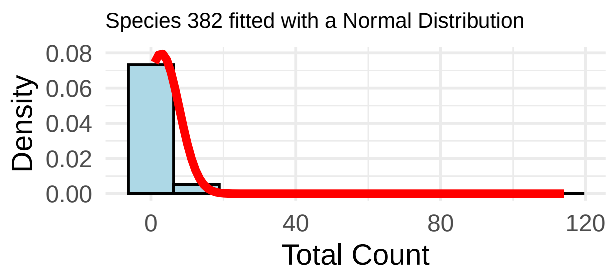

Species 180 fitted with a Geometric Distribution
Density
0.15
0.10
0.05
0.00
0
50
100
Total Count

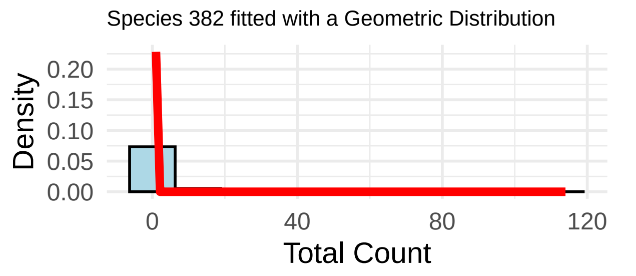

Species 180 fitted with a Poisson Distribution
Density
0.06
0.04
0.02
0.00
0
50
100
Total Count

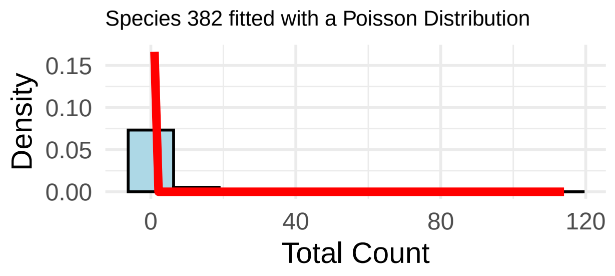

Species 180 fitted with a Negative Binomial Distribution
Density
0.10
0.05
0.00
0
50
100
Total Count

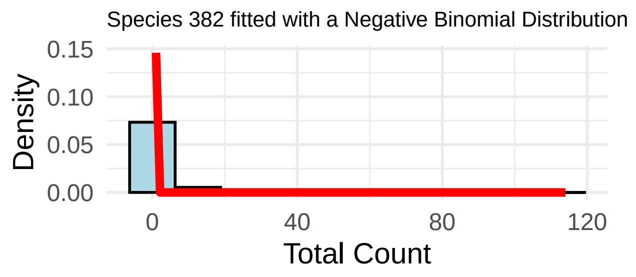

Species 180 fitted with an Exponential Distribution
Density
0.15
0.10
0.05
0.00
0
50
100
Total Count

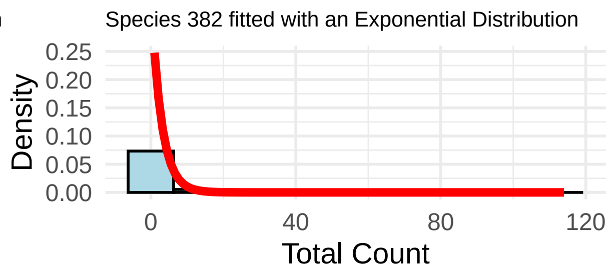

Species 382 fitted with a Gamma Distribution
Density
0.15
0.10
0.05
0.00
0
40
80
120
Total Count

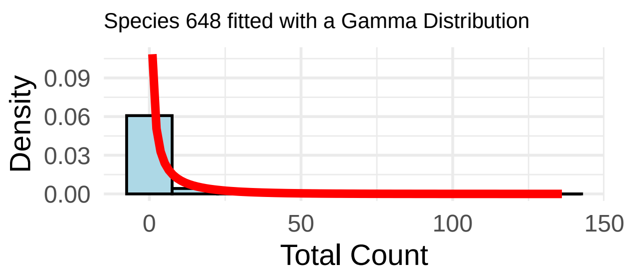

Species 382 fitted with a Normal Distribution
Density
0.08
0.06
0.04
0.02
0.00
0
40
80
120
Total Count

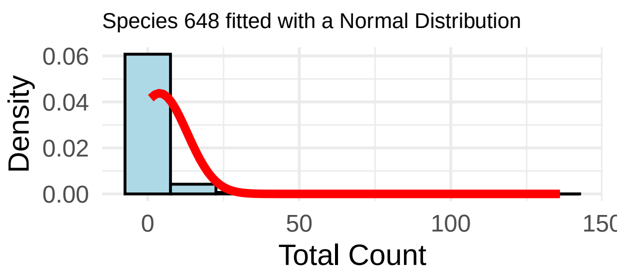

Species 382 fitted with a Geometric Distribution
Density
0.20
0.15
0.10
0.05
0.00
0
40
80
120
Total Count

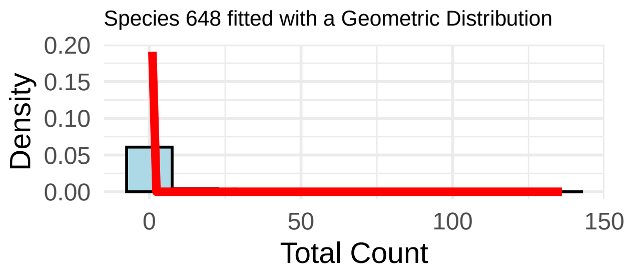

Species 382 fitted with a Poisson Distribution
Density
0.15
0.10
0.05
0.00
0
40
80
120
Total Count

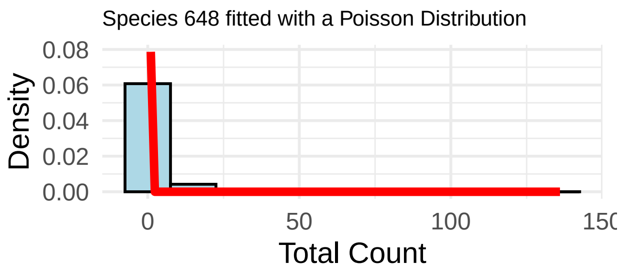

Species 382 fitted with a Negative Binomial Distribution
Density
0.15
0.10
0.05
0.00
0
40
80
120
Total Count

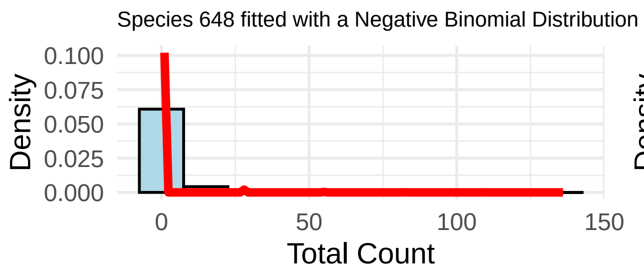

Species 382 fitted with an Exponential Distribution
Density
0.25
0.20
0.15
0.10
0.05
0.00
0
40
80
120
Total Count

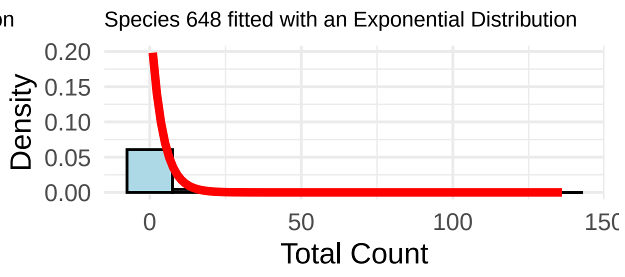

Species 648 fitted with a Gamma Distribution
Density
0.09
0.06
0.03
0.00
0
50
100
150
Total Count

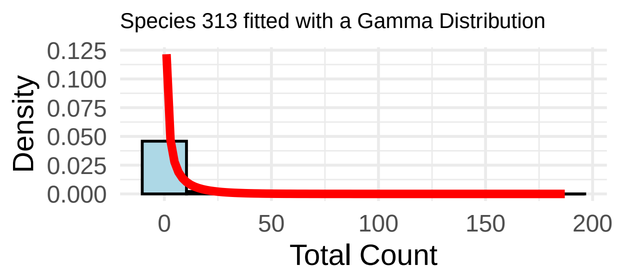

Species 648 fitted with a Normal Distribution
Density
0.06
0.04
0.02
0.00
0
50
100
150
Total Count

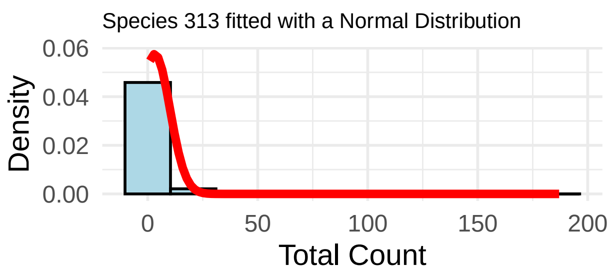

Species 648 fitted with a Geometric Distribution
Density
0.20
0.15
0.10
0.05
0.00
0
50
100
150
Total Count

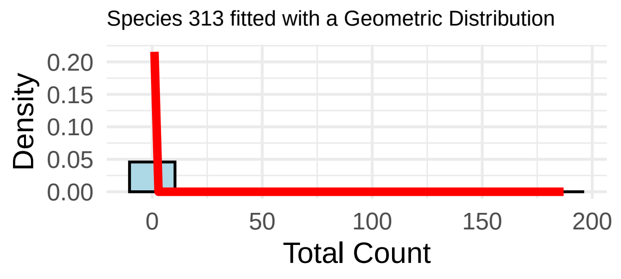

Species 648 fitted with a Poisson Distribution
Density
0.08
0.06
0.04
0.02
0.00
0
50
100
150
Total Count

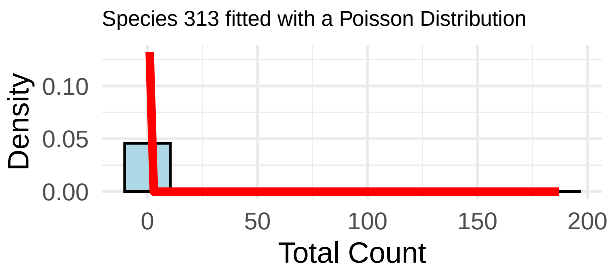

Species 648 fitted with a Negative Binomial Distribution
Density
0.100
0.075
0.050
0.025
0.000
0
50
100
150
Total Count

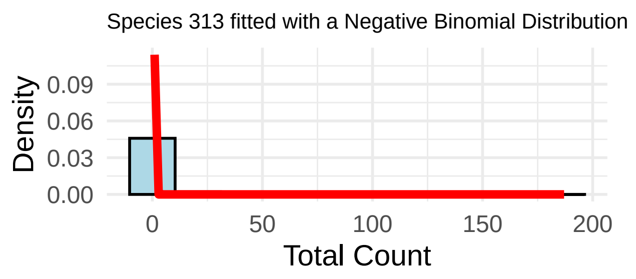

Species 648 fitted with an Exponential Distribution
Density
0.20
0.15
0.10
0.05
0.00
0
50
100
150
Total Count

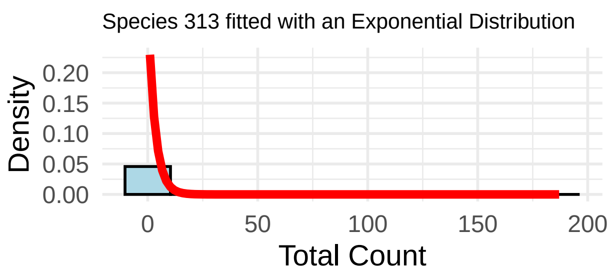












```r
# add column names to the data frame
names(Table_Df) = c("Distribution",specieslist)
#print the data frame as a table
print(Table_Df)
```

```
  Distribution        302        180        382        648        313
1  Exponential  -8125.622  -6979.351  -8400.924  -5775.535  -5992.432
2        Gamma  -9825.653  -7457.103 -10091.000  -7163.225  -7593.934
3    Geometric  -8663.817  -7319.301  -9358.655  -6155.020  -6549.621
4       Normal -12215.411  -9138.488 -12458.292  -8869.914  -9320.963
5      Poisson -15784.999 -11376.855 -12182.927 -11950.259 -10280.073
6      Neg Bin  -9941.636  -7597.821 -10251.078  -7235.375  -7677.741
```

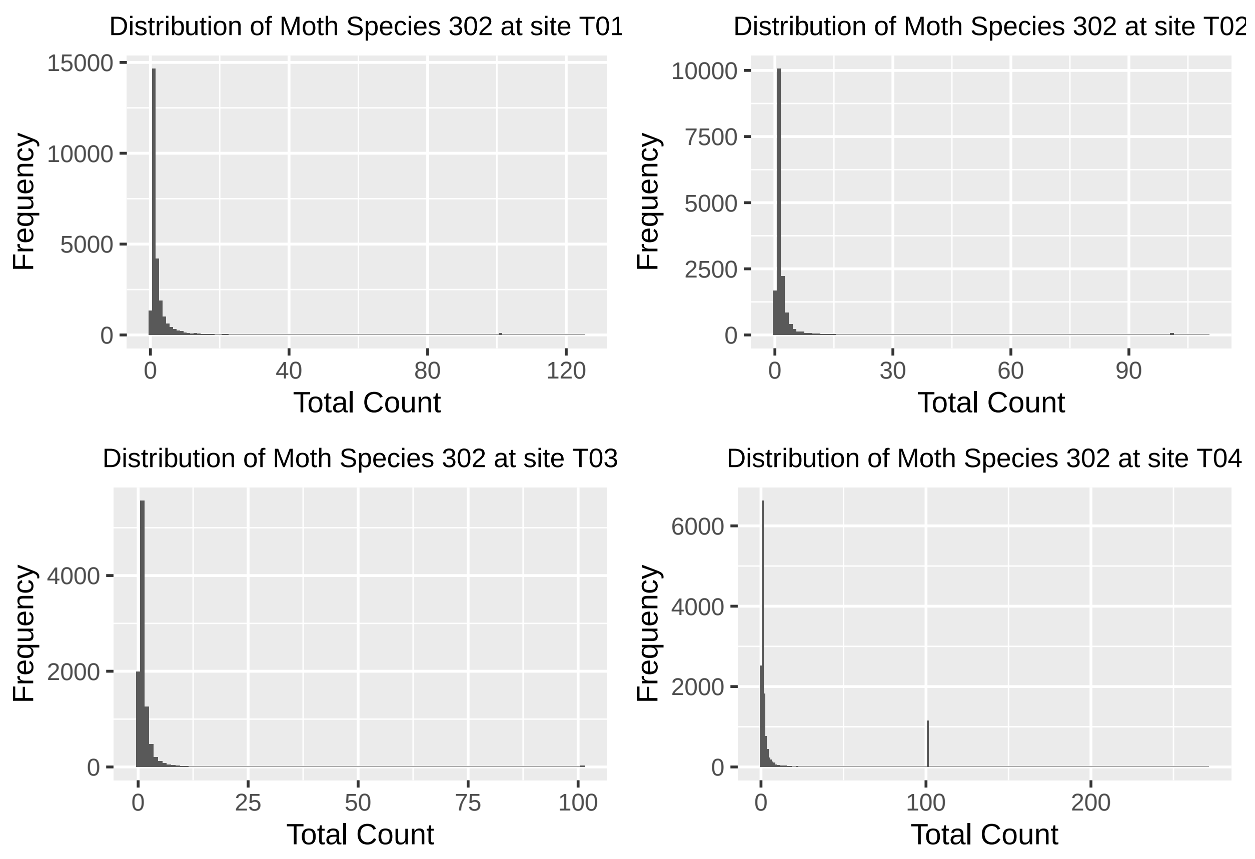
We note that the distribution with the best log likelihood across all species was the Exponential distribution. However, we note that none of the distribution families provide a good log likelihood since they are all large negative numbers. This fits into the explanation of part a) since the distribution of moth counts is difficult to be fitted by a distribution family. In the context of the ECN moth dataset, this may be because moth counts are measured by light traps which are emptied daily, but if not possible the counts are accumulated, perhaps leading to outliers that make it impossible for a given parametric distribution to provide a good fit for.

c) Most frequent species, locations separate

First state what the most frequent of your 5 moths species (overall) is.

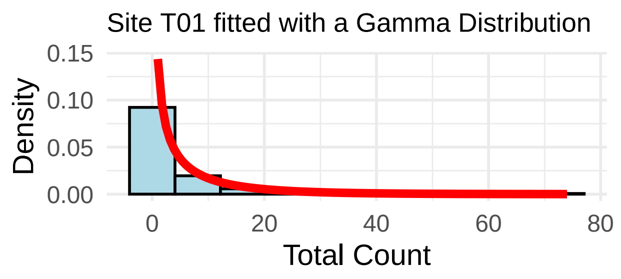
Species 302 was the most frequently occurring out of the five given to the report group.

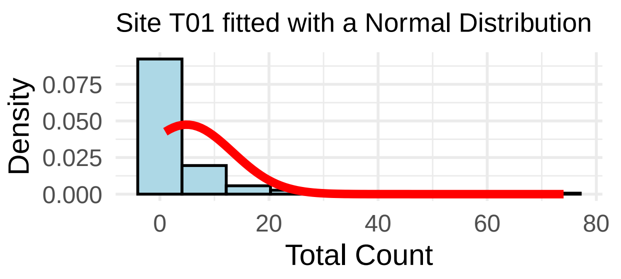
```r
#create an empty data frame to store parameters of distributions
parameter_table = data.frame(
  test = rep(5,10)
)

# Filter data for each site in the sitelist
filtered_data <- lapply(sitelist, function(site){
  Moths %>%
    filter(SITECODE == site)
})

# Plot histograms for each species in a grid
plots <- lapply(filtered_data, function(df){
  ggplot(df, aes(x = VALUE)) +
    geom_histogram(binwidth = 1) +
    labs(title = paste("Distribution of Moth Species 302 at site", unique(df$SITECODE)),
         x = "Total Count",
         y = "Frequency") +
    theme(plot.title = element_text(size = 10,hjust =.5))
})

grid.arrange(grobs = plots, ncol = 2)
```

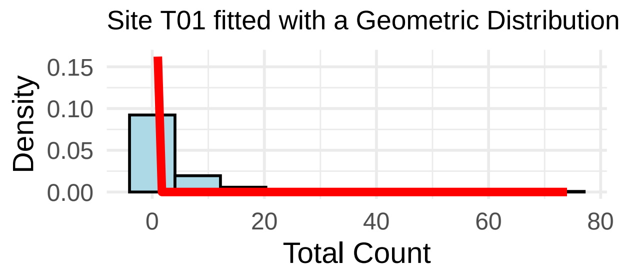


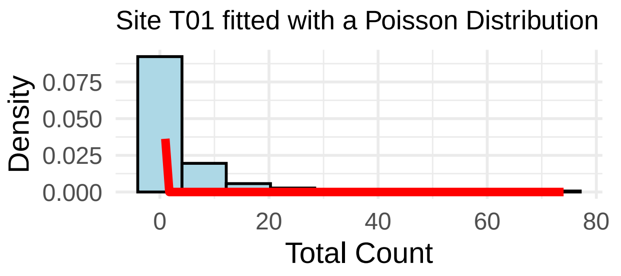
```r
#Create a table for the log likelihood of each distribution
Table_Df = data.frame(
  Distribution = c("Exponential","Gamma","Geometric","Normal","Poisson","Neg Bin")
)

sitelist = c("T01","T02","T03","T04")

# for each site in the sitelist
for (site in sitelist){

# create a new dataframe consisting only of the site and species 302
df_species <- Moths %>%
  filter(FIELDNAME == 302,SITECODE==site)

# calculate the parameters of each distribution from the sample data
first_moment = mean(df_species$VALUE)
second_moment = mean((df_species$VALUE)^2)
normal_sd = sd(df_species$VALUE)

geometric_p = 1/first_moment

betadist_alpha = (first_moment^2-first_moment*second_moment)/(second_moment-first_moment^2)
betadist_beta = betadist_alpha*((1-first_moment)/first_moment)

gammadist_alpha = first_moment^2/(second_moment-first_moment^2)
gammadist_beta = first_moment/(second_moment-first_moment^2)

binomial_p = 1 - var(df_species$VALUE)/first_moment

negbin_p = first_moment/normal_sd^2
negbin_k = first_moment^2/(normal_sd^2-first_moment)

parameters = c(first_moment,second_moment,normal_sd,geometric_p,betadist_alpha,
               betadist_alpha,gammadist_alpha,gammadist_beta,negbin_p,negbin_k)

#create a table with parameters of distributions for each site
parameter_table = cbind(parameter_table,parameters)

# add the log likelihoods of each distribution to a new vector
newvector = c(log_likelihood_exponential(df_species,geometric_p),
              log_likelihood_gamma(df_species,gammadist_alpha,gammadist_beta),
              log_likelihood_geometric(df_species,geometric_p),
              log_likelihood_normal(df_species,first_moment,normal_sd),
              log_likelihood_poisson(df_species,first_moment),
              log_likelihood_negbin(df_species,negbin_p,negbin_k))

# add this vector to the data frame
Table_Df = cbind(Table_Df,newvector)
}
# add column names to the data frame
names(Table_Df) = c("Distribution",sitelist)
```

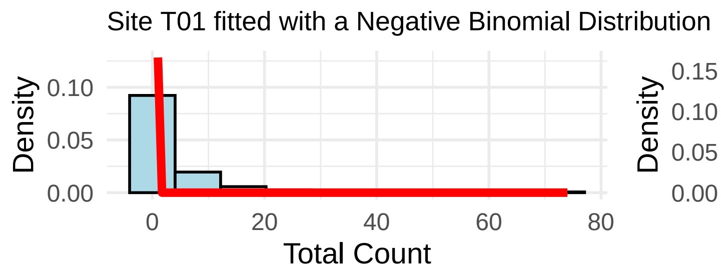
```r
# remove first column of parameter table since it is redundant
parameter_table = parameter_table[c(-1)]

# assign appropriate column names to the parameter table
names(parameter_table) = sitelist

# for each site in the list of sites
for (site in sitelist) {

# create a new data frame for moth species 302 in that given site
df_species <- Moths %>%
  filter(FIELDNAME == 302,SITECODE==site)

gammaplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dgamma,
args = list(shape = parameter_table[[site]][7],
rate = parameter_table[[site]][8]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Site", site, "fitted with a Gamma Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 10))

poissonplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dpois,
args = list(lambda = parameter_table[[site]][1]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Site", site, "fitted with a Poisson Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 10))

normalplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dnorm,
args = list(mean = parameter_table[[site]][1],
sd = parameter_table[[site]][3]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Site", site, "fitted with a Normal Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 10))

geomplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dgeom,
args = list(parameter_table[[site]][4]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Site", site, "fitted with a Geometric Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 10))

negbinplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dnbinom,
args = list(size = parameter_table[[site]][10], prob = parameter_table[[site]][9]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Site", site, "fitted with a Negative Binomial Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 10))

exponplot <- ggplot(data = df_species,
aes(x = VALUE)) +
geom_histogram(aes(y = ..density..), bins = 10,
fill = "lightblue", color = "black") +
stat_function(fun = dexp,
args = list(rate = 1/parameter_table[[site]][1]),
color = "red", linewidth = 1.5) +
theme_minimal() +
labs(title = paste("Site", site, "fitted with an Exponential Distribution"),
x = "Total Count", y = "Density") +
theme(plot.title = element_text(size = 10))

grid.arrange(gammaplot,normalplot,geomplot ,poissonplot, negbinplot, exponplot, ncol = 2)
}
```

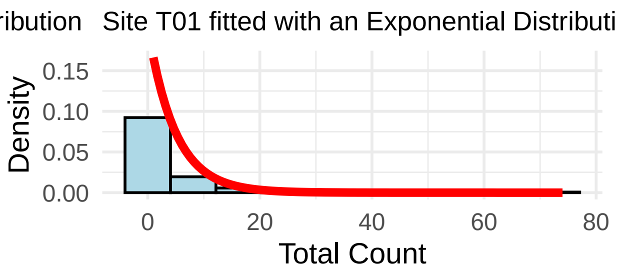

Site T01 fitted with a Gamma Distribution
Density
0.15
0.10
0.05
0.00
0
20
40
60
80
Total Count

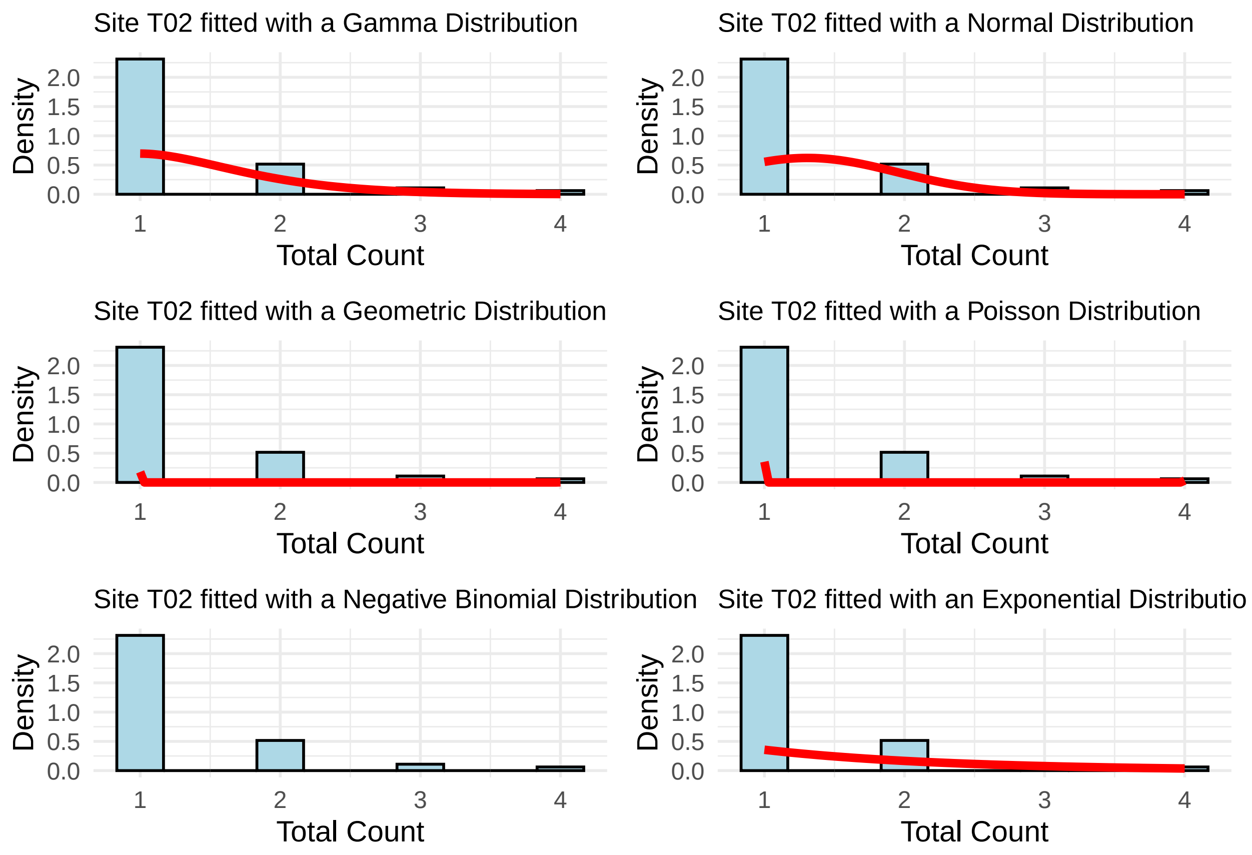

Site T01 fitted with a Normal Distribution
Density
0.075
0.050
0.025
0.000
0
20
40
60
80
Total Count

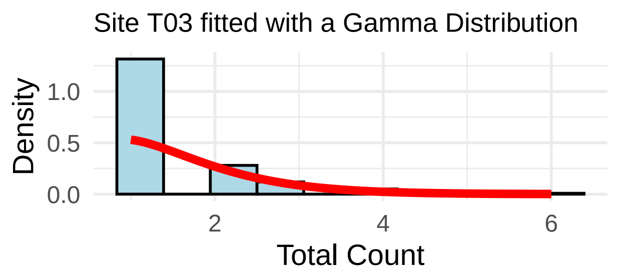

Site T01 fitted with a Geometric Distribution
Density
0.15
0.10
0.05
0.00
0
20
40
60
80
Total Count

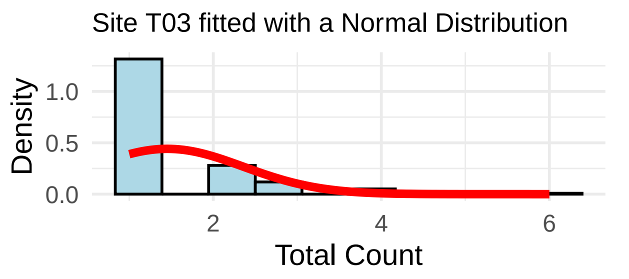

Site T01 fitted with a Poisson Distribution
Density
0.075
0.050
0.025
0.000
0
20
40
60
80
Total Count

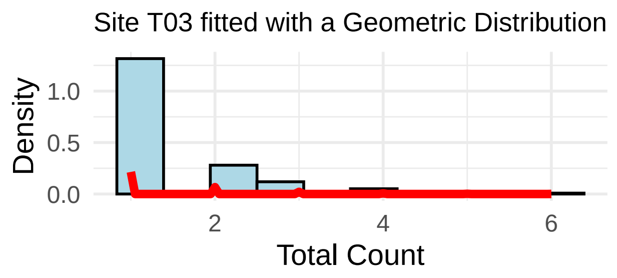

Site T01 fitted with a Negative Binomial Distribution
Density
0.10
0.05
0.00
0
20
40
60
80
Total Count

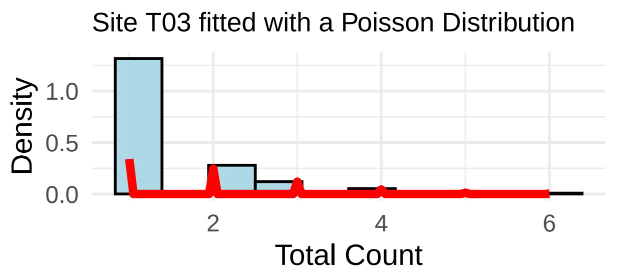

Site T01 fitted with an Exponential Distributi
Density
0.15
0.10
0.05
0.00
0
20
40
60
80
Total Count

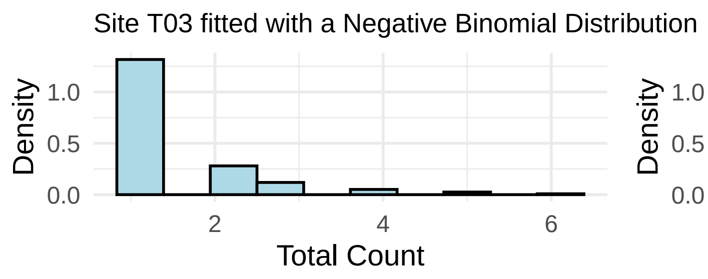

Site T02 fitted with a Gamma Distribution
Site T02 fitted with a Normal Distribution
Site T02 fitted with a Geometric Distribution
Site T02 fitted with a Poisson Distribution
Site T02 fitted with a Negative Binomial Distribution
Site T02 fitted with an Exponential Distributio
Density
Total Count
0.0
0.5
1.0
1.5
2.0
1
2
3
4

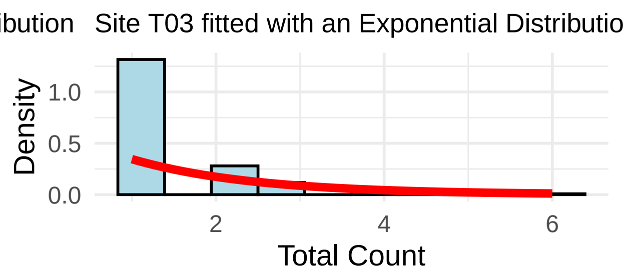

Site T03 fitted with a Gamma Distribution
Density
1.0
0.5
0.0
2
4
6
Total Count

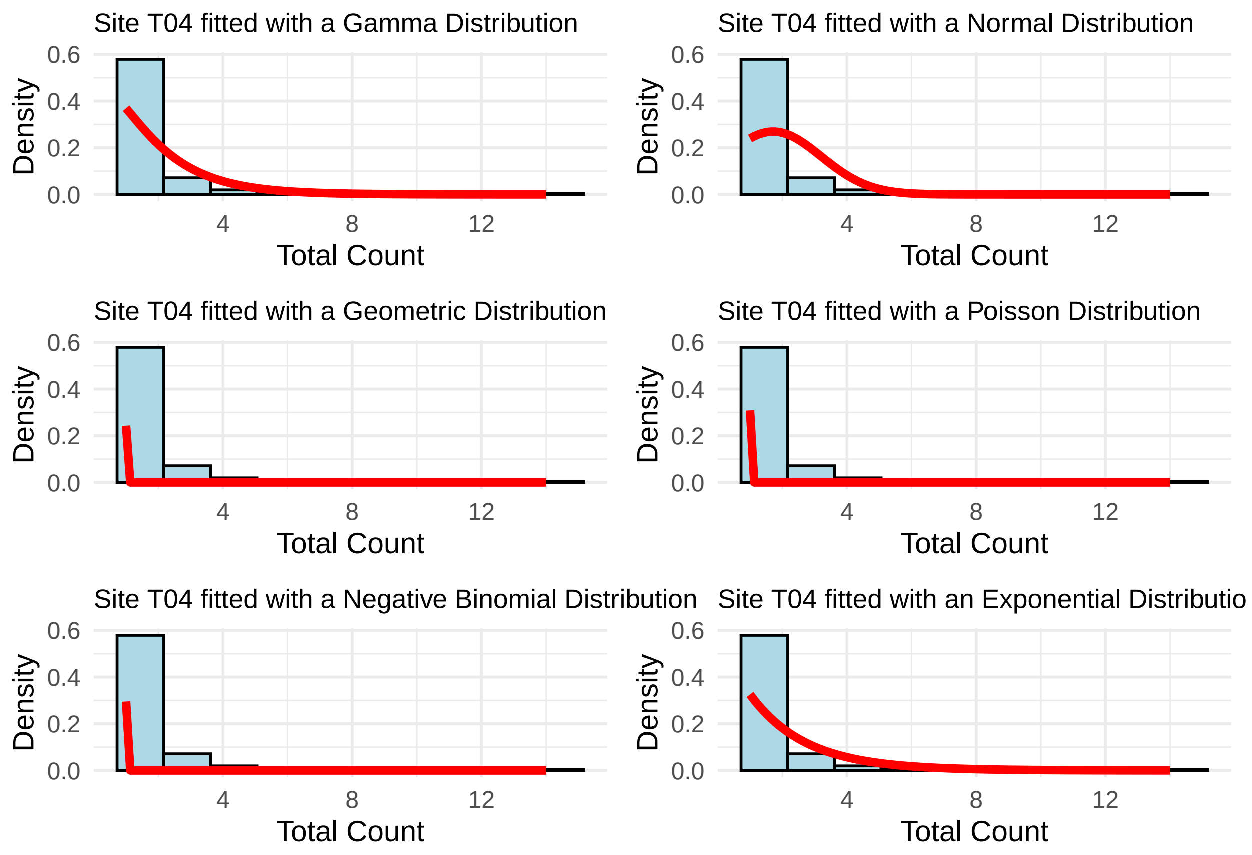

Site T03 fitted with a Normal Distribution
Density
1.0
0.5
0.0
2
4
6
Total Count


Site T03 fitted with a Geometric Distribution
Density
1.0
0.5
0.0
2
4
6
Total Count


Site T03 fitted with a Poisson Distribution
Density
1.0
0.5
0.0
2
4
6
Total Count


Site T03 fitted with a Negative Binomial Distribution
Density
1.0
0.5
0.0
2
4
6
Total Count


Site T03 fitted with an Exponential Distributio
Density
1.0
0.5
0.0
2
4
6
Total Count



```r
# print data frame as a table
print(Table_Df)
```

```
  Distribution        T01       T02       T03       T04
1  Exponential  -948.1903 -243.4479 -291.1849 -329.4286
2        Gamma -1033.4093 -143.3397 -221.1776 -307.1542
3    Geometric  -991.4763 -414.8746 -438.2364 -436.5256
4       Normal -1298.0514 -187.0266 -278.9775 -387.2920
5      Poisson -1864.0123 -231.8710 -280.9264 -333.9515
6      Neg Bin -1051.7229       NaN       NaN -334.7961
```

We see that the Gamma distribution provided the best log likelihood for species 302 across all sitecodes with the exception of site T01 where the exponential distribution had a better log likelihood. We can confirm that this is the case since the Gamma distribution is able to depict the sharp decline of moth species counts after 1. For T02 and T03, the negative binomial parameters that we calculated using the first and second moment (method of moments) from the data set give us values that are unable to fit a negative binomial distribution model. Hence in our log likelihood table in the T02, T03 columns and in the negative binomial row we have NaN. This is another example of how distributions are difficult to fit the moth data.