title: “Final Project”
output: pdf_document

Name: Mitchell Wang

I worked with:

Click the “Knit” button in RStudio to knit this file to a pdf.

Data Wrangling

```
library(readxl)
Covid_Carleton<- read.csv("Covid Data - Carleton.csv")
```

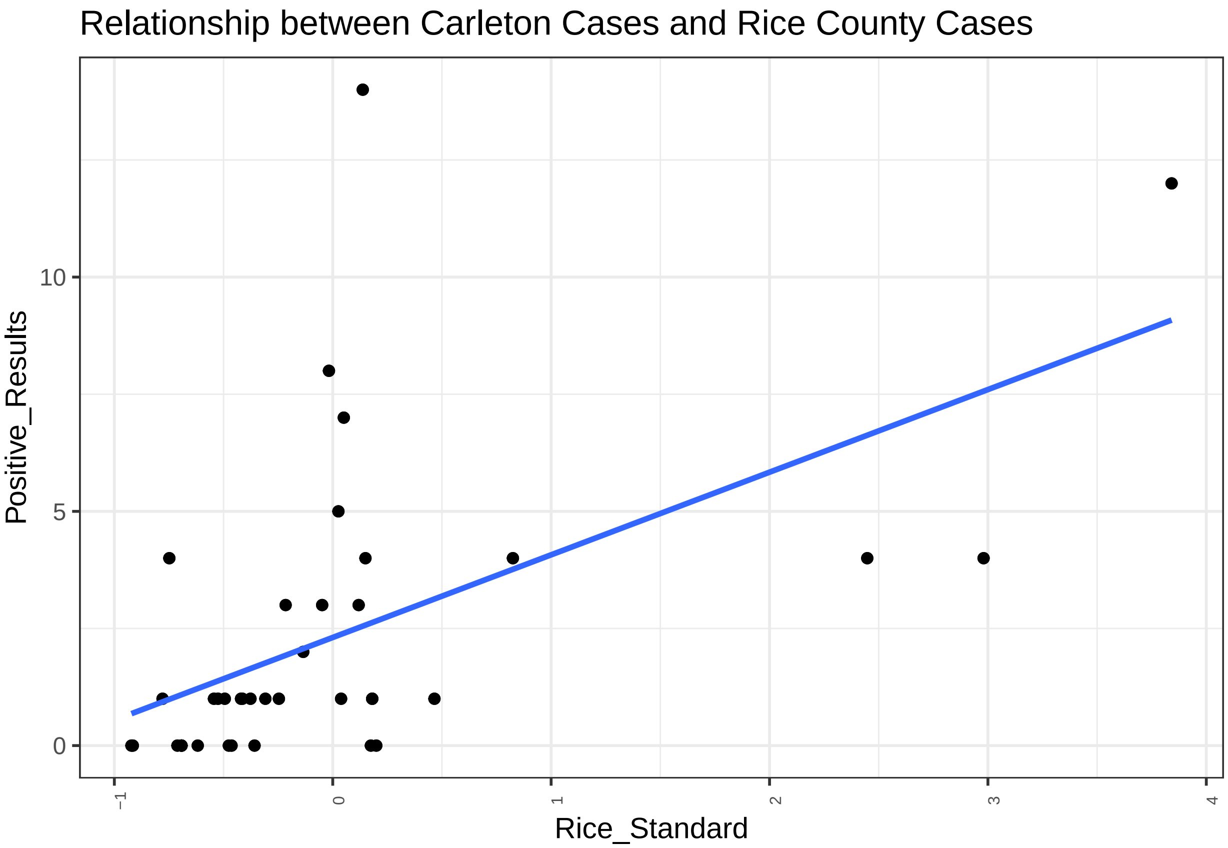
```
Covid_Carleton <- Covid_Carleton %>% filter(Term == "FA20"| Term == "WI21" | Term == "SP21" | Term == "
```

EDA

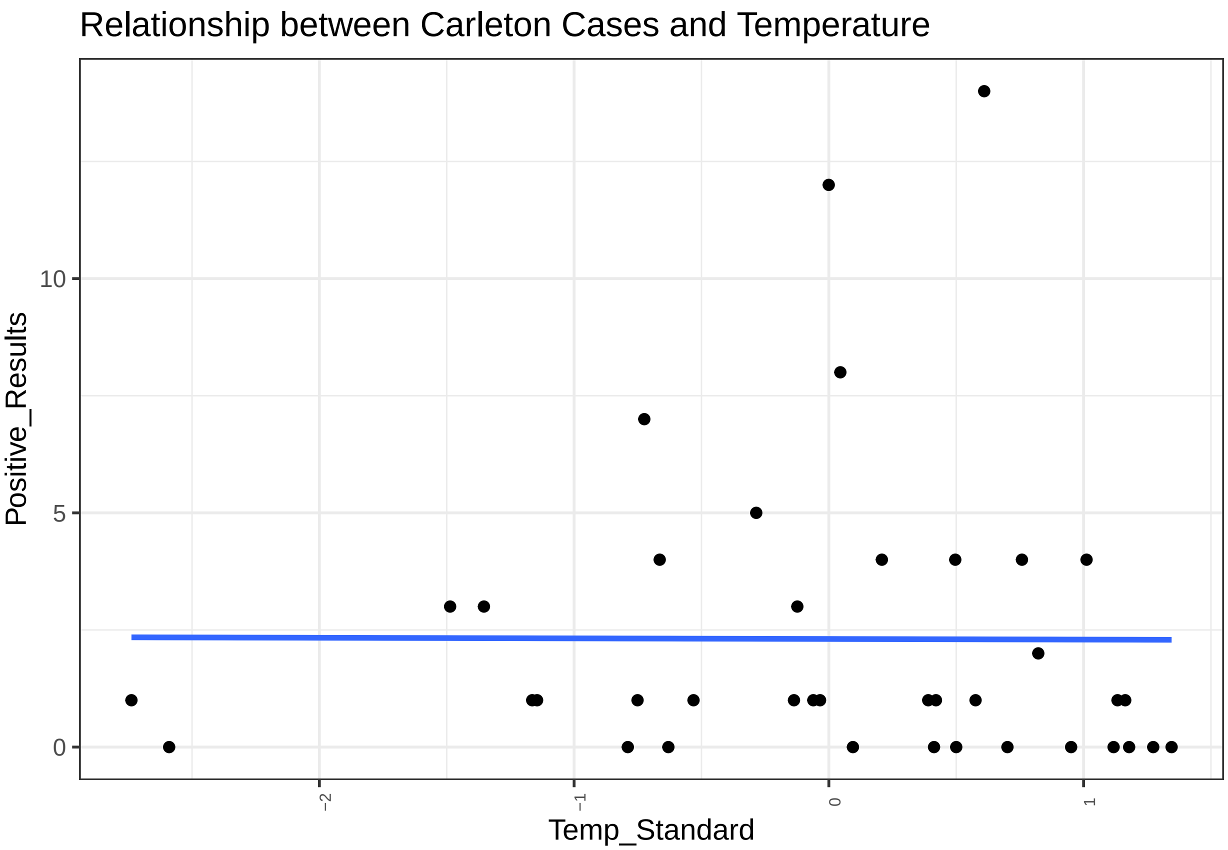
```
Covid_Summary <- Covid_Carleton %>% mutate(Term = recode(term, "1"="FA20","2"="WI21","3"="SP21","4"="FA

# at most 4 decimal places
knitr::kable(Covid_Summary, digits = 3)
```

term	Mean_Positive_Case	Mean_Positive_Rate_
1	2.900	NA
2	1.700	NA
3	2.100	NA
4	2.556	NA

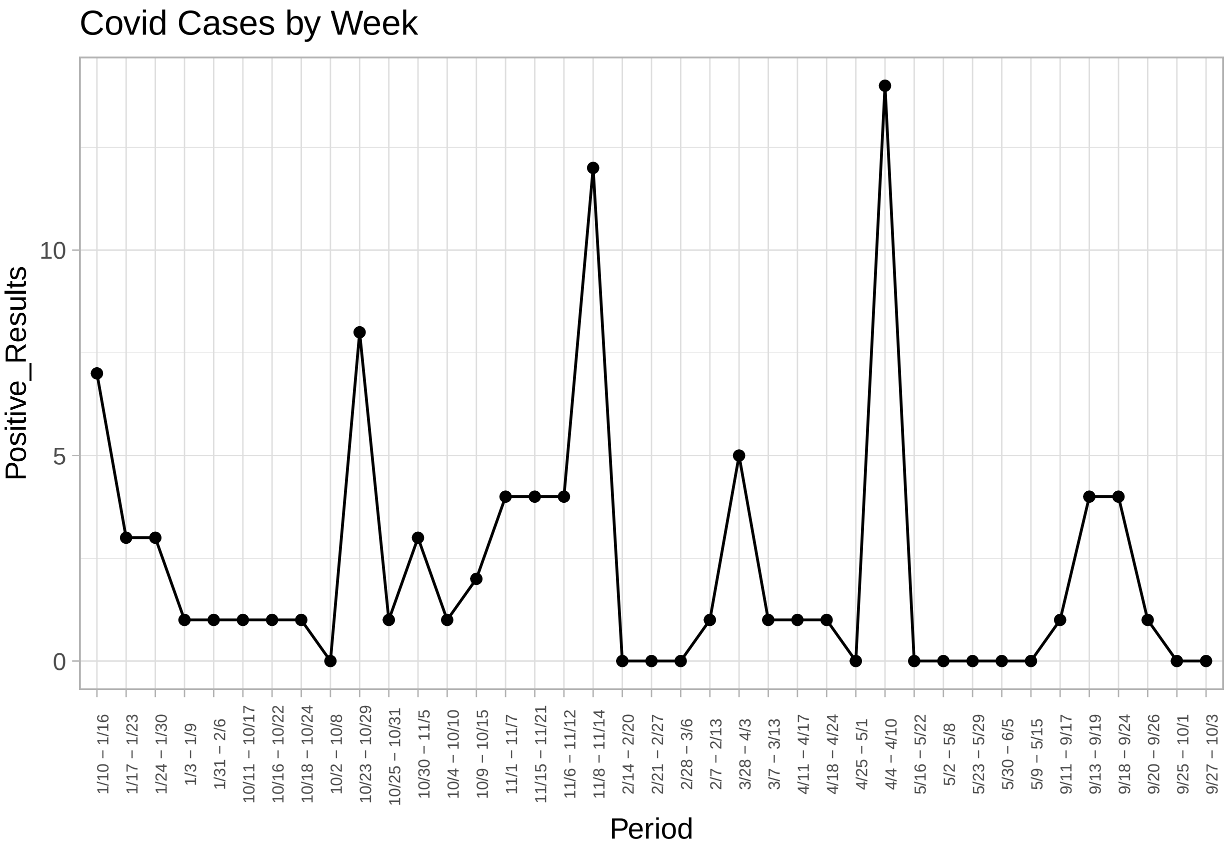
```
Covid_Carleton$Rice_Standard <- as.vector(scale(Covid_Carleton$Rice_County))
ggplot(Covid_Carleton, aes(Rice_Standard, Positive_Results, group = 1)) +
  geom_point() + theme_bw() +
  geom_point() + geom_smooth(method = "lm", se = F) +
  xaxis_text(angle = 90, size = 6) +
  ggtitle("Relationship between Carleton Cases and Rice County Cases")
```



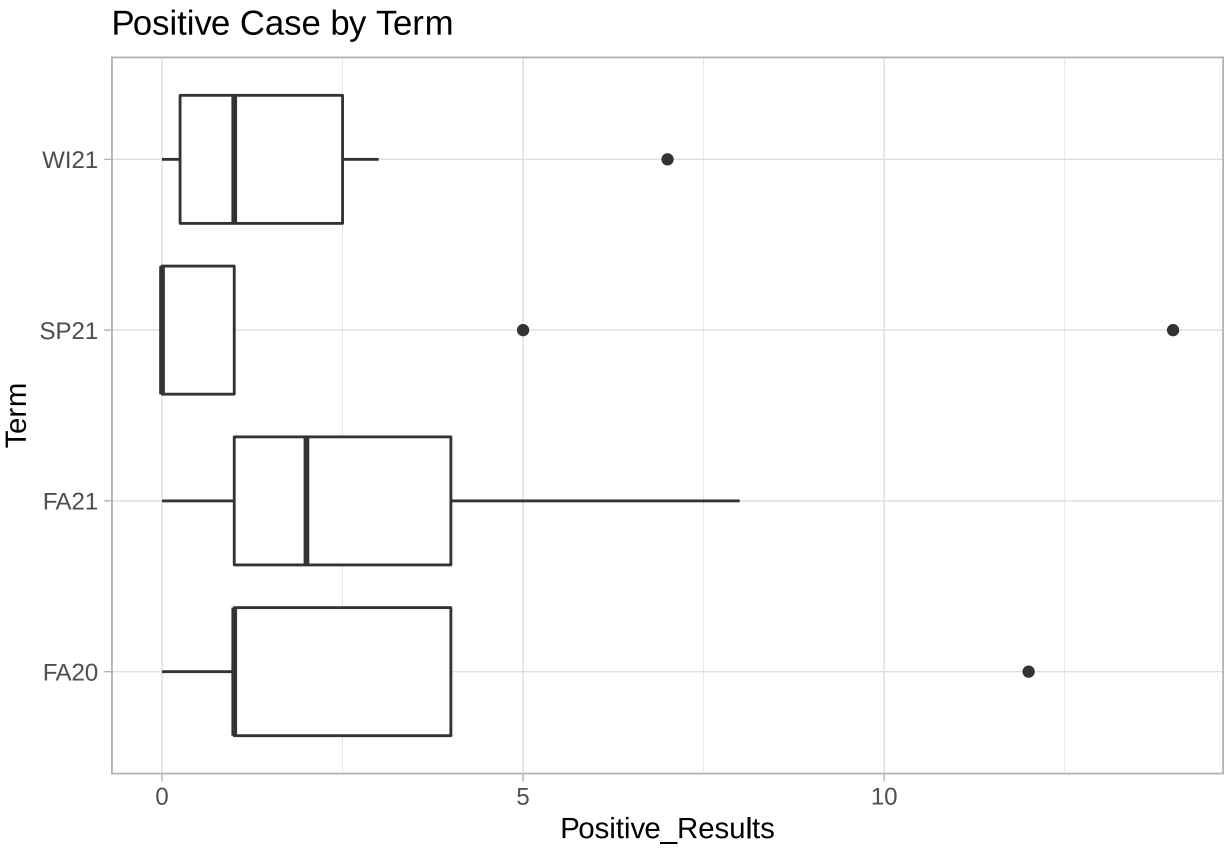
```
Covid_Carleton$Temp_Standard <- as.vector(scale(Covid_Carleton$Ave_temp))
ggplot(Covid_Carleton, aes(Temp_Standard, Positive_Results, group = 1)) +
  geom_point() + geom_smooth(method = "lm", se = F) +
  theme_bw() +
  xaxis_text(angle = 90, size = 6)+
  ggtitle("Relationship between Carleton Cases and Temperature")
```



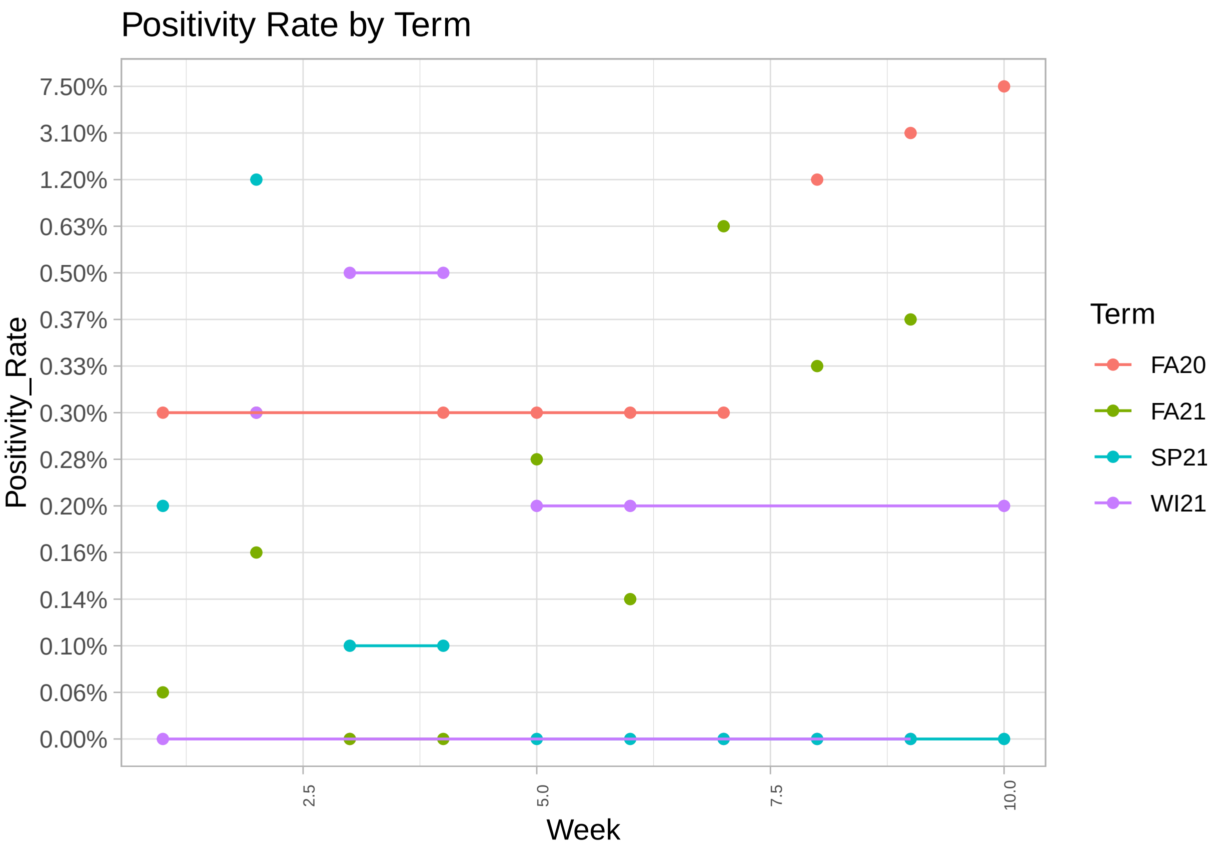
```
ggplot(Covid_Carleton, aes(Period, Positive_Results, group = 1)) +
  geom_point() +  geom_line() +
  theme_light() +
  xaxis_text(angle = 90, size = 6)+ggtitle("Covid Cases by Week")
```



```
Covid_Carleton  %>%
  group_by(Term) %>%
  ggplot(aes(Term, Positive_Results)) + geom_boxplot() +
  theme_light() +
  coord_flip()+
  ggtitle("Positive Case by Term")
```



```
ggplot(Covid_Carleton, aes(Week, Positivity_Rate, color = Term)) +
  geom_point() + geom_line() +
  theme_light()  +
  xaxis_text(angle = 90, size = 6)+
  ggtitle("Positivity Rate by Term")
```



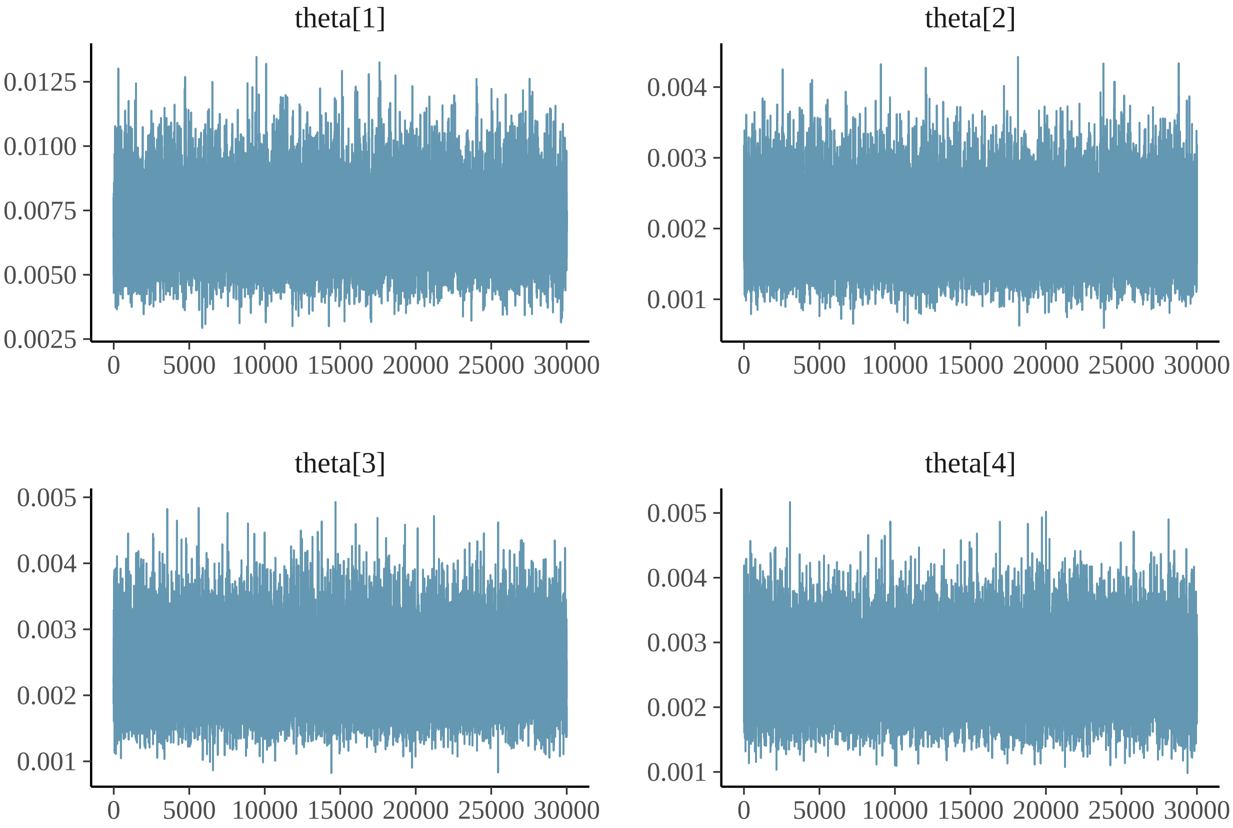
## JAGS

```
modelString <-"
model{
## sampling
for (i in 1:N){
 y[i] ~ dbin(theta[term[i]], n[i])}

## priors
for (j in 1:M){
 theta[j] ~ dbeta(alpha, beta)}

alpha <- mu / pow(eta,2)
beta <- (1-mu) / pow(eta,2)
mu ~ dbeta(1,1)
eta <- exp(logeta)
logeta ~ dlogis(log(39), 1)
}"
```

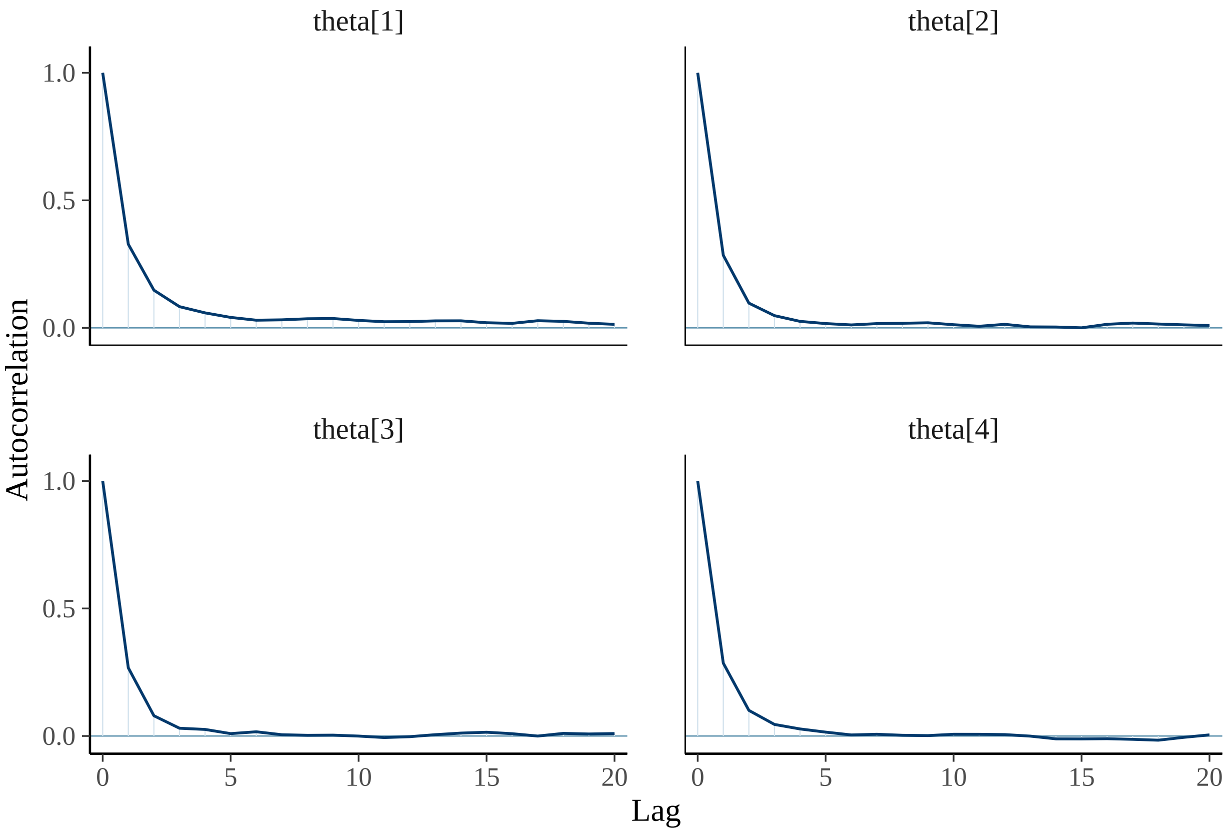
```
y <- Covid_Carleton$Positive_Results
n <- Covid_Carleton$Total_Tests
N <- length(y)
M <- length(unique(Covid_Carleton$Term))
term <- Covid_Carleton$term
the_data <- list(y = y, n=n, N=N,M=M, term = term)

init = list(
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987654),
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987653),
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987652)
            )

carleton_post <- run.jags(
  model = modelString,
  n.chains = 3,
  data = the_data,
  monitor = c("theta"),
  adapt = 1000,
  burnin = 2000,
  sample = 10000,
  silent.jags = TRUE  # Eliminates progress bar
)
## Finished running the simulation
```

Diagnostics

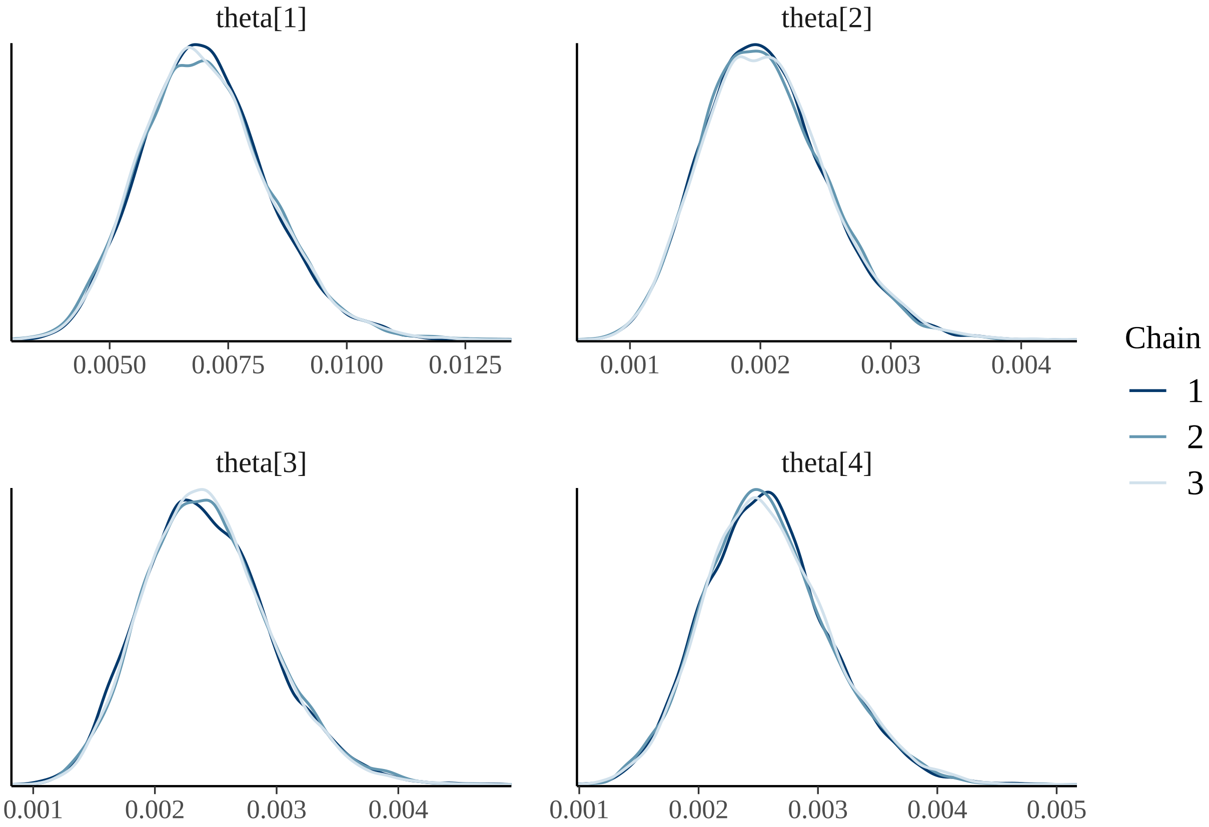
```
post_mcmc <- as.data.frame(as.mcmc(carleton_post))
mcmc_trace(post_mcmc)
```



```
mcmc_acf(post_mcmc)
```



```
mcmc_dens_overlay(carleton_post$mcmc)
```



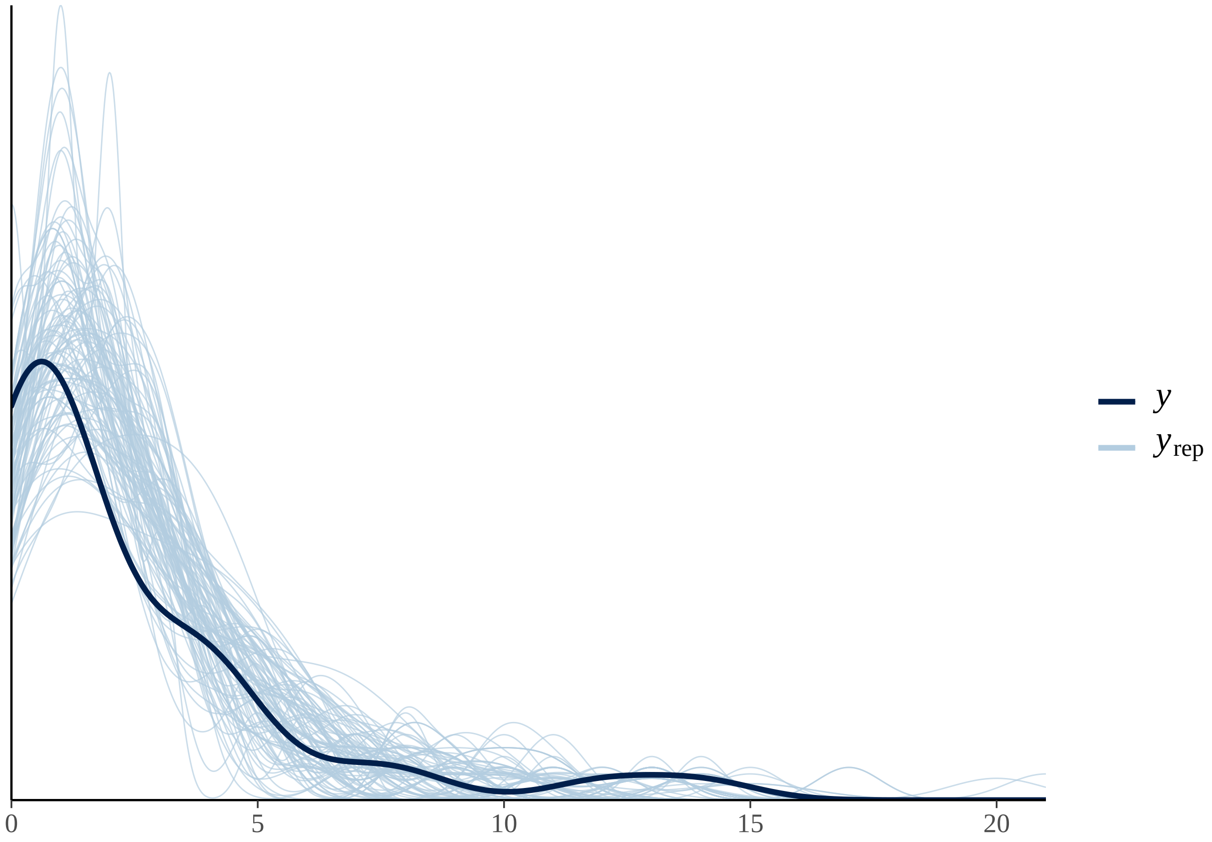
```
summary(post_mcmc)
##     theta[1]           theta[2]            theta[3]            theta[4]        
##  Min.   :0.002928   Min.   :0.0005936   Min.   :0.0008209   Min.   :0.0009809  
##  1st Qu.:0.006077   1st Qu.:0.0017006   1st Qu.:0.0020655   1st Qu.:0.0022118  
##  Median :0.006940   Median :0.0020090   Median :0.0023992   Median :0.0025454  
##  Mean   :0.007021   Mean   :0.0020402   Mean   :0.0024315   Mean   :0.0025783  
##  3rd Qu.:0.007860   3rd Qu.:0.0023429   3rd Qu.:0.0027567   3rd Qu.:0.0029055  
##  Max.   :0.013471   Max.   :0.0044274   Max.   :0.0049289   Max.   :0.0051699  
```

Posterior Predictive

```
modelString <-"
model{
## sampling
for (i in 1:N){
 y[i] ~ dbin(theta[term[i]], n[i])}

## priors
for (j in 1:M){
 theta[j] ~ dbeta(alpha, beta)}

alpha <- mu / pow(eta,2)
beta <- (1-mu) / pow(eta,2)
mu ~ dbeta(1,1)
eta <- exp(logeta)
logeta ~ dlogis(log(100), 1)
for (i in 1:N) {
   y_pred[i] ~ dbin(theta[term[i]], n[i])
}

}
"
```

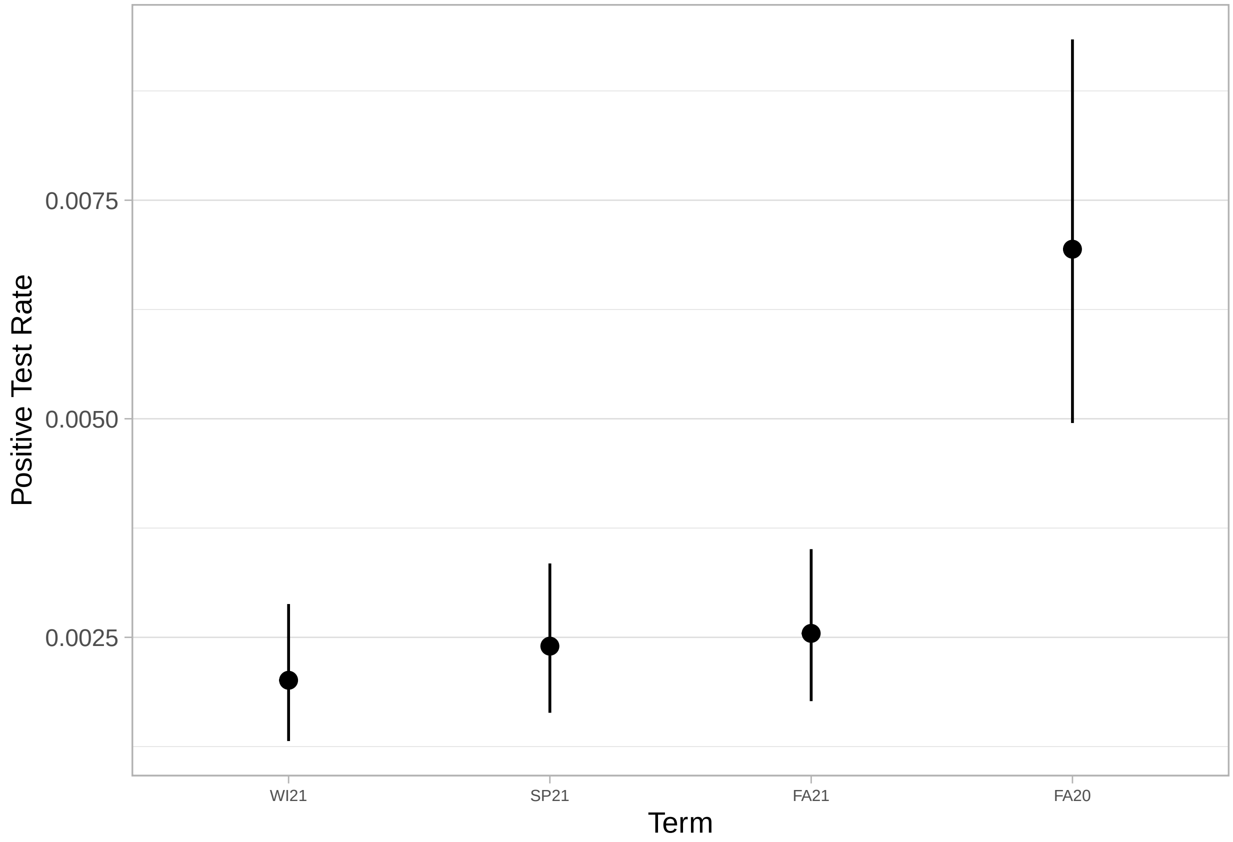
```
y <- Covid_Carleton$Positive_Results
n <- Covid_Carleton$Total_Tests
N <- length(y)
M <- length(unique(Covid_Carleton$Term))
term <- Covid_Carleton$term
the_data <- list(y = y, n=n, N=N,M=M, term = term)

init = list(
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987654),
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987653),
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987652)
            )
carleton_pred <- run.jags(
  model = modelString,
  n.chains = 3,
  data = the_data,
  monitor = c("y_pred"),
  adapt = 1000,
  burnin = 2000,
  sample = 10000,
  thin=5,
  silent.jags = TRUE  # Eliminates progress bar
)
## Finished running the simulation
```

```
ppc_dens_overlay(y = Covid_Carleton$Positive_Results, yrep = carleton_pred$mcmc[[1]][1:100,])
```



Inference

```
post_intervals <- mcmc_intervals_data(carleton_post$mcmc) %>% mutate(para = c("FA20","WI21","SP21","FA2
head(post_intervals)
## # A tibble: 4 x 10
##   parameter outer_width inner_width point_est      ll       l       m       h
##   <fct>           <dbl>       <dbl> <chr>       <dbl>   <dbl>   <dbl>   <dbl>
## 1 theta[1]          0.9         0.5 median    0.00495 0.00608 0.00694 0.00786
## 2 theta[2]          0.9         0.5 median    0.00131 0.00170 0.00201 0.00234
## 3 theta[3]          0.9         0.5 median    0.00164 0.00207 0.00240 0.00276
## 4 theta[4]          0.9         0.5 median    0.00177 0.00221 0.00255 0.00291
## # ... with 2 more variables: hh <dbl>, para <chr>

slice(post_intervals, 1:4) %>%
  ggplot(
    aes(x = reorder(para, (m)), y = (m), ymin = (ll), ymax = (hh))) +
  geom_pointrange() +
  theme_light() +
  xaxis_text(angle = 0, size = 6) +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank()
)+
xlab("Term") + ylab("Positive Test Rate")
```



```
quantile(post_mcmc$`theta[4]`/post_mcmc$`theta[1]`,c(0.025,0.975))
##      2.5%     97.5%
## 0.2079559 0.6388523
```

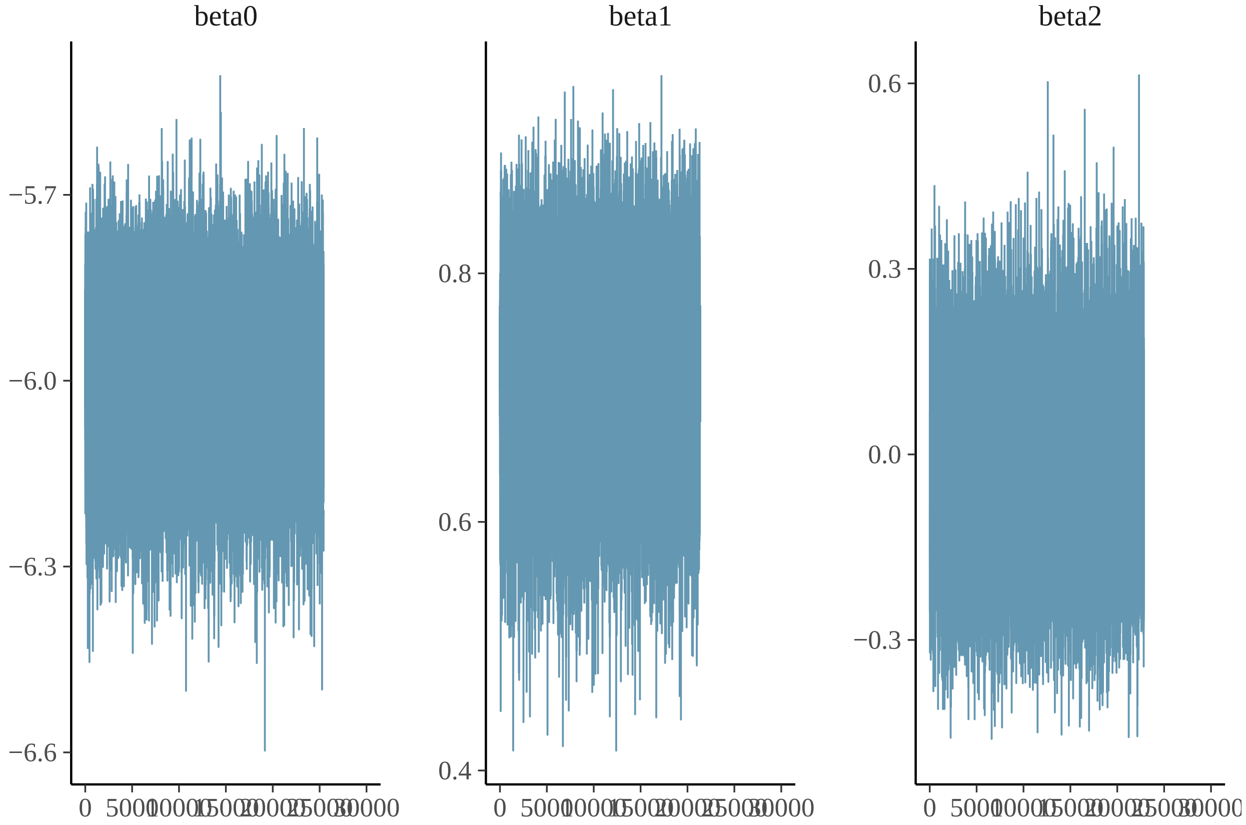
Logistical Regression

```
modelString <-"
model {
## likelihood
for (i in 1:n){
   y[i] ~ dbin(p[i], N[i])
}
## priors and regression
for (i in 1:n){
   logit(p[i]) <- beta0 +beta1*rice[i]+beta2*temp[i]
   }
## hyperpriors
beta0 ~ dnorm(0, 0.0001)
beta1 ~ dnorm(0, 0.0001)
beta2 ~ dnorm(0, 0.0001)
```

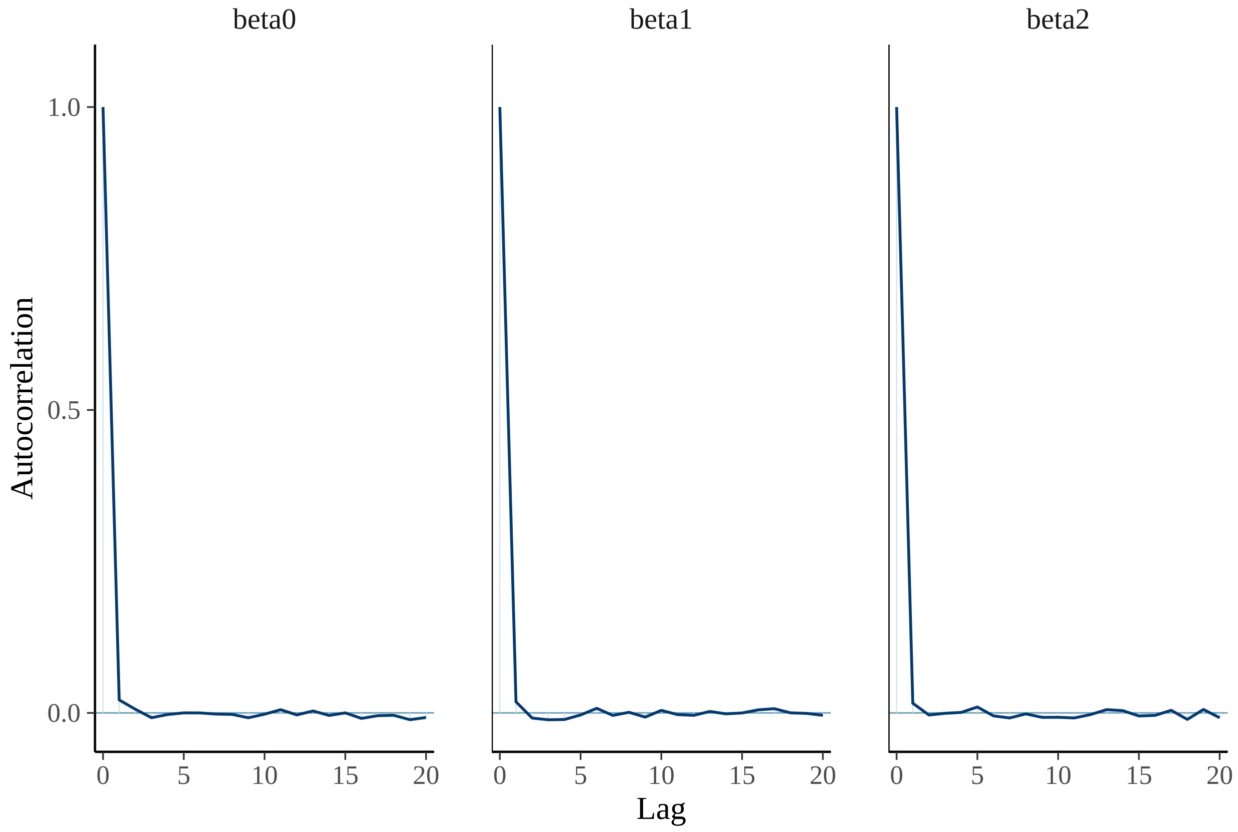
```
}
"
mean(Covid_Carleton$Rice_County)
## [1] 164.8205
sd(Covid_Carleton$Rice_County)
## [1] 161.4578
Covid_Carleton <- Covid_Carleton %>% mutate(Rice_Standard = as.vector(scale(Rice_County)),temp_standard

y <- Covid_Carleton$Positive_Results
n <- length(y)
N <- Covid_Carleton$Total_Tests
rice <- Covid_Carleton$Rice_Standard
temp <- Covid_Carleton$temp_standard

the_data <- list(y=y,n=n,N=N,rice=rice,temp=temp)

init = list(
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987654),
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987653),
            list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 987652)
            )

carleton_post <- run.jags(
  model = modelString,
  n.chains = 3,
  data = the_data,
  monitor = c("beta0","beta1","beta2","y_pred"),
  adapt = 1000,
  burnin = 2000,
  sample = 10000,
  thin=5,
  silent.jags = TRUE  # Eliminates progress bar
)
## Finished running the simulation
```

```
post_mcmc <- as.data.frame(as.mcmc(carleton_post))
quantile(post_mcmc$beta0,c(0.05,0.95))
##        5%       95%
## -6.198277 -5.811840
quantile(post_mcmc$beta1,c(0.05,0.95))
##        5%       95%
## 0.5987119 0.8271097
quantile(post_mcmc$beta2,c(0.05,0.95))
##         5%        95%
## -0.2287410  0.1995735
```

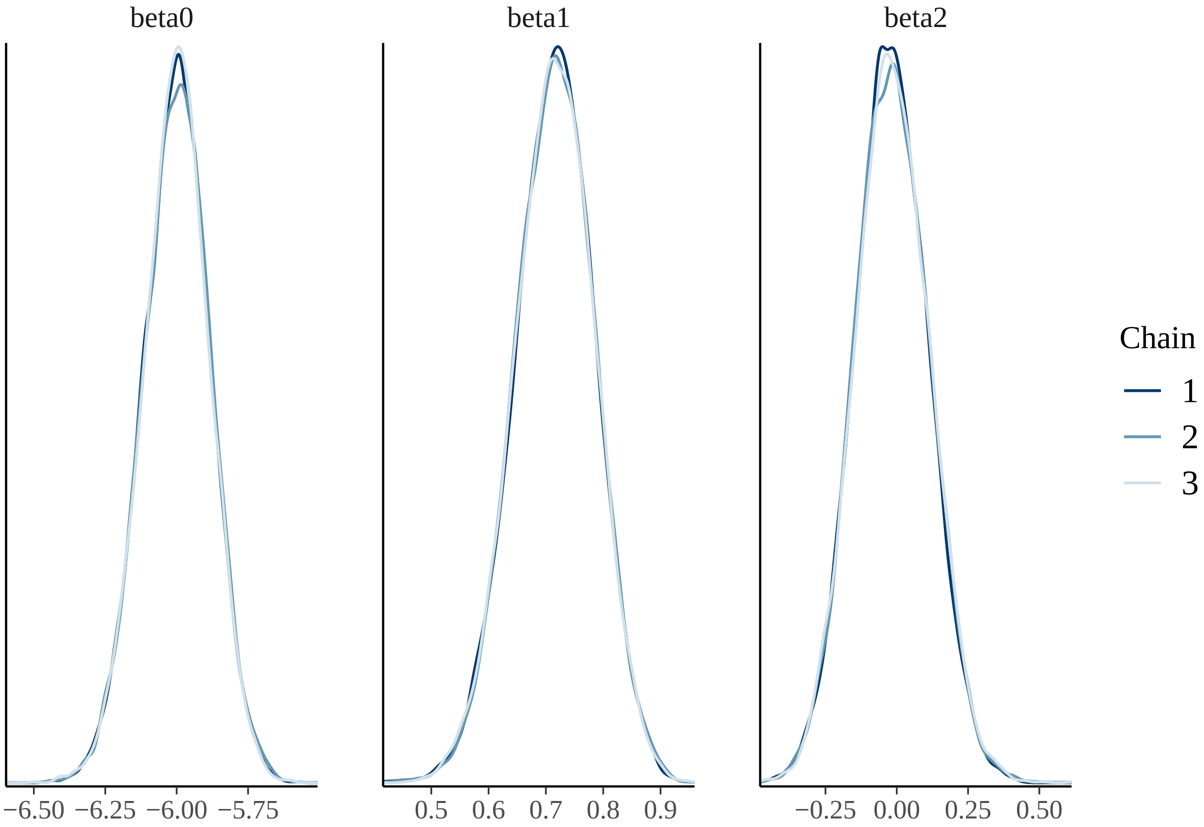
```
mcmc_trace(post_mcmc)
```



```
mcmc_acf(post_mcmc)
```



```
mcmc_dens_overlay(carleton_post$mcmc)
```

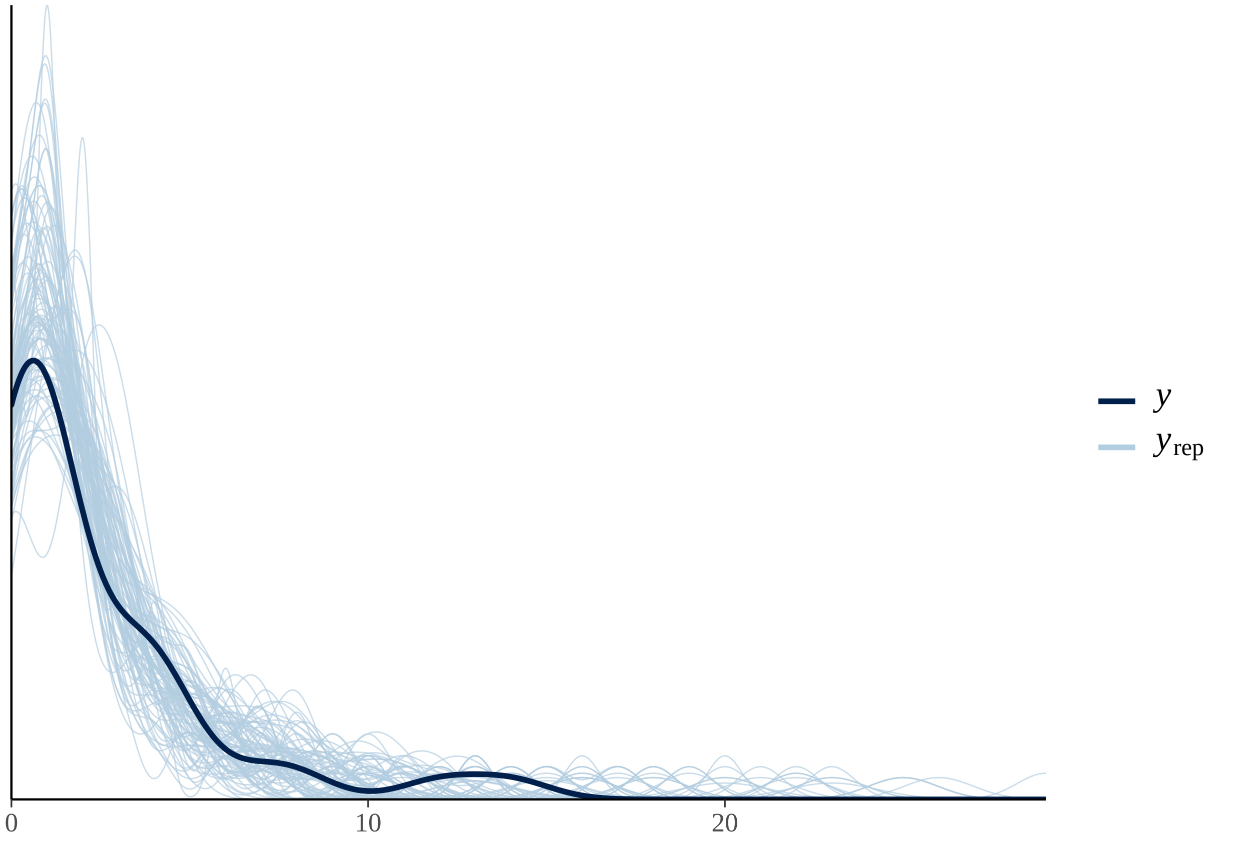


```
summary(post_mcmc)
##      beta0            beta1            beta2
##  Min.   :-6.597   Min.   :0.4160   Min.   :-0.47903
##  1st Qu.:-6.077   1st Qu.:0.6687   1st Qu.:-0.10450
##  Median :-5.998   Median :0.7165   Median :-0.01784
##  Mean   :-6.001   Mean   :0.7151   Mean   :-0.01614
##  3rd Qu.:-5.921   3rd Qu.:0.7629   3rd Qu.: 0.07155
##  Max.   :-5.507   Max.   :0.9594   Max.   : 0.61320
```

Posterior Predictive

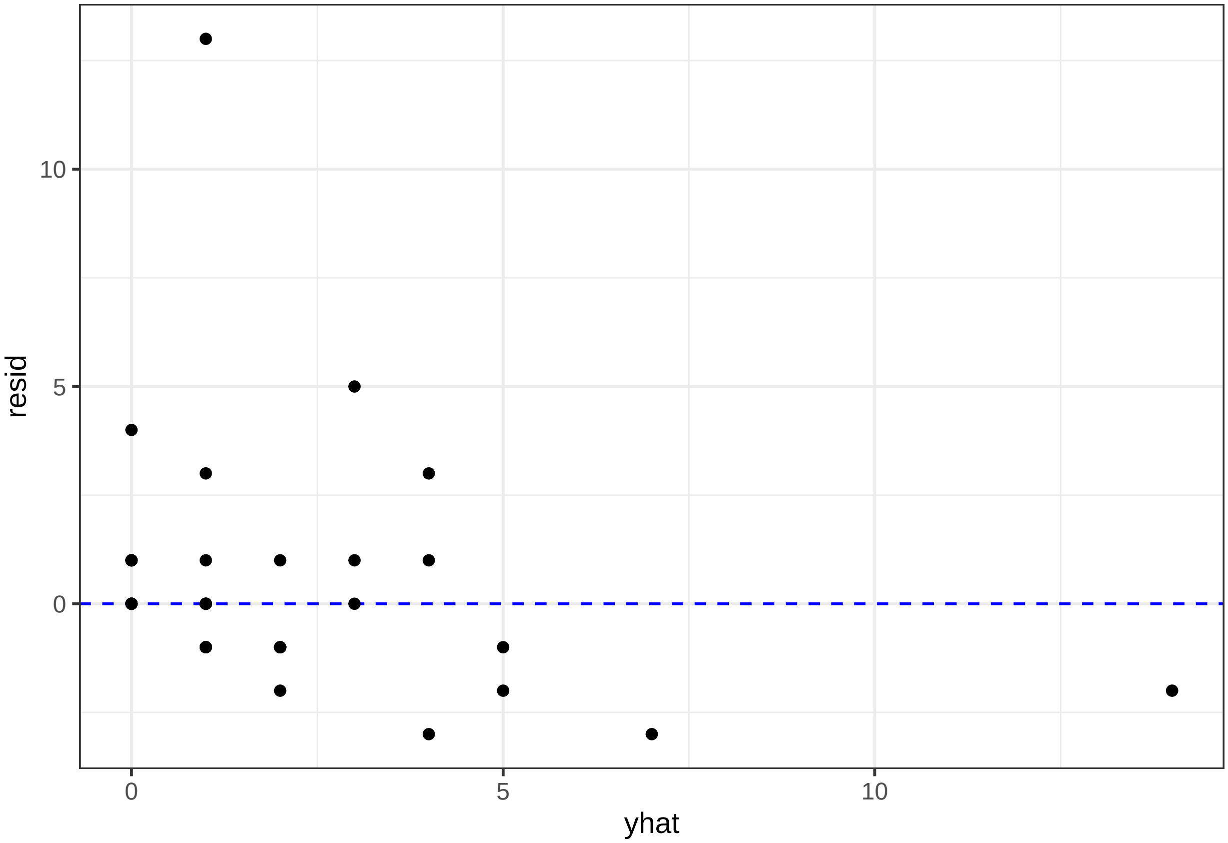
```
modelString <-"
model {
## likelihood
for (i in 1:n){
   y[i] ~ dbin(p[i], N[i])
}
## priors and regression
for (i in 1:n){
   logit(p[i]) <- beta0 +beta1*rice[i]+beta2*temp[i]
   }
## hyperpriors
beta0 ~ dnorm(0, 0.0001)
beta1 ~ dnorm(0, 0.0001)
beta2 ~ dnorm(0, 0.0001)
for (i in 1:n) {
   y_pred[i] ~ dbin(p[i], N[i])
}

}
"
carleton_pred <- run.jags(
  model = modelString,
  n.chains = 3,
  data = the_data,
  monitor = c("y_pred"),
  adapt = 1000,
  burnin = 2000,
  sample = 10000,
  thin=5,
  silent.jags = TRUE  # Eliminates progress bar
)
## Finished running the simulation
```

```
ppc_dens_overlay(y = Covid_Carleton$Positive_Results, yrep = carleton_pred$mcmc[[1]][1:100,])
```

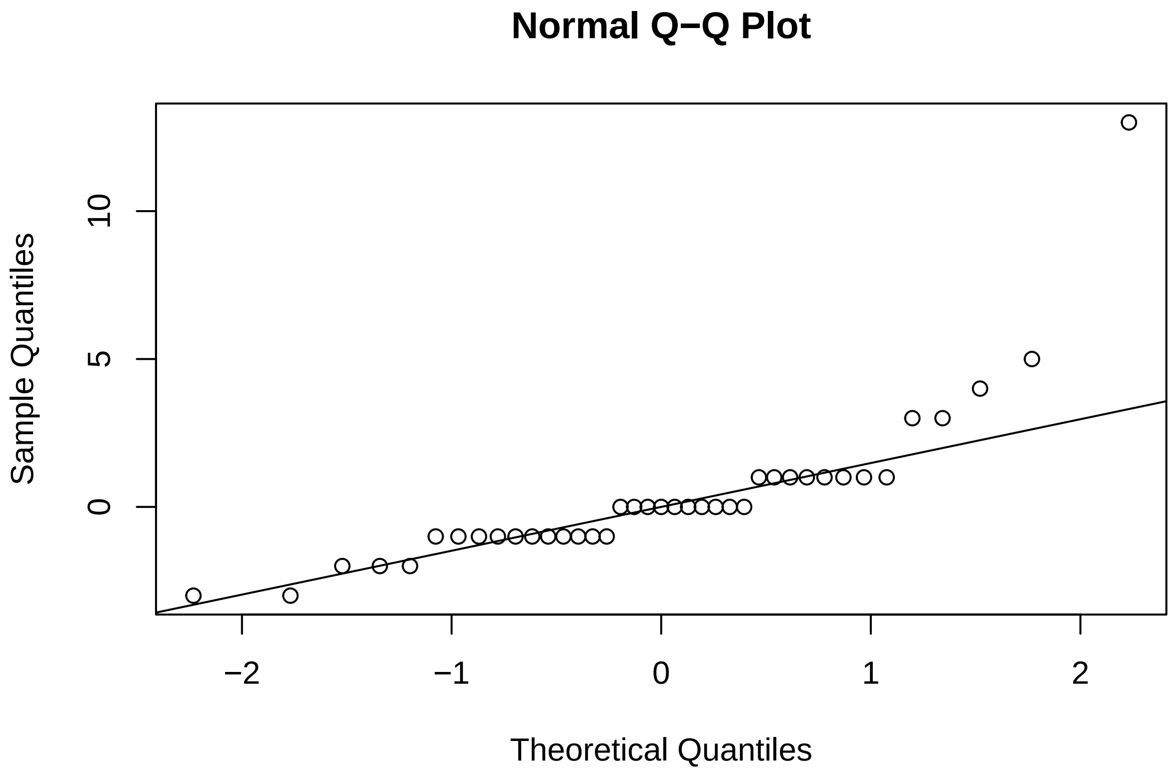


Residual Diagnostics

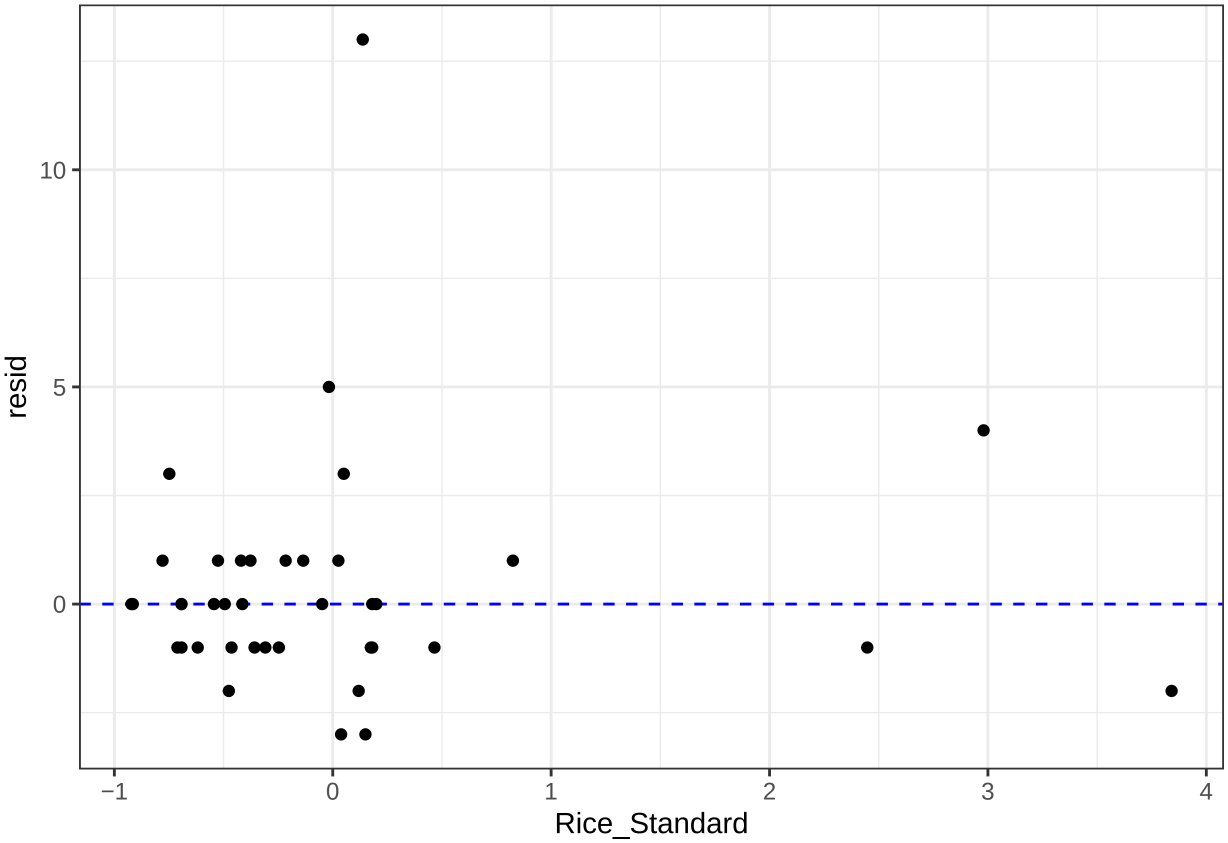
```
resids <- Covid_Carleton %>%
mutate(
beta0 = mean(post_mcmc$beta0),
beta1 = mean(post_mcmc$beta1),
beta2 = mean(post_mcmc$beta2),
phat_logit = beta0 + beta1 * Rice_Standard + beta2 * temp_standard,
phat = exp(phat_logit)/(1+exp(phat_logit)),
yhat = rbinom(39,Total_Tests, phat),
resid = Positive_Results - yhat
)

ggplot(data = resids, aes(x = yhat, y = resid)) + # fitted values
  geom_hline(yintercept = 0, linetype = 2, color = "blue") +
  geom_point() + theme_bw()
```



```
qqnorm(resids$resid)
qqline(resids$resid)
```



```
ggplot(data = resids, aes(x = Rice_Standard, y = resid)) + # fitted values
  geom_hline(yintercept = 0, linetype = 2, color = "blue") +
  geom_point() + theme_bw()
```



```
quantile(post_mcmc$beta2,c(0.05,0.95))
##         5%        95%
## -0.2287410  0.1995735
```