730 Group Project

Rebekah Kristal with collaborators Amani Chehimi & Shane Fitzgerald

2024-11-30

Data reformatting

```
newdata <- read_csv("FreqCategories.csv") %>%  mutate(Weight = Freq / sum(Freq))
```

```
## New names:
## Rows: 5462 Columns: 9
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (2): AgeCat, EduCat dbl (7): ...1, y, REGION, SEX, RACENEW, POORYN, Freq
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
newdata<-mutate(newdata, weight.var=1/Freq) %>% mutate(REGION=as.factor(REGION)) %>% mutate(AgeCat=as.fa

#converting y's into factor variable, changing range from 0-8 to 1-9 to match with model output
newdata1<-mutate(newdata, y=y+1) %>% mutate(y, factor(y, ordered=TRUE))
```

Model fitting

Amani's model: weighted linear regression with hierarchical variable

```
modALoo <- brm(
  y | weights(Weight) ~ (1 | REGION + AgeCat + SEX + RACENEW + EduCat + POORYN),
  data = newdata1,
  family = gaussian(),
  iter = 1000,
  chains = 4,
  cores = getOption("mc.cores", 4),
  seed = 12345
)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

Shane's model: weighted ordinal regression

```
modSLoo <- brm(
  y|weights(Weight)~REGION + AgeCat + SEX + RACENEW + EduCat + POORYN,
  data = newdata1,
  family=cumulative(link="logit"),
  iter = 1000,
  chains = 4,
  cores = getOption("mc.cores", 4),
  seed = 12345
)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

Rebekah's model 1: weighted ordinal regression with hierarchical variable

```
modRLoo1 <- brm(
  y | weights(Weight) ~ (1 | REGION + AgeCat + SEX + RACENEW + EduCat + POORYN),
  data = newdata1,
  family=cumulative(link="logit"),
  iter = 1000,
  chains = 4,
  cores = getOption("mc.cores", 4),
  seed = 12345
)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

Rebekah's model 2: weighted ordinal regression with interactions

```
modRLoo2 <- brm(
  y|weights(Weight)~REGION + AgeCat + SEX + RACENEW + EduCat + POORYN + REGION*POORYN + REGION*RACENEW +
  data = newdata1,
  family=cumulative(link="logit"),
  iter = 1000,
  chains = 4,
  cores = getOption("mc.cores", 4),
  seed = 12345
)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
## Warning: There were 752 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess
```

Our own PPC plots of expected vs observed counts

get observed counts from data

```
observed_counts <- select(newdata1, c(y, Freq))
total_freq<-group_by(observed_counts, y) %>% summarise(total=sum(Freq))
observed_props<-mutate(total_freq, observed=total/sum(total)) %>% mutate(y=as.factor(y))


get_sum_stat<-function(y, row){(sum(y==5))/nrow(row)}

tobs<-observed_props[5,3]
```

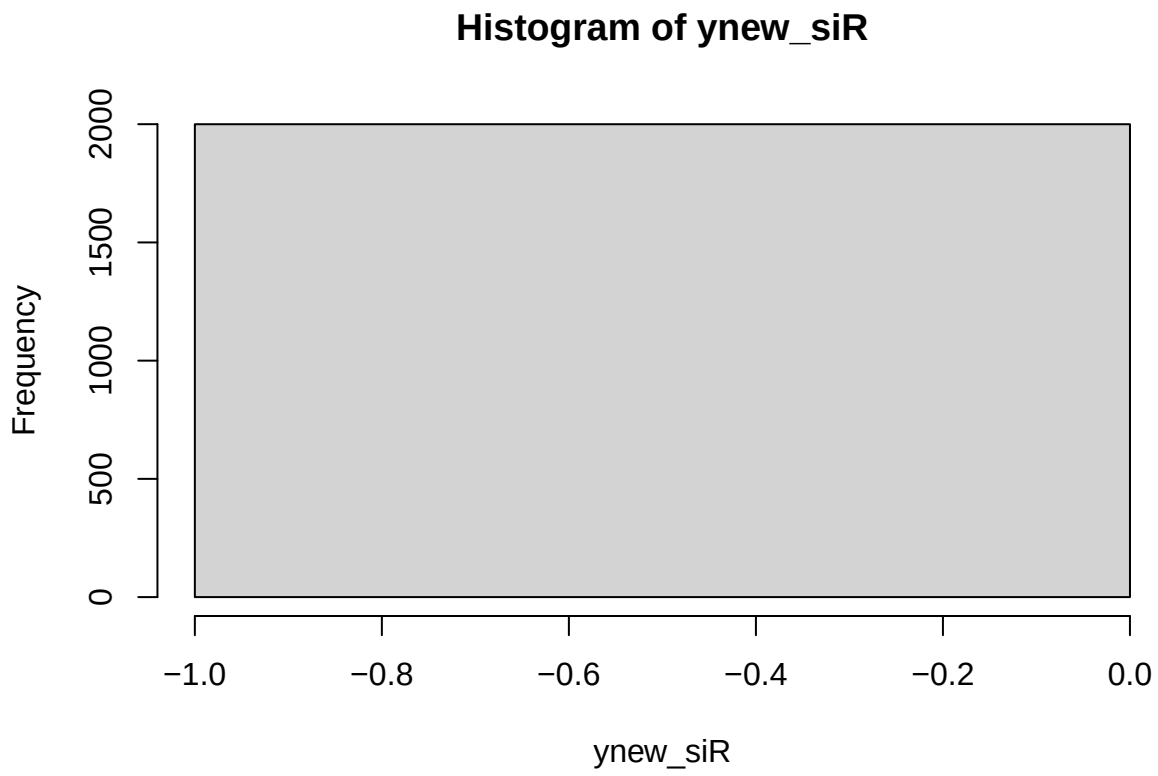
function to make ppc plot for each model of proportion of people in each response category

```
make_ppc_plot <- function(model_name){
    predicted_catsR<-as.data.frame(posterior_predict(model_name))
    ynew_siR<-apply(predicted_catsR, 1, get_sum_stat, newdata)
    #ppc for proportion of observations in category 5
    hist(ynew_siR)
    abline(v = tobs)

    #ppc for all categories
    #formatting for ggplot
    posterior_preds_longR <- predicted_catsR %>%
      pivot_longer(cols = everything(), names_to = "chain", values_to = "predicted_category")

    posterior_preds_longR$predicted_category <- as.factor(posterior_preds_longR$predicted_category)

    category_countsR <- table(posterior_preds_longR$predicted_category)
    category_counts_dfR <- as.data.frame(category_countsR)
    colnames(category_counts_dfR) <- c("y", "Count")
    category_counts_propR<-mutate(category_counts_dfR, predicted=Count/(4000*5462))

    combinedR<-left_join(observed_props, category_counts_propR, by="y")
    combined1R<-pivot_longer(combinedR, c(3,5), names_to = "Freq")

    #plot of proportion of each category for observed and predicted data
```

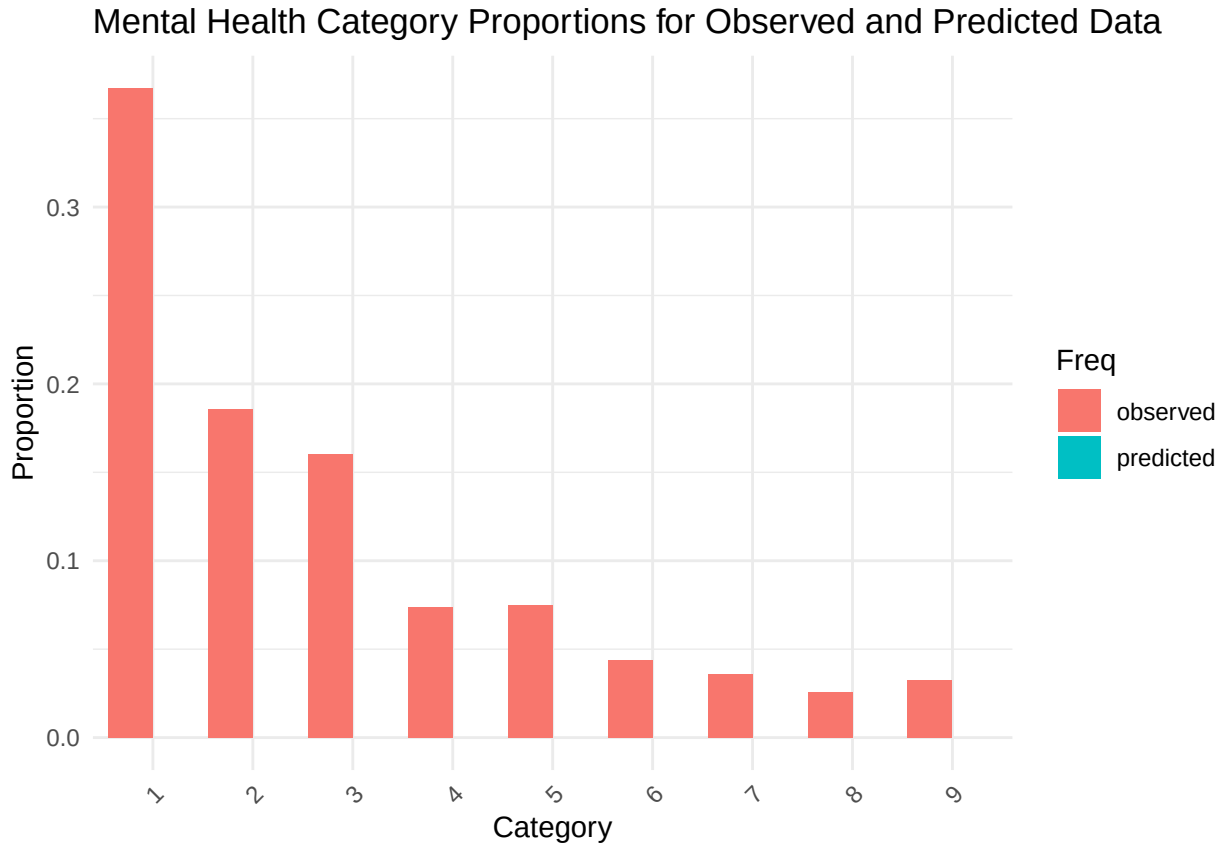
```
    ggplot(combined1R, mapping=aes(x=y, y=value, fill=Freq))+
      geom_bar(stat="identity", position="dodge")+
      labs(title = "Mental Health Category Proportions for Observed and Predicted Data",
           x = "Category",
           y = "Proportion") +
      theme_minimal() +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))
}
```

Make proportion PPC plots

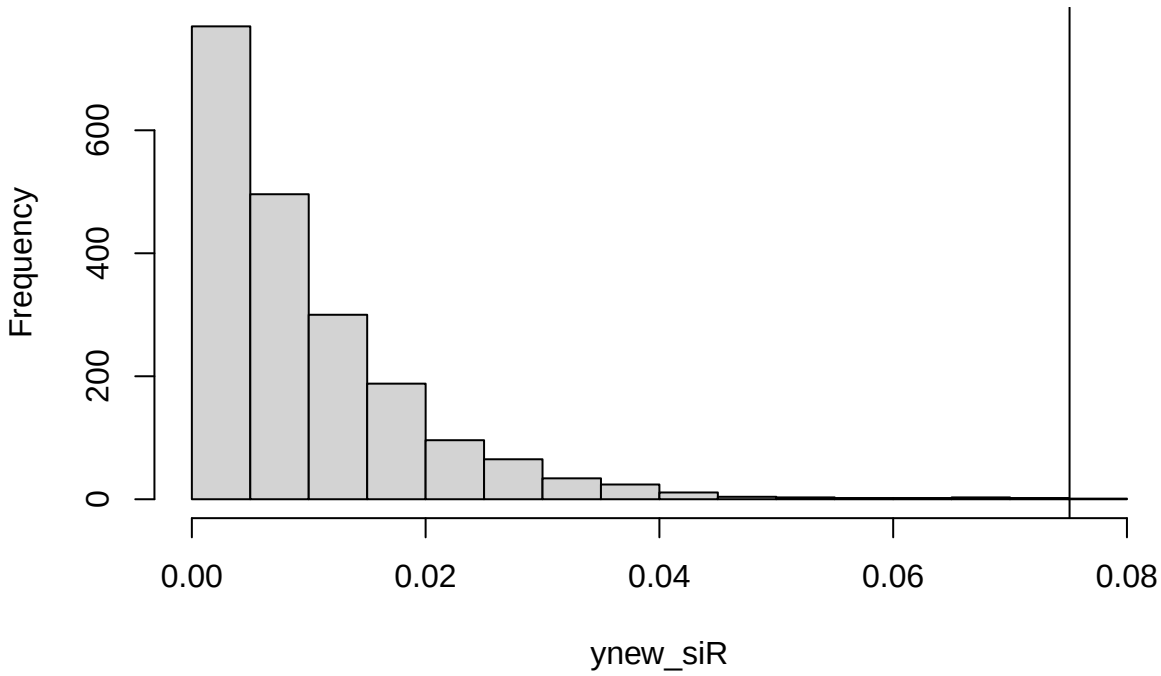
```
# Amani's model
make_ppc_plot(modALoo)
```

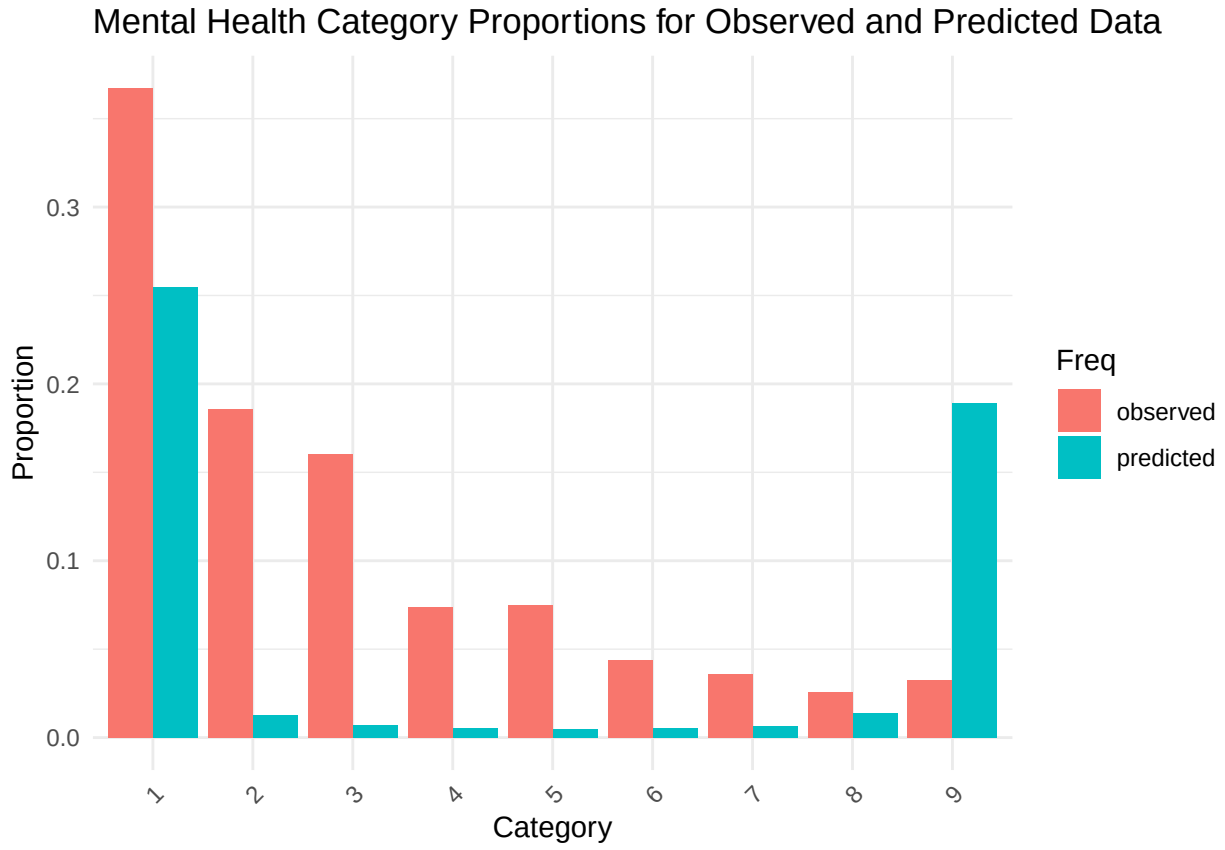



```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```

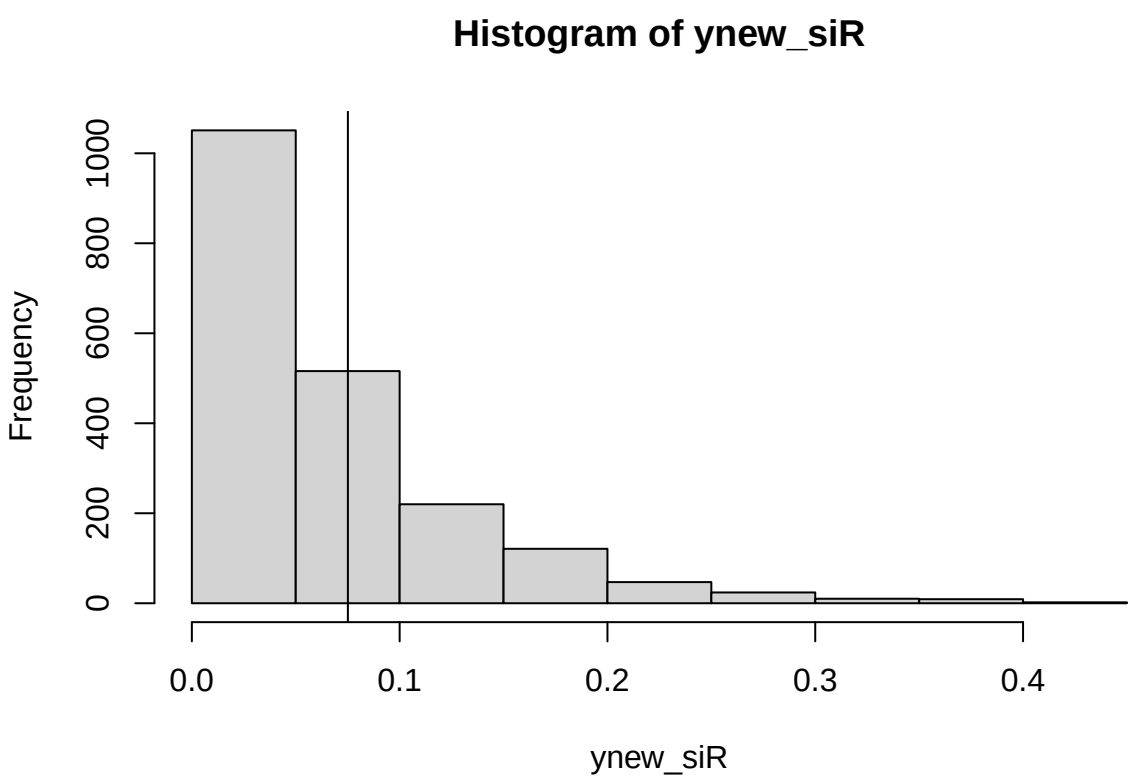


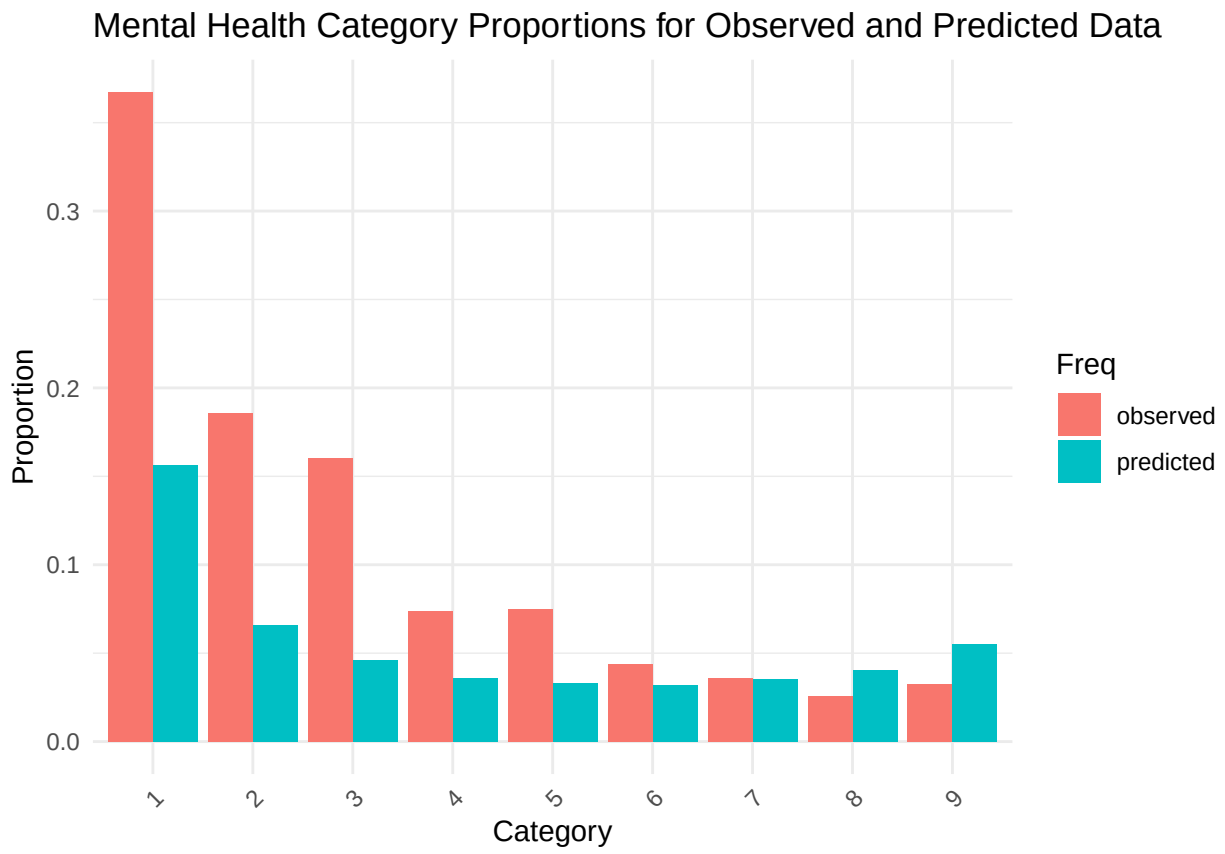

```
# Shane's model
make_ppc_plot(modSLoo)
```


Histogram of ynew_siR
Frequency
0
200
400
600
0.00
0.02
0.04
0.06
0.08
ynew_siR

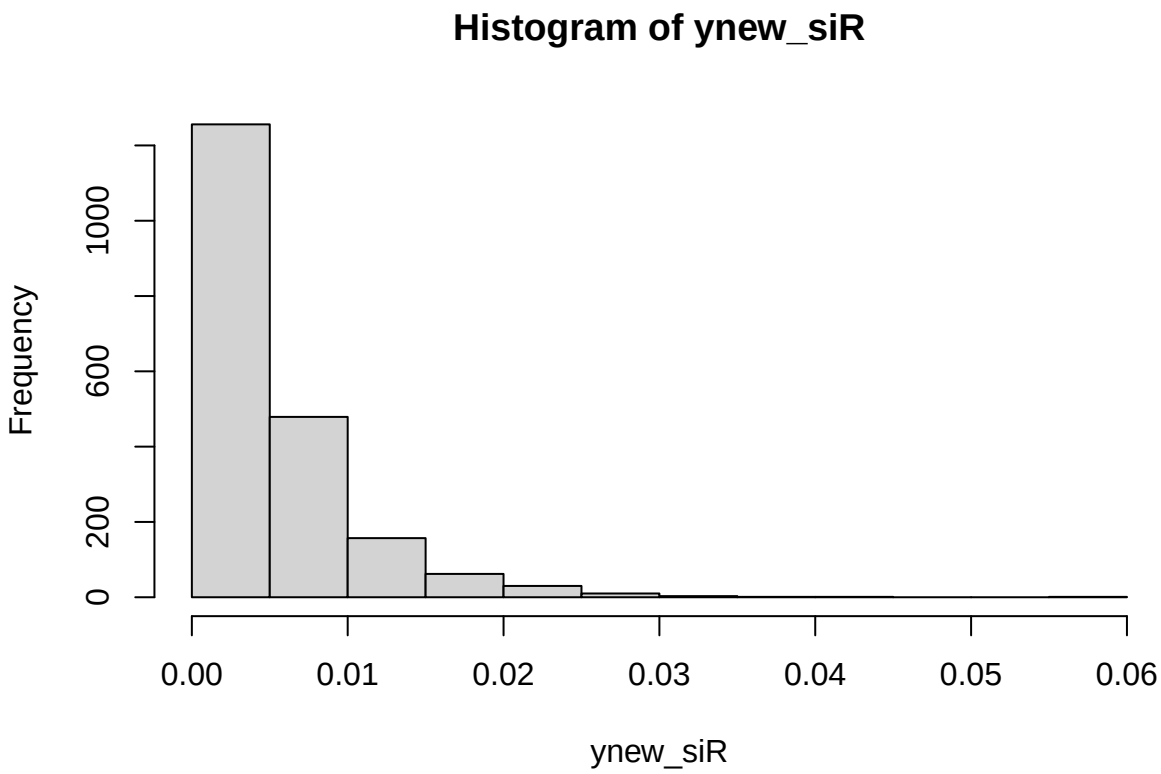


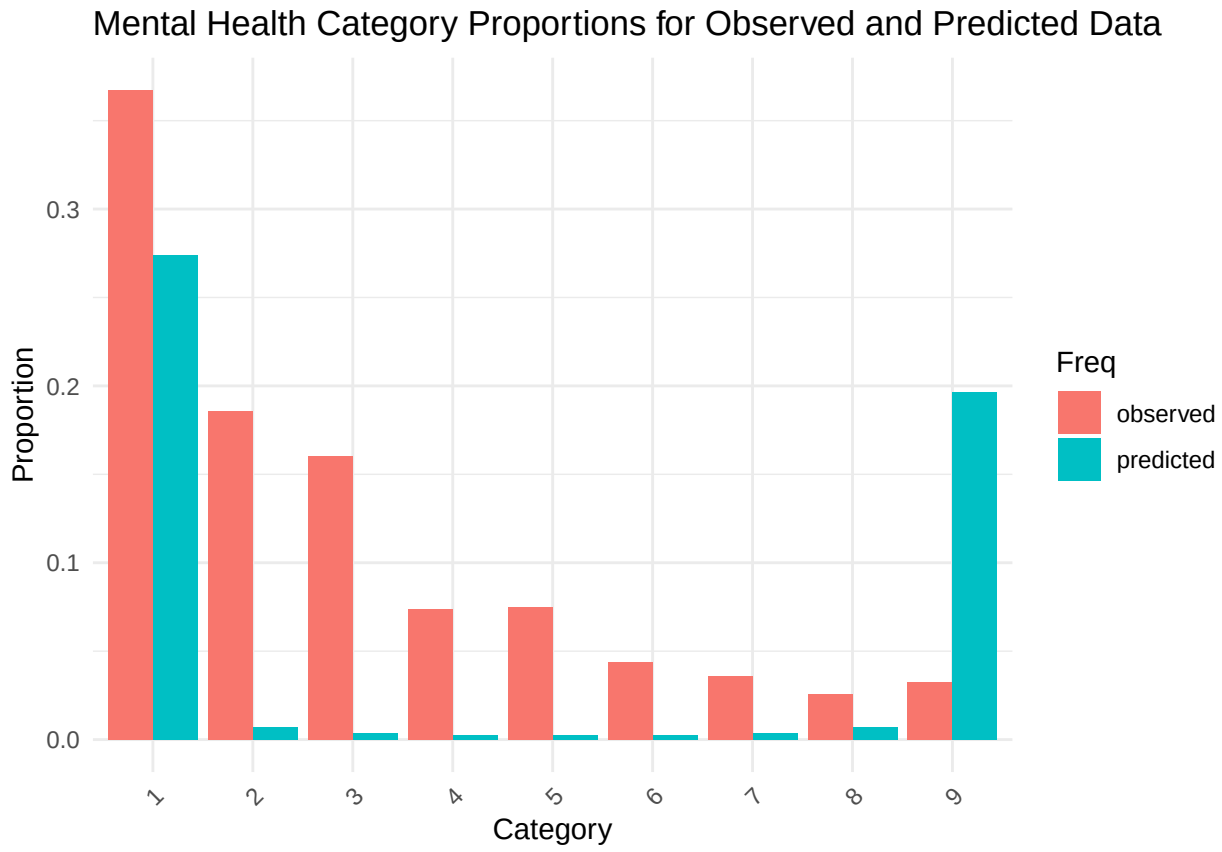

```
# Rebekah's model 1
make_ppc_plot(modRLoo1)
```


Histogram of ynew_siR
Frequency
0
200
400
600
800
1000
0.0
0.1
0.2
0.3
0.4
ynew_siR




```
# Rebekah's model 2
make_ppc_plot(modRLoo2)
```


Histogram of ynew_siR
Frequency
0
200
600
1000
0.00
0.01
0.02
0.03
0.04
0.05
0.06
ynew_siR


Mental Health Category Proportions for Observed and Predicted Data
Proportion
0.0
0.1
0.2
0.3
1
2
3
4
5
6
7
8
9
Category
Freq
observed
predicted