HW5_ModelSelection&linearregression

N. Barrus

2024-02-21

Purpose:

The purpose of this markdown document is to work through Homework 5 in Dr. Babcock's Bayesian Statistics Course at the University of Miami. Homework 5 deals with model selection and linear regression.

General Start to Code

```
rm(list = ls())

######github#####
#note, only needed after 90 days from 1/16/2024

#  usethis::create_github_token()
#  gitcreds::gitcreds_set()

#####check for r updates#####
#note, updateing may take some time so plan accordingly

#require(installr)

#check.for.updates.R()

#updateR() #only if needed

#######check for package updates#####
#note, updateing may take some time so plan accordingly

#old.packages()

# update.packages() #make the decision to the update the packages
```

Load packages

```
library(tidyverse)
library(R2jags)
library(rstan)
library(ggmcmc)
```

```
library(purrr)
library(magrittr)
library(here)
library(loo)
theme_set(theme_bw(base_size=15))
```

Data

For model selection, the data consist of counts of the number of trees in 30 equal sized quadrats.

And for the linear regression, the data consist of DEET repellent levels and the number of mosquito bits suffered by volunteers.

```
Y <- c(11, 3, 7, 6, 2, 36, 14, 9, 2, 10, 2, 7, 3, 1, 0, 0, 0, 1,
 5, 0, 2, 11, 5, 3, 0, 3, 3, 27, 0, 11)

tree.counts <- list(Y = Y,
                    N = length(Y))

tree.counts
```

```
## $Y
##  [1] 11  3  7  6  2 36 14  9  2 10  2  7  3  1  0  0  0  1  5  0  2 11  5  3  0  3  3 27  0 11
##
## $N
## [1] 30
```

```
deet.data <- read_csv(here("data","deet.csv")) |>
  mutate(z.dose = (dose - mean(dose))/sd(dose))
```

```
## Rows: 52 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): dose, bites
##
## 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.
```

```
knitr::kable(head(deet.data), caption = "Data Preview")
```

Table 1: Data Preview

dose	bites	z.dose
1.5	4.062019	-1.8480349
1.4	4.183300	-1.9407040
2.0	3.535534	-1.3846896
2.3	3.535534	-1.1066824
2.3	3.240370	-1.1066824
2.5	3.240370	-0.9213443

Problem 1: Model Selection

In this problem we will rerun the negative binomial (either parameterization) and Poisson models from homework 4, now adding calculations of the DIC, WAIC and LOOIC for model selection

A) Calculate the DIC for both the Poisson and negative binomial in JAGS, and make a table with the DIC, pD, deltaDIC and DIC weights. Which model is preferred? *run the JAGS models*

```
treemod.HW4.Q1.poiss <- jags(data = tree.counts,
  parameters.to.save = c("lamda"),
  n.chains = 2,
  n.burnin = 1000,
  n.iter = 20000,
  model.file = here("JAGS_mods","HW3-Q1-Poisson.txt")
)

treemod.HW4.Q3.negbin <- jags(data = tree.counts,
  parameters.to.save = c("p","r","m","v"),
  n.chains = 2,
  n.burnin = 1000,
  n.iter = 20000,
  model.file = here("JAGS_mods","HW3-Q3-NegBinom.txt"))
```

create the model selection table

```
DIC.table <- tibble(model = c("Poisson", "Neg-Binomial"),
                    model.ls = c(list(treemod.HW4.Q1.poiss), list(treemod.HW4.Q3.negbin))) |>
  mutate(DIC = map_dbl(model.ls, c(2,24)),
         pD = map_dbl(model.ls, c(2,23)),
         deltaDIC = DIC - min(DIC),
         weight = round(exp(-2*deltaDIC)/sum(exp(-2*deltaDIC)),digits = 5))

DIC.table |> select(-model.ls) |> knitr::kable(caption = "DIC Table")
```

Table 2: DIC Table

model	DIC	pD	deltaDIC	weight
Poisson	324.1352	0.9840344	147.7144	0
Neg-Binomial	176.4207	1.9872850	0.0000	1

The Negative binomial model is preferred.

B) Do the same with the WAIC in STAN. Which model is preferred? *run the STAN mods*

```
treemod.poiss <- stan(file = here("STAN_docs","HW5_treecounts_poisson_modsel.stan"),
                      data = tree.counts)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## 'C:\Users\nbarr\OneDrive\Pictures\Documents\FIU\Classes\BayesianStatistc_Course\BayesianStatistic_R\S
```

```
treemod.negbin <- stan(file = here("STAN_docs","HW5_treecounts_negbin_modsel.stan"),
                       data = tree.counts)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## 'C:\Users\nbarr\OneDrive\Pictures\Documents\FIU\Classes\BayesianStatistc_Course\BayesianStatistic_R\S
```

create WAIC table

```
WAIC.table <- tibble(model = c("Poisson", "Neg-Binomial"),
                     model.ls = c(treemod.poiss, treemod.negbin)) |>
  mutate(LL.ls = map(model.ls,extract_log_lik, parameter_name = "LL"),
         WAIC.ls = map(LL.ls, waic),
         elpd_waic = map_dbl(WAIC.ls,c(1,1)),
         p_waic = map_dbl(WAIC.ls,c(1,2)),
         waic = map_dbl(WAIC.ls,c(1,3)),
         deltaWAIC = waic - min(waic),
         weight = round(exp(-2*deltaWAIC)/sum(exp(-2*deltaWAIC)),digits = 5))
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `WAIC.ls = map(LL.ls, waic)`.
## Caused by warning:
## !
## 2 (6.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
WAIC.table |> select(-model.ls,-LL.ls,-WAIC.ls) |> knitr::kable(caption = "WAIC Table")
```

Table 3: WAIC Table

model	elpd_waic	p_waic	waic	deltaWAIC	weight
Poisson	-166.8121	9.932478	333.6242	156.552	0
Neg-Binomial	-88.5361	2.379678	177.0722	0.000	1

C) Do the same with the LOOIC in STAN. Which model is preferred?

create LOOIC table

```
LOOIC.table <- tibble(model = c("Poisson", "Neg-Binomial"),
                     model.ls = c(treemod.poiss, treemod.negbin)) |>
  mutate(LL.ls = map(model.ls,extract_log_lik, parameter_name = "LL"),
         looic.ls = map(LL.ls, loo),
         elpd_looic = map_dbl(looic.ls,c(1,1)),
         p_looic = map_dbl(looic.ls,c(1,2)),
         looic = map_dbl(looic.ls,c(1,3)),
         deltaLooic = looic - min(looic),
         weight = round(exp(-2*deltaLooic)/sum(exp(-2*deltaLooic)),digits = 5))
```

```
## Warning: There were 4 warnings in `mutate()`.
## The first warning was:
## i In argument: `looic.ls = map(LL.ls, loo)`.
## Caused by warning:
## ! Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
## i Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.
```

```
LOOIC.table |> select(-model.ls,-LL.ls,-looic.ls) |> knitr::kable(caption = "LOOIC Table")
```

Table 4: LOOIC Table

model	elpd_looic	p_looic	looic	deltaLooic	weight
Poisson	-166.86156	9.981919	333.7231	156.506	0
Neg-Binomial	-88.60854	2.452113	177.2171	0.000	1

D) Compare the three information criteria in part a, b and c. Do they all give the same results? Is this result consistent with what you learn by looking at the mean and variance that you estimated when you ran the negative binomial model? The three methods of model selection all selected the negative binomial model. This is consistent with the mean and variance that we estimated in the negative binomial model. The mean and the variance were not equal so the Poisson model should not be preferred.

Problem 2: Linear Regression

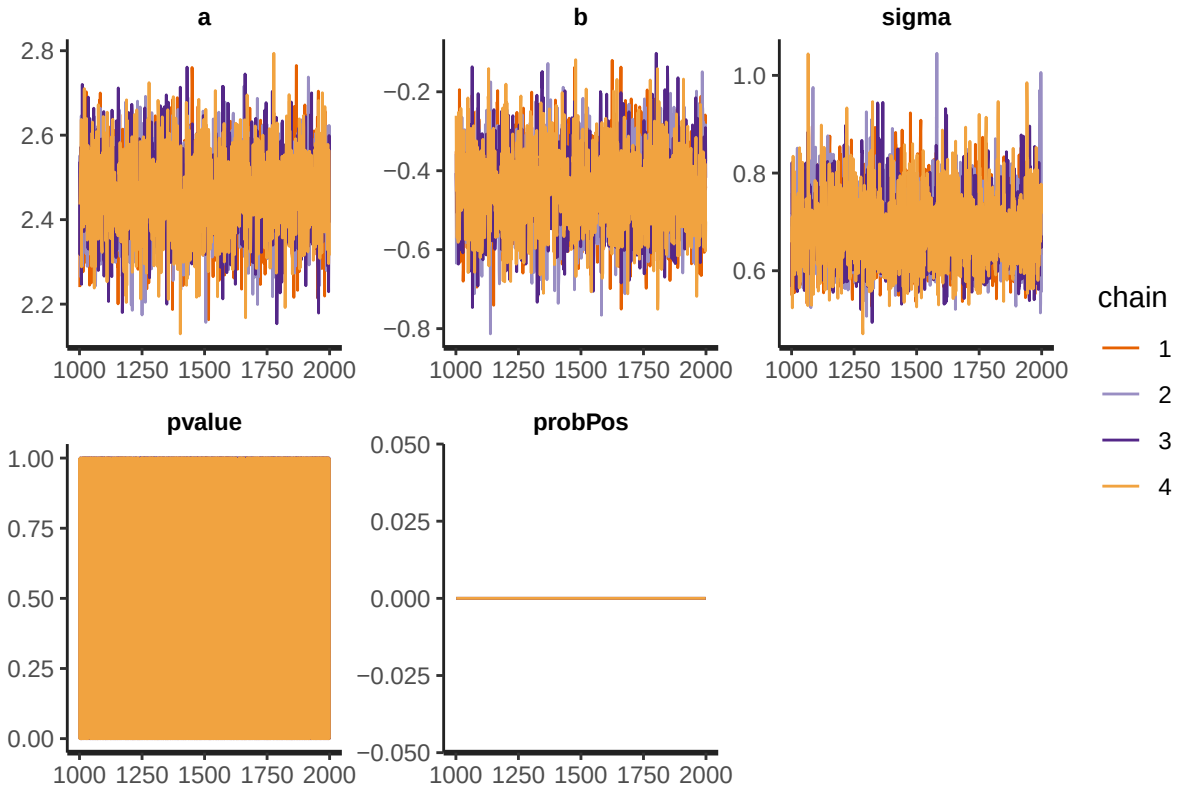
The data in the file called deet.csv is from a study of the effect of DEET insect repellent on the number of mosquito bites suffered by volunteers. The x variable is dose of DEET, and the y variable is the number of bites, square root transformed for normality. This example is from https://whitlockschluter3e.zoology.ubc.ca/chapter17.html

A) Use STAN to run a linear regression to predict bites from dose, using a normal prior for the intercept (a) and slope (b), and an exponential prior for the residual variance. Standardize the X variable by subtracting the mean and dividing by the standard deviation. Give the summary statistics for a, b and the residual variance, along with the probability that the slope is positive. *fit the regression model*

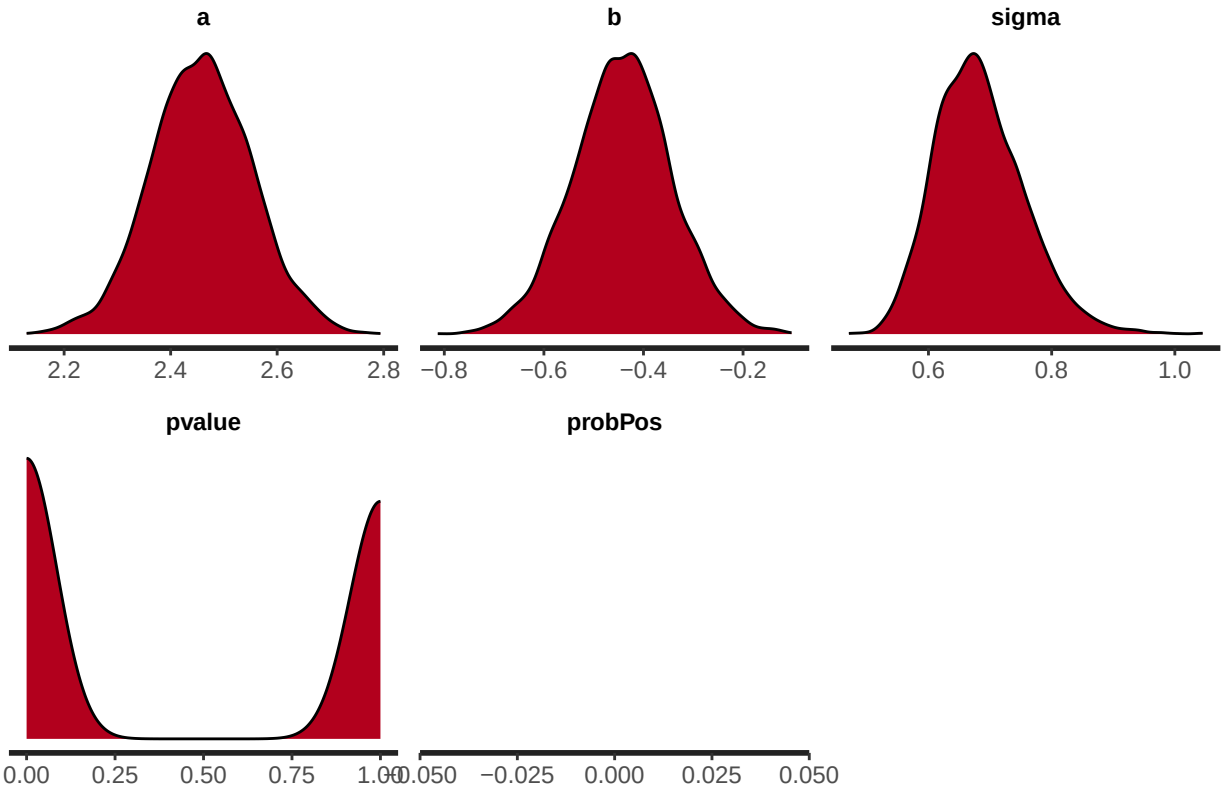
```
stan.deet.data <- list(x = deet.data$z.dose,
                       y = deet.data$bites,
                       N = length(deet.data$z.dose),
                       volDose = (2.8 - mean(deet.data$dose))/sd(deet.data$dose))

fit.stan.reg <- stan(file = here("STAN_docs","HW5_linearregression.stan"),
                     data = stan.deet.data)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## 'C:\Users\nbarr\OneDrive\Pictures\Documents\FIU\Classes\BayesianStatistc_Course\BayesianStatistic_R\S
```

```
stan_trace(fit.stan.reg,pars=c("a","b","sigma","pvalue","probPos"))
```




```
stan_dens(fit.stan.reg,pars=c("a","b","sigma","pvalue","probPos"))
```



summary stats of model

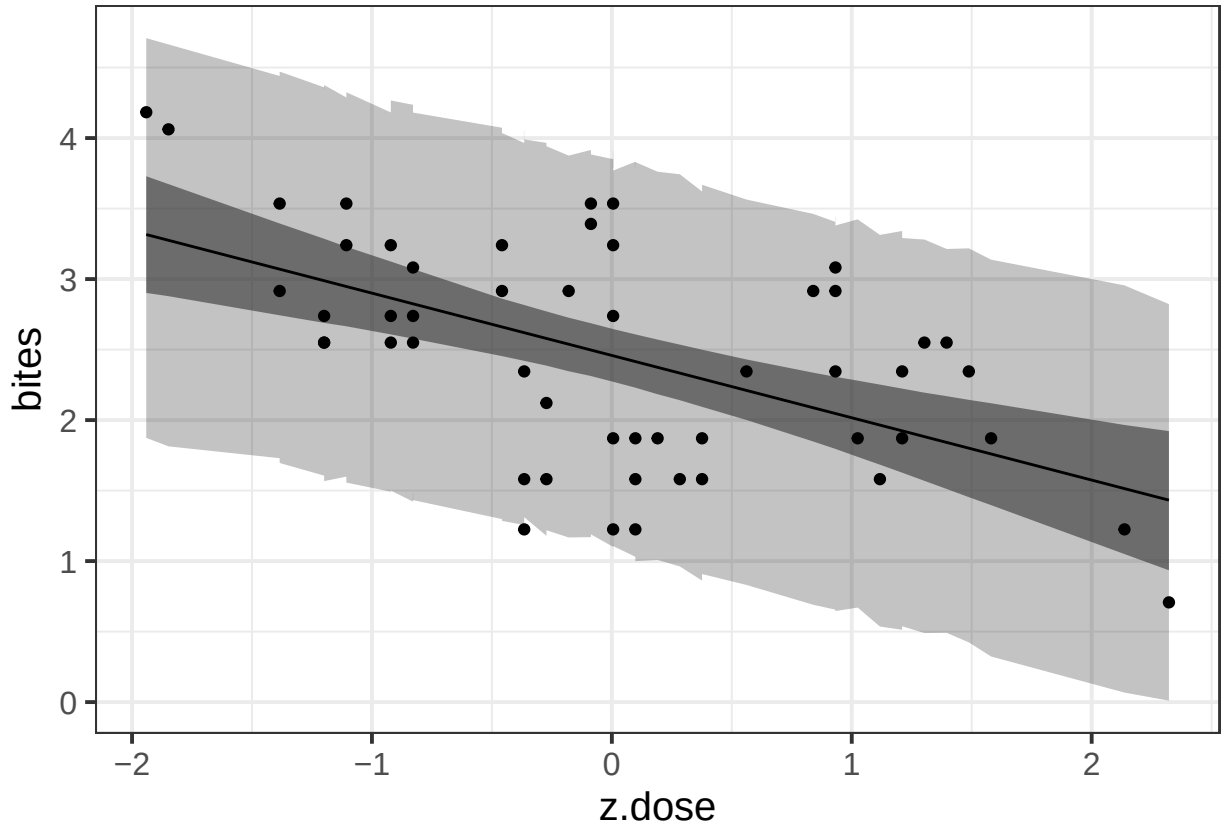
```
print(fit.stan.reg,pars=c("a","b","sigma","probPos"))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## a        2.46       0 0.09  2.27  2.39  2.46  2.52  2.65  3586    1
## b       -0.44       0 0.10 -0.64 -0.51 -0.44 -0.38 -0.24  3533    1
## sigma    0.68       0 0.07  0.56  0.63  0.68  0.73  0.84  3179    1
## probPos  0.00     NaN 0.00  0.00  0.00  0.00  0.00  0.00   NaN  NaN
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 22 22:31:24 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

B) Plot the data and the model fit, with the credible interval of the line. In STAN, you can make a new generated quantity for the mean prediction, say ymean, and then, in R, extract its summary statistics with summary(stanobjectname,par=”ymean”) *plot of the data and fit*

```
line.posterior <- summary(fit.stan.reg, par = "ymean")
predicted.posterior <- summary(fit.stan.reg, par = "ypred")
```

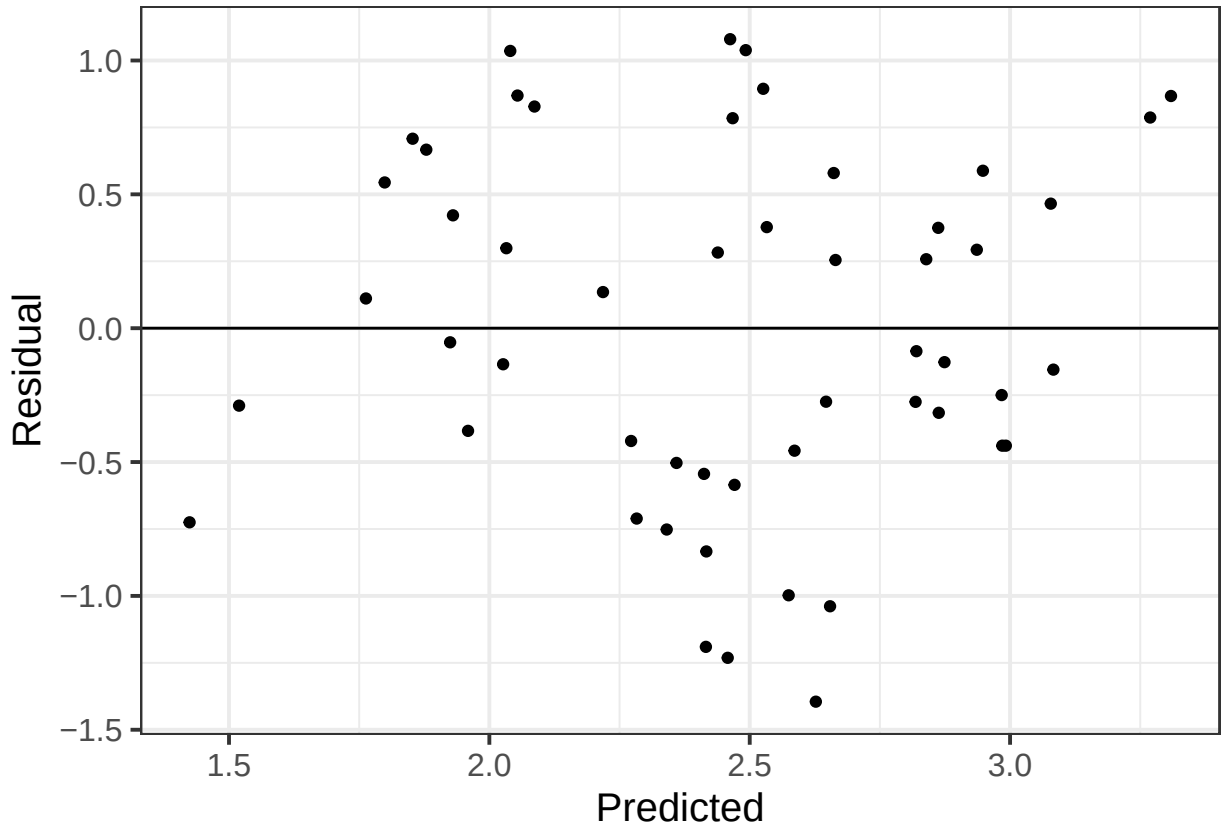
```
deet.data.fit<- deet.data |>
  mutate(yline = line.posterior$summary[1:(stan.deet.data$N)],
         yline.low = line.posterior$summary[(stan.deet.data$N*3+1):(stan.deet.data$N*4)],
         yline.upp = line.posterior$summary[(stan.deet.data$N*7+1):(stan.deet.data$N*8)],
         ypred.low = predicted.posterior$summary[(stan.deet.data$N*3+1):(stan.deet.data$N*4)],
         ypred.upp = predicted.posterior$summary[(stan.deet.data$N*7+1):(stan.deet.data$N*8)],
         Residual = get_posterior_mean(fit.stan.reg,pars="resid")[,"mean-all chains"],
         Predicted = get_posterior_mean(fit.stan.reg,pars="ypred")[,"mean-all chains"])


deet.data.fit |>
  ggplot(aes(y = bites,x = z.dose))+
  geom_ribbon(aes(x = z.dose, ymax = ypred.upp, ymin = ypred.low), alpha = 0.3)+
  geom_ribbon(aes(x = z.dose, ymax = yline.upp, ymin = yline.low), alpha = 0.6)+
  geom_line(aes(x = z.dose, y = yline))+
  geom_point()
```



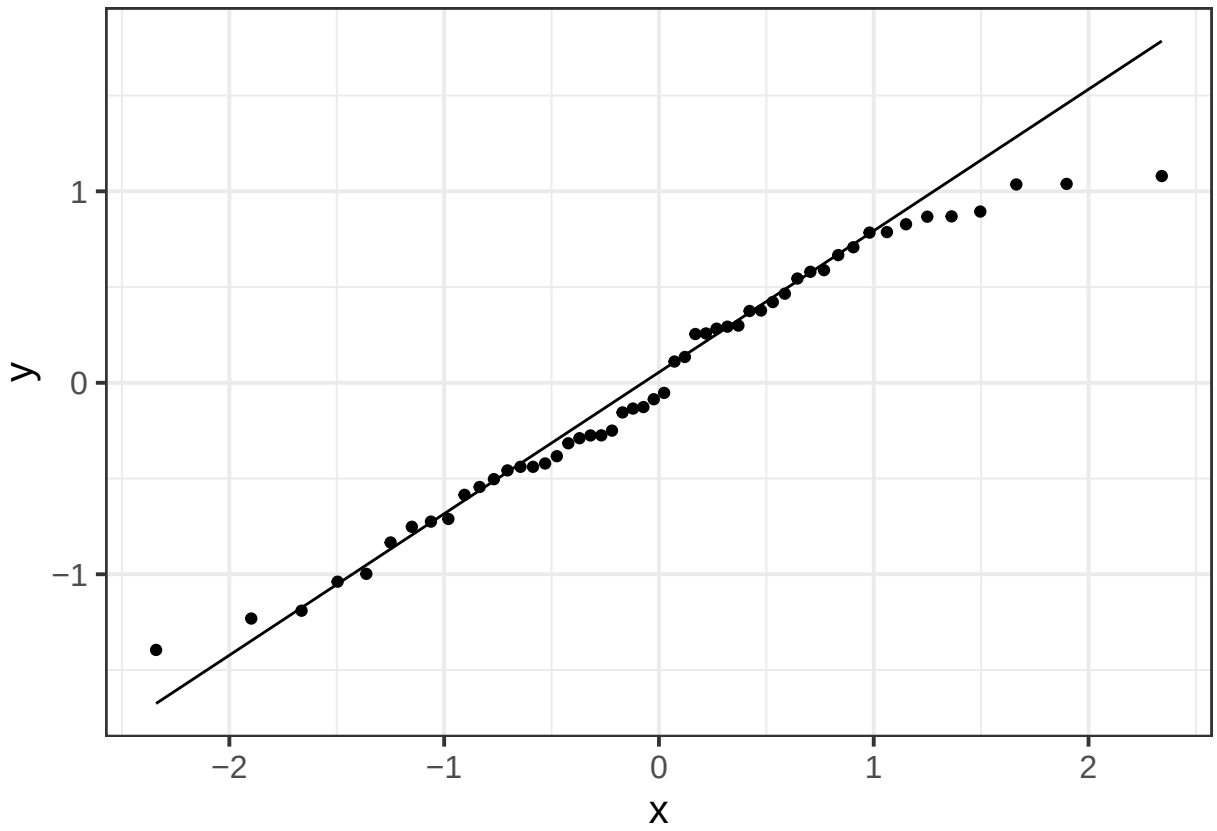
C) Plot the residuals against the predicted values, and the qq normal plot of the residuals. Are the assumptions of the model met? note) I extracted the predicted and residuals in the code from part B

residual plots

```
deet.data.fit |>
  ggplot(aes(x=Predicted,y=Residual))+
  geom_point()+
  geom_hline(yintercept=0)
```



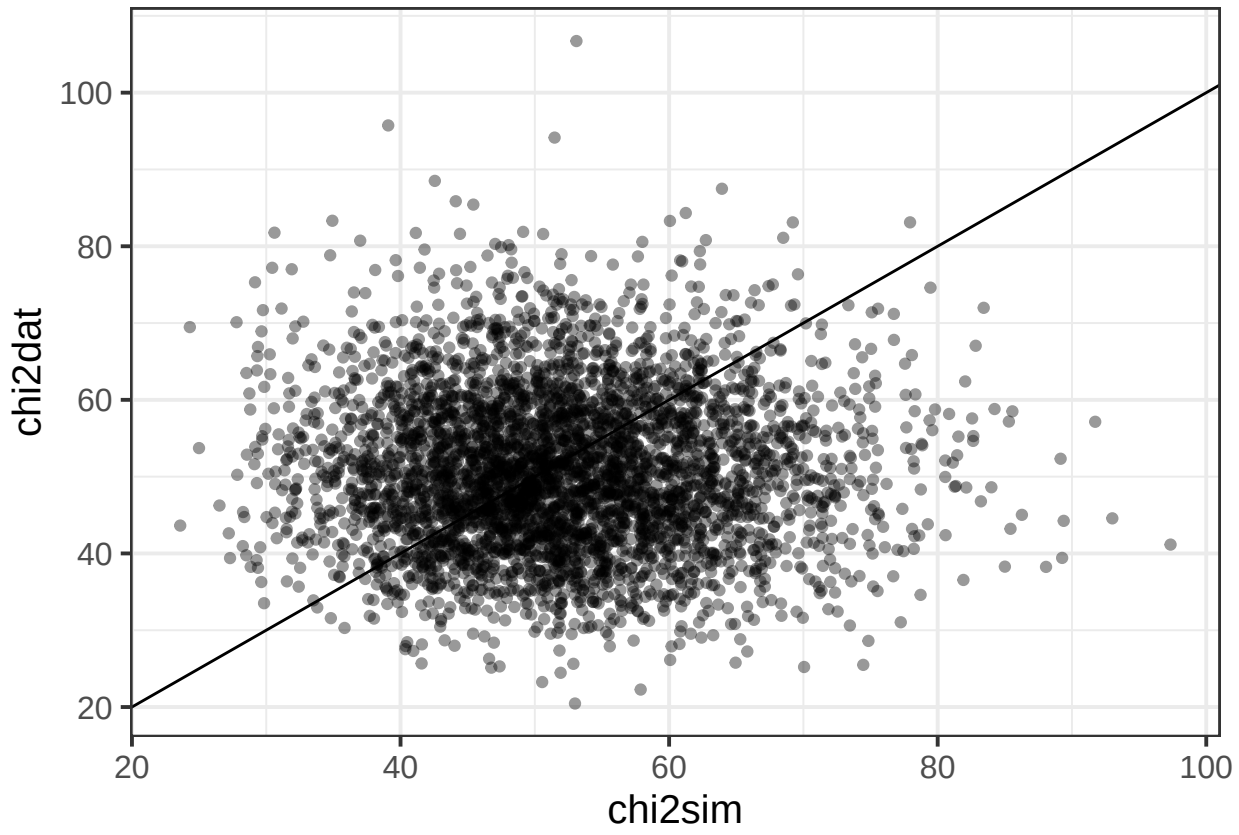

```
deet.data.fit |>
ggplot(aes(sample=Residual))+
  geom_qq()+geom_qq_line()
```



The model assumption are met based on the residual plots, there is no drastic change in residual variance across the predicted values and the residuals fall fairly close to the normal QQ line.

D) Plot the chi-squared discrepancy plot and calculate the Bayesian P value. Is the model adequate by this metric? *plots*

```
chi2.discr <- as_tibble(rstan::extract(fit.stan.reg, pars = c("chi2sim","chi2dat")))

chi2.discr |>
ggplot(aes(x=chi2sim,y=chi2dat))+
  geom_point(alpha=0.4)+
  geom_abline(intercept=0,slope=1)
```




```
print(fit.stan.reg,pars=c("pvalue"))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##        mean se_mean  sd 2.5% 25% 50% 75% 97.5% n_eff Rhat
## pvalue 0.46    0.01 0.5    0   0   0   1     1  4042    1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 22 22:31:24 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

By this metric the fit seems adequate. The points in the plot fall equally above and below the line, thoough there is a small group at chi2sim values greater than 85 that only fall below the line. The bayesian pvalue of 0.47 is close to the ideal of 0.5. So the group that only fell below didn't appear to cause any dramatic problems in fit.

E) Say we want to use the model to predict a plausible range of values for the number of bites a particular volunteer might get with a DEET dose of exactly 2.8. Add this calculation to your STAN code and give the mean and 95% interval of this person's square root transformed number of bites. note) I added the transformed dose to the stan.deet.data

```r
print(fit.stan.reg,pars=c("volMean","volPred"))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##         mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
## volMean 2.74    0.00 0.11 2.51 2.66 2.74 2.82  2.97  3290    1
## volPred 2.73    0.01 0.70 1.32 2.27 2.73 3.20  4.15  4129    1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 22 22:31:24 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

A new patient receiving a dose of 2.8 is predicted to have a mean number of bites of 2.74 with a 95% credible interval of (2.52,2.97), and a 95% prediction interval of (1.35,4.11).