

CGS-698C::Assignment-4

Submitted_by::Anil Yadav(210138)

PART-1:: A simple linear regression: Power posing and testosterone

1.1

```
1 # Install and load necessary libraries
2 if (!requireNamespace("brms", quietly = TRUE)) {
3   install.packages("brms")
4 }
5 if (!requireNamespace("dplyr", quietly = TRUE)) {
6   install.packages("dplyr")
7 }
8
9 library(brms)
10 library(dplyr)
11
12 # Load the data (assuming df_powerpose.csv is available in the working directory)
13 df_powerpose <- read.csv("C:/Users/user/Downloads/download from chrome/df_powerpose_99f68d8a-2128-47c5-84c4-93af776c0468.csv", header = TRUE, sep = ",")
14
15 # Display the first few rows of the data
16 head(df_powerpose)
17
18 # Create a new column for the change in testosterone levels
19 df_powerpose <- df_powerpose %>%
20   mutate(change = testm2 - testm1)
21
22 # Fit the Bayesian linear model
23 fit_powerpose <- brm(
24   formula = change ~ hptreat,
25   data = df_powerpose,
26   family = gaussian(),
27   iter = 2000,
28   chains = 4,
29   cores = 4
30 )
31
32 # Display the model summary
33 summary(fit_powerpose)
34
35 # Plot the model results
36 plot(fit_powerpose)
37
38 # Inspect the default priors used by the model
39 prior_summary(fit_powerpose)
40
41 # Get the default priors even before fitting the model
42 get_prior(change ~ hptreat, data = df_powerpose)
```

```

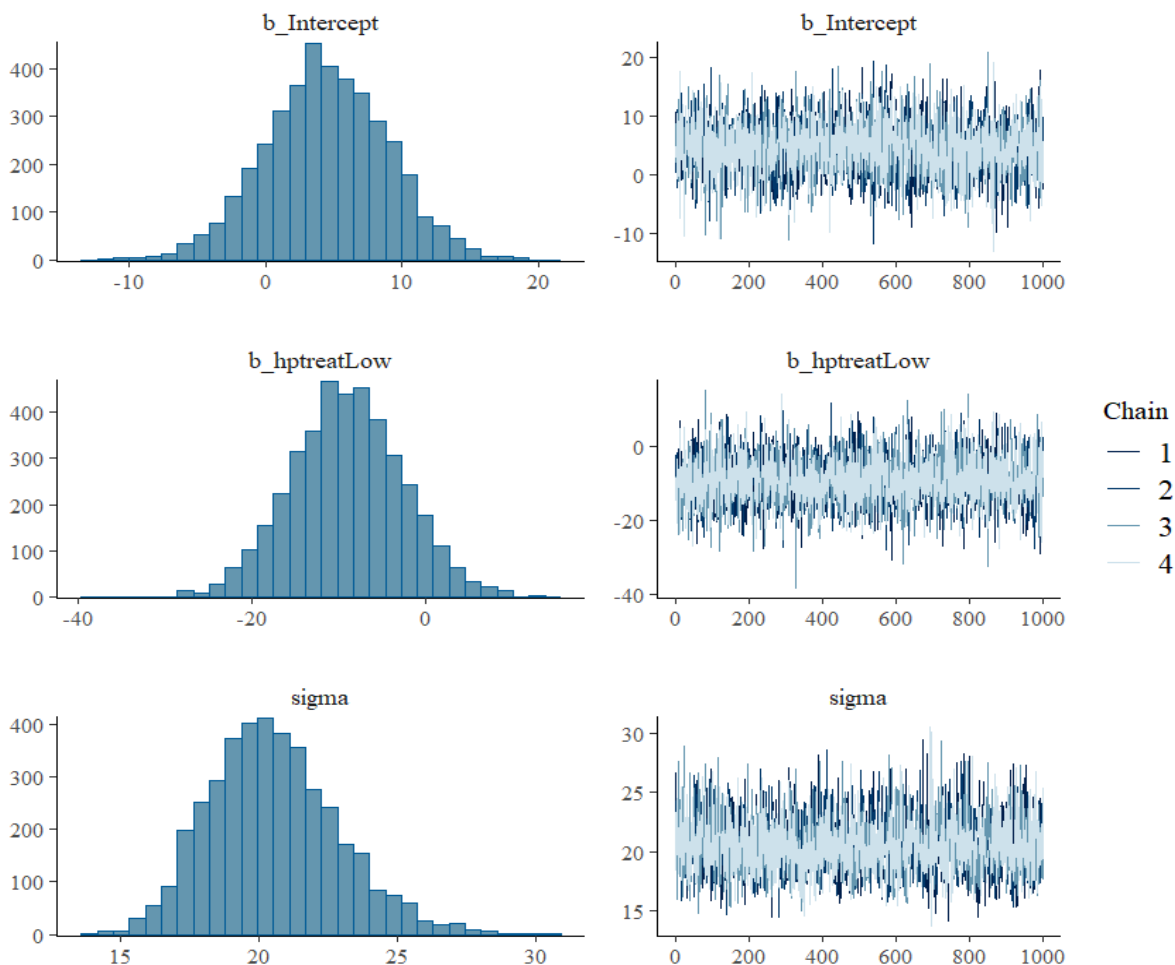
Family: gaussian
Links: mu = identity; sigma = identity
Formula: change ~ hptreat
Data: df_powerpose (Number of observations: 39)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000

Regression Coefficients:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept      4.49      4.53   -4.35   13.42  1.00    4146    3181
hptreatLow     -9.03      6.53   -21.91    3.79  1.00    4294    3182

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     20.53      2.33    16.49    25.52  1.00    3623    2978

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
>
> # Plot the model results
> plot(fit_powerpose)
>
> # Inspect the default priors used by the model
> prior_summary(fit_powerpose)
      prior      class      coef group resp dpar nlprior lb
student_t(3, -1.9, 18.7) Intercept      b
student_t(3, 0, 18.7)      sigma      b hptreatLow
ub      source
default
(vectorized)
default
default
>
> # Get the default priors even before fitting the model
> get_prior(change ~ hptreat, data = df_powerpose)
      prior      class      coef group resp dpar nlprior lb
student_t(3, -1.9, 18.7) Intercept      b
student_t(3, 0, 18.7)      sigma      b hptreatLow
ub      source
default

```



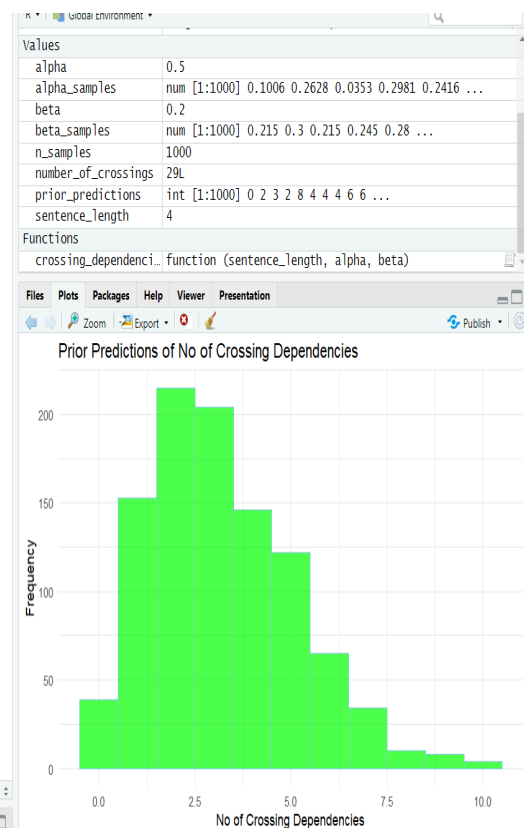
Part 2: Poisson regression models and hypothesis testing

2.1

```
1 # Install and load necessary library
2 if (!requireNamespace("MASS", quietly = TRUE)) {
3   install.packages("MASS")
4 }
5 library(MASS)
6
7 # Define the function to calculate number of crossings
8 crossing_dependencies <- function(sentence_length, alpha, beta) {
9   # Calculate lambda (expected rate of crossings)
10  lambda <- exp(alpha + beta * sentence_length)
11
12  # Simulate the number of crossing dependencies using Poisson distribution
13  crossings <- rpois(1, lambda)
14
15  return(crossings)
16 }
17
18 # Example usage
19 sentence_length <- 13
20 alpha <- 0.55
21 beta <- 0.45
22 number_of_crossings <- crossing_dependencies(sentence_length, alpha, beta)
23 summary(crossing_dependencies(sentence_length, alpha, beta))
24 print(number_of_crossings)
25
26
```

2.2

```
1 if (!requireNamespace("ggplot2", quietly = TRUE)) {
2   install.packages("ggplot2")
3 }
4 library(ggplot2)
5
6 # Define the function to calculate number of crossings
7 crossing_dependencies <- function(sentence_length, alpha, beta) {
8   lambda <- exp(alpha + beta * sentence_length)
9   crossings <- rpois(1, lambda)
10  return(crossings)
11 }
12
13 sentence_length <- 4
14 n_samples <- 1000 # Number of prior samples
15
16 # Generate prior predictions
17 alpha_samples <- rnorm(n_samples, mean = 0.15, sd = 0.1)
18 beta_samples <- rnorm(n_samples, mean = 0.25, sd = 0.05)
19
20 # Generate prior predictions
21 prior_predictions <- sapply(1:n_samples, function(i) {
22   crossing_dependencies(sentence_length, alpha_samples[i], beta_samples[i])
23 })
24
25 summary(prior_predictions)
26
27 # Plot the prior predictions
28 df <- data.frame(prior_predictions)
29 ggplot(df, aes(x = prior_predictions)) +
30   geom_histogram(binwidth = 1, fill = "green", color = "sky blue", alpha = 0.7) +
31   labs(title = "Prior Predictions of No of Crossing Dependencies",
32        x = "No of Crossing Dependencies",
33        y = "Frequency") +
34   theme_minimal()
35
```



2.4

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

library(brms)
library(readr)
library(dplyr)
library(ggplot2)

observed <- read_csv("C:/Users/user/Downloads/download from chrome/crossings_24a167f3-2f8f-4f5c-bca9-884567bb1c33.csv")

observed$language <- as.factor(observed$language)
# Visualize average rate of crossings

observed %>%
  group_by(language, s.length) %>%
  summarise(mean_crossings = mean(ncross)) %>%
  ggplot(aes(x = s.length, y = mean_crossings, group = language, color = language)) +
  geom_point() +
  geom_line()

observed$s.length <- observed$s.length - mean(observed$s.length)

observed$lang <- ifelse(observed$language == "German", 1, 0)

lpls.m1 <- c()
lpls.m2 <- c()
untested <- observed

# Number of folds
k_folds <- 5

for (k in 1:k_folds) {
  ytest <- sample_n(untested, size = nrow(observed) / k_folds)
  ytrain <- setdiff(observed, ytest)
  untested <- setdiff(untested, ytest)

  fit.m1 <- brm(
    ncross ~ 1 + s.length,
    data = ytrain,
    family = poisson(link = "log")
  )
}
```

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

prior = c(prior(normal(0.15, 0.1), class = Intercept), prior(normal(0, 0.15), class = b)),
cores = 4)

fit.m2 <- brm(
  ncross ~ 1 + s.length + lang + s.length * lang,
  data = ytrain,
  family = poisson(link = "log"),
  prior = c(prior(normal(0.15, 0.1), class = Intercept), prior(normal(0, 0.15), class = b)),
  cores = 4)

post.m1 <- posterior_samples(fit.m1)
post.m2 <- posterior_samples(fit.m2)

lppd.m1 <- 0
lppd.m2 <- 0

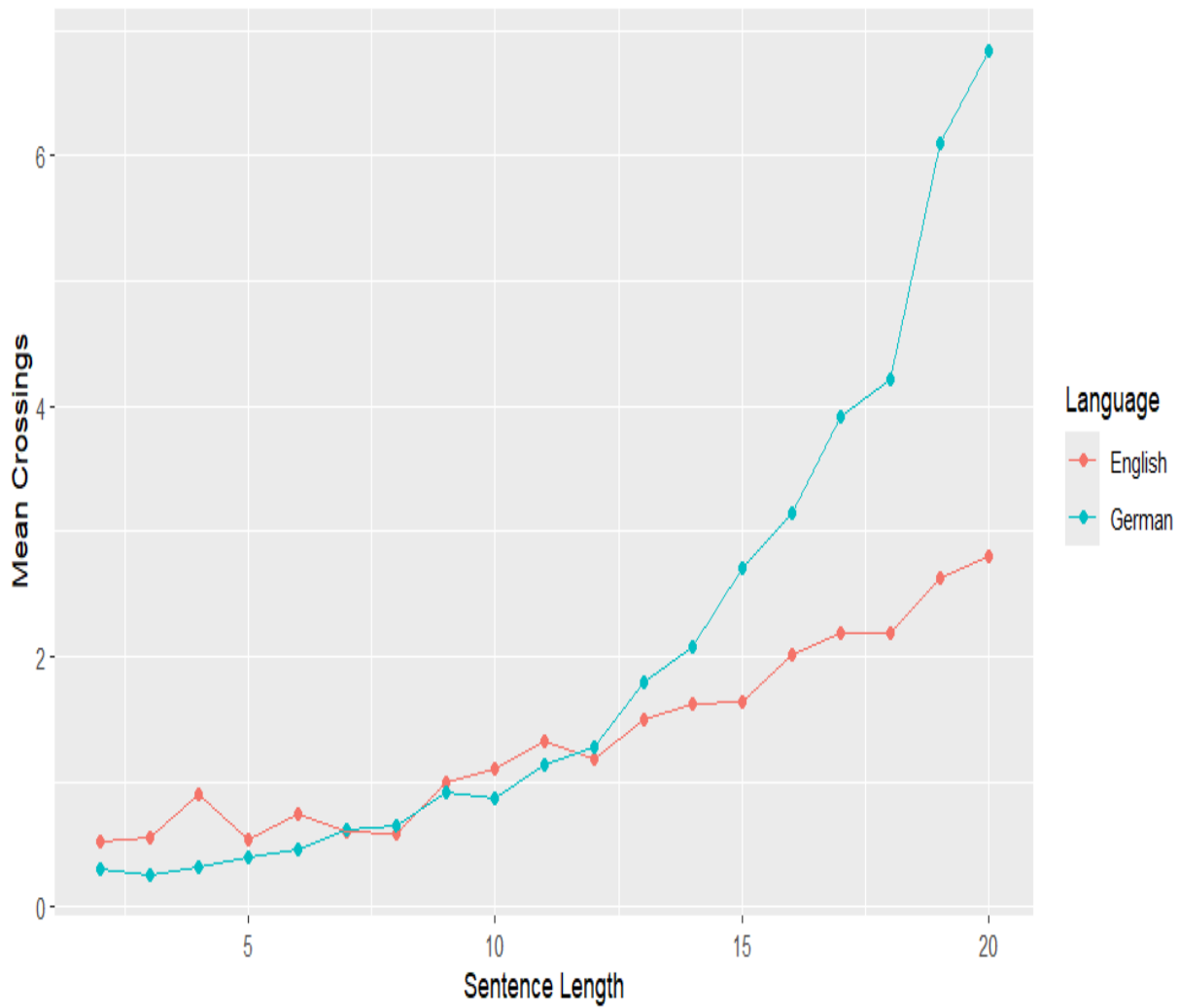
for (i in 1:nrow(ytest)) {
  lpd_im1 <- log(mean(dpois(ytest[i, ]$ncross,
    lambda = exp(post.m1[, 1] + post.m1[, 2] * ytest[i, ]$s.length)
  )))
  lppd.m1 <- lppd.m1 + lpd_im1

  lpd_im2 <- log(mean(dpois(ytest[i, ]$ncross,
    lambda = exp(post.m2[, 1] +
      post.m2[, 2] * ytest[i, ]$s.length +
      post.m2[, 3] * ytest[i, ]$lang +
      post.m2[, 4] * ytest[i, ]$s.length * ytest[i, ]$lang
    )))
  lppd.m2 <- lppd.m2 + lpd_im2
}

lpls.m1 <- c(lpls.m1, lppd.m1)
lpls.m2 <- c(lpls.m2, lppd.m2)
}

e1pd.m1 <- sum(lpls.m1)
```

Average Rate of Crossings by Sentence Length and Language



Model-1, elpd~ **-3.041.78**

Model-2, elpd~ **-2687.31**

Therefore M2 is better than M1 bcz $M2 > M1$

2.3

```
2 library(brms)
3 library(readr)
4
5
6 data <- read_csv("C:/Users/user/Downloads/download from chrome/crossings_24a167f3-2f8f-4
7
8
9 head(data)
10
11 data$Language <- as.factor(data$Language)
12
13
14 priors <- c(
15   prior(normal(0.15, 0.1), class = "Intercept"),
16   prior(normal(0, 0.15), class = "b")
17 )
18
19 # Model M1
20 fit_M1 <- brm(
21   formula = nCross ~ 1 + s.length,
22   data = data,
23   family = poisson(),
24   prior = priors
25 )
26
27 # Model M2
28 fit_M2 <- brm(
29   formula = nCross ~ 1 + s.length + Language + s.length:Language,
30   data = data,
31   family = poisson(),
32   prior = priors
33 )
34
35 # Summarize the results
36 summary(fit_M1)
37 summary(fit_M2)
```

```
Chain 4: 3.684 seconds (Sampling)
Chain 4: 7.665 seconds (Total)
Chain 4:
```

```
>
> # Summarize the results
> summary(fit_M1)
Family: poisson
Links: mu = log
Formula: nCross ~ 1 + s.length
Data: data (Number of observations: 1900)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-1.45	0.06	-1.56	-1.33	1.00	1158	1544
s.length	0.15	0.00	0.14	0.16	1.00	1368	2099

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS and Tail_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

```
> summary(fit_M2)
Family: poisson
Links: mu = log
Formula: nCross ~ 1 + s.length + Language + s.length:Language
Data: data (Number of observations: 1900)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS
Intercept	-1.10	0.07	-1.25	-0.96	1.00	1512
s.length	0.11	0.01	0.10	0.12	1.00	1594
LanguageGerman	-0.62	0.09	-0.80	-0.44	1.00	1435
s.length:LanguageGerman	0.07	0.01	0.06	0.08	1.00	1368

	Tail_ESS
Intercept	1370
s.length	1636
LanguageGerman	1389
s.length:LanguageGerman	1525

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS and Tail_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

```
> |
```

