Assignment 6

anonymous

1 General information

> ! Reporting accuracy
>
> **For posterior statistics of interest, only report digits for which the Monte Carlo standard error (MCSE) is zero.**
> *Example:* If you estimate $E(\mu) = 1.234$ with $\text{MCSE}(E(\mu)) = 0.01$, you should report $E(\mu) = 1.2$.
> See lecture video 4.1, the chapter notes, and a case study for more information.

This is the template for assignment 6. You can download the qmd-file or copy the code from this rendered document after clicking on `</> Code` in the top right corner.

Please replace the instructions in this template by your own text, explaining what you are doing in each exercise.

2 Stan warm-up: linear model of BDA retention with Stan (2 points)

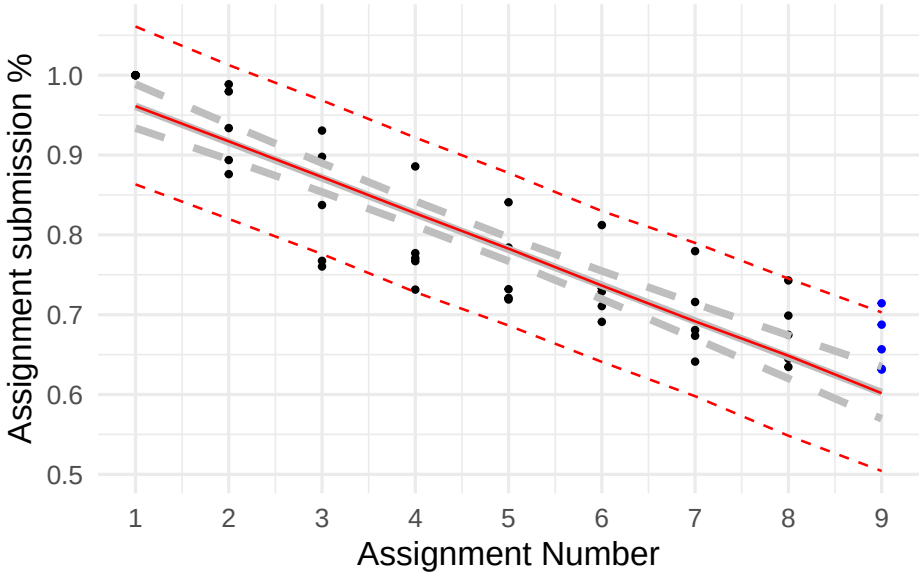
Write your answers/code here!

```
# These are our observations y: the proportion of students handing in each assignment (1-
# sorted by year (row-wise) and assignment (column-wise).
# While the code suggest a matrix structure,
# the result will actually be a vector of length N = no_years * no_assignments
propstudents<-c(c(176, 174, 158, 135, 138, 129, 126, 123)/176,
                c(242, 212, 184, 177, 174, 172, 163, 156)/242,
                c(332, 310, 278, 258, 243, 242, 226, 224)/332,
                c(301, 269, 231, 232, 217, 208, 193, 191)/301,
                c(245, 240, 228, 217, 206, 199, 191, 182)/245)
# These are our predictors x: for each observation, the corresponding assignment number.
```

```r
assignment <- rep(1:8, 5)
# These are in some sense our test data: the proportion of students handing in the last a
# sorted by year.
# Usually, we would not want to split our data like that and instead
# use e.g. Leave-One-Out Cross-Validation (LOO-CV, see e.g. http://mc-stan.org/loo/index.
# to evaluate model performance.
propstudents9 = c(121/176, 153/242, 218/332, 190/301, 175/245)
# The total number of assignments
no_assignments = 9
# The assignment numbers for which we want to generate predictions
x_predictions = 1:no_assignments
# (Cmd)Stan(R) expects the data to be passed in the below format:
model_data = list(N=length(assignment),
                  x=assignment,
                  y=propstudents,
                  no_predictions=no_assignments,
                  x_predictions=x_predictions)
# This reads the file at the specified path and tries to compile it.
# If it fails, an error is thrown.
retention_model = cmdstan_model("./assignment6_linear_model.stan")
# This "out <- capture.output(...)" construction suppresses output from cmdstanr
# See also https://github.com/stan-dev/cmdstanr/issues/646
out <- capture.output(
    # Sampling from the model happens here:
    fit <- retention_model$sample(data=model_data, refresh=0, show_messages=FALSE)
)
# This extracts the draws from the sampling result as a data.frame.
draws_df = fit$draws(format="draws_df")

# This does some data/draws wrangling to compute the 5, 50 and 95 percentiles of
# the mean at the specified covariate values (x_predictions).
# It can be instructive to play around with each of the data processing steps
# to find out what each step does, e.g. by removing parts from the back like "|>  gather(
# and printing the resulting data.frame.
mu_quantiles_df = draws_df |>
      select(starts_with(c("mu_pred"))) |>
      apply(2, quantile, c(0.05, 0.5, 0.95)) |>
      t() |> data.frame(x=x_predictions) |>  gather(pct,y,-x)
# Same as above, but for the predictiions.
y_quantiles_df = draws_df |>
      select(starts_with(c("y_pred"))) |>
      apply(2, quantile, c(0.05, 0.5, 0.95)) |>
      t() |> data.frame(x=x_predictions) |>  gather(pct,y,-x)
```

```r
# Plotting happens here:
ggplot() +
  # scatter plot of the training data:
  geom_point(aes(x, y), data = data.frame(x=assignment, y=propstudents), size = 1) +
  # scatter plot of the test data:
  geom_point(aes(x, y), data = data.frame(x=no_assignments, y=propstudents9), size = 1, c
  # you have to tell us what this plots:
  geom_line(aes(x,y,linetype=pct), data=mu_quantiles_df, color = 'grey', linewidth=1.5) +
  # you have to tell us what this plots:
  geom_line(aes(x,y,linetype=pct), data=y_quantiles_df, color = 'red') +
  # adding xticks for each assignment:
  scale_x_continuous(breaks=1:no_assignments) +
  # adding labels to the plot:
  labs(y = "Assignment submission %", x= "Assignment Number") +
  # specifying that line types repeat:
  scale_linetype_manual(values=c(2,1,2)) +
  # remove the legend for the linetypes:
  guides(linetype = "none")
```



Write your answers/code here!

3 Generalized linear model: Bioassay with Stan (4 points)

Write your answers/code here!

```
data("bioassay")
```

Write your answers/code here!

Write your answers/code here!

Write your answers/code here!