Assignment 6

anonymous

1 General information

I did not use AI for solving this exercise.

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

2.1 (a)

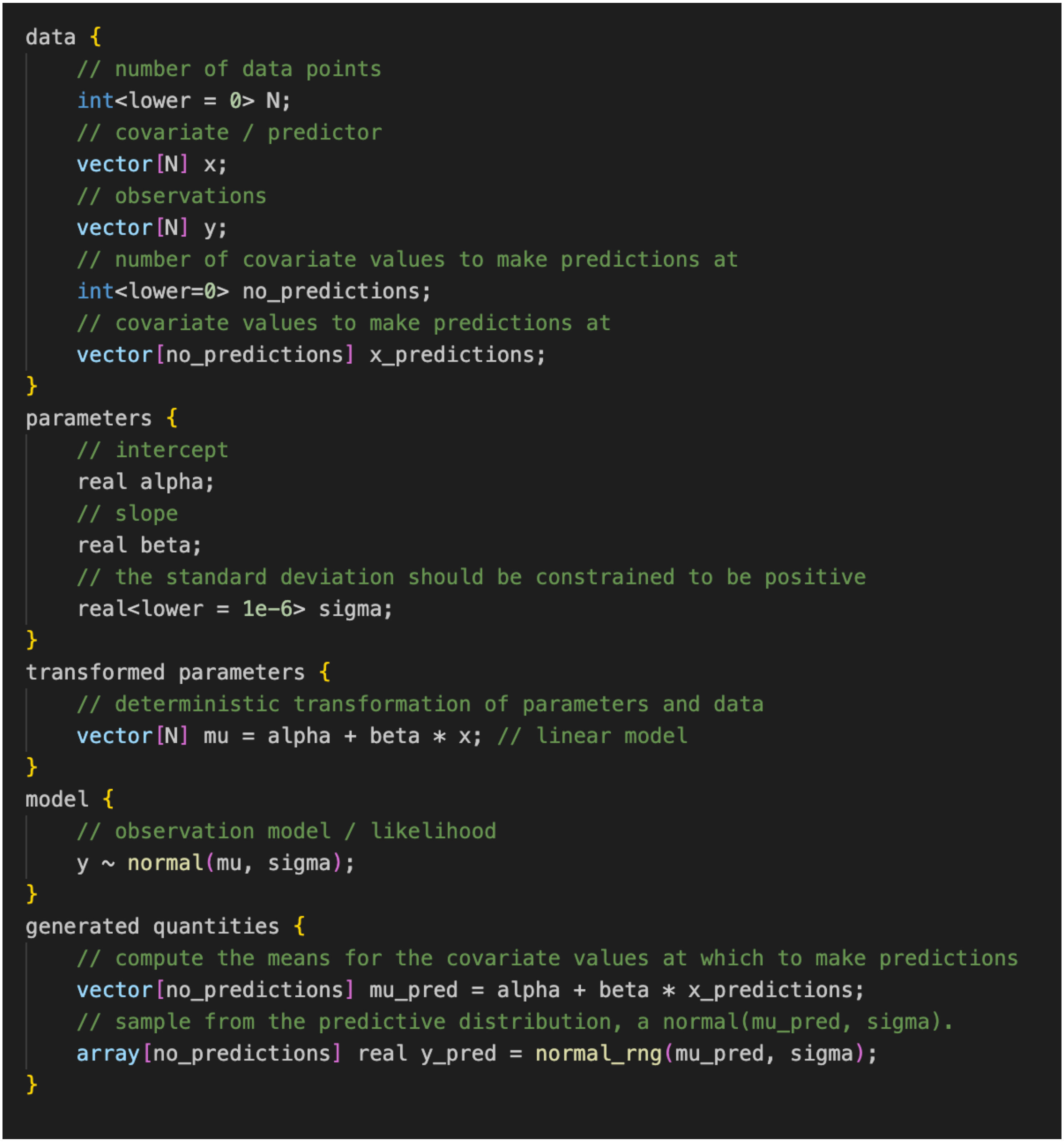
Error 1: On Line 23, a semicolon was absent within the transformed parameters {} block, and it has been added to rectify this issue.

Error 2: At Line 19, it is imperative that the sigma parameter has a value strictly greater than zero. Consequently, the parameter ‘real<upper=0> sigma;’ in the parameters block was replaced with ‘real<lower=1e-6> sigma;’ to meet this requirement.

Error 3: Located at Line 33, the use of ‘mu’ to compute ‘y_pred’ in generated quantities is incorrect due to a dimensional disparity. Therefore, ‘mu’ has been substituted with ‘mu_pred’ in the normal_rng function.

The corrected code is shown in below:

```r
knitr::include_graphics("lin-model.png")
```



```stan
data {
    // number of data points
    int<lower = 0> N;
    // covariate / predictor
    vector[N] x;
    // observations
    vector[N] y;
    // number of covariate values to make predictions at
    int<lower=0> no_predictions;
    // covariate values to make predictions at
    vector[no_predictions] x_predictions;
}
parameters {
    // intercept
    real alpha;
    // slope
    real beta;
    // the standard deviation should be constrained to be positive
    real<lower = 1e-6> sigma;
}
transformed parameters {
    // deterministic transformation of parameters and data
    vector[N] mu = alpha + beta * x; // linear model
}
model {
    // observation model / likelihood
    y ~ normal(mu, sigma);
}
generated quantities {
    // compute the means for the covariate values at which to make predictions
    vector[no_predictions] mu_pred = alpha + beta * x_predictions;
    // sample from the predictive distribution, a normal(mu_pred, sigma).
    array[no_predictions] real y_pred = normal_rng(mu_pred, sigma);
}
```

2.2 (b)

Plotting happens here:

```
ggplot() +
  # scatter plot of the training data:
  geom_point(
    aes(x, y, color=assignment),
    data=data.frame(x=assignment, y=propstudents, assignment="1-8")
) +
  # scatter plot of the test data:
  geom_point(
    aes(x, y, color=assignment),
    data=data.frame(x=no_assignments, y=propstudents9, assignment="9")
) +
  # you have to tell us what this plots:
  geom_line(aes(x,y=value,linetype=pct), data=mu_quantiles_df, color='grey', linewidth=1.5) +
  # you have to tell us what this plots:
  geom_line(aes(x,y=value,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)) +
  # Specify colours of the observations:
  scale_colour_manual(values = c("1-8"="black", "9"="blue")) +
  # remove the legend for the linetypes:
  guides(linetype="none")
```

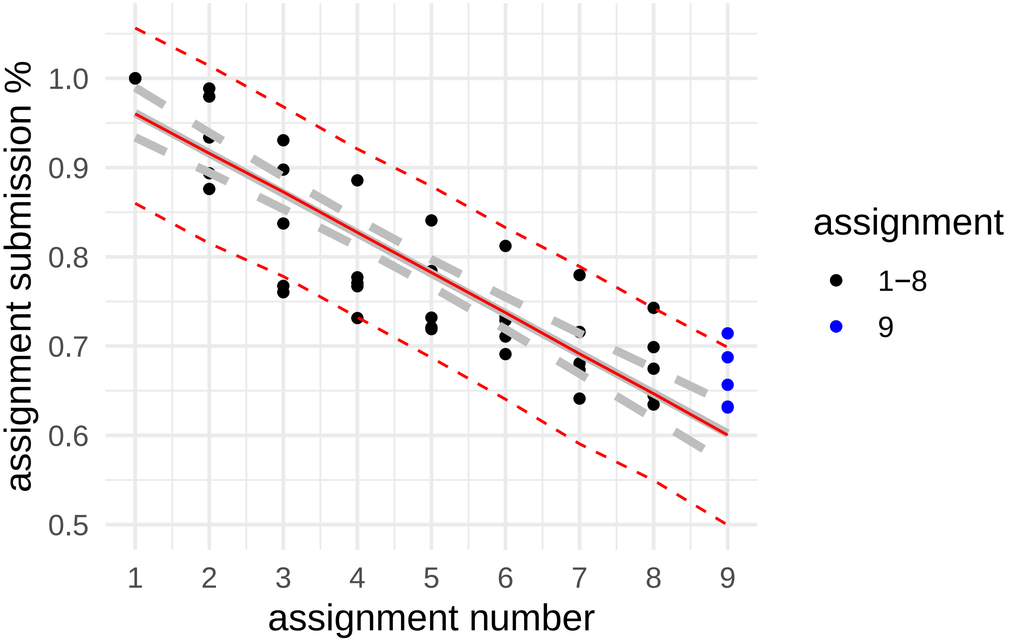


Figure 1: Describe me in your submission!

The output of cmdstan_diagnose is shown in below, and we can conclud that:

1. All chains demonstrated convergence
2. Effective sample size satisfactory shows that the sampling process was successful.

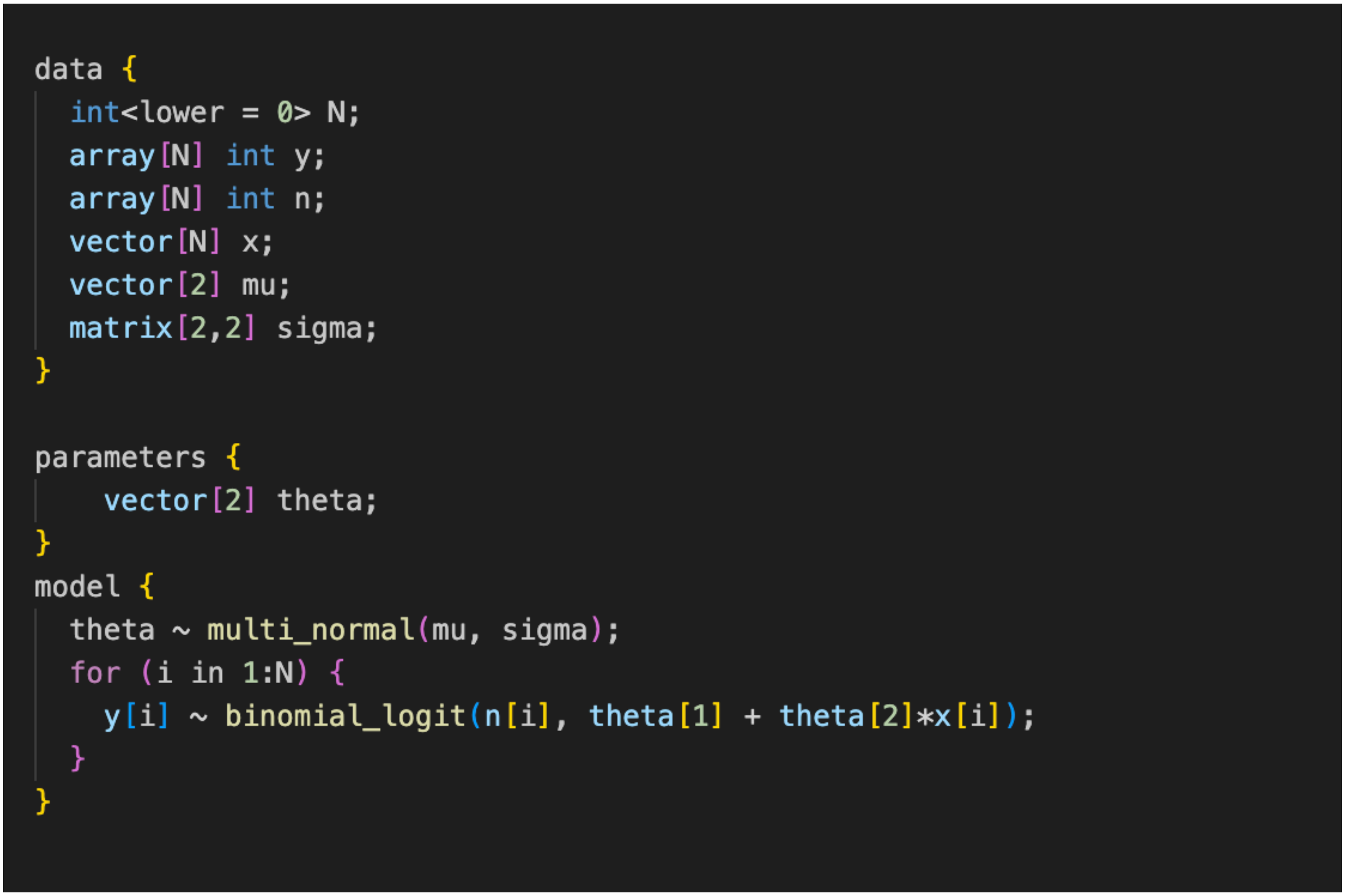
```
Checking sampler transitions treedepth. Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences. No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy. E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
```

1. The plot consists of the following lines:
   - A solid red line represents the median of the predicted assignment submission percentage.
   - Two dotted lines, one lower and one upper, correspond to the 5th and 95th quantiles of the predicted assignment submission percentage.
   - These predictions are generated using a normal model.
   - A solid gray line represents the median of the mean parameter of the same normal model. Two dotted gray lines, lower and upper, denote the 5th and 95th quantiles of the mean parameter.
   - Remarkably, the median values of both the predicted assignment submission percentage and the predicted mean parameter align perfectly.
   - However, it's important to note that the 5th quantile of the assignment submission percentage is lower than that of the mean parameter, and the 95th quantile of the assignment percentage is greater than that of the mean parameter.
2. As the assignment number increases, there's a linear decrease in the percentage of submission. This decline implies a reduction in student retention, as evidenced by the negative slope.
3. When comparing actual values to predictions:
   - The actual values tend to be greater than the median of the predicted submission percentage.
   - In some cases, actual values surpass the predicted 95th quantile.
   - Most actual values, however, fall within the range between the median and the 95th quantile, suggesting that the model performs well.
4. It's worth noting that averaging the submission percentage for each year before fitting the model could have been considered.

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

3.1 (a)

```
knitr::include_graphics("bioassay_model.png")
```



```
data {
  int<lower = 0> N;
  array[N] int y;
  array[N] int n;
  vector[N] x;
  vector[2] mu;
  matrix[2,2] sigma;
}

parameters {
    vector[2] theta;
}
model {
  theta ~ multi_normal(mu, sigma);
  for (i in 1:N) {
    y[i] ~ binomial_logit(n[i], theta[1] + theta[2]*x[i]);
  }
}
```

```
data("bioassay")
   cov_mat <- cbind(c(4,12), c(12,100))
   mean <- c(0,10)
   model_data = list(N=nrow(bioassay),
                    y=bioassay$y,
                    n=bioassay$n,
                    x=bioassay$x,
                    mu=mean,
                    sigma = cov_mat)
   retention_model = cmdstan_model("./additional_files/assignment6/bioassay_model.stan")
```

```
Warning in readLines(stan_file): incomplete final line found on
'./additional_files/assignment6/bioassay_model.stan'
```

```
   out <- capture.output(
       fit <- retention_model$sample(data=model_data, refresh=0, show_messages=FALSE)
```

```
)
fit$cmdstan_diagnose()
```

```
Processing csv files: /var/folders/11/n0lc1krs0gb265hzfktnr3l40000gp/T/Rtmp6WIXQk/bioassay_model-20231

Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
```

3.2 (b)

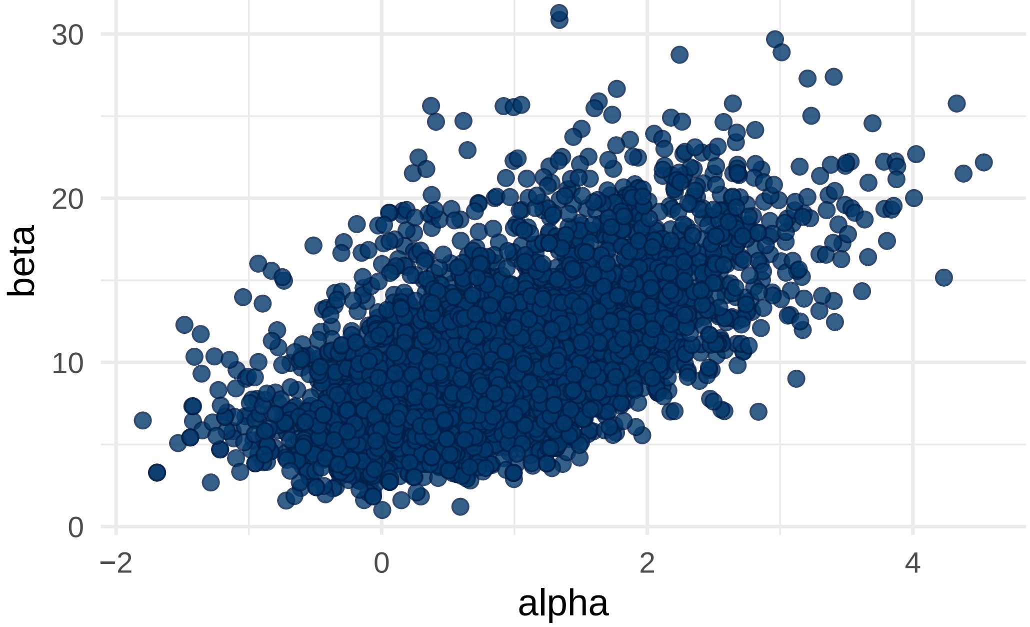
```
rhat_alpha = fit$summary()$rhat[2]
rhat_beta = fit$summary()$rhat[3]
cat('rhat alpha = ', rhat_alpha, '\n', 'rhat beta = ', rhat_beta, '\n')
```

```
rhat alpha =  1.000825 
 rhat beta =  1.00046 
```

We can see that both rhat values are less than 1.05, so we can say the chains have converged perfectly.

3.3 (c)

```
draws_f = fit$draws(format="draws_df")
draw_final <- data.frame(alpha = draws_f$`theta[1]`,
                         beta = draws_f$`theta[2]`,
                         chain=draws_f$.chain)
mcmc_scatter(draw_final, pars=c("alpha", "beta"))
```



3.4 (d)

- ***Operating system (Linux, Mac, Windows) or jupyter.cs.aalto.fi?*** Mac
- ***Programming environment used: R or Python?*** R
- ***Interface used: RStan, CmdStanR, PyStan, or CmdStanPy?*** CmdStanR
- ***Did you have installation or compilation problems? Did you try first installing locally, but switched to jupyter.cs.aalto.fi?*** I did all things locally, and there was just a mistake in referring to the stan file in the first task. Besides that, it was fine.
- ***In addition of these you can write what other things you found out difficult (or even frustrating) when making this assignment with Stan.*** The examples in Stan are incredibly valuable; however, in my opinion, the reference manual for the functions is lacking the level of detail I would find more comprehensive and helpful.