STSB6816 Test 2 of 2022

Mathematical Statistics and Actuarial Science; University of the Free State

2022/05/26

Time: 180 minutes; Marks: 50

MEMORANDUM

Instructions

- Answer all questions in a single R Markdown document. Please knit to Word or PDF at the end and submit both the PDF/Word document and the .Rmd file for assessment, in that order.
- Label questions clearly, as it is done on this question paper.
- All results accurate to at least 1 decimal place, ensure that simulation error is small enough (by doing enough simulations).
- Show all derivations, formulas, code, sources and reasoning.
- Intervals should cover 95% probability unless stated otherwise.
- No communication software, no devices, and no communication capable websites may be accessed prior to submission. You may not (nor even appear to) attempt to communicate or pass information to another student.

Question 1

A patient is asked to take their blood sugar readings every morning for four weeks. They present the following to the doctor:

8.6, 5.6, 7.2, Forgot, 6.9, 6.8, 5.4, 6.8, 5.9, Didn't have time, 4.4, 5.9, 4.8, 5.2, 3.7, 5.3, ? (illegible), 3.4, 4, 4.2, 3.1, BDL, 3.6, BDL, 3.2, BDL, BDL, BDL; where BDL means below the detection limit of 3.0 on the home machine.

The doctor asks you to tell them what the missing and censored values might have been, with uncertainty.

1.1) Capture the data. Ensure that you have columns for Day (1 to 28), Values (numeric), Missing (indicator with 3 1s), and BDL (indicator with 5 1s). Check that you have captured the data correctly by giving a summary and showing that the observed values sum to 104. The data is also given in tabular format below if that makes capturing easier for you. **[4]**

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Week 1	8.6	5.6	7.2	Forgot	6.9	6.8	5.4
Week 2	6.8	5.9	Didn't have time	4.4	5.9	4.8	5.2
Week 3	3.7	5.3	? (illegible)	3.4	4	4.2	3.1

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Week 4	BDL	3.6	BDL	3.2	BDL	BDL	BDL

```
sugar <- data.frame(Day = 1:28, Values = c(8.6, 5.6, 7.2, NA, 6.9, 6.8, 5.4, 6.8,
5.9, NA, 4.4, 5.9, 4.8, 5.2, 3.7, 5.3, NA, 3.4, 4, 4.2, 3.1, NA, 3.6, NA, 3.2, NA,
NA, NA), Missing = c(rep(0,3), 1, rep(0,5), 1, rep(0,6), 1, rep(0, 11)), BDL =
c(rep(0, 21), 1, 0, 1, 0, 1, 1, 1))
summary(sugar)
```

```
|    Day       Values      Missing       BDL
| Min.  : 1.00  Min.  :3.100  Min.  :0.0000  Min.  :0.0000
| 1st Qu.: 7.75  1st Qu.:3.925  1st Qu.:0.0000  1st Qu.:0.0000
| Median :14.50  Median :5.250  Median :0.0000  Median :0.0000
| Mean  :14.50  Mean  :5.200  Mean  :0.1071  Mean  :0.1786
| 3rd Qu.:21.25  3rd Qu.:6.125  3rd Qu.:0.0000  3rd Qu.:0.0000
| Max.  :28.00  Max.  :8.600  Max.  :1.0000  Max.  :1.0000
|         NA's  :8
```

```
sum(sugar$Values, na.rm = TRUE)
```

```
| [1] 104
```

Typing in numbers correctly [2], showing summaries [2].

1.2) Fit a linear model with Student-t errors through the points on the log scale (equivalent to an exponential slope on the original scale). Use the objective log prior $\log\pi(v,m,s) = log(v) - 3\log(v+0.75) - 2\log(s) + c_1$. Incorporate the censored observations appropriately. Give a summary of the parameters. **[20]**

HINT First fit the model with a normal distribution on only the complete data and make sure you get sensible results. You can do the rest of the test and get everything nearly right on that alone. Then switch to *t* and add the prior, and lastly incorporate the censoring only when you are sure it is working without it.

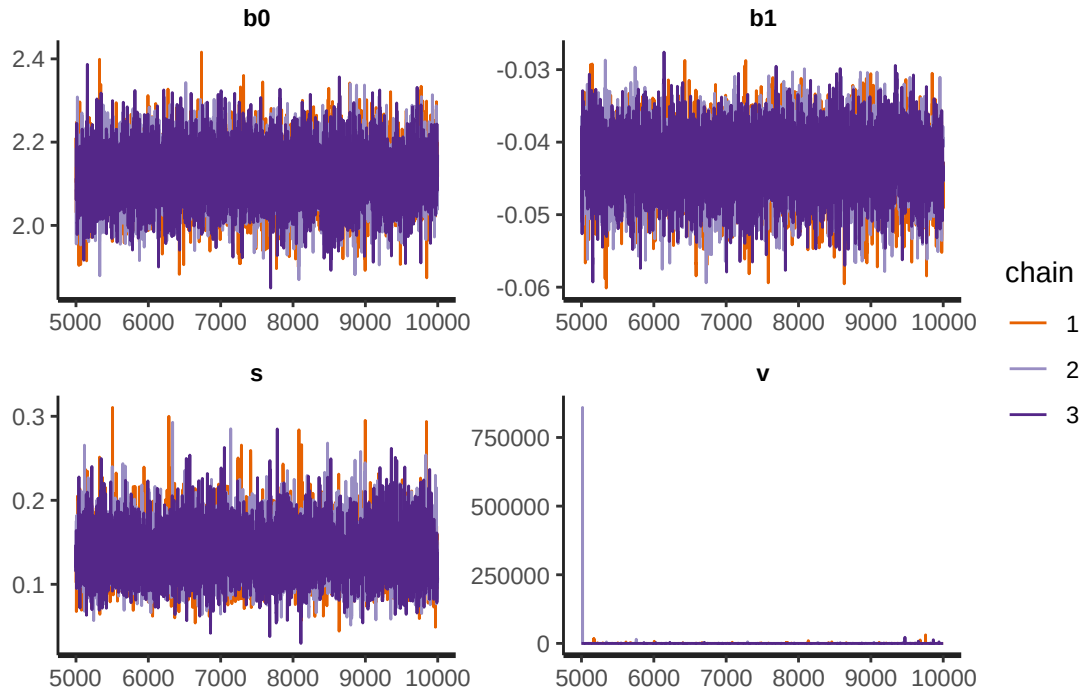
```
library(rstan)
mycores <- 3
options(mc.cores = mycores)

// This Stan block defines a simple Student-t regression model with fixed lower
censoring point, by Sean van der Merwe, UFS
data {
  int<lower=1> n;                  // number of observations
  real y[n];                 // observations
  real x[n];                 // explanatory variables
  int<lower=1> ncens;                  // number of censored observations
  real ycens;              // censoring point
  real xcens[ncens];             // explanatory variables
}
// The parameters of the model
parameters {
  real b0;
  real b1;
  real<lower=0> s;
  real<lower=0.5> v;
}
model {
  for (i in 1:n) {
    y[i] ~ student_t(v, b0 + b1*x[i], s);
```

```
  }
  for (i in 1:ncens) {
    target += student_t_lcdf(ycens | v, b0 + b1*xcens[i], s);
  }
  target += log(v) - 3*log(v+0.75) - 2*log(s);   // joint objective prior
}

stan_data <- sugar |> subset((Missing == 0) & (BDL == 0))
ModelFit <- sampling(tline, list(n = nrow(stan_data), y = log(stan_data$Values), x =
stan_data$Day, ncens = sum(sugar$BDL), ycens = log(3), xcens = sugar$Day[sugar$BDL
== 1]), iter = 10000, chains = mycores)

ModelFit |> traceplot()
```




```
summary(ModelFit)$summary |> kable(digits = 3)
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
b0	2.123	0.001	0.066	1.991	2.080	2.124	2.167	2.254	5876.893	1
b1	-0.043	0.000	0.004	-0.052	-0.046	-0.043	-0.040	-0.035	6122.702	1
s	0.135	0.000	0.029	0.084	0.116	0.132	0.151	0.199	8134.904	1
v	111.159	61.585	7051.990	1.490	4.070	7.904	19.026	234.474	13112.052	1
lp__	16.715	0.022	1.557	12.817	15.963	17.068	17.860	18.652	4917.015	1

Fitting a regression model to the clean data [4], log scale [2], giving parameter summary [3], t density [2], using the prior as given [3], censoring data incorporated correctly [6].

1.3) For the first missing day (Day 4) obtain a reasonable approximation of the posterior predictive distribution given the model and other data. Give a histogram or density plot of some kind to illustrate it. **[7]**

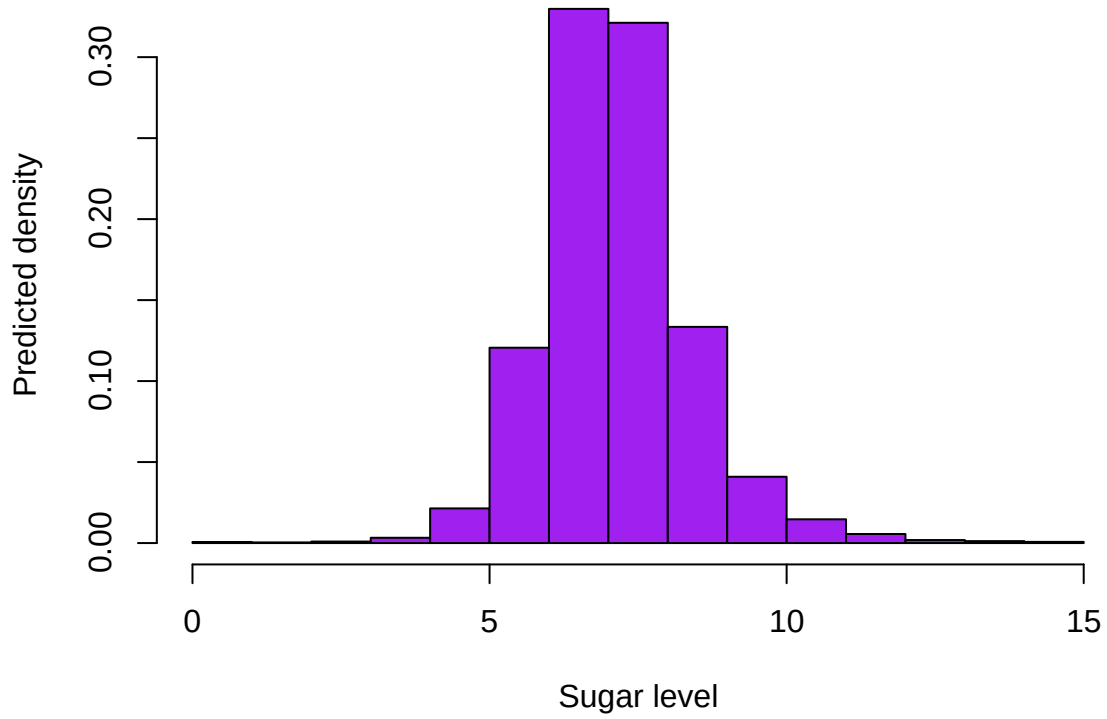
HINT For all plots limit the range of sugar values to lie from 0 to 15.

```
ModelFit |> extract() -> draws

pred_func <- function(day) {
  mu <- draws$b1*day + draws$b0
  nsims <- length(mu)
  preds <- rt(nsims, draws$v)*draws$s + mu
  exp(preds)
}

day4preds <- pred_func(4)

par(mar = c(4.5, 4.5, 0.5, 0.5))
day4preds |> hist(c(0:15, Inf), freq = FALSE, col = 'purple', main = '', xlab =
'Sugar level', ylab = 'Predicted density', xlim = c(0, 15))
```



mu [2], t sims [3], exp [1], hist [1].

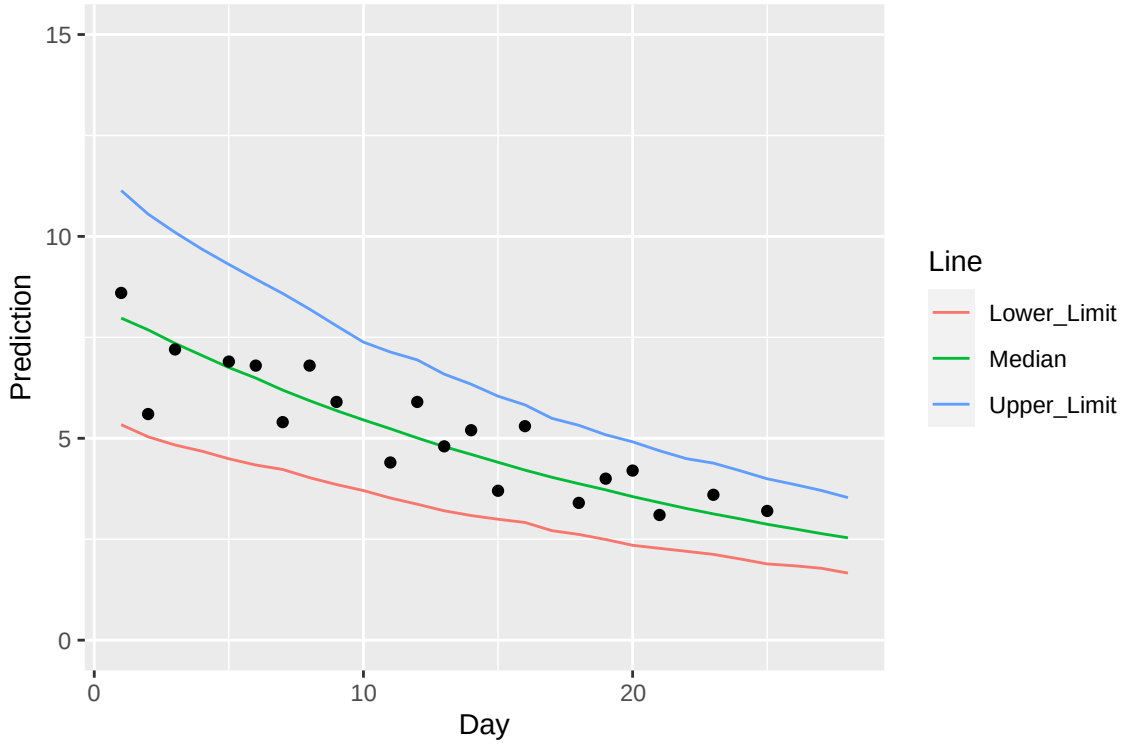
1.4) For each day (1 to 28), obtain a reasonable approximation of the posterior predictive distribution given the model and data. Based on these distributions calculate **median values and highest posterior density intervals for each day**. Produce a plot on the original scale showing these expected values and intervals as curves, as well as the observed data points. **[11]**

```
shortestinterval <- function(postsims, alpha=0.05) { # Coded by Sean van der Merwe,
UFS
  postsims |> sort() -> sorted.postsims
  round(length(postsims)*(1-alpha)) -> gap
  sorted.postsims |> diff(gap) |> which.min() -> pos
  sorted.postsims[c(pos, pos + gap)] }

(1:28) |> sapply(\(day) {
  day |> pred_func() -> preds
  c(median(preds), shortestinterval(preds))
}) |> t() -> allpreds
```

```
suppressPackageStartupMessages(library(tidyverse))
colnames(allpreds) <- c('Median', 'Lower_Limit', 'Upper_Limit')
allpreds |> data.frame() |>
  pivot_longer(everything(), names_to = 'Line', values_to = 'Prediction') |>
  data.frame(Day = rep(1:28, each = 3)) -> plot_data
plot_data |>
  ggplot(aes(x = Day, y = Prediction)) + ylim(0, 15) + geom_line(aes(colour = Line))
+ geom_point(data = sugar, mapping = aes(x = Day, y = Values))
```

```
| Warning: Removed 8 rows containing missing values (geom_point).
```



Medians [3], HPDs [4], Plot with lines visible [2], points added [2]

1.5) Give an assessment of your model fit, including at least: whether the observed coverage is roughly in line with the nominal level and whether the fit seems reasonable. **[4]**

Coverage might be a little high since all the points are in the intervals - model is too conservative [3], but the median fit looks good [1]

1.6) What is the probability that the sugar level will be below detection limit on Day 30? **[2]**

```
mean(pred_func(30) < 3)
```

```
| [1] 0.9317333
```

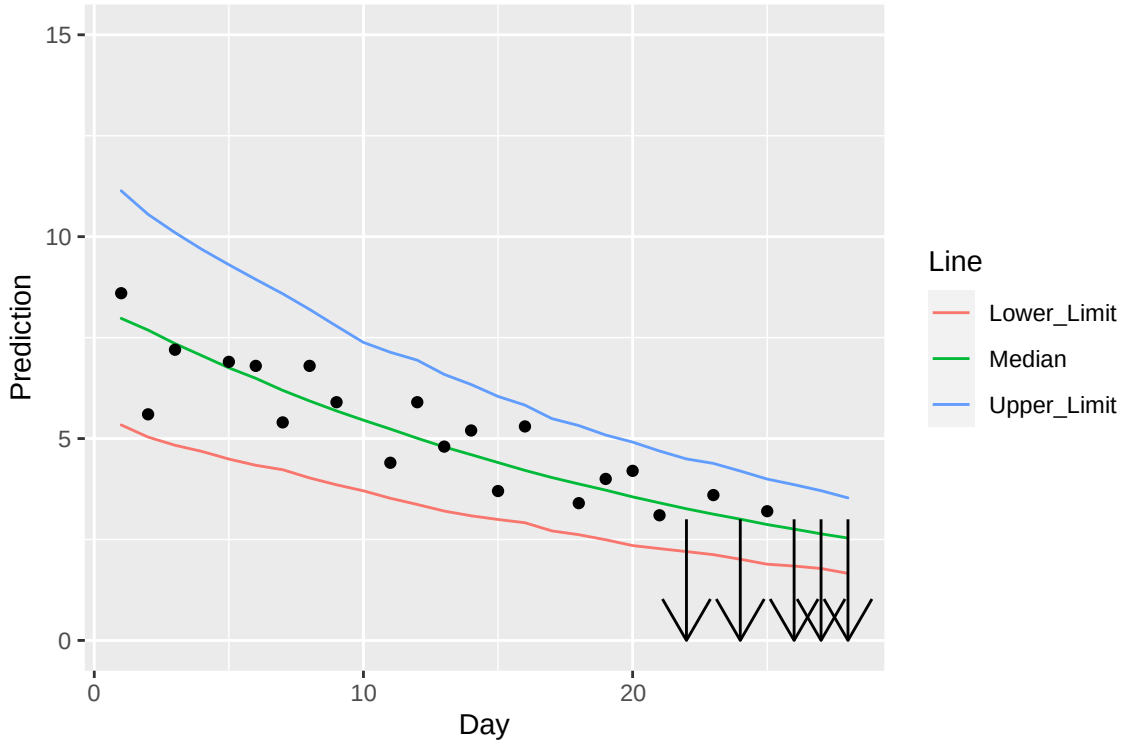
Reasonable probability based on predictions (high) [2]

1.7) Explain why it would be wrong to extrapolate your fit to the left (Day -1, -2, etc.) specifically. **[2]**

The data while taking the treatment is not representative of the time before the treatment. [2]

1.8) For up to 2 bonus marks, illustrate the censored observations on your final plot in a sensible way.

```r
ncens <- sum(sugar$BDL)
arrow_data <- data.frame(Day = sugar$Day[sugar$BDL == 1], y0 = rep(3, ncens), y1 = rep(0, ncens))
plot_data |>
  ggplot(aes(x = Day, y = Prediction)) + ylim(0, 15) +
  geom_line(aes(colour = Line)) +
  geom_point(data = sugar, mapping = aes(x = Day, y = Values)) +
  geom_segment(data = arrow_data, mapping = aes(x = Day, y = y0, xend = Day, yend = y1), arrow = arrow())
```



Points total

The points on the test add up to **50**