STSB6816 Special Test 2024

Mathematical Statistics and Actuarial Science; University of the Free State

2024/06/06

## Time: 170 minutes; Marks: 45

# MEMORANDUM

# Instructions

* Answer all questions in a single R Markdown document. Please knit to PDF or Word 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 about 3 decimal places.
* Show all derivations, formulas, code, sources, and reasoning.
* Intervals should cover 95% probability unless stated otherwise.
* No communication software, devices, or 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.
* Use of AI tools must be disclosed and summarised.

# Introduction

The data is 0, 0, 1, 0, 1, 3, 0, 2, 1, 2. It is a time series of annual claim numbers for a small insurer.

They wish to model the data as a count process with different underlying forms. You are requested to fit Negative Binomial (Type II) regressions (with log link functions).

# Question 1

**1.1)** You should fit the following forms and in each case plot the resulting density of the time dependence coefficient :

1. a linear regression on time :
2. an AR(1) process, where
3. an underlying moving average of form MA(1):

They suggest a prior on the time dependence coefficients. Additionally, you are requested to use the prior for any scale or precision parameters. For the AR(1) form, assume that . Lastly, for the MA(1) model only, let . **[26]**

*[Hint: fit the first two forms to start, as that is sufficient to answer the rest of the questions (also, they can be fit with a single model specification). The MA(1) is more challenging and should only be attempted once the rest of the test is completed.]*

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

data {  
 int n;  
 int y[n];  
 real x[n];  
}  
parameters {  
 real alpha;  
 real<lower=0, upper=0.9> beta;  
 real<lower=0> phi;  
}  
transformed parameters {  
 real mu[n];  
 for (i in 1:n) {  
 mu[i] = exp(x[i]\*beta + alpha);  
 }  
}  
model {  
 for (i in 1:n) {  
 y[i] ~ neg\_binomial\_2(mu[i], phi);  
 }  
 phi ~ exponential(0.01);  
}  
generated quantities {  
 vector[n] log\_lik;  
 for (i in 1:n) {  
 log\_lik[i] = neg\_binomial\_2\_lpmf(y[i] | mu[i], phi);  
 }  
}

data {  
 int n;  
 int y[n];  
}  
parameters {  
 real alpha;  
 real<lower=0, upper=0.9> beta;  
 real<lower=0> phi;  
 real eta0;  
}  
transformed parameters {  
 real eta[n];  
 eta[1] = eta0\*beta + alpha;  
 for (i in 2:n) {  
 eta[i] = eta[i-1]\*beta + alpha;  
 }  
}  
model {  
 for (i in 1:n) {  
 y[i] ~ neg\_binomial\_2(exp(eta[i]), phi);  
 }  
 phi ~ exponential(0.01);  
 eta0 ~ normal(-3, 1);  
}  
generated quantities {  
 vector[n] log\_lik;  
 for (i in 1:n) {  
 log\_lik[i] = neg\_binomial\_2\_lpmf(y[i] | exp(eta[i]), phi);  
 }  
}

y <- c(0, 0, 1, 0, 1, 3, 0, 2, 1, 2)  
n <- length(y)  
y0 <- c(0, y[seq\_len(n-1)])  
x <- seq\_len(n)

fits <- list(  
 Form1 = negbin2model |> sampling(  
 list(n = n, y = y, x = x), chains = mycores, iter = 4000),  
 Form2 = negbin2model |> sampling(  
 list(n = n, y = y, x = y0), chains = mycores, iter = 4000),   
 Form3 = negbin2modelMA1 |> sampling(  
 list(n = n, y = y), chains = mycores, iter = 4000)  
)

fits$Form1 |> traceplot(pars = c("alpha", "beta", "phi"))



post\_sims <- fits$Form1 |> rstan::extract(pars = c("alpha", "beta", "phi"))  
post\_sims$beta |> density() |>   
 plot(main = "Form 1", xlab = "Time dependence coefficient", col = 'purple', lwd = 3)



fits$Form2 |> traceplot(pars = c("alpha", "beta", "phi"))



post\_sims <- fits$Form2 |> rstan::extract(pars = c("alpha", "beta", "phi"))  
post\_sims$beta |> density() |>   
 plot(main = "Form 2", xlab = "Time dependence coefficient", col = 'purple', lwd = 3)



fits$Form3 |> traceplot(pars = c("alpha", "beta", "phi"))



post\_sims <- fits$Form3 |> rstan::extract(pars = c("alpha", "beta", "phi"))  
post\_sims$beta |> density() |>   
 plot(main = "Form 3", xlab = "Time dependence coefficient", col = 'purple', lwd = 3)



###### Form 1 [8]. Form 2 [8]. Form 3 [10].

**1.2)** Compare the forms to determine which best fits the data. **[7]**

library(loo)  
fits |> lapply(\(fit) {extract\_log\_lik(fit, merge\_chains = FALSE)}) -> log\_lik  
log\_lik |> lapply(\(ll) {relative\_eff(exp(ll), cores = mycores)}) -> r\_eff  
fits |> length() |> seq\_len() |>   
 lapply(\(i) {loo(log\_lik[[i]], r\_eff = r\_eff[[i]], cores = mycores)}) |>   
 loo\_compare() -> comparison  
model\_order <- order(rownames(comparison))  
rownames(comparison) <- names(fits)[model\_order]  
comparison |> knitr::kable(digits = 2)

|  | elpd\_diff | se\_diff | elpd\_loo | se\_elpd\_loo | p\_loo | se\_p\_loo | looic | se\_looic |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Form2 | 0.00 | 0.00 | -12.96 | 2.20 | 1.13 | 0.36 | 25.92 | 4.40 |
| Form3 | -0.52 | 0.98 | -13.48 | 2.95 | 1.48 | 0.60 | 26.96 | 5.90 |
| Form1 | -2.42 | 0.83 | -15.38 | 2.37 | 1.79 | 0.69 | 30.76 | 4.75 |

###### Comparison statistics [4]. Comparison conclusion [3].

**For the AR(1) model:**

**1.3)** Illustrate the posterior predictive distributions for and taking the full parameter uncertainty into account (say using a bar plot). What is the probability that the company fails at Time 11 or Time 12 given that more than 5 claims in a month will cause them to fail? **[9]**

post\_sims <- fits$Form2 |> rstan::extract(pars = c("alpha", "beta", "phi"))  
n\_sims <- length(post\_sims$alpha)  
mu11 <- exp(y[10]\*post\_sims$beta + post\_sims$alpha)  
y11\_pred <- rnbinom(n\_sims, mu = mu11, size = post\_sims$phi)  
y11\_pred |> table() |> barplot()



cat('\nThe probability of more than 5 claims at time 11 is', round(mean(y11\_pred > 5),3), '\n')

|   
| The probability of more than 5 claims at time 11 is 0.007

mu12 <- exp(y11\_pred\*post\_sims$beta + post\_sims$alpha)  
y12\_pred <- rnbinom(n\_sims, mu = mu12, size = post\_sims$phi)  
y12\_pred |> table() |> barplot()



cat('\nThe probability of more than 5 claims at time 12 is', round(mean(y12\_pred > 5),3), '\n')

|   
| The probability of more than 5 claims at time 12 is 0.021

cat('\nThe probability of more than 5 claims at either time 11 or time 12 is',   
 round(mean((y11\_pred > 5) | (y12\_pred > 5)),3), '\n')

|   
| The probability of more than 5 claims at either time 11 or time 12 is 0.024

###### Using the full simulations for alpha and beta at each step [2]. Simulating new random negative binomial values correctly [2]. Using the full simulations of y11 to predict y12, with parameters lined up [2]. Bar plots [1]. Probabilities [2].

**1.4)** Suppose that for the last observation (Year 10) there may be claims incurred but not reported (IBNR), so the number is at least 2 (censored). Explain, in detail, everything you would change in order to incorporate this into your model. **[3]**

*[NB: Do not actually fit the model with censoring, the simulations will slow down too much for a test environment.]*

###### Add a censoring indicator [1]. Add a term for the (log) survival function (right censoring) [1]. Add an if or a weighting to have the model correctly select the pmf versus the survival function [1].

data {  
 int n;  
 int y[n];  
 real x[n];  
 real cens[n];  
}  
parameters {  
 real alpha;  
 real<lower=0, upper=0.9> beta;  
 real<lower=0> phi;  
}  
transformed parameters {  
 real mu[n];  
 for (i in 1:n) {  
 mu[i] = exp(x[i]\*beta + alpha);  
 }  
}  
model {  
 for (i in 1:n) {  
 target += neg\_binomial\_2\_lpmf(y[i] | mu[i], phi)\*(1.0-cens[i]) +   
 neg\_binomial\_2\_lccdf(y[i] | mu[i], phi)\*cens[i];  
 }  
 phi ~ exponential(0.01);  
}  
generated quantities {  
 vector[n] log\_lik;  
 for (i in 1:n) {  
 log\_lik[i] = neg\_binomial\_2\_lpmf(y[i] | mu[i], phi)\*(1.0-cens[i]) +   
 neg\_binomial\_2\_lccdf(y[i] | mu[i], phi)\*cens[i];  
 }  
}

cens <- c(rep(0, n - 1), 1)

fits <- list(  
 Form1 = negbin2modelcens |> sampling(  
 list(n = n, y = y2, x = x, cens = cens), chains = mycores, iter = 4000),  
 Form2 = negbin2modelcens |> sampling(  
 list(n = n, y = y2, x = y1, cens = cens), chains = mycores, iter = 4000)  
)

## Points total

The points on the test add up to **45**