MATH392: Mathematical Statistics

Final Exam

May 7, 2018

Instructions

1. You must show your work/thinking to receive full credit.
2. You are permitted to consult the following exhaustive list of resources: your own class notes, your previous homeworks, in-R help files, slides from our lectures, and the textbook. If you do utilize a direct result from one of those resources, please cite it. All other resources - the internet, other people - are not permitted. Be mindful of the honor code. If you struggle with R code, you are welcome to inquire with me over slack.
3. You're encouraged to write your answers directly into this document by writing your answers into the spots right after the three asterisks.
4. You can take up to 24 hrs to work on this exam but it must be submitted to gradescope by 4 pm PST May 14. Please be sure to select the pages that your questions show up on to make grading easier for me.
5. Be sure to knit your pdf and be sure you're proud of how the final document looks. Clean up your code and plots and resize them if necessary.

Question 1: What time did you start the exam? What time did you end?

Answer: I started at 6pm on Wednesday and finished at 4pm on Thursday.

Poisson Regression

Poisson regression is a form of Generalized Linear Model often used to model count data, Y, as a function of a set of predictors X (an n by $p+1$ matrix). The conditional distribution of the response is $Y \mid X \sim$ Poisson($\lambda = e^{X\beta}$). For reference, if Y is Poisson, $P(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$ for $\lambda > 0$ and y on the non-negative integers.

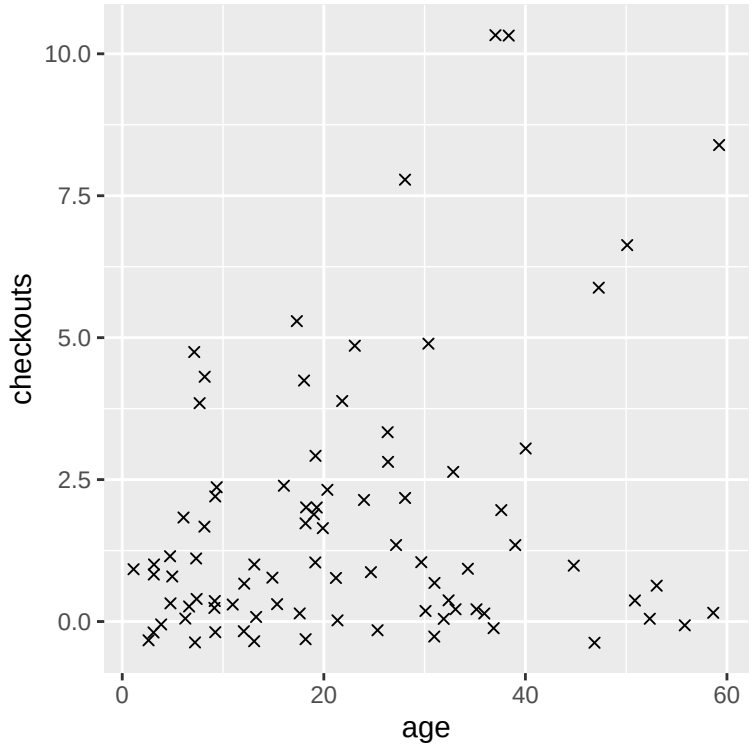
Question 2: *GLM.* Every generalized linear model consists of three components: a distribution for the response, the linear predictor, and a link function to map the linear predictor to the expected value of the response. Please identify these components in the case of Poisson regression.

Answer: In the case of Poisson regression, our conditional response variable is $Y|X \sim \text{Pois}(\lambda = e^{X\beta})$, which has a linear predictor $X\beta$, and link function as the exponential, $g(x) = e^x$. (Equivualently, it can be said that the link function is log, where $\log(\lambda) = X\beta$)

An example Every semester in Math 141, students collect a sample of data from the thesis tower that records the age of the thesis and the number of times that it has been checked out. You can read that data in with the following code.

```
theses <- read.csv("https://www.dropbox.com/s/88x6jjv5ehekonk/theses.csv?dl=1")
```

Question 3: *Plotting the data.* Construct a scatter plot of the relationship between these two variables (I recommend `geom_jitter()`).



I. Frequentist Model

Question 4: *Parameter estimation.* For our first pass, let's fit this model (i.e. estimate β_0 and β_1) using maximum likelihood. Start with an analytical approach to this estimation. If you can't find a closed-form solution, turn to a numerical solution by writing the likelihood (or log-liklihood) as a function to be optimized by the `maxLik()` function in the `maxLik` library. Check your solutions against the built-in function for estimating Poisson regression.

Answer: For this problem, we couldn't find a closed form solution, so we are left to derive the log-likelihood for our response variable, $Y|X$. In order to do that, we utitilize the Poisson density $f(k|\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$.

$$
\begin{aligned}
f(\mathbf{Y}|\lambda) &= \prod_{i=1}^{n} f(Y_i|\lambda) \\
\log f(\mathbf{Y}|\lambda) &= \log\left[\prod_{i=1}^{n} f(Y_i|\lambda)\right] \\
&= \sum_{i=1}^{n} \log f(Y_i|\lambda) \\
&= \sum_{i=1}^{n} \log\left(\frac{e^{-\lambda}\lambda^{y_i}}{y_i!}\right) \\
&= \sum_{i=1}^{n} -\lambda + \log\lambda^{y_i} - \log(y_i!) \\
&= \boxed{\sum_{i=1}^{n} -\lambda + y_i\log\lambda - \log(y_i!)}
\end{aligned}
$$

```
pois_likelihood <- function(B, X, Y){
  lambdas <- exp(X %*% B)
  sum(-lambdas + Y*log(lambdas) - log(factorial(Y)))
}
```

```
#Prepare data
X <- cbind(1, theses$age)
Y <- theses$checkouts
# Computational solution using R
fit <- glm(checkouts ~ age, data = theses, family = "poisson")
B_fit <- fit$coefficients
# Analytical solution, using maxLik and manual likelihood fxn
B_mle <- maxLik(pois_likelihood, start = c(0,0), X = X, Y = Y)$estimate
```

```
## [1] "Using glm in R:"
```

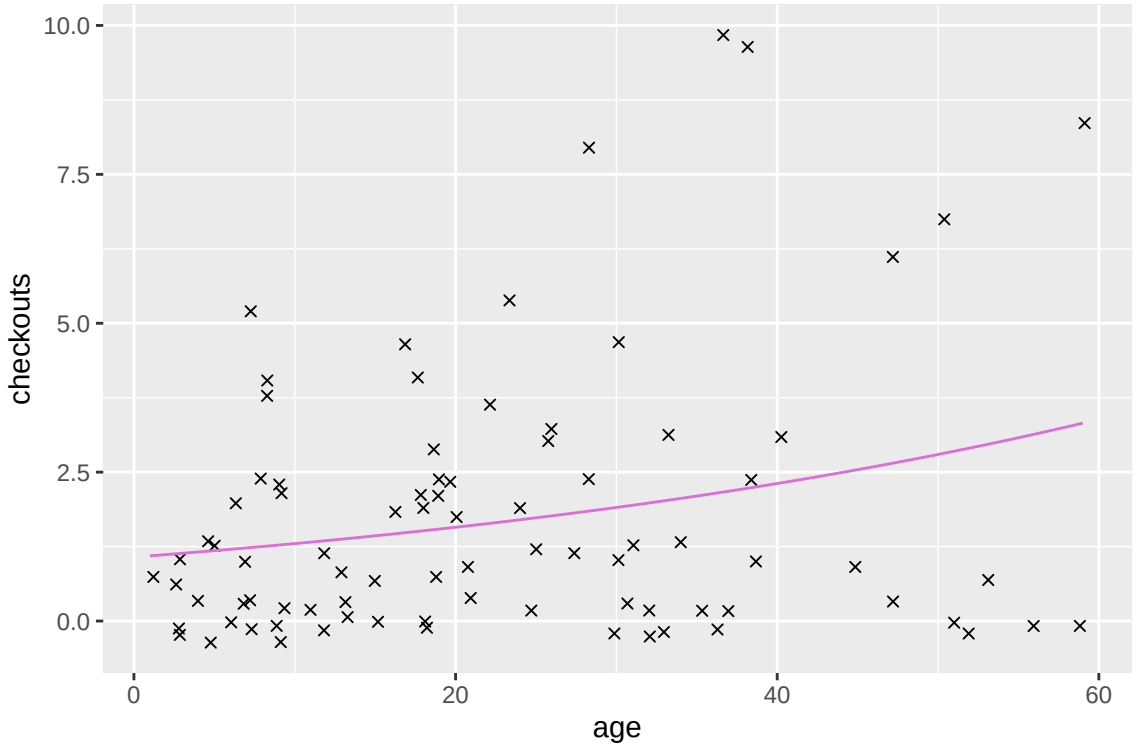
```
## (Intercept)         age
##  0.07121037  0.01914517
```

```
## [1] "Using manual likelihood function:"
```

```
## [1] 0.07121039 0.01914517
```

Question 5: *Plotting the model.* Modify the plot above to include a line showing the value of your regression function, $\hat{E}(Y|X)$. You can do this by writing this as an R function and plugging it into `stat_function()`.

```
pois_by_X1 <- function(X1){
  .lambda_hats <- exp(cbind(1, X1) %*% B_mle)
  .lambda_hats
}
if(plotbool){checkouts_plot +
    stat_function(data = data.frame(theses$age), mapping = aes(x=theses$age),
                  fun = pois_by_X1, col = "orchid")}
```



We can see that checkouts increase with the age of a thesis (which sure would make sense) but that the effect is slight.

Question 6: *Theoretical CI.* Construct a 95% confidence interval for β_1 using the standard error estimate that appears in the `summary()` of the model object that resulted from your call to `glm()`. What is the theoretical justification of using this interval? Summarize the mathematical argument in 4 sentences or less.

```
confint(fit)
```

```
##                    2.5 %     97.5 %
## (Intercept) -0.255698853 0.38001222
## age          0.009026984 0.02908795
```

Answer: The results above say that given $\alpha = 0.05$, we have the following confidence intervals:

$$\beta_0 \in [-0.26, 0.38] \quad | \quad \beta_1 \in [0.01, 0.03]$$

We can use this interval since it represents the values of β_1 which give us the interval with which we can sample a mean rate, and expect it to be included in the interval given $\mathcal{L}^2$-loss 95% of the time. Since $\mathcal{L}^2$-loss is reliable and tends to be able to explain most data, this confidence interval tells us, if we were to sample independently of β_0, then sampling from this interval, under the paradigm of Poisson regression and a link function $e^{X\beta}$ will tend to explain our data optimally given $\mathcal{L}^2$-loss.

Question 7: *Bootstrap CI.* Form an alternative CI for β_1 using a bootstrap method of your choice. How do the two intervals compare? Are you convinced that this parameter is positive?

Answer: Plotted below is the density of our bootstrapped values for β_1. As we can see in both the quantiles and the plot, the density's left tail "seeps" into the negative reigon. A negative value for β_1 would defy our intuition, since that would negatively correlate checkouts with age, which we do not antipcate being true.

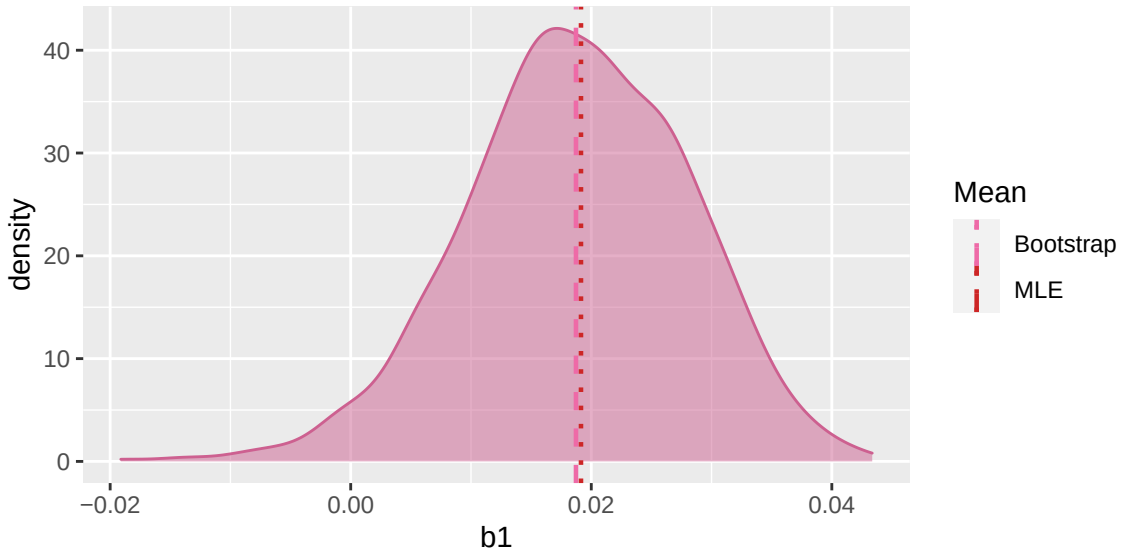
Overall, compared to the non-bootstrap quantiles, the bootstrap quantiles are certainly wider and less sharp. This can be explained by the fact that although our sample size is reasonable, it still has size $n = 85$. There are a lot of zeros (since this is Poisson data) so bootstrapping and getting values that state that β_1 may be negative only reflects that there are a good amount of samples that have a high age but no or little checkouts, which is definitely true. On the other hand, the higher value of the 95% quantile reflects that there are a good amount of samples that have a low age but still a good amount of checkouts (an interesting thesis, perhaps).

```
bootstrap <- function(B){
  fit_samples <- rep(NA, B)
  for(i in 1:B){
    sample <- sample_frac(tbl = theses, replace = TRUE, size = 1)
    fit_samples[i] <- glm(checkouts ~ age, data = sample, family = "poisson")$coef[2]
  }
  fit_samples
}
sample_b1s <- bootstrap(B = 1000)
mean(sample_b1s)
```

```
## [1] 0.01872704
```

```
bootstrapped_b1s <- data.frame(b1 = sample_b1s)
quantile(bootstrapped_b1s$b1, probs = c(0.025,0.975))
```

```
##          2.5%         97.5%
## -0.0002980887  0.0352900879
```



II. Penalized Model

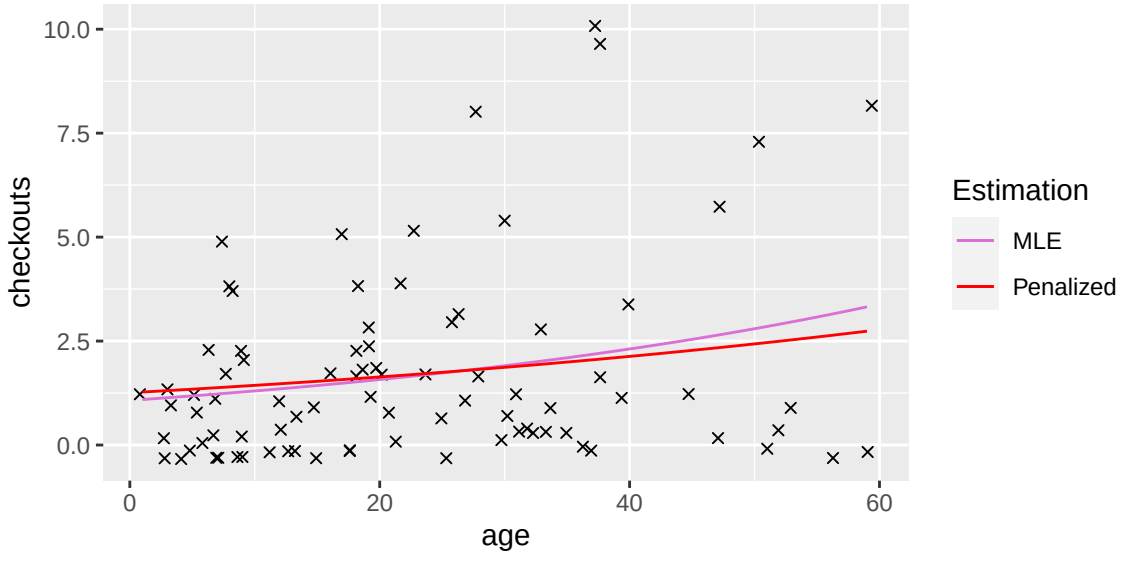
Question 8: *Penalized vs. unpenalized.* In what scenario is it useful to add a penalty term to your loss/likelihood function (a procedure alternatively called *regularization* or *shrinkage*)? In general, how do the properties of an estimate based on maximum likelihood compare to an estimate based on penalized maximum likelihood?

Answer: Shrinkage is often useful in settings where there is concern for overfitting. However, because of both the small amount of data and the lack of a clear pattern to overfit to in the data (it is very hard to see one, at least), this is not one of those settings. The regularized estimator is similar in the sense that it inherits many properties of the MLE, like consistently converging to nearly the same values. One difference, however, is that while it does inherit things from the MLE estimator (since it is maximizing some likelihood, at the end of the day), it does outdo the MLE in that there is not necessarily a unique MLE, but for certain λ, there may exist a unique regularized estimator. However, is the fact that penalized loss can lead to a unique estimator, unlike the MLE. However, in this case it doesn't really matter since β_1 is the more weighed variable, and the MLE seems to be marginally unique in β_1.

Admittedly, we're not actually in that scenario in our example, but let's proceed anyhow to see how the penalized model compares.

Question 9: *Plotting two models.* Modify your likelihood function to include a penalty term and re-run the optimization get an new estimate of β_1. Plot this new model alongside the existing unpenalized model on a scatterplot of the data.

```
.lambda <- 100
penalized_likelihood <- function(B, X, Y){
  penalty <- .lambda * (B[2])^2
  lambdas <- exp(X %*% B)
  sum(-lambdas + Y*log(lambdas) - log(factorial(Y)) - penalty)
}
```



III. Bayesian Model

Question 10: *Formulating priors.* Next, consider modeling this data to include prior knowledge about the values of the parameters, β_0 and β_1. There is no conjugate prior for Poisson regression, so you're free to formulate your own priors. Please write down in probability notation a prior distribution of your choosing that is moderately informative.

Some advice:

- The scale of the parameters is challenging to think about due to the link function, so be thoughtful here. You can always experiment with running particular values of the parameters through the function for various values of x and check that the corresponding value of $\hat{E}(Y|X)$ is sensible.
- Be deliberate in your decision to think about these prior distributions as independent of one another or as a draw from a bivariate distribution with non-zero covariace.

Answer: Consider the following priors:

$$\beta_0 \sim \Gamma(a_0 = 0.05, b_0 = 3) \quad | \quad \beta_1 \sim \Gamma(a_1 = 0.10, b_1 = 2)$$

```
# set priors
a0 <- 0.05
b0 <- 3
a1 <- 0.10
b1 <- 2
```

We choose the gamma distribution as priors, so $\beta_j \sim \Gamma(a_j, b_j)$. The gamma distribution provides moderate information and satisfy the domain of the parameter we are trying to estimate, which is a Poisson rate ($\lambda \in \mathbb{R}^+$). This is only the case because surely, the link function g will map our linear predictor $X\beta \mapsto \exp(X\beta) \in \mathbb{R}^+$.

Question 11: *Writing down the posterior distribution.* Write out the form of the posterior distribution up to the constant of proportionality (`\propto` in latex is handy here), first as the product of the likelihood and the prior, then with the corresponding densities substituted in. No need to simplify from there - we'll be using the Metropolis algorithm.

Construct three corresponding R functions, one for the prior, one for the likelihood, and one for the posterior. You're encouraged to follow the form used in the slides and your last problem set.

Answer: Let our parameter vector be denoted by $\vec{\beta} = (\beta_0, \beta_1)$. To avoid notational overload, let our prior distributions be denoted $\beta_j \sim \Gamma(a_j, b_j)$. Although we compactly use the notation $\vec{\beta} = (\beta_0, \beta_1)$, this does **not** imply that we are sampling $\vec{\beta}$ from a bivariate gamma distribution. However, because we are assuming conditional independence of the parameters, this should have no effect on the posterior (the joint density would simply be a product of both densities, by independence). So it is only in writing that there is a difference between $f(\vec{\beta})$ and $f(\beta_0, \beta_1)$, since they both equal $f(\beta_0) \cdot f(\beta_1)$, by conditional independence. All that said, we now derive the joint posterior density of our parameters as follows:

$$\begin{aligned} f(\vec{\beta}|X,Y) &= \frac{f(Y|\vec{\beta}) \cdot f(\vec{\beta})}{f(X,Y)} \\ &\propto f(Y|\vec{\beta}) \cdot f(\vec{\beta}) \\ &\propto \prod_{i=1}^{n} f(Y_i|\lambda = e^{X\vec{\beta}}) \cdot \prod_{j=0}^{p} f(\beta_j) \\ &\propto \prod_{i=1}^{n} \left(\frac{e^{-\lambda}\lambda^{y_i}}{y_i!} \right) \cdot \prod_{j=0}^{p} \left(\frac{b_j^{a_j}}{\Gamma(a_j)} \cdot {\beta_j}^{a_j - 1} e^{-b_j \beta_j} \right) \end{aligned}$$

And below we define our three R functions. The prior yields the sum of the log densities of our prior distributions, $\beta_j \sim \Gamma(a_j, b_j)$, and the likelihood is defined just like previously.

```
# prior function
prior <- function(theta){
  B0 <- theta[1]
  B1 <- theta[2]
  #uses gamma priors
  B0_prior <- dgamma(B0, a0, b0, log = T)
  B1_prior <- dgamma(B1, a1, b1, log = T)
  sum(B0_prior,B1_prior) # return sum of log-densities
}
# likelihood function
likelihood <- function(theta){
  B0 <- theta[1]
  B1 <- theta[2]
  .B <- c(B0,B1)
  lambdas <- exp(X %*% .B) # link function
  sum(-lambdas + Y*log(lambdas) - log(factorial(Y))) # return log-likelihood
}
# posterior function
posterior <- function(theta){
  sum(prior(theta),likelihood(theta)) # return log-likelihood of prior * likelihood
  }
```

Question 12: *MCMC via Metropolis.* Implement the Metropolis algorithm to draw at least 10,000 samples from the posterior distribution. Also address the following questions:

1. What was your proposal distribution?
2. What was your acceptance rate (tune it to be in the .4 range)?
3. What was your burn-in period?
4. Does it appear that the Markov chain has converged?

Answer: We can see below that our Markov Chain indeed appears to converge, with an acceptance rate of 36.3% and a burn-in period of 5000. This is given the proposal distributions:

$$\beta_0^* \sim \mathcal{N}(\beta_0, 0.01) \quad | \quad \beta_1^* \sim \mathcal{N}(\beta_1, 0.0015)$$

```
set.seed(92)
it <- 50000
chain <- matrix(rep(NA, (it+1) * 2), ncol = 2)
colnames(chain) <- c("beta0","beta1")
theta_0 <- c(0.25,0.5) #initialize theta_0
chain[1, ] <- theta_0
for (i in 1:it){
  b0_prop <- rnorm(1, chain[i,1], 0.01)
  b1_prop <- rnorm(1, chain[i,2], 0.0015)
  proposal <- c(b0_prop,b1_prop) #propose theta_star
  p_move <- exp(posterior(proposal) - posterior(chain[i,1:2])) #obtain ratio
  if (runif(1) < p_move) {chain[i+1, 1:2] <- proposal}
  else {chain[i+1, 1:2] <- chain[i, 1:2]}
}
head(chain)
```

```
##          beta0     beta1
## [1,] 0.2500000 0.5000000
## [2,] 0.2352122 0.4992537
## [3,] 0.2352122 0.4992537
## [4,] 0.2352122 0.4992537
## [5,] 0.2322289 0.4980173
## [6,] 0.2269678 0.4977119
```

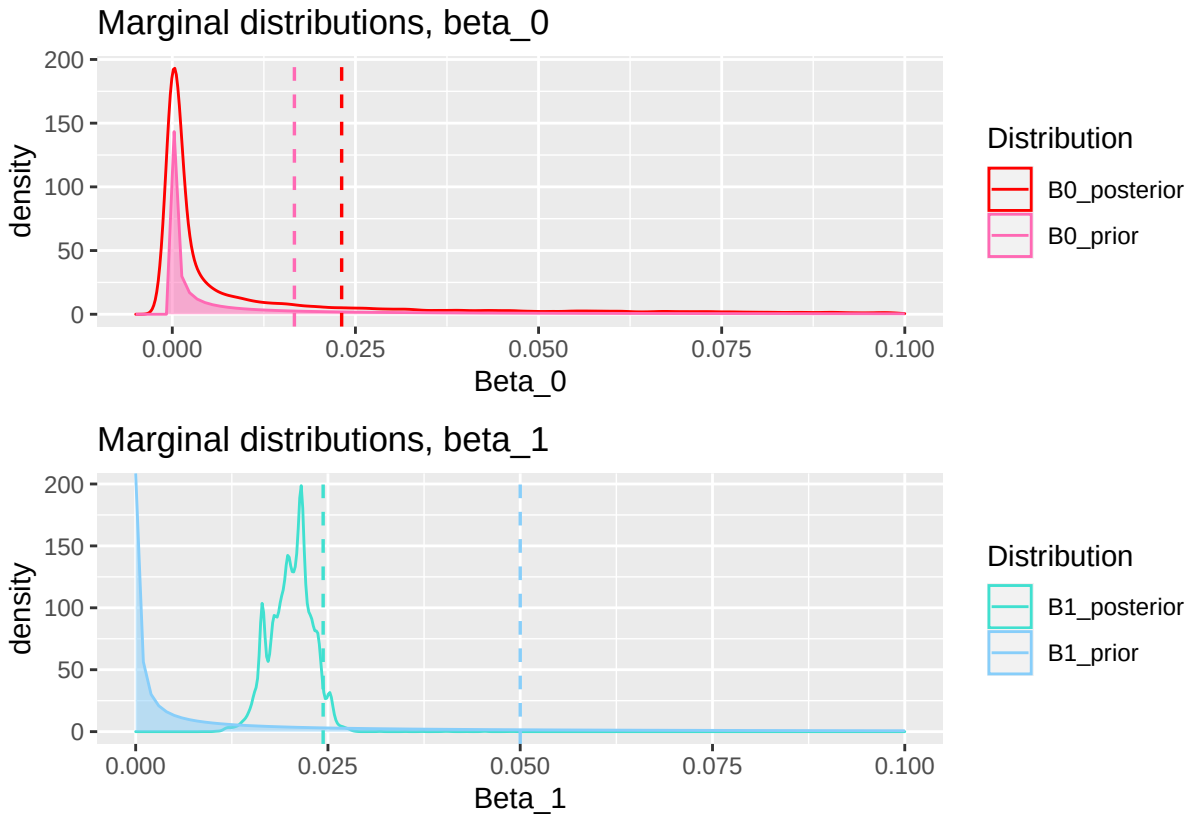
```
tail(chain)
```

```
##                 beta0      beta1
## [49996,] 1.185073e-06 0.02143426
## [49997,] 1.185073e-06 0.02143426
## [49998,] 1.185073e-06 0.02143426
## [49999,] 1.185073e-06 0.02143426
## [50000,] 1.185073e-06 0.02143426
## [50001,] 1.185073e-06 0.02143426
```

```
burn_in <- 5000
acceptance <- 1 - mean(duplicated(chain[-(1:burn_in),]))
paste("Acceptance Rate: ",acceptance)
```

```
## [1] "Acceptance Rate:  0.363703028821582"
```

Question 13: *Visualizing the posterior*. Use the samples from both MCMC procedures to construct plots of the joint and marginal posterior distributions as well as the joint and marginal prior distributions. How did the data update our knowledge of these parameters?



Answer: Recall the we used the prior distributions:

$$\beta_0 \sim \Gamma(a_0 = 0.05, b_0 = 3) \quad | \quad \beta_1 \sim \Gamma(a_1 = 0.10, b_1 = 2)$$

Since our priors are gamma-distributed, their expectation is easy to compute. We have $\mathrm{E}(\beta_j) = \frac{a_j}{b_j}$, which result in the corrosponding **prior** means (which we have plotted):

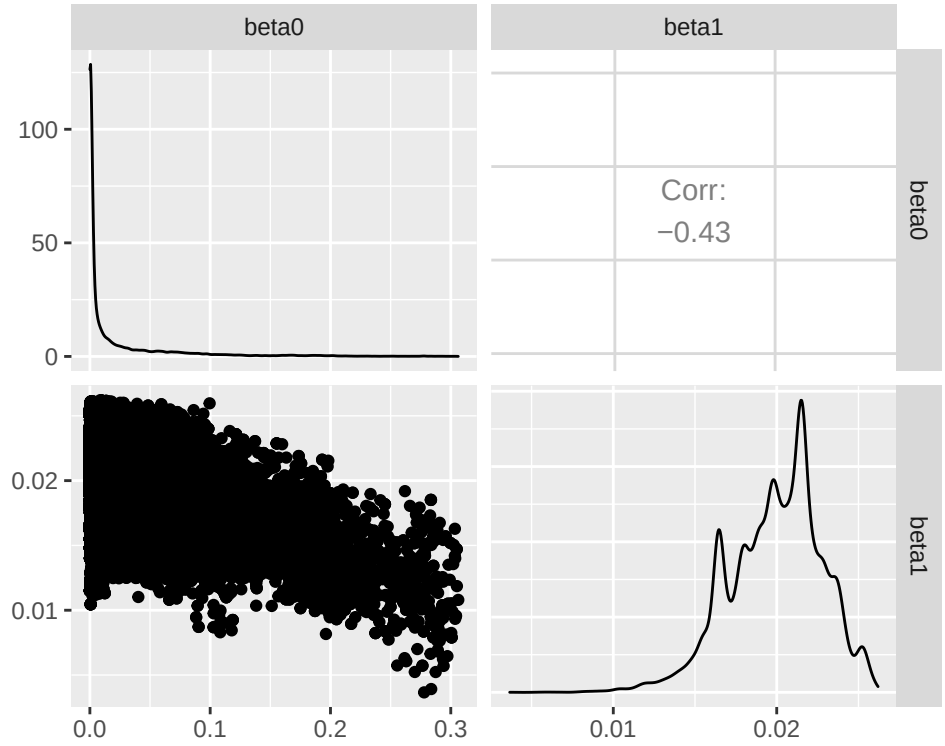
$$\mathrm{E}(\beta_0) \approx 0.0167 \quad | \quad \mathrm{E}(\beta_1) = 0.050$$

Now, compare these with the analytical **posterior** means, which we find to be:

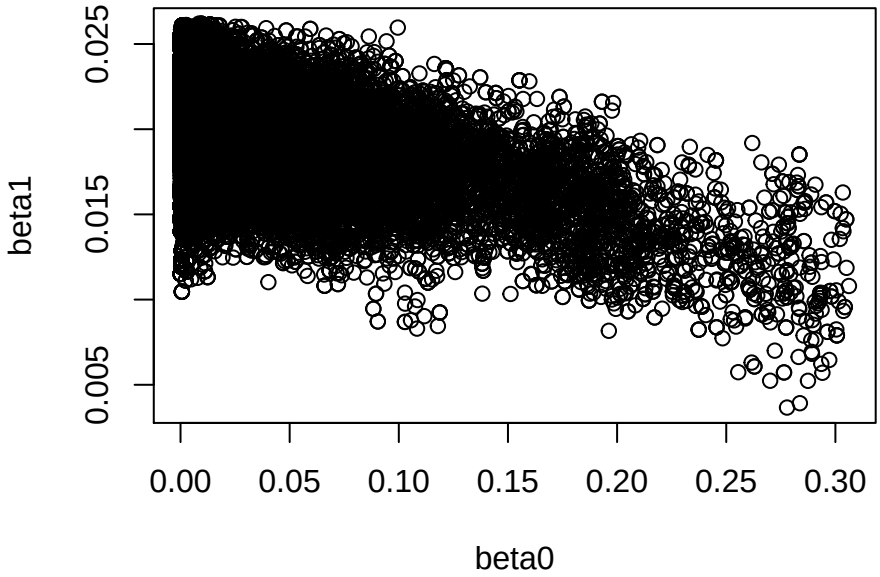
$$\hat{\mathrm{E}}(\beta_0) \approx 0.022 \quad | \quad \hat{\mathrm{E}}(\beta_1) \approx 0.024$$

Therefore, we can summarize our updated knowledge by looking at the prior mean and the updated posterior mean. In our case, the prior mean was $\vec{\beta} = (0.0167, 0.050)$ and our posterior mean was $\vec{\beta} = (0.022, 0.024)$. It is good news that our posterior mean has made a *large* stride towards $\hat{\beta}_{MLE} = (0.071, 0.191)$ if not in both componentns, then at least in the β_1 component!

Joint posterior distribution



A closer look



Joint prior distribution

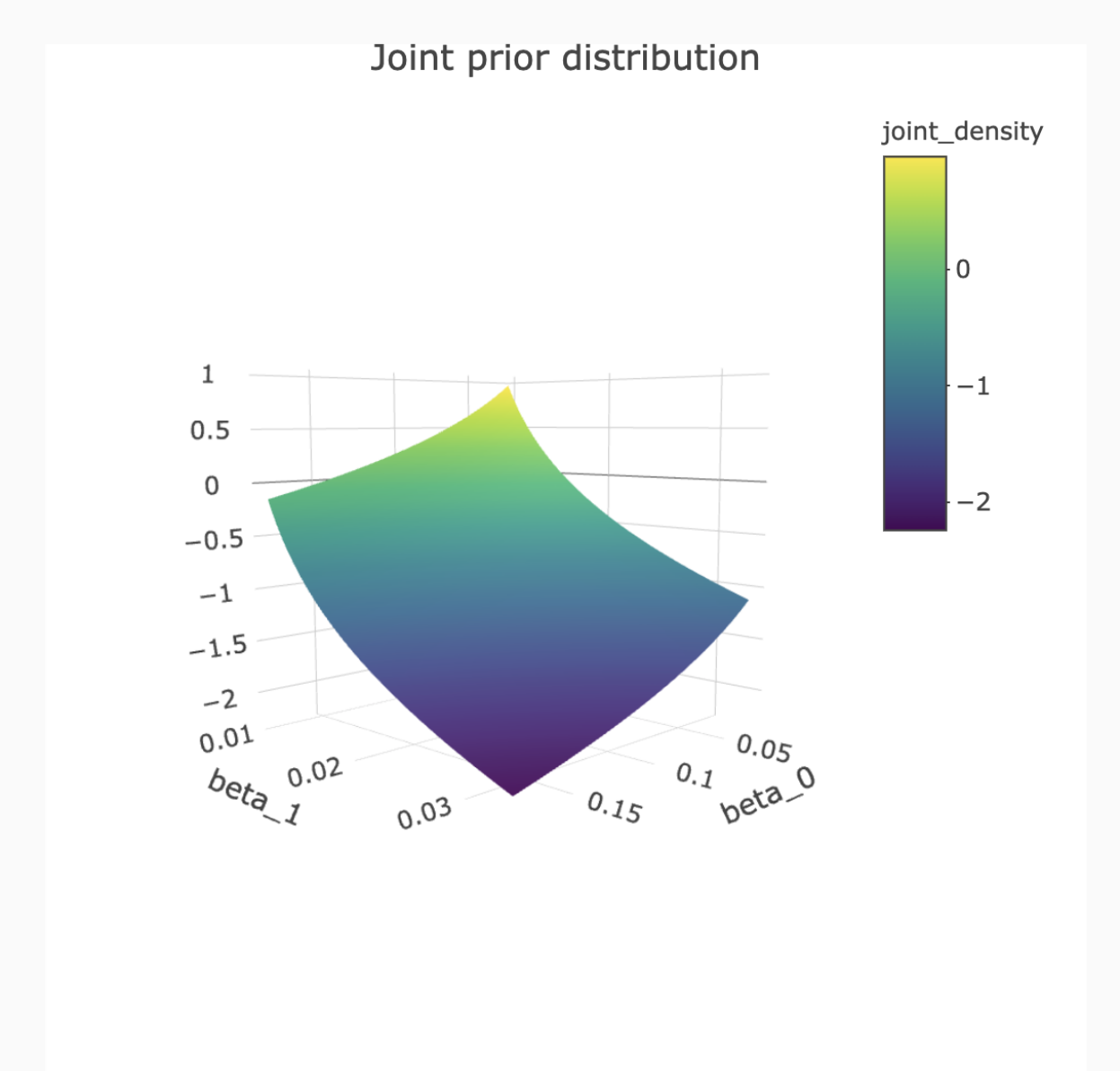


Figure 1: Plot

Question 14: *Bayesian Intervals.* Form a 95% credible interval for β_1 by taking the 2.5% and 97.5% quantiles from the posterior distribution. How does this compare to the frequentist confidence intervals? Why do you think this is?

```
## [1] "Credible interval: beta_0"
```

```
##         2.5%        97.5%
## 4.890327e-07 1.768768e-01
```

```
## [1] "Credible interval: beta_1"
```

```
##       2.5%      97.5%
## 0.01381391 0.02599991
```

Answer: The results above say that given $\alpha = 0.05$, we have the following confidence intervals for $\vec{\beta}$:

$$\beta_0 \in [0.00, 0.177] \quad | \quad \beta_1 \in [0.014, 0.026]$$

Recall, the frequentist CI gave:

$$\beta_0 \in [-0.26, 0.38] \quad | \quad \beta_1 \in [0.01, 0.03]$$

In our case, the Bayesian CI was very clearly the narrower range. The reason majorly falls in the design of these intervals. The Bayesian credible intervals incorporate specific contextual information from some informative prior distribution, which gives it an upper-hand over the frequentist confidence interval, where intervals are based only on the data. So, the Bayesian CI has the advantage of being able to disregard worse fits and incorporate new information recursively, which makes its distribution much sharper/more information than the frequentist CI, which is strictly a mapping from the raw data.

Question 15: *Plotting all three models.* Add the Bayesian model to your existing plot featuring the frequentist and penalized models. To get Bayesian point estimates you will need to pull a sensible value from the posterior distributions.

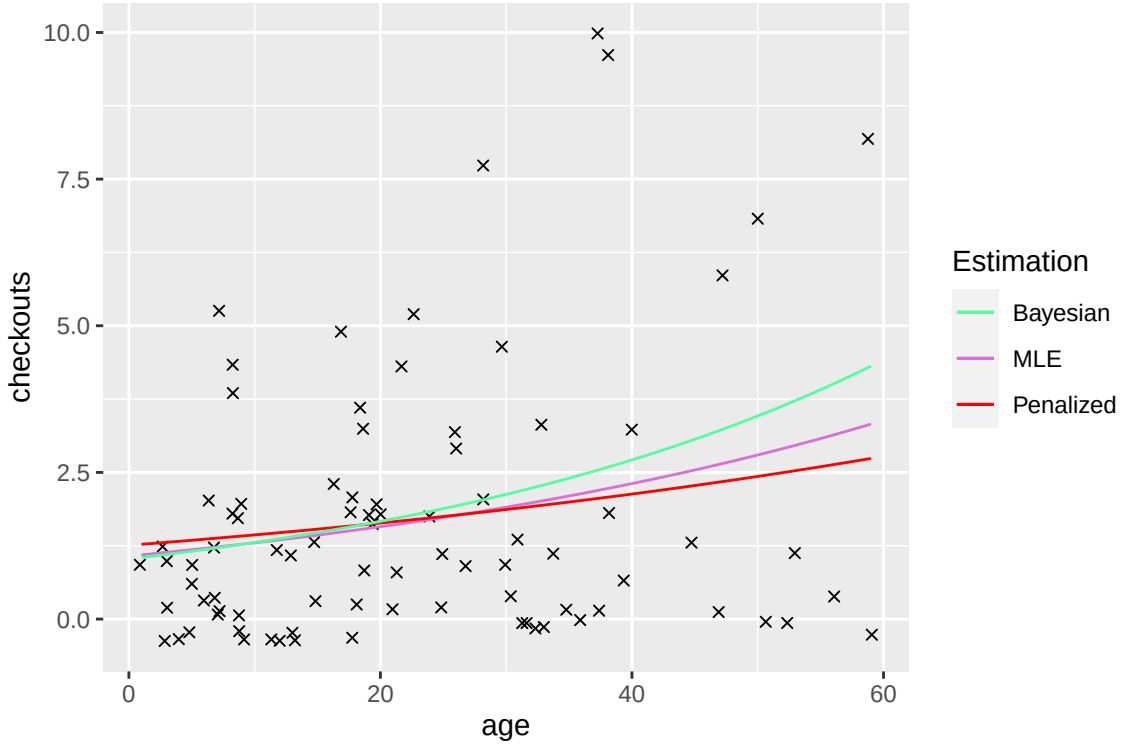
Are these models fairly similar or quite different? Please explain why.

Answer: Although the models all seem to converge towards some range of the parameter, there are slight variations to be noted. In the instance of the Bayesian Metropolis estimator, there is certainly more variation in what $\vec{\beta}$ converges to. This is in comparison to the MLE, which despite the fact that there does not necessarily exist a unique MLE, it reliablely converges to atleast the same β_1 time and time again. However, both the Bayesian and the MLE estimators are fairly similar in the sense that they tend to "hug" in the beginning (indicating similar values of β_0) but then diverge in the rate (indicating different values of β_1). This variation in the Bayes estimator is expected, since we are simulating from a non-conjugate prior, and by design include Monte Carlo randomness in the process. Nonetheless, the Bayesian estimator, when given any starting θ_0, in our case $(0.25, 0.5)$, it will remarkably always move towards the MLE, which matches our intuition.

The penalized model, on the other hand, inherits everything about the MLE, except that it seems to result in nudging up the value of β_0 in compensation for the penalization, which means it will almost always have a larger offset than the other two estimators. One difference which is not displayed, however, is the fact that penalized loss can lead to a unique estimator, unlike the MLE. However, in this case it doesn't really matter since β_1 is the more weighed variable, and the MLE seems to be marginally unique in β_1.

```
B_bayesian <- c(mean(df_chain$beta0),mean(df_chain$beta1))
B_bayesian
```

```
## [1] 0.02312066 0.02437528
```



IV. Reflection

If you've made it this far and been able to put good thought into these questions, congratulations! Reflect how far you've come: I was able to throw a novel model class at you and you were able to think about it in a frequentist and Bayesian framework, address estimation and inference using power algorithms and theoretical results, and think about how these different flavors of Poisson regression behave the way they do on this particular data set. At this point, I believe you can call yourself a statistician!

I have two last reflection questions for you.

Question 16: Take one question from a problem set that you worked on this semester that you struggled to understand and solve, and explain how the struggle itself was valuable.

Answer: I remember struggling with MCMC algorithms, and specifically Metropolis-Hastings (ha..) in Problem Set 10. After going through the lecture notes, and meticulously trying to dissect all the moving parts and pieces by writing them down, I was able to slowly and more confidently make progress debugging my code. In each little bit of progress, I understood the algorithm more and what every piece was doing, until alas, by the end of it, I understood the major steps that comprise the algorithm! Learning the MCMC simulation algorithms were generally very valuable since they crystallized all my prior (no pun intended) knowledge since it involved CI's, estimators, properties of estimators, sampling, and forced myself to struggle and organize all these objects in a meaningful way.

Question 17: Give one example of a statistical idea from this class that you found creative and explain what you find creative about it.

Answer: I think the German Tank problem was one of the most memorable things covered in class. Because it was both simple and historical, it garnered my focus. When I was able to dissect and categorize the parameters in the problem (which had a very transparent, clear goal), I believe it was one of the first times I lucidly thought about what it was that I was optimizing, like a true statistician! I think that was the day that Bayesian thinking clicked in my head, because before that day, I always thought the Unif(a, b) was just a distribution statisticians made up for completeness, but come this day, boy do I realize how I was wrong...
