Homework 2

Neal Marquez

October 12, 2018

```
rm(list=ls())
library(tidyverse)
library(latex2exp)
```

Question 1

Consider the following dataset: y = (1, 0, 0, 1, 0, 0, 0, 0), which happens to be a series of independent realizations of a Bernoulli distributed random variable. Plot the likelihood function for the Bernoulli parameter π given y. 1 If the experiment were repeated, roughly what fraction of the observations would you expect to be successes (y = 1)? Why?

We can calculate the likelihood function by using the probability mass function of the Bernoulli distribution which can be expressed as

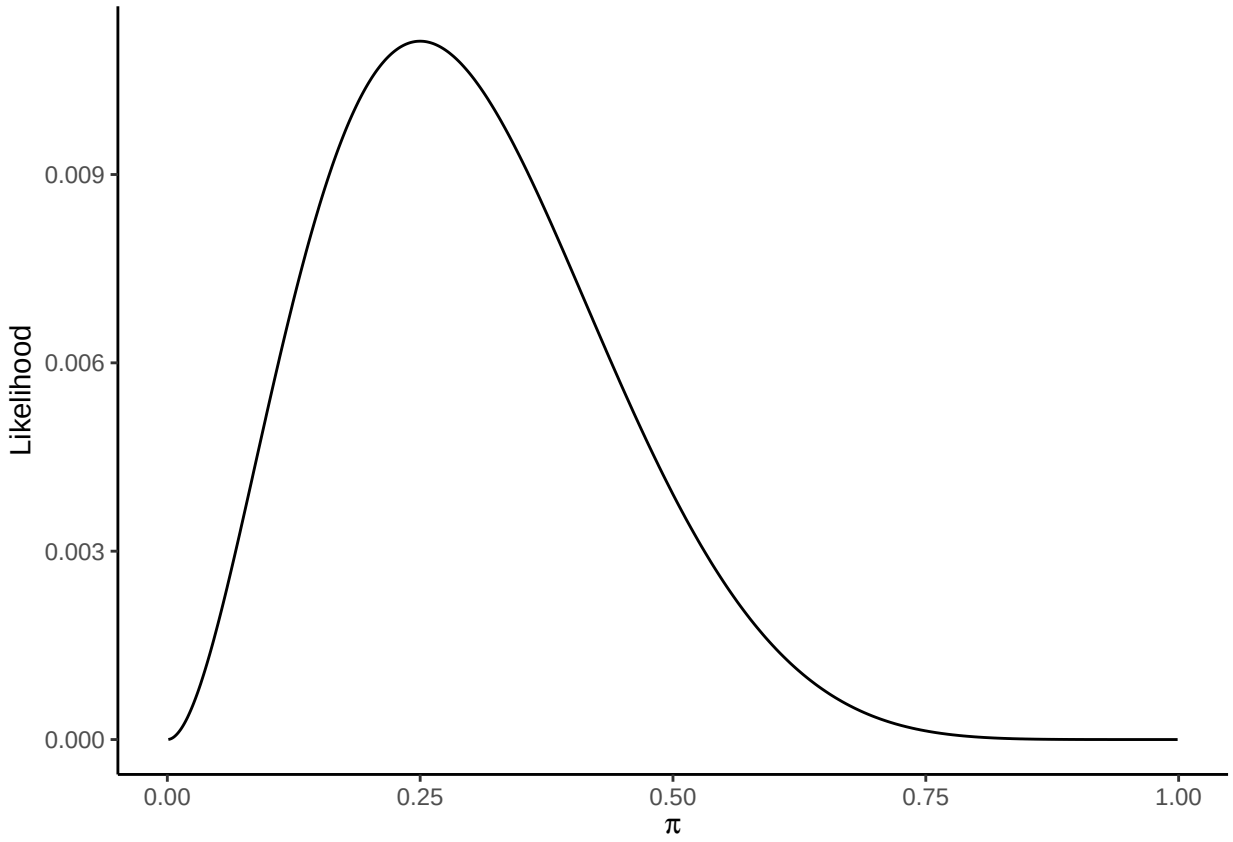
$$(1-\pi)^{1-x}\pi^x$$

The probability mass function can then be calculated for each observation for a specified value of π and multiplied together in order to get a likelihood for π. The plot below shows the process of calculating the likelihood for nearly 1000 values evenly spaced between 0 and 1. We see that the peak of the profile likelihood is at .25 which is not only the mean of our data but also the MLE for the parameter. This would mean that we would expect $\frac{1}{4}$ of future observations to be 1.

```
dbernoulli <- function(x, p){
    (1-p)^(1-x) * p^(x)
}

y <- c(1, 0, 0, 1, 0, 0, 0, 0)

data.frame(pi=seq(.001, .999, by=.001)) %>%
    mutate(lik=sapply(pi, function(p) prod(dbernoulli(y, p)))) %>%
    ggplot(aes(x=pi, y=lik)) +
    geom_line() +
    theme_classic() +
    xlab(TeX("$\\pi$")) +
    ylab("Likelihood")
```



Question 2

Section A

Derive the log-likelihood function corresponding to this pdf:

$$f_{\text{Poisson}}(\lambda) = \lambda^{y} e^{-\lambda} (y!)^{-1}$$

Derivation

$$\log\Big(\prod_{i=1}^{N} \lambda_i^{y_i} e^{-\lambda_i} (y_i!)^{-1}\Big)$$

$$\sum_{i=1}^{N} \log(\lambda_i^{y_i} e^{-\lambda_i} (y_i!)^{-1})$$

$$\sum_{i=1}^{N} \log(\lambda_i^{y_i}) + log(e^{-\lambda_i}) + log((y_i!)^{-1})$$

$$\sum_{i=1}^{N} y_i \log(\lambda_i) - \lambda_i - log(y_i!)$$

$$\propto \sum_{i=1}^{N} y_i \log(\lambda_i) - \lambda_i$$

Section B

Suppose that for periods $i \in \{1, ..., i, ..., n\}$, each of (potentially varying) length t_i, we observe counts y_i. We would like to use the Poisson distribution to model these data, but an assumption of the Poisson distribution(and hence the Poisson regression model) is that the periods observed are of equal length. Relax this assumption, and revise your derivation in part a accordingly.

Derivation Revised

$$\log\Big(\prod_{i=1}^{N} t_i\lambda_i^{y_i} e^{-t_i\lambda_i}(y_i!)^{-1}\Big)$$

$$\sum_{i=1}^{N} \log((t_i\lambda_i)^{y_i} e^{-t_i\lambda_i}(y_i!)^{-1})$$

$$\sum_{i=1}^{N} \log((t_i\lambda_i)^{y_i}) + log(e^{-t_i\lambda_i}) + log((y_i!)^{-1})$$

$$\sum_{i=1}^{N} y_i \log(t_i\lambda_i) - t_i\lambda_i - log(y_i!)$$

$$\propto \sum_{i=1}^{N} y_i \log(t_i\lambda_i) - t_i\lambda_i$$

Section C

Write an `R` function, usable by `optim()` , implementing this variable-period Poisson estimator.

```r
evalPoisson <- function(y, t, lamb, negative=TRUE){
    ll <- sum(y * log(t * lamb) - lamb*t)
    if(negative){
        ll <- -ll
    }
    ll
}

evalBetas <- function(betas, df){
    lambs <- exp(c(as.matrix(cbind(rep(1, N), select(df, x1, x2))) %*% betas))
    evalPoisson(df$y, df$time, lambs)

}
```

Section D

Generate an artifical dataset with 1000 observations consisting of three variables:

$$x_{1i} \sim \text{Uniform}(0, 1)$$

$$x_{2i} \sim \text{Uniform}(0, 1)$$

$$y_i \sim \text{Poisson}(t_i\lambda_i)$$

where

$$\lambda_i = \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i})$$

and t_i is a random draw from the integers $\{1, 2, 3, 4, 5\}$; this is the length of the period. Assume that the true values of the model parameters are $\beta_0 = 0$, $\beta_1 = 1$, and $\beta_2 = 2$. Confirm that the mean of y_i is approximately 16.5, and the standard deviation of y i is approximately 14.7 (your results may differ by as much as one unit in each case due to Monte Carlo error).

```
set.seed(123)
N <- 1000
b0 <- 0
b1 <- 1
b2 <- 2
simDF <- tibble(x1=runif(N, min=0, max=1), x2=runif(N, min=0, max=1)) %>%
    mutate(time=sample.int(5, size=N, replace=TRUE)) %>%
    mutate(y=rpois(N, time*exp(b0 + b1 * x1 + b2*x2)))

cat(paste0("Y has a mean of ", round(mean(simDF$y), 4), "\n"))
```

```
## Y has a mean of 16.35
```

```
cat(paste0("Y has a standard deviation of ", round(sd(simDF$y), 4)))
```

```
## Y has a standard deviation of 15.1319
```

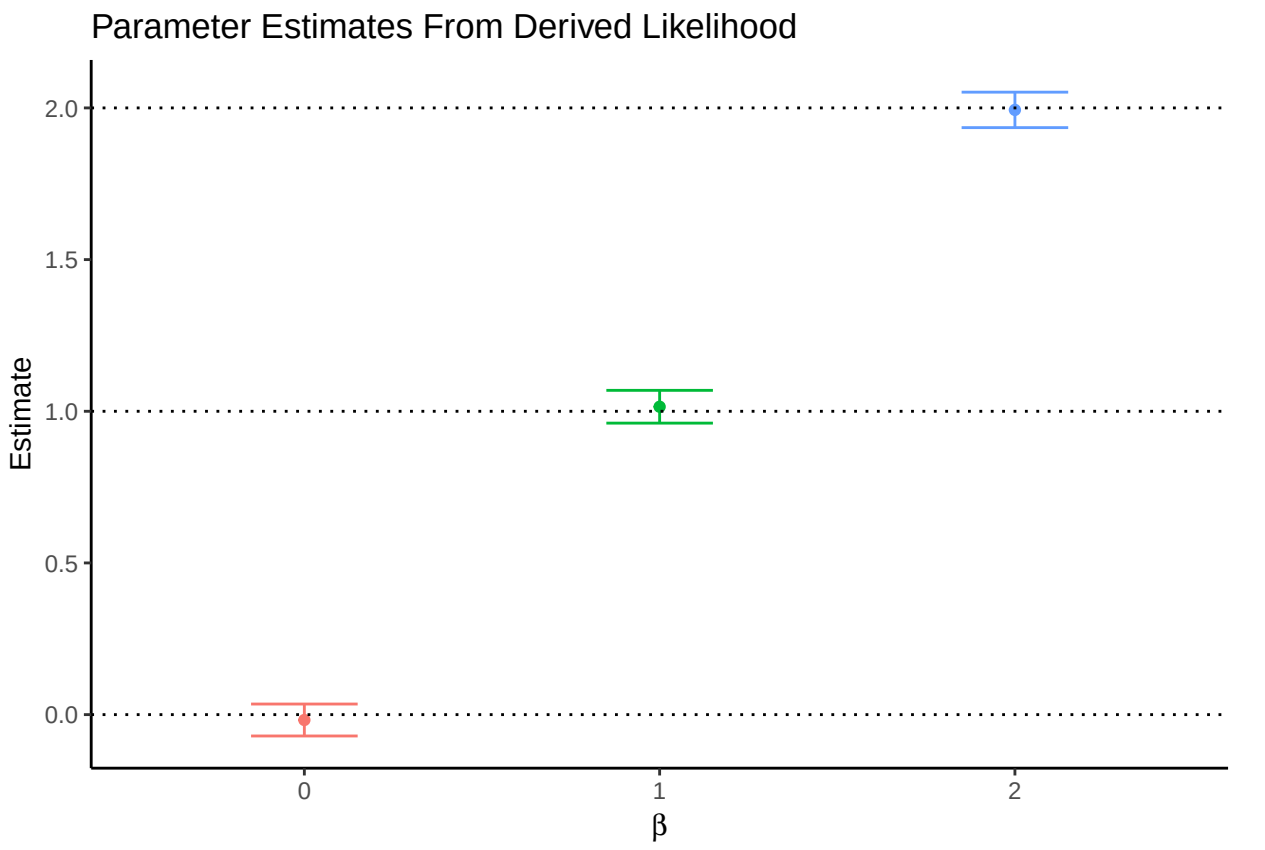
Section E

Using the artificial data you generated in part d and the `R` function for the unequal periods Poisson likelihood you wrote in part c, estimate the parameters β_0, β_1, and β_2 and report your point estimates and their standard errors. Does this exercise give you confidence that you have correctly derived and implemented the Poisson MLE for unequal periods? What could you do to increase your confidence further?

When we use the `optim` function we can obtain the maximum likelihood of our parameters via an optimization algorithm of our choosing. In this case we use the `Nelder-Mead` method for finding the minimum so we need to be sure that we use the negative log likelihood rather than the log likelihood. A benefit of this approach is that we also obtain estimates of our parameter uncertainty based on the curvature of our likelihood space. The results obtained from this experiment appears to recover the true parameter values in the uncertainty band giving us evidence that we correctly derived the likelihood function. We could increase our confidence by doing this experiment many times with different parameters and making sure that we capture the true parameters in the confidence interval bands an appropriate number of times.

```
runoptim1 <- optim(c(0, 0, 0), evalBetas, df=simDF, hessian=TRUE)
vcovMat <- solve(runoptim1$hessian)
se <- sqrt(diag(vcovMat))

data.frame(Beta=factor(0:2), est=runoptim1$par) %>%
    mutate(upper=est + se*1.96, lower=est - se*1.96) %>%
    ggplot(aes(x=Beta, y=est, ymin=lower, ymax=upper, color=Beta)) +
    geom_point() +
    geom_errorbar(width=.3) +
    theme_classic() +
    geom_hline(yintercept=c(0,1,2), linetype=3) +
    labs(x=TeX("$\\beta$"), y="Estimate") +
    guides(color=FALSE) +
    ggtitle("Parameter Estimates From Derived Likelihood")
```



Question 3

A famous probability problem asks "What is the probability that at least two people in a classroom of n people share the same birthday?"" Write an R program to solve this problem using simulation. Produce as output a plot of the probabilities of at least one shared birthday for n= $\{2, \ldots, 50\}$.

```
birthdaySim <- function(n){
    peopleInRoom <- sample.int(365, n, replace=T)
    # Unique is slow!!!
    length(unique(peopleInRoom)) < length(peopleInRoom)
}

people <- 2:50
N <- 100
M <- 1000

resultMat <- sapply(1:N, function(j){
    sapply(people, function(n){
        sapply(1:M, function(i) birthdaySim(n))}) %>%
        apply(2, mean)
})

tibble(`Room Size`=people, Probability=apply(resultMat, 1, mean)) %>%
    mutate(lwr=apply(resultMat, 1, quantile, probs=.025)) %>%
```

```
    mutate(upr=apply(resultMat, 1, quantile, probs=.975)) %>%
    ggplot(aes(x=`Room Size`, y=Probability, ymax=upr, ymin=lwr)) +
    geom_line() +
    geom_ribbon(alpha=.2) +
    theme_classic() +
    ggtitle("Probability of Observing A Common Birthday in a Room") +
    xlab("Number of People in Room")
```