Homework 2

Amy Kuang, Shravan Shenoy - PSTAT 115, Spring 2021

Due on May 9, 2021 at 11:59 pm

Note: If you are working with a partner, please submit only one homework per group with both names and whether you are taking the course for graduate credit or not. Submit your Rmarkdown (.Rmd) and the compiled pdf on Gauchospace.

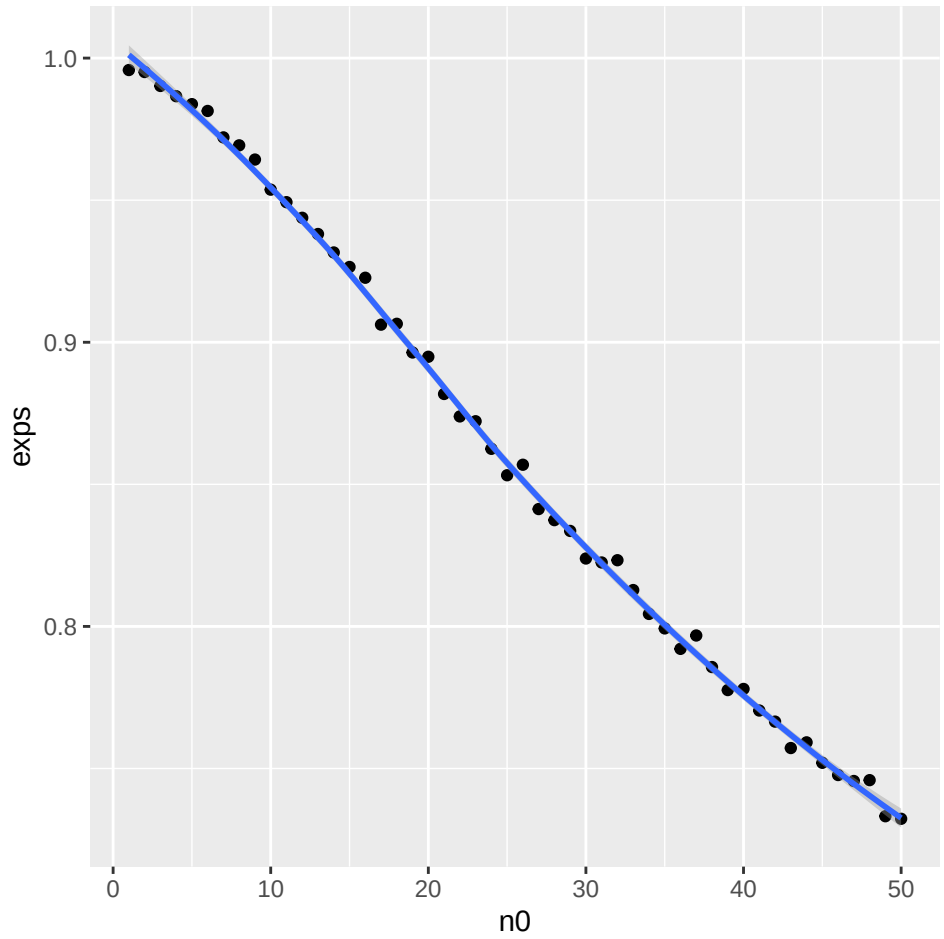
Problem 1. Cancer Research in Laboratory Mice

As a reminder from homework 1, a laboratory is estimating the rate of tumorigenesis (the formation of tumors) in two strains of mice, A and B. They have tumor count data for 10 mice in strain A and 13 mice in strain B. Type A mice have been well studied, and information from other laboratories suggests that type A mice have tumor counts that are approximately Poisson-distributed. Tumor count rates for type B mice are unknown, but type B mice are related to type A mice. Assuming a Poisson sampling distribution for each group with rates θ_A and θ_B. We assume $\theta_A \sim \text{gamma}(120, 10)$ and $\theta_B \sim \text{gamma}(12, 1)$. We observe $y_A = (12, 9, 12, 14, 13, 13, 15, 8, 15, 6)$ and $y_B = (11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)$. Now we will actually investigate evidence that Type A mice are have higher rates of tumor formation than Type B mice.

a. For $n_0 \in \{1, 2, ..., 50\}$, obtain $Pr(\theta_B < \theta_A \mid y_A, y_B)$ via Monte Carlo sampling for $\theta_A \sim \text{gamma}(120, 10)$ and $\theta_B \sim \text{gamma}(12 \times n_0, n_0)$. Make a line plot of $Pr(\theta_B < \theta_A \mid y_A, y_B)$ vs n_0. Describe how sensitive the conclusions about the event $\{\theta_B < \theta_A\}$ are to the prior distribution on θ_B.

```
y_A <- c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)
y_B <- c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

### BEGIN SOLUTION
n0 <- c(1:50)
theta_a <- rgamma(10000, sum(y_A) + 120, length(y_A) +10) # rgamma(10000, 237, 20)

theta_b <- rgamma(10000, sum(y_B) + 12, length(y_B) + 1) # rgamma(10000, 125, 14)

exps <- sapply(n0, function(n) {
  mean(rgamma(10000, (12*n) + 113, n+13) < theta_a)
})
qplot(n0, exps, geom = c('point', 'smooth'))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

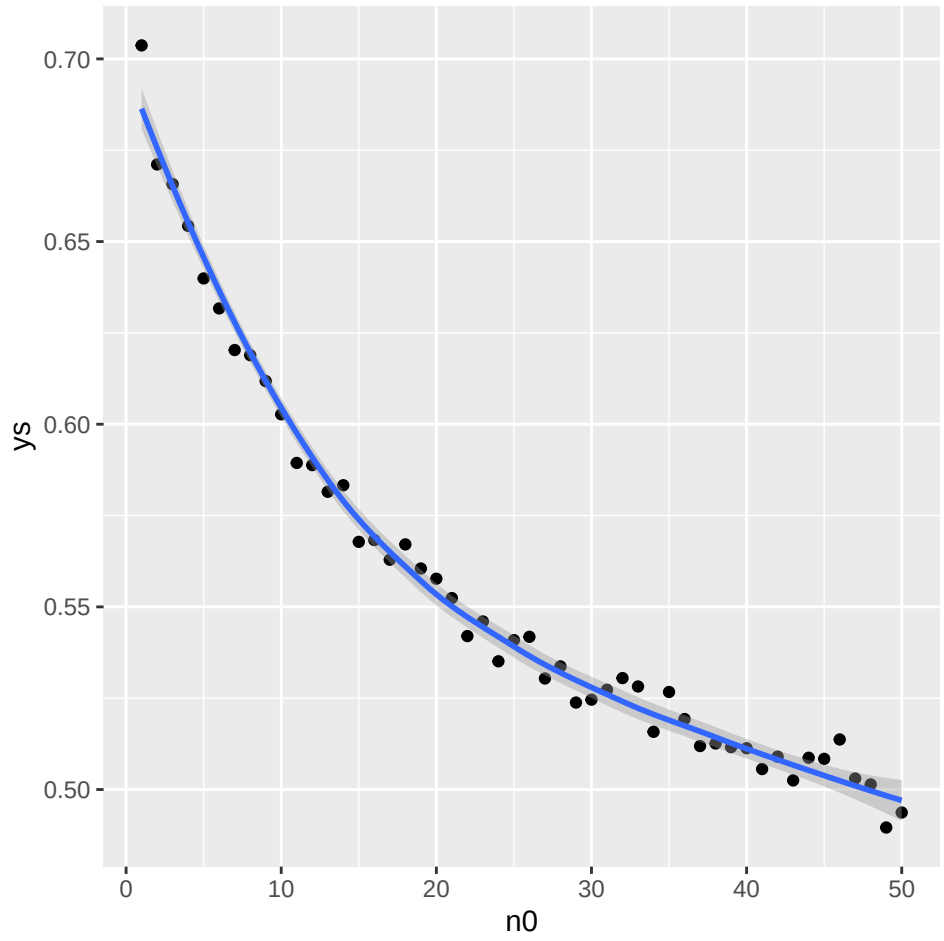


Strong factors of n_0 have little effect on the posterior distribution.

b. Repeat the previous part replacing the event $\{\theta_B < \theta_A\}$ with the event $\{\tilde{Y}_B < \tilde{Y}_A\}$, where $\tilde{Y}_A$ and $\tilde{Y}_B$ are samples from the posterior predictive distribution.

```
y_A = c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)
y_B = c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

### BEGIN SOLUTION
N <- 10000
ys <- sapply(n0, function(n){
  theta_a = rgamma(N, 237, 20)
  theta_b = rgamma(N, 12*n + 113, n+13)
  y_a = rpois(N, theta_a)
  y_b = rpois(N, theta_b)
  mean(y_b < y_a)
})
qplot(n0, ys, geom = c('point', 'smooth'))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



c. In the context of this problem, describe the meaning of the events $\{\theta_B < \theta_A\}$ and $\{\tilde{Y}_B < \tilde{Y}_A\}$. How are they different?

The posterior's predictive distribution is a negative binomial distribution. Further, $\{\theta_B < \theta_A\}$ are parameters of Poisson (which is like the expectation of the tumor count) and ~Y is the *realization* of the tumor count.

2. Posterior Predictive Model Checking

Model checking and refinement is an essential part of Bayesian data analysis. Let's investigate the adequacy of the Poisson model for the tumor count data. Consider strain A mice only for now, and generate posterior predictive datasets $y_A^{(1)}, ..., y_A^{(1000)}$. Each $y_A^{(s)}$ is a sample of size $n_A = 10$ from the Poisson distribution with parameter $\theta_A^{(s)}$, $\theta_A^{(s)}$ is itself a sample from the posterior distribution $p(\theta_A \mid y_A)$ and y_A is the observed data. For each s, let $t^{(s)}$ be the sample average divided by the sample variance of $y_A^{(s)}$.

a. If the Poisson model was a reasonable one, what would a "typical" value $t^{(s)}$ be? Why?

```
y_A = c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)

### BEGIN SOLUTION
S <- 1000
t_s <- numeric(S)
for(s in 1:S){
  theta_s <- rgamma(1, 237, 20)
  ytilde_s <- rpois(n=10, theta_s)
  t_s[s] <- mean(ytilde_s)/var(ytilde_s)
}
```

```
mean(ytilde_s)/var(ytilde_s)
```

```
## [1] 1.400616
```

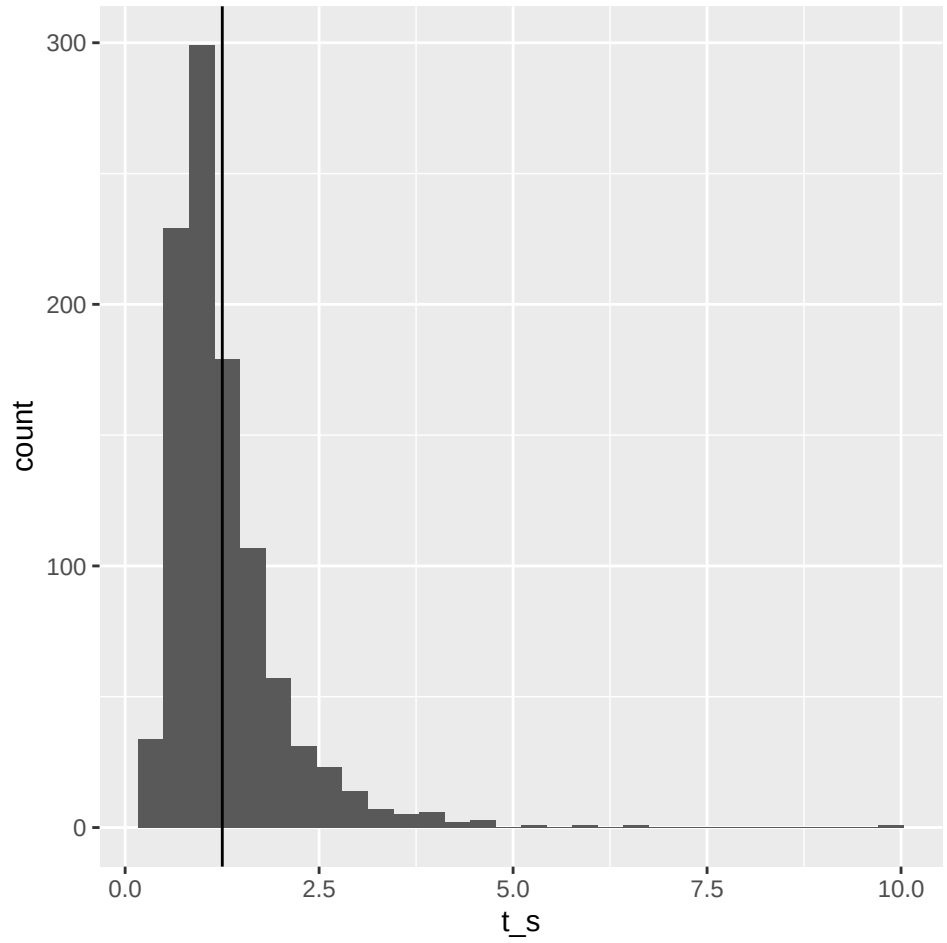
```
# ggplot(data.frame(t_s=t_s), aes(x=t_s)) +
#   geom_histogram() +
#   geom_vline(xintercept = mean(y_A)/var(y_A))
```

If a Poisson model was reasonable, a 'typical value' should be around 1 since mean and variance is the same for Poisson.

b. In any given experiment, the realized value of t^s will not be exactly the "typical value" due to sampling variability. Make a histogram of $t^{(s)}$ and compare to the observed value of this statistic, $\frac{\text{mean}(y_A)}{\text{var}(y_A)}$. Based on this statistic, make a comment on if the Poisson model seems reasonable for these data (at least by this one metric).

```
### BEGIN SOLUTION
S <- 1000
t_s <- numeric(S)

for(s in 1:S){
  theta_s <- rgamma(1, 237, 20)
  ytilde_s <- rpois(n=10, theta_s)
  t_s[s] <- mean(ytilde_s)/var(ytilde_s)
}

ggplot(data.frame(t_s=t_s), aes(x=t_s)) + geom_histogram() + geom_vline(xintercept = mean(y_A)/var(y_A))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean(y_A)/var(y_A)
```

```
## [1] 1.252081
```

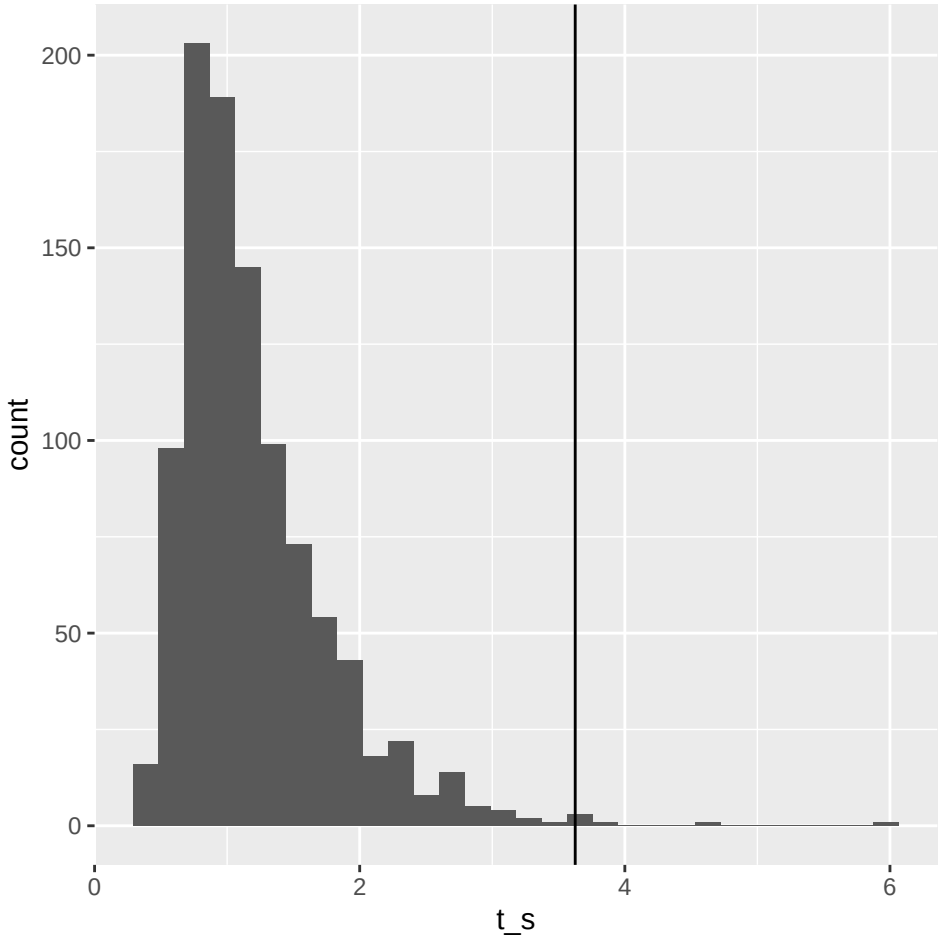
The Poisson model is a reasonable one with t^s observed statistic value of 1.252081. We say it is a reasonable model because the observed statistic ($\frac{mean(y_A)}{SD(y_A)}$) is within the spread of t^s with t^s depending on parameter of θ.

c. Repeat the part b) above for strain B mice, using Y_B and $n_B = 13$ to generate the samples. Assume the prior distribution $p(\theta_B) \sim \text{Gamma}(12, 1)$. Again make a comment on the Poisson model fit.

```
y_B = c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

### BEGIN SOLUTION
S <- 1000
t_s <- numeric(S)

for(s in 1:S){
  theta_s <- rgamma(1,125, 14)
  ytilde_s <- rpois(n=13, theta_s)
  t_s[s] <- mean(ytilde_s)/var(ytilde_s)
}

ggplot(data.frame(t_s=t_s), aes(x=t_s)) +
  geom_histogram() +
```

```
  geom_vline(xintercept = mean(y_B)/var(y_B))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean(y_B)/var(y_B)
```

```
## [1] 3.625668
```

Our observed statistic ($\frac{mean(y_A)}{SD(y_A)}$) is not well within the spread of t^s, and it seems to be an outlier. Thus, we conclude that the Poisson model does not seem to be a reasonable model.

3. Interval estimation with rejection sampling.

a. Use rejection sampling to sample from the following density:

$$p(x) = \frac{1}{4}|\sin(x)| \times I\{x \in [0, 2\pi]\}$$

Use a proposal density which is uniform from 0 to 2π and generate at least 1000 true samples from p(x). Compute and report the Monte Carlo estimate of the upper and lower bound for the 50% quantile interval using the `quantile` function on your samples. Compare this to the 50% HPD region calculated on the samples. What are the bounds on the HPD region? Report the length of the quantile interval and the total length of the HPD region. What explains the difference? Hint: to compute the HPD use the `hdi` function from the `HDInterval` package. As the first argument pass in `density(samples)`, where `samples` is the name of your vector of true samples from the density. Set the `allowSplit` argument to true and use the `credMass` argument to set the total probability mass in the HPD region to 50%.

HPD gives a 95% confidence interval but it minimizes the length in between. As a result, theres a diffe

b. Plot $p(x)$ using the `curve` function (base plotting) or `stat_function` (ggplot). Add lines corresponding to the intervals / probability regions computed in the previous part to your plot using them `segments` function. To ensure that the lines don't overlap visually, for the HPD region set `y0` and `y1` to 0 and for the quantile interval set `y0` and `y1` to 0.01. Make the segments for HPD region and the segment for quantile interval different colors. Report the length of the quantile interval and the total length of the HPD region, verifying that indeed the HPD region is smaller.

```
### Rejection sampling and interval construction
### BEGIN SOLUTION

samples <-numeric()

    while(length(samples)<1000){
    xval<-runif(1,min=0,max=2*pi)
    yval<-runif(1,min=0,max=.25)
    p_x<-.25*abs(sin(xval))
    if (yval<=p_x){
      samples=append(samples, xval)
    }}

#print(samples) # check to see it runs

### hd_region is the result of calling hdi function
hd_region <- HDInterval::hdi(density(samples), allowSplit=TRUE, credMass=0.5) # SOLUTION
print(hd_region)
```

```
##         begin      end
## [1,] 1.003895 2.219455
## [2,] 4.207059 5.241927
## attr(,"credMass")
## [1] 0.5
## attr(,"height")
## [1] 0.1941567
```

```
print(sprintf("Total region length: %.02f", sum(hd_region[, "end"] - hd_region[,"begin"])))
```

```
## [1] "Total region length: 2.25"
```

```
quantile_interval <- quantile(samples, c(0.25, 0.75)) # SOLUTION
print(quantile_interval)
```

```
##      25%      75%
## 1.584235 4.726566
```

```
print(sprintf("Total region length: %.02f", quantile_interval[2] - quantile_interval[1]))
```

```
## [1] "Total region length: 3.14"
```

```
  ### Make the plot
  ### BEGIN SOLUTION


p_xfunc <- function() {
    .25*(sin(x))
}
```

```
# bplot <- ggplot(data.frame(x = c(0:(2*pi))), aes(x = x)) +
#         stat_function(fun = p_xfunc)
# bplot
#
```