Overview

This assignment tests your understanding of the concepts, math, and programming required to learn distributions from data. You are required to perform a mixture of derivations and programming to solve the questions.

READ the problem statements CAREFULLY!

IMPORTANT: The RMarkdown assumes you have downloaded two data sets (CSV files) and saved them to the same directory you saved the template Rmarkdown file. If you do not have the CSV files in the correct location, the data will not be loaded correctly.

IMPORTANT!!!

Certain code chunks are created for you. Each code chunk has `eval=FALSE` set in the chunk options. You **MUST** change it to be `eval=TRUE` in order for the code chunks to be evaluated when rendering the document.

You are free to add more code chunks if you would like.

Load packages

This assignment will use packages from the `tidyverse` suite.

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Problem 01

You have fit discrete and continuous distributions to data, using non-Bayesian and Bayesian approaches. Bayesian analyses require a prior to be formulated, and it can be difficult to understand how a prior is specified in a general setting. This exam seeks to give you some practice doing that by using the **Empirical Bayes** approach. Empirical Bayes is a rather odd sounding name, but the idea is that you will estimate the parameters of the prior using all of the available data. It is useful when the data can be structured into **groups**. Some groups might have many observations, while others may have a limited number of samples. Empirical Bayes is useful when there are many groups (potentially in the thousands) that can be used to estimate the prior parameters. Once estimated, the prior is applied to each group separately. In this manner you have made use of data to understand the relevant bounds on your unknowns and specified those bounds within a prior probability distribution. The *informative* prior is updated based on each group's data to yield the updated belief (the posterior) for each group. (Note that if we would have very few groups we could not use Empirical Bayes and thus would need to use full Bayesian approaches via multilevel, hierarchical, or partial pooling models.)

To see how the Empirical Bayes process works you will work with a Movie Rating application. You are interested in learning the probability a movie receives a POSITIVE user review. Movies with a high probability of a positive review correspond to movies that audiences enjoyed and liked to watch. Knowing how to estimate the probability of a positive review is important for **RECOMMENDATION ENGINES**. Companies like Amazon, Netflix, and others design RECOMMENDATION ENGINES that recommend POPULAR items to customers. You are working with **real** data for this exam. The data were **not** generated artificially.

The data come from a movie rating database that is commonly used for teaching the fundamentals of RECOMMENDATION ENGINES. You will not work with the complete database however. Instead you are provided a representative sample. Also, the identifying information as been removed and altered relative to the database. In this way you can not look up the movie the data are associated with.

You will use Bayesian techniques in this exam to learn a fundamental quantity associated with RECOMMENDATION ENGINES. You will learn the probability a product (movies in this case) is popular given user ratings! By using Bayesian techniques you will naturally be able to provide the **uncertainty** on the probabilities, rather than just relying on simple **point estimates**. The movie ratings have been BINARIZED into a user liked the movie OR the user did not like the movie. The way the original movie ratings were BINARIZED is not relevant to this exam. What matters is the **EVENT** of interest is a user giving a POSITIVE review (liking the movie), while the **NON-EVENT** is a user giving a NEGATIVE review (not-liking the movie). The probability of a POSITIVE review is therefore the **event probability** you are interested in learning.

You will work with two data sets for this exam. Both are loaded for you below. The first, `df_all`, is the larger of the two. The second, `df_focus`, is a subset of `df_all` so that we way can focus on 30 movies to help with visualization, interpretation, and discussion in the exam.

```r
file_with_all <- "midterm_all_data.csv"
df_all <- readr::read_csv(file_with_all, col_names = TRUE)
```

```
## Rows: 11801 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): movie_id, num_trials, num_events
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
file_with_focus <- "midterm_focus_data.csv"
df_focus <- readr::read_csv(file_with_focus, col_names = TRUE)
```

```
## Rows: 30 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): movie_id, num_trials, num_events
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Both data sets consist of 3 variables, `movie_id`, `num_events`, and `num_trials`. The `num_events` is the number of POSITIVE reviews, and `num_trials` is the number of reviews. The variables are written in the general terms that we have used in the class rather than movie or recommendation engine specific terms. The `movie_id` variable is an ID variable for each movie. Thus, one row in either data set tells us the number of POSITIVE reviews and the total number of reviews associated with a movie. Again, the data in this exam are real. The `movie_id` variable is an anonymous identification number I created so that you cannot link the exam data back to the original online data base.

1a)

You will eventually use the Empirical Bayes approach in this exam. However, let's motivate why such an approach is useful. You will begin by assuming you are not capable of creating an informative prior. For example, even if you watch a lot of movies, you may not feel comfortable expressing the chance someone else will like a movie! Since you do not feel comfortable specifying reasonable bounds, you decide to use a VAGUE and uninformative prior formulation.

You will use a Binomial likelihood and a conjugate Beta prior on the unknown event probability, μ. For generality, you will denote each movie with a subscript j and the total number of movies as J. Thus, the unknown event probability for the j-th movie is μ_j where $j = 1, ..., J$. The posterior distribution on the j-th movie's unknown event probability, μ_j given the m_j events out of N_j trials is proportional to:

$$p\left(\mu_j \mid (m, N)_j\right) \propto \text{Binomial}\left(m_j \mid \mu_j, N_j\right) \times \text{Beta}\left(\mu_j \mid a, b\right)$$

Notice that in the above posterior formulation, each movie has a potentially distinct event probability, μ_j. The prior consists of two shape hyperparameters, a and b. The **SAME** prior shape parameters are applied to every movie.

You will assume prior shape parameters of $a = 0.5$ and $b = 0.5$. How many "prior trials" does this specification correspond to? Why do you think it represents being "uninformed" about the process?

SOLUTION

1) "a" is the number of "events" and "b" is the number of "non-events". So, for a=0.5 and b=0.5, the specification corresponds to 0.5+0.5=1 "prior trials".
2) Since 1 trial is a small number of trials, it represents being "uninformed" about the process.

1b)

You are using a conjugate prior to the Binomial likelihood, for each movie.

What type of distribution is the posterior for the unknown event probability, μ_j, for each movie, $j = 1, ..., J$?

SOLUTION The beta distribution has the same functional form as the Binomial distribution. Therefore, the posterior for the unknown event probability is a beta distribution.

1c)

Write out the formula for the updated or posterior shape parameters, $a_{new,j}$ and $b_{new,j}$, based on each movie's observed number of events m_j and observed number of trials N_j, as well as the prior shape parameters, a and b.

You do not have to derive the formula for these updated parameters. You may simply write their formula below.

SOLUTION We can write the formulas as follows:

$$a_{new,j} = a + m_j$$
$$b_{new,j} = b + (N_j - m_j)$$

1d)

What is the mean, 0.05 Quantile, 0.95 Quantile, and middle 90% uncertainty interval on the event probability according to the assumed "uninformed" prior?

SOLUTION The mean is $\frac{a}{a+b} = \frac{0.5}{0.5+0.5} = 0.5$.

For the 0.05 Quantile and 0.95 Quantile, we can use:

```
q05 <- qbeta(0.05, shape1=0.5, shape2=0.5)
q95 <- qbeta(0.95, shape1=0.5, shape2=0.5)
q05
```

```
## [1] 0.00615583
```

```
q95
```

```
## [1] 0.9938442
```

So, we obtain that “0.05 Quantile = 0.00615583”, and “0.95 Quantile = 0.9938442”.

Finally, the middle 90% uncertainty interval is [0.00615583,0.9938442].

1e)

Based on your formula in Problem 1c), calculate the updated shape parameters for the 30 movies in the `df_focus` tibble. You should add two columns using `mutate()` named `anew` and `bnew`. Assign your result to the `post_df_focus_from_vague` object.

```
post_df_focus_from_vague <- df_focus %>%
  mutate(anew = 0.5 + num_events,
         bnew = 0.5 + (num_trials - num_events))
```

SOLUTION

1f)

Calculate the Posterior Mean, Posterior 0.05 Quantile, and the Posterior 0.95 Quantile for each movie in `post_df_focus_from_vague`. You should add 3 columns using `mutate()` named `post_avg`, `post_q05`, and `post_q95`. Assign the result to the variable `summary_post_df_focus_from_vague`.

```
summary_post_df_focus_from_vague <- post_df_focus_from_vague %>%
  mutate(post_avg = anew/(anew+bnew),
         post_q05 = qbeta(0.05, anew, bnew),
         post_q95 = qbeta(0.95, anew, bnew))
```

SOLUTION

1g)

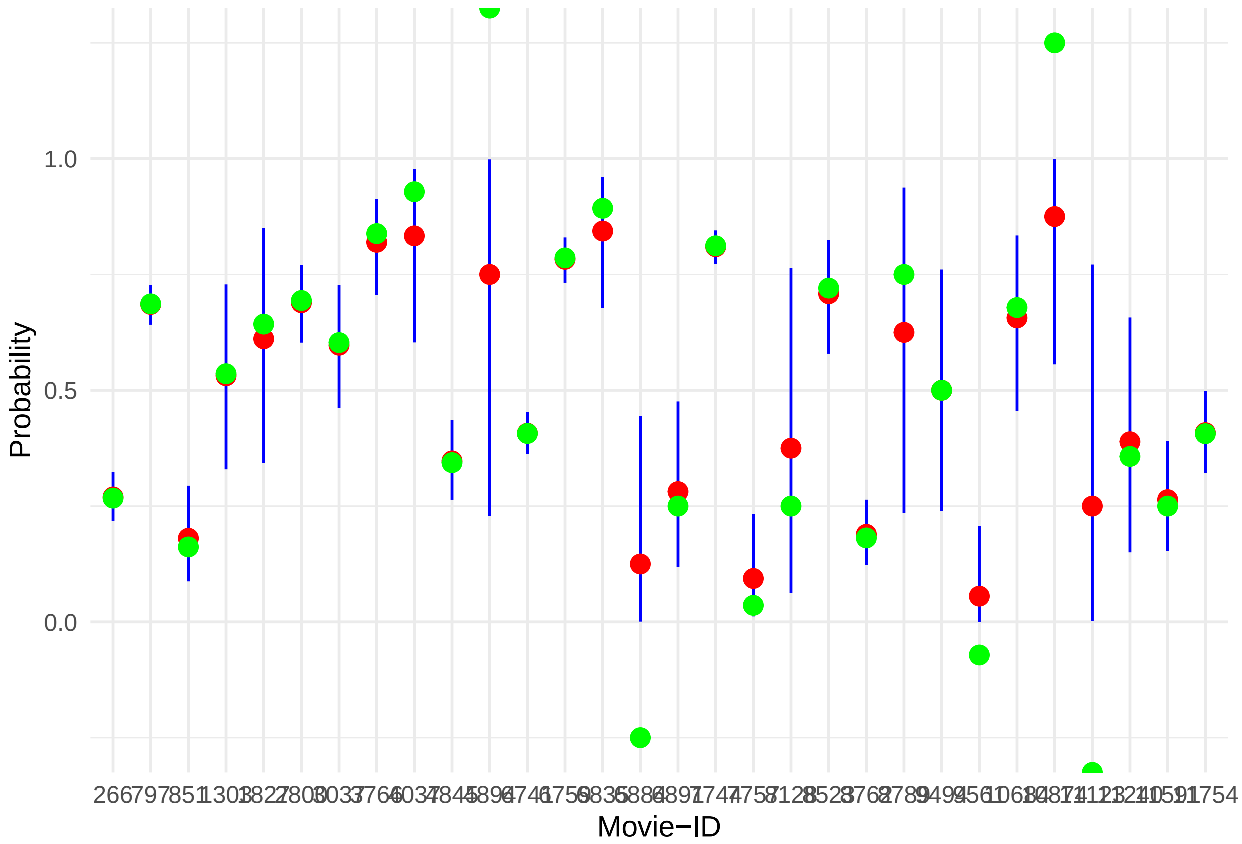
You will now visualize the posterior summaries for the 30 movies associated with the `df_focus` data set. The bold face font below provides specific instructions for visualizing the posterior summaries for this problem. Please read the instructions CAREFULLY! That said, you are free to set colors as you wish.

Pipe `summary_post_df_focus_from_vague` into `ggplot()` and map the `x` aesthetic to `as.factor(movie_id)`. You will use the `geom_linerange()` to represent the posterior uncertainty by setting the `ymin` and `ymax` aesthetics to `post_q05` and `post_q95`, respectively. You will display the posterior mean with a `geom_point()` by setting the `y` aesthetic to `post_avg`.

Include the maximum likelihood estimate (MLE) on the event probability as an additional `geom_point()` geom by mapping the `y` aesthetic to the correct value, which you must calculate.

Are there movies with MLEs that are outside the posterior uncertainty interval? Are there movies with posterior mean values that are quite close to the MLEs?

```
ggplot(summary_post_df_focus_from_vague, aes(x = as.factor(movie_id)))+
  geom_linerange(aes(ymin = post_q05, ymax = post_q95), color = "blue")+
  geom_point(aes(y=post_avg), color = "red", size=3)+
  geom_point(aes(y = (anew-1)/(anew+bnew-2)), color ="green", size=3) +
  labs(x = "Movie-ID", y = "Probability")+
  theme_minimal()
```



SOLUTION

1) In the figure above, red dots represent posterior mean values and green dots represent the maximum likelihood estimate.

2) Yes, there are some movies with MLEs that are outside the posterior uncertainty interval. Also, there are some movies with posterior mean values that are quite close to the MLEs.

3) Here we can also note that some MLE's are calculated outside of the interval [0,1]. This is because the formula for MLE in posterior beta distribution is: $\mu_{MLE} = \frac{anew-1}{anew+bnew-2}$, and when a_{new} or b_{new} (or both) values are less than 1, the formula gives values outside the interval [0,1]. In those cases, we can consider μ_{MLE} equals one of the endpoints of the interval [0,1].

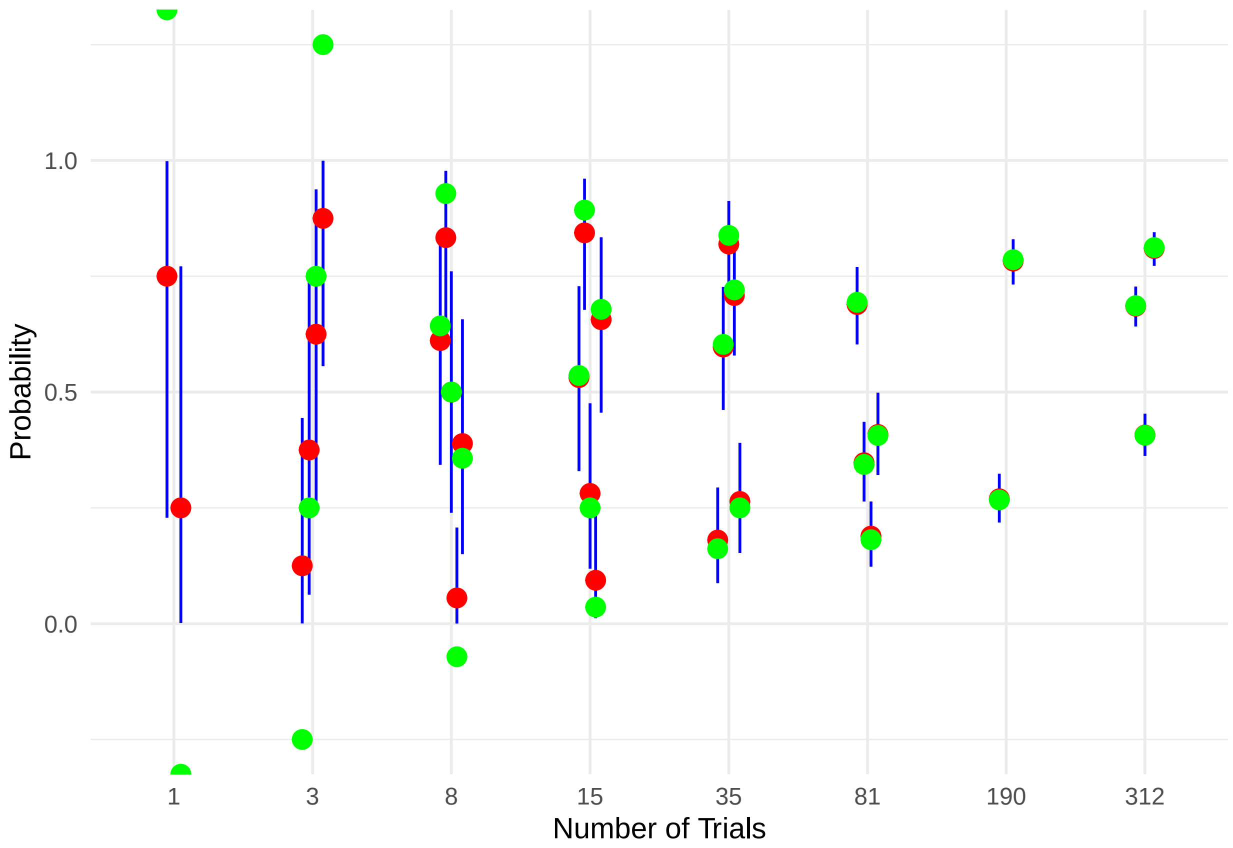
1h)

You will create a similar visualization to that from Problem 1g), except you will map the `x` aesthetic to `as.factor(num_trials)` instead of mapping the `x` aesthetic to `as.factor(movie_id)`. You must also map the `group` aesthetic in each geom to the `movie_id` variable. Doing so allows you to DODGE the posterior summaries for each movie associated with each `num_trials` value.

To properly apply the dodging, set the `position` argument to be `position = position_dodge(0.2)` in `geom_linerange()` and both `geom_point()` calls. You should not place `position` inside `aes()`, it should be outside `aes()`.

Based on your visualization, which movies have high posterior uncertainty on the event probability?

```r
ggplot(summary_post_df_focus_from_vague, aes(x = as.factor(num_trials), group = movie_id)) +
  geom_linerange(aes(ymin = post_q05, ymax = post_q95), color="blue", position = position_dodge(0.2)) +
  geom_point(aes(y = post_avg), color = "red", size =3, position = position_dodge(0.2)) +
  geom_point(aes(y = (anew-1)/(anew+bnew-2)), color ="green", size =3, position = position_dodge(0.2)) +
  labs(x = "Number of Trials", y = "Probability") +
  theme_minimal()
```



SOLUTION

Based on the visualization, the movies with smaller number of reviews (that is “num_trials”) have high posterior uncertainty on the event probability. As “num_trials” increase, the uncertainty intervals shrink.

1i)

You will now calculate the posteriors for **ALL** movies based on the uninformed or VAGUE prior. Thus, you will NOT work with the limited number of movies in the “focused” data set.

Calculate the updated shape parameters for ALL movies in the `df_all` tibble. You should add two columns using `mutate()` named `anew` and `bnew`. Assign your result to the `post_df_all_from_vague` object.

```r
post_df_all_from_vague <- df_all %>%
  mutate(anew = 0.5 + num_events,
         bnew = 0.5 + (num_trials - num_events))
```

SOLUTION

1j)

Calculate the Posterior Mean, 0.05 Quantile, and 0.95 Quantile for each movie in `post_df_all_from_vague`. You should add 3 columns using `mutate()` named `post_avg`, `post_q05`, and `post_q95`. Assign the result to the variable `summary_post_df_all_from_vague`.

```
summary_post_df_all_from_vague <- post_df_all_from_vague %>%
  mutate(post_avg = anew/(anew+bnew),
         post_q05 = qbeta(0.05, anew, bnew),
         post_q95 = qbeta(0.95, anew, bnew))
```

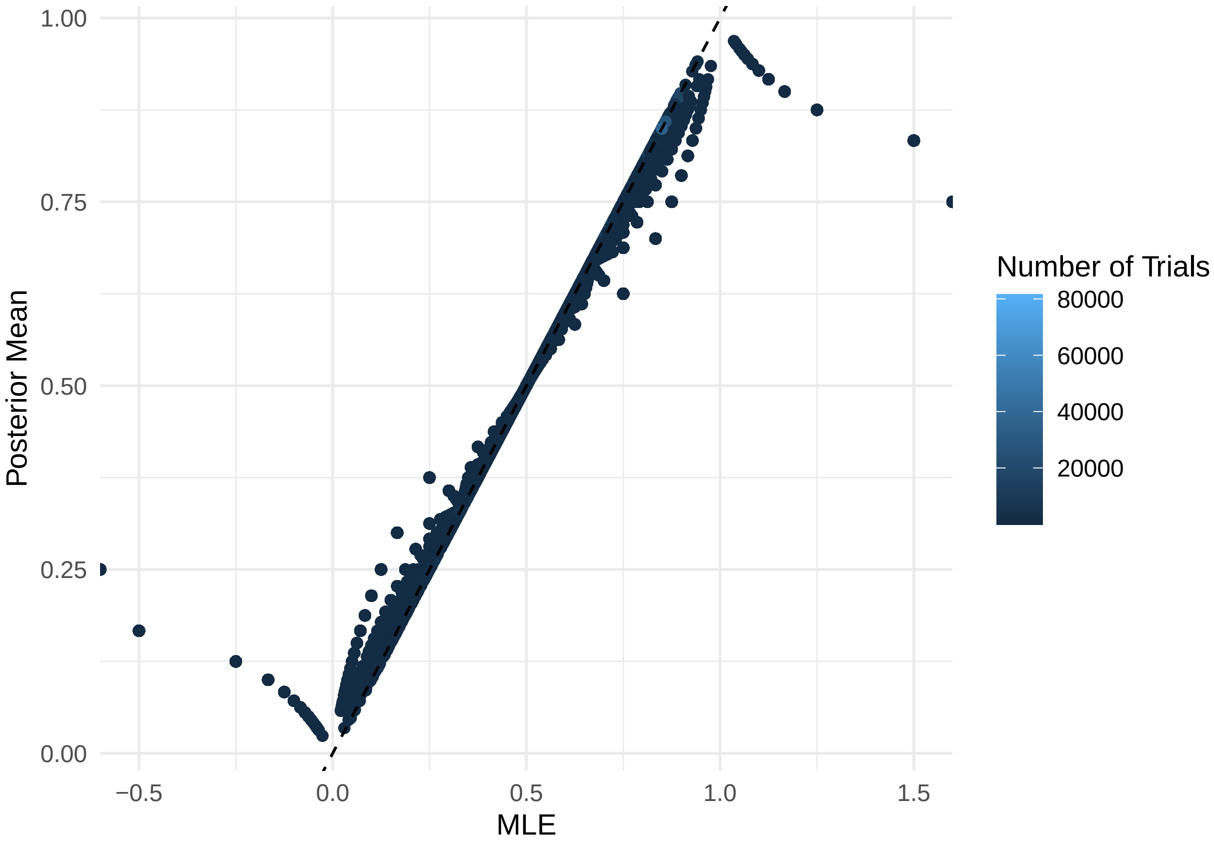
SOLUTION

1k)

You will now visualize the Posterior Mean, based on the uninformed or VAGUE prior, relative to the Maximum Likelihood Estimate for the event probability.

Create a scatter plot with ggplot2 where you plot the `post_avg` with respect to the maximum likelihood estimate to the unknown event probability for all movies. Map the `color` aesthetic to `num_trials` and include a `geom_abline()` layer with `slope = 1` and `intercept=0`.

```
ggplot(summary_post_df_all_from_vague, aes(x=(anew-1)/(anew+bnew-2), y=post_avg, color= num_trials)) +
  geom_point() +
  geom_abline(intercept =0, slope= 1, linetype= "dashed") +
  labs(x ="MLE", y ="Posterior Mean", color ="Number of Trials") +
  theme_minimal()
```

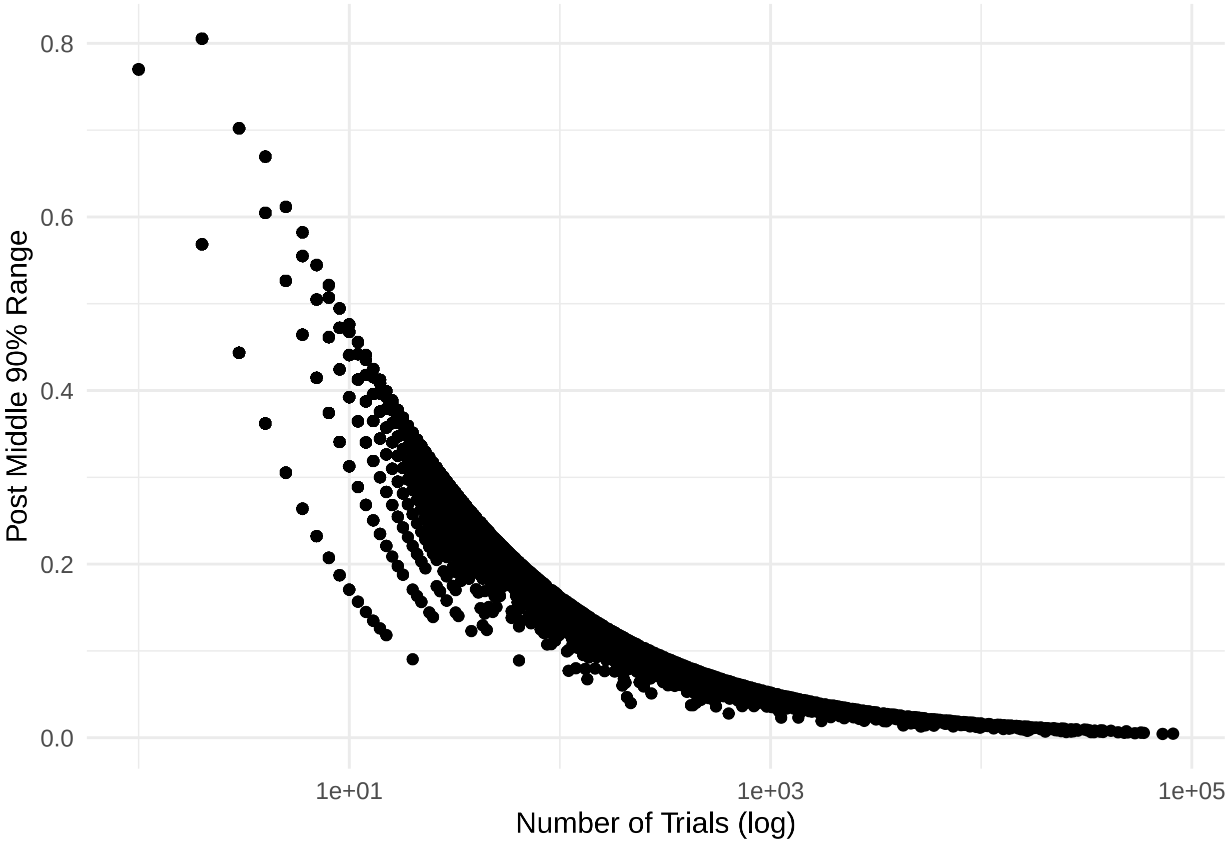


SOLUTION

11)

Create a scatter plot for the Posterior middle 90% uncertainty interval range (difference between the 0.95 and 0.05 Quantiles) with respect to the `num_trials` using `ggplot2`. Include the `scale_x_log()` layer to transform the `x` aesthetic scale via the `log10()` function.

```r
ggplot(data = summary_post_df_all_from_vague, aes(x =num_trials, y=post_q95-post_q05)) +
  geom_point()+
  scale_x_log10()+
  labs(x="Number of Trials (log)", y="Post Middle 90% Range")+
  theme_minimal()
```



SOLUTION

Problem 02

In Problem 01, you estimated the unknown event probability for each movie separately from all other movies. Essentially, you were focused on one movie at a time. This style of analysis is known as the **UNPOOLED ESTIMATE**, since you are not combining or "pooling" the movies (or in general terms the "groups") together.

The opposite view point is to **COMPLETELY POOL** all movies together in order to estimate a SINGLE unknown event probability μ. For this, you will assume that all movies are independent. Thus the posterior distribution on the unknown "pooled" event probability, μ, is proportional to:

$$p\left(\mu \mid \left((m, N)_j\right)_{j=1}^{J}\right) \propto \prod_{j=1}^{J} \left(\text{Binomial}\left(m_j \mid \mu, N_j\right)\right) \times \text{Beta}\left(\mu \mid a, b\right)$$

Pay close attention to the subscripts in the above expression. And notice that the prior on the "pooled" unknown μ relies on the prior shape parameters a and b.

2a)

Write out the log-posterior on the pooled unknown μ up to a normalizing constant in terms of the observations, m_j and N_j for $j = 1, ..., J$, and the prior shape parameters, a and b. Your result should contain a summation series over the J movies.

SOLUTION Using the definitions of Binomial and Beta distributions, we get the following:

$$p\left(\mu \mid \left((m,N)_j\right)_{j=1}^{J}\right) \propto \prod_{j=1}^{J} \left(\mu^{m_j}(1-\mu)^{N_j-m_j}\right) \times \mu^{a-1}(1-\mu)^{b-1}$$

If we take natural log of both sides and then use the properties of logarithm, the expression simplifies as:

$$\log\left(p\left(\mu \mid \left((m,N)_j\right)_{j=1}^{J}\right)\right) \propto \left(a-1+\sum_{j=1}^{J} m_j\right)\log\mu + \left(b-1+\sum_{j=1}^{J}(N_j-m_j)\right)\log(1-\mu)$$

which gives the desired result.

2b)

The summation series in your solution to 2a) can be simplified by using the average number of events, $\bar{m}$ and the average number of trials $\bar{N}$. The average number of events is defined as:

$$\bar{m} = \frac{1}{J}\sum_{j=1}^{J}(m_j)$$

and the average number of trials is defined as:

$$\bar{N} = \frac{1}{J}\sum_{j=1}^{J}(N_j)$$

Write your result from 2a) in terms of $\bar{m}$, $\bar{N}$, J, and the prior shape parameters a and b.

SOLUTION If we use the expressions given above for $\bar{m}$ and $\bar{N}$, in the final expression of 2a), we get the following:

$$\log\left(p\left(\mu \mid \left((m,N)_j\right)_{j=1}^{J}\right)\right) \propto (a-1+J\bar{m})\log\mu + \big(b-1+J(\bar{N}-\bar{m})\big)\log(1-\mu)$$

which is the desired result.

2c)

Your expression in 2b) should look familiar.

What type of posterior distribution does the unknown "pooled" estimate μ have?

Write out the formulas for the posterior or updated shape parameters for your specified posterior distribution.

SOLUTION The posterior distribution for the unknown "pooled" estimate μ is a beta distribution. Using the result in 2b), we can express the updated shape parameters as follows:

$$a_{new} = a + J\bar{m}$$
$$b_{new} = b + J(\bar{N} - \bar{m})$$

2d)

Let's use your formulas to learn the completely pooled estimate for the event probability. You will still assume an uninformed or VAGUE prior and thus use $a = b = 0.5$ as you did in Problem 01. It is important to remember that you are pooling **ALL** movies together to learn the pooled estimate, **NOT** just those in the focused set. You are using the focused set of movies right now for visualization purposes.

Based on your formula in Problem 2c), calculate the updated shape parameters for the 30 movies in the `df_focus` tibble. You should add two columns using `mutate()` named `anew` and `bnew`. Assign your result to the `post_df_focus_pooled` object.

```
post_df_focus_pooled <- df_focus %>%
  mutate(anew = 0.5 + nrow(df_focus)*mean(df_focus$num_events),
         bnew = 0.5 + nrow(df_focus)*(mean(df_focus$num_trials) - mean(df_focus$num_events)))
```

SOLUTION

2e)

Calculate the Posterior Mean, Posterior 0.05 Quantile, and Posterior 0.95 Quantile for each movie in `post_df_focus_pooled`. You should add 3 columns using `mutate()` named `post_avg`, `post_q05`, and `post_q95`. Assign the result to the variable `summary_post_df_focus_pooled`.

```
summary_post_df_focus_pooled <- post_df_focus_pooled %>%
  mutate(post_avg = anew/(anew+bnew),
         post_q05 = qbeta(0.05, anew, bnew),
         post_q95 = qbeta(0.95, anew, bnew))
```

SOLUTION

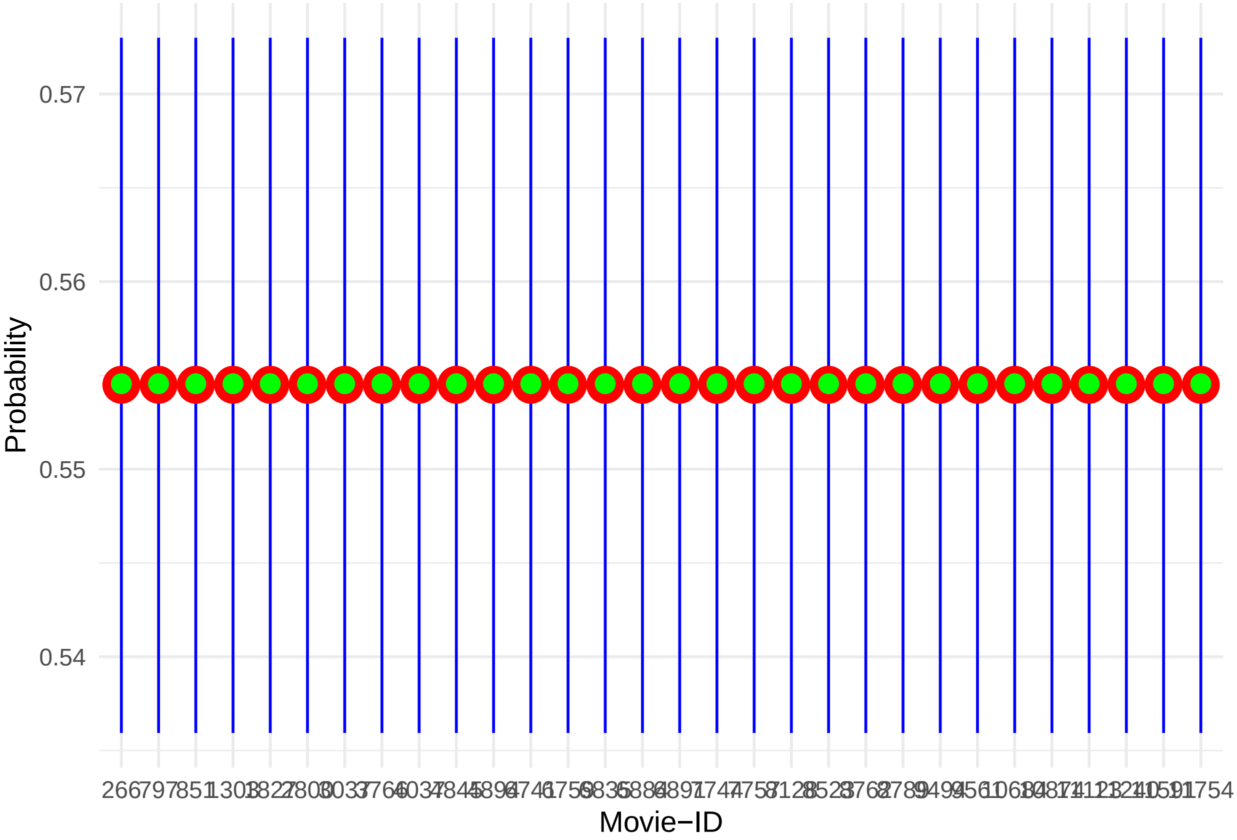
2f)

Pipe `summary_post_df_focus_pooled` into `ggplot()` and map the `x` aesthetic to `as.factor(movie_id)`. You will use the `geom_linerange()` to represent the posterior uncertainty by setting the `ymin` and `ymax` aesthetics to `post_q05` and `post_q95` respectively. You will display the posterior mean with a `geom_point()` by setting the `y` aesthetic to `post_avg`. Include the maximum likelihood estimate (MLE) on the event probability for EACH movie as an additional `geom_point()` geom by mapping the `y` aesthetic to the correct value, which you must calculate.

Are there movies with MLEs that are outside the posterior uncertainty interval? Are there movies with posterior mean values that are quite close to the MLEs?

```
ggplot(summary_post_df_focus_pooled, aes(x = as.factor(movie_id)))+
  geom_linerange(aes(ymin = post_q05, ymax = post_q95), color = "blue")+
  geom_point(aes(y=post_avg), color = "red", size=6)+
  geom_point(aes(y = (anew-1)/(anew+bnew-2)), color ="green", size=3) +
```

```r
  labs(x = "Movie-ID", y = "Probability")+
  theme_minimal()
```



SOLUTION

1) In the figure above, red dots represent posterior mean values and green dots represent the maximum likelihood estimate.
2) There movies NO movies with MLEs that are outside the posterior uncertainty interval. Also, for all movies the posterior mean values and MLEs are equal to each other.

2g)

Your visualization in Problem 2f) should not "feel right". Something should seem off.

Why does the "POOLED" estimate seem incorrect for this application?

SOLUTION In Problem 2c) we obtained that

$$a_{new} = a + J\bar{m}$$

$$b_{new} = b + J(\bar{N} - \bar{m}).$$

In other words, for all movies the distribution is a beta distribution with the same values a_{new} and b_{new}, therefore we get the same posterior results for all movies, as obtained in the figure in Problem 2f).

Problem 03

You have now worked through two extremes, the **UNPOOLED** and the completely **POOLED** estimates on the unknown event probabilities. You will now try to **BLEND** the two approaches to reach a compromise by using the Empirical Bayes approach.

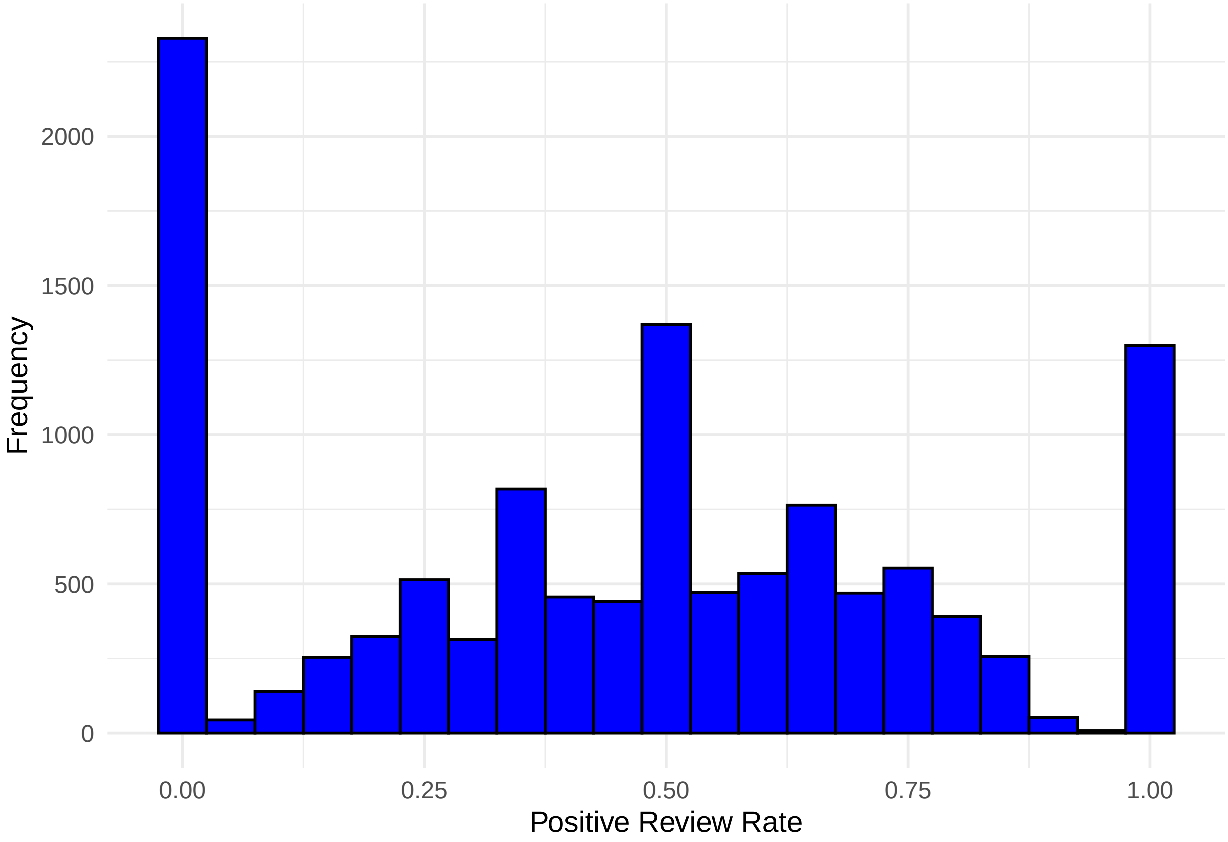
As stated at the beginning of the document, Empirical Bayes estimates the prior from data. In this setting you are interested in deciding informative values for the prior shape parameters, a and b, of the Beta prior on each μ_j. If you have a relevant informative prior you will be able to apply that prior to each movie separately (the unpooled approach) while "borrowing strength" from the rest of the data. The Empirical Bayes approach is an approximation to more formal **PARTIAL POOLING** models where groups with larger sample sizes help estimate parameters associated with small sample size groups. Empirical Bayes is useful when there are hundreds to thousands of separate groups. Estimating the prior shape parameters from many groups allows specifying relevant informative priors without requiring numerous conversations with Subject Matter Experts (SMEs) and allows the data to provide representative bounds.

3a)

The Beta prior defines the prior belief on a probability (a fraction or proportion). From an Empirical Bayes approach, you can therefore view the "data" of interest as the observed "positive review rate".

Plot the histogram of the "positive review rate" for all movies in the `df_all` data set. Use the `geom_histogram()` geom and set the `binwidth` to be 0.05.

```r
ggplot(df_all, aes(x =num_events/num_trials))+
  geom_histogram(binwidth =0.05, fill ="blue", color= "black")+
  labs(x = "Positive Review Rate", y ="Frequency")+
  theme_minimal()
```



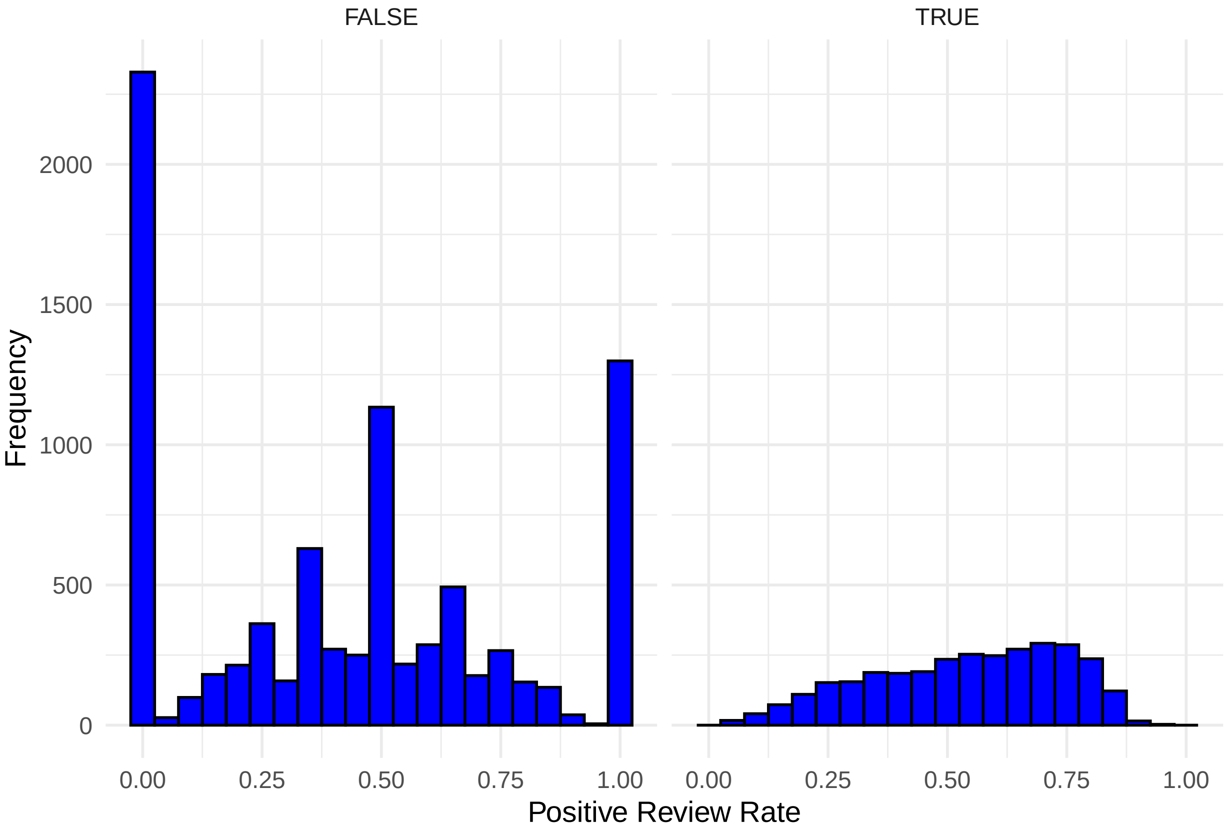
SOLUTION

3b)

Plot the histogram for all "positive review rates" in the `df_all` data set again. However, this time use `facet_wrap()` to break up the visualization into `num_trials > 30`.

What can you say about the observations of the movies with greater than 30 reviews?

```
ggplot(df_all, aes(x =num_events / num_trials))+
  geom_histogram(binwidth =0.05, fill ="blue", color = "black")+
  labs(x = "Positive Review Rate", y ="Frequency")+
  theme_minimal() +
  facet_wrap(~ num_trials>30)
```



SOLUTION

The movies with greater than 30 reviews are represented by the facet on the right hand side in the figure above. The observation rate looks smooth in this case (compared to the cases with less than 30 reviews).

3c)

To keep things simple for this exam, you will estimate the prior shape parameters, a and b, based only on the movies with greater than 30 reviews.

Use the `filter()` function to keep all movies with greater than 30 reviews and assign the result to the `df_30` object. Use the `summary()` function to check the summary stats on `num_trials` to make sure you performed the operation correctly.

```
df_30 <- df_all %>% filter(num_trials > 30)
summary(df_30$num_trials)
```

SOLUTION

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    31.0    68.0   210.0  1666.9   884.5 81491.0
```

3d)

Since the "positive review rate" is a fraction, we can use a Beta distribution as the likelihood of the "fraction" given the shape parameters. Those shape parameters, a and b, are unknown and so you must estimate them from the data. Within the Empirical Bayes approach, you will treat this step as finding a and b which **maximize the likelihood**, and so you will not specify prior distributions on the parameters.

Each observation of the "positive review rate" is assumed conditionally independent given the unknown a and b shape parameters. The observed "positive review rate" will be denoted as, θ_j, for each movie and is defined as:

$$\theta_j = \frac{m_j}{N_j}$$

The likelihood on all $j = 1, ..., J$ "positive review rates" is therefore the product of J conditionally independent Beta distributions:

$$p\left((\theta_j)_{j=1}^{J} \mid a, b \right) = \prod_{j=1}^{J} \mathrm{Beta}\left(\theta_j \mid a, b\right)$$

You will define a log-likelihood function in the style of the log-posterior functions we have used so far this semester by completing the two code chunks below.

In the first code chunk, the list of required information, `info_for_ab`, is defined and contains a single variable `theta`. You must calculate it based on the movies in the `df_30` data set.

The second code chunk defines the `my_beta_loglik()` function. The first argument, `unknowns`, is the vector of unknown parameters. The second argument, `my_info`, is the list of required information. The comments and variable names provide hints for actions you should perform to calculate the log-likelihood.

The a and b parameters are lower-bounded at zero and thus you MUST apply the NATURAL LOG-TRANSFORMATION to both parameters. You must properly account for the log-derivative adjustment on both parameters when you calculate the log-likelihood.

NOTE: Several test points are provided for you to check that you have coded your function correctly.

SOLUTION Define the list of required information. The observed data in your `my_beta_loglik()` must be named `theta`.

```r
info_for_ab <- list(
  theta = df_30$num_events/df_30$num_trials
)
```

Define the Beta log-likelihood. The first element in `unknowns` is the log-transformed a parameter and the second element is the log-transformed b parameter. **You are allowed to use built-in density functions to complete this question.**

```r
my_beta_loglik <- function(unknowns, my_info)
{
  # unpack the log-transformed shape parameters
  log_a <- unknowns[1]
  log_b <- unknowns[2]

  # back transform
  a <- exp(log_a)
  b <- exp(log_b)

  # calculate the log-likelihood for all observations
  log_lik <- sum(dbeta(my_info$theta, shape1 = a, shape2 = b,log=TRUE))

  # account for the change of variables
  deriv_adj<- log_a+log_b

  log_lik + deriv_adj
}
```

Try out values of -2 for both log-transformed parameters. If your function is coded correctly you should get a value of -3968.915.

```
unknowns <- c(-2,-2)
my_beta_loglik(unknowns, info_for_ab)
```

```
## [1] -3968.915
```

Try out values of 2.5 for both log-transformed parameters. If your function is coded correctly you should get a value of -2963.884.

```
unknowns <- c(2.5,2.5)
my_beta_loglik(unknowns, info_for_ab)
```

```
## [1] -2963.884
```

3e)

You will now identify the maximum likelihood estimates for a and b. You should use the `optim()` function to manage the optimization for you. Be sure to specify the arguments to `optim()` to make sure that `optim()` knows to *MAXIMIZE* and not *MINIMIZE* the function. Set the `method` argument to `"BFGS"` when you call `optim()`. The gradient argument should be set to `NULL`, `gr=NULL`.

Try out two different starting guesses values. The first guess, `init_guess_01`, should be zeros for both parameters and the second guess, `init_guess_02`, should be -1 for both parameters.

Assign your `optim()` results to `log_ab_opt_01` and `log_ab_opt_02`.

Do you get the same parameter estimates regardless of your initial guess?

SOLUTION Set the initial guesses.

```
init_guess_01 <- c(0, 0)
init_guess_02 <- c(-1, -1)
```

Perform the optimization using the first starting guess.

```
log_ab_res_01 <- optim(init_guess_01,
                       my_beta_loglik,
                       gr=NULL,
                       info_for_ab,
                       method = "BFGS",
                       hessian = TRUE,
                       control = list(fnscale = -1, maxit = 1001))
   log_ab_res_01
```

```
## $par
## [1] 1.0683322 0.9267191
##
## $value
## [1] 721.7789
##
## $counts
## function gradient
##       52        9
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##             [,1]      [,2]
## [1,] -5401.859  4564.349
## [2,]  4564.349 -5542.039
```

Perform the optimization using the second starting guess.

```
log_ab_res_02 <- optim(init_guess_02,
                    my_beta_loglik,
                    gr=NULL,
                    info_for_ab,
                    method = "BFGS",
                    hessian = TRUE,
                    control = list(fnscale = -1, maxit = 1001))
log_ab_res_02
```

```
## $par
## [1] 1.0683325 0.9267193
##
## $value
## [1] 721.7789
##
## $counts
## function gradient
##       47        9
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##            [,1]     [,2]
## [1,] -5401.86  4564.35
## [2,]  4564.35 -5542.04
```

Are the identified log-transformed estimates the same?

Yes, the identified log-transformed estimates the same values.

3f)

The optimal parameters in the Problem 3e) are in the log-transformed space.

You must back-transform them to calculate the estimates for the prior a and b shape hyperparameters. Assign the back-transformed parameters to `ab_emp_bayes`.

How many a-priori trials does your estimated hyperparameters represent?

```
ab_emp_bayes <- exp(log_ab_res_01$par)
ab_emp_bayes
```

SOLUTION

```
## [1] 2.910521 2.526207
```

How many a-priori trials?

The estimated hyperparameters represent 2.910521+2.526207 = 5.436728 trials.

3g)

You will now visualize the prior distribution you calculated using the Empirical Bayes approach and compare it to the histogram of the observed "positive reviews rates" for all movies with more than 30 reviews.

Complete the two code chunks below. In the first, set the `x` variable within the `prior_for_viz` `tibble` to be 1001 evenly spaced points between the minimum observed POSITIVE review rate in `df_30` and the maximum observed POSITIVE review rate in `df_30`. Pipe the result into `mutate()` and calculate the beta density using the `ab_emp_bayes` shape parameters and assign the result to the `beta_pdf` variable.

In the second code chunk, pipe the `df_30` `tibble` into `ggplot()` and map the `x` aesthetic to the observed positive review rates. Use a `geom_histogram()` geom and set the `binwidth` to be 0.05. Modify the `y` aesthetic so that way `geom_histogram()` displays the estimated density on the `y` axis instead of the count. To do so you must set `y=after_stat(density)` within `aes()`. Include a `geom_line()` geom and specify the `data` argument to be the `prior_for_viz` object and map the `x` and `y` aesthetics to `x` and `beta_pdf`, respectively. Set the `color` argument (outside the `aes()` call) to be `'red'` and the `linewidth` argument to 1.15.

How does the empirically derived prior distribution on the event probability compare to the observed histogram of the POSITIVE review rates?

IMPORTANT: If you are *not* comfortable with your `ab_emp_bayes` values, you may use `shape1=4` and `shape2=2.5`. These are **not** the correct answers, though they are in the right ballpark…

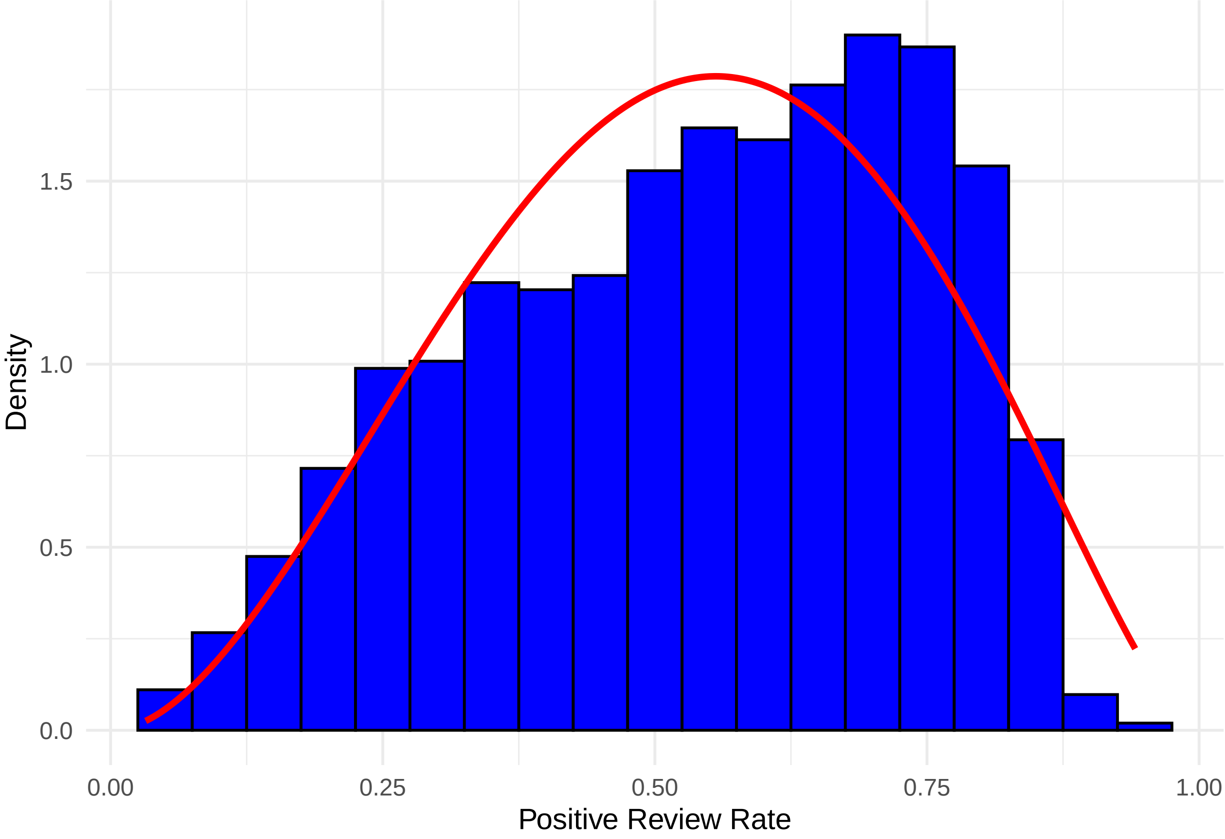
SOLUTION Calculate the Beta PDF based on the calculated prior hyperparameters.

```
prior_for_viz <- tibble::tibble(
  x = seq(min(df_30$num_events/df_30$num_trials), max(df_30$num_events/df_30$num_trials), length.out =
  mutate(beta_pdf = dbeta(x, shape1=ab_emp_bayes[1], shape2=ab_emp_bayes[2]))
```

Visualize the derived prior relative to the observed "POSITIVE review rates" in the data set.

```
ggplot(data = df_30, aes(x =num_events/num_trials)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 0.05, fill = "blue", color = "black") +
  geom_line(data = prior_for_viz, aes(x = x, y = beta_pdf), color = "red", size = 1.15) +
  labs(x = "Positive Review Rate", y = "Density") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



3h)

Calculate the Mean, 0.05 Quantile, 0.95 Quantile, and middle 90% uncertainty interval associated with your informative prior.

IMPORTANT: If you are *not* comfortable with your `ab_emp_bayes` values, you may use `shape1=4` and `shape2=2.5`. These are **not** the correct answers, though they are in the right ballpark…

SOLUTION The mean is $\frac{a}{a+b} = \frac{2.910521}{2.910521+2.526207} = 0.5353442$.

For the 0.05 Quantile and 0.95 Quantile, we can use:

```
q05_emp <- qbeta(0.05, shape1=ab_emp_bayes[1], shape2=ab_emp_bayes[2])
q95_emp <- qbeta(0.95, shape1=ab_emp_bayes[1], shape2=ab_emp_bayes[2])
q05_emp
```

```
## [1] 0.2046878
```

```
q95_emp
```

```
## [1] 0.8500796
```

So, we obtain that "0.05 Quantile = 0.2046878", and "0.95 Quantile = 0.8500796".

Finally, the middle 90% uncertainty interval is [0.2046878,0.8500796].

3i)

How do the Mean and middle 90% uncertainty intervals compare between the original uninformed prior and the Empirical Bayes derived informative prior?

SOLUTION

1) Mean of the original uninformed prior was 0.5, and mean of the Empirical Bayes derived informative prior is 0.5353442.
2) 90% uncertainty interval of the original uninformed prior was [0.00615583,0.9938442], and 90% uncertainty interval of the Empirical Bayes derived informative prior is [0.2046878,0.8500796]. So the uncertainty interval shrinks in the case of Empirical Bayes derived informative prior.

Problem 04

You now have everything in place to calculate the posterior on the event probability associated with each movie, μ_j. The a and b parameters that you had originally set to 0.5, are now equal to your Empirical Bayes estimated values.

If you are not comfortable with your estimates you may use the same values as in Problem 3g) of `shape1=4` and `shape2=2.5`.

4a)

Calculate the updated or new shape parameters for the movies in the `df_focus` tibble. You should add two columns using `mutate()` named `anew` and `bnew`. Assign your result to the `post_df_focus_empbayes` object.

```
post_df_focus_empbayes <- df_focus %>%
  mutate(anew = ab_emp_bayes[1] + num_events,
         bnew = ab_emp_bayes[2] + (num_trials - num_events))
```

SOLUTION

4b)

Calculate the posterior Mean, Posterior 0.05 Quantile, and Posterior 0.95 Quantile for each movie in `post_df_focus_empbayes`. You should add 3 columns using `mutate()` named `post_avg`, `post_q05`, and `post_q95`. Assign the result to the variable `summary_post_df_focus_empbayes`.

```
summary_post_df_focus_empbayes <- post_df_focus_empbayes %>%
  mutate(post_avg = anew/(anew+bnew),
         post_q05 = qbeta(0.05, anew, bnew),
         post_q95 = qbeta(0.95, anew, bnew))
```

SOLUTION

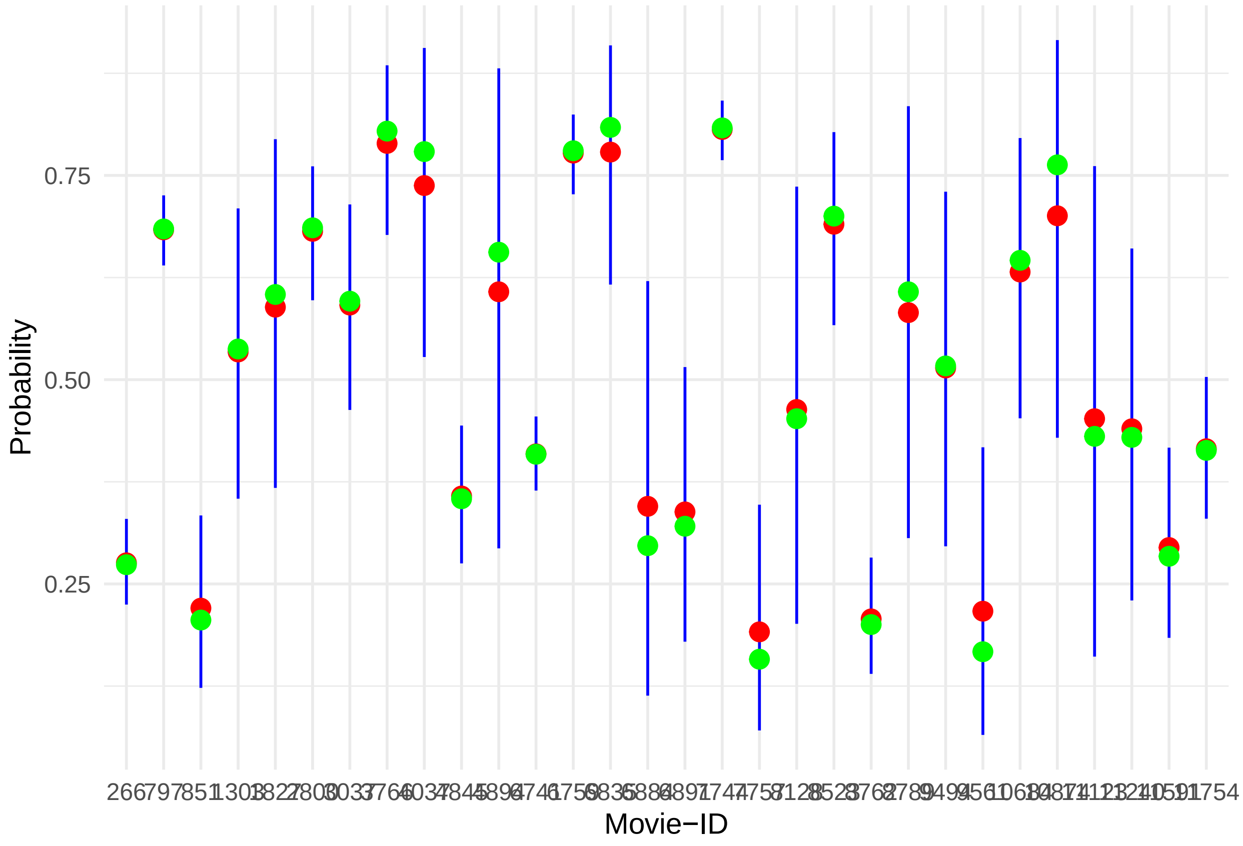
4c)

You will repeat the visualizations from Problem 1) to understand the effect of your informative prior distribution.

Pipe `summary_post_df_focus_empbayes` into `ggplot()` and map the `x` aesthetic to `as.factor(movie_id)`. You will use the `geom_linerange()` to represent the posterior uncertainty by setting the `ymin` and `ymax` aesthetics to `post_q05` and `post_q95` respectively. You will display the posterior mean with a `geom_point()` by setting the `y` aesthetic to `post_avg`. Include the maximum likelihood estimate (MLE) on the event probability as an additional `geom_point()` geom by mapping the `y` aesthetic to the correct value, which you must calculate.

How does this visualization compare to those you made using the vague unpooled estimate and the completely pooled estimate?

```r
ggplot(summary_post_df_focus_empbayes, aes(x = as.factor(movie_id)))+
  geom_linerange(aes(ymin = post_q05, ymax = post_q95), color = "blue")+
  geom_point(aes(y=post_avg), color = "red", size=3)+
  geom_point(aes(y = (anew-1)/(anew+bnew-2)), color ="green", size=3) +
  labs(x = "Movie-ID", y = "Probability")+
  theme_minimal()
```



SOLUTION

1) In the figure above, red dots represent posterior mean values and green dots represent the maximum likelihood estimate.

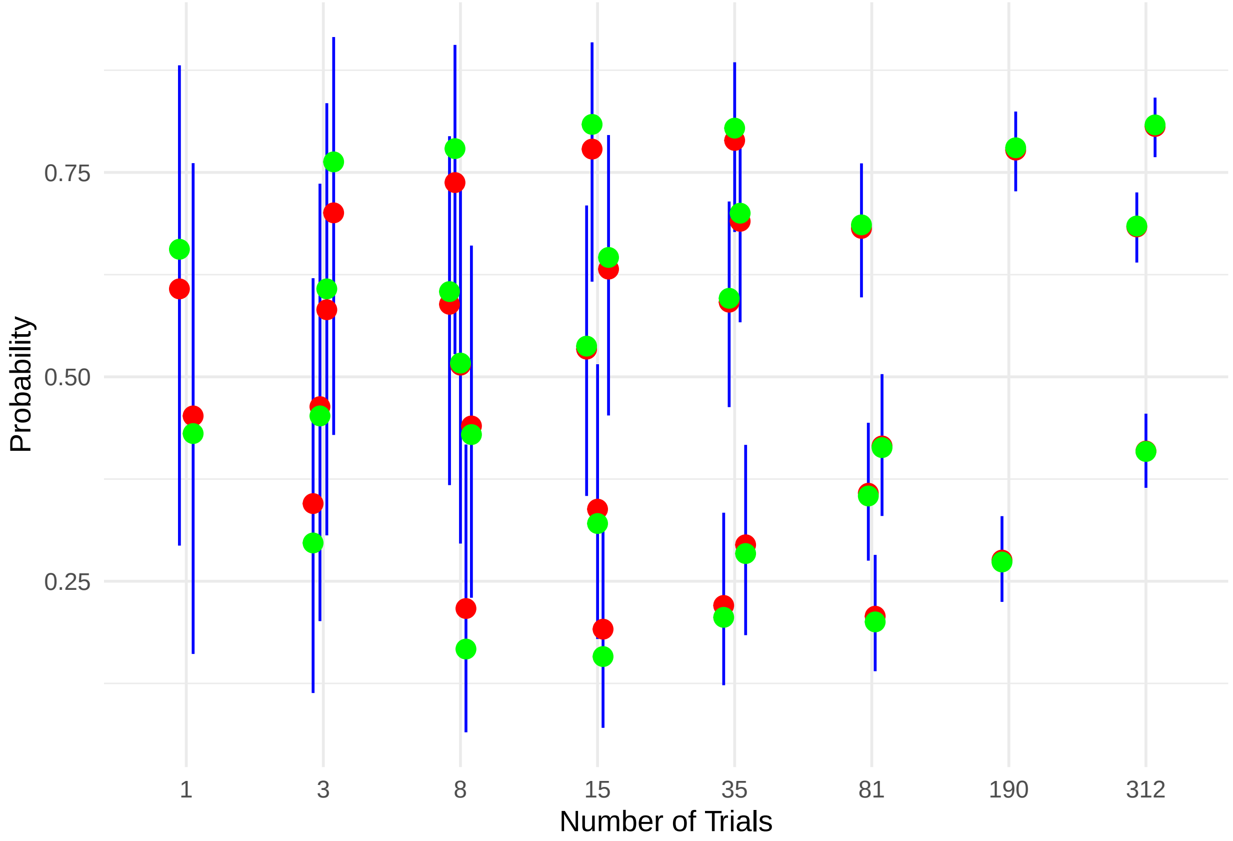
2) Compared to the previous 2 cases (the vague unpooled estimate and the completely pooled estimate), the uncertainty intervals shrink for some movies. Also, the values for the posterior mean and the maximum likelihood estimate are closer to each other.

4d)

You will create a similar visualization, except instead of mapping the x aesthetic to `as.factor(movie_id)` you will map the x aesthetic to `as.factor(num_trials)`. You must also map the `group` aesthetic in each geom to the `movie_id` variable. Doing so allows you "dodge" the posterior summaries for each movie associated with each `num_trials` value.

To properly apply the dodging, set the `position` argument to be `position = position_dodge(0.2)` in `geom_linerange()` and both `geom_point()` calls. You should not place `position` inside `aes()`, it should be outside `aes()`.

```r
ggplot(summary_post_df_focus_empbayes, aes(x = as.factor(num_trials), group = movie_id)) +
  geom_linerange(aes(ymin = post_q05, ymax = post_q95), color="blue", position = position_dodge(0.2)) +
  geom_point(aes(y = post_avg), color = "red", size =3, position = position_dodge(0.2)) +
  geom_point(aes(y = (anew-1)/(anew+bnew-2)), color ="green", size =3, position = position_dodge(0.2)) +
  labs(x = "Number of Trials", y = "Probability") +
  theme_minimal()
```



SOLUTION

4e)

You will now calculate the posteriors for **ALL** movies using the Empirical Bayes approach, not just the limited number of movies in the "focused" data set.

Calculate the updated shape parameters for all movies in the `df_all` tibble. You should add two columns using `mutate()` named `anew` and `bnew`. Assign your result to the `post_df_all_empbayes` object.

```r
post_df_all_empbayes <- df_all %>%
  mutate(anew = ab_emp_bayes[1] + num_events,
         bnew = ab_emp_bayes[2] + (num_trials - num_events))
```

SOLUTION

4f)

Calculate the Posterior Mean, Posterior 0.05 Quantile, and Posterior 0.95 Quantile for each movie in `post_df_all_empbayes`. You should add 3 columns using `mutate()` named `post_avg`, `post_q05`, and `post_q95`. Assign the result to the variable `summary_post_df_all_empbayes`.

```r
summary_post_df_all_empbayes <- post_df_all_empbayes %>%
  mutate(post_avg = anew/(anew+bnew),
         post_q05 = qbeta(0.05, anew, bnew),
         post_q95 = qbeta(0.95, anew, bnew))
```

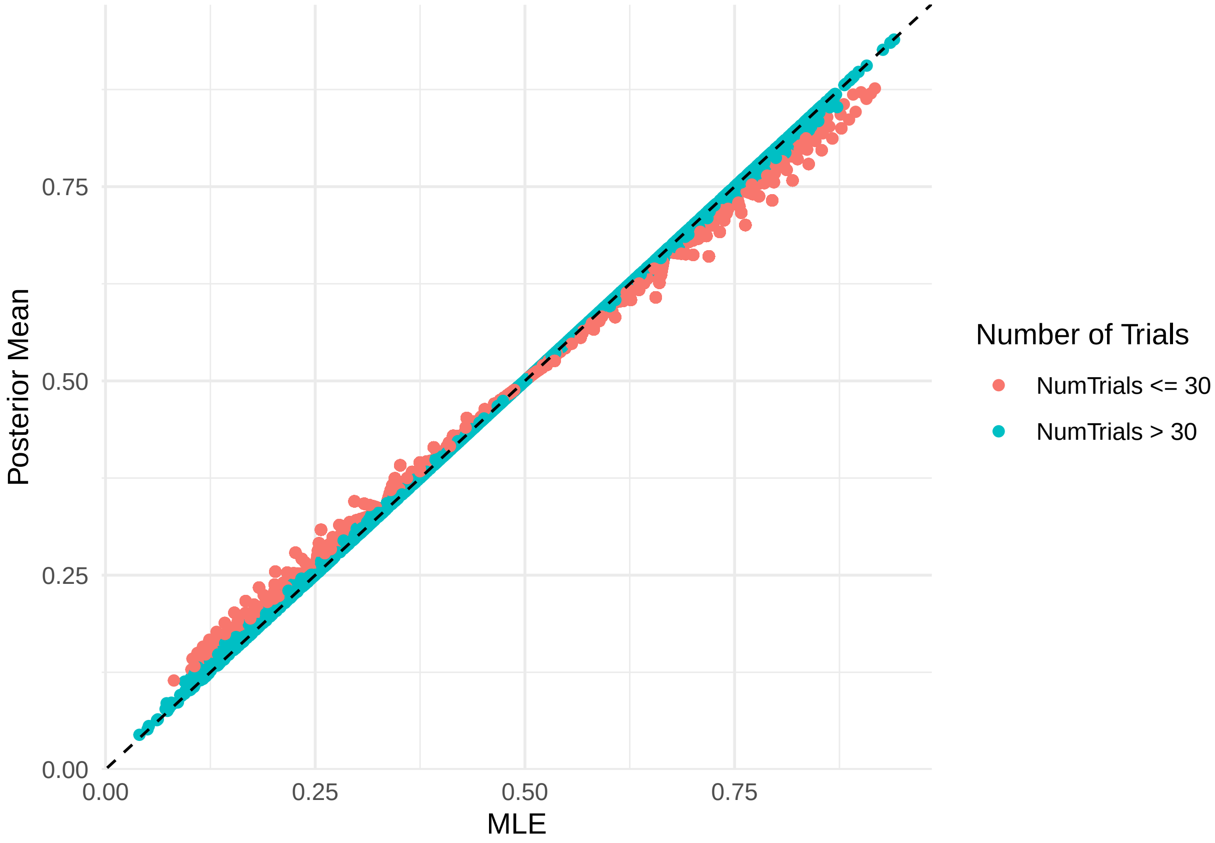
SOLUTION

4g)

You will now visualize the Posterior Mean, based on the Empirical Bayes informative prior, relative to the Maximum Likelihood Estimate for the event probability.

Create a scatter plot with `ggplot2` where you plot the `post_mean` with respect to the maximum likelihood estimate to the unknown event probability for all movies. Map the `color` aesthetic to the conditional test `num_trials > 30` and include a `geom_abline()` layer with `slope = 1` and `intercept=0`. You are therefore coloring by a discrete or binarized version of the `num_trials` rather than the continuous value.

```r
ggplot(summary_post_df_all_empbayes, aes(x = (anew-1)/(anew+bnew-2), y = post_avg, color = ifelse(num_ti
  geom_point() +
  geom_abline(intercept =0, slope =1,linetype = "dashed")+
  labs(x="MLE", y = "Posterior Mean", color = "Number of Trials")+
  theme_minimal()
```

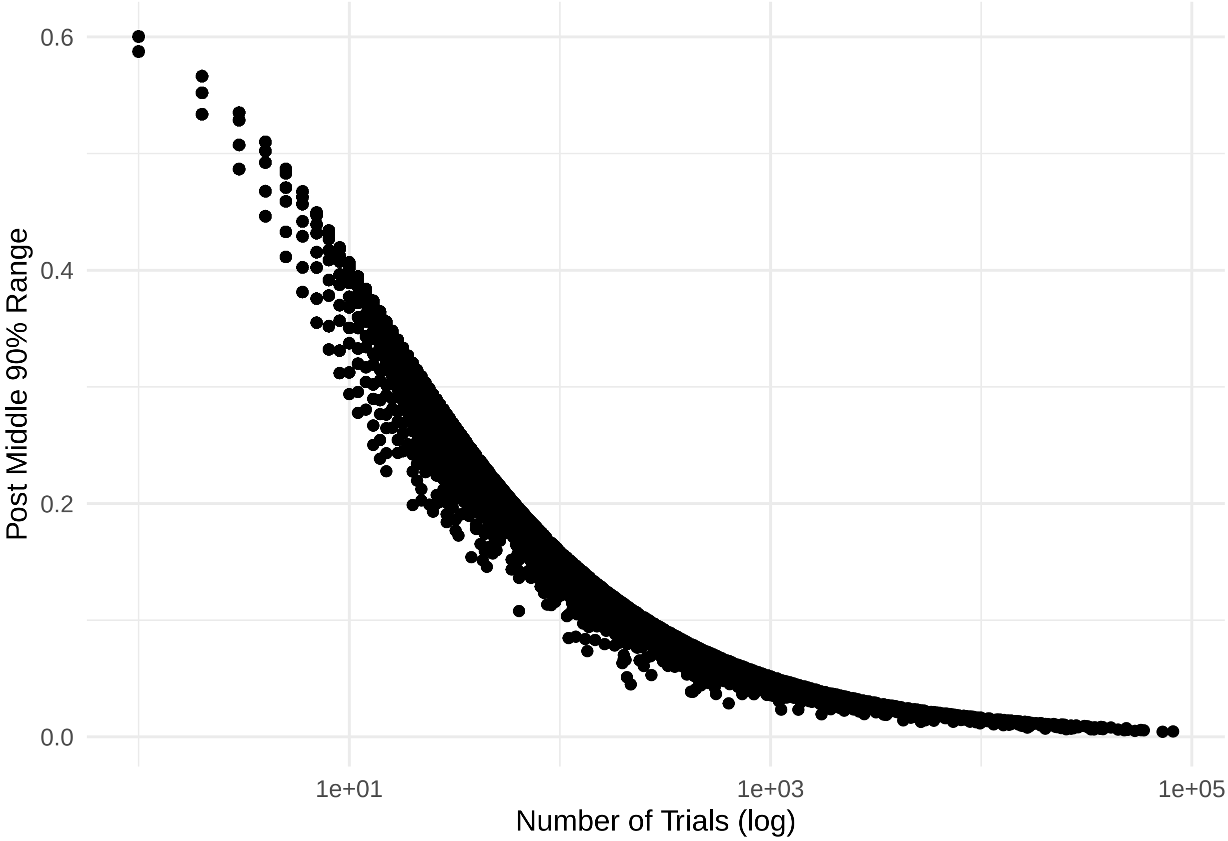


SOLUTION

4h)

Create a scatter plot for the Posterior middle 90% uncertainty interval range (difference between the 0.95 and 0.05 Quantiles) with respect to the `num_trials` using `ggplot2`. Include the `scale_x_log()` layer to transform the `x` aesthetic scale via the `log10()` function.

```r
ggplot(data = summary_post_df_all_empbayes, aes(x =num_trials, y=post_q95-post_q05)) +
  geom_point()+
  scale_x_log10()+
  labs(x="Number of Trials (log)", y="Post Middle 90% Range")+
  theme_minimal()
```



SOLUTION

4i)

The code chunk below vertically concatenates (binds) the `summary_post_df_all_from_vague` and `summary_post_df_all_empbayes` `tibbles` into a single `tibble` for you. This new `tibble`, `compare_post_summaries` includes a new column, `from_prior`, which denotes if the prior was the original uninformed prior or the Empirical Bayes informative prior.

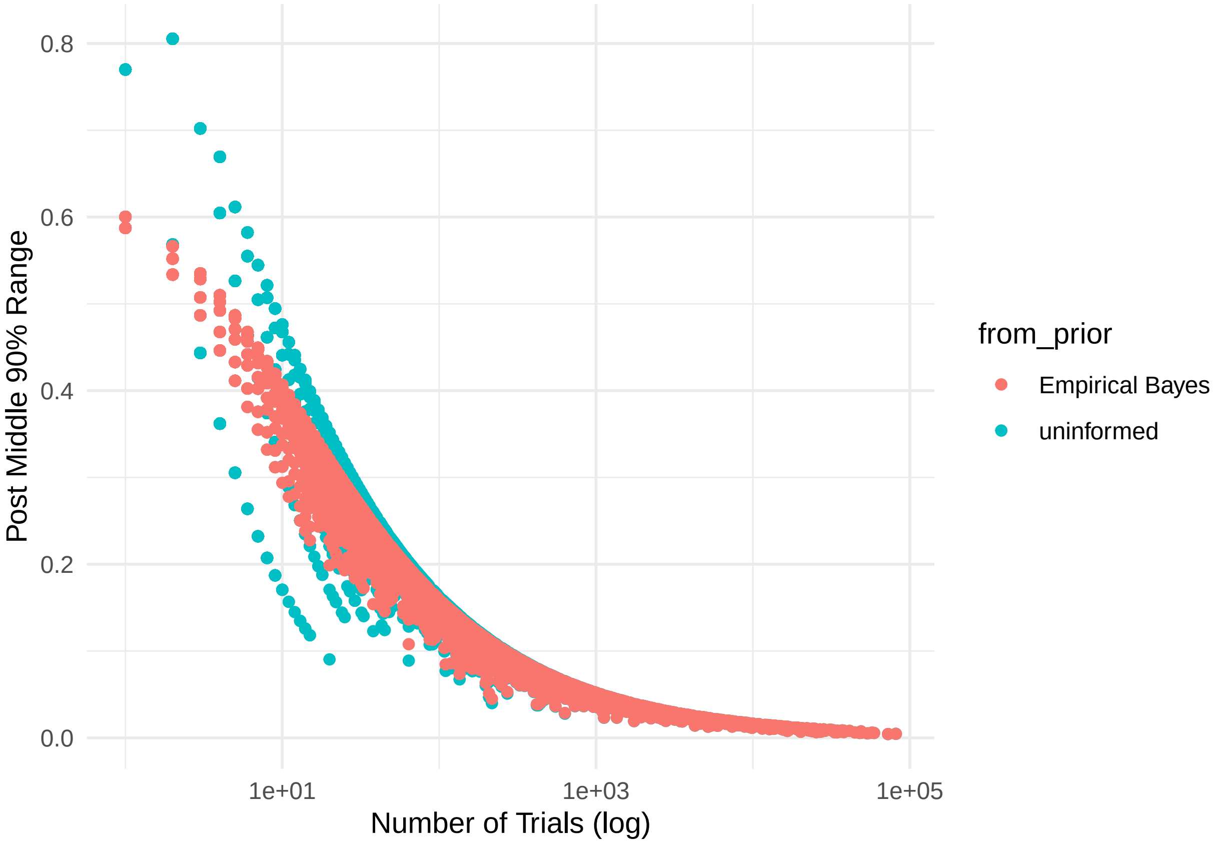
```
compare_post_summaries <- summary_post_df_all_from_vague %>%
  mutate(from_prior = 'uninformed') %>%
  bind_rows(summary_post_df_all_empbayes %>%
              mutate(from_prior = 'Empirical Bayes'))

compare_post_summaries %>% glimpse()
```

```
## Rows: 23,602
## Columns: 9
## $ movie_id   <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ~
## $ num_trials <dbl> 8, 1, 23, 79, 1, 5, 4, 2, 1, 120, 2, 4, 1, 62, 43, 9, 2, 74~
## $ num_events <dbl> 5, 0, 4, 33, 0, 5, 2, 1, 0, 69, 0, 2, 0, 47, 27, 5, 0, 61, ~
## $ anew       <dbl> 5.5, 0.5, 4.5, 33.5, 0.5, 5.5, 2.5, 1.5, 0.5, 69.5, 0.5, 2.~
## $ bnew       <dbl> 3.5, 1.5, 19.5, 46.5, 1.5, 0.5, 2.5, 1.5, 1.5, 51.5, 2.5, 2~
## $ post_avg   <dbl> 0.6111111, 0.2500000, 0.1875000, 0.4187500, 0.2500000, 0.91~
## $ post_q05   <dbl> 0.3428253101, 0.0015429193, 0.0754237349, 0.3296846704, 0.0~
## $ post_q95   <dbl> 0.8498930, 0.7714802, 0.3296132, 0.5101402, 0.7714802, 0.99~
## $ from_prior <chr> "uninformed", "uninformed", "uninformed", "uninformed", "un~
```

You will use this “composite” `tibble` to examine the sensitivity of the Posterior to the prior choice and sample size.

Create a scatter plot for the Posterior middle 90% uncertainty interval range (difference between the 0.95 and 0.05 Quantiles) with respect to the `num_trials` using `ggplot2`. This plot will be similar to the one from 4h) except this time you must use the `compare_post_summaries` tibble instead of `summary_post_df_all_empbayes`. You must map the `color` aesthetic to the `from_prior` variable within the scatter plot. Include the `scale_x_log()` layer to transform the `x` aesthetic scale via the `log10()` function.

```
ggplot(data = compare_post_summaries, aes(x =num_trials, y=post_q95-post_q05,color = from_prior)) +
  geom_point()+
  scale_x_log10()+
  labs(x="Number of Trials (log)", y="Post Middle 90% Range")+
  theme_minimal()
```



SOLUTION

4j)

Based on your visualizations in this exam, discuss how an informative prior influences the Posterior when the sample size is small compared with large sample sizes.

SOLUTION

1) When the sample size is small, informative prior shrinks the "Post Middle 90% Range". However, when the sample size is large, the informative prior does not have much affect on the "Post Middle 90% Range". This can be observed on the figure in Problem 4j).

2) When the sample size is large, Posterior Mean and MLE values which are obtained using the informative prior are close to each other. However, when the sample size is large, Posterior Mean and MLE values

which are obtained using the informative prior are generally not very close. This can be observed on the figure in Problem 4g).

Problem 05

Let's now use the posterior distribution to answer questions typically associated with this application. It is time to RECOMMEND movies! We will primarily focus on the DIFFERENCES between non-Bayesian recommendations with those provided from the informative Empirical Bayesian prior. This way you can gain practical experience with WHY the Bayesian approach is useful!

5a)

Display the BEST 10 movies according to the MLE for the POSITIVE review probability.

What are the trial sizes associated with the BEST 10 movies?

```
df_all_MLE <- df_all %>%
  mutate(MLE_positive_prob = num_events/num_trials)

df_all_MLE_sorted <- df_all_MLE %>%
  arrange(desc(MLE_positive_prob))

top_10_movies <- df_all_MLE_sorted %>%
  head(10)

best_10_movies<-top_10_movies %>%
  select(Movie_ID= movie_id, MLE_Probability = MLE_positive_prob, Trial_Size = num_trials)

best_10_movies
```

SOLUTION

```
## # A tibble: 10 x 3
##    Movie_ID MLE_Probability Trial_Size
##       <dbl>           <dbl>      <dbl>
##  1        6               1          5
##  2       27               1          1
##  3       38               1          2
##  4       94               1          2
##  5      124               1          1
##  6      133               1          1
##  7      142               1          3
##  8      160               1          1
##  9      162               1          1
## 10      183               1          3
```

The trial sizes can be seen in the last column above.

5b)

Display the WORST 10 movies according to the MLE for the POSITIVE review probability.

What are the trial sizes associated with the WORST 10 movies?

```
worst_10_movies <- df_all_MLE_sorted %>%
  tail(10)

worst_10_movies <- worst_10_movies %>%
  select(Movie_ID = movie_id, MLE_Probability =MLE_positive_prob,Trial_Size =num_trials)

worst_10_movies
```

SOLUTION

```
## # A tibble: 10 x 3
##    Movie_ID MLE_Probability Trial_Size
##       <dbl>           <dbl>      <dbl>
##  1    11732               0          2
##  2    11737               0          1
##  3    11738               0          5
##  4    11739               0          2
##  5    11747               0          3
##  6    11749               0          1
##  7    11767               0          1
##  8    11775               0          1
##  9    11776               0          1
## 10    11797               0          1
```

5c)

Display the BEST 10 movies according to the posterior mean associated with the informative Empirical Bayes prior.

What are the trial sizes associated with the BEST 10 movies?

```
df_all_EMP_sorted <- summary_post_df_all_empbayes %>%
  arrange(desc(post_avg))

top_10_movies_EMP <- df_all_EMP_sorted %>%
  head(10)

best_10_movies_EMP <- top_10_movies_EMP %>%
  select(Movie_ID = movie_id, Posterior_Mean = post_avg, Trial_Size = num_trials)

best_10_movies_EMP
```

SOLUTION

```
## # A tibble: 10 x 3
##    Movie_ID Posterior_Mean Trial_Size
##       <dbl>          <dbl>      <dbl>
##  1     8975          0.939       1124
##  2     2943          0.935       1747
##  3    10170          0.926       1356
##  4    10298          0.906        467
##  5     3812          0.898      20162
##  6     8489          0.892      16569
##  7     2312          0.890       2786
##  8     2100          0.888      25343
## 9      2214           0.888       5404
## 10      524           0.887       1931
```

The trial sizes can be seen in the last column above.

5d)

Display the WORST 10 movies according to the posterior mean associated with the informative Empirical Bayes prior.

What are the trial sizes associated with the WORST 10 movies?

```
worst_10_movies_EMP <- df_all_EMP_sorted %>%
  tail(10)

worst_10_movies_EMP <- worst_10_movies_EMP %>%
  select(Movie_ID = movie_id, Posterior_Mean = post_avg, Trial_Size = num_trials)

worst_10_movies_EMP
```

SOLUTION

```
## # A tibble: 10 x 3
##    Movie_ID Posterior_Mean Trial_Size
##       <dbl>          <dbl>      <dbl>
##  1     9430         0.0831        198
##  2     3031         0.0790        272
##  3    11333         0.0777        135
##  4    10402         0.0758        442
##  5    10991         0.0753        551
##  6     1960         0.0644        428
##  7    10569         0.0634        419
##  8     5587         0.0558        208
##  9     2206         0.0515        633
## 10     3800         0.0446        217
```

The trial sizes can be seen in the last column above.

5e)

What are the primary differences between the BAYESIAN and NON-BAYESIAN recommendations?

SOLUTION

1) Bayesian methods explicitly include prior information in decision-making. Non-Bayesian methods typically do not incorporate prior beliefs.
2) Bayesian recommendation provides probabilistic recommendations with credible intervals. Non-Bayesian method does not explicitly model uncertainty (in this case).
3) Bayesian methods provide parameter estimates as posterior distributions. Non-Bayesian methods typically provide point estimates.
4) As we can see in Problems 5a) and 5b), Non-Bayesian method recommend movies with small trial sizes. However, as it can be seen in Problems 5c) and 5d), Bayesian method recommend movies with large trial sizes.

5f)

Movies with many, many reviews (trials) mean the movies are well known. Let's now consider recommendations associated with movies with 15 reviews. Fewer reviews may represent fewer people have watched the movie. We therefore may want to RECOMMEND less widely known movies to our customers to **surprise** them.

Filter the `summary_post_df_all_empbayes` object use the `filter()` function to keep all movies with 15 reviews. Assign the result to the `df_15` object. Use the `summary()` function to check the summary stats on `num_trials` to make sure you performed the operation correctly.

```
df_15 <- summary_post_df_all_empbayes %>%
  filter(num_trials == 15)

summary(df_15$num_trials)
```

SOLUTION

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      15      15      15      15      15      15 
```

5g)

Let's ask a probabilistic question to compare the movies with just 15 reviews! Specifically, you will examine the posterior probability that the movies with 15 reviews have POSITIVE review probabilities greater than 0.627.

Calculate the posterior probability that each movie in `df_15` has a POSITIVE review probability greater than 0.53. Add a column to `df_15` named `prob_grt_62.7`. Assign the result to the `post_movie_eval` object.

```
post_movie_eval <- df_15 %>%
  mutate(prob_grt_62.7 = pbeta(0.627, shape1=anew, shape2=bnew, lower.tail = FALSE))
```

SOLUTION

5h)

Display the BEST 10 movies with 15 reviews based on the posterior probability that their POSITIVE review probability is greater than 0.627.

```
post_movie_eval_sorted <- post_movie_eval %>%
  arrange(desc(prob_grt_62.7))

top_10_movies_grt_62.7 <- post_movie_eval_sorted %>%
  head(10)

best_10_movies_grt_62.7 <- top_10_movies_grt_62.7 %>%
  select(Movie_ID = movie_id, Prob_Grt_62.7 = prob_grt_62.7, Trial_Size = num_trials)

best_10_movies_grt_62.7
```

SOLUTION

```
## # A tibble: 10 x 3
##    Movie_ID Prob_Grt_62.7 Trial_Size
##       <dbl>         <dbl>      <dbl>
##  1     9507         0.996         15
##  2     1419         0.981         15
##  3     1757         0.940         15
##  4     6835         0.940         15
##  5     8633         0.940         15
##  6     9389         0.940         15
##  7     1884         0.852         15
##  8     5067         0.852         15
##  9     6479         0.852         15
## 10    10708         0.852         15
```

5i)

Why do you think Problem 5g) was focused on calculating the probability that the POSITIVE review probability is greater than 0.627? What is the interpretation of such a question?

HINT: Consider the interpretation of the completely pooled estimate.

SOLUTION

1) Problem 5g) focuses on calculating the probability that the POSITIVE review probability is greater than 0.627 to assess the probability of movies having high positive review rates.
2) The probability of exceeding 0.627 is an indicator of the confidence in individual movie estimates. For example, a high probability of exceeding 0.627 suggests that the model believes that many of these movies indeed have a high positive review rate, even when there is limited data available (which is 15 in this case).
3) This probability helps us understand the level of confidence we have in the success probability exceeding a specified threshold, which can have implications for decision-making in recommendations.