INFSCI 2595 - MIDTERM

Vedant Kansara

Collaborators You are **NOT** allowed to collaborate within anyone. Collaboration, copying, and/or cheating of any kind will not be tolerated.

Overview

This midterm 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 on the exam.

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.

Load packages

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     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     1.0.1
## -- 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 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 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 Sports related application. You are interested in learning the catch probability (or catch rate) in the National Football League (NFL). The catch rate is defined as the number of successful receptions (catches) by a player divided by the number of targets (a target corresponds to a pass thrown at the player). You can therefore consider successfully catching a pass as the **event**, and the number of times the player was targeted as the number of **trials**. The probability of catching a pass is therefore the **event probability** we are interested in learning.

Let's consider you are working on this application because you were recently hired as a sports analytics intern for an NFL team. You are provided with 3 seasons worth of data (2018, 2019, and 2020) of every player with at least 1 target (thus at least 1 trial). Calculating the catch rate is simple to do. It is also easy to search for and find. For example, here are the catch rates for all NFL players in the 2019 season. You were hired because the NFL team wishes to move away from simple *point estimates*. The team wants to have a better understanding of the **uncertainty** in the performance. Understanding the uncertainty is critical when evaluating talent, and making decisions for which players to sign in free agency.

You will work with two datasets for this exam. Both are loaded for you in code chunk 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 23 players to help with visualization and discussion in the exam.

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

```
## Rows: 849 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): player_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.
```

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

```
## Rows: 23 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): player_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, `player_id`, `num_events`, and `num_trials`. The `num_events` is the number of receptions, and `num_trials` is the number of targets (therefore written in the general terms that we have used in the class). The `player_id` variable is an ID variable for each player. Thus, one row in either data set tells us the number of receptions and number of targets associated with an individual player over the three seasons. The data in this exam are real and were downloaded from the `nflfastR` package (documentation available here if you are interested). The `player_id` variable is an anonymous identification number I created so that NFL fans cannot easily tell identify players.

1a)

To help understand why Empirical Bayes is be useful, let's suppose you're not sure how to specify an informative prior for this example. Even if you watch every Pittsburgh Steelers' game, you might not know what the average catch rate is in the NFL. 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 catch rate (or event probability in general terms), μ. For generality, you will denote each player with a subscript j and the total number of players as J. Thus, the unknown event probability for the j-th player is μ_j where $j = 1, ..., J$. The posterior distribution on the j-th player's unknown catch rate, μ_j given the m_j catches (events) out of N_j targets (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 player 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 player.

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

SOLUTION The Beta prior shape parameter a represents the prior number of events and b represents the prior number of non-events. Thus, the prior number of trials is their sum, a+b. In the context of the NFL application, a is the prior number of receptions (catches) and a+b is the prior number of targets. Thus, by using a=0.5 and b=0.5 in the Beta prior we are saying that we have only seen one previous trial! We cannot rule out any values as being implausible and we do not know what typical or appropriate values are. Our uninformative prior will therefore not place any prior weight on probabilities, except those very close to 0 and very close to 1.

Essentially, our prior specification represents that we have a-priori observed only a single pass attempt in an NFL game. Our prior therefore reflects that our only experience with the NFL is a single play in a single game.

1b)

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

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

SOLUTION The Beta is the conjugate prior to the Binomial likelihood, thus the posterior will be the same distribution as the prior. The posterior will therefore be 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 player's observed number of catches m_j and observed number of targets N_j, as well as the prior shape parameters, a and b.

SOLUTION the updated or posterior shape parameters sum the prior shape parameters with the observations. The posterior a parameter, anew sums the prior number of events with the observed number of events. The posterior b parameter, bnew, sums the prior number of non-events with the observed number of non-events. Since each player has an individual set of observations, (m,N)j, each player will have an individual set of posterior parameters.

The posterior a parameter for the j-th parameter is:

$$a_{new,j} = a + m_j$$

The posterior b parameter for the j-th parameter is:

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

1d)

Based on your formula in Problem 1c), calculate the updated shape parameters for the 23 players 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

1e)

Calculate the posterior mean, 0.05 Quantile, and 0.95 Quantile for each player 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`.

SOLUTION The posterior is a beta distribution. Thus, we can calculate the quantiles using the qbeta() function. The shape1 argument will be set to the anew values and the shape2 argument will be set to the bnew values. The expected value (mean) of a Beta distribution is the a shape parameter divided by the sum of the a and b shape parameters. The formula for the posterior expected value for the j-th player can be written as:

$$\mathrm{E}\left[\mu_j \mid a_{newj}, b_{newj}\right] = \frac{a_{newj}}{a_{newj} + b_{newj}}$$

```
summary_post_df_focus_from_vague <- post_df_focus_from_vague %>%
  mutate(post_avg = anew / (anew + bnew),
         post_q05 = qbeta(0.05, shape1 = anew, shape2 = bnew),
         post_q95 = qbeta(0.95, shape1 = anew, shape2 = bnew))
```

1f)

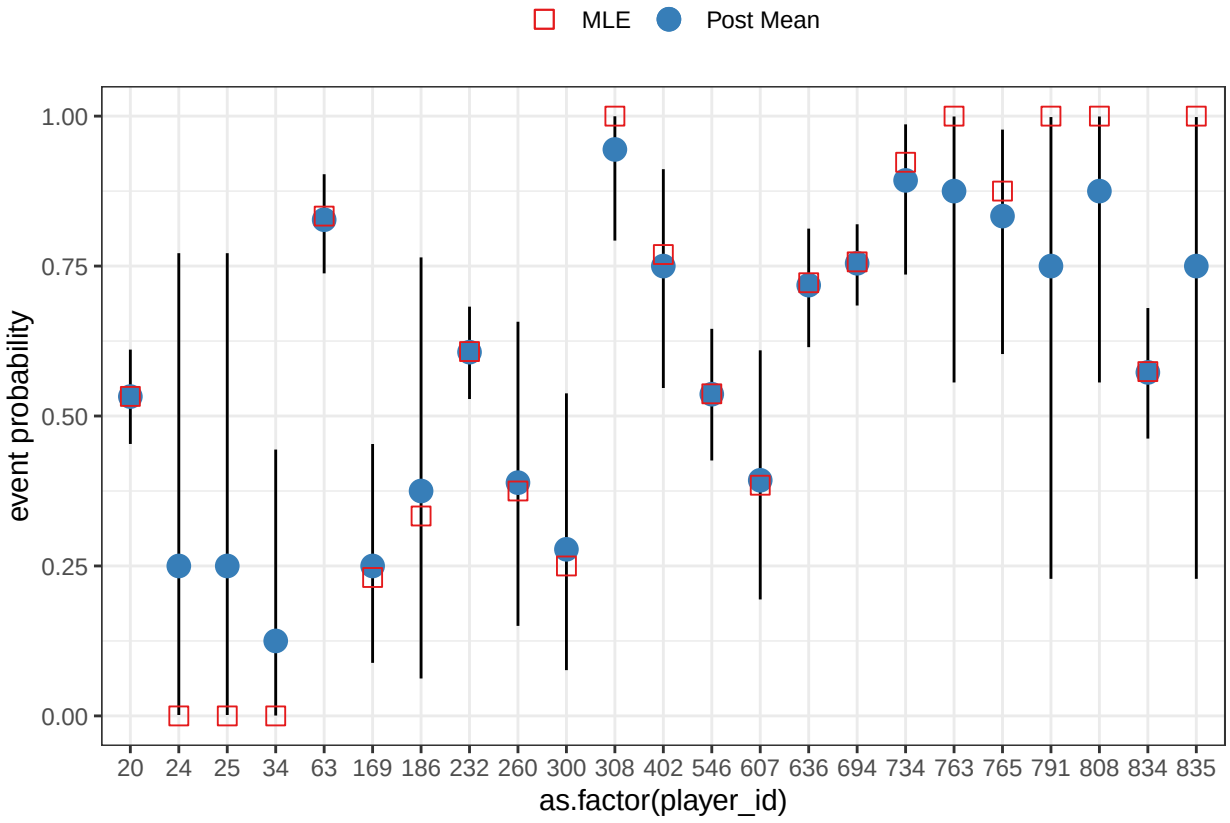
You will now visualize the posterior summaries for the 23 players associated with the `df_focus` data set.

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

SOLUTION The players with MLEs located at zero or 1 have the largest difference between the MLE and the posterior mean. The players with the smallest posterior 90% uncertainty intervals appear to have the smallest differences between the MLE and the posterior mean.

```
summary_post_df_focus_from_vague %>%
  ggplot(mapping = aes(x = as.factor(player_id))) +
  geom_linerange(mapping = aes(ymin = post_q05, ymax = post_q95)) +
  geom_point(mapping = aes(y = post_avg,
                           color = "Post Mean",
                           shape = "Post Mean",
                           size = "Post Mean")) +
  geom_point(mapping = aes(y = num_events / num_trials,
                           color = "MLE",
                           shape = "MLE",
                           size = "MLE")) +
  scale_color_brewer("", palette = "Set1") +
  scale_shape_manual("", values = c("MLE" = 0,
                                    "Post Mean" = 16)) +
  scale_size_manual("", values = c("MLE" = 3,
                                   "Post Mean" = 4)) +
  labs(y = "event probability") +
  theme_bw() +
  theme(legend.position = "top")
```



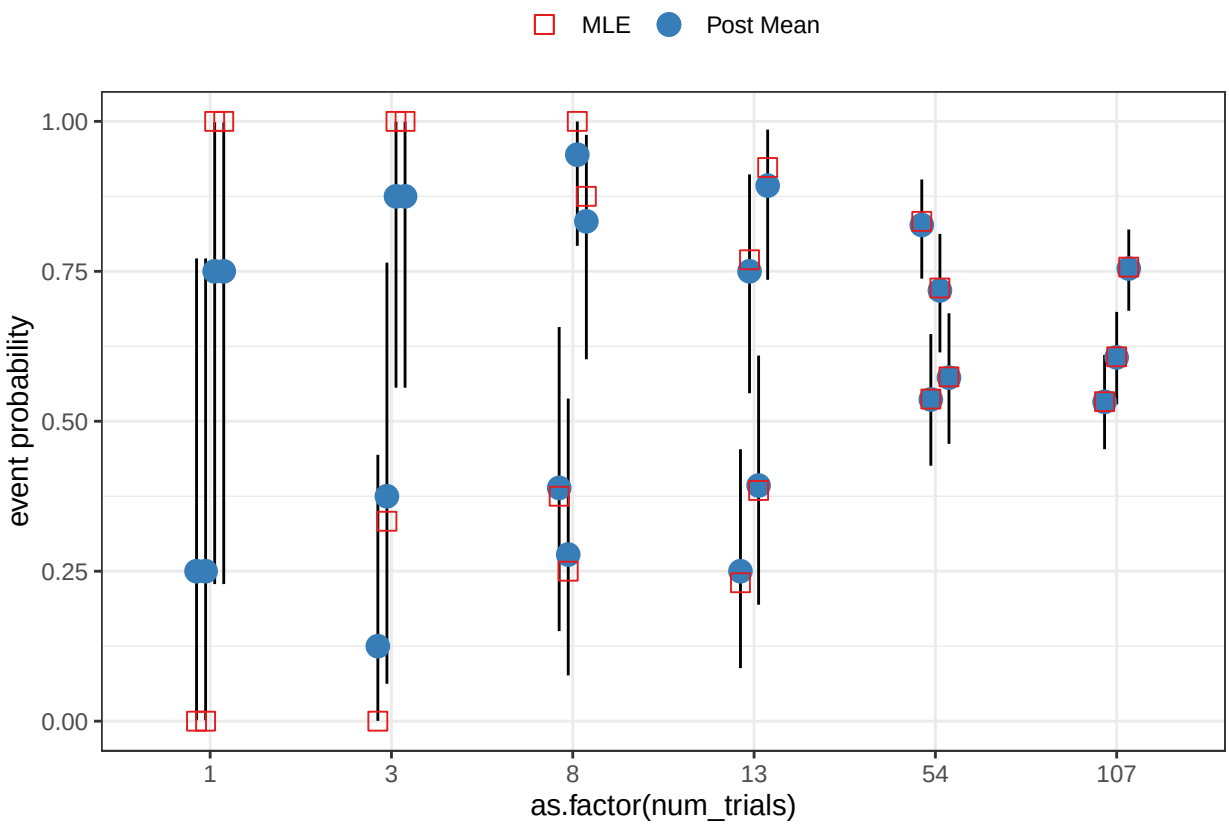
1g)

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

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

```
summary_post_df_focus_from_vague %>%
  ggplot(mapping = aes(x = as.factor(num_trials))) +
  geom_linerange(mapping = aes(ymin = post_q05, ymax = post_q95,
                               group = player_id),
                 position = position_dodge(0.2)) +
  geom_point(mapping = aes(y = post_avg,
                           color = "Post Mean",
                           shape = "Post Mean",
                           size = "Post Mean",
                           group = player_id),
             position = position_dodge(0.2)) +
  geom_point(mapping = aes(y = num_events / num_trials,
                           color = "MLE",
```

```
                          shape = "MLE",
                          size = "MLE",
                          group = player_id),
            position = position_dodge(0.2)) +
scale_color_brewer("", palette = "Set1") +
scale_shape_manual("", values = c("MLE" = 0,
                                  "Post Mean" = 16)) +
scale_size_manual("", values = c("MLE" = 3,
                                 "Post Mean" = 4)) +
labs(y = "event probability") +
theme_bw() +
theme(legend.position = "top")
```



SOLUTION

As we can see above, the posterior uncertainty intervals are decreasing as the number of trials increases. With just a single trial, the MLEs are either 0 or 1 since we observed either 0 events out of 1 trial or 1 event out of 1 trial. With an uninformative prior, we are not ruling out any values for the event probability. The 90% uncertainty intervals cover over 75% of the possible range to represent just how unsure we are about the event after just a single trial. As the number of trials increase, the posterior means and MLEs get closer together.

Problem 02

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

The opposite view point is to **completely pool** all players together in order to estimate a single unknown event probability μ. For this, you will assume that all players 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 players.

SOLUTION Add as many equation blocks as you feel are necessary to show the steps to derive the answer.

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

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

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

The log turns the product of the J Binomials into the summation of the J log-Binomials:

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

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

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

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

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 The log-likelihood's contribution to the log-posterior involves two summation series. The first summation series sums over the number of events (catches in this context). This summation can therefore be related to the average number of events by rearranging the expression for $\bar{m}$:

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

The second summation series involves summing over the number of non-events across all players. This summation series can be broken into two separate summations, as shown below.

$$\sum_{j=1}^{J}(N_j - m_j) = \sum_{j=1}^{J}(N_j) - \sum_{j=1}^{J}(m_j)$$

Rearrange the average number of trials (targets in this context), $\bar{N}$:

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

The total number of non-events across players is therefore equal to the number of players multiplied by the difference of the average number of trials and average number of events.

$$\sum_{j=1}^{J}(N_j - m_j) = \sum_{j=1}^{J}(N_j) - \sum_{j=1}^{J}(m_j) = J\bar{N} - J\bar{m} = J\left(\bar{N} - \bar{m}\right)$$

We can now substitute in for the summation series in the log-posterior. The simplified expression is given below.

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

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 hyperparameters for your specified posterior distribution.

SOLUTION The solution to Problem 2b) has the same functional form as a Beta distribution. Thus, the posterior distribution on the pooled event probability is a Beta! The posterior or updated hyperparameters are the updated shape parameters of the Beta.

The new hyperparameters for the beta posterior distribution is

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

2d)

Based on your formula in Problem 2c), calculate the updated shape parameters for the 23 players 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.

You will still assume a vague prior and thus use $a = b = 0.5$ as you did in Problem 01. And remember that we are pooling **ALL** players together to learn the pooled estimate, **not** just those in the focused set.

```
### calculate the average trial size
N_avg <- mean(df_all$num_trials)

### calculate the average number of events
m_avg <- mean(df_all$num_events)

### total number of players
J <- nrow(df_all)
```

```
post_df_focus_pooled <- df_focus %>%
  mutate(anew = J * m_avg + 0.5,
         bnew = J * (N_avg - m_avg) + 0.5)
```

```
post_df_focus_pooled %>%
  count(anew, bnew)
```

SOLUTION

```
## # A tibble: 1 x 3
##     anew   bnew     n
##    <dbl>  <dbl> <int>
## 1 36102. 17910.    23
```

2e)

Calculate the posterior mean, 0.05 Quantile, and 0.95 Quantile for each player 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, shape1 = anew, shape2 = bnew),
         post_q95 = qbeta(0.95, shape1 = anew, shape2 = bnew))
```

SOLUTION

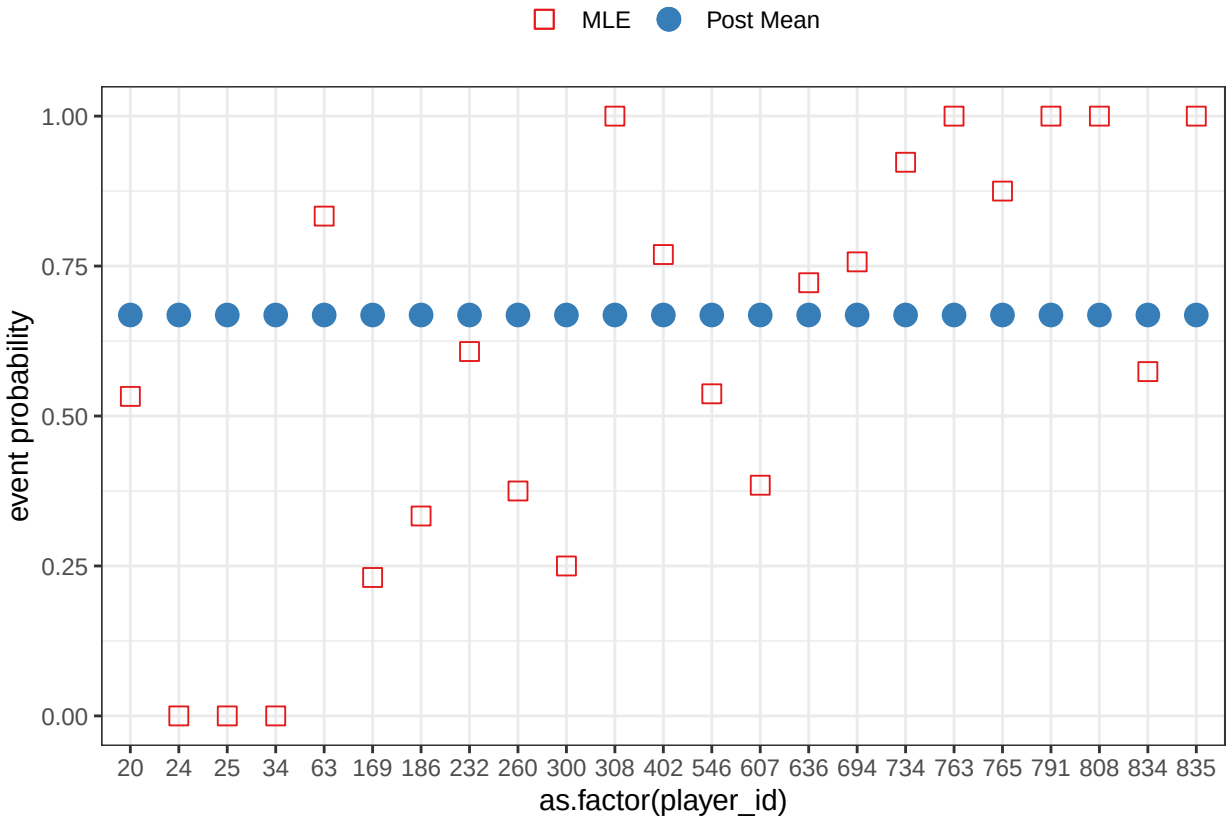
2f)

Pipe `summary_post_df_focus_pooled` into `ggplot()` and map the `x` aesthetic to `as.factor(player_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 players with MLEs that are outside the posterior uncertainty interval? Are there players with posterior mean values that are quite close to the MLEs?

SOLUTION The posterior means are the same for all players in the data set! With complete pooling, we assumed there is a constant unknown event probability. We were essentially calculating the average probability of catching a pass! The posterior uncertainty intervals are not visible in the plot below because the marker for the posterior mean is larger than the interval. The average event probability was calculated based on nearly 800 observations, and thus we are quite confident in the average estimate.

```
summary_post_df_focus_pooled %>%
  ggplot(mapping = aes(x = as.factor(player_id))) +
  geom_linerange(mapping = aes(ymin = post_q05, ymax = post_q95)) +
  geom_point(mapping = aes(y = post_avg,
                           color = "Post Mean",
                           shape = "Post Mean",
                           size = "Post Mean")) +
  geom_point(mapping = aes(y = num_events / num_trials,
                           color = "MLE",
                           shape = "MLE",
                           size = "MLE")) +
  scale_color_brewer("", palette = "Set1") +
  scale_shape_manual("", values = c("MLE" = 0,
                                    "Post Mean" = 16)) +
  scale_size_manual("", values = c("MLE" = 3,
                                   "Post Mean" = 4)) +
  labs(y = "event probability") +
  theme_bw() +
  theme(legend.position = "top")
```



2g)

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

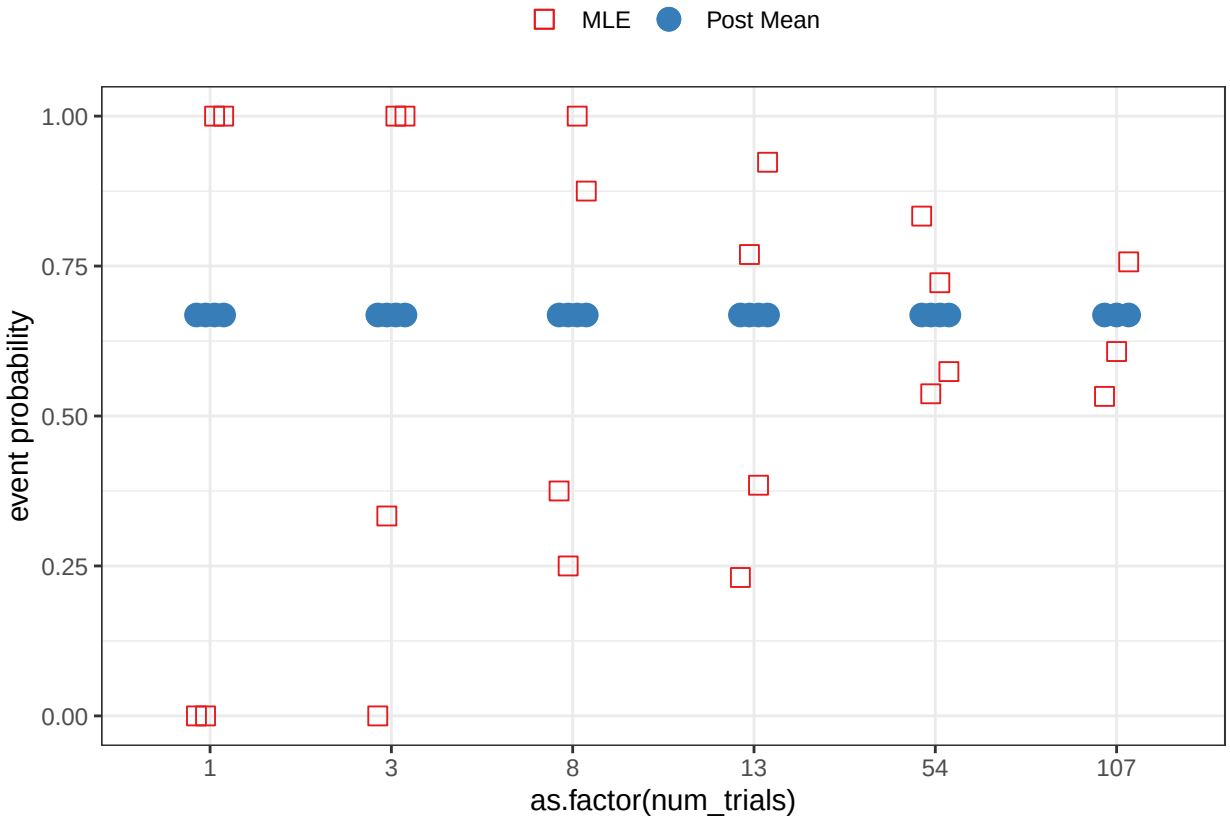
Why does the "pooled" estimate seem incorrect for this application?

SOLUTION The last figure we created in Problem 1), visualized the event probability summary statistics with respect to the number of trials. There were some differences between the players that had 107 trials (targets). Thus, we should not expect that all players have the same chance of successfully catching a pass. Some players are better than others. The complete pooling approach seems wrong because it does not allow the players to be different. It essentially thinks all players are the same!

As we can see below, the complete pooling posterior means are all the same even though the MLEs for the players at larger sample sizes are different.

```
summary_post_df_focus_pooled %>%
  ggplot(mapping = aes(x = as.factor(num_trials))) +
  geom_linerange(mapping = aes(ymin = post_q05, ymax = post_q95,
                               group = player_id),
                 position = position_dodge(0.2)) +
  geom_point(mapping = aes(y = post_avg,
                           color = "Post Mean",
                           shape = "Post Mean",
                           size = "Post Mean",
                           group = player_id),
```

```
            position = position_dodge(0.2)) +
geom_point(mapping = aes(y = num_events / num_trials,
                         color = "MLE",
                         shape = "MLE",
                         size = "MLE",
                         group = player_id),
            position = position_dodge(0.2)) +
scale_color_brewer("", palette = "Set1") +
scale_shape_manual("", values = c("MLE" = 0,
                                  "Post Mean" = 16)) +
scale_size_manual("", values = c("MLE" = 3,
                                 "Post Mean" = 4)) +
labs(y = "event probability") +
theme_bw() +
theme(legend.position = "top")
```



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 hyperparameters, 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 player

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.

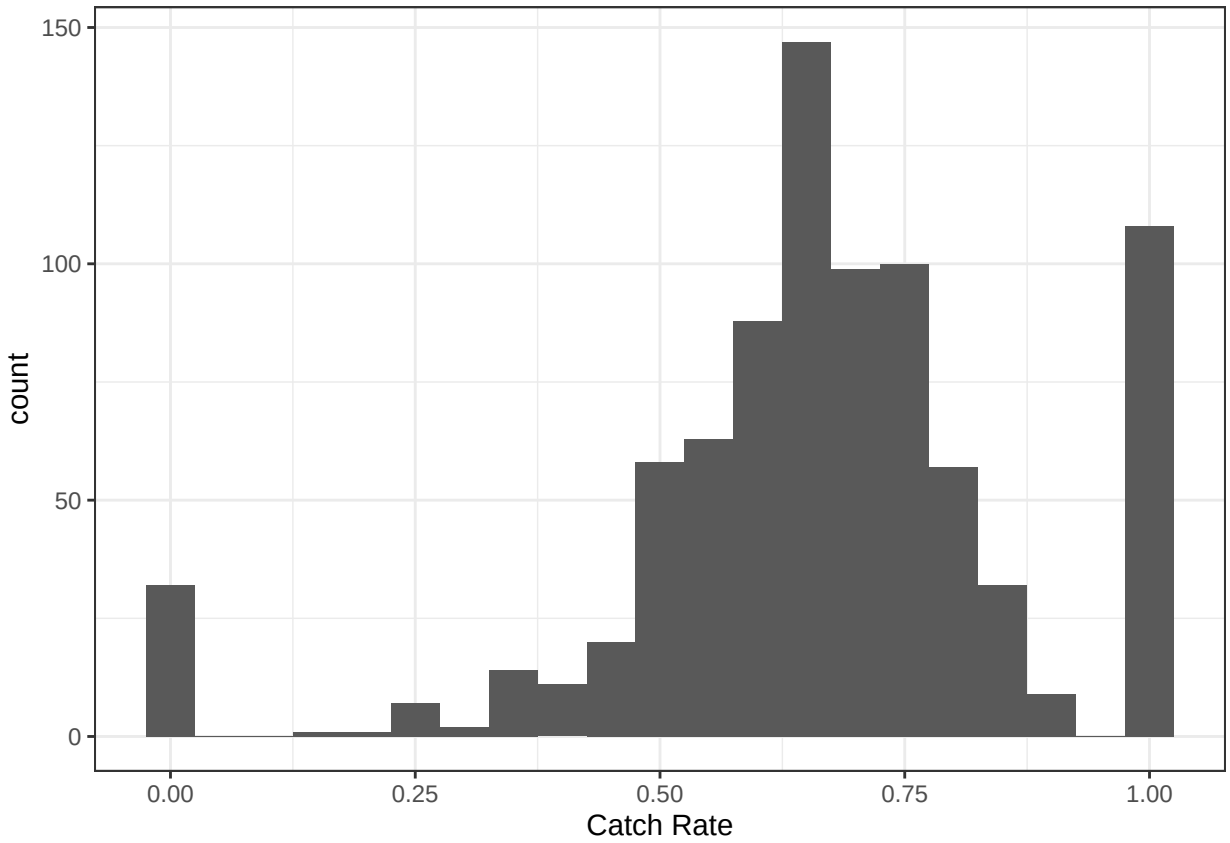
3a)

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

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

SOLUTION The "catch rate" is the MLE! It is equal to the number of events divided by the number of trials.

```
df_all %>%
  ggplot(mapping = aes(x = num_events / num_trials )) +
  geom_histogram(binwidth = 0.05) +
  labs(x = "Catch Rate") +
  theme_bw()
```



3b)

Plot the histogram for all "catch rates" in the `df_all` data set again. However, this time use `facet_wrap()` to break up the visualization into `num_trials > 24`.

What can you say about the observations of the players with greater than 25 targets?

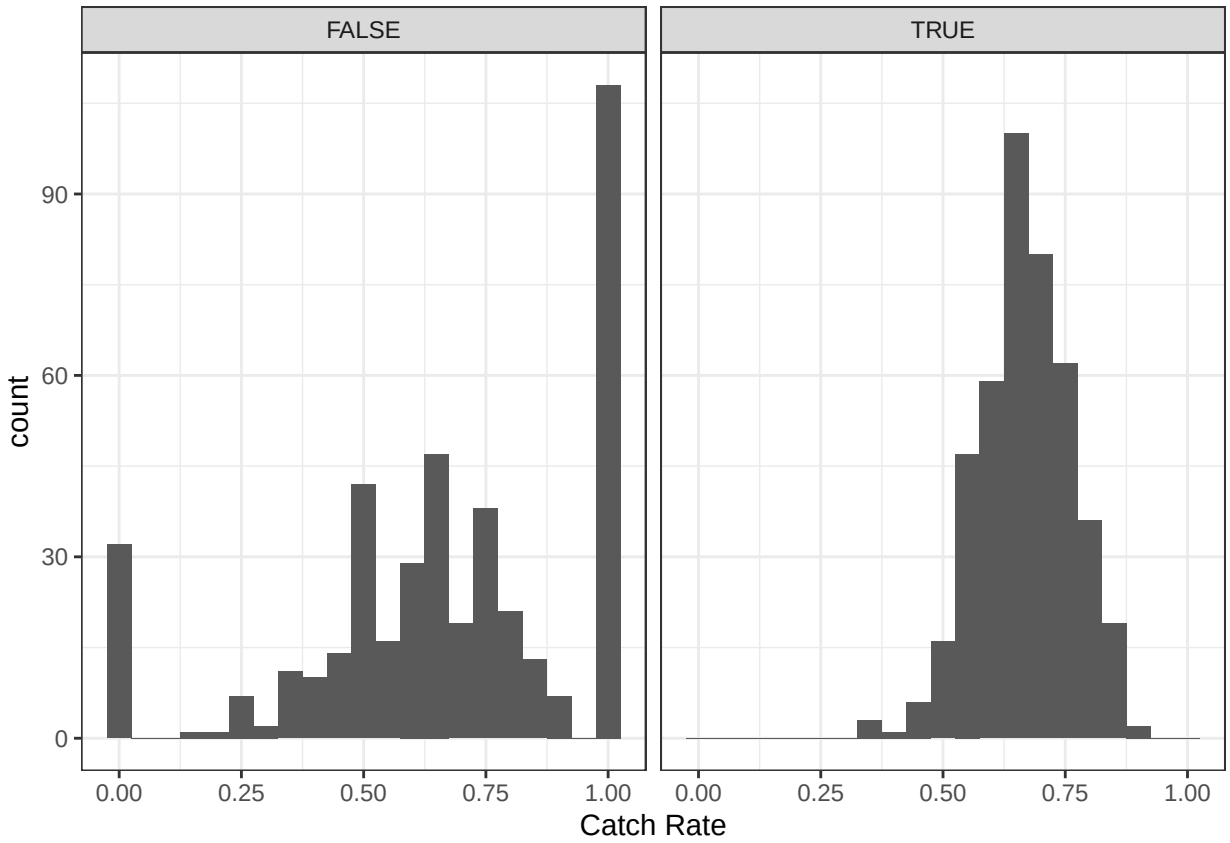
SOLUTION The left facet (subplot) below corresponds to all players with less than 25 trials (targets). The right facet shows the histogram of the MLEs for all players with 25 and more targets. At first glance, the two distributions appear different, but that is because of the "spikes" present for players with fewer targets. We can see in the left facet that there are about 100 players with a "catch rate" of 1 (or 100%)! There are also about 30 players with a "catch rate" of 0.

When when number of trials is larger, the "catch rate" distribution is more "well behaved" and looks somewhat like a bell curve. The spikes at 1 and 0 are not present, but there are also no spikes at other "common" fractions such 0.5, 2/3, and 0.75.

Larger sample sizes reveal the plausible or relevant values for the "catch rate". We see that values less than 50% are relatively rare, and values less than 25% are completely ruled out. We can assume that a professional NFL player will typically catch a pass more often than not catch a pass (otherwise the player will not be in the league very long).

The larger sample size plot also shows us that a "catch rate" of 100% is not plausible. This makes sense since there are many potential reasons why the player may not catch a pass. The throw could be poor from the quarterback. The defense may have "broken up" the pass. Or perhaps the receiver simply dropped the pass. After all, nobody is perfect!

```
df_all %>%
  ggplot(mapping = aes(x = num_events / num_trials)) +
  facet_wrap(~num_trials > 24) +
  geom_histogram(binwidth = 0.05) +
  labs(x = "Catch Rate") +
  theme_bw()
```



3c)

To keep things simple for now, you will estimate the prior shape parameters, a and b, based only on the players with greater than 24 targets.

Use the `filter()` function to keep all players with greater than 24 targets and assign the result to the `df_24` object. Use the `summary()` function to check the summary stats on `num_trials` to make sure you performed the operation correctly.

```
df_24 <- df_all %>% filter(num_trials > 24)

df_24 %>% summary()
```

SOLUTION

```
##    player_id       num_trials      num_events
##  Min.   :  3.0   Min.   : 25.0   Min.   : 10.00
##  1st Qu.:204.0   1st Qu.: 47.0   1st Qu.: 31.00
##  Median :406.0   Median : 83.0   Median : 56.00
##  Mean   :406.5   Mean   :118.0   Mean   : 78.94
##  3rd Qu.:599.0   3rd Qu.:155.5   3rd Qu.:103.00
##  Max.   :841.0   Max.   :509.0   Max.   :365.00
```

From the above table we can see that the minimum value of the num_trials data is 25.
Hence our filtering operation is correct.

3d)

Since the "catch 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 "catch rate" is assumed conditionally independent given the unknown a and b shape parameters. The observed "catch rate" will be denoted as, θ_j, for each player and is defined as:

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

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

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

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 players in the `df_24` 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 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`.

```
info_for_ab <- list(
  theta = df_24$num_events / df_24$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.**

```
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(x = my_info$theta, shape1 = a, shape2 = b, log = TRUE))

  # account for the change of variables
  log_lik + log_a + log_b

}
```

Try out values of -2 for both log-transformed parameters. If your function is coded correctly you should get a value of -571.8519.

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

```
## [1] -571.8519
```

Try out values of 2.5 for both log-transformed parameters. If your function is coded correctly you should get a value of -254.3934.

```
my_beta_loglik(c(2.5,2.5), info_for_ab)
```

```
## [1] -254.3934
```

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.

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

log_ab_res_01
```

```
## $par
## [1] 2.759740 2.058504
##
## $value
## [1] 410.5272
##
## $counts
## function gradient
##       34       11
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]      [,2]
## [1,] -2380.488  2305.533
## [2,]  2305.533 -2458.724
```

Perform the optimization using the second starting guess.

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

log_ab_res_02
```

```
## $par
## [1] 2.759736 2.058500
##
## $value
## [1] 410.5272
##
## $counts
## function gradient
##       35       11
```

```
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]      [,2]
## [1,] -2380.481  2305.524
## [2,]  2305.524 -2458.712
```

Are the identified log-transformed estimates the same?
Yes the identified log-transformed estimates are the same.

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 <- c(exp(log_ab_res_01$par[1]), exp(log_ab_res_01$par[2]))

ab_emp_bayes
```

SOLUTION

```
## [1] 15.795734  7.834244
```

The Empirical Bayes parameters therefore represent having observed roughly 16 receptions out of about 24 targets (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 "catch rates" for all players with more than 24 targets.

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 catch rate in `df_24` and the maximum observed catch rate in `df_24`. Pipe the result into `mutate()` and calculate the beta density using the `ab_emp_bayes` shape hyperparameters and assign the result to the `beta_pdf` variable.

In the second code chunk, pipe the `df_24 tibble` into `ggplot()` and map the `x` aesthetic to the observed catch 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=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 `size` argument to 1.15.

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

IMPORTANT: If you are *not* comfortable with your `ab_emp_bayes` values, you may use `shape1=13` and `shape2=8`. These are **not** the correct answers, though they are in the right ballpark...

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

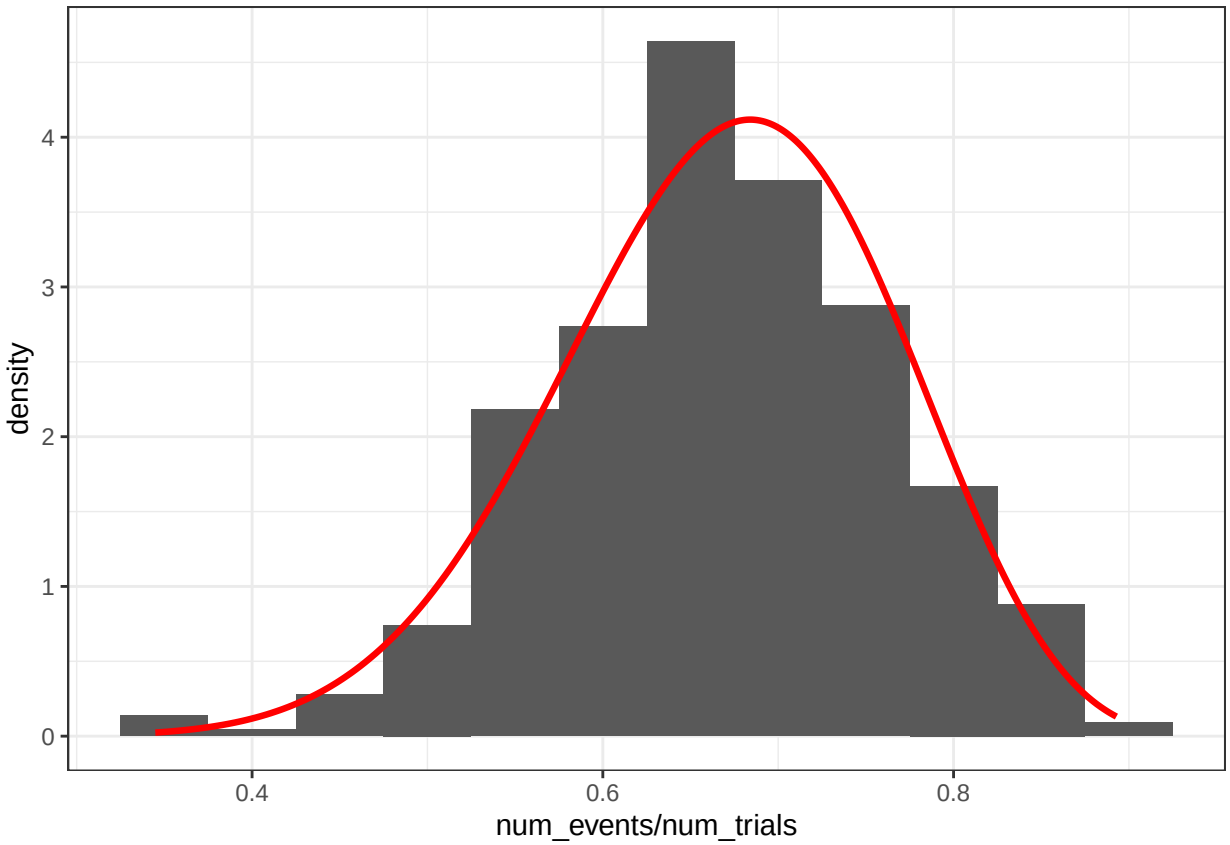
```
prior_for_viz <- tibble::tibble(
  x = seq(min(info_for_ab$theta), max(info_for_ab$theta), length.out = 1001)
) %>%
  mutate(beta_pdf = dbeta(x = x, shape1 = ab_emp_bayes[1], shape2 = ab_emp_bayes[2]))
```

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

As we can see below, the the plotted Beta density seems to agree with the shape of the catch rates for the players with more than 24 targets. We therefore fit a Beta distribution to observations of fractions!

```
df_24 %>%
  ggplot(mapping = aes(x = num_events / num_trials)) +
  geom_histogram(binwidth = 0.05,
                 mapping = aes(y = stat(density))) +
  geom_line(data = prior_for_viz,
            mapping = aes(x = x, y = beta_pdf),
            color = "red", linewidth = 1.15) +
  theme_bw()
```

```
## Warning: `stat(density)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



The empirically derived prior distribution on the event probability is almost similar compared to the observed histogram of the catch rates as we can see for several cases the red line crossing the top of the bin.

3h)

Calculate the 0.05 Quantile and 0.95 Quantile associated with your informative prior.

IMPORTANT: If you are *not* comfortable with your `ab_emp_bayes` values, you may use `shape1=13` and `shape2=8`. These are **not** the correct answers, though they are in the right ballpark...

SOLUTION We can calculate the 5th and 95th quantiles on the informative or Empirical Bayes derived prior using the qbeta() function. We simply need to assign the first element in ab_emp_bayes to shape1 and the second element in ab_emp_bayes to shape2.

As shown below, the middle 90% prior uncertainty interval is between about 0.5 and 0.82. Thus, the Empirical Bayes derived prior feels a player's catch rate is outside that interval with just 10% probability. The prior is therefore ruling out catch rates near 1, as well as catch rates near 0!

```
qbeta(c(0.05, 0.95), shape1 = ab_emp_bayes[1], shape2 = ab_emp_bayes[2])
```

```
## [1] 0.5042066 0.8161717
```

Problem 04

You now have everything in place to calculate the posterior on the event probability associated with each player, μ_j. The a and b parameters that you had originally set to both be 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=13` and `shape2=8`.

4a)

Calculate the updated or new shape parameters for the players 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, 0.05 Quantile, and 0.95 Quantile for each player 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)

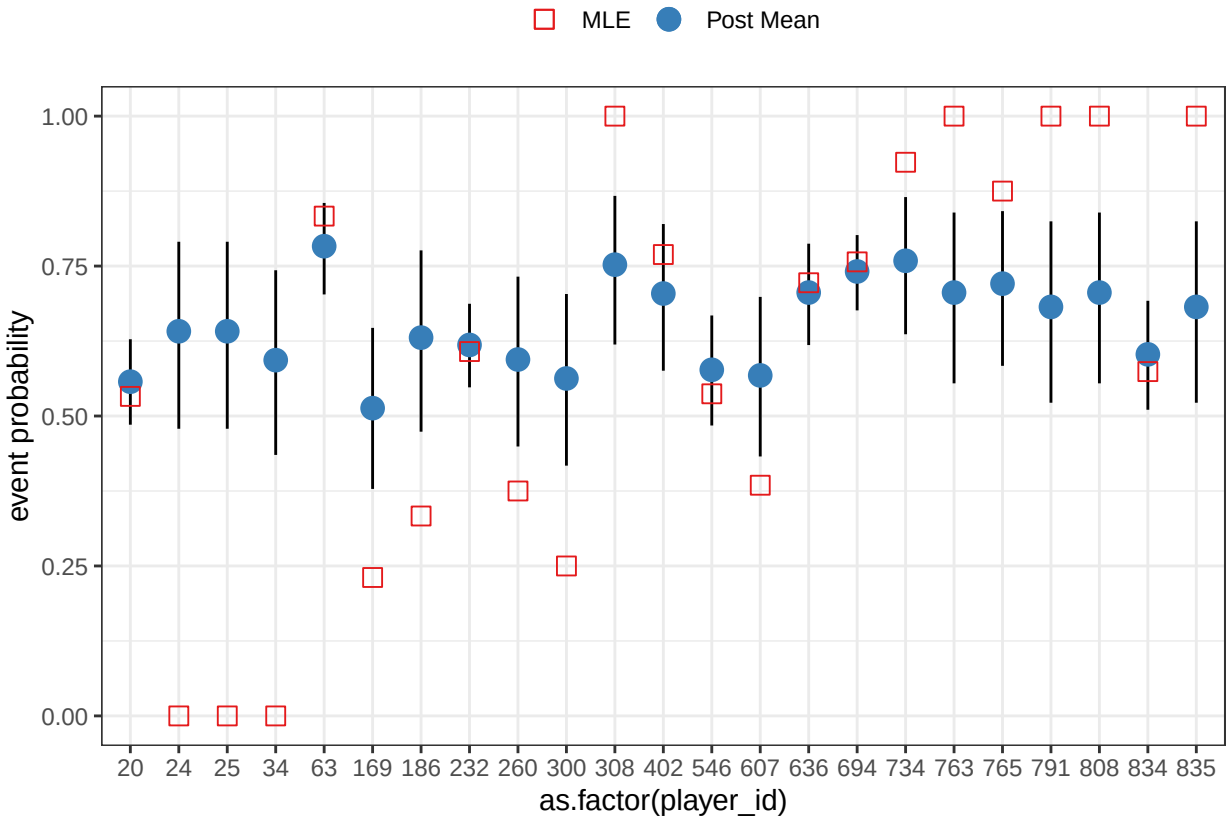
You will repeat the visualizations from Problem 1) to understand the effect of your informative prior distribution.

Pipe `summary_post_df_focus_empbayes` into `ggplot()` and map the `x` aesthetic to `as.factor(player_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?

SOLUTION As we can see below, none of the posterior means are less than 0.25 or greater than 0.8. The posterior middle 90% uncertainty intervals are also narrower than what we saw originally in Problem 1. The unpooled approach with the uninformative prior had posterior middle 90% uncertainty intervals covering 75% of the interval between 0 and 1. The informative Empirical Bayes derived prior however constrains the posterior uncertainty and prevents the posterior averages from moving too far away from the prior expected behavior **when the sample size is small.**

```
summary_post_df_focus_empbayes %>%
  ggplot(mapping = aes(x = as.factor(player_id))) +
  geom_linerange(mapping = aes(ymin = post_q05, ymax = post_q95)) +
  geom_point(mapping = aes(y = post_avg,
                           color = "Post Mean",
                           shape = "Post Mean",
                           size = "Post Mean")) +
  geom_point(mapping = aes(y = num_events / num_trials,
                           color = "MLE",
                           shape = "MLE",
                           size = "MLE")) +
  scale_color_brewer("", palette = "Set1") +
  scale_shape_manual("", values = c("MLE" = 0,
                                    "Post Mean" = 16)) +
  scale_size_manual("", values = c("MLE" = 3,
                                   "Post Mean" = 4)) +
  labs(y = "event probability") +
  theme_bw() +
  theme(legend.position = "top")
```



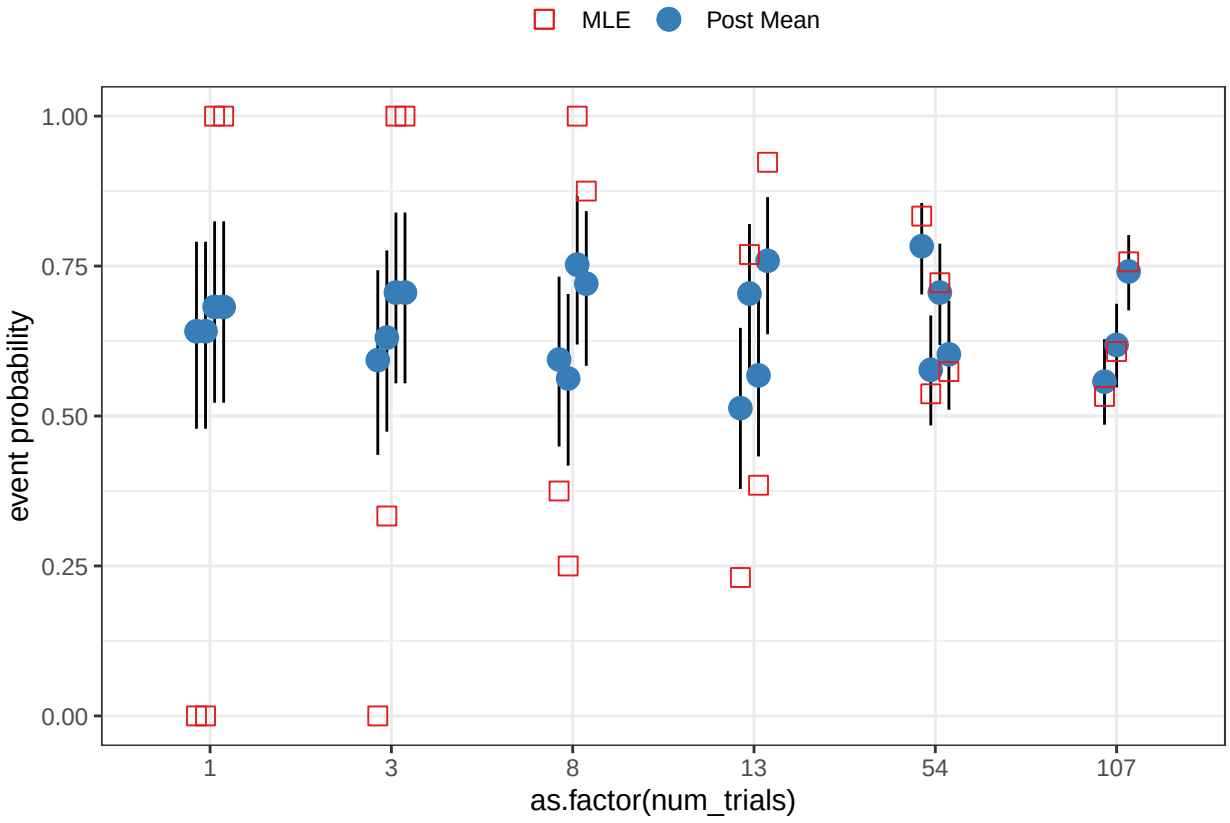
4d)

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

SOLUTION As we can see below, the players with the smallest sample sizes have similar posterior distributions. Their posterior means are slightly different, but the posterior uncertainty intervals have considerable overlap. This represents that the posterior is not changing that much relative to the prior **when the sample size is small.** The posterior means with num_trials=1 are quite different from the MLEs because the prior biases the event probability to be around 2/3. A posterior mean near 2/3 corresponds to the average behavior across all players, as we saw with the completely pooled estimate! However, by using the Empirical Bayes approach the uncertainty on the event probabilities are wider than the completely pooled approach when the sample size is small because we are treating each player as different.

```
summary_post_df_focus_empbayes %>%
  ggplot(mapping = aes(x = as.factor(num_trials))) +
  geom_linerange(mapping = aes(ymin = post_q05, ymax = post_q95,
                               group = player_id),
                 position = position_dodge(0.2)) +
  geom_point(mapping = aes(y = post_avg,
                           color = "Post Mean",
                           shape = "Post Mean",
                           size = "Post Mean",
                           group = player_id),
             position = position_dodge(0.2)) +
  geom_point(mapping = aes(y = num_events / num_trials,
                           color = "MLE",
                           shape = "MLE",
                           size = "MLE",
                           group = player_id),
             position = position_dodge(0.2)) +
  scale_color_brewer("", palette = "Set1") +
  scale_shape_manual("", values = c("MLE" = 0,
                                    "Post Mean" = 16)) +
  scale_size_manual("", values = c("MLE" = 3,
                                   "Post Mean" = 4)) +
  labs(y = "event probability") +
  theme_bw() +
  theme(legend.position = "top")
```



4e)

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

Calculate the updated shape parameters for all players 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.

```
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, 0.05 Quantile, and 0.95 Quantile for each player 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))
```

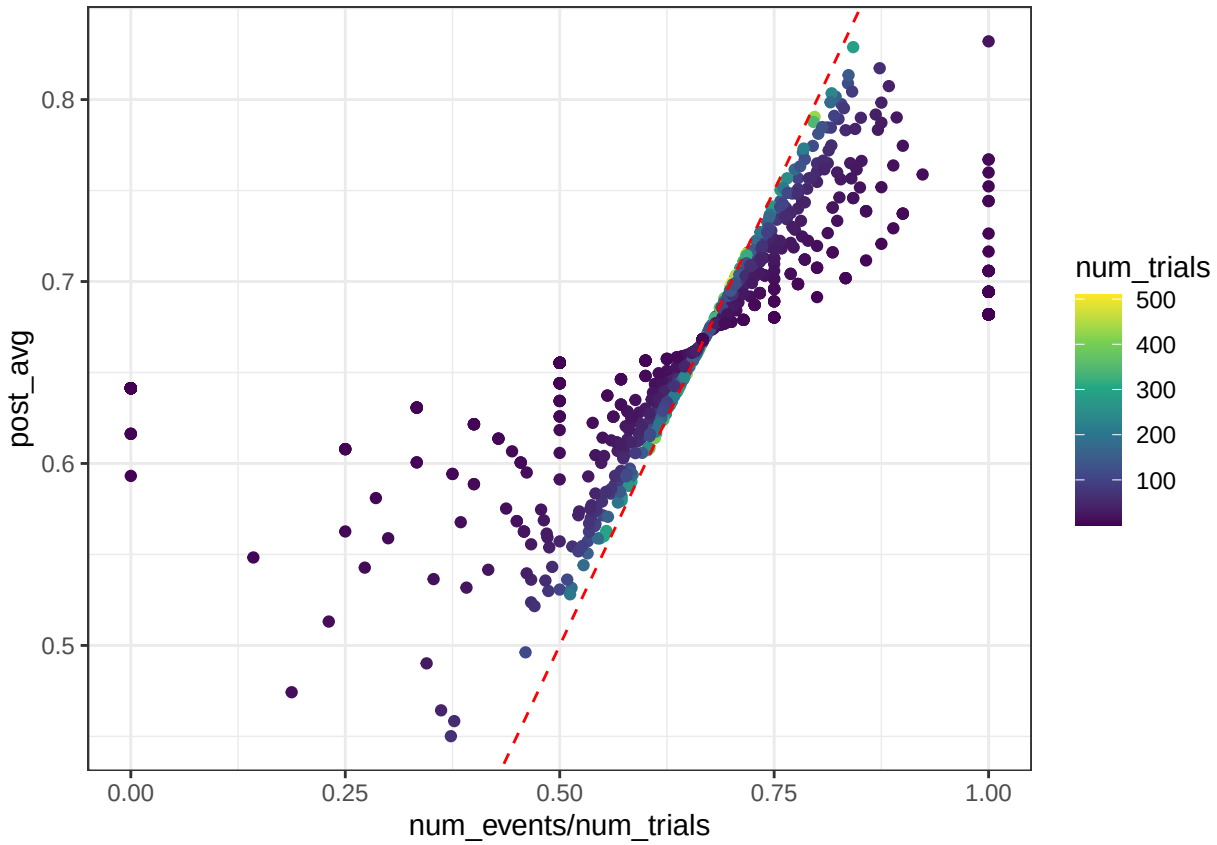
4g)

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 players. Map the `color` aesthetic to `num_trials` and include a `geom_abline()` layer with `slope = 1` and `intercept=0`.

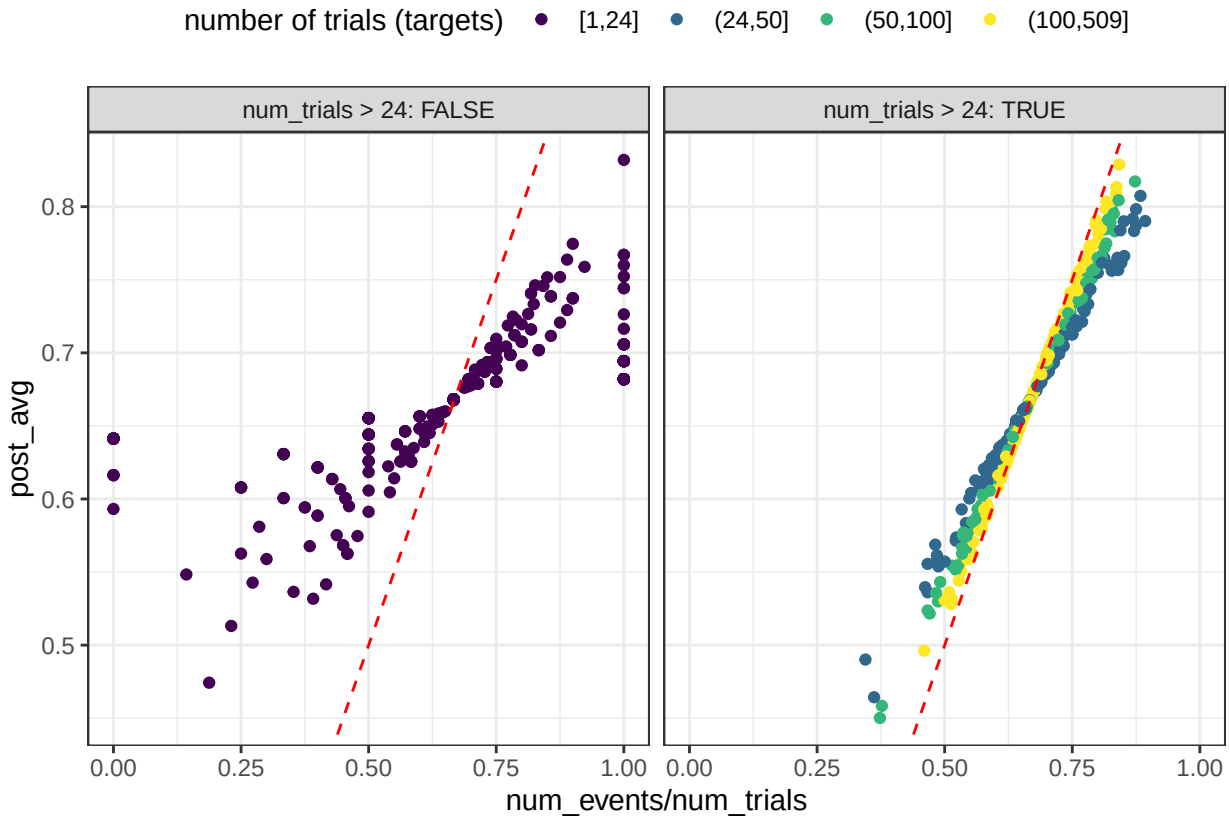
SOLUTION The color makes it clear that as the sample size increases the posterior mean becomes closer to the MLE. The figure below is therefore representing the fact that the prior is overwhelmed by the likelihood as the sample size gets larger and larger. The posterior is a compromise between the prior and data, and "follows" whichever is more precise. Large sample size cases correspond to precise likelihoods, and so the posterior mean is more aligned with the MLEs.

```r
summary_post_df_all_empbayes %>%
  ggplot(mapping = aes(x = num_events/num_trials, y = post_avg)) +
  geom_point(mapping = aes(color = num_trials)) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = 'dashed') +
  scale_color_viridis_c() +
  theme_bw()
```



As we see below, the left facet shows that small sample sizes cause the posterior mean to be different from the MLE. Our informative prior is more precise than the small amount of data we have collected and so the posterior "trusts" the prior more. However, when the sample size increases the posterior mean starts to line up with the MLE since the data precision is getting larger and larger and eventually overwhelms the prior for large sample sizes. That is why the yellow dots within the right facet (largest sample size markers) have the posterior means closest to the "45-degree" line.

```
summary_post_df_all_empbayes %>%
  ggplot(mapping = aes(x = num_events/num_trials, y = post_avg)) +
  geom_point(mapping = aes(color = cut(num_trials,
                                       breaks = c(1, 24, 50, 100, max(df_all$num_trials)),
                                       include.lowest = TRUE))) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = 'dashed') +
  facet_wrap(~num_trials > 24, labeller = "label_both") +
  scale_color_viridis_d("number of trials (targets)") +
  theme_bw() +
  theme(legend.position = "top")
```

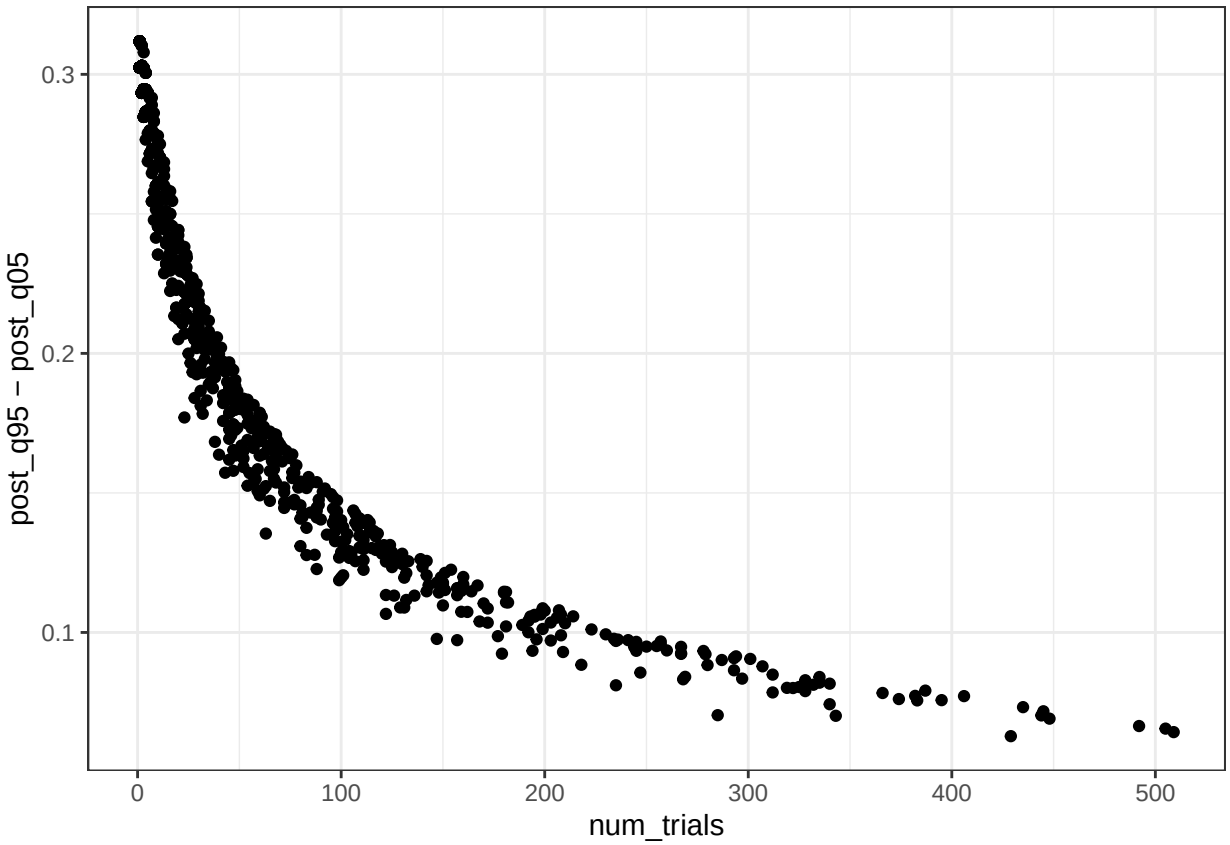


4h)

Create a scatter plot for the middle 90% uncertainty interval range (difference between the 0.95 and 0.05 Quantiles) with respect to the `num_trials` using `ggplot2`.

SOLUTION As we can see below, the posterior middle 90% uncertainty interval range is decreasing as the sample size increases.

```
summary_post_df_all_empbayes %>%
  ggplot(mapping = aes(x = num_trials, y = post_q95 - post_q05)) +
  geom_point() +
  theme_bw()
```



4i)

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 An informative prior prevents the posterior from taking on implausible values when the sample size is small. As we saw with the uninformative prior, the posterior mean could be near 0.75 when there was a trial size of 1 and 1 event or near 0.25 where there was a trial size of 1 and 0 events. An informative prior sets meaningful bounds on the plausible set of values the parameter can take.

Problem 05

Now that you have posterior distributions based on an informative prior for every player in the data set, it is time to consider answering a question the NFL team is interested in. The team wants to identify the best receivers in the data set, and it wants to be confident in that selection. Your Bayesian analysis allows answering probabilistic questions. You will answer several such questions now.

5a)

Calculate the probability that each player has a catch rate (event probability) of greater than 0.67. Add a column to the `summary_post_df_all_empbayes` object named `prob_grt_67`. Assign the result to a new variable `post_player_eval`.

```r
summary_post_df_all_empbayes <- summary_post_df_all_empbayes %>%
  mutate(prob_grt_67 = 1 - pbeta(0.67, anew, bnew))
```

SOLUTION

5b)

Identify the top 10 players based on the posterior probability that their catch rate is greater than 0.67. What do these players all have in common, besides the `prob_grt_67` value?

SOLUTION The posterior probability that the catch rate is greater than 0.67 is near 1 for the top 10 players! All of these players have a large number of trials, with the smallest trial size of 122, as given in the print out below. Thus, these are not simply players that have caught the ball everytime out of just a few targets (small sample size). We are confident in their performance because of the large number of samples available.

```r
summary_post_df_all_empbayes %>%
  arrange(desc(prob_grt_67)) %>%
  select(player_id, num_trials, post_q05, post_avg, post_q95, prob_grt_67) %>%
  head(10)
```

```
## # A tibble: 10 x 6
##    player_id num_trials post_q05 post_avg post_q95 prob_grt_67
##        <dbl>      <dbl>    <dbl>    <dbl>    <dbl>       <dbl>
##  1       409        285    0.792    0.829    0.863        1.00
##  2       353        429    0.758    0.790    0.821        1.00
##  3       498        343    0.752    0.788    0.822        1.00
##  4       467        235    0.762    0.803    0.843        1.00
##  5       382        147    0.762    0.813    0.860        1.00
##  6       493        179    0.751    0.798    0.843        1.00
##  7       447        157    0.751    0.802    0.848        1.00
##  8       278        122    0.753    0.809    0.860        1.00
##  9       567        218    0.728    0.773    0.816        1.00
## 10       302        340    0.715    0.753    0.789        1.00
```

5c)

Identify the 10 players with the lowest posterior probability that their catch is greater than 0.67. What is the smallest number of targets (trial size) associated with these 10 players?

SOLUTION As shown below, of the players with the 10 lowest posterior probabilities of having catch rates greater than 0.67, the smallest sample size is 47. Thus, the rankings are not simply just picking players with a few targets! These are players that have large enough sample sizes that we are confident in their performance.

```r
summary_post_df_all_empbayes %>%
  arrange(prob_grt_67) %>%
  select(player_id, num_trials, post_q05, post_avg, post_q95, prob_grt_67) %>%
  head(10)
```

```
## # A tibble: 10 x 6
##    player_id num_trials post_q05 post_avg post_q95 prob_grt_67
##        <dbl>      <dbl>    <dbl>    <dbl>    <dbl>       <dbl>
##  1       545        207    0.474    0.528    0.582  0.00000426
##  2       222         67    0.365    0.450    0.536  0.00000897
##  3       837        113    0.426    0.496    0.566  0.0000146
##  4       624        301    0.515    0.560    0.605  0.0000196
##  5        61        181    0.474    0.532    0.589  0.0000217
##  6       435         61    0.370    0.458    0.548  0.0000330
##  7       195        294    0.517    0.563    0.608  0.0000372
##  8       615        180    0.487    0.544    0.601  0.000101
##  9       559        214    0.506    0.559    0.611  0.000190
## 10       506         47    0.368    0.464    0.562  0.000197
```

5d)

A player with a large sample size could mean that player is well known, especially around the NFL. The team is interested in identifying players that are not as well known, and yet seem to have high catch rates.

Identify 10 players with the smallest sample sizes (number of trials) while still having `prob_grt_67` values greater than 0.75.

```
summary_post_df_all_empbayes %>%
  arrange(num_trials) %>%
  filter(prob_grt_67 > 0.75) %>%
  select(player_id, num_events, num_trials, post_avg, prob_grt_67) %>%
  head(10)
```

SOLUTION

```
## # A tibble: 10 x 5
##    player_id num_events num_trials post_avg prob_grt_67
##        <dbl>      <dbl>      <dbl>    <dbl>       <dbl>
##  1       701          5          5    0.726       0.761
##  2        81          7          7    0.744       0.830
##  3        88          7          7    0.744       0.830
##  4       308          8          8    0.752       0.858
##  5        41          9          9    0.760       0.882
##  6       606          8          9    0.729       0.784
##  7       264          9         10    0.737       0.816
##  8       321          9         10    0.737       0.816
##  9       376         10         10    0.767       0.902
## 10       476          9         10    0.737       0.816
```

5e)

Why do you think the questions in this problem were focused on calculating the probability that the catch rate is greater than 0.67? What is the interpretation of such a question?

HINT: Consider the interpretation of the completely pooled estimate.

SOLUTION What do you think?
We were focused on comparing player performance with 0.67 because a catch rate of 0.67 is the overall or grand average catch rate across all players in the league! We saw that in the complete pooling estimate since we essentially summed up the total number of catches and divided by the total number of targets. We are thus considering which players are confidently better than or worse than league average!

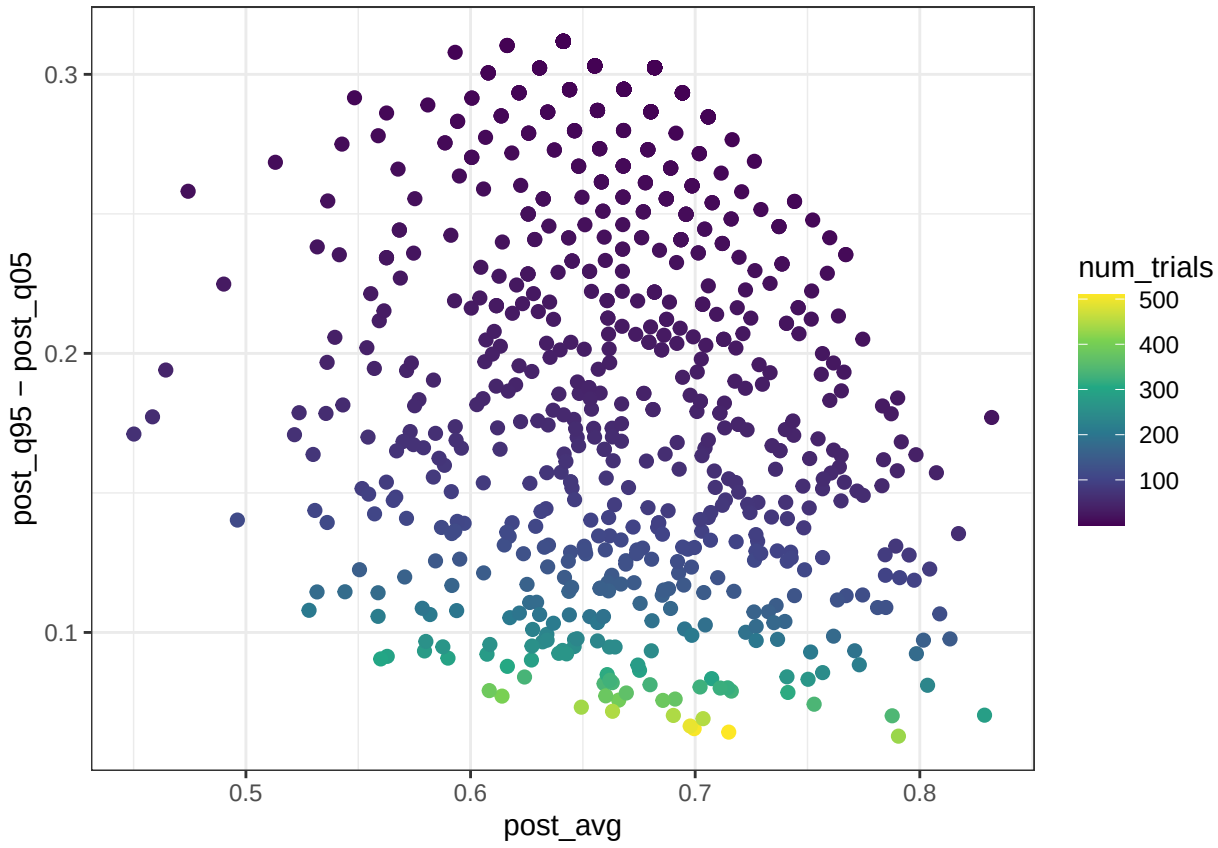
5f)

Sometimes teams are willing to take risks on players that they are uncertain about. Teams hope the player will succeed, but they know there is a chance the player could not meet expectations. To manage risks, these players are signed to contracts with less guaranteed money but offer substantial bonuses should the player meet performance goals to help incentive the player.

The posterior distribution allows you to quantify the uncertainty on each player. You will measure the uncertainty as the range of the posterior 90% uncertainty interval.

Create a scatter plot in `ggplot2` where you plot the posterior 90% uncertainty interval with respect to the event probability's posterior mean. Color the markers by the number of trials.

```
summary_post_df_all_empbayes %>%
  ggplot(mapping = aes(x = post_avg, y = post_q95 - post_q05)) +
  geom_point(mapping = aes(color = num_trials),
             size = 2) +
  scale_color_viridis_c() +
  theme_bw()
```



SOLUTION

5g)

Which players would you recommend to the team to take a risk on based on your figure in 5f)?

SOLUTION What do you think?
From above figure we should look for players with high levels of uncertainty and secondly, we want to identify players with high mean values. In this way, there is a risk the player could do worse than what we expect them to do (a downside), but there is also a chance the player can outperform our expectations. If you look closely at the previous scatter plot the players with the highest posterior uncertainty are those near the grand average! These players are uncertainty because their sample size is very small and thus the posterior is controlled by the prior.

If we had to select specific players, it would be those identified by red in the figure below. These players have relatively high posterior mean values, as well as higher uncertainty values. Thus, these players are expected to perform well (high catch rate), yet the sample sizes are relatively low leading to high uncertainty on the performance.

```
summary_post_df_all_empbayes %>%
  ggplot(mapping = aes(x = post_avg, y = post_q95 - post_q05)) +
  geom_point(data = summary_post_df_all_empbayes %>%
               mutate(post_unc_interval = post_q95 - post_q05) %>%
               filter(post_avg > 0.78,
                      post_unc_interval > 0.15),
             color = 'red', size = 5) +
```

```
geom_point(mapping = aes(color = num_trials),
           size = 2) +
scale_color_viridis_c() +
theme_bw()
```