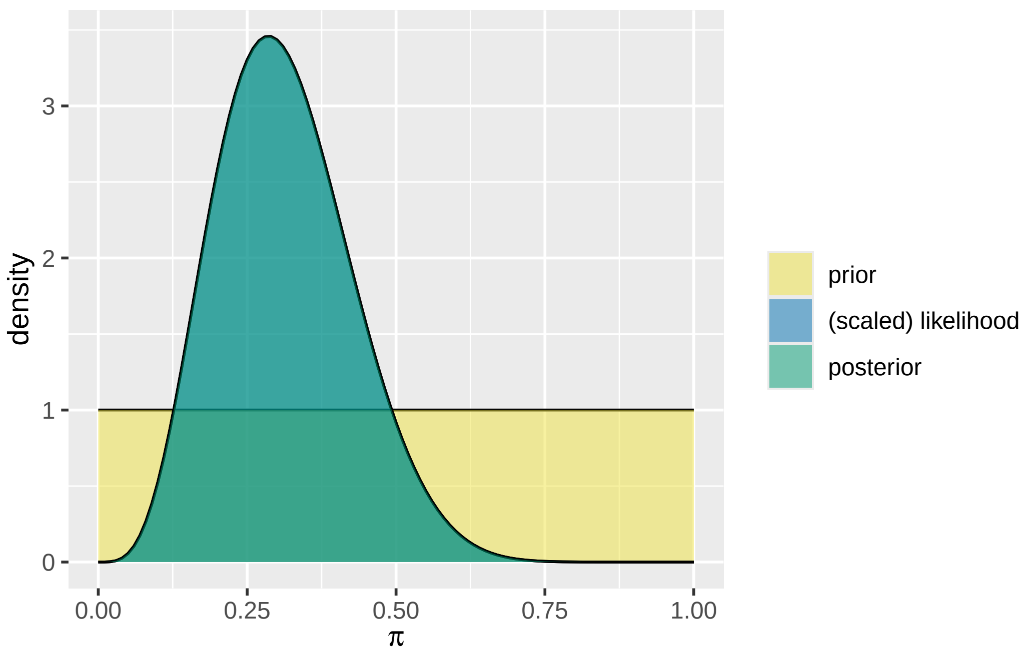
Ejercicios 4 y 5

Sofi Terra, Santi R

Ejercicio 4.19

```
binary  n   percent
  FAIL 10 0.7142857
  PASS  4 0.2857143
 Total 14 1.0000000
```

Generamos el grafico de prior, verosimilitud y posterior con plot beta binomial



Se observa que la verosimilitud es igual al posterior. Esto ocurre porque el prior es una beta (1,1) que es uniforme en el recorrido. Inter pretando esto de una manera bayesiana, signfiica que no aporta nada de informacion. Se utiliza para repersentar neutralidad absoltua. Siempre que partimos de una beta con tales parametros, la verosimilud sera igual al posterior.

Para el **Calculo teorico de la esperanza y modo** utilizaremos el calculo del posterior del modelo beta binomial, que sabemos que distribuye Beta (α+y,β+n-y).

```
      model alpha beta   mean      mode        var        sd
1     prior     1    1 0.5000       NaN 0.08333333 0.2886751
2 posterior     5   11 0.3125 0.2857143 0.01263787 0.1124183
```

Parte dos: Calculo para 1990

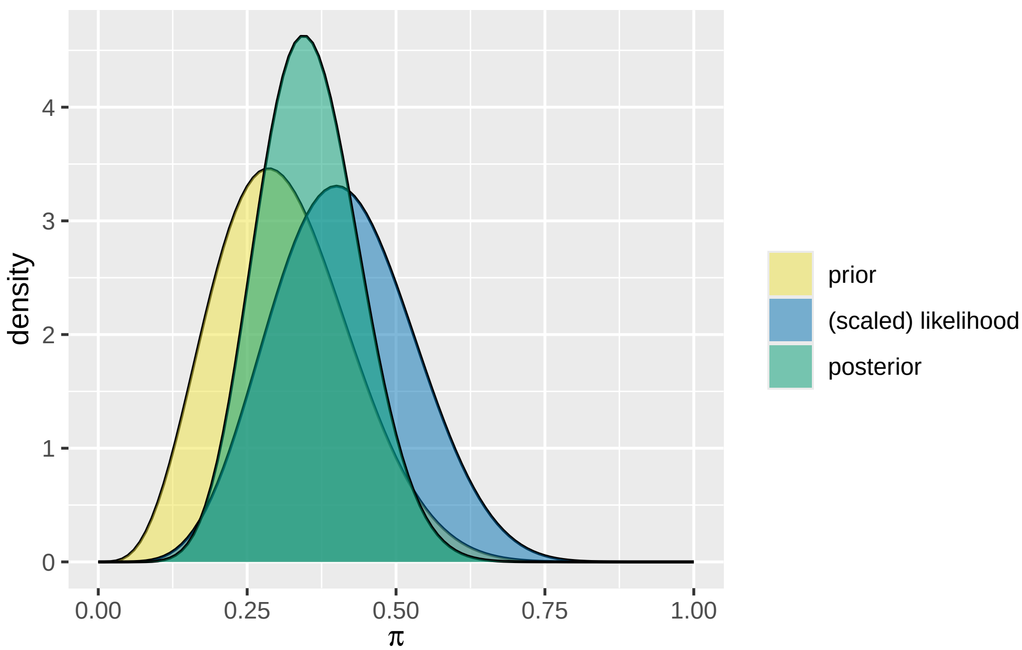
Busqueda de la informacion para 1990 (Verosimilitud)

```
bechdel %>%
  filter(year == 1990) %>%
  tabyl(binary) %>%
  adorn_totals("row")
```

```
 binary  n percent
   FAIL  9     0.6
   PASS  6     0.4
  Total 15     1.0
```

Partimos del posterior anterior que ya sabemos como distribuye gracias a la formula del posterior en el modelo beta binomial y cargamos la verosimilitud buscada en el paso anterior. Dicho "posterior" se convirtio en nuestro nuevo prior y se agrega una nueva verosimilitud.

```
plot_beta_binomial(alpha = 5, beta = 11, y = 6, n = 15)
```



Calculo de la esperanza teorica

```
summarize_beta_binomial(5, 11, 6, 15)
```

```
      model alpha beta      mean      mode         var         sd
1     prior     5   11 0.3125000 0.2857143 0.012637868 0.11241827
2 posterior    11   20 0.3548387 0.3448276 0.007154006 0.08458136
```

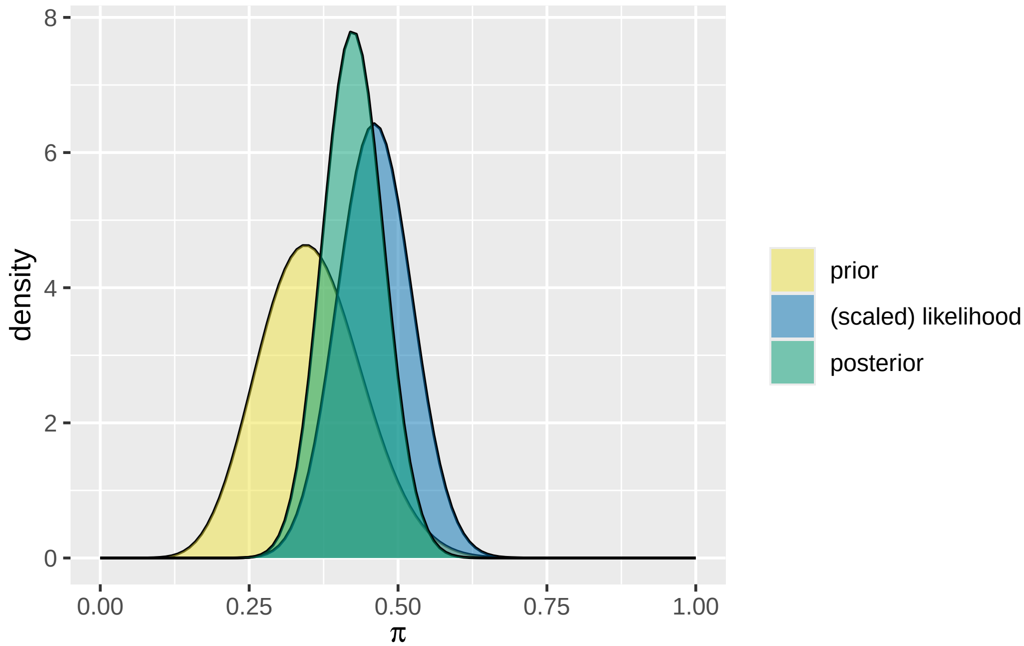
Parte 3: Ano 2000

Nuestro nuevo prior distribuira Beta(11, 20)

```
bechdel %>%
  filter(year == 2000) %>%
  tabyl(binary) %>%
  adorn_totals("row")
```

```
 binary  n   percent
   FAIL 34 0.5396825
   PASS 29 0.4603175
  Total 63 1.0000000
```

```r
plot_beta_binomial(alpha = 11, beta = 20 , y = 29, n = 63)
```




```r
summarize_beta_binomial(11, 20, 29, 63)
```

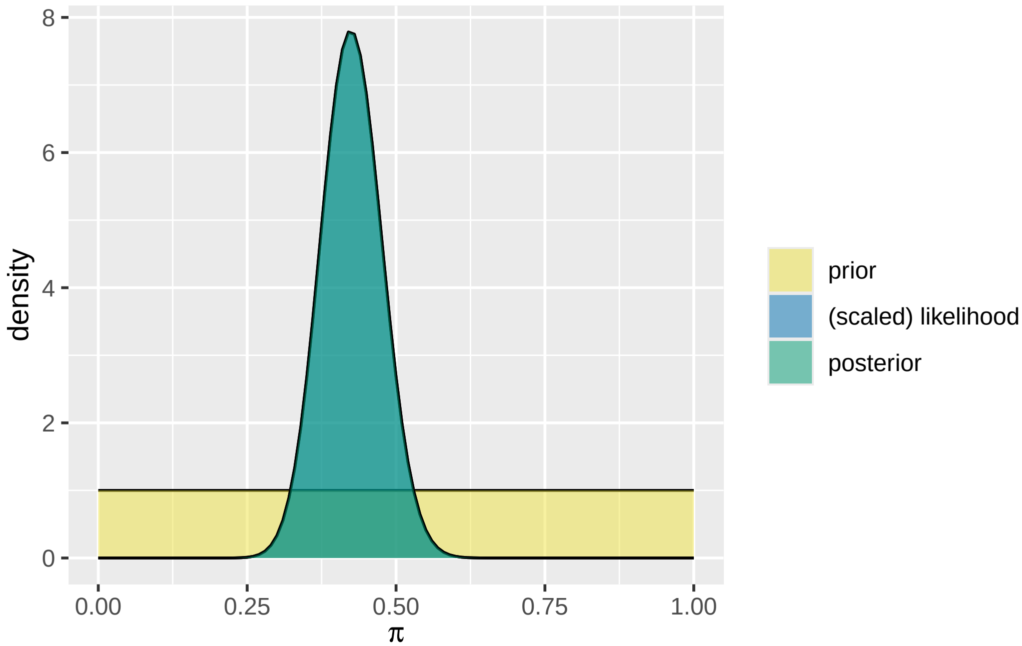
```
      model alpha beta      mean      mode         var         sd
1     prior    11   20 0.3548387 0.3448276 0.007154006 0.08458136
2 posterior    40   54 0.4255319 0.4239130 0.002573205 0.05072677
```

Parte 4: Calculo de Jenna

```r
bechdel %>%
  filter(year %in% c(1980, 1990, 2000)) %>%
  tabyl(binary) %>%
  adorn_totals("row")
```

```
 binary  n  percent
   FAIL 53 0.576087
   PASS 39 0.423913
  Total 92 1.000000
```

```r
plot_beta_binomial(alpha = 1, beta = 1 , y = 39, n = 92)
```




```r
summarize_beta_binomial(1, 1, 39, 92)
```

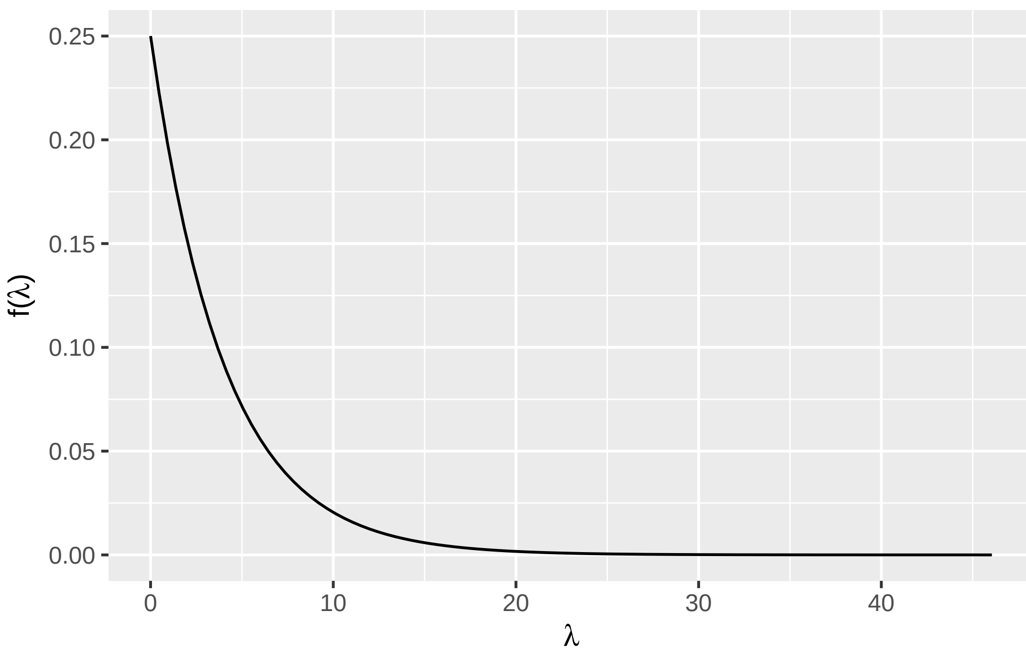
```
      model alpha beta      mean     mode         var         sd
1     prior     1    1 0.5000000      NaN 0.083333333 0.28867513
2 posterior    40   54 0.4255319 0.423913 0.002573205 0.05072677
```

Se observa que Jenna llega a la misma posterior que John; probando que no importa el orden en el que mires la informacion, se llegara a los mismos resultados. John realiza su analisis en tres dias (pasos), mientras que Jenna solo en uno. Pero al partir del mismo prior (Beta(1,1)) y utilizar las mismas muestras, llegan a los mismos posteriors.

Ejercicio 5.7: Womens world cup

Utilizaremos plot_gamma (1, 0.25) dado que es el prior dado.

```r
plot_gamma(1, 0.25)
```



Se observa que λ es decreciente y parte desde cero, con asimetria a la derecha. Es sumamente coherente con el contexto de que λ representa la cantidad de goles promedio, seria extrano que 20 tenga probabilidad alta, por ejemplo. Sugiere que no habra goles en la mayoria de partidos.

Parte 2: Yi para modelo Poisson

Se utiliza el modelo Poisson para representar a Y dado que este es util para los conteos. Y es una variable aleatoria para representar los goles.

Cada Yi es una observacion, es decir los goles de cada partido puntual. Por otro lado λ representa la tasa media de ocurrencia, osea, la cantidad promedio de goles por partido.

Parte 3: Total de goles por partido

```r
library(fivethirtyeight)
```

```
Some larger datasets need to be installed separately, like senators and
house_district_forecast. To install these, we recommend you install the
fivethirtyeightdata package by running:
install.packages('fivethirtyeightdata', repos =
```

```
'https://fivethirtyeightdata.github.io/drat/', type = 'source')
```

```
Adjuntando el paquete: 'fivethirtyeight'
```

```
The following object is masked _by_ '.GlobalEnv':

    bechdel
```

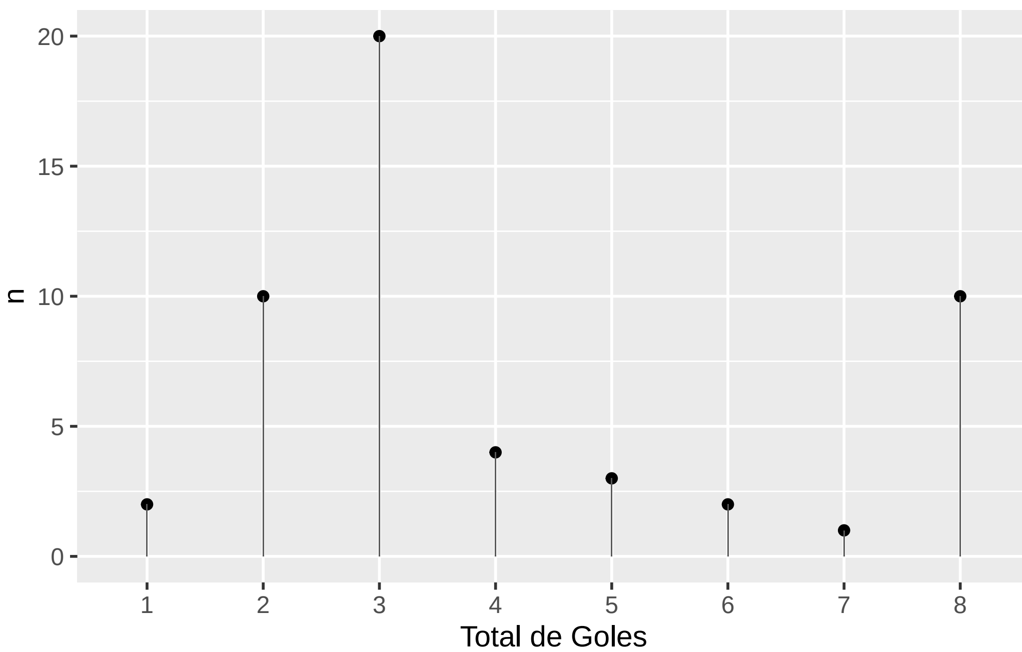
```
The following object is masked from 'package:bayesrules':

    bechdel
```

```
data("wwc_2019_matches")
wwc_2019_matches <- wwc_2019_matches %>%
  mutate(total_goals = score1 + score2)


wwc_2019_matches <- wwc_2019_matches %>%
  tabyl(total_goals) %>%
  adorn_totals("row") %>%
  filter (total_goals != "Total")
```

```
ggplot(data=wwc_2019_matches,aes(x=factor(order(total_goals)), y= n)) + scale_x_discrete(name
```



Falta la observacion 13 pero estuvimps 1 hora para ponerla y no pudimos y tm falta el cero pero no ha habido chance!!!!!

Se observan la cantidad de goles por partido y cuantas veces se repite ese resutado, es decir la frecuencia para cada resultado.

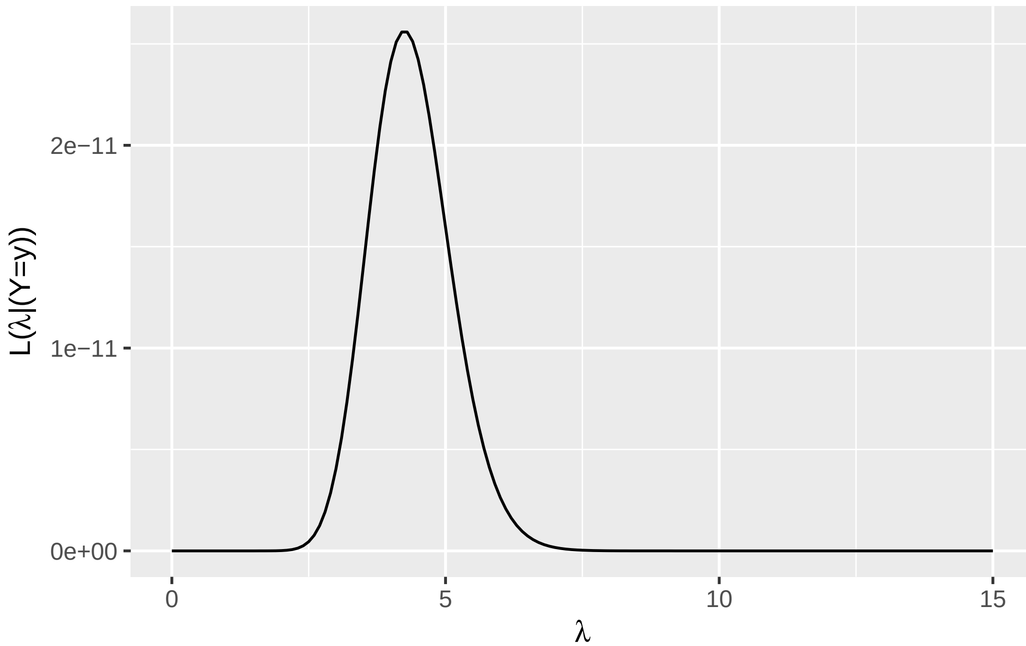
Para poder graficar el posterior y la verosimilitud necesitaremos calcular el total de goles que se convirtieron y el total de partidos. Este ultimo se observa que es 52 en wwc_2019_matches.

Para calcular el total de goles, debemos multriplcar la frecuencia de cada resultado por el total de goles de ese resultad yluego sumarlo.

```
wwc_2019_matches <- wwc_2019_matches %>%
  mutate (total_goals=as.numeric(total_goals)) %>%
  mutate (golesaux = n * total_goals)
sum_goles <- sum(wwc_2019_matches$golesaux, na.rm = TRUE)
```

Removeremos el valor NA que generamos antes al utilizar as.numeric (generamos NA el total, que NO interesa graficarlo y alteraria la consistencia de nuestros datos)

```
wwc_2019_matches <- wwc_2019_matches %>%
  filter(!is.na(total_goals))
plot_poisson_likelihood(wwc_2019_matches$total_goals, 15)
```

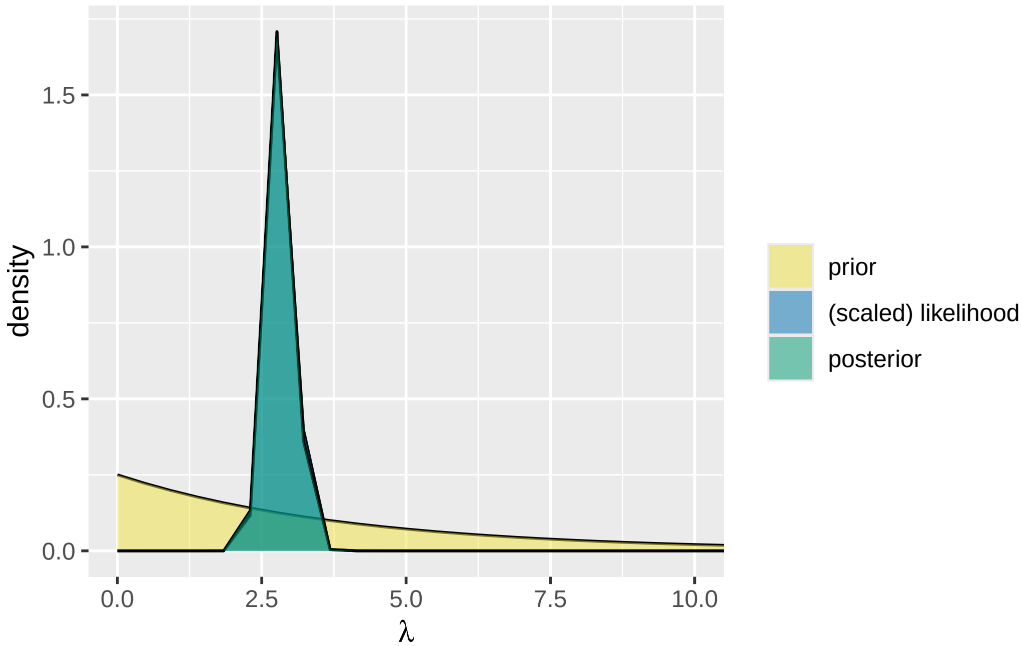


Se observa que los valores mas verosimiles de λ rondan entre 3 y 5 goles de media por partido. Es una likelihood muy agresiva, en el sentido de que asigna probabilidad practicamente nula a 0 y 1 gol.

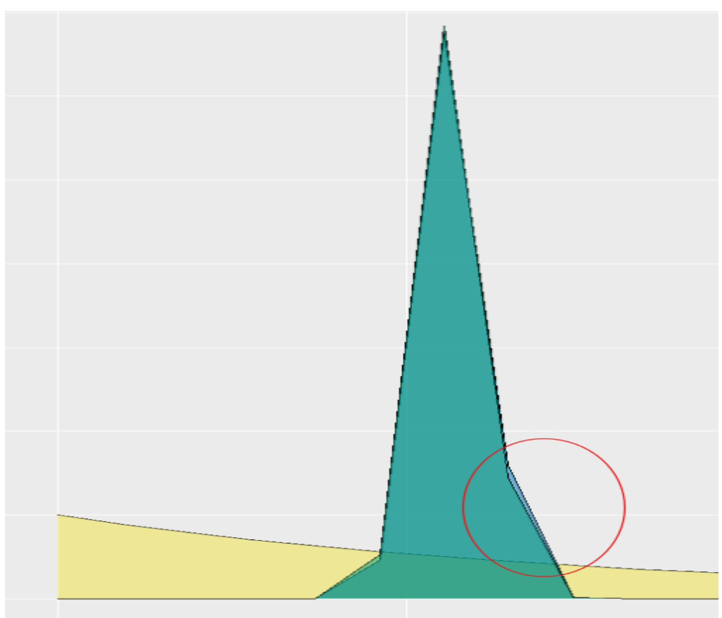
Grafico del posterior

Ahora interesa calcular el posterior. Para ello utilizaremos la funcion plot_gamma_poisson.

```
plot_gamma_poisson(1 , 0.25, sum_goles,52) + coord_cartesian(xlim = c(0, 10))
```



En este caso, el posterior y la likelihood son practicamente iguales. Recien al hacer zoom encontramos una pequena diferencia:



Pensamos que esto ocurre por varios motivos: Tenemos un n considerablemente grande (52), la funcion de verosimilitud esta sumamente concentrada entorno a un valor (mucha seguridad) y el prior no esta tan concentrado, sino que es mas disperso.

De esta manera, hay un cambio radical en nuestro entendimiento de la media de goles por partido. De pasar de pensar que la media podia ser 0, 1 o 2, pasamos a poder afirmar que la media de goles estara en el entorno de 3.

#Ejercicio 5.12: Control brains

Aclaracion: Cuando la letra del ejercicio plantea "control subjects who have not been diagnosed with a concussion", entendemos que refiere al grupo denominado fb_no_concuss, pero puede haber habido un error de interpretacion.

Primero procederemos con cargar la data y filtrarla por los que no tuvieron contusiones.

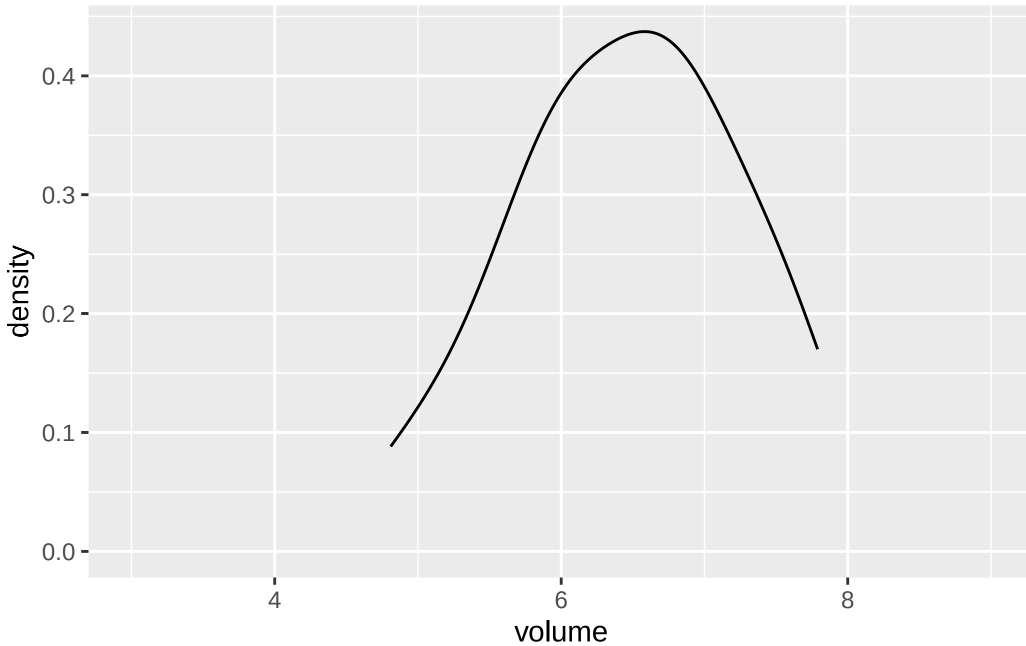
```
data(football)
concussion_subjects <- football %>%
  filter(group == "fb_no_concuss")
```

```
concussion_subjects %>%
  summarize(mean(volume))
```

```
  mean(volume)
1       6.4592
```

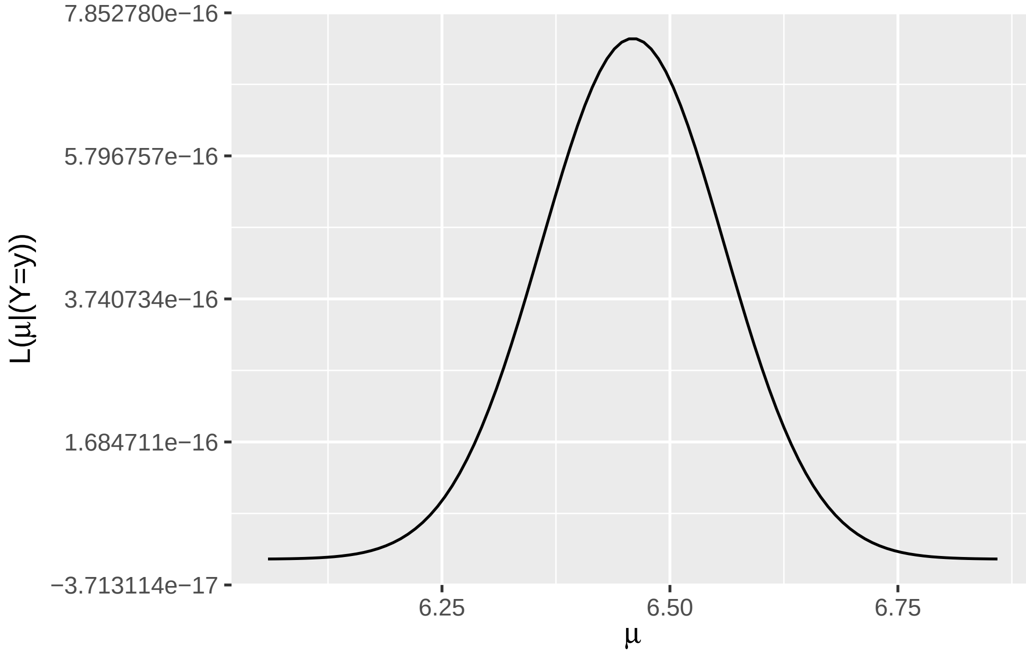
El promedio de los sujetos que no tuvieron contusiones es de 6.4592 cm3 en una muestra de 25 personas.

```
ggplot(concussion_subjects, aes(x = volume)) +
  geom_density()+coord_cartesian(xlim = c(3, 9))
```



Se observa que los valores van desde 4.81 hasta 7.79 y alcanzan el maximo de densidad entorno a 6.5, que es el promedio.

```
plot_normal_likelihood(y = concussion_subjects$volume, sigma = 0.5)
```



De esta manera se visualiza la verosimilitud de u.

Calculo del posterior

Ya tenemos todo lo necesario para identificar el posterior: El prior tiene media =6.5 y =0.4 La muestra de los que no tuvieron contusion tiene media y=6.4592 y asumimos desvio conocido (por letra) tal que =0.5.

Entonces el posterior se calculara de la siguiente forma, segun el modelo normal- normal.

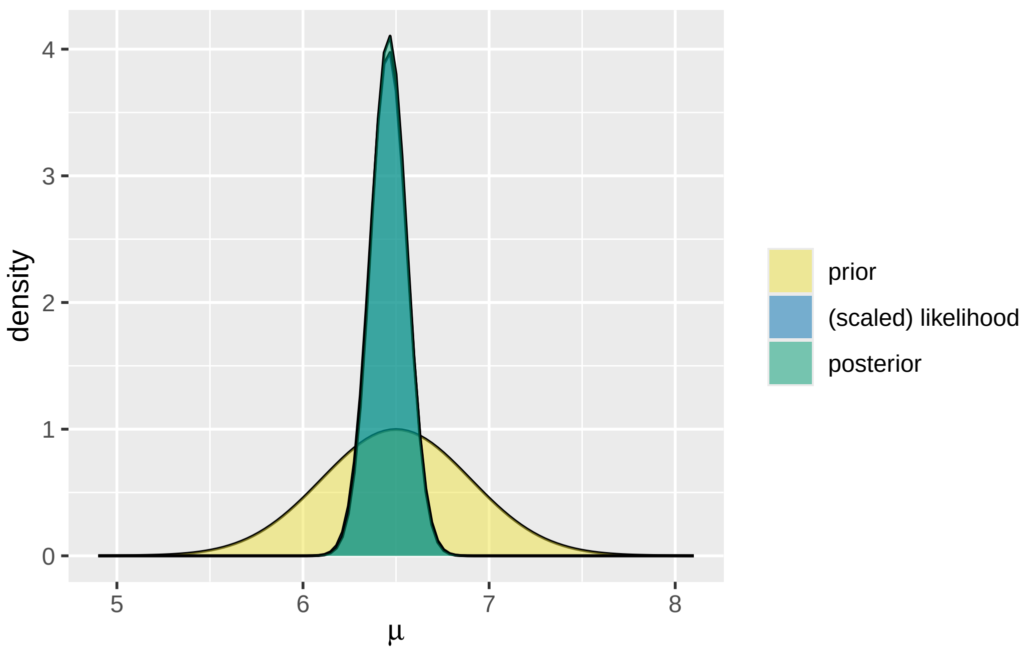
eEl chunk de aca abajo solo lo meti para poder renderizar

```
#\$  \|vec{y}  N(6.5\*0.5\^2 / 25\* (0.4^2()+(0.5^2) + 6.4592\* (0.4\^2)/25 \* (0.4^2)+(0.5^2)

#Es decir: $u|vec{y} ~ N ( 6.46, 0.009^2)$
```

Parte 3: Plot del prior, verosimilitud y posterior

```
plot_normal_normal(mean = 6.5, sd = 0.4, sigma = 0.5,
                   y_bar = 6.4592, n = 25)
```



Nuevamente, dado que tenemos una verosimilitud cuyos datos estan tan concentrados, el posterior se pega mucho a este. En este caso la verosimilitud reafirma lo planteado por el prior, dado que sus medias son sumamente parecidas.

Es decir que en este caso, gracias al analisis bayesiano, reforzamos nuestro pensamiento inicial sobre donde se situa la media y ganamos precision.

```r
summarize_normal_normal(mean = 6.5, sd = 0.4, sigma = 0.5,
                        y_bar = 6.4592, n = 25)
```

```
      model   mean   mode         var         sd
1     prior 6.5000 6.5000 0.160000000 0.40000000
2 posterior 6.4616 6.4616 0.009411765 0.09701425
```

De esta manera confirmamos que nuestrio posterior calculado manualmente en la Parte 2 era correcto.