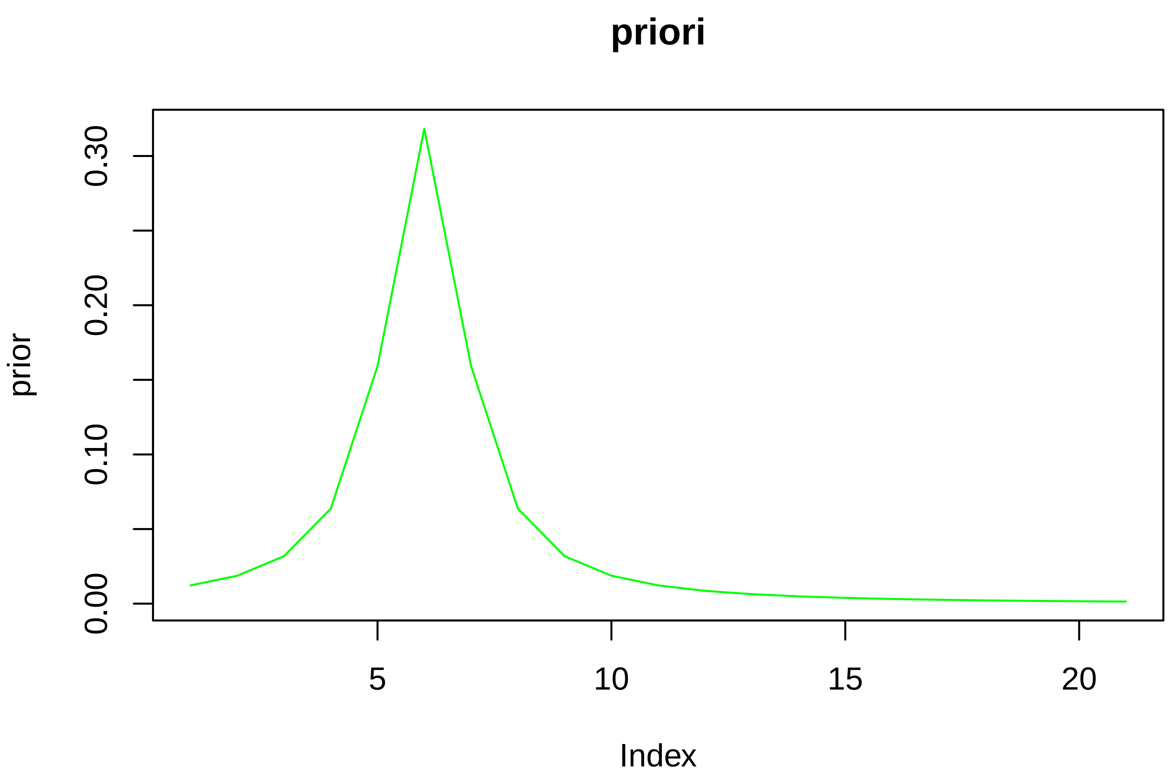
Labb3

Simon Jakobsson (simja649), Erik Halvarsson (eriha353), och Gustav Hanstorp (gusha433)

10/9/2020

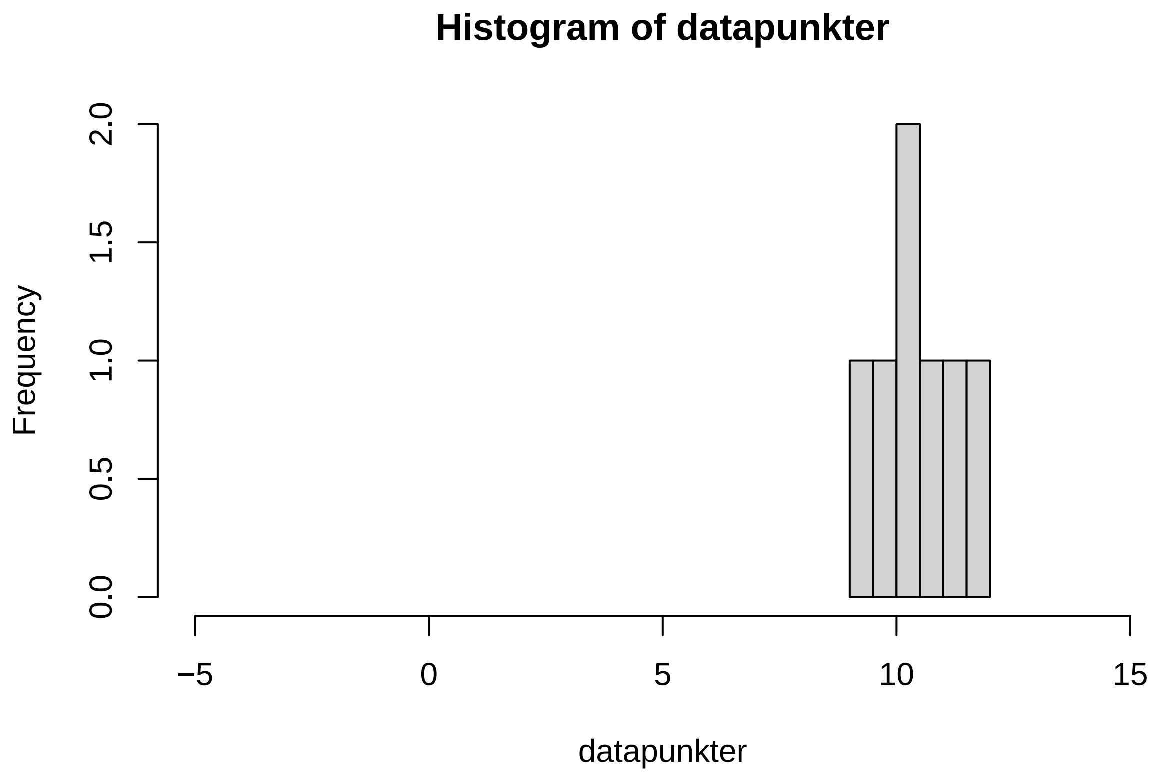
##3.1.1 ### Del 1 Vi använder os av de givna värdena och den givna funktionen och plottar den för:

```
prior <- dt(x = seq(-5, 15, 1), df = 1)
plot(prior, col="green",type="l",main="priori")
```



###Del 2 Vi skapar ett histogram med de givna datapunkterna och sätter intervallet från -5 till 15:

```
datapunkter <- c(11.3710, 9.4353, 10.3631, 10.6329, 10.4043, 9.8938, 11.5115)
hist(datapunkter, xlim = c(-5, 15))
```



Del 3 Vi använder oss av formeln från log_likelihoodfunktionen (som vi använde i lab2) och visar den över intervallet -5 till 15:

```
normal_log_likliehood <- function(mu, data){
  n <- length(data)
  return (-((n/2) * log(2 * pi)) - (n/2 * log(1)) - (sum((data - mu)^2)/2))
}
llik <- normal_log_likliehood(5, seq(-5, 15, 1))
round(llik, 1)
```

```
## [1] -404.3
```

Del 4

Vi börjar med formeln:

$$p(\mu|Y) \propto p(Y|\mu)p(\mu)$$

Från den ska vi nu härleda posteriorn. Vi behöver då ta reda på $p(Y|\mu)$ och $p(\mu)$. Vi börjar med att härleda $p(Y|\mu)$, vilket är formeln för en normalfördelning och ser ut som sådan:

$$p(Y|\mu) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

Från detta kan vi stryka alla faktorer som inte innehåller μ och ersätta σ med 1. Detta ger oss då:

$$p(Y|\mu) = \prod_{i=1}^{n} e^{-\frac{(y-\mu)^2}{2*1}}$$

Som, genom att göra om den multiplikativa summan till en summa i exponenten, där vi då kan bryta ut 1/2, kan förenklas till:

$$p(Y|\mu) = e^{-\frac{1}{2}\sum_{i=1}^{n}(y-\mu)^2}$$

Vi härleder nu $p(\mu)$, vilket är vår prior, som då är en t-fördelning där $\nu = 1$, och får:

$$p(\mu) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\Gamma(\frac{v}{2})}(1 + \frac{\mu^2}{v})^{-\frac{v+1}{2}}$$

Åter igen så kan vi ta bort alla faktorer som inte innehåller μ och ersätter ν med 1 och får:

$$p(\mu) = (1 + \mu^2)^{-1}$$

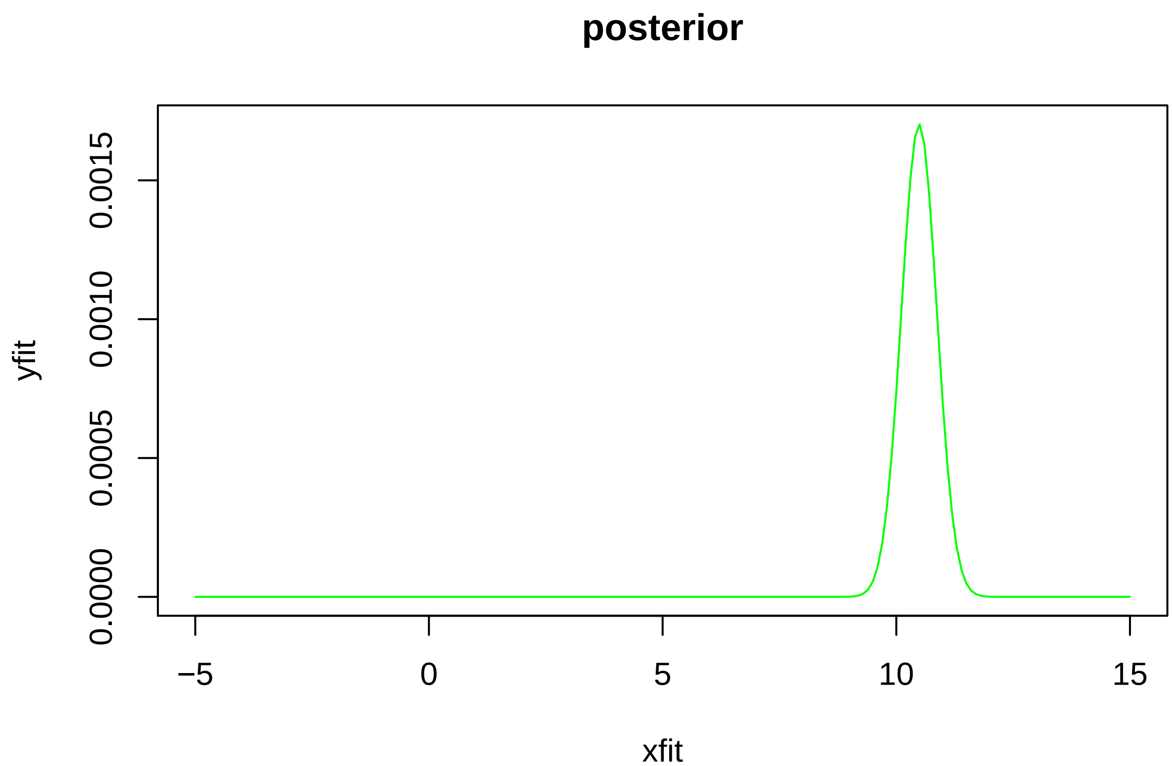
Nu kan vi härleda $p(\mu|Y)$ genom att sätta in $p(Y|\mu)$ och $p(\mu)$. Eftersom vi vet att något upphöjt -1 hamnar i nämnaren så kan vi skriva:

$$p(\mu|Y) \propto \frac{e^{-\frac{1}{2}\sum_{i=1}^{n}(y-\mu)^2}}{(1 + \mu^2)}$$

Del 5

Genom att använda oss av formeln vi får ovan och skapar en funktion så kan vi visualisera med samma intervall och datapunkter som i de tidigare uppgifterna och får:

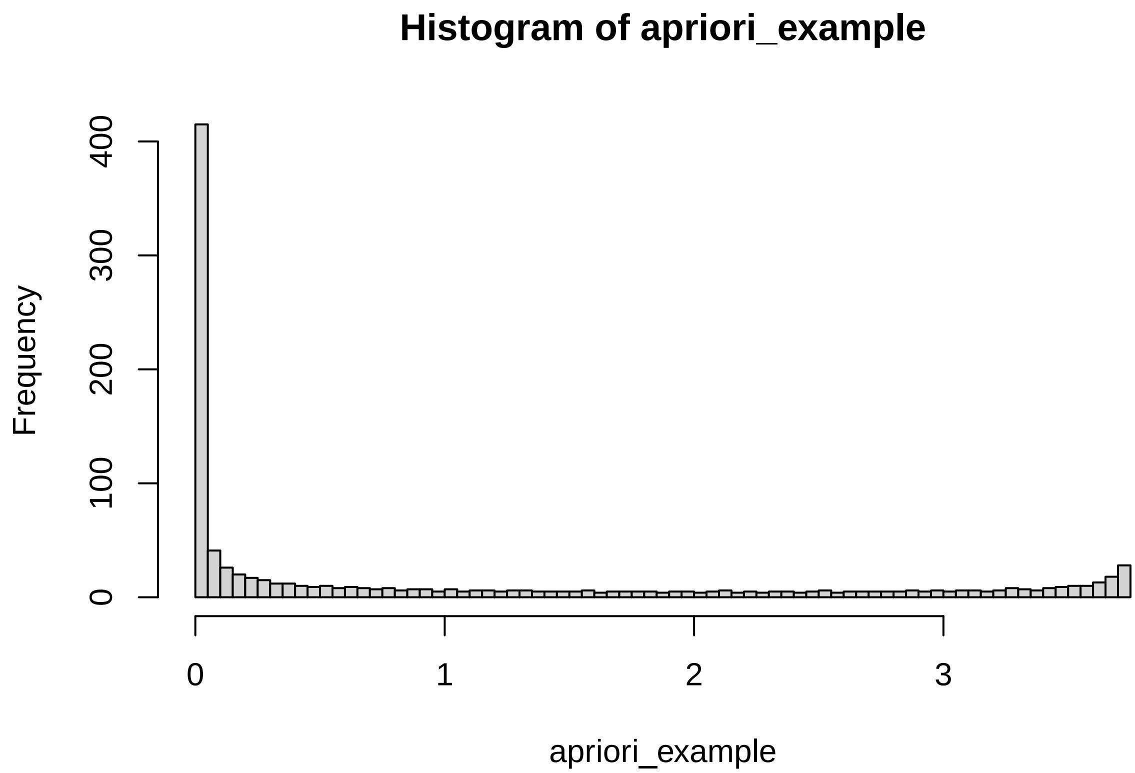
```r
mu_in <- seq(-5, 15, 1)
y_in <- c(11.3710, 9.4353, 10.3631, 10.6329, 10.4043, 9.8938, 11.5115)

onorm_posterior <- function(mu, y){
  return (exp((-1/2) * sum((y - mu)^2)) / (1 + mu^2))
}

xfit <- seq(-5,15, 0.1)
yfit <-lapply(xfit, function(mu_in) onorm_posterior(mu_in,y_in))
plot(xfit, yfit, col="green",type="l",main="posterior")
```



##3.2.1 ### Del 1

Enligt wikipedia så är $\alpha + \beta$ "sample size" (dock endast som prior i Bayes theorem) och således väljer vi att ha vår α som antalet lyckade demonstrationer och vår β som antalet misslyckade demonstrationer.

```
alpha_example = 13
beta_example = 7
apriori_example <- dbeta(seq(0, 1, 0.001), alpha_example, beta_example, ncp=0)
hist(apriori_example, 100)
```



Del 2

Enligt boken är posterior parametrarna för en betafördelning:

$$\alpha + \sum_{i=1}^{n} x_i \text{ och } \beta + nk - \sum_{i=1}^{n} x_i$$

Enligt Wikipedia räknas väntevärdet för betafördelningen ut genom följande ekvation: $E(X) = \frac{\alpha}{\alpha+\beta}$

Produkt A har värdena: $\alpha = 8,\ \beta = 5,\ n = 13,\ k = 1$. Vi får då väntevärdet för A: $E(X) = \frac{8}{8+5} \approx 0.615$.

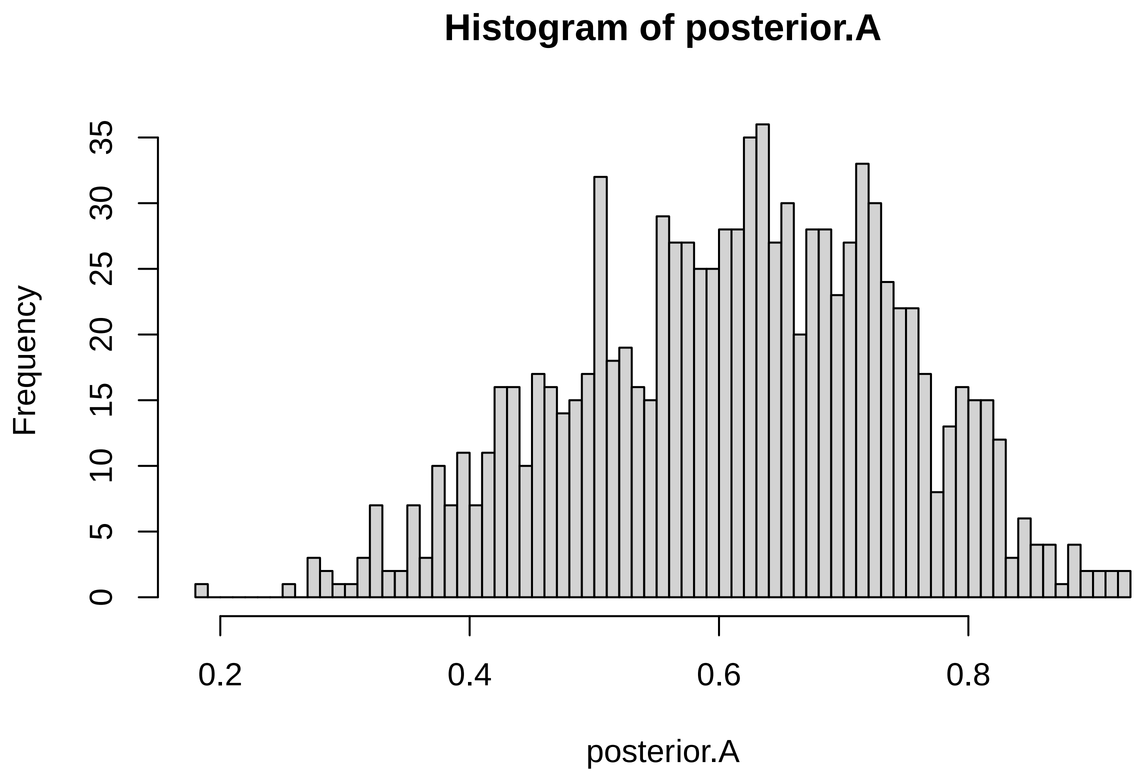
Produkt B har värdena: $\alpha = 2,\ \beta = 1,\ n = 3,\ k = 1$ Vi får då väntevärdet för B: $E(X) = \frac{2}{2+1} \approx 0.666$

Således får vi att produkt B har den högsta förväntade proportionen av intresserade.

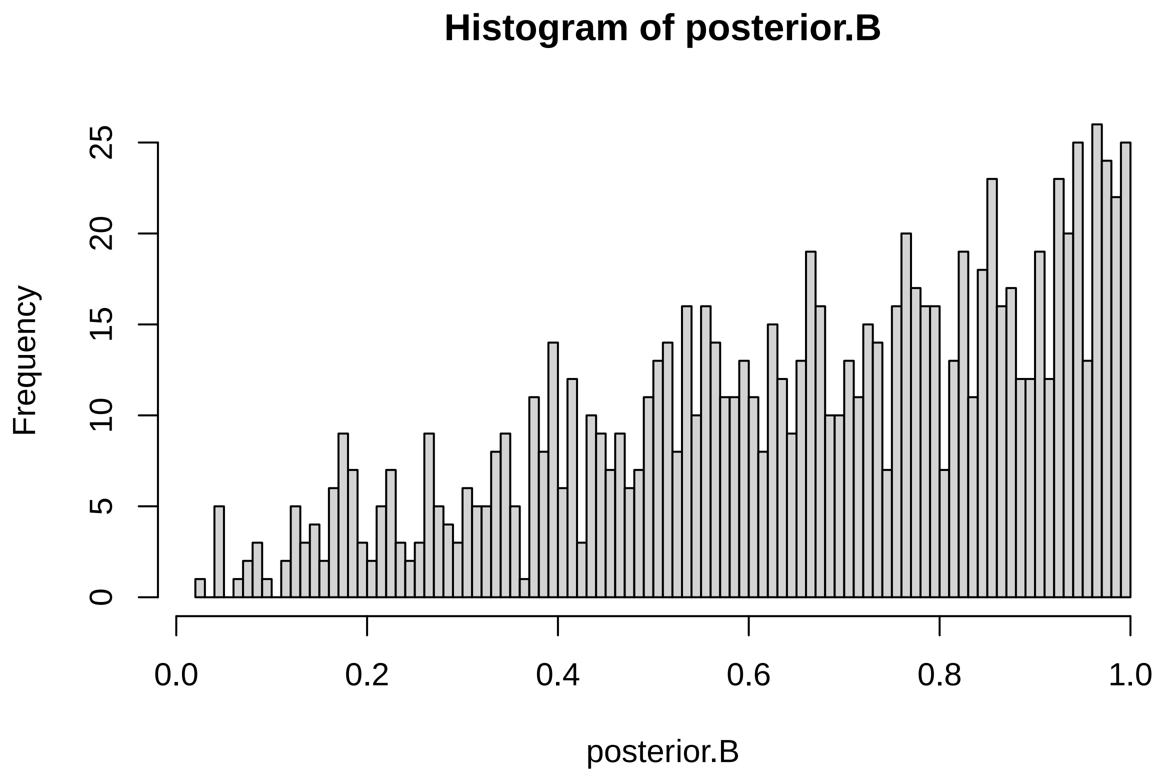
Del 3

Vi gör en simulering för posteriorbetafördelningen:

```
alpha.A = 8
beta.A = 5
posterior.A <- rbeta(seq(0.001, 1, 0.001), alpha.A, beta.A)
hist(posterior.A, 100)
```

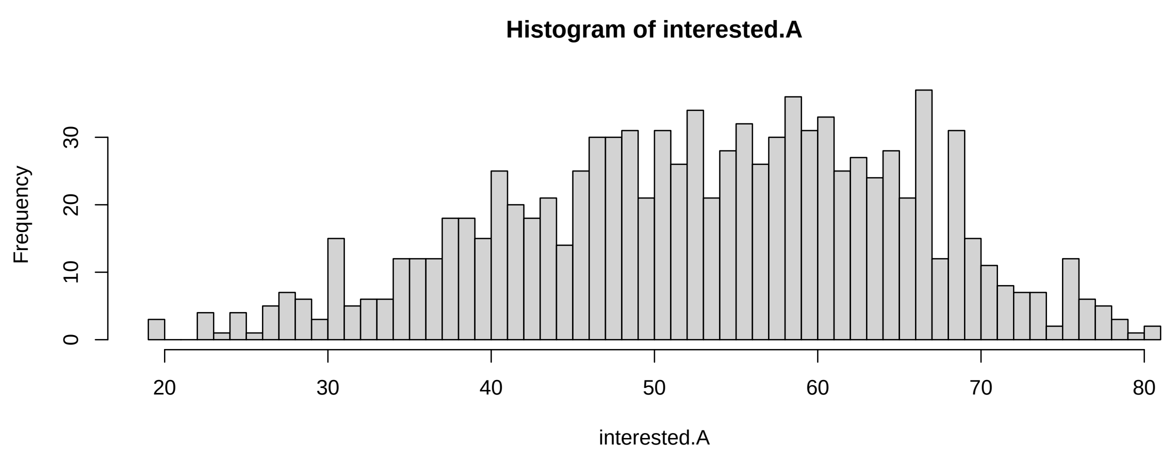



```
alpha.B = 2
beta.B = 1
posterior.B <- rbeta(seq(0.001, 1, 0.001), alpha.B, beta.B)
hist(posterior.B, 100)
```

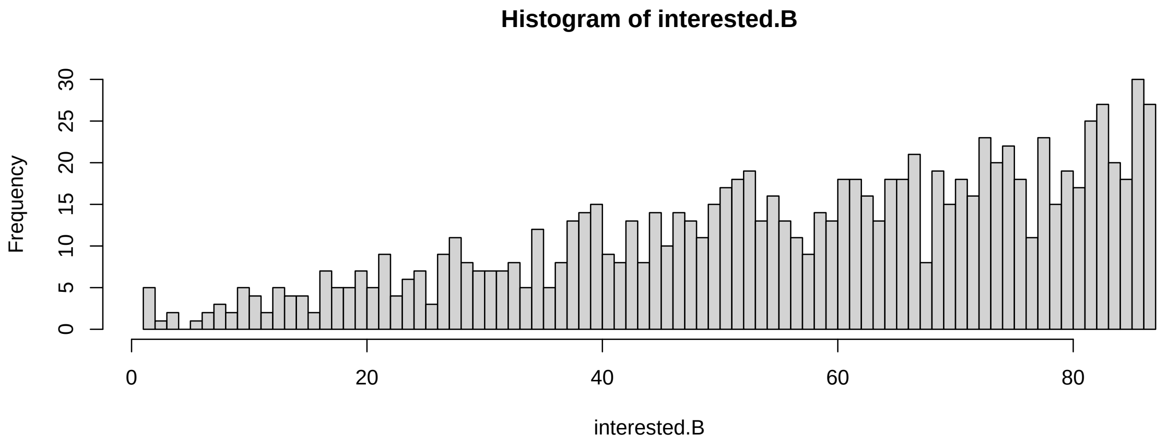


Sedan visualiserar vi med vår kundbas på 87:

```
interested.A <- rbinom(n = seq(0.001, 1, 0.001), size = 87, prob = posterior.A)
interested.B <- rbinom(n = seq(0.001, 1, 0.001), size = 87, prob = posterior.B)
hist(interested.A, 50)
```




```
hist(interested.B, 100)
```



Sannolikheten att mer än 40 är intresserade av respektive produkt:

```
sum(interested.A > 40)/1000
```

```
## [1] 0.847
```

```
sum(interested.B > 40)/1000
```

```
## [1] 0.771
```

Det förväntade antalet kunder för respektive produkt:

```
round(mean(interested.A), 0)
```

```
## [1] 54
```

```
round(mean(interested.B),0)
```

```
## [1] 58
```

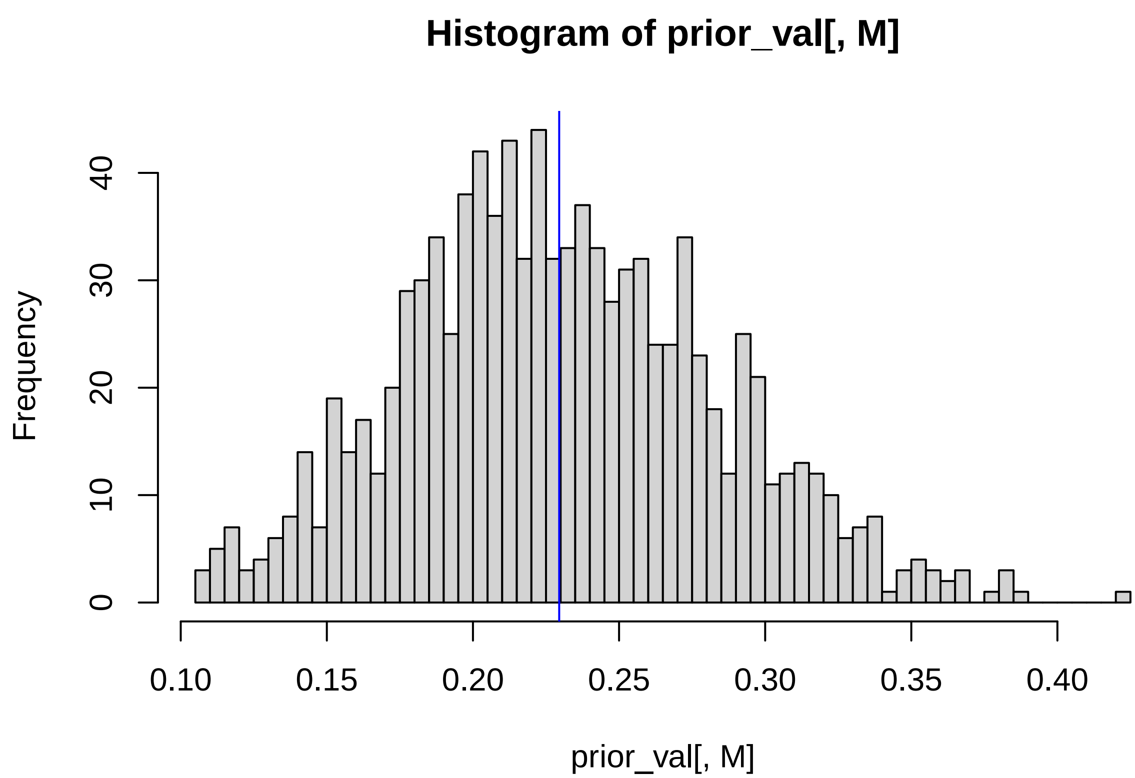
Uppgift 3.3.1

Del 1

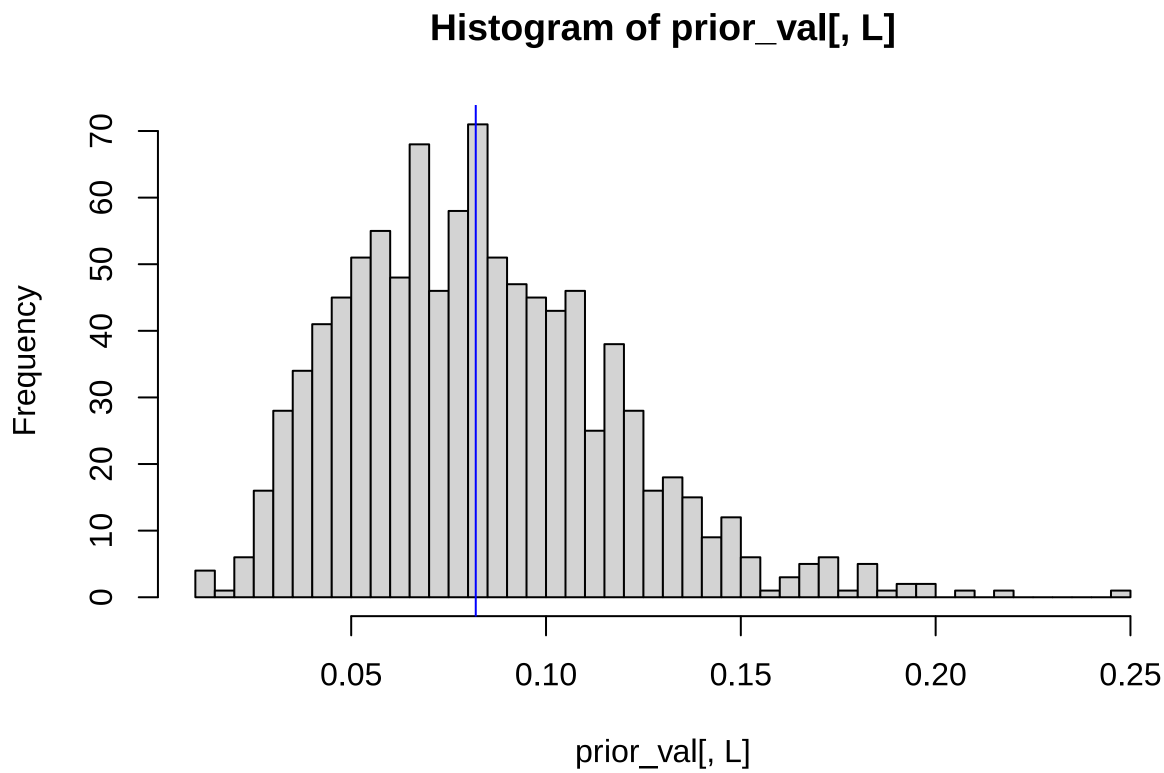
Vi bestämmer rimliga parametrar utifrån valet 2014 och visualiserar dessa:

```
library(gtools)
M <- 1
L <- 2
C <- 3
KD <- 4
S <- 5
V <- 6
MP <- 7
```

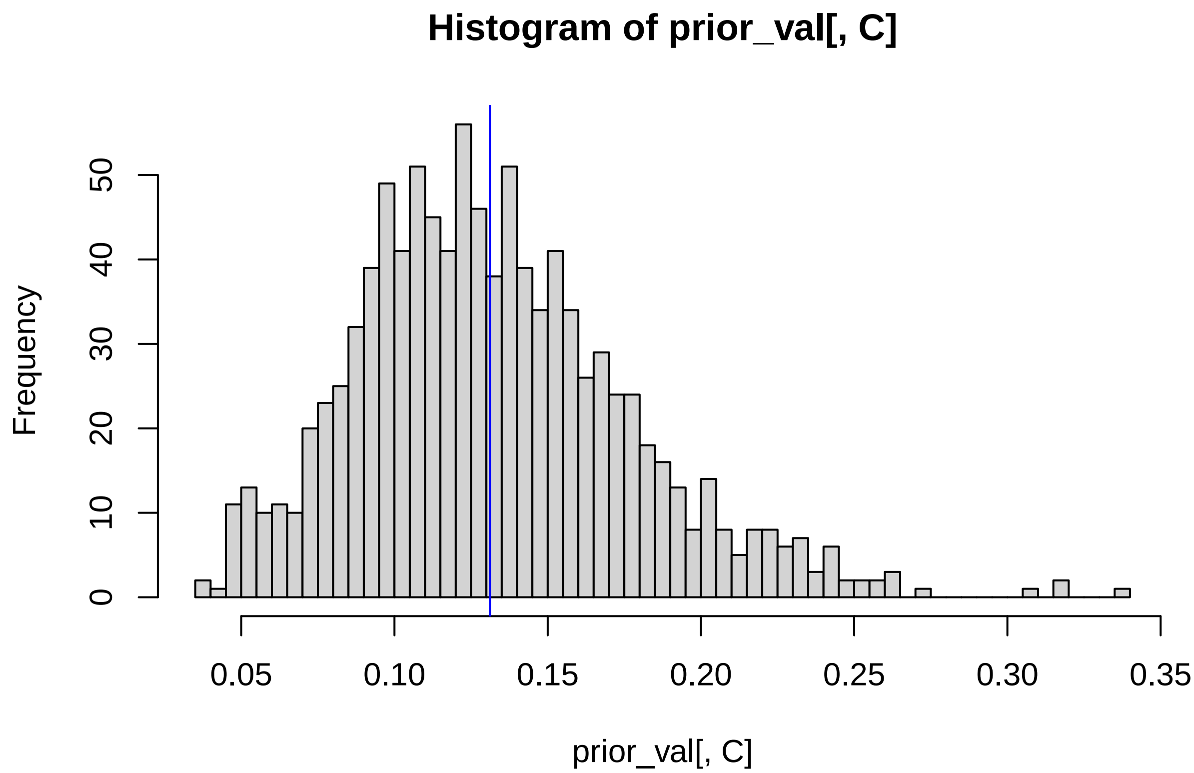
```
SD <- 8
FI <- 9


alpha = c(14, 5 ,8 ,3 ,20 ,2 , 4, 4, 1)

prior_val <-rdirichlet(n= 1000, alpha = alpha)

hist(prior_val[,M], 61)
abline(v = alpha[M]/61, col = "blue")
```



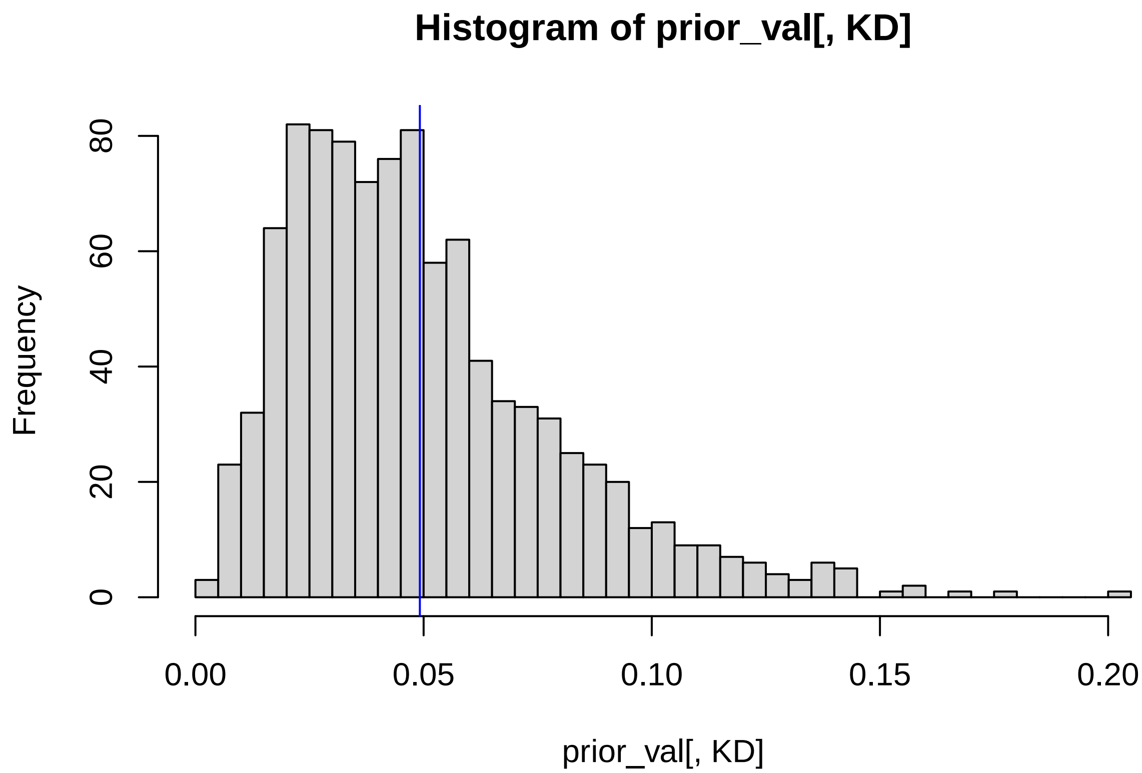

```
hist(prior_val[,L], 61)
abline(v = alpha[L]/61, col = "blue")
```



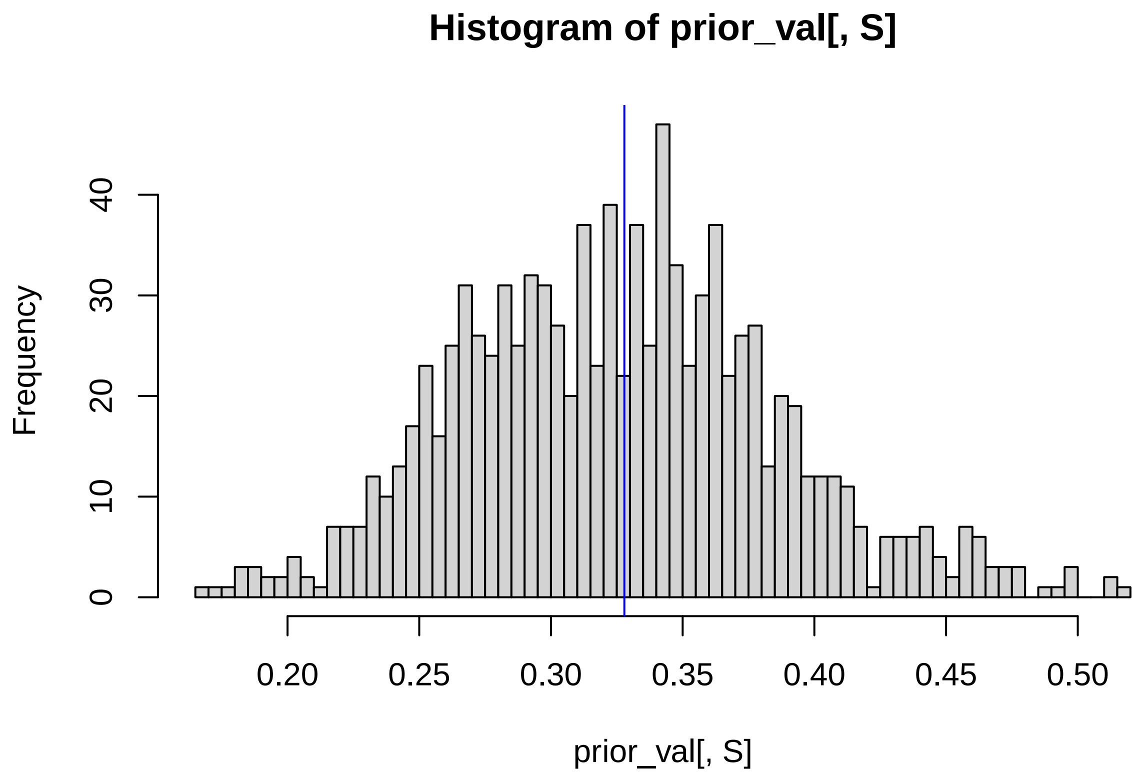

```
hist(prior_val[,C], 61)
abline(v = alpha[C]/61, col = "blue")
```



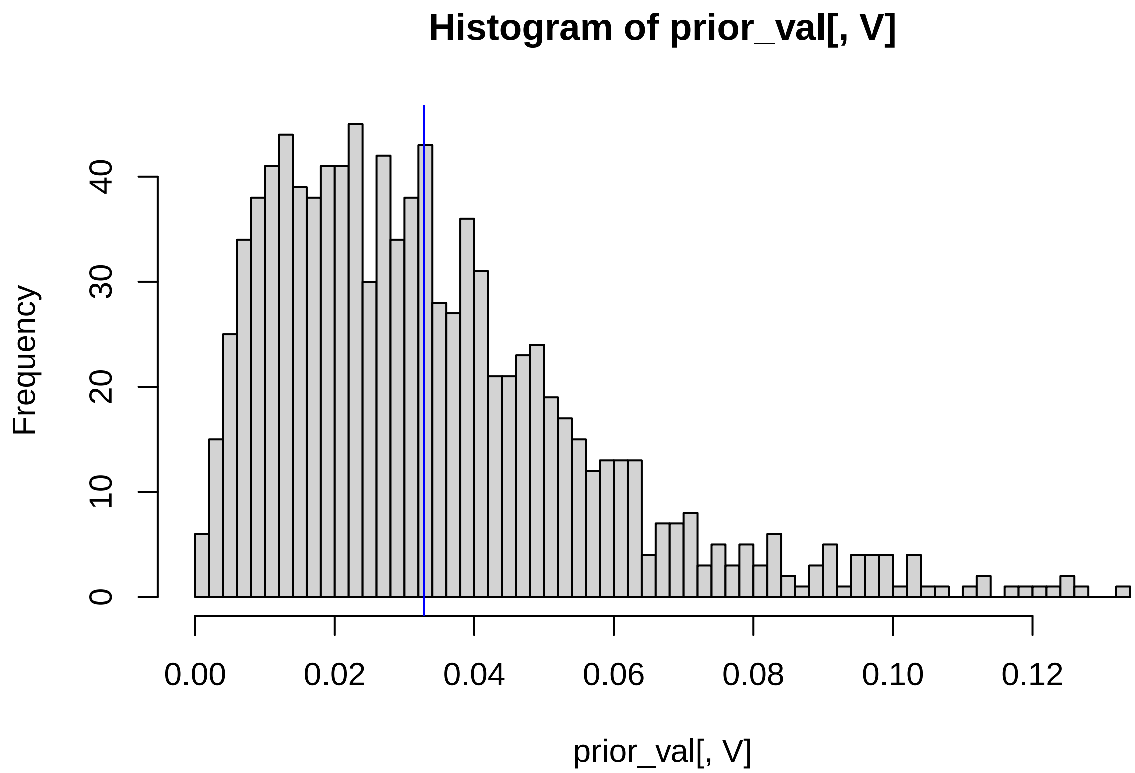

```r
hist(prior_val[,KD], 61)
abline(v = alpha[KD]/61, col = "blue")
```



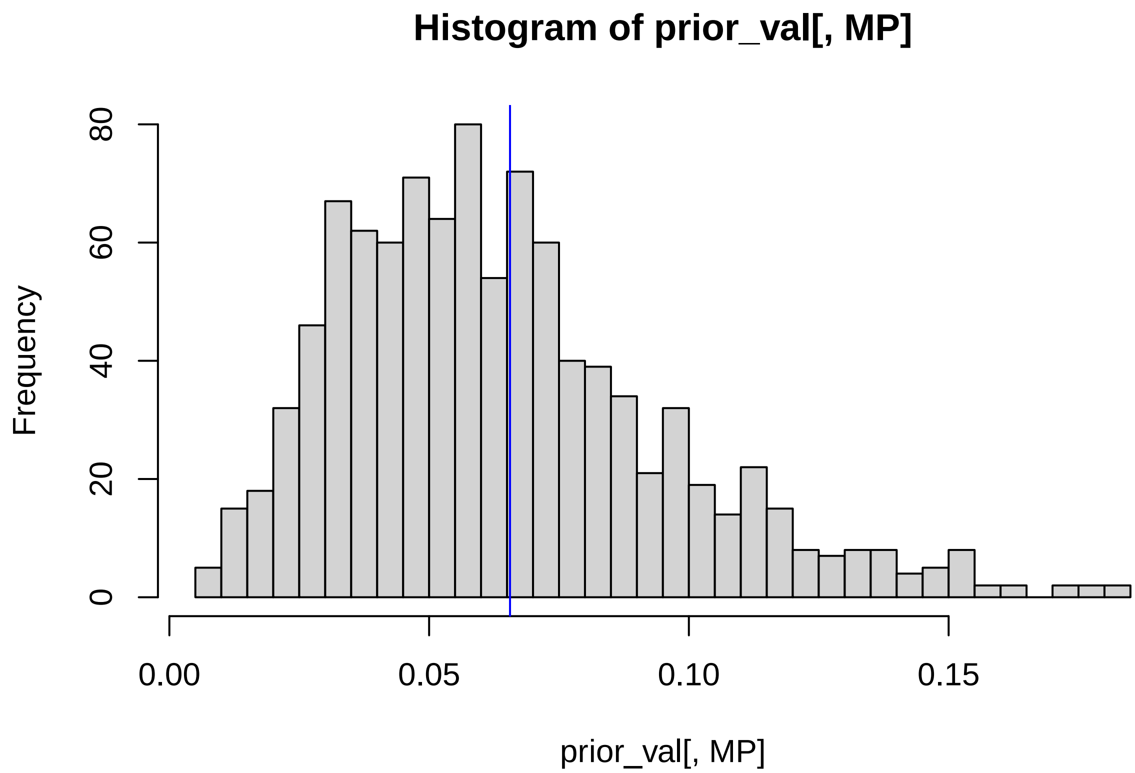

```r
hist(prior_val[,S], 61)
abline(v = alpha[S]/61, col = "blue")
```



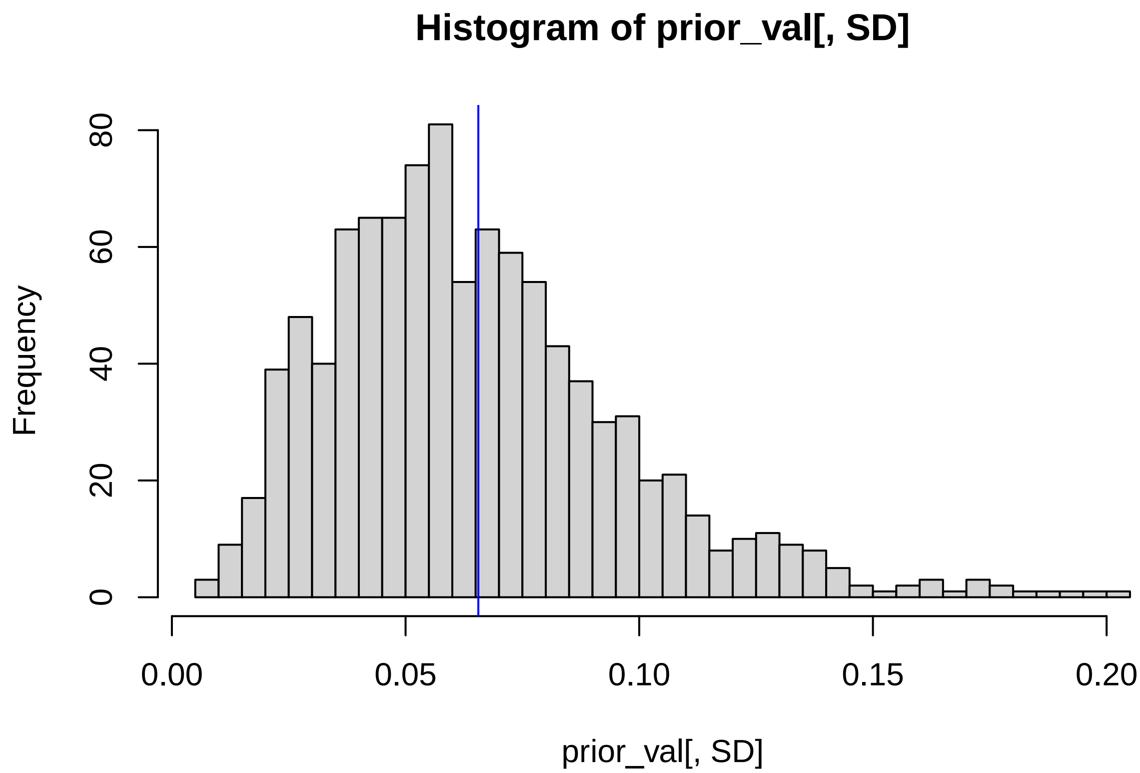

```
hist(prior_val[,V], 61)
abline(v = alpha[V]/61, col = "blue")
```



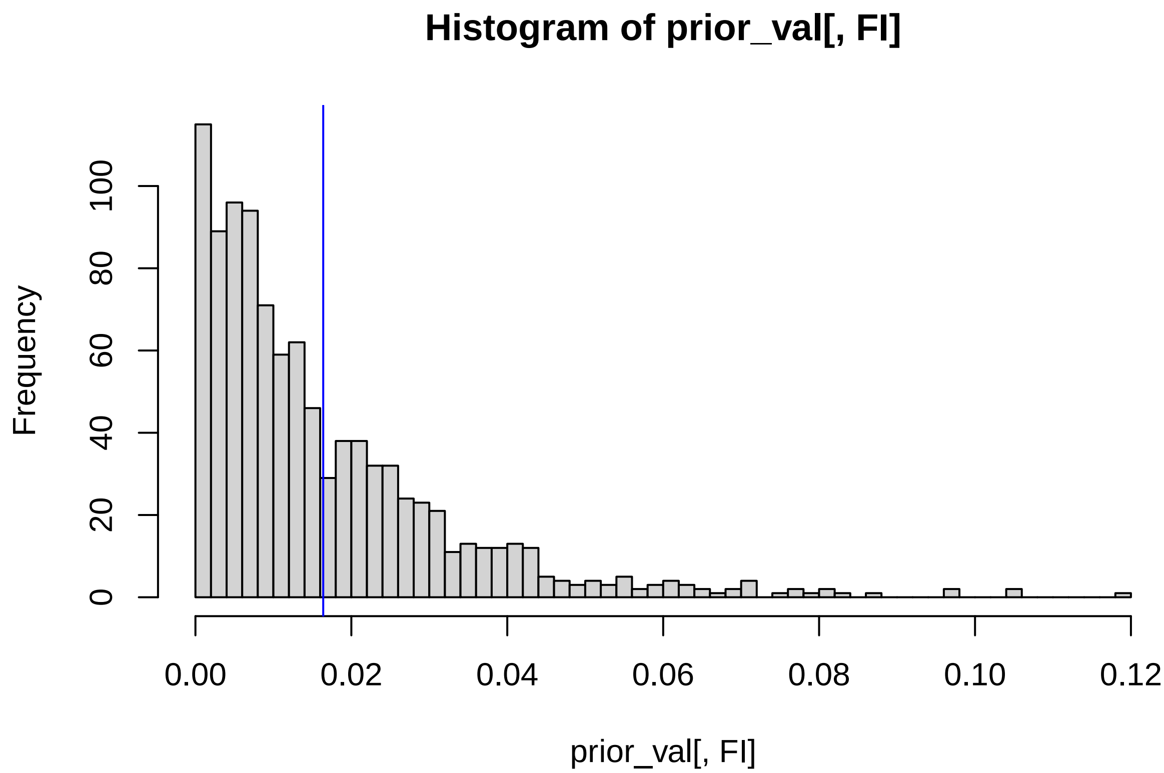

```r
hist(prior_val[,MP], 61)
abline(v = alpha[MP]/61, col = "blue")
```




```r
hist(prior_val[,SD], 61)
abline(v = alpha[SD]/61, col = "blue")
```




```
hist(prior_val[,FI], 61)
abline(v = alpha[FI]/61, col = "blue")
```



###Del 2

Vi använder oss av 2018-sep Skop datat då det är en av de senaste som faktiskt innehåller information om FI

För att räkna ut så tar vi % vad folk har svarat och multiplicerar med de 200 personer vi har valt ut, resulatet följer:

M: $0.176 * 200 \approx 35$

L: $0.065 * 200 \approx 13$

C: $0.079 * 200 \approx 16$

KD: $0.064 * 200 \approx 13$

S: $0.259 * 200 \approx 52$

V: $0.106 * 200 \approx 21$

MP: $0.049 * 200 \approx 10$

SD: $0.174 * 200 \approx 35$

FI $0.011 * 200 \approx 2$

Sammanlagt blev detta då 197 personer av 200. Detta är en anledning av avrundingar samt att vi inte räknar in gruppen "Uncertain".

###Del 3

En funktion för att sålla ut och omnormalisera om ett parti hamnar under 4%-gränsen:

```r
library(gtools)
post <- c(17.6, 6.5, 7.9, 6.4, 25.9, 10.6, 4.9, 17.4, 1.1)
dragningar <- rdirichlet(n= 10000, alpha = alpha + post)

sparr_normalisering <- function(drag){
  rader <- length(drag[,1])
  kolumner <- length(drag[1,])
  norm_dragning <- drag

  for (i in 1:rader){
    parlament_ratio <- 1.0
    for (j in 1:(kolumner-1)){
      if (drag[i,j] > 0 & drag[i,j] < 0.04){
      parlament_ratio <- parlament_ratio + (drag[i,j]/100)
      norm_dragning[i,j] <- 0
      }
    }
    for (j in 1:(kolumner -1)){
      norm_dragning[i,j] <- norm_dragning[i,j] * parlament_ratio
    }
  }
  return(norm_dragning)
}
```

Och delar upp alla dragningar fördelade till ett specifikt parti:

```r
omnormaliserade_dragningar <- sparr_normalisering(dragningar)

m_andelslista <- omnormaliserade_dragningar[,M]
l_andelslista <- omnormaliserade_dragningar[,L]
c_andelslista <- omnormaliserade_dragningar[,C]
kd_andelslista <- omnormaliserade_dragningar[,KD]
s_andelslista <- omnormaliserade_dragningar[,S]
v_andelslista <- omnormaliserade_dragningar[,V]
mp_andelslista <- omnormaliserade_dragningar[,MP]
sd_andelslista <- omnormaliserade_dragningar[,SD]
fi_andelslista <- omnormaliserade_dragningar[,FI]
```

a) Delar vi upp fördelningen av rösterna inom blocken och ställer alla itereringar mot varandra och delar på 10000 får vi:

```r
a_andelslista <- m_andelslista + l_andelslista + c_andelslista + kd_andelslista
rg_andelslista <- s_andelslista + v_andelslista + mp_andelslista

sum(rg_andelslista > a_andelslista)/10000
```

```
## [1] 0.4639
```

b) Av alla 10000 dragningar kollar vi hur många dragningar SD är större än M och delar på 10000:

```r
sum(sd_andelslista > m_andelslista)/10000
```

```
## [1] 0.0791
```

c) Av alla 10000 dragningar kollar vi hur många dragningar KD är mindre än gränsen och delar på 10000:

```r
sum(kd_andelslista < 0.04)/10000
```

```
## [1] 0.1506
```

d) Av alla 10000 dragningar kollar vi hur många dragningar MP är mindre än gränsen och delar på 10000:

```r
sum(mp_andelslista < 0.04)/10000
```

```
## [1] 0.1979
```

e) För att beräkna konfidensintervallet på 95% så måste vi ha en lägre gräns på 2.5% och en övre gräns på 97.5 då det ger ett intervall på 95%, vilket ger oss:

```r
lagre_grans <- quantile(prob = 0.025, x = s_andelslista)
lagre_grans
```

```
##      2.5%
## 0.2208671
```

```r
ovre_grans <- quantile(prob = 0.975, x = s_andelslista)
ovre_grans
```

```
##     97.5%
## 0.3616756
```