Aufgabe 3

Wir betrachten ein Binomial-Experiment. Sei $x \sim B(n, \pi)$. Wir betrachten im Folgenden zwei Ansätze:

(1) π ist stetig
(2) π sei diskret und nehme Werte auf einem Gitter $\{0.00, 0.01, 0.02, \ldots, 0.99, 1\}$ an.

(a) Ausgehend von Laplace' Prinzip vom unzureichenden Grund: Wie sieht in beiden Fällen die Priori von π aus?

(b) Leiten Sie jeweils die Posteriori für $\pi|x$ her.

(c) Berechnen Sie jeweils den Posteriori-Erwartungswert und den Posteriori-Median (Sie können auch R oder eine andere Programmiersprache benutzen) für folgende Daten:
- $n = 10, x = 3$
- $n = 100, x = 13$
- $n = 1000, x = 33$

(d) Vergleichen Sie die Posteriori-Erwartungswerte und -Mediane mit beiden Ansätzen.

(e) Welchen Ansatz würden Sie eher bevorzugen?

Lösung

a) stetige Priori: $\pi \sim U[0, 1]$

diskrete Priori:

$$p(\pi) = \frac{1}{101} I_{\Pi}(\pi)$$

mit $\Pi = 0, 0.01, 0.02, 0.03, \ldots, 0.99, 1$

b) stetige Priori: Siehe Billard-Beispiel.

$$\pi|x \sim Beta(x + 1, n - x + 1)$$

diskrete Priori:

$$p(\pi) \propto \pi^x 1 - \pi^{n-1}$$

c) Stetige Priori, Erwartungswert der Beta(a, b)-Verteilung $\pi|\curvearrowleft = \frac{a}{a+b}$, also hier $\pi|\curvearrowleft = \frac{x+1}{n-x+1+x+1} = \frac{x+1}{n+2}$

- $n = 10, x = 3 \rightsquigarrow \pi|\curvearrowleft = \frac{4}{12} = \frac{1}{3}$
- $n = 100, x = 13 \rightsquigarrow \pi|\curvearrowleft = \frac{14}{102} \approx 0.1372549$
- $n = 1000, x = 33 \rightsquigarrow \pi|\curvearrowleft = \frac{34}{1002} \approx 0.03393214$

Median per R:

```
qbeta(0.5,4,8)
```

```
## [1] 0.3238045
```

```r
qbeta(0.5,14,88)
```

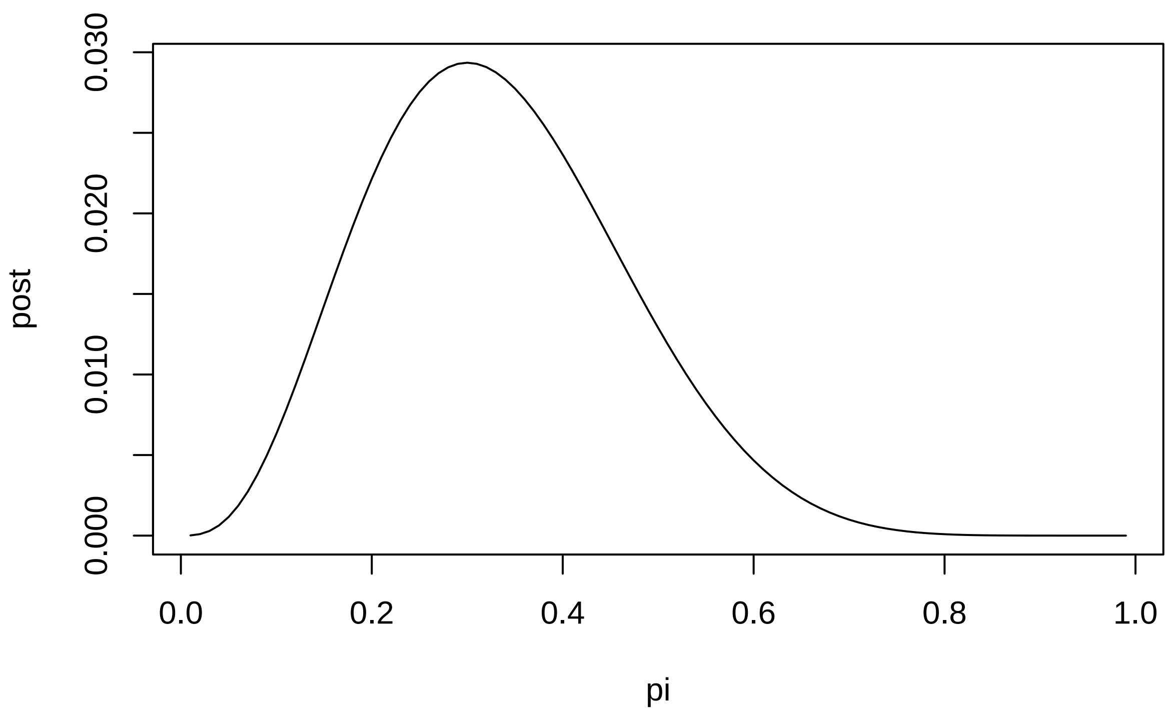
```
## [1] 0.1348812
```

```r
qbeta(0.5,34,968)
```

```
## [1] 0.03362241
```

$$p(\pi) \propto \pi^x 1 - \pi^{n-1}$$

```r
pi<-seq(0.01,0.99,by=0.01) # ignoriere 0 und 1
prior=rep(1/101,99)
# für x=3 n=10
x=3; n=10
logpost=x*log(pi)+(n-x)*log(1-pi) # log posteriori, Logarithmus aus numerischen Gründen
post<-exp(logpost)
post<-post/sum(post) # Normalisierung
plot(pi,post,type="l")
```



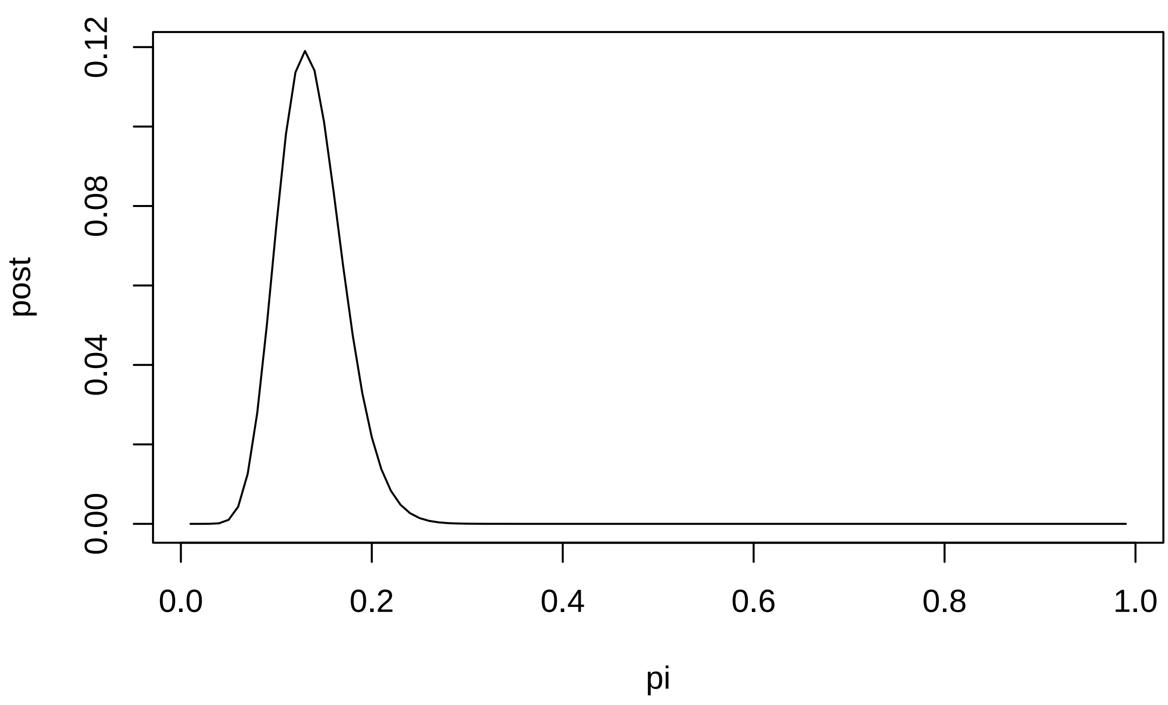
```r
# Erwartungswert
(E = sum(pi*post))
```

```
## [1] 0.3333333
```

```r
# Median
verteilungsfunktion <- cumsum(post)
pi[min(which(verteilungsfunktion>=0.5))]
```

```
## [1] 0.32
```

```r
# für x=13 n=100
x=13; n=100
logpost=x*log(pi)+(n-x)*log(1-pi) # log posteriori, Logarithmus aus numerischen Gründen
post<-exp(logpost)
post<-post/sum(post) # Normalisierung
plot(pi,post,type="l")
```



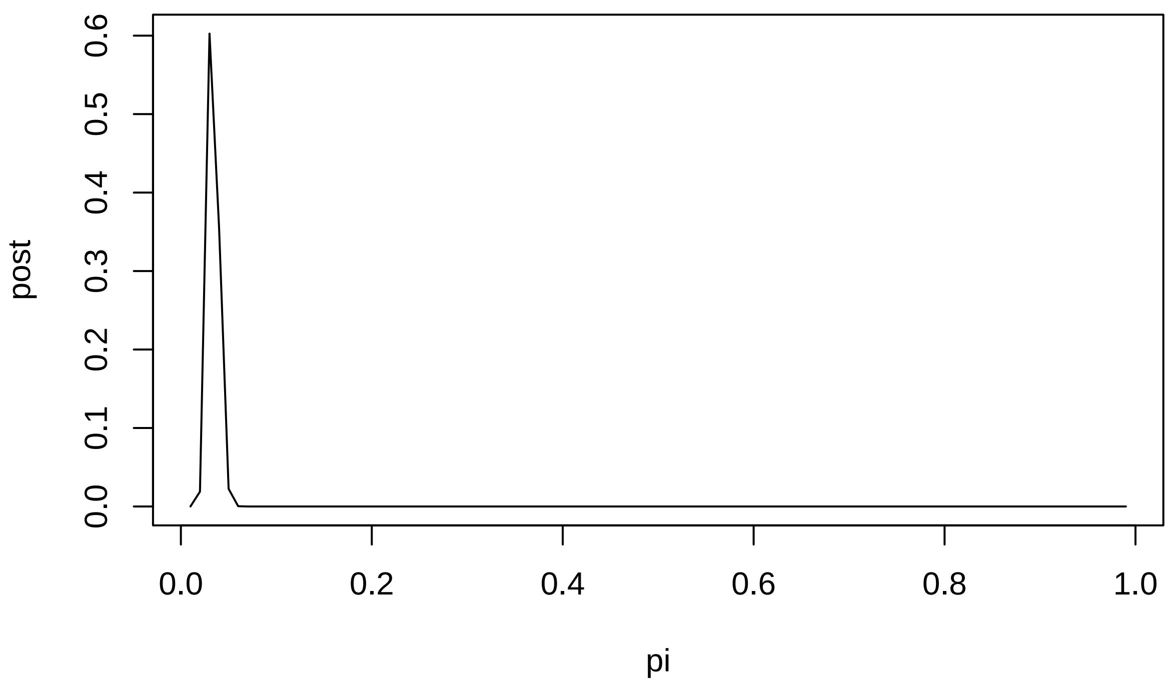

```r
# Erwartungswert
(E = sum(pi*post))
```

```
## [1] 0.1372549
```

```r
# Median
verteilungsfunktion <- cumsum(post)
pi[min(which(verteilungsfunktion>=0.5))]
```

```
## [1] 0.13
```

```r
# für x=33; n=1000
x=33; n=1000
logpost=x*log(pi)+(n-x)*log(1-pi) # log posteriori, Logarithmus aus numerischen Gründen
post<-exp(logpost)
post<-post/sum(post) # Normalisierung
plot(pi,post,type="l")
```




```r
# Erwartungswert
(E = sum(pi*post))
```

```
## [1] 0.03382706
```

```r
# Median
verteilungsfunktion <- cumsum(post)
pi[min(which(verteilungsfunktion>=0.5))]
```

```
## [1] 0.03
```