Hausübung 1 Angewandte Bayes Statistik

Philipp Lintl

22 Mai 2017

Einlesen der Daten, Bestimmung der Summe und Anzahl der Daten.

```
kaiser <- read.table("kaiserschnitt.raw.txt",header=T)
daten  <- kaiser[,1]
sum_x <- sum(daten)
n <- length(daten)
```

Posteriori

Aufgabe a

Als Daten wird die erste Spalte der Datei 'kaiserschnitt.raw.txt' verwendet. Für den Fall a sollten 1000 Zufallszahlen aus der Posteriori- Verteilung (a=b=1) gezogen werden, anschließend analytisch der Posteriori Erwartungswert berechnet werden und abschließend mit dem Erwartungswert (Mittelwert) der 1000 gezogenen Zufallszahlenverglichen werden.
$x_1, ..., x_n iid$
$X_i \sim Poi(\lambda)$.
Priori: $\lambda \sim GA(a, b)$
Formel für Posteriori nach Skript:

$$p(\lambda|x) \propto f(x_i|\lambda) * p(\lambda)$$

Datendichte:

$$f(x_i|\lambda) = \frac{\lambda^{x_i}}{x_i!} * exp(-\lambda)$$

$$f(x|\lambda) \stackrel{iid}{=} \prod_{i=1}^{n} f(x_i|\lambda)$$

$$\propto \prod_{i=1}^{n} \frac{\lambda^{x_i}}{x_i!} * exp(-\lambda)$$

$$\propto \frac{\lambda^{\sum_{i=1}^{n} x_i}}{\prod_{i=1}^{n} x_i} * exp(-n\lambda)$$

$$\propto \lambda^{\sum_{i=1}^{n} x_i} * exp(-n\lambda)$$

Priori:

$$\lambda \sim GA(a, b) p(\lambda) = \frac{b^a}{\tau(a)} * \lambda^{a-1} * exp(-b\lambda)$$

$$\propto \lambda^{a-1} * exp(-b\lambda)$$

Posteriori:
$$p(\lambda|x) \propto f(x|\lambda) * p(\lambda)$$

$$\propto \lambda^{a-1} * exp(-b\lambda) * \lambda^{\sum_{i=1}^{n} x_i} * exp(-n\lambda)$$

$$\propto \lambda^{(\sum_{i=1}^{n} x_i + a)-1} * exp(-\lambda(n+b))$$

Diese Form eintspricht dem Kern einer Gamma-Verteilung $GA(\tilde{a}, \tilde{b})$. Mit $\tilde{a} = \sum_{i=1}^{n} x_i + a$ und $\tilde{b} = b + n$.
Aufgabe b
Insgesamt also $p(\lambda|x) \sim GA(\sum_{i=1}^{n} x_i + a, b+n)$. Daten x_i bleiben wie in a) verteilt. Der Parameter ist nun jedoch $\lambda \sim LN(\mu, 1)$ verteilt. Für die Datendichte wird die in a) bereits proportionale Form verwendet.
Posteriori:

$$p(\lambda|x) \propto f(x|\lambda) * p(\lambda)$$

$$\propto \lambda^{\sum_{i=1}^{n} x_i} * exp(-n\lambda) * \frac{1}{\sqrt{2\pi} * \lambda} * exp(-\frac{1}{2} * (\log\lambda - \mu)^2)$$
$$\propto \lambda^{\sum_{i=1}^{n} x_i} * \frac{1}{\lambda} * exp(-n\lambda) * exp(-\frac{1}{2} * (\log\lambda)^2) * exp(\mu * \log\lambda) * exp(-\frac{1}{2} * \mu^2)$$
$$\propto \lambda^{\sum_{i=1}^{n} x_i - 1} * exp(\log(\lambda^{\mu})) * exp(-\frac{1}{2} * (\log\lambda)^2) * exp(-n\lambda)$$
$$\propto \lambda^{\sum_{i=1}^{n} x_i - 1 + \mu} * exp(-n\lambda - \frac{1}{2} * (\log\lambda)^2)$$

Diese Posteriori im Fall b), also $p(\lambda|x)$ entspricht keinem Kern einer bekannten Verteilung.

Samplen

Aufgabe a
Verwenden der für a) gewonnenen Posteriori $\mu|x \sim Ga(a + \sum_{i=1}^{n} x_i, b+n)$
Aus dieser bekannten Verteilung (Gamma) kann gezogen werden. Die Parameter waren in der Aufgabenstellung gegeben. a_zz ist der gesuchte Vektor auf 1000 gezogenen Zufallszahlen aus der aus a) gewonnenen Posteriori. In der Angabe waren a=b=1 festgesetzt.

```
a_1 <- 1
b_1 <- 1
set.seed(1000)
a_zz <- rgamma(1000,shape=a_1+sum_x,rate=b_1+n)
mean_a_zz <- mean(a_zz)
mean_a_zz
```

```
## [1] 10.0636
```

Erwartungswert analytisch aus der Verteilungstabelle gammaverteilter Zufallsvariablen: $\mathbb{E}(X|a,b) = \frac{a}{b}$.

```
# Analytisch: Verteilungstabelle:
a_tilde<- a_1+sum_x
b_tilde <- b_1+n
post_erw <- a_tilde/b_tilde
post_erw
```

```
## [1] 10.08
```

Theoretisch: 10,08
Aus der Stichprobe: 10,0636

Somit kommt der aus der Stichprobe stammende Erwartugswert dem theoretischen sehr nahe und würde bei größerem Stichprobenumfang wahrscheinlich noch näher daran liegen. ***Aufgabe b***
Ziehen aus der Posteriori, die keiner bekannten Verteilung entspricht. Dafür verwendet werden soll das sogenannte Acception Rejection Verfahren. Nach Tipp auf dem Angabenblatt sollte die aus Aufgabenteil a gewonnene Posteriori verwendet werden. Über die Parameter wurde keine Aussage getroffen, also wählte ich nach 'Herumprobieren' die Werte a=2.4 und b=0.5, da hierfür die Posteriori relativ symmetrisch darunter passt. Wählt man wie in Aufgabe 2a) a=b=1, so sind Posteriori und Vorschlagsdichte leicht verschoben. Dann wäre es schwierig, die Posteriori 'gut anzupassen', da sonst die Vorschlagsdichte nicht komplett darüber liegen würde, bzw. viele gezogene Werte abgelehnt werden müssten, da sich die Poseriori nicht wirklich knapp an die Vorschlagsdichte annähern lässt.
Also $\lambda \sim Ga(\tilde{a}, \tilde{b})$ verteilt. Nun gilt es also ein c zu finden, das garantiert, dass

$$c * p(\lambda_a|x) >= p(\lambda_b|x) \ \ \forall x$$

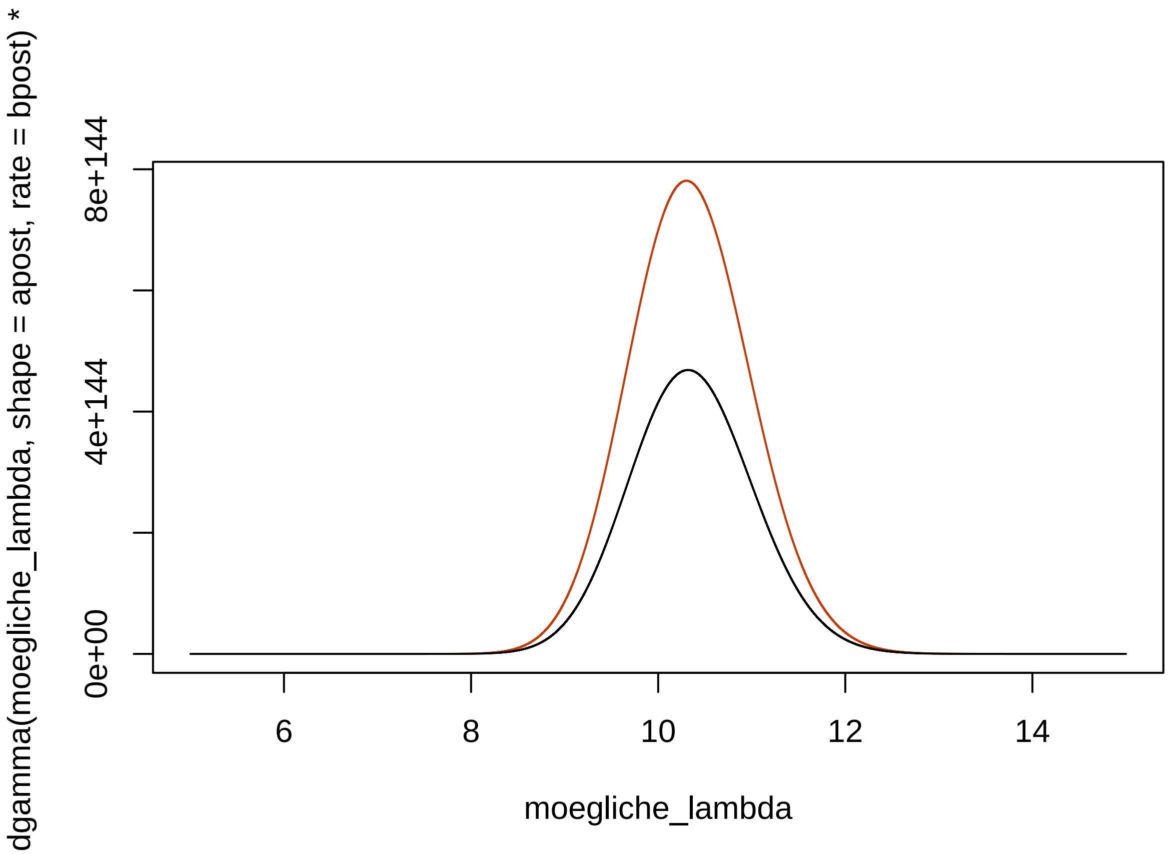
Dieses c fand ich nach ein wenig 'herumprobieren' und einschätzen in welchen Wertebereichen sich die beiden Dichten bewegen.
Aus der Posteriori aus a) kann gezogen werden. Wenn dies garantiert ist, wird zuerst eine Zufallszahl λ_i aus $p(\lambda_a|x)$ gezogen. Anschließend wird dieses λ_i in die Posteriori $p(\lambda_b|x) = y_1$ und in die Posteriori $p(\lambda_a|x) = y_2$ eingesetzt. Aus diesen Werten wird ein Quotient gebildet:

$$\frac{y_1}{y_2} = prop$$

Danach wird eine Zufallszahl $u \sim U[0, 1]$ gezogen.
Falls $u <= prop$ wird das λ_i als Zufallszahl akzeptiert und in den Sample Vektor aufgenommen. Das ganze wird dann solange wiederholt, bis 1000 Zufallszahlen akzeptiert wurden. Um zu überprüfen, ob die Vorschlagsdichte wirklich immer größer gleich der Posteriori ist, wurden Zahlen von 5-15 (festgelegt, da aus erstem Plot ersichtlich, dass ausserhalb nichts mehr liegt) in 0.0001er Schritten in die Vorschlagsdichte und in die Posteriori eingesetzt. Anschließend eine Differenz der Vektoren gebildet und getestet, ob diese größer gleich 0 ist. Für meine gewählten a,b und c war dies der Fall, die Summe ergab also 100001, was bedeutet, dass es nur True Werte gab und somnit die Vorschlagsdichte immer überhalb der Posteriori liegt.

```
a_2 <- 2.4                                    # Priori-Parameter
b_2 <- 0.5                                    # Priori-Parameter
apost <- a_2 + sum_x                          # Posteriori-Parameter
bpost <- b_2 + n                              # Posteriori-Parameter
# Aus Angabe gegeben
mu <- 0
# Intervall von 5-15 in 0,0001er Schritten, um alle moeglichen lambda Werte abzudecken
moegliche_lambda <- seq(5,15, by= 0.0001)
# Posteriori aus b)
postJoint <- function(mu, moegliche_lambda, x)
# Aus analytischer Form der Posteriori (b)
{
  z <- length(x)
  summe <- sum(x)
  return(moegliche_lambda^(summe+mu-1)*
           exp((-1/2)*((log(moegliche_lambda)^2)+2*z*moegliche_lambda)))
}
# Selbst gewählte Konstante, die beide Dichten in einen Bereich bringt
c <- 1.27*10^145
# Berechnet Posteriori an den Stellen Lambdas
y <- postJoint(mu, moegliche_lambda, daten)
# Plotten der Vorschlagsdichte
plot(moegliche_lambda, dgamma(moegliche_lambda, shape = apost, rate = bpost)*c,
```

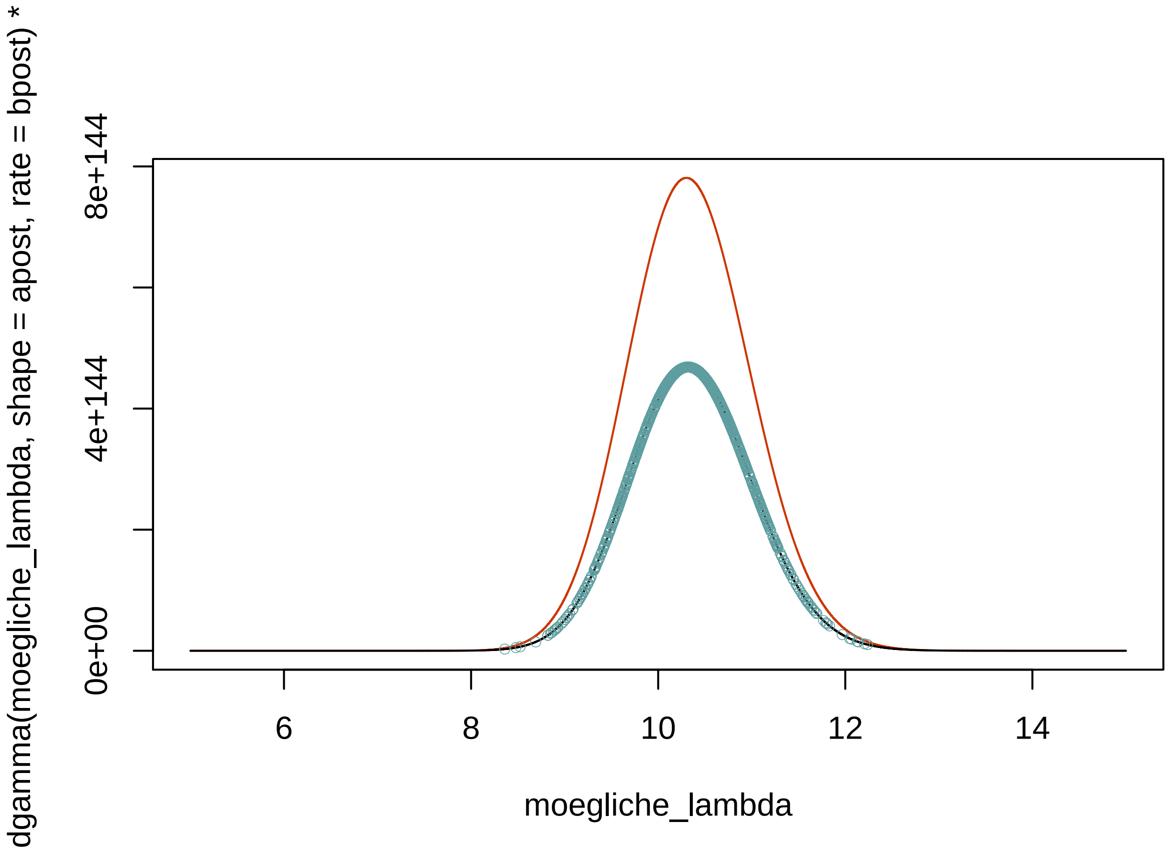
```r
     type = "l", col = "orangered3", xlim = c(5,15))
# Einzeichnen der Posteriori
lines(moegliche_lambda, y)
```




```r
# Stichprobenvektor, in dem erfolgreiche Ziehungen gespeichert werden
sample_1 <- c()
# 1000 ZZ zu ziehen
N<-1000
# Zählvariable um Anzahl abgelehnter ZZ zu zählen
i <- 0
# Ziehvorgang inklusive Gleichverteilter ZZ
set.seed(1000)
while(N!=0)
{
  cat(".")
  prop <- rgamma(1, shape = apost, rate = bpost)
  alpha <- postJoint(mu, prop, daten)/(dgamma(prop, shape = apost, rate = bpost)*c)
  u <- runif(1,min=0,max=1)
  if(alpha>=u){
    sample_1 <- c(sample_1, prop)
  }
  N <- 1000-length(sample_1)
  i <- i + 1
}
```

```
## ..........................................................................................
```

```r
# Posteriori-Werte der gezogenen Stichprobe
yy <- postJoint(0, sample_1, daten)
# Nochmals Zeichnen beider DIchten plus gezogene Lambda Werte
plot(moegliche_lambda, dgamma(moegliche_lambda, shape = apost, rate = bpost)*c,
     type = "l", col = "orangered3", xlim = c(5,15))
# Einzeichnen der Posteriori
lines(moegliche_lambda, y)
# Einzeichnen der Stichprobenwerte
points(sample_1,yy,col="cadetblue",cex=.7,lwd=.4)
```




```r
# Ueberpruefen, ob Vorschlagsdichte tatsaechlich immer groesser Posteriori
# Alle Lambdawerte in die Vorschlagsdichte eingesetzt muessen goesser gleich der Posteriori sein
l <- dgamma(moegliche_lambda, shape = apost, rate = bpost)*c
# Differenz der Funktionswerte beider Dichten (100001 Elemente)
ll <- l-y
# Test, an welchen Stellen Posteriori groesser -> Differenz negativ
ll <- ll >= 0
# Wenn Ergebnis = 100001 dann kein Wert FALSE in der Differenz, also alle groesser
sum(ll)
```

```
## [1] 100001
```

MCMC

Ausgangslage: Zusätzlich zu

$$X_i \sim Poi(\lambda)$$
$$\lambda \sim LN(\mu, 1)$$

auch noch

$$\mu \sim N(0, \tau)$$

Es handelt sich hierbei um ein hierarchisches bayesianisches Modell mit Hyperparameter μ und Hyperpriori $N(0, \tau)$
Gesucht ist nun die gemeinsame Posteriori:

$$p(\mu, \lambda|x)$$

Nach Satz von Bayes für Dichten folgt:

$$= \frac{p(x|\mu, \lambda) * p(\lambda, \mu)}{p(x)}$$

$$= \frac{p(x|\mu, \lambda) * p(\lambda|\mu) * p(\mu)}{p(x)}$$

$$\propto p(x|\mu, \lambda) * p(\lambda|\mu) * p(\mu)$$

$$\propto \lambda^{\sum_{i=1}^{n} x_i} * exp(-n\lambda) * \frac{1}{\lambda} * exp(-\frac{1}{2} * (\log \lambda - \mu)^2) * \frac{1}{\sqrt{2\pi * 1}} * exp(-\frac{\tau}{2} * (\mu - 0)^2)$$

$$\propto \lambda^{\sum_{i=1}^{n} x_i - 1} * exp(-n\lambda) * exp(-\frac{1}{2} * ((\log(\lambda)^2 * -2 * \log(\lambda) * \mu + \mu^2 + \tau * \mu^2))$$

$$\propto \lambda^{\sum_{i=1}^{n} x_i - 1} * exp(-n\lambda) * exp(-\frac{1}{2} * ((\log(\lambda)^2 * -2 * \log(\lambda) * \mu + \mu^2 * (\tau + 1))$$

Dies ist nun die gemeinsame Posteriori für λ, μ.
Die für das Gibbs-Sampling benötigten Full Conditionals erhält man, indem aus der gemeinsamen Posteriori alle multiplikativ nicht von dem zu bedingenden Parameter abhängenden Konstanten weglässt.

Full Conditional für μ:

$$p(\mu|\lambda,x) \propto \lambda^{\sum_{i=1}^n x_i - 1} * exp(-n\lambda) * exp(-\frac{1}{2} * ((\log(\lambda)^2 - 2 * \log(\lambda) * \mu + \mu^2 * (\tau+1))$$

$$\propto \exp(-\frac{1}{2} * (-2 * \log(\lambda) * \mu + \mu^2 * (\tau+1)))$$

$$\propto \exp(-\frac{1}{2} * (-2 * \log(\lambda) * \mu + \mu^2 * (\tau+1) + \frac{\log(\lambda)^2}{(\tau+1)} - \frac{\log(\lambda)^2}{(\tau+1)}))$$

$$\propto \exp(-\frac{1}{2} * (-2 * \log(\lambda) * \mu + \mu^2 * (\tau+1) + \frac{\log(\lambda)^2}{(\tau+1)}))$$

$$\propto \exp(-\frac{\tau+1}{2} * (\mu^2 - \frac{2 * \log(\lambda) * \mu}{\tau+1} + \frac{\log(\lambda)^2}{(\tau+1)^2}))$$

$$\propto \exp\left(-\frac{\tau+1}{2} * \left(\mu - \left(\frac{\log(\lambda)}{\tau+1}\right)\right)^2\right)$$

Durch quadratisches Ergänzen im Exp-Term, wodurch der negativ zu ergänzende Term wieder multiplikativ proportional ist, erhält man einen Term, der dem Kern der Normalverteilung entspricht. Die Full Conditional ist also:

$$\mu|\lambda, x \sim N\left(\frac{\log(\lambda)}{\tau+1}, \tau+1\right)$$

Aus dieser kann dann im Gibbs-Sampling ganz einfach gezogen werden, da es sich um eine bekannte Verteilung handelt.
Nun zur Full Conditional von λ, also $\lambda|\mu, x$:

$$p(\lambda|\mu,x) \propto \lambda^{\sum_{i=1}^n x_i - 1} * exp(-n\lambda) * exp(-\frac{1}{2} * ((\log(\lambda)^2 - 2 * \log(\lambda) * \mu + \mu^2 * (\tau+1))$$

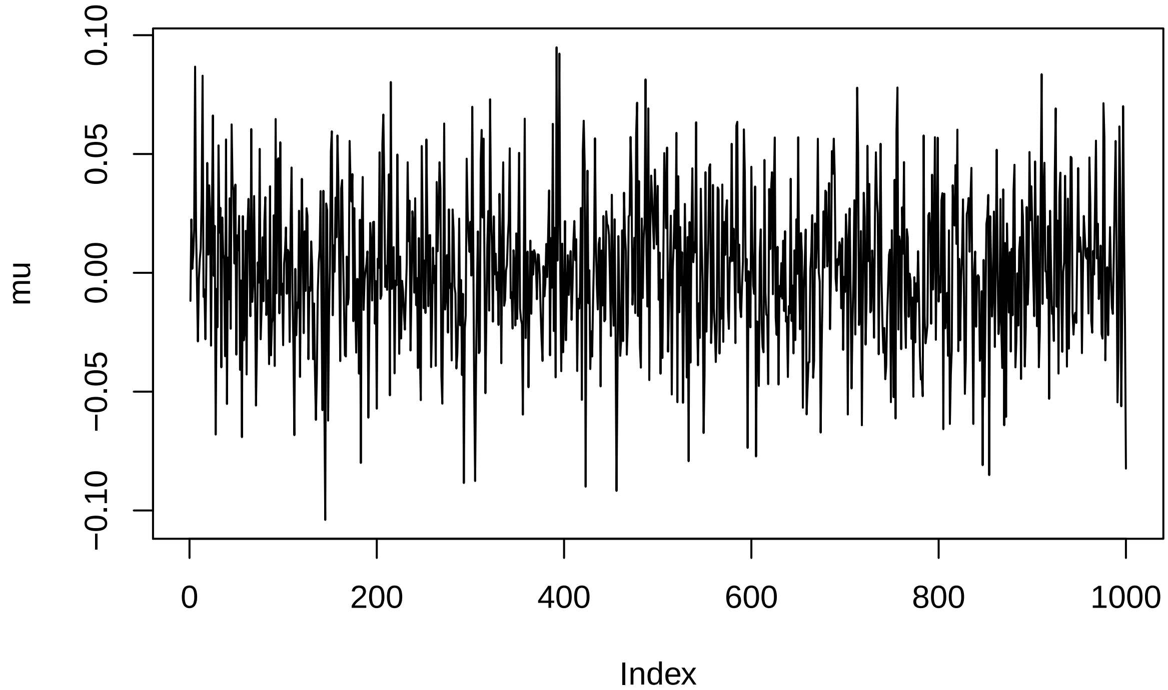
$$\propto \lambda^{\sum_{i=1}^n x_i - 1} * exp(-n\lambda) * exp(-\frac{1}{2} * ((\log(\lambda)^2 - 2 * \log(\lambda) * \mu))$$

$$\propto \lambda^{\sum_{i=1}^n x_i - 1 + \mu} * exp(-n\lambda - \frac{1}{2} * (\log\lambda)^2)$$

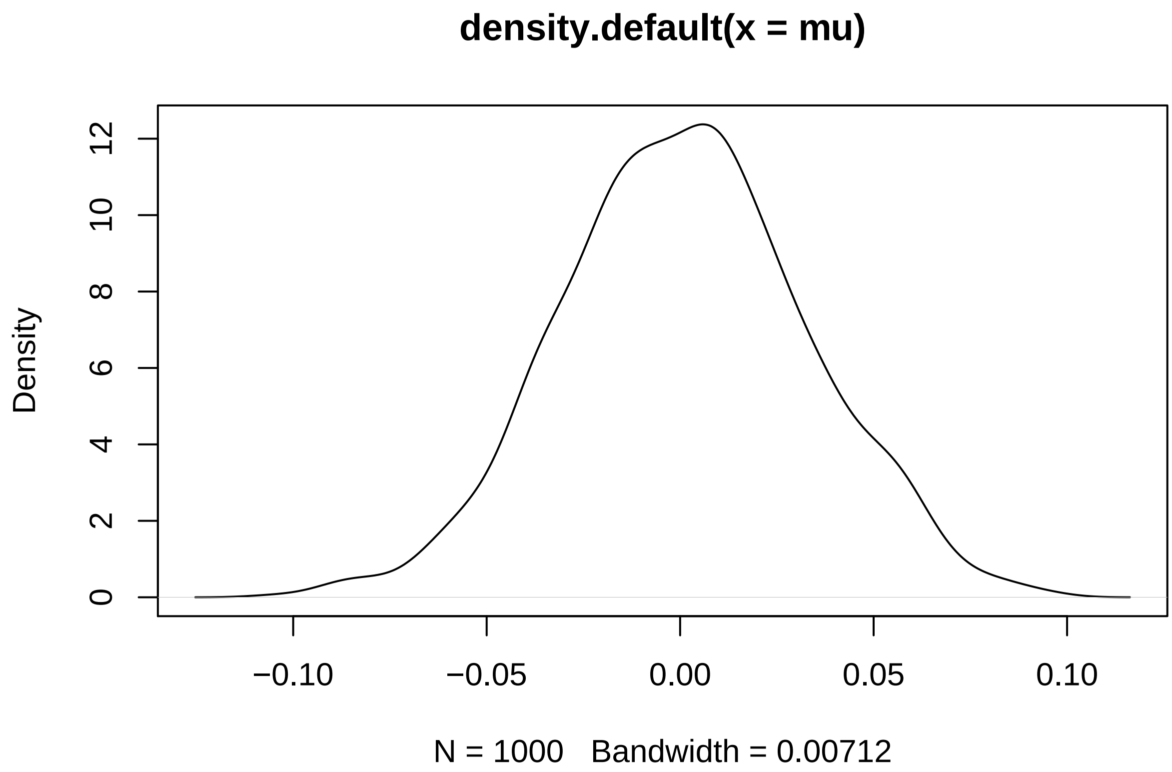
Diese Full Conditional entspricht genau der Posteriori aus Aufgabenteil b), aus der wieder mittels Acception Rejection gezogen werden kann, nur anstatt wie in Teil 'Samplen' diesmal keine 1000 ZZ sondern 1 ZZ. Außerdem ist im Unterschied mu nicht auf 0 festgesetzt, sondern ergibt sich für jede Ziehung aus der zuvor gezogenen Zufallszahl.

```
# 1000 Zufallszahlen
N <-1000
# Präzision tau = 1000 (aus Angabe)
tau <- 1000
# Stichprobenvektor, in den N lambda-Werte gespeichert werden
# (1. Wert entfernt, da selbstgewählter Startwert)
lambda<-mu <- rep(NA,N+1)
# Startwert für Lambda
```

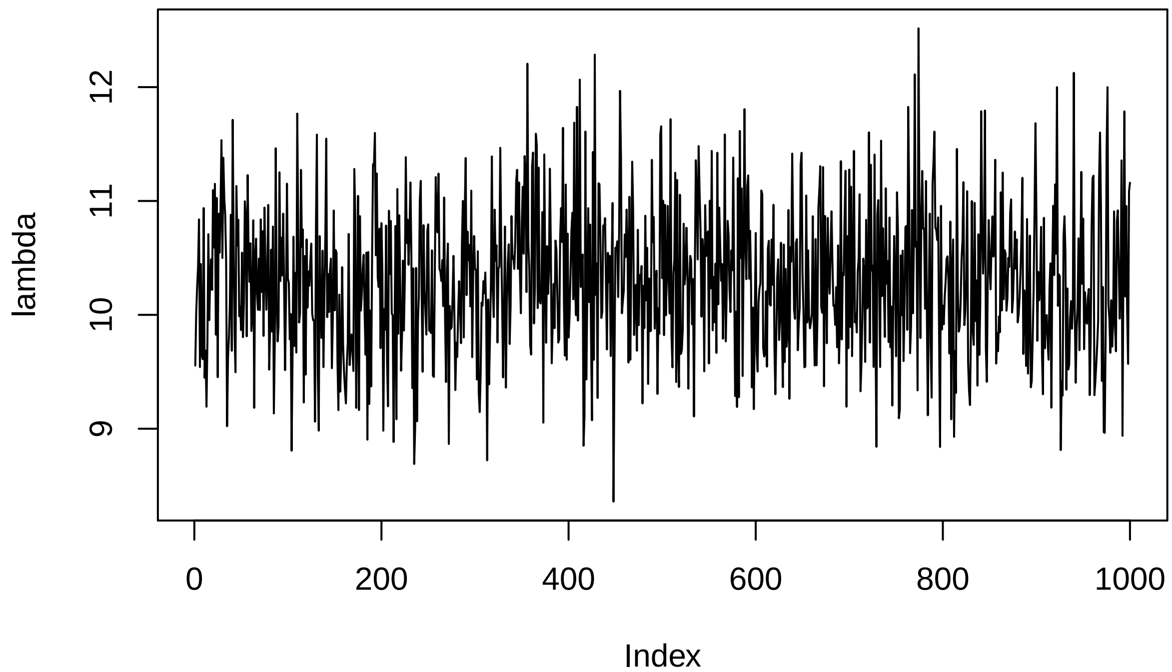
```r
lambda[1] <- 10
# Index für die Stichprobe
j <- 1
# Schleife für die Ziehung durch das Gibbs-Sampling
set.seed(1000)
while(j<(N+1))
{
  # Indexerhöhung
  j <- j+1
  # Ziehung eines Mu's aus der Full-Conditional für mu
  # (Standardabweichung ist Wurzel(1/Praezision))
  mu[j]<- rnorm(1, mean = (log(lambda[j-1])/(tau+1)), sd = sqrt((1/(tau+1))))
  # Ziehung der Lambda-Werte
  # Acception-Rejection-Ziehung wie bei (b) aus der Full Conditional für lambda
  # Nur diesmal anstatt 1000 Zufallszahlen nur Ziehen einer ZZ
  sample_2 <- c()
  N_2 <-1
  # Ziehung genau wie in Aufgabenteil 'Samplen', nur anstatt festem mu=0 nun zuvor
  # gezogener Wert
  while(N_2!=0)
  {
    prop <- rgamma(1, shape = apost, rate = bpost)
    quotient <- (postJoint(mu[j], prop, daten))/(c*(dgamma(prop, shape = apost, rate = bpost)))
    u <- runif(1, min = 0, max = 1)
    if(quotient>u){
      sample_2 <- c(sample_2, prop)
    }
    N_2 <- N_2-length(sample_2)
  }
  # Einfügen in den lambda-Vektor
  lambda[j] <- sample_2[1]
}

# Löschen des selbst gewählten Startwerts
lambda<-lambda[-1]
# NA löschen
mu<-mu[-1]

## Plots
# Plot für mu
plot(mu,type="l")
```



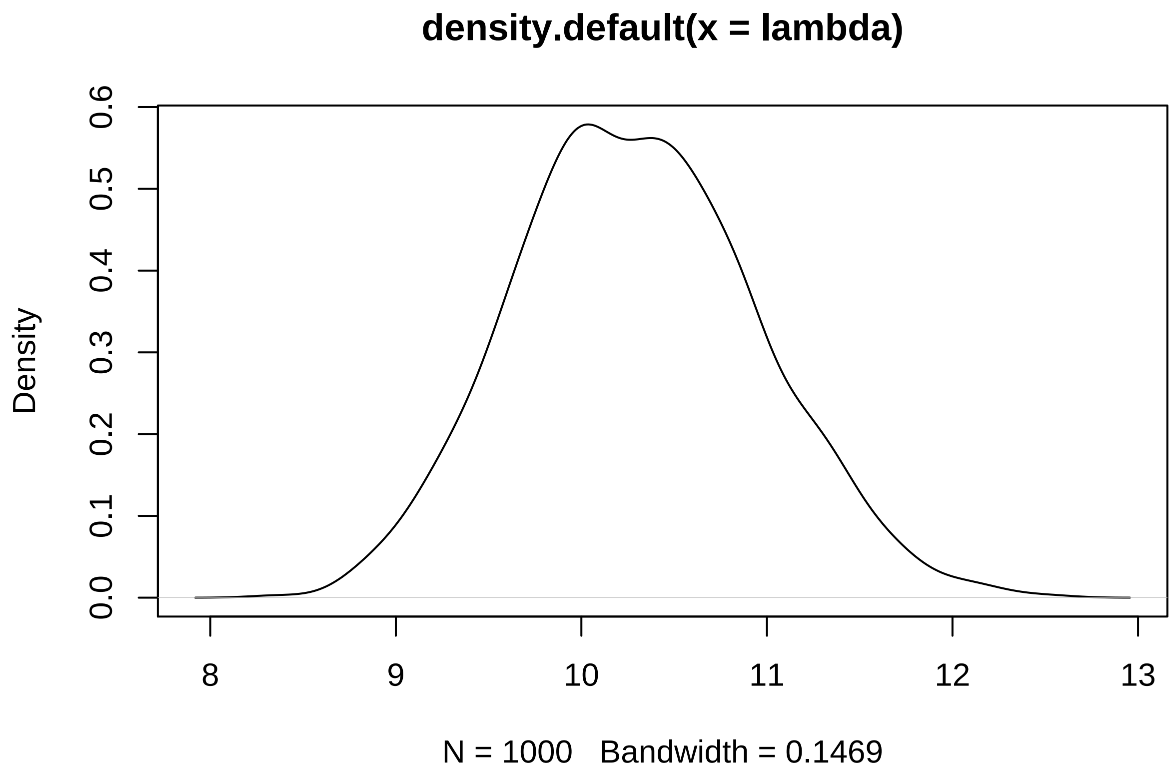

```r
# Dichtekurve für gezogene mu
plot(density(mu))
```



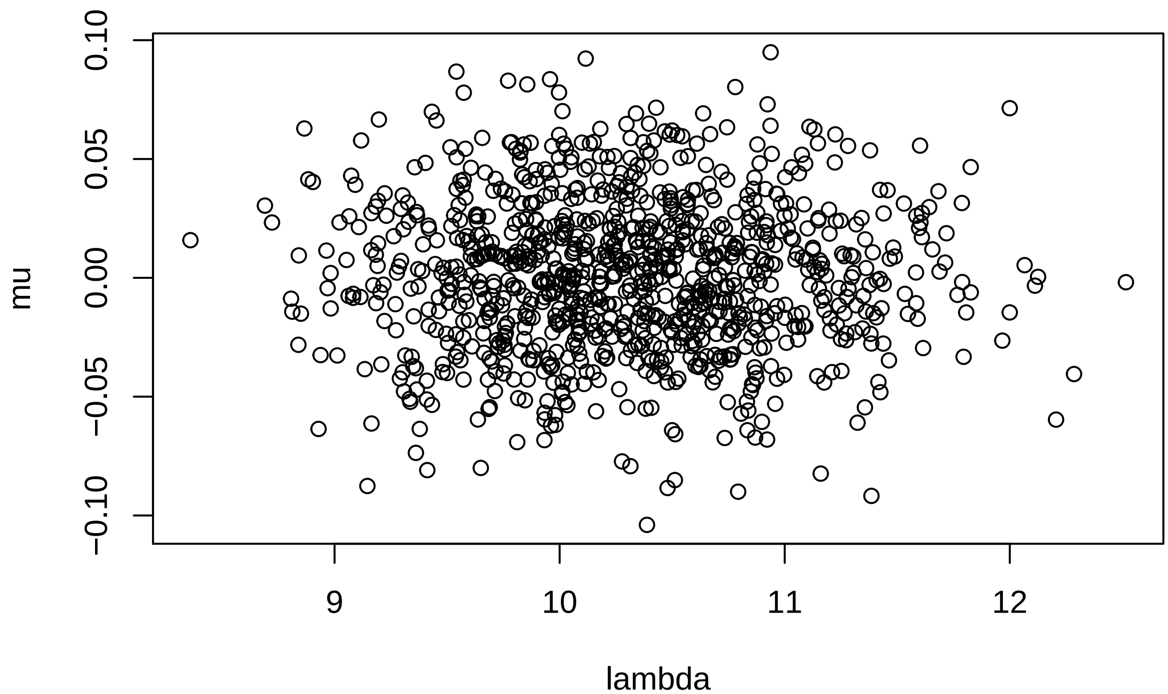

```r
# Plot für lambda
plot(lambda, type = "l")
```




```r
# Dichtekurve für gezogene lambda
plot(density(lambda))
```




```r
# Scatterplot für lambda;mu
plot(lambda, mu)
```




```
## Erwartungswerte
# Posteriori-Erwartungswert fuer mu
mean(mu)
```

```
## [1] 0.001095566
```

```
# Posteriori-Erwartungswert für lambda
mean(lambda)
```

```
## [1] 10.29771
```

```
## Varianzen
# Posteriori-Varianz für mu
var(mu)
```

```
## [1] 0.001008694
```

```
# Posteriori-Varianz für lambda
var(lambda)
```

```
## [1] 0.4221333
```

```
# Posteriori-Korrelation zwischen den Parametern
cor(mu,lambda)
```

```
## [1] -0.009861079
```

Kurze Interpretation:
Die kleine Veränderung von μ im Vergleich zur Posteriori Ziehung in Aufgabenteil 'Samplen' verändert den Posteriori Erwartungswert von λ von 10.0636 auf 10.29771. Und das obwohl der Posteriori Erwartungswert von μ im MCMC bei 0.001095566 liegt, also doch nahe dem in Aufgabenteil 2 festgelegten Wert 0.

Die Varianz der MCMC Ziehung ist auch sehr ähnlich der Ziehung aus Teil 2.
Außerdem besagt die ausgerechnete Posteriori Korrelation der beiden Parameter, dass durch die Hyperpriori von μ kein 'großer Einfluss' auf das Ziehverhalten von Lambda ausgeübt wird.