Ex8

Yelyzaveta Klysa

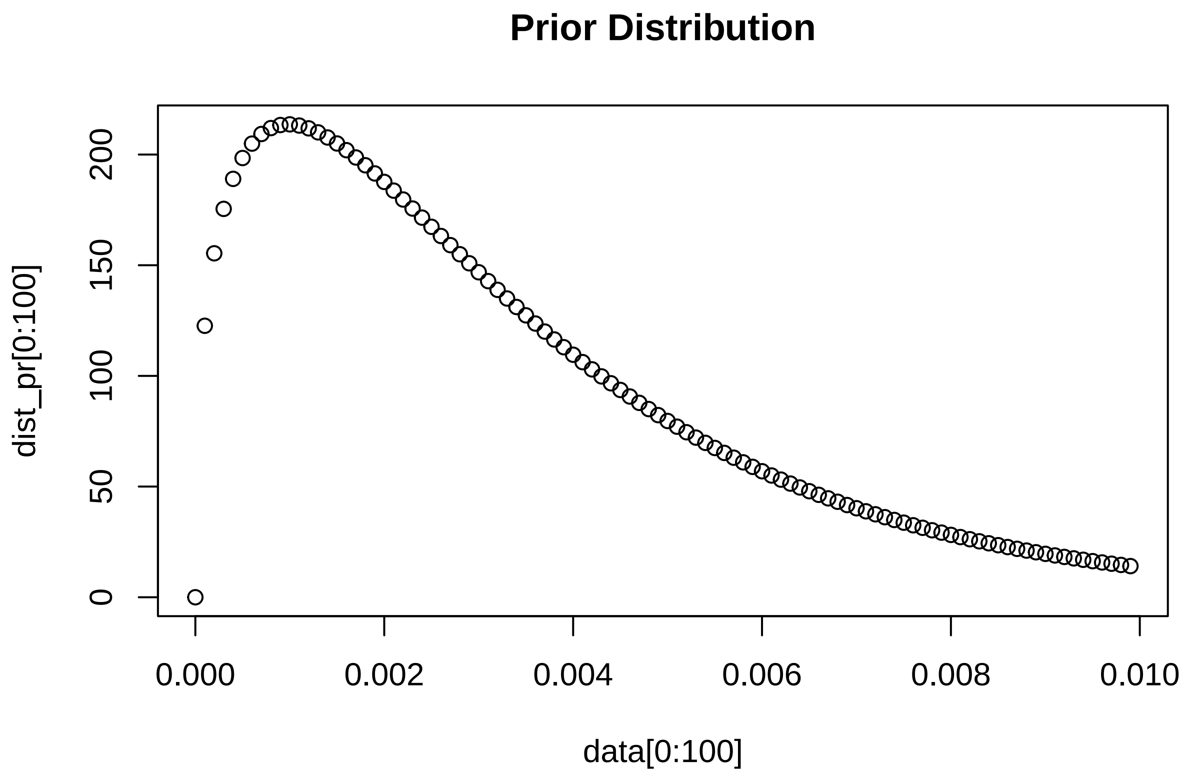
2023-10-25

Contents

Task 1

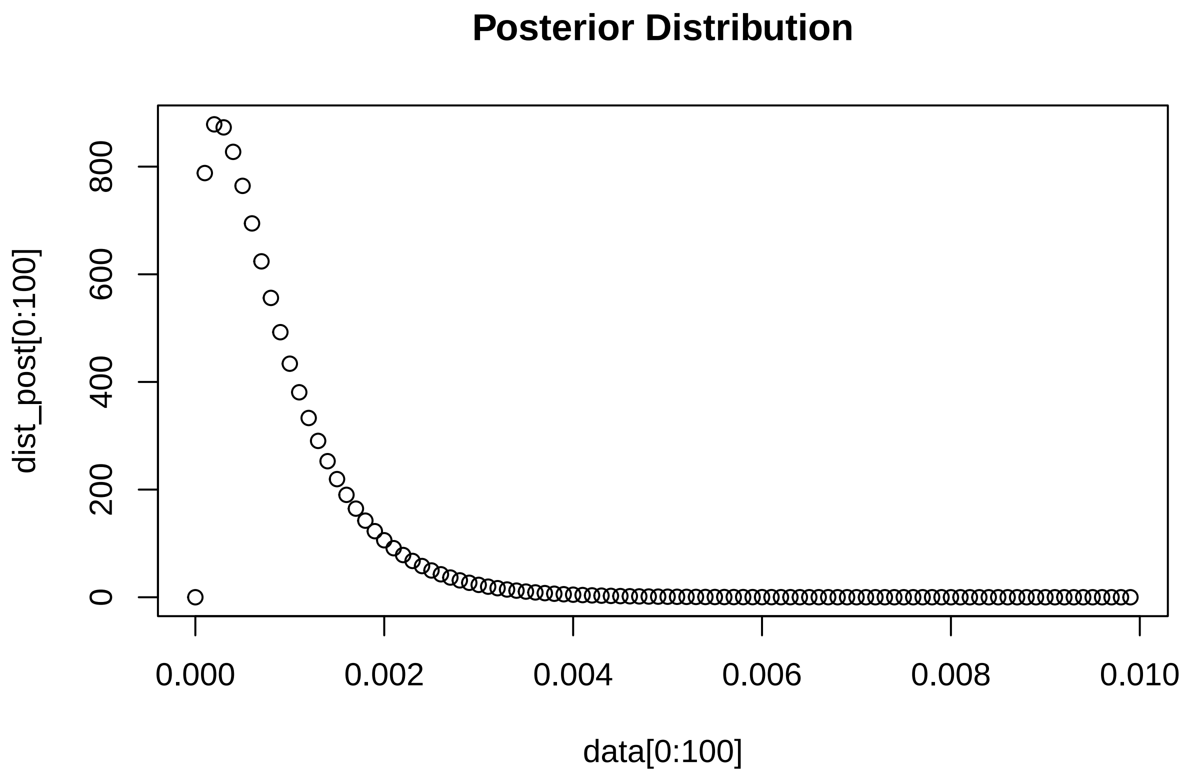
1.1 Build a Beta prior distribution for this Binomial scenario, which encodes the information of the German study.

```
set.seed(12208877)
a_pr <- 4/10 +1
b_pr <- (4068 - 4)/10 +1

data <- seq(from = 0, to = 1 , len = 10000)
dist_pr <- dbeta(data, a_pr, b_pr)
plot(data[0:100], dist_pr[0:100], main="Prior Distribution")
```



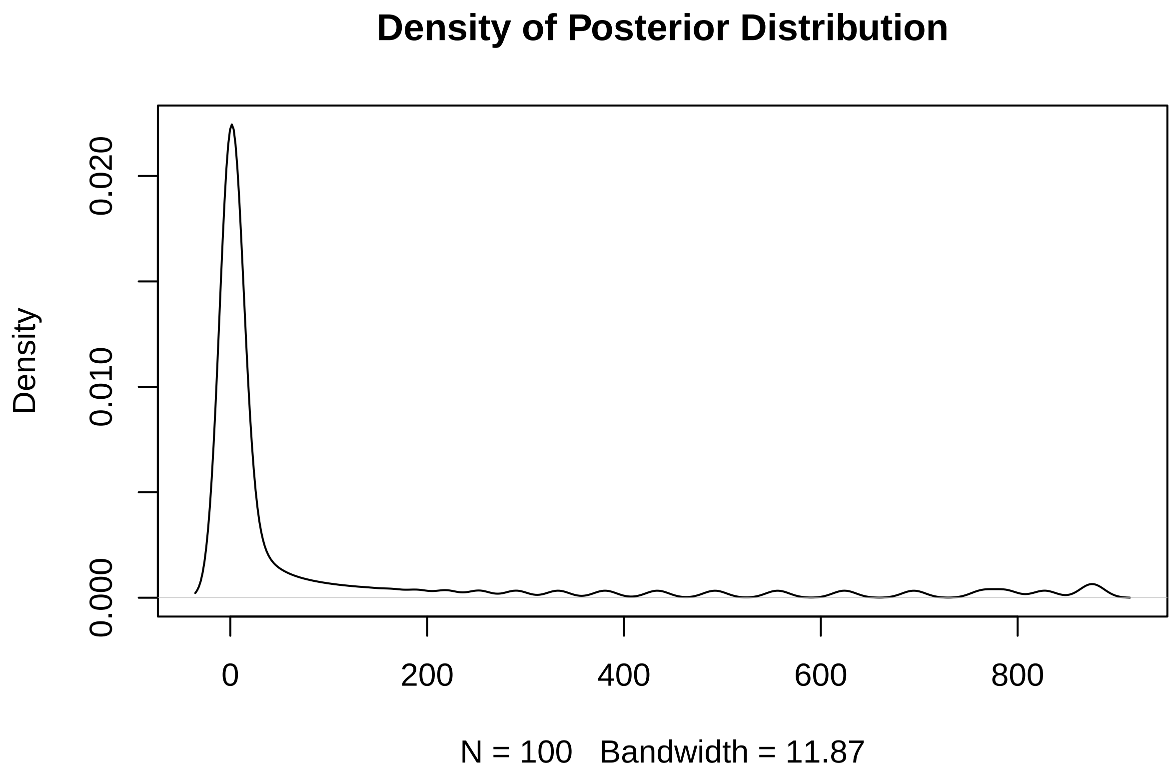
1.2 Build the corresponding Binomial model for the number of people suffering from the disease based on the 1279 test. Obtain the theoretical posterior distribution for this scenario.

```
a_post <- a_pr
b_post <- b_pr + 1279

dist_post <- dbeta(data, a_post, b_post)
plot(data[0:100], dist_post[0:100], main="Posterior Distribution")
```



1.3 Plot the posterior density and obtain the point estimators and 95% Highest posterior density interval of the prevalence of Covid19 (=proportion of inhabitants suffering from the disease).

```
plot(density(dist_post[0:100]), main="Density of Posterior Distribution")
```




```
mean_post <- a_post/(a_post + b_post)
mean_post
```

```
## [1] 0.0008294822
```

```
mode_post <- (a_post - 1)/(a_post + b_post - 2)
mode_post
```

```
## [1] 0.0002372761
```

```
median_post <- (a_post - 1/3)/(a_post + b_post - 2/3)
median_post
```

```
## [1] 0.0006322361
```

```
hpd(qbeta, shape1=a_post, shape2=b_post)
```

```
## [1] 1.972584e-07 2.210735e-03
```

1.4 Explain why Statistik Austria chose this method instead of simulationbased or frequentist inference for obtaining intervals of the prevalence.

The Bayesian methods have a few advantages that can be beneficial in this case, which might cause the choice of this method instead of others. It includes the prior information, which is good for transferable knowledge. The Bayesian methods are suited for the small sizes of samples, since the estimates from them will be more informative. Also we get the full posterior distribution instead of just estimates. Additionally, the uncertainty in case of dealing with health diseases is treated quite well.

Task 2

2.1 Define conjugate priors for the coefficient parameter and the residual variance independently. Explain how the parameters can be set to be uninformative. Compare different choice of prior parameters.

```
b_pr_i <- dnorm(0, sd = sqrt(0.1))
sigma2_pr_i <- 1 / rgamma(100, 100, 100)
```

To make the priors informative, the variance should be small, while a and b should be large and a=b. So that a prior mean of sigma2 will be 1 but the prior has a small variance.

```
b_pr_u <- dnorm(0, sd = sqrt(1000))
sigma2_pr_u <- 1 / rgamma(100, 0.5, 0.5)
```

However, if the standard deviation is a large value, the prior will be uninformative. The same way, making a and b small, will make the prior uninformative by getting a large variance.

2.2 Build the corresponding normal model the regression inference. Obtain the theoretical posterior distribution for both parameters separately assuming the other one to be "known".

Firstly, let's do it for the β assuming λ to be known: $\pi(\beta|x,y) \propto \mathcal{L}(x,y|\beta) \cdot \pi(\beta)$ where $\mathcal{L}$ is the likelihood of the data given β and λ, which is normally distributed by the definition of the task; $\pi(\beta)$ is a prior distribution for β. $\pi(\beta|x,y) \sim N(\frac{(\frac{\sum_i x_i y_i}{\sigma^2} + \frac{m}{s^2})}{(\frac{1}{s^2} + \sum_i \frac{x_i^2}{\sigma^2})}, (\frac{1}{s^2} + \sum_i \frac{x_i^2}{\sigma^2})^{-1})$

When we assume β to be known, we can do the same for λ: $\pi(\lambda|x,y) \propto \mathcal{L}(x,y|\lambda) \cdot \pi(\lambda)$ $\pi(\lambda|x,y) \sim G(\frac{n}{2} + a, \frac{\sum_i (x_i\beta - y_i)^2}{2} + b)$

2.3 Provide the formulas for point estimators and 95% Highest posterior density interval of the regression parameters separately assuming the other one to be "known".

The mean, median and mode of normal distributions coincide. Though we can not provide the formula for the median of the Gamma distribution, there can be found mode and mean.

$$E(\pi(\beta|x,y)) = median(\pi(\beta|x,y)) = mode(\pi(\beta|x,y)) = \frac{\frac{\sum_i x_i y_i}{\sigma^2} + \frac{m}{s^2}}{\frac{1}{s^2} + \sum_i \frac{x_i^2}{\sigma^2}} \; E(\pi(\lambda|x,y)) = \frac{\frac{n}{2} + a}{\frac{\sum_i (x_i\beta - y_i)^2}{2} + b}$$

Considering the fact that normal distribution is symmetric, for the 95%-HPD interval for β we can use the inverse distribution function of $N(\frac{(\frac{\sum_i x_i y_i}{\sigma^2} + \frac{m}{s^2})}{(\frac{1}{s^2} + \sum_i \frac{x_i^2}{\sigma^2})}, (\frac{1}{s^2} + \sum_i \frac{x_i^2}{\sigma^2})^{-1})$.

$HDP_{lower} = N^{-1}(0.025, mean, variance)$ $HDP_{upper} = N^{-1}(0.975, mean, variance)$

However, the Gamma distribution is not symmetric, so we can not define a formula for it here, we need to calculate it with numerical method.

2.4 Test this with the data from your exercise 6: dataset Auto and model

```
data("Auto")
set.seed(12208877)
model <- lm(mpg ~ horsepower, data = Auto)
```

```r
beta_estimate <- coef(model)["horsepower"]
beta_estimate
```

```
## horsepower
## -0.1578447
```

```r
sigma2_estimate <- var(residuals(model))
sigma2_estimate
```

```
## [1] 24.0049
```

Beta:

```r
m <- mean(Auto$mpg)
s <- 1
m_beta <- (sum(Auto$horsepower * Auto$mpg) / sigma2_estimate + m / s) /
  (1 / s + sum((Auto$horsepower)^2) / sigma2_estimate)
m_beta
```

```
## [1] 0.1789548
```

If we set `s2` smaller, the resulting beta will be closer to the one from `lm` model, which proves the fact that the increase of variance makes the priors uninformative.

```r
set.seed(12208877)
var_beta <- 1 / (1 / s + sum((Auto$horsepower)^2) / sigma2_estimate)
beta_dist <- 1 / rnorm(length(Auto$mpg), m_beta, sqrt(var_beta))
hpd_lower_beta <- quantile(beta_dist, 0.025)
hpd_upper_beta <- quantile(beta_dist, 0.975)

hpd_lower_beta
```

```
##     2.5%
## 5.455155
```

```r
hpd_upper_beta
```

```
##   97.5%
## 5.72675
```

Lambda:

```r
a <- 10000
b <- 10000

lambda_dist <- rgamma(length(Auto$mpg), (length(Auto$mpg) / 2) + a,
                      ((sum((Auto$horsepower * beta_estimate - Auto$mpg)^2) / 2) + b))

m_lambda <- ((length(Auto$mpg) / 2) + a) /
  ((sum((Auto$horsepower * beta_estimate - Auto$mpg)^2) / 2) + b)
m_lambda
```

```
## [1] 0.03115298
```

```r
sigma2 <- 1 / m_lambda
sigma2
```

```
## [1] 32.09965
```

```r
hdi_lam <- hdi(qbeta, 0.95, shape1=((length(Auto$mpg) / 2) + a),
               shape2=(sum((Auto$horsepower * beta_estimate - Auto$mpg)^2) / 2) + b)
hdi_lam
```

```
##      lower      upper
## 0.02963510 0.03079006
## attr(,"credMass")
## [1] 0.95
```

By experimental testing: the bigger the values of `a` and `b` the closer the value of `sigma2` is to the estimated one from `lm` model. Considering the fact that `a=b=0.5` corresponds to the non-informative priors, it is quite correct.

Compare the Bayesian against the frequentist results.

```r
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
bayreg <- bayesreg(mpg ~ horsepower, data = Auto)
summary(bayreg)
```

```
## ==========================================================================================
## |                   Bayesian Penalised Regression Estimation ver. 1.2                    |
## |                     (c) Enes Makalic, Daniel F Schmidt. 2016-2021                      |
## ==========================================================================================
## Bayesian Gaussian ridge regression                             Number of obs   =       392
##                                                                Number of vars  =         1
## MCMC Samples   =   1000                                        std(Error)      =    4.9154
## MCMC Burnin    =   1000                                        R-squared       =    0.6059
## MCMC Thinning  =      5                                        WAIC            =      1182
##
## -------------+-----------------------------------------------------------------------------
##    Parameter |  mean(Coef)  std(Coef)    [95% Cred. Interval]      tStat    Rank        ESS
## -------------+-----------------------------------------------------------------------------
##   horsepower |    -0.15730    0.00641     -0.17011   -0.14499    -24.533       1 **     1000
##        _cons |    39.91958    0.71821     38.51578   41.31908          .       .           .
## -------------+-----------------------------------------------------------------------------
```

The beta coefficients are very similar for the 2 models: both mean is -0.157. The R-squared is completely the same: 0.6059, while std error is a little bit better in frequentist model but also very similar. Overall, both models provide similar results.