Systems Genetics 02 - Primer in statistical modeling - Exercises

Julien Gagneur

28 October 2021

Package

BiocStyle 2.20.2

Contents

1 Maximum likelihood: Tossing coins

Consider n independent random tosses of a coin. We denote $x_i \in \{0; 1\}$ the outcome of the i-th toss (1 for head and 0 for tail) and p the probablity to get a head.

Question 1: What is the maximum likelihood estimate of p? Prove it.

Answer

The likelihood of observing given series of heads and tails can be expressed as:

$$\mathcal{L}(p; \mathbf{X}) = \prod_{i=1\dots n} p^{x_i}(1-p)^{1-x_i}$$

In order to find the maximum likelihood estimate of p we minimize the negative log likelihood with respect to p:

$$\begin{aligned}\nabla_p \, \mathrm{NLL}(p; \mathbf{X}) &= \nabla_p \left[-\log \mathcal{L}(p; \mathbf{X})\right] \\ &= -\nabla_p \left[\sum_{i=1\dots n} log(p^{x_i}) + \sum_{i=1\dots n} log((1-p)^{1-x_i})\right]\end{aligned}$$

where we used that taking the log of a product is equivalent to the sum of the logs. Using the logarithmic power rule leads to:

$$\begin{aligned}\nabla_p \, \mathrm{NLL}(p; \mathbf{X}) &= -\nabla_p \left[\sum_{i=1\dots n} x_i log(p) + \sum_{i=1\dots n} (1-x_i) log(1-p)\right] \\ &= -\nabla_p \left[log(p) \cdot \sum_{i=1\dots n} x_i + log(1-p) \cdot \sum_{i=1\dots n} (1-x_i)\right] \\ &= -\left[\frac{1}{p} \cdot \sum_{i=1\dots n} x_i - \frac{1}{1-p} \cdot \sum_{i=1\dots n} (1-x_i)\right] \\ &= -\left[\frac{1}{p} \cdot \sum_{i=1\dots n} x_i - \frac{n}{1-p} + \frac{1}{1-p} \cdot \sum_{i=1\dots n} x_i\right] \stackrel{!}{=} 0\end{aligned}$$

Hence:

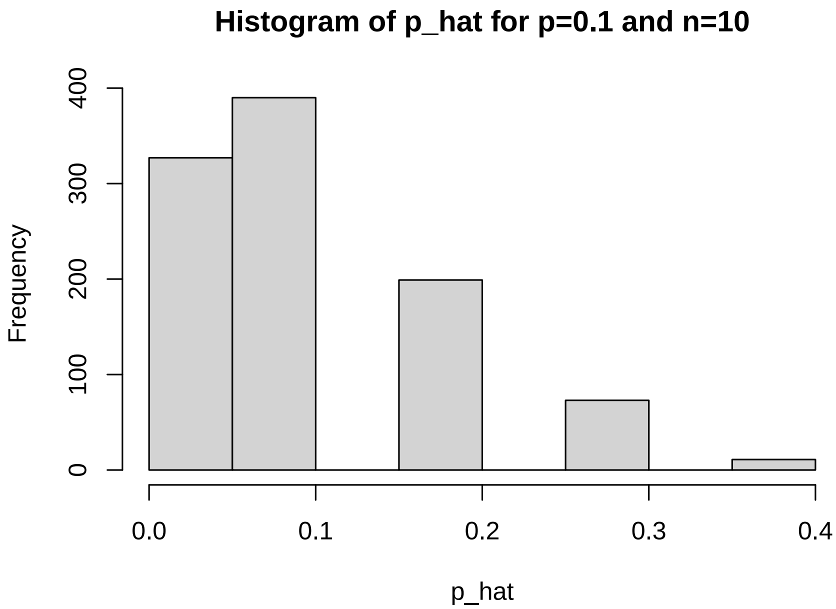
$$\begin{aligned}\frac{1-p+p}{p(1-p)} \cdot \sum_{i=1\dots n} x_i &= \frac{n}{1-p} \\ p &= \frac{1}{n} \sum_{i=1\dots n} x_i\end{aligned}$$

The R snippet `sample( c(0,1), size=n, prob=c(1-p,p), replace=TRUE)` draws `n` realizations of single tosses with probability `p` to return 1.

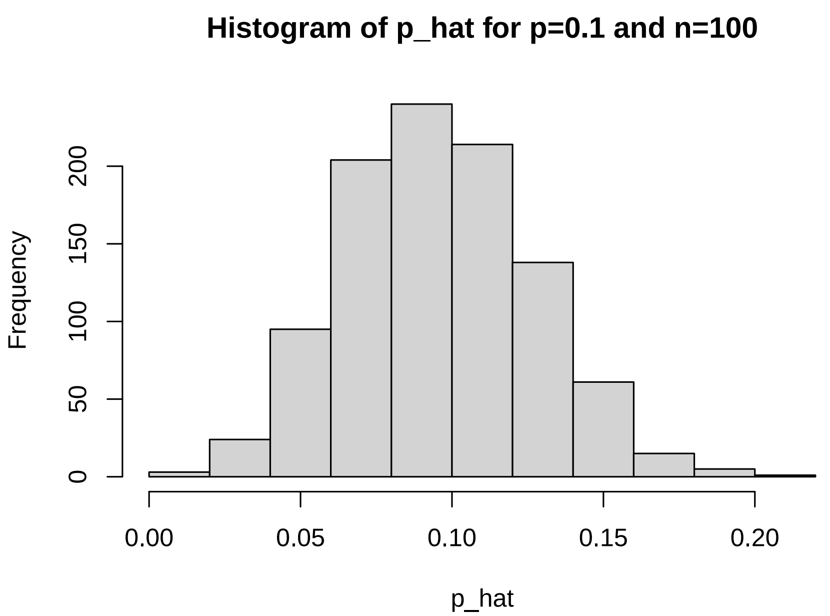
Question 2: Building on the code snippet above, implement a simulator and a max-likelihood estimator in R. Using simulations with various sample sizes n and probabilities p, investigate empirically the bias (is it on average on target?) and robustness (how far is it from the true p) of the ML estimator.

Answer

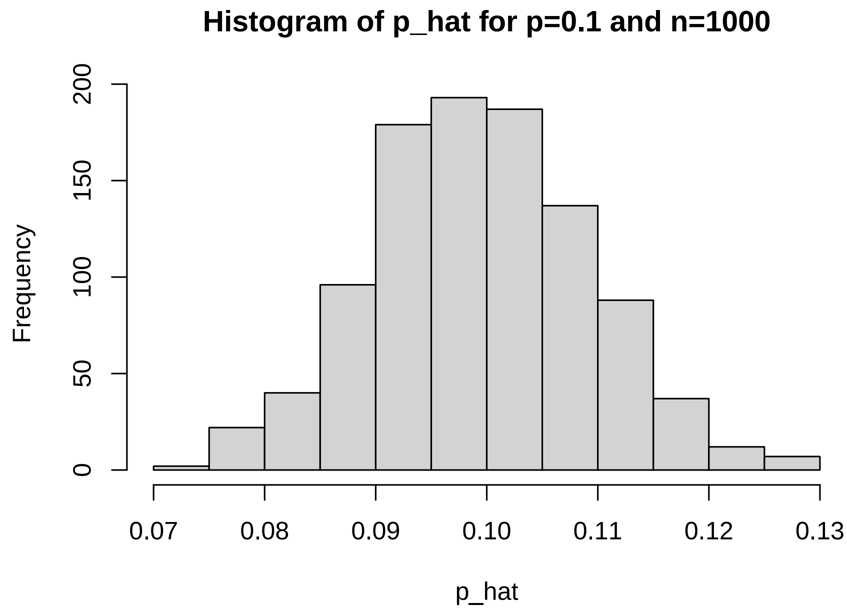
```r
# Simulate data
simulate <- function(n, p){
  x <- sample( c(0,1), size=n, prob=c(1-p,p), replace=TRUE)
  return(x)
}

# Estimate p_hat
estimate <- function(x){
    p_hat = mean(x)
  return(p_hat)
}

# Apply maximum likelihood estimation on some simulated data
n = 100
p = 0.5
x <- simulate(n, p)
p_hat <- estimate(x)
print(p_hat)
## [1] 0.5
# Investigate the bias and robustness of the ML estimator for various combinations of n,p
checkBiasAndRobustness <- function(n, p, N_sims=1000){
    p_hat <- sapply(seq_len(N_sims), n=n, p=p, function(i, n, p){
        x <- simulate(n, p)
        p_hat <- estimate(x)
        return(p_hat)
    })
    # bias: is it on average on target?
    print(paste("Mean of p_hat:", mean(p_hat)))
    # robustness: how far is it from the true p?
    print(paste("Mean absolute difference from true p:", mean(abs(p_hat - p))))
    # plot a histogram of the values of p_hat in each of the N_sims simulations
    hist(p_hat, main=paste0("Histogram of p_hat for p=", p, " and n=", n))
}
checkBiasAndRobustness(n=10, p=0.1)
## [1] "Mean of p_hat: 0.1051"
## [1] "Mean absolute difference from true p: 0.0705"
```



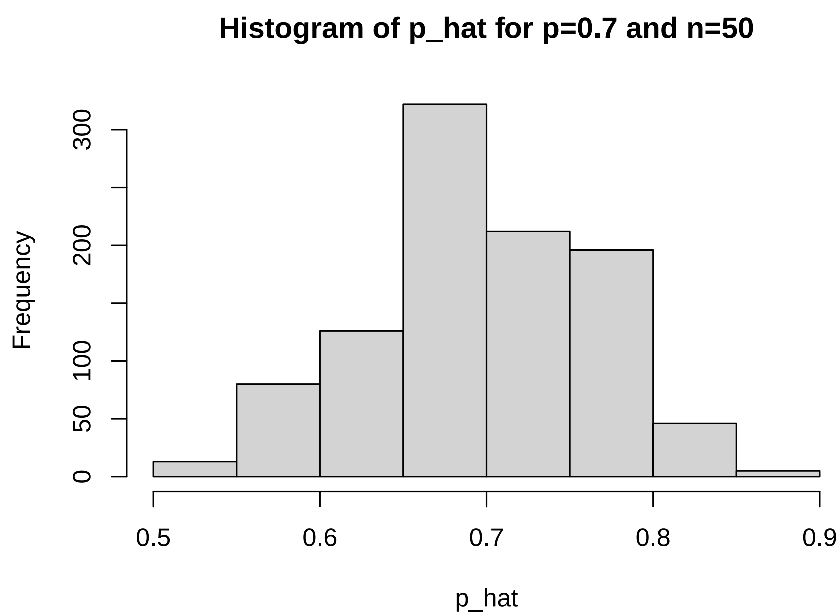

```
checkBiasAndRobustness(n=100, p=0.1)
## [1] "Mean of p_hat: 0.10071"
## [1] "Mean absolute difference from true p: 0.02447"
```



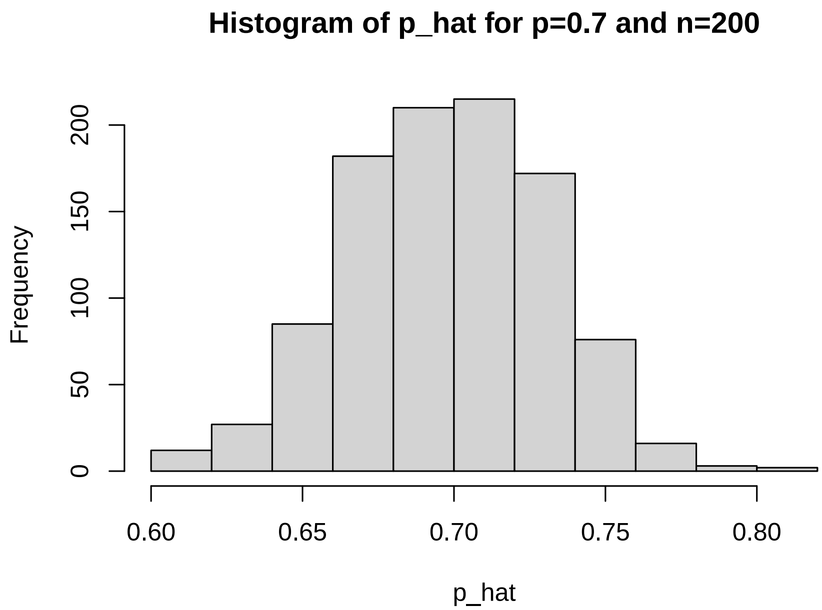

```
checkBiasAndRobustness(n=1000, p=0.1)
## [1] "Mean of p_hat: 0.099869"
## [1] "Mean absolute difference from true p: 0.007741"
```




```
checkBiasAndRobustness(n=50, p=0.7)
## [1] "Mean of p_hat: 0.70132"
## [1] "Mean absolute difference from true p: 0.0532"
```




```
checkBiasAndRobustness(n=200, p=0.7)
## [1] "Mean of p_hat: 0.70013"
## [1] "Mean absolute difference from true p: 0.02647"
```



2 Gaussian linear systems

2.1 Marginalization

Assume:

$$p(x) = \mathcal{N}(x|a, \sigma_1^2) \quad (1)$$
$$p(y|x) = \mathcal{N}(y|x + b, \sigma_2^2) \quad (2)$$

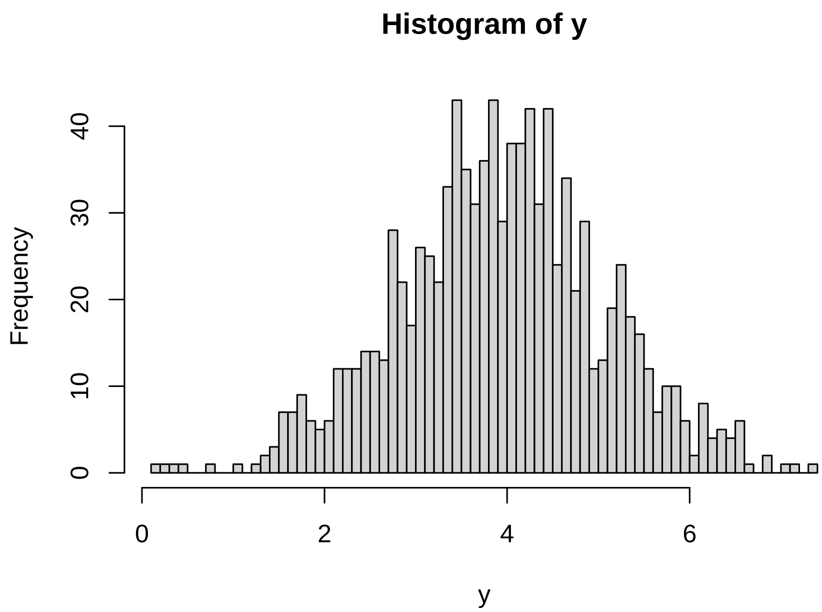
Question 3: In R, simulate 10^3 random draws of $p(y)$ according to this model for various values $a, b, \sigma_1^2, \sigma_2^2$ of your choice. Check with normal quantile plots that $p(y)$ is normal and that its mean and variance depend on $a, b, \sigma_1^2, \sigma_2^2$ as expected by the relevant result(s) from the course.

Hint: The function `rnorm()` performs random draws according to the normal distribution. The calls `qqnorm(y)` and `qqline(y)` draw Q-Q plots against the normal distribution and the line of expected quantiles.

Answer

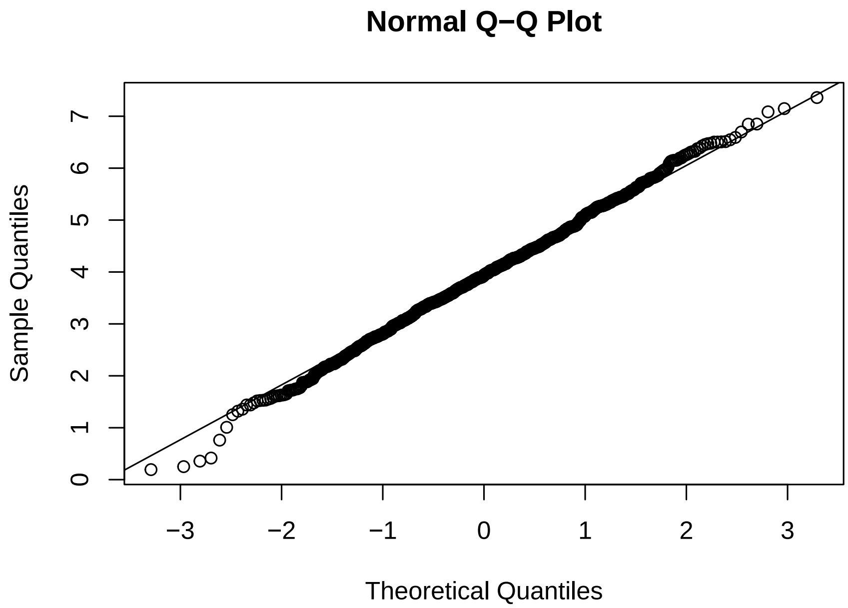
```
a <- 1
b <- 3
s1 <- 0.5
s2 <- 1
n <- 1e3
y <- rep(NA,n)
```

```
for(i in 1:n){
  x <- rnorm(1, a, s1)
    y[i] <- rnorm(1, x+b, s2)
    }
hist(y, breaks=100)
```



We show with normal quantile plots that y follows a normal distribution and its mean is about $a+b$ and variance the sum of the variance.

```
a+b
## [1] 4
mean(y)
## [1] 3.93
s1^2+s2^2
## [1] 1.25
var(y)
## [1] 1.26
qqnorm(y)
qqline(y)
```



2.2 Conditioning

Assume x and y are 1-dimensional variables such that:

$$p(x) = \mathcal{N}(x|\mu_x, \sigma_x^2) \tag{3}$$

$$p(y|x) = \mathcal{N}(y|a\,x + b, \sigma_y^2) \tag{4}$$

for $a, b \in \mathbb{R}$.

Question 4: Show that $\mu_{x|y} := \mathrm{E}[x|y]$ is a convex combination of $(y-b)/a$ and μ_x:

$$\mu_{x|y} = w\frac{y-b}{a} + (1-w)\,\mu_x \tag{5}$$

What values of the parameter a, σ_x^2, and σ_y^2 can lead to the extreme cases $w \to 0$ and $w \to 1$? Interpret.

Answer

Using results for conditional and marginal Gaussians, it follows that

$$\frac{1}{\sigma_{x|y}^2} = \frac{1}{\sigma_x^2} + \frac{a^2}{\sigma_y^2}\,.$$

Furthermore,

$$
\begin{aligned}
\mu_{x|y} &= \sigma^2_{x|y}\left[\frac{a}{\sigma^2_y}(y-b)+\frac{1}{\sigma^2_x}\mu_x\right] \\
&= \sigma^2_{x|y}\left[\frac{a^2}{\sigma^2_y}\frac{y-b}{a}+\frac{1}{\sigma^2_x}\mu_x\right] \\
&= \frac{\frac{a^2}{\sigma^2_y}\frac{y-b}{a}+\frac{1}{\sigma^2_x}\mu_x}{a^2/\sigma^2_y+1/\sigma^2_x} \\
&= w\frac{y-b}{a}+(1-w)\,\mu_x
\end{aligned}
$$

with

$$
w=\frac{a^2/\sigma^2_y}{a^2/\sigma^2_y+1/\sigma^2_x}
$$

When $w \to 0$, our guess is essentially μ_x , i.e. the information of y does not allow us to infer anything useful about x. $w \to 0$ if $a^2/\sigma^2_y \ll 1/\sigma^2_x$. All the rest being fixed, this is obtained for either $\sigma^2_x \to 0$ in which case there is too little variation in x for useful inference about y, $\sigma^2_y \to +\infty$, in which case there is too much variance on y conditioned on x, and for $a \to 0$ in which case the linear relationship is not strong enough.