Computer practical 9 and 10

Supplementary material Statistics 1 (term 2)

Professor Philip Brainard

2019-01-30

Contents

Notice:

This document contains material which is not examinable for Statistics 1.

The only purpose of this document is to provide with an Rmarkdown templete those interested in becoming familiar with Rmarkdown. This document does not aim at introducing new statistical concepts that will be examined.

Students are not required to learn, or understand the statistical concepts or methods mentioned in this document. The statistical concepts introduced in Section 2, are for the sake of presentation and only.

Rmarkdown is not examinable concept, however, students can use it, if they want in order to produce reports or present data analysis results.

---

1 First contact

[R Markdown] is a great way to create dynamic documents with embedded chunks of R code. It produces fully interactive documents that allow the readers to change the parameters underlying the analysis, and see the results immediately. It is be self contained and fully reproducible which makes it very easy to share.

Here, we provide a simple templete to start with. There are several cheat sheets briefly representing the commands and syntax of R-markdown, some of them are:

- R-markdown reference: [Click here]
- Cheat sheet about r-markdown: [Click here]
- Cheat sheet about R: [Click here]

More details can be found in [Xie et al., 2018], an online book which is available from [here]. In particular, Sections 1, 2, and 3 are quite interesting.

1.1 Practice

1.1.1 Let's practice a lillte bit.

Try executing the chunk below by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

The user to specify how the chunk will be executed, or how its output will be presented, by setting the appropriate flags inside {r, . . . }.

```
x <- rnorm(100, 0, 1)
x2_mean = mean(x^2)
hist( x )
```

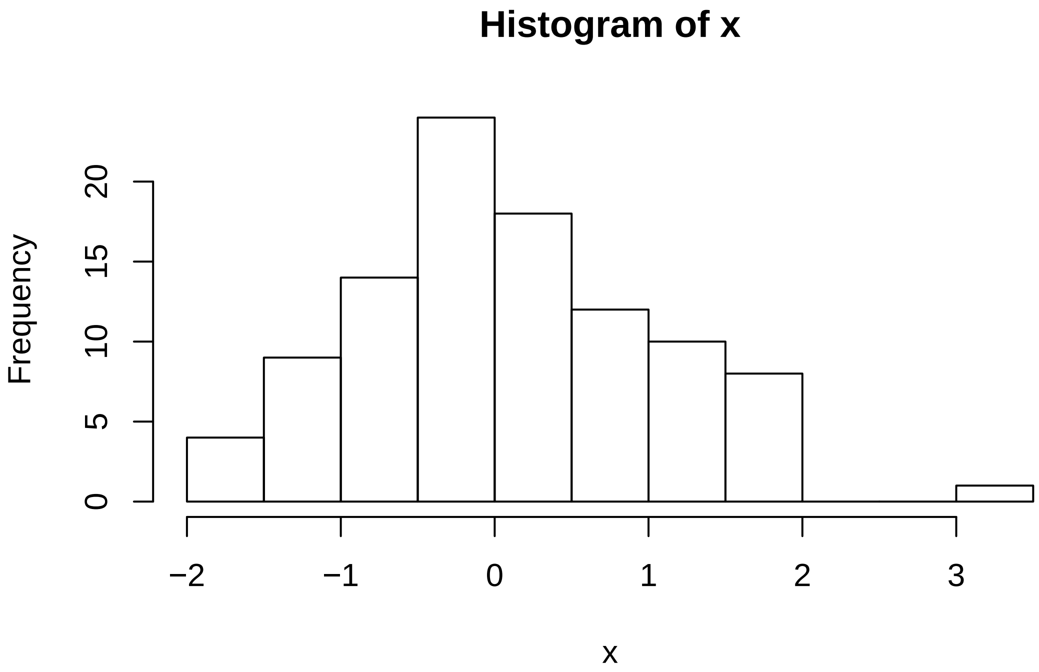


Figure 1: The histogram of values generated from a standard normal distribution

In Figure 1, we present a histogram of random values drawn from Normal distribution

Inline r code can run by typing 2, and 0.9723011 .Now see what it is print in the produced PDF.

You can create a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

1.1.2 Let's practice a lillte more.

Some times if your dataset is in the form of a `data.frame`, it can be nicely tabulated in the output document by using the function `knitr::kable()`. For instance, the following R-chunk will produce Table 1.:

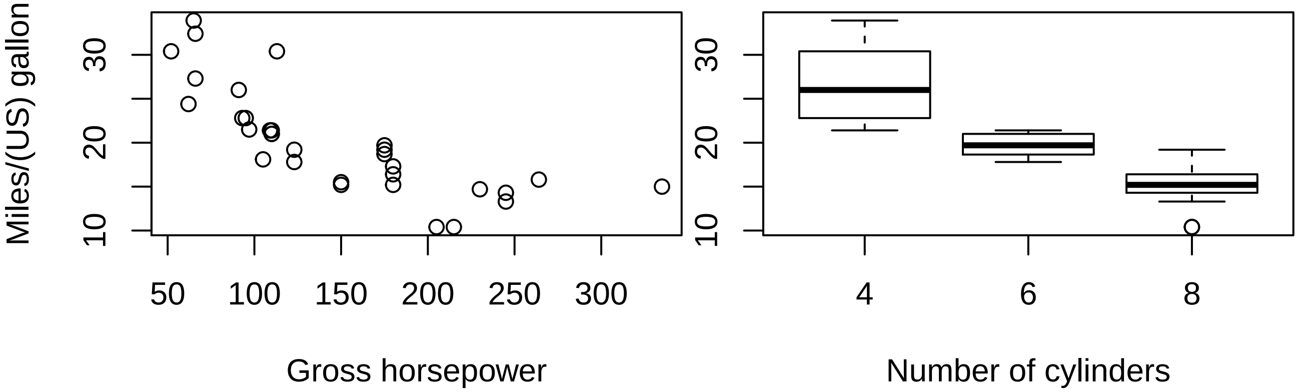
```
library(knitr)
knitr::kable( mtcars,
              caption = "\\label{tbl:mtcar}The data-set ntcars")
```

Table 1: The data-set ntcars

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Also, you can plot several sub-figures, with captions, the follwoing way. Please pay attention to the options specified in the R-chunk below. These options can produce the sub-titles of the sub-figures.

```r
plot( mtcars$hp,
      mtcars$mpg,
      xlab = "Gross horsepower",
      ylab = "Miles/(US) gallon",
      main = " ",
      type ="p")
boxplot(mtcars$mpg ~ mtcars$cyl,
        xlab="Number of cylinders",
        ylabs="Miles/(US) gallon")
```



(a) A scatter plot

(b) A box plot

Figure 2: Some plots

2 Example: Discovering cheating while maintaining privacy

Professor Philip Brainard (Medfield College) was interested in learning the rate of the students cheated in a recent exam test in chemistry. Obviously, he could not just ask his students

> Did you cheat on the test?

because a number of students might not be that honest with their responses. In fact, it would introduce a bias: the true rate is less than your observed rate, (cheaters may reply NO, and not-cheater would of course never reply YES).

To infer proportions of cheaters, while also maintaining the privacy of the population, he thought the following procedure.

Privacy Algorithm [Davidson-Pilon et al., 2015]

1. Have the user privately flip a coin. If heads, answer "Did you cheat?"truthfully.

2. If tails, flip again. If heads, answer "Yes" regardless of the truth; if tails, answer "No".

This way, Prof Brainard would not know whether a cheating confession is a result of cheating or a heads on the second coin flip. One could of course argue that the interviewers are still receiving false data since some Yes's are not confessions but instead randomness. However, this way the systematic data generation process that can be modeled. Furthermore, they do not have to incorporate (perhaps somewhat naively) the possibility of deceitful answers.

2.1 The experiment

The experiment is performed as follows:

> In the interview process for each student, the student flips a coin, hidden from the interviewer. The student agrees to answer honestly if the coin comes up heads. Otherwise, if the coin comes up tails, the student (secretly) flips the coin again, and answers "Yes, I did cheat" if the coin flip lands heads, and "No, I did not cheat", if the coin flip lands tails. This way, the interviewer does not know if a "Yes" was the result of a guilty plea, or a Heads on a second coin toss. Thus privacy is preserved and the researchers receive honest answers.

Table 2 presents the data collected. Our sample size is $n = 100$, and we have $x = 35$ positive responses.

Table 2: The responses of the students collected by using the privacy algorithm.

Positive	Negative	Total
35	65	100

2.2 Sampling distribution

Let Y_i be the response of the i-th student with

$$Y = \begin{cases} 1 & \text{, positive} \\ 0 & \text{, negative} \end{cases}$$

let C_1 be the output of the first coin with

$$C_1 = \begin{cases} 1 & \text{, Head} \\ 0 & \text{, Tail} \end{cases}$$

and let C_2 be the output of the second coin with

$$C_2 = \begin{cases} 1 & \text{, Head} \\ 0 & \text{, Tail} \end{cases}$$

Let $p \in [0, 1]$ be the proportion of cheaters in the exam. Then the probability that a student gives a positive response is

$$\begin{aligned} P(Y = 1) =& P(C_1 = 1)p + P(C_1 = 0)P(C_2 = 1) && (1) \\ =& \frac{1}{2}p + \frac{1}{2}\frac{1}{2} && (2) \\ =& \frac{1}{2}p + \frac{1}{4} && (3) \end{aligned}$$

and the probability that a student gives a negative response

$$\begin{aligned} P(Y = 0) =& 1 - P(Y = 1) && (4) \\ =& \frac{3}{4} - \frac{1}{2}p && (5) \end{aligned}$$

Therefore for the sample $\{Y_1, ..., Y_n\}$, the sample distribution is

$$Y_i \sim \text{Bernoulli}(\frac{1}{2}p + \frac{1}{4}), \quad \text{for } i = 1, ..., n$$

and hence

$$\begin{aligned} P(Y_{1:n}|p) =& \prod_{i=1}^{n}(\frac{1}{2}p + \frac{1}{4})^{y_i}(\frac{3}{4} - \frac{1}{2}p)^{1-y_i} && (6) \\ =& (\frac{1}{2}p + \frac{1}{4})^{x}(\frac{3}{4} - \frac{1}{2}p)^{n-x} && (7) \\ & && (8) \end{aligned}$$

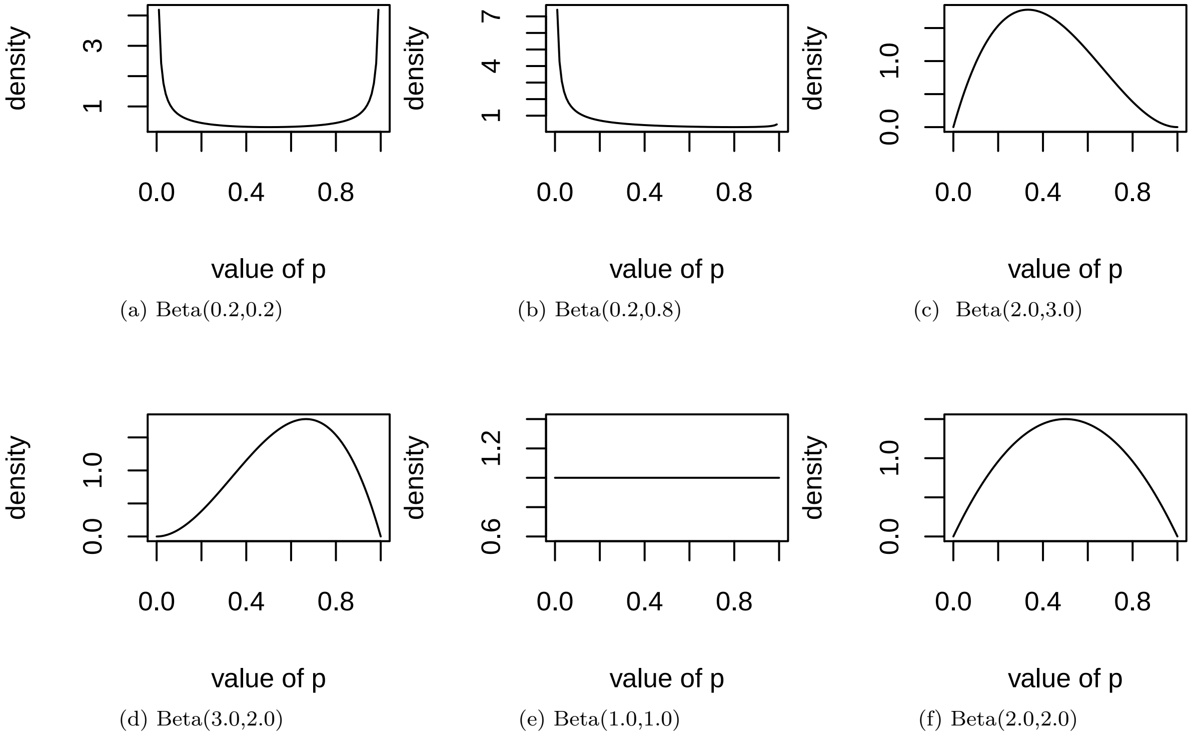
where $X = \sum_{i=1}^{n} Y_i$.

To learn the proportion of cheaters $p \in [0, 1]$, one can use the Bayes theorem, and invert the probability 6. For thease reason, we need to specify a prior probability distribution for p.

We assign a Beta prior distribution on p, $p \sim \text{Beta}(a, b)$, with density

$$f(p|a, b) = \frac{1}{B(a, b)} \int_0^1 p^{1-a}(1 - p)^{1-b} dp$$

Because Prof Brainard had no prior knowledge whether his students tend to cheat or not (they were 1st year students) he chooses prior hyper-parameters a and b which give prior whose density/concentration is uniformly distributed in the interval $[0, 1]$. Figure 5 presents the shapes of the prior densities for different combinations of a and b. Hence, a priori ignotance can be expressed with $(a, b) = (1, 1)$.



(a) Beta(0.2,0.2)

(b) Beta(0.2,0.8)

(c) Beta(2.0,3.0)

(d) Beta(3.0,2.0)

(e) Beta(1.0,1.0)

(f) Beta(2.0,2.0)

Figure 3: Beta prior distribution density functions for p, denoted as Beta(a,b)

According to the Bayes theorem, the posterior probability distribution of p given the observations has density

$$\begin{aligned} f(p|Y_{1:n}, a, b) =& \frac{\prod_{i=1}^{n} P(Y_i = y_i|p) f(p|a, b)}{\int_0^1 \prod_{i=1}^{n} P(Y_i = y_i|p) f(p|a, b) dp} \\ =& \frac{(\frac{1}{2}p + \frac{1}{4})^x (\frac{3}{4} - \frac{1}{2}p)^{n-x} p^{a-1} (1-p)^{b-1}}{\int_0^1 (\frac{1}{2}p + \frac{1}{4})^x (\frac{3}{4} - \frac{1}{2}p)^{n-x} p^{a-1} (1-p)^{b-1} dp} \end{aligned}$$

The density of the posterior distribution of p is presented in Figure 5.

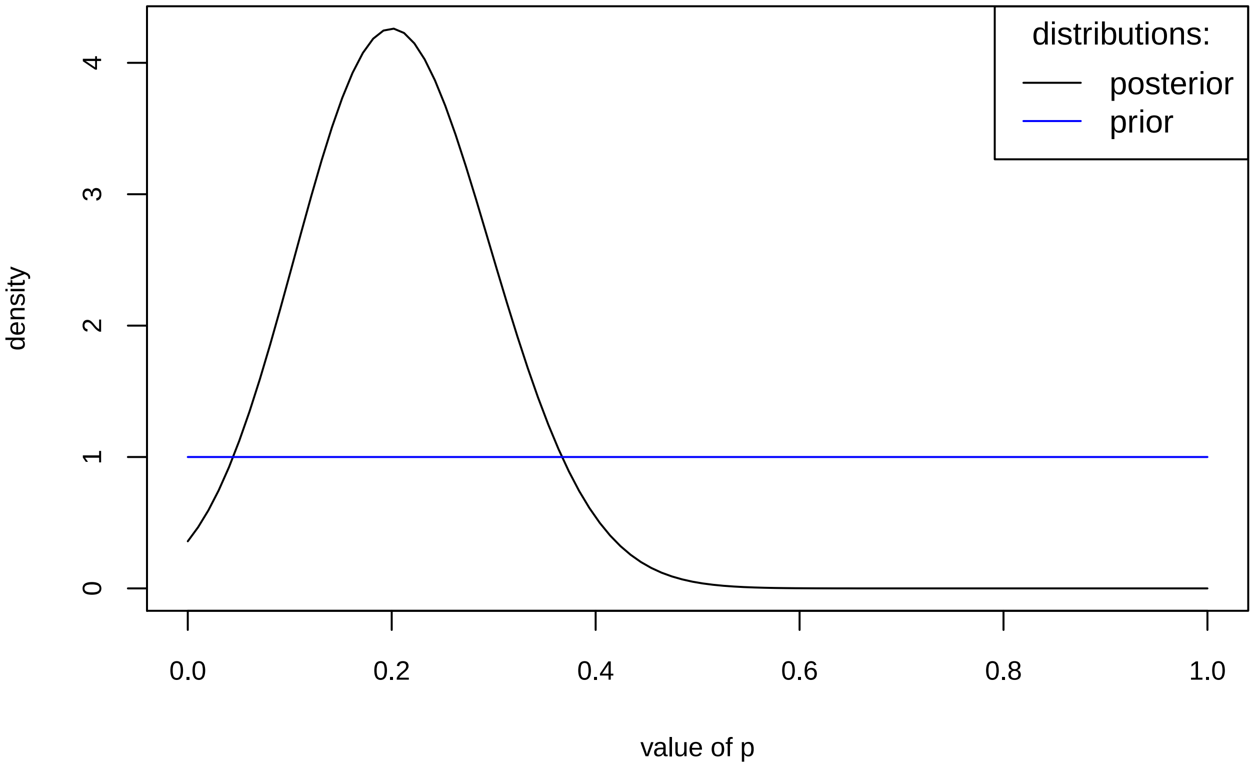


Figure 4: Density functions for p

The posterior expected value of the probability that a student cheat is

$$E(p|Y_{1:n}, a, b) = \int_0^1 p f(p|Y_{1:n}, a, b) dp$$

with standard error

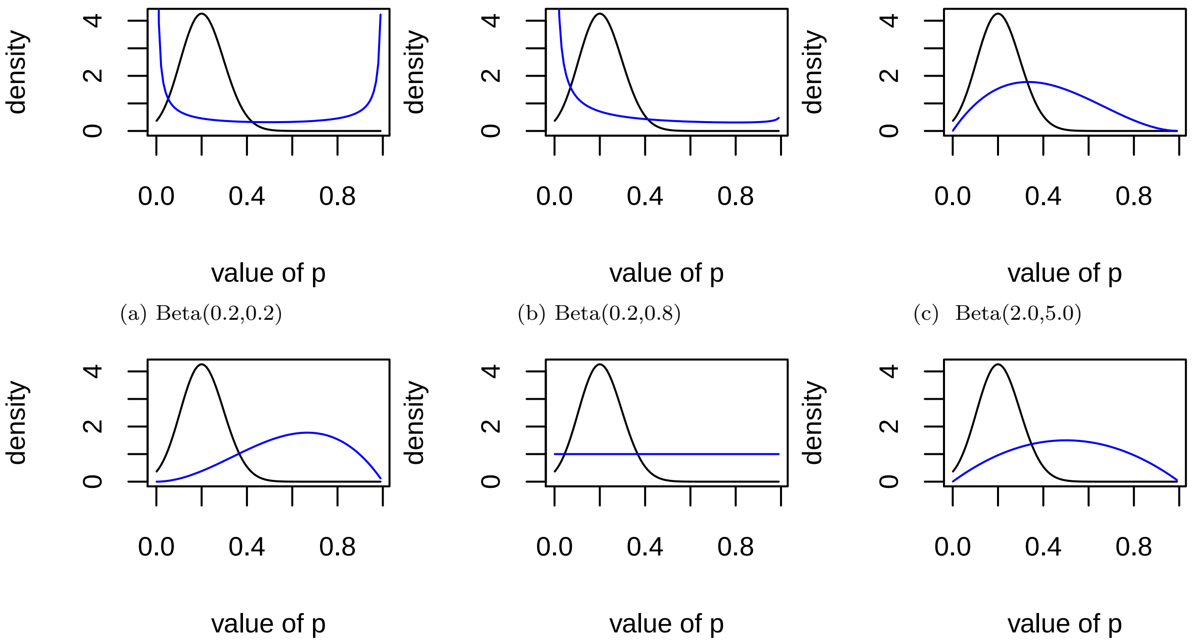
$$SD(p|Y_{1:n}, a, b) = \sqrt{E(p^2|Y_{1:n}, a, b) - E(p|Y_{1:n}, a, b)^2}$$

with $E(p^2|Y_{1:n}, a, b) = \int_0^1 p^2 f(p|Y_{1:n}, a, b) dp$.

The posterior expected rate of cheater is $E(p|Y_{1:n}, a, b) = 0.208594$, with standard error $SD(p|Y_{1:n}, a, b) = 0.0912515$.

2.3 Sensitivity analysis

We are interested in how sensitive to different values of the hyper-parameters a, and b, the posterior distribution is. In Figure 5, we present posterior distribution densities associated to priors.



(a) Beta(0.2,0.2)

(b) Beta(0.2,0.8)

(c) Beta(2.0,5.0)

(d) Beta(5.0,2.0)

(e) Beta(1.0,1.0)

(f) Beta(2.0,2.0)

Figure 5: POsterior and prior distributions density functions for p, with different prior hyper-parameter values. The prior density is colored in blue, and the posterior density is colored in black.

In Table 3, we present the posterior expected value and the standard deviation of p, for different values of hyper-parameters a, and b. We observe that changing the hyper-parameters of the priors can significantly effects the posterior inference.

Table 3: The poseterior expected value, and standard deviation of p.

a	b	$E(p;a,b)$	$SD(p;a,b)$
0.2	0.2	0.1636051	0.1019968
0.2	0.8	0.1562692	0.10063
2	3	0.3390941	0
3	2	0.8861159	0
1	1	0.208594	0.0912515
2	2	0.4059721	0

References

Cameron Davidson-Pilon et al. Probabilistic programming & bayesian methods for hackers, 2015.

Yihui Xie, JJ Allaire, and Garrett Grolemund. *R Markdown: The Definitive Guide*. CRC Press, 2018.