

Course name  
Topic of the Lecture

**Your name**



**ISEG Lisbon School of Economics & Management**

April 27, 2020

# Contents



- ① Introduction
  - ▶ Before we start
  - ▶ Environments
- ② Some formulas
  - ▶ Multinomial - Dirichlet Model
  - ▶ Binomial - Beta Model
- ③ Code chunks
  - ▶ Python code
  - ▶ R code
- ④ References



# About this lecture

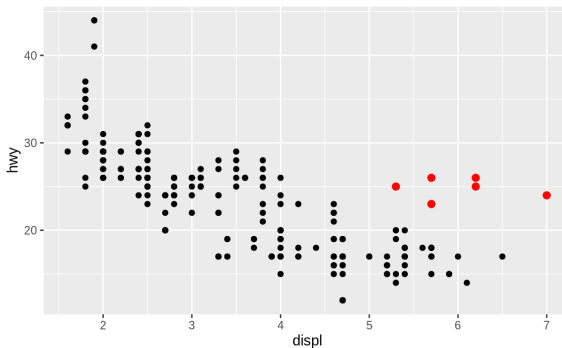
In this lecture you will see:

- a graph
- some formulas
- some code
- some customized environments
- some citations (Kass and Raftery, 1995)
- a link to [ISEG's](#) website



# A scatterplot

This picture is in the “figures” folder:



**Figure 1:** Fuel consumption (Hwy) vs engine size (displ)



# Three ways to make citations

- Kass and Raftery, 1995
- (Kass and Raftery, 1995)
- Kass and Raftery (1995)



# Definitions and theorems

I customized the definition theorem and block environments:

## Definition (Fibration)

A fibration is a mapping between two topological spaces that has the homotopy lifting property for every space  $X$ .

## Theorem (Bayes)

$$P(\theta|\mathbf{D}) = P(\theta) \frac{P(\mathbf{D}|\theta)}{P(\mathbf{D})}$$



# The block environment

The block environment is more versatile. You can write whatever you want on the red block:

## Pythagorean theorem

This is a theorem about right triangles and can be summarised in the next equation

$$x^2 + y^2 = z^2$$

## Corollary

There's no right rectangle whose sides measure 3cm, 4cm, and 6cm.



# Two part functions

I also included an environment that allows you to easily write two part functions:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$





# Bayesian stuff

- Likelihood:  $\mathbf{x}|\boldsymbol{\theta} \sim \mathcal{M}_k(N, \boldsymbol{\theta})$
- Hypotheses:  $H_0 : \boldsymbol{\theta} = \boldsymbol{\theta}_0$  vs  $H_1 : \boldsymbol{\theta} \neq \boldsymbol{\theta}_0$
- Prior:  $\pi(\boldsymbol{\theta}|H_0) = 1_{\boldsymbol{\theta}_0}(\boldsymbol{\theta}_0)$  and  $\boldsymbol{\theta}|H_1, \boldsymbol{\alpha} \sim \text{Dir}_k(\boldsymbol{\alpha})$
- Marginal density:  $m_i(\mathbf{x}) = \int_{\Theta_i} f(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|H_i) d\boldsymbol{\theta}$ ,  $i = 0, 1$

$$\blacktriangleright m_0(\mathbf{x}) = \frac{N!}{\prod_{i=1}^{k+1} x_i!} \prod_{i=1}^{k+1} \theta_{0i}^{x_i}$$

$$\blacktriangleright m_1(\mathbf{x}) = \frac{N!}{\prod_{i=1}^{k+1} x_i!} \frac{B(\boldsymbol{\alpha} + \mathbf{x})}{B(\boldsymbol{\alpha})}$$

- Bayes factor:

$$B_{01}(\mathbf{x}) = \frac{m_0(\mathbf{x})}{m_1(\mathbf{x})} = \frac{\prod_{i=1}^{k+1} (\theta_{0i}^{x_i}) \prod_{i=1}^{k+1} [\Gamma(\alpha_i)] \Gamma[\sum_{i=1}^{k+1} (\alpha_i + x_i)]}{\Gamma(\sum_{i=1}^{k+1} \alpha_i) \prod_{i=1}^{k+1} \Gamma(\alpha_i + x_i)}$$

- For an interesting application see Pericchi and Torres (2011).

# Bayesian stuff



- Prior:  $\theta|\alpha \sim \text{Dir}_k(\alpha)$
- Posterior:  $\theta|\alpha, \mathbf{x} \sim \text{Dir}_k(\alpha + \mathbf{x}) \rightarrow$  Mean, mode, credible intervals.
- For an interesting application see Ley (1996).



## More Bayesian stuff

- Likelihood:  $x \sim \text{Bin}(N, \theta)$
- Hypotheses:  $H_0 : \theta = \theta_0$  vs  $H_1 : \theta \neq \theta_0$
- Prior:  $\pi(\theta|H_0) = 1_{\theta_0}(\theta)$  and  $\theta|H_1 \sim \text{Beta}(a,b)$
- Marginal density:  $m_i(x) = \int_0^1 f(x|\theta)\pi(\theta|H_i)d\theta, i = 0, 1$ 
  - ▶  $m_0(x) = \binom{N}{x} \theta_0^x (1 - \theta_0)^{N-x}$
  - ▶  $m_1(x) = \binom{N}{x} \frac{B(x+a, n-x+b)}{B(a,b)}$
- Bayes factor:  $B_{01}(x) = \frac{m_0(x)}{m_1(x)} = \frac{\theta_0(1-\theta_0)^{N-x} \Gamma(a) \Gamma(b) \Gamma(n+a+b)}{\Gamma(a+b) \Gamma(n+a-x) \Gamma(x+a)}$



# More Bayesian stuff

- Prior:  $\theta \sim \text{Beta}(a, b)$
- Posterior:  $\theta|x \sim \text{Beta}(a + x, b + N - x) \rightarrow$  Mean, mode, credible intervals



## Python code looks good on verbatim

This code prints the Fibonacci sequence:

```
nterms = int(input("How many terms? "))
n1, n2 = 0, 1
count = 0
print("Fibonacci sequence:")
while count < nterms:
    print(n1)
    nth = n1 + n2
    n1 = n2
    n2 = nth
    count += 1
}
```



## Some R code

I also included a verbatim environment with background color:

```
getmode <- function(v) {  
  uniqv <- unique(v)  
  uniqv[which.max(tabulate(match(v, uniqv)))]  
  
}
```

With this R code you can build a function that calculates the mode.

# References



- Kass, RE and AE Raftery (1995). Bayes factors. *Journal of the American Statistical Association* **90**(430), 773–795.
- Ley, E (1996). On the peculiar distribution of the US stock indexes' digits. *The American Statistician* **50**(4), 311–313.
- Pericchi, L and D Torres (2011). Quick anomaly detection by the Newcomb—Benford law, with applications to electoral processes data from the USA, Puerto Rico and Venezuela. *Statistical Science* **26**(4), 502–516.