

**‘From Statistics to Data Mining’
Computer Lab Session n° 3:
Probability, Random Variables and Probability Distributions**

**Master 1 COSI / CPS²
Saint-Étienne, France**

Fabrice Muhlenbach

Laboratoire Hubert Curien, UMR CNRS 5516
Université Jean Monnet de Saint-Étienne
18 rue du Professeur Benoît Luras
42000 SAINT-ÉTIENNE, FRANCE
<http://perso.univ-st-etienne.fr/muhlfabr/>

Outcome



The objective of this lab is to become familiar with  functions for working with probabilities, random variables and probability distribution.

1 Probabilities

Relative Frequency Approach to Probability

Heads or tails?

The probability of an event E (possibly unknown), denoted by $P(E)$, is defined to be the value approached by the relative frequency of occurrence of E in a very long series of trials of a chance experiment.

With , we will represent the relative frequency of heads as a function of the number of tosses. In , the `runif` function will generate random values (it requires the `stats` library), e.g. `runif(1)` will produce a value between 0 and 1. We imagine that the coin is not loaded. They are 2 events: heads or tails, each having the same chance = $\frac{1}{2}$.

First, print 10 random values 0 or 1 (0 for the event ‘tails’, 1 for the event ‘heads’) by using the `runif` and `round` functions (the last one provides the rounded value of a decimal number).


```
require( stats )
for ( i in 1:10 )
{
  x <- round( runif( 1 ) )
  print( x )
}
```

Second, represent the relative frequency of the number of heads as a function of the number of tosses (for 1 to 1000 tosses). For this, we will use a vector called *proba* and we will initialize a *sum* value to 0. The *proba* value will be assigned by the relative frequency of the number of heads computed by the number of heads (given by *sum*) divided by the number of tosses.

Then plot this relative frequency as a graph with lines and add with the `segments` function a red segment at the 0.5 level of the relative frequency of heads.

```
proba <- as.vector(1:1000)
sum <- 0
for (i in 1:1000)
{
  x <- round(runif(1))
  sum <- sum + x
  proba[i] <- sum / i
}

plot(1:1000, proba[1:1000], "l", xlim=c(1,1000), ylim=c(0,1))
segments(0,0.5,1000,0.5, col="red")
```

You can compare your result with your neighbors and/or run the random assignment again for having another result. Modify your  code for having the representation of the relative frequency of the number of heads as a function of the number of tosses for 1 to 10,000 tosses (instead of 1,000 tosses). What is the difference between the new result and the previous one?

2 Random Variables and Probability Distributions

2.1 Probability Distribution for Discrete Random Variables

One Die

Represent the probability histogram of the results obtained by a die (not loaded).

The value $max = 6$ is the maximal number of points obtained by one die. The total number of events is equal to 6 (the 6 sides of the die). The probability for having the value 1, 2, 3... or 6 is the same, equals to $\frac{1}{6}$.

```
max <- 6
nb_events <- max
proba <- as.vector(1:max)
for (i in 1:max)
{
  proba[i] <- 1/nb_events
}
```

We can now represent the bar graph of the probability of having each value (from 1 to 6) of a die. These values are considered as a factor. We will use the `qplot` function from the `ggplot2` package.

```
require(ggplot2)
```

```
qplot(factor(1:max), proba[1:max],
      xlab="Value_obtained_by_one_die",
      ylab="Probability") + geom_bar(stat="identity")
```

Two Dice

Represent the probability histogram of the results obtained by the sum of two dice (not loaded).
With two dice, how many possibilities do we have for obtaining the value:

- 1?
- 2?
- 3? ...

What is the total number of events?

What is the maximal score obtained with two dice?

```
max_dice <- 6
score_max <- max_dice * 2
nb_events <- max_dice ^ 2

proba <- rep.int(0, score_max)
proba <- as.vector(proba)

for (i in 1:max_dice)
{
  for (j in 1:max_dice)
  {
    proba[i+j] <- proba[i+j] + 1/nb_events
  }
}

qplot(factor(1:score_max), proba[1:score_max],
      xlab="Value_obtained_with_the_sum_of_two_dice",
      ylab="Probability") + geom_bar(stat="identity")
```

2.1.1 Expected Value of a Discrete Random Variable

The expected value (mean value) of a discrete random variable X , denoted by $E(X)$, is defined as follows:

$$E(X) = \sum_{\text{all possible } x \text{ values}} x \times P(x)$$

Compute the expected value of the sum of two dice (not loaded).

```
expec <- 0
for (i in 1:score_max)
{
```

```

    expec <- expec + proba[i] * i
  }
  print(expec)

```

What is the expected value? Is it consistent with the probability histogram plotted before?

2.1.2 Variance and Standard Deviation of a Discrete Variable

The variance of a discrete random variable X , denoted by σ_X^2 or $V(X)$, is defined as follows:

$$\sigma_X^2 = V(X) = \sum_{\text{all possible } x \text{ values}} [x - E(X)]^2 \times P(x)$$

The standard deviation of X is $\sigma_X = \sqrt{\sigma_X^2}$.

Compute the variance and the standard deviation of the sum of two dice (not loaded).

```

variance <- 0
for (i in 1:score_max)
{
  variance <- variance + (expec - i)^2 * proba[i]
}
print(variance)
st_deviation <- sqrt(variance)
print(st_deviation)

```

There is another way to compute the variance: $V(X) = E(X^2) - E^2(X)$

```

expec_square <- 0
for (i in 1:score_max)
{
  expec_square <- expec_square + proba[i] * i^2
}
variance_bis <- expec_square - (expec^2)
print(variance_bis)

```


Most of the time, the denominator $n - 1$ is used which gives an unbiased estimator of the (co)variance for i.i.d. observations.

```

u_variance <- variance * (nb_events / (nb_events - 1))

```

2.1.3 R Statistical Functions

 is language specially designed for statistics and data analysis. We can work with a vector *sum* storing the different values of the sum of two dice and we can apply directly the statistical functions **mean**, **var** and **sd** for having respectively the expected value (the mean), the variance and the standard deviation. By the way, we can plot the histogram with the **hist** function. Note that the variance and standard deviation are computed as unbiased estimators.

```

sum <- rep.int(0,nb_events)
sum <- as.vector(sum)
event <- 0

for (i in 1:max_dice)
{
  for (j in 1:max_dice)
  {
    event <- event + 1
    sum[event] <- i+j
  }
}

hist(sum, breaks = c(1:12), col = "blue1")
mean(sum)
var(sum)
sd(sum)

```

2.1.4 Correlation Coefficient

Import the dataset `revision_grade.csv` (in RStudio, Workspace window, “Import Dataset” > “From Text File...”) and print the values of the data frame. In this dataset, X and Y are two discrete random variables where X is the number of hours of revision the day before an exam in Data Analysis and Y is the grade got by the student (where F-FX is encoded by 0, E by 1, ..., A by 5).

Print the cross tabulation of this dataset with the function `table`. The function `table` uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels.

Compute the correlation coefficient between X and Y with the function `cor`.


```

table(revision_grade)
cor(revision_grade)

```

2.2 Probability Distribution for Continuous Random Variables

Transition: Correlation Coefficient for Continuous Random Variables

During the previous lab session, we have seen how we can compare the different attributes of a dataset with a matrix of scatterplots (subsection 3.7 “Pairwise Graph”). This can be done in  with the `pairs` function.

Load the *iris* dataset (the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris) and print the summary of the data.

```

data("iris")
summary(iris)

```

Then print the pairwise graph of the graph with the 4 first continuous attributes, and the correlation coefficient between all attributes but the 5th:

```
pairs(iris[-5], bg=iris$Species, pch=21)
cor(iris[-5])
```

Load the package **PerformanceAnalytics**. If this package is not yet installed, install it before on your personal library (on your personal drive 'U:')

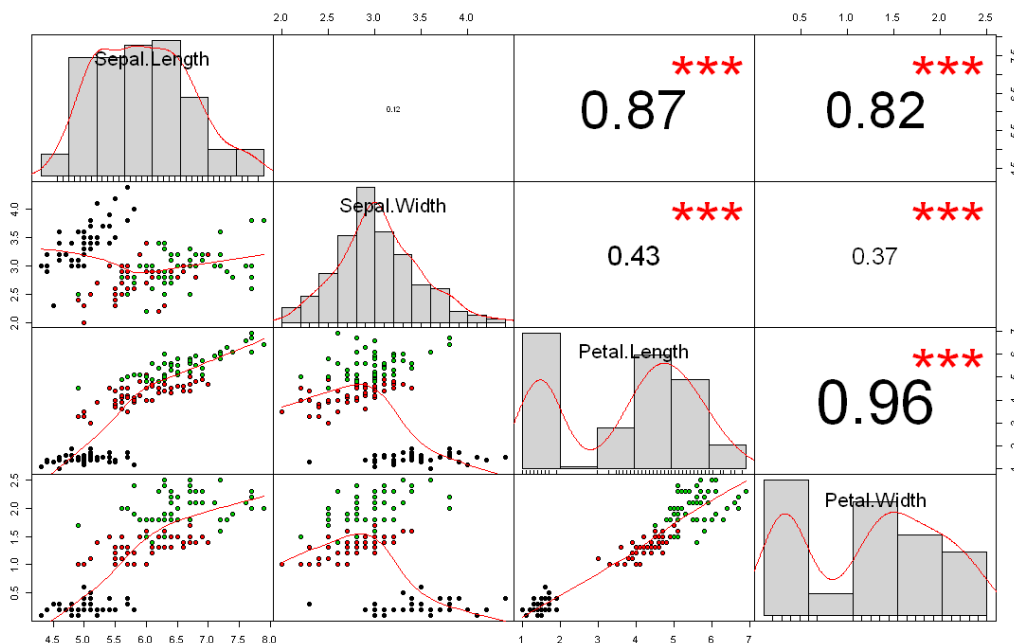
```
install.packages("PerformanceAnalytics",
  lib="U:/My_R_personal_library")
library(PerformanceAnalytics)
```

Print the char correlation of the iris dataset:

```
chart.Correlation(iris[-5], bg=iris$Species, pch=21)
```

This chart contains a lot of information:

- On the diagonal are the univariate distributions, plotted as histograms and kernel density plots.
- On the right of the diagonal are the pair-wise correlations, with red stars signifying significance levels.
- As the correlations get bigger the font size of the coefficient gets bigger.
- On the left side of the diagonal is the scatter-plot matrix, with loess smoothers in red to help illustrate the underlying relationship.



This plot combines a large amount of information into one command and one easy to follow plot.

2.3 Binomial and Geometric Distributions

2.3.1 Binomial Distribution $\mathcal{B}(n, p)$

The *binomial random variable* X is defined as X = number of successes observed among n trials. Then, if X is a binomial variable (noted $X \equiv \mathcal{B}(n, p)$), we get:

$$P(X = x) = p(x) = \binom{n}{x} p^x (1 - p)^{n-x} = \frac{n!}{x!(n-x)!} p^x (1 - p)^{n-x}$$

where $\binom{n}{x}$ is read as “ n choose x ” and represents the number of ways of choosing x items from a set of n . Note that for $n = 1$, the binomial variable is equivalent to the Bernoulli variable $\mathcal{B}(1, p)$.

$$E(X) = np$$

$$V(X) = np(1 - p)$$

In **R**, the relevant functions for a binomially distributed random variable X for n trials and with success probability p are:

- `dbinom(x, n, p)`, to find $P(X = x)$
- `pbinom(x, n, p)`, to find $P(X \leq x)$
- `qbinom(q, n, p)`, to find c such that $P(X \leq c) = q$
- `rbinom(n, x, p)`, to generate n independent values of X

Here, the size n and the probability p are the binomial parameters $\mathcal{B}(n, p)$, while x denotes the number of “successes.” The output from this function is the value of $P(X = x)$.

Example: Compute the probability of getting four heads in six tosses of a fair coin.

```
dbinom(x = 4, size = 6, prob = 0.5)
```

We will obtain the value 0.234375. Thus, $P(X = 4) = 0.234$, when X is a binomial random variable with $n = 6$ and $p = 0.5$.

Cumulative probabilities of the form $P(X \leq x)$ can be computed using `pbinom()`; this function takes the same arguments as `dbinom()`. For example, we can calculate $P(X \leq 4)$ where X is the number of heads obtained in six tosses of a fair coin as:

```
pbinom(4, 6, 0.5)
```

Plot the probability mass function of the binomial distributions for $p = 1/2$ and $n = 5, 10, 15$ and 20, then the probability mass function of the binomial distributions for $n = 10$ and $p = 1/2, p = 1/3, p = 1/4, p = 1/5$.

```
colors<-c("black", "blue", "red", "green")
n_max <- 20
n <- 5
p <- 1/2
```

```

fd <- function(x) {dbinom(x,n,p)}

plot(cbind(0:n, sapply(0:n,fd)),
     xlim=c(0,n_max), ylim=c(0,.40),
     type="p", ylab="", xlab="",
     pch=15, cex=2, col=colors[1], cex.axis=2)

n <- 10
fd <- function(x) {dbinom(x,n,p)}
points(cbind(0:n, sapply(0:n,fd)),
       xlim=c(0,n_max), ylim=c(0,.40),
       type="p", ylab="", xlab="",
       pch=16, cex=2, col=colors[2], cex.axis=2)

n <- 15
fd <- function(x) {dbinom(x,n,p)}
points(cbind(0:n, sapply(0:n,fd)),
       xlim=c(0,n_max), ylim=c(0,.40),
       type="p", ylab="", xlab="",
       pch=17, cex=2, col=colors[3], cex.axis=2)

n <- 20
fd <- function(x) {dbinom(x,n,p)}
points(cbind(0:n, sapply(0:n,fd)),
       xlim=c(0,n_max), ylim=c(0,.40),
       type="p", ylab="", xlab="",
       pch=18, cex=2, col=colors[4], cex.axis=2)

mtext(c(expression(paste(italic(n), "=5"))),
      adj=0, at=1, col=colors[1])

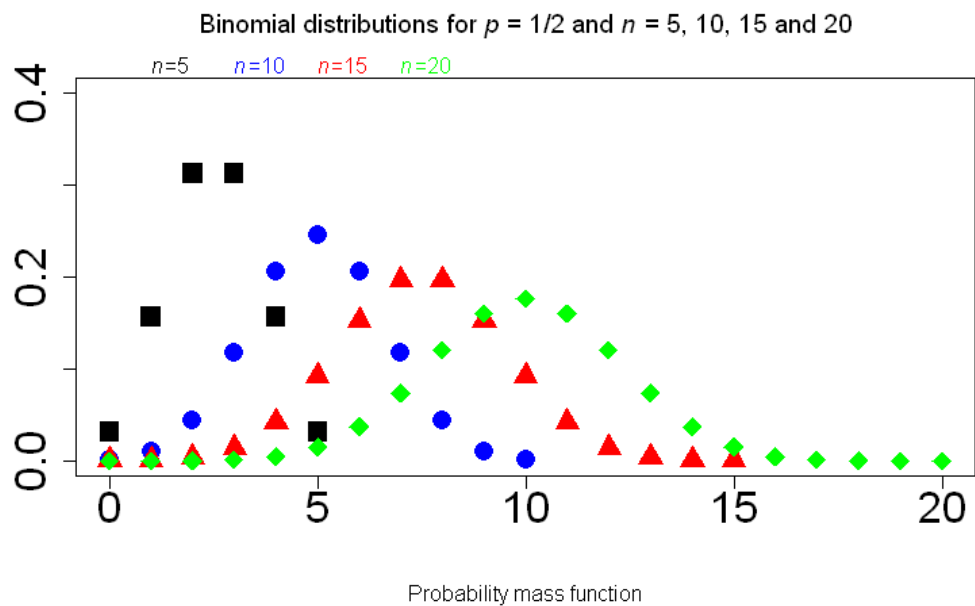
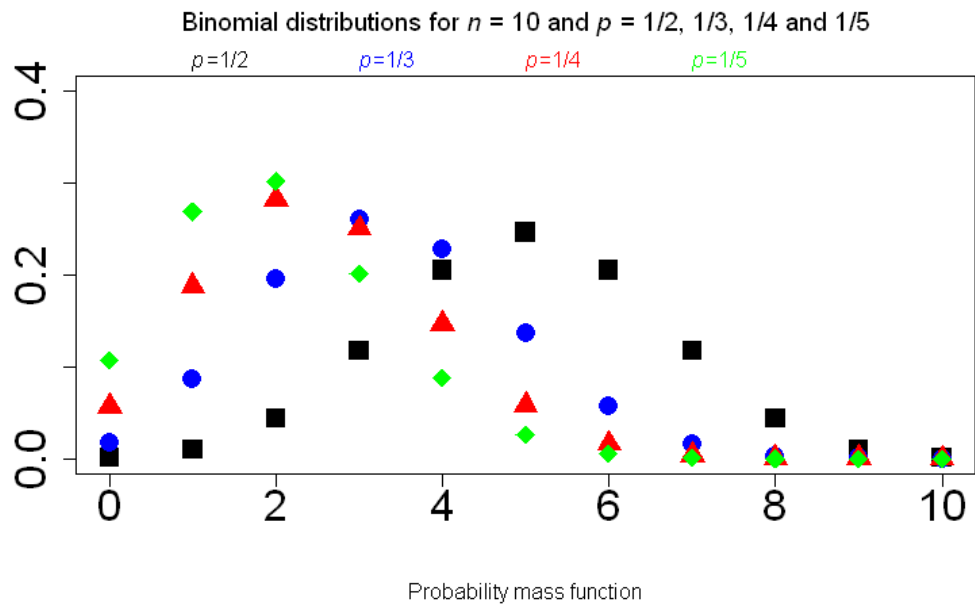
mtext(c(expression(paste(italic(n), "=10"))),
      adj=0, at=3, col=colors[2])

mtext(c(expression(paste(italic(n), "=15"))),
      adj=0, at=5, col=colors[3])

mtext(c(expression(paste(italic(n), "=20"))),
      adj=0, at=7, col=colors[4])

title(main=c(expression(paste("Binomial_distributions_for_",
                             italic(p), "_=1/2_and_",
                             italic(n), "_=5,10,15_and_20"))),
      sub="Probability_mass_function")

```

Problem

Suppose there are twelve multiple choice questions in an English class quiz. Each question has five possible answers, and only one of them is correct. Find the probability of having four or less correct answers if a student attempts to answer every question at random.

Solution

Since only one out of five possible answers is correct, the probability of answering a question correctly by random is $1/5 = 0.2$. We can find the probability of having exactly 4 correct answers by random attempts as follows.

```
> dbinom(4, size=12, prob=0.2)
[1] 0.1329
```

To find the probability of having four or less correct answers by random attempts, we apply the function `dbinom` with $x = 0, \dots, 4$.

```
> dbinom(0, size=12, prob=0.2) +
+ dbinom(1, size=12, prob=0.2) +
+ dbinom(2, size=12, prob=0.2) +
+ dbinom(3, size=12, prob=0.2) +
+ dbinom(4, size=12, prob=0.2)
[1] 0.9274
```

Alternatively, we can use the cumulative probability function for binomial distribution `pbinom`.

```
> pbinom(4, size=12, prob=0.2)
[1] 0.92744
```

Answer

The probability of four or less questions answered correctly by random in a twelve question multiple choice quiz is 92.7%.

2.3.2 Geometric Distribution $\mathcal{G}(p)$

A *geometric random variable* $\mathcal{G}(p)$ is defined as X : number of trials until the first success is observed (including the success trial). The probability distribution of X is called the geometric probability distribution. If X is a geometric random variable with probability of success p for each trial, i.e. $X \equiv \mathcal{G}(p)$, then

$$P(X = x) = (1 - p)^{x-1}p$$

$$E(X) = \frac{1}{p} \text{ and } V(X) = \frac{1-p}{p^2}$$

In **R**, `dgeom(x, p)` gives the density, `pgeom(q, p)` gives the distribution function and `rgeom(n, p)` gives the random generation for the geometric distribution.

For example, suppose an ordinary die is thrown repeatedly until the first time a '1' appears. The probability distribution of the number of times it is thrown is supported on the infinite set $\{1, 2, 3, \dots\}$ and is a geometric distribution with $p = 1/6$.

2.4 Poisson Distribution $\mathcal{P}(\lambda)$

The Poisson distribution $\mathcal{P}(\lambda)$ is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event. The Poisson distribution can also be used for the number of events in other specified intervals such as distance, area or volume.

The Poisson distribution is the probability distribution of independent event occurrences in an interval. A discrete random variable X is said to have a Poisson distribution with parameter $\lambda > 0$, if for $x = 0, 1, 2, \dots$ the probability mass function of X is given by:

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad \text{where } x = 1, 2, 3, \dots$$

The positive real number λ is equal to the expected value of X and also to its variance:

$$\lambda = E(X) = V(X).$$

In **R**, `dpois(x, lambda)` gives the density, `ppois(q, lambda)` gives the distribution function, `qpois(p, lambda)` gives the quantile function and `rpois(n, lambda)` provides random generation for the Poisson distribution with parameter λ .

Problem

If there are twelve cars crossing a bridge per minute on average, find the probability of having seventeen or more cars crossing the bridge in a particular minute.

Solution

The probability of having sixteen or less cars crossing the bridge in a particular minute is given by the function `ppois`.

```
> ppois(16, lambda=12)    # lower tail
[1] 0.89871
```

Hence the probability of having seventeen or more cars crossing the bridge in a minute is in the upper tail of the probability density function.

```
> ppois(16, lambda=12, lower=FALSE)    # upper tail
[1] 0.10129
```

Answer

If there are twelve cars crossing a bridge per minute on average, the probability of having seventeen or more cars crossing the bridge in a particular minute is 10.1%.

2.5 Normal Distribution $\mathcal{N}(\mu, \sigma)$

A Normal distribution $\mathcal{N}(\mu, \sigma)$ is bell-shaped and symmetric. It is characterized by a mean μ and standard deviation σ . μ describes where the corresponding curve is centered, and σ describes how much the curve spreads out around that center.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

In **R**, there is a large number of functions for dealing with the normal distributions. Density, distribution function or quantile function and random generation for the normal distribution are easily computed with mean (μ) equal to parameter *mean* and standard deviation (σ) equal to parameter *sd*.

The function **dnorm** gives the density, **pnorm** gives the distribution function, **qnorm** gives the quantile function, and **rnorm** generates random deviates.

If we let the mean $\mu = 0$ and the standard deviation $\sigma = 1$ in the *probability density function*, we get the probability density function for the *standard normal distribution*. The standard normal distribution can be plot in **R** as follows:

```
x=seq(-4,4,length=200)
y=1/sqrt(2*pi)*exp(-x^2/2)
plot(x,y,type="l",lwd=5,col="blue")
```

We can use the **dnorm** function for doing the same (with a small curve in yellow). We will assign before the **par** function to **TRUE**, so the next plotting will not clean the frame before drawing as if it were on a new device.

```
par(new = TRUE)
y=dnorm(x,mean=0,sd=1)
plot(x,y,type="l",lwd=1,col="yellow")
```

Plot then on the same graph three different probability density functions, the first with $\mu = 10$ and $\sigma = 5$ (in red), the second with $\mu = 40$ and $\sigma = 2.5$ (in green) and the third with $\mu = 70$ and $\sigma = 10$ (in blue) [slide 52 of Marc Sebban's lesson].

```
x <- seq(-10,100,.1)
normdensity1 <- dnorm(x,mean=10,sd=5)

normdensity2 <- dnorm(x,mean=40,sd=2.5)

normdensity3 <- dnorm(x,mean=70,sd=10)

plot(x, normdensity1, type="l", col="red",
      ylim=range(c(normdensity1, normdensity2, normdensity3)))
par(new = TRUE)
plot(x, normdensity2, type="l", col="green",
      ylim=range(c(normdensity1, normdensity2, normdensity3)),
      axes = FALSE, xlab = "", ylab = "")
par(new = TRUE)
```

```
plot(x, normdensity3, type="l", col="blue",
      ylim=range(c(normdensity1, normdensity2, normdensity3)),
      axes = FALSE, xlab = "", ylab = "")
```

68%, 95.7% and 99.7%

We will examine the probability that a randomly selected number from the standard normal distribution occurs within one standard deviation of the mean. This probability is represented by the area under the standard normal curve between $x = -1$ and $x = 1$. We will color this area in gray with the `polygon` function:

```
x=seq(-4,4,length=200)
y=dnorm(x)
plot(x,y,type="l", lwd=2, col="blue")
x=seq(-1,1,length=100)
y=dnorm(x)
polygon(c(-1,x,1),c(0,y,0),col="gray")
```

To find the area between $x = -1$ and $x = 1$, we must subtract the area to the left of $x = -1$ from the area to the left of $x = 1$:

```
pnorm(1,mean=0,sd=1)-pnorm(-1,mean=0,sd=1)
```

Do the same for two and three standard deviations:

```
x=seq(-4,4,length=200)
y=dnorm(x)
plot(x,y,type="l", lwd=2, col="blue")
x=seq(-2,2,length=200)
y=dnorm(x)
polygon(c(-2,x,2),c(0,y,0),col="gray")
pnorm(2,mean=0,sd=1)-pnorm(-2,mean=0,sd=1)
pnorm(3,mean=0,sd=1)-pnorm(-3,mean=0,sd=1)
```

Problem

A normal distribution with mean = 3500 grams and standard deviation = 600 grams is a reasonable model for the probability distribution of the continuous variable X : birth weight of a randomly selected full-term baby.

Question 1

What proportion of birth weights are between 2900 and 4700 grams?

Solution

```
pnorm(4700,mean=3500,sd=600)-pnorm(2900,mean=3500,sd=600)
```

Question 2

What birth weight w is exceeded only in 2.5% of the cases?


Solution

```
qnorm(1 - 2.5 / 100, mean = 3500, sd = 600)
```

Another solution by indicating that we are interested in the *upper tail* of the normal distribution:

```
qnorm(2.5 / 100, mean = 3500, sd = 600, lower.tail = FALSE)
```

Answers

The answers are the results computed in . You can check your results with the slides 56 and 57 of Marc Sebban's lesson. Moreover you will find below a graphical solution for the question 2.

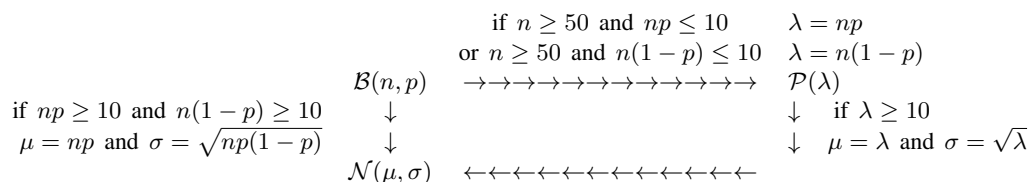
```
x <- seq(0, 7000, length=200)
y <- dnorm(x, mean= 3500, sd = 600)
# same as y <- dnorm((x - 3500) / 600)

plot(x, y, type="l", lwd=2, col="gray")

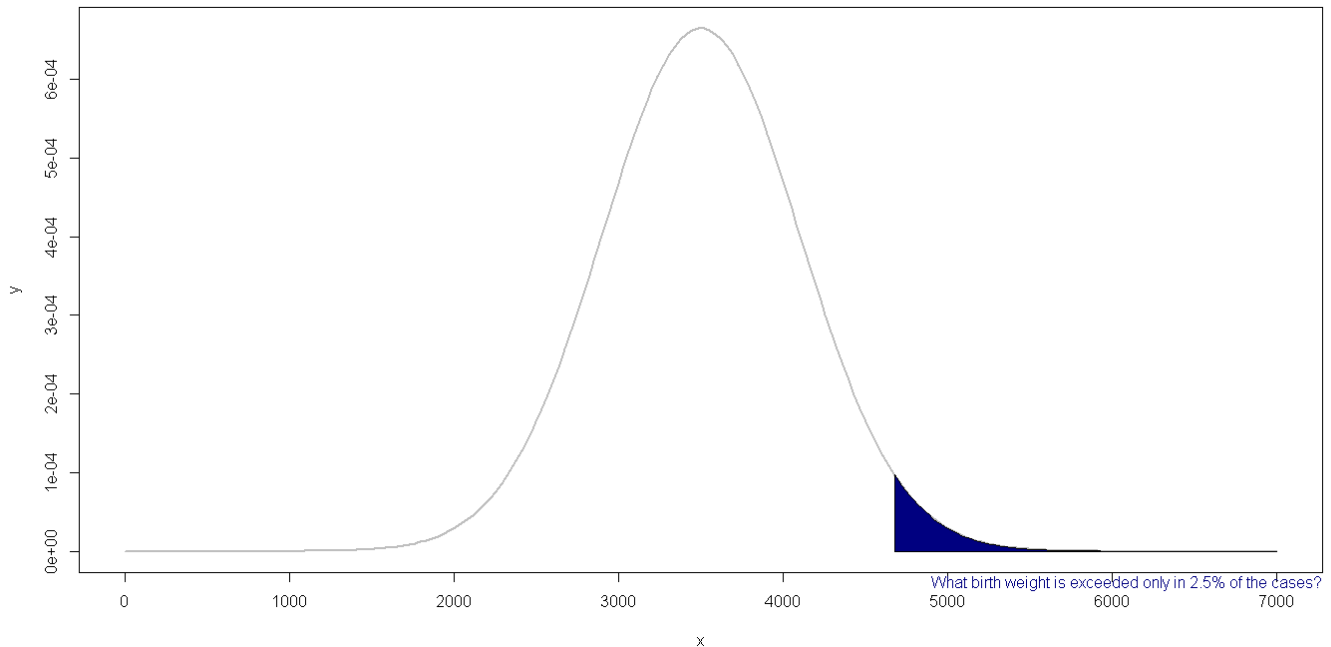
z <- qnorm(2.5 / 100, mean = 3500, sd = 600, lower.tail = FALSE)
x <- seq(z, 7000, length=100)
y <- dnorm(x, mean= 3500, sd = 600)

polygon(c(z, x, 7000), c(0, y, 0), col="navy")

mtext("What birth weight is exceeded only in 2.5% of the cases?",
      side=1, adj=1, col="navy")
```

Approximations

Note that when histograms don't follow the normal curve, don't use the normal approximation!



Binomial Approximation

The normal distribution can be used as an approximation to the binomial distribution, under certain circumstances, namely:

If $X \sim \mathcal{B}(n, p)$ and if n is large and/or p is close to $\frac{1}{2}$, then X is approximately $\mathcal{N}(np, np(1 - p))$.

In some cases, working out a problem using the normal distribution may be easier than using a binomial.

Poisson Approximation


The normal distribution can also be used to approximate the Poisson distribution for large values of λ (the mean of the Poisson distribution).

If $X \sim \mathcal{P}(\lambda)$ then for large values of λ , $X \sim \mathcal{N}(\lambda, \lambda)$ approximately.

Continuity Correction

The binomial and Poisson distributions are discrete random variables, whereas the normal distribution is continuous. We need to take this into account when we are using the normal distribution to approximate a binomial or Poisson using a continuity correction.

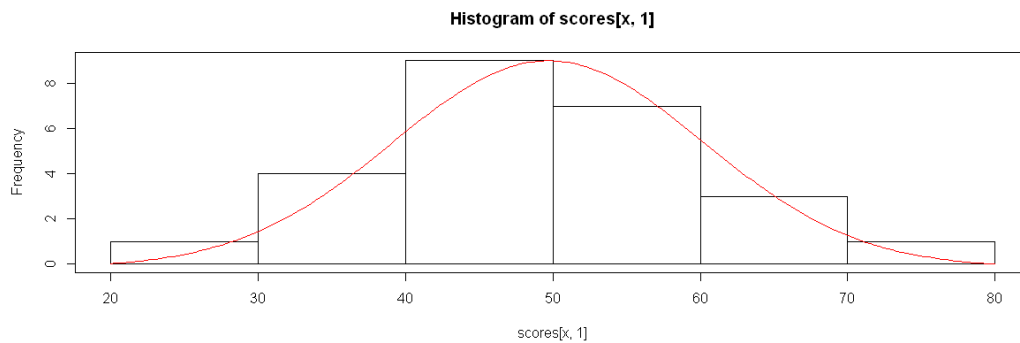
Example

Import the exam scores of 25 students. The data are: 39 41 47 58 65 37 37 49 56 59 62 36 48 52 64 29 44 43 41 52 57 54 72 50 and 50. You can download the file `scores.txt` from *Claroline* platform, import it into  (Import Dataset From Text File...) and the data will be stored in a data frame `scores` with 25 observations of 1 variable.

Plot the scores, then the histogram of the scores. Compute the mean and the standard deviation, then –on the same plot– plot the normal distribution that approximate the real data (without labels for X and Y-axis).

```
x <- seq(1:25)
plot(x, scores[x,1])
hist(scores[x,1])
mu <- mean(scores[,1])
sigma <- sd(scores[,1])
x2=seq(20,80,length=200)
par(new = TRUE)
y=dnorm(x2,mean=mu,sd=sigma)

plot(x2,y,type="l",lwd=1,col="red", xlab="", ylab="", yaxt="n")
```



How many students do you expect to find with a score equal or greater than 55, 60, 65 or 70? How many are they in reality?

```
# How many students do you expect to find with a score
# equal or greater than 55?

# Answer by using the normal distribution:
# 25 is the number of students


round(25 * (pnorm(55, mean=mu, sd=sigma, lower.tail=FALSE)))

# By counting the real values:

nb_scores <- 0
for (i in 1:25)
{
  if (scores[i,1] >= 55)
    nb_scores <- nb_scores + 1
}
print(nb_scores)
```


Exercises

Data Pre-processing

- Download the CSV file “students_data.csv” from *Claroline* (on your working directory) and import it into  (with the following commands) or with “Import Dataset” on RStudio (“From Text (readr)” and changing the Birthdate from character to Date format with `%d/%m/%Y`) then print the summary:

```
library(readr)
students <- read_csv("~/R/data/student_data.csv",
                     col_types =
                       cols(Birthdate = col_datetime(format = "%d/%m/%Y")))
summary(students)
```

- Transform the first column (first name and last name) for labelling the row names (e.g., by using a temporary dataframe “df”), then print the summary of the dataset and the dataset itself:

```
rownames(students) <- df[,1]
rm(df)
students[,1] <- NULL
students
summary(students)
```

- Remove the observations with “Non Available data” (NA), then print the summary:

```
students <- na.omit(students) # Remove NA observations
summary(students)
```

Questions

- Print the coefficient correlations between all the continuous attributes (you can remove the birthdate). What are the significant correlations?
- Plot the histogram of the size of the students (try different breaks).
- Compute the mean and standard deviation of the size variable. Does it follow a normal distribution?
- By using a normal approximation of the size, how many students do you expect to find in the classroom with a size larger or equal to 180 cm? How many are they in reality?
- Plot the histogram of the age of the students (use the age variable as factor).
- Compute the mean and standard deviation of the age variable. Does it follow a normal distribution?
- By using a normal approximation of the age, how many students do you expect to find in the classroom with an age smaller than the mean less one standard deviation? And with an age smaller than the mean less two standard deviations? How many are they in reality?

3 Recommended Readings

The main literature for this section is:

- Adler (2010), “R in a Nutshell – a Desktop Quick Reference,” Chapter 17 “Probability Distributions.”
- Baayen (2008), “Analyzing Linguistic Data: A Practical Introduction to Statistics using R,” Chapter 3 “Probability distributions.”
- Chihara and Hesterberg (2011), “Mathematical Statistics with Resampling and R.”
- Cohen and Cohen (2008), “Statistics and Data with R: An Applied Approach Through Examples,” Part II “Probability, densities and distributions.”
- Crawley (2005), “Statistics: An Introduction using R,” Chapter 5 “Single Samples.”
- Dalgaard (2008), “Introductory Statistics with R,” Chapter 3 “Probability and distributions.”
- Kerns (2010), “Introduction to Probability and Statistics Using R,” Chapter 5 to 9.
- Peck et al. (2012), “Introduction to Statistics and Data Analysis,” Chapter 6 “Probability” and Chapter 7 “Random Variables and Probability Distributions.”
- Teetor (2011), “R Cookbook”, Chapter 8 “Probability.”
- Venables et al. (2013), “An Introduction to R,” Chapter 8 “Probability distributions.”

References

- Adler, J. (2010). *R in a Nutshell – a Desktop Quick Reference*. O’Reilly.
- Baayen, R. H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.
- Chihara, L. M. and T. C. Hesterberg (2011). *Mathematical Statistics with Resampling and R*. John Wiley & Sons, Ltd.
- Cohen, Y. and J. Y. Cohen (2008). *Statistics and Data with R: An Applied Approach Through Examples*. John Wiley & Sons, Ltd.
- Crawley, M. J. (2005). *Statistics: An Introduction using R*. John Wiley & Sons, Ltd.
- Dalgaard, P. (2008). *Introductory Statistics with R* (2nd ed.). Springer.
- Kerns, G. J. (2010). *Introduction to Probability and Statistics Using R* (1st ed.).
- Peck, R., C. Olsen, and J. L. Devore (2012). *Introduction to Statistics and Data Analysis* (4th ed.). Brooks / Cole, Cengage Learning.
- Teetor, P. (2011). *R Cookbook*. O’Reilly.
- Venables, W. N., D. M. Smith, and the R Core Team (2013). An introduction to R –notes on R: A programming environment for data analysis and graphics.
URL <http://cran.r-project.org/doc/manuals/R-intro.html>.