**PERCENTILES & DISTRIBUTION FUNCTIONS OF RANDOM VARIABLE**

1. **Find the percentiles of a data set (vector)**

Use the command: > quantile(x, …)

Example 1:

Command: > x <- c(14, 12, 19, 23, 5, 13, 28, 17)

> quantile(x, probs = 0.3) or quantile(x, 0.3)

Result: 30%

13.1

Noted that the result is different from the value presented in the lecture. In fact, there are nine methods to compute quantile. By default, R uses **Type 7**

Remarks:

* Definition of quantile: Let  be some observations and  be the increasingly sorted observations (i.e., order statistics). Then the  for  is defined by



* If  is no integer the is the  (in which *ceiling* means rounding to the next larger integer)
* If  is integer the is not unique and all values in the interval  are valid . The most obvious approach is to use the midpoint of the interval. But other methods were also proposed (totally 9 methods)
* Percentiles are special cases of quantiles when 

Example 2:

Command: > x <- c(14, 12, 19, 23, 5, 13, 28, 17)

> quantile(x, probs = 0.3, Type =2)

Result: 30%

13

Example 3: Find the quartiles

Command: > x <- c(106, 109, 114, 116, 121, 122, 125, 129)

> quantile(x, probs = seq(0,1,0.25))

Result: 0% 25% 50% 75% 100%

106.00 112.75 118.50 122.75 129.00

> quantile(x, probs = seq(0,1,0.25), type =2)

Result: 0% 25% 50% 75% 100%

106.0 111.5 118.5 123.5 129.0

1. **Construct a histogram**

Use the command: > hist(x, …)

Example 3: Construct a histogram using “trees” data set included with R. The data set has three variables, i.e., Girth, Height, and Volume of timber in 31 black cherry trees

The data in this data set can be viewed by use of the command:

> trees or > View(tress)

Command:

* > hist(trees$Height)

A screenshot of a cell phone

Description automatically generated

In the above figure, the number of groups is defined using “Sturges Rule”:  (in which *n* is the number of observations

* > hist(trees$Height, breaks = 3, col = “green”)

A picture containing screenshot

Description automatically generated

Parameter “breaks”:  the number of cells for the histogram (but it is just a suggestion; this value can be changed as the breakpoints will be set to convenient values)

* > hist(trees$Height, breaks = 3, probability = TRUE, col = “green”)

A picture containing screenshot

Description automatically generated

We can also use the function qplot (the package ggplot2 must be installed and loaded!)

Command: > qplot(Height, data = trees, binwidth = 15)

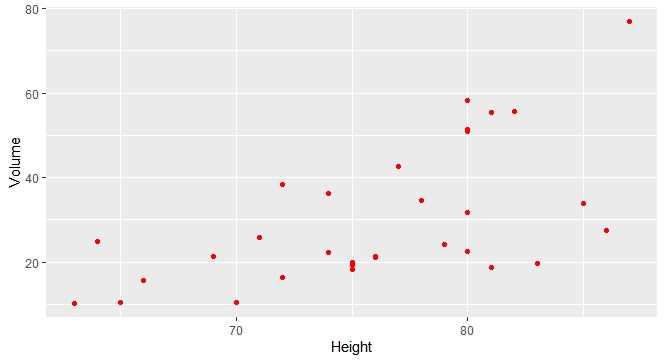
Or > qplot(trees$Height, binwidth = 15)

A screenshot of a cell phone

Description automatically generated

Note: qplot can also be used to construct scatter plot

Command: > qplot(Height, Volume, data = trees, col = I(“red”))



1. **Construct a boxplot**

Use the command: boxplot

Example: > boxplot(trees)

A screenshot of a cell phone

Description automatically generated

Example: > x <- c(1,2,3,4,5)

> y <- c(10,13,15,28,19)

> boxplot(x,y, names = c("data 1", "data 2"))

A screenshot of a cell phone

Description automatically generated

1. **Generate data from a random variable**

Use the commands: rnorm, rt, rweibull, rbinorm, rpois, … to generate data following normal, student, Weibull, Binomial, Poisson, … distributions

Remark: Replace r by d, p, q to generate density function, distribution function, quantile function (i.e, the inverse!)

Example:

Command: > rnorm(5, mean = 4, sd = 1)

[1] 3.971451 4.356434 4.109787 5.359577 5.036098

Command: > dnorm(3, mean = 0, sd = 1) !density function of standard normal distribution

[1] 0.004431848 !no need to specify mean and sd in this case

Command: > pnorm(3, mean = 0, sd = 1) !cumulation function of standard normal distribution

[1] 0.9986501 !no need to specify mean and sd in this case

Command: > qnorm(0.00135, mean = 0, sd = 1) !inverse function of standard normal distribution

[1] -2.999977 !no need to specify mean and sd in this case

|  |  |  |  |
| --- | --- | --- | --- |
| **Distribution** | **R name** | **Distribution** | **R name** |
| **Beta** | beta | **Lognormal** | lnorm |
| **Binomial** | binom | **Negative Binomial** | nbinom |
| **Cauchy** | cauchy | **Normal** | norm |
| **Chisquare** | chisq | **Poisson** | pois |
| **Exponential** | exp | **Student t** | t |
| **F** | f | **Uniform** | unif |
| **Gamma** | gamma | **Tukey** | tukey |
| **Geometric** | geom | **Weibull** | weibull |
| **Hypergeometric** | hyper | **Wilcoxon** | wilcox |
| **Logistic** | logis |  |  |

Tukey: Studentized Range Distribution. Only have ptukey and qtukey

Wilcoxon: Wilcoxon ran sum statistic

**PROBABILITY PLOTS**

1. **Normal Probability Plot**

Example 1:

Command:

> y <- rnorm(200, mean = 0, sd = 1) ! mean = 0, sd = 1: default values, no need to declare

> qqnorm(y)

> qqline(y, col = 2) ! col : define a color for the line

You can declare col = “red”, “blue”, “green”, “magenta”, “pink”,….

A screenshot of a social media post

Description automatically generated

Example 2:

Command:

> y <- rnorm(200, mean = 10, sd = 2)

> qqnorm(y)

> qqline(y, col = 2)

A close up of a map

Description automatically generated

Example 3:

Command:

> y <- rpois(200, 3)

> qqnorm(y)

> qqline(y, col = 2)

A close up of a map

Description automatically generated

Example 4:

Command:

> y <- rbinom(200, 50, 0.3)

> qqnorm(y)

> qqline(y, col = 2)

A close up of a map

Description automatically generated

Example 5:

Command:

> y <- runif(200, min = 0, max = 1) ! min = 0, max = 1: default values, no need to declare

> qqnorm(y)

> qqline(y, col = 2)

A close up of a map

Description automatically generated

1. **Probability Plot for Other Distributions**

Example 1:

Command:

> x <- rpois(200, 3) ! considered as sample data

> qqplot(x, rpois(1000,3)) ! rpois(300,3): considered as the second sample data or

data from a theoretical distribution

A screenshot of a social media post

Description automatically generated

Example 2:

Command:

> x <- rt(300, df = 5) ! considered as the first sample data

> y <- rt(500, df = 2) ! considered as the second sample data

> qqplot(x, y, xlab = "Sample Data 1", ylab = "Sample Data 2")

! check if the two samples have the same distribution

A screenshot of a cell phone

Description automatically generated