University of California, Los Angeles
Department of Statistics

Statistics 13 **Instructor: Nicolas Christou**

Data analysis with R - Some simple commands

When you are in R, the command line begins with

```
>
```

To read data from a website use the following command:

```
a <- read.table("http://www.stat.ucla.edu/~nchristo/statistics13/body_fat.txt", header=TRUE)
```

The result of the command `read.table` is a "data frame" (it looks like a table). In our example we give the name `data` to our data frame. The columns of a data frame are variables. This file contains data on percentage of body fat determined by underwater weighing and various body circumference measurements for 251 men. Here is the variable description:

Variable	Description
x_1	Density determined from underwater weighing
y	Percent body fat from Siri's (1956) equation
x_3	Age (years)
x_4	Weight (lbs)
x_5	Height (inches)
x_6	Neck circumference (cm)
x_7	Chest circumference (cm)
x_8	Abdomen 2 circumference (cm)
x_9	Hip circumference (cm)
x_{10}	Thigh circumference (cm)
x_{11}	Knee circumference (cm)
x_{12}	Ankle circumference (cm)
x_{13}	Biceps (extended) circumference (cm)
x_{14}	Forearm circumference (cm)
x_{15}	Wrist circumference (cm)

If the data file is on your computer (e.g. on your desktop), first you need to change the working directory by clicking on `Misc` at the top of your screen and then read the data as follows:

```
> a <- read.table("filename.txt", header=TRUE)
```

Note: the expression `<-` is an assignment operator.
Once we read the data we can display them by simply typing at the command line `< a`. Or if we want we can display the first 6 rows of the data by typing `> head(a)`. Here is the output:

```
> head(a)
     x1    y x3     x4    x5   x6   x7   x8    x9  x10  x11  x12  x13  x14  x15
1 1.0853  6.1 22 173.25 72.25 38.5 93.6 83.0  98.7 58.7 37.3 23.4 30.5 28.9 18.2
2 1.0414 25.3 22 154.00 66.25 34.0 95.8 87.9  99.2 59.6 38.9 24.0 28.8 25.2 16.6
3 1.0754 10.3 23 188.15 77.50 38.0 96.6 85.3 102.5 59.1 37.6 23.2 31.8 29.7 18.3
4 1.0722 11.7 23 198.25 73.50 42.1 99.6 88.6 104.1 63.1 41.7 25.0 35.6 30.0 19.2
5 1.0708 12.3 23 154.25 67.75 36.2 93.1 85.2  94.5 59.0 37.3 21.9 32.0 27.4 17.1
6 1.0775  9.4 23 159.75 72.25 35.5 92.1 77.1  93.9 56.1 36.1 22.7 30.5 27.2 18.2
```

Useful commands:

- Extracting one variable from the data frame (e.g. the second variable): `> a[,2]`
- Another way to extract a variable : `> a$y`
- Similarly if we want to access a particular row in our data (e.g. first row): `> a[1,]`
- To list all the data simply type: `> a`
- To compute the mean of all the variables in the data set: `> mean(a)`
- To compute the mean of just one variable: `> mean(a$y)`
- To compute the mean of variables 2 and 3: `> mean(a[,c(2,3)])`
- To compute the variance of one variable: `> var(a$y)`
- To compute the variance-covariance matrix of all the variables: `> cov(a)`
- To compute the variance-covariance matrix of all the variables except the first variable: `> cov(a[,-1])`
- To compute the variance-covariance matrix of variables 1, 2, and 3: `> cov(a[,c(1,2,3)])` or `cov(a[,1:3])`
- To compute the variance-covariance matrix of variables 1, 2, and 5: `> cov(a[,c(1,2,5)])`
- To compute the correlation matrix: As above, replace `cov` with `cor`, for example: `> cor(data[,c(1,2,3)])`
- To compute summary statistics for all the variables: `> summary(a)`.
- To construct stem-and-leaf plot, histogram, boxplot:

```
> stem(a$y)
> boxplot(a$y)
> hist(a$y)
```

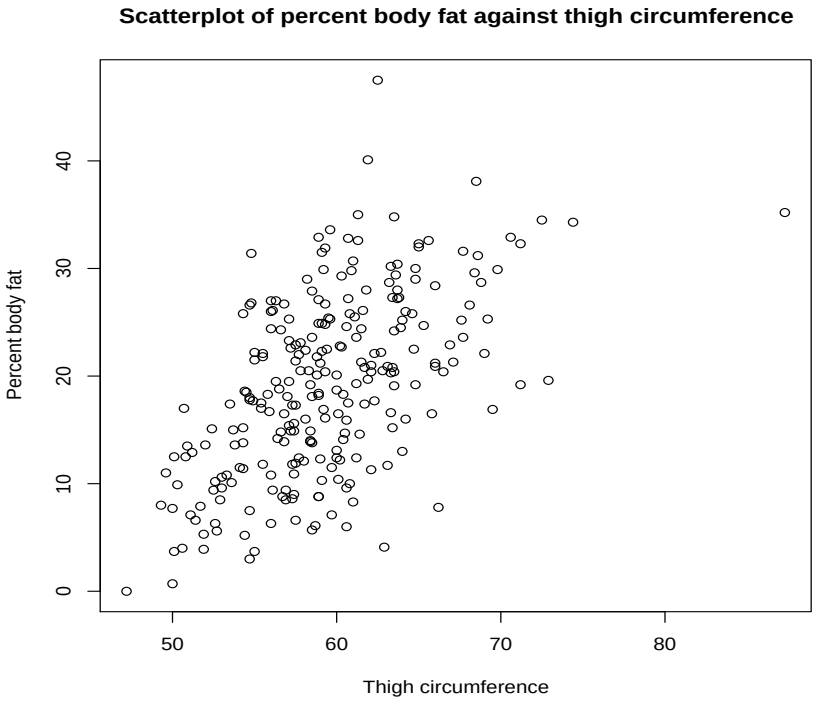
- To plot variable `y` against variable `x10`:

```
> plot(a$x10, a$y)
```

- And you can give names to the axes and to your plot:

```
> plot(a$x10, a$y, main="Scatterplot of percent body fat against
thigh circumference", ylab="Percent body fat",
xlab="Thigh circumference")
```

And here is the plot:



- To save a plot as a pdf file under the working directory (e.g. your desktop):

```
> pdf("box.pdf")
> boxplot(a$y)
> dev.off()
```

A box plot of the variable `y` can be found on your current working directory with the name `box.pdf`.

If you want to read more about a specific command (for example about histograms and boxplots) at the command line you type the following:

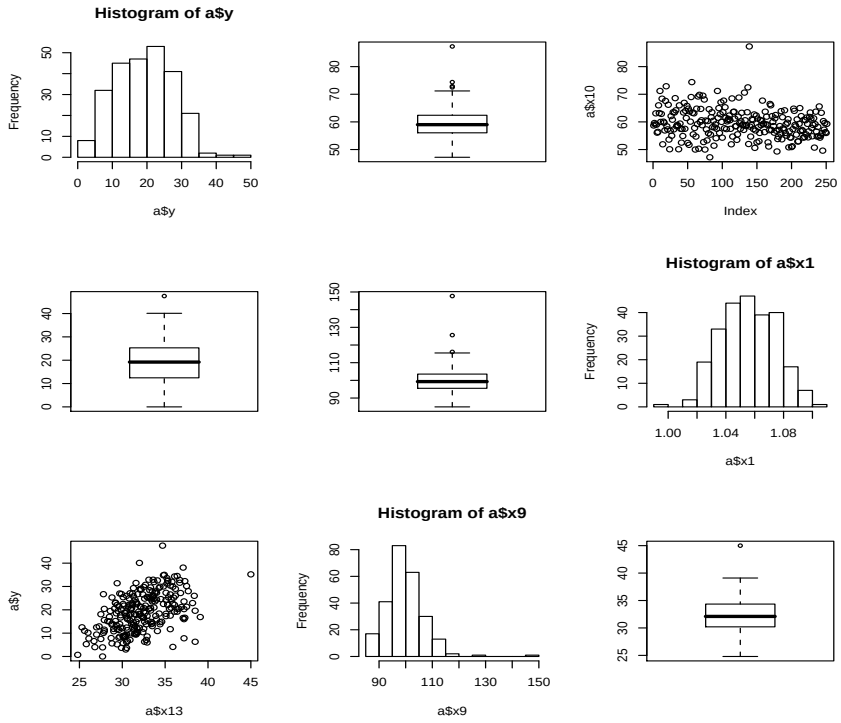
```
> ?hist
> ?boxplot
```

- **Exercise:**
Construct the same plots with different variables and save them on your desktop.

Create multiple graphs on one page. Suppose 9 graphs, 3×3:

```
pdf("plot9.pdf")
par(mfrow=c(3,3))
hist(a$y)
boxplot(a$x10)
plot(a$x10, a$y)
boxplot(a$y)
boxplot(a$x9)
hist(a$x1)
plot(a$x13, a$y)
hist(a$x9)
boxplot(a$x13)
dev.off()
```

And here is the plot:

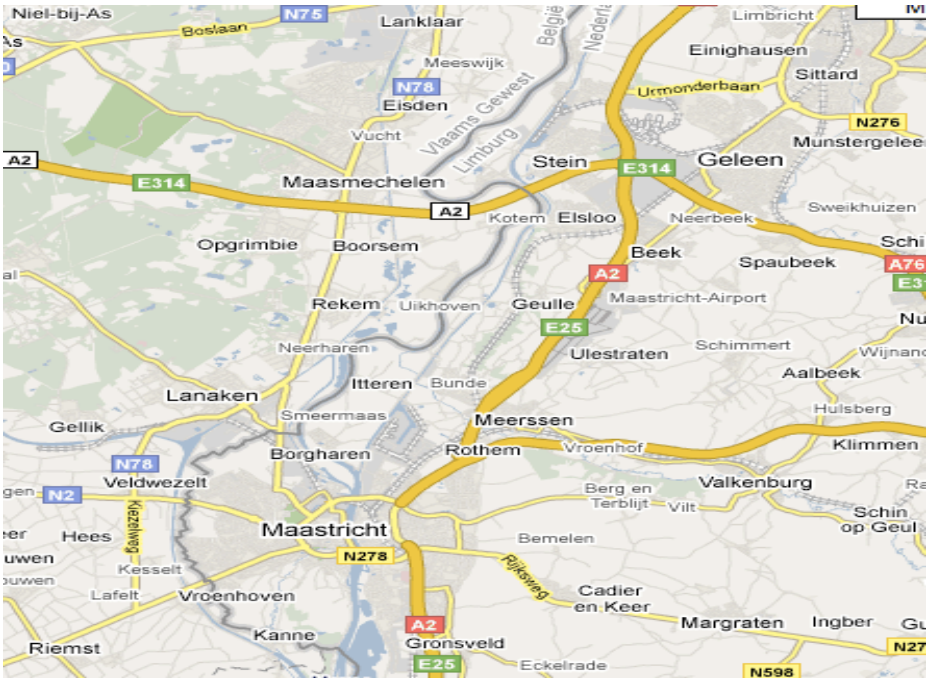


Create subsets:
The following simple commands will create subsets of the original data frame `a`:

```
a1 <- a[, 1:3] #A new data frame with only the first three columns.
a2 <- a[, c(1:3,8,10)] #A new data frame with columns 1,2,3,8,10.
```

Another data set:

The following data were collected in the area west of the town Stein in the Netherlands near the river *Meuse* (Dutch *Maas*) river (see map below). The actual data set contains many variables but here we will use the x, y coordinates and the concentration of lead and zinc in *ppm* at each data point. The motivation for this study was to predict the concentration of heavy metals around the banks of the Maas river in this area. These heavy metals were accumulated over the years because of the river pollution. Here is the area of study:



Exercise:

a. You can access these data using:

```
b <- read.table("http://www.stat.ucla.edu/~nchristo/statistics13/soil.txt", header=TRUE)
```

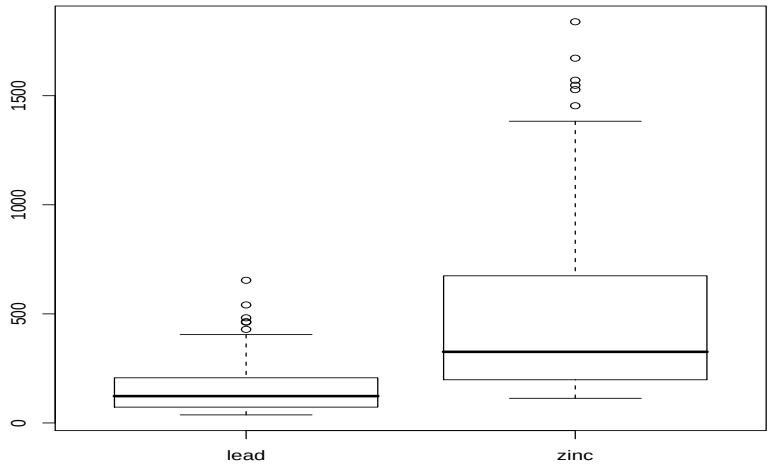
b. Construct the stem-and-leaf plot, histrogram, and boxplot for each one of the two variables (lead and zinc), and compute the summary statistics. What do you observe?

c. Transform the data in order to produce a symmetrical histrogram. Here is what you can do:

```
> log_lead <- log(b$lead)
> log_zinc <- log(b$zinc)
```

Construct the stem-and-leaf plot, histrogram, and boxplot for each one of the new variables (`log_lead` and `log_zinc`), and compute the summary statistics. What do you observe now.

Here is a side by side boxplot of the variables `lead` and `zinc`. `boxplot(b[,3:4])` or `boxplot(b$lead, b$zinc)`.



More on subsets:
Suppose we want to create a new data frame with the rows corresponding to lead concentration above 450 ppm, or lead concentration between 45 and 60 ppm.

```
which(b$lead>450)  #This will identify the rows.
b1 <- b[which(b$lead>450),]  #This will create the new data frame.

which(b$lead>45 & b$lead<60)  #Another subset.
b2 <- b[which(b$lead>45 & b$lead<60),]
```

Other useful commands in R:

- To enter data in R use <- or the equal sign =. The <- is preferred. Here are some examples:

```
> x <- c(1,2,3,4,5)
> y <- c(10,20,30,40,50)

> q <- data.frame(cbind(x,y))
```

And here is what you get:

```
> x
[1] 1 2 3 4 5

> q
     x  y
[1,] 1 10
[2,] 2 20
[3,] 3 30
[4,] 4 40
[5,] 5 50
```

- To rename variables:

```
> names(q) <- c("a", "b")
> q
  a  b
1 1 10
2 2 20
3 3 30
4 4 40
5 5 50
```

An example using the `maps` package

Data on ozone and other pollutants are collected on a regular basis. The data set for this example concerns 175 locations for ozone (ppm) in California on 08 August 2005. You can read more about smog-causing pollutants at

`http://www.nytimes.com/2010/01/08/science/earth/08smog.html?th&emc=th`

The data can be accessed here:

```
a <- read.table("http://www.stat.ucla.edu/~nchristo/statistics13/ozone.txt", header=TRUE)
```

Once you install the package `maps` using `install.packages("maps")` you can load it in `R` as follows:

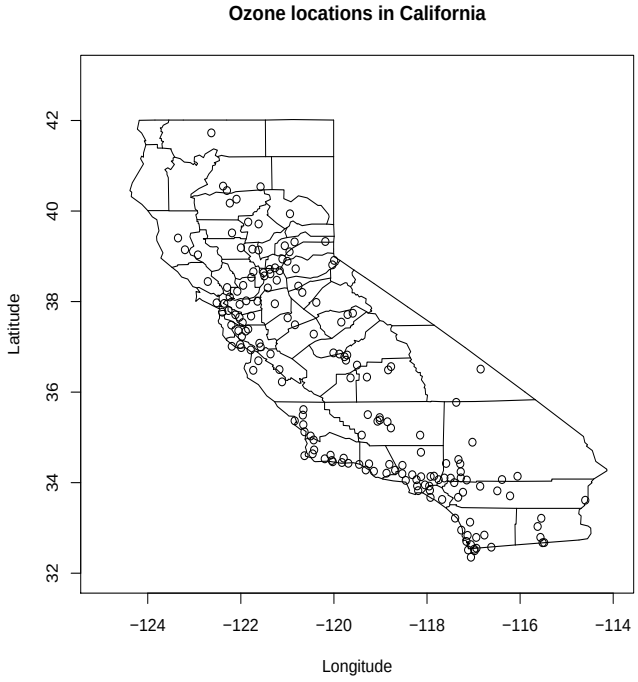
```
library(maps)
```

We can display the data points and the map using the following commands:

```
plot(a$x, a$y, xlim=c(-125,-114),ylim=c(32,43), xlab="Longitude",
ylab="Latitude", main="Ozone locations in California")

map("county", "ca",add=TRUE)
```

Here is the plot:



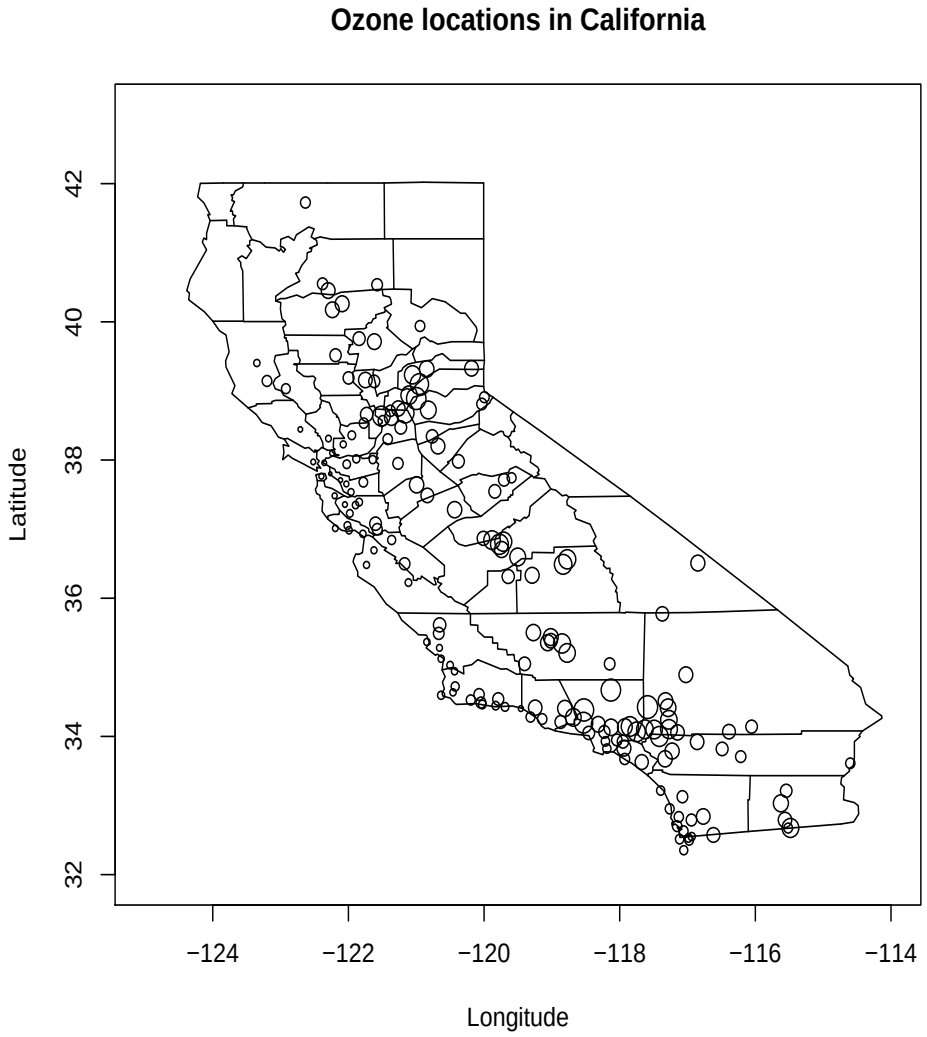
We can also plot the points relative to their value (larger values will be displayed with larger circles). Here are the commands:

```
plot(a$x, a$y, xlim=c(-125,-114),ylim=c(32,43), xlab="Longitude",
ylab="Latitude", main="Ozone locations in California", "n")

map("county", "ca",add=TRUE)

points(a$x, a$y, cex=a$o3/mean(a$o3))
```

Here is the plot:



The following chart illustrates the health-related interpretation of the Ozone data in terms of the particulate (particles per million, ppm) recordings, according to the National Oceanic and Atmospheric Administration's (NOAA) Air Quality Index (AQI).

`http://www.noaa.gov/`

Air Quality Index for Ozone

Concentration Range (ppm)	Air Quality Description	Cautionary Statements for Ozone
0.00 – 0.060 ppm	Good	No health impacts are expected
0.061 – 0.075 ppm	Moderate	Unusually sensitive people should consider limited prolonged outdoor exertion
0.076 – 0.104 ppm	Unhealthy for Sensitive Groups	Active children and adults, and people with respiratory conditions (e.g., asthma) should limit prolonged outdoor exertion
0.105 – 0.115 ppm	Unhealthy	Active children and adults, and people with respiratory conditions (e.g., asthma) should avoid prolonged outdoor exertion. Everyone else, especially children and elderly, should limit prolonged outdoor exertion
0.116 – 0.374 ppm	Very Unhealthy	Active children and adults, and people with respiratory conditions (e.g., asthma) should avoid all outdoor exertion. Everyone else, especially children and elderly, should limit outdoor exertion

What is next?

Try to match data location with the Air Quality Index:

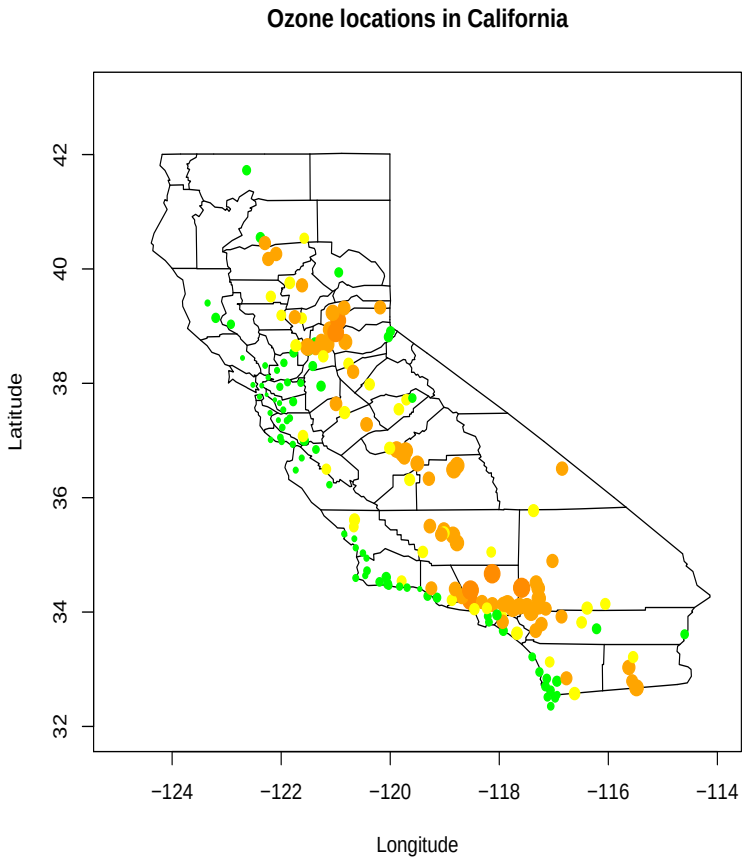
```
AQI_colors <- c("green", "yellow", "orange", "dark orange", "red")
AQI_levels <- cut(a$o3, c(0, 0.06, 0.075, 0.104, 0.115, 0.374))

as.numeric(AQI_levels)

plot(a$x,a$y, xlim=c(-125,-114),ylim=c(32,43), xlab="Longitude",
ylab="Latitude", main="Ozone locations in California", "n")

map("county", "ca",add=TRUE)
points(a$x,a$y, cex=a$o3/mean(a$o3),
       col=AQI_colors[as.numeric(AQI_levels)], pch=19)
```

Here is the plot:



Computing probabilities and percentiles using `R`:

- Probabilities of the standard normal distribution, $Z \sim N(0,1)$:
  $P(Z < -1.96)$: `pnorm(-1.96)`
  $P(Z < 1.96)$: `pnorm(1.96)`
  $P(Z < 0.50)$: `pnorm(0.50)`

- Probabilities of the $X \sim N(5,2)$, this is normal with $\mu = 5, \sigma = 2$:
  $P(X < 2)$: `pnorm(2, mean=5, sd=2)`
  $P(3.5 < X < 6.5)$: `pnorm(6.5, mean=5, sd=2) - pnorm(3.5, mean=5, sd=2)`

- Percentiles of $Z \sim N(0,1)$:
  Find c such that $P(Z < c) = 0.95$: `qnorm(0.95)`
  80th percentile: `qnorm(0.80)`

- Density of $N(0,1)$:
  `dnorm(1.5)`

- Probabilities of $X \sim \chi^2_{10}$:
  $P(X < 5)$: `pchisq(5, 10)`
  $P(X > 5)$: pchisq(5, 10, lower.tail=FALSE)

- Percentiles of $X \sim \chi^2_{10}$:
  95th percentile of χ^2_{10}: `qchisq(0.95, 10)`.

- Probabilities of $X \sim t_{10}$:
  $P(X < 2)$: `pt(2, 10)`
  $P(X > 2)$: pt(2, 10, lower.tail=FALSE)

- Percentiles of $X \sim t_{10}$:
  95th percentile of t_{10}: `qt(0.95, 10)`.

Random samples from distributions:

```
x <- rnorm(100)
hist(x)

rnorm(100, mean=5, sd=2)
```