

Assignment 1, EDDA 2017

Arash and Fabio, group 22

5 April 2017

Introduction

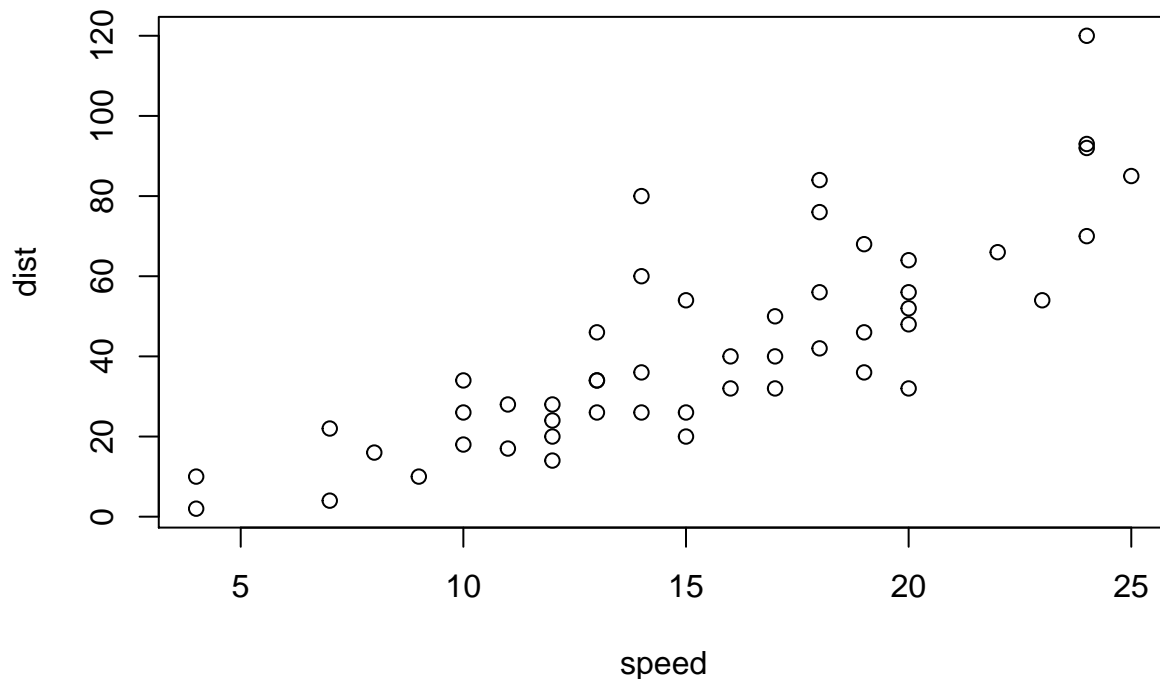
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. R Markdown files permit you to interweave R code with ordinary text to produce well-formatted data analysis reports that are easy to modify. The R Markdown file itself shows the readers exactly how you got the results in your report. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. Inline R code: for the build-in R dataset `cars`, there were 50 cars studied. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##   Mean  :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
##   Max.  :25.0    Max.   :120.00
```

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Equations

You can also include L^AT_EX equations in your report: inline $\frac{d}{dx} (\int_0^x f(u) du) = f(x)$ and in the display mode:

$$\frac{d}{dx} \left(\int_0^x f(u) du \right) = f(x).$$

Footnotes

Here is the use of a footnote¹.

Tables

It is also easy to make table by using knit's `kable` function:

Table 1: A knit kable.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

Varia

A friend once said:

It's always better to give than to receive.

This text is displayed verbatim / preformatted

<http://example.com>

linked phrase

italic **bold**

italic **bold**

¹This is a footnote.

Exercise 1

1.1 Some R-commands

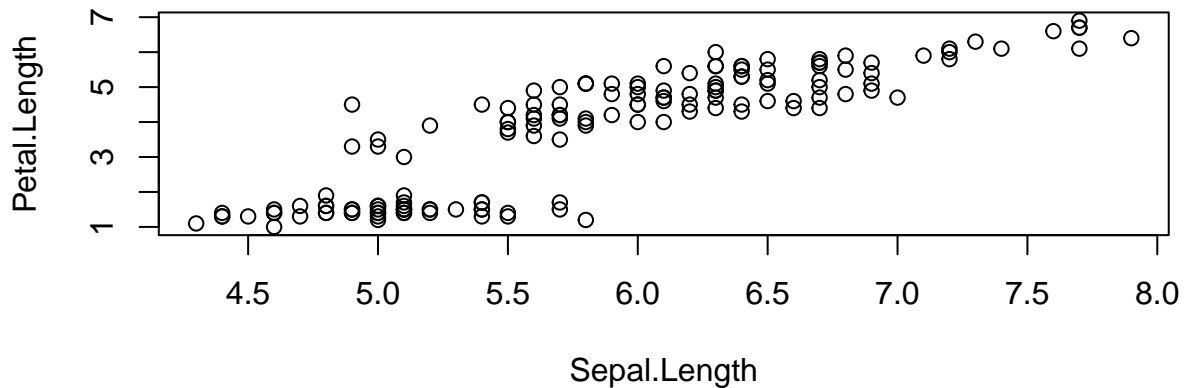
```
x=rep(c("A","B"),each=5); x  
## [1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B"  
sample(x)  
## [1] "B" "B" "A" "B" "A" "A" "B" "A" "B" "A"  
rbinom(10,1,0.5)  
## [1] 1 1 0 0 0 0 1 1 0 1  
rbinom(10,1,0.5)  
## [1] 1 1 1 0 0 1 1 0 1 0  
rbinom(5,1,0.8)  
## [1] 1 1 1 1 1  
x=rnorm(100)
```

The same code chunk but with all the output collapsed into single block.

```
x=rep(c("A","B"),each=5); x  
## [1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B"  
sample(x)  
## [1] "B" "B" "A" "A" "B" "A" "B" "B" "A" "A"  
rbinom(10,1,0.5)  
## [1] 0 1 1 0 1 0 0 0 1 0  
rbinom(10,1,0.5)  
## [1] 1 0 1 1 0 0 0 0 1 0  
rbinom(5,1,0.8)  
## [1] 0 1 0 0 1  
x=rnorm(100)
```

You can arrange for figures to span across the entire page by using the `fig.fullwidth` chunk option. Note the use of the `fig.width` and `fig.height` chunk options to establish the proportions of the figure. Full width figures look much better if their height is minimized.

```
plot(iris$Sepal.Length,iris$Petal.Length,xlab="Sepal.Length",ylab="Petal.Length")
```



Chunk option `fig.align` takes values 'left', 'right', or 'center'.

Exercise 2

2.1 T-test

Below we perform a one sample t-test for the artificial data (that we generated ourselves).

```
mu=0.2
x=rnorm(100,mu,1) # creating artificial data
t.test(x,mean=0)  # t.test(x,alternative=c("two.sided"),conf.level=0.95,mu=10)

##
## One Sample t-test
##
## data:  x
## t = 1.4598, df = 99, p-value = 0.1475
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.05368907  0.35262567
## sample estimates:
## mean of x
## 0.1494683
```

Of course, we often do not need to report the whole output of R-commands, only certain values of the output. For example, below we perform a two-sample t-test and report only the value of t-statistics and the p-value.

```
mu=0;nu=0.5
x=rnorm(50,mu,1); y=rnorm(50,nu,1) # creating artificial data
ttest=t.test(x,y)
```

The value of t-statistics in the above evaluation is -2.83 and the p-value is 0.0056.

EXERCISE 1 If you place the file `assign1.RData` in your R-directory and type `load(file="assign1.RData")` you will have data vectors `x1`, `x2`, ..., `x5` in your R working directory. Make a histogram and a QQ-plot for each of them, and decide which ones could have been sampled from a normal

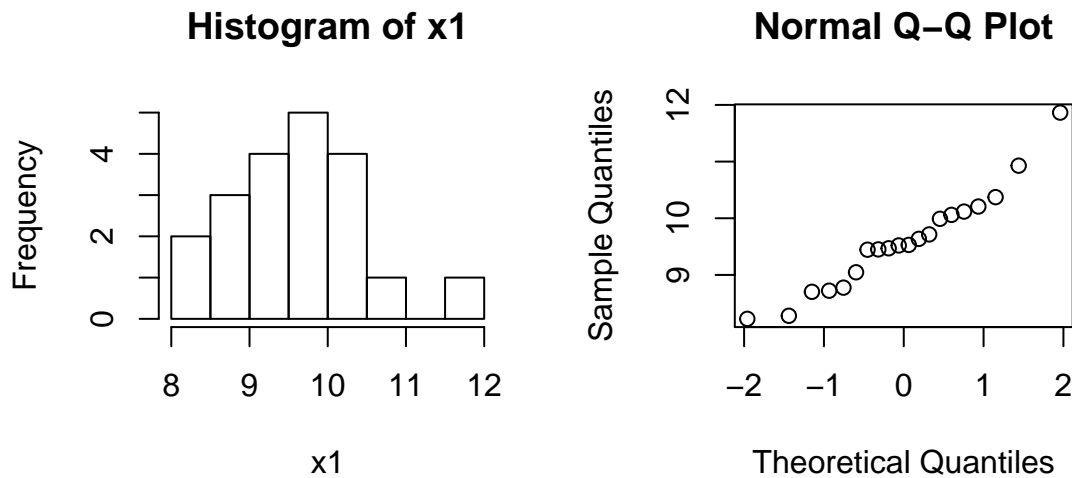
distribution. Experiment by simulating some normal samples of similar sizes and looking at their QQ-plots, before you make up your mind!

```
load(file="assign1.RData")
```

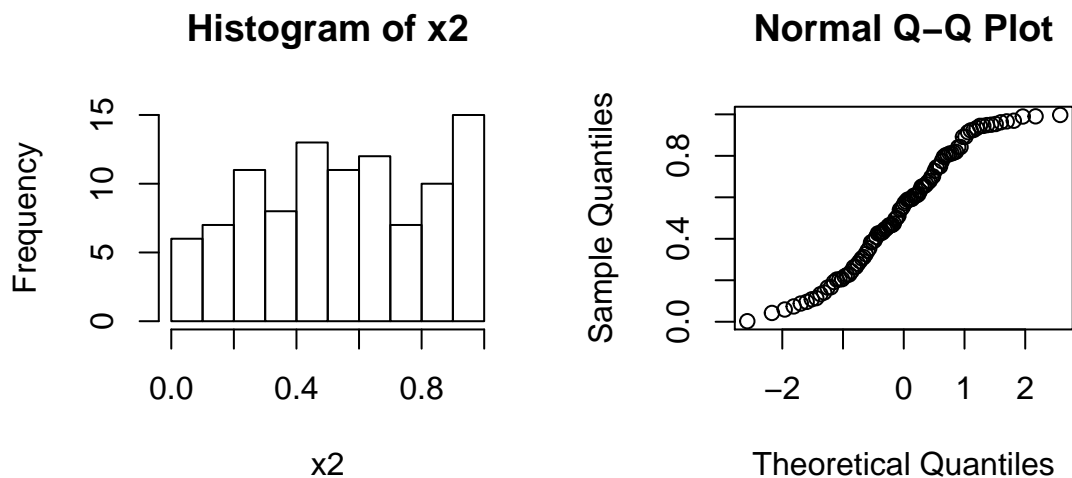
1.2 Figures

You can control the size and placement of figures. For example, you can put two figures (or more) next to each other. Use `par(mfrow=c(n,m))` to create `n` by `m` plots in one picture in R. You can adjust the proportions of figures using the `fig.width` and `fig.height` chunk options. These are specified in inches, and will be automatically scaled down to fit within the handout margin.

```
par(mfrow=c(1,2))
hist(x1); qqnorm(x1)
```

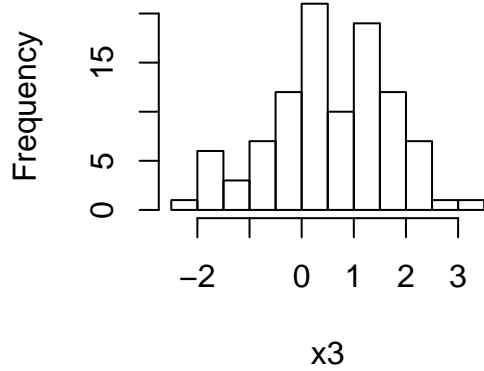


```
hist(x2); qqnorm(x2)
```

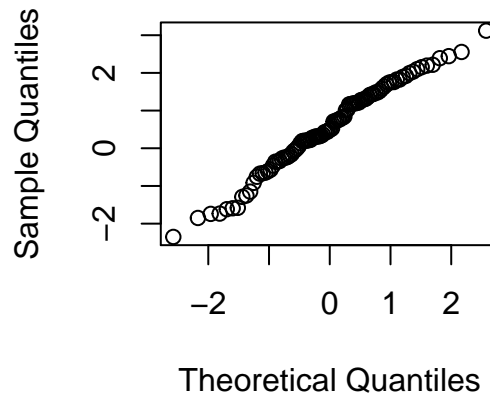


```
hist(x3); qqnorm(x3)
```

Histogram of x3

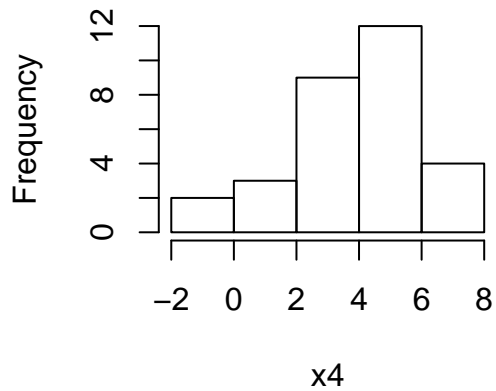


Normal Q-Q Plot

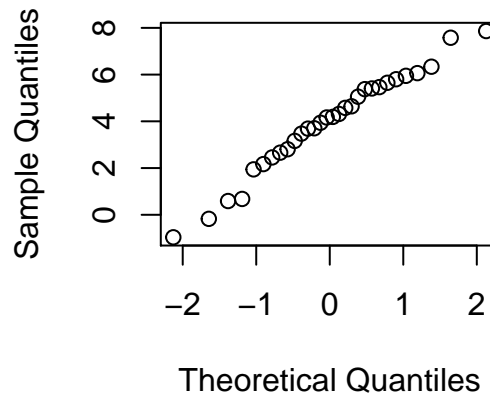


```
hist(x4); qqnorm(x4)
```

Histogram of x4

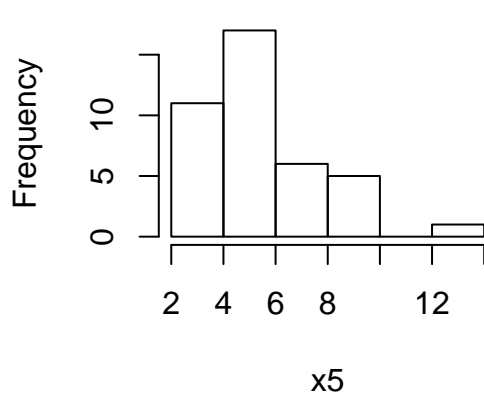


Normal Q-Q Plot



```
hist(x5); qqnorm(x5)
```

Histogram of x5



Normal Q-Q Plot

