

# STA258H5

Al Nosedal  
and Alison Weir

Winter 2017

# INTRODUCING R AND ASSESSING NORMALITY

# History of R

- S: language for data analysis developed at Bell Labs circa 1976
- Licensed by AT&T/Lucent to Insightful Corp. Product name: S-plus.
- R: initially written and released as an open source software by Ross Ihaka and Robert Gentleman at U Auckland during 90s (R plays on name S)
- Since 1997: international R-core team ~15 people & 1000s of code writers and statisticians happy to share their libraries! AWESOME!

# But it's open source...

Doesn't that mean it's lousy? NO!!

- Provides hundreds of thousands of functions and algorithms
- Lets users fix bugs and add functionality
- It is CUTTING EDGE, statistically and every other way.
- Ensures that researchers around the world - not just ones in rich countries - are the co-owners of the software tools needed to carry out research
- Most of R is written in? R! This makes it quite easy to see what functions are actually doing.

# What exactly is R?

R is used for data manipulation, statistics, and graphics.

It is made of:

- operators ( $+$   $-$   $<$   $-$   $?$ ) for calculations on vectors, arrays and matrices
- a huge collection of functions
- facilities for making unlimited types quality graphs
- user contributed packages (sets of related functions)
- the ability to interface with procedures written in C, C+, or FORTRAN and to write additional primitives.

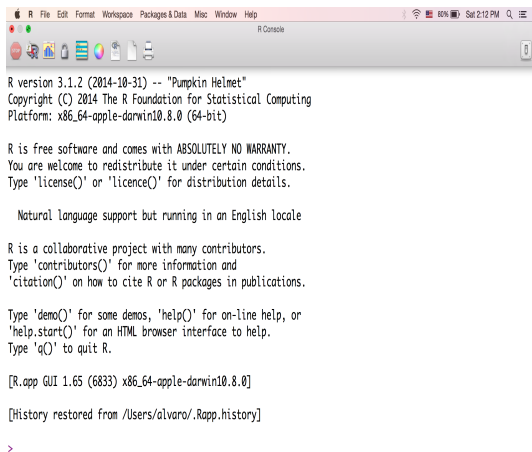
# Advantages of R

- Fast and free.
- State of the art statistically
- Only MATLAB has better graphics.
- Lots of other programs use R (like Google).
- Active user community
- Excellent for simulation, programming, analysis ?
- Forces you to think about your analysis.
- Easy interface with database storage software

# Disadvantages of R

- Not user friendly, minimal GUI.
- Steep initial learning curve
- No commercial support
- Easy to make mistakes and not know.
- Data prep and cleaning can be messy

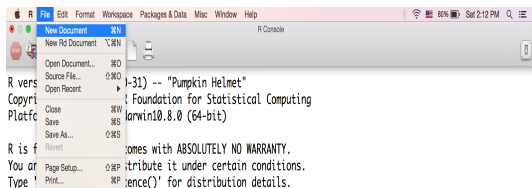
# The R GUI

A screenshot of the R GUI window on a Mac. The window title is "R Console". The menu bar includes "R", "File", "Edit", "Format", "Workspace", "Packages & Data", "Misc", "Window", and "Help". The system status bar shows "Sat 2:12 PM" and "80%" battery. The console output is as follows:

```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"  
Copyright (C) 2014 The R Foundation for Statistical Computing  
Platform: x86_64-apple-darwin10.8.0 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
[R.app GUI 1.65 (6833) x86_64-apple-darwin10.8.0]  
  
[History restored from /Users/alvaro/.Rapp.history]  
  
>
```



# Opening a script in R



Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

```
[R.app GUI 1.65 (6833) x86_64-apple-darwin10.8.0]
```

```
[History restored from /Users/alvaro/.Rapp.history]
```

```
> |
```

# Homework

This course uses R. R is an open-source computing package which has seen a huge growth in popularity in the last few years. R can be downloaded from <https://cran.r-project.org>

**Please, download R and bring your laptop next time.**

# What is RStudio?

RStudio is a relatively new editor specially targeted at R. RStudio is cross-platform, free and open-source software.

# More homework: Obtaining RStudio

Just go to:

<http://www.rstudio.com>

download the corresponding file, execute it locally and follow the instructions given by the installer.

The screenshot displays the R Studio interface with the following components:

- Script Editor:** Contains the following R code:

```
1 x=seq(1,10,by=1)
2 y=x
3 plot(x,y)
4 plot(x,y,col="blue")
5
6
```
- Console:** Shows the execution of the code and introductory text:

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=seq(1,10,by=1)
> y=x
> plot(x,y)
> plot(x,y,col="blue")
>
```
- Environment:** Shows the global environment with the following values:

Variable	Value
x	run [1:10] 1 2 3 4 5 6 7 8 9 10
y	run [1:10] 1 2 3 4 5 6 7 8 9 10
- Plots:** Displays a scatter plot of y versus x. The x-axis ranges from 0 to 10, and the y-axis ranges from 0 to 10. The data points are blue circles forming a straight line from (1,1) to (10,10).

# Getting your data into R Studio

- 1 Save data file as a *something.csv* file.
- 2 Click on Import Dataset in top right window

# Working in R Studio

Enter your commands in the console window

Arithmetic Operations

```
2+3;
```

```
## [1] 5
```

```
3-2;
```

```
## [1] 1
```

```
3*2;
```

```
## [1] 6
```

# Working in R Studio

Enter your commands in the console window  
Arithmetic Operations

```
3/2;
```

```
## [1] 1.5
```

```
3^2;
```

```
## [1] 9
```



Enter your commands in the console window  
Assignment

- To assign a value to a variable use `<-`

Example:

```
a<-19;
```

# How to use help in R?

If you know which function you want help with simply use **help**. Example:

```
help(histogram);
```

## For now: R-Fiddle

R-Fiddle is a programming environment for R available online. It allows us to encode and to run a program written in R. The tool is available at this URL: <http://www.r-fiddle.org>

# Old Faithful Geyser Data

## Description

Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.  
(A data frame with 272 observations on 2 variables.)

# R Fiddle

The screenshot shows a web browser window with the address bar at `r-fiddle.org`. The page title is "R-Fiddle" and there is a "Save" button. A notification says "Install the R-Fiddle Chrome App" with social media icons for Facebook, Twitter, and a help icon. The main content area contains a single line of R code: `1 dero("graphics")`. At the bottom right of the editor area are two buttons: "Graphs" and "Run Code". Below the editor is a grey console area with a prompt character `>` and a cursor.

# R Fiddle



The screenshot shows the R-Fiddle web application interface. At the top, there is a navigation bar with the R-Fiddle logo, a "Save" button, and a prompt to "Install the R-Fiddle Chrome App". Social media icons for Facebook, email, help, and chat are also present. Below the navigation bar, the code editor contains the R code: `1 demo("graphics")`. To the right of the code editor is a Facebook "Like" button showing 563 likes. Below the code editor are two buttons: "Graphs" and "Run Code". The console output shows the result of the `names(faithful)` command: `[1] "eruptions" "waiting"`.

# R Fiddle



The screenshot shows the R-Fiddle web application interface. At the top, there is a navigation bar with the R-Fiddle logo, a "Save" button, and a prompt to "Install the R-Fiddle Chrome App" with social media icons for Facebook, email, help, and chat. Below the navigation bar, the code editor contains the following R code:

```
1 demo("graphics")
```

To the right of the code editor, there is a Facebook "Like" button and a counter showing "563". Below the code editor, there are two buttons: "Graphs" and "Run Code". The console output area shows the following R commands and their results:

```
> names(faithful)
[1] "eruptions" "waiting"
> faithful$eruptions
```



The screenshot shows the R-Fiddle web application interface. At the top, there is a navigation bar with the R-Fiddle logo, a 'Save' button, and a prompt to 'Install the R-Fiddle Chrome App'. Below the navigation bar, the code editor contains the R command `demo("graphics")`. To the right of the code editor is a 'Like' button with a count of 563. Below the code editor are two buttons: 'Graphs' and 'Run Code'. The output area below the buttons displays the following text:

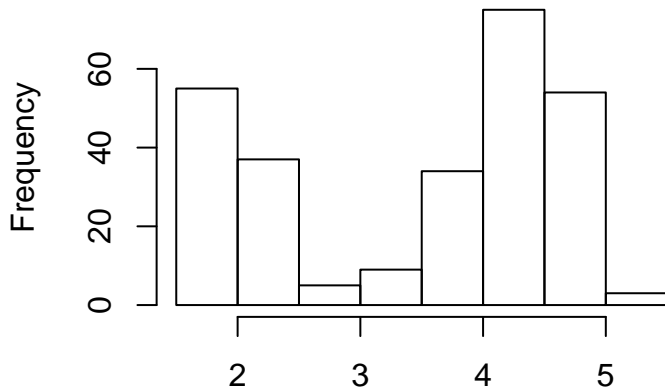
```
4.235  
[217] 2.400 4.800 2.000 4.150 1.867 4.267 1.750 4.483 4.000 4.117 4.083  
4.267  
[229] 3.917 4.550 4.083 2.417 4.183 2.217 4.450 1.883 1.850 4.283 3.950  
2.333  
[241] 4.150 2.350 4.933 2.900 4.583 3.833 2.083 4.367 2.133 4.350 2.200  
4.450  
[253] 3.567 4.500 4.150 3.817 3.917 4.450 2.000 4.283 4.767 4.533 1.850  
4.250  
[265] 1.983 2.250 4.750 4.117 2.150 4.417 1.817 4.467  
> |
```



# Histogram

```
## Basic plot.  
  
hist(faithful$eruptions);
```

## Histogram of faithful\$eruptions

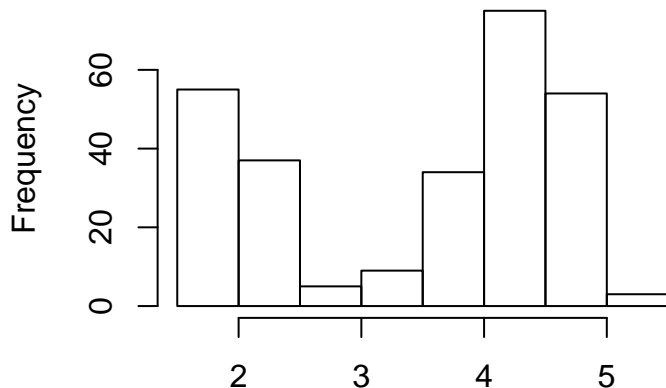


faithful\$eruptions

# Histogram (with title)

```
## Nicer plot.  
  
hist(faithful$eruptions,  
main="Duration of Old Faithful Eruptions (min)");
```

## Duration of Old Faithful Eruptions (min)

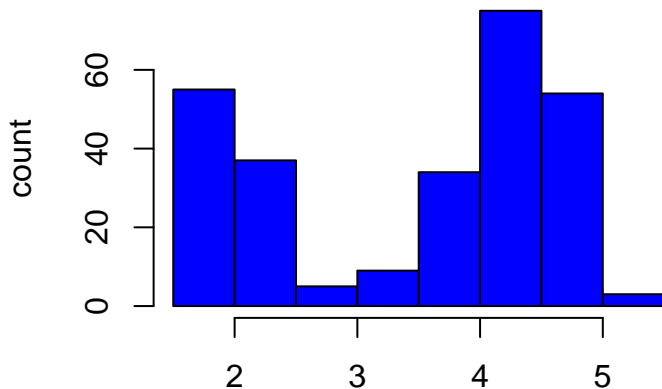


# Histogram

```
## Add axes labels and color.
```

```
hist(faithful$eruptions,  
main="Duration of Old Faithful Eruptions (min)",  
xlab="duration",ylab="count", col="blue");
```

## Duration of Old Faithful Eruptions (min)

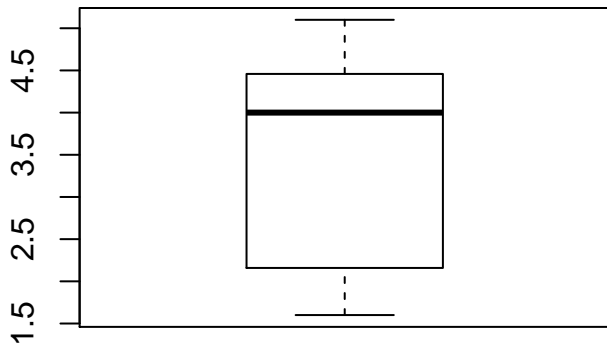


duration

# Boxplot

```
## Basic plot.  
  
boxplot(faithful$eruptions);
```

# Boxplot

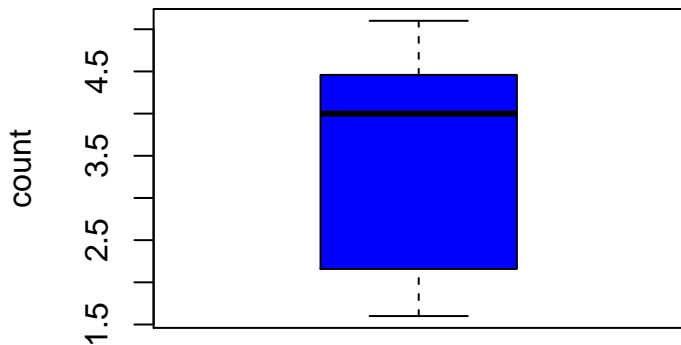




# Boxplot

```
## Add axes labels and color.  
  
boxplot(faithful$eruptions,  
main="Duration of Old Faithful Eruptions (min)",  
xlab="duration",ylab="count", col="blue");
```

## Duration of Old Faithful Eruptions (min)



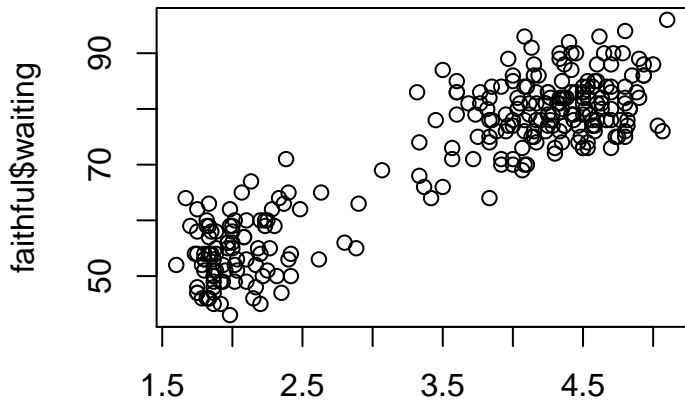
duration

# Scatterplot

```
## Basic plot.
```

```
plot(faithful$eruptions,faithful$waiting);
```

# Scatterplot



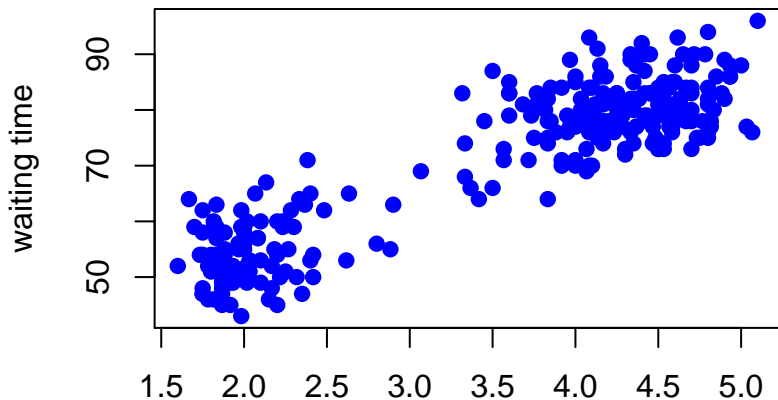
faithful\$eruptions

# Scatterplot

```
## Nicer plot.
```

```
plot(faithful$eruptions,faithful$waiting,  
main="Eruption Duration vs Waiting Times (mins)",  
xlab="duration",ylab="waiting time",  
pch=19, col="blue");
```

## Eruption Duration vs Waiting Times (mins)



# Making a panel of graphs

If you want more than one graph in a panel.

```
par(mfrow=c(nrow,ncol ) )
```

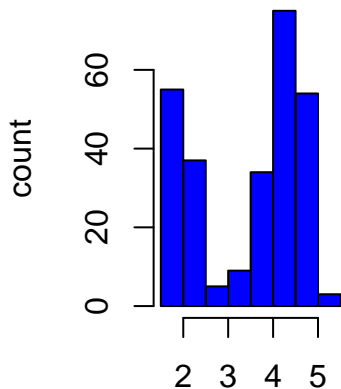
```
# where nrow= number of rows  
# and ncol=number of columns;
```

# Panel of graphs

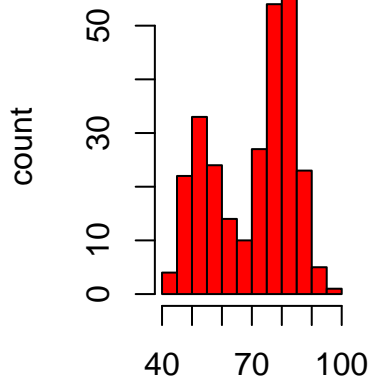
```
par(mfrow=c(1,2) )  
  
hist(faithful$eruptions,  
main="Duration (min)",  
xlab="duration",ylab="count", col="blue");  
  
hist(faithful$waiting,  
main="Waiting (min)",  
xlab="waiting time",ylab="count", col="red");
```



## Duration (min)



## Waiting (min)



# Panel of graphs

```
prototype<-rnorm(1000,mean=0,sd=1);

par(mfrow=c(2,2) )

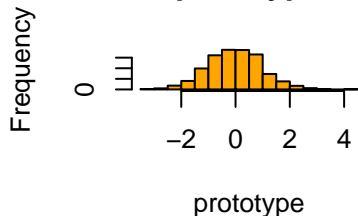
hist(prototype,
main="prototype",
col="orange");

hist(faithful$eruptions,
main="Duration (min)",
xlab="duration",ylab="count", col="blue");

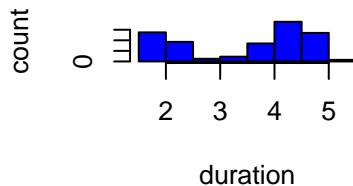
hist(faithful$waiting,
main="Waiting (min)",
xlab="waiting time",ylab="count", col="red");
```

# Panel of graphs

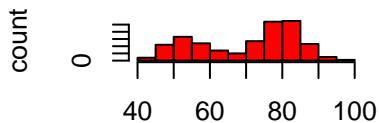
## prototype



## Duration (min)



## Waiting (min)



# Problem

How much do people with a bachelor's degree (but no higher degree) earn? Here are the incomes of 15 such people, chosen at random by the Census Bureau in March 2002 and asked how much they earned in 2001. Most people reported their incomes to the nearest thousand dollars, so we have rounded their responses to thousands of dollars: 110 25 50 50 55 30 35 30 4 32 50 30 32 74 60.

How could we find the "typical" income for people with a bachelor's degree (but no higher degree)?

# Measuring center: the median

The **median**  $M$  is the **midpoint** of a distribution, the number such that half the observations are smaller and the other half are larger. To find the median of the distribution:

Arrange all observations in order of size, from smallest to largest.

If the number of observations  $n$  is odd, the median  $M$  is the center observation in the ordered list. Find the location of the median by counting  $\frac{n+1}{2}$  observations up from the bottom of the list.

If the number of observations  $n$  is even, the median  $M$  is the mean of the two center observations in the ordered list. Find the location of the median by counting  $\frac{n+1}{2}$  observations up from the bottom of the list.

# Income Problem (Median)

We know that if we want to find the median,  $M$ , we have to order our observations from smallest to largest: 4 25 30 30 30 32 32 35 50 50 50 55 60 74 110. Let's find the location of  $M$

$$\text{location of } M = \frac{n+1}{2} = \frac{15+1}{2} = 8$$

Therefore,  $M = x_8 = 35$  ( $x_8 = 8$ th observation on our ordered list).

# The quartiles $Q_1$ and $Q_3$

To calculate the quartiles:

Arrange the observations in increasing order and locate the median  $M$  in the ordered list of observations.

The first quartile  $Q_1$  is the median of the observations whose position in the ordered list is to the left of the location of the overall median.

The third quartile  $Q_3$  is the median of the observations whose position in the ordered list is to the right of the location of the overall median.

# Income Problem ( $Q_1$ )

Data:

4 25 30 30 30 32 32 35 50 50 50 55 60 74 110.

From previous work, we know that  $M = x_8 = 35$ .

This implies that the first half of our data has  $n_1 = 7$  observations. Let us find the location of  $Q_1$ :

$$\text{location of } Q_1 = \frac{n_1+1}{2} = \frac{7+1}{2} = 4.$$

This means that  $Q_1 = x_4 = 30$ .



# Income Problem ( $Q_3$ )

Data:

4 25 30 30 30 32 32 35 50 50 50 55 60 74 110.

From previous work, we know that  $M = x_8 = 35$ .

This implies that the first half of our data has  $n_2 = 7$  observations. Let us find the location of  $Q_3$ :

$$\text{location of } Q_3 = \frac{n_2+1}{2} = \frac{7+1}{2} = 4.$$

This means that  $Q_3 = 55$ .

# Five-number summary

The five-number summary of a distribution consists of the smallest observation, the first quartile, the median, the third quartile, and the largest observation, written in order from smallest to largest. In symbols, the five-number summary is

*min*  $Q_1$   $M$   $Q_3$  *MAX*.

# Income Problem (five-number summary)

Data: 4 25 30 30 30 32 32 35 50 50 50 55 60 74 110. The five-number summary for our income problem is given by:  
4 30 35 55 110

```
# Step 1. Entering Data;
```

```
income=c(4,25,30,30,30,32,32,35,50,50,50,55,60,74,110);
```

```
# Step 2. Finding five-number summary;
```

```
fivenum(income);
```

```
## [1] 4.0 30.0 35.0 52.5 110.0
```

Note. Sometimes, R will give you a slightly different five-number summary.

# Box plot

A boxplot is a graph of the five-number summary.

A central box spans the quartiles  $Q_1$  and  $Q_3$ .

A line in the box marks the median  $M$ .

Lines extended from the box out to the smallest and largest observations.

# Boxplot

```
par(mfrow=c(1,3) )

boxplot(prototype,
main="prototype",
col="orange");

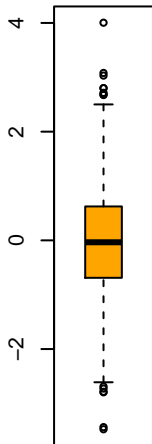
boxplot(faithful$eruptions,
main="eruption duration ",
col="blue");

boxplot(faithful$waiting,
main="time between eruptions",
col="red");
```

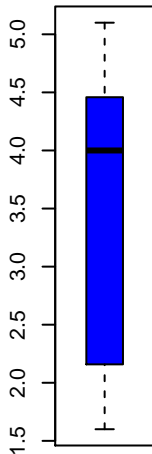


# Boxplot

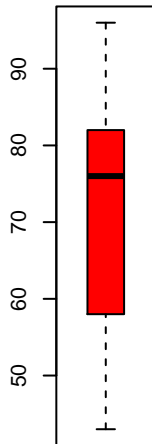
prototype

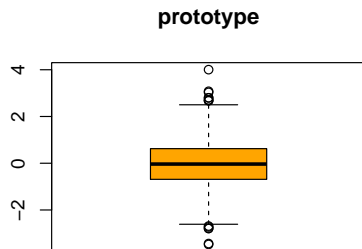


eruption duration



time between eruption





Should be:

- Symmetric
- $\approx 5\%$  outliers
- Tails  $\approx 1.5$  IQR

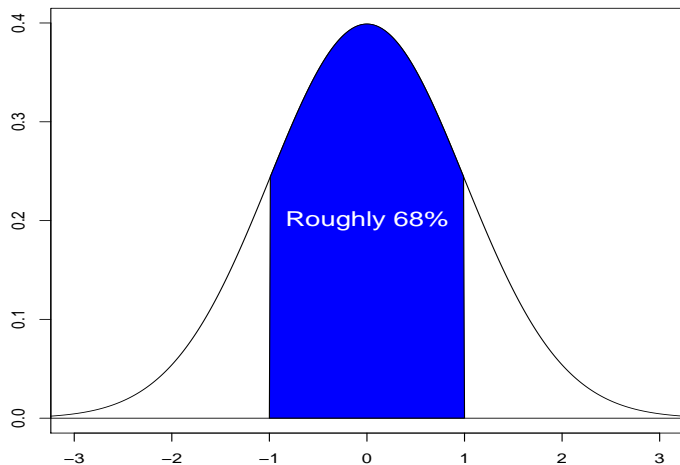
# The 68-95-99.7 rule

In the Normal distribution with mean  $\mu$  and standard deviation  $\sigma$ :

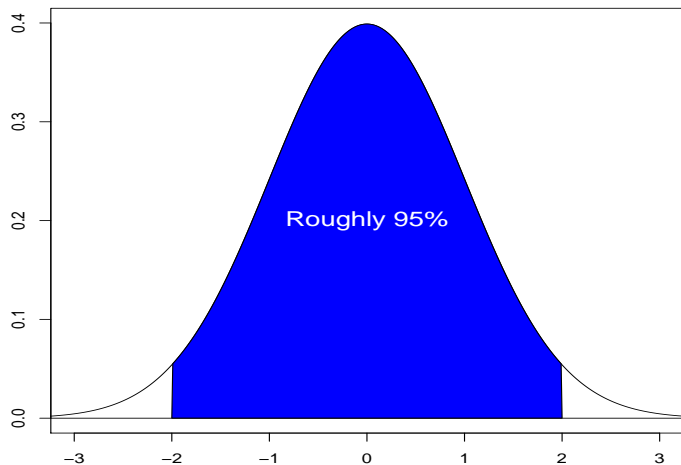
- Approximately 68% of the observations fall within  $\sigma$  of the mean  $\mu$ .
- Approximately 95% of the observations fall within  $2\sigma$  of  $\mu$ .
- Approximately 99.7% of the observations fall within  $3\sigma$  of  $\mu$ .

Note. The 68-95-99.7 rule is also known as the empirical rule.

# Example $N(\mu = 0, \sigma = 1)$



# Example $N(\mu = 0, \sigma = 1)$



```
meanP<-mean(prototype);  
sdP<-sqrt(var(prototype));  
  
lower<-meanP-sdP;  
upper<-meanP+sdP;  
  
N<-length(prototype);  
  
100*length(prototype[ lower<prototype & prototype<upper])/N;  
  
## [1] 69.8
```

```
meanP<-mean(prototype);  
sdP<-sqrt(var(prototype));  
  
lower<-meanP-2*sdP;  
upper<-meanP+2*sdP;  
  
N<-length(prototype);  
  
100*length(prototype[ lower<prototype & prototype<upper])/N;  
  
## [1] 94.9
```

```
meanP<-mean(prototype);  
sdP<-sqrt(var(prototype));  
  
lower<-meanP-3*sdP;  
upper<-meanP+3*sdP;  
  
N<-length(prototype);  
  
100*length(prototype[ lower<prototype & prototype<upper])/N;  
  
## [1] 99.5
```



```
eruptions<-faithful$eruptions;

meanE<-mean(eruptions);

sdE<-sqrt(var(eruptions));

lower<-meanE-sdE;

upper<-meanE+sdE;

N<-length(eruptions);

100*length(eruptions[ lower<eruptions & eruptions<upper])/N;

## [1] 55.14706
```

```
eruptions<-faithful$eruptions;

meanE<-mean(eruptions);

sdE<-sqrt(var(eruptions));

lower<-meanE-2*sdE;

upper<-meanE+2*sdE;

N<-length(eruptions);

100*length(eruptions[ lower<eruptions & eruptions<upper])/N;

## [1] 100
```

```
eruptions<-faithful$eruptions;

meanE<-mean(eruptions);

sdE<-sqrt(var(eruptions));

lower<-meanE-3*sdE;

upper<-meanE+3*sdE;

N<-length(eruptions);

100*length(eruptions[ lower<eruptions & eruptions<upper])/N;

## [1] 100
```

# Homework?

Modify R Code given above and see what happens with waiting time.

## Q-Q Plot (Example)

A sample of  $n = 10$  observations gives the values in the following table:

<i>Ordered Observations</i> $x_{(j)}$	<i>Probability levels</i> $(j - 1/2)/n$	<i>Standard Normal</i> <i>Quantiles</i> $q_{(j)}$
-1	0.05	-1.645
-0.10	0.15	-1.036
0.16	0.25	-0.674
0.41	0.35	-0.385
0.62	0.45	-0.125
0.80	0.55	0.125
1.26	0.65	0.385
1.54	0.75	0.674
1.71	0.85	1.036
2.30	0.95	1.645

Here, for example,  $P[Z \leq 0.385] = \int_{-\infty}^{0.385} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = 0.65$ .

## Q-Q Plot (Example)

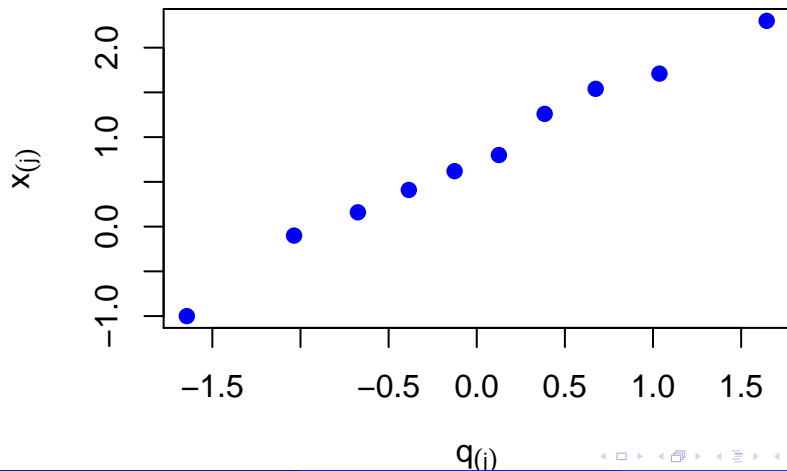
Let us now construct the Q-Q plot and comment on its appearance. The Q-Q plot for the foregoing data, which is a plot of the ordered data  $x_{(j)}$  against the normal quantiles is shown below. The pairs of points  $(q_{(j)}, x_{(j)})$  lie very nearly along a straight line, and we would not reject the notion that these data are Normally distributed-particularly with a sample size as small as  $n = 10$ .

```
## Ordered observations;  
  
obs<-c(-1,-0.1,0.16,0.41,0.62,0.80,1.26,1.54,1.71,2.30);  
  
n<-length(obs);  
  
## Corresponding probability values;  
  
prob.levels<-(seq(1:n)-0.5)/n;  
  
## Standard Normal Quantiles;  
  
norm.quantiles<-qnorm(prob.levels);
```

```
## Q-Q plot;  
  
plot(norm.quantiles,obs,  
xlab=expression(q[(j)]),  
ylab=expression(x[(j)]),  
main="Ours",col="blue",pch=19);  
  
## Q-Q plot (using R function);  
  
qqnorm(obs,col="blue",pch=19);
```

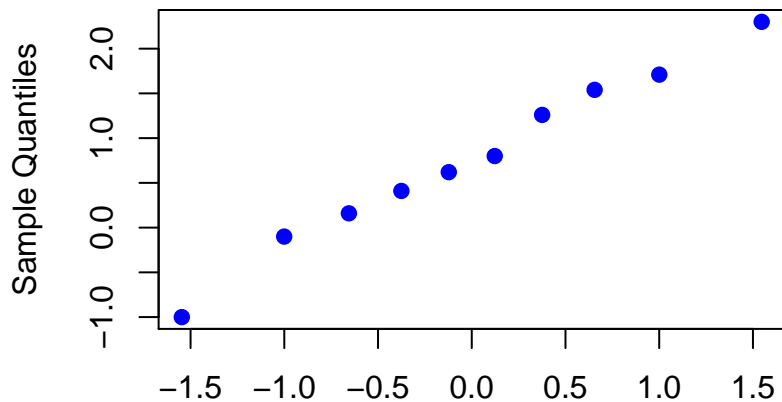


## Ours



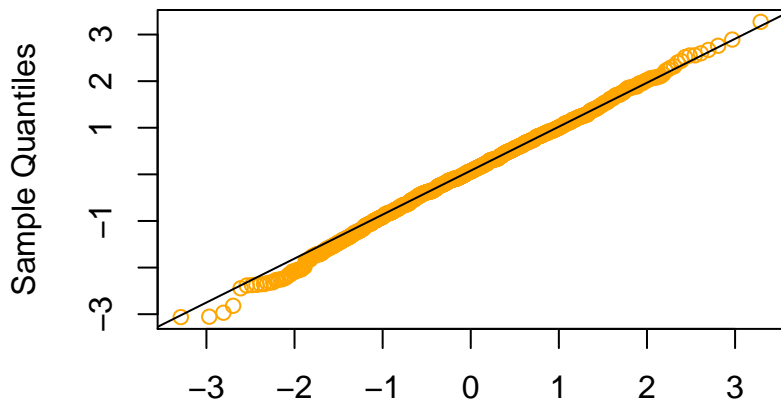
```
## Q-Q plot (using R function);  
qqnorm(obs,col="blue",pch=19);
```

## Normal Q-Q Plot



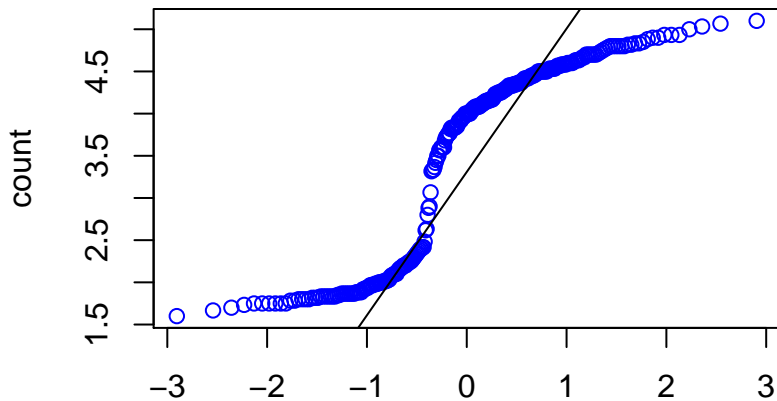
Theoretical Quantiles

## prototype

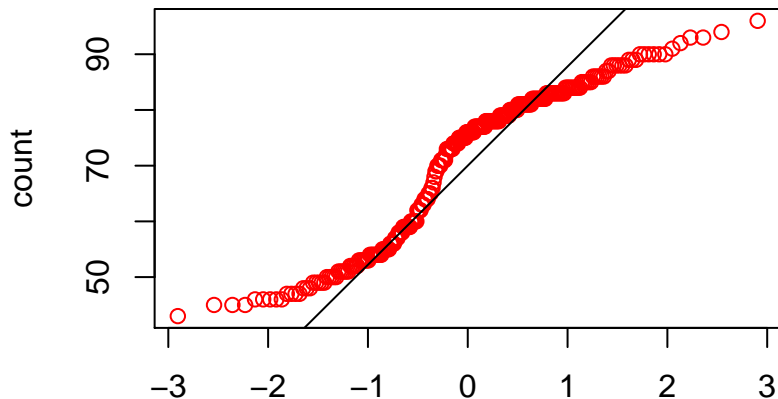


Theoretical Quantiles

## Duration (min)

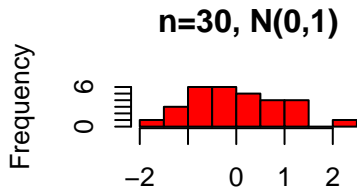
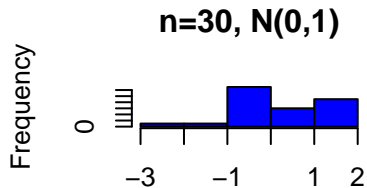
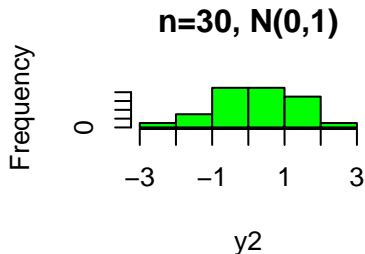
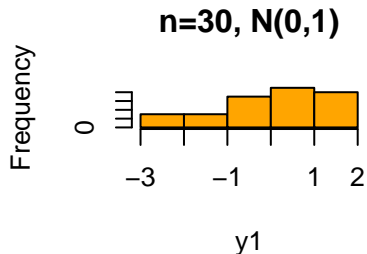


## Waiting (min)



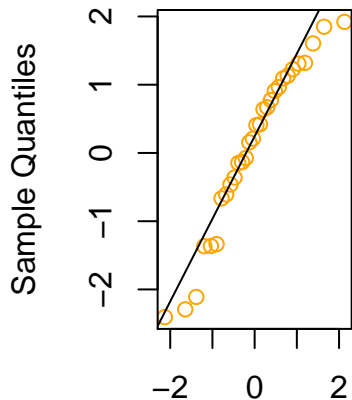
- If plots are OK  $\rightarrow$  data could come from a Normal distribution . . . but it could come from some other distribution. So good plots don't prove data came from a Normal distribution
- If plots are not OK  $\rightarrow$  data probably does not come from a Normal distribution, we can't assume data is from Normal population

# Panel of graphs

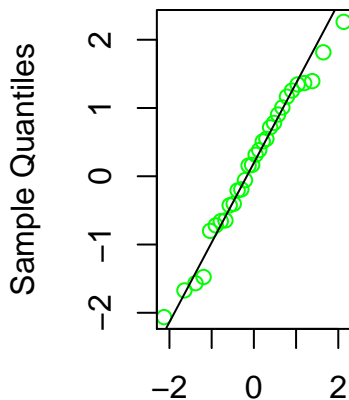




**n=30, N(0,1)**

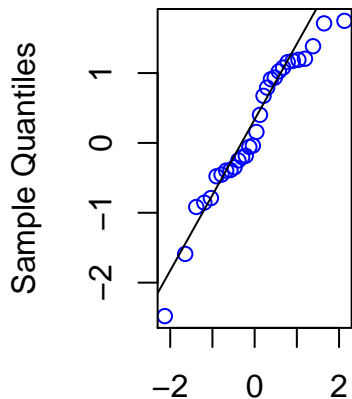
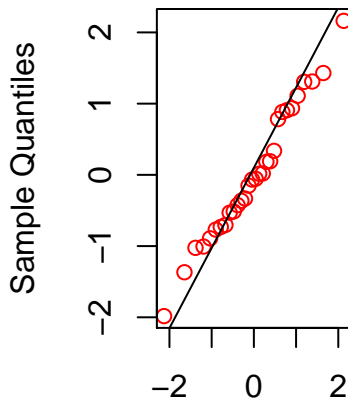


**n=30, N(0,1)**



Theoretical Quantiles

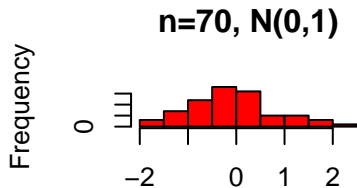
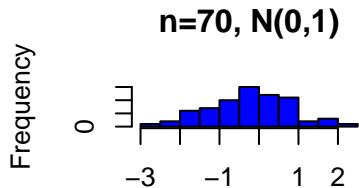
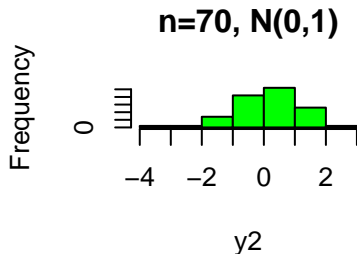
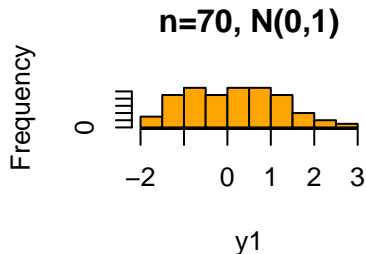
Theoretical Quantiles

$n=30, N(0,1)$  $n=30, N(0,1)$ 

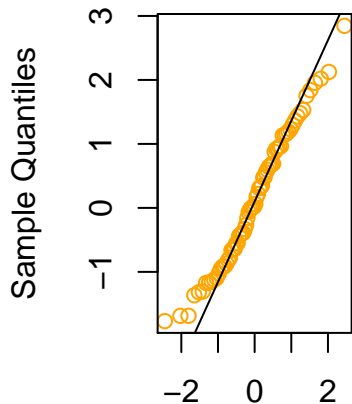
Theoretical Quantiles

Theoretical Quantiles

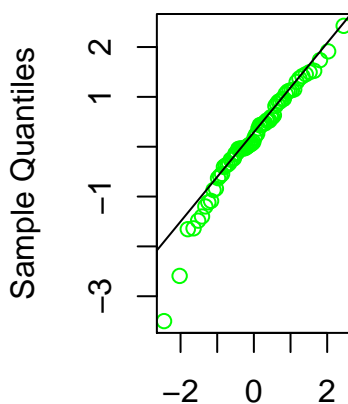
# Panel of graphs



**n=70, N(0,1)**



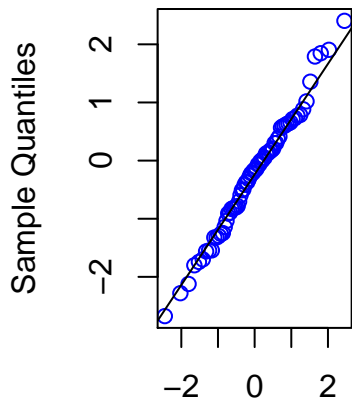
**n=70, N(0,1)**



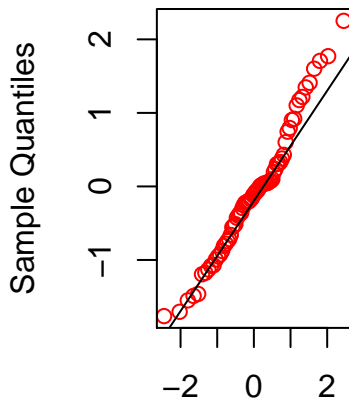
Theoretical Quantiles

Theoretical Quantiles

**n=70, N(0,1)**



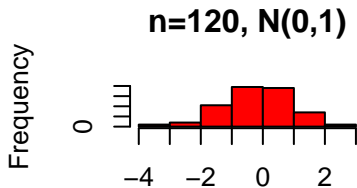
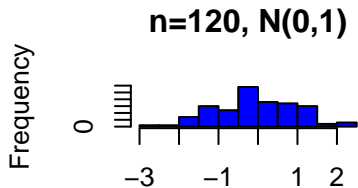
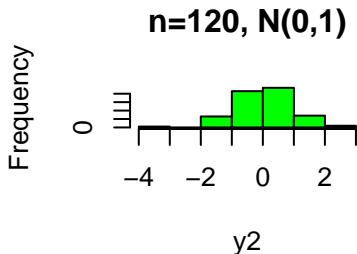
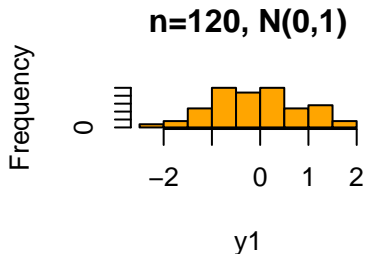
**n=70, N(0,1)**



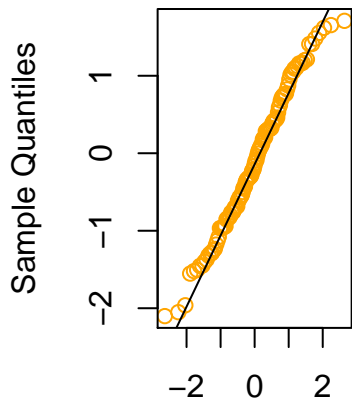
Theoretical Quantiles

Theoretical Quantiles

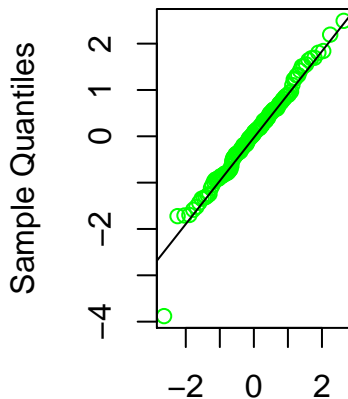
# Panel of graphs



**n=120, N(0,1)**



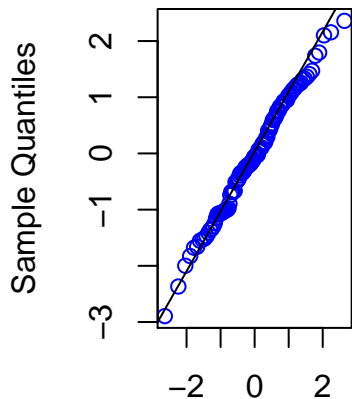
**n=120, N(0,1)**



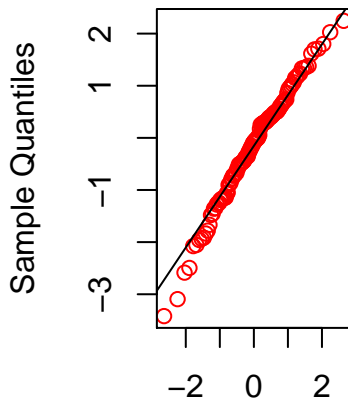
Theoretical Quantiles

Theoretical Quantiles

**n=120, N(0,1)**



**n=120, N(0,1)**



Theoretical Quantiles

Theoretical Quantiles