Data Visualization

Exercise: Business Intelligence (Part 3)
Summer Term 2014
Stefan Feuerriegel

Today's Lecture

Objectives

1. Calculating descriptive statistics in order to understand datasets
2. Visualizing data in R graphically
3. Choosing appropriate plots in a given context

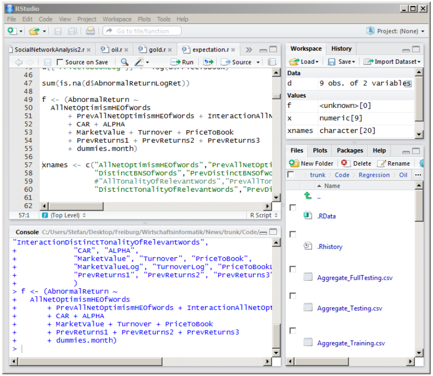
Outline

1. Recap: Introduction to R
2. Point Plot & Line Plot
3. Bar Plot & Pie Chart
4. Histogram & Boxplot
5. Excursus: Random Numbers & Normal Distribution
6. Q-Q Plot
7. Wrap-Up

Outline

R as a Statistical Software

- Free software environment aimed at statistical computing
- Supports many operating systems (Linux, Mac OS X, Windows)
- Based on commands



Retrieving R Studio (recommended)

Download at `http://www.rstudio.com/`

Operations, Functions and Variables

- Applying operators and evaluating functions

```
sqrt(-4 + 2 * 3)   # sqrt = square root

## [1] 1.414
```

- Storing values in variables and accessing them

```
x <- 2
x

## [1] 2
```

Vectors

- Creating vector by concatenation

```
x <- c(4, 0, 6)
```

- Output of first component

```
x[1]
## [1] 4
```

- Compute average value and standard deviation

```
mean(x)
## [1] 3.333
sd(x)
## [1] 3.055
```

- Generating arbitrary sequences (notation: from, to, step size)

```
seq(4, 5, 0.1)
##  [1] 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.0
```

Creating Matrices

1. Generating matrices by combining vectors

```
height <- c(163, 186, 172)
shoe_size <- c(39, 44, 41)
m <- as.data.frame(cbind(height, shoe_size))
```

2. By reading file (in CSV format) via

```
d <- as.data.frame(read.csv("persons.csv",
     header=TRUE, sep=","))
d

##     name height shoesize age
## 1  Julia    163       39  24
## 2  Robin    186       44  26
## 3  Kevin    172       41  21
## 4    Max    184       43  22
```

Accessing Matrices

- Access columns by name

```
d$height

## [1] 163 186 172 184
```

- Accessing individual elements (notation: #row, #column)

```
d[1, 2]

## [1] 163
```

- Selecting rows using a boolean condition

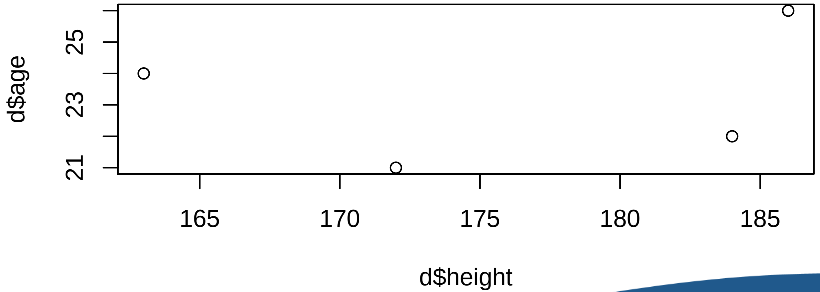
```
d[d$age > 25, ]

##    name height shoesize age
## 2 Robin    186       44  26
```

Outline

Point Plot

- Creating simple point plots (also named scatter plots) via `plot(...)`
- Relies upon vectors denoting the x-axis and y-axis locations
- Various options can be added to change appearance

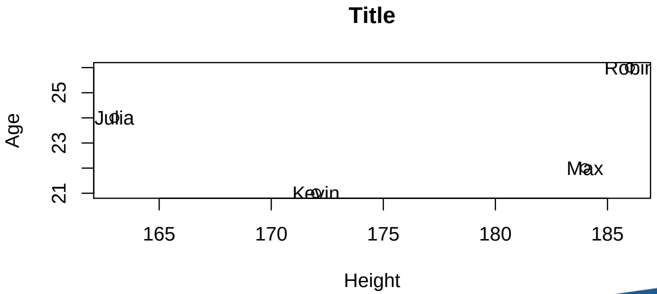
```
plot(d$height, d$age)
```



Adding Titles and Labels

- Titles are added through additional parameters (`main`, `xlab`, `ylab`)
- Labels are drawn next to given points with `text(...)`

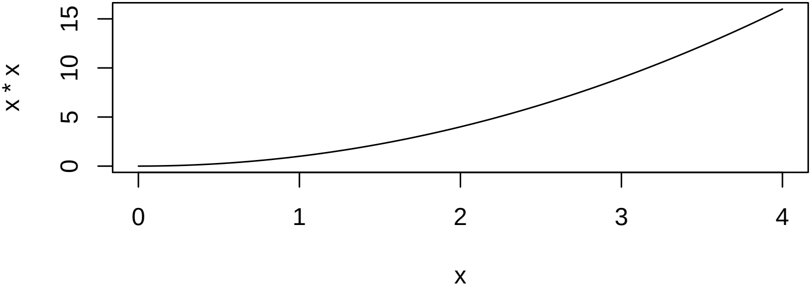
```
plot(d$height, d$age,
main="Title",                    # an overall title for the plot
xlab="Height", ylab="Age") # titles for x and y axis
text(d$height, d$age, d$name) # d$name are labels
```



Line Plot

Generate line plot using the additional option `type='l'`

```
x <- seq(0, 4, 0.01)
plot(x, x * x, type = "l")
```



Outline

Data Frequency

BI Case Study

Participants were asked, in a representative study, what the first day away from work was during their last illness

Question: Are you more likely to become sick on certain working days?

Example File: numberofstaffill.csv

```
"DAYOFWEEK"
"MON"
"THU"
"THU"
"THU"
...
```

Accessing Data

- Reading data

```
d <- as.data.frame(read.csv("numberofstaffill.csv",
                            sep=",", header=TRUE))
```

- Printing first rows of data

```
head(d)
```

```
##   DAYOFWEEK
## 1       MON
## 2       THU
## 3       THU
## 4       THU
## 5       TUE
## 6       MON
```

- Calculating number of observations

```
dim(d)
```

```
## [1] 300   1
```

```
obs <- dim(d)[1]  # 300 rows/observations
```

Data Frequency (Solution A)

- Count frequencies for each weekday

```r
mo <- length(d[d$DAYOFWEEK == "MON", ])
tu <- length(d[d$DAYOFWEEK == "TUE", ])
we <- length(d[d$DAYOFWEEK == "WED", ])
th <- length(d[d$DAYOFWEEK == "THU", ])
fr <- length(d[d$DAYOFWEEK == "FRI", ])
sa <- length(d[d$DAYOFWEEK == "SAT", ])
su <- length(d[d$DAYOFWEEK == "SUN", ])
```

- Print absolute and proportional frequencies → peak on mondays

```r
freq <- as.data.frame(cbind(mo, tu, we, th, fr, sa, su))
freq  # absolute frequencies

##   mo tu we th fr sa su
## 1 96 60 51 45 30  9  9

freq/obs  # proportional frequencies

##     mo  tu   we   th  fr   sa   su
## 1 0.32 0.2 0.17 0.15 0.1 0.03 0.03
```

Data Frequency (Solution B)

- Absolute frequencies via `table(...)`

```
table(d$DAYOFWEEK)

##
## FRI MON SAT SUN THU TUE WED
##  30  96   9   9  45  60  51
```

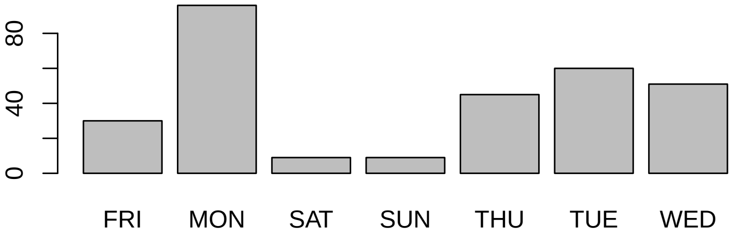
- Proportional occurrences by subsequent scaling

```
table(d$DAYOFWEEK)/obs

##
##  FRI  MON  SAT  SUN  THU  TUE  WED
## 0.10 0.32 0.03 0.03 0.15 0.20 0.17
```

Histogram

- `barplot(...)` creates a bar plot using given frequencies in `abs.freq`
- Useful for visualizing absolute frequencies of categories

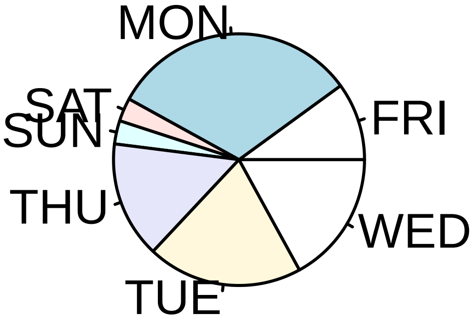
```
abs.freq <- table(d$DAYOFWEEK)
barplot(abs.freq)
```



Pie Chart

- `pie(...)` draws a pie chart using frequencies in `abs.freq`
- Useful for visualizing relative frequencies

```
abs.freq <- table(d$DAYOFWEEK)
pie(abs.freq)
```



Outline

Data Distribution

BI Case Study

In a study (Hornik et al., 2008), all court (VwGH) decisions between 2000 and 2004 were analyzed in terms of their length.

Question: What is the distribution of lawsuit durations?

Example File: court_decisions.csv

```
year,senate,senatesize,decision,durationrev,duration
2004,13,5,2,893,2738
2004,13,5,3,2738,1624
2004,13,5,3,2372,1624
2004,13,3,2,888,1282
...
```

- `duration` gives duration in days
- unknown data marked as "`-9999`" in `duration`

Accessing Data

- Reading data

```
decisions <- as.data.frame(read.csv("court_decisions.csv",
                           sep=",", header=TRUE))
```

- Filtering data to remove those with unknown lawsuit duration

```
d <- decisions[decisions$duration != -9999, ]
```

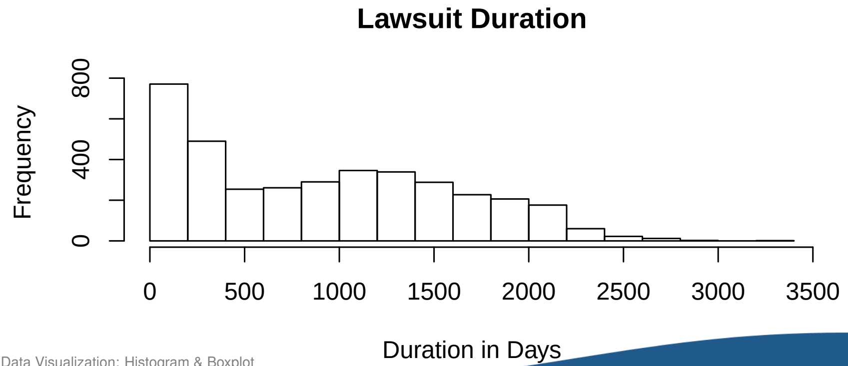
- Calculating dimensions of data

```
dim(d)

## [1] 3745    6
```

Histograms with Frequencies

- Histograms are a graphical representation of the distribution of data
- Created via `hist(data)` to get fixed width of classes
- y-axis gives frequency $\rightarrow$ estimating probability distribution

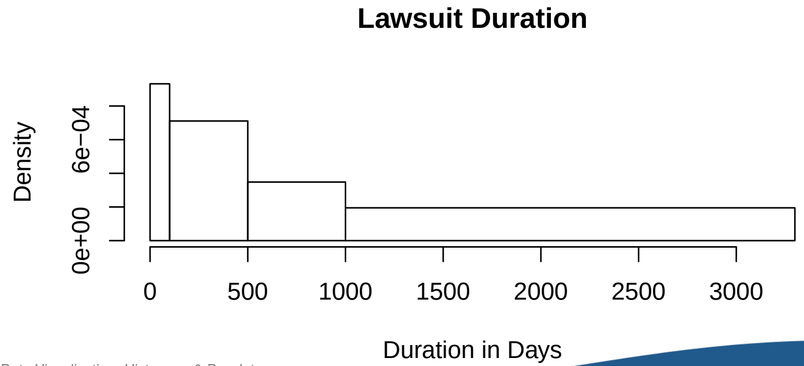
```
hist(d$duration,
main = "Lawsuit Duration", xlab = "Duration in Days")
```



Histograms with Densities

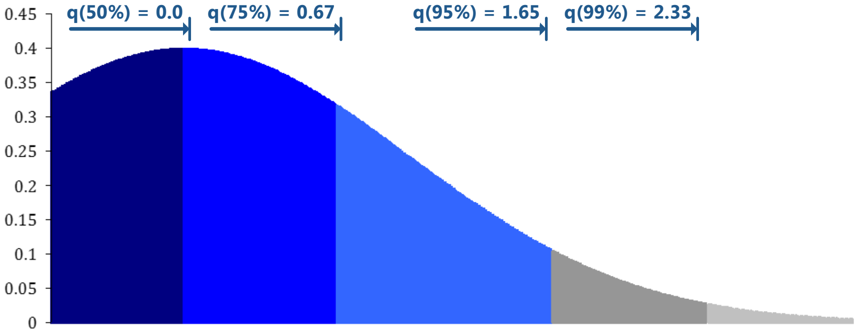
- Density (1.00 $\widehat{=}$ 100 %) on *y*-axis via `hist(data, freq=FALSE)`
- Parameter `breaks=b` gets a variable width of classes

```
b <- c(0, 100,500,1000,3300)
hist(d$duration, breaks = b,
main = "Lawsuit Duration", xlab = "Duration in Days")
```



Quantiles

- Quantiles are points taken at regular intervals from the cumulative distribution function (CDF) of a random variable
- p-percent quantile for a variable X is $\Pr[X < x] \leq q$
- 50 %-quantile named median; 25 %-quantiles called quartiles



Descriptive Statistics

- Minimum and maximum

```r
min(d$duration)
```

```
## [1] 2
```

```r
max(d$duration)
```

```
## [1] 3262
```

- Median (i. e. 50 %-quantile)

```r
median(d$duration)
```

```
## [1] 868
```

- Arbitrary *p*-percent quantiles

```r
# with p = 25%
quantile(d$duration, 0.25)
```

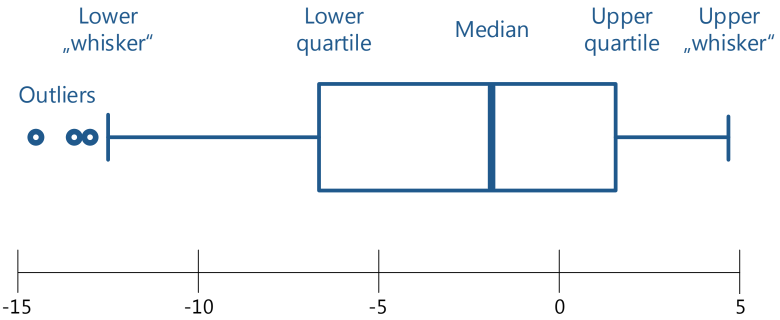
```
## 25%
## 258
```

- Combined descriptive statistics

```r
summary(d$duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2     258     868     915    1440    3260
```

Boxplot: Elements

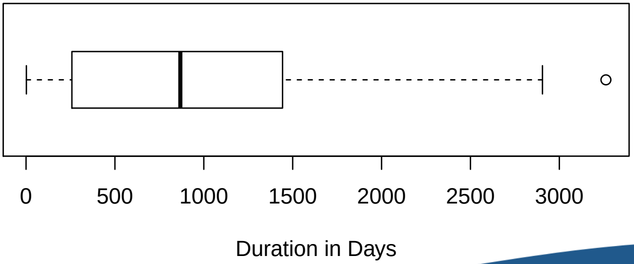


- Interquartile Range (IQR) is between first and third quartile
- 50 % of the data is in the IQR
- Lower/first quartile means the 25 % quantile
- Upper/third quartile means the 75 % quantile

Boxplot

- Use `boxplot(...)` to draw boxplot visualizing outliers (as circles), range and quartiles
- Default is vertical mode (`horizontal=FALSE`)

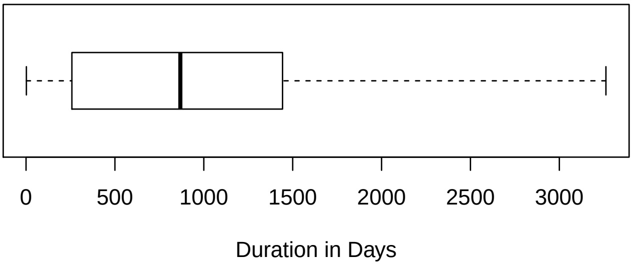
```
boxplot(d$duration, horizontal=TRUE,
xlab="Duration in Days")
```



Boxplot

- To prevent highlighting of outliers, use `range=0`

```
boxplot(d$duration, horizontal=TRUE,
xlab="Duration in Days", range=0)
```



Outline

Random Numbers from Uniform Distribution

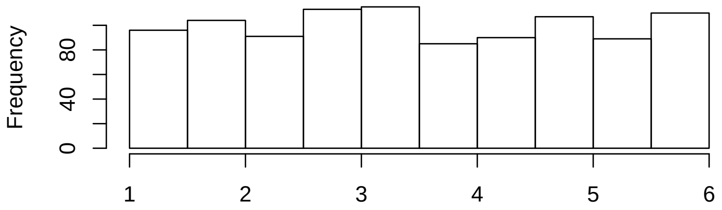
- In a uniform distribution, all floating-point numbers equally likely
- Generate `n` random numbers in range `min` to `max` via `runif(n, min, max)`

```
runif(1, 5, 7.5)  # generate 1 number between 5.0 and 7.5

## [1] 7.242
```

- Example

```
hist(runif(1000, 1, 6), xlab = "", main = "")
```



Random Numbers from Discrete Uniform Distribution

- Discrete uniform distribution considers only equally-likely integers
- Generate `n` random numbers via
  `sample(min:max, n, replace=TRUE)`

```
# generates 2 numbers from the set 1, ..., 10
sample(1:10, 2, replace = TRUE)

## [1] 9 3
```

- Example (e. g. rolling dice 1000 times)

```
table(sample(1:6, 1000, replace = TRUE))

##
##   1   2   3   4   5   6
## 167 152 200 145 167 169
```

Normal Distribution

Definition: Normal (or Gaussian) Distribution

- Defined by

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

  with mean μ and standard deviation σ

- Standard normal distribution: $\mu = 0$ and $\sigma = 1$; then its probability density function becomes

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

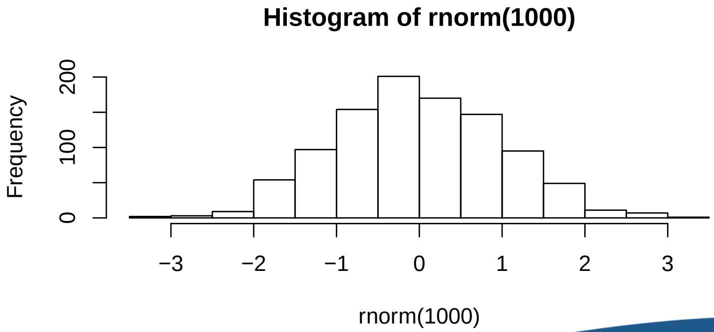
Random Numbers from a Normal Distribution

- Generate `n` random numbers from standard normal distribution ($\mu = 0$ and $\sigma = 1$) with `rnorm(n)`

```
rnorm(1)  # 1 number from the std. normal distribution
## [1] 1.263
```

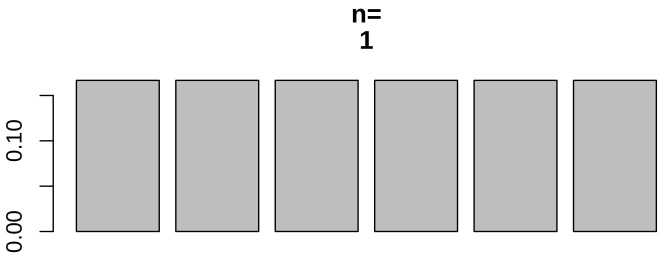
- Example (resembles density)

```
hist(rnorm(1000))
```



Normal Distribution: Example

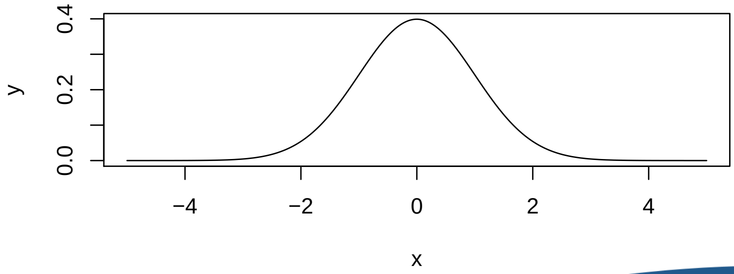
Sum of rolling *n* fair 6-sided dice converges to a shape of a normal distribution



Normal Distribution: Plotting

- Density of normal distribution with mean μ and standard deviation σ is computed by `dnorm(x, mean=`μ`, sigma=`σ`)`
- Plot shows probability density function of standard normal distribution

```
x <- seq(-5, 5, 0.01)
y <- dnorm(x, mean = 0, sd = 1)
plot(x, y, type = "l")  # visualize as line plot
```

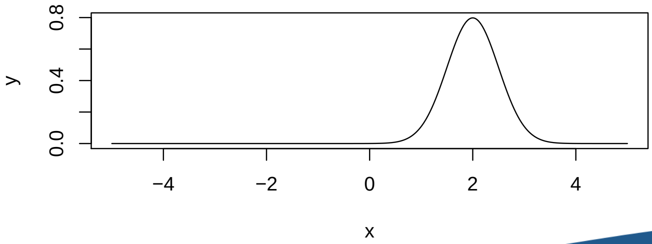


Normal Distribution: Plotting

Exercise

Plot the normal distribution with mean $\mu = 2$ and standard deviation $\sigma = 0.5$

```
x <- seq(-5, 5, 0.01)
y <- dnorm(x, mean = 2, sd = 0.5)
plot(x, y, type = "l")
```



Outline

Comparing Distributions

BI Case Study

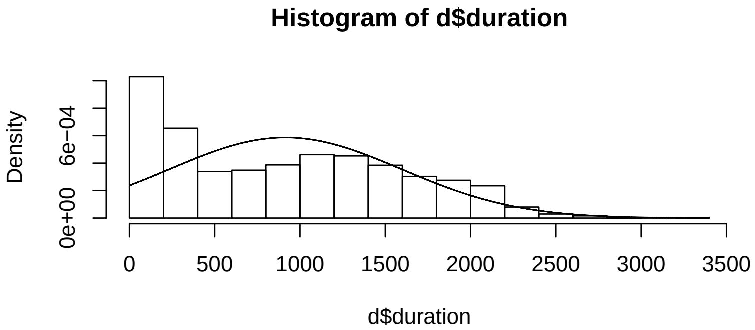
Is the duration of lawsuits normally distributed?

Solutions:

1. Histogram (also showing baseline distribution)
2. Q-Q plot

Comparing Distributions: Histogram

- Not recommended: Compare histogram and corresponding normal distribution by overlapping plot
- ```
  hist(d$duration, freq=FALSE)
  xx <- seq(min(d$duration), max(d$duration), 0.01)
  lines(xx, dnorm(xx, mean=mean(d$duration),
                      sd=sd(d$duration)))
  ```

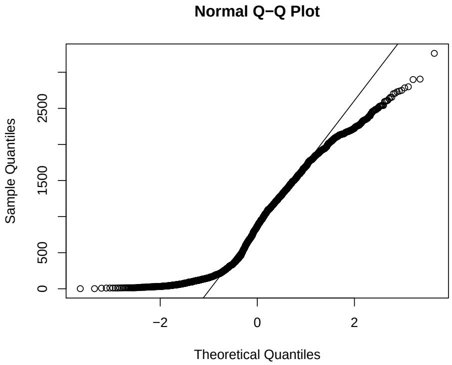

  ```

Q-Q Plot

- Q-Q plot ("Q" stands for quantile) compares two probability distributions by plotting their quantiles against each other
- `qqnorm(d)`, `qqline(d)` use standard normal distribution

```
# plot sample against
# theoretical standard
# normal distribution
qqnorm(d$duration)

# line that represents
# true normal distribution
qqline(d$duration)
```



$\rightarrow$ No standard normal distribution because of strong offset at tails

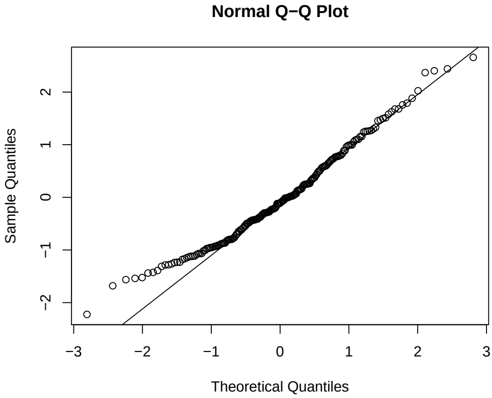
Q-Q Plot

Exercise

Verify that `rnorm(200)` is, in fact, normally distributed

```
x <- rnorm(200)
qqnorm(x)
qqline(x)
```

→ Strong linear pattern suggests standard normal distribution

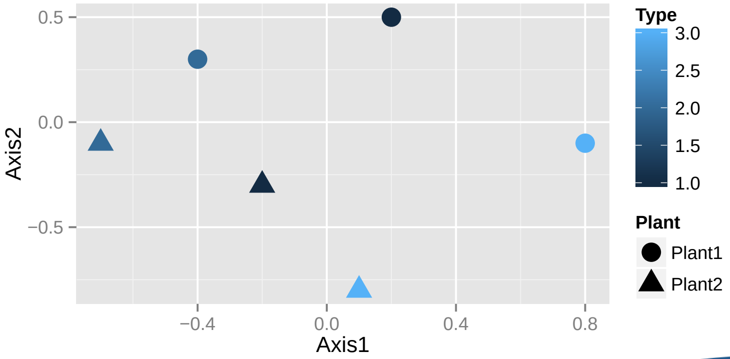


Outline

Fancy Diagrams with ggplot2

```r
library(ggplot2)
```

```r
df <- data.frame(Plant=c("Plant1", "Plant1", "Plant1", "Plant2", "Plant2", "Plant2"),
                 Type=c(1, 2, 3, 1, 2, 3),
                 Axis1=c(0.2, -0.4, 0.8, -0.2, -0.7, 0.1),
                 Axis2=c(0.5, 0.3, -0.1, -0.3, -0.1, -0.8))
ggplot(df, aes(x=Axis1, y=Axis2, shape=Plant,
               color=Type)) + geom_point(size=5)
```



Guideline to Choosing Plots

Data Structure	Plot	R Command
Relationship (2-dim.)	Point Plot	`plot(x, y)`
Evolving Time Series	Line Plot	`plot(x, y, type='l')`
Absolute Frequencies	Bar Plot	`barplot(freq)`
Proportions	Pie Chart	`pie(freq)`
Frequencies (Fixed Ranges)	Histogram	`hist(d)`
Densities (Variable Ranges)	Histogram	`hist(d, freq=FALSE, breaks=b)`
Distribution Variation	Boxplot	`boxplot(d)`
Distribution Comparison	Q-Q Plot	`qqnorm(d),qqline(d)`

Summary: Commands

Descriptive Statistics

`table(data)`	Absolute frequencies of categories
`median(data)`	Median value
`quantile(data, p)`	*p*-percent quantile
`summary(data)`	Descriptive statistics

Generating Random Numbers

`runif(n, min, max)`	from uniform distribution
`sample(from:to, n, replace=TRUE)`	from discrete uniform distribution
`rnorm(n)`	from normal distribution
dnorm(x, mean=μ, sigma=σ)	Density of standard normal distribution

Further Exercises

$\rightarrow$ available online as homework