Statistics Summary

Let's discover **Introductory Statistics**.

Most important information / Condensed Information.

R Functions Summary : Analysis of Scientific Data

This sheet gives a summary of the basic functions in R. You can also get more information using help() or ? in R.

Importing Data

- Use `data = read.csv("file")` for CSV data and `data = read.delim("file")` for tab-separated data.
- Use `View(data)`, `str(data)`, `names(data)`, `nrow(data)` or `head(data)` as checks that your data was imported correctly.
- Use `data$X`, for example, to extract the variable X from the data frame data.
- Use `data$Z = data$Y1 – data$Y2`, for example, to calculate a new variable.
- Use `subset(data, X == "x")` to extract a new data frame with just only the cases where X is x.
- Use `data$X = ordered(data$X, c(A,B,C))` to make X an ordinal variable with A < B < C, for example.

Lattice Graphics

- Use `library(lattice)` to load the lattice graphics library.

Strip plots	**`stripplot(data$Y)`; `stripplot(~ Y, data)`; `stripplot(Y ~ X, data)`**
Histogram	`histogram(data$Y)`; `histogram(~ Y, data)`
Density plot	`densityplot(data$Y)`; `densityplot(~ Y, data)`
Box plot	`bwplot(data$Y)`; `bwplot(~ Y, data)`; `bwplot(Y ~ X, data)`
Quantile plot	`qqmath(data$Y)`; `qqmath(~ Y, data)`
Scatter plot	`xyplot(Y ~ X, data)`; use type="p" for data points, "l" for joining with lines, "g" for a grid, "r" for a regression line, and "smooth" for a smoothing line.
Bar chart	`barchart(table(data$X, data$Y))`
Spine plot	`spineplot(table(data$X, data$Y))`

- Use the group=Z option to separate by variable Z. Add a title to your plot using the main="title" option. Change the axes labels using xlab and ylab. Get a simple legend with auto.key=TRUE.

Summary Statistics

- Get basic statistics with
  - `summary(data)`
  - `mean(data$Y)`
  - `median(data$Y)`
  - `sd(data$Y)`
  - `IQR(data$Y)`
  - `fivenum(data$Y)`
- For categorical data use:
  - `table(data$Y)`, or
  - `table(data$Y, data$X)` for a two-way table.
  - `prop.table()` can be applied to table() to get proportions and marginal proportions.
- Use aggregate() to get statistics by group.
  - For example, `aggregate(Y ~ X,data,mean)` gives the mean Y value for each category in X. This outputs a data frame which you can also use for plotting.

Basic Inference

Kind of Test	Statistical Test
One-sample t test	`t.test(data$X)` See power.t.test() for power calculations
Two-sample t test	`t.test(Y~X, data)`
One proportion	`prop.test(x, n)`
Two proportions	`prop.test(table(data$X, data$Y))`
Chi square test	`chisq.test(table(data$X, data$Y))`

Model Building

- For each of the following you can use the following functions for details of the analysis:
  - `summary()`
  - `anova()`
  - `predict()`
  - `residuals()`

Model	Command
Linear regression	`lm(Y ~ X, data)`
ANOVA	`aov(Y ~ X, data)`
Logistic regression	`glm(Y ~ X, data, family="binomial")`

- Use `Y ~ X1*X2` for a two-factor model with an interaction term or `Y ~ X1+X2` for a model without an interaction term.

Other Calculations

Calculations	Command
Logarithms	`log(x)` for natural logs and `log10(x)` for base 10 logs
Exponentials	`exp(x)` for e^x and 10^x for 10x
Factorial	`factorial(n)` gives n!
Combinations	`choose(n, k)`

Distributions

- Note that the following distribution functions all give areas to the left (matching the theoretical definitions of these functions) whereas the tables

in the textbook all give areas to the right (matching our use of the distributions).

Distribution	Command
Binomial	`dbinom(x,n,p)` for `P(X = x)`; `pbinom(x,n,p)` for `P(X ≤ x)`
Normal	`pnorm(z)` for `P(Z ≤ z)`; `qnorm(p)` for finding `z` such that `P(Z ≤ z) = p`
T	`pt(t,df)` for `P(Tdf ≤ t)`; `qt(p,df)` for finding `t` such that `P(Tdf ≤ t) = p`
Chi square	`pchisq(x,df)` for `P(Xdf ≤ x)`; `qchisq(p,df)` for finding `x` such that `P(Xdf ≤ x) = p`
F	`pf(f,df1,df2)` for `P(Fdf1,df2 ≤ f)`; `qf(p,df1,df2)` for finding `f` such that `P(Fdf1,df2 ≤ f) = p`

Randomness

- You can use `rbinom()`, `rnorm()`, etc. to generate sequences of random numbers from distributions.
- Use `sample()` to generate a random sample of a certain size from a list of numbers or data.
- Use `replicate()` to create a list by repeating a process multiple times.

Home page : Introduction

Hypothesis Testing

The Language of Hypothesis Testing

Two types of Hypotheses.

- Null hypothesis (H0)

```
* Usually a statement of “no effect”. Also, refer to the
status quo (no change from the past, the old standard still
correct).
* Either reject or do not reject H0
* For example, In our caffeinated drink example, the null
hypothesis is as follows:
```

> H0: the population mean increase in pulse rate is the same for caffeinated and decaffeinated drinkers among young adults (or caffeinated drinks has no effect on pulse rate among young adults)

- Alternative hypothesis (H1)

```
* Usually a statement of “an effect”.
* Also refers challenges to the status quo (something new is
```

> H1: the population mean increase in pulse rate is higher for caffeinated drinkers among young adults (or caffeinated drinks increase the pulse rate among young adults)

The concept of p-value

- We use the concept of p-value to reject or do not reject the null hypothesis.
- This p-value is always reported in scientific papers that use hypothesis testing.
- p-value is mostly denoted by p.
  - If p-value is small, we reject the null hypothesis and conclude that we have evidence to accept the alternative hypothesis.
  - If p-value is large, we do not reject the null hypothesis and conclude that we do not have evidence to accept the alternative hypothesis.
- The strength of evidence against the null hypothesis is determined by the magnitude of the p-value.

p-value	Interpretation
$p<0.01$	strong evidence against H0
$0.01 \leq p<0.05$	moderate evidence against H0
$0.05 \leq p<0.1$	weak evidence against H0
$p \geq 0.1$	no evidence against H0

- The commonly used threshold is 0.05. If we find $p < 0.05$, then we say that the results are significant at 5% level of significance.
- You will see in scientific journal articles *“theresults were found to be*

significant ($p < 0.05$)”.

Randomness and Probability Theory

Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Expected Value (μ) and Variance (Discrete Probability)

- The expected value of a discrte distribution is the sum of the value multiplied that probability of the value occuring.

$$\mathbb{E}[X] = \mu = \sum x \cdot P(X = x)$$

- We can quantify the variability of a discrete random variable using squared deviations about the mean.

$$Var(X) = \sum P(X = x) \cdot (x - \mu)^2$$
$$SD(X) = \sqrt{Var(X)}$$

Expected Value (μ) and Variance (Continuous Probability)

Probability Distributions

Binomial Distribution

- 2 outcomes, `success` and `failure`
- P(`success`) = p and is constant.
- A Bernoulli trial is a random process with only two possible outcomes. These outcomes are usually labelled “success” and “failure”.
- Consider a series of independent Bernoulli trials and count thenumber of successes.
- Let X be the number of successes from n number of independent Bernoulli trials and P(Success) = p.
- Then we call X has a Binomial distribution with parameters n and p.
- Mathematically represent:

$$X \sim Binom(n, p)$$

Example:

X		P(X=x)	R Code
0	X ~ Binom(3, 0.5)	0.125	dbinom(0,3,0.5)
1	X ~ Binom(3, 0.5)	0.375	dbinom(1,3,0.5)

X		P(X=x)	R Code
2	X ~ Binom(3, 0.5)	0.375	dbinom(2,3,0.5)
3	X ~ Binom(3, 0.5)	0.125	dbinom(3,3,0.5)

`dbinom` vs `pbinom`

- `dbinom(x, n, p)` returns the probability of the `x` discrete number of successes in `n` independent bernoulli trial with `p` probability of success.
  - `dbinom(x, size, prob, log = FALSE)`
- `pbinom(x, n, p, lower.tail = TRUE, log.p = FALSE)` returns the probability of the $X \leq x$ discrete number of successes in `n` independent bernoulli trial with `p` probability of success.
  - `pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)`

Usage

`dbinom(x, size, prob, log = FALSE)`
`pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)`
`qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)`
`rbinom(n, size, prob)`

Arguments

Arguments	Description
x, q	vector of quantiles.

Arguments	Description
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
size	number of trials (zero or more).
prob	probability of success on each trial.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.

Binomial Distribution Summary

`dbinom(x, size, prob)`

Put simply, `dbinom` finds the **probability of getting a certain number of successes (x) in a certain number of trials (size) where the probability of success on each trial is fixed (prob)**.

```
# find the probability of 10 successes during 12 trials where the
probability of
# success on each trial is 0.6
dbinom(x=10, size=12, prob=.6)
# [1] 0.06385228
```

`pbinom(q, size, prob)`

Put simply, **pbinom returns the area to the left of a given value q in the binomial distribution**. If you're interested in the **area to the right of a given value q, you can simply add the argument** `lower.tail = FALSE` as in:

`pbinom(q, size, prob, lower.tail = FALSE)`

```
#find the probability of more than 2 successes during 5 trials
where the
#probability of success on each trial is 0.5
pbinom(2, size=5, prob=.5, lower.tail=FALSE)
# [1] 0.5

#find the probability of less then or equal to 1 success during 5
trials where the
#probability of success on each trial is 0.5
pbinom(1, size=5, prob=.5, lower.tail=TRUE)
# [1] 0.1875
```

`qbinom(q, size, prob)`

The function qbinom returns the value of the inverse cumulative density function (cdf) of the binomial distribution given a certain random variable q, number of trials (size) and probability of success on each trial (prob).

Put simply, you can use **qbinom to find out the p^{th} quantile of the binomial distribution** or **what is expected to happen with probability** `p`.

```
#find the 10th quantile of a binomial distribution with 10 trials
and prob
#of success on each trial = 0.4
qbinom(.10, size=10, prob=.4)
# [1] 2

#find the 40th quantile of a binomial distribution with 30 trials
and prob
#of success on each trial = 0.25
qbinom(.40, size=30, prob=.25)
# [1] 7
```

`rbinom(n, size, prob)`

The function **rbinom generates a vector of binomial distributed random variables given a vector length n, number of trials (size) and probability of success on each trial (prob)**. The syntax for using rbinom is as follows:

```
#generate a vector that shows the number of successes of 10
binomial experiments with
#100 trials where the probability of success on each trial is 0.3.
results <- rbinom(10, size=100, prob=.3)
results
# [1] 31 29 28 30 35 30 27 39 30 28

#find mean number of successes in the 10 experiments (compared to
expected
#mean of 30)
mean(results)
# [1] 32.8

#generate a vector that shows the number of successes of 1000
```

Important Equations (μ and σ etc)

- $X \sim Binom(n, p)$
- $Mean = E(X) = np$
- $Var(X) = np(1-p)$
- $sd(X) = np(1-p)$

where `n` is the number of trials and `p` is the probability of `success` on each trial.

Normal Distribution

Let X be a continuous random variable. If `X` has a Normal distribution, we write $X \sim Normal(\mu, \sigma)$.

There is a rough rule to calculate the areas under Normal density curve.

```
* the area within 1 standard deviation of the mean is 68%
* the area within 2 standard deviation of the mean is 95%
* the area within 3 standard deviation of the mean is 97%
```

The area under the Normal density curve is 1.

$$
\begin{aligned}
X &\sim Normal(\mu, \sigma) \\
Z &= \frac{X - \mu}{\sigma} \\
Z &\sim Normal(0, 1)
\end{aligned}
$$

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to mean and standard deviation equal to sd.

Usage:

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

Arguments	**Description**
x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
mean	vector of means.
sd	vector of standard deviations.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$

Arguments	Description
	otherwise, $P[X > x]$.

`dnorm(x, mean, sd)`

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
```

- The function dnorm returns the value of the probability density function (pdf) of the normal distribution given a certain random variable x, a population mean μ and population standard deviation σ. The syntax for using dnorm is as follows:
- **Essentially gives the height of the probability density function at the value `x`**
- Usally not used, but `pnorm` and `qnorm` are more often used.
- The following code illustrates a few examples of dnorm in action:

```
#find the value of the standard normal distribution pdf at x=0
dnorm(x=0, mean=0, sd=1)
# [1] 0.3989423

#by default, R uses mean=0 and sd=1
dnorm(x=0)
# [1] 0.3989423

#find the value of the normal distribution pdf at x=10 with
mean=20 and sd=5
dnorm(x=10, mean=20, sd=5)
# [1] 0.01079819
```

`pnorm(q, mean, sd)`

```
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

- The function pnorm returns the value of the cumulative density function (cdf) of the normal distribution given a certain random variable q, a population mean μ and population standard deviation σ. The syntax for using pnorm is as follows:
- Put simply, **pnorm returns the area to the left of a given value x in the normal distribution**. If you're interested in the **area to the right of a given value q, you can simply add the argument lower.tail = FALSE** `pnorm(q, mean, sd, lower.tail = FALSE)`.
- The following examples illustrates how to solve some probability questions using pnorm.

Suppose the height of males at a certain school is normally distributed with a mean of μ=70 inches and a standard deviation of σ = 2 inches. Approximately what percentage of males at this school are taller than 74 inches?

Values Greater than a Number

```
# find percentage of males that are taller than 74 inches in a
population with
# mean = 70 and sd = 2
pnorm(74, mean=70, sd=2, lower.tail=FALSE)

# [1] 0.02275013
```

Values Less than a Number

```
# Additionally and unrelated to the data set above
# find percentage of otters that weight less than 22 lbs in a
population with
# mean = 30 and sd = 5
pnorm(22, mean=30, sd=5)

# [1] 0.05479929
```

Values Between Two Numbers

```
# Percentage with values between two points
# find percentage of plants that are less than 14 inches tall,
then subtract the
# percentage of plants that are less than 10 inches tall, based
on a population
# with mean = 13 and sd = 2
pnorm(14, mean=13, sd=2) - pnorm(10, mean=13, sd=2)

# [1] 0.6246553
```

`qnorm(p, mean, sd)`

```
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

- The function qnorm returns the value of the inverse cumulative density function (cdf) of the normal distribution given a certain random variable p, a population mean μ and population standard deviation σ.
- The syntax for using qnorm is as follows:
- Put simply, you can **use qnorm to find out what the Z-score is of the p^{th} quantile of the normal distribution**.

```
#find the Z-score of the 99th quantile of the standard normal
distribution
qnorm(.99, mean=0, sd=1)
# [1] 2.326348

#by default, R uses mean=0 and sd=1
qnorm(.99)
# [1] 2.326348

#find the Z-score of the 95th quantile of the standard normal
distribution
qnorm(.95)
# [1] 1.644854

#find the Z-score of the 10th quantile of the standard normal
distribution
qnorm(.10)
# [1] -1.281552
```

VERY IMPORTANT

Find Z-value of a variable x given μ and σ.

What is the Z-value of a variable with value 132, in a population with $\mu = 100$ and $\sigma = 15$.

```
# Get the percentage of population below the value.
# Answer is the percentile rank / score of the value
percent <- pnorm(132, mean=100, sd=15, lower.tail=TRUE);
percent
[1] 0.9835513

# Get the Z-score of a value at the $p^{th}$ percentile in a
standard normal distribution
```

`rnorm(n, mean, sd)`

```
rnorm(n, mean = 0, sd = 1)
```

- The function rnorm generates a vector of normally distributed random variables given a vector length `n`, a population meanμ and population standard deviation σ.

```
#generate a vector of 5 normally distributed random variables
with mean=10 and sd=2
five <- rnorm(5, mean = 10, sd = 2)
five
# [1] 10.658117 8.613495 10.561760 11.123492 10.802768

#generate a vector of 1000 normally distributed random variables
with mean=50 and sd=5
narrowDistribution <- rnorm(1000, mean = 50, sd = 15)

#generate a vector of 1000 normally distributed random variables
with mean=50 and sd=25
wideDistribution <- rnorm(1000, mean = 50, sd = 25)

#generate two histograms to view these two distributions side by
side, specify
#50 bars in histogram and x-axis limits of -50 to 150
par(mfrow=c(1, 2)) #one row, two columns
hist(narrowDistribution, breaks=50, xlim=c(-50, 150))
hist(wideDistribution, breaks=50, xlim=c(-50, 150))
```

Poputation vs Sample Standard Deviation

$$sd(\overline{X}) = \frac{\sigma}{\sqrt{n}}$$
$$\sigma = sd(\overline{X}) \cdot \sqrt{n}$$

where, `n` is the sample size.

If $X \sim Normal(\mu, \sigma)$:

$$Z = \frac{X - \mu}{\sigma}$$
$$Z \sim Normal(0, 1)$$

The distribution of sample means ($\overline{X}$):

$$\overline{X} \sim Normal(\mu, \frac{\sigma}{\sqrt{n}})$$
$$Z = \frac{X - \mu}{\frac{\sigma}{\sqrt{n}}}$$
$$Z \sim Normal(0, 1)$$

This is true even if the population is not normally distributed for sufficiently large `n` sample size, ***Central Limit Theorem*** applies and the sample behaves like a normally distributed sample.

Sampling Distribution of the Sample Proportions

- The sample proportion ($\hat{p}$)
- Again, consider our height of STAT1201 students.
- Now define `p` as the population proportion of students whose height is less than or equal to 155cm.

- Following similar steps like we did for the sampling distribution of the sample mean we can find the distribution of the sample proportion.

$$\hat{p} = \frac{x}{n}$$

where `x` is the number of students in the sample whose height is less than or equal to 155cm.

- If all possible samples of size `n` are selected from a binomial distribution, We can show that:
  - $\mathbb{E}(\hat{p}) = p$
  - $\text{sd}(\hat{p}) = \sqrt{\frac{p\,(1-p)}{n}}$

Summary

Binomial Distribution

Find the probability of `x` successes in `n` trials with `p` probability of success on each trial

Find the probability of `x` successes between 2 values, less than a value, or greater than a value in `n` trials with `p` probability of success on each trial

Find what discrete value `x` has a `q` probability of happening in `n` trials with `p` probability of success on each trial

Generate vector of `n` values from a binomial distribution, `size` trials and `prob` chance of success

Normal Distribution

Finding percent of values less than or greater than a number and between 2 numbers

Finding the Z-value of a number, given μ and σ

Generate vector of `n` values from a normal distribution, with μ and σ

Distribution of Proportion Variables

Sample mean and standard deviation for proportion values

Descriptive Statistics

Tukey Five-Number Summaries

- Description: Returns Tukey's five number summary (minimum, lower-hinge, median, upper-hinge, maximum) for the input data.
- Usage:

```
fivenum(x, na.rm = TRUE)
```

- Arguments:
  - `x`: numeric, maybe including NAs and $\pm\pm Inf$s.
  - na.rm : logical; if TRUE, all NA and NaNs are dropped, before the statistics are computed.
- Value
  - A numeric vector of length 5 containing the summary information. See `boxplot.stats` for more details.

```
# Load the file
mucus <- read.csv("MucusCells(1).csv")

summary(mucus)
     Cells          Group
 Min.   : 8.00   Length:20
 1st Qu.:10.00   Class :character
 Median :12.50   Mode  :character
 Mean   :12.75
```

IQR (Inter-Quartile Range)

- Description: computes interquartile range of the x values.
- Usage:

```
IQR(x, na.rm = FALSE, type = 7)
```

- Arguments
  - `x` : a numeric vector.
  - na.rm : logical. Should missing values be removed?
  - type : an integer selecting one of the many quantile algorithms, see quantile.
- Details : Note that this function computes the quartiles using the quantile function rather than following Tukey's recommendations, i.e., IQR(x) = quantile(x, 3/4) - quantile(x, 1/4).
- For normally N(m,1)N(m,1) distributed XX, the expected value of IQR(X) is 2*qnorm(3/4) = 1.3490, i.e., for a normal-consistent estimate of the standard deviation, use IQR(x) / 1.349.

```
IQR(mucus$Cells)
[1] 5.5
```

Return value depends on the algorithm used.

Boxplot

- Description: Produce box-and-whisker plot(s) of the given (grouped) values.

- Usage:

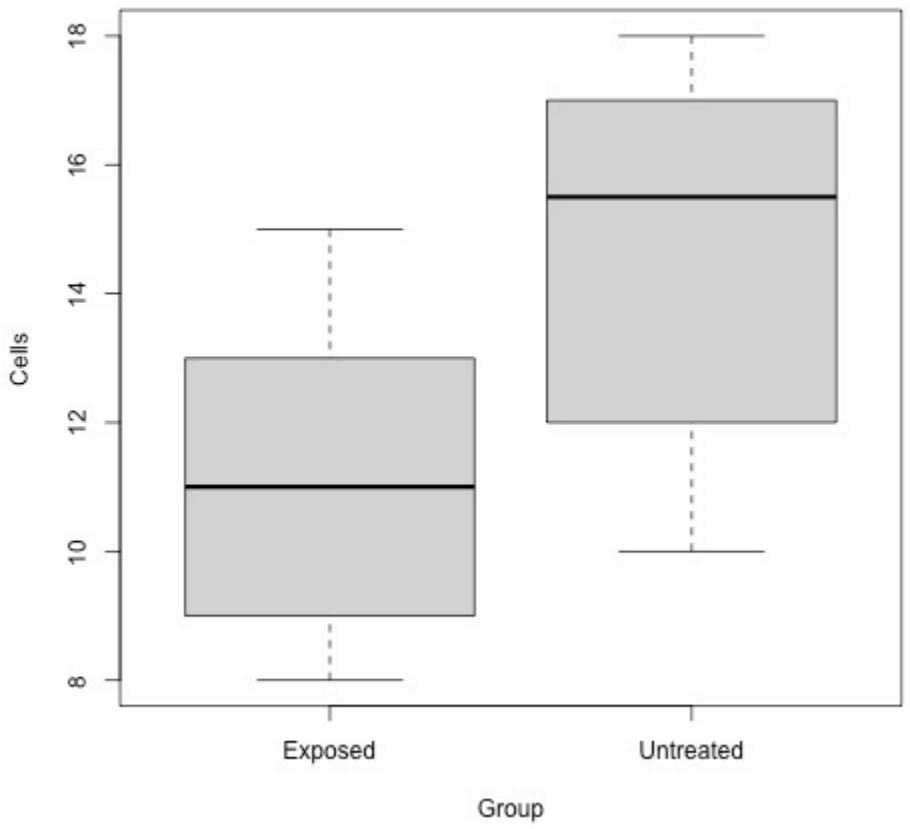
```
boxplot(x, ...)

## S3 method for class 'formula'
boxplot(formula, data = NULL, ..., subset, na.action = NULL,
        xlab = mklab(y_var = horizontal),
        ylab = mklab(y_var =!horizontal),
        add = FALSE, ann = !add, horizontal = FALSE,
        drop = FALSE, sep = ".", lex.order = FALSE)

## Default S3 method:
boxplot(x, ..., range = 1.5, width = NULL, varwidth = FALSE,
        notch = FALSE, outline = TRUE, names, plot = TRUE,
        border = par("fg"), col = "lightgray", log = "",
        pars = list(boxwex = 0.8, staplewex = 0.5, outwex = 0.5),
         ann = !add, horizontal = FALSE, add = FALSE, at = NULL)
```

- Arguments
  - formula: a formula, such as y ~ grp, where y is a numeric vector of data values to be split into groups according to the grouping variable grp (usually a factor). Note that ~ g1 + g2 is equivalent to g1:g2.
  - data: a data.frame (or list) from which the variables in formula should be taken.
  - subset: an optional vector specifying a subset of observations to be used for plotting.
  - na.action: a function which indicates what should happen when the data contain NAs. The default is to ignore missing values in either the response or the group.
  - xlab, ylab: x- and y-axis annotation, since R 3.6.0 with a non-empty default. Can be suppressed by ann=FALSE.
  - ann: logical indicating if axes should be annotated (by xlab and ylab).

```
boxplot(Cells ~ Group, mucus)
```



Aggregate

Compute Summary Statistics of Data Subsets

- Description: Splits the data into subsets, computes summary statistics for each, and returns the result in a convenient form.
- Usage:

```
aggregate(x, ...)

## Default S3 method:
aggregate(x, ...)

## S3 method for class 'data.frame'
aggregate(x, by, FUN, ..., simplify = TRUE, drop = TRUE)
```

- Arguments
  - x: an R object. For the formula method a formula, such as y ~ x or cbind(y1, y2) ~ x1 + x2, where the y variables are numeric data to be split into groups according to the grouping x variables (usually factors).
  - by: a list of grouping elements, each as long as the variables in the data frame x. The elements are coerced to factors before use.
  - FUN: a function to compute the summary statistics which can be applied to all data subsets.

```
# Aggregate Means by Group
aggregate(Cells ~ Group, mucus, mean)
      Group Cells
1   Exposed  11.0
2 Untreated  14.5

# Aggregate SD by Group
aggregate(Cells ~ Group, mucus, sd)
      Group    Cells
1   Exposed 2.309401
2 Untreated 2.798809
```

Sample Standard Deviation

- `s` is the sample standard deviation, `n` is the sample size.

$$s = \sqrt{\frac{\sum(x_j - \overline{x})^2}{n-1}}$$

Final Notes

Binomial Distribution Summary

`dbinom(x, size, prob)`

Put simply, `dbinom` finds the **probability of getting a certain number of successes (x) in a certain number of trials (size) where the probability of success on each trial is fixed (prob)**.

```
#find the probability of 10 successes during 12 trials where the
probability of
#success on each trial is 0.6
dbinom(x=10, size=12, prob=.6)
# [1] 0.06385228
```

`pbinom(q, size, prob)`

Put simply, **pbinom returns the area to the left of a given value q in the binomial distribution**. If you're interested in the **area to the right of a given value q, you can simply add the argument** `lower.tail = FALSE` as in:

`pbinom(q, size, prob, lower.tail = FALSE)`

```
#find the probability of more than 2 successes during 5 trials
where the
```

`qbinom(q, size, prob)`

The function qbinom returns the value of the inverse cumulative density function (cdf) of the binomial distribution given a certain random variable q, number of trials (size) and probability of success on each trial (prob).

Put simply, you can use **qbinom to find out the p^{th} quantile of the binomial distribution** or **what is expected to happen with probability `p`**.

```
#find the 10th quantile of a binomial distribution with 10 trials
and prob
#of success on each trial = 0.4
qbinom(.10, size=10, prob=.4)
# [1] 2

#find the 40th quantile of a binomial distribution with 30 trials
and prob
#of success on each trial = 0.25
qbinom(.40, size=30, prob=.25)
# [1] 7
```

`rbinom(n, size, prob)`

The function **rbinom generates a vector of binomial distributed random variables given a vector length n, number of trials (size) and probability of success on each trial (prob)**. The syntax for using rbinom is as follows:

```
#generate a vector that shows the number of successes of 10
binomial experiments with
```

Important Equations (μ and σ etc)

- $X \sim Binom(n, p)$
- $Mean = E(X) = np$
- $Var(X) = np(1-p)$
- $sd(X) = np(1-p)$

where `n` is the number of trials and `p` is the probability of `success` on each trial.