Statistics Summary

Let's discover **Introductory Statistics**.

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 the number 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 pth 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