Biostatistics with **R**

Guo-Liang TIAN

Department of Mathematics,
Southern University of Science and Technology,

Shenzhen, Guangdong Province, P.R. China

1 September 2018

To Yanli, Margaret and Adam

Contents

Chapter 1

Introduction to Data Analysis

1• Statistics

- The term *statistics* is derived from the Latin for state, and originally conceived as the science of the state — the collection and analysis of facts about a country: its economy, land, military, population, and so forth.

- Statistics is a mathematical body of science that pertains to the collection, analysis, interpretation, and presentation of data.

- Some consider statistics to be a distinct mathematical science rather than a branch of mathematics.

- While many scientific investigations make use of data, statistics is concerned with the use of data in the context of uncertainty and decision making in the face of uncertainty.

2• Mathematical statistics

- Mathematical statistics is the application of mathematics to statistics.

- Mathematical techniques used for this include mathematical analysis, linear algebra, stochastic analysis, differential equations, measure theory, and probability theory.

3• Biostatistics

- Biostatistics is the application of statistics to a wide range of topics in biology.

- Biostatistics includes the design of biological experiments (especially in medicine, pharmacy, agriculture and fishery); the collection, summarization and analysis of data from those experiments; and the interpretation of the results.

- A major branch of biostatistics is *medical statistics*, which is exclusively concerned with medicine and health.

4• Outline of this course

- In this course, a number of statistical methods learned in the first and/or second year statistics courses will be reviewed and more statistical methods will be introduced.

- This course will present a wide range of statistical methodologies using real data sets.

- The emphasis will be on the appropriate practical application of the statistical methods in analyzing data and its general strategy with the help of the advanced computer technology, such as

  — how to process the data,

  — how to formulate a sensible model, and

  — how to choose an appropriate method of analysis.

- We will make use of the statistical software R extensively.

1.1 The basic research process

1.1.1 Eight steps of the basic research process

5• Observation of a particular event

- Generally, an observation can be classified as either *quantitative* or *qualitative*. For further details, see Section 1.3.

- *Quantitative observations* are based on some sort of measurements, e.g., length, weight, and temperature.

- *Qualitative observations* are based on categories reflecting a quality or characteristic of the observed event, e.g., male versus female, diseased versus healthy, and mutant versus wild type.

6• Statement of the problem (your objectives)

6.1• Cause and effect relationship

— A series of observations often leads to the formulation of a particular problem or unanswered questions.

— This usually takes the form of a "why" question and implies a *cause and effect relationship.*

6.2• An example of hypertension

— In an investigation of a remote Fijian island community, you realized that the vast majority of the adults suffer from *hypertension.*

— Abnormally elevated blood pressures with the *systolic* over 165 mmHg and the *diastolic* over 95 mmHg.

— Note that the individual observations here are quantitative while the percentage that are hypertension is based on a qualitative evaluation of the sample.

— From these preliminary observations, one may formulate the question as "*Why are so many adults in this population hypertensive?*"

7• Formulation of a hypothesis

- A hypothesis is a *tentative explanation* for the observations made.
- A good hypothesis suggests a *cause and effect relationship* and is testable.
- The Fijian community may demonstrate hypertension because of diet, life style, genetic makeup, or combinations of these factors.
- Because we have noticed extraordinary consumption of *octopi* in their diet and knowing octopods have a very high cholesterol content, we might hypothesize: "*The high level of hypertension is caused by diet.*"

8• Making a prediction

8.1• Prediction taking the form of an if–then statement

— If the hypothesis is properly constructed, it can and should be used to make predictions.

— Predictions are based on deductive reasoning and take the form of an "if–then" statement.

— For example, a good prediction based on the hypothesis above would be: "*If the hypertension is caused by a high cholesterol diet, then changing the diet to a low cholesterol one should lower the incidence of hypertension.*"

8.2• Criteria for a valid prediction

An "if clause" states the hypothesis while a "then clause"

— suggests altering a causative factor in the hypothesis [*change the diet*];

— predicts the outcome [*lower level of hypertension*];

— provides the basis for an experiment.

9• Selection of an appropriate experimental design

9.1• Aim of an experimental design

— The purpose of an experimental design is to accomplish one goal, that is, to test the hypothesis.

— Theoretically, an experiment should alter or test only the factor suggested by the prediction, while all other factors remain constant.

9.2• Case-control design

— This involves using two randomly chosen groups of adults from the community and treating both identically with the exception of the one factor being tested.

— *Control group*: represents the "normal" situation, has all factors present, and is used as a basis for comparison.

— *Case/experiment group*: represents the "test" situation and includes all factors except the factor that has been altered, in above case the diet.

10• Data collection

- Commonly utilized data collection approaches in clinical research include

  — Questionnaire surveys and patient reported data.

  — Proxy or informant data.

  — Review of ambulatory or hospital medical records.

  — Collection of biologic material.

11• Data analysis

- If the group with the low cholesterol diet exhibits *significantly* lower levels of hypertension, the hypothesis is supported by the data.
- If the change in diet has no effect on hypertension, then a new or revised hypothesis should be formulated.

12• Reporting results and assessing their implications

- The results of your statistical analyses help you to understand the outcome of your study, e.g.,

  — whether or not some variable has an effect,

  — whether variables are related,

  — whether differences among groups of observations are the same or different, etc.

- Statistics should be used to substantiate your findings and help you to say objectively when you have significant results.
- Therefore, when reporting the statistical outcomes relevant to your study, subordinate them to the actual biological results.

1.1.2 The model-building process

13$^\bullet$ Statistical model

- In the analysis step, the aim of many statistical techniques is a simplified description of the structure of the observations by means of what is usually referred to as a *statistical model.*
- This statistical model could be a *graphical representation* or a *mathematical equation.*
- The purpose of building a model is to provide the simplest description of the population being studied.

14$^\bullet$ Five steps in the model-building

- Step 1: Preliminary exploration of the data.
- Step 2: Postulate a general class of models from prior knowledge.
- Step 3: Identify a model.
- Step 4: Estimate the parameters of the model selected in Step 3.
- Step 5: Check the adequacy of the model using significance tests and graphical examination of the residuals.
  - — If model is inadequate, go back to Step 3 by identifying a new model.
  - — If model is adequate then stop and draw conclusions.

15$^\bullet$ Interpretations to a statistical model

- In general, such model-building process can be summarized in terms of the following mathematical equation:

$$\textit{Observation} = \textit{model} + \textit{residual} \quad \text{or} \quad y = f(x, \beta) + \varepsilon.$$

- The model $f(x, \beta)$ is the underlying, simplified structure of the observations y's.
- The residual ε represents random fluctuation, which is the difference between the observed data points and the model.

- — Hopefully the residuals should contain no additional pattern or structure or else the model needs to be modified.
- — The process should continue until no structure or pattern can be found in the residuals.

16• No perfect model

- However the following quotation from G. Box (a famous statistician) in 1965 should be kept firmly in mind when constructing a model.

  “*All models are wrong, but some are useful.*”

- We are not trying to find a model which fits perfectly to the data.
- Instead, we would like to find a simple or parsimonious model which has good intuitive interpretation, and can help us to describe certain phenomena or characteristics of the population.

1.2 Populations and samples

1.2.1 Populations

17• Definition of a population

- A complete set of elements (persons or objects) that possess some common characteristic defined by the sampling criteria established by the researcher.
- Population is composed of two groups: *target population* and *accessible population.*

18• Target population (universe)

- The entire group of people/objects to which the researchers wish to generalize the study findings.
- Meet set of criteria of interest to researchers.

18.1• Some illustration examples

— All institutionalized elderly with Alzheimer’s;

— All people with AIDS;

— All low birth weight infants;

— All school-age children with asthma;

— All pregnant teens.

19• Accessible population

- The portion of the population to which the researchers have reasonable access; may be a subset of the target population.
- May be limited to region, state, city, county, or institution.

19.1• Some illustration examples

— All institutionalized elderly with Alzheimer's in St. Louis county nursing homes;

— All people with AIDS in the metropolitan St. Louis area;

— All low birth weight infants admitted to the neonatal ICUs in St. Louis city & county;

— All school-age children with asthma treated in pediatric asthma clinics in university-affiliated medical centers in the Midwest;

— All pregnant teens in the state of Missouri.

1.2.2 Samples

Terminologies used to describe samples and sampling methods include

20• Sample

- The selected elements (people or objects) chosen for participation in a study; people are referred to as subjects or participants.

21• Sampling

- The process of selecting a group of people, events, behaviors, or other elements with which to conduct a study.

22• Sampling frame

- A list of all the elements in the population from which the sample is drawn.
- Could be extremely large if population is national or international in nature.
- Frame is needed so that everyone in the population is identified so they will have an equal opportunity for selection as a subject (element).

22.1• Some illustration examples

— A list of all institutionalized elderly with Alzheimer's in St. Louis county nursing homes affiliated with BJC;

— A list of all people with AIDS in the metropolitan St. Louis area who are members of the St. Louis Effort for AIDS;

— A list of all low birth weight infants admitted to the neonatal ICUs in St. Louis city & county in 1998;

— A list of all school-age children with asthma treated in pediatric asthma clinics in university-affiliated medical centers in the Midwest;

— A list of all pregnant teens in the Henderson school district.

23• Randomization

- Each individual in the population has an equal opportunity to be selected for the sample.

24• Random selection

- From all people who meet the inclusion criteria, a sample is randomly chosen.

25• Random assignment

- The assignment of subjects to treatment conditions in a random manner.

- It has no bearing on how the subjects participating in an experiment are initially selected.

26• Representativeness

- Sample must be as much like the population in as many ways as possible.
- Sample reflects the characteristics of the population, so those sample findings can be generalized to the population.
- Most effective way to achieve representativeness is through randomization; random selection or random assignment.

27• Parameter

- A numerical value or measure of a characteristic of the population; remember P for parameter & population.

28• Statistic

- Numerical value or measure of a characteristic of the sample; remember S for sample & statistic.

29• Precision

- The accuracy with which the population parameters have been estimated; remember that population parameters often are based on the sample statistics.

30• Sampling error

- The difference between the sample statistic (e.g., sample mean) and the population parameter (e.g., population mean) is due to the random fluctuations in data that occur when the sample is selected.

31• Sampling bias

- Also called systematic bias or systematic variance.

— The difference between sample data and population data that can be attributed to faulty sampling of the population.

— Consequence of selecting subjects whose characteristics (scores) are different in some way from the population they are suppose to represent.

— This usually occurs when randomization is not used.

1.3 Data types

32• Statistical model and type of data

- In data analysis, the most appropriate statistical model depends on the answers to two primary questions:

  — What is the purpose or objective of the statistical analysis?

  — What type of data is to be analyzed?

33• Importance of data classification

- The nature of the observations is of major importance in relation to the choice of correct statistical models and methods of analysis.
- It is sensible to start with a brief discussion of various types of data that may be encountered in applications.
- Data can be classified into *categorical* (or qualitative) and *numerical* (or quantitative) data.

1.3.1 Categorical/qualitative data

34• Two-category data

- The simplest type of observations on an individual is the allocation of that individual to one of only two possible categories.
- Often these relate to the absence (usually denoted by 0) or presence (usually denoted by 1) of some attribute.
- Such data have numerous other names such as *binary* data, *dichotomous* data, attribute data, yes/no data, and 0–1 data.

34.1• Examples of such categorizations for patients include:

— male/female;

— pregnant/not pregnant;

— married/single;

— diabetic/non-diabetic;

— smoker/non-smoker;

— hypertensive/normotensive.

35• Multi-category data

35.1• Examples

Clearly many classifications require more than two categories, e.g.,

— married/single/divorced/separated/widowed;

— juvenile-onset diabetes/maturity-onset diabetes/non-diabetic.

35.2• Nominal data

Multi-category data *without* obvious order in the categories are called nominal data. In the strict sense of words, there are no measurements and no scales involved. For example:

— Blood types can be classified as A, B, AB, and O;

— USA citizen can be classified as White, African-American, Asian or Pacific Islander, and Native America;

— Hair color can be classified as brown, black, blonde, gray, and others.

35.3• Ordinal data

Multi-category data *with* a natural order in the categories are called ordinal data. For example:

— Smokers can be classified as non-smokers, ex-smokers, light smokers, and heavy smokers;

— Degree of pain can be classified as minimal, moderate, severe, and unbearable.

35.4• A caution in rating ordinal data

An example is the response to the question: "*How about this statistics course when comparing it with other courses you are taking*? "

— The answer can be any one of the five choices, namely (1) superior; (2) good; (3) average; (4) poor; and (5) inferior.

— One category is higher than the next one; that is *superior* is a higher rating than *good*, *good* is higher than *average* and so on.

— If 1 is substituted for *superior*, 2 substituted for *good*, and so on.

— A ranking of 1 is obviously higher than a ranking of 2, and a 2 is higher than a 3.

— However, it cannot be said that a student rated *good* likes this course twice as much as a student who rated average.

— It can only be said that a rating of good is greater than a rating of average.

1.3.2 Numerical/quantitative data

36• Discrete data

36.1• Definition and examples

Discrete numerical data arise when the observations in question can only take certain numerical values. Virtually all examples are counts of events, such as

— number of children;

— number of visits to Southern University of Science and Technology;

— number of ectopic heart beats in 24 hours, etc.

36.2• Discrete numerical data and ordinal data

The difference between the discrete numerical data and the ordinal data can be seen by considering an example of each:

Discrete numerical data	Number of children: 0, 1, 2, 3, 4, 5+
Ordinal data	Stage of breast cancer: I, II, III, IV, V

— We cannot say that stage IV is twice as bad as stage II nor that the difference between stages I and II is equivalent to that between stages III and IV.

— In contrast, four children are twice as many as two children, and a difference of one means the same throughout the range of values.

— Even ordered categories (e.g., social class or disease stage) are numbered, it is not sensible to calculate the average social class or stage of cancer.

— The only information the numbers contained in ordinal data is in the ordering, which would be conveyed equally by calling them A, B, C, D and so on.

37• Continuous data

37.1• Definition and examples

Continuous data are usually obtained by some form of measurement. Common examples include

— height;

— weight;

— age;

— blood pressure;

— serum cholesterol;

— haemoglobin; and

— month or year salary.

37.2• From discrete to continuous

Sometimes it is reasonable to treat discrete data as if they were continuous. For example,

— Age at last birthday is discrete. In studies of adults with ages ranging from, say 16 to 80, no harm is done in considering age in years as a continuous measurement.

— Heart rate (in beats per minute) is another discrete measurement that is usually regarded as continuous.

37.3• From continuous to categorical

Conversely, continuous data are often reduced to several categories. For example:

— If the variable is known to be imprecise, such as reported number of cigarettes smoked per day, it may be sensible to have categories such as 0, 1–10, 11–20, 21 or more;

— Another example is the length of oral contraceptive use: 0–1 years, 2–5 years, 6–10 years, more than 10 years.

1.3.3 Censored data

38• Survival data

- When recording times from some fixed starting point (e.g., surgery) to the death of patients, we usually refer to *survival times* or *survival data*.

- An observation is called *censored* if we cannot measure it precisely but know that it is beyond some limit.

39• Right censored data

- In a study to compare the survival of patients having different types of surgery from breast cancer, although the patients will be followed up for several years there will be many who are still alive at the end of the study.

- For these patients we do not know when they will die, only that they are still alive at the end of the study.
- We call their survival times right censored.

40• Left censoring data

- If a tumor is shrunken and its volume is less than $0.01cm^3$, the volume of tumor can not be measured by the machine.
- We do not know the exact volume, only know the volume is less than $0.01cm^3$.
- Such data are known as left censoring.

1.3.4 Other types of data

41• Interval data

- The interval data include all the characteristics of the *ordinal data*, but in addition, the unit distance between values is a constant size.
- For example, temperature is on the Celsius (or Fahrenheit) scale.
  - — If the highest and lowest temperatures of yesterday were 16°C and 8°C, then the interval data are denoted by [8°C, 16°C].
  - — These two temperatures can easily be ranked, but we can also determine the differences between the temperatures.
  - — One degree Celsius represents a constant unit of measurement.
  - — Moreover, it is important to note that the zero point is arbitrary and it does not represent the absence of heat, just that it is cold.

42• Ratio data

- Ratio data arise when we take ratio of two variables.
- For example,
  - — ejection fraction (an important cardiac function index) is the ratio of the difference between end systolic volume and diastolic volume to end systolic volume (cardiac output);

— the percent change in renal function (e.g., the glomerular filtration rate) from certain baseline;

— More recently, the microarray gene expression ratio has become a focus of many cutting-edge medical research.

— The microarray technology has allowed fast large scale (up to thousands of genes) analysis of gene expression.

— In these experiments, the ratios of gene expression from the diseased tissue samples to that of reference samples are expressed as spot for each gene.

43• Continuous proportional data

- This is really a subtype of ratio data when the ratio is a percentage between 0 and 1.

- It includes data such as the percent decrease in renal functions at different follow-up times from the baseline, and percent changes from pre-treatment to post-treatment in terms of certain physiological variables or some molecular or genetic targets.

- Statistical methods to directly model the means of the proportional responses have just emerged using the simplex distribution of Barndorff–Nielsen and Jorgensen (Jorgensen, 1997).

- The simplex distribution takes into account the fact that such responses are percentages restricted between 0 and 1 and may as well have large dispersion.

- It has been discovered recently that there may well be large dispersion in this kind of data.

44• Repeated measures data

- In medical studies, subjects are often followed overtime, measurements or observations are obtained within certain experimental units or clusters (e.g., eyes or limbs of an individual).

- These observations are called *repeated measures data.*

- If they are obtained over different times from the same individual, they are sometimes call *longitudinal data.*

- This kind of design is often necessary in order to assess how patients do overtime.

- For example, we may be interested how certain physiological variables (glomerular filtration rate) or genetic variables (for instance, telomere length) change over time, or whether certain events (e.g., ear infection) occur overtime.

45• A discussion

- From the viewpoint of *scales of measurements*, the nominal scale is the lowest, or most limited level of measurement, and the ratio scale is the highest, or least limited scale among nominal, ordinal, interval and ratio data.

- Since the nature among the interval data, ratio data and continuous data are rather similar, we do not distinguish them in this course.

- We will concentrate on the analysis of continuous data, interval data, ratio data and nominal data.

1.4 Measures of location, variability and shape

46• Descriptive statistics

- Descriptive statistics quantitatively describe or summarize features of a sample.

- Descriptive statistics are distinguished from *inferential statistics*, in that descriptive statistics aim to summarize a sample, rather than use the data to learn about the population.

- There are three important characteristics:

  — Location, center or central tendency;

  — Variability, dispersion or spread;

  — Shape.

1.4.1 Measures of location

Measures of location include the *mean*, *median* and *mode*, which describe the center of a distribution.

47• Mean

47.1• Population mean

— Let $X_1, \ldots, X_n$ be a random sample from a population random variable X with unknown density $f(x)$.

— If $f(x)$ is symmetric or bell-shaped, then the central location (or the population mean) of $f(x)$ is defined by $\mu = E(X)$.

47.2• Sample mean

— The sample mean is defined by

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i, \tag{1.1}$$

which is an unbiased estimate of μ.

— The sample mean is sensitive to extreme values or outliers.

48• Quantile

48.1• Population quantile

— The q-th quantile of the population random variable X, denoted by ξ_q, is defined as the smallest number ξ satisfying

$$F_X(\xi) = \Pr(X \leqslant \xi) \geqslant q.$$

48.2• The continuous case

— If X is continuous, then the q-th quantile of X is defined as the smallest number ξ satisfying

$$F_X(\xi) = \Pr(X \leqslant \xi) = q.$$

48.3• Sample quantiles

— Sample quantiles of a random sample $X_1, \ldots, X_n$ are defined by their *order statistics*: $X_{(1)} \leqslant \cdots \leqslant X_{(n)}$.

49• Median

49.1• Population median

— In particular, the 0.5-th quantile $\xi_{.5}$ is defined as the median of X, denoted by $\text{med}(X)$.

— Alternatively, the median of X satisfies

$$\Pr\{X \leqslant \text{med}(X)\} \geqslant 0.5 \quad \text{and} \quad \Pr\{X \geqslant \text{med}(X)\} \geqslant 0.5.$$

49.2• The continuous case

— If X is continuous, then the median of X satisfies

$$\int_{-\infty}^{\text{med}(X)} f(x)\,\mathrm{d}x = 0.5 = \int_{\text{med}(X)}^{\infty} f(x)\,\mathrm{d}x. \tag{1.2}$$

49.3• Sample median

— The $\text{med}(X)$ is usually estimated by the sample median defined by

$$\text{Median}(X_1, \ldots, X_n) = \begin{cases} X_{(\frac{n+1}{2})}, & \text{if } n \text{ is odd,} \\ \dfrac{X_{(n/2)} + X_{(n/2+1)}}{2}, & \text{if } n \text{ is even.} \end{cases} \tag{1.3}$$

— When there exist outliers, the sample median of $X_1, \ldots, X_n$ is better than $\bar{X}$ to represent the central location of $f(x)$.

— In other words, the population median is the central value, lying above and below half of the population values.

— The sample median is the middle value when the data are arranged in ascending or descending order.

49.4• Demonstration by R

```
==============================================================
> x <- c(1.1, -2.3, 3.4, 4.6, 5)
> mean(x)          # = sum(x)/length(x)
[1] 2.36
> quantile(x)
  0%  25%  50%  75% 100%  # The 1-st, 2-nd and 3-rd quartiles
-2.3  1.1  3.4  4.6  5.0  # are 1.1, 3.4, and 4.6
> median(x)
[1] 3.4
**************************************************************
```

50• Mode

50.1• Population mode

— The population mode of $f(x)$ is defined by

$$\tilde{x} = \arg\max_{x \in \mathbb{X}} f(x). \tag{1.4}$$

50.2• Sample mode

— If $f(x)$ is skewed (e.g., chi-squared density) or monotone (e.g., exponential density) or bimodal (e.g., the mixture of two normal densities), we would like to find the sample mode defined as *the most frequent point in the sample*, to estimate $\tilde{x}$.

— In other words, the mode is the value at which the density of the population is at a maximum.

— Some densities have more than one local maximum (peak) and are said to be multi-modal.

— The sample mode is the value that occurs most often in the sample.

50.3• Demonstration by R

```
==============================================================
> x <- c(1, 1, 1, 2, 3, 1, 2, 6, 7)
> table(x)
```

```
1 2 3 6 7
4 2 1 1 1
> table(c(1, 2, 2, 10, 11, 11, 30))

 1  2 10 11 30
 1  2  1  2  1
```

****

1.4.2 Measures of variability

Five measures (i.e., variance, standard deviation, range, interquartile range and coefficient of variation) are used to measure the variability/dispersion of a density $f(x)$ or a random sample $X_1, \ldots, X_n$.

51• Variance and standard deviation

51.1• Population variance

— The variance $\sigma^2 = \mathrm{Var}(X) = E(X - \mu)^2$ is a measure of the dispersion of $f(x)$.

51.2• Some concepts

— The quantities $\{X_i - \bar{X}\}_{i=1}^n$ are called *deviates*, and $\sum_{i=1}^n (X_i - \bar{X}) = 0$.

— The quantity $\sum_{i=1}^n (X_i - \bar{X})^2$ is the sum of these squared deviates and is referred to as the *corrected sum of squares* (CSS).

— $\sum_{i=1}^n X_i^2$ is the *uncorrected sum of squares.*

51.3• Sample variance

— The sample variance of $X_1, \ldots, X_n$ is defined by

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2, \tag{1.5}$$

which is an unbiased estimate of σ^2.

51.4• Standard deviation

— The population standard deviation σ can be estimated by (sample) s-tandard deviation S.

51.5• Demonstration by R

```
==============================================================
> x <- 1:10
> sum(x - mean(x))
[1] 0
> var(x)       # = sum((x - mean(x))^2)/(length(x) - 1)
[1] 9.166667
> sd(x)        # = sqrt(var(x))
[1] 3.02765
**************************************************************
```

52• Quartile, decile, centile and percentile

52.1• Quartile

— The *first quartile* (Q_1 or *lower quartile*), is the 0.25-th quantile.

— The *second quartile* (Q_2 or *median*) is the 0.50-th quantile.

— The *third quartile* (Q_3 or *upper quartile*) is the 0.75-th quantile.

52.2• Deciles

— On the one hand, we have deciles for the 0.1-th, 0.2-th, . . . , 0.9-th, 1.0-th quantiles.

— For example:

```
==============================================================
> pv <- seq(0, 1, 0.1)
> pv
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> quantile(x, pv)
   0%   10%   20%   30%   40%   50% 
191.0 195.9 197.8 199.0 200.0 200.0 
  60%   70%   80%   90%  100% 
201.0 202.0 204.0 205.1 207.0 
**************************************************************
```

52.3• Centiles and percentiles

— On the other hand, we have centiles for the 0.01-th, 0.02-th, . . . , 0.99-th, 1.00-th quantiles or for the 1-th, 2-th, . . . , 99-th, 100-th percentiles.

— For example: `quantile(x, seq(0, 1, 0.01))`

52.4• Differences among quartile, percentile and quantile

— First quartile $= Q_1 =$ 25-th percentile = 0.25-th quantile;

— Second quartile $= Q_2 =$ 50-th percentile = 0.50-th quantile;

— Third quartile $= Q_3 =$ 75-th percentile = 0.75-th quantile.

53• Range and interquartile range

53.1• Sample range

— The sample range is defined by $X_{(n)} - X_{(1)}$.

53.2• Interquartile range

— The *interquartile range* (IQR) of the sample is defined by $Q_3 - Q_1$.

— IQR is used as a robust alternative to the standard deviation.

53.3• Demonstration by R

```
==============================================================
> x <- sqrt(1:1100)
> summary(x)
   Min. 1st Quartile  Median  Mean  3rd Quartile  Max.
   1.00    16.61       23.46   22.13   28.73       33.17
> quantile(x)
      0%       25%       50%       75%      100%
 1.00000 16.60572 23.46273 28.72716 33.16625
> fivenum(x)
[1]  1.00000 16.59819 23.46273 28.73151 33.16625
> IQR(x)
[1] 12.12145
> quantile(x)[[4]] - quantile(x)[[2]]
[1] 12.12145
```

```
> fivenum(x)[4] - fivenum(x)[2]
[1] 12.13333
> range(x)
[1]  1.00000 33.16625
--------------------------------------------------------------
> quantile(rnorm(100), c(0.32, 0.57, 0.98))
       32%        57%        98% 
-0.6404643 -0.1605341  1.9705579 
# The 32nd, 57th and 98th percentiles
# The 0.32, 0.57, 0.98 quantiles
**************************************************************
```

54$^\bullet$ Coefficient of variation.

- The *coefficient of variation* (CV) is a unitless measure of relative variability.

- It is defined as the ratio of the standard deviation to the mean expressed as a percentage:

$$\mathrm{CV} = \frac{100S}{\bar{X}}. \tag{1.6}$$

- The CV is meaningful only if the variable is measured on a ratio scale.

- If all sample values are multiplied by a constant, then the sample coefficient of variation remains unchanged.

1.4.3 Measures of shape

The variance is a measure of the overall size of the deviations from the mean. Since the formula for the variance squares the deviations, both positive and negative deviations contribute to the variance in the same way. In many distributions, positive deviations might tend to be larger in magnitude than negative deviations, or vice versa.

55$^\bullet$ Skewness

55.1$^\bullet$ Population skewness

— Skewness is a measure of the tendency of the deviations to be larger in one direction than in the other.

— The population skewness is defined as

$$\frac{E(X-\mu)^3}{\sigma^3}.$$

— Because the deviations are cubed rather than squared, the signs of the deviations are maintained.

— Cubing the deviations also emphasizes the effects of large deviations.

— The formula includes a divisor of σ^3 to remove the effect of scale, so multiplying all values by a constant does not change the skewness.

— Skewness can thus be interpreted as a tendency for one tail of the population to be heavier than the other.

— Skewness can be positive or negative and is unbounded.

55.2• Sample skewness

— The sample skewness is defined by

$$\frac{n}{(n-1)(n-2)}\sum_{i=1}^{n}\frac{(X_i-\bar{X})^3}{S^3}. \qquad (1.7)$$

— For a normal random sample, the sample skewness tends to $N(0, 6/n)$ as $n \to \infty$.

56• Kurtosis

56.1• Population kurtosis

— The heaviness of the tails of a distribution affects the behavior of many statistics.

— Hence it is useful to have a measure of tail heaviness.

— One such measure is the (population) kurtosis, which is usually defined as

$$\frac{E(X-\mu)^4}{\sigma^4}-3.$$

— Because the deviations are raised to the fourth power, positive and negative deviations make the same contribution, while large deviations are strongly emphasized.

— Because of the divisor σ^4, multiplying each value by a constant has no effect on kurtosis.

56.2• Sample kurtosis

— The sample kurtosis is defined by

$$\frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{n} \frac{(X_i - \bar{X})^4}{S^4} - \frac{3(n-1)^2}{(n-2)(n-3)}. \qquad (1.8)$$

— For a normal random sample, the sample kurtosis tends to $N(0, 24/n)$ as $n \to \infty$.

57• R function

```
CV.skewness.kurtosis <- function(x)
{ # Function name: CV.skewness.kurtosis(x)
  # -------------- Aims -------------------------------------
  # Aim 1: Compute coefficient of variation from (1.6)
  # Aim 2: Compute sample skewness from (1.7)
  # Aim 3: Compute sample kurtosis from (1.8)
  # -------------- Input ------------------------------------
  # x = an n x 1 vector
  # -------------- Output -----------------------------------
  # CV, skewness, kurtosis
  ############################################################
  n <- length(x)
  xbar <- mean(x)
  s <- sd(x)
  # ------ Calculate CV from (1.6)----------------------------
  CV <- 100*s/xbar
  # ------ Calculate skewness based on (1.7)------------------
  a <-  n/((n-1)*(n-2)*s^3)
  sk <- a * sum((x - xbar)^3)
  # ------ Calculate kurtosis based on (1.8)------------------
  a <- n*(n+1)/((n-1)*(n-2)*(n-3)*s^4)
  b <- 3*(n-1)^2/((n-2)*(n-3))
  ku <- a * sum((x - xbar)^4) - b
  resultM <- matrix(c(CV, sk, ku), nrow=3, byrow=F)
  rownames(resultM) <- c("Coefficient.variation",
                      "Sample.kewness", "Sample.kurtosis")
  colnames(resultM) <- c("  Estimate")
  return(resultM)
}**************************************************************
```

57.1• Demonstration

— We first generate 100 i.i.d. random variables from $N(\mu, \sigma^2)$ with $\mu = 0.1$ and $\sigma = 2$, and calculate their CV, skewness and kurtosis.

— We second generate 100 i.i.d. random variables from $U(0, 1)$, and calculate their CV, skewness and kurtosis.

```
==============================================================
> x <- rnorm(100, mean=0.1, sd=2)
> CV.skewness.kurtosis(x)
                          Estimate
Coefficient.variation 399.05298074
Sample.kewness          0.04303304
Sample.kurtosis        -0.25577514
--------------------------------------------------------------
> x <- runif(100)
> CV.skewness.kurtosis(x)
                        Estimate
Coefficient.variation 70.5550973
Sample.kewness         0.3915098
Sample.kurtosis       -1.0001535
**************************************************************
```

1.5 Sample mean/variance for frequency tables

58• The issue and aim

- When large data sets are organized into *frequency tables* or presented as *grouped data*, there are simple formulae to calculate the sample mean $\bar{X}$ and sample variance S^2.

59• The carex flacca data

- The following table shows the number of sedge plants, *carex flacca*, found in 800 sample quadrats in an ecological study of grasses. Each quadrat is $1m^2$.

Table 1.1 *The carex flacca data*

Plants/quadrat (a_j)	0	1	2	3	4	5	6	7	Total
Frequency (f_j)	268	316	135	61	15	3	1	1	800

60• The salmo gairdnerii data

- The following data were collected by randomly sampling a large population of rainbow trout, *salmo gairdnerii*. The variable of interest is weight (in lb).

a_j	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
f_j	2	1	4	7	13	15	20	24	7	9	2	4	2	110

61• The general case

- In general, we have following table

a_j	a_1	a_2	$\cdots$	a_m	Total
f_j	f_1	f_2	$\cdots$	f_m	n

- Note that

$$\begin{aligned} X_i &= a_1, \text{ when } i = 1, \ldots, f_1; \\ X_i &= a_2, \text{ when } i = f_1 + 1, \ldots, f_1 + f_2; \\ X_i &= a_3, \text{ when } i = f_1 + f_2 + 1, \ldots, f_1 + f_2 + f_3; \\ &\vdots \end{aligned}$$

$$
\begin{aligned}
X_i &= a_{m-1}, \text{ when } i = \sum_{j=1}^{m-2} f_j + 1, \ldots, \sum_{j=1}^{m-1} f_j, \\
X_i &= a_m, \text{ when } i = \sum_{j=1}^{m-1} f_j + 1, \ldots, n,
\end{aligned}
$$

where $n = \sum_{j=1}^{m} f_j$, we have

$$
\bar{X} = \frac{a_1 f_1 + \cdots + a_m f_m}{n} = \frac{\sum_{j=1}^{m} a_j f_j}{n} \quad \text{and} \tag{1.9}
$$

$$
\begin{aligned}
S^2 &= \frac{(a_1 - \bar{X})^2 f_1 + \cdots + (a_m - \bar{X})^2 f_m}{n-1} \\
&= \frac{\sum_{j=1}^{m} (a_j - \bar{X})^2 f_j}{n-1}.
\end{aligned} \tag{1.10}
$$

62• R function

```
function(a, f)
{ # Function name: mean.var.freq.table(a, f)
  # -------------- Aims --------------------------------------
  # Aim 1: Calculate sample mean     based on (1.9)
  # Aim 2: Calculate sample variance based on (1.10)
  # -------------- Input -------------------------------------
  # a = an m x 1 vector
  # f = an m x 1 vector
  # -------------- Output ------------------------------------
  # Sample mean and Sample variance
  ############################################################
  n <- sum(f)
  # ------ Calculate mean based on (1.9)----------------------
  xbar <-  c( t(a) %*% f/n )
  # ------ Calculate variance based on (1.10)------------------
  b <- (a - xbar)^2
  S2 <- c( t(b) %*% f/(n-1) )
  resultM <- matrix(c(xbar, S2), nrow=2, byrow=F)
  rownames(resultM) <- c("Sample.mean", "Sample.variance")
  colnames(resultM) <- c("  Estimate")
  return(resultM)
}*************************************************************
```

62.1• Demonstration

— Using the carex flacca data, we calculate $\bar{X}$ and S^2.

```
===============================================================
> a <- 0:7
> f <- c(268, 316, 135, 61, 15, 3, 1, 1)
> mean.var.freq.table(a, f)
                  Estimate
Sample.mean       1.071250
Sample.variance   1.110061
---------------------------------------------------------------
> a <- 1:13
> f <- c(2, 1, 4, 7, 13, 15, 20, 24, 7, 9, 2, 4, 2)
> mean.var.freq.table(a, f)
                  Estimate
Sample.mean       7.090909
Sample.variance   5.753128
***************************************************************
```

1.6 The effect of coding data

63• The issue of coding data

- While grouping data can save considerable time and effort, coding data may also offer similar savings.
- Coding involves conversion of measurements or statistics into easier to work with values by simple arithmetic operations.
- It is sometimes used to change units or investigate experimental effects.

64• Additive coding

- Additive coding involves the addition or subtraction of a constant a from each observation in a data set.
- The coded sample mean

$$\bar{X}_c = \frac{\sum_{i=1}^{n}(X_i + a)}{n} = \bar{X} + a,$$

while the coded sample variance is unchanged since

$$S_c^2 = \frac{\sum_{i=1}^n [(X_i + a) - (\bar{X} + a)]^2}{n-1} = S^2.$$

65• Multiplicative coding

- Multiplicative coding involves the multiplying or dividing each observation in a data set by a constant a.

- The new mean is a times the old mean because

$$\bar{X}_c = \frac{\sum_{i=1}^n aX_i}{n} = a\bar{X},$$

and the new variance is a^2 times the old variance because

$$S_c^2 = \frac{\sum_{i=1}^n (aX_i - a\bar{X})^2}{n-1} = a^2 S^2.$$

1.7 Specialized disciplines

66• Specialized statistics

- Statistical techniques are used in a wide range of types of scientific and social research.

- Some fields of inquiry use applied statistics so extensively that they have specialized terminology.

- These disciplines include:

— Actuarial science (assesses risk in insurance and finance);

— Applied information economics;

— Astrostatistics (statistical evaluation of astronomical data);

— Biostatistics;

— Business statistics;

— Chemometrics (statistical analysis of data from chemistry);

— Computational biology;

— Computational sociology;

- — Data mining (applying statistics & *Pattern Recognition* to discover knowledge from data);
- — Data science;
- — Demography;
- — Econometrics (statistical analysis of economic data);
- — Energy statistics;
- — Engineering statistics;
- — Epidemiology (statistical analysis of disease);
- — Geography and *Geographic Information Systems*, specifically in *Spatial Analysis*;
- — Image processing;
- — Medical statistics;
- — Political science;
- — Psychological statistics;
- — Reliability engineering;
- — Sabermetrics (statistical analysis of baseball)
- — Social statistics;
- — Statistical mechanics;
- — Statistical process control.

67• Statistical own research fields

- In addition, there are particular types of statistical analysis that have also developed their own specialised terminology and methodology:
  - — Bayesian statistics;
  - — Bootstrap/Jackknife resampling;
  - — Multivariate statistics;
  - — Statistical classification;
  - — Structured data analysis (statistics);
  - — Structural equation modelling;
  - — Survey methodology;
  - — Survival analysis.

Chapter 2

Basic Graphics

1• Aim of this chapter

- In this chapter, we will introduce the following important built-in R functions in drawing basic graphics.

— `hist()` computes and plots a histogram for a given data set.

— `curve()` draws a curve corresponding to a function or density.

— `plot()` is a generic function for plotting of R objects.

— `ecdf()` draws an empirical cumulative distribution function.

— `qqnorm()` produces a normal QQ plot while `qqline()` adds a line on the plot.

— `boxplot()`, `barplot()`, `dotchart()` and `pie()` produce a box plot, a bar plot, a dot chart and a pie chart, respectively.

2• Two data sets to be used in this chapter

```
function(ind)
{ # Function name: data2(ind)
  # -------------- Aims --------------------------------------
  # Aim 1: Output data set 1
  # Aim 2: Output data set 2
  # -------------- Input -------------------------------------
  # ind = 1: Produce the data set 1
  # ind = 2: Produce the data set 2
  # -------------- Output ------------------------------------
  # x:       a vector
  # age.acc: a vector
  ############################################################
  if (ind == 1) {
    x1 <- c(200, 200, 202, 204, 206, 197, 199, 200, 204, 195)
    x2 <- c(193, 196, 200, 195, 202, 199, 202, 200, 206, 197)
    x3 <- c(198, 203, 201, 198, 198, 200, 205, 205, 206, 200)
    x4 <- c(203, 201, 198, 202, 206, 205, 207, 196, 199, 199)
    x5 <- c(196, 205, 203, 201, 200, 191, 199, 200, 193, 200)
    x  <- c(x1, x2, x3, x4, x5)
    return(x)
  }
  if (ind == 2) {
    mid.age <- c(2.5,7.5,13,16.5,17.5,19,22.5,44.5,70.5)
    acc.count <- c(28, 46, 58, 20, 31, 64, 149, 316, 103)
    age.acc <- rep(mid.age, acc.count)
    return(age.acc)
  }
}*************************************************************
```

2.1 Graphical display of distributions

2.1.1 Histograms

We can get a reasonable impression of the shape of a distribution by drawing a histogram; that is, a count of how many observations fall within specified division ("bins") of the x-axis.

3• Usage of hist()

3.1• The syntax

— The built-in R function hist() computes a histogram of a given data set.

— The syntax is as follows:

```
==============================================================
hist(x, freq = [NULL, T, F], prob = !freq,
     breaks = NULL,
     main = paste("Histogram of" , xname),
     xlim = range(breaks), ylim = NULL,
     xlab = xname, ylab = NULL,
     axes = TRUE, plot = TRUE, labels = FALSE, ...)
**************************************************************
```

3.2• The usage of freq

```
==============================================================
> x <- data2(ind == 1)
> par(mfrow=c(1, 2)) # multiframe, rowwise, 1 x 2 layout
> hist(x, freq=T, col="red")      # Figure 2.1 --- left plot
> hist(x, prob=T, col="blue")     # Figure 2.1 --- right plot
**************************************************************
```

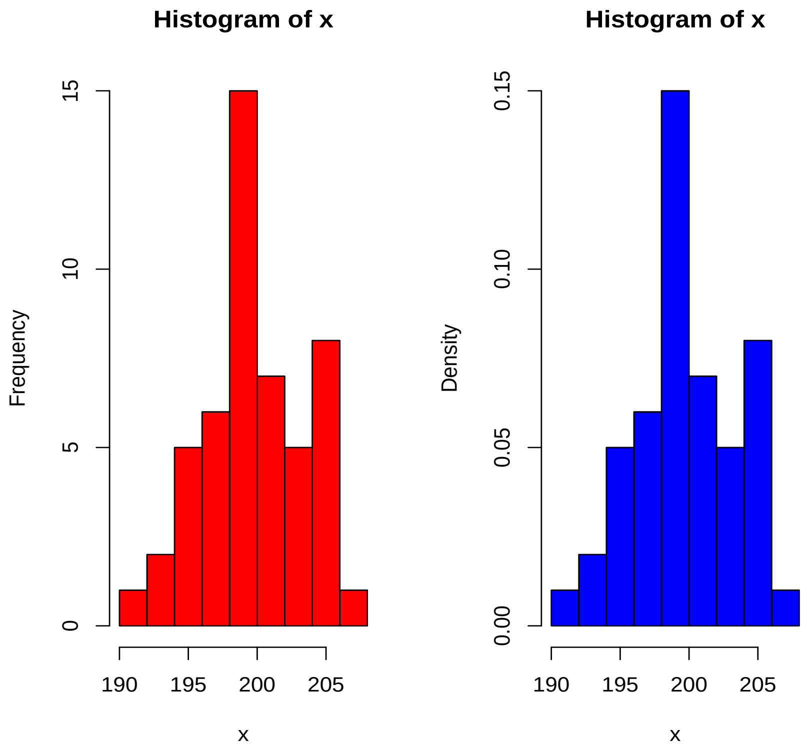

Figure 2.1 Histogram with two forms: left plot — frequency and right plot — density.

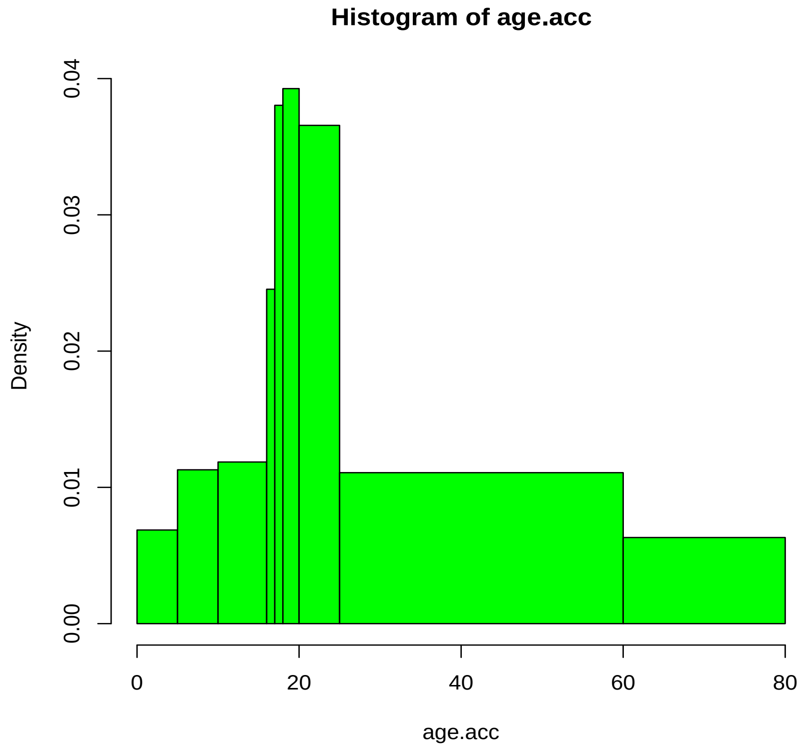

Figure 2.2 Histogram with unequal divisions.

3.3• Comments on Figure 2.1

— The left plot of Figure 2.1 is the display of hist(x), which is equivalent to hist(x, freq=T) by default.

— In fact, hist(x, freq=T) = hist(x, prob=F).

— By setting prob=T, we can get densities displayed, where the y-axis is in density units so that the total area of the histogram will be 1.

3.4• The usage of breaks

— By specifying breaks=n, we get *approximately* n bars in the histogram.

— We can have full control over the interval divisions by specifying breaks as a vector.

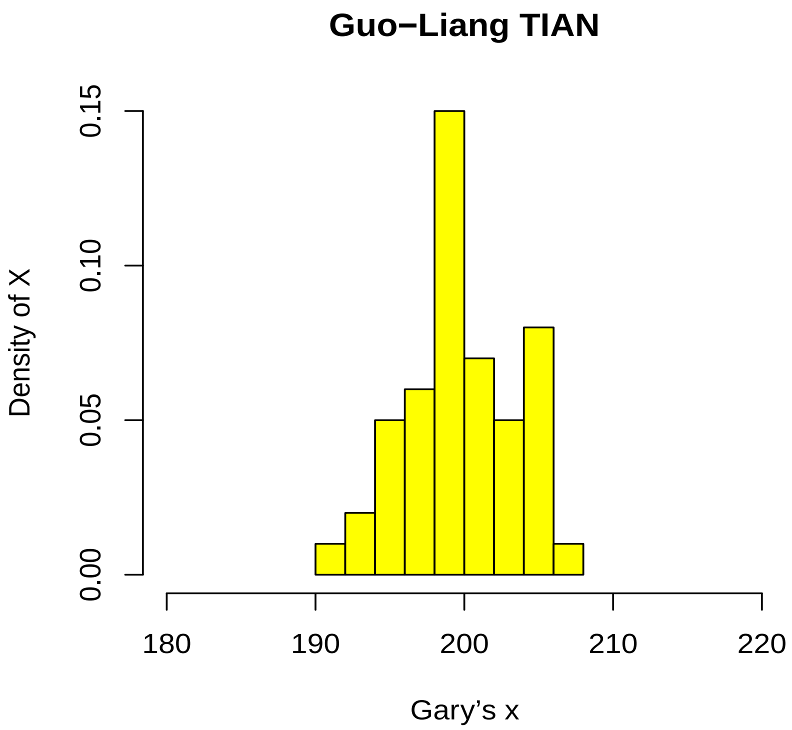

Figure 2.3 A general histogram.

3.5• An example

— Altman (*Practical Statistics for Medical Research*, Chapman & Hall, London, 1991, pp.25–26) contains an example of accident rates by age group.

— These are given as a count in age groups 0–4, 5–9, 10–15, 16, 17, 18–19, 20-24, 25-59, and 60–79 years of age.

— The corresponding histogram is given by Figure 2.2.

```
==============================================================
> age.acc <- data2(ind=2)
> hist(age.acc, breaks=c(0,5,10,16,17,18,20,25,60,80),
                col="green")
**************************************************************
```

3.6• The general use of histogram.

```
==============================================================
> x <- data2(ind=1)
> hist(x, prob = T, main = "Guo-Liang TIAN", xlab= "Gary’s x",
       ylab = "Density of X", xlim = c(180, 220),
       ylim = c(0, 0.15), col="pink")  #  Figure 2.3
**************************************************************
```

4• Drawing a histogram with a density

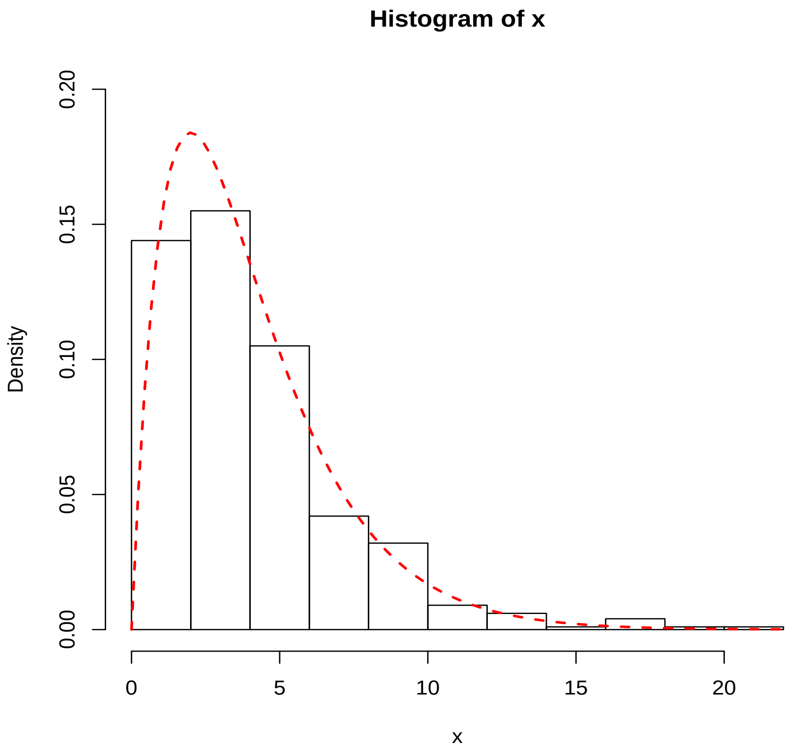

Figure 2.4 Plot of a histogram and the corresponding density curve from a sample of 500 i.i.d. $\chi^2(4)$.

```
function(ind)
{ # Function name: histogram.curve(ind)
  # -------------- Aim --------------------------------------
  # Draw a histogram and the density on the same figure
  # -------------- Input -----------------------------------
  # ind = 1
  ############################################################
  set.seed(14)
  x <- rchisq(500, df = 4)
  hist(x, freq = FALSE, ylim = c(0, 0.2))
  curve(dchisq(x, df=4), col = 2, lty = 2, lwd = 2, add=TRUE)
}
```

```
==============================================================
> histogram.curve(1)                               #  Figure 2.4
**************************************************************
```

2.1.2 Empirical cumulative distribution function

5• Definition of the empirical cdf

- Given the observations $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, the empirical distribution function is defined by

$$\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{(x_i \leqslant x)}, \tag{2.1}$$

  where we assume $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$.

6• Plotting an empirical cdf

```
==============================================================
> x <- rnorm(100, mean=0, sd=1)
> n <- length(x)
> plot(sort(x), (1:n)/n, type ="s", ylim = c(0, 1),
       xlab="x", ylab="F_n(x)", main="Empirical CDF")
**************************************************************
```

- The plotting parameter type ="s" gives a step function, where (x, y) is the left end of the steps.

- The empirical cdf is displayed in Figure 2.5.

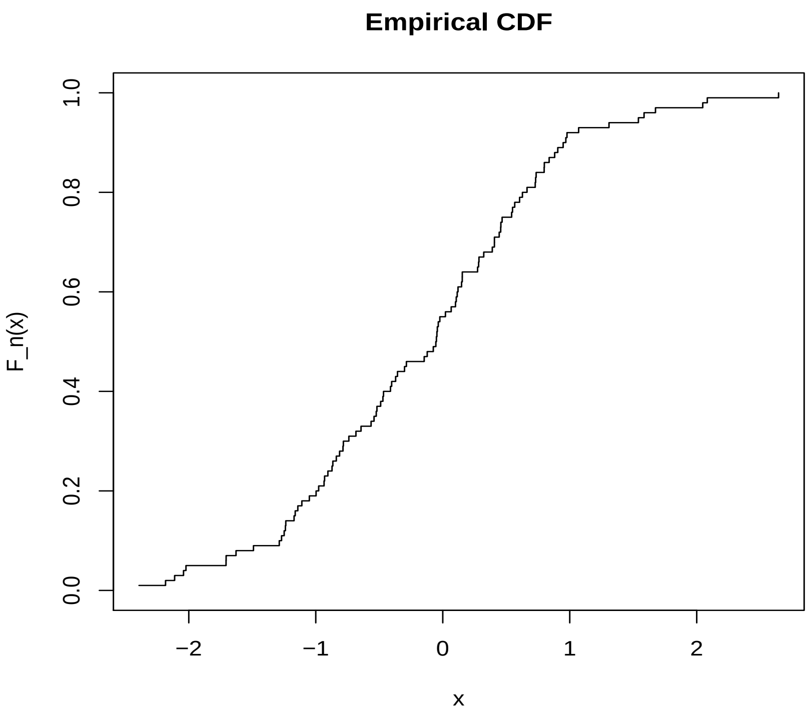

Figure 2.5 Empirical cdf from a sample of 100 i.i.d. $N(0, 1)$.

7• Advanced plotting technique by means of ecdf()

- The built-in R function `ecdf()` computes an empirical cdf.

```
empirical.cdf.and.plot <- function(x)
{ # -------------- Aims -------------------------------------
  # Aim 1: Plotting ecdf with points
  # Aim 2: Plotting ecdf with verticals but no points
  # Aim 3: Calculate min, Q1, median, mean, Q3, max
  # -------------- Input ------------------------------------
  # x = an n x 1 vector
  # -------------- Output -----------------------------------
  # Two plots and summary of Fn
  ############################################################
    par(mfrow = c(2, 1)); Fn <- ecdf(x)
    plot(Fn, main = "Empirical CDF")
    plot(Fn, main = "Empirical CDF", verticals = TRUE,
             do.points = FALSE)
    summary(Fn)
}
```

```
==============================================================
> empirical.cdf.and.plot(runif(10))
Empirical CDF:    10 unique values with summary
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.03596 0.27160 0.57040 0.49170 0.75120 0.77820
**************************************************************
```

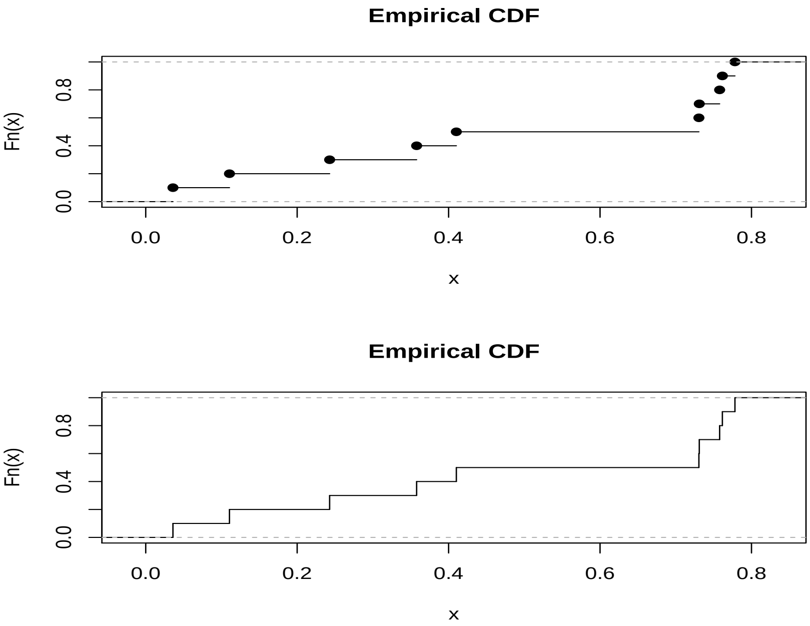

Figure 2.6 Empirical cdf from a sample of 10 i.i.d. $U(0,1)$.

2.1.3 P-P and Q-Q plots

8• Aim

- The *probability–probability* (P-P) plot and *quantile–quantile* (Q-Q) plot are used to compare two distributions or to assess whether data have a particular distribution.

8.1$^\bullet$ The difference between P-P and Q-Q plots

— A plot of points whose coordinates are the cumulative probabilities $\{p_x(q), p_y(q)\}$ for different values of q is a P-P plot.

— A plot of points whose coordinates are the quantiles $\{q_x(p), q_y(p)\}$ for different values of p is a Q-Q plot.

— Figure 2.7 may be used to describe each type.

— The P-P plots can immediately tell where the sample cdf fits good and bad, while the Q-Q plots can detect outliers better.

8.2$^\bullet$ qqnorm(): a normal Q-Q plot

— In particular, a normal Q-Q plot is used to check the assumption that a data set is from a normal distribution.

— A normal Q-Q plot involves plotting the ordered sample values $x_{(1)}, \ldots, x_{(n)}$ against the quantiles of a standard normal distribution, i.e., $\Phi^{-1}(p_i)$, where usually $p_i = (i - 0.5)/n$ and $\Phi(\cdot)$ is the cdf of $N(0, 1)$.

8.3$^\bullet$ qqline()

— If the resultant Q-Q plot appears linear, then the data are consistent with the assumption of normality.

— If the resultant Q-Q plot shows departures from linearity, such as a "S" shape or concavity, then, it suggests non-normality.

— The non-linearity indicates a need to do transformation.

8.4$^\bullet$ qqplot()

— The Q-Q plot is also used to assess whether two data sets have the same distribution.

— A plot with a "U" shape means that one distribution is skewed relative to the other.

— An "S" shape implies that one distribution has longer tails than the other.

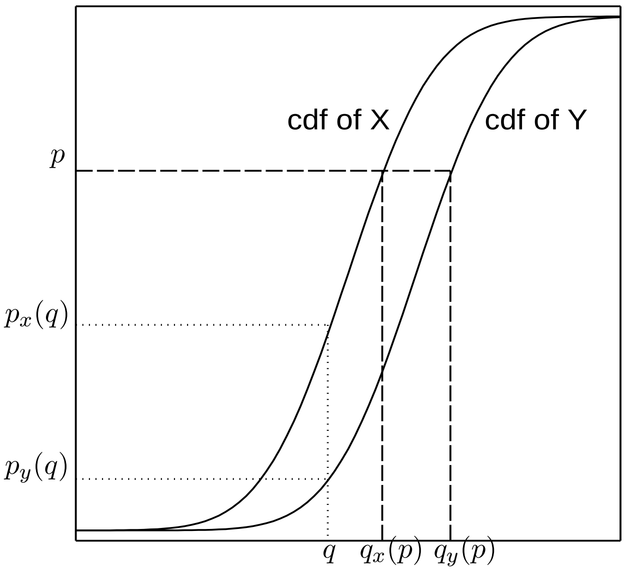

Figure 2.7 Illustration for P-P and Q-Q plots.

9$^\bullet$ The use of qqnorm(), qqline() and qqplot()

```
=============================================================
> x <- rnorm(200, mean=0, sd = 2)
> qqnorm(x); qqline(x, col = 2)                  # Figure 2.8
> y <- rchisq(500, df = 3)
> qqplot(x, y)                                   # Figure 2.9
**************************************************************
```

2.1.4 Boxplots

10$^\bullet$ Aim

- A *boxplot* or *box-and-whisker* plot is a graphical summary of a distribution.
- It is used to identify the observations with extreme values.
- It has the following explanations:
  - — The bottom and top edges of the box are located at the 25-th and 75-th percentiles, namely Q_1 and Q_3, respectively.

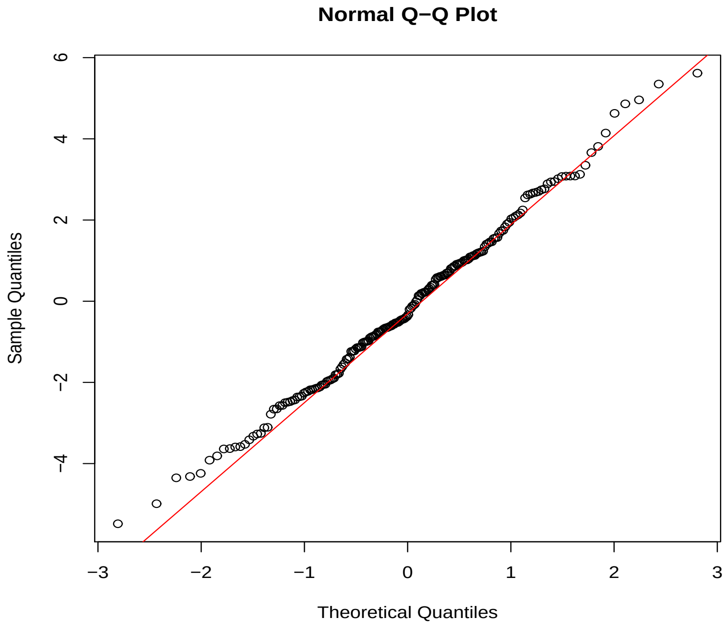

Figure 2.8 Test the normality by the Q-Q plot for a sample of 200 i.i.d. $N(0, 2^2)$.

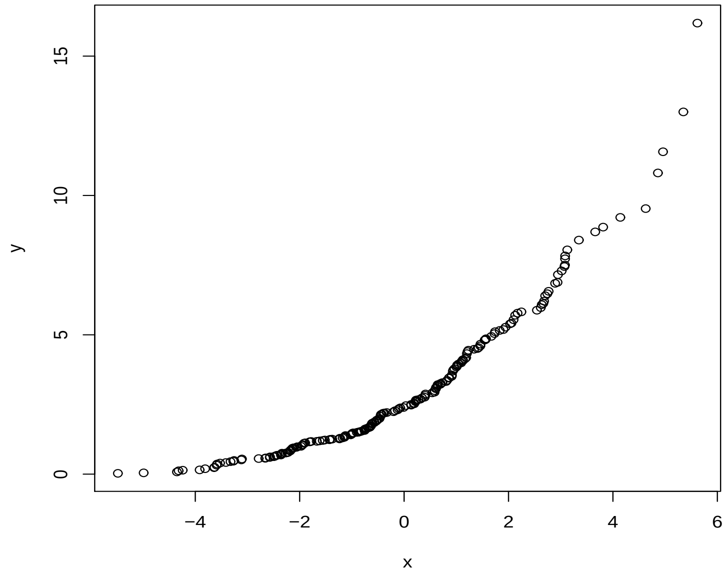

Figure 2.9 Test $H_0 : F_X(\cdot) = G_Y(\cdot)$ by the Q-Q plot, where $F_X(\cdot)$ is the cdf of a sample of 200 i.i.d. $N(0, 2^2)$ and $G_Y(\cdot)$ is the cdf of a sample of 500 i.i.d. $\chi^2(3)$.

— A center horizontal line is drawn at the sample median.

— Central vertical lines, called **whiskers**, extend *at most* 1.5 interquartile ranges beyond Q_1 and Q_3 as appropriate.

— Values more extreme but within 3 interquartile ranges of the box are marked as ‘0’.

11• The use of boxplot()

```
=============================================================
> x <- data2(1); y <- rchisq(100, df=5)
> par(mfrow=c(1, 2))
> boxplot(x, xlab = "x", col="red")
> boxplot(y, xlab = "y", col="blue")           # Figure 2.10
> par(mfrow=c(1, 1))
*************************************************************
```

- It is necessary to reset the layout parameter to `c(1, 1)` at the end, unless you also want two plots side by side subsequently.

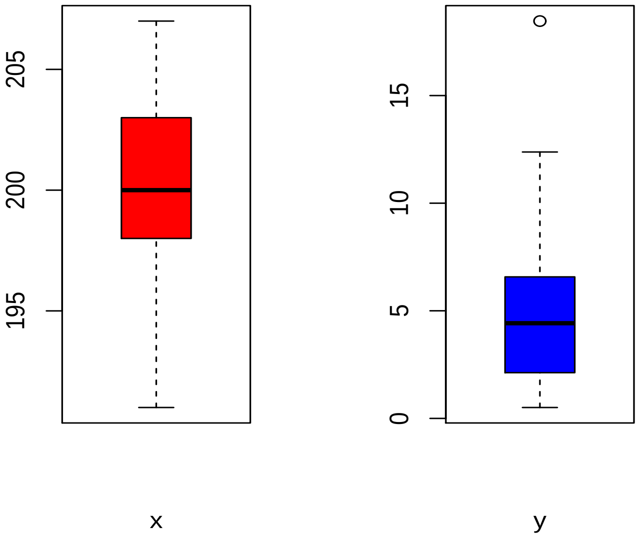

Figure 2.10 Boxplots for x and y.

2.2 Summary statistics by groups

12• Daily energy expenditure data

- Consider the following data set on energy expenditure for lean and obese women.

```
==============================================================
   expend stature
1    9.21   obese
2    7.53    lean
3    7.48    lean
4    8.08    lean
5    8.09    lean
6   10.15    lean
7    8.40    lean
8   10.88    lean
9    6.13    lean
10   7.90    lean
11  11.51   obese
12  12.79   obese
13   7.05    lean
14  11.85   obese
15   9.97   obese
16   7.48    lean
17   8.79   obese
18   9.69   obese
19   9.68   obese
20   7.58    lean
21   9.19   obese
22   8.11    lean
**************************************************************
```

13• Data entry

- To enter data into a blank data frame, we use

```
> d <- data.frame()
> fix(d)
```

- An alternative would be `d <- edit(data.frame())`.
- After the data input, we obtain

```
==============================================================
> d
   expend stature
1    9.21   obese
2    7.53    lean
3    7.48    lean
4    8.08    lean
5    8.09    lean
6   10.15    lean
7    8.40    lean
8   10.88    lean
9    6.13    lean
10   7.90    lean
11  11.51   obese
12  12.79   obese
13   7.05    lean
14  11.85   obese
15   9.97   obese
16   7.48    lean
17   8.79   obese
18   9.69   obese
19   9.68   obese
20   7.58    lean
21   9.19   obese
22   8.11    lean
**************************************************************
```

14• Calculation of summary statistics

- We can use `tapply` like this

```
==============================================================
> tapply(d$expend, d$stature, mean)
     lean     obese
 8.066154 10.297778
```

```
> tapply(d$expend, d$stature, median)
 lean obese
 7.90  9.69
> tapply(d$expend, d$stature, sd)
    lean    obese
1.238080 1.397871
> tapply(d$expend, d$stature, length)
 lean obese
   13     9
--------------------------------------------------------------
> xbar <- tapply(d$expend, d$stature, mean)
> s <- tapply(d$expend, d$stature, sd)
> n <- tapply(d$expend, d$stature, length)
> cbind(mean=xbar, std.dev=s, n=n)
           mean  std.dev  n
lean   8.066154 1.238080 13
obese 10.297778 1.397871  9
**************************************************************
```

15• Checking

- We can use the `split()` function to generate a list of vectors according to grouping

```
==============================================================
> L <- split(d$expend, d$stature)
> L
$lean
[1]  7.53  7.48  8.08  8.09 10.15  8.40 10.88  6.13
[9]  7.90  7.05  7.48  7.58  8.11
$obese
[1]  9.21 11.51 12.79 11.85  9.97  8.79  9.69  9.68  9.19
--------------------------------------------------------------
> mean(L$lean); sd(L$lean); length(L$lean)
[1] 8.066154
[1] 1.23808
[1] 13
> mean(L$obese); sd(L$obese); length(L$obese)
```

```
[1] 10.29778
[1] 1.397871
[1] 9
**************************************************************
```

2.3 Graphics for grouped data

2.3.1 Histograms

16• Data set in the form of data frame

- Assume that we want to compare two groups in the energy data frame introduced in §2.2 by plotting both histograms.
- We first need to separate the `expend` vector in the energy data frame into two vectors according to the value of the factor `stature`.

```
==============================================================
> energy <- d                                   # data frame
> attach(energy)
> expend
 [1]  9.21  7.53  7.48  8.08  8.09 10.15  8.40 10.88
 [9]  6.13  7.90 11.51 12.79  7.05 11.85  9.97  7.48
[17]  8.79  9.69  9.68  7.58  9.19  8.11
> stature
 [1] "obese" "lean"  "lean"  "lean"  "lean"  "lean"  "lean"
 [8] "lean"  "lean"  "lean"  "obese" "obese" "lean"  "obese"
[15] "obese" "lean"  "obese" "obese" "obese" "lean"  "obese"
[22] "lean"
> expend.lean <- expend[stature=="lean"]
> expend.obese <- expend[stature=="obese"]
--------------------------------------------------------------
> par(mfrow=c(2,1))
> hist(expend.lean, breaks=10, xlim=c(5,13),
+      ylim=c(0, 4), col="red")      # Figure 2.11 upper plot
> hist(expend.obese, breaks=10, xlim=c(5,13),
+      ylim=c(0, 4), col="blue")     # Figure 2.11 lower plot
**************************************************************
```

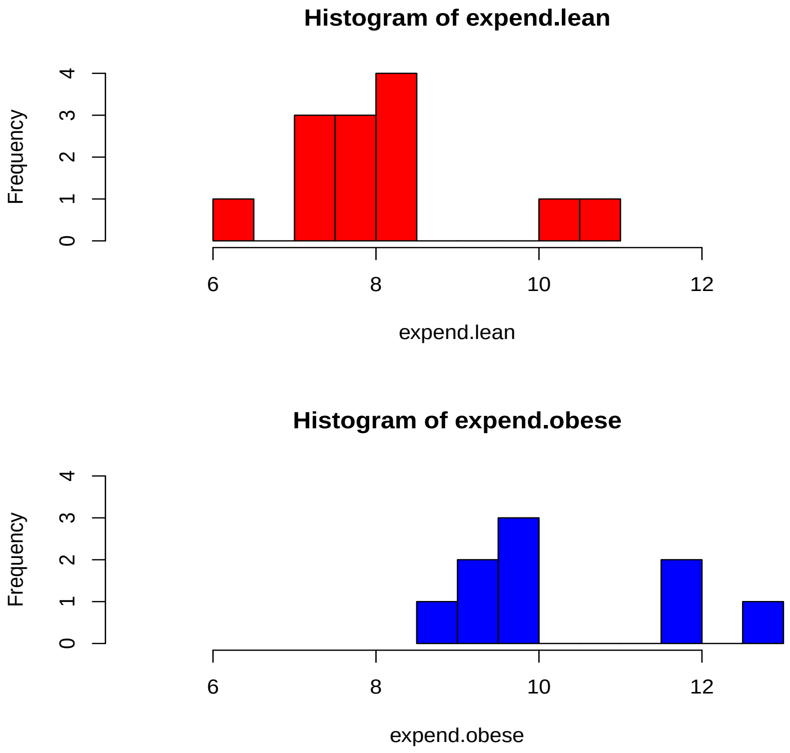

Figure 2.11 Histograms with refinements.

2.3.2 Parallel boxplots

If we want a set of boxplots from several groups in the same data frame, two equivalent ways are available.

```
================================================================
> boxplot(expend˜stature, col=c("green", "yellow"))
                                                # Figure 2.12
----------------------------------------------------------------
> expend.lean
 [1]  7.53  7.48  8.08  8.09 10.15  8.40 10.88
 [8]  6.13  7.90  7.05  7.48  7.58  8.11
> expend.obese
 [1]  9.21 11.51 12.79 11.85  9.97  8.79  9.69  9.68  9.19
> boxplot(expend.lean, expend.obese, col=c("green", "yellow"),
+                                  names=c("lean", "obese"))
****************************************************************
```

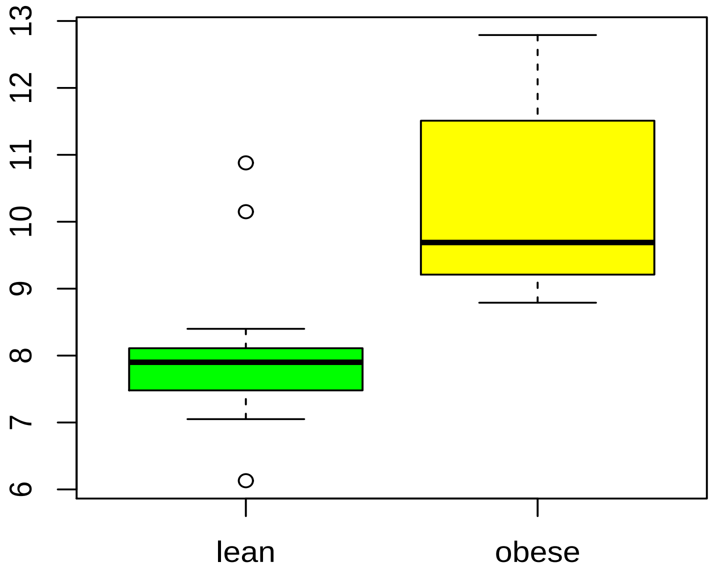

Figure 2.12 Parallel boxplots.

2.4 Generating tables

17• Two-way tables

- Categorical data are usually described in the form of tables.
- This section outline how we can create tables from your data.
- We deal mainly with two-way tables, which need to be in a `matrix` object.
- Altman (1991, pp.242) contains an example on caffeine consumption by marital status among women giving birth.
- That table may be input as follows.

```
==============================================================
> caff.marital <- matrix(c(652, 1537, 598, 242, 36, 46, 38,
+                  21, 218, 327, 106, 67), nrow=3, byrow=T)
> caff.marital
     [,1] [,2] [,3] [,4]
[1,]  652 1537  598  242
[2,]   36   46   38   21
[3,]  218  327  106   67
**************************************************************
```

- To get readable printouts, we can add row and column names to the matrix.

```
==============================================================
> colnames(caff.marital) <- c("0", "1-150", "151-300", ">300")
> rownames(caff.marital) <- c("Married", "Prev.married",
+                             "Single")
> caff.marital
               0 1-150 151-300 >300
Married      652  1537     598  242
Prev.married  36    46      38   21
Single       218   327     106   67
**************************************************************
```

2.5 Graphics display of tables

For presentation purpose, it may be desirable to display a graph rather than a table of counts or percentages.

2.5.1 Bar plots

18• Figure 2.13

- The following barplot() will produce Figure 2.13.

```
==============================================================
> total.caff <- margin.table(caff.marital, 2)
> total.caff
      0   1-150 151-300    >300
```

```
    906    1910     742     330
> barplot(total.caff, col=c("red", "blue", "green", "black"),
          ylim=c(0, 2000))                    #  Figure 2.13
**************************************************************
```

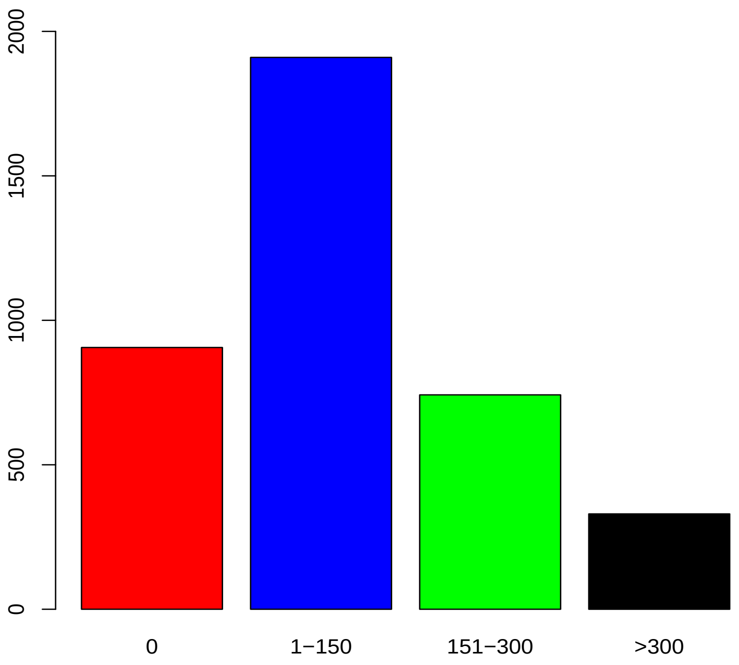

Figure 2.13 Simple bar plots of total caffeine consumption.

19• Figure 2.14

- The following barplot() will produce Figure 2.14.

```
==============================================================
> A <- t(caff.marital)
> A
        Married Prev.married Single
0           652           36    218
1-150      1537           46    327
```

```
151-300     598           38    106
>300        242           21     67
--------------------------------------------------------------
> B <- prop.table(A, 2)
> B
           Married Prev.married     Single
0       0.21525256    0.2553191 0.30362117
1-150   0.50742819    0.3262411 0.45543175
151-300 0.19742489    0.2695035 0.14763231
>300    0.07989435    0.1489362 0.09331476
--------------------------------------------------------------
> c(652, 1537, 598, 242)/sum(c(652, 1537, 598, 242))
[1] 0.21525256 0.50742819 0.19742489 0.07989435
--------------------------------------------------------------
> barplot(B, beside=T, legend.text= colnames(caff.marital),
+  col=c("red", "blue", "green", "black"), ylim=c(0, 0.8))
**************************************************************
```

2.5.2 Dotcharts

The Cleveland dotcharts, named after William S. Cleveland (1994), can be employed to study a table from both sides at the same time. They contain the same information as bar plots with beside=T but give quite a different visual impression.

```
> dotchart(A)  # Figure 2.15
```

2.5.3 Pie charts

```
==============================================================
> opar <- par(mfrow=c(2,2), mex=0.8, mar = c(1, 1, 2, 1))
> slices <- c("red", "blue", "green", "black")
> pie(caff.marital["Married",], main = "Married", col=slices)
> pie(caff.marital["Prev.married",], main =
+                  "Previously married", col=slices)
> pie(caff.marital["Single",], main = "Single", col=slices)
> par(opar)                                    # Figure 2.16
**************************************************************
```

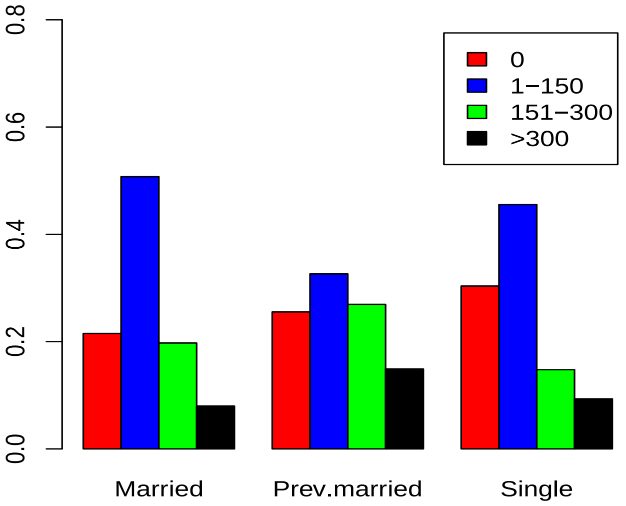

Figure 2.14 Bar plots with specified colours and legend.

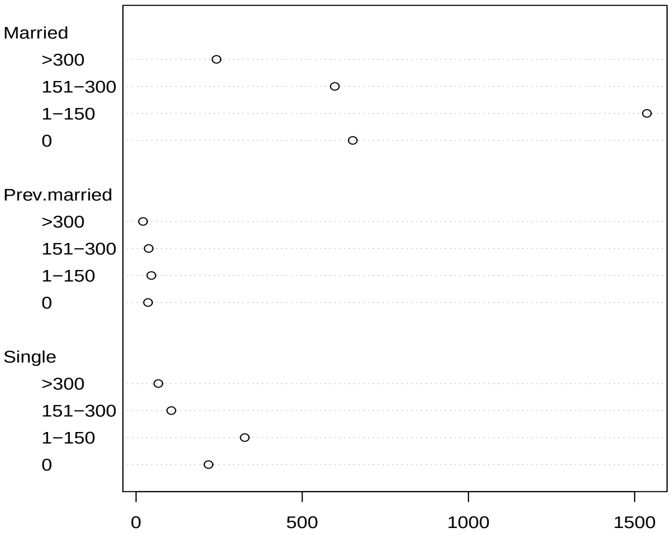

Figure 2.15 Dotchart of caffeine consumption.

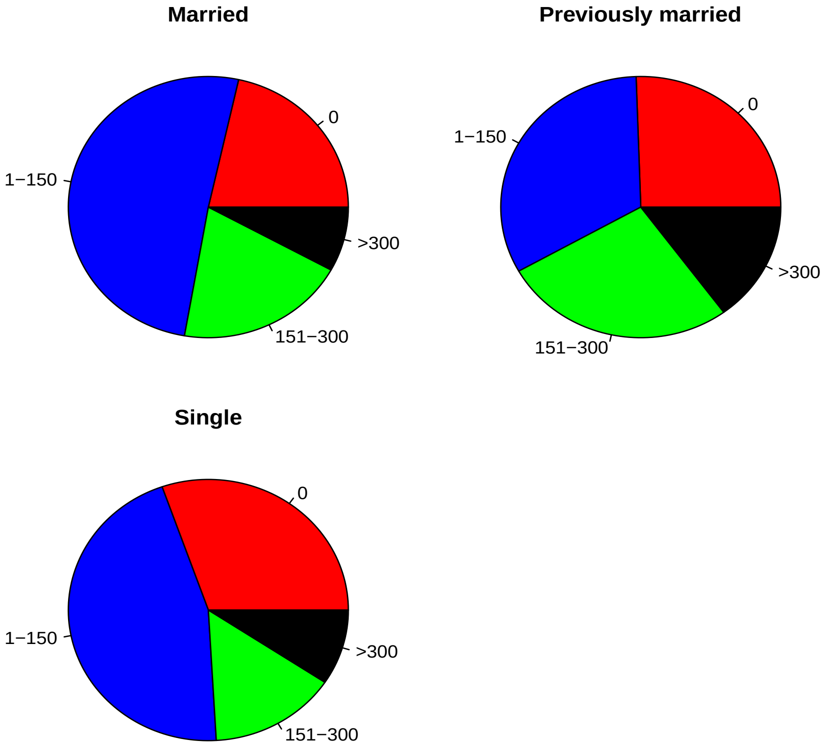

Figure 2.16 Pie charts of caffeine consumption according to marital status.

Chapter 3

One- and Two-sample Tests for Continuous Data

1• Aims in statistical aspects

- In this chapter, we consider one- and two-sample problems for continuous data with *valid normality assumption* by considering hypotheses testing and confidence interval estimation for parameters of interest.

- When the normality assumption is violated, some *distribution-free methods* are employed.

2• Aims in software aspects

2.1• Introduction of four R functions

— `t.test()` for t tests;

— `binom.test()` for the exact binomial test;

— `sign.test()` for the sign test;

— `wilcox.test()` for the Wilcoxon signed-rank test.

— Both `t.test()` and `wilcox.test()` can be applied to one- and two-sample problems as well as paired data.

2.2• Two R functions for testing the normality assumption

— `shapiro.test()` for one-sample Shapiro–Wilk test,

— `ks.test()` for one- and two-sample Kolmogorov–Smirnov tests.

3.1 The one-sample t test, sign test and Wilcoxon signed-rank test

3.1.1 The one-sample t test

$3^{\bullet}$ Statistical issue

- The one-sample t test is used to test the null hypothesis that the mean of a *normal* population is equal to a pre-specified constant.
- That is, H_0: $\mu = \mu_0$ against one of the three alternatives: $\mu > \mu_0$, $\mu < \mu_0$ or $\mu \neq \mu_0$.

$4^{\bullet}$ Assumptions

- Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} N(\mu, \sigma^2)$.
- Let $x_1, \ldots, x_n$ denote the realizations of $X_1, \ldots, X_n$.

$5^{\bullet}$ Test statistic and t value

- The test statistic and t value are given by

$$T = \frac{\bar{X} - \mu_0}{\sqrt{S^2/n}} \quad \text{and} \quad t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}, \tag{3.1}$$

respectively, where

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \quad \text{and} \quad S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2,$$

$\bar{x}$ and s^2 are the sample mean and the sample variance.

- The *standard deviation* (SD) is defined by

$$s = \sqrt{\textstyle\sum_{i=1}^{n}(x_i - \bar{x})^2/(n-1)}. \tag{3.2}$$

- The *standard error of the mean* (SEM) is defined by SEM $= \sigma/\sqrt{n}$. Hence, $s/\sqrt{n}$ is the estimate of SEM.

$6^{\bullet}$ p-value

- Under H_0, $T \sim t(n-1)$.
- The corresponding p-values are

$$\begin{aligned} p\text{-value} &= \Pr(T > t), & \text{if } H_1\colon\ \mu > \mu_0, && (3.3)\\ p\text{-value} &= \Pr(T < t), & \text{if } H_1\colon\ \mu < \mu_0, && (3.4)\\ p\text{-value} &= 2\Pr(T > |t|), & \text{if } H_1\colon\ \mu \neq \mu_0. && (3.5) \end{aligned}$$

- When p-value $\geqslant \alpha$ (in general, $\alpha = 0.05$), we cannot reject the H_0.
- The corresponding R codes are as follows:

```
==============================================================
> n <- length(x); xbar <- mean(x);
> s <- sd(x); SEM <- s/sqrt(n)
> t <- (xbar - mu0)/SEM                         # c.f. (3.1)
> p.larger <- 1 - pt(t, df=n-1)                 # c.f. (3.3)
> p.smaller <- pt(t, df=n-1)                    # c.f. (3.4)
> p.value <- 2*( 1 - pt(abs(t), df=n-1) )       # c.f. (3.5)
**************************************************************
```

7• Confidence intervals

- A $(1-\alpha)100\%$ two-sided CI, lower one-sided CI and upper one-sided CI for μ are given by

$$\begin{aligned} &[\bar{x} - t(\alpha/2, n-1)s/\sqrt{n},\ \bar{x} + t(\alpha/2, n-1)s/\sqrt{n}\,], && (3.6)\\ &(-\infty,\ \bar{x} + t(\alpha, n-1)s/\sqrt{n}\,], \quad \text{and} \\ &[\bar{x} - t(\alpha, n-1)s/\sqrt{n},\ +\infty), \end{aligned}$$

  respectively, where $t(\alpha, n-1)$ denotes the upper α quantile of the $t(n-1)$ distribution.
- The corresponding R codes are as follows:

```
==============================================================
> muL <- xbar - qt(1-alpha/2, df=n-1)*SEM
> muU <- xbar + qt(1-alpha/2, df=n-1)*SEM
> mu2 <- xbar + qt(1-alpha, df=n-1)*SEM
> mu1 <- xbar - qt(1-alpha, df=n-1)*SEM
**************************************************************
```

8• One-sample tests for normality assumption

8.1• Shapiro–wilk test

— This test calculates a W statistic that tests

$$H_0\text{: } \boldsymbol{x} = (x_1, \dots, x_n)^\top \text{ comes from a normal distribution.}$$

— Small value of W concludes that the distribution is not normal.

— Percentage points for the W statistic, obtained via Monte Carlo simulations, were reproduced by Pearson and Hartley (1972, Table 16, Biometrica Tables for Statisticians, Vol.2).

— This test has done very well in comparison studies with other goodness of fit tests.

— The W statistic is defined by

$$W = \frac{(\sum_{i=1}^n a_i x_{(i)})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \tag{3.7}$$

where the $\{x_{(i)}\}$ are the ordered sample values and the $\{a_i\}$ are constants generated from the means, variances and covariances of the order statistics of a sample of size n from a normal distribution (see Pearson and Hartley (1972, Table 15)).

— For more information about the Shapiro-Wilk test the reader is referred to the original Shapiro and Wilk (1965, Biometrika) paper and the tables in Pearson and Hartley (1972).

— R uses `shapiro.test(x)` to test the normality.

8.2• Kolmogorov–Smirnov test

— The one-sample Kolmogorov–Smirnov test statistic (D) is based on the difference between the empirical cdf and null cdf.

— For example, to test whether $\boldsymbol{x} = (x_1, \dots, x_n)^\top$ comes from the $N(\bar{x}, s^2)$, we need to calculate

$$D = D_n = \max_x |\hat{F}_n(x) - \Phi(x; \bar{x}, s^2)|, \tag{3.8}$$

where $\hat{F}_n(x)$ is the empirical cdf of $\boldsymbol{x}$ defined by (2.1) and $\Phi(x; \mu, \sigma^2)$ is the cdf of $N(\mu, \sigma^2)$.

— Large value of D implies that the distribution is not normal.

— To test the normality, R uses

```
ks.test(x, "pnorm", mean(x), sd(x))
```

9• Example 3.1 (Daily intake data)

- Suppose that we wish to compare the average dietary intake of a particular group of individuals with the recommended daily intake (Altman, 1991, p.183).
- The average daily energy intake (kJ) over 10 days in 11 healthy women aged 22–30 is as follows:

  5260, 5470, 5640, 6180, 6390, 6515, 6805, 7515, 7515, 8230, 8770.

  — What can we say about the energy intake of these women in relation to a recommended daily intake of 7725 kJ?

9.1• Formulation into a statistical problem

— Assume that these data come from a normal distribution.

— The aim is to test whether this distribution might have mean $\mu = \mu_0 = 7725$.

— We first test the normality assumption and then perform a two-sided t test.

9.2• Visualization via graphics

```
daily.intake.plots <- function(ind, x)
{ # -------------- Aims --------------------------------------
  # Drawing boxplots, Q-Q plot, histogram & density
  # -------------- Input -------------------------------------
  # ind=1: boxplot
  # ind=2: Q-Q plot
  # ind=3: histogram & density
  # x    : An n x 1 vector
  ############################################################
  daily.intake <- x
  di.bar <- mean(daily.intake)
  di.std <- sd(daily.intake)
  di.range <- range(daily.intake)
  if (ind == 1) {
     boxplot(daily.intake, xlab= "daily.intake", col= "blue")
  #  boxplot(daily.intake, horizontal=T) # horizontal boxplot
  } else if (ind == 2) {
     qqnorm(daily.intake); qqline(daily.intake, col= 2)
  } else
     x <- seq(di.range[1], di.range[2], length= 100)
     y <- dnorm(x, mean= di.bar, sd= di.std)
     hist(daily.intake, xlab= "daily.intake (kJ)",
          prob= T, col= "grey")
     lines(x, y, lwd= 2, col= "red")
}**************************************************************
```

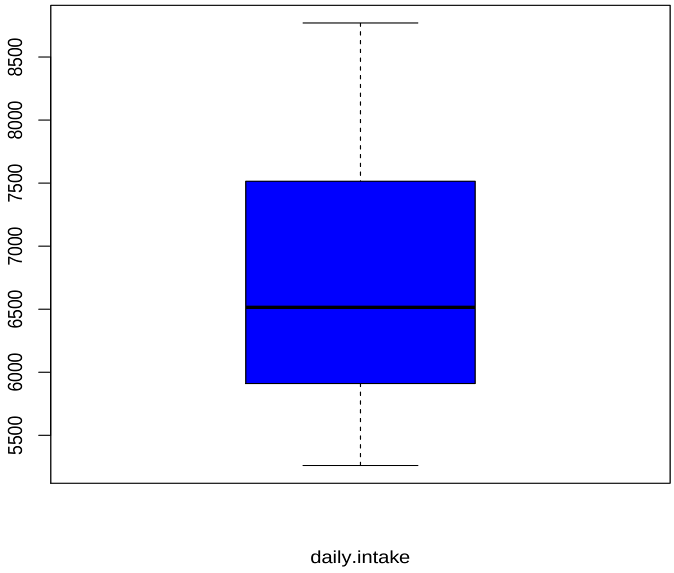

Figure 3.1 Boxplot for daily.intake.

```
==============================================================
> x <- c(5260, 5470, 5640, 6180, 6390, 6515,
+        6805, 7515, 7515, 8230, 8770)
> daily.intake.plots(ind=1, x)                  # Figure 3.1
> daily.intake.plots(ind=2, x)                  # Figure 3.2
> daily.intake.plots(ind=3, x)                  # Figure 3.3
***************************************************************
```

9.3• Writing a universal R function

```
daily.intake <- function(ind, x, mu0, alpha)
{ # Function name: daily.intake(ind, x, mu0, alpha)
  # -------------- Aims ------------------------------------
  # Aim 1: Calculate summary statistics
  # Aim 2: Perform Shapiro-Wilk test for normality
  # Aim 3: Perform Kolmogorov-Smirnov test for normality
  # Aim 4: Perform two-sided t test
  # Aim 5: Perform one-sided t test with H_1: mu > mu_0
  # Aim 6: Perform one-sided t test with H_1: mu < mu_0
  # -------------- Input -----------------------------------
  # ind=1: Aim 1
  # ind=2: Aim 2
  # ind=3: Aim 3
  # ind=4: Aim 4
  # ind=5: Aim 5
  # ind=6: Aim 6
  # x    : An n x 1 vector
  # mu0  : 7725
  # alpha: 0.05
  # -------------- Output ----------------------------------
  # result = list(mean, std, quantile)
  ############################################################
  xbar <- mean(x)
  s <- sd(x)
  quan <- quantile(x)
  if (ind == 1) {
     result <- list(mean= xbar, std= s, quantile= quan)
     return(result)
```

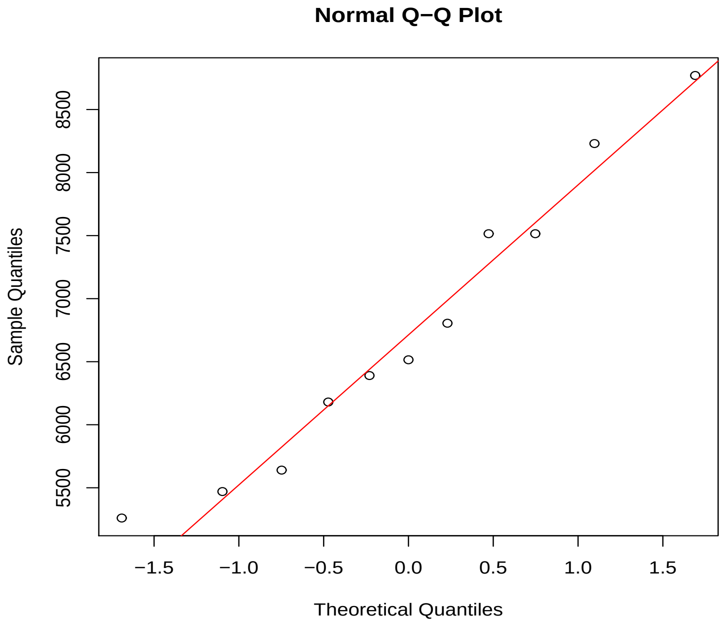

Figure 3.2 Q-Q plot for daily.intake.

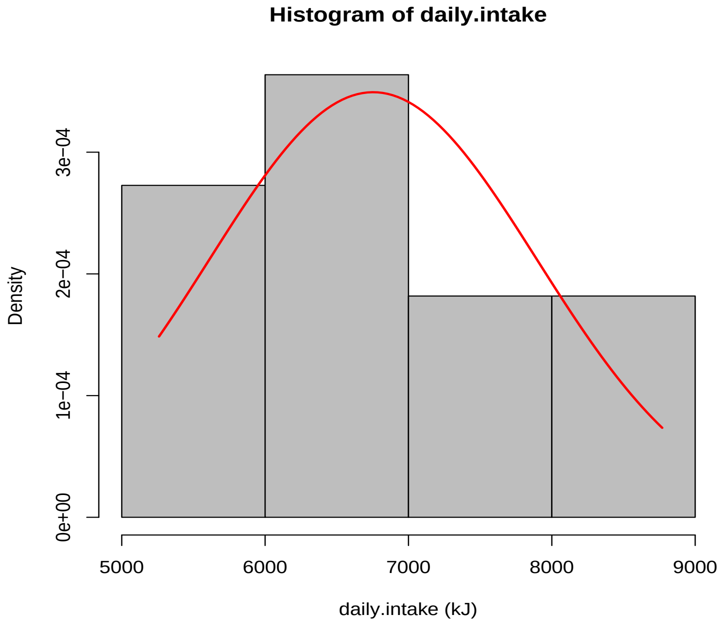

Figure 3.3 Histogram for daily.intake with overlaid normal density $N(\hat{\mu}, \hat{\sigma}^2)$, where $\hat{\mu} = 6753.636$ and $\hat{\sigma} = 1142.123$.

```
  } else if (ind == 2) {
     shapiro.test(x)
  } else if (ind == 3) {
     ks.test(x, "pnorm", xbar, s)
  } else if (ind == 4) {
     t.test(x, mu=mu0, alt = "two.sided", conf.level=1-alpha)
  } else if (ind == 5) {
     t.test(x, mu=mu0, alt = "greater", conf.level=1-alpha)
  } else
     t.test(x, mu=mu0, alt = "less", conf.level=1-alpha)
}*************************************************************
```

9.3.1• Running daily.intake()

```
==============================================================
> x <- c(5260, 5470, 5640, 6180, 6390, 6515,
+        6805, 7515, 7515, 8230, 8770)
> daily.intake(ind=1, x, mu0=7725, alpha=0.05)
$mean
[1] 6753.636

$std
[1] 1142.123

$quantile
  0%  25%  50%  75% 100%
5260 5910 6515 7515 8770
--------------------------------------------------------------
> daily.intake(ind=2, x, mu0=7725, alpha=0.05)

        Shapiro-Wilk normality test

data:  x
W = 0.95237, p-value = 0.6743
--------------------------------------------------------------
> daily.intake(ind=3, x, mu0=7725, alpha=0.05)

        One-sample Kolmogorov-Smirnov test
```

```
data:  x
D = 0.12821, p-value = 0.9936
alternative hypothesis: two-sided

Warning message:
In ks.test(x, "pnorm", xbar, s) :
  ties should not be present for the Kolmogorov-Smirnov test
--------------------------------------------------------------
> daily.intake(ind=4, x, mu0=7725, alpha=0.05)

        One Sample t-test

data:  x
t = -2.8208, df = 10, p-value = 0.01814
alternative hypothesis: true mean is not equal to 7725
95 percent confidence interval:
 5986.348 7520.925
sample estimates:
mean of x
 6753.636
--------------------------------------------------------------
> daily.intake(ind=6, x, mu0=7725, alpha=0.05)

        One Sample t-test

data:  x
t = -2.8208, df = 10, p-value = 0.009069
alternative hypothesis: true mean is less than 7725
95 percent confidence interval:
     -Inf 7377.781
sample estimates:
mean of x
 6753.636
**************************************************************
```

9.4$^\bullet$ Interpretations and conclusions

— First, we first calculate the sample mean (6753.636), standard deviation (1142.123), minimum (5260), median (6515) and maximum (8770).

— Second, we perform the Shapiro–Wilk normality test with test statistic $W = 0.95237$ and p-value $= 0.6743$. Since $0.6743 \gg 0.05$, we cannot reject the null hypothesis that the data come from a normal distribution.

— Third, we perform the one-sample Kolmogorov–Smirnov normality test with test statistic $D = 0.12821$ and p-value $= 0.9936$. Since $0.9936 \gg 0.05$, we cannot reject the null hypothesis that the data come from a normal distribution.

— Fourth, we perform a one-sample two-sided t test for testing H_0: $\mu = 7725$ versus H_1: $\mu \neq 7725$. The t-value is -2.8208. Since p-value $= 0.01814 < 0.05$, we should reject the H_0 at 0.05 level of significance. The 95% CI of μ is [5986.348, 7520.925].

— Finally, if H_1 is replaced by H_1': $\mu < 7725$, then the corresponding p-value $= 0.009069 < 0.05$, we should reject the H_0 at 0.05 level of significance. The 95% lower CI of μ is $(-\infty, 7377.781]$.

3.1.2 Central limit theorem

10• Non-normal populations with large sample sizes

- In practice, when the normality assumption is violated, the one-sample t test cannot be used.

- For a non-normal population with large sample sizes, fortunately, we have *Central limit theorem* (CLT) which states: If we have a random sample $X_1, \ldots, X_n$ drawn from a population with mean μ and variance $\sigma^2 < \infty$, then

$$\frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} \dot{\sim} N(0, 1) \quad \text{as} \quad n \to \infty. \tag{3.9}$$

- In other words, if the sample does not come from a normal population, the sample mean $\bar{X}$ is still asymptotically normally distributed provided that the sample size n is large enough.

11• Question on large sample sizes

- The question is: How large is the 'large'?
- We do not have a definite answer.
- It depends on how non-normal the population is.
- For moderately non-normal populations, $n > 10$ may be enough; for highly skewed population, $n > 30$ may be enough.

3.1.3 The sign test and Wilcoxon signed-rank test

12• Background

- For non-normal populations (e.g., data are skewed) and small sample sizes where the one-sample t test or the CLT cannot be applied, we can make inference on the *location* (usually, being *median*) rather than the *mean* by using the non-parametric tests (or distribution-free tests) for a single sample: the sign test and the Wilcoxon signed-rank test.
- The p-value for the one-sample t test is given by `t.test()`.
- The sign test and the Wilcoxon signed-rank test are provided by `binom.test()` and `wilcox.test()`, respectively.

(a) The sign test

13• Statistical issue

- The sign test is often used as a non-parametric alternative to the one-sample t test.
- For the sign test, we only assume that the population is continuous.
- We would like to test the null hypothesis

$$H_0\colon\ \tilde{\mu} = \tilde{\mu}_0, \tag{3.10}$$

  against one of the three alternatives: $\tilde{\mu} > \tilde{\mu}_0$, $\tilde{\mu} < \tilde{\mu}_0$ or $\tilde{\mu} \neq \tilde{\mu}_0$, where $\tilde{\mu}$ is the population *location parameter* (e.g., median or mean) and $\tilde{\mu}_0$ is a pre-specified constant.

14• Statistical procedure

- In the sign test, let $\boldsymbol{x} = (x_1, \dots, x_n)^\top$.
- We first replace each x_i exceeding $\tilde{\mu}_0$ with a plus sign $(+)$ and each value less than $\tilde{\mu}_0$ with a minus sign $(-)$.
- If a sample value equals $\tilde{\mu}_0$, we simply discard it.
- We then test an equivalent null hypothesis that the number of plus signs (denoted by x) is a value of a random variable X having the binomial distribution with the parameter n (the total number of plus signs and minus signs) and $p = 0.5$, i.e., $X \sim \text{Binomial}(n, p)$. Hence, the original null hypothesis H_0 specified by (3.12) is equivalent to

$$H_0'\colon\ p = 0.5. \tag{3.11}$$

- The two-sided alternative $\tilde{\mu} \neq \tilde{\mu}_0$ thus becomes $p \neq 0.5$, and the one-sided alternatives $\tilde{\mu} > \tilde{\mu}_0$ and $\tilde{\mu} < \tilde{\mu}_0$ become $p > 0.5$ and $p < 0.5$, respectively.
- When the sample size is small, the p-value can be obtained by computing the binomial probabilities.
- When the sample size is large, the p-value can be obtained by using the normal approximation to the binomial distribution.

15• EXAMPLE 3.2 (Cotton breaking strength data)

- The following are measurements of the breaking strength of a certain kind of 2-inch cotton ribbon in pounds:

163	165	160	189	161	171	**158**	**151**	169	162
163	**139**	172	165	**148**	166	172	163	187	173

- Use the sign test to test the null hypothesis H_0: $\tilde{\mu} = 160$ against the alternative hypothesis H_1: $\tilde{\mu} > 160$ at the 0.05 level of significance.

15.1• Solution by means of direct calculation

— Use the test statistic X (i.e., the number of plus signs), we have $X \sim \text{Binomial}(n, p)$.

— Replacing each value exceeding 160 with a plus sign (+), each value less than 160 with a minus sign (−), and discarding the one value that equals 160, we get

$$+ + + + + - - + + + - + + - + + + + +$$

so $x = 15$ and $n = 19$.

— Under H_0 (or H_0'), we have $X \sim \text{Binomial}(n, 0.5)$. Thus,

$$\begin{aligned}
p\text{-value} &= \Pr(X \geqslant x | H_0 \text{ is true}) \\
&= \Pr(X \geqslant 15) \\
&= \sum_{y \geqslant 15} \Pr(X = y) \\
&= \sum_{y \geqslant 15} \binom{n}{y} 0.5^y (1 - 0.5)^{n-y} \\
&= \sum_{y \geqslant 15} \binom{19}{y} 0.5^n \\
&= \left[\binom{19}{15} + \binom{19}{16} + \binom{19}{17} + \binom{19}{18} + \binom{19}{19} \right] 0.5^{19} \\
&= 0.0096.
\end{aligned}$$

— Since the p-value is less than 0.05, the null hypothesis must be rejected, and we conclude that the mean breaking strength of the given kind of ribbon exceeds 160 pounds.

15.2$^\bullet$ Solution by using `binom.test()` given the reduced data (x, n)

— Alternatively, we can perform an exact binomial test by binom.test() for testing H_0' specified by (3.13).

```
===============================================================
> binom.test(x= 15, n= 19, p= 0.5, alt= "g", conf.level= 0.95)

        Exact binomial test

data:  15 and 19
number of successes= 15, number of trials= 19, p-value= 0.0096
```

```
alternative hypothesis: true probability of success is greater
than 0.5
95 percent confidence interval:
 0.5808798 1.0000000
sample estimates:
probability of success
             0.7894737
**************************************************************
```

15.3$^\bullet$ Solution by using sign.test() with the raw data

```
sign.test <- function(y, mu0, ALT, alpha)
{ # Function name: sign.test(y, mu0, ALT, alpha)
  # -------------- Aim ---------------------------------------
  # Perform an exact binomial test with the raw data
  # -------------- Input -------------------------------------
  # y    : an m x 1 vector
  # mu0  : a pre-specified constant for the location
  # ALT  : c("two.sided", "less", "greater")
  # alpha: 0.05
  ############################################################
  y_mu0 <- y - mu0
  zs <- y_mu0[y_mu0>0]; zf <- y_mu0[y_mu0<0]
  s <- length(zs); f <- length(zf)
  binom.test(x= s, n= s+f, p= 0.5, alt = ALT,
                     conf.level= 1- alpha)
}*************************************************************
```

```
==============================================================
> y<- c(163, 165, 160, 189, 161, 171, 158, 151, 169, 162,
+       163, 139, 172, 165, 148, 166, 172, 163, 187, 173)
> sign.test(y, mu0=160, ALT="greater", alpha=0.05)

        Exact binomial test

data:  s and s + f
number of successes= 15, number of trials= 19, p-value= 0.0096
```

```
H_1: true probability of success is greater
than 0.5
95 percent confidence interval:
 0.5808798 1.0000000
sample estimates:
probability of success
             0.7894737
**************************************************************
```

(b) The Wilcoxon signed-rank test

16• Why this test?

- Although the sign test is easy to perform, it tends to be wasteful of information since it utilizes only the signs of the differences between the observations and $\tilde{\mu}_0$.
- An alternative nonparametric test, *the Wilcoxon signed-rank test*, is less wasteful in the sense that it takes into account also the magnitudes of the differences.

17• Statistical procedure

- In the Wilcoxon signed-rank test, let $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$.
- We first rank the differences $\{x_i - \tilde{\mu}_0\}_{i=1}^n$ without regard to their signs, assigning

  — rank 1 to the smallest difference in absolute value,

  — rank 2 to the 2-nd smallest difference in absolute value,

  — $\ldots$,

  — rank n to the largest difference in absolute value.

- Zero differences are discarded.
- If the absolute values of two or more differences are the same, we assign each one the mean of the ranks that they jointly occupy.
- Then, the Wilcoxon signed-rank test is based on

— V^+, the sum of the ranks assigned to the positive differences;

— V^-, the sum of the ranks assigned to the negative differences;

— $V^+ - V^-$; or

— $V = \min(V^+, V^-)$.

- Since

$$V^+ + V^- = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}, \tag{3.12}$$

  the resulting tests are all equivalent.

- When the sample size is small (say, $n < 15$), the distribution of the test statistic can be calculated exactly, at least in principle.

- When the sample size is large (for example, $n \geqslant 15$), it is reasonable to assume that V^+ (or V^-) is a value of a random variable having approximately a normal distribution with mean and variance given by

$$\mu = \frac{n(n+1)}{4} \quad \text{and} \quad \sigma^2 = \frac{n(n+1)(2n+1)}{24}. \tag{3.13}$$

- Therefore, the p-value can be obtained by using this normal approximation.

18• Example 3.1 (Revisited)

- Practical application of the Wilcoxon signed-rank test is done almost exactly as the t test.

```
==============================================================
> x
 [1] 5260 5470 5640 6180 6390 6515 6805 7515 7515 8230 8770
> x-7725
 [1] -2465 -2255 -2085 -1545 -1335 -1210 -920 -210 -210
       505  1045
> abs(x-7725)
 [1] 2465 2255 2085 1545 1335 1210  920  210  210  505 1045
> rank(abs(x-7725))
 [1] 11.0 10.0  9.0  8.0  7.0  6.0  4.0  1.5  1.5  3.0  5.0
--------------------------------------------------------------
> wilcox.test(x, mu=7725)
```

```
        Wilcoxon signed rank test with continuity correction

data:  x
V = 8, p-value = 0.0293
alternative hypothesis: true location is not equal to 7725

Warning message:
Cannot compute exact p-value with ties in:
wilcox.test.default(x, mu = 7725)
*************************************************************
```

18.1• Conclusion

— We perform a two-sided Wilcoxon signed-rank test for testing H_0: $\tilde{\mu} = 7725$ versus H_1: $\tilde{\mu} \neq 7725$, where $\tilde{\mu}$ denotes the true location of the daily energy intake.

— The values of $V^+ = 3 + 5 = 8$ and $V^- = 11 * 12/2 - V^+ = 66 - 8 = 58$, so that the V-value $= \min(V^+, V^-) = 8$.

— Since the p-value $= 0.0293 < 0.05$, we should reject the H_0 at 0.05 level of significance.

— There is no confidence limits for the location parameter in a nonparametric test.

3.2 The paired *t* test, paired sign test and paired Wilcoxon test

3.2.1 The paired *t* test

19• Aim and background

- The paired t test provides a hypothesis test of the difference between population means for a pair of random samples whose differences are *approximately normally distributed.*

- Subjects are often tested in a *before-after* situation or with subjects as alike as possible (e.g., two hands, two ears, twins).

- For example, the weights of subjects before and after participating a certain diet program.

20• Notation and assumption

- Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be a random sample of n paired observations.
- Define the differences $D_i = X_i - Y_i$ for $i = 1, \ldots, n$.
- Assume that $D_1, \ldots, D_n \overset{\text{iid}}{\sim} N(\mu_d, \sigma_d^2)$.

21• The p-value and CI

- The null hypothesis is that the mean difference is equal to some constant, i.e., $H_0 : \mu_d = \mu_0$.
- The paired t-test statistic and the t value are given by

$$T_{\rm d} = \frac{\bar{D} - \mu_0}{\sqrt{S_{\rm d}^2/n}} \quad \text{and} \quad t = \frac{\bar{d} - \mu_0}{s_d/\sqrt{n}}, \tag{3.14}$$

where $\bar{d} = (1/n)\sum_{i=1}^n d_i$ and $s_d^2 = \frac{1}{n-1}\sum_{i=1}^n (d_i - \bar{d})^2$.

- Under H_0, $T_{\rm d} \sim t(n-1)$. The corresponding p-values are given by

$$\begin{aligned} p\text{-value} &= \Pr(T_{\rm d} > t), & &\text{if } H_1\text{: } \mu_d > \mu_0, & (3.15)\\ p\text{-value} &= \Pr(T_{\rm d} < t), & &\text{if } H_1\text{: } \mu_d < \mu_0, & (3.16)\\ p\text{-value} &= 2\Pr(T_{\rm d} > |t|), & &\text{if } H_1\text{: } \mu_d \neq \mu_0. & (3.17)\end{aligned}$$

- A $(1-\alpha)100\%$ two-sided CI, lower one-sided CI and upper one-sided CI for μ_d are given by

$$[\bar{d} - t(\alpha/2, n-1)s_d/\sqrt{n}, \ \ \bar{d} + t(\alpha/2, n-1)s_d/\sqrt{n}], \tag{3.18}$$
$$(-\infty, \ \ \bar{d} + t(\alpha, n-1)s_d/\sqrt{n}], \quad \text{and}$$
$$[\bar{d} - t(\alpha, n-1)s_d/\sqrt{n}, \ \ +\infty),$$

respectively.

22• Paired versus unpaired t-test

- When studying about paired t-test and unpaired t-test (i.e., *two-independent-sample t-test*), the similarity between both is that both assume data from the *normal distribution*.

22.1• Characteristics of unpaired t-test

— The two groups taken should be independent.

— The sample size of the two groups need not be equal.

— It compares the mean of the data of the two groups.

— 95% confidence interval for the mean difference is calculated.

22.2• Characteristics of paired t-test

— The data are taken from subjects who have been measured twice.

— 95% CI is derived from the difference between the two sets of paired observations.

23• Example 3.3 (Paired daily intake data)

- The daily intake data in Example 3.1 come from a study in which the 11 women recorded their dietary intake for 60 consecutive days.
- They were unaware that the purpose of the study was to compare intake on the pre- and post-menstrual days of the menstrual cycle.

23.1• Paired data

— The data in Example 3.1 already analyzed were pre-menstrual dietary intakes.

— The following shows both the pre-menstrual and post-menstrual dietary intakes for one cycle for the same women.

Pre: 5260, 5470, 5640, 6180, 6390, 6515, 6805, 7515, 7515, 8230, 8770.
Post: 3910, 4220, 3885, 5160, 5645, 4680, 5265, 5975, 6790, 6900, 7335.

— Whether is there a significant difference between the pre- and post-menstrual dietary intakes for these women?

23.2$^\bullet$ Data entry. To enter data into a blank data frame, we use

```
==============================================================
> intake <- data.frame()
> fix(intake)
> intake
    pre post
1  5260 3910
2  5470 4220
3  5640 3885
4  6180 5160
5  6390 5645
6  6515 4680
7  6805 5265
8  7515 5795
9  7515 6790
10 8230 6900
11 8770 7335
> intake$pre
 [1] 5260 5470 5640 6180 6390 6515 6805 7515 7515 8230 8770
> intake$post
 [1] 3910 4220 3885 5160 5645 4680 5265 5795 6790 6900 7335
--------------------------------------------------------------
> attach(intake)
> pre
 [1] 5260 5470 5640 6180 6390 6515 6805 7515 7515 8230 8770
> post
 [1] 3910 4220 3885 5160 5645 4680 5265 5795 6790 6900 7335
**************************************************************
```

23.3$^\bullet$ Performing one-sample t-test

```
==============================================================
> post-pre
 [1] -1350 -1250 -1755 -1020  -745 -1835 -1540 -1720
      -725 -1330 -1435
> shapiro.test(post-pre)

        Shapiro-Wilk normality test
```

```
data:  post - pre
W = 0.9314, p-value = 0.4252
# p-value > 0.05, we cannot reject the normality assumption
-------------------------------------------------------------
> t.test(post-pre)

        One Sample t-test

data:  post - pre
t = -11.629, df = 10, p-value = 3.922e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -1592.945 -1080.691
sample estimates: mean of x is -1336.818
-------------------------------------------------------------
> t.test(post-pre, mu=0, alt="t", conf.level=0.95) # default

        One Sample t-test

data:  post - pre
t = -11.629, df = 10, p-value = 3.922e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -1592.945 -1080.691
sample estimates: mean of x is -1336.818
**************************************************************
```

23.4$^\bullet$ Performing paired *t*-test

```
==============================================================
> t.test(post, pre, paired=T)

        Paired t-test

data:  post and pre
t = -11.629, df = 10, p-value= 3.922e-07 #<0.05, reject H_0
alternative hypothesis: true difference in means is not = 0
```

```
95 percent confidence interval:
 -1592.945 -1080.691
sample estimates: mean of the differences is -1336.818
---------------------------------------------------------------
> t.test(pre, post, paired=T) # paired=T cannot be dropped

        Paired t-test

data:  pre and post
t = 11.629, df = 10, p-value = 3.922e-07
alternative hypothesis: true difference in means is not = 0
95 percent confidence interval:
 1080.691 1592.945
sample estimates: mean of the differences is 1336.818
---------------------------------------------------------------
> t.test(pre, post)  # Wrong!

        Welch Two Sample t-test

data:  pre and post
t = 2.6647, df = 19.934, p-value = 0.01491
alternative hypothesis: true difference in means is not =0
95 percent confidence interval:
  290.0983 2383.5381
sample estimates:
mean of x mean of y
 6753.636  5416.818
***************************************************************
```

3.2.2 The paired sign test

When the normality assumption is violated, we can apply the one-sample sign.test() to the differences between paired observations.

```
===============================================================
> sign.test(post-pre, mu0=0, ALT="t", alph=0.05)

        Exact binomial test
```

```
data:  s and s + f
number of successes = 0, number of trials = 11,
p-value = 0.0009766
alternative hypothesis: true probability of success is not=0.5
95 percent confidence interval:
 0.0000000 0.2849142
sample estimates:
probability of success
                     0
**************************************************************
```

3.2.3 The paired Wilcoxon test

When the normality is violated, we can also apply the `wilcox.test()` to paired data.

```
==============================================================
> wilcox.test(pre, post, paired=T)

        Wilcoxon signed rank test

data:  pre and post
V = 66, p-value = 0.0009766
alternative hypothesis: true location shift is not equal to 0
--------------------------------------------------------------
> wilcox.test(pre, post, paired=T, exact=F)

        Wilcoxon signed rank test with continuity correction

data:  pre and post
V = 66, p-value = 0.003857
alternative hypothesis: true location shift is not equal to 0
-------------------------------------------------------------
> wilcox.test(post, pre, paired=T)

        Wilcoxon signed rank test

data:  post and pre
V = 0, p-value = 0.0009766
```

```
alternative hypothesis: true location shift is not equal to 0
-------------------------------------------------------------
> wilcox.test(post-pre, mu=0, alt="t", conf.level=0.95)

        Wilcoxon signed rank test

data:  post - pre
V = 0, p-value = 0.0009766
alternative hypothesis: true location is not equal to 0
-------------------------------------------------------------
> wilcox.test(post-pre, mu=0, alt="t", exact=F)

        Wilcoxon signed rank test with continuity correction

data:  post - pre
V = 0, p-value = 0.003857
alternative hypothesis: true location is not equal to 0
**************************************************************
```

3.3 The two-sample t test and Wilcoxon test

3.3.1 Equal variances

24• Statistical issue

- The two-sample t test is used to test the null hypothesis that the means of two *independent normal* populations are identical.
- That is, H_0: $\mu_1 = \mu_2$ against one of the three alternatives: $\mu_1 > \mu_2$, $\mu_1 < \mu_2$ or $\mu_1 \neq \mu_2$.

25• Assumptions

- Let $X_{11}, \ldots, X_{n_1 1} \overset{\text{iid}}{\sim} N(\mu_1, \sigma_1^2)$ and $X_{12}, \ldots, X_{n_2 2} \overset{\text{iid}}{\sim} N(\mu_2, \sigma_2^2)$.
- The two samples are independent.
- Let x_{ij} denote the realization of X_{ij} for $i = 1, \ldots, n_j$ and $j = 1, 2$.

26• Test statistic and t value

- When $\sigma_1^2 = \sigma_2^2$, the test statistic and t value are given by

$$T_{\mathrm{p}} = \frac{\bar{X}_1 - \bar{X}_2}{S_p/\sqrt{n_{12}}} \quad \text{and} \quad t = \frac{\bar{x}_1 - \bar{x}_2}{s_p/\sqrt{n_{12}}}, \tag{3.19}$$

respectively, where

— $n_{12} \hat{=} n_1 n_2/(n_1 + n_2)$,

— $\bar{x}_j$ is the sample mean in group j, and

— s_p is the pooled standard deviation defined by

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{m - 1}}, \tag{3.20}$$

— s_j^2 is the sample variance in group j, and

— $m \hat{=} n_1 + n_2 - 1$.

27• p-value

- Under H_0, $T_{\mathrm{p}} \sim t(m - 1)$.

- The corresponding p-values are given by

$$\begin{aligned} p\text{-value} &= \Pr(T_{\mathrm{p}} > t), & &\text{if } H_1: \mu_1 > \mu_2, & (3.21)\\ p\text{-value} &= \Pr(T_{\mathrm{p}} < t), & &\text{if } H_1: \mu_1 < \mu_2, & (3.22)\\ p\text{-value} &= 2\Pr(T_{\mathrm{p}} > |t|), & &\text{if } H_1: \mu_1 \neq \mu_2. & (3.23) \end{aligned}$$

28• Confidence intervals

- A $(1 - \alpha)100\%$ two-sided CI, lower one-sided CI and upper one-sided CI for $\mu_1 - \mu_2$ are given by

$$\begin{aligned} &\bar{x}_1 - \bar{x}_2 \pm t(\alpha/2, m - 1)s_p/\sqrt{n_{12}}, \\ &(-\infty, \ \bar{x}_1 - \bar{x}_2 + t(\alpha, m - 1)s_p/\sqrt{n_{12}}\,], \quad \text{and} \\ &[\bar{x}_1 - \bar{x}_2 - t(\alpha, m - 1)s_p/\sqrt{n_{12}}, \ +\infty), \end{aligned} \tag{3.24}$$

respectively.

29$^\bullet$ Example 3.4 (Daily energy expenditure data)

- We return to the daily energy expenditure data (Section 2.3) and consider the problem of comparing energy expenditure between lean and obese women.

29.1$^\bullet$ When the data set is in the pattern of data.frame

```
==============================================================
> d <- data.frame()
> fix(d)
> d
   expend stature
1    9.21   obese
2    7.53    lean
3    7.48    lean
4    8.08    lean
5    8.09    lean
6   10.15    lean
7    8.40    lean
8   10.88    lean
9    6.13    lean
10   7.90    lean
11  11.51   obese
12  12.79   obese
13   7.05    lean
14  11.85   obese
15   9.97   obese
16   7.48    lean
17   8.79   obese
18   9.69   obese
19   9.68   obese
20   7.58    lean
21   9.19   obese
22   8.11    lean
**************************************************************
```

29.2$^\bullet$ Perform a two-sample t test

— The objective is to see whether there is a shift in level between the two groups, so we apply a t test as follows:

```
==============================================================
> t.test(expend˜stature, var.equal=T)
Error in eval(expr, envir, enclos) : cannot find the
                                     object ‘expend’
> attach(d)
> expend
 [1]  9.21  7.53  7.48  8.08  8.09 10.15  8.40 10.88  6.13
[10]  7.90 11.51 12.79  7.05 11.85  9.97  7.48  8.79  9.69
[19]  9.68  7.58  9.19  8.11
> stature
 [1] "obese" "lean"  "lean"  "lean"  "lean"  "lean"  "lean"
 [8]  "lean" "lean"  "lean"  "obese" "obese" "lean"  "obese"
[15] "obese" "lean"  "obese" "obese" "obese" "lean"  "obese"
[22]  "lean"
--------------------------------------------------------------
> L <- split(d$expend, d$stature)
# We can use the split() function to generate a list
# of vectors according to grouping.
> L
$lean
 [1]  7.53  7.48  8.08  8.09 10.15  8.40 10.88  6.13
 [9]  7.90  7.05  7.48  7.58  8.11

$obese
[1]  9.21 11.51 12.79 11.85  9.97  8.79  9.69  9.68  9.19
--------------------------------------------------------------
> shapiro.test(L$lean)

        Shapiro-Wilk normality test

data:  L$lean
W = 0.86733, p-value = 0.04818
# p-value < 0.05, we reject the normality assumption
--------------------------------------------------------------
> shapiro.test(L$obese)
```

```
        Shapiro-Wilk normality test

data:  L$obese
W = 0.87603, p-value = 0.1426
# p-value > 0.05, we cannot reject the normality assumption
**************************************************************
> t.test(expend~stature, var.equal=T)

        Two Sample t-test

data:  expend by stature
t = -3.9456, df = 20, p-value = 0.000799
alternative hypothesis: true difference in means is not 0
95 percent confidence interval of \mu_1 - \mu_2:
 -3.411451 -1.051796
sample estimates:
 mean in group lean      mean in group obese
           8.066154                10.297778
--------------------------------------------------------------
> t.test(L$lean, L$obese, var.equal=T) # Equivalent
**************************************************************
```

29.3• Interpretations of the output. The data in the lean group approximately follow a normal distribution. We further assume $\sigma_1^2 = \sigma_2^2$.

— `df` is equal to $m - 1 = n_1 + n_2 - 2 = 13 + 9 - 2 = 20$.

— p-value $\ll 0.05$, we reject H_0: $\mu_1 = \mu_2$ (i.e., accept H_1: $\mu_1 \neq \mu_2$) at the 0.05 significance level.

— 95% CI of $\mu_1 - \mu_2$ is $[-3.411451, -1.051796]$, not containing 0, which is in accordance with the p-value, indicating a significant difference at the 5% level.

— The MLEs of $\hat{\mu}_1 = 8.066154$ and $\hat{\mu}_2 = 10.297778$.

3.3.2 Testing the ratio of two variances

30• Background and aim

- The two-sample t test relies on the assumption that $\sigma_1^2 = \sigma_2^2$ or $\sigma_1^2/\sigma_2^2 = 1$.

- To verify the validity of this assumption for proper use of the two-sample t test, the R function `var.test()` provides the F test for ratio of two population variances being equal to 1.

31• F test

- The null and alternative hypotheses are

$$H_0^*\colon \frac{\sigma_1^2}{\sigma_2^2} = 1 \quad \text{against} \quad H_1^*\colon \frac{\sigma_1^2}{\sigma_2^2} \neq 1.$$

- Let $\nu_j = n_j - 1$ for $j = 1, 2$. Since $\sigma_2^2 S_1^2/(\sigma_1^2 S_2^2) \sim F(\nu_1, \nu_2)$, the test statistic and the corresponding f value are given by

$$F = \frac{S_1^2}{S_2^2} \quad \text{and} \quad f = \frac{s_1^2}{s_2^2}. \tag{3.25}$$

- Under H_0^*, $F \sim F(\nu_1, \nu_2)$. When p-value $\geqslant \alpha$, we cannot reject H_0^*.

32• Confidence interval

- A $(1-\alpha)100\%$ two-sided CI for σ_1^2/σ_2^2 is

$$\left[\frac{f}{f(\alpha/2; \nu_1, \nu_2)}, \quad \frac{f}{f(1-\alpha/2; \nu_1, \nu_2)}\right],$$

  where $f(\alpha; \nu_1, \nu_2)$ is the upper α quantile of $F(\nu_1, \nu_2)$ distribution.

- The corresponding R code is

```
==============================================================
> f <- var(x1)/var(x2)
> LB <- f/qf(1-alpha/2, length(x1)-1, length(x2)-1)
> UB <- f/qf(alpha/2, length(x1)-1, length(x2)-1)
**************************************************************
```

33• Example 3.4 (Revisited)

- Before we perform a two-sample t test, we need to check the equality of two variances.

```
==============================================================
> var.test(expend~stature) # = var.test(L$lean, L$obese)

        F test to compare two variances

data:  expend by stature
F = 0.78445, num df = 12, denom df = 8, p-value = 0.6797
H_1: true ratio of variances is not equal to 1
95 percent confidence interval for \sigma_1^2/\sigma_2^2:
 0.1867876 2.7547991
sample estimates:
ratio of variances
          0.784446
--------------------------------------------------------------
> f <- var(L$lean)/var(L$obese)
> f
[1] 0.784446
> f/qf(1-0.05/2, length(L$lean)-1, length(L$obese)-1)
[1] 0.1867876
> f/qf(0.05/2, length(L$lean)-1, length(L$obese)-1)
[1] 2.754799
--------------------------------------------------------------
> tapply(d$expend, d$stature, var)
    lean    obese
1.532842 1.954044
> 1.532842/1.954044
[1] 0.784446
**************************************************************
```

33.1• Interpretations of the output and comments

— Since the p-value $\gg 0.05$, we cannot reject H_0^*, i.e., $\sigma_1^2/\sigma_2^2 = 1$ is true.

— In addition, the CI of σ_1^2/σ_2^2 contains 1, indicating that H_0^* is true.

— The F test is based on the assumption that the groups are independent. We should not apply this test to paired data.

— Each group must follow normal distribution.

3.3.3 Unequal variances: Welch two-sample t test

34• Background

- When the null hypothesis H_0^*: $\sigma_1^2/\sigma_2^2 = 1$ was rejected, we should consider the case of unequal variances, which is known as the *Behrens–Fisher problem.*

35• Welch two-sample t test

- A modified version of the two-sample t test, called Welch two-sample t test or *Satterthwaite t test*, can be utilized.

35.1• Test statistic

— The test statistic and the corresponding t value are given by

$$T_{\mathrm{S}} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\dfrac{S_1^2}{n_1} + \dfrac{S_2^2}{n_2}}} \quad \text{and} \quad t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\dfrac{s_1^2}{n_1} + \dfrac{s_2^2}{n_2}}}. \tag{3.26}$$

35.2• Approximative null distribution

— Under the null hypothesis that $\mu_1 = \mu_2$, we have

$$T_{\mathrm{S}} \,\dot{\sim}\, t(\nu), \qquad \nu = \frac{\left(\dfrac{s_1^2}{n_1} + \dfrac{s_2^2}{n_2}\right)^2}{\left(\dfrac{s_1^2}{n_1}\right)^2 \dfrac{1}{n_1 - 1} + \left(\dfrac{s_2^2}{n_2}\right)^2 \dfrac{1}{n_2 - 1}}. \tag{3.27}$$

35.3• Proof of (3.27)

— The following proof was originally given by Welch (Biometrika, 1947, 28–35).

— Define $W = S_1^2/n_1 + S_2^2/n_2$. Since

$$\begin{aligned} W &= \frac{\sigma_1^2}{n_1(n_1-1)} \cdot \frac{(n_1-1)S_1^2}{\sigma_1^2} + \frac{\sigma_2^2}{n_2(n_2-1)} \cdot \frac{(n_2-1)S_2^2}{\sigma_2^2} \\ &\hat{=} a_1\chi_1^2 + a_2\chi_2^2 \end{aligned}$$

is a linear combination of two independent chi-square random variables, where $\chi_j^2 \sim \chi^2(\nu_j)$, $\nu_j = n_j - 1$, $j = 1, 2$, we could approximate W/m by a chi-square distribution with ν degrees of freedom, i.e.,

$$\frac{W}{m} \dot{\sim} \chi^2(\nu) \quad \text{or} \quad a_1\chi_1^2 + a_2\chi_2^2 \dot{\sim} m \cdot \chi^2(\nu). \tag{3.28}$$

— To determine the m and ν, let the corresponding means and variances in both sides of (3.28) be equal, i.e.,

$$a_1\nu_1 + a_2\nu_2 = m\nu \quad \text{and} \quad a_1^2 \cdot 2\nu_1 + a_2^2 \cdot 2\nu_2 = m^2 \cdot 2\nu. \tag{3.29}$$

— We obtain

$$m = \frac{a_1^2\nu_1 + a_2^2\nu_2}{a_1\nu_1 + a_2\nu_2}$$

and

$$\nu = \frac{(a_1\nu_1 + a_2\nu_2)^2}{a_1^2\nu_1 + a_2^2\nu_2} = \frac{\left(\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_2^2}{n_2}\right)^2}{\left(\dfrac{\sigma_1^2}{n_1}\right)^2 \dfrac{1}{n_1-1} + \left(\dfrac{\sigma_2^2}{n_2}\right)^2 \dfrac{1}{n_2-1}}. \tag{3.30}$$

— Under the null hypothesis that $\mu_1 = \mu_2$, we have

$$\begin{aligned} T_{\mathrm{s}} &= \frac{(\bar{X}_1 - \bar{X}_2)/\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}}{\sqrt{W}/\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}} \\ &= \frac{N(0,1)}{\sqrt{W/(a_1\nu_1 + a_2\nu_2)}} \overset{(3.29)}{=} \frac{N(0,1)}{\sqrt{\frac{W}{m}/\nu}} \dot{=} \frac{N(0,1)}{\sqrt{\chi^2(\nu)/\nu}} \sim t(\nu). \end{aligned}$$

— Finally, since ν is a function of both σ_1^2 and σ_2^2, we replace σ_j^2 in (3.30) by s_j^2 ($j = 1, 2$) and obtain the estimate of ν, denoted by ν. □

36• Example 3.4 (Revisited)

- When $\sigma_1^2 \neq \sigma_2^2$, we employ Welch two-sample t test as follows.

```
==============================================================
> t.test(expend˜stature)   # by default

        Welch Two Sample t-test

data:  expend by stature
t = -3.8555, df = 15.919, p-value = 0.001411
H_1: true difference in means is not equal to 0
95 percent confidence interval:
 -3.459167 -1.004081
sample estimates:
 mean in group lean      mean in group obese
           8.066154                10.297778
--------------------------------------------------------------
> t.test(L$lean, L$obese)  # Equivalent
**************************************************************
```

3.3.4 The two-sample Wilcoxon test

37• Different names of the test

- The two-sample Wilcoxon test is truly the non-parametric counterpart of the two-sample t-test.
  - To see this, one needs to recall that the two-sample t-test tests for equality of means when the underlying assumptions of normality and equality of variance are satisfied.
  - Thus the t-test tests if the two samples have been drawn from identical normal population.
  - The two-sample Wilcoxon test is its generalization.
- It is called the *Mann–Whitney (U) test*, or *Mann–Whitney–Wilcoxon test*.
- In R, it is called the non-parametric *Wilcoxon rank sum test* or *two-sample Wilcoxon test*.

38• Statistical issue

- Using the two-sample Wilcoxon test, we can decide whether the population distributions are identical *without* assuming them to follow normal distributions.

- The two-sample Wilcoxon test assumes that the observations are from *continuous* populations with distributions that are identical in shape, and differ only in location; i.e.,

$$H_0\colon\ F_1(x) = F_2(x) \quad \text{against} \quad H_1\colon\ F_1(x) = F_2(x+\delta) \text{ with } \delta \neq 0.$$

39• Statistical procedure

- Let $\boldsymbol{x} = (x_1, \ldots, x_{n_1})^\top$, $\boldsymbol{y} = (y_1, \ldots, y_{n_2})^\top$, and $n = n_1 + n_2$.
- All $\{x_i, y_j\}$ are ranked as if they were from a single sample.
- The sum of all ranks of n observations must be $n(n+1)/2$.
- We can use two alternative statistics, T_s and U.
- The statistic T_s (due to Wilcoxon) is the sum of the ranks in the *smaller group.*
- The statistic U (due to Mann and Whitney) is more complicated, being calculated as

$$U = n_1 n_2 + \frac{n_s(n_s+1)}{2} - T_s,$$

  where $n_s = \min(n_1, n_2)$.

- When $n_s \geqslant 10$, under H_0, the statistic $T_s \dot{\sim} N(\mu_s, \sigma_s^2)$, where

$$\mu_s = \frac{n_s(n+1)}{2} \quad \text{and} \quad \sigma_s^2 = \frac{n_1 n_2 (n+1)}{12}.$$

  — The z-value is $z_s = (T_s - \mu_s)/\sigma_s$ and p-value is $2\,\Pr(Z > |z_s|)$.

  — The corresponding R code is `2*( 1- pnorm(abs(zs)) )`.

- When $n_s < 10$, the z-value with continuity correction is

$$z_{s,\text{wcc}} = \frac{|T_s - \mu_s| - 0.5}{\sigma_s},$$

  and the p-value is $2\,\Pr(Z > |z_{s,\text{wcc}}|)$.

40• Example 3.4 (Revisited)

40.1• Solution by means of direct calculation

```
==============================================================
> x <- sort(L$lean)
> x
 [1]  6.13  7.05  7.48  7.48  7.53  7.58  7.90  8.08  8.09
[10]  8.11  8.40 10.15 10.88
> y <- sort(L$obese)
> y
[1]  8.79  9.19  9.21  9.68  9.69  9.97 11.51 11.85 12.79
> z <- c(x, y)
> z
 [1]  6.13  7.05  7.48  7.48  7.53  7.58  7.90  8.08  8.09
[10]  8.11  8.40 10.15 10.88
[14]  8.79  9.19  9.21  9.68  9.69  9.97 11.51 11.85 12.79
> R <- rank(z)
> R
 [1]  1.0    2.0   3.5   3.5   5.0   6.0   7.0   8.0   9.0
[10]  10.0  11.0  18.0  19.0
[14]  12.0  13.0  14.0  15.0  16.0  17.0  20.0  21.0  22.0
> sum(R)
[1] 253                                       # = n(n+1)/2
--------------------------------------------------------------
> n1 <- length(x); n2 <- length(y); n <- n1 + n2;
> n1
[1] 13
> n2
[1] 9
> n
[1] 22
> ns <- min(n1, n2)
> ns
[1] 9
> Ts <- sum(R[(n1+1):n])
> Ts
[1] 150
> U <- n1*n2 + ns*(ns+1)/2 - Ts
> U
[1] 12                              # = W in wilcox.test()
--------------------------------------------------------------
> mus <- ns*(n+1)/2; sigmas <- sqrt(n1*n2*(n+1)/12)
> zs.wcc <- (abs(Ts - mu) - 0.5)/sigma
> zs.wcc
[1] 3.071791
> p.value.wcc <- 2*(1 - pnorm(abs(zs.wcc)))
> p.value.wcc
[1] 0.002127789                    # < 0.05, we reject H_0
**************************************************************
```

40.2• Solution by using wilcox.test()

```
==============================================================
> wilcox.test(expend~stature) # =wilcox.test(L$lean, L$obese)

        Wilcoxon rank sum test with continuity correction

data:  expend by stature
W = 12, p-value = 0.002122
H_1: true location shift is not equal to 0

Warning message:
Cannot compute exact p-value with ties in
wilcox.test.default(...
**************************************************************
```

41• Two-sample Kolmogorov–Smirnov test

41.1• The procedure

— The two-sample Kolmogorov–Smirnov test is used to test whether two samples come from the same distribution. The procedure is very similar to the one-sample Kolmogorov–Smirnov test (see, Kolmogorov–Smirnov test for normality).

— Let $\boldsymbol{x} = (x_1, \ldots, x_{n_1})^\top$ be the first sample with the empirical cdf $\hat{F}_{1,n_1}(x)$ and $\boldsymbol{y} = (y_1, \ldots, y_{n_2})^\top$ be the second sample with the empirical cdf

$\hat{F}_{2,n_2}(x)$. Define

$$D_{n_1,n_2} = \max_x |\hat{F}_{1,n_1}(x) - \hat{F}_{2,n_2}(x)|.$$

— The null hypothesis is H_0: both samples come from a population with the same distribution. As for the Kolmogorov–Smirnov test for normality, we reject the null hypothesis (at significance level α) if $D_{n_1,n_2} > D_{n_1,n_2,\alpha}$, where $D_{n_1,n_2,\alpha}$ is the critical value.

— In R, we use `ks.text(x, y)`.

41.2• Demonstration

```
==============================================================
> ks.test(L$lean, L$obese)

        Two-sample Kolmogorov-Smirnov test

data:  L$lean and L$obese
D = 0.84615, p-value = 0.0009856
alternative hypothesis: two-sided

Warning message:
cannot compute correct p-values with ties in:
ks.test(L$lean, L$obese)
**************************************************************
```

3.3.5 A complete data analysis

42• Example 3.5 (Infant birthweight data)

- The following data set displays the birthweights (kg) of 50 infants with severe idiopathic respiratory distress syndrome (SIRDS).

- This is a serious condition that can result in death and did so in the case of 27 of these children.

- One question is whether the babies who died differed in birthweight from those who survived.

```
===============================================================
               Children who survived (n1 = 23)
1.130 1.575 1.680 1.760 1.930 2.015 2.090 2.600 2.700
2.950 3.160 3.400 3.640 2.830 1.410 1.715 1.720 2.040
2.200 2.400 2.550 2.570 3.005
---------------------------------------------------------------
               Children who died (n2 = 27)
1.050 1.175 1.230 1.310 1.500 1.600 1.720 1.750 1.770
2.275 2.500 1.030 1.100 1.185 1.225 1.262 1.295 1.300
1.550 1.820 1.890 1.940 2.200 2.270 2.440 2.560 2.730
***************************************************************
```

42.1$^{\bullet}$ Box plots

— As the first step to answering this question, we shall examine box plots of birthweight for each group.

— The birthweight box plots are shown in Figure 3.4.

```
===============================================================
> birthwt.surv <- c(1.130, 1.575, 1.680, 1.760, 1.930, 2.015,
+                   2.090, 2.600, 2.700, 2.950, 3.160, 3.400,
+                   3.640, 2.830, 1.410, 1.715, 1.720, 2.040,
+                   2.200, 2.400, 2.550, 2.570, 3.005)
> birthwt.died <- c(1.050, 1.175, 1.230, 1.310, 1.500, 1.600,
+                   1.720, 1.750, 1.770, 2.275, 2.500, 1.030,
+                   1.100, 1.185, 1.225, 1.262, 1.295, 1.300,
+                   1.550, 1.820, 1.890, 1.940, 2.200, 2.270,
+                   2.440, 2.560, 2.730)
> boxplot(birthwt.surv, birthwt.died, ylab="Birthweight (kg)",
  names=c("Baby survived", "Baby died"), col=c("red", "blue"))
***************************************************************
```

42.2$^{\bullet}$ Q-Q plots

— To perform a two-sample t test, we need to test the normality assumption by drawing two Q-Q plots as shown in Figure 3.5.

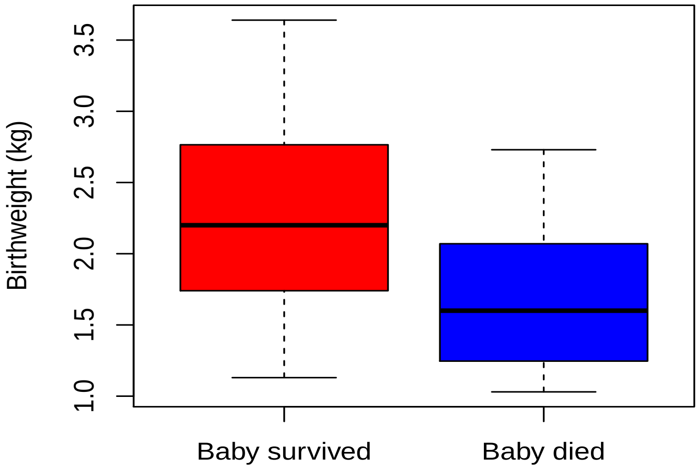

Figure 3.4 Box plots of birthweight by group.

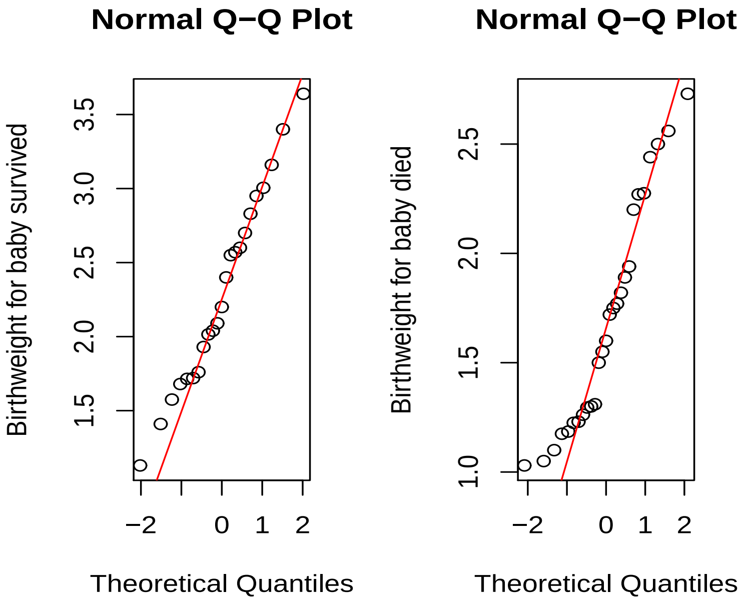

Figure 3.5 Q-Q plots of birthweight by group.

```
==============================================================
> par(mfrow=c(1, 2))
> qqnorm(birthwt.surv, ylab="Birthweight for baby survived")
> qqline(birthwt.surv, col=2)
> qqnorm(birthwt.died, ylab="Birthweight for baby died")
> qqline(birthwt.died, col=2)
**************************************************************
```

42.3• Shapiro–Wilk test

— To perform a two-sample *t* test, we need to test the normality assumption by performing Shapiro–Wilk test.

```
==============================================================
> shapiro.test(birthwt.surv)

        Shapiro-Wilk normality test

data:  birthwt.surv
W = 0.97699, p-value = 0.8491
--------------------------------------------------------------
> shapiro.test(birthwt.died)

        Shapiro-Wilk normality test

data:  birthwt.died
W = 0.91899, p-value = 0.03733
**************************************************************
```

42.4• *F* test

— To perform a two-sample *t* test, we need to test the equality of two variances by performing the *F* test.

```
==============================================================
> var.test(birthwt.surv, birthwt.died)

        F test to compare two variances
```

```
data:  birthwt.surv and birthwt.died
F = 1.649, num df = 22, denom df = 26, p-value = 0.2218
H_1: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.7347471 3.8071917
sample estimates:
ratio of variances
          1.648972
**************************************************************
```

42.5• Two-sample t test

```
==============================================================
> t.test(birthwt.surv, birthwt.died, var.equal=T)

        Two Sample t-test

data:  birthwt.surv and birthwt.died
t = 3.6797, df = 48, p-value = 0.0005902
H_1: true difference in means is not equal to 0
95 percent confidence interval:
 0.2792545 0.9520466
sample estimates:
mean of x mean of y
 2.307391  1.691741
--------------------------------------------------------------
> t.test(birthwt.surv, birthwt.died)

        Welch Two Sample t-test

data:  birthwt.surv and birthwt.died
t = 3.6068, df = 41.28, p-value = 0.0008289
H_1: true difference in means is not equal to 0
95 percent confidence interval:
 0.2710033 0.9602979
sample estimates:
mean of x mean of y
 2.307391  1.691741
```

```
**************************************************************
```

- — These results indicate that there is clearly a significant difference in the average birthweights of the two groups with those who survived having a larger value than those who died.
- — The 95% CIs indicate that the true difference in means is somewhere between a third to one kilogram.

42.6• Two-sample Wilcoxon test

```
==============================================================
> wilcox.test(birthwt.surv, birthwt.died)

        Wilcoxon rank sum test with continuity correction

data:  birthwt.surv and birthwt.died
W = 473, p-value = 0.001613
H_1: true location shift is not equal to 0
**************************************************************
```

42.7• Two-sample Kolmogorov–Smirnov test

```
==============================================================
> ks.test(birthwt.surv, birthwt.died)

        Two-sample Kolmogorov-Smirnov test

data:  birthwt.surv and birthwt.died
D = 0.39936, p-value = 0.03806
alternative hypothesis: two-sided
**************************************************************
```

Chapter 4

One- and K-sample Tests for Categorical Data

1• Aims in statistical aspects

- In this chapter, we consider one-, two-, and K-sample proportions test problems in independent groups and corresponding confidence interval estimation for parameters of interest.

- Next we consider McNemar's test for two paired proportions and independency tests in general $r \times c$ tables.

- We consider approximate normal and chi-squared tests with/without continuity correction.

- We also consider exact binomial test and Fisher's exact test.

2• Aims in software aspects

2.1• Introduction of four R functions

— `prop.test()` for testing the null hypothesis that the proportions (probabilities of success) in several groups are identical, or that one proportion is equal to a given value.

— `binom.test()` for an exact binomial test of a simple null hypothesis about the probability of success in a Bernoulli experiment.

— `chisq.test()` for performing Pearson's chi-squared test for contingency tables and goodness-of-fit test for one-way tables.

— `fisher.test()` for performing Fisher's exact test for testing the independency of rows and columns in a contingency table with fixed marginals.

2.2• Two R functions for testing paired proportions and trend

— `mcnemar.test()` for testing two paired proportions.

— `prop.trend.test()` for testing a trend in the proportions.

4.1 One-sample proportion test and exact binomial test

4.1.1 Hypothesis test

3• Statistical issue

- The one-sample proportion test (or z test or *normal* test) is used to test the null hypothesis that the successful proportion of a *Bernoulli* population is equal to a pre-specified proportion.

- That is, H_0: $p = p_0$ against one of the three alternatives: $p > p_0$, $p < p_0$ or $p \neq p_0$.

4• Assumptions

- Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim}$ Bernoulli(p), i.e., the Bernoulli distribution with the successful proportion (or true response rate) $p = \Pr(X_i = 1)$, where X_i is a binary random variable ($i = 1, \ldots, n$).

- Let $x_1, \ldots, x_n$ denote the realizations of $X_1, \ldots, X_n$.

5• Test statistic and z value

- For *large sample sizes*, the normal distribution is used to approximate the binomial distribution.

- The test statistic and z value are given by

$$Z = \frac{\bar{X} - p_0}{\sqrt{p_0(1-p_0)/n}} = \frac{\sum_{i=1}^{n} X_i - np_0}{\sqrt{np_0(1-p_0)}}$$

  and

$$z = \frac{\hat{p} - p_0}{\sqrt{p_0(1-p_0)/n}} = \frac{\sum_{i=1}^{n} x_i - np_0}{\sqrt{np_0(1-p_0)}} = \frac{n_1 - np_0}{\sqrt{np_0(1-p_0)}}, \tag{4.1}$$

  respectively, where

  — $\bar{X} = \sum_{i=1}^{n} X_i/n$, and

  — $\hat{p} = \bar{x} = \sum_{i=1}^{n} x_i/n = n_1/n$ is an unbiased point estimate of p.

6• Approximate p-values

- Under H_0, $Z \mathrel{\dot\sim} N(0,1)$. As a rule of thumb, the normal approximation is satisfactory when both n_1 (the number of "successes") and $n - n_1$ (the number of "failures") are larger than 5.

- The corresponding p-values are given by

$$\begin{aligned}
p\text{-value} &= \Pr\{Z > z\}, && \text{if } H_1\colon\ p > p_0, \quad (4.2)\\
p\text{-value} &= \Pr\{Z < z\}, && \text{if } H_1\colon\ p < p_0, \quad (4.3)\\
p\text{-value} &= 2\Pr\{Z > |z|\} && \\
&= \Pr\{Z^2 > z^2\} && \\
&= \Pr\{\chi^2(1) > z^2\}, && \text{if } H_1\colon\ p \neq p_0. \quad (4.4)
\end{aligned}$$

- When p-value $\geqslant \alpha$, we cannot reject the H_0.

- The corresponding R codes are as follows:

```
===============================================================
> n1 <- sum(x)
> n <- length(x)
> s <- sqrt(n*p0*(1-p0))
> z <- (n1 - n*p0)/s                             # c.f. (4.1)
> p.larger <- 1 - pnorm(z)                       # c.f. (4.2)
> p.smaller <- pnorm(z)                          # c.f. (4.3)
> p.value <- 2*( 1 - pnorm(abs(z)) )             # c.f. (4.4)
***************************************************************
```

7$^\bullet$ Continuity correction

- A continuity correction is an adjustment that is made when a discrete distribution is approximated by a continuous distribution.

- Recalculating the z value with continuity correction gives

$$z_{\text{wcc}} = \frac{|n_1 - np_0| - 0.5}{\sqrt{np_0(1-p_0)}}. \tag{4.5}$$

- The p-value is $2\,\Pr\{Z > |z_{\text{wcc}}|\}$.

- The corresponding R codes are as follows:

```
==============================================================
> z.wcc <- (abs(n1 - n*p0) - 0.5)/s              # c.f. (4.5)
> p.value.wcc <- 2*( 1 - pnorm(abs(z.wcc)) )
**************************************************************
```

8$^\bullet$ Exact binomial test in R

- When the sample size is not too large, we need to compute the exact p-values by means of the exact binomial test.

8.1$^\bullet$ Test statistic

— Note that $Y = \sum_{i=1}^n X_i$ is the test statistic and $n_1 = \sum_{i=1}^n x_i$ is the observed value of Y.

— Since $Y \sim \text{Binomial}(n, p)$, we have $Y|H_0 \sim \text{Binomial}(n, p_0)$.

— Define

$$\Pr(Y = y|H_0) = \binom{n}{y} p_0^y (1-p_0)^{n-y} \mathrel{\hat{=}} \theta_y, \quad y = 0, 1, \dots, n. \tag{4.6}$$

8.2$^\bullet$ Left- and right-sided p-values

— $\Pr(Y \leqslant n_1|H_0) = \sum_{y=0}^{n_1} \theta_y$ is called the *left-sided p-value.*

— $\Pr(Y \geqslant n_1|H_0) = \sum_{y=n_1}^{n} \theta_y$ is called the *right-sided p-value.*

8.3$^\bullet$ Exact one- and two-sided p-values

— The *exact one-sided p-value* is calculated by

$$p\text{-value} = \min\Big\{\Pr(Y \leqslant n_1|H_0),\ \Pr(Y \geqslant n_1|H_0)\Big\}. \tag{4.7}$$

— The *exact two-sided p-value* in SAS is computed as

$$p\text{-value} = 2 \times \min\Big\{\textstyle\sum_{y=0}^{n_1}\theta_y,\ \sum_{y=n_1}^{n}\theta_y\Big\}. \tag{4.8}$$

8.4• The corresponding R codes

```
==============================================================
> n1 <- sum(x)
> n <- length(x)
> L.p.value <- pbinom(n1, n, p0)
> R.p.value <- 1 - pbinom(n1-1, n, p0)
> eos.p.value <- min(L.p.value, R.p.value)       # c.f. (4.7)
> ets.p.value <- 2*eos.p.value                   # c.f. (4.8)
**************************************************************
```

9• The correct formula of the exact two-sided p-value

- In fact, the two-sided p-value calculated via (4.8) is just an approximate to the exact two-sided p-value and is possible to have a value beyond 1, leading to a useless p-value.
- The *correct* method for computing the exact two-sided p-value is

  $$p\text{-value} = \sum_{y=0}^{n}\theta_y I_{(\theta_y \leqslant \theta_{n_1})}, \tag{4.9}$$

  where $I_{(\cdot)}$ denotes the indicator function.
- The corresponding R code is

```
==============================================================
> n1 <- sum(x); n <- length(x)
> pv <- 0
> theta.n1 <- dbinom(n1, n, p0)
> for (y in 0:n) {
      theta.y <- dbinom(y, n, p0)
      if (theta.y <= theta.n1) { pv <- pv + theta.y }
  }
> p.value <- pv
***************************************************************
```

4.1.2 Confidence intervals

10• Equivalence between CI method and hypothesis test

- Alternatively, the CI method can determine whether or not to reject H_0: $p = p_0$.

- Let $[\hat{p}_{\rm L}, \hat{p}_{\rm U}]$ denote a $(1-\alpha)100\%$ CI of p.

- The rule of thumb is as follows:

— If $p_0 \in [\hat{p}_{\rm L}, \hat{p}_{\rm U}]$, we cannot reject the H_0 at α level of significance.

— If $p_0 \notin [\hat{p}_{\rm L}, \hat{p}_{\rm U}]$, we reject the H_0 at α level of significance.

11• The central limit theorem (CLT)

- According to the CLT: $[\bar{X} - E(\bar{X})]/[{\rm Var}(\bar{X})]^{1/2}$ converges in distribution to a random variable following $N(0,1)$, we have

$$\frac{\bar{X} - p}{\sqrt{p(1-p)/n}} \xrightarrow{\rm L} Z_0 \sim N(0,1). \qquad (4.10)$$

- Let z_α be the upper α quantile of $N(0,1)$ satisfying $\Pr(Z_0 \geqslant z_\alpha) = \alpha$.

12• Wald confidence limits

12.1• Derivation

— Based on limiting properties of MLE, we approximately have

$$\frac{\bar{X} - p}{\sqrt{\hat{p}(1-\hat{p})/n}} \mathrel{\dot\sim} N(0,1) \quad \text{as} \quad n \to \infty. \qquad (4.11)$$

— Therefore, the asymptotic $100(1-\alpha)\%$ CI of p can be derived from

$$\begin{aligned} 1-\alpha &= \Pr\left\{ -z_{\alpha/2} \leqslant \frac{\bar{X}-p}{\sqrt{\hat{p}(1-\hat{p})/n}} \leqslant z_{\alpha/2} \right\} \\ &= \Pr\left\{ \hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leqslant p \leqslant \hat{p} + z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right\}. \end{aligned}$$

— The Wald CI for p is

$$[\hat{p}_{\text{W,L}},\ \hat{p}_{\text{W,U}}] = \left[\hat{p} - z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n},\ \ \hat{p} + z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}\,\right]. \quad (4.12)$$

12.2• A drawback. However, one drawback of the Wald CI (4.12) is that

— the lower bound may be beyond zero when the true value of p is close to zero

— while the upper bound may be beyond one when the true value of p is near to one.

13• Wilson or score confidence limits

13.1• Derivation

— When the lower bound of the Wald CI (4.12) is less than zero or the upper bound is larger than one, we can construct the second asymptotic $(1-\alpha)100\%$ CI of p based on

$$\begin{aligned} 1-\alpha &\overset{(4.10)}{=} \Pr\left\{ \left| \frac{\bar{X}-p}{\sqrt{p(1-p)/n}} \right| \leqslant z_{\alpha/2} \right\} \\ &= \Pr\left\{ \left| \frac{\hat{p}-p}{\sqrt{p(1-p)/n}} \right| \leqslant z_{\alpha/2} \right\} \\ &= \Pr\{(\hat{p}-p)^2 \leqslant z_{\alpha/2}^2 p(1-p)/n\} \\ &= \Pr\{(1+z_*)p^2 - (2\hat{p}+z_*)p + \hat{p}^2 \leqslant 0\}, \qquad (4.13) \end{aligned}$$

where $z_* = z_{\alpha/2}^2/n$.

— Solving the quadratic inequality inside the probability in (4.13), we obtain the Wilson (score) CI of p as follows:

$$\begin{aligned}[\hat{p}_{\text{WS,L}},\ \hat{p}_{\text{WS,U}}] &= \frac{2\hat{p} + z_* \pm \sqrt{(2\hat{p} + z_*)^2 - 4(1 + z_*)\hat{p}^2}}{2(1 + z_*)} \\ &= \frac{2\hat{p} + z_* \pm \sqrt{4z_*\hat{p}(1 - \hat{p}) + z_*^2}}{2(1 + z_*)}, \qquad (4.14)\end{aligned}$$

which is within $[0, 1]$.

13.2• The corresponding R function

```
function (phat, zstar)
{ # Function name: Wilson.CI(phat, zstar)
  # -------------------- Aim ----------------------------------
  # Compute Wilson CI of p using (4.14)
  # -------------------- Input --------------------------------
  #   phat: the MLE of p = mean(x)
  #  zstar: qnorm(1-alpha/2)^2/n
  # -------------------- Output -------------------------------
  # result: Wilson CI of p
  #############################################################
  zs <- zstar
  bb <- sqrt(4*zs*phat*(1-phat) + zs^2)
  pL <- (2*phat + zs - bb)/(2*(1+zs))
  pU <- (2*phat + zs + bb)/(2*(1+zs))
  result <- c(pL, pU)
  return(result)
}**************************************************************
```

13.3• A merit

— The Wilson CI has been shown to have better performance than the Wald CI and the exact (Clopper–Pearson) CI.

— See Agresti and Coull (1998, The American Statistician, **52**, 119–126), Brown, Cai and DasGupta (2001, Statistical Science, **16**, 101–133), and Newcombe (1998, Statistics in Medicine, **17**, 857–872) for more detail.

13.4• Continuity correction

— With continuity correction, the Wilson CI of p can be obtained based on

$$1-\alpha = \Pr\left\{\left|\frac{|\hat{p}-p|-1/(2n)}{\sqrt{p(1-p)/n}}\right| \leqslant z_{\alpha/2}\right\}.$$

— From (4.14), it is clear that the Wilson (score) CI of p depends on the value of $\hat{p}$, so we denote it by $[\hat{p}_{\text{WS,L}}(\hat{p}),\ \hat{p}_{\text{WS,U}}(\hat{p})]$.

— When $\hat{p}$ is replaced by $\hat{p}-1/(2n)$, the corresponding Wilson CI is $[\hat{p}_{\text{WS,L}}(\hat{p}-1/(2n)),\ \hat{p}_{\text{WS,U}}(\hat{p}-1/(2n))]$.

— When $\hat{p}$ is replaced by $\hat{p}+1/(2n)$, the corresponding Wilson CI is $[\hat{p}_{\text{WS,L}}(\hat{p}+1/(2n)),\ \hat{p}_{\text{WS,U}}(\hat{p}+1/(2n))]$.

— Hence, with continuity correction, the Wilson CI of p is $[\hat{p}^{\text{wcc}}_{\text{WS,L}},\ \hat{p}^{\text{wcc}}_{\text{WS,U}}]$, where

$$\begin{aligned}\hat{p}^{\text{wcc}}_{\text{WS,L}} &= \min\left\{\hat{p}_{\text{WS,L}}(\hat{p}-1/(2n)),\ \hat{p}_{\text{WS,L}}(\hat{p}+1/(2n))\right\} \quad \text{and}\\ \hat{p}^{\text{wcc}}_{\text{WS,U}} &= \max\left\{\hat{p}_{\text{WS,U}}(\hat{p}-1/(2n)),\ \hat{p}_{\text{WS,U}}(\hat{p}+1/(2n))\right\}. \qquad (4.15)\end{aligned}$$

14• Exact or clopper–pearson confidence limits

14.1• Background and definition

— When the sample size is small to moderate, we can compute the exact or Clopper–Pearson confidence limits for the binomial proportion by inverting the equal-tailed test based on the binomial distribution.

— This method is attributed to Clopper and Pearson (1934, Biometrika, **26**, 404–413).

— The exact confidence limits $p_{\text{E,L}}$ and $p_{\text{E,U}}$ satisfy the following equations:

$$\begin{aligned} p_{\text{E,L}} &= 0, && \text{when} \quad n_1 = 0,\\ \sum_{x=n_1}^{n}\binom{n}{x}p^x_{\text{E,L}}(1-p_{\text{E,L}})^{n-x} &= \frac{\alpha}{2}, && n_1 = 1,\ldots,n-1, \qquad (4.16)\\ \sum_{x=0}^{n_1}\binom{n}{x}p^x_{\text{E,U}}(1-p_{\text{E,U}})^{n-x} &= \frac{\alpha}{2}, && n_1 = 1,\ldots,n-1, \qquad (4.17)\\ p_{\text{E,U}} &= 1, && \text{when} \quad n_1 = n.\end{aligned}$$

14.2• Relationship between binomial and beta distributions

— The binomial and beta distributions have the following relationship:

$$\sum_{x=0}^{k} \binom{n}{x} p^x (1-p)^{n-x} = \int_0^{1-p} \frac{x^{n-k-1}(1-x)^k}{B(n-k,k+1)}\, \mathrm{d}x, \ 0 \leqslant k \leqslant n. \tag{4.18}$$

— Let $\sum_{x=0}^{k} \binom{n}{x} p^x (1-p)^{n-x} = q$, then (4.18) is equivalent to

$$1-p = \beta(1-q; n-k, k+1) \quad \text{or} \quad p = \beta(q; k+1, n-k),$$

where $\beta(\alpha; a, b)$ is the upper α quantile of the beta distribution $\text{Beta}(a,b)$.

— It is easy to show that $\beta(1-\alpha; a, b) = 1 - \beta(\alpha; b, a)$.

— Thus, solving (4.16) and (4.17), we obtain

$$\begin{aligned} p_{\text{E,L}} &= \beta(1-\alpha/2; n_1, n-n_1+1) \\ &\overset{(4.21)}{=} \left[1 + \frac{n-n_1+1}{n_1 F(1-\alpha/2; 2n_1, 2(n-n_1+1))}\right]^{-1}, \end{aligned} \tag{4.19}$$

$$\begin{aligned} p_{\text{E,U}} &= \beta(\alpha/2; n_1+1, n-n_1) \\ &\overset{(4.21)}{=} \left[1 + \frac{n-n_1}{(n_1+1) F(\alpha/2; 2(n_1+1), 2(n-n_1))}\right]^{-1}. \end{aligned} \tag{4.20}$$

where $F(\alpha; k_1, k_2)$ is the upper α quantile of the F distribution $F(k_1, k_2)$.

14.3• Relationship between beta and F distributions

— Let $Z \sim \text{Beta}(k_1, k_2)$, then

$$X = \frac{k_2}{k_1} \cdot \frac{Z}{1-Z} \sim F(2k_1, 2k_2).$$

— Since $Z = [1 + k_2/(k_1 X)]^{-1}$ is a monotone increasing function of X, we have

$$\beta(\alpha; k_1, k_2) = \left(1 + \frac{k_2}{k_1 F(\alpha; 2k_1, 2k_2)}\right)^{-1}. \tag{4.21}$$

14.4• Conservative CI

— Because this is a discrete problem, the confidence coefficient (or coverage probability) of the exact (Clopper–Pearson) CI is not exactly $1 - \alpha$ but is at least $1 - \alpha$.

— Thus, this exact CI is conservative.

4.1.3 Example 4.1

15• Data and questions

- Let
$$1,\ 1,\ 0,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1 \tag{4.22}$$
be an observed sample of size $n = 10$ from a Bernoulli distribution with the proportion parameter p.

  — We want to test H_0: $p = p_0 = 0.2$ against H_1: $p \neq 0.2$ at $\alpha = 0.05$ level of significance.

  — Find the 95% Wald, Wilson and exact CIs of p by using (4.12), (4.14), (4.19) and (4.20), respectively.

16• R function

```
function (ind, x, p0, alpha)
{ # Function name:pvalue.CIs.for.single.prop(ind, x, p0,alpha)
  # -------------------- Aim ----------------------------------
  # Testing hypothesis and CIs for a single proportion
  # -------------------- Input --------------------------------
  # ind = 1: compute z- & p-value w/o continuity correction
  # ind = 2: compute exact one/two-sided p-values in SAS
  # ind = 3: compute correct exact two-sided p-value via (4.9)
  # ind = 4: compute Wald, Wilson, Wilson.wcc & exact CI of p
  #       x: a binary vector of length n
  #      p0: H_0: p = p0
  #   alpha: 0.05
  # -------------------- Output -------------------------------
  # z-values, p-values, four CIs
  ###############################################################
  n1 <- sum(x); n <- length(x)
```

```
if (ind == 1) {
  # ---- z-value & p-value w/o continuity correction -------
  s <- sqrt(n*p0*(1-p0))
  z <- (n1 - n*p0)/s                           # c.f. (4.1)
  pv <- 2*( 1 - pnorm(abs(z)) )                # c.f. (4.4)
  z.wcc <- (abs(n1 - n*p0) - 0.5)/s            # c.f. (4.5)
  pv.wcc <- 2*( 1 - pnorm(abs(z.wcc)) )
  resultM <- matrix(c(z, pv, z.wcc, pv.wcc), nrow=2,byrow=F)
  rownames(resultM) <- c("z-value", "p-value")
  colnames(resultM) <- c("   Without conti. correction",
                         "   With conti. correction")
  return(resultM)
} else if (ind == 2) {
# -------- exact one/two-sided p-values in SAS -------------
  L.pv <- pbinom(n1, n, p0)
  R.pv <- 1 - pbinom(n1-1, n, p0)
  eos.pv <- min(L.pv, R.pv)                    # c.f. (4.7)
  ets.pv <- 2*eos.pv                           # c.f. (4.8)
  resultM <- matrix(c(eos.pv, ets.pv), nrow=1, byrow=T)
  rownames(resultM) <- c("Exact p-value in SAS")
  colnames(resultM) <- c("   One-sided", "   Two-sided")
  return(resultM)
} else if (ind == 3) {
# --------- correct exact two-sided p-value using (4.9) ----
  pv <- 0
  theta.n1 <- dbinom(n1, n, p0)
  for (y in 0:n) {
    theta.y <- dbinom(y, n, p0)
    if (theta.y <= theta.n1) { pv <- pv + theta.y }
  }
  resultM <- matrix(c(pv), nrow=1, byrow=T)
  rownames(resultM) <- c("Correct exact p-value")
  colnames(resultM) <- c("   Two-sided")
  return(resultM)
} else
# ----- compute Wald CI using (4.12) ----------------------
phat <- n1/n
SEM <- sqrt(phat*(1-phat)/n)
```

```
  qN <- qnorm(1-alpha/2)
  pWL <- phat - qN*SEM; pWU <- phat + qN*SEM
  # ----- compute Wilson CI using (4.14) ---------------------
  result <- Wilson.CI(phat, zstar= qN^2/n)
  pWSL <- result[1]
  pWSU <- result[2]
  # ----- compute Wilson CI with cc using (4.15) -------------
  result1 <- Wilson.CI(phat-1/(2*n), zstar= qN^2/n)
  result2 <- Wilson.CI(phat+1/(2*n), zstar= qN^2/n)
  pWSL.wcc <- min(result1[1], result2[1])
  pWSU.wcc <- max(result1[2], result2[2])
  # ----- compute exact CI using (4.19) and (4.20) -----------
  qF1 <- qf(alpha/2, 2*n1, 2*(n-n1+1))
  qF2 <- qf(1-alpha/2, 2*(n1+1), 2*(n-n1))
  pEL <- 1/(1 + (n-n1+1)/(n1*qF1))
  pEU <- 1/(1 + (n-n1)/((n1+1)*qF2))
  resultM <- matrix(c(pWL, pWU, pWU-pWL, pWSL, pWSU, pWSU-pWSL,
                      pWSL.wcc, pWSU.wcc, pWSU.wcc-pWSL.wcc,
                      pEL, pEU, pEU-pEL), nrow=4, byrow=T)
  rownames(resultM) <- c("95% Wald CI", "95% Wilson CI",
                         "95% Wilson CI wcc", "95% Exact CI")
  colnames(resultM) <- c("   Lower bound", "   Upper bound",
                                           "      Width")
  return(resultM)
}**************************************************************
```

16.1• R output

```
==============================================================
> x <- c(1, 1, 0, 1, 1, 1, 1, 1, 1, 1)
> pvalue.CIs.for.single.prop(ind= 1, x, p0= 0.2, alpha= 0.05)
          Without conti. correction    With conti. correction
z-value                5.533986e+00              5.138701e+00
p-value                3.130341e-08              2.766439e-07
--------------------------------------------------------------
> pvalue.CIs.for.single.prop(ind= 2, x, p0= 0.2, alpha= 0.05)
                        One-sided    Two-sided
Exact p-value in SAS   4.1984e-06   8.3968e-06
```

```
--------------------------------------------------------------
> pvalue.CIs.for.single.prop(ind= 3, x, p0= 0.2, alpha= 0.05)
                         Two-sided
Correct exact p-value   4.1984e-06
--------------------------------------------------------------
> pvalue.CIs.for.single.prop(ind= 4, x, p0= 0.2, alpha= 0.05)
                     Lower bound   Upper bound       Width
95% Wald CI            0.7140615     1.0859385   0.3718770
95% Wilson CI          0.5958500     0.9821238   0.3862738
95% Wilson CI wcc      0.5411540     0.9947577   0.4536037
95% Exact CI           0.5549839     0.9974714   0.4424875
**************************************************************
```

16.2• Comments on above R output

— We note that the correct exact two-sided p-value (i.e., 4.1984×10^{-6}) calculated via (4.9) is different from the exact two-sided p-value in SAS (i.e., 8.3968×10^{-6}) calculated via (4.8). Since both p-values $\ll 0.05$, we reject H_0.

— The 95% Wald upper bound $\hat{p}_{\mathrm{W,U}} = 1.0859385 > 1$, leading to a useless Wald CI of p.

— Since the 95% Wilson CI, 95% Wilson CI with continuity correction and 95% exact CI exclude $p_0 = 0.2$, we reject H_0, which is consistent with the conclusion from the p-value method.

— The width of the Wilson CI (i.e., 0.3862738) is shorter than the width of the Wilson CI wcc (i.e., 0.4536037) and the width of the exact CI (i.e., 0.4424875).

17• Using the built-in R function prop.test()

```
==============================================================
> x <- c(1, 1, 0, 1, 1, 1, 1, 1, 1, 1)
> n1 <- sum(x); n <- length(x)
--------------------------------------------------------------
> prop.test(n1, n, p=0.2, correct=F, alt="t", conf.level=0.95)
```

```
     1-sample proportions test without continuity correction

data:  n1 out of n, null probability 0.2
X-squared = 30.625, df = 1, p-value = 3.13e-08
alternative hypothesis: true p is not equal to 0.2
95 percent Wilson CI:
 0.5958500 0.9821238
sample estimates:
  p
0.9
--------------------------------------------------------------
> prop.test(n1, n, p=0.2, correct=T, alt="t", conf.level=0.95)

       1-sample proportions test with continuity correction

data:  n1 out of n, null probability 0.2
X-squared = 26.406, df = 1, p-value = 2.766e-07
alternative hypothesis: true p is not equal to 0.2
95 percent Wilson CI:
 0.5411540 0.9947577
sample estimates:
  p
0.9
**************************************************************
```

17.1• Comments on above R output

— Without continuity correction, z-value = 5.533986. Its square is 30.625.

— With continuity correction, z-value = 5.138701. Its square is 26.406.

18• Using the built-in R function binom.test()

```
==============================================================
> x <- c(1, 1, 0, 1, 1, 1, 1, 1, 1, 1)
> n1 <- sum(x); n <- length(x)
--------------------------------------------------------------
> binom.test(n1, n, p=0.2, alt="t", conf.level=0.95)
```

```
        Exact binomial test

data:  n1 and n
number of successes = 9, number of trials = 10,
p-value = 4.198e-06               # see (4.9)
H_1: true probability of success is not equal to 0.2
95 percent confidence interval:   # exact CI
 0.5549839 0.9974714              # see (4.19) & (4.20)
sample estimates:
probability of success
                   0.9
**************************************************************
```

4.1.4 Example 4.2

19$^\bullet$ Data and questions

- Assume that the observed sample in (4.22) and $p_0 = 0.2$ are respectively replaced by

$$1,\ 1,\ 0,\ 1,\ 0,\ 1,\ 1,\ 1,\ 0,\ 1,\ 0,\ 1,\ 1,\ 0,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1, \qquad (4.23)$$

  and by $p_0 = 0.73$.

- Two questions are the same as those in Example 4.1.

20$^\bullet$ R output with $p_0 = 0.73$

```
==============================================================
> x <- c(1,1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,1,1,1,1,1)
> pvalue.CIs.for.single.prop(ind= 1, x, p0= 0.73, alpha= 0.05)
           Without conti. correction   With conti. correction
z-value                    0.2014660              -0.05036649
p-value                    0.8403342               0.95983034
--------------------------------------------------------------
> pvalue.CIs.for.single.prop(ind= 2, x, p0= 0.73, alpha= 0.05)
                        One-sided    Two-sided
Exact p-value in SAS    0.5357104     1.071421
--------------------------------------------------------------
```

```
> pvalue.CIs.for.single.prop(ind= 3, x, p0= 0.73, alpha= 0.05)
                          Two-sided
Correct exact p-value             1
--------------------------------------------------------------
> pvalue.CIs.for.single.prop(ind= 4, x, p0= 0.73, alpha= 0.05)
                     Lower bound    Upper bound        Width
95% Wald CI            0.5602273      0.9397727    0.3795454
95% Wilson CI          0.5312991      0.8881383    0.3568392
95% Wilson CI wcc      0.5058845      0.9040674    0.3981829
95% Exact CI           0.5089541      0.9134285    0.4044744
**************************************************************
```

20.1• Comments on above R output

— We noted that the exact two-sided p-value in SAS (i.e., 1.071421) calculated via (4.8) is beyond 1,

— The four CIs are within [0,1] and the width of the Wilson CI is the shortest.

— Since the correct exact two-sided p-values $= 1 > 0.05$ and the four 95% CIs include $p_0 = 0.73$, we cannot reject H_0: $p = p_0 = 0.73$.

— For the same observations in (5.22), let $p_0 = 0.70$ instead of $p_0 = 0.73$, we obtain the following results.

21• R output with $p_0 = 0.70$

```
==============================================================
> x <- c(1,1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,1,1,1,1,1)
> pvalue.CIs.for.single.prop(ind= 1, x, p0= 0.70, alpha= 0.05)
           Without conti. correction    With conti. correction
z-value                    0.4879500                 0.2439750
p-value                    0.6255852                 0.8072502
--------------------------------------------------------------
> pvalue.CIs.for.single.prop(ind= 2, x, p0= 0.70, alpha= 0.05)
                        One-sided    Two-sided
Exact p-value in SAS    0.4163708    0.8327417
--------------------------------------------------------------
```

```
> pvalue.CIs.for.single.prop(ind= 3, x, p0= 0.70, alpha= 0.05)
                              Two-sided
Correct exact p-value     0.808361
--------------------------------------------------------------
> pvalue.CIs.for.single.prop(ind= 4, x, p0= 0.70, alpha= 0.05)
                     Lower bound    Upper bound        Width
95% Wald CI            0.5602273      0.9397727    0.3795454
95% Wilson CI          0.5312991      0.8881383    0.3568392
95% Wilson CI wcc      0.5058845      0.9040674    0.3981829
95% Exact CI           0.5089541      0.9134285    0.4044744
**************************************************************
```

22• Using the built-in R function prop.test()

```
==============================================================
> x <- c(1,1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,1,1,1,1,1)
> n1 <- sum(x); n <- length(x)
--------------------------------------------------------------
> prop.test(n1, n, p=0.7, correct=F, alt="t", conf.level=0.95)

     1-sample proportions test without continuity correction

data:  n1 out of n, null probability 0.7
X-squared = 0.2381, df = 1, p-value = 0.6256
alternative hypothesis: true p is not equal to 0.7
95 percent Wilson CI:
 0.5312991 0.8881383
sample estimates:
   p
0.75
--------------------------------------------------------------
> prop.test(n1, n, p=0.7, correct=T, alt="t", conf.level=0.95)

        1-sample proportions test with continuity correction

data:  n1 out of n, null probability 0.7
X-squared = 0.059524, df = 1, p-value = 0.8073
alternative hypothesis: true p is not equal to 0.7
```

```
95 percent confidence interval:
 0.5058845 0.9040674
sample estimates:
   p
0.75
*************************************************************
```

23• Using the built-in R function binom.test()

```
=============================================================
> x <- c(1,1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,1,1,1,1,1)
> n1 <- sum(x); n <- length(x)
-------------------------------------------------------------
> binom.test(n1, n, p=0.7, alt="t", conf.level=0.95)

        Exact binomial test

data:  n1 and n
number of successes = 15, number of trials = 20,
p-value = 0.8084
H_1: true probability of success is not equal to 0.7
95 percent exact confidence interval:
 0.5089541 0.9134285
sample estimates:
probability of success
                  0.75
*************************************************************
```

4.2 Two-sample proportions test and Fisher's exact test

4.2.1 Hypothesis test and confidence interval

24• Statistical issue

- The two-sample proportions test (or two-sample z test) is used to test the null hypothesis that the proportions of two independent Bernoulli populations are identical.

- That is, H_0: $p_1 = p_2$ against one of the three alternatives: $p_1 > p_2$, $p_1 < p_2$ or $p_1 \neq p_2$.

25• Assumptions

- Let $X_{i1}, \ldots, X_{in_i} \overset{\text{iid}}{\sim} \text{Bernoulli}(p_i)$ be two *independent* samples, where the population proportion $p_i = \Pr(X_{ij} = 1)$, $i = 1, 2$.
- Let $x_{i1}, \ldots, x_{in_i}$ denote the realizations of $X_{i1}, \ldots, X_{in_i}$ for $i = 1, 2$.

26• Test statistic and z value

- For *large sample sizes*, we use the normal distribution to approximate the binomial distribution.
- The test statistic and z value are given by

$$Z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}} \quad \text{and} \quad z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})/m}}, \tag{4.24}$$

respectively, where

$$\hat{p}_i = \bar{x}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij} \mathrel{\hat{=}} \frac{r_i}{n_i}, \quad i = 1, 2, \quad \hat{p} = \frac{r_1 + r_2}{n_1 + n_2}, \tag{4.25}$$

$$\frac{1}{m} = \frac{1}{n_1} + \frac{1}{n_2} \quad \text{or} \quad m = \frac{n_1 n_2}{n_1 + n_2}.$$

27• Approximate p-values

- Under H_0, $Z \mathrel{\dot{\sim}} N(0, 1)$.
- The corresponding p-values are given by

$$p\text{-value} = \Pr\{Z > z\}, \quad \text{if } H_1\text{: } p_1 > p_2, \tag{4.26}$$

$$p\text{-value} = \Pr\{Z < z\}, \quad \text{if } H_1\text{: } p_1 < p_2, \tag{4.27}$$

$$\begin{aligned} p\text{-value} &= 2\Pr\{Z > |z|\} \\ &= \Pr\{Z^2 > z^2\} \\ &= \Pr\{\chi^2(1) > z^2\}, \quad \text{if } H_1\text{: } p_1 \neq p_2. \end{aligned} \tag{4.28}$$

- When p-value $\geqslant \alpha$ (in general, $\alpha = 0.05$), we cannot reject the H_0.

- The corresponding R codes are as follows:

```
==============================================================
> r1 <- sum(x1); n1 <- length(x1)
> r2 <- sum(x2); n2 <- length(x2)
> p1hat <- r1/n1
> p2hat <- r2/n2
> phat <- (r1 + r2)/(n1 + n2)                   # c.f. (4.25)
> m <- n1*n2/(n1 + n2)
> s <- sqrt(phat*(1-phat)/m)
> z <- (p1hat - p2hat)/s                        # c.f. (4.24)
> p.larger <- 1 - pnorm(z)                      # c.f. (4.26)
> p.smaller <- pnorm(z)                         # c.f. (4.27)
> p.value <- 2*( 1 - pnorm(abs(z)) )            # c.f. (4.28)
**************************************************************
```

28• Continuity correction

- Recalculating the z value with continuity correction gives

$$z_{\text{wcc}} = \frac{|\hat{p}_1 - \hat{p}_2| - \frac{1}{2m}}{\sqrt{\hat{p}(1-\hat{p})/m}}. \tag{4.29}$$

- The p-value is $2\,\Pr\{Z > |z_{\text{wcc}}|\}$.

- The corresponding R codes are as follows:

```
==============================================================
> z.wcc <- (abs(p1hat-p2hat) - 0.5/m)/s   # c.f. (4.29)
> p.value.wcc <- 2*( 1 - pnorm(abs(z.wcc)) )
**************************************************************
```

29• Asymptotic confidence interval for $p_1 - p_2$

- According to the Central Limit Theorem, $\bar{X}_i \,\dot{\sim}\, N(p_i, p_i(1-p_i)/n_i)$, $i = 1, 2$, we have

$$\frac{\bar{X}_1 - \bar{X}_2 - (p_1 - p_2)}{\sqrt{p_1(1-p_1)/n_1 + p_2(1-p_2)/n_2}} \,\dot{\sim}\, N(0,1). \tag{4.30}$$

- Based on limiting properties of MLE, we obtain

$$\frac{\bar{X}_1 - \bar{X}_2 - (p_1 - p_2)}{\sqrt{\hat{p}_1(1-\hat{p}_1)/n_1 + \hat{p}_2(1-\hat{p}_2)/n_2}} \dot{\sim} N(0,1).$$

- Therefore, a $100(1-\alpha)\%$ asymptotic CI for $p_1 - p_2$ is given by

$$\hat{p}_1 - \hat{p}_2 \pm z_{\alpha/2}\sqrt{\hat{p}_1(1-\hat{p}_1)/n_1 + \hat{p}_2(1-\hat{p}_2)/n_2}. \tag{4.31}$$

- If this CI includes zero, we cannot reject the H_0: $p_1 = p_2$ at α level of significance.

- The corresponding R codes are as follows:

```
==============================================================
> diff <- p1hat - p2hat
> SEM <- sqrt(p1hat*(1-p1hat)/n1 + p2hat*(1-p2hat)/n2)
> qN <- qnorm(1-alpha/2)
> pL <- diff - qN*SEM; pU <- diff + qN*SEM
**************************************************************
```

4.2.2 Example 4.3

30• Data and questions

- Consider data from a randomized clinical trial comparing *infra-red stimulation* (IRS) with placebo on the pain caused by cervical osteoarthrosis.

- The placebo treatment was mock transcutaneous electrical stimulation and the patients were blind to the treatment given.

- Twenty-six patients were entered into the trial, but one dropped out before the end.

- Nine of the 12 patients in the IRS group reported an improvement in pain compared with four of the 13 receiving the placebo treatment.

- In this example, we have $n_1 = 12$, $r_1 = 9$, $n_2 = 13$, $r_2 = 4$.

  — We want to test H_0: $p_1 = p_2$ against H_1: $p_1 \neq p_2$ at $\alpha = 0.05$.

  — Find the 95% asymptotic CI of $p_1 - p_2$.

31• R function

```
function (ind, n1, r1, n2, r2, alpha)
{ # Name: pvalue.CI.for.two.props(ind, n1, r1, n2, r2, alpha)
  # -------------------- Aim ---------------------------------
  # Testing hypothesis and CI for p_1-p_2 in two proportions
  # -------------------- Input -------------------------------
  # ind = 1: Summary statistics
  # ind = 2: compute z- & p-value w/o continuity correction
  # ind = 3: compute asymptotic CI of p_1 - p_2
  #  n1, r1: sample size and number of successes in group 1
  #  n2, r2: sample size and number of successes in group 2
  #   alpha: 0.05
  # -------------------- Output ------------------------------
  # Summary statistics, z-value, z-square, p-value, and CI
  ############################################################
  p1hat <- r1/n1; p2hat <- r2/n2
  diff = p1hat - p2hat
  phat <- (r1 + r2)/(n1 + n2)
  m <- n1*n2/(n1 + n2)
  s <- sqrt(phat*(1-phat)/m)
  if (ind == 1) {
    resultM <- matrix(c(p1hat, p2hat, diff, phat),
                       nrow=4, byrow=T)
    rownames(resultM) <- c("p1.hat", "p2.hat",
                           "p1.hat-p2.hat", "p.hat")
    colnames(resultM) <- c("       MLE")
    return(resultM)
  } else if (ind == 2) {
    # ----------------- without continuity correction --------
    z <- diff/s
    pv <- 2*( 1 - pnorm(abs(z)) )
    # ----------------- with continuity correction -----------
    z.wcc <- (abs(diff) - 0.5/m)/s
    pv.wcc <- 2*( 1 - pnorm(abs(z.wcc)) )
    resultM <- matrix(c(z, z^2, pv, z.wcc, z.wcc^2, pv.wcc),
                                    nrow=3, byrow=F)
    rownames(resultM) <- c("z-value", "z-square", "p-value")
    colnames(resultM) <- c("   Without conti. correction",
                           "   With conti. correction")
    return(resultM)
  } else
    #---- asymptotic CI of p_1 - p_2 ------------------------
    SEM <- sqrt(p1hat*(1-p1hat)/n1 + p2hat*(1-p2hat)/n2)
    qN <- qnorm(1-alpha/2)
    pL <- diff - qN*SEM
    pU <- diff + qN*SEM
    resultM <- matrix(c(pL, pU, pU-pL), nrow=1, byrow=T)
    rownames(resultM) <- c("95% asymptotic CI of p_1-p_2")
    colnames(resultM) <- c("   Lower bound", "   Upper bound",
                           "      Width")
    return(resultM)
}**************************************************************
```

31.1• R output

```
==============================================================
> n1 <- 12; r1 <- 9; n2 <- 13; r2 <- 4; alpha <- 0.05
--------------------------------------------------------------
> pvalue.CI.for.two.props(ind= 1, n1, r1, n2, r2, alpha)
                    MLE
p1.hat         0.7500000
p2.hat         0.3076923
p1.hat-p2.hat  0.4423077
p.hat          0.5200000
--------------------------------------------------------------
> pvalue.CI.for.two.props(ind= 2, n1, r1, n2, r2, alpha)
           Without conti. correction   With conti. correction
z-value                   2.21153846               1.81089744
z-square                  4.89090237               3.27934952
p-value                   0.02699857               0.07015673
--------------------------------------------------------------
> pvalue.CI.for.two.props(ind= 3, n1, r1, n2, r2, alpha)
                           Lower bound   Upper bound      Width
95% asym. CI of p_1-p_2     0.09163853     0.7929769  0.7013383
**************************************************************
```

31.2• Conclusion from above R output

— The p-value without continuity correction is $0.027 < 0.05$.

— The 95% asymptotic CI of $p_1 - p_2$ is $[0.0916, 0.7929]$, which excludes zero.

— Therefore, we reject H_0, implying that there is evidence of a difference between the treatments.

32• Using the built-in R function prop.test()

- The built-in R function `prop.test()` can also be used to compare two proportions.
- For the purpose, the arguments should be given as two vectors, where the first contains the numbers of positive outcomes and the second the total numbers for each group.

```
==============================================================
> succ <- c(9, 4) # c(r1, r2): a vector of counts of successes
> total <- c(12, 13) # c(n1, n2): a vector of counts of trials
--------------------------------------------------------------
> prop.test(succ, total, correct=F, alt="t", conf.level=0.95)

        2-sample test for equality of proportions
                    without continuity correction

data:  succ out of total
X-squared = 4.8909, df = 1, p-value = 0.027
alternative hypothesis: two.sided
95 percent CI of p_1 - p_2:
 0.09163853 0.79297686
sample estimates:
   prop 1    prop 2
0.7500000 0.3076923
--------------------------------------------------------------
> prop.test(succ, total, correct=T, alt="t", conf.level=0.95)

        2-sample test for equality of proportions
```

```
                       with continuity correction

data:  succ out of total
X-squared = 3.2793, df = 1, p-value = 0.07016
alternative hypothesis: two.sided
95 percent  CI of p_1 - p_2:
 0.01151032 0.87310506
sample estimates:
   prop 1    prop 2
0.7500000 0.3076923
**************************************************************
```

33• Alternative data summary in a two by two table

- We can re-arrange the pain data in a 2×2 table in Table 4.1.

Table 4.1 *Pain data in the form of a* 2×2 *table*

	Improvement in Pain		
Treatment	Yes	No	Total
IRS	p_1 $(r_1 = 9)$	$1 - p_1$ $(n_1 - r_1 = 3)$	$n_1 = 12$
Placebo	p_2 $(r_2 = 4)$	$1 - p_2$ $(n_2 - r_2 = 9)$	$n_2 = 13$

34• Using the built-in R function chisq.test()

- One aim of the Pearson chi-squared test is to test the null hypothesis H_0: the row variable and the column variable in a contingency table are unrelated/independent. For more details, see §4.4.2.

- In §4.4.3, we will show that the two-sample z test (i.e., two-sample proportions test) presented in this section is equivalent to the Pearson chi-squared test for a 2×2 table.

- In other words, the test for the difference of two proportions being zero in two independent Bernoulli populations is also a test for independency between the row variable and the column variable in a 2×2 table.

- For a 2×2 table, the built-in R function `chisq.test()` is exactly equivalent to `prop.test()`.

```
===============================================================
> pain.data <- matrix(c(9, 3, 4, 9), nrow=2, byrow=T)
> pain.data
     [,1] [,2]
[1,]    9    3
[2,]    4    9
---------------------------------------------------------------
> chisq.test(pain.data, correct=F)

        Pearson's Chi-squared test

data:  pain.data
X-squared = 4.8909, df = 1, p-value = 0.027
---------------------------------------------------------------
> chisq.test(pain.data, correct=T)

 Pearson's Chi-squared test with Yates' continuity correction

data:  pain.data
X-squared = 3.2793, df = 1, p-value = 0.07016
***************************************************************
```

35• Using the built-in R function fisher.test()

- For the pain data in Table 4.1, the p-values of Pearson's chi-squared test with/withou Yates' continuity correction are different, resulting in a self-contradictory conclusion.

- In general, when some cell counts in a contingency table are less than 5, Pearson's chi-squared test is not reliable. For such situations, we should employ Fisher's exact test.

- The odds ratio in a 2 × 2 table is defined as

$$\psi = \frac{p_1/(1-p_1)}{p_2/(1-p_2)} = \frac{p_1(1-p_2)}{p_2(1-p_1)}.$$

- $\psi = 1$ if and only if the row variable and column variable are independent. Therefore, the null hypothesis for Fisher's exact test is H_0: $\psi = 1$.

```
==============================================================
> fisher.test(pain.data)

        Fisher's Exact Test for Count Data

data:  pain.data
p-value = 0.04718
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  0.9006803 57.2549701
sample estimates:
odds ratio
  6.180528
**************************************************************
```

4.3 McNemar's test for two paired proportions

4.3.1 Confidence interval estimation and hypothesis test

36• Background

- There are several cases in which we may observe two proportions on the same individuals.

- For example, we may wish to compare the pain relief by two different drugs in the same subjects (see Table 4.2) or to compare the proportions of subjects with a particular symptom before and after treatment (see Table 4.3).

- A statistically identical problem arises when we wish to compare one characteristic in two pair-matched groups.

Table 4.2 *Pattern 1 for two raw paired-data*

Subject	Drug A	Drug B
1	1	0
2	1	1
3	0	0
4	0	1
⋮	⋮	⋮
n	0	1

NOTE: 1 = pain relief (yes); 0 = pain relief (no).

Table 4.3 *Pattern 2 for two raw paired-data*

Subject	Before Treatment	After Treatment
1	1	0
2	1	1
3	0	0
4	0	1
⋮	⋮	⋮
n	0	1

NOTE: 1 = with symptom; 0 = without symptom.

37• Summary of the paired observations

- We can summarize the paired observations into four groups, according to whether the characteristic is present or not in each member of the pair, as shown in Table 4.4 or Table 4.5.

Table 4.4 *Frequency and probability of each combination of paired characteristics*

Observation		Number	Cell
Group 1	Group 2	of Pairs	Probability
Present	Present	a	θ_1
Present	Absent	b	θ_2
Absent	Present	c	θ_3
Absent	Absent	d	θ_4
Total		n	1

Table 4.5 *Frequency and probability in a general 2×2 table*

	Group 2		
Group 1	Present	Absent	Total
Present	$a\,(\theta_1)$	$b\,(\theta_2)$	$a+b\,(\theta_1+\theta_2)$
Absent	$c\,(\theta_3)$	$d\,(\theta_4)$	$c+d\,(\theta_3+\theta_4)$
Total	$a+c\,(\theta_1+\theta_3)$	$b+d\,(\theta_2+\theta_4)$	$n\,(1)$

38• Confidence interval

38.1• Point estimate of $p_1 - p_2$

— Suppose that we want to find a CI for the difference between two proportions p_1 and p_2, where the two groups of observations are not independent.

— Note that

$$\begin{aligned} p_1 &= \text{Pr(Present in Group 1)} = \theta_1 + \theta_2 \quad \text{and} \\ p_2 &= \text{Pr(Present in Group 2)} = \theta_1 + \theta_3, \end{aligned}$$

we obtain $p_1 - p_2 = \theta_2 - \theta_3$.

— Therefore, the point estimate of $p_1 - p_2$ is

$$\hat{p}_1 - \hat{p}_2 = \hat{\theta}_2 - \hat{\theta}_3 = \frac{b}{n} - \frac{c}{n} = \frac{b-c}{n}. \tag{4.32}$$

38.2• Standard error of $\hat{p}_1 - \hat{p}_2$

— To derive the estimate of the standard error of $\hat{p}_1 - \hat{p}_2$, we note that

$$(a, b, c, d)^\top \sim \text{Multinomial}(n; \theta_1, \theta_2, \theta_3, \theta_4),$$

so that $E(a) = n\theta_1$, $\text{Var}(a) = n\theta_1(1-\theta_1)$ and $\text{Cov}(a, b) = -n\theta_1\theta_2$.

— Thus,

$$\begin{aligned} \text{Var}(\hat{p}_1 - \hat{p}_2) &= \frac{1}{n^2}\text{Var}(b-c) \\ &= \frac{1}{n^2}[\text{Var}(b) + \text{Var}(c) - 2\text{Cov}(b, c)] \\ &= \frac{1}{n^2}[n\theta_2(1-\theta_2) + n\theta_3(1-\theta_3) + 2n\theta_2\theta_3], \end{aligned}$$

and its estimate is given by

$$\begin{aligned}\widehat{\mathrm{Var}}(\hat{p}_1-\hat{p}_2) &= \frac{1}{n^2}\left[b(1-\frac{b}{n})+c(1-\frac{c}{n})+2\frac{bc}{n}\right], \\ &= \frac{1}{n^2}\left[b+c-\frac{(b-c)^2}{n}\right].\end{aligned}$$

— The estimate of the standard error of $\hat{p}_1-\hat{p}_2$ is

$$\widehat{\mathrm{Se}}(\hat{p}_1-\hat{p}_2)=\frac{1}{n}\sqrt{b+c-\frac{(b-c)^2}{n}}. \tag{4.33}$$

38.3• Confidence interval of p_1-p_2

— The $(1-\alpha)100\%$ CI for p_1-p_2 is thus obtained as

$$\hat{p}_1-\hat{p}_2 \;\pm\; z_{\alpha/2}\,\widehat{\mathrm{Se}}(\hat{p}_1-\hat{p}_2). \tag{4.34}$$

39• The z test

39.1• A derivation

— Consider the following hypotheses:

$$H_0\colon\ p_1=p_2 \quad \text{or} \quad (\theta_2=\theta_3) \quad \text{versus} \quad H_1\colon\ p_1\neq p_2.$$

— When H_0 is true, we replace both b and c by $(b+c)/2$ in (4.33), resulting in

$$\widehat{\mathrm{Se}}(\hat{p}_1-\hat{p}_2)=\frac{\sqrt{b+c}}{n}. \tag{4.35}$$

— The z value for the test statistic Z is

$$z=\frac{\hat{p}_1-\hat{p}_2}{\widehat{\mathrm{Se}}(\hat{p}_1-\hat{p}_2)}=\frac{(b-c)/n}{\sqrt{b+c}/n}=\frac{b-c}{\sqrt{b+c}}, \tag{4.36}$$

which is free from a and d.

— The two-sided p-value is $2\,\mathrm{Pr}(Z>|z|)$.

39.2• An alternative derivation of (4.36)

— It is clear that a and d showing agreement do not appear in (4.36).

— Let us look at the total number of disagreements $b + c$.

— Under H_0, we expect the number of 'Present–Absent' and 'Absent–Present' pairs to be the same.

— So we can evaluate the probability of observing b out of $b + c$, i.e.,

$$b|H_0 \sim \text{Binomial}(b + c,\ 0.5).$$

— Because $p = 0.5$, the normal approximation to the binomial distribution is very good even for quite small sample sizes.

— Therefore, the z value for the test statistic Z is

$$z = \frac{b - E(b)}{\sqrt{\text{Var}(b)}} = \frac{b - (b + c)/2}{\sqrt{b + c/2}} = \frac{b - c}{\sqrt{b + c}},$$

which is identical to (4.36).

39.3• Continuity correction

— Similar to (4.5), the z value <u>w</u>ith <u>c</u>ontinuity <u>c</u>orrection gives

$$z_{\text{wcc}} = \frac{|b - E(b)| - 0.5}{\sqrt{\text{Var}(b)}} = \frac{|b - c| - 1}{\sqrt{b + c}}. \tag{4.37}$$

40• McNemar's test

- McNemar's test statistic χ^2 is computed as

$$\chi^2 = z^2 = \frac{(b - c)^2}{b + c}. \tag{4.38}$$

- Under the null hypothesis H_0, χ^2 has an asymptotic chi-squared distribution with one degree of freedom.

4.3.2 Example 4.4

41• Data

- Karacan *et al.* (1976) compared a group of 32 marijuana users with 32 matched controls with respect to their sleeping difficulties (see Table 4.6).

- Seven of the marijuana users reported sleeping difficulties sometimes or always compared with 13 of the controls.

Table 4.6 *Numbers of marijuana users and matched controls reporting sleeping difficulties*

Sleeping Difficulties		Number
Marijuana (Case) Group	Control Group	of Pairs
Present	Present	$a = 4$
Present	Absent	$b = 3$
Absent	Present	$c = 9$
Absent	Absent	$d = 16$
Total		$n = 32$

42• Simple analysis

- From (4.34), the 95% CI for $p_1 - p_2$ is

  $$\frac{3-9}{32} \pm 1.96\,\frac{\sqrt{3+9-(3-9)^2/32}}{32} = -0.1875 \pm 1.96 \times 0.1031,$$

  or $[-0.38958, 0.014576]$, which includes zero.

- From (4.36), we obtain $z = (3-9)/\sqrt{(3+9)} = -1.7321$, so that the p-value $= 2(1 - \Phi(1.7321)) = 0.0832 > 0.05$.

- From (4.37), we obtain $z_{\text{wcc}} = (|3-9|-1)/\sqrt{(3+9)} = 1.443376$, so that the p-value $= 2(1 - \Phi(1.443376)) = 0.1489 > 0.05$.

- We cannot reject the null hypothesis H_0: $p_1 = p_2$ at the 5% level.

43• Using the built-in R function mcnemar.test()

```
==============================================================
> marijuana <- matrix(c(4, 3, 9, 16), nrow=2, byrow=T)
> marijuana
     [,1] [,2]
[1,]    4    3
[2,]    9   16
```

```
--------------------------------------------------------------
> mcnemar.test(marijuana, correct = F)

        McNemar's Chi-squared test

data:  marijuana
McNemar's chi-squared = 3, df = 1, p-value = 0.0832
--------------------------------------------------------------
> mcnemar.test(marijuana, correct = T)

        McNemar's Chi-squared test with continuity correction

data:  marijuana
McNemar's chi-squared = 2.0833, df = 1, p-value = 0.1489
**************************************************************
```

4.4 Tests for $r \times c$ contingency tables

4.4.1 The χ^2 goodness-of-fit test for one-way tables

44• One-way table

- Let $X_1, \ldots, X_n$ be classified into m categories.
- Let f_i and p_i denote the frequency and probability of category i.
- We have $(f_1, \ldots, f_m)^\top \sim \text{Multinomial}(n; p_1, \ldots, p_m)$.
- That is, the multinomial distribution is closely related to the one-way table as shown in Table 4.7.

Table 4.7 *The general one-way table*

Category	C_1	C_2	$\cdots$	C_m	Total
Frequency	f_1	f_2	$\cdots$	f_m	$n = \sum_{i=1}^m f_i$
Probability	p_1	p_2	$\cdots$	p_m	1

45• The chi-squared goodness-of-fit test

- For one-way frequency tables, the built-in R function `chisq.test()` provides a chi-squared goodness-of-fit test.

- The null hypothesis is

$$H_0\colon\ (p_1, \ldots, p_m) = (p_{10}, \ldots, p_{m0}), \tag{4.39}$$

  where $p_{10}, \ldots, p_{m0}$ are given probabilities.

- The chi-squared statistic is computed as

$$\chi^2 = \sum_{i=1}^{m} \frac{(f_i - e_i)^2}{e_i}, \tag{4.40}$$

  where $e_i = np_{i0}$ is the expected frequency for class i under H_0.

- Under H_0, we have $\chi^2 \mathrel{\dot\sim} \chi^2(m-1)$.

46$^\bullet$ Using the built-in R function chisq.test()

```
==============================================================
> f <- c(89, 37, 30, 28, 2)
> p0 <- c(40, 20, 20, 15, 5)   # sum(p0) not = 1
> chisq.test(f, p= p0, rescale.p = TRUE)

        Chi-squared test for given probabilities

data:  f
X-squared = 9.9901, df = 4, p-value = 0.04059
--------------------------------------------------------------
> p0 <- c(0.40, 0.20, 0.20, 0.15, 0.05)
> chisq.test(f, p= p0)            #  by default p= rep(1/5, 5)

        Chi-squared test for given probabilities

data:  f
X-squared = 9.9901, df = 4, p-value = 0.04059
--------------------------------------------------------------
> p0 <- c(0.40, 0.20, 0.20, 0.19, 0.01)
> sum(f)*0.01
[1] 1.86
#  Expected count in category 5 is 1.86 < 5,
#  implying that the chi-squared approximation may be doubtful
```

```
---------------------------------------------------------------
> chisq.test(f, p= p0, simulate.p.value = TRUE)

        Chi-squared test for given probabilities with
        simulated p-value (based on 2000 replicates)

data:  f
X-squared = 5.7947, df = NA, p-value = 0.2194
---------------------------------------------------------------
> chisq.test(f, p= p0)

        Chi-squared test for given probabilities

data:  f
X-squared = 5.7947, df = 4, p-value = 0.215
---------------------------------------------------------------
> chisq.test(f, p = p0, simulate.p.value = TRUE, B= 20000)

        Chi-squared test for given probabilities with
        simulated p-value (based on 20000 replicates)

data:  f
X-squared = 5.7947, df = NA, p-value = 0.2071
***************************************************************
```

4.4.2 Pearson's chi-squared and Fisher's exact tests for 2 × 2 tables

47• Two by two table

- Let O_{ij} and π_{ij} denote the count and probability in the cell (i, j).
- The general 2×2 table can be summarized in Tables 4.8 or 4.9.

Table 4.8 *General* 2×2 *contingency table*

Variable A	Variable B		Total
	1	2	
1	$O_{11}\,(\pi_{11})$	$O_{12}\,(\pi_{12})$	$O_{1\cdot}\,(\pi_{1\cdot})$
2	$O_{21}\,(\pi_{21})$	$O_{22}\,(\pi_{22})$	$O_{2\cdot}\,(\pi_{2\cdot})$
Total	$O_{\cdot 1}\,(\pi_{\cdot 1})$	$O_{\cdot 2}\,(\pi_{\cdot 2})$	$n\,(1)$

Table 4.9 *General* 2×2 *contingency table*

Variable A	Variable B		Total
	1	2	
1	$a\,(\pi_{11})$	$b\,(\pi_{12})$	$a+b\,(\pi_{1\cdot})$
2	$c\,(\pi_{21})$	$d\,(\pi_{22})$	$c+d\,(\pi_{2\cdot})$
Total	$a+c\,(\pi_{\cdot 1})$	$b+d\,(\pi_{\cdot 2})$	$n\,(1)$

48• The chi-squared independence test

48.1• A derivation

— We have $(O_{11}, O_{12}, O_{21}, O_{22})^{\top} \sim \text{Multinomial}(n; \pi_{11}, \pi_{12}, \pi_{21}, \pi_{22})$.

— So $E_{ij} \mathrel{\hat{=}} E(O_{ij}) = n\pi_{ij}$ for $i, j = 1, 2$.

— H_0: A and B are independent or $\pi_{ij} = \pi_{i\cdot}\pi_{\cdot j}$.

— Under H_0, $E_{ij} = n\pi_{i\cdot}\pi_{\cdot j}$ can be estimated by $O_{i\cdot}O_{\cdot j}/n$.

— From (4.40), we know that the chi-squared statistic under H_0 is

$$X^2 = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{n(ad - bc)^2}{(a+b)(c+d)(a+c)(b+d)}, \tag{4.41}$$

where $O_{11} \mathrel{\hat{=}} a$, $O_{12} \mathrel{\hat{=}} b$, $O_{21} \mathrel{\hat{=}} c$ and $O_{22} \mathrel{\hat{=}} d$ as in Table 4.9.

— Under H_0, $X^2 \mathrel{\dot{\sim}} \chi^2(1)$.

— The p-value is $\Pr\{\chi^2(1) > x^2\}$, where x^2 denotes the observed value of the test statistic X^2 specified by (4.41).

— When p-value $\geqslant \alpha$, we cannot reject the null hypothesis H_0.

48.2• Continuity correction

— In the context of 2 × 2 tables, the continuity correction is known as *Yates' correction* after the statistician who devised it.

— We replace $O_{ij} - E_{ij}$ by $|O_{ij} - E_{ij}| - 0.5$.

— From (4.41), the chi-squared statistic X^2 with Yates' correction is

$$X^2_{\text{wyc}} = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{(|O_{ij} - E_{ij}| - 0.5)^2}{E_{ij}}. \tag{4.42}$$

49• Fisher's exact test

49.1• Background

— If the expected frequency of any cell is less than 5, then the χ^2 approximation may be questionable.

— For such situations, a Fisher's exact test will be used instead and it is valid for all sample sizes.

— It is named after its inventor, R. A. Fisher, and is one of a class of exact tests, so called because the significance of the deviation from a null hypothesis can be calculated exactly, rather than relying on an approximation that becomes exact in the limit as the sample size grows to infinity.

49.2• The procedure

— For simplicity, we use the notations in Table 4.9.

— When H_0 is true and the row & column totals (i.e., $a+b, c+d, a+c, b+d$) are fixed, the probability of obtaining the observed cell counts (a, b, c, d) is calculated as

$$p_{\text{obs}} = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}}. \tag{4.43}$$

— In R, we can compute p_{obs} via `dhyper(a, a+b, c+d, a+c)`.

— The method consists of evaluating the probability associated with all possible 2×2 tables which have the same row and column totals as the observed data, making the assumption that the null hypothesis is true.

— The p-value of the test can be simply computed by the sum of all probabilities which are less than or equal to p_{obs}.

49.3$^\bullet$ An illustration

— Let the observed 2×2 table be as follows:

```
                  Total
------|--------|------
      | 5   0  |   5
      |        |
      | 1   4  |   5
------|--------|------
Total | 6   4  |  10
```

— From (4.43), we have $p_{\text{obs}} = 0.02380952$.

— The other possible matrices (with the same row and column totals as the observed data)

$$\begin{pmatrix} 4 & 1 \\ 2 & 3 \end{pmatrix}, \quad \begin{pmatrix} 3 & 2 \\ 3 & 2 \end{pmatrix}, \quad \begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 4 \\ 5 & 0 \end{pmatrix},$$

and their probabilities are $p_1 = 0.2380952$, $p_2 = 0.4761905$, $p_3 = 0.2380952$ and $p_4 = 0.02380952$, which indeed sum to 1.

— The p-value is $p_{\text{obs}} + p_4 = 0.04761904 < 0.05$, we reject H_0.

50$^\bullet$ Using the built-in R functions chisq.test() & fisher.test()

```
===============================================================
> M <- matrix(c(5, 0, 1, 4), nrow=2, byrow=T)
> M
     [,1] [,2]
[1,]    5    0
[2,]    1    4
---------------------------------------------------------------
```

```
> chisq.test(M, correct= F)

        Pearson’s Chi-squared test

data:  M
X-squared = 6.6667, df = 1, p-value = 0.009823

Warning message:
In chisq.test(M, correct = F) : Chi-squared approximation
may be incorrect
--------------------------------------------------------------
> chisq.test(M, correct= T)

        Pearson’s Chi-squared test with
        Yates’ continuity correction

data:  M
X-squared = 3.75, df = 1, p-value = 0.05281

Warning message:
In chisq.test(M, correct = T) : Chi-squared approximation
may be incorrect
--------------------------------------------------------------
> fisher.test(M)

        Fisher’s Exact Test for Count Data

data:  M
p-value = 0.04762
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.024822      Inf
sample estimates:
odds ratio
       Inf
**************************************************************
```

4.4.3 Equivalence of the two-sample proportions test and the Pearson chi-squared test

51• Purpose of this subsection

- We will show that the two-sample proportions test presented in §4.2 is equivalent to the Pearson chi-squared test for a 2×2 table.

- In other words, the test for the difference of two proportions being zero in two independent Bernoulli populations is also a test for independency between the row variable and the column variable in a 2×2 table.

52• Proof of the equivalence

- By expressing the comparison of two proportions in the notation of Table 4.9, we have

$$\hat{p}_1 = \frac{a}{a+b}, \qquad \hat{p}_2 = \frac{c}{c+d},$$

  and the pooled proportion is $\hat{p} = (a+c)/n$.

- From (5.25), the z value of the test statistic Z for comparing the two proportions is

$$\begin{aligned} z &= \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})/m}} \\ &= \frac{\dfrac{a}{a+b} - \dfrac{c}{c+d}}{\sqrt{\dfrac{(a+c)}{n}\dfrac{(b+d)}{n}\left(\dfrac{1}{a+b} + \dfrac{1}{c+d}\right)}} \\ &= \sqrt{\frac{n(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)}} \\ &\overset{(4.41)}{=} \sqrt{X^2}; \end{aligned}$$

  that is, the z value is the square root of the value of X^2.

- The two tests are equivalent because $\chi^2(1) \overset{d}{=} Z^2$, where $Z \dot{\sim} N(0,1)$ under H_0.

4.4.4 The chi-squared independence test for an $r \times c$ table

53• An $r \times c$ table

- Let O_{ij} and π_{ij} denote the count and probability in the cell (i, j).
- The general $r \times c$ table can be summarized in Table 4.10.

Table 4.10 *General $r \times c$ contingency table*

Variable A	Variable B			Total
	1	$\cdots$	c	
1	$O_{11}\,(\pi_{11})$	$\cdots$	$O_{1c}\,(\pi_{1c})$	$O_{1\cdot}\,(\pi_{1\cdot})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
r	$O_{r1}\,(\pi_{r1})$	$\cdots$	$O_{rc}\,(\pi_{rc})$	$O_{r\cdot}\,(\pi_{r\cdot})$
Total	$O_{\cdot 1}\,(\pi_{\cdot 1})$	$\cdots$	$O_{\cdot c}\,(\pi_{\cdot c})$	$n\,(1)$

54• The chi-squared independence test

- We have $\{O_{ij}\} \sim \text{Multinomial}(n; \{\pi_{ij}\})$.
- So $E_{ij} \mathrel{\hat{=}} E(O_{ij}) = n\pi_{ij}$ for $i = 1, \dots, r$ and $j = 1, \dots, c$.
- H_0: A and B are independent or $\pi_{ij} = \pi_{i\cdot}\pi_{\cdot j}$.
- Under H_0, $E_{ij} = n\pi_{i\cdot}\pi_{\cdot j}$ can be estimated by $O_{i\cdot}O_{\cdot j}/n$.
- From (4.40), we know that the chi-squared statistic under H_0 is

$$X^2 = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \sum \frac{(O - E)^2}{E}. \tag{4.44}$$

- Under H_0, $X^2 \mathrel{\dot\sim} \chi^2((r-1)(c-1))$.

55• An example

55.1• Caffeine consumption data

— We consider the table with caffeine consumption by marital status from §2.5 and perform the χ^2 test.

55.2• R output

```
==============================================================
> caff.marital
               0 1-150 151-300 >300
Married      652  1537     598  242
Prev.married  36    46      38   21
Single       218   327     106   67
--------------------------------------------------------------
> chisq.test(caff.marital)

        Pearson's Chi-squared test

data:  caff.marital
X-squared = 51.656, df = 6, p-value = 2.187e-09
--------------------------------------------------------------
> chisq.test(caff.marital)$observed
               0 1-150 151-300 >300
Married      652  1537     598  242
Prev.married  36    46      38   21
Single       218   327     106   67
--------------------------------------------------------------
> chisq.test(caff.marital)$expected
                     0      1-150   151-300      >300
Married      705.83179 1488.01183 578.06533 257.09105
Prev.married  32.85648   69.26698  26.90895  11.96759
Single       167.31173  352.72119 137.02572  60.94136
--------------------------------------------------------------
> O <- chisq.test(caff.marital)$observed
> E <- chisq.test(caff.marital)$expected
> (O-E)/sqrt(E)
                      0     1-150    151-300       >300
Married      -2.0262275  1.269954  0.8291261 -0.9411871
Prev.married  0.5484102 -2.795611  2.1380815  2.6109594
Single        3.9187205 -1.369542 -2.6504574  0.7761028
--------------------------------------------------------------
> chisq.test(caff.marital)$residuals
                      0     1-150    151-300       >300
```

```
Married      -2.0262275  1.269954  0.8291261 -0.9411871
Prev.married  0.5484102 -2.795611  2.1380815  2.6109594
Single        3.9187205 -1.369542 -2.6504574  0.7761028
**************************************************************
```

4.5 K-sample proportions test and chi-squared test for trend

4.5.1 K-sample proportions test for unordered categories

56• A $2 \times K$ table

- Let (r_k, f_k, n_k) and p_k denote the numbers of successes, failures, trials and the probability of success in the k-th group, $k = 1, 2, \ldots, K$.
- The aim is to test H_0: $p_1 = p_2 = \cdots = p_K$.
- The observations can be summarized in the form of $2 \times K$ table as shown in Table 4.11.

Table 4.11 *Comparison of proportions from K independent groups in the form of a $2 \times K$ table*

Presence or absence	Group			
of a symptom	1	2	$\cdots$	K
Yes	r_1	r_2	$\cdots$	r_K
No	f_1	f_2	$\cdots$	f_K
Total	n_1	n_2	$\cdots$	n_K

57• An example

57.1• Eye strain data for four types of office workers

— The data in Table 4.12 below are from a study carried out to assess possible harmful effects of using *visual display units* (VDUs) (i.e. computer monitors).

— Four types of work include (i) Data entry in VDUs, (ii) Conversational use of VDUs, (iii) Full-time typing; and (iv) Traditional office work (clerical).

— H_0: There is no difference in the proportions reporting eye strain in the four types of work.

Table 4.12 *Eye strain reported by four groups of office workers*

	Type of work				
Number	W.type 1	W.type 2	W.type 3	W.type 4	Total
with eye strain	11	30	14	3	58
without eye strain	42	79	64	52	237
Total	53	109	78	55	295

57.2• R output

```
==============================================================
> eye.worktype <- matrix(c(11, 42, 30, 79, 14, 64, 3, 52),
+                                         nrow=2, byrow=F)
> eye.worktype
     [,1] [,2] [,3] [,4]
[1,]   11   30   14    3
[2,]   42   79   64   52
> rownames(eye.worktype) <- c("Number with eye strain",
+                             "Number without eye strain")
> colnames(eye.worktype) <- c("W.type 1", "W.type 2",
+                             "W.type 3", "W.type 4")
# ---------------- W.type = work type ------------------------
> eye.worktype
                          W.type 1 W.type 2 W.type 3 W.type 4
Number with eye strain          11       30       14        3
Number without eye strain       42       79       64       52
--------------------------------------------------------------
> is.matrix(eye.worktype)
[1] TRUE
**************************************************************
```

58• Using the built-in R function prop.test()

58.1• Comments

— The built-in R function `prop.test()` can also be used to compare K ($K > 2$) proportions.

— If $K > 2$, the alternative is always “two.sided”, the returned confidence interval is NULL, and continuity correction is never used.

— In `prop.test()`, the arguments should be given as two vectors, where the first is a vector of “successes” and the second is a vector of “trials”.

58.2$^{\bullet}$ R output

```
==============================================================
> succ <- eye.worktype["Number with eye strain", ]
> succ
W.type 1 W.type 2 W.type 3 W.type 4
      11       30       14        3
> is.vector(succ)
[1] TRUE
--------------------------------------------------------------
> total <- margin.table(eye.worktype, 2)
> total
W.type 1 W.type 2 W.type 3 W.type 4
      53      109       78       55
> is.vector(total)
[1] FALSE
> is.matrix(total)
[1] FALSE
> is.table(total)
[1] FALSE
> is.factor(total)
[1] FALSE
> is.array(total)
[1] TRUE
> dim(total)
[1] 4
> sum(total)   # Using like a vector
[1] 295        # sum(c(53, 109, 78, 55))
--------------------------------------------------------------
> margin.table(eye.worktype, 1)
```

```
   Number with eye strain    Number without eye strain
                       58                          237
> sum(margin.table(eye.worktype, 1))
[1] 295
--------------------------------------------------------------
> prop.test(succ, total)

        4-sample test for equality of proportions
        without continuity correction

data:  succ out of total
X-squared = 11.478, df = 3, p-value = 0.009404
alternative hypothesis: two.sided
sample estimates:
    prop 1     prop 2     prop 3     prop 4
0.20754717 0.27522936 0.17948718 0.05454545
**************************************************************
```

59• Using the built-in R function chisq.test()

59.1• Comments

— Of course, the Pearson chi-squared test can be used to test H_0: the row variable and the column variable in a $2 \times K$ table are independent.

— Similar to §4.4.3, we could show that the K-sample proportions test is equivalent to the Pearson chi-squared test for a $2 \times K$ table.

— In other words, the test for the equality of K proportions in K independent groups is also a test for independency between the row variable and the column variable in a $2 \times K$ table.

— So, for a $2 \times K$ table, the built-in R function `chisq.test()` is exactly equivalent to `prop.test()`.

59.2• R output

```
==============================================================
> chisq.test(eye.worktype)
```

```
        Pearson's Chi-squared test

data:  eye.worktype
X-squared = 11.478, df = 3, p-value = 0.009404
***************************************************************
```

4.5.2 Chi-squared test for ordered categories

60• Backgroud

- If there is a meaningful order to the K groups, then the chi-squared test for trend provides a more powerful test than the unordered independence test above.
- The built-in R function `prop.trend.test(x, n, score)` performs a test for linear trend across the K groups, where `(x, n)` are exactly same as those in `prop.test(x, n)` while the last one is the score given to the groups, by default simply $1, 2, \ldots, K$.
- The basis of the test is essentially a weighted linear regression of the proportions on the group score, where we test for a *zero slope*, which becomes a χ^2 test with 1 degree of freedom.

61• The procedure

- We use the same notations as in Table 4.11.
- Let s_k be the score allocated to group k, $k = 1, 2, \ldots, K$.
- The test statistic is

$$X^2_{\text{trend}} = \frac{\left(\sum_{k=1}^K r_k s_k - r\bar{s}\right)^2}{p(1-p)\left(\sum_{k=1}^K n_k s_k^2 - n\bar{s}^2\right)} \dot{\sim} \chi^2(1), \tag{4.45}$$

where $n = \sum_{k=1}^K n_k$, $r = \sum_{k=1}^K r_k$, $p = r/n$ and $\bar{s} = (1/n)\sum_{k=1}^K n_k s_k$.

62• An example

62.1• Caesarean by shoe size data

— Table 4.13 reported the frequency of babies delivered by Caesarean section to maternal shoe size.

— The rationale of this study was that small shoe size is a simple indicator of possible birth difficulty due to a small pelvis.

— The aim is to find if there is a decreasing trend in the proportions with the shoe sizes.

Table 4.13 *Relation between frequency of Caesarean section and maternal shoe size*

Caesarean	Maternal shoe size						
section	< 4	4	$4\frac{1}{2}$	5	$5\frac{1}{2}$	6+	Total
Yes	5	7	6	7	8	10	43
No	17	28	36	41	46	140	308
Total	22	35	42	48	54	150	351

62.2• R output from prop.test() and chisq.test()

```
==============================================================
> caesar.shoe <- matrix(c(5, 7, 6, 7, 8, 10,
+                 17, 28, 36, 41, 46, 140), nrow=2, byrow=T)
> caesar.shoe
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    5    7    6    7    8   10
[2,]   17   28   36   41   46  140
> colnames(caesar.shoe) <- c("<4", "4", "4.5", "5", "5.5",
                                                    "6+")
> rownames(caesar.shoe) <- c("Yes", "No")
> caesar.shoe
    <4  4 4.5  5 5.5  6+
Yes  5  7   6  7   8  10
No  17 28  36 41  46 140
--------------------------------------------------------------
> caesar.shoe.yes <- c(5, 7, 6, 7, 8, 10)
> caesar.shoe.total <- c(22, 35, 42, 48, 54, 150)
> prop.test(caesar.shoe.yes, caesar.shoe.total)
```

```
        6-sample test for equality of proportions
        without continuity correction

data:  caesar.shoe.yes out of caesar.shoe.total
X-squared = 9.2874, df = 5, p-value = 0.09814
alternative hypothesis: two.sided
sample estimates:
   prop 1    prop 2    prop 3    prop 4    prop 5    prop 6
0.2272727 0.2000000 0.1428571 0.1458333 0.1481481 0.0666667

Warning message:
Chi-squared approximation may be incorrect in:
prop.test(caesar.shoe.yes, caesar.shoe.total)
--------------------------------------------------------------
> chisq.test(caesar.shoe)

        Pearson's Chi-squared test

data:  caesar.shoe
X-squared = 9.2874, df = 5, p-value = 0.09814

Warning message:
Chi-squared approximation may be incorrect in:
chisq.test(caesar.shoe)
--------------------------------------------------------------
> chisq.test(caesar.shoe)$expected
          <4        4      4.5        5      5.5       6+
Yes  2.69515  4.28774  5.14529  5.88034  6.61538  18.3760
No  19.30484 30.71225 36.85470 42.11965 47.38461 131.6239
Warning message:
Chi-squared approximation may be incorrect in:
chisq.test(caesar.shoe)
**************************************************************
```

62.3• Comments on above results

— Since the p-value $= 0.09814 > 0.05$, we cannot reject H_0: $p_1 = p_2 = \cdots p_6$.

— However, the point estimates of $\{p_k\}$ really exhibit a decreasing trend.

— The warning message told us that the chi-squared approximation may be incorrect.

— In fact, we found some cells having an expected count less than 5.

— Therefore, `prop.trend.test()` can be used to test for a trend in the proportions.

62.4• R output from prop.trend.test()

```
==============================================================
> x <- caesar.shoe.yes
> n <- caesar.shoe.total
> prop.trend.test(x, n, score= 1:6)

        Chi-squared Test for Trend in Proportions

data:  caesar.shoe.yes out of caesar.shoe.total ,
 using scores: 1 2 3 4 5 6
X-squared = 8.0237, df = 1, p-value = 0.004617
**************************************************************
```

62.5• Comments on above results

— So if we assume that the effect of shoe size is linear in the group score, then we can see a significant difference, since the p-value $= 0.004617 < 0.05$.

— Therefore, the proportions in 6 groups really exhibit a decreasing trend.

Chapter 5

Analysis of Variance

1• Aims in statistical aspects

- In this chapter, we consider one- and two-way *analysis of variance* (ANOVA) problems for more than two independent populations under the assumptions of normality and homogeneity of variances.

- Next we consider the non-parametric Kruskal–Wallis rank sum test and Friedman rank sum test.

2• Aims in software aspects

2.1• Introduction of five R functions

— `lm(y ~ f)` for one-way ANOVA and `lm(y ~ f1 + f2)` for two-way ANOVA.

— `anova()` for ANOVA with equal variances.

— `oneway.test()` for one-way ANOVA with unequal variances.

— `bartlett.test()` for Bartlett's test of the equality of k variances.

— `pairwise.t.test()` for pairwise comparisons.

2.2• Two R functions for the non-parametric case

— `kruskal.test()` for Kruskal–Wallis rank sum test.

— `friedman.test()` for Friedman rank sum test.

5.1 One-way analysis of variance

5.1.1 Aim and assumptions

3• Background

- Suppose that we have k *independent* populations (or groups, or treatments).
- Independent random samples are drawn from the k independent populations and the continuous response variable Y is observed for each randomly selected individual.
- In the *analysis of variance* (ANOVA), one is interested in testing equality in means for $k\,(k \geqslant 3)$ independent populations.
- The one-way ANOVA is an extension of the two-independent-sample t test to a k independent-sample F test.

4• Objective

- The one-way ANOVA is used to test the null hypothesis that the means of k independent normal populations with a common variance are identical, i.e.,
  $$H_0\colon\ \mu_1 = \cdots = \mu_k \tag{5.1}$$
  against
  $$H_1\colon\ \text{at least two means are not equal.}$$
- For the details, please see §5.2.

5• Data structure

- Let
  $$\begin{aligned} Y_{11}, \ldots, Y_{1n_1} &\overset{\text{iid}}{\sim} N(\mu_1, \sigma^2), \\ &\ \vdots \\ Y_{i1}, \ldots, Y_{in_i} &\overset{\text{iid}}{\sim} N(\mu_i, \sigma^2), \\ &\ \vdots \\ Y_{k1}, \ldots, Y_{kn_k} &\overset{\text{iid}}{\sim} N(\mu_k, \sigma^2), \end{aligned} \tag{5.2}$$
  and the k samples be independent.

- Let y_{ij} denote the realization of Y_{ij} for $i = 1, \ldots, k$ and $j = 1, \ldots, n_i$.

6• Three assumptions

- **Independence**. This assumption will probably hold if the observations are drawn independently and are not drawn over time.
- **Constant variance**. See §5.3.
- **Normality**. See §5.6.

5.1.2 Model and MLEs of parameters

7• The one-way anova model

- The model (5.2) can be rewritten as

$$Y_{ij} = \mu_i + \varepsilon_{ij}, \quad i = 1, \ldots, k; \quad j = 1, \ldots, n_i. \tag{5.3}$$

where

$$\{\varepsilon_{ij}\} \overset{\text{iid}}{\sim} N(0, \sigma^2). \tag{5.4}$$

8• MLEs

- Under the assumptions of (5.3) and (5.4), the likelihood function of $\mu_1, \ldots, \mu_k$ and σ^2 for the observed data $Y_{\text{obs}} = \{y_{ij}\colon i = 1, \ldots, k; j = 1, \ldots, n_i\}$ is

$$\begin{aligned} L(\mu_1, \ldots, \mu_k, \sigma^2) &= \prod_{i=1}^{k} \prod_{j=1}^{n_i} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y_{ij}-\mu_i)^2}{2\sigma^2}\right\} \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left\{-\frac{\sum_{i=1}^k \sum_{j=1}^{n_i}(y_{ij}-\mu_i)^2}{2\sigma^2}\right\}, \end{aligned}$$

where $n = \sum_{i=1}^k n_i$ denotes the total number of observations.

- The log-likelihood function is given by

$$\ell(\mu_1, \ldots, \mu_k, \sigma^2) = c - \frac{n}{2}\log\sigma^2 - \frac{\sum_{i=1}^k \sum_{j=1}^{n_i}(y_{ij}-\mu_i)^2}{2\sigma^2}.$$

- Let

$$\begin{aligned}\frac{\partial \ell(\mu_1,\ldots,\mu_k,\sigma^2)}{\partial \mu_i} &= 0, \quad i=1,\ldots,k, \quad \text{and}\\ \frac{\partial \ell(\mu_1,\ldots,\mu_k,\sigma^2)}{\partial \sigma^2} &= 0,\end{aligned}$$

  we obtain the MLEs of μ_i and σ^2:

$$\begin{aligned}\hat{\mu}_i &= \frac{1}{n_i}\sum_{j=1}^{n_i} y_{ij} \;\hat{=}\; \bar{y}_i, \quad i=1,\ldots,k, & (5.5)\\ \hat{\sigma}^2 &= \frac{\sum_{i=1}^k \sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2}{n}. & (5.6)\end{aligned}$$

9• Unbiased estimates

- Obviously, $\hat{\mu}_i$ is an unbiased estimate of μ_i.
- An unbiased estimate of σ^2 is

$$s^2 = \frac{\sum_{i=1}^k \sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2}{n-k} \;\hat{=}\, \text{MSE}. \qquad (5.7)$$

10• Alternative one-way anova model

- In model (5.3), we make the following one-to-one transformation

$$\begin{aligned}\mu_1 &= \mu,\\ &\vdots\\ \mu_i &= \mu+\alpha_i, \qquad (5.8)\\ &\vdots\\ \mu_k &= \mu+\alpha_k,\end{aligned}$$

  and then obtain

$$Y_{ij} = \mu+\alpha_i+\varepsilon_{ij}, \quad i=1,\ldots,k; \quad j=1,\ldots,n_i, \qquad (5.9)$$

  where $\alpha_1 = 0$.

- From (5.5), we have

$$\hat{\mu} = \hat{\mu}_1 = \bar{y}_1 \quad \text{and} \quad \hat{\alpha}_i = \bar{y}_i - \bar{y}_1, \quad i=2,\ldots,k. \qquad (5.10)$$

5.1.3 Relationship with the linear regression model

11• Linear model representation of the one-way anova model

- Let $\boldsymbol{e}_m^{(i)} = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ denote the m-dimensional base vector with the i-th element being 1 and the others being 0.

- Let $\boldsymbol{\beta} \mathrel{\hat{=}} (\mu, \alpha_2, \ldots, \alpha_k)^\top$, the model (5.9) can be rewritten as

$$\begin{aligned} Y_{1j} &= \mu + 0 \cdot \alpha_2 + \cdots + 0 \cdot \alpha_k + \varepsilon_{1j} \\ &= (1, 0, \ldots, 0)\boldsymbol{\beta} + \varepsilon_{1j}, \quad j = 1, \ldots, n_1, \quad \text{and} \\ Y_{ij} &= \mu + 0 \cdot \alpha_2 + \cdots + 1 \cdot \alpha_i + \cdots + 0 \cdot \alpha_k + \varepsilon_{ij} \\ &= (1, \boldsymbol{e}_{k-1}^{(i-1)\top})\boldsymbol{\beta} + \varepsilon_{ij}, \quad i = 2, \ldots, k; \quad j = 1, \ldots, n_i. \end{aligned}$$

- In matrix form, we have

$$\begin{aligned} \mathbf{y}_1 &\mathrel{\hat{=}} \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{1n_1} \end{pmatrix} = (\mathbf{1}_{n_1} \ \mathbf{O}_{n_1 \times (k-1)})\boldsymbol{\beta} + \begin{pmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{1n_1} \end{pmatrix} \\ &\mathrel{\hat{=}} \mathbf{X}_1\boldsymbol{\beta} + \boldsymbol{\varepsilon}_1, \quad \text{and} \\ \mathbf{y}_i &\mathrel{\hat{=}} \begin{pmatrix} Y_{i1} \\ \vdots \\ Y_{in_i} \end{pmatrix} = \begin{pmatrix} 1 & \boldsymbol{e}_{k-1}^{(i-1)\top} \\ \vdots & \vdots \\ 1 & \boldsymbol{e}_{k-1}^{(i-1)\top} \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{in_i} \end{pmatrix} \\ &\mathrel{\hat{=}} \mathbf{X}_i\boldsymbol{\beta} + \boldsymbol{\varepsilon}_i, \quad i = 2, \ldots, k. \end{aligned}$$

- Finally, we obtain

$$\mathbf{y} \mathrel{\hat{=}} \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{k-1} \\ \mathbf{y}_k \end{pmatrix} = \begin{pmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_{k-1} \\ \mathbf{X}_k \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \boldsymbol{\varepsilon}_1 \\ \vdots \\ \boldsymbol{\varepsilon}_{k-1} \\ \boldsymbol{\varepsilon}_k \end{pmatrix} \mathrel{\hat{=}} \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{5.11}$$

- All elements in the second column to the last column of the design matrix $\mathbf{X}$ in (5.11) are zeros or ones.

12• An illustration example

- For instance, when $k = 3$, $n_1 = 2$, $n_2 = 3$ and $n_3 = 4$, the model (5.11) becomes

$$\begin{pmatrix} Y_{11} \\ Y_{12} \\ Y_{21} \\ Y_{22} \\ Y_{23} \\ Y_{31} \\ Y_{32} \\ Y_{33} \\ Y_{34} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_2 \\ \alpha_3 \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \varepsilon_{21} \\ \varepsilon_{22} \\ \varepsilon_{23} \\ \varepsilon_{31} \\ \varepsilon_{32} \\ \varepsilon_{33} \\ \varepsilon_{34} \end{pmatrix}.$$

13• Difference between regression analysis and anova

- Regression analysis is to investigate the effect of one or more continuous independent variables on Y.

- ANOVA is to investigate the effect of one or more categorical variables/factors on Y.

5.2 Testing hypothesis for no group effect

5.2.1 *F* test

14• Aim

- The major objective of this section is to test the null hypothesis H_0 specified by (5.1).

- Equivalently, from (5.8), we only need to consider testing

$$H_0^*\colon\ \alpha_2 = \cdots = \alpha_k = 0 \tag{5.12}$$

against

$$H_1^*\colon \text{ at least one of } \alpha\text{'s is not zero.} \tag{5.13}$$

15• The overall/grand mean

- When H_0 or H_0^* is true (i.e., there is no group/treatment effect), from (5.9), we know that

$$\{Y_{ij}\} \overset{\text{iid}}{\sim} N(\mu, \sigma^2) \quad \forall\, i = 1, \ldots, k; \quad j = 1, \ldots, n_i.$$

- The parameter μ is estimated by the *overall/grand mean*:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{n_i} y_{ij} = \frac{1}{n} \sum_{i=1}^{k} n_i \bar{y}_i, \tag{5.14}$$

where $\bar{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij}$ denotes the sample mean of the i-th group.

16• Decomposition of the corrected total sum of squares (CTSS)

- We can decompose the observations as

$$y_{ij} = \bar{y} + (y_{ij} - \bar{y}_i) + (\bar{y}_i - \bar{y})$$

or equivalently

$$\underbrace{y_{ij} - \bar{y}}_{\substack{\text{deviation of}\\ \text{observation from}\\ \text{grand mean}}} \quad = \quad \underbrace{(y_{ij} - \bar{y}_i)}_{\substack{\text{deviation of}\\ \text{observation from}\\ \text{group mean}}} \quad + \quad \underbrace{(\bar{y}_i - \bar{y})}_{\substack{\text{deviation of}\\ \text{group mean from}\\ \text{grand mean}}}. \tag{5.15}$$

- The total variability of the data set is measured by the CTSS, which can be decomposed into the *sum of squares within groups* (SS_{WG}) plus the *sum of squares between groups* (SS_{BG}), i.e.,

$$\begin{aligned}
\text{CTSS} \quad &\hat{=} \quad \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2 \\
&\overset{(5.15)}{=} \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i + \bar{y}_i - \bar{y})^2 \\
&= \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 + \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\bar{y}_i - \bar{y})^2 \\
&= \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 + \sum_{i=1}^{k} n_i (\bar{y}_i - \bar{y})^2 \\
&\hat{=} \quad \text{SS}_{\text{WG}} + \text{SS}_{\text{BG}}.
\end{aligned} \tag{5.16}$$

- Divided both sides of (5.16) by the common variance, we obtain
$$\frac{\text{CTSS}}{\sigma^2} = \frac{\text{SS}_{\text{WG}}}{\sigma^2} + \frac{\text{SS}_{\text{BG}}}{\sigma^2}.$$

- It can be shown that
$$\frac{\text{CTSS}}{\sigma^2} \sim \chi^2(n-1), \quad \frac{\text{SS}_{\text{WG}}}{\sigma^2} \sim \chi^2(n-k), \quad \frac{\text{SS}_{\text{BG}}}{\sigma^2} \sim \chi^2(k-1), \tag{5.17}$$
and SS_{WG} and SS_{BG} are independent.

- Under H_0 or H_0^*, we expect the group means ($\bar{y}_i$) and the overall mean ($\bar{y}$) to be approximately equal (i.e., CTSS $\approx \text{SS}_{\text{WG}}$ or $\text{SS}_{\text{BG}} \approx 0$), so we expect the ratio $\text{SS}_{\text{BG}}/\text{SS}_{\text{WG}}$ to be near zero.

- We use the following F test to compare the variances:
$$F = \frac{\text{SS}_{\text{BG}}/(k-1)}{\text{SS}_{\text{WG}}/(n-k)} \sim F(k-1, n-k). \tag{5.18}$$

- It is clear that the test for testing H_0^* against H_1^* is a one-sided test.

17• Anova table

- The above discussions can be summarized in Table 5.1.

Table 5.1 *The ANOVA table*

		Sum of	Mean	F	Pr
Source	DF	Squares	Square	Value	(>F)
Between Groups (Factor)	$k-1$	$\text{SS}_{\text{BG}} =$ $\sum_{i=1}^{k} n_i(\bar{y}_i - \bar{y})^2$	$\frac{\text{SS}_{\text{BG}}}{k-1}$	$\frac{\text{SS}_{\text{BG}}/(k-1)}{\text{MSE}}$	
Within Groups (Residuals)	$n-k$	$\text{SS}_{\text{WG}} =$ $\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i)^2$	$\frac{\text{SS}_{\text{WG}}}{n-k}$ $= \text{MSE}$		
Corrected Total	$n-1$	$\text{CTSS} =$ $\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij} - \bar{y})^2$			

18• F TEST IS AN EXTENSION OF THE TWO-SAMPLE t TEST

- When $k = 2$, we have $F = t^2$, where

$$t = \frac{\bar{y}_1 - \bar{y}_2}{\sqrt{s_p^2(\frac{1}{n_1} + \frac{1}{n_2})}} \quad \text{and} \quad s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

  denotes the pooled variance, see (3.19) and (3.20).

5.2.2 Example 5.1

19• DATA AND QUESTIONS

- Suppose there are three feeds (A, B and C) for pigs.
- A sample of $n = 19$ pigs, initially with more or less the same weight, are randomly assigned to the three feeds.
- Let

Feed A:	60.8	74.0	69.8	71.6	67.5		
Feed B:	102.6	102.1	98.7	106.8	89.5	96.5	99.7
Feed C:	87.9	84.2	90.3	77.6	86.9	75.2	82.7

  denote the weights of the 19 pigs after treated with the feeds for six months.

  — Is there any difference between the effects of the three feeds?

  — If the answer is yes, which feed is the best and which feed is the worst?

20• R FUNCTION

```
function (ind, A, B, C)
{ # Function name: anova.three.group(ind, A, B, C)
  # -------------------- Aim ---------------------------------
  # Perform one-way ANOVA with three groups
  # -------------------- Input -------------------------------
  # ind = 1: Summary statistics
  # ind = 2: ANOVA table
  #       A: a vector of observations in group 1
  #       B: a vector of observations in group 2
  #       C: a vector of observations in group 3
  # -------------------- Output ------------------------------
  # Summary statistics & ANOVA table
  ############################################################
  n1 <- length(A); n2 <- length(B); n3 <- length(C)
  y1.bar <- mean(A); y2.bar <- mean(B); y3.bar <- mean(C)
  y1.css <- (n1 - 1)*var(A)
  y2.css <- (n2 - 1)*var(B)
  y3.css <- (n3 - 1)*var(C)
  k <- 3
  n <- n1 + n2 + n3
  if (ind == 1) {
    # --- MLEs of \mu_i & \sigma^2 using (5.5) & (5.6) -------
    mu1.hat <- y1.bar
    mu2.hat <- y2.bar
    mu3.hat <- y3.bar
    SSWG <- y1.css + y2.css + y3.css
    sigma.sq.hat <- SSWG/n
    # --- Unbiased estimate of \sigma^2 using (5.7) ----------
    MSE <- SSWG/(n-k)
    resultM <- matrix(c(mu1.hat, mu2.hat, mu3.hat,
                         sigma.sq.hat, MSE), nrow=5, byrow=T)
    rownames(resultM) <- c("mu1.hat", "mu2.hat", "mu3.hat",
                                       "sigma.sq.hat", "MSE")
    colnames(resultM) <- c("       MLE")
    return(resultM)
  } else
    # ------------------ F test in Table 5.1 -----------------
    y.bar <- (n1*y1.bar + n2*y2.bar + n3*y3.bar)/n
    SSBG <- n1*(y1.bar - y.bar)^2 + n2*(y2.bar - y.bar)^2
                                  + n3*(y3.bar - y.bar)^2
    SSWG <- y1.css + y2.css + y3.css
    CTSS <- SSBG + SSWG
    MSE <- SSWG/(n-k)
    F.value <- SSBG/((k-1)*MSE)
    p.value <- 1 - pf(F.value, k-1, n-k)
    resultM <- matrix(c(k-1, SSBG, SSBG/(k-1), F.value,
                        p.value, n-k, SSWG, MSE, NA, NA),
                        nrow=2, byrow=T)
    rownames(resultM) <- c("Factor", "Residuals")
    colnames(resultM) <- c("Df", " Sum Sq", " Mean Sq",
                           " F-value", "p-value")
    return(resultM)
}*************************************************************
```

20.1$^\bullet$ Comments on above R function

— Let $\mathbf{z} = (z_1, \ldots, z_m)^\top$, the corrected sum of squares is

$$\texttt{css}(\mathbf{z}) = \sum_{i=1}^{m} (z_i - \bar{z})^2 = (m-1) * \texttt{var}(\mathbf{z}).$$

20.2$^\bullet$ R output

```
==============================================================
> A <- c(60.8, 74.0, 69.8, 71.6, 67.5)
> B <- c(102.6, 102.1, 98.7, 106.8, 89.5, 96.5, 99.7)
> C <- c(87.9, 84.2, 90.3, 77.6, 86.9, 75.2, 82.7)
--------------------------------------------------------------
> anova.three.group(1, A, B, C)
                    MLE
mu1.hat        68.74000
mu2.hat        99.41429
mu3.hat        83.54286
sigma.sq.hat   24.35883
MSE            28.92611
--------------------------------------------------------------
> anova.three.group(2, A, B, C)
          Df      Sum Sq      Mean Sq    F-value        p-value
Factor     2   2786.5518   1393.27588   48.16673   1.693897e-07
Residuals 16    462.8177     28.92611         NA             NA
**************************************************************
```

20.3$^\bullet$ Conclusions from above R output

— Since the p-values < 0.05, we reject H_0; that is, there is a difference between the effects of the three feeds.

— Because $\hat{\mu}_2 > \hat{\mu}_3 > \hat{\mu}_1$, Feed B is the best and Feed A is the worst.

21• Using the built-in R function lm().

21.1• Creating data in the form of data.frame()

```
==============================================================
> feeds <- data.frame()
> fix(feeds)
> feeds
   feed weight
1     A   60.8
2     A   74.0
3     A   69.8
4     A   71.6
5     A   67.5
6     B  102.6
7     B  102.1
8     B   98.7
9     B  106.8
10    B   89.5
11    B   96.5
12    B   99.7
13    C   87.9
14    C   84.2
15    C   90.3
16    C   77.6
17    C   86.9
18    C   75.2
19    C   82.7
--------------------------------------------------------------
> attach(feeds)
> feed
 [1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "B" "B"
[13] "C" "C" "C" "C" "C" "C" "C"
> is.character(feed)
[1] TRUE
--------------------------------------------------------------
> weight
```

```
 [1]  60.8  74.0  69.8  71.6  67.5 102.6 102.1  98.7 106.8
[10]  89.5  96.5  99.7  87.9  84.2  90.3  77.6  86.9  75.2
[19]  82.7
> is.numeric(weight)
[1] TRUE
--------------------------------------------------------------
> summary(feeds)
     feed                weight
 Length:19          Min.   : 60.80
 Class :character   1st Qu.: 74.60
 Mode  :character   Median : 86.90
                    Mean   : 85.49
                    3rd Qu.: 97.60
                    Max.   :106.80
**************************************************************
```

21.2• Simple analysis of variance by lm()

```
==============================================================
> lm(weight~feed)             # To fit the linear model (5.9)

Call: lm(formula = weight ~ feed)

Coefficients:
(Intercept)          feedB           feedC
      68.74          30.67           14.80
#    mu.hat       alpha2.hat      alpha3.hat
#   mu1.hat = 68.4
#   mu2.hat = 68.74 + 30.67 = 99.41
#   mu3.hat = 68.74 + 14.80 = 83.54
--------------------------------------------------------------
> summary(lm(weight~feed))

Call: lm(formula = weight ~ feed)

Residuals:
    Min      1Q  Median      3Q     Max
-9.9143 -2.0771  0.6571  3.2714  7.3857
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   68.740      2.405   28.58 3.68e-15 ***
feedB         30.674      3.149    9.74 3.96e-08 ***
feedC         14.803      3.149    4.70 0.000241 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 5.378 on 16 degrees of freedom
Multiple R-squared:  0.8576,    Adjusted R-squared:  0.8398
F-statistic: 48.17 on 2 and 16 DF,  p-value: 1.694e-07
**************************************************************
```

22• Using the built-in R function anova()

22.1• Extracting group observations from the original data frame

```
==============================================================
> weightA <- weight[feed=="A"]
> weightB <- weight[feed=="B"]
> weightC <- weight[feed=="C"]
--------------------------------------------------------------
> weightA
[1] 60.8 74.0 69.8 71.6 67.5
> weightB
[1] 102.6 102.1  98.7 106.8  89.5  96.5  99.7
> weightC
[1] 87.9 84.2 90.3 77.6 86.9 75.2 82.7
**************************************************************
```

22.2• Box plots

— We examine box plots of weight for each group.

— The weight box plots are shown in Figure 5.1.

```
==============================================================
boxplot(weight~feed, col=c("red", "blue", "green"),
        names=c("Feed A", "Feed B", "Feed C"), ylab="Weight")
**************************************************************
```

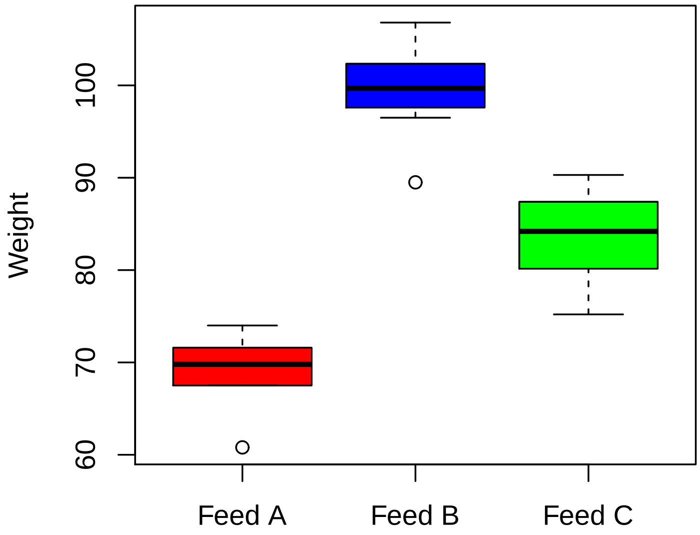

Figure 5.1 Box plots of weight by group.

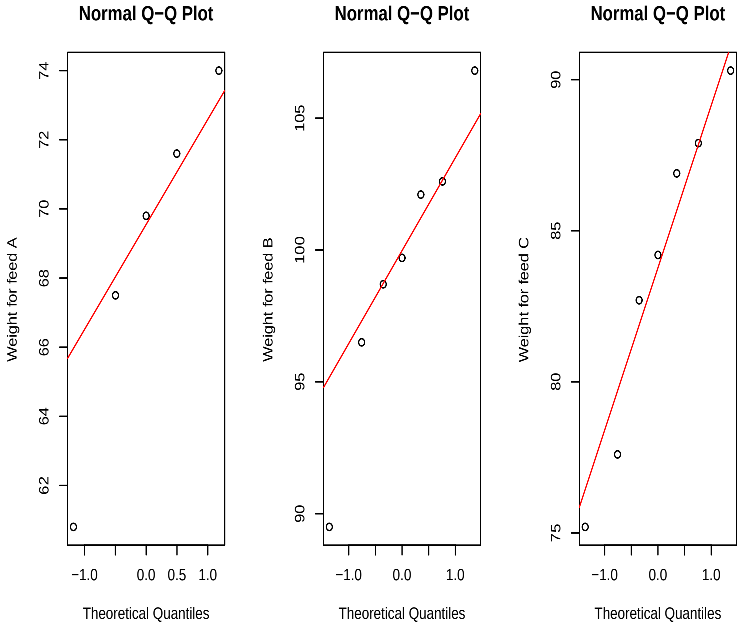

Figure 5.2 Q-Q plots of weight by group.

22.3• Q-Q plots

— To perform a one-way ANOVA, we need to test the normality assumption by drawing three Q-Q plots as shown in Figure 5.2.

```
==============================================================
> par(mfrow=c(1, 3))
> qqnorm(weightA, ylab="Weight for feed A")
> qqline(weightA, col=2)
> qqnorm(weightB, ylab="Weight for feed B")
> qqline(weightB, col=2)
> qqnorm(weightC, ylab="Weight for feed C")
> qqline(weightC, col=2)
**************************************************************
```

22.4• Shapiro–Wilk test

— To perform a one-way ANOVA, we need to test the normality assumption by performing Shapiro–Wilk test.

```
==============================================================
> shapiro.test(weightA)

        Shapiro-Wilk normality test

data:  weightA
W = 0.93856, p-value = 0.6558
--------------------------------------------------------------
> shapiro.test(weightB)

        Shapiro-Wilk normality test

data:  weightB
W = 0.95834, p-value = 0.8044
--------------------------------------------------------------
> shapiro.test(weightC)

        Shapiro-Wilk normality test
```

```
data:  weightC
W = 0.94631, p-value = 0.696
**************************************************************
```

22.5• Bartlett's test

— To perform a one-way ANOVA, we need to test the equality of three variances by performing Bartlett's test, see §5.3.1 for more details.

```
==============================================================
> bartlett.test(weight~feed)

        Bartlett test of homogeneity of variances

data:  weight by feed
Bartlett's K-squared = 0.039247, df = 2, p-value = 0.9806
**************************************************************
```

22.6• *F* test for one-way ANOVA

```
==============================================================
> anova(lm(weight~feed))
Analysis of Variance Table

Response: weight
          Df  Sum Sq Mean Sq F value    Pr(>F)
feed       2 2786.55 1393.28  48.167 1.694e-07 ***
Residuals 16  462.82   28.93
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
**************************************************************
```

5.3 Testing equality of group variances and multiple comparisons

5.3.1 Testing the equality of group variances

23• The issue

- When a one-way ANOVA is performed, it is assumed that the group variances are statistically equal.

- If this assumption is not valid, then the resulting F test is invalid.

- In general, we should consider the following model

$$Y_{i1}, \ldots, Y_{in_i} \overset{\text{iid}}{\sim} N(\mu_i, \sigma_i^2), \quad i = 1, \ldots, k.$$

- To produce the model (5.2), we first need to test

$$H_0\colon\ \sigma_1^2 = \cdots = \sigma_k^2 = \sigma^2 \tag{5.19}$$

  against

$$H_1\colon\ \text{at least two variances are different.}$$

- Equal variances across groups is called *homoscedasticity* or *homogeneity* of variances.

24• Bartlett's test

24.1• Test procedure

— Bartlett's test statistic is defined by

$$B = \frac{(n-k)\log(\text{MSE}) - \sum_{i=1}^{k}(n_i - 1)\log(s_i^2)}{1 + \frac{1}{3(k-1)}\left(\sum_{i=1}^{k}\frac{1}{n_i-1} - \frac{1}{n-k}\right)}, \tag{5.20}$$

where MSE is given by (5.7) and

$$s_i^2 = \frac{\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i)^2}{n_i - 1}, \quad i = 1, \ldots, k. \tag{5.21}$$

— Under H_0, the test statistic has approximately a $\chi^2(k-1)$ distribution.

— The p-value is $\Pr(\chi^2(k-1) > B)$.

— To test the homogeneity of variances, R uses `bartlett.test()`.

```
==============================================================
> bartlett.test(weight~feed, data= feeds)

        Bartlett test of homogeneity of variances

data:  weight by feed
Bartlett's K-squared = 0.039247, df = 2, p-value = 0.9806
**************************************************************
```

24.2• Comments

— Bartlett's test is a modification of the corresponding likelihood ratio test designed to make the approximation to the distribution better (Bartlett, M. S. (1937). Properties of sufficiency and statistical tests. *Proceedings of the Royal Statistical Society Series A* **160**, 268–282).

— Bartlett's test is *sensitive* to departure from normality. That is, if the samples come from non-normal distributions, then Bartlett's test may simply be testing for non-normality.

— The Levene test and Brown–Forsythe test are alternatives to Bartlett's test that are less sensitive to departures from normality.

25• The Levene test

- Define new variables $z_{ij} = |y_{ij} - \bar{y}_i|$ and perform Bartlett's test in (5.20) based on the $\{z_{ij}\}$ instead of the $\{y_{ij}\}$.

- The corresponding one-way ANOVA is to perform the F test in (5.18) based on the $\{z_{ij}\}$ instead of the $\{y_{ij}\}$.

26• The Brown–Forsythe test

- Define new variables $w_{ij} = |y_{ij} - \tilde{y}_i|$ and perform Bartlett's test in (5.20) based on the $\{z_{ij}\}$ instead of the $\{y_{ij}\}$, where $\tilde{y}_i$ is the median of group i.

- The corresponding one-way ANOVA is to perform the F test in (5.18) based on the $\{w_{ij}\}$ instead of the $\{y_{ij}\}$.

27• Unequal group variances

- The traditional one-way ANOVA requires an assumption of equal variance for all groups.
- However, there is an alternative procedure that does not require that assumption.
- It is due to Welch and similar to the unequal variances t test.
- This has been implemented in the built-in R function `oneway.test()`.

```
==============================================================
> oneway.test(weight~feed, data= feeds, var.equal= F)

     One-way analysis of means (not assuming equal variances)

data:  weight and feed
F = 47.467, num df = 2.000, denom df = 10.114,
p-value = 7.242e-06
--------------------------------------------------------------
> oneway.test(weight~feed, data= feeds, var.equal= T)

        One-way analysis of means

data:  weight and feed
F = 48.167, num df = 2, denom df = 16, p-value = 1.694e-07
--------------------------------------------------------------
######  which gives the same result as  ######################
--------------------------------------------------------------
> anova(lm(weight ~ feed, data = feeds))
Analysis of Variance Table

Response: weight
          Df  Sum Sq Mean Sq F value    Pr(>F)
feed       2 2786.55 1393.28  48.167 1.694e-07 ***
Residuals 16  462.82   28.93
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
**************************************************************
```

5.3.2 Multiple comparisons

28• The issue

- If we perform a one-way ANOVA and the null hypothesis (5.1) is rejected, we know that at least two group means are different.
- Actually we do not know which means are different from the others.
- In order to answer this question, we have to consider methods for multiple comparison.

29• Two-sample t test for each pairwise comparison

- The null is $H_0^{(i,j)}$: $\mu_i = \mu_j$ for a given pair (i, j) with $i \neq j$.

```
==============================================================
> t.test(weightA, weightB,  var.equal= T)

        Two Sample t-test

data:  weightA and weightB
t = -9.8853, df = 10, p-value = 1.767e-06
H_1: true difference in means is not equal to 0
95 percent confidence interval:
 -37.58828 -23.76030
sample estimates:
mean of x mean of y
 68.74000  99.41429
--------------------------------------------------------------
> t.test(weightA, weightC,  var.equal= T)

        Two Sample t-test

data:  weightA and weightC
t = -4.7478, df = 10, p-value = 0.0007829
H_1: true difference in means is not equal to 0
95 percent confidence interval:
 -21.749864  -7.855851
sample estimates:
```

```
mean of x mean of y
 68.74000  83.54286
--------------------------------------------------------------
> t.test(weightB, weightC,  var.equal= T)

        Two Sample t-test

data:  weightB and weightC
t = 5.4116, df = 12, p-value = 0.0001571
H_1: true difference in means is not equal to 0
95 percent confidence interval:
  9.481314 22.261543
sample estimates:
mean of x mean of y
 99.41429  83.54286
**************************************************************
```

30• Bonferroni’s correction

- The null hypothesis is H_0: $\mu_i = \mu_j$ for all $i, j = 1, \ldots, k$ such that $i \neq j$.

- The Bonferroni correction or ‘Bonferroni adjustment’ performs $m = k(k-1)/2$ two-sample t tests simultaneously, but the p-value for each t-test must equal to α/m.

- The Bonferroni correction is based on the fact that the probability of observing at least one of m events is less than or equal to the sum of the probabilities for each event:

$$\Pr\left(\bigcup_{i=1}^{m} \mathbb{A}_i\right) \leqslant \sum_{i=1}^{m} \Pr(\mathbb{A}_i).$$

```
==============================================================
> pairwise.t.test(weight, feed, p.adj="bonf", pool.sd= T)

        Pairwise comparisons using t tests with pooled SD

data:  weight and feed
```

```
  A       B
B 1.2e-07 -
C 0.00072 0.00014

P value adjustment method: bonferroni
**************************************************************
```

31• Other correction methods

- The Bonferroni correction (‘`bonferroni`’) is very conservative.
- Less conservative corrections are also included by Holm (1979) (‘`holm`’), Hochberg (1988) (‘`hochberg`’), Hommel (1988) (‘`hommel`’), Benjamini & Hochberg (1995) (‘`BH`’ or its alias ‘`fdr`’), and Benjamini & Yekutieli (2001) (‘`BY`’), respectively.
- A pass-through option (‘`none`’) is also included.
- The set of methods are contained in the `p.adjust.methods` vector.

```
==============================================================
> pairwise.t.test(weight, feed, p.adj="holm", pool.sd= T)

        Pairwise comparisons using t tests with pooled SD

data:  weight and feed

  A       B
B 1.2e-07 -
C 0.00024 9.3e-05

P value adjustment method: holm
--------------------------------------------------------------
> pairwise.t.test(weight, feed)  # The default is  "holm"

        Pairwise comparisons using t tests with pooled SD

data:  weight and feed
```

```
  A       B
B 1.2e-07 -
C 0.00024 9.3e-05

P value adjustment method: holm
--------------------------------------------------------------
p.adjust.methods
# c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY",
#   "fdr", "none")
**************************************************************
```

5.4 The Kruskal–Wallis rank sum test

32• Background

- When the normality assumption is violated, we could use the Kruskal–Wallis rank sum test, which is a non-parametric alternative to the one-way ANOVA.

- It is identical to a one-way ANOVA with the data replaced by their ranks.

- It is an extension of the two-sample Wilcoxon test (see §3.3.4) to three or more groups.

33• The Kruskal–Wallis rank sum test method

- Rank all data from all groups together; i.e., rank the data from 1 to n ignoring group membership.

- Assign any tied values the average of the ranks they would have received.

- The test statistic is given by

$$\text{KW} = \frac{(n-1)\sum_{i=1}^{k} n_i(\bar{r}_i - \bar{r})^2}{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(r_{ij} - \bar{r})^2},$$

  where n_i is the number of observations in group i, r_{ij} is the rank (among all observations) of observation j from group i, $n = \sum_{i=1}^{k} n_i$

is the total number of observations across all groups, $\bar{r}_i = \sum_{j=1}^{n_i} r_{ij}/n_i$ and $\bar{r} = (n+1)/2$ is the average of all the r_{ij}.

- The null hypothesis of equal population medians would then be rejected if $\text{KW} \geqslant \chi^2(\alpha, k-1)$.

34• An illustration

```
================================================================
> weightA
[1] 60.8 74.0 69.8 71.6 67.5
> weightB
[1] 102.6 102.1  98.7 106.8  89.5  96.5  99.7
> weightC
[1] 87.9 84.2 90.3 77.6 86.9 75.2 82.7
----------------------------------------------------------------
> kruskal.test(list(weightA, weightB, weightC))

        Kruskal-Wallis rank sum test

data:  list(weightA, weightB, weightC)
Kruskal-Wallis chi-squared = 15.483, df = 2,
p-value = 0.0004345
****************************************************************
```

5.5 Two real examples

5.5.1 Example 5.2

35• Data and questions

- Amphetamine is a drug that suppresses appetite.
- In a study of this effect, a pharmacologist randomly allocated 24 rats to three treatment groups to receive an injection of amphetamine at one of two dosage levels, or an injection of saline solution.
- He measured the amount of food consumed by each animal in the 3-hour period following injection.

- The results (gram of food consumed per kg body weight, Y) are shown as follows.

Dose = 0:	112.7	102.1	90.2	81.5	105.6	93.0	106.6	108.3
Dose = 2.5:	73.3	84.8	67.3	55.3	80.7	90.0	75.5	77.1
Dose = 5.0:	38.6	81.3	57.1	62.3	51.6	48.3	42.8	58.0

(1) Construct three boxplots for the amounts of food consumed per kg body weight by the animals taking the drug at three dosage levels.

(2) Do the animals have significant different amounts of food consumed per kg body weight if they take drug at different levels of dosage? Is the assumption of equality of variance valid?

(3) Let X = dose of amphetamine (mg/kg). Since X is actually continuous, a regression of Y on X could be considered. Fit a simple linear regression and a quadratic regression of Y on X.

36• Using boxplot() for question (1)

```
==============================================================
> food.rats <- data.frame()
> fix(food.rats)
> food.rats
   dose   x     y
1    L1 0.0 112.7
2    L1 0.0 102.1
3    L1 0.0  90.2
4    L1 0.0  81.5
5    L1 0.0 105.6
6    L1 0.0  93.0
7    L1 0.0 106.6
8    L1 0.0 108.3
9    L2 2.5  73.3
10   L2 2.5  84.8
11   L2 2.5  67.3
12   L2 2.5  55.3
13   L2 2.5  80.7
```

```
14   L2 2.5  90.0
15   L2 2.5  75.5
16   L2 2.5  77.1
17   L3 5.0  38.6
18   L3 5.0  81.3
19   L3 5.0  57.1
20   L3 5.0  62.3
21   L3 5.0  51.6
22   L3 5.0  48.3
23   L3 5.0  42.8
24   L3 5.0  58.0
> attach(food.rats)
> dose
 [1] "L1" "L1" "L1" "L1" "L1" "L1" "L1" "L1"
 [9] "L2" "L2" "L2" "L2" "L2" "L2" "L2" "L2"
[19] "L3" "L3" "L3" "L3" "L3" "L3" "L3" "L3"
> x
 [1] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 [9] 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5
[19] 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0
> y
 [1] 112.7 102.1  90.2  81.5 105.6  93.0 106.6 108.3
 [9]  73.3  84.8  67.3  55.3  80.7  90.0  75.5  77.1
[17]  38.6  81.3  57.1  62.3  51.6  48.3  42.8  58.0
--------------------------------------------------------------
> boxplot(y~dose, col=c("red", "blue", "green"),
          names=c("Dose 0", "Dose 2.5", "Dose 5.0"),
          ylab="food (g/kg)")                ### Figure 5.3
**************************************************************
```

37• Using anova() for question (2)

```
==============================================================
> yL1 <- y[dose=="L1"]
> yL2 <- y[dose=="L2"]
> yL3 <- y[dose=="L3"]
> yL1
[1] 112.7 102.1  90.2  81.5 105.6  93.0 106.6 108.3
```

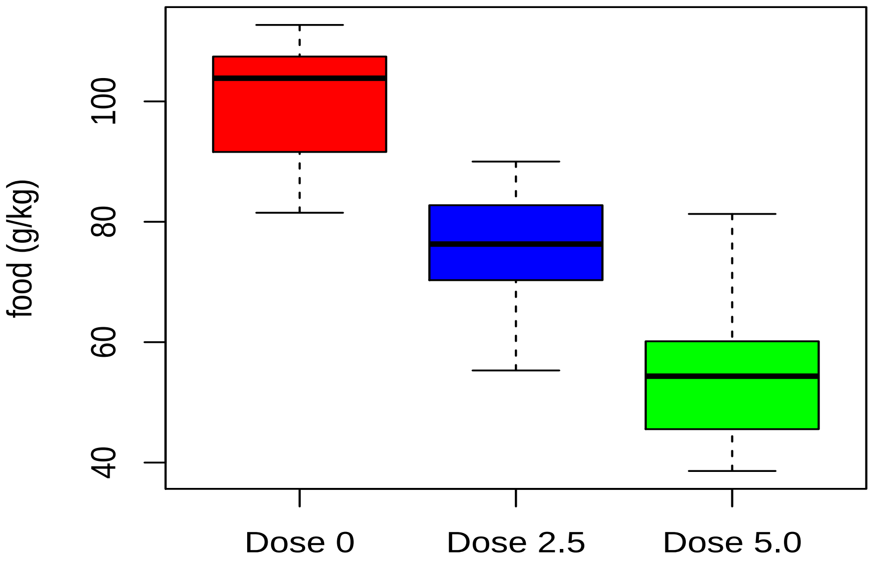

Figure 5.3 Box plots of y by group.

```
> yL2
[1] 73.3 84.8 67.3 55.3 80.7 90.0 75.5 77.1
> yL3
[1] 38.6 81.3 57.1 62.3 51.6 48.3 42.8 58.0
--------------------------------------------------------------
> shapiro.test(yL1)

        Shapiro-Wilk normality test

data:  yL1
W = 0.9273, p-value = 0.4918
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(yL2)

        Shapiro-Wilk normality test

data:  yL2
W = 0.9664, p-value = 0.8684
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(yL3)

        Shapiro-Wilk normality test

data:  yL3
W = 0.93855, p-value = 0.5969
--------------------------------------------------------------
> bartlett.test(y˜dose)

        Bartlett test of homogeneity of variances

data:  y by dose
Bartlett's K-squared = 0.42346, df = 2, p-value = 0.8092
--------------------------------------------------------------
> anova(lm(y˜dose))
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
dose       2 8121.3  4060.7   30.07 6.861e-07 ***
Residuals 21 2835.9   135.0
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> pairwise.t.test(y, dose, p.adj="bonf", pool.sd= T)

        Pairwise comparisons using t tests with pooled SD

data:  y and dose

   L1      L2
L2 0.0012  -
L3 4.1e-07 0.0060

P value adjustment method: bonferroni
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> kruskal.test(list(yL1, yL2, yL3))
```

```
        Kruskal-Wallis rank sum test

data:  list(yL1, yL2, yL3)
Kruskal-Wallis chi-squared = 17.295, df = 2,
p-value = 0.0001756
--------------------------------------------------------------
> lm(y~dose)

Call:
lm(formula = y ~ dose)

Coefficients:
(Intercept)       doseL2       doseL3
      100.0        -24.5        -45.0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> summary(lm(y~dose))

Call:
lm(formula = y ~ dose)

Residuals:
    Min      1Q  Median      3Q     Max
-20.200  -7.300   1.850   6.775  26.300

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  100.000      4.109  24.339  < 2e-16 ***
doseL2       -24.500      5.810  -4.217 0.000387 ***
doseL3       -45.000      5.810  -7.745 1.38e-07 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 11.62 on 21 degrees of freedom
Multiple R-squared:  0.7412,    Adjusted R-squared:  0.7165
F-statistic: 30.07 on 2 and 21 DF,  p-value: 6.861e-07
**************************************************************
```

38• Using lm() for question (3)

```
==============================================================
> lm(y~x)                            # Model y = a + b x + e

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      99.33        -9.00            # Model y = 99.33 - 9.0 x
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> summary(lm(y~x))

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-21.533  -7.033   1.517   7.442  26.967

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   99.333      3.678  27.007  < 2e-16 ***
x             -9.000      1.140  -7.897 7.31e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 11.4 on 22 degrees of freedom
Multiple R-squared:  0.7392,    Adjusted R-squared:  0.7274
F-statistic: 62.37 on 1 and 22 DF,  p-value: 7.31e-08
--------------------------------------------------------------
> lm(y~x + x^2)                           # incorrect fitting

Call:
lm(formula = y ~ x + x^2)

Coefficients:
```

```
(Intercept)            x
      99.33        -9.00
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> x.square <- x^2
> x
 [1] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 [9] 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5
[19] 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0
> x.square
 [1]  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 [9]  6.25  6.25  6.25  6.25  6.25  6.25  6.25  6.25
[19] 25.00 25.00 25.00 25.00 25.00 25.00 25.00 25.00
--------------------------------------------------------------
> lm(y~x + x.square)          # Model y = a + b x + c x*x + e

Call:
lm(formula = y ~ x + x.square)

Coefficients:
(Intercept)        x x.square
     100.00   -10.60     0.32  # Model y=100-10.6 x + 0.32 x*x
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> summary(lm(y~x + x.square))

Call:
lm(formula = y ~ x + x.square)

Residuals:
    Min      1Q  Median      3Q     Max
-20.200  -7.300   1.850   6.775  26.300

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 100.0000     4.1086  24.339   <2e-16 ***
x           -10.6000     4.1899  -2.530   0.0195 *
x.square      0.3200     0.8051   0.397   0.6950
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
Residual standard error: 11.62 on 21 degrees of freedom
Multiple R-squared:  0.7412,    Adjusted R-squared:  0.7165
F-statistic: 30.07 on 2 and 21 DF,  p-value: 6.861e-07
--------------------------------------------------------------
> summary(lm(y~x.square))     # Model y = a + c x*x + e
                              # Model y = 93.8846 - 1.6369 x*x
Call:
lm(formula = y ~ x.square)

Residuals:
    Min      1Q  Median      3Q     Max
-28.354  -8.656  -1.123   8.496  28.338

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  93.8846     3.7076  25.322  < 2e-16 ***
x.square     -1.6369     0.2492  -6.569 1.32e-06 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 12.97 on 22 degrees of freedom
Multiple R-squared:  0.6623,    Adjusted R-squared:  0.647
F-statistic: 43.15 on 1 and 22 DF,  p-value: 1.319e-06
**************************************************************
```

5.5.2 Example 5.3

39• Data and questions

- Patients with advanced cancer of the stomach, bronchus, colon, ovary or breast were treated with ascorbate and their survival times are recorded as follows.

Stomach: 124 42 25 45 412 51 1112 46 103 876
146 340 396
Bronchus: 81 461 20 450 246 166 63 64 155 859
151 166 37 223 138 72 245

Colon: 248 377 189 1843 180 537 519 455 406 365
942 776 372 163 101 20 283

Ovary: 1234 89 201 356 2970 456

Breast: 1235 24 1581 1166 40 727 3808 791 1804 3460
719

(1) Construct five box plots for the survival times of patients with advanced cancer.

(2) Do the survival times differ with the organ affected?

(3) If the normality assumption is violated, we make a logarithm transformation on the survival times, and perform the above analyses again.

40• Using boxplot() for question (1)

```
==============================================================
> cancer.st <- data.frame()
> fix(cancer.st)
## input cancer <- rep("a", 64)
##   input time <- rep(0,   64)
--------------------------------------------------------------
> cancer.st$cancer[1:13] <- rep("stomach", 13)
> cancer.st$cancer[14:30] <- rep("bronchus", 17)
> cancer.st$cancer[31:47] <- rep("colon", 17)
> cancer.st$cancer[48:53] <- rep("ovary", 6)
> cancer.st$cancer[54:64] <- rep("breast", 11)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> cancer.st$time[1:13] <- c(124, 42, 25, 45, 412, 51, 1112,
                             46, 103, 876, 146, 340, 396)
> cancer.st$time[14:30] <- c(81, 461,20, 450, 246, 166, 63,
                64, 155, 859, 151, 166,37, 223, 138, 72, 245)
> cancer.st$time[31:47] <- c(248,377,189, 1843, 180, 537, 519,
              455, 406, 365, 942, 776, 372, 163, 101, 20, 283)
> cancer.st$time[48:53] <- c(1234,  89,  201, 356, 2970, 456)
> cancer.st$time[54:64] <- c(1235, 24, 1581, 1166, 40, 727,
                             3808, 791, 1804, 3460, 719)
> cancer.st$logtime <- log(cancer.st$time)
```

```
---------------------------------------------------------------
> cancer.st
     cancer time  logtime
1   stomach  124 4.820282
2   stomach   42 3.737670
3   stomach   25 3.218876
4   stomach   45 3.806662
5   stomach  412 6.021023
6   stomach   51 3.931826
7   stomach 1112 7.013915
8   stomach   46 3.828641
9   stomach  103 4.634729
10  stomach  876 6.775366
11  stomach  146 4.983607
12  stomach  340 5.828946
13  stomach  396 5.981414
14 bronchus   81 4.394449
15 bronchus  461 6.133398
16 bronchus   20 2.995732
17 bronchus  450 6.109248
18 bronchus  246 5.505332
19 bronchus  166 5.111988
20 bronchus   63 4.143135
21 bronchus   64 4.158883
22 bronchus  155 5.043425
23 bronchus  859 6.755769
24 bronchus  151 5.017280
25 bronchus  166 5.111988
26 bronchus   37 3.610918
27 bronchus  223 5.407172
28 bronchus  138 4.927254
29 bronchus   72 4.276666
30 bronchus  245 5.501258
31    colon  248 5.513429
32    colon  377 5.932245
33    colon  189 5.241747
34    colon 1843 7.519150
35    colon  180 5.192957
```

```
36    colon  537 6.285998
37    colon  519 6.251904
38    colon  455 6.120297
39    colon  406 6.006353
40    colon  365 5.899897
41    colon  942 6.848005
42    colon  776 6.654153
43    colon  372 5.918894
44    colon  163 5.093750
45    colon  101 4.615121
46    colon   20 2.995732
47    colon  283 5.645447
48    ovary 1234 7.118016
49    ovary   89 4.488636
50    ovary  201 5.303305
51    ovary  356 5.874931
52    ovary 2970 7.996317
53    ovary  456 6.122493
54   breast 1235 7.118826
55   breast   24 3.178054
56   breast 1581 7.365813
57   breast 1166 7.061334
58   breast   40 3.688879
59   breast  727 6.588926
60   breast 3808 8.244859
61   breast  791 6.673298
62   breast 1804 7.497762
63   breast 3460 8.149024
64   breast  719 6.577861
--------------------------------------------------------------
> attach(cancer.st)
> boxplot(time~cancer, col=c("red", "blue", "green", "yellow",
  "grey"), names=c("Stomach", "Bronchus", "Colon", "Ovary",
  "Breast"), ylab="Survival time")
**************************************************************
```

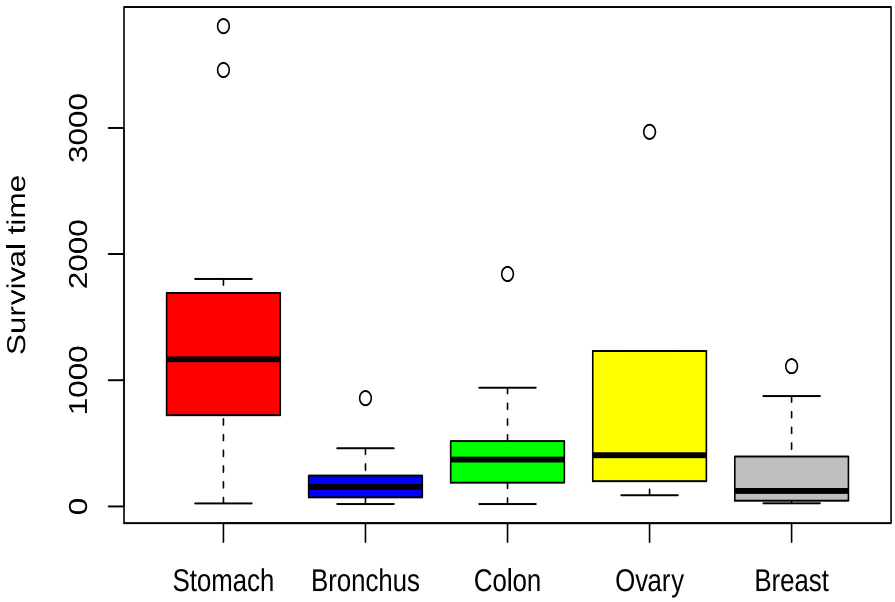

Figure 5.4 Box plots of survival time by group.

41• Using anova() for question (2)

```
==============================================================
> time.stomach <- time[cancer=="stomach"]
> time.bronchus <- time[cancer=="bronchus"]
> time.colon <- time[cancer=="colon"]
> time.ovary <- time[cancer=="ovary"]
> time.breast <- time[cancer=="breast"]
--------------------------------------------------------------
> shapiro.test(time.stomach)

        Shapiro-Wilk normality test

data:  time.stomach
W = 0.75473, p-value = 0.002075
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(time.bronchus)
```

```
        Shapiro-Wilk normality test

data:  time.bronchus
W = 0.76596, p-value = 0.0007186
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(time.colon)

        Shapiro-Wilk normality test

data:  time.colon
W = 0.76056, p-value = 0.0006134
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(time.ovary)

        Shapiro-Wilk normality test

data:  time.ovary
W = 0.76688, p-value = 0.029
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(time.breast)

        Shapiro-Wilk normality test

data:  time.breast
W = 0.86857, p-value = 0.07431
--------------------------------------------------------------
>  bartlett.test(time~cancer)

        Bartlett test of homogeneity of variances

data:  time by cancer
Bartlett's K-squared = 48.097, df = 4, p-value = 9.009e-10
--------------------------------------------------------------
> anova(lm(time~cancer))
Analysis of Variance Table

Response: time
          Df   Sum Sq Mean Sq F value    Pr(>F)
```

```
cancer     4 11535761 2883940  6.4334 0.0002295 ***
Residuals 59 26448144  448274
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> oneway.test(time˜cancer, data= cancer.st, var.equal= F)

     One-way analysis of means (not assuming equal variances)

data:  time and cancer
F = 3.5152, num df = 4.000, denom df = 19.862,
p-value = 0.02514
--------------------------------------------------------------
> summary(lm(time˜cancer))

Call:
lm(formula = time ˜ cancer)

Residuals:
     Min       1Q   Median       3Q      Max
-1371.91  -241.75  -111.50    87.19  2412.09

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      1395.9      201.9   6.915 3.77e-09 ***
cancerbronchus  -1184.3      259.1  -4.571 2.53e-05 ***
cancercolon      -938.5      259.1  -3.622 0.000608 ***
cancerovary      -511.6      339.8  -1.506 0.137526
cancerstomach   -1109.9      274.3  -4.046 0.000153 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 669.5 on 59 degrees of freedom
Multiple R-squared:  0.3037,    Adjusted R-squared:  0.2565
F-statistic: 6.433 on 4 and 59 DF,  p-value: 0.0002295
**************************************************************
```

42• Using anova() for question (3)

```
==============================================================
> logtime.stomach <- logtime[cancer=="stomach"]
> logtime.bronchus <- logtime[cancer=="bronchus"]
> logtime.colon <- logtime[cancer=="colon"]
> logtime.ovary <- logtime[cancer=="ovary"]
> logtime.breast <- logtime[cancer=="breast"]
--------------------------------------------------------------
> shapiro.test(logtime.stomach)

        Shapiro-Wilk normality test

data:  logtime.stomach
W = 0.92837, p-value = 0.3245
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(logtime.bronchus)

        Shapiro-Wilk normality test

data:  logtime.bronchus
W = 0.98047, p-value = 0.9613
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(logtime.colon)

        Shapiro-Wilk normality test

data:  logtime.colon
W = 0.92636, p-value = 0.1891
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(logtime.ovary)

        Shapiro-Wilk normality test

data:  logtime.ovary
W = 0.983, p-value = 0.9655
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> shapiro.test(logtime.breast)
```

```
        Shapiro-Wilk normality test

data:  logtime.breast
W = 0.802, p-value = 0.009995
--------------------------------------------------------------
> bartlett.test(logtime˜cancer)

        Bartlett test of homogeneity of variances

data:  logtime by cancer
Bartlett's K-squared = 4.809, df = 4, p-value = 0.3075
--------------------------------------------------------------
>  anova(lm(logtime˜cancer))
Analysis of Variance Table

Response: logtime
          Df Sum Sq Mean Sq F value   Pr(>F)
cancer     4 24.487  6.1216   4.286 0.004122 **
Residuals 59 84.270  1.4283
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> kruskal.test(list(logtime.stomach, logtime.bronchus,
               logtime.colon, logtime.ovary, logtime.breast))

        Kruskal-Wallis rank sum test

data:  list(logtime.stomach, logtime.bronchus, logtime.colon,
            logtime.ovary, logtime.breast)
Kruskal-Wallis chi-squared = 14.954, df = 4,
p-value = 0.004798
--------------------------------------------------------------
> summary(lm(logtime˜cancer))

Call:
lm(formula = logtime ˜ cancer)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.3805 -0.6607  0.1025  0.8207  2.0460

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      6.5586     0.3603  18.201  < 2e-16 ***
cancerbronchus  -1.6054     0.4625  -3.472 0.000975 ***
cancercolon     -0.8095     0.4625  -1.750 0.085247 .
cancerovary     -0.4080     0.6065  -0.673 0.503801
cancerstomach   -1.5907     0.4896  -3.249 0.001915 **
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.195 on 59 degrees of freedom
Multiple R-squared:  0.2252,    Adjusted R-squared:  0.1726
F-statistic: 4.286 on 4 and 59 DF,  p-value: 0.004122
**************************************************************
```

5.6 Two-way analysis of variance

5.6.1 Two-way ANOVA without interaction effect

43• Randomized block design

- A randomized block design (containing I treatments and J blocks) consists of I experimental units in each of the J blocks.

- The treatments are randomly assigned to the units in each block, with each treatment appearing exactly once in every block, i.e., a single observation per cell.

44• The model

- Suppose that we have I levels of *treatments* and J levels of *blocks*.

- Let the response be Y_{ij}, the measurement of the i-th treatment and the j-th block and the model can be written as

$$Y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}, \quad i = 1, \ldots, I; \quad j = 1, \ldots, J, \tag{5.22}$$

where μ is the intercept.

- We now have a total of $I \times J$ samples, each of size one.

45• Assumptions in the model (5.22)

- Each Y_{ij} observed constitutes a random independent sample of size one from one of $I \times J$ populations represented.

- Each of these $I \times J$ populations is normally distributed with mean $\mu_{ij} = \mu + \alpha_i + \beta_j$ and the same variance σ^2. This implies that the $\varepsilon_{ij} \overset{\text{iid}}{\sim} N(0, \sigma^2)$.

- The treatment (or row) effects (i.e., α_i) and the block (or column) effects (i.e., β_j) are additive. This assumption may be interpreted as no interaction between treatments and blocks.

- In other words, a particular block–treatment combination does not produce an effect that is greater or less than the sum of their individual effects.

46• Constraints on parameters

- The parameters in model (5.22) are not uniquely defined unless we impose some restrictions on these parameters.

- For example, if we impose

$$\sum_{i=1}^{I} \alpha_i = \sum_{j=1}^{J} \beta_j = 0, \tag{5.23}$$

  then μ is the overall mean. The restrictions in (5.23) are known as the *sum to zero constraints.*

- If we impose

$$\alpha_I = \beta_J = 0, \tag{5.24}$$

  then μ is just an intercept/constant. The restrictions in (5.24) are known as the *set to zero constraints* as adopted by SAS.

47• Anova table

- As in the one-way ANOVA, the total sum of squares may be partitioned into three parts, the sums of squares for blocks, treatments, and error.

- For a randomized block design with I treatments and J blocks, the analysis of variance can be summarized in the following ANOVA table.

Table 5.2 *The two-way ANOVA table without interaction*

Source of Variation	DF	Sum of Squares	Mean Square	F Value	Prob $> F$
Treatments (Factor A)	$I - 1$	SSA	MSA	MSA/MSE	
Blocks (Factor B)	$J - 1$	SSB	MSB	MSB/MSE	
Error	$IJ - I - J + 1$	SSE	MSE		
Corrected Total	$IJ - 1$	CTSS			

48• Test on no treatment effects

- To test the null hypothesis that there is no difference in treatment means, i.e.,

$$H_0\colon\ \alpha_i = 0, \quad i = 1, \ldots, I,$$

we use the F test

$$F = \frac{\text{MSA}}{\text{MSE}} \sim F(I-1,\ IJ-I-J+1).$$

49• Test on no block effects

- To test the null hypothesis that there is no difference in block means, i.e.,

$$H_0\colon\ \beta_j = 0, \quad j = 1, \ldots, J,$$

we use the F test

$$F = \frac{\text{MSB}}{\text{MSE}} \sim F(J-1,\ IJ-I-J+1).$$

5.6.2 Example 5.4

50• Data and questions

-

5.6.3 Two-way ANOVA with interaction effect

51• More general data structure

- We consider more general cases with two categorical variables (or factors): row variable with I levels and column variable with J levels.
- For the cell (i, j), we have observed n_{ij} independent replicates:

$$Y_{ij1}, Y_{ij2}, \ldots, Y_{ijn_{ij}}.$$

- The aim of the two-way ANOVA is to study the effects of the row and column on the response variable Y.

52• General two-way anova model

- Consider the following model:

$$Y_{ijk} = \mu_{ij} + \varepsilon_{ijk}, \tag{5.25}$$

for $i = 1, \ldots, I$; $j = 1, \ldots, J$; $k = 1, \ldots, n_{ij}$, where $\varepsilon_{ijk} \overset{\text{iid}}{\sim} N(0, \sigma^2)$.

- The general two-way ANOVA model can be shown by Table 5.3.

Table 5.3 *General two-way ANOVA model*

	Column				
Row	1	$\cdots$	j	$\cdots$	J
1	μ_{11}	$\cdots$	μ_{1j}	$\cdots$	μ_{1J}
$\vdots$	$\vdots$		$\vdots$		$\vdots$
i	μ_{i1}	$\cdots$	μ_{ij}	$\cdots$	μ_{iJ}
$\vdots$	$\vdots$		$\vdots$		$\vdots$
I	μ_{I1}	$\cdots$	μ_{Ij}	$\cdots$	μ_{IJ}

53• Assumptions in the model (5.25)

- The observations in each of the $I \times J$ cells constitute a random, independent sample of size n_{ij} drawn from the population defined by the particular combinations of the levels of the two factors.
- Each of the $I \times J$ populations can be modeled by a normal distribution.
- The populations all have the same variance.

54• Alternative forms of the two-way ANOVA model

- Consider the model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \delta_{ij} + \varepsilon_{ijk}, \tag{5.26}$$

for $i = 1, \ldots, I$; $j = 1, \ldots, J$; $k = 1, \ldots, n_{ij}$.

- Similar to (5.23), we impose

$$\sum_{i=1}^{I} \alpha_i = \sum_{j=1}^{J} \beta_j = 0 \quad \text{and} \quad \sum_{i=1}^{I} \delta_{ij} = \sum_{j=1}^{J} \delta_{ij} = 0.$$

- Similar to (5.24), the adopted restrictions in SAS are $\alpha_I = 0$, $\beta_J = 0$, $\delta_{Ij} = 0$ for all j and $\delta_{iJ} = 0$ for all i.

55• Testing for no row/column/interaction effects

- Based on the model (5.26), we consider the following hypotheses

$$H_0^1\colon\ \alpha_1 = \cdots = \alpha_I = 0 \quad \text{(no row effect)},$$

$$H_0^2\colon\ \beta_1 = \cdots = \beta_J = 0 \quad \text{(no column effect)},$$

and

$$H_0^3\colon\ (\alpha\beta)_{ij} = 0 \quad \text{(no interaction effect)},$$

by constructing the ANOVA table as shown in Table 5.4, where $N = \sum_{i=1}^{I} \sum_{j=1}^{J} n_{ij}$.

Table 5.4 *The two-way ANOVA table with interaction*

Source of Variation	DF	Sum of Squares	Mean Square	F Value	Prob $> F$
Factor A	$I-1$	SSA	MSA	MSA/MSE	
Factor B	$J-1$	SSB	MSB	MSB/MSE	
Interaction	$(I-1)(J-1)$	SSAB	MSAB	MSAB/MSE	
Error	$N-IJ$	SSE	MSE		
Corrected Total	$N-1$	CTSS			

56• Remarks

- Row and column effects are called main effects.
- The tests are valid if there are no empty cells, i.e., all cells have at least one observations.
- If $n_{ij} = 1$ for all i and j, then we can only fit the two-way ANOVA model without interaction.
- In general if the interaction is significant, the main effect components are usually included for the ease of interpretation.
- If a significant result for the interaction is detected, multiple comparisons among levels of one factor at each level of the other factor are suggested.

5.7 The Friedman rank sum test

There is a non-parametric form of two-way ANOVA.

APPENDIX A

Basic Statistical Distributions

A.1 Discrete distributions

A.1.1 Finite discrete distribution

Notation: $X \sim \text{FDiscrete}_n(\boldsymbol{x}, \boldsymbol{p})$, $\boldsymbol{x} = (x_1, \dots, x_n)^{\top}$, $\boldsymbol{p} = (p_1, \dots, p_n)^{\top} \in \mathbb{T}_n \,\hat{=}\, \{(p_1, \dots, p_n)\colon\ p_i \geqslant 0, \sum_{i=1}^n p_i = 1\}$.

Density: $\Pr(X = x_i) = p_i, \quad i = 1, \dots, n.$

Moments: $E(X) = \sum_{i=1}^n x_i p_i$, $\text{Var}(X) = \sum_{i=1}^n x_i^2 p_i - (\sum_{i=1}^n x_i p_i)^2$.

Note: The *uniform discrete* distribution is a special case of the finite discrete distribution with $p_i = 1/n$ for all i.

Sampling: `sample(x, size, replace = FALSE, prob = NULL)` takes a sample of the specified size from the elements of `x` using either with or without replacement.

Examples:

```
> sample(c(0,1), 100, replace= T, prob=c(0.8, 0.2))
> sample(1:20, 4)    # the default: replace= F
```

A.1.2 Hypergeometric distribution

Notation: $X \sim \text{Hgeometric}(m, n, k)$, m, n, k are positive integers.

Density: $\text{Hgeometric}(x|m, n, k) = \binom{m}{x}\binom{n}{k-x}/\binom{m+n}{k}$,
where $x = \max(0, k - n), \dots, \min(m, k)$.

Moments: $E(X) = km/N'$, $\text{Var}(X) = kmn(N' - k)/[N'^2(N' - 1)]$,
where $N' \,\hat{=}\, m + n$.

Computing:
`> prod(5:1)` $= 5!$
`> prod(20:16)` $= 20 \times 19 \times 18 \times 17 \times 16$
`> choose(40,5)` $= \binom{40}{5}$

Functions:
```
dhyper(x, m, n, k)
phyper(q, m, n, k)
qhyper(p, m, n, k)
rhyper(nn, m, n, k)
```

A.1.3 Poisson distribution

Notation: $X \sim \text{Poisson}(\lambda)$, $\lambda > 0$

Density: $\text{Poisson}(x|\lambda) = \lambda^x \, \mathrm{e}^{-\lambda}/x!$, $x = 0, 1, 2, \ldots$

Moments: $E(X) = \lambda$, $\text{Var}(X) = \lambda$.

Properties:
- If $\{X_i\}_{i=1}^n \overset{\text{ind}}{\sim} \text{Poisson}(\lambda_i)$, then

$$\begin{aligned} \textstyle\sum_{i=1}^n X_i &\sim \text{Poisson}(\textstyle\sum_{i=1}^n \lambda_i), \quad \text{and} \\ (X_1, \ldots, X_n)|(\textstyle\sum_{i=1}^n X_i = m) &\sim \text{Multinomial}_n(m, \boldsymbol{p}), \end{aligned}$$

  where $\boldsymbol{p} = (\lambda_1, \ldots, \lambda_n)^\top / \sum_{i=1}^n \lambda_i$;

- The Poisson and gamma distribution have relationship:

$$\textstyle\sum_{x=k}^\infty \text{Poisson}(x|\lambda) = \int_0^\lambda \text{Gamma}(y|k, 1)\, \mathrm{d}y.$$

Functions:
```
dpois(x, lambda)
ppois(q, lambda)
qpois(p, lambda)
rpois(n, lambda)
```

```
=================================================================
> x <- 0:20
> plot(x, dpois(x, 4), type="h")            # histogram-like
                                                # Figure A.1
*****************************************************************
```

A.1.4 Binomial distribution

Notation: $X \sim \text{Binomial}(n, p)$, n is a positive integer, $p \in (0, 1)$.

Density: $\text{Binomial}(x|n, p) = \binom{n}{x} p^x (1-p)^{n-x}$, $x = 0, 1, \ldots, n$.

Moments: $E(X) = np$, $\text{Var}(X) = np(1-p)$.

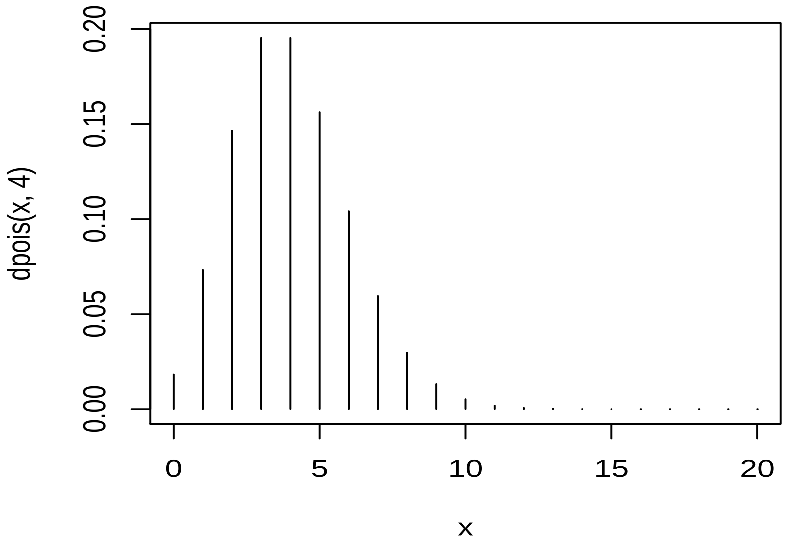

Figure A.1 Point probabilities of Poisson(4).

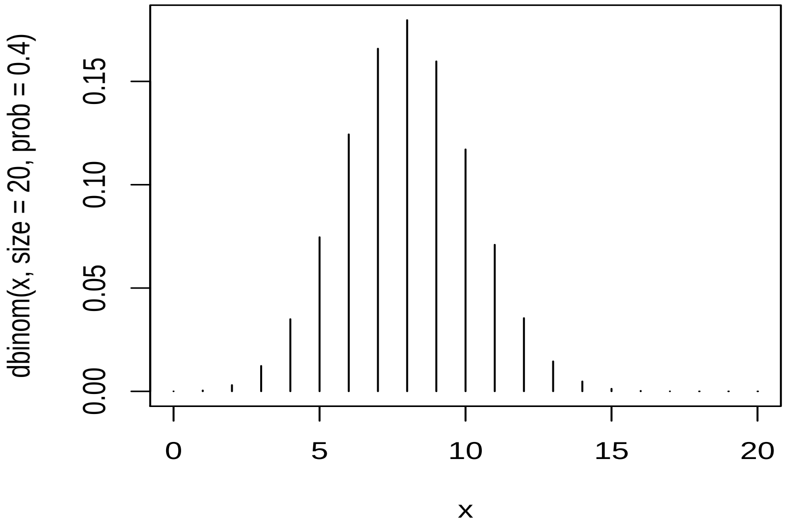

Figure A.2 Point probabilities of Binomial(20, 0.4).

Properties: • If $\{X_i\}_{i=1}^d \overset{\text{ind}}{\sim} \text{Binomial}(n_i, p)$, then

$$\textstyle\sum_{i=1}^d X_i \sim \text{Binomial}(\sum_{i=1}^d n_i, p);$$

• The binomial and beta distribution have relationship:

$$\textstyle\sum_{x=0}^k \text{Binomial}(x|n,p) = \int_0^{1-p} \text{Beta}(x|n-k, k+1)\,\mathrm{d}x,$$

where $0 \leqslant k \leqslant n$.

Note: When $n = 1$, binomial distribution is called *Bernoulli* distribution.

Functions:

```
dbinom(x, size, prob)     # size= n,  prob= p
pbinom(q, size, prob)
qbinom(p, size, prob)
rbinom(nn, size, prob)
```

```
==============================================================
> x <- 0:20
> plot(x, dbinom(x, size=20, prob=0.4), type="h")
                                               # Figure A.2
**************************************************************
```

A.1.5 Multinomial distribution

Notation: $\mathbf{x} = (X_1, \ldots, X_d)^\top \sim \text{Multinomial}(n; p_1, \ldots, p_d)$ or
$\mathbf{x} = (X_1, \ldots, X_d)^\top \sim \text{Multinomial}_d(n, \boldsymbol{p})$,
n is a positive integer, $\boldsymbol{p} = (p_1, \ldots, p_d)^\top \in \mathbb{T}_d$,

Density: $\text{Multinomial}_d(\boldsymbol{x}|n, \boldsymbol{p}) = \binom{n}{x_1, \ldots, x_d} \prod_{i=1}^d p_i^{x_i}$,
$\boldsymbol{x} = (x_1, \ldots, x_d)^\top$, $x_i \geqslant 0$, $\sum_{i=1}^d x_i = n$.

Moments: $E(X_i) = np_i$, $\text{Var}(X_i) = np_i(1-p_i)$, $\text{Cov}(X_i, X_j) = -np_ip_j$.

Note: The binomial distribution is a special case of the multinomial with $d = 2$.

Functions:

```
dmultinom(x, size = NULL, prob) # size= n, prob= p
rmultinom(nn, size, prob)
```

A.2 Continuous distributions

A.2.1 Uniform distribution

Notation: $X \sim U(a, b)$, $a < b$

Density: $U(x|a, b) = 1/(b - a)$, $x \in (a, b)$.

Moments: $E(X) = (a + b)/2$, $\mathrm{Var}(X) = (b - a)^2/12$.

Properties: If $Y \sim U(0, 1)$, then $X = a + (b - a)Y \sim U(a, b)$.

Functions:

```
dunif(x, min= 0, max= 1)     # min= a,  max= b
punif(q, min= 0, max= 1)
qunif(p, min= 0, max= 1)
runif(n, min= 0, max= 1)
```

A.2.2 Beta distribution

Notation: $X \sim \mathrm{Beta}(a, b)$, $a > 0, b > 0$.

Density: $\mathrm{Beta}(x|a, b) = x^{a-1}(1 - x)^{b-1}/B(a, b)$, $0 < x < 1$.

Moments: $E(X) = a/(a + b)$, $E(X^2) = a(a + 1)/[(a + b)(a + b + 1)]$, $\mathrm{Var}(X) = ab/[(a + b)^2(a + b + 1)]$.

Properties: If $Y_1 \sim \mathrm{Gamma}(a, 1)$, $Y_2 \sim \mathrm{Gamma}(b, 1)$, and $Y_1 \perp\!\!\!\perp Y_2$, then $Y_1/(Y_1 + Y_2) \sim \mathrm{Beta}(a, b)$.

Note: When $a = b = 1$, $\mathrm{Beta}(1, 1) = U(0, 1)$.

Functions:

```
dbeta(x, shape1, shape2)   # shape1= a,  shape2= b
pbeta(q, shape1, shape2)
qbeta(p, shape1, shape2)
rbeta(n, shape1, shape2)
```

A.2.3 Exponential distribution

Notation: $X \sim \mathrm{Exponential}(\beta)$, rate parameter $\beta > 0$.

Density: $\mathrm{Exponential}(x|\beta) = \beta\, \mathrm{e}^{-\beta x}$, $x > 0$.

Moments: $E(X) = 1/\beta$, $\mathrm{Var}(X) = 1/\beta^2$.

Properties: • If $U \sim U(0,1)$, then $-\frac{\log U}{\beta} \sim \text{Exponential}(\beta)$;

• If $\{X_i\}_{i=1}^{n} \overset{\text{iid}}{\sim} \text{Exponential}(\beta)$, then $\sum_{i=1}^{n} X_i \sim \text{Gamma}(n,\beta)$.

Functions:

```
dexp(x, rate= 1)          # rate= β
pexp(q, rate= 1)
qexp(p, rate= 1)
rexp(n, rate= 1)
```

A.2.4 Gamma distribution

Notation: $X \sim \text{Gamma}(\alpha,\beta)$, shape parameter $\alpha > 0$, rate parameter $\beta > 0$.

Density: $\text{Gamma}(x|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}\, \mathrm{e}^{-\beta x}$, $x > 0$.

Moments: $E(X) = \alpha/\beta$, $\text{Var}(X) = \alpha/\beta^2$.

Properties: • If $X \sim \text{Gamma}(\alpha,\beta)$ and $c > 0$, then $cX \sim \text{Gamma}(\alpha,\beta/c)$;

• If $\{X_i\}_{i=1}^{n} \overset{\text{ind}}{\sim} \text{Gamma}(\alpha_i,\beta)$, then $\sum X_i \sim \text{Gamma}(\sum \alpha_i,\beta)$;

• $\Gamma(\alpha+1) = \alpha\Gamma(\alpha)$, $\Gamma(1) = 1$ and $\Gamma(1/2) = \sqrt{\pi}$.

Note: $\text{Gamma}(1,\beta) = \text{Exponential}(\beta)$. $\text{Gamma}(\nu/2, 1/2) = \chi^2(\nu)$.

Functions:

```
dgamma(x, shape, rate= 1)      # shape= α,   rate= β
pgamma(q, shape, rate= 1)
qgamma(p, shape, rate= 1)
rgamma(n, shape, rate= 1)
```

A.2.5 Chi-squared distribution

Notation: $X \sim \chi^2(n) \equiv \text{Gamma}(\frac{n}{2}, \frac{1}{2})$, degree of freedom $n > 0$.

Density: $\chi^2(x|n) = \frac{2^{-n/2}}{\Gamma(n/2)} x^{n/2-1}\, \mathrm{e}^{-x/2}$, $x > 0$.

Moments: $E(X) = n$, $\text{Var}(X) = 2n$.

Properties: • If $Y \sim N(0,1)$, then $X = Y^2 \sim \chi^2(1)$;

• If $\{X_j\}_{j=1}^{m} \overset{\text{ind}}{\sim} \chi^2(n_j)$, then $\sum_{j=1}^{m} X_j \sim \chi^2(\sum_{j=1}^{m} n_j)$.

Functions:

```
dchisq(x, df)          # df = n
pchisq(q, df)
qchisq(p, df)
rchisq(nn, df)
```

```
=================================================================
> x <- seq(0.01, 25, 0.1)
> par(mfrow=c(2, 2))                                # Figure A.3
> curve(dchisq(x, df= 1), from=0.1, to = 25)
> curve(dchisq(x, df= 2), from=0.1, to = 25)
> curve(dchisq(x, df= 3), from=0.1, to = 25)
> curve(dchisq(x, df= 4), from=0.1, to = 25)
*****************************************************************
```

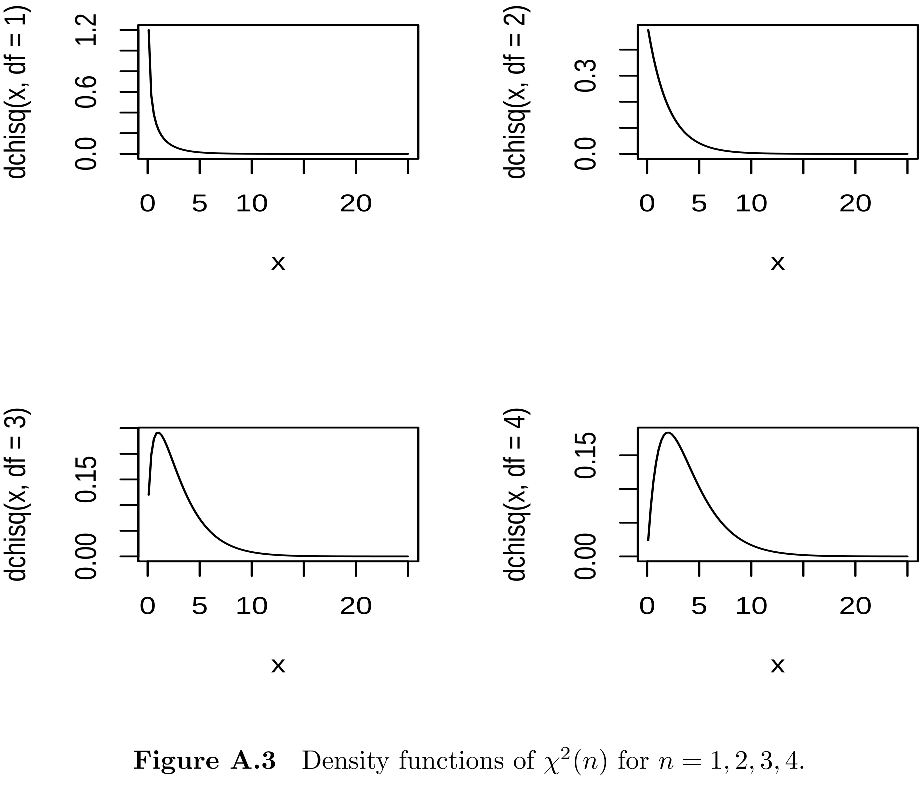

Figure A.3 Density functions of $\chi^2(n)$ for $n = 1, 2, 3, 4$.

A.2.6 *t*- or Student's *t*-distribution

Notation: $X \sim t(n)$, n is a positive integer.

Density: $$t(x|n) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{\pi n}\,\Gamma(\frac{n}{2})}\left(1+\frac{x^2}{n}\right)^{-\frac{n+1}{2}}, \quad -\infty < x \leqslant \infty.$$

Moments: $E(X) = 0$ (if $n > 1$), $\mathrm{Var}(X) = \frac{n}{n-2}$ (if $n > 2$).

Properties: Let $Z \sim N(0,1)$, $Y \sim \chi^2(n)$, and $Z \perp\!\!\!\perp Y$, then

$$\frac{Z}{\sqrt{Y/n}} \sim t(n).$$

Note: When $n = 1$, $t(n) = t(1)$ is called *standard Cauchy distribution*, whose mean and variance do not exist.

Functions:

```
dt(x, df)            # df = n
pt(q, df)
qt(p, df)
rt(nn, df)
```

A.2.7 F or Fisher's F-distribution

Notation: $X \sim F(n_1, n_2)$, n_1, n_2 are positive integers.

Density: $$F(x|n_1,n_2) = \frac{(n_1/n_2)^{n_1/2}}{B(\frac{n_1}{2},\frac{n_2}{2})}x^{\frac{n_1}{2}-1}(1+\frac{n_1 x}{n_2})^{-\frac{n_1+n_2}{2}},\ x > 0.$$

Moments: $E(X) = \frac{n_2}{n_2-2}$ (if $n_2 > 2$), $\mathrm{Var}(X) = \frac{2n_2^2(n_1+n_2-2)}{n_1(n_2-4)(n_2-2)^2}$ (if $n_2 > 4$).

Properties: Let $Y_i \sim \chi^2(n_i)$, $i = 1, 2$, and $Y_1 \perp\!\!\!\perp Y_2$, then

$$\frac{Y_1/n_1}{Y_2/n_2} \sim F(n_1, n_2).$$

Functions:

```
df(x, df1, df2)            # df1= n1,   df2= n2
pf(q, df1, df2)
qf(p, df1, df2)
rf(n, df1, df2)
```

A.2.8 Normal or Gaussian distribution

Notation: $X \sim N(\mu, \sigma^2)$, $-\infty < \mu < \infty$, $\sigma^2 > 0$.

Density: $N(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{(x-\mu)^2}{2\sigma^2}]$, $-\infty < x < \infty$.

Moments: $E(X) = \mu$, $\mathrm{Var}(X) = \sigma^2$.

Properties:
- If $\{X_i\} \overset{\text{ind}}{\sim} N(\mu_i, \sigma_i^2)$, then $\sum a_i X_i \sim N(\sum a_i \mu_i, \sum a_i^2 \sigma_i^2)$;
- If $X_1|X_2 \sim N(X_2, \sigma_1^2)$ and $X_2 \sim N(\mu_2, \sigma_2^2)$, then

$$X_1 \sim N(\mu_2, \sigma_1^2 + \sigma_2^2).$$

Functions:

```
dnorm(x, mean=0, sd= 1)   # mean= μ, sd= σ
pnorm(q, mean=0, sd= 1)
qnorm(p, mean=0, sd= 1)
rnorm(n, mean=0, sd= 1)
```

```
==============================================================
> x <- seq(-4, 4, 0.1)
> plot(x, dnorm(x), type="l",
                          ylab="Density function of N(0,1)")
# Note that this is the letter "l", not the digit "1"
                                                  # Figure A.4
--------------------------------------------------------------
# An alternative way of creating the plot is

> curve(dnorm(x), from=-4, to = 4,
                          ylab="Density function of N(0,1)")
**************************************************************
```

A.2.9 Multivariate normal or Gaussian distribution

Notation: $\mathbf{x} = (X_1, \ldots, X_d)^\top \sim N_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\Sigma} > 0$.

Density: $N_d(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(\sqrt{2\pi})^d |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}$, $\boldsymbol{x} \in \mathbb{R}^d$.

Moments: $E(\mathbf{x}) = \boldsymbol{\mu}$, $\mathrm{Var}(\mathbf{x}) = \boldsymbol{\Sigma}$.

Functions: Producing one or more samples from the specified multivariate normal distribution

```
mvrnorm(n= 1, mu, Sigma, tol= 1e-6, empirical= F)
rmvn(n, mu, V)
```

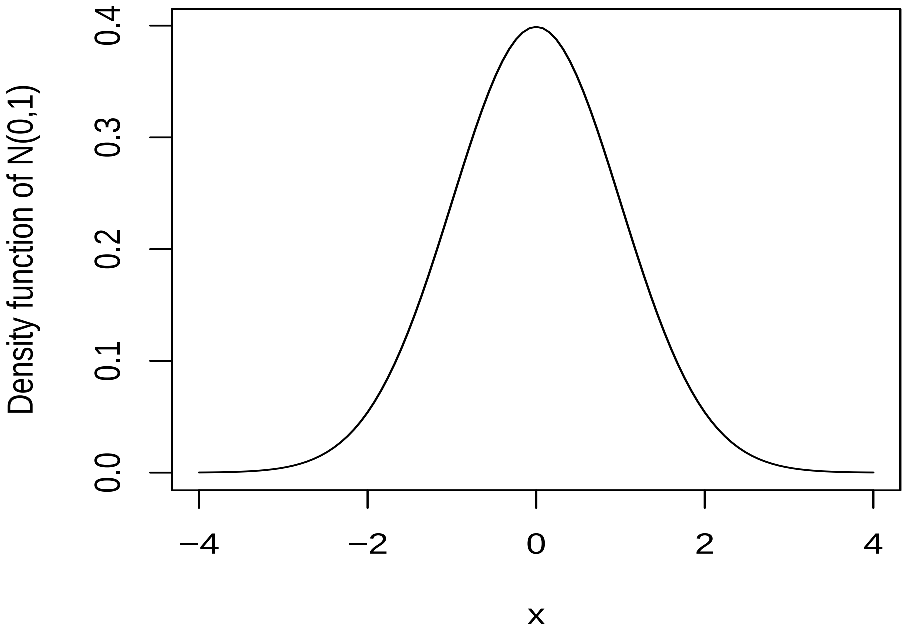

Figure A.4 Density functions of $N(0, 1)$.

APPENDIX B

R Programming

B.1 What is R?

R is a statistical computer program, made available through the Internet under the general public license. That is, it is supplied with a license that allows you to use it freely, distribute it, or even sell it, as long as the receiver has the same rights and the source code is freely available.

R provides an environment in which you can perform statistical analysis and produce graphics. It is designed in such a way that it is always possible to do further computations on the results of a statistical procedure. It is actually a complete programming language. Here we only learn the elementary concepts and see a number of cookbook examples.

R owes its name to typical Internet humor. You may be familiar with the programming language C (whose name is a story in itself). Inspired by this, Becker and Chambers chose in the early 1980s to call their newly developed statistical programming language S. This language was further developed into the commercial product S-plus, which by the end of the decade was in widespread use among statisticians of all kinds. Ross Ihaka and Robert Gentleman from the University of Auckland, New Zealand, chose to write a reduced version of S for teaching purpose. In 1995, Martin Maechler persuaded Ross and Robert to release the source codes for R under the general public license.

R implements a dialect of the S language. There are some differences, but in everyday use the two are very similar. However, some functions do differ, often because the R version tries to simplify things for the user. The differences are not all that big.

B.2 Obtaining R

The way to obtain R is to download it from one of the CRAN (Comprehensive R Archive Network) sites. The main site is

`http://cran.r.-project.org/`

It has a number of mirror sites worldwide, which may be closer to you and give faster download times. Installation details tend to vary over time, so you should read the accompanying documents and any other information offered on CRAN.

Information and further Internet resources for R can be obtained from the R home-page at

www.r-project.org

B.3 Basic commands

B.3.1 Expressions

1• Numeric expressions

- When R is ready for input, it prints out its prompt, a '>'.
- One of the simplest possible tasks in R is to enter an expression and receive a result (the second line is the answer from the computer).

```
> 2 + 2
 [1] 4
```

- So the computer knows that 2 plus 2 makes 4.
- Of course, it also knows how to do other standard calculations.
- For example, the following is how to compute e^{-2}:

```
> exp(-2)
[1] 0.13533528
```

Table B.1 *Arithmetic operators*

Operator	Meaning	Expression	Results
+	plus, addition	$4 + 3$	7
−	minus, subtraction, sign	$9 - 5$	4
$*$	times, multiplication	$3 * 5$	15
/	division	7/3	2.3333
		8/3	2.6667
% / %	integer division	7 %/% 3	2
		8 %/% 3	3
^	power	2^3	8

2• Commonly used functions

Table B.2 *Commonly used functions*

R function	Meaning
`sqrt()`	square root
`log()`	natural logarithm
`log10()`	logarithm base 10
`exp()`	exponential
`abs()`	absolute value
`round()`	round to nearest integer
`ceiling()`	round up
`floor()`	round down
`sin(), cos(), tan()`	sine, cosine, tangent
`asin(), acos(), atan()`	arc-sine, arc-cosine, arc-tangent
`min(x)`	smallest value in vector `x`
`min(x1, x2, ...)`	minimum over several vectors (one number)
`pmin(x1, x2, ...)`	parallel (element-wise) minimum over multiple equally long vectors
`max(x)`	largest value in vector `x`
`max(x1, x2, ...)`	maximum over several vectors (one number)
`pmax(x1, x2, ...)`	parallel (element-wise) maximum
`range(x)`	like `c(min(x), max(x))`
`length(x)`	number of elements in vector `x`

3• The `options()` and `help()` functions

- The `options()` function can be used to control the appearance of the output:

```
==========================================================
> options(width=68, digits=8)
> pi
[1] 3.1415927
> options(width=68, digits=4)
> pi
[1] 3.142
> -5/3
```

```
[1] -1.667
> 1/3
[1] 0.3333
> -1/3
[1] -0.3333
**********************************************************
```

- The `[1]` in front of the result is part of R's way of printing numbers and vectors.
- It is useless here, but it becomes useful when the result is a longer vector.
- Consider the case of generating 10 random numbers from uniform distribution on $(0, 1)$:

```
> runif(10)
 [1] 0.050808 0.195130 0.391954 0.300020 0.143770 0.895648
 [7] 0.031605 0.723146 0.528792 0.887409
```

- Here the `[7]` indicates that 0.031605 is the seventh element in the vector.
- More information about any R functions can be found using the `help()` function.

```
> help(t.test)
> ?t.test
> ??t.test
```

4• Logical expressions

- So far, we have mentioned values of type numeric.
- When a numeric value is missing, it is of type `NA`, i.e., not available.
- Another type in R is logical with three values, `TRUE` (or its abbreviation `T`), `FALSE` (or `F`), and `NA`.
- Logical operations are extremely useful when making comparisons and choosing particular elements from vectors and matrices.

- The symbols used for logical operations are listed in Table B.3.

Table B.3 *Logical operators*

Operator	Meaning
`<`	less than
`>`	greater than
`<=`	less than or equal to
`>=`	greater than or equal to
`==`	equal to
`!=`	not equal to
`&`	and
`|`	or
`!`	not
`is.na(x)`	missing?

- We can use a logical expression to assign a logical value.

```
=========================================================
> x <- 3 == 4
> x
[1] FALSE
> x <- 3 < 4
> x
[1] TRUE
> 3 == 4 & 3 < 4
[1] FALSE
> 3 == 4 | 3 < 4
[1] TRUE
> 1/0
[1] Inf
> is.numeric(3)
[1] TRUE
> is.character("3")
[1] TRUE
> is.infinite(1/0)
[1] TRUE
---------------------------------------------------------
```

```
> x <- 1:15
> x[x < 10]
[1] 1 2 3 4 5 6 7 8 9
> y <- c(rep(0, 10), rep(1, 5))
> x[y == 0]
 [1]  1  2  3  4  5  6  7  8  9 10
*********************************************************
```

B.3.2 Assignment operator

5• Assignment statement

- To assign the value 2 to the variable `x`, you can input

```
> x <- 2
```

- The two characters `<-` should be read as a single symbol: an arrow pointing to the variable to which the value is assigned.
- There is no immediately visible result, but from now on, `x` has the value 2 and can be used in subsequent calculations. For instance,

```
> x*x
[1] 4
```

- Assignment can also be made using the function assign().
- Assignments can also be made in the other direction.

```
=========================================================
> assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
> x
[1] 10.4  5.6  3.1  6.4 21.7
---------------------------------------------------------
>  1:4 -> y
> y
[1] 1 2 3 4
---------------------------------------------------------
> x1 <- 1; x2 <- -2; x3 <- 4
> c(x1, x2, x3)
```

```
[1]  1 -2  4
---------------------------------------------------------
> a <- b <- c <- 2
> c(a, b, c)
[1] 2 2 2
*********************************************************
```

6• Names of variables

- Names of variables in R can be built from letters, digits and the period (dot) symbol.
- However, names must not start with a digit and avoid starting with period.
- For example, `height.2yr` may be used to describe the height of a child at the age of 2 years.
- Names are case-sensitive: `WT` and `wt` do not refer to the same variable.
- Some names, e.g., `c, q, t, C, D, F, I, T, diff, df, pt` are already used/defined by the system.

B.4 Vectors and matrices

B.4.1 Vectors

7• Numeric vectors

7.1• The colon operator “:”

— Many methods can be used to generate vectors in R.

— The simplest way is to use the colon operator.

— The colon operator has high priority within an expression.

```
===========================================================
> x <- 1:5
> x
[1] 1 2 3 4 5
```

```
> 5:1
[1] 5 4 3 2 1
> 2*1:5
[1]  2  4  6  8 10
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> is.vector(5:1)
[1] TRUE
> is.vector(5:1, mode="integer")
[1] TRUE
> is.vector(5:1, mode="numeric")
[1] TRUE
> is.vector(5:1, mode="character")
[1] FALSE
> is.matrix(5:1)
[1] FALSE
----------------------------------------------------------
> n <- 10
> 1:n-1                                   # = (1:n) - 1
 [1] 0 1 2 3 4 5 6 7 8 9
> 1:(n-1)
[1] 1 2 3 4 5 6 7 8 9
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> is.numeric(n)
[1] TRUE
> is.character(n)
[1] FALSE
**********************************************************
```

7.2• Concatenate function `c()`

— `c()` function is the second way to generate a (column) vector.

— The number of elements in a vector can be determined using the `length()` function.

— The `t()` function transposes an n-dimensional column vector into a $1 \times n$ matrix.

— In R, there is no row vector.

```
============================================================
> x <- c(1,2,3,4)
> x
[1] 1 2 3 4
> length(x)
[1] 4
> tx <- t(x)
> tx
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
> is.vector(tx)
[1] FALSE
> is.matrix(tx)
[1] TRUE
************************************************************
```

7.3• Sequence function `seq()`

— The general syntax of `seq()` is

```
============================================================
seq(from= 1, to = 1, by = (to - from)/(length.out - 1),
                      length.out = NULL)

# Typical usages are

seq(from, to)
seq(from, to, by= )
seq(from, to, length.out= )
************************************************************
```

— If `by` is 1, the `seq()` function can be replaced by `from:to`.

```
============================================================
> seq(from= 1, to= 4)
[1] 1 2 3 4
> seq(1, 4)
[1] 1 2 3 4
```

```
> seq(from= 1, to= 4, by= 1)
[1] 1 2 3 4
> 1:4
[1] 1 2 3 4
> seq(-10, 0, 1)
 [1] -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0
-----------------------------------------------------------
> seq(-pi, pi, length= 10)  # length = length.out
 [1] -3.14159 -2.44346 -1.74533 -1.04720 -0.34907
 [6]  0.34907  1.04720  1.74533  2.44346  3.14159
-----------------------------------------------------------
> seq(0, 1, by= 0.1)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> seq(0, 1, length.out = 11)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
***********************************************************
```

7.4• Repeat function `rep()`

— The general syntax of `rep()` is

```
rep(x, times= 1, length.out= NA, each= 1)
```

```
===========================================================
> rep(10, times= 5)
[1] 10 10 10 10 10
> rep(1:3, 3)
[1] 1 2 3 1 2 3 1 2 3
> rep(1:3, c(3,3,3))
[1] 1 1 1 2 2 2 3 3 3
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> rep(1:3, c(1,2,3))
[1] 1 2 2 3 3 3
> rep(c(2,3,4,5), 1:4)
 [1] 2 3 3 4 4 4 5 5 5 5
-----------------------------------------------------------
> rep(1:4, 2)
[1] 1 2 3 4 1 2 3 4
```

```
> rep(1:4, each= 2)
[1] 1 1 2 2 3 3 4 4
> rep(1:4, c(2,2,2,2))
[1] 1 1 2 2 3 3 4 4
> rep(1:4, c(2,1,2,1))
[1] 1 1 2 3 3 4
> rep(1:4, each= 2, len= 4)         # first 4 only
[1] 1 1 2 2
> rep(1:4, each= 2, length.out= 4)  # length.out = len
[1] 1 1 2 2
> rep(1:4, each = 2, len = 10)      # 8 integers plus two
 [1] 1 1 2 2 3 3 4 4 1 1            # recycled 1’s
> rep(1:4, each= 2, times= 3)
 [1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
***********************************************************
```

7.5• Obtaining a sub-vector by square brackets

```
================================================================
> x <- 1:6
> x[3]           # [1] 3
> x[c(1,3)]
[1] 1 3
> x[1:3]
[1] 1 2 3
> x[-2]
[1] 1 3 4 5 6
> x[-c(2,4)]
[1] 1 3 5 6
----------------------------------------------------------------
> y <- c(3, 3, 3, 3, 3, 3)
> x[x>y]
[1] 4 5 6
> x[x==y]
[1] 3
> x[x!=y]
[1] 1 2 4 5 6
****************************************************************
```

7.6• Missing values

— Missing data are frequently encountered in practice (e.g., some patient withdrew from a study; an experiment failed).

— In R, missing value is denoted by NA.

— Operations on NA yield NA as the result.

```
============================================================
> x <- c(1, 2, 3, NA, 5)
> x
[1]  1  2  3 NA  5
> x+1
[1]  2  3  4 NA  6
> x*4
[1]  4  8 12 NA 20
> x*0
[1]  0  0  0 NA  0
------------------------------------------------------------
> y <- c(1, NA, 3, 4, 5)
> x+y
[1]  2 NA  6 NA 10
> x*y
[1]  1 NA  9 NA 25
------------------------------------------------------------
> is.na(x)
[1] FALSE FALSE FALSE  TRUE FALSE
> is.vector(x)
[1] TRUE
> is.vector(x, mode="integer")
[1] FALSE
> is.vector(x, mode="character")
[1] FALSE
> is.vector(x, mode="numeric")
[1] TRUE
************************************************************
```

8• Character vectors

8.1$^\bullet$ Concatenate function c()

— A character vector is a vector of (text) strings, whose elements are specified and printed in double quotes.

— It also works with a mixture of numeric and string values, but in this case, all elements will be converted to strings

```
============================================================
> BMN <- c("Brain", "Mouth", "Nose")
> BMN
[1] "Brain" "Mouth" "Nose"
> > is.character(BMN)
[1] TRUE
> is.vector(BMN)
[1] TRUE
> is.vector(BMN, mode="character")
[1] TRUE
------------------------------------------------------------
> mix <- c(BMN, 45, -20)
> mix
[1] "Brain" "Mouth" "Nose"  "45"    "-20"
************************************************************
```

8.2$^\bullet$ The paste() function

— The `paste()` function takes an arbitrary number of arguments and concatenates them one by one into character strings.

— The general syntax of `paste()` is

```
paste (..., sep = " ", collapse = NULL)
```

```
============================================================
>  paste(c("X","Y"), 1:4)
[1] "X 1" "Y 2" "X 3" "Y 4"

> paste(c("X","Y"), 1:4, sep="")
[1] "X1" "Y2" "X3" "Y4"
```

```
> paste(c("X","Y"), 1:4, sep="_")
[1] "X_1" "Y_2" "X_3" "Y_4"
-----------------------------------------------------------
> x <- c("st", "nd", "rd", "th", "th")
> paste(1:5, x, sep="-")
[1] "1-st" "2-nd" "3-rd" "4-th" "5-th"

> paste(1:5, x, sep= "-", collapse =" ")
[1] "1-st 2-nd 3-rd 4-th 5-th"

> paste(1:5, x, sep= "-", collapse =", ")
[1] "1-st, 2-nd, 3-rd, 4-th, 5-th"

> paste(1:5, x, sep= "-", collapse =" | ")
[1] "1-st | 2-nd | 3-rd | 4-th | 5-th"
-----------------------------------------------------------
> p <- 0.03
> paste("The p-value = ", p)
[1] "The p-value =  0.03"
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> tv <- 2.14; pv <- 0.03
> paste(c("The t-value is ", "The p-value = "), c(tv, pv))
[1] "The t-value is  2.14" "The p-value =  0.03"

> paste(c("The t-value is ", "the p-value = "), c(tv, pv),
                                        collapse =" and ")
[1] "The t-value is  2.14 and the p-value =  0.03"
***********************************************************
```

9• Logical vectors

- The elements of a logical vector can have the values `TRUE` (or its abbreviation `T`), `FALSE` (or `F`), and `NA`.

```
=========================================================
> c(T, T, F, T)
[1]  TRUE  TRUE FALSE  TRUE
```

```
---------------------------------------------------------
> x <- c(1, 2, 3, NA, 5)
> is.na(x)
[1] FALSE FALSE FALSE  TRUE FALSE
> is.vector(is.na(x), mode="logical")
[1] TRUE
> !is.na(x)
[1]  TRUE  TRUE  TRUE FALSE  TRUE
> x[!is.na(x)]      # A vector containing the non-missing
[1] 1 2 3 5         # values of x
---------------------------------------------------------
> x <- c(-1, 2, 3, NA, -5, 6)
> x[x>0]
[1]  2  3 NA  6
> x[(!is.na(x)) & x>0]
[1] 2 3 6
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> x+1
[1]  0  3  4 NA -4  7
> (x+1)[(!is.na(x)) & x>0]
[1] 3 4 7
# A sub-vector of x+1 with those elements, where the
# corresponding elements in x are non-missing & positive
*********************************************************
```

B.4.2 Matrices

10• Dimension function dim()

- A matrix in mathematics is just a two-dimensional array of *numbers*.
- In R, the matrix notation is extended to elements of any type, e.g., a matrix of *character strings*.
- A matrix is represented as a vector with dimensions:

```
=========================================================
> M1 <- 1:15
> dim(M1) <- c(3, 5)       # The storage is column-wise
> M1
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> M1[, c(5, 4, 3, 1, 2)]       # no change in rows
     [,1] [,2] [,3] [,4] [,5]  # change order of columns
[1,]   13   10    7    1    4
[2,]   14   11    8    2    5
[3,]   15   12    9    3    6
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> M1[c(3, 2, 1), ]             # exchange row 3 with
     [,1] [,2] [,3] [,4] [,5]  # row 1
[1,]    3    6    9   12   15
[2,]    2    5    8   11   14
[3,]    1    4    7   10   13
---------------------------------------------------------
> M2 <- c("a1", "a2", "a3", "b1", "b2", "b3",
                            "c1", "c2", "c3")
> dim(M2) <- c(3, 3)
> M2
     [,1] [,2] [,3]
[1,] "a1" "b1" "c1"
[2,] "a2" "b2" "c2"
[3,] "a3" "b3" "c3"
---------------------------------------------------------
> M3 <- c(1:3, "a1", "a2", "a3")   # This why we need
> M3                               # data.frame()
[1] "1"  "2"  "3"  "a1" "a2" "a3"
> dim(M3) <- c(3, 2)
> M3
     [,1] [,2]
[1,] "1"  "a1"
[2,] "2"  "a2"
[3,] "3"  "a3"
*********************************************************
```

11• Matrix function matrix()

- The general syntax of `matrix()` is

  `matrix(data, nrow, ncol, byrow= F, dimnames= NULL).`

- `byrow = T` specifies that the matrix is to be filled row by row and `byrow = F` is the default.

```
=========================================================
> matrix(1:6, nrow = 2, byrow = T)        #list by rows
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> matrix(1:6, nrow = 2)
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
---------------------------------------------------------
> X <- matrix(1:6, 2, 3)      #list by columns (default)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> rownames(X)<- LETTERS[1:2]
> colnames(X)<- letters[1:3]
> X
  a b c
A 1 3 5
B 2 4 6
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> rownames(X) <- month.name[1:2]
> colnames(X) <- month.abb[1:3]
> X
         Jan Feb Mar
January    1   3   5
February   2   4   6
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> rownames(X) <- c("R1", "R2")
> colnames(X) <- c("C1", "C2", "C3")
> X
   C1 C2 C3
R1  1  3  5
R2  2  4  6
---------------------------------------------------------
> Y <- matrix(1:6, nrow= 2, dimnames= list(c("R1", "R2"),
                                   c("C1", "C2", "C3")))
> Y
   C1 C2 C3
R1  1  3  5
R2  2  4  6
*********************************************************
```

12• Merging vectors/matrices by rbind() and cbind()

```
==============================================================
> M1 <- rbind(A= 1:4, B= 5:8, C= 9:12)
> M1
  [,1] [,2] [,3] [,4]
A    1    2    3    4
B    5    6    7    8
C    9   10   11   12
> M2 <- cbind(a= -(1:3), b= -(4:6), c= -(7:9))
> M2
      a  b  c
[1,] -1 -4 -7
[2,] -2 -5 -8
[3,] -3 -6 -9
> rownames(M2) <- LETTERS[1:3]
> M2
   a  b  c
A -1 -4 -7
B -2 -5 -8
C -3 -6 -9
--------------------------------------------------------------
> M <- cbind(M1, M2)
> M
              a  b  c
```

```
A 1  2  3  4 -1 -4 -7
B 5  6  7  8 -2 -5 -8
C 9 10 11 12 -3 -6 -9
> colnames(M)[1:4] <- month.abb[1:4]
> M
  Jan Feb Mar Apr  a  b  c
A   1   2   3   4 -1 -4 -7
B   5   6   7   8 -2 -5 -8
C   9  10  11  12 -3 -6 -9
--------------------------------------------------------------
> cbind(1, 1:3)
     [,1] [,2]
[1,]    1    1
[2,]    1    2
[3,]    1    3
> cbind(1:3, diag(3))
     [,1] [,2] [,3] [,4]
[1,]    1    1    0    0
[2,]    2    0    1    0
[3,]    3    0    0    1
**************************************************************
```

13• Obtaining a sub-matrix by square brackets

```
==============================================================
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
--------------------------------------------------------------
> X[2, 3]      # X[i,j] is the (i,j)-th element of X
[1] 6
> X[1, ]       # X[i, ] is the vector of the i-th row of X
[1] 1 3 5
> X[, 2]       # X[, j] is the vector of the j-th column of X
[1] 3 4
--------------------------------------------------------------
> X * X               # multiply element by element
```

```
     [,1] [,2] [,3]
[1,]    1    9   25
[2,]    4   16   36
> X %*% t(X)           # matrix multiplication
     [,1] [,2]
[1,]   35   44
[2,]   44   56
--------------------------------------------------------------
> dim(X)               # dimension function
[1] 2 3
> nrow(X)              # number of rows of X
[1] 2
> ncol(X)              # number of columns of X
[1] 3
**************************************************************
```

14• The diag() function

- If n is a single numeric value, then diag(n) is the n by n identity matrix.
- If v is a vector, then diag(v) gives a diagonal matrix.
- If M is a matrix, then diag(M) gives the vector of main diagonal entries of M.

```
=========================================================
> diag(3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
---------------------------------------------------------
> diag(1:3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
---------------------------------------------------------
```

```
> M <- matrix(1:9, 3, 3)
> M
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> diag(M)
[1] 1 5 9
---------------------------------------------------------
> diag(diag(M))
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    5    0
[3,]    0    0    9
*********************************************************
```

15• The crossprod() function

- crossprod(X, y) is the same as t(X) %*% y but the operation is more efficient.
- If the second argument to crossprod() is omitted, it is taken to be the same as the first.

```
=========================================================
> x <- 1:3
> crossprod(x)          # = crossprod(x, x) = t(x) %*% x
     [,1]
[1,]   14
> t(x) %*% x
     [,1]
[1,]   14
> x %*% t(x)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    4    6
[3,]    3    6    9
*********************************************************
```

16• Linear equations and matrix inversion

- Let $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, then $\boldsymbol{x} = \mathbf{A}^{-1}\boldsymbol{b}$ is the solution of the linear equations.
- In R, we use `x <- solve(A, b)`.
- Although, we can compute `x <- solve(A) %*% b`, it is inefficient and potentially unstable.
- The quadratic form $\boldsymbol{x}^{\top}\mathbf{A}^{-1}\boldsymbol{x}$ should be computed as `x %*% solve(A,x)`.

17• Eigenvalues and eigenvectors for a symmetric matrix

- Let $\mathbf{A} = (a_{ij})$ be an $n \times n$ symmetric matrix, then $\mathbf{A} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}^{\top}$ or $\mathbf{A}\boldsymbol{\Gamma} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}$, where
  - — $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$,
  - — $\{\lambda_1, \ldots, \lambda_n\}$ are eigenvalues of $\mathbf{A}$,
  - — $\boldsymbol{\Gamma} = (\gamma_1, \ldots, \gamma_n)$ is orthogonal satisfying $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\top} = \boldsymbol{\Gamma}^{\top}\boldsymbol{\Gamma} = \mathbf{I}_n$, and
  - — $\{\gamma_1, \ldots, \gamma_n\}$ are corresponding eigenvectors of $\mathbf{A}$.
- The function `eigen(A)` calculates the eigenvalues and eigenvectors of a symmetric matrix `A`.
  - — The result of this function is a list of two components named `values` and `vectors`.
- The function `det(A)` computes the determinant of an arbitrary square matrix $\mathbf{A}$.
  - — However, for a symmetric matrix $\mathbf{A}$, we have $|\mathbf{A}| = \prod_{i=1}^{n} \lambda_i$.
- The trace of $\mathbf{A}$ is defined as $\mathrm{tr}\,(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$ for any square matrix $\mathbf{A}$.
  - — Note that, there is no `tr(A)` function in R because it is simply computed as `sum(diag(A))`.
  - — However, for the symmetric matrix $\mathbf{A}$, we have $\mathrm{tr}\,(\mathbf{A}) = \sum_{i=1}^{n} \lambda_i$.

```
==========================================================
> X <- matrix(rnorm(9), 3, 3)
> A <- t(X) %*% X
> ev <- eigen(A)
> ev
$values
[1] 2.2833 0.8654 0.4781

$vectors
         [,1]     [,2]     [,3]
[1,] -0.08887  0.92143  0.37825
[2,]  0.02771  0.38190 -0.92379
[3,] -0.99566 -0.07161 -0.05947
----------------------------------------------------------
> L <- diag(ev$values); G <- ev$vectors;
> A %*% G                        #  Checking: A G = G L
         [,1]     [,2]     [,3]
[1,] -0.20291  0.79736  0.18085
[2,]  0.06326  0.33048 -0.44169
[3,] -2.27335 -0.06197 -0.02843
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> G %*% L
         [,1]     [,2]     [,3]
[1,] -0.20291  0.79736  0.18085
[2,]  0.06326  0.33048 -0.44169
[3,] -2.27335 -0.06197 -0.02843
----------------------------------------------------------
> G %*% t(G)                         # Checking: G G'
          [,1]       [,2]       [,3] #  = G' G = I_n
[1,] 1.000e+00  1.110e-16  5.551e-17
[2,] 1.110e-16  1.000e+00 -6.939e-18
[3,] 5.551e-17 -6.939e-18  1.000e+00
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> t(G) %*% G
           [,1]       [,2]      [,3]
[1,]  1.000e+00 -4.163e-17 1.388e-17
[2,] -4.163e-17  1.000e+00 5.985e-17
[3,]  1.388e-17  5.985e-17 1.000e+00
```

```
---------------------------------------------------------
> det(A)                    # = 2.2833 *0.8654 *0.4781
[1] 0.9447
> prod(ev$values)           # = 2.2833 *0.8654 *0.4781
[1] 0.9447
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> sum(diag(A))              # tr(A)
[1] 3.627
> sum(ev$values)
[1] 3.627
*********************************************************
```

B.5 Lists, data frames and arrays

B.5.1 Lists

18• Why need we `list()` besides vectors and matrices?

- An R `list()` is an object consisting of a collection of objects known as its *components*.
- The components could consist of a numeric vector, a logical value, a matrix, a complex vector, a character array, a function, and so on.
- Vectors and matrices are not enough to store such data. For example, the outcome of `eigen()` is a list of two components: a vector of eigenvalues and a matrix of eigenvectors.

```
=========================================================
> L <- list(husband= "Fred", wife= "Mary",
              number.children= 3, child.ages= c(4,7,9) )
> L                          # is the name of this list
$husband                     # with four components
[1] "Fred"

$wife
[1] "Mary"

$number.children
[1] 3
```

```
     $child.ages
     [1] 4 7 9
     ---------------------------------------------------------
     > length(L)     # gives the number of components
     [1] 4
     ---------------------------------------------------------
     > L$husband     # = L[[1]] = L[["husband"]]
     [1] "Fred"      # name of the 1-st component of the list
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     > L$wife        # = L[[2]] = L[["wife"]]
     [1] "Mary"      #    [[ ]] : double square brackets
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     > L$child.ages  # = L[[4]]
     [1] 4 7 9
     > L[[4]][1]     # the 1-st element of the vector L[[4]]
     [1] 4
     ---------------------------------------------------------
     > L[1]             # a sub-list with the first component
     $husband
     [1] "Fred"
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     > L[4]
     $child.ages
     [1] 4 7 9
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     > L[c(1, 2)]       # a sub-list with two components
     $husband
     [1] "Fred"

     $wife
     [1] "Mary"
     *********************************************************
```

19• Forming a new list from existing lists via c()

```
==============================================================
> La <- list(score.child= c(80, 90, 100), university.child=
                                      c("U1", "U2", "U3"))
> La
$score.child
[1]  80  90 100

$university.child
[1] "U1" "U2" "U3"
--------------------------------------------------------------
> L.new <- c(L, La)
> L.new
$husband
[1] "Fred"

$wife
[1] "Mary"

$number.children
[1] 3

$child.ages
[1] 4 7 9

$score.child
[1]  80  90 100

$university.child
[1] "U1" "U2" "U3"
**************************************************************
```

B.5.2 Data frames

20• Why do we need `data.frame()` besides matrices?

- We have only three kinds of matrix: numeric matrix, character matrix, and logical matrix.

- A data frame, a matrix-like structure whose columns may be of differing types (numeric, character, logical, factor and so on).

21• Data entry

21.1• Creating data frame from pre-existing variables

```
==============================================================
> sex <- c(rep("F", 3), rep("M", 3))
> sex
[1] "F" "F" "F" "M" "M" "M"
> y <- 1:6
--------------------------------------------------------------
> d <- data.frame(y= y, sex= sex)
> d
  y sex
1 1   F
2 2   F
3 3   F
4 4   M
5 5   M
6 6   M
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> d$y
[1] 1 2 3 4 5 6
> d$sex          # character vectors is coerced to be factors
[1] F F F M M M
Levels: F M
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> is.data.frame(d)
[1] TRUE
> is.vector(d$y, mode="numeric")
[1] TRUE
> is.vector(d$sex, mode="character")
[1] FALSE
> is.factor(d$sex)
[1] TRUE
--------------------------------------------------------------
> d$z <- 6:1               # add a new variable
> d
  y sex z
1 1   F 6
```

```
2 2   F 5
3 3   F 4
4 4   M 3
5 5   M 2
6 6   M 1
> d <- d[c(1, 3, 2)]          # insert a new variable
> d
  y z sex
1 1 6   F
2 2 5   F
3 3 4   F
4 4 3   M
5 5 2   M
6 6 1   M
> d <- d[-2]                  # delete the 2-nd column
> d
  y sex
1 1   F
2 2   F
3 3   F
4 4   M
5 5   M
6 6   M
**************************************************************
```

21.2• The data-frame editor for small data sets

— To enter data into a blank data frame, use

```
===========================================================
> dd <- data.frame()
> fix(dd)
***********************************************************
```

— This brings up a spreadsheet-like editor.

— An alternative would be `dd <- edit(data.frame())`.

22• Indexing of data frames

```
==============================================================
> d
     y sex
1  1.2   F
2  3.0   F
3  2.5   F
4 -2.6   M
5 10.0   M
6  7.0   M
--------------------------------------------------------------
> d[5, 1]
[1] 10
> d[5, 2]
[1] M
Levels: F M
> d[5, ]
   y sex
5 10   M
> d[, 2]
[1] F F F M M M
Levels: F M
--------------------------------------------------------------
> d[d$y>2, ]
     y sex
2  3.0   F
3  2.5   F
5 10.0   M
6  7.0   M
**************************************************************
```

23• subset AND transform

```
==============================================================
> d
     y sex
1  1.2   F
2  3.0   F
3  2.5   F
```

```
4 -2.6   M
5 10.0   M
6  7.0   M
--------------------------------------------------------------
> d2 <- subset(d, d$y>2)        # delete some rows
> d2
     y sex
2  3.0   F
3  2.5   F
5 10.0   M
6  7.0   M
--------------------------------------------------------------
> d3 <- transform(d, z= y*y)    # add a row named as z
> d3
     y sex      z
1  1.2   F   1.44
2  3.0   F   9.00
3  2.5   F   6.25
4 -2.6   M   6.76
5 10.0   M 100.00
6  7.0   M  49.00
**************************************************************
```

B.5.3 Arrays

24• Dimension function dim()

- A vector is a 1-dimensional array.
- A matrix is a 2-dimensional array.
- The following is an example of 3–dimensional array.

```
==============================================================
> z <- 1:24
> dim(z) <- c(2, 4, 3)
> z
, , 1

     [,1] [,2] [,3] [,4]
```

```
[1,]    1    3    5    7
[2,]    2    4    6    8

, , 2

     [,1] [,2] [,3] [,4]
[1,]    9   11   13   15
[2,]   10   12   14   16

, , 3

     [,1] [,2] [,3] [,4]
[1,]   17   19   21   23
[2,]   18   20   22   24
**************************************************************
```

25• Array function array()

```
==============================================================
> Z <- array(1:24, dim= c(3, 4, 2))
> Z
, , 1

     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2

     [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24
**************************************************************
```

B.6 Flow control

26• `while` STATEMENT

- The R allows conditional execution and looping constructs.
- Note that `while (condition) expression` construction, which says that the expression should be evaluated as long as the condition is `TRUE`.
- For example, suppose that we want to use a version of Newton's method for calculating the square root of y.

```
==========================================================
> y <- 12345
> x <- y/2
> while (abs(x^2 - y) > 1e-10)    x <- (x + y/x)/2
> x
[1] 111.11
> x^2
[1] 12345
**********************************************************
```

- The test occurs at the top of the loop so that the expression might never be evaluated.

27• `repeat` STATEMENT

- A variation of the same algorithm with test at the bottom of the loop can be written with a `repeat` construction:

```
==========================================================
> x <- y/2
> repeat {
+    x <- (x + y/x)/2
+    if (abs(x^2 - y) < 1e-10)  break
+}
> x
[1] 111.11
**********************************************************
```

28• Other loops

- A *compound expression*: several expressions held together between curly braces.
- An `if` construction for conditional execution.
- A `break` expression, which causes the enclosing loop to exit.
- Table 4 lists other loops.

Table B.4 *Flow controls*

Function	Meaning
`if(p<1) print(‘good’)`	conditional execution

`if(p<1) print(‘good’)`
`else print(‘bad’)`
	conditional execution
	with alternative

`for(i in 1:9) print(i)`	loop over list

B.7 User functions

29• Creating a user function

- The R provides an extremely powerful method of writing functions for specific tasks of interest.
- First, we use `ls()` to list all objects in the workspace, use `rm()` to remove objects from the working directory, use `q()` to terminate the current R session.
- Then, we use `fix()` to edit your new function.
- For example, when you type `fix(mysum)` and press the key "Enter", a window will jump out so that you can edit your function with name `mysum`.

```
==========================================================
function(a, b)
{
  # Function name: mysum(a, b)
  x <- a^2 + b^2
  return(x)
}
**********************************************************
```

- Here `a` and `b` are two arguments.

```
==========================================================
> mysum(3, 4)
[1] 25
**********************************************************
```

30• Inserting a sub-function into a main function

- When a main function requires to *repeatedly* call a sub-function, we can insert the sub-function into the main function.

- For example,

```
==========================================================
function(k)
{
  # Function name: main.mysum(k)
  nest.fun <- function(x, y, p)
  {
         (x + y)^p
  }
  x <- y <- 1:4
  z <- nest.fun(x,y,1) + nest.fun(x,y,2)*nest.fun(x,y,3)
  w <- sum(z)
  result <- list(z= z, w= w)
  return(result)
}
**********************************************************
```

B.8 Some commonly used R functions for data analysis

31• apply() function

- Table B.5 lists some statistical functions.
- For large data sets, we can use the `apply()` function.
- The syntax is

  ```
  apply(object, dim, function)
  ```

  where `object` is the name of a matrix, `dim` can take the value 1 (row) or 2 (column), and `function` is the name of an R function (already available or created by the user).

```
==========================================================
> X <- matrix(1:9, 3, 3)
> X
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> apply(X, 2, sum)
[1]  6 15 24
> apply(X, 1, mean)
[1] 4 5 6
> apply(X, 2, median)
[1] 2 5 8
> apply(X, 1, var)
[1] 9 9 9
**********************************************************
```

Table B.5 *Some statistical functions*

Function	Meaning
`sum()`	summation
`prod()`	multiplication
`mean()`	average
`var()`	variance
`sd()`	standard deviation
`median()`	median
`quantile(x, p)`	quantiles
`cor(x, y)`	correlation

Table B.6 *Data manipulation functions*

Function	Meaning
`sort(x)`	returns a vector which is a sorted version of x
`order(x)`	returns an integer vector containing the permutation that will sort x into ascending order
`sort.list(x)`	= `order(x)`
`rank(x)`	returns a vector of the ranks of x
`rev(x)`	returns an object with the same length as x but with the elements or components in the reverse order

Table B.7 *Normal distribution*

Function	Meaning
`dnorm(x, mean, sd)`	density
`pnorm(x, mean, sd)`	cumulative distribution function
`qnorm(p, mean, sd)`	lower p-quantile, $x, \Pr(X \leqslant x) = p$
`rnorm(n, mean, sd)`	n random numbers

Table B.8 *Cumulative distribution functions*

Function	Meaning
`pnorm(x, mean, sd)`	normal
`plnorm(x, mean, sd)`	log-normal
`pt(x, df)`	Student-t
`pf(x, n1, n2)`	F
`pgamma(x, shape, scale)`	gamma
`pchisq(x, df)`	χ^2
`pexp(x, rate)`	exponential
`punif(x, min, max)`	uniform
`pbeta(x, a, b)`	beta
`pbinom(x, n, p)`	binomial
`ppois(x, lambda)`	Poisson

Table B.9 *Parametric and non-parametric methods for continuous data*

Function	Meaning
t.test	one- and two-sample t test
pairwise.t.test	pairwise comparisons
var.test	comparison of two variances (F test)
bartlett.test	Bartlett's test (k variances)
cor.test	correlation
cor.test variants:	
method=‘‘kendall’’	Kendall's τ
method=‘‘spearman’’	Spearman's ρ
lm(y ~ x)	regression analysis
lm(y ~ f)	one-way analysis of variance
lm(y ~ f1 + f2)	two-way analysis of variance
lm(y ~ f + x)	analysis of covariance
lm(y ~ x1 + x2 + x3)	multiple regression analysis
wilcox.test	one- and two-sample Wilcoxon test
kruskal.test	Kruskal–Wallis test
friedman.test	Friedman's two-way analysis of variance

Table B.10 *Parametric methods for discrete data*

Function	Meaning
binom.test	binomial test (incl.sign test)
prop.test	comparison of proportions
prop.trend.test	test for trend in relative proportions
fisher.test	Fisher's exact test in small tables
chisq.test	chi-square test
glm(y~x1+x2+x3, binomial)	logistic regression

Table B.11 *Model formulas*

Function	Meaning
~	distributed by
+	additive effects
:	interaction
*	main effects + interaction
	(a*b = a + b + a:b)
-1	remove intercept

Table B.12 *Linear and generalized linear models*

Function	Meaning
`lm.out <- lm(y ~ x)`	fit model and save result
`summary(lm.out)`	coefficient, etc.
`anova(lm.out)`	analysis of variance table
`fitted(lm.out)`	fitted values
`resid(lm.out)`	residuals
`predict(lm.out, newdata)`	predictions for new data frame
`glm(y ~ x, binomial)`	logistic regression

Table B.13 *Survival analysis*

Function	Meaning
`S <- Surv(time, ev)`	create survival object
`survfit(S)`	Kaplan–Meier estimate
`plot(survfit(S))`	survival curve
`survdiff(S ~ g)`	log-rank test for equal survival curves
`coxph(S ~ x1 + x2)`	Cox's proportional hazards model

Table B.14 *Graphics*

Function	Meaning
`plot()`	scatterplot and more
`hist()`	histogram
`boxplot()`	box-and-whiskers plot
`stripplot()`	stripplot
`barplot()`	bar diagram
`dotplot()`	dot diagram
`piechart()`	cakes
`interaction.plot()`	interaction plot
`lines()`	lines
`abline()`	line given by intercept and slope
`points()`	points
`segments()`	line segments
`arrows()`	arrows
`axis()`	axis
`box()`	frame around plot
`title()`	title above plot
`text()`	text in plot
`mtext()`	text in margin
`legend()`	list of symbols
`pch`	symbol (*p*lotting *ch*aracter)
`mfrow, mfcol`	several plots on one (*m*ulti*f*rame)
`xlim, ylim`	plot limits
`lty, lwd`	line type/width
`col`	colour
`cex, mex`	character size and line spacing in margins