Chapter 2. Data Visualization and Preparation

- Data presentation and visualization
- Tidy data: preparation and cleaning (`tidyverse`)
- Data manipulation and extraction with `dplyr`
- Error Handling
- Data transformation

To learn more about R techniques in handling data, read and practise along with Parts I & II of Wichham and Grolemund (2017).

Introduction to Data

Data lies at the root of modern technological and methodological advancements.

The goal of data collection is to answer scientific, social or business questions.

Though the question of interest should be specified prior to data collection, the preliminary phase of data analysis typically involves solidifying the question to be as precise as possible, often taking into account the limitations of the available data.

This process can be repeated as many times as desired, as articulated by George Box: data analysis itself is an iterative and sequential process.

A primary step: presenting and visualising data.

Wisdom of Crowds

In 1907 Francis Galton, cousin of Carles Darwin, conducted an experiment in Plymouth: a large ox is on display, contestants guess ‘dressed’ weight of the resulting meat after the ox is slaughtered.

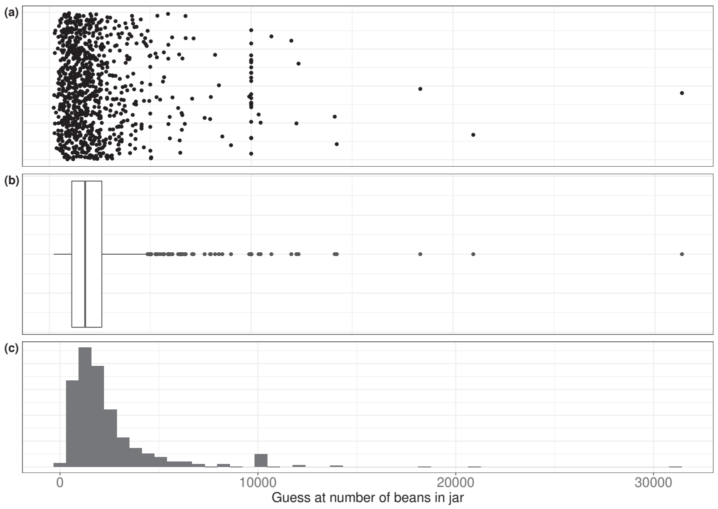
He received 787 answers and chose the middle value (i.e. median) 547kg as the democratic choice.

The dressed weight turned out to be 543kg!

Galton reported the story in a letter in *Nature* with the title ‘Voice of the People’.

James Grime and David Spiegelhalter (Maths/Stats communicators) posted a video of a jar on YouTube, asked anyone watching to guess the number of jelly beans in the jar.

915 responses received, ranging from 219 to 31,337



3 ways to show 915 guesses of the number of jelly beans: (a) Dot-diagram, (b) Box-plot, (c) Histogram.

Dot-diagram: show individual points

Box-plot: highlight 5 summary statistics

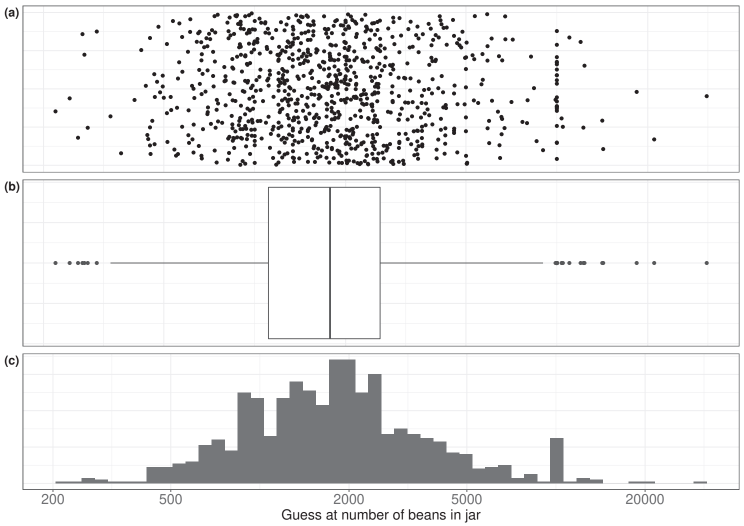
Histogram: show underlying distribution

What have we learned?

Highly skewed: a long tail at right

Preference for round numbers such as 10,000

The figures are not effective as a large space on the right is for a few points only: take a logarithmic transformation



log(guess of the number of jelly beans)

Summary statistics of 915 guesses of the number of jelly beans in a jar. The true number is 1,616.

Mean	Median	Mode
2,408	1,775	10,000
Range	Inter-quartile range	Standard deviation
(219, 31,337)	(1,109, 2,599)	2,422

The way at which this data set is assembled is far from scientific, leading to poor quality in data, containing bizza value 31,337 with 10,000 as the most frequent round number. Nevertheless the median is a good estimate for the truth, unaffected by odd values such as 31,337.

Good enough, Wisdom of Crowds

But mean 2,408 is a poor estimate for the true number!

For log-guesses, mean and median are 7.482 and 7.495 respectively, both are good estimates for the true value 7.388= log(1616).

When to use mean, and when to use median?

3 most commonly used measures for location or average

- Mean: the sum of all data points divided by the number of point
- Median: the middle value when all data points are put in order
- Mode: the most common value

Mean is the average value, Meadian is the value of the average individual.

For a symmetric distribution, mean and median are the same.

Mean is sensitive to outliers, while Median is robust wrt outliers.

Mean should be used when the distribution is about symmetric

Describing the spread/variation of data

Range: sensitive to outliers/extremes

Inter-quartile range (IQR): the range between the 25th and the 75th percentiles, the range of the half data points in the middle, insensitive to extremes

Standard deviation (STD): widely used measure, sensitive to extremes.

Removing a single value 31,337 from the data reduces the STD from 2,422 to 1,398.

Some pitfalls of mean

Nearly everyone has greater than average number (which is around 1.99999) of legs

Mean income has little resemblamce to most people's experience. (If used, leave out people like Bill Gates, Jack Ma.)

UK life tables report that 1% of 63-year-old men die before their 64th birthday.

Mean should not be used to form house price index, as house prices have a long tail on the right due to high-end properties. Be mindful the difference between average house-price and average-house price.

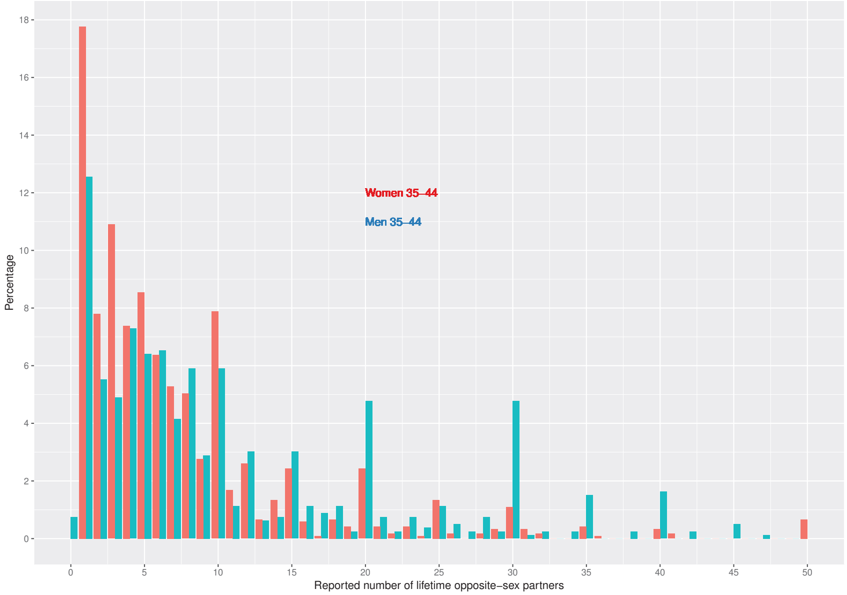
Describing differences between different groups of numbers

National Sexual Attitudes and Lifestyle Survey (Natsal) has been carried out in UK every 10 years since 1990.

The 3rd survey, Natsal-3, was done in 2010 and cost £7 million.

Reported number of opposite-sex sexual partners in lifetime	Men aged 35-44	Women aged 35-44
Mean	14.3	8.5
Median	8	5
Mode	1	1
Range	(0, 500)	(0, 550)
IQR	(4, 18)	(3, 10)
STD	24.2	19.7

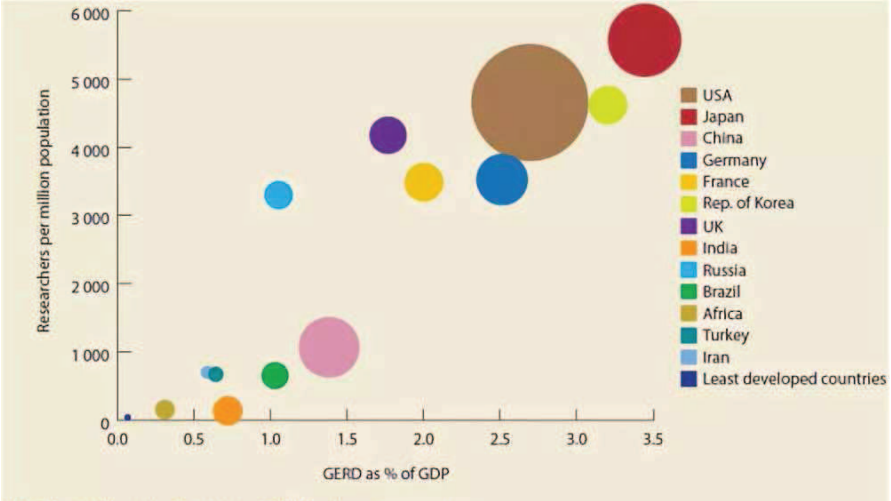
Finding: 1 partner is most common; men report about 60% more partners in both mean and median; mean is much larger than median; range is large and STD is influenced by those a few large values.



The clear use of round numbers for 10 or more partners by both men and women

The tendency for men to report more partners than women.

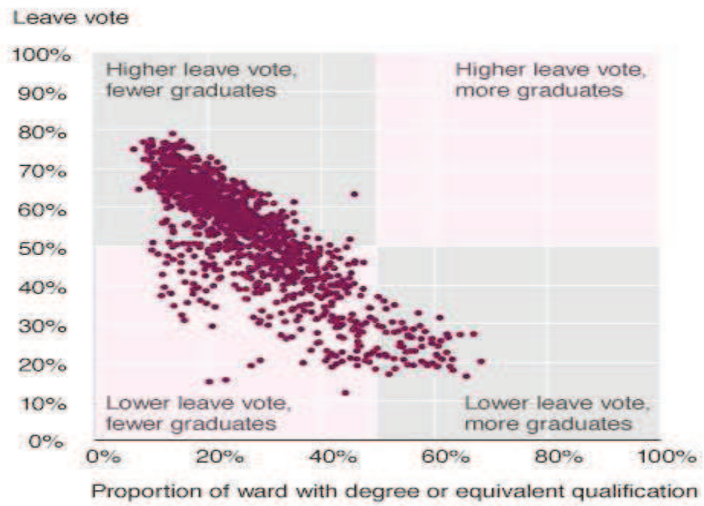
Possible cause: Systematic differences in the way men and women count and report their sexual histories: Men may be likely to overplay their number of partners, or women underplay them, or both



Note: The size of the circle reflects the size of GERD for the country or grouping.

Source: UNU-MERIT based on data from the UNESCO Institute for Statistics and World Bank

Wards with more graduates had lower Leave vote

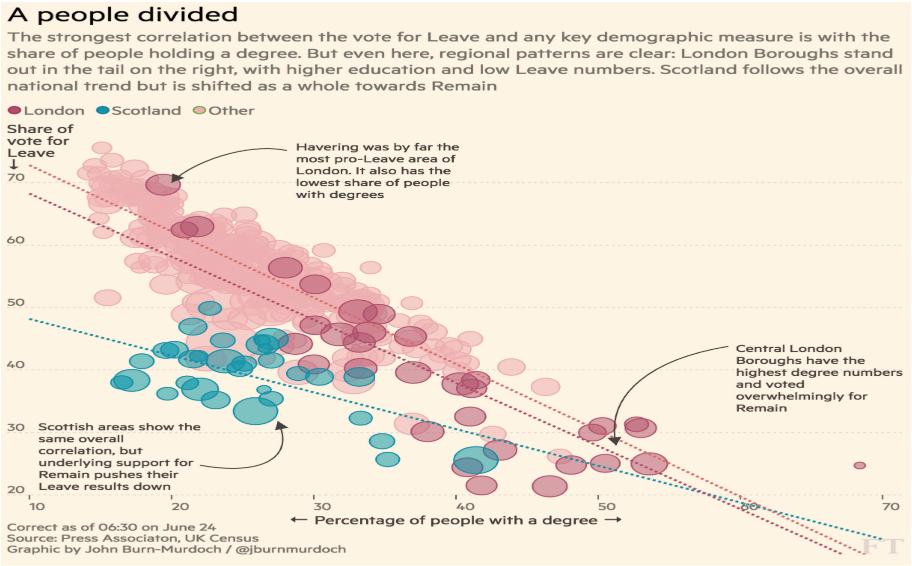


Source: BBC analysis of 1070 local government wards

BBC

Some comments by public:

- It seemed strange to split the x-axis at 50%. Surely this should have been split at the median of the data?
- The only real criticism that comes immediately to attention is the overplotting of the points in the center: that makes it difficult to assess the numbers of points there, which makes the plot a little less useful than it could be
- the graph is misleading in a sense that it purports to show that there are no data points in the quadrant categorically described as high leave vote %, high % of graduates. What is high and low becomes relative to the axis limits, not the actual data. A more objective way to visualise this data would be to set the scatter plot axis limits at the maxmin of the data and then divide the chart into quadrants of an equal area.



A people divided

The strongest correlation between the vote for Leave and any key demographic measure is with the share of people holding a degree. But even here, regional patterns are clear: London Boroughs stand out in the tail on the right, with higher education and low Leave numbers. Scotland follows the overall national trend but is shifted as a whole towards Remain

Havering was by far the most pro-Leave area of London. It also has the lowest share of people with degrees

Central London Boroughs have the highest degree numbers and voted overwhelmingly for Remain

Scottish areas show the same overall correlation, but underlying support for Remain pushes their Leave results down

Correct as of 06:30 on June 24
Source: Press Associaton, UK Census
Graphic by John Burn-Murdoch / @jburnmurdoch

● London
● Scotland
○ Other
Share of vote for Leave ↓
70
60
50
40
30
20
10
20
30
40
50
60
70
← Percentage of people with a degree →
FT

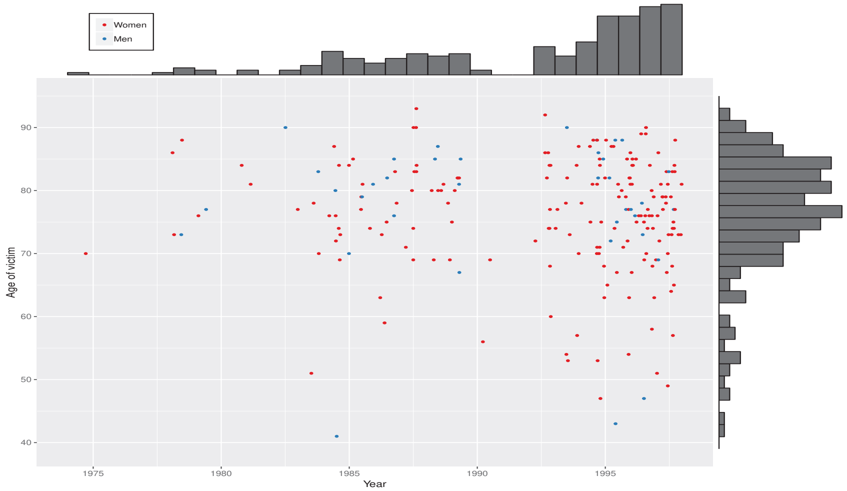
- I view this as a clear informative plot. The message is immediately apparent and is not misleading. The BBC has plotted the actual data points. They have not manipulated the x or y axes. The annotation on the plot is correct and not over-stated. They have not added spurious trend lines or any other unnecessary interpretation. Compared to most data figures presented in the media, this plot is excellent – it is quite a good example of letting the data speak for itself. You can find some ways to improve the plot, but simple is usually best.

Harold Shipman was Britain's most prolific convicted murderer: between 1975 and 1998 he injected at least 215 of his mostly elderly patients with a massive opiate overdose.

He finally made the mistake of forging the will of one of his victims so as to leave him some money: her daughter was a solicitor, suspicions were aroused, and forensic analysis of his computer showed he had been retrospectively changing patient records to make his victims appear sicker than they really were.

He was well known as an enthusiastic early adopter of technology, but he was not tech-savvy enough to realize that every change he made was time-stamped.

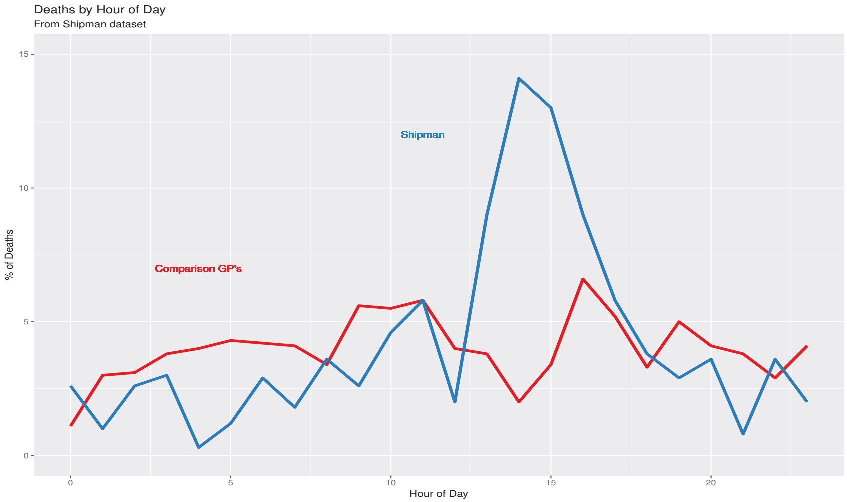
Of his patients who had not been cremated, 15 were exhumed and lethal levels of diamorphine, the medical form of heroin, were found in their bodies. Shipman was subsequently tried for 15 murders in 1999, but chose not to offer any defence and never uttered a word at his trial. He was found guilty and jailed for life.



A scatter-plot showing the age and the year of death of Shipman's 215 confirmed victims. Bar-charts at the top and on the right to reveal the pattern of victims' ages and the pattern of years in which he committed murders

A public inquiry was set up to determine what crimes he might have committed apart from those for which he had been tried, and whether he could have been caught earlier. A number of statistician were called to give evidence at the public inquiry, which concluded that he had definitely murdered 215 of his patients, and possibly 45 more.

His reasons for committing these murders have never been explained: he gave no evidence at his trial, never spoke about his misdeeds to anyone, including his family, and committed suicide in prison, conveniently just in time for his wife to collect his pension.



The time at which Harold Shipman's patients died, compared to the times at which patients of other local general practitioners died. The pattern does not require sophisticated statistical analysis.

Data visualization: to help people understand the significance of data by placing it in a visual context: Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier when displayed visually.

An effective visual display is often a powerful tool for data analysis.

Hans Rosling (Gapminder inventor): find catchy ways to illustrate statistics, and

> Having the data is not enough, I have to show it in the ways people both enjoy and understand.

Watch "The Joy of Statistics" on YouTUbe

It is common in this modern age that collect data first, and ask questions later, as data collection becomes easy, cheap and automated especially for big companies such as Google, Facebook, and also for supermarkets, banks, hospitals etc.

Nevertheless collecting relevant and high quality data which are readily usable for answering the questions of interest is not easy, or difficult, as data comes in many shapes and sizes, some data are messy, confusing with many missing values.

Data are often not in a ‘tidy’ format which can be readily employed for analysis.

Tidy data format is in the form of matrix/spreadsheet: (i) each variable is a column, (ii) each observation is a row, and (iii) each value is a cell.

To extract the relevant information from data is challenging, hence *Data Science* becomes one of the most important scientific disciplines in this information era.

Data Preparation and Cleaning – necessary and time-consuming!

Important but how?

`tidyverse` is a collection of R packages for data science, including core packages:

- `tidyr`: provides functions `gather`, `spread`, `separate`, `unite` for tidying data
- `dplyr`: provides functions `filter`, `arrange`, `select`, `mutate`, `summarise` etc for data manipulation/transformation
- `ggplot2`: for creating graphics and data visualization
- `readr`: contains functions `read_csv`, `read_csv2`, `read_tsv`, `read_delim` for importing data fast and friendly, and `parse_*` for parsing various types of data
- `tibble`: a modern re-imagining of the data frame which is lazy and surly: do less and complain more; leading to cleaner and more expressive codes

Installation: `install.packages(tidyverse)`

References: Grolemund and Wickham (2017), and many online sources such as

`https://www.tidyverse.org/`

1. Tidying the data structure

Put data into a tidy format (e.g. a dataframe/tibble in R):

Each variable/attribute forms a column

Each observation forms a row

Each value has its own cell

As an illustration, consider the *Communities and Crime* data set downloaded from UC-Irvine Machine Learning Repository (`archive.ics.uci.edu/ml/`)

Source: socio-economic data for communities in the US obtained from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. The data consist of 2,215 communities, and each has 147 attributes.

Question to ask: what features of a community affect the violent crime rates

Open an RStudio session, specify a working directory, and put in the working directory 3 files: `communityCrimeData.csv`, `communityCrimeInfo.pdf`, `communityCrimeAttributes.txt`.

There are two data files:

communityCrimeData.txt – a table of 2215×147 entries
communityCrimeAttributes.txt – names and definitions of 147 attributes

More detailed information can be found at

archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized#

or in the file communityCrimeInfo.pdf.

To import data,

```
> CC=read.csv("communityCrimeData.csv", header=F)
> dim(CC)
[1] 2215  147
```

read.csv is a special case of read.table for reading ‘comma separated value’ files. Note read_csv in readr could be about 10 times faster in

```
> ccAttr=read.delim("communityCrimeAttributes.txt", header=F,
           sep=" ", skip=2)
> dim(ccAttr) # show size 147 x 23
> names(CC)=ccAttr[,1] # assign column names to file CC
> View(CC)
```

2. Data manipulation and extraction

Now we manipulate the dataset using the package dplyr

```
> library(dplyr)  # upload the package to the current session
> tbl_df(CC)
```

tbl_df displays only the first 10 rows and the number of columns fit to the display.

To extract the communities with population ≥ 800K

```
> bigPopul=filter(CC, population>=800000)
                      # slect rows with population >= 800K
```

importing big data files. However data are imported as tibble instead of standard data.frame.

To view the whole data set, View(CC).

We also need to attach the attribute names to each columns. The information on the attributes is in another file

communityCrimeAttributes.txt

which has more than 2215 lines, and each line has different number of entries (separated by space). The names of attributes are the 1st entry in each line. The file looks like

```
Attribute Information
(125 predictive, 4 non-predictive, 18 potential goal)
communityname  Community name - not predictive - for information only (string)
state  US state (by 2 letter postal abbreviation)(nominal)
countyCode  numeric code for county - not predictive, and many missing values (numeric)
communityCode  numeric code for community - not predictive and many missing values (numeric)
fold  fold number for non-random 10 fold cross validation, potentially useful for debugging, paired tests -
population  population for community (numeric - expected to be integer)
householdsize  mean people per household (numeric - decimal)
... ...
```

We need to input this unstructured file into R

```
> dim(bigPopul)
[1]  10 147
> bigPopul[,1]
 [1] NewYorkcity      Philadelphiacity LosAngelescity   Dallascity
 [5] Detroitcity      Houstoncity      SanDiegocity     SanAntoniocity
 [9] Chicagocity      Phoenixcity
```

murdPerPop records the number of murders per 100K population

```
> attach(CC) # make the columns recognizable in R
> summary(murdPerPop)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   2.170   5.859   8.365  91.090
> tt=filter(CC, population>=800000, murdPerPop>8.365)
> tt[,1]
 [1] NewYorkcity      Philadelphiacity LosAngelescity   Dallascity
 [5] Detroitcity      Houstoncity      SanDiegocity     SanAntoniocity
 [9] Chicagocity      Phoenixcity
```

```
> tt=filter(CC, population>=800000, murdPerPop>20)
> tt[,1]
[1] NewYorkcity      Philadelphiacity LosAngelescity   Dallascity
[5] Detroitcity      Houstoncity      SanAntoniocity   Chicagocity
> tt=filter(CC, population>=800000, murdPerPop>30)
> tt[,1]
[1] LosAngelescity Dallascity     Detroitcity    Chicagocity
```

To extract columns of interest for those 10 cities with big population:

```
> big10C = select(bigPopul, communityname, PctKidsBornNeverMar, racepctblack, pctWPubAsst,
            TotalPctDiv, PctUnemployed, ViolentCrimesPerPop)
> big10C
   communityname PctBornNeverMar racepctblack pctWPubAsst TotalPctDiv PctUnemployed VioCrimePerPop
     NewYorkcity           10.50        28.71       13.12       11.77          8.98        2097.71
Philadelphiacity           11.53        39.86       13.98       12.53          9.62         1279.6
  LosAngelescity            9.32        13.99       10.68       12.26          8.34        2414.77
      Dallascity            7.28        29.50        5.77       15.58          7.43        1744.19
     Detroitcity           16.59        75.67       26.14       17.88         19.67              ?
     Houstoncity            6.91        28.09        7.06       15.07          8.18              ?
    SanDiegocity            4.50         9.39        8.81       12.80          5.72        1162.84
  SanAntoniocity            3.48         7.04        9.58       14.01          8.92         672.57
     Chicagocity           12.08        39.07       14.36       12.99         11.32              ?
     Phoenixcity            4.43         5.19        5.79       14.74          6.64        1097.07
```

Note. `names(CC)` prints out all column names.

5 attributes/variables: percentage of children born to unmarried parents, black population, population receiving public assistance income, divorced, and unemployed, respectively

Response: no. of violent crimes (i.e. murder, rape, robbery, and assault) per 100K population

Detroit has the highest proportions for all the 5 variables

Chicago is the 2nd highest in children with unmarried parents, public assistance in income, unemployment, the 3rd in the other two variables

Violent crime rates for Detroit and Chicago are missing: Controversy concerning the reporting of rapes resulted in missing values for the number of rapes, and subsequently for violent crime rate.

Further investigation: among all cities whose population is at least 500,000 Chicago and Detroit had the second and third highest assault rates respectively, and Detroit had the second highest murder rate, and Chicago had the seventh highest

Not entirely implausible to believe that Chicago and Detroit might have higher violent crime rates than most other communities

These missing values are informative in the sense that the values that are missing may correspond to the communities with higher crime rates.

Therefore, subsequent estimates of overall violent crime rates (e.g. over states or even country-wide) may be biased downwards (underestimated).

`dplyr` can also perform more sophisticated data manipulations and cleaning. For example, to rearrange in the rows according to the ascending order of `population`:

```
> tt=arrange(CC, population)
> tt[2215,1]
[1] NewYorkcity
```

See also

`ran.rstudio.com/web/packages/dplyr/vignettes/introduction.html`

3. **Identifying inconsistencies**

After the data is put in a tidy format, the next step is to ensure that the data makes sense: do the range of values for each variable match what you expect? are there any outliers? how many missing values?

A useful function in R: `summary` – list a summary for each column:

For numerical variables: min, 1st & 3rd quartiles, median, mean, max

For categorical variables: list the frequency at each level

Unmarr. parents	Black	Public assist.	Divorced	Unemp.	Violent crimes
Min 0.000	Min 0.000	Min 0.180	Min 2.830	Min 1.320	Min 0.0
1st-Q 1.070	1st-Q 0.860	1st-Q 3.270	1st-Q 8.575	1st-Q 4.045	1st-Q 161.7
Med 2.040	Med 2.870	Med 5.610	Med 10.900	Med 5.450	Med 374.1
Mean 3.115	Mean 9.335	Mean 6.801	Mean 10.813	Mean 6.045	Mean 589.1
3rd-Q 3.910	3rd-Q 11.145	3rd-Q 9.105	3rd-Q 12.985	3rd-Q 7.440	3rd-Q 794.4
Max 27.350	Max 96.670	Max 44.820	Max 22.230	Max 31.230	Max 4877.1
					NA's 221

All communities with population greater than 800K have much higher violent crime rates than the median 374.1

The distribution for violent crime rates is skewed towards the right as mean $>>$ median; indicating some excessive high crime rates.

Normalization by transformation: logarithmic or square-root transformations are the most frequently used in practice. Both 'squeeze' data towards the mean, to make data look more normally distributed.

Many statistical models and methods are designed for normal data, or work better for normal data.

Note. (i) Linear regression models do not rely on the normality assumption. But when errors are normally distributed, OLS is MLE and the t-tests are exact instead of being asymptotically approximate

(ii) if $y = x_1 x_2$, $\log y = \log x_1 + \log x_2$. Be mindful on the interpretation of the models for $\log y$!

(iii) Box-Cox transformation

(iv) log-transformation may transform a skewed distribution into a normal-like one

4. **Normalization**: convert different quantities to a common scale for the purpose of comparability, numerical stabilization, or outlier detection

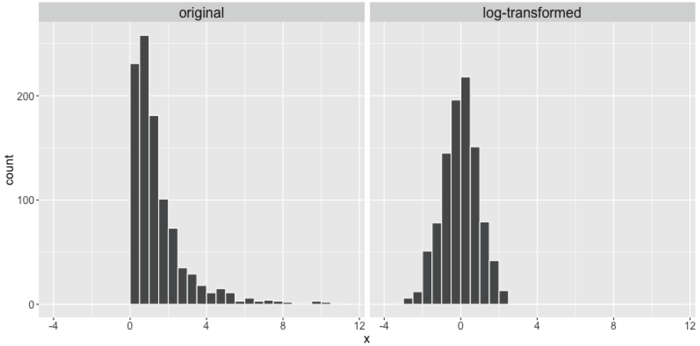
A source of confusion: Normalize or not? – the decision will be a judgment call.

- *Standardization*: $x_{new} = (x - \bar{x})/\text{STD}(x)$ Then $\bar{x}_{new} = 0$ and $\text{STD}(x_{new}) = 1$

- *Making data between 0 and 1*: $x_{new} = (x - x_{\min})/(x_{\max} - x_{\min})$

Normalization is usually applied to each variables (columns) not to individuals (rows).

To standardize only if it is necessary: Standardization tends to remove a lot of the natural dependence structures that exist within the data. Further, if the variables had meaningful units such that interpretations after scaling become less transparent, we might be enticed to leave the data alone.



5. **Errors and Outliers**

Most real data are subject to errors.

Errors can be divided into two types: Random and Deterministic.

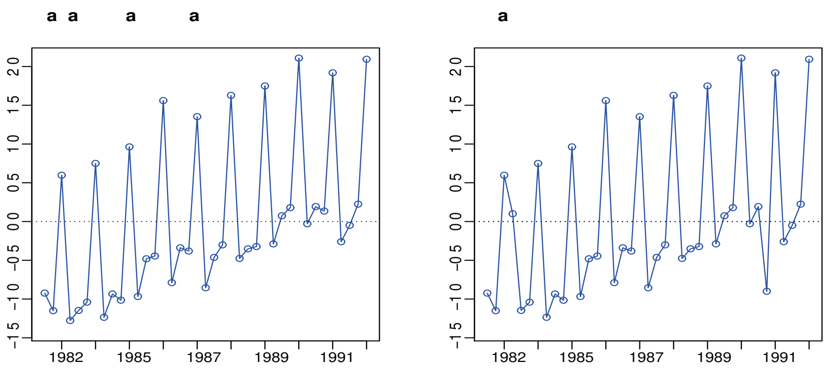
Deterministic errors can be corrected, if the source of the errors can be correctly identified.

Random errors cannot be corrected from data, though their impact on data mining may be controlled or eliminated sometimes.

Examples of random (stochastic) errors: measurement or transmission errors – typically additive

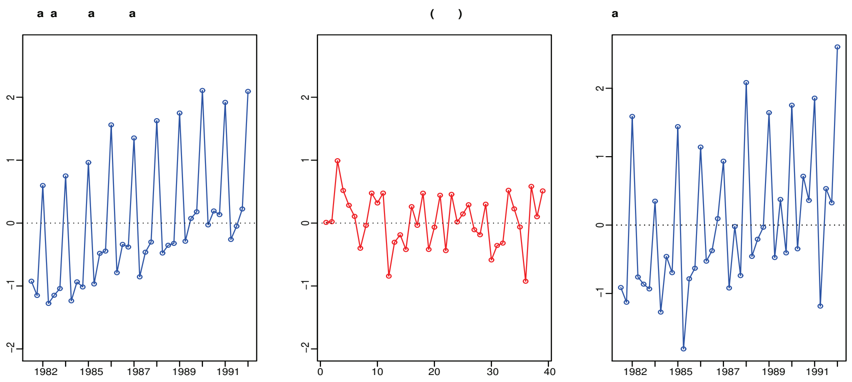
Examples of deterministic errors: wrong formulas for computing data, wrong calibration/scaling, sensor drift, different meanings of '.' or ',' on different systems (i.e. 1.234 and 1,234 may denote the same number)

Outliers: individual data points showing-off odd behaviour (e.g. too large or too small, or off phase) in comparison with the majority of data, may be either random or deterministic.



Standardization: $x \to (x - \text{mean})/\text{std}$

mean = 14623.17,   std=1298.926



Out-range outliers detection

Rule of thumb: If $|x - \text{mean}|/\text{std} > 2$, x is regarded as outlier.

Caution: unusual but correct values should not be removed/modified. Such as financial crisis, earthquake.

Distinguish exotic but valuable data from erroneous data: using domain knowledge!

For the *Communities and Crime Data*, the population (i.e. Column 6 in the dataset) in New York City is 7323000, which is much greater than

$$\text{mean} + 2\,\text{std} = 53120 + 2 * 204620.3 = 462360.6$$

Of course this is not an outlier, even the 2nd largest population is Los Angeles 3485398. The summary of this variable is

```
Min.    1st Qu.  Median   Mean   3rd Qu.    Max.
10000    14370   22790   53120   43020    7323000
```

Error Handling: The identified outliers and missing values can be handled in various ways as follows.

1. *Invalidity list*: the indices of the invalid data are stored in a separate list that is checked in each data processing step.

2. *Invalid values*: replace invalid values directly by `NA` (not available).

3. Remove outliers.

4. *Correction or estimation*: invalid values can be estimated by one of the following methods.
   - replacing by the mean, median, minimum, or maximum of the valid data

- nearest neighbour correction – applicable for correcting one outlier component of vector data: replacing invalid x_k^i by x_j^i, where $\mathbf{x}_k = (x_k^1, \cdots, x_k^p)'$, and

$$\|\mathbf{x}_j - \mathbf{x}_k\|_{-i} = \min_{\ell} \|\mathbf{x}_\ell - \mathbf{x}_k\|_{-i}.$$

- *linear interpolation for time series*:

$$x_t = (x_{t-1} + x_{t+1})/2, \qquad \text{(regularly sampled data)}$$

$$x_k = \frac{x_{k-1}(t_{k+1} - t_k) + x_{k+1}(t_k - t_{k-1})}{t_{k+1} - t_{k-1}}, \qquad \text{(irregularly sampled data)}$$

- *Nonlinear interpolation* using, eg. splines.

- *Filtering*: remove outliers, more often used to remove noise from time series data

- *Model-based estimation* by, eg. regression.

6. **Data Merging**: merge together relevant data from different data sets, files, database or systems.

- identify a clearly defined rule for merging

- identify relevant IT tools to merge data

Data cleaning can be a complex and challenging process; requiring open-minded, critical thinking and relevant IT skills to manipulate complex and large datasets. The goal is first to identify all suspicious data points, and then to treat (i.e. correct, remove or down-weight) them. The former is relatively easy than the latter for which *common sense, subject knowledge, and the knowledge on data collection and recording* are brought to bear.

Data cleaning ≠ Oiling data!

Descriptive Data Analysis in R (II)

1.5 Writing functions in R

For some repeated task, it is convenient to define a function in R. We illustrate this idea by an example.

Consider the famous 'Birthday Coincidences' problem: *In a class of k students, what is the probability that at least two students have the same birthday*?

Let us make some assumptions to simplify the problems:
(i) only 365 days in every year,
(ii) every day is equally likely to be a birthday,
(iii) students' birthdays are independent with each other.

With k people, the total possibilities is $(365)^k$.

Consider the complementary event: all k birthdays are different. The total such possibility is

$$365 \times 364 \times 363 \times \cdots \times (365 - k + 1) = \frac{365!}{(365-k)!}$$

So the probability that at least two students have the same birthday is

$$p(k) = 1 - \frac{365!}{(365-k)!(365)^k}.$$

We may use R to compute $p(k)$. Unfortunately factorials are often too large, e.g. $52! = 8.065525e + 67$, and often cause overflow in computer. We adopt the alternative formula

$$p(k) = 1 - \exp\{\log(365!) - \log((365-k)!) - k\log(365)\}.$$

We define a R-function pBirthday to perform this calculation for different k.

```
> pBirthday <- function(k)
```

```
+ 1 - exp(lfactorial(365) - lfactorial(365-k) - k*log(365))
         # lfactorial(n) returns log(n!)
> pBirthday(100)
[1] 0.9999997   # probability with a class of 100 students
> x <- c(20, 30, 40, 50, 60)
> pBirthday(x)
[1] 0.4114384 0.7063162 0.8912318 0.9703736 0.9941227
```

With 20 students in class, the probability of having overlapping birthdays is about 0.41. But with 60 students, the probability is almost 1, i.e. *it is almost always true that at least 2 out of 60 students have the same birthday.*

Note. The expression in a function may have several lines. In this case the expression is enclosed in curly braces { }, and the final line determines the return value.

Instead of writing R functions directly in a console, RStudio offers the top left window for scripts (or data) editing. To create a new file, use the `File -> New File` menu. To open an existing file, use `File -> Open File`.

RStudio's script editor includes a variety of productivity enhancing features including syntax highlighting, code completion, multiple-file editing, and find/replace. To execute commands in the editing window, highlight them and click on `run`-button.

Another Example — **The capture and recapture problem**

To estimate the number of whitefish in a lake, 50 whitefish are caught, tagged and returned to the lake. Some time later another 50 are caught and only 3 are tagged ones. Find a reasonable estimate for the number of whitefish in the lake.

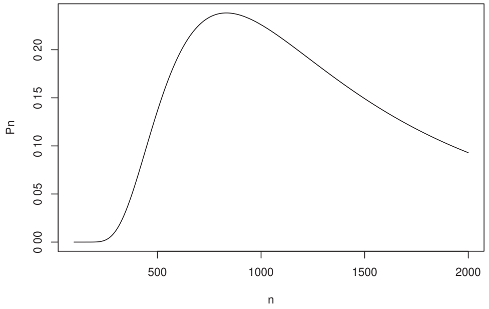
Suppose there are n whitefish in the lake. Catching 50 fish can be done in $\binom{n}{50} = \frac{n!}{50!(n-50)!}$ ways, while catching 3 tagged ones and 47 untagged can be done in $\binom{50}{3}\binom{n-50}{47}$ ways. Therefore the probability for the latter event to occur is

$$P_n = \binom{50}{3}\binom{n-50}{47}\Big/\binom{n}{50}.$$

Therefore, a reasonable estimate for n should be the value at which P_n obtains its maximum. We use R to compute P_n and to find the estimate.

```
> Pn <- function(n) {
+      tmp <- choose(50,3)*choose(n-50,47)
+      tmp/choose(n,50)
+ }           # Definition for function Pn ends here
> n <- 97:2000    # as there are at least 97 fish in the lake
> plot(n, Pn(n), type='l')
```

It produces the plot of P_n against n:



To find the maximum:

```
> m <- max(Pn(n)); m
[1] 0.2382917
> n[Pn(n)==m]
[1] 833
```

Hence the estimated number of fish in the lake is 833.

1.6 Control structure: loops and conditionals

An `if` statement has the form

```
if (condition) expression1 else expression2
```

It executes 'expression1' if 'condition' is true, and 'expression2' otherwise.
When 'condition' contains several lines, they should be enclosed in curly braces { }. The same applies to expressions.

The above statement can be compactly written in the form

```
ifelse(condition, expression1, expression2)
```

When the else-part is not present:

```
if (condition) expression
```

It executes 'expression' if 'condition' is true, and does nothing otherwise.

A `for` loop allows a statement to be iterated as a variables assumes values in a specified sequence. It has the form:

```
for(variable in sequence) statement
```

A `while` loop does not use an explicit loop variable:

```
while (condition) expression
```

It repeats 'expression' as long as 'condition' holds. This makes it differently from the "if-statement" above.

We illustrate those control commands by examining a simple 'doubling' strategy in gambling.

You go to a casino to play a simple 0-1 game: you bet x dollars and flip a coin. You win $2x$ dollars and keep your bet if 'Head', and lose x dollars if 'Tail'. You start 1 dollar in first game, and double your bet in each new games, i.e. you bet 2^{i-1} dollars in the i-th game, $i = 1, 2, \cdots$.

With this strategy, once you win, say, at the $(k+1)$-th game, you will recover all your losses in your previous games plus a profit of $2^k + 1$ dollars, as

$$2 \times 2^k > \sum_{i=1}^{k} 2^{i-1} = 2^k - 1.$$

Hence as long as (i) the probability p of the occurrence of 'Head' is positive (no matter how small), and (ii) you have enough capital to keep you in the games, you may win handsomely at the end — is it really true?

Condition (ii) is not trivial, as the maximum loss in 20 games is $2^{20} - 1 = 1,048,575$ dollars!

Plan A: Suppose you could afford to lose maximum n games and, therefore, decide to play n games. We define the R-function `nGames` below to simulate your final earning/loss (after n games).

```
nGames <- function(n,p) {
      # n is the No. of games to play
      # p is the prob of winning each game
x <- 0 # earning after each game
for(i in 1:n)  ifelse(runif(1)<p, x <- x+2^i, x <- x-2^(i-1))
      # runif(1) returns a random number from uniform dist on (0, 1)
x     # print out your final earning/loss
}
```

To play $n = 20$ games with $p = 0.1$:

```
> nGames(20, 0.1)
[1] -999411
> nGames(20, 0.1)
[1] -1048575
> nGames(20, 0.1)
[1] 524289
```

```
> nGames(20, 0.1)
[1] -655263
> nGames(20, 0.1)
[1] -1016895
```

We repeated the experience 5 times above, with 5 different results.

One way to assess this gameplan is to repeat a large number of times and look at the average earning/loss:

```
> x = vector(length=5000)
> for(i in 1:5000) x[i] <- nGames(20, 0.1)
> mean(x)
[1] -733915
```

In fact, this mean -733915 is stable measure reflecting the average loss of this gameplan.

Plan B: Play the maximum n, but quit as soon as winning one game. The R-function `winStop` simulates the earning/loss.

```
winStop <- function(n,p) {
      # n -- maximum No. of games, p -- prob of winning each game
i <- 1
ifelse((runif(1)<p), x<- 2, x<- -1)  # play 1st game
while((x<0)&(i<n)){ i <- i+1      # i records the no. of games played
                 ifelse(runif(1)<p, x <- x+2^i, x <- x-2^(i-1))
               }
x
}
```

Set $n = 20$, $p = 0.1$, we repeat the experience a few times:

```
>winStop(20, 0.1)
[1] 2
```

```
> winStop(20, 0.1)
[1] 17
> winStop(20, 0.1)
[1] 129
> winStop(20, 0.1)
[1] -1048575
> winStop(20, 0.1)
[1] 16385
```

To assess the gameplan:

```
> x<- 1:5000
> for(i in 1:5000) x[i] <- winStop(20, 0.1)
> mean(x)
[1] -112672.9   # This indicates "Plan B" is better than "Plan A"
> for(i in 1:5000) x[i] <- winStop(80, 0.1)
          # the maximum no. of games is 80 now
```

```
> mean(x)
[1] -7.22886e+20
> for(i in 1:5000) x[i] <- winStop(90, 0.1)
          # the maximum no. of games is 90 now
> mean(x)
3.790896e+18
```

With p as small as 0.1, you need a huge capital in order to play about 90 games to generate the positive returns in average.

The best and the most effective way to learn R: use it!

Hands-on experience is the most illuminating.

R Markdown: A powerful authoring framwork for

- creating, executing and saving R codes, and
- generating a quality report, in HTML, pdf or word, containing codes, R-outputs (graphics) and text. It could be a normal document, or a set of slides for presentation

To install, `install.packages("rmarkdown")`

R Markdown Cheatsheet:
www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf

R Markdown Reference Guide:
www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf

An R Markdown file is a plain text file with the extension `.Rmd`. It contains three types of content:

- An optional header sandwiched by two sets of `---`
- R code chunks sandwiched by “‘{r } and “‘
- text chunks with simple text formatting (see R Markdown Reference Guide)

To start a session with R Markdown, on the top left panel of RStudio, click on

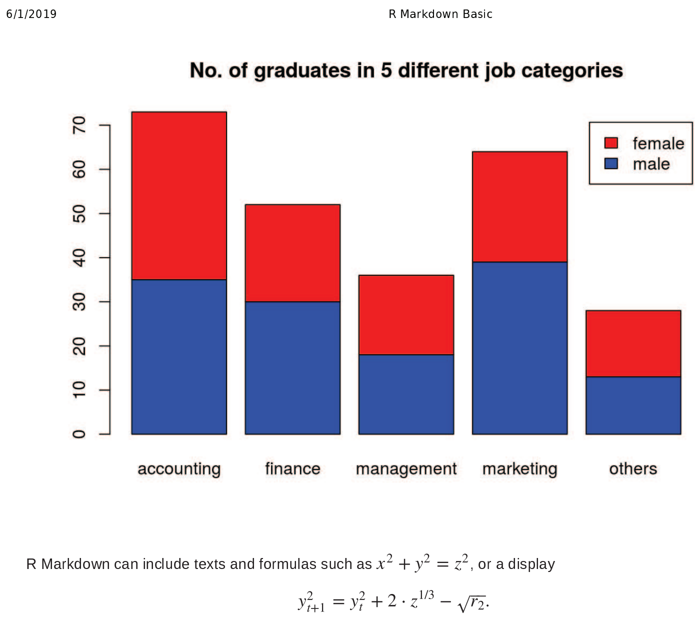
`File` → `New File` → `R Markdown ...`

This creates a template markdown file which one can edit. For example, the file below illustrates some elementary and useful functions.

```
---
title: "R Markdown Basic"
author: "Qiwei Yao"
date: "May 31, 2019"
output: html_document
---

## On this document
This is an R Markdown document illustrating its rudimentary function
‘‘‘{r }
setwd("~/teaching/bigData/data")
jobs <- read.table("Jobs.txt", header=T, row.names=1)
t <-table(jobs[,2], jobs[,1], deparse.level=2)  # store table in t
t
100*t[1,]/sum(t[1,])
100*t[2,]/sum(t[2,])
‘‘‘
```

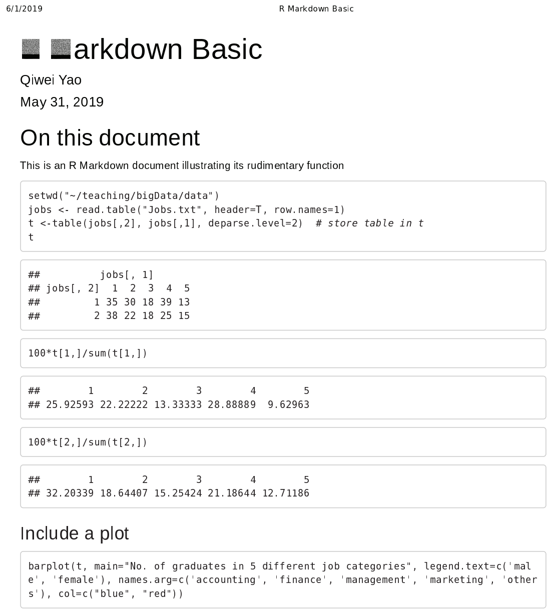
```
### Include a plot
‘‘‘{r }
barplot(t, main="No. of graduates in 5 different job categories",
    legend.text=c(’male’, ’female’), names.arg=c(’accounting’,
    ’finance’, ’management’, ’marketing’, ’others’),
    col=c("blue", "red"))
‘‘‘

R Markdown can include texts and formulas such as $x^2 + y^2 = z^2$,
or a display

$$y^2_{t+1}= y_t^2 + 2\cdot z^{1/3} - \sqrt{r_2}.$$
```

Then remember to save the file first with the last name `.Rmd`, then click on `Knit`-button to produce the html-document below.



R Markdown can include texts and formulas such as $x^2 + y^2 = z^2$, or a display

$$y^2_{t+1} = y_t^2 + 2 \cdot z^{1/3} - \sqrt{r_2}.$$



▒ ▒arkdown Basic

Qiwei Yao

May 31, 2019

On this document

This is an R Markdown document illustrating its rudimentary function

```
setwd("~/teaching/bigData/data")
jobs <- read.table("Jobs.txt", header=T, row.names=1)
t <-table(jobs[,2], jobs[,1], deparse.level=2)  # store table in t
t
```

```
##          jobs[, 1]
## jobs[, 2]  1  2  3  4  5
##         1 35 30 18 39 13
##         2 38 22 18 25 15
```

```
100*t[1,]/sum(t[1,])
```

```
##        1        2        3        4        5
## 25.92593 22.22222 13.33333 28.88889  9.62963
```

```
100*t[2,]/sum(t[2,])
```

```
##        1        2        3        4        5
## 32.20339 18.64407 15.25424 21.18644 12.71186
```

Include a plot

```
barplot(t, main="No. of graduates in 5 different job categories", legend.text=c('mal
e', 'female'), names.arg=c('accounting', 'finance', 'management', 'marketing', 'other
s'), col=c("blue", "red"))
```