Introduction to Quantitative Science Seminar
Week 1

ricardo.mellado.19@ucl.ac.uk

Ricardo Mellado Labbe

Brief recap (1)

-**Statistics** is the art and science of collecting, presenting and analyzing data to answer an investigative question. Its ultimate goal is translating data into knowledge and an understanding of the world around us.

Example: Using a Survey to investigate People Beliefs

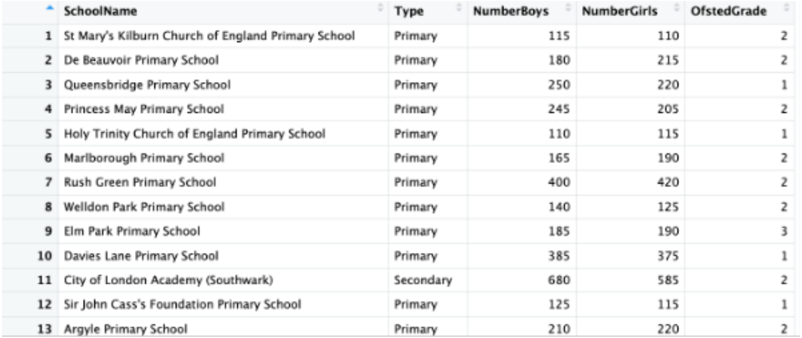
The General Social Survey (GSS) is a survey of a few thousand adult Americans provides data about opinions and behaviors of the American public (**"Would you be willing to pay higher prices in order to protect the environment?" "How much TV do you watch per day**)

Elements of Statistical Analysis

a) **Design:** Stating the goal and/or statistical question of interesting and planning how to obtain data that will adress them
b) **Description:** Summarizing and analyzing the data that are obtained
c) **Inference:** Making decisions and predictions based on the data for answering the statistical question

Brief recap (2)

When statiscians speak of data, they typically refer to the type of data that it is organized in a spreadsheet in rows and columns, forming a grid of cells. **Example: All schools in London.csv**



	SchoolName	Type	NumberBoys	NumberGirls	OfstedGrade
1	St Mary's Kilburn Church of England Primary School	Primary	115	110	2
2	De Beauvoir Primary School	Primary	180	215	2
3	Queensbridge Primary School	Primary	250	220	1
4	Princess May Primary School	Primary	245	205	2
5	Holy Trinity Church of England Primary School	Primary	110	115	1
6	Marlborough Primary School	Primary	165	190	2
7	Rush Green Primary School	Primary	400	420	2
8	Welldon Park Primary School	Primary	140	125	2
9	Elm Park Primary School	Primary	185	190	3
10	Davies Lane Primary School	Primary	385	375	1
11	City of London Academy (Southwark)	Secondary	680	585	2
12	Sir John Cass's Foundation Primary School	Primary	125	115	1
13	Argyle Primary School	Primary	210	220	2

- An **observation** is the information collected from a particular individual or entity in the study (e.g. Schools). We refer to an observation by the row number in the dataframe
- A **variable** captures the values of a changing characteristic for the multiple individuals or entities in the study

Brief recap (3) - Creating Objects and using Functions in R

In order to manipulate and analyze data, we need to load and store datasets. R stores information in what are known as objects. To create and object in R we use the **assignment operator <-**

object_name <- object_contents (e.g.the data we want to store)

Example: four <- 1 + 3

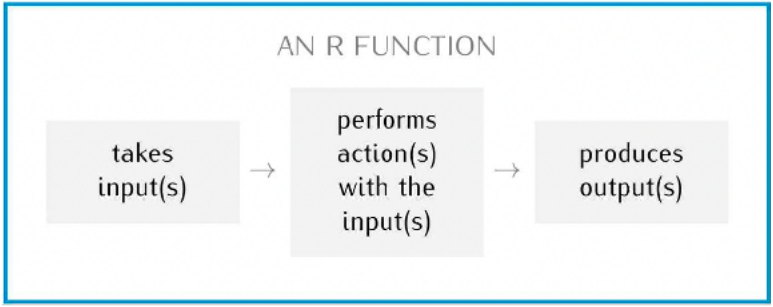
four

4

Example 2

school_data <- read.csv("All Schools in London.csv")

Brief recap (3) - Creating Objects and using Functions in R



AN R FUNCTION

takes input(s) → performs action(s) with the input(s) → produces output(s)

Brief recap (3) - Creating Objects and using Functions in R

We use R to interact with data, which requires using functions. **Example: request R to calculate the square root of 4**

```
sqrt(4)
```

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

- The name of the function is always followed by parentheses: **function_name()**
- Inside the parentheses, we specify the inputs to be used by the function, which we refer to as arguments: **function_name(argument)**
- Sometimes we want to do certain specifications to the function. For this we use an optional argument: **function_name(argument, optional_argument_name = optional_argument)**

Brief recap (4) - Accesing variables inside dataframes

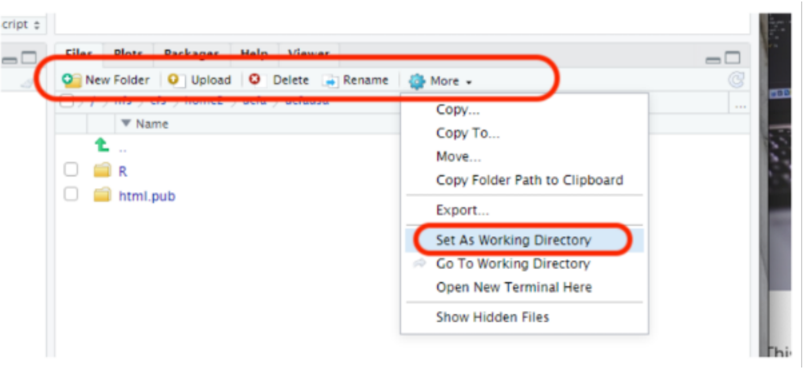
To access the values of a single variable in a dataframe , we use the **$** character. - To its left, we specify the name of the object where the dataframe is stored (without quotes) - To its right, we specify the name of the variable (without quotes)

```
school_aux$NumberBoys
```

```
##  [1] 120 340 975 425 120  85  75 175 250 210
```

Before start(. . .)

Make sure to upload data to RStudio Server and importing data into the environment



Seminar tasks and questions

Seminar Task : Use the seminar data set All Schools in London.csv and import it into RStudio. Perform the following tasks:

1. Create a column called TotalStudents containing the total number of students for each school
2. Create a column called PercentGirls which contains the estimated percentage of girls attending each school.
3. Use the tapply() function to calculate the total number of student by subgroup or type of school

Seminar Questions

1. What are the names of schools that have the highest and lowest number of students? **[HINT: use max(), min() and View() functions]**
2. Generate a fully labeled `barplot()` and describe the distribution of schools by OFSTED scores.
3. Explore the frequency of schools in London based on the total number of students. Make a frequency table in R by creating the groups based on total students per school - starting from 0 to 2200, use the interval of 200 students to generate the groups as 0-200, 201-400, 401-600, . . . , 2001-2200 and hence the number of schools falling into these groupings with these numbers of students
4. Create a histogram.
5. Create a cumulative frequency plot.
6. Create a relative cumulative frequency plot.
7. Provide an overall interpretation (see above example and those on slide 45 for Week 1's lecture).

1. Create a column called TotalStudents containing the total number of students for each school

```r
school_data$TotalStudents <- school_data$NumberBoys +  school_data$NumberGirls
```

```
##  NumberBoys NumberGirls TotalStudents
##         115         110           225
##         180         215           395
##         250         220           470
##         245         205           450
##         110         115           225
##         165         190           355
##         400         420           820
##         140         125           265
##         185         190           375
```

2. Create a column called PercentGirls which contains the estimated percentage of girls attending each school.

```
school_data$PercentGirls <- school_data$NumberGirls/school_data$TotalStudents * 100
```

```
##                                        SchoolName NumberGirls PercentGirls
## 1 St Mary's Kilburn Church of England Primary School         110     48.88889
## 2                        De Beauvoir Primary School         215     54.43038
## 3                       Queensbridge Primary School         220     46.80851
## 4                        Princess May Primary School         205     45.55556
## 5      Holy Trinity Church of England Primary School         115     51.11111
## 6                        Marlborough Primary School         190     53.52113
```

3. Use the tapply() function to calculate the total number of student by subgroup or type of school

The **tapply()** function it is a tool for applying a specific function (‘FUN’) to a vector (dataframe) (‘School_data’), segmented by one or more categories. It splits the data into groups based on the levels of the factors, applies the function to each group, and returns the results in a convenient format

```
tapply(school_data$TotalStudents, school_data$Type, FUN = sum)
```

```
## All Through     Primary   Secondary
##       33040      687510      228610
```

Seminar Questions

1. What are the names of schools that have the highest and lowest number of students? **[HINT: use max(), min() and View() functions]**

2. Generate a fully labeled `barplot()` and describe the distribution of schools by OFSTED scores.

3. Explore the frequency of schools in London based on the total number of students. Make a frequency table in R by creating the groups based on total students per school - starting from 0 to 2200, use the interval of 200 students to generate the groups as 0-200, 201-400, 401-600, . . . , 2001-2200 and hence the number of schools falling into these groupings with these numbers of students

4. Create a histogram.

5. Create a cumulative frequency plot.

6. Create a relative cumulative frequency plot.

7. Provide an overall interpretation (see above example and those on slide 45 for Week 1's lecture).

1. What are the names of schools that have the highest and lowest number of students? **[HINT: use max(), min() and View() functions]**

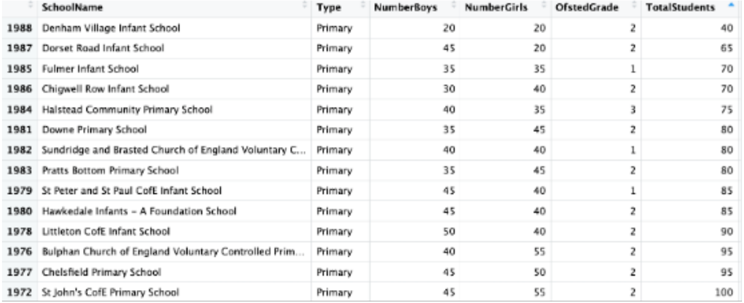
```
# Name of school with lowest value has only 40 students
min(school_data$TotalStudents)
```

```
## [1] 40
```

```
# Name of school with highest value has up to 2,065 students
max(school_data$TotalStudents)
```

```
## [1] 2065
```

1. What are the names of schools that have the highest and lowest number of students? **[HINT: use max(), min() and View() functions]**



	SchoolName	Type	NumberBoys	NumberGirls	OfstedGrade	TotalStudents
1988	Denham Village Infant School	Primary	20	20	2	40
1987	Dorset Road Infant School	Primary	45	20	2	65
1985	Fulmer Infant School	Primary	35	35	1	70
1986	Chigwell Row Infant School	Primary	30	40	2	70
1984	Halstead Community Primary School	Primary	40	35	3	75
1981	Downe Primary School	Primary	35	45	2	80
1982	Sundridge and Brasted Church of England Voluntary C...	Primary	40	40	1	80
1983	Pratts Bottom Primary School	Primary	35	45	2	80
1979	St Peter and St Paul CofE Infant School	Primary	45	40	1	85
1980	Hawkedale Infants – A Foundation School	Primary	45	40	2	85
1978	Littleton CofE Infant School	Primary	50	40	2	90
1976	Bulphan Church of England Voluntary Controlled Prim...	Primary	40	55	2	95
1977	Chelsfield Primary School	Primary	45	50	2	95
1972	St John's CofE Primary School	Primary	45	55	2	100

2. Generate a fully labeled `barplot()` and describe the distribution of schools by OFSTED scores.

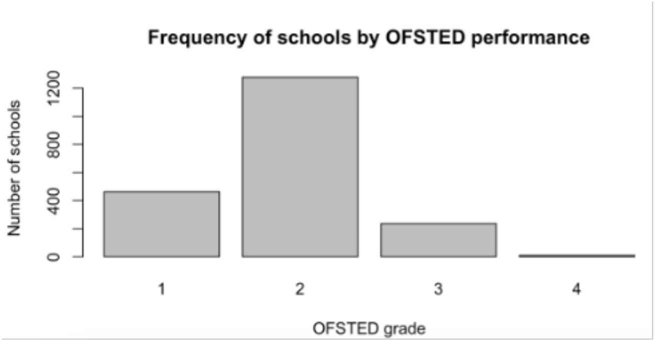
```r
# step 1: to create a barplot, use the table() function to
# tabulate number of schools based on ofsted grades
table(school_data$OfstedGrade)
```

```
##
##    1    2    3    4
##  463 1279  236   10
```

```r
# step 2: pass the results from table() in an object called 'freq'
freq <- table(school_data$OfstedGrade)
```

```r
# step 3
barplot(freq, main="Frequency of schools by OFSTED performance",
        xlab="OFSTED grade", ylab="Number of schools")
```

2. Generate a fully labeled `barplot()` and describe the distribution of schools by OFSTED scores.



3. Explore the frequency of schools in London based on the total number of students(. . .)

```
# create this classes ingredient for the plots
classes <- seq(0, 2200, 200)
classes
```

```
##  [1]    0  200  400  600  800 1000 1200 1400 1600 1800 2000 2200
```

```
# apply the cut() function onto the TotalStudents column to create a groups column
# dig.lab removes scientific notation in interval
school_data$Groups <- cut(school_data$TotalStudents, breaks=classes, dig.lab = 5)
```

```
##                                           SchoolName TotalStudents    Groups
## 1 St Mary's Kilburn Church of England Primary School           225 (200,400]
## 2                         De Beauvoir Primary School           395 (200,400]
## 3                        Queensbridge Primary School           470 (400,600]
## 4                        Princess May Primary School           450 (400,600]
## 5      Holy Trinity Church of England Primary School           225 (200,400]
## 6                         Marlborough Primary School           355 (200,400]
```

3. Explore the frequency of schools in London based on the total number of students(. . .)

```
frequency_results <- data.frame(table(school_data$Groups))
#this renames the columns
colnames(frequency_results)[1] <- "Groups"
colnames(frequency_results)[2] <- "Frequency"
# prints partial frequency table
frequency_results
```

```
##          Groups Frequency
## 1       (0,200]       108
## 2     (200,400]       811
## 3     (400,600]       643
## 4     (600,800]       198
## 5    (800,1000]        87
## 6   (1000,1200]        60
## 7   (1200,1400]        41
## 8   (1400,1600]        24
## 9   (1600,1800]         9
## 10  (1800,2000]         6
## 11  (2000,2200]         1
```

3. Explore the frequency of schools in London based on the total number of students(. . .)

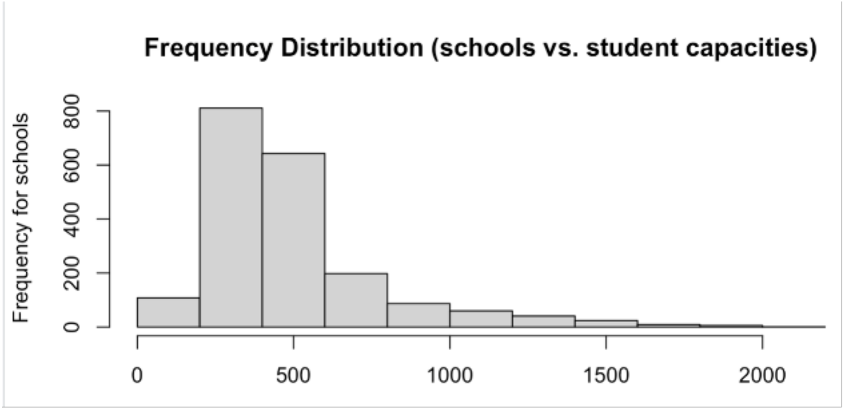
```
# calculating the relative freq., cumulative freq., and cumulative relative frequency
frequency_results$RelativeFreq <- frequency_results$Frequency/sum(frequency_results$Frequen
frequency_results$CumulativeFreq <- cumsum(frequency_results$Frequency)
frequency_results$CumulativeRelFreq <- cumsum(frequency_results$RelativeFreq)

# prints full frequency table (1 output)
frequency_results
```

```
##         Groups Frequency RelativeFreq CumulativeFreq CumulativeRelFreq
## 1      (0,200]       108 0.0543259557            108        0.05432596
## 2    (200,400]       811 0.4079476861            919        0.46227364
## 3    (400,600]       643 0.3234406439           1562        0.78571429
## 4    (600,800]       198 0.0995975855           1760        0.88531187
## 5   (800,1000]        87 0.0437625755           1847        0.92907445
## 6  (1000,1200]        60 0.0301810865           1907        0.95925553
## 7  (1200,1400]        41 0.0206237425           1948        0.97987928
## 8  (1400,1600]        24 0.0120724346           1972        0.99195171
## 9  (1600,1800]         9 0.0045271630           1981        0.99647887
## 10 (1800,2000]         6 0.0030181087           1987        0.99949698
## 11 (2000,2200]         1 0.0005030181           1988        1.00000000
```

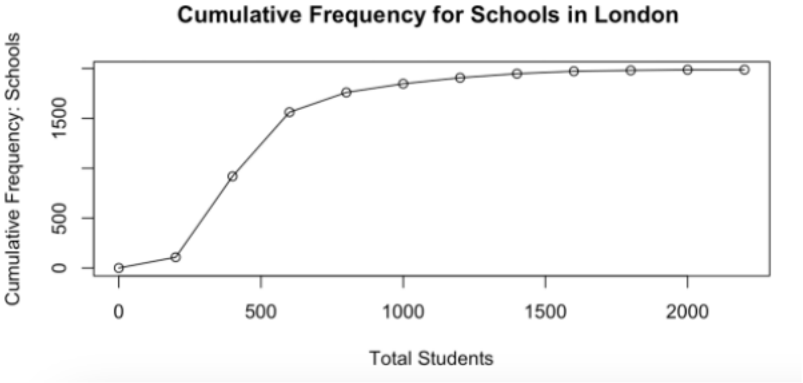
4. Create a histogram of frequency of schools in London based on total number of student

```
hist(school_data$TotalStudents, breaks = classes,
    ylab="Frequency for schools",
    xlab="Total students",
    main= "Frequency Distribution schools")
```



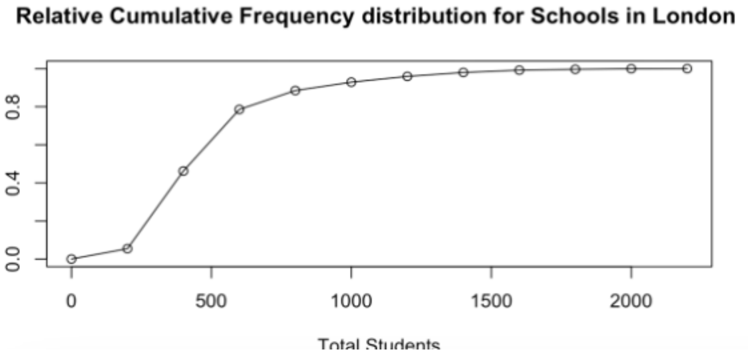
5. Create a cumulative frequency plot of schools in London based on total number of student

```
# create the plot with student values on x-axis vs
# cumulative frequency values (school number) on y-axis
plot(classes, cumulative.freq0, main="Cumulative Frequency" ,
     xlab="Total Students", ylab="Cumulative Frequency: Schools")
#connect the dots and BOOM
lines(classes, cumulative.freq0)
```



6. Create a relative cumulative frequency plot.

```
# extract the relative cumulative frequency from output frequency table
rel.cumulative.freq0 <- c(0, frequency_results$CumulativeRelFreq)
# create the plot with student values on x-axis vs cumulative frequency values (school numbe
plot(classes, rel.cumulative.freq0, main="Relative Cumulative Frequency distribution for Sc
#connect the dots and BOOM
lines(classes, rel.cumulative.freq0)
```


Relative Cumulative Frequency distribution for Schools in London
Total Students