Summarizing and Visualizing Data: Part I

Eli Gurarie

StatR 101 - Lecture 2a
October 1, 2012

October 1, 2012

What to do when confronted with data

- Load it
- Make a table
- Draw some plots
- Summarize

These steps are known as **exploratory data analysis**

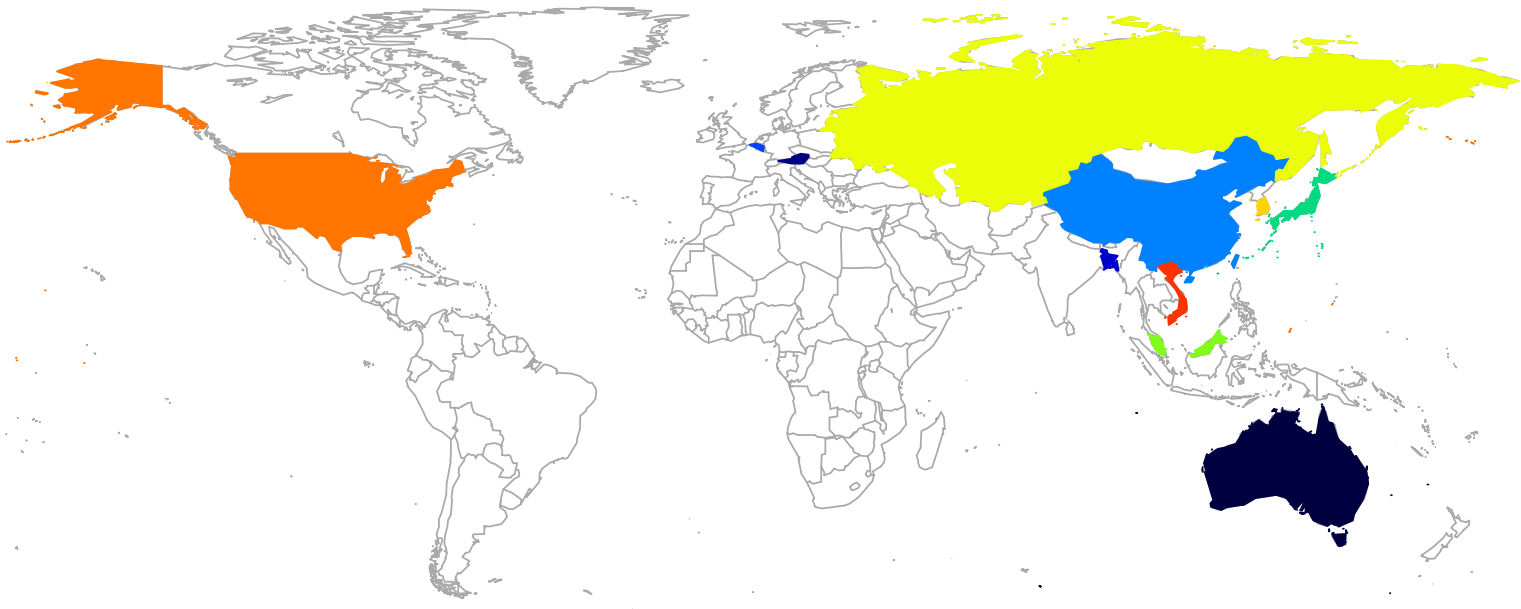
Data set 1: Undergraduate Statistics Students

Country of birth: (from two quiz-sections)

95 students responded, from 15 countries.

```
 [1] South Korea USA         USA         Bangladesh  Singapore   Vietnam
 [7] USA         China       China       China       China       South Korea
[13] USA         Taiwan      China       USA         USA         USA
[19] China       China       USA         China       China       China
[25] China       Taiwan      Hong Kong   Macau       USA         <NA>
[31] China       Taiwan      China       USA         USA         USA
[37] USA         USA         USA         USA         Singapore   South Korea
[43] China       USA         China       USA         USA         USA
[49] USA         China       China       USA         China       USA
[55] Malaysia    Japan       China       China       Taiwan      USA
[61] USA         China       USA         USA         USA         USA
[67] USA         Japan       Vietnam     Austria     China       USA
[73] China       USA         Russia      South Korea USA         South Korea
[79] USA         USA         Taiwan      USA         USA         Taiwan
[85] Australia   Belgium     Japan       South Korea China       China
[91] China       China       China       China       Taiwan
15 Levels: Australia Austria Bangladesh Belgium China Hong Kong Japan ... Vietnam
```

Country of birth: Map



Type of data: **Categorical/qualitative**

- **Discrete** categories
  - Country of birth
  - Major
  - Color of something
- Can be **ordered**
  - Letter grade: A, B, C, D, F. (but not grade point score)
  - Survey response: "strongly disagree", "somewhat disagree" ...
  - Body Condition: "Poor", "Fair", "Good", "Very Good", "Excellent"),
  - Year in school: ("freshman", "sophomore", ...)
- Can be **binary** (take only two values)
  - Sex (Male or Female)
  - True / False

Type of data: **Categorical/qualitative**

- **Discrete** categories
  - Country of birth
  - Major
  - Color of something
- Can be **ordered**
  - Letter grade: A, B, C, D, F. (but not grade point score)
  - Survey response: "strongly disagree", "somewhat disagree" ...
  - Body Condition: "Poor", "Fair", "Good", "Very Good", "Excellent"),
  - Year in school: ("freshman", "sophomore", ...)
- Can be **binary** (take only two values)
  - Sex (Male or Female)
  - True / False

Type of data: **Categorical/qualitative**

- **Discrete** categories
  - Country of birth
  - Major
  - Color of something
- Can be **ordered**
  - Letter grade: A, B, C, D, F. (but not grade point score)
  - Survey response: "strongly disagree", "somewhat disagree" ...
  - Body Condition: "Poor", "Fair", "Good", "Very Good", "Excellent"),
  - Year in school: ("freshman", "sophomore", ...)
- Can be **binary** (take only two values)
  - Sex (Male or Female)
  - True / False

Categorical/qualitative data in R

Categorical data are stored in R as characters

```
>Countries <- c("Albania","Botswana","Chile","Chile","Djibouti")
>is(Countries)
[1] "character"           "vector"
> unique(Countries)
[1] "Albania"  "Botswana" "Chile"    "Djibouti"
> substr(Countries, 1, 4)
[1] "Alba" "Bots" "Chil" "Chil" "Djib"
```

or, more generally and usefully, as factors

```
> Countries <- factor(c("Albania", "Botswana", "Chile", "Chile", "Denmark"))
> Countries
[1] Albania  Botswana Chile    Chile    Djibouti
 Levels: Albania Botswana Chile Djibouti
> levels(Countries)
[1] "Albania"  "Botswana" "Chile"    "Djibouti"
> as.integer(Countries)
[1] 1 2 3 3 4
 > # Note: factors come with levels (i.e.~possible values) which are
 > #internally coded as integers.
```

Categorical/qualitative data in R

Categorical data are stored in R as characters

```
>Countries <- c("Albania","Botswana","Chile","Chile","Djibouti")
>is(Countries)
[1] "character"           "vector"
> unique(Countries)
[1] "Albania"  "Botswana" "Chile"    "Djibouti"
> substr(Countries, 1, 4)
[1] "Alba" "Bots" "Chil" "Chil" "Djib"
```

or, more generally and usefully, as factors

```
> Countries <- factor(c("Albania", "Botswana", "Chile", "Chile", "Denmark"))
> Countries
[1] Albania  Botswana Chile    Chile    Djibouti
 Levels: Albania Botswana Chile Djibouti
> levels(Countries)
[1] "Albania"  "Botswana" "Chile"    "Djibouti"
> as.integer(Countries)
[1] 1 2 3 3 4
 > # Note: factors come with levels (i.e.~possible values) which are
 > #internally coded as integers.
```

Country of birth: Make a Table

	Country	Number of students
1	Australia	1
2	Austria	1
3	Bangladesh	1
4	Belgium	1
5	China	29
6	Hong Kong	1
7	Japan	3
8	Macau	1
9	Malaysia	1
10	Russia	1
11	Singapore	2
12	South Korea	6
13	Taiwan	7
14	USA	37
15	Vietnam	2
	Total	95

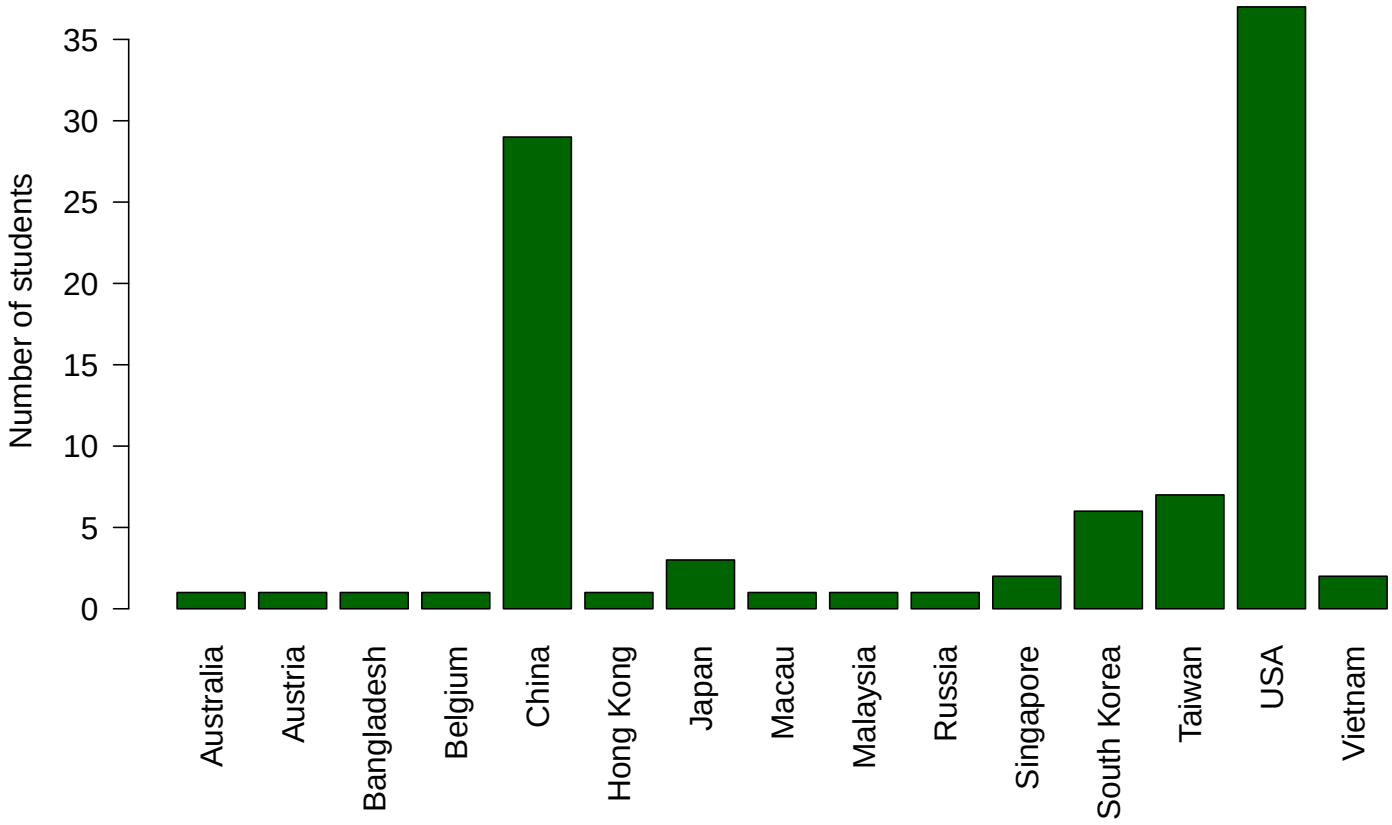
R code: tabulating data with `table()`

```
> table(countries)
countries
 Australia     Austria  Bangladesh     Belgium       China   Hong Kong       Japan
         1           1           1           1          29           1           3
  Malaysia      Russia   Singapore South Korea      Taiwan         USA     Vietnam
         1           1           2           6           7          37           2
```

R code: using `data.frame()`

```
> data.frame(table(countries))
      countries Freq
1     Australia    1
2       Austria    1
3    Bangladesh    1
4       Belgium    1
5         China   29
6     Hong Kong    1
7         Japan    3
8         Macau    1
9      Malaysia    1
10       Russia    1
11    Singapore    2
12 South Korea    6
13       Taiwan    7
14          USA   37
15      Vietnam    2
```

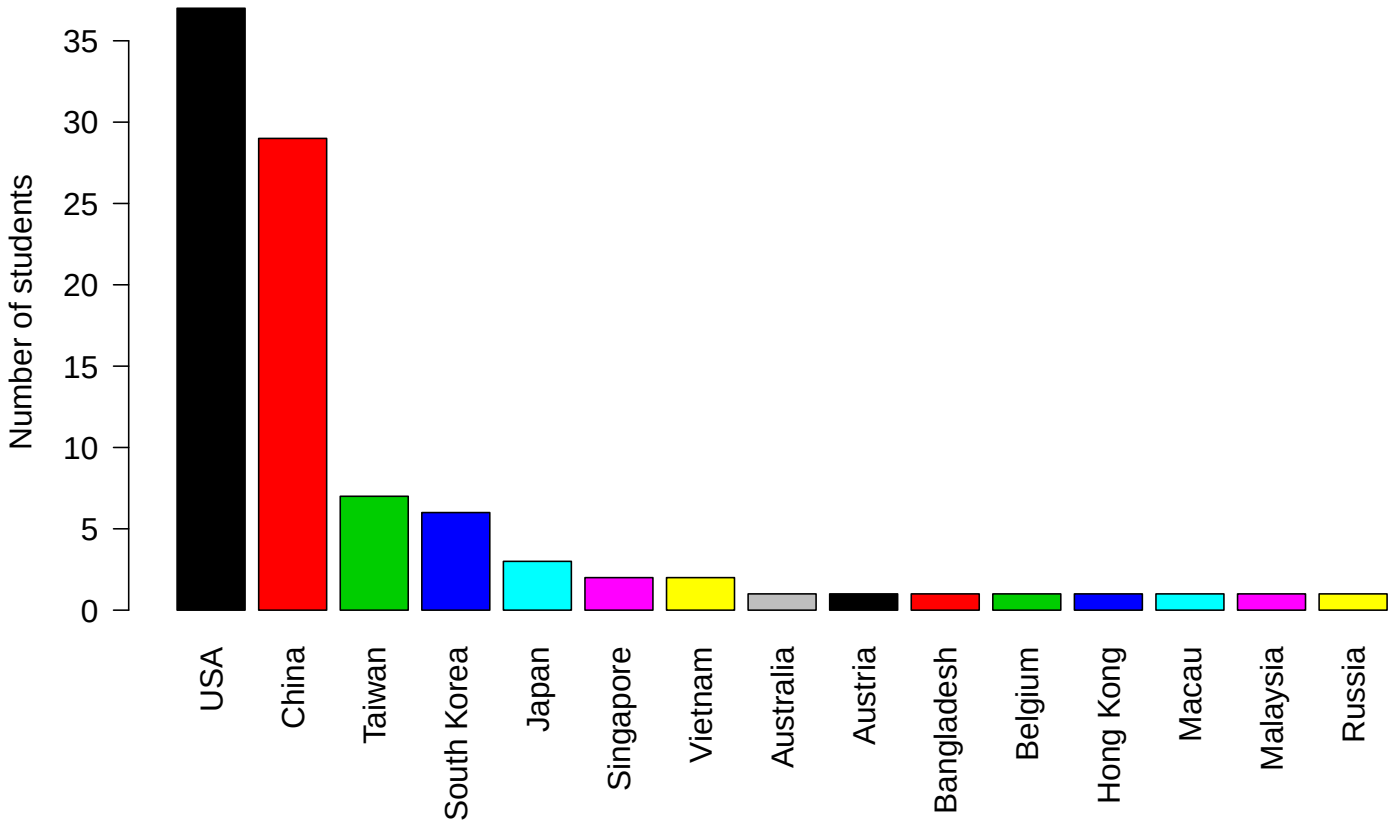
Country of birth: Bar plot



Rcode: `barplot()`

```
> barplot(table(countries), col="darkgreen",
          ylab="Number of students")
```

Country of birth: Ordered bar plot

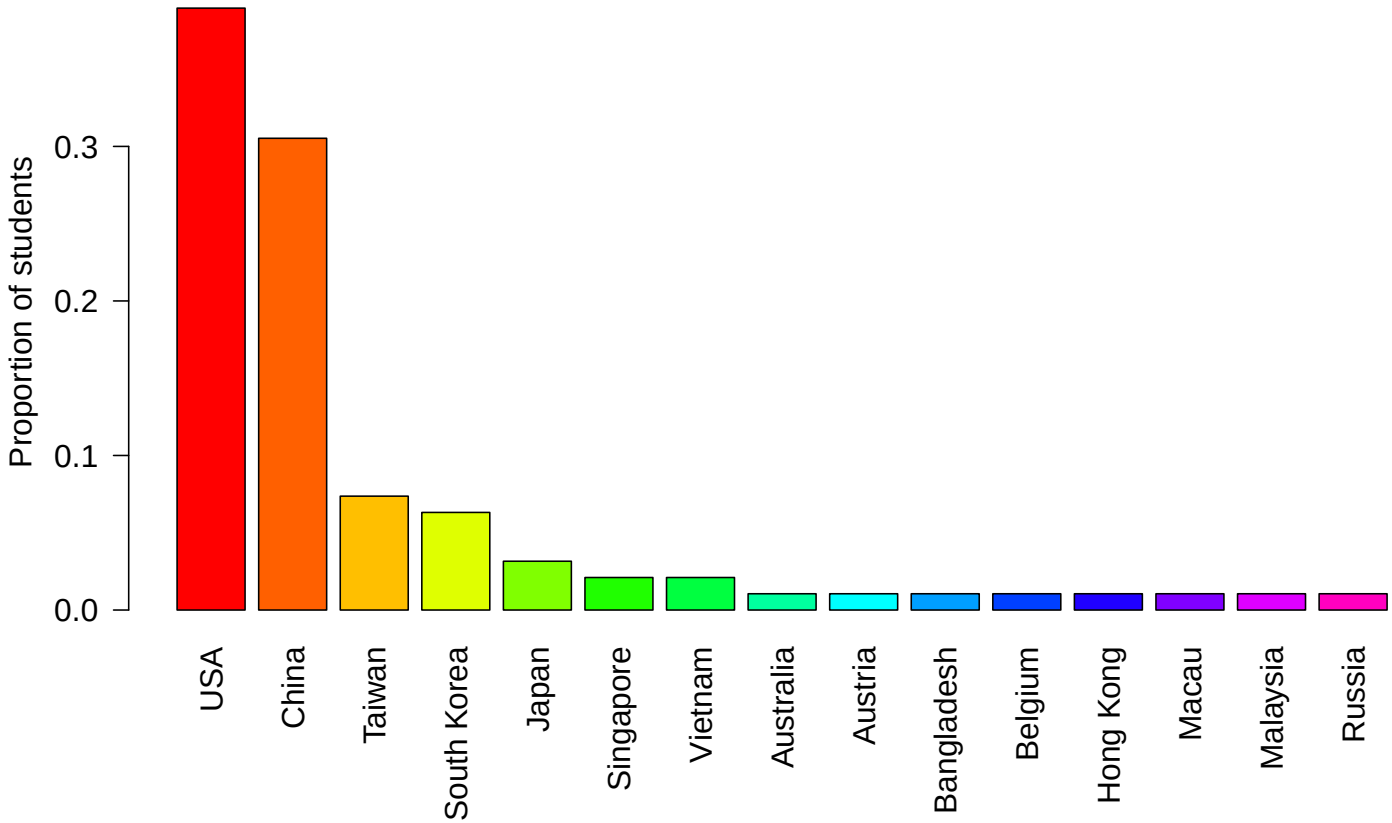


Rcode: `barplot()` **and** `sort()`

```
> n.countries <- length(unique(countries))
> barplot(sort(table(countries), decreasing=TRUE),
          col=1:n.countries, ylab="Number of students")
```

Note: the enumerated colors.

Country of birth: Bar plot of proportions

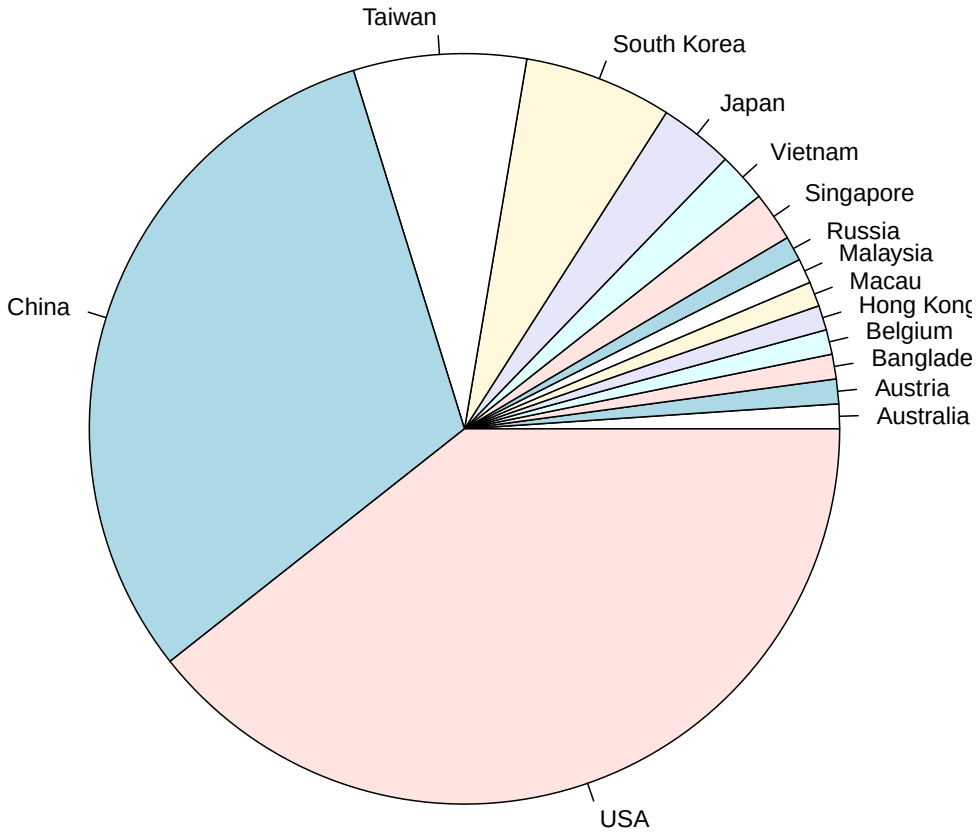


Rcode: `barplot()` **and** `sort()`

```
P.countries <- sort(table(countries), decreasing=TRUE)/n.countries
barplot(P.countries, col=rainbow(n.countries),
        ylab="Proportion of students")
```

Note: `sum(P.countries) == 1`

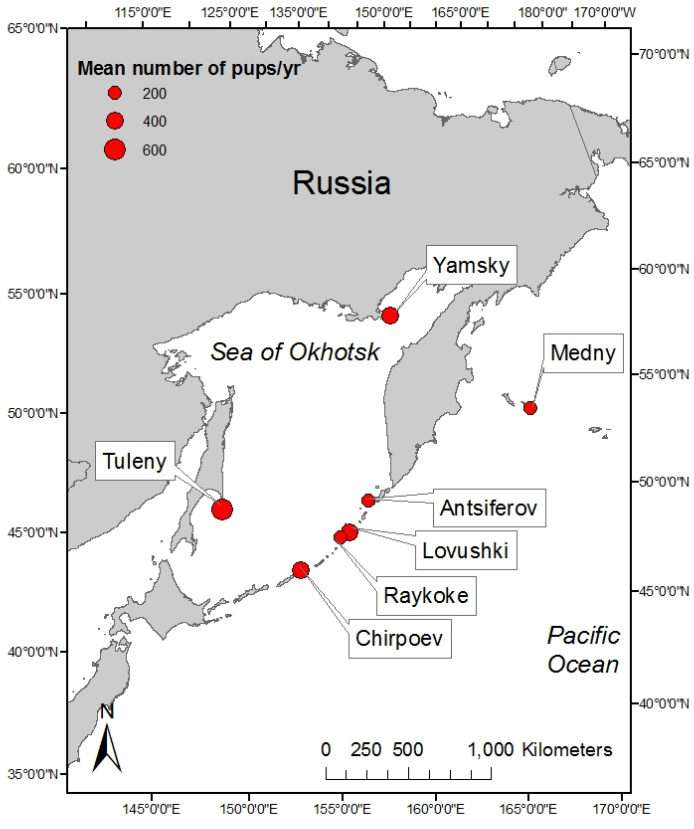
Country of birth: Pie chart

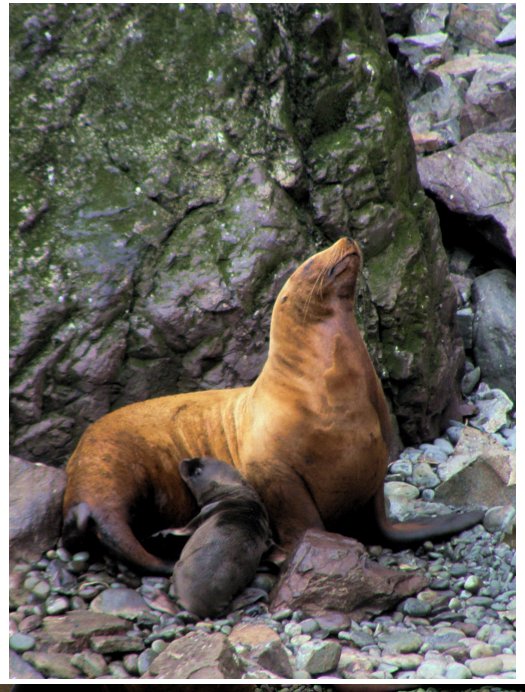


Rcode: `pie()`

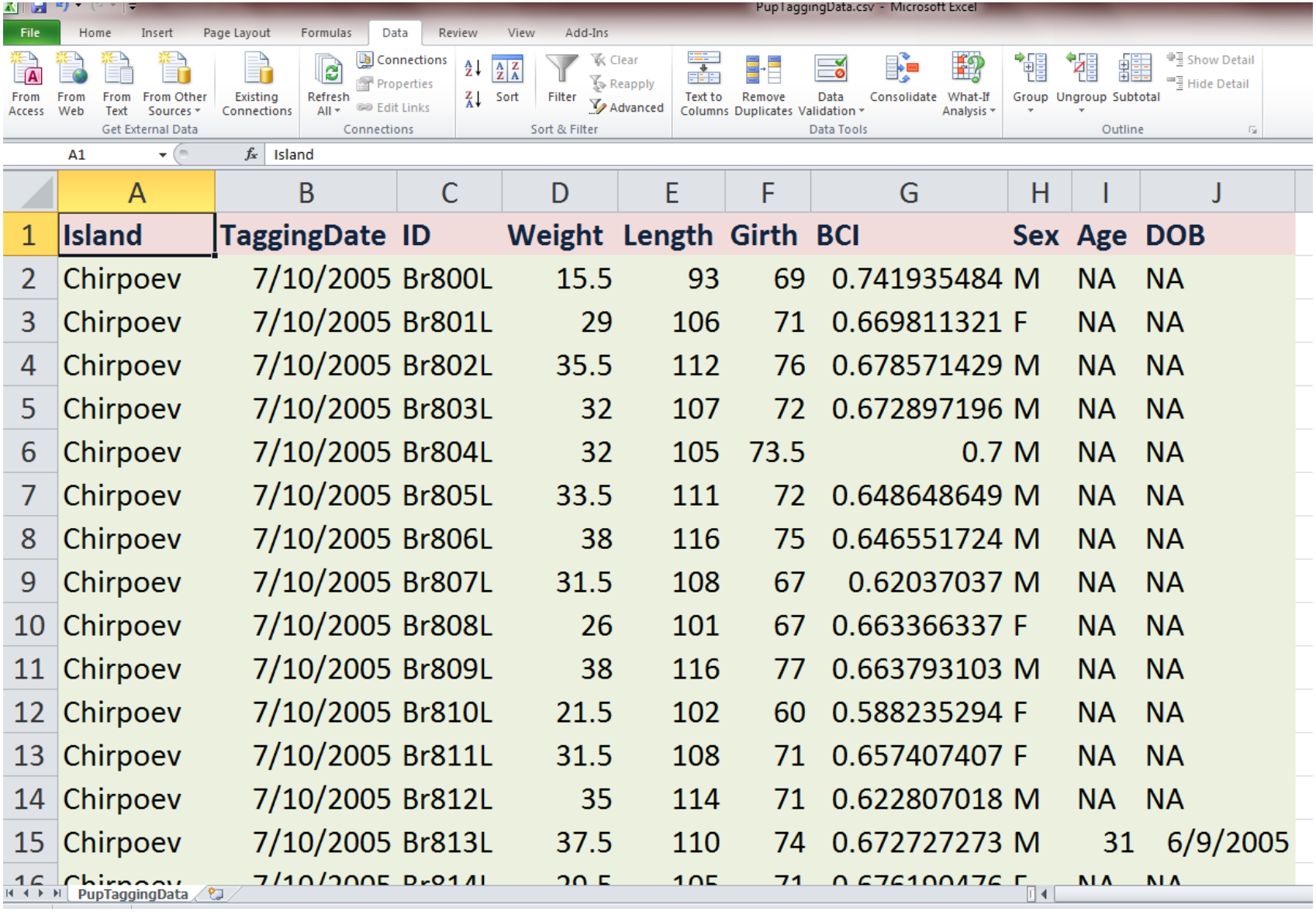
```
> pie(sort(table(countries)))
# Very simple! (but, in my experience, not very useful)
```

Data set 2: Steller sea lions in Russia





Raw data: Tagging pups



	A	B	C	D	E	F	G	H	I	J
1	Island	TaggingDate	ID	Weight	Length	Girth	BCI	Sex	Age	DOB
2	Chirpoev	7/10/2005	Br800L	15.5	93	69	0.741935484	M	NA	NA
3	Chirpoev	7/10/2005	Br801L	29	106	71	0.669811321	F	NA	NA
4	Chirpoev	7/10/2005	Br802L	35.5	112	76	0.678571429	M	NA	NA
5	Chirpoev	7/10/2005	Br803L	32	107	72	0.672897196	M	NA	NA
6	Chirpoev	7/10/2005	Br804L	32	105	73.5	0.7	M	NA	NA
7	Chirpoev	7/10/2005	Br805L	33.5	111	72	0.648648649	M	NA	NA
8	Chirpoev	7/10/2005	Br806L	38	116	75	0.646551724	M	NA	NA
9	Chirpoev	7/10/2005	Br807L	31.5	108	67	0.62037037	M	NA	NA
10	Chirpoev	7/10/2005	Br808L	26	101	67	0.663366337	F	NA	NA
11	Chirpoev	7/10/2005	Br809L	38	116	77	0.663793103	M	NA	NA
12	Chirpoev	7/10/2005	Br810L	21.5	102	60	0.588235294	F	NA	NA
13	Chirpoev	7/10/2005	Br811L	31.5	108	71	0.657407407	F	NA	NA
14	Chirpoev	7/10/2005	Br812L	35	114	71	0.622807018	M	NA	NA
15	Chirpoev	7/10/2005	Br813L	37.5	110	74	0.672727273	M	31	6/9/2005
16	Chirpoev	7/10/2005	Br814L	29.5	105	71	0.676190476	F	NA	NA

R note: Loading data

There are several ways of loading data in R (other than entering each number by hand). The single most useful one is: `read.csv()`.

- "CSV" stands for *comma separated values*, which is a text file with:
  1. commas separating columns
  2. a header row of names for columns
- This is very typical for data organization, and it is easy to export data from Excel, Access, or any data base program in this format.

Rcode: `read.csv()`

```
> # set your working directory
> setwd("c:/eli/teaching/statR101/Week2/")
> # load the data
> Pups <- read.csv("./data/PupTaggingData.csv")
```

R note: data frames

Rcode: What `is()` Pups?

```
> is(Pups)
[1] "data.frame" "list"       "oldClass"   "vector"
```

Data frames are the most common way of storing data in R.

- They combine *vectors* of different *types* (e.g. numeric, characters, factors)
- They are the default object type of data that is read using the `read.csv()` (or `read.table()`) functions.

Rcode: Attributes of data frames

```
> names(Pups)  # names of columns
 [1] "Island"      "TaggingDate" "ID"          "Weight"      "Length"      "Girth"
 [7] "BCI"         "Sex"         "Age"         "DOB"
> dim(Pups)    # dimension of data frame
[1] 498  10
> head(Pups)   # first few rows
    Island TaggingDate     ID Weight Length Girth       BCI Sex Age  DOB
1 Chirpoev   7/10/2005 Br800L   15.5     93  69.0 0.7419355   M  NA <NA>
2 Chirpoev   7/10/2005 Br801L   29.0    106  71.0 0.6698113   F  NA <NA>
3 Chirpoev   7/10/2005 Br802L   35.5    112  76.0 0.6785714   M  NA <NA>
4 Chirpoev   7/10/2005 Br803L   32.0    107  72.0 0.6728972   M  NA <NA>
5 Chirpoev   7/10/2005 Br804L   32.0    105  73.5 0.7000000   M  NA <NA>
6 Chirpoev   7/10/2005 Br805L   33.5    111  72.0 0.6486486   M  NA <NA>
```

R note: data frames

Rcode: What `is()` Pups?

```
> is(Pups)
[1] "data.frame" "list"       "oldClass"   "vector"
```

Data frames are the most common way of storing data in R.

- They combine *vectors* of different *types* (e.g. numeric, characters, factors)
- They are the default object type of data that is read using the `read.csv()` (or `read.table()`) functions.

Rcode: Attributes of data frames

```
> names(Pups)  # names of columns
 [1] "Island"      "TaggingDate" "ID"          "Weight"      "Length"      "Girth"
 [7] "BCI"         "Sex"         "Age"         "DOB"
> dim(Pups)    # dimension of data frame
[1] 498  10
> head(Pups)   # first few rows
    Island TaggingDate     ID Weight Length Girth       BCI Sex Age  DOB
1 Chirpoev   7/10/2005 Br800L   15.5     93  69.0 0.7419355   M  NA <NA>
2 Chirpoev   7/10/2005 Br801L   29.0    106  71.0 0.6698113   F  NA <NA>
3 Chirpoev   7/10/2005 Br802L   35.5    112  76.0 0.6785714   M  NA <NA>
4 Chirpoev   7/10/2005 Br803L   32.0    107  72.0 0.6728972   M  NA <NA>
5 Chirpoev   7/10/2005 Br804L   32.0    105  73.5 0.7000000   M  NA <NA>
6 Chirpoev   7/10/2005 Br805L   33.5    111  72.0 0.6486486   M  NA <NA>
```

R note: "structure" of data objects

Rcode: for an astonishing amount of information, use `str()`

```
> str(Pups)
'data.frame':   498 obs. of  10 variables:
 $ Island     : Factor w/ 5 levels "Antsiferov","Chirpoev",..: 2 2 2 2 2 2 2  ...
 $ TaggingDate: Factor w/ 5 levels "7/10/2005","7/4/2005",..: 1 1 1 1 1 1 1  ...
 $ ID         : Factor w/ 498 levels "Br800L","Br801L",..: 1 2 3 4 5 6 7 8  ...
 $ Weight     : num  15.5 29 35.5 32 32 33.5 38 31.5 26 38 ...
 $ Length     : int  93 106 112 107 105 111 116 108 101 116 ...
 $ Girth      : num  69 71 76 72 73.5 72 75 67 67 77 ...
 $ BCI        : num  0.742 0.67 0.679 0.673 0.7 ...
 $ Sex        : Factor w/ 2 levels "F","M": 2 1 2 2 2 2 2 2 1 2 ...
 $ Age        : int  NA NA NA NA NA NA NA NA NA NA ...
 $ DOB        : Factor w/ 19 levels "5/16/2005","6/1/2005",..: NA NA NA NA NA ...
```

R note: Extracting data from data frames

Rcode: Using matrix subsetting and the all-important $

```
> # sample 6 random pups
> Pups2 <- Pups[sample(1:nrow(Pups),6),]
> Pups2[1,]      # first row
        Island TaggingDate    ID Weight Length Girth   BCI Sex Age  DOB
498 Antsiferov    7/6/2005 Y650L   26.5    100  71.5 0.715   M  NA <NA>
> Pups2[,1]      # first column
[1] Antsiferov Raykoke    Raykoke    Antsiferov Lovushki   Chirpoev
Levels: Antsiferov Chirpoev Lovushki Raykoke Srednova
> Pups2$Island   # indexed by name
[1] Antsiferov Raykoke    Raykoke    Antsiferov Lovushki   Chirpoev
Levels: Antsiferov Chirpoev Lovushki Raykoke Srednova
> Pups2$Weight
[1] 26.5 38.5 43.0 38.5 36.8 38.0
> Pups2$Sex
[1] M M M M F M
Levels: F M
> Pups2[Pups2$Length < 110,]
            Island TaggingDate     ID Weight Length Girth       BCI Sex Age  DOB
253 Lovushki    7/4/2005 Lr654L   26.5    102  67.0 0.6568627   M  NA <NA>
131 Srednova    7/9/2005  C631L   33.5    106  75.5 0.7122642   F  NA <NA>
311  Raykoke    7/8/2005  P911L   29.5    107  70.5 0.6588785   F  NA <NA>
8   Chirpoev   7/10/2005 Br807L   31.5    108  67.0 0.6203704   M  NA <NA>
```

Tabulating data

Island	Males	Females	Total
Brat Chirpoev	55	43	98
Srednova	53	46	99
Lovushki	50	50	100
Raykoke	51	49	100
Antsiferov	62	38	100

Rcode: tables

```
> table(Pups$Island, Pups$Sex)
             F  M
 Antsiferov 38 62
 Chirpoev   43 56
 Lovushki   50 50
 Raykoke    49 51
 Srednova   46 53
> cbind(table(Pups$Island, Pups$Sex), table(Pups$Island))
            F  M
Antsiferov 38 62 100
Chirpoev   43 56  99
Lovushki   50 50 100
Raykoke    49 51 100
Srednova   46 53  99
```

Tabulating data

Island	Males	Females	Total
Brat Chirpoev	55	43	98
Srednova	53	46	99
Lovushki	50	50	100
Raykoke	51	49	100
Antsiferov	62	38	100

Rcode: tables

```
> table(Pups$Island, Pups$Sex)
             F  M
 Antsiferov 38 62
 Chirpoev   43 56
 Lovushki   50 50
 Raykoke    49 51
 Srednova   46 53
> cbind(table(Pups$Island, Pups$Sex), table(Pups$Island))
             F  M
Antsiferov 38 62 100
Chirpoev   43 56  99
Lovushki   50 50 100
Raykoke    49 51 100
Srednova   46 53  99
```

Weights (kg) of pups on Brat Chirpoev - Ordered

Males: 15.5, 26.5, 26.5, 26.5, 27, 27, 28.5, 28.5, 29, 29.5, 30, 31, 31, 31, 31, 31.5, 31.5, 32, 32, 32, 33, 33, 33.5, 33.5, 33.5, 34.5, 35, 35.5, 35.5, 35.5, 35.5, 36, 36, 37, 37.5, 37.5, 38, 38, 38.5, 38.5, 38.5, 39, 40, 40, 40, 40, 40, 40, 40.5, 41.5, 41.5, 42.5, 43, 43.5, 44, 44.5, 45

Females: 20, 21.5, 22, 22.5, 23, 23, 24, 24, 25.5, 26, 26, 26, 26.5, 27, 27, 27, 28.5, 28.5, 28.5, 28.5, 28.5, 29, 29, 29.5, 30, 30, 30.5, 30.5, 31, 31, 31, 31.5, 31.5, 31.5, 32, 32.5, 32.5, 32.5, 34.5, 34.5, 35.5, 35.5, 36.5

R code

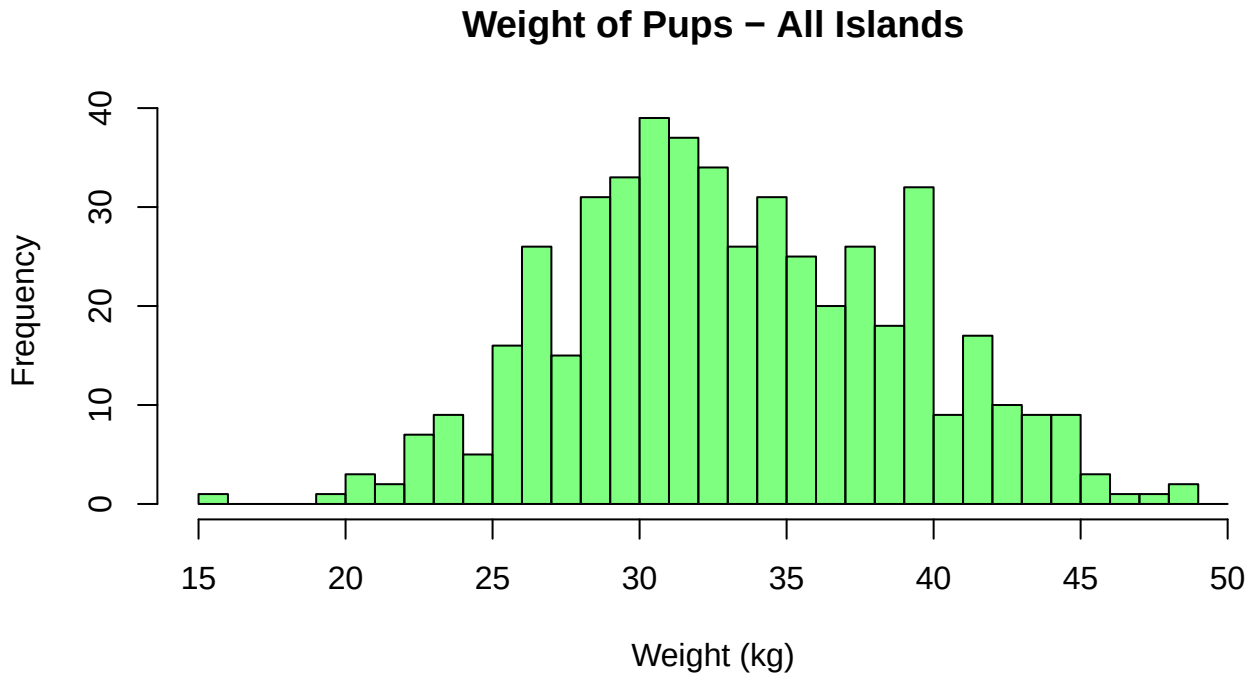
```
> sort(Pups$Weight[Pups$Island == "Chirpoev" & Pups$Sex== "M"])
> sort(Pups$Weight[Pups$Island == "Chirpoev" & Pups$Sex== "F"])
```

Continuous or **Quantitative** measurement

- Any variable that can take a "continuous" range of values.
- "Bigger" and "Smaller" are well defined.
  - e.g. Length , Weight , Girthand Time. Note that the first three are always positive, whereas Time depends on your frame of reference.
- Often **discretized**
  - weights of pups measured in units of a single kilogram, or 0.5 kilogram
  - age often reported in "years" (18, 19, 20, 21 ...), not (19.2654)
- Can be **Wrapped**
  - e.g. Time of Day is between 0:00 and 24:00, but 0:00 = 24:00
- In `R` as *numeric* (same as *double*) or *integer* values.

Histogram of **frequencies**

Y-axis: Counts of observations made in each bin.



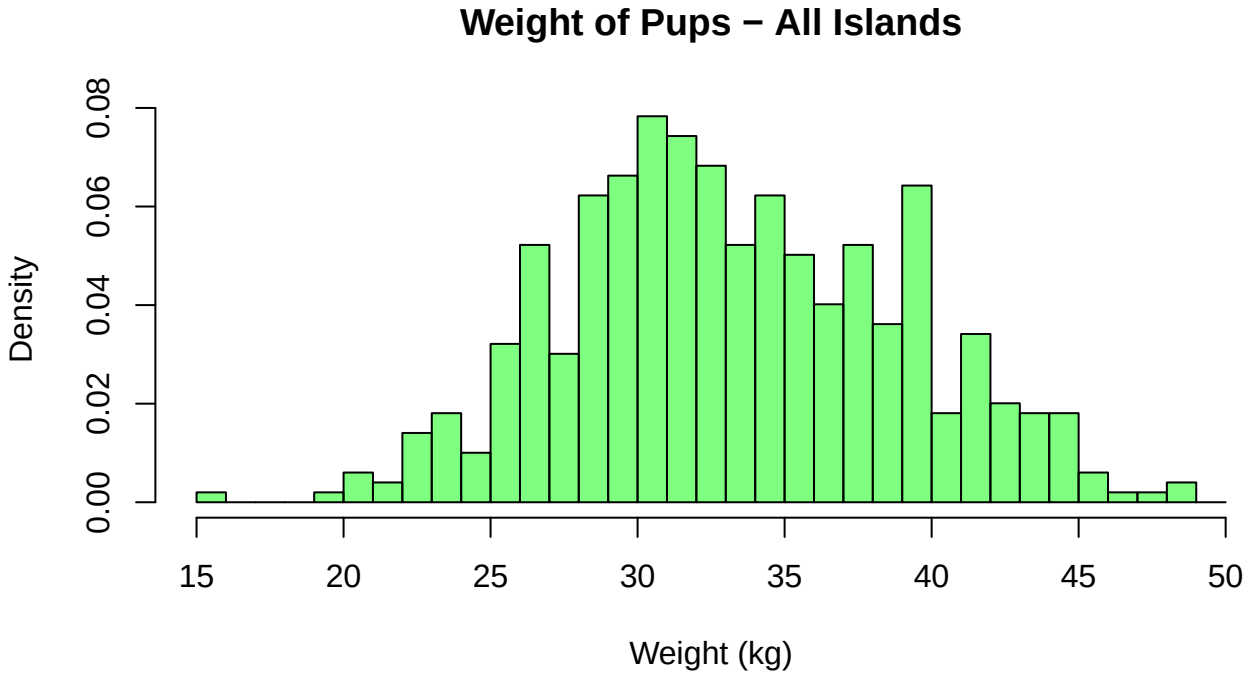
Note: the sum of the all the bars is equal to the total number $\sum_{i=1}^{n} F_i = N_{total}$

R code

```
> hist(P$Weight[P$Island=="Chirpoev"],
>   main="Weight of Pups - Chirpoev Island", xlab="Weight (kg)")
```

Histogram of **densities**

Y-axis: Proportion of observations made in each bin.



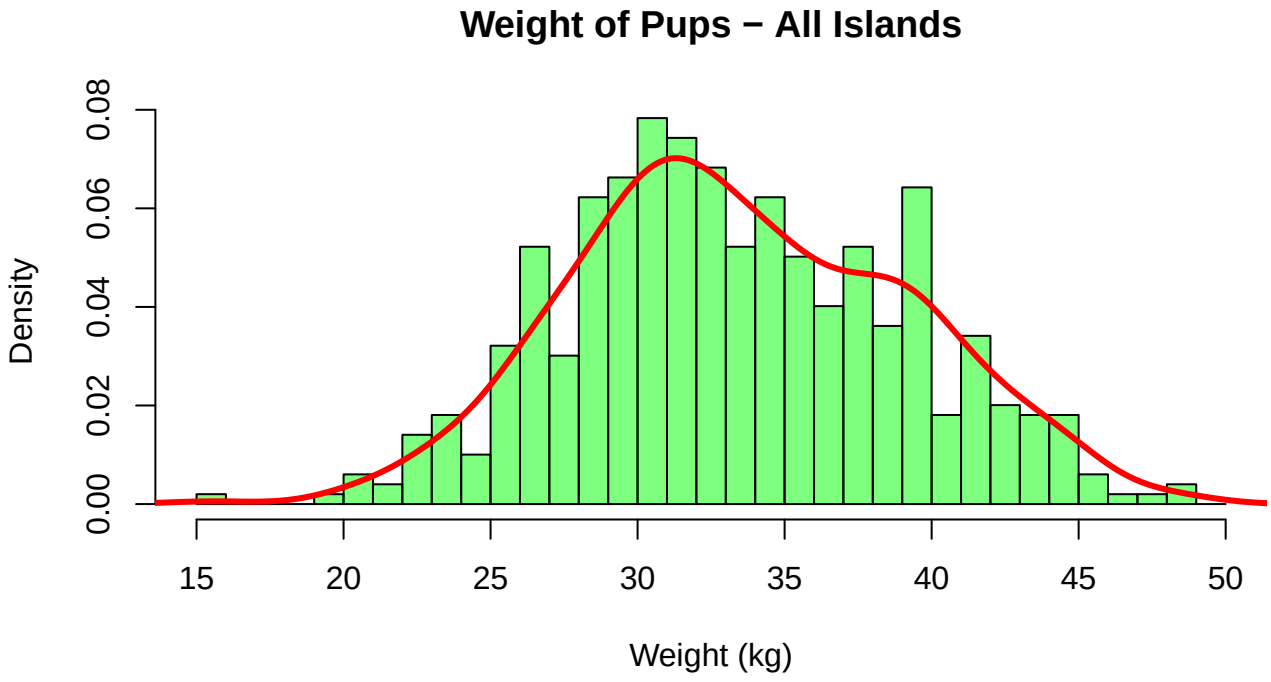
Note: the sum of the AREA of all the bars is equal to 1 $\sum_{i=1}^{n} D_i \times w = 1$ ... where w is the width of the bin

R code

```
> hist(P$Weight, freq=FALSE,
       main="Weight of Pups - All Islands", xlab="Weight (kg)")
```

Histogram of **densities**

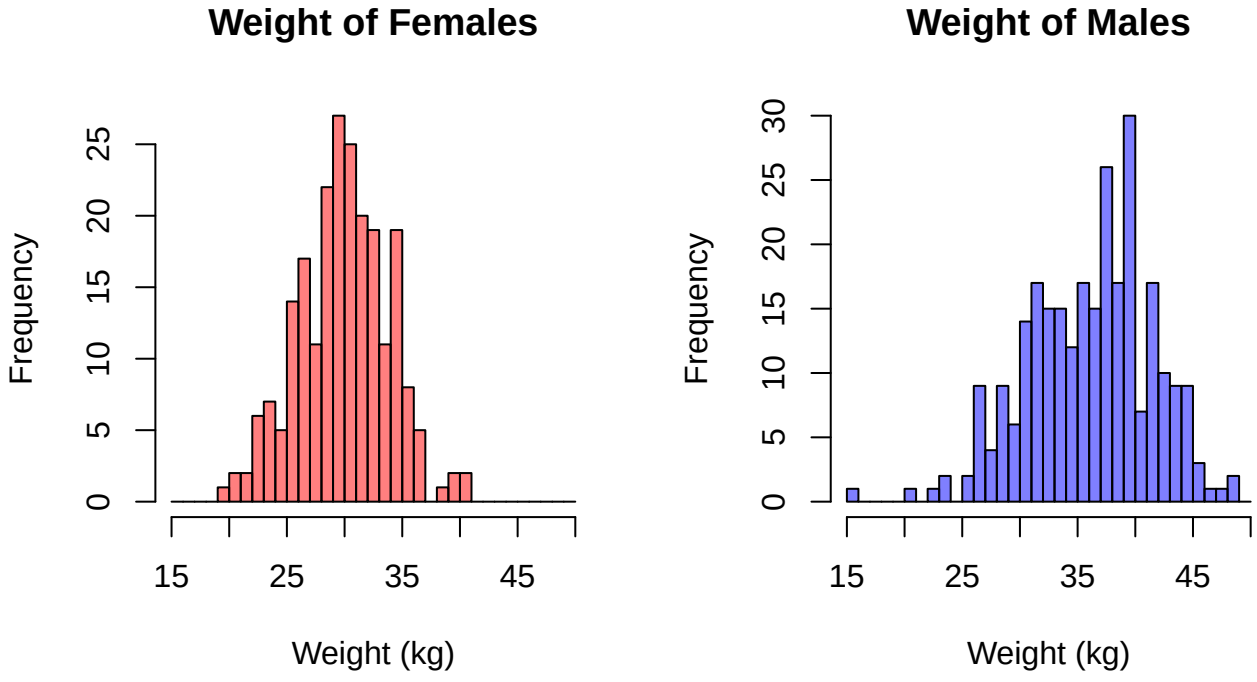
Histograms allows us visualize the **distribution** of the measurement



R code: density curve

```
 > W.density <- density(P$Weight)
> hist(P$Weight, freq=FALSE)
 > lines(W.density$x, W.density$y, col="red")
```

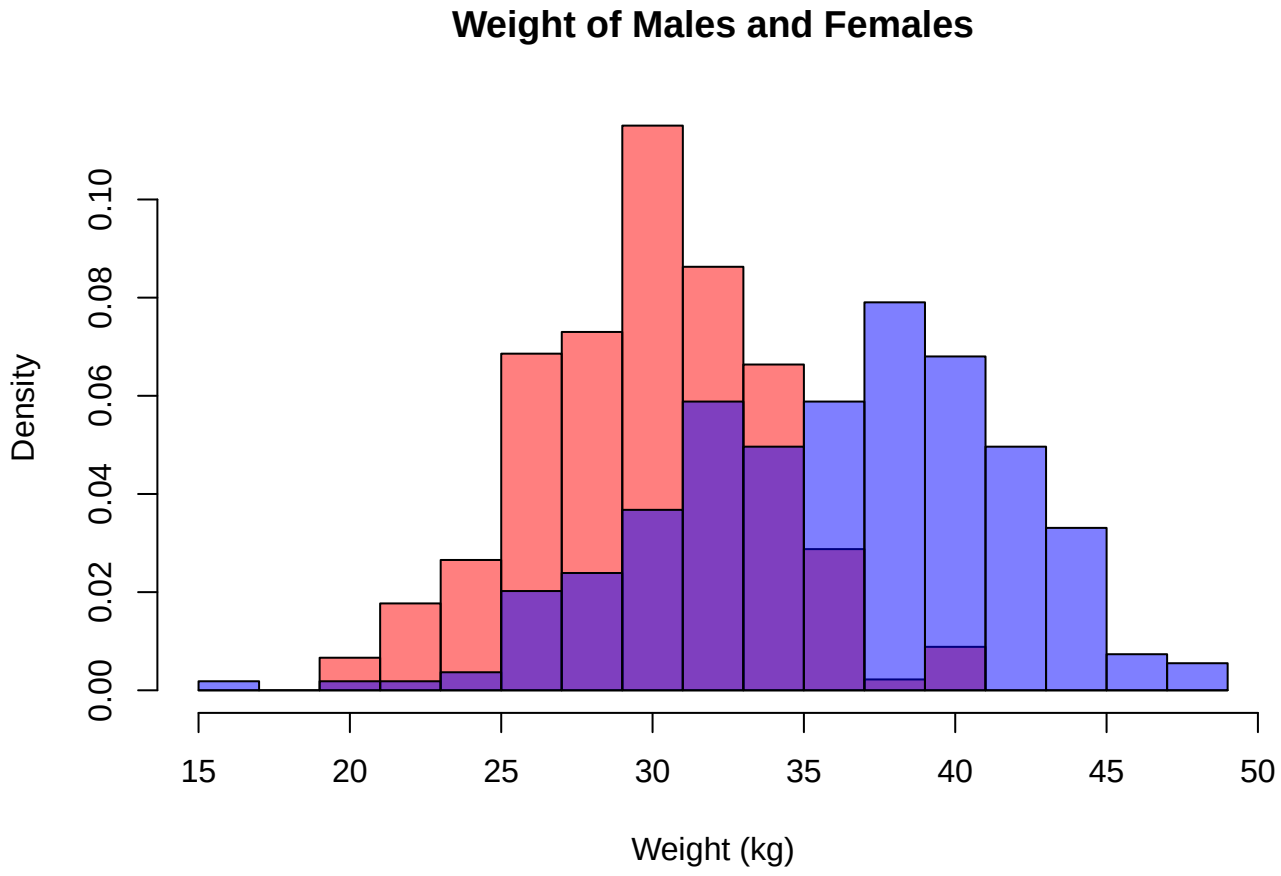
Two histograms



R code: multiple plots

```
> par(mfrow=c(1,2)) # splits graphics window into two screens
> hist(P$Weight[P$Sex=="F"], breaks=15:50, col=rgb(1,0,0,.5))
> hist(P$Weight[P$Sex=="M"], breaks=15:50, col=rgb(0,0,1,.5))
```

Joint histograms



R code: overlapping histograms

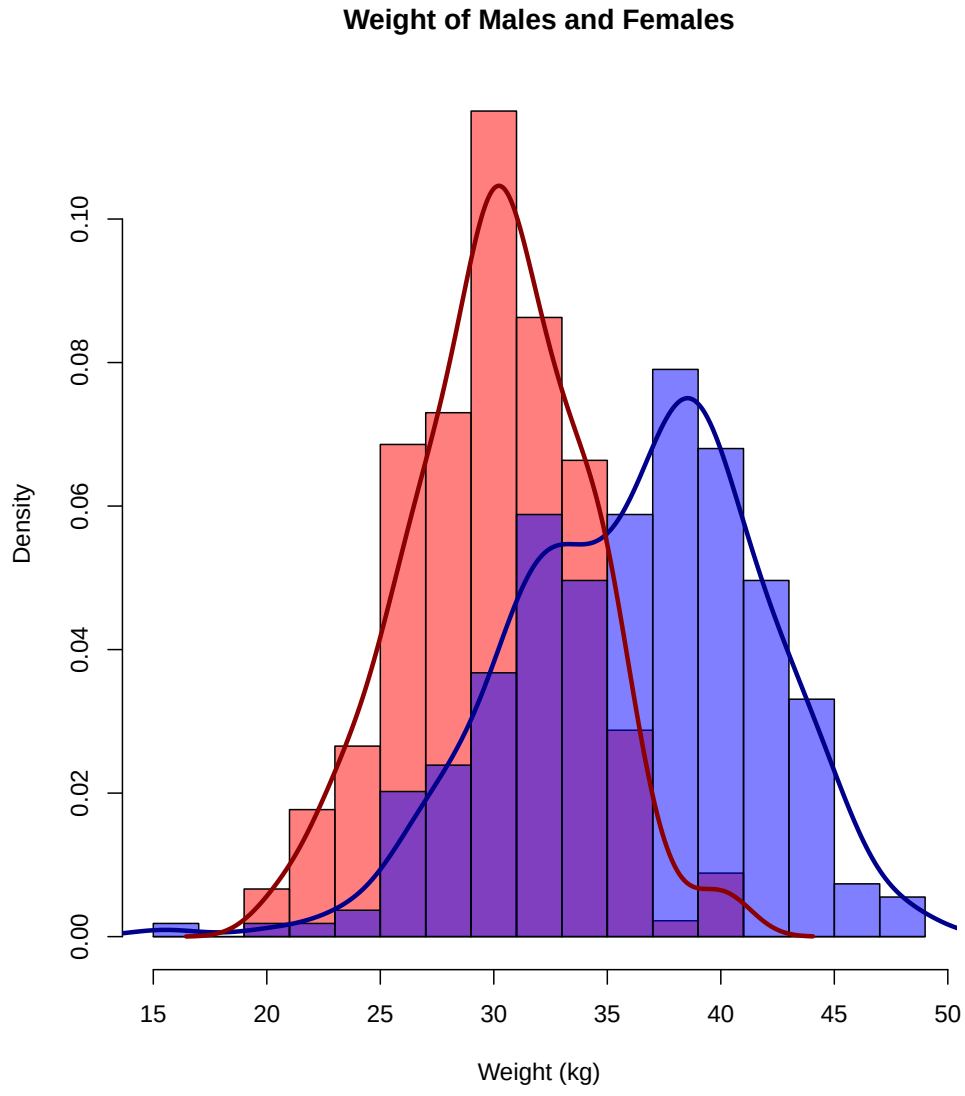
```
> hist(P$Weight[P$Sex=="F"], col=rgb(1,0,0,.5), freq=FALSE,
>      main="Weight of Males and Females", xlab="Weight (kg)")
> hist(P$Weight[P$Sex=="M"], col=rgb(0,0,1,.5), add=TRUE)
```

What can you learn from looking at a distributions?

Note overall patterns,

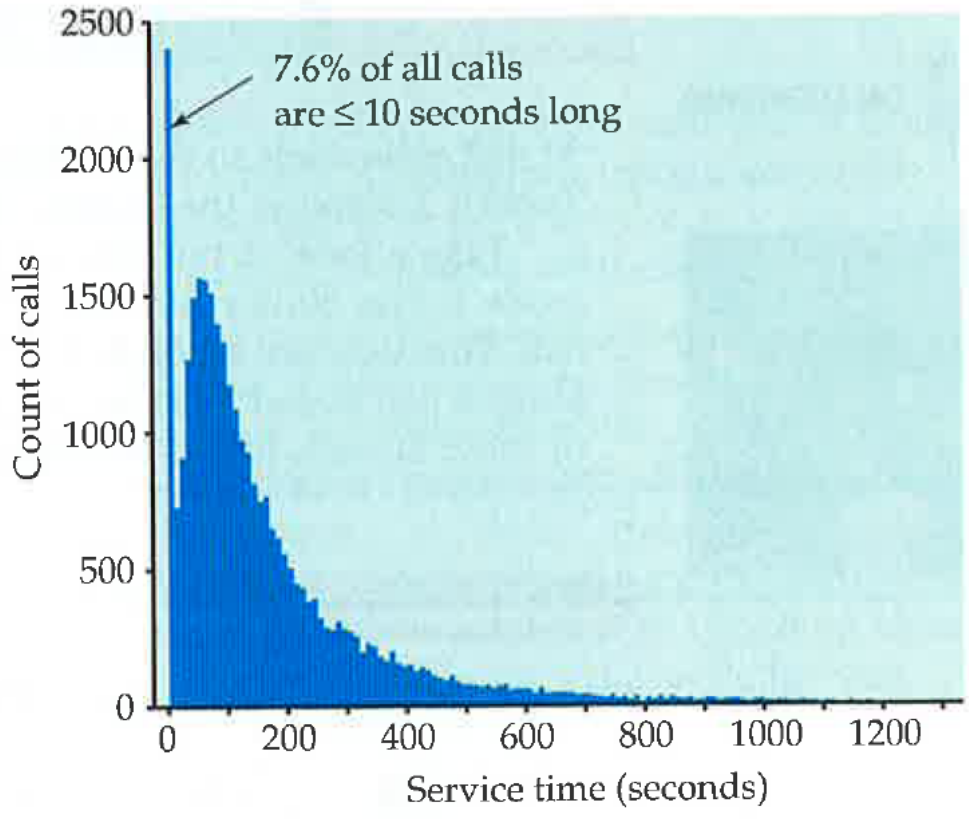
- **Range**
- **Center**
- **Spread**
- **Shape**

Identify **outliers** (e.g. One very small male)

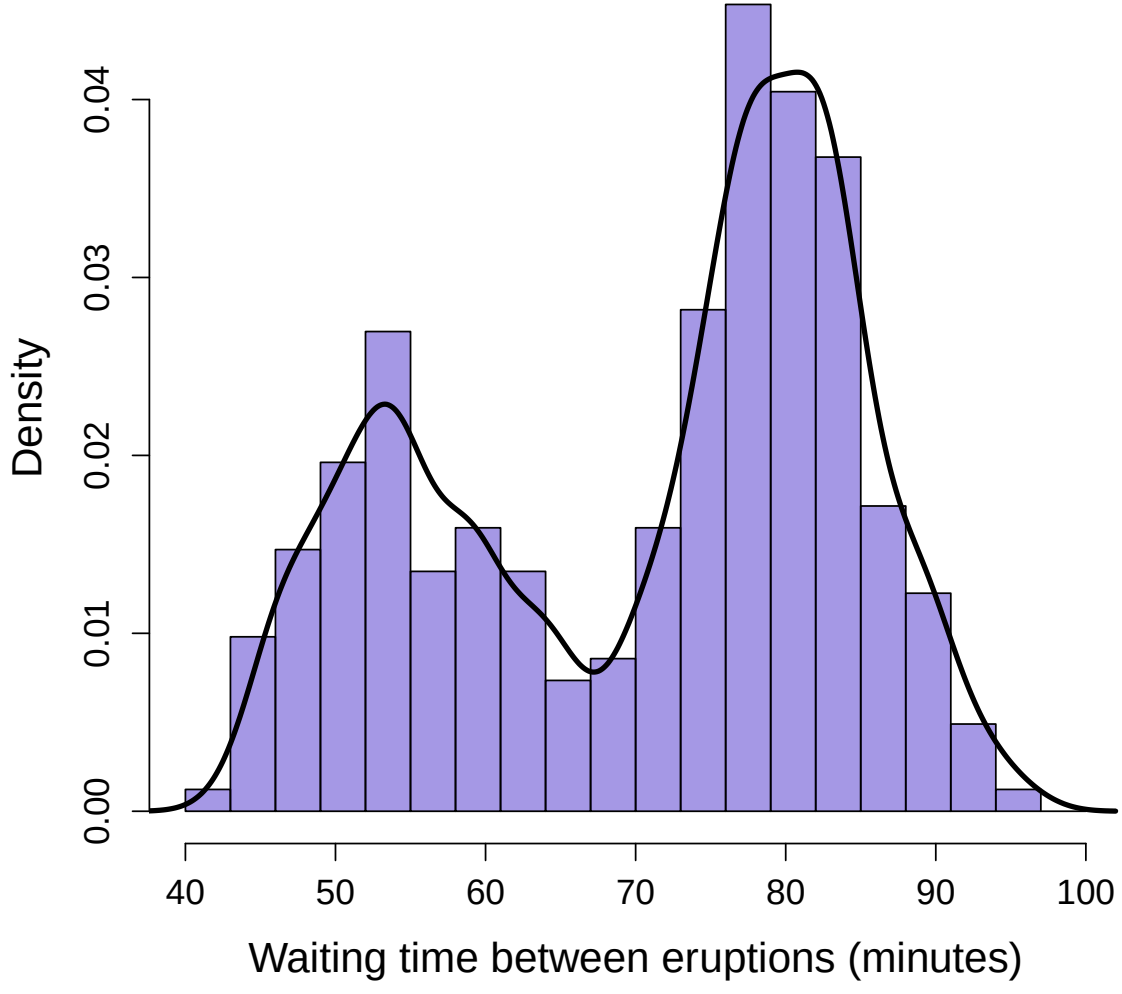


Example 1: Call center data

FIGURE 1.10 The distribution of call lengths for 31,492 calls to a bank's customer service center, for Example 1.15. The data show a surprising number of very short calls. These are mostly due to representatives deliberately hanging up in order to bring down their average call length.

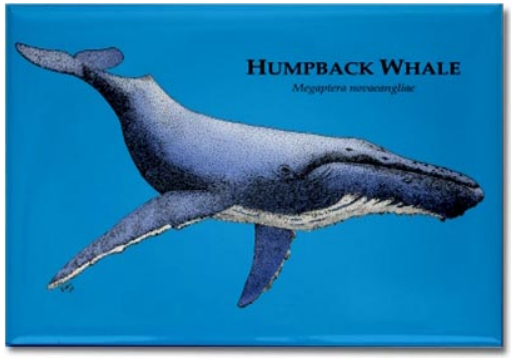


Example 2: How faithful is Old Faithful?

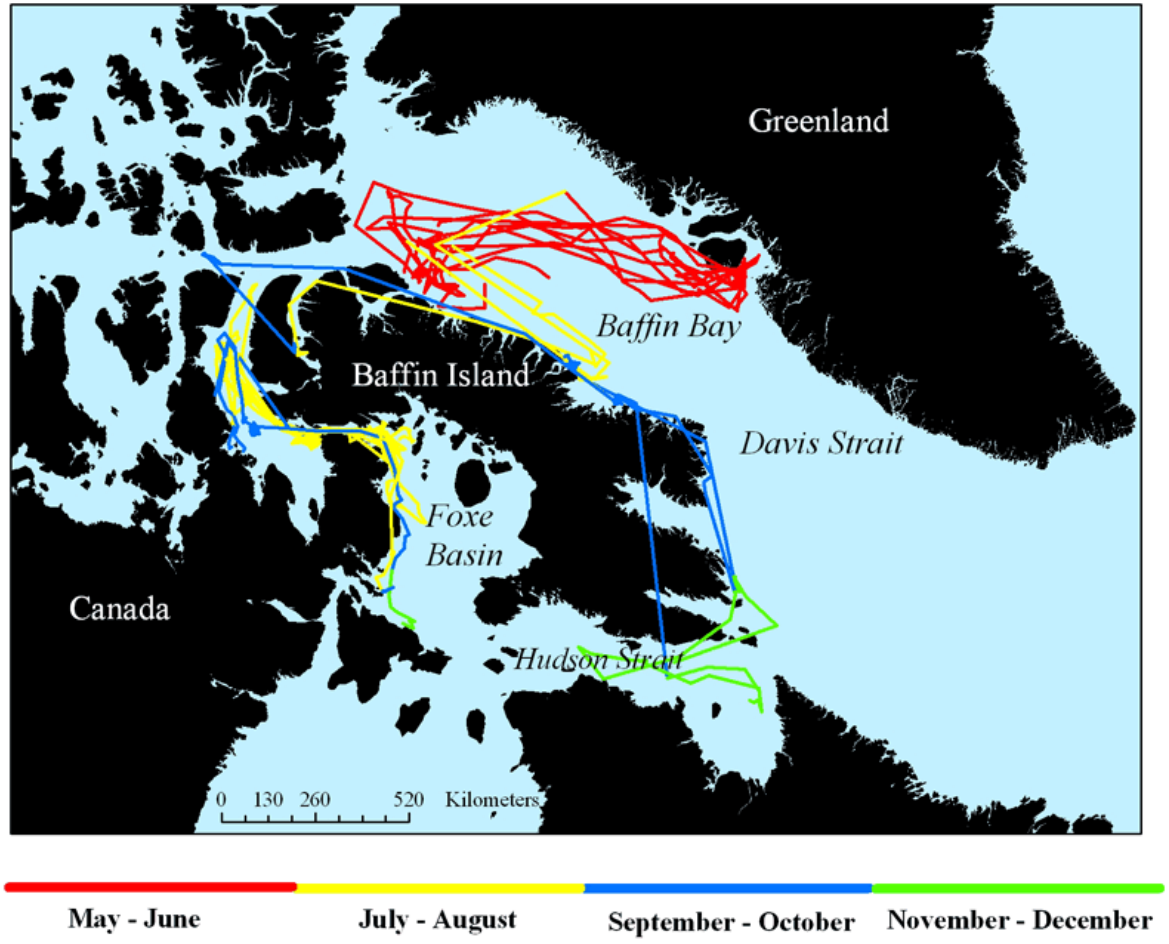


Example 3: How do whale species spatially segregate?

Bowhead whale



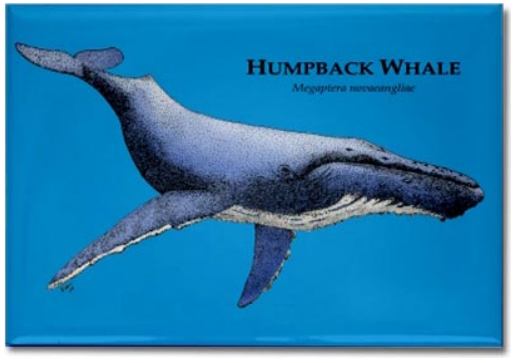
Humpback whale



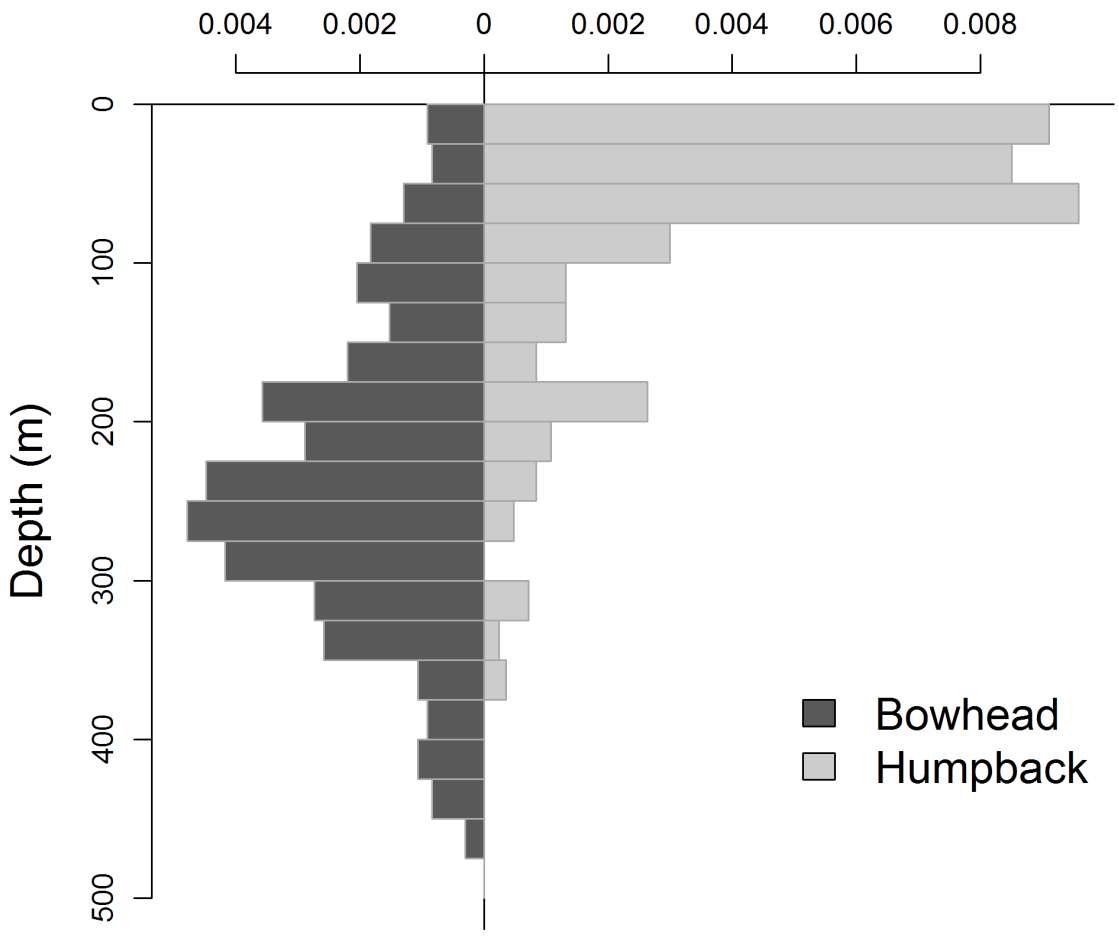
(Images thanks to K. Laidre and M. Villum)

Example 3: How do whale species spatially segregate?

Bowhead whale



Humpback whale

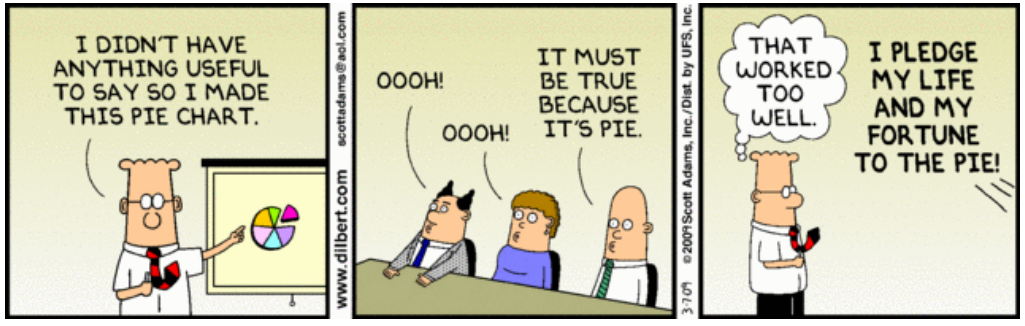


(Images thanks to K. Laidre and M. Villum)

Visualization

Take home message

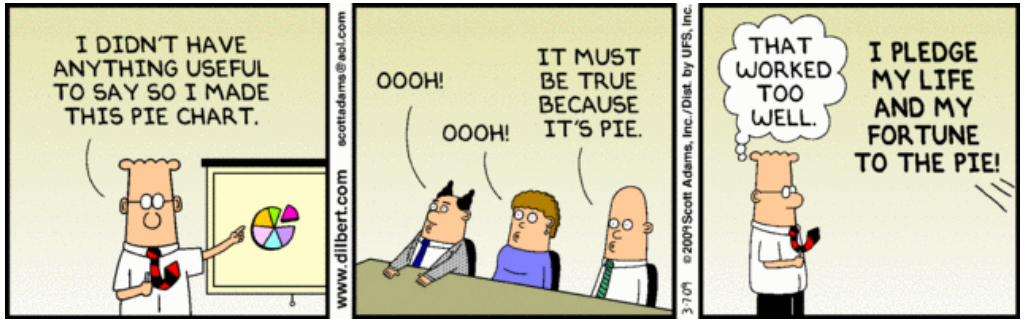
A picture is worth at least dozens of words! If it's a good one, you don't really *need* any fancy statistics.



Visualization

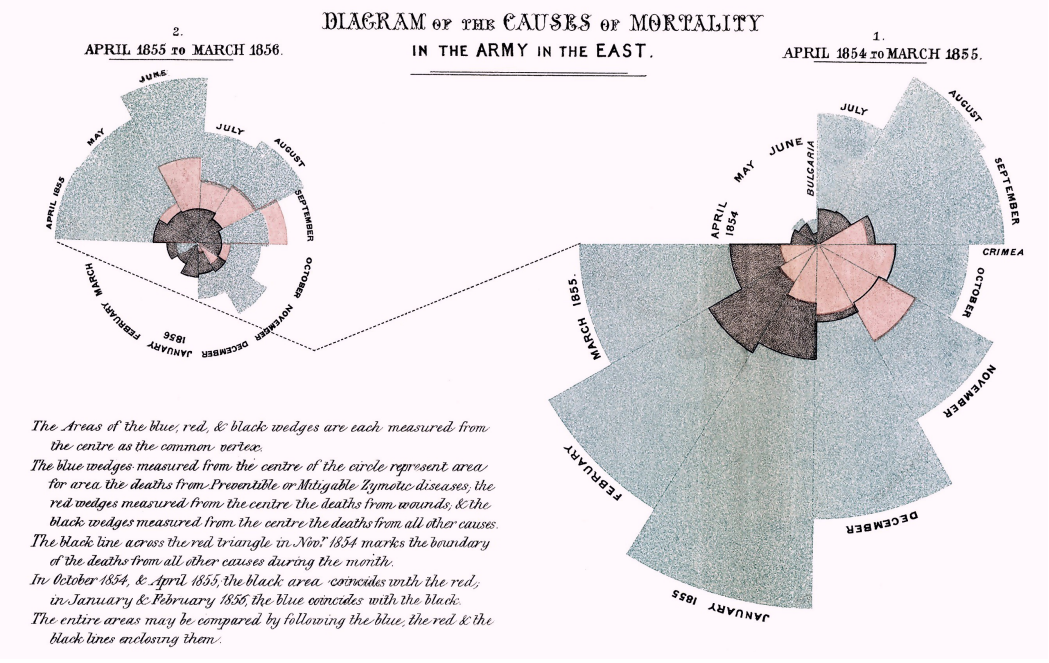
Take home message

A picture is worth at least dozens of words! If it's a good one, you don't really *need* any fancy statistics.



Historical aside on visual statistics

The mother of visual statistics was **Florence Nightingale** (1820-1910),



most famous as a **nurse** during several wars in the 19th century, and as the founder of the **Red Cross**. Among other tools, she invented the "rose-diagrams" (sometimes called "Nightingale rose") - a circular histogram for wrapped data. She used figures like these to convince bureaucrats in London about the importance of deploying nurses for managing preventable diseases in the battlefield.