# Lecture 2: Univariate Data

Thomas Chadefaux

PO7001: Quantitative Methods I

## Summarizing Categorical Data

- Categorical data has no ordering and no associated metric. E.g., Party, country, continent, religion, etc.

## Summarizing Categorical Data

- Categorical data has no ordering and no associated metric. E.g., Party, country, continent, religion, etc.
- In R, this data will typically appear as “factors”

# Summarizing Categorical Data

- Categorical data has no ordering and no associated metric. E.g., Party, country, continent, religion, etc.
- In R, this data will typically appear as “factors”
- e.g., we import the correlates of war data

## An Example from the CoW Data

```
url <- 'http://www.correlatesofwar.org/data-sets/COW-war/i
cow <- read.csv(url)
head(cow$StateName)
```

```
## [1] Spain                    France                   O
## [4] Russia                   Mexico                   U
## 105 Levels: Afghanistan Angola Argentina Armenia Austra
```

```
class(cow$StateName)
```

```
## [1] "factor"
```

```
levels(cow$StateName)
```

```
##   [1] "Afghanistan"                       "Angola"
##   [3] "Argentina"                         "Armenia"
##   [5] "Australia"                         "Austria"
##   [7] "Austria-Hungary"                   "Azerbaijan"
```

# Summarizing Categorial Data

- Typically using a table
- E.g.:

```
table(cow$StateName)
```

# Now that's not very pretty, is it?

| Outcome | Frequency |
|---|---:|
| 1 | 155.00 |
| 2 | 119.00 |
| 3 | 4.00 |
| 4 | 28.00 |
| 6 | 30.00 |
| 8 | 1.00 |

**Table 1:** Frequency Distribution of war outcomes

# You can convert data from one type to another

```
table(cow$Outcome)
```

```
##
##   1   2   3   4   6   8
## 155 119   4  28  30   1
```

```
cow$Outcomef <- factor(cow$Outcome,
                       labels=c("Winner", "Loser", "Tied",
                                "Different type",  "Stalema
```

| Outcome | Frequency |
|---|---|
| Winner | 155.00 |
| Loser | 119.00 |
| Tied | 4.00 |
| Different type | 28.00 |
| Stalemate | 30.00 |
| Changed sides | 1.00 |

## Plotting Categorical Variables: The Barplot

- Very intuitive and common way of representing data

## Plotting Categorical Variables: The Barplot

- Very intuitive and common way of representing data
- The height (or length if it is horizontal) of the bar corresponds to the frequency of a given category

# Plotting Categorical Variables: The Barplot

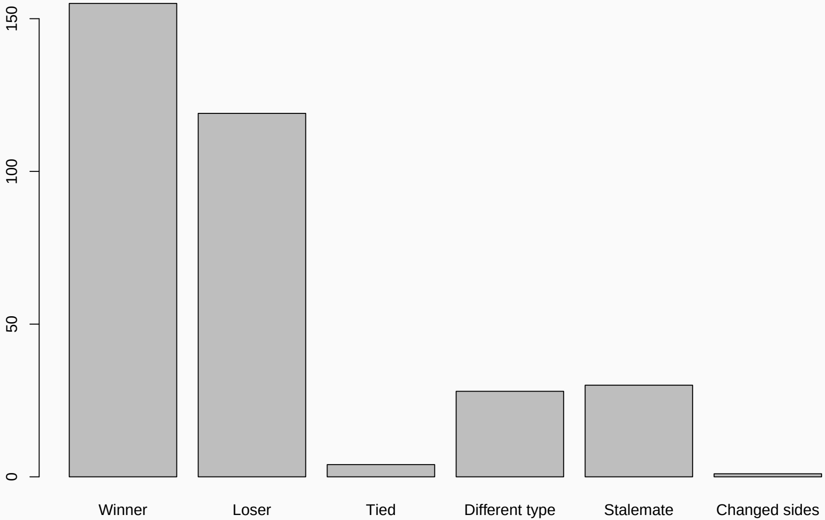
- Very intuitive and common way of representing data
- The height (or length if it is horizontal) of the bar corresponds to the frequency of a given category
- With some exceptions, height of bars should start at 0. Why?

# Plotting Categorical Variables: The Barplot

- Very intuitive and common way of representing data
- The height (or length if it is horizontal) of the bar corresponds to the frequency of a given category
- With some exceptions, height of bars should start at 0. Why?
- But sometimes rules need to be broken. . .

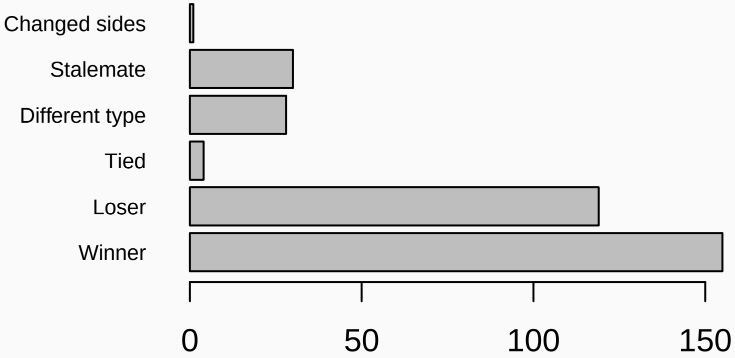
# The Barplot (cont'd)

```
barplot(table(cow$Outcomef))
```



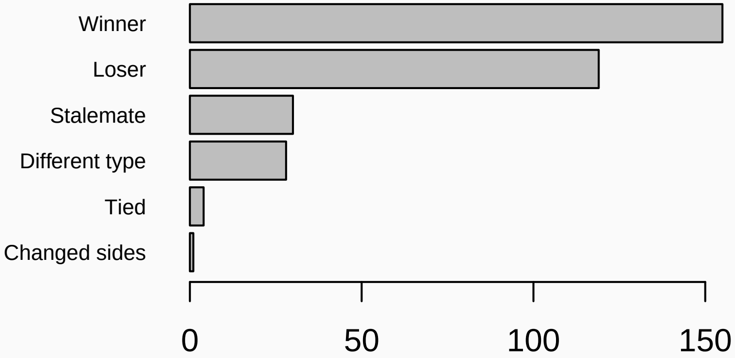
# Horizontal Barplot (same thing)

```
par(mar=c(3,5,2,1))
barplot(table(cow$Outcomef),
        horiz=TRUE,
        las=1,
        cex.names=0.7)
```



# Horizontal Barplot, ordered

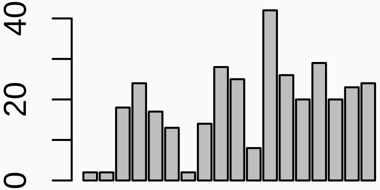
```
par(mar=c(3,5,2,1))
barplot(sort(table(cow$Outcomef)),
        horiz=TRUE,
        las=1,
        cex.names=0.7)
```

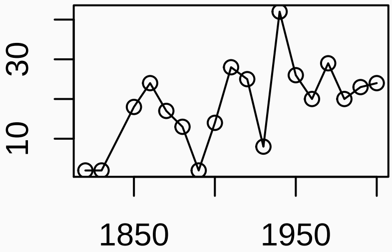


## Barplots are not the best for time series

- For example, let us calculate the number of wars per decade:

```
par(mfrow = c(1,2), mar=c(2,2,1,1)) # tells R to print 2 p
cow$decade <- round(cow$StartYear1/10)*10
wars.by.year <- aggregate(cow$WarName,
                          by=list(cow$decade),
                          FUN = length)
barplot(wars.by.year$x)
plot(wars.by.year, type='o')
```

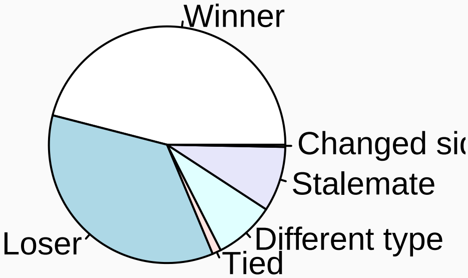


# Pie charts

- I rarely, if ever, see these graphs in publications. They don't look professional and are not particularly useful. If you insist on using them, though:

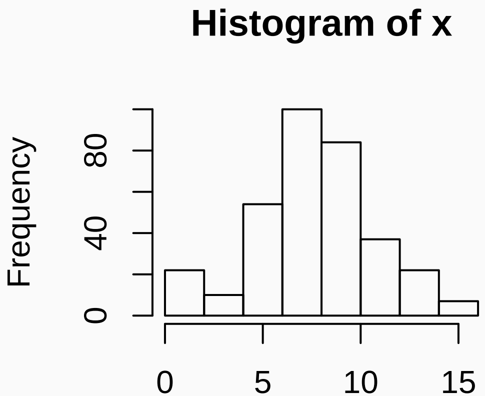
```
pie(table(cow$Outcomef))
```



## Histograms

A graphical display of tabulated frequencies shown as bars, showing the proportion of cases that fall into non-overlapping intervals of a variable

```
x <- log1p(cow$BatDeath)
hist(x)
```

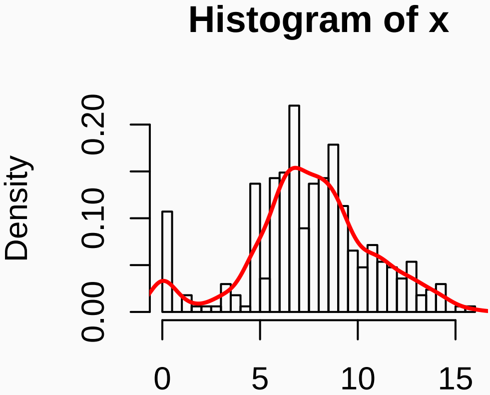


# Histograms with density

```r
x <- log1p(cow$BatDeath)
```

```
## Warning in log1p(cow$BatDeath): NaNs produced
```

```r
hist(x, breaks=50, freq = FALSE)
lines(density(x, na.rm = TRUE), col=2, lwd=2)
```

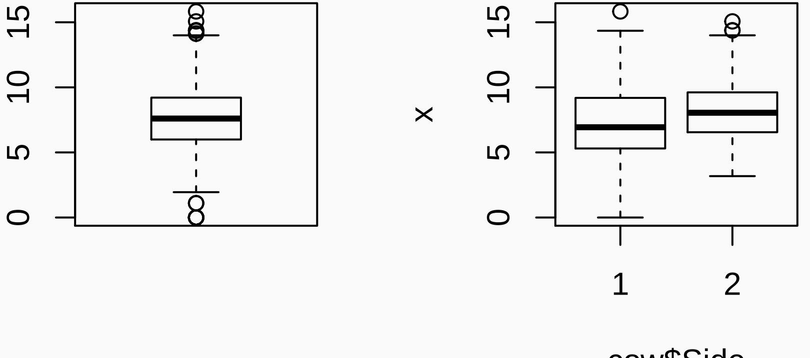


# Boxplots

```r
par(mfrow=c(1,2))
x <- log1p(cow$BatDeath)
```

```
## Warning in log1p(cow$BatDeath): NaNs produced
```

```r
boxplot(x)
boxplot(x ~ cow$Side)
```



# Measures of Central Tendency

## Measures of Central Tendency

- Central Tendency: a single number that characterizes the "typical" unit in a set of data
- Several measures:
  - Mode
  - Median
  - Mean
- Choose depending on nature of data, what you need to convey, and the distribution of the data

## The Mode

- The most *frequently* occurring value in a distribution. I,e, the category with the largest frequency.
- E.g.:
  - The mode of $\{1, 2, 1, 3, 4, 5\}$ is one
  - The mode of {Republican, Republican, Democrat, Republican, Libertarian} is Republican

# The mode in R

- Unfortunately, 'mode' does not work as expected:

```
mode(cow$Outcomef)
```

```
## [1] "numeric"
```

- Luckily it's easy enough from the table:

```
table(cow$Outcomef)
```

```
##
##          Winner              Loser               Tied Different t
##             155                119                  4
##   Changed sides
##               1
```

- Or, if you're lazy/have too many categories:

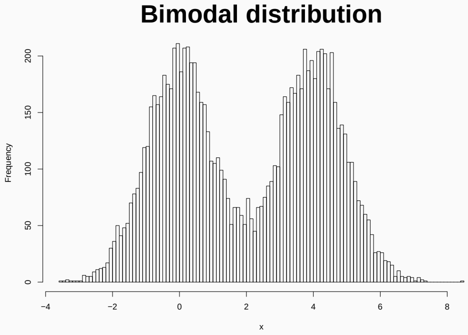
```
which(table(cow$Outcomef) == max(table(cow$Outcomef)))
```

## Multiple modes

- There may be more than one mode in a sample
- For example, $\{1,2,3,1,2,4,5\}$ has two modes: 1 and 2. The uniform distribution has an infinity of modes
- In pratice, large datasets make it unlikely that you have exactly two modes.

## Bimodal Distributions

A *distribution* has two modes even if one of the modes is smaller than the others. What we mean then is that these are local maxima. E.g.:



# The median

- The median divides the sample in two groups of equal size. So 50% of the data will be below the median, 50% will be above.
- Find the median by ordering the data and looking for the $(N+1)/2$ point.
  - e.g.: median of $\{1,2,3,4,5\}$ is 3
  - e.g.: median of $\{1,2,3,4,5,6\}$ is 3.5
- In R:

```r
x <- c(1,2,3,4,5,6)
median(x)
```

```
## [1] 3.5
```

# The Mean (arithmetic)

- Same thing as the *average*
- Often written as $\bar{X}$ or $\mu$
- Calculated as $\frac{1}{N}\sum_{i=1}^{N} x_i$

```
mean(c(1,2,3,4,5))
```

```
## [1] 3
```

```
mean(c(1,2,3,4,5, 1000))
```

```
## [1] 169.1667
```

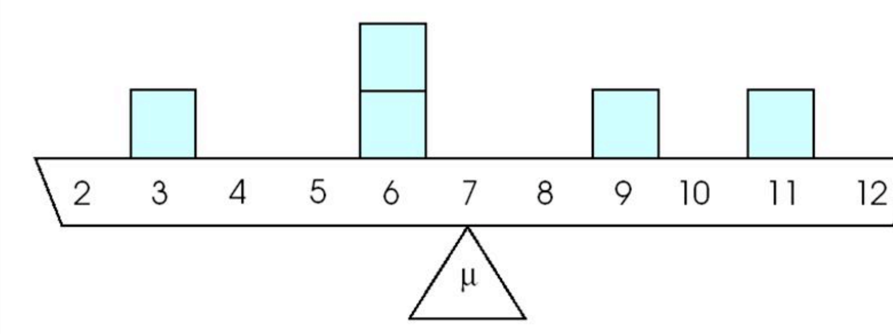
```
mean(c(1,2,3,4,5, 1000, NA))
```

```
## [1] NA
```

```
mean(c(1,2,3,4,5, 1000, NA), na.rm=TRUE)
```

```
## [1] 169.1667
```

# Mean as a center of gravity



# An aside on summation signs

- $\sum_{i=1}^{N} x_i = x_1 + x_2 + x_3 + \ldots + x_n$

# An aside on summation signs

- $\sum_{i=1}^{N} x_i = x_1 + x_2 + x_3 + \ldots + x_n$
- $\sum_{i=1}^{N} 1 = N$

# An aside on summation signs

- $\sum_{i=1}^{N} x_i = x_1 + x_2 + x_3 + \ldots + x_n$
- $\sum_{i=1}^{N} 1 = N$
- $\sum_{i=4}^{6} \frac{1}{i} = \frac{1}{4} + \frac{1}{5} + \frac{1}{6}$

# In class exercise

- Write a function called ‘mymean’, which will take a vector of numbers and return the mean

## In class exercise

- Write a function called ‘mymean’, which will take a vector of numbers and return the mean
- Write a function called ‘mymedian’, which will take a vector of numbers and return the median (much more difficult)

## In class exercise

- Write a function called 'mymean', which will take a vector of numbers and return the mean
- Write a function called 'mymedian', which will take a vector of numbers and return the median (much more difficult)
- Create a function that will report both the mean and the median

# In class exercise

- Write a function called mymean', which will take a vector of numbers and return the mean, without actually using themean' function

```
mymean <- function(x){
     return(sum(x)/length(x))
}
mymean(1:10)
```

```
## [1] 5.5
```

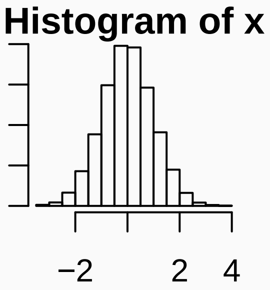
```
mymeanAndMedian <- function(x){
     this.mean <- sum(x)/length(x)
     midpoint <- (length(x)+1)/2
     if(length(x)%%2!=0){ #we have an odd number of observ
          this.median <- sort(x)[midpoint]
     }
```

# Weighted mean

- E.g., to calculate grades with different weights
- Or surveys to count observations differently
- $\bar{X}_{weighted} = \sum_i w_i X_i$

# Mean, Median, and skewness

```
par(mar=c(2,1,1,1))
x <- rnorm(10000) # Symmetric distribution (standard norma
hist(x)
```

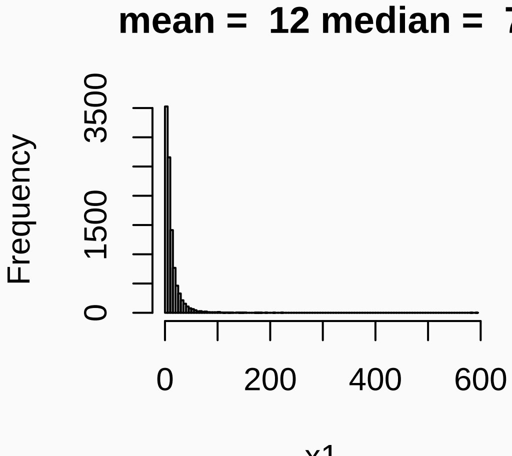



```
mean(x)
```

```
## [1] 0.003090373
```

```
median(x)
```

```r
# nonsymmetric distribution (lognormal here for illustratio
x1 = rlnorm(10000, 2, 1)
hist(x1,
     main=paste('mean = ', round(mean(x1)),
                'median = ', round(median(x1))),
     breaks=100)
```



# The range

- Simply the difference between largest and smallest observation
- I.e., range $= max(x) - min(x)$
- Dependent on extreme values

# Percentiles

- The percentage of the data that is below a certain level.
- E.g., the 5th percentile means that 5% of the data is below that level
- Given an ordered variable with 100 observations, the $x^{th}$ percentile is simply the $x^{th}$ value
- Some percentiles have special designations:
  - the 25th percentile is the 1st *quartile*
  - the 50th percentile is the median
  - the 75th percentile is the 3rd *quartile*
  - deciles refer to every 10th percentile. E.g., 9th decile is the 90th percentile

# Percentiles in R

```
x <- rnorm(1000)
# 25th percentiles
quantile(x, 0.25)
```

```
##        25%
## -0.6356508
```

```
quantile(x, 0.5) == median(x)
```

```
##  50%
## TRUE
```

```
quantile(x, probs = seq(0,1,0.1))
```

```
##          0%         10%         20%         30%
## -2.89082816 -1.18155106 -0.81261117 -0.49673094 -0.2122
##         60%         70%         80%         90%
```

# Interquartile Range

- The difference between the 3rd and 1st quartile

```
x <- rnorm(1000)
IQR(x)
```

```
## [1] 1.352439
```

```
quantile(x, 0.75) - quantile(x, 0.25) # same thing
```

```
##      75%
## 1.352439
```

# Variance

- $Var(x) = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$

# Variance

- $Var(x) = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$
- Note that the sample variance is often calculated as

$$\hat{Var}(x) = s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

# Variance

- $Var(x) = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$
- Note that the sample variance is often calculated as

$$\hat{Var}(x) = s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

  - Why $N-1$? Because that is an unbiased estimate of the population variance. No need to worry too much about why, but do remember it...

# Variance

- $Var(x) = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$
- Note that the sample variance is often calculated as

$$\hat{Var}(x) = s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

  - Why $N-1$? Because that is an unbiased estimate of the population variance. No need to worry too much about why, but do remember it...
- In R, very simple: var(x)

# Standard Deviation

- Simply the square root of the variance
- sd(x) $= \sigma = \sqrt{\sigma^2}$
- *sample* standard deviation is denoted by $s$

# A word about logs

- $log_{10}(10) = 1$
- $log_{10}(100) = 2$
- $log_{10}(1000) = 3$
- $ln(2.718) = 1 = log_e(2.718)$
- $ln(100) = 4.6$
- $ln(1000) = 6.9$

# Recommended:


- Tufte, Edward R. The visual display of quantitative information. Cheshire, CT: Graphics press, 1983. (esp. ch. 6)