STA130 W2

Monday, October 28, 2019 9:34 PM

Typical Value:
The **mean** is a common way to measure the **center of a distribution** of numerical data.

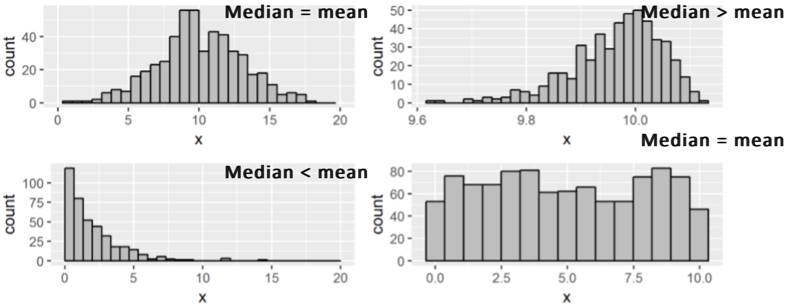
- The average
- Captures the contribution of **extreme values**

The **median** is another way to measure the **center** of a numerical variable.

- The value such that **50%** of the data are less than and 50% are greater than it.
- Rank the values from smallest to largest
- Less affected by extreme values

*When its **bimodal**, neither of them are good value
The **mode** is the most **frequent** value in a dataset

- Not necessary in the center, but better for talking about the shape
- **Not typical value**



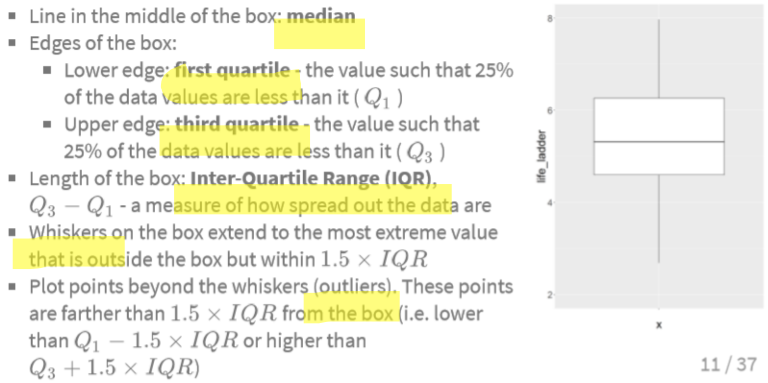
<u>Numerical summaries of **the spread of a distribution:**</u>
The **variance** is roughly **the average squared distance** from the mean.
The **standard deviation** is the **square root of the variance**.

- Unlike the variance, it is measured **in the same units as the data** so is easier to interpret.
- Small standard deviation/variance means that on average the data is **close to the mean**
- Large standard deviation means **further away from the mean**(more spread out)

<u>Visualizing **summary of center / spread: boxplots:**</u>

```
ggplot(data=happinessdata_2017,
       aes(x=" ", y=life_ladder)) +
  geom_boxplot()
```

A boxplot summarizes the distribution of a quantitative (numerical) variable using **five statistics**, while also plotting **unusual observations** (outliers).



- Line in the middle of the box: **median**
- Edges of the box:
  - Lower edge: **first quartile** - the value such that 25% of the data values are less than it (Q_1)
  - Upper edge: **third quartile** - the value such that 25% of the data values are less than it (Q_3)
- Length of the box: **Inter-Quartile Range (IQR)**, $Q_3 - Q_1$ - a measure of how spread out the data are
- Whiskers on the box extend to the most extreme value that is outside the box but within $1.5 \times IQR$
- Plot points beyond the whiskers (outliers). These points are farther than $1.5 \times IQR$ from the box (i.e. lower than $Q_1 - 1.5 \times IQR$ or higher than $Q_3 + 1.5 \times IQR$)

11 / 37

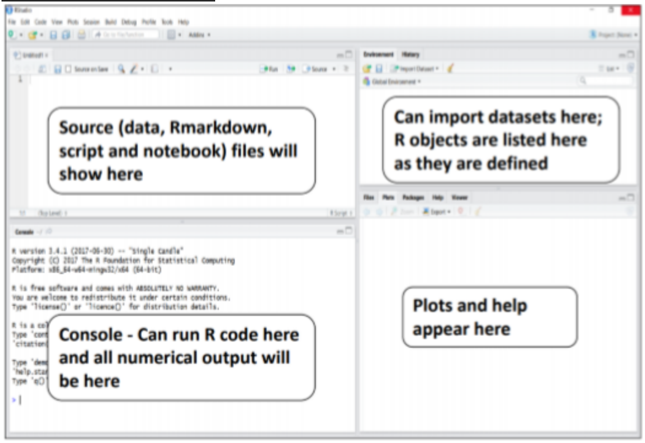
<u>Compare distributions of a quantitative variable **across groups:**</u>
*Association between a **categorical(x)** and a **numerical(y)** variables

```
ggplot(data=happinessdata_2017,
       aes(x=continent, y=life_ladder)) +
  geom_boxplot() +
 coord_flip()
```

If a distribution is **symmetrical**:

- The median will be in the **middle** of the box
- The whiskers will be the **same length**

Rstudio User Interference:



Source (data, Rmarkdown, script and notebook) files will show here

Can import datasets here; R objects are listed here as they are defined

```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a co
Type 'con
'citation

Type 'dem
'help.sta
Type 'q()'

> |
```

Console - Can run R code here and all numerical output will be here

Plots and help appear here

Use console (bottom left window) as a calculator:

+ - * / ^

Saving R objects:

R lets you **save data** by storing it inside an "R object"

An **R object** is a name that you can use to call up stored data

```
x <- 1
x
## [1] 1
```

When you create an object, it will be **listed in the environment pane** (top right)

Atomic vectors:

Vectors are the simplest data structure in R.

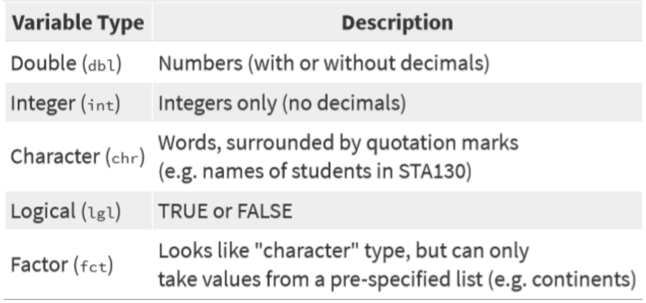
Make an atomic vector by **grouping some values of data together** with c()

The **c() function** combines elements of one type into a vector

A 6-sided die:

```
die <- c(1, 2, 3, 4, 5, 6) die
## [1] 1 2 3 4 5 6
is.vector(die)
## [1] TRUE
length(die)
## [1] 6
```

Types of variables in R:

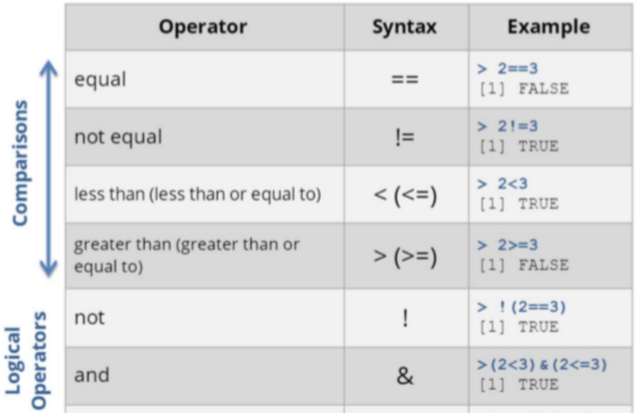


Variable Type	Description
Double (dbl)	Numbers (with or without decimals)
Integer (int)	Integers only (no decimals)
Character (chr)	Words, surrounded by quotation marks (e.g. names of students in STA130)
Logical (lgl)	TRUE or FALSE
Factor (fct)	Looks like "character" type, but can only take values from a pre-specified list (e.g. continents)

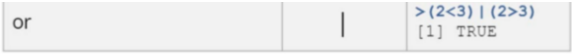
Each atomic vector can store only **one** type of data

Use **is.** functions (e.g. is.numeric(), is.character()) to check the data type of a vector

Logicals:



	Operator	Syntax	Example
Comparisons	equal	==	> 2==3 [1] FALSE
	not equal	!=	> 2!=3 [1] TRUE
	less than (less than or equal to)	< (<=)	> 2<3 [1] TRUE
	greater than (greater than or equal to)	> (>=)	> 2>=3 [1] FALSE
Logical Operators	not	!	> !(2==3) [1] TRUE
	and	&	>(2<3) & (2<=3) [1] TRUE

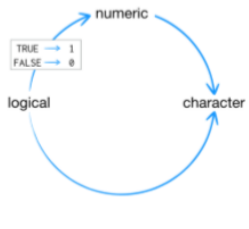


or	|	> (2<3) | (2>3) [1] TRUE

Coercion:

R switches between data types automatically for certain operations.

I.e. sum(c(TRUE, FALSE)) becomes sum(c(1,0)) which counts the number of values of TRUE in a vector



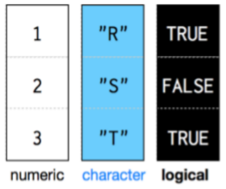
Command	Output
3 + "2"	Error
c(1, "2")	"1""2"
c(TRUE, "FALSE")	"TRUE""FALSE"
sum(c(TRUE, TRUE, TRUE))	3
sum(c(FALSE, FALSE, FALSE))	0
sum(c(10 == 5*2, 2 != 3, 2 <= 1.5*2))	3

Data Frames:

An R data frame is used for storing data sets (similar to Excel spreadsheets)

- rows: individual observations/records
- columns: variables Each column of a data frame can contain a different type of data

Within a column, every cell must be the **same type of data**.



1	"R"	TRUE
2	"S"	FALSE
3	"T"	TRUE
numeric	character	logical

Access/create data frames in R:

1-Download & open a package to access a data frame which is included in the package

2-Import a data frame from an external file (e.g. Excel file) and save it as a **dataframe** object in R

- Using the **read_csv()** or **read excel()** functions, as with the happiness data last week

After you have loaded it, you can view a data frame in RStudio by clicking on the data frame name in the Environment tab (top right corner)

Built-in functions:

round(-2.718282, digits = 2)
## [1] -2.72 	
abs(-2.718282)
## [1] 2.718282 	
data <- c(1,2,3,4,5,6)
	length(data)
	## [1] 6
	mean(data)
	## [1] 3.5
	median(data)
	## [1] 3.5
	round(sd(data), digits = 1)
	## [1] 1.9

Built-in **help** documentation on R functions: Type ?round in the R console window

```
glimpse(AutoClaims)
## Observations: 6,773
## Variables: 5
## $ STATE  <fct> STATE 14, STATE 15, STATE 15, STATE 15, STATE 15, STATE...
## $ CLASS  <fct> C6 , C6 , C11, F6 , F6 , F6 , C11, C6 , C11, C11, C6 , ...
## $ GENDER <fct> M, M, M, F, M, M, M, M, M, M, M, M, M, F, F, F, M, F...
## $ AGE    <int> 97, 96, 95, 95, 95, 95, 94, 94, 93, 93, 93, 93, 92, 92,...
## $ PAID   <dbl> 1134.44, 3761.24, 7842.31, 2384.67, 650.00, 391.12, 377...
```

fct: factors
int: integers
dbl: doubles

Using the **summarise**(in tidyverse) function:

summarise(AutoClaims, mean = mean(PAID),
median = median(PAID),
sd = sd(PAID),
min = min(PAID),
max = max(PAID))
	Columns in our summary table
	PAID is the $ value of the claim

```
##      mean    median     sd     min   max
## 1 1853.035 1001.7 2646.909 9.5 60000
```

*1 output, grouped data

Using the **group_by** function with summarise:

AutoClaims_grpGender <- group_by(AutoClaims, GENDER) *what variable to group by

summarise(AutoClaims_grpGender,
n = n(),
mean = mean(PAID),
median = median(PAID),
sd = sd(PAID))
	1 row per group

```
## # A tibble: 2 x 5
##   GENDER    n    mean  median  sd
##   <fct>    <int>   <dbl>   <dbl>  <dbl>
## 1 F      2582     1864.   963. 2761.
## 2 M       4191     1847.  1032. 2575.
```

x1 = sum(PAID > 5000))
x2 = sum(PAID > 5000) / n)
x3 = mean(PAID > 5000))
	X1: The number of people with claims larger than 5K
	X2 & X3: The proportion of people with claims larger than 5K

Vocabulary/ terms:

- Mean, average
- Median
- Standard deviation
- Variance
- Boxplot
- Interquartile range
- Quartile
- Outlier
- R object
- Vector
- Types of variables: e.g. character, numeric, logical
- Data frame
- Summary table, summary statistics
- Proportion