The University of Texas at Austin
School of Information

Statistics: Intro

Mick McQuaid

2024-01-11

Week ONE

uncertainty

Variability in data

- Your GPA fluctuates
- Your gas mileage fluctuates
- The cost of basic necessities fluctuates

Summarizing data, 1 of 2

We use the mean and standard deviation to summarize many data items

```
1 u ← c(1,2,3,4,5)
2 mean(u)
```

```
[1] 3
```

```
1 sd(u)
```

```
[1] 1.581139
```

Summarizing data, 2 of 2

```
1 v ← c(2,3,3,3,4)
2 mean(v)
```

```
[1] 3
```

```
1 sd(v)
```

```
[1] 0.7071068
```

v is less variable than *u*, even though they are the same on average. We need the standard deviation to know that, although we will later learn some pictures we can draw to illustrate it.

Let's do some examples on the computer

- Install R (just google the letter R)
- Install R Studio (after installing R)
- Be sure you install these on your local disk, not in the cloud

Now, recreate the above examples

- *u* and *v* are vectors and can be any size you like
- try using different numbers
- try using prices at different vendors for something you might want to buy or sell
- `c()` is a function that *combines* numbers or words into vectors
- The symbol ← can be read as the word *gets*, as in "*u* gets the vector of numbers 1 through 5"

Bundles of vectors called dataframes

- R has built-in dataframes
- One is called `mtcars`
- Type the word `mtcars` into the R console and press Enter to see it
- You can find the mean and standard deviation of any given column by saying, e.g., `mean(mtcars$mpg)`
- The part that says `mtcars$` tells R what dataframe to use to find `mpg`
- Each column of `mtcars` is a vector

The `mtcars` dataframe

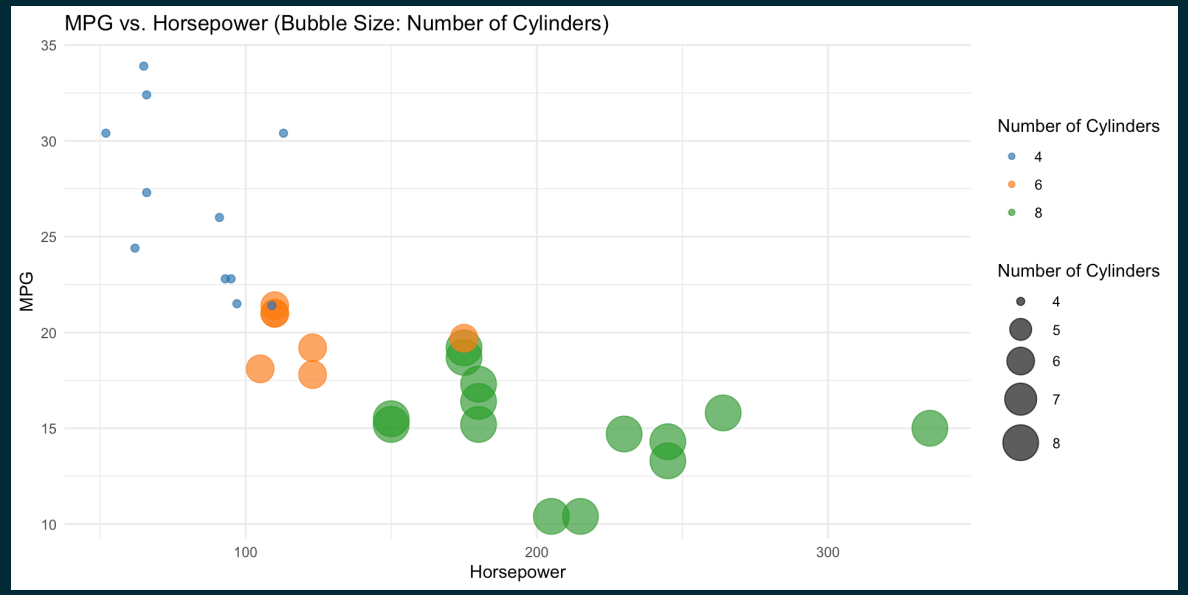
- The name of each vector is an abbreviation at the top
- The full names can be found by saying `?mtcars` which also gives other information about the dataframe
- You can find all the column means by saying `colMeans(mtcars)`
- You can find all the standard deviations by saying `sapply(mtcars,sd)`
- The call `colMeans(mtcars)` is a faster version of `sapply(mtcars,mean)`

Packages

- Most of the functionality in R is in packages
- You install packages into the `library` and retrieve them from there
- Some packages are actually collections of packages
- We'll use a collection called the `tidyverse`
- So say `install.packages("tidyverse")` now
- Also say `install.packages("pacman")` because it simplifies installation and loading

A picture of mtcars

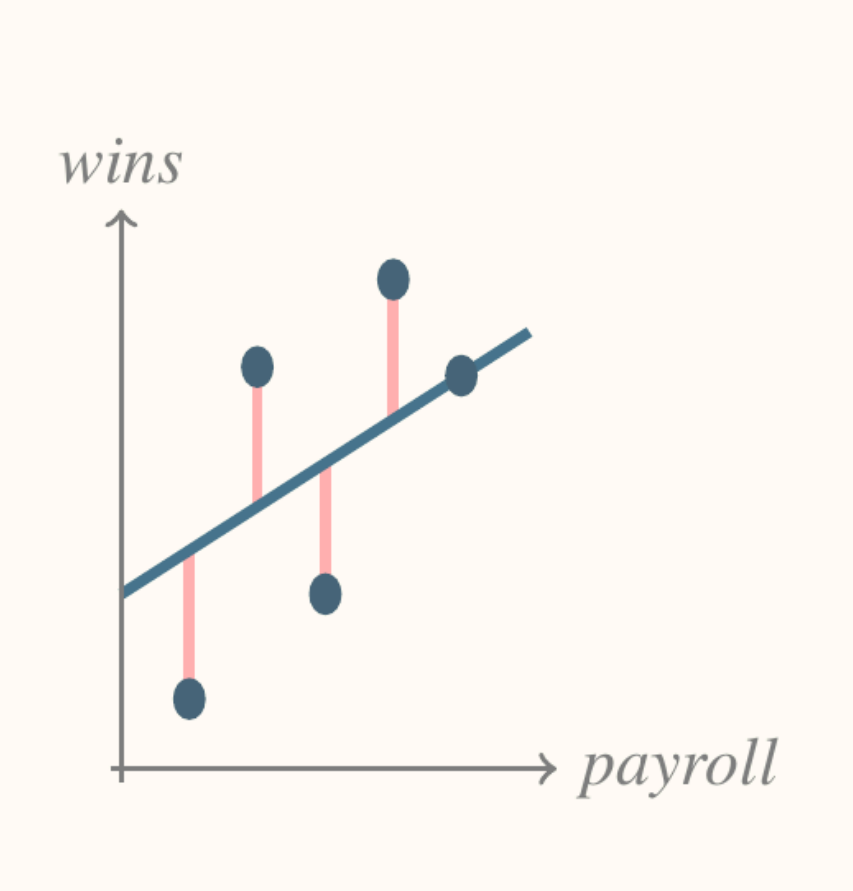
```
#. install.packages("pacman")
pacman::p_load(tidyverse)
ggplot(mtcars, aes(x = hp, y = mpg, size = cyl, color = factor(cyl))) +
  geom_point(alpha = 0.7) +
  scale_size_continuous(range = c(2, 10)) +
  scale_color_manual(values = c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
  labs(x = "Horsepower", y = "MPG", size = "Number of Cylinders", color = "
  theme_minimal()
```

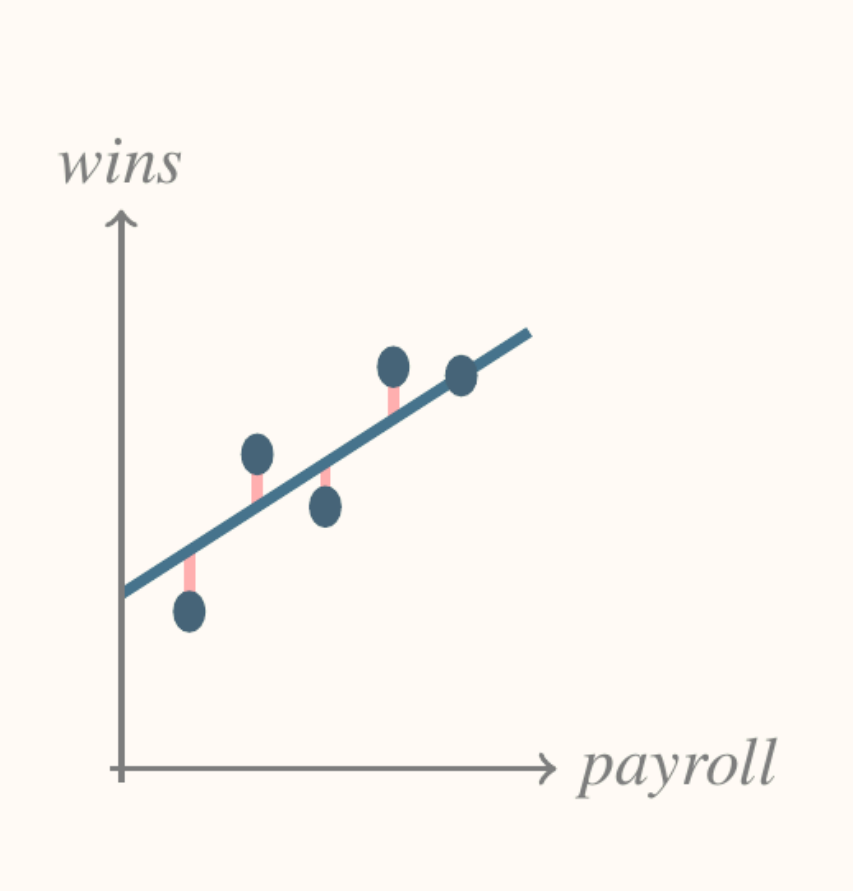


Goal

- Interpret the variability in data
- Describe unwieldly data
- Estimate unknown quantities
- Predict the future
- Understand the mechanisms affecting stochastic processes

Difference between prediction lines





This course is about finding and assessing the best line

- Consider a desired outcome (e.g., wins)
- Identify one or more factors contributing (e.g., payroll dollars)
- Find the slope and intercept to predict how much of the factor leads to how much of the outcome
- Figure out how good or bad the prediction is

… and there you have a simplified view of regression, the heart of this course.

Think about the prediction lines as models of reality

- reduces reality to a manageable fiction
- requires deep knowledge of the subject you model
- easy to do badly
- hard to figure out what aspects of reality to leave in and what to take out

Modeling

- That's the process I've just described
- Mainly equations of lines in this class
- Parsimonious description of some aspect of reality

Expense of modeling

- More realistic ⇒ more expensive
- Example: fishing

Death penalty in Florida example

- Analysts tried to find racism in the death penalty in Florida
- Most failed
- How could they go wrong?
- Finally, one analyst figured it out. How?

Technicality of this course

- You will learn the mechanics
- You will learn how to build models
- You have to supply the intuition and insight
- No one can teach you to think of the best model

Summary statistics

Goals for this section

- Distinguish between samples and populations.
- Know how to calculate the arithmetic mean.
- Know how to calculate standard deviation.
- Know the definition of median.
- Review other summary statistics.

We use samples to make inferences about populations

finding a sum is denoted like this

The Greek letter Sigma, Σ, usually means to sum the values represented by the expression that follows:

$$\sum_{i=1}^{n} y_i$$

which is the same as

$$y_1 + y_2 + \cdots + y_n$$

Sigma notation may be inconsistent

You may see Σ used in an inconsistent way in math and stats:

$$\sum_{i=1}^{n} y_i$$

may be replaced by a synonymous shortcut like

$$\sum_{i} y_i \quad \text{or} \quad \sum y$$

Means

The arithmetic mean is the average of a set of values.

Usually when we use the word *mean*, we refer to

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

which is the same as

$$\bar{y} = \frac{y_1 + y_2 + \cdots + y_n}{n}$$

Sample mean and population mean

We use the sample mean to estimate the population mean.

The sample mean is often denoted as $\bar{y}$.

The population mean is called the expected value of y and is often denoted as

$$E(y) = \mu$$

and in the case of the boxes, we would have to destroy all of them to be sure of its value, so we destroy a sample to estimate μ.

Range

A sample's range is the difference between its max and min.

If the grades of a sample of six students are

$$(2, 2, 3, 3, 4, 4)$$

then the range is

$$4 - 2 = 2$$

The mean of the sample is

$$\bar{y} = (2 + 2 + 3 + 3 + 4 + 4)/6 = 3$$

Standard deviation

Standard deviation is used to describe data variation.

The standard deviation of a population is σ and of a sample is s. It's painfully easy to confuse the spreadsheet functions for σ and s, usually `stdev` and `sstdev`.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{n}}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}}$$

Standard deviation example

Find the standard deviation of the grade sample.

- sum: $2 + 2 + 3 + 3 + 4 + 4 = 18$
- mean: $18/6 = 3$
- deviations:
  $(2 - 3)^2 + (2 - 3)^2 + (3 - 3)^2 + (3 - 3)^2 + (4 - 3)^2 + (4 - 3)^2$
- deviations part two: $1 + 1 + 0 + 0 + 1 + 1$
- divide: $4/(6 - 1) = 4/5 = 0.8$
- square root: just write $\sqrt{0.8}$ unless you're allowed a calculator / computer

Deviations

The *deviations part two* step shown previously is the numerical version of what I previously showed in the graph with pink lines between data and some imaginary prediction line. In this case, the imaginary line is $\bar{y}$. (In the previous case, it was the *least squares line*, which we'll learn about later.)

Standard deviation calculation

Calculating s emphasizes its interpretation

Here's a shortcut equivalent to the previous formula for s.

$$s = \sqrt{\frac{\sum_{i=1}^{n} y_i^2 - n(\bar{y})^2}{n-1}}$$

Two rough guidelines to interpret s

1. For any data set, at least three-fourths of the measurements will lie within two standard deviations of the mean.
2. For most data sets with enough measurements (25 or more) and a mound-shaped distribution, about 95 percent of the measurements will lie within two standard deviations of the mean. (We'll study mound-shaped distributions later.)

Standard deviation and mean work as a pair.

When you want to describe a set of data, the two most frequently used numbers, used as a pair, are mean and standard deviation. Suppose two websites, tra.com and la.com, both sell used phones. The last five sales of the ZZ11 on tra.com, in chronological order, were $36, $29, $59, $18, $23, $35, $25, $63, $69, and $43.

The last fives sales of the ZZ11 on la.com, in chronological order, were $44, $36, $47, $38, $35, $36, $37, $38, $50, and $39. Using only this info, what is the expected value of the next sale in each market? How is it spread out in each market?

Do it in R

```r
tra ← c(36,29,59,18,23,35,25,63,69,43)
mean(tra)
```

```
[1] 40
```

```r
sd(tra)
```

```
[1] 17.95055
```

```r
la ← c(44,36,47,38,35,36,37,38,50,39)
mean(la)
```

```
[1] 40
```

```r
sd(la)
```

```
[1] 5.163978
```

Further, suppose bla.com has mean 40 and sd 36.89. Where would you sell?

Summary statistics in R

```r
#. help(mtcars)
#. ?mtcars
df ← mtcars
summary(df$mpg)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.40   15.43   19.20   20.09   22.80   33.90
```

```r
(s ← sd(df$mpg))
```

```
[1] 6.026948
```

```r
(m ← mean(df$mpg))
```

```
[1] 20.09062
```

```r
(lower ← m-(2*s))
```

```
[1] 8.036729
```

```r
(upper ← m+(2*s))
```

```
[1] 32.14452
```

Summarize the entire dataframe

```
1 str(df)
```

```
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
1 summary(df)
```

```
      mpg             cyl             disp             hp       
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
      drat             wt             qsec             vs        
 Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
```

Less ugly summaries

```
1  pacman::p_load(vtable)
2  df ← mtcars
3  df ▷ sumtable(summ=c('mean(x)','median(x)'),out='return')
```

	Variable	Mean	Median
1	mpg	20	19
2	cyl	6.2	6
3	disp	231	196
4	hp	147	123
5	drat	3.6	3.7
6	wt	3.2	3.3
7	qsec	18	18
8	vs	0.44	0
9	am	0.41	0
10	gear	3.7	4
11	carb	2.8	2

There's just one problem

- Not all of the columns *should* be numeric
- Discover this by saying `?mtcars` in the R console
- You may also say `str(mtcars)` to discover that all of the columns are currently numeric
- You have to manually change each of columns 2 and 8 through 11 to factors (see next slide)

Changing columns to factors, first way

```
df[,2] ← as.factor(df[,2])
df[,8] ← as.factor(df[,8])
df[,9] ← as.factor(df[,9])
df[,10] ← as.factor(df[,10])
df[,11] ← as.factor(df[,11])
```

Changing columns to factors, second way

```
1 df ▷ mutate(across(c(2,8:11),as.factor))
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4

Changing columns to factors, third way

```r
df$vs ← factor(df$vs,labels=c("V","S"))
df$am ← factor(df$am,labels=c("automatic","manual"))
df$cyl ← ordered(df$cyl)
df$gear ← ordered(df$gear)
df$carb ← ordered(df$carb)
```

Structure after repair

```
1 str(df)
```

```
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : Ord.factor w/ 3 levels "4"<"6"<"8": 2 2 1 2 3 2 3 1 1 2 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : Factor w/ 2 levels "V","S": 1 1 2 2 1 2 1 2 2 2 ...
 $ am  : Factor w/ 2 levels "automatic","manual": 2 2 2 1 1 1 1 1 1 1 ...
 $ gear: Ord.factor w/ 3 levels "3"<"4"<"5": 2 2 2 1 1 1 1 2 2 2 ...
 $ carb: Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<..: 4 4 1 1 2 1 4 2 2 4 ...
```

Less ugly summaries, just the numeric columns

```
1 df[,c(1,3:7)] ▷ sumtable(summ=c('min(x)','median(x)','mean(x)','max(x)'),o
```

	Variable	Min	Median	Mean	Max
1	mpg	10	19	20	34
2	disp	71	196	231	472
3	hp	52	123	147	335
4	drat	2.8	3.7	3.6	4.9
5	wt	1.5	3.3	3.2	5.4
6	qsec	14	18	18	23

It's tough to make predictions, especially about the future.

— Yogi Berra

END

Colophon

This slideshow was produced using `quarto`

Fonts are *Roboto Condensed Bold*, *JetBrains Mono Nerd Font*, and *STIX*