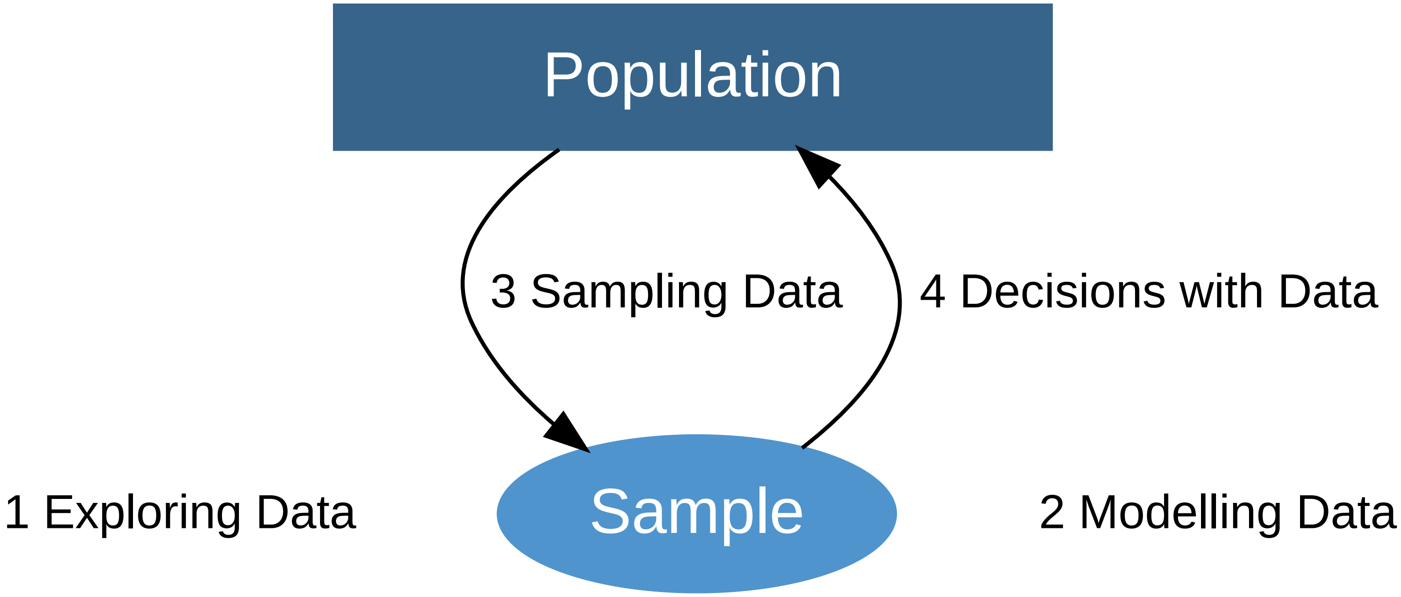
Spread

Exploring Data | Numerical Summaries

Unit Overview



Population

3 Sampling Data

4 Decisions with Data

1 Exploring Data

Sample

2 Modelling Data

Module1 Exploring Data

Design of Experiments

Where did the data come from & can we make reliable conclusions?

Data & Graphical Summaries

What type of data do we have & how can we visualise it?

Numerical Summaries

What are the main features of the data?

Spread

Data Story | How much does a property in Newtown cost?

Standard deviation

Standard Units

IQR

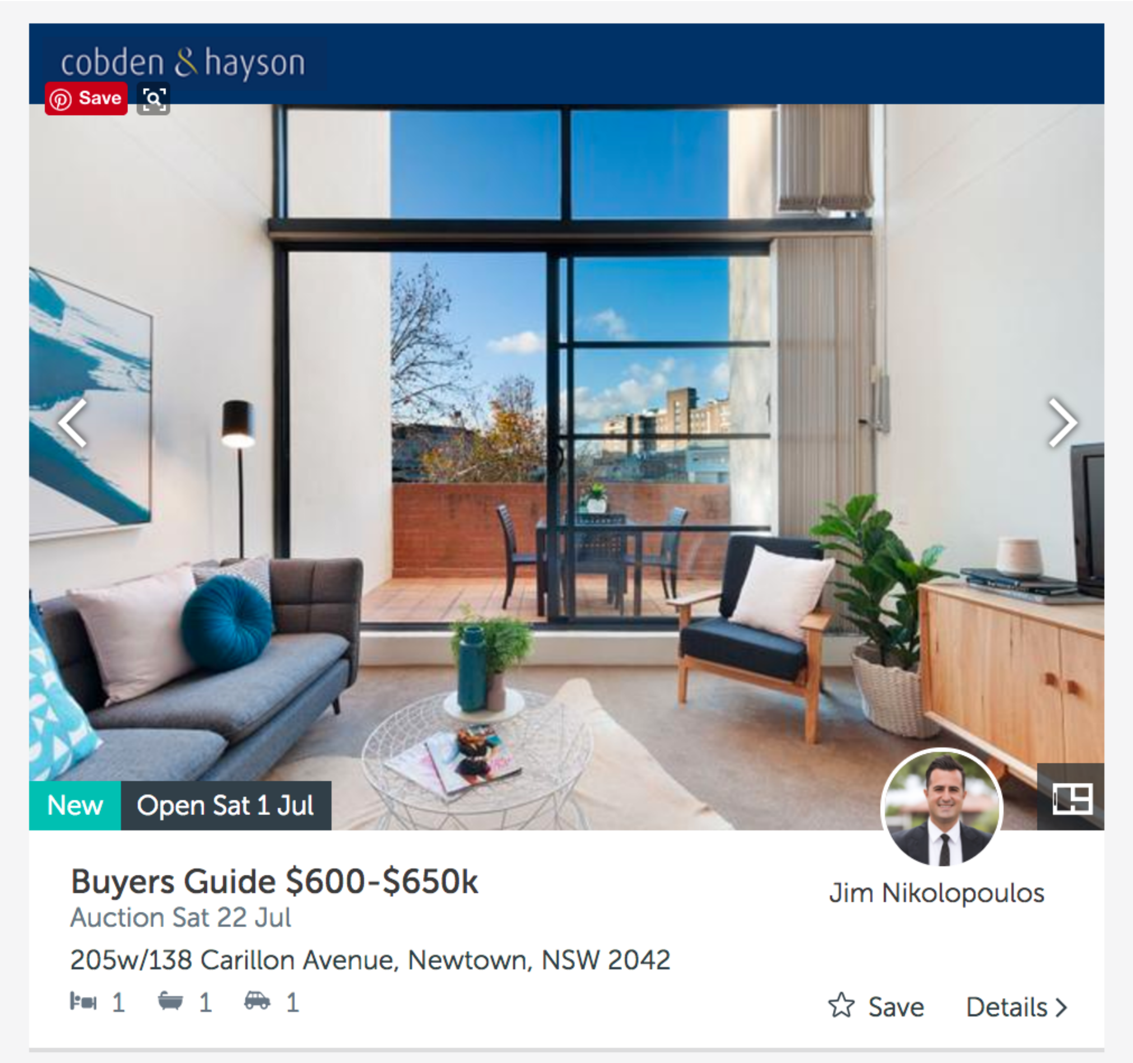
Machine learning (Ext)

Population vs Sample (Ext)

Summary

Data Story

How much does a property in Newtown cost?



Buyers Guide $600-$650k

Auction Sat 22 Jul

205w/138 Carillon Avenue, Newtown, NSW 2042

🛏 1 🛁 1 🚗 1

Jim Nikolopoulos

Data on Newtown Property Sales

Data from domain.com.au for Newtown (NSW 2042) between April-June 2017.

```r
data <- read.csv("data/NewtownJune2017.csv",header=T)
head(data, n=2)
```

```
##                   Property  Type    Agent Bedrooms Bathrooms Carspots Sold
## 1 19 Watkin Street Newtown House RayWhite        4         1        1 1975
## 2  30 Pearl Street Newtown House RayWhite        2         1        0 1250
##      Date
## 1 23/6/17
## 2 23/6/17
```

```
dim(data)
```

```
## [1] 56  8
```

```
str(data)
```

```
## 'data.frame':    56 obs. of  8 variables:
##  $ Property : Factor w/ 56 levels "1 Pearl Street Newtown",..: 20 26 24 23 8 55 47 33 14 54 ...
##  $ Type     : Factor w/ 5 levels "Apartment","House",..: 2 2 2 1 1 2 5 1 1 2 ...
##  $ Agent    : Factor w/ 18 levels "Belle","BresicWhitney",..: 15 15 1 15 14 15 16 6 13 10 ...
##  $ Bedrooms : int  4 2 2 1 1 5 1 1 1 3 ...
##  $ Bathrooms: int  1 1 1 1 1 1 1 1 1 2 ...
##  $ Carspots : int  1 0 0 1 1 1 0 1 1 0 ...
##  $ Sold     : int  1975 1250 1280 780 650 2100 675 740 625 1950 ...
##  $ Date     : Factor w/ 30 levels "1/4/17","1/5/17",..: 15 15 8 8 8 7 14 10 7 29 ...
```

```
mean(data$Sold)
```

```
## [1] 1407.143
```

Statistical Thinking

We now know that the mean house of Newtown properties is $1.407 million.

- In what ways is this limited in usefulness?
- How could it be misreported?

Standard deviation

How to measure spread?

For each property sold, we could calculate the **gap** between the house and the mean $1407 (thousands).

Property	Sold	Gap	Conclusion
19 Watkin Street	$1950 (thousands)	1950-1407=543	More than half a million dollars more expensive than the average house price
30 Pearl St	$1250 (thousands)	1250-1407=-157	Cheaper than the average house price

```
gaps = data$Sold - mean(data$Sold)
gaps
```

```
##  [1]   567.857143  -157.142857  -127.142857  -627.142857  -757.142857
##  [6]   692.857143  -732.142857  -667.142857  -782.142857   542.857143
## [11]   -32.142857   167.857143  -408.142857  -452.142857  -547.142857
## [16]   197.857143   182.857143  -167.142857 -1037.142857   532.857143
## [21]  -687.142857  -452.142857  -487.142857   442.857143   192.857143
## [26]  -652.142857    -7.142857   145.857143  1402.857143   192.857143
## [31]   792.857143   372.857143   398.857143   293.857143   -98.142857
## [36]  -307.142857  -472.142857  -762.142857    52.857143   -37.142857
## [41]  -715.142857  1742.857143  1002.857143  -637.142857   254.857143
## [46]   827.857143   592.857143   382.857143   342.857143   302.857143
## [51]   192.857143  -546.142857  -667.142857   -92.142857   892.857143
## [56]  -595.142857
```

```
max(gaps)
```

```
## [1] 1742.857
```

What are the biggest and smallest gaps?

How we do **summarise** all the gaps into **1 number** (“spread”)?

1st attempt: The mean gap

We could calculate the **average** of the gaps.

Mean gap

$$\text{Mean gap} = \text{Mean of (data-mean)}$$

```
round(mean(gaps))
```

```
## [1] 0
```

What’s the problem?

Note: It will always be 0!

- From the definition, the mean gap must be 0, as the mean is the **balancing point** of the gaps.
- Or for those who like algebra, the mean gap is $\frac{\sum_{i=1}^{n}(x_i - \bar{x})}{n} = \frac{\sum_{i=1}^{n} x_i}{n} - \frac{n\bar{x}}{n} = 0$.

Better option: Standard deviation

First define the Root Mean Square (RMS).

Root Mean Square

- The RMS measures the **average** of a set of numbers, regardless of the signs.
- The steps are: *Square* the numbers, then *Mean* the result, then *Root* the result.

$$\text{RMS of numbers} = \sqrt{\text{Mean of }(\text{numbers})^2}$$

- So effectively, the *Square* and *Root* operations “reverse” each other.

- Applying RMS to the gaps, we get

$$\text{RMS of gaps} = \sqrt{\text{Mean of }(\text{gaps})^2}$$

- To avoid the cancellation of the gaps, another possible method is to consider the average of the absolute values of the gaps: $\frac{\sum_{i=1}^{n} |\text{gaps}|}{n}$. However, this is harder algebraically.

Standard deviation in terms of RMS

Population Standard deviation

- The standard deviation measures the **spread** of the data.

$$\text{SD}_{pop} = \text{RMS of (gaps from the mean)}$$

- Formally,

$$\text{SD}_{pop} = \sqrt{\text{Mean of (gaps from the mean)}^2} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}}$$

```
sqrt(mean(gaps^2))
```

```
## [1] 593.7166
```

Standard deviation in R?

It is easy to calculate in R.

```
sd(data$Sold)
```

```
## [1] 599.0897
```

💬 But why is this slightly different?

Adjusting the standard deviation

- It turns out that there are **2** slightly different formulas for the standard deviation, depending on whether your data is the **population** or a **sample**.
- The `sd` command in R always works out the **sample** version, as we most commonly have samples!

So here, we need to adjust by `sqrt(55/56)`.

```
sd(data$Sold)*sqrt(55/56)   # Magic!! This now calculates the population SD.
```

```
## [1] 593.7166
```

```
library(multicon)
popsd(data$Sold)  # Quickest way!
```

```
## [1] 593.7166
```

Overall Summary: population vs sample

Summary	Formula	In R
Population or **Sample** mean	Mean (Average)	`mean(data)`
Population standard deviation SD_{pop}	RMS of gaps from the mean	`popsd(data)`
Sample standard deviation SD_{sample}	Adjusted RMS of gaps from the mean	`sd(data)`

Note:

- The squared standard deviation is called the **variance**: SD^2.
- Formally, $\mathrm{SD}_{pop} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(gaps)^2}$ and $\mathrm{SD}_{sample} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(gaps)^2}$.
- `popsd(data)` requires the `multicon` package.

Large samples?

```
# population SD
popsd(data$Sold)
```

```
## [1] 593.7166
```

```
# sample SD
sd(data$Sold)
```

```
## [1] 599.0897
```

Note for large sample sizes, the difference becomes negligible.

How to tell the difference?

- It can be tricky to work out whether your data is a population or sample!
- Look at the information about the data story and the research questions.
  - If we are just interested in the Newtown property prices during April-June 2017, then the `data` is the whole **population**.
  - If we are studying the property prices during April-June 2017 as a window into more general property prices (for the rest of the year or for the Inner West area) , then the `data` could be considered a **sample**.
  - More of this interesting topic in 3rd year courses!
- For simplicity, let’s now assume the Newtown data is a **sample** for the following work.

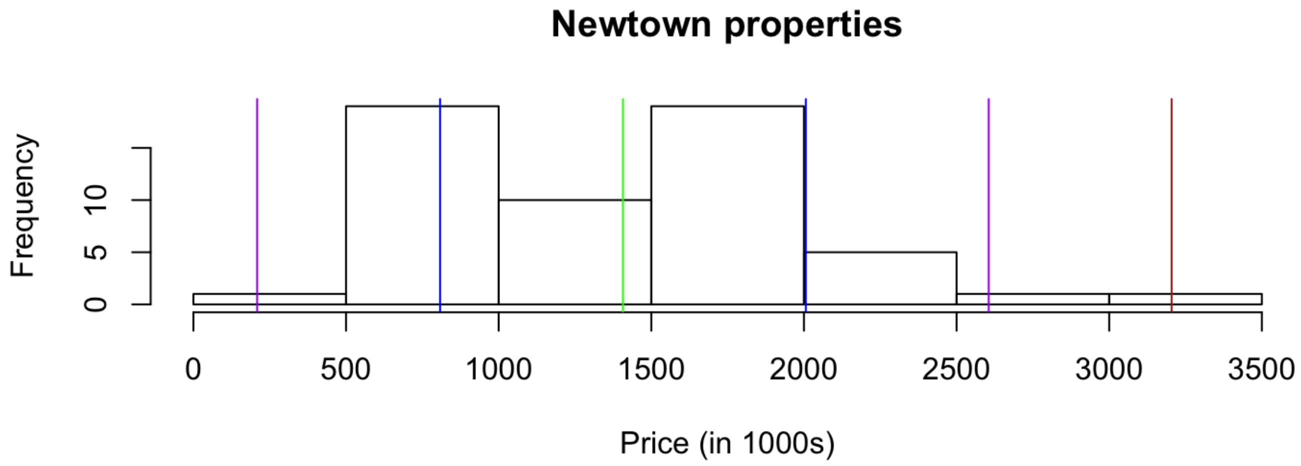
```
sd=sd(data$Sold)
sd
```

```
## [1] 599.0897
```

Standard Units

Standard deviation on the histogram

```
hist(data$Sold, main="Newtown properties", xlab="Price (in 1000s)")
abline(v=mean(data$Sold),col="green")
abline(v=mean(data$Sold)-sd,col="blue")
abline(v=mean(data$Sold)+sd,col="blue")
abline(v=mean(data$Sold)-2*sd,col="purple")
abline(v=mean(data$Sold)+2*sd,col="purple")
abline(v=mean(data$Sold)-3*sd,col="brown")
abline(v=mean(data$Sold)+3*sd,col="brown")
```



Useful rule of thumb

- For many data sets, we find that roughly

Percentage of data	Distance from mean
68%	within 1 SD
95%	within 2 SDs
99.7%	within 3 SDs

- When we study the Normal curve, we shall see why this is reasonable.

Standard units

Standard units (“Z score”)

Standard units of a data point = how many standard deviations is it below or below the mean

$$\text{Standard units} = \frac{\text{data point - mean}}{\text{SD}}$$

This means that

$$\text{data point} = \text{mean} + \text{SD} \times \text{standard units}$$

Comparing 2 data points

To compare 2 data points, we can compare the standard units.

Property	Sold	Standard units	Conclusion
19 Watkin Street	$1950 (thousands)	$\frac{1950-1407}{599} = 0.91$	Almost 1 SD higher than the average house price
30 Pearl St	$1250 (thousands)	$\frac{1250-1407}{599} = -0.26$	0.26 SDs cheaper than the average house price

So 19 Watkin is a more unusual purchase than 30 Pearl St, relative to the mean.

IQR

IQR

The IQR is another measure of spread.

Interquartile Range (IQR)

$$\text{IQR} = \text{Range of the middle } 50\% \text{ of the data}$$

- More formally, $\text{IQR} = Q_3 - Q_1$, where
  - Q_1 is the 25% percentile (1st quartile) and Q_3 is the 75% percentile (3rd quartile).
  - The median is the 50% percentile, or 2nd quartile $\tilde{x} = Q_2$.

Quantile vs quartile

The set of q **quantiles** divides the data into $q - 1$ equal size sets (in terms of percentage of data).

The set of **quartiles** divides the data into quarters.

```
quantile(data$Sold)
```

```
##      0%     25%     50%     75%    100% 
##  370.00  860.75 1387.50 1782.50 3150.00
```

```
quantile(data$Sold)[4] - quantile(data$Sold)[2]
```

```
##    75% 
## 921.75
```

So the range of the middle 50% of properties sold is almost a million dollars!

IQR on the boxplot

- The IQR is the length of the box in the boxplot. It represents the span of the middle 50% of the houses sold.

- The **lower** and **upper thresholds** are a distance of 1.5 from the quartiles (by convention).

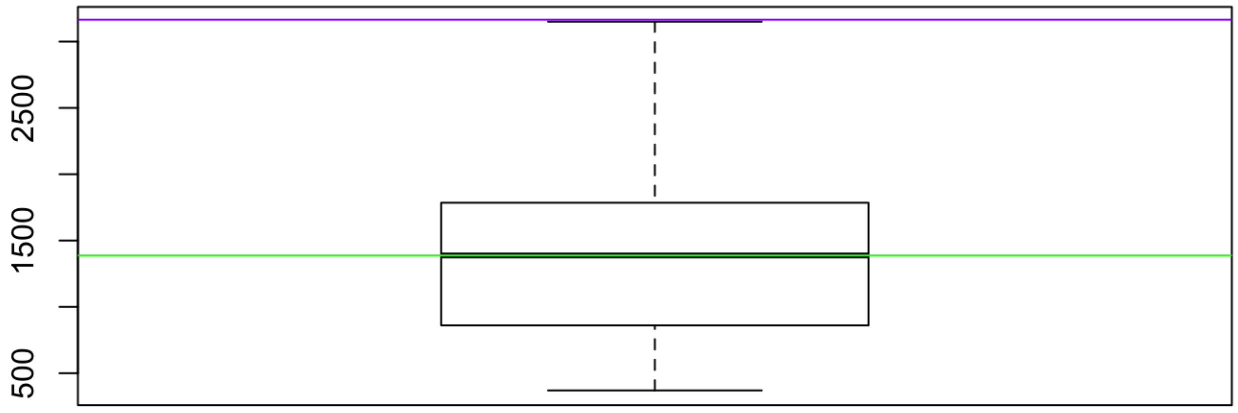
$$LT = Q_1 - 1.5IQR$$

and

$$UT = Q_3 + 1.5IQR$$

- Data outside these thresholds is considered an **outlier** (“extreme reading”).

```
boxplot(data$Sold)
iqr=quantile(data$Sold)[4] - quantile(data$Sold)[2]
abline(h=median(data$Sold),col="green")
abline(h=quantile(data$Sold)[2]- 1.5*iqr,col="purple")
abline(h=quantile(data$Sold)[4]+ 1.5*iqr,col="purple")
```



Note there were no outliers in this data, even the 3.150 million dollar purchase!

Reporting

- Like the median, the IQR is **robust**, so it’s suitable as a summary of spread for skewed data.
- We report in pairs: (mean,SD) or (median,IQR).

Coefficient of Variation

- The **Coefficient of Variation (CV)** combines the mean and standard deviation into 1 summary:

$$\mathrm{CV} = \frac{\mathrm{SD}}{\mathrm{mean}}$$

- The CV is used in:
    - analytical chemistry to express the precision and repeatability of an assay;
    - engineering and physics for quality assurance studies;
    - economics for determining the volatility of a security.

CV of Newtown properties

```
# all properties
m = mean(data$Sold)
sd = sd(data$Sold)
sd/m
```

```
## [1] 0.4257491
```

```
# 4B properties
data1=data$Sold[data$Type=="House" & data$Bedrooms=="4"]
m1 = mean(data1)
sd1 = sd(data1)
sd1/m1
```

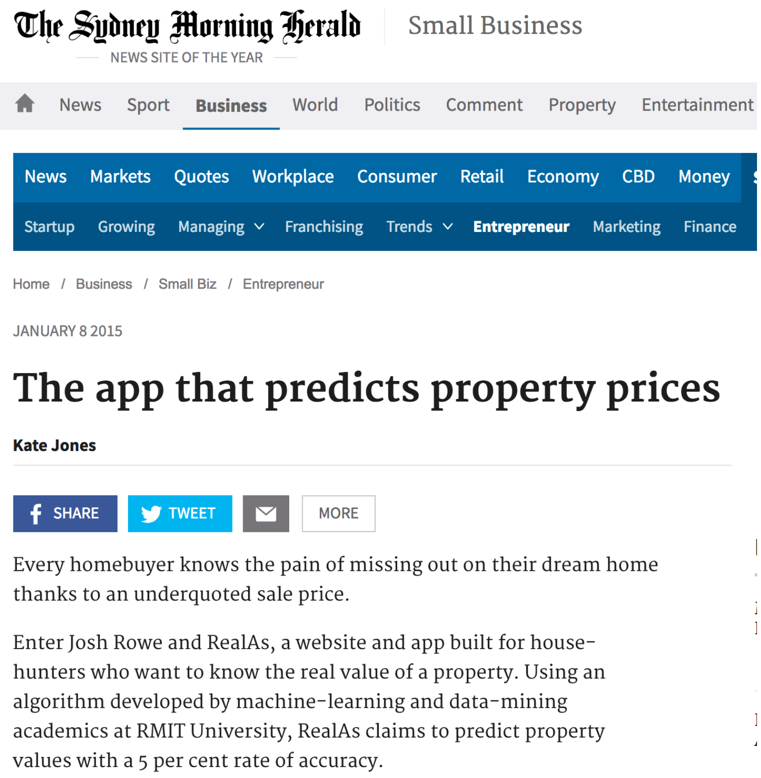
```
## [1] 0.2747063
```

- The CV for all properties is 0.43 (2dp), and for 4B houses is 0.27.
- Hence, there is more volatility in the overall house market.

Machine learning (Ext)

Machine learning

Machine learning is being used to successfully predict house prices.



JANUARY 8 2015

The app that predicts property prices

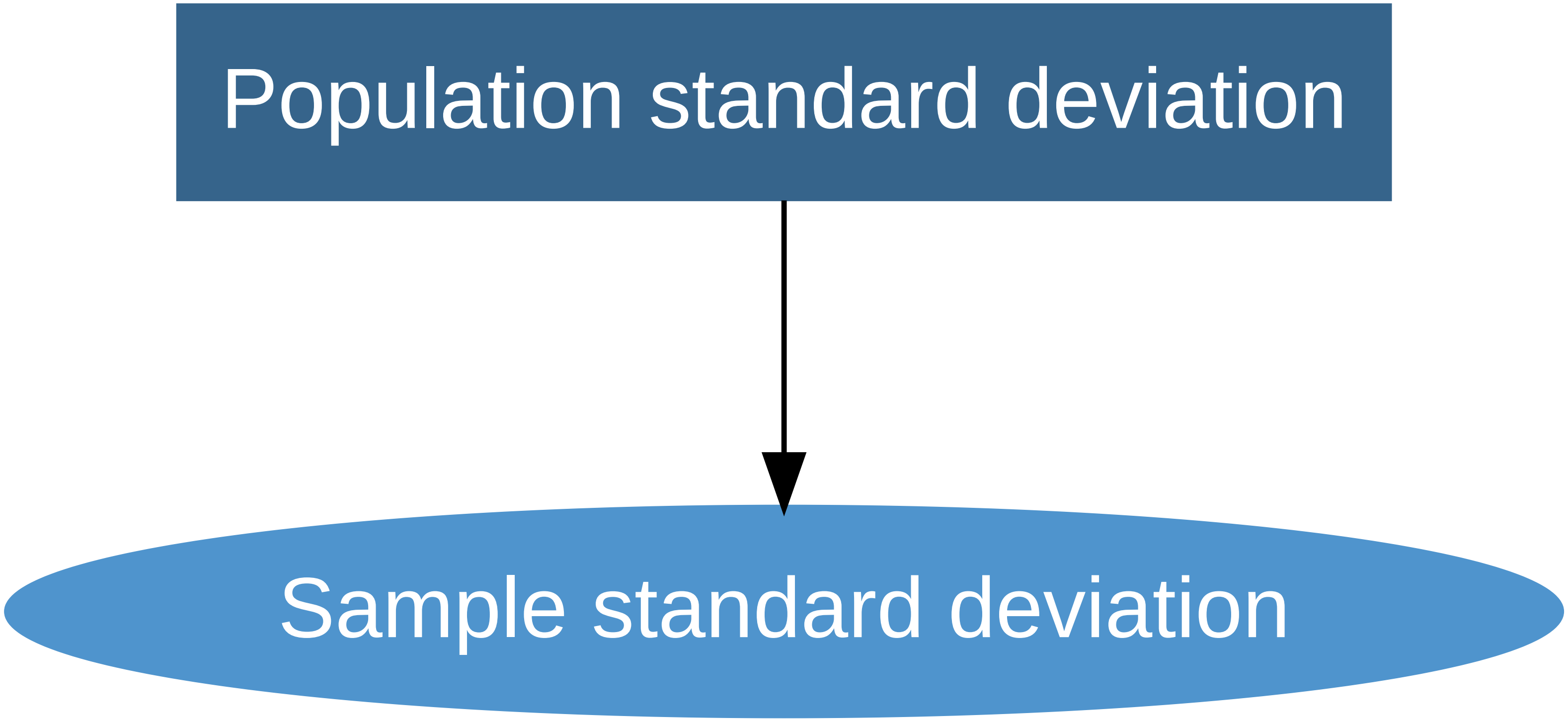
Kate Jones

Every homebuyer knows the pain of missing out on their dream home thanks to an underquoted sale price.

Enter Josh Rowe and RealAs, a website and app built for house-hunters who want to know the real value of a property. Using an algorithm developed by machine-learning and data-mining academics at RMIT University, RealAs claims to predict property values with a 5 per cent rate of accuracy.

SMH RealAs

Population vs Sample (Ext)



Population standard deviation

Sample standard deviation

Estimation Theory

- Applying the **population** standard deviation formula to the **sample**, **underestimates** the **true** population standard deviation.
- Hence, we need an adjusted formula for the sample standard deviation where we divide by $\frac{1}{n-1}$ rather then $\frac{1}{n}$.
- This will be covered in 2nd year courses on estimation theory.
- Here, we simply illustrate by example.

Example

Suppose want to know the average property price in Newtown between April-June 2017. (Note: The use of code in Method4 is very extension!)

Method1: Use the full data (assume it’s a population).

```
# all properties
n = length(data$Sold)
mean(data$Sold)   # this is the true mean
```

```
## [1] 1407.143
```

```
sd(data$Sold)*sqrt((n-1)/n)  # this is the true population sd
```

```
## [1] 593.7166
```

Method2: Ask some agents who sold houses (samples).

```
# RayWhite agent
data1=data$Sold[data$Agent=="RayWhite"]
n1=length(data1)
length(data1)
```

```
## [1] 11
```

```
mean(data1)
```

```
## [1] 1580.091
```

```
sd(data1)*sqrt((n1-1)/n1)  # population formula
```

```
## [1] 667.8657
```

```
sd(data1)  # sample formula
```

```
## [1] 700.4635
```

```
# LJHooker agent
data2=data$Sold[data$Agent=="LJHooker"]
n2=length(data2)
n2
```

```
## [1] 4
```

```
mean(data2)
```

```
## [1] 1265.25
```

```
sd(data2)*sqrt((n2-1)/n2)  # population formula
```

```
## [1] 346.5959
```

```
sd(data2)  # sample formula
```

```
## [1] 400.2144
```

```
# McGrath agent
data3=data$Sold[data$Agent=="McGrath"]
n3=length(data3)
n3
```

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

```
mean(data3)
```

```
## [1] 1540
```

```
sd(data3)*sqrt((n3-1)/n3)  # population formula
```

```
## [1] 839.5634
```

```
sd(data3)  # sample formula
```

```
## [1] 1028.251
```

Method3: Ask 4 friends who just sold their properties.

```
# RayWhite agent
data4=data$Sold[c(2,12,34,55)]
n4=length(data4)
length(data4)
```

```
## [1] 4
```

```
mean(data4)
```

```
## [1] 1706.5
```

```
sd(data4)*sqrt((n4-1)/n4)  # population formula
```

```
## [1] 380.1174
```

```
sd(data4)  # sample formula
```

```
## [1] 438.9218
```

Method4: Simulate 50 samples of size 10, and average the results.

```
set.seed(1)   # Ensures we run same simulation each time
means=rep(NA,50)    # Empty vectors for the "for" loop
sdpop=rep(NA,50)
sdsample=rep(NA,50)
for (i in 1:50)  # Simulate the samples 50 times
{
sample =sample(data$Sold,10)   # Select a sample of 10 readings
means[i]=mean(sample)  # Calculate the mean of sample, and store in means
sdpop[i]=sd(sample)*sqrt(9/10) # Calculate the population sd of sample, and store in sdpop
sdsample[i]=sd(sample)  # Calculate the sample sd of sample, and store in sdsample
}
mean(means)  # average of means
```

```
## [1] 1417.88
```

```
mean(sdpop)  # average of population sds
```

```
## [1] 592.6231
```

```
mean(sdsample)   # average of sample sds
```

```
## [1] 624.6796
```

Summary of results

Data	Size	Mean	Population SD formula	(Adjusted) Sample SD formula
Full population	56	1407.143 (true)	**593.7166 (true)**	
Sample: Ray White	11	1580.091	667.8657	700.4635
Sample: LJHooker	4	1265.25	346.5959	400.2144
Sample: McGrath	3	1540	839.5634	1028.251
Sample:Friends	4	1706.5	380.1174	438.9218
Average of simulations	10	1382.38	548.7902	**578.4757**

Notice for the simulations, that the sample SD is closer to the true population SD.

Summary

The spread of data is commonly measured by the standard deviation or the interquartile range.

We need to distinguish between population and sample to work out the correct standard deviation.

Type of data	Formula	In R
Population standard deviation	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(gaps)^2}$	`popsd(data)`
Sample standard deviation	$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(gaps)^2}$	`sd(data)`

Key Words

root mean square (RMS), standard deviation, variance, standard units, interquartile range (IQR), quantile, quartile, robust, coefficient of variation (CV), machine learning

Further Thinking

🔗 Numerical Summaries