Short crash on correlations

Propositional logic

- **You have two statements:**
  - *If it's raining, the street is wet*
  - *If it's snowing, the street is wet*

- **What can you deduce**
  - *The street is wet,*
  - *The street is not wet*
  - *It have rained*
  - *It have not rained*

- *The street is wet,*
  - => nothing it could rain or snow or something else like flooding

- *The street is not wet*
  - => it have not rained nor snowed

- *It have rained*
  - => The street is wet

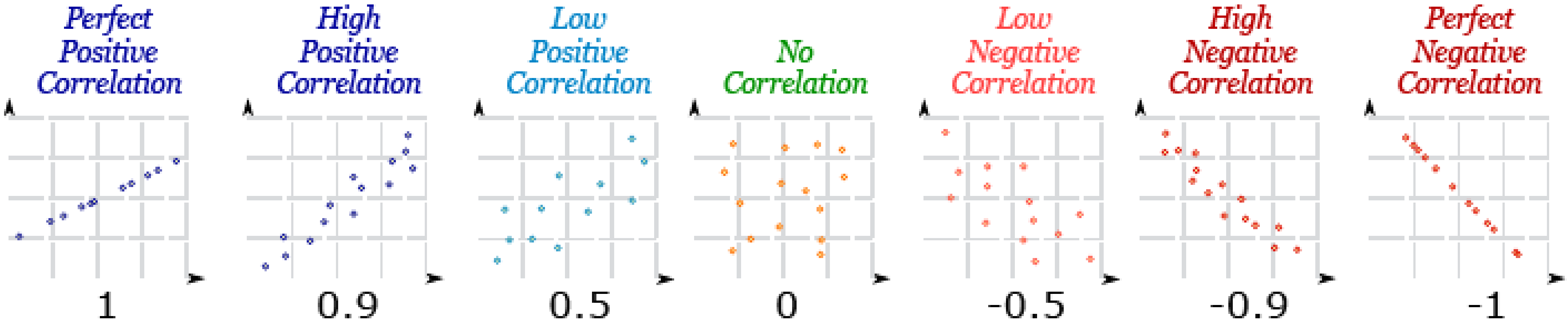
- *It have not rained*
  - => nothing the street could be dry or wet from snowing

Causation and Correlation

- **If we have a dependency between two figures abstract X and Y**
  - X causes Y
  - T causes X
  - Both X and Y are caused by something else like Z
  - There is no causation going on; it’s just a coincidence

Correlations - linear correlations

1. **Correlation is Positive when the values increase together, and**
2. **Correlation is Negative when one value decreases as the other increases**



Source: https://www.mathsisfun.com/data/correlation.html

How to calculate correlation by your self

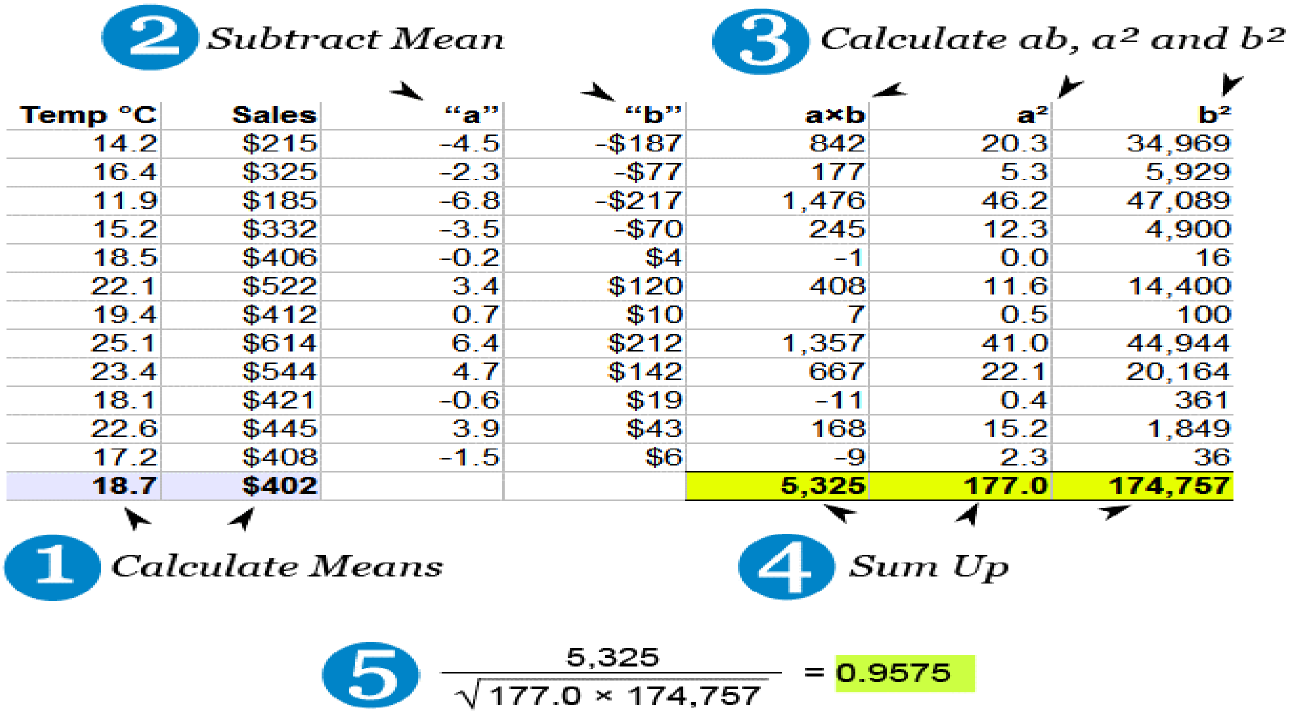
- Let us call the two sets of data "x" and "y"
  (in example Temperature is x and Ice Cream Sales is y):

  - Step 1: Find the mean of x, and the mean of y
  - Step 2: Subtract the mean of x from every x value (call them "a"), do the same for y (call them "b")
  - Step 3: Calculate: a × b, a2 and b2 for every value
  - Step 4: Sum up a × b, sum up a2 and sum up b2
  - Step 5: Divide the sum of a × b by the square root of [(sum of a2) × (sum of b2)]

Source: https://www.mathsisfun.com/data/correlation.html

How to calculate … cont.

- Here is how I calculated the first Ice Cream example (values rounded to 1 or 0 decimal places):



2 *Subtract Mean*

3 *Calculate ab, a² and b²*

Temp °C	Sales	"a"	"b"	a×b	a²	b²
14.2	$215	-4.5	-$187	842	20.3	34,969
16.4	$325	-2.3	-$77	177	5.3	5,929
11.9	$185	-6.8	-$217	1,476	46.2	47,089
15.2	$332	-3.5	-$70	245	12.3	4,900
18.5	$406	-0.2	$4	-1	0.0	16
22.1	$522	3.4	$120	408	11.6	14,400
19.4	$412	0.7	$10	7	0.5	100
25.1	$614	6.4	$212	1,357	41.0	44,944
23.4	$544	4.7	$142	667	22.1	20,164
18.1	$421	-0.6	$19	-11	0.4	361
22.6	$445	3.9	$43	168	15.2	1,849
17.2	$408	-1.5	$6	-9	2.3	36
18.7	**$402**			**5,325**	**177.0**	**174,757**

1 *Calculate Means*

4 *Sum Up*

5 $$\frac{5{,}325}{\sqrt{177.0 \times 174{,}757}} = 0.9575$$

https://www.mathsisfun.com/data/correlation.html

How to calculate … cont.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Where:

- Σ is Sigma, the symbol for "sum up"
- (x_i - $\bar{x}$) is each x-value minus the mean of x (called "a" above)
- (y_i - $\bar{y}$) is each y-value minus the mean of y (called "b" above)

https://www.mathsisfun.com/data/correlation.html

Let move to practice

- If you choose humidity (x) and particles(y)
  - For the same timestamp you have x_i and y_i
  - Now do the 5 steps

    => then you have r (the correlation coefficient)

  Example:

X	Y	Avg(x)	Avg(y)	X-avg(x)=a	Y-avg(y)=b	a*b	a^2	b^2
1.6	20	1.58	19.66	0.0166	0,3333	0.0055	0,000275	0.1111
1.7	22			0.1166	2,3333	0.2720	0,013595	5,444
1.45	17			-0,1333	-2,6666	0.35545	0,01776	7.1107
						0,63295	0.03163	12,6662

  - R=0.999 i.e. In this example high correlation

Pig support

```
inpt = load '~/pig_data/…' as
(amnt:double,id:chararray,c2:chararray);
grp = group inpt by id;
mean = foreach grp {
    sum = SUM(inpt.amnt);
    count = COUNT(inpt);
    generate group as id, sum/count as mean, sum as sum,
count as count;
};
```

- http://stackoverflow.com/questions/12593527/finding-mean-using-pig-or-hadoop

Spark support

```
seriesX = sc.parallelize([1.0, 2.0, 3.0, 3.0, 5.0])
seriesY = sc.parallelize([11.0, 22.0, 33.0, 33.0, 555.0])

print("Correlation is: " + str(Statistics.corr(seriesX, seriesY,
method="pearson")))

data = sc.parallelize( [np.array([1.0, 10.0, 100.0]),
np.array([2.0, 20.0, 200.0]), np.array([5.0, 33.0, 366.0])])

print(Statistics.corr(data, method="pearson"))
```

https://spark.apache.org/docs/latest/mllib-statistics.html