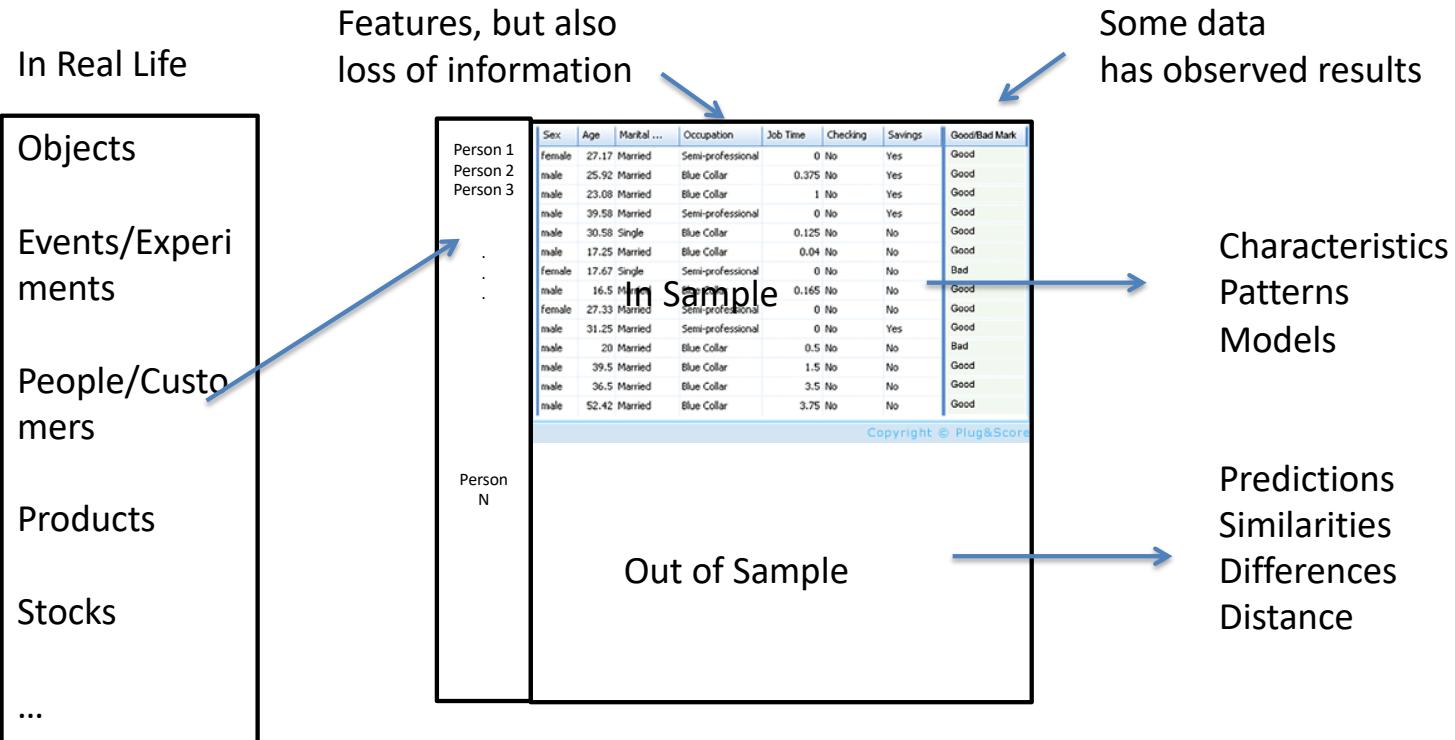


Data X

Data as a Signal and Correlation  
Data, Signals, and Systems

Ikhlaq Sidhu  
Chief Scientist & Founding Director,  
Sutardja Center for Entrepreneurship & Technology  
IEOR Emerging Area Professor Award, UC Berkeley

# A High Level Framework



# Converting From Time Sequence Data to Features

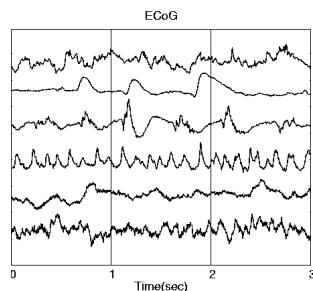
Of course, not all data has a time property, but lets start with this type.  
For example( key1, value 1),( key 2, value 2)... in this case, the keys are indexed by time.

Data X

# Converting From Time Sequence Data to Features

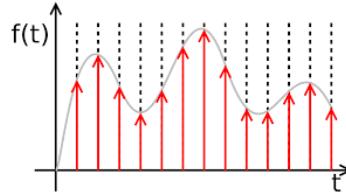
Many Types of data  
are signals in time

- Stock market
- Temperature
- Instrument readings



Continuous signals  
 $x(t)$

Sometimes we  
sample them,  
record at intervals  
of  $T$



Sampled signals (data)  
 $x(nT)$

We get a  
list in a table,  
array, or vector

Rec	Observed
1	60.323
2	61.122
3	60.171
4	61.187
5	63.221
6	63.639
7	64.980
8	63.761
9	66.019
10	67.857
11	68.169
12	66.513
13	68.655
14	69.564
15	69.331
16	70.551



What we want  
(for now):  
features and  
characteristics

For example:

- Means
- Variances
- Pattern matches
- Changes
- accumulation
- Frequency

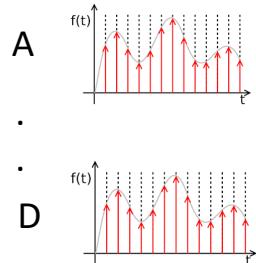
Discrete data  
 $x_n = x_1, x_2, x_3, \dots$

(might lose time reference)



Data X

## Approaches to the Data Sequences from Multiple Sources in Tables



Discrete data for each source  
A, B, C..  
 $x_n = x_1, x_2, x_3, \dots$

One row for each source  
A, B, C, D..

Eg Numpy arrays

One column for each series:  
A, B, C, D..

eg Pandas  
(add rows with every new time sample)

Time ->

A:  $x_1, x_2, x_3, \dots$

B:  $x_1, x_2, x_3, \dots$

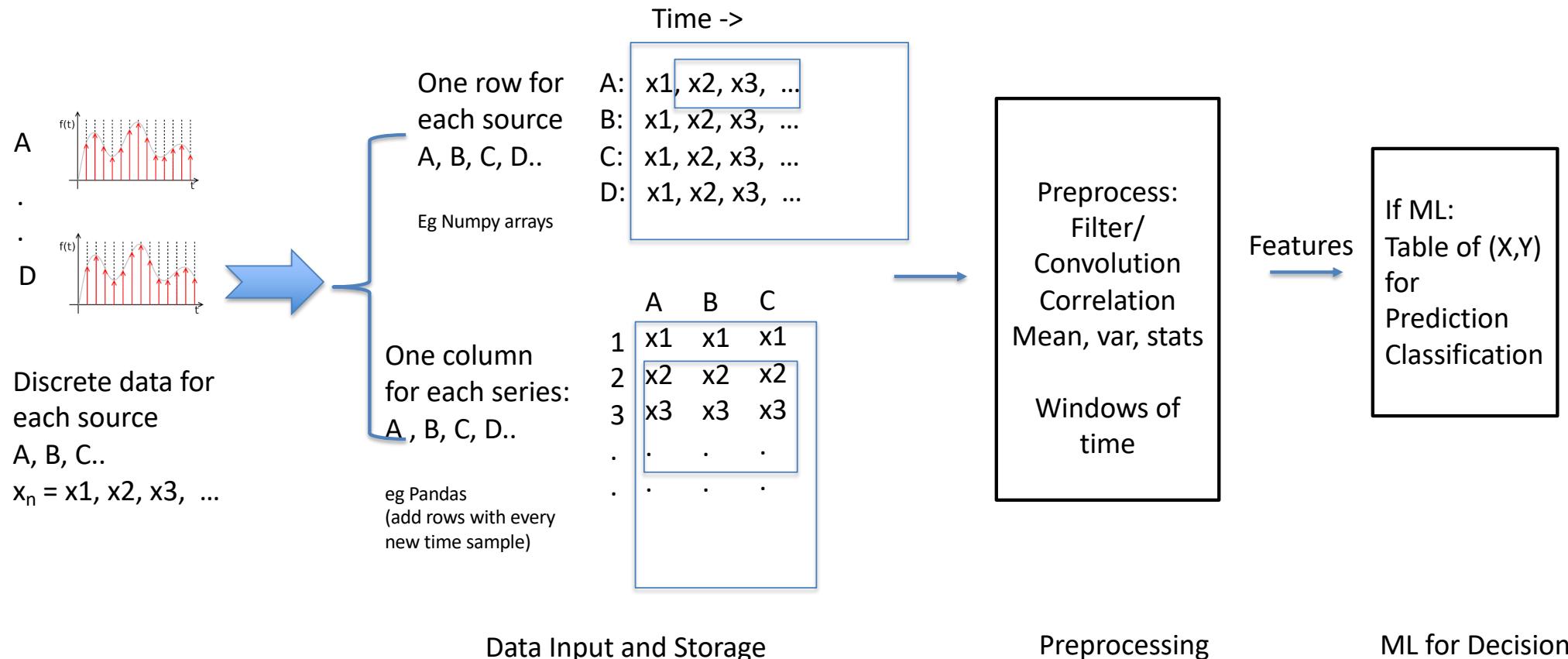
C:  $x_1, x_2, x_3, \dots$

D:  $x_1, x_2, x_3, \dots$

	A	B	C
1	$x_1$	$x_1$	$x_1$
2	$x_2$	$x_2$	$x_2$
3	$x_3$	$x_3$	$x_3$
.	.	.	.
.	.	.	.



# Approaches to the Data Sequences in Tables



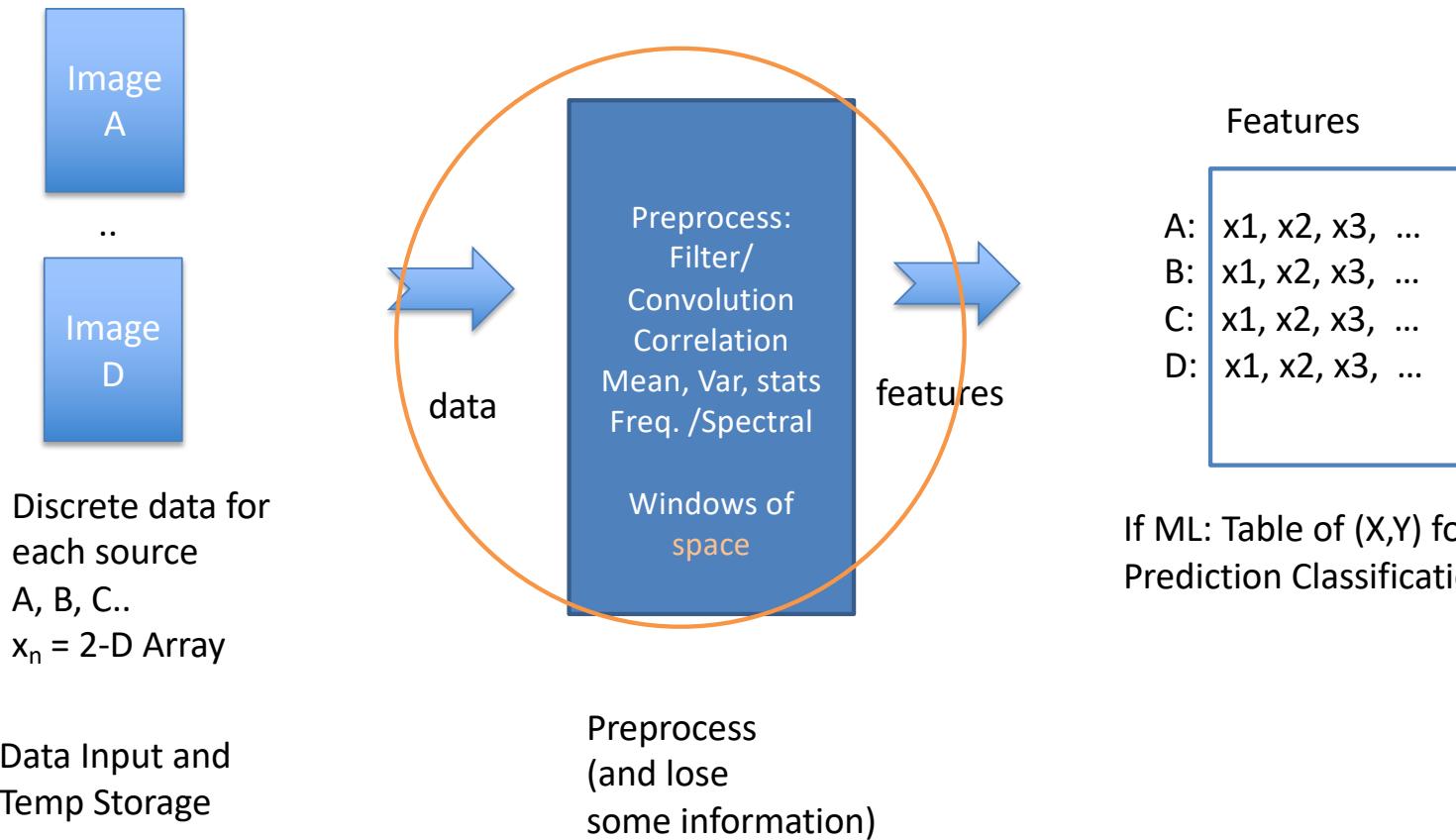
## Data Input and Storage

## Preprocessing

ML for Decisions

Data X

## Data Sequence in Tables Example



## Correlation Matrices

Data X

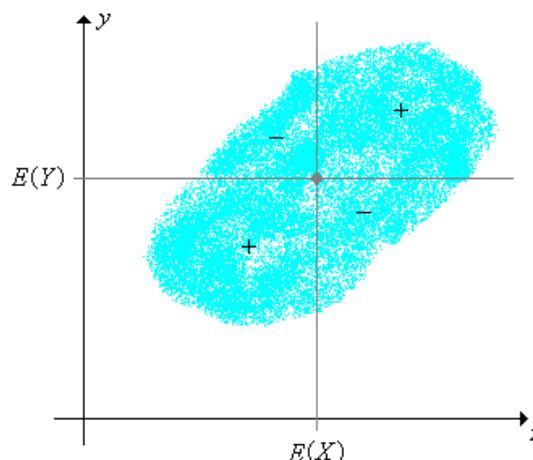


# Review Correlation and Covariance

1. The *covariance* of  $(X, Y)$  is defined by

$$\text{cov}(X, Y) = \mathbb{E}([X - \mathbb{E}(X)][Y - \mathbb{E}(Y)])$$

and, assuming the variances are positive, the *correlation* of  $(X, Y)$  is defined by



$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y},$$

<http://www.math.uah.edu/stat/expect/Covariance.html>

Data X

## Correlation and Covariance

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},$$

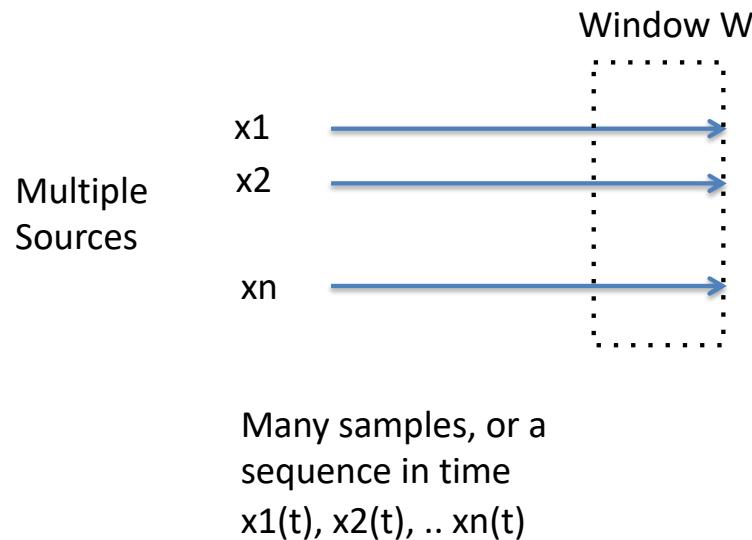
### Properties

- a.  $-1 \leq \text{cor}(X, Y) \leq 1$
- b.  $-\text{sd}(X)\text{sd}(Y) \leq \text{cov}(X, Y) \leq \text{sd}(X)\text{sd}(Y)$
- c.  $\text{cor}(X, Y) = 1$  if and only if  $Y$  is a linear function of  $X$  with positive slope.
- d.  $\text{cor}(X, Y) = -1$  if and only if  $Y$  is a linear function of  $X$  with negative slope.

<http://www.math.uah.edu/stat/expect/Covariance.html>



## Correlation Matrix



Table

Samples	$x_1$	$x_2$	...	$x_n$
1				
2				
3				
$n$				
.				
$N+W$				

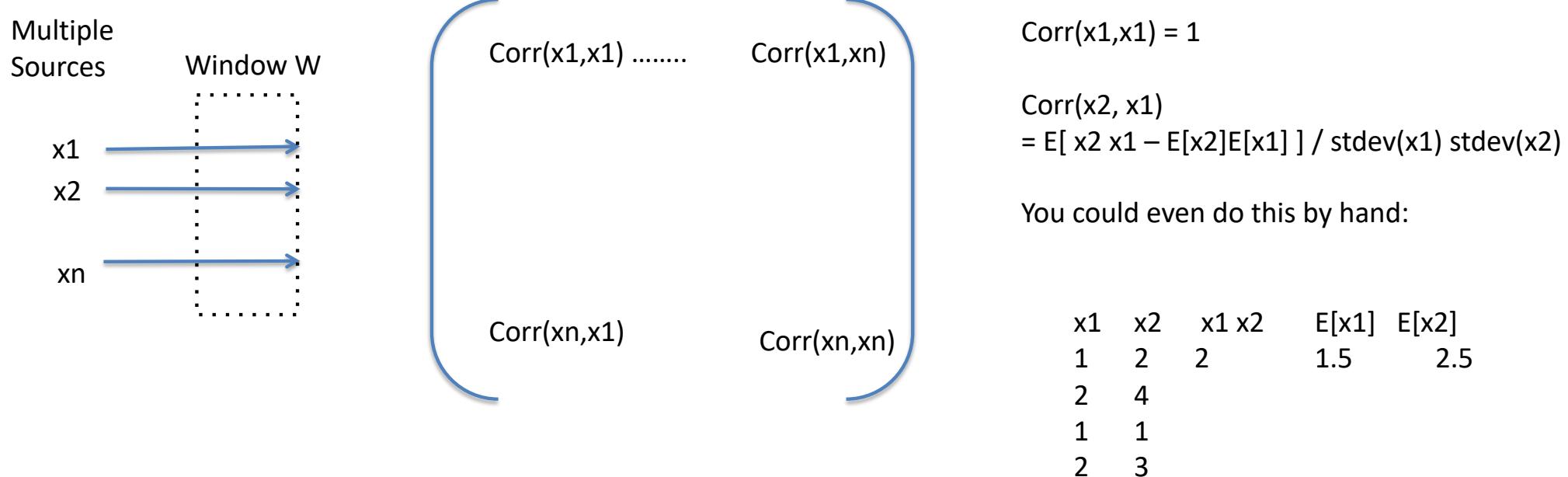
Samples from Window of  $W$

To estimate from data:

- Use all samples ever collected
- Use window size of  $W$  samples of each to estimate a recent Corr Matrix



## Correlation Matrix



To estimate from data:

- Use all samples ever collected
- Use window size of  $W$  samples of each to estimate recent Corr Matrix



## Code Examples: Correlation of Rows with NumPy

```
Import numpy as np

# ignore line formatting
x = np.array(
    [[0.1, .32, .2, 0.4, 0.8],
     [.23, .18, .56, .61, .12],
     [.9, .3, .6, .5, .3],
     [.34, .75, .91, .19, .21]])

np.corrcoef(x)
Out[4]: array([
 [ 1.          , -0.35153114, -0.74736506, -0.48917666],
 [-0.35153114,  1.          ,  0.23810227,  0.15958285],
 [-0.74736506,  0.23810227,  1.          , -0.03960706],
 [-0.48917666,  0.15958285, -0.03960706,  1.          ]])
```

Here each row is a vector of length 5  
There are 4 vectors

Correlation matrix is 4 x 4

If you want the correlation of the columns,  
just use transpose

`np.corrcoef ( np.transpose(x) )`

For a window, use a slice:  
`window = x[0:4,3:5]` for the last  
two columns



# Correlation of Features from Different Sources

Mazda RX4  
Mazda RX4 Wag  
Datsun 710  
Hornet 4 Drive  
Hornet Sportabout  
Valiant

	mpg	disp	hp	drat	wt	qsec
Mazda RX4	21.0	160	110	3.90	2.620	16.46
Mazda RX4 Wag	21.0	160	110	3.90	2.875	17.02
Datsun 710	22.8	108	93	3.85	2.320	18.61
Hornet 4 Drive	21.4	258	110	3.08	3.215	19.44
Hornet Sportabout	18.7	360	175	3.15	3.440	17.02
Valiant	18.1	225	105	2.76	3.460	20.22

Pandas Table  
Use corr()  
like dataframe.corr()

## pandas.DataFrame.corr

DataFrame.corr(method='pearson', min\_periods=1)

[source]

Compute pairwise correlation of columns (excluding NA/null values)

Parameters:

method : {‘pearson’, ‘kendall’, ‘spearman’}

- pearson : standard correlation coefficient
- kendall : Kendall Tau correlation coefficient
- spearman : Spearman rank correlation

min\_periods : int, optional

Minimum number of observations required per pair of columns to have a valid result.  
Currently only available for pearson and spearman correlation

Returns:

y : DataFrame

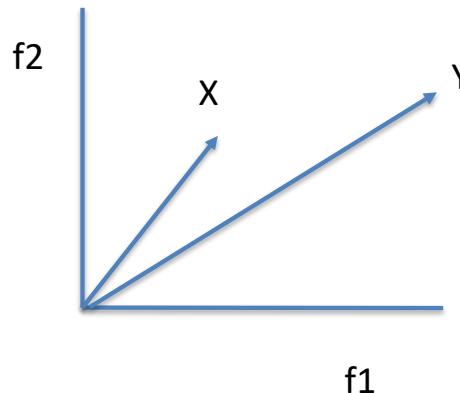
	mpg	disp	hp	drat	wt	qsec
mpg	1.00	-0.85	-0.78	0.68	-0.87	0.42
disp	-0.85	1.00	0.79	-0.71	0.89	-0.43
hp	-0.78	0.79	1.00	-0.45	0.66	-0.71
drat	0.68	-0.71	-0.45	1.00	-0.71	0.09
wt	-0.87	0.89	0.66	-0.71	1.00	-0.17
qsec	0.42	-0.43	-0.71	0.09	-0.17	1.00



Data X

## Correlation Types: Pearson, Kendal, Spearman

Understanding Correlation in a different way



$$X \bullet Y = |X| |Y| \cos \Theta$$

## pandas.DataFrame.corr

`DataFrame.corr(method='pearson', min_periods=1)`

Compute pairwise correlation of columns, excluding NA/null values

**Parameters:**

- `method` : {`'pearson'`, `'kendall'`, `'spearman'`}
  - `pearson` : standard correlation coefficient
  - `kendall` : Kendall Tau correlation coefficient
  - `spearman` : Spearman rank correlation

**min\_periods** : `int, optional`  
Minimum number of observations required per pair  
Currently only available for pearson and spearman

**Returns:**

`y` : `DataFrame`

	mpg	disp	hp	drat	wt	qsec
mpg	1.00	-0.85	-0.78	0.68	-0.87	0.42
disp	-0.85	1.00	0.79	-0.71	0.89	-0.43
hp	-0.78	0.79	1.00	-0.45	0.66	-0.71
drat	0.68	-0.71	-0.45	1.00	-0.71	0.09
wt	-0.87	0.89	0.66	-0.71	1.00	-0.17
qsec	0.42	-0.43	-0.71	0.09	-0.17	1.00



Data X

## Pandas will create a correlation matrix with “columns”

```
In [15]: frame = pd.DataFrame(np.random.randn(1000, 5), columns=['a', 'b', 'c', 'd', 'e'])

In [16]: frame.ix[::2] = np.nan

# Series with Series
In [17]: frame['a'].corr(frame['b'])
Out[17]: 0.013479040400098775

In [18]: frame['a'].corr(frame['b'], method='spearman')
Out[18]: -0.0072898851595406371

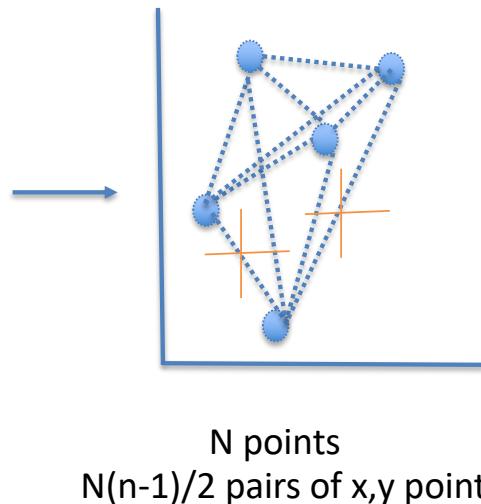
# Pairwise correlation of DataFrame columns
In [19]: frame.corr()
Out[19]:
      a          b          c          d          e
a  1.000000  0.013479 -0.049269 -0.042239 -0.028525
b  0.013479  1.000000 -0.020433 -0.011139  0.005654
c -0.049269 -0.020433  1.000000  0.018587 -0.054269
d -0.042239 -0.011139  0.018587  1.000000 -0.017060
e -0.028525  0.005654 -0.054269 -0.017060  1.000000
```



## Kendall Correlation

List of (x,y) points

No	X	Y
1	2	3
2	4	6
3	3	8
4	9	12



Concordant pairs: for  $(x_i, y_i)$  and  $(x_j, y_j)$ , where  $i \neq j$ ,  
 $x_i > x_j$  and  $y_i > y_j$       or       $x_i < x_j$  and  $y_i < y_j$

Disconcordant pairs: when the above is not true  
if  $x_i > x_j$  and  $y_i < y_j$   
or if  $x_i < x_j$  and  $y_i > y_j$

The Kendall  $\tau$  coefficient is defined as:

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{n(n - 1)/2}$$



$$r_s = \rho_{rg_X, rg_Y} = \frac{\text{cov}(rg_X, rg_Y)}{\sigma_{rg_X} \sigma_{rg_Y}}$$

where

- $\rho$  denotes the usual Pearson correlation coefficient, but applied to the rank variables.
- $\text{cov}(rg_X, rg_Y)$  is the covariance of the rank variables.
- $\sigma_{rg_X}$  and  $\sigma_{rg_Y}$  are the standard deviations of the rank variables.

Data ( $x=IQ, y=TV$ )

IQ, $X_i$	Hours of TV per week, $Y_i$
106	7
86	0
100	27
101	50
99	28
103	29
97	20
113	12
112	6
110	17

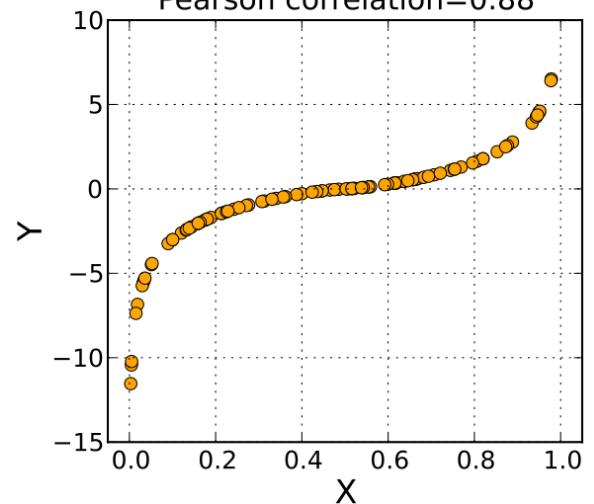
Order rows by X and  
Index X and Y in  
increasing order

$x \ y \ rgx \ rgy$

97	20	2	6	-4	16
99	28	3	8	-5	25
100	27	4	7	-3	9
101	50	5	10	-5	25
103	29	6	9	-3	9
106	7	7	3	4	16
110	17	8	5	3	9
112	6	9	2	7	49
113	12	10	4	6	36

## Spearman Correlation

Spearman correlation=1  
Pearson correlation=0.88



Then find  
Pearson Correlation  
of  $(rgx, rgy)$

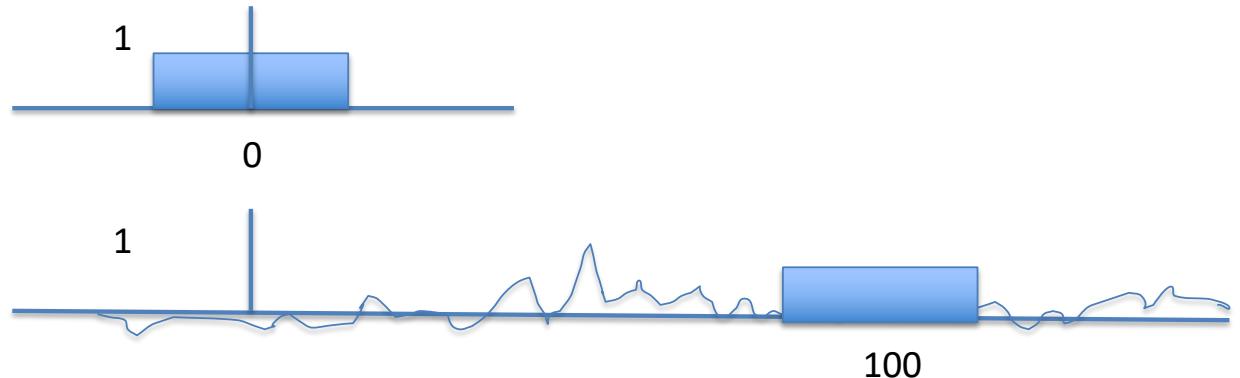
A Spearman correlation of 1 results when the two variables being compared are monotonically related, even if their relationship is not linear. This means that all data-points with greater x-values than that of a given data-point will have greater y-values as well. In contrast, this does not give a perfect Pearson correlation.

Wikipedia



## Correlation Matrix with multiple sources and time segments

Suppose this is  $x_1$   
as an array of numbers 0 0 1 1..1 0 0 0



Suppose this is  $x_2$

What is `np.corr(x1,x2[n:n+w])`?

Data X

End of Section

0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 1 1 1 0 0  
1 0 1 1 X 1 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 1 1 1 0 1 0 1 1 1 0 0  
1 Data 0 0 1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 0