Data Exploration

Numerical Data Analysis
Covariance / Correlation
Visualizing

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Lesson Objectives

- Learn to do explorative data analysis

- Learn some Statistics for Data Science

Data Types

Numerical Data Analysis
Covariance / Correlation
Visualizing

Data Types

Type	Description	Example
Continuous	Data can take any value within an interval Numeric, float, int	Exam score: 0 - 100
Discrete	Only integer values	Clicks per day
Categorical	Specific values from a set Enums, factors	Colors: Red, White, Blue States: AL, CA
Binary	Just two values, binary 0/1 or true/false	Transaction fraud or not
Ordinal	Categorical data, but with ordering	Grades: A, B, C, D A > B > C > D

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Structured Data

DataFrame	Spreadsheet like data
Feature	Column in the table Attribute / input / predictor / variable
Outcome	Predicted. Dependent variable / response / target / output
Records	A row in the DataFrame

Dataframe

Feature

Feature

Outcome

Row

	Income	Assets	Approved?
Application 1			
Application 2			

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Statistics Primer

Numerical Data Analysis

Covariance / Correlation

Visualizing

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Numerical Data Analysis

➔ Numerical Data Analysis

Covariance / Correlation

Visualizing

Numerical Data Analysis

- Analyze the following salary data.
  [30k, 35k, 22k, 70k, 50k, 55k, 45k, 40k, 25k, 42k, 60k, 65k]

- Sorting the data
  [22k, 25k, 30k, 35k, 40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k]

- Min : 22k
  Max : 70k
  → Range of data : 22k to 70k

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Mean

Mean	Sum (values) / total number of samples
Weighted Mean	Sum (values * weights) / total number of samples

- [30k, 35k, 22k, 70k, 50k, 55k, 45k, 40k, 25k, 42k, 60k, 65k]

- **Average / Mean**
  = Total sum of all salaries / (number of salaries)
  = (30k + 35k + 22k + 70k + 50k + 55k + 45k + 40k + 25k + 42k + 60k + 65k) / 12
  = 44.9k

$$\text{Mean} = \bar{x} = \frac{\sum_{i}^{n} x_i}{n}$$

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

$$\text{Weighted mean} = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i}^{n} w_i}$$

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Outliers & Trimmed Mean

Outliers	Extreme values. These influence plain mean. e.g. When Bill Gates walks into a bar, everyone's net worth goes up by few 100s of millions! ☺
Trimmed Mean Truncated Mean	Take mean, after dropping a number of extreme values from the bottom and top. 10% Trimmed Mean drops 10% of largest and 10% of smallest values and calculates mean in remaining 80% of data Used in competition scoring, to avoid one judge influencing the outcome. Example : [5, 6, 7, 8, 10] Mean = sum(5+6+7+8+10) / 5 = 7.2 Trimmed Mean = sum (6,7,8) / 3 = 7

Outliers / Trimmed Mean Example

- Consider this annual income data:
  [5k, 40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k, 400k]
- Mean income, considering all data
  = (5 + 40 + 42 + 45 + 50 + 55 + 60 + 65 + 70 + 400) / 10
  = **83.2**
- 10% trimmed mean
  ➔ drop lowest 10% (5k)
  ➔ drop highest 10% (400k)
  = (40 + 42 + 45 + 50 + 55 + 60 + 65 + 70) / 8
  = **53.4**
- As you can see, trimmed mean helps us deal with outliers

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Median (≠ Mean!)

- Median is the middle/center point of sorted data
- Example, find median of
  [50k, 55k, 40k, 42k, 45k, 65k, 70k, 75k, 60k]
- First sort the data
  [40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k, 75k]
- Find middle point :
  [40k, 42k, 45k, 50k, **55k**, 60k, 65k, 70k, 75k]
- If there are even number of records:
  [40k, 42k, 45k, 50k, **55k**, 60k, 65k, 70k, 75k, 80k]
- Median is average of both middle numbers : (55k + 60k)/2 = 57.5k

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Median, Mean and Outliers

- Consider this dataset
  [40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k]

- Mean = 53.4 = (40 + 42 + 45 + 50 + 55 + 60 + 65 + 70) / 8
  Median = 52.5 = (50 + 55) / 2

median

- [40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k]

mean

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Median, Mean and Outliers

- Now introduce an outlier (400k)
  [40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k, 400k]

- Mean = 91.89 = (40 + 42 + 45 + 50 + 55 + 60 + 65 + 70 + 400) / 9
  Median = 55

median

- [40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k, 400k]

mean

- Median is influenced less by outliers.

- Example: "Median house price in San Jose is 1M" (usually median is used, not mean/average)

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Mean : Sample Code (R)


```
a  = c (5,40,42,45,50,55,60,65,70,400)

summary(a)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    5.00   42.75   52.50   83.20   63.75  400.00

mean(a)
83.2

median(a)
52.5

# trimmed mean
mean(a, trim=0.1)
53.375
```

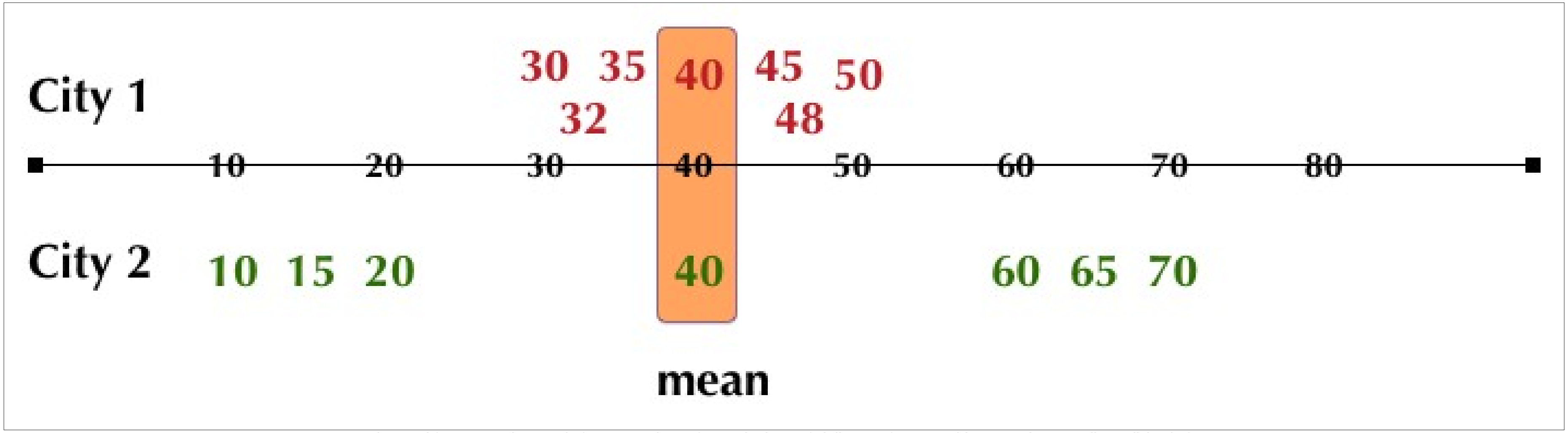
Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Mean : Sample Code (Python)


```python
import numpy as np
import pandas as pd
from scipy import stats

a = np.array([5,40,42,45,50,55,60,65,70,400])
# [  5  40  42  45  50  55  60  65  70 400]

np.mean(a)
#  83.2

stats.trim_mean(a,0.1)) # 10%
# 53.375

np.median(a)
# 52.5
```

Variability / Dispersion

- Consider sample annual incomes from two cities.
  `City1 = [ 30k, 32k, 35k, 40k, 45k, 48k, 50k ]`
  `City2 = [ 10k, 15k, 20k, 40k, 60k, 65k, 70k ]`
- Mean for both datasets is **40k**
- But it doesn't tell the whole story
- City2 data is more widely 'dispersed' than City1



Measuring Variability / Dispersion

Term	Description	Also known as
Range	Largest Value – Smallest Value	spread
Deviations	Difference between estimated value and actual value.	Residuals , errors
Variance	Sum(squared deviations from mean) / N N = number of samples	Mean-squared-error, MSE, S^2
Standard deviation	Square root of variance. (most used measurement of dispersion)	l2-norm, Euclidean norm
Percentile	The value such that P percent of the values take on this value or less and (100–P) percent take on this value or more.	quantile
Interquartile range	The difference between the 75th percentile and the 25th percentile.	IQR

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Variance - s^2, σ^2, var(x)

- Measures how far apart the data is spread out from their mean
- Symbols : s^2 , σ^2 , var(x)
- Method:
  - Find differences from X_i and mean (μ)
  - Square it
  - Add them all up
  - Divide by number of observations (N)

$$\mathrm{Var}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2,$$

- Properties
  - Variance is positive or zero (since we are squaring the diff)
  - If Variance of a dataset is zero, they all have the same value

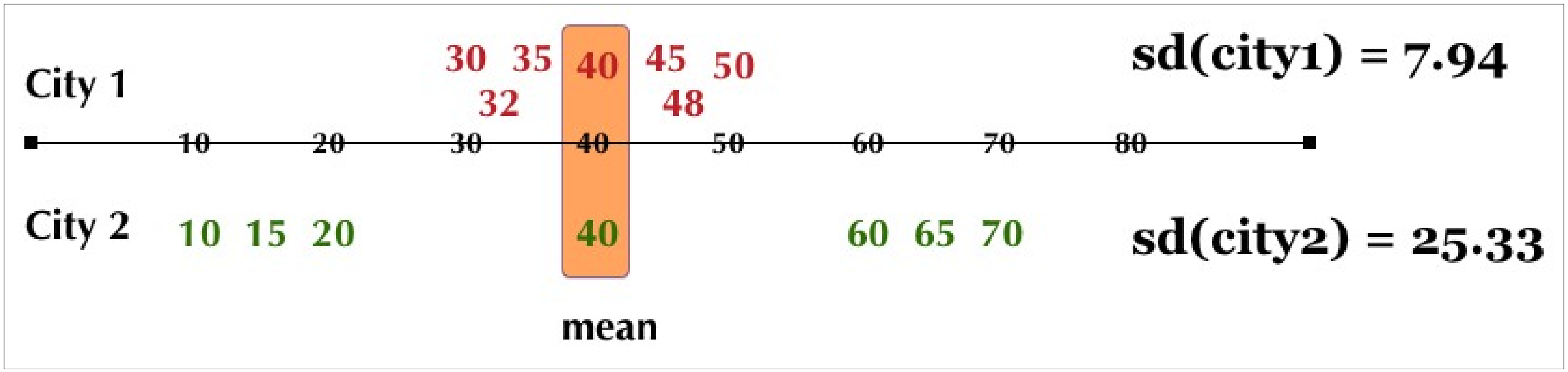
Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Standard Deviation (SD) : σ (sigma)

- SD is the most used measure of dispersion
- Measures how closely data values are clustered around mean
- Lower SD means values are closely clustered around mean
- Higher SD indicates larger dispersion

$$\text{Variance} = s^2 = \frac{\sum (x - \overline{x})^2}{n - 1}$$

$$\text{Standard deviation} = s = \sqrt{\text{Variance}}$$



Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Standard Deviation : Sample Code (R)

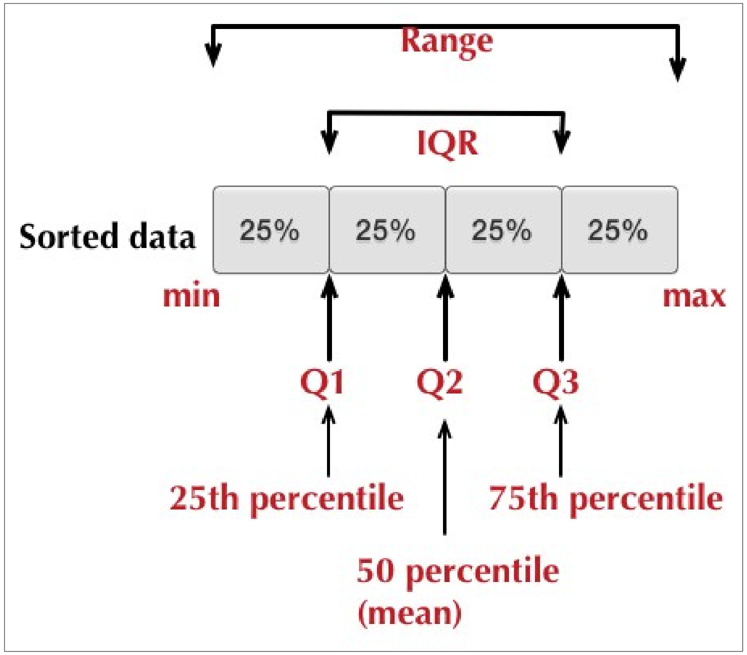

```
city1 = c(30,32,35,40,45,48,50)
city2 = c(10,15,20,40,60,65,70)

mean(city1)
40

mean(city2)
40

var(city1)
63

var(city2)
641.6667

sd(city1)
7.937254

sd(city2)
25.33114
```

Standard Deviation : Sample Code (Python)


```python
import numpy as np
import pandas as pd
from scipy import stats

city1 = np.array([30,32,35,40,45,48,50])
city2 = np.array([10,15,20,40,60,65,70])

### Mean
np.mean(city1)    # 40.0

np.mean(city2)   # 40.0

### variance
np.var(city1)   # 54.0

np.var(city2)   #  550.0  <- much larger than var(city1)

### Standard Deviation
np.std(city1)    #  7.34846922835

np.std(city2)   # 23.4520787991  <-- larger than sd(city1)
```

Quartiles

- Quartiles are summary measures that divide the ranked (sorted) data into four equal parts
- First quartile @ 25% mark = Q1 = 25th percentile
- Second quartile @ 50% mark = Q2 = 50th percentile
  - Equals to median'
- Third quartile @ 75% mark = Q3 = 75th percentile
- IQR = distance between Q3 and Q1



Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Quartiles

- Income data (sorted):
  [22k, 25k, 30k, 35k, 40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k]

- Approximately 25% of data is below Q1
  75% is more than Q1

22	25	30	35	40	42	45	50	55	60	65	70

33.75
1st Quartile /
25th percentile

43.50
2nd quartile / Median /
50th percentile

56.25
3rd quartile /
75th percentile

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Quartiles: Sample Code (R)

```
a  = c (5,40,42,45,50,55,60,65,70,400)

summary(a)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   42.75   52.50   83.20   63.75  400.00

quantile(a)
    0%    25%    50%    75%   100%
   5.00  42.75  52.50  63.75 400.00

quantile(a)["25%"]
25%
42.75

IQR(a)
21
```

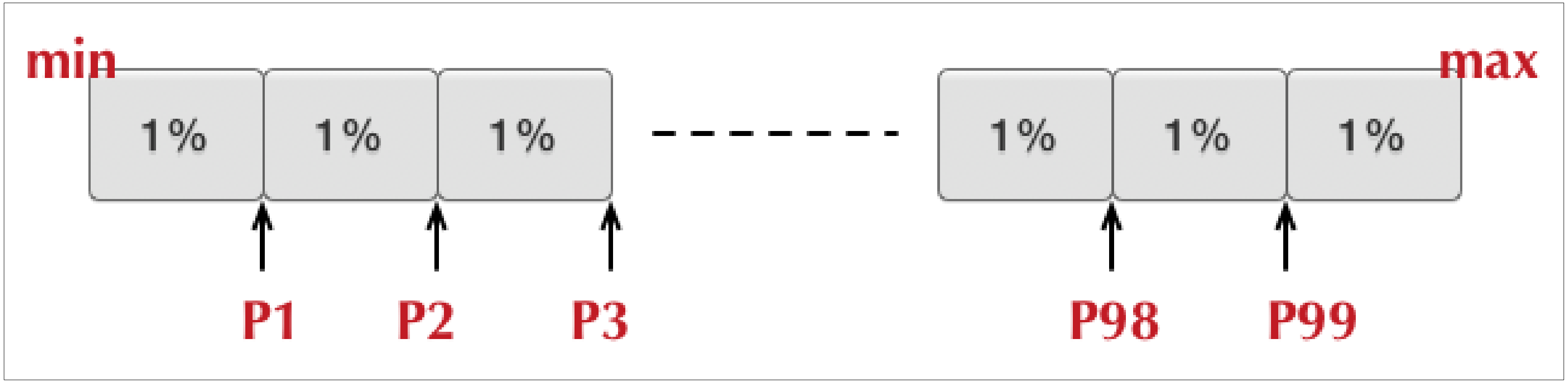
Quartiles: Sample Code (Python)


```python
import numpy as np

a = np.array([5,40,42,45,50,55,60,65,70,400])

# 20 pc
print (np.percentile(a, 20))
# 41.6

# q1, q2, q3
print (np.percentile(a, [25, 50, 75]))
# [ 42.75  52.5   63.75]
```

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Percentiles

- Percentiles are summary measures that divide the ranked (sorted) data into 100 equal parts
- k% of values $< P_k <$ (100-k) % of values
- 95th percentile: P_{95}
  - 95% of data below this point
  - 5% of data above this point



Calculating Percentiles Example

- Income data (sorted):
  [22k, 25k, 30k, 35k, 40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k]

- Finding k percentile point = k * N / 100
  N = number of data points = 12

- Find 30^{th} percentile point:
  = 30 * 12 / 100 = 3.6^{th} item = 4^{th} item (approx)
  = 35k
  = 30% of data is below 35k

- Finding percentile rank k
  = number of values less than X_k * 100 / N
  (N number of items)

- What is the percentile rank of income 52k
  = number of items less than 52k / 12 * 100
  = 8/12 * 100
  = 66.67%

Percentiles : Sample Code (R)


```
income = c(22, 25, 30, 35, 40, 42, 45, 50, 55, 60, 65, 70)

# find 30th percentile
quantile(income, c(0.3))
36.5
# 36.5k is the 30th percentile

# what percentile is income 52k
ecdf(income)(52)
0.6666667
# 52k is at 66.67%
```

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Percentiles Sample Code (Python)


```python
import numpy as np

a = np.array([5,40,42,45,50,55,60,65,70,400])

# 20 pc
print (np.percentile(a, 20))
# 41.6

# q1, q2, q3
print (np.percentile(a, [25, 50, 75]))
# [ 42.75  52.5   63.75]
```

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Relationship Between Two Variables

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Covariance

Numerical Data Analysis

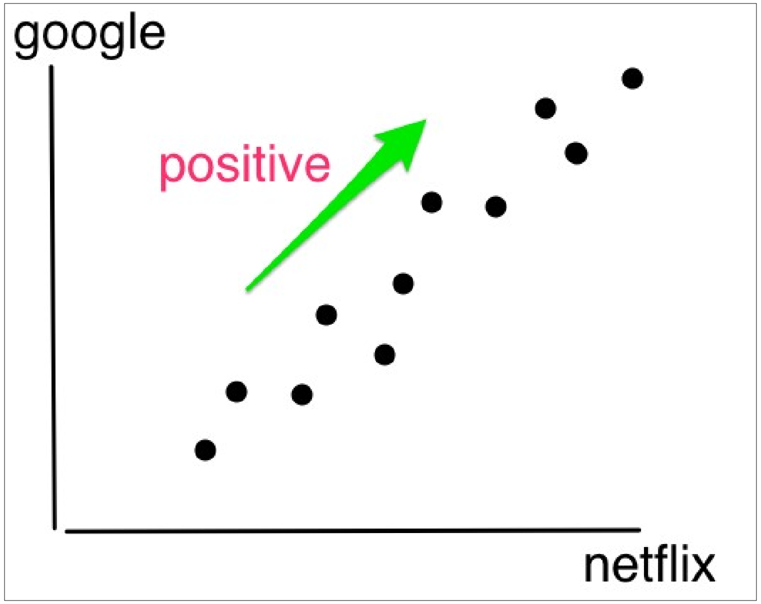
➔ Covariance / Correlation

Visualizing

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Covariance

- Variance, and Standard Deviation measures the data dispersion in a SINGLE variable
- How can we tell if two variables X & Y are related
- Here we see positive trend between Netflix stock price and Google stock pricing.
  When one goes up, other one goes up too



Covariance Formula

Formula to find the mean for X

$$\mu_x = \frac{\sum_{i=1}^{n} x_i}{n}$$

Formula to find the mean for Y

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

Formula to find covariance of X & Y

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{(n-1)}$$

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Covariance Example

x	y	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$
12	20	-9.3	-21.2	197.16
30	60	8.7	18.8	163.56
15	27	-6.3	-14.2	89.46
24	50	2.7	8.8	23.76
14	21	-7.3	-20.2	147.46
18	30	-3.3	-11.2	36.96
28	61	6.7	19.8	132.66
26	54	4.7	12.8	60.16
19	32	-2.3	-9.2	21.16
27	57	5.7	15.8	90.06
$\bar{x} = 21.3$	$\bar{y} = 41.2$			$\Sigma = 962.4$

Covariance Example

x	y	$(x_i - \bar{x})(y_i - \bar{y})$
12	20	197.16
30	60	163.56
15	27	89.46
24	50	23.76
14	21	147.46
18	30	36.96
28	61	132.66
26	54	60.16
19	32	21.16
27	57	90.06
$\bar{x} = 21.3$	$\bar{y} = 41.2$	$\Sigma = 962.4$

$$Cov(x, y) = s_{xy} = \frac{962.4}{n-1}$$

$$\frac{962.4}{9}$$

$$Cov(x, y) = 106.93$$

Covariance Summary

- We only care about the positive / negative / zero of covariance
  - Positive means, both variables move in the same direction
  - Negative → they move in opposite direction
  - Zero → no relation

- We don't care about the actual number (could be 2.3 or 2300) of covariance
  - It does NOT indicate the strength of the relationship
  - It has no upper / lower bound – it is not standardized
  - That is done by **Correlation** (later)

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Correlation

Numerical Data Analysis

➔ **Covariance / Correlation**

Visualizing

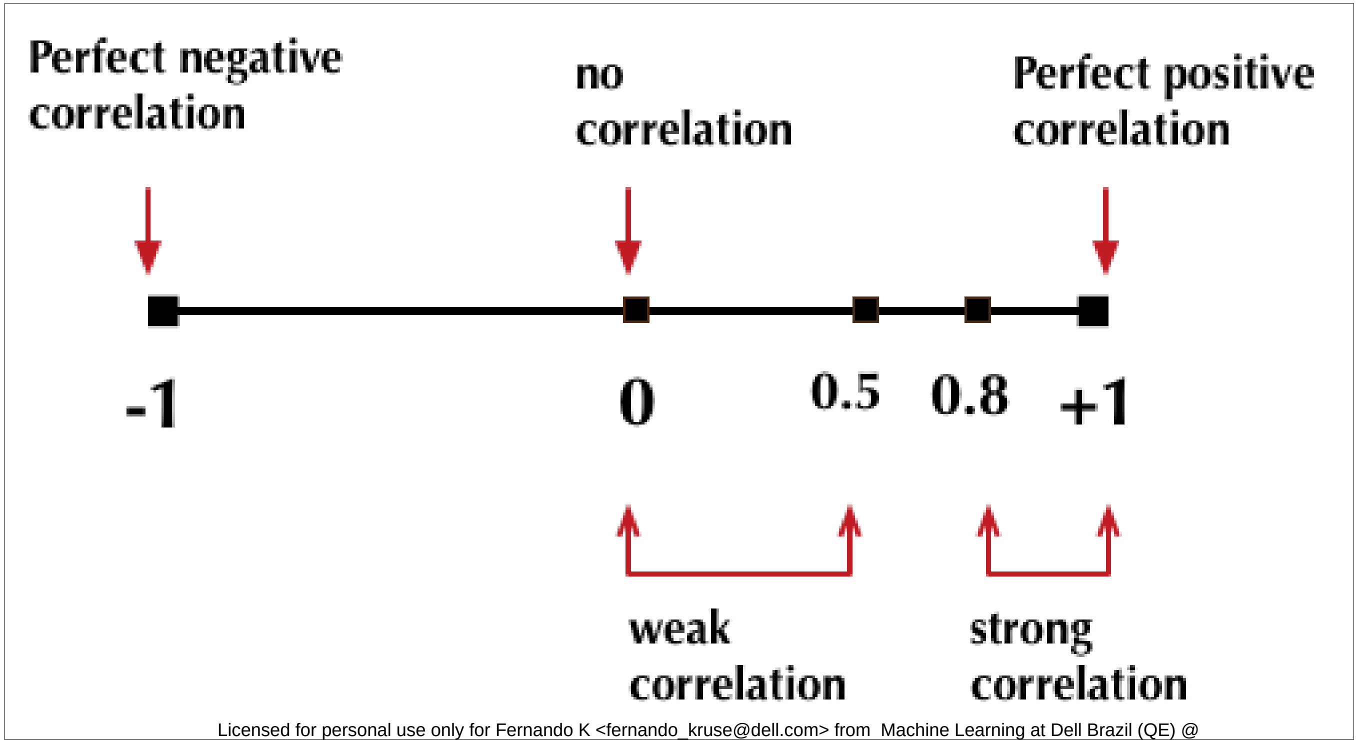
Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Correlation / Pearson Correlation Coefficient (r)

- Measures **strength and direction of linear relationship** between two variables
- Also known as **Pearson Correlation Coefficient** (in honor of its developer Karl Pearson)
- Values between -1 and +1 (standardized)
  (-1 <= r <= +1)
- If X & Y are positively related, r will be close +1
  - When X goes up Y goes up too
  - E.g. When 'years of experience' goes up 'salary' goes up too
- If X & Y are negatively related, r will be close to -1
  - When X goes up Y goes down
  - E.g. ??? (quiz for class)
- If no correlation between X & Y , then r will be close to 0

Correlation Coefficient

- **Perfect correlation** occurs when
  - r = -1 (negative)
  - r = +1 (positive)
  - This is when the data points all lie in straight line (regression line!)
- A correlation |r| >= 0.8 is considered **strong**
- A correlation |r| < 0.5 is considered **weak**.

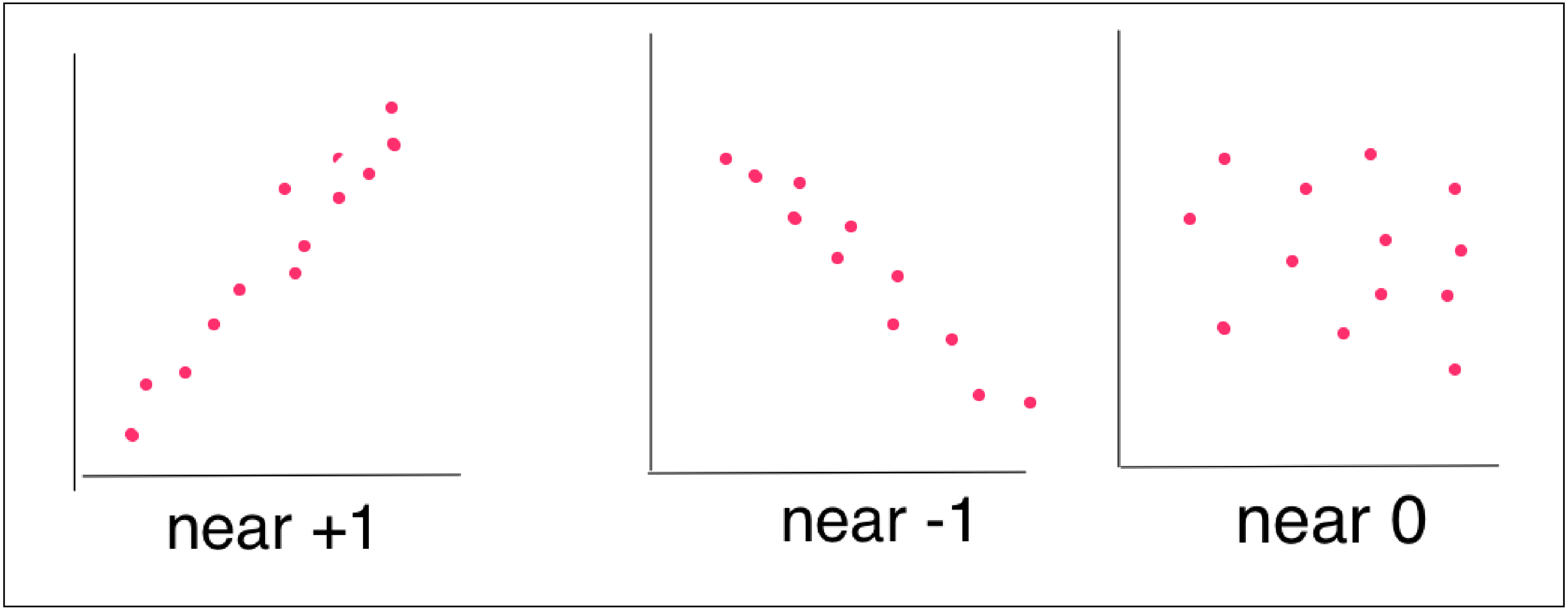


Covariance vs. Correlation

Covariance	Correlation
Measures linear relationship between two variables	(ditto)
Provides the **DIRECTION** (positive / negative / zero) of the linear relationship between 2 variables	Provides **DIRECTION and STRENGH**
No upper / lower bound. Not standardized	Between -1 and +1 standardized

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Correlation Patterns



Correlation Formula

$\sum$ *Math* $\int$

$$r = \frac{\text{Covariance (x,y)}}{\text{Standard Deviation (x) * Standard deviation (y)}}$$

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

$$r = \frac{1}{(n-1)} \sum \frac{(X-\mu_X)(Y-\mu_Y)}{\sigma_X \sigma_Y}$$

Correlation Summary

- Correlation is NOT Causation
- Two independent variables can have mathematical correlation, but have NO sensible connection / correlation in real life
- E.g : Number of cars sold vs number of pets adopted

Correlation Code (R)


```
bill = c(50,30,60,40,65,20,10,15,25,35)
tip = c(12,7,13,8,15,5,2,2,3,4)

cor(bill, tip)
# [1] 0.9522154

## strong correlation!
```

Correlation Code (Python)

```
import numpy as np
import pandas as pd

bills = np.array([50,30,60,40,65,20,10,15,25,35])
tips= np.array([12,7,13,8,15,5,2,2,3,4])

# correlation
print(np.corrcoef(bills,tips))
# array([[ 1.        ,  0.95221535],
#        [ 0.95221535,  1.        ]])
```

Covariance Matrix

Numerical Data Analysis

➔ **Covariance / Correlation**

Visualizing

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Covariance Matrix

- Covariance measures how two variables behave
- When we have more than two variables we create a covariance matrix
- The **diagonal is simply Variance** of that variable
  cov(x1,x1) = variance(x1) ☺
- The matrix is **symmetric**,
  cov(x1,x2) = cov(x2,x1)

	x1	**x2**	**x3**	**x4**
x1	Var(x1)	Cov(x1,x2)	Cov(x1,x3)	Cov(x1,x4)
x2	Cov(x2,x1)	Var(x2)	Cov(x2,x3)	Cov(x2,x4)
x3	Cov(x3,x1)	Cov(x3,x2)	Var(x3)	Cov(x3,x4)
x4	Cov(x4,x1)	Cov(x4,x2)	Cov(x4,x3)	Var(x4)

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Covariance Matrix Code (R)

```
a <- c(1,2,3,4,5,6)
b <- c(2,3,5,6,1,9)
c <- c(3,5,5,5,10,8)
d <- c(10,20,30,40,50,55)
e <- c(7,8,9,4,6,10)

m <- cbind(a,b,c,d,e)
m

cor_matrix = cor(m)
cor_matrix
```

	a	b	c	d	e
1	1	2	3	10	7
2	2	3	5	20	8
3	3	5	5	30	9
4	4	6	5	40	4
5	5	1	10	50	6
6	6	9	8	55	10

cor_matrix

	a	b	c	d	e
a	1.00000000	0.54470478	0.84515425	0.99607842	0.09897433
b	0.54470478	1.00000000	0.05370862	0.49341288	0.38786539
c	0.84515425	0.05370862	1.00000000	0.86126699	0.07319251
d	0.99607842	0.49341288	0.86126699	1.00000000	0.03538992
e	0.09897433	0.38786539	0.07319251	0.03538992	1.00000000

- Which of the variables are strongly correlated?

Covariance Matrix Code (Python)

```python
import numpy as np

a = np.array([1,2,3,4,5,6])
b = np.array([2,3,5,6,1,9])
c = np.array([3,5,5,5,10,8])
d = np.array([10,20,30,40,50,55])
e = np.array([7,8,9,4,6,10])

m = np.vstack([a,b,c,d,e])
print(m)

print(np.corrcoef(m))
```

```
# output : m
[[ 1  2  3  4  5  6]
 [ 2  3  5  6  1  9]
 [ 3  5  5  5 10  8]
 [10 20 30 40 50 55]
 [ 7  8  9  4  6 10]]
```

- Which of the variables are strongly correlated?

```
# output : correlation matrix
          a           b           c           d           e
a [[ 1.          0.54470478  0.84515425  0.99607842  0.09897433]
b  [ 0.54470478  1.          0.05370862  0.49341288  0.38786539]
c  [ 0.84515425  0.05370862  1.          0.86126699  0.07319251]
d  [ 0.99607842  0.49341288  0.86126699  1.          0.03538992]
e  [ 0.09897433  0.38786539  0.07319251  0.03538992  1.        ]]
```

Covariance Matrix Applications

- Financial economics
  - Figure our relationships with different stocks

- Principal Component Analysis (PCA)
  This will be covered in PCA section

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Data Analytics With R / Python

- **Ends here**
- Jump off to **'data-analytics-R/slides/Analytics.pptx'**

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Lab Preparation for Machine Learning Class

Please follow instructions in
0-Labs-Prep.pptx

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Optional Lab 2.1 & 2.2: Basic Numpy, Pandas

Lab

- **Overview**:
  Get familiar with Numpy and Pandas

- **Approximate time**:
  10 mins

- **Instructions**:
  - 2.1 : Numpy
  - 2.2 : Pandas

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Optional Lab 3.1 : Statistics

Lab

- **Overview**:
  Learn basic statistics functions

- **Approximate time**:
  10 mins

- **Instructions**:
  - **3.1 Basics/stats**
  - Follow appropriate instructions for R / Python / Spark

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Visualizing Data

Numerical Data Analysis

Covariance / Correlation

➔ **Visualizing**

Visualizing Data

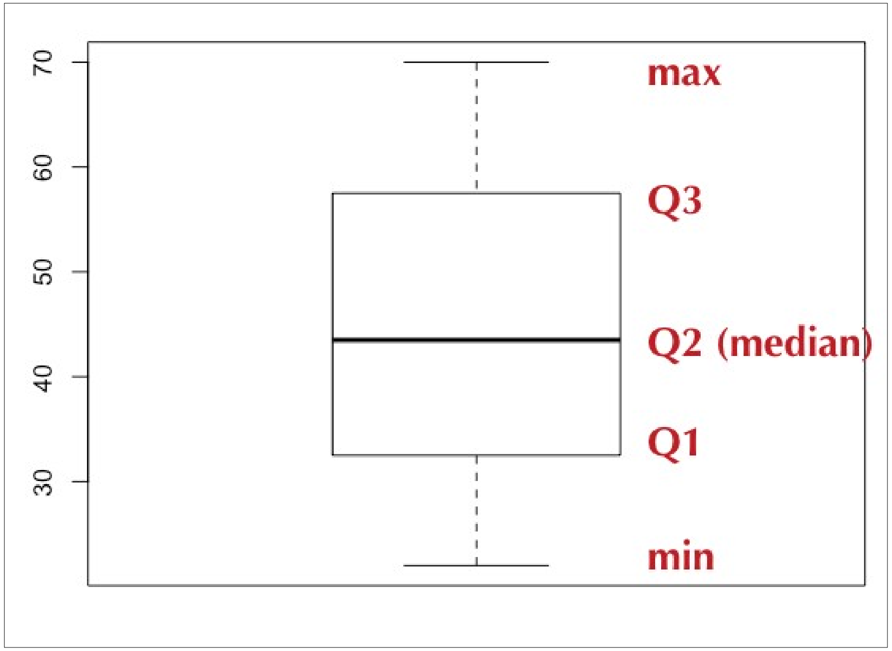
Method	Description
Boxplot	A quick way to visualize the data
Frequency table	Count number of data points that fall into intervals (bins)
Histogram	Plot of frequency table
Density plot	Smoothed version of histogram (Kernel Density Estimate)

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Boxplot / Box-and-Whisker Plot

- Boxplot displays 5 measures : min, Q1, Q2 (median), Q3, max
- Smallest / Largest values are measured within upper/lower fences
- Fences are 1.5 times IQR

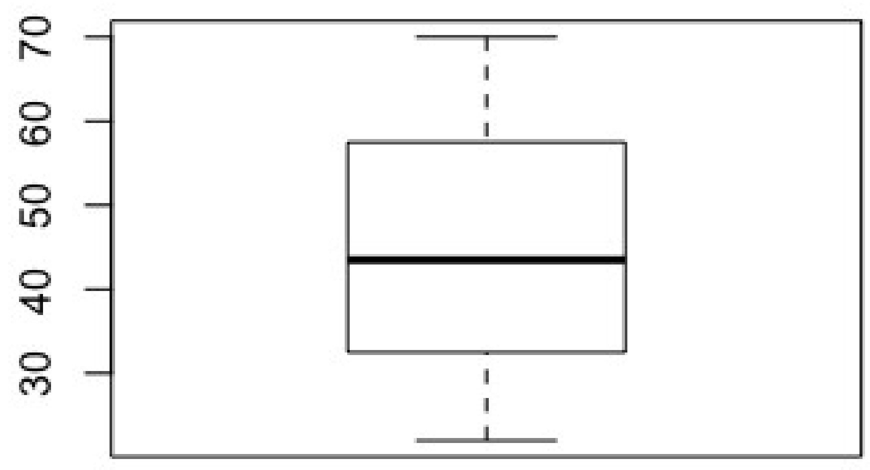
- Income data (sorted):
  [22k, 25k, 30k, 35k, 40k, 42k, 45k, 50k, 55k, 60k, 65k, 70k]



Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

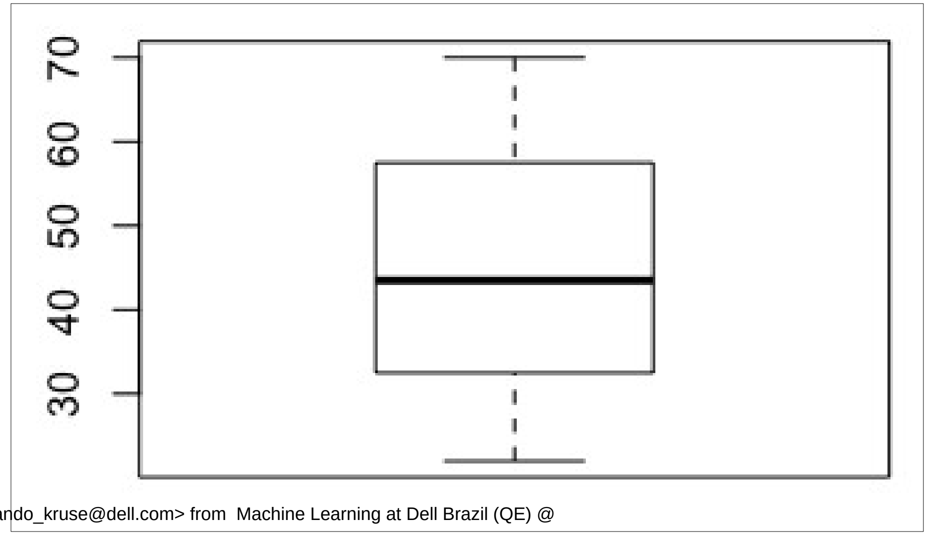
BoxPlot : Sample Code (R)

```
income = c(22, 25, 30, 35, 40, 42, 45, 50, 55, 60, 65, 70)

bp = boxplot(income)

bp

$stats
      [,1]
[1,]  22.0
[2,]  32.5
[3,]  43.5
[4,]  57.5
[5,]  70.0

$n
12
```



BoxPlot : Sample Code (Python)


```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

salaries = np.array([22, 25, 30, 35, 40, 42, 45, 50, 55, 60, 65, 70])

plt.boxplot(salaries)
```



Histogram (R)

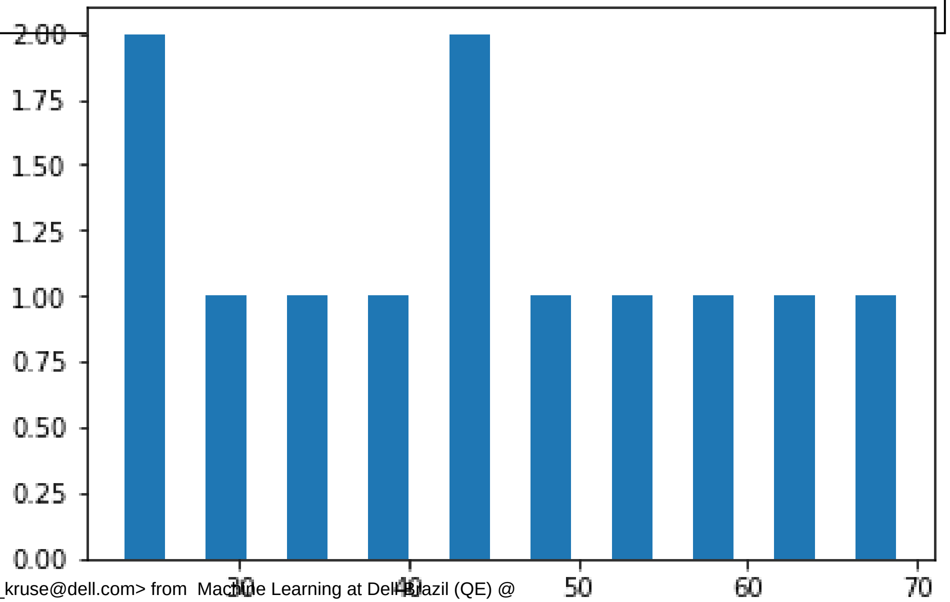
- Histogram counts data points per bin

```
income = c(22, 25, 30, 35, 40, 42, 45, 50, 55, 60, 65, 70)
hist(income)
```

Histogram (Python)

- Histogram counts data points per bin

```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

salaries = np.array([22, 25, 30, 35, 40, 42, 45, 50, 55, 60,
65, 70])

plt.hist(salaries, rwidth=0.7)
```

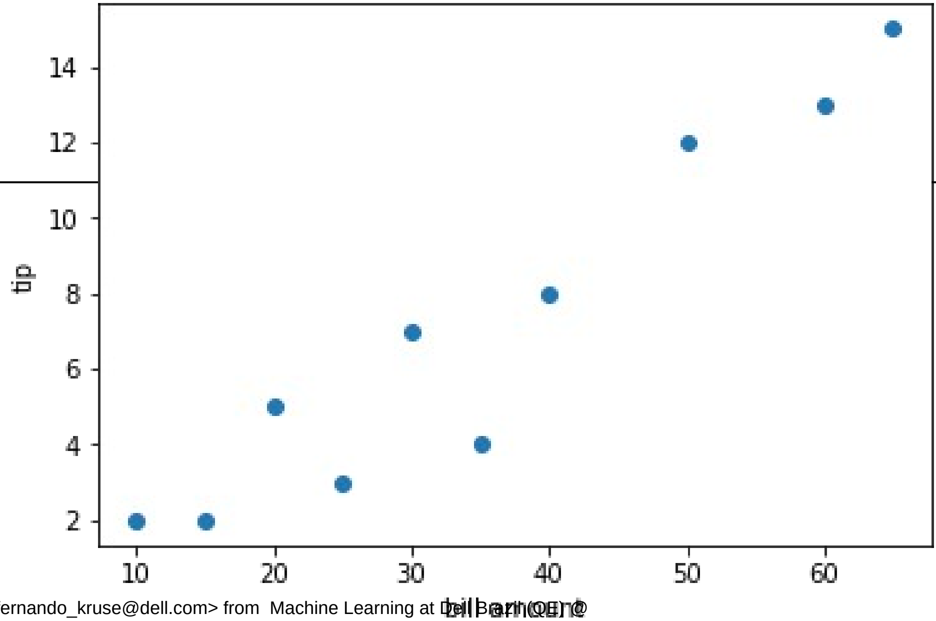


Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12

Scatter Plot (Python)

- Histogram counts data points per bin

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

bills = np.array([50,30,60,40,65,20,10,15,25,35])
tips= np.array([12,7,13,8,15,5,2,2,3,4])

plt.xlabel("bill amount")
plt.ylabel("tip")
plt.scatter(bills, tips)
```



Lab 3.2 : Visualizing

Lab

- **Overview**:
  Learn basic plot functions

- **Builds on previous labs**:

- **Approximate time**:
  10 mins

- **Instructions**:
  - **3.2 : basic/visualizing**
  - Follow appropriate instructions for R / Python / Spark

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Lab 3.3: Data Cleanup

Lab

- **Overview**:
  Cleaning up data, getting it ready for analytics

- **Approximate Time**:
  10 – 15 mins

- **Instructions**:
  - '3.3 **exploration/data-cleanup**' lab for Python / R / Spark

- **To Instructor**:
  Demo this lab on screen first, and explain the results

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

To Instructor

- Option 1 : STOP here, if continuing onto ‘ML-Concepts’
- Option 2 : continue to next 2 labs, if this is standalone module

[Optional] Lab 3.4: Exploring Dataset

Lab

- **Instructor, If covering ML-Concepts, do this at the end of Part-1 ML-Concepts**

- **Overview**:
  Explore a dataset

- **Approximate Time**:
  10 – 15 mins

- **Instructions**:
  - **3.4: 'exploration/explore-house-sales'** lab for Python / R / Spark

- **To Instructor**:
  Demo this lab on screen first, and explain the results

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

BONUS Lab 3.5: Graphing And Visualizing

Lab

- **Overview**:
  Visualize house-sales dataset

- **Approximate Time**:
  10 – 15 mins

- **Instructions**:
  - **3.5: 'exploration/visualize-house-sales'** lab for Python / R / Spark

- **To Instructor**:
  Demo this lab on screen first, and explain the results

Further Reading

- "Practical Statistics for Data Scientists" O'Reilly books
- http://www.cabrillo.edu/~evenable/ch03.pdf
- Fantastic YouTube video series on Statistics by Brandon Foltz
  - Covariance : https://www.youtube.com/watch?v=xGbpuFNR1ME
  - Correlation : https://www.youtube.com/watch?v=4EXNedimDMs
  - Covariance Matrix : https://www.youtube.com/watch?v=locZabK4Als

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @ 2019-03-12

Review Questions

Licensed for personal use only for Fernando K <fernando_kruse@dell.com> from Machine Learning at Dell Brazil (QE) @
2019-03-12