cyber & data

"From bits to information"

Introduction to Data Science

Outline

- Combinations/Permutations.
- Sets.
- Probabilities.
- Bayes.
- Correlations.
- Distributions.

cyber & data

"From bits to information"

Combinations/ Permutations

Combinations/ Permutations

[UK, France, Germany, UK, France, Ireland, France, Germany, Ireland and UK, Germany, Ireland]

$$^{4}C_3 = \frac{4!}{3!(4-3)!} = 4$$

$$^{n}C_k = \frac{n!}{k!(n-k)!}$$

$$^{n}P_k = \frac{n!}{(n-k)!}$$

[UK, France, Germany, UK, France, Ireland, UK, Germany, France, UK, Germany, Ireland, UK, Ireland, France, UK, Ireland, Germany, ... France, Germany, Ireland]

$$^{4}P_3 = \frac{4!}{(4-3)!} = 24$$

Combinations/ Permutations

```python
import math

n=4
k=3

Combinations = int(math.factorial(n)/math.factorial(k)/math.factorial(n-k))

print ("For {} from {}".format(n,k))

print ("Combinations ",Combinations)

Permuations = int(math.factorial(n)//math.factorial(n-k))
print ("Permuations: ",Permuations)
```

```
For 4 from 3
Combinations  4
Permuations:  24
```

Code

Combinations/ Permutations

main.py

```python
from itertools import permutations, combinations


countries = ["UK","France","Germany","Ireland"]
print("Original Cofllection: ",countries)

print("Combinations:")
res=combinations(countries,3)
for r in res:
  print(r)

print("\nPermutations:",)
res=permutations(countries,3)
for r in res:
  print(r)
```

https://perm.billbuchanan.repl.run

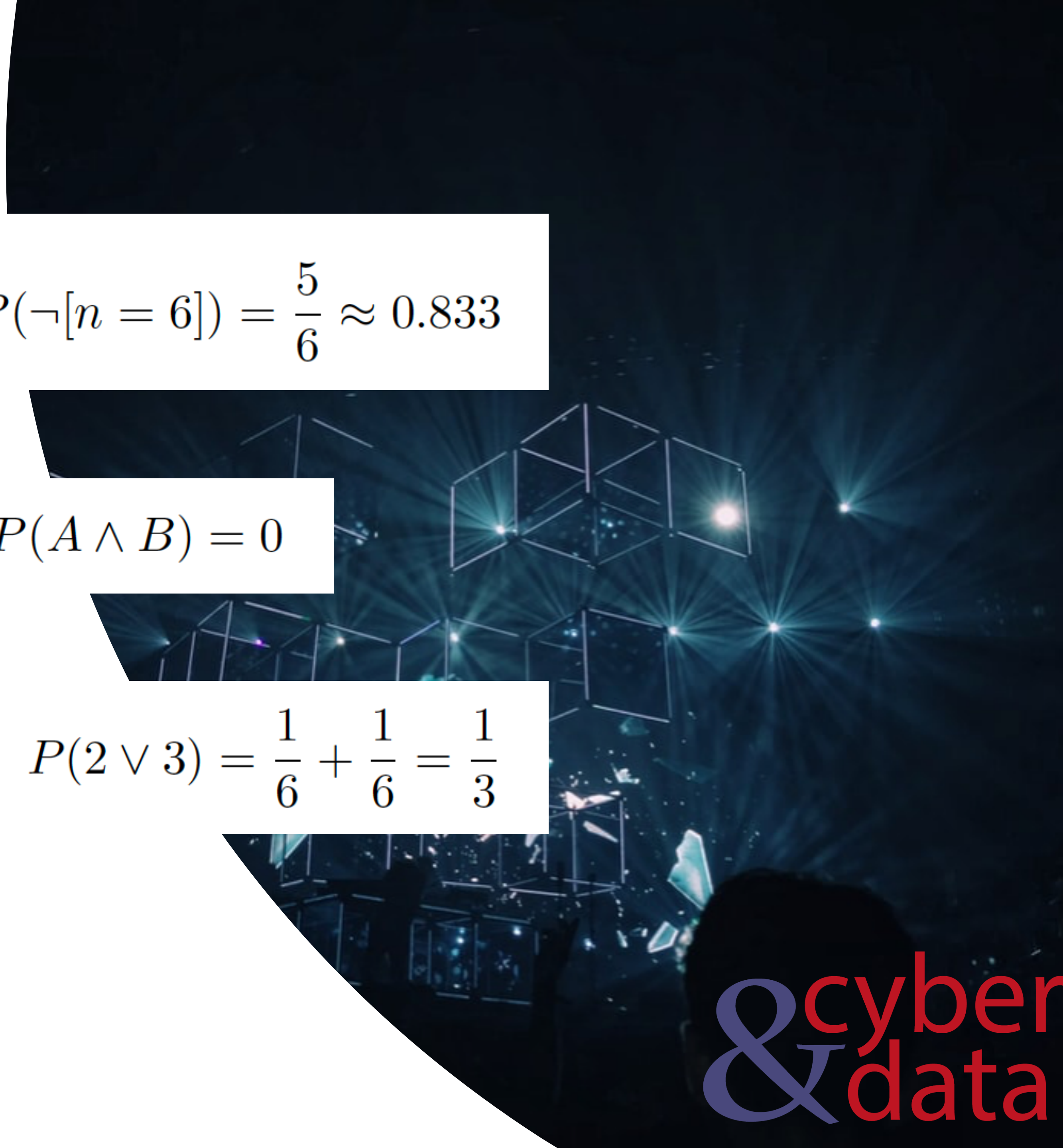
```
Original Cofllection:  ['UK', 'France', 'Germany', 'Ireland']
Combinations:
('UK', 'France', 'Germany')
('UK', 'France', 'Ireland')
('UK', 'Germany', 'Ireland')
('France', 'Germany', 'Ireland')

Permutations:
('UK', 'France', 'Germany')
('UK', 'France', 'Ireland')
('UK', 'Germany', 'France')
('UK', 'Germany', 'Ireland')
('UK', 'Ireland', 'France')
('UK', 'Ireland', 'Germany')
('France', 'UK', 'Germany')
('France', 'UK', 'Ireland')
('France', 'Germany', 'UK')
('France', 'Germany', 'Ireland')
('France', 'Ireland', 'UK')
('France', 'Ireland', 'Germany')
('Germany', 'UK', 'France')
('Germany', 'UK', 'Ireland')
('Germany', 'France', 'UK')
('Germany', 'France', 'Ireland')
('Germany', 'Ireland', 'UK')
('Germany', 'Ireland', 'France')
('Ireland', 'UK', 'France')
('Ireland', 'UK', 'Germany')
('Ireland', 'France', 'UK')
('Ireland', 'France', 'Germany')
('Ireland', 'Germany', 'UK')
('Ireland', 'Germany', 'France')
```

Code

“From bits to information”

Probabilities

Probability



$$P(n) = \frac{1}{6} \approx 0.167$$

$$P(\neg[n = 6]) = \frac{5}{6} \approx 0.833$$

$$P(A \wedge B) = P(A).P(B)$$

$$P(A \wedge B) = 0$$

$$P(A \vee B) = P(A) + P(B)$$

$$P(2 \vee 3) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$$

$$P(A \wedge B) = P(A)P(B|A)$$

cyber & data

"From bits to information"

Sets

Sets

Symbol	Symbol Name	Description
$\mid$	such that	so that
$A \cap B$	intersection	objects belong to A and set B
$A \cup B$	union	objects belong to A or set B
$A \subseteq B$	subset	subset has fewer elements or equal to the set
$\in$	belongs to	when an object is within a set
$\notin$	does not belong to	when an object is not in a set

Players — mike, fred, bert

Spectators — ian, michael, mike

Thus $A \cap B$ — mike and $A \cup B$ —mike,fred,bert,ian,michael. Then 'mike' $\in$ Players, and 'ian' $\notin$ Players.

Sets

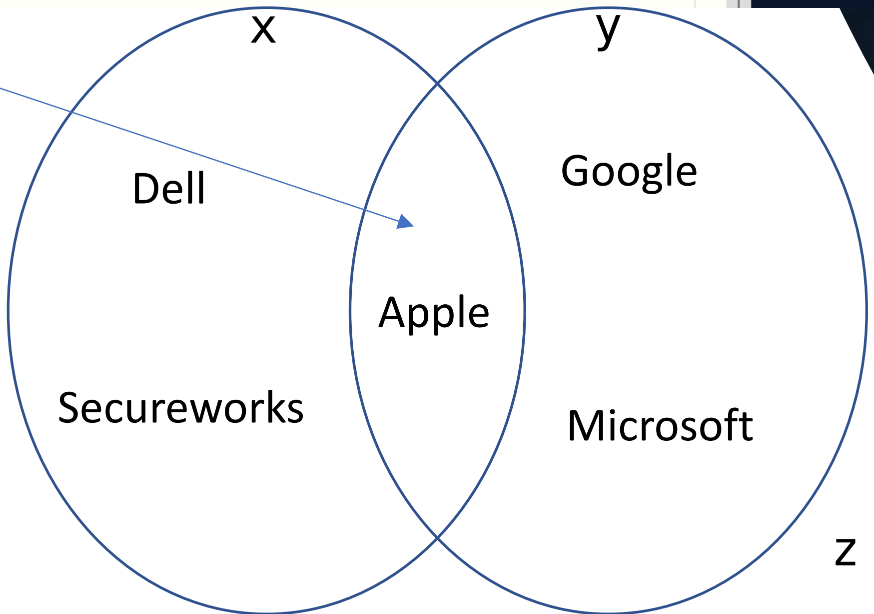
main.py

```python
x = {"Apple", "Dell", "Secureworks"}
y = {"Google", "Microsoft", "Apple"}

# z = x ∩  y
# z = x AND y
z = x.intersection(y)
print("Intersection : ",z)

# z = x ∪ y
# z = x OR y
z = x.union(y)
print("Union: ",z)

# z = x − y
z=x.difference(y)
print("Difference: ",z)
```

https://set.billbuchanan.repl.run

```
Intersection :  {'Apple'}
Union:  {'Microsoft', 'Google', 'Secureworks', 'Dell', 'Apple'}
Difference:  {'Dell', 'Secureworks'}
```

$z = x \cap y$



"Google" $\in$ y
"Google" $\notin$ x

Code

$z = x \cup y$

“From bits to information”

Bayesian

Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

The probability of A knowing B is the probably of B if we know A, multiplied by the probably of A and divided by the probability of B.

Eg A={‘Sunny’, ’Overcast’, ’Raining’}
B={‘Cloudy’, ’No Clouds’}

Eg
P(‘Sunny’)=0.3
P(’Clouds’)=0.2
P(‘Clouds’ | ‘Sunny’) = 0.5

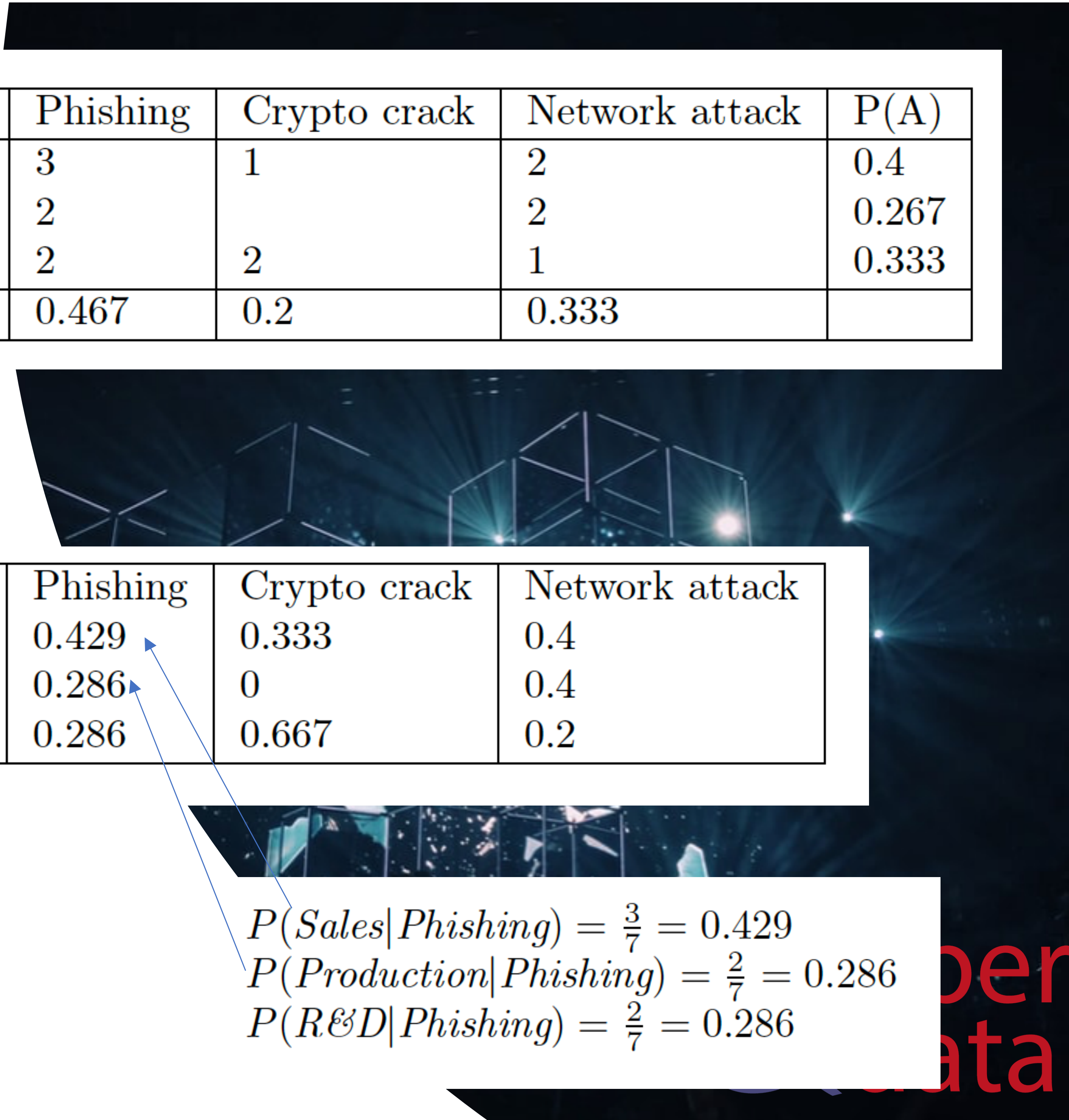
Then:

P(‘Sunny’ | ‘Clouds) = 0.5 * 0.3/0.2 = 0.75

Bayes $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

Systems (A)	Hack (B)
Production	Phishing
Production	Network attack
Production	Phishing
Production	Network attack
R&D	Network attack
R&D	Crypto crack
R&D	Crypto crack
R&D	Phishing
R&D	Phishing
Sales	Phishing
Sales	Network attack
Sales	Phishing
Sales	Crypto crack
Sales	Phishing
Sales	Network attack

Systems	Phishing	Crypto crack	Network attack	P(A)
Sales	3	1	2	0.4
Production	2		2	0.267
R&D	2	2	1	0.333
P(B)	0.467	0.2	0.333	



P(A | B)

Systems	Phishing	Crypto crack	Network attack
Sales	0.429	0.333	0.4
Production	0.286	0	0.4
R&D	0.286	0.667	0.2

$$P(Sales|Phishing) = \frac{3}{7} = 0.429$$
$$P(Production|Phishing) = \frac{2}{7} = 0.286$$
$$P(R\&D|Phishing) = \frac{2}{7} = 0.286$$

Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Systems (A)	Hack (B)
Production	Phishing
Production	Network attack
Production	Phishing
Production	Network attack
R&D	Network attack
R&D	Crypto crack
R&D	Crypto crack
R&D	Phishing
R&D	Phishing
Sales	Phishing
Sales	Network attack
Sales	Phishing
Sales	Crypto crack
Sales	Phishing
Sales	Network attack

Systems	Phishing	Crypto crack	Network attack	P(A)
Sales	3	1	2	0.4
Production	2		2	0.267
R&D	2	2	1	0.333
P(B)	0.467	0.2	0.333	

P(A | B)

Systems	Phishing	Crypto crack	Network attack
Sales	0.429	0.333	0.4
Production	0.286	0	0.4
R&D	0.286	0.667	0.2

$$P(Crypto|Sales) = \frac{P(Sales|Crypto) \times P(Crypto)}{P(Sales)} \tag{13}$$

$$P(Cryptocracking|Sales) = \frac{0.333 \times 0.214}{0.429} = 0.166 \tag{14}$$

Table 5: $P(B|A)$

Attack	Sales	Production	R&D
Phishing	0.501	0.5	0.401
Crypto crack	0.167	0	0.401
Network attack	0.333	0.499	0.2

ber
ata

Bayes

(subject_field_characters=34,words=100)

```python
from sklearn.naive_bayes import GaussianNB

import numpy as np
X=np.array([[34,100],[80,230],[70,400],[55,20],[28,30],[20,25],[18,40]])

Y=np.array([1,1,1,0,0,0,0])
print ("Samples")
print (X)
print (Y)

binary_class = GaussianNB()

binary_class.fit(X, Y)


print (binary_class.score(X, Y))
data = np.array([[28, 30], [40, 100], [4, 500], [10, 10]])
print (binary_class.predict(data))
```

So let's say that we have a phishing email detector, and we take samples and determine the number of characters in the subject field, and the number of words in the email. Let say that the samples for true phishing are (subject_field_characters=34,words=100), (80,230), and (70,400), and the samples for not phishing are (55,20), (38,30), (20,25) and (18,40). In case the first variable is the number of characters in the subject field, and the second one is the number of words in the email. We can now go ahead and define these, and use a GaussianNB() classifier, and then fit.

```
Samples
[[ 34 100]
 [ 80 230]
 [ 70 400]
 [ 55  20]
 [ 28  30]
 [ 20  25]
 [ 18  40]]
[1 1 1 0 0 0 0]
1.0
[0 1 1 0]
```

Code

And where the classifier has identified that (28, 30) and (10, 10) are not phishing emails, and (40, 100), (4, 500) are.

cyber & data

“From bits to information”

Bayesian Decision Engine

Bayes

Battery (b) - where the battery is empty or full.

Gauge (g) - where we have fuel or empty.

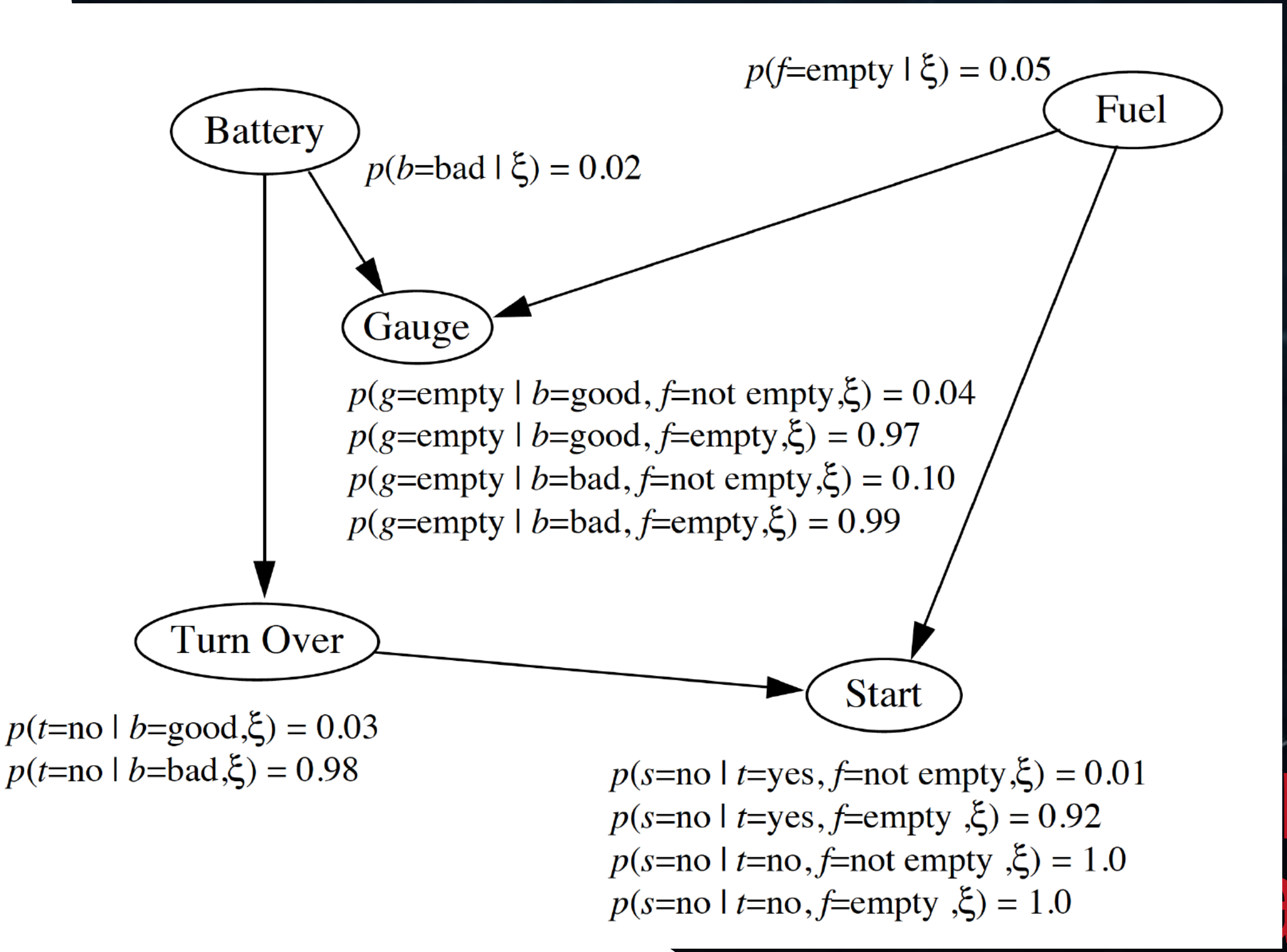
Turn over (t) - where the engine will turn over or not

Start (s) - where the engine starts or not.

$$p(x_1, x, 2|\epsilon) = p(x_2|x_1, \epsilon)p(x_1|\epsilon)$$

$$p(x_1, \ldots x_n|\epsilon) = \prod_{i=1}^{n} p(x_i|x_1 \ldots x_n, \epsilon)$$

The network is then defined as a **directed acrylic graph** of conditional interdependence, and where an arc is drawn from a cause to an effect. In Figure, the Gauge is a direct casual effect of Battery and Fuel, the Turn Over is the direct casual effect of Battery, and Start is the direct casual effect for Fuel and Turn Over. The probabilities associated with each of the nodes is defined beside the node.



cyber & data

"From bits to information"

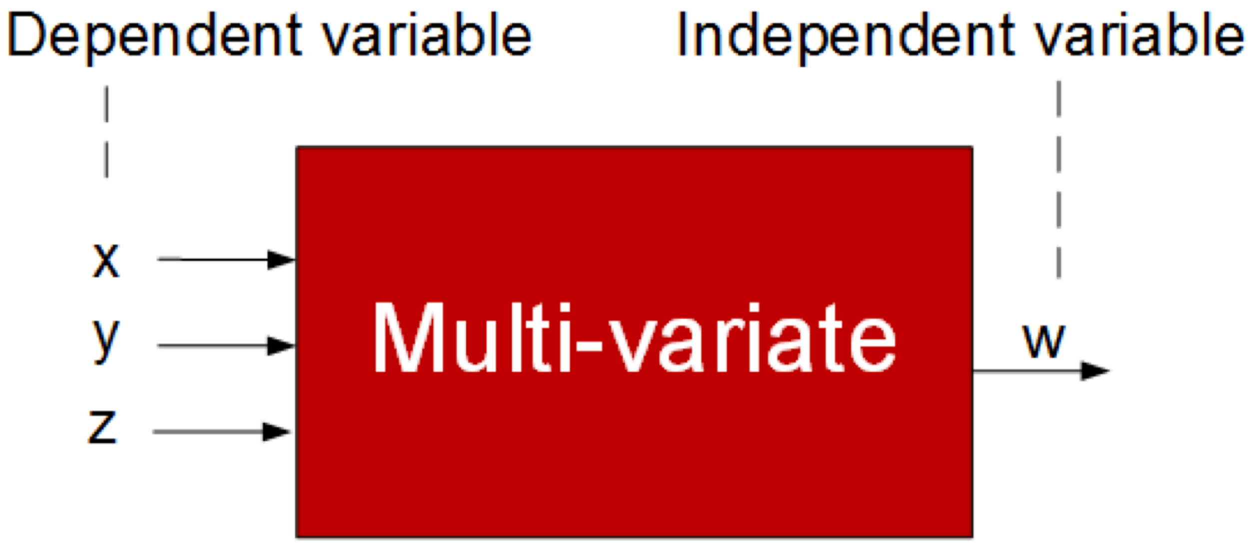
Correlation

Variables

$$y = f(x)$$

$$\text{eg } y = 5x$$

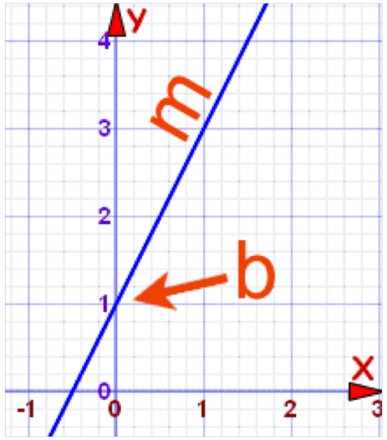
$$y = 5x^2 + 5$$

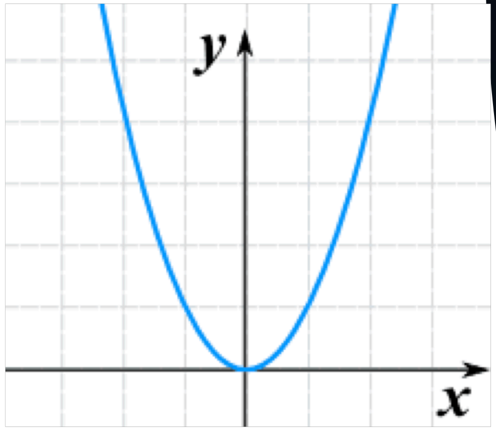


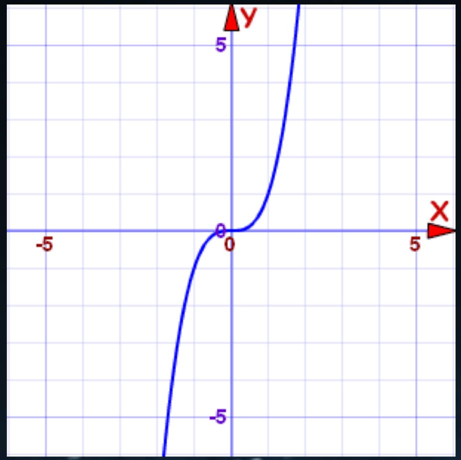
$$w = f(x, y, z)$$

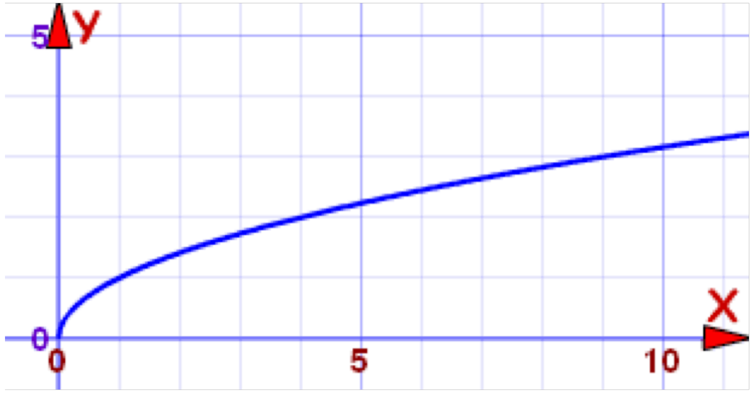
$$\text{eg } w = 3x + 4y - 2z$$

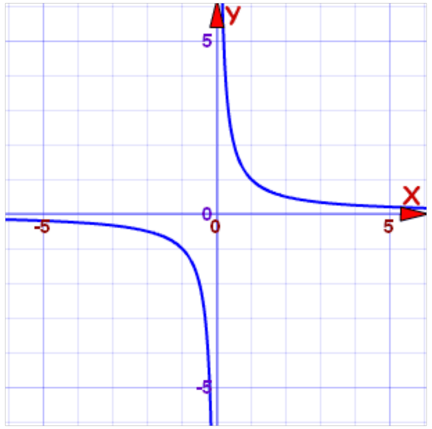
Correlation

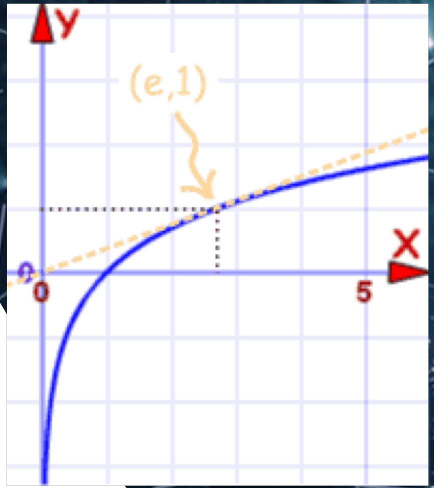


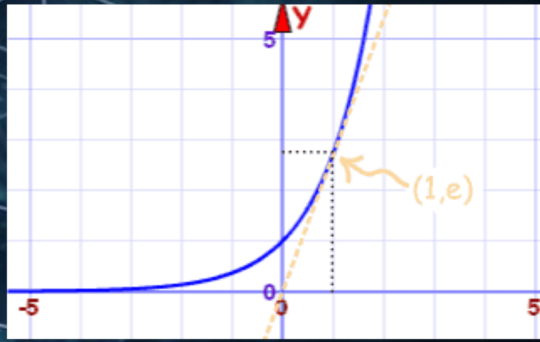


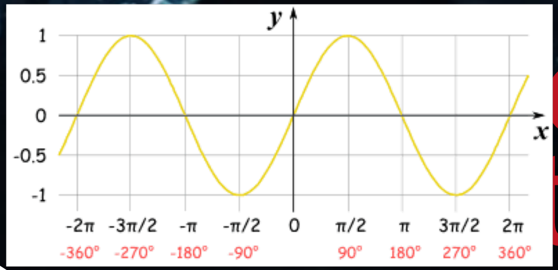






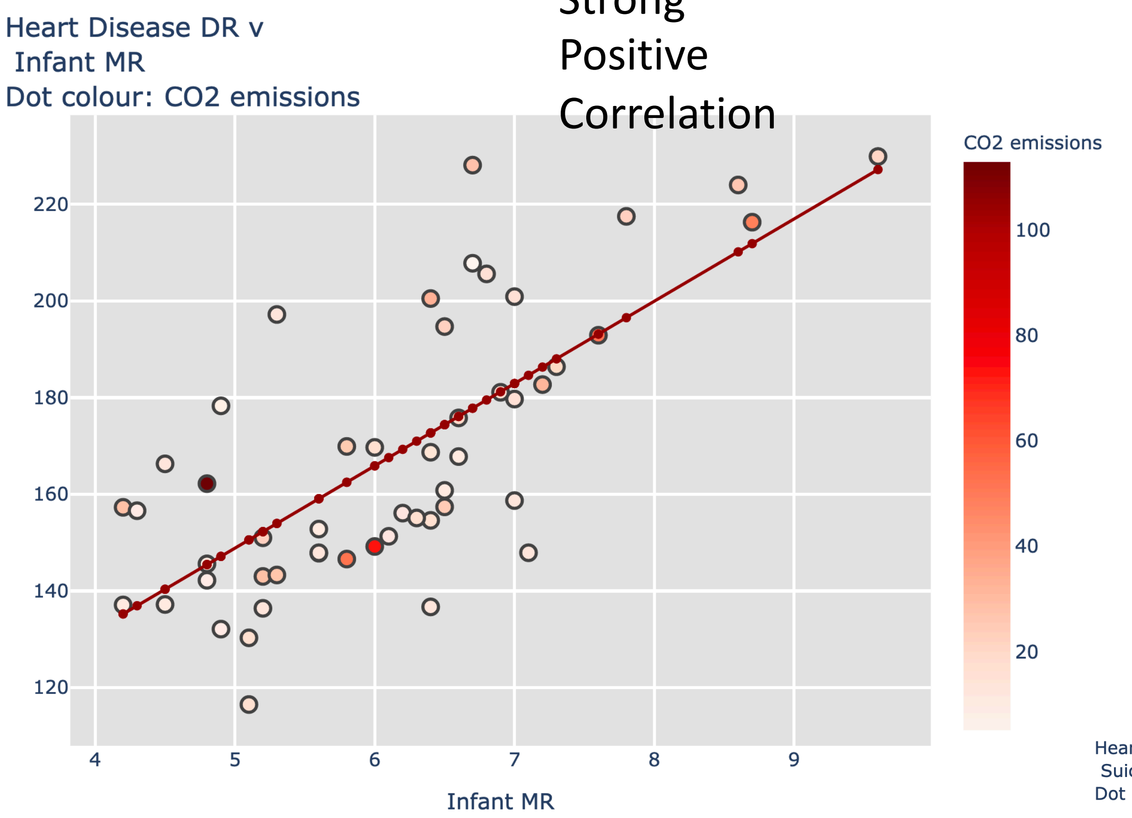




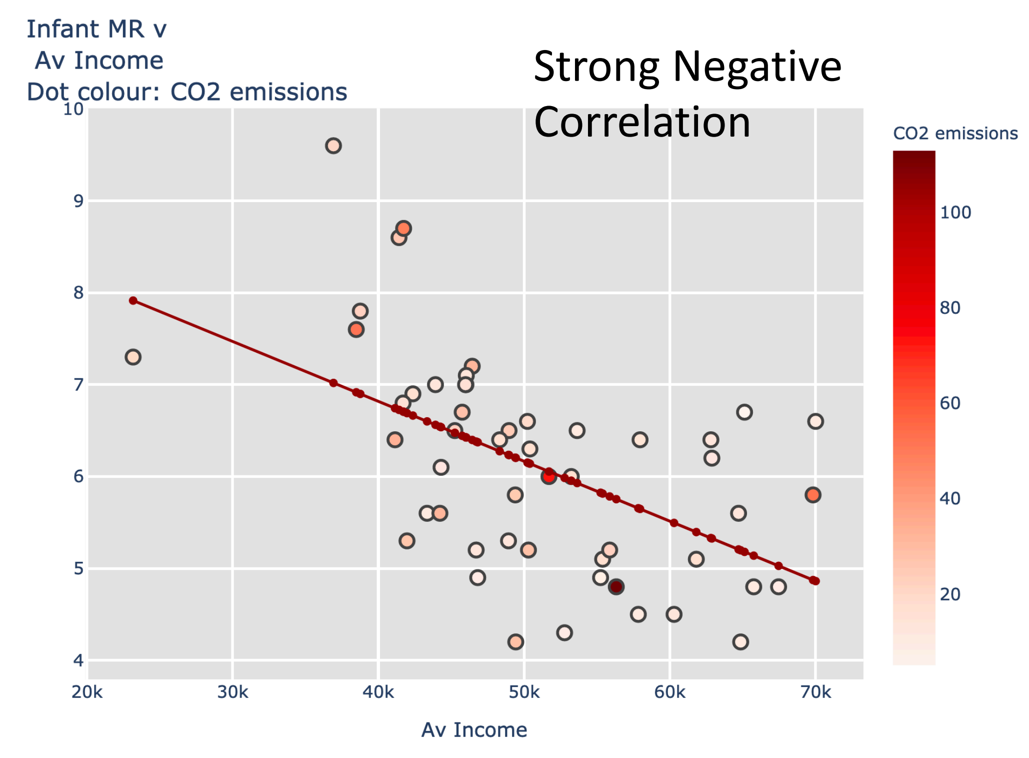


Correlation

Strong Positive Correlation

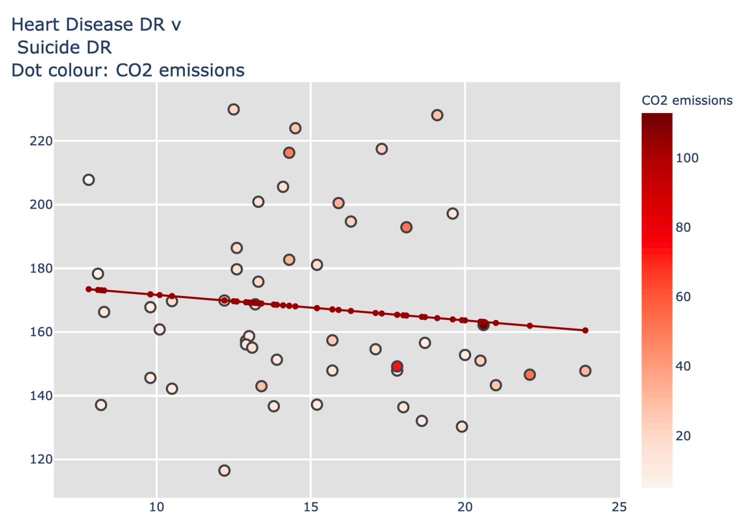


Strong Negative Correlation



Example

LIttle Correlation

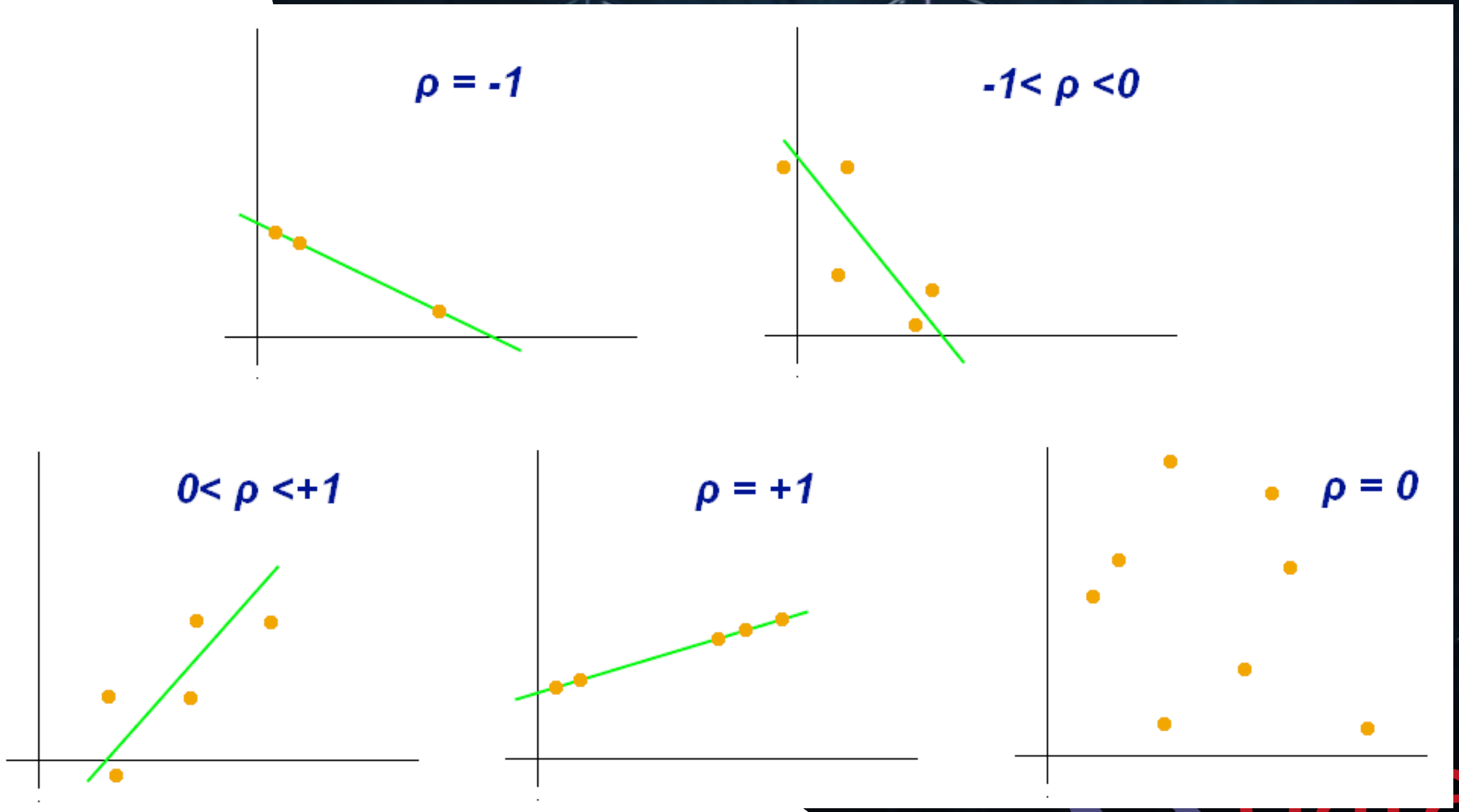


cyber & data

Pearson’s coefficient

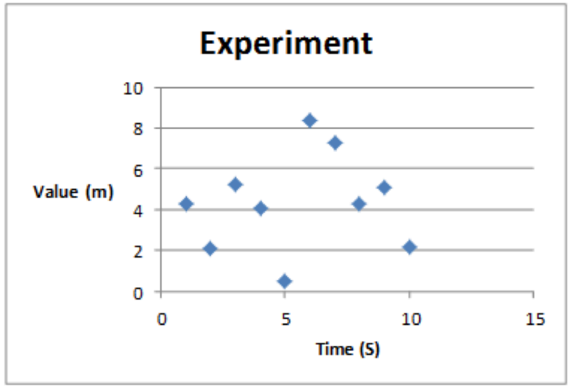
Pearson's coefficient measures the linear dependence between two variables.

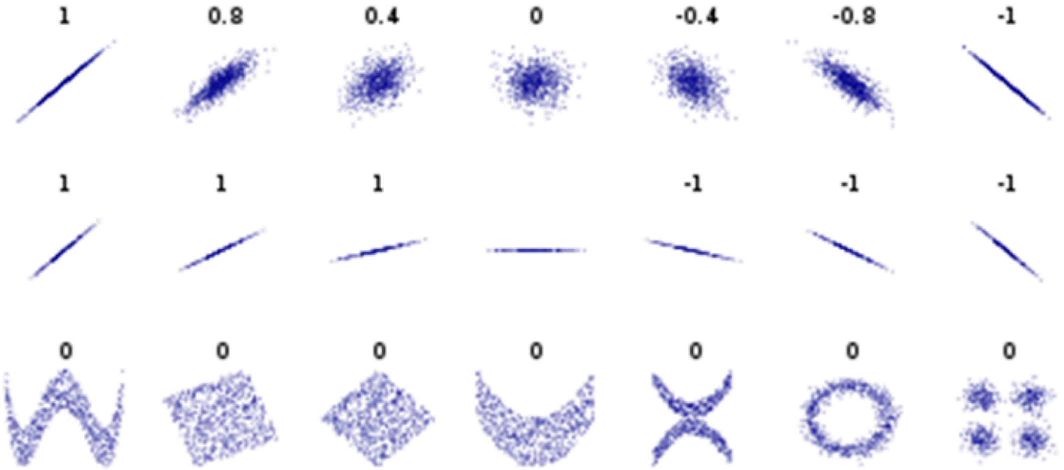
1 is total positive linear correlation,
0 is no linear correlation
−1 is total negative linear correlation.



Pearson’s coefficient

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





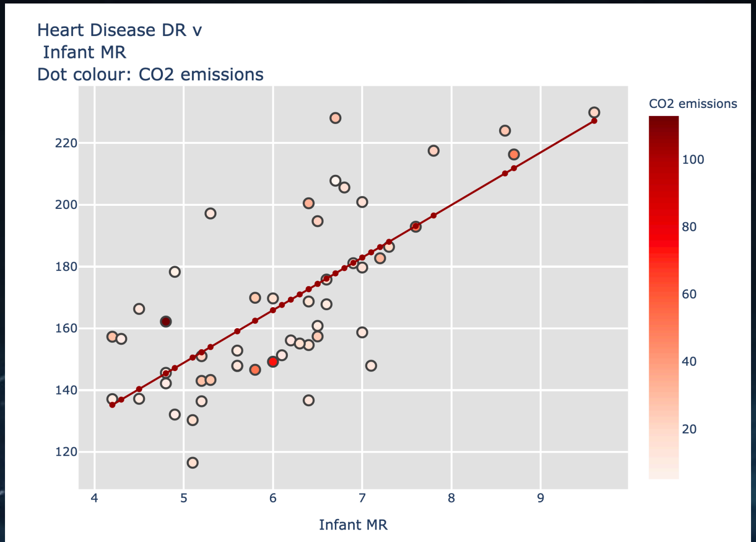
1	4.3
2	2.1
3	5.2
4	4.1
5	0.5
6	8.4
7	7.3
8	4.3
9	5.1
10	2.2

t-Test: Paired Two Sample for M

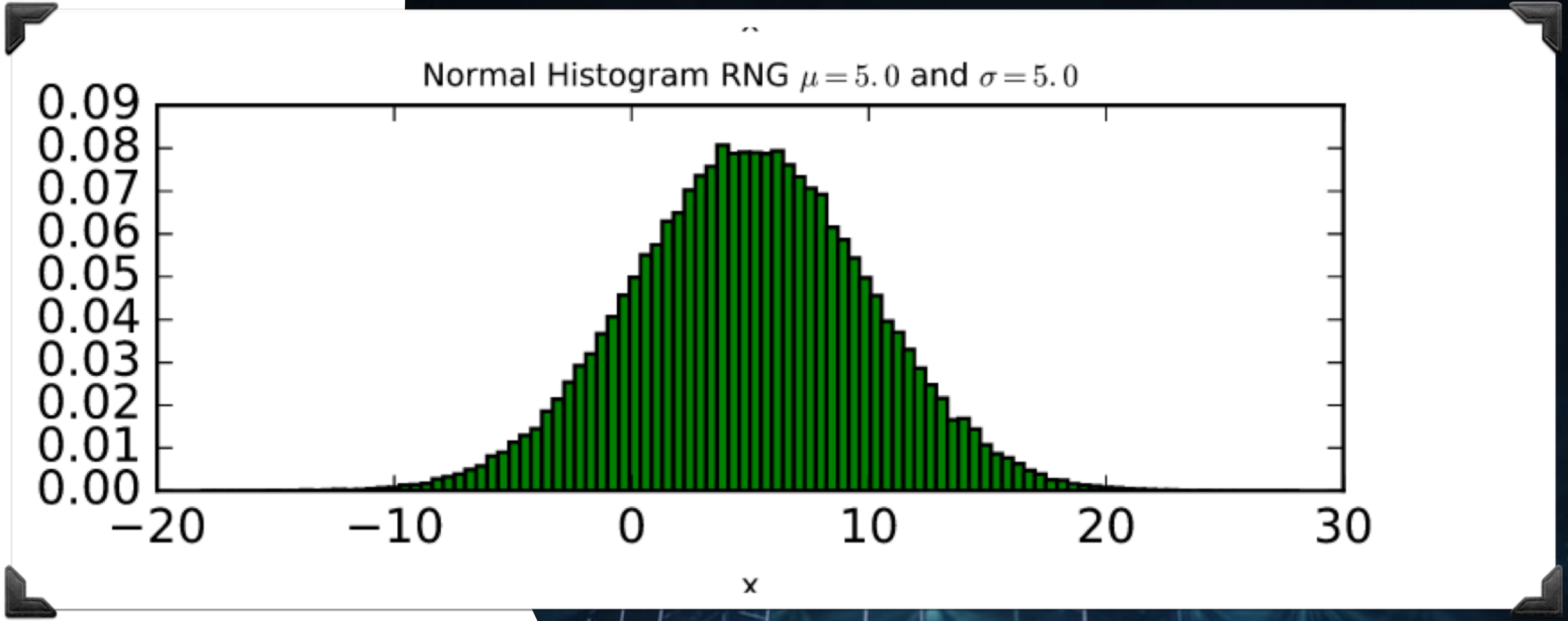
	Variable 1	*Variable 2*
Mean	5.5	4.35
Variance	9.166667	5.662778
Observati	10	10
Pearson C	0.116435	
Hypothesi	0	
df	9	
t Stat	1.002784	
P(T<=t) on	0.171081	
t Critical o	1.833113	
P(T<=t) tw	0.342162	
t Critical t	2.262157	

Correlation (Linear Regression)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              Infant MR   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.982
Method:                 Least Squares   F-statistic:                     2755.
Date:                Thu, 23 Jul 2020   Prob (F-statistic):           2.13e-45
Time:                        12:56:57   Log-Likelihood:                -62.894
No. Observations:                  51   AIC:                             127.8
Df Residuals:                      50   BIC:                             129.7
Df Model:                           1
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Heart Disease DR  0.0363      0.001     52.492      0.000       0.035       0.038
==============================================================================
Omnibus:                        2.129   Durbin-Watson:                   2.241
Prob(Omnibus):                  0.345   Jarque-Bera (JB):                2.040
Skew:                          -0.458   Prob(JB):                        0.361
Kurtosis:                       2.652   Cond. No.                         1.00
==============================================================================
```



Correlation (Linear Regression)




```
import pandas as pd
ver=pd.read_csv("df.csv")
ver.describe()
--------------------
```

	Infant MR	Heart Disease DR	Stroke DR	Suicide DR	Drug Poisoning DR
count	52.000000	52.000000	52.000000	52.000000	52.000000
mean	6.107692	167.740385	36.788462	14.851923	16.036538
std	1.170863	27.728214	5.678361	3.877006	5.602311
min	4.200000	116.500000	25.600000	7.800000	6.300000
25%	5.200000	147.875000	33.550000	12.600000	12.275000
50%	6.150000	159.750000	36.600000	14.200000	14.900000
75%	6.725000	183.625000	41.150000	17.850000	18.550000
max	9.600000	229.900000	48.800000	23.900000	35.500000

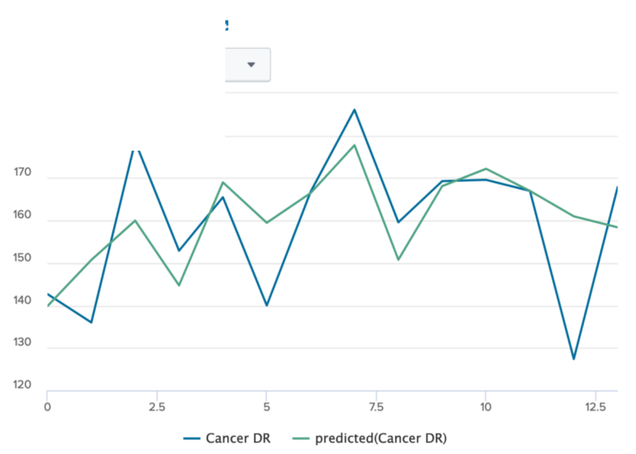
R^2 statistic

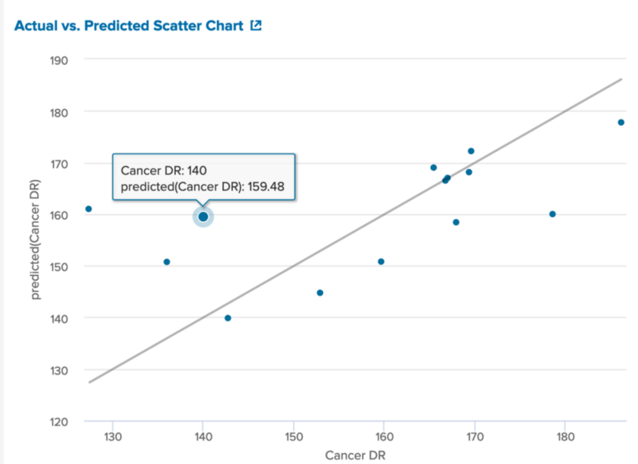
R^2 Statistic

0.3631

Root Mean Squared Error (RMSE)

13.10



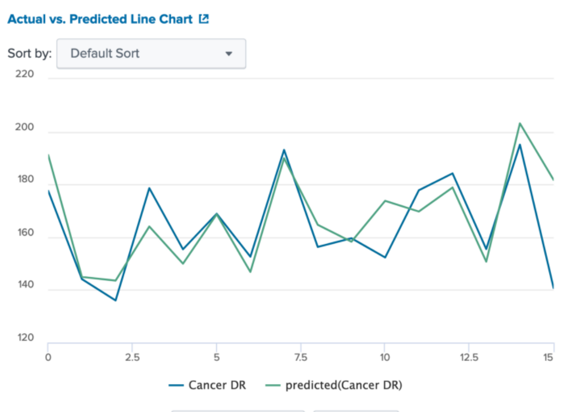


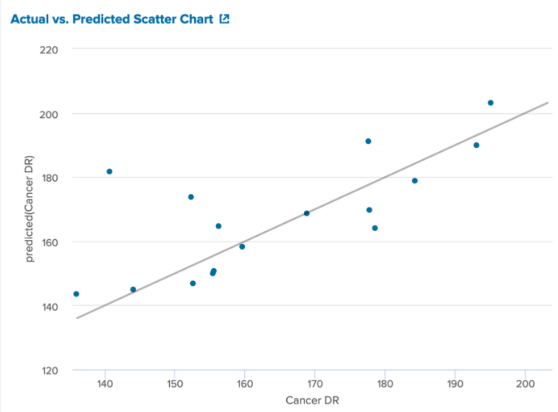
R^2 Statistic

0.7280

Root Mean Squared Error (RMSE)

7.70





“From bits to information”

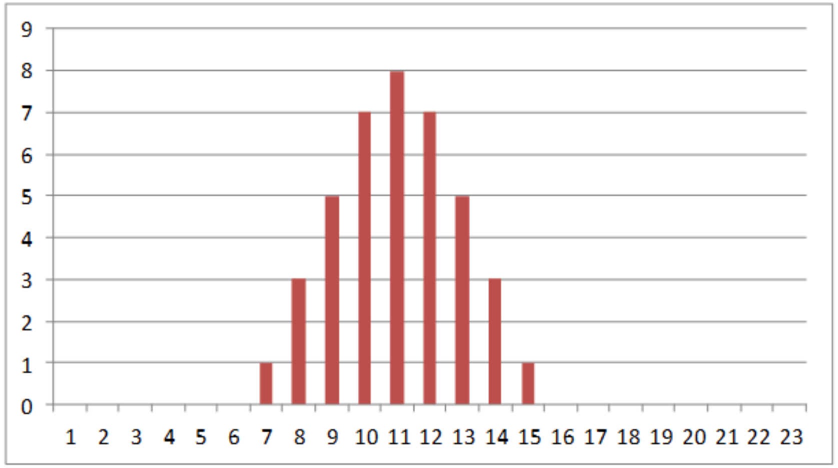
Distributions

Normal distribution

Samples
6
7
7
7
8
8
8
8
8
9
9
9
9
9
9
9
10
10
10
10
10
10
10
10
10
10
11
11
11
11
11
11
11
12
12
12
12
12
13
13
13
14

Experimental Setup

Value	Frequency
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	3
8	5
9	7
10	8
11	7
12	5
13	3
14	1
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0

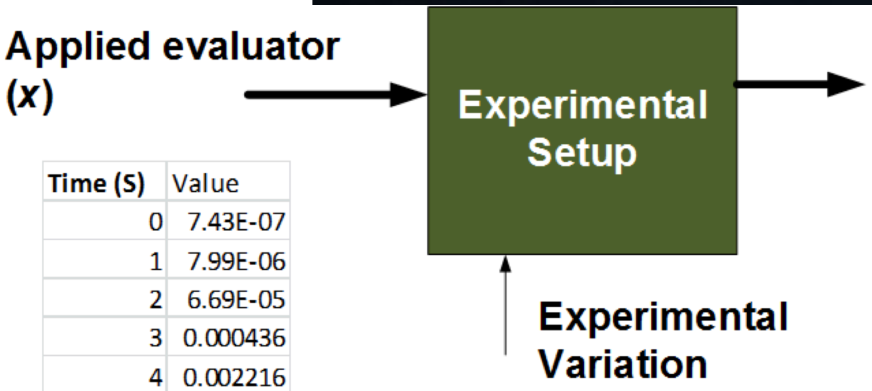


Average	10
Std Dev	1.847873

Normal distribution

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

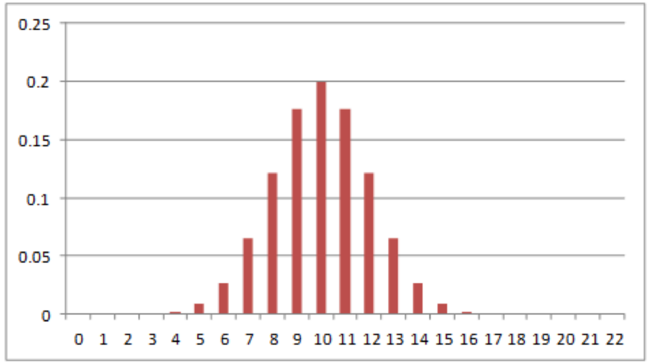
Example



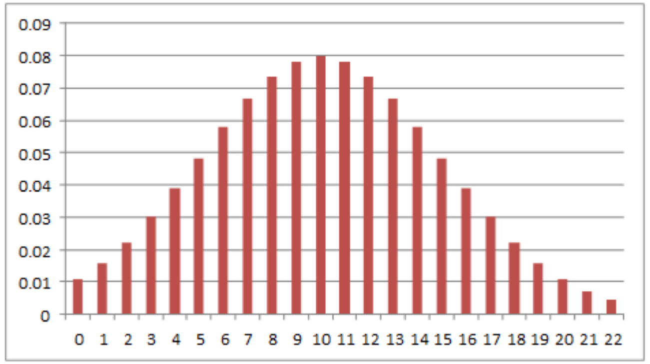
Time (S)	Value
0	7.43E-07
1	7.99E-06
2	6.69E-05
3	0.000436
4	0.002216
5	0.008764
6	0.026995
7	0.064759
8	0.120985
9	0.176033
10	0.199471
11	0.176033
12	0.120985
13	0.064759
14	0.026995
15	0.008764
16	0.002216
17	0.000436
18	6.69E-05
19	7.99E-06
20	7.43E-07
21	5.38E-08
22	3.04E-09

Normal distribution:

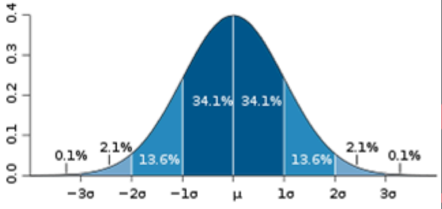
- **68% of the observations fall within of the one SDs of the mean.**
- **95% in two SDs.**
- **97.3% in three SDs.**



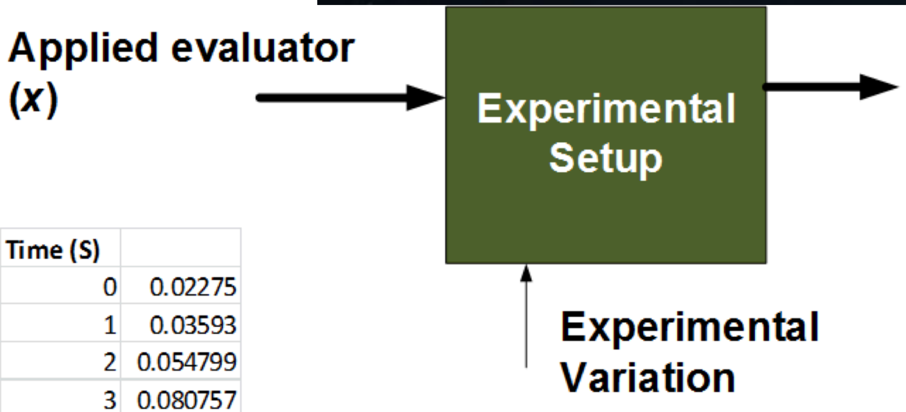
Std Dev (σ)=2
Mean (μ) = 10



Std Dev (σ)=5
Mean (μ) = 10



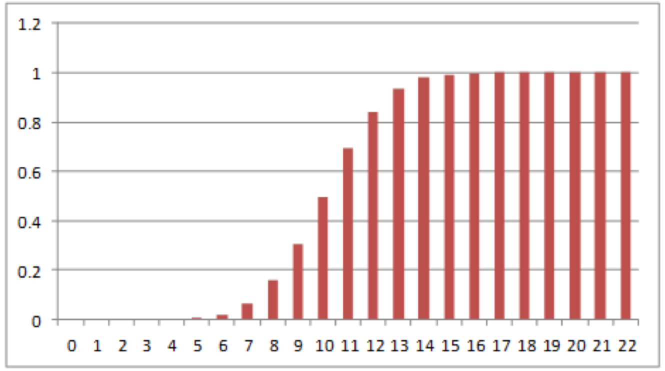
Cumulative



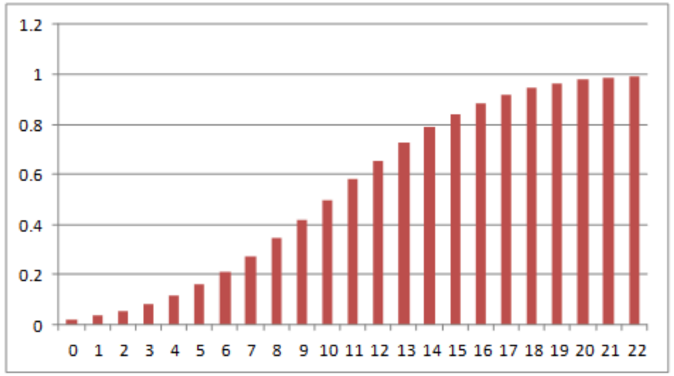
Normal distribution:

- 68% of the observations fall within of the one SDs of the mean.
- 95% in two SDs.
- 97.3% in three SDs.

Time (S)	
0	0.02275
1	0.03593
2	0.054799
3	0.080757
4	0.11507
5	0.158655
6	0.211855
7	0.274253
8	0.344578
9	0.42074
10	0.5
11	0.57926
12	0.655422
13	0.725747
14	0.788145
15	0.841345
16	0.88493
17	0.919243
18	0.945201
19	0.96407
20	0.97725
21	0.986097
22	0.991802



Std Dev (σ)=2
Mean (μ) = 10



Std Dev (σ)=5
Mean (μ) = 10

er
ta

Cumulative

Applied evaluator
(*x*)

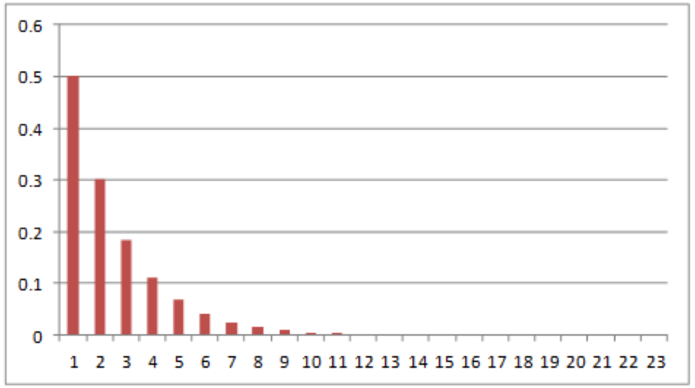
Experimental
Setup

Experimental
Variation

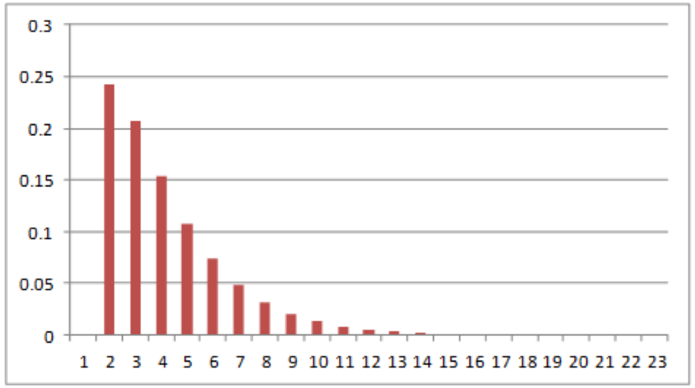
Chi-squared distribution
(also chi-square or χ^2-distribution)
with k degrees of freedom

Sum of the squares of k

0	0.5
1	0.303265
2	0.18394
3	0.111565
4	0.067668
5	0.041042
6	0.024894
7	0.015099
8	0.009158
9	0.005554
10	0.003369
11	0.002043
12	0.001239
13	0.000752
14	0.000456
15	0.000277
16	0.000168
17	0.000102
18	6.17E-05
19	3.74E-05
20	2.27E-05
21	1.38E-05
22	8.35E-06

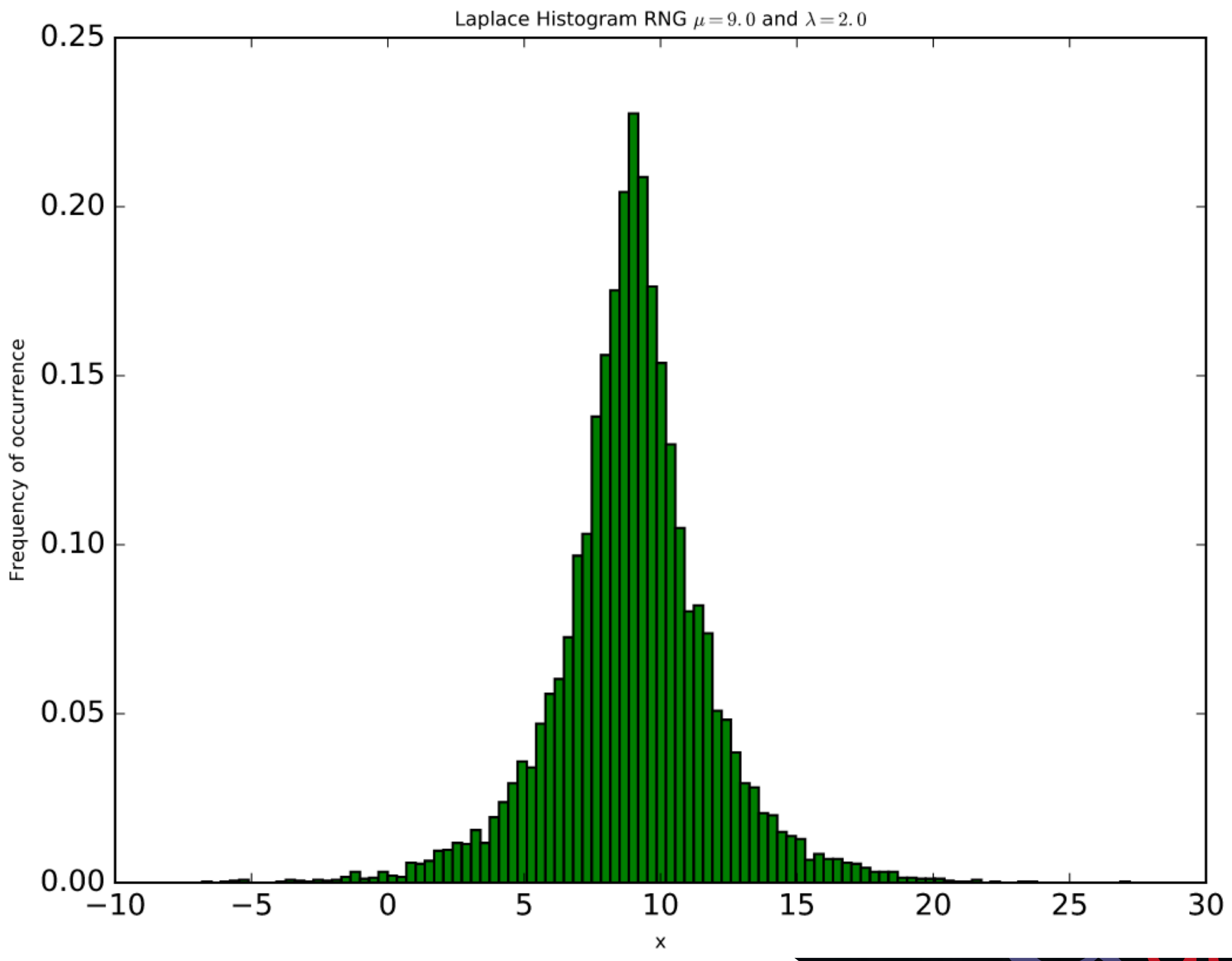


Degrees of freedom = 2



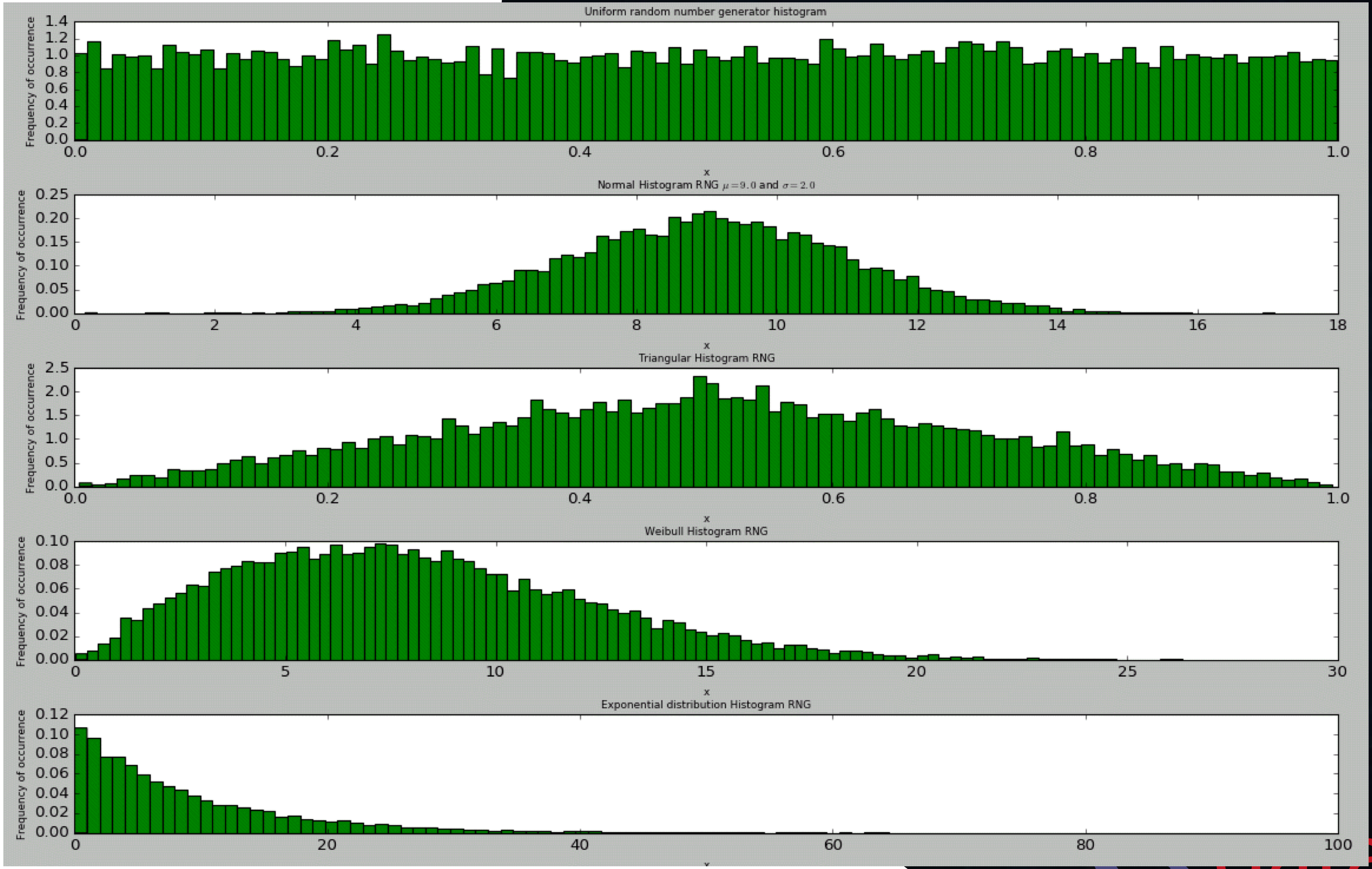
Degrees of freedom = 3

Laplace



Example

Others



Example

cyber & data

"From bits to information"

Introduction to Data Science