Statistics:

The branch of mathematics that deals with collecting, organizing, analyzing, and interpreting data for decision-making.

Types

- Descriptive Statistics
- Inferences
- Probability Distribution

Descriptive Statistics

A branch of statistics that summarizes and describes the main features of a dataset using measures like mean, median, mode, variance, and visualizations (charts, graphs, tables) without making predictions or inferences.

Import the Required Libraries

In [2]:
```
from sklearn.datasets import load_breast_cancer
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the Dataset

In [3]:
```
df=pd.read_csv(r"C:\Users\User-PC\Downloads\Statistics-20250831T034916Z-1-001\Stati
df.head()
```

Out[3]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4
3	GP	F	15	U	GT3	T	4	2	health	services	...	3
4	GP	F	16	U	GT3	T	3	3	other	other	...	4

5 rows × 33 columns

1) Measure of Central Tendacy

It summarize large datasets into a single representative value, helping AI models understand data distribution.

Techniques (Mean, Median, Mode)

Mean

Average Value Sensitive to Outliers

In [4]:
```
mn=np.mean(df['age'])
print(mn)
```

```
16.696202531645568
```

Median

Middle Value in an ordered Numemric Sequence

In [5]:
```
md=np.median(df['age'])
print(md)
```

```
17.0
```

Mode

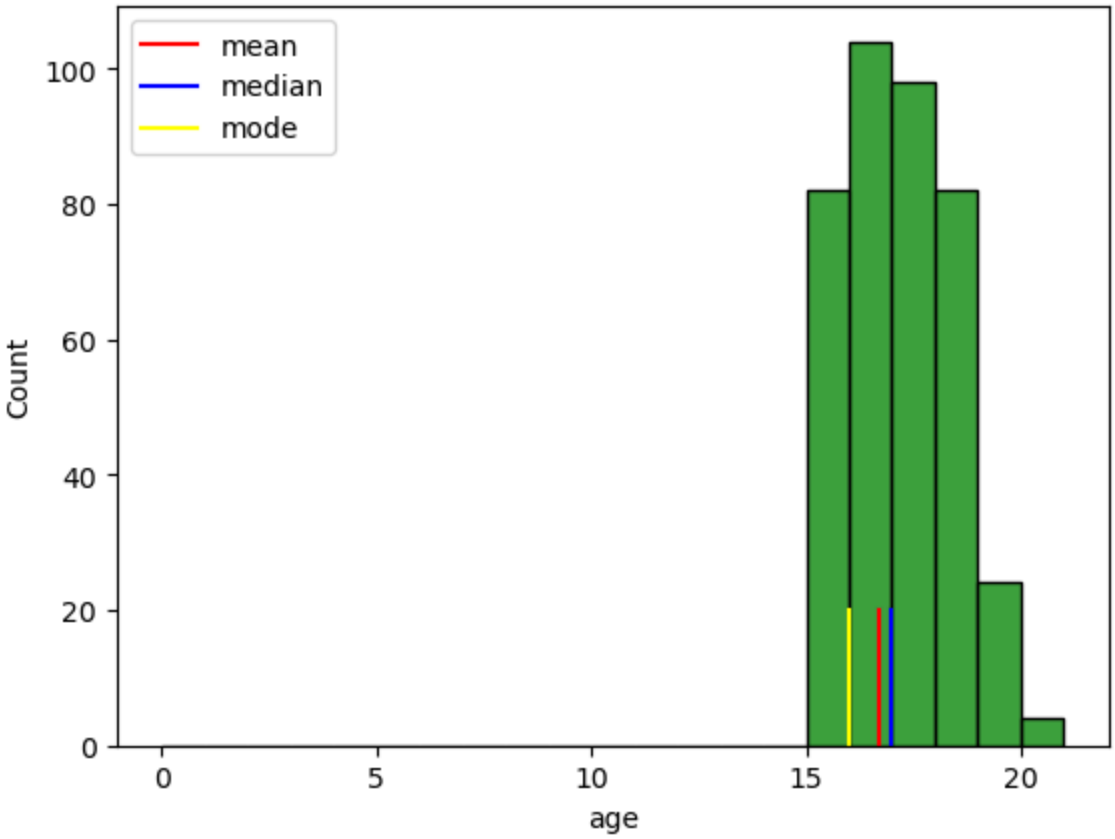
The most frequent value in the dataset.

In [6]:
```
mo=df['age'].mode()[0]
print(mo)
```

```
16
```

Visualization

In [7]:
```
sns.histplot(x='age',data=df,bins=[i for i in range(0,22,1)],color='green')
plt.plot([mn for i in range(0,21)],[i for i in range(0,21)], c='red',label='mean')
plt.plot([md for i in range(0,21)],[i for i in range(0,21)], c='blue',label='median
plt.plot([mo for i in range(0,21)],[i for i in range(0,21)], c='yellow',label="mode
plt.legend()
plt.show()
```



2) Measure of Variability

A measure of variability (or dispersion) describes how spread out or scattered the data values are around the center (mean/median). It shows the degree to which data points differ from each other.

- ## Common Measures of Variability

Range

Difference between maximum and minimum values.

In [8]:
```
min_r=df['G1'].min()
max_r=df['G1'].max()

range=max_r - min_r
range
```

Out[8]: 16

In [9]:
```
class_1=np.array([75,65,73,68,72,76])
class_2=np.array([90,47,43,96,93,51])
no = [1,2,3,4,5,6]
```

Standard Deviation

It is a measure of amount of variation or dispersion of set of values. A low standard deviation indicates the value tends to be close to mean and higher S.D tells that values are spread out over a wide range.

Example:

In [10]:
```
np.std(class_1),np.std(class_2)
```

Out[10]: (np.float64(3.8622100754188224), np.float64(23.18045153428495))

Variance

Average of squared differences from the mean.

In [11]:
```
np.var(class_1), np.var(class_2)
```

Out[11]: (np.float64(14.916666666666666), np.float64(537.3333333333334))

Mean Absolute Deviation

$$\text{MAD} = \frac{\sum |x_i - \bar{x}|}{n}$$

The mean Absolute deviation of a dataset is the average distance between each data point and the mean. It gives us idea about the dispersion in a dataset.

Example:

In [12]:
```
mean=np.mean(class_1)
mean
```

Out[12]: np.float64(71.5)

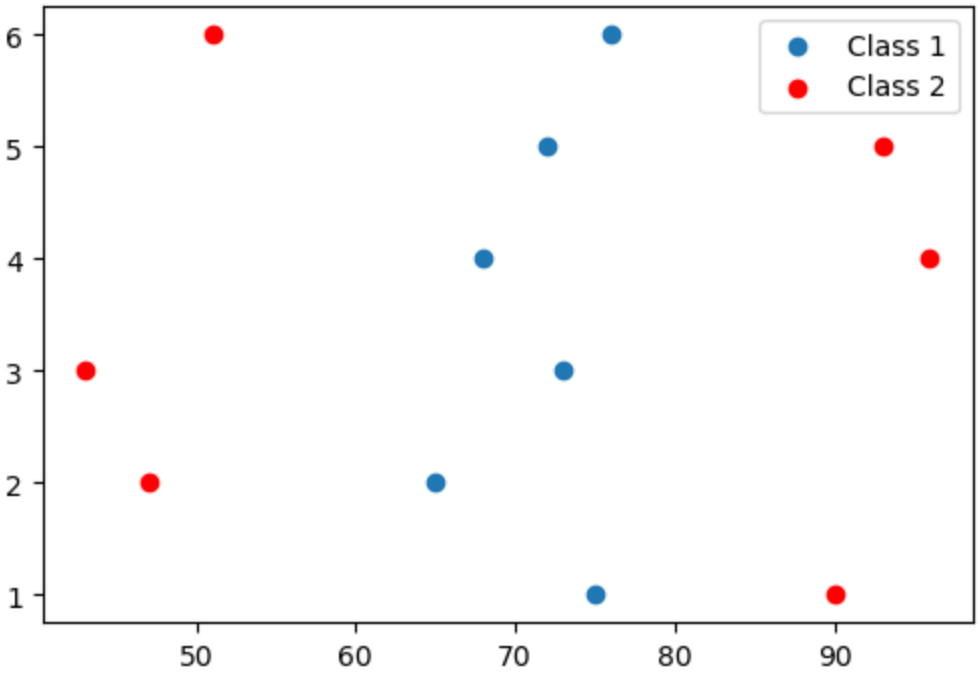
In [13]:
```
class_1_mad=np.sum(np.abs(class_1-mean)/len(class_1))
class_2_mad=np.sum(np.abs(class_2-mean)/len(class_2))
```

In [14]:
```
class_1_mad,class_2_mad
```

Out[14]:
```
(np.float64(3.3333333333333335), np.float64(23.0))
```

Visualization

In [15]:
```
plt.figure(figsize=(6,4))
plt.scatter(class_1,no,label='Class 1')
plt.scatter(class_2,no, color='red',label='Class 2')
plt.legend()
plt.show()
```



Percentiles & Quartiles

Percentiles:

Percentiles divide ordered data into 100 equal parts. The k-th percentile is the value below which k% of the data falls.

Example: If a student's test score is at the 90th percentile, they scored better than 90% of students.

In [16]:
```
q1 = np.percentile(df['age'], 25)
q2 = np.percentile(df['age'], 50)
q3 = np.percentile(df['age'], 75)
```

```
print("Q1:", q1, "Q2:", q2, "Q3:", q3, "IQR:", q3-q1)
```

```
Q1: 16.0 Q2: 17.0 Q3: 18.0 IQR: 2.0
```

Quartiles

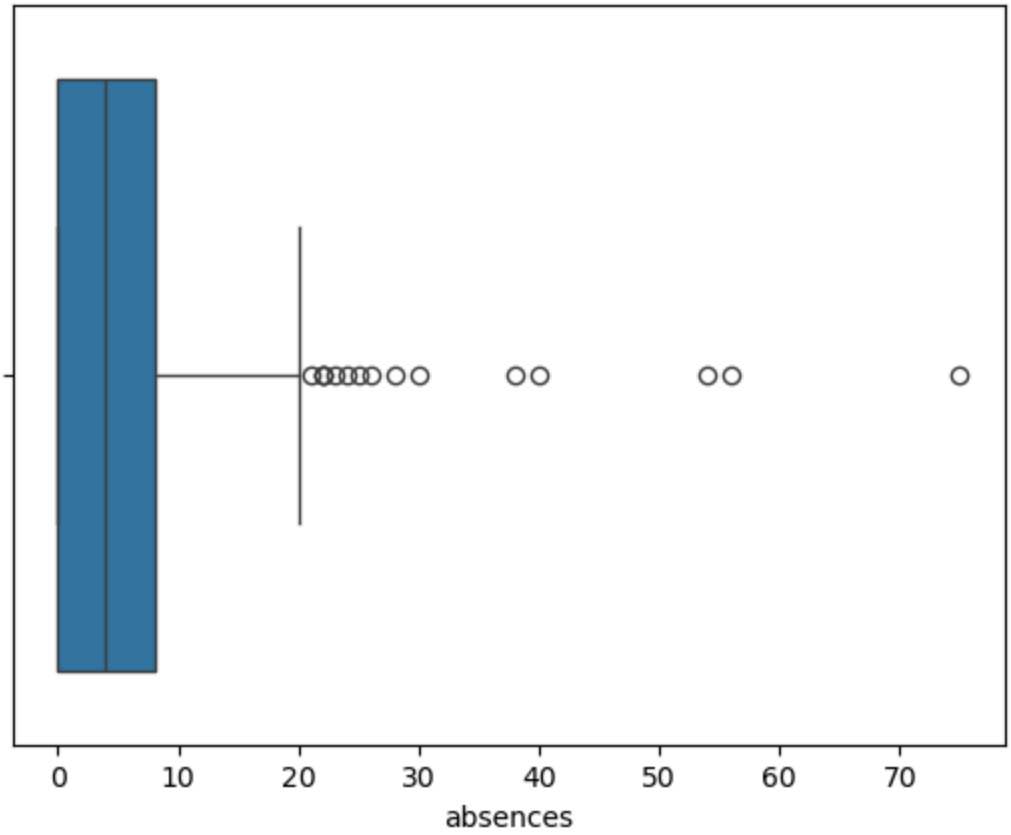
- Quartiles divide ordered data into 4 equal parts (25% each).
- Q1 (25th percentile): 25% of data is below this value.
- Q2 (50th percentile / Median): Middle of the data.
- Q3 (75th percentile): 75% of data is below this value.
- IQR (Interquartile Range): Q3 − Q1 → measures spread of the middle 50% (helps detect outliers).

In [17]:
```
df.describe()
```

Out[17]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000

In [18]:
```
sns.boxplot(x='absences',data=df)
plt.show()
```



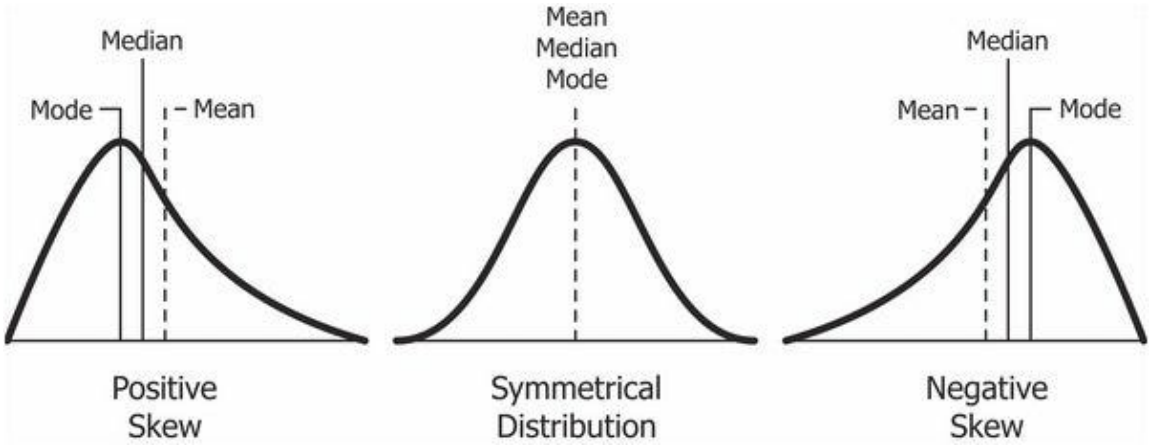
Measure of Shapes

1.) Skewness

$$\text{Skewness} = \frac{\sum(x_i - \bar{x})^3}{(N-1)\cdot\sigma^3}$$

It tells where the distribution is symmetrical or tilted. There are two types of Skewness.

- Positive Skewness- Tail on the right (Mode < Median < Mode)
- Negative Skewness- Tail on the left (Mean < Median < Mode)

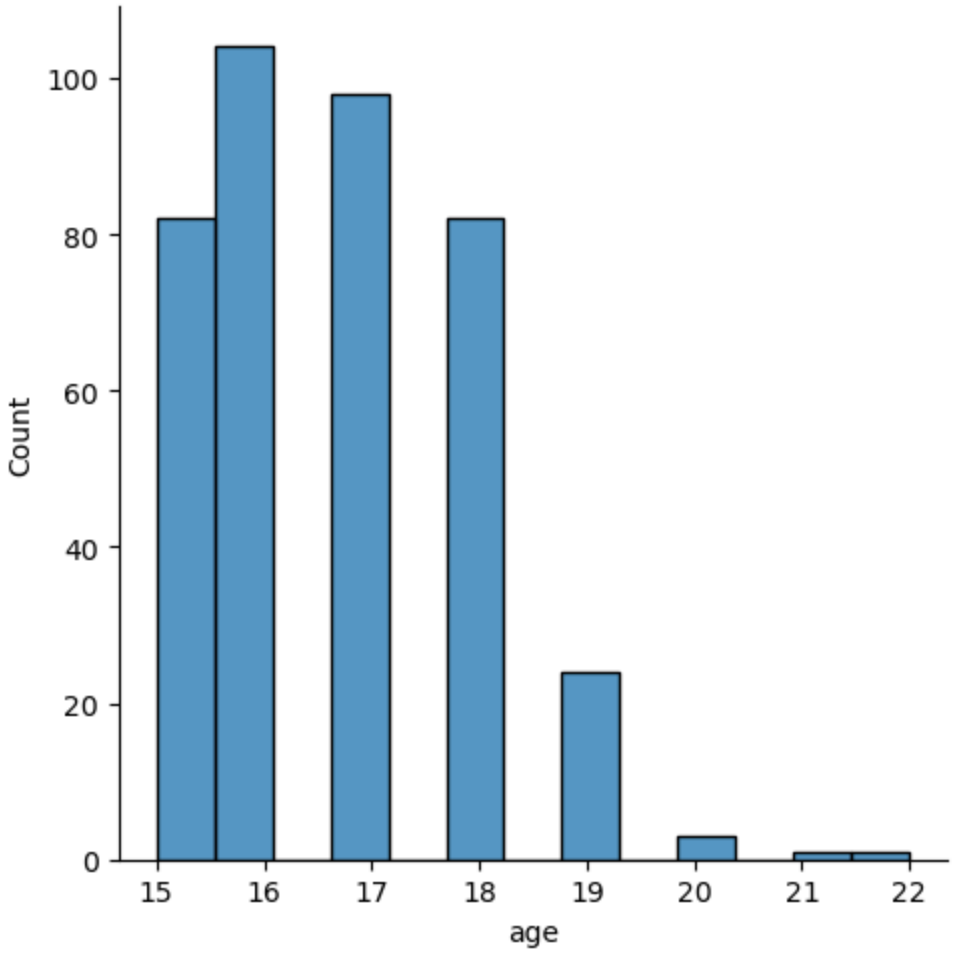


Positive Skew

In [19]:
```
df['age'].skew()
```

Out[19]:
```
np.float64(0.46627016141836425)
```

In [20]:
```
sns.displot(df['age'])
plt.show()
```

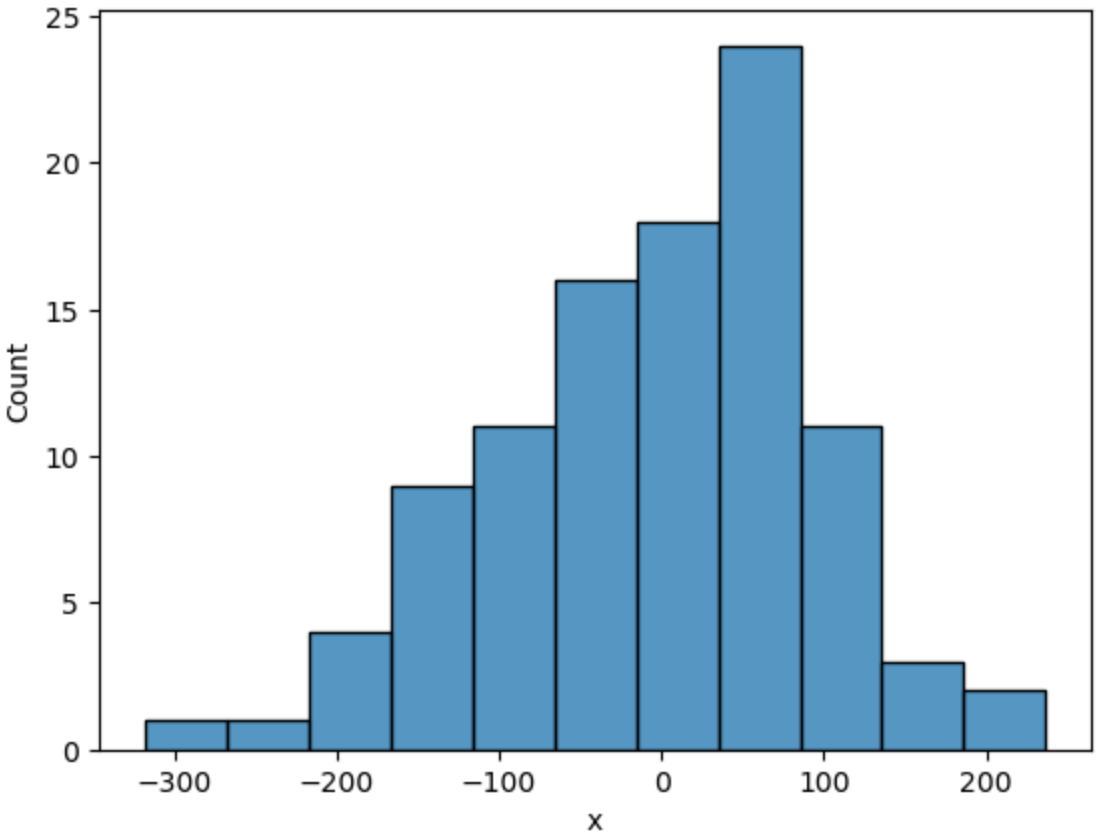


Negative Skew

In [21]:
```
data= np.random.normal(0,100,100)
df_2=pd.DataFrame({"x":data})
df_2['x'].skew()
```

Out[21]: np.float64(-0.41915340636494075)

In [22]:
```
sns.histplot(df_2['x'])
plt.show()
```



In [23]:
```
df_2['x'].mean(),df_2['x'].median()
```

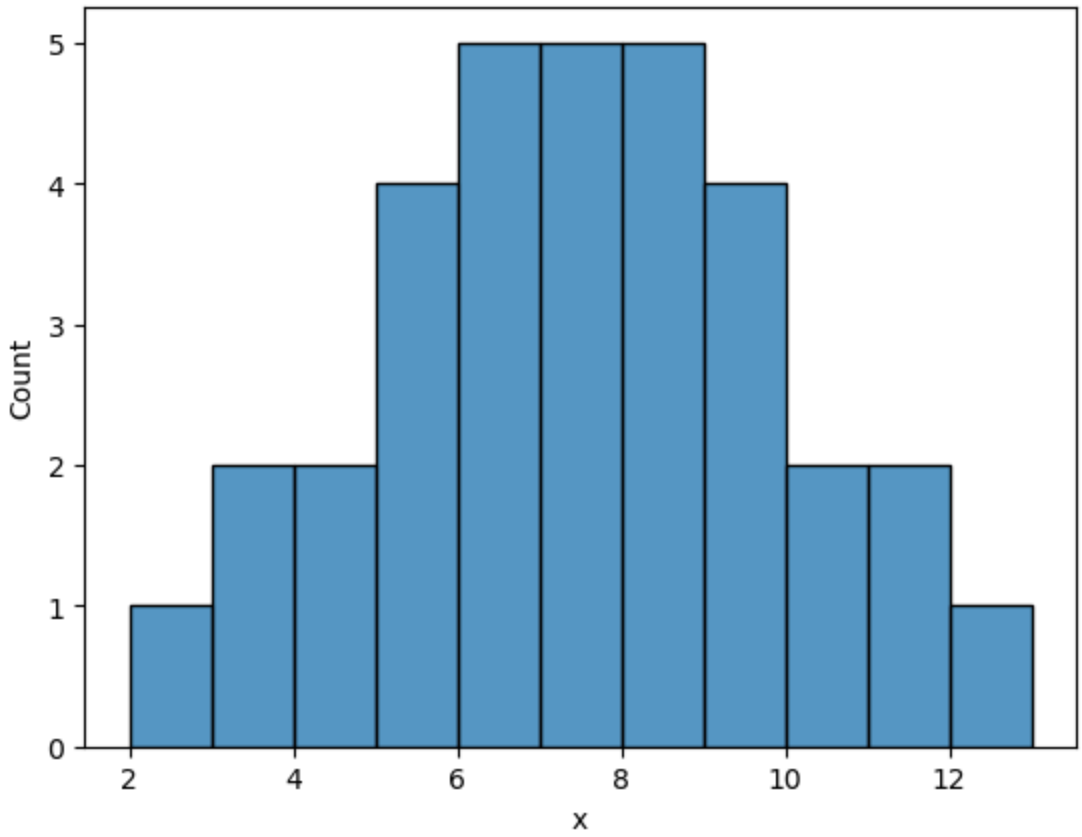
Out[23]: (np.float64(-5.4244234194927525), -3.100787838801281)

Symmetric Distribution

In [24]:
```
data_2=[2,3,3,4,4,5,5,5,5,6,6,6,6,6,7,7,7,7,7,8,8,8,8,8,9,9,9,9,10,10,11,11,12]
df_3=pd.DataFrame({"x":data_2})
df_3['x'].skew()
```

Out[24]: np.float64(0.0)

In [25]:
```
sns.histplot(x='x',data=df_3,bins=[2,3,4,5,6,7,8,9,10,11,12,13])
plt.show()
```



In [26]:
```
print(df_3['x'].mean(), df_3['x'].median(), df_3['x'].mode())
```

```
7.0 7.0 0    6
1    7
2    8
Name: x, dtype: int64
```

Probability

Probability measures the likelihood of a particular outcome or event occuring. It is typically expressed as a number between 0 and 1, where 0 indicates impossiblity (event will not occur) and 1 indicates event certainity (event will occur).

- P(A) = Number of times A occur /Total number of possible outcomes

Random Variable

A random variable is a variable whose possible values are outcomes of a random experiment. It assigns a numeric value to each outcome.

Types of Random Variable

1.) Discrete Random Variable:

Takes countable values (finite or infinite).

Example: Number of students in a class, number of heads in 10 coin tosses.

2.) Continuous Random Variable:

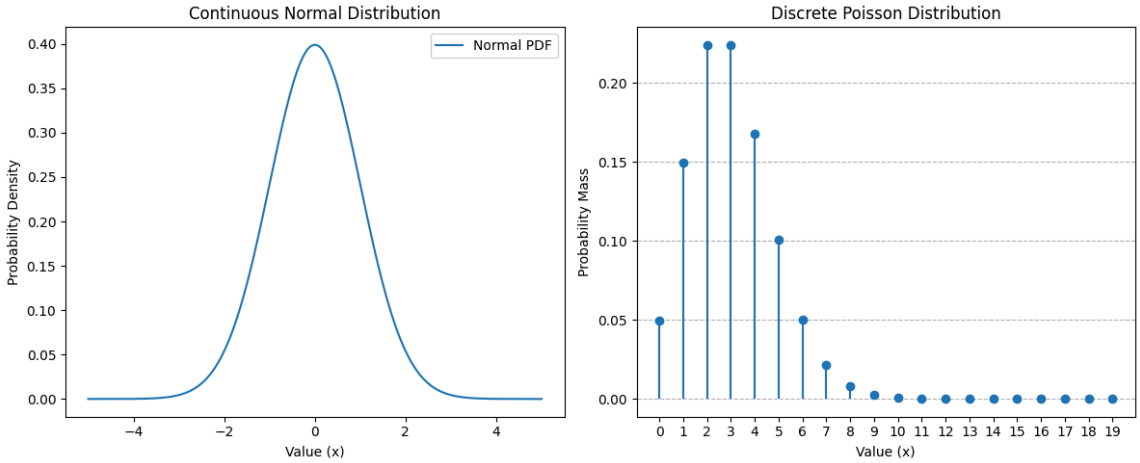
Takes any value within a range (infinite, uncountable).

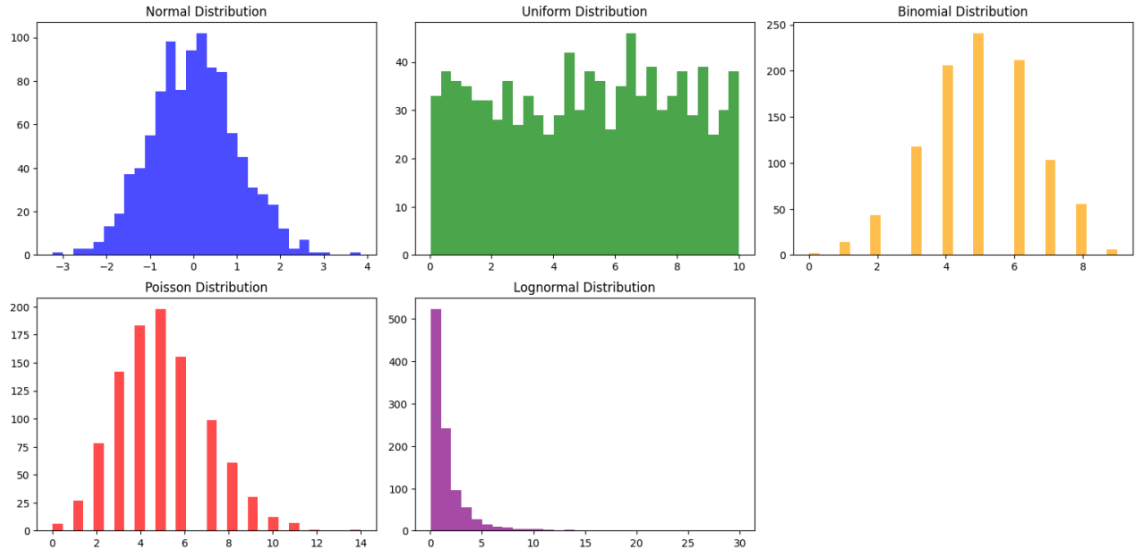
Example: Height of students, time taken to complete a task, temperature.

Probability Distribution

Probability distribution describes how the probabilities of different outcomes are distributed over the sample space of random variable.

- Discrete Probability Distribution
- Continuous Probability Distribution



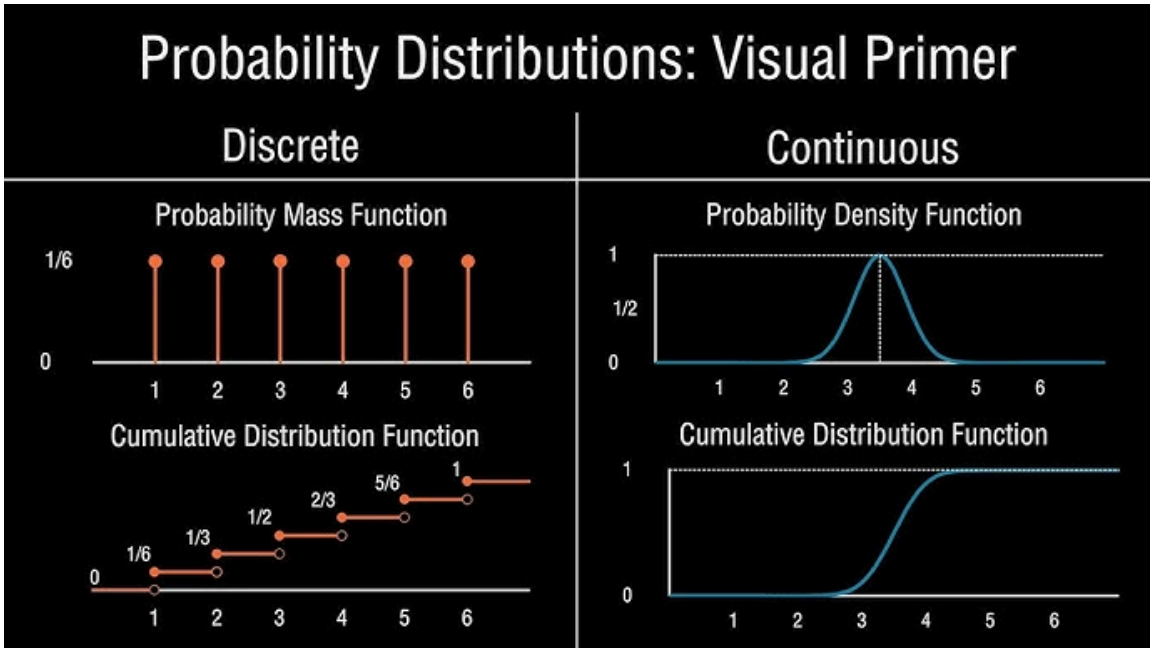


How Probability Distributions Shape Our World: A Dive into Normal, Uniform, Binomial, Poisson, and Lognormal Distributions

Probability Distribution Function

It is a mathematical function that gives the probabilities of occurence of different possible outcomes for an expirement.

- Probability Distributive Function (PDF)
- Probability Mass Function (PMF)
- Cumulative Density Function (CDF)

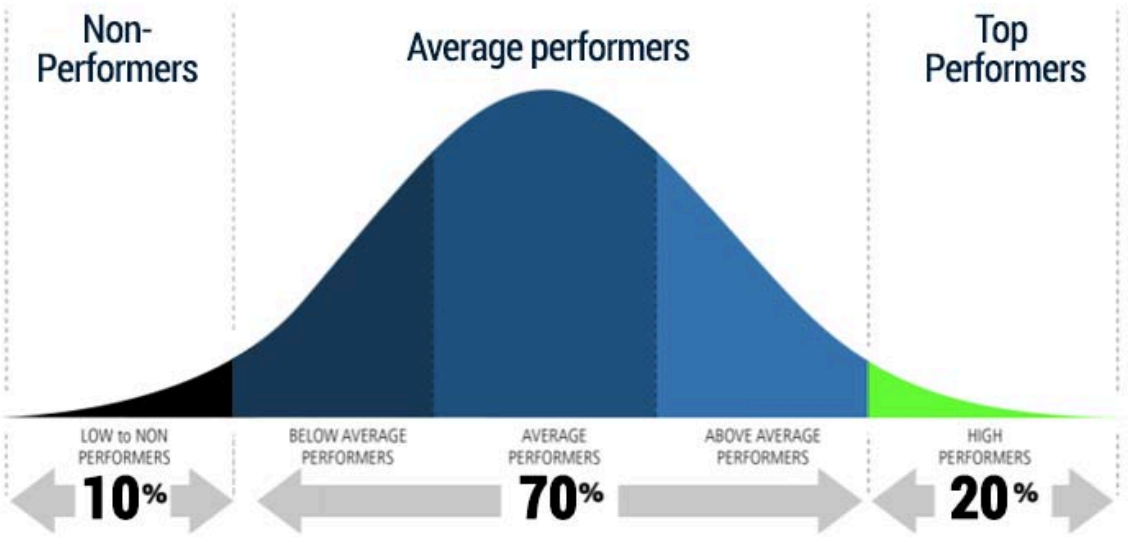


Normal Distribution

It is known as Gaussian Distribution, that is symmetic about the mean, showing that the data near the mean are more frequent in occurence than the data from the mean.

- Formula:

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



- In graph form normal distribution will appear as a bell curve

Standard Normal Distribution

- The Standard normal distribution, as known as Z-distribution or Z-score, is a special case of the normal distribution.
- mean(u) of 0 and a standard deviation of 1.

Covariance & Correlation

- Covariance signifies the direction of the linear relationship between the two variables. By direction we mean if the variable is directly propotional or inversely propotional to each other.

$$Cov(x,y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

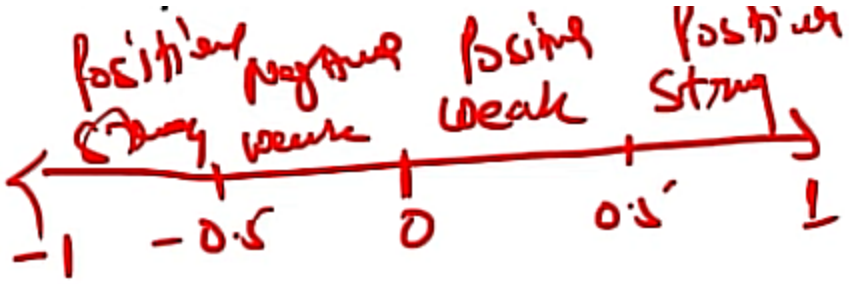
- Increasing the value of one variable might have a positive or negative impact on the value of other variable.)
- x- Positive, y- Positive -> Positive Covariance/Correlation
- x- Negative, y- Positive -> Negative Covariance/Correlation
- x- Positive, y- Negative -> 0 Covariance/Correlation

Correlation

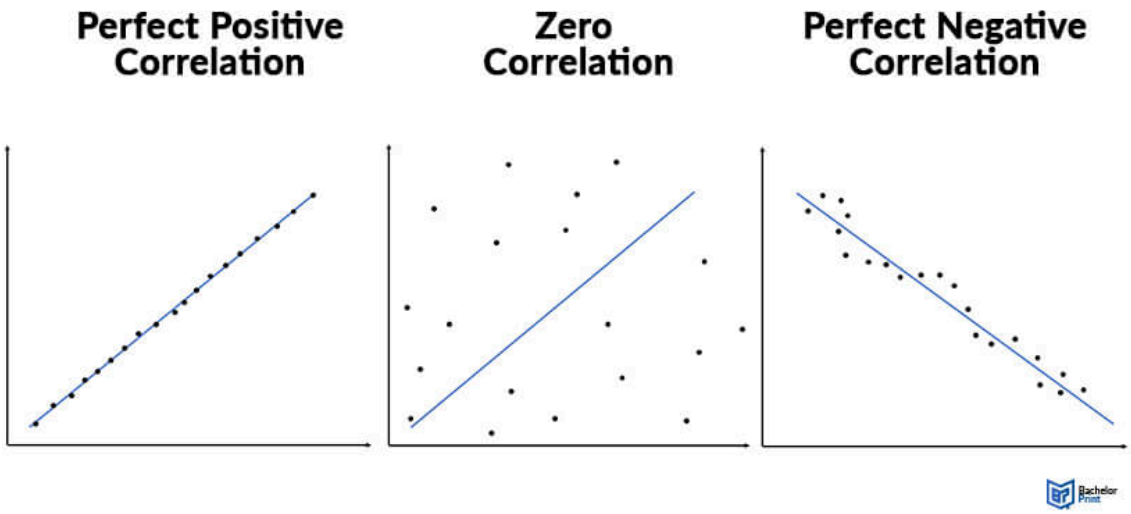
- Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables.

$$Correlation = \frac{Cov\ (x, y)}{\sigma x * \sigma y}$$

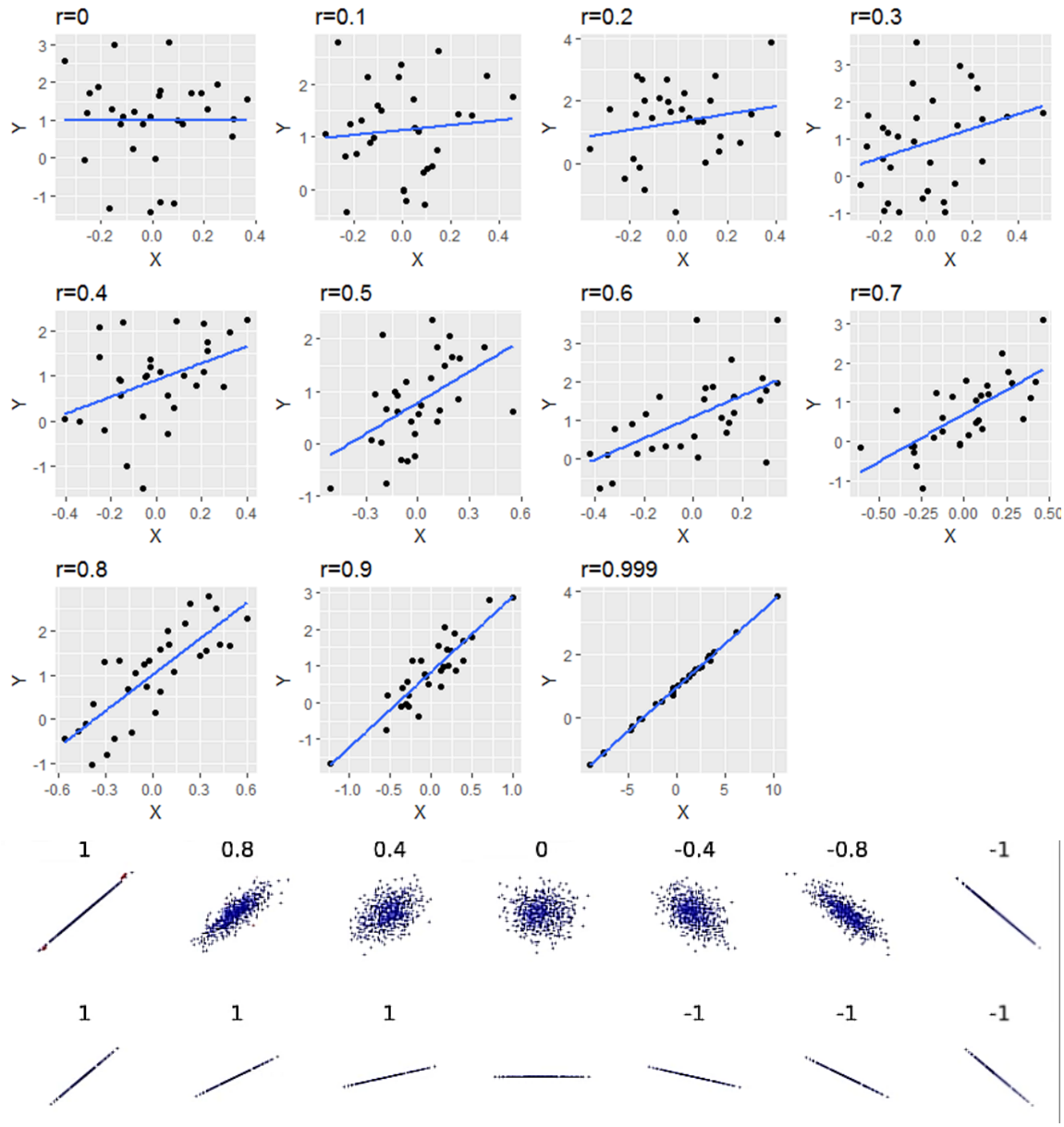
- where Cov is the covariance
- varianc x is the standard Deviation of x
- variance y is the standard deviation of y



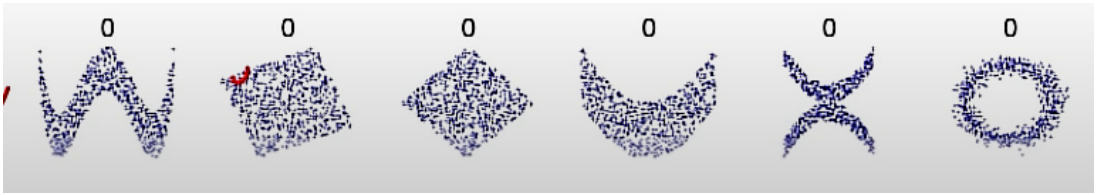
Correlation Graph:



Pearson Correlation Graph



- We notice that graph is getting scattered towards 0 and -0 values



Example

In [27]:
```
df.head(3)
```

Out[27]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4

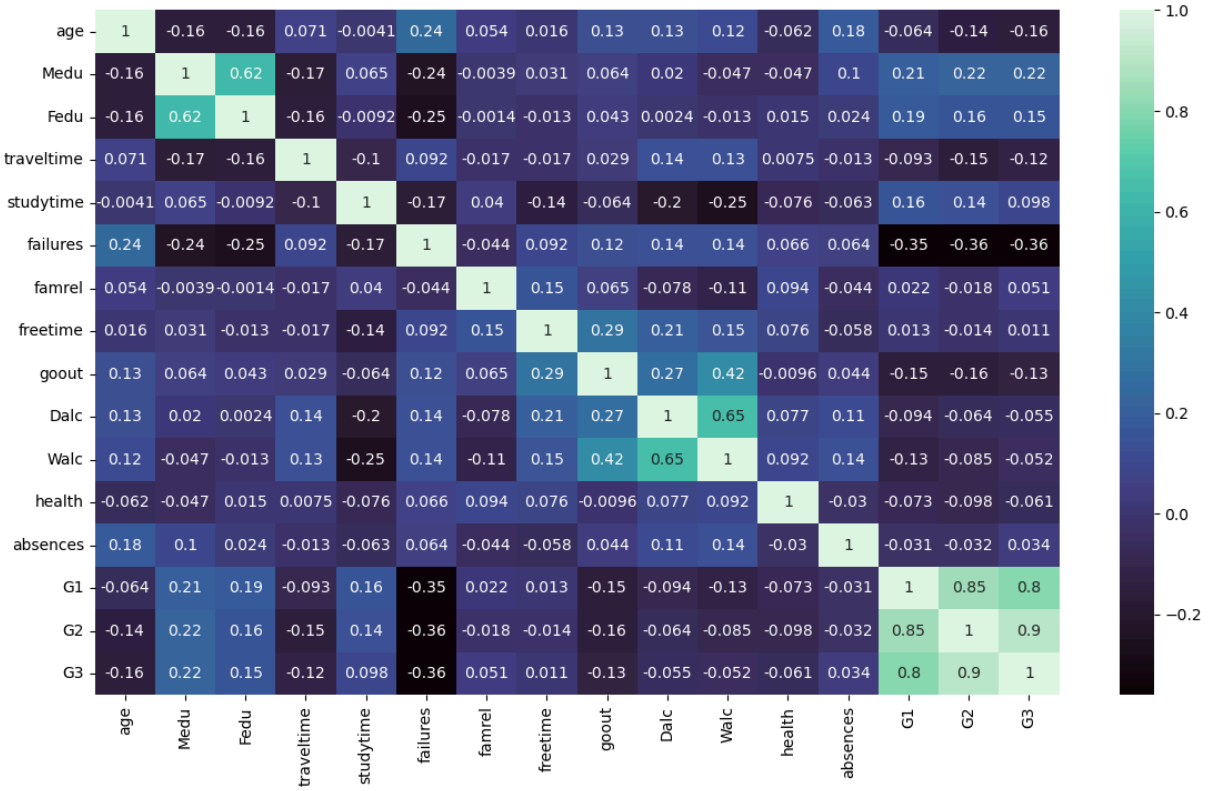
3 rows × 33 columns

In [28]:
```
data_corr=df.select_dtypes(include= ['int']).corr()
data_corr
```

Out[28]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	fr
age	1.000000	-0.163658	-0.163438	0.070641	-0.004140	0.243665	0.053940	0.
Medu	-0.163658	1.000000	0.623455	-0.171639	0.064944	-0.236680	-0.003914	0.
Fedu	-0.163438	0.623455	1.000000	-0.158194	-0.009175	-0.250408	-0.001370	-0.
traveltime	0.070641	-0.171639	-0.158194	1.000000	-0.100909	0.092239	-0.016808	-0.
studytime	-0.004140	0.064944	-0.009175	-0.100909	1.000000	-0.173563	0.039731	-0.
failures	0.243665	-0.236680	-0.250408	0.092239	-0.173563	1.000000	-0.044337	0.
famrel	0.053940	-0.003914	-0.001370	-0.016808	0.039731	-0.044337	1.000000	0.
freetime	0.016434	0.030891	-0.012846	-0.017025	-0.143198	0.091987	0.150701	1.
goout	0.126964	0.064094	0.043105	0.028540	-0.063904	0.124561	0.064568	0.
Dalc	0.131125	0.019834	0.002386	0.138325	-0.196019	0.136047	-0.077594	0.
Walc	0.117276	-0.047123	-0.012631	0.134116	-0.253785	0.141962	-0.113397	0.
health	-0.062187	-0.046878	0.014742	0.007501	-0.075616	0.065827	0.094056	0.
absences	0.175230	0.100285	0.024473	-0.012944	-0.062700	0.063726	-0.044354	-0.
G1	-0.064081	0.205341	0.190270	-0.093040	0.160612	-0.354718	0.022168	0.
G2	-0.143474	0.215527	0.164893	-0.153198	0.135880	-0.355896	-0.018281	-0.
G3	-0.161579	0.217147	0.152457	-0.117142	0.097820	-0.360415	0.051363	0.

In [29]:
```
plt.figure(figsize=(14,8))
sns.heatmap(data_corr, annot=True, cmap='mako')
plt.show()
```



In [30]:
```
data_cov=df.select_dtypes(include= ['int']).cov()
```

In [31]:
```
plt.figure(figsize=(14,8))
sns.heatmap(data_cov, annot=True)
plt.show()
```