

1- Generate an array of size 2x3 from normal distribution with mean 0 and standard deviation of 2.

```
import numpy as np
array = np.random.normal(loc=0, scale=2, size=(2, 3))
array

array([[ 1.30090525, -0.31823242,  3.38196244],
       [-0.07196257, -1.49847399,  4.16624289]])
```

2- Generate an array of size 2x3 from standard normal distribution.

```
array = np.random.randn(2, 3)
array

array([[ -0.2217965 , -0.24614534,  0.15398277],
       [-0.68735986, -0.19089692, -0.0123638 ]])
```

3- Create [1, 2, 3, 4, 5, 6, 7] as a numpy array and do the following:

- Slice elements from index 1 to index 5
- Slice elements from index 4 to the end of the array
- Slice elements from the beginning to index 4 (not included)
- Slice from the index 3 from the end to index 1 from the end
- Return every other element from index 1 to index 5
- Return every other element from the entire array.

```
# Create the array
x = np.array([1, 2, 3, 4, 5, 6, 7])
x

array([1, 2, 3, 4, 5, 6, 7])

# Slice elements from index 1 to index 5
x[1:5]

array([2, 3, 4, 5])

# Slice elements from index 4 to the end of the array
x1 = x [4:]
x1

array([5, 6, 7])

# Slice elements from the beginning to index 4 (not included)
x2 = x[:4]
x2

array([1, 2, 3, 4])
```

```
# Slice from the index 3 from the end to index 1 from the end
```

```
x3 = x[-3:-1]
```

```
x3
```

```
array([5, 6])
```

```
# Return every other element from index 1 to index 5
```

```
x4 = x[1:5:2]
```

```
x4
```

```
array([2, 4])
```

```
# Return every other element from the entire array
```

```
x5 = x[::2]
```

```
x5
```

```
array([1, 3, 5, 7])
```

4- Create the following Pandas Series.

```
Programming 10
```

```
for 20
```

```
DS 30
```

```
import pandas as pd
```

```
X = pd.Series({'Programming': 10, 'for': 20, 'DS': 30})
```

```
X
```

```
Programming    10
```

```
for            20
```

```
DS             30
```




```
dtype: int64
```

5- Write a Pandas program to display a summary of the basic information about a specified DataFrame and its data.

Sample DataFrame:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
exam_data = {
    'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'J
    'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
    'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
    'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']
}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
df
```

	name	score	attempts	qualify	
a	Anastasia	12.5	1	yes	
b	Dima	9.0	3	no	
c	Katherine	16.5	2	yes	
d	James	NaN	3	no	
e	Emily	9.0	2	no	
f	Michael	20.0	3	yes	
g	Matthew	14.5	1	yes	
h	Laura	NaN	1	no	
i	Kevin	8.0	2	no	
j	Jonas	19.0	1	yes	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0    name         10 non-null     object
1    score         8 non-null      float64
2    attempts     10 non-null     int64
3    qualify      10 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
```

6- Write a Pandas program to join the two given dataframes along rows and assign all data.

```
student_data1: {'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'], 'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen', 'Ed Bernal', 'Kwame Morin'], 'marks': [200, 210, 190, 222, 199]}




student_data2: { 'student_id': ['S4', 'S5', 'S6', 'S7', 'S8'], 'name': ['Scarlette Fisher', 'Carla Williamson', 'Dante Morse', 'Kaiser William', 'Madeeha Preston'], 'marks': [201, 200, 198, 219, 201]}
```

```
student_data1 = {
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen', 'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]}

student_data2 = {
    'student_id': ['S4', 'S5', 'S6', 'S7', 'S8'],
    'name': ['Scarlette Fisher', 'Carla Williamson', 'Dante Morse', 'Kaiser William', 'Madeeha Preston'],
    'marks': [201, 200, 198, 219, 201]}
```

```
df1 = pd.DataFrame(student_data1)
df2 = pd.DataFrame(student_data2)

df= pd.concat([df1, df2], ignore_index=True)
df
```

	student_id	name	marks	
0	S1	Danniella Fenton	200	
1	S2	Ryder Storey	210	
2	S3	Bryce Jensen	190	
3	S4	Ed Bernal	222	
4	S5	Kwame Morin	199	
5	S4	Scarlette Fisher	201	
6	S5	Carla Williamson	200	
7	S6	Dante Morse	198	
8	S7	Kaiser William	219	
9	S8	Madeeha Preston	201	




7- Write a Pandas program to join the two previously given dataframes along columns.

```
student_data1 = {
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen', 'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]}

student_data2 = {
    'student_id': ['S4', 'S5', 'S6', 'S7', 'S8'],
    'name': ['Scarlette Fisher', 'Carla Williamson', 'Dante Morse', 'Kaiser William', 'Madeeha Preston'],
    'marks': [201, 200, 198, 219, 201]}

df1 = pd.DataFrame(student_data1)
df2 = pd.DataFrame(student_data2)

df= pd.concat([df1, df2], axis=1)
df
```

	student_id	name	marks	student_id	name	marks	
0	S1	Danniella Fenton	200	S4	Scarlette Fisher	201	
1	S2	Ryder Storey	210	S5	Carla Williamson	200	
2	S3	Bryce Jensen	190	S6	Dante Morse	198	
3	S4	Ed Bernal	222	S7	Kaiser William	219	
4	S5	Kwame Morin	199	S8	Madeeha Preston	201	

8- Create a dataframe using the following dictionary and answer the questions:

```
data: {'Brand' : ['Maruti', 'Hyundai', 'Tata', 'Mahindra', 'Maruti', 'Hyundai', 'Renault', 'Tata',
'Maruti'], 'Year' : [2012, 2014, 2011, 2015, 2012, 2016, 2014, 2018, 2019], 'Kms Driven' : [50000,
30000, 60000, 25000, 10000, 46000, 31000, 15000, 12000], 'City' : ['Gurgaon', 'Delhi', 'Mumbai',
```

```
'Delhi', 'Mumbai', 'Delhi', 'Mumbai', 'Chennai', 'Ghaziabad'], 'Mileage' : [28, 27, 25, 26, 28, 29, 24, 21, 24]]}
```




- Select cars with brand 'Maruti' and Mileage > 25
- Select a range of rows from 2 to 5
- Update values of 'Mileage' if Year < 2015
- Select 0th, 2th, 4th, and 7th index rows
- Select rows from 1 to 4 and columns from 2 to 4

```
data = {
    'Brand': ['Maruti', 'Hyundai', 'Tata', 'Mahindra', 'Maruti', 'Hyundai', 'Renault', 'Tata', 'Maruti'],
    'Year': [2012, 2014, 2011, 2015, 2012, 2016, 2014, 2018, 2019],
    'Kms Driven': [50000, 30000, 60000, 25000, 10000, 46000, 31000, 15000, 12000],
    'City': ['Gurgaon', 'Delhi', 'Mumbai', 'Delhi', 'Mumbai', 'Delhi', 'Mumbai', 'Chennai', 'Ghaziabad'],
    'Mileage': [28, 27, 25, 26, 28, 29, 24, 21, 24]}
```

```
df = pd.DataFrame(data)
df
```

	Brand	Year	Kms Driven	City	Mileage	
0	Maruti	2012	50000	Gurgaon	28	
1	Hyundai	2014	30000	Delhi	27	
2	Tata	2011	60000	Mumbai	25	
3	Mahindra	2015	25000	Delhi	26	
4	Maruti	2012	10000	Mumbai	28	
5	Hyundai	2016	46000	Delhi	29	
6	Renault	2014	31000	Mumbai	24	
7	Tata	2018	15000	Chennai	21	
8	Maruti	2019	12000	Ghaziabad	24	

```
# Select cars with brand 'Maruti' and Mileage > 25
selection = df[(df['Brand'] == 'Maruti') & (df['Mileage'] > 25)]
selection
```

	Brand	Year	Kms Driven	City	Mileage	
0	Maruti	2012	50000	Gurgaon	28	
4	Maruti	2012	10000	Mumbai	28	

```
# Select a range of rows from 2 to 5
selection1 = df.iloc[2:6]
selection1
```

	Brand	Year	Kms Driven	City	Mileage	
2	Tata	2011	60000	Mumbai	25	
3	Mahindra	2015	25000	Delhi	26	
4	Maruti	2012	10000	Mumbai	28	
5	Hyundai	2016	46000	Delhi	29	

```
# Update values of 'Mileage' if Year < 2015
df.loc[df['Year'] < 2015, 'Mileage'] += 30
```

```
0    True
1    True
2    True
4    True
6    True
Name: Mileage, dtype: bool
```

```
# Select 0th, 2th, 4th, and 7th index rows
selection2 = df.iloc[[0, 2, 4, 7]]
selection2
```

	Brand	Year	Kms Driven	City	Mileage	
0	Maruti	2012	50000	Gurgaon	30	
2	Tata	2011	60000	Mumbai	30	
4	Maruti	2012	10000	Mumbai	30	
7	Tata	2018	15000	Chennai	21	

```
# Select rows from 1 to 4 and columns from 2 to 4
selection3 = df.iloc[1:5, 2:5]
selection3
```

	Kms Driven	City	Mileage	
1	30000	Delhi	30	
2	60000	Mumbai	30	
3	25000	Delhi	26	
4	10000	Mumbai	30	

9- Create a dataframe using the following dictionary and labels and answer the questions:

```
{'school_code': ['s001','s002','s003','s001','s002','s004'], 'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'], 'name': ['Alberto Franco','Gino
Mcneill','Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'], 'date_Of_Birth ': ['15-05-2002','17-05-2002','16-02-
1999','25-09-1998','11-05-2002','15-09-1997'], 'age': [12, 12, 13, 13, 14, 12], 'height': [173, 192, 186, 167, 151, 159],
'weight': [35, 32, 33, 30, 31, 32], 'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']}
```

```
labels: ['S1', 'S2', 'S3', 'S4', 'S5', 'S6']
```

- Write a Pandas program to split the dataframe into groups based on school code.

- Write a Pandas program to split the dataframe into groups based on school code and class.

```
# Create the DataFrame
data = {
    'school_code': ['s001', 's002', 's003', 's001', 's002', 's004'],
    'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'],
    'name': ['Alberto Franco', 'Gino Mcneill', 'Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],
    'date_Of_Birth': ['15-05-2002', '17-05-2002', '16-02-1999', '25-09-1998', '11-05-2002', '15-09-1997'],
    'age': [12, 12, 13, 13, 14, 12],
    'height': [173, 192, 186, 167, 151, 159],
    'weight': [35, 32, 33, 30, 31, 32],
    'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']}

labels = ['S1', 'S2', 'S3', 'S4', 'S5', 'S6']

df = pd.DataFrame(data, index=labels)
df
```

	school_code	class	name	date_Of_Birth	age	height	weight	address	
S1	s001	V	Alberto Franco	15-05-2002	12	173	35	street1	
S2	s002	V	Gino Mcneill	17-05-2002	12	192	32	street2	
S3	s003	VI	Ryan Parkes	16-02-1999	13	186	33	street3	
S4	s001	VI	Eesha Hinton	25-09-1998	13	167	30	street1	
S5	s002	V	Gino Mcneill	11-05-2002	14	151	31	street2	
S6	s004	VI	David Parkes	15-09-1997	12	159	32	street4	

```
# Split the DataFrame into groups based on school code
groups_by_school_code = df.groupby('school_code')
for name, group in groups_by_school_code:
    print("\nGroup:", name)
    print(group)
```

Group: s001

	school_code	class	name	date_Of_Birth	age	height	weight	\
S1	s001	V	Alberto Franco	15-05-2002	12	173	35	
S4	s001	VI	Eesha Hinton	25-09-1998	13	167	30	

	address
S1	street1
S4	street1

Group: s002

	school_code	class	name	date_Of_Birth	age	height	weight	address
S2	s002	V	Gino Mcneill	17-05-2002	12	192	32	street2
S5	s002	V	Gino Mcneill	11-05-2002	14	151	31	street2

Group: s003

	school_code	class	name	date_Of_Birth	age	height	weight	address
S3	s003	VI	Ryan Parkes	16-02-1999	13	186	33	street3

Group: s004

	school_code	class	name	date_Of_Birth	age	height	weight	address
S6	s004	VI	David Parkes	15-09-1997	12	159	32	street4

```
# Split the DataFrame into groups based on school code and class
groups_by_school_and_class = df.groupby(['school_code', 'class'])
for name, group in groups_by_school_and_class:
    print("\nGroup:", name)
    print(group)
```

Group: ('s001', 'V')

	school_code	class	name	date_Of_Birth	age	height	weight	\
S1	s001	V	Alberto Franco	15-05-2002	12	173	35	

address

S1 street1

Group: ('s001', 'VI')

	school_code	class	name	date_Of_Birth	age	height	weight	address
S4	s001	VI	Eesha Hinton	25-09-1998	13	167	30	street1

Group: ('s002', 'V')

	school_code	class	name	date_Of_Birth	age	height	weight	address
S2	s002	V	Gino Mcneill	17-05-2002	12	192	32	street2
S5	s002	V	Gino Mcneill	11-05-2002	14	151	31	street2

Group: ('s003', 'VI')

	school_code	class	name	date_Of_Birth	age	height	weight	address
S3	s003	VI	Ryan Parkes	16-02-1999	13	186	33	street3

Group: ('s004', 'VI')

	school_code	class	name	date_Of_Birth	age	height	weight	address
S6	s004	VI	David Parkes	15-09-1997	12	159	32	street4

10- First, create a dictionary for the following data, then create a dataframe based on that. Finally save the data frame as a csv file.



```
name = ["aparna", "pankaj", "sudhir", "Geeku"]
```

```
deg = ["MBA", "BCA", "M.Tech", "MBA"]
```

```
scr = [90, 40, 80, 98]
```

```
data = {
    'name': ["aparna", "pankaj", "sudhir", "Geeku"],
    'degree': ["MBA", "BCA", "M.Tech", "MBA"],
    'score': [90, 40, 80, 98]}
```

```
# Create DataFrame
df = pd.DataFrame(data)
df
```

	name	degree	score	
0	aparna	MBA	90	
1	pankaj	BCA	40	
2	sudhir	M.Tech	80	
3	Geeku	MBA	98	

```
# Save the DataFrame as a CSV file
```



```
df.to_csv('data.csv', index=False)  
print("DataFrame saved as data.csv")
```

DataFrame saved as data.csv

11- Convert the previously created csv file to a dataframe.

12- Consider the dataset of flights departing from NYC in 2013 and answer the following questions:

- Display the first five rows of the dataframe
- Extract the main information of the dataframe
- Display the basic statistical information of the dataframe
- Select flights with origin from JFK airport
- Select flights with 2 letters carrier code B6 with origin from JFK airport

```
df = pd.read_csv("https://raw.githubusercontent.com/JackyP/testing/master/datasets/nycflights.csv", index_c  
df
```