

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("saurabh00007/diabetescsv")

print("Path to dataset files:", path)

Downloading from https://www.kaggle.com/api/v1/datasets/download/saurabh00007/diabetescsv?dataset\_version\_number=1...
100%|██████████| 8.91k/8.91k [00:00<00:00, 10.4MB/s]Extracting files...
Path to dataset files: /root/.cache/kagglehub/datasets/saurabh00007/diabetescsv/versions/1
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import pandas as pd
```

```
# create a data frame
```

```
data = {
    'Name': ['Alice', 'Bob', 'Charile', 'David'],
    'Age': [25, 30, 17, 28],
    'City': ['New york', 'San Francisco', 'Los Anglos', 'Alexandria'],
    'Salary': [100, 150, 200, 300]
}
```

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

```

Name  Age      City  Salary
0  Alice   25  New york    100
1   Bob   30 San Francisco   150
2 Charile  17  Los Anglos    200
3  David  28  Alexandria    300
```

```
df[['Name', 'Salary']]
```

```

Name  Salary
0  Alice    100
1   Bob    150
2 Charile   200
3  David    300
```

```
df.iloc[2]
```

```

      2
Name  Charile
Age      17
City  Los Anglos
Salary    200
```

```
# calculate some statistics on each column
```

```
df['Age'].mean()
```

```
25.0
```

```
print("The minimum salary is", df['Salary'].min())
print("The maximum salary is", df['Salary'].max())
print("The mean salary is", df['Salary'].mean())
```

```
↕ The minimum salary is 17
The maximum salary is 30
The mean salary is 25.0
```

```
ndata = [10,20,30,40,50]
s = pd.Series(ndata)
```

```
print(s.min())
print(s.max())
print(s.mean())
```

```
↕ 10
50
30.0
```

df

```
↕
```

	Name	Age	City	Salary
0	Alice	25	New york	100
1	Bob	30	San Francisco	150
2	Charile	17	Los Anglos	200
3	David	28	Alexandria	300

df.head(2)

```
↕
```

	Name	Age	City	Salary
0	Alice	25	New york	100
1	Bob	30	San Francisco	150

df.tail()

```
↕
```

	Name	Age	City	Salary
0	Alice	25	New york	100
1	Bob	30	San Francisco	150
2	Charile	17	Los Anglos	200
3	David	28	Alexandria	300

df.info()

```
↕ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Name    4 non-null       object
1   Age     4 non-null       int64
2   City    4 non-null       object
3   Salary  4 non-null       int64
dtypes: int64(2), object(2)
memory usage: 260.0+ bytes
```

df.describe().T

```
↕
```

	count	mean	std	min	25%	50%	75%	max
Age	4.0	25.0	5.715476	17.0	23.0	26.5	28.5	30.0
Salary	4.0	187.5	85.391256	100.0	137.5	175.0	225.0	300.0

df

	Name	Age	City	Salary
0	Alice	25	New york	100
1	Bob	30	San Francisco	150
2	Charile	17	Los Anglos	200
3	David	28	Alexandria	300

```
merged_df = pd.merge(df,df, on='Name')
merged_df
```

	Name	Age_x	City_x	Salary_x	Age_y	City_y	Salary_y
0	Alice	25	New york	100	25	New york	100
1	Bob	30	San Francisco	150	30	San Francisco	150
2	Charile	17	Los Anglos	200	17	Los Anglos	200
3	David	28	Alexandria	300	28	Alexandria	300

```
# Apply a function on a column
df['New_Salary'] = df['Salary'].apply(lambda x: x + 30)
df
```

	Name	Age	City	Salary	New_Salary
0	Alice	25	New york	100	130
1	Bob	30	San Francisco	150	180
2	Charile	17	Los Anglos	200	230
3	David	28	Alexandria	300	330

```
# sorting the data based on a column
```

```
sorted_df = df.sort_values(by='Age')
sorted_df
```

	Name	Age	City	Salary	New_Salary
2	Charile	17	Los Anglos	200	230
0	Alice	25	New york	100	130
3	David	28	Alexandria	300	330
1	Bob	30	San Francisco	150	180

```
# create a new column based on certian conditions
df['is_adult'] = df['Age'].apply(lambda x: True if x>=18 else False)
df
```

	Name	Age	City	Salary	New_Salary	is_adult
0	Alice	25	New york	100	130	True
1	Bob	30	San Francisco	150	180	True
2	Charile	17	Los Anglos	200	230	False
3	David	28	Alexandria	300	330	True

```
df1 = pd.read_csv('/root/.cache/kagglehub/datasets/saurabh00007/diabetescsv/versions/1/diabetes.csv')
```

```
df1
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows x 9 columns

```
df1['Outcome'].unique()
```

```
array([1, 0])
```

```
# develop correlation matrix
```

```
correlation_matrix = df1.corr()
correlation_matrix
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

```
df1 = df1.rename(columns={"DiabetesPedigreeFunction": "DPF"})
df1
```



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows x 9 columns