1) Loading the dataset

```
import pandas as pd
# Load the dataset
df = pd.read_csv('/content/sample_data/rain (1).csv')
```

2) EDA

```
# Dimensions of the dataframe
df.shape

# Datatypes of all the attributes
df.dtypes

#first five rows of the dataframe
df.head()

# basic stats
df.describe()

# Summary of dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4116 entries, 0 to 4115
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   SUBDIVISION  4116 non-null   object 
 1   YEAR         4116 non-null   int64  
 2   JAN          4112 non-null   float64
 3   FEB          4113 non-null   float64
 4   MAR          4110 non-null   float64
 5   APR          4112 non-null   float64
 6   MAY          4113 non-null   float64
 7   JUN          4111 non-null   float64
 8   JUL          4109 non-null   float64
 9   AUG          4112 non-null   float64
 10  SEP          4110 non-null   float64
 11  OCT          4109 non-null   float64
 12  NOV          4105 non-null   float64
 13  DEC          4106 non-null   float64
 14  ANNUAL       4090 non-null   float64
 15  Jan-Feb      4110 non-null   float64
 16  Mar-May      4107 non-null   float64
 17  Jun-Sep      4106 non-null   float64
 18  Oct-Dec      4103 non-null   float64
dtypes: float64(17), int64(1), object(1)
memory usage: 611.1+ KB
```

3) Handling missing values

```
# Check missing values in each attributes
print(df.isnull().sum())

# Mean imputation to fill missing values
for column in df.columns:
  if df[column].dtype == 'object':
    df[column].fillna(df[column].mode()[0], inplace=True)
  else:
    df[column].fillna(df[column].mean(), inplace=True)

# Try using this also
# df = df.fillna(df.select_dtypes(include='number').mean())

# After imputing missing values
print(df.isnull().sum())
```

```
SUBDIVISION     0
YEAR            0
JAN             4
FEB             3
MAR             6
APR             4
MAY             3
JUN             5
JUL             7
AUG             4
SEP             6
```

```
OCT            7
NOV           11
DEC           10
ANNUAL        26
Jan-Feb        6
Mar-May        9
Jun-Sep       10
Oct-Dec       13
dtype: int64
SUBDIVISION    0
YEAR           0
JAN            0
FEB            0
MAR            0
APR            0
MAY            0
JUN            0
JUL            0
AUG            0
SEP            0
OCT            0
NOV            0
DEC            0
ANNUAL         0
Jan-Feb        0
Mar-May        0
Jun-Sep        0
Oct-Dec        0
dtype: int64
```

4) Standardization

```
# Standardization transforms the data to have a mean of 0 and a standard deviation of 1

from sklearn.preprocessing import StandardScaler
# Select columns for standardization (excluding 'SUBDIVISION' and 'YEAR')
rainfall_columns = df.columns[2:]

# Apply standardization
scaler = StandardScaler()
df[rainfall_columns] = scaler.fit_transform(df[rainfall_columns])
df
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	
0	ANDAMAN & NICOBAR ISLANDS	1901	0.901019	1.819197	0.039233	-0.602264	3.596952	1.224806	0.066421	1.011559	0.999593	2.946952	7.5
1	ANDAMAN & NICOBAR ISLANDS	1902	-0.564795	3.844716	-0.323090	-0.636192	2.925549	1.308374	-0.439377	2.456519	3.465350	1.022838	4.6
2	ANDAMAN & NICOBAR ISLANDS	1903	-0.186424	3.404507	-0.583110	-0.621441	1.212540	1.064492	1.415586	0.193137	1.046898	0.861909	3.5
3	ANDAMAN & NICOBAR ISLANDS	1904	-0.284741	-0.197964	-0.583110	2.349501	1.775966	1.129300	0.574818	-0.689952	4.605097	1.274291	3.9
4	ANDAMAN & NICOBAR ISLANDS	1905	-0.526064	-0.607525	-0.512776	-0.239378	1.573002	1.698926	0.079790	0.213280	0.736461	1.661527	-0.2
...	...	...	...	...	...	...	...	...	...	...	...	...	
4111	LAKSHADWEEP	2011	-0.412851	-0.529514	-0.517039	0.630957	0.174180	-0.326744	0.011088	-0.192220	0.427502	0.220202	2.1
4112	LAKSHADWEEP	2012	0.007230	-0.604739	-0.549009	0.496719	-0.524014	0.412577	-0.429721	0.482023	-0.129806	0.506858	-0.4
4113	LAKSHADWEEP	2013	0.215781	0.350904	0.216132	-0.558009	0.020739	0.835533	-0.188706	-0.720166	-0.128328	-0.228389	0.5
4114	LAKSHADWEEP	2014	1.020191	-0.158958	-0.489332	-0.416395	-0.230123	0.059118	-0.858275	0.932049	-0.481635	0.741211	0.2
4115	LAKSHADWEEP	2015	-0.499250	-0.593595	-0.504251	0.648659	0.384450	0.282961	-0.333167	-0.762571	-0.273199	0.702991	2.7

4116 rows × 19 columns

Next steps: Generate code with df View recommended plots New interactive sheet

5) Normalization

```
# Normalization rescales the data to fit within a specific range, typically [0, 1]
```

```
from sklearn.preprocessing import MinMaxScaler
# Apply normalization (to range [0,1])
normalizer = MinMaxScaler()
df[rainfall_columns] = normalizer.fit_transform(df[rainfall_columns])
df
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	ANDAMAN & NICOBAR ISLANDS	1901	0.084290	0.215861	0.048217	0.003865	0.452507	0.321280	0.154520	0.289018	0.272117	0.409680	0.860225
1	ANDAMAN & NICOBAR ISLANDS	1902	0.000000	0.396035	0.020145	0.000000	0.381739	0.333458	0.096877	0.452781	0.545135	0.207951	0.553244
2	ANDAMAN & NICOBAR ISLANDS	1903	0.021758	0.356877	0.000000	0.001680	0.201181	0.297919	0.308278	0.196263	0.277355	0.191079	0.438280
3	ANDAMAN & NICOBAR ISLANDS	1904	0.016104	0.036431	0.000000	0.340111	0.260568	0.307363	0.212460	0.096179	0.671332	0.234314	0.475728
4	ANDAMAN & NICOBAR ISLANDS	1905	0.002227	0.000000	0.005449	0.045202	0.239175	0.390370	0.156044	0.198546	0.242982	0.274913	0.039143
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4111	LAKSHADWEEP	2011	0.008737	0.006939	0.005119	0.144345	0.091734	0.095185	0.148214	0.152589	0.208773	0.123800	0.284019
4112	LAKSHADWEEP	2012	0.032894	0.000248	0.002642	0.129054	0.018141	0.202920	0.097977	0.229004	0.147066	0.153854	0.019109
4113	LAKSHADWEEP	2013	0.044886	0.085254	0.061922	0.008906	0.075560	0.264554	0.125444	0.092755	0.147230	0.076769	0.120358
4114	LAKSHADWEEP	2014	0.091143	0.039901	0.007266	0.025038	0.049119	0.151413	0.049137	0.280007	0.108110	0.178425	0.090923
4115	LAKSHADWEEP	2015	0.003769	0.001239	0.006110	0.146362	0.113897	0.184032	0.108981	0.087949	0.131189	0.174417	0.355987

4116 rows × 19 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

6) Log Transformation

```
# Log transformation is used to stabilize variance and make the data more normally distributed, especially for skewed data.

import numpy as np
# Log transformation (adding 1 to avoid log(0))
df[rainfall_columns] = np.log1p(df[rainfall_columns])
df
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	ANDAMAN & NICOBAR ISLANDS	1901	0.080925	0.195453	0.047090	0.003857	0.373291	0.278601	0.143685	0.253881	0.240683	0.343363	0.620697
1	ANDAMAN & NICOBAR ISLANDS	1902	0.000000	0.333636	0.019945	0.000000	0.323343	0.287775	0.092467	0.373480	0.435111	0.188926	0.440346
2	ANDAMAN & NICOBAR ISLANDS	1903	0.021524	0.305186	0.000000	0.001679	0.183305	0.260762	0.268712	0.179203	0.244791	0.174859	0.363448
3	ANDAMAN & NICOBAR ISLANDS	1904	0.015976	0.035783	0.000000	0.292752	0.231563	0.268012	0.192651	0.091831	0.513621	0.210515	0.389152
4	ANDAMAN & NICOBAR ISLANDS	1905	0.002225	0.000000	0.005434	0.044211	0.214446	0.329570	0.145004	0.181109	0.217514	0.242878	0.038396
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4111	LAKSHADWEEP	2011	0.008699	0.006915	0.005106	0.134833	0.087767	0.090923	0.138208	0.142011	0.189606	0.116716	0.249995
4112	LAKSHADWEEP	2012	0.032364	0.000248	0.002639	0.121380	0.017979	0.184752	0.093469	0.206204	0.137207	0.143108	0.018929
4113	LAKSHADWEEP	2013	0.043908	0.081814	0.060081	0.008867	0.072842	0.234720	0.118178	0.088702	0.137350	0.073965	0.113648
4114	LAKSHADWEEP	2014	0.087226	0.039125	0.007239	0.024729	0.047950	0.140990	0.047968	0.246866	0.102656	0.164178	0.087024
4115	LAKSHADWEEP	2015	0.003762	0.001238	0.006091	0.136593	0.107865	0.168926	0.103441	0.084294	0.123269	0.160772	0.304530

4116 rows × 19 columns

Next steps: Generate code with `df` View recommended plots New interactive sheet

7) Aggregation

```
# Aggregation is a way to group data and compute aggregate functions, such as the mean, sum, or count.

# Aggregating the data by 'SUBDIVISION' and 'YEAR' (calculating the mean for each group)
rain_aggregated = df.groupby(['SUBDIVISION','YEAR']).mean().reset_index()
rain_aggregated
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	
0	ANDAMAN & NICOBAR ISLANDS	1901	0.080925	0.195453	0.047090	0.003857	0.373291	0.278601	0.143685	0.253881	0.240683	0.343363	0.620697	0.0
1	ANDAMAN & NICOBAR ISLANDS	1902	0.000000	0.333636	0.019945	0.000000	0.323343	0.287775	0.092467	0.373480	0.435111	0.188926	0.440346	0.2
2	ANDAMAN & NICOBAR ISLANDS	1903	0.021524	0.305186	0.000000	0.001679	0.183305	0.260762	0.268712	0.179203	0.244791	0.174859	0.363448	0.3
3	ANDAMAN & NICOBAR ISLANDS	1904	0.015976	0.035783	0.000000	0.292752	0.231563	0.268012	0.192651	0.091831	0.513621	0.210515	0.389152	0.0
4	ANDAMAN & NICOBAR ISLANDS	1905	0.002225	0.000000	0.005434	0.044211	0.214446	0.329570	0.145004	0.181109	0.217514	0.242878	0.038396	0.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4111	WEST UTTAR PRADESH	2011	0.003591	0.025448	0.006419	0.004694	0.025014	0.103495	0.087438	0.130636	0.079873	0.000738	0.000770	0.0
4112	WEST UTTAR PRADESH	2012	0.024538	0.000248	0.002309	0.007867	0.000257	0.002234	0.059598	0.085784	0.053925	0.000527	0.000154	0.0
4113	WEST UTTAR PRADESH	2013	0.034353	0.158919	0.005763	0.002685	0.001795	0.111696	0.094394	0.159116	0.041755	0.062540	0.002616	0.0
4114	WEST UTTAR PRADESH	2014	0.079502	0.070330	0.036639	0.008867	0.009369	0.013331	0.062187	0.047513	0.066945	0.015279	0.000000	0.0

Next steps: Generate code with `rain_aggregated` View recommended plots New interactive sheet

8) Discretization

```
# Discretization involves converting continuous variables into discrete categories. For example, we can categorize the ANNUAL rainfall in
# "medium", and "high" bins.

# Discretizing the 'ANNUAL' rainfall into three categories: low, medium, and high
df['rainfall_category'] = pd.cut(df['ANNUAL'], bins=[-np.inf, 0.33, 0.66, np.inf],
labels=["low", "medium", "high"])
df
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	ANDAMAN & NICOBAR ISLANDS	1901	0.080925	0.195453	0.047090	0.003857	0.373291	0.278601	0.143685	0.253881	0.240683	0.343363	0.620697
1	ANDAMAN & NICOBAR ISLANDS	1902	0.000000	0.333636	0.019945	0.000000	0.323343	0.287775	0.092467	0.373480	0.435111	0.188926	0.440346
2	ANDAMAN & NICOBAR ISLANDS	1903	0.021524	0.305186	0.000000	0.001679	0.183305	0.260762	0.268712	0.179203	0.244791	0.174859	0.363448
3	ANDAMAN & NICOBAR ISLANDS	1904	0.015976	0.035783	0.000000	0.292752	0.231563	0.268012	0.192651	0.091831	0.513621	0.210515	0.389152
4	ANDAMAN & NICOBAR ISLANDS	1905	0.002225	0.000000	0.005434	0.044211	0.214446	0.329570	0.145004	0.181109	0.217514	0.242878	0.038396
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4111	LAKSHADWEEP	2011	0.008699	0.006915	0.005106	0.134833	0.087767	0.090923	0.138208	0.142011	0.189606	0.116716	0.249995
4112	LAKSHADWEEP	2012	0.032364	0.000248	0.002639	0.121380	0.017979	0.184752	0.093469	0.206204	0.137207	0.143108	0.018929