

FINAL ASSESSMENT 2

```
In [1]: #importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: #importing dataset
data=pd.read_csv(r"C:\Users\user\Downloads\rainfall in india 1901-2015.csv")
data
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	
0	0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	:
1	1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	:
2	2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	:
3	3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	:
4	4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	:
...	
4111	4111	LAKSHADWEEP	2011	5.1	2.8	3.1	85.9	107.2	153.6	350.2	254.0	255.2	
4112	4112	LAKSHADWEEP	2012	19.2	0.1	1.6	76.8	21.2	327.0	231.5	381.2	179.8	:
4113	4113	LAKSHADWEEP	2013	26.2	34.4	37.5	5.3	88.3	426.2	296.4	154.4	180.0	
4114	4114	LAKSHADWEEP	2014	53.2	16.1	4.4	14.9	57.4	244.1	116.1	466.1	132.2	:
4115	4115	LAKSHADWEEP	2015	2.2	0.5	3.7	87.1	133.1	296.6	257.5	146.4	160.4	:

4116 rows × 20 columns

ORISSA

In [3]:

df=data.iloc[667:782]
df

Out[3]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
667	667	ORISSA	1901	39.5	65.1	16.1	51.6	79.0	78.2	288.4	307.7	185.3	76.6
668	668	ORISSA	1902	3.4	0.2	14.2	101.1	56.7	108.3	437.4	349.1	202.7	33.2
669	669	ORISSA	1903	19.7	18.9	10.5	34.6	73.3	154.3	410.4	295.2	265.6	228.5
670	670	ORISSA	1904	0.2	12.2	20.6	10.1	100.2	342.9	336.7	350.4	227.8	111.8
671	671	ORISSA	1905	24.3	17.2	66.3	56.9	107.5	92.0	330.1	281.4	344.1	36.4
...
777	777	ORISSA	2011	3.7	16.2	4.9	58.2	75.6	210.1	199.6	358.6	398.7	20.2
778	778	ORISSA	2012	50.8	3.6	0.9	34.8	21.3	169.6	324.3	417.0	242.4	66.0
779	779	ORISSA	2013	3.3	7.8	2.1	53.6	57.7	272.6	380.0	254.9	208.1	391.0
780	780	ORISSA	2014	0.0	17.6	25.1	11.7	111.9	92.2	496.2	386.3	281.1	111.8
781	781	ORISSA	2015	15.1	3.3	10.5	67.6	32.6	238.6	294.8	264.0	237.0	24.7

115 rows × 20 columns

Data Cleaning and Preprocessing

In [4]:

df.head()

Out[4]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
667	667	ORISSA	1901	39.5	65.1	16.1	51.6	79.0	78.2	288.4	307.7	185.3	76.6
668	668	ORISSA	1902	3.4	0.2	14.2	101.1	56.7	108.3	437.4	349.1	202.7	33.2
669	669	ORISSA	1903	19.7	18.9	10.5	34.6	73.3	154.3	410.4	295.2	265.6	228.5
670	670	ORISSA	1904	0.2	12.2	20.6	10.1	100.2	342.9	336.7	350.4	227.8	111.8
671	671	ORISSA	1905	24.3	17.2	66.3	56.9	107.5	92.0	330.1	281.4	344.1	36.4

In [5]: `df.tail()`

Out[5]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
777	777	ORISSA	2011	3.7	16.2	4.9	58.2	75.6	210.1	199.6	358.6	398.7	20.2
778	778	ORISSA	2012	50.8	3.6	0.9	34.8	21.3	169.6	324.3	417.0	242.4	66.0
779	779	ORISSA	2013	3.3	7.8	2.1	53.6	57.7	272.6	380.0	254.9	208.1	391.0
780	780	ORISSA	2014	0.0	17.6	25.1	11.7	111.9	92.2	496.2	386.3	281.1	111.8
781	781	ORISSA	2015	15.1	3.3	10.5	67.6	32.6	238.6	294.8	264.0	237.0	24.7

In [6]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 667 to 781
Data columns (total 20 columns):
#   Column          Non-Null Count  Dtype
---  -
0   index           115 non-null    int64
1   SUBDIVISION     115 non-null    object
2   YEAR            115 non-null    int64
3   JAN             115 non-null    float64
4   FEB             115 non-null    float64
5   MAR             115 non-null    float64
6   APR             115 non-null    float64
7   MAY             115 non-null    float64
8   JUN             115 non-null    float64
9   JUL             115 non-null    float64
10  AUG             115 non-null    float64
11  SEP             115 non-null    float64
12  OCT             115 non-null    float64
13  NOV             115 non-null    float64
14  DEC             115 non-null    float64
15  ANNUAL          115 non-null    float64
16  Jan-Feb         115 non-null    float64
17  Mar-May         115 non-null    float64
18  Jun-Sep         115 non-null    float64
19  Oct-Dec         115 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 18.1+ KB
```

```
In [7]: #filling null values
df1=df.fillna(0)
df1
```

Out[7]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
667	667	ORISSA	1901	39.5	65.1	16.1	51.6	79.0	78.2	288.4	307.7	185.3	76.6
668	668	ORISSA	1902	3.4	0.2	14.2	101.1	56.7	108.3	437.4	349.1	202.7	33.2
669	669	ORISSA	1903	19.7	18.9	10.5	34.6	73.3	154.3	410.4	295.2	265.6	228.5
670	670	ORISSA	1904	0.2	12.2	20.6	10.1	100.2	342.9	336.7	350.4	227.8	111.8
671	671	ORISSA	1905	24.3	17.2	66.3	56.9	107.5	92.0	330.1	281.4	344.1	36.4
...
777	777	ORISSA	2011	3.7	16.2	4.9	58.2	75.6	210.1	199.6	358.6	398.7	20.2
778	778	ORISSA	2012	50.8	3.6	0.9	34.8	21.3	169.6	324.3	417.0	242.4	66.0
779	779	ORISSA	2013	3.3	7.8	2.1	53.6	57.7	272.6	380.0	254.9	208.1	391.0
780	780	ORISSA	2014	0.0	17.6	25.1	11.7	111.9	92.2	496.2	386.3	281.1	111.8
781	781	ORISSA	2015	15.1	3.3	10.5	67.6	32.6	238.6	294.8	264.0	237.0	24.7

115 rows × 20 columns

```
In [8]: df1.describe()
```

Out[8]:

	index	YEAR	JAN	FEB	MAR	APR	MAY
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	724.000000	1958.000000	12.329565	19.71913	21.134783	34.160000	64.886087
std	33.341666	33.341666	15.620842	22.23721	22.259542	23.243417	39.851027
min	667.000000	1901.000000	0.000000	0.00000	0.100000	1.100000	16.200000
25%	695.500000	1929.500000	0.500000	3.45000	5.500000	18.400000	38.350000
50%	724.000000	1958.000000	5.500000	12.10000	12.600000	28.500000	57.700000
75%	752.500000	1986.500000	18.850000	28.45000	26.800000	46.850000	77.300000
max	781.000000	2015.000000	70.900000	116.20000	89.800000	148.400000	293.000000

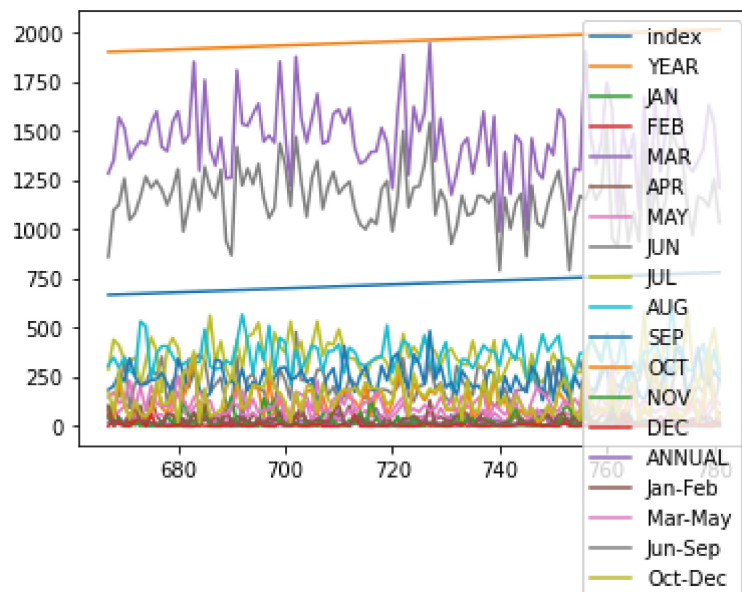
```
In [9]: df1.columns
```

Out[9]: Index(['index', 'SUBDIVISION', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC', 'ANNUAL', 'Jan-Feb', 'Mar-May', 'Jun-Sep', 'Oct-Dec'], dtype='object')

Data Visulaization

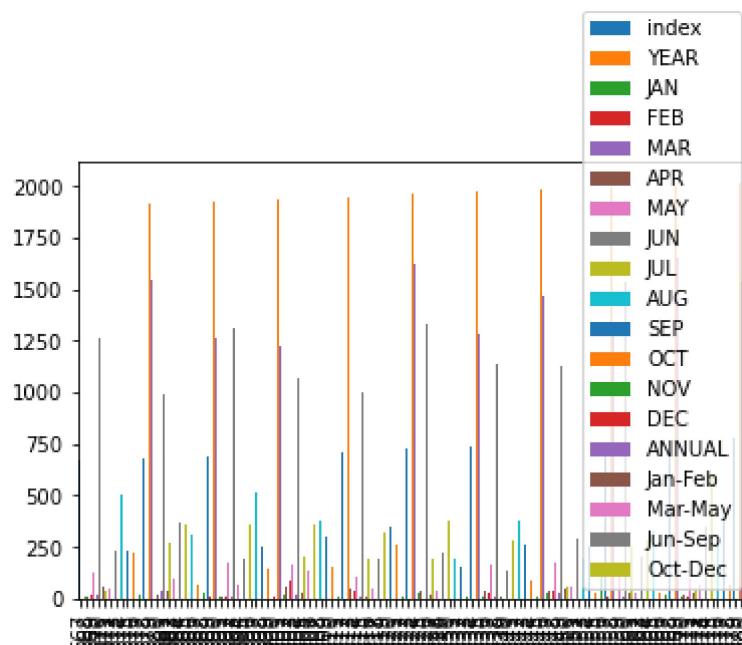
```
In [10]: df1.plot.line()
```

```
Out[10]: <AxesSubplot:>
```



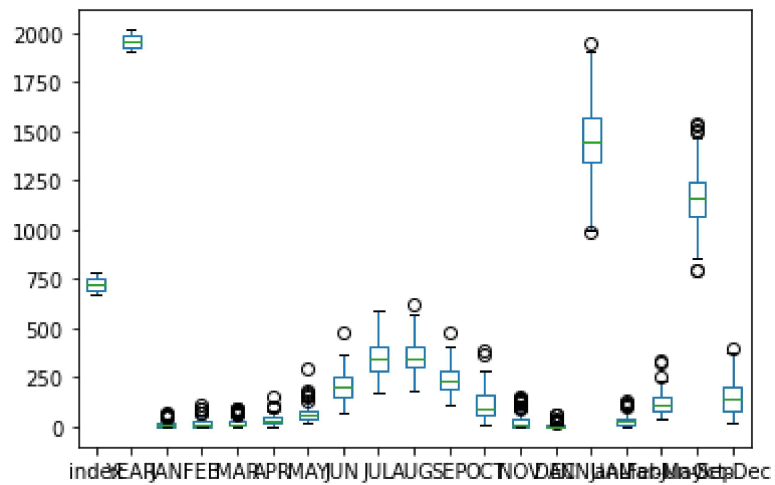
```
In [11]: df1.plot.bar()
```

```
Out[11]: <AxesSubplot:>
```



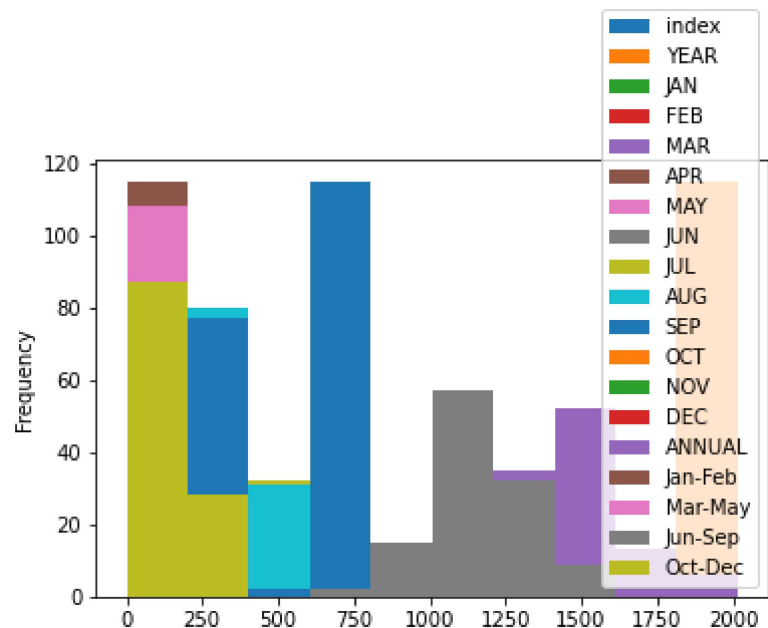
```
In [12]: df1.plot.box()
```

```
Out[12]: <AxesSubplot:>
```



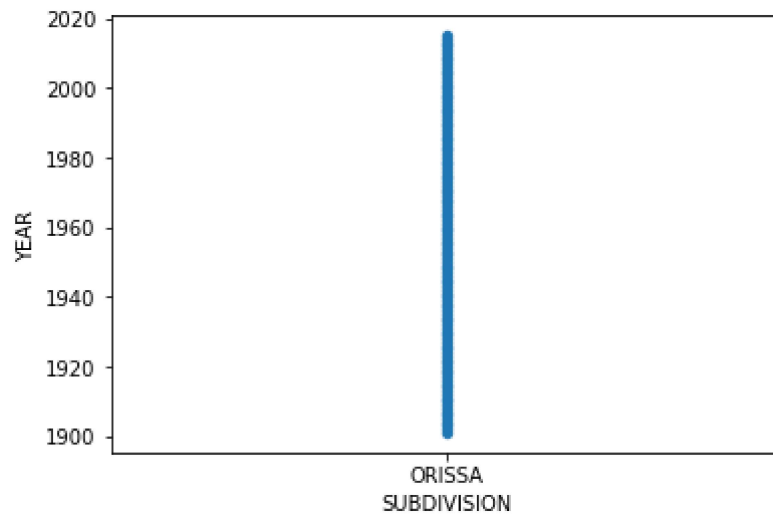
```
In [13]: df1.plot.hist()
```

```
Out[13]: <AxesSubplot:ylabel='Frequency'>
```



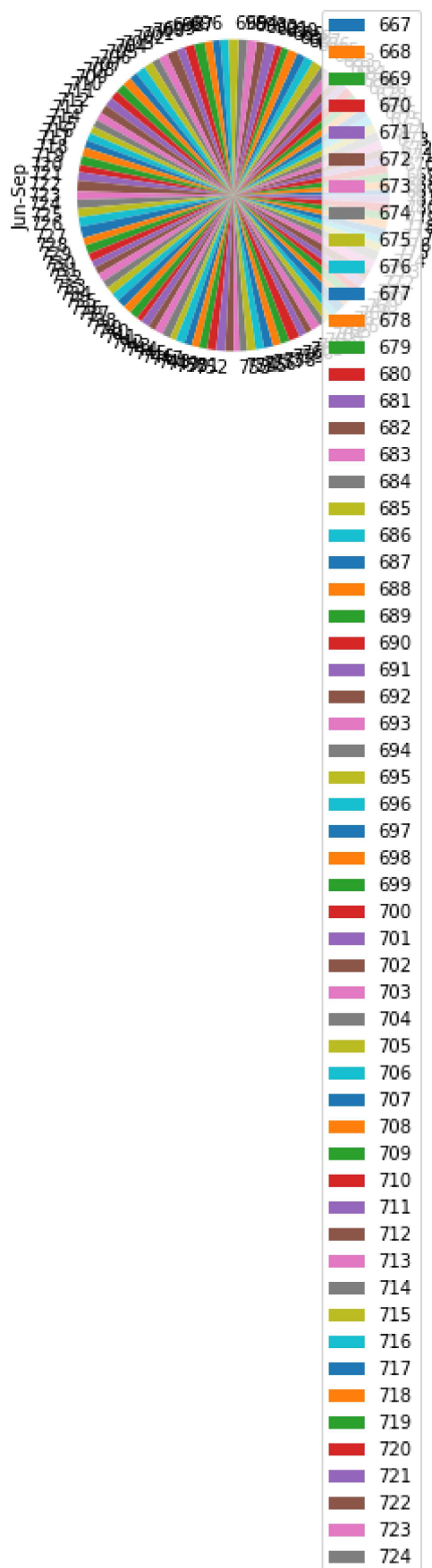
```
In [14]: df1.plot.scatter(x="SUBDIVISION",y="YEAR")
```

```
Out[14]: <AxesSubplot:xlabel='SUBDIVISION', ylabel='YEAR'>
```



```
In [15]: df2=df1[[ 'Jun-Sep']]
df2.plot.pie(subplots=True)
```

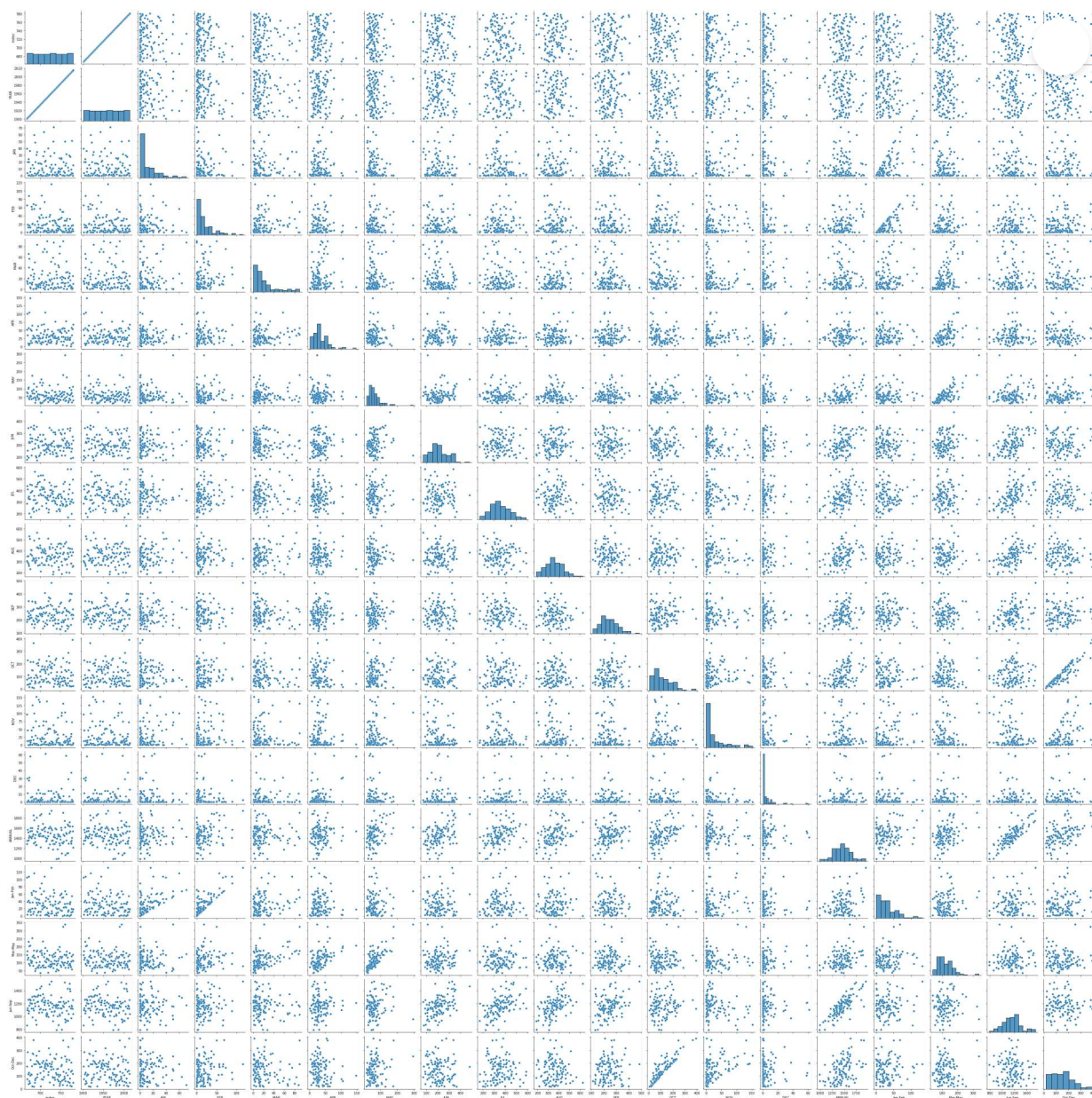
```
Out[15]: array([<AxesSubplot:ylabel='Jun-Sep'>], dtype=object)
```



725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781

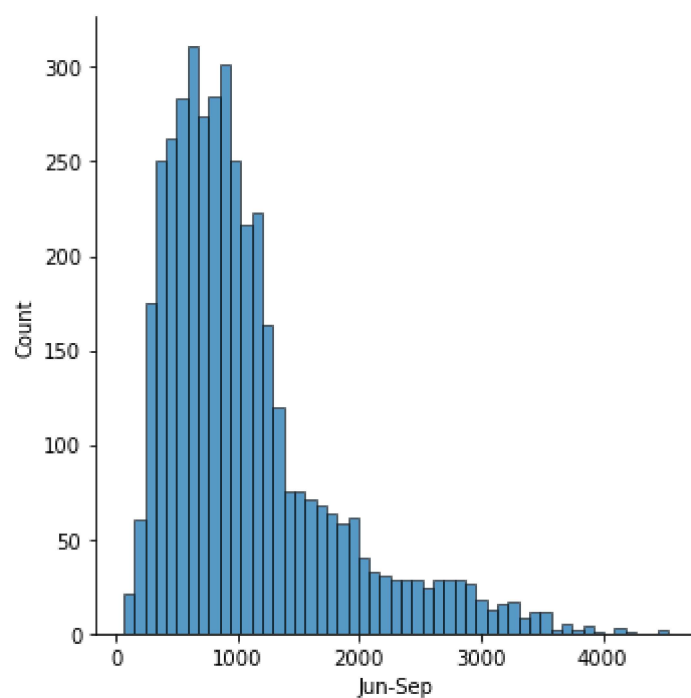
```
In [16]: sns.pairplot(df1)
```

```
Out[16]: <seaborn.axisgrid.PairGrid at 0x27f00930a90>
```



```
In [17]: sns.displot(data["Jun-Sep"])
```

```
Out[17]: <seaborn.axisgrid.FacetGrid at 0x27f0090bd90>
```



```
In [18]: sns.heatmap(df1.corr())
```

```
Out[18]: <AxesSubplot:>
```

