

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
import plotly.express as px
from wordcloud import WordCloud
from datetime import datetime
import warnings
```

```
!gdown --id 111dAAIUeL1YcuAMpf4G0mAgwH6_AcTfi
```

```
⚡ /usr/local/lib/python3.11/dist-packages/gdown/__main__.py:140: FutureWarning: Option `--id` was deprecated in version 4.3.1 and will
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=111dAAIUeL1YcuAMpf4G0mAgwH6\_AcTfi
To: /content/kerala.csv
100% 10.3k/10.3k [00:00<00:00, 19.7MB/s]
```

```
data = pd.read_csv('kerala.csv')
data
```

⚡

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL RAINFALL	FLOODS	
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6	YES	
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6	YES	
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2	YES	
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	YES	
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	NO	
...	
113	KERALA	2014	4.6	10.3	17.9	95.7	251.0	454.4	677.8	733.9	298.8	355.5	99.5	47.2	3046.4	YES	
114	KERALA	2015	3.1	5.8	50.1	214.1	201.8	563.6	406.0	252.2	292.9	308.1	223.6	79.4	2600.6	NO	
115	KERALA	2016	2.4	3.8	35.9	143.0	186.4	522.2	412.3	325.5	173.2	225.9	125.4	23.6	2176.6	NO	
116	KERALA	2017	1.9	6.8	8.9	43.6	173.5	498.5	319.6	531.8	209.5	192.4	92.5	38.1	2117.1	NO	
117	KERALA	2018	29.1	52.1	48.6	116.4	183.8	625.4	1048.5	1398.9	423.6	356.1	125.4	65.1	4473.0	YES	

118 rows × 16 columns

Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

1) what is the Average of the Rainfall in Kerala Across Every month

```
def averagecalculator(data,x):
    sumofflood=data[x].sum()
    l=len(data[x])
    avg=sumofflood/l
    return float(avg)

def averagedriver(data):
    x=['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']
    y=[]
    for i in x:
        avg = round(averagecalculator(data,i),2)
        y.append(avg)

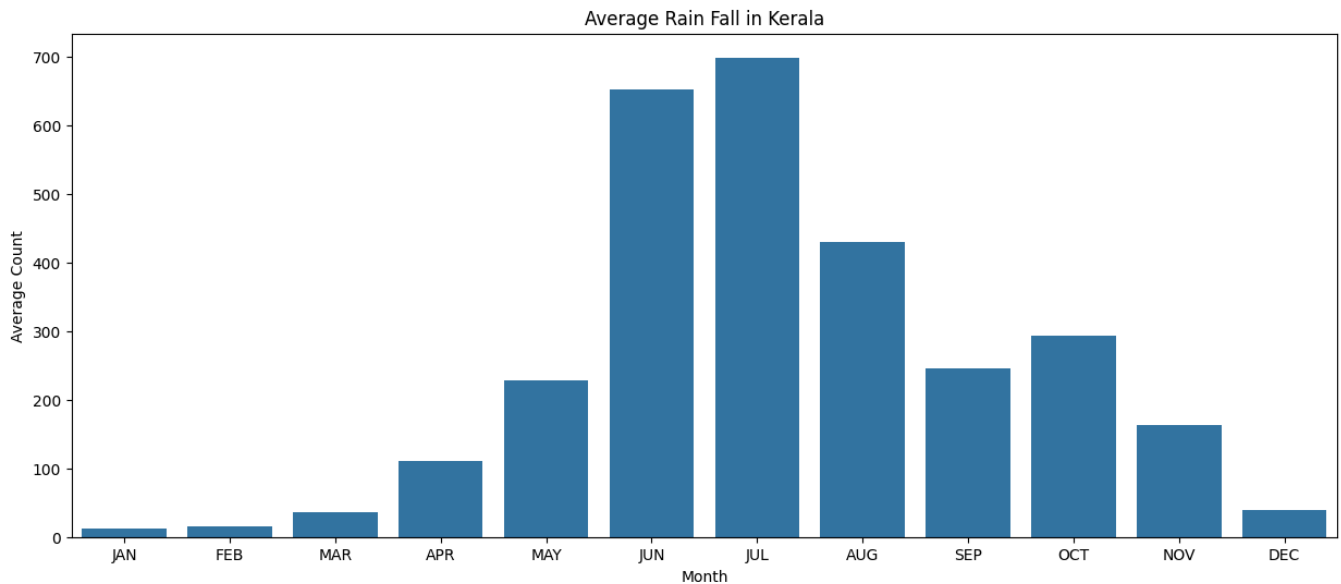
    df_avg = pd.DataFrame({
        'Month': x,
        'Average': y
    })

    print(df_avg)

    plt.figure(figsize=(15, 6))
    sns.barplot(data=df_avg, x='Month', y='Average')
    plt.xlabel('Month')
    plt.ylabel('Average Count')
    plt.title('Average Rain Fall in Kerala')
    plt.show()
    plt.close()
```

averagedriver(data)

	Month	Average
0	JAN	12.22
1	FEB	15.63
2	MAR	36.67
3	APR	110.33
4	MAY	228.64
5	JUN	651.62
6	JUL	698.22
7	AUG	430.37
8	SEP	246.21
9	OCT	293.21
10	NOV	162.31
11	DEC	40.01



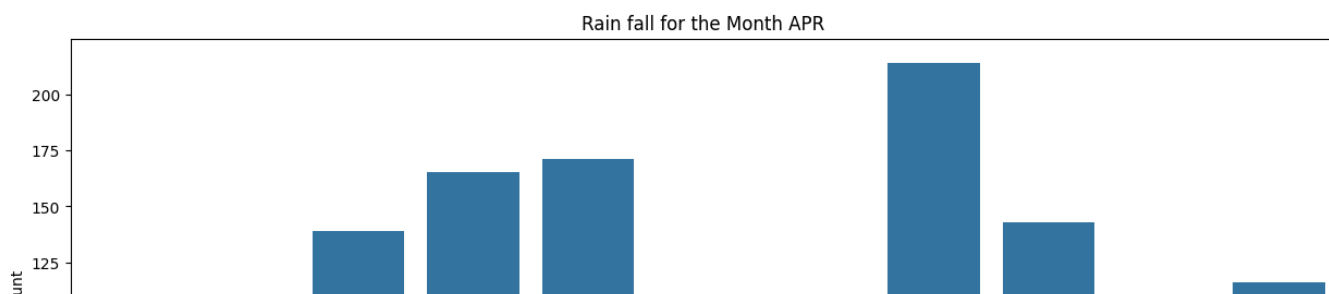
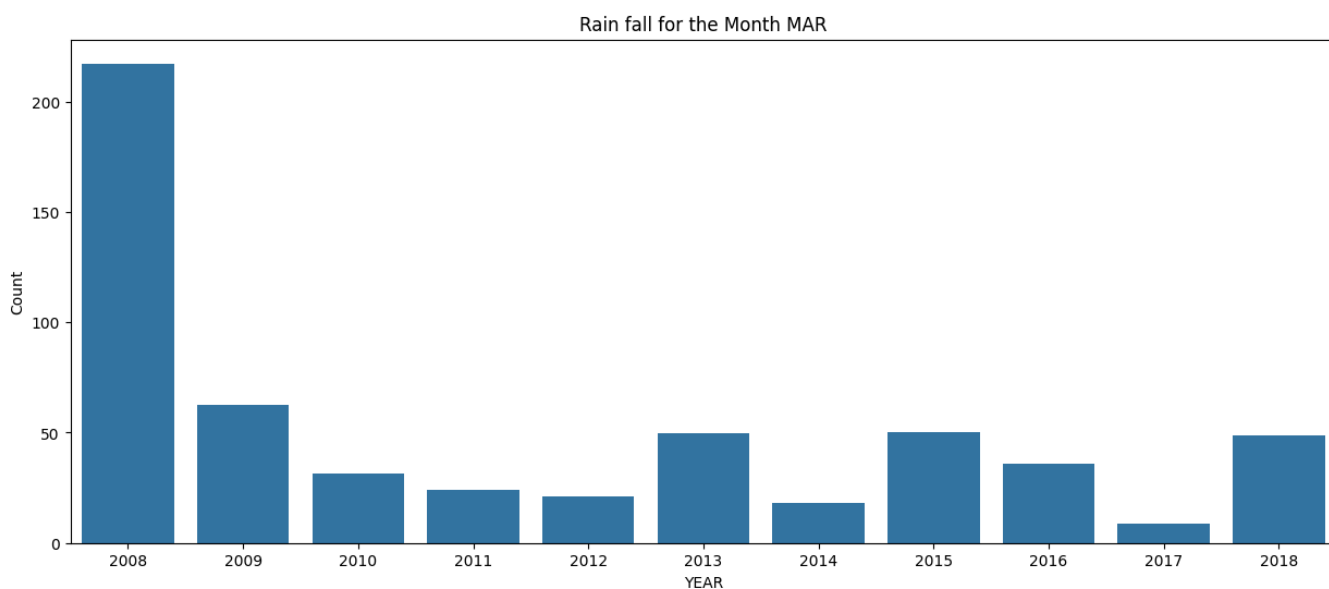
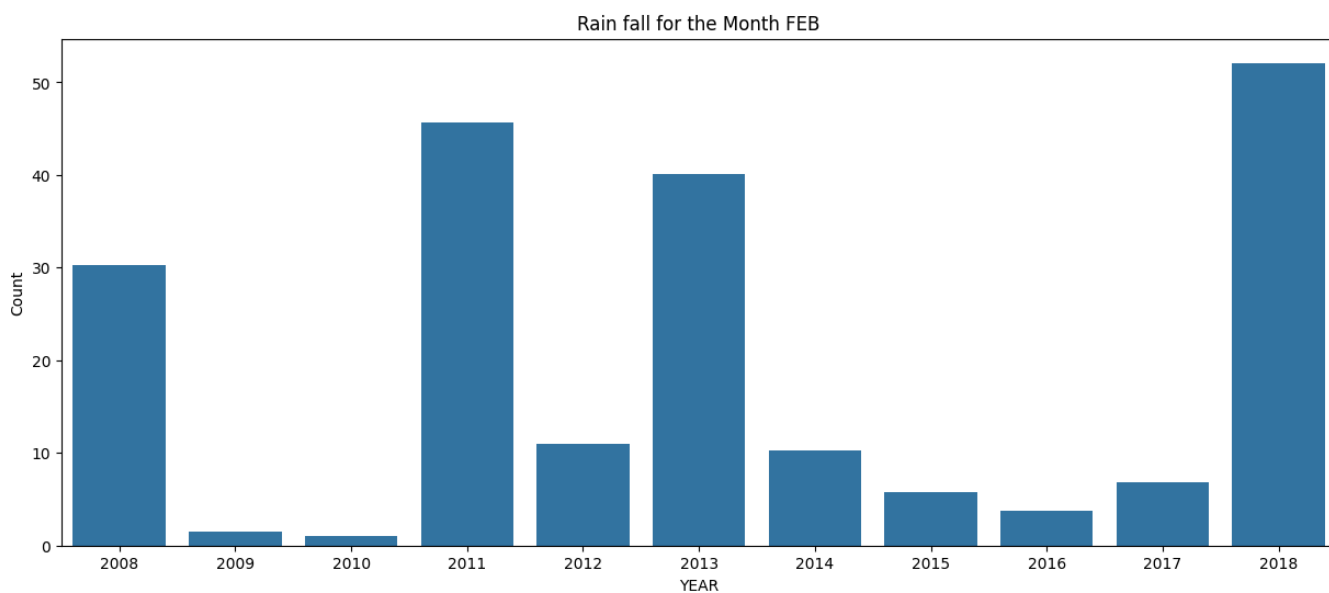
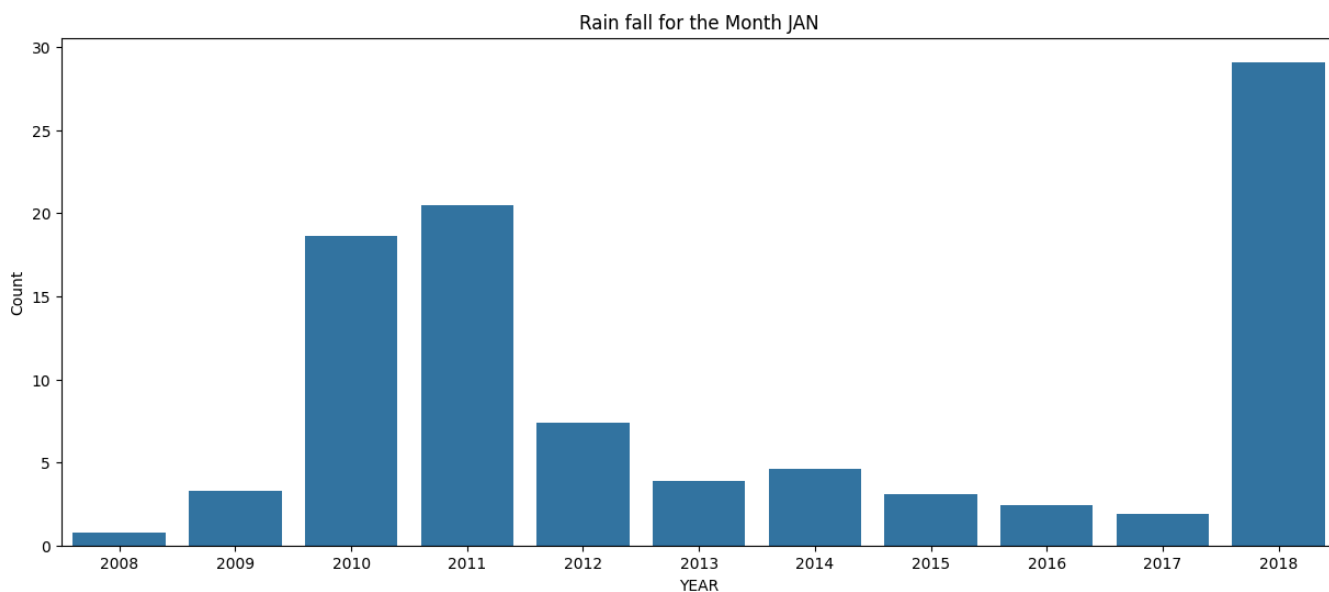
From this we can get to know that the highest rainfall occurs in the month of July which is about 698.22 and least rain fall occurs in the month of Jan which is 12.22 on an average across the years from 1901 to 2018

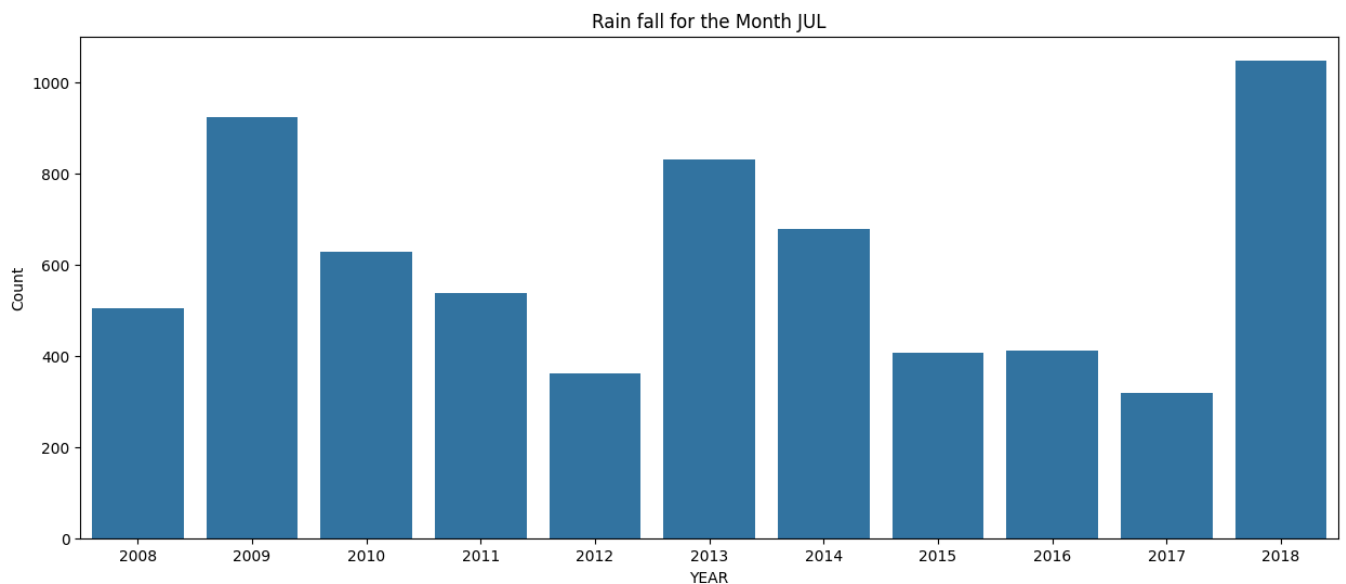
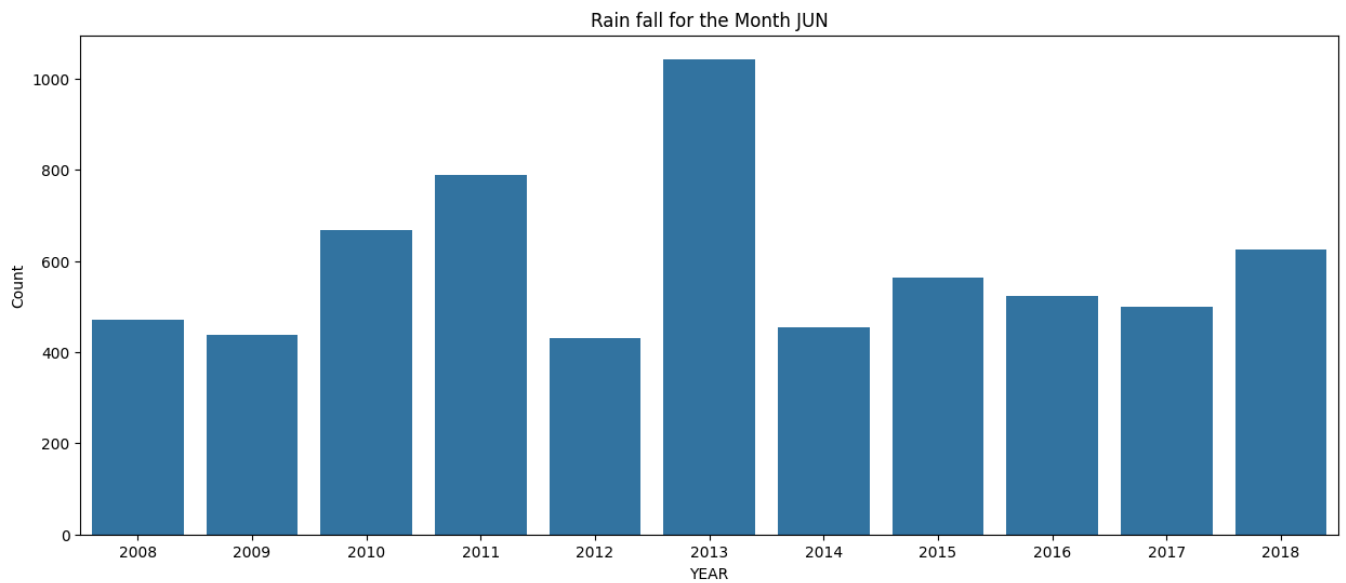
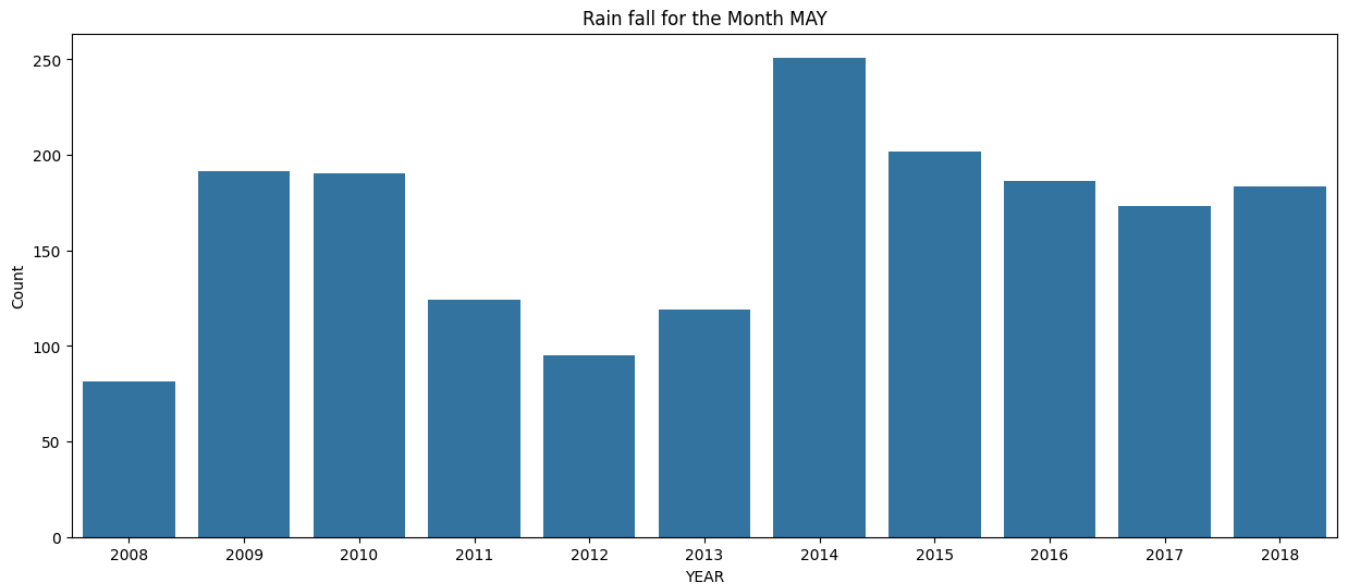
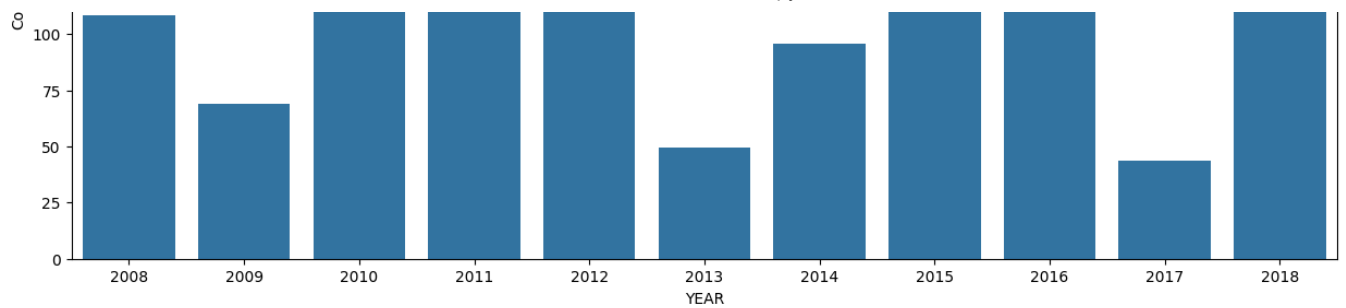
✓ 2) what is the The Rainfall in Kerala Across Every month for the last 10 years

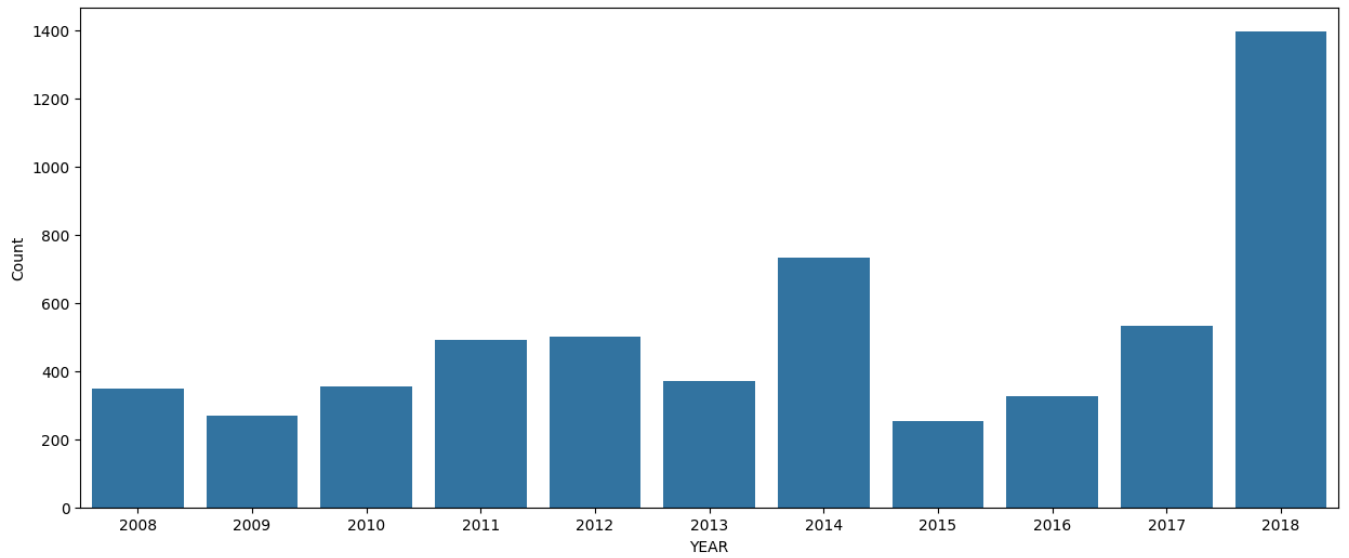
```
df_last10=data.loc[data['YEAR']>=2008]
```

```
def a10yeardriver(data):
    x=['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']
    for i in x:
        plt.figure(figsize=(15, 6))
        sns.barplot(data=data, x='YEAR', y=i)
        plt.xlabel('YEAR')
        plt.ylabel('Count')
        title_name = f"Rain fall for the Month {i}"
        plt.title(title_name)
        plt.show()
        plt.close()
```

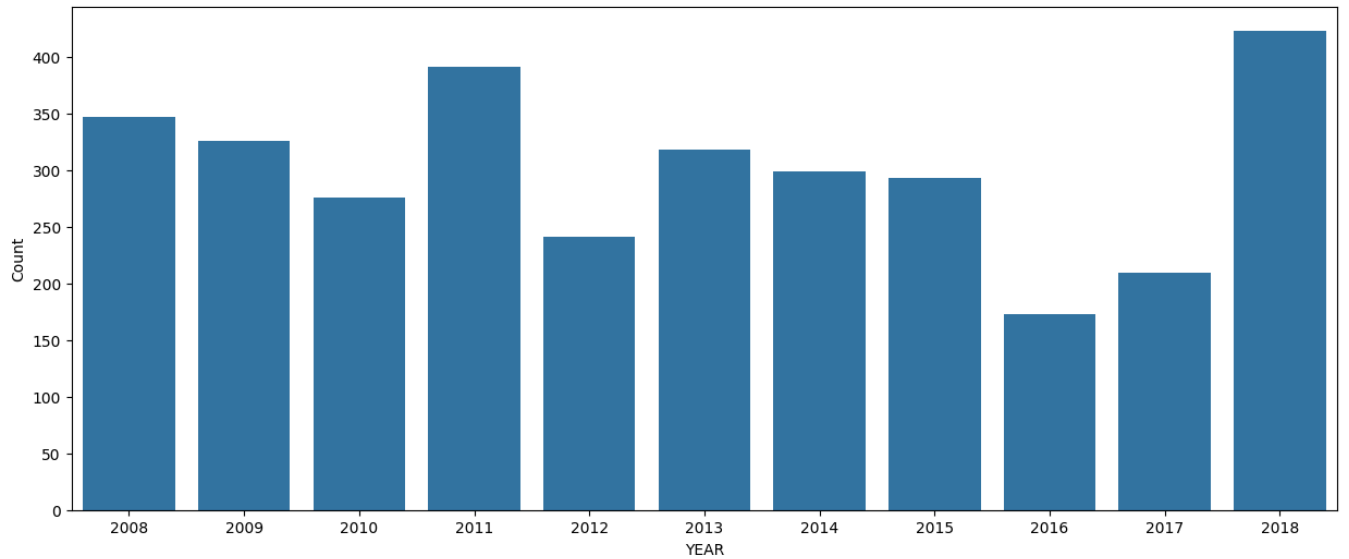
```
a10yeardriver(df_last10)
```



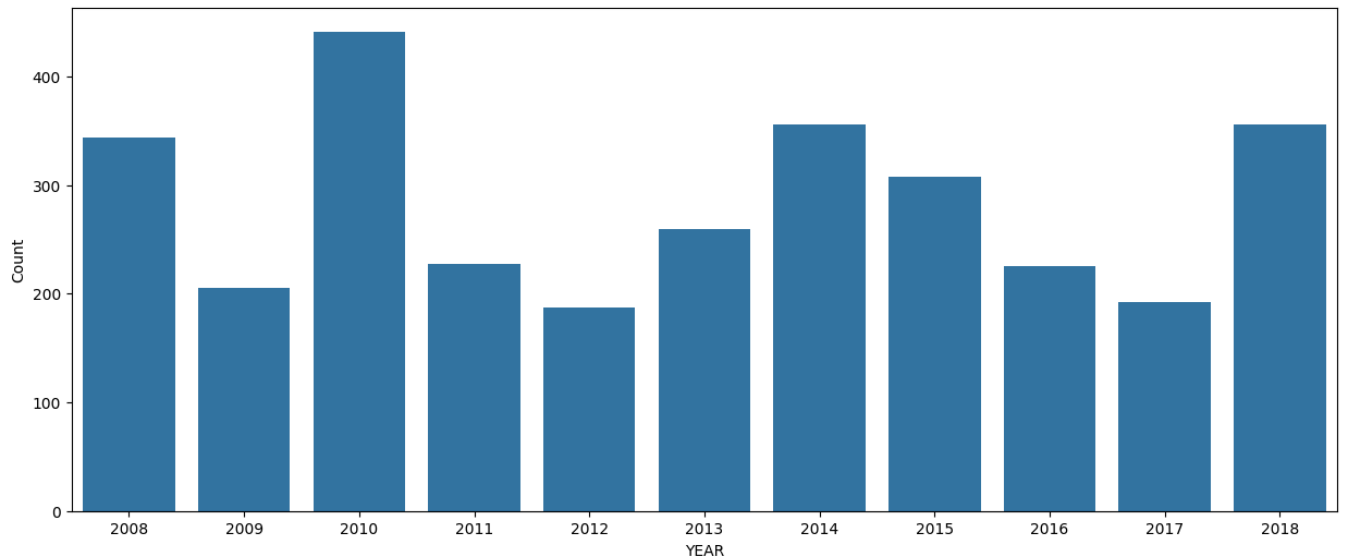




Rain fall for the Month SEP

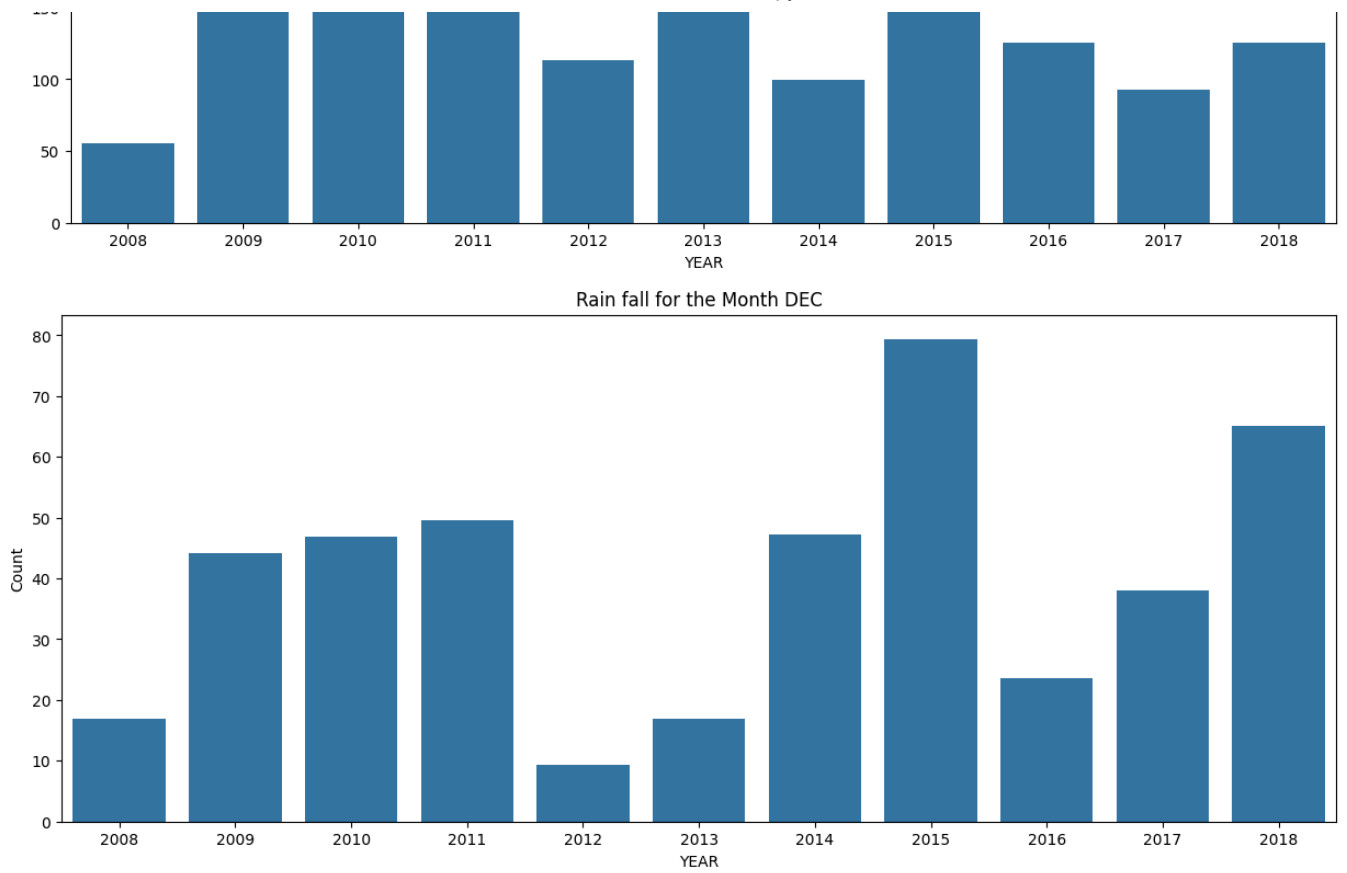


Rain fall for the Month OCT



Rain fall for the Month NOV

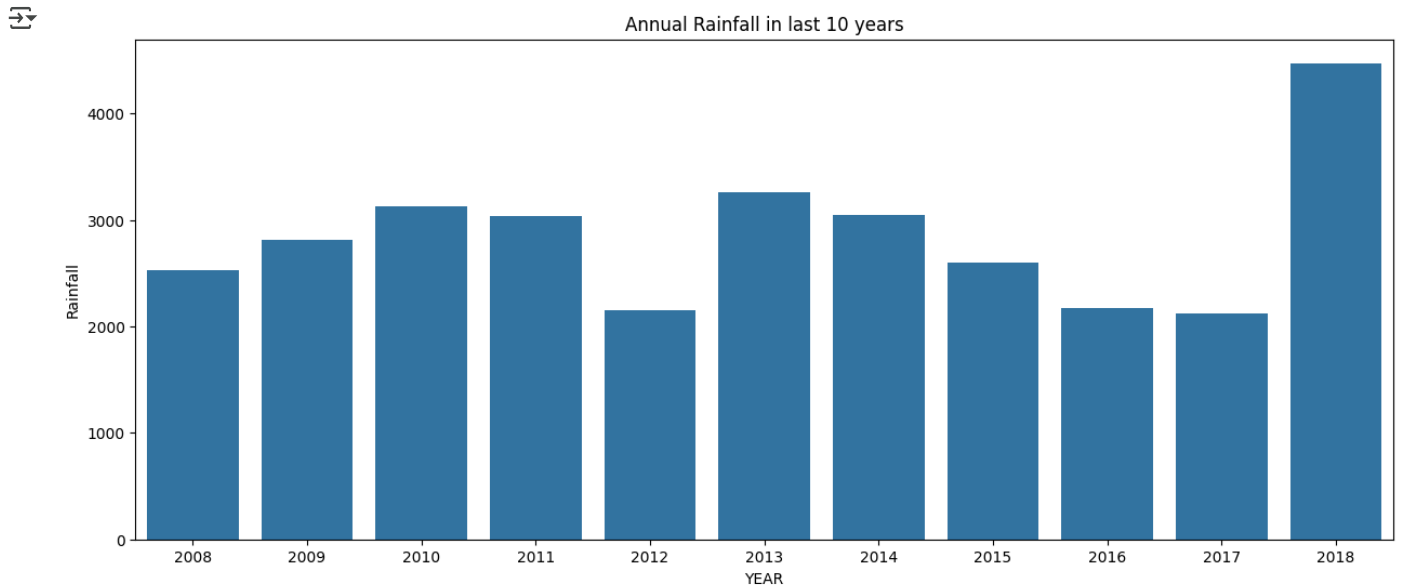




3) What is the Annual Rainfall in Kerala for the last 10 Years

```
def AnnualRainfallfor10years(df_last10):
    plt.figure(figsize=(15, 6))
    sns.barplot(data=df_last10, x='YEAR', y=' ANNUAL RAINFALL')
    plt.xlabel('YEAR')
    plt.ylabel('Rainfall')
    plt.title("Annual Rainfall in last 10 years")
    plt.show()
    plt.close()
```

AnnualRainfallfor10years(df_last10)

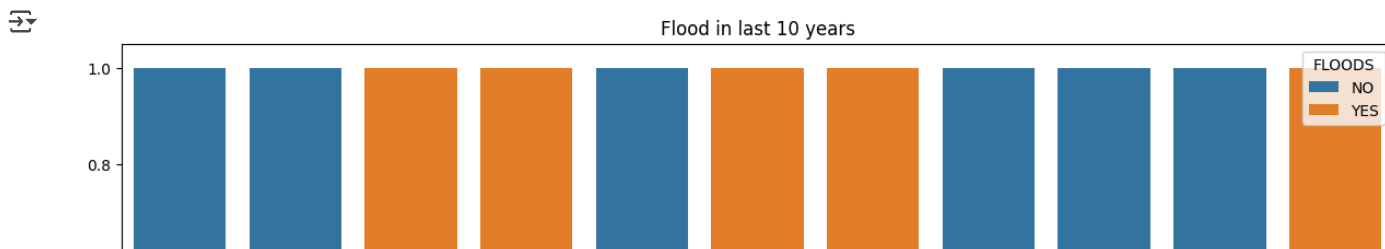


From this we can say that 2018 had the highest ammount of Rain Fall in the last 10 years

✓ 4) Which all years had flood in the last 10 years

```
def Floodinlast10years(df_last10):
    plt.figure(figsize=(15, 6))
    sns.countplot(data=df_last10, x='YEAR', hue='FLOODS')
    plt.xlabel('YEAR')
    plt.ylabel('Flood')
    plt.title("Flood in last 10 years")
    plt.show()
    plt.close()
```

Floodinlast10years(df_last10)



from this we can say that Kerala had floods in the year 2010,11,13,14 and 18.



5) how much rainfall index is considered as a heavy rainfall?



```
warnings.simplefilter(action='ignore', category=Warning)
impactful_columns = ['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC', 'ANNUAL RAINFALL', 'FI
refineddata=data[impactful_columns]
months=['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']
```

```
def mediancalculator(data, months):
    threshold={}
    for i in months:
        threshold[i]=int(data[i].median())

    threshold['ANNUAL RAINFALL']=int(data['ANNUAL RAINFALL'].median())

    return threshold
```

```
def mediandriver(data, months):
    heavyrainfall=refineddata
    threshold=mediancalculator(refineddata, months)
    for key, value in threshold.items():
        heavyrainfall[key] = (heavyrainfall[key] > value).astype(int)
    valuesinmonths=[]
    for i in months:
        counts = heavyrainfall[i].value_counts()
        valuesinmonths.append(counts.rename(i))

    valuesinmonths_df = pd.concat(valuesinmonths, axis=1)
    print(valuesinmonths_df)
    return valuesinmonths_df,heavyrainfall
```

```
valuesinmonths_df,heavyrainfall = mediandriver(refineddata, months)
```

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1	62	62	62	59	59	60	59	59	60	59	59	60
0	56	56	56	59	59	58	59	59	58	59	59	58

Given is the rainfall index for all the months across the years which is considered heavy rainfall

6) Probability of flood given that rainfall in June is greater than the median june rainfall value (threshold for heavy rainfall)

```
heavyrainfall['FLOODS'] = data['FLOODS']

crosstabofjun= pd.crosstab(heavyrainfall['JUN'],
                           heavyrainfall['FLOODS'],
                           margins=True,
                           margins_name='Total')

print(crosstabofjun)
probabilityofjune = round(crosstabofjun.loc[1, 'YES'] / crosstabofjun.loc[1, 'Total'],2)
print(probabilityofjune)
```

FLOODS	NO	YES	Total
JUN			
0	42	16	58
1	16	44	60
Total	58	60	118
0.73			

```
heavyrainfall['FLOODS'] = data['FLOODS']

crosstabofjun= pd.crosstab(heavyrainfall['JUN'],
                           heavyrainfall['FLOODS'],
                           margins=True,
                           margins_name='Total')
print(crosstabofjun)
probabilityofjune = round(crosstabofjun.loc[1, 'YES'] / crosstabofjun.loc[1, 'Total'],2)
print(probabilityofjune)
```

```
↕ FLOODS  NO  YES  Total
JUN
0         42  16    58
1         16  44    60
Total    58  60   118
0.73
```

we can say that the Probability of having flood given that heavy rain fall has occurred is 0.74

7) Given that there is a flooding, calculate the probability that heavy rainfall has occurred in July (more than threshold value)?

```
crosstabofjul= pd.crosstab(heavyrainfall['JUL'],
                           heavyrainfall['FLOODS'],
                           margins=True,
                           margins_name='Total')
print(crosstabofjul)
probabilityofjuly = round(crosstabofjul.loc[1, 'YES'] / crosstabofjul.loc['Total', 'YES'],2)
print(probabilityofjune)
```

```
↕ FLOODS  NO  YES  Total
JUL
0         38  21    59
1         20  39    59
Total    58  60   118
0.73
```

The probability of rain fall given there is flooding is 0.73

8)probability of flood given that june and july rainfall was greater than their median rainfall value?

```
crosstabofjunjul=pd.crosstab(index = [heavyrainfall['JUN'], heavyrainfall['JUL']],
                              columns = heavyrainfall['FLOODS'],
                              margins=True,
                              margins_name='Total')

print(crosstabofjunjul)

probabilityofjunejuly=crosstabofjunjul.loc[1, 1]['YES']/crosstabofjunjul.loc['Total']['YES']
print(probabilityofjunejuly)
```

```
↕ FLOODS      NO  YES  Total
JUN  JUL
0     0     25   4     29
     1     17  12     29
1     0     13  17     30
     1      3  27     30
Total      58  60    118
JUL
0.45
Name: YES, dtype: float64
```

The Probability of flood givenb that huyne and july there is a rainfall is 0.45

Double-click (or enter) to edit

