Abhishek Murthy
21BDS0064
Fall Sem 2024-2025
DA -1
Data Mining Lab
30-07-2024

In [ ]:

```
#21BDS0064 Abhishek Murthy
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import math
import statistics
```

In [ ]:

```
#21BDS0064
df = pd.read_csv('temperatures.csv')
```

In [ ]:

```
#21BDS0064
df.head()
```

Out[ ]:

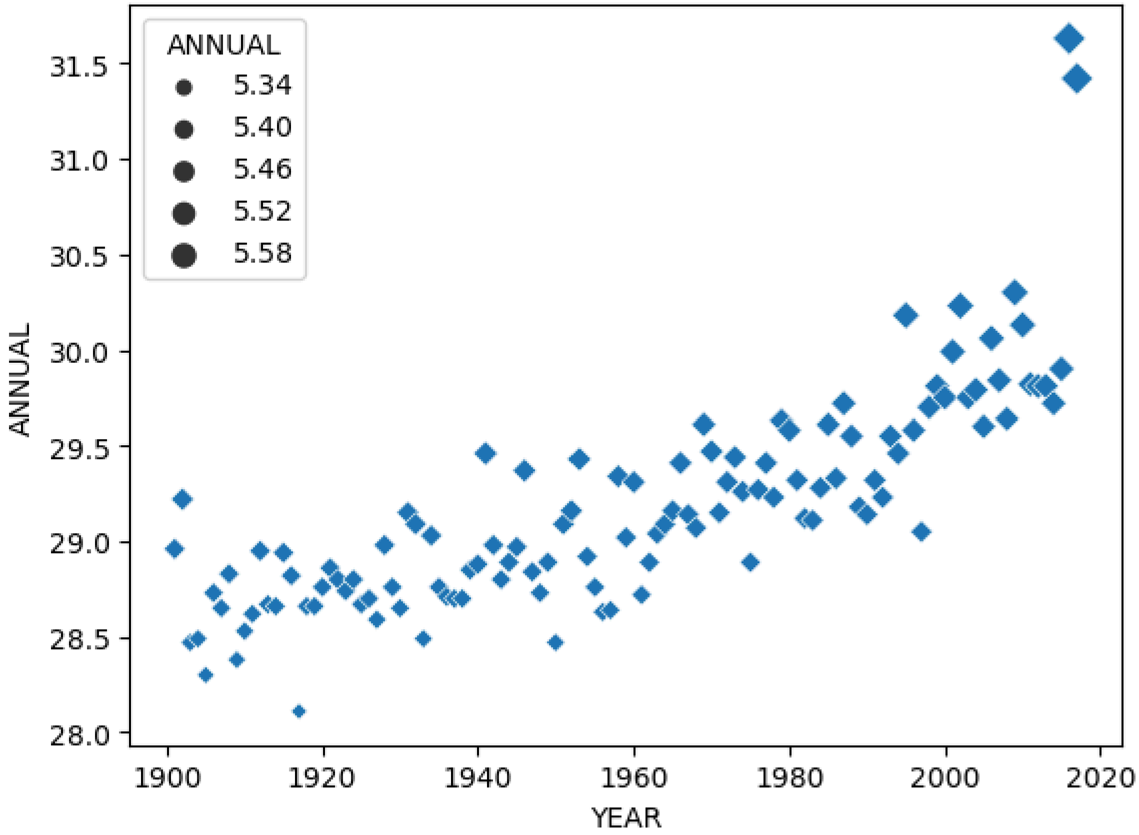
	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.49	28.96
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.04	29.22
2	1903	23.44	25.03	27.83	31.39	32.91	33.00	31.34	29.98	29.85	29.04	26.08	23.65	28.47
3	1904	22.50	24.73	28.21	32.02	32.64	32.07	30.36	30.09	30.04	29.20	26.36	23.63	28.49
4	1905	22.00	22.83	26.68	30.01	33.32	33.25	31.44	30.68	30.12	30.67	27.52	23.82	28.30

In [ ]:

```
# 21BDS0064
# Plot a scatter plot where (with x axis and y axis labels) X-axis should be the YEA
# Change the marker to a diamond and size as the square root of the ANNUAL temperatu
sns.scatterplot(data = df, x = df['YEAR'], y = df['ANNUAL'], marker = 'D', size=df['
```

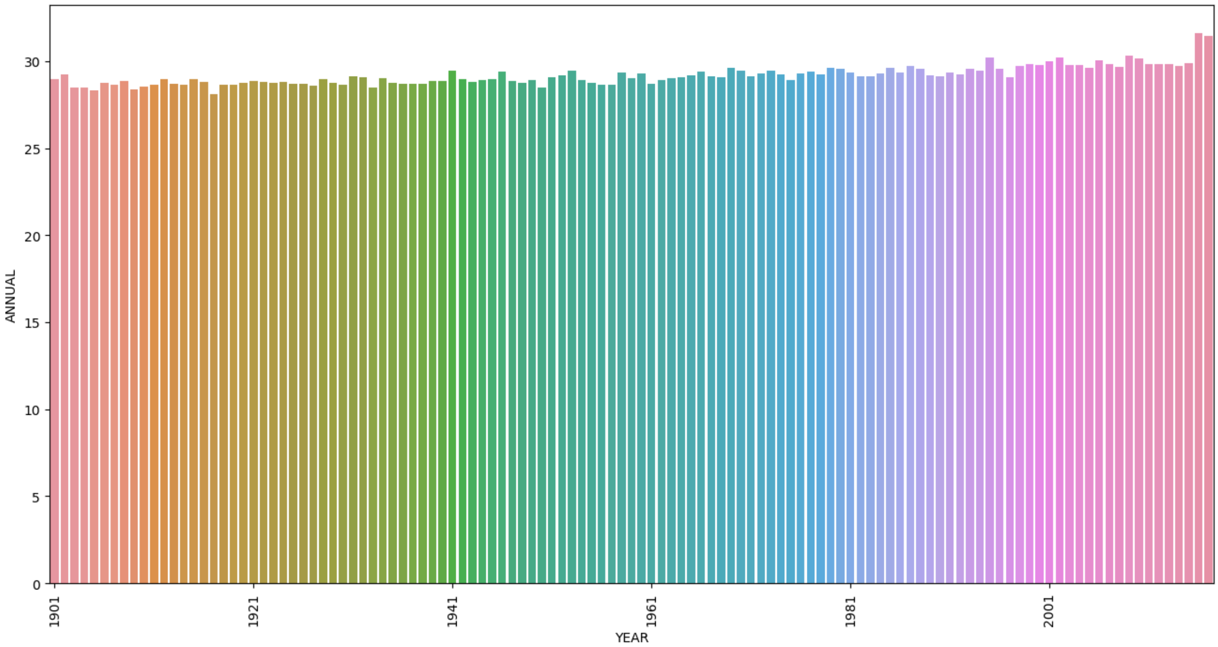
Out[ ]:

```
<Axes: xlabel='YEAR', ylabel='ANNUAL'>
```



In [ ]:

```
# 21BDS0064
grouped_df = df.groupby('YEAR').mean().reset_index()
plt.figure(figsize=(16, 8))
sns.barplot(data=grouped_df, x='YEAR', y='ANNUAL')
xticks = grouped_df['YEAR'][::20]  # Select every 20th year
plt.xticks(ticks=xticks.index, labels=xticks, rotation=90)
plt.show()
```



In [ ]:

```
# 21BDS0064
#  c) Calculate the range (maximum - minimum) of temperatures for each month across
df.loc[:, 'JAN':"DEC"].max() - df.loc[:, 'JAN':"DEC"].min()
```

Out[ ]:

```
JAN    4.94
FEB    6.89
MAR    5.94
APR    5.37
MAY    3.91
JUN    3.38
JUL    3.00
AUG    2.53
SEP    3.15
OCT    4.39
NOV    4.41
DEC    4.99
dtype: float64
```

In [ ]:

```
# 21BDS0064
# d) Compute the standard deviation and variance for the temperatures of each month
print("Standard Deviation: ", df.loc[:, 'JAN':"DEC"].std())
print("Variance: ", df.loc[:, 'JAN':"DEC"].var())
```

```
Standard Deviation:  JAN    0.834588
FEB    1.150757
MAR    1.068451
APR    0.889478
MAY    0.724905
JUN    0.633132
JUL    0.468818
AUG    0.476312
SEP    0.544295
OCT    0.705492
NOV    0.714518
DEC    0.782644
dtype: float64
Variance:  JAN    0.696536
FEB    1.324241
MAR    1.141588
APR    0.791171
MAY    0.525487
JUN    0.400856
JUL    0.219790
AUG    0.226873
SEP    0.296257
OCT    0.497719
NOV    0.510535
DEC    0.612532
dtype: float64
```

In [ ]:

```
# 21BDS0064
# e) Display the median of the ANNUAL temperatures for the years 2010 to 2020
filtered_df = df[(df['YEAR'] >= 2010) & (df['YEAR'] <= 2020)]
median_annual_temp = filtered_df['ANNUAL'].median()
print("Median annual temperature: ",median_annual_temp)
```

```
Median annual temperature:  29.86
```

In [ ]:

```
# 21BDS0064
# f) Display the YEAR-wise average temperature for each season (JAN-FEB, MAR-MAY, JU
print(df[['YEAR', 'JAN-FEB', 'MAR-MAY', 'JUN-SEP', 'OCT-DEC']])
```

```
     YEAR  JAN-FEB  MAR-MAY  JUN-SEP  OCT-DEC
0    1901    23.27    31.46    31.27    27.25
1    1902    25.75    31.76    31.09    26.49
2    1903    24.24    30.71    30.92    26.26
```

```
3    1904    23.62    30.95    30.66    26.40
4    1905    22.25    30.00    31.33    26.57
..    ...      ...      ...      ...      ...
112  2013    25.58    32.58    31.33    27.83
113  2014    24.90    31.82    32.00    27.81
114  2015    25.74    31.68    31.87    28.27
115  2016    28.33    34.57    32.28    30.03
116  2017    27.95    34.13    32.41    29.69

[117 rows x 5 columns]
```

In [ ]:

```
# 21BDS0064
# g) Count the number of years with an ANNUAL temperature above a certain threshold
print("Number of years with annual temperature greater than 25°C: ", len(df[df['ANNU
```

```
Number of years with annual temperature greater than 25°C:  117
```

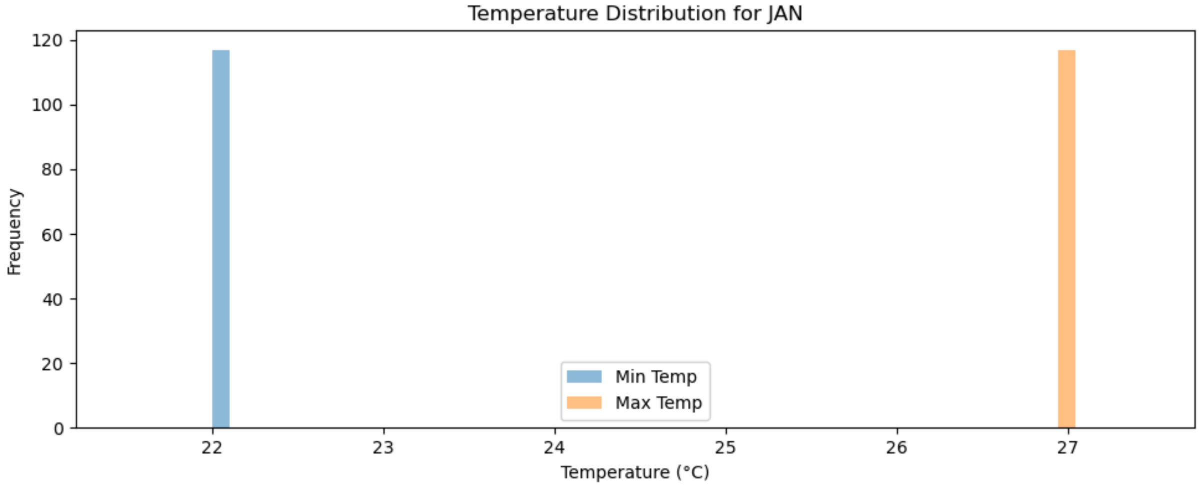
In [ ]:

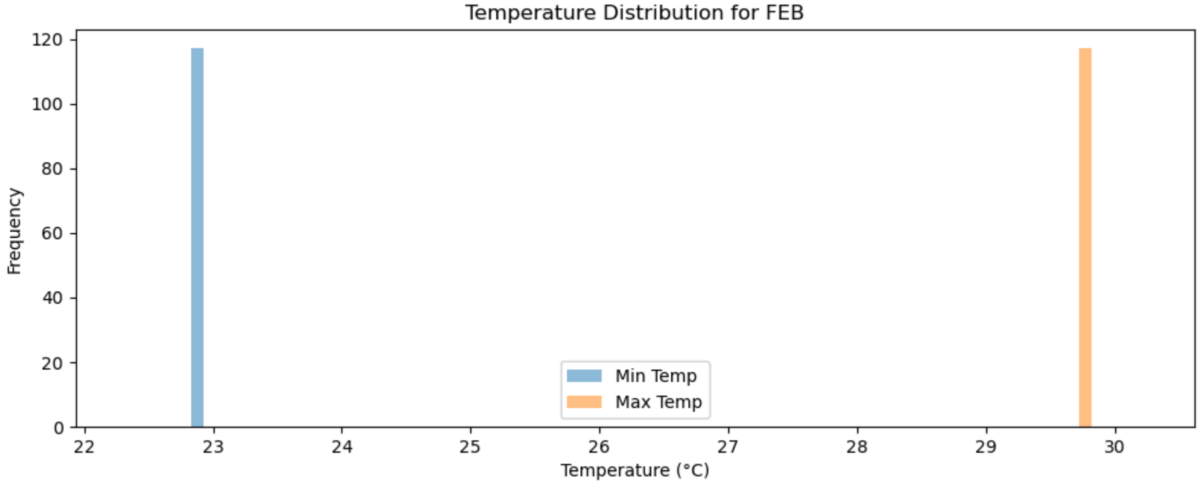
```
# 21BDS0064
# h) Print the mode temperature for the month of JULY
print("Mode temperature for the month of July : ",statistics.mode(df['JUL']))
```

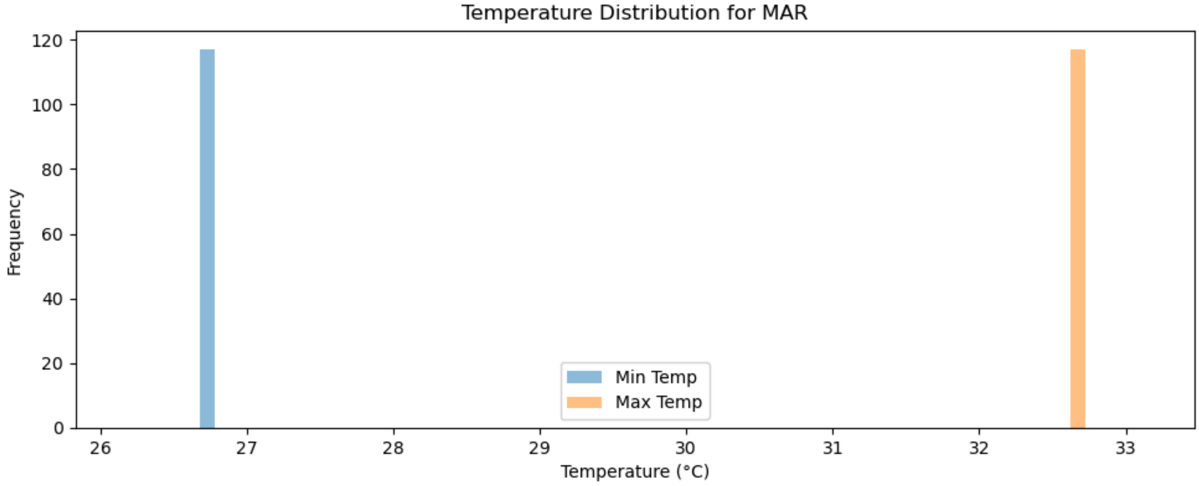
```
Mode temperature for the month of July :  30.9
```

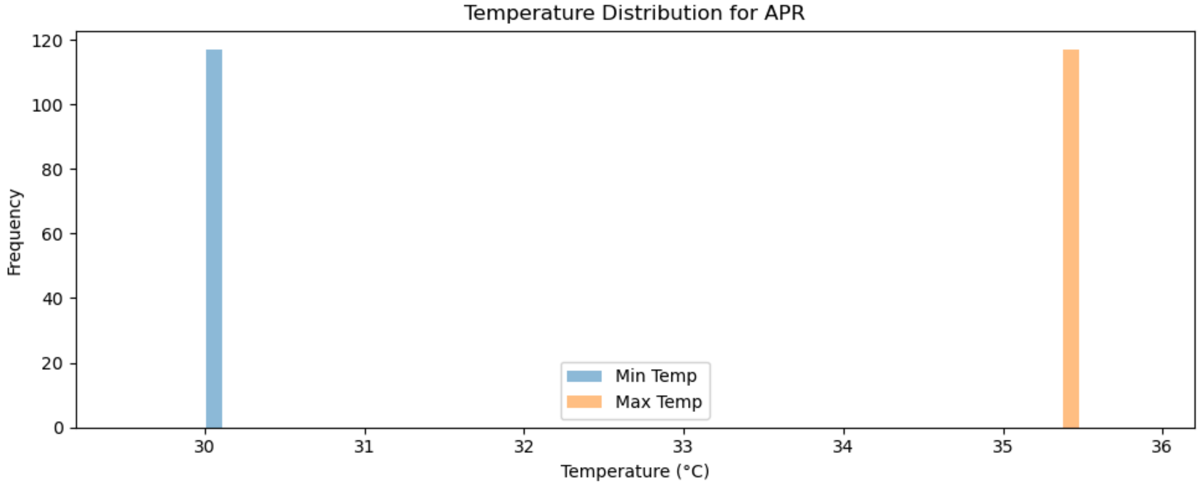
In [ ]:

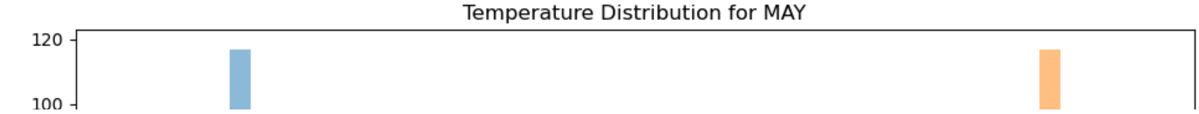
```
# 21BDS0064
temperatures = []

for month in df.columns[1:13]:
    minimum_temp = df[month].min()
    maximum_temp = df[month].max()
    for year in df['YEAR']:
        temperatures.append([year, month, minimum_temp, maximum_temp])

final_df = pd.DataFrame(temperatures, columns=['YEAR', 'Month', 'Min_Temp', 'Max_Tem

# Plot histograms for each month
months = df.columns[1:13]
fig, axes = plt.subplots(len(months), 1, figsize=(10, len(months) * 4))

for i, month in enumerate(months):
    ax = axes[i]
    minimum_temp_data = final_df[final_df['Month'] == month]['Min_Temp']
    maximum_temp_data = final_df[final_df['Month'] == month]['Max_Temp']
    ax.hist(minimum_temp_data, alpha=0.5, label='Min Temp', bins=10)
    ax.hist(maximum_temp_data, alpha=0.5, label='Max Temp', bins=10)
    ax.set_title(f'Temperature Distribution for {month}')
    ax.set_xlabel('Temperature (°C)')
    ax.set_ylabel('Frequency')
    ax.legend()

plt.tight_layout()
plt.show()
```

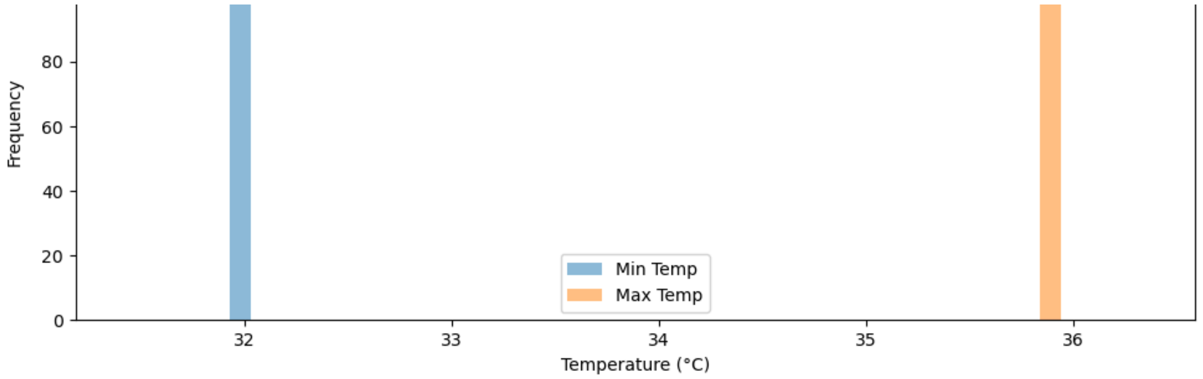

Temperature Distribution for JAN
Frequency
0
20
40
60
80
100
120
22
23
24
25
26
27
Temperature (°C)
Min Temp
Max Temp

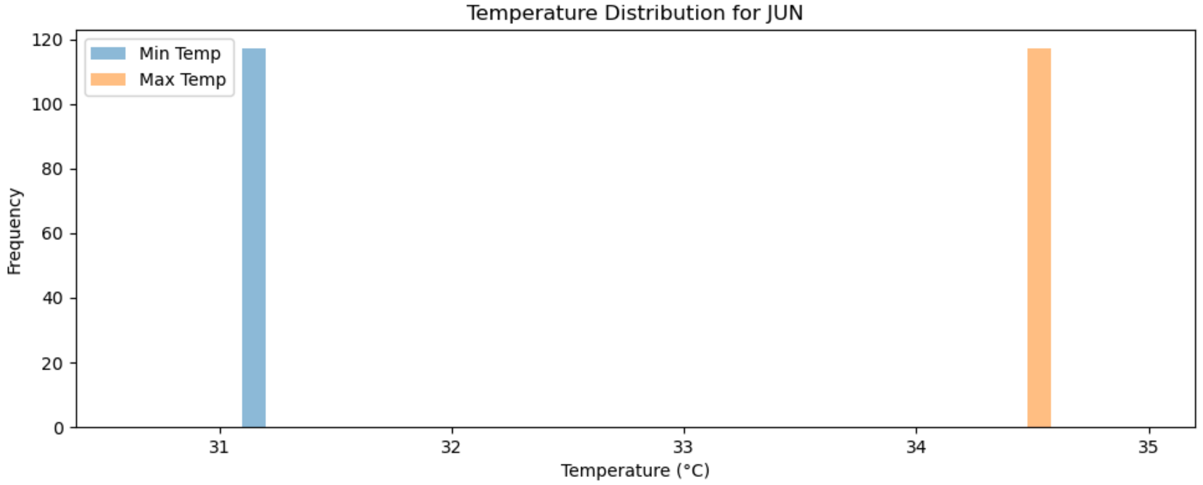

Temperature Distribution for FEB
Frequency
0
20
40
60
80
100
120
22
23
24
25
26
27
28
29
30
Temperature (°C)
Min Temp
Max Temp

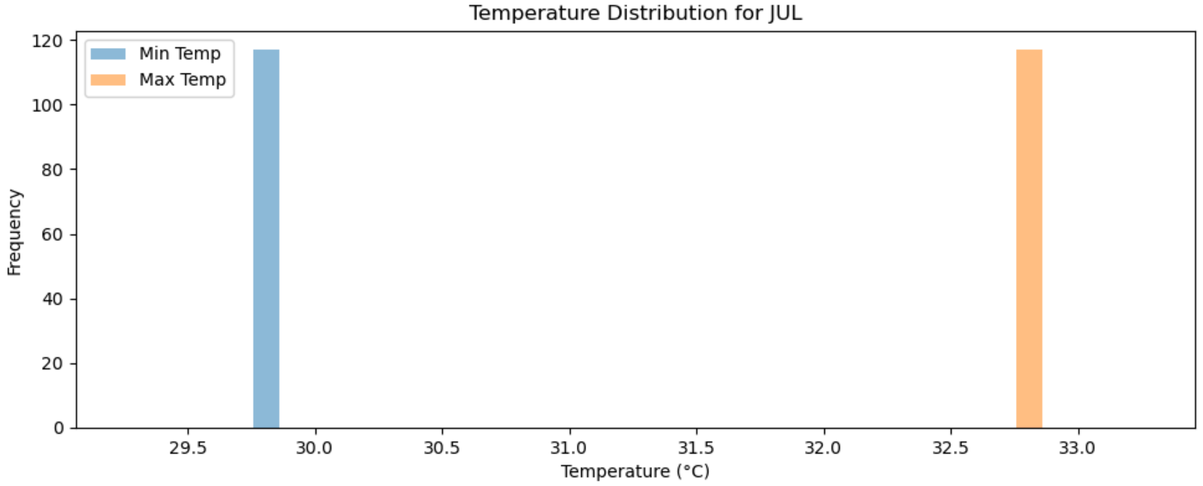

Temperature Distribution for MAR
Frequency
0
20
40
60
80
100
120
26
27
28
29
30
31
32
33
Temperature (°C)
Min Temp
Max Temp

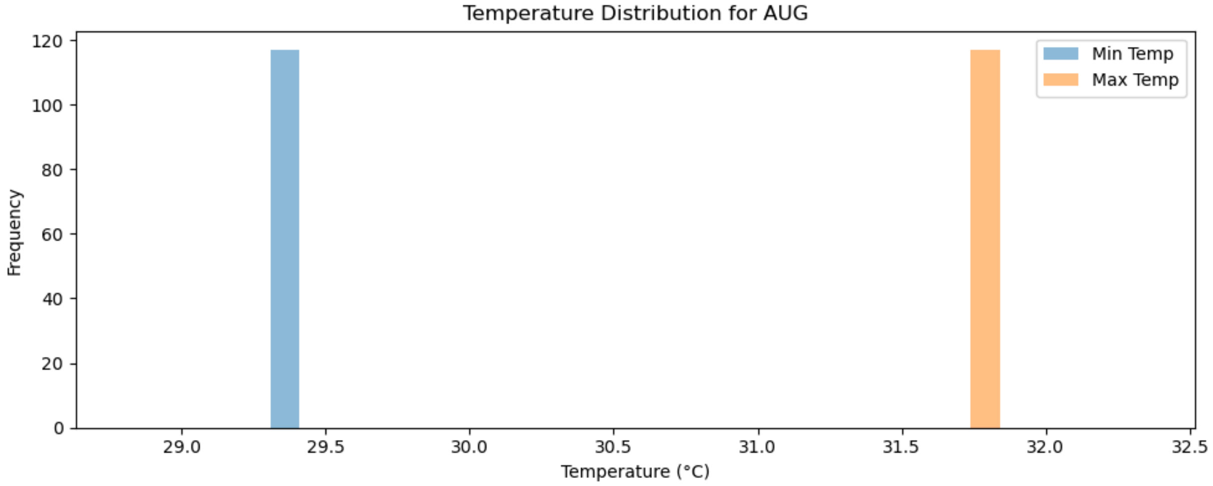

Temperature Distribution for APR
Frequency
0
20
40
60
80
100
120
30
31
32
33
34
35
36
Temperature (°C)
Min Temp
Max Temp

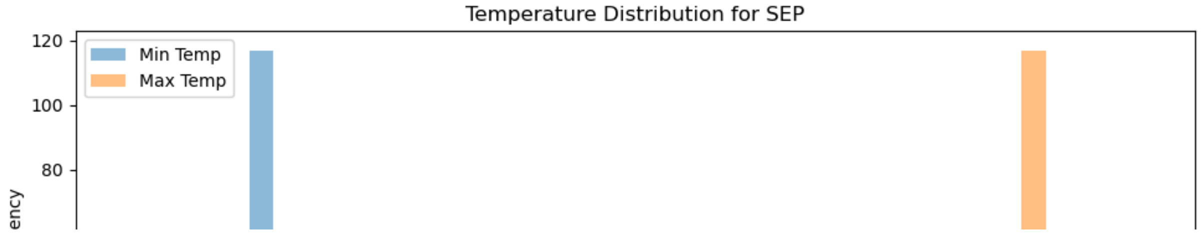

Temperature Distribution for MAY
100
120




Temperature Distribution for JUN

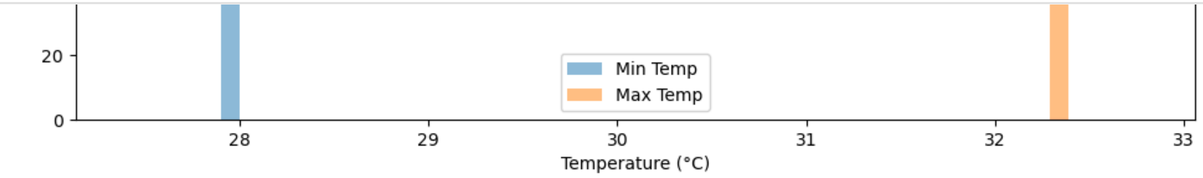

Temperature Distribution for JUL

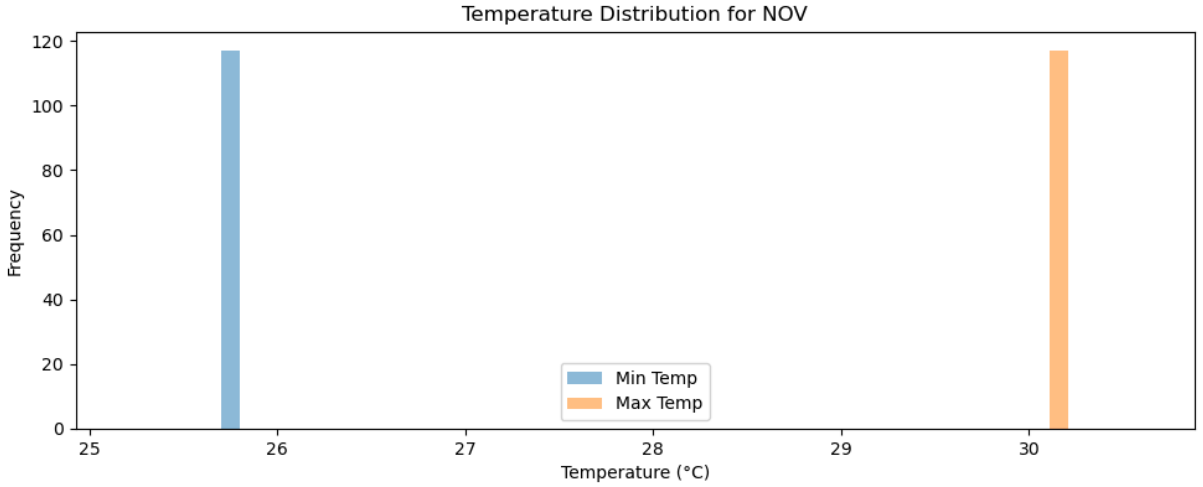

Temperature Distribution for AUG

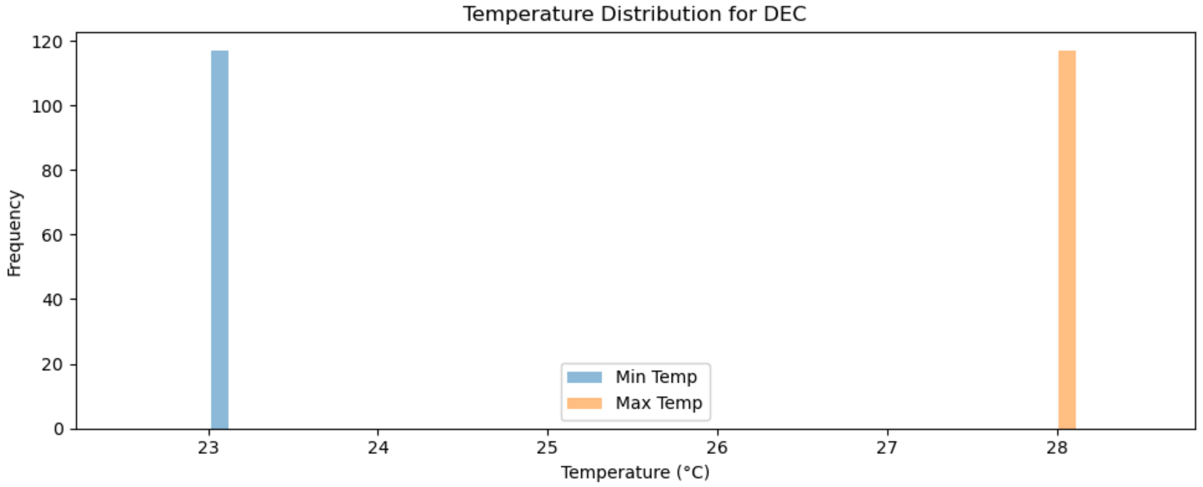

Temperature Distribution for SEP

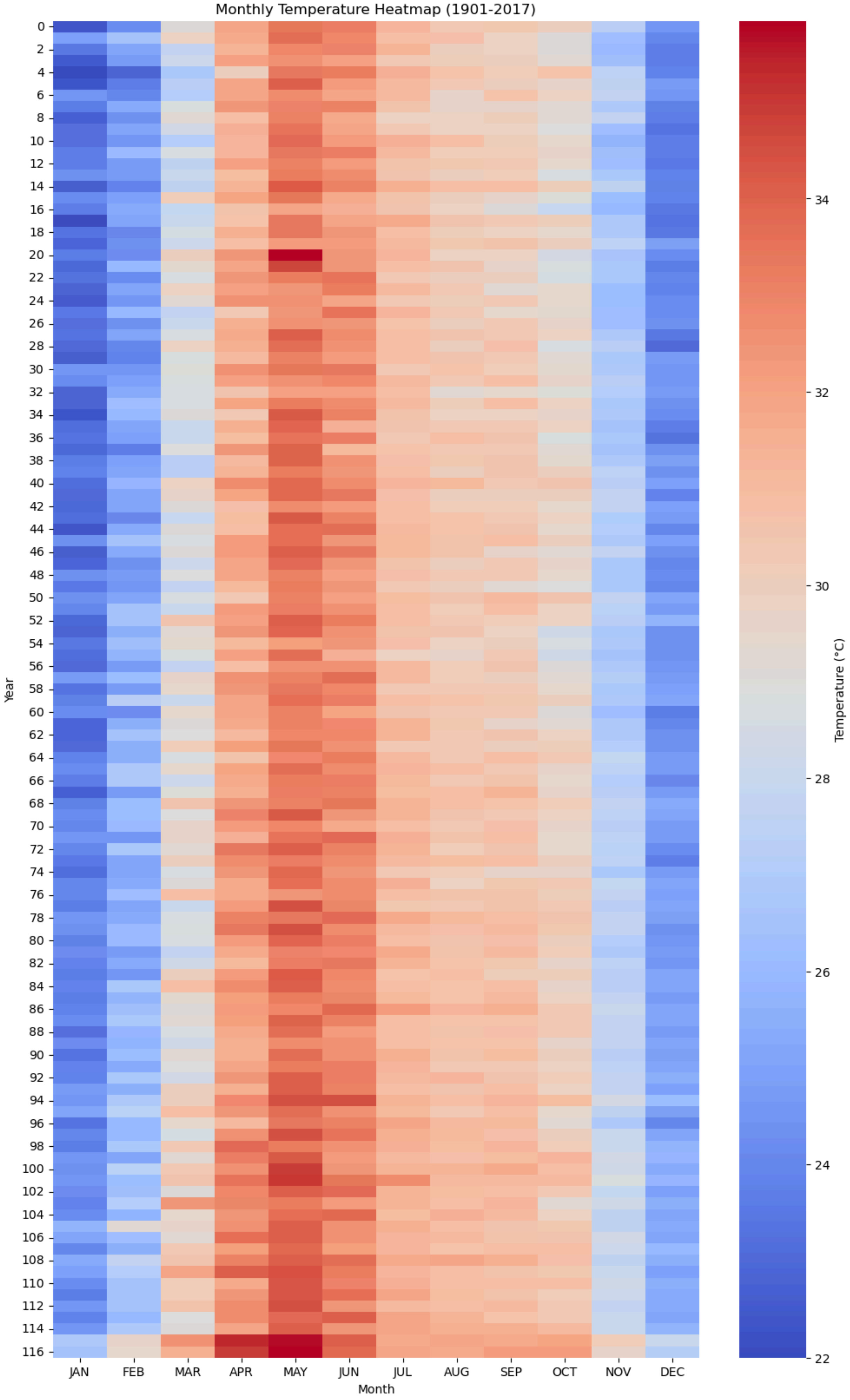
In [ ]:

```
# 21BDS0064
# Plot a heatmap to visualize the temperature data: X-axis should represent the mont
# Y-axis should represent the years. The heatmap should display the temperature valu
# Use different colors to represent different temperature range
months = ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV
df_months = df[months]

plt.figure(figsize=(12, 20))
sns.heatmap(df_months, cmap='coolwarm', cbar_kws={'label': 'Temperature (°C)'})

plt.title('Monthly Temperature Heatmap (1901-2017)')
plt.xlabel('Month')
plt.ylabel('Year')

plt.show()
```








Monthly Temperature Heatmap (1901-2017)
Year
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
98
100
102
104
106
108
110
112
114
116
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
Month
Temperature (°C)
34
32
30
28
26
24
22