基于第二次世界大战天气状况的探索性数据分析与预测

任务目的

对二战中的空袭行动与天气状况数据进行分析，基于第二次世界大战中的空中轰炸行动和天气状况的多种数据源，利用 EDA (Exploratory Data Analysis)分析数据并对数据进行清理，根据时间序列预测方式，预测出轰炸目标城市、天气状况以及行动何时完成等。

任务内容

下载数据集
数据清洗
数据可视化

任务原理

Plotly 库：Plotly 图形库可以在线生成交互式的、出版物质量的图形。以及制作折线图、散点图、面积图、条形图、误差线、方框图、直方图、热图、子图、多轴、极坐标图和气泡图等数据分析图形。

折线图：

导入 graph_objs 库

- graph_objs.Scatter(x=x 轴,
    - y=y 轴,
    - mode=绘制标记类型，如标记、直线或线+标记,
    - name=绘图名称,
    - marker=定义标记的颜色形状等(color=线条的颜色),
    - text=悬停文本,
    -)

iplot():绘制由数据和布局创建的图形（图）

条形图：

- graph_objs.Bar(x=x 轴,
    - y=y 轴,
    - mode=绘制标记类型，如标记、直线或线+标记,
    - name=绘图名称,
    - marker=定义标记的颜色形状等(color=线条的颜色,line=bars 之间的线),
    - text=悬停文本,
    -)

iplot():绘制由数据和布局创建的图形（图）

饼图：

- fig = 创建图形{
  "data": 绘图类型[{
    - "values": 绘图的值,
    - "labels": 绘图标签,
    - "name":图形名称,
    - "hoverinfo":悬停信息,
    - "hole": 孔宽度,

      "type": 饼图类型

  },],
  "layout":绘图布局 {

      "title":布局标题,

      "annotations": 注释[

        { "font": 字体,

          "showarrow":显示箭头,

          "text": 文本,

            "x":x 轴,

```
            "y": y 轴

        },

    ]
```

}}
iplot(fig):绘制图形

气泡图：

- data = [{
    - 'y': y 轴,
    - 'x': x 轴,
    - 'mode': 标记,
    - 'marker':标记属性 {
    - 'color':绘图的第三维,
    - 'size':绘图的第四维,
    - 'showscale': True
    - },

      "text" :名称

}]
iplot(data):绘制图形

直方图：

导入 graph_objs 库

- graph_objs.Histogram(x=x 轴, y=y 轴， opacity=直方图的不透明度, name = 图例名称, marker=定义标记的颜色形状等(color=颜色)
- layout = graph_objs.Layout(barmode=直方图状覆盖模式)

fig = graph_objs.Figure(data=data, layout=layout)

iplot(fig):绘制图形

任务步骤

```
import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import chart_studio.plotly as py

from plotly.offline import init_notebook_mode, iplot

init_notebook_mode(connected=True)

import plotly.graph_objs as go
```

In [ ]:

下载数据集

1、数据集。

- 二战中的空中轰炸行动（Aerial Bombing Operations in WW2）

这一数据包括轰炸行动。例如，1945 年，美国使用庞特奥利沃机场炸弹德国（柏林）和 A36 飞机。

- 二战期间的天气状况（Wether Conditions in WW2）

二战期间的天气状况。例如，根据乔治镇气象站，1942 年 1 月 7 日的平均气温是 23.88 度。
该数据集中有 2 个子集：第一个包括气象站的位置，如国家、纬度和经度。 第二个包括气象站的测量最低、最高和平均温度。

```
# 轰炸数据

aerial = pd.read_csv("../../data/NationalUniversityofDefenseTechnology/operations.csv")

# 第一个天气数据，气象站的位置

weather_station_location = pd.read_csv("../../data/NationalUniversityofDefenseTechnology/Weather Station Locations.csv")
```

```
# 第二个天气数据，气象站的测量最低、最高和平均温度

weather = pd.read_csv("../../data/NationalUniversityofDefe
nseTechnology/Summary of Weather.csv")
```

数据清洗

2、数据清洗。

（1）空袭数据包含了大量的 NaN 值。项目中没有使用它们，而是删除了一些 NaN 值。它不仅消除了不确定性，而且是一个可视化的过程。

- 删除值为 NaN 的国家
- 如果目标经度为 NaN，则删除
- 起飞经度为 NaN 时删除
- 删除未使用的特征

（2）天气状况数据不需要任何清理。通过对勘探数据的分析和可视化，选择一定的地点进行深入研究。只放入使用的数据变量。

3、输出信息。

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2555 entries, 0 to 178080
Data columns (total 17 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Mission Date           2555 non-null   object
 1   Theater of Operations  2555 non-null   object
 2   Country                2555 non-null   object
 3   Air Force              2505 non-null   object
 4   Aircraft Series        2528 non-null   object
 5   Callsign               10 non-null     object
 6   Takeoff Base           2555 non-null   object
 7   Takeoff Location       2555 non-null   object
 8   Takeoff Latitude       2555 non-null   object
 9   Takeoff Longitude      2555 non-null   float64
 10  Target Country         2499 non-null   object
 11  Target City            2552 non-null   object
 12  Target Type            602 non-null    object
 13  Target Industry        81 non-null     object
 14  Target Priority        230 non-null    object
 15  Target Latitude        2555 non-null   float64
 16  Target Longitude       2555 non-null   float64
dtypes: float64(3), object(14)
memory usage: 359.3+ KB
```

```
#使用到的数据
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 161 entries, 0 to 160
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   WBAN              161 non-null    int64  
 1   NAME              161 non-null    object 
 2   STATE/COUNTRY ID  161 non-null    object 
 3   Latitude          161 non-null    float64
 4   Longitude         161 non-null    float64
dtypes: float64(2), int64(1), object(2)
memory usage: 6.4+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119040 entries, 0 to 119039
Data columns (total 3 columns):
 #   Column    Non-Null Count   Dtype  
---  ------    --------------   -----  
 0   STA       119040 non-null  int64  
 1   Date      119040 non-null  object 
 2   MeanTemp  119040 non-null  float64
dtypes: float64(1), int64(1), object(1)
memory usage: 2.7+ MB
```

数据可视化

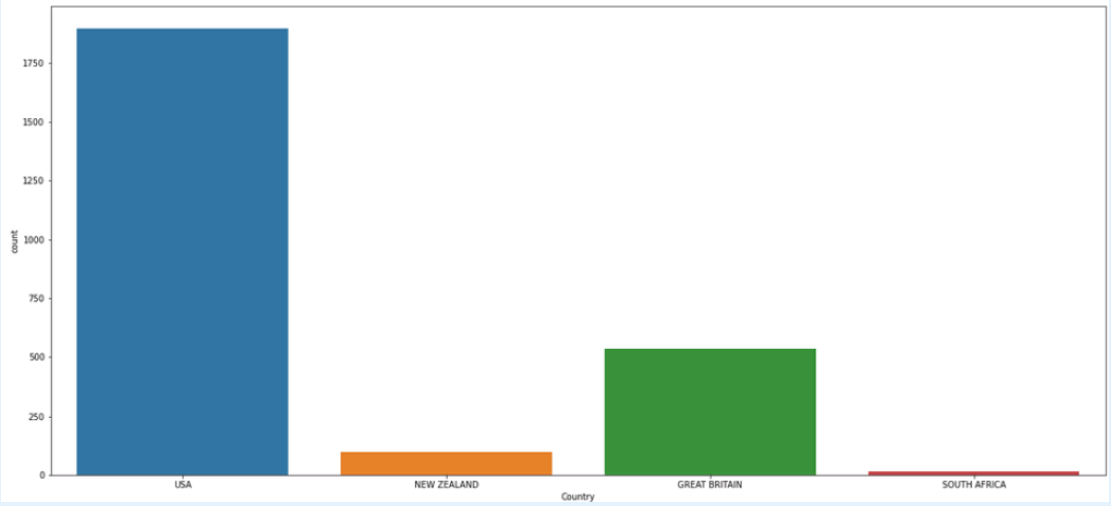
4、数据可视化。

分为以下几部分：

- 袭击了多少个国家
- 主要目标国家
- 十大飞机系列
- 起飞基地位置（攻击国家）
- 目标位置
- 轰炸路径
- 战区
- 气象站位置

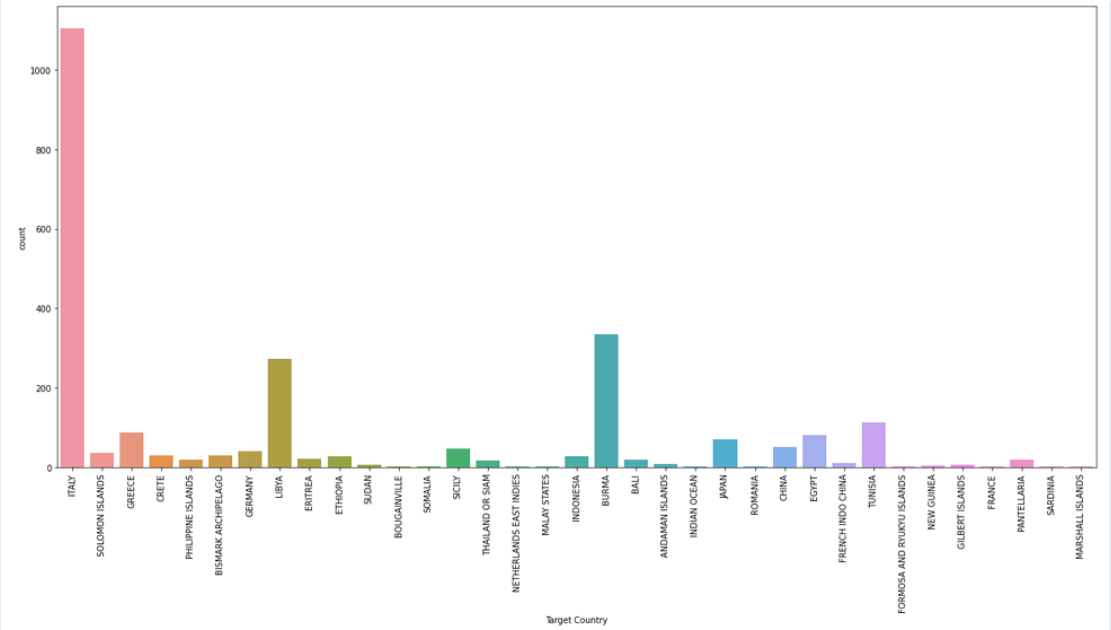
#国家

```
USA              1895
GREAT BRITAIN     544
NEW ZEALAND       102
SOUTH AFRICA       14
Name: Country, dtype: int64
```



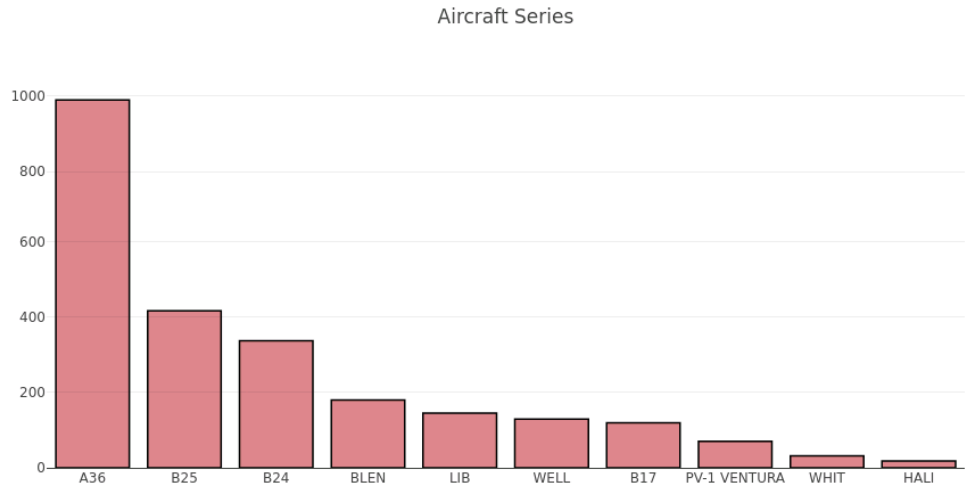

```
#目标城市
```

```
ITALY      1104
BURMA       335
LIBYA       272
TUNISIA     113
GREECE       87
EGYPT        80
JAPAN        71
CHINA        52
SICILY       46
GERMANY      41
Name: Target Country, dtype: int64
```

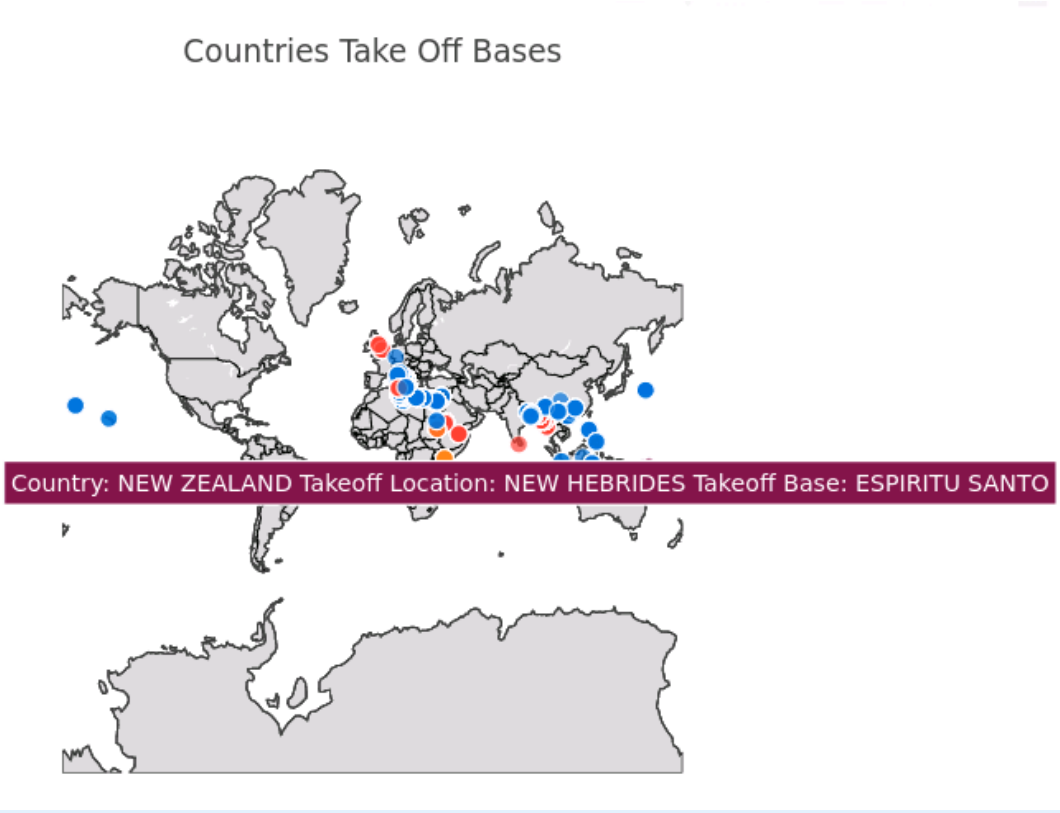


 # 飞机系列

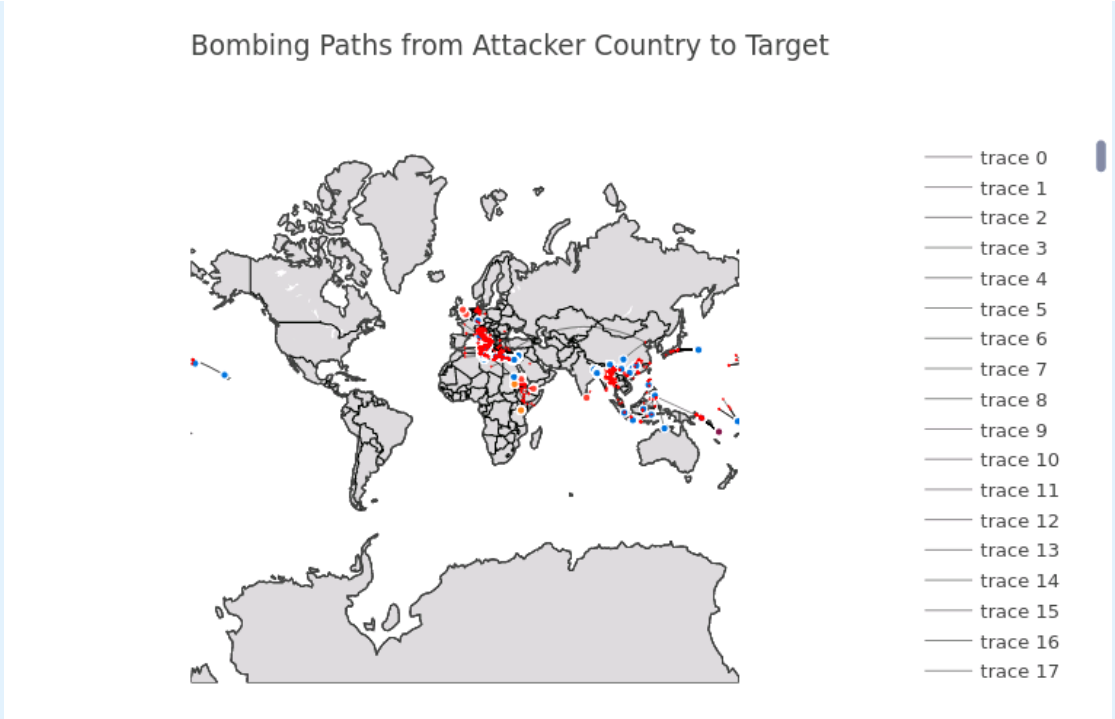
```
A36             990
B25             416
B24             337
BLEN            180
LIB             145
WELL            129
B17             119
PV-1 VENTURA     70
WHIT             32
HALI             18
Name: Aircraft Series, dtype: int64
```



 # 目标位置



```
# 轰炸路径
```

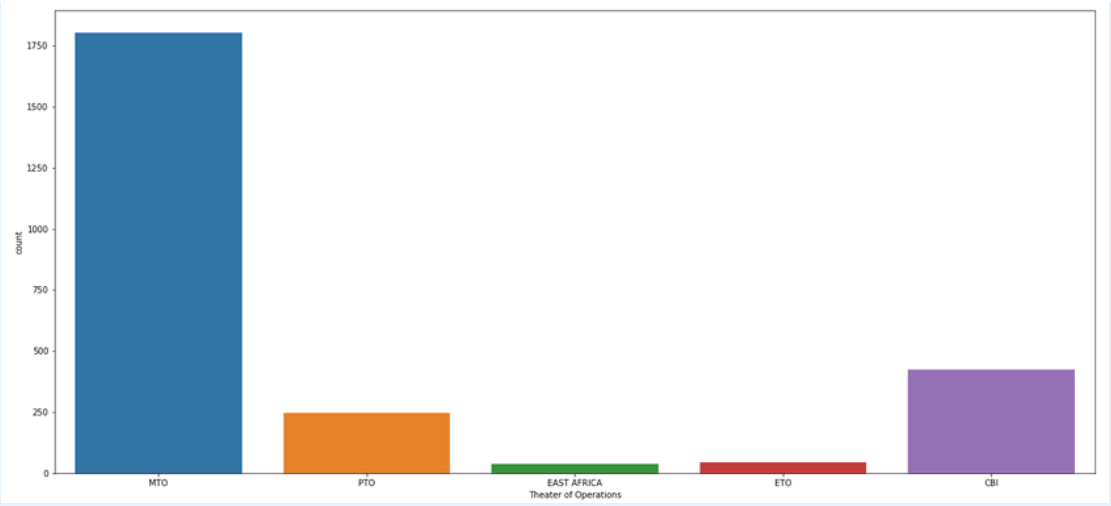


从轰炸路径可以看出，大部分轰炸攻击是在地中海战区进行的。战区：

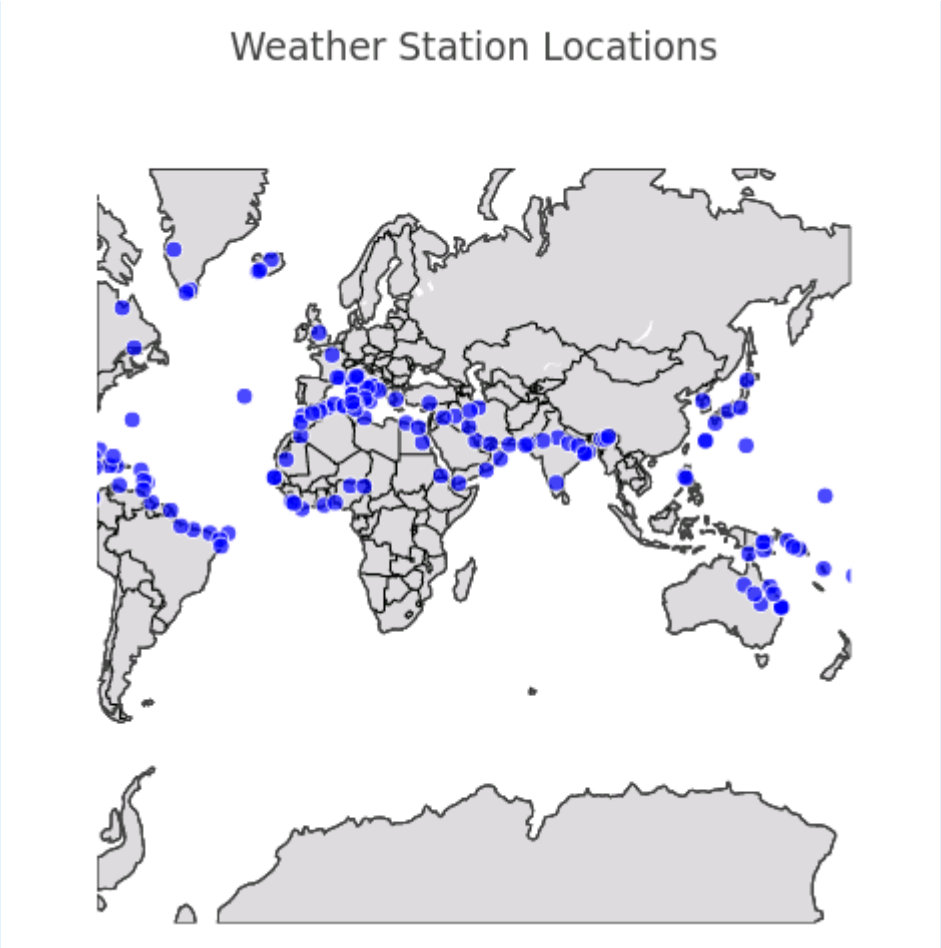
- ETO: European Theater of Operations 欧洲战区
- PTO: Pasific Theater of Operations 太平洋战区
- MTO: Mediterranean Theater of Operations 地中海战区
- MTO: Mediterranean Theater of Operations 中缅印战区
- EAST AFRICA: East Africa Theater of Operations 东非战区

```
#战区
```

```
MTO             1802
CBI              425
PTO              247
ETO               44
EAST AFRICA       37
Name: Theater of Operations, dtype: int64
```

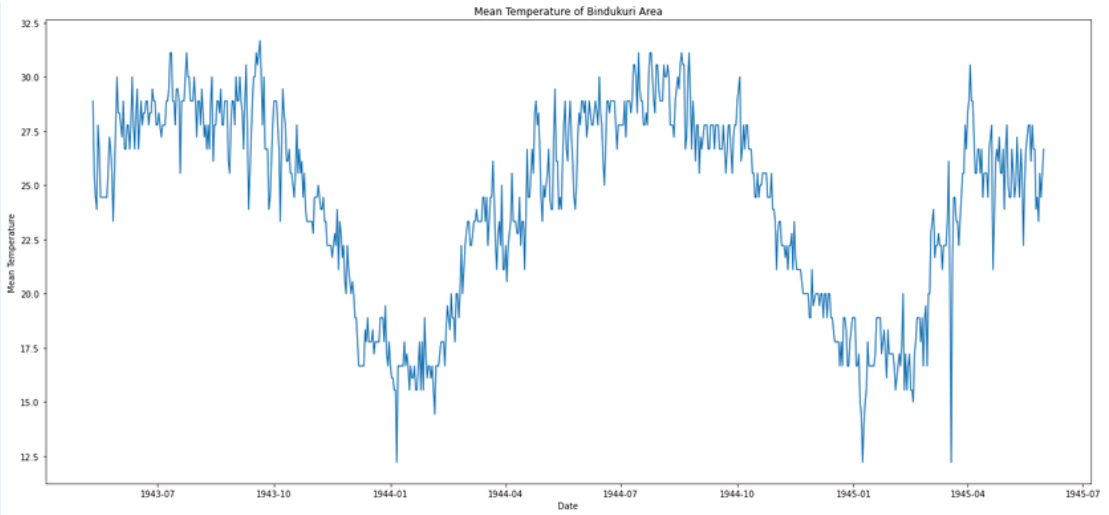


#气象站位置



聚焦美缅战争:

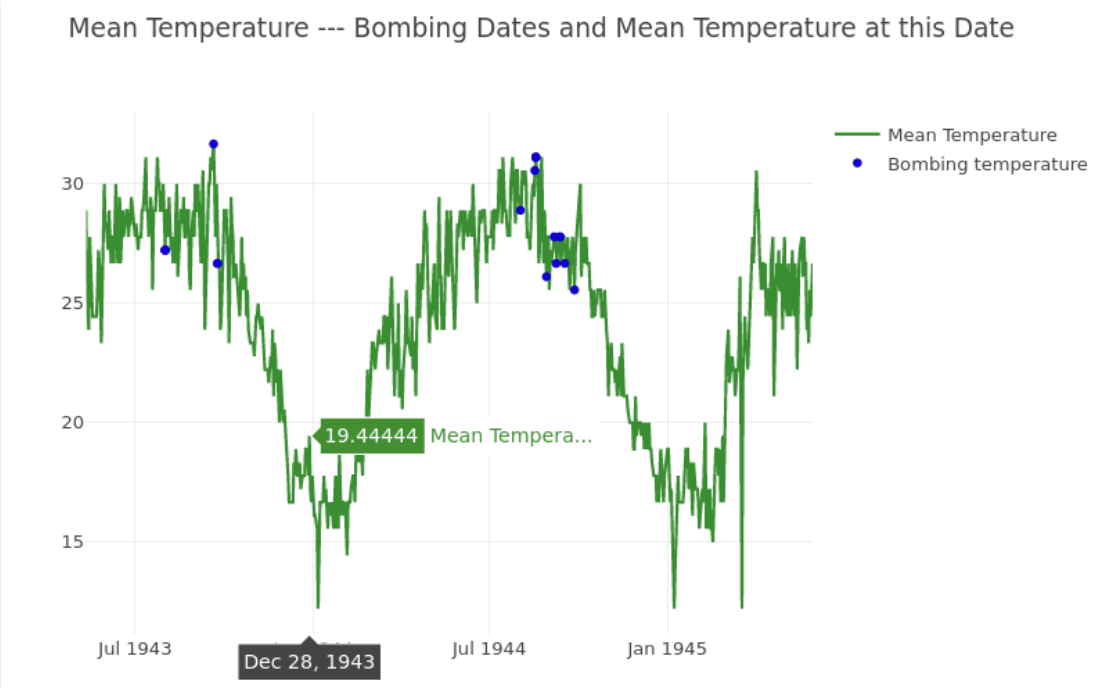
- 在这场战争中，美国从 1942 年到 1945 年轰炸了缅甸（卡萨市）。
- 距离这场战争最近的气象站是宾杜库里，它有 1943 年至 1945 年的气温记录。



从 1943 年到 1945 年进行了温度测量。

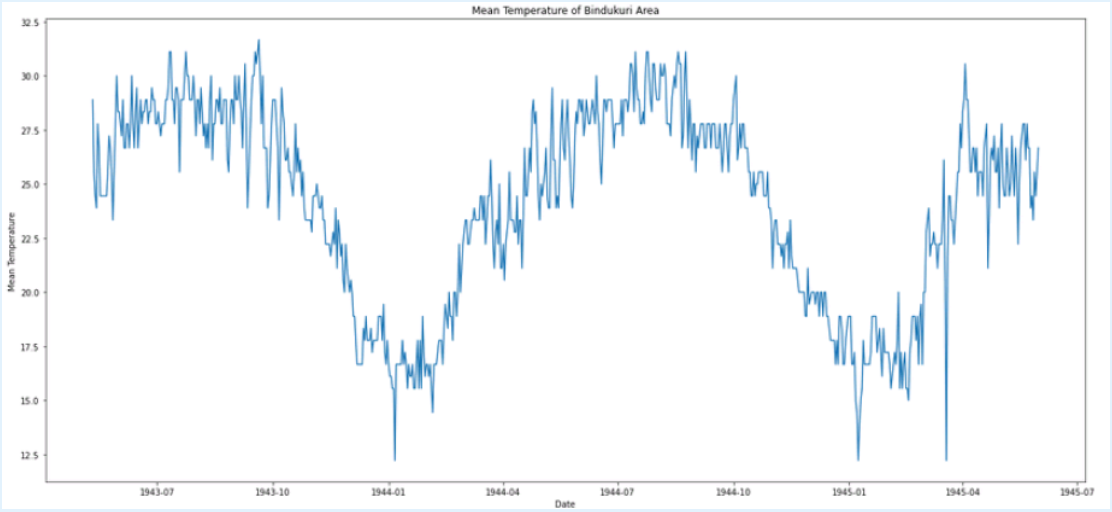
- 温度在 12 到 32 度之间。
- 冬季的温度比夏季的温度低。

```
aerial = pd.read_csv("../../data/NationalUniversityofDefenseTechnology/operations.csv")
```



基于 ARIMA 的时间序列预测

5、时间序列的季节性趋势。



从上图可以看出，时间序列有季节性变化。夏季平均气温较高，冬季平均气温较低。

6、检查时间序列的平稳性。

可以使用以下方法检查平稳性：

- 绘制滚动统计：有一个窗口，假设窗口大小为 6，然后找到滚动均值和方差来检查平稳性。
- Dickey-Fuller 检验：检验结果包括一个检验统计量和一些不同置信水平的临界值。如果检验统计量小于临界值，可以说时间序列是平稳的。

```
from statsmodels.tsa.stattools import adfuller
```


```
Test statistic:  -1.4095966745887758
p-value:  0.5776668028526356
Critical Values: {'1%': -3.439229783394421, '5%': -2.86545894814762, '10%': -2.56885687561
91392}
```

对于平稳的第一个标准是常数均值。所以不符合标准，因为平均值不是常数，正如从上面的图（黑线）看到的(无静止）。

第二个是常数方差。它看起来是恒定的（静止）。

第三，如果检验统计量小于临界值，可以说时间序列是平稳的。

检验统计量=-1.4，临界值={1%：-3.439229783394421，'5%：-2.86545894814762，'10%：-2.5688568756191392}。检验统计量大于临界值(无静止）。

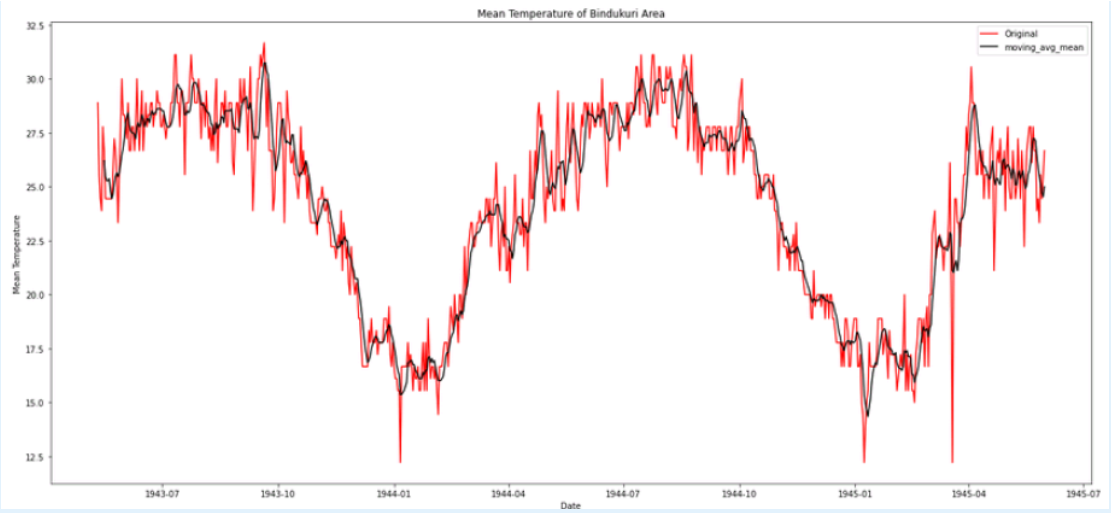
因此，我们时间序列不是平稳的。
7、使时间序列静止。

如前所述，时间序列的非平稳性背后有两个原因：

- 趋势：随时间变化的平均值。我们需要时间序列平稳的常数均值。
- 季节性：特定时间的变化。我们需要时间序列平稳的常数变化。

先解趋势（常均值）问题。最流行的方法是移动平均法。

- 移动平均线：有一个窗口，用来计算过去 n 个样本的平均值是窗口大小。




```
Test statistic:  -11.138514335138469
p-value:  3.150868563164674e-20
Critical Values: {'1%': -3.4392539652094154, '5%': -2.86546960465041, '10%': -2.5688625527
782327}
```

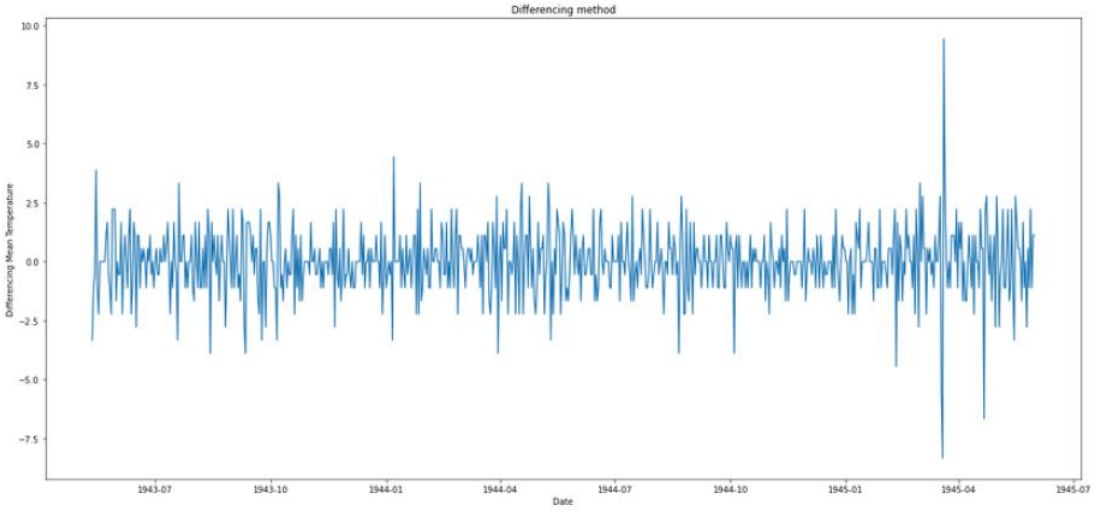
常量平均值标准：从上面的图（黑线）可以看出，平均值看起来像常量(静止)。

第二个是常数方差。它看起来是恒定的(静止）。

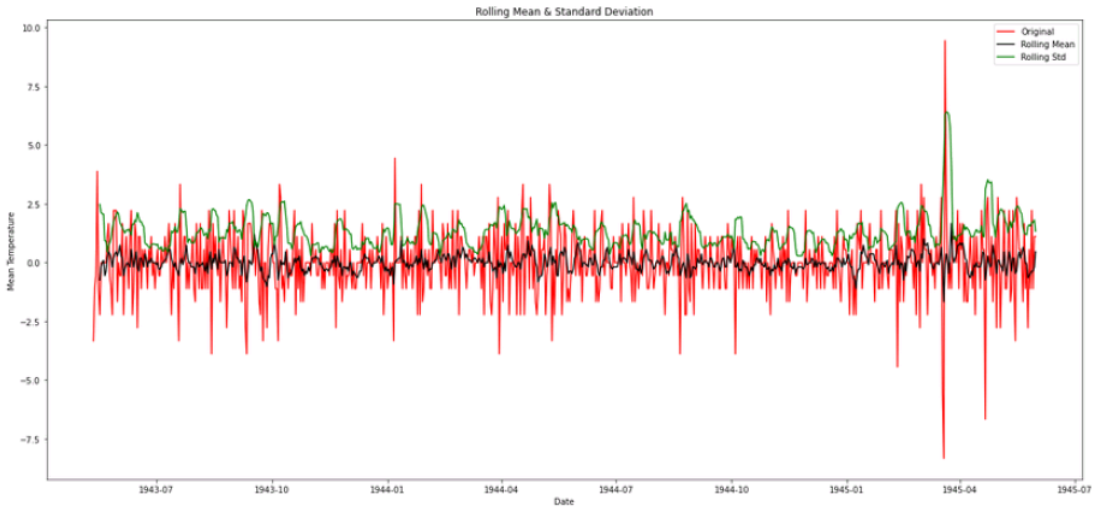
检验统计量小于 1%的临界值，所以 99%的置信度说这是一个平稳序列(是（静止），实现了平稳时间序列。

8、避免趋势性和季节性的方法。

- 差分法：是最常用的方法之一。其思想是取时间序列和移位时间序列之间的差。




```
# 检查平稳性：均值、方差（std）和 adfuller 检验
```




```
Test statistic:  -11.678955575105388
p-value:  1.7602075693557824e-21
Critical Values: {'1%': -3.439229783394421, '5%': -2.86545894814762, '10%': -2.5688568756191392}
```

常量平均值标准：从上面的图（黑线）可以看出，平均值看起来像常量(静止）。

第二个是常数方差。它看起来是恒定的(静止）

检验统计量小于 1%的临界值，所以 99%的置信度说这是一个平稳序列(是（静止）
9、预测时间序列。

上文学习了两种不同的方法：移动平均法和差分法，以避免趋势和季节性问题。

使用差分方法进行预测时间序列。同时，预测方法为自回归综合移动平均值 ARIMA。

- AR：自回归（p）：AR 项只是依赖变量的滞后。例如，假设 p 是 3，则使用 x（t-1）、x（t-2）和 x（t-3）来预测 x（t）。
- I： 综合（d）：这些是非测量差异的数量。例如，在例子中采用一阶差。传递这个变量，然后把 d=0。
- MA：移动平均值（q）：MA 项是预测方程中滞后预测误差。

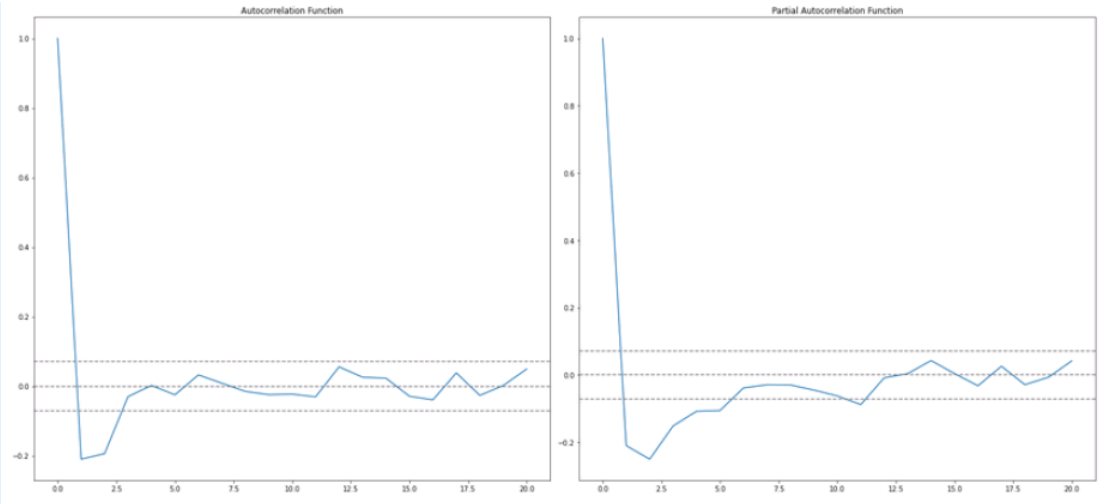
（p，d，q）是 ARIMA 模型的参数。

为了选择 p，d，q 参数，将使用两个不同的图。

- 自相关函数（ACF）：时间序列与滞后时间序列相关性的测量。
- 部分自相关函数（PACF）：测量时间序列与滞后时间序列之间的相关性。

```
# ACF 和 PACF

from statsmodels.tsa.stattools import acf, pacf
```



两条虚线是置信区间。使用这些线来确定“p”和“q”值。

选择 p：PACF 图表第一次穿过上置信区间的滞后值。p=1。

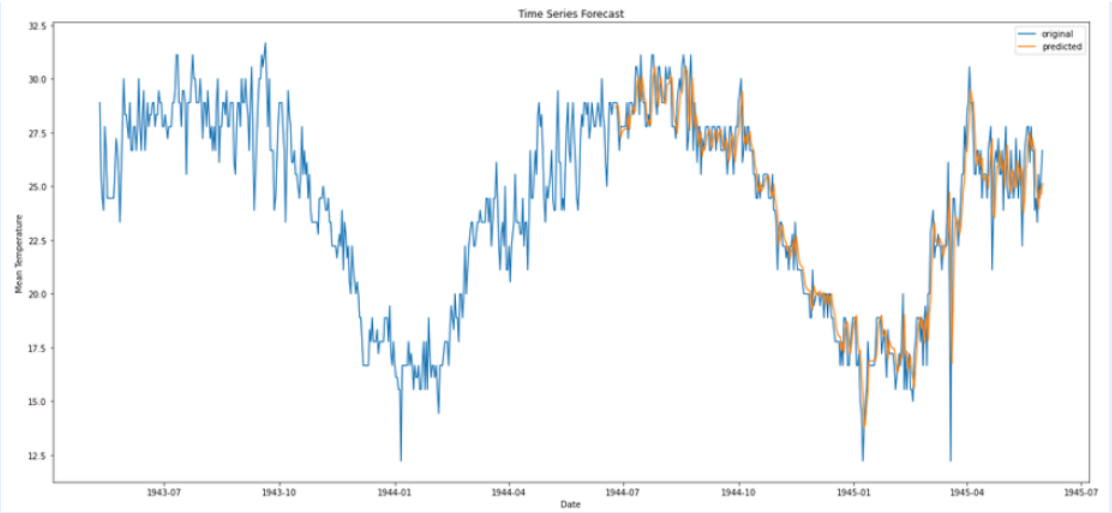
选择 q：ACF 图表第一次穿过置信区间上限的滞后值。q=1。
10、使用（1,0,1）作为 ARIMA 模型的参数并预测。

ARIMA：来自 statsmodels 库。

datetime：使用它的 start 和 end 索引的 predict 方法。

```
from statsmodels.tsa.arima_model import ARIMA

from pandas import datetime
```



11、预测和可视化所有路径并查找均方误差。

```
from sklearn.metrics import mean_squared_error
```

```
error:  1.8625820192380373
```