

بخش تشریحی:

سوال اول:

- سن : پیوسته،
- جنسیت : باینری،
- درآمد : پیوسته،
- وضعیت تاهل : اسمی،
- فرزند دارد : باینری،
- شغل : اسمی،
- میزان تحصیلات : ترتیبی،
- تعداد اعضای خانواده : گسسته

❖ Histogram: پیوسته مانند سن، درآمد

❖ Pie Chart: باینری و اسمی مانند جنسیت، وضعیت تاهل، فرزند دارد و شغل

❖ Box Plot: مناسب داده های ترتیبی و پیوسته مثل میزان تحصیلات، سن و درآمد

❖ Bar Chart: مناسب دیتای گسسته و باینری و اسمی مانند تعداد اعضای خانواده، جنسیت، وضعیت تاهل، فرزند دارد و شغل

سوال دوم:

۱. ابتدا فرمول میانگین، میانه، چارک اول و سوم و همچنین انحراف معیار را مینویسیم:

$$\bar{x}_A = \frac{\sum_{i=1}^n x_i}{n}$$

- میانگین:
- میانه: اگر تعداد فرد باشد و دیتاها هم سورت شوند وسطی و اگر زوج باشند و سورت شده میانگین دو عدد وسط

- چارک اول: میانه نیمه اول دیتا
- چارک سوم: میانه نیمه دوم دیتا

$$\sigma_A = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x}_A)^2}{n}}$$

- انحراف معیار:

پس از فهم فرمول ها به سراغ انجام محاسبات برای هر سری ویژگی میرویم که نتایج آن به شرح زیر است:

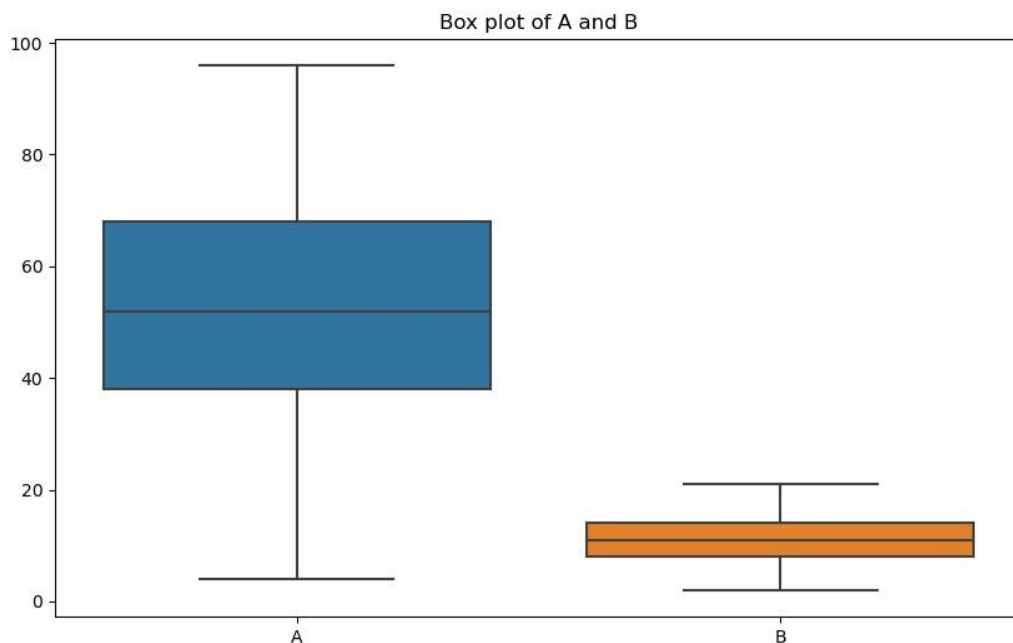
| سری ویژگی | میانگین | میانه | چارک اول | چارک سوم | انحراف معیار |
|-----------|---------|-------|----------|----------|--------------|
| A | ۵۰.۷۳ | ۵۲ | ۳۸ | ۶۸ | ۲۵.۷۸ |
| B | ۱۰.۸ | ۱۱ | ۸ | ۱۴ | ۵.۳۹ |

۲. برای رسم Box Plot برای این قسمت از کد پایتون استفاده میکنیم ساختار کد و خروجی به شرح زیر است:

```

boxplot.py ×
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 A = [55, 72, 60, 54, 42, 64, 43, 89, 96, 38, 79, 52, 56, 92, 7, 8, 24, 39, 44, 68, 68, 52, 4, 16, 73, 46, 96, 38, 20, 27]
5 B = [11, 16, 13, 11, 9, 14, 9, 19, 20, 8, 17, 11, 12, 20, 2, 3, 5, 8, 9, 14, 14, 11, 2, 4, 15, 9, 21, 8, 4, 5]
6
7 plt.figure(figsize=(10, 6))
8 sns.boxplot(data=[A, B])
9 plt.xticks([0, 1], ['A', 'B'])
10 plt.title('Box plot of A and B')
11 plt.show()

```



در مورد نحوه پراکندگی هم واضح است که در سری ویژگی A دامنه و رنج تغییرات بالاست و به طور کلی فشردگی داده ها حول میانه بالاست که همین دلیل باعث شده که دامنه رنج آبی کم باشد در مورد سری ویژگی B هم میتوان گفت به طور کلی داده ها اندازه کوچکتری دارند و تغییرات آن ها نیز کم است باز هم بازه چارک اول تا سوم نسبت به کل سری ویژگی اندازه کوچکی دارد که نشان دهنده تجمع داده در حول میانه است.

۳. نمودار هیستوگرام (۲۰ بین) :

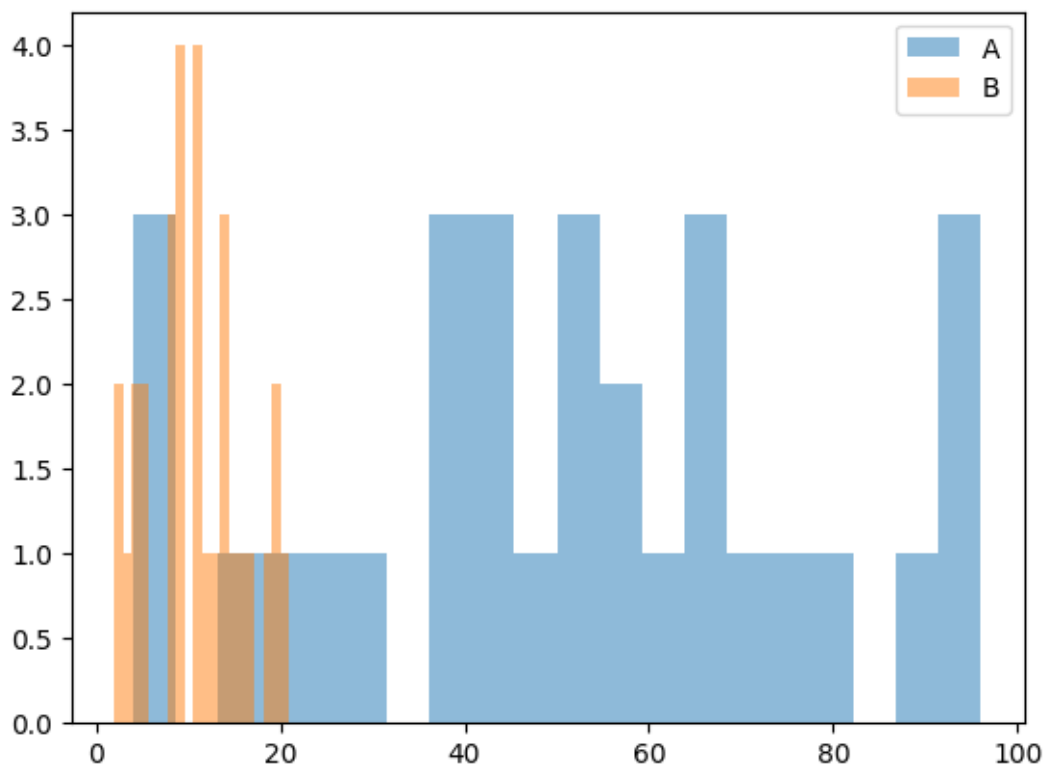
```
import matplotlib.pyplot as plt

A = [55, 72, 60, 54, 42, 64, 43, 89, 96, 38, 79, 52, 56, 92, 7, 8, 24, 39, 44, 68, 68, 52, 4, 16, 73, 46, 96, 38, 20, 27]
B = [11, 16, 13, 11, 9, 14, 9, 19, 20, 8, 17, 11, 12, 20, 2, 3, 5, 8, 9, 14, 14, 11, 2, 4, 15, 9, 21, 8, 4, 5]

plt.hist(A, bins=20, alpha=0.5, label='A')
plt.hist(B, bins=20, alpha=0.5, label='B')

plt.legend(loc='upper right')

plt.show()
```



۴. فرمول آن به صورت مقابل است که از میانگین کم میکنیم و بر انحراف معیار تقسیم میکنیم به عبارتی هر ویژگی چند انحراف معیار از میانگین فاصله دارد.

$$z = \frac{x - \mu}{\sigma}$$

انجام یک مورد : $۵۰.۷۳ - ۵۵$ تقسیم بر ۲۵.۷۸ که معادل تقریباً ۰.۱۶ میشود.

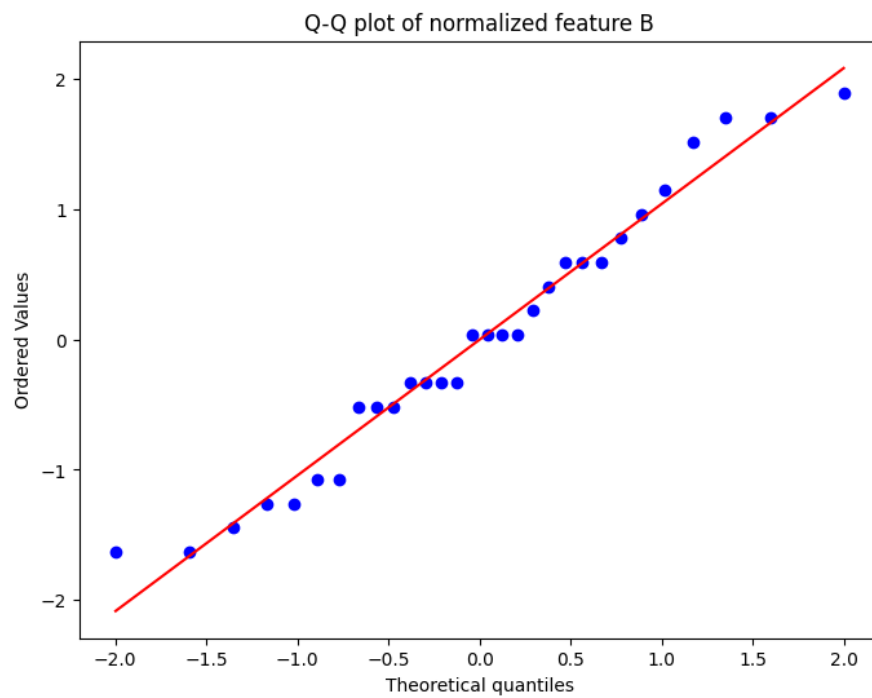
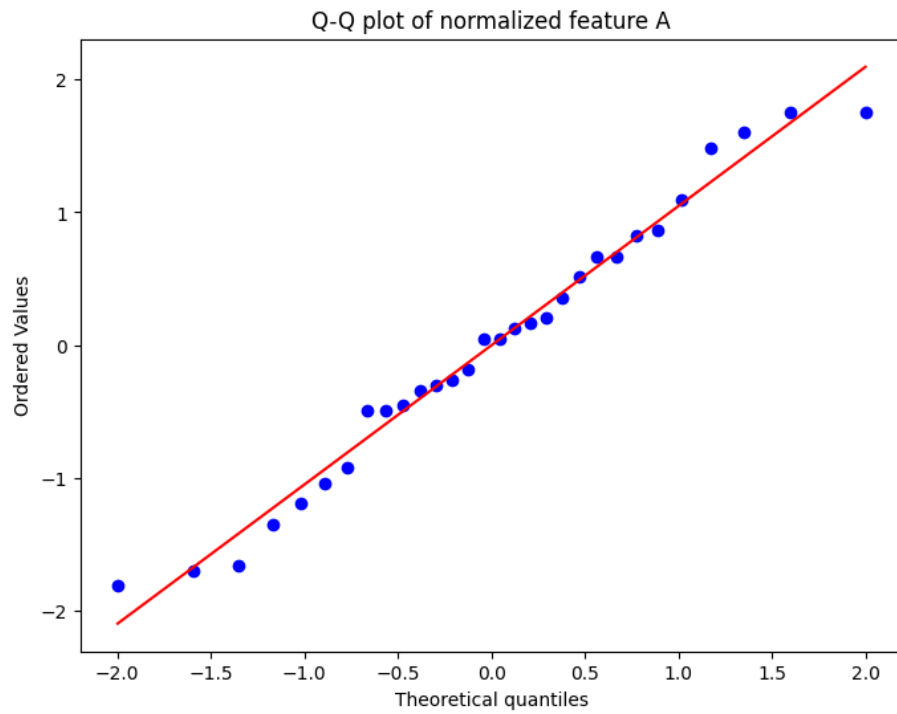
نرمال شده A:

```
[ 0.16546308  0.82473006  0.35936513  0.12668267 -0.33868225  0.51448678
-0.29990184  1.48399703  1.7554599  -0.49380389  1.09619293  0.04912185
 0.20424349  1.60033826 -1.69599661 -1.6572162  -1.03672963 -0.45502348
-0.26112143  0.66960842  0.66960842  0.04912185 -1.81233784 -1.34697292
 0.86351047 -0.18356061  1.7554599  -0.49380389 -1.19185127 -0.9203884 ]
```

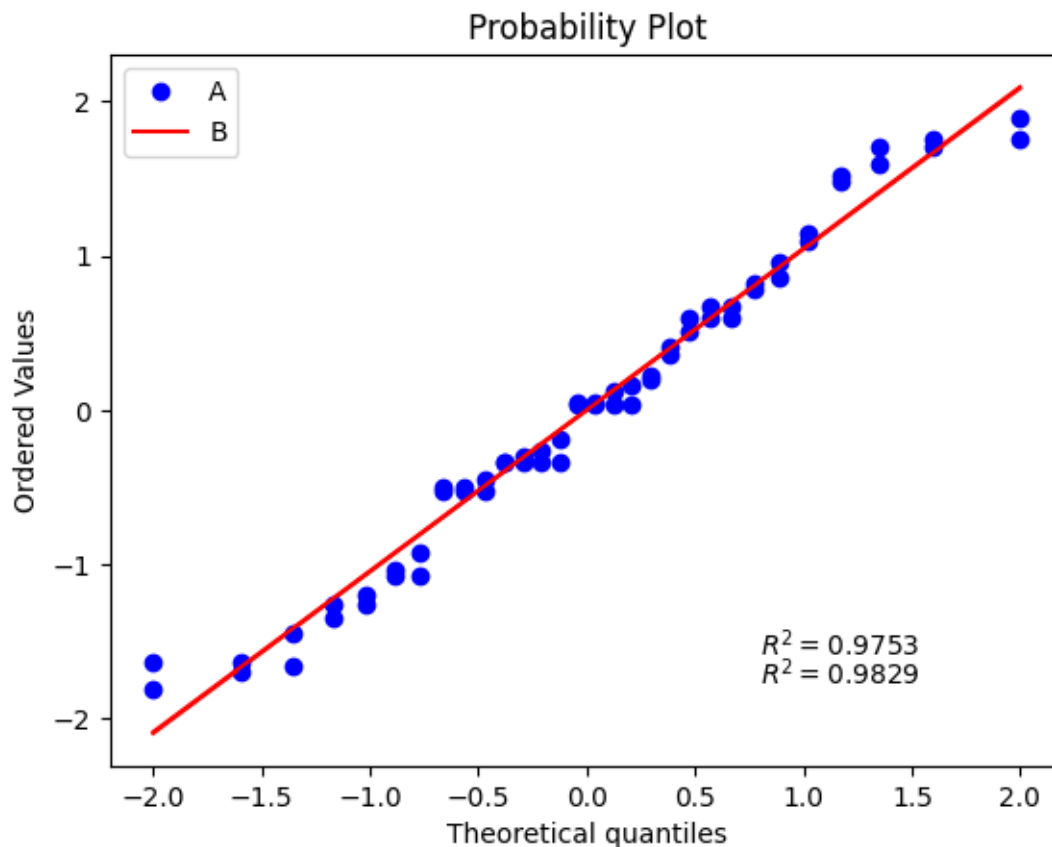
نرمال شده B:

```
[ 0.03707945  0.96406563  0.40787392  0.03707945 -0.33371503  0.59327116
-0.33371503  1.52025735  1.70565458 -0.51911226  1.14946287  0.03707945
 0.22247668  1.70565458 -1.63149569 -1.44609845 -1.07530398 -0.51911226
-0.33371503  0.59327116  0.59327116  0.03707945 -1.63149569 -1.26070121
 0.7786684  -0.33371503  1.89105182 -0.51911226 -1.26070121 -1.07530398]
```

در ادامه به سراغ نمودار Q-Q در مقایسه با توزیع نرمال می‌رویم :



و اگر بخواهیم آنها را با هم مقایسه کنیم:



چیزی که قابل بیان است، این است که در مقادیر میانی، شباهت دو توزیع به هم زیاد است ولی در انتها و ابتدا این انحراف زیاد میشود.

۵. برای تشخیص این مورد از ضریب همبستگی پیرسون استفاده میکنیم، که میزان همبستگی خطی دو توزیع را بدست میآورد:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

با محاسبه این مقدار برای توزیع به عدد تقریباً ۰.۹۹۶ میرسیم که به شدت بالاست ، پس میتوان همبستگی را تایید کرد.

بخش عملی:

بر روی **colab** ران میشود و تنها لازم است فایل زیپ دیتاست در کنار فایل آپلود شود.

پیش پردازش:

۱. ابتدا در رابطه با مشکلات ادغام کردن فایل ها و سپس راه حل های آن میپردازیم، همانطور که در pdf هم اشاره شد اصلی ترین مشکل تفاوت های موجود در نام ستون ها بود، مشکل بعدی هم وجود یک سری پارامترها در یک سری فایل ها بود که در دیگر ایستگاه ها حضور نداشتند.

با نگاه دقیق به فایل ها دریافتیم که این تفاوت ها معمولاً آنچنان قوی نیستند که نام پارامتر را به طور کلی تغییر دهند مثلاً صرفاً _ اضافه شده است یا حروف اول کلمات بزرگ هستند یا بین بخش های مختلف . گذاشته است، که این موارد به راحتی به کمک پایتون قابل حل است (اگر تفاوت در نام ها هم موجود بود دو روش میتوانستیم استفاده کنیم ۱. یک فانکشن استخراجگر بنویسیم که مثلاً به دنبال تکه temp و ۳ برود که هر ترکیبی از آن میتواند بیانگر دما در ساعت ۳ باشد ۲. یک فانکشن تعریف کنیم برای تشخیص نزدیکی کلمات بدین معنی که نزدیک ترین کلمه در فایل های مختلف یک ستون را تشکیل دهند و برای فاصله بین کلمات هم میتوانیم مثلاً از cosine similarity استفاده کنیم.) ولی الان با استفاده از فانکشن های زیر که ابتدا همه را lower case کنیم سپس _ و . را حذف کنیم مشکلمان حل میشود :

```
import pandas as pd
import os

def standardize_column_names(df):
    df.columns = df.columns.str.lower()
    df.columns = df.columns.str.replace(' ', '')
    df.columns = df.columns.str.replace('_', '')
    df.columns = df.columns.str.replace('.', '')
    return df
```

در گام بعدی برای هندل کردن ستون های متفاوت در ایستگاه های متفاوت تنها موقع مرج کردن دیتافریم ignore_index را True میگذاریم که دیتافریم جدید را دوباره index گذاری میکند و ستون های اضافی هم هندل میشود.

```
def standardize_column_names(df):
    df.columns = df.columns.str.lower()
    df.columns = df.columns.str.replace(' ', '')
    df.columns = df.columns.str.replace('-', '')
    df.columns = df.columns.str.replace('.', '')
    return df

def merge_excel_files(directory):
    merged_df = pd.DataFrame()

    for filename in os.listdir(directory):
        if filename.endswith(".csv"):
            df = pd.read_csv(os.path.join(directory, filename))

            df = standardize_column_names(df)

            merged_df = pd.concat([merged_df, df], ignore_index=True)

    return merged_df

merged_df = merge_excel_files('weatherAUS/')
merged_df.to_excel("combined_data.xlsx", index=False)
```

کد را هم از این جا به بعد روی google colab ران میکنیم چون حجم محاسبات زیاد شده است و میتوان از GPU TPU استفاده کرد.

در مورد توضیح کد هم ابتدا تمامی فایل های csv را میخواند نام ستون ها را نرمال میکند و سپس با کمک concat آنها را با هم ترکیب میکند و در نهایت در یک فایل excel ذخیره میکند خروجی هم ساختاری مانند شکل زیر دارد:

File Home Insert Page Layout Formulas Data Review View Help

combined_data (4).xlsx - Excel

barbod mazlaghani

Clipboard Font Alignment Number Styles Cells Editing Add-ins

Cut Copy Paste Format Painter

Calibri 11

B I U

Font

Alignment

Number

Styles

Cells

Editing

Add-ins

General

Conditional Formatting

Format as Table

Cell Styles

Insert

Delete

Format

Σ AutoSum

Fill

Sort & Filter

Find & Select

Add-ins

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| A1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|

| combined_data (4).xlsx - Excel | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|-------------|---------|--------------|------|----------|------------|------|---------|--------------|----------|------------|-------------|----------|---------|---------|-------------|--------------|------|-------|-------------|----------|-------------|----------|-------------|---------|---------|
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 1 | windgustdir | temp9am | windspeed9am | rain | tomorrow | winddir9am | date | temp3pm | windspeed3pm | location | winddir3pm | pressure9am | cloud3pm | mintemp | maxtemp | pressure3pm | windspeed3pm | rain | today | humidity3pm | rainfall | humidity9am | cloud9am | evaporation | mintemp | maxtemp |
| 2 | ESE | 21.7 | 30 | No | E | 2013-03-01 | 28.4 | 24 | Uluru | E | 1010.6 | | 19.7 | 30 | 1007.5 | 48 | No | 54 | 0.8 | 76 | | | | | | |
| 3 | E | 24.6 | 22 | No | E | 2013-03-02 | 31.3 | 11 | Uluru | N | 1010.5 | | 21.6 | 33.1 | 1006.5 | 33 | No | 33 | 0 | 44 | | | | | | |
| 4 | E | 27.6 | 24 | No | ENE | 2013-03-03 | 34.5 | 13 | Uluru | SSE | 1006.9 | | 21.3 | 36.1 | 1002.7 | 33 | No | 27 | 0 | 39 | | | | | | |
| 5 | ENE | 28.7 | 28 | No | E | 2013-03-04 | 35.4 | 13 | Uluru | SSE | 1006 | | 22.9 | 37.7 | 1002.1 | 39 | No | 22 | 0 | 35 | | | | | | |
| 6 | S | 29.9 | 20 | No | E | 2013-03-05 | 37.3 | 19 | Uluru | S | 1006.9 | | 24 | 39 | 1003.5 | 39 | No | 21 | 0 | 33 | | | | | | |
| 7 | S | 30.4 | 22 | No | E | 2013-03-06 | 38.1 | 15 | Uluru | SW | 1007.9 | | 24.3 | 39.7 | 1003.9 | 46 | No | 17 | 0 | 36 | | | | | | |
| 8 | WSW | 29.9 | 24 | No | E | 2013-03-07 | 37.1 | 15 | Uluru | SSE | 1008.3 | | 26.6 | 38.6 | 1004.5 | 33 | No | 21 | 0 | 41 | | | | | | |
| 9 | SSE | 30.9 | 13 | No | E | 2013-03-08 | 36.8 | 15 | Uluru | SSW | 1008.9 | | 26.9 | 39.4 | 1005.1 | 54 | No | 24 | 0 | 39 | | | | | | |
| 10 | SE | 28.6 | 22 | No | ESE | 2013-03-09 | 37.5 | 9 | Uluru | NE | 1010.4 | | 24.4 | 40 | 1005.9 | 54 | No | 24 | 0 | 54 | | | | | | |
| 11 | NE | 31 | 24 | No | E | 2013-03-10 | 37.7 | 13 | Uluru | SE | 1010.7 | | 25.4 | 39.8 | 1006.8 | 37 | No | 20 | 0 | 40 | | | | | | |
| 12 | ENE | 31.9 | 17 | No | NNE | 2013-03-11 | 37.9 | 19 | Uluru | E | 1010.3 | | 23.9 | 39.5 | 1006 | 41 | No | 13 | 0 | 26 | | | | | | |
| 13 | S | 32 | 15 | No | ENE | 2013-03-12 | 39.7 | 7 | Uluru | NE | 1009.2 | | 20.6 | 40.9 | 1005.3 | 57 | No | 13 | 0 | 22 | | | | | | |
| 14 | S | 24.1 | 28 | No | SSE | 2013-03-13 | 34.6 | 19 | Uluru | SSW | 1012.6 | | 22.3 | 37.3 | 1008.5 | 52 | No | 24 | 0 | 56 | | | | | | |
| 15 | ESE | 23.3 | 26 | No | ESE | 2013-03-14 | 32.7 | 11 | Uluru | SE | 1013.8 | | 19.8 | 33.8 | 1009.4 | 39 | No | 24 | 0 | 34 | | | | | | |
| 16 | E | 23.5 | 24 | No | E | 2013-03-15 | 34.6 | 7 | Uluru | E | 1015.3 | | 17.2 | 35.9 | 1010.7 | 35 | No | 13 | 0 | 29 | | | | | | |
| 17 | SE | 27.9 | 22 | No | E | 2013-03-16 | 36.9 | 22 | Uluru | ESE | 1016.1 | | 19.1 | 37.9 | 1012.7 | 43 | No | 10 | 0 | 20 | | | | | | |
| 18 | ESE | 26.4 | 33 | No | ESE | 2013-03-17 | 34.2 | 17 | Uluru | SE | 1018.6 | | 21.9 | 35.4 | 1013.7 | 46 | No | 20 | 0 | 33 | | | | | | |
| 19 | E | 24.4 | 26 | No | E | 2013-03-18 | 33.5 | 17 | Uluru | SSE | 1016.3 | | 19.6 | 34.9 | 1012 | 35 | No | 20 | 0 | 32 | | | | | | |
| 20 | E | 25.5 | 22 | No | E | 2013-03-19 | 34.9 | 19 | Uluru | ENE | 1015.6 | | 20 | 36.6 | 1011.5 | 30 | No | 15 | 0 | 17 | | | | | | |
| 21 | WSW | 27.5 | 9 | No | ENE | 2013-03-20 | 37.8 | 11 | Uluru | NNW | 1014.8 | | 19.7 | 39.6 | 1009.3 | 30 | No | 13 | 0 | 17 | | | | | | |
| 22 | SSW | 33.2 | 22 | No | S | 2013-03-21 | 38.7 | 19 | Uluru | SW | 1010.4 | | 25.9 | 39.8 | 1006.7 | 39 | No | 17 | 0 | 25 | | | | | | |
| 23 | S | 25.5 | 20 | No | SE | 2013-03-22 | 36.1 | 17 | Uluru | SSE | 1013.9 | | 22.9 | 37.8 | 1010.2 | 35 | No | 17 | 0 | 39 | | | | | | |
| 24 | E | 24.5 | 28 | No | E | 2013-03-23 | 32.6 | 22 | Uluru | ESE | 1014.9 | | 22.4 | 34.6 | 1012.2 | 46 | No | 25 | 0.4 | 41 | | | | | | |
| 25 | SSW | 28.1 | 26 | No | E | 2013-03-24 | 35.4 | 20 | Uluru | NNE | 1015.6 | | 22.5 | 37.2 | 1011.1 | 61 | No | 21 | 0 | 28 | | | | | | |
| 26 | SSW | 23.5 | 9 | No | SE | 2013-03-25 | 26.9 | 19 | Uluru | NNE | 1009.3 | | 21.3 | 32.5 | 1011.1 | 48 | No | 54 | 0 | 55 | | | | | | |
| 27 | N | 28.4 | 26 | No | ENE | 2013-03-26 | 36.7 | 26 | Uluru | N | 1010.1 | | 19.6 | 38 | 1004.5 | 46 | No | 22 | 0 | 20 | | | | | | |
| 28 | SW | 26.8 | 15 | Yes | S | 2013-03-27 | 32.7 | 17 | Uluru | SW | 1009.7 | | 25.4 | 34.7 | 1006.6 | 50 | No | 39 | 0 | 61 | | | | | | |
| 29 | ESE | 17.7 | 22 | No | SE | 2013-03-28 | 19.5 | 30 | Uluru | ESE | 1018.1 | | 16.9 | 20 | 1016.6 | 46 | Yes | 71 | 40.2 | 98 | | | | | | |

۲. پنج سطر ابتدایی به صورت زیر است :

DM_HW1.ipynb - Colaboratory

Google

colab.research.google.com/drive/1wSypD6PmCIVw6H0EWq4cLcDaNf6RODN#scrollTo=Vi9MxpWC9ju

lms

هوش محاسباتی

Future

cad

anime

courses to watch in...

bootstrap snippets

khafan youtube

IUST course

sports

EVERYTIME need this

Untitled Blank Proje...

All Bookmarks

DM_HW1.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

Files

sample_data

weatherAUS

combined_data.xlsx

weatherAUS.zip

Code

Text

```
[6] <ipython-input-6-d0a4871e8ca>:8: FutureWarning: The default value of regex will change from True to False in a future version. In addition, s
df.columns = df.columns.str.replace('.', '')

merged_df.head(5)
```

| | windgustdir | temp9am | windspeed9am | rain | tomorrow | winddir9am | date | temp3pm | windspeed3pm | location | winddir3pm | ... | humidity3pm | rainfall |
|---|-------------|---------|--------------|------|----------|------------|------|---------|--------------|----------|------------|------|-------------|----------|
| 0 | ESE | 21.7 | 30.0 | No | E | 2013-03-01 | 28.4 | 24.0 | Uluru | E | ... | 54.0 | 0.8 | |
| 1 | E | 24.6 | 22.0 | No | E | 2013-03-02 | 31.3 | 11.0 | Uluru | N | ... | 33.0 | 0.0 | |
| 2 | E | 27.6 | 24.0 | No | ENE | 2013-03-03 | 34.5 | 13.0 | Uluru | SSE | ... | 27.0 | 0.0 | |
| 3 | ENE | 28.7 | 28.0 | No | E | 2013-03-04 | 35.4 | 13.0 | Uluru | SSE | ... | 22.0 | 0.0 | |
| 4 | S | 29.9 | 20.0 | No | E | 2013-03-05 | 37.3 | 19.0 | Uluru | S | ... | 21.0 | 0.0 | |

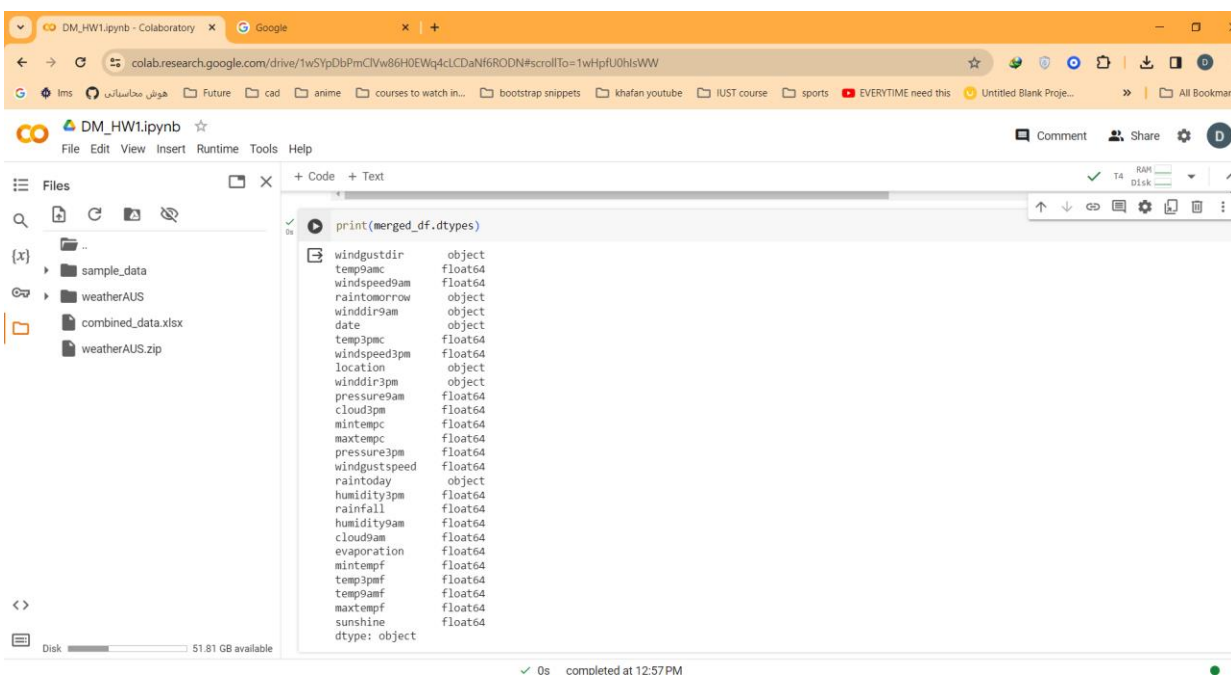
5 rows x 27 columns

Disk

51.81 GB available

0s completed at 12:32 PM

۳. به کمک dtypes این کار را انجام می‌دهیم :



```
print(merged_df.dtypes)
```

| | |
|---------------|---------|
| windgustdir | object |
| temp9amc | float64 |
| windspeed9am | float64 |
| rain tomorrow | object |
| winddir9am | object |
| date | object |
| temp3pmc | float64 |
| windspeed3pm | float64 |
| location | object |
| winddir3pm | object |
| pressure9am | float64 |
| cloud3pm | float64 |
| mintempc | float64 |
| maxtempc | float64 |
| pressure3pm | float64 |
| windgustspeed | float64 |
| rain today | object |
| humidity3pm | float64 |
| rainfall | float64 |
| humidity9am | float64 |
| cloud9am | float64 |
| evaporation | float64 |
| mintempf | float64 |
| temp3pmf | float64 |
| temp9amf | float64 |
| maxtempf | float64 |
| sunshine | float64 |
| dtype: | object |

۴. برای اینکار ابتدا لازم است که تمامی ستون‌هایی که temp دارند را شناسایی کنیم سپس تایپ آنها را به int تغییر دهیم، همچنین در اینجا یک چالش داشتیم آن هم اینکه Nan را نمیتوان به Int تغییر داد برای همین این مقادیر را در این ستون‌ها به مقدار -۹۹ تغییر دادیم که در داده‌های خودمان غیرممکن است (اگر با صفر پر میکردیم ممکن بود داده واقعا در آن لحظه صفر باشد و در ادامه به مشکل می‌خوردیم). خروجی و کد را در ادامه می‌بینیم :

```
temp_columns = [col for col in merged_df.columns if 'temp' in col]

for col in temp_columns:
    merged_df[col] = merged_df[col].fillna(-99).astype(int)
print(merged_df.dtypes)
```

The screenshot shows a Google Colab notebook with the following code and output:

```

mtempf
temp3pmf
temp9amf
maxtempf
sunshine
dtype: object

```

```

[ ] merged_df.head(5)

```

| | windgustdir | temp9amc | windspeed9am | raintomorrow | winddir9am | date | temp3pmc | windspeed3pm | location | winddir3pm | humidity3pm | rainfall |
|---|-------------|----------|--------------|--------------|------------|------------|----------|--------------|----------|------------|-------------|----------|
| 0 | ESE | 21 | 30.0 | No | E | 2013-03-01 | 28 | 24.0 | Uluru | E | 54.0 | 0.8 |
| 1 | E | 24 | 22.0 | No | E | 2013-03-02 | 31 | 11.0 | Uluru | N | 33.0 | 0.0 |
| 2 | E | 27 | 24.0 | No | ENE | 2013-03-03 | 34 | 13.0 | Uluru | SSE | 27.0 | 0.0 |
| 3 | ENE | 28 | 28.0 | No | E | 2013-03-04 | 35 | 13.0 | Uluru | SSE | 22.0 | 0.0 |
| 4 | S | 29 | 20.0 | No | E | 2013-03-05 | 37 | 19.0 | Uluru | S | 21.0 | 0.0 |

5 rows x 27 columns

Connecting to Python 3 Google Compute Engine backend (GPU)

۵. برای بدست آوردن سائز دیتافریم در RAM از کتابخانه sys استفاده میکنیم :

```

import sys

size_in_bytes = sys.getsizeof(merged_df)
size_in_kb = size_in_bytes / 1024
size_in_mb = size_in_kb / 1024

print(f"Size of df in Bytes: {size_in_bytes}")
print(f"Size of df in KB: {size_in_kb}")
print(f"Size of df in MB: {size_in_mb}")

```

```

Size of df in Bytes: 84778309
Size of df in KB: 82791.3173828125
Size of df in MB: 80.85089588165283

```

۶. بدین صورت تمامی متغیرهای باینری و اسمی را به category تغییر میدهم، ابتدا مقادیر را از NULL به Unknown تغییر میدهم سپس به کمک astype به category تبدیل میکنیم :

The screenshot shows a Google Colab notebook with the following code and output:

```

merged_df['windgustdir'].fillna('unknown', inplace=True)
merged_df['raintomorrow'].fillna('unknown', inplace=True)
merged_df['date'].fillna('unknown', inplace=True)
merged_df['location'].fillna('unknown', inplace=True)
merged_df['winddir3pm'].fillna('unknown', inplace=True)
merged_df['raintoday'].fillna('unknown', inplace=True)
merged_df['winddir9am'].fillna('unknown', inplace=True)

merged_df['windgustdir'] = merged_df['windgustdir'].astype('category')
merged_df['raintomorrow'] = merged_df['raintomorrow'].astype('category')
merged_df['date'] = merged_df['date'].astype('category')
merged_df['location'] = merged_df['location'].astype('category')
merged_df['winddir3pm'] = merged_df['winddir3pm'].astype('category')
merged_df['raintoday'] = merged_df['raintoday'].astype('category')
merged_df['winddir9am'] = merged_df['winddir9am'].astype('category')

print(merged_df.dtypes)

```

```

windgustdir    category
temp3pmc       int64
windspeed9am   float64
raintomorrow   category
winddir9am     category
date           category
temp3pmc       int64
windspeed3pm   float64
location       category
winddir3pm     category

```

completed at 1:57 PM

۷. با توجه به تغییرات داده شده و تغییرات که در ادامه میبینیم حجم به ۲۳ مگابایت کاهش یافت:

```
print(merged_df.dtypes)
```

```
winddir9am      category
humidity3pm     float64
raintoday       category
location        category
raintomorrow    category
humidity9am     float64
cloud3pm        float64
windgustdir     category
temp9amc       int64
date           category
temp3pmc       int64
mintempc       int64
pressure9am    float64
windspeed3pm   float64
evaporation     float64
windgustspeed   float64
rainfall       float64
pressure3pm     float64
sunshine       float64
windspeed9am    float64
winddir3pm     category
maxtempc       int64
cloud9am       float64
mintemp        int64
maxtempf       int64
temp9amf       int64
temp3pmf       int64
dtype: object
```

```
import sys

size_in_bytes = sys.getsizeof(merged_df)
size_in_kb = size_in_bytes / 1024
size_in_mb = size_in_kb / 1024

print(f"Size of df in Bytes: {size_in_bytes}")
print(f"Size of df in KB: {size_in_kb}")
print(f"Size of df in MB: {size_in_mb}")
```

```
Size of df in Bytes: 24809347
Size of df in KB: 24227.8779296875
Size of df in MB: 23.66083704871045
```

که اگر بخواهیم تغییرات و درصد آن‌ها را نمایش دهیم به صورت مقابل میشود: ۲۳ - ۸۰ که ۵۷ مگ کاهش یافته است که درصد آن هم به صورت حدودا ۷۱٪ کاهش داشتیم.

۸. برای هر کدام از ستون‌ها چون یک سری مقادیر را برای محاسبات تغییر دادیم لازم است تا اندکی تابع محاسبه گر تعداد را تغییر دهیم:

```
import numpy as np

def count_missing_values(column):
    if column.dtype == np.number or column.dtype == np.int64 or column.dtype == np.float64:
        return sum(column == -99) + column.isnull().sum()
    else:
        return sum(column == 'Unknown') + column.isnull().sum()

missing_values_count = merged_df.apply(count_missing_values)
print(missing_values_count)
```

که نتایج آن به صورت زیر است:

```
<ipython-input-18-0c4daf7e9541>:4:
if column.dtype == np.number or c
winddir9am      10566
humidity3pm     4507
raintoday       3261
location        0
raintomorrow    3267
humidity9am     2654
cloud3pm        59358
windgustdir     10326
temp9amc       87251
date           0
temp3pmc       87886
mintempc       87232
pressure9am    15065
windspeed3pm   3062
evaporation     62790
windgustspeed   10263
rainfall       3261
pressure3pm    15028
sunshine       69835
windspeed9am   1767
winddir3pm     4228
maxtempc       87194
cloud9am       55888
mintemp        59713
maxtempf       59527
temp9amf       59976
temp3pmf       61183
dtype: int64
```

۹. برای ستون هایی که خودمان تغییر دادیم مثلا temp9amc که ممکن است در یک ایستگاه کلا بر مبنای فارنهایت ارائه شده باشد، حذف کردن سطر کاملاً کار اشتباهی است، چون دیتا کامل موجود است ولی در سطرهایی که مثلاً جهت باد وجود ندارد و ما هم نمیتوانیم پیش بینی داشته باشیم حذف سطر منطقی است همچنین در سطرهایی که مثلاً داده قبلی و بعدی را داریم، با یک تقریبی میتوان گفت که داده میانی هم مثل قبلی و بعدی بوده است، یا مثلاً بین دو روز ابری، احتمال زیاد روز بین هم ابری بوده است، همچنین در مورد داده های عددی مانند دما میتوان با یک احتمال خوب میانگین چند داده حول آن را در نظر گرفت، همچنین این مورد پر کردن داده های گم شده کاملاً به کاربرد مورد استفاده ما بستگی دارد شاید مثلاً تنها برای ما مفید باشد و از آن استفاده کنیم پس اگر رطوبت یا ابری بودن را نداشتیم، مشکلی بوجود نیاید. همچنین یک روش دیگری که میتوان استفاده کرد و بنده هم سعی میکنم این روش را در نظر بگیرم این است که چندین ستون با هم چک شوند اگر حجم زیادی از دیتا (تعداد زیادی ستون) از دست رفته بود آن سطر را حذف میکنیم اینطوری هم مطمئن میشیم اشتباه حذف نمیکنیم (مثال توضیح داده شده در سطر دوم همین صفحه) هم اگر دیتای باقی مانده مفید نبود حذف میشود.

۱۰. با توجه به توضیحات بالا کد را پیاده سازی میکنیم ، بدین صورت که در هر row مقدار missing value ها را پیدا میکنیم و بر مبنای آن عمل میکنیم :

```
[34] print(len(merged_df))

145460

def count_missing_values(row):
    if row.dtype == np.number or row.dtype == np.int64 or row.dtype == np.float64:
        return sum(row == -99) + row.isnull().sum()
    else:
        return sum(row == 'Unknown') + row.isnull().sum()

missing_values_count_row = merged_df.apply(count_missing_values, axis=1)

filled_df = merged_df[missing_values_count_row <= 10]

<ipython-input-36-8ba0236c03b1>:2: DeprecationWarning: Converting 'np.inexact' or 'np.floating' to a dtype
if row.dtype == np.number or row.dtype == np.int64 or row.dtype == np.float64:
<ipython-input-37-8ba0236c03b1>:2: DeprecationWarning: Converting 'np.inexact' or 'np.floating' to a dtype
if row.dtype == np.number or row.dtype == np.int64 or row.dtype == np.float64:

print(len(filled_df))

143776
```

۱۱. برای اینکار ابتدا ستون های را پیدا میکنیم که در اسم ستون temp وجود داشته باشد و با f تمام شود سپس با کمک فرمول تبدیل فارنهایت به سلیسیوس این تبدیل را انجام میدهیم :

```
def fahrenheit_to_celsius(value):
    if value != -99:
        return (value - 32) * 5.0/9.0
    else:
        return value

temp_columns = filled_df.filter(regex='temp.*f$').columns

for column in temp_columns:
    filled_df[column] = filled_df[column].apply(fahrenheit_to_celsius)

<ipython-input-39-4f0a866c6687>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

۱۲. از روش چارک برای این کار استفاده کردیم، برای ستون های عددی چارک اول و سوم را حساب کردیم سپس به کمک این دو مقدار IQR را حساب میکنیم، سپس داده هایی که بیش از ۱.۵ برابر IQR، کوچکتر از چارک اول هستند یا همین مقدار بزرگ از چارک سوم هستند را به عنوان outlier تشخیص میدهد که برای درک این مقادیر تعدادی را نیز چاپ کردیم:

```
def find_outliers(column):
    if column.dtype == np.number or column.dtype == np.int64 or column.dtype == np.float64:
        column = column.replace(-99, np.nan)

        Q1 = column.quantile(0.25)
        Q3 = column.quantile(0.75)
        IQR = Q3 - Q1

        outliers = (column < (Q1 - 1.5 * IQR)) | (column > (Q3 + 1.5 * IQR))

        return outliers
    else:
        column = column.replace('Unknown', np.nan)

        return pd.Series([np.nan]*len(column), index=column.index)

outliers = filled_df.apply(find_outliers)
```

```
outlier_rows = outliers.any(axis=1)

outlier_df = filled_df[outlier_rows]

outlier_df = outlier_df.dropna()

print(outlier_df.head(10))
```

| | winddir9am | humidity3pm | raintoday | location | raintomorrow | humidity9am |
|------|------------|-------------|-----------|----------|--------------|-------------|
| 1004 | S | 77.0 | Yes | Sydney | Yes | 78.0 |
| 1008 | SE | 53.0 | Yes | Sydney | Yes | 67.0 |
| 1011 | E | 52.0 | No | Sydney | Yes | 64.0 |
| 1012 | SSE | 54.0 | Yes | Sydney | No | 80.0 |
| 1014 | ESE | 43.0 | Yes | Sydney | No | 77.0 |
| 1019 | S | 65.0 | Yes | Sydney | No | 82.0 |
| 1033 | S | 91.0 | Yes | Sydney | Yes | 94.0 |
| 1034 | ENE | 89.0 | Yes | Sydney | Yes | 94.0 |
| 1035 | ENE | 78.0 | Yes | Sydney | No | 91.0 |
| 1050 | SSW | 71.0 | Yes | Sydney | No | 75.0 |

| | cloud3pm | windgustdir | temp9amc | date | ... | pressure3pm | sunshine |
|------|----------|-------------|----------|------------|-----|-------------|----------|
| 1004 | 8.0 | SSE | 18 | 2010-11-01 | ... | 1012.1 | 0.1 |
| 1008 | 7.0 | SE | 15 | 2010-11-05 | ... | 1022.1 | 2.7 |
| 1011 | 7.0 | SW | 23 | 2010-11-08 | ... | 1015.1 | 7.9 |
| 1012 | 8.0 | SSE | 19 | 2010-11-09 | ... | 1024.2 | 7.0 |
| 1014 | 3.0 | E | 20 | 2010-11-11 | ... | 1012.0 | 9.3 |
| 1019 | 3.0 | ESE | 18 | 2010-11-16 | ... | 1011.3 | 7.9 |
| 1033 | 8.0 | SSW | 17 | 2010-11-30 | ... | 1014.5 | 0.9 |
| 1034 | 8.0 | E | 18 | 2010-12-01 | ... | 1015.3 | 0.0 |
| 1035 | 7.0 | ENE | 20 | 2010-12-02 | ... | 1015.9 | 5.4 |

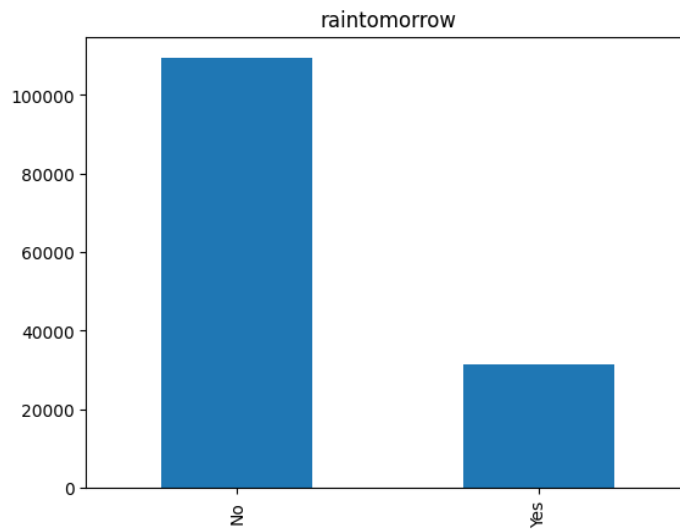
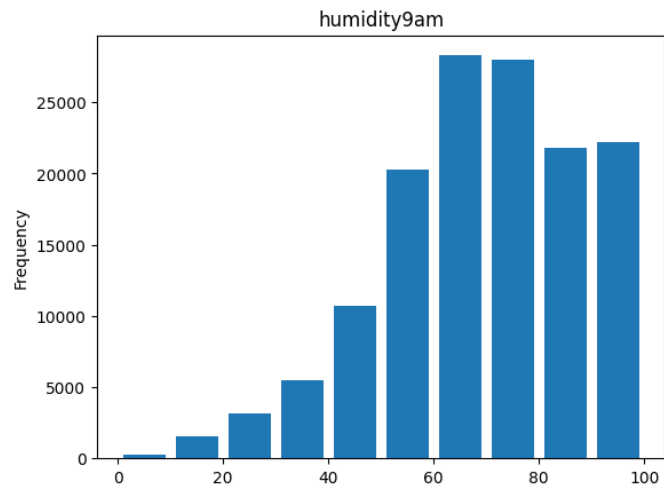
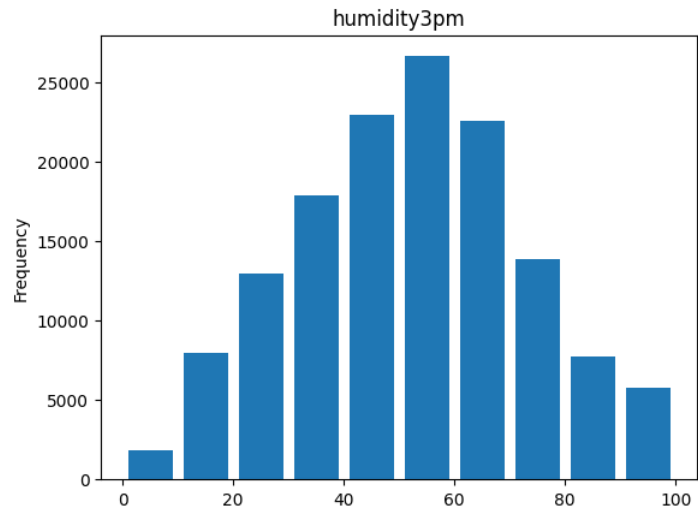
0s completed at 2:40 PM

نمایش دادگان:

۱. در این قسمت برای هر ستون نمودار رسم میکنیم برای مقادیر categorical از نمودار bar chart و برای مقادیر عددی از هیستوگرام. خروجی را در ادامه برای چند مورد میبینیم که کامل آن ها در فایل نوتبوک که در کنار گزارش قرار دارد موجود است :

```
import matplotlib.pyplot as plt
import pandas as pd

for column in df.columns:
    if df[column].dtype.name == 'category':
        df[column].value_counts().plot(kind='bar', title=column)
    else:
        df[column].plot(kind='hist', rwidth=0.8, title=column)
plt.show()
```



۲. این قسمت یک مقدار پیچیده بود، ابتدا لازم بود تا مقدار ماکسیمم و مینیمم را پیدا کنیم، چون هم فارنهایت هم سانتیگراد داشتیم، سپس میانگین را بین این دو مورد میگیریم تا میانگین هر روز بدست بیاید، سپس بر مبنای لوکیشن داده ها را گروه میکنیم و بین مقادیر در روزهای مختلف برای آن میانگین میگیریم، همچنین برای نمایش بهتر داده ها و تغییر رنگ یک سری پراسس روی دیتای انجام میدهیم و نمودار را هم طبق خواسته سوال به صورت افقی نمایش میدهیم.



The screenshot shows a Jupyter Notebook titled "DM_HW1.ipynb". The left sidebar displays a file explorer with folders like "sample_data" and "weatherAUS", and files like "combined_data.x..." and "weatherAUS.zip". The main area contains Python code for data processing and visualization. The code identifies max and min temperature columns, calculates their means, groups by location, and creates a bar chart of average temperatures. A color bar is also included at the bottom.

```
maxtemp_cols = [col for col in df.columns if 'maxtemp' in col.lower() and not df[col].isna().all()]
mintemp_cols = [col for col in df.columns if 'mintemp' in col.lower() and not df[col].isna().all()]

df['avg_maxtemp'] = df[maxtemp_cols].apply(lambda row: np.nanmean(row), axis=1)
df['avg_mintemp'] = df[mintemp_cols].apply(lambda row: np.nanmean(row), axis=1)

df['avg_temp'] = df[['avg_maxtemp', 'avg_mintemp']].mean(axis=1)

location_avg_temp = df.groupby('location')['avg_temp'].mean().reset_index()

location_avg_temp_sorted = location_avg_temp.sort_values(by='avg_temp', ascending=True)
cmap = cm.get_cmap('coolwarm')

norm = plt.Normalize(location_avg_temp_sorted['avg_temp'].min(), location_avg_temp_sorted['avg_temp'].max())

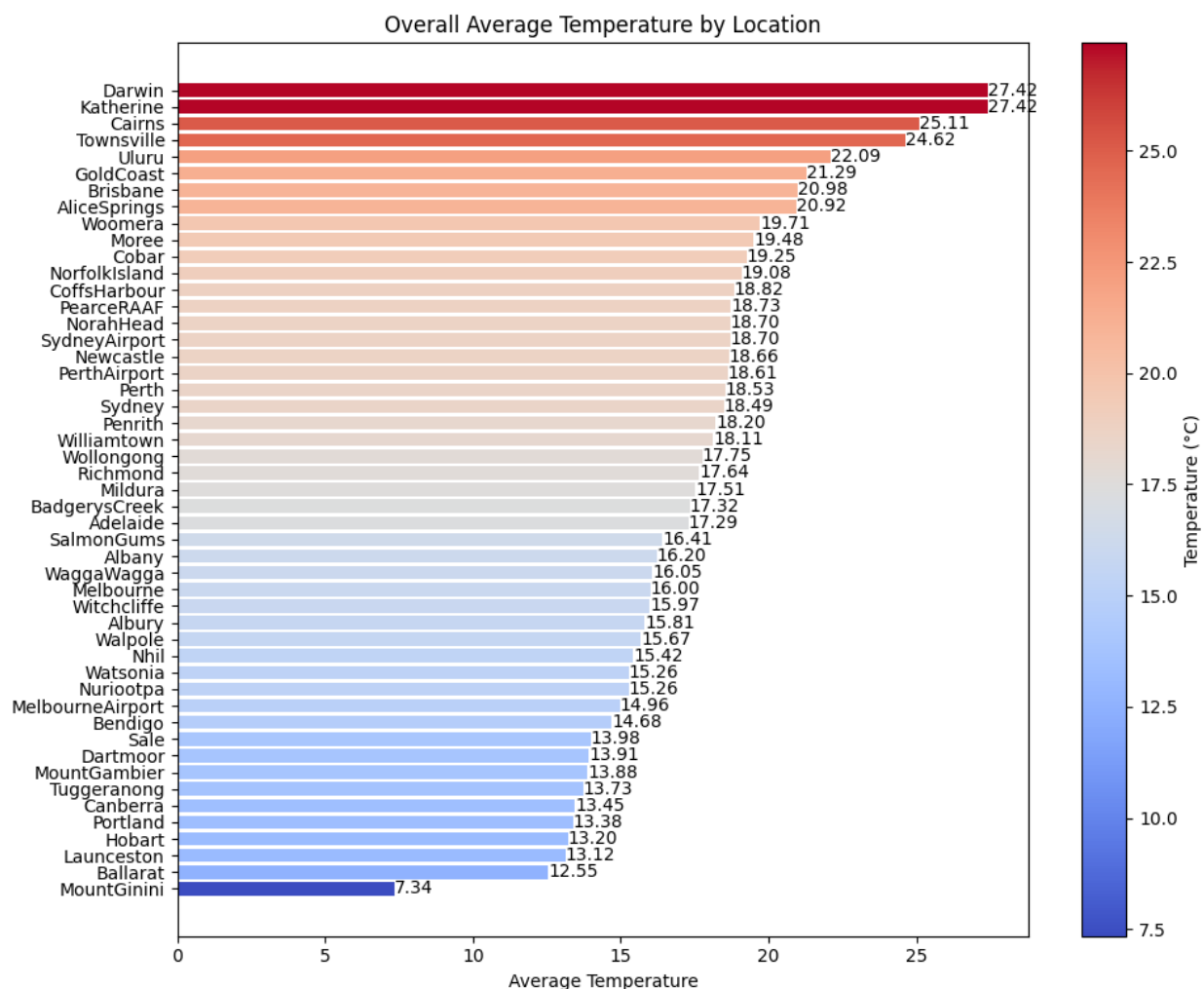
colors = cmap(norm(location_avg_temp_sorted['avg_temp'].values))

plt.figure(figsize=(10, 8))
bars = plt.barh(location_avg_temp_sorted['location'], location_avg_temp_sorted['avg_temp'], color=colors)
plt.xlabel('Average Temperature')
plt.title('Overall Average Temperature by Location')
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)

for bar, value in zip(bars, location_avg_temp_sorted['avg_temp']):
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2,
             f'{value:.2f}', va='center', ha='left')

plt.colorbar(cm.ScalarMappable(norm=norm, cmap=cmap), label='Temperature (°C)')
```


خروجی آن هم به صورت زیر شد :



۳. خب ابتدا لازم است که ماکسیمم هر روز را بدست بیاوریم من ابتدا فک میکردم باید بین speed ها بگردیم و ماکسیمم بگیریم که دیدم نه یک ستون مخصوص بیشترین باد هر روز داریم به اسم `windgustspeed`. سپس ایستگاه و همین متغیر را جدا میکنیم، سپس سورت میکنیم و پنج تای بیشتر آن را بر میداریم سپس برای نشان دادن از اسم ایستگاه و مقدار آن و نمودار میله ای کمک گرفتیم، در ادامه ابتدا کد مربوطه سپس خروجی را مشاهده میکنیم :

```

DM_HW1.ipynb
File Edit View Insert Runtime Tools Help

Files
[x]
sample_data
weatherAUS
combined_data.xlsx
weatherAUS.zip

+ Code + Text
if 'windgustspeed' in df.columns:
    wind_speeds_long = df[['location', 'windgustspeed']].copy()
    wind_speeds_long.rename(columns={'windgustspeed': 'SpeedValue'}, inplace=True)
    wind_speeds_long['windSpeedCol'] = 'windgustspeed'

    top_5_speeds = wind_speeds_long.sort_values(by='SpeedValue', ascending=False).head(5)

    plt.figure(figsize=(10, 6))
    colors = ['skyblue', 'orange', 'lightgreen', 'pink', 'yellow']

    bars = plt.bar([f"{row['location']} ({row['windSpeedCol']})" for index, row in top_5_speeds.iterrows()],
                    top_5_speeds['SpeedValue'],
                    color=[colors[i % len(colors)] for i in range(len(top_5_speeds))])

    plt.xlabel('Location (Wind Speed Column)')
    plt.ylabel('Wind Speed')
    plt.title('Top 5 Wind Gust Speeds')

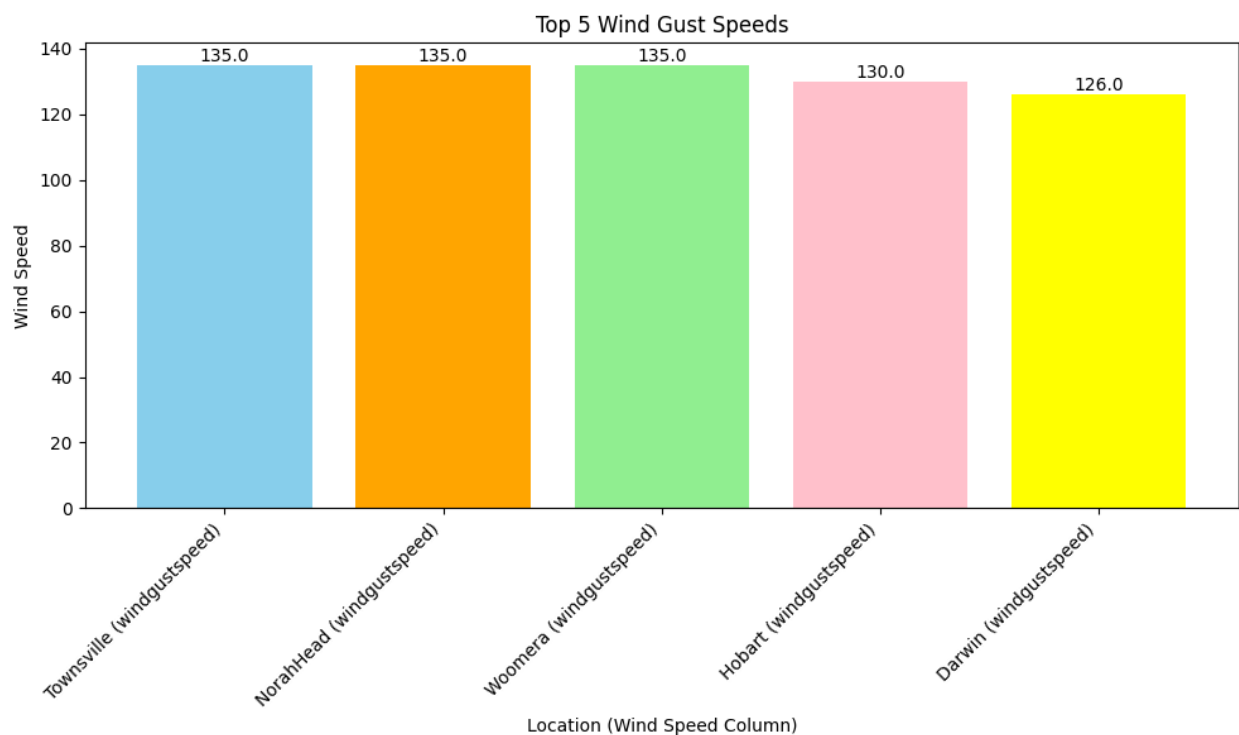
    plt.xticks(rotation=45, ha="right")

    for bar in bars:
        height = bar.get_height()
        plt.text(bar.get_x() + bar.get_width() / 2.0, height, f'{height}', ha='center', va='bottom')

    plt.tight_layout()
    plt.show()
else:
    print("The 'windgustspeed' column does not exist in the DataFrame.")

0s completed at 3:50 PM

```



۴. برای شناسایی این مورد از scatter plot استفاده میکنیم، بدین نحو که برای هر روز میانگین دمایی که خودمان حساب کردیم را در مقابل مقدار sunshine رسم میکنیم که خروجی و کد به صورت زیر میشود همچنین یک correlation matrix محاسبه میکنیم که مقدار عددی این وابستگی هم بدست بیاید :

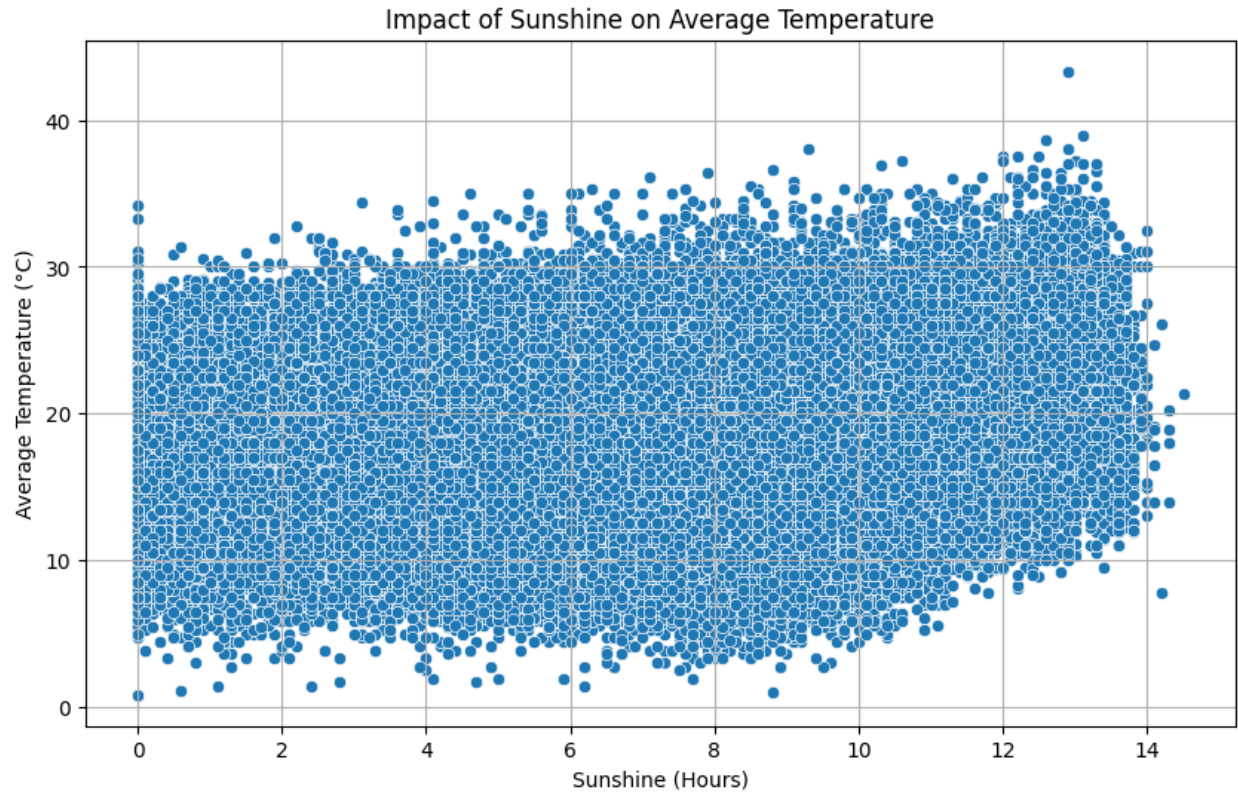
```

import seaborn as sns

plt.figure(figsize=(10, 6))
sns.scatterplot(x='sunshine', y='avg_temp', data=df)
plt.title('Impact of Sunshine on Average Temperature')
plt.xlabel('Sunshine (Hours)')
plt.ylabel('Average Temperature (°C)')
plt.grid(True)
plt.show()

correlation_matrix = df.corr()
correlation_matrix

```



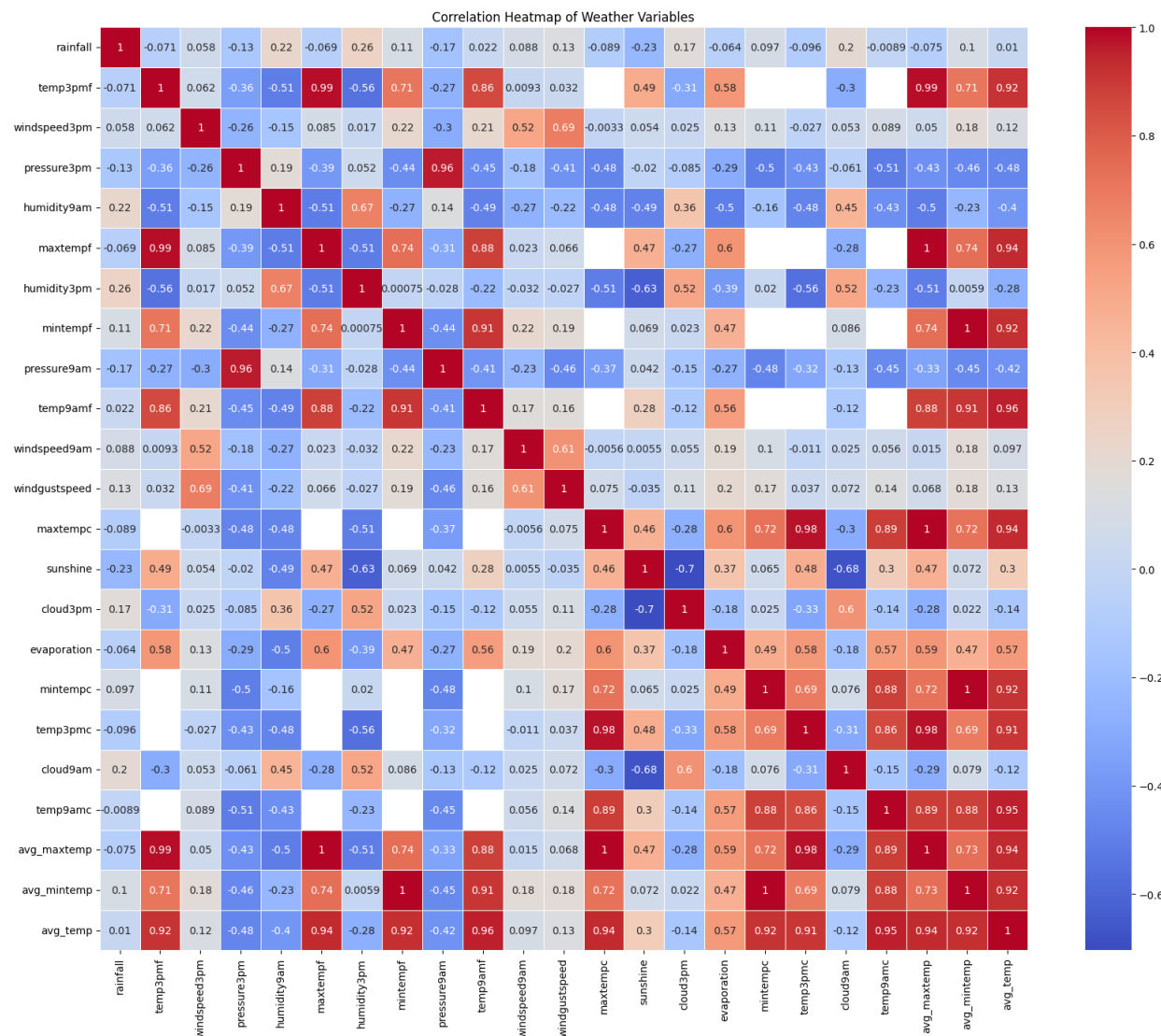
| | rainfall | temp3pmf | windspeed3pm | pressure3pm | humidity9am | maxtempf | humidity3pm | mintempf | pressure9am | temp9amf | ... | sunshi |
|---------------|-----------|-----------|--------------|-------------|-------------|-----------|-------------|-----------|-------------|-----------|-----|---------|
| rainfall | 1.000000 | -0.071414 | 0.057887 | -0.126548 | 0.223918 | -0.069287 | 0.255379 | 0.107409 | -0.168154 | 0.021628 | ... | -0.2275 |
| temp3pmf | -0.071414 | 1.000000 | 0.061835 | -0.361844 | -0.505558 | 0.985661 | -0.556331 | 0.713673 | -0.266823 | 0.857489 | ... | 0.4921 |
| windspeed3pm | 0.057887 | 0.061835 | 1.000000 | -0.255487 | -0.145496 | 0.084983 | 0.016662 | 0.220104 | -0.296381 | 0.211486 | ... | 0.0538 |
| pressure3pm | -0.126548 | -0.361844 | -0.255487 | 1.000000 | 0.186879 | -0.393637 | 0.051989 | -0.438189 | 0.961327 | -0.446175 | ... | -0.0196 |
| humidity9am | 0.223918 | -0.505558 | -0.145496 | 0.186879 | 1.000000 | -0.513845 | 0.667476 | -0.265932 | 0.139463 | -0.488465 | ... | -0.4908 |
| maxtempf | -0.069287 | 0.985661 | 0.084983 | -0.393637 | -0.513845 | 1.000000 | -0.511981 | 0.738645 | -0.306853 | 0.880607 | ... | 0.4729 |
| humidity3pm | 0.255379 | -0.556331 | 0.016662 | 0.051989 | 0.667476 | -0.511981 | 1.000000 | 0.000746 | -0.027527 | -0.215329 | ... | -0.6291 |
| mintempf | 0.107409 | 0.713673 | 0.220104 | -0.438189 | -0.265932 | 0.738645 | 0.000746 | 1.000000 | -0.436559 | 0.910442 | ... | 0.0690 |
| pressure9am | -0.168154 | -0.266823 | -0.296381 | 0.961327 | 0.139463 | -0.306853 | -0.027527 | -0.436559 | 1.000000 | -0.406697 | ... | 0.0420 |
| temp9amf | 0.021628 | 0.857489 | 0.211486 | -0.446175 | -0.488465 | 0.880607 | -0.215329 | 0.910442 | -0.406697 | 1.000000 | ... | 0.2785 |
| windspeed9am | 0.088281 | 0.009279 | 0.519516 | -0.175807 | -0.270128 | 0.023129 | -0.031561 | 0.216906 | -0.228729 | 0.167279 | ... | 0.0055 |
| windrustspeed | 0.133701 | 0.031977 | 0.686465 | -0.413732 | -0.215077 | 0.066362 | -0.026542 | 0.187851 | -0.458779 | 0.163940 | ... | 0.0347 |

که مقدار آن هم برابر ۰.۳۰۱۶۰۹ است که یک وابستگی نسبتاً ضعیف را بیان میکند.

۵. خوبی این سوال این است که اصل کار را در مرحله قبل انجام دادیم حالا میتوانیم آن را با heatmap نشان دهیم که کد و خروجی به صورت زیر است :

```
correlation_matrix = df.corr()

plt.figure(figsize=(20, 16))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)
plt.title('Correlation Heatmap of Weather Variables')
plt.show()
```



۶. در این قسمت با توجه به زیاد بودن داده های صفر و خیلی کوچک که کلا نمودار را خالی نشان میدادند لازم بود تا یک سری پیش پردازش انجام دهیم، سپس سه نمودار را به کمک sns رسم میکنیم :

```
df_large_nonzero = df[df['rainfall'] > 1]

rainfall_nonzero_descriptive = df_large_nonzero['rainfall'].describe()

fig, axs = plt.subplots(3, 1, figsize=(10, 12))

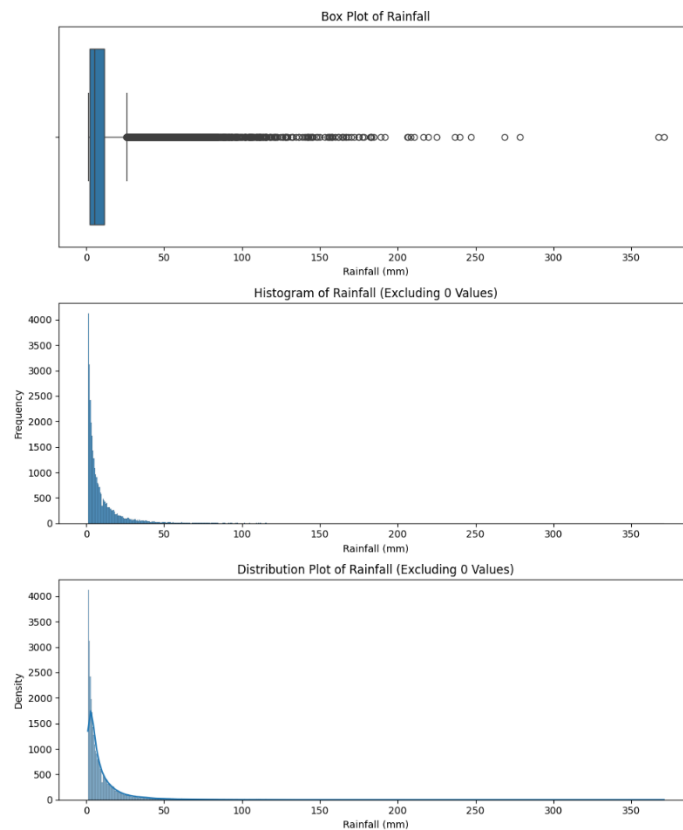
sns.boxplot(x=df_large_nonzero['rainfall'], ax=axs[0])
axs[0].set_title('Box Plot of Rainfall')
axs[0].set_xlabel('Rainfall (mm)')

sns.histplot(df_large_nonzero['rainfall'], kde=False, ax=axs[1])
axs[1].set_title('Histogram of Rainfall (Excluding 0 Values)')
axs[1].set_xlabel('Rainfall (mm)')
axs[1].set_ylabel('Frequency')

sns.histplot(df_large_nonzero['rainfall'], kde=True, ax=axs[2])
axs[2].set_title('Distribution Plot of Rainfall (Excluding 0 Values)')
axs[2].set_xlabel('Rainfall (mm)')
axs[2].set_ylabel('Density')

plt.tight_layout()
plt.show()

rainfall_nonzero_descriptive
```

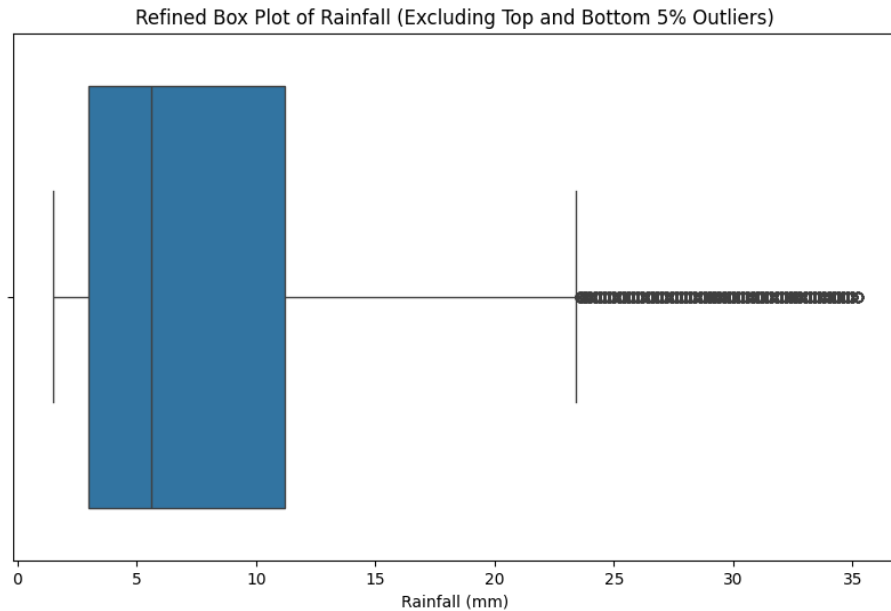


در گام بعدی برای واضح تر شدن طبق خواسته صورت سوال داده های پرت را حذف میکنیم :

```
rainfall_refined = df_large_nonzero['rainfall'].quantile([0.05, 0.95])

df_large_refined = df_large_nonzero[(df_large_nonzero['rainfall'] > rainfall_refined.loc[0.05]) & (df_large_nonzero['rainfall'] < rainfall_refined.loc[0.95])]

plt.figure(figsize=(10, 6))
sns.boxplot(x=df_large_refined['rainfall'])
plt.title('Refined Box Plot of Rainfall (Excluding Top and Bottom 5% Outliers)')
plt.xlabel('Rainfall (mm)')
plt.show()
```

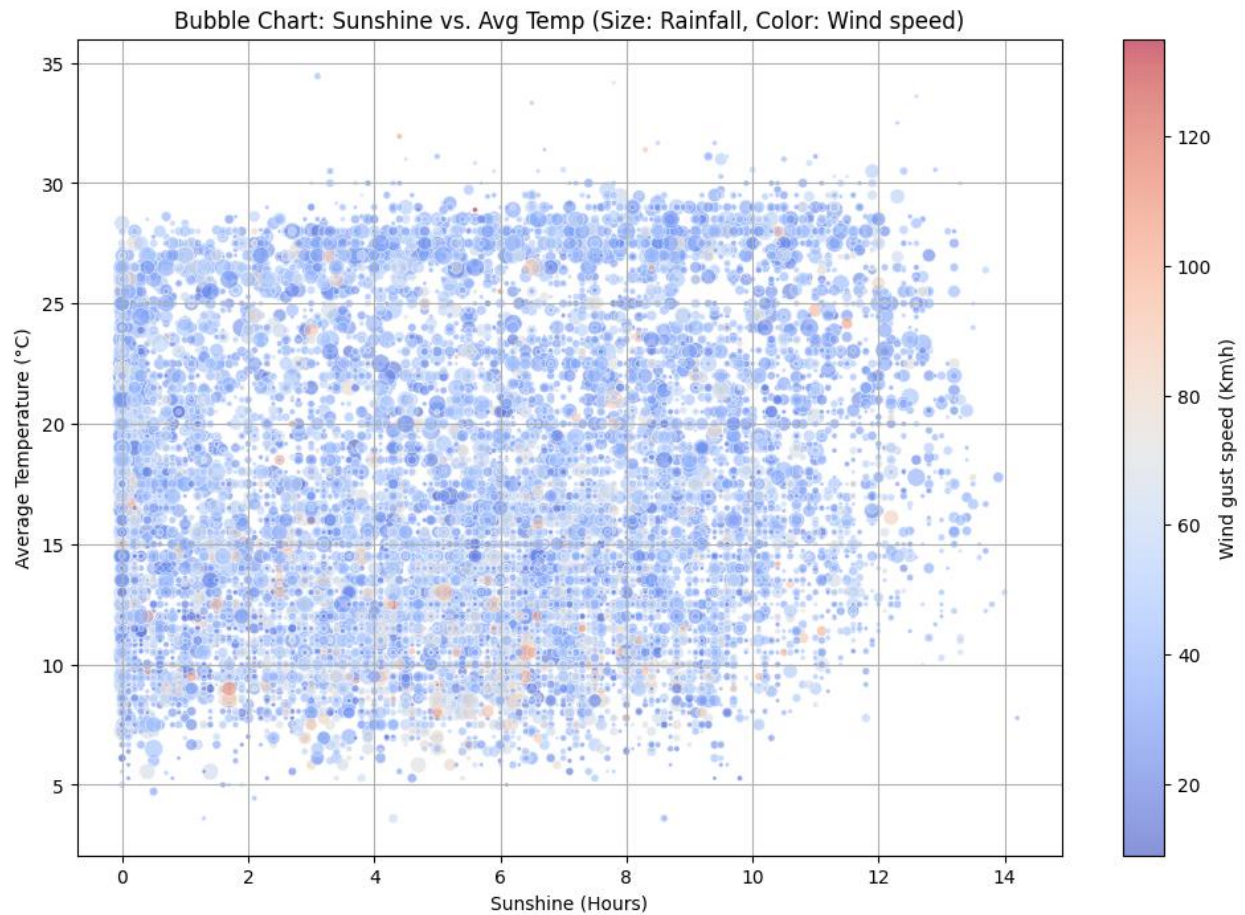


در کل میتوان گفت با توجه به تجمع زیاد داده در حوالی صفر استفاده از نموداری که بتواند شیوا باشد کار سختی است، ولی در بین این نمودار همین نمودار جعبه ای بعد از اصلاح از همه واضح تر است و بهتر میتواند پخش دیتا و توزیع آن را بیان کند .

۷. بله برای بیشتر از دو میتوان از نمودار حبابی استفاده کرد، که متغیر سوم سایز حباب را نشان میدهد همچنین میتوان چهار ویژگی کرد و چهارمی را به کمک رنگ حباب نشان داد. ما در اینجا سعی کردیم تا چهار متغیر مقدار بارش، حداکثر سرعت باد، آفتاب و میانگین دما را در یک scatter نمایش دهیم.

```
df_plot = df_large_nonzero[(df_large_nonzero['rainfall'] > df_large_nonzero['rainfall'].quantile(0.05)) &
(df_large_nonzero['rainfall'] < df_large_nonzero['rainfall'].quantile(0.95))]

plt.figure(figsize=(12, 8))
plt.scatter(df_plot['sunshine'], df_plot['avg_temp'],
            s=df_plot['rainfall']*3,
            c=df_plot['windgustspeed'], cmap='coolwarm', alpha=0.6, edgecolors="w", linewidth=0.5)
plt.title('Bubble Chart: Sunshine vs. Avg Temp (Size: Rainfall, color: Wind speed)')
plt.xlabel('Sunshine (Hours)')
plt.ylabel('Average Temperature (°C)')
plt.colorbar(label='Wind gust speed (Km/h)')
plt.grid(True)
plt.show()
```

۸. کد این قسمت بدین صورت است که ابتدا یک ایستگاه را انتخاب میکنیم، سپس نمودار خطی موارد خواسته شده را رسم میکنیم که پیچیدگی خاصی ندارد (تنها مشکل این بود برخی ایستگاه ها تبخیر نداشتند که باید این مورد را در نظر گرفت).

```
location_to_analyze = df['location'].unique()[15]

df_filtered = df[df['location'] == location_to_analyze]

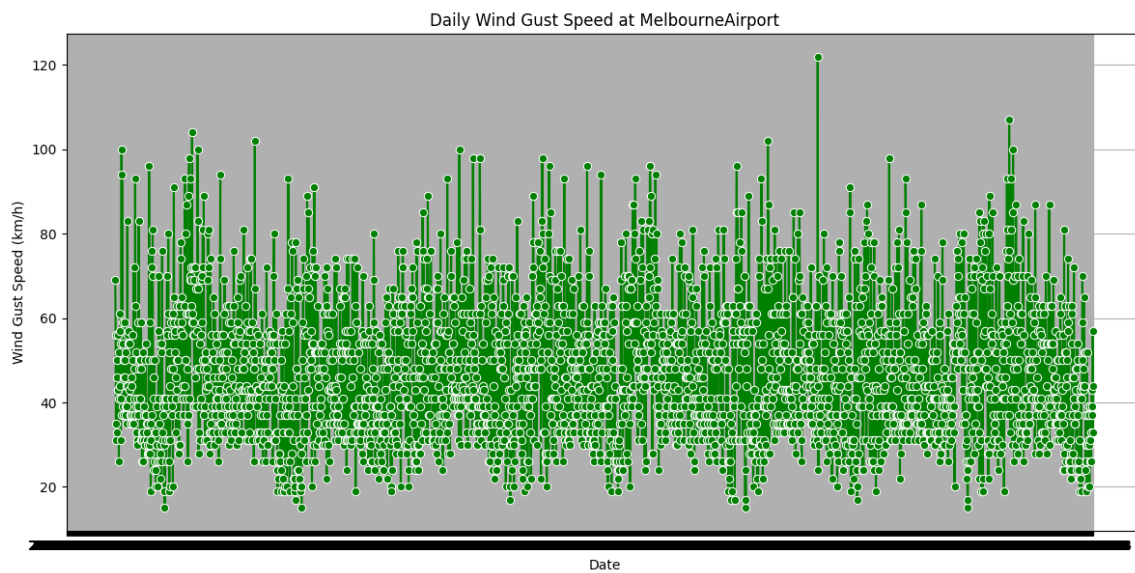
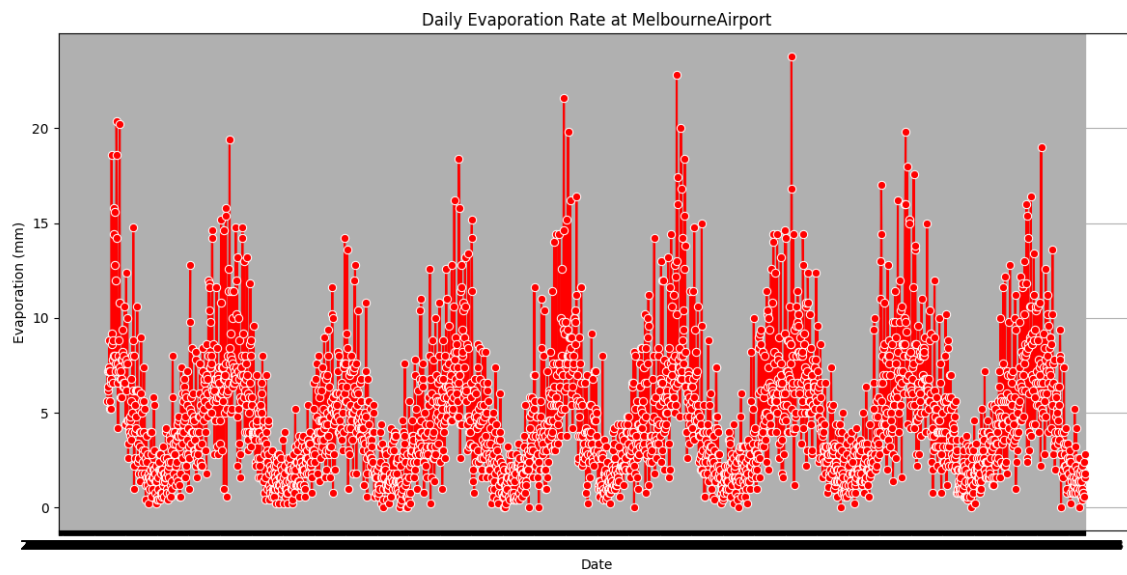
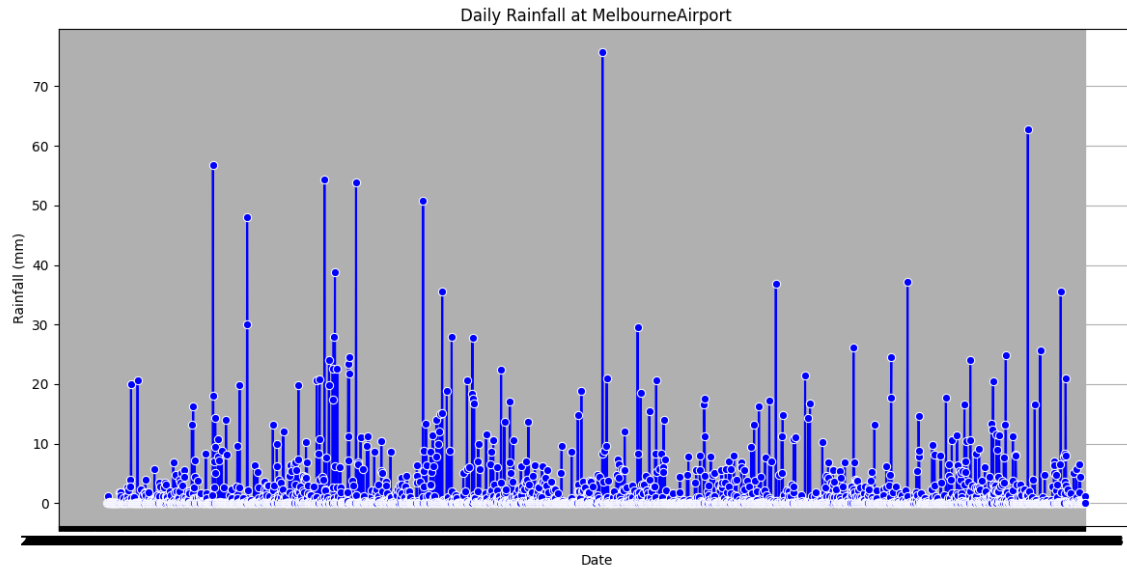
fig, axs = plt.subplots(3, 1, figsize=(12, 18))

sns.lineplot(x='date', y='rainfall', data=df_filtered, ax=axs[0], marker='o', color='blue')
axs[0].set_title(f'Daily Rainfall at {location_to_analyze}')
axs[0].set_xlabel('Date')
axs[0].set_ylabel('Rainfall (mm)')
axs[0].grid(True)

sns.lineplot(x='date', y='evaporation', data=df_filtered, ax=axs[1], marker='o', color='red')
axs[1].set_title(f'Daily Evaporation Rate at {location_to_analyze}')
axs[1].set_xlabel('Date')
axs[1].set_ylabel('Evaporation (mm)')
axs[1].grid(True)

sns.lineplot(x='date', y='windgustspeed', data=df_filtered, ax=axs[2], marker='o', color='green')
axs[2].set_title(f'Daily Wind Gust Speed at {location_to_analyze}')
axs[2].set_xlabel('Date')
axs[2].set_ylabel('Wind Gust Speed (km/h)')
axs[2].grid(True)

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



۹. ابتدا کوانتایل ها را طبق خواسته سوال تعریف میکنیم، سپس یک فانکشن برای اساین کردن هر کتگوری تعریف کردیم، در آخر بر مبنای لوکیشن و کتگوری گروپ کردیم و بیشترین را برای هر کدام پرینت کردیم :

```
quantiles = df['avg_temp'].quantile([0.1, 0.3, 0.7, 0.9])

def categorize_temperature_adjusted(temp):
    if temp <= quantiles[0.1]:
        return 'Very cold days'
    elif temp <= quantiles[0.3]:
        return 'Cold days'
    elif temp <= quantiles[0.7]:
        return 'Moderate days'
    elif temp <= quantiles[0.9]:
        return 'Warm days'
    else:
        return 'Very warm days'

df['Temperature Category'] = df['avg_temp'].apply(categorize_temperature_adjusted)

category_counts = df.groupby(['location', 'Temperature Category']).size().unstack(fill_value=0)
highest_days_by_category = category_counts.idxmax()

highest_days_by_category
```

که خروجی هم به صورت زیر شد :

The screenshot shows a Google Colab notebook interface. The code cell contains the same Python code as shown in the previous block. The output cell displays the results of the `highest_days_by_category` variable, which is a pandas Series. The output is as follows:

```
Temperature Category
Cold days          Portland
Moderate days      NorfolkIsland
Very cold days     MountGinini
Very warm days     Darwin
Warm days          Cairns
dtype: object
```