India

October 2, 2022

```
import pandas as pd
[2]:
     import numpy as np
[3]:
     import warnings
[4]:
     import matplotlib.pyplot as plt
[5]:
     import seaborn as sns
[6]:
    df = pd.read_csv(r'C:\Users\Damian\Desktop\Data_Delhi.csv')
    time - czas kiedy zostały wykonane pomiary, tavg - średnia temperatura wyrażana w stopniach
    Celciusza, tmin - minimalna temperatura wyrażana w stopniach Celciusza, tmax - maksymalna
    temperatura wyrażana w stopniach Celciusza, prcp - opady wyrażane w milimetrach.
[7]: df
[7]:
                   time
                          tavg
                                tmin
                                       tmax
                                             prcp
     0
             01-01-1990
                           9.4
                                 6.0
                                       15.1
                                              0.0
                           9.3
                                       14.2
     1
             02-01-1990
                                 5.2
                                              0.0
     2
             03-01-1990
                           9.0
                                 6.5
                                       13.6
                                              0.0
     3
             04-01-1990
                          10.7
                                  6.0
                                       17.5
                                               0.0
     4
                          12.6
                                  7.3
                                       20.8
             05-01-1990
                                               0.0
             21-07-2022
                                       30.6
                                              21.2
     11889
                          28.6
                                26.8
                                27.0
     11890
             22-07-2022
                          29.3
                                       32.9
                                              0.3
     11891
             23-07-2022
                          30.1
                                25.5
                                       34.9
                                              8.9
                                27.1
     11892
             24-07-2022
                          30.6
                                       35.7
                                               0.0
     11893
             25-07-2022
                          30.7
                                26.8
                                       35.7
                                              0.0
     [11894 rows x 5 columns]
[8]:
     df.describe()
[8]:
                                     tmin
                                                    tmax
                     tavg
                                                                  prcp
             11800.000000
                            10358.000000
                                           11361.000000
                                                           5754.000000
     count
                               18.877563
                                               31.788232
                24.996415
                                                              3.662478
     mean
     std
                 7.326116
                                7.786944
                                                6.838998
                                                             12.473498
```

```
min
                  6.600000
                                 0.100000
                                                9.800000
                                                               0.000000
      25%
                 18.500000
                                11.800000
                                                26.700000
                                                               0.000000
      50%
                 27.000000
                                20.000000
                                                33.200000
                                                               0.000000
      75%
                 30.900000
                                26.000000
                                                36.600000
                                                               0.500000
                 39.800000
                                34.200000
                                                48.100000
                                                             262.900000
      max
 [9]: df.isnull().sum()
 [9]: time
                  0
      tavg
                 94
      tmin
               1536
      tmax
                533
               6140
      prcp
      dtype: int64
[10]: %%capture --no-display
      df['time'] = pd.to_datetime(df['time'],infer_datetime_format=True) #Metoda_\_
        → % capture pozwala zignorować ostrzeżenia o formacie daty w poniższej linii u
        \hookrightarrow kodu
[11]: df.set_index(['time'], inplace = True) #Ustawienie czasu jako index
[12]: df_zero = df.dropna()
[13]: missing_data = round((df_zero.shape[0]/df.shape[0])*100,2)
[14]: print(missing_data,'% danych zawiera NaN wartości')
     39.73 % danych zawiera NaN wartości
[15]: df[df.isna().sum(axis=1) > 3]
[15]:
                   tavg
                         tmin tmax prcp
      time
                                        NaN
      1990-09-27
                    {\tt NaN}
                           NaN
                                 NaN
      1990-09-28
                    NaN
                           NaN
                                 NaN
                                        NaN
      1990-10-20
                    NaN
                           NaN
                                 NaN
                                        NaN
      1990-10-21
                    NaN
                                        NaN
                           NaN
                                 NaN
      1990-10-22
                    NaN
                           NaN
                                 NaN
                                        NaN
      1999-04-30
                           NaN
                                 {\tt NaN}
                                        NaN
                    NaN
      2001-04-28
                    NaN
                           NaN
                                 NaN
                                        NaN
      2002-10-26
                    NaN
                           {\tt NaN}
                                        NaN
                                 NaN
      2010-04-17
                    NaN
                           NaN
                                 {\tt NaN}
                                        NaN
      2010-04-18
                    NaN
                           NaN
                                 NaN
                                        NaN
      [94 rows x 4 columns]
```

94 wierszy nie posiada żadnych wartości w każdej kolumnie

```
[16]: df = df.dropna(how='all') #usunięcie wszystkich wierszy nieposiadającychu
       ⇒żadnych wartości
[17]: df = df.interpolate(method='time') #wykorzystanie funkcji interpolate dou
       →wypełnienia pozostałych brakujących danych, danymi szacunkowymi
[18]: df
[18]:
                 tavg tmin tmax prcp
      time
      1990-01-01
                  9.4
                        6.0
                             15.1
                                    0.0
      1990-02-01
                  9.3
                        5.2
                             14.2
                                    0.0
      1990-03-01
                  9.0
                        6.5
                             13.6
                                    0.0
      1990-04-01 10.7
                             17.5
                        6.0
                                    0.0
      1990-05-01 12.6
                        7.3 20.8
                                    0.0
     2022-07-21 28.6
                       26.8 30.6
                                   21.2
      2022-07-22 29.3
                       27.0
                             32.9
                                    0.3
      2022-07-23 30.1
                       25.5 34.9
                                    8.9
      2022-07-24 30.6
                       27.1
                             35.7
                                    0.0
      2022-07-25 30.7
                       26.8
                            35.7
                                    0.0
      [11800 rows x 4 columns]
[19]: df.isna().sum()
[19]: tavg
      tmin
             0
      tmax
             0
             0
     prcp
     dtype: int64
[20]: df[df['tmax'] == max(df['tmax'])] #maksymalna temperatura
[20]:
                 tavg tmin tmax prcp
      time
      2017-04-06 39.8 30.4 48.1
                                    1.0
[21]: df[df['tmin'] == min(df['tmin'])] #minimalna temperatura
[21]:
                 tavg tmin tmax prcp
      time
      2003-07-06 36.6
                        0.1 42.2
                                    0.0
[22]: df[df['prcp'] == max(df['prcp'])] #maksymalne opady
[22]:
                 tavg
                            tmin tmax
                                         prcp
      time
```

```
[24]: df_years = df.groupby(df.index.year).mean()
df_years
```

[24]:		tavg	tmin	tmax	prcp
	time				
	1990	24.755587	19.171572	31.125024	2.193171
	1991	24.778933	18.833806	31.525796	1.752107
	1992	24.065273	17.474069	30.730890	2.350364
	1993	24.824242	18.815886	31.517906	2.644628
	1994	24.693906	18.455956	31.367036	2.815028
	1995	24.965659	19.193269	31.273077	2.321291
	1996	24.320765	18.228962	31.102459	3.292350
	1997	23.587123	18.048219	29.647123	1.368630
	1998	24.534521	18.900137	30.908767	2.440822
	1999	24.974033	19.140331	31.950829	2.027762
	2000	24.769672	18.976639	31.519262	4.347678
	2001	24.524176	18.751740	31.524038	3.285837
	2002	25.329396	19.424313	32.220879	2.809159
	2003	24.373699	18.840822	30.933973	3.252329
	2004	25.155464	19.188115	31.855738	1.511885
	2005	24.754521	18.752603	31.626438	2.649543

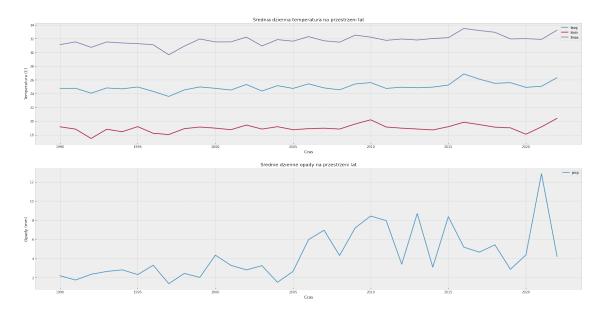
```
2006 25.420548 18.898356
                                32.291233
                                            5.969856
     2007 24.830959 18.967808 31.684795
                                            6.953948
     2008 24.551913 18.842896
                                31.484563
                                            4.310187
     2009 25.403014 19.579315
                                32.503288
                                            7.188017
     2010 25.601102 20.191322 32.213912
                                            8.432605
     2011 24.755890 19.140548 31.748630
                                            7.954064
     2012 24.933880 18.972131 31.947951
                                            3.397723
     2013 24.856986 18.849315 31.797808
                                            8.695342
     2014 24.944110 18.714932 32.019041
                                            3.088265
     2015 25.242466 19.185753
                                32.132192
                                            8.364217
     2016 26.864481 19.835792
                                33.478142
                                            5.180753
     2017 26.100822 19.492877
                                33.176986
                                            4.661201
     2018 25.499452 19.120959
                                32.922740
                                            5.429452
     2019 25.592877 19.020000
                                31.954795
                                            2.870685
     2020 24.910109 18.090984
                                31.992623
                                            4.373419
     2021 25.070959 19.163288
                                31.894521
                                           12.855558
     2022 26.311165 20.399515
                                33.227184
                                            4.230510
[25]: f, axes = plt.subplots(nrows=2, figsize=(30,15))
     ax = df_years[['tavg', 'tmin', 'tmax']].plot(ax=axes[0])
     ax.set_ylabel('Temperatura (C)')
     ax.set xlabel('Czas')
     ax.set_title('Średnia dzienna temperatura na przestrzeni lat')
     ax = df_years[['prcp']].plot(ax=axes[1])
```

[25]: Text(0.5, 1.0, 'Średnie dzienne opady na przestrzeni lat')

ax.set_title('Średnie dzienne opady na przestrzeni lat')

ax.set_ylabel('Opady (mm)')

ax.set_xlabel('Czas')

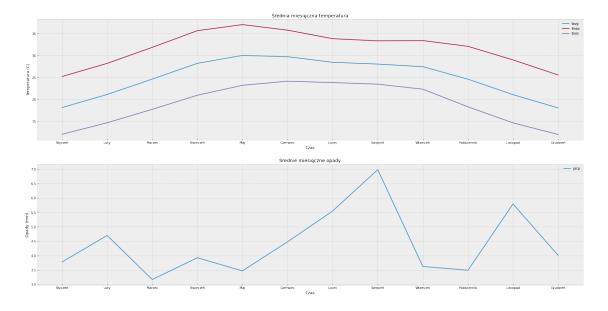


Z wykresu średniej rocznej temperatury zauważyć można nieznaczne wachania temperatury od 1990 roku z niewielką tendencją wzrostową po roku 2020

Średnie roczne opady od roku 1990 do roku 2005 wachały sie między 2-4mm. Po roku 2005 można zauważyć dynamiczny wzrost ilości opadów, mieszący się w przedziale 4-8mm do roku 2020, po którym nastąpił gwałtowny wzrost średniej ilości opadów.

```
[26]: months=['Styczeń','Luty','Marzec','Kwiecień','Maj','Czerwiec','Lipiec','Sierpień','Wrzesień','
       →#wizualizacja danych z podziałem na miesiące
      df_months = df.groupby(df.index.month,__
       →as_index=False)[['tavg','tmin','tmax','prcp']].mean()
      f, axes = plt.subplots(nrows=2, figsize=(30,15))
      ax = df_months[['tavg','tmax','tmin']].plot(ax=axes[0])
      ax.set_ylabel('Temperatura (C)')
      ax.set_xlabel('Czas')
      ax.set_xticks(np.arange(0,12))
      ax.set_xticklabels(months)
      ax.set_title('Średnia miesięczna temperatura')
      ax = df_months[['prcp']].plot(ax=axes[1])
      ax.set_ylabel('Opady (mm)')
      ax.set_xlabel('Czas')
      ax.set_xticks(np.arange(0,12))
      ax.set_xticklabels(months)
      ax.set_title('Średnie miesięczne opady')
```

[26]: Text(0.5, 1.0, 'Średnie miesięczne opady')



```
[27]: df['year'] = df.index.year
    df['month'] = df.index.month

[28]: heatmap = df.groupby(['year','month'], as_index=False)[['tavg','prcp']].mean()

[29]: data = heatmap.pivot('year','month','tavg')
    data.columns = months

    plt.subplots(figsize=(15,15))
    sns.heatmap(data, cmap='YlOrBr', annot=True, fmt='.2f')
    plt.title('Średnia dzienna temperatura (C) w danym miesiącu')
    plt.yticks='auto'
    plt.show()
```

			Śre	ednia dz	ienna te	emperat	ura (C)	w danyr	n miesi	ącu				
1990	- 20.01	18.74	22.82	27.77	29.65	29.62	27.93	27.75	26.71	24.29	21.28	19.07		32
1991	- 18.35	20.73		27.21	29.57	30.08	29.04	27.03	27.69	24.10	20.31	18.14		
1992	- 18.96	19.01	23.73	27.38	29.16	30.01	26.60	26.81	26.51	23.58	20.48	17.87		
1993	- 17.89	21.31	22.61	27.96	30.16	29.57	28.27	29.73	27.32	23.60	21.05	18.18		
1994	- 18.69	20.85	25.64	27.22	30.62	30.08	27.38	27.33	26.44	23.82	20.65	18.06		30
1995	- 17.82	20.26	23.92	27.65	30.94	30.20	28.47	27.26	28.09	24.52	21.25	18.95		
1996	- 18.02	20.80	24.49	27.88	28.94	28.12	28.12	26.79	27.28	23.86	19.83	17.68		
1997	- 16.55	20.00	22.33	26.33	28.41	28.39	27.91	27.41	27.11	22.80	19.82	15.87		
1998	- 17.55	20.70	22.98	28.29	31.74	29.46	28.65	26.94	26.74	23.93	20.57	16.73		- 28
1999	- 17.92	21.54	24.65	27.87	29.47	29.95	28.18	28.85	27.87		21.30	17.02		
2000	- 18.58	19.50	23.44	29.25	31.08	29.11	26.96	27.46	27.44	24.89	21.02	18.42		
2001	- 17.71	21.34		26.59	28.45	28.14	27.61	28.38	27.56		21.06	18.16		25
2002	- 18.11	21.32		28.68	30.03	29.99	30.72	28.21	26.91		21.08	18.93		- 26
2003	- 17.00	21.06	24.35	28.10	29.71	28.93	27.58	27.62	26.93	24.06	19.51	17.47		
2004	- 17.95	21.68	26.01	28.14	30.30	28.66	29.63	27.54	27.32	23.77	21.93	18.89		
2005	- 17.86	20.25	24.93	27.59	30.26	30.16	28.29	29.15	26.48	23.84	20.82	17.20		- 24
2006	- 19.31	23.90	23.85	28.17	30.11	29.30	28.81	28.82	27.75		21.48	18.62		24
	- 18.76	20.39	23.92	28.91	29.53	28.53	28.66	28.41	27.34		21.08	17.70		
2008	- 17.09	20.84	25.55	28.57	27.25	27.64	28.25	27.90	26.65		20.96	18.92		- 22
2009	- 19.71	22.06	24.77	29.32	30.03	31.34	29.35	27.87	27.88	24.02	20.36	18.04		
2010	- 18.45	21.74	27.02	30.07	31.47	31.27	28.51	27.44	26.58		21.62	18.19		
2011	- 18.29	20.32	25.19	27.39	29.86	29.08	28.51	27.69	27.27		21.89	17.07		
2012	- 17.64	20.76	24.76	27.09	31.38	32.18	28.98	27.08	26.91	23.98	20.16	18.23		
2013	- 17.93	20.53	24.38	27.79	30.94	28.60	28.12	27.65	27.71	25.26	20.82	18.32		- 20
2014	- 18.45	19.76	24.07	27.62	29.30	30.78	29.32	29.45	28.37	24.72	20.63	16.60		
2015	- 17.61	22.90		27.77	30.49	28.93	28.29	28.77	28.11		21.67	18.60		
2016	- 19.43	23.55	26.78	31.08	31.85	31.18	29.09	29.09	29.65	26.86	23.41	20.44		
2017	- 19.39	23.12	26.59	30.80	31.21	29.92	29.14	29.05	28.40		20.63	19.07		- 18
2018	- 18.78	22.26	25.68	28.73	30.60	30.52	28.12	28.21	26.82		22.92	18.03		
2019	- 18.71	20.93	24.93	29.08	30.21	31.32	28.72	29.20	28.29	26.24	22.64	16.69		
2020	- 17.80	21.56	23.44	27.24	29.51	30.29	28.47	27.86	29.26	25.24	20.77	17.52		
2021	- 17.13	23.11	25.84	27.94	28.39	30.02	28.44	29.15	27.92	24.63	20.78	17.51		- 16
2022	- 15.27	20.18	27.20	30.40	30.97	30.53	29.39	27.36	26.66	26.47	26.59	26.60		
	Styczeń	Luty	Marzec	Kwiecień	Maj	Czerwiec	Lipiec	Sierpień	Wrzesień	Październik	Listopad	Grudzień		

Najwyższe średnie dzienne temperatury odnotowane zostały w miesiącach maj oraz czerwiec, natomiast najniższe w miesiącach grudzień oraz styczeń