ТМО ЛР1 ИУ5-63Б Горкунов Николай

5 июня 2024 г.

1 *ТМО ЛР1 ИУ5-63Б Горкунов Николай*

2 Создать ноутбук, который содержит следующие разделы:

- Текстовое описание выбранного Вами набора данных.
- Основные характеристики датасета.
- Визуальное исследование датасета.
- Информация о корреляции признаков.
- Сформировать отчет и разместить его в своем репозитории на github.

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

pd.set_option('display.max_columns', None)
```

[2]:
```
df = pd.read_csv(r"C:\Users\gorku\Downloads\austin_weather.csv")
df.head()
```

[2]:

	Date	TempHighF	TempAvgF	TempLowF	DewPointHighF	DewPointAvgF	DewPointLowF	HumidityHighPercent	HumidityAvgPercent	HumidityLowPercent	SeaLevelPressureHighInches	SeaLevelPressureAvgInches
0	2013-12-21	74	60	45	67	49	43	93	75	57	29.86	29.68
1	2013-12-22	56	48	39	43	36	28	93	68	43	30.41	30.13
2	2013-12-23	58	45	32	31	27	23	76	52	27	30.56	30.49
3	2013-12-24	61	46	31	36	28	21	89	56	22		
4	2013-12-25	58	50	41	44	40	36	86	71	56		

```
3                      30.56                    30.45
4                      30.41                    30.33

  SeaLevelPressureLowInches VisibilityHighMiles VisibilityAvgMiles  \
0                     29.59                  10                   7
1                     29.87                  10                  10
2                     30.41                  10                  10
3                      30.3                  10                  10
4                     30.27                  10                  10

  VisibilityLowMiles WindHighMPH WindAvgMPH WindGustMPH  \
0                  2          20          4          31
1                  5          16          6          25
2                 10           8          3          12
3                  7          12          4          20
4                  7          10          2          16

  PrecipitationSumInches              Events
0                   0.46  Rain , Thunderstorm
1                      0
2                      0
3                      0
4                      T
```

[3]:
```
df.dtypes
```

[3]:
```
Date                         object
TempHighF                     int64
TempAvgF                      int64
TempLowF                      int64
DewPointHighF                object
DewPointAvgF                 object
DewPointLowF                 object
HumidityHighPercent          object
HumidityAvgPercent           object
HumidityLowPercent           object
SeaLevelPressureHighInches   object
SeaLevelPressureAvgInches    object
SeaLevelPressureLowInches    object
VisibilityHighMiles          object
VisibilityAvgMiles           object
VisibilityLowMiles           object
WindHighMPH                  object
WindAvgMPH                   object
WindGustMPH                  object
PrecipitationSumInches       object
Events                       object
```

```
dtype: object
```

[4]:
```
df['Date'] = pd.to_datetime(df['Date'])
df['TempHighF'] = df['TempHighF'].astype('float64')
df['TempAvgF'] =df['TempAvgF'].astype('float64')
df['TempLowF'] = df['TempLowF'].astype('float64')
df['DewPointHighF'] = pd.to_numeric(df['DewPointHighF'], errors='coerce')
df['DewPointAvgF'] = pd.to_numeric(df['DewPointAvgF'], errors='coerce')
df['DewPointLowF'] = pd.to_numeric(df['DewPointLowF'], errors='coerce')
df['HumidityHighPercent'] = pd.to_numeric(df['HumidityHighPercent'],␣
 ↪errors='coerce').astype('Int64')
df['HumidityAvgPercent'] = pd.to_numeric(df['HumidityAvgPercent'],␣
 ↪errors='coerce').astype('Int64')
df['HumidityLowPercent'] = pd.to_numeric(df['HumidityLowPercent'],␣
 ↪errors='coerce').astype('Int64')
df['SeaLevelPressureHighInches'] = pd.
 ↪to_numeric(df['SeaLevelPressureHighInches'], errors='coerce')
df['SeaLevelPressureAvgInches'] = pd.to_numeric(df['SeaLevelPressureAvgInches'],␣
 ↪errors='coerce')
df['SeaLevelPressureLowInches'] = pd.to_numeric(df['SeaLevelPressureLowInches'],␣
 ↪errors='coerce')
df['VisibilityHighMiles'] = pd.to_numeric(df['VisibilityHighMiles'],␣
 ↪errors='coerce')
df['VisibilityAvgMiles'] = pd.to_numeric(df['VisibilityAvgMiles'],␣
 ↪errors='coerce')
df['VisibilityLowMiles'] = pd.to_numeric(df['VisibilityLowMiles'],␣
 ↪errors='coerce')
df['WindHighMPH'] = pd.to_numeric(df['WindHighMPH'], errors='coerce')
df['WindAvgMPH'] = pd.to_numeric(df['WindAvgMPH'], errors='coerce')
df['WindGustMPH'] = pd.to_numeric(df['WindGustMPH'], errors='coerce')
df['PrecipitationSumInches'] = pd.to_numeric(df['PrecipitationSumInches'].
 ↪replace('T', '0'), errors='coerce')  # PrecipitationSumInches (Total␣
 ↪precipitation, in inches) ('T' if Trace) - из описания датасета
df['Events'] = df['Events'].astype(str)
df.dtypes
```

[4]:
```
Date                          datetime64[ns]
TempHighF                            float64
TempAvgF                             float64
TempLowF                             float64
DewPointHighF                        float64
DewPointAvgF                         float64
DewPointLowF                         float64
HumidityHighPercent                    Int64
HumidityAvgPercent                     Int64
HumidityLowPercent                     Int64
```

```
SeaLevelPressureHighInches           float64
SeaLevelPressureAvgInches            float64
SeaLevelPressureLowInches            float64
VisibilityHighMiles                  float64
VisibilityAvgMiles                   float64
VisibilityLowMiles                   float64
WindHighMPH                          float64
WindAvgMPH                           float64
WindGustMPH                          float64
PrecipitationSumInches               float64
Events                                object
dtype: object
```

[5]:
```
df.head()
```

[5]:

	Date	TempHighF	TempAvgF	TempLowF	DewPointHighF	DewPointAvgF
0	2013-12-21	74.0	60.0	45.0	67.0	49.0
1	2013-12-22	56.0	48.0	39.0	43.0	36.0
2	2013-12-23	58.0	45.0	32.0	31.0	27.0
3	2013-12-24	61.0	46.0	31.0	36.0	28.0
4	2013-12-25	58.0	50.0	41.0	44.0	40.0

	DewPointLowF	HumidityHighPercent	HumidityAvgPercent	HumidityLowPercent
0	43.0	93	75	57
1	28.0	93	68	43
2	23.0	76	52	27
3	21.0	89	56	22
4	36.0	86	71	56

	SeaLevelPressureHighInches	SeaLevelPressureAvgInches
0	29.86	29.68
1	30.41	30.13
2	30.56	30.49
3	30.56	30.45
4	30.41	30.33

	SeaLevelPressureLowInches	VisibilityHighMiles	VisibilityAvgMiles
0	29.59	10.0	7.0
1	29.87	10.0	10.0
2	30.41	10.0	10.0
3	30.30	10.0	10.0
4	30.27	10.0	10.0

	VisibilityLowMiles	WindHighMPH	WindAvgMPH	WindGustMPH
0	2.0	20.0	4.0	31.0
1	5.0	16.0	6.0	25.0
2	10.0	8.0	3.0	12.0

```
3                 7.0         12.0         4.0         20.0
4                 7.0         10.0         2.0         16.0

   PrecipitationSumInches                Events
0                    0.46  Rain , Thunderstorm
1                    0.00
2                    0.00
3                    0.00
4                    0.00
```

[6]:
```
df.isna().sum() #isnull().sum() #discribe()
```

[6]:
```
Date                          0
TempHighF                     0
TempAvgF                      0
TempLowF                      0
DewPointHighF                 7
DewPointAvgF                  7
DewPointLowF                  7
HumidityHighPercent           2
HumidityAvgPercent            2
HumidityLowPercent            2
SeaLevelPressureHighInches    3
SeaLevelPressureAvgInches     3
SeaLevelPressureLowInches     3
VisibilityHighMiles          12
VisibilityAvgMiles           12
VisibilityLowMiles           12
WindHighMPH                   2
WindAvgMPH                    2
WindGustMPH                   4
PrecipitationSumInches        0
Events                        0
dtype: int64
```

[7]:
```
df = df.fillna(df.loc[:, df.columns != 'Events'].median()) #dropna(axis=1,␣
↪how="any") #fit #transform #fit_transform #model_encoder #inverse_transform
df.isna().sum() #one_hot_coder #get_dummies #category encoders: count(frequency)␣
↪encoding #minmax scaling #z-(avg) scaling #robust(median) scaling #sum
```

[7]:
```
Date                          0
TempHighF                     0
TempAvgF                      0
TempLowF                      0
DewPointHighF                 0
DewPointAvgF                  0
DewPointLowF                  0
```

```
HumidityHighPercent           0
HumidityAvgPercent            0
HumidityLowPercent            0
SeaLevelPressureHighInches    0
SeaLevelPressureAvgInches     0
SeaLevelPressureLowInches     0
VisibilityHighMiles           0
VisibilityAvgMiles            0
VisibilityLowMiles            0
WindHighMPH                   0
WindAvgMPH                    0
WindGustMPH                   0
PrecipitationSumInches        0
Events                        0
dtype: int64
```

[8]:
```
for i in df.columns:
  if i.endswith('Inches'):
    df[i] = [round(x * 25.4, 1) for x in df[i]]
    df.rename(columns={i: i.replace('Inches', 'Millimeters')}, inplace = True)
  elif i.endswith('F'):
    df[i] = [round((x - 32) / 1.8, 1) for x in df[i]]
    df.rename(columns={i: i[:-1] + 'C'}, inplace = True)
  elif i.endswith('Miles'):
    df[i] = [round(x * 1.6093445, 1) for x in df[i]]
    df.rename(columns={i: i.replace('Miles', 'Kilometers')}, inplace = True)
  elif i.endswith('MPH'):
    df[i] = [round(x * 1.6093445, 1) for x in df[i]]
    df.rename(columns={i: i.replace('MPH', 'KmPH')}, inplace = True)
df.head()
```

[8]:

	Date	TempHighC	TempAvgC	TempLowC	DewPointHighC	DewPointAvgC	DewPointLowC	HumidityHighPercent	HumidityAvgPercent	HumidityLowPercent	SeaLevelPressureHighMillimeters	SeaLevelPressureAvgMillimeters
0	2013-12-21	23.3	15.6	7.2	19.4	9.4	6.1	93	75	57	758.4	753.9
1	2013-12-22	13.3	8.9	3.9	6.1	2.2	-2.2	93	68	43		
2	2013-12-23	14.4	7.2	0.0	-0.6	-2.8	-5.0	76	52	27		
3	2013-12-24	16.1	7.8	-0.6	2.2	-2.2	-6.1	89	56	22		
4	2013-12-25	14.4	10.0	5.0	6.7	4.4	2.2	86	71	56		

```
1                          772.4                          765.3
2                          776.2                          774.4
3                          776.2                          773.4
4                          772.4                          770.4

   SeaLevelPressureLowMillimeters  VisibilityHighKilometers  \
0                           751.6                      16.1
1                           758.7                      16.1
2                           772.4                      16.1
3                           769.6                      16.1
4                           768.9                      16.1

   VisibilityAvgKilometers  VisibilityLowKilometers  WindHighKmPH  \
0                     11.3                      3.2          32.2
1                     16.1                      8.0          25.7
2                     16.1                     16.1          12.9
3                     16.1                     11.3          19.3
4                     16.1                     11.3          16.1

   WindAvgKmPH  WindGustKmPH  PrecipitationSumMillimeters                Events
0          6.4          49.9                         11.7  Rain , Thunderstorm
1          9.7          40.2                          0.0
2          4.8          19.3                          0.0
3          6.4          32.2                          0.0
4          3.2          25.7                          0.0
```

[9]:
```
df_deltas = df.drop(['Date', 'Events'], axis=1).diff().fillna(0).
 ↪add_prefix('Delta')
df_deltas.head()
```

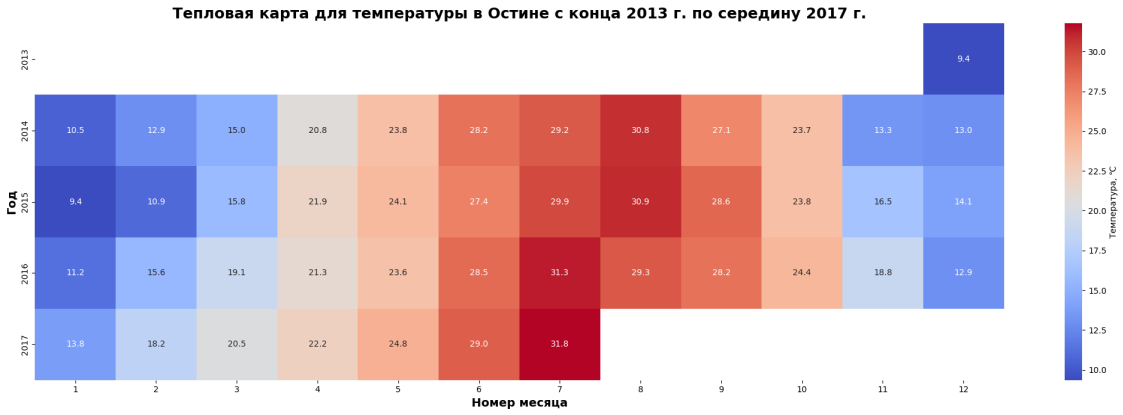
[9]:
```
   DeltaTempHighC  DeltaTempAvgC  DeltaTempLowC  DeltaDewPointHighC  \
0             0.0            0.0            0.0                 0.0
1           -10.0           -6.7           -3.3               -13.3
2             1.1           -1.7           -3.9                -6.7
3             1.7            0.6           -0.6                 2.8
4            -1.7            2.2            5.6                 4.5

   DeltaDewPointAvgC  DeltaDewPointLowC  DeltaHumidityHighPercent  \
0                0.0                0.0                         0
1               -7.2               -8.3                         0
2               -5.0               -2.8                       -17
3                0.6               -1.1                        13
4                6.6                8.3                        -3

   DeltaHumidityAvgPercent  DeltaHumidityLowPercent  \
0                        0                        0
1                       -7                      -14
```

```
2                      -16                      -16
3                        4                       -5
4                       15                       34

   DeltaSeaLevelPressureHighMillimeters  DeltaSeaLevelPressureAvgMillimeters  \
0                                   0.0                                 0.0
1                                  14.0                                11.4
2                                   3.8                                 9.1
3                                   0.0                                -1.0
4                                  -3.8                                -3.0

   DeltaSeaLevelPressureLowMillimeters  DeltaVisibilityHighKilometers  \
0                                  0.0                            0.0
1                                  7.1                            0.0
2                                 13.7                            0.0
3                                 -2.8                            0.0
4                                 -0.7                            0.0

   DeltaVisibilityAvgKilometers  DeltaVisibilityLowKilometers  \
0                           0.0                           0.0
1                           4.8                           4.8
2                           0.0                           8.1
3                           0.0                          -4.8
4                           0.0                           0.0

   DeltaWindHighKmPH  DeltaWindAvgKmPH  DeltaWindGustKmPH  \
0                0.0               0.0                0.0
1               -6.5               3.3               -9.7
2              -12.8              -4.9              -20.9
3                6.4               1.6               12.9
4               -3.2              -3.2               -6.5

   DeltaPrecipitationSumMillimeters
0                               0.0
1                             -11.7
2                               0.0
3                               0.0
4                               0.0
```

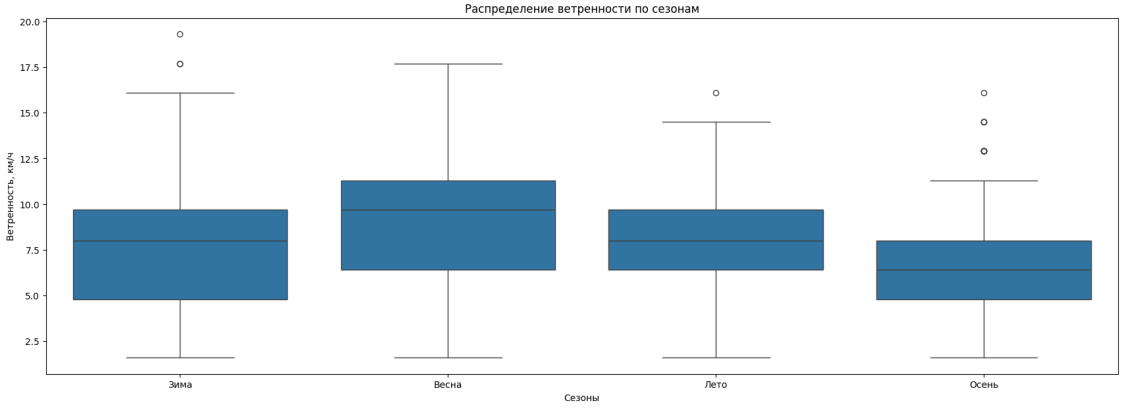
[10]:
```
df_sum = pd.concat([df.drop(['Date', 'Events'], axis=1), df_deltas], axis=1)
```

[11]:
```
seasons = df
heatmap_data = seasons.pivot_table(values='TempAvgC', index=seasons['Date'].dt.
 ↪year, columns=seasons['Date'].dt.month, aggfunc='mean')
plt.figure(figsize=(21, 7))
sns.heatmap(heatmap_data, cmap='coolwarm', annot=True, fmt=".1f",␣
 ↪cbar_kws={'label': 'Температура, ℃'})
```

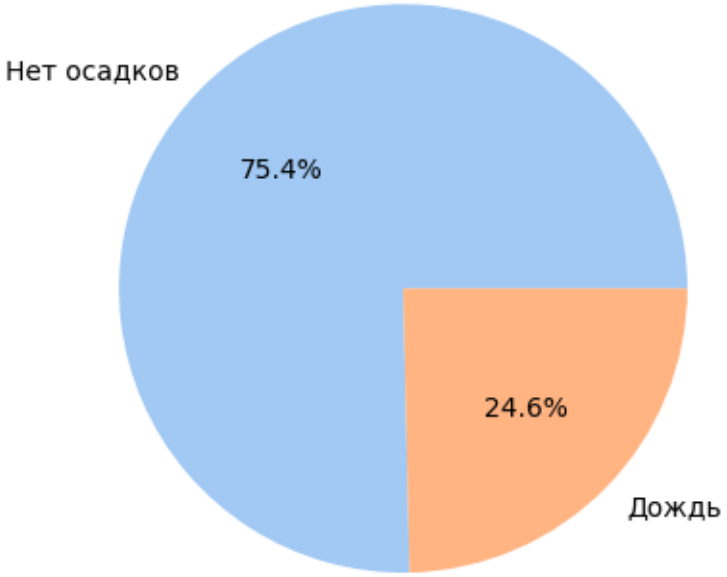
```
plt.title("Тепловая карта для температуры в Остине с конца 2013 г. по середину␣
 ↪2017 г.", fontsize=18, fontweight='bold')
plt.xlabel('Номер месяца', fontsize=14, fontweight='bold')
plt.ylabel('Год', fontsize=14, fontweight='bold')
plt.tight_layout()
plt.show()
```



[12]:
```
seasons_prec = df[['Date', 'WindAvgKmPH']]
plt.figure(figsize=(21,7))
s = seasons_prec['Date'].dt.month%12 // 3
s.replace({0: 'Зима', 1: 'Весна', 2 : 'Лето', 3 : 'Осень'}, inplace=True)
sns.boxplot(x=s, y='WindAvgKmPH', data=seasons_prec)
plt.title('Распределение ветренности по сезонам')
plt.xlabel('Сезоны')
plt.ylabel('Ветренность, км/ч')
plt.show()
```



[13]:
```
df_prec = df
func = lambda row: 'Нет осадков' if row['PrecipitationSumMillimeters'] == 0 else␣
↪('Снег' if row['TempAvgC'] < 0 else 'Дождь')
df_prec['PrecCategory'] = df_prec.apply(func, axis=1)
prec_category_counts = df_prec['PrecCategory'].value_counts()
colors = sns.color_palette('pastel')[0:5]
plt.pie(prec_category_counts, labels = prec_category_counts.index, colors =␣
↪colors, autopct='%.1f%%')
plt.show()
```



[14]:
```
import itertools
to_comb = [
    ['DeltaPrecipitationSumMillimeters'],
    ['DeltaTempHighC', 'DeltaTempAvgC', 'DeltaTempLowC']
]
pairs = list(itertools.product(*to_comb))
print(pairs)
```

```
[('DeltaPrecipitationSumMillimeters', 'DeltaTempHighC'),
('DeltaPrecipitationSumMillimeters', 'DeltaTempAvgC'),
('DeltaPrecipitationSumMillimeters', 'DeltaTempLowC')]
```

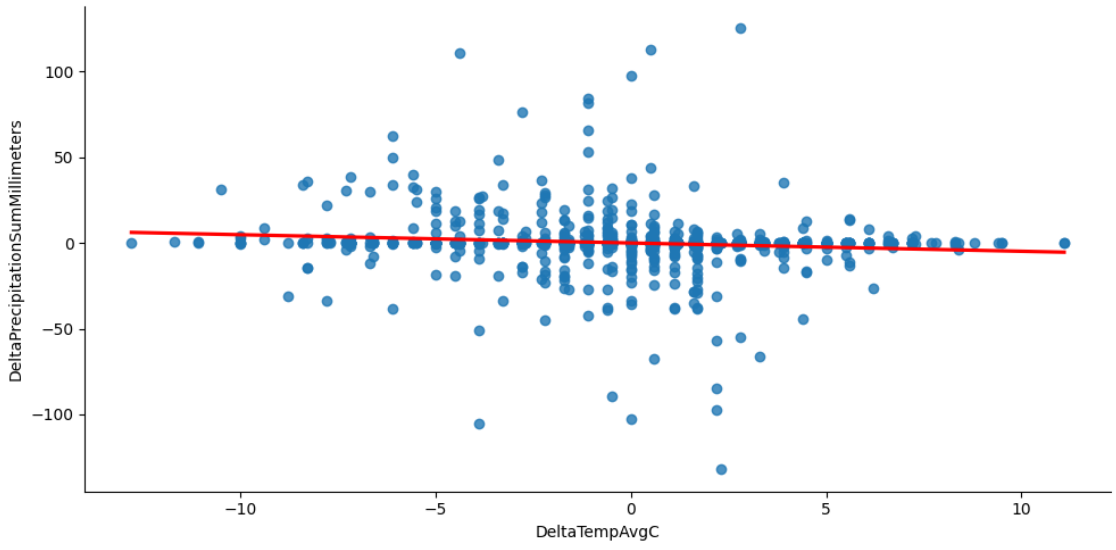
[15]:
```
def get_top_abs_correlations(df, n=5, ascending=False, method='pearson'):
    au_corr = df.corr(method=method).abs().unstack()
    labels_to_drop = []
    for i in au_corr.keys():
        mv = 0
        if i[0].startswith("Delta") and i[1].startswith("Delta"):
            mv = 5
        if i[0][mv:].startswith(i[1][mv:(mv+4)]):
            labels_to_drop.append((i[0], i[1]))
    au_corr = au_corr.drop(labels=labels_to_drop).
↪sort_values(ascending=ascending)
    return au_corr[0:n]
```

[16]:
```
print("Топ зависимостей изменения влажности от изменения температуры")
print(get_top_abs_correlations(df_deltas, 1000).loc[[
    ('DeltaPrecipitationSumMillimeters', 'DeltaTempHighC'),
    ('DeltaPrecipitationSumMillimeters', 'DeltaTempAvgC'),
    ('DeltaPrecipitationSumMillimeters', 'DeltaTempLowC')
]].sort_values(ascending=False))
```

```
Топ зависимостей изменения влажности от изменения температуры
DeltaPrecipitationSumMillimeters  DeltaTempAvgC     0.108690
                                  DeltaTempHighC    0.105533
                                  DeltaTempLowC     0.071729
dtype: float64
```

[17]:
```
sns.lmplot(
    data=df_deltas, x="DeltaTempAvgC", y="DeltaPrecipitationSumMillimeters",
    ci=None, height=5, aspect=2, line_kws={'color': 'red'}
)
```

[17]:
```
<seaborn.axisgrid.FacetGrid at 0x22ead4f6810>
```

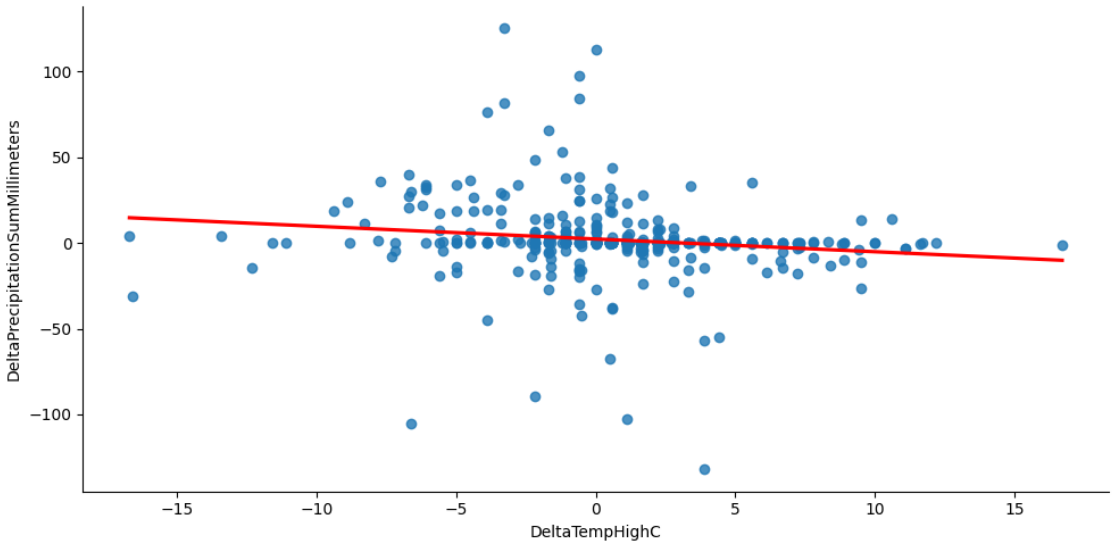


[18]:
```
df_good = df_sum[
    (df_sum['DewPointAvgC'] <= df_sum['TempHighC'].shift(-1).
 ↪fillna(df_sum['TempHighC']))
    & (df_sum['DewPointAvgC'] >= df_sum['TempLowC'])
]
print("Скорректированный топ зависимостей изменения влажности от изменения␣
 ↪температуры")
print(get_top_abs_correlations(df_good, 1000, method='spearman').loc[[
    ('DeltaPrecipitationSumMillimeters', 'DeltaTempHighC'),
    ('DeltaPrecipitationSumMillimeters', 'DeltaTempAvgC'),
    ('DeltaPrecipitationSumMillimeters', 'DeltaTempLowC')
]].sort_values(ascending=False))
```

```
Скорректированный топ зависимостей изменения влажности от изменения температуры
DeltaPrecipitationSumMillimeters  DeltaTempHighC    0.308873
                                  DeltaTempAvgC     0.234257
                                  DeltaTempLowC     0.090832
dtype: float64
```

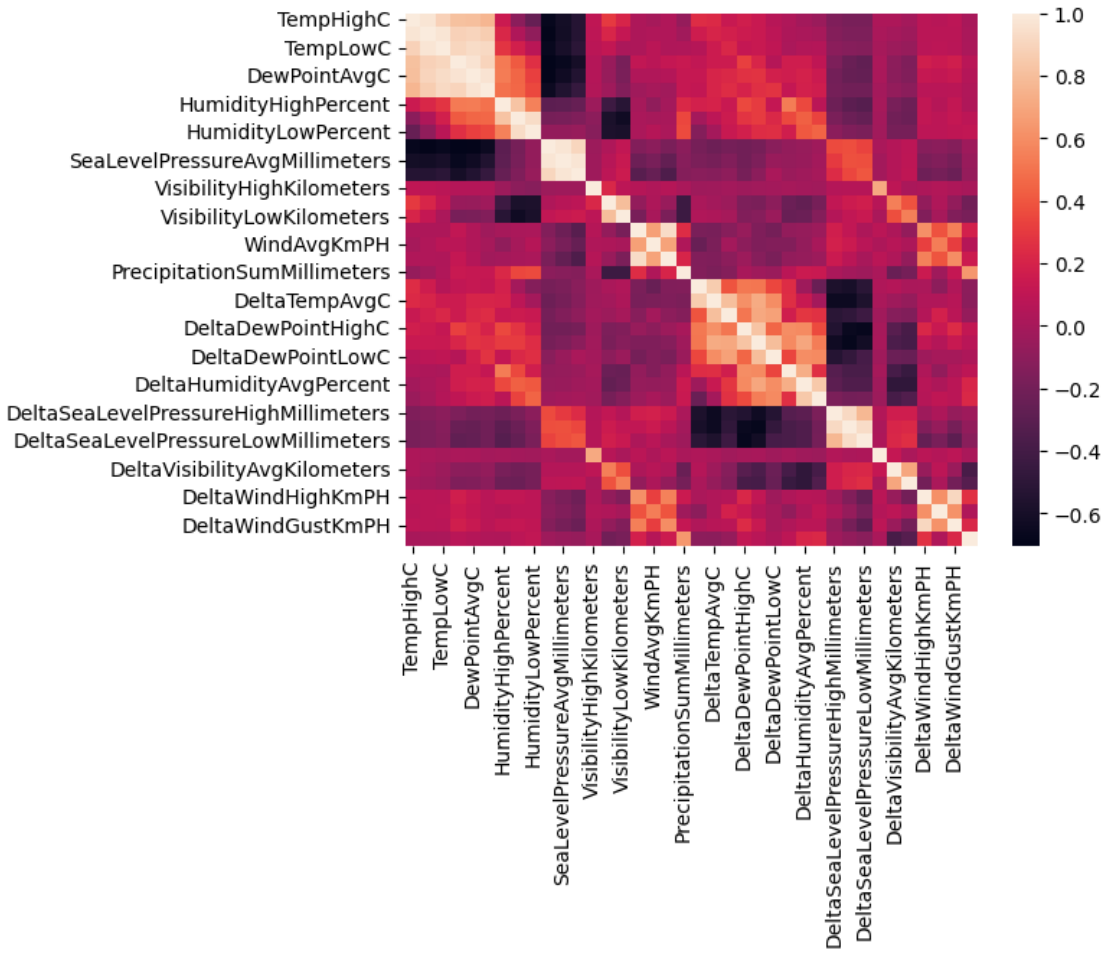
[19]:
```
sns.lmplot(
    data=df_good, x="DeltaTempHighC", y="DeltaPrecipitationSumMillimeters",
    ci=None, height=5, aspect=2, line_kws={'color': 'red'}
)
```

[19]:
```
<seaborn.axisgrid.FacetGrid at 0x22ead482a80>
```



[20]:
```
df_sum = pd.concat([df.drop(['Date', 'Events', 'PrecCategory'], axis=1),␣
↪df_deltas], axis=1)
corr = df_sum.corr()
sns.heatmap(corr)
```

[20]: <Axes: >



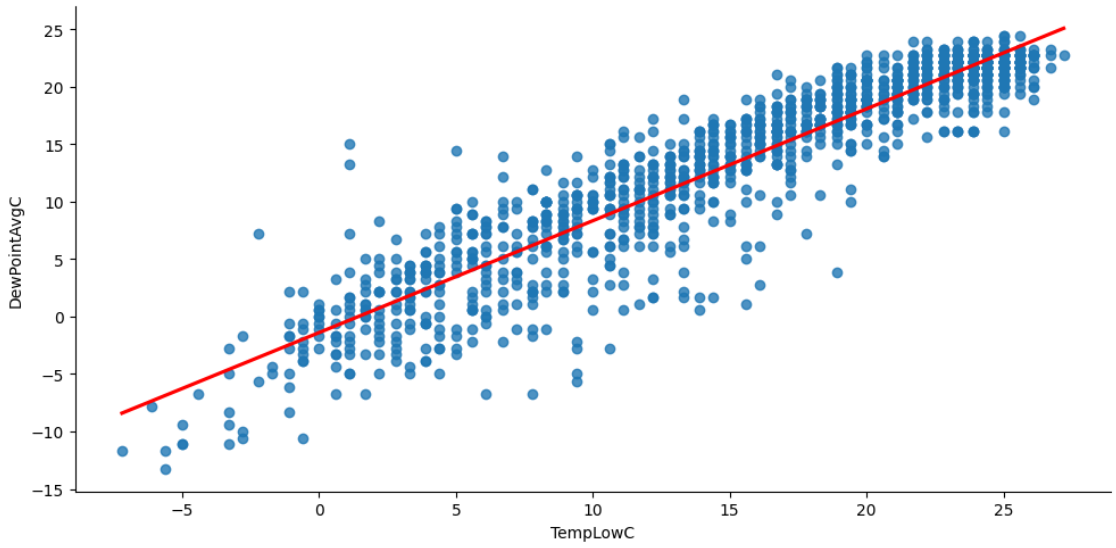
[21]:
```
print("Самые взаимозависимые пары")
print(get_top_abs_correlations(df_sum, 20))
```

```
Самые взаимозависимые пары
DewPointAvgC       TempLowC             0.931366
TempLowC           DewPointAvgC         0.931366
                   DewPointLowC         0.914180
DewPointLowC       TempLowC             0.914180
DewPointHighC      TempLowC             0.900668
TempLowC           DewPointHighC        0.900668
TempAvgC           DewPointAvgC         0.893751
```

```
DewPointAvgC       TempAvgC             0.893751
TempAvgC           DewPointHighC        0.880192
DewPointHighC      TempAvgC             0.880192
TempAvgC           DewPointLowC         0.862480
DewPointLowC       TempAvgC             0.862480
TempHighC          DewPointHighC        0.810382
DewPointHighC      TempHighC            0.810382
TempHighC          DewPointAvgC         0.806514
DewPointAvgC       TempHighC            0.806514
TempHighC          DewPointLowC         0.763156
DewPointLowC       TempHighC            0.763156
DeltaDewPointAvgC  DeltaTempAvgC        0.715610
DeltaTempAvgC      DeltaDewPointAvgC    0.715610
dtype: float64
```

[22]:
```
sns.lmplot(
    data=df, x="TempLowC", y="DewPointAvgC", ci=None, height=5, aspect=2,␣
 ↪line_kws={'color': 'red'}
)
```

[22]: <seaborn.axisgrid.FacetGrid at 0x22ead886210>



[ ]: