In [ ]:
```
# Packages
import pandas as pd
import numpy as np
import seaborn as sns
import scipy.stats as stats
from scipy.stats import binom

from matplotlib import pyplot as plt
```

Brief description of the dataset and a summary of its attributes

We import the dataset and explore its characteristics. The dataset contains one numerical variable and 8 objects, that are either strings or dates. The measurements are monthly measurements and the date column, holds the respective dates.

In [ ]:
```
df = pd.read_csv('data.csv')
```

In [ ]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50799 entries, 0 to 50798
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Unnamed: 0      50799 non-null  int64  
 1   date            50799 non-null  object 
 2   datatype        50799 non-null  object 
 3   station         50799 non-null  object 
 4   attributes      50778 non-null  object 
 5   value           50799 non-null  float64
 6   countryid       50799 non-null  object 
 7   id              50799 non-null  object 
 8   datacategoryid  50799 non-null  object 
 9   name            50799 non-null  object 
dtypes: float64(1), int64(1), object(8)
memory usage: 3.9+ MB
None
```

The dataset includes 10 different features on temperature and precipitation.

In [ ]:
```
print(df[['datacategoryid', 'id', 'name']].drop_duplicates(ignore_index=True))
```

	datacategoryid	id	name
0	TEMP	CDSD	Cooling Degree Days Season to Date
1	TEMP	EMNT	Extreme minimum temperature for the period.
2	TEMP	EMXT	Extreme maximum temperature for the period.
3	TEMP	HDSD	Heating Degree Days Season to Date
4	TEMP	TAVG	Average Temperature.
5	TEMP	TMAX	Maximum temperature
6	TEMP	TMIN	Minimum temperature
7	PRCP	DSND	Number days with snow depth > 1 inch(25.4mm) f...
8	PRCP	EMSD	Extreme maximum snow depth for the period.
9	PRCP	EMXP	Extreme maximum precipitation for the period.
10	PRCP	PRCP	Precipitation

In [ ]:
```
print(min(df['date']))
print(max(df['date']))
```

```
2014-01-01T00:00:00
2023-06-01T00:00:00
```

The measurements cover the period of Jan. 2014 - June 2023.

Initial plan for data exploration

We cast each feature to a numerical variable and we check the basic statistics of those variables.

In [ ]:
```
df.pivot(index = ['date', 'station'], columns = 'id', values = 'value')
```

Out[ ]:

date	station	CDSD	DSND	EMNT	EMSD	EMXP	EMXT	HDSD	PRCP	TAVG
2014-01-01T00:00:00	GHCND:FG000081405	NaN	NaN	NaN	NaN	85.1	NaN	NaN	594.7	NaN
	GHCND:FGM00081401	NaN	NaN	19.7	NaN	35.1	NaN	NaN	233.7	NaN
	GHCND:FGM00081415	NaN	NaN	20.1	NaN	25.9	NaN	NaN	161.8	NaN
	GHCND:FP000091925	NaN	NaN	NaN	NaN	30.0	32.7	NaN	81.4	NaN
	GHCND:FP000091948	NaN	NaN	NaN	NaN	29.0	NaN	NaN	126.4	NaN
...	...	...	...	...	...	...	...	...	...	...
2023-06-01T00:00:00	GHCND:GRM00016622	NaN	NaN	NaN	NaN	37.3	NaN	NaN	94.3	NaN
	GHCND:GRM00016719	185.5	NaN	13.0	NaN	0.5	33.8	938.2	0.5	23.59
	GHCND:GRM00016726	NaN	NaN	NaN	NaN	1.8	33.4	NaN	3.9	NaN
	GHCND:MFM00067005	NaN	NaN	NaN	NaN	0.3	NaN	NaN	2.4	NaN
	GHCND:SBM00071805	NaN	NaN	NaN	NaN	17.8	NaN	NaN	76.8	NaN

9777 rows × 11 columns

In [ ]:
```
df_summary = df.groupby(['countryid', 'name', 'id'])['value'].agg([np.mean, np.median,
                                                                   lambda x : np.quant
    .rename(columns = {"<lambda_0>":"25 percentile", "<lambda_1>":"75 percentile", "<l
```

```
print(df_summary)
```

countryid	name	id	mean
FIPS:FR	Average Temperature.	TAVG	13.641054
	Cooling Degree Days Season to Date	CDSD	147.630388
	Extreme maximum precipitation for the period.	EMXP	20.382597
	Extreme maximum snow depth for the period.	EMSD	214.636364
	Extreme maximum temperature for the period.	EMXT	24.945502
	Extreme minimum temperature for the period.	EMNT	3.745144
	Heating Degree Days Season to Date	HDSD	962.318617
	Maximum temperature	TMAX	18.299411
	Minimum temperature	TMIN	9.418633
	Number days with snow depth > 1 inch(25.4mm) fo...	DSND	20.454545
	Precipitation	PRCP	68.164582
FIPS:GR	Average Temperature.	TAVG	20.550979
	Cooling Degree Days Season to Date	CDSD	303.608108
	Extreme maximum precipitation for the period.	EMXP	19.572738
	Extreme maximum temperature for the period.	EMXT	28.422034
	Extreme minimum temperature for the period.	EMNT	11.586339
	Heating Degree Days Season to Date	HDSD	162.616667
	Maximum temperature	TMAX	23.785297
	Minimum temperature	TMIN	15.950929
	Precipitation	PRCP	49.995360

countryid	name	id	median
FIPS:FR	Average Temperature.	TAVG	13.360
	Cooling Degree Days Season to Date	CDSD	49.900
	Extreme maximum precipitation for the period.	EMXP	15.700
	Extreme maximum snow depth for the period.	EMSD	211.000
	Extreme maximum temperature for the period.	EMXT	24.800
	Extreme minimum temperature for the period.	EMNT	3.300
	Heating Degree Days Season to Date	HDSD	830.000
	Maximum temperature	TMAX	18.095
	Minimum temperature	TMIN	9.050
	Number days with snow depth > 1 inch(25.4mm) fo...	DSND	19.000
	Precipitation	PRCP	53.700
FIPS:GR	Average Temperature.	TAVG	21.300
	Cooling Degree Days Season to Date	CDSD	50.200
	Extreme maximum precipitation for the period.	EMXP	13.000
	Extreme maximum temperature for the period.	EMXT	28.400
	Extreme minimum temperature for the period.	EMNT	11.900
	Heating Degree Days Season to Date	HDSD	0.000
	Maximum temperature	TMAX	23.545
	Minimum temperature	TMIN	16.420
	Precipitation	PRCP	27.700

countryid	name	id	std
FIPS:FR	Average Temperature.	TAVG	5.939523
	Cooling Degree Days Season to Date	CDSD	195.952333
	Extreme maximum precipitation for the period.	EMXP	18.448003
	Extreme maximum snow depth for the period.	EMSD	84.184646
	Extreme maximum temperature for the period.	EMXT	7.526989
	Extreme minimum temperature for the period.	EMNT	6.653705
	Heating Degree Days Season to Date	HDSD	860.134014
	Maximum temperature	TMAX	6.814612
	Minimum temperature	TMIN	5.761973
	Number days with snow depth > 1 inch(25.4mm) fo...	DSND	6.547727
	Precipitation	PRCP	65.168376
FIPS:GR	Average Temperature.	TAVG	5.734915

```
                Cooling Degree Days Season to Date                CDSD  372.805368
                Extreme maximum precipitation for the period.     EMXP   21.208681
                Extreme maximum temperature for the period.       EMXT    6.968606
                Extreme minimum temperature for the period.       EMNT    6.106147
                Heating Degree Days Season to Date                HDSD  326.478472
                Maximum temperature                               TMAX    6.518079
                Minimum temperature                               TMIN    5.430642
                Precipitation                                     PRCP   63.664680

                                                                    min  \
countryid name                                                  id
FIPS:FR   Average Temperature.                                  TAVG  -1.75
          Cooling Degree Days Season to Date                    CDSD   0.00
          Extreme maximum precipitation for the period.         EMXP   0.00
          Extreme maximum snow depth for the period.            EMSD  79.00
          Extreme maximum temperature for the period.           EMXT   6.10
          Extreme minimum temperature for the period.           EMNT -16.80
          Heating Degree Days Season to Date                    HDSD   0.00
          Maximum temperature                                   TMAX  -1.93
          Minimum temperature                                   TMIN  -6.07
          Number days with snow depth > 1 inch(25.4mm) fo...    DSND  11.00
          Precipitation                                         PRCP   0.00
FIPS:GR   Average Temperature.                                  TAVG   8.44
          Cooling Degree Days Season to Date                    CDSD   0.00
          Extreme maximum precipitation for the period.         EMXP   0.00
          Extreme maximum temperature for the period.           EMXT  14.80
          Extreme minimum temperature for the period.           EMNT  -2.00
          Heating Degree Days Season to Date                    HDSD   0.00
          Maximum temperature                                   TMAX  11.79
          Minimum temperature                                   TMIN   4.14
          Precipitation                                         PRCP   0.00

                                                                    max  \
countryid name                                                  id
FIPS:FR   Average Temperature.                                  TAVG   29.10
          Cooling Degree Days Season to Date                    CDSD  941.30
          Extreme maximum precipitation for the period.         EMXP  299.50
          Extreme maximum snow depth for the period.            EMSD  361.00
          Extreme maximum temperature for the period.           EMXT   43.50
          Extreme minimum temperature for the period.           EMNT   23.00
          Heating Degree Days Season to Date                    HDSD 3217.90
          Maximum temperature                                   TMAX   35.74
          Minimum temperature                                   TMIN   24.88
          Number days with snow depth > 1 inch(25.4mm) fo...    DSND   29.00
          Precipitation                                         PRCP  943.90
FIPS:GR   Average Temperature.                                  TAVG   30.36
          Cooling Degree Days Season to Date                    CDSD  957.20
          Extreme maximum precipitation for the period.         EMXP  142.70
          Extreme maximum temperature for the period.           EMXT   43.20
          Extreme minimum temperature for the period.           EMNT   23.20
          Heating Degree Days Season to Date                    HDSD  991.60
          Maximum temperature                                   TMAX   35.29
          Minimum temperature                                   TMIN   25.42
          Precipitation                                         PRCP  568.40

                                                          25 percentile  \
countryid name                                                  id
FIPS:FR   Average Temperature.                                  TAVG        8.9175
          Cooling Degree Days Season to Date                    CDSD        0.0000
          Extreme maximum precipitation for the period.         EMXP        9.9000
```

	Extreme maximum snow depth for the period.	EMSD	150.0000	
	Extreme maximum temperature for the period.	EMXT	18.8000	
	Extreme minimum temperature for the period.	EMNT	-1.3000	
	Heating Degree Days Season to Date	HDSD	78.0000	
	Maximum temperature	TMAX	12.8400	
	Minimum temperature	TMIN	4.8450	
	Number days with snow depth > 1 inch(25.4mm) fo...	DSND	15.5000	
	Precipitation	PRCP	30.1000	
FIPS:GR	Average Temperature.	TAVG	15.4350	
	Cooling Degree Days Season to Date	CDSD	0.0000	
	Extreme maximum precipitation for the period.	EMXP	4.6000	
	Extreme maximum temperature for the period.	EMXT	21.8750	
	Extreme minimum temperature for the period.	EMNT	6.8500	
	Heating Degree Days Season to Date	HDSD	0.0000	
	Maximum temperature	TMAX	17.8750	
	Minimum temperature	TMIN	11.3550	
	Precipitation	PRCP	7.1250	

			75 percentile	\
countryid	name	id		
FIPS:FR	Average Temperature.	TAVG	18.4300	
	Cooling Degree Days Season to Date	CDSD	242.0000	
	Extreme maximum precipitation for the period.	EMXP	24.4000	
	Extreme maximum snow depth for the period.	EMSD	254.0000	
	Extreme maximum temperature for the period.	EMXT	31.3000	
	Extreme minimum temperature for the period.	EMNT	8.3000	
	Heating Degree Days Season to Date	HDSD	1719.7000	
	Maximum temperature	TMAX	23.8725	
	Minimum temperature	TMIN	13.6550	
	Number days with snow depth > 1 inch(25.4mm) fo...	DSND	26.5000	
	Precipitation	PRCP	86.7000	
FIPS:GR	Average Temperature.	TAVG	25.6100	
	Cooling Degree Days Season to Date	CDSD	697.5000	
	Extreme maximum precipitation for the period.	EMXP	27.7000	
	Extreme maximum temperature for the period.	EMXT	34.6000	
	Extreme minimum temperature for the period.	EMNT	16.8500	
	Heating Degree Days Season to Date	HDSD	62.0250	
	Maximum temperature	TMAX	29.6125	
	Minimum temperature	TMIN	20.8250	
	Precipitation	PRCP	66.5750	

			Count
countryid	name	id	
FIPS:FR	Average Temperature.	TAVG	4052
	Cooling Degree Days Season to Date	CDSD	3508
	Extreme maximum precipitation for the period.	EMXP	8826
	Extreme maximum snow depth for the period.	EMSD	11
	Extreme maximum temperature for the period.	EMXT	5380
	Extreme minimum temperature for the period.	EMNT	4191
	Heating Degree Days Season to Date	HDSD	3615
	Maximum temperature	TMAX	5380
	Minimum temperature	TMIN	4191
	Number days with snow depth > 1 inch(25.4mm) fo...	DSND	11
	Precipitation	PRCP	8826
FIPS:GR	Average Temperature.	TAVG	143
	Cooling Degree Days Season to Date	CDSD	37
	Extreme maximum precipitation for the period.	EMXP	862
	Extreme maximum temperature for the period.	EMXT	236
	Extreme minimum temperature for the period.	EMNT	183
	Heating Degree Days Season to Date	HDSD	66

```
Maximum temperature                    TMAX    236
Minimum temperature                    TMIN    183
Precipitation                          PRCP    862
```

Data cleaning and feature engineering

we explore the dataset in its initial form, looking for outliers. We also simplify by removing the stations feature. Finally we explore the distribution of variables and look for the need of transformations or scaling. We do not need to deal with categorical variables. We only have numerical variables.

We get the interquartile range, and look for outliers.

In [ ]:
```
df_summary['iqr'] = df_summary['75 percentile'] - df_summary['25 percentile']
```

In [ ]:
```
df_summary['upper_limit'] = df_summary['75 percentile'] + 1.5 * df_summary['iqr']
df_summary['lower_limit'] = df_summary['25 percentile'] - 1.5 * df_summary['iqr']
```

In [ ]:
```
display(df_summary[['upper_limit', 'lower_limit']])
```

countryid	name	id	upper_limit	lower_limit
FIPS:FR	Average Temperature.	TAVG	32.69875	-5.35125
	Cooling Degree Days Season to Date	CDSD	605.00000	-363.00000
	Extreme maximum precipitation for the period.	EMXP	46.15000	-11.85000
	Extreme maximum snow depth for the period.	EMSD	410.00000	-6.00000
	Extreme maximum temperature for the period.	EMXT	50.05000	0.05000
	Extreme minimum temperature for the period.	EMNT	22.70000	-15.70000
	Heating Degree Days Season to Date	HDSD	4182.25000	-2384.55000
	Maximum temperature	TMAX	40.42125	-3.70875
	Minimum temperature	TMIN	26.87000	-8.37000
	Number days with snow depth > 1 inch(25.4mm) for the period.	DSND	43.00000	-1.00000
	Precipitation	PRCP	171.60000	-54.80000
FIPS:GR	Average Temperature.	TAVG	40.87250	0.17250
	Cooling Degree Days Season to Date	CDSD	1743.75000	-1046.25000
	Extreme maximum precipitation for the period.	EMXP	62.35000	-30.05000
	Extreme maximum temperature for the period.	EMXT	53.68750	2.78750
	Extreme minimum temperature for the period.	EMNT	31.85000	-8.15000
	Heating Degree Days Season to Date	HDSD	155.06250	-93.03750
	Maximum temperature	TMAX	47.21875	0.26875
	Minimum temperature	TMIN	35.03000	-2.85000
	Precipitation	PRCP	155.75000	-82.05000

In [ ]:
```
# Just some cleaning of the main table
df.reset_index()
df.drop(['Unnamed: 0'], axis = 1, inplace = True)
```

In [ ]:
```
df = df.merge(df_summary[['upper_limit', 'lower_limit']], on = ['countryid', 'id'], ho
```

In [ ]:
```
# Check for outliers
outliers = df[(df['value']>df['upper_limit'])|(df['value']<df['lower_limit'])]
```

In [ ]:
```
outliers.shape
```

Out[ ]:
```
(1296, 11)
```

We find that approx. 1000 rows in 50000 rows of dataset are extreme values, which is around 2%. Since there are several extreme values, we are going to keep them.

In [ ]:
```
df.index
```

Out[ ]:
```
Int64Index([    0,     1,     2,     3,     4,     5,     6,     7,     8,
                9,
            ...
            50789, 50790, 50791, 50792, 50793, 50794, 50795, 50796, 50797,
            50798],
           dtype='int64', length=50799)
```

In [ ]:
```
display(df)
```

	date	datatype	station	attributes	value	countryid	id	datacategory
0	2014-01-01T00:00:00	CDSD	GHCND:FR000007130	E	0.0	FIPS:FR	CDSD	TEM
1	2014-01-01T00:00:00	CDSD	GHCND:FR000007190	E	0.0	FIPS:FR	CDSD	TEM
2	2014-01-01T00:00:00	CDSD	GHCND:FR000007255	E	0.0	FIPS:FR	CDSD	TEM
3	2014-01-01T00:00:00	CDSD	GHCND:FR000007510	E	0.0	FIPS:FR	CDSD	TEM
4	2014-01-01T00:00:00	CDSD	GHCND:FR000007630	E	0.0	FIPS:FR	CDSD	TEM
...	...	...	...	...	...	...	...	
50794	2023-06-01T00:00:00	PRCP	GHCND:GR000016754	,,,S	43.2	FIPS:GR	PRCP	PRC
50795	2023-06-01T00:00:00	PRCP	GHCND:GR000167230	1,,,S	0.0	FIPS:GR	PRCP	PRC
50796	2023-06-01T00:00:00	PRCP	GHCND:GRM00016622	,,,S	94.3	FIPS:GR	PRCP	PRC
50797	2023-06-01T00:00:00	PRCP	GHCND:GRM00016719	,,,S	0.5	FIPS:GR	PRCP	PRC
50798	2023-06-01T00:00:00	PRCP	GHCND:GRM00016726	,,,S	3.9	FIPS:GR	PRCP	PRC

50799 rows × 11 columns

For each day, we can have measurements taken at several stations. We can summarize those measurements and reduce the size of the dataset.

In [ ]:
```
df_mean = df.groupby(['countryid', 'datacategoryid', 'datatype', 'name', 'date'])[['va
```

In [ ]:
```
display(df_mean)
```

					value	
					mean	median
countryid	**datacategoryid**	**datatype**	**name**	**date**		
FIPS:FR	**PRCP**	**DSND**	**Number days with snow depth > 1 inch(25.4mm) for the period.**	**2014-02-01T00:00:00**	28.000000	28.00
				2014-03-01T00:00:00	19.000000	19.00
				2015-01-01T00:00:00	12.000000	12.00
				2015-02-01T00:00:00	14.000000	14.00
				2015-03-01T00:00:00	19.000000	19.00
...	**...**	**...**	**...**	**...**	...	...
FIPS:GR	**TEMP**	**TMIN**	**Minimum temperature**	**2023-02-01T00:00:00**	5.080000	5.08
				2023-03-01T00:00:00	8.010000	8.01
				2023-04-01T00:00:00	10.546667	9.98
				2023-05-01T00:00:00	13.850000	13.85
				2023-06-01T00:00:00	19.026667	18.40

1804 rows × 2 columns

In [ ]:
```
df_mean.columns = ['_'.join(col) for col in df_mean.columns.values]
```

In [ ]:
```
df_mean.columns.values
```

Out[ ]:
```
array(['value_mean', 'value_median'], dtype=object)
```

In [ ]:
```
df_mean[['value_mean']].isna().sum() # No missing values here
```

Out[ ]:
```
value_mean    0
dtype: int64
```

In [ ]:
```
df_mean_wide = df_mean.reset_index(['datatype', 'datacategoryid', 'name']).pivot(colum
```

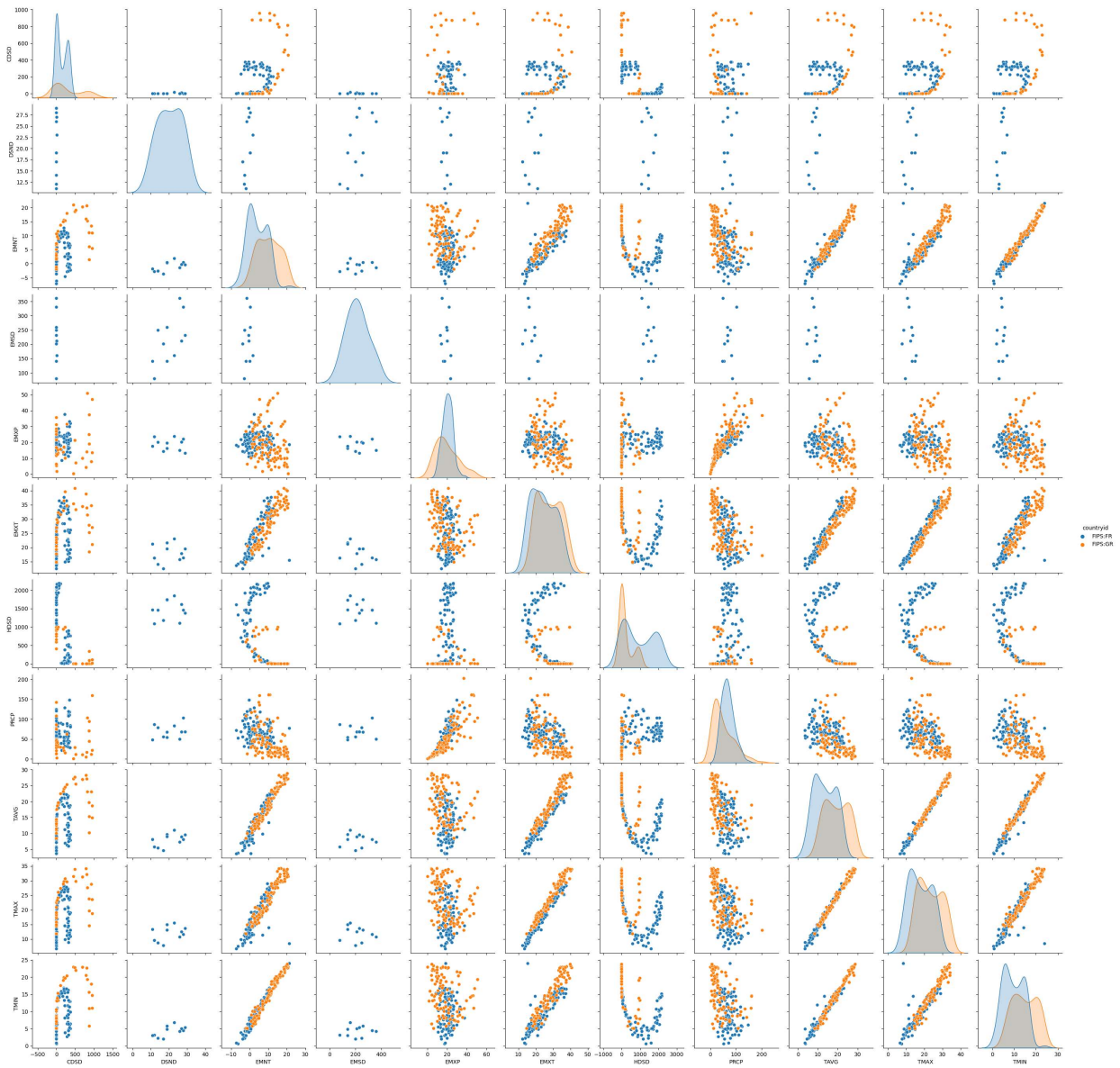
In [ ]:
```
df_mean_wide.isna().sum()
```

Out[ ]:
```
datatype
CDSD    105
DSND    217
EMNT     19
EMSD    217
EMXP      0
EMXT      6
HDSD     88
PRCP      0
TAVG     27
TMAX      6
TMIN     19
dtype: int64
```

After transforming to a wide form, we see that missing values appear in the e.g. temperature measurements, but not in the e.g. precipitation.

In [ ]:
```
sns.pairplot(df_mean_wide.reset_index(), hue = 'countryid')
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x2a553291e10>
```



We check for skewed measurements in the combinations of datatype and countryid columns.

In [ ]:
```
df_mean_skew = df_mean_wide.groupby('countryid').skew().reset_index().melt(id_vars = '
```

In [ ]:
```
df_mean_skew = df_mean_skew.set_index(['datatype', 'countryid'])
skew_cols = df_mean_skew[abs(df_mean_skew['value'])>0.75]
```

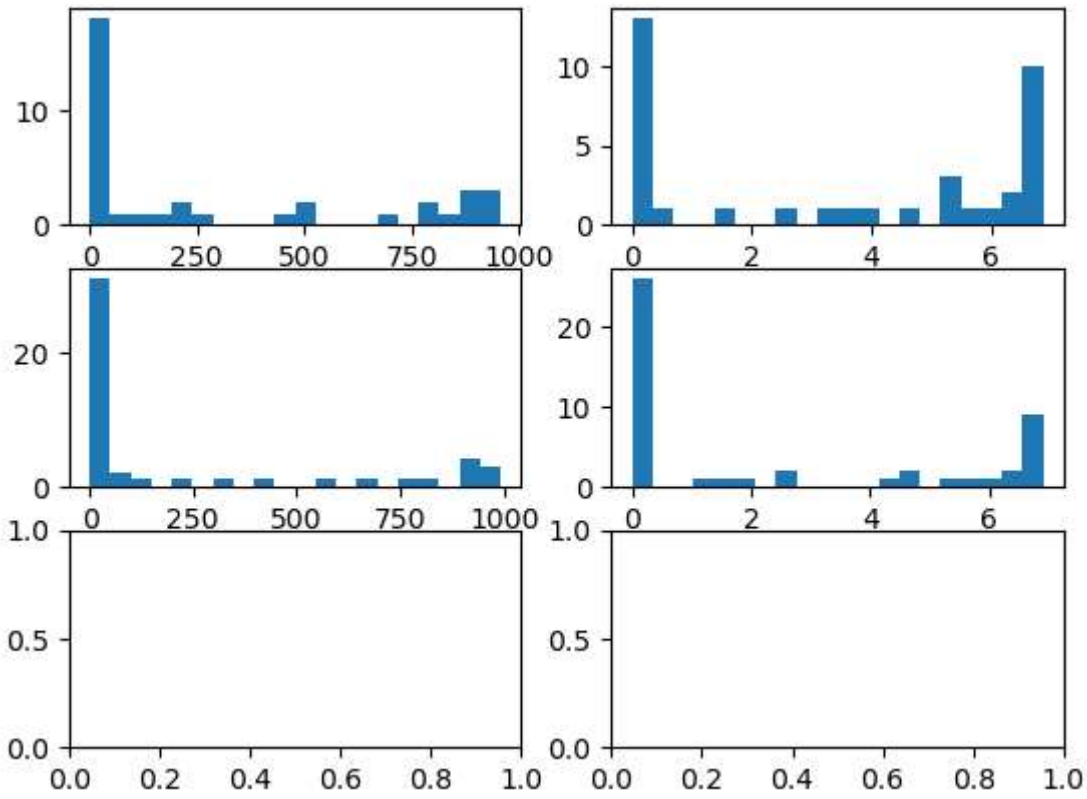
In [ ]:
```
skew_cols
```

Out[ ]:

		value
datatype	**countryid**	
CDSD	**FIPS:GR**	0.756271
HDSD	**FIPS:GR**	1.297929
PRCP	**FIPS:GR**	1.198044

For the variables that we transformed, compare before and after the transformation.

In [ ]:
```
fig, ax = plt.subplots(3,2)
[ax[i,0].hist(df_mean_wide.loc[skew_cols.index.values[i][1], skew_cols.index.values[i]
[ax[i,1].hist(df_mean_wide.loc[skew_cols.index.values[i][1], skew_cols.index.values[i]
plt.show()
```



In [ ]:
```
# Apply the log transformation to the dataframe if we want
logTransform = False
if (logTransform == True) :
     for variable, country in skew_cols.index.values:
          df_mean_wide.loc[country, variable] = df_mean_wide.loc[country, variable].ap
```

In [ ]:
```
# Check that there is no skeweness in the transformed variables
df_mean_skew = df_mean_wide.groupby('countryid').skew().reset_index().melt(id_vars = '
df_mean_skew = df_mean_skew.set_index(['datatype', 'countryid'])
skew_cols = df_mean_skew[abs(df_mean_skew['value'])>0.75]
[print(variable + country) for variable, country in skew_cols.index]
```

Out[ ]:
```
CDSDFIPS:GR
HDSDFIPS:GR
PRCPFIPS:GR
[None, None, None]
```

Key findings and insights

Observations:

- Min, max and average temperatures, they follow linear relations one with each other.
- Temperatures follow a quadratic relation with heating/cooling degrees days. Cooling degrees days tends to increase with temperature, but there seems to be a threshold of days where it starts decreasing. It probably means that looking at large number of HDD, because it is cumulative it occurs at season with low temperatures. Similarly for heating degrees days, which is the demand of energy to heat a building, we see that HDD inrease as temperatures decrease, it reaches a minimum and then it is increasing also as temperatures increase.
- looking at the relation between temperature and precipitation, it is not clear whether there is correlation between the two variables. Maybe further transformations can help to further understand the relation.

Hypothesis testing

Hypothesis:

- There is no correlation between temperature and precipitation.
- Greece and France have similar behaviour on temperature. When the average temperature in Greece increases, also the same happens in France.
- Greece and France have similar behaviour on precipitation. When the precipitation in Greece increases, also the same happens in France.

1st hypothesis

- Null hypothesis: The correlation between temperature and precipitation is zero.

- Alternative hypothesis: The correlation between temperature and precipitation is different than zero.

To answer the question we need the true rate of the correlation coefficient. We take let's say 100 samples of measurements and calculate the probability that there is no correlation. We define as zero correlation when the correlation coefficient is in the interval [-0.1,0.1]

In [ ]:
```
# Look for missing values in temperature and precipitation
df_mean_wide[['PRCP', 'TAVG']].isna().sum()
```

Out[ ]:
```
datatype
PRCP     0
TAVG    27
dtype: int64
```

In [ ]:
```
# Remove the missing values in temperature and precipitation
df_T_PRCP = df_mean_wide[(pd.isna(df_mean_wide['PRCP']) == False)&(pd.isna(df_mean_wic
print(df_T_PRCP[['PRCP', 'TAVG']].isna().sum()) # check
```

```
datatype
PRCP    0
TAVG    0
dtype: int64
```

In [ ]:
```
df_T_PRCP = df_T_PRCP[['PRCP', 'TAVG']]

# Take 1000 samples of 50 measurements and calculate the pearson correlation coefficie
dt_corrcoef = pd.DataFrame()

for i in range(1000):
    corrcoef_sample = stats.pearsonr(x = np.array(df_T_PRCP.sample(50, replace = True)
    corrcoef_dt = pd.DataFrame([{'Sample':i, 'CorrCoef': corrcoef_sample}])
    dt_corrcoef = pd.concat([dt_corrcoef, corrcoef_dt], axis = 0)
```

In [ ]:
```
#Calculate the probability of zero correlation in the samples
proba = dt_corrcoef[(dt_corrcoef['CorrCoef']<=0.1)&(dt_corrcoef['CorrCoef']>=-0.1)].sh
print(proba)
```

```
0.517
```

In [ ]:
```
# We know the probability of getting zero correlation. And so we calculate the probabi
prob1 = 1-binom.cdf(60, 100, (1-proba))
print("Out of 100 samples, the probability to have correlation in more than 60 of thos
```

```
Out of 100 samples, the probability to have correlation in more than 60 of those is
0.7%
```

In [ ]:
```
print("Out of 100 samples we have correlation in at least "+ str(binom.ppf(0.95,100,1-
```

```
Out of 100 samples we have correlation in at least 58.0 samples with confidence level
95%.
```

Therefore there is some form of correlation present and it is rather improbable that we have zero correlation.

2nd hypothesis

Greece and France have different temperatures.

- Null hypothesis: The temperatures in Greece and in France are the same.
- Alternative hyppothesis: The temperatures in Greece and in France are different.

We choose 5% significance level.

In [ ]:
```
df_T = df_mean_wide[(pd.isna(df_mean_wide['TAVG']) == False)]
df_T = df_T.reset_index()
df_T = df_T[['countryid','TAVG']]
print(df_T)
```

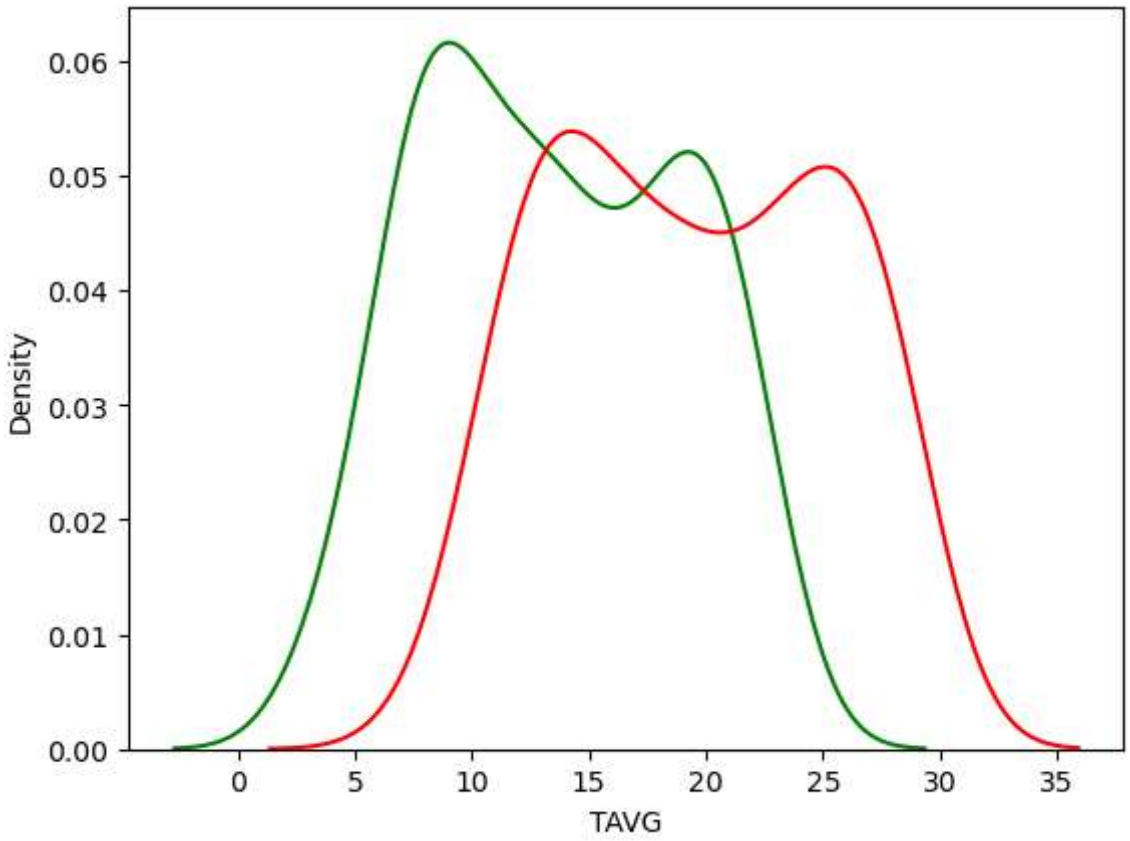
```
datatype countryid       TAVG
0          FIPS:FR   7.576977
1          FIPS:FR   8.031957
2          FIPS:FR   9.610000
3          FIPS:FR  13.044894
4          FIPS:FR  14.306889
..             ...        ...
196        FIPS:GR   9.980000
197        FIPS:GR  12.990000
198        FIPS:GR  14.340000
199        FIPS:GR  18.610000
200        FIPS:GR  23.115000

[201 rows x 2 columns]
```

In [ ]:
```
france=df_T.loc[df_T['countryid']=="FIPS:FR", 'TAVG']
greece=df_T.loc[df_T['countryid']=="FIPS:GR", 'TAVG']
```

In [ ]:
```
sns.distplot(france,color='green',hist=False)
sns.distplot(greece,color='red',hist=False)
```

```
C:\Users\tzava\AppData\Local\Temp\ipykernel_10188\2272142649.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(france,color='green',hist=False)
C:\Users\tzava\AppData\Local\Temp\ipykernel_10188\2272142649.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(greece,color='red',hist=False)
```

Out[ ]:
```
<Axes: xlabel='TAVG', ylabel='Density'>
```



From the density plots we observe that the distribution of the temperatures in Greece and in France are similar but there is a shift between them, showing that Greece has higher temperatures than France. Also we observe that in both countries the most common temperatures are towards the edges of the distribution rather than in the middle.

In [ ]:
```
print(france.mean())
print(greece.mean())
```

```
13.501033918342605
19.411666666666665
```

We see that the mean temperature in Greece is higher than the mean temperature in France.

In [ ]:
```
alpha=0.05
t_value1, p_value1 = stats.ttest_ind(france, greece)
print("t_value1 = ",t_value1, ", p_value1 = ", p_value1)
```

```
t_value1 =  -7.464618912987033 , p_value1 =  2.553322148087781e-12
```

In [ ]:
```
if p_value1 <alpha:
    print("Conclusion: since p_value {} is less than alpha {} ". format (p_value1,alph
    print("Reject the null hypothesis that there is no difference between the temperat

else:
    print("Conclusion: since p_value {} is greater than alpha {} ". format (p_value1,a
    print("Fail to reject the null hypothesis that there is no difference between the
```

```
Conclusion: since p_value 2.553322148087781e-12 is less than alpha 0.05
Reject the null hypothesis that there is no difference between the temperatures in Gr
eece and France.
```

3rd hypothesis

Greece and France have different precipitation levels.

- Null hypothesis: The precipitation in Greece and in France is the same.
- Alternative hyppothesis: The precipitation in Greece and in France is different.

We choose 5% significance level.

In [ ]:
```
df_PRCP = df_mean_wide[(pd.isna(df_mean_wide['PRCP']) == False)]
df_PRCP = df_PRCP.reset_index()
df_PRCP = df_PRCP[['countryid','PRCP']]
print(df_PRCP)
```

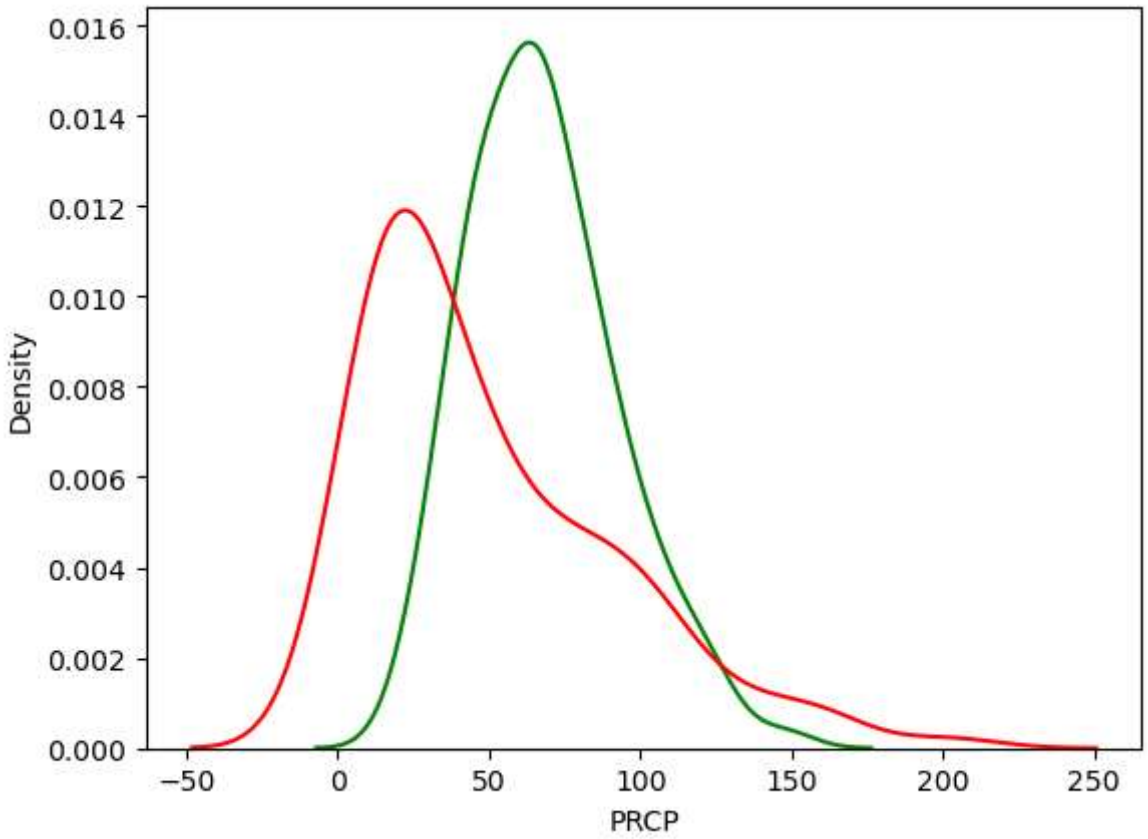
```
datatype countryid        PRCP
0          FIPS:FR  124.025301
1          FIPS:FR  102.704938
2          FIPS:FR   53.793750
3          FIPS:FR   49.677778
4          FIPS:FR   78.538272
..             ...         ...
223        FIPS:GR   12.775000
224        FIPS:GR   40.787500
225        FIPS:GR   45.712500
226        FIPS:GR   41.385714
227        FIPS:GR   27.900000

[228 rows x 2 columns]
```

In [ ]:
```
france_PRCP=df_PRCP.loc[df_PRCP['countryid']=="FIPS:FR", 'PRCP']
greece_PRCP=df_PRCP.loc[df_PRCP['countryid']=="FIPS:GR", 'PRCP']
```

In [ ]:
```
sns.distplot(france_PRCP,color='green',hist=False)
sns.distplot(greece_PRCP,color='red',hist=False)
```

```
C:\Users\tzava\AppData\Local\Temp\ipykernel_10188\2817746939.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(france_PRCP,color='green',hist=False)
C:\Users\tzava\AppData\Local\Temp\ipykernel_10188\2817746939.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(greece_PRCP,color='red',hist=False)
```

Out[ ]: <Axes: xlabel='PRCP', ylabel='Density'>



In [ ]:
```
print(france_PRCP.mean())
print(greece_PRCP.mean())
```

```
67.92803348086044
50.262224310776936
```

In [ ]:
```
alpha=0.05
t_value1, p_value1 = stats.ttest_ind(france_PRCP, greece_PRCP)
print("t_value1 = ",t_value1, ", p_value1 = ", p_value1)
```

```
t_value1 =  3.886752810839097 , p_value1 =  0.0001335345719358334
```

In [ ]:
```
if p_value1 <alpha:
    print("Conclusion: since p_value {} is less than alpha {} ". format (p_value1,alph
    print("Reject the null hypothesis that there is no difference between the precipit

else:
    print("Conclusion: since p_value {} is greater than alpha {} ". format (p_value1,a
    print("Fail to reject the null hypothesis that there is no difference between the
```

```
Conclusion: since p_value 0.0001335345719358334 is less than alpha 0.05
Reject the null hypothesis that there is no difference between the precipitation in G
reece and France.
```