Imports

In [1]:

```
!pip install loguru
!pip install impyute
!pip install missingpy
!pip install tqdm
```

```
Requirement already satisfied: loguru in /home/alexander/HSE_Stuff/envs/data_analysis/lib
/python3.7/site-packages (0.5.3)
Requirement already satisfied: impyute in /home/alexander/HSE_Stuff/envs/data_analysis/li
b/python3.7/site-packages (0.0.8)
Requirement already satisfied: scipy in /home/alexander/HSE_Stuff/envs/data_analysis/lib/
python3.7/site-packages (from impyute) (1.7.1)
Requirement already satisfied: numpy in /home/alexander/HSE_Stuff/envs/data_analysis/lib/
python3.7/site-packages (from impyute) (1.21.2)
Requirement already satisfied: scikit-learn in /home/alexander/HSE_Stuff/envs/data_analys
is/lib/python3.7/site-packages (from impyute) (0.22.1)
Requirement already satisfied: joblib>=0.11 in /home/alexander/HSE_Stuff/envs/data_analys
is/lib/python3.7/site-packages (from scikit-learn->impyute) (1.0.1)
Requirement already satisfied: missingpy in /home/alexander/HSE_Stuff/envs/data_analysis/
lib/python3.7/site-packages (0.2.0)
Requirement already satisfied: tqdm in /home/alexander/HSE_Stuff/envs/data_analysis/lib/p
ython3.7/site-packages (4.62.3)
```

In [2]:

```
import pandas as pd
import missingno as msno
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from loguru import logger
from tqdm import tqdm
%matplotlib inline
```

1. Main characteristics of features

In [5]:

```
col_names = [f'feature{i}' for i in range(1,6)]
df = pd.read_excel('data/data.xlsx',
                   header=None,
                   names=col_names)
```

In [6]:

```
df.head()
```

Out[6]:

	feature1	feature2	feature3	feature4	feature5
0	1.117915	-2.065559	NaN	-1.362986	-0.709196
1	-0.503055	-2.892927	0.713895	-1.631237	-1.790220
2	NaN	-0.077614	0.772394	-1.500325	-0.164499
3	1.163245	1.046673	1.674523	-1.867193	-0.534950
4	0.185329	-2.824514	2.036840	-2.040093	NaN

In [7]:

```
df.shape
```

df.shape

Out[7]:

```
(50, 5)
```

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   feature1  45 non-null     float64
 1   feature2  46 non-null     float64
 2   feature3  45 non-null     float64
 3   feature4  47 non-null     float64
 4   feature5  45 non-null     float64
dtypes: float64(5)
memory usage: 2.1 KB
```

In [9]:

```
df.describe()
```

Out[9]:

	feature1	feature2	feature3	feature4	feature5
count	45.000000	46.000000	45.000000	47.000000	45.000000
mean	0.731691	-0.832829	0.867845	-0.768877	0.388719
std	1.151209	1.616996	1.053143	0.963373	1.347656
min	-1.815804	-4.122851	-1.379847	-2.729031	-2.626825
25%	-0.057839	-2.136559	0.163378	-1.555415	-0.304180
50%	0.991282	-0.587220	0.946280	-0.704662	0.222050
75%	1.621050	0.357034	1.435358	-0.045401	1.280085
max	3.184440	1.999813	4.266199	0.838120	3.467592

2. Data visualization

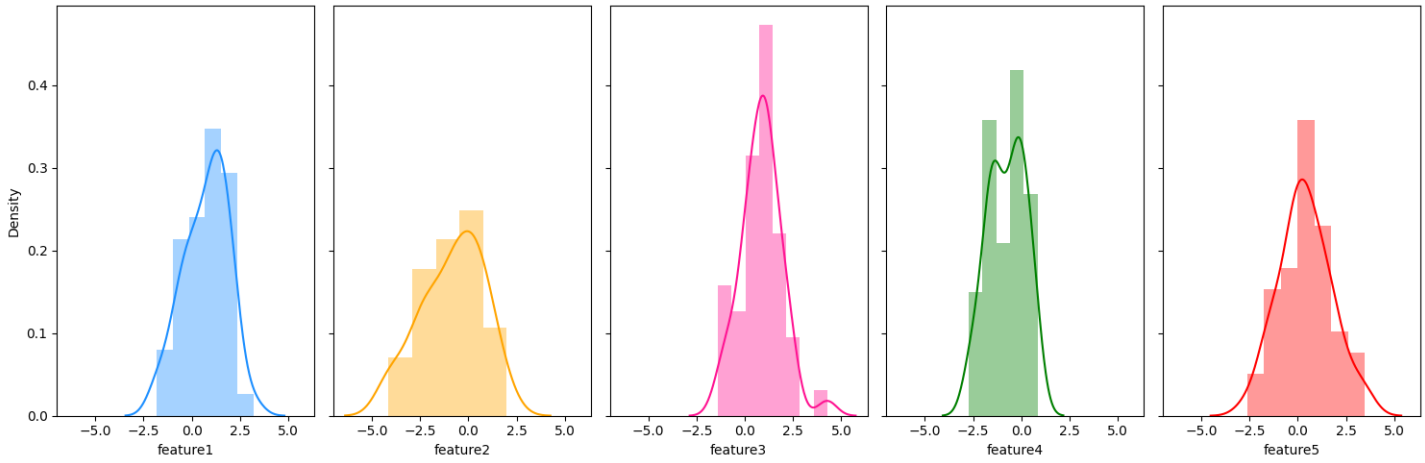
Probability Histogram

In [10]:

```
fig, axes = plt.subplots(1, 5, figsize=(15, 5), dpi=100, sharex=True, sharey=True)
colors = ['dodgerblue', 'orange', 'deeppink', 'green', 'red']

for feature, ax, color in zip(df, axes, colors):
    sns.distplot(df[feature], ax=ax, color=color)

plt.tight_layout();
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
```

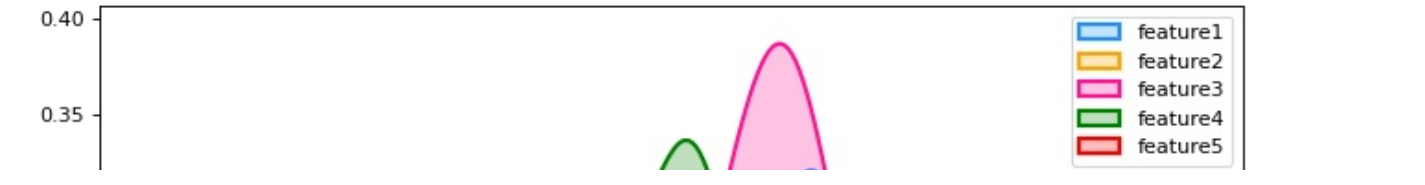
```
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

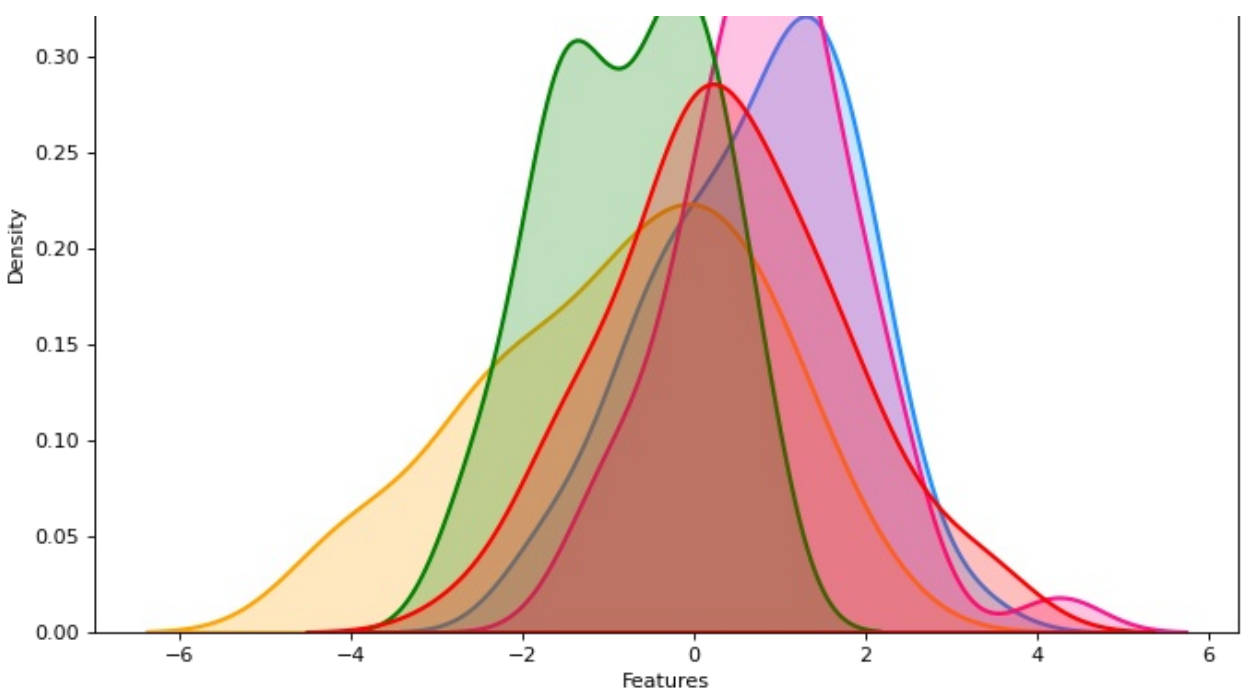


In [11]:

```python
kwargs = dict(hist_kws={'alpha':.6}, kde_kws={'shade': True, 'linewidth':2})
colors = ['dodgerblue', 'orange', 'deeppink', 'green', 'red']
plt.figure(figsize=(10,7), dpi=80)
for feature, color in zip(df, colors):
    sns.distplot(df[feature], hist=False, color=color, label=feature, **kwargs)
plt.xlabel('Features')
plt.legend();
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. 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).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. 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).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. 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).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. 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).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. 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).
  warnings.warn(msg, FutureWarning)
```





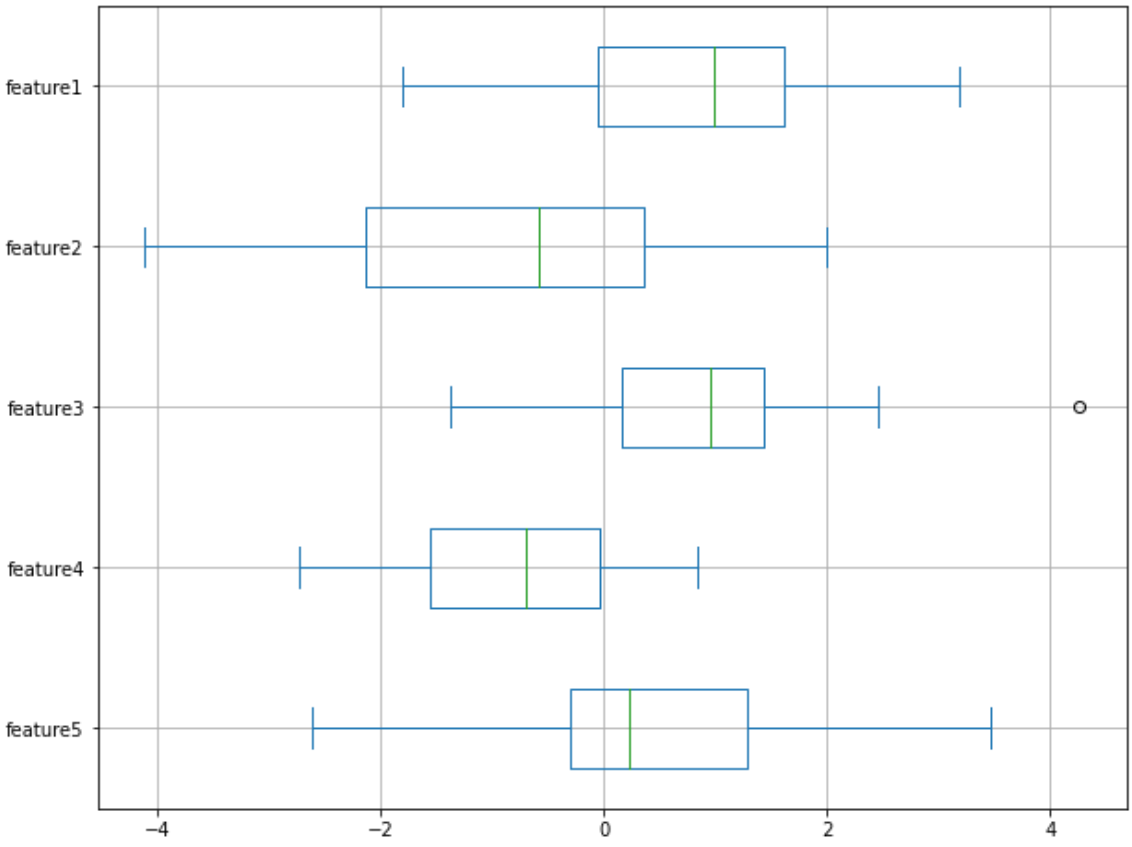
Box plot

In [12]:

```
fig, ax = plt.subplots(figsize=(10, 8))
df[['feature5', 'feature4', 'feature3', 'feature2', 'feature1']].plot.box(vert=False,
                                                                         grid=True,
                                                                         ax=ax)
```

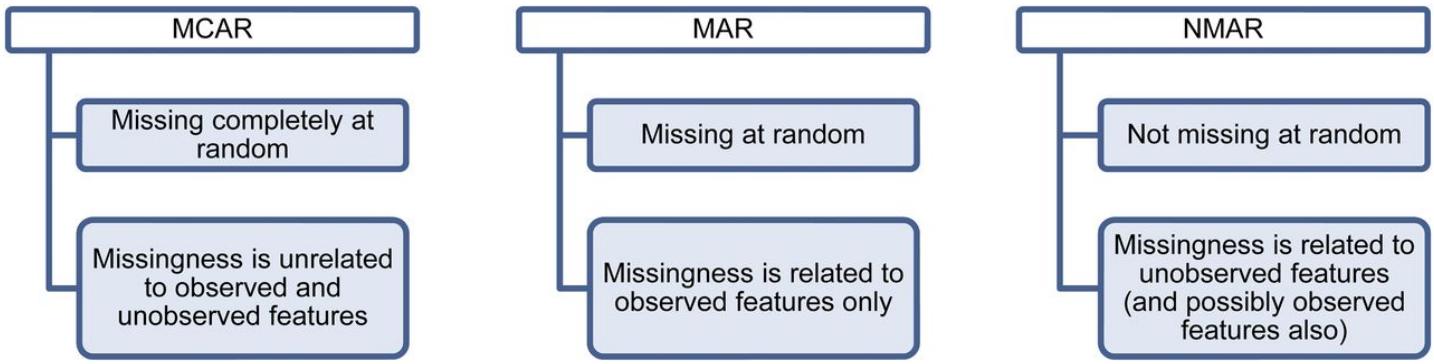
Out[12]:

<AxesSubplot:>



3. Missing Values Analysis

Usually missing data problems areclassified into three categories. We'll try to categorize the missing values in our data.



In [13]:

```
def missing_values_table(df):
    # Total missing values
    mis_val = df.isnull().sum()

    # Percentage of missing values
    mis_val_percent = 100 * df.isnull().sum() / len(df)

    # Make a table with the results
    mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)

    # Rename the columns
    mis_val_table_ren_columns = mis_val_table.rename(
    columns = {0 : 'Missing Values Amount ', 1 : 'Percent of Total Values'})

    # Sort the table by percentage of missing descending
    mis_val_table_ren_columns = mis_val_table_ren_columns[
        mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
    'Percent of Total Values', ascending=False).round(1)

    return mis_val_table_ren_columns
```

In [ ]:

```
missing_values_table(df)
```

Out[ ]:

	Missing Values Amount	Percent of Total Values
feature1	5	10.0
feature3	5	10.0
feature5	5	10.0
feature2	4	8.0
feature4	3	6.0

Tools

! For better understanding of missing values patterns, we will use the missingno library which allows us to get a quick visual summary of the completeness of our data.

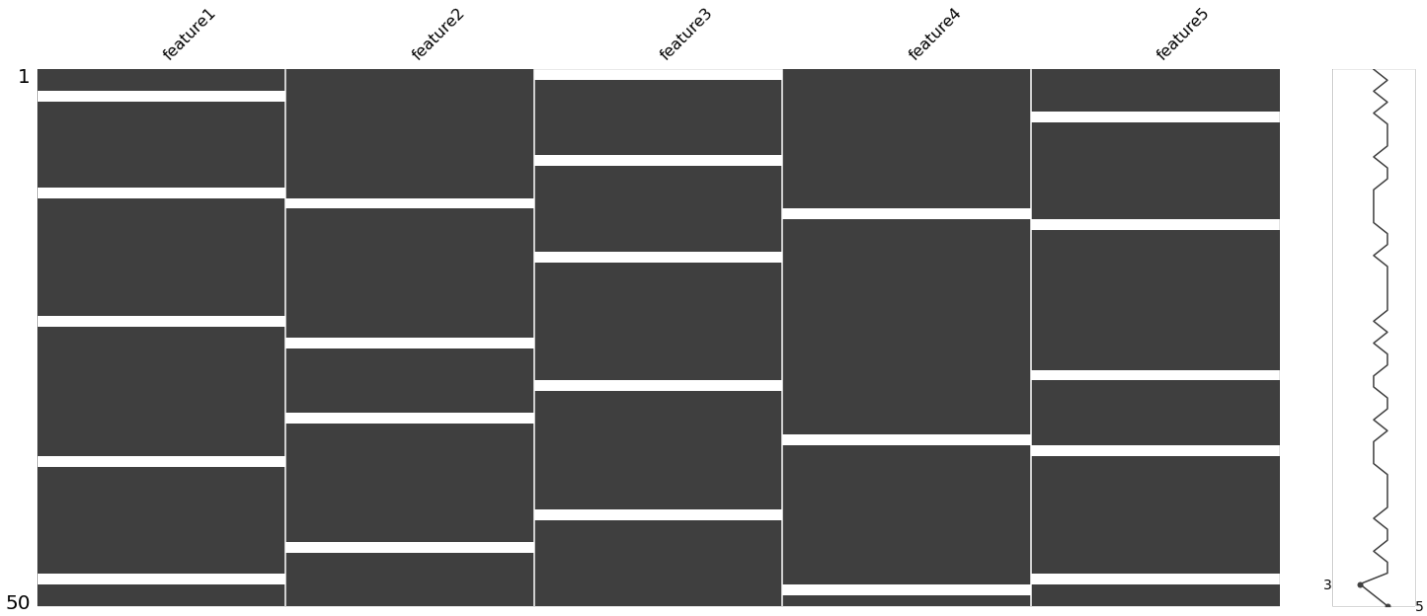
Visualization location of the missing data

In [14]:

```
msno.matrix(df)
```

Out[14]:

```
<AxesSubplot:>
```



Correlation of features missingness.

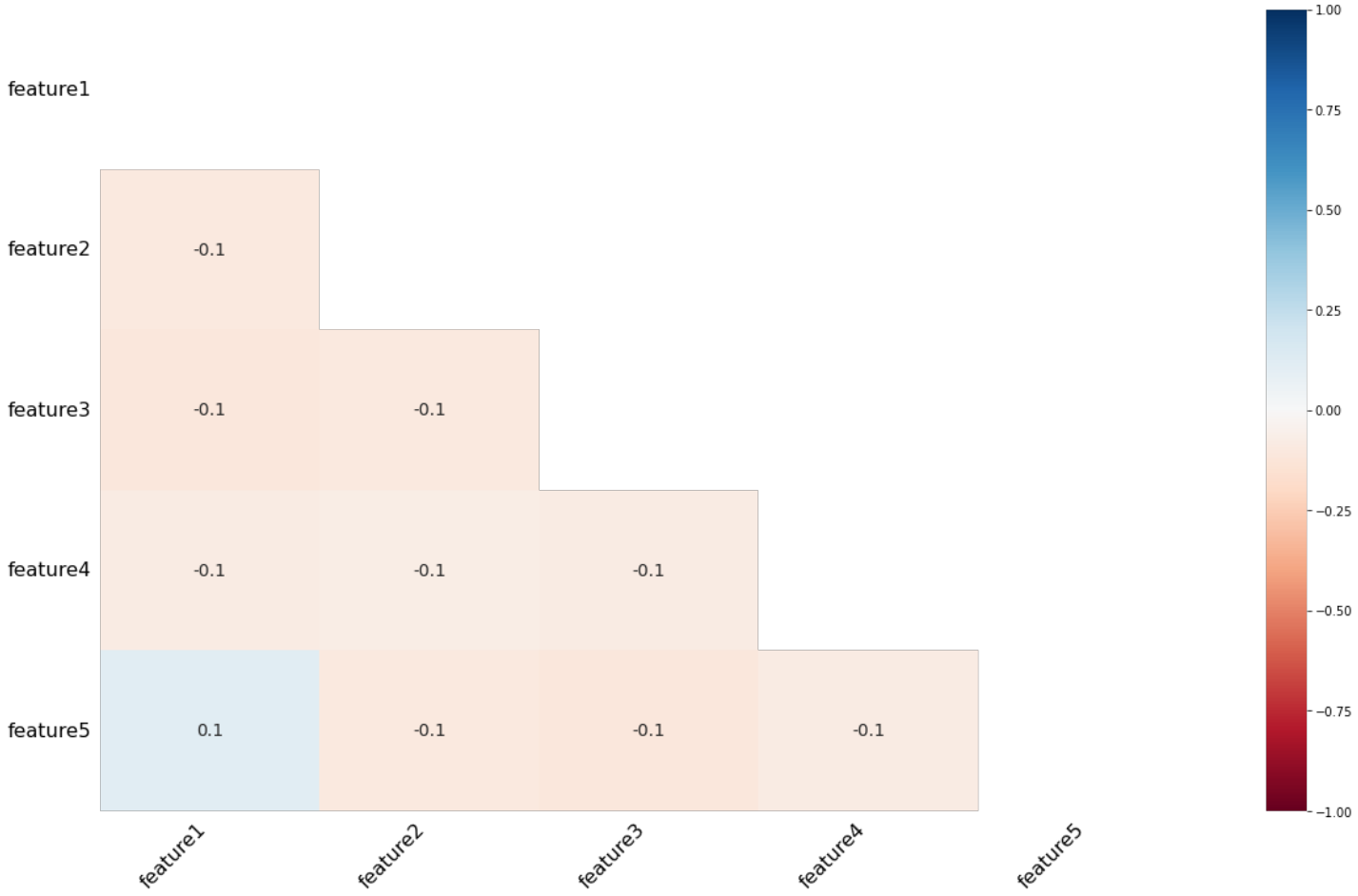
It represents how strongly the presence or absence of one variable affects the presence of another

In [15]:

```
msno.heatmap(df)
```

Out[15]:

<AxesSubplot:>



The heatmap shows that there are no strong correlations between missing values of different features. The low correlations indicate that the data are Missing at Random (MAR).

Features Dendrogram (Hierarchy Clusterization)

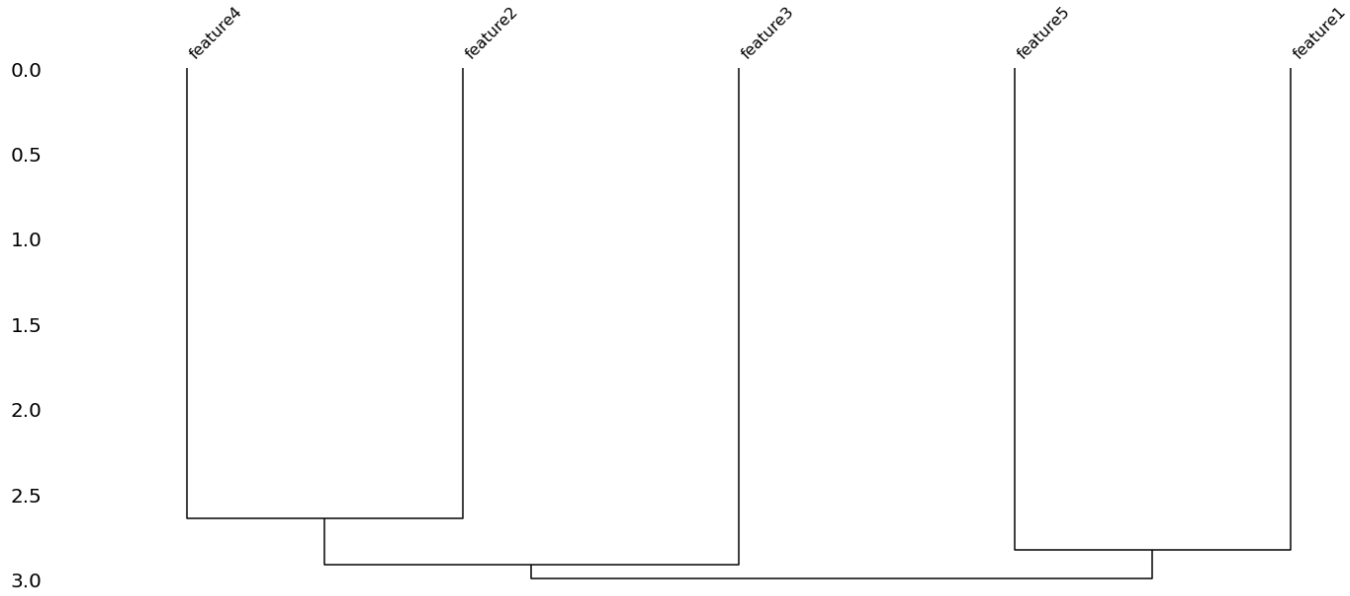
The dendrogram allows you to revealing trends deeper than the pairwise ones visible in the correlation heatmap:

In [16]:

```
msno.dendrogram(df)
```

Out[16]:

<AxesSubplot:>



As the missingno documentaion states:

> **Cluster leaves which linked together at a distance of zero fully predict one another's presence—one variable might always be empty when another is filled, or they >might always both be filled or both empty, and so on(missingno documentation)**

So the conclusion after these analysis charts is that there are no strong correlation between the missingness in features. First of all, due to small amount of missing values.

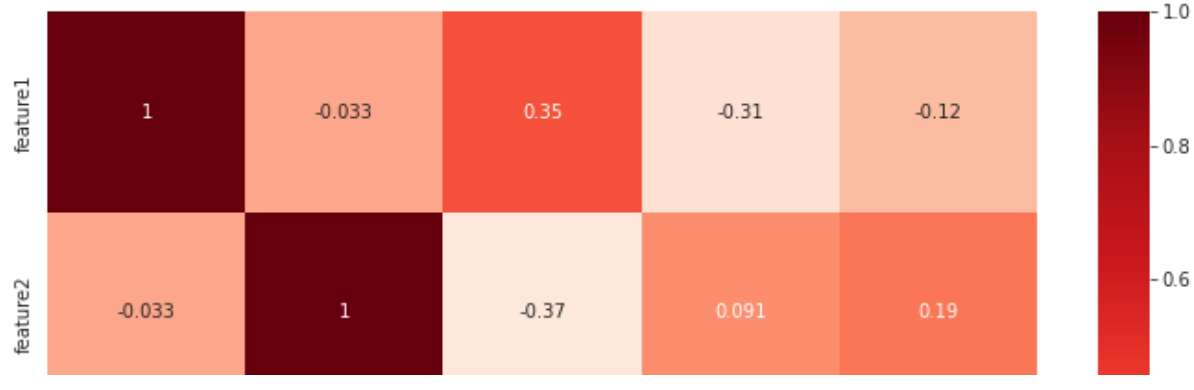
Feature Correlation heatmap

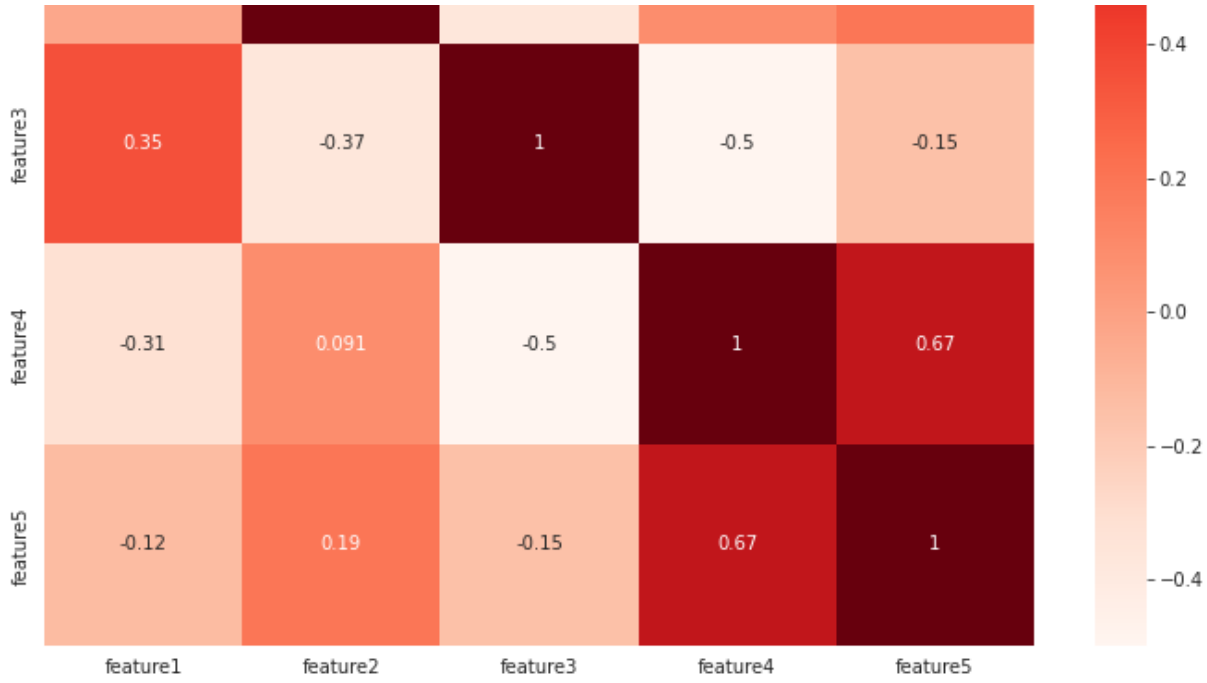
In [69]:

```
def draw_corr_mat(df):
    plt.figure(figsize=(12,10))
    cor = df.corr()
    sns.heatmap(cor, annot=True, cmap=plt.cm.Reds)
    plt.show()
```

In [70]:

```
draw_corr_mat(df)
```





By the way, some features itself have strong enough correlation. For example:

- **feature4 and feature5 have strong positive correlation with 0.67 value**
- **feature4 and feature3 in contrast have negative correlation with 0.5 value**

4. Imputation Methods

There are a lot of missing data imputation scenarios. In this lab the following methods will be implemented:

1. **Basic Data Imputatuion Techniques**
   - **With a constant value**
   - **With a descriptive statistics (mean, median)**
2. **Advanced Data Imputatuion Techniques**
   - **EM Algorithm**
   - **KNN Imputater**
   - **MissForest Imputer**

In [81]:

```
def draw_data_distrib(df):
    fig, axes = plt.subplots(1, 5, figsize=(10,4.5), dpi=140, sharex=True, sharey=True)
    colors = ['dodgerblue', 'orange', 'deeppink', 'green', 'red']
    for idx, (feature, ax, color) in enumerate(zip(df, axes, colors)):
        sns.distplot(df[feature], ax=ax, color=color)
        axes[idx].axvline(x=df[feature].median(),
                color='blue',
                ls='--',
                lw=1.7,
                label='median')

        axes[idx].axvline(x=df[feature].mean(),
                    color='red',
                    ls='--',
                    lw=1.7,
                    label='mean')
        axes[idx].legend()
    plt.tight_layout()
```

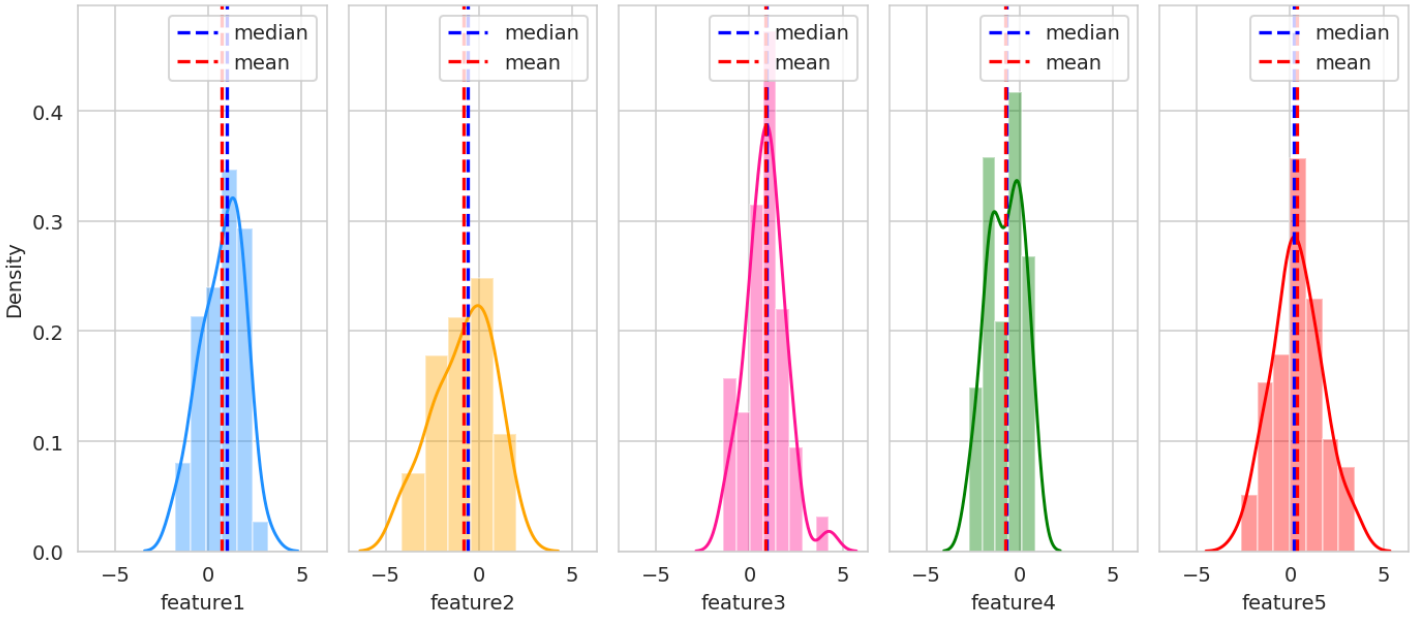
Check the data distribution

In [82]:

```
```

```
draw_data_distrib(df)
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



The data looks to be is not skewed. So, in case if data right or left skewed(long tail in some side) the mean imputation may be not the best choise.

5, 6. Basic Data Imputatuion Techniques

Random Constant Imputation

In [20]:

```
from sklearn.impute import SimpleImputer
import random
constant_fill_value = random.uniform(-10, 10)
logger.info(f"The data will be filled with {constant_fill_value}")
contant_imputer = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value=constant_fill_value)

contant_imputer.fit(df)
filled_constant_np_data = contant_imputer.transform(df)
```

```
2021-09-27 11:59:30.802 | INFO     | __main__:<module>:4 - The data will be filled with -
```

```
8.623089224747451
```

In [21]:

```
constant_filled_df = pd.DataFrame(data=filled_constant_np_data,columns=col_names)
```

In [22]:

```
constant_filled_df.head()
```

Out[22]:

	feature1	feature2	feature3	feature4	feature5
0	1.117915	-2.065559	-8.623089	-1.362986	-0.709196
1	-0.503055	-2.892927	0.713895	-1.631237	-1.790220
2	-8.623089	-0.077614	0.772394	-1.500325	-0.164499
3	1.163245	1.046673	1.674523	-1.867193	-0.534950
4	0.185329	-2.824514	2.036840	-2.040093	-8.623089

Mean Imputation

In [23]:

```
mean_imputer = SimpleImputer(missing_values=np.nan, strategy='mean')

mean_imputer.fit(df)
filled_mean_np_data = mean_imputer.transform(df)
mean_filled_df = pd.DataFrame(data=filled_mean_np_data,columns=col_names)
```

In [24]:

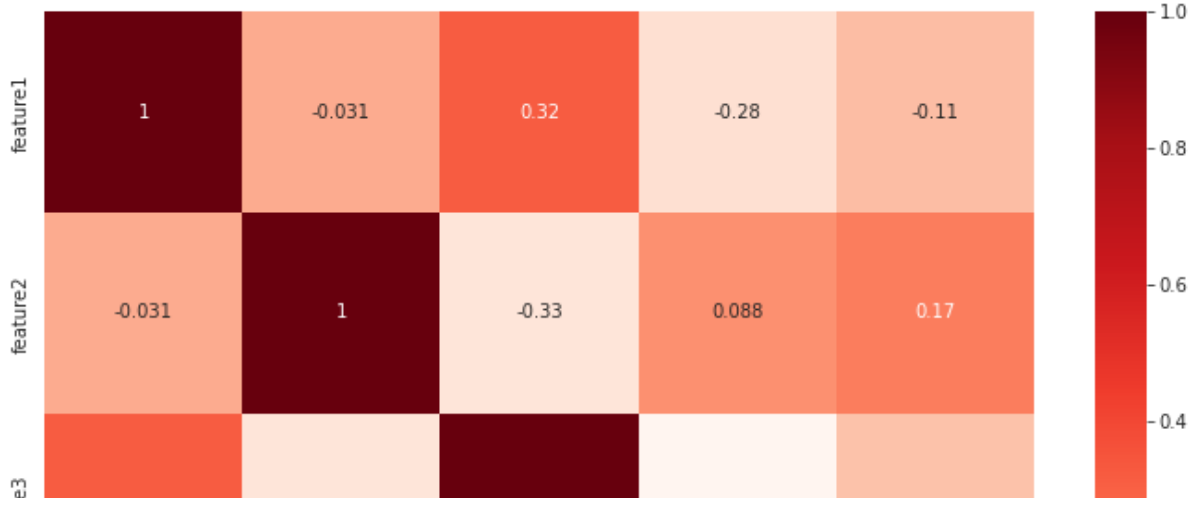
```
mean_filled_df.head()
```

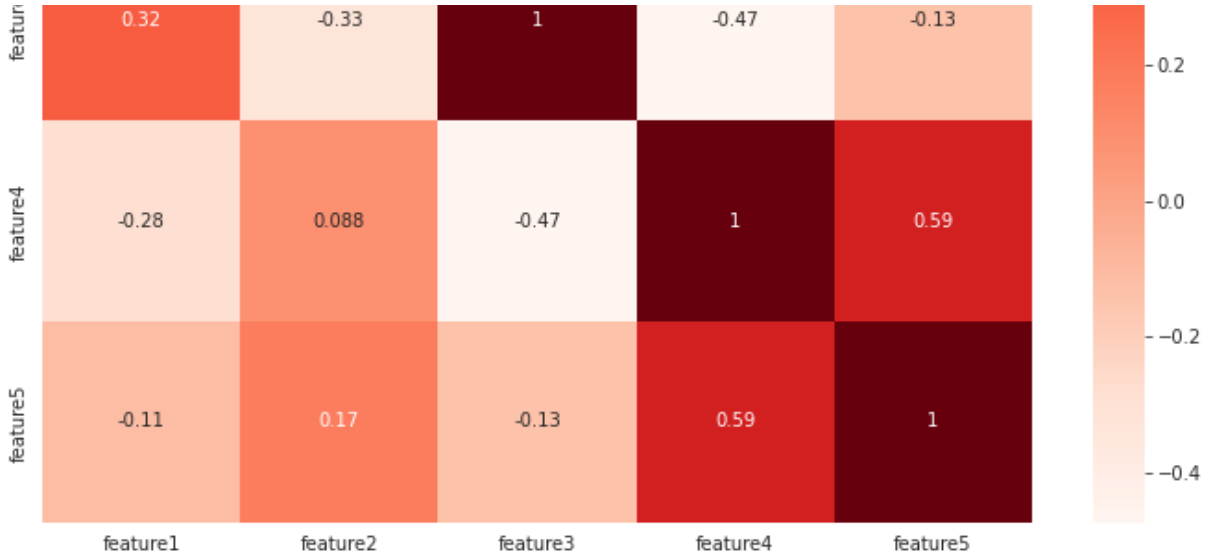
Out[24]:

	feature1	feature2	feature3	feature4	feature5
0	1.117915	-2.065559	0.867845	-1.362986	-0.709196
1	-0.503055	-2.892927	0.713895	-1.631237	-1.790220
2	0.731691	-0.077614	0.772394	-1.500325	-0.164499
3	1.163245	1.046673	1.674523	-1.867193	-0.534950
4	0.185329	-2.824514	2.036840	-2.040093	0.388719

In [71]:

```
draw_corr_mat(mean_filled_df)
```

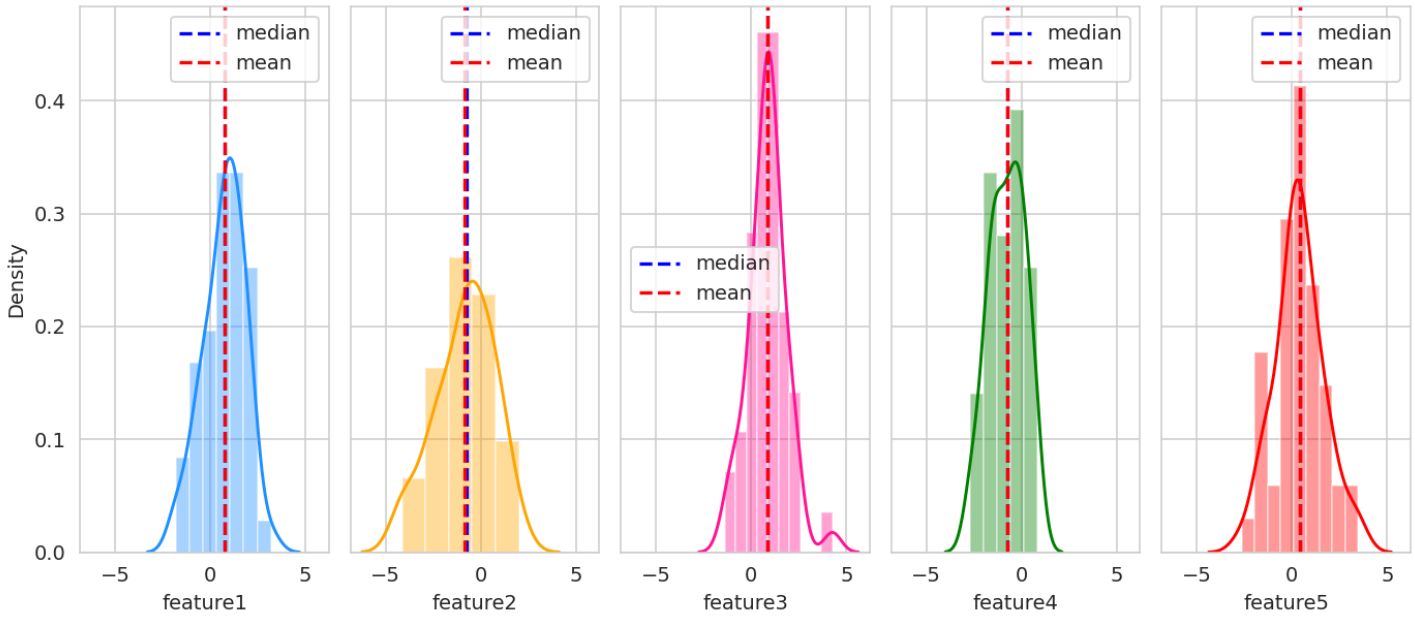




In [83]:

```
draw_data_distrib(mean_filled_df)
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



Median Imputation

In [25]:

```
median_imputer = SimpleImputer(missing_values=np.nan, strategy='median')

median_imputer.fit(df)
filled_median_np_data = median_imputer.transform(df)
median_filled_df = pd.DataFrame(data=filled_median_np_data,columns=col_names)
```

In [26]:

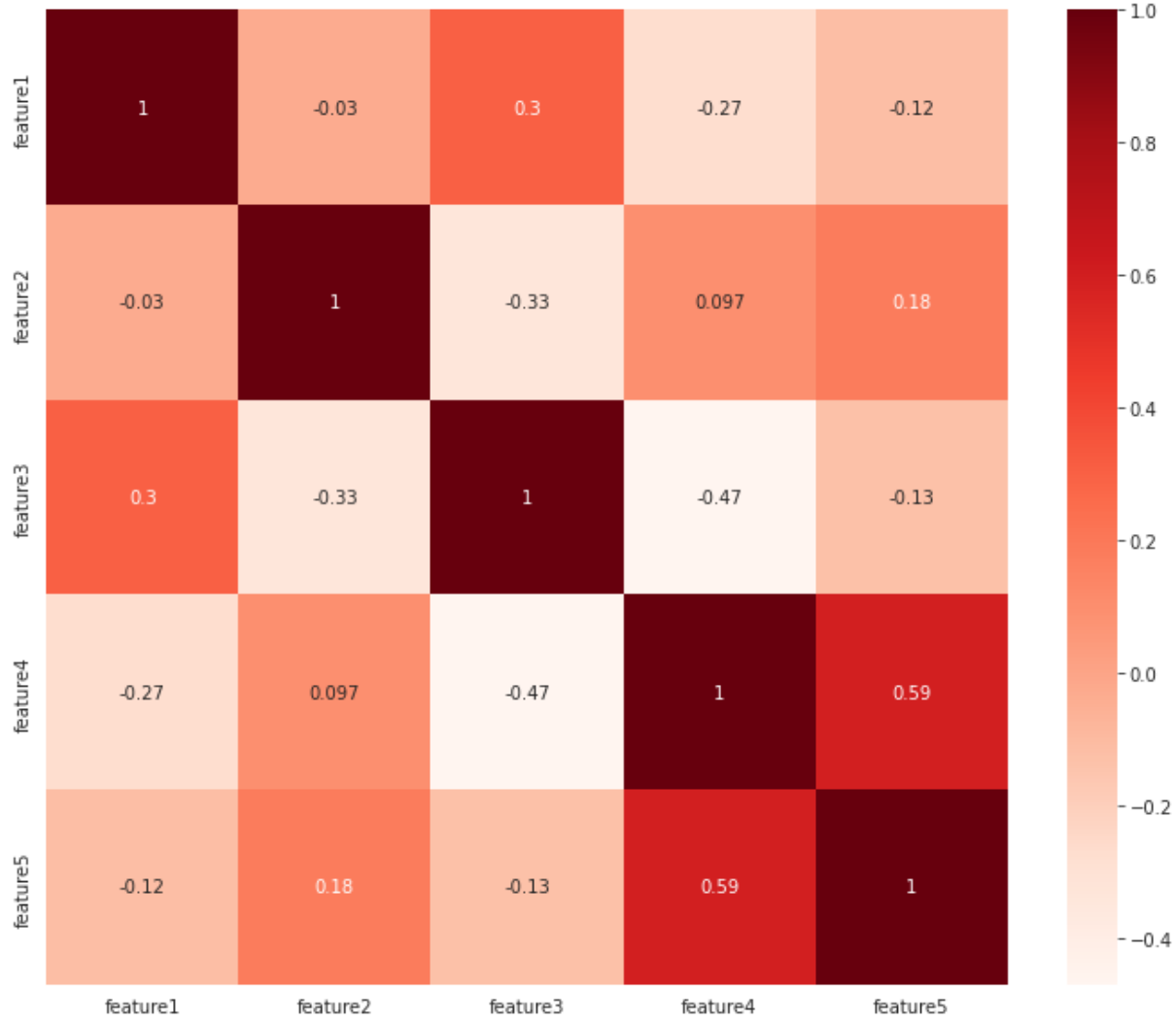
```
median_filled_df.head()
```

Out[26]:

	feature1	feature2	feature3	feature4	feature5
0	1.117915	-2.065559	0.946280	-1.362986	-0.709196
1	-0.503055	-2.892927	0.713895	-1.631237	-1.790220
2	0.991282	-0.077614	0.772394	-1.500325	-0.164499
3	1.163245	1.046673	1.674523	-1.867193	-0.534950
4	0.185329	-2.824514	2.036840	-2.040093	0.222050

In [74]:

```
draw_corr_mat(median_filled_df)
```

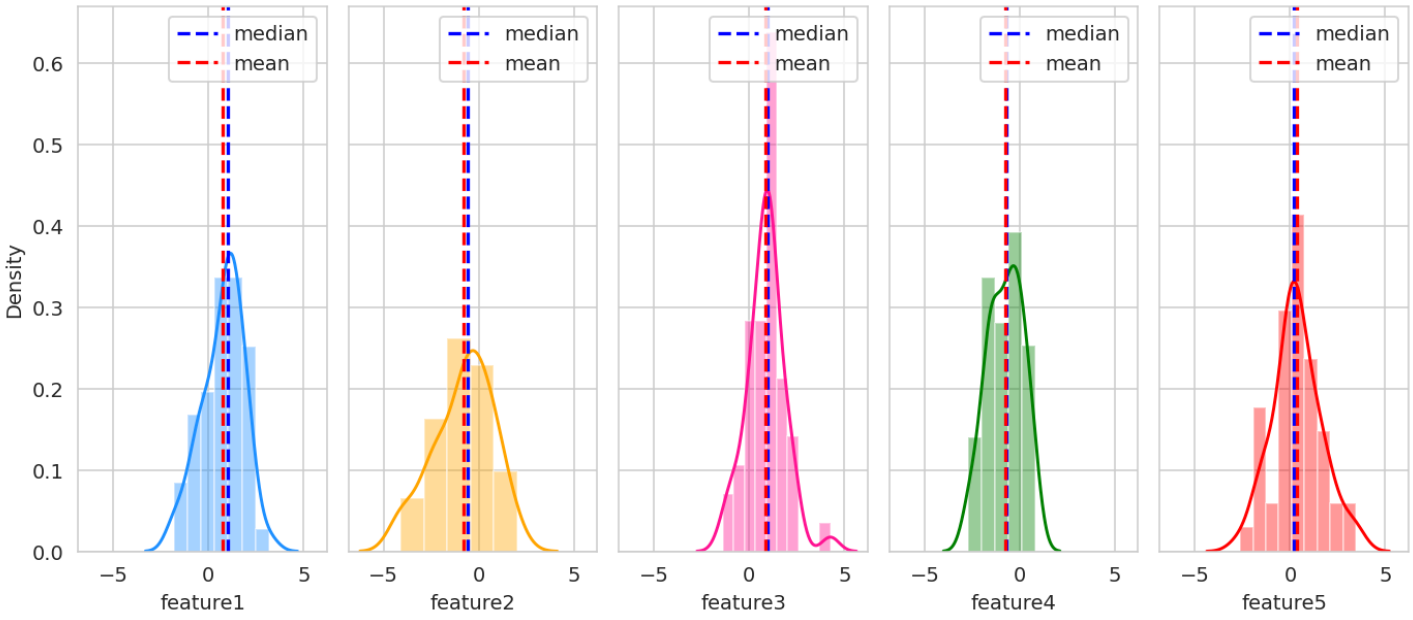


In [84]:

```
draw_data_distrib(median_filled_df)
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
```

```
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



7. Advanced Data Imputatuion Techniques

EM Imputation

In [27]:

```
import impyute as impy
em_data = impy.em(df.to_numpy())
em_filled_df= pd.DataFrame(data=em_data, columns = df.columns)
display(em_filled_df.head())
```

	feature1	feature2	feature3	feature4	feature5
0	1.117915	-2.065559	2.296486	-1.362986	-0.709196
1	-0.503055	-2.892927	0.713895	-1.631237	-1.790220
2	0.973267	-0.077614	0.772394	-1.500325	-0.164499
3	1.163245	1.046673	1.674523	-1.867193	-0.534950
4	0.185329	-2.824514	2.036840	-2.040093	0.053678

KNN Imputation

Using this method the missing values will be replaced by the mean value of N nearest neighbors measured by Euclidean distance.

In [28]:

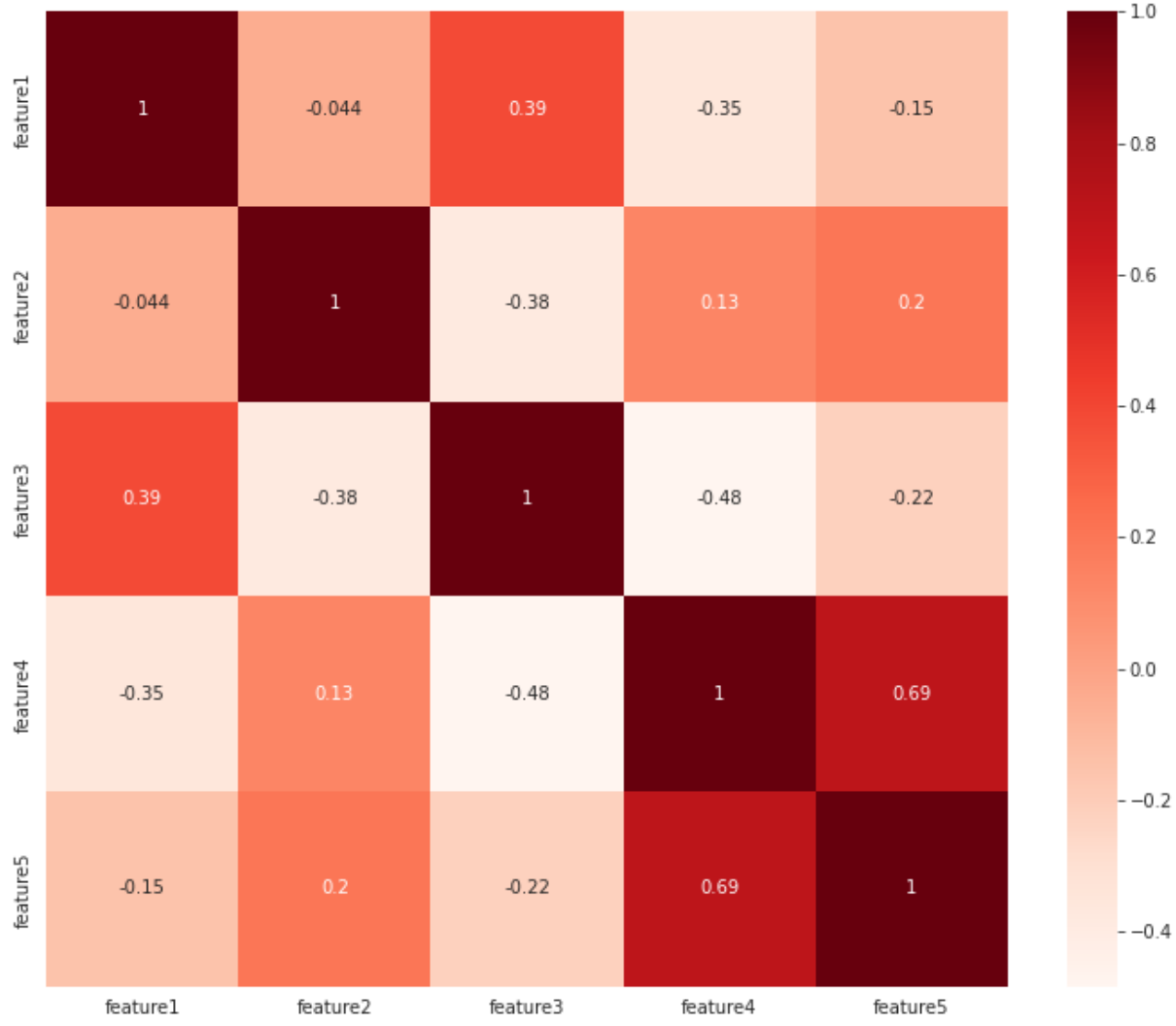
```
from sklearn.impute import KNNImputer
knn_imputer = KNNImputer(n_neighbors=5)
knn_filled_df = pd.DataFrame(knn_imputer.fit_transform(df),columns = df.columns)
knn_filled_df.head()
```

Out[28]:

	feature1	feature2	feature3	feature4	feature5
0	1.117915	-2.065559	1.973483	-1.362986	-0.709196
1	-0.503055	-2.892927	0.713895	-1.631237	-1.790220
2	1.413207	-0.077614	0.772394	-1.500325	-0.164499
3	1.163245	1.046673	1.674523	-1.867193	-0.534950
4	0.185329	-2.824514	2.036840	-2.040093	-0.709077

In [73]:

```
draw_corr_mat(knn_filled_df)
```

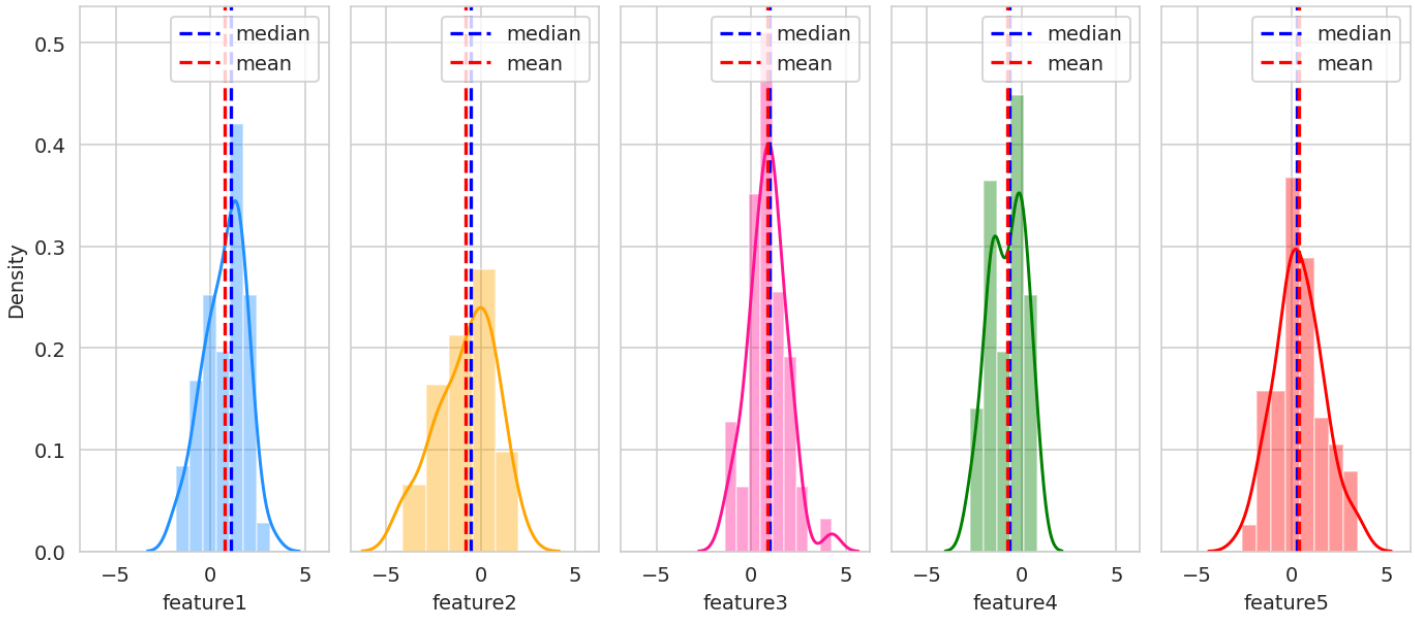


In [85]:

```
draw_data_distrib(knn_filled_df)
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
```

```
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



KNN Imputation

Advantages:

- **It's a great step from the simple average or median imputation**

Disadvantages:

- **Need to choice the K value**
- **It is sensitive to outliers**

MissForest Imputation(Random Forest imputation)

In [29]:

```
from missingpy import MissForest
random_forest_imputer = MissForest()
rf_filled_np = random_forest_imputer.fit_transform(df)
rf_filled_df = pd.DataFrame(rf_filled_np,columns = df.columns)
rf_filled_df.head()
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/sklearn/utils/de
precation.py:144: FutureWarning: The sklearn.neighbors.base module is  deprecated in vers
ion 0.22 and will be removed in version 0.24. The corresponding classes / functions shoul
d instead be imported from sklearn.neighbors. Anything that cannot be imported from sklea
```

```
rn.neighbors is now part of the private API.
  warnings.warn(message, FutureWarning)
```

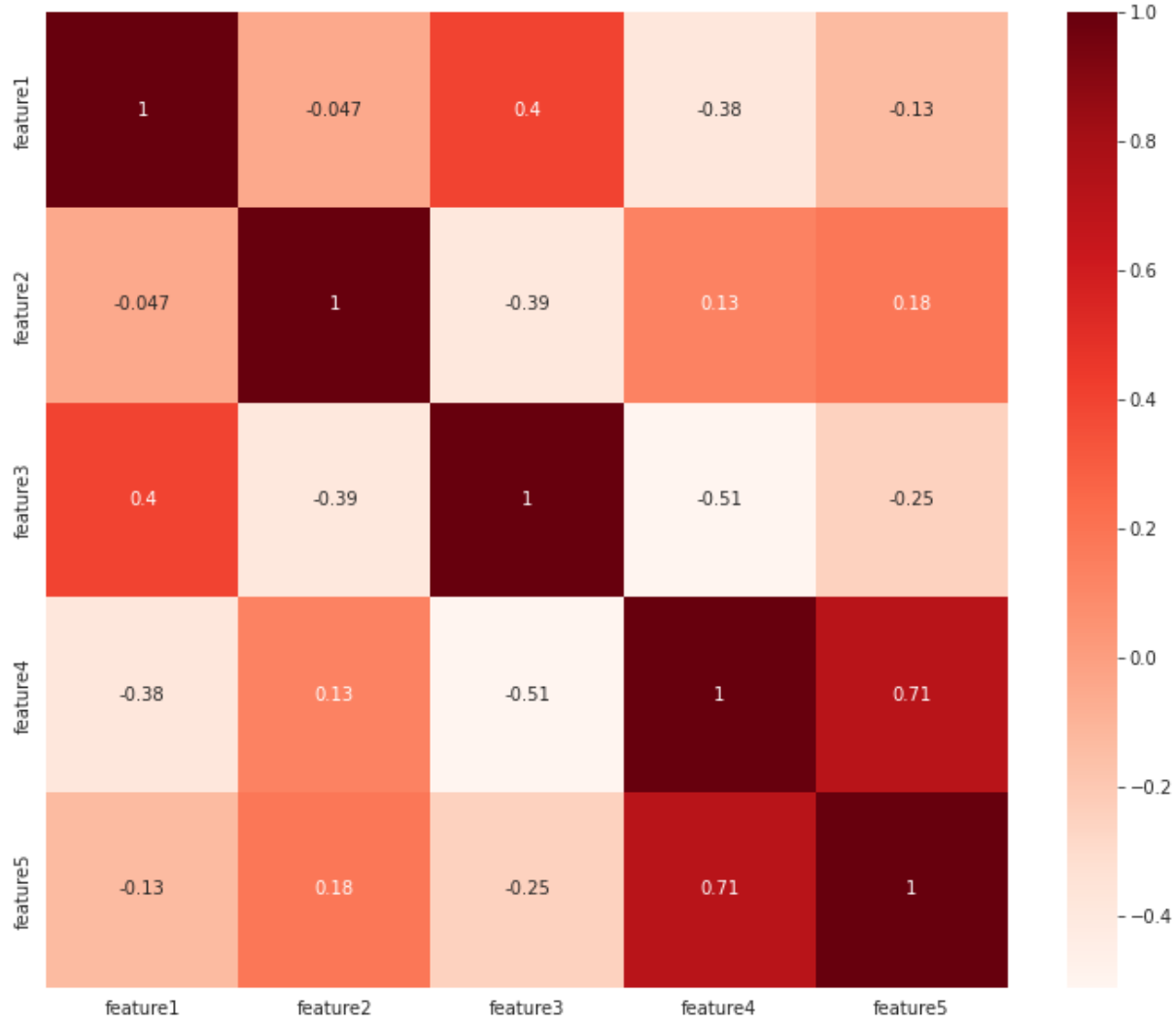
```
Iteration: 0
Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
```

Out[29]:

	feature1	feature2	feature3	feature4	feature5
0	1.117915	-2.065559	2.192582	-1.362986	-0.709196
1	-0.503055	-2.892927	0.713895	-1.631237	-1.790220
2	1.031663	-0.077614	0.772394	-1.500325	-0.164499
3	1.163245	1.046673	1.674523	-1.867193	-0.534950
4	0.185329	-2.824514	2.036840	-2.040093	-1.523016

In [76]:

```
draw_corr_mat(rf_filled_df)
```

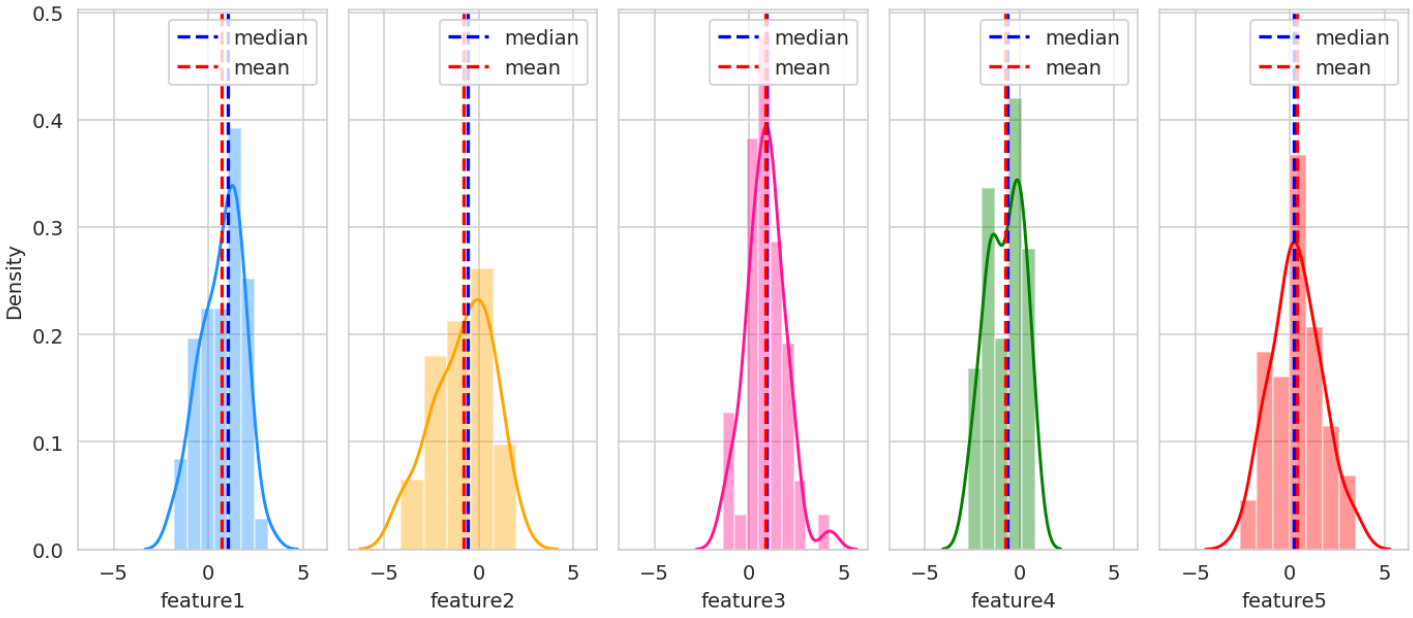


In [86]:

```
draw_data_distrib(rf_filled_df)
```

```
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
```

```
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/alexander/HSE_Stuff/envs/data_analysis/lib/python3.7/site-packages/seaborn/distribu
tions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



8. Outlier Detection

IQR - Outlier Detection per column

The interquartile range (IQR), also called the midspread or middle 50%, or technically H-spread, is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, IQR = Q3 − Q1.

In other words, the IQR is the first quartile subtracted from the third quartile; these quartiles can be clearly seen on a box plot on the data.

It is a measure of the dispersion similar to standard deviation or variance, but is much more robust against outliers.

In [56]:

```
from collections import namedtuple
filled_dfms = [constant_filled_df, mean_filled_df,median_filled_df, knn_filled_df, em_fil
led_df, rf_filled_df]
fill_types = ["Constant Value", "Mean", "Median", "KNN", "EM", "MissForrest"]
iqrs = []
for fill_type, data in zip(fill_types, filled_dfms):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    percentile_data = namedtuple('Data' , 'Q1 Q3 IQR')
    iqrs.append(percentile_data(Q1, Q3, IQR))
```

```
igrs
```

Out[56]:

```
[Data(Q1=feature1   -0.468975
 feature2   -2.286625
 feature3   -0.009836
 feature4   -1.578402
 feature5   -0.927164
 Name: 0.25, dtype: float64, Q3=feature1    1.409376
 feature2    0.318400
 feature3    1.311851
 feature4   -0.053578
 feature5    1.114915
 Name: 0.75, dtype: float64, IQR=feature1    1.878351
 feature2    2.605025
 feature3    1.321686
 feature4    1.524824
 feature5    2.042079
 dtype: float64),
 Data(Q1=feature1    0.033257
 feature2   -2.026360
 feature3    0.226913
 feature4   -1.528697
 feature5   -0.257579
 Name: 0.25, dtype: float64, Q3=feature1    1.409376
 feature2    0.318400
 feature3    1.311851
 feature4   -0.053578
 feature5    1.114915
 Name: 0.75, dtype: float64, IQR=feature1    1.376119
 feature2    2.344760
 feature3    1.084938
 feature4    1.475118
 feature5    1.372494
 dtype: float64),
 Data(Q1=feature1    0.033257
 feature2   -2.026360
 feature3    0.226913
 feature4   -1.528697
 feature5   -0.257579
 Name: 0.25, dtype: float64, Q3=feature1    1.409376
 feature2    0.318400
 feature3    1.311851
 feature4   -0.053578
 feature5    1.114915
 Name: 0.75, dtype: float64, IQR=feature1    1.376119
 feature2    2.344760
 feature3    1.084938
 feature4    1.475118
 feature5    1.372494
 dtype: float64),
 Data(Q1=feature1    0.033257
 feature2   -2.026360
 feature3    0.168452
 feature4   -1.545083
 feature5   -0.337801
 Name: 0.25, dtype: float64, Q3=feature1    1.412721
 feature2    0.357034
 feature3    1.408932
 feature4   -0.041452
 feature5    1.116440
 Name: 0.75, dtype: float64, IQR=feature1    1.379465
 feature2    2.383394
 feature3    1.240480
 feature4    1.503631
 feature5    1.454241
 dtype: float64),
 Data(Q1=feature1   -0.215097
 feature2   -2.026360
 feature3    0.168452
 feature4   -1.528697
 feature5   -0.293738
```

```
Name: 0.25, dtype: float64, Q3=feature1    1.409376
feature2    0.386928
feature3    1.536907
feature4   -0.041452
feature5    1.244879
Name: 0.75, dtype: float64, IQR=feature1    1.624472
feature2    2.413288
feature3    1.368456
feature4    1.487244
feature5    1.538617
dtype: float64),
Data(Q1=feature1   -0.047738
feature2   -2.077529
feature3    0.168452
feature4   -1.564045
feature5   -0.471671
Name: 0.25, dtype: float64, Q3=feature1    1.409376
feature2    0.357034
feature3    1.408932
feature4   -0.041452
feature5    1.239056
Name: 0.75, dtype: float64, IQR=feature1    1.457114
feature2    2.434563
feature3    1.240480
feature4    1.522593
feature5    1.710726
dtype: float64)]
```

In [57]:

```
outlier_indexes = {imputer: [] for imputer in fill_types}
for percentile_data, data, fill_type in zip(igrs, filled_dfms, fill_types):
    indexes = (data < (percentile_data.Q1 - 1.5 * percentile_data.IQR) ) | (data > (perc
entile_data.Q3 + 1.5 * percentile_data.IQR))
    indexes.replace({False: 'blue', True: 'red'}, inplace=True)
    outlier_indexes[fill_type] = np.array(indexes.values.tolist()).T.tolist()
```

In [58]:

```
outlier_indexes.items()
```

Out[58]:

```
dict_items([('Constant Value', [['blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', '
blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'bl
ue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue
', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue'], ['blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'b
lue', 'blue', 'blue'], ['red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 're
d', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red',
'blue'], ['blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'red', 'blue', 'blue']]), ('Mean', [['blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
```

, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'red', 'blue', 'blue', 'blue']])
, ('Median', [['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'bl
ue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue
', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'],
['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', '
blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'bl
ue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue
', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'red', 'red', 'blue', 'blue', 'blue']]), ('KNN', [['blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue
', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'red', 'red', 'blue', 'blue', 'blue']]), ('EM', [['
blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'bl
ue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue
', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu

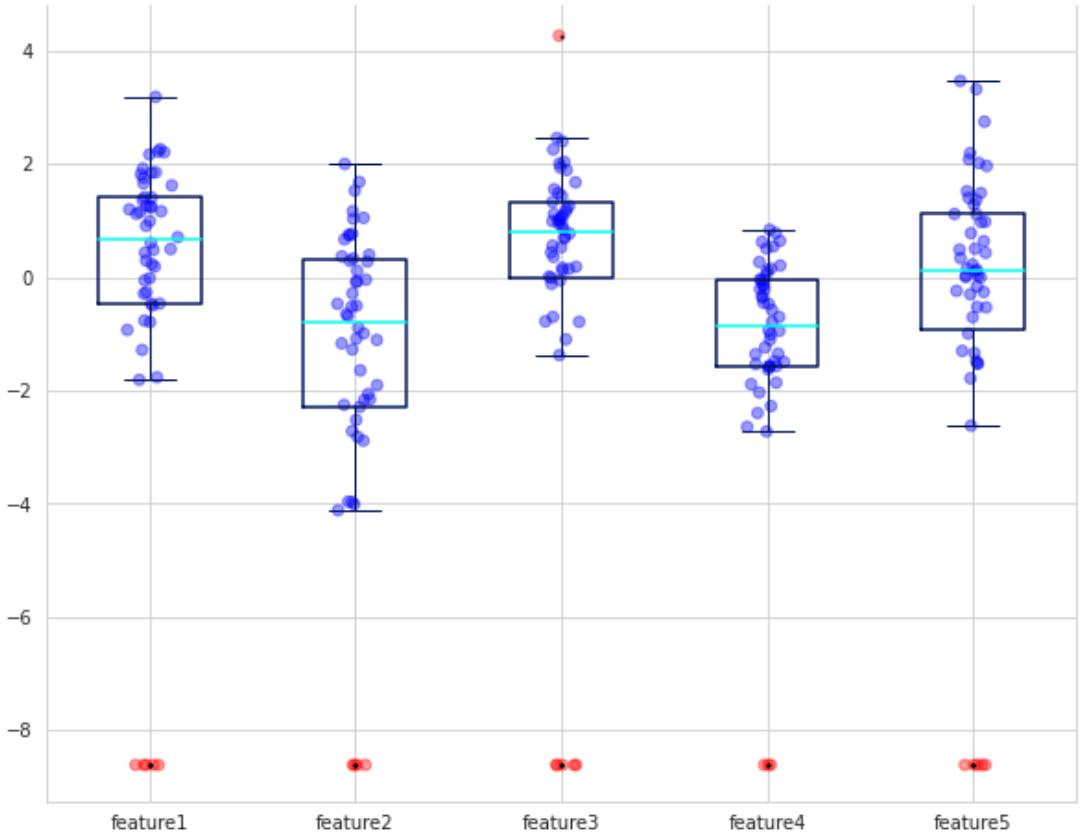
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue
', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'red',
'blue', 'blue', 'blue']]), ('MissForrest', [['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'red', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'], ['blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue'], ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue',
'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'b
lue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blu
e', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue'
, 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue']])])

In [59]:

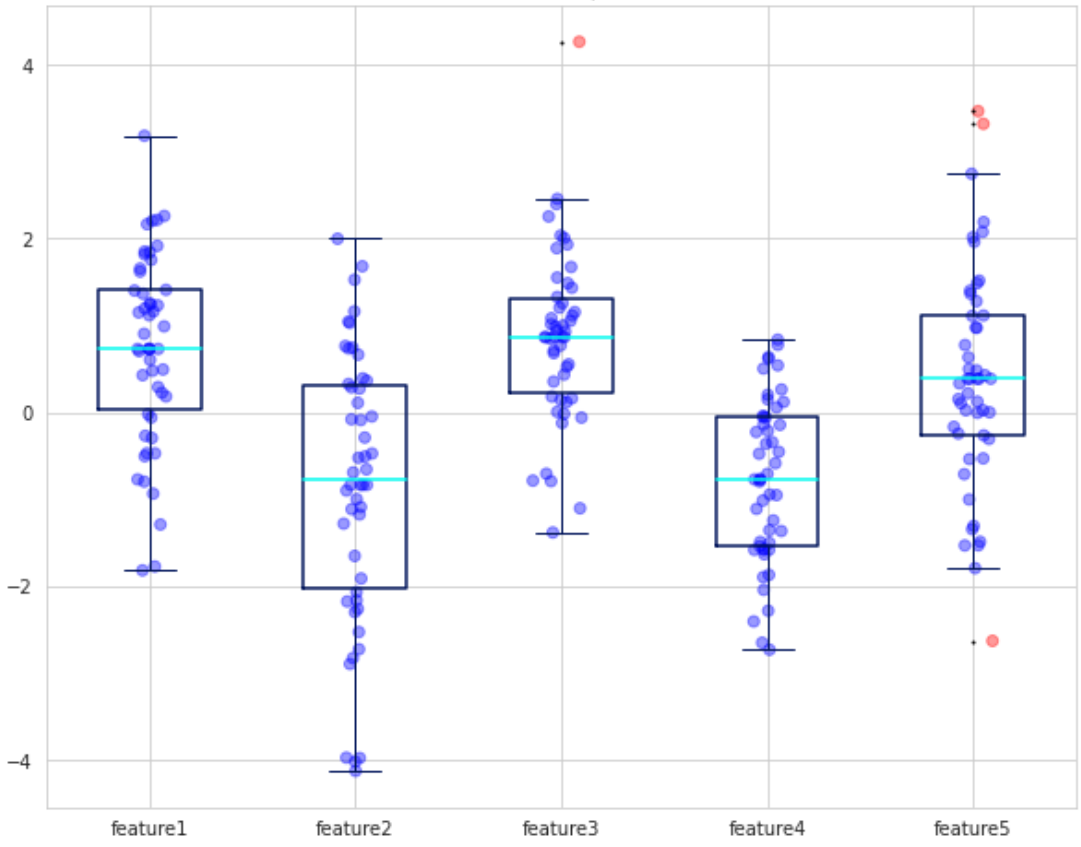
```
def draw_outliers_iqr(list_data, outlier_indexes, title):
    for data, (imputer, indexes) in zip(list_data, outlier_indexes.items()):
        fig, ax = plt.subplots(figsize=(10, 8))
        vals, names, xs = [], [], []
        for i, col in enumerate(data.columns):
            vals.append(data[col].values)
            names.append(col)
            xs.append(np.random.normal(i + 1, 0.04, data[col].values.shape[0]))

        sns.set_style("whitegrid")
        boxprops = dict(linestyle='-', linewidth=1.5, color='#00145A')
        flierprops = dict(marker='o', markersize=1,
                          linestyle='none')

        whiskerprops = dict(color='#00145A')
        capprops = dict(color='#00145A')
        medianprops = dict(linewidth=1.5, linestyle='-', color='#01FBEE')

        plt.boxplot(vals, labels=names, notch=False,
                    boxprops=boxprops, whiskerprops=whiskerprops,
                    capprops=capprops, flierprops=flierprops,
                    medianprops=medianprops,showmeans=False)

        for x, val, ind in zip(xs, vals, indexes):
            plt.scatter(x, val, alpha=0.4, color=ind)

        ax.set_title(f"{title} on {imputer} Imputer data")
        plt.show()
```

In [60]:

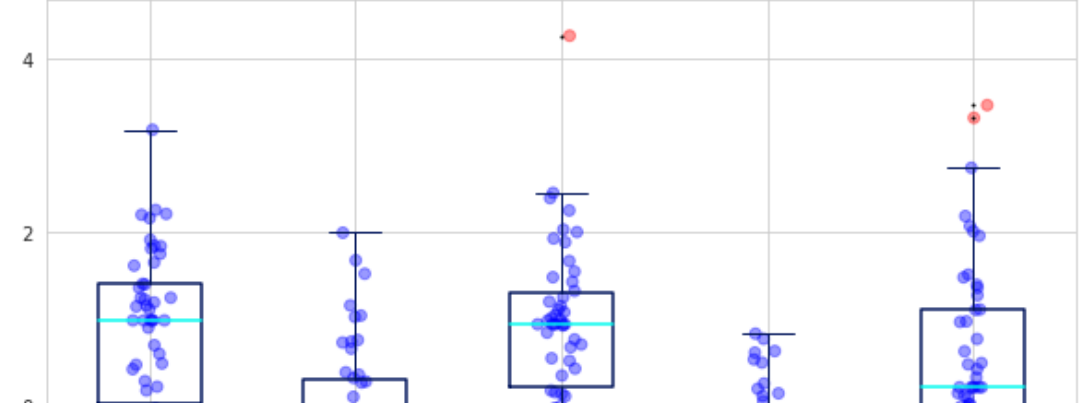
```
draw_outliers_iqr(filled_dfms, outlier_indexes, "IQR")
```

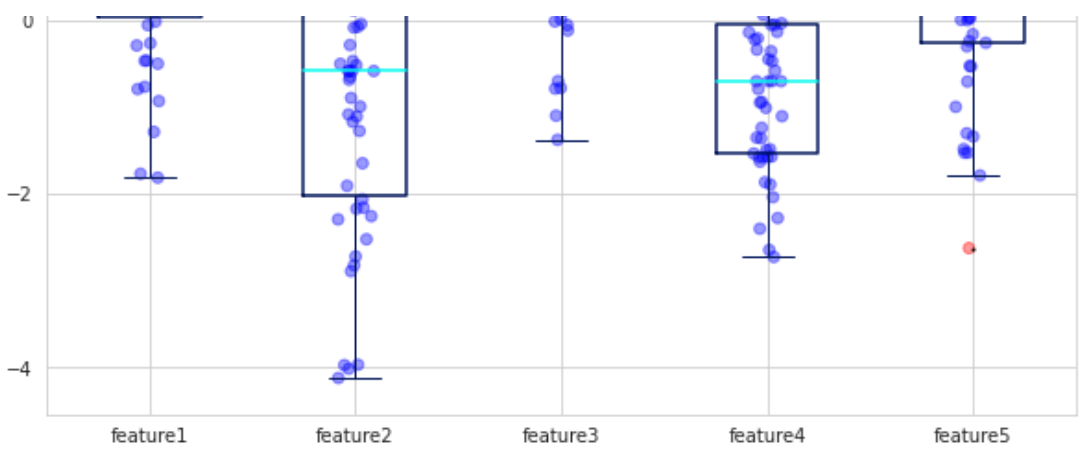


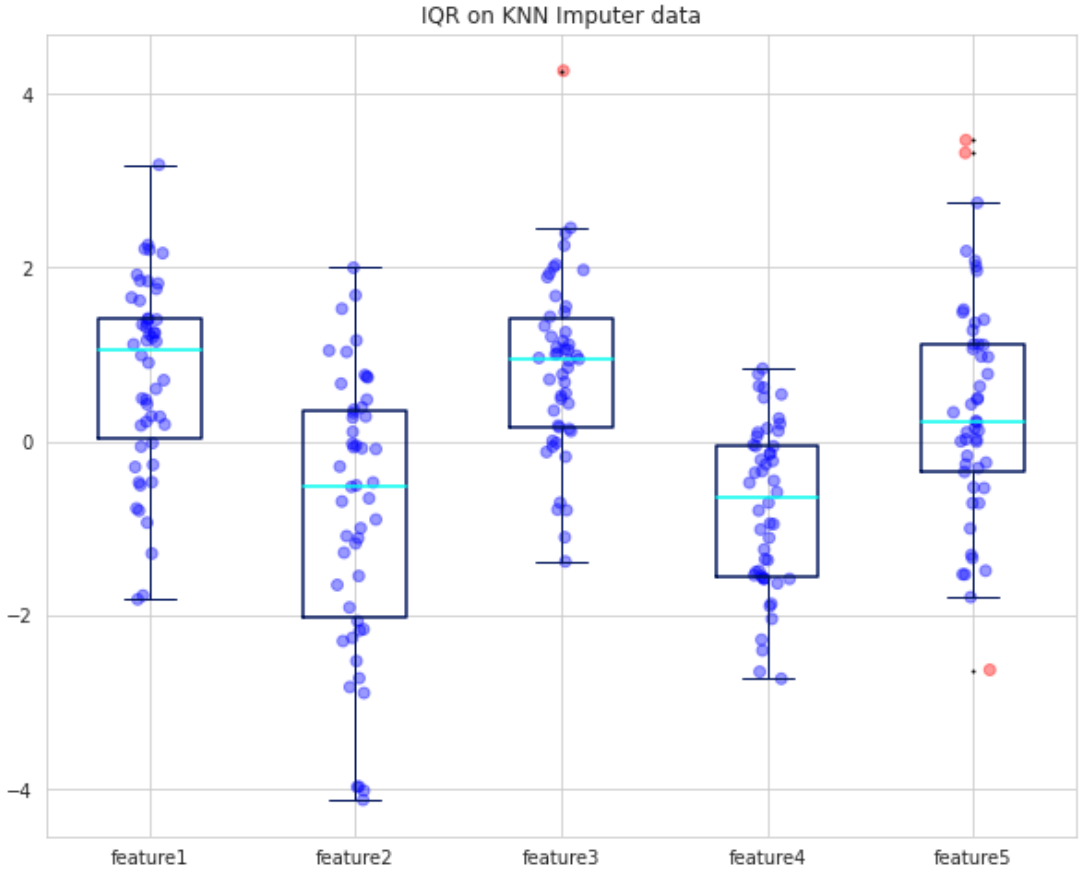
IQR on Mean Imputer data

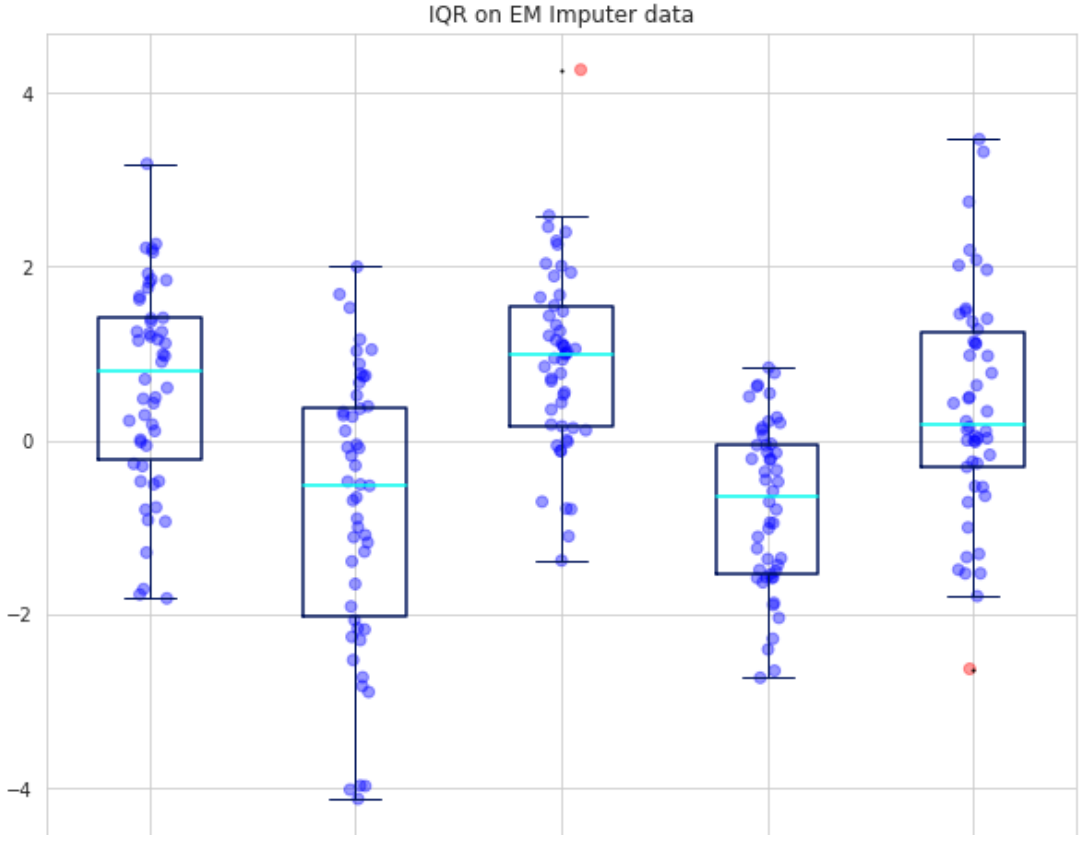


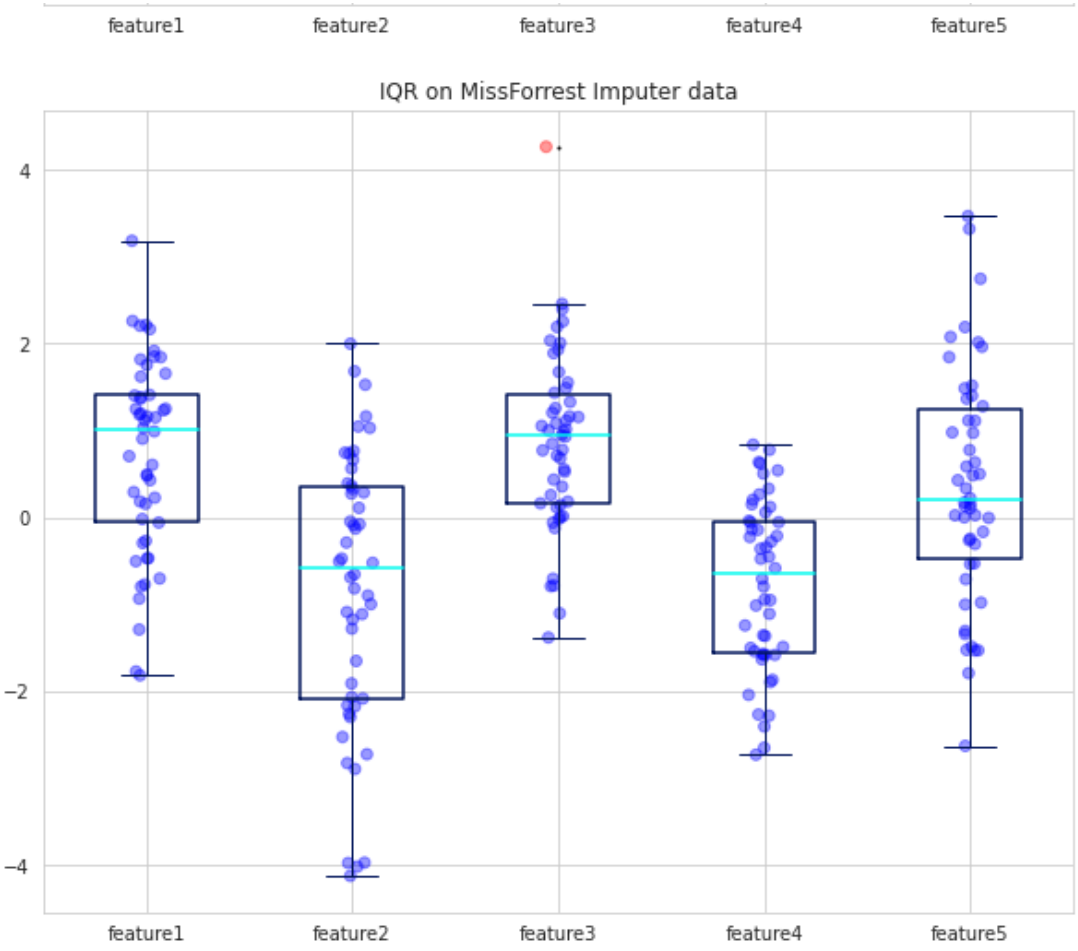
IQR on Median Imputer data




0
−2
−4
feature1
feature2
feature3
feature4
feature5


IQR on KNN Imputer data
4
2
0
−2
−4
feature1
feature2
feature3
feature4
feature5

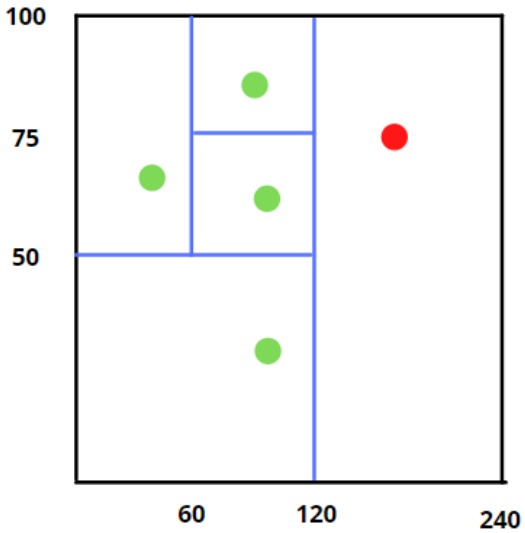

IQR on EM Imputer data
4
2
0
−2
−4

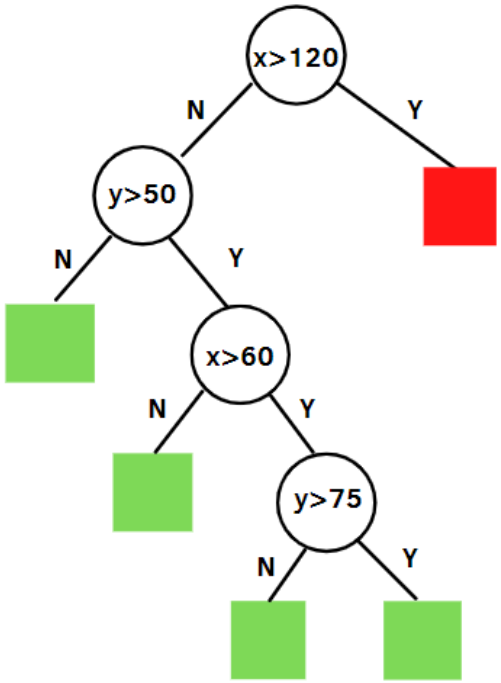


Isolation Forest - Outlier Detection per sample

Isolation is the keyword of this algorithm because it isolates anomalies from the rest of the observations. This isolation procedure separates all the data points by randomly splitting the region into smaller pieces.Isolation Forest identifies anomalies as the observations with short average path lengths on the isolation trees . There is a procedure applied for each isolation tree:

1. **Randomly select two features.**
2. **Split the data points by randomly selecting a value between the minimum and the maximum of the selected features.**





In [61]:

```
# prepare ground truths
ground_truths = pd.isnull(df).any(1).to_numpy(dtype=int)
```

In [62]:

```
ground_truths
```

Out[62]:

```
array([1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 0, 1, 1, 0])
```

In [63]:

```
from sklearn.metrics import f1_score
def map_outliers_score_to_binary_labels(scores):
    '''Map Outliers Score of Isolation Forest algo into Classification Binary Labels.
    Parameters
    ----------
    outliers_scores: 1d array-like with outliers scores, where -1 indicate outlier, 1 nor
mal data.
    Returns
    -------
    labels: 1d array-like with binary labels, 1 - outlier,0 - normal data

    '''
    labels = scores.copy()
    labels[labels == 1] = 0
    labels[labels == -1] = 1
    return labels

def get_outlier_detection_accuracy(ground_truths, outliers_scores, nan_fill_type="mean"):
    f1 = f1_score(ground_truths, map_outliers_score_to_binary_labels(outliers_scores))
    return f1
```

In [64]:

```
# Isolation Forrest objects learning
from sklearn.ensemble import IsolationForest
outlier_detectors = {}
for fill_type, data in zip(fill_types, filled_dfms):
    iso = IsolationForest()
    iso.fit(data)
    outlier_detectors[fill_type] = iso
print(outlier_detectors)
```

```
{'Constant Value': IsolationForest(behaviour='deprecated', bootstrap=False, contamination
='auto',
                max_features=1.0, max_samples='auto', n_estimators=100,
                n_jobs=None, random_state=None, verbose=0, warm_start=False), 'Mean': Iso
lationForest(behaviour='deprecated', bootstrap=False, contamination='auto',
                max_features=1.0, max_samples='auto', n_estimators=100,
                n_jobs=None, random_state=None, verbose=0, warm_start=False), 'Median': I
solationForest(behaviour='deprecated', bootstrap=False, contamination='auto',
                max_features=1.0, max_samples='auto', n_estimators=100,
                n_jobs=None, random_state=None, verbose=0, warm_start=False), 'KNN': Isol
ationForest(behaviour='deprecated', bootstrap=False, contamination='auto',
                max_features=1.0, max_samples='auto', n_estimators=100,
                n_jobs=None, random_state=None, verbose=0, warm_start=False), 'EM': Isola
tionForest(behaviour='deprecated', bootstrap=False, contamination='auto',
                max_features=1.0, max_samples='auto', n_estimators=100,
                n_jobs=None, random_state=None, verbose=0, warm_start=False), 'MissForres
t': IsolationForest(behaviour='deprecated', bootstrap=False, contamination='auto',
                max_features=1.0, max_samples='auto', n_estimators=100,
                n_jobs=None, random_state=None, verbose=0, warm_start=False)}
```

In [66]:

```
from sklearn.decomposition import PCA
```

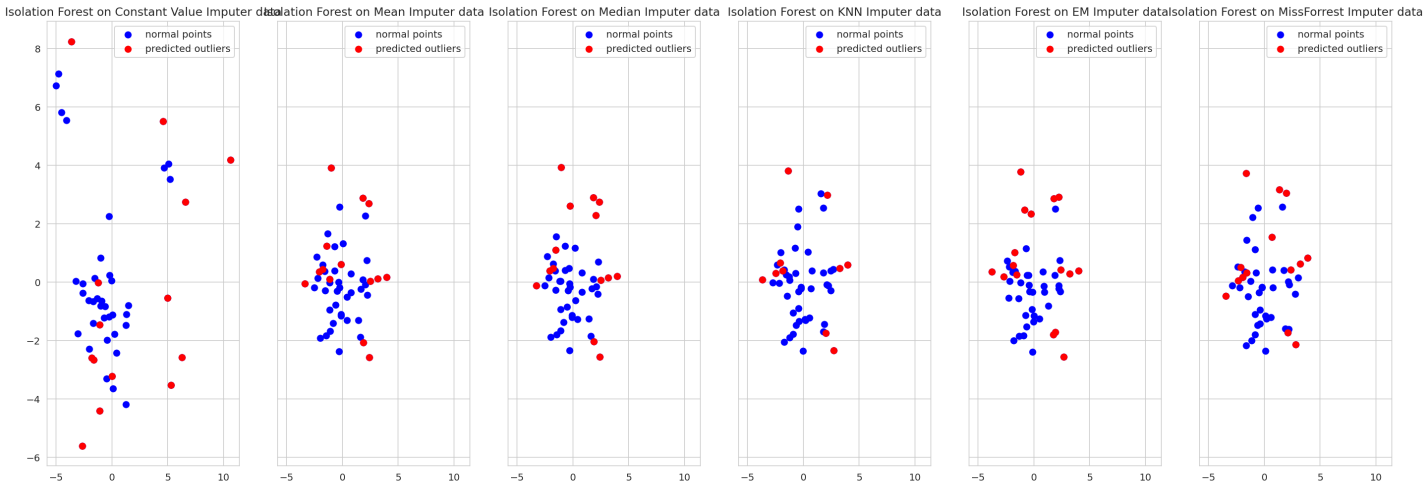
```
def draw_outliers(list_data, outlier_indexes, title):
    data_num = len(list_data)
    fig, axes = plt.subplots(1, data_num, figsize=(25,8.5), dpi=140, sharex=True, sharey
=True)
    for idx, (data, (imputer, indexes), ax) in enumerate(zip(list_data,outlier_indexes.i
tems(), axes)):
        pca = PCA(2)
        pca.fit(data)


        res=pd.DataFrame(pca.transform(data))

        Z = np.array(res)
        axes[idx].set_title(f"{title} on {imputer} Imputer data")

        b1 = axes[idx].scatter(res[0], res[1], c='blue',
                               s=40,label="normal points")

        b1 = axes[idx].scatter(res.iloc[indexes,0],res.iloc[indexes,1], c='red',
                               s=40,  edgecolor="red",label="predicted outliers")
        axes[idx].legend(loc="upper right")
```

In [68]:

```
f1_scores = {imputer: [] for imputer in fill_types}
outlier_indexes = {imputer: [] for imputer in fill_types}
num_iteration = 10
for _ in tqdm(range(num_iteration)):
    for fill_type, data in zip(fill_types, filled_dfms):
        outliers_score_constant = outlier_detectors[fill_type].predict(data)
        # get outliers
        outliers=data.loc[outliers_score_constant==-1]
        outlier_indexes[fill_type] = list(outliers.index)
        f1_scores[fill_type] = get_outlier_detection_accuracy(ground_truths, outliers_sc
ore_constant, nan_fill_type=fill_type)
for imputer, f1 in f1_scores.items():
    print(f"Outlier Detection F1 on data with {imputer} NaN Imputation is {round(f1.mean(
), 3)}%")

draw_outliers(filled_dfms, outlier_indexes, "Isolation Forest")
```

```
100%|██████████████████████████████████████████████████| 10/10 [00:01<00:00,  6.27it/s]
```

```
Outlier Detection F1 on data with Constant Value NaN Imputation is 0.8%
Outlier Detection F1 on data with Mean NaN Imputation is 0.286%
Outlier Detection F1 on data with Median NaN Imputation is 0.171%
Outlier Detection F1 on data with KNN NaN Imputation is 0.194%
Outlier Detection F1 on data with EM NaN Imputation is 0.27%
Outlier Detection F1 on data with MissForrest NaN Imputation is 0.286%
```



DBSCAN - Outlier Detection per sample

In [42]:

```
from sklearn.cluster import DBSCAN
```

find the median euclidean distance between sets:

In [43]:

```
df_list = mean_filled_df.to_numpy()
dist = []
for row1 in df_list:
    for row2 in df_list:
        dist.append(np.linalg.norm(row1-row2))

print('median distance between sets:', np.median(dist))
```

```
median distance between sets: 3.3862282321799437
```

In [44]:

```
def cluster_outliers(data_df, eps=3.0, min_samples=3):
    db = DBSCAN(eps=eps,
                min_samples=min_samples)
    db.fit(data_df)
    return db.labels_
```

In [45]:

```
df_clear = df.dropna()
```

In [49]:

```
filled_dfms = [df_clear, constant_filled_df, mean_filled_df,median_filled_df, knn_filled_
df, em_filled_df, rf_filled_df]
fill_types = ["Missing  data", "Constant Value", "Mean", "Median", "KNN", "EM", "MissFor
rest"]
outlier_indexes = {imputer: [] for imputer in fill_types}

for fill_type, data in zip(fill_types, filled_dfms):
    outlier_indexes[fill_type] = cluster_outliers(data, 2.5, 4)
```

In [50]:

```
def draw_outliers_dbscan(list_data, outlier_indexes, title):
    data_num = len(list_data)
    fig, axes = plt.subplots(1, data_num, figsize=(25,8.5), dpi=140, sharex=True, sharey
=True)
    for idx, (data, (imputer, indexes), ax) in enumerate(zip(list_data,outlier_indexes.i
tems(), axes)):
        pca = PCA(2)
        pca.fit(data)
        res=pd.DataFrame(pca.transform(data))

        Z = np.array(res)
        axes[idx].set_title(f"{title} with {imputer}")
        colors = []
        for ind in indexes:
            if ind == -1:
                colors.append('red')
            if ind == 0:
                colors.append('blue')
            if ind == 1:
                colors.append('green')
            if ind == 2:
                colors.append('yellow')
            if ind == 3:
                colors.append('brown')
        b1 = axes[idx].scatter(res[0], res[1], c=colors,
                               s=40)

        b1 = axes[idx].scatter(res.iloc[indexes,0],res.iloc[indexes,1], c='red',
                               s=40, edgecolor="red",label="predicted outliers")
```

```
        axes[idx].legend(loc="upper right")
```

In [51]:

```
draw_outliers_dbscan(filled_dfms, outlier_indexes, "DBSCAN")
```