remove_outliers_using_z-score

July 11, 2024

[307]:
```
import pandas as pd
df = pd.read_csv("Outliers_z-score.csv")
df
```

[307]:

	Unnamed: 0	height	age	weight	income	hours_per_week	years_experience	satisfaction	performance	projects_completed	bonus
0	0	174.967142	27.923146	75.366810	37565.074836	32.027862	12.778533	8.513977	69.772770	6.876568	4737.346618
1	1	168.617357	32.896773	78.411768	41597.284397	37.003125	15.728250	5.155669	85.490092	3.967911	3213.322375
2	2	176.476885	33.286427	86.245769	61209.404077	40.026218	5.804297	8.739212	67.956563	5.192242	4057.489646
3	3	185.230299	30.988614	85.807031	59155.553982	40.234903	11.688908	9.711276	60.915387	4.075449	6556.903725
4	4	167.658466	34.193571	49.334959	49686.476091	37.749673	8.048072	7.826870	59.433708	4.131008	4382.198136
..	…	…	…	…	…	…	…	…	…	…	…
97	97	172.610553	35.768626	74.609493	70308.085612	39.048307	7.063709	4.497773	81.815007		
98	98	170.051135	35.291044	82.192932	48281.902321	35.621909	8.667120	8.848054	75.283184		
99	99	167.654129	29.285149	79.444433	68567.244680	33.086001	11.131901	6.630196	75.297561		
100	100	250.000000	80.000000	150.000000	150000.000000	80.000000	30.000000	10.000000	100.000000		
101	101	120.000000	10.000000	30.000000	20000.000000	10.000000	1.000000	0.000000	50.000000		

```
97              5.293427   5281.685723
98              7.413018   2857.642020
99              3.366129   5145.165563
100            15.000000  20000.000000
101             0.000000      0.000000

[102 rows x 11 columns]
```

[306]:
```
df.drop(columns= "Unnamed: 0",inplace= True)
```

0.0.1 Calculate the Z-scores for the 'height' column and identify outliers.

[302]:
```
upper_limit = df["height"].mean() + 3 * df["height"].std()
lower_limit = df["height"].mean() - 3 * df["height"].std()
upper_limit , lower_limit
```

[302]:
```
(208.3337773191619, 130.2182517515421)
```

[303]:
```
df[(df["height"]>=upper_limit)|(df["height"]<=lower_limit)]
```

[303]:
```
     Unnamed: 0  height   age  weight    income  hours_per_week  \
100         100   250.0  80.0   150.0  150000.0            80.0
101         101   120.0  10.0    30.0   20000.0            10.0

     years_experience  satisfaction  performance  projects_completed    bonus
100              30.0          10.0        100.0                15.0  20000.0
101               1.0           0.0         50.0                 0.0      0.0
```

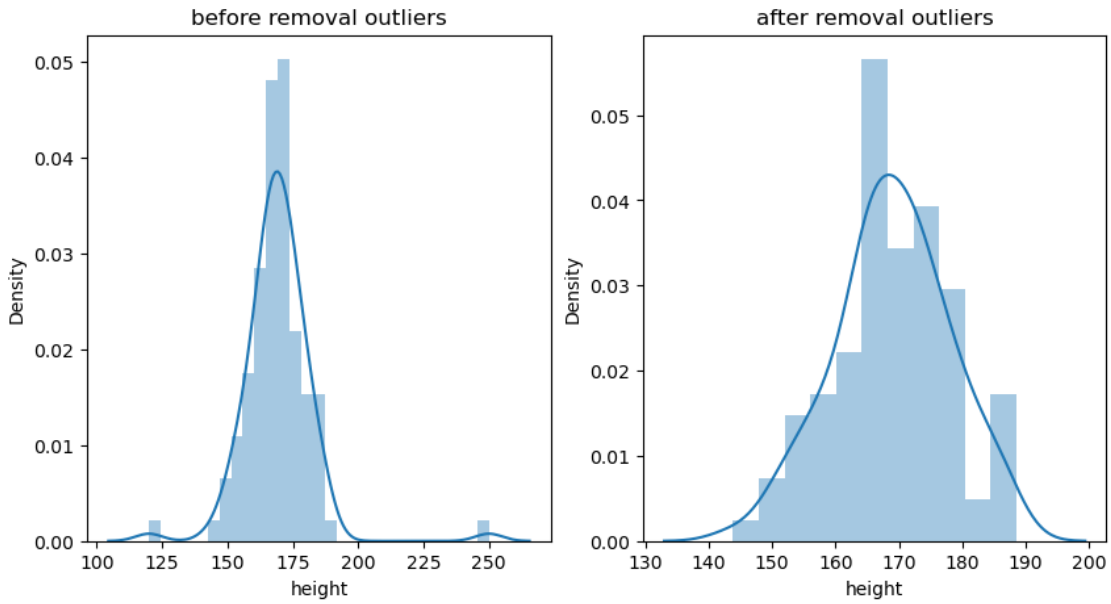
0.0.2 Remove the outliers from the 'height' column based on Z-scores and display the cleaned column.

[299]:
```
df_height = df[(df["height"]<=upper_limit )&(df["height"]>=lower_limit)]
df_height
```

[299]:
```
    Unnamed: 0      height        age     weight        income  \
0            0  174.967142  27.923146  75.366810  37565.074836
1            1  168.617357  32.896773  78.411768  41597.284397
2            2  176.476885  33.286427  86.245769  61209.404077
3            3  185.230299  30.988614  85.807031  59155.553982
4            4  167.658466  34.193571  49.334959  49686.476091
..         ...         ...        ...        ...           ...
95          95  155.364851  36.926587  59.606356  42962.365218
96          96  172.961203  30.580713  83.493998  24302.982064
97          97  172.610553  35.768626  74.609493  70308.085612
98          98  170.051135  35.291044  82.192932  48281.902321
99          99  167.654129  29.285149  79.444433  68567.244680
```

```
    hours_per_week  years_experience  satisfaction  performance  \
0        32.027862         12.778533      8.513977    69.772770
1        37.003125         15.728250      5.155669    85.490092
2        40.026218          5.804297      8.739212    67.956563
3        40.234903         11.688908      9.711276    60.915387
4        37.749673          8.048072      7.826870    59.433708
..             …                 …           …           …
95       42.694550          8.469951      6.380907    85.531529
96       34.813769          9.190375      7.652266    74.604448
97       39.048307          7.063709      4.497773    81.815007
98       35.621909          8.667120      8.848054    75.283184
99       33.086001         11.131901      6.630196    75.297561

    projects_completed        bonus
0             6.876568  4737.346618
1             3.967911  3213.322375
2             5.192242  4057.489646
3             4.075449  6556.903725
4             4.131008  4382.198136
..                 …           …
95            6.297420  3437.799414
96            4.665764  7595.373054
97            5.293427  5281.685723
98            7.413018  2857.642020
99            3.366129  5145.165563

[100 rows x 11 columns]
```

[300]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
# before removal outliers

plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["height"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.distplot(df_height["height"])
plt.title("after removal outliers")

plt.show()
```



0.0.3 Calculate the Z-scores for the 'age' column and identify outliers.

[254]:
```
upper_limit = df["age"].mean() + 3 * df["age"].std()
lower_limit = df["age"].mean() - 3 * df["age"].std()
upper_limit , lower_limit
```

[254]:
```
(56.18563207927538, 14.425197205527795)
```

[255]:
```
df[(df["age"]>=upper_limit)|(df["age"]<=lower_limit)]
```

[255]:

	height	age	weight	income	hours_per_week	years_experience	satisfaction	performance	projects_completed	bonus
100	250.0	80.0	150.0	150000.0	80.0	30.0	10.0	100.0	15.0	20000.0
101	120.0	10.0	30.0	20000.0	10.0	1.0	0.0	50.0	0.0	0.0

0.0.4 Remove the outliers from the 'age' column based on Z-scores and display the cleaned column.

[256]:
```
df_age = df[(df["age"]<=upper_limit)&(df["age"]>=lower_limit)]
df_age
```

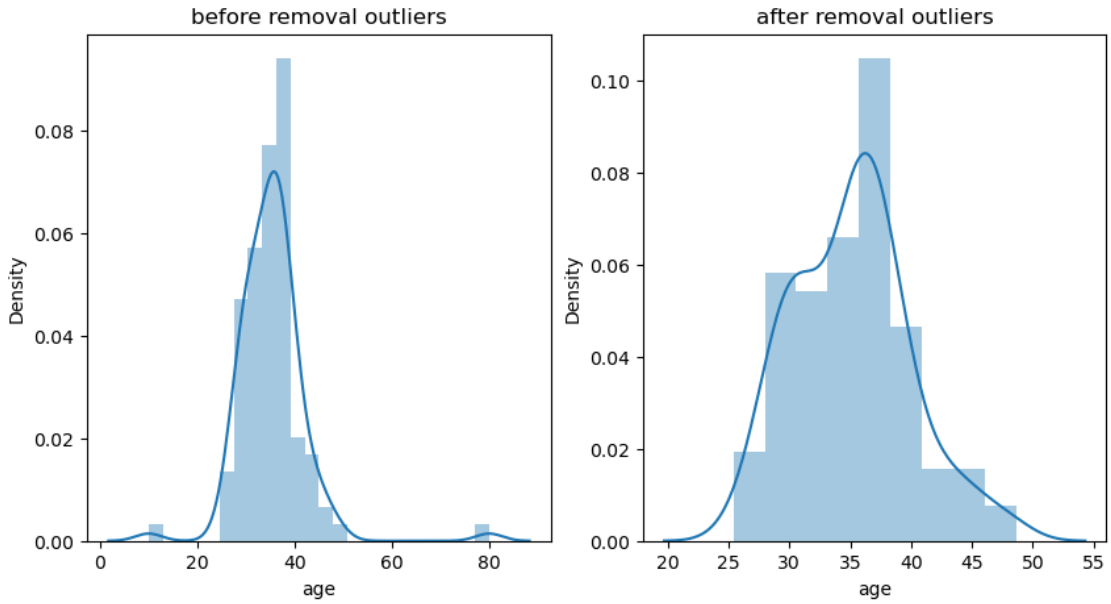
[256]:

	height	age	weight	income	hours_per_week
0	174.967142	27.923146	75.366810	37565.074836	32.027862

```
1   168.617357  32.896773  78.411768  41597.284397       37.003125
2   176.476885  33.286427  86.245769  61209.404077       40.026218
3   185.230299  30.988614  85.807031  59155.553982       40.234903
4   167.658466  34.193571  49.334959  49686.476091       37.749673
..         …          …          …           …                …
95  155.364851  36.926587  59.606356  42962.365218       42.694550
96  172.961203  30.580713  83.493998  24302.982064       34.813769
97  172.610553  35.768626  74.609493  70308.085612       39.048307
98  170.051135  35.291044  82.192932  48281.902321       35.621909
99  167.654129  29.285149  79.444433  68567.244680       33.086001

    years_experience  satisfaction  performance  projects_completed  \
0          12.778533      8.513977    69.772770            6.876568
1          15.728250      5.155669    85.490092            3.967911
2           5.804297      8.739212    67.956563            5.192242
3          11.688908      9.711276    60.915387            4.075449
4           8.048072      7.826870    59.433708            4.131008
..               …             …           …                  …
95          8.469951      6.380907    85.531529            6.297420
96          9.190375      7.652266    74.604448            4.665764
97          7.063709      4.497773    81.815007            5.293427
98          8.667120      8.848054    75.283184            7.413018
99         11.131901      6.630196    75.297561            3.366129

          bonus
0   4737.346618
1   3213.322375
2   4057.489646
3   6556.903725
4   4382.198136
..         …
95  3437.799414
96  7595.373054
97  5281.685723
98  2857.642020
99  5145.165563

[100 rows x 10 columns]
```

[257]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["age"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.distplot(df_age["age"])
```

```
plt.title("after removal outliers")

plt.show()
```



0.0.5 Calculate the Z-scores for the 'weight' column and identify outliers.

[258]:
```
upper_limit = df["weight"].mean() + 3 * df["weight"].std()
lower_limit = df["weight"].mean() - 3 * df["weight"].std()
upper_limit , lower_limit
```

[258]: (126.46840597257743, 16.2246210793965)

[259]:
```
df[(df["weight"]>= upper_limit)|(df["weight"]<=lower_limit)]
```

[259]:

	height	age	weight	income	hours_per_week	years_experience	satisfaction	performance	projects_completed	bonus
9	175.4256	34.62777	127.790972	46967.110213	42.572194	7.50715	9.992089	91.964564	3.210785	8326.509447
100	250.0000	80.00000	150.000000	150000.000000	80.000000	30.00000	10.000000	100.000000	15.000000	20000.000000

0.0.6 Remove the outliers from the 'weight' column based on Z-scores and display the cleaned column.

[260]:
```
df_weight = df[(df["weight"]<= upper_limit)&(df["weight"]>=lower_limit)]
df_weight
```

[260]:

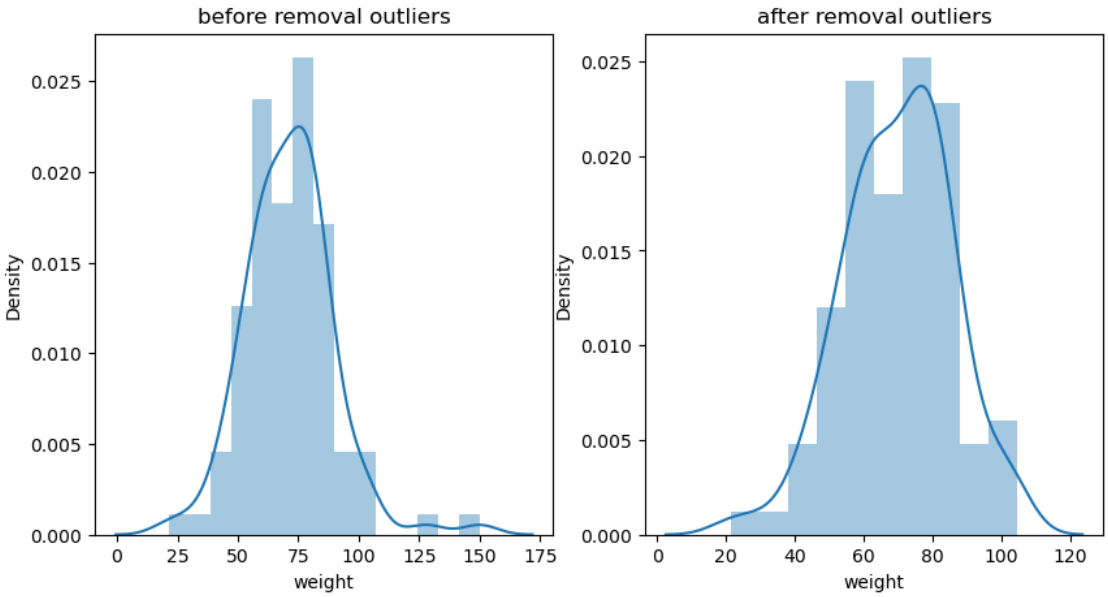
	height	age	weight	income	hours_per_week	\
0	174.967142	27.923146	75.366810	37565.074836	32.027862	
1	168.617357	32.896773	78.411768	41597.284397	37.003125	
2	176.476885	33.286427	86.245769	61209.404077	40.026218	
3	185.230299	30.988614	85.807031	59155.553982	40.234903	
4	167.658466	34.193571	49.334959	49686.476091	37.749673	
..	…	…	…	…	…	
96	172.961203	30.580713	83.493998	24302.982064	34.813769	
97	172.610553	35.768626	74.609493	70308.085612	39.048307	
98	170.051135	35.291044	82.192932	48281.902321	35.621909	
99	167.654129	29.285149	79.444433	68567.244680	33.086001	
101	120.000000	10.000000	30.000000	20000.000000	10.000000	

	years_experience	satisfaction	performance	projects_completed	\
0	12.778533	8.513977	69.772770	6.876568	
1	15.728250	5.155669	85.490092	3.967911	
2	5.804297	8.739212	67.956563	5.192242	
3	11.688908	9.711276	60.915387	4.075449	
4	8.048072	7.826870	59.433708	4.131008	
..	…	…	…	…	
96	9.190375	7.652266	74.604448	4.665764	
97	7.063709	4.497773	81.815007	5.293427	
98	8.667120	8.848054	75.283184	7.413018	
99	11.131901	6.630196	75.297561	3.366129	
101	1.000000	0.000000	50.000000	0.000000	

	bonus
0	4737.346618
1	3213.322375
2	4057.489646
3	6556.903725
4	4382.198136
..	…
96	7595.373054
97	5281.685723
98	2857.642020
99	5145.165563
101	0.000000

[100 rows x 10 columns]

[261]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["weight"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.distplot(df_weight["weight"])
plt.title("after removal outliers")

plt.show()
```



0.0.7 Calculate the Z-scores for the 'income' column and identify outliers.

[262]:
```
upper_limit = df["income"].mean() + 3 * df["income"].std()
lower_limit = df["income"].mean() - 3 * df["income"].std()
upper_limit , lower_limit
```

[262]: (102250.68256025728, 2264.2229504477727)

[263]:
```
df[(df["income"]>= upper_limit)|(df["income"]<=lower_limit)]
```

[263]:

	height	age	weight	income	hours_per_week	years_experience	satisfaction	performance	projects_completed	bonus
100	250.0	80.0	150.0	150000.0	80.0	30.0	10.0	100.0	15.0	20000.0

0.0.8 Remove the outliers from the 'income' column based on Z-scores and display the cleaned column.

[264]:
```
df_income = df[(df["income"]<= upper_limit)&(df["income"]>=lower_limit)]
df_income
```

[264]:

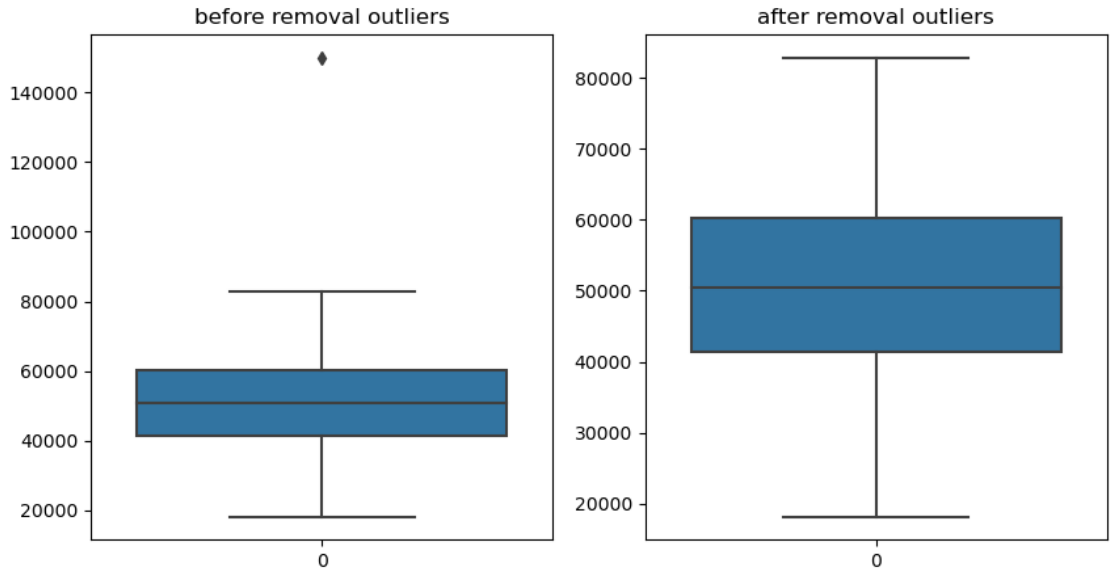
	height	age	weight	income	hours_per_week	\
0	174.967142	27.923146	75.366810	37565.074836	32.027862	
1	168.617357	32.896773	78.411768	41597.284397	37.003125	
2	176.476885	33.286427	86.245769	61209.404077	40.026218	
3	185.230299	30.988614	85.807031	59155.553982	40.234903	
4	167.658466	34.193571	49.334959	49686.476091	37.749673	
..	…	…	…	…	…	
96	172.961203	30.580713	83.493998	24302.982064	34.813769	
97	172.610553	35.768626	74.609493	70308.085612	39.048307	
98	170.051135	35.291044	82.192932	48281.902321	35.621909	
99	167.654129	29.285149	79.444433	68567.244680	33.086001	
101	120.000000	10.000000	30.000000	20000.000000	10.000000	

	years_experience	satisfaction	performance	projects_completed	\
0	12.778533	8.513977	69.772770	6.876568	
1	15.728250	5.155669	85.490092	3.967911	
2	5.804297	8.739212	67.956563	5.192242	
3	11.688908	9.711276	60.915387	4.075449	
4	8.048072	7.826870	59.433708	4.131008	
..	…	…	…	…	
96	9.190375	7.652266	74.604448	4.665764	
97	7.063709	4.497773	81.815007	5.293427	
98	8.667120	8.848054	75.283184	7.413018	
99	11.131901	6.630196	75.297561	3.366129	
101	1.000000	0.000000	50.000000	0.000000	

	bonus
0	4737.346618
1	3213.322375
2	4057.489646
3	6556.903725
4	4382.198136
..	…
96	7595.373054
97	5281.685723
98	2857.642020
99	5145.165563
101	0.000000

[101 rows x 10 columns]

[265]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.boxplot(df["income"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.boxplot(df_income["income"])
plt.title("after removal outliers")

plt.show()
```



0.0.9 Calculate the Z-scores for the 'hours_per_week' column and identify outliers.

[266]:
```
upper_limit = df["hours_per_week"].mean() + 3 * df["hours_per_week"].std()
lower_limit = df["hours_per_week"].mean() - 3 * df["hours_per_week"].std()
upper_limit , lower_limit
```

[266]:
```
(61.56606805165089, 18.080946943345722)
```

[267]:
```
df[(df["hours_per_week"]>=upper_limit)|(df["hours_per_week"]<= lower_limit)]
```

[267]:

	height	age	weight	income	hours_per_week	years_experience	\
100	250.0	80.0	150.0	150000.0	80.0	30.0	
101	120.0	10.0	30.0	20000.0	10.0	1.0	

	satisfaction	performance	projects_completed	bonus

```
100          10.0        100.0                15.0  20000.0
101           0.0         50.0                 0.0      0.0
```

0.0.10 Remove the outliers from the 'hours_per_week' column based on Z-scores and display the cleaned column.

[268]:
```
df_hpr = df[(df["hours_per_week"]<=upper_limit)&(df["hours_per_week"]>=
 ↪lower_limit)]
df_hpr
```

[268]:

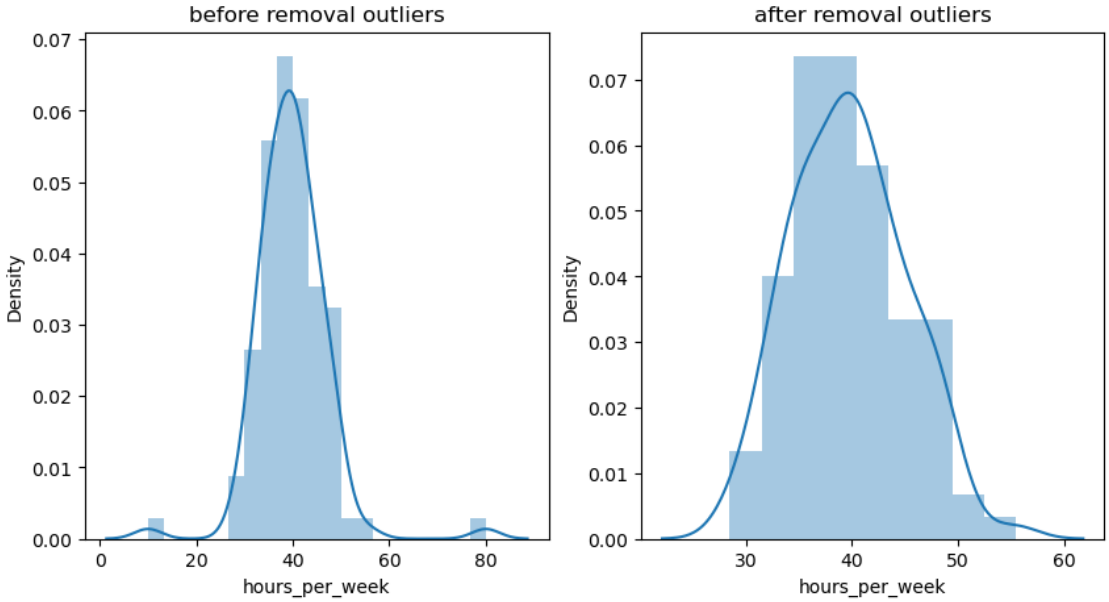
	height	age	weight	income	hours_per_week	\
0	174.967142	27.923146	75.366810	37565.074836	32.027862	
1	168.617357	32.896773	78.411768	41597.284397	37.003125	
2	176.476885	33.286427	86.245769	61209.404077	40.026218	
3	185.230299	30.988614	85.807031	59155.553982	40.234903	
4	167.658466	34.193571	49.334959	49686.476091	37.749673	
..	…	…	…	…	…	
95	155.364851	36.926587	59.606356	42962.365218	42.694550	
96	172.961203	30.580713	83.493998	24302.982064	34.813769	
97	172.610553	35.768626	74.609493	70308.085612	39.048307	
98	170.051135	35.291044	82.192932	48281.902321	35.621909	
99	167.654129	29.285149	79.444433	68567.244680	33.086001	

	years_experience	satisfaction	performance	projects_completed	\
0	12.778533	8.513977	69.772770	6.876568	
1	15.728250	5.155669	85.490092	3.967911	
2	5.804297	8.739212	67.956563	5.192242	
3	11.688908	9.711276	60.915387	4.075449	
4	8.048072	7.826870	59.433708	4.131008	
..	…	…	…	…	
95	8.469951	6.380907	85.531529	6.297420	
96	9.190375	7.652266	74.604448	4.665764	
97	7.063709	4.497773	81.815007	5.293427	
98	8.667120	8.848054	75.283184	7.413018	
99	11.131901	6.630196	75.297561	3.366129	

	bonus
0	4737.346618
1	3213.322375
2	4057.489646
3	6556.903725
4	4382.198136
..	…
95	3437.799414
96	7595.373054
97	5281.685723
98	2857.642020

```
99  5145.165563

[100 rows x 10 columns]
```

[269]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["hours_per_week"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.distplot(df_hpr["hours_per_week"])
plt.title("after removal outliers")

plt.show()
```



0.0.11 Calculate the Z-scores for the 'years_experience' column and identify outliers.

[270]:
```
upper_limit = df["years_experience"].mean() + 3 * df["years_experience"].std()
lower_limit = df["years_experience"].mean() - 3 * df["years_experience"].std()
upper_limit , lower_limit
```

[270]: (20.311954358681092, -0.7745363092471109)

[271]:
```
df[(df["years_experience"]>= upper_limit)|(df["years_experience"]<=lower_limit)]
```

[271]:
```
      height   age  weight    income  hours_per_week  years_experience  \
100   250.0  80.0   150.0  150000.0            80.0              30.0

     satisfaction  performance  projects_completed    bonus
100          10.0        100.0                15.0  20000.0
```

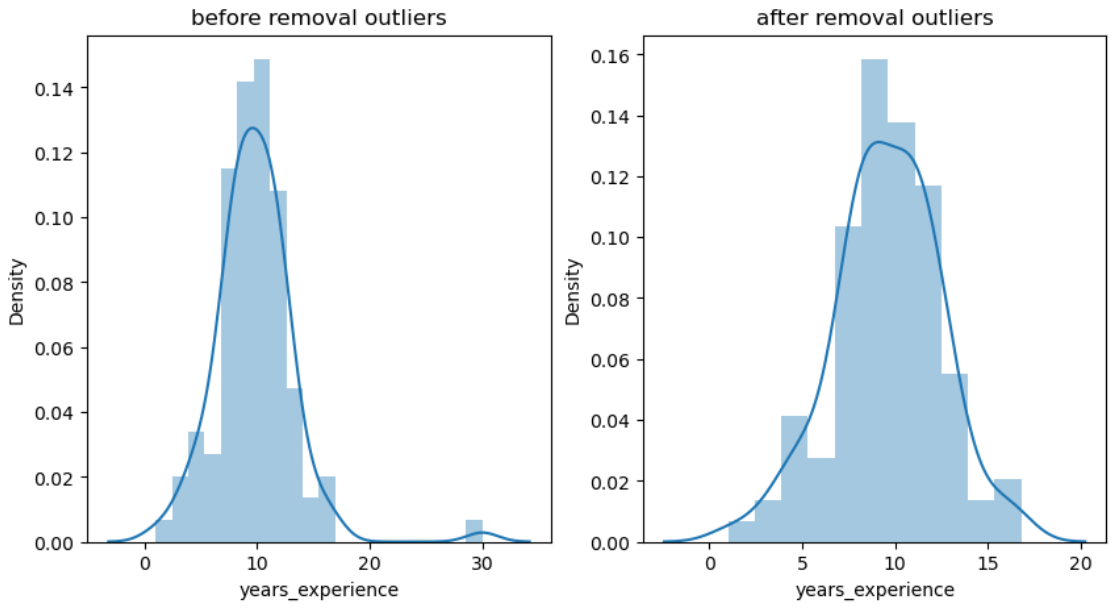
0.0.12 Remove the outliers from the 'years_experience' column based on Z-scores and display the cleaned column.

[272]:
```
df_ye = df[(df["years_experience"]<=␣
 ↪upper_limit)&(df["years_experience"]>=lower_limit)]
df_ye
```

[272]:
```
          height        age     weight        income  hours_per_week  \
0     174.967142  27.923146  75.366810  37565.074836       32.027862
1     168.617357  32.896773  78.411768  41597.284397       37.003125
2     176.476885  33.286427  86.245769  61209.404077       40.026218
3     185.230299  30.988614  85.807031  59155.553982       40.234903
4     167.658466  34.193571  49.334959  49686.476091       37.749673
..           …         …         …            …              …
96    172.961203  30.580713  83.493998  24302.982064       34.813769
97    172.610553  35.768626  74.609493  70308.085612       39.048307
98    170.051135  35.291044  82.192932  48281.902321       35.621909
99    167.654129  29.285149  79.444433  68567.244680       33.086001
101   120.000000  10.000000  30.000000  20000.000000       10.000000

     years_experience  satisfaction  performance  projects_completed  \
0           12.778533      8.513977    69.772770            6.876568
1           15.728250      5.155669    85.490092            3.967911
2            5.804297      8.739212    67.956563            5.192242
3           11.688908      9.711276    60.915387            4.075449
4            8.048072      7.826870    59.433708            4.131008
..                …             …           …                   …
96           9.190375      7.652266    74.604448            4.665764
97           7.063709      4.497773    81.815007            5.293427
98           8.667120      8.848054    75.283184            7.413018
99          11.131901      6.630196    75.297561            3.366129
101          1.000000      0.000000    50.000000            0.000000

           bonus
0     4737.346618
1     3213.322375
2     4057.489646
3     6556.903725
4     4382.198136
..           …
96    7595.373054
```

```
97    5281.685723
98    2857.642020
99    5145.165563
101      0.000000

[101 rows x 10 columns]
```

[273]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["years_experience"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.distplot(df_ye["years_experience"])
plt.title("after removal outliers")

plt.show()
```



0.0.13 Calculate the Z-scores for the 'satisfaction' column and identify outliers.

[274]:
```
upper_limit = df["satisfaction"].mean() + 3 * df["satisfaction"].std()
lower_limit = df["satisfaction"].mean() - 3 * df["satisfaction"].std()
upper_limit , lower_limit
```

[274]: (13.744440812343601, 0.27161355379184116)

[275]:
```
df[(df["satisfaction"]>=upper_limit)|(df["satisfaction"]<=lower_limit)]
```

[275]:
```
     height   age  weight   income  hours_per_week  years_experience  \
101   120.0  10.0    30.0  20000.0            10.0               1.0

     satisfaction  performance  projects_completed  bonus
101           0.0         50.0                 0.0    0.0
```

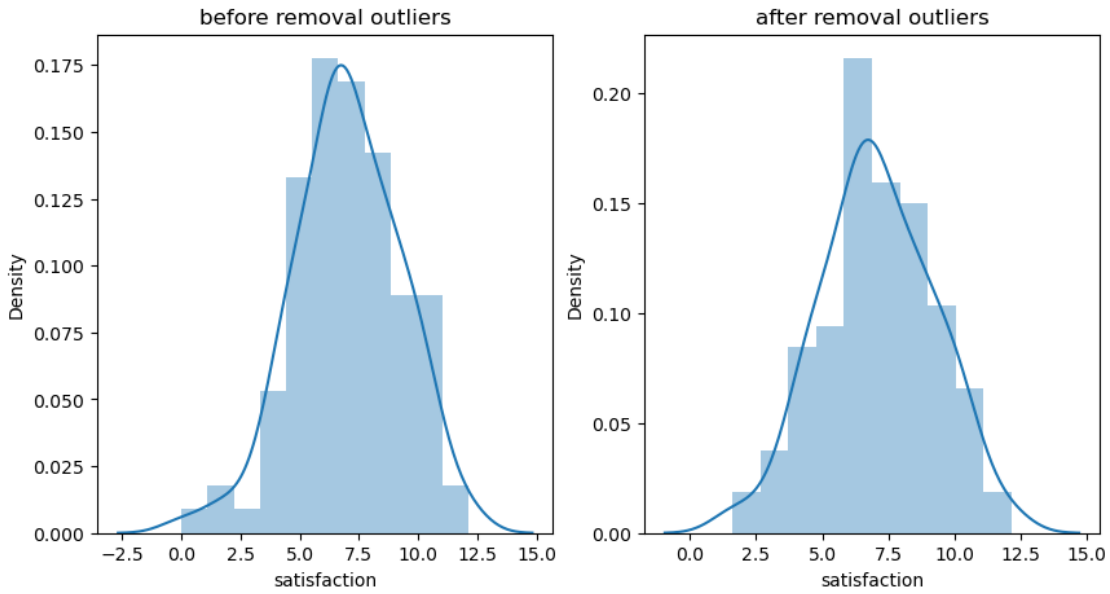
0.0.14 Remove the outliers from the 'satisfaction' column based on Z-scores and display the cleaned column.

[276]:
```
df_sat = df[(df["satisfaction"]<=upper_limit)&(df["satisfaction"]>=lower_limit)]
df_sat
```

[276]:
```
         height        age      weight         income  hours_per_week  \
0    174.967142  27.923146   75.366810   37565.074836       32.027862
1    168.617357  32.896773   78.411768   41597.284397       37.003125
2    176.476885  33.286427   86.245769   61209.404077       40.026218
3    185.230299  30.988614   85.807031   59155.553982       40.234903
4    167.658466  34.193571   49.334959   49686.476091       37.749673
..          …         …          …             …                …
96   172.961203  30.580713   83.493998   24302.982064       34.813769
97   172.610553  35.768626   74.609493   70308.085612       39.048307
98   170.051135  35.291044   82.192932   48281.902321       35.621909
99   167.654129  29.285149   79.444433   68567.244680       33.086001
100  250.000000  80.000000  150.000000  150000.000000       80.000000

     years_experience  satisfaction  performance  projects_completed  \
0           12.778533      8.513977    69.772770            6.876568
1           15.728250      5.155669    85.490092            3.967911
2            5.804297      8.739212    67.956563            5.192242
3           11.688908      9.711276    60.915387            4.075449
4            8.048072      7.826870    59.433708            4.131008
..                …             …           …                   …
96           9.190375      7.652266    74.604448            4.665764
97           7.063709      4.497773    81.815007            5.293427
98           8.667120      8.848054    75.283184            7.413018
99          11.131901      6.630196    75.297561            3.366129
100         30.000000     10.000000   100.000000           15.000000

            bonus
0     4737.346618
1     3213.322375
2     4057.489646
3     6556.903725
4     4382.198136
..            …
```

```
96     7595.373054
97     5281.685723
98     2857.642020
99     5145.165563
100   20000.000000

[101 rows x 10 columns]
```

[277]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["satisfaction"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.distplot(df_sat["satisfaction"])
plt.title("after removal outliers")

plt.show()
```



0.0.15 Calculate the Z-scores for the 'performance' column and identify outliers.

[278]:
```
upper_limit = df["performance"].mean() + 3 * df["performance"].std()
lower_limit = df["performance"].mean() - 3 * df["performance"].std()
upper_limit , lower_limit
```

[278]: (105.75478823672613, 44.12441624352225)

[279]:
```
df[(df["performance"]>= upper_limit)|(df["performance"]<= lower_limit)]
```

[279]:
```
Empty DataFrame
Columns: [height, age, weight, income, hours_per_week, years_experience,
satisfaction, performance, projects_completed, bonus]
Index: []
```

[280]:
```
Q1 = df["performance"].quantile(0.25)
Q3 = df["performance"].quantile(0.75)
iqr = Q3 - Q1
iqr
```

[280]:
```
13.385442919847549
```

[281]:
```
upper_limit = Q3 + 1.5 * iqr
lower_limit = Q1 - 1.5 * iqr
upper_limit , lower_limit
```

[281]:
```
(101.25966947461097, 47.71789779522078)
```

[282]:
```
df[(df["performance"]>= upper_limit)|(df["performance"]<= lower_limit)]
```

[282]:

	height	age	weight	income	hours_per_week	years_experience	satisfaction	performance	projects_completed	bonus
55	179.312801	31.428243	62.736489	34962.05953	36.035636	8.764369	7.118437	101.323821	3.858507	6234.591663

0.0.16 Remove the outliers from the 'performance' column based on IQR and display the cleaned column.

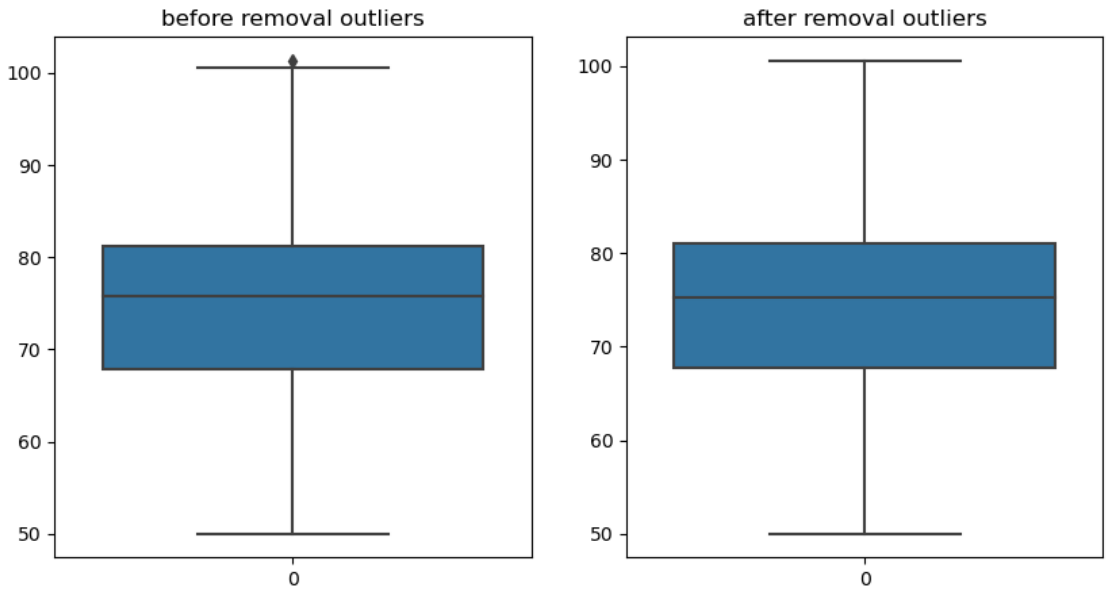
[283]:
```
df_per = df[(df["performance"]<= upper_limit)&(df["performance"]>= lower_limit)]
df_per
```

[283]:

	height	age	weight	income	hours_per_week
0	174.967142	27.923146	75.366810	37565.074836	32.027862
1	168.617357	32.896773	78.411768	41597.284397	37.003125
2	176.476885	33.286427	86.245769	61209.404077	40.026218
3	185.230299	30.988614	85.807031	59155.553982	40.234903
4	167.658466	34.193571	49.334959	49686.476091	37.749673
..	…	…	…	…	…
97	172.610553	35.768626	74.609493	70308.085612	39.048307
98	170.051135	35.291044	82.192932	48281.902321	35.621909
99	167.654129	29.285149	79.444433	68567.244680	33.086001

```
100  250.000000  80.000000  150.000000  150000.000000       80.000000
101  120.000000  10.000000   30.000000   20000.000000       10.000000

     years_experience  satisfaction  performance  projects_completed  \
0           12.778533      8.513977    69.772770            6.876568
1           15.728250      5.155669    85.490092            3.967911
2            5.804297      8.739212    67.956563            5.192242
3           11.688908      9.711276    60.915387            4.075449
4            8.048072      7.826870    59.433708            4.131008
..                ...           ...          ...                 ...
97           7.063709      4.497773    81.815007            5.293427
98           8.667120      8.848054    75.283184            7.413018
99          11.131901      6.630196    75.297561            3.366129
100         30.000000     10.000000   100.000000           15.000000
101          1.000000      0.000000    50.000000            0.000000

             bonus
0      4737.346618
1      3213.322375
2      4057.489646
3      6556.903725
4      4382.198136
..             ...
97     5281.685723
98     2857.642020
99     5145.165563
100   20000.000000
101       0.000000

[101 rows x 10 columns]
```

[284]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.boxplot(df["performance"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.boxplot(df_per["performance"])
plt.title("after removal outliers")

plt.show()
```



0.0.17 Calculate the Z-scores for the 'projects_completed' column and identify outliers.

[285]:
```
upper_limit = df["projects_completed"].mean() + 3 * df["projects_completed"].
 ↪std()
lower_limit = df["projects_completed"].mean() - 3 * df["projects_completed"].
 ↪std()
upper_limit , lower_limit
```

[285]:
```
(11.957950322014844, -0.9632480992744075)
```

[286]:
```
df[(df["projects_completed"]>=␣
 ↪upper_limit)|(df["projects_completed"]<=lower_limit)]
```

[286]:
```
     height   age  weight    income  hours_per_week  years_experience  \
100   250.0  80.0   150.0  150000.0            80.0              30.0

     satisfaction  performance  projects_completed    bonus
100          10.0        100.0                15.0  20000.0
```

0.0.18 Remove the outliers from the 'projects_completed' column based on Z-scores and display the cleaned column.

[287]:
```
df_pro = df[(df["projects_completed"]<=␣
 ↪upper_limit)&(df["projects_completed"]>=lower_limit)]
df_pro
```

[287]:

	height	age	weight	income	hours_per_week	\
0	174.967142	27.923146	75.366810	37565.074836	32.027862	
1	168.617357	32.896773	78.411768	41597.284397	37.003125	
2	176.476885	33.286427	86.245769	61209.404077	40.026218	
3	185.230299	30.988614	85.807031	59155.553982	40.234903	
4	167.658466	34.193571	49.334959	49686.476091	37.749673	
..	…	…	…	…	…	
96	172.961203	30.580713	83.493998	24302.982064	34.813769	
97	172.610553	35.768626	74.609493	70308.085612	39.048307	
98	170.051135	35.291044	82.192932	48281.902321	35.621909	
99	167.654129	29.285149	79.444433	68567.244680	33.086001	
101	120.000000	10.000000	30.000000	20000.000000	10.000000	

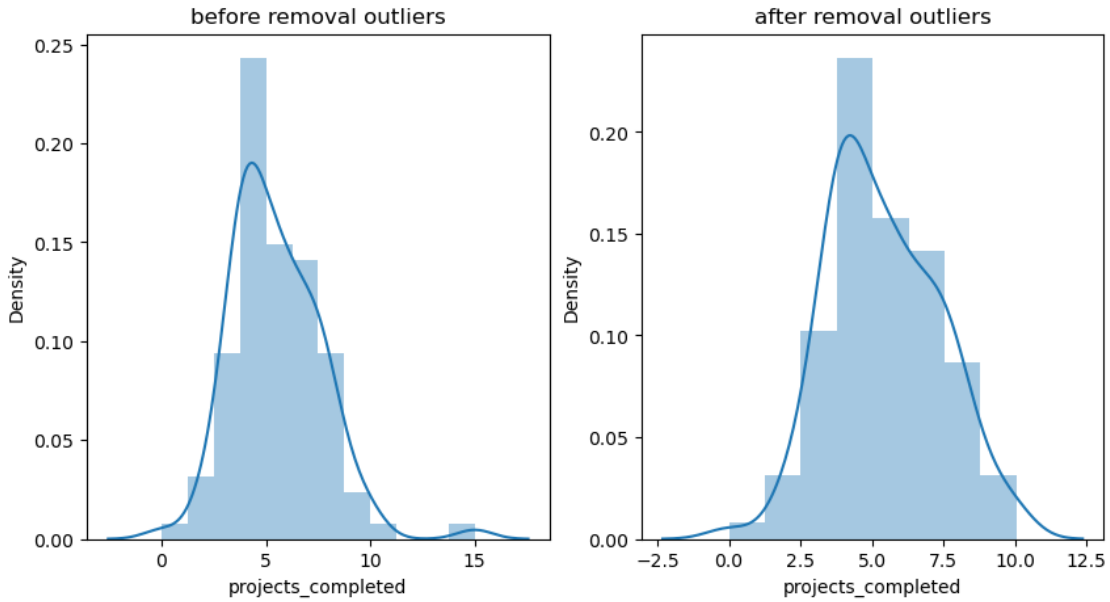
	years_experience	satisfaction	performance	projects_completed	\
0	12.778533	8.513977	69.772770	6.876568	
1	15.728250	5.155669	85.490092	3.967911	
2	5.804297	8.739212	67.956563	5.192242	
3	11.688908	9.711276	60.915387	4.075449	
4	8.048072	7.826870	59.433708	4.131008	
..	…	…	…	…	
96	9.190375	7.652266	74.604448	4.665764	
97	7.063709	4.497773	81.815007	5.293427	
98	8.667120	8.848054	75.283184	7.413018	
99	11.131901	6.630196	75.297561	3.366129	
101	1.000000	0.000000	50.000000	0.000000	

	bonus
0	4737.346618
1	3213.322375
2	4057.489646
3	6556.903725
4	4382.198136
..	…
96	7595.373054
97	5281.685723
98	2857.642020
99	5145.165563
101	0.000000

[101 rows x 10 columns]

[288]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["projects_completed"])
plt.title("before removal outliers")

# after removal ourliers
```

```
plt.subplot(1,2,2)
sns.distplot(df_pro["projects_completed"])
plt.title("after removal outliers")

plt.show()
```



0.0.19 Calculate the Z-scores for the 'bonus' column and identify outliers.

[289]:
```
upper_limit = df["bonus"].mean() + 3 * df["bonus"].std()
lower_limit = df["bonus"].mean()  - 3 * df["bonus"].std()
upper_limit , lower_limit
```

[289]: (11729.281190114018, -3384.7563866360197)

[290]:
```
df[(df["bonus"]>=upper_limit)|(df["bonus"]<= lower_limit)]
```

[290]:

	height	age	weight	income	hours_per_week	years_experience	\
100	250.0	80.0	150.0	150000.0	80.0	30.0	

	satisfaction	performance	projects_completed	bonus
100	10.0	100.0	15.0	20000.0

0.0.20 Remove the outliers from the 'bonus' column based on Z-scores and display the cleaned column.

[291]:
```
df_bonus =df[(df["bonus"]<=upper_limit)&(df["bonus"]>= lower_limit)]
df_bonus
```

[291]:

	height	age	weight	income	hours_per_week	\
0	174.967142	27.923146	75.366810	37565.074836	32.027862	
1	168.617357	32.896773	78.411768	41597.284397	37.003125	
2	176.476885	33.286427	86.245769	61209.404077	40.026218	
3	185.230299	30.988614	85.807031	59155.553982	40.234903	
4	167.658466	34.193571	49.334959	49686.476091	37.749673	
..	…	…	…	…	…	
96	172.961203	30.580713	83.493998	24302.982064	34.813769	
97	172.610553	35.768626	74.609493	70308.085612	39.048307	
98	170.051135	35.291044	82.192932	48281.902321	35.621909	
99	167.654129	29.285149	79.444433	68567.244680	33.086001	
101	120.000000	10.000000	30.000000	20000.000000	10.000000	

	years_experience	satisfaction	performance	projects_completed	\
0	12.778533	8.513977	69.772770	6.876568	
1	15.728250	5.155669	85.490092	3.967911	
2	5.804297	8.739212	67.956563	5.192242	
3	11.688908	9.711276	60.915387	4.075449	
4	8.048072	7.826870	59.433708	4.131008	
..	…	…	…	…	
96	9.190375	7.652266	74.604448	4.665764	
97	7.063709	4.497773	81.815007	5.293427	
98	8.667120	8.848054	75.283184	7.413018	
99	11.131901	6.630196	75.297561	3.366129	
101	1.000000	0.000000	50.000000	0.000000	

	bonus
0	4737.346618
1	3213.322375
2	4057.489646
3	6556.903725
4	4382.198136
..	…
96	7595.373054
97	5281.685723
98	2857.642020
99	5145.165563
101	0.000000

[101 rows x 10 columns]

[292]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df["bonus"])
plt.title("before removal outliers")

# after removal ourliers
plt.subplot(1,2,2)
sns.distplot(df_bonus["bonus"])
plt.title("after removal outliers")

plt.show()
```