In [9]:

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**%**matplotlib inline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| In [10]: | s1 | **=** | pd.Series(range(1,10,1)) | |
|  |  |  |  | |
| In [11]: | s1 |  |  | |
| Out[11]: | 0 |  | 1 | |
|  | 1 |  | 2 | |
|  | 2 |  | 3 | |
|  | 3 |  | 4 | |
|  | 4 |  | 5 | |
|  | 5 |  | 6 | |
|  | 6 |  | 7 | |
|  | 7 |  | 8 | |
|  | 8 |  | 9 | |
| dtype: | | | | int64 |

In [12]:

s3 **=** pd.Series({1:21, 2:13,3:45})

In [13]:

s3

Out[13]:1 21

2 13

3 45

dtype: int64

In [14]:

s2 **=** pd.Series([1, 2, 3, 4], index**=**['p', 'q', 'r','s'], name**=**'one')

In [15]:

s2

Out[15]:p 1

q 2

r 3

s 4

Name: one, dtype: int64

In [16]:

df1 **=** pd.DataFrame(s2)df1

Out[16]:

**one**

**p** 1

**q** 2

**r** 3

**s** 4

In [17]:

df2 **=** pd.read\_csv("E:\JankeepA46 DSBDA\california\_housing\_test.csv"

In [18]:

df2.head(10)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Out[18]: |  | | | | | | |
|  |  | **longitude** | **latitude** | **housing\_median\_age** | **total\_rooms** | **total\_bedrooms** | **populatio** |
|  | **0** | -122.05 | 37.37 | 27.0 | 3885.0 | 661.0 | 1537. |
|  | **1** | -118.30 | 34.26 | 43.0 | 1510.0 | 310.0 | 809. |
|  | **2** | -117.81 | 33.78 | 27.0 | 3589.0 | 507.0 | 1484. |
|  | **3** | -118.36 | 33.82 | 28.0 | 67.0 | 15.0 | 49. |
|  | **4** | -119.67 | 36.33 | 19.0 | 1241.0 | 244.0 | 850. |
|  | **5** | -119.56 | 36.51 | 37.0 | 1018.0 | 213.0 | 663. |
|  | **6** | -121.43 | 38.63 | 43.0 | 1009.0 | 225.0 | 604. |
|  | **7** | -120.65 | 35.48 | 19.0 | 2310.0 | 471.0 | 1341. |
|  | **8** | -122.84 | 38.40 | 15.0 | 3080.0 | 617.0 | 1446. |
|  | **9** | -118.02 | 34.08 | 31.0 | 2402.0 | 632.0 | 2830. |

In [19]:

df2.tail(3)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Out[19]: |  | | | | | | |
|  |  | **longitude** | **latitude** | **housing\_median\_age** | **total\_rooms** | **total\_bedrooms** | **popula** |
|  | **2997** | -119.70 | 36.30 | 10.0 | 956.0 | 201.0 | 6 |
|  | **2998** | -117.12 | 34.10 | 40.0 | 96.0 | 14.0 |  |
|  | **2999** | -119.63 | 34.42 | 42.0 | 1765.0 | 263.0 | 7 |

In [20]:

df2['median\_house\_value\_new']**=**df2['median\_house\_value']**+**111

In [21]:

df2.tail(3)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Out[21]: |  | | | | | | |
|  |  | **longitude** | **latitude** | **housing\_median\_age** | **total\_rooms** | **total\_bedrooms** | **popula** |
|  | **2997** | -119.70 | 36.30 | 10.0 | 956.0 | 201.0 | 6 |
|  | **2998** | -117.12 | 34.10 | 40.0 | 96.0 | 14.0 |  |
|  | **2999** | -119.63 | 34.42 | 42.0 | 1765.0 | 263.0 | 7 |

In [22]:

df2.to\_json('data1.json')

In [23]:

len(df2['total\_rooms'])

Out[23]:3000

In [24]:

df2['total\_rooms'].count()

Out[24]:3000

In [25]:

df2['total\_rooms'].mean()

Out[25]:2599.578666666667

In [26]:

df2['total\_rooms'].sum()

Out[26]:7798736.0

In [27]:

df2['total\_rooms'].median()

Out[27]:2106.0

In [28]:

df2['total\_rooms'].std()

Out[28]:2155.59333162558

In [29]:

df2['total\_rooms'].min()

Out[29]:6.0

In [30]:

df2['total\_rooms'].max()

Out[30]:30450.0

In [31]:

df2['total\_rooms'].describe()

Out[31]:count 3000.000000

mean 2599.578667

std 2155.593332

min 6.000000

25% 1401.000000

50% 2106.000000

75% 3129.000000

max 30450.000000

Name: total\_rooms, dtype: float64

In [32]:

df2['total\_rooms'].cumsum()

Out[32]:0 3885.0

1 5395.0

2 8984.0

3 9051.0

4 10292.0

...

2995 7790662.0

2996 7795919.0

2997 7796875.0

2998 7796971.0

2999 7798736.0

Name: total\_rooms, Length: 3000, dtype: float64

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Out[33]: | longitude | | | | -119.589200 | | |
|  | latitude | | | | 35.635390 | | |
|  | housing\_median\_age | | | | 28.845333 | | |
|  | total\_rooms | | | | 2599.578667 | | |
|  | total\_bedrooms | | | | 529.950667 | | |
|  | population | | | | 1402.798667 | | |
|  | households | | | | 489.912000 | | |
|  | median\_income | | | | 3.807272 | | |
|  | median\_house\_value | | | | 205846.275000 | | |
|  | median\_house\_value\_new | | | | 205957.275000 | | |
|  | dtype: float64 | | | |  | | |
| In [34]: | df2.describe() | | | |  | | |
| Out[34]: | |  | **longitude** | **latitude** | **housing\_median\_age** | **total\_rooms** | **total\_bedrooms** |
|  | | **count** | 3000.000000 | 3000.00000 | 3000.000000 | 3000.000000 | 3000.000000 |
|  | | **mean** | -119.589200 | 35.63539 | 28.845333 | 2599.578667 | 529.950667 |
|  | | **std** | 1.994936 | 2.12967 | 12.555396 | 2155.593332 | 415.654368 |
|  | | **min** | -124.180000 | 32.56000 | 1.000000 | 6.000000 | 2.000000 |
|  | | **25%** | -121.810000 | 33.93000 | 18.000000 | 1401.000000 | 291.000000 |
|  | | **50%** | -118.485000 | 34.27000 | 29.000000 | 2106.000000 | 437.000000 |
|  | | **75%** | -118.020000 | 37.69000 | 37.000000 | 3129.000000 | 636.000000 |
|  | | **max** | -114.490000 | 41.92000 | 52.000000 | 30450.000000 | 5419.000000 |

In [36]:

df **=** pd.read\_csv("E:\JankeepA46 DSBDA\california\_housing\_test.csv")

In [37]:

df.describe()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Out[37]: |  | | | | | |
|  |  | **longitude** | **latitude** | **housing\_median\_age** | **total\_rooms** | **total\_bedrooms** |
|  | **count** | 3000.000000 | 3000.00000 | 3000.000000 | 3000.000000 | 3000.000000 |
|  | **mean** | -119.589200 | 35.63539 | 28.845333 | 2599.578667 | 529.950667 |
|  | **std** | 1.994936 | 2.12967 | 12.555396 | 2155.593332 | 415.654368 |
|  | **min** | -124.180000 | 32.56000 | 1.000000 | 6.000000 | 2.000000 |
|  | **25%** | -121.810000 | 33.93000 | 18.000000 | 1401.000000 | 291.000000 |
|  | **50%** | -118.485000 | 34.27000 | 29.000000 | 2106.000000 | 437.000000 |
|  | **75%** | -118.020000 | 37.69000 | 37.000000 | 3129.000000 | 636.000000 |
|  | **max** | -114.490000 | 41.92000 | 52.000000 | 30450.000000 | 5419.000000 |

Out[38]:Index(['longitude','latitude','housing\_median\_age','total\_rooms',

e',

'total\_bedrooms', 'population', 'households', 'median\_incom

'median\_house\_value'],dtype='object')

In [39]:

df['longitude']

Out[39]:0 -122.05

1 -118.30

2 -117.81

3 -118.36

4 -119.67

...

2995 -119.86

2996 -118.14

2997 -119.70

2998 -117.12

2999 -119.63

Name: longitude, Length: 3000, dtype: float64

In [40]:

df.longitude

Out[40]:0 -122.05

1 -118.30

2 -117.81

3 -118.36

4 -119.67

...

2995 -119.86

2996 -118.14

2997 -119.70

2998 -117.12

2999 -119.63

Name: longitude, Length: 3000, dtype: float64

|  |  |  |
| --- | --- | --- |
|  | **latitude** | **housing\_median\_age** |
| **0** | 37.37 | 27.0 |
| **1** | 34.26 | 43.0 |
| **2** | 33.78 | 27.0 |
| **3** | 33.82 | 28.0 |
| **4** | 36.33 | 19.0 |
| **...** | ... | ... |
| **2995** | 34.42 | 23.0 |
| **2996** | 34.06 | 27.0 |
| **2997** | 36.30 | 10.0 |
| **2998** | 34.10 | 40.0 |
| **2999** | 34.42 | 42.0 |

3000 rows × 2 columns

In [43]:

data **=** pd.read\_csv("E:\JankeepA46 DSBDA\employees.csv")

In [44]:

data.head(10)

Out[44]:

**First**

|  |  |  |
| --- | --- | --- |
| **Name** |  | |
| **0**Douglas | Male | 8/6/1993 |
| **1**Thomas | Male | 3/31/1996 |
| **2** Maria | Female | 4/23/1993 |
| **3** Jerry | Male | 3/4/2005 |
| **4** Larry | Male | 1/24/1998 |
| **5** Dennis | Male | 4/18/1987 |
| **6** Ruby | Female | 8/17/1987 |
| **7** NaN | Female | 7/20/2015 |
| **8** Angela | Female | 11/22/2005 |
| **9**Frances | Female | 8/8/2002 |

**Gender StartDate**

**LastLoginTime**

**Salary**

**Bonus**

**%**

**Senior TeamManagement**

12:42

PM

6:53

AM

11:17

AM

1:00

PM

4:47

PM

1:35

AM

4:20

PM

10:43

AM

6:29

AM

6:51

AM

97308.0 6.945 True Marketing

61933.0 4.170 True NaN

130590.0 11.858 False Finance

138705.0 9.340 True Finance

101004.0 1.389 True Client

|  |  |  |  |
| --- | --- | --- | --- |
| 115163.0 | 10.125 | False | Services  Legal |
| 65476.0 | 10.012 | True | Product |
| 45906.0 | 11.598 | NaN | Finance |
| 95570.0 | 18.523 | True | Engineering |
| 139852.0 | 7.524 | True | BusinessDevelopment |

|  |  |  |
| --- | --- | --- |
| In [45]: | data.describe() |  |
| Out[45]: | **Salary** | **Bonus %** |
|  | **count** 969.000000 | 1000.000000 |
|  | **mean** 90579.972136 | 10.207555 |
|  | **std** 32916.214577 | 5.528481 |
|  | **min** 35013.000000 | 1.015000 |
|  | **25%** 62666.000000 | 5.401750 |
|  | **50%** 90370.000000 | 9.838500 |
|  | **75%**118733.000000 | 14.838000 |
|  | **max**149908.000000 | 19.944000 |
| In [46]: | data.isnull() |  |
| Out[46]: |  |  |

**FirstGenderName**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | **Time** |  | | | |
| **0** | False | False | False | False | False | False | False | False |
| **1** | False | False | False | False | False | False | False | True |
| **2** | False | False | False | False | False | False | False | False |
| **3** | False | False | False | False | False | False | False | False |
| **4** | False | False | False | False | False | False | False | False |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **995** | False | True | False | False | False | False | False | False |
| **996** | False | False | False | False | False | False | False | False |
| **997** | False | False | False | False | False | False | False | False |
| **998** | False | False | False | False | False | False | False | False |
| **999** | False | False | False | False | False | False | False | False |

**Start Date**

**Last Login**

**Salary**

**Bonus**

**%**

**SeniorTeamManagement**

1000 rows × 8 columns

In [47]:

data.notnull()

Out[47]:

**FirstGenderName**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | **Time** |  | | | |
| **0** | True | True | True | True | True | True | True | True |
| **1** | True | True | True | True | True | True | True | False |
| **2** | True | True | True | True | True | True | True | True |
| **3** | True | True | True | True | True | True | True | True |
| **4** | True | True | True | True | True | True | True | True |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **995** | True | False | True | True | True | True | True | True |
| **996** | True | True | True | True | True | True | True | True |
| **997** | True | True | True | True | True | True | True | True |
| **998** | True | True | True | True | True | True | True | True |
| **999** | True | True | True | True | True | True | True | True |

**StartDate**

**Last Login**

**Salary**

**Bonus**

**%**

**SeniorTeamManagement**

1000 rows × 8 columns

In [48]:

**import** numpy **as** np

data.replace(to\_replace **=** np.nan, value **= -**99)

Out[48]:

**First**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** |  | |
| **0** | Douglas | Male | 8/6/1993 |
| **1** | Thomas | Male | 3/31/1996 |
| **2** | Maria | Female | 4/23/1993 |
| **3** | Jerry | Male | 3/4/2005 |
| **4** | Larry | Male | 1/24/1998 |
| **...** | ... | ... | ... |
| **995** | Henry | -99 | 11/23/2014 |
| **996** | Phillip | Male | 1/31/1984 |
| **997** | Russell | Male | 5/20/2013 |
| **998** | Larry | Male | 4/20/2013 |
| **999** | Albert | Male | 5/15/2012 |

**Gender StartDate**

**LastLoginTime**

**Salary**

**Bonus**

**%**

**Senior TeaManagement**

12:42

PM

6:53

AM

11:17

AM

1:00

PM

4:47

PM

... 6:09

AM

6:30

AM

12:39

PM

4:45

PM

6:24

PM

97308.0 6.945 True Marketi

61933.0 4.170 True -

130590.0 11.858 False Finan

138705.0 9.340 True Finan

101004.0 1.389 True Clie

|  |  |  |  |
| --- | --- | --- | --- |
| ... | ... | ... | Servic |
| 132483.0 | 16.655 | False | Distributi |
| 42392.0 | 19.675 | False | Finan |
| 96914.0 | 1.421 | False | Produ |
| 60500.0 | 11.985 | False | BusineDevelopme |
| 129949.0 | 10.169 | True | Sal |

1000 rows × 8 columns

In [49]:

data.fillna(method **=**'pad')

Out[49]:

**First**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** |  | |
| **0** | Douglas | Male | 8/6/1993 |
| **1** | Thomas | Male | 3/31/1996 |
| **2** | Maria | Female | 4/23/1993 |
| **3** | Jerry | Male | 3/4/2005 |
| **4** | Larry | Male | 1/24/1998 |
| **...** | ... | ... | ... |
| **995** | Henry | Male | 11/23/2014 |
| **996** | Phillip | Male | 1/31/1984 |
| **997** | Russell | Male | 5/20/2013 |
| **998** | Larry | Male | 4/20/2013 |
| **999** | Albert | Male | 5/15/2012 |

**Gender StartDate**

**LastLoginTime**

**Salary**

**Bonus**

**%**

**Senior TeaManagement**

12:42

PM

6:53

AM

11:17

AM

1:00

PM

4:47

PM

... 6:09

AM

6:30

AM

12:39

PM

4:45

PM

6:24

PM

97308.0 6.945 True Marketi

61933.0 4.170 True Marketi

130590.0 11.858 False Finan

138705.0 9.340 True Finan

101004.0 1.389 True Clie

|  |  |  |  |
| --- | --- | --- | --- |
| ... | ... | ... | Servic |
| 132483.0 | 16.655 | False | Distributi |
| 42392.0 | 19.675 | False | Finan |
| 96914.0 | 1.421 | False | Produ |
| 60500.0 | 11.985 | False | BusineDevelopme |
| 129949.0 | 10.169 | True | Sal |

1000 rows × 8 columns

In [50]:

data['Salary'].fillna(int(data['Salary'].mean()), inplace**=True**)

In [51]:

data.dropna(axis**=**1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Out[51]: |  | | | | |
|  |  | **Start Date** | **LastLoginTime** | **Salary** | **Bonus %** |
|  | **0** | 8/6/1993 | 12:42 PM | 97308.0 | 6.945 |
|  | **1** | 3/31/1996 | 6:53AM | 61933.0 | 4.170 |
|  | **2** | 4/23/1993 | 11:17AM | 130590.0 | 11.858 |
|  | **3** | 3/4/2005 | 1:00 PM | 138705.0 | 9.340 |
|  | **4** | 1/24/1998 | 4:47 PM | 101004.0 | 1.389 |
|  | **...** | ... | ... | ... | ... |
|  | **995** | 11/23/2014 | 6:09AM | 132483.0 | 16.655 |
|  | **996** | 1/31/1984 | 6:30AM | 42392.0 | 19.675 |
|  | **997** | 5/20/2013 | 12:39 PM | 96914.0 | 1.421 |
|  | **998** | 4/20/2013 | 4:45 PM | 60500.0 | 11.985 |
|  | **999** | 5/15/2012 | 6:24 PM | 129949.0 | 10.169 |

1000 rows × 4 columns

In [52]:

**import** pandas **as** pd

df **=** pd.DataFrame({"A":[12, 4, 5, **None**, 1],

"B":[**None**, 2, 54, 3, **None**],

"C":[20, 16, **None**, 3, 8],

"D":[14, 3, **None**, **None**, 6]})

df

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Out[52]: |  | | | | |
|  |  | **A** | **B** | **C** | **D** |
|  | **0** | 12.0 | NaN | 20.0 | 14.0 |
|  | **1** | 4.0 | 2.0 | 16.0 | 3.0 |
|  | **2** | 5.0 | 54.0 | NaN | NaN |
|  | **3** | NaN | 3.0 | 3.0 | NaN |
|  | **4** | 1.0 | NaN | 8.0 | 6.0 |

In [53]:

df.interpolate(method **=**'linear', limit\_direction **=**'forward')

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Out[53]: |  | | | | |
|  |  | **A** | **B** | **C** | **D** |
|  | **0** | 12.0 | NaN | 20.0 | 14.0 |
|  | **1** | 4.0 | 2.0 | 16.0 | 3.0 |
|  | **2** | 5.0 | 54.0 | 9.5 | 4.0 |
|  | **3** | 3.0 | 3.0 | 3.0 | 5.0 |
|  | **4** | 1.0 | 3.0 | 8.0 | 6.0 |

In [55]:

text**=**"today is Monday"text.replace(' ','')

Out[55]:'todayisMonday'

In [56]:

text**=**' Today'text.lstrip()

Out[56]:'Today'

In [57]:

text**=**'Today 'text.rstrip()

Out[57]:'Today'

In [58]:

text**=**' Today 'text.strip()

Out[58]:'Today'

In [60]:

**import** pandas**import** scipy**import** numpy

**from** sklearn.preprocessing **import** MinMaxScaler

X **=** [ [110, 200], [120, 800], [310, 400], [140, 900], [510, 200], [

] ]

scaler **=** MinMaxScaler(feature\_range**=**(0,5))rescaledX **=** scaler.fit\_transform(X)

X

|  |  |  |
| --- | --- | --- |
| Out[60]: | [[110, | 200], |
|  | [120, | 800], |
|  | [310, | 400], |
|  | [140, | 900], |
|  | [510, | 200], |
|  | [653, | 400], |
|  | [310, | 880]] |

In [61]:

rescaledX

Out[61]:array([[0. , 0. ],

|  |  |
| --- | --- |
| [0.09208103, | 4.28571429], |
| [1.84162063, | 1.42857143], |
| [0.27624309, | 5. ], |
| [3.68324125, | 0. ], |
| [5. , | 1.42857143], |
| [1.84162063, | 4.85714286]]) |

In [62]:

**from** sklearn.preprocessing **import** StandardScaler

**import** pandas

**import** numpy

X **=** [ [110, 200], [120, 800], [310, 400], [140, 900], [510, 200], [

] ]

scaler **=** StandardScaler().fit(X)rescaledX **=** scaler.transform(X)rescaledX

|  |  |  |
| --- | --- | --- |
| Out[62]: | array([[-1.02004521, | -1.17792918], |
|  | [-0.96841602, | 0.90076937], |
|  | [ 0.01253852, | -0.48502966], |
|  | [-0.86515765, | 1.24721913], |
|  | [ 1.04512224, | -1.17792918], |
|  | [ 1.78341961, | -0.48502966], |
|  | [ 0.01253852, | 1.17792918]]) |

In [63]:

**from** sklearn.preprocessing **import** Normalizer

**import** pandas

**import** numpy

X **=** [ [110, 200], [120, 800], [310, 400], [140, 900], [510, 200], [

] ]

scaler **=** Normalizer().fit(X)

normalizedX **=** scaler.transform(X)normalizedX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Out[63]: | array([[0.48191875, | | 0.87621591], | | | | |
|  | [0.14834045, | | 0.98893635], | | | | |
|  | [0.61257167, | | 0.79041505], | | | | |
|  | [0.15370701, | | 0.98811647], | | | | |
|  | [0.9309732 , | | 0.36508753], | | | | |
|  | [0.8527326 , | | 0.52234769], | | | | |
|  | [0.33225942, | | 0.94318804]]) | | | | |
| In [64]: | **from** sklearn.preprocessing **import** Binarizer  **import** pandas  **import** numpy  X **=** [ [501, 200], [120, 800], [310, 400], [140, | | | 900], | [510, | 200], | [ |
|  | ] ]  binarizer **=** Binarizer(threshold**=**500).fit(X)binaryX **=** binarizer.transform(X)  binaryX | | |  |  |  |  |
| Out[64]: | array([[1, | 0], | | | | | |
|  | [0, | 1], | | | | | |
|  | [0, | 0], | | | | | |
|  | [0, | 1], | | | | | |
|  | [1, | 0], | | | | | |
|  | [1, | 0], | | | | | |
|  | [0, | 1]]) | | | | | |
| In [ ]: |  |  | | | | | |
|  |  |  | | | | | |
| In [ ]: |  |  | | | | | |