Python Programming (Basic-Intermediate)

Module 5 - Data Operation

Part 1 - Pandas

```
import pandas as pd
import numpy as np
```

```
import requests
resp = requests.get('https://api.coingecko.com/api/v3/coins/markets/?vs_
crypto_data = resp.json()
print(crypto_data)
```

```
[{'id': 'bitcoin', 'symbol': 'btc', 'name': 'Bitcoin', 'image': 'https
```

```
coin_id = pd.Series([coin['id'] for coin in crypto_data])
coin_id
```

```
0            bitcoin
1           ethereum
2             tether
3        binancecoin
4             solana
           ...
95     oasis-network
96              usdd
97    flare-networks
98              bonk
99        klay-token
Length: 100, dtype: object
```

```
coin_id.str.match('usd')
```

```
0     False
1     False
2     False
3     False
```

```
4     False
      ...
95    False
96     True
97    False
98    False
99    False
Length: 100, dtype: bool
```

```
c1  = coin_id.sample(10)
```

```
c1
```

```
52            the-graph
0               bitcoin
90                  neo
55      rocket-pool-eth
20              litecoin
5                 ripple
41     crypto-com-chain
11                 tron
17    internet-computer
51         render-token
dtype: object
```

```
c1[5:6]
```

```
5    ripple
dtype: object
```

1. Series - create series from numpy ndarray

```
s = pd.Series(np.random.randn(5), index = ['a','b','c','d','e'])
print(s)
```

```
a   -0.539572
b   -1.560616
c   -0.692304
d   -0.645033
e   -0.236308
dtype: float64
```

2. access series values

```
s[0] # use numerical index
```

```
-0.539571855277893
```

```
s['a'] # use named index
```

```
-0.539571855277893
```

```
s[['b','c','b']]
```

```
b   -1.560616
c   -0.692304
b   -1.560616
dtype: float64
```

```
s[:2]
```

```
a   -0.539572
b   -1.560616
dtype: float64
```

```
s.index
```

```
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

```
s.values
```

```
array([-0.53957186, -1.56061601, -0.69230411, -0.64503271, -0.23630757
```

3. Series operations

```
s
```

```
a   -0.539572
b   -1.560616
c   -0.692304
d   -0.645033
e   -0.236308
dtype: float64
```

```
s.sort_values()
```

```
b   -1.560616
c   -0.692304
d   -0.645033
a   -0.539572
e   -0.236308
dtype: float64
```

```
s.sort_index(ascending=False)
```

```
e   -0.236308
d   -0.645033
c   -0.692304
b   -1.560616
a   -0.539572
dtype: float64
```

```
s.astype('str')
```

```
a      -0.539571855277893
b     -1.5606160126067818
c      -0.692304111430397
d     -0.6450327148537638
e    -0.23630757263039115
dtype: object
```

```
s.loc['b']
```

```
-1.5606160126067818
```

```
s['b']
```

```
-1.5606160126067818
```

```
s.iloc[0]
```

```
-0.539571855277893
```

```
s[0]
```

```
-0.539571855277893
```

```
c1
```

```
52            the-graph
0               bitcoin
90                  neo
55      rocket-pool-eth
20              litecoin
5                 ripple
41     crypto-com-chain
11                 tron
17    internet-computer
51         render-token
dtype: object
```

```
c1.loc[5]
```

```
'ripple'
```

```
c1.iloc[5]
```

```
'ripple'
```

```
s_max = max(abs(s))
s_max
```

```
1.5606160126067818
```

```
s.abs().max()
```

```
1.5606160126067818
```

```
s
```

```
a   -0.539572
b   -1.560616
c   -0.692304
d   -0.645033
e   -0.236308
dtype: float64
```

```
s.apply(lambda x: x/s_max)
```

```
a   -0.345743
b   -1.000000
c   -0.443610
d   -0.413319
e   -0.151419
dtype: float64
```

```
s/s_max
```

```
a   -0.345743
b   -1.000000
c   -0.443610
d   -0.413319
e   -0.151419
dtype: float64
```

```
s.clip(lower=-.1, upper = .5)
```

```
a   -0.1
b   -0.1
c   -0.1
d   -0.1
e   -0.1
dtype: float64
```

```
s.append(pd.Series([-1,1], index=['f','g']))
```

```
a   -0.539572
b   -1.560616
c   -0.692304
d   -0.645033
e   -0.236308
f   -1.000000
g    1.000000
dtype: float64
```

```
<ipython-input-32-f2bfcf76504e>:1: FutureWarning: The series.append me
  s.append(pd.Series([-1,1], index=['f','g']))
```

4. Time series related

We will revisit this topic again once we finished the introduction of data frame.

5. DataFrame

```
d = {'one': [1,2,3,4],
     'two': [4,3,2,1]}
df = pd.DataFrame(d)
print(df)
```

```
   one  two
0    1    4
1    2    3
2    3    2
3    4    1
```

```
df['one']
```

```
0    1
1    2
2    3
3    4
Name: one, dtype: int64
```

```
df[0:1]
```

```
   one  two
0    1    4
```

	one	two
0	1	4

```
df['two'].shift(1)
```

```
0    NaN
1    4.0
2    3.0
3    2.0
Name: two, dtype: float64
```

```
df.values
```

```
array([[1, 4],
       [2, 3],
       [3, 2],
       [4, 1]])
```

```
df.columns
```

```
Index(['one', 'two'], dtype='object')
```

```
df.index
```

```
RangeIndex(start=0, stop=4, step=1)
```

6. Create DataFrame from numpy array

```
d = np.array([['','Col1','Col2'],
                  ['Row1',1,2],
                  ['Row2',3,4]])
df = pd.DataFrame(data=d[1:,1:],
                  index=['Row1','Row2'],
                  columns=['Col1','Col2'])
print(df)
```

```
     Col1 Col2
Row1    1    2
Row2    3    4
```

```
df
```

```
     Col1 Col2
Row1    1    2
Row2    3    4
```

	Col1	Col2
Row1	1	2

7. Create DataFrame from files

```
pd.read_
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_excel('/content/drive/MyDrive/AIS_DG/Superstore.xlsx',
                   sheet_name='Order',
                   index_col='Row ID')
df
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
...                ...        ...        ...             ...
9990    CA-2011-110422 2012-01-22 2012-01-24    Second Class    TB-21
9991    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9992    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9993    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9994    CA-2014-119914 2015-05-05 2015-05-10    Second Class    CC-12

           Customer Name    Segment        Country             City
Row ID
1            Claire Gute   Consumer  United States        Henderson
2            Claire Gute   Consumer  United States        Henderson
3         Darrin Van Huff  Corporate  United States      Los Angeles
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
...	...	...	...	...	...	...	...	...	...	...
9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	Florida
9991	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9992	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9993	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9994	CA-2014	2015-	2015-	Second	CC-	Chris Cortes	Corporate	United	Westminster	California

```
pd.set_option('display.max_columns', 500)
```

```
df
```

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20
...	...	...	...	...	
9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	TB-21
9991	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9992	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9993	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9994	CA-2014-119914	2015-05-05	2015-05-10	Second Class	CC-12

Row ID	Customer Name	Segment	Country	City
1	Claire Gute	Consumer	United States	Henderson
2	Claire Gute	Consumer	United States	Henderson
3	Darrin Van Huff	Corporate	United States	Los Angeles

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida

8. DataFrame dimension

```
df.shape
```

```
(9994, 20)
```

```
df.memory_usage()
```

```
Index            79952
Order ID         79952
Order Date       79952
Ship Date        79952
Ship Mode        79952
Customer ID      79952
Customer Name    79952
Segment          79952
Country          79952
City             79952
State            79952
Postal Code      79952
Region           79952
Product ID       79952
Category         79952
```

```
Sub-Category     79952
Product Name     79952
Sales            79952
Quantity         79952
Discount         79952
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9994 entries, 1 to 9994
Data columns (total 20 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order ID       9994 non-null   object
 1   Order Date     9994 non-null   datetime64[ns]
 2   Ship Date      9994 non-null   datetime64[ns]
 3   Ship Mode      9994 non-null   object
 4   Customer ID    9994 non-null   object
 5   Customer Name  9994 non-null   object
 6   Segment        9994 non-null   object
 7   Country        9994 non-null   object
 8   City           9994 non-null   object
 9   State          9994 non-null   object
 10  Postal Code    9994 non-null   int64
 11  Region         9994 non-null   object
 12  Product ID     9994 non-null   object
 13  Category       9994 non-null   object
 14  Sub-Category   9994 non-null   object
```

```
df.columns
```

```
Index(['Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer I
       'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Na
       'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

9. DataFrame Viewing: head()

```
df.head()
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
```

```
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20

          Customer Name    Segment        Country             City  \
Row ID
1           Claire Gute   Consumer  United States        Henderson
2           Claire Gute   Consumer  United States        Henderson
3       Darrin Van Huff  Corporate  United States      Los Angeles
4        Sean O'Donnell   Consumer  United States  Fort Lauderdale
5        Sean O'Donnell   Consumer  United States  Fort Lauderdale

             State  Postal Code Region       Product ID         Categ
Row ID
1         Kentucky        42420  South  FUR-BO-10001798         Furnit
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Posta Code
Row ID											
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	4242
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	4242
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	9003
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	3331
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	3331

```
df[:5]
```

```
             Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
```

```
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20

          Customer Name    Segment        Country             City  \
Row ID
1           Claire Gute   Consumer  United States        Henderson
2           Claire Gute   Consumer  United States        Henderson
3       Darrin Van Huff  Corporate  United States      Los Angeles
4        Sean O'Donnell   Consumer  United States  Fort Lauderdale
5        Sean O'Donnell   Consumer  United States  Fort Lauderdale

             State  Postal Code Region       Product ID         Categ
Row ID
1         Kentucky        42420  South  FUR-BO-10001798        Furnit
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Posta Code
Row ID											
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	4242
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	4242
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	9003
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	3331
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	3331

10. DataFrame Viewing: tail()

```
df.tail()
```

```
             Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
9990    CA-2011-110422 2012-01-22 2012-01-24    Second Class    TB-21
9991    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9992    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9993    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9994    CA-2014-119914 2015-05-05 2015-05-10    Second Class    CC-12

           Customer Name    Segment        Country         City
Row ID
9990    Tom Boeckenhauer  Consumer  United States        Miami     Fl
9991         Dave Brooks  Consumer  United States   Costa Mesa  Calif
9992         Dave Brooks  Consumer  United States   Costa Mesa  Calif
9993         Dave Brooks  Consumer  United States   Costa Mesa  Calif
9994        Chris Cortes  Consumer  United States  Westminster  Calif

        Postal Code Region       Product ID         Category Sub-Cate
Row ID
9990           33180  South  FUR-FU-10001889        Furniture  Furnish
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	Florida
9991	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9992	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9993	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9994	CA-2014-119914	2015-05-05	2015-05-10	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	California

11. Viewing DataFrame: columns

```
df.columns
```

```
Index(['Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer I
       'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Na
       'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

```
df.index
```

```
Int64Index([   1,    2,    3,    4,    5,    6,    7,    8,    9,   10
            ...
            9985, 9986, 9987, 9988, 9989, 9990, 9991, 9992, 9993, 9994
           dtype='int64', name='Row ID', length=9994)
```

```
df.values
```

```
array([['CA-2013-152156', Timestamp('2014-11-09 00:00:00'),
        Timestamp('2014-11-12 00:00:00'), ..., 2, 0.0, 41.9136],
       ['CA-2013-152156', Timestamp('2014-11-09 00:00:00'),
        Timestamp('2014-11-12 00:00:00'), ..., 3, 0.0,
        219.58199999999997],
       ['CA-2013-138688', Timestamp('2014-06-13 00:00:00'),
        Timestamp('2014-06-17 00:00:00'), ..., 2, 0.0,
        6.8713999999999995],
       ...,
       ['CA-2014-121258', Timestamp('2015-02-27 00:00:00'),
        Timestamp('2015-03-04 00:00:00'), ..., 2, 0.2,
        19.393200000000007],
       ['CA-2014-121258', Timestamp('2015-02-27 00:00:00'),
        Timestamp('2015-03-04 00:00:00'), ..., 4, 0.0, 13.32],
       ['CA-2014-119914', Timestamp('2015-05-05 00:00:00'),
        Timestamp('2015-05-10 00:00:00'), ..., 2, 0.0, 72.947999999999
      dtype=object)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9994 entries, 1 to 9994
Data columns (total 20 columns):
```

```
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order ID       9994 non-null   object
 1   Order Date     9994 non-null   datetime64[ns]
 2   Ship Date      9994 non-null   datetime64[ns]
 3   Ship Mode      9994 non-null   object
 4   Customer ID    9994 non-null   object
 5   Customer Name  9994 non-null   object
 6   Segment        9994 non-null   object
 7   Country        9994 non-null   object
 8   City           9994 non-null   object
 9   State          9994 non-null   object
 10  Postal Code    9994 non-null   int64
 11  Region         9994 non-null   object
 12  Product ID     9994 non-null   object
 13  Category       9994 non-null   object
 14  Sub-Category   9994 non-null   object
```

12. Viewing DataFrame: describe

```
df.describe(percentiles=[0.9,0.99])
```

```
        Postal Code         Sales     Quantity     Discount       Prof
count   9994.000000   9994.000000  9994.000000  9994.000000  9994.0000
mean   55190.379428    229.858001     3.789574     0.156203    28.6568
std    32063.693350    623.245101     2.225110     0.206452   234.2601
min     1040.000000      0.444000     1.000000     0.000000 -6599.9780
50%    56430.500000     54.490000     3.000000     0.200000     8.6665
90%    94122.000000    572.706000     7.000000     0.400000    89.2816
99%    98115.000000   2481.694600    11.000000     0.800000   580.6578
max    99301.000000  22638.480000    14.000000     0.800000  8399.9760
```

	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.379428	229.858001	3.789574	0.156203	28.656896
std	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
90%	94122.000000	572.706000	7.000000	0.400000	89.281620
99%	98115.000000	2481.694600	11.000000	0.800000	580.657882
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

```
df.describe(include='object')
```

```
              Order ID       Ship Mode Customer ID  Customer Name   Se
count             9994            9994        9994           9994
unique            5009               4         793            793
top     CA-2014-100111  Standard Class    WB-21850  William Brown  Con
freq                14            5968          37             37

             Country           City       State Region       Product
count           9994           9994        9994   9994            99
unique             1            531          49      4            18
top     United States  New York City  California   West  OFF-PA-100019
freq            9994            915        2001   3203

              Category Sub-Category     Product Name
count             9994         9994             9994
unique               3           17             1850
top     Office Supplies      Binders  Staple envelope
freq              6026         1523               48
```

	Order ID	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Region	Product ID	Ca
count	9994	9994	9994	9994	9994	9994	9994	9994	9994	9994	99
unique	5009	4	793	793	3	1	531	49	4	1862	3
top	CA-2014-100111	Standard Class	WB-21850	William Brown	Consumer	United States	New York City	California	West	OFF-PA-10001970	Of Su
freq	14	5968	37	37	5191	9994	915	2001	3203	19	60

13. Sorting data by specific columns

```
df
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
...                ...        ...        ...             ...
9990    CA-2011-110422 2012-01-22 2012-01-24    Second Class    TB-21
9991    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9992    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
```

```
9993    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9994    CA-2014-119914 2015-05-05 2015-05-10    Second Class    CC-12

           Customer Name    Segment        Country             City
Row ID
1            Claire Gute   Consumer  United States        Henderson
2            Claire Gute   Consumer  United States        Henderson
3         Darrin Van Huff  Corporate  United States      Los Angeles
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky

```
df.sort_values(by=['Order Date'])
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
7981    CA-2011-103800 2012-01-04 2012-01-08  Standard Class    DP-13
740     CA-2011-112326 2012-01-05 2012-01-09  Standard Class    PO-19
741     CA-2011-112326 2012-01-05 2012-01-09  Standard Class    PO-19
742     CA-2011-112326 2012-01-05 2012-01-09  Standard Class    PO-19
1760    CA-2011-141817 2012-01-06 2012-01-13  Standard Class    MB-18
...                ...        ...        ...            ...
5092    CA-2014-156720 2015-12-31 2016-01-04  Standard Class    JM-15
909     CA-2014-143259 2015-12-31 2016-01-04  Standard Class    PO-18
908     CA-2014-143259 2015-12-31 2016-01-04  Standard Class    PO-18
1297    CA-2014-115427 2015-12-31 2016-01-04  Standard Class    EB-13
907     CA-2014-143259 2015-12-31 2016-01-04  Standard Class    PO-18

            Customer Name      Segment        Country           City
Row ID
7981        Darren Powers     Consumer  United States        Houston
740         Phillina Ober  Home Office  United States     Naperville
741         Phillina Ober  Home Office  United States     Naperville
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
7981	CA-2011-103800	2012-01-04	2012-01-08	Standard Class	DP-13000	Darren Powers	Consumer	United States	Houston	Texas
740	CA-2011-112326	2012-01-05	2012-01-09	Standard Class	PO-19195	Phillina Ober	Home Office	United States	Naperville	Illinois
741	CA-2011-112326	2012-01-05	2012-01-09	Standard Class	PO-19195	Phillina Ober	Home Office	United States	Naperville	Illinois
742	CA-2011-112326	2012-01-05	2012-01-09	Standard Class	PO-19195	Phillina Ober	Home Office	United States	Naperville	Illinois
1760	CA-2011-141817	2012-01-06	2012-01-13	Standard Class	MB-18085	Mick Brown	Consumer	United States	Philadelphia	Pennsylvania
...	...	...	...	...	...	...	...	...	...	...
5092	CA-2014-156720	2015-12-31	2016-01-04	Standard Class	JM-15580	Jill Matthias	Consumer	United States	Loveland	Colorado

```
df\
  .sort_values(by=['Customer ID','Ship Date'],
               ascending=False)
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
8342    CA-2014-141481 2015-06-12 2015-06-15     First Class    ZD-21
3815    CA-2013-152471 2014-07-09 2014-07-09        Same Day    ZD-21
3816    CA-2013-152471 2014-07-09 2014-07-09        Same Day    ZD-21
3041    US-2013-147991 2014-05-06 2014-05-10  Standard Class    ZD-21
5898    CA-2013-167682 2014-04-04 2014-04-10  Standard Class    ZD-21
...                ...        ...        ...            ...
1300    CA-2012-121391 2013-10-04 2013-10-07     First Class    AA-10
7469    CA-2011-138100 2012-09-15 2012-09-20  Standard Class    AA-10
7470    CA-2011-138100 2012-09-15 2012-09-20  Standard Class    AA-10
2230    CA-2011-128055 2012-03-31 2012-04-05  Standard Class    AA-10
2231    CA-2011-128055 2012-03-31 2012-04-05  Standard Class    AA-10
```

Row ID	Customer Name	Segment	Country	City	\
8342	Zuschuss Donatelli	Consumer	United States	Los Angeles	
3815	Zuschuss Donatelli	Consumer	United States	Jacksonville	

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Po Co

14. Select a column

```
df
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
...                ...        ...        ...             ...
9990    CA-2011-110422 2012-01-22 2012-01-24    Second Class    TB-21
9991    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9992    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9993    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9994    CA-2014-119914 2015-05-05 2015-05-10    Second Class    CC-12

           Customer Name    Segment        Country             City
Row ID
1            Claire Gute   Consumer  United States        Henderson
2            Claire Gute   Consumer  United States        Henderson
3        Darrin Van Huff  Corporate  United States      Los Angeles
4         Sean O'Donnell   Consumer  United States  Fort Lauderdale
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
...	...	...	...	...	...	...	...	...	...	...
9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	Florida

```
df['Segment']
```

```
Row ID
1         Consumer
2         Consumer
3        Corporate
4         Consumer
5         Consumer
           ...
9990      Consumer
9991      Consumer
9992      Consumer
9993      Consumer
9994      Consumer
Name: Segment, Length: 9994, dtype: object
```

```
df.Segment
```

```
Row ID
1        Consumer
2        Consumer
3       Corporate
4        Consumer
5        Consumer
          ...
9990     Consumer
9991     Consumer
9992     Consumer
9993     Consumer
9994     Consumer
Name: Segment, Length: 9994, dtype: object
```

```
df['Order ID']
```

```
Row ID
1       CA-2013-152156
2       CA-2013-152156
3       CA-2013-138688
4       US-2012-108966
5       US-2012-108966
             ...
9990    CA-2011-110422
9991    CA-2014-121258
9992    CA-2014-121258
9993    CA-2014-121258
9994    CA-2014-119914
Name: Order ID, Length: 9994, dtype: object
```

```
df[0:2]
```

```
              Order ID Order Date  Ship Date     Ship Mode Customer I
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12  Second Class    CG-1252
2       CA-2013-152156 2014-11-09 2014-11-12  Second Class    CG-1252

       Customer Name   Segment        Country       City     State  \
Row ID
1        Claire Gute  Consumer  United States  Henderson  Kentucky
2        Claire Gute  Consumer  United States  Henderson  Kentucky

        Postal Code Region       Product ID   Category Sub-Category
Row ID
1             42420  South  FUR-BO-10001798  Furniture    Bookcases
2             42420  South  FUR-CH-10000454  Furniture       Chairs

                                          Product Name   Sales  Qu
Row ID
1                    Bush Somerset Collection Bookcase  261.96
```

```
2       Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.94
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code
Row ID											
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420

```
df.loc[:,'Customer ID']
```

```
Row ID
1       CG-12520
2       CG-12520
3       DV-13045
4       SO-20335
5       SO-20335
          ...
9990    TB-21400
9991    DB-13060
9992    DB-13060
9993    DB-13060
9994    CC-12220
Name: Customer ID, Length: 9994, dtype: object
```

```
df.iloc[:,6]
```

```
Row ID
1        Consumer
2        Consumer
3       Corporate
4        Consumer
5        Consumer
          ...
9990     Consumer
9991     Consumer
9992     Consumer
9993     Consumer
9994     Consumer
Name: Segment, Length: 9994, dtype: object
```

15. Select multiple columns

```
df[['Order Date','Sales']]
```

```
       Order Date     Sales
Row ID
1      2014-11-09  261.9600
2      2014-11-09  731.9400
3      2014-06-13   14.6200
4      2013-10-11  957.5775
5      2013-10-11   22.3680
...           ...       ...
9990   2012-01-22   25.2480
9991   2015-02-27   91.9600
9992   2015-02-27  258.5760
9993   2015-02-27   29.6000
9994   2015-05-05  243.1600

[9994 rows x 2 columns]
```

	Order Date	Sales
Row ID		
1	2014-11-09	261.9600
2	2014-11-09	731.9400
3	2014-06-13	14.6200
4	2013-10-11	957.5775
5	2013-10-11	22.3680
...	...	...
9990	2012-01-22	25.2480
9991	2015-02-27	91.9600
9992	2015-02-27	258.5760
9993	2015-02-27	29.6000
9994	2015-05-05	243.1600

9994 rows × 2 columns

```
df.loc[:,['Order Date','Sales']]
```

```
       Order Date     Sales
Row ID
```

```
1       2014-11-09  261.9600
2       2014-11-09  731.9400
3       2014-06-13   14.6200
4       2013-10-11  957.5775
5       2013-10-11   22.3680
...            ...       ...
9990    2012-01-22   25.2480
9991    2015-02-27   91.9600
9992    2015-02-27  258.5760
9993    2015-02-27   29.6000
9994    2015-05-05  243.1600

[9994 rows x 2 columns]
```

	Order Date	Sales
Row ID		
1	2014-11-09	261.9600
2	2014-11-09	731.9400
3	2014-06-13	14.6200
4	2013-10-11	957.5775
5	2013-10-11	22.3680
...	...	...
9990	2012-01-22	25.2480
9991	2015-02-27	91.9600
9992	2015-02-27	258.5760
9993	2015-02-27	29.6000
9994	2015-05-05	243.1600

9994 rows × 2 columns

Select custmer-related data

```
DIM_customer = df[['Customer ID','Customer Name','Segment']]
```

```
DIM_customer
```

```
       Customer ID     Customer Name    Segment
Row ID
1         CG-12520       Claire Gute   Consumer
```

```
2          CG-12520       Claire Gute   Consumer
3          DV-13045   Darrin Van Huff  Corporate
4          SO-20335    Sean O'Donnell   Consumer
5          SO-20335    Sean O'Donnell   Consumer
...             ...               ...        ...
9990       TB-21400  Tom Boeckenhauer   Consumer
9991       DB-13060       Dave Brooks   Consumer
9992       DB-13060       Dave Brooks   Consumer
9993       DB-13060       Dave Brooks   Consumer
9994       CC-12220      Chris Cortes   Consumer

[9994 rows x 3 columns]
```

	Customer ID	Customer Name	Segment
Row ID			
1	CG-12520	Claire Gute	Consumer
2	CG-12520	Claire Gute	Consumer
3	DV-13045	Darrin Van Huff	Corporate
4	SO-20335	Sean O'Donnell	Consumer
5	SO-20335	Sean O'Donnell	Consumer
...	...	...	...
9990	TB-21400	Tom Boeckenhauer	Consumer
9991	DB-13060	Dave Brooks	Consumer
9992	DB-13060	Dave Brooks	Consumer
9993	DB-13060	Dave Brooks	Consumer
9994	CC-12220	Chris Cortes	Consumer

9994 rows × 3 columns

```
DIM_customer = DIM_customer.drop_duplicates()
```

```
DIM_customer.shape
```

```
(793, 3)
```

16. More complex selections - .loc and .iloc

```
df.head(n=6)
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
6       CA-2011-115812 2012-06-09 2012-06-14  Standard Class    BH-11

          Customer Name    Segment        Country             City  \
Row ID
1           Claire Gute   Consumer  United States        Henderson
2           Claire Gute   Consumer  United States        Henderson
3       Darrin Van Huff  Corporate  United States      Los Angeles
4        Sean O'Donnell   Consumer  United States  Fort Lauderdale
5        Sean O'Donnell   Consumer  United States  Fort Lauderdale
6       Brosina Hoffman   Consumer  United States      Los Angeles

             State  Postal Code Region       Product ID         Categ
Row ID
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Posta Code

```
df.loc[[2,4,5],['Order Date','Sales']]
```

```
        Order Date     Sales
Row ID
2       2014-11-09  731.9400
4       2013-10-11  957.5775
5       2013-10-11   22.3680
```

	Order Date	Sales
Row ID		
2	2014-11-09	731.9400
4	2013-10-11	957.5775
5	2013-10-11	22.3680

```
df.iloc[[2,4,5],[0,1,2]]
```

```
              Order ID Order Date  Ship Date
Row ID
3       CA-2013-138688 2014-06-13 2014-06-17
5       US-2012-108966 2013-10-11 2013-10-18
6       CA-2011-115812 2012-06-09 2012-06-14
```

	Order ID	Order Date	Ship Date
Row ID			
3	CA-2013-138688	2014-06-13	2014-06-17
5	US-2012-108966	2013-10-11	2013-10-18
6	CA-2011-115812	2012-06-09	2012-06-14

```
df.columns
```

```
Index(['Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer I
       'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Na
       'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

```
pd.Series(df.columns)
```

```
0          Order ID
1        Order Date
2         Ship Date
3         Ship Mode
4       Customer ID
5     Customer Name
6           Segment
7           Country
8              City
9             State
10      Postal Code
11           Region
12       Product ID
13         Category
14     Sub-Category
15     Product Name
16            Sales
17         Quantity
18         Discount
19           Profit
```

```
pd.Series(df.columns).str.contains('ID')
```

```
0      True
1     False
2     False
3     False
4      True
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12     True
13    False
14    False
15    False
16    False
17    False
18    False
19    False
```

```
df.columns[pd.Series(df.columns).str.contains('ID')]
```

```
Index(['Order ID', 'Customer ID', 'Product ID'], dtype='object')
```

```
df.columns[pd.Series(df.columns).str.endswith('ID')]
```

```
Index(['Order ID', 'Customer ID', 'Product ID'], dtype='object')
```

```
df[df.columns[pd.Series(df.columns).str.endswith('ID')]]
```

```
              Order ID Customer ID       Product ID
Row ID
1       CA-2013-152156    CG-12520  FUR-BO-10001798
2       CA-2013-152156    CG-12520  FUR-CH-10000454
3       CA-2013-138688    DV-13045  OFF-LA-10000240
4       US-2012-108966    SO-20335  FUR-TA-10000577
5       US-2012-108966    SO-20335  OFF-ST-10000760
...                ...         ...              ...
9990    CA-2011-110422    TB-21400  FUR-FU-10001889
9991    CA-2014-121258    DB-13060  FUR-FU-10000747
9992    CA-2014-121258    DB-13060  TEC-PH-10003645
9993    CA-2014-121258    DB-13060  OFF-PA-10004041
9994    CA-2014-119914    CC-12220  OFF-AP-10002684

[9994 rows x 3 columns]
```

17. Indexing (aka filtering)

```
df[0:2]
```

```
              Order ID Order Date  Ship Date     Ship Mode Customer I
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12  Second Class    CG-1252
2       CA-2013-152156 2014-11-09 2014-11-12  Second Class    CG-1252

       Customer Name    Segment        Country       City     State  \
Row ID
1        Claire Gute  Consumer  United States  Henderson  Kentucky
2        Claire Gute  Consumer  United States  Henderson  Kentucky

        Postal Code Region       Product ID   Category Sub-Category
Row ID
1             42420  South  FUR-BO-10001798  Furniture    Bookcases
2             42420  South  FUR-CH-10000454  Furniture       Chairs

                                             Product Name   Sales  Qu
Row ID
1                       Bush Somerset Collection Bookcase  261.96
2       Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.94
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code
Row ID											
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420

```
df.loc[[2,4],:]
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
```

```
         Customer Name    Segment        Country             City
Row ID
2          Claire Gute   Consumer  United States        Henderson  Ker
4       Sean O'Donnell   Consumer  United States  Fort Lauderdale   Fl

        Postal Code Region       Product ID    Category Sub-Category
Row ID
2              42420  South  FUR-CH-10000454   Furniture       Chairs
4              33311  South  FUR-TA-10000577   Furniture       Tables

                                                Product Name     Sales
Row ID
2       Hon Deluxe Fabric Upholstered Stacking Chairs,...   731.9400
4           Bretford CR4500 Series Slim Rectangular Table   957.5775
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Posta Code
Row ID											
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	4242
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	3331

18. Boolean indexing by isin

```
df['Ship Mode'].value_counts()
```

```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

```
df['Ship Mode'].isin(['First Class','Same Day'])
```

```
Row ID
1       False
```

```
2       False
3       False
4       False
5       False
        ...
9990    False
9991    False
9992    False
9993    False
9994    False
Name: Ship Mode, Length: 9994, dtype: bool
```

```
df.loc[df['Ship Mode'].isin(['First Class', 'Same Day']), :]
```

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID
36	CA-2013-117590	2014-12-09	2014-12-11	First Class	GH-14485
37	CA-2013-117590	2014-12-09	2014-12-11	First Class	GH-14485
45	CA-2013-118255	2014-03-12	2014-03-14	First Class	ON-18715
46	CA-2013-118255	2014-03-12	2014-03-14	First Class	ON-18715
56	CA-2013-111682	2014-06-18	2014-06-19	First Class	TB-21055
...	...	...	...	...	...
9934	CA-2011-166555	2012-07-11	2012-07-14	First Class	JK-15205
9962	CA-2012-168088	2013-03-19	2013-03-22	First Class	CM-12655
9963	CA-2012-168088	2013-03-19	2013-03-22	First Class	CM-12655
9964	CA-2012-143700	2013-07-26	2013-07-26	Same Day	AS-10240
9982	CA-2014-163566	2015-08-04	2015-08-07	First Class	TB-21055

Row ID	Customer Name	Segment	Country	City
36	Gene Hale	Corporate	United States	Richardson
37	Gene Hale	Corporate	United States	Richardson
45	Odella Nelson	Corporate	United States	Eagan

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Pc Cc
Row ID											
36	CA-2013-117590	2014-12-09	2014-12-11	First Class	GH-14485	Gene Hale	Corporate	United States	Richardson	Texas	75
37	CA-2013-117590	2014-12-09	2014-12-11	First Class	GH-14485	Gene Hale	Corporate	United States	Richardson	Texas	75
45	CA-2013-118255	2014-03-12	2014-03-14	First Class	ON-18715	Odella Nelson	Corporate	United States	Eagan	Minnesota	55
46	CA-2013-118255	2014-03-12	2014-03-14	First Class	ON-18715	Odella Nelson	Corporate	United States	Eagan	Minnesota	55
56	CA-2013-111682	2014-06-18	2014-06-19	First Class	TB-21055	Ted Butterfield	Consumer	United States	Troy	New York	12
...	...	...	...	...	...	...	...	...	...	...	...
9934	CA-2011-166555	2012-07-11	2012-07-14	First Class	JK-15205	Jamie Kunitz	Consumer	United States	Niagara Falls	New York	14
	CA-	2013-	2013-	First	CM-	Corinna	Home	United			

```
df[df['Ship Mode'].isin(['First Class'])]
```

```
             Order ID Order Date  Ship Date    Ship Mode Customer ID
Row ID
36       CA-2013-117590 2014-12-09 2014-12-11  First Class    GH-14485
37       CA-2013-117590 2014-12-09 2014-12-11  First Class    GH-14485
45       CA-2013-118255 2014-03-12 2014-03-14  First Class    ON-18715
46       CA-2013-118255 2014-03-12 2014-03-14  First Class    ON-18715
56       CA-2013-111682 2014-06-18 2014-06-19  First Class    TB-21055
...                 ...        ...        ...          ...         ...
9928     CA-2012-159534 2013-03-20 2013-03-23  First Class    DH-13075
9934     CA-2011-166555 2012-07-11 2012-07-14  First Class    JK-15205
9962     CA-2012-168088 2013-03-19 2013-03-22  First Class    CM-12655
9963     CA-2012-168088 2013-03-19 2013-03-22  First Class    CM-12655
9982     CA-2014-163566 2015-08-04 2015-08-07  First Class    TB-21055

          Customer Name      Segment        Country           City
Row ID
36              Gene Hale    Corporate  United States     Richardson
```

```
37          Gene Hale    Corporate  United States     Richardson
45       Odella Nelson    Corporate  United States          Eagan
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Posta Code
Row ID											
36	CA-2013-117590	2014-12-09	2014-12-11	First Class	GH-14485	Gene Hale	Corporate	United States	Richardson	Texas	7508(
37	CA-2013-117590	2014-12-09	2014-12-11	First Class	GH-14485	Gene Hale	Corporate	United States	Richardson	Texas	7508(
45	CA-2013-118255	2014-03-12	2014-03-14	First Class	ON-18715	Odella Nelson	Corporate	United States	Eagan	Minnesota	5512
46	CA-2013-118255	2014-03-12	2014-03-14	First Class	ON-18715	Odella Nelson	Corporate	United States	Eagan	Minnesota	5512
56	CA-2013-111682	2014-06-18	2014-06-19	First Class	TB-21055	Ted Butterfield	Consumer	United States	Troy	New York	1218(
...	...	...	...	...	...	...	...	...	...	...	...
9928	CA-2012-159534	2013-03-20	2013-03-23	First Class	DH-13075	Dave Hallsten	Corporate	United States	New York City	New York	1003
9934	CA-2011-166555	2012-07-11	2012-07-14	First Class	JK-15205	Jamie Kunitz	Consumer	United States	Niagara Falls	New York	1430
9962	CA-2012-168088	2013-03-19	2013-03-22	First Class	CM-12655	Corinna Mitchell	Home Office	United States	Houston	Texas	7704
9963	CA-2012-168088	2013-03-19	2013-03-22	First Class	CM-12655	Corinna Mitchell	Home Office	United States	Houston	Texas	7704
9982	CA-2014-163566	2015-08-04	2015-08-07	First Class	TB-21055	Ted Butterfield	Consumer	United States	Fairfield	Ohio	4501

1538 rows × 20 columns

19. Boolean indexing by conditions

```
df['Profit']
```

```
Row ID
1        41.9136
2       219.5820
3         6.8714
4      -383.0310
5         2.5164
          ...
9990      4.1028
9991     15.6332
9992     19.3932
9993     13.3200
9994     72.9480
Name: Profit, Length: 9994, dtype: float64
```

```
df['Profit'] < 0
```

```
Row ID
1       False
2       False
3       False
4        True
5       False
        ...
9990    False
9991    False
9992    False
9993    False
9994    False
Name: Profit, Length: 9994, dtype: bool
```

```
df[df['Profit'] < 0]
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
15      US-2012-118983 2013-11-22 2013-11-26  Standard Class    HP-14
16      US-2012-118983 2013-11-22 2013-11-26  Standard Class    HP-14
24      US-2014-156909 2015-07-17 2015-07-19    Second Class    SF-20
28      US-2012-150630 2013-09-17 2013-09-21  Standard Class    TB-21
...                ...        ...        ...             ...
9921    CA-2013-149272 2014-03-16 2014-03-20  Standard Class    MY-18
9922    CA-2011-111360 2012-11-24 2012-11-30  Standard Class    AT-10
```

```
9932    CA-2012-104948 2013-11-13 2013-11-17  Standard Class    KH-16
9938    CA-2013-164889 2014-06-04 2014-06-07    Second Class    CP-12
9963    CA-2012-168088 2013-03-19 2013-03-22     First Class    CM-12

           Customer Name      Segment        Country             City
Row ID
4          Sean O'Donnell     Consumer  United States  Fort Lauderdale
15          Harold Pawlan  Home Office  United States       Fort Worth
16          Harold Pawlan  Home Office  United States       Fort Worth
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
15	US-2012-118983	2013-11-22	2013-11-26	Standard Class	HP-14815	Harold Pawlan	Home Office	United States	Fort Worth	Texas

```
cond = (df['Profit'] < 0) & (df['Ship Mode'].isin(['First Class']))
cond
```

```
Row ID
1       False
2       False
3       False
4       False
5       False
        ...
9990    False
9991    False
9992    False
9993    False
9994    False
Length: 9994, dtype: bool
```

```
df[cond]
```

```
             Order ID Order Date  Ship Date    Ship Mode Customer ID
Row ID
37      CA-2013-117590 2014-12-09 2014-12-11  First Class    GH-14485
76      US-2014-118038 2015-12-10 2015-12-12  First Class    KB-16600
77      US-2014-118038 2015-12-10 2015-12-12  First Class    KB-16600
85      US-2014-119662 2015-11-14 2015-11-17  First Class    CS-12400
131     US-2014-164147 2015-02-03 2015-02-06  First Class    DW-13585
...                ...        ...        ...          ...         ...
9877    US-2014-166324 2015-04-21 2015-04-22  First Class    BE-11455
9878    US-2014-166324 2015-04-21 2015-04-22  First Class    BE-11455
9879    US-2014-166324 2015-04-21 2015-04-22  First Class    BE-11455
9913    CA-2012-132388 2013-10-10 2013-10-12  First Class    KN-16390
9963    CA-2012-168088 2013-03-19 2013-03-22  First Class    CM-12655

            Customer Name      Segment        Country           City
Row ID
```

```
37            Gene Hale    Corporate  United States     Richardson
76          Ken Brennan    Corporate  United States        Houston
77          Ken Brennan    Corporate  United States        Houston
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Posta Code
Row ID											
37	CA-2013-117590	2014-12-09	2014-12-11	First Class	GH-14485	Gene Hale	Corporate	United States	Richardson	Texas	7508(
76	US-2014-118038	2015-12-10	2015-12-12	First Class	KB-16600	Ken Brennan	Corporate	United States	Houston	Texas	7704
77	US-2014-118038	2015-12-10	2015-12-12	First Class	KB-16600	Ken Brennan	Corporate	United States	Houston	Texas	7704
85	US-2014-119662	2015-11-14	2015-11-17	First Class	CS-12400	Christopher Schild	Home Office	United States	Chicago	Illinois	6062
131	US-2014-164147	2015-02-03	2015-02-06	First Class	DW-13585	Dorothy Wardle	Corporate	United States	Columbus	Ohio	4322
...	...	...	...	...	...	...	...	...	...	...	...
9877	US-2014-166324	2015-04-21	2015-04-22	First Class	BE-11455	Brad Eason	Home Office	United States	Cleveland	Ohio	4410
9878	US-2014-166324	2015-04-21	2015-04-22	First Class	BE-11455	Brad Eason	Home Office	United States	Cleveland	Ohio	4410
9879	US-2014-166324	2015-04-21	2015-04-22	First Class	BE-11455	Brad Eason	Home Office	United States	Cleveland	Ohio	4410
9913	CA-2012-132388	2013-10-10	2013-10-12	First Class	KN-16390	Katherine Nockton	Corporate	United States	Santa Barbara	California	9310
9963	CA-2012-168088	2013-03-19	2013-03-22	First Class	CM-12655	Corinna Mitchell	Home Office	United States	Houston	Texas	7704

293 rows × 20 columns

20. Add column

```
df['Unit Sales'] = df['Sales']/df['Quantity']
df[['Product ID','Unit Sales']]
```

```
             Product ID  Unit Sales
Row ID
1       FUR-BO-10001798    130.9800
2       FUR-CH-10000454    243.9800
3       OFF-LA-10000240      7.3100
4       FUR-TA-10000577    191.5155
5       OFF-ST-10000760     11.1840
...                 ...         ...
9990    FUR-FU-10001889      8.4160
9991    FUR-FU-10000747     45.9800
9992    TEC-PH-10003645    129.2880
9993    OFF-PA-10004041      7.4000
9994    OFF-AP-10002684    121.5800

[9994 rows x 2 columns]
```

	Product ID	Unit Sales
Row ID		
1	FUR-BO-10001798	130.9800
2	FUR-CH-10000454	243.9800
3	OFF-LA-10000240	7.3100
4	FUR-TA-10000577	191.5155
5	OFF-ST-10000760	11.1840
...	...	...
9990	FUR-FU-10001889	8.4160
9991	FUR-FU-10000747	45.9800
9992	TEC-PH-10003645	129.2880
9993	OFF-PA-10004041	7.4000
9994	OFF-AP-10002684	121.5800

9994 rows × 2 columns

```
df
```

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20
...	...	...	...	...	
9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	TB-21
9991	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9992	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9993	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9994	CA-2014-119914	2015-05-05	2015-05-10	Second Class	CC-12

Row ID	Customer Name	Segment	Country	City
1	Claire Gute	Consumer	United States	Henderson
2	Claire Gute	Consumer	United States	Henderson
3	Darrin Van Huff	Corporate	United States	Los Angeles
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California

```
#df['Ones'] = 1
df = df.assign(Ones = 1)
```

```
df.head()
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20

          Customer Name    Segment        Country             City  \
Row ID
1           Claire Gute   Consumer  United States        Henderson
2           Claire Gute   Consumer  United States        Henderson
3       Darrin Van Huff  Corporate  United States      Los Angeles
4        Sean O'Donnell   Consumer  United States  Fort Lauderdale
5        Sean O'Donnell   Consumer  United States  Fort Lauderdale

             State  Postal Code Region       Product ID         Categ
Row ID
1         Kentucky        42420  South  FUR-BO-10001798        Furnit
2         Kentucky        42420  South  FUR-CH-10000454        Furnit
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Posta Code
Row ID											
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	4242
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	4242
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	9003
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	3331

```
df1 = df
```

```
df1
```

```
             Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
...                ...        ...        ...             ...
9990    CA-2011-110422 2012-01-22 2012-01-24    Second Class    TB-21
9991    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9992    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9993    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9994    CA-2014-119914 2015-05-05 2015-05-10    Second Class    CC-12

            Customer Name    Segment        Country             City
Row ID
1             Claire Gute   Consumer  United States        Henderson
2             Claire Gute   Consumer  United States        Henderson
3          Darrin Van Huff  Corporate  United States      Los Angeles
4           Sean O'Donnell   Consumer  United States  Fort Lauderdale
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
...	...	...	...	...	...	...	...	...	...	...
9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	Florida
9991	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9992	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9993	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California
9994	CA-2014-119914	2015-05-05	2015-05-10	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	California

9994 rows × 22 columns

```
del df1['Ones']
```

```
df
```

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20
...	...	...	...	...	
9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	TB-21
9991	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9992	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9993	CA-2014-121258	2015-02-27	2015-03-04	Standard Class	DB-13
9994	CA-2014-119914	2015-05-05	2015-05-10	Second Class	CC-12

Row ID	Customer Name	Segment	Country	City
1	Claire Gute	Consumer	United States	Henderson
2	Claire Gute	Consumer	United States	Henderson
3	Darrin Van Huff	Corporate	United States	Los Angeles
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky
3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California
4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida
5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida

```
df1 = df.copy()
```

```
del df1['Unit Sales']
```

```
df
```

```
             Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
...                ...        ...        ...             ...
9990    CA-2011-110422 2012-01-22 2012-01-24    Second Class    TB-21
9991    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9992    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9993    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9994    CA-2014-119914 2015-05-05 2015-05-10    Second Class    CC-12

            Customer Name    Segment        Country             City
```

Row ID				
1	Claire Gute	Consumer	United States	Henderson
2	Claire Gute	Consumer	United States	Henderson
3	Darrin Van Huff	Corporate	United States	Los Angeles
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										

21. Delete (drop) column (safely)

```
df.drop(columns=['Unit Sales'], inplace=True)
df.columns
```

```
Index(['Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer I
       'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Na
       'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

```
df1 = df.drop(columns=['Order Date','Ship Date']) # if not inplace=True
df1.columns
```

```
Index(['Order ID', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segme
       'Country', 'City', 'State', 'Postal Code', 'Region', 'Product I
       'Category', 'Sub-Category', 'Product Name', 'Sales', 'Quantity'
       'Discount', 'Profit'],
      dtype='object')
```

```
df['Ones'] = 0
```

```
df
```

```
              Order ID Order Date  Ship Date       Ship Mode Customer
Row ID
1       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
2       CA-2013-152156 2014-11-09 2014-11-12    Second Class    CG-12
3       CA-2013-138688 2014-06-13 2014-06-17    Second Class    DV-13
4       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
5       US-2012-108966 2013-10-11 2013-10-18  Standard Class    SO-20
...                ...        ...        ...             ...
9990    CA-2011-110422 2012-01-22 2012-01-24    Second Class    TB-21
9991    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9992    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
9993    CA-2014-121258 2015-02-27 2015-03-04  Standard Class    DB-13
```

```
9994    CA-2014-119914 2015-05-05 2015-05-10    Second Class    CC-12

            Customer Name    Segment        Country             City
Row ID
1             Claire Gute   Consumer  United States        Henderson
2             Claire Gute   Consumer  United States        Henderson
3          Darrin Van Huff  Corporate  United States      Los Angeles
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State
Row ID										
1	CA-2013-	2014-	2014-	Second	CG-	Claire Gute	Consumer	United	Henderson	Kentucky

22. Missing data

```
test = pd.read_csv('/content/drive/MyDrive/AIS_DG/Telco-Churn.csv')
```

```
test.head()
```

```
   customerID  gender  SeniorCitizen Partner Dependents  tenure Phone
0  7590-VHVEG  Female              0     Yes         No       1
1  5575-GNVDE    Male              0      No         No      34
2  3668-QPYBK    Male              0      No         No       2
3  7795-CFOCW    Male              0      No         No      45
4  9237-HQITU  Female              0      No         No       2

      MultipleLines InternetService OnlineSecurity OnlineBackup  \
0  No phone service             DSL             No          Yes
1                No             DSL            Yes           No
2                No             DSL            Yes          Yes
3  No phone service             DSL            Yes           No
4                No     Fiber optic             No           No

  DeviceProtection TechSupport StreamingTV StreamingMovies        Cor
0               No          No          No              No  Month-to-
1              Yes          No          No              No        One
2               No          No          No              No  Month-to-
3              Yes         Yes          No              No        One
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetServi
0	7590-	Female	0	Yes	No	1	No	No phone	DSL

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64 
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64 
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
```

```
test['TotalCharges'].value_counts().index
```

```
Index([' ', '20.2', '19.75', '20.05', '19.9', '19.65', '45.3', '19.55'
       '20.15', '20.25',
       ...
       '3306.85', '424.75', '6565.85', '2117.2', '203.95', '6849.4', '
       '130.15', '3211.9', '6844.5'],
      dtype='object', length=6531)
```

```
test.isnull().any()
```

```
customerID          False
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      False
```

```
OnlineBackup        False
DeviceProtection    False
TechSupport         False
StreamingTV         False
StreamingMovies     False
Contract            False
PaperlessBilling    False
PaymentMethod       False
MonthlyCharges      False
TotalCharges        False
```

```
churnData = pd.read_csv('/content/drive/MyDrive/AIS_DG/Telco-Churn.csv',
```

```
churnData.isna().any()
```

```
customerID          False
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      False
OnlineBackup        False
DeviceProtection    False
TechSupport         False
StreamingTV         False
StreamingMovies     False
Contract            False
PaperlessBilling    False
PaymentMethod       False
MonthlyCharges      False
TotalCharges         True
```

```
churnData['TotalCharges'].isna()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
7038    False
7039    False
7040    False
7041    False
```

```
7042    False
Name: TotalCharges, Length: 7043, dtype: bool
```

```
churnData.head()
```

```
   customerID  gender  SeniorCitizen Partner Dependents  tenure Phone
0  7590-VHVEG  Female              0     Yes         No       1
1  5575-GNVDE    Male              0      No         No      34
2  3668-QPYBK    Male              0      No         No       2
3  7795-CFOCW    Male              0      No         No      45
4  9237-HQITU  Female              0      No         No       2

      MultipleLines InternetService OnlineSecurity OnlineBackup  \
0  No phone service             DSL             No          Yes
1                No             DSL            Yes           No
2                No             DSL            Yes          Yes
3  No phone service             DSL            Yes           No
4                No     Fiber optic             No           No

  DeviceProtection TechSupport StreamingTV StreamingMovies        Cor
0               No          No          No              No  Month-to-
1              Yes          No          No              No        One
2               No          No          No              No  Month-to-
3              Yes         Yes          No              No        One
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetServi
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic

```
churnData[churnData['TotalCharges'].isna()]
```

```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
488   4472-LVYGI  Female              0     Yes        Yes       0
753   3115-CZMZD    Male              0      No        Yes       0
936   5709-LVOEQ  Female              0     Yes        Yes       0
1082  4367-NUYAO    Male              0     Yes        Yes       0
1340  1371-DWPAZ  Female              0     Yes        Yes       0
```

```
3331  7644-OMVMY    Male              0     Yes         Yes        0
3826  3213-VVOLG    Male              0     Yes         Yes        0
4380  2520-SGTTA  Female              0     Yes         Yes        0
5218  2923-ARZLG    Male              0     Yes         Yes        0
6670  4075-WKNIU  Female              0     Yes         Yes        0
6754  2775-SEFEE    Male              0      No         Yes        0

     PhoneService     MultipleLines InternetService       OnlineSecur
488            No  No phone service             DSL
753           Yes                No              No  No internet serv
936           Yes                No             DSL
1082          Yes               Yes              No  No internet serv
1340           No  No phone service             DSL
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	DSL
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	No
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	DSL
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	No
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	DSL
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	No
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	No
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	No	No
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	No	No
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Yes	DSL
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Yes	DSL

```
churnData[churnData['tenure'] == 0]
```

```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
488   4472-LVYGI  Female              0     Yes         Yes       0
753   3115-CZMZD    Male              0      No         Yes       0
936   5709-LVOEQ  Female              0     Yes         Yes       0
1082  4367-NUYAO    Male              0     Yes         Yes       0
1340  1371-DWPAZ  Female              0     Yes         Yes       0
3331  7644-OMVMY    Male              0     Yes         Yes       0
3826  3213-VVOLG    Male              0     Yes         Yes       0
4380  2520-SGTTA  Female              0     Yes         Yes       0
```

```
5218  2923-ARZLG    Male              0     Yes         Yes       0
6670  4075-WKNIU  Female              0     Yes         Yes       0
6754  2775-SEFEE    Male              0      No         Yes       0

     PhoneService     MultipleLines InternetService       OnlineSecur
488            No  No phone service             DSL
753           Yes                No              No  No internet serv
936           Yes                No             DSL
1082          Yes               Yes              No  No internet serv
1340           No  No phone service             DSL
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	DSL
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	No
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	DSL
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	No
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	DSL
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	No
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	No
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	No	No
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	No	No
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Yes	DSL
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Yes	DSL

23. Removing missing data rows

```
churnData.shape
```

```
(7043, 21)
```

```
churnData_1 = churnData.dropna(how='any')
churnData_1.shape
```

```
(7032, 21)
```

```
churnData_1.isnull().any()
```

```
customerID          False
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      False
OnlineBackup        False
DeviceProtection    False
TechSupport         False
StreamingTV         False
StreamingMovies     False
Contract            False
PaperlessBilling    False
PaymentMethod       False
MonthlyCharges      False
TotalCharges        False
```

24. Filling missing data: investigate

```
churnData.isnull()
```

```
      customerID  gender  SeniorCitizen  Partner  Dependents  tenure
0          False   False          False    False       False   False
1          False   False          False    False       False   False
2          False   False          False    False       False   False
3          False   False          False    False       False   False
4          False   False          False    False       False   False
...          ...     ...            ...      ...         ...     ...
7038       False   False          False    False       False   False
7039       False   False          False    False       False   False
7040       False   False          False    False       False   False
7041       False   False          False    False       False   False
7042       False   False          False    False       False   False

      PhoneService  MultipleLines  InternetService  OnlineSecurity  \
0            False          False            False           False
1            False          False            False           False
2            False          False            False           False
3            False          False            False           False
```

```
4              False          False            False           False
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
7038	False	False	False	False	False	False	False	False	False
7039	False	False	False	False	False	False	False	False	False
7040	False	False	False	False	False	False	False	False	False
7041	False	False	False	False	False	False	False	False	False
7042	False	False	False	False	False	False	False	False	False

7043 rows × 21 columns

```
churnData.isnull().any()
```

```
customerID          False
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      False
OnlineBackup        False
DeviceProtection    False
TechSupport         False
StreamingTV         False
StreamingMovies     False
Contract            False
PaperlessBilling    False
PaymentMethod       False
MonthlyCharges      False
TotalCharges         True
```

```
churnData[churnData['TotalCharges'].isnull()]
```

```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
488   4472-LVYGI  Female              0     Yes        Yes       0
753   3115-CZMZD    Male              0      No        Yes       0
936   5709-LVOEQ  Female              0     Yes        Yes       0
1082  4367-NUYAO    Male              0     Yes        Yes       0
1340  1371-DWPAZ  Female              0     Yes        Yes       0
3331  7644-OMVMY    Male              0     Yes        Yes       0
3826  3213-VVOLG    Male              0     Yes        Yes       0
4380  2520-SGTTA  Female              0     Yes        Yes       0
5218  2923-ARZLG    Male              0     Yes        Yes       0
6670  4075-WKNIU  Female              0     Yes        Yes       0
6754  2775-SEFEE    Male              0      No        Yes       0

     PhoneService     MultipleLines InternetService       OnlineSecur
488            No  No phone service             DSL
753           Yes                No              No  No internet serv
936           Yes                No             DSL
1082          Yes               Yes              No  No internet serv
1340           No  No phone service             DSL
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	DSL
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	No
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	DSL
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	No
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	DSL
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	No
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	No
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	No	No
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	No	No
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Yes	DSL
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Yes	DSL

25. Filling missing data: fillna

```
churnData.dropna().isna().any()
```

```
customerID          False
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      False
OnlineBackup        False
DeviceProtection    False
TechSupport         False
StreamingTV         False
StreamingMovies     False
Contract            False
PaperlessBilling    False
PaymentMethod       False
MonthlyCharges      False
TotalCharges        False
```

```
churnData_2 = churnData.fillna(value={'TotalCharges':0, 'tenure':0})
churnData_2.isnull().any()
```

```
customerID          False
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      False
OnlineBackup        False
DeviceProtection    False
TechSupport         False
StreamingTV         False
StreamingMovies     False
Contract            False
PaperlessBilling    False
PaymentMethod       False
MonthlyCharges      False
TotalCharges        False
```

26. Statistical operations

```
churnData.mean()
```

```
SeniorCitizen        0.162147
tenure              32.371149
MonthlyCharges      64.761692
TotalCharges      2283.300441
dtype: float64

<ipython-input-132-cccca424e68e>:1: FutureWarning: The default value o
  churnData.mean()
```

```
churnData.std()
```

```
SeniorCitizen        0.368612
tenure              24.559481
MonthlyCharges      30.090047
TotalCharges      2266.771362
dtype: float64

<ipython-input-133-efd5c85dfcbf>:1: FutureWarning: The default value o
  churnData.std()
```

```
churnData.median()
```

```
SeniorCitizen        0.000
tenure              29.000
MonthlyCharges      70.350
TotalCharges      1397.475
dtype: float64

<ipython-input-134-61ddd473a11a>:1: FutureWarning: The default value o
  churnData.median()
```

```
churnData.quantile([0.1,0.9,0.99,1])
```

```
      SeniorCitizen  tenure  MonthlyCharges  TotalCharges
0.10            0.0     2.0          20.050        84.600
0.90            1.0    69.0         102.600      5976.640
0.99            1.0    72.0         114.729      8039.883
1.00            1.0    72.0         118.750      8684.800
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
0.10	0.0	2.0	20.050	84.600
0.90	1.0	69.0	102.600	5976.640
0.99	1.0	72.0	114.729	8039.883
1.00	1.0	72.0	118.750	8684.800

```
<ipython-input-135-48bfe3edbd92>:1: FutureWarning: The default value o
  churnData.quantile([0.1,0.9,0.99,1])
```

27. Apply

```
churnData.select_dtypes('object')
```

```
      customerID  gender Partner Dependents PhoneService     Multiple
0     7590-VHVEG  Female     Yes         No            No  No phone se
1     5575-GNVDE    Male      No         No           Yes
2     3668-QPYBK    Male      No         No           Yes
3     7795-CFOCW    Male      No         No            No  No phone se
4     9237-HQITU  Female      No         No           Yes
...          ...     ...     ...        ...           ...
7038  6840-RESVB    Male     Yes        Yes           Yes
7039  2234-XADUH  Female     Yes        Yes           Yes
7040  4801-JZAZL  Female     Yes        Yes            No  No phone se
7041  8361-LTMKD    Male     Yes         No           Yes
7042  3186-AJIEK    Male      No         No           Yes

     InternetService OnlineSecurity OnlineBackup DeviceProtection Tec
0                DSL             No          Yes               No
1                DSL            Yes           No              Yes
2                DSL            Yes          Yes               No
3                DSL            Yes           No              Yes
4        Fiber optic             No           No               No
```

	customerID	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity
0	7590-VHVEG	Female	Yes	No	No	No phone service	DSL	No
1	5575-GNVDE	Male	No	No	Yes	No	DSL	Yes
2	3668-QPYBK	Male	No	No	Yes	No	DSL	Yes
3	7795-CFOCW	Male	No	No	No	No phone service	DSL	Yes
4	9237-HQITU	Female	No	No	Yes	No	Fiber optic	No
...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	Yes	Yes	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	Yes	Yes	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	Yes	Yes	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	Yes	No	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	No	No	Yes	No	Fiber optic	Yes

7043 rows × 17 columns

```
churnData._get_numeric_data()
```

```
      SeniorCitizen  tenure  MonthlyCharges  TotalCharges
0                 0       1           29.85         29.85
1                 0      34           56.95       1889.50
2                 0       2           53.85        108.15
3                 0      45           42.30       1840.75
4                 0       2           70.70        151.65
...             ...     ...             ...           ...
7038              0      24           84.80       1990.50
7039              0      72          103.20       7362.90
7040              0      11           29.60        346.45
7041              1       4           74.40        306.60
7042              0      66          105.65       6844.50

[7043 rows x 4 columns]
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
0	0	1	29.85	29.85
1	0	34	56.95	1889.50
2	0	2	53.85	108.15
3	0	45	42.30	1840.75
4	0	2	70.70	151.65
...	...	...	...	...
7038	0	24	84.80	1990.50
7039	0	72	103.20	7362.90
7040	0	11	29.60	346.45
7041	1	4	74.40	306.60
7042	0	66	105.65	6844.50

```
churnData['tenure']/churnData['tenure'].max()
```

```
0       0.013889
1       0.472222
2       0.027778
3       0.625000
4       0.027778
          ...
7038    0.333333
7039    1.000000
7040    0.152778
7041    0.055556
7042    0.916667
Name: tenure, Length: 7043, dtype: float64
```

```
df.select_dtypes('datetime64[ns]')
```

```
       Order Date  Ship Date
Row ID
1      2014-11-09 2014-11-12
2      2014-11-09 2014-11-12
3      2014-06-13 2014-06-17
4      2013-10-11 2013-10-18
5      2013-10-11 2013-10-18
...           ...        ...
9990   2012-01-22 2012-01-24
9991   2015-02-27 2015-03-04
9992   2015-02-27 2015-03-04
9993   2015-02-27 2015-03-04
9994   2015-05-05 2015-05-10
```

```
[9994 rows x 2 columns]
```

	Order Date	Ship Date
Row ID		
1	2014-11-09	2014-11-12
2	2014-11-09	2014-11-12
3	2014-06-13	2014-06-17
4	2013-10-11	2013-10-18
5	2013-10-11	2013-10-18
...	...	...
9990	2012-01-22	2012-01-24
9991	2015-02-27	2015-03-04
9992	2015-02-27	2015-03-04
9993	2015-02-27	2015-03-04
9994	2015-05-05	2015-05-10

9994 rows × 2 columns

```
churnData_num = churnData._get_numeric_data()
churnData_num.apply(lambda x:x/x.max())
```

```
      SeniorCitizen    tenure  MonthlyCharges  TotalCharges
0               0.0  0.013889        0.251368      0.003437
1               0.0  0.472222        0.479579      0.217564
2               0.0  0.027778        0.453474      0.012453
3               0.0  0.625000        0.356211      0.211951
4               0.0  0.027778        0.595368      0.017462
...             ...       ...             ...           ...
7038            0.0  0.333333        0.714105      0.229194
7039            0.0  1.000000        0.869053      0.847792
7040            0.0  0.152778        0.249263      0.039892
7041            1.0  0.055556        0.626526      0.035303
7042            0.0  0.916667        0.889684      0.788101

[7043 rows x 4 columns]
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
0	0.0	0.013889	0.251368	0.003437
1	0.0	0.472222	0.479579	0.217564
2	0.0	0.027778	0.453474	0.012453
3	0.0	0.625000	0.356211	0.211951
4	0.0	0.027778	0.595368	0.017462
...	...	...	...	...
7038	0.0	0.333333	0.714105	0.229194

28. Transform

```
x = churnData.transform({'TotalCharges':[np.log, lambda x: x+1],
                         'PaperlessBilling':lambda x: 1 if x=='Yes' else 0})
x.columns = ['TotalCharge_Log','PlusOne','PaperLess']
x
```

```
      TotalCharge_Log  PlusOne  PaperLess
0            3.396185    30.85          1
1            7.544068  1890.50          0
2            4.683519   109.15          1
3            7.517928  1841.75          0
4            5.021575   152.65          1
...               ...      ...        ...
7038         7.596141  1991.50          1
7039         8.904209  7363.90          1
7040         5.847739   347.45          1
7041         5.725544   307.60          1
7042         8.831201  6845.50          1

[7043 rows x 3 columns]
```

	TotalCharge_Log	PlusOne	PaperLess
0	3.396185	30.85	1
1	7.544068	1890.50	0
2	4.683519	109.15	1
3	7.517928	1841.75	0
4	5.021575	152.65	1

29. Concatenate

```
churnData['customerID']
```

```
0       7590-VHVEG
1       5575-GNVDE
2       3668-QPYBK
3       7795-CFOCW
4       9237-HQITU
           ...
7038    6840-RESVB
7039    2234-XADUH
7040    4801-JZAZL
7041    8361-LTMKD
7042    3186-AJIEK
Name: customerID, Length: 7043, dtype: object
```

```
churnData[['TotalCharges', 'MonthlyCharges']]
```

```
      TotalCharges  MonthlyCharges
0            29.85           29.85
1          1889.50           56.95
2           108.15           53.85
3          1840.75           42.30
4           151.65           70.70
...            ...             ...
7038       1990.50           84.80
7039       7362.90          103.20
7040        346.45           29.60
7041        306.60           74.40
7042       6844.50          105.65

[7043 rows x 2 columns]
```

	TotalCharges	MonthlyCharges
0	29.85	29.85
1	1889.50	56.95
2	108.15	53.85
3	1840.75	42.30
4	151.65	70.70
...	...	...
7038	1990.50	84.80
7039	7362.90	103.20
7040	346.45	29.60
7041	306.60	74.40

```
pd.concat([churnData['customerID'][1:7040],
           churnData[['TotalCharges', 'MonthlyCharges']]],
          axis = 1)
```

```
      customerID  TotalCharges  MonthlyCharges
1     5575-GNVDE       1889.50           56.95
2     3668-QPYBK        108.15           53.85
3     7795-CFOCW       1840.75           42.30
4     9237-HQITU        151.65           70.70
5     9305-CDSKC        820.50           99.65
...          ...           ...             ...
7039  2234-XADUH       7362.90          103.20
0            NaN         29.85           29.85
7040         NaN        346.45           29.60
7041         NaN        306.60           74.40
7042         NaN       6844.50          105.65

[7043 rows x 3 columns]
```

	customerID	TotalCharges	MonthlyCharges
1	5575-GNVDE	1889.50	56.95
2	3668-QPYBK	108.15	53.85
3	7795-CFOCW	1840.75	42.30

```
churnData.iloc[0:4,:]
```

```
   customerID  gender  SeniorCitizen Partner Dependents  tenure Phone

0  7590-VHVEG  Female              0     Yes         No       1

1  5575-GNVDE    Male              0      No         No      34

2  3668-QPYBK    Male              0      No         No       2

3  7795-CFOCW    Male              0      No         No      45


      MultipleLines InternetService OnlineSecurity OnlineBackup  \

0  No phone service             DSL             No          Yes

1                No             DSL            Yes           No

2                No             DSL            Yes          Yes

3  No phone service             DSL            Yes           No


  DeviceProtection TechSupport StreamingTV StreamingMovies        Cor

0               No          No          No              No  Month-to-

1              Yes          No          No              No        One

2               No          No          No              No  Month-to-

3              Yes         Yes          No              No        One


  PaperlessBilling              PaymentMethod  MonthlyCharges  TotalC

0              Yes           Electronic check           29.85
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetServi
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL

```
pd.concat([churnData.head().drop(columns='customerID'),churnData.tail()]
```

```
      gender  SeniorCitizen Partner Dependents  tenure PhoneService

0     Female              0     Yes         No       1           No

1       Male              0      No         No      34          Yes

2       Male              0      No         No       2          Yes
```

```
3       Male              0      No         No      45           No  \
4     Female              0      No         No       2          Yes
7038    Male              0     Yes        Yes      24          Yes
7039  Female              0     Yes        Yes      72          Yes
7040  Female              0     Yes        Yes      11           No
7041    Male              1     Yes         No       4          Yes
7042    Male              0      No         No      66          Yes

         MultipleLines InternetService OnlineSecurity OnlineBackup  \
0     No phone service             DSL             No          Yes
1                   No             DSL            Yes           No
2                   No             DSL            Yes          Yes
3     No phone service             DSL            Yes           No
4                   No     Fiber optic             No           No
7038               Yes             DSL            Yes           No
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	Online
0	Female	0	Yes	No	1	No	No phone service	DSL	No
1	Male	0	No	No	34	Yes	No	DSL	Yes
2	Male	0	No	No	2	Yes	No	DSL	Yes
3	Male	0	No	No	45	No	No phone service	DSL	Yes
4	Female	0	No	No	2	Yes	No	Fiber optic	No
7038	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No
7040	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes
7041	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No
7042	Male	0	No	No	66	Yes	No	Fiber optic	Yes

30. Join (merge)

```
import pandas as pd
flights = pd.read_csv('/content/drive/MyDrive/AIS_DG/Flight_flights.csv'
airlines = pd.read_csv('/content/drive/MyDrive/AIS_DG/Flight_airlines.cs
```

```
flights
```

	year	month	day	dep_time	sched_dep_time	dep_delay	arr_ti
1	2013	1	1	517.0	515	2.0	830
2	2013	1	1	533.0	529	4.0	850
3	2013	1	1	542.0	540	2.0	923
4	2013	1	1	544.0	545	-1.0	1004
5	2013	1	1	554.0	600	-6.0	812
...	...	...	...	...	...	...	.
336772	2013	9	30	NaN	1455	NaN	N
336773	2013	9	30	NaN	2200	NaN	N
336774	2013	9	30	NaN	1210	NaN	N
336775	2013	9	30	NaN	1159	NaN	N
336776	2013	9	30	NaN	840	NaN	N

	sched_arr_time	arr_delay	carrier	flight	tailnum	origin	dest
1	819	11.0	UA	1545	N14228	EWR	IAH
2	830	20.0	UA	1714	N24211	LGA	IAH
3	850	33.0	AA	1141	N619AA	JFK	MIA
4	1022	-18.0	B6	725	N804JB	JFK	BQN
5	837	-25.0	DL	461	N668DN	LGA	ATL

	year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	carr
1	2013	1	1	517.0	515	2.0	830.0	819	11.0	UA
2	2013	1	1	533.0	529	4.0	850.0	830	20.0	UA
3	2013	1	1	542.0	540	2.0	923.0	850	33.0	AA
4	2013	1	1	544.0	545	-1.0	1004.0	1022	-18.0	B6

```
airlines
```

```
   carrier                         name
1       9E           Endeavor Air Inc.
2       AA      American Airlines Inc.
3       AS        Alaska Airlines Inc.
4       B6             JetBlue Airways
5       DL        Delta Air Lines Inc.
6       EV    ExpressJet Airlines Inc.
7       F9      Frontier Airlines Inc.
8       FL  AirTran Airways Corporation
9       HA      Hawaiian Airlines Inc.
10      MQ                  Envoy Air
11      OO       SkyWest Airlines Inc.
12      UA       United Air Lines Inc.
13      US             US Airways Inc.
14      VX              Virgin America
15      WN      Southwest Airlines Co.
16      YV          Mesa Airlines Inc.
```

	carrier	name
1	9E	Endeavor Air Inc.
2	AA	American Airlines Inc.
3	AS	Alaska Airlines Inc.
4	B6	JetBlue Airways

```
flights[['time_hour','carrier','flight']]
```

```
                time_hour carrier  flight
1        2013-01-01 05:00:00      UA    1545
2        2013-01-01 05:00:00      UA    1714
3        2013-01-01 05:00:00      AA    1141
4        2013-01-01 05:00:00      B6     725
5        2013-01-01 06:00:00      DL     461
...                      ...     ...     ...
336772   2013-09-30 14:00:00      9E    3393
336773   2013-09-30 22:00:00      9E    3525
336774   2013-09-30 12:00:00      MQ    3461
336775   2013-09-30 11:00:00      MQ    3572
336776   2013-09-30 08:00:00      MQ    3531

[336776 rows x 3 columns]
```

	time_hour	carrier	flight
1	2013-01-01 05:00:00	UA	1545
2	2013-01-01 05:00:00	UA	1714
3	2013-01-01 05:00:00	AA	1141
4	2013-01-01 05:00:00	B6	725
5	2013-01-01 06:00:00	DL	461
...	...	...	...
336772	2013-09-30 14:00:00	9E	3393
336773	2013-09-30 22:00:00	9E	3525
336774	2013-09-30 12:00:00	MQ	3461
336775	2013-09-30 11:00:00	MQ	3572
336776	2013-09-30 08:00:00	MQ	3531

336776 rows × 3 columns

```
flightData = pd.merge(flights[['time_hour','carrier','flight']],
                      airlines, on='carrier', how='left')
```

```
flightData.head()
```

```
             time_hour carrier  flight                     name
0  2013-01-01 05:00:00      UA    1545   United Air Lines Inc.
1  2013-01-01 05:00:00      UA    1714   United Air Lines Inc.
2  2013-01-01 05:00:00      AA    1141  American Airlines Inc.
3  2013-01-01 05:00:00      B6     725         JetBlue Airways
4  2013-01-01 06:00:00      DL     461    Delta Air Lines Inc.
```

	time_hour	carrier	flight	name
0	2013-01-01 05:00:00	UA	1545	United Air Lines Inc.
1	2013-01-01 05:00:00	UA	1714	United Air Lines Inc.
2	2013-01-01 05:00:00	AA	1141	American Airlines Inc.
3	2013-01-01 05:00:00	B6	725	JetBlue Airways
4	2013-01-01 06:00:00	DL	461	Delta Air Lines Inc.

31. groupby + agg

```
flights['dep_delay'].count()
```

```
328521
```

```
flights[['carrier','dep_delay']][0:5]
```

```
  carrier  dep_delay
1      UA        2.0
2      UA        4.0
3      AA        2.0
4      B6       -1.0
5      DL       -6.0
```

	carrier	dep_delay
1	UA	2.0

```
flights.count()
```

```
year              336776
month             336776
day               336776
dep_time          328521
sched_dep_time    336776
dep_delay         328521
arr_time          328063
sched_arr_time    336776
arr_delay         327346
carrier           336776
flight            336776
tailnum           334264
origin            336776
dest              336776
air_time          327346
distance          336776
hour              336776
minute            336776
time_hour         336776
dtype: int64
```

```
import numpy as np
flights.groupby('carrier').agg({'dep_delay':['count',np.max],
                                'arr_delay':[np.mean,np.max]})
```

```
        dep_delay          arr_delay
            count    amax       mean    amax
carrier
9E          17416   747.0   7.379669   744.0
AA          32093  1014.0   0.364291  1007.0
AS            712   225.0  -9.930889   198.0
B6          54169   502.0   9.457973   497.0
DL          47761   960.0   1.644341   931.0
EV          51356   548.0  15.796431   577.0
F9            682   853.0  21.920705   834.0
FL           3187   602.0  20.115906   572.0
HA            342  1301.0  -6.915205  1272.0
MQ          25163  1137.0  10.774733  1127.0
OO             29   154.0  11.931034   157.0
UA          57979   483.0   3.558011   455.0
US          19873   500.0   2.129595   492.0
VX           5131   653.0   1.764464   676.0
WN          12083   471.0   9.649120   453.0
YV            545   387.0  15.556985   381.0
```

	dep_delay		arr_delay	
	count	amax	mean	amax
carrier				
9E	17416	747.0	7.379669	744.0
AA	32093	1014.0	0.364291	1007.0
AS	712	225.0	-9.930889	198.0
B6	54169	502.0	9.457973	497.0
DL	47761	960.0	1.644341	931.0
EV	51356	548.0	15.796431	577.0
F9	682	853.0	21.920705	834.0
FL	3187	602.0	20.115906	572.0
HA	342	1301.0	-6.915205	1272.0
MQ	25163	1137.0	10.774733	1127.0
OO	29	154.0	11.931034	157.0
UA	57979	483.0	3.558011	455.0
US	19873	500.0	2.129595	492.0
VX	5131	653.0	1.764464	676.0
WN	12083	471.0	9.649120	453.0
YV	545	387.0	15.556985	381.0

```
flight_groups = flights.groupby('carrier')
```

```
for c,x in flight_groups:
    print(c,x.count())
```

```
9E year              18460
month               18460
day                 18460
dep_time            17416
sched_dep_time      18460
dep_delay           17416
arr_time            17345
sched_arr_time      18460
arr_delay           17294
carrier             18460
flight              18460
tailnum             17416
origin              18460
dest                18460
```

```
air_time          17294
distance          18460
hour              18460
minute            18460
time_hour         18460
dtype: int64
```

```
flight_groups.apply(lambda x: x.shape)
```

```
carrier
9E    (18460, 19)
AA    (32729, 19)
AS      (714, 19)
B6    (54635, 19)
DL    (48110, 19)
EV    (54173, 19)
F9      (685, 19)
FL     (3260, 19)
HA      (342, 19)
MQ    (26397, 19)
OO       (32, 19)
UA    (58665, 19)
US    (20536, 19)
VX     (5162, 19)
WN    (12275, 19)
YV      (601, 19)
dtype: object
```

```
flight_groups.apply(lambda x: x.sample(3))
```

		year	month	day	dep_time	sched_dep_time	dep_delay
carrier							
9E	2884	2013	1	4	830.0	835	-5.0
	165973	2013	4	1	2053.0	2015	38.0
	216111	2013	5	25	952.0	955	-3.0
AA	302953	2013	8	25	911.0	915	-4.0
	102927	2013	12	22	1225.0	1225	0.0
	72163	2013	11	18	1650.0	1700	-10.0
AS	15655	2013	1	18	1811.0	1815	-4.0
	77848	2013	11	24	1821.0	1825	-4.0
	230207	2013	6	9	1814.0	1825	-11.0
B6	268381	2013	7	19	2333.0	2245	48.0
	31756	2013	10	6	819.0	820	-1.0
	43111	2013	10	18	800.0	805	-5.0
DL	149220	2013	3	14	2011.0	2015	-4.0
	96625	2013	12	15	1717.0	1710	7.0
	335419	2013	9	29	1554.0	1600	-6.0
EV	107347	2013	12	27	1343.0	1221	82.0
	65700	2013	11	11	1724.0	1630	54.0

		year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_de
carrier										
9E	2884	2013	1	4	830.0	835	-5.0	1029.0	1028	1.0
	165973	2013	4	1	2053.0	2015	38.0	2313.0	2239	34.0
	216111	2013	5	25	952.0	955	-3.0	1212.0	1212	0.0
AA	302953	2013	8	25	911.0	915	-4.0	1221.0	1210	11.0
	102927	2013	12	22	1225.0	1225	0.0	1525.0	1535	-10.0
	72163	2013	11	18	1650.0	1700	-10.0	1959.0	2000	-1.0
AS	15655	2013	1	18	1811.0	1815	-4.0	2155.0	2130	25.0
	77848	2013	11	24	1821.0	1825	-4.0	2132.0	2149	-17.0
	230207	2013	6	9	1814.0	1825	-11.0	2118.0	2145	-27.0
B6	268381	2013	7	19	2333.0	2245	48.0	110.0	3	67.0
	31756	2013	10	6	819.0	820	-1.0	1040.0	1052	-12.0
	43111	2013	10	18	800.0	805	-5.0	907.0	926	-19.0
DL	149220	2013	3	14	2011.0	2015	-4.0	2238.0	2318	-40.0
	96625	2013	12	15	1717.0	1710	7.0	2204.0	2021	NaN
	335419	2013	9	29	1554.0	1600	-6.0	1825.0	1835	-10.0
EV	107347	2013	12	27	1343.0	1221	82.0	1612.0	1454	78.0
	65700	2013	11	11	1724.0	1630	54.0	1856.0	1808	48.0

	8366	2013	1	10	1356.0	1359	-3.0	1516.0	1517	-1.0
F9	117569	2013	2	8	853.0	830	23.0	1145.0	1106	39.0
	240759	2013	6	20	1734.0	1730	4.0	1950.0	1956	-6.0
	58231	2013	11	3	1745.0	1729	16.0	2023.0	1951	32.0
FL	107257	2013	12	27	1151.0	1156	-5.0	1421.0	1424	-3.0
	154364	2013	3	20	1318.0	1314	4.0	1542.0	1535	7.0
	147283	2013	3	12	2207.0	2030	97.0	2341.0	2204	97.0
HA	300282	2013	8	22	953.0	1000	-7.0	1514.0	1440	34.0
	266680	2013	7	18	953.0	1000	-7.0	1421.0	1430	-9.0
	23578	2013	1	28	856.0	900	-4.0	1513.0	1530	-17.0
MQ	60550	2013	11	6	750.0	750	0.0	1045.0	1040	5.0
	138391	2013	3	3	1401.0	1410	-9.0	1519.0	1555	-36.0
	308366	2013	8	30	1906.0	1900	6.0	2038.0	2040	-2.0
OO	317066									

32. Pivot

```
flights
```

```
        year  month  day  dep_time  sched_dep_time  dep_delay  arr_ti
1       2013      1    1     517.0             515        2.0     830
2       2013      1    1     533.0             529        4.0     850
```

3	2013	1	1	542.0	540	2.0	923	
4	2013	1	1	544.0	545	-1.0	1004	
5	2013	1	1	554.0	600	-6.0	812	
...	...	...	...	...	...	...	.	
336772	2013	9	30	NaN	1455	NaN	N	
336773	2013	9	30	NaN	2200	NaN	N	
336774	2013	9	30	NaN	1210	NaN	N	
336775	2013	9	30	NaN	1159	NaN	N	
336776	2013	9	30	NaN	840	NaN	N	

	sched_arr_time	arr_delay	carrier	flight	tailnum	origin	dest
1	819	11.0	UA	1545	N14228	EWR	IAH
2	830	20.0	UA	1714	N24211	LGA	IAH
3	850	33.0	AA	1141	N619AA	JFK	MIA
4	1022	-18.0	B6	725	N804JB	JFK	BQN
5	837	-25.0	DL	461	N668DN	LGA	ATL

```
flights.groupby(['carrier','month'])\
       .dep_delay.mean()\
       .reset_index()
```

```
    carrier  month  dep_delay
0        9E      1  16.882510
1        9E      2  16.486327
2        9E      3  13.407530
3        9E      4  13.567164
4        9E      5  22.672190
..      ...    ...        ...
180      YV      8  19.066667
181      YV      9   8.880952
182      YV     10  20.000000
183      YV     11  10.520833
184      YV     12  13.113636

[185 rows x 3 columns]
```

	carrier	month	dep_delay
0	9E	1	16.882510
1	9E	2	16.486327
2	9E	3	13.407530
3	9E	4	13.567164
4	9E	5	22.672190
...	...	...	...
180	YV	8	19.066667
181	YV	9	8.880952
182	YV	10	20.000000
183	YV	11	10.520833
184	YV	12	13.113636

185 rows × 3 columns

```
flights.groupby(['carrier','month'])\
       .dep_delay.mean()\
       .reset_index()\
       .pivot(index='carrier',columns='month',values='dep_delay')
```

```
month           1          2          3          4          5          
carrier
9E       16.882510  16.486327  13.407530  13.567164  22.672190  28.95
```

```
AA        6.932358   8.276923   8.700291  11.696207   9.664621  14.62
AS        7.354839   0.722222   8.419355  11.316667   6.774194  13.08
B6        9.493436  13.772911  14.240690  15.165175   9.778560  20.39
DL        3.849768   5.537440   9.933430   8.166544   9.741168  18.73
EV       24.228879  21.523328  26.169820  22.767549  20.242477  25.49
F9       10.000000  29.770833  16.754386  24.631579  35.948276  29.43
FL        1.972222   5.180851  17.252459  13.121311  19.183230  38.80
HA       54.387097  17.357143   1.161290  -2.100000  -1.451613   1.46
MQ        6.485494   8.092962   7.193262  13.738095  13.925859  20.84
OO       67.000000        NaN        NaN        NaN        NaN  61.00
UA        8.326167   7.711234  11.689606  13.651643  12.259470  20.26
US        1.817363   0.980164   2.722694   4.062574   3.184888  10.40
VX        1.063492   6.609195   9.676568  14.643777  17.830645  28.41
WN        9.137056  11.751452  15.246073  18.223602  16.939271  30.51
YV       15.846154  10.673913  31.888889  27.111111  15.409091  42.79
```

month	1	2	3	4	5	6	7	8	9
carrier									
9E	16.882510	16.486327	13.407530	13.567164	22.672190	28.952978	31.398827	17.296807	7.754232
AA	6.932358	8.276923	8.700291	11.696207	9.664621	14.627778	12.112621	7.169965	5.694272
AS	7.354839	0.722222	8.419355	11.316667	6.774194	13.083333	2.419355	2.870968	-4.516667
B6	9.493436	13.772911	14.240690	15.165175	9.778560	20.392170	24.902315	15.678593	6.634260
DL	3.849768	5.537440	9.933430	8.166544	9.741168	18.735941	20.582242	9.846974	5.526071
EV	24.228879	21.523328	26.169820	22.767549	20.242477	25.496834	26.504722	16.261828	8.237970
F9	10.000000	29.770833	16.754386	24.631579	35.948276	29.436364	31.810345	22.218182	8.263158
FL	1.972222	5.180851	17.252459	13.121311	19.183230	38.806584	41.162698	23.410156	16.948819
HA	54.387097	17.357143	1.161290	-2.100000	-1.451613	1.466667	-1.709677	1.677419	-5.440000
MQ	6.485494	8.092962	7.193262	13.738095	13.925859	20.842342	20.745310	10.050277	5.350545
OO	67.000000	NaN	NaN	NaN	NaN	61.000000	NaN	64.000000	-4.941176
UA	8.326167	7.711234	11.689606	13.651643	12.259470	20.265377	20.105200	12.424403	6.890823
US	1.817363	0.980164	2.722694	4.062574	3.184888	10.404834	9.443335	4.960942	1.962583
VX	1.063492	6.609195	9.676568	14.643777	17.830645	28.412500	35.263374	6.607362	6.988962
WN	9.137056	11.751452	15.246073	18.223602	16.939271	30.514735	24.628169	21.787966	14.166832
YV	15.846154	10.673913	31.888889	27.111111	15.409091	42.794872	22.434783	19.066667	8.880952

```
pd.pivot_table(data=flights,
               values='dep_delay',
               index='carrier',
               columns='month',
               aggfunc='mean')
```

month	1	2	3	4	5	
carrier						
9E	16.882510	16.486327	13.407530	13.567164	22.672190	28.95
AA	6.932358	8.276923	8.700291	11.696207	9.664621	14.62
AS	7.354839	0.722222	8.419355	11.316667	6.774194	13.08
B6	9.493436	13.772911	14.240690	15.165175	9.778560	20.39
DL	3.849768	5.537440	9.933430	8.166544	9.741168	18.73
EV	24.228879	21.523328	26.169820	22.767549	20.242477	25.49
F9	10.000000	29.770833	16.754386	24.631579	35.948276	29.43
FL	1.972222	5.180851	17.252459	13.121311	19.183230	38.80
HA	54.387097	17.357143	1.161290	-2.100000	-1.451613	1.46
MQ	6.485494	8.092962	7.193262	13.738095	13.925859	20.84
OO	67.000000	NaN	NaN	NaN	NaN	61.00
UA	8.326167	7.711234	11.689606	13.651643	12.259470	20.26
US	1.817363	0.980164	2.722694	4.062574	3.184888	10.40
VX	1.063492	6.609195	9.676568	14.643777	17.830645	28.41
WN	9.137056	11.751452	15.246073	18.223602	16.939271	30.51
YV	15.846154	10.673913	31.888889	27.111111	15.409091	42.79

month	7	8	9	10	11	

month	1	2	3	4	5	6	7	8	9
carrier									

33. Melt

```
expenditure = pd.read_excel('/content/drive/MyDrive/AIS_DG/Gov_Expenditu
```

```
expenditure
```

```
         Country Name Country Code  \
0               Aruba          ABW
1             Andorra          AND
2         Afghanistan          AFG
3              Angola          AGO
4             Albania          ALB
..                ...          ...
244       Yemen, Rep.          YEM
245      South Africa          ZAF
246  Congo, Dem. Rep.          COD
247            Zambia          ZMB
248          Zimbabwe          ZWE

                                        Indicator Name     Indicator
0    Government expenditure on education, total (% ...  SE.XPD.TOTL.G
1    Government expenditure on education, total (% ...  SE.XPD.TOTL.G
2    Government expenditure on education, total (% ...  SE.XPD.TOTL.G
3    Government expenditure on education, total (% ...  SE.XPD.TOTL.G
4    Government expenditure on education, total (% ...  SE.XPD.TOTL.G
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966
0	Aruba	ABW	Government expenditure on education, total (% ...	SE.XPD.TOTL.GB.ZS	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Andorra	AND	Government expenditure on education, total (% ...	SE.XPD.TOTL.GB.ZS	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Afghanistan	AFG	Government expenditure on education, total (% ...	SE.XPD.TOTL.GB.ZS	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Angola	AGO	Government expenditure on education, total (% ...	SE.XPD.TOTL.GB.ZS	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Albania	ALB	Government expenditure on education, total (% ...	SE.XPD.TOTL.GB.ZS	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
244	Yemen, Rep.	YEM	Government expenditure on education, total (% ...	SE.XPD.TOTL.GB.ZS	NaN	NaN	NaN	NaN	NaN	NaN	NaN
245	South Africa	ZAF	Government expenditure on education, total (% ...	SE.XPD.TOTL.GB.ZS	NaN	NaN	NaN	NaN	NaN	NaN	NaN
			Government								

```
expenditure['Indicator Name'][0]
```

```
'Government expenditure on education, total (% of government expenditu
```

```
expenditure.melt(id_vars=['Country Name','Country Code'],
                 value_vars=np.arange(1960,2015,1).astype('str'),
                 var_name='Year',
                 value_name='Expenditure')
```

```
           Country Name Country Code  Year  Expenditure
0                 Aruba          ABW  1960          NaN
1               Andorra          AND  1960          NaN
2           Afghanistan          AFG  1960          NaN
3                Angola          AGO  1960          NaN
4               Albania          ALB  1960          NaN
...                 ...          ...   ...          ...
13690       Yemen, Rep.          YEM  2014          NaN
13691      South Africa          ZAF  2014          NaN
13692  Congo, Dem. Rep.          COD  2014          NaN
13693            Zambia          ZMB  2014          NaN
13694          Zimbabwe          ZWE  2014          NaN

[13695 rows x 4 columns]
```

	Country Name	Country Code	Year	Expenditure
0	Aruba	ABW	1960	NaN
1	Andorra	AND	1960	NaN
2	Afghanistan	AFG	1960	NaN
3	Angola	AGO	1960	NaN
4	Albania	ALB	1960	NaN
...	...	...	...	...
13690	Yemen, Rep.	YEM	2014	NaN
13691	South Africa	ZAF	2014	NaN
13692	Congo, Dem. Rep.	COD	2014	NaN
13693	Zambia	ZMB	2014	NaN
13694	Zimbabwe	ZWE	2014	NaN

13695 rows × 4 columns

```
expenditureByCountry = expenditure.drop(columns=['Indicator Name','Indic
                                .melt(id_vars='Country Name',
                                      value_vars=np.arange(1960,2015,1
                                      var_name='Year',
                                      value_name='Amount')\
                                .sort_values(by=['Country Name','Year'
expenditureByCountry
```

```
      Country Name  Year   Amount
2      Afghanistan  1960      NaN
251    Afghanistan  1961      NaN
500    Afghanistan  1962      NaN
749    Afghanistan  1963      NaN
998    Afghanistan  1964      NaN
```

```
...              ...   ...       ...
12698      Zimbabwe  2010  8.72091
12947      Zimbabwe  2011       NaN
13196      Zimbabwe  2012       NaN
13445      Zimbabwe  2013       NaN
13694      Zimbabwe  2014       NaN

[13695 rows x 3 columns]
```

	Country Name	Year	Amount
2	Afghanistan	1960	NaN
251	Afghanistan	1961	NaN
500	Afghanistan	1962	NaN
749	Afghanistan	1963	NaN
998	Afghanistan	1964	NaN
...	...	...	...
12698	Zimbabwe	2010	8.72091
12947	Zimbabwe	2011	NaN
13196	Zimbabwe	2012	NaN
13445	Zimbabwe	2013	NaN
13694	Zimbabwe	2014	NaN

13695 rows × 3 columns

```
expenditureByCountry[expenditureByCountry['Country Name'].isin(['Thailan
```

```
      Country Name  Year    Amount
219       Thailand  1960       NaN
468       Thailand  1961       NaN
717       Thailand  1962       NaN
966       Thailand  1963       NaN
1215      Thailand  1964       NaN
1464      Thailand  1965       NaN
1713      Thailand  1966       NaN
1962      Thailand  1967       NaN
2211      Thailand  1968       NaN
2460      Thailand  1969       NaN
2709      Thailand  1970       NaN
2958      Thailand  1971       NaN
3207      Thailand  1972       NaN
3456      Thailand  1973       NaN
3705      Thailand  1974       NaN
3954      Thailand  1975       NaN
4203      Thailand  1976       NaN
4452      Thailand  1977       NaN
```

```
      4701      Thailand  1978       NaN
```

	Country Name	Year	Amount
219	Thailand	1960	NaN
468	Thailand	1961	NaN
717	Thailand	1962	NaN
966	Thailand	1963	NaN
1215	Thailand	1964	NaN
1464	Thailand	1965	NaN
1713	Thailand	1966	NaN
1962	Thailand	1967	NaN
2211	Thailand	1968	NaN
2460	Thailand	1969	NaN
2709	Thailand	1970	NaN
2958	Thailand	1971	NaN
3207	Thailand	1972	NaN
3456	Thailand	1973	NaN
3705	Thailand	1974	NaN
3954	Thailand	1975	NaN
4203	Thailand	1976	NaN
4452	Thailand	1977	NaN
4701	Thailand	1978	NaN
4950	Thailand	1979	NaN
5199	Thailand	1980	NaN
5448	Thailand	1981	NaN

34. Time series - set time as index

```
flightData
```

```
                 time_hour carrier  flight                    name
0       2013-01-01 05:00:00      UA    1545   United Air Lines Inc.
1       2013-01-01 05:00:00      UA    1714   United Air Lines Inc.
2       2013-01-01 05:00:00      AA    1141  American Airlines Inc.
3       2013-01-01 05:00:00      B6     725         JetBlue Airways
4       2013-01-01 06:00:00      DL     461    Delta Air Lines Inc.
```

```
...                       ...     ...     ...                      ...
336771  2013-09-30 14:00:00      9E    3393       Endeavor Air Inc.
336772  2013-09-30 22:00:00      9E    3525       Endeavor Air Inc.
336773  2013-09-30 12:00:00      MQ    3461              Envoy Air
336774  2013-09-30 11:00:00      MQ    3572              Envoy Air
336775  2013-09-30 08:00:00      MQ    3531              Envoy Air

[336776 rows x 4 columns]
```

	time_hour	carrier	flight	name
0	2013-01-01 05:00:00	UA	1545	United Air Lines Inc.
1	2013-01-01 05:00:00	UA	1714	United Air Lines Inc.
2	2013-01-01 05:00:00	AA	1141	American Airlines Inc.
3	2013-01-01 05:00:00	B6	725	JetBlue Airways
4	2013-01-01 06:00:00	DL	461	Delta Air Lines Inc.
...	...	...	...	...
336771	2013-09-30 14:00:00	9E	3393	Endeavor Air Inc.
336772	2013-09-30 22:00:00	9E	3525	Endeavor Air Inc.
336773	2013-09-30 12:00:00	MQ	3461	Envoy Air
336774	2013-09-30 11:00:00	MQ	3572	Envoy Air
336775	2013-09-30 08:00:00	MQ	3531	Envoy Air

336776 rows × 4 columns

```
flightData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 336776 entries, 0 to 336775
Data columns (total 4 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   time_hour  336776 non-null  object
 1   carrier    336776 non-null  object
 2   flight     336776 non-null  int64 
 3   name       336776 non-null  object
dtypes: int64(1), object(3)
memory usage: 12.8+ MB
```

```
flightData['time_hour'] = pd.to_datetime(flightData['time_hour'])
```

```
flightData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 336776 entries, 0 to 336775
Data columns (total 4 columns):
 #   Column     Non-Null Count   Dtype         
---  ------     --------------   -----         
 0   time_hour  336776 non-null  datetime64[ns]
 1   carrier    336776 non-null  object        
 2   flight     336776 non-null  int64         
 3   name       336776 non-null  object        
dtypes: datetime64[ns](1), int64(1), object(2)
memory usage: 12.8+ MB
```

```
flightData
```

```
                 time_hour carrier  flight                     name
0      2013-01-01 05:00:00      UA    1545   United Air Lines Inc.
1      2013-01-01 05:00:00      UA    1714   United Air Lines Inc.
2      2013-01-01 05:00:00      AA    1141  American Airlines Inc.
3      2013-01-01 05:00:00      B6     725         JetBlue Airways
4      2013-01-01 06:00:00      DL     461    Delta Air Lines Inc.
...                    ...     ...     ...                     ...
336771 2013-09-30 14:00:00      9E    3393       Endeavor Air Inc.
336772 2013-09-30 22:00:00      9E    3525       Endeavor Air Inc.
336773 2013-09-30 12:00:00      MQ    3461               Envoy Air
336774 2013-09-30 11:00:00      MQ    3572               Envoy Air
336775 2013-09-30 08:00:00      MQ    3531               Envoy Air

[336776 rows x 4 columns]
```

	time_hour	carrier	flight	name
0	2013-01-01 05:00:00	UA	1545	United Air Lines Inc.
1	2013-01-01 05:00:00	UA	1714	United Air Lines Inc.
2	2013-01-01 05:00:00	AA	1141	American Airlines Inc.
3	2013-01-01 05:00:00	B6	725	JetBlue Airways
4	2013-01-01 06:00:00	DL	461	Delta Air Lines Inc.
...	...	...	...	...
336771	2013-09-30 14:00:00	9E	3393	Endeavor Air Inc.
336772	2013-09-30 22:00:00	9E	3525	Endeavor Air Inc.
336773	2013-09-30 12:00:00	MQ	3461	Envoy Air
336774	2013-09-30 11:00:00	MQ	3572	Envoy Air
336775	2013-09-30 08:00:00	MQ	3531	Envoy Air

336776 rows × 4 columns

```
flightData.set_index('time_hour',inplace=True)
```

```
flightData.head()
```

```
                     carrier  flight                    name
time_hour
2013-01-01 05:00:00      UA    1545   United Air Lines Inc.
2013-01-01 05:00:00      UA    1714   United Air Lines Inc.
2013-01-01 05:00:00      AA    1141  American Airlines Inc.
2013-01-01 05:00:00      B6     725         JetBlue Airways
2013-01-01 06:00:00      DL     461    Delta Air Lines Inc.
```

	carrier	flight	name
time_hour			
2013-01-01 05:00:00	UA	1545	United Air Lines Inc.
2013-01-01 05:00:00	UA	1714	United Air Lines Inc.
2013-01-01 05:00:00	AA	1141	American Airlines Inc.
2013-01-01 05:00:00	B6	725	JetBlue Airways
2013-01-01 06:00:00	DL	461	Delta Air Lines Inc.

35. Time series - aggregate

```
fpw = flightData.resample('W').carrier.count()
fpw
```

```
time_hour
2013-01-06    5166
2013-01-13    6114
2013-01-20    6034
2013-01-27    6049
2013-02-03    6063
2013-02-10    6104
2013-02-17    6236
2013-02-24    6381
2013-03-03    6444
2013-03-10    6546
2013-03-17    6555
2013-03-24    6547
2013-03-31    6550
2013-04-07    6592
2013-04-14    6613
2013-04-21    6622
```

```
2013-04-28    6560
2013-05-05    6525
2013-05-12    6493
```

```
fpw.rolling(2).mean()
```

```
time_hour
2013-01-06       NaN
2013-01-13    5640.0
2013-01-20    6074.0
2013-01-27    6041.5
2013-02-03    6056.0
2013-02-10    6083.5
2013-02-17    6170.0
2013-02-24    6308.5
2013-03-03    6412.5
2013-03-10    6495.0
2013-03-17    6550.5
2013-03-24    6551.0
2013-03-31    6548.5
2013-04-07    6571.0
2013-04-14    6602.5
2013-04-21    6617.5
2013-04-28    6591.0
2013-05-05    6542.5
2013-05-12    6509.0
```

36. Dummies variables

```
churnData = pd.read_csv('/content/drive/MyDrive/AIS_DG/Telco-Churn.csv',
churnData['Contract'].head()
```

```
0    Month-to-month
1          One year
2    Month-to-month
3          One year
4    Month-to-month
Name: Contract, dtype: object
```

```
churnData['Contract'].unique()
```

```
array(['Month-to-month', 'One year', 'Two year'], dtype=object)
```

```
pd.get_dummies(churnData['Contract'],prefix='Contract')
```

```
      Contract_Month-to-month  Contract_One year  Contract_Two year
0                           1                  0                  0
1                           0                  1                  0
2                           1                  0                  0
3                           0                  1                  0
4                           1                  0                  0
...                       ...                ...                ...
7038                        0                  1                  0
7039                        0                  1                  0
7040                        1                  0                  0
7041                        1                  0                  0
7042                        0                  0                  1

[7043 rows x 3 columns]
```

	Contract_Month-to-month	Contract_One year	Contract_Two year
0	1	0	0
1	0	1	0
2	1	0	0
3	0	1	0
4	1	0	0
...	...	...	...
7038	0	1	0
7039	0	1	0
7040	1	0	0
7041	1	0	0
7042	0	0	1

7043 rows × 3 columns

```
pd.get_dummies(churnData['Contract'],
               prefix='Contract',
               drop_first=True)
```

```
      Contract_One year  Contract_Two year
0                     0                  0
1                     1                  0
2                     0                  0
3                     1                  0
4                     0                  0
...                 ...                ...
```

```
7038                    1                    0
7039                    1                    0
7040                    0                    0
7041                    0                    0
7042                    0                    1

[7043 rows x 2 columns]
```

	Contract_One year	Contract_Two year
0	0	0
1	1	0
2	0	0
3	1	0
4	0	0
...	...	...
7038	1	0
7039	1	0
7040	0	0
7041	0	0
7042	0	1

7043 rows × 2 columns

```
pd.get_dummies(churnData[['Churn','Contract']])
```

```
      Churn_No  Churn_Yes  Contract_Month-to-month  Contract_One year
0            1          0                        1                  0
1            1          0                        0                  1
2            0          1                        1                  0
3            1          0                        0                  1
4            0          1                        1                  0
...        ...        ...                      ...                ...
7038         1          0                        0                  1
7039         1          0                        0                  1
7040         1          0                        1                  0
7041         0          1                        1                  0
7042         1          0                        0                  0

      Contract_Two year
0                     0
1                     0
2                     0
3                     0
4                     0
```

	Churn_No	Churn_Yes	Contract_Month-to-month	Contract_One year	Contract_Two year
0	1	0	1	0	0
1	1	0	0	1	0
2	0	1	1	0	0
3	1	0	0	1	0
4	0	1	1	0	0
...	...	...	...	...	...
7038	1	0	0	1	0
7039	1	0	0	1	0
7040	1	0	1	0	0
7041	0	1	1	0	0
7042	1	0	0	0	1

7043 rows × 5 columns

37. Sample data

```
# Simple random sampling
churnData.sample(frac=0.01)
```

```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
5494  8837-VVWLQ  Female              0      No         No       8
5995  2193-SFWQW    Male              0     Yes        Yes      72
4983  1862-SKORY  Female              1     Yes         No      40
5171  3566-VVORZ  Female              0     Yes         No      12
6684  0305-SQECB  Female              0      No        Yes      11
...          ...     ...            ...     ...        ...     ...
3675  9753-OYLBX  Female              0      No         No       1
2177  2878-RMWXY    Male              1     Yes         No      72
4496  9489-JMTTN  Female              0     Yes        Yes      72
6856  4534-WGCIR  Female              0     Yes        Yes      58
3262  5498-TXHLF  Female              0     Yes        Yes      34

     PhoneService     MultipleLines InternetService       OnlineSecur
5494          Yes                No     Fiber optic
5995          Yes                No     Fiber optic
4983           No  No phone service             DSL
5171          Yes                No             DSL
6684           No  No phone service             DSL
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
5494	8837-VVWLQ	Female	0	No	No	8	Yes	No	Fiber opt
5995	2193-SFWQW	Male	0	Yes	Yes	72	Yes	No	Fiber opt
4983	1862-SKORY	Female	1	Yes	No	40	No	No phone service	DSL
5171	3566-VVORZ	Female	0	Yes	No	12	Yes	No	DSL
6684	0305-SQECB	Female	0	No	Yes	11	No	No phone service	DSL
...	...	...	...	...	...	...	...	...	...
3675	9753-OYLBX	Female	0	No	No	1	Yes	No	No
2177	2878-RMWXY	Male	1	Yes	No	72	Yes	Yes	Fiber opt
4496	9489-JMTTN	Female	0	Yes	Yes	72	Yes	Yes	DSL
6856	4534-WGCIR	Female	0	Yes	Yes	58	Yes	Yes	No
3262	5498-TXHLF	Female	0	Yes	Yes	34	Yes	Yes	Fiber opt

70 rows × 21 columns

```
#Stratified sampling
churnData.groupby('Churn', group_keys=False).apply(lambda x: x.sample(n=
```

```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
5376  9229-RQABD    Male              0      No         No      18
3514  6993-YCOBK    Male              0     Yes        Yes      60
5783  1415-YFWLT  Female              1      No         No       1
3326  7657-DYEPJ    Male              1      No         No      38

     PhoneService MultipleLines InternetService       OnlineSecurity
5376          Yes           Yes              No  No internet service
3514          Yes           Yes     Fiber optic                  Yes
5783          Yes            No     Fiber optic                   No
3326          Yes            No             DSL                   No

            OnlineBackup     DeviceProtection          TechSupport
5376  No internet service  No internet service  No internet service
3514                   No                  Yes                  Yes
5783                   No                   No                   No
3326                  Yes                  Yes                  Yes
```

```
            StreamingTV      StreamingMovies        Contract  \
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
5376	9229-RQABD	Male	0	No	No	18	Yes	Yes	No
3514	6993-YCOBK	Male	0	Yes	Yes	60	Yes	Yes	Fiber opt
5783	1415-YFWLT	Female	1	No	No	1	Yes	No	Fiber opt
3326	7657-DYEPJ	Male	1	No	No	38	Yes	No	DSL

38. Variable binning

```
churnData['TotalCharges']
```

```
0         29.85
1       1889.50
2        108.15
3       1840.75
4        151.65
         ...
7038    1990.50
7039    7362.90
7040     346.45
7041     306.60
7042    6844.50
Name: TotalCharges, Length: 7043, dtype: float64
```

```
pd.cut(churnData['TotalCharges'],5).value_counts()
```

```
(10.134, 1752.0]    3937
(1752.0, 3485.2]    1179
(3485.2, 5218.4]     866
(5218.4, 6951.6]     705
(6951.6, 8684.8]     345
Name: TotalCharges, dtype: int64
```

```
pd.cut(churnData['TotalCharges'],bins=[0, 10, 50, 1000, 5000, 10000])
```

```
0            (10, 50]
1         (1000, 5000]
2           (50, 1000]
3         (1000, 5000]
4           (50, 1000]
             ...
7038      (1000, 5000]
7039     (5000, 10000]
7040        (50, 1000]
7041        (50, 1000]
7042     (5000, 10000]
Name: TotalCharges, Length: 7043, dtype: category
Categories (5, interval[int64, right]): [(0, 10] < (10, 50] < (50, 100
                                         (5000, 10000]]
```

```
pd.qcut(churnData['TotalCharges'],q=4).value_counts()
```

```
(18.799, 401.45]        1758
(401.45, 1397.475]      1758
(1397.475, 3794.738]    1758
(3794.738, 8684.8]      1758
Name: TotalCharges, dtype: int64
```

Exercise

- From flight data
  - Which flight numbers (carrier + flight) delayed the worst?
  - Obtain the 10 worst flight numbers of each month

```
# Work here
  flights.sort_values(by='dep_delay', ascending=False)
```

```
        year  month  day  dep_time  sched_dep_time  dep_delay  arr_ti
7073    2013      1    9     641.0             900     1301.0    1242
235779  2013      6   15    1432.0            1935     1137.0    1607
8240    2013      1   10    1121.0            1635     1126.0    1239
327044  2013      9   20    1139.0            1845     1014.0    1457
270377  2013      7   22     845.0            1600     1005.0    1044
...      ...    ...  ...       ...             ...        ...       .
336772  2013      9   30       NaN            1455        NaN       N
336773  2013      9   30       NaN            2200        NaN       N
336774  2013      9   30       NaN            1210        NaN       N
336775  2013      9   30       NaN            1159        NaN       N
336776  2013      9   30       NaN             840        NaN       N

        sched_arr_time  arr_delay carrier  flight tailnum origin dest
```

```
7073              1530     1272.0      HA       51  N384HA    JFK  HNL
235779            2120     1127.0      MQ     3535  N504MQ    JFK  CMH
8240              1810     1109.0      MQ     3695  N517MQ    EWR  ORD
327044            2210     1007.0      AA      177  N338AA    JFK  SFO
270377            1815      989.0      MQ     3075  N665MQ    JFK  CVG
```

	year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	carr
7073	2013	1	9	641.0	900	1301.0	1242.0	1530	1272.0	HA
235779	2013	6	15	1432.0	1935	1137.0	1607.0	2120	1127.0	MQ
8240	2013	1	10	1121.0	1635	1126.0	1239.0	1810	1109.0	MQ
327044	2013	9	20	1139.0	1845	1014.0	1457.0	2210	1007.0	AA
270377	2013	7	22	845.0	1600	1005.0	1044.0	1815	989.0	MQ
...	...	...	...	...	...	...	...	...	...	...
336772	2013	9	30	NaN	1455	NaN	NaN	1634	NaN	9E
336773	2013	9	30	NaN	2200	NaN	NaN	2312	NaN	9E
336774	2013	9	30	NaN	1210	NaN	NaN	1330	NaN	MQ
336775	2013	9	30	NaN	1159	NaN	NaN	1344	NaN	MQ
336776	2013	9	30	NaN	840	NaN	NaN	1020	NaN	MQ

336776 rows × 19 columns

```
flights\
  .groupby('month', as_index=False)\
  .apply(lambda x: x.sort_values(by='dep_delay').head(10))
```

```
          year  month  day  dep_time  sched_dep_time  dep_delay  arr
0  9620   2013      1   11    1900.0            1930      -30.0    2
   24916  2013      1   29    1703.0            1730      -27.0    1
   18194  2013      1   21    2137.0            2159      -22.0    2
```

```
   10124   2013      1   12    1354.0             1416      -22.0    1
   16582   2013      1   20     704.0              725      -21.0    1
...          ...    ...  ...       ...              ...        ...
11 92123   2013     12   10    1841.0             1900      -19.0    2
   109630  2013     12   30     657.0              715      -18.0
   88214   2013     12    6     811.0              829      -18.0    1
   89870   2013     12    8     853.0              910      -17.0    1
   89677   2013     12    7    2043.0             2100      -17.0    2

           sched_arr_time  arr_delay carrier  flight tailnum origin d
0  9620               2243      -10.0      DL    1435  N934DL    LGA
   24916              1957      -10.0      F9     837  N208FR    LGA
   18194              2316      -44.0      DL    2155  N377NW    LGA
   10124              1650      -44.0      FL     349  N929AT    LGA
   16582              1035      -10.0      AS      11  N556AS    EWR
```

		year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay
0	9620	2013	1	11	1900.0	1930	-30.0	2233.0	2243	-10.0
	24916	2013	1	29	1703.0	1730	-27.0	1947.0	1957	-10.0
	18194	2013	1	21	2137.0	2159	-22.0	2232.0	2316	-44.0
	10124	2013	1	12	1354.0	1416	-22.0	1606.0	1650	-44.0
	16582	2013	1	20	704.0	725	-21.0	1025.0	1035	-10.0
...	...	...	...	...	...	...	...	...	...	...
11	92123	2013	12	10	1841.0	1900	-19.0	2028.0	2047	-19.0
	109630	2013	12	30	657.0	715	-18.0	927.0	940	-13.0
	88214	2013	12	6	811.0	829	-18.0	1119.0	1055	24.0
	89870	2013	12	8	853.0	910	-17.0	1105.0	1147	-42.0
	89677	2013	12	7	2043.0	2100	-17.0	2250.0	2315	-25.0

120 rows × 19 columns