Pandas - III

🎯 Session Objectives:

- ✅ Understand and perform statistical operations
- ✅ Conduct frequency and aggregation analysis
- ✅ Learn about sorting and indexing
- ✅ Export DataFrames to CSV/Excel files

```
# TypeCasting .astype()
Customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, FirstCustomer to 999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              1000 non-null   float64
 1   first_name      1000 non-null   object
 2   last_name       1000 non-null   object
 3   email           878 non-null    object
 4   gender          957 non-null    object
 5   street_number   738 non-null    float64
 6   street_address  963 non-null    object
 7   street_suffix   963 non-null    object
 8   city            921 non-null    object
 9   state           920 non-null    object
 10  postcode        1000 non-null   int32
dtypes: float64(2), int32(1), object(8)
memory usage: 89.8+ KB
```

```
Customers['postcode'] = Customers['postcode'].fillna(0).astype(int)
Customers
```

ıe	last_name	email	gender	street_number	street_address	street_suffix	city	state	postcode
in	Southcott	rsouthcott0@clickbank.net	Male	1.0	Trailsway	Road	San Diego	California	92127
ıo	Molyneaux	cmolyneaux1@wiley.com	Male	NaN	NaN	NaN	El Paso	Texas	0
bi	Westrip	bwestrip2@symantec.com	Female	4057.0	Arkansas	Circle	San Antonio	Texas	78220
:e	Pankettman	rpankettman3@wiley.com	Male	74.0	Debs	Point	Memphis	Tennessee	38150
la	Althorpe	malthorpe4@51.la	NaN	NaN	2nd	Drive	NaN	NaN	83732
...	...	...	...	...	...	...	...	...	...
ili	Alman	malmanrn@cornell.edu	Female	0.0	Thompson	Way	Reading	Pennsylvania	19610
ıh	Heckle	whecklero@fc2.com	Female	701.0	Rowland	Hill	Indianapolis	Indiana	46278
ıit	Birt	NaN	Male	2.0	Basil	Road	Waterbury	Connecticut	6721
ıh	Standbrooke	istandbrookerq@yellowpages.com	Male	NaN	Kenwood	Drive	Savannah	Georgia	31405
're	Malpas	NaN	Male	205.0	Farwell	Park	Atlanta	Georgia	30386

```
# Cleaning Customers, Products & Purchases DataFrame
Customers['street_number'] = Customers['street_number'].astype('Int64')
Customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix
FirstCustomer	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road
SecondCustomer	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	<NA>	NaN	NaN
ThirdCustomer	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle
FourthCustomer	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point
FifthCustomer	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	<NA>	2nd	Drive
...	...	...	...	...	...	...	...	...
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill
997	998.0	Tobit	Birt	NaN	Male	2	Basil	Road
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	<NA>	Kenwood	Drive
999	1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park

1000 rows × 11 columns

```
# TypeCasting .astype()
Customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, FirstCustomer to 999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              1000 non-null   float64
 1   first_name      1000 non-null   object
 2   last_name       1000 non-null   object
 3   email           878 non-null    object
 4   gender          957 non-null    object
 5   street_number   738 non-null    Int64
 6   street_address  963 non-null    object
 7   street_suffix   963 non-null    object
 8   city            921 non-null    object
 9   state           920 non-null    object
 10  postcode        1000 non-null   int32
dtypes: Int64(1), float64(1), int32(1), object(8)
memory usage: 90.8+ KB
```

```
Purchases['paid'] = Purchases['paid'].astype('str').str.replace('[$,]' , '' , regex = True).astype(float)
Purchases
```

	Unnamed: 0	id	purch_date	customer_num	product_num	amount	paid
0	0	1	2019-01-03 00:00:00	823	27	12	568.92
1	1	2	2019-01-03 00:00:00	606	28	14	395.36
2	2	3	2019-01-03 00:00:00	955	9	17	510.17
3	3	4	2019-01-03 00:00:00	577	19	3	68.49
4	4	5	2019-01-03 00:00:00	429	8	18	759.42
...	...	...	...	...	...	...	...
5995	5995	5996	06/20/2019	893	33	5	411.10
5996	5996	5997	06/20/2019	566	23	11	178.97
5997	5997	5998	06/20/2019	114	19	9	205.47
5998	5998	5999	06/20/2019	404	11	20	429.40
5999	5999	6000	06/20/2019	88	57	4	274.52

6000 rows × 7 columns

```
Purchases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6000 entries, 0 to 5999
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    6000 non-null   int64  
 1   id            6000 non-null   int64  
 2   purch_date    6000 non-null   object 
 3   customer_num  6000 non-null   int64  
 4   product_num   6000 non-null   int64  
 5   amount        6000 non-null   int64  
 6   paid          6000 non-null   float64
dtypes: float64(1), int64(5), object(1)
memory usage: 328.3+ KB
```

```
Purchases['purch_date'] = pd.to_datetime(Purchases['purch_date'])
Purchases
```

	Unnamed: 0	id	purch_date	customer_num	product_num	amount	paid
0	0	1	2019-01-03	823	27	12	568.92
1	1	2	2019-01-03	606	28	14	395.36
2	2	3	2019-01-03	955	9	17	510.17
3	3	4	2019-01-03	577	19	3	68.49
4	4	5	2019-01-03	429	8	18	759.42
...	...	...	...	...	...	...	...
5995	5995	5996	2019-06-20	893	33	5	411.10
5996	5996	5997	2019-06-20	566	23	11	178.97
5997	5997	5998	2019-06-20	114	19	9	205.47
5998	5998	5999	2019-06-20	404	11	20	429.40
5999	5999	6000	2019-06-20	88	57	4	274.52

6000 rows × 7 columns

```
Purchases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6000 entries, 0 to 5999
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   Unnamed: 0    6000 non-null   int64         
 1   id            6000 non-null   int64         
 2   purch_date    6000 non-null   datetime64[ns]
 3   customer_num  6000 non-null   int64         
 4   product_num   6000 non-null   int64         
 5   amount        6000 non-null   int64         
 6   paid          6000 non-null   float64       
dtypes: datetime64[ns](1), float64(1), int64(5)
memory usage: 328.3 KB
```

```
Products
```

	Unnamed: 0	id	product	cost	company
0	0	1	Liners - Baking Cups	$6.36	Skipfire
1	1	2	Nori Sea Weed - Gold Label	$85.74	Dynazzy
2	2	3	Bar Bran Honey Nut	$65.40	Ntag
3	3	4	Soup - Campbells Beef Stew	$68.16	Photojam
4	4	5	Wine - Shiraz Wolf Blass Premium	$87.39	Eare
5	5	6	Wine - White, Riesling, Semi - Dry	$99.22	Livepath
6	6	7	Brandy - Bar	$13.83	Oloo
7	7	8	Onions - White	$42.19	Oozz
8	8	9	Lettuce - Baby Salad Greens	$30.01	Meevee

```
Products['cost'] = Products['cost'].astype('str').str.replace('[$,]' , '' , regex = True).astype(float)
Products
```

	Unnamed: 0	id	product	cost	company
0	0	1	Liners - Baking Cups	6.36	Skipfire
1	1	2	Nori Sea Weed - Gold Label	85.74	Dynazzy
2	2	3	Bar Bran Honey Nut	65.40	Ntag
3	3	4	Soup - Campbells Beef Stew	68.16	Photojam
4	4	5	Wine - Shiraz Wolf Blass Premium	87.39	Eare
5	5	6	Wine - White, Riesling, Semi - Dry	99.22	Livepath
6	6	7	Brandy - Bar	13.83	Oloo
7	7	8	Onions - White	42.19	Oozz
8	8	9	Lettuce - Baby Salad Greens	30.01	Meevee

```
Products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  60 non-null     int64
 1   id          60 non-null     int64
 2   product     60 non-null     object
 3   cost        60 non-null     float64
 4   company     55 non-null     object
dtypes: float64(1), int64(2), object(2)
memory usage: 2.5+ KB
```

```python
# Concatenate : pd.concat() -> Append Queries [Union / Union All]
# stack [axis = 0]
combined_df = pd.concat([Customers,Products] , axis = 0)
combined_df
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
FirstCustomer	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
SecondCustomer	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
ThirdCustomer	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
FourthCustomer	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
FifthCustomer	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
55	56.0	NaN	NaN	NaN	NaN	<NA>	NaN	NaN	NaN
56	57.0	NaN	NaN	NaN	NaN	<NA>	NaN	NaN	NaN
57	58.0	NaN	NaN	NaN	NaN	<NA>	NaN	NaN	NaN
58	59.0	NaN	NaN	NaN	NaN	<NA>	NaN	NaN	NaN
59	60.0	NaN	NaN	NaN	NaN	<NA>	NaN	NaN	NaN

1060 rows × 15 columns

```python
# Merging - Merge Queries in Power BI ['Joins' -> Inner, Left , Right]
inner_merged_df = pd.merge(Customers, Products, on = 'id' , how = 'inner')
inner_merged_df
```

gender	street_number	street_address	street_suffix	city	state	postcode	Unnamed: 0	product	cost	company
Male	1	Trailsway	Road	San Diego	California	92127	0	Liners - Baking Cups	6.36	Skipfire
Male	0	NaN	NaN	El Paso	Texas	0	1	Nori Sea Weed - Gold Label	85.74	Dynazzy
Female	4057	Arkansas	Circle	San Antonio	Texas	78220	2	Bar Bran Honey Nut	65.40	Ntag
Male	74	Debs	Point	Memphis	Tennessee	38150	3	Soup - Campbells Beef Stew	68.16	Photojam
								Wine -		

```
left_merged_df = pd.merge(Customers, Products, on = 'id' , how = 'left')
left_merged_df
```

51.la	NaN	0	2nd	Drive	NaN	NaN	83732	4.0	Wine - Shiraz Wolf Blass Premium	87.39	Eare
...	...	...	...	...	...	...	...	...	...	...	...
l.edu	Female	0	Thompson	Way	Reading	Pennsylvania	19610	NaN	NaN	NaN	NaN
.com	Female	701	Rowland	Hill	Indianapolis	Indiana	46278	NaN	NaN	NaN	NaN
NaN	Male	2	Basil	Road	Waterbury	Connecticut	6721	NaN	NaN	NaN	NaN
.com	Male	0	Kenwood	Drive	Savannah	Georgia	31405	NaN	NaN	NaN	NaN
NaN	Male	205	Farwell	Park	Atlanta	Georgia	30386	NaN	NaN	NaN	NaN

```
right_merged_df = pd.merge(Customers, Purchases, on = 'id' , how = 'right')
right_merged_df
```

nber	street_address	street_suffix	city	state	postcode	Unnamed: 0	purch_date	customer_num	product_num	amount	paid
1	Trailsway	Road	San Diego	California	92127.0	0	2019-01-03	823	27	12	568.92
0	NaN	NaN	El Paso	Texas	0.0	1	2019-01-03	606	28	14	395.36
4057	Arkansas	Circle	San Antonio	Texas	78220.0	2	2019-01-03	955	9	17	510.17
74	Debs	Point	Memphis	Tennessee	38150.0	3	2019-01-03	577	19	3	68.49
0	2nd	Drive	NaN	NaN	83732.0	4	2019-01-03	429	8	18	759.42
...	...	...	...	...	...	...	...	...	...	...	...
NA>	NaN	NaN	NaN	NaN	NaN	5995	2019-06-20	893	33	5	411.10
NA>	NaN	NaN	NaN	NaN	NaN	5996	2019-06-20	566	23	11	178.97
NA>	NaN	NaN	NaN	NaN	NaN	5997	2019-06-20	114	19	9	205.47
NA>	NaN	NaN	NaN	NaN	NaN	5998	2019-06-20	404	11	20	429.40
NA>	NaN	NaN	NaN	NaN	NaN	5999	2019-06-20	88	57	4	274.52

```
# Overlapping Columns Handles with Suffixes
Customer = pd.DataFrame(
    {
        'id' : [1,2,3],
        'name' : ['Neha','Palash','Lubhani'],
        'product' : ['Book','Pen','Notebook']
    }
)
Product = pd.DataFrame(
    {
        'id' : [2,3,4],
        'product' : ['Pen','Notebook','Pencil'],
        'amount' : [10,25,15]
    }
)
merged_df = pd.merge(Customer, Product, on='id', how='inner', suffixes = ('_cust','_prod'))
merged_df
```

	id	name	product_cust	product_prod	amount
0	2	Palash	Pen	Pen	10
1	3	Lubhani	Notebook	Notebook	25

```
merged_df = pd.merge(Customer, Product, on='id', how='left', suffixes = ('_cust','_prod'))
merged_df
```

	id	name	product_cust	product_prod	amount
0	1	Neha	Book	NaN	NaN
1	2	Palash	Pen	Pen	10.0
2	3	Lubhani	Notebook	Notebook	25.0

```
merged_df = pd.merge(Customer, Product, on='id', how='right', suffixes = ('_cust','_prod'))
merged_df
```

	id	name	product_cust	product_prod	amount
0	2	Palash	Pen	Pen	10
1	3	Lubhani	Notebook	Notebook	25
2	4	NaN	NaN	Pencil	15

```
# by default -> _x [First DF(Customer)] , _y[Second DF(Product)]
merged_df = pd.merge(Customer, Product, on='id', how='right')
merged_df
```

	id	name	product_x	product_y	amount
0	2	Palash	Pen	Pen	10
1	3	Lubhani	Notebook	Notebook	25
2	4	NaN	NaN	Pencil	15

```
# set_index
Customers = Customers.set_index('id')
Customers
```

id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...
996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury
999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta

1000 rows × 10 columns

```
Purchases = Purchases.set_index('id')
Purchases
```

id	Unnamed: 0	purch_date	customer_num	product_num	amount	paid
1	0	2019-01-03	823	27	12	568.92
2	1	2019-01-03	606	28	14	395.36
3	2	2019-01-03	955	9	17	510.17
4	3	2019-01-03	577	19	3	68.49
5	4	2019-01-03	429	8	18	759.42
...	...	...	...	...	...	...
5996	5995	2019-06-20	893	33	5	411.10
5997	5996	2019-06-20	566	23	11	178.97
5998	5997	2019-06-20	114	19	9	205.47
5999	5998	2019-06-20	404	11	20	429.40
6000	5999	2019-06-20	88	57	4	274.52

6000 rows × 6 columns

```
# Joining with index : DataFrame.join()
# Combine the DataFrame on index basis.
# joined_df = Customers.set_index('id').join(Purchases.set_index('id'))
joined_df = Customers.join(Purchases)
joined_df
```

id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...
996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury
999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta

1000 rows × 16 columns

```
# poping a column [Series]
popped_postcode = Customers.pop('postcode')
Customers
```

first_name	last_name	email	gender	street_number	street_address	street_suffix	city	state
Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego	California
Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso	Texas
Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio	Texas
Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis	Tennessee
Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN	NaN
...	...	...	...	...	...	...	...	...
Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading	Pennsylvania
Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis	Indiana
Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury	Connecticut
Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah	Georgia
Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta	Georgia

/s × 9 columns

```
popped_postcode
```

```
id
1.0       92127
2.0           0
3.0       78220
4.0       38150
5.0       83732
          ...
996.0     19610
997.0     46278
998.0      6721
999.0     31405
1000.0    30386
Name: postcode, Length: 1000, dtype: int32
```

```
Customers['postcode'] = popped_postcode
Customers
```

e	last_name	email	gender	street_number	street_address	street_suffix	city	state	postcode
n	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego	California	92127
o	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso	Texas	0
oi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio	Texas	78220
:e	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis	Tennessee	38150
la	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN	NaN	83732
...	...	...	...	...	...	...	...	...	...
ili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading	Pennsylvania	19610
h	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis	Indiana	46278
it	Birt	NaN	Male	2	Basil	Road	Waterbury	Connecticut	6721
h	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah	Georgia	31405
e	Malpas	NaN	Male	205	Farwell	Park	Atlanta	Georgia	30386

ns

```
Products['postcode'] = popped_postcode
```

```
Products
```

	Unnamed: 0	id	product	cost	company	postcode
0	0	1	Liners - Baking Cups	6.36	Skipfire	NaN
1	1	2	Nori Sea Weed - Gold Label	85.74	Dynazzy	92127.0
2	2	3	Bar Bran Honey Nut	65.40	Ntag	0.0
3	3	4	Soup - Campbells Beef Stew	68.16	Photojam	78220.0
4	4	5	Wine - Shiraz Wolf Blass Premium	87.39	Eare	38150.0
5	5	6	Wine - White, Riesling, Semi - Dry	99.22	Livepath	83732.0
6	6	7	Brandy - Bar	13.83	Oloo	21229.0
7	7	8	Onions - White	42.19	Oozz	60351.0
8	8	9	Lettuce - Baby Salad Greens	30.01	Meevee	64193.0

```
# Data Types are Cleaned now for every DataFrame
Products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  60 non-null     int64
 1   id          60 non-null     int64
 2   product     60 non-null     object
 3   cost        60 non-null     float64
 4   company     55 non-null     object
 5   postcode    59 non-null     float64
dtypes: float64(2), int64(2), object(2)
memory usage: 2.9+ KB
```

```
Purchases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6000 entries, 1 to 6000
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    6000 non-null   int64
 1   purch_date    6000 non-null   datetime64[ns]
 2   customer_num  6000 non-null   int64
 3   product_num   6000 non-null   int64
 4   amount        6000 non-null   int64
 5   paid          6000 non-null   float64
dtypes: datetime64[ns](1), float64(1), int64(4)
memory usage: 328.1 KB
```

```
Customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, 1.0 to 1000.0
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   first_name      1000 non-null   object
 1   last_name       1000 non-null   object
 2   email           878 non-null    object
 3   gender          957 non-null    object
 4   street_number   1000 non-null   Int64 
 5   street_address  963 non-null    object
 6   street_suffix   963 non-null    object
 7   city            921 non-null    object
 8   state           920 non-null    object
 9   postcode        1000 non-null   int32 
dtypes: Int64(1), int32(1), object(8)
memory usage: 115.3+ KB
```

```
-> Finding the missing
-> Handling the missing Values -> 'NaN' , 'NaT' , 'None'
1. Identify them, isnull() , isna() [Boolean Returns] -> Filter
Customers[Customers['gender'].isna()]
Customers[Customers['gender'].isnull()]

2. Strategies to handle Missing Values
- fillna() -> Object['Unknown']
- fillna() -> Numerical[0 or Statistical Value]
- bfill() -> Backward Filling [Fill up]
- ffill() -> Forward Filling [Fill Down]
- dropna() -> Removing the NaN Values [axis = 0/1]
- interpolate -> [linear/ degree - 2] -> 0 2 4 6 _ _ 12 14 [linear]
```

```
# reset_index
Customers = Customers.reset_index()
Customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
1	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
997	998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
999	1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta

1000 rows × 11 columns

```
Purchases = Purchases.reset_index()
Purchases
```

	id	Unnamed: 0	purch_date	customer_num	product_num	amount	paid
0	1	0	2019-01-03	823	27	12	568.92
1	2	1	2019-01-03	606	28	14	395.36
2	3	2	2019-01-03	955	9	17	510.17
3	4	3	2019-01-03	577	19	3	68.49
4	5	4	2019-01-03	429	8	18	759.42
...	...	...	...	...	...	...	...
5995	5996	5995	2019-06-20	893	33	5	411.10
5996	5997	5996	2019-06-20	566	23	11	178.97
5997	5998	5997	2019-06-20	114	19	9	205.47
5998	5999	5998	2019-06-20	404	11	20	429.40
5999	6000	5999	2019-06-20	88	57	4	274.52

6000 rows × 7 columns

```
joined_df = Customers.set_index('id').join(Purchases.set_index('id'))
joined_df
```

id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...
996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury
999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta

1000 rows × 16 columns

```
Customers['email'].isna()
```

```
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997     True
998    False
999     True
Name: email, Length: 1000, dtype: bool
```

```
Customers['email'].isna().sum()
```

```
122
```

```
Customers[Customers['email'].isna()]
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city	state	postcode
24	25.0	Tomlin	Massinger	NaN	Male	9	Victoria	Trail	Washington	District of Columbia	20409
30	31.0	Steve	Pantridge	NaN	Male	6124	Kropf	Street	Los Angeles	California	90065
38	39.0	Stanislas	Hacaud	NaN	Male	76	Summer Ridge	Place	Atlanta	Georgia	31119
48	49.0	Flinn	Hoston	NaN	Male	84	Waxwing	Terrace	Washington	District of Columbia	20530
49	50.0	Claudette	Ivatts	NaN	Female	5552	Bunting	Junction	San Francisco	California	94137
...	...	...	...	...	...	...	...	...	...	...	...
973	974.0	Ingmar	Muzzlewhite	NaN	Male	0	Carpenter	Parkway	Houston	Texas	77260
976	977.0	Corena	Pelz	NaN	Female	62987	Grayhawk	Plaza	Dallas	Texas	0
981	982.0	Judith	Otham	NaN	Female	96973	Saint Paul	Avenue	Arlington	Texas	76096
997	998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury	Connecticut	6721
999	1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta	Georgia	30386

122 rows × 11 columns

```
Customers['state'].isnull()
```

```
0      False
1      False
2      False
3      False
4       True
       ...
995    False
996    False
997    False
998    False
999    False
Name: state, Length: 1000, dtype: bool
```

```
Customers[Customers['state'].isnull()]
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city	state
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN	NaN
16	17.0	Tobi	Dewicke	tdewickeg@hostgator.com	Female	6817	Orin	Junction	San Rafael	NaN
21	22.0	Woodrow	Fidoe	wfidoel@creativecommons.org	Male	1	Artisan	Junction	Gainesville	NaN
22	23.0	Loralie	Normand	lnormandm@usa.gov	Female	1	Transport	Terrace	Duluth	NaN
58	59.0	Helga	Barajas	hbarajas1m@tmall.com	Female	0	Kropf	Drive	Houston	NaN
...	...	...	...	...	...	...	...	...	...	...
892	893.0	Arman	Haste	ahasteos@constantcontact.com	Male	89637	NaN	NaN	Miami	NaN
896	897.0	Ingamar	Anthona	ianthonaow@statcounter.com	Male	0	Rigney	Avenue	Mount Vernon	NaN
939	940.0	Evie	Greaser	egreaserq3@yahoo.com	Female	6	Loomis	Avenue	Bronx	NaN
946	947.0	Paco	Murrock	pmurrockqa@parallels.com	Male	1	Pepper Wood	Drive	Sioux City	NaN
990	991.0	Susana	Pere	spereri@scribd.com	Female	5018	Clemons	Way	Baton Rouge	NaN

80 rows × 11 columns

```
Customers['street_address'].isnull()
```

```
0      False
1       True
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Name: street_address, Length: 1000, dtype: bool
```

```
Customers[Customers['street_address'].isnull()]
```

last_name	email	gender	street_number	street_address	street_suffix	city	state	postcode
Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso	Texas	0
Norrington	nnorringtonb@netvibes.com	Female	5	NaN	NaN	Baltimore	Maryland	21282
Sonley	gsonleyy@deviantart.com	Male	1	NaN	NaN	NaN	Virginia	22070
Issacov	bissacov1a@tiny.cc	Male	54399	NaN	NaN	Dallas	Texas	75397
Padwick	wpadwick4o@godaddy.com	Male	365	NaN	NaN	Houston	Texas	77218
Sidnell	nsidnell5k@apple.com	Female	0	NaN	NaN	NaN	Michigan	49018
Pala	mpala6l@indiegogo.com	Female	0	NaN	NaN	Milwaukee	Wisconsin	53285
Selvey	cselvey6v@adobe.com	Male	66953	NaN	NaN	Helena	NaN	59623
Velden	svelden8l@utexas.edu	Female	8	NaN	NaN	Denver	Colorado	80204

```
# Checking for any missing data
Customers['email'].isnull().any() # Boolean Returns [True means column has missing value else not(False)]
```

```
True
```

```
Customers.isnull().any()
# Providing info of all the columns and letting us know does that column consist of any missing value in it..
```

```
id              False
first_name      False
last_name       False
email            True
gender           True
street_number   False
street_address   True
street_suffix    True
city             True
state            True
postcode        False
dtype: bool
```

```
Customers.isnull().any().any() # Complete DataFrame Evaluate
```

```
True
```

```
Customers.isnull().sum() # How many missing values in each column of a DataFrame # axis = 0
```

```
id                0
first_name        0
last_name         0
email           122
gender           43
street_number     0
street_address   37
street_suffix    37
city             79
state            80
postcode          0
dtype: int64
```

```
Purchases.isnull().any()
```

```
id             False
Unnamed: 0     False
purch_date     False
customer_num   False
product_num    False
amount         False
paid           False
dtype: bool
```

```
Purchases.isnull().any().any() # False
```

```
False
```

```
Purchases.isnull().sum()
```

```
id              0
Unnamed: 0      0
purch_date      0
customer_num    0
product_num     0
amount          0
paid            0
dtype: int64
```

```
Products.isnull().any()
```

```
Unnamed: 0    False
id            False
product       False
cost          False
company        True
postcode       True
dtype: bool
```

```
Products.isnull().any().any()
```

```
True
```

```
Products.isnull().sum()
```

```
Unnamed: 0    0
id            0
product       0
cost          0
company       5
postcode      1
dtype: int64
```

```
Customers.isnull().sum(axis = 1) # Column reviews
```

```
0      0
1      2
2      0
3      0
4      3
      ..
995    0
996    0
997    1
998    0
999    1
Length: 1000, dtype: int64
```

```
Customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
1	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
997	998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
999	1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta

1000 rows × 11 columns

```
Customers.isnull().sum().sum() # axis = 0
```

```
398
```

```
Products.isnull().sum().sum() # axis = 0
```

```
6
```

```
Purchases.isnull().sum().sum() # axis = 0
```

```
0
```

```
# Dropping the missing Values
unique_customers = Customers.copy() # Shallow Copy
unique_customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              1000 non-null   float64
 1   first_name      1000 non-null   object
 2   last_name       1000 non-null   object
 3   email           878 non-null    object
 4   gender          957 non-null    object
 5   street_number   1000 non-null   Int64
 6   street_address  963 non-null    object
 7   street_suffix   963 non-null    object
 8   city            921 non-null    object
 9   state           920 non-null    object
 10  postcode        1000 non-null   int32
dtypes: Int64(1), float64(1), int32(1), object(8)
memory usage: 83.1+ KB
```

```
# Dropping the duplicates of first_name from customers table
unique_customers = Customers.drop_duplicates(subset = 'first_name')
unique_customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
1	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
994	995.0	Brana	Dixon	bdixonrm@myspace.com	Female	97	Truax	Avenue	Maple Plain
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
997	998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury
999	1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta

932 rows × 11 columns

```
unique_customers = Customers.drop_duplicates(subset = 'last_name')
unique_customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
1	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
997	998.0	Tobit	Birt	NaN	Male	2	Basil	Road	Waterbury
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
999	1000.0	Elmore	Malpas	NaN	Male	205	Farwell	Park	Atlanta

993 rows × 11 columns

```
Customers['first_name'].value_counts().head(10)
```

```
first_name
Berty        4
Leland       3
Abel         3
Gunther      2
Silvain      2
Corry        2
Obediah      2
Gabriellia   2
Denyse       2
Margarita    2
Name: count, dtype: int64
```

```
Customers['last_name'].value_counts().head(10)
```

```
last_name
Sedworth     3
Swatridge    2
Robardey     2
Truitt       2
Sunnex       2
Pladen       2
Southcott    1
Marquese     1
Hanrott      1
Borton       1
Name: count, dtype: int64
```

```
# dropna () -> axis = 0/1
unique_customers.dropna() # By Default axis=0 : Removes complete rows if it exist with atleast one 'NaN' in it
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
5	6.0	Magdalena	Cullip	mcullip5@tiny.cc	Female	9190	Packers	Drive	Baltimore
6	7.0	Marietta	Heball	mheball6@blog.com	Female	12	Mallory	Center	Carol Stream
...	...	...	...	...	...	...	...	...	...
992	993.0	Johnath	Clancy	jclancyrk@smugmug.com	Female	7	Washington	Crossing	Juneau
994	995.0	Brana	Dixon	bdixonrm@myspace.com	Female	97	Truax	Avenue	Maple Plain
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading P
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah

674 rows × 11 columns

```
unique_customers.dropna(axis = 1) # It any columns has atleast one NaN value -> Columns ✗
```

	id	first_name	last_name	street_number	postcode
0	1.0	Romain	Southcott	1	92127
1	2.0	Cosimo	Molyneaux	0	0
2	3.0	Bambi	Westrip	4057	78220
3	4.0	Roarke	Pankettman	74	38150
4	5.0	Mikaela	Althorpe	0	83732
...	...	...	...	...	...
995	996.0	Merrili	Alman	0	19610
996	997.0	Winonah	Heckle	701	46278
997	998.0	Tobit	Birt	2	6721
998	999.0	Issiah	Standbrooke	0	31405
999	1000.0	Elmore	Malpas	205	30386

993 rows × 5 columns

```
# Identify is Done ✅
# Now its time to Handle Missing Value -> By using fillna()
Customers.fillna({'email' : 'Unknown'} , inplace = True)
Customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
1	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	NaN	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
997	998.0	Tobit	Birt	Unknown	Male	2	Basil	Road	Waterbury
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
999	1000.0	Elmore	Malpas	Unknown	Male	205	Farwell	Park	Atlanta

1000 rows × 11 columns

```
Customers['gender'].mode()
```

```
0    Male
Name: gender, dtype: object
```

```
type(Customers['gender'].mode())
```

```
pandas.core.series.Series
```

```
Customers['gender'].mode()[0]
```

```
'Male'
```

```
type(Customers['gender'].mode()[0])
```

```
str
```

```
Customers.fillna({'gender' : Customers['gender'].mode()[0]} , inplace = True)
Customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
1	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	0	NaN	NaN	El Paso
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	Male	0	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	0	Thompson	Way	Reading
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
997	998.0	Tobit	Birt	Unknown	Male	2	Basil	Road	Waterbury
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	0	Kenwood	Drive	Savannah
999	1000.0	Elmore	Malpas	Unknown	Male	205	Farwell	Park	Atlanta

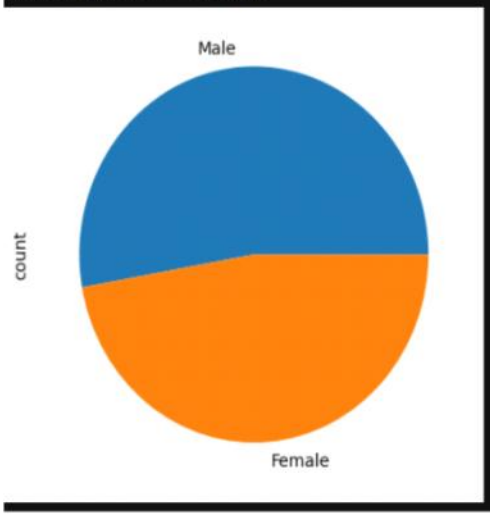
1000 rows × 11 columns

```
Customers['gender'].value_counts()
```

```
gender
Male      528
Female    472
Name: count, dtype: int64
```

```
Customers['gender'].value_counts().plot(kind = 'pie')
```

```
<Axes: ylabel='count'>
```




```
Customers['street_number'] = Customers['street_number'].replace({0 : None})
```

```
Customers
```

	id	first_name	last_name	email	gender	street_number	street_address	street_suffix	city
0	1.0	Romain	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego
1	2.0	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	None	NaN	NaN	El Paso
2	3.0	Bambi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio
3	4.0	Roarke	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis
4	5.0	Mikaela	Althorpe	malthorpe4@51.la	Male	None	2nd	Drive	NaN
...	...	...	...	...	...	...	...	...	...
995	996.0	Merrili	Alman	malmanrn@cornell.edu	Female	None	Thompson	Way	Reading
996	997.0	Winonah	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis
997	998.0	Tobit	Birt	Unknown	Male	2	Basil	Road	Waterbury
998	999.0	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	None	Kenwood	Drive	Savannah
999	1000.0	Elmore	Malpas	Unknown	Male	205	Farwell	Park	Atlanta

1000 rows × 11 columns

```
# Forward Filling -> fill down
Customers['street_number'] = Customers['street_number'].ffill()
Customers['street_address'] = Customers['street_address'].ffill()
Customers['street_suffix'] = Customers['street_suffix'].ffill()
Customers
```

```
C:\Users\krish\AppData\Local\Temp\ipykernel_4976\2638314501.py:2: FutureWarning: Downcasting object dtype arra
ys on .fillna, .ffill, .bfill is deprecated and will change in a future version. Call result.infer_objects(cop
y=False) instead. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  Customers['street_number'] = Customers['street_number'].ffill()
```

e	last_name	email	gender	street_number	street_address	street_suffix	city	state	postcode
n	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego	California	92127
o	Molyneaux	cmolyneaux1@wiley.com	Male	1	Trailsway	Road	El Paso	Texas	0
ɔi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio	Texas	78220
:e	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis	Tennessee	38150
la	Althorpe	malthorpe4@51.la	Male	74	2nd	Drive	NaN	NaN	83732
...	...	...	...	...	...	...	...	...	...
ili	Alman	malmanrn@cornell.edu	Female	97	Thompson	Way	Reading	Pennsylvania	19610
h	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis	Indiana	46278
it	Birt	Unknown	Male	2	Basil	Road	Waterbury	Connecticut	6721
h	Standbrooke	istandbrookerq@yellowpages.com	Male	2	Kenwood	Drive	Savannah	Georgia	31405
'e	Malpas	Unknown	Male	205	Farwell	Park	Atlanta	Georgia	30386

```
# Backward Filling -> .bfill -> fill up
Customers['city'] = Customers['city'].bfill()
Customers['state'] = Customers['state'].bfill()
Customers
```

e	last_name	email	gender	street_number	street_address	street_suffix	city	state	postcode
n	Southcott	rsouthcott0@clickbank.net	Male	1	Trailsway	Road	San Diego	California	92127
o	Molyneaux	cmolyneaux1@wiley.com	Male	1	Trailsway	Road	El Paso	Texas	0
ɔi	Westrip	bwestrip2@symantec.com	Female	4057	Arkansas	Circle	San Antonio	Texas	78220
:e	Pankettman	rpankettman3@wiley.com	Male	74	Debs	Point	Memphis	Tennessee	38150
la	Althorpe	malthorpe4@51.la	Male	74	2nd	Drive	Baltimore	Maryland	83732
...	...	...	...	...	...	...	...	...	...
ili	Alman	malmanrn@cornell.edu	Female	97	Thompson	Way	Reading	Pennsylvania	19610
h	Heckle	whecklero@fc2.com	Female	701	Rowland	Hill	Indianapolis	Indiana	46278
it	Birt	Unknown	Male	2	Basil	Road	Waterbury	Connecticut	6721
h	Standbrooke	istandbrookerq@yellowpages.com	Male	2	Kenwood	Drive	Savannah	Georgia	31405
'e	Malpas	Unknown	Male	205	Farwell	Park	Atlanta	Georgia	30386

```
Customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              1000 non-null   float64
 1   first_name      1000 non-null   object
 2   last_name       1000 non-null   object
 3   email           1000 non-null   object
 4   gender          1000 non-null   object
 5   street_number   1000 non-null   int64
 6   street_address  1000 non-null   object
 7   street_suffix   1000 non-null   object
 8   city            1000 non-null   object
 9   state           1000 non-null   object
 10  postcode        1000 non-null   int32
dtypes: float64(1), int32(1), int64(1), object(8)
memory usage: 82.2+ KB
```

```
Products['company'] = Products['company'].ffill()
Products.drop(columns = ['Unnamed: 0' , 'postcode'] , inplace = True)
Products
```

5	6	Wine - White, Riesling, Semi - Dry	99.22	Livepath
6	7	Brandy - Bar	13.83	Oloo
7	8	Onions - White	42.19	Oozz
8	9	Lettuce - Baby Salad Greens	30.01	Meevee
9	10	Sambuca - Ramazzotti	88.99	Livepath
10	11	Coffee - Decafenated	21.47	Meezzy
11	12	Lamb Leg - Bone - In Nz	42.56	Photofeed
12	13	Lettuce - Spring Mix	19.90	Twitterworks
13	14	Cookies Oatmeal Raisin	99.40	Dynabox
14	15	Cookies Oatmeal Raisin	14.13	Vitz

```
Products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   id       60 non-null     int64
 1   product  60 non-null     object
 2   cost     60 non-null     float64
 3   company  60 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 2.0+ KB
```

```
Purchases.drop(columns = ['Unnamed: 0'] , inplace = True)
Purchases
```

	id	purch_date	customer_num	product_num	amount	paid
0	1	2019-01-03	823	27	12	568.92
1	2	2019-01-03	606	28	14	395.36
2	3	2019-01-03	955	9	17	510.17
3	4	2019-01-03	577	19	3	68.49
4	5	2019-01-03	429	8	18	759.42
...	...	...	...	...	...	...
5995	5996	2019-06-20	893	33	5	411.10
5996	5997	2019-06-20	566	23	11	178.97
5997	5998	2019-06-20	114	19	9	205.47
5998	5999	2019-06-20	404	11	20	429.40
5999	6000	2019-06-20	88	57	4	274.52

6000 rows × 6 columns

```
Purchases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6000 entries, 0 to 5999
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   id            6000 non-null   int64
 1   purch_date    6000 non-null   datetime64[ns]
 2   customer_num  6000 non-null   int64
 3   product_num   6000 non-null   int64
 4   amount        6000 non-null   int64
 5   paid          6000 non-null   float64
dtypes: datetime64[ns](1), float64(1), int64(4)
memory usage: 281.4 KB
```

```
# Interpolate -> Filling the Missing Value with pattern
data = {
    'index' : range(1,11), # [1,2,3,4....]
    'cost' : [100 , 200 , 150 , None , 250 , None , 350, None , 450 , 600]
}
df = pd.DataFrame(data)
df
```

	index	cost
0	1	100.0
1	2	200.0
2	3	150.0
3	4	NaN
4	5	250.0
5	6	NaN
6	7	350.0
7	8	NaN
8	9	450.0
9	10	600.0

```
df['cost'] = df['cost'].interpolate(method = 'linear')
df
```

	index	cost
0	1	100.0
1	2	200.0
2	3	150.0
3	4	200.0
4	5	250.0
5	6	300.0
6	7	350.0
7	8	400.0
8	9	450.0
9	10	600.0

```
# Interpolate -> Filling the Missing Value with pattern
data = {
    'index' : range(1,11), # [1,2,3,4....]
    'cost' : [100 , 200 , 150 , None , 250 , None , 350, None , 450 , 600]
}
df = pd.DataFrame(data)
df['cost'] = df['cost'].interpolate(method = 'polynomial' , order = 2)
df
```

	index	cost
0	1	100.000000
1	2	200.000000
2	3	150.000000
3	4	175.893566
4	5	250.000000
5	6	306.989674
6	7	350.000000
7	8	382.168388
8	9	450.000000
9	10	600.000000

```
# Statiscal Operations
Products['cost'].sum()
```

```
3309.37
```

```
Products['cost'].min()
```

```
6.36
```

```
Products['cost'].max()
```

```
99.54
```

```
Products['cost'].mean()
```

```
55.156166666666664
```

```
Products['cost'].median()
```

```
57.78
```

```
Products['cost'].var()
```

```
759.1239630225987
```

```
Products['cost'].std()
```

```
27.552204322387684
```

```
Products['company'].value_counts().head(10)
```

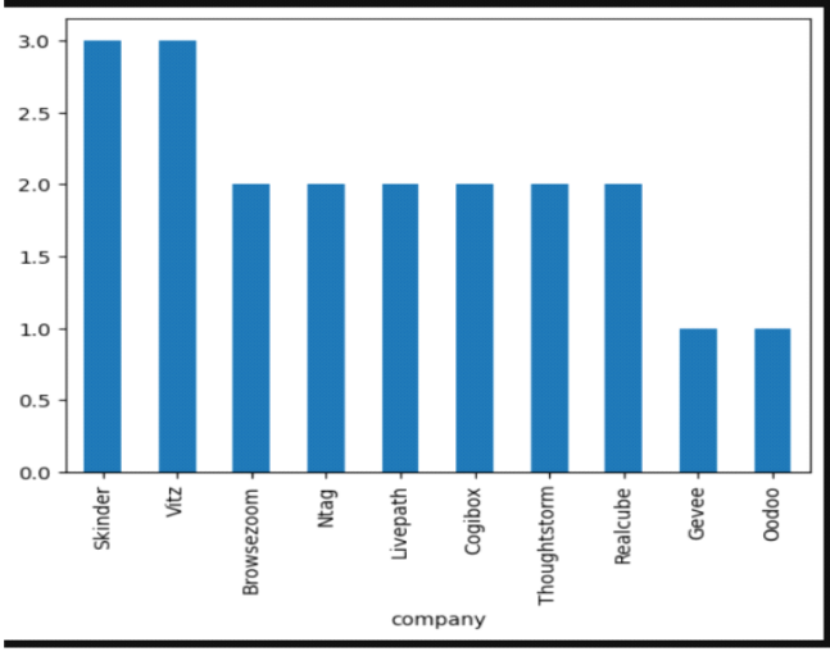
```
company
Skinder          3
Vitz             3
Browsezoom       2
Ntag             2
Livepath         2
Cogibox          2
Thoughtstorm     2
Realcube         2
Gevee            1
Oodoo            1
Name: count, dtype: int64
```

```
Products['company'].mode()
```

```
0    Skinder
1       Vitz
Name: company, dtype: object
```

```
Products['company'].value_counts().head(10).plot(kind = 'bar')
```

```
<Axes: xlabel='company'>
```




```
Products['company'].value_counts().head(10).reset_index()
```

	company	count
0	Skinder	3
1	Vitz	3
2	Browsezoom	2
3	Ntag	2
4	Livepath	2
5	Cogibox	2
6	Thoughtstorm	2
7	Realcube	2
8	Gevee	1
9	Oodoo	1

```
# Data Aggregation -> Group BY + Aggregation
Products['cost'].aggregate('max')
```

```
99.54
```

```
Products['cost'].aggregate(['max','min','sum','mean','median','std','var'])
```

```
max         99.540000
min          6.360000
sum       3309.370000
mean        55.156167
median      57.780000
std         27.552204
var        759.123963
Name: cost, dtype: float64
```

```
Purchases['amount'].aggregate(['max','min','sum','mean','median','std','var'])
```

```
max          20.000000
min           1.000000
sum       63457.000000
mean         10.576167
median       11.000000
std           5.768889
var          33.280079
Name: amount, dtype: float64
```

```
Purchases['paid'].aggregate(['max','min','sum','mean','median','std','var'])
```

```
max       1.990800e+03
min       3.630000e+00
sum       3.293228e+06
mean      5.488714e+02
median    4.227000e+02
std       4.442372e+02
var       1.973467e+05
Name: paid, dtype: float64
```

```
Products.groupby('company')['cost'].sum()
```

```
company
Aibox           56.33
Babbleopia      63.98
Brainsphere     22.83
Browsezoom      82.10
Cogibox        102.27
Digitube         8.55
Dynabox         99.40
Dynazzy         85.74
Eare            87.39
Eazzy           74.28
Edgeclub        40.69
Eimbee          98.74
Fanoodle        68.63
Flipstorm       37.79
Fliptune        75.32
```