

```
In [1]: import numpy as np
import pandas as pd
```

```
In [2]: data=pd.read_csv(r"c:\Users\user\Downloads\Salesworkload.csv")
data
```

Out[2]:

|      | MonthYear | Time index | Country        | StoreID | City       | Dept_ID | Dept. Name          | HoursOwn | HoursLea |
|------|-----------|------------|----------------|---------|------------|---------|---------------------|----------|----------|
| 0    | 10.2016   | 1.0        | United Kingdom | 88253.0 | London (I) | 1.0     | Dry                 | 3184.764 |          |
| 1    | 10.2016   | 1.0        | United Kingdom | 88253.0 | London (I) | 2.0     | Frozen              | 1582.941 |          |
| 2    | 10.2016   | 1.0        | United Kingdom | 88253.0 | London (I) | 3.0     | other               | 47.205   |          |
| 3    | 10.2016   | 1.0        | United Kingdom | 88253.0 | London (I) | 4.0     | Fish                | 1623.852 |          |
| 4    | 10.2016   | 1.0        | United Kingdom | 88253.0 | London (I) | 5.0     | Fruits & Vegetables | 1759.173 |          |
| ...  | ...       | ...        | ...            | ...     | ...        | ...     | ...                 | ...      | ...      |
| 7653 | 6.2017    | 9.0        | Sweden         | 29650.0 | Gothenburg | 12.0    | Checkout            | 6322.323 |          |
| 7654 | 6.2017    | 9.0        | Sweden         | 29650.0 | Gothenburg | 16.0    | Customer Services   | 4270.479 |          |
| 7655 | 6.2017    | 9.0        | Sweden         | 29650.0 | Gothenburg | 11.0    | Delivery            | 0        |          |
| 7656 | 6.2017    | 9.0        | Sweden         | 29650.0 | Gothenburg | 17.0    | others              | 2224.929 |          |
| 7657 | 6.2017    | 9.0        | Sweden         | 29650.0 | Gothenburg | 18.0    | all                 | 39652.2  |          |

7658 rows × 14 columns



Find mean, median, mode and describe

```
In [3]: print(data.mean())
```

```
Time index      5.000000e+00
StoreID         6.199522e+04
Dept_ID         9.470588e+00
HoursLease      2.203608e+01
Sales units     1.076471e+06
Turnover        3.721393e+06
Customer        NaN
dtype: float64
```

```
In [8]: print(data.median())
```

```
Time index      5.0
StoreID         75400.5
Dept_ID         9.0
HoursLease       0.0
Sales units     293230.0
Turnover        931957.5
Customer        NaN
dtype: float64
```

```
In [7]: print(data.mode())
```

|    |     |     |     |         |             |     |
|----|-----|-----|-----|---------|-------------|-----|
| 24 | NaN | NaN | NaN | 73949.0 | Leicester   | NaN |
| 25 | NaN | NaN | NaN | 76852.0 | Liverpool   | NaN |
| 26 | NaN | NaN | NaN | 77348.0 | London (I)  | NaN |
| 27 | NaN | NaN | NaN | 78325.0 | London (II) | NaN |
| 28 | NaN | NaN | NaN | 78450.0 | Lyon        | NaN |
| 29 | NaN | NaN | NaN | 79785.0 | Madrid (I)  | NaN |
| 30 | NaN | NaN | NaN | 81473.0 | Madrid (II) | NaN |
| 31 | NaN | NaN | NaN | 83160.0 | Malmö       | NaN |
| 32 | NaN | NaN | NaN | 85124.0 | Manchester  | NaN |
| 33 | NaN | NaN | NaN | 85321.0 | Marseille   | NaN |
| 34 | NaN | NaN | NaN | 85696.0 | Milano      | NaN |
| 35 | NaN | NaN | NaN | 86089.0 | Munich      | NaN |
| 36 | NaN | NaN | NaN | 86208.0 | Nantes      | NaN |
| 37 | NaN | NaN | NaN | 87703.0 | Napoli      | NaN |
| 38 | NaN | NaN | NaN | 88253.0 | Ostrava     | NaN |
| 39 | NaN | NaN | NaN | 88750.0 | Paris (I)   | NaN |
| 40 | NaN | NaN | NaN | 88994.0 | Paris (II)  | NaN |
| 41 | NaN | NaN | NaN | 90992.0 | Poznan      | NaN |
| 42 | NaN | NaN | NaN | 91973.0 | Prague (I)  | NaN |
| 43 | NaN | NaN | NaN | 93033.0 | Prague (II) | NaN |

```
In [9]: print(data.describe())
```

|       |             |              |             |             |               |
|-------|-------------|--------------|-------------|-------------|---------------|
|       | Time index  | StoreID      | Dept_ID     | HoursLease  | Sales units \ |
| count | 7650.000000 | 7650.000000  | 7650.000000 | 7650.000000 | 7.650000e+03  |
| mean  | 5.000000    | 61995.220000 | 9.470588    | 22.036078   | 1.076471e+06  |
| std   | 2.582158    | 29924.581631 | 5.337429    | 133.299513  | 1.728113e+06  |
| min   | 1.000000    | 12227.000000 | 1.000000    | 0.000000    | 0.000000e+00  |
| 25%   | 3.000000    | 29650.000000 | 5.000000    | 0.000000    | 5.457125e+04  |
| 50%   | 5.000000    | 75400.500000 | 9.000000    | 0.000000    | 2.932300e+05  |
| 75%   | 7.000000    | 87703.000000 | 14.000000   | 0.000000    | 9.175075e+05  |
| max   | 9.000000    | 98422.000000 | 18.000000   | 3984.000000 | 1.124296e+07  |

  

|       |              |          |
|-------|--------------|----------|
|       | Turnover     | Customer |
| count | 7.650000e+03 | 0.0      |
| mean  | 3.721393e+06 | NaN      |
| std   | 6.003380e+06 | NaN      |
| min   | 0.000000e+00 | NaN      |
| 25%   | 2.726798e+05 | NaN      |
| 50%   | 9.319575e+05 | NaN      |
| 75%   | 3.264432e+06 | NaN      |
| max   | 4.271739e+07 | NaN      |

Find sum(), cumsum(), count, min and max values

```
In [10]: print(data.sum())
```

```
MonthYear      10.201610.201610.201610.201610.201610.201610.2...
Time index                                38250.0
StoreID        474263433.0
Dept_ID        72450.0
HoursLease     168576.0
Sales units    8235000965.0
Turnover       28468656015.0
Customer              0.0
dtype: object
```

```
In [11]: print(data.count())
```

```
MonthYear      7658
Time index     7650
Country        7650
StoreID        7650
City           7650
Dept_ID        7650
Dept. Name     7650
HoursOwn       7650
HoursLease     7650
Sales units    7650
Turnover       7650
Customer        0
Area (m2)      7650
Opening hours  7650
dtype: int64
```

```
In [12]: print(data.max())
```

```
MonthYear      6.2017
Time index      9.0
StoreID        98422.0
Dept_ID        18.0
HoursLease     3984.0
Sales units    11242955.0
Turnover       42717390.0
Customer       NaN
dtype: object
```

```
In [13]: print(data.min())
```

```
MonthYear      - - - -  
Time index      1.0  
StoreID        12227.0  
Dept_ID        1.0  
HoursLease      0.0  
Sales units     0.0  
Turnover        0.0  
Customer        NaN  
dtype: object
```

```
In [14]: print(data.cumsum())
```

```
-----  
TypeError                                Traceback (most recent call last)  
C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\fromnumeric.py in _wrap  
apfunc(obj, method, *args, **kws)  
    57     try:  
--> 58         return bound(*args, **kws)  
    59     except TypeError:
```

**TypeError:** can only concatenate str (not "float") to str

During handling of the above exception, another exception occurred:

```
TypeError                                Traceback (most recent call last)  
<ipython-input-14-2dfb061a761e> in <module>  
----> 1 print(data.cumsum())  
  
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in cumsum  
(self, axis, skipna, *args, **kwargs)  
    11027     )  
    11028     def cumsum(self, axis=None, skipna=True, *args, **kwargs):
```

Find covariance and correlation (spearman and pearsons)

```
In [17]: from numpy import cov  
from numpy import mean,std  
from numpy.random import randn,seed  
from matplotlib import pyplot
```

```
In [18]: print(mean(data.HoursOwn),std(data.HoursOwn))  
         print(mean(data.Dept_ID),std(data.Dept_ID))
```

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-18-a6b41187dd7f> in <module>
----> 1 print(mean(data.HoursOwn),std(data.HoursOwn))
      2 print(mean(data.Dept_ID),std(data.Dept_ID))

<__array_function__ internals> in mean(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\fromnumeric.py in mean(a, axis, dtype, out, keepdims, where)
    3415         pass
    3416     else:
-> 3417         return mean(axis=axis, dtype=dtype, out=out, **kwargs)
    3418
    3419     return _methods._mean(a, axis=axis, dtype=dtype,

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in mean(self, axis, skipna, level, numeric_only, **kwargs)
    11116         )
    11117     def mean(self, axis=None, skipna=None, level=None, numeric_only=None, **kwargs):
> 11118         return NDFrame.mean(self, axis, skipna, level, numeric_only, **kwargs)
    11119
    11120     # pandas\core\generic.py:10924: error: Cannot assign to a method
hod

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in mean(self, axis, skipna, level, numeric_only, **kwargs)
    10724
    10725     def mean(self, axis=None, skipna=None, level=None, numeric_only=None, **kwargs):
> 10726         return self._stat_function(
    10727             "mean", nanops.nanmean, axis, skipna, level, numeric_only, **kwargs
    10728         )

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in _stat_function(self, name, func, axis, skipna, level, numeric_only, **kwargs)
    10709         if level is not None:
    10710             return self._agg_by_level(name, axis=axis, level=level, skipna=skipna)
> 10711         return self._reduce(
    10712             func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_only
    10713         )

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\series.py in _reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kws)
    4180         )
    4181         with np.errstate(all="ignore"):
-> 4182             return op(delegate, skipna=skipna, **kws)
    4183
    4184     def _reindex_indexer(self, new_index, indexer, copy):

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\nanops.py in _f(*args, **kwargs)

```

```

71         try:
72             with np.errstate(invalid="ignore"):
---> 73                 return f(*args, **kwargs)
74         except ValueError as e:
75             # we want to transform an object array

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\nanops.py in f(values,
axis, skipna, **kws)
    133         result = alt(values, axis=axis, skipna=skipna, **
kws)
    134     else:
--> 135         result = alt(values, axis=axis, skipna=skipna, **kwd
s)
    136
    137     return result

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\nanops.py in new_func
(values, axis, skipna, mask, **kwargs)
    392         mask = isna(values)
    393
--> 394         result = func(values, axis=axis, skipna=skipna, mask=mask, **
kwargs)
    395
    396         if datetimelike:

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\nanops.py in nanmean(v
alues, axis, skipna, mask)
    631
    632     count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 633     the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    634
    635     if axis is not None and getattr(the_sum, "ndim", False):

C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\_methods.py in _sum(a,
axis, dtype, out, keepdims, initial, where)
    45 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
    46         initial=_NoValue, where=True):
---> 47     return umr_sum(a, axis, dtype, out, keepdims, initial, where)
    48
    49 def _prod(a, axis=None, dtype=None, out=None, keepdims=False,

```

**TypeError:** can only concatenate str (not "int") to str

```
In [22]: print(cov(data.HoursOwn,data.Dept_ID))
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-22-81f0f79bd685> in <module>
----> 1 print(cov(data.HoursOwn,data.Dept_ID))

<__array_function__ internals> in cov(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\function_base.py in cov
(m, y, rowvar, bias, ddof, fweights, aweights, dtype)
    2467         w *= aweights
    2468
--> 2469     avg, w_sum = average(X, axis=1, weights=w, returned=True)
    2470     w_sum = w_sum[0]
    2471

<__array_function__ internals> in average(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\function_base.py in aver
age(a, axis, weights, returned)
    378
    379     if weights is None:
--> 380         avg = a.mean(axis)
    381         scl = avg.dtype.type(a.size/avg.size)
    382     else:

C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\_methods.py in _mean(a,
axis, dtype, out, keepdims, where)
    176         is_float16_result = True
    177
--> 178     ret = umr_sum(arr, axis, dtype, out, keepdims, where=where)
    179     if isinstance(ret, mu.ndarray):
    180         ret = um.true_divide(

TypeError: can only concatenate str (not "float") to str
```



```
In [23]: from scipy.stats import pearsonr
print(pearsonr(data.HoursOwn,data.Dept_ID))
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-23-38a94f383ba5> in <module>
      1 from scipy.stats import pearsonr
----> 2 print(pearsonr(data.HoursOwn,data.Dept_ID))

C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py in pearsonr
(x, y)
    3917     # that the data type is at least 64 bit floating point.  It might
have
    3918     # more precision if the input is, for example, np.longdouble.
-> 3919     dtype = type(1.0 + x[0] + y[0])
    3920
    3921     if n == 2:
```

**TypeError:** unsupported operand type(s) for +: 'float' and 'str'

```
In [24]: from scipy.stats import spearmanr
print(spearmanr(data.HoursOwn,data.Dept_ID))
```

```
SpearmanrResult(correlation=nan, pvalue=nan)
```

```
In [ ]:
```