

# 01-SF Salaries Exercise

July 2, 2025

## 1 Assignment - 3

**AICTE Faculty ID:** 1-3241967546

**Faculty Name:** Milav Jayeshkuamar Dabgar

**Date:** July 2, 2025

---

## 2 SF Salaries Exercise

Welcome to a quick exercise for you to practice your pandas skills! We will be using the [SF Salaries Dataset](#) from Kaggle! Just follow along and complete the tasks outlined in bold below. The tasks will get harder and harder as you go along.

**\*\* Import pandas as pd.\*\***

```
[1]: import pandas as pd
```

**\*\* Read Salaries.csv as a dataframe called sal.\*\***

```
[2]: sal = pd.read_csv('Salaries.csv')
```

**\*\* Check the head of the DataFrame. \*\***

```
[3]: sal.head()
```

```
[3]:
```

	Id	EmployeeName	JobTitle	\
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	

  

	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	\
0	167411.18	0.00	400184.25	NaN	567595.43	567595.43	
1	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	
2	212739.13	106088.18	16452.60	NaN	335279.91	335279.91	
3	77916.00	56120.71	198306.90	NaN	332343.61	332343.61	
4	134401.60	9737.00	182234.59	NaN	326373.19	326373.19	

	Year	Notes	Agency	Status
0	2011	NaN	San Francisco	NaN
1	2011	NaN	San Francisco	NaN
2	2011	NaN	San Francisco	NaN
3	2011	NaN	San Francisco	NaN
4	2011	NaN	San Francisco	NaN

**\*\* Use the .info() method to find out how many entries there are.\*\***

```
[4]: sal.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148654 entries, 0 to 148653
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Id                    148654 non-null int64
1   EmployeeName          148654 non-null object
2   JobTitle              148654 non-null object
3   BasePay               148045 non-null float64
4   OvertimePay           148650 non-null float64
5   OtherPay              148650 non-null float64
6   Benefits              112491 non-null float64
7   TotalPay              148654 non-null float64
8   TotalPayBenefits      148654 non-null float64
9   Year                  148654 non-null int64
10  Notes                 0 non-null      float64
11  Agency                148654 non-null object
12  Status                0 non-null      float64
dtypes: float64(8), int64(2), object(3)
memory usage: 14.7+ MB
```

**What is the average BasePay ?**

```
[5]: sal['BasePay'].mean()
```

```
[5]: np.float64(66325.44884048769)
```

**\*\* What is the highest amount of OvertimePay in the dataset ? \*\***

```
[6]: sal['OvertimePay'].max()
```

```
[6]: np.float64(245131.88)
```

**\*\* What is the job title of JOSEPH DRISCOLL ? Note: Use all caps, otherwise you may get an answer that doesn't match up (there is also a lowercase Joseph Driscoll). \*\***

```
[7]: sal[sal['EmployeeName'] == 'JOSEPH DRISCOLL']['JobTitle']
```

```
[7]: 24    CAPTAIN, FIRE SUPPRESSION
      Name: JobTitle, dtype: object
```

**\*\* How much does JOSEPH DRISCOLL make (including benefits)? \*\***

```
[8]: sal[sal['EmployeeName'] == 'JOSEPH DRISCOLL']['TotalPayBenefits']
```

```
[8]: 24    270324.91
      Name: TotalPayBenefits, dtype: float64
```

**\*\* What is the name of highest paid person (including benefits)?\*\***

```
[9]: sal[sal['TotalPayBenefits'] == sal['TotalPayBenefits'].max()]['EmployeeName']
```

```
[9]: 0    NATHANIEL FORD
      Name: EmployeeName, dtype: object
```

**\*\* What is the name of lowest paid person (including benefits)? Do you notice something strange about how much he or she is paid?\*\***

```
[10]: sal[sal['TotalPayBenefits'] == sal['TotalPayBenefits'].min()]['EmployeeName']
```

```
[10]: 148653    Joe Lopez
      Name: EmployeeName, dtype: object
```

**\*\* What was the average (mean) BasePay of all employees per year? (2011-2014) ? \*\***

```
[11]: sal.groupby('Year')['BasePay'].mean()
```

```
[11]: Year
      2011    63595.956517
      2012    65436.406857
      2013    69630.030216
      2014    66564.421924
      Name: BasePay, dtype: float64
```

**\*\* How many unique job titles are there? \*\***

```
[12]: sal['JobTitle'].nunique()
```

```
[12]: 2159
```

**\*\* What are the top 5 most common jobs? \*\***

```
[13]: sal['JobTitle'].value_counts().head()
```

```
[13]: JobTitle
      Transit Operator    7036
      Special Nurse      4389
      Registered Nurse    3736
      Public Svc Aide-Public Works    2518
```

```
Police Officer 3                2421
Name: count, dtype: int64
```

**\*\* How many Job Titles were represented by only one person in 2013? (e.g. Job Titles with only one occurrence in 2013?) \*\***

```
[14]: sum(sal[sal['Year'] == 2013]['JobTitle'].value_counts() == 1)
```

```
[14]: 202
```

**\*\* How many people have the word Chief in their job title? (This is pretty tricky) \*\***

```
[15]: sum(sal['JobTitle'].str.contains('chief', case=False, na=False))
```

```
[15]: 627
```

**\*\* Bonus: Is there a correlation between length of the Job Title string and Salary? \*\***

```
[16]: # Create a new column for job title length
sal['title_len'] = sal['JobTitle'].apply(len)

# Calculate correlation between title length and TotalPay
correlation = sal[['title_len', 'TotalPay']].corr()
print("Correlation Matrix:")
print(correlation)
print("\nInterpretation:")
print(f"The correlation coefficient between job title length and total pay is {correlation.iloc[0,1]:.6f}")
print("This indicates a very weak negative correlation, meaning there is
↳ virtually no relationship")
print("between the length of a job title and the salary amount. Job title
↳ length does not")
print("predict or influence salary in any meaningful way.")
```

Correlation Matrix:

	title_len	TotalPay
title_len	1.000000	-0.015356
TotalPay	-0.015356	1.000000

Interpretation:

The correlation coefficient between job title length and total pay is -0.015356  
This indicates a very weak negative correlation, meaning there is virtually no relationship  
between the length of a job title and the salary amount. Job title length does not  
predict or influence salary in any meaningful way.

### 3 Great Job!

# 03-Ecommerce Purchases Exercise

July 2, 2025

## 1 Assignment - 3

**AICTE Faculty ID:** 1-3241967546

**Faculty Name:** Milav Jayeshkuamar Dabgar

**Date:** July 2, 2025

---

## 2 Ecommerce Purchases Exercise

In this Exercise you will be given some Fake Data about some purchases done through Amazon! Just go ahead and follow the directions and try your best to answer the questions and complete the tasks. Feel free to reference the solutions. Most of the tasks can be solved in different ways. For the most part, the questions get progressively harder.

Please excuse anything that doesn't make "Real-World" sense in the dataframe, all the data is fake and made-up.

Also note that all of these questions can be answered with one line of code. `__` **\*\* Import pandas and read in the Ecommerce Purchases csv file and set it to a DataFrame called ecom. \*\***

```
[1]: import pandas as pd
ecom = pd.read_csv('Ecommerce Purchases')
```

Check the head of the DataFrame.

```
[2]: ecom.head()
```

```
[2]:
```

	Address	Lot	AM or PM	\
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	
4	23012 Munoz Drive Suite 337\nNew Cynthia, TX 5...	20 IE	AM	

  

```
Browser Info \
```

0	Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...
1	Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Pr...
2	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...
3	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...

4 Opera/9.58.(X11; Linux x86\_64; it-IT) Presto/2...

	Company	Credit Card	CC Exp Date	\
0	Martinez-Herman	6011929061123406	02/20	
1	Fletcher, Richards and Whitaker	3337758169645356	11/18	
2	Simpson, Williams and Pham	675957666125	08/19	
3	Williams, Marshall and Buchanan	6011578504430710	02/24	
4	Brown, Watson and Andrews	6011456623207998	10/25	

	CC Security Code	CC Provider	\
0	900	JCB 16 digit	
1	561	Mastercard	
2	699	JCB 16 digit	
3	384	Discover	
4	678	Diners Club / Carte Blanche	

	Email	Job	\
0	pdunlap@yahoo.com	Scientist, product/process development	
1	anthony41@reed.com	Drilling engineer	
2	amymiller@morales-harrison.com	Customer service manager	
3	brent16@olson-robinson.info	Drilling engineer	
4	christopherwright@gmail.com	Fine artist	

	IP Address	Language	Purchase Price
0	149.146.147.205	el	98.14
1	15.160.41.51	fr	70.73
2	132.207.160.22	de	0.95
3	30.250.74.19	es	78.04
4	24.140.33.94	es	77.82

\*\* How many rows and columns are there? \*\*

```
[3]: ecom.shape
```

```
[3]: (10000, 14)
```

\*\* What is the average Purchase Price? \*\*

```
[4]: ecom['Purchase Price'].mean()
```

```
[4]: np.float64(50.347302)
```

\*\* What were the highest and lowest purchase prices? \*\*

```
[5]: ecom['Purchase Price'].max()
```

```
[5]: np.float64(99.99)
```

```
[6]: ecom['Purchase Price'].min()
```

```
[6]: np.float64(0.0)
```

**\*\* How many people have English 'en' as their Language of choice on the website? \*\***

```
[7]: ecom[ecom['Language'] == 'en'].shape[0]
```

```
[7]: 1098
```

**\*\* How many people have the job title of "Lawyer" ? \*\***

```
[8]: ecom[ecom['Job'] == 'Lawyer'].shape[0]
```

```
[8]: 30
```

**\*\* How many people made the purchase during the AM and how many people made the purchase during PM ? \*\***

**(Hint: Check out `value_counts()` )**

```
[9]: ecom['AM or PM'].value_counts()
```

```
[9]: AM or PM
     PM      5068
     AM      4932
     Name: count, dtype: int64
```

**\*\* What are the 5 most common Job Titles? \*\***

```
[10]: ecom['Job'].value_counts().head()
```

```
[10]: Job
     Interior and spatial designer    31
     Lawyer                          30
     Social researcher                28
     Purchasing manager              27
     Designer, jewellery             27
     Name: count, dtype: int64
```

**\*\* Someone made a purchase that came from Lot: "90 WT" , what was the Purchase Price for this transaction? \*\***

```
[11]: ecom[ecom['Lot'] == '90 WT']['Purchase Price']
```

```
[11]: 513      75.1
     Name: Purchase Price, dtype: float64
```

**\*\* What is the email of the person with the following Credit Card Number: 4926535242672853 \*\***

```
[12]: ecom[ecom['Credit Card'] == 4926535242672853]['Email']
```

```
[12]: 1234      bondellen@williams-garza.com
     Name: Email, dtype: object
```

**\*\* How many people have American Express as their Credit Card Provider *and* made a purchase above \$95 ?\*\***

```
[13]: ecom[(ecom['CC Provider'] == 'American Express') & (ecom['Purchase Price'] > 95)].shape[0]
```

[13]: 39

**\*\* Hard: How many people have a credit card that expires in 2025? \*\***

```
[14]: ecom[ecom['CC Exp Date'].str.contains('/25')].shape[0]
```

[14]: 1033

**\*\* Hard: What are the top 5 most popular email providers/hosts (e.g. gmail.com, yahoo.com, etc...) \*\***

```
[15]: ecom['Email'].str.split('@').str[1].value_counts().head()
```

```
[15]: Email
hotmail.com      1638
yahoo.com        1616
gmail.com        1605
smith.com         42
williams.com      37
Name: count, dtype: int64
```

### 3 Great Job!