week-01-solution

August 27, 2024

[21]:
```
import logging
#logging config
logging.basicConfig(
                                        level=logging.DEBUG,
                                        format='%(levelname)-6s | %(asctime)s |␣
 ↪%(message)s',
                                        filename=f'logs/week-01-activities.log',
                                        filemode='w',
                                        )
logger = logging.getLogger()
```

0.0.1 Q-3

- How many rows are in dataset: `week-01/datasets/A.csv`? How about in `week-01/datasets/B.txt` and in `week-01/datasets/C.csv`?

[32]:
```
with open('datasets/A.csv','r') as f:
    counter = 0
    for line in f:
        counter += 1
    logger.info(f'num lines of A is {counter}')
```

[33]:
```
with open('datasets/B.txt','r') as f:
    counter = 0
    for line in f:
        counter += 1
    logger.info(f'num lines of B is {counter}')
```

[34]:
```
with open('datasets/C.csv','r') as f:
    counter = 0
    for line in f:
        counter += 1
    logger.info(f'num lines of C is {counter}')
```

[36]:
```
#a better ~ time-efficient way
with open('datasets/A.csv','r') as f:
    logger.info(f'num lines of A is {sum([1 for _ in f])}')
```

0.0.2 Q-4

- How many samples are in dataset: `week-01/datasets/A.csv`? How about in `week-01/datasets/B.txt` and in `week-01/datasets/C.csv`?

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

[8]:
```
A = pd.read_csv('datasets/A.csv',
                    sep=',', #default is comma (,)
                    header=0, #default is `infer` ~ header=0th row
                   )
```

[9]:
```
A.head(n=1)
```

[9]:
```
   CustomerId Surname  CreditScore Geography Gender  Age  Tenure   Balance  \
0    15647572  Greece          504     Spain   Male   34       0  54980.81

   NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  Exited
0              1          1               1        136909.88       0
```

[10]:
```
A.columns
```

[10]:
```
Index(['CustomerId', 'Surname', 'CreditScore', 'Geography', 'Gender', 'Age',
       'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember',
       'EstimatedSalary', 'Exited'],
      dtype='object')
```

[11]:
```
A.shape
```

[11]:
```
(9000, 13)
```

[22]:
```
logger.info(f'num rows of A is {A.shape[0]}')
```

[15]:
```
B = pd.read_csv('datasets/B.txt',
                header=None, #no header row present
                sep='\s+',
               )
```

[16]:
```
B.head(n=1)
```

[16]:
```
            0     1     2     3      4     5     6     7     8     9     10  \
0  AQC00914000  1981  4279  3745  10762  6067  4096  3606  6203  5292  3092

     11    12    13
0  6866  7163  7866
```

[17]:
```
B.shape
```

[17]:
```
(232043, 14)
```

[23]:
```
logger.info(f'num rows of B is {B.shape[0]}')
```

[ ]:
```

```

[28]:
```
C = pd.read_csv('datasets/C.csv',
                header=0, #first row is header row
                sep=';',
                skiprows=[1] #skip the second row
               )
```

[30]:
```
C.head(n=5)
```

[30]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.0	0.33	68.402973
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.0	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.0	0.33	59.425505
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	3	1.0	0.50	93.704912
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	3	1.0	0.75	34.384843

[31]:
```
logger.info(f'num rows of C is {C.shape[0]}')
```

0.0.3 Q-5

- How many columns are there in each of the 3 datasets?

[37]:
```
logger.info(f'num of columns in A is {len(A.columns)}')
logger.info(f'num of columns in B is {len(B.columns)}')
logger.info(f'num of columns in C is {len(C.columns)}')
```

0.0.4 Q-6

- Compute the mean of the last (i.e., rightmost) column of `week-01/datasets/C.csv`?

[50]:
```
logger.info(f'mean of rightmost column of C is {C[["rating"]].mean().iloc[0,]:.
↪2f}')
```

0.0.5 Q-7

- Where (i.e, in which sample) the two datasets: `week-01/datasets/B.txt` and `week-01/datasets/D.txt` differ?

[51]:
```
!diff datasets/B.txt datasets/D.txt
```

```
/bin/bash: /home/ashiskb/miniconda3/envs/venv-p39-tf213user/lib/libtinfo.so.6:
no version information available (required by /bin/bash)
112590c112590
< USC00286460 1981   279   1511    261    916   1559    910   1351    386    855
1114    396   1192
---
> USC00286460 1981   279   1511    261    916   1559    911   1351    386    855
1114    396   1192
```

0.0.6 Q-8

- What is the average credit score among the samples found in `week-01/datasets/A.csv`?

[52]:
```
A.head()
```

[52]:

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	\
0	15647572	Greece	504	Spain	Male	34	0	
1	15797692	Volkova	659	France	Female	33	7	
2	15713559	Onyemauchechukwu	473	Germany	Female	32	5	
3	15595067	Zhirov	637	Spain	Female	40	6	
4	15810167	Scott	657	Spain	Male	75	7	

	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	\
0	54980.81	1	1	1	136909.88	
1	89939.62	1	1	0	136540.09	
2	146602.25	2	1	1	72946.95	
3	0.00	2	1	1	181610.60	
4	126273.95	1	0	1	91673.60	

	Exited
0	0
1	0
2	0
3	0
4	0

[56]:
```
logger.info(f'Average credit score of {A[["CreditScore"]].mean().iloc[0,]:.2f}')
```

0.0.7 Q-9

- How many different countries are listed in `week-01/datasts/A.csv`?

[61]:
```
A['Geography'].unique()
```

[61]:
```
array(['Spain', 'France', 'Germany'], dtype=object)
```

[63]:
```
logger.info(f'num of different countries in A is: {A["Geography"].nunique()}')
```

0.0.8 Q-10

- Please briefly describe each of the 3 datasets (i.e., what the datasets are about)
- **A dataset**
- Bank Customer Churn Prediction data.
- Source: Kaggle
- **B dataset**
- Monthly precipation values (normals) data from ~9000 weather stations from National Centers for Environmental Information between 1981-2010.
- source mly-prcp-filled.txt
- **C dataset**
- dataset about 80 cereals and their nutrition facts
- Source 80-cereals

0.0.9 Q-11

- Care to explore more of the datasets?

[ ]:
```

```