Seneca

Academic Year	**2022 – 2023**
Semester	☒ Fall ☐ Winter ☐ Summer
Course Code - Name	BAN130
Instructor	Zeynep Cevik, PhD
Assessment	Projects
Due Date	Tuesday, December 05, 2023

Student ID	**Student Name**	**Role**
141305227	Mamat Jasseh	Member

Bonus Project

This project is completely optional. You are required to choose a project from the list of the projects specified in this document and complete it by yourself. The project is expected to test your technical skills in Python programming.

The detailed requirements for each project are available in this document, so please go through the details and fulfil all the requirements to avoid missing any marks.

Finally, follow the below mentioned instructions carefully.

Instructions:

In order to obtain maximum marks in this assessment, please ensure the followings:

- Don’t forget to write your name and ID on the first page of this document.
- Submit the project by writing your solution in this document under the Solution heading below. Do not use a separate document. Everything related to the project should be included in this document, e.g., code, screenshots and etc.
- This project has a weightage of **10%** marks of the course as a bonus.
- The project deadline is **December 5, 2023**. Submissions after the deadline will not be accepted.

Rubric:

Your assessment will be graded based on the following rubric:

	Excellent (7 - 10)	**Average (4 – 6.9)**	**Poor (<4)**
Project Completion and Code (10)	The project was completed without any errors and output is as expected. Fulfills all/most of the requirements for the project.	The project was completed with few errors. Fulfills some of the requirements for the project.	The project is incomplete. Does not fulfill all/most of the requirements.
Detailed Explanation (10)	The student has a good contribution to the project. Knows ins and outs of the project. The student has written his/her part of the project very well. Knows everything / most of his/her part.	The student has average contribution to the project. Does not know the whole project. The student has averagely written his/her part of the project. Knows few of the things about his/her part.	The student has no contribution to the project. Does not know anything / most about the project. The student has poorly written the project. Does not know much about the project.
Report (10)	Student has contributed well in preparing the project	Student has contributed partially in preparing the project	Student has not contributed in preparing the report.

	report and knows all the aspects of the report.	report and knows some aspects of the report.	

Project 1

Project Name: Adventure Works Product Sales Analysis

Dataset: AdventureWorks.xlsx (Available on Blackboard)

Requirements:
Below are bare minimum requirements for this project, however, you are free to add more features to your project: (hint: pandas library)

1. Data Import
    - This phase requires you to import the data from the provided excel file into Python.
        - Product sheet in excel file should be imported as Product dataset.
        - SalesOrderDetail sheet in excel file should be imported as SalesOrderDetail dataset.
2. Data Cleaning
    - This phase requires you to clean your data before data analysis phase.
        - Product_Clean:
            - Create a Product_Clean dataset from Product dataset by bringing in only ProductID, Name, ProductNumber, Color and ListPrice
            - All the missing values in Color column should be replaced by ‘NA’
            - ListPrice column should be float (final column name should be ListPrice) with 2 decimal places
        - SalesOrderDetail_Clean:
            - Create SalesOrderDetail_Clean dataset from SalesOrderDetail dataset by bringing in only SalesOrderID SalesOrderDetailID OrderQty ProductID UnitPrice LineTotal and ModifiedDate
            - ModifiedDate should be date with column name ModifiedDate
            - UnitPrice should be float with column name UnitPrice
            - LineTotal should be float with column name LineTotal
            - OrderQty should be integer with column name OrderQty
            - Include date for year 2013 and 2014 in ModifiedDate only
            - ModifiedDate should be date format
            - UnitPrice and LineTotal are float with 2 decimal places

Project 2

Project Name: Adventure Works Territory Sales Analysis

Dataset: AdventureWorks.xlsx (Available on Blackboard)

Requirements:
Below are bare minimum requirements for this project, however, you are free to add more features to your project: (hint: pandas library)

1. Data Import
   - This phase requires you to import the data from the provided excel file into Python.
     - SalesTerritory sheet in excel file should be imported as SalesTerritory dataset.
     - SalesOrderHeader sheet in excel file should be imported as SalesOrderHeader dataset.
2. Data Cleaning
   - This phase requires you to clean your data before data analysis phase.
     - SalesOrderHeader_Clean:
       - Create a SalesOrderHeader_Clean dataset from SalesOrderHeader dataset by bringing in only SalesOrderID OrderDate OnlineOrderFlag TerritoryID TotalDue
       - TotalDue column should be float (final column name should be TotalDue) with 2 decimal places
       - OnlineOrderFlag column should be integer (final column name should be OnlineOrderFlag)
       - OrderDate column should be date (final column name should be OrderDate).
       - TerritoryID column should be integer (final column name should be TerritoryID)
       - No un-necessary columns should be part of the SalesOrderHeader_Clean dataset.

     - Territory_Clean:
       - Create Territory_Clean dataset from SalesTerritory dataset by bringing in only TerritoryID Name CountryRegionCode Group SalesYTD
       - SalesYTD column should be floar (final column name should be SalesYTD) with 2 decimal places
       - TerritoryID column should be integer (final column name should be TerritoryID

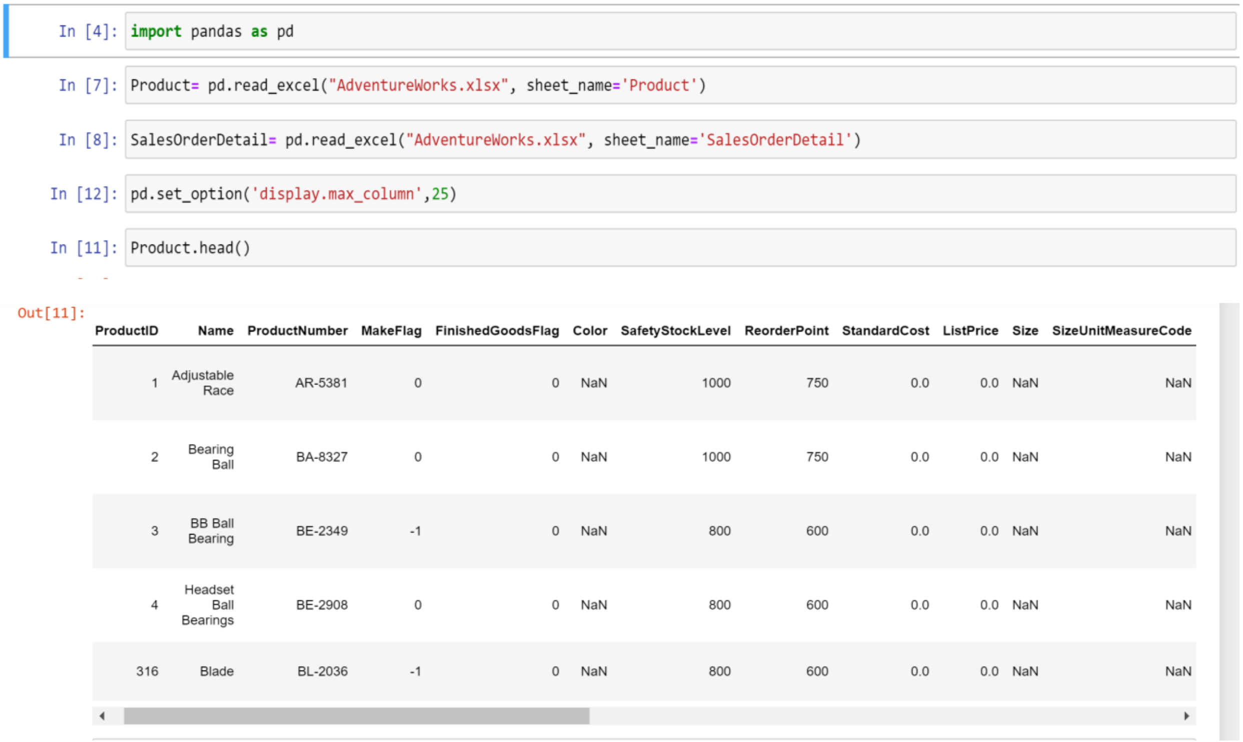
SOLUTIONS

PROJECT 1 REPORT:

1. Data Import

The product sheet needs to be imported as a product dataset from an Excel file.

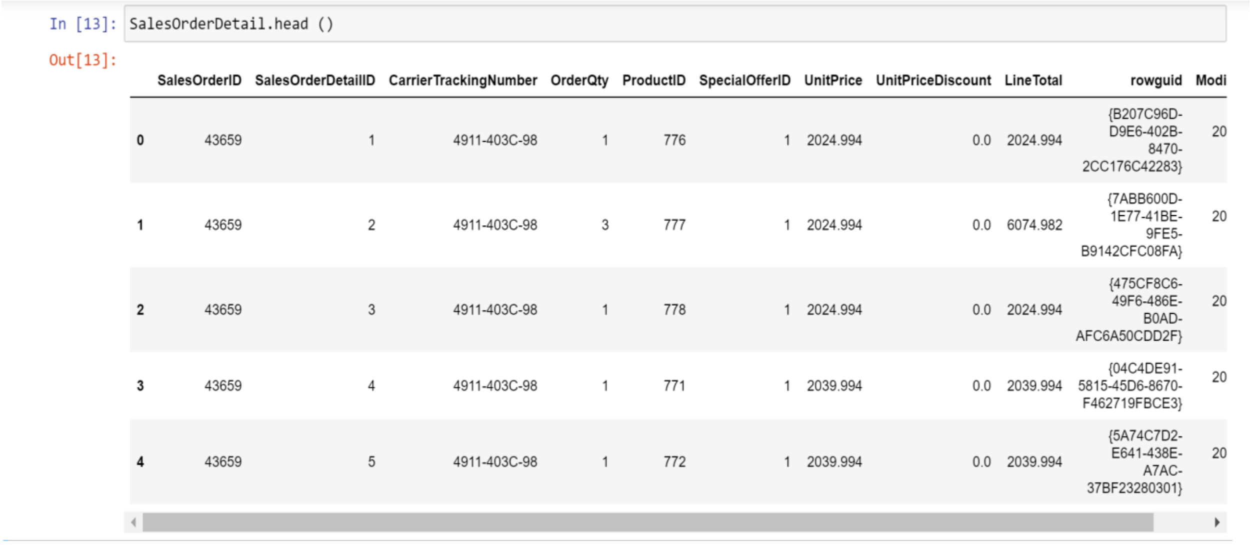
The Excel file containing the SalesOrderDetail sheet needs to be imported as a SalesOrderDetail dataset.



```
In [4]: import pandas as pd

In [7]: Product= pd.read_excel("AdventureWorks.xlsx", sheet_name='Product')

In [8]: SalesOrderDetail= pd.read_excel("AdventureWorks.xlsx", sheet_name='SalesOrderDetail')

In [12]: pd.set_option('display.max_column',25)

In [11]: Product.head()
```

Out[11]:

ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint	StandardCost	ListPrice	Size	SizeUnitMeasureCode
1	Adjustable Race	AR-5381	0	0	NaN	1000	750	0.0	0.0	NaN	NaN
2	Bearing Ball	BA-8327	0	0	NaN	1000	750	0.0	0.0	NaN	NaN
3	BB Ball Bearing	BE-2349	-1	0	NaN	800	600	0.0	0.0	NaN	NaN
4	Headset Ball Bearings	BE-2908	0	0	NaN	800	600	0.0	0.0	NaN	NaN
316	Blade	BL-2036	-1	0	NaN	800	600	0.0	0.0	NaN	NaN



```
In [13]: SalesOrderDetail.head ()
```

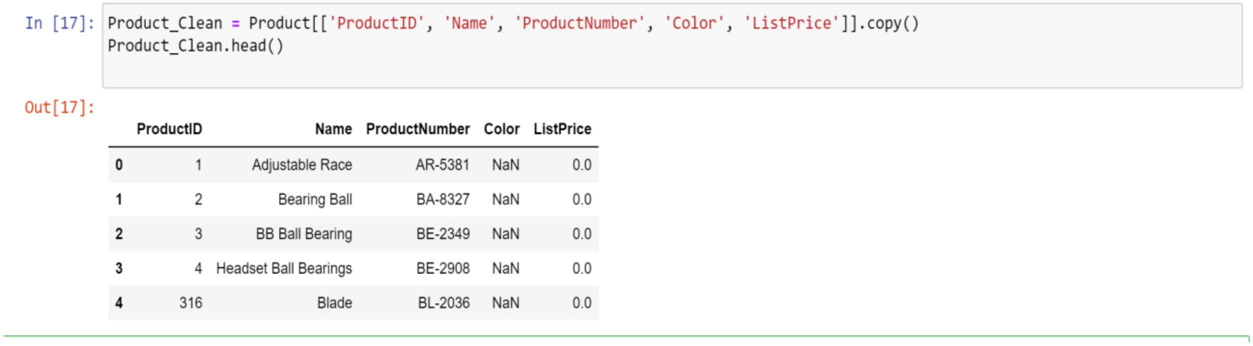
Out[13]:

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid	Modi
0	43659	1	4911-403C-98	1	776	1	2024.994	0.0	2024.994	{B207C96D-D9E6-402B-8470-2CC176C42283}	20
1	43659	2	4911-403C-98	3	777	1	2024.994	0.0	6074.982	{7ABB600D-1E77-41BE-9FE5-B9142CFC08FA}	20
2	43659	3	4911-403C-98	1	778	1	2024.994	0.0	2024.994	{475CF8C6-49F6-486E-B0AD-AFC6A50CDD2F}	20
3	43659	4	4911-403C-98	1	771	1	2039.994	0.0	2039.994	{04C4DE91-5815-45D6-8670-F462719FBCE3}	20
4	43659	5	4911-403C-98	1	772	1	2039.994	0.0	2039.994	{5A74C7D2-E641-438E-A7AC-37BF23280301}	20

2. Data Cleaning

Product Clean

Created a Product_Clean dataset from the Product dataset by bringing in only ProductID, Name, ProductNumber, Color and ListPrice

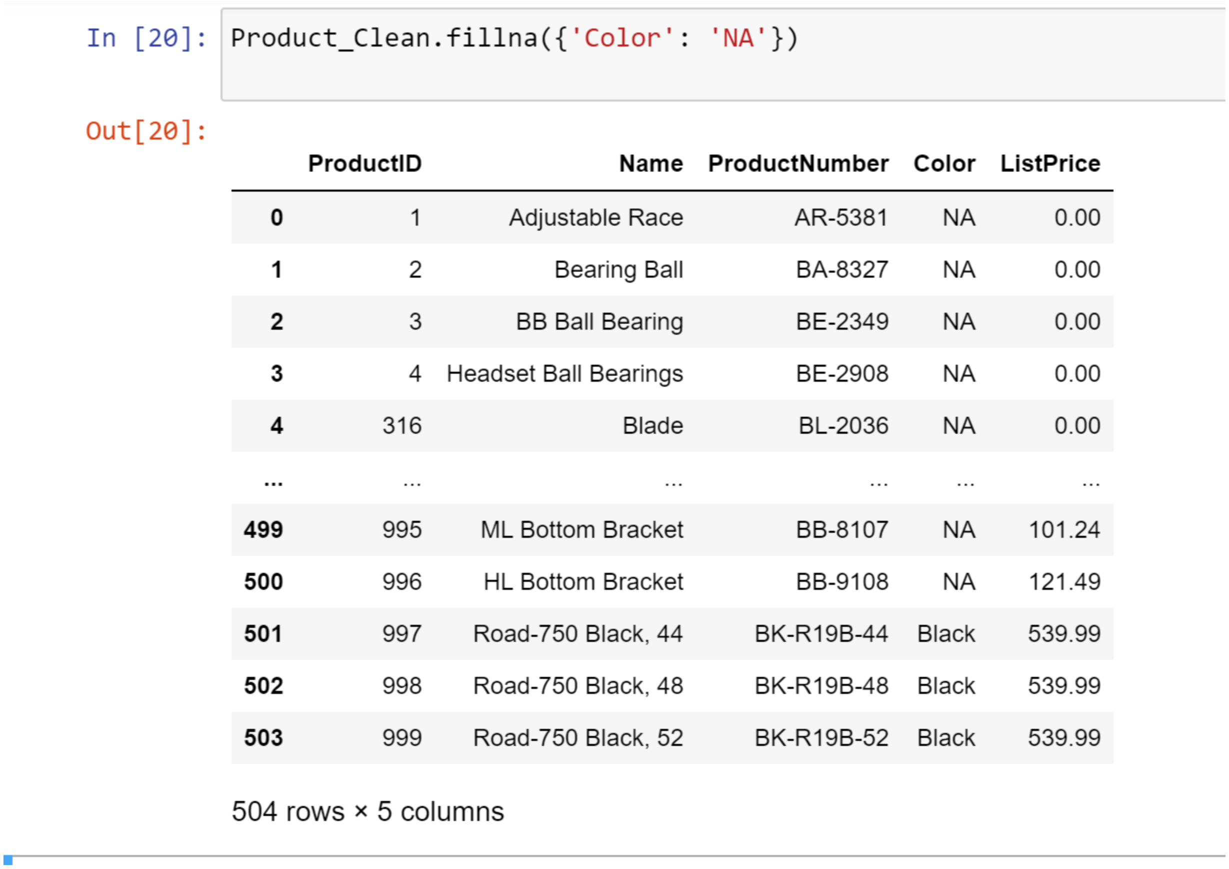


```
In [17]: Product_Clean = Product[['ProductID', 'Name', 'ProductNumber', 'Color', 'ListPrice']].copy()
         Product_Clean.head()
```

Out[17]:

	ProductID	Name	ProductNumber	Color	ListPrice
0	1	Adjustable Race	AR-5381	NaN	0.0
1	2	Bearing Ball	BA-8327	NaN	0.0
2	3	BB Ball Bearing	BE-2349	NaN	0.0
3	4	Headset Ball Bearings	BE-2908	NaN	0.0
4	316	Blade	BL-2036	NaN	0.0

All instances of missing values in the 'Color' column are replaced with the value 'NA'.



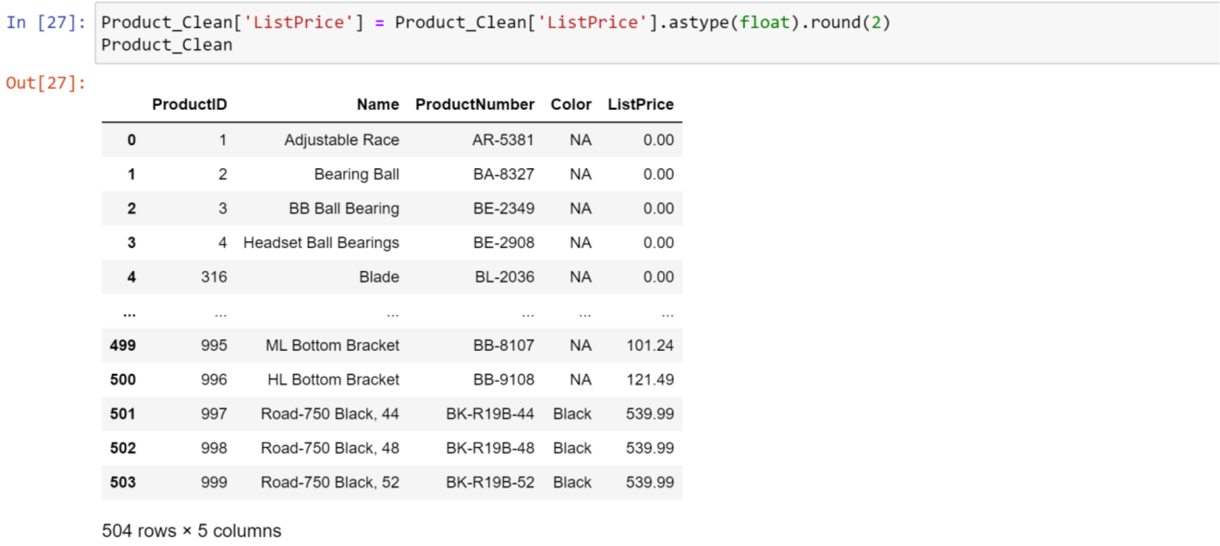
In [20]:
```python
Product_Clean.fillna({'Color': 'NA'})
```

Out[20]:

	ProductID	Name	ProductNumber	Color	ListPrice
0	1	Adjustable Race	AR-5381	NA	0.00
1	2	Bearing Ball	BA-8327	NA	0.00
2	3	BB Ball Bearing	BE-2349	NA	0.00
3	4	Headset Ball Bearings	BE-2908	NA	0.00
4	316	Blade	BL-2036	NA	0.00
...	...	...	...	...	...
499	995	ML Bottom Bracket	BB-8107	NA	101.24
500	996	HL Bottom Bracket	BB-9108	NA	121.49
501	997	Road-750 Black, 44	BK-R19B-44	Black	539.99
502	998	Road-750 Black, 48	BK-R19B-48	Black	539.99
503	999	Road-750 Black, 52	BK-R19B-52	Black	539.99

504 rows × 5 columns

T ListPrice column should be a float with two decimal places, and its final column name should be ListPrice.



In [27]:
```python
Product_Clean['ListPrice'] = Product_Clean['ListPrice'].astype(float).round(2)
Product_Clean
```

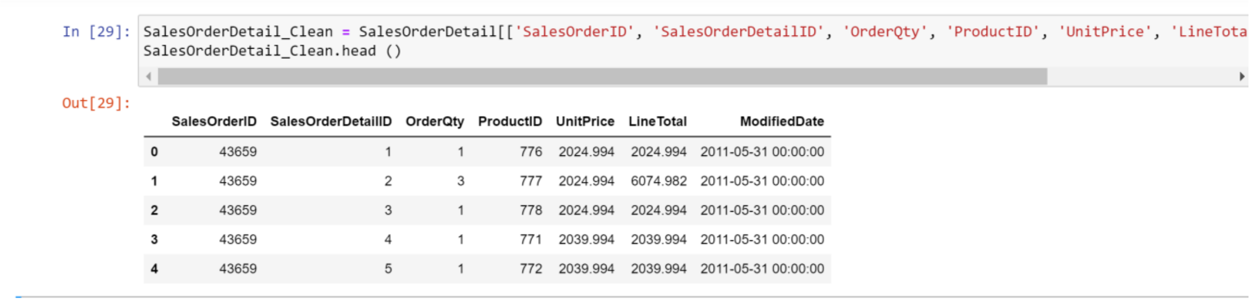
Out[27]:

	ProductID	Name	ProductNumber	Color	ListPrice
0	1	Adjustable Race	AR-5381	NA	0.00
1	2	Bearing Ball	BA-8327	NA	0.00
2	3	BB Ball Bearing	BE-2349	NA	0.00
3	4	Headset Ball Bearings	BE-2908	NA	0.00
4	316	Blade	BL-2036	NA	0.00
...	...	...	...	...	...
499	995	ML Bottom Bracket	BB-8107	NA	101.24
500	996	HL Bottom Bracket	BB-9108	NA	121.49
501	997	Road-750 Black, 44	BK-R19B-44	Black	539.99
502	998	Road-750 Black, 48	BK-R19B-48	Black	539.99
503	999	Road-750 Black, 52	BK-R19B-52	Black	539.99

504 rows × 5 columns

SalesOrderDetail_Clean:

Created SalesOrderDetail_Clean dataset with only SalesOrderID, SalesOrderDetailID, OrderQty, ProductID, UnitPrice, LineTotal, and ModifiedDate from the SalesOrderDetail dataset.

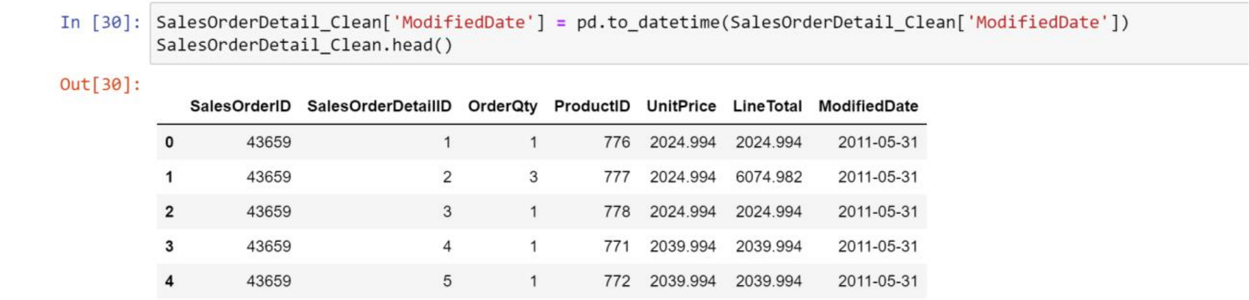


```
In [29]: SalesOrderDetail_Clean = SalesOrderDetail[['SalesOrderID', 'SalesOrderDetailID', 'OrderQty', 'ProductID', 'UnitPrice', 'LineTota
         SalesOrderDetail_Clean.head ()
```

Out[29]:

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal	ModifiedDate
0	43659	1	1	776	2024.994	2024.994	2011-05-31 00:00:00
1	43659	2	3	777	2024.994	6074.982	2011-05-31 00:00:00
2	43659	3	1	778	2024.994	2024.994	2011-05-31 00:00:00
3	43659	4	1	771	2039.994	2039.994	2011-05-31 00:00:00
4	43659	5	1	772	2039.994	2039.994	2011-05-31 00:00:00

The column name ModifiedDate should contain a date that indicates when the modification was made.

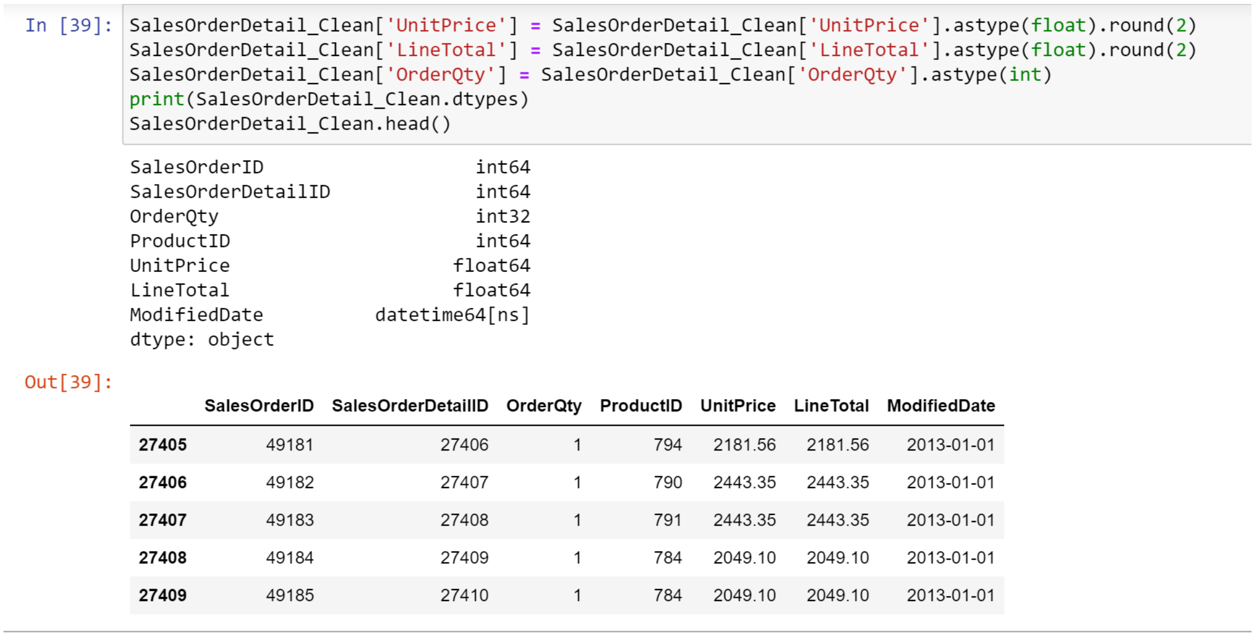


```
In [30]: SalesOrderDetail_Clean['ModifiedDate'] = pd.to_datetime(SalesOrderDetail_Clean['ModifiedDate'])
         SalesOrderDetail_Clean.head()
```

Out[30]:

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal	ModifiedDate
0	43659	1	1	776	2024.994	2024.994	2011-05-31
1	43659	2	3	777	2024.994	6074.982	2011-05-31
2	43659	3	1	778	2024.994	2024.994	2011-05-31
3	43659	4	1	771	2039.994	2039.994	2011-05-31
4	43659	5	1	772	2039.994	2039.994	2011-05-31

Column names and data types are specified. UnitPrice and LineTotal are now floats with 2 decimal places, while OrderQty is an integer.



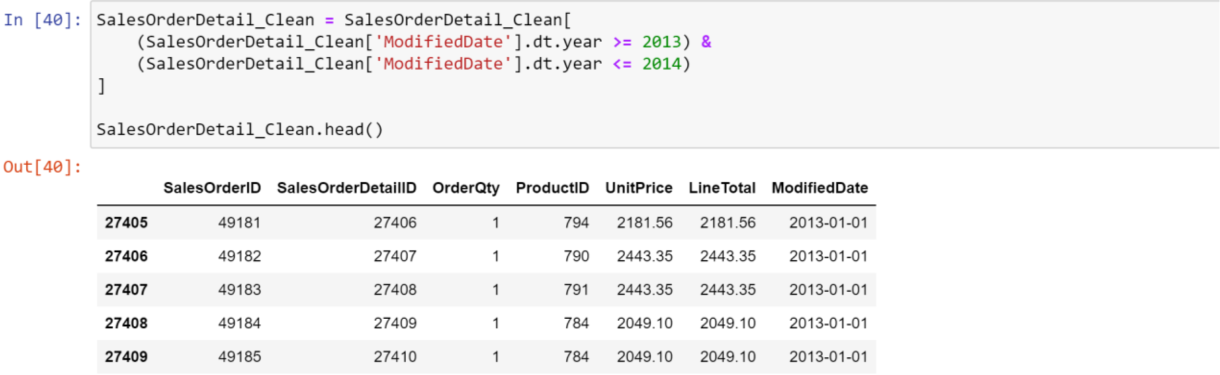
```
In [39]: SalesOrderDetail_Clean['UnitPrice'] = SalesOrderDetail_Clean['UnitPrice'].astype(float).round(2)
         SalesOrderDetail_Clean['LineTotal'] = SalesOrderDetail_Clean['LineTotal'].astype(float).round(2)
         SalesOrderDetail_Clean['OrderQty'] = SalesOrderDetail_Clean['OrderQty'].astype(int)
         print(SalesOrderDetail_Clean.dtypes)
         SalesOrderDetail_Clean.head()
```

```
SalesOrderID                   int64
SalesOrderDetailID             int64
OrderQty                       int32
ProductID                      int64
UnitPrice                    float64
LineTotal                    float64
ModifiedDate          datetime64[ns]
dtype: object
```

Out[39]:

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal	ModifiedDate
27405	49181	27406	1	794	2181.56	2181.56	2013-01-01
27406	49182	27407	1	790	2443.35	2443.35	2013-01-01
27407	49183	27408	1	791	2443.35	2443.35	2013-01-01
27408	49184	27409	1	784	2049.10	2049.10	2013-01-01
27409	49185	27410	1	784	2049.10	2049.10	2013-01-01

Included date for year 2013 and 2014 in ModifiedDate field as date format.



In [40]:

```
SalesOrderDetail_Clean = SalesOrderDetail_Clean[
    (SalesOrderDetail_Clean['ModifiedDate'].dt.year >= 2013) &
    (SalesOrderDetail_Clean['ModifiedDate'].dt.year <= 2014)
]

SalesOrderDetail_Clean.head()
```

Out[40]:

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal	ModifiedDate
27405	49181	27406	1	794	2181.56	2181.56	2013-01-01
27406	49182	27407	1	790	2443.35	2443.35	2013-01-01
27407	49183	27408	1	791	2443.35	2443.35	2013-01-01
27408	49184	27409	1	784	2049.10	2049.10	2013-01-01
27409	49185	27410	1	784	2049.10	2049.10	2013-01-01