Till Now,

01 Importing Liberaries & Data Sets.

02 How to Deal With null values?

03 Data Formatting and Data Standardization

04 How to Deal With Categorical Values?

05 Spliting Dataset.

06 Data Normalization & Future Scaling.

07 Working on Real World Dataset (Project)

MISSING VALUES

Missing Values occur when no data value is stored for a variable(feature) in an observation.

Reasons :

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Standard Missing Values

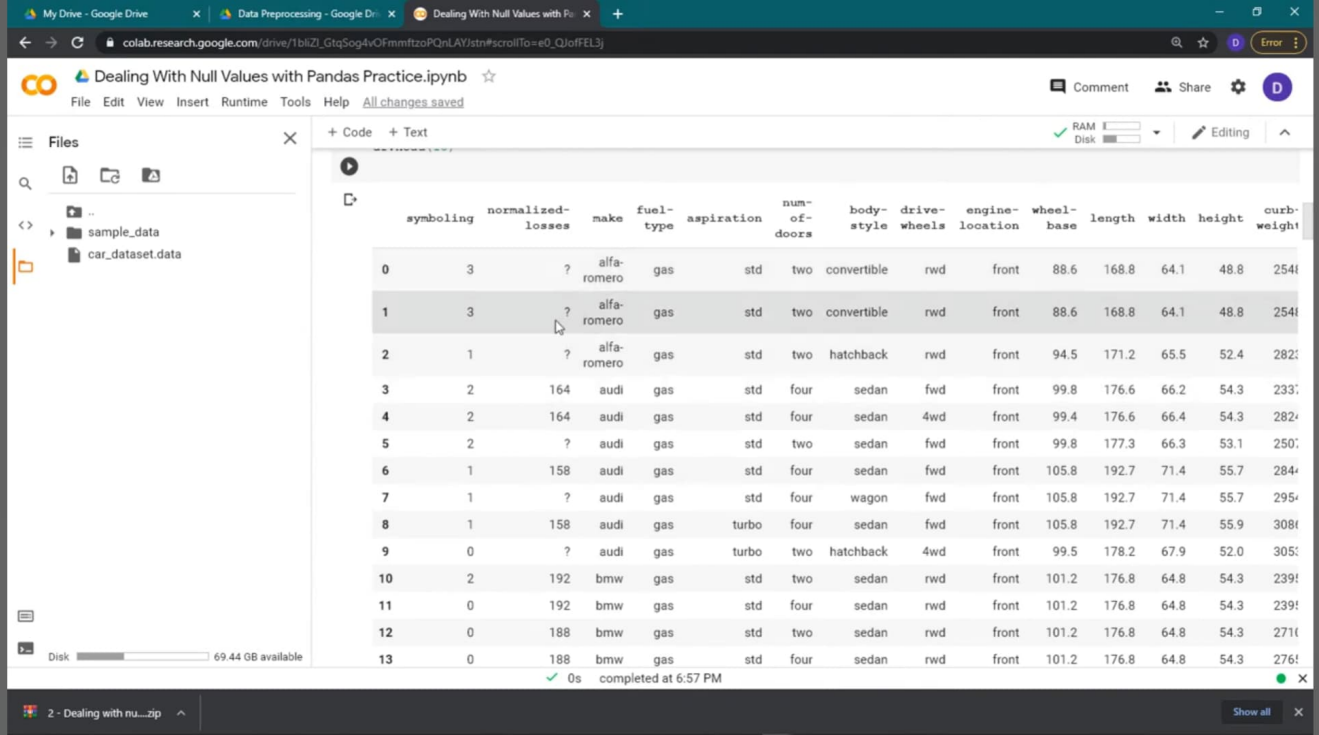
ST_NUM	ST_NAME	NUM_BEDROOMS	OWN_OCCUPIED
104	PUTNAM	3	Y
197	LEXINGTON	3	N
	LEXINGTON	n/a	N
201	BERKELEY	1	12
203	BERKELEY	3	Y
207	BERKELEY	NA	Y
	WASHINGTON	2	
213	TREMONT	--	Y
215	TREMONT	na	Y

Non-Standard Missing Values

ST_NUM	ST_NAME	NUM_BEDROOMS	OWN_OCCUPIED
104	PUTNAM	3	Y
197	LEXINGTON	3	N
	LEXINGTON	n/a	N
201	BERKELEY	1	12
203	BERKELEY	3	Y
207	BERKELEY	NA	Y
NA	WASHINGTON	2	
213	TREMONT	--	Y
215	TREMONT	na	Y

Unexpected Missing Values

ST_NUM	ST_NAME	NUM_BEDROOMS	OWN_OCCUPIED
104	PUTNAM	3	Y
197	LEXINGTON	3	N
	LEXINGTON	n/a	N
201	BERKELEY	1	12
203	BERKELEY	3	Y
207	BERKELEY	NA	Y
NA	WASHINGTON	2	
213	TREMONT	--	Y
215	TREMONT	na	Y


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZI_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=e0_QJofFEL3j
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
D
Files
..
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
symboling normalized-losses make fuel-type aspiration num-of-doors body-style drive-wheels engine-location wheel-base length width height curb-weight
0 3 ? alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 2548
1 3 ? alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 2548
2 1 ? alfa-romero gas std two hatchback rwd front 94.5 171.2 65.5 52.4 2823
3 2 164 audi gas std four sedan fwd front 99.8 176.6 66.2 54.3 2337
4 2 164 audi gas std four sedan 4wd front 99.4 176.6 66.4 54.3 2824
5 2 ? audi gas std two sedan fwd front 99.8 177.3 66.3 53.1 2507
6 1 158 audi gas std four sedan fwd front 105.8 192.7 71.4 55.7 2844
7 1 ? audi gas std four wagon fwd front 105.8 192.7 71.4 55.7 2954
8 1 158 audi gas turbo four sedan fwd front 105.8 192.7 71.4 55.9 3086
9 0 ? audi gas turbo two hatchback 4wd front 99.5 178.2 67.9 52.0 3053
10 2 192 bmw gas std two sedan rwd front 101.2 176.8 64.8 54.3 2395
11 0 192 bmw gas std four sedan rwd front 101.2 176.8 64.8 54.3 2395
12 0 188 bmw gas std two sedan rwd front 101.2 176.8 64.8 54.3 2710
13 0 188 bmw gas std four sedan rwd front 101.2 176.8 64.8 54.3 2765
0s completed at 6:57 PM
2 - Dealing with nu....zip
Show all
BIT ML

```
[5]   19  stroke             205 non-null    object
      20  compression-ratio  205 non-null    float64
      21  horsepower         205 non-null    object
      22  peak-rpm           205 non-null    object
      23  city-mpg           205 non-null    int64
      24  highway-mpg        205 non-null    int64
      25  price              205 non-null    object
     dtypes: float64(5), int64(5), object(16)
     memory usage: 41.8+ KB
```

As we can see, several question marks appeared in the dataframe; those are missing values which may hinder our further analysis.

So, how do we identify all those missing values and deal with them?

How to work with missing data?

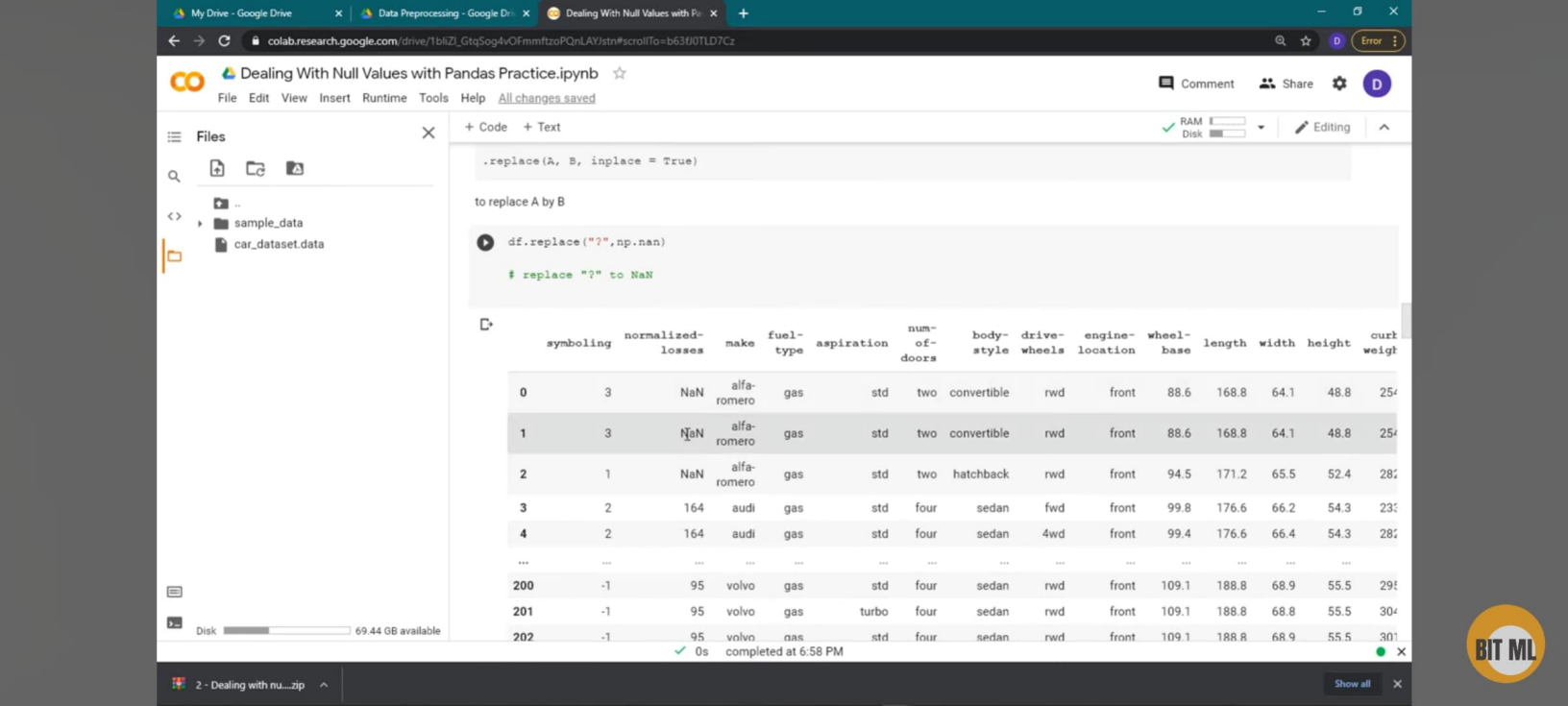
Steps for working with missing data:

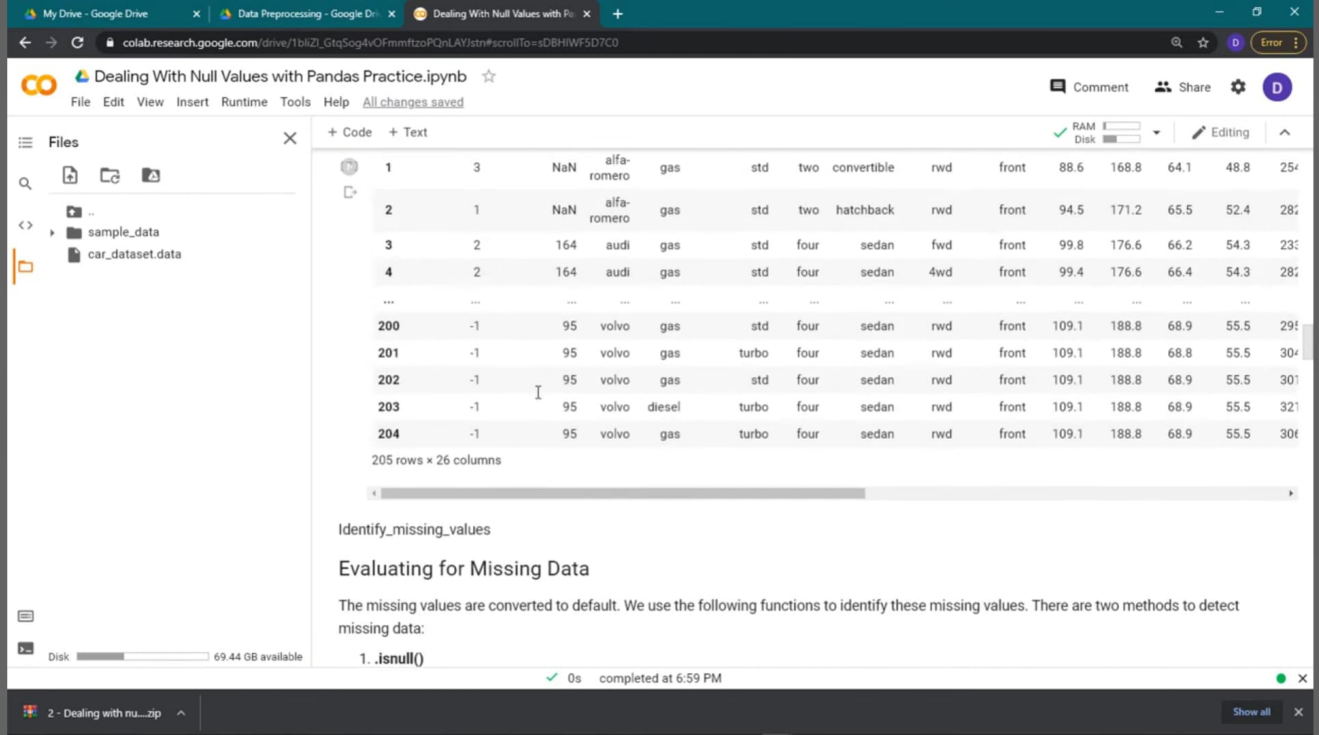
1. Identify missing data
2. deal with missing data
3. correct data format

Identify and handle missing values

Identify missing values

Convert "?" to NaN


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=b63fJ0TLD7Cz
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
D
Files
..
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
.replace(A, B, inplace = True)
to replace A by B
df.replace("?",np.nan)
# replace "?" to NaN
symboling normalized-losses make fuel-type aspiration num-of-doors body-style drive-wheels engine-location wheel-base length width height curb weigh
0 3 NaN alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 254
1 3 NaN alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 254
2 1 NaN alfa-romero gas std two hatchback rwd front 94.5 171.2 65.5 52.4 282
3 2 164 audi gas std four sedan fwd front 99.8 176.6 66.2 54.3 233
4 2 164 audi gas std four sedan 4wd front 99.4 176.6 66.4 54.3 282
... ...
200 -1 95 volvo gas std four sedan rwd front 109.1 188.8 68.9 55.5 295
201 -1 95 volvo gas turbo four sedan rwd front 109.1 188.8 68.8 55.5 304
202 -1 95 volvo gas std four sedan rwd front 109.1 188.8 68.9 55.5 301
0s completed at 6:58 PM
2 - Dealing with nu....zip
Show all
BIT ML



My Drive - Google Drive | Data Preprocessing - Google Dri | Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_Gtq5og4vOFmmftzoPQnLAYJstn#scrollTo=sDBHIWF5D7C0
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share D
Files
..
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM Disk Editing

1	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	254
2	1	NaN	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	282
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	233
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	282
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5	295
201	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.8	55.5	304
202	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5	301
203	-1	95	volvo	diesel	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5	321
204	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5	306

205 rows × 26 columns

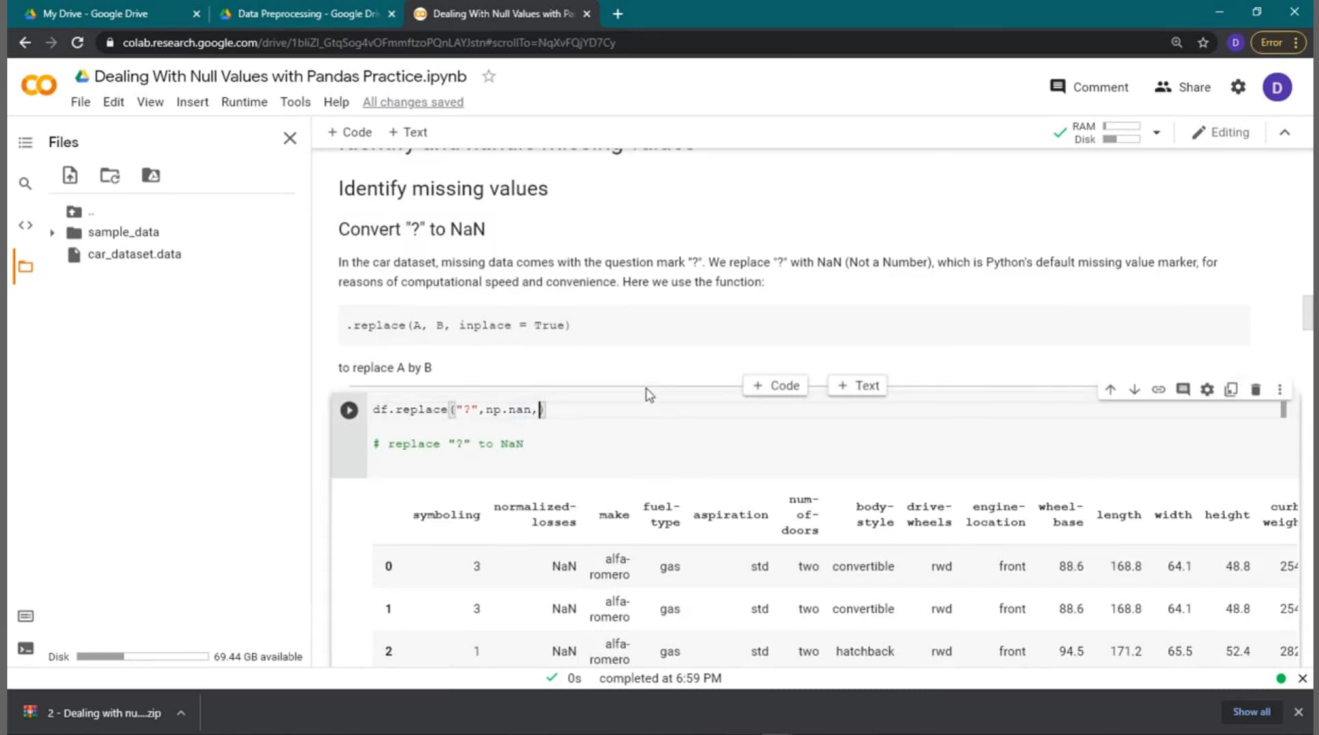
Identify_missing_values

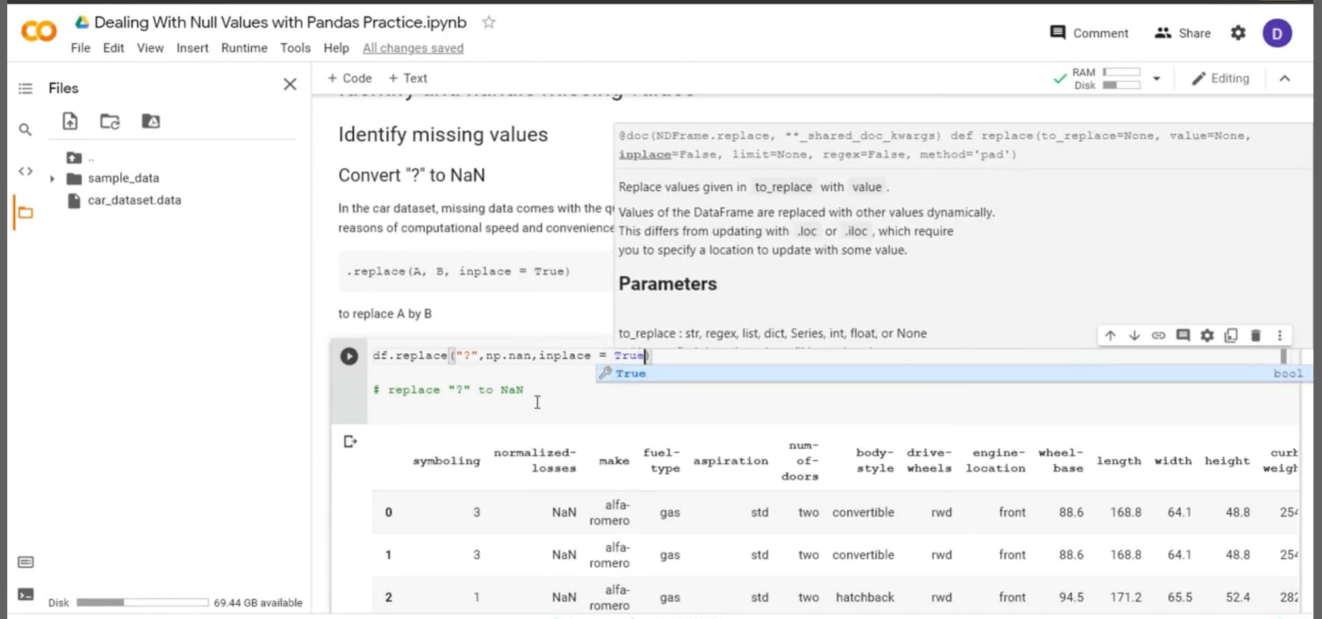
Evaluating for Missing Data

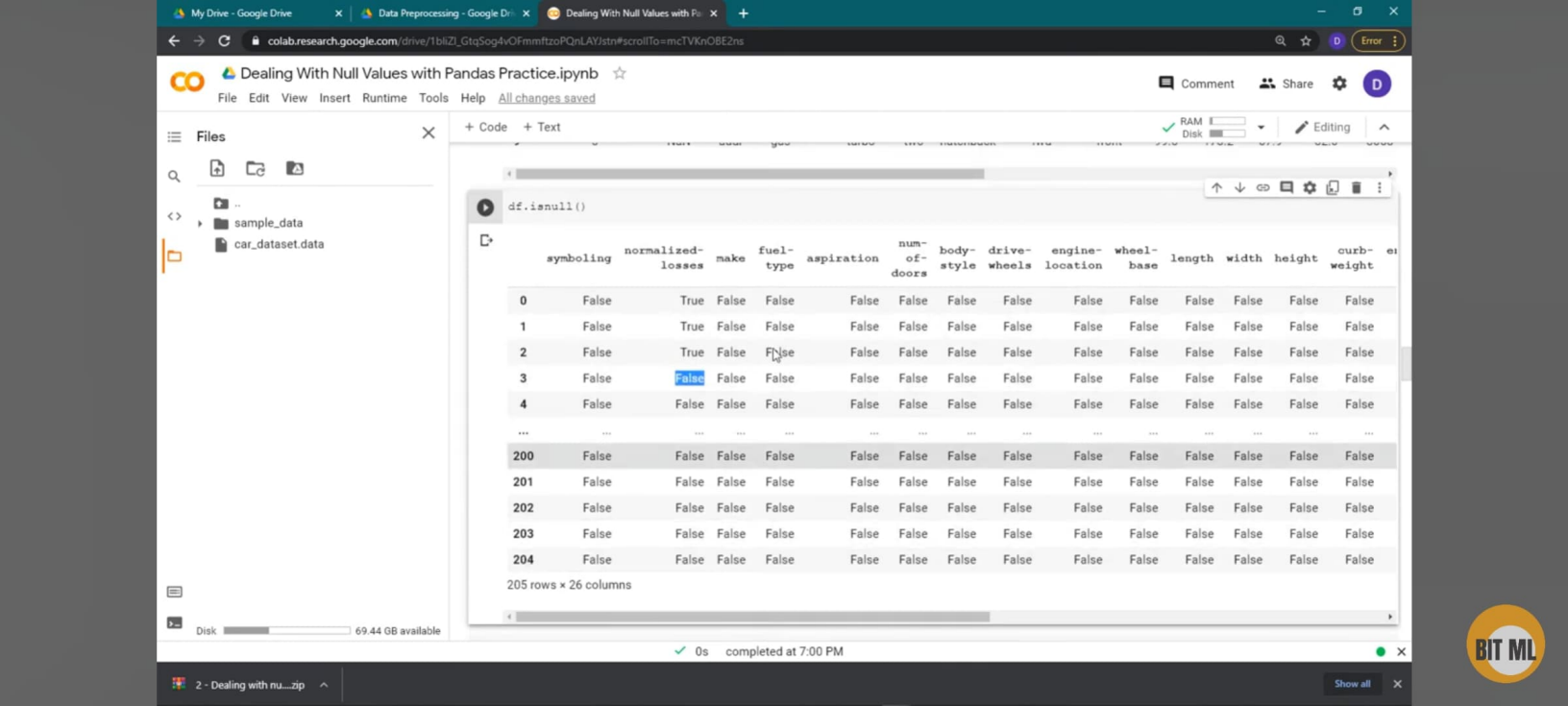
The missing values are converted to default. We use the following functions to identify these missing values. There are two methods to detect missing data:

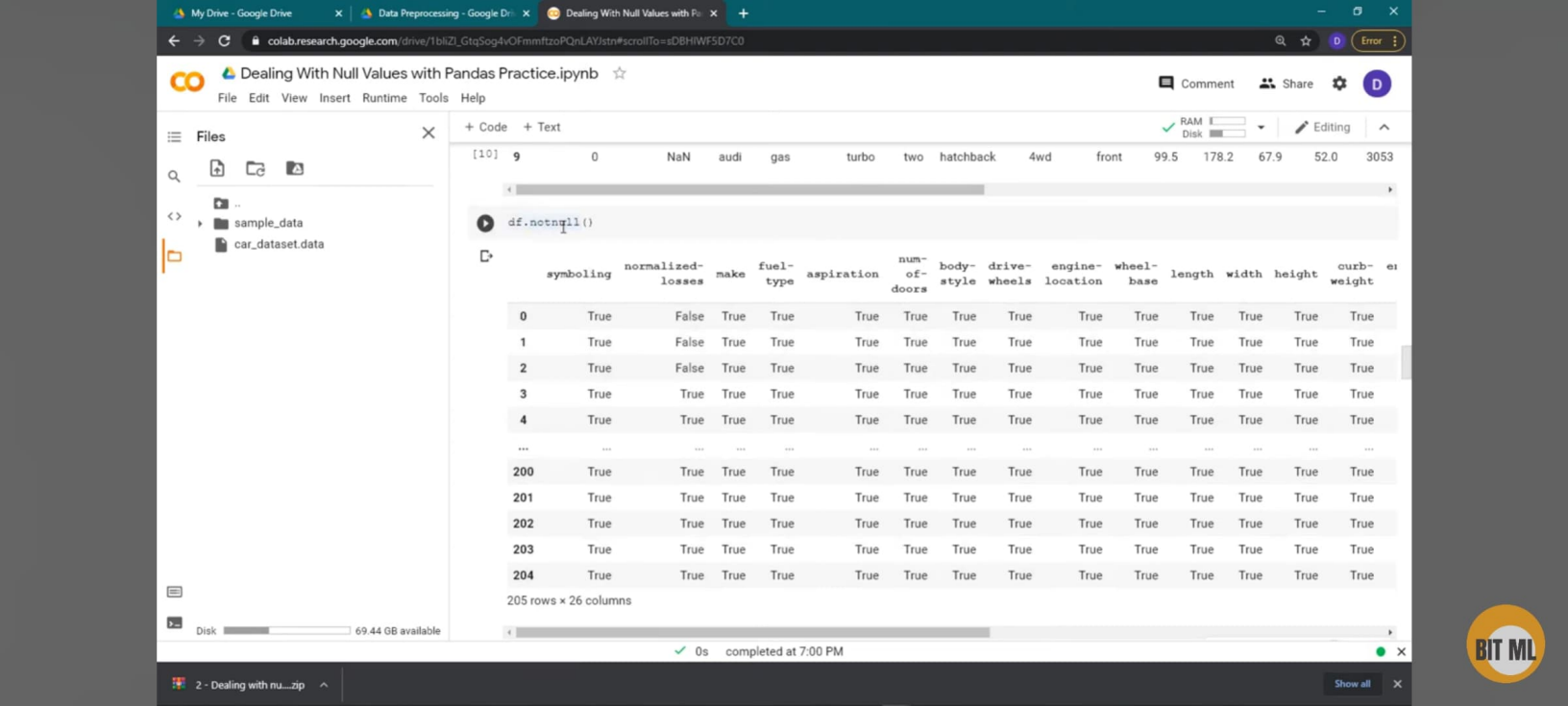
1. **.isnull()**

0s completed at 6:59 PM
2 - Dealing with nu....zip
Show all
BIT ML


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_Gtq5og4vOFmmftzoPQnLAYJstn#scrollTo=NqXvFQjYD7Cy
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
Identify missing values
Convert "?" to NaN
In the car dataset, missing data comes with the question mark "?". We replace "?" with NaN (Not a Number), which is Python's default missing value marker, for reasons of computational speed and convenience. Here we use the function:
.replace(A, B, inplace = True)
to replace A by B
+ Code + Text
df.replace("?",np.nan,
replace "?" to NaN
symboling normalized-losses make fuel-type aspiration num-of-doors body-style drive-wheels engine-location wheel-base length width height curb weigh
0 3 NaN alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 254
1 3 NaN alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 254
2 1 NaN alfa-romero gas std two hatchback rwd front 94.5 171.2 65.5 52.4 287
0s completed at 6:59 PM
2 - Dealing with nu....zip
Show all


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=NqXvFQjYD7Cy
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
D
Files
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
Identify missing values
Convert "?" to NaN
In the car dataset, missing data comes with the q
reasons of computational speed and convenience
.replace(A, B, inplace = True)
to replace A by B
@doc(NDFrame.replace, **_shared_doc_kwargs) def replace(to_replace=None, value=None, inplace=False, limit=None, regex=False, method='pad')
Replace values given in to_replace with value .
Values of the DataFrame are replaced with other values dynamically.
This differs from updating with .loc or .iloc , which require
you to specify a location to update with some value.
Parameters
to_replace : str, regex, list, dict, Series, int, float, or None
df.replace("?",np.nan,inplace = True)
# replace "?" to NaN
True bool
symboling normalized-losses make fuel-type aspiration num-of-doors body-style drive-wheels engine-location wheel-base length width height curb weigh
0 3 NaN alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 254
1 3 NaN alfa-romero gas std two convertible rwd front 88.6 168.8 64.1 48.8 254
2 1 NaN alfa-romero gas std two hatchback rwd front 94.5 171.2 65.5 52.4 282
0s completed at 6:59 PM
2 - Dealing with nu....zip
Show all


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=mcTVKnOBE2ns
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
D
Files
..
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
df.isnull()
symboling normalized-losses make fuel-type aspiration num-of-doors body-style drive-wheels engine-location wheel-base length width height curb-weight
0 False True False False False False False False False False False False False False
1 False True False False False False False False False False False False False False
2 False True False False False False False False False False False False False False
3 False False False False False False False False False False False False False False
4 False False False False False False False False False False False False False False
... ...
200 False False False False False False False False False False False False False False
201 False False False False False False False False False False False False False False
202 False False False False False False False False False False False False False False
203 False False False False False False False False False False False False False False
204 False False False False False False False False False False False False False False
205 rows × 26 columns
0s completed at 7:00 PM
2 - Dealing with nu....zip
Show all
BIT ML


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZI_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=sDBHWF5D7C0
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
D
Files
..
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
[10] 9 0 NaN audi gas turbo two hatchback 4wd front 99.5 178.2 67.9 52.0 3053
df.notnull()
symboling normalized-losses make fuel-type aspiration num-of-doors body-style drive-wheels engine-location wheel-base length width height curb-weight
0 True False True True True True True True True True True True True True
1 True False True True True True True True True True True True True True
2 True False True True True True True True True True True True True True
3 True True True True True True True True True True True True True True
4 True True True True True True True True True True True True True True
... ...
200 True True True True True True True True True True True True True True
201 True True True True True True True True True True True True True True
202 True True True True True True True True True True True True True True
203 True True True True True True True True True True True True True True
204 True True True True True True True True True True True True True True
205 rows × 26 columns
0s completed at 7:00 PM
2 - Dealing with nu....zip
Show all
BIT ML

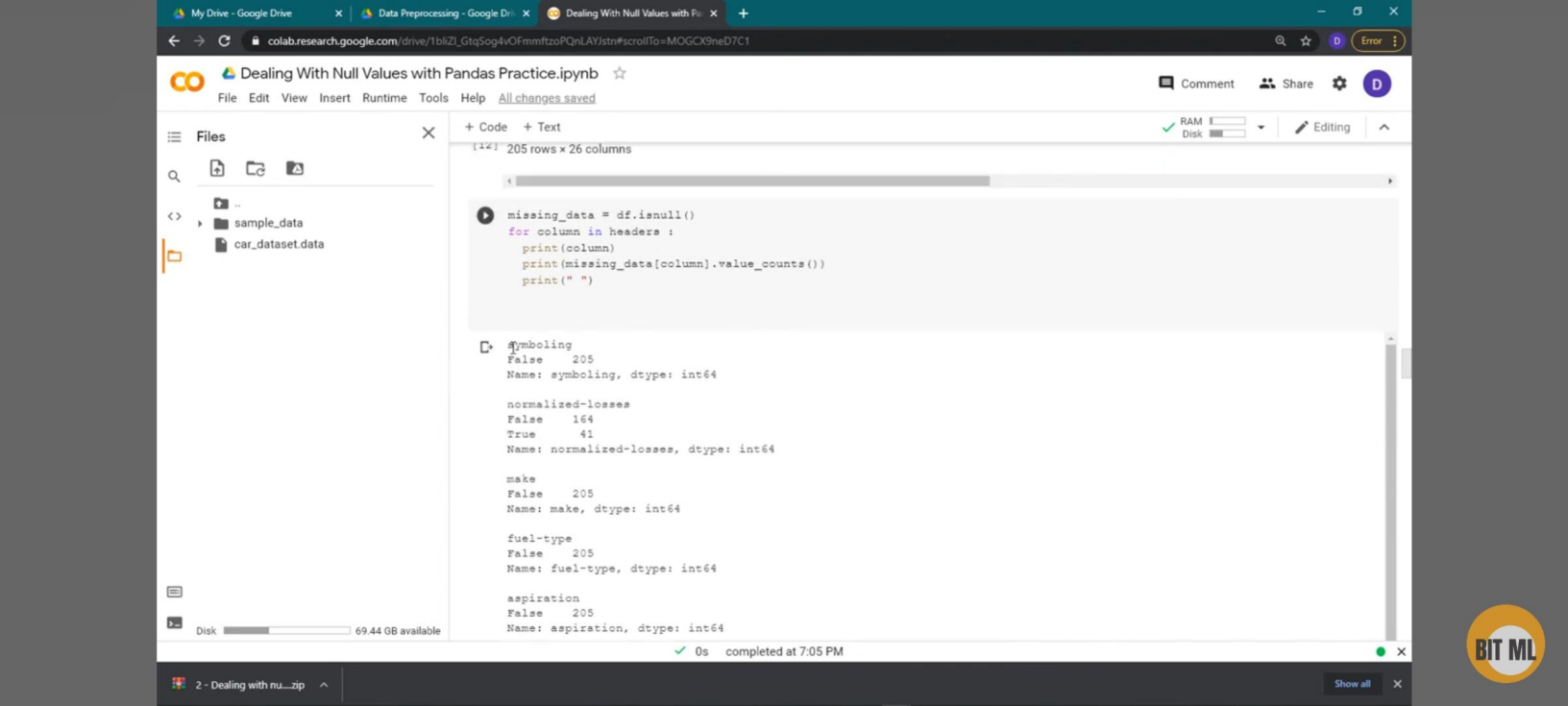
[12]

203	True	True	True	True	True	True	True	True	True	Tr
204	True	True	True	True	True	True	True	True	True	Tr

205 rows × 26 columns

```
missing_data = df.isnull()
for column in headers :
  print(column)
  print(missing_data[column].value_counts())
  print(" ")
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel ba
0	False	True	False	False	False	False	False	False	False	Fal
1	False	True	False	False	False	False	False	False	False	Fal
2	False	True	False	False	False	False	False	False	False	Fal
3	False	False	False	False	False	False	False	False	False	Fal


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZI_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=MOGCX9neD7C1
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
D
Files
..
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
[12] 205 rows × 26 columns
missing_data = df.isnull()
for column in headers :
print(column)
print(missing_data[column].value_counts())
print(" ")
symboling
False 205
Name: symboling, dtype: int64
normalized-losses
False 164
True 41
Name: normalized-losses, dtype: int64
make
False 205
Name: make, dtype: int64
fuel-type
False 205
Name: fuel-type, dtype: int64
aspiration
False 205
Name: aspiration, dtype: int64
0s completed at 7:05 PM
2 - Dealing with nu....zip
Show all
BIT ML

Based on the summary above, each column has 205 rows of data, seven columns containing missing data:

1. "normalized-losses": 41 missing data
2. "num-of-doors": 2 missing data
3. "bore": 4 missing data
4. "stroke" : 4 missing data
5. "horsepower": 2 missing data
6. "peak-rpm": 2 missing data
7. "price": 4 missing data

Deal with missing data

How to deal with missing data?

1. drop data
   a. drop the whole row
   b. drop the whole column
2. replace data
   a. replace it by mean
   b. replace it by frequency

How to deal with missing data

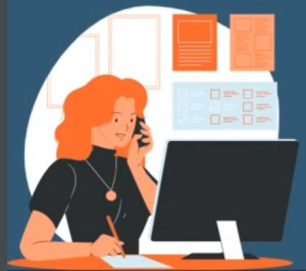
First : Drop the missing values

- drop the whole row
- drop the whole column

Second : Replace the missing values

- replace it with an average(Mean or Median)
- replace it by frequency(Mode)

Third : Leave it as missing data.



Now we have 7 columns containing missing data

Mean

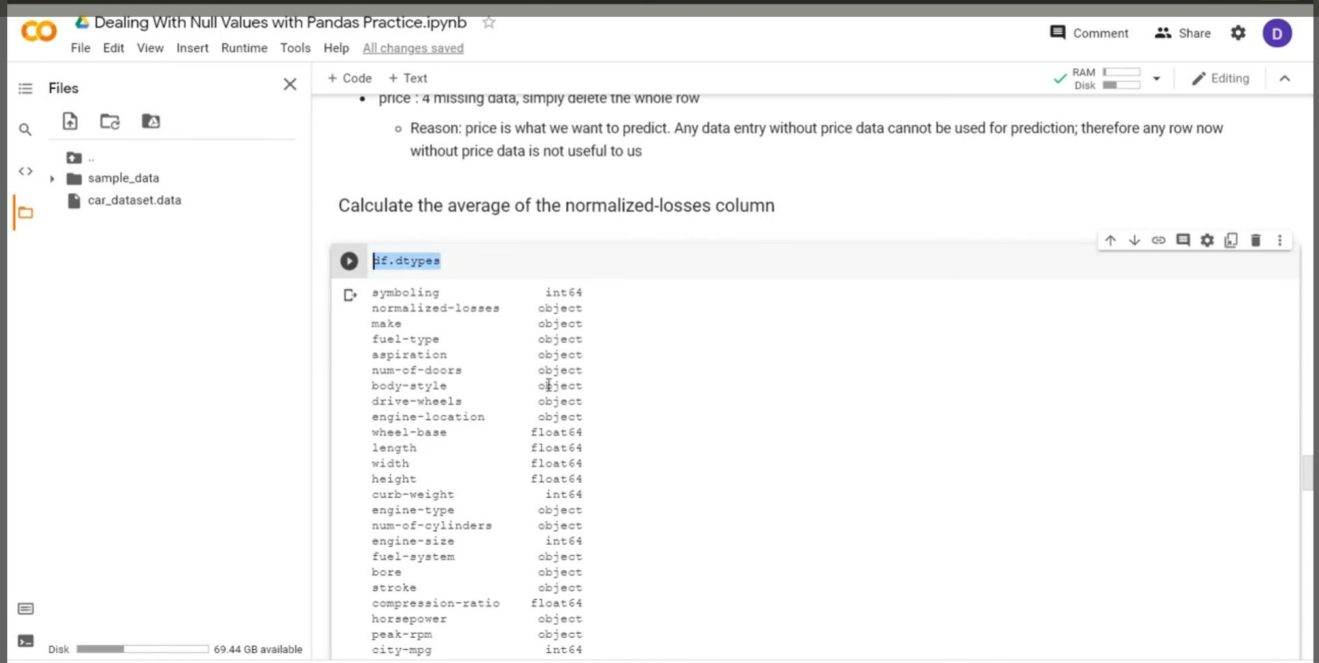
- normalized-losses
- bore
- stroke
- horsepower
- peak-rpm

Most Freq.

- num-of-doors

Remove

- price


Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
price : 4 missing data, simply delete the whole row
Reason: price is what we want to predict. Any data entry without price data cannot be used for prediction; therefore any row now without price data is not useful to us
Calculate the average of the normalized-losses column
df.dtypes
symboling int64
normalized-losses object
make object
fuel-type object
aspiration object
num-of-doors object
body-style object
drive-wheels object
engine-location object
wheel-base float64
length float64
width float64
height float64
curb-weight int64
engine-type object
num-of-cylinders object
engine-size int64
fuel-system object
bore object
stroke object
compression-ratio float64
horsepower object
peak-rpm object
city-mpg int64
0s completed at 7:13 PM

Files
..
sample_data
car_dataset.data
Disk 69.44 GB available

Replace by frequency:

- "num-of-doors": 2 missing data, replace them with "four".
  - Reason: 84% sedans is four doors. Since four doors is most frequent, it is most likely to occur

Drop the whole row:

- "price": 4 missing data, simply delete the whole row
  - Reason: price is what we want to predict. Any data entry without price data cannot be used for prediction; therefore any row now without price data is not useful to us

Calculate the average of the normalized-losses column

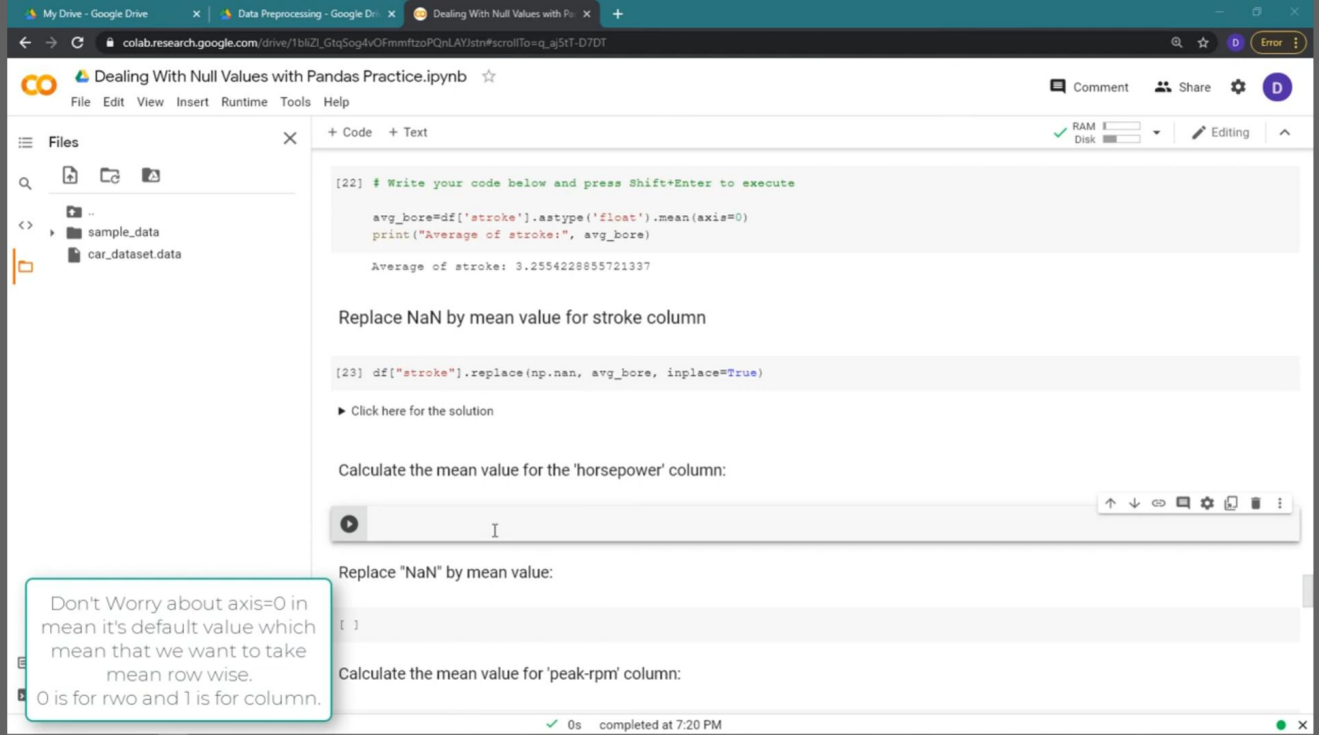
```
avg_nor_loss = df['normalized-losses'].astype("float").mean()
avg_nor_loss
```

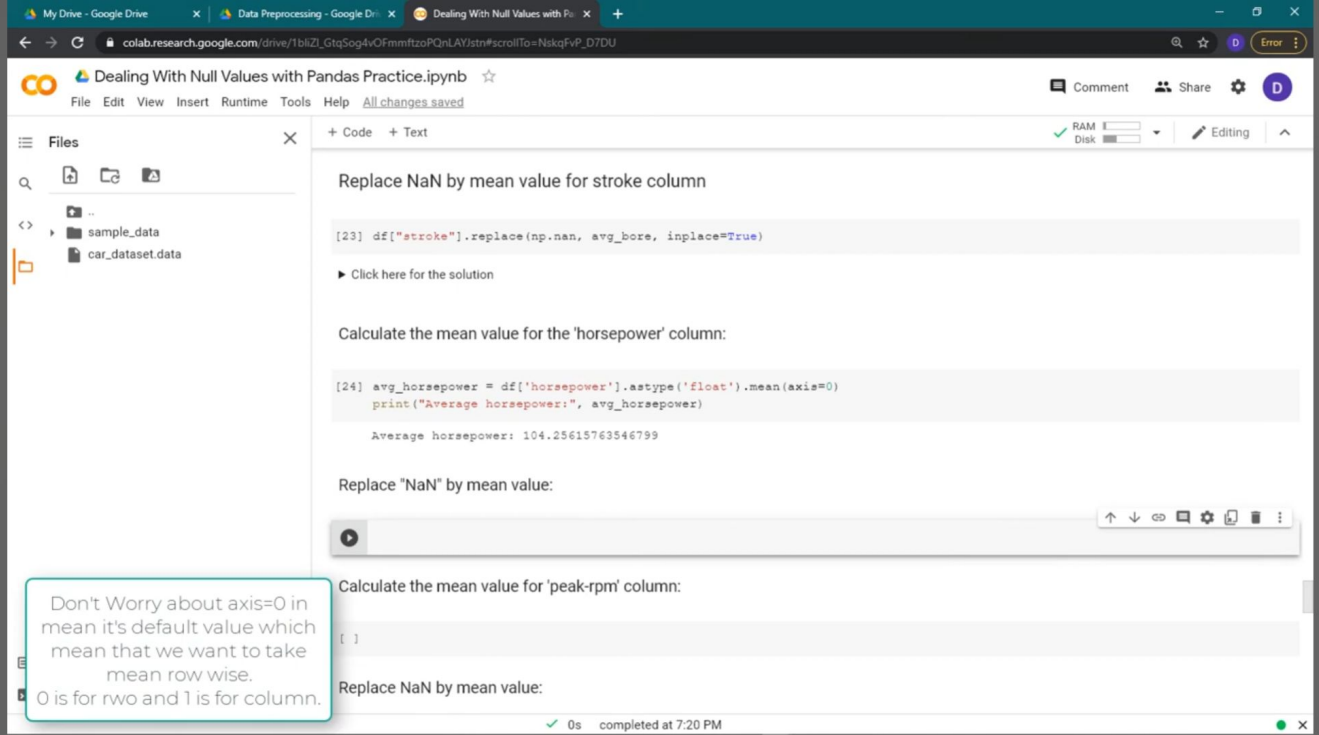
```
122.0
```

Replace "NaN" by mean value in "normalized-losses" column

```
[ ]
```

Calculate the mean value for 'bore' column


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=q_aj5tT-D7DT
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
D
Files
..
sample_data
car_dataset.data
+ Code + Text
RAM
Disk
Editing
[22] # Write your code below and press Shift+Enter to execute
avg_bore=df['stroke'].astype('float').mean(axis=0)
print("Average of stroke:", avg_bore)
Average of stroke: 3.2554228855721337
Replace NaN by mean value for stroke column
[23] df["stroke"].replace(np.nan, avg_bore, inplace=True)
▶ Click here for the solution
Calculate the mean value for the 'horsepower' column:
Replace "NaN" by mean value:
[ ]
Calculate the mean value for 'peak-rpm' column:
Don't Worry about axis=0 in mean it's default value which mean that we want to take mean row wise.
0 is for rwo and 1 is for column.
0s completed at 7:20 PM
BIT ML


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=NskqFvP_D7DU
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
D
Files
..
sample_data
car_dataset.data
+ Code + Text
RAM
Disk
Editing
Replace NaN by mean value for stroke column
[23] df["stroke"].replace(np.nan, avg_bore, inplace=True)
Click here for the solution
Calculate the mean value for the 'horsepower' column:
[24] avg_horsepower = df['horsepower'].astype('float').mean(axis=0)
print("Average horsepower:", avg_horsepower)
Average horsepower: 104.25615763546799
Replace "NaN" by mean value:
Calculate the mean value for 'peak-rpm' column:
[ ]
Replace NaN by mean value:
Don't Worry about axis=0 in mean it's default value which mean that we want to take mean row wise.
0 is for rwo and 1 is for column.
0s completed at 7:20 PM

+ Code + Text

RAM
Disk

```
[27] df['peak-rpm'].replace(np.nan, avg_peakrpm, inplace=True)
```

Alright you have done great job till now, but its not finished yet dude!!!

- Now we have done for replacing with mean.
- Its time to replace null values with most frequent values
- We will do this for num-of-doors column

Find the most frequent value for num of doors

```
df['num-of-doors'].value_counts()
```

```
four     114
two       89
Name: num-of-doors, dtype: int64
```

We can see that four doors are the most common type. We can also use the ".idxmax()" method to calculat
common type automatically:

+ Code + Text

RAM Disk

Alright you have done great job till now, but its not finished yet dude!!!

- Now we have done for replacing with mean.
- Its time to replace null values with most frequent values
- We will do this for num-of-doors column

Find the most frequent value for num of doors

```
df['num-of-doors'].value_counts()
df['body-style'].value_counts()
```

```
sedan          96
hatchback      70
wagon          25
hardtop         8
convertible     6
Name: body-style, dtype: int64
```

We can see that four doors are the most common type. We can also use the ".idxmax()" method to calculat
common type automatically:

+ Code + Text

RAM
Disk

```
convertible     6
Name: body-style, dtype: int64
```

We can see that four doors are the most common type. We can also use the ".idxmax()" method to calculat
common type automatically:

```
[33] df['num-of-doors'].value_counts().idxmax()
```

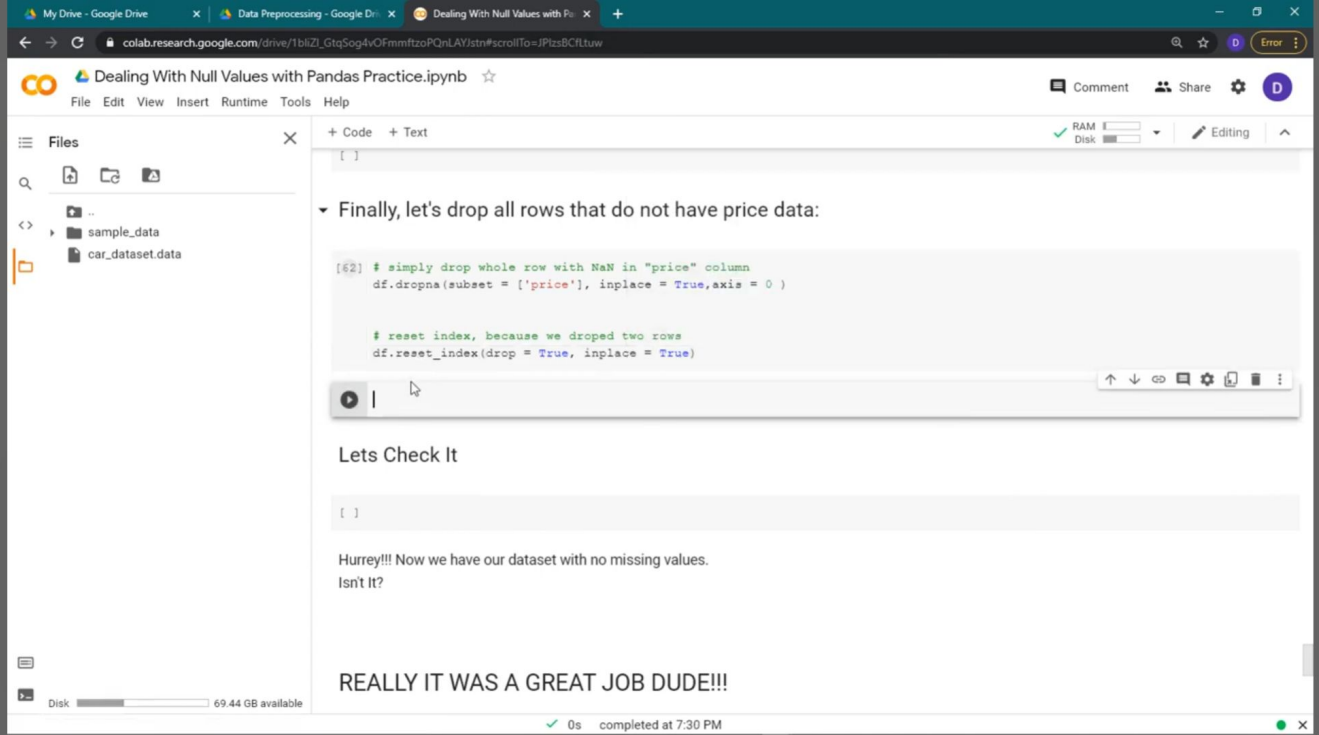
```
'four'
```

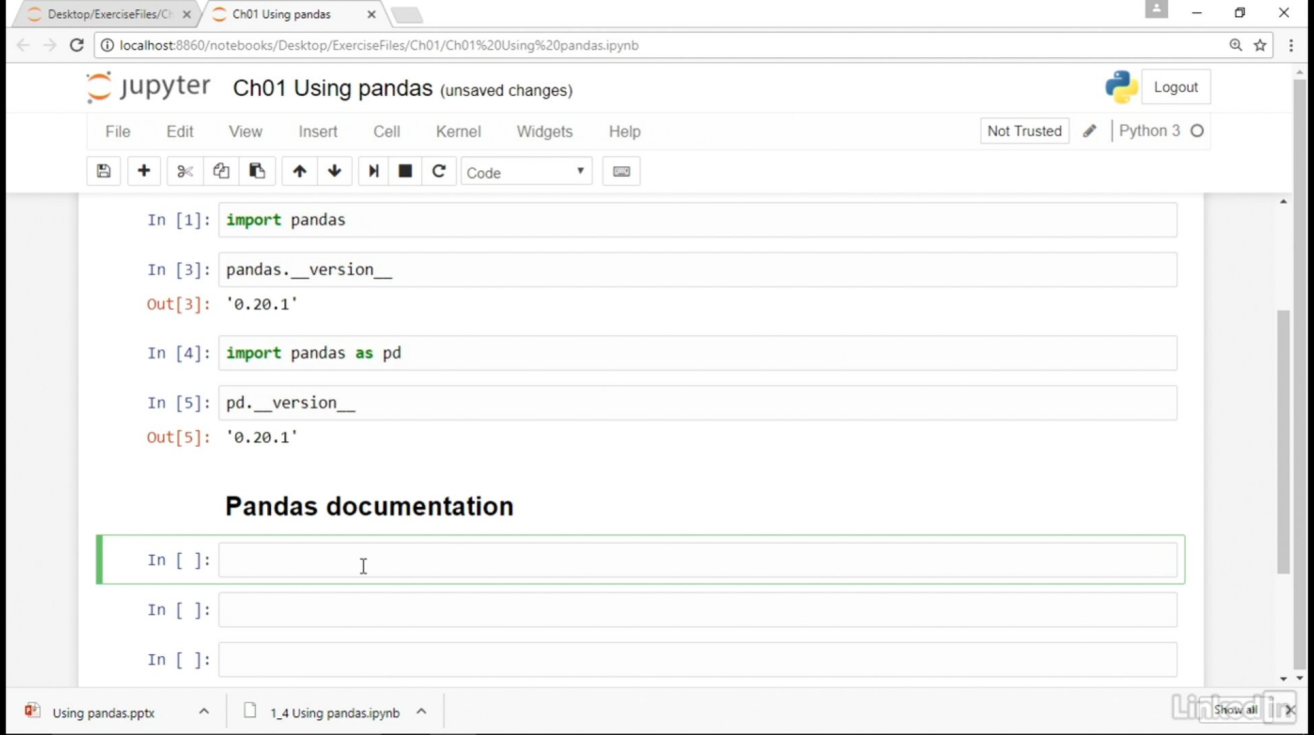
The replacement procedure is very similar to what we have seen previously

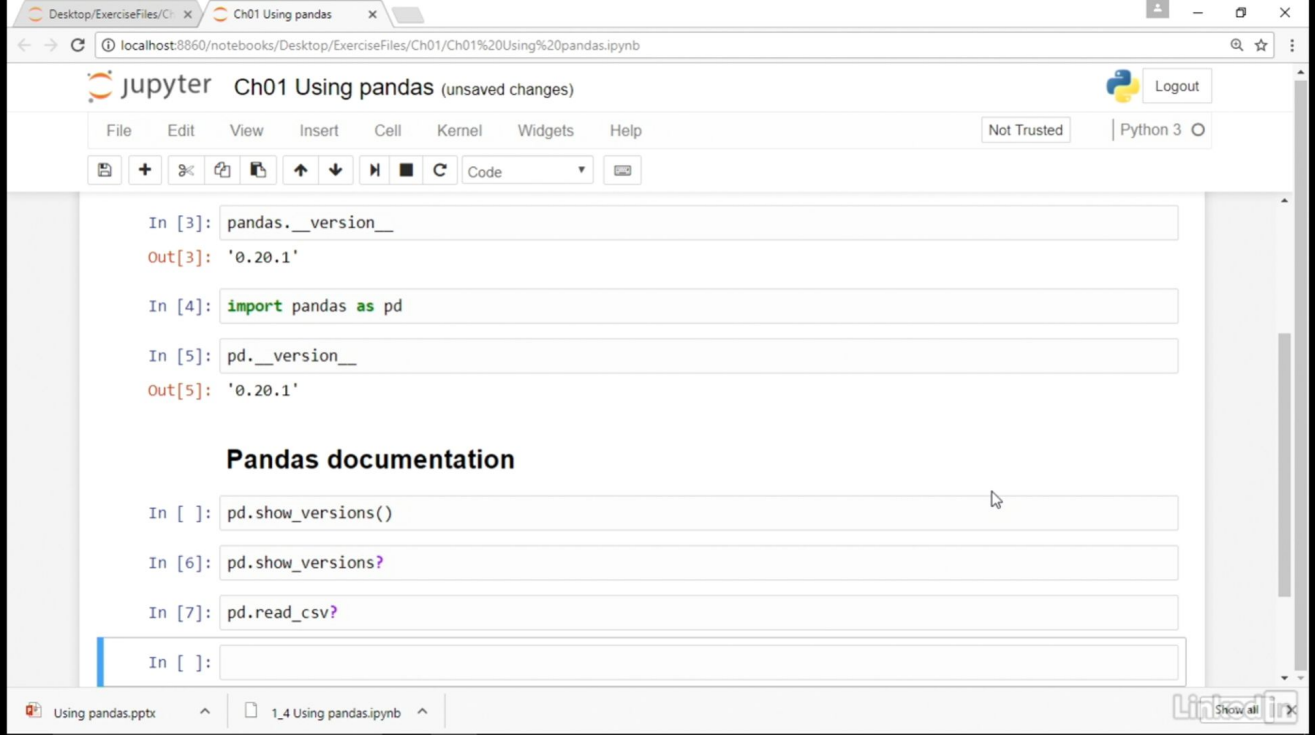
```
#replace the missing 'num-of-doors' values by the most frequent
```

Finally, let's drop all rows that do not have price data:

```
[ ] # simply drop whole row with NaN in "price" column


    # reset index, because we droped two rows
```


My Drive - Google Drive
Data Preprocessing - Google Dri
Dealing With Null Values with Pa
colab.research.google.com/drive/1bliZl_GtqSog4vOFmmftzoPQnLAYJstn#scrollTo=JPlzsBCfLtuw
Error
Dealing With Null Values with Pandas Practice.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
D
Files
..
sample_data
car_dataset.data
Disk 69.44 GB available
+ Code + Text
RAM
Disk
Editing
[ ]
Finally, let's drop all rows that do not have price data:
[62] # simply drop whole row with NaN in "price" column
df.dropna(subset = ['price'], inplace = True,axis = 0)
# reset index, because we droped two rows
df.reset_index(drop = True, inplace = True)
Lets Check It
[ ]
Hurrey!!! Now we have our dataset with no missing values.
Isn't It?
REALLY IT WAS A GREAT JOB DUDE!!!
0s completed at 7:30 PM


Desktop/ExerciseFiles/Ch
Ch01 Using pandas
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch01/Ch01%20Using%20pandas.ipynb
jupyter
Ch01 Using pandas (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Code
In [1]: import pandas
In [3]: pandas.__version__
Out[3]: '0.20.1'
In [4]: import pandas as pd
In [5]: pd.__version__
Out[5]: '0.20.1'
Pandas documentation
In [ ]:
In [ ]:
In [ ]:
Using pandas.pptx
1_4 Using pandas.ipynb
Show all


Desktop/ExerciseFiles/Ch
Ch01 Using pandas
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch01/Ch01%20Using%20pandas.ipynb
jupyter
Ch01 Using pandas (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Code
In [3]:
pandas.__version__
Out[3]:
'0.20.1'
In [4]:
import pandas as pd
In [5]:
pd.__version__
Out[5]:
'0.20.1'
Pandas documentation
In [ ]:
pd.show_versions()
In [6]:
pd.show_versions?
In [7]:
pd.read_csv?
In [ ]:
Using pandas.pptx
1_4 Using pandas.ipynb
Show all

DataFrames

- A DataFrame is like a two-dimensional array

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_Gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver
5	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	1200m freestyle	M	Bronze
6	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200m freestyle	M	Gold
7	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200m freestyle	M	Silver
8	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	400m freestyle	M	Bronze
9	Athens	1896	Aquatics	Swimming	NEUMANN, Paul	AUT	Men	400m freestyle	M	Gold
10	Athens	1896	Aquatics	Swimming	PEPANOS, Antonios	GRE	Men	400m freestyle	M	Silver
11	Athens	1896	Athletics	Athletics	LANE, Francis	USA	Men	100m	M	Bronze

- A DataFrame is a sequence of series that share the same index

Series

- A Series is a one-dimensional array of indexed data

	City	Edition	Sport	Discipline	Athlete	...	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred		Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto		Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios		Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis		Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon		Silver
5	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios		Bronze
6	Athens	1896	Aquatics	Swimming	HAJOS, Alfred		Gold
7	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis		Silver
8	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios		Bronze
9	Athens	1896	Aquatics	Swimming	NEUMANN, Paul		Gold
10	Athens	1896	Aquatics	Swimming	PEPANOS, Antonios		Silver
11	Athens	1896	Athletics	Athletics	LANE, Francis		Bronze

Series

- Accessing a single Series

  DataFrame['SeriesName'] or
  DataFrame["SeriesName"]

  DataFrame.SeriesName

- Accessing multiple Series

  DataFrame[['SeriesName1','SeriesName2']]

On the Job Tip

- type(DataFrame)

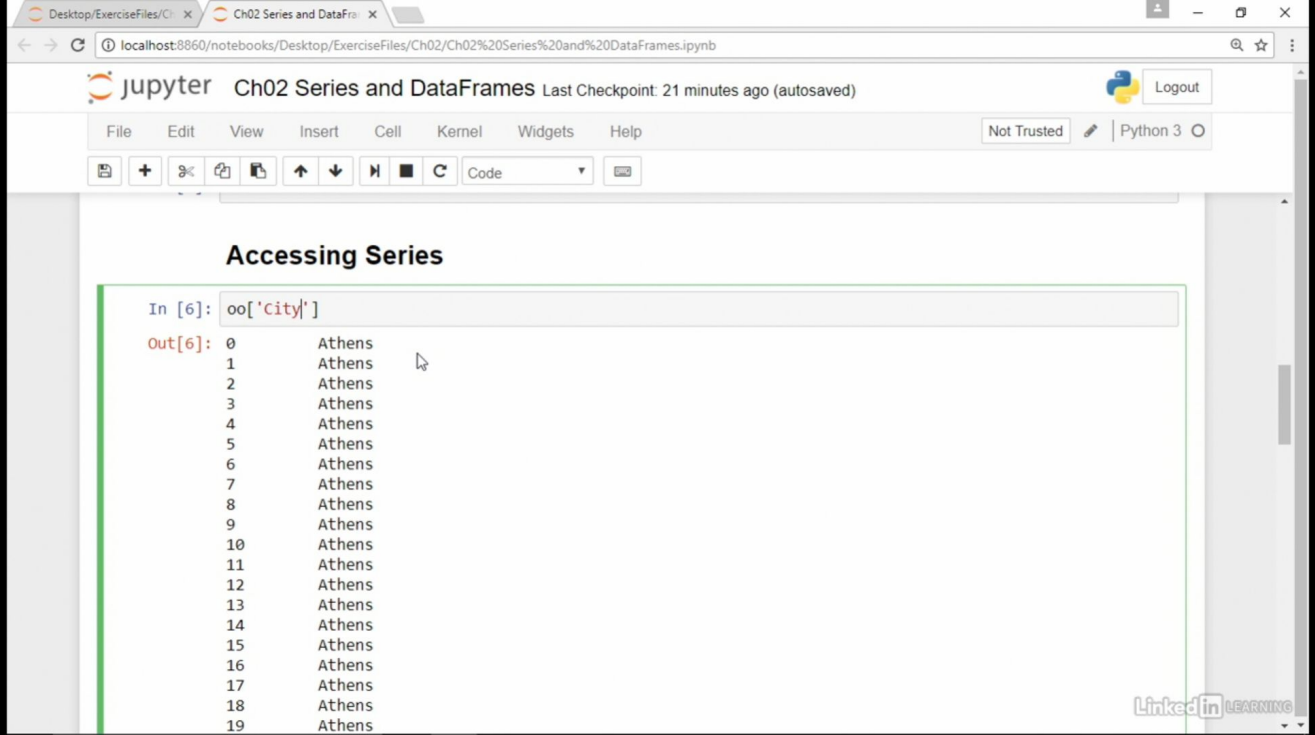
  pandas.core.frame.DataFrame

- type(DataFrame.SeriesName)

  pandas.core.series.Series

- type(DataFrame[['SeriesName1,'1SeriesName2']])

  pandas.core.frame.DataFrame


Desktop/ExerciseFiles/Ch
Ch02 Series and DataFra
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch02/Ch02%20Series%20and%20DataFrames.ipynb
jupyter
Ch02 Series and DataFrames Last Checkpoint: 21 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Code
Accessing Series
In [6]: oo['City']
Out[6]: 0 Athens
1 Athens
2 Athens
3 Athens
4 Athens
5 Athens
6 Athens
7 Athens
8 Athens
9 Athens
10 Athens
11 Athens
12 Athens
13 Athens
14 Athens
15 Athens
16 Athens
17 Athens
18 Athens
19 Athens
LinkedIn LEARNING

```
29212          MIZGAITIS, Mindaugas
29213               PATRIKEEV, Yuri
29214                 LOPEZ, Mijain
29215                  BAROEV, Khasan
Name: Athlete, Length: 29216, dtype: object
```

In [9]:
```
oo[['City','Edition','Athlete']]
```

Out[9]:

	City	Edition	Athlete
0	Athens	1896	HAJOS, Alfred
1	Athens	1896	HERSCHMANN, Otto
2	Athens	1896	DRIVAS, Dimitrios
3	Athens	1896	MALOKINIS, Ioannis
4	Athens	1896	CHASAPIS, Spiridon
5	Athens	1896	CHOROPHAS, Efstathios
6	Athens	1896	HAJOS, Alfred
7	Athens	1896	ANDREOU, Joannis
8	Athens	1896	CHOROPHAS, Efstathios
9	Athens	1896	NEUMANN, Paul
10	Athens	1896	PEPANOS, Antonios

jupyter Ch02 Series and DataFrames Last Checkpoint: 23 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Code

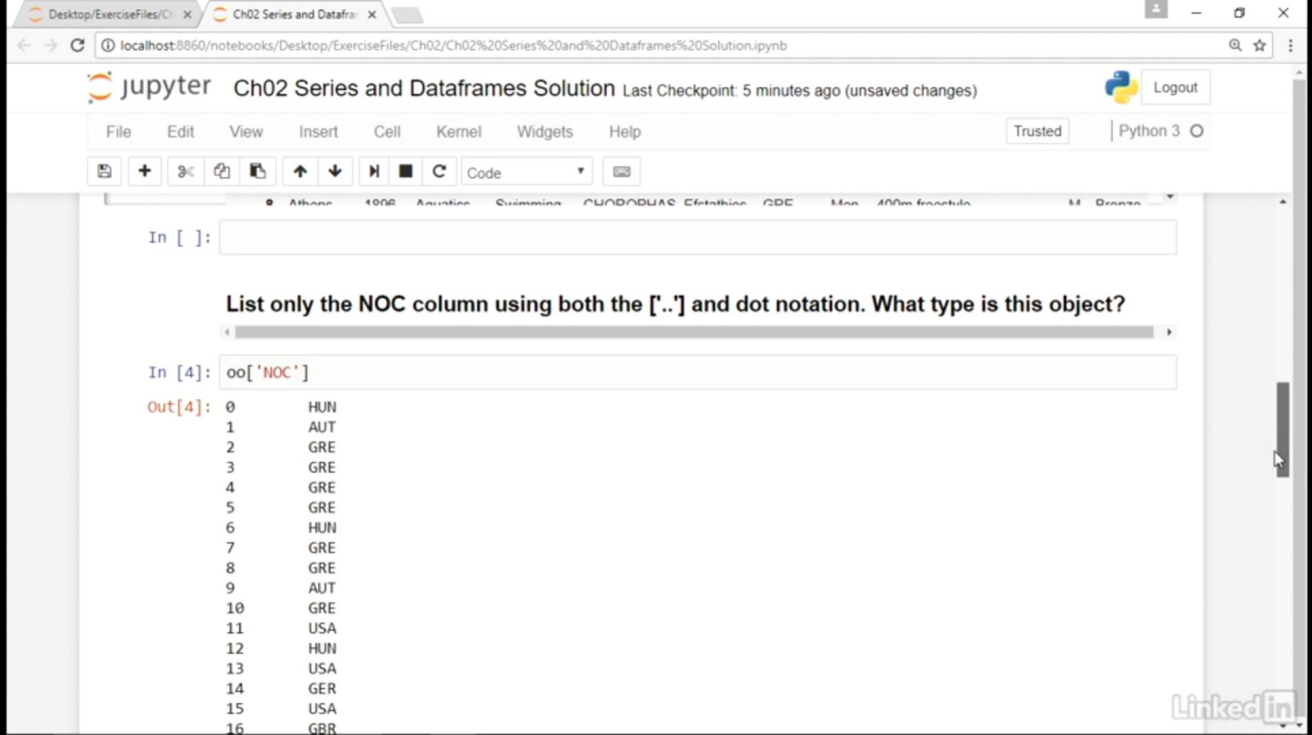
29208	Beijing	2008	MAMBETOV, Asset
29209	Beijing	2008	WHEELER, Adam
29210	Beijing	2008	KHUSHTOV, Aslanbek
29211	Beijing	2008	ENGLICH, Mirko
29212	Beijing	2008	MIZGAITIS, Mindaugas
29213	Beijing	2008	PATRIKEEV, Yuri
29214	Beijing	2008	LOPEZ, Mijain
29215	Beijing	2008	BAROEV, Khasan

29216 rows × 3 columns

```
In [10]: type(oo)
Out[10]: pandas.core.frame.DataFrame

In [11]: type(oo.City)
Out[11]: pandas.core.series.Series

In [ ]:

In [ ]:
```

Challenge

- List our Olympics DataFrame.
- List only the NOC column using both the ['..'] and dot notation. What type is this object?
- List the Edition, City, Athlete Name, and Medal columns. What type is this object?


Desktop/ExerciseFiles/Ch
Ch02 Series and Datafra
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch02/Ch02%20Series%20and%20Dataframes%20Solution.ipynb
jupyter
Ch02 Series and Dataframes Solution
Last Checkpoint: 5 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
In [ ]:
List only the NOC column using both the ['..'] and dot notation. What type is this object?
In [4]:
oo['NOC']
Out[4]:
0 HUN
1 AUT
2 GRE
3 GRE
4 GRE
5 GRE
6 HUN
7 GRE
8 GRE
9 AUT
10 GRE
11 USA
12 HUN
13 USA
14 GER
15 USA
16 GBR

jupyter Ch02 Series and Dataframes Solution Last Checkpoint: 7 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Markdown

```
29197    BLR
29198    UKR
29199    FRA
29200    KGZ
29201    FRA
29202    BUL
29203    GEO
29204    CHN
29205    TUR
29206    ITA
29207    HUN
29208    KAZ
29209    USA
29210    RUS
29211    GER
29212    LTU
29213    ARM
29214    CUB
29215    RUS
Name: NOC, Length: 29216, dtype: object
```

In [7]:
```
type(oo.NOC)
```

Out[7]:
```
pandas.core.series.Series
```

List the Edition, City, Athlete Name and Medal columns. What type is this object?

```
29213    ARM
29214    CUB
29215    RUS
Name: NOC, Length: 29216, dtype: object
```

In [7]:
```
type(oo.NOC)
```

Out[7]:
```
pandas.core.series.Series
```

List the Edition, City, Athlete Name and Medal columns. What type is this object?

In [8]:
```
oo[['Edition','City','Athlete','Medal']]
```

Out[8]:

	Edition	City	Athlete	Medal
0	1896	Athens	HAJOS, Alfred	Gold
1	1896	Athens	HERSCHMANN, Otto	Silver
2	1896	Athens	DRIVAS, Dimitrios	Bronze
3	1896	Athens	MALOKINIS, Ioannis	Gold
4	1896	Athens	CHASAPIS, Spiridon	Silver
5	1896	Athens	CHOROPHAS, Efstathios	Bronze

Data Input

- Input

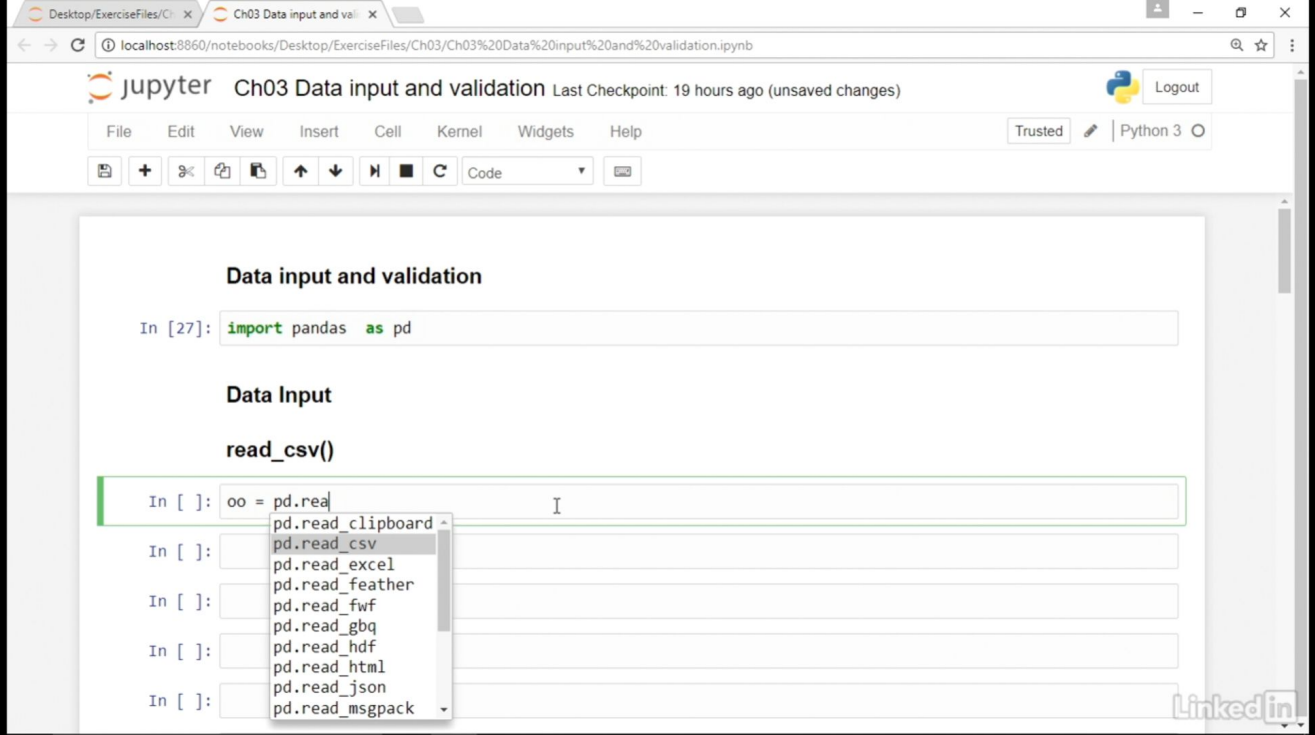
  read_excel(…)

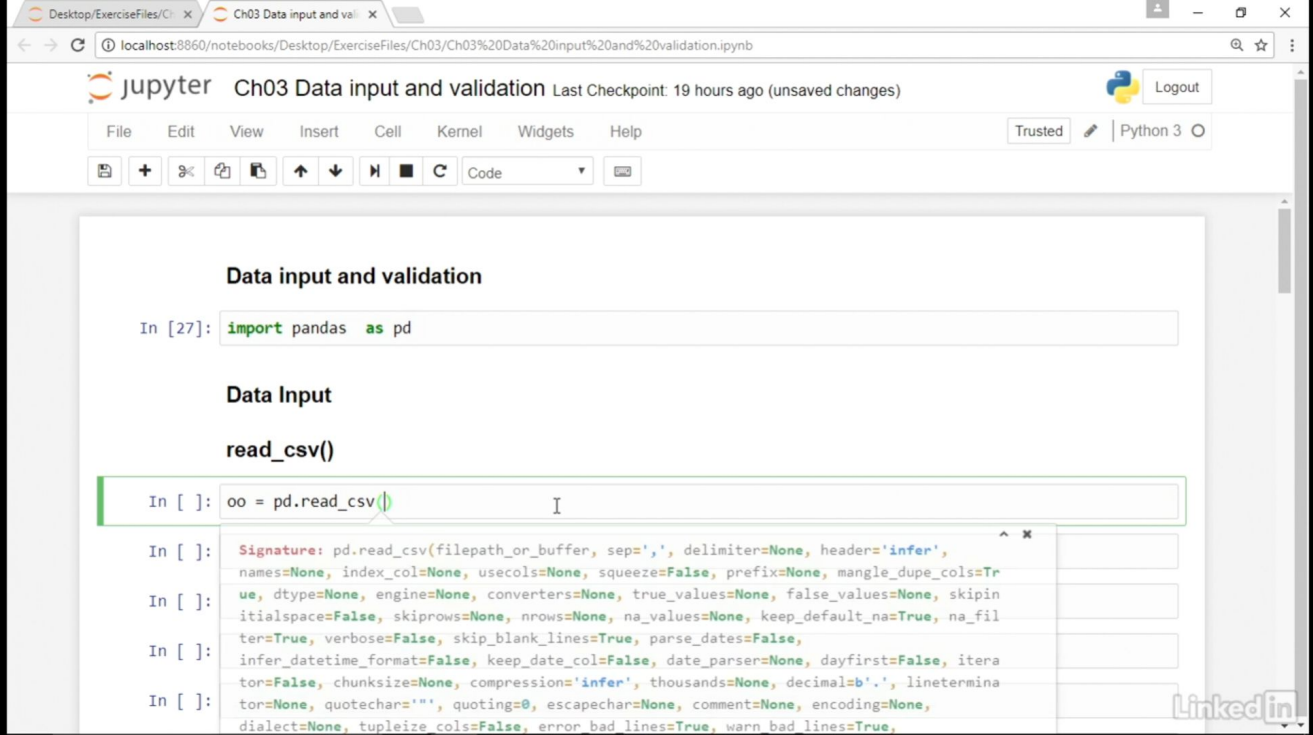
  read_json(…)

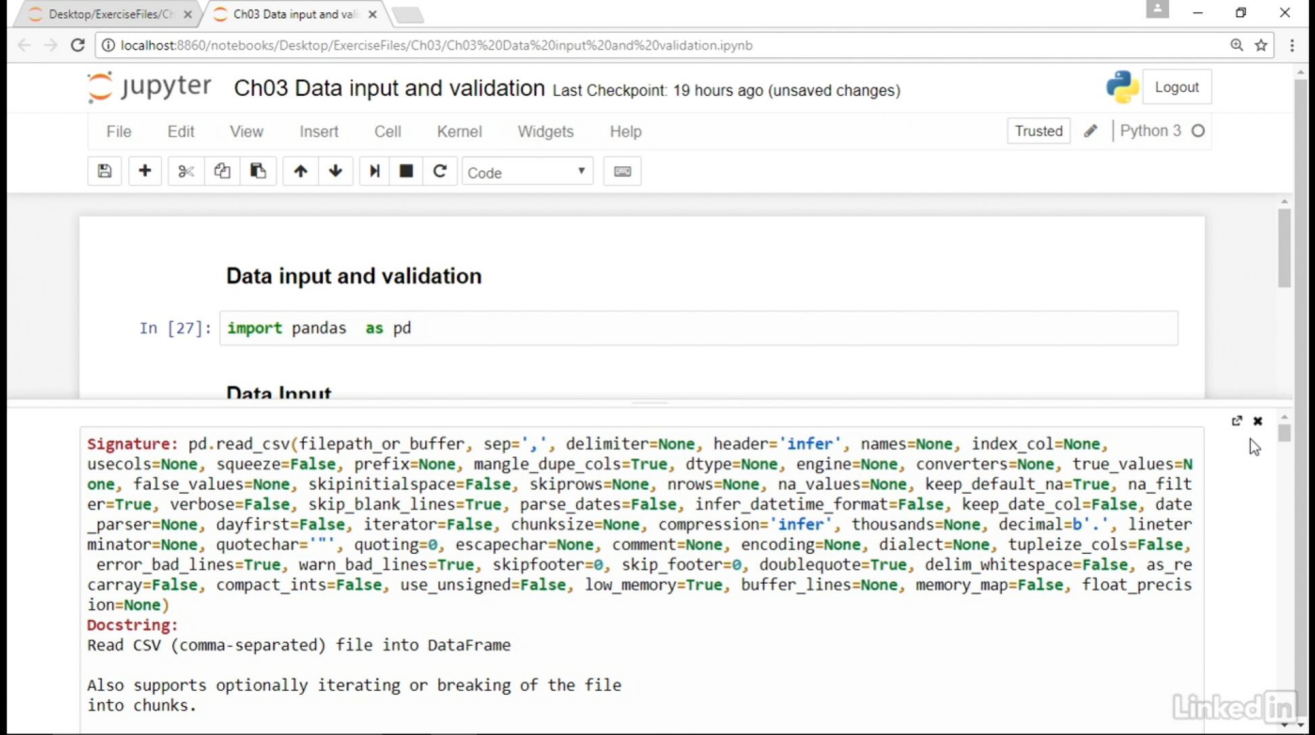
  read_sql_table(…)

- Read a CSV file into a DataFrame

- pandas.read_csv(*filepath*, …, *skiprows=None*, ..)


Desktop/ExerciseFiles/Ch
Ch03 Data input and vali
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch03/Ch03%20Data%20input%20and%20validation.ipynb
jupyter
Ch03 Data input and validation
Last Checkpoint: 19 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
Data input and validation
In [27]: import pandas as pd
Data Input
read_csv()
In [ ]: oo = pd.rea
pd.read_clipboard
pd.read_csv
pd.read_excel
pd.read_feather
pd.read_fwf
pd.read_gbq
pd.read_hdf
pd.read_html
pd.read_json
pd.read_msgpack
In [ ]:
In [ ]:
In [ ]:
In [ ]:
LinkedIn


Desktop/ExerciseFiles/Ch
Ch03 Data input and val
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch03/Ch03%20Data%20input%20and%20validation.ipynb
jupyter
Ch03 Data input and validation
Last Checkpoint: 19 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
Data input and validation
In [27]:
import pandas as pd
Data Input
read_csv()
In [ ]:
oo = pd.read_csv()
In [ ]:
In [ ]:
In [ ]:
In [ ]:
Signature: pd.read_csv(filepath_or_buffer, sep=',', delimiter=None, header='infer',
names=None, index_col=None, usecols=None, squeeze=False, prefix=None, mangle_dupe_cols=Tr
ue, dtype=None, engine=None, converters=None, true_values=None, false_values=None, skipin
itialspace=False, skiprows=None, nrows=None, na_values=None, keep_default_na=True, na_fil
ter=True, verbose=False, skip_blank_lines=True, parse_dates=False,
infer_datetime_format=False, keep_date_col=False, date_parser=None, dayfirst=False, itera
tor=False, chunksize=None, compression='infer', thousands=None, decimal=b'.', linetermina
tor=None, quotechar='"', quoting=0, escapechar=None, comment=None, encoding=None,
dialect=None, tupleize_cols=False, error_bad_lines=True, warn_bad_lines=True,


Desktop/ExerciseFiles/Ch
Ch03 Data input and val
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch03/Ch03%20Data%20input%20and%20validation.ipynb
jupyter
Ch03 Data input and validation
Last Checkpoint: 19 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
Data input and validation
In [27]: import pandas as pd
Data Input
Signature: pd.read_csv(filepath_or_buffer, sep=',', delimiter=None, header='infer', names=None, index_col=None,
usecols=None, squeeze=False, prefix=None, mangle_dupe_cols=True, dtype=None, engine=None, converters=None, true_values=N
one, false_values=None, skipinitialspace=False, skiprows=None, nrows=None, na_values=None, keep_default_na=True, na_filt
er=True, verbose=False, skip_blank_lines=True, parse_dates=False, infer_datetime_format=False, keep_date_col=False, date
_parser=None, dayfirst=False, iterator=False, chunksize=None, compression='infer', thousands=None, decimal=b'.', lineter
minator=None, quotechar='"', quoting=0, escapechar=None, comment=None, encoding=None, dialect=None, tupleize_cols=False,
error_bad_lines=True, warn_bad_lines=True, skipfooter=0, skip_footer=0, doublequote=True, delim_whitespace=False, as_re
carray=False, compact_ints=False, use_unsigned=False, low_memory=True, buffer_lines=None, memory_map=False, float_precis
ion=None)
Docstring:
Read CSV (comma-separated) file into DataFrame
Also supports optionally iterating or breaking of the file
into chunks.
Linked in

Desktop/ExerciseFiles/Ch | Ch03 Data input and vali

localhost:8860/notebooks/Desktop/ExerciseFiles/Ch03/Ch03%20Data%20input%20and%20validation.ipynb

jupyter Ch03 Data input and validation Last Checkpoint: 19 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

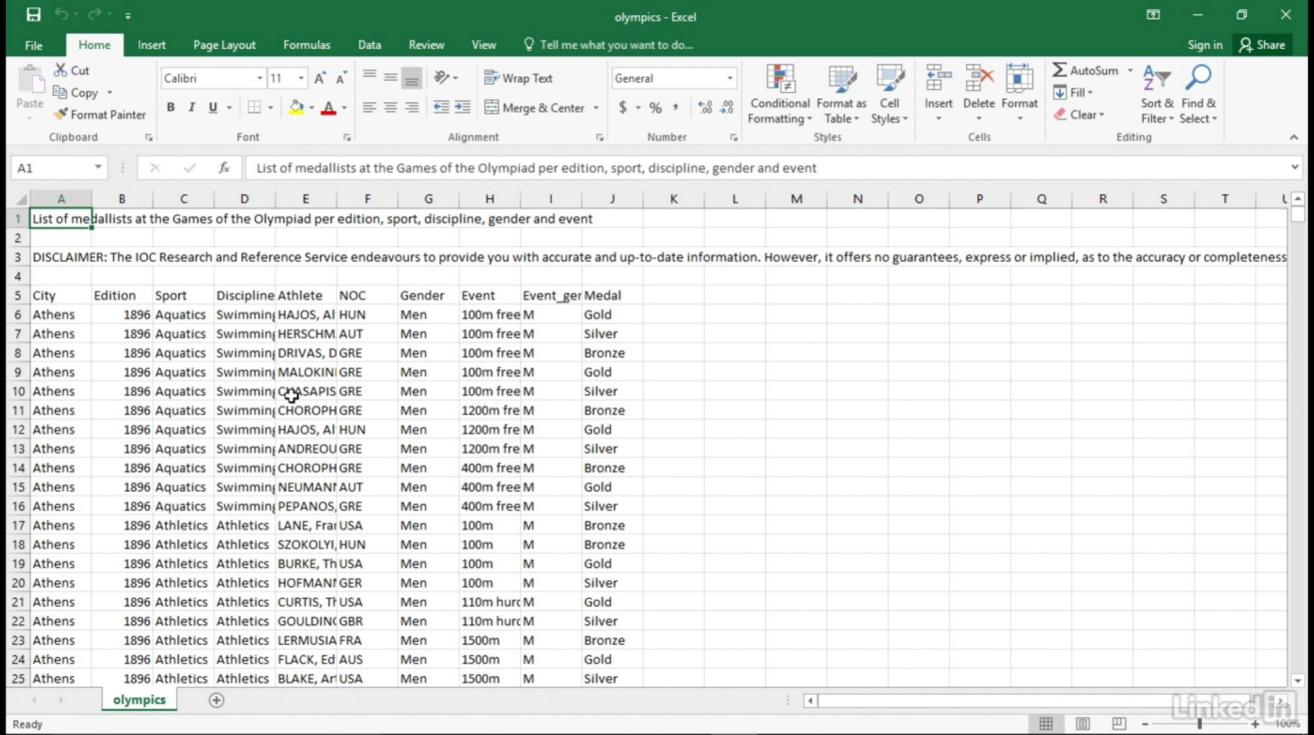
In [28]:
```
oo = pd.read_csv('../data/olympics.csv')
```

In [29]:
```
oo
```

Out[29]:

	List of medallists at the Games of the Olympiad per edition, sport, discipline, gender and event	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unn
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	DISCLAIMER: The IOC Research and Reference Ser...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	
4								100m		

In [ ]:



olympics - Excel

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Sign in Share

Cut Copy Format Painter Paste Clipboard

Calibri 11 Font

Wrap Text Merge & Center Alignment

General Number

Conditional Formatting Format as Table Cell Styles Styles

Insert Delete Format Cells

AutoSum Fill Clear Sort & Filter Find & Select Editing

A1 List of medallists at the Games of the Olympiad per edition, sport, discipline, gender and event

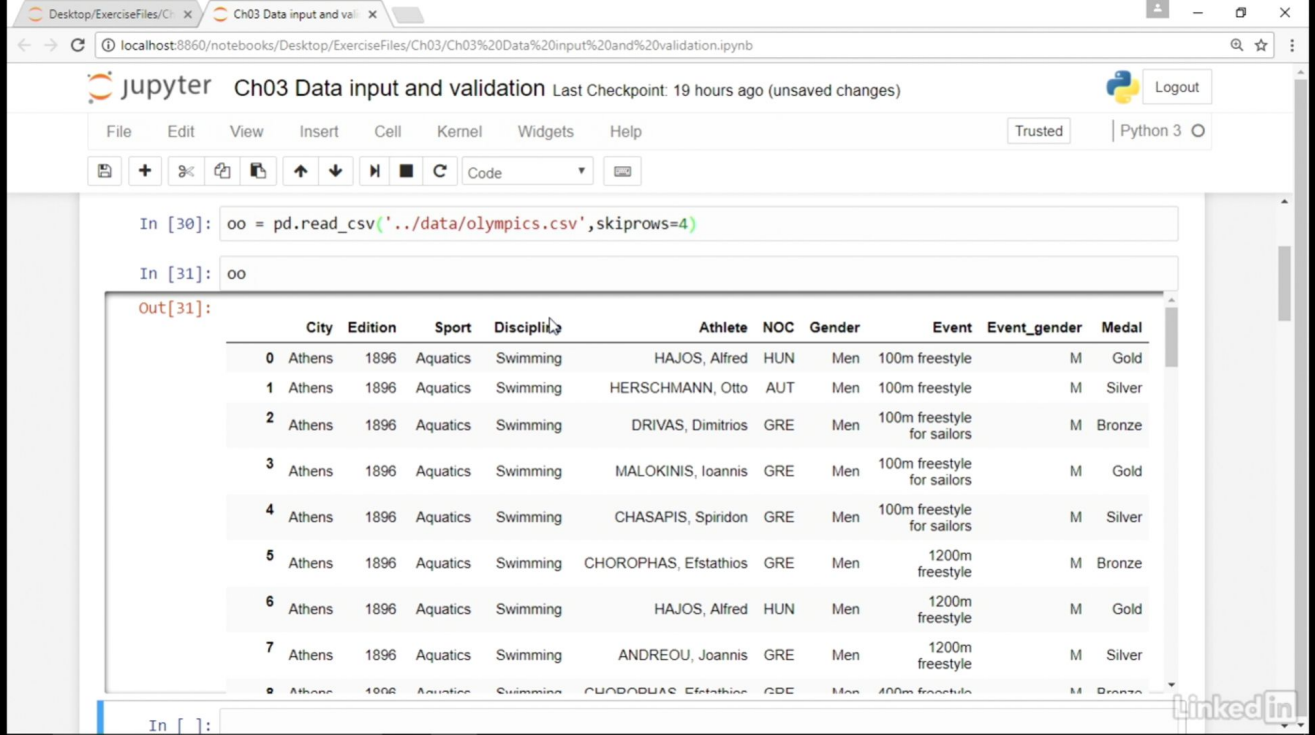
List of medallists at the Games of the Olympiad per edition, sport, discipline, gender and event

DISCLAIMER: The IOC Research and Reference Service endeavours to provide you with accurate and up-to-date information. However, it offers no guarantees, express or implied, as to the accuracy or completeness

City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_ger	Medal
Athens	1896	Aquatics	Swimming	HAJOS, Al	HUN	Men	100m free	M	Gold
Athens	1896	Aquatics	Swimming	HERSCHM	AUT	Men	100m free	M	Silver
Athens	1896	Aquatics	Swimming	DRIVAS, D	GRE	Men	100m free	M	Bronze
Athens	1896	Aquatics	Swimming	MALOKIN	GRE	Men	100m free	M	Gold
Athens	1896	Aquatics	Swimming	CHASAPIS	GRE	Men	100m free	M	Silver
Athens	1896	Aquatics	Swimming	CHOROPH	GRE	Men	1200m fre	M	Bronze
Athens	1896	Aquatics	Swimming	HAJOS, Al	HUN	Men	1200m fre	M	Gold
Athens	1896	Aquatics	Swimming	ANDREOU	GRE	Men	1200m fre	M	Silver
Athens	1896	Aquatics	Swimming	CHOROPH	GRE	Men	400m free	M	Bronze
Athens	1896	Aquatics	Swimming	NEUMANI	AUT	Men	400m free	M	Gold
Athens	1896	Aquatics	Swimming	PEPANOS,	GRE	Men	400m free	M	Silver
Athens	1896	Athletics	Athletics	LANE, Fra	USA	Men	100m	M	Bronze
Athens	1896	Athletics	Athletics	SZOKOLYI,	HUN	Men	100m	M	Bronze
Athens	1896	Athletics	Athletics	BURKE, Th	USA	Men	100m	M	Gold
Athens	1896	Athletics	Athletics	HOFMANI	GER	Men	100m	M	Silver
Athens	1896	Athletics	Athletics	CURTIS, Th	USA	Men	110m hurc	M	Gold
Athens	1896	Athletics	Athletics	GOULDIN	GBR	Men	110m hurc	M	Silver
Athens	1896	Athletics	Athletics	LERMUSIA	FRA	Men	1500m	M	Bronze
Athens	1896	Athletics	Athletics	FLACK, Ed	AUS	Men	1500m	M	Gold
Athens	1896	Athletics	Athletics	BLAKE, Ar	USA	Men	1500m	M	Silver

olympics

Ready



	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver
5	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	1200m freestyle	M	Bronze
6	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200m freestyle	M	Gold
7	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200m freestyle	M	Silver

Shape

- DataFrame.shape

jupyter Ch03 Data input and validation Last Checkpoint: 19 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

In [ ]:

Overview and validation of data

Shape

In [14]:
```
oo.shape
```
Out[14]: (29216, 10)

In [16]:
```
oo.shape[1]
```
Out[16]: 10

In [ ]:

In [ ]:

In [ ]:

In [ ]:

head( ) and tail( )

- DataFrame.head(n)
- DataFrame.tail(n)

head() and tail()

In [19]:
```
oo.head()
```

Out[19]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [20]:
```
oo.tail()
```

Out[20]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
29211	Beijing	2008	Wrestling	Wrestling Gre-R	ENGLICH, Mirko	GER	Men	84 - 96kg	M	Silver
29212	Beijing	2008	Wrestling	Wrestling Gre-R	MIZGAITIS, Mindaugas	LTU	Men	96 - 120kg	M	Bronze
29213	Beijing	2008	Wrestling	Wrestling Gre-R	PATRIKEEV, Yuri	ARM	Men	96 - 120kg	M	Bronze
29214	Beijing	2008	Wrestling	Wrestling Gre-R	LOPEZ, Mijain	CUB	Men	96 - 120kg	M	Gold

jupyter Ch03 Data input and validation Last Checkpoint: 19 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

In [22]:
```
oo.tail()
```

Out[22]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
29211	Beijing	2008	Wrestling	Wrestling Gre-R	ENGLICH, Mirko	GER	Men	84 - 96kg	M	Silver
29212	Beijing	2008	Wrestling	Wrestling Gre-R	MIZGAITIS, Mindaugas	LTU	Men	96 - 120kg	M	Bronze
29213	Beijing	2008	Wrestling	Wrestling Gre-R	PATRIKEEV, Yuri	ARM	Men	96 - 120kg	M	Bronze
29214	Beijing	2008	Wrestling	Wrestling Gre-R	LOPEZ, Mijain	CUB	Men	96 - 120kg	M	Gold
29215	Beijing	2008	Wrestling	Wrestling Gre-R	BAROEV, Khasan	RUS	Men	96 - 120kg	M	Silver

In [25]:
```
oo = pd.read_csv('../data/olympics.csv',skiprows=4)
oo.head()
```

Out[25]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [ ]:

info( )

- DataFrame.info( )

jupyter Ch03 Data input and validation Last Checkpoint: 19 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

info()

In [26]:
```
oo.info()
```

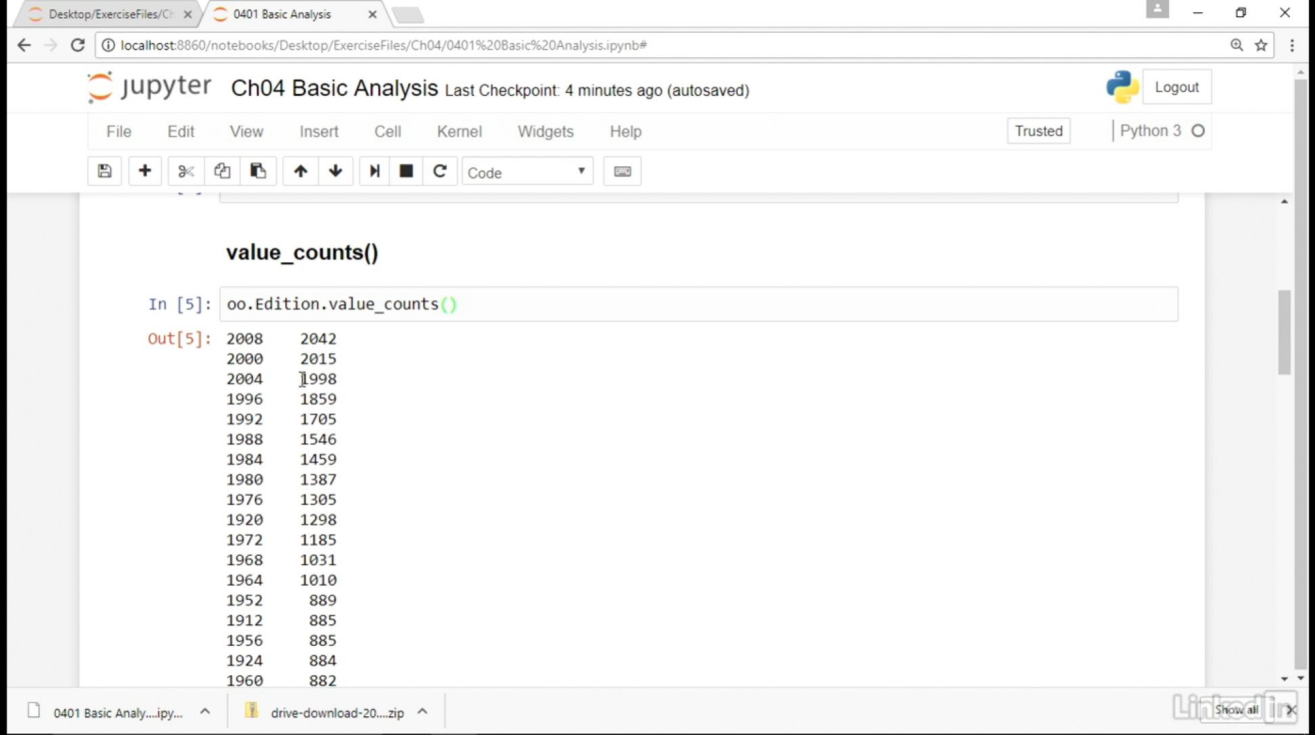
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29216 entries, 0 to 29215
Data columns (total 10 columns):
City            29216 non-null object
Edition         29216 non-null int64
Sport           29216 non-null object
Discipline      29216 non-null object
Athlete         29216 non-null object
NOC             29216 non-null object
Gender          29216 non-null object
Event           29216 non-null object
Event_gender    29216 non-null object
Medal           29216 non-null object
dtypes: int64(1), object(9)
memory usage: 2.2+ MB
```

In [ ]:

In [ ]:

value_counts( )

- Series.**value_counts**(normalize=False, sort=True, ascending=False, bins=None, dropna=True)
- Returns a object containing counts of unique values
- By default, results are in descending order so first element is most frequently occurring element


Desktop/ExerciseFiles/Ch
0401 Basic Analysis
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch04/0401%20Basic%20Analysis.ipynb#
jupyter
Ch04 Basic Analysis
Last Checkpoint: 4 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
value_counts()
In [5]:
oo.Edition.value_counts()
Out[5]:
2008 2042
2000 2015
2004 1998
1996 1859
1992 1705
1988 1546
1984 1459
1980 1387
1976 1305
1920 1298
1972 1185
1968 1031
1964 1010
1952 889
1912 885
1956 885
1924 884
1960 882
0401 Basic Analy....ipy...
drive-download-20....zip
Show all

jupyter Ch04 Basic Analysis Last Checkpoint: 5 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

```
1932     615
1900     512
1904     470
1896     151
Name: Edition, dtype: int64
```

In [6]:
```
oo.Gender.value_counts()
```

Out[6]:
```
Men      21721
Women     7495
Name: Gender, dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

jupyter Ch04 Basic Analysis Last Checkpoint: 6 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

```
1932     615
1900     512
1904     470
1896     151
Name: Edition, dtype: int64
```

In [7]:
```
oo.Gender.value_counts(ascending=True)
```

Out[7]:
```
Women     7495
Men      21721
Name: Gender, dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

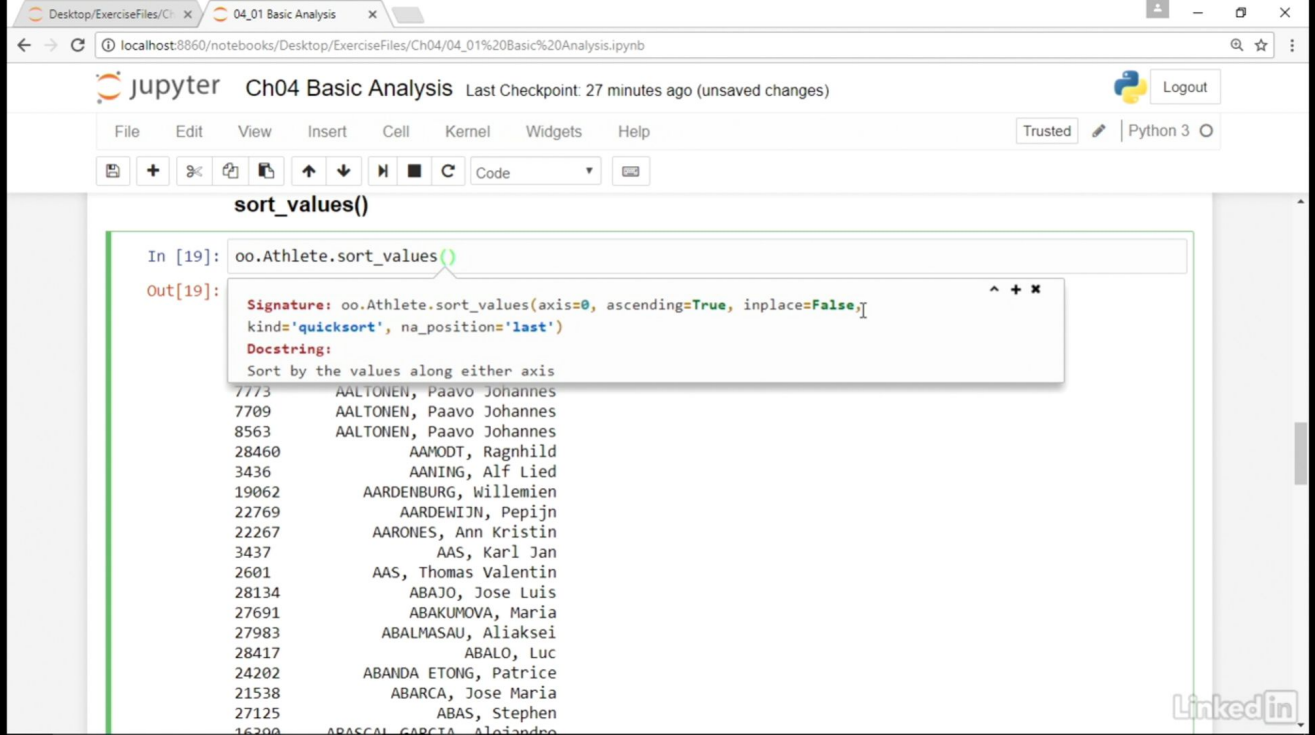
In [ ]:

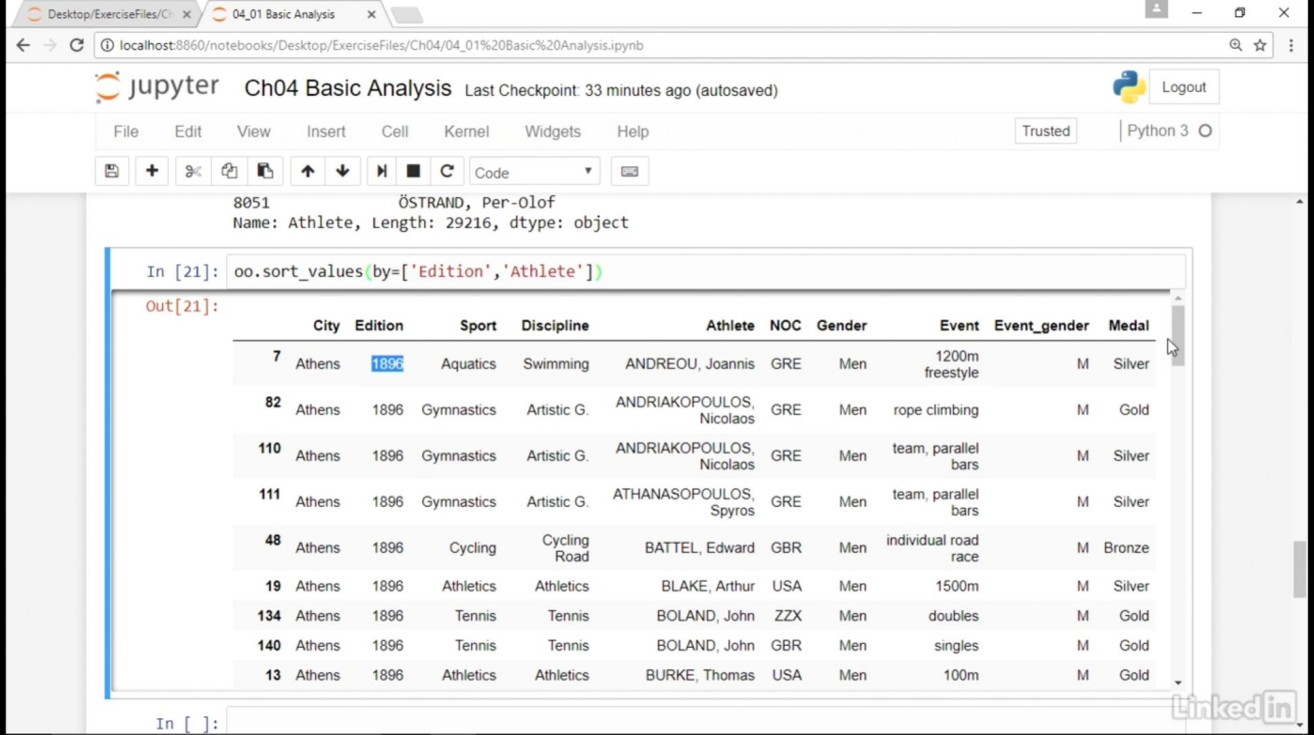
In [ ]:

sort_values( )

- Series.**sort_values**(*axis=0, ascending=True, inplace=False, kind='quicksort', na_position='last'*)
- Sort values along either axis
- DataFrame.sort_values(*by, axis=0, ascending=True, inplace=False, kind='quicksort', na_position='last'*)

```
DataFrame.sort_values(['Series1','Series2'])
```


Desktop/ExerciseFiles/Ch
04_01 Basic Analysis
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch04/04_01%20Basic%20Analysis.ipynb
jupyter
Ch04 Basic Analysis
Last Checkpoint: 27 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
sort_values()
In [19]:
oo.Athlete.sort_values()
Out[19]:
Signature: oo.Athlete.sort_values(axis=0, ascending=True, inplace=False,
kind='quicksort', na_position='last')
Docstring:
Sort by the values along either axis
7773 AALTONEN, Paavo Johannes
7709 AALTONEN, Paavo Johannes
8563 AALTONEN, Paavo Johannes
28460 AAMODT, Ragnhild
3436 AANING, Alf Lied
19062 AARDENBURG, Willemien
22769 AARDEWIJN, Pepijn
22267 AARONES, Ann Kristin
3437 AAS, Karl Jan
2601 AAS, Thomas Valentin
28134 ABAJO, Jose Luis
27691 ABAKUMOVA, Maria
27983 ABALMASAU, Aliaksei
28417 ABALO, Luc
24202 ABANDA ETONG, Patrice
21538 ABARCA, Jose Maria
27125 ABAS, Stephen



	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
7	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200m freestyle	M	Silver
82	Athens	1896	Gymnastics	Artistic G.	ANDRIAKOPOULOS, Nicolaos	GRE	Men	rope climbing	M	Gold
110	Athens	1896	Gymnastics	Artistic G.	ANDRIAKOPOULOS, Nicolaos	GRE	Men	team, parallel bars	M	Silver
111	Athens	1896	Gymnastics	Artistic G.	ATHANASOPOULOS, Spyros	GRE	Men	team, parallel bars	M	Silver
48	Athens	1896	Cycling	Cycling Road	BATTEL, Edward	GBR	Men	individual road race	M	Bronze
19	Athens	1896	Athletics	Athletics	BLAKE, Arthur	USA	Men	1500m	M	Silver
134	Athens	1896	Tennis	Tennis	BOLAND, John	ZZX	Men	doubles	M	Gold
140	Athens	1896	Tennis	Tennis	BOLAND, John	GBR	Men	singles	M	Gold
13	Athens	1896	Athletics	Athletics	BURKE, Thomas	USA	Men	100m	M	Gold

Boolean Indexing

- Boolean vectors can be used to filter data

Operator	Symbol
AND	&
OR	|
NOT	~

- Multiple conditions must be grouped using brackets

in brackets or parentheses.

Boolean indexing

In [22]:
```
oo.head()
```

Out[22]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [23]:
```
oo.Medal == 'Gold'
```

Out[23]:
```
0         True
1        False
2        False
3         True
4        False
5        False
6         True
7        False
8        False
```

1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [24]:
```
oo[oo.Medal == 'Gold']
```

Out[24]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
6	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200m freestyle	M	Gold
9	Athens	1896	Aquatics	Swimming	NEUMANN, Paul	AUT	Men	400m freestyle	M	Gold
13	Athens	1896	Athletics	Athletics	BURKE, Thomas	USA	Men	100m	M	Gold
15	Athens	1896	Athletics	Athletics	CURTIS, Thomas	USA	Men	110m hurdles	M	Gold
18	Athens	1896	Athletics	Athletics	FLACK, Edwin	AUS	Men	1500m	M	Gold
21	Athens	1896	Athletics	Athletics	BURKE, Thomas	USA	Men	400m	M	Gold

Desktop/ExerciseFiles/Ch | 04_01 Basic Analysis

localhost:8860/notebooks/Desktop/ExerciseFiles/Ch04/04_01%20Basic%20Analysis.ipynb

jupyter Ch04 Basic Analysis Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [25]:
```
oo[(oo.Medal == 'Gold') & (oo.Gender == 'Women')]
```

Out[25]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	M
417	Paris	1900	Golf	Golf	ABBOTT, Margaret Ives	USA	Women	individual	W	
641	Paris	1900	Tennis	Tennis	COOPER, Charlotte	GBR	Women	mixed doubles	X	
649	Paris	1900	Tennis	Tennis	COOPER, Charlotte	GBR	Women	singles	W	
710	St Louis	1904	Archery	Archery	HOWELL, Matilda Scott	USA	Women	double columbia round (50y - 40y - 30y)	W	
713	St Louis	1904	Archery	Archery	HOWELL, Matilda Scott	USA	Women	double national round (60y - 50y)	W	
730	St Louis	1904	Archery	Archery	HOWELL, Matilda Scott	USA	Women	teams FITA round	W	

String Handling

- Available to every Series using the str attribute
- Series.str – access values of series as strings and apply several methods to it

  Series.str.contains()

  Series.str.startswith()

  Series.str.isnumeric()

String handling

In [29]:
```
oo[oo.Athlete.str.contains('Florence')]
```

Out[29]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
1843	London	1908	Skating	Figure skating	SYERS, Florence	GBR	Women	individual	W	Gold
1848	London	1908	Skating	Figure skating	SYERS, Florence	GBR	Women	pairs	X	Bronze
4173	Paris	1924	Aquatics	Swimming	BARKER, Florence	GBR	Women	4x100m freestyle relay	W	Silver
8162	Helsinki	1952	Athletics	Athletics	FOULDS-PAUL, June Florence	GBR	Women	4x100m relay	W	Bronze
9060	Melbourne / Stockholm	1956	Athletics	Athletics	FOULDS-PAUL, June Florence	GBR	Women	4x100m relay	W	Silver
10849	Tokyo	1964	Athletics	Athletics	AMOORE-POLLOCK, Judith Florence	AUS	Women	400m	W	Bronze
16817	Los Angeles	1984	Athletics	Athletics	GRIFFITH-JOYNER, Florence	USA	Women	200m	W	Silver
18287	Seoul	1988	Athletics	Athletics	GRIFFITH-JOYNER, Florence	USA	Women	100m	W	Gold

Challenge

- In which events did Jesse Owens win a medal?
- Which country has won the most men's gold medals in singles badminton over the years?

  Sort the results alphabetically by the player's names.

Challenge

- Which three countries have won the most medals in recent years (from 1984 to 2008)?

- Display the male gold medal winners for the 100m track and field sprint event over the years.

  List the results starting with the most recent.

  Show the Olympic city, edition, athlete, and the country they represent.

In which events did Jesse Owens win a medal?

In [46]:
```
jo = oo[oo.Athlete == 'OWENS, Jesse']
jo
```

Out[46]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
6427	Berlin	1936	Athletics	Athletics	OWENS, Jesse	USA	Men	100m	M	Gold
6439	Berlin	1936	Athletics	Athletics	OWENS, Jesse	USA	Men	200m	M	Gold
6456	Berlin	1936	Athletics	Athletics	OWENS, Jesse	USA	Men	4x100m relay	M	Gold
6523	Berlin	1936	Athletics	Athletics	OWENS, Jesse	USA	Men	long jump	M	Gold

In [47]:
```
jo.Event.value_counts()
```

Out[47]:
```
4x100m relay    1
200m            1
long jump       1
100m            1
Name: Event, dtype: int64
```

In [ ]:

Which country has won the most men's gold medals in singles badminton over the years? Sort the results alphabetically by the player's names.

In [49]:
```
gbm = oo[(oo.Medal == 'Gold') & (oo.Gender == 'Men') & (oo.Sport == 'Badminton')]
gbm
```

Out[49]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
20031	Barcelona	1992	Badminton	Badminton	KIM, Moon-Soo	KOR	Men	doubles	M	Gold
20032	Barcelona	1992	Badminton	Badminton	PARK, Joo-Bong	KOR	Men	doubles	M	Gold
20045	Barcelona	1992	Badminton	Badminton	BUDI KUSUMA, Alan	INA	Men	singles	M	Gold
21771	Atlanta	1996	Badminton	Badminton	MAINAKY, Rexy Ronald	INA	Men	doubles	M	Gold
21772	Atlanta	1996	Badminton	Badminton	SUBAGJA, Ricky Achmad	INA	Men	doubles	M	Gold
21773	Atlanta	1996	Badminton	Badminton	KIM, Dong Moon	KOR	Men	doubles	X	Gold
21787	Atlanta	1996	Badminton	Badminton	HOYER-LARSEN, Poul Erik	DEN	Men	singles	M	Gold
23701	Sydney	2000	Badminton	Badminton	GUNAWAN, Tony	INA	Men	doubles	M	Gold
23702	Sydney	2000	Badminton	Badminton	WIJAYA, Candra	INA	Men	doubles	M	Gold
23703	Sydney	2000	Badminton	Badminton	ZHANG, Jun	CHN	Men	doubles	X	Gold
23717	Sydney	2000	Badminton	Badminton	JI, Xinpeng	CHN	Men	singles	M	Gold
25718	Athens	2004	Badminton	Badminton	HA, Tae-Kwon	KOR	Men	doubles	M	Gold
25719	Athens	2004	Badminton	Badminton	KIM, Dong Moon	KOR	Men	doubles	M	Gold

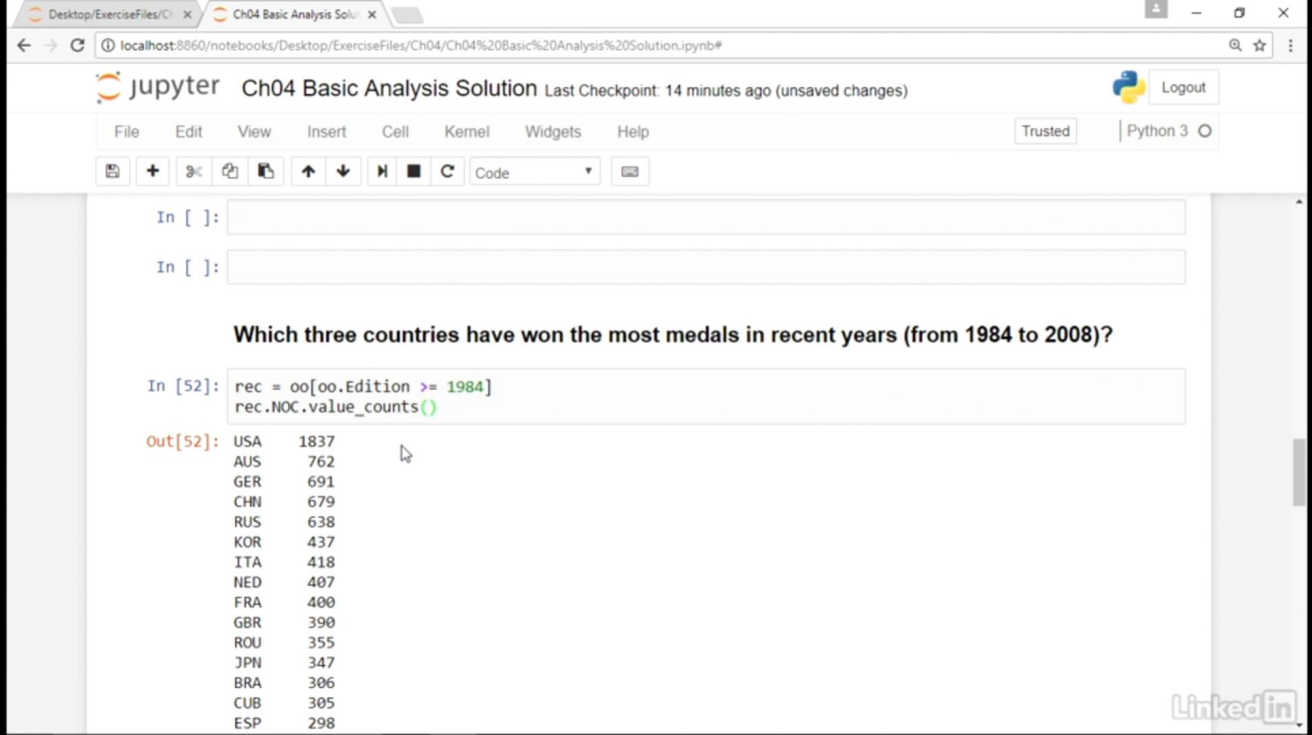
In [ ]:

Which country has won the most men's gold medals in singles badminton over the years? Sort the results alphabetically by the player's names.

In [50]:
```
gbm = oo[(oo.Medal == 'Gold') & (oo.Gender == 'Men') & (oo.Sport == 'Badminton')]
gbm.sort_values(by='Athlete')
```

Out[50]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
20045	Barcelona	1992	Badminton	Badminton	BUDI KUSUMA, Alan	INA	Men	singles	M	Gold
23701	Sydney	2000	Badminton	Badminton	GUNAWAN, Tony	INA	Men	doubles	M	Gold
25718	Athens	2004	Badminton	Badminton	HA, Tae-Kwon	KOR	Men	doubles	M	Gold
25734	Athens	2004	Badminton	Badminton	HIDAYAT, Taufik	INA	Men	singles	M	Gold
21787	Atlanta	1996	Badminton	Badminton	HOYER-LARSEN, Poul Erik	DEN	Men	singles	M	Gold
23717	Sydney	2000	Badminton	Badminton	JI, Xinpeng	CHN	Men	singles	M	Gold
27725	Beijing	2008	Badminton	Badminton	KIDO, Markis	INA	Men	doubles	M	Gold
25719	Athens	2004	Badminton	Badminton	KIM, Dong Moon	KOR	Men	doubles	M	Gold
21773	Atlanta	1996	Badminton	Badminton	KIM, Dong Moon	KOR	Men	doubles	X	Gold


Desktop/ExerciseFiles/Ch
Ch04 Basic Analysis Solu
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch04/Ch04%20Basic%20Analysis%20Solution.ipynb#
jupyter
Ch04 Basic Analysis Solution
Last Checkpoint: 14 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
In [ ]:
In [ ]:
Which three countries have won the most medals in recent years (from 1984 to 2008)?
In [52]:
rec = oo[oo.Edition >= 1984]
rec.NOC.value_counts()
Out[52]:
USA 1837
AUS 762
GER 691
CHN 679
RUS 638
KOR 437
ITA 418
NED 407
FRA 400
GBR 390
ROU 355
JPN 347
BRA 306
CUB 305
ESP 298

In [ ]:

In [ ]:

Which three countries have won the most medals in recent years (from 1984 to 2008)?

In [53]:
```
rec = oo[oo.Edition >= 1984]
rec.NOC.value_counts().head(3)
```

Out[53]:
```
USA    1837
AUS     762
GER     691
Name: NOC, dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

Display the male gold medal winners for the 100m Track & Field sprint event over the years.

Display the male gold medal winners for the 100m Track & Field sprint event over the years. List the results starting with the most recent. Show the Olympic City, Edition, Athlete and the country they represent.

In [57]:
```
gmh = oo[(oo.Gender == 'Men') & (oo.Medal == 'Gold') & (oo.Event == '100m')]
gmh.sort_values('Edition',ascending=False)[['City','Edition','Athlete','NOC']]
```

Out[57]:

	City	Edition	Athlete	NOC
27552	Beijing	2008	BOLT, Usain	JAM
25539	Athens	2004	GATLIN, Justin	USA
23521	Sydney	2000	GREENE, Maurice	USA
21598	Atlanta	1996	BAILEY, Donovan	CAN
19859	Barcelona	1992	CHRISTIE, Linford	GBR
18284	Seoul	1988	LEWIS, Carl	USA
16794	Los Angeles	1984	LEWIS, Carl	USA
15374	Moscow	1980	WELLS, Allan	GBR
14069	Montreal	1976	CRAWFORD, Hasely	TRI
12902	Munich	1972	BORZOV, Valery	URS
11865	Mexico	1968	HINES, James Ray	USA

Matplotlib

In [1]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

Matplotlib

- In [1]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

- Object-oriented method vs. pyplot

Plot Types

- plot()

  plot(kind='line')

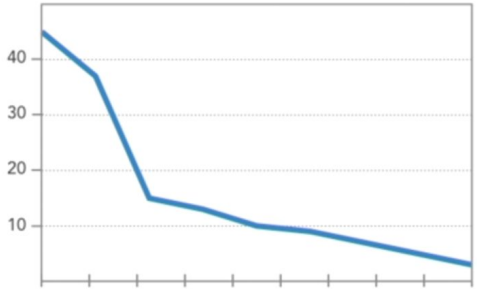
  plot(kind='bar')

  plot(kind='barh')

  plot(kind='pie')

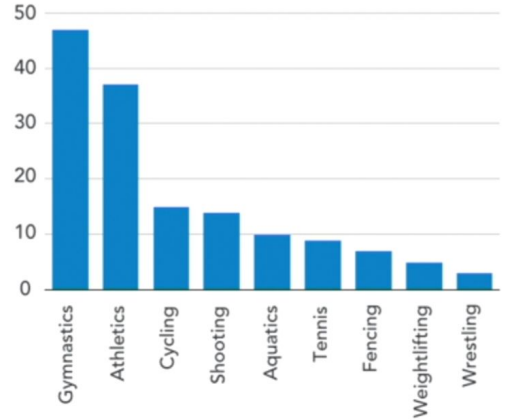
plot(kind='line')

- Best for tracking changes over a period of time



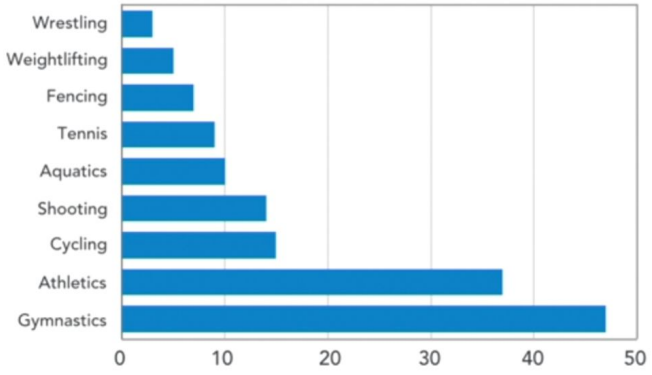
plot(kind='bar')

- Best for comparing between different group



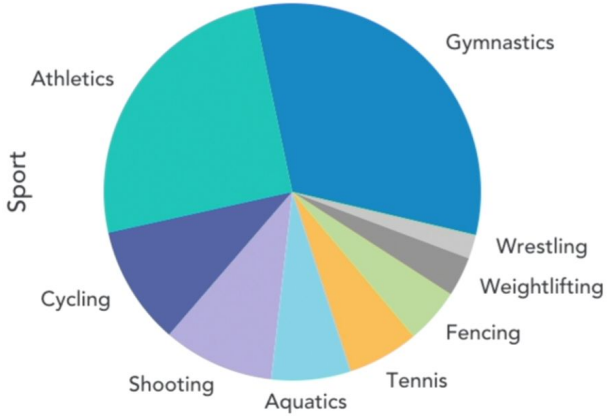
plot(kind='barh')

- Best for comparing between different groups



plot(kind='pie')

- Best for comparing parts of whole



Plot types

What were the different sports in the first olympics? Plot them using different graphs.

In [5]:
```
fo = oo[oo.Edition == 1896]
fo.head()
```

Out[5]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [ ]:
```

```

Line plot

In [ ]:

2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [ ]:

Line plot

In [6]:
```
fo.Sport.value_counts()
```

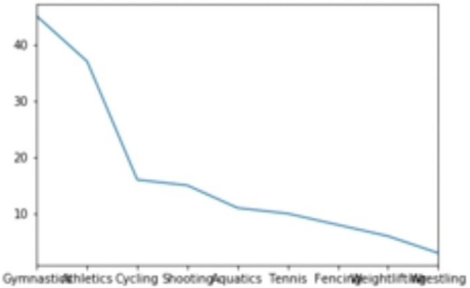
Out[6]:
```
Gymnastics      45
Athletics       37
Cycling         16
Shooting        15
Aquatics        11
Tennis          10
Fencing          8
Weightlifting    6
Wrestling        3
Name: Sport, dtype: int64
```

In [ ]:

Line plot

In [7]:
```
fo.Sport.value_counts().plot()
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21bae2604a8>
```



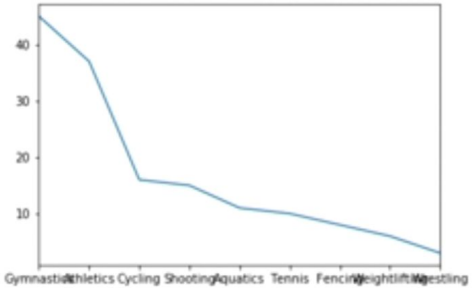
In [ ]:

In [ ]:

Line plot

In [10]:
```
fo.Sport.value_counts().plot(kind='line')
```

Out[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21baeaa9588>
```



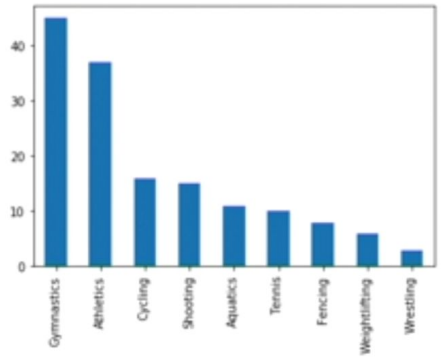
In [ ]:

In [ ]:

Bar plot

```
In [12]: fo.Sport.value_counts().plot(kind='bar');
```

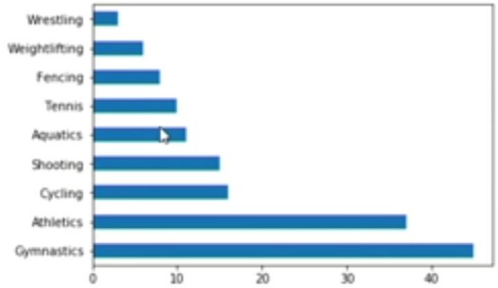
```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x21baea05a20>
```



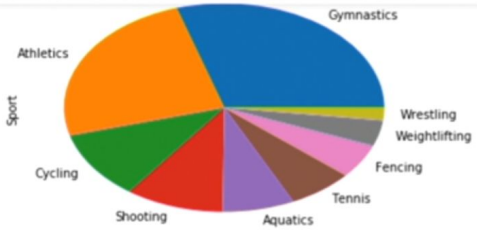
In [ ]:

Horizontal bar plot

In [14]:
```
fo.Sport.value_counts().plot(kind='barh');
```



In [ ]:



In [ ]:

In [ ]:

Plot colors

In [ ]:

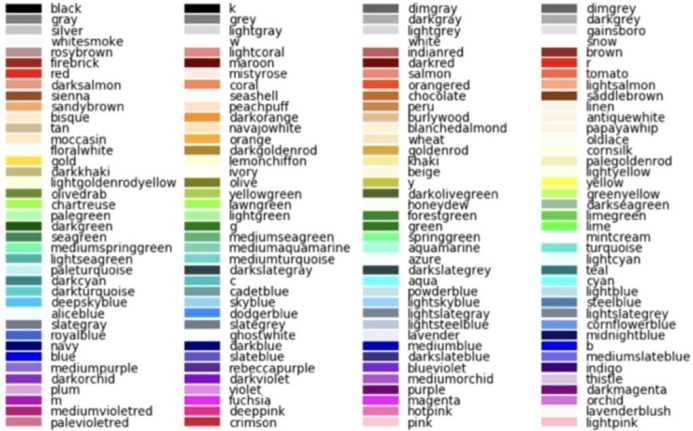
In [ ]:

Plot Colors

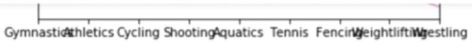
- plot()

```
plot(color='red')

plot(kind='bar', color='blue')
```

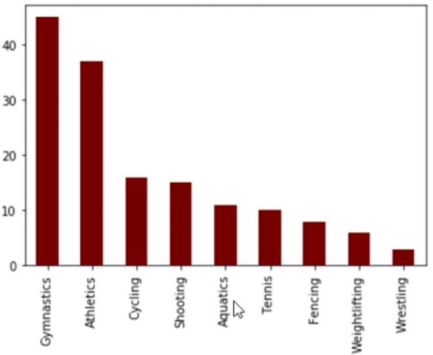
Matplotlib



Source: http://matplotlib.org/examples/color/named_colors.html



In [19]:
```
fo.Sport.value_counts().plot(kind='bar',color='maroon');
```



In [ ]:

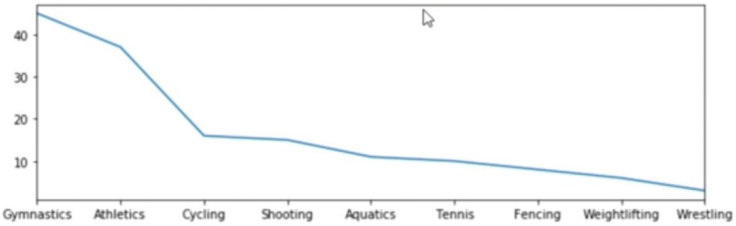
Figure Size

- plot(figsize=(width, height))

```
plot(kind='bar',color='yellow',figsize(5,5))
```

figsize()

In [21]:
```
fo.Sport.value_counts().plot(figsize=(10,3));
```



In [ ]:

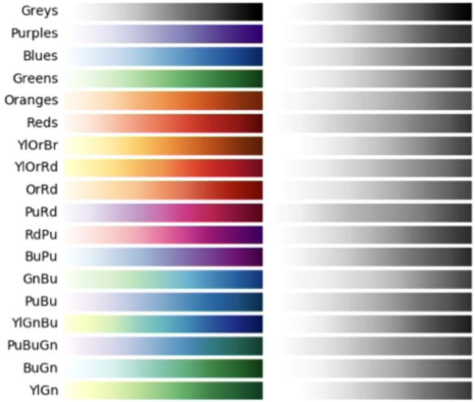
In [ ]:

Colormaps

Colormaps

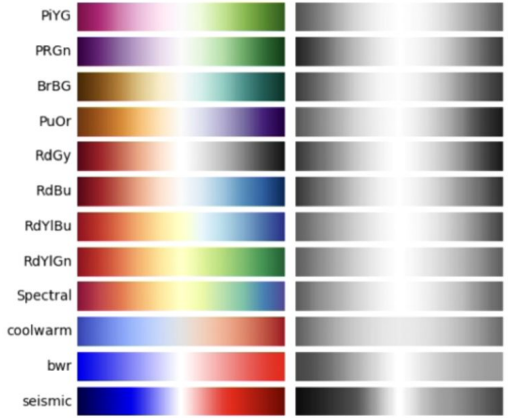
- Classes of colormaps
  - Sequential
  - Diverging
  - Qualitative

Colormaps: Sequential



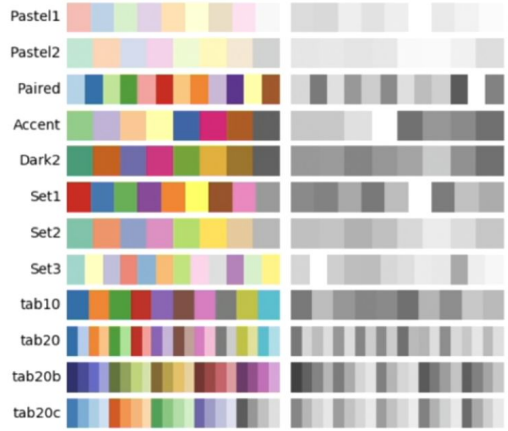
Source: Matplotlib documentation

Colormaps: Diverging



Source: Matplotlib documentation

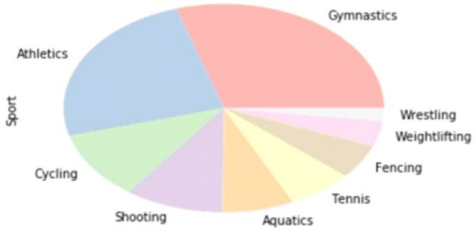
Colormaps: Qualitative



Source: Matplotlib documentation

Colormaps

In [22]:
```
fo.Sport.value_counts().plot(kind='pie',colormap='Pastel1');
```



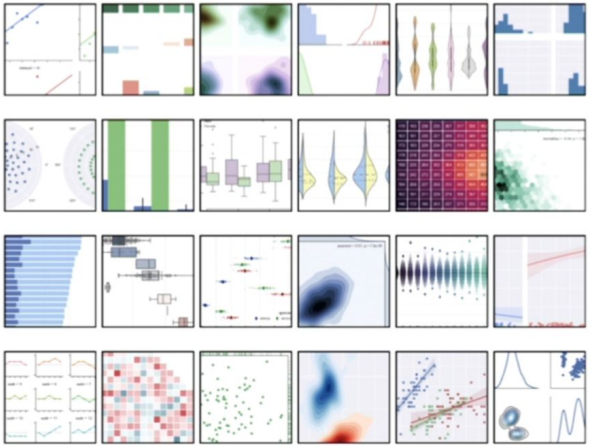
In [ ]:
```
fo.Sport.value_counts().plot(kind='pie',color);
```

In [ ]:

Why Seaborn

- Attractive statistical plots
- A complement and not a substitute to Matplotlib
- Integrates well with pandas

Example Gallery



Source: http://seaborn.pydata.org/example/

countplot()

- seaboard.**countplot**(*x=None, y=None, hue=None, data=None, order=None, hue_order=None, orient=None, color=None, palette=None, …*)

  data: the DataFrame or source of data

  hue: categorical variables

  order: the sequence when using categorical levels

  palette: colors to use for the different levels of the hue variable

Seaborn vs Matplotlib

- Matplotlib

  Short scripts and working in conjunction with pyplot

  Simple plot types - line, bar, plot

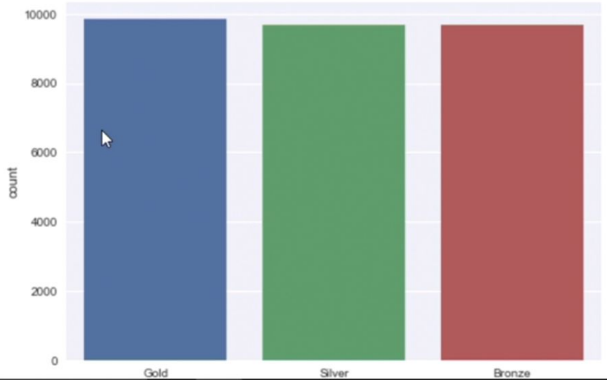
- Seaborn

  Dealing with statistical data or categorical data, or requiring more advanced plots

How many medals have been won by men and women in the history of the Olympics. How many gold, silver and bronze medals were won for each gender?

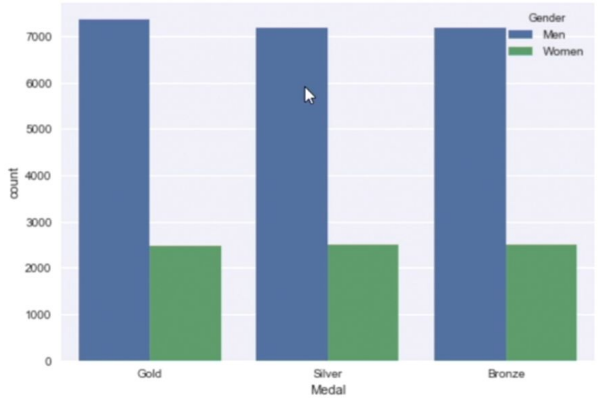
In [26]:
```
sns.countplot(x='Medal',data=oo)
```

Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21bb109bbe0>
```



How many medals have been won by men and women in the history of the Olympics. How many gold, silver and bronze medals were won for each gender?

In [28]:
```
sns.countplot(x='Medal',data=oo, hue='Gender');
```



Challenge

- Plot the number of medals achieved by the Chinese team (men and women) in Beijing 2008 using

  Matplotlib

  Seaborn

- How can you use colormaps to give the data more meaning?

Challenge

- Plot the number of gold, silver, and bronze medals for each gender.

- How can you give the data more meaning? Is there anything else you can change to make it more intuitive?

jupyter Basic plotting Solution (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Code

Basic plotting challenge

In [3]:
```
import pandas as pd
```

In [4]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [5]:
```
import seaborn as sns
```

In [6]:
```
oo = pd.read_csv('../data/olympics.csv',skiprows=4)
oo.head()
```

Out[6]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

run cell | shift + enter

jupyter Basic plotting Solution (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Code

In [ ]:

Plot the number of medals achieved by the Chinese team (men and women) in Beijing 2008 using:

- matplotlib
- Seaborn

In [ ]:
```
oo[(oo.Edition == 2008) & (oo.NOC == 'CHN')]
```

In [ ]:

In [ ]:

How can you use colormaps to give the data more meaning?

https://matplotlib.org/mpl_examples/color/colormaps_reference_02.png

In [ ]:

jupyter Basic plotting Solution (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Code

In [ ]:

Plot the number of medals achieved by the Chinese team (men and women) in Beijing 2008 using:

- matplotlib
- Seaborn

In [ ]:
```
mw = oo[(oo.Edition == 2008) & (oo.NOC == 'CHN')]
mw.head()
```

In [ ]:

In [ ]:

How can you use colormaps to give the data more meaning?

https://matplotlib.org/mpl_examples/color/colormaps_reference_02.png

In [ ]:

using:

- matplotlib
- Seaborn

In [9]:
```
mw = oo[(oo.Edition == 2008) & (oo.NOC == 'CHN')]
mw.head()
```

Out[9]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
27176	Beijing	2008	Aquatics	Diving	ZHOU, Luxin	CHN	Men	10m platform	M	Silver
27177	Beijing	2008	Aquatics	Diving	WANG, Xin	CHN	Women	10m platform	W	Bronze
27178	Beijing	2008	Aquatics	Diving	CHEN, Ruolin	CHN	Women	10m platform	W	Gold
27180	Beijing	2008	Aquatics	Diving	QIN, Kai	CHN	Men	3m springboard	M	Bronze
27181	Beijing	2008	Aquatics	Diving	HE, Chong	CHN	Men	3m springboard	M	Gold

In [10]:
```
mw.Gender.value_counts().plot(kind='bar')
```

Out[10]:
```
Women    128
Men       56
Name: Gender, dtype: int64
```

In [ ]:

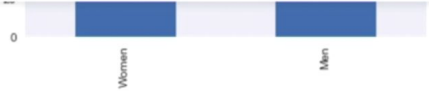
jupyter Basic plotting Solution (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

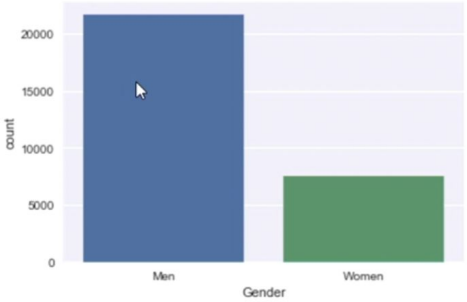
Not Trusted | Python 3

Markdown



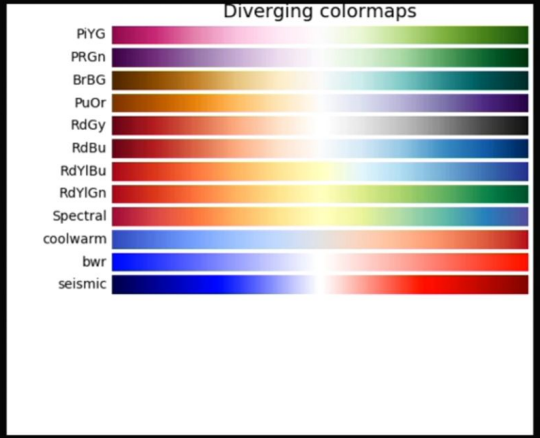
In [12]:
```
sns.countplot(data=oo, x='Gender')
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x1a2a2709cf8>



Desktop/ExerciseFiles/Ch | Basic plotting challenge | colormaps_reference_03

Secure | https://matplotlib.org/_images/colormaps_reference_03.png



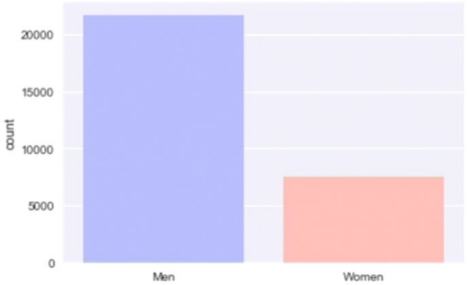
Gender

How can you use colormaps to give the data more meaning?

https://matplotlib.org/_images/colormaps_reference_03.png

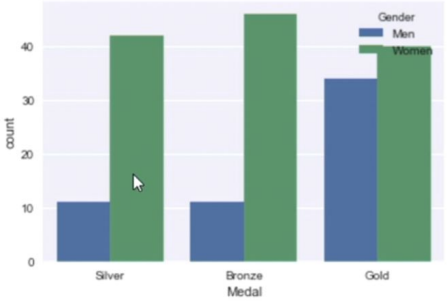
In [13]:
```
sns.countplot(data=oo, x='Gender',palette='bwr')
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1a2a2b03ba8>



Plot the number of Gold, Silver and Bronze medals for each gender.

In [17]:
```
sns.countplot(x='Medal',data=mw, hue='Gender');
```

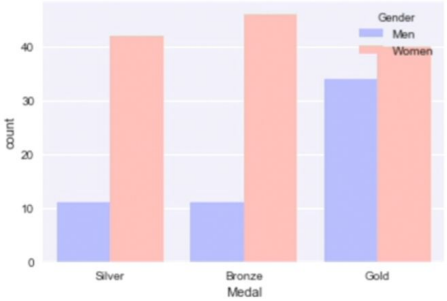


In [ ]:

How can you give the data more meaning? Is there anything else you can change to make it

How can you give the data more meaning? Is there anything else you can change to make it more intuitive?

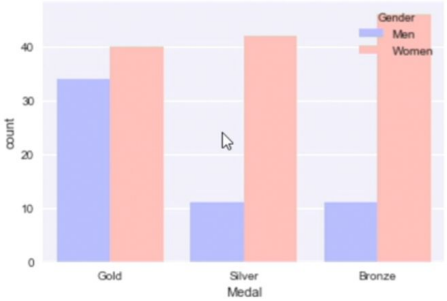
In [19]:
```
sns.countplot(x='Medal',data=mw, hue='Gender',palette='bwr');
```



In [ ]:
```

```

How can you give the data more meaning? Is there anything else you can change to make it more intuitive?

In [20]:
```
sns.countplot(x='Medal',data=mw, hue='Gender',palette='bwr',order=['Gold','Silver','Bronze']);
```



In [ ]:
```

```

Index

- The index object is an immutable array
- Indexing allows you to access a row or column using a label

3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

In [ ]:

Index

In [3]:
```
type(oo.index)
```

Out[3]: pandas.core.indexes.range.RangeIndex

In [4]:
```
oo.index[100]
```

Out[4]: 100

In [5]:
```
oo.index[100] = 5
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-d016cfc6c475> in <module>()
----> 1 oo.index[100] = 5

C:\Users\Jonathan\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in __setitem__(self, key, va
lue)
   1618
```

set_index( )

- DataFrame.**set_index**(keys, drop=True, append=False, inplace=False, verify_integrity=False)
- Set the DataFrame using one or more columns
- set_index(keys, inplace=True)

jupyter Ch06 Indexing Last Checkpoint: 22 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

In [ ]:

set_index()

In [23]:
```
oo.head()
```

Out[23]:

	Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
0	HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	100m freestyle	M	Gold
1	HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
2	DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
4	CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [ ]:
```
oo.set_index('Athlete')
```

In [ ]:

In [ ]:

3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
4	CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [24]:
```
oo.set_index('Athlete')
```

Out[24]:

Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	100m freestyle	M	Gold
HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver
CHOROPHAS, Efstathios	Athens	1896	Aquatics	Swimming	GRE	Men	1200m freestyle	M	Bronze
HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	1200m freestyle	M	Gold
ANDREOU, Joannis	Athens	1896	Aquatics	Swimming	GRE	Men	1200m freestyle	M	Silver
CHOROPHAS, Efstathios	Athens	1896	Aquatics	Swimming	GRE	Men	400m freestyle	M	Bronze

CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver
CHOROPHAS, Efstathios	Athens	1896	Aquatics	Swimming	GRE	Men	1200m freestyle	M	Bronze
HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	1200m freestyle	M	Gold
ANDREOU, Joannis	Athens	1896	Aquatics	Swimming	GRE	Men	1200m freestyle	M	Silver
CHOROPHAS, Efstathios	Athens	1896	Aquatics	Swimming	GRE	Men	400m freestyle	M	Bronze

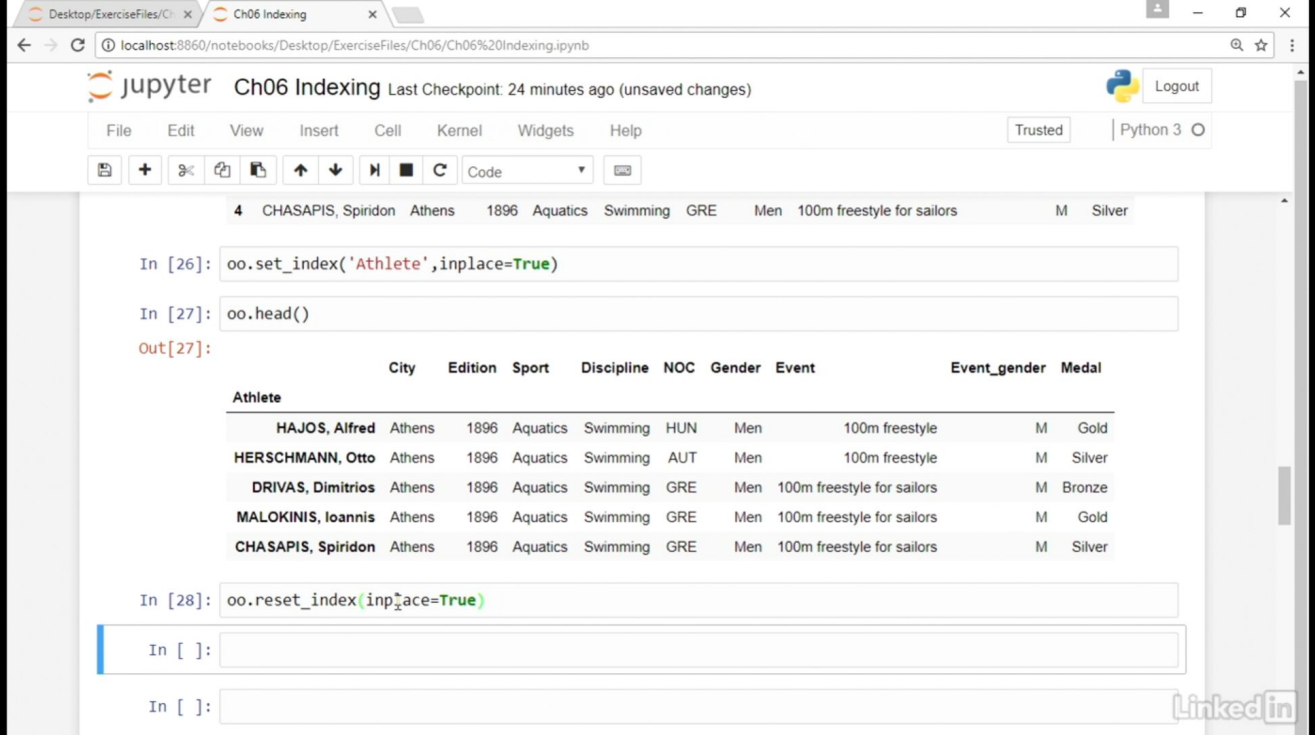
In [25]:
```
oo.head()
```

Out[25]:

	Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
0	HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	100m freestyle	M	Gold
1	HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
2	DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
4	CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [ ]:

In [ ]:


Desktop/ExerciseFiles/Ch
Ch06 Indexing
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch06/Ch06%20Indexing.ipynb
jupyter
Ch06 Indexing Last Checkpoint: 24 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Code
4 CHASAPIS, Spiridon Athens 1896 Aquatics Swimming GRE Men 100m freestyle for sailors M Silver
In [26]: oo.set_index('Athlete',inplace=True)
In [27]: oo.head()
Out[27]:
City Edition Sport Discipline NOC Gender Event Event_gender Medal
Athlete
HAJOS, Alfred Athens 1896 Aquatics Swimming HUN Men 100m freestyle M Gold
HERSCHMANN, Otto Athens 1896 Aquatics Swimming AUT Men 100m freestyle M Silver
DRIVAS, Dimitrios Athens 1896 Aquatics Swimming GRE Men 100m freestyle for sailors M Bronze
MALOKINIS, Ioannis Athens 1896 Aquatics Swimming GRE Men 100m freestyle for sailors M Gold
CHASAPIS, Spiridon Athens 1896 Aquatics Swimming GRE Men 100m freestyle for sailors M Silver
In [28]: oo.reset_index(inplace=True)
In [ ]:
In [ ]:
LinkedIn

reset_index()

- DataFrame.**reset_index**(level=None, drop=False, inplace=False, …)
- Returns a DataFrame with the default (integer-based) index

CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [31]:
```
ath.reset_index(inplace=True)
ath.head()
```

Out[31]:

	Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
0	HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	100m freestyle	M	Gold
1	HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
2	DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
4	CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [ ]:

In [ ]:

In [ ]:

sort_index()

sort_index()

- DataFrame.**sort_index**(*axis=0, level=None, ascending=True, inplace=False, … by=None*)

- Sort objects by a label along the axis

Desktop/ExerciseFiles/Ch | Ch06 Indexing

localhost:8860/notebooks/Desktop/ExerciseFiles/Ch06/Ch06%20Indexing.ipynb#sort_index()

jupyter Ch06 Indexing Last Checkpoint: 6 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Markdown

Out[43]:

Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	100m freestyle	M	Gold
HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [44]:
```
ath.sort_index(inplace=True)
ath.head()
```

Out[44]:

Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
ÅABYE, Edgar	Paris	1900	Tug of War	Tug of War	ZZX	Men	tug of war	M	Gold
AALTONEN, Arvo Ossian	Antwerp	1920	Aquatics	Swimming	FIN	Men	200m breaststroke	M	Bronze
AALTONEN, Arvo Ossian	Antwerp	1920	Aquatics	Swimming	FIN	Men	400m breaststroke	M	Bronze
AALTONEN, Paavo Johannes	London	1948	Gymnastics	Artistic G.	FIN	Men	pommel horse	M	Gold
AALTONEN, Paavo Johannes	London	1948	Gymnastics	Artistic G.	FIN	Men	team competition	M	Gold

HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [45]:
```
ath.sort_index(inplace=True,ascending=False)
ath.head()
```

Out[45]:

Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
ÖSTRAND, Per-Olof	Helsinki	1952	Aquatics	Swimming	SWE	Men	400m freestyle	M	Bronze
ÖSTMO, Ole	Paris	1900	Shooting	Shooting	NOR	Men	army rifle, 300m, 3 positions	M	Bronze
ÖSTMO, Ole	Paris	1900	Shooting	Shooting	NOR	Men	free rifle, team	M	Silver
ÖSTMO, Ole	Paris	1900	Shooting	Shooting	NOR	Men	army rifle, 300m, standing	M	Silver
ÖSTMO, Ole	Paris	1900	Shooting	Shooting	NOR	Men	army rifle, 300m, prone	M	Bronze

In [ ]:

In [ ]:

loc[]

- DataFrame.**loc[]** / DataFrame.Series.**loc[]**
- A label-based indexer for selection by label
- loc[] will raise a KeyError when the items are not found

3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
4	CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [48]:
```
oo.set_index('Athlete',inplace=True)
```

In [49]:
```
oo.loc['BOLT, Usain']
```

Out[49]:

Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
BOLT, Usain	Beijing	2008	Athletics	Athletics	JAM	Men	100m	M	Gold
BOLT, Usain	Beijing	2008	Athletics	Athletics	JAM	Men	200m	M	Gold
BOLT, Usain	Beijing	2008	Athletics	Athletics	JAM	Men	4x100m relay	M	Gold

In [ ]:
```

```

iloc[...]

In [ ]:
```

```

iloc[]

- DataFrame.**iloc[]**
- iloc[] is primarily integer position based (from 0 to length-1 of the axis)
- Allows traditional Pythonic slicing

jupyter Ch06 Indexing Last Checkpoint: 17 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

	Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
0	HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	100m freestyle	M	Gold
1	HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
2	DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
4	CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [55]:
```
oo.iloc[1700]
```

Out[55]:
```
Athlete          RABOT, Pierre
City                    London
Edition                   1908
Sport                  Sailing
Discipline             Sailing
NOC                        FRA
Gender                     Men
Event                       6m
Event_gender                 X
Medal                   Bronze
Name: 1700, dtype: object
```

In [ ]:
```
oo.iloc[[1542, 2390,6000,15000]]
```

```
Medal                    Bronze
Name: 1700, dtype: object
```

In [56]:
```
oo.iloc[[1542, 2390,6000,15000]]
```

Out[56]:

	Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
1542	DUCKETT, Richard Louis	London	1908	Lacrosse	Lacrosse	CAN	Men	lacrosse	M	Gold
2390	SAASTAMOINEN, Eino	Stockholm	1912	Gymnastics	Artistic G.	FIN	Men	team, free system	M	Silver
6000	AGOSTONI, Carlo	Los Angeles	1932	Fencing	Fencing	ITA	Men	épée individual	M	Bronze
15000	JENSEN, Poul Richard Hoj	Montreal	1976	Sailing	Sailing	DEN	Men	fleet/match race keelboat open (Soling)	X	Gold

In [ ]:

In [ ]:

In [ ]:

In [57]:
```
oo.head()
```

Out[57]:

	Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
0	HAJOS, Alfred	Athens	1896	Aquatics	Swimming	HUN	Men	100m freestyle	M	Gold
1	HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
2	DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold
4	CHASAPIS, Spiridon	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Silver

In [58]:
```
oo.iloc[1:4]
```

Out[58]:

	Athlete	City	Edition	Sport	Discipline	NOC	Gender	Event	Event_gender	Medal
1	HERSCHMANN, Otto	Athens	1896	Aquatics	Swimming	AUT	Men	100m freestyle	M	Silver
2	DRIVAS, Dimitrios	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Bronze
3	MALOKINIS, Ioannis	Athens	1896	Aquatics	Swimming	GRE	Men	100m freestyle for sailors	M	Gold

In [ ]:

In [ ]:

jupyter Ch06 Indexing Solution Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

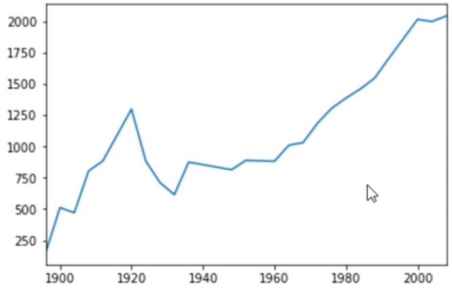
Code

In [ ]:

Plot the total number of medals awarded at each of the Olympic games throughout history.

In [12]:
```
oo.Edition.value_counts().sort_index().plot()
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x217494d9ba8>
```



In [ ]:

How Groupby Works

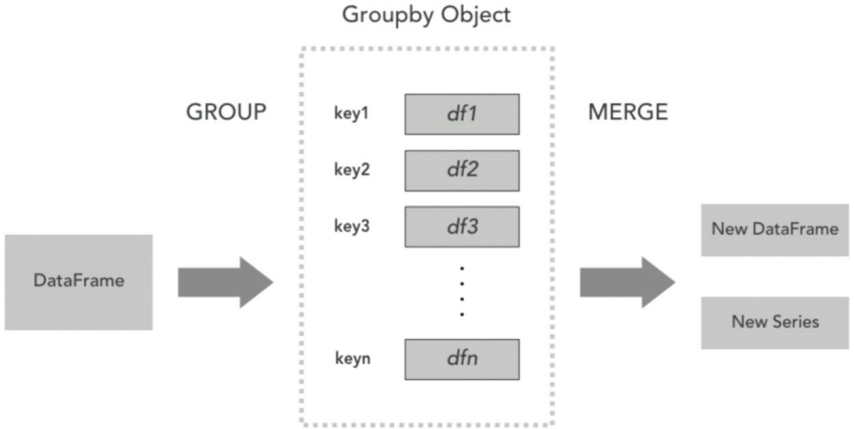
- Split a DataFrame into groups based on some criteria
- Apply a function to each group independently
- Combine the results into a DataFrame

Groupby

- pandas.DataFrame.**groupby**(by=None, axis=0, level=None, as_index=True, sort=True, group_keys=True, squeeze=False, **kwargs)

- Returns a groupby object

Groupby Visualization



jupyter Ch07 Groupby Last Checkpoint: 4 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Code

3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

The Groupby object

In [3]:
```
oo.groupby('Edition')
```

Out[3]:
```
<pandas.core.groupby.DataFrameGroupBy object at 0x0000022C6E28F5C0>
```

In [4]:
```
type(oo.groupby('Edition'))
```

Out[4]:
```
pandas.core.groupby.DataFrameGroupBy
```

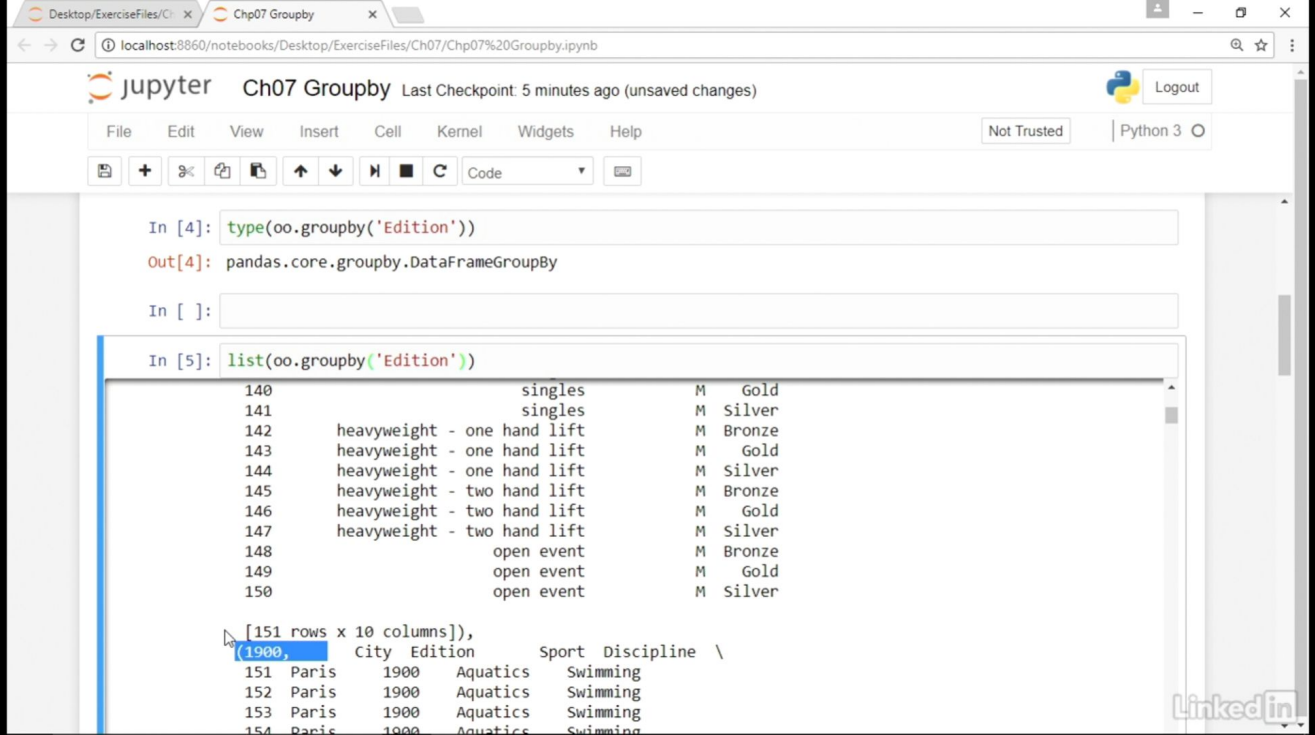
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:


Desktop/ExerciseFiles/Ch
Ch07 Groupby
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch07/Chp07%20Groupby.ipynb
jupyter
Ch07 Groupby
Last Checkpoint: 5 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Code
In [4]: type(oo.groupby('Edition'))
Out[4]: pandas.core.groupby.DataFrameGroupBy
In [ ]:
In [5]: list(oo.groupby('Edition'))
140 singles M Gold
141 singles M Silver
142 heavyweight - one hand lift M Bronze
143 heavyweight - one hand lift M Gold
144 heavyweight - one hand lift M Silver
145 heavyweight - two hand lift M Bronze
146 heavyweight - two hand lift M Gold
147 heavyweight - two hand lift M Silver
148 open event M Bronze
149 open event M Gold
150 open event M Silver
[151 rows x 10 columns]),
(1900, City Edition Sport Discipline \
151 Paris 1900 Aquatics Swimming
152 Paris 1900 Aquatics Swimming
153 Paris 1900 Aquatics Swimming

Iterate through a Group

- for key,group in DataFrame.groupby():

  print(key)

  print(group)

Iterating through groups

In [6]:
```
for group_key, group_value in oo.groupby('Edition'):
    print(group_key)
    print(group_value)
```

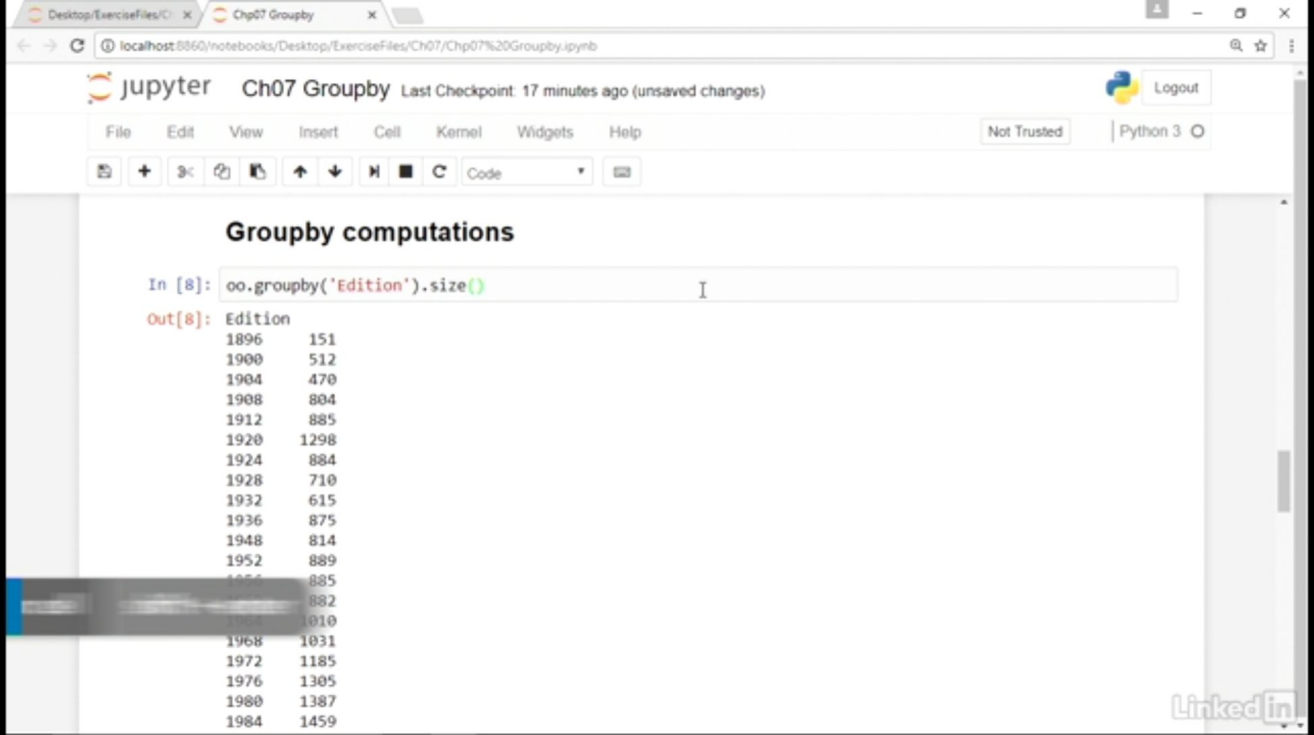
```
1896
       City  Edition          Sport       Discipline  \
0    Athens     1896       Aquatics         Swimming
1    Athens     1896       Aquatics         Swimming
2    Athens     1896       Aquatics         Swimming
3    Athens     1896       Aquatics         Swimming
4    Athens     1896       Aquatics         Swimming
5    Athens     1896       Aquatics         Swimming
6    Athens     1896       Aquatics         Swimming
7    Athens     1896       Aquatics         Swimming
8    Athens     1896       Aquatics         Swimming
9    Athens     1896       Aquatics         Swimming
10   Athens     1896       Aquatics         Swimming
11   Athens     1896      Athletics        Athletics
12   Athens     1896      Athletics        Athletics
13   Athens     1896      Athletics        Athletics
14   Athens     1896      Athletics        Athletics
15   Athens     1896      Athletics        Athletics
```

Groupby Computations

- GroupBy.size()
- GroupBy.count()
- GroupBy.first(), GroupBy.last()
- GroupBy.head(), GroupBy.tail()
- GroupBy.mean()
- GroupBy.max(), GroupBy.min()

On the Job

- agg() – multiple statistics in one calculation per group
- DataFrame.groupby(agg([…]))
- DataFrame.groupby(agg({ .. : […] }))


Desktop/ExerciseFiles/Ch
Ch07 Groupby
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch07/Ch07%20Groupby.ipynb
jupyter
Ch07 Groupby
Last Checkpoint: 17 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Code
Groupby computations
In [8]:
oo.groupby('Edition').size()
Out[8]:
Edition
1896 151
1900 512
1904 470
1908 804
1912 885
1920 1298
1924 884
1928 710
1932 615
1936 875
1948 814
1952 889
1968 1031
1972 1185
1976 1305
1980 1387
1984 1459

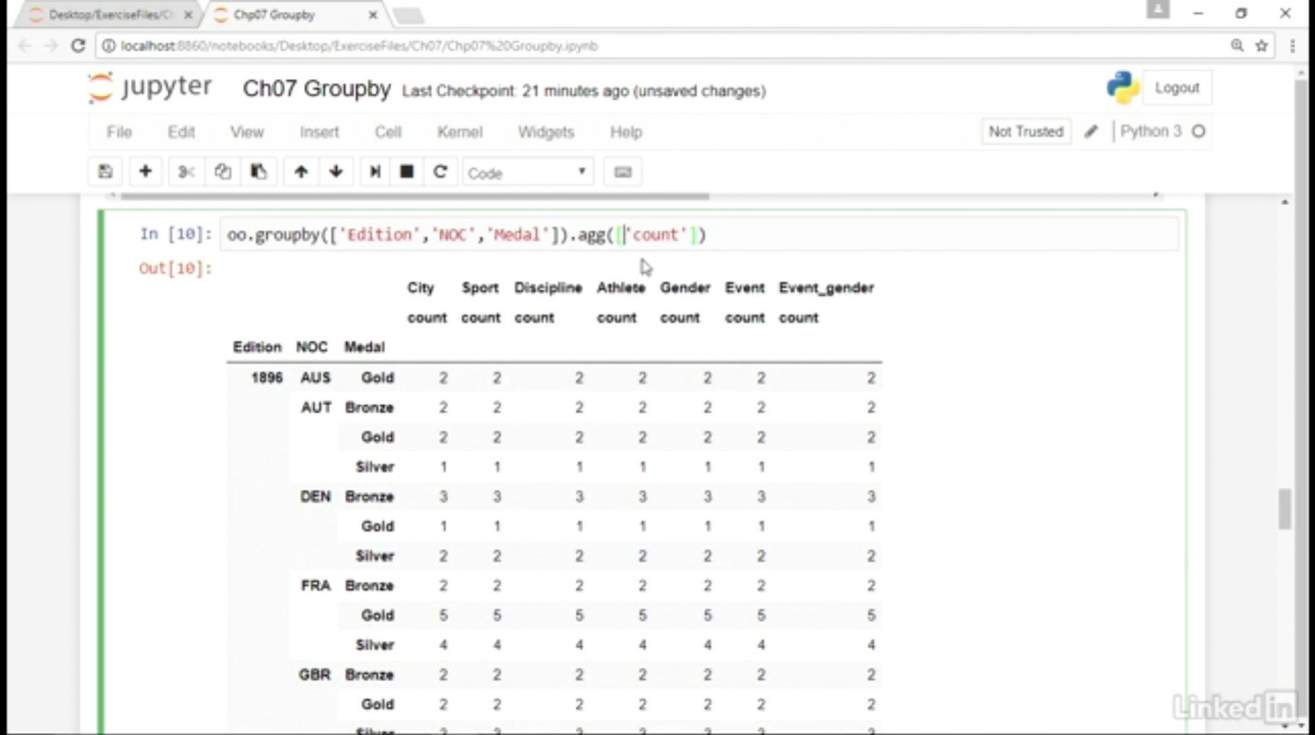
agg([...])

In [9]:
```
oo.groupby(['Edition','NOC','Medal']).agg(['min','max','count'])
```

Out[9]:

			City			Sport			Discipline			Athlete
			min	max	count	min	max	count	min	max	count	min
Edition	NOC	Medal										
1896	AUS	Gold	Athens	Athens	2	Athletics	Athletics	2	Athletics	Athletics	2	
	AUT	Bronze	Athens	Athens	2	Cycling	Cycling	2	Cycling Track	Cycling Track	2	
		Gold	Athens	Athens	2	Aquatics	Cycling	2	Cycling Track	Swimming	2	N
		Silver	Athens	Athens	1	Aquatics	Aquatics	1	Swimming	Swimming	1	HERS
	DEN	Bronze	Athens	Athens	3	Fencing	Shooting	3	Fencing	Shooting	3	
		Gold	Athens	Athens	1	Weightlifting	Weightlifting	1	Weightlifting	Weightlifting	1	
		Silver										

In [ ]:



Edition	NOC	Medal	City count	Sport count	Discipline count	Athlete count	Gender count	Event count	Event_gender count
1896	AUS	Gold	2	2	2	2	2	2	2
	AUT	Bronze	2	2	2	2	2	2	2
		Gold	2	2	2	2	2	2	2
		Silver	1	1	1	1	1	1	1
	DEN	Bronze	3	3	3	3	3	3	3
		Gold	1	1	1	1	1	1	1
		Silver	2	2	2	2	2	2	2
	FRA	Bronze	2	2	2	2	2	2	2
		Gold	5	5	5	5	5	5	5
		Silver	4	4	4	4	4	4	4
	GBR	Bronze	2	2	2	2	2	2	2
		Gold	2	2	2	2	2	2	2
		Silver	3	3	3	3	3	3	3

agg({ : [...]})

In [14]:
```
oo.groupby(['Edition','NOC','Medal']).agg({'Edition' :['min','max','count']})
```

Out[14]:

			Edition		
			min	max	count
Edition	NOC	Medal			
1896	AUS	Gold	1896	1896	2
	AUT	Bronze	1896	1896	2
		Gold	1896	1896	2
		Silver	1896	1896	1
	DEN	Bronze	1896	1896	3
		Gold	1896	1896	1
		Silver	1896	1896	2
	FRA	Bronze	1896	1896	2
		Gold	1896	1896	5
		Silver	1896	1896	4
	GBR	Bronze	1896	1896	2

Challenge

- Using groupby(), plot the total number of medals awarded at each of the Olympic games throughout history.

- Create a list showing the total number of medals won for each country over the history of the Olympics. For each country, include the year of the first and most recent Olympic medal wins.

Desktop/ExerciseFiles/Ch | Ch07 Groupby Solution

localhost:8860/notebooks/Desktop/ExerciseFiles/Ch07/Ch07%20Groupby%20Solution.ipynb

jupyter Ch07 Groupby Solution Last Checkpoint: 35 minutes ago (unsaved changes)

Logout

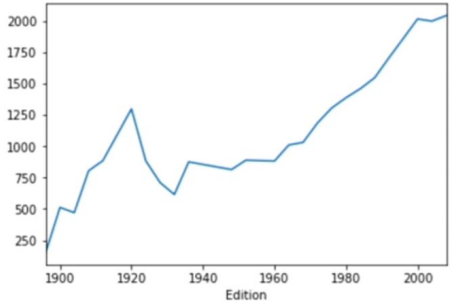
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

In [4]:
```
oo.groupby('Edition').size().plot()
```

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x276b32497b8>



In [ ]:

Create a list showing the total number of medals won for each country over the history of

Desktop/ExerciseFiles/Ch | Ch07 Groupby Solution

localhost:8860/notebooks/Desktop/ExerciseFiles/Ch07/Ch07%20Groupby%20Solution.ipynb

jupyter **Ch07 Groupby Solution** Last Checkpoint: 38 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

In [5]:
```
oo.groupby('NOC').agg(['count','min','max'])
```

UGA	6	Atlanta	Munich	6	1968	1996	6	Athletics	Boxing	6	...	Men	6
UKR	148	Athens	Sydney	148	1996	2008	148	Aquatics	Wrestling	148	...	Women	148
URS	2049	Helsinki	Tokyo	2049	1952	1988	2049	Aquatics	Wrestling	2049	...	Women	2049
URU	76	Amsterdam	Tokyo	76	1924	2000	76	Basketball	Rowing	76	...	Men	76
USA	4335	Amsterdam	Tokyo	4335	1896	2008	4335	Aquatics	Wrestling	4335	...	Women	4335
UZB	17	Athens	Sydney	17	1996	2008	17	Boxing	Wrestling	17	...	Women	17
VEN	11	Athens	Rome	11	1952	2008	11	Aquatics	Weightlifting	11	...	Women	11
VIE	2	Beijing	Sydney	2	2000	2008	2	Taekwondo	Weightlifting	2	...	Women	2
YUG	435	Amsterdam	Tokyo	435	1924	2000	435	Aquatics	Wrestling	435	...	Women	435

In [ ]:

In [ ]:

Reshaping

- Stacking
- Unstacking

0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

Athletes winning medals in Beijing Olympics 100m or 200m track event

In [3]:
```
mw = oo[(oo.Edition == 2008) & ( (oo.Event == '100m') | (oo.Event == '200m'))]
mw
```

Out[3]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
27551	Beijing	2008	Athletics	Athletics	DIX, Walter	USA	Men	100m	M	Bronze
27552	Beijing	2008	Athletics	Athletics	BOLT, Usain	JAM	Men	100m	M	Gold
27553	Beijing	2008	Athletics	Athletics	THOMPSON, Richard	TRI	Men	100m	M	Silver
27554	Beijing	2008	Athletics	Athletics	FRASER, Shelly-ann	JAM	Women	100m	W	Gold
27555	Beijing	2008	Athletics	Athletics	SIMPSON, Sherone	JAM	Women	100m	W	Silver
27556	Beijing	2008	Athletics	Athletics	STEWART, Kerron	JAM	Women	100m	W	Silver

	Discipline	Athletics	
	Event	100m	200m
NOC	Gender		
JAM	Men	1.0	1.0
	Women	3.0	2.0
TRI	Men	1.0	NaN
USA	Men	1.0	2.0
	Women	NaN	1.0

	Discipline	Athletics	
	Event	100m	200m
NOC	Gender		
JAM	Men	1.0	1.0
	Women	3.0	2.0
TRI	Men	1.0	NaN
USA	Men	1.0	2.0
	Women	NaN	1.0

DataFrame.stack()

Returns a DataFrame (or Series) that is taller.

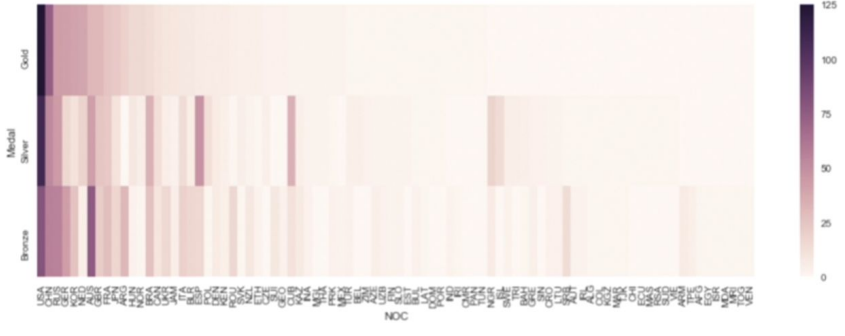
		Discipline	Athletics
NOC	Gender	Event	
JAM	Men	100m	1.0
		200m	1.0
	Women	100m	3.0
		200m	2.0
TRI	Men	100m	1.0
USA	Men	100m	1.0
		200m	2.0
	Women	200m	1.0

Data Visualizations

Data Visualizations

NOC	USA	CHN	RUS	GER	KOR	NED	AUS	GBR	FRA	JPN	...	VIE	ARM	TPE	AFG	EGY	ISR	MDA	MRI	TOG	VEN
Medal																					
Gold	125	74	43	42	41	40	31	31	25	23	...	0	0	0	0	0	0	0	0	0	0
Silver	109	53	44	16	11	18	42	25	23	11	...	1	0	0	0	0	0	0	0	0	0
Bronze	81	57	56	43	26	4	76	21	28	17	...	0	6	4	1	1	1	1	1	1	1

vs.



Seaborn with heatmaps

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
import seaborn as sns
```

In [3]:
```
oo = pd.read_csv('../data/olympics.csv',skiprows=4)
oo.head()
```

Out[3]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

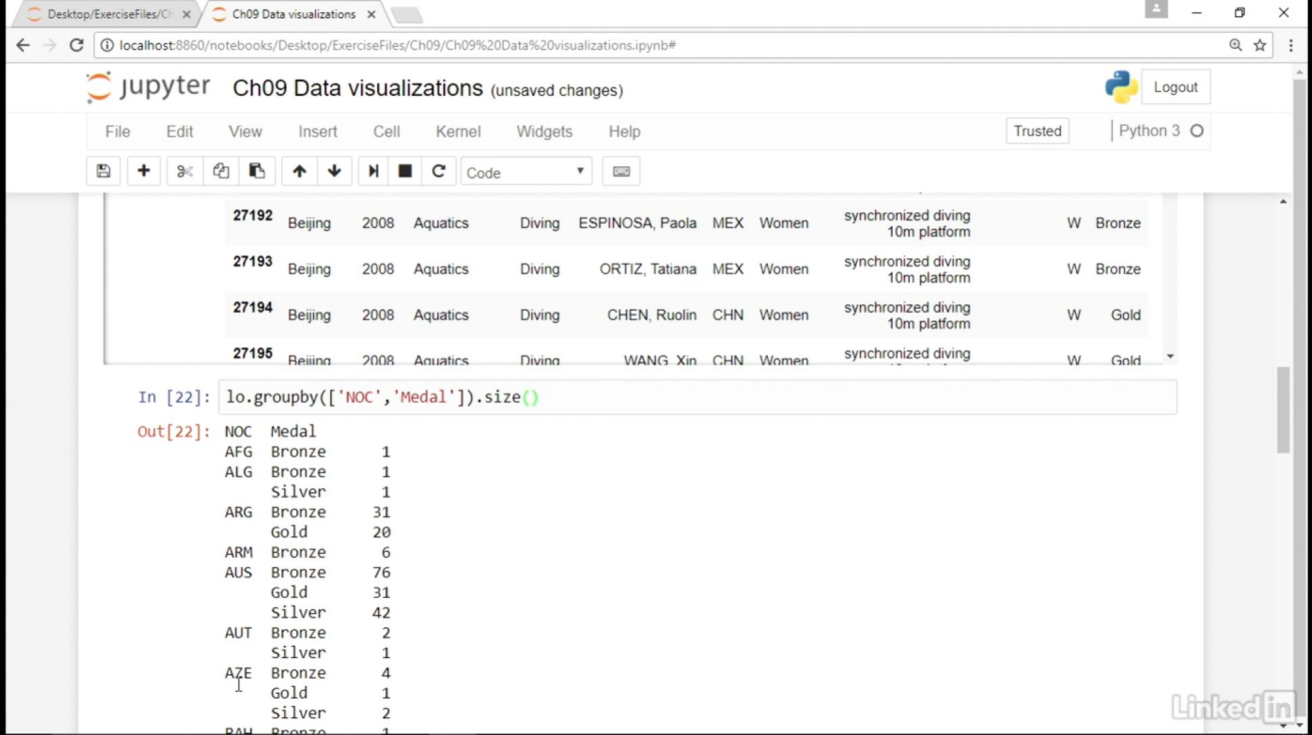
	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver

Using the Olympic dataset, present a summary of the total medals won by participating countries in the 2008 Olympics.

In [9]:
```
lo = oo[oo.Edition == 2008]
lo
```

Out[9]:

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
27174	Beijing	2008	Aquatics	Diving	GALPERIN, Gleb	RUS	Men	10m platform	M	Bronze
27175	Beijing	2008	Aquatics	Diving	MITCHAM, Matthew	AUS	Men	10m platform	M	Gold
27176	Beijing	2008	Aquatics	Diving	ZHOU, Luxin	CHN	Men	10m platform	M	Silver
27177	Beijing	2008	Aquatics	Diving	WANG, Xin	CHN	Women	10m platform	W	Bronze
27178	Beijing	2008	Aquatics	Diving	CHEN, Ruolin	CHN	Women	10m platform	W	Gold


Desktop/ExerciseFiles/C
Ch09 Data visualizations
localhost:8860/notebooks/Desktop/ExerciseFiles/Ch09/Ch09%20Data%20visualizations.ipynb#
jupyter
Ch09 Data visualizations (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Code
27192 Beijing 2008 Aquatics Diving ESPINOSA, Paola MEX Women synchronized diving 10m platform W Bronze
27193 Beijing 2008 Aquatics Diving ORTIZ, Tatiana MEX Women synchronized diving 10m platform W Bronze
27194 Beijing 2008 Aquatics Diving CHEN, Ruolin CHN Women synchronized diving 10m platform W Gold
27195 Beijing 2008 Aquatics Diving WANG, Xin CHN Women synchronized diving W Gold
In [22]: lo.groupby(['NOC','Medal']).size()
Out[22]:
NOC Medal
AFG Bronze 1
ALG Bronze 1
Silver 1
ARG Bronze 31
Gold 20
ARM Bronze 6
AUS Bronze 76
Gold 31
Silver 42
AUT Bronze 2
Silver 1
AZE Bronze 4
Gold 1
Silver 2
BAH Bronze 1

jupyter Ch09 Data visualizations (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

27192	Beijing	2008	Aquatics	Diving	ESPINOSA, Paola	MEX	Women	synchronized diving 10m platform	W	Bronze
27193	Beijing	2008	Aquatics	Diving	ORTIZ, Tatiana	MEX	Women	synchronized diving 10m platform	W	Bronze
27194	Beijing	2008	Aquatics	Diving	CHEN, Ruolin	CHN	Women	synchronized diving 10m platform	W	Gold
27195	Beijing	2008	Aquatics	Diving	WANG, Xin	CHN	Women	synchronized diving	W	Gold

In [23]:
```
lo.groupby(['NOC','Medal']).size().unstack('Medal',fill_value=0)
```

Out[23]:

NOC	Bronze	Gold	Silver
AFG	1	0	0
ALG	1	0	1
ARG	31	20	0
ARM	6	0	0
AUS	76	31	42
AUT	2	0	1
AZE	4	1	2
BAH	1	0	4

27192	Beijing	2008	Aquatics	Diving	ESPINOSA, Paola	MEX	Women	synchronized diving 10m platform	W	Bronze
27193	Beijing	2008	Aquatics	Diving	ORTIZ, Tatiana	MEX	Women	synchronized diving 10m platform	W	Bronze
27194	Beijing	2008	Aquatics	Diving	CHEN, Ruolin	CHN	Women	synchronized diving 10m platform	W	Gold
27195	Beijing	2008	Aquatics	Diving	WANG, Xin	CHN	Women	synchronized diving	W	Gold

In [24]:
```
g = lo.groupby(['NOC','Medal']).size().unstack('Medal',fill_value=0)
g.sort_values('Gold','Silver','Bronze')
```

Out[24]:

NOC	Bronze	Gold	Silver
AFG	1	0	0
ALG	1	0	1
ARG	31	20	0
ARM	6	0	0
AUS	76	31	42
AUT	2	0	1
AZE	4	1	2

jupyter Ch09 Data visualizations (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

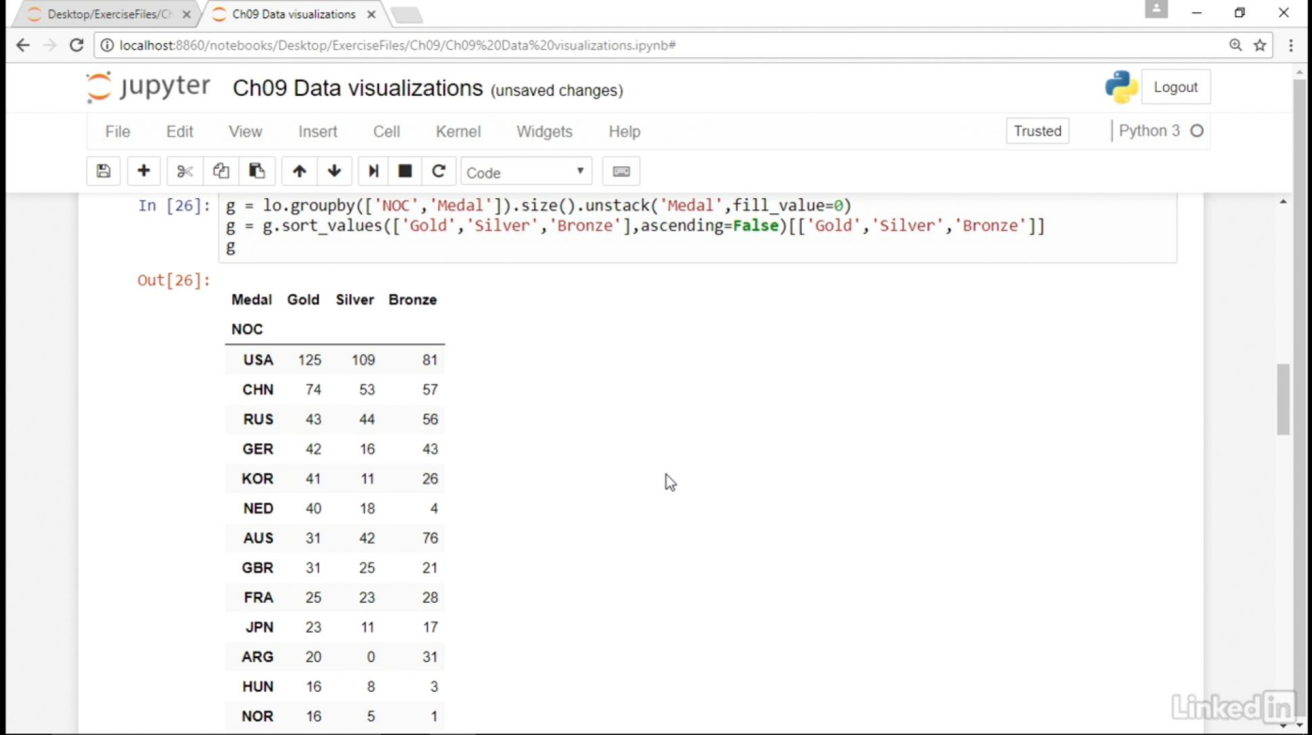
Code

27192	Beijing	2008	Aquatics	Diving	ESPINOSA, Paola	MEX	Women	synchronized diving 10m platform	W	Bronze
27193	Beijing	2008	Aquatics	Diving	ORTIZ, Tatiana	MEX	Women	synchronized diving 10m platform	W	Bronze
27194	Beijing	2008	Aquatics	Diving	CHEN, Ruolin	CHN	Women	synchronized diving 10m platform	W	Gold
27195	Beijing	2008	Aquatics	Diving	WANG, Xin	CHN	Women	synchronized diving	W	Gold

In [25]:
```
g = lo.groupby(['NOC','Medal']).size().unstack('Medal',fill_value=0)
g.sort_values(['Gold','Silver','Bronze'],ascending=False)
```

Out[25]:

Medal	Bronze	Gold	Silver
NOC			
USA	81	125	109
CHN	57	74	53
RUS	56	43	44
GER	43	42	16
KOR	26	41	11
NED	4	40	18
AUS	76	31	42



Medal	Gold	Silver	Bronze
NOC			
USA	125	109	81
CHN	74	53	57
RUS	43	44	56
GER	42	16	43
KOR	41	11	26
NED	40	18	4
AUS	31	42	76
GBR	31	25	21
FRA	25	23	28
JPN	23	11	17
ARG	20	0	31
HUN	16	8	3
NOR	16	5	1

jupyter Ch09 Data visualizations (unsaved changes) Logout

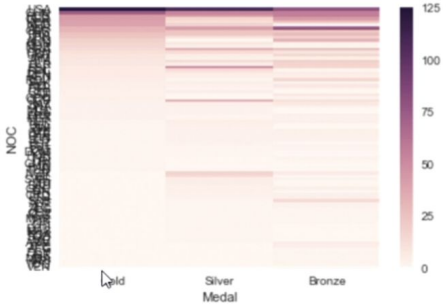
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

86 rows × 3 columns

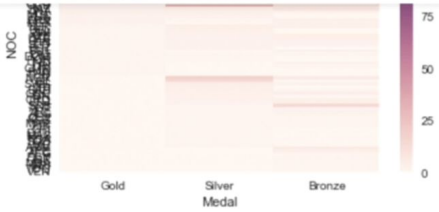
In [27]:
```
sns.heatmap(g)
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x13298f63f60>



In [ ]:

In [ ]:



In [28]:
```
g = g.transpose()
g
```

Out[28]:

NOC	USA	CHN	RUS	GER	KOR	NED	AUS	GBR	FRA	JPN	...	VIE	ARM	TPE	AFG	EGY	ISR	MDA	MRI	TOG
Medal																				
Gold	125	74	43	42	41	40	31	31	25	23	...	0	0	0	0	0	0	0	0	0
Silver	109	53	44	16	11	18	42	25	23	11	...	1	0	0	0	0	0	0	0	0
Bronze	81	57	56	43	26	4	76	21	28	17	...	0	6	4	1	1	1	1	1	1

3 rows × 86 columns

jupyter Ch09 Data visualizations (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

In [28]:
```
g = g.transpose()
g
```

Out[28]:

NOC	USA	CHN	RUS	GER	KOR	NED	AUS	GBR	FRA	JPN	...	VIE	ARM	TPE	AFG	EGY	ISR	MDA	MRI	TOG
Medal																				
Gold	125	74	43	42	41	40	31	31	25	23	...	0	0	0	0	0	0	0	0	0
Silver	109	53	44	16	11	18	42	25	23	11	...	1	0	0	0	0	0	0	0	0
Bronze	81	57	56	43	26	4	76	21	28	17	...	0	6	4	1	1	1	1	1	1

3 rows × 86 columns

In [ ]:
```
plt.figure(figsize=(16,5))
sns.heatmap(g)
```

In [ ]:

Desktop/ExerciseFiles/Ch | Ch09 Data visualizations

localhost:8860/notebooks/Desktop/ExerciseFiles/Ch09/Ch09%20Data%20visualizations.ipynb#

jupyter Ch09 Data visualizations (unsaved changes)

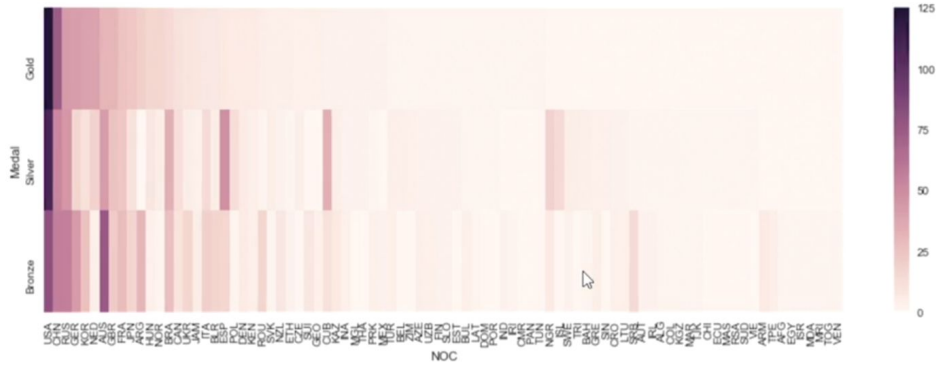
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

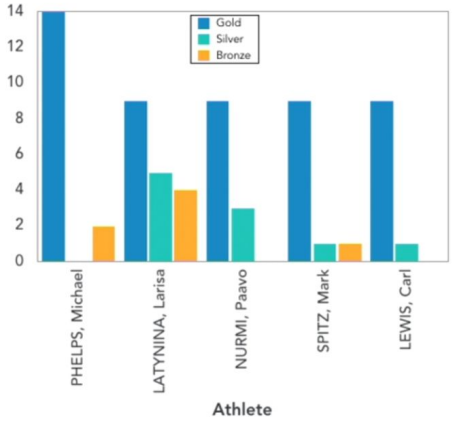
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x13298e6b7b8>



In [ ]:

In [ ]:

Create Your Own Colormap



Or