PANDAS FOR MACHINE LEARNING

For you!

Sadia Khatun

TABLE OF CONTENTS

Chapter 1: Loading Data

NOTE: df and pd just veriables, you can use any name.

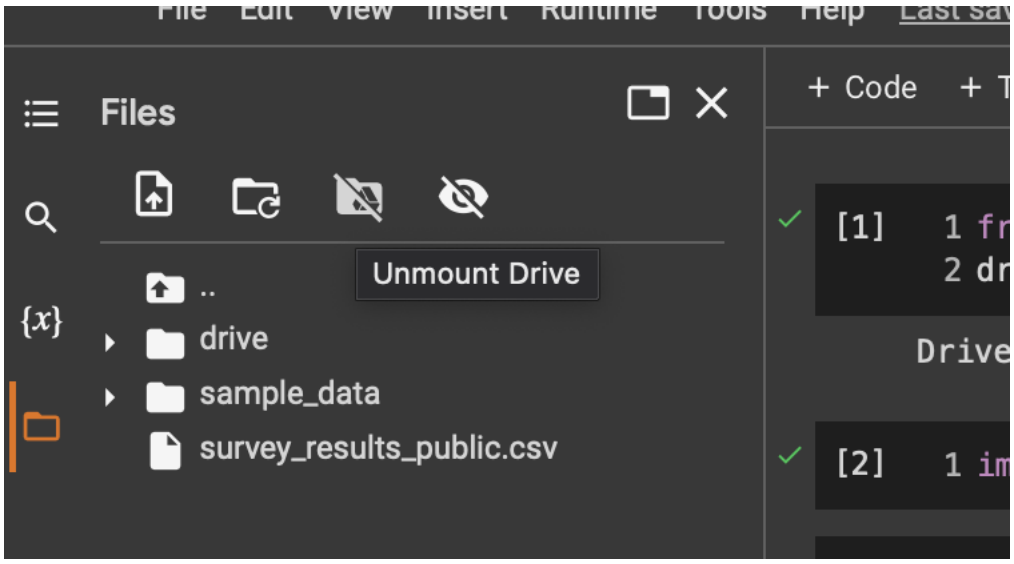
This section covers

Pd.read_csv(), df.shape, df.info(), pd.set_option(), df.head(),df.tail()

Mount Google Drive:

```
[1]  1 from google.colab import drive
     2 drive.mount('/content/drive/')

Drive already mounted at /content/drive/; to attempt to forcibly remount, call drive.mount("/content/d
```

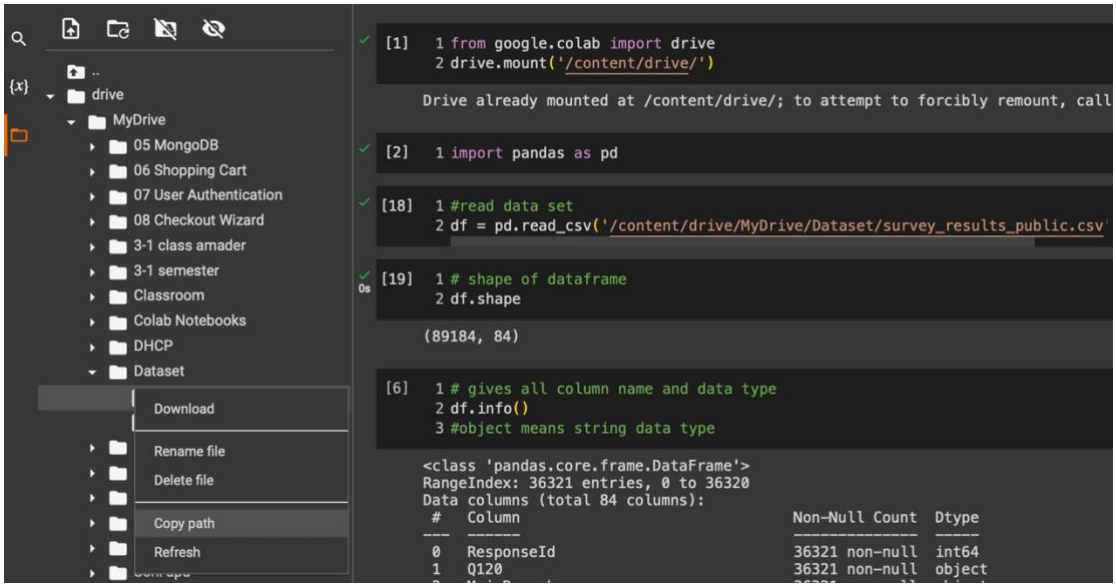


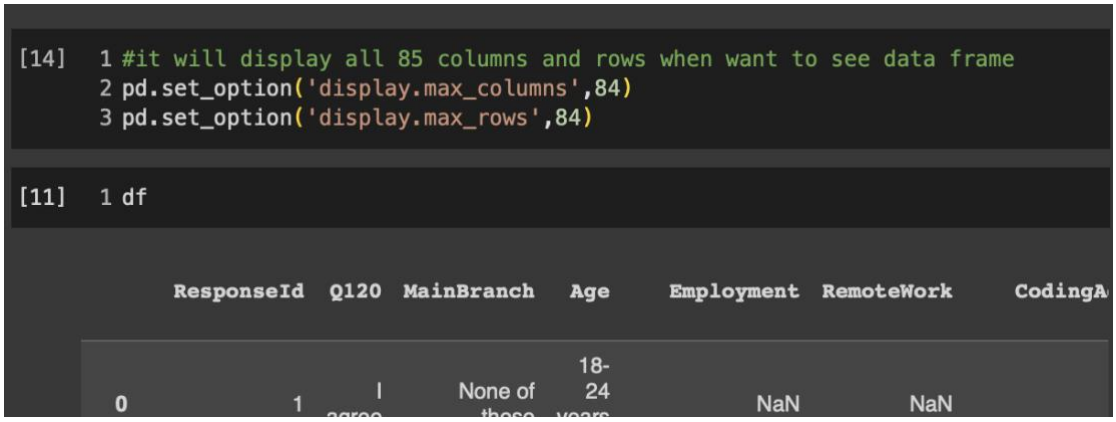
Import pandas:

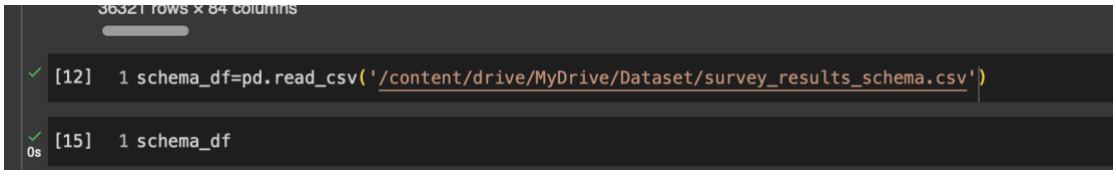
```
[2]  1 import pandas as pd
```

Read .csv data set:

Copy path and put in it pd.read_csv(‘ ’)


drive
MyDrive
05 MongoDB
06 Shopping Cart
07 User Authentication
08 Checkout Wizard
3-1 class amader
3-1 semester
Classroom
Colab Notebooks
DHCP
Dataset
Download
Rename file
Delete file
Copy path
Refresh
[1] 1 from google.colab import drive
2 drive.mount('/content/drive/')
Drive already mounted at /content/drive/; to attempt to forcibly remount, call
[2] 1 import pandas as pd
[18] 1 #read data set
2 df = pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_public.csv'
[19] 1 # shape of dataframe
2 df.shape
(89184, 84)
[6] 1 # gives all column name and data type
2 df.info()
3 #object means string data type
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36321 entries, 0 to 36320
Data columns (total 84 columns):
Column Non-Null Count Dtype
0 ResponseId 36321 non-null int64
1 Q120 36321 non-null object


[14] 1 #it will display all 85 columns and rows when want to see data frame
2 pd.set_option('display.max_columns',84)
3 pd.set_option('display.max_rows',84)
[11] 1 df
ResponseId Q120 MainBranch Age Employment RemoteWork CodingA
0 1 I None of 18- NaN NaN


36321 rows × 84 columns
[12] 1 schema_df=pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_schema.csv')
[15] 1 schema_df


[17] 1 #last 10 instance
2 df.tail(10)


[16] 1 #1st 10 data show
2 df.head(10)

Chapter 2: Data frame and Series Basics- Selecting Rows and Columns

This section covers

df.iloc[], df.loc, df.columns, df[].value_counts()

slicing, access row and column, python dictionary elc.

python dictionary

```
[32]  1 # key: value
      2 #method 1
      3 person = {
      4     "first":"sadia",
      5     "last" :"khatun",
      6     "email":"sadiasultana44444a@gmail.com"
      7 }
      8 #method 2
      9 # key: list of value
     10 people = {
     11     "first":["sadia"],
     12     "last" :["khatun"],
     13     "email":["sadiasultana44444a@gmail.com"]
     14 }
     15
     16 #method 2
     17 # key: list of multiple  values
     18 #email is column name  and this column have  two value
     19 #so here key means column name
     20 people = {
     21     "first":["sadia","samira"],
     22     "last" :["khatun","samira"],
     23     "email":["sadiasultana44444a@gmail.com","samira1234@.gmail.com"]
     24 }
```

[41]
```
1 import pandas as pd
```

[34]
```
1 #data frame just like dictionary
2 people['email']
```

```
['sadiasultana44444a@gmail.com', 'samira1234@.gmail.com']
```

[35]
```
1 #create data frame from this dictionary
2 dataframe=pd.DataFrame(people)
3 dataframe
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com

[36]
```
1 # want to access email column
2 dataframe['email']
```

```
0    sadiasultana44444a@gmail.com
1          samira1234@.gmail.com
Name: email, dtype: object
```

[39]
```
1 type(dataframe['email'])
2 # series like 1-DIM array have also index
```

```
pandas.core.series.Series
```

[40]
```
1 #we also access eamil column like below command
2 dataframe.email
```

```
0    sadiasultana44444a@gmail.com
1          samira1234@.gmail.com
Name: email, dtype: object
```

[43]
```
1 #by this we can access multiple column just like we access value in array
2 dataframe[['last','email']]
```

	last	email
0	khatun	sadiasultana44444a@gmail.com
1	samira	samira1234@.gmail.com

```
[45]  1 # if we want to see what columns are available
      2 dataframe.columns
```

```
Index(['first', 'last', 'email'], dtype='object')
```

access Rows

loc,by loc we can search values by column label

iloc ,by iloc we can search value by column integer or column index

```
[46]  1 #give information about 1st row
      2 dataframe.iloc[0]
```

```
first                          sadia
last                          khatun
email    sadiasultana44444a@gmail.com
Name: 0, dtype: object
```

```
[47]  1 # acess multiple rows
      2 dataframe.iloc[[0,1]]
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com

```
[48]  1 # we want to acess email address of first 2 row (row,column)
      2 dataframe.iloc[[0,1],2]
```

```
0    sadiasultana44444a@gmail.com
1           samira1234@.gmail.com
Name: email, dtype: object
```

```
[49]  1 #we and last name of 2nd row
      2 dataframe.iloc[1,1]
```

```
'samira'
```

```
[50]  1 #we want 1st name and email address of 2nd person
      2
      3 dataframe.iloc[1,[0,2]]
```

```
first                  samira
email    samira1234@.gmail.com
Name: 1, dtype: object
```

```
[51]  1 dataframe.loc[[0,1]]
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com

```
[52]  1 dataframe.loc[[0,1],'email']
```

```
0    sadiasultana44444a@gmail.com
1           samira1234@.gmail.com
Name: email, dtype: object
```

```
[53]  1 #notice that the column display in the order that we used in
      2 #our list up here within loc
      3 dataframe.loc[[0,1],['email','last']]
```

	email	last
0	sadiasultana44444a@gmail.com	khatun
1	samira1234@.gmail.com	samira

```
[55]  1 #correct in iloc
      2 dataframe.iloc[[0,1],[0,1]]
```

	first	last
0	sadia	khatun
1	samira	samira

Notice the error:

```
1 # incorrect in loc
2 # in loc we have to pass column name
3 dataframe.loc[[0,1],[0,1]]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-57-201313ed706f> in <cell line: 3>()
      1 # incorrect in loc
      2 # in loc we have to pass column name
----> 3 dataframe.loc[[0,1],[0,1]]
```

back to stack overflow data set

```
[58]  1 df.shape
```

```
(89184, 84)
```

```
[64]  1 df.columns
```

```
Index(['ResponseId', 'Q120', 'MainBranch', 'Age', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'TechList', 'BuyNewTool', 'Country',
       'Currency', 'CompTotal', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysPersonal use',
       'OpSysProfessional use', 'OfficeStackAsyncHaveWorkedWith',
       'OfficeStackAsyncWantToWorkWith', 'OfficeStackSyncHaveWorkedWith',
       'OfficeStackSyncWantToWorkWith', 'AISearchHaveWorkedWith',
       'AISearchWantToWorkWith', 'AIDevHaveWorkedWith', 'AIDevWantToWorkWith',
       'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq', 'SOComm',
       'SOAI', 'AISelect', 'AISent', 'AIAcc', 'AIBen',
       'AIToolInterested in Using', 'AIToolCurrently Using',
       'AIToolNot interested in Using', 'AINextVery different',
       'AINextNeither different nor similar', 'AINextSomewhat similar',
       'AINextVery similar', 'AINextSomewhat different', 'TBranch', 'ICorPM',
       'WorkExp', 'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Knowledge_8',
       'Frequency_1', 'Frequency_2', 'Frequency_3', 'TimeSearching',
       'TimeAnswering', 'ProfessionalTech', 'Industry', 'SurveyLength',
       'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

```
[63]  1 # we want to grab all responses from  Age column
      2 df['Age']
```

```
0           18-24 years old
1           25-34 years old
2           45-54 years old
3           25-34 years old
4           25-34 years old
                 ...
89179       25-34 years old
89180       18-24 years old
89181     Prefer not to say
89182    Under 18 years old
89183       35-44 years old
Name: Age, Length: 89184, dtype: object
```

```
[ ]  1 # grab 1st row and Age column value
     2 #i prefer use loc beacuse i can use label means column
     3 #name to grab this column
     4 df.loc[0,'Age']
```

```
'18-24 years old'
```

```
     1 #1st person entire survey result
     2 df.loc[0]
```

```
ResponseId                                      1
Q120                                      I agree
MainBranch                          None of these
Age                                 18-24 years old
Employment                                      NaN
RemoteWork                                      NaN
CodingActivities                                NaN
EdLevel                                         NaN
LearnCode                                       NaN
LearnCodeOnline                                 NaN
LearnCodeCoursesCert                            NaN
```

```
[69]  1 #1st 3 person age
      2
      3 df.loc[[0,1,2],'Age']
```

```
0    18-24 years old
1    25-34 years old
2    45-54 years old
Name: Age, dtype: object
```

```
[70]  1 #slicing is same as list slicing only last values is going to be inclusive
      2 #note: using slicing we don't wrap these in brackets
      3 #error
      4 df.loc[[0:2],'Age']
```

```
  File "<ipython-input-70-238d91520029>", line 4
    df.loc[[0:2],'Age']
           ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
[71]  1 #correct way
      2 df.loc[0:2,'Age']
```

```
0    18-24 years old
1    25-34 years old
2    45-54 years old
Name: Age, dtype: object
```

```
[72]  1 #correct way
      2 #that slice from age to Learncode
      3 #and 1st three row
      4 df.loc[0:2,'Age':'LearnCode']
```

	Age	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode
0	18-24 years old	NaN	NaN	NaN	NaN	NaN
1	25-34 years old	Employed, full-time	Remote	Hobby;Contribute to open-source projects;Boots...	Bachelor's degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;Colleague;Friend or fam...
2	45-54 years old	Employed, full-time	Hybrid (some remote, some in-person)	Hobby;Professional development or self-paced l...	Bachelor's degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;Colleague;On the job tr...

Chapter 3: Indexes - How To Set, Reset And Use Indexes

index

	Fast name	Last name	email
0			
1			
2			
3			

This section covers

#df.set_index(),df.reset_index()

df.sort_index()

NOTE: For permanent change we use inplace= True

Indexes-how to set, reset and use indexes

```
[97]  1 dataframe
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com

```
[76]  1 dataframe['email']
```

```
0    sadiasultana44444a@gmail.com
1           samira1234@.gmail.com
Name: email, dtype: object
```

```
[98]  1 #set email as index
      2 dataframe.set_index('email')
```

email	first	last
sadiasultana44444a@gmail.com	sadia	khatun
samira1234@.gmail.com	samira	samira

```
[78]  1 #actual daraframe did not change
      2 dataframe
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com

```
[92]  1 #if we want to have those changes carry over into future
      2
      3 dataframe.set_index('email',inplace=True)
      4
```

```
[81]  1 dataframe
```

email	first	last
sadiasultana44444a@gmail.com	sadia	khatun
samira1234@.gmail.com	samira	samira

```
[84]  1 #our email column value now become index
      2 #so we can access row value by passing email address
      3 dataframe.loc['sadiasultana44444a@gmail.com']
```

```
first     sadia
last     khatun
Name: sadiasultana44444a@gmail.com, dtype: object
```

```
[85]  1 # now our index no longer in integer value
      2 # so if we use below command we get error
      3 dataframe.loc[0]
```

```
[86]  1 #but iloc not give error if we need to use integer index
      2 # then we can just use iloc
      3 dataframe.iloc[0]
```

```
first     sadia
last     khatun
Name: sadiasultana44444a@gmail.com, dtype: object
```

```
[99]  1 # we can also reset what we change
      2
      3 dataframe.reset_index(inplace=True)
      4
```

```
[100] 1 dataframe
```

	index	first	last	email
0	0	sadia	khatun	sadiasultana44444a@gmail.com
1	1	samira	samira	samira1234@.gmail.com

how to apply above learning in real world dataset

```
[104] 1 #make RespondeId is index
      2 df=pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_public.csv', index_col='ResponseId')
```

```
[105] 1 df.head()
```

ResponseId	Q120	MainBranch	Age	Employment	RemoteWork	CodingActivities	EdLevel	Learn
1	I agree	None of these	18-24 years old	NaN	NaN	NaN	NaN	

```
[108] 1 #we make column name as index of schema_df  so that we can search info about a column name
      2 # directly sothat we do not need to look for index of a particular column name
      3 schema_df= pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_schema.csv',index_col='qname')
```

```
[109] 1 schema_df
```

qname	qid	question	force_resp	type	selector
SOAI	QID325	Artificial Intelligence (AI) tools have gained...	False	TE	SL
AISelect	QID314	Do you currently use AI tools in your developm...	True	MC	SAVR
AISent	QID315	How favorable is your stance on using AI tools...	False	MC	SAVR
AIAcc	QID324	For the AI tools you use as part of your devel...	False	MC	MAVR
AIBen	QID316	How much do you trust the accuracy of the outp...	False	MC	SAVR
AITool	QID319	Which parts of your development workflow are y...	False	Matrix	Likert
AINext	QID320	Thinking about how your workflow and process c...	False	Matrix	Likert

```
[110] 1 #now we want to know what AIOpen column name means
      2 schema_df.loc['AIOpen']
```

```
qid                                                    QID321
question      Please describe how you would expect your work...
force_resp                                             False
type                                                      TE
selector                                                  SL
Name: AIOpen, dtype: object
```

```
[111] 1 #if we want to read full question text
      2 schema_df.loc['AIOpen','question']
```

```
'Please describe how you would expect your workflow to be different, if at all, in 1 year as a result of AI
advancements.'
```

```
[112] 1 schema_df.sort_index()
```

qname	qid	question	force_resp	type	selector
OpSys	QID71	What is the primary operating system in...	False	Matrix	Likert
OrgSize	QID29	Approximately how many people are employed by ...	False	MC	SAVR
Platform	QID263	Which cloud platforms have you done ext...	False	Matrix	Likert
ProfessionalTech	QID304	My company has:	False	MC	MAVR
PurchaseInfluence	QID278	What level of influence do you, personally, ha...	False	MC	SAVR
Q120	QID312		True	MC	SAVR
Q310	QID310	You	False	DB	TB

```
[113] 1 schema_df.sort_index(ascending=False)
```

qname	qid	question	force_resp	type	selector
YearsCodePro	QID34	NOT including education, how many years have y...	False	MC	DL
YearsCode	QID32	Including any education, how many years have y...	False	MC	DL
WorkExp	QID288	How many years of working experience do you have?	False	Slider	HSLIDER
Webframe	QID264	Which web frameworks and web technologies</...	False	Matrix	Likert
ToolsTech	QID275	Which developer tools for compi...	False	Matrix	Likert

NOTE: if we want to carry change of sorting command, we just need to use inpalce=True like before..

```
[114]  1 schema_df.sort_index(inplace=True)
       2
```

```
[115]  1 schema_df
```

qname	qid	question	force_resp	type	selector
AIAcc	QID324	For the AI tools you use as part of your devel...	False	MC	MAVR
AIBen	QID316	How much do you trust the accuracy of the outp...	False	MC	SAVR
AIDev	QID328	Which AI-powered developer tools did yo...	False	Matrix	Likert
AINext	QID320	Thinking about how your workflow and process c...	False	Matrix	Likert
AIOpen	QID321	Please describe how you would expect your work...	False	TE	SL
AISearch	QID327	Which AI-powered search tools did you u...	False	Matrix	Likert
AISelect	QID314	Do you currently use AI tools in your developm...	True	MC	SAVR
AISent	QID315	How favorable is your stance on using AI tools...	False	MC	SAVR
AITool	QID319	Which parts of your development workflow are y...	False	Matrix	Likert
Age	QID127	What is your age? *	True	MC	MAVR
BuyNewTool	QID279	When buying a new tool or software, how do you...	False	MC	MAVR
CodingActivities	QID297	Which of the following best describes the code...	False	MC	MAVR

Chapter 4: Filtering – Using Conditionals To Filter Rows And Columns

This section covers

.isin(), .str.contains() , and(&), or(|), not(~)

```
[66]  1 import pandas as pd
```

```
[69]  1 people = {
      2     "first":["sadia","samira","sadia"],
      3     "last" :["khatun","samira","sultana"],
      4     "email":["sadiasultana44444a@gmail.com","samira1234@.gmail.com","sadia12345@gmail.com"]
      5 }
```

```
[70]  1 df=pd.DataFrame(people)
```

```
[71]  1 df
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[72]  1 #we  want to find person which frist name is sadia
      2 #we just get true flase value
      3 df['first']=='sadia'
```

```
0     True
1    False
2     True
Name: first, dtype: bool
```

```
[73]  1 # Note: filter is key word in python so use filt as veriable
      2 # apply this filter to dataframe
      3 filt=df['first']=='sadia'
```

```
[74]  1 df[filt]
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[75]  1 #method 2
      2 #we can do it directly
      3 df[df['first']=='sadia']
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
2	sadia	sultana	sadia12345@gmail.com

[76]
```
# we can also use loc
df.loc[filt]
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
2	sadia	sultana	sadia12345@gmail.com

[77]
```
# we can also use loc
# we just want email address whose have same first name
df.loc[filt,'email']
```

```
0    sadiasultana44444a@gmail.com
2           sadia12345@gmail.com
Name: email, dtype: object
```

[84]
```
# we can not use built-in and, or operator in filter
# &=and, |=or

# we want to those values whoes frist name is sadia and last name is sultana

filt=(df['first']=='sadia') & (df['last']=='sultana')

```

[87]
```
df.loc[filt]
```

	first	last	email
2	sadia	sultana	sadia12345@gmail.com

```
df.loc[filt,'email']
```

```
2    sadia12345@gmail.com
Name: email, dtype: object
```

```
df[(df['first']=='sadia') & (df['last']=='sultana')]
```

	first	last	email
2	sadia	sultana	sadia12345@gmail.com

[89]
```
#we want last name equal to sultana or  1st name equal to samira

filt=(df['last']=='sultana') | (df['first']=='samira')
```

[90]
```
df.loc[filt]
```

	first	last	email
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
# if  we want to oposite the filter just use ~
df.loc[~filt]
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com

Back to our survey data apply above knowledge

```
[93] df=pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_public.csv',index_col='ResponseId')
     schema_df=pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_schema.csv',index_col='qname')
```

```
df.head()
```

ResponseId	Q120	MainBranch	Age	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode
1	I agree	None of these	18-24 years old	NaN	NaN	NaN	NaN	NaN
2	I	I am a developer by	25-34	Employed, full-	Remote	Hobby;Contribute to open-source	Bachelor's degree (B.A.,	Books / Physical media;Colleague;Friend

```
[109] #
      high_salary=(df['ConvertedCompYearly']>2220000)
```

```
[110] df.loc[high_salary,['Country','Age','Currency','LearnCodeCoursesCert','LearnCodeOnline','ConvertedCompYearly']]
```

ResponseId	Country	Age	Currency	LearnCodeCoursesCert	LearnCodeOnline	ConvertedCompYearly
771	United States of America	25-34 years old	USD\tUnited States dollar	NaN	Formal documentation provided by the owner of ...	9000000.0
10519	Thailand	45-54 years old	THB\tThai baht	NaN	NaN	2875692.0

```
[111] #people from this country
      countries=['United States','India','United Kindom','Germany','canada']
      filt=df['Country'].isin(countries)
```

```
[112] df.loc[filt,'Country']
```

```
ResponseId
10         India
16       Germany
20       Germany
22       Germany
24       Germany
```

```
[113] 1 #now i want to see Respondent  from bangladesh
      2 filt=df['Country']=='Bangladesh'
```

```
[118] 1 df.loc[filt,['Age','LearnCodeOnline','ConvertedCompYearly']]
      2
```

ResponseId	Age	LearnCodeOnline	ConvertedCompYearly
40	25-34 years old	Formal documentation provided by the owner of ...	14503.0
208	25-34 years old	Formal documentation provided by the owner of ...	NaN
256	18-24 years old	Formal documentation provided by the owner of ...	NaN
275	25-34 years old	Formal documentation provided by the owner of ...	NaN

```
[120] 1 df['LearnCode']
```

```
ResponseId
1                                                     NaN
2        Books / Physical media;Colleague;Friend or fam...
3        Books / Physical media;Colleague;On the job tr...
4        Colleague;Friend or family member;Other online...
5        Books / Physical media;Online Courses or Certi...
                               ...
89180    Online Courses or Certification;Other online r...
89181    Colleague;Online Courses or Certification;Othe...
89182    Books / Physical media;Hackathons (virtual or ...
89183    Online Courses or Certification;Other online r...
89184    Colleague;Online Courses or Certification;Othe...
Name: LearnCode, Length: 89184, dtype: object
```

```
[121] 1 #those values which have 'books' in string
      2 filt=df['LearnCode'].str.contains('Books',na=False)
      3
```

```
[122] 1 df.loc[filt, 'LearnCode']
```

```
ResponseId
2        Books / Physical media;Colleague;Friend or fam...
3        Books / Physical media;Colleague;On the job tr...
5        Books / Physical media;Online Courses or Certi...
6        Books / Physical media;Colleague;Online Course...
8        Books / Physical media;Online Courses or Certi...
                               ...
89162    Books / Physical media;Other online resources ...
89167    Books / Physical media;Hackathons (virtual or ...
89173    Books / Physical media;School (i.e., Universit...
89178    Books / Physical media;Other online resources ...
89182    Books / Physical media;Hackathons (virtual or ...
Name: LearnCode, Length: 45406, dtype: object
```

```
[123] 1 #filter return true false in series when we apply this in
      2 #data frame then we get those value where have true
      3 filt=df['LearnCode'].str.contains('Books',na=False)
```

```
[124] 1 filt
```

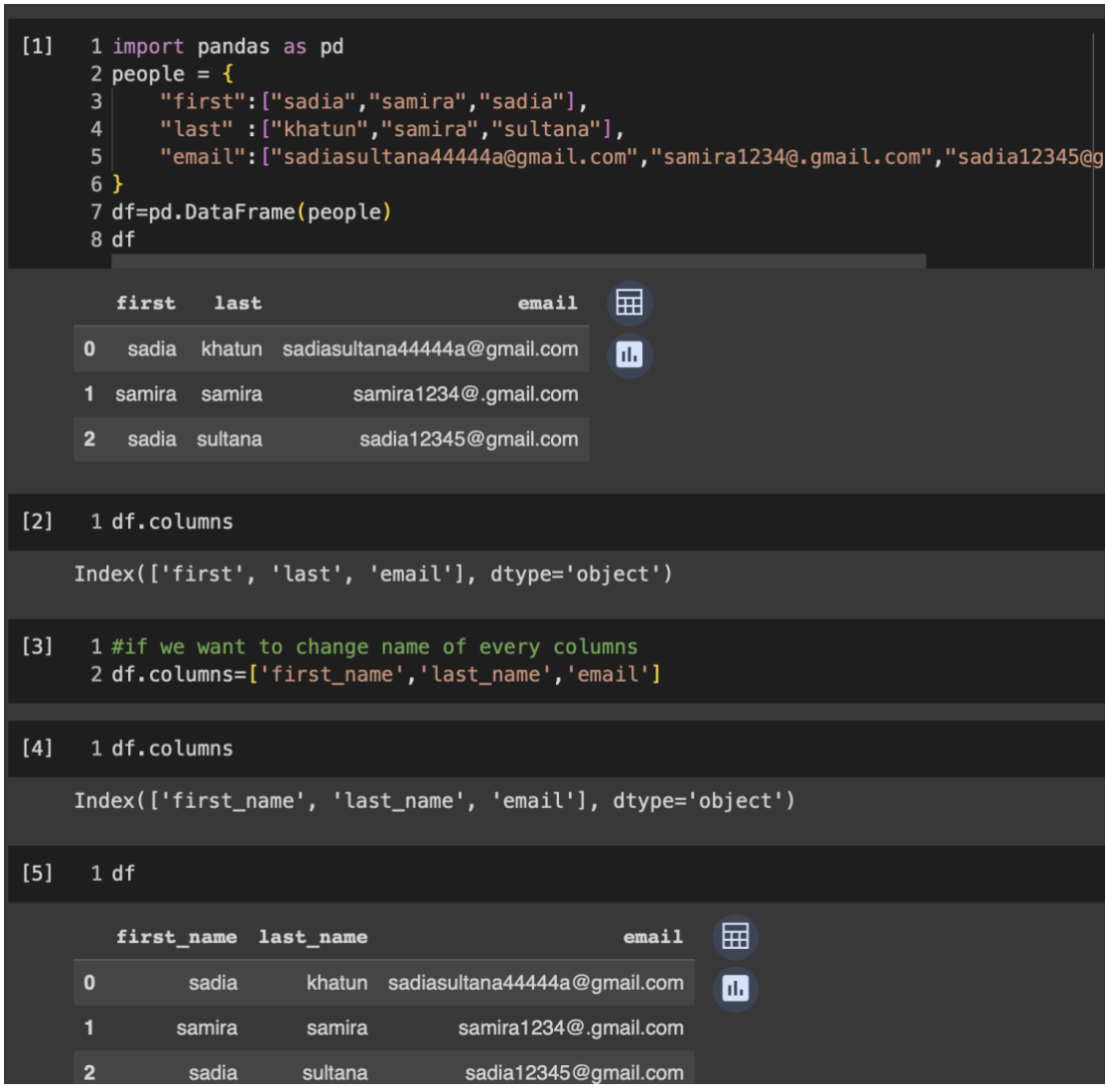
```
ResponseId
1        False
2         True
3         True
4        False
5         True
```

Chapter 5: Updating Rows and Columns – Modifying Data With In Data frames

This section covers

#x.upper() , df.columns.str.replace('_',' '), x.lower(),df.rename(columns={}),df.at(),

.apply(), lambda, .map(), .applymap, pd.Series.min, .replace({' ':' '})



```
[1]  1 import pandas as pd
     2 people = {
     3     "first":["sadia","samira","sadia"],
     4     "last" :["khatun","samira","sultana"],
     5     "email":["sadiasultana44444a@gmail.com","samira1234@.gmail.com","sadia12345@g
     6 }
     7 df=pd.DataFrame(people)
     8 df
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[2]  1 df.columns
```

```
Index(['first', 'last', 'email'], dtype='object')
```

```
[3]  1 #if we want to change name of every columns
     2 df.columns=['first_name','last_name','email']
```

```
[4]  1 df.columns
```

```
Index(['first_name', 'last_name', 'email'], dtype='object')
```

```
[5]  1 df
```

	first_name	last_name	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[7]  1 #if i wanted uppercase all of the columns name here
     2 #i could use a list comprehension
     3 df.columns = [x.upper() for x in df.columns]
     4 df
```

	FIRST_NAME	LAST_NAME	EMAIL
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[8]  1 #if we want to replace all under score of columns name with space
     2
     3 df.columns=df.columns.str.replace('_',' ')
     4 df
```

	FIRST NAME	LAST NAME	EMAIL
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[10] 1 df.columns = [x.lower() for x in df.columns]
     2 df
```

	first name	last name	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[11] 1 df.columns=df.columns.str.replace(' ','_')
     2 df
```

	first_name	last_name	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

[13]
```
#WHAT IF WE WANTED TO CHANGE JUST SOME COLUMNS NAME
# this case we can use rename
#first_name will be map into first
df.rename(columns={'first_name':'first','last_name':'last'},inplace=True)
df
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	samira	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
#let grab 2nd row and change its frist name sirajam
df.loc[1,'first']='sirajam'
df
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	sirajam	samira	samira1234@.gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
#just grab first ,email of 2nd row and change first name and email
#in this we have to pass name and email in a list
df.loc[1,['first','email']]=['sirajam','sirajamsamira1234@gmail.com']
df
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	sirajam	samira	sirajamsamira1234@gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[18]  1 #the place of loc we can also use 'at' perform same operation
      2 #i prefer loc
      3
      4 df.at[1,'first']='sim'
      5 df
      6
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	sim	samira	sirajamsamira1234@gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[19]  1 df.at[1,'first']='sirajam'
      2 df
      3
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	sirajam	samira	sirajamsamira1234@gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[20]  1 #multiple row update
      2 # make all email addrss lower case
      3
      4 df['email'].str.lower()
```

```
0    sadiasultana44444a@gmail.com
1     sirajamsamira1234@gmail.com
2            sadia12345@gmail.com
Name: email, dtype: object
```

```
[23]  1 df['email']=df['email'].str.upper()
      2 df
      3
```

	first	last	email
0	sadia	khatun	SADIASULTANA44444A@GMAIL.COM
1	sirajam	samira	SIRAJAMSAMIRA1234@GMAIL.COM
2	sadia	sultana	SADIA12345@GMAIL.COM

for working with multiple row we have 4 methods

apply

map

applymap

replace

```
[24]  1 #apply
      2 df['email'].apply(len)
```

```
0    28
1    27
2    20
Name: email, dtype: int64
```

```
  1 def update_email(email):
  2     return email.lower()
```

```
[27]  1 df['email'].apply(update_email)
```

```
0    sadiasultana44444a@gmail.com
1     sirajamsamira1234@gmail.com
2           sadia12345@gmail.com
Name: email, dtype: object
```

```
[28]  1 df['email']=df['email'].apply(update_email)
```

```
[29]  1 df
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	sirajam	samira	sirajamsamira1234@gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[30]  1 #if we want to use no name function we can use lambda function
      2 df['email']=df['email'].apply(lambda x:x.lower())
```

```
[31]  1 df
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	sirajam	samira	sirajamsamira1234@gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[32]  1 df['email'].apply(len)
```

```
0    28
1    27
2    20
Name: email, dtype: int64
```

```
[33]  1 #number of row each column
      2 df.apply(len)
```

```
first    3
last     3
email    3
dtype: int64
```

```
[34]  1 #number of row each column
      2 df.apply(len, axis='columns')
```

```
0    3
1    3
2    3
dtype: int64
```

```
[35]  1 #each column minimum value
      2 df.apply(pd.Series.min)
```

```
first                  sadia
last                  khatun
email    sadia12345@gmail.com
dtype: object
```

```
[36]  1 df.apply(lambda x:x.min())
```

```
first                  sadia
last                  khatun
email    sadia12345@gmail.com
dtype: object
```

```
[37]  1 #applymap only work for dataframe
      2 #apply this function on entire dATA FARME
      3 df.applymap(len)
```

	first	last	email
0	5	6	28
1	7	6	27
2	5	7	20

```
[38]  1 df.applymap(str.lower)
```

	first	last	email
0	sadia	khatun	sadiasultana44444a@gmail.com
1	sirajam	samira	sirajamsamira1234@gmail.com
2	sadia	sultana	sadia12345@gmail.com

```
[39]  1 #map work for series
      2 #undefine change will be NaN
      3 df['first'].map({'sadia':'aidas'})
```

```
0    aidas
1      NaN
2    aidas
Name: first, dtype: object
```

```
[41]  1 #undefine change will be same as previous
      2 df['first'].replace({'sadia':'aidas'})
```

```
0      aidas
1    sirajam
2      aidas
```

survey dataset

```
[43]  1 df=pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_public.csv',index_c
```

```
[44]  1 schema_df=pd.read_csv('/content/drive/MyDrive/Dataset/survey_results_schema.csv',
```

```
[45]  1 df.head()
```

```
[53]  1 df.columns
```

```
Index(['Q120', 'MainBranch', 'Age', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'TechList', 'BuyNewTool', 'Country',
       'Currency', 'CompTotal', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysPersonal use',
       'OpSysProfessional use', 'OfficeStackAsyncHaveWorkedWith',
       'OfficeStackAsyncWantToWorkWith', 'OfficeStackSyncHaveWorkedWith',
       'OfficeStackSyncWantToWorkWith', 'AISearchHaveWorkedWith',
       'AISearchWantToWorkWith', 'AIDevHaveWorkedWith', 'AIDevWantToWorkWith',
       'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq', 'SOComm',
       'SOAI', 'AISelect', 'AISent', 'AIAcc', 'AIBen',
       'AIToolInterested in Using', 'AIToolCurrently Using',
       'AIToolNot interested in Using', 'AINextVery different',
       'AINextNeither different nor similar', 'AINextSomewhat similar',
       'AINextVery similar', 'AINextSomewhat different', 'TBranch', 'ICorPM',
       'WorkExp', 'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Knowledge_8',
       'Frequency_1', 'Frequency_2', 'Frequency_3', 'TimeSearching',
       'TimeAnswering', 'ProfessionalTech', 'Industry', 'SurveyLength',
       'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

```
[55]  1 df.rename(columns={'ConvertedCompYearly':'SalaryUSD'},inplace=True)
```

```
[56]  1 df.head()
```

```
1 #make "i agree",value of q120 columns equal to yes
2 #substitute i agree with yes
3 df['Q120'].map({'I agree':'YES'})
```

```
ResponseId
1         YES
2         YES
3         YES
4         YES
5         YES
```