1. Pandas Basics

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
# pd.set_option('display.max_columns', None)  # Display all columns
# pd.set_option('display.max_rows', 10)     # Display all rows
# pd.reset_option('display.max_columns')
# pd.reset_option('display.max_rows')
# pd.reset_option('display.width')

data_2022 = pd.read_csv(".//stack-overflow-developer-survey-2022//survey_results_public.csv")
data_2022.head(3)
```

Out[2]:

	ResponseId	MainBranch	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode	LearnCodeOnline	LearnC
0	1	None of these	NaN	NaN	NaN	NaN	NaN	NaN	
1	2	I am a developer by profession	Employed, full-time	Fully remote	Hobby;Contribute to open-source projects	NaN	NaN	NaN	
2	3	I am not primarily a developer, but I write co...	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	Books / Physical media;Friend or family member...	Technical documentation;Blogs;Programming Game...	

3 rows × 79 columns

In [3]:
```
data_schema = pd.read_csv(".//stack-overflow-developer-survey-2022//survey_results_schema.csv")
data_schema.head(3)
```

Out[3]:

	qid	qname	question	force_resp	type	selector
0	QID16	S0	Hel...	False	DB	TB
1	QID12	MetaInfo	Browser Meta Info	False	Meta	Browser
2	QID1	S1	<span style="font-size:22px; font-family: aria...	False	DB	TB

In [4]:
```
data_2022.columns
```

Out[4]:
```
Index(['ResponseId', 'MainBranch', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'Country', 'Currency',
       'CompTotal', 'CompFreq', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysProfessional use',
       'OpSysPersonal use', 'VersionControlSystem', 'VCInteraction',
       'VCHostingPersonal use', 'VCHostingProfessional use',
       'OfficeStackAsyncHaveWorkedWith', 'OfficeStackAsyncWantToWorkWith',
       'OfficeStackSyncHaveWorkedWith', 'OfficeStackSyncWantToWorkWith',
       'Blockchain', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq',
       'SOComm', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity',
       'Accessibility', 'MentalHealth', 'TBranch', 'ICorPM', 'WorkExp',
       'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Frequency_1',
       'Frequency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering',
       'Onboarding', 'ProfessionalTech', 'TrueFalse_1', 'TrueFalse_2',
       'TrueFalse_3', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

In [5]:
```
data_2022.shape
```

Out[5]:
```
(73268, 79)
```

In [6]:
```
data_2022.describe()
```

Out[6]:

	ResponseId	CompTotal	VCHostingPersonal use	VCHostingProfessional use	WorkExp	ConvertedCompYearly
count	73268.000000	3.842200e+04	0.0	0.0	36769.000000	3.807100e+04
mean	36634.500000	2.342434e+52	NaN	NaN	10.242378	1.707613e+05
std	21150.794099	4.591478e+54	NaN	NaN	8.706850	7.814132e+05
min	1.000000	0.000000e+00	NaN	NaN	0.000000	1.000000e+00
25%	18317.750000	3.000000e+04	NaN	NaN	4.000000	3.583200e+04
50%	36634.500000	7.750000e+04	NaN	NaN	8.000000	6.784500e+04
75%	54951.250000	1.540000e+05	NaN	NaN	15.000000	1.200000e+05
max	73268.000000	9.000000e+56	NaN	NaN	50.000000	5.000000e+07

In [7]:
```
data_2022.loc[:,"ResponseId":"YearsCode"].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73268 entries, 0 to 73267
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   ResponseId           73268 non-null  int64 
 1   MainBranch           73268 non-null  object
 2   Employment           71709 non-null  object
 3   RemoteWork           58958 non-null  object
 4   CodingActivities     58899 non-null  object
 5   EdLevel              71571 non-null  object
 6   LearnCode            71580 non-null  object
 7   LearnCodeOnline      50685 non-null  object
 8   LearnCodeCoursesCert 29389 non-null  object
 9   YearsCode            71331 non-null  object
dtypes: int64(1), object(9)
memory usage: 5.6+ MB
```

In [8]:
```
data_2022.columns
```

Out[8]:
```
Index(['ResponseId', 'MainBranch', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'Country', 'Currency',
       'CompTotal', 'CompFreq', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysProfessional use',
       'OpSysPersonal use', 'VersionControlSystem', 'VCInteraction',
       'VCHostingPersonal use', 'VCHostingProfessional use',
       'OfficeStackAsyncHaveWorkedWith', 'OfficeStackAsyncWantToWorkWith',
       'OfficeStackSyncHaveWorkedWith', 'OfficeStackSyncWantToWorkWith',
       'Blockchain', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq',
       'SOComm', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity',
       'Accessibility', 'MentalHealth', 'TBranch', 'ICorPM', 'WorkExp',
       'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Frequency_1',
       'Frequency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering',
       'Onboarding', 'ProfessionalTech', 'TrueFalse_1', 'TrueFalse_2',
       'TrueFalse_3', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

2. Selecting Rows and Columns

In [9]:
```
data_2022.iloc[0]
```

Out[9]:
```
ResponseId                        1
MainBranch           None of these
Employment                     NaN
RemoteWork                     NaN
CodingActivities               NaN
                            ...
TrueFalse_2                    NaN
TrueFalse_3                    NaN
SurveyLength                   NaN
SurveyEase                     NaN
ConvertedCompYearly            NaN
Name: 0, Length: 79, dtype: object
```

In [10]:
```
data_2022.iloc[:3,:4].head(5)
```

Out[10]:

	ResponseId	MainBranch	Employment	RemoteWork
0	1	None of these	NaN	NaN
1	2	I am a developer by profession	Employed, full-time	Fully remote
2	3	I am not primarily a developer, but I write co...	Employed, full-time	Hybrid (some remote, some in-person)

In [11]:
```
data_2022.iloc[1:10:2, 1:4].head(5)
```

Out[11]:

	MainBranch	Employment	RemoteWork
1	I am a developer by profession	Employed, full-time	Fully remote
3	I am a developer by profession	Employed, full-time	Fully remote
5	I am not primarily a developer, but I write co...	Student, full-time	NaN
7	I am a developer by profession	Not employed, but looking for work	NaN
9	I am a developer by profession	Independent contractor, freelancer, or self-em...	Fully remote

In [12]:
```
data_2022.iloc[[1,-1],[1,-1]].head(5)
```

Out[12]:

	MainBranch	ConvertedCompYearly
1	I am a developer by profession	NaN
73267	I used to be a developer by profession, but no...	NaN

In [13]:
```
data_2022.loc[0]
```

Out[13]:
```
ResponseId                        1
MainBranch           None of these
Employment                     NaN
RemoteWork                     NaN
CodingActivities               NaN
                              ...
TrueFalse_2                    NaN
TrueFalse_3                    NaN
SurveyLength                   NaN
SurveyEase                     NaN
ConvertedCompYearly            NaN
Name: 0, Length: 79, dtype: object
```

In [14]:
```
data_2022.loc[[0,1], ["MainBranch", "RemoteWork"]]
```

Out[14]:

	MainBranch	RemoteWork
0	None of these	NaN
1	I am a developer by profession	Fully remote

In [15]:
```
data_2022.loc[0:5, "LanguageHaveWorkedWith":"NEWCollabToolsWantToWorkWith"]
```

Out[15]:

	LanguageHaveWorkedWith	LanguageWantToWorkWith	DatabaseHaveWorkedWith	DatabaseWantToW
0	NaN	NaN	NaN	
1	JavaScript;TypeScript	Rust;TypeScript	NaN	
2	C#;C++;HTML/CSS;JavaScript;Python	C#;C++;HTML/CSS;JavaScript;TypeScript	Microsoft SQL Server	Microsoft SC
3	C#;JavaScript;SQL;TypeScript	C#;SQL;TypeScript	Microsoft SQL Server	Microsoft SC
4	C#;HTML/CSS;JavaScript;SQL;Swift;TypeScript	C#;Elixir;F#;Go;JavaScript;Rust;TypeScript	Cloud Firestore;Elasticsearch;Microsoft SQL Se...	Firestore;Elasticsearch
5	C++;Lua	Lua	NaN	

3. How to Set, Reset, and Use Indexes

In [16]:
```
people = {
    "first": ["Corey", 'Jane', 'John'],
    "last": ["Schafer", 'Doe', 'Doe'],
    "email": ["CoreyMSchafer@gmail.com", 'JaneDoe@email.com', 'JohnDoe@email.com']
}

df =pd.DataFrame(people)
print(df)
```

```
   first     last                   email
0  Corey  Schafer  CoreyMSchafer@gmail.com
1   Jane      Doe        JaneDoe@email.com
2   John      Doe        JohnDoe@email.com
```

In [17]:
```
df.set_index("email", inplace=True)
```

In [18]:
```
df.sort_index(ascending=False)
```

Out[18]:

email	first	last
JohnDoe@email.com	John	Doe
JaneDoe@email.com	Jane	Doe
CoreyMSchafer@gmail.com	Corey	Schafer

In [19]:
```
df.sort_index(inplace=True)
df
```

Out[19]:

email	first	last
CoreyMSchafer@gmail.com	Corey	Schafer
JaneDoe@email.com	Jane	Doe
JohnDoe@email.com	John	Doe

4. Filtering - Using Conditionals to Filter Rows and Columns

In [20]:
```
data_2022_002 = pd.read_csv(".//stack-overflow-developer-survey-2022//survey_results_public.csv")
data_2022_002.columns
```

Out[20]:
```
Index(['ResponseId', 'MainBranch', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'Country', 'Currency',
       'CompTotal', 'CompFreq', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysProfessional use',
       'OpSysPersonal use', 'VersionControlSystem', 'VCInteraction',
       'VCHostingPersonal use', 'VCHostingProfessional use',
       'OfficeStackAsyncHaveWorkedWith', 'OfficeStackAsyncWantToWorkWith',
       'OfficeStackSyncHaveWorkedWith', 'OfficeStackSyncWantToWorkWith',
       'Blockchain', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq',
       'SOComm', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity',
       'Accessibility', 'MentalHealth', 'TBranch', 'ICorPM', 'WorkExp',
       'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Frequency_1',
       'Frequency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering',
       'Onboarding', 'ProfessionalTech', 'TrueFalse_1', 'TrueFalse_2',
       'TrueFalse_3', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

In [21]:
```
# data_2022_002["RemoteWork"]
remote_work_filter = (data_2022_002["RemoteWork"] == "Fully remote") | (data_2022_002["RemoteWork"] == "Full
remote_work_data = data_2022_002[remote_work_filter]
remote_work_data.loc[:10,:"RemoteWork"]
```

Out[21]:

	ResponseId	MainBranch	Employment	RemoteWork
1	2	I am a developer by profession	Employed, full-time	Fully remote
3	4	I am a developer by profession	Employed, full-time	Fully remote
9	10	I am a developer by profession	Independent contractor, freelancer, or self-em...	Fully remote

In [22]:
```
lang_worked_with_filter = data_2022_002["LanguageHaveWorkedWith"].str.contains("Python", na=False)
lang_worked_with_data = data_2022_002[lang_worked_with_filter]
lang_worked_with_data[["LanguageHaveWorkedWith", "CompTotal"]]
```

Out[22]:

	LanguageHaveWorkedWith	CompTotal
2	C#;C++;HTML/CSS;JavaScript;Python	32000.0
6	C++;HTML/CSS;JavaScript;PHP;Python;TypeScript	NaN
11	C#;HTML/CSS;JavaScript;PowerShell;Python;Rust;SQL	194400.0
14	HTML/CSS;JavaScript;PHP;Python;R;Ruby;Scala	110000.0
16	C#;Java;PHP;Python;R	37000.0
...	...	...
73257	C;Python;SQL	NaN
73261	Bash/Shell;HTML/CSS;Java;JavaScript;Python;SQL...	36000.0
73263	Bash/Shell;Dart;JavaScript;PHP;Python;SQL;Type...	60000.0
73264	Bash/Shell;HTML/CSS;JavaScript;Python;SQL	107000.0
73265	HTML/CSS;JavaScript;PHP;Python;SQL	NaN

34155 rows × 2 columns

In [23]:
```
country_filter = data_2022_002["Country"].isin(["Australia", "India"])
country_data = data_2022_002[country_filter]
country_data["Country"].value_counts()
```

Out[23]:
```
Country
India        6639
Australia    1462
Name: count, dtype: int64
```

In [24]:
```
country_data = data_2022_002.loc[~country_filter, "Country"]
country_data.value_counts()
```

Out[24]:
```
Country
United States of America                              13543
Germany                                                5395
United Kingdom of Great Britain and Northern Ireland   4190
Canada                                                 2490
France                                                 2328
                                                      ...
Monaco                                                    1
Djibouti                                                  1
Seychelles                                                1
Solomon Islands                                           1
Saint Kitts and Nevis                                     1
Name: count, Length: 178, dtype: int64
```

In [25]:
```
pay_filter = (data_2022_002["CompTotal"] > 20000) & (data_2022_002["CompTotal"] < 30000)
pay_data = data_2022_002.loc[pay_filter, ["Country", "CompTotal"]]
pay_data.head()
```

Out[25]:

	Country	CompTotal
165	United Kingdom of Great Britain and Northern I...	22000.0
191	Greece	28000.0
212	United Kingdom of Great Britain and Northern I...	22000.0
245	Spain	24500.0
261	United Kingdom of Great Britain and Northern I...	22000.0

5. Updating Rows and Columns - Modifying Data Within DataFrames

In [26]:
```
data_2022_003 = pd.read_csv(".//stack-overflow-developer-survey-2022//survey_results_public.csv")
# help(data_2022_003.columns.str)
print(data_2022_003.columns.str.replace(" ", "_"))
print(data_2022_003.columns.str.lower())
data_2022_003.rename(columns={"CompTotal": "Total_Pay"}, inplace=True)
data_2022_003.columns
data_2022_003.columns = ["_".join(i.split()) for i in data_2022_003.columns]
data_2022_003.columns
```

```
Index(['ResponseId', 'MainBranch', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'Country', 'Currency',
       'CompTotal', 'CompFreq', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysProfessional_use',
       'OpSysPersonal_use', 'VersionControlSystem', 'VCInteraction',
       'VCHostingPersonal_use', 'VCHostingProfessional_use',
       'OfficeStackAsyncHaveWorkedWith', 'OfficeStackAsyncWantToWorkWith',
       'OfficeStackSyncHaveWorkedWith', 'OfficeStackSyncWantToWorkWith',
       'Blockchain', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq',
       'SOComm', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity',
       'Accessibility', 'MentalHealth', 'TBranch', 'ICorPM', 'WorkExp',
       'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Frequency_1',
       'Frequency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering',
       'Onboarding', 'ProfessionalTech', 'TrueFalse_1', 'TrueFalse_2',
       'TrueFalse_3', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
Index(['responseid', 'mainbranch', 'employment', 'remotework',
       'codingactivities', 'edlevel', 'learncode', 'learncodeonline',
       'learncodecoursescert', 'yearscode', 'yearscodepro', 'devtype',
       'orgsize', 'purchaseinfluence', 'buynewtool', 'country', 'currency',
       'comptotal', 'compfreq', 'languagehaveworkedwith',
       'languagewanttoworkwith', 'databasehaveworkedwith',
       'databasewanttoworkwith', 'platformhaveworkedwith',
       'platformwanttoworkwith', 'webframehaveworkedwith',
       'webframewanttoworkwith', 'misctechhaveworkedwith',
       'misctechwanttoworkwith', 'toolstechhaveworkedwith',
       'toolstechwanttoworkwith', 'newcollabtoolshaveworkedwith',
       'newcollabtoolswanttoworkwith', 'opsysprofessional use',
       'opsyspersonal use', 'versioncontrolsystem', 'vcinteraction',
       'vchostingpersonal use', 'vchostingprofessional use',
       'officestackasynchaveworkedwith', 'officestackasyncwanttoworkwith',
       'officestacksynchaveworkedwith', 'officestacksyncwanttoworkwith',
       'blockchain', 'newsosites', 'sovisitfreq', 'soaccount', 'sopartfreq',
       'socomm', 'age', 'gender', 'trans', 'sexuality', 'ethnicity',
       'accessibility', 'mentalhealth', 'tbranch', 'icorpm', 'workexp',
       'knowledge_1', 'knowledge_2', 'knowledge_3', 'knowledge_4',
       'knowledge_5', 'knowledge_6', 'knowledge_7', 'frequency_1',
       'frequency_2', 'frequency_3', 'timesearching', 'timeanswering',
       'onboarding', 'professionaltech', 'truefalse_1', 'truefalse_2',
       'truefalse_3', 'surveylength', 'surveyease', 'convertedcompyearly'],
      dtype='object')
```

Out[26]:
```
Index(['ResponseId', 'MainBranch', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'Country', 'Currency',
       'Total_Pay', 'CompFreq', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysProfessional_use',
       'OpSysPersonal_use', 'VersionControlSystem', 'VCInteraction',
       'VCHostingPersonal_use', 'VCHostingProfessional_use',
       'OfficeStackAsyncHaveWorkedWith', 'OfficeStackAsyncWantToWorkWith',
       'OfficeStackSyncHaveWorkedWith', 'OfficeStackSyncWantToWorkWith',
       'Blockchain', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq',
       'SOComm', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity',
       'Accessibility', 'MentalHealth', 'TBranch', 'ICorPM', 'WorkExp',
       'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Frequency_1',
       'Frequency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering',
       'Onboarding', 'ProfessionalTech', 'TrueFalse_1', 'TrueFalse_2',
       'TrueFalse_3', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

In [27]:
```
data_2022_004 = data_2022_003.loc[:,['Total_Pay', 'CompFreq', 'LanguageHaveWorkedWith']]
data_2022_004.loc[2] = [32000, "Per Year", "C#;C+++;HTML/CSS;JavaScript;Python"]
print(data_2022_004.head(4))
data_2022_004.loc[2,"Total_Pay"] = "32000"
print("======================================================================")
print(data_2022_004.head(4))
time_filter = (data_2022_004["CompFreq"] == "Monthly")
data_2022_004.loc[time_filter, "CompFreq"] = "Per Month"
print("======================================================================")
data_2022_004
```

```
   Total_Pay  CompFreq              LanguageHaveWorkedWith
0        NaN       NaN                                 NaN
1        NaN       NaN               JavaScript;TypeScript
2    32000.0  Per Year  C#;C+++;HTML/CSS;JavaScript;Python
3    60000.0   Monthly        C#;JavaScript;SQL;TypeScript
======================================================================
  Total_Pay  CompFreq              LanguageHaveWorkedWith
0       NaN       NaN                                 NaN
1       NaN       NaN               JavaScript;TypeScript
2     32000  Per Year  C#;C+++;HTML/CSS;JavaScript;Python
3   60000.0   Monthly        C#;JavaScript;SQL;TypeScript
======================================================================
```

Out[27]:

	Total_Pay	CompFreq	LanguageHaveWorkedWith
0	NaN	NaN	NaN
1	NaN	NaN	JavaScript;TypeScript
2	32000	Per Year	C#;C+++;HTML/CSS;JavaScript;Python
3	60000.0	Per Month	C#;JavaScript;SQL;TypeScript
4	NaN	NaN	C#;HTML/CSS;JavaScript;SQL;Swift;TypeScript
...	...	...	...
73263	60000.0	Yearly	Bash/Shell;Dart;JavaScript;PHP;Python;SQL;Type...
73264	107000.0	Yearly	Bash/Shell;HTML/CSS;JavaScript;Python;SQL
73265	NaN	NaN	HTML/CSS;JavaScript;PHP;Python;SQL
73266	58500.0	Yearly	C#;Delphi;VBA
73267	NaN	NaN	C#;JavaScript;Lua;PowerShell;SQL;TypeScript

73268 rows × 3 columns

In [28]:
```
data_2022_004["CompFreq"] = data_2022_004["CompFreq"].str.upper()
data_2022_004["CompFreq"].head()
```

Out[28]:
```
0          NaN
1          NaN
2     PER YEAR
3    PER MONTH
4          NaN
Name: CompFreq, dtype: object
```

In [29]:
```
people = {
    "first": ["Corey", 'Jane', 'John'],
    "last": ["Schafer", 'Doe', 'Doe'],
    "email": ["CoreyMSchafer@gmail.com", 'JaneDoe@email.com', 'JohnDoe@email.com']
}

df =pd.DataFrame(people)
print(df["email"].apply(len))
df["email"].apply(lambda x:x.upper())
```

```
0    23
1    17
2    17
Name: email, dtype: int64
```

Out[29]:
```
0    COREYMSCHAFER@GMAIL.COM
1          JANEDOE@EMAIL.COM
2          JOHNDOE@EMAIL.COM
Name: email, dtype: object
```

In [30]:
```
print(df.apply(len))
print("========================================")
print(df)
```

```
first    3
last     3
email    3
dtype: int64
========================================
   first     last                    email
0  Corey   Schafer  CoreyMSchafer@gmail.com
1   Jane       Doe        JaneDoe@email.com
2   John       Doe        JohnDoe@email.com
```

In [31]:
```
print(df.apply(pd.Series.min))
# help(pd.Series)
```

```
first                     Corey
last                        Doe
email    CoreyMSchafer@gmail.com
dtype: object
```

In [32]:
```
df.apply(lambda x : x.min())
```

Out[32]:
```
first                     Corey
last                        Doe
email    CoreyMSchafer@gmail.com
dtype: object
```

In [33]:
```
df.applymap(len) # applymap===> It only works for DataFrame, However, apply works with Series & dataframe both
```

Out[33]:

	first	last	email
0	5	7	23
1	4	3	17
2	4	3	17

In [34]:
```
df.applymap(lambda x: x.upper())
```

Out[34]:

	first	last	email
0	COREY	SCHAFER	COREYMSCHAFER@GMAIL.COM
1	JANE	DOE	JANEDOE@EMAIL.COM
2	JOHN	DOE	JOHNDOE@EMAIL.COM

In [35]:
```
df.applymap(str.lower)
```

Out[35]:

	first	last	email
0	corey	schafer	coreymschafer@gmail.com
1	jane	doe	janedoe@email.com
2	john	doe	johndoe@email.com

In [36]:
```
df["first"].map({"Corey": "gunjan"})
```

Out[36]:
```
0    gunjan
1       NaN
2       NaN
Name: first, dtype: object
```

In [37]:
```
df["first"].replace({"Corey": "gunjan"})
```

Out[37]:
```
0    gunjan
1      Jane
2      John
Name: first, dtype: object
```

6. Add/Remove Rows and Columns From DataFrames

In [38]:
```
print(len(data_2022_003.columns))
data_2022_003.drop(columns=["Onboarding"], inplace=True)
print(len(data_2022_003.columns))
data_2022_003.columns
```

```
79
78
```

Out[38]:
```
Index(['ResponseId', 'MainBranch', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'Country', 'Currency',
       'Total_Pay', 'CompFreq', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysProfessional_use',
       'OpSysPersonal_use', 'VersionControlSystem', 'VCInteraction',
       'VCHostingPersonal_use', 'VCHostingProfessional_use',
       'OfficeStackAsyncHaveWorkedWith', 'OfficeStackAsyncWantToWorkWith',
       'OfficeStackSyncHaveWorkedWith', 'OfficeStackSyncWantToWorkWith',
       'Blockchain', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq',
       'SOComm', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity',
       'Accessibility', 'MentalHealth', 'TBranch', 'ICorPM', 'WorkExp',
       'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Frequency_1',
       'Frequency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering',
       'ProfessionalTech', 'TrueFalse_1', 'TrueFalse_2', 'TrueFalse_3',
       'SurveyLength', 'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

In [39]:
```
people = {
    "first": ["Corey", 'Jane', 'John'],
    "last": ["Schafer", 'Doe', 'Doe'],
    "email": ["CoreyMSchafer@gmail.com", 'JaneDoe@email.com', 'JohnDoe@email.com']
}

df =pd.DataFrame(people)

df["full_name"] = df ["first"] + " " + df["last"]
df["host"], df["domain"] = df["email"].apply(lambda x:x.split("@")[0]), df["email"].apply(lambda x :x.split("
# df["host"], df["domain"] = df["email"].str.split(" ", expand=True)
df
```

Out[39]:

	first	last	email	full_name	host	domain
0	Corey	Schafer	CoreyMSchafer@gmail.com	Corey Schafer	CoreyMSchafer	gmail.com
1	Jane	Doe	JaneDoe@email.com	Jane Doe	JaneDoe	email.com
2	John	Doe	JohnDoe@email.com	John Doe	JohnDoe	email.com

In [40]:
```
df.drop(columns=["host", "domain"], inplace=True)
```

In [41]:
```
df.rename(columns={"full_name": "name"})
```

Out[41]:

	first	last	email	name
0	Corey	Schafer	CoreyMSchafer@gmail.com	Corey Schafer
1	Jane	Doe	JaneDoe@email.com	Jane Doe
2	John	Doe	JohnDoe@email.com	John Doe

In [42]:
```
# df.append({"first": "Gunjan"}, ignore_index=True) #==> It do not work
people2 = {
    "first": ["Gunjan", 'Chandan', 'Kundan'],
    "last": ["Kumar", 'Singh', 'Singh']
}
df2 =pd.DataFrame(people2)
print(df2)
# print(df2.drop(index=0))
print("=================")
print(df2.drop(index=df2[df2["last"] == "Singh" ].index))
```

```
     first   last
0   Gunjan  Kumar
1  Chandan  Singh
2   Kundan  Singh
=================
    first   last
0  Gunjan  Kumar
```

In [43]:
```
help(df)
```

```
Help on DataFrame in module pandas.core.frame object:

class DataFrame(pandas.core.generic.NDFrame, pandas.core.arraylike.OpsMixin)
 |  DataFrame(data=None, index: 'Axes | None' = None, columns: 'Axes | None' = None, dtype: 'Dtype | Non
e' = None, copy: 'bool | None' = None) -> 'None'
 |
 |  Two-dimensional, size-mutable, potentially heterogeneous tabular data.
 |
 |  Data structure also contains labeled axes (rows and columns).
 |  Arithmetic operations align on both row and column labels. Can be
 |  thought of as a dict-like container for Series objects. The primary
 |  pandas data structure.
 |
 |  Parameters
 |  ----------
 |  data : ndarray (structured or homogeneous), Iterable, dict, or DataFrame
 |      Dict can contain Series, arrays, constants, dataclass or list-like objects. If
 |      data is a dict, column order follows insertion-order. If a dict contains Series
 |      which have an index defined, it is aligned by its index. This alignment also
```

7. Sorting by value & index

In [44]:
```
df2, df
```

Out[44]:
```
(     first   last
 0   Gunjan  Kumar
 1  Chandan  Singh
 2   Kundan  Singh,
    first     last                    email      full_name
 0  Corey  Schafer  CoreyMSchafer@gmail.com  Corey Schafer
 1   Jane      Doe        JaneDoe@email.com       Jane Doe
 2   John      Doe        JohnDoe@email.com       John Doe)
```

In [45]:
```
df2.sort_values(by="first", ascending=False)
```

Out[45]:

	first	last
2	Kundan	Singh
0	Gunjan	Kumar
1	Chandan	Singh

In [46]:
```
people = {
    "first": ["Gunjan", 'Chandan', 'Kundan', "Corey", 'Jane', 'John'],
    "last": ["Kumar", 'Singh', 'Singh', "Schafer", 'Doe', 'Doe'],
    "email": ["GunjanKumar@gmail.com", "ChandanSingh@gmail.com", "KundanSingh@gmail.com","CoreyMSchafer@gmail
}
df3 = pd.DataFrame(people)
print(df3)
```

```
     first     last                    email
0   Gunjan    Kumar    GunjanKumar@gmail.com
1  Chandan    Singh   ChandanSingh@gmail.com
2   Kundan    Singh    KundanSingh@gmail.com
3    Corey  Schafer  CoreyMSchafer@gmail.com
4     Jane      Doe        JaneDoe@email.com
5     John      Doe        JohnDoe@email.com
```

In [47]:
```
df3.sort_values(by=["last", "first"])
```

Out[47]:

	first	last	email
4	Jane	Doe	JaneDoe@email.com
5	John	Doe	JohnDoe@email.com
0	Gunjan	Kumar	GunjanKumar@gmail.com
3	Corey	Schafer	CoreyMSchafer@gmail.com
1	Chandan	Singh	ChandanSingh@gmail.com
2	Kundan	Singh	KundanSingh@gmail.com

In [48]:
```
df3.sort_values(by=["last", "first"], ascending=[False, True])
```

Out[48]:

	first	last	email
1	Chandan	Singh	ChandanSingh@gmail.com
2	Kundan	Singh	KundanSingh@gmail.com
3	Corey	Schafer	CoreyMSchafer@gmail.com
0	Gunjan	Kumar	GunjanKumar@gmail.com
4	Jane	Doe	JaneDoe@email.com
5	John	Doe	JohnDoe@email.com

In [49]:
```
df3.sort_index()
```

Out[49]:

	first	last	email
0	Gunjan	Kumar	GunjanKumar@gmail.com
1	Chandan	Singh	ChandanSingh@gmail.com
2	Kundan	Singh	KundanSingh@gmail.com
3	Corey	Schafer	CoreyMSchafer@gmail.com
4	Jane	Doe	JaneDoe@email.com
5	John	Doe	JohnDoe@email.com

In [50]:
```
df3["email"].sort_values()
```

Out[50]:
```
1     ChandanSingh@gmail.com
3    CoreyMSchafer@gmail.com
0      GunjanKumar@gmail.com
4          JaneDoe@email.com
5          JohnDoe@email.com
2     KundanSingh@gmail.com
Name: email, dtype: object
```

In [51]:
```
data_2022.sort_values(by=["Country", "CompTotal"], ascending=[True, False], inplace=True)
data_2022.loc[:, "Country":"CompTotal"].head()
```

Out[51]:

	Country	Currency	CompTotal
6891	Afghanistan	IRR\tIranian rial	55000000.0
38512	Afghanistan	AFN\tAfghan afghani	1200000.0
43639	Afghanistan	AFN\tAfghan afghani	700000.0
43690	Afghanistan	AFN\tAfghan afghani	100000.0
2448	Afghanistan	AED United Arab Emirates dirham	40000.0

In [52]:
```
data_2022["CompTotal"].nlargest(5), data_2022["CompTotal"].nsmallest()
```

Out[52]:
```
(35786    9.000000e+56
 3068     1.000000e+52
 70597    1.000000e+22
 17567    1.000000e+15
 19244    5.000000e+12
 Name: CompTotal, dtype: float64,
 2737     0.0
 67666    0.0
 22777    0.0
 27810    0.0
 57028    0.0
 Name: CompTotal, dtype: float64)
```

In [53]:
```
pd.set_option("display.max_columns", 100)
data_2022.nlargest(10, "CompTotal")
```

Out[53]:

	ResponseId	MainBranch	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode	
35786	35787	I am a developer by profession	Employed, full-time;Employed, part-time	Hybrid (some remote, some in-person)	I don't code outside of work	Secondary school (e.g. American high school, G...	Colleague	
3068	3069	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	Hobby;Contribute to open-source projects;Boots...	Bachelor's degree (B.A., B.S., B.Eng., etc.)	School (i.e., University, College, etc);Other ...	
70597	70598	I am a developer by profession	Independent contractor, freelancer, or self-em...	Fully remote	Hobby;Contribute to open-source projects	Bachelor's degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;Other online resources ...	documenta
17567	17568	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	Hobby;Contribute to open-source projects;Freel...	Secondary school (e.g. American high school, G...	Other online resources (e.g., videos, blogs, f...	Technical do
19244	19245	I am not primarily a developer, but I write co...	Independent contractor, freelancer, or self-em...	Fully remote	Hobby;Contribute to open-source projects;Freel...	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	Books / Physical media;Friend or family member...	documenta
24640	24641	I am a developer by profession	Employed, full-time	Full in-person	Hobby;Contribute to open-source projects;Boots...	Secondary school (e.g. American high school, G...	Books / Physical media	
31190	31191	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	I don't code outside of work	Other doctoral degree (Ph.D., Ed.D., etc.)	Books / Physical media;Friend or family member...	documentation;B
71603	71604	I am a developer by profession	Employed, part-time	Fully remote	Hobby	Something else	Other (please specify):	
34411	34412	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	Bootstrapping a business;Freelance/contract work	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	Books / Physical media;Other online resources ...	documer
49874	49875	I am a developer by profession	Employed, full-time	Fully remote	Contribute to open-source projects	Primary/elementary school	Books / Physical media;Friend or family member...	documentation;B

8. Grouping and Aggregating - Analyzing and Exploring data

In [54]:
```
data_2022.columns
```

Out[54]:
```
Index(['ResponseId', 'MainBranch', 'Employment', 'RemoteWork',
       'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline',
       'LearnCodeCoursesCert', 'YearsCode', 'YearsCodePro', 'DevType',
       'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'Country', 'Currency',
       'CompTotal', 'CompFreq', 'LanguageHaveWorkedWith',
       'LanguageWantToWorkWith', 'DatabaseHaveWorkedWith',
       'DatabaseWantToWorkWith', 'PlatformHaveWorkedWith',
       'PlatformWantToWorkWith', 'WebframeHaveWorkedWith',
       'WebframeWantToWorkWith', 'MiscTechHaveWorkedWith',
       'MiscTechWantToWorkWith', 'ToolsTechHaveWorkedWith',
       'ToolsTechWantToWorkWith', 'NEWCollabToolsHaveWorkedWith',
       'NEWCollabToolsWantToWorkWith', 'OpSysProfessional use',
       'OpSysPersonal use', 'VersionControlSystem', 'VCInteraction',
       'VCHostingPersonal use', 'VCHostingProfessional use',
       'OfficeStackAsyncHaveWorkedWith', 'OfficeStackAsyncWantToWorkWith',
       'OfficeStackSyncHaveWorkedWith', 'OfficeStackSyncWantToWorkWith',
       'Blockchain', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq',
       'SOComm', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity',
       'Accessibility', 'MentalHealth', 'TBranch', 'ICorPM', 'WorkExp',
       'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge_4',
       'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Frequency_1',
       'Frequency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering',
       'Onboarding', 'ProfessionalTech', 'TrueFalse_1', 'TrueFalse_2',
       'TrueFalse_3', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly'],
      dtype='object')
```

In [55]:
```
country_grp = data_2022.groupby(["Country"])
country_grp.get_group("India").head(3)
```

Out[55]:

Employment	RemoteWork	CodingActivities	EdLevel	LearnCode	LearnCodeOnline	LearnCodeCoursesCert	YearsCode	YearsC
Employed, full-time	Fully remote	Hobby	Bachelor's degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;Other online resources ...	Technical documentation;Blogs;Written Tutorial...	Udemy	26	
Employed, full-time	Hybrid (some remote, some in-person)	School or academic work	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	Books / Physical media;Other online resources ...	Technical documentation;Blogs;Written Tutorial...	Udemy;edX	20	
Employed, full-time	Hybrid (some remote, some in-person)	Hobby;Contribute to open-source projects;Boots...	Bachelor's degree (B.A., B.S., B.Eng., etc.)	School (i.e., University, College, etc)	NaN	NaN	25	

In [56]:
```
country_filter = (data_2022["Country"] == "India")
data_2022.loc[country_filter]["RemoteWork"].value_counts()
```

Out[56]:
```
RemoteWork
Fully remote                          1952
Hybrid (some remote, some in-person)  1942
Full in-person                        1005
Name: count, dtype: int64
```

In [57]:
```
country_filter = (data_2022["Country"].isin(["India", "Canada"]))
data_2022.loc[country_filter]["RemoteWork"].value_counts()
```

Out[57]:
```
RemoteWork
Fully remote                          3260
Hybrid (some remote, some in-person)  2594
Full in-person                        1141
Name: count, dtype: int64
```

In [58]:
```
country_grp["RemoteWork"].value_counts()
```

Out[58]:
```
Country      RemoteWork
Afghanistan  Hybrid (some remote, some in-person)    17
             Fully remote                           15
             Full in-person                         14
Albania      Full in-person                         19
             Hybrid (some remote, some in-person)    13
                                                    ..
Zambia       Hybrid (some remote, some in-person)     4
             Fully remote                           3
Zimbabwe     Hybrid (some remote, some in-person)     8
             Fully remote                           5
             Full in-person                         5
Name: count, Length: 474, dtype: int64
```

In [59]:
```
country_grp["CompTotal"].agg(["mean", "median"]).head(5)
```

Out[59]:

Country	mean	median
Afghanistan	3.808414e+06	6000.0
Albania	5.104657e+05	5500.0
Algeria	8.754444e+04	75000.0
Andorra	7.677250e+04	31500.0
Angola	1.287990e+09	600000.0

In [60]:
```
country_grp["RemoteWork"].value_counts().loc["India"]
```

Out[60]:
```
RemoteWork
Fully remote                           1952
Hybrid (some remote, some in-person)    1942
Full in-person                         1005
Name: count, dtype: int64
```

In [61]:
```
country_grp["RemoteWork"].value_counts().loc["China": "Cuba"]
```

Out[61]:
```
Country                   RemoteWork
China                     Full in-person                         195
                          Hybrid (some remote, some in-person)    127
                          Fully remote                            66
Colombia                  Fully remote                           228
                          Hybrid (some remote, some in-person)     51
                          Full in-person                          19
Congo, Republic of the...  Hybrid (some remote, some in-person)      5
                          Fully remote                             1
Costa Rica                Fully remote                            49
                          Hybrid (some remote, some in-person)     15
                          Full in-person                           4
Croatia                   Hybrid (some remote, some in-person)     72
                          Fully remote                            60
                          Full in-person                          27
Cuba                      Fully remote                            19
                          Full in-person                           5
                          Hybrid (some remote, some in-person)      4
Name: count, dtype: int64
```

In [62]:
```
country_grp["CompTotal"].median().loc[["India","China"]]
```

Out[62]:
```
Country
India    885000.0
China     32500.0
Name: CompTotal, dtype: float64
```

In [63]:
```
# c_filter = data_2022["Country"].str.startswith("C", na=False) ==> This do not work
# help(data_2022.iloc[:,1].str)
country_filter = data_2022["Country"] == "India"
print(data_2022.loc[country_filter, "LanguageHaveWorkedWith"].str.contains("Python").sum())
print(data_2022.loc[country_filter]["LanguageHaveWorkedWith"].str.contains("Python").sum())
```

```
3081
3081
```

In [64]:
```
# country_grp["LanguageHaveWorkedWith"].str.contains("Python").sum()
```

In [65]:
```
country_grp["LanguageHaveWorkedWith"].apply(lambda x: x.str.contains("Python").sum())
```

Out[65]:
```
Country
Afghanistan                              25
Albania                                  19
Algeria                                  19
Andorra                                   4
Angola                                    3
                                        ...
Venezuela, Bolivarian Republic of...     52
Viet Nam                                136
Yemen                                     4
Zambia                                    6
Zimbabwe                                 11
Name: LanguageHaveWorkedWith, Length: 180, dtype: int64
```

In [66]:
```
country_grp["LanguageHaveWorkedWith"].apply(lambda x: x.str.contains("Python").sum()).loc["India"]
```

Out[66]:
```
3081
```

9. Cleaning Data, Casting DataTypes and Handling Missing Values

In [67]:
```
people_data = {
    "first": ["Gunjan", 'Chandan', 'Kundan', "Corey", 'Jane', 'John', np.nan, None, 'NA'],
    "last": ["Kumar", 'Singh', 'Singh', "Schafer", 'Doe', 'Doe', np.nan, np.nan, 'Missing'],
    "email": ["GunjanKumar@gmail.com", "ChandanSingh@gmail.com", "KundanSingh@gmail.com",
              "CoreyMSchafer@gmail.com", 'JaneDoe@email.com', None, np.nan, "Anonymus@gmail.com", "NA"],
    "age": [31, 45, 67, 89, 90, 100, None, None, "Missing"]
}

df = pd.DataFrame(people_data)
print(df)
print("====================================")
print(df.dropna())
```

```
     first     last                    email      age
0   Gunjan    Kumar    GunjanKumar@gmail.com       31
1  Chandan    Singh   ChandanSingh@gmail.com       45
2   Kundan    Singh    KundanSingh@gmail.com       67
3    Corey  Schafer  CoreyMSchafer@gmail.com       89
4     Jane      Doe        JaneDoe@email.com       90
5     John      Doe                     None      100
6      NaN      NaN                      NaN     None
7     None      NaN       Anonymus@gmail.com     None
8       NA  Missing                       NA  Missing
====================================
     first     last                    email      age
0   Gunjan    Kumar    GunjanKumar@gmail.com       31
1  Chandan    Singh   ChandanSingh@gmail.com       45
2   Kundan    Singh    KundanSingh@gmail.com       67
3    Corey  Schafer  CoreyMSchafer@gmail.com       89
4     Jane      Doe        JaneDoe@email.com       90
8       NA  Missing                       NA  Missing
```

In [68]:
```
print(df.dropna(axis='index', how='any'))
```

```
     first     last                    email      age
0   Gunjan    Kumar    GunjanKumar@gmail.com       31
1  Chandan    Singh   ChandanSingh@gmail.com       45
2   Kundan    Singh    KundanSingh@gmail.com       67
3    Corey  Schafer  CoreyMSchafer@gmail.com       89
4     Jane      Doe        JaneDoe@email.com       90
8       NA  Missing                       NA  Missing
```

In [69]:
```
print(df.dropna(axis='index', how='all'))
```

```
     first     last                    email      age
0   Gunjan    Kumar    GunjanKumar@gmail.com       31
1  Chandan    Singh   ChandanSingh@gmail.com       45
2   Kundan    Singh    KundanSingh@gmail.com       67
3    Corey  Schafer  CoreyMSchafer@gmail.com       89
4     Jane      Doe        JaneDoe@email.com       90
5     John      Doe                     None      100
7     None      NaN       Anonymus@gmail.com     None
8       NA  Missing                       NA  Missing
```

In [70]:
```
print(df.dropna(axis='columns', how='all'))
print("================================")
print(df.dropna(axis='columns', how='any'))
```

```
     first     last                    email      age
0   Gunjan    Kumar    GunjanKumar@gmail.com       31
1  Chandan    Singh   ChandanSingh@gmail.com       45
2   Kundan    Singh    KundanSingh@gmail.com       67
3    Corey  Schafer  CoreyMSchafer@gmail.com       89
4     Jane      Doe        JaneDoe@email.com       90
5     John      Doe                     None      100
6      NaN      NaN                      NaN     None
7     None      NaN       Anonymus@gmail.com     None
8       NA  Missing                       NA  Missing
================================
Empty DataFrame
Columns: []
Index: [0, 1, 2, 3, 4, 5, 6, 7, 8]
```

In [71]:
```
print(df.dropna(axis='index', how='any', subset=["email"]))
```

```
     first     last                    email      age
0   Gunjan    Kumar    GunjanKumar@gmail.com       31
1  Chandan    Singh   ChandanSingh@gmail.com       45
2   Kundan    Singh    KundanSingh@gmail.com       67
3    Corey  Schafer  CoreyMSchafer@gmail.com       89
4     Jane      Doe        JaneDoe@email.com       90
7     None      NaN       Anonymus@gmail.com     None
8       NA  Missing                       NA  Missing
```

In [72]:
```
df.replace("NA", np.nan)
df.replace(["Missing", None], np.nan, inplace=True)
print(df)
```

```
     first     last                    email    age
0   Gunjan    Kumar    GunjanKumar@gmail.com   31.0
1  Chandan    Singh   ChandanSingh@gmail.com   45.0
2   Kundan    Singh    KundanSingh@gmail.com   67.0
3    Corey  Schafer  CoreyMSchafer@gmail.com   89.0
4     Jane      Doe        JaneDoe@email.com   90.0
5     John      Doe                      NaN  100.0
6      NaN      NaN                      NaN    NaN
7      NaN      NaN       Anonymus@gmail.com    NaN
8       NA      NaN                       NA    NaN
```

In [73]:
```
df.isna()
```

Out[73]:

	first	last	email	age
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
5	False	False	True	False
6	True	True	True	True
7	True	True	False	True
8	False	True	False	True

In [74]:
```
df.fillna("MISSING")
```

Out[74]:

	first	last	email	age
0	Gunjan	Kumar	GunjanKumar@gmail.com	31.0
1	Chandan	Singh	ChandanSingh@gmail.com	45.0
2	Kundan	Singh	KundanSingh@gmail.com	67.0
3	Corey	Schafer	CoreyMSchafer@gmail.com	89.0
4	Jane	Doe	JaneDoe@email.com	90.0
5	John	Doe	MISSING	100.0
6	MISSING	MISSING	MISSING	MISSING
7	MISSING	MISSING	Anonymus@gmail.com	MISSING
8	NA	MISSING	NA	MISSING

In [75]:
```
df.fillna(0)
```

Out[75]:

	first	last	email	age
0	Gunjan	Kumar	GunjanKumar@gmail.com	31.0
1	Chandan	Singh	ChandanSingh@gmail.com	45.0
2	Kundan	Singh	KundanSingh@gmail.com	67.0
3	Corey	Schafer	CoreyMSchafer@gmail.com	89.0
4	Jane	Doe	JaneDoe@email.com	90.0
5	John	Doe	0	100.0
6	0	0	0	0.0
7	0	0	Anonymus@gmail.com	0.0
8	NA	0	NA	0.0

In [76]:
```
df.dtypes
```

Out[76]:
```
first     object
last      object
email     object
age      float64
dtype: object
```

In [81]:
```
print(df["age"].median())
df["age"] = df["age"].astype(float)
print(df["age"])
```

```
78.0
0     31.0
1     45.0
2     67.0
3     89.0
4     90.0
5    100.0
6      NaN
7      NaN
8      NaN
Name: age, dtype: float64
```

In [90]:
```
na_vals = ["NA", "MISSING"]
data_2022_005 = pd.read_csv(".//stack-overflow-developer-survey-2022//survey_results_public.csv", index_col="
# data_2022_005 = pd.read_csv(".//stack-overflow-developer-survey-2022//survey_results_public.csv", index_col=
data_2022_005.YearsCode.unique()
data_2022_005.YearsCode.replace("Less than 1 year", 0, inplace=True)
data_2022_005.YearsCode.replace("More than 50 years", 51, inplace=True)
data_2022_005.YearsCode.unique()
```

Out[90]:
```
array([nan, '14', '20', '8', '15', '3', '1', '6', '37', '5', '12', '22',
       '11', '4', '7', '13', '36', '2', '25', '10', '40', '16', '27',
       '24', '19', '9', '17', '18', '26', 51, '29', '30', '32', 0, '48',
       '45', '38', '39', '28', '23', '43', '21', '41', '35', '50', '33',
       '31', '34', '46', '44', '42', '47', '49'], dtype=object)
```