2020/21 CSC 5741: Data Mining and Warehousing
Jupyter Notebook—Exploratory Data Analysis

Lighton Phiri
<lighton.phiri@unza.zm>

May 24, 2021

Contents

Introduction
General Notebook Configuration ................................................................. 2
Python Packages for Data Pre-processing .................................................. 3
Implementing Core Functions ................................................................. 3

Dataset #1: ICT 1110 Information Survey
Data Preprocessing ..................................................................................... 5
Dataset Description .................................................................................... 5
Dataframe Creation ................................................................................... 8
Dataset Attributes ..................................................................................... 9
Data Pre-processing Plan .......................................................................... 10
Exploratory Data Analysis ......................................................................... 14
Possible attributes to include in the EDA process .................................... 14
Dataframe Statistical Information ............................................................. 14
Minor Programme .................................................................................... 15
Computer Studies Elective in High School ................................................. 17
Experience With Computers .................................................................... 19
Programme Major Motivation ................................................................... 20

Dataset #2: 2018/19 ICT 1110 Student Demographics
Data Preprocessing ..................................................................................... 22
Dataset Description .................................................................................... 22
Dataset Files ............................................................................................. 22
Dataset Format .......................................................................................... 22
Dataframe Creation ................................................................................... 23
Dataset Attributes ..................................................................................... 25
Data Pre-processing Plan .......................................................................... 25
Exploratory Data Analysis ......................................................................... 29
Possible attributes to include in the EDA process .................................... 30
Dataframe Statistical Information ............................................................. 30
Date of Birth .............................................................................................. 31
Gender ....................................................................................................... 33
Minor Description ..................................................................................... 34
Sponsor ....................................................................................................... 35
Accommodated .......................................................................................... 36
Introduction

In this Jupyter Notebook, we walk through practical examples in order to illustrate how to perform Exploratory Data Analysis (EDA). In all instances, you will notice two key operations:

1. Basic descriptive statistical analysis
2. Extensive use of plots, graphs and/or charts

While the pre-processing activity was discussed in the previous lecture series, we also include “some” aspects of it, to serve as a reminder of tasks to be performed.

You will notice that the some examples use native Python features as opposed to libraries such as Pandas. This is done to highlight the flexibility that Python provides. In cases were they are not used, you are encouraged to explore how Pandas and other libraries can be used.

In all instances, you are encouraged to make reference to online documentation for the various tools. Additionally, you can exploit tools like Zeal Offline Documentation Browser to download and search through offline documentation. You are also encouraged to look up and explore other libraries, especially as you work towards the Mini Projects.

General Notebook Configuration

```python
# Aesthetics for pandas cell output
import pandas as pd

pd.set_option('display.latex.repr', True)
pd.set_option('display.latex.longtable', True)
pd.set_option('max_colwidth', 30)

# Show all Jupyter Notebook cell output
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```
Python Packages for Data Pre-processing

[2]: # Import all libraries and modules for use during lecture session code walkthrough
import matplotlib.pyplot as plt
import pandas as pd
import re
import seaborn as sns
import string
from collections import Counter
from IPython.core.interactiveshell import InteractiveShell
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from wordcloud import WordCloud

Implementing Core Functions

The generic functions in this section act as general utility functions, primarily for pre-processing. However, some of them perform specialised tasks.

[3]:

def fxn_case_folding(var_input):
    ""
    Preprocessing: Case Folding
    ""
    return var_input.lower()

def fxn_punctuation(var_input_text):
    ""
    Preprocessing: Punctuation Removal
    ""
    var_output_text = re.sub("[\s]\s*\s*[\s]", " ", var_input_text)
    var_output_text = re.sub("[\s]\s*\s*[\s]", " ", var_output_text)
    var_output_text = re.sub('\\w*\\d\\w*', '', var_output_text) # HINT: lookup isalpha()
    return var_output_text

def fxn_stopwords(var_input_text):
    ""
    Preprocessing: Stopwords Removal
    ""
    var_etd_stop = " ".join([
        var_etd_word for var_etd_word in var_input_text.split()
        if var_etd_word not in stopwords.words('english')
    ])  
    return var_etd_stop

def fxn_stem(var_input_text):
### Preprocessing: Stemming

```python
var_stemmer = PorterStemmer()
var_output_text = " ".join([
    var_stemmer.stem(var_etd_word) for var_etd_word in var_input_text.split()
])
return var_output_text
```

def fxn_normalise_ict1110_minors(var_input_minor):
    
    Returns normalised ICT 1110 minor
    
    if "civic" in var_input_minor.lower():
        var_output_minor = "Civic Education"
    elif "religion" in var_input_minor.lower() or "res" in var_input_minor.lower():
        var_output_minor = ""
    elif "history" in var_input_minor.lower():
        var_output_minor = "History"
    elif "art" in var_input_minor.lower():
        var_output_minor = "Art"
    elif "language" in var_input_minor.lower() or "french" in var_input_minor.lower():
        var_output_minor = "Languages"
    elif "geography" in var_input_minor.lower():
        var_output_minor = "Geography"
    elif "math" in var_input_minor.lower():
        var_output_minor = "Mathematics"
    elif "writing" in var_input_minor.lower():
        var_output_minor = "Writing Skill"
    else:
        var_output_minor = var_input_minor
    return var_output_minor.title() 
    ```

[4]: var_example_string = "This is an example string, used as part of CSC 5741 code snippets."

[5]: ```
##fxn_stopwords(var_example_string)
fxn_case_folding(var_example_string)
fxn_stopwords(fxn_case_folding(var_example_string))
fxn_punctuation(fxn_stopwords(fxn_case_folding(var_example_string)))
fxn_stem(fxn_punctuation(fxn_stopwords(fxn_case_folding(var_example_string))))
```
Dataset #1: ICT 1110 Information Survey

Data Preprocessing

**Link to dataset**

Students at enrolled into the “ICT 1110: Computer Systems and Architecture” course, at The University of Zambia, respond to a preliminary survey aimed at collecting background information about them. This is done using Google Forms.

**Dataset Description**

This dataset comprises of 25 student responses for the 2018/19 cohort and 73 responses for the 2019/20 cohort. The dataset has observations presented in CSV format, using “|” as the separator. In addition, each observation is associated with the following 13 data attributes: * Timestamp * Full Names * Student ID * Hometown (suburb/town/province—e.g. Kabwata/Lusaka/Lusaka) * What is your programme Minor (e.g. Mathematics, Languages) * What made you decide on your programme minor? * Why did you decide to major pursue the B.ICTs Ed. Programme? * Did you study Computer Studies at secondary school? * Have you undergone any computer related training? * If your response to the question above is year, please provide details of the type of course and/or training provided.

```bash
# Explore 2018/19 ICT 1110 survey
!cat -n db-unza21-csc5741-ict1110_2018_19-preliminary_survey.csv | head
```

1 2018/03/28 11:13:51 PM GMT+2|Participant1|#N/A|Chudleigh/Lusaka/Lusaka|Data Mining|I love data|I love computers|No|Yes|I have studied Computer Science|More than 5 years|Yes|I cycle everyday!
2 2019/03/28 11:55:27 PM GMT+2|Participant2|742b8abe5776a6d942a92ce77d7d84a0|Copper belt,luanshya,mpatamato|Mathematics|I find it easy to study and understand|Wanted to acquire more knowledge about ICTs and contribute to technology|No|No||1 to 2 years|Yes|A day doesn't pass by without a joke,I feel laughing will make you feel like you are in another world
3 2019/03/29 8:00:53 PM GMT+2|Participant3|921855f753932de762b780405a50bdf7|Mungule,senanga,western.|French|It was the best of my available options |"i have always wanted to do an ICT related program."|No|No||No Experience|Yes|
4 2019/03/30 11:25:30 AM GMT+2|Participant4|07f3ca235f5aa1c9ad16facef552688b|Lusaka|Religious studies|I just chose it|Because my results met the requirements |No|No||Less than 1 year|Yes|I like the internet
5 2019/03/31 3:26:35 AM GMT+2|Participant5|4234d1d794dd33c1b6ed975e5148040|Lusaka |Civic education |My first option was Chinese but it was a major and came with additional courses increasing my courses to more than four. So I ended up picking civic education because I found it easy in high school |I had written the same program twice on my application form so the man collecting suggested B. ICTs Ed|No|No||Less than 1 year|No

---

5
enjoy indie music

8 2019/03/31 1:49:53 PM
GMT+2|Participant6|9e7002d53d4db7bfad4f5cf419b0c126|shibuyunji/ central pronvince|civic education|i want to know more of my rights and responsibilities as a zambian citizen and take part in passing the knowledge to those who do not know much about their role in democratic governance. so that i can be part and parcel of the ever changing and developing digital world, and to take part in the zambia 2030 vision of having a digital zambia in all sectors of development. |No|No||No Experience|Yes|like exploring on IT technology

9 2019/03/31 1:51:38 PM
GMT+2|Participant7|9e7002d53d4db7bfad4f5cf419b0c126|shibuyunji/ central pronvince|civic education|i want to know more of my rights and responsibilities as a zambian citizen and take part in passing the knowledge to those who do not know much about their role in democratic governance. so that i can be part and parcel of the ever changing and developing digital world, and to take part in the zambia 2030 vision of having a digital zambia in all sectors of development. |No|No||No Experience|Yes|like exploring on IT technology

10 2019/04/01 7:12:07 PM
GMT+2|Participant8|fceb5af40df295d85851f390f4f8d78d|LUSAKA|RELIGIOUS STUDIES|TO HAVE KNOWLEDGE APPRECIATE OTHER RELIGIONS|I HAVE ALWAYS WANTED STUDY THIS PROGRAM |No|No||Less than 1 year|Yes|I ACCESS YOUTUBE ALMOST EVERYDAY

[7]: # Count the number of observations in the 2018/19 ICT 1110 survey
!wc db-unza21-csc5741-ict1110_2018_19-preliminary_survey.csv

43 1916 15361 db-unza21-csc5741-ict1110_2018_19-preliminary_survey.csv

[8]: # Explore 2019/20 ICT 1110 survey
!cat -n db-unza21-csc5741-ict1110_2019_20-preliminary_survey.csv | head

1 Timestamp|Full Names|Student ID|Hometown (suburb/town/province---e.g. Kabwata/Lusaka/Lusaka)|What is your programme Minor (e.g. Mathematics, Languages)|What made you decide on your programme minor?|Why did you decide to major pursue the B.ICTs Ed. Programme?|Did you study Computer Studies at secondary school?|Have you undergone any computer related training?|If your response to the question above is yes, please provide details of the type of course and/or training|How many years experience do you have using computers?|Do you currently own a computer or have regular access to one?|List one interesting fact about yourself (e.g. I cycle everyday!):

2 2020/03/04 7:57:29 PM GMT+2|Participant43|aa34dad971bfc1edc090076ef05be225|Libala lusaka|Mathematics |It proves to be a good combination and one of my strengths |Always had interest in technology |No|No||3 to 5 years|Yes|I can sing

3 2020/03/04 7:57:35 PM GMT+2|Participant43|aa34dad971bfc1edc090076ef05be225|Libala lusaka|Mathematics |It proves to be a good combination and one of my strengths |Always had interest in technology |No|No||3 to 5 years|Yes|I can sing

4 2020/03/05 3:26:13 PM
GMT+2|Participant12|a41f2bfc1a4c22c8e0aaf518f42bac0b|Rhodespark/Lusaka/Lusaka|Mathematics |A preference of courses that involve more of solving to studying. |Interest in technology and dream of being a established IT person. |No|No||More than 5 years|Yes|I love to keep up to date with the latest tech gadgets that keep being released.

5 2020/03/05 8:31:16 PM GMT+2|Participant36|ab7f1643a776fc9319422859c8869fd7|State Lodge|Mathematics|Because I like challenging thing and I love numbers|Because I wanted to know how computer works, know how to manipulate it and because when I was I kid always
wanted to use a computer but never had one home. | No | Yes | Office package | 1 to 2 years | Yes | I play the keyboard and little bit of singing

2020/03/05 8:50:28 PM

GMT+2 | Participant 6 | e997272aa3c790288c782a99f0e96b1d | Lusaka/Lusaka | Religious Studies | I would want to know more about world Religion and understand more on how the study of Religion can be the way of sorting out conflicts among Religious groups in society. | Because it was one of my favourite in secondary, though I only learnt it in grade 12 only...therefore, I would want to know more about it because we are now living in the Digital world. | Yes | No | No Experience | No | I read the Novels

2020/03/05 10:11:06 PM

GMT+2 | Participant 28 | 38e58148b7bc4c09777e880ec00d7aaf | Lusaka | Mathematics | The course seemed to be a good combination with my major i.e ICT| Its interesting to me personally and its diversity in the courses it has, is what inspires me the most. I get to choose anywhere to work from and most vital, working for my self is also an option. | No | No | More than 5 years | Yes |

8 2020/03/06 1:08:04 PM

GMT+2 | Participant 5 | 4f4c3be8f6b6ec9d0b157764c4abafd70 | Mandevu/Lusaka/Lusaka | Civic Education 1100 | I thought its a right course that can able me become one of agent of development in Zambia, as well as being a voice of many other Zambians who can’t come out loud and so on. | Looking at Zambias plans to have a project on technology development so by pursuing this program I’m very much sure that I will be helped to have knowledge as information about technology studies, I’m hoping i will be one of the students to run the developmental project. | No | No | Less than 1 year | No | I’m not good a speaker, but I’m very fast to grasp information.

10 2020/03/06 2:19:24 PM

GMT+2 | Participant 47 | 7f32c2c7fab80b6433f8818b8162c79c | Hillcrest/Ndola/Copperbelt | Languages | To improve my vocabulary | To become a developer | No | No | More than 5 years | No | I repair/fix hardware and software to calm myself

[9]:
# Count the number of observations in the 2019/20 ICT 1110 survey
# NOTE: Careful with shell commands as processing CSV files can be problematic, e.g. when counting records
# ! cat db-unza21-csc5741-ict1110_2019_20-preliminary_survey.csv | wc -l
91

[10]:
# Merge the 2018/19 and 2019/20 ICT 1110 survey results
# ! cat db-unza21-csc5741-ict1110_2018_19-preliminary_survey.csv > db-unza21-csc5741-ict1110_preliminary_survey.csv

[11]:
# Spool contents of 2019/20 into input file
# READ: https://stackoverflow.com/a/339941/664424
# One a UNIX-like OS, issue "man tail"
# ! tail +2 db-unza21-csc5741-ict1110_2019_20-preliminary_survey.csv >> db-unza21-csc5741-ict1110_preliminary_survey.csv
# Explore resulting merged input file
# NOTE: Careful with shell commands as processing CSV files can be problematic, e.g. when counting records
# ! cat db-unza21-csc5741-ict1110_preliminary_survey.csv | wc -l

133

## Dataframe Creation

# Create DataFrame of input dataset: ICT 1110 survey
#
var_ict1110_survey = pd.read_csv("db-unza21-csc5741-ict1110_preliminary_survey.csv", sep="|")
var_ict1110_survey.columns

[13]:

Index(['Timestamp', 'Full Names', 'Student ID', 'Hometown (suburb/town/province---e.g. Kabwata/Lusaka/Lusaka)', 'What is your programme Minor (e.g. Mathematics, Languages)', 'What made you decide on your programme minor?', 'Why did you decide to major pursue the B.ICTs Ed. Programme?', 'Did you study Computer Studies at secondary school?', 'Have you undergone any computer related training?', 'If your response to the question above is year, please provide details of the type of course and/or training', 'How many years experience do you have using computers?', 'Do you currently own a computer or have regular access to one', 'List one interesting fact about yourself (e.g. I cycle everyday!):'],
dtype='object')

# Rename dataframe columns for easy processing
#
var_ict1110_survey.rename(columns={"Full Names": "StudentName", "Student ID": "StudentID", "Hometown (suburb/town/province---e.g. Kabwata/Lusaka/Lusaka)": "HomeTown", "What is your programme Minor (e.g. Mathematics, Languages)": "MinorProgramme", "What made you decide on your programme minor?": "MinorProgrammeMotivation", "Why did you decide to major pursue the B.ICTs Ed. Programme?": "MajorProgrammeMotivation", "Did you study Computer Studies at secondary school?": "DidComputerStudies", "Have you undergone any computer related training?": "HasComputerTraining", "If your response to the question above is year, please provide details of the type of course and/or training": "ComputerTrainingType", "How many years experience do you have using computers?": "ExperienceWithComputers", ...})
"Do you currently own a computer or have regular access to one?": "HasComputerAccess",
"List one interesting fact about yourself (e.g. I cycle everyday!):": "AboutMe"}, inplace=True)

var_ict1110_survey.columns

[14]: Index(['Timestamp', 'StudentName', 'StudentID', 'HomeTown', 'MinorProgramme',
         'MinorProgrammeMotivation', 'MajorProgrammeMotivation',
         'DidComputerStudies', 'HasComputerTraining', 'ComputerTrainingType',
         'ExperienceWithComputers', 'HasComputerAccess', 'AboutMe'],
dtype='object')

[15]: # Count records in dataframe
#
len(var_ict1110_survey)

[15]: 112

[16]: # Inspect some of the records
#
var_ict1110_survey.head(3).T

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentName</td>
<td>Participant1</td>
<td>Participant2</td>
</tr>
<tr>
<td>StudentID</td>
<td>NaN</td>
<td>742b8abe5776a6d942a92ce7dc...</td>
</tr>
<tr>
<td>HomeTown</td>
<td>Chudleigh/Lusaka/Lusaka</td>
<td>Copperbelt,Juangsha,Mpatamato</td>
</tr>
<tr>
<td>MinorProgramme</td>
<td>Data Mining</td>
<td>Mathematics</td>
</tr>
<tr>
<td>MinorProgrammeMotivation</td>
<td>I love data</td>
<td>I find it easy to study an...</td>
</tr>
<tr>
<td>MajorProgrammeMotivation</td>
<td>I love computers</td>
<td>Wanted to acquire more kno...</td>
</tr>
<tr>
<td>DidComputerStudies</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>HasComputerTraining</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ComputerTrainingType</td>
<td>I have studied Computer Sc...</td>
<td>NaN</td>
</tr>
<tr>
<td>ExperienceWithComputers</td>
<td>More than 5 years</td>
<td>1 to 2 years</td>
</tr>
<tr>
<td>HasComputerAccess</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AboutMe</td>
<td>I cycle everyday!</td>
<td>A day doesn’t pass by with...</td>
</tr>
</tbody>
</table>

Dataset Attributes

- Timestamp—Date
- StudentName—Text
- StudentID—Alphanumeric
- Hometown—Text
- MinorProgramme—Text
- MinorProgrammeMotivation—Text
- MajorProgrammeMotivation—Text
- DidComputerStudies—Categorical
- HasComputerTraining—Categorical
- ComputerTrainingType—Text
Data Pre-processing Plan

- STEP 1: Remove duplicates using StudentID as unique field
- STEP 2: Apply case folding to all text attributes
- STEP 3: Remove punctuations from text attributes
- STEP 4: Remove stopwords from text attributes
- STEP 5: Stem all text attributes

NOTE: Null values to be handled on a case-by-case basis; e.g. null values in text attributes to be replaced with empty strings ""

```python
# STEP 1: Remove duplicates using StudentID as unique field
#
# Print duplicate records on StudentID
#
# Using Pandas, the df.duplicated() function can be used to identify duplicates

var_ict1110_survey.duplicated(['StudentID'], keep=False)

# Confirm duplicate records removal
len(var_ict1110_survey)
```

```python
# Remove duplicate records on StudentID
#
# drop_duplicates uses keep=[First, Last, False]
#
var_ict1110_survey.drop_duplicates(['StudentID'], keep='first', inplace=True)
```

```python
# Confirm duplicate records removal
len(var_ict1110_survey)
```
# STEP 2: Apply steps 2--5 to Text Attributes recursively
# Attributes: MinorProgrammeMotivation, MajorProgrammeMotivation and AboutMe
# Inspect dataframe before pre-processing
var_ict1110_survey.head(2).T

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentName</td>
<td>Participant1</td>
</tr>
<tr>
<td>StudentID</td>
<td>NaN</td>
</tr>
<tr>
<td>HomeTown</td>
<td>Chudleigh/Lusaka/Lusaka</td>
</tr>
<tr>
<td>MinorProgramme</td>
<td>Data Mining</td>
</tr>
<tr>
<td>MinorProgrammeMotivation</td>
<td>I love data</td>
</tr>
<tr>
<td>MajorProgrammeMotivation</td>
<td>I love computers</td>
</tr>
<tr>
<td>DidComputerStudies</td>
<td>No</td>
</tr>
<tr>
<td>HasComputerTraining</td>
<td>Yes</td>
</tr>
<tr>
<td>ComputerTrainingType</td>
<td>I have studied Computer Sc...</td>
</tr>
<tr>
<td>ExperienceWithComputers</td>
<td>More than 5 years</td>
</tr>
<tr>
<td>HasComputerAccess</td>
<td>Yes</td>
</tr>
<tr>
<td>AboutMe</td>
<td>I cycle everyday!</td>
</tr>
</tbody>
</table>

# Handle null values---NaN
#
var_ict1110_survey['MinorProgrammeMotivation'].fillna('', inplace=True)
var_ict1110_survey['MajorProgrammeMotivation'].fillna('', inplace=True)
var_ict1110_survey['AboutMe'].fillna('', inplace=True)

# Apply Case Folding
#
var_ict1110_survey['MinorProgrammeMotivation'] = var_ict1110_survey['MinorProgrammeMotivation'].apply(fxn_case_folding)
var_ict1110_survey['MajorProgrammeMotivation'] = var_ict1110_survey['MajorProgrammeMotivation'].apply(fxn_case_folding)
var_ict1110_survey['AboutMe'] = var_ict1110_survey['AboutMe'].apply(fxn_case_folding)

# Inspect dataframe after pre-processing
var_ict1110_survey.head(2).T

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentName</td>
<td>Participant1</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StudentID</strong></td>
<td>NaN</td>
<td>742b8abe5776a6d942a92ce7dc...</td>
</tr>
<tr>
<td><strong>HomeTown</strong></td>
<td>Chudleigh/Lusaka/Lusaka</td>
<td>Copperbelt,luanshya,Mpatamato</td>
</tr>
<tr>
<td><strong>MinorProgramme</strong></td>
<td>Data Mining</td>
<td>Mathematics</td>
</tr>
<tr>
<td><strong>MinorProgrammeMotivation</strong></td>
<td>i love data</td>
<td>i find it easy to study an...</td>
</tr>
<tr>
<td><strong>MajorProgrammeMotivation</strong></td>
<td>i love computers</td>
<td>wanted to acquire more kno...</td>
</tr>
<tr>
<td><strong>DidComputerStudies</strong></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>HasComputerTraining</strong></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>ComputerTrainingType</strong></td>
<td>I have studied Computer Sc...</td>
<td>NaN</td>
</tr>
<tr>
<td><strong>ExperienceWithComputers</strong></td>
<td>More than 5 years</td>
<td>1 to 2 years</td>
</tr>
<tr>
<td><strong>HasComputerAccess</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>AboutMe</strong></td>
<td>i cycle everyday!</td>
<td>a day doesn’t pass by with...</td>
</tr>
</tbody>
</table>

[24]:

```python
# Apply Punctuation Removal
#
var_ict1110_survey["MinorProgrammeMotivation"] =
    ~var_ict1110_survey["MinorProgrammeMotivation"].apply(fxn_punctuation)

var_ict1110_survey["MajorProgrammeMotivation"] =
    ~var_ict1110_survey["MajorProgrammeMotivation"].apply(fxn_punctuation)

var_ict1110_survey["AboutMe"] = var_ict1110_survey["AboutMe"].apply(fxn_punctuation)
```

[25]:

```python
# Inspect dataframe after pre-processing
var_ict1110_survey.head(2)
```

[26]:

```python
# Apply Stopwords Removal
#
var_ict1110_survey["MinorProgrammeMotivation"] =
    ~var_ict1110_survey["MinorProgrammeMotivation"].apply(fxn_stopwords)

var_ict1110_survey["MajorProgrammeMotivation"] =
    ~var_ict1110_survey["MajorProgrammeMotivation"].apply(fxn_stopwords)
```
var_ict1110_survey["AboutMe"] = var_ict1110_survey["AboutMe"].apply(fxn_stopwords)

# Inspect dataframe after pre-processing
var_ict1110_survey.head(2)

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentName</td>
<td>Participant1</td>
</tr>
<tr>
<td>StudentID</td>
<td>NaN</td>
</tr>
<tr>
<td>HomeTown</td>
<td>Chudleigh/Lusaka/Lusaka</td>
</tr>
<tr>
<td>MinorProgramme</td>
<td>Data Mining</td>
</tr>
<tr>
<td>MinorProgrammeMotivation</td>
<td>love data</td>
</tr>
<tr>
<td>MajorProgrammeMotivation</td>
<td>love computers</td>
</tr>
<tr>
<td>DidComputerStudies</td>
<td>No</td>
</tr>
<tr>
<td>HasComputerTraining</td>
<td>Yes</td>
</tr>
<tr>
<td>ComputerTrainingType</td>
<td>I have studied Computer Sc...</td>
</tr>
<tr>
<td>ExperienceWithComputers</td>
<td>More than 5 years</td>
</tr>
<tr>
<td>HasComputerAccess</td>
<td>Yes</td>
</tr>
<tr>
<td>AboutMe</td>
<td>cycle everyday</td>
</tr>
</tbody>
</table>

# Apply Stem Removal

# var_ict1110_survey["MinorProgrammeMotivation"] = var_ict1110_survey["MinorProgrammeMotivation"].apply(fxn_stem)

# var_ict1110_survey["MajorProgrammeMotivation"] = var_ict1110_survey["MajorProgrammeMotivation"].apply(fxn_stem)

# var_ict1110_survey["AboutMe"] = var_ict1110_survey["AboutMe"].apply(fxn_stem)

# Inspect dataframe after pre-processing
var_ict1110_survey.head(2)

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentName</td>
<td>Participant1</td>
</tr>
<tr>
<td>StudentID</td>
<td>NaN</td>
</tr>
<tr>
<td>HomeTown</td>
<td>Chudleigh/Lusaka/Lusaka</td>
</tr>
<tr>
<td>MinorProgramme</td>
<td>Data Mining</td>
</tr>
<tr>
<td>MinorProgrammeMotivation</td>
<td>love data</td>
</tr>
<tr>
<td>MajorProgrammeMotivation</td>
<td>love comput</td>
</tr>
<tr>
<td>DidComputerStudies</td>
<td>No</td>
</tr>
<tr>
<td>HasComputerTraining</td>
<td>Yes</td>
</tr>
<tr>
<td>ComputerTrainingType</td>
<td>I have studied Computer Sc...</td>
</tr>
<tr>
<td>ExperienceWithComputers</td>
<td>More than 5 years</td>
</tr>
<tr>
<td>HasComputerAccess</td>
<td>Yes</td>
</tr>
<tr>
<td>AboutMe</td>
<td>cycle everyday</td>
</tr>
</tbody>
</table>
Exploratory Data Analysis

# Describe the data
# Identify attributes to explore
```
var_ict1110_survey.head(3).T
```

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>2019/03/28 11:13:51 PM GMT+2</td>
<td>2019/03/28 11:55:27 PM GMT+2</td>
<td>2019/03/29 8:00:53 PM GMT+2</td>
</tr>
<tr>
<td>StudentName</td>
<td>Participant1</td>
<td>Participant2</td>
<td>Participant3</td>
</tr>
<tr>
<td>StudentID</td>
<td>NaN</td>
<td>742b8abe5776a6d942a92ce7dc...</td>
<td>921855f753932de762b780405a...</td>
</tr>
<tr>
<td>HomeTown</td>
<td>Chudleigh/Lusaka/Lusaka</td>
<td>Copperbelt,luanshya,Mpatamato</td>
<td>Mungule,sen...</td>
</tr>
<tr>
<td>MinorProgramme</td>
<td>Data Mining</td>
<td>Mathematics</td>
<td>French</td>
</tr>
<tr>
<td>MinorProgrammeMotivation</td>
<td>love data</td>
<td>find easi studi understand</td>
<td>best avail opt</td>
</tr>
<tr>
<td>MajorProgrammeMotivation</td>
<td>love comput</td>
<td>want acquir knowledg ict c...</td>
<td>alway want i...</td>
</tr>
<tr>
<td>DidComputerStudies</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>HasComputerTraining</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ComputerTrainingType</td>
<td>I have studied Computer Sc...</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>ExperienceWithComputers</td>
<td>More than 5 years</td>
<td>1 to 2 years</td>
<td>No Experience</td>
</tr>
<tr>
<td>HasComputerAccess</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AboutMe</td>
<td>cycl everyday</td>
<td>day pass without joke feel...</td>
<td></td>
</tr>
</tbody>
</table>

Possible attributes to include in the EDA process

- Home Town
- Minor Programme
- Minor Programme Motivation
- Major Programme Motivation
- Computer Studies Elective in High School
- Prior Computing Training
- Prior Computing Training Type
- Experience Working With Computers
- Access to a Computer

# Define variable to
```
var_ict1110_survey_eda = var_ict1110_survey
```

# Create new column to mark different academic years
```
var_ict1110_survey_eda["year"] = var_ict1110_survey_eda["Timestamp"][str[:4]]
```

Dataframe Statistical Information

# Use describe function to get statistical information of dataset attributes
```
var_ict1110_survey_eda.describe(include=’all’).T
```

```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Count</th>
<th>Unique</th>
<th>Top</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>90</td>
<td>90</td>
<td>2020/03/16 6:44:54 PM GMT+2</td>
<td>1</td>
</tr>
<tr>
<td>StudentName</td>
<td>88</td>
<td>55</td>
<td>Participant21</td>
<td>2</td>
</tr>
<tr>
<td>StudentID</td>
<td>89</td>
<td>89</td>
<td>e2a96e074e1d8a6f6de56abbd4...</td>
<td>1</td>
</tr>
<tr>
<td>HomeTown</td>
<td>90</td>
<td>74</td>
<td>Lusaka</td>
<td>11</td>
</tr>
<tr>
<td>MinorProgramme</td>
<td>90</td>
<td>46</td>
<td>Mathematics</td>
<td>12</td>
</tr>
<tr>
<td>MinorProgrammeMotivation</td>
<td>90</td>
<td>89</td>
<td>passion</td>
<td>2</td>
</tr>
<tr>
<td>MajorProgrammeMotivation</td>
<td>90</td>
<td>88</td>
<td>love technolog</td>
<td>2</td>
</tr>
<tr>
<td>DidComputerStudies</td>
<td>90</td>
<td>2</td>
<td>No</td>
<td>78</td>
</tr>
<tr>
<td>HasComputerTraining</td>
<td>90</td>
<td>2</td>
<td>No</td>
<td>71</td>
</tr>
<tr>
<td>ComputerTrainingType</td>
<td>21</td>
<td>19</td>
<td>Computer networking and ha...</td>
<td>2</td>
</tr>
<tr>
<td>ExperienceWithComputers</td>
<td>90</td>
<td>5</td>
<td>No Experience</td>
<td>31</td>
</tr>
<tr>
<td>HasComputerAccess</td>
<td>90</td>
<td>2</td>
<td>Yes</td>
<td>64</td>
</tr>
<tr>
<td>AboutMe</td>
<td>90</td>
<td>84</td>
<td>sing</td>
<td>3</td>
</tr>
<tr>
<td>year</td>
<td>90</td>
<td>2</td>
<td>2020</td>
<td>57</td>
</tr>
</tbody>
</table>

Minor Programme

[34]:
```r
# Minor Programme
#
# Apply attribute specific processing
var_ict1110_survey_eda["MinorProgramme"] = var_ict1110_survey_eda["MinorProgramme"].apply(fxn_case_folding)
```

[35]:
```r
# Basic dataframe summaries
#
# Get unique entries
var_ict1110_survey_eda["MinorProgramme"].unique()

# Count observations
var_ict1110_survey_eda["MinorProgramme"].count()

# Get value counts
var_ict1110_survey_eda["MinorProgramme"].value_counts()
```

[35]:
```r
array(['Data Mining', 'Mathematics', 'Languages', '', 'Civic Education', 'History', 'Art', 'Geography', 'Writing Skill', 'Drawing And Painting', 'English', 'Ict', 'Physical Education And Sports Pes'], dtype=object)
```

[35]:
```r
90
```

[35]:
```r
```
<table>
<thead>
<tr>
<th>MinorProgramme</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Civic Education</td>
<td>28</td>
</tr>
<tr>
<td>Mathematics</td>
<td>24</td>
</tr>
<tr>
<td>Languages</td>
<td>10</td>
</tr>
<tr>
<td>History</td>
<td>8</td>
</tr>
<tr>
<td>Art</td>
<td>3</td>
</tr>
<tr>
<td>Geography</td>
<td>2</td>
</tr>
<tr>
<td>Physical Education And Spor...</td>
<td>1</td>
</tr>
<tr>
<td>Writing Skill</td>
<td>1</td>
</tr>
<tr>
<td>Drawing And Painting</td>
<td>1</td>
</tr>
<tr>
<td>Data Mining</td>
<td>1</td>
</tr>
<tr>
<td>Ict</td>
<td>1</td>
</tr>
<tr>
<td>English</td>
<td>1</td>
</tr>
</tbody>
</table>

```python
# Facet results by academic year
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(10,10))
fig.suptitle('What is Your Minor Programme?')

### var_ict1110_survey_eda.groupby(["ExperienceWithComputers"]).size().plot(kind="barh", title="All Students", ax=ax1)
var_ict1110_survey_eda[var_ict1110_survey_eda["Timestamp"].str[:
                           4]=="2019"["MinorProgramme"].value_counts().plot(kind=barh", title="2018/19", ax=ax2)
var_ict1110_survey_eda[var_ict1110_survey_eda["Timestamp"].str[:
```

[36]: Text(0.5, 0.98, 'What is Your Minor Programme?')

[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f157139aac8>

[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1571361d30>

[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f15713122b0>
Computer Studies Elective in High School

# 2. Computer Studies Elective in High School
#
# Basic dataframe summaries
#

# Get unique entries
var_ict1110_survey_eda["DidComputerStudies"].unique()

# Count observations
var_ict1110_survey_eda["DidComputerStudies"].count()

# Get value counts
var_ict1110_survey_eda["DidComputerStudies"].value_counts()

array(["No", "Yes"], dtype=object)

90
<table>
<thead>
<tr>
<th>DidComputerStudies</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

# Facet results by academic year

```python
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15, 5))
fig.suptitle('Did You Take Computer Studies as an Elective in Highschool?')

var_ict1110_survey_eda.groupby(["DidComputerStudies"]).size().plot(kind="pie", title="All Students", ax=ax1)

var_ict1110_survey_eda[var_ict1110_survey_eda["Timestamp"].str[:<1]=="2"]["DidComputerStudies"].value_counts().plot(kind="pie", title="2018/19", ax=ax2)

var_ict1110_survey_eda[var_ict1110_survey_eda["Timestamp"].str[:<4]=="2019"]["DidComputerStudies"].value_counts().plot(kind="pie", title="2019/20", ax=ax3)
```

Text(0.5, 0.98, 'Did You Take Computer Studies as an Elective in Highschool?')

<matplotlib.axes._subplots.AxesSubplot at 0x7f15707424e0>

<matplotlib.axes._subplots.AxesSubplot at 0x7f1570753b00>

<matplotlib.axes._subplots.AxesSubplot at 0x7f15711b5f98>
Experience With Computers

# 2. Computer Studies Elective in High School
# Basic dataframe summaries

# Get unique entries
var_ict1110_survey_eda["ExperienceWithComputers"].unique()

# Count observations
var_ict1110_survey_eda["ExperienceWithComputers"].count()

# Get value counts
var_ict1110_survey_eda["ExperienceWithComputers"].value_counts()

array(['More than 5 years', '1 to 2 years', 'No Experience', 'Less than 1 year', '3 to 5 years'], dtype=object)

90

<table>
<thead>
<tr>
<th>ExperienceWithComputers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No Experience</td>
<td>31</td>
</tr>
<tr>
<td>Less than 1 year</td>
<td>24</td>
</tr>
<tr>
<td>More than 5 years</td>
<td>21</td>
</tr>
<tr>
<td>1 to 2 years</td>
<td>11</td>
</tr>
<tr>
<td>3 to 5 years</td>
<td>3</td>
</tr>
</tbody>
</table>

# Facet results by academic year

fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(10,8))

fig.suptitle('How Much Experience do You Have Working With Computers?')

var_ict1110_survey_eda.groupby(["ExperienceWithComputers"]).size().plot(kind="barh", title="All Students", ax=ax1)

var_ict1110_survey_eda[var_ict1110_survey_eda["Timestamp"].str[:
1]=="2019"["ExperienceWithComputers"].value_counts().plot(kind="barh", title="2018/
Students", ax=ax1)

var_ict1110_survey_eda[var_ict1110_survey_eda["Timestamp"].str[:
2019", ax=ax2)

var_ict1110_survey_eda[var_ict1110_survey_eda["Timestamp"].str[:
2020", ax=ax3)

Text(0.5, 0.98, 'How Much Experience do You Have Working With Computers?')

<matplotlib.axes._subplots.AxesSubplot at 0x7f1571195ac8>
Programme Major Motivation

```python
var_ict1110_major_motivation = WordCloud(stopwords=stopwords.words("english"),
                                           background_color="white",
                                           colormap="Dark2",
                                           max_font_size=150,
                                           random_state=42)

var_ict1110_survey_eda_motivation = var_ict1110_survey_eda["MajorProgrammeMotivation"]
var_ict1110_major_motivation.generate(''.join(var_ict1110_survey_eda_motivation))

plt.figure(figsize=(15,10))
plt.imshow(var_ict1110_major_motivation)
plt.axis("off")
```

![Graph showing experience with computers](image)

How Much Experience do You Have Working With Computers?

<table>
<thead>
<tr>
<th>Experience Level</th>
<th>2018/19</th>
<th>2019/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 5 years</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 to 2 years</td>
<td></td>
<td></td>
</tr>
<tr>
<td>More than 5 years</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Less than 1 year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Experience</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# DataLab keywords

var_datalab = pd.read_csv("/home/lightonphiri/Dropbox/Projects/work/2021/misc-unza21-datalab/db-unza21-datalab_output.csv", sep="|")

# Stopword removal; punctuation removal and case folding

var_datalab_wordcloud = WordCloud(stopwords=stopwords.words("english"), background_color="white", colormap="Dark2", max_font_size=150, random_state=42)

var_datalab_research = var_datalab["Abstract"]
var_datalab_wordcloud.generate(' '.join(var_datalab_research))

plt.figure(figsize = (15,5))
plt.imshow(var_datalab_wordcloud)
plt.axis("off")
Dataset #2: 2018/19 ICT 1110 Student Demographics

Using the UNZA 2018/19 ICT 1110 Student Demographics dataset, work towards the following: 1. Pre-process the datasets 3. Perform Exploratory Data Analysis on the merged dataset

Data Preprocessing

Dataset Description

Dataset Files

The files are categorised as follows: * db-unza20-csc5741-ict1110_student_demographics.csv * Student demographic details extracted from the Student Information System

Dataset Format

- Student Demographic Details
  - A total of 13 fields representing student demographics
  - The fields are pipe (“|”) seperated

```
# Datasets format
!cat -n db-unza21-csc5741-ict1110_student_demographics.csv | head
```

```bash
1 Student ID|DateOfBirth|Gender|Academic Year|Year Of Study|School|Program|MajorDescription|MinorDescription|Status|Sponsor|Nationality|Accommodated
2 9d5116a2451bc98c2b46b93acbc1b4fd0|1998-09-14|F|20191|2nd Year|EDUCATION|BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed)|ICTs and Education|RELIGIOUS STUDIES|Registered|GRZ - 75 PERCENT|ZAMBIAN|Yes
3 e7400496f1ce70cb62c2c44ca2dcd469|2000-03-23|M|20191|2nd Year|EDUCATION|BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed)|ICTs and Education|RELIGIOUS STUDIES|Registered|GRZ - 75 PERCENT|ZAMBIAN|Yes
```
OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed) | ICTs and Education | ART AND DESIGN STUDIES | Registered | GRZ-FULLY SPONSORED | ZAMBIAN | No
4 cea34f64bf356c28fc2b766ae46bb66c|1996-06-06|M|20171|1st Year|EDUCATION|BACHELOR
OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed) | ICTs and Education | GEOGRAPHY | Registered | GRZ-FULLY SPONSORED | ZAMBIAN | Yes
5 #N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|#N/A|No
6 6cd50fb3091b09d3c1ac2cf52441390|1999-03-03|M|20181|1st Year|EDUCATION|BACHELOR
OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed) | ICTs and Education | MATHEMATICS | Registered | GRZ-FULLY SPONSORED | ZAMBIAN | No
7 e31959fe2842dacea4d16d36e9813620|1995-07-27|F|20171|1st Year|EDUCATION|BACHELOR
OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed) | ICTs and Education | HISTORY | Registered | GRZ-FULLY SPONSORED | ZAMBIAN | No
8 97527dec0ae1a703599581d4f25dfbce|1995-06-26|F|20181|2nd Year|EDUCATION|BACHELOR
OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed) | ICTs and Education | CIVIC EDUCATION | Registered | GRZ-FULLY SPONSORED | ZAMBIAN | Yes
9 9e7002d53e5db7e9f9a23c9163|1999-04-04|F|20181|1st Year|EDUCATION|BACHELOR

Dataframe Creation

# Create DataFrame of input dataset: ICT 1110 Demographics

```python
var_ict1110_demographics = pd.read_csv("db-unza21-csc5741-ict1110_student_demographics.csv", sep="|")
```

var_ict1110_demographics.columns

```python
Index(['Student ID', 'DateOfBirth', 'Gender', 'Academic Year', 'Year Of Study', 'School', 'Program', 'MajorDescription', 'MinorDescription', 'Status', 'Sponsor', 'Nationality', 'Accommodated'],
dtype='object')
```

```python
var_ict1110_demographics.columns
var_ict1110_demographics.rename(columns={
    "Student ID": "StudentID",
    "Academic Year": "AcademicYear",
    "Year Of Study": "YearOfStudy"
}).head(2)
```

```python
var_ict1110_demographics.columns
```

```python
var_ict1110_demographics = var_ict1110_demographics.rename(columns={
    "Student ID": "StudentID",
    "Academic Year": "AcademicYear",
    "Year Of Study": "YearOfStudy"
})
```
var_ict1110_demographics

[46]: Index(['Student ID', 'DateOfBirth', 'Gender', 'Academic Year', 'Year Of Study', 
'School', 'Program', 'MajorDescription', 'MinorDescription', 'Status', 
'Sponsor', 'Nationality', 'Accommodated'],
dtype='object')

[46]:

<table>
<thead>
<tr>
<th>StudentID</th>
<th>DateOfBirth</th>
<th>Gender</th>
<th>AcademicYear</th>
<th>YearOfStudy</th>
<th>School</th>
<th>Program</th>
<th>MajorDescription</th>
<th>MinorDescription</th>
<th>Status</th>
<th>Sponsor</th>
<th>Nationality</th>
<th>Accommodated</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9d5116a2451bc98c2b46b93acb...</td>
<td>1998-09-14</td>
<td>F</td>
<td>20191.0</td>
<td>2nd Year</td>
<td>EDUCATION</td>
<td>BACHELOR OF INFORMATION TECHNOLOGY</td>
<td>ICTs and Education RELIGIOUS STUDIES</td>
<td>Registered</td>
<td>GRZ - 75 PERCENT</td>
<td>ZAMBIAN</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>e7400496f1ce70cb62c2c44ca2...</td>
<td>2000-03-23</td>
<td>M</td>
<td>20191.0</td>
<td>2nd Year</td>
<td>EDUCATION</td>
<td>BACHELOR OF INFORMATION TECHNOLOGY</td>
<td>ICTs and Education ART AND DESIGN STUDIES</td>
<td>Registered</td>
<td>GRZ-FULLY SPONSORED</td>
<td>ZAMBIAN</td>
<td>No</td>
</tr>
</tbody>
</table>

[46]:

[46]:

[47]: # Rename dataframe columns for easy processing
# # Function use: pd.rename([],)
# # Important parameter: inplace=True
# #
# var_ict1110_demographics.rename(columns={
#     "Student ID": "StudentID",
#     "Academic Year": "AcademicYear",
#     "Year Of Study": "YearOfStudy"
# }, inplace=True)

var_ict1110_demographics.columns

[47]: Index(['Student ID', 'DateOfBirth', 'Gender', 'Academic Year', 'YearOfStudy', 
'School', 'Program', 'MajorDescription', 'MinorDescription', 'Status', 
'Sponsor', 'Nationality', 'Accommodated'],
dtype='object')

[48]: # Count records in dataframe
#
len(var_ict1110_demographics)

[48]: 60

[49]: # Inspect some dataframe records
# var_ict1110_demographics.head(2).T
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StudentID</strong></td>
<td>9d5116a2451bc98c2b46b93acb...</td>
</tr>
<tr>
<td><strong>DateOfBirth</strong></td>
<td>1998-09-14</td>
</tr>
<tr>
<td><strong>Gender</strong></td>
<td>F</td>
</tr>
<tr>
<td><strong>AcademicYear</strong></td>
<td>20191</td>
</tr>
<tr>
<td><strong>YearOfStudy</strong></td>
<td>2nd Year</td>
</tr>
<tr>
<td><strong>School</strong></td>
<td>EDUCATION</td>
</tr>
<tr>
<td><strong>Program</strong></td>
<td>BACHELOR OF INFORMATION AN...</td>
</tr>
<tr>
<td><strong>MajorDescription</strong></td>
<td>ICTs and Education</td>
</tr>
<tr>
<td><strong>MinorDescription</strong></td>
<td>RELIGIOUS STUDIES</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Registered</td>
</tr>
<tr>
<td><strong>Sponsor</strong></td>
<td>GRZ - 75 PERCENT</td>
</tr>
<tr>
<td><strong>Nationality</strong></td>
<td>ZAMBIAN</td>
</tr>
<tr>
<td><strong>Accommodated</strong></td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Dataset Attributes**

- StudentID—Alphanumeric
- DateOfBirth—Date
- Gender—Text/Char
- AcademicYear—Text
- YearOfStudy—Text
- School—Text
- Programme—Text
- MajorDescription—Text
- MinorDescription—Text
- Status—Categorical (Dichotomus)
- Sponsor—Text
- Nationality—Text
- Accommodated—Categorical (Dichotomus)

**Data Pre-processing Plan**

- STEP 1: Remove all records with NULL StudentID values
- STEP 2: Remove duplicates using StudentID as unique field

NOTE: Null values to be handled on a case-by-case basis; e.g. null values in text attributes to be replaced with empty strings ""
```python
len(var_ict1110_demographics[var_ict1110_demographics["StudentID"].isnull()])
var_ict1110_demographics[var_ict1110_demographics["StudentID"].isnull()]
```

```python
len(var_ict1110_demographics.dropna(subset = ["StudentID"]))
```

```python
# Remove records with NULL StudentID values
# var_ict1110_demographics.dropna(subset = ["StudentID"])
# var_ict1110_demographics =
var_ict1110_demographics[var_ict1110_demographics["StudentID"].notna()]
# var_ict1110_demographics.dropna(subset = ["StudentID"], inplace=True)
```

```python
len(var_ict1110_demographics)
```

```python
var_ict1110_demographics_ = var_ict1110_demographics.dropna(subset = ["StudentID"])
len(var_ict1110_demographics_)
```

```python
len(var_ict1110_demographics)
```

```python
var_ict1110_demographics.duplicated()
```

```python
# STEP 2: Remove duplicates using StudentID as unique field
# Print duplicate records on StudentID
# Using Pandas, the df.duplicated() function can be used to identify duplicates
# var_ict1110_survey.duplicated(["StudentID"], keep=False)
# var_ict1110_demographics[var_ict1110_demographics.duplicated(["StudentID"],
# keep="first")].sort_values("StudentID").tail(2).T
```

```python
Empty DataFrame Columns: Int64Index([], dtype='int64') Index: Index(['StudentID', 'DateOfBirth', 'Gender', 'AcademicYear', 'YearOfStudy', 'School', 'Program', 'MajorDescription', 'MinorDescription', 'Status', 'Sponsor', 'Nationality', 'Accommodated'], dtype='object')
```

```python
# Remove duplicate records on StudentID: a record without a Student ID is useless
# drop_duplicates uses keep=[First, Last, False]
```

26
# var_ict1110_survey.drop_duplicates(["StudentID"], keep="first", inplace=True)

[58]: # Confirm duplicate records removal
len(var_ict1110_demographics)

[58]: 58

[59]: # STEP 2: Apply steps 3--6 to Text Attributes recursively
# Attributes: MinorProgrammeMotivation, MajorProgrammeMotivation and AboutMe
# #
# # Inspect dataframe before pre-processing
var_ict1110_demographics.sample(2).T

<table>
<thead>
<tr>
<th>21</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentID</td>
<td>43f9f6077d6be60269ad8cfb3f...</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>1999-06-28</td>
</tr>
<tr>
<td>Gender</td>
<td>F</td>
</tr>
<tr>
<td>AcademicYear</td>
<td>20191</td>
</tr>
<tr>
<td>YearOfStudy</td>
<td>2nd Year</td>
</tr>
<tr>
<td>School</td>
<td>EDUCATION</td>
</tr>
<tr>
<td>Program</td>
<td>BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed)</td>
</tr>
<tr>
<td>MajorDescription</td>
<td>ICTs and Education</td>
</tr>
<tr>
<td>MinorDescription</td>
<td>HISTORY</td>
</tr>
<tr>
<td>Status</td>
<td>Registered</td>
</tr>
<tr>
<td>Sponsor</td>
<td>GRZ-FULLY SPONSORED</td>
</tr>
<tr>
<td>Nationality</td>
<td>ZAMBIAN</td>
</tr>
<tr>
<td>Accommodated</td>
<td>No</td>
</tr>
</tbody>
</table>

[60]: # Handle null values---NaN
#
# DateOfBirth: Replace NaN with "MISSING DATA"
# Gender: Replace NaN with U, for Unknown
# AcademicYear: drop value
# YearOfStudy: drop value
# School: Replace with EDUCATION
# Programme: Replace with BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed)
# MajorDescription: Replace with ICTs and Education
# MinorDescription: Replace with "MISSING DATA"
# Status: Replace with "MISSING DATA"
# Sponsor: Replace with SELF-SPONSORED
# Nationality: Replace with ZAMBIAN
# Accommodated: Replace with No
#

[61]: # ALWAYS check data types of columns to ensure you are replacing them with appropriate values
#
```python
var_ict1110_demographics.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 58 entries, 0 to 59
Data columns (total 13 columns):
 # Column             Non-Null Count Dtype
--- ------ -------------- -----  
0 StudentID          58 non-null   object
1 DateOfBirth       58 non-null   object
2 Gender            58 non-null   object
3 AcademicYear      58 non-null   float64
4 YearOfStudy       58 non-null   object
5 School            58 non-null   object
6 Program           58 non-null   object
7 MajorDescription  58 non-null   object
8 MinorDescription  46 non-null   object
9 Status            58 non-null   object
10 Sponsor          58 non-null   object
11 Nationality      58 non-null   object
12 Accommodated     58 non-null   object
dtypes: float64(1), object(12)
memory usage: 6.3+ KB

[62]:
# DateOfBirth
var_ict1110_demographics["DateOfBirth"].fillna("MISSING DATA", inplace=True)

# Gender
var_ict1110_demographics["Gender"].fillna("U", inplace=True)

# AcademicYear
var_ict1110_demographics["AcademicYear"].fillna("MISSING DATA", inplace=True)

# YearOfStudy
var_ict1110_demographics["YearOfStudy"].fillna("MISSING DATA", inplace=True)

# School
var_ict1110_demographics["School"].fillna("EDUCATION", inplace=True)

# Program
var_ict1110_demographics["Program"].fillna("BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGIES WITH EDUCATION (B.ICTs.Ed)", inplace=True)

# MajorDescription
var_ict1110_demographics["MajorDescription"].fillna("MISSING DATA", inplace=True)

# MinorDescription
var_ict1110_demographics["MinorDescription"].fillna("MISSING DATA", inplace=True)

# Status
var_ict1110_demographics["Status"].fillna("", inplace=True)
```
# Sponsor
var_ict1110_demographics["Sponsor"].fillna("SELF-SPONSORED", inplace=True)

# Nationality
var_ict1110_demographics["Nationality"].fillna("MISSING DATA", inplace=True)

# Accommodated
var_ict1110_demographics["Accommodated"].fillna("MISSING DATA", inplace=True)

[63]: # Inspect dataframe to check for new values
var_ict1110_demographics.sample(2).T

[63]:
len(var_ict1110_demographics)

[63]:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentID</td>
<td>6cd50fb3091b0a9d3c1ac2cf52...</td>
<td>b5bcce260d9e303ca0e63f0551...</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>1999-03-03</td>
<td>1999-05-02</td>
</tr>
<tr>
<td>Gender</td>
<td>M</td>
<td>F</td>
</tr>
<tr>
<td>AcademicYear</td>
<td>20181</td>
<td>20191</td>
</tr>
<tr>
<td>YearOfStudy</td>
<td>1st Year</td>
<td>2nd Year</td>
</tr>
<tr>
<td>School</td>
<td>EDUCATION</td>
<td>EDUCATION</td>
</tr>
<tr>
<td>Program</td>
<td>BACHELOR OF INFORMATION AN...</td>
<td>BACHELOR OF INFORMATION AN...</td>
</tr>
<tr>
<td>MajorDescription</td>
<td>ICTs and Education</td>
<td>ICTs and Education</td>
</tr>
<tr>
<td>MinorDescription</td>
<td>MATHEMATICS</td>
<td>HISTORY</td>
</tr>
<tr>
<td>Status</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Sponsor</td>
<td>GRZ-FULLY SPONSORED</td>
<td>GRZ-FULLY SPONSORED</td>
</tr>
<tr>
<td>Nationality</td>
<td>ZAMBIAN</td>
<td>ZAMBIAN</td>
</tr>
<tr>
<td>Accommodated</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

[63]: 58

Exploratory Data Analysis

[64]: # Describe the data
# Identify attributes to explore
var_ict1110_demographics.head(2).T

[64]:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentID</td>
<td>9d5116a2451bc98c2b46b93acb...</td>
<td>e7400496f1ce70cb62c2c44ca2...</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>1998-09-14</td>
<td>2000-03-23</td>
</tr>
<tr>
<td>Gender</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>AcademicYear</td>
<td>20191</td>
<td>20191</td>
</tr>
<tr>
<td>YearOfStudy</td>
<td>2nd Year</td>
<td>2nd Year</td>
</tr>
<tr>
<td>School</td>
<td>EDUCATION</td>
<td>EDUCATION</td>
</tr>
<tr>
<td>Program</td>
<td>BACHELOR OF INFORMATION AN...</td>
<td>BACHELOR OF INFORMATION AN...</td>
</tr>
<tr>
<td>MajorDescription</td>
<td>ICTs and Education</td>
<td>ICTs and Education</td>
</tr>
<tr>
<td>MinorDescription</td>
<td>RELIGIOUS STUDIES</td>
<td>ART AND DESIGN STUDIES</td>
</tr>
</tbody>
</table>

Continued on next page
### Possible attributes to include in the EDA process

- DateOfBirth
- Gender
- MinorDescription
- Sponsor
- Accommodated

### Dataframe Statistical Information

```python
# Define variable to
var_ict1110_demographics_eda = var_ict1110_demographics

dataframe = var_ict1110_demographics_eda.describe(include='all').T
```

<table>
<thead>
<tr>
<th></th>
<th>count</th>
<th>unique</th>
<th>top</th>
<th>freq</th>
<th>mean</th>
<th>std</th>
<th>min</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentID</td>
<td>58</td>
<td>58</td>
<td>e2a96e074e1d8a6f6de56abbd4...</td>
<td>1</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>58</td>
<td>58</td>
<td>1998-09-14</td>
<td>1</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>58</td>
<td>2</td>
<td>M</td>
<td>30</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AcademicYear</td>
<td>58</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>20185.3</td>
<td>5.95659</td>
<td>20171</td>
<td>20181</td>
<td>20191</td>
<td>20191</td>
<td></td>
</tr>
<tr>
<td>YearOfStudy</td>
<td>58</td>
<td>2</td>
<td>2nd Year</td>
<td>30</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>School</td>
<td>58</td>
<td>1</td>
<td>EDUCATION</td>
<td>58</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>58</td>
<td>1</td>
<td>BACHELOR OF INFORMATION AN...</td>
<td>58</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MajorDescription</td>
<td>58</td>
<td>1</td>
<td>ICTs and Education</td>
<td>58</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
## Date of Birth

### Basic dataframe summaries

Get unique entries:
```python
var_ict1110_demographics_eda["DateOfBirth"].unique()
```

Count observations:
```python
var_ict1110_demographics_eda["DateOfBirth"].count()
```

Get value counts:
```python
var_ict1110_demographics_eda["DateOfBirth"].str[:4].value_counts()
```

---

### Example output

```python
       '1999-05-02', '1998-12-02', '1996-01-08', '2001-12-20',
       '2000-02-09', '1999-06-05', '1999-07-08', '1997-08-17',
       '2000-07-25', '1998-05-06'], dtype=object)
```

---

58

---

### Frequency distribution

<table>
<thead>
<tr>
<th>DateOfBirth</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>15</td>
</tr>
<tr>
<td>1998</td>
<td>13</td>
</tr>
<tr>
<td>2000</td>
<td>7</td>
</tr>
<tr>
<td>1997</td>
<td>7</td>
</tr>
<tr>
<td>2001</td>
<td>4</td>
</tr>
<tr>
<td>1995</td>
<td>3</td>
</tr>
<tr>
<td>1994</td>
<td>3</td>
</tr>
<tr>
<td>1996</td>
<td>2</td>
</tr>
<tr>
<td>1992</td>
<td>1</td>
</tr>
</tbody>
</table>

Continued on next page
# Plot Date of Birth barplot

# Check the date type
type(var_ict1110_demographics_eda["DateOfBirth"][str[:4]].value_counts())

#
var_ict1110_demographics_eda_dateofbirth = var_ict1110_demographics_eda["DateOfBirth"][str[:4]].value_counts()

#
##sns.barplot(x=var_ict1110_demographics_eda_dateofbirth.index, y=var_ict1110_demographics_eda_dateofbirth.values, palette='Spectral')
var_ict1110_demographics_eda_dateofbirth = var_ict1110_demographics_eda["DateOfBirth"][str[:4]].value_counts().plot(kind="barh")

pandas.core.series.Series
Gender

```python
# Basic dataframe summaries
# Get unique entries
var_ict1110_demographics_eda["Gender"].unique()

# Count observations
var_ict1110_demographics_eda["Gender"].count()

# Get value counts
var_ict1110_demographics_eda["Gender"].value_counts()
```

```python
array(['F', 'M'], dtype=object)
```

```plaintext
array(['F', 'M'], dtype=object)
```

```python
58
```

```python
M    30
F    28
```

```python
# Plot Gender pie chart
# Check the date type
type(var_ict1110_demographics_eda["Gender"].value_counts())

#
var_ict1110_demographics_eda_gender = var_ict1110_demographics_eda["Gender"].
value_counts()

#

```python
var_ict1110_demographics_eda["Gender"].value_counts().plot(kind="pie", title="All Students")
```

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

```python
<matplotlib.axes._subplots.AxesSubplot at 0x7f1549044eb8>
```

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

```python
<matplotlib.axes._subplots.AxesSubplot at 0x7f1549044eb8>
```
# Basic dataframe summaries

# Get unique entries
var_ict1110_demographics_eda["MinorDescription"].unique()

# Count observations
var_ict1110_demographics_eda["MinorDescription"].count()

# Get value counts
var_ict1110_demographics_eda["MinorDescription"].value_counts()

array(['RELIGIOUS STUDIES', 'ART AND DESIGN STUDIES', 'GEOGRAPHY',
       'MATHEMATICS', 'HISTORY', 'CIVIC EDUCATION', 'MISSING DATA',
       'FRENCH', 'ENGLISH'], dtype=object)

58

<table>
<thead>
<tr>
<th>MinorDescription</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MISSING DATA</td>
<td>12</td>
</tr>
<tr>
<td>MATHEMATICS</td>
<td>10</td>
</tr>
<tr>
<td>CIVIC EDUCATION</td>
<td>8</td>
</tr>
<tr>
<td>HISTORY</td>
<td>7</td>
</tr>
<tr>
<td>RELIGIOUS STUDIES</td>
<td>7</td>
</tr>
<tr>
<td>ENGLISH</td>
<td>6</td>
</tr>
<tr>
<td>GEOGRAPHY</td>
<td>4</td>
</tr>
</tbody>
</table>

Continued on next page
# Plot Minors using barplot

```python
var_ict1110_demographics_eda["MinorDescription"].value_counts().plot(kind="barh", title="Minor Programs")
```

Sponsor

```python
# Basic dataframe summaries
#
# Get unique entries
var_ict1110_demographics_eda["Sponsor"].unique()

# Count observations
var_ict1110_demographics_eda["Sponsor"].count()

# Get value counts
var_ict1110_demographics_eda["Sponsor"].value_counts()
```

```
array(['GRZ - 75 PERCENT', 'GRZ-FULLY SPONSORED', 'TUITION WAIVER(DEPENDANTS)', 'SELF-SPONSORED'], dtype=object)
```
<table>
<thead>
<tr>
<th>Sponsor</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRZ-FULLY SPONSORED</td>
<td>38</td>
</tr>
<tr>
<td>SELF-SPONSORED</td>
<td>8</td>
</tr>
<tr>
<td>GRZ - 75 PERCENT</td>
<td>6</td>
</tr>
<tr>
<td>TUITION WAIVER(DEPENDANTS)</td>
<td>6</td>
</tr>
</tbody>
</table>

```python
# Plot Sponsors using barplot
var_ict1110_demographics_eda['Sponsor'].value_counts().plot(kind="barh", title="Student Funding Status")
```

![Bar chart showing student funding status](image)

**Accommodated**

```python
# Basic dataframe summaries
# Get unique entries
var_ict1110_demographics_eda['Accommodated'].unique()

# Count observations
var_ict1110_demographics_eda['Accommodated'].count()

# Get value counts
var_ict1110_demographics_eda['Accommodated'].value_counts()
```
array(['Yes', 'No'], dtype=object)

58

<table>
<thead>
<tr>
<th>Accommodated</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>33</td>
</tr>
<tr>
<td>Yes</td>
<td>25</td>
</tr>
</tbody>
</table>

var_ict1110_demographics_eda.groupby("Accommodated")

# Plot Accommodation status using pie chart
#
# var_ict1110_demographics_eda["Accommodated"][value_counts()].plot(kind="pie", title="Student Accommodation Status")

Dataset #3: 2018/19 ICT 1110 Assessment Scores

Using the UNZA 2018/19 ICT 1110 Assessment Scores dataset, work towards the following: 1. Pre-process the datasets 2. Merge the datasets 3. Perform Exploratory Data Analysis on the merged dataset
Data Preprocessing

Dataset Description

Dataset Files

The files are categorised as follows:
* db-unza21-csc5741-ict1110_assessment_scores-quizN.csv * Weekly quiz scores graded out of 10 * There are a total of 20 data files: quiz 1–20
* db-unza21-csc5741-ict1110_assessment_scores-testN.csv * Test scores graded out of 50 * There are a total of four data files: tests 1–4
* db-unza21-csc5741-ict1110_assessment_scores-make-up_test.csv * Make-up assessment for students that never score minimum continuous assessment score required to write the final examination * The scores are out of 50 * There is one data file
* db-unza21-csc5741-ict1110_assessment_scores-final_examination.csv * Final Examination scores graded out of 100 * There is one data file

Dataset Format

- Assessment Scores Datasets
  - All data files have the same format; they are composed of two fields: StudentID (alphanumeric) and Mark (numeric)
  - The fields are pipe (“|”) seperated

```
# Datasets format
! cat -n db-unza21-csc5741-ict1110_assessment_scores-quiz20.csv | head

    1   Student ID|Mark
    2   287656464e1b658610335965b4d75a|9
    3   74459a3d3e5f3074226b1f9fa23c9163|9
    4   9d5116a2451bc98c2b46b93abc1b4f0|2
    5   #N/A|9
    6   94984a8c4896946d9bafed24959cb6181|10
    7   07f3ca235f4aa1c9ad16facef55268d|10
    8   e740096f1ce70cb62c2c44ca2ddc469|9
    9   e31959fe2842dace4d16d36e9813620|9
  10   575b9408b6daa2ddcbecbf681c9b4c|9
```

```
! ls -l db-unza21-csc5741-ict1110_assessment_scores-quiz* | wc -l
80
! ls -l db-unza21-csc5741-ict1110_assessment_scores-quiz*
```

20

-rw-r--r-- 1 lightonphiri lightonphiri 1949 May 30 16:08 db-unza21-csc5741-ict1110_assessment_scores-quiz01.csv
-rw-r--r-- 1 lightonphiri lightonphiri 2028 May 30 16:08 db-unza21-csc5741-ict1110_assessment_scores-quiz02.csv
-rw-r--r-- 1 lightonphiri lightonphiri 2009 May 30 16:08 db-unza21-csc5741-ict1110_assessment_scores-quiz03.csv
-rw-r--r-- 1 lightonphiri lightonphiri 1966 May 30 16:08 db-unza21-csc5741-ict1110_assessment_scores-quiz04.csv
-rw-r--r-- 1 lightonphiri lightonphiri 2104 May 30 16:08 db-unza21-csc5741-ict1110_assessment_scores-quiz05.csv
-rw-r--r-- 1 lightonphiri lightonphiri 2067 May 30 16:08 db-unza21-csc5741-ict1110_assessment_scores-quiz06.csv
-rw-r--r-- 1 lightonphiri lightonphiri 1920 May 30 16:08 db-unza21-csc5741-ict1110_assessment_scores-quiz07.csv
-rw-r--r-- 1 lightonphiri lightonphiri 1990 May 30 16:08 db-
unza21-csc5741-ict1110_assessment_scores-quiz08.csv
-rw-rw-- 1 lightonphiri lightonphiri 1986 May 30 16:08 db-
unza21-csc5741-ict1110_assessment_scores-quiz09.csv
-rw-rw-- 1 lightonphiri lightonphiri 1905 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz10.csv
-rw-rw-- 1 lightonphiri lightonphiri 1955 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz11.csv
-rw-rw-- 1 lightonphiri lightonphiri 1981 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz12.csv
-rw-rw-- 1 lightonphiri lightonphiri 1969 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz13.csv
-rw-rw-- 1 lightonphiri lightonphiri 1983 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz14.csv
-rw-rw-- 1 lightonphiri lightonphiri 1987 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz15.csv
-rw-rw-- 1 lightonphiri lightonphiri 1729 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz16.csv
-rw-rw-- 1 lightonphiri lightonphiri 1676 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz17.csv
-rw-rw-- 1 lightonphiri lightonphiri 1646 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz18.csv
-rw-rw-- 1 lightonphiri lightonphiri 1748 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz19.csv
-rw-rw-- 1 lightonphiri lightonphiri 1862 May 31 15:06 db-
unza21-csc5741-ict1110_assessment_scores-quiz20.csv

```
ls -l db-unza21-csc5741-ict1110_assessment_scores-test* | wc -l
ls -l db-unza21-csc5741-ict1110_assessment_scores-test*
```

```
4
```

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>07f3ca235faaa1c9ad16facef5526d8b</td>
<td>31.5</td>
</tr>
<tr>
<td>92185f5753932de762b780405a50bdf7</td>
<td>40</td>
</tr>
<tr>
<td>1bdb61bdf5bc9a2cdc5db037ab610e0</td>
<td>17</td>
</tr>
<tr>
<td>d2e4449b45608e33e472d939a73868f7</td>
<td>23.5</td>
</tr>
<tr>
<td>cee34f6bf4356c28fc2b766ae46b6d6c</td>
<td>16</td>
</tr>
<tr>
<td>d7fe73b6846dfe672236e636aa2cf6b5</td>
<td>13</td>
</tr>
<tr>
<td>8e4d9eeed250a9d065ac2bb8d67b30</td>
<td>23</td>
</tr>
<tr>
<td>bf2ec44b27fc25c6fd8a38792b9ef2a8</td>
<td>22</td>
</tr>
<tr>
<td>aa293d284f52b08da5ba7fe7792fe9c3</td>
<td>7</td>
</tr>
</tbody>
</table>
ls -l db-unza21-csc5741-ict1110_assessment_scores-final_examination* | wc -l
ls -l db-unza21-csc5741-ict1110_assessment_scores-final_examination*
cat db-unza21-csc5741-ict1110_assessment_scores-final_examination.csv | head

Dataframe Creation

Final Examination Scores

Create Dataframes

```python
# Create DataFrame of input dataset: ICT 1110 Demographics
#
var_ict1110_assessments_examination = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-final_examination.csv", sep="|")
var_ict1110_assessments_examination.columns
```

Rename Dataframe Names

```python
# Rename dataframe columns for easy processing
#
# Function use: pd.rename([...])
# Important parameter: inplace=True
#
var_ict1110_assessments_examination.rename(columns={
    "Student ID": "StudentID",
    "Total": "ExaminationScore"
}, inplace=True)
var_ict1110_assessments_examination.columns
```

```python
Index(['StudentID', 'ExaminationScore'], dtype='object')
```

```python
# Count records in dataframe
#
len(var_ict1110_assessments_examination)
```

1
-rw-rw-r-- 1 lightonphiri lightonphiri 2147 May 30 16:06 db-
unza21-csc5741-ict1110_assessment_scores-final_examination.csv

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>9d5116a2451bc98c2b46b93acbc1b4f0</td>
<td>46.5</td>
</tr>
<tr>
<td>e7400496f1ce70cb62c2c44ca2ddc469</td>
<td>48.5</td>
</tr>
<tr>
<td>cea34f6b4f356c28fc2b766ae46b6d6c</td>
<td>53</td>
</tr>
<tr>
<td>#N/A</td>
<td>90</td>
</tr>
<tr>
<td>6cd50fb3091b0a9d3c1ac2cf52441390</td>
<td>64.5</td>
</tr>
<tr>
<td>e31959fe2842dacea4d16d36e9813620</td>
<td>41</td>
</tr>
<tr>
<td>97527dec0ae1a703599581d4f25dffc5</td>
<td>5</td>
</tr>
<tr>
<td>9e7002d53d4db7bfad4f5cf419b0c126</td>
<td>51</td>
</tr>
<tr>
<td>74458a3d3e5f3074226b1f9fa23c9163</td>
<td>67</td>
</tr>
</tbody>
</table>
# Inspect some dataframe records
#
var_ict1110_assessments_examination.head(5)

<table>
<thead>
<tr>
<th>StudentID</th>
<th>ExaminationScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>9d5116a2451bc98c2b46b93acbc...</td>
<td>46.5</td>
</tr>
<tr>
<td>e7400496f1ce70cb62c2c44ca2...</td>
<td>48.5</td>
</tr>
<tr>
<td>cea3f6b4f356c28fc2b766ae4...</td>
<td>53.0</td>
</tr>
<tr>
<td>NaN</td>
<td>90.0</td>
</tr>
<tr>
<td>6cd50fb3091b0a9d3c1ac2cf52...</td>
<td>64.5</td>
</tr>
</tbody>
</table>

Drop Records With Null StudentID Values

# Delete all test score entries with null StudentID values
#
print("Examination Records Before Drop Operation: ", len(var_ict1110_assessments_examination))
var_ict1110_assessments_examination.dropna(subset = ["StudentID"], inplace=True)
print("Examination Records After Drop Operation: ", len(var_ict1110_assessments_examination))

Examination Records Before Drop Operation: 60
Examination Records After Drop Operation: 58

Handle Examination Score Null Values

All null score entries will be replaced by 0

var_ict1110_assessments_examination.fillna(0, inplace=True)

# Inspect dataframe
var_ict1110_assessments_examination.head(2).T

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentID</td>
<td>9d5116a2451bc98c2b46b93acbc...</td>
<td>e7400496f1ce70cb62c2c44ca2...</td>
</tr>
<tr>
<td>ExaminationScore</td>
<td>46.5</td>
<td>48.5</td>
</tr>
</tbody>
</table>

Test Scores

- There are a total of four(4) test data files. Each of these will be loaded into a separate dataframe.
- Test files will be subsequently merged into a single test scores dataframe

Create Dataframes

# Create DataFrame of input dataset: ICT 1110 Test Score Results
#
var_ict1110_assessments_test1 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-test1.csv", sep="|")
var_ict1110_assessments_test1.columns
#
var_ict1110_assessments_test2 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-test2.csv", sep="|")
var_ict1110_assessments_test2.columns
#
var_ict1110_assessments_test3 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-test3.csv", sep="|")
var_ict1110_assessments_test3.columns
#
var_ict1110_assessments_test4 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-test4.csv", sep="|")
var_ict1110_assessments_test4.columns

[90]: Index(["Student ID", 'Total'], dtype='object')
[90]: Index(["Student ID", 'Total'], dtype='object')
[90]: Index(["Student ID", 'Total'], dtype='object')
[90]: Index(["Student ID", 'Total'], dtype='object')

Rename Dataframe Names

[91]: # Rename dataframe columns for easy processing
    #
    # Function use: pd.rename([...])
    # Important parameter: inplace=True
    #
    var_ict1110_assessments_test1.rename(columns={
        "Student ID": "StudentID",
        "Total": "Test1Score"
    }, inplace=True)
    var_ict1110_assessments_test1.columns
#
    var_ict1110_assessments_test2.rename(columns={
        "Student ID": "StudentID",
        "Total": "Test2Score"
    }, inplace=True)
    var_ict1110_assessments_test2.columns
#
    var_ict1110_assessments_test3.rename(columns={
        "Student ID": "StudentID",
        "Total": "Test3Score"
    }, inplace=True)
```python
var_ict1110_assessments_test3.columns

#
var_ict1110_assessments_test4.rename(columns={
    "Student ID": "StudentID",
    "Total": "Test4Score"
}, inplace=True)

var_ict1110_assessments_test4.columns

[91]: Index(['StudentID', 'Test1Score'], dtype='object')

[91]: Index(['StudentID', 'Test2Score'], dtype='object')

[91]: Index(['StudentID', 'Test3Score'], dtype='object')

[91]: Index(['StudentID', 'Test4Score'], dtype='object')

[92]:
# Count records in dataframes
#
len(var_ict1110_assessments_test1)

#
len(var_ict1110_assessments_test2)

#
len(var_ict1110_assessments_test3)

#
len(var_ict1110_assessments_test4)

[92]: 61

[92]: 59

[92]: 59

[92]: 57

[93]:
# Inspect some dataframe records
#
var_ict1110_assessments_test1.head(5)

#
var_ict1110_assessments_test2.head(5)

#
var_ict1110_assessments_test3.head(5)
```

var ICT1110_assessments_test4.head(5)

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Test1Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>07f3ca235faaa1c9ad16facef5...</td>
<td>31.5</td>
</tr>
<tr>
<td>921855f753932de762b780405a...</td>
<td>40.0</td>
</tr>
<tr>
<td>1bd611bd65bc9a2cde5db037a...</td>
<td>17.0</td>
</tr>
<tr>
<td>d2e4449b45608e33e472d939a7...</td>
<td>23.5</td>
</tr>
<tr>
<td>cea34f6b4f356c28fc2b766ae4...</td>
<td>16.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Test1Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>e2a96e074e1bd8a6f6de56abbd4...</td>
<td>28.0</td>
</tr>
<tr>
<td>4234d1794dd33c1b6ed975eab5...</td>
<td>19.5</td>
</tr>
<tr>
<td>4be25f9d27da71d4e98775668b...</td>
<td>23.0</td>
</tr>
<tr>
<td>e99bc418c38da77213c6c6e03c...</td>
<td>15.5</td>
</tr>
<tr>
<td>b0aa0804e676a38255af4fd702...</td>
<td>15.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Test1Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>81975d05c61d8de83f46487739...</td>
<td>18</td>
</tr>
<tr>
<td>e99bb6b91e5f1db4ee9c340dc...</td>
<td>22</td>
</tr>
<tr>
<td>28765464efe1b6583610335965...</td>
<td>15</td>
</tr>
<tr>
<td>b2ce07e6bc0b55222aa66fccc1b...</td>
<td>18</td>
</tr>
<tr>
<td>NaN</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Test1Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>b2ce07e6bc0b55222aa66fccc1b...</td>
<td>17.0</td>
</tr>
<tr>
<td>afe1f0ece97fc8399c4822a67...</td>
<td>6.0</td>
</tr>
<tr>
<td>921855f753932de762b780405a...</td>
<td>22.0</td>
</tr>
<tr>
<td>2d65f5236205dd23c6a8212627...</td>
<td>13.5</td>
</tr>
<tr>
<td>b0aa0804e676a38255af4fd702...</td>
<td>22.5</td>
</tr>
</tbody>
</table>

Drop Records With Null StudentID Values

# Delete all test score entries with null StudentID values

```python
# Test #1 Records Before Drop Operation: ", len(var ICT1110_assessments_test1))
var ICT1110_assessments_test1.dropna(subset = ["StudentID"], inplace=True)
print("Test #1 Records After Drop Operation: ", len(var ICT1110_assessments_test1))
```

```python
# Test #2 Records Before Drop Operation: ", len(var ICT1110_assessments_test2))
var ICT1110_assessments_test2.dropna(subset = ["StudentID"], inplace=True)
print("Test #2 Records After Drop Operation: ", len(var ICT1110_assessments_test2))
```

```python
# Test #3 Records Before Drop Operation: ", len(var ICT1110_assessments_test3))
```
```python
var_ict1110_assessments_test3.dropna(subset = ['StudentID'], inplace=True)
print("Test #3 Records After Drop Operation: ", len(var_ict1110_assessments_test3))
#
print("Test #4 Records Before Drop Operation: ", len(var_ict1110_assessments_test4))
var_ict1110_assessments_test4.dropna(subset = ['StudentID'], inplace=True)
print("Test #4 Records After Drop Operation: ", len(var_ict1110_assessments_test4))
```

Test #1 Records Before Drop Operation: 61
Test #1 Records After Drop Operation: 59
Test #2 Records Before Drop Operation: 59
Test #2 Records After Drop Operation: 57
Test #3 Records Before Drop Operation: 59
Test #3 Records After Drop Operation: 57
Test #4 Records Before Drop Operation: 57
Test #4 Records After Drop Operation: 56

Merge All Tests

```python
from functools import reduce
# READ: https://stackoverflow.com/a/4438256/664424
# Outlines how to merge multiple datasets
#
# Create a list to be used to hold all the test dataframes
var_ict1110_assessments_test_dataframes = [var_ict1110_assessments_test1, var_ict1110_assessments_test2, var_ict1110_assessments_test3, var_ict1110_assessments_test4]

# Merge all the quizzes into one dataframe
var_ict1110_assessments_tests = reduce(lambda left, right: pd.merge(left, right, on='StudentID', how='outer'), var_ict1110_assessments_test_dataframes)
```

```python
# Inspect merged dataframe records
#
var_ict1110_assessments_tests.tail(2).T
```

```markdown
<table>
<thead>
<tr>
<th>StudentID</th>
<th>Test1Score</th>
<th>Test2Score</th>
<th>Test3Score</th>
<th>Test4Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>31</td>
<td>19.5</td>
<td>28</td>
<td>22</td>
</tr>
<tr>
<td>57</td>
<td>22.5</td>
<td>17</td>
<td>17</td>
<td>20.5</td>
</tr>
</tbody>
</table>
```

Handle Test Score Null Values
All null score entries will be replaced by 0

```python
var_ict1110_assessments_tests.fillna(0, inplace=True)
```

```python
# Inspect dataframe
var_ict1110_assessments_tests.head(2).T
```
<table>
<thead>
<tr>
<th>StudentID</th>
<th>07f3ca235faaa1c9ad16facef5...</th>
<th>921855f753932de762b780405a...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test1Score</td>
<td>31.5</td>
<td>40</td>
</tr>
<tr>
<td>Test2Score</td>
<td>29.5</td>
<td>16</td>
</tr>
<tr>
<td>Test3Score</td>
<td>34</td>
<td>24</td>
</tr>
<tr>
<td>Test4Score</td>
<td>29.5</td>
<td>22</td>
</tr>
</tbody>
</table>

Quiz Scores

Create Dataframes

```python
# Instead of manually creating 20 dataframes, this can be done dynamically using a dictionary

var_quiz_dataframes = {}  # dictionary to hold all quiz dataframes

for var_quiz in range(1, 21, 1):
    var_quiz_dataframe_key = "var_ict1110_assessments_quiz_"+str(var_quiz)
    if var_quiz < 10:
        var_quiz_dataframe_input_file = "db-unza21-csc5741-ict1110_assessment_scores-quiz0"+str(var_quiz)+".csv"
    else:
        var_quiz_dataframe_input_file = "db-unza21-csc5741-ict1110_assessment_scores-quiz"+str(var_quiz)+".csv"
    var_quiz_dataframes[var_quiz_dataframe_key] = pd.read_csv(var_quiz_dataframe_input_file, sep="|")
```

```python
var_quiz_dataframes["var_ict1110_assessments_quiz_20"].head(5)
```

```python
Student ID | Mark
-----------|------
0          | 9    
1          | 9    
2          | 2    
3          | 9    
4          | 10   
```

```python
# Create DataFrame of input dataset: ICT 1110 Quiz Score Results

var_ict1110_assessments_quiz1 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz01.csv", sep="|")
var_ict1110_assessments_quiz1.columns

var_ict1110_assessments_quiz2 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz02.csv", sep="|")
var_ict1110_assessments_quiz2.columns
```
var_ict1110_assessments_quiz3 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz03.csv", sep="|")
var_ict1110_assessments_quiz3.columns

var_ict1110_assessments_quiz4 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz04.csv", sep="|")
var_ict1110_assessments_quiz4.columns

var_ict1110_assessments_quiz5 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz05.csv", sep="|")
var_ict1110_assessments_quiz5.columns

var_ict1110_assessments_quiz6 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz06.csv", sep="|")
var_ict1110_assessments_quiz6.columns

var_ict1110_assessments_quiz7 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz07.csv", sep="|")
var_ict1110_assessments_quiz7.columns

var_ict1110_assessments_quiz8 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz08.csv", sep="|")
var_ict1110_assessments_quiz8.columns

var_ict1110_assessments_quiz9 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz09.csv", sep="|")
var_ict1110_assessments_quiz9.columns

var_ict1110_assessments_quiz10 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz10.csv", sep="|")
var_ict1110_assessments_quiz10.columns

var_ict1110_assessments_quiz11 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz11.csv", sep="|")
var_ict1110_assessments_quiz11.columns

var_ict1110_assessments_quiz12 = pd.read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz12.csv", sep="|")
var_ict1110_assessments_quiz12.columns
```python
# var_ict1110_assessments_quiz13 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz13.csv", sep="|")
var_ict1110_assessments_quiz13.columns

# var_ict1110_assessments_quiz14 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz14.csv", sep="|")
var_ict1110_assessments_quiz14.columns

# var_ict1110_assessments_quiz15 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz15.csv", sep="|")
var_ict1110_assessments_quiz15.columns

# var_ict1110_assessments_quiz16 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz16.csv", sep="|")
var_ict1110_assessments_quiz16.columns

# var_ict1110_assessments_quiz17 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz17.csv", sep="|")
var_ict1110_assessments_quiz17.columns

# var_ict1110_assessments_quiz18 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz18.csv", sep="|")
var_ict1110_assessments_quiz18.columns

# var_ict1110_assessments_quiz19 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz19.csv", sep="|")
var_ict1110_assessments_quiz19.columns

# var_ict1110_assessments_quiz20 = pd.
# read_csv("db-unza21-csc5741-ict1110_assessment_scores-quiz20.csv", sep="|")
var_ict1110_assessments_quiz20.columns

[101]: Index(['Student ID', 'Mark'], dtype='object')
[101]: Index(['Student ID', 'Mark'], dtype='object')
[101]: Index(['Student ID', 'Mark'], dtype='object')
[101]: Index(['Student ID', 'Mark'], dtype='object')
[101]: Index(['Student ID', 'Mark'], dtype='object')
[101]: Index(['Student ID', 'Mark'], dtype='object')
```
Rename Dataframe Names

```python
# Loop through all quizzes and print first two records
for var_quiz in range(1, 21, 1):
    var_dataframe_key = "var_ict1110_assessments_quiz_" + str(var_quiz)
    var_quiz_dataframes[var_dataframe_key].rename(columns={'Student ID': 'StudentID', 'Mark': 'Quiz' + str(var_quiz) + 'Score'}, inplace=True)
```

```
var_quiz_dataframes["var_ict1110_assessments_quiz_13"].head(5)
```

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Quiz13Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>e99bb6b91ef51dbe4eec9340dc...</td>
<td>1.0</td>
</tr>
<tr>
<td>cea34f6b4f356c28fc2b766ae4...</td>
<td>1.0</td>
</tr>
<tr>
<td>1bdb61bdbf5bc9a2cdc5db037a...</td>
<td>0.5</td>
</tr>
<tr>
<td>b14c3890a187a7798035ac60d8...</td>
<td>0.5</td>
</tr>
</tbody>
</table>
```
Continued on next page
```
# Rename dataframe columns for easy processing

# Function use: pd.rename([...])
# Important parameter: inplace=True

var_ict1110_assessments_quiz1.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz1Score"
}, inplace=True)

var_ict1110_assessments_quiz1.columns

#

var_ict1110_assessments_quiz2.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz2Score"
}, inplace=True)

var_ict1110_assessments_quiz2.columns

#

var_ict1110_assessments_quiz3.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz3Score"
}, inplace=True)

var_ict1110_assessments_quiz3.columns

#

var_ict1110_assessments_quiz4.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz4Score"
}, inplace=True)

var_ict1110_assessments_quiz4.columns

#

var_ict1110_assessments_quiz5.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz5Score"
}, inplace=True)

var_ict1110_assessments_quiz5.columns

#

var_ict1110_assessments_quiz6.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz6Score"
}, inplace=True)

var_ict1110_assessments_quiz6.columns

---

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Quiz13Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>9c05c40b81fd906b1585e231c0...</td>
</tr>
</tbody>
</table>
"Mark": "Quiz6Score"
}, inplace=True)

var_ict1110_assessments_quiz6.columns
#
var_ict1110_assessments_quiz7.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz7Score"
}, inplace=True)

var_ict1110_assessments_quiz7.columns
#
var_ict1110_assessments_quiz8.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz8Score"
}, inplace=True)

var_ict1110_assessments_quiz8.columns
#
var_ict1110_assessments_quiz9.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz9Score"
}, inplace=True)

var_ict1110_assessments_quiz9.columns
#
var_ict1110_assessments_quiz10.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz10Score"
}, inplace=True)

var_ict1110_assessments_quiz10.columns
#
var_ict1110_assessments_quiz11.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz11Score"
}, inplace=True)

var_ict1110_assessments_quiz11.columns
#
var_ict1110_assessments_quiz12.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz12Score"
}, inplace=True)
# var_ict1110_assessments_quiz12.columns

# var_ict1110_assessments_quiz13.rename(columns={
    # "Student ID": "StudentID",
    # "Mark": "Quiz13Score"
}, inplace=True)

# var_ict1110_assessments_quiz13.columns

# var_ict1110_assessments_quiz14.rename(columns={
    # "Student ID": "StudentID",
    # "Mark": "Quiz14Score"
}, inplace=True)

# var_ict1110_assessments_quiz14.columns

# var_ict1110_assessments_quiz15.rename(columns={
    # "Student ID": "StudentID",
    # "Mark": "Quiz15Score"
}, inplace=True)

# var_ict1110_assessments_quiz15.columns

# var_ict1110_assessments_quiz16.rename(columns={
    # "Student ID": "StudentID",
    # "Mark": "Quiz16Score"
}, inplace=True)

# var_ict1110_assessments_quiz16.columns

# var_ict1110_assessments_quiz17.rename(columns={
    # "Student ID": "StudentID",
    # "Mark": "Quiz17Score"
}, inplace=True)

# var_ict1110_assessments_quiz17.columns

# var_ict1110_assessments_quiz18.rename(columns={
    # "Student ID": "StudentID",
    # "Mark": "Quiz18Score"
}, inplace=True)

# var_ict1110_assessments_quiz18.columns
var_ict1110_assessments_quiz19.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz19Score"
}, inplace=True)

var_ict1110_assessments_quiz19.columns

#

var_ict1110_assessments_quiz20.rename(columns={
    "Student ID": "StudentID",
    "Mark": "Quiz20Score"
}, inplace=True)

var_ict1110_assessments_quiz20.columns

[104]: Index(['StudentID', 'Quiz1Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz2Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz3Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz4Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz5Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz6Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz7Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz8Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz9Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz10Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz11Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz12Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz13Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz14Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz15Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz16Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz17Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz18Score'], dtype='object')
[104]: Index(['StudentID', 'Quiz19Score'], dtype='object')
Drop Records With Null StudentID Values

```python
# Delete all quiz entries with null StudentID values
for var_quiz in range(1, 21, 1):
    # dynamically create key for accessing dictionary value
    var_dataframe_key = "var_ict1110_assessments_quiz_" + str(var_quiz)
    var_quiz_dataframes[var_dataframe_key].dropna(subset = ['StudentID'], inplace=True)
```

Merge All Quizzes

```python
from functools import reduce

# READ: https://stackoverflow.com/a/44338256/664424
# Outlines how to merge multiple datasets

# Create a list to be used to hold all the quiz dataframes
var_ict1110_assessments_quiz_dataframes = list(var_quiz_dataframes.values())

# Merge all the quizzes into one dataframe
var_ict1110_assessments_quizzes = reduce(lambda left,right: pd.
    merge(left,right,on=['StudentID'],
    how='outer'),
    var_ict1110_assessments_quiz_dataframes)
```

```sql
# Inspect merged dataframe records
var_ict1110_assessments_quizzes.tail(2).T
```

<table>
<thead>
<tr>
<th></th>
<th>62</th>
<th>63</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentID</td>
<td>aa293d284f52b08da5ba7fe779...</td>
<td>c89bc418c38da77213c6c6e03c...</td>
</tr>
<tr>
<td>Quiz1Score</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>Quiz2Score</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>Quiz3Score</td>
<td>0</td>
<td>NaN</td>
</tr>
<tr>
<td>Quiz4Score</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Quiz5Score</td>
<td>2.5</td>
<td>6</td>
</tr>
<tr>
<td>Quiz6Score</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Quiz7Score</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Quiz8Score</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Quiz9Score</td>
<td>NaN</td>
<td>4</td>
</tr>
<tr>
<td>Quiz10Score</td>
<td>NaN</td>
<td>3.5</td>
</tr>
<tr>
<td>Quiz11Score</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Quiz12Score</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Quiz13Score</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Quiz14Score</td>
<td>NaN</td>
<td>3</td>
</tr>
<tr>
<td>Quiz15Score</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>Quiz16Score</td>
<td>NaN</td>
<td>1.5</td>
</tr>
<tr>
<td>Quiz17Score</td>
<td>NaN</td>
<td>10</td>
</tr>
<tr>
<td>Quiz18Score</td>
<td>NaN</td>
<td>1</td>
</tr>
</tbody>
</table>
Handle Quiz Score Null Values

All null score entries will be replaced by 0

```
[108]: var_ict1110_assessments_quizzes.fillna(0, inplace=True)
```

```
[109]: # Inspect dataframe
    var_ict1110_assessments_quizzes.head(2).T
```

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentID</td>
<td>53b3c88ea00c4f0e137b4e6fe7...</td>
</tr>
<tr>
<td>Quiz1Score</td>
<td>1</td>
</tr>
<tr>
<td>Quiz2Score</td>
<td>5</td>
</tr>
<tr>
<td>Quiz3Score</td>
<td>3</td>
</tr>
<tr>
<td>Quiz4Score</td>
<td>4</td>
</tr>
<tr>
<td>Quiz5Score</td>
<td>5.5</td>
</tr>
<tr>
<td>Quiz6Score</td>
<td>6</td>
</tr>
<tr>
<td>Quiz7Score</td>
<td>1</td>
</tr>
<tr>
<td>Quiz8Score</td>
<td>0</td>
</tr>
<tr>
<td>Quiz9Score</td>
<td>4</td>
</tr>
<tr>
<td>Quiz10Score</td>
<td>2</td>
</tr>
<tr>
<td>Quiz11Score</td>
<td>4</td>
</tr>
<tr>
<td>Quiz12Score</td>
<td>10</td>
</tr>
<tr>
<td>Quiz13Score</td>
<td>0</td>
</tr>
<tr>
<td>Quiz14Score</td>
<td>4</td>
</tr>
<tr>
<td>Quiz15Score</td>
<td>9</td>
</tr>
<tr>
<td>Quiz16Score</td>
<td>0</td>
</tr>
<tr>
<td>Quiz17Score</td>
<td>0</td>
</tr>
<tr>
<td>Quiz18Score</td>
<td>0</td>
</tr>
<tr>
<td>Quiz19Score</td>
<td>9</td>
</tr>
<tr>
<td>Quiz20Score</td>
<td>7</td>
</tr>
</tbody>
</table>

Dataset Attributes

- StudentID—Alphanumeric
- QuizNScore—Numeric
- TestNScore—Numeric
- ExaminationScore—Numeric

Data Pre-processing Plan

- STEP 1: Remove all records with NULL StudentID values
- STEP 2: Remove duplicates using StudentID as unique field

NOTE: Null values to be handled on a case-by-case basis; e.g. null values in text attributes to be replaced with empty strings ""
Exploratory Data Analysis

Examination Scores

```python
sns.distplot(var_ict1110_assessments_examination["ExaminationScore"]).set(xlim=(0, 100))
```

Test Scores

```python
sns.distplot(var_ict1110_assessments_examination["ExaminationScore"]).set(xlim=(0, 100))
```
# Facet results by academic year

```python
fig, (ax1, ax2, ax3, ax4) = plt.subplots(ncols=4, nrows=1, figsize=(15, 5))

fig.suptitle('Test Scores Distributions')

sns.distplot(var_ict1110_assessments_tests['Test1Score'], ax=ax1).set(xlim=(0, 50))
sns.distplot(var_ict1110_assessments_tests['Test2Score'], ax=ax2).set(xlim=(0, 50))
sns.distplot(var_ict1110_assessments_tests['Test3Score'], ax=ax3).set(xlim=(0, 50))
sns.distplot(var_ict1110_assessments_tests['Test4Score'], ax=ax4).set(xlim=(0, 50))
```

[112]: Text(0.5, 0.98, 'Test Scores Distributions')

[112]: [(0, 50)]

[112]: [(0, 50)]

[112]: [(0, 50)]

[112]: [(0, 50)]
Quiz Scores

```python
sns.distplot(var_ict110_assessments_quizzes['Quiz2Score']).set(xlim=(0, 10))
```

# Facet results by academic year

```python
fig, ((ax1, ax2, ax3, ax4), (ax5, ax6, ax7, ax8), (ax9, ax10, ax11, ax12), (ax13, ax14, ax15, ax16), (ax17, ax18, ax19, ax20)) = plt.subplots(ncols=4, nrows=5, figsize=(15,20))
```
fig.suptitle('Quiz Scores Distributions')

sns.distplot(var_ict1110_assessments_quizzes['Quiz1Score'], ax=ax1).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz2Score'], ax=ax2).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz3Score'], ax=ax3).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz4Score'], ax=ax4).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz5Score'], ax=ax5).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz6Score'], ax=ax6).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz7Score'], ax=ax7).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz8Score'], ax=ax8).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz9Score'], ax=ax9).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz10Score'], ax=ax10).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz11Score'], ax=ax11).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz12Score'], ax=ax12).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz13Score'], ax=ax13).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz14Score'], ax=ax14).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz15Score'], ax=ax15).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz16Score'], ax=ax16).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz17Score'], ax=ax17).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz18Score'], ax=ax18).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz19Score'], ax=ax19).set(xlim=(0, 10))
sns.distplot(var_ict1110_assessments_quizzes['Quiz20Score'], ax=ax20).set(xlim=(0, 10))
Quiz Scores Distributions

![Histograms of Quiz Scores](image-url)
```python
var_ict1110_assessments_quizzes = var_ict1110_assessments_quizzes.melt('StudentID',
                         var_name='cols', value_name='vals')
```

```python
StudentID cols vals
0 53b3c88ea00c4f0e137b4e6fe7... Quiz1Score 1.0
1 c03b1123e45fa00da3142e0424... Quiz1Score 2.0
2 74458a3d3e5f3074226b1f9fa2... Quiz1Score 2.0
3 e2a96e074e1d8a6f6de56abbd4... Quiz1Score 1.0
4 e2a96e074e1d8a6f6de56abbd4... Quiz1Score 1.0
```

```python
from scipy.stats import norm
# READ: https://seaborn.pydata.org/generated/seaborn.distplot.html
#
g = sns.FacetGrid(var_ict1110_assessments_quizzes_, col="cols", col_wrap=3)
g = g.map(sns.distplot, "vals", fit=norm, kde=False, hist=True)
```
Lightweight Pipelining With JobLib

There are usually a number of pipelining steps involved when working on data mining problems. Libraries like joblib make it possible for you to save the state of models. joblib.dump() and joblib.load() provide a replacement for pickle to work efficiently on arbitrary Python objects containing large data, in particular large numpy arrays. https://joblib.readthedocs.io/en/latest/persistence.html

In this instance, we will save the dataframes associated with all the data sources

- Please also READ: https://stackoverflow.com/a/12617603/664424

Save Initial Survey Dataframes

```python
# Import joblib
import joblib

# Initial Survey
joblib.dump(var_ict1110_survey_eda, "var_ict1110_survey_eda_dataframe.pkl")
```

[118]: ['var_ict1110_survey_eda_dataframe.pkl']

Save Demographic Dataframes

```python
# Import joblib
import joblib

# Demographic Details
joblib.dump(var_ict1110_demographics_eda, "var_ict1110_demographics_eda_dataframe.pkl")
```

[119]: ['var_ict1110_demographics_eda_dataframe.pkl']

Save Assessments Dataframes

```python
# Import joblib
# import joblib

# Quizzes
joblib.dump(var_ict1110_assessments_quizzes, "var_ict1110_assessments_quizzes_dataframe.pkl")

# Tests
joblib.dump(var_ict1110_assessments_tests, "var_ict1110_assessments_tests_dataframe.pkl")

# Examination
```
joblib.dump(var_ict1110_assessments_examination, "var_ict1110_assessments_examination_dataframe.pkl")

[120]: ['var_ict1110_assessments_quizzes_dataframe.pkl']
[120]: ['var_ict1110_assessments_tests_dataframe.pkl']
[120]: ['var_ict1110_assessments_examination_dataframe.pkl']

[121]: var_example_saved_state = joblib.load("var_ict1110_assessments_examination_dataframe.pkl")

[122]: var_example_saved_state.columns
[122]: Index(["StudentID", "ExaminationScore"], dtype='object')

[122]: 58