MDSA D206 Data Cleaning

Justin Jordan

3/2/2024

WGU

Part I: Research Question

A. Description of Question

What factors, if any, can be used to predict a customers churn?

B. Description of all variables within data set with data type and example.

In order to look at all vairables we first need to setup our environment. First we will import any packages needed.

In [ ]:
```
# package imports
import numpy as np
import pandas as pd
import missingno as msno
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn import preprocessing
import seaborn as sns
import scipy.stats as stats
import random
```

Data Frame Setup

We then setup our data frame from the data set provided using the pd.read_csv function from pandas.

In [ ]:
```
# code for data frame
churn = pd.read_csv("C:/users/jjord/Documents/WGU/D206/PA/churn_raw_data.csv")
```

After we have imported our CSV into our data frame, we can then use the .info() function to get our variable info. This info will include variable name, counts and data types.

In [ ]:
```
# Profile of Data Frame
churn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 52 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            10000 non-null  int64  
 1   CaseOrder             10000 non-null  int64  
 2   Customer_id           10000 non-null  object 
 3   Interaction           10000 non-null  object 
 4   City                  10000 non-null  object 
 5   State                 10000 non-null  object 
 6   County                10000 non-null  object 
 7   Zip                   10000 non-null  int64  
 8   Lat                   10000 non-null  float64
 9   Lng                   10000 non-null  float64
 10  Population            10000 non-null  int64  
 11  Area                  10000 non-null  object 
 12  Timezone              10000 non-null  object 
 13  Job                   10000 non-null  object 
 14  Children              7505 non-null   float64
 15  Age                   7525 non-null   float64
 16  Education             10000 non-null  object 
 17  Employment            10000 non-null  object 
 18  Income                7510 non-null   float64
 19  Marital               10000 non-null  object 
 20  Gender                10000 non-null  object 
 21  Churn                 10000 non-null  object 
 22  Outage_sec_perweek    10000 non-null  float64
 23  Email                 10000 non-null  int64  
 24  Contacts              10000 non-null  int64  
 25  Yearly_equip_failure  10000 non-null  int64  
 26  Techie                7523 non-null   object 
 27  Contract              10000 non-null  object 
 28  Port_modem            10000 non-null  object 
 29  Tablet                10000 non-null  object 
 30  InternetService       7871 non-null   object 
 31  Phone                 8974 non-null   object 
 32  Multiple              10000 non-null  object 
 33  OnlineSecurity        10000 non-null  object 
 34  OnlineBackup          10000 non-null  object 
 35  DeviceProtection      10000 non-null  object 
 36  TechSupport           9009 non-null   object 
 37  StreamingTV           10000 non-null  object 
 38  StreamingMovies       10000 non-null  object 
 39  PaperlessBilling      10000 non-null  object 
 40  PaymentMethod         10000 non-null  object 
 41  Tenure                9069 non-null   float64
 42  MonthlyCharge         10000 non-null  float64
 43  Bandwidth_GB_Year     8979 non-null   float64
 44  item1                 10000 non-null  int64  
 45  item2                 10000 non-null  int64  
 46  item3                 10000 non-null  int64  
 47  item4                 10000 non-null  int64  
 48  item5                 10000 non-null  int64  
 49  item6                 10000 non-null  int64  
 50  item7                 10000 non-null  int64  
```

```
 51  item8                 10000 non-null  int64
dtypes: float64(9), int64(15), object(28)
memory usage: 4.0+ MB
```

Part II: Data-Cleaning Plan

C. Explanation of Plan for Data Cleaning

1. Python will used to asses the quality of the data within the data set. The data set will first be checked for duplicate records. If any dulpicates are found using the .duplicated() function, they will be removed from the data set. Once we have dealt with any duplicates, we will move on to identifiying any missing data within the data set. In order to determine if there are any missing values, we will use the .isnull().sum() function. This function will return the column names within the data set as well as a count of any missing values for each column. Once we have determined any missing values, we will use the msno.matrix() function to visualize the missing values. We will then visualize each column that has any missing values individually. To do this, we will use the plt.hist() function. This function will create a histogram so we can better examine the missing values. This method can only be used on quantitative data types. For qualitative data types with missing values, we will use the value_counts() function.

Once we have identified any missing values, we can then determine the proper way to treat them. Treatment methods that will be considered are deletion and imputation. Deletion includes removing either variables(columns) or observations(rows). Deletion will only be used if a variable is missing beyond 60% of its data or an observation is 5% or less. (Middleton, 2023) Imputation includes using Univariate statistical imputation, backward/forward fill, multiple imputation by chained equations or K-nearest neighbor. (Middleton, 2023) Imputation will replace a missing value with a substitute value from the data set. Once we visualize any missing data, we can then determine which method we will use to treat the missing data.

After missing values have been treated, we can then look for any outliers within the data frame. To visualize any outliers we will be using Seaborn and the .boxplot() function. Once we have determined if any outliers exsist, we can treat them. Treatment methods include imputation, retention, exclusion and removal. Each method has their advantages and disadvantages and will be taken into consideration when examining any outliers. Outliers will only be detected on numeric data types.

Finally, PCA will be performed on any numerical variables within the data set.

2. CLeaning data is the most important part of data analysis. When using the methods above to clean the data, we can ensure that our analysis and conclusion of the data are accurate. Missing values and outliers can greatly effect outcomes and possibly lead to misinfomation given to the business. By using different techniques and visualizations,

we can determine the best way to clean the data so the business has an accurate representation. Multiple treatment methods are mentioned above but not all will be used. Once we get a better understanding of the data at hand, we can then choose the best method for treatment.

3. We will be using Python as the programming language for the data cleaning process. Python is a widely used language in data analysis. JupyterNotebook will be used as well. It will allow us to combine the use of mardown and code cells into one file. We will also be using certain libraries within Python. Those libraries include Pandas, Numpy, Matplotlib, Missingno, Seaborn and sklearn. Pandas will be used to import the data set into a data frame. Numpy will be used for any array's needed. Matplotlib and Seaborn will be used for visualization. Missingno will be used for detecting missing values and visualization. Sklearn will be used for the PCA.

4. The following section will include the code with comments of what the code does and well as the output of the code.

In [ ]:
```
with pd.option_context("display.max_columns", 40):
    print(churn.describe(include=[np.number]))
```

```
        Unnamed: 0    CaseOrder           Zip           Lat           Lng  \
count  10000.00000  10000.00000  10000.000000  10000.000000  10000.000000   
mean    5000.50000   5000.50000  49153.319600     38.757567    -90.782536   
std     2886.89568   2886.89568  27532.196108      5.437389     15.156142   
min        1.00000      1.00000    601.000000     17.966120   -171.688150   
25%     2500.75000   2500.75000  26292.500000     35.341828    -97.082812   
50%     5000.50000   5000.50000  48869.500000     39.395800    -87.918800   
75%     7500.25000   7500.25000  71866.500000     42.106908    -80.088745   
max    10000.00000  10000.00000  99929.000000     70.640660    -65.667850   

          Population     Children          Age         Income  \
count   10000.000000  7505.000000  7525.000000    7510.000000   
mean     9756.562400     2.095936    53.275748   39936.762226   
std     14432.698671     2.154758    20.753928   28358.469482   
min         0.000000     0.000000    18.000000     740.660000   
25%       738.000000     0.000000    35.000000   19285.522500   
50%      2910.500000     1.000000    53.000000   33186.785000   
75%     13168.000000     3.000000    71.000000   53472.395000   
max    111850.000000    10.000000    89.000000  258900.700000   

       Outage_sec_perweek         Email      Contacts  Yearly_equip_failure  \
count        10000.000000  10000.000000  10000.000000          10000.000000   
mean            11.452955     12.016000      0.994200              0.398000   
std              7.025921      3.025898      0.988466              0.635953   
min             -1.348571      1.000000      0.000000              0.000000   
25%              8.054362     10.000000      0.000000              0.000000   
50%             10.202896     12.000000      1.000000              0.000000   
75%             12.487644     14.000000      2.000000              1.000000   
max             47.049280     23.000000      7.000000              6.000000   

            Tenure  MonthlyCharge  Bandwidth_GB_Year         item1  \
count  9069.000000   10000.000000        8979.000000  10000.000000   
mean     34.498858     174.076305        3398.842752      3.490800   
std      26.438904      43.335473        2187.396807      1.037797   
min       1.000259      77.505230         155.506715      1.000000   
25%       7.890442     141.071078        1234.110529      3.000000   
50%      36.196030     169.915400        3382.424000      3.000000   
75%      61.426670     203.777441        5587.096500      4.000000   
max      71.999280     315.878600        7158.982000      7.000000   

              item2         item3         item4         item5         item6  \
count  10000.000000  10000.000000  10000.000000  10000.000000  10000.000000   
mean       3.505100      3.487000      3.497500      3.492900      3.497300   
std        1.034641      1.027977      1.025816      1.024819      1.033586   
min        1.000000      1.000000      1.000000      1.000000      1.000000   
25%        3.000000      3.000000      3.000000      3.000000      3.000000   
50%        4.000000      3.000000      3.000000      3.000000      3.000000   
75%        4.000000      4.000000      4.000000      4.000000      4.000000   
max        7.000000      8.000000      7.000000      7.000000      8.000000   

              item7         item8  
count  10000.000000  10000.000000  
mean       3.509500      3.495600  
std        1.028502      1.028633  
min        1.000000      1.000000  
25%        3.000000      3.000000  
```

```
50%        4.000000      3.000000
75%        4.000000      4.000000
max        7.000000      8.000000
```

In [ ]:
```
# Duplicate Detection
# False means no duplicate
churn.duplicated()
```

Out[ ]:
```
0       False
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Length: 10000, dtype: bool
```

In [ ]:
```
# how many duplicates
print(churn.duplicated().value_counts())
```

```
False    10000
Name: count, dtype: int64
```

We can see from the above code that no duplicates were found within the data.

Next we will look at any missing values.

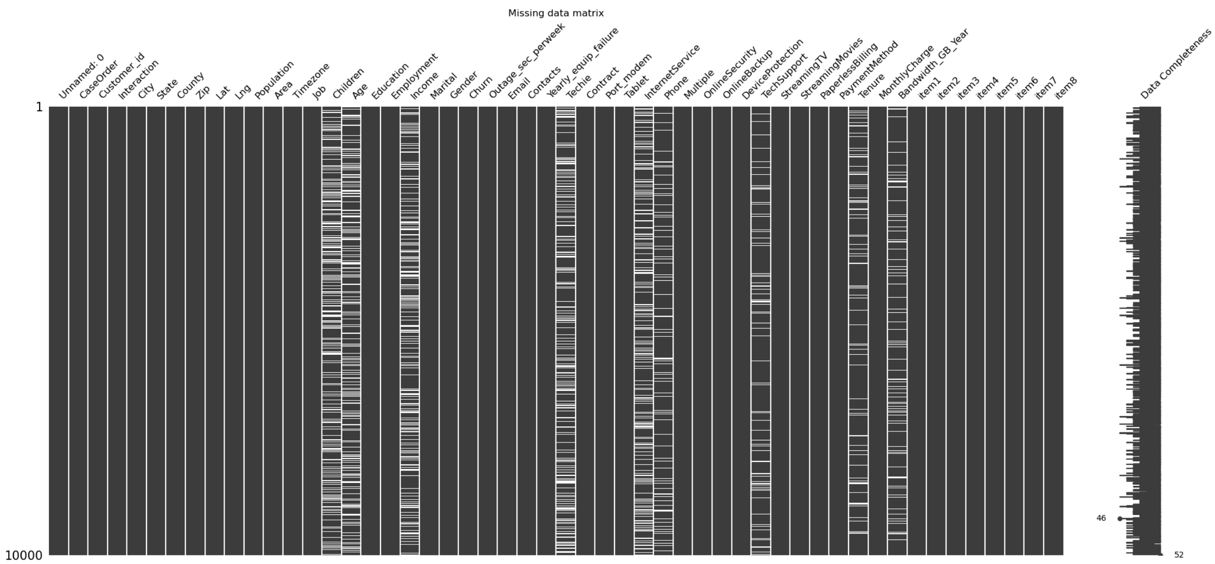
In [ ]:
```
# Missing values using isnull
churn.isnull().sum()
```

Out[ ]:
```
Unnamed: 0                 0
CaseOrder                  0
Customer_id                0
Interaction                0
City                       0
State                      0
County                     0
Zip                        0
Lat                        0
Lng                        0
Population                 0
Area                       0
Timezone                   0
Job                        0
Children                2495
Age                     2475
Education                  0
Employment                 0
Income                  2490
Marital                    0
Gender                     0
Churn                      0
Outage_sec_perweek         0
Email                      0
Contacts                   0
Yearly_equip_failure       0
Techie                  2477
Contract                   0
Port_modem                 0
Tablet                     0
InternetService         2129
Phone                   1026
Multiple                   0
OnlineSecurity             0
OnlineBackup               0
DeviceProtection           0
TechSupport              991
StreamingTV                0
StreamingMovies            0
PaperlessBilling           0
PaymentMethod              0
Tenure                   931
MonthlyCharge              0
Bandwidth_GB_Year       1021
item1                      0
item2                      0
item3                      0
item4                      0
item5                      0
item6                      0
item7                      0
item8                      0
dtype: int64
```

The above code shows missing values detected in the following columns: Children, Age, Income, Techie, InternetService, Phone, TechSupport, Tenure and Bandwidth_GB_Year

We will now use missingno to visualize each column that contains missing values.

In [ ]:
```
# Visulazation of missing values using missingno
msno.matrix(churn, fontsize=12, labels=True)
plt.title("Missing data matrix")
plt.show()
```



We will now look at unquie values within each column

In [ ]:
```
for col in churn:
    print(col)
    print(churn[col].unique())
```

```
Unnamed: 0
[    1     2     3 ...  9998  9999 10000]
CaseOrder
[    1     2     3 ...  9998  9999 10000]
Customer_id
['K409198' 'S120509' 'K191035' ... 'I243405' 'I641617' 'T38070']
Interaction
['aa90260b-4141-4a24-8e36-b04ce1f4f77b'
 'fb76459f-c047-4a9d-8af9-e0f7d4ac2524'
 '344d114c-3736-4be5-98f7-c72c281e2d35' ...
 'e8307ddf-9a01-4fff-bc59-4742e03fd24f'
 '3775ccfc-0052-4107-81ae-9657f81ecdf3'
 '9de5fb6e-bd33-4995-aec8-f01d0172a499']
City
['Point Baker' 'West Branch' 'Yamhill' ... 'Haubstadt' 'West Kill'
 'Clarkesville']
State
['AK' 'MI' 'OR' 'CA' 'TX' 'GA' 'TN' 'OK' 'FL' 'OH' 'PA' 'PR' 'IA' 'ME'
 'IL' 'WI' 'NC' 'AL' 'NM' 'VT' 'MD' 'NY' 'WA' 'CT' 'NJ' 'DC' 'ND' 'LA'
 'NE' 'WV' 'AZ' 'MO' 'WY' 'MT' 'VA' 'KY' 'MN' 'KS' 'MA' 'IN' 'SC' 'NH'
 'DE' 'MS' 'ID' 'AR' 'SD' 'CO' 'HI' 'UT' 'RI' 'NV']
County
['Prince of Wales-Hyder' 'Ogemaw' 'Yamhill' ... 'Billings' 'Haralson'
 'Briscoe']
Zip
[99927 48661 97148 ...  5758 37042 30523]
Lat
[56.251   44.32893 45.35589 ... 43.43391 36.56907 34.70783]
Lng
[-133.37571  -84.2408  -123.24657 ...  -72.78734  -87.41694  -83.53648]
Population
[   38 10446  3735 ...  4261 77168 12230]
Area
['Urban' 'Suburban' 'Rural']
Timezone
['America/Sitka' 'America/Detroit' 'America/Los_Angeles' 'America/Chicago'
 'America/New_York' 'America/Puerto_Rico' 'America/Denver'
 'America/Menominee' 'America/Phoenix' 'America/Indiana/Indianapolis'
 'America/Boise' 'America/Kentucky/Louisville' 'Pacific/Honolulu'
 'America/Indiana/Petersburg' 'America/Nome' 'America/Anchorage'
 'America/Indiana/Knox' 'America/Juneau' 'America/Toronto'
 'America/Indiana/Winamac' 'America/Indiana/Vincennes'
 'America/North_Dakota/New_Salem' 'America/Indiana/Tell_City'
 'America/Indiana/Marengo' 'America/Ojinaga']
Job
['Environmental health practitioner' 'Programmer, multimedia'
 'Chief Financial Officer' 'Solicitor' 'Medical illustrator'
 'Chief Technology Officer' 'Surveyor, hydrographic'
 'Sales promotion account executive' 'Teaching laboratory technician'
 'Museum education officer' 'Teacher, special educational needs'
 'Maintenance engineer' 'Engineer, broadcasting (operations)'
 'Learning disability nurse' 'Automotive engineer'
 'Amenity horticulturist' 'Applications developer' 'Immunologist'
 'Engineer, electrical' 'Broadcast presenter' 'Counsellor'
 'Geophysical data processor' 'Designer, multimedia' 'Event organiser'
 'Equality and diversity officer' 'Psychiatrist'
```

```
 'Surveyor, commercial/residential' 'Civil Service administrator'
 'Radiographer, diagnostic' 'Air traffic controller' 'Dietitian'
 'Therapist, occupational' 'Building services engineer'
 'Information officer' 'Outdoor activities/education manager'
 'Market researcher' 'Surveyor, insurance' 'Office manager'
 'Editorial assistant' 'Customer service manager'
 'Production designer, theatre/television/film' 'Analytical chemist'
 'Print production planner' 'Conservation officer, nature'
 'Librarian, public' 'Financial adviser' 'Surveyor, building'
 'Horticulturist, amenity' 'Diagnostic radiographer'
 'Doctor, general practice' 'Insurance risk surveyor' 'Heritage manager'
 'Legal executive' 'Professor Emeritus' 'Radio producer'
 "Barrister's clerk" 'Engineer, automotive' 'Recruitment consultant'
 'Commercial horticulturist' 'Pharmacist, community'
 'Forest/woodland manager' 'Designer, graphic'
 'Civil engineer, consulting' 'Science writer'
 'Health and safety inspector' 'Administrator, Civil Service'
 'Technical sales engineer' 'Special educational needs teacher'
 'Sports therapist' 'Engineer, communications' 'Oceanographer'
 'Archaeologist' 'Personal assistant' 'Animal nutritionist' 'Hydrologist'
 'Arts development officer' 'Herpetologist' 'Medical sales representative'
 'Scientist, research (physical sciences)' 'Higher education lecturer'
 'Nurse, adult' 'Chiropodist' 'Therapeutic radiographer'
 'Designer, television/film set' 'Education officer, environmental'
 'Colour technologist' 'Academic librarian' 'Mudlogger'
 'Designer, textile' 'Chief Strategy Officer' 'Loss adjuster, chartered'
 'Pharmacologist' 'Hydrographic surveyor' 'Engineer, manufacturing'
 'Research scientist (medical)' 'Wellsite geologist'
 'Embryologist, clinical' 'Occupational psychologist'
 'Sales professional, IT' 'Advertising copywriter'
 'Radiographer, therapeutic' 'English as a second language teacher'
 'Occupational therapist'
 'Armed forces logistics/support/administrative officer'
 'Technical author' 'Regulatory affairs officer' 'Optician, dispensing'
 'Theme park manager' 'IT trainer' 'Contracting civil engineer'
 'Psychologist, sport and exercise' 'Manufacturing engineer' 'Musician'
 'Senior tax professional/tax inspector' 'Engineer, biomedical'
 'Facilities manager' 'Osteopath' 'Corporate investment banker'
 'Psychotherapist' 'Copywriter, advertising' 'Horticultural consultant'
 'Microbiologist' 'Educational psychologist'
 'Sport and exercise psychologist' 'Risk manager' 'Health visitor'
 'Visual merchandiser' 'Clinical biochemist' 'Water quality scientist'
 'Optometrist' 'Petroleum engineer' 'Building control surveyor'
 'Financial planner' 'Theatre director' 'Secretary, company'
 'Materials engineer' 'Civil Service fast streamer'
 'Health service manager' 'Scientist, forensic' 'Immigration officer'
 'Dealer' 'Planning and development surveyor' 'Broadcast engineer'
 'Local government officer' 'Nature conservation officer'
 'Private music teacher' 'Geologist, wellsite' 'Gaffer' 'Curator'
 'Editor, commissioning' 'Barrister' 'TEFL teacher'
 'Public relations account executive' 'Audiological scientist'
 'Travel agency manager' 'Land' 'Music therapist' 'Librarian, academic'
 'Film/video editor' 'Journalist, broadcasting' 'Waste management officer'
 'Scientist, water quality' 'Sub' 'Neurosurgeon'
 'Scientist, research (maths)' 'Public house manager' 'Building surveyor'
 'Animator' 'Production assistant, television' 'Transport planner'
```

```
'Geneticist, molecular' 'Merchant navy officer'
'Research scientist (life sciences)' 'Engineer, building services'
'Solicitor, Scotland' 'Hospital pharmacist' 'Engineer, petroleum'
'Oncologist' 'IT technical support officer' 'Site engineer'
'Early years teacher' 'Plant breeder/geneticist'
'Chartered management accountant' 'Runner, broadcasting/film/video'
'Newspaper journalist' 'Naval architect' 'Agricultural engineer'
'Meteorologist' 'Designer, ceramics/pottery'
'Environmental education officer' 'Textile designer'
'Engineer, materials' 'Magazine journalist' 'Conference centre manager'
'Dance movement psychotherapist' 'Warden/ranger'
'Teacher, English as a foreign language'
'Producer, television/film/video' 'Make' 'Pharmacist, hospital'
'Therapist, horticultural' 'Journalist, newspaper' 'Retail merchandiser'
'Nurse, mental health' 'Chief of Staff' 'Systems analyst'
'Electronics engineer' 'Quantity surveyor' 'Minerals surveyor'
'Scientist, research (life sciences)' 'Archivist' 'Brewing technologist'
'Investment banker, operational' 'Accountant, chartered certified'
'Surveyor, minerals' 'Hospital doctor' 'Theatre stage manager'
'Operational researcher' 'Tax inspector' 'Television camera operator'
'Arts administrator' 'Patent attorney' 'Bonds trader' 'Ship broker'
'Furniture conservator/restorer' 'Media planner'
'Radio broadcast assistant' 'Mental health nurse' 'Purchasing manager'
'Scientist, biomedical' 'Photographer' 'Sports coach'
'Environmental manager' 'Estate agent' 'Public librarian'
'Designer, blown glass/stained glass' 'Occupational hygienist' 'Surgeon'
'Youth worker' 'Hotel manager' 'Programmer, systems'
"Politician's assistant" 'Social researcher' 'Publishing copy'
'Tax adviser' 'Quarry manager' 'Buyer, industrial' 'Production manager'
'Police officer' 'Theatre manager' 'Sports administrator'
'Research scientist (maths)' 'Therapist, music' 'Soil scientist'
'Holiday representative' 'Producer, radio' 'Intelligence analyst'
'Geochemist' 'Probation officer' 'Fish farm manager'
'Chartered accountant' 'Architect' 'Psychiatric nurse' 'Farm manager'
'Geoscientist' 'Lecturer, further education' 'Horticulturist, commercial'
'Surveyor, quantity' 'Clothing/textile technologist' 'Technical brewer'
'Landscape architect' 'Information systems manager' 'Sales executive'
'Exercise physiologist' 'Administrator, arts' 'Careers adviser'
'Lobbyist' 'Claims inspector/assessor' 'Recycling officer'
'Product/process development scientist' 'Paramedic' 'Fine artist'
'Teacher, secondary school' 'Data processing manager'
'Government social research officer' 'Product manager'
'Accounting technician' 'Engineer, land' 'Lawyer' 'Restaurant manager'
'Catering manager' 'Contractor' 'Research officer, government'
'Medical secretary' 'Podiatrist' 'Phytotherapist'
'Surveyor, building control' 'Comptroller'
'Lighting technician, broadcasting/film/video' 'Paediatric nurse'
'Designer, furniture' 'Adult guidance worker'
'Clinical molecular geneticist' 'Games developer' 'Metallurgist'
'Armed forces technical officer' 'Risk analyst'
'Careers information officer' 'Garment/textile technologist'
'Multimedia specialist' 'Trade union research officer'
'Museum/gallery exhibitions officer' 'Armed forces operational officer'
'Air broker' 'Mechanical engineer' 'Ceramics designer' 'Airline pilot'
'Trading standards officer' 'Advice worker' 'Music tutor'
'Leisure centre manager' 'Surveyor, rural practice'
```

```
'Scientist, physiological' 'Fisheries officer'
'Research officer, trade union' 'Licensed conveyancer'
"Nurse, children's" 'Museum/gallery curator' 'Psychologist, occupational'
'Astronomer' 'Therapist, drama' 'Therapist, speech and language'
'Surveyor, land/geomatics' 'Production assistant, radio'
'Human resources officer' 'Fast food restaurant manager' 'Orthoptist'
'Public relations officer' 'Bookseller' 'Health and safety adviser'
'Clinical cytogeneticist' 'Ergonomist'
'Psychologist, prison and probation services' 'Actuary'
'Scientist, clinical (histocompatibility and immunogenetics)'
'Community development worker' 'Consulting civil engineer'
'Television production assistant' 'Veterinary surgeon'
'Teacher, adult education' 'Civil engineer, contracting'
'Architectural technologist' 'Volunteer coordinator'
'Primary school teacher' 'Insurance underwriter'
'Research officer, political party' 'Radiation protection practitioner'
'Psychotherapist, child' 'Interior and spatial designer'
'Therapist, nutritional' 'Jewellery designer' 'Press sub'
'Clinical scientist, histocompatibility and immunogenetics'
'Administrator, sports' 'Insurance account manager'
'Museum/gallery conservator' 'Furniture designer' 'Haematologist'
'Associate Professor' 'Physicist, medical' 'Pathologist'
'Chartered public finance accountant' 'Printmaker' 'Surveyor, mining'
'Chief Marketing Officer' 'General practice doctor' 'Chemical engineer'
'Forensic scientist' 'Marketing executive' 'Art gallery manager'
'Therapist, sports' 'Insurance claims handler' 'Secondary school teacher'
'Development worker, international aid' 'Quality manager'
'Conservator, furniture' 'Tour manager'
'Control and instrumentation engineer' 'Adult nurse'
'Diplomatic Services operational officer' 'Cartographer' 'Chiropractor'
'Land/geomatics surveyor' 'Statistician' 'Financial trader'
'Special effects artist' 'Clinical psychologist'
'Further education lecturer' 'Engineer, water' 'Energy manager'
'Education administrator' 'Art therapist' 'Television floor manager'
'Legal secretary' 'Merchandiser, retail' 'Web designer'
'Nurse, learning disability' 'International aid/development worker'
'Warehouse manager' 'Engineer, mining' 'Exhibition designer'
'Administrator, local government' 'Water engineer' 'Physiotherapist'
'Engineer, electronics' 'Equities trader' 'Telecommunications researcher'
'Hydrogeologist' 'Community education officer' 'Engineer, energy'
'Scientist, audiological' 'Patent examiner' 'Retail manager'
'Engineer, aeronautical' 'Engineer, site' 'Engineer, civil (contracting)'
'Proofreader' 'Scientist, marine' 'Speech and language therapist'
'IT sales professional' 'Buyer, retail' 'Network engineer'
'Commercial art gallery manager'
'Chartered legal executive (England and Wales)' 'Presenter, broadcasting'
'Surveyor, planning and development'
'Research scientist (physical sciences)' 'Commissioning editor'
'Operational investment banker' 'Seismic interpreter' 'Charity officer'
'English as a foreign language teacher' 'Scientist, research (medical)'
'Designer, interior/spatial' 'Lexicographer' 'Therapist, art'
'Clinical embryologist' 'Child psychotherapist' 'Midwife'
'Pensions consultant' 'Tree surgeon' 'Health physicist' 'Artist'
'Company secretary' 'Fashion designer' 'IT consultant'
'Teacher, early years/pre' 'Geographical information systems officer'
'Tourist information centre manager' 'Biomedical engineer'
```

```
'Biomedical scientist' 'Financial risk analyst' 'Multimedia programmer'
'Engineer, control and instrumentation' 'Insurance broker'
'Drilling engineer' 'Development worker, community'
'Designer, industrial/product' 'Medical technical officer'
'Advertising account executive' 'Counselling psychologist'
'Tourism officer' 'Dancer' 'Social research officer, government'
'Teacher, music' 'Translator' 'Race relations officer'
'Engineer, civil (consulting)'
'Historic buildings inspector/conservation officer' 'Financial manager'
'Financial controller' 'Designer, jewellery' 'Retail banker'
'Administrator, charities/voluntary organisations'
'Magazine features editor' 'Psychotherapist, dance movement' 'Barista'
'Passenger transport manager' 'Mining engineer'
'Administrator, education'
'Programme researcher, broadcasting/film/video' 'Ranger/warden' 'Actor'
'Pension scheme manager' 'Investment analyst' 'Physiological scientist'
'Advertising art director' 'Sports development officer'
'Manufacturing systems engineer' 'Accommodation manager'
'Television/film/video producer' 'Accountant, chartered'
'Engineer, agricultural' 'Horticultural therapist' 'Economist'
'Training and development officer' 'Engineer, maintenance'
'Logistics and distribution manager' 'Psychologist, clinical'
'Accountant, chartered management' 'Rural practice surveyor'
'Biochemist, clinical' 'Set designer' 'Nutritional therapist'
'Illustrator' 'Designer, exhibition/display'
'Armed forces training and education officer' 'Location manager'
'Charity fundraiser' 'Community pharmacist'
'Geophysicist/field seismologist' 'Designer, fashion/clothing'
'Computer games developer' 'Acupuncturist' 'Database administrator'
'Stage manager' 'Operations geologist' 'Marine scientist'
'Glass blower/designer' 'Corporate treasurer' 'Ecologist'
'Structural engineer' 'Housing manager/officer' 'Chief Operating Officer'
'Engineer, manufacturing systems' 'Herbalist' 'Editor, film/video'
'Retail buyer' 'Doctor, hospital' 'Prison officer' 'Ophthalmologist'
'Chemist, analytical' 'Chartered certified accountant' 'Industrial buyer'
'Video editor' 'Publishing rights manager' 'Engineer, drilling'
'Food technologist' 'Arboriculturist' 'Engineer, technical sales'
'Systems developer' 'Firefighter' 'Education officer, museum'
'Media buyer' 'Records manager' 'Aid worker' 'Pilot, airline'
'Advertising account planner' 'Psychologist, counselling'
'Environmental consultant' 'Copy' 'Trade mark attorney'
'Psychologist, forensic' 'Social worker' 'Administrator'
'Agricultural consultant' 'Education officer, community'
'Management consultant' 'Field trials officer' 'Graphic designer'
'Teacher, primary school' 'Homeopath' 'Cabin crew'
'Editor, magazine features' 'Medical physicist'
'Medical laboratory scientific officer' 'Press photographer'
'Field seismologist' 'Estate manager/land agent'
'Industrial/product designer' 'Software engineer' 'Air cabin crew'
'Freight forwarder' 'Engineer, structural' 'Fitness centre manager'
'Interpreter' 'Scientific laboratory technician' 'Data scientist'
'Electrical engineer' 'Clinical research associate'
'Engineering geologist' 'Call centre manager' 'Psychologist, educational'
'Conservator, museum/gallery' 'Emergency planning/management officer'
'Communications engineer' 'Conservation officer, historic buildings'
'Cytogeneticist' 'Personnel officer' 'Dramatherapist'
```

```
 'Investment banker, corporate' 'Camera operator'
 'Chartered loss adjuster' 'Health promotion specialist'
 'Scientist, product/process development' 'Learning mentor'
 'Lecturer, higher education' 'Sound technician, broadcasting/film/video'
 'Restaurant manager, fast food' 'Engineer, maintenance (IT)'
 'Energy engineer' 'Dispensing optician' 'Chief Executive Officer'
 'Ambulance person' 'Public affairs consultant' 'Product designer'
 'Community arts worker' 'Higher education careers adviser' 'Dentist'
 'Exhibitions officer, museum/gallery' 'Futures trader'
 'Commercial/residential surveyor' 'Engineer, production'
 'Animal technologist' 'Banker' 'Programmer, applications' 'Best boy'
 'Secretary/administrator' 'Journalist, magazine' 'Production engineer'
 'Accountant, chartered public finance' 'Geologist, engineering'
 'Aeronautical engineer' 'Engineer, chemical' 'Forensic psychologist'
 'Broadcast journalist' 'Town planner' 'Toxicologist' 'Writer']
Children
[nan  1.  4.  0.  3.  2.  7.  5.  9.  6. 10.  8.]
Age
[68. 27. 50. 48. 83. nan 49. 86. 23. 56. 30. 39. 63. 60. 61. 52. 75. 77.
 47. 70. 69. 45. 40. 82. 26. 25. 66. 72. 41. 44. 43. 84. 59. 31. 51. 58.
 73. 33. 42. 81. 87. 54. 67. 46. 24. 20. 71. 32. 29. 80. 53. 79. 65. 35.
 34. 74. 55. 76. 57. 38. 78. 19. 36. 88. 62. 37. 28. 22. 85. 89. 18. 21.
 64.]
Education
["Master's Degree" 'Regular High School Diploma' 'Doctorate Degree'
 'No Schooling Completed' "Associate's Degree" "Bachelor's Degree"
 'Some College, Less than 1 Year' 'GED or Alternative Credential'
 'Some College, 1 or More Years, No Degree'
 '9th Grade to 12th Grade, No Diploma' 'Nursery School to 8th Grade'
 'Professional School Degree']
Employment
['Part Time' 'Retired' 'Student' 'Full Time' 'Unemployed']
Income
[28561.99 21704.77      nan ... 56384.35 55723.74 16667.58]
Marital
['Widowed' 'Married' 'Separated' 'Never Married' 'Divorced']
Gender
['Male' 'Female' 'Prefer not to answer']
Churn
['No' 'Yes']
Outage_sec_perweek
[ 6.97256609 12.01454108 10.24561565 ...  4.837696   12.07646
 12.64176   ]
Email
[10 12  9 15 16 20 18 17 14 13 11  8  6  5 19  3  7 21  4  2  1 23 22]
Contacts
[0 2 3 1 4 5 6 7]
Yearly_equip_failure
[1 0 3 2 4 6]
Techie
['No' 'Yes' nan]
Contract
['One year' 'Month-to-month' 'Two Year']
Port_modem
['Yes' 'No']
Tablet
```

```
['Yes' 'No']
InternetService
['Fiber Optic' 'DSL' nan]
Phone
['Yes' 'No' nan]
Multiple
['No' 'Yes']
OnlineSecurity
['Yes' 'No']
OnlineBackup
['Yes' 'No']
DeviceProtection
['No' 'Yes']
TechSupport
['No' 'Yes' nan]
StreamingTV
['No' 'Yes']
StreamingMovies
['Yes' 'No']
PaperlessBilling
['Yes' 'No']
PaymentMethod
['Credit Card (automatic)' 'Bank Transfer(automatic)' 'Mailed Check'
 'Electronic Check']
Tenure
[ 6.79551295  1.156681   15.75414408 ... 61.04037    71.0956
 63.35086   ]
MonthlyCharge
[171.4497621 242.9480155 159.4403984 ... 168.2209    252.6286
 218.371    ]
Bandwidth_GB_Year
[ 904.5361102  800.9827661 2054.706961  ... 4159.306     6468.457
 5857.586    ]
item1
[5 3 4 6 2 1 7]
item2
[5 4 3 2 6 1 7]
item3
[5 3 2 4 6 1 7 8]
item4
[3 4 2 5 7 6 1]
item5
[4 5 1 2 3 7 6]
item6
[4 3 5 2 6 7 1 8]
item7
[3 4 5 2 6 7 1]
item8
[4 3 5 2 1 6 7 8]
```

Below, we will check to see what percentage of each column is missing to determine treatment method.

In [ ]:
```
# percentage of missing
churn.isnull().mean() * 100
```

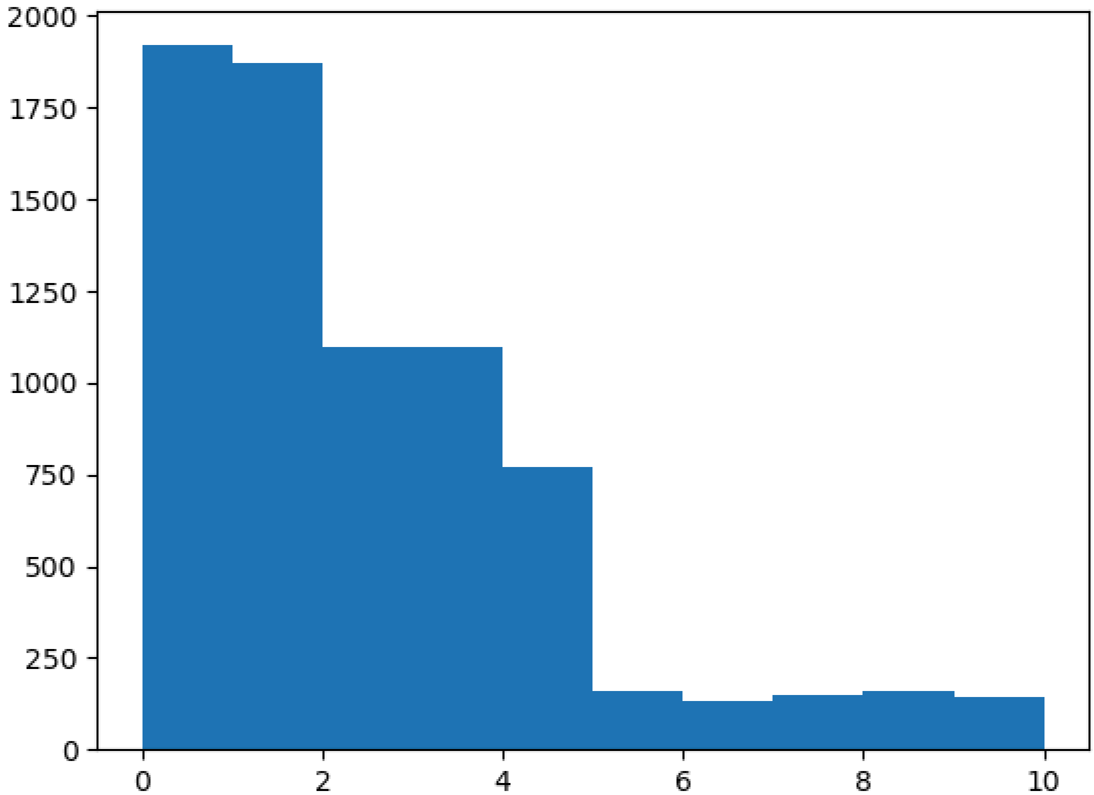
Out[ ]:
```
Unnamed: 0              0.00
CaseOrder               0.00
Customer_id             0.00
Interaction             0.00
City                    0.00
State                   0.00
County                  0.00
Zip                     0.00
Lat                     0.00
Lng                     0.00
Population              0.00
Area                    0.00
Timezone                0.00
Job                     0.00
Children               24.95
Age                    24.75
Education               0.00
Employment              0.00
Income                 24.90
Marital                 0.00
Gender                  0.00
Churn                   0.00
Outage_sec_perweek      0.00
Email                   0.00
Contacts                0.00
Yearly_equip_failure    0.00
Techie                 24.77
Contract                0.00
Port_modem              0.00
Tablet                  0.00
InternetService        21.29
Phone                  10.26
Multiple                0.00
OnlineSecurity          0.00
OnlineBackup            0.00
DeviceProtection        0.00
TechSupport             9.91
StreamingTV             0.00
StreamingMovies         0.00
PaperlessBilling        0.00
PaymentMethod           0.00
Tenure                  9.31
MonthlyCharge           0.00
Bandwidth_GB_Year      10.21
item1                   0.00
item2                   0.00
item3                   0.00
item4                   0.00
item5                   0.00
item6                   0.00
item7                   0.00
item8                   0.00
dtype: float64
```

Since none of the columns are within the threshold for deleteion, that will not be an option for treatment.

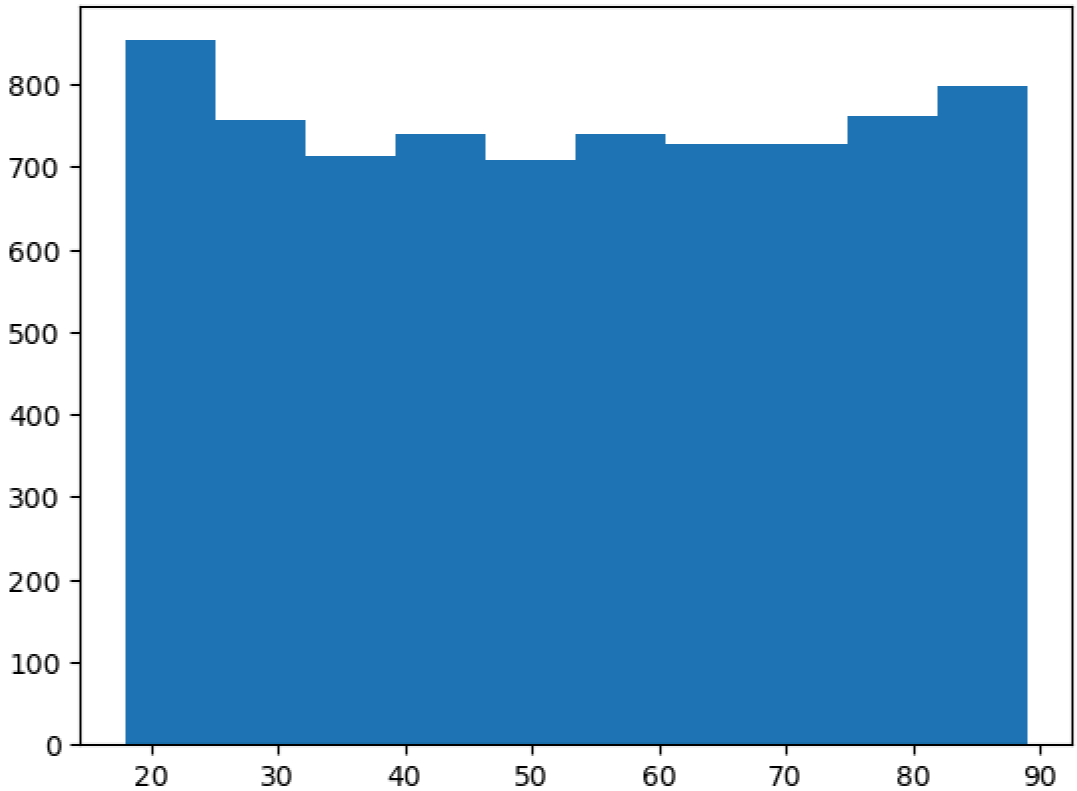
We will now use the plt.hist() function to visualize each columns missing valued on a histogram. This will help determine which method of imputation we will use to clean the missing values.

Note, for columns that are not numerical will not use plt.hist() function.

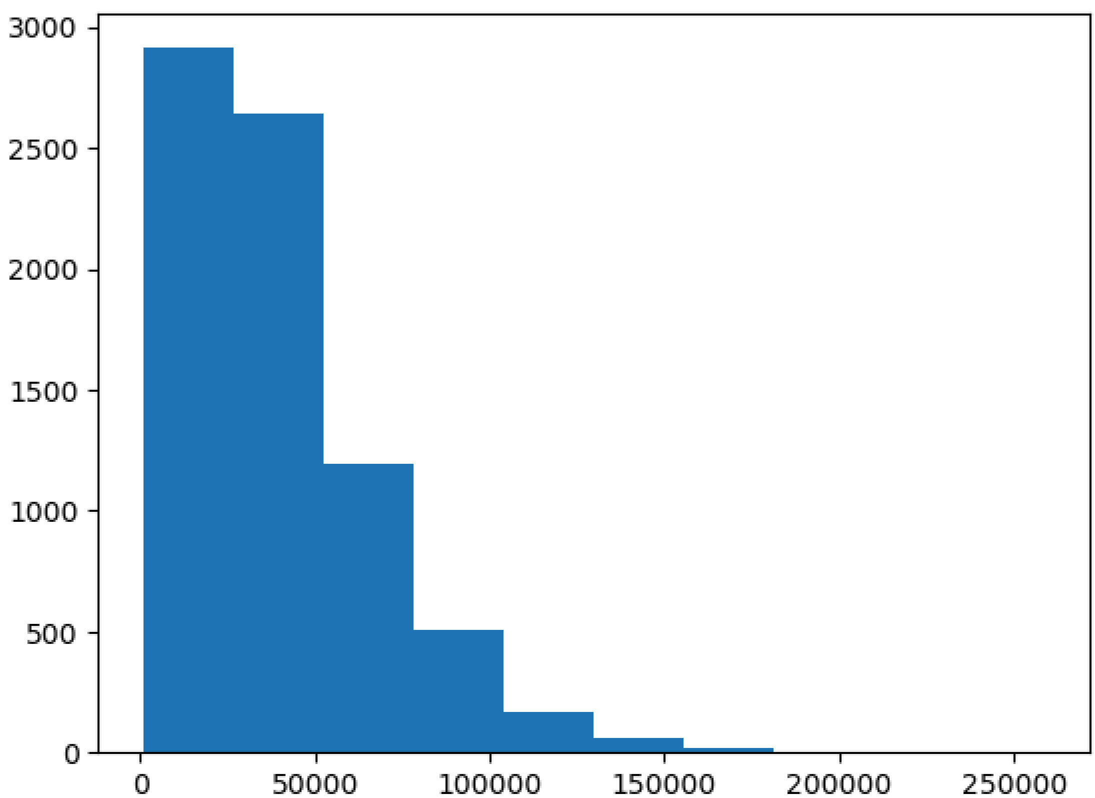
In [ ]:
```
# distribution examanation of Children column
plt.hist(churn["Children"])
plt.show()
```



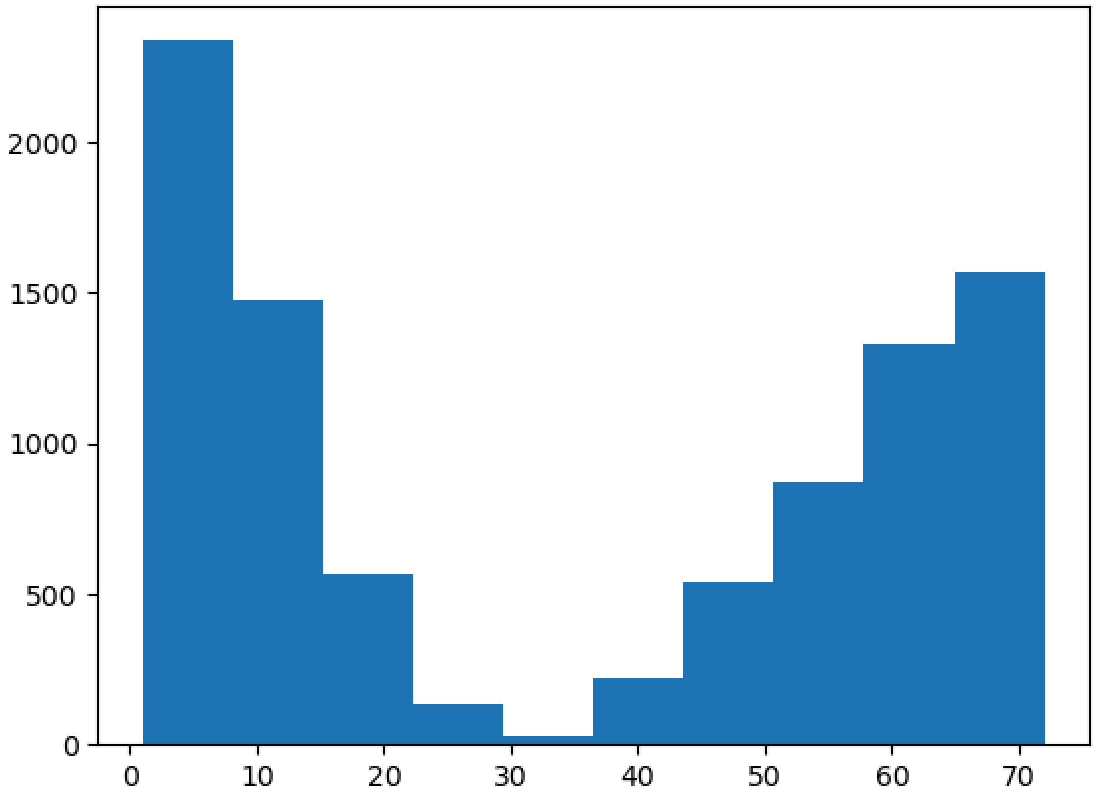
In [ ]:
```
# distribution examination of Age column
plt.hist(churn["Age"])
plt.show()
```



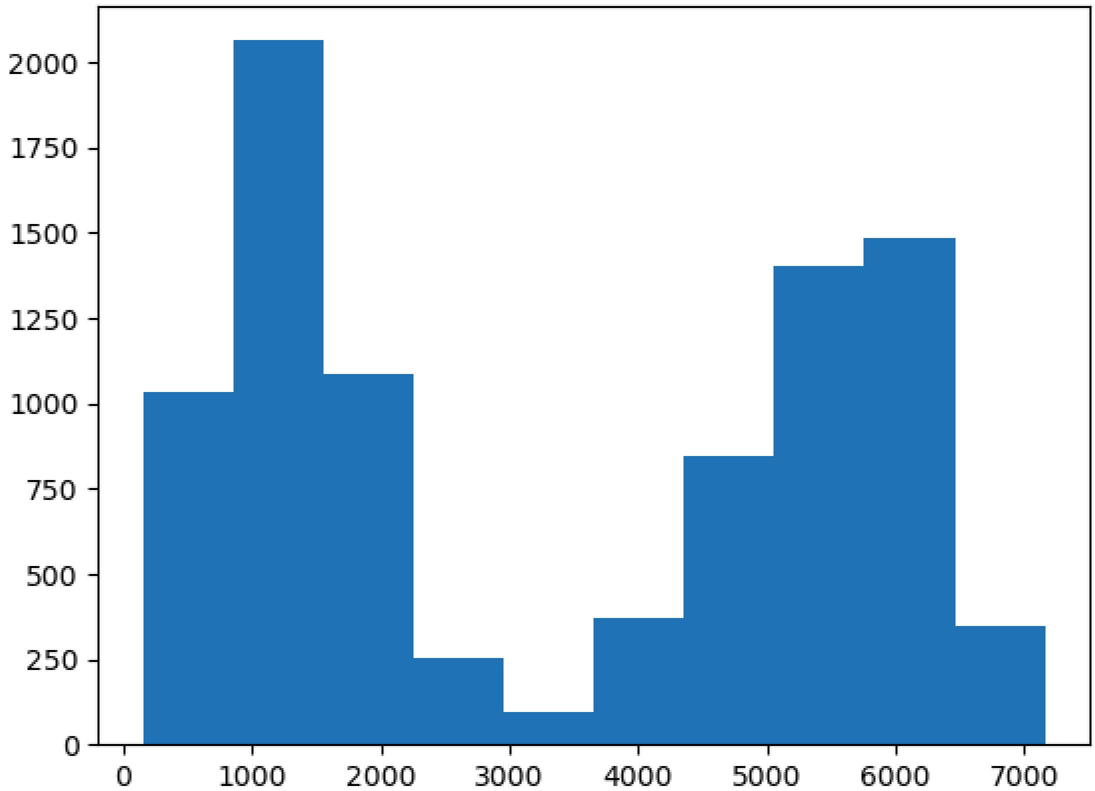
In [ ]:
```
# distribution examination of Income column
plt.hist(churn["Income"])
plt.show()
```



In [ ]:
```
# distribution examination of Tenure column
plt.hist(churn["Tenure"])
plt.show()
```



In [ ]:
```
# distribution examination of Bandwidth_GB_Year column
plt.hist(churn["Bandwidth_GB_Year"])
plt.show()
```



In [ ]:
```
# look at unique values for Techie
churn.Techie.unique()
```

Out[ ]:
```
array(['No', 'Yes', nan], dtype=object)
```

In [ ]:
```
# look at values in Techie
churn.Techie.value_counts()
```

Out[ ]:
```
Techie
No     6266
Yes    1257
Name: count, dtype: int64
```

In [ ]:
```
# look at unique values for InternetService
churn.InternetService.unique()
```

Out[ ]:
```
array(['Fiber Optic', 'DSL', nan], dtype=object)
```

In [ ]:
```
# look at values in InternetService
churn.InternetService.value_counts()
```

Out[ ]:
```
InternetService
Fiber Optic    4408
DSL            3463
Name: count, dtype: int64
```

In [ ]:
```
# look at values in Phone
churn.Phone.value_counts()
```

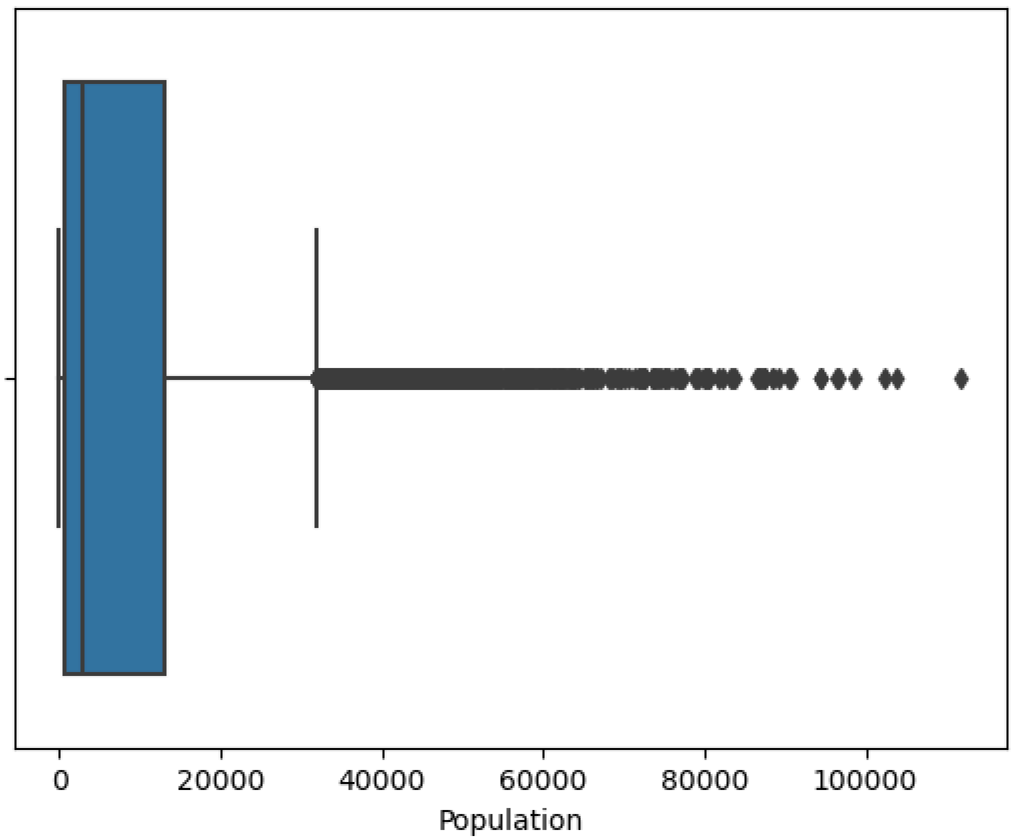
Out[ ]:
```
Phone
Yes    8128
No      846
Name: count, dtype: int64
```

In [ ]:
```
# look at values in TechSupport
churn.TechSupport.value_counts()
```

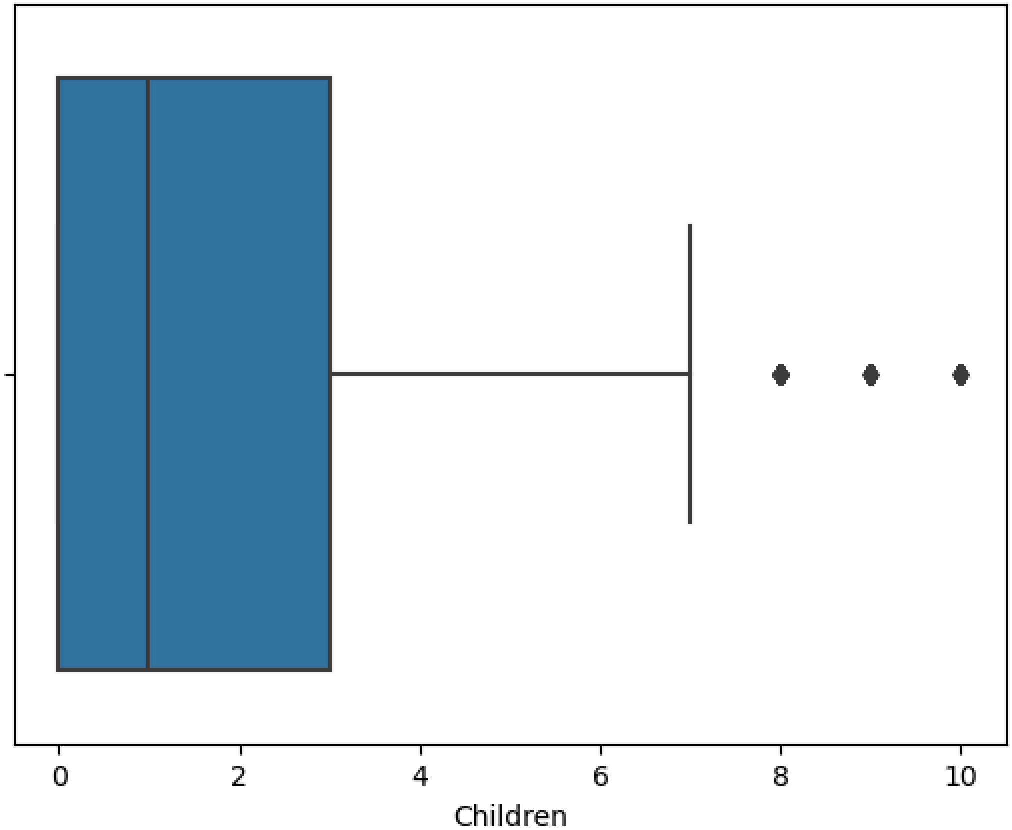
Out[ ]:
```
TechSupport
No     5635
Yes    3374
Name: count, dtype: int64
```

Detection of outliers

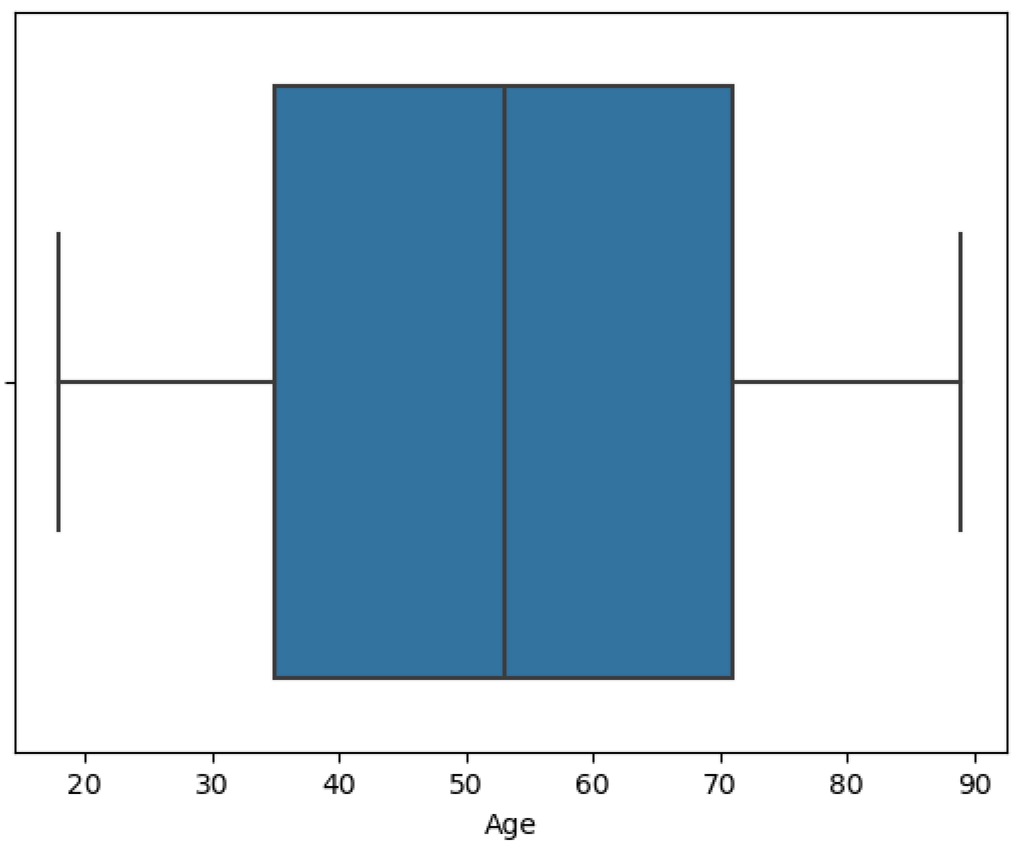
In [ ]:
```
boxplot = sns.boxplot(x="Population", data=churn)
```



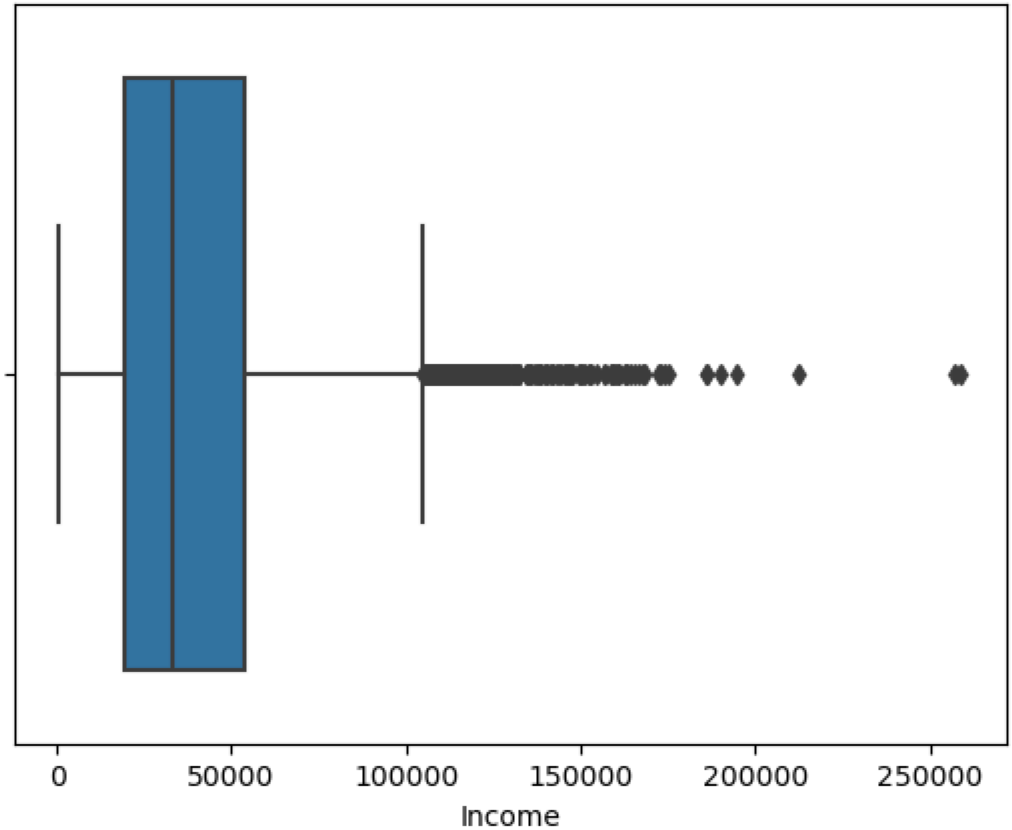
In [ ]:
```
boxplot = sns.boxplot(x="Children", data=churn)
```



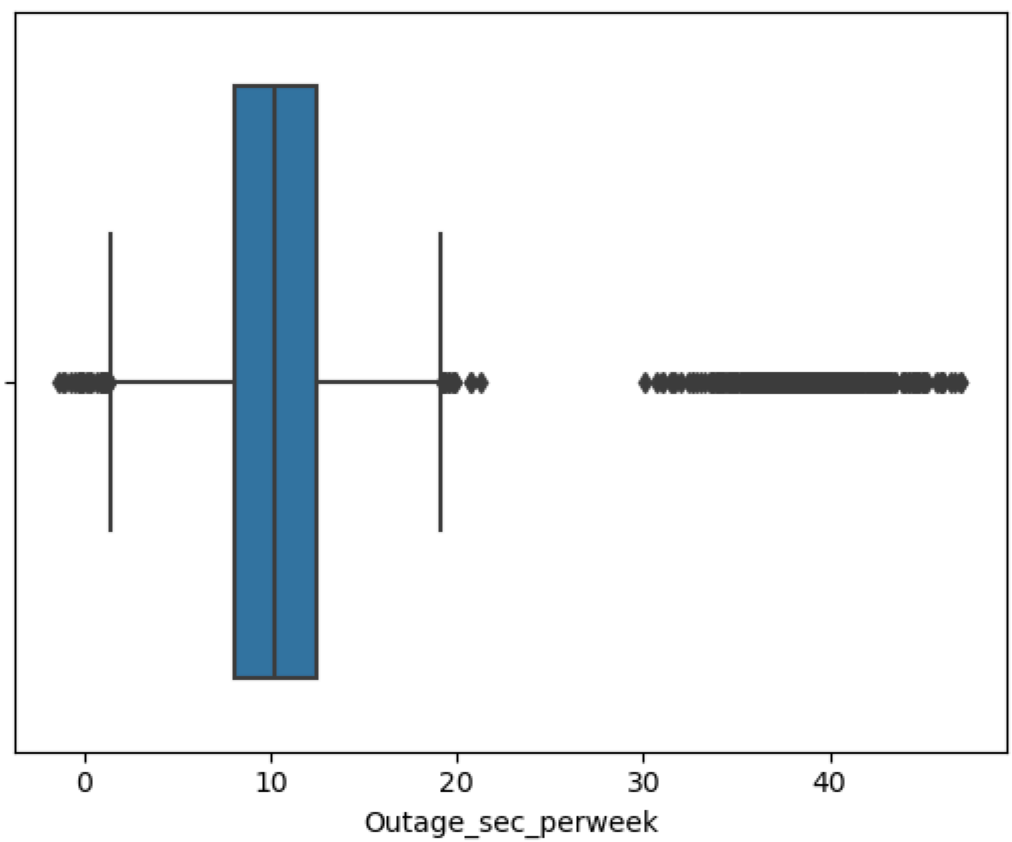

```
In [ ]: boxplot = sns.boxplot(x="Age", data=churn)
```



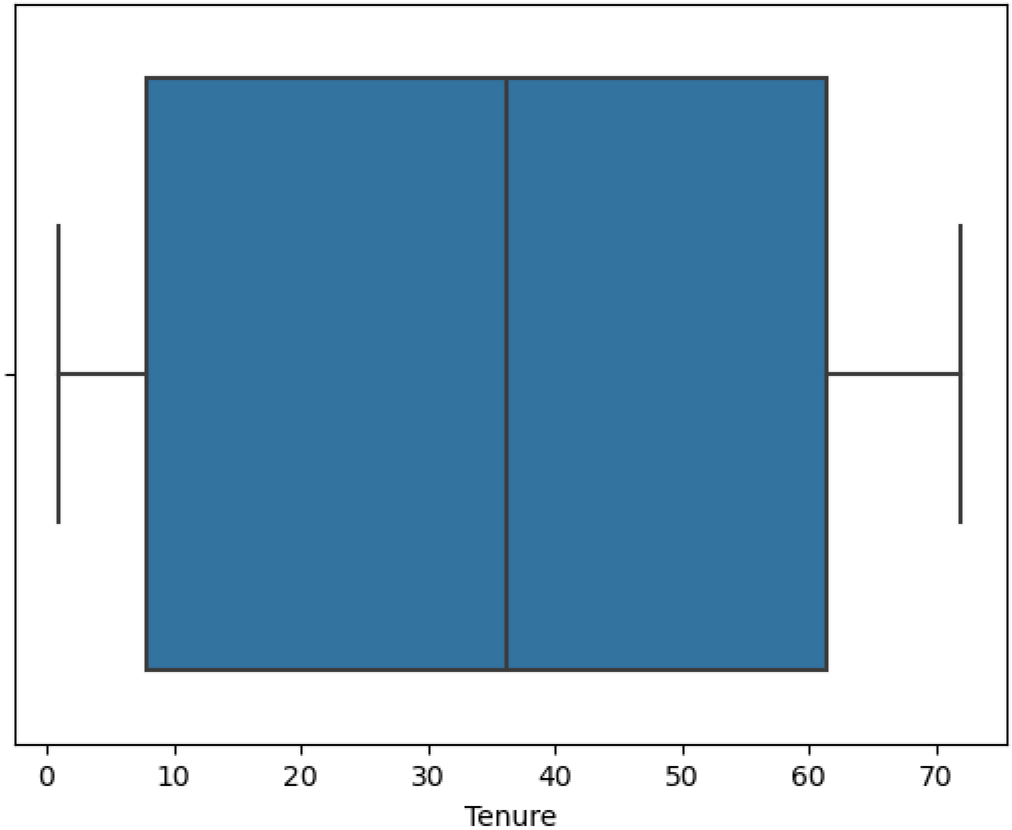

```
In [ ]: boxplot = sns.boxplot(x="Income", data=churn)
```



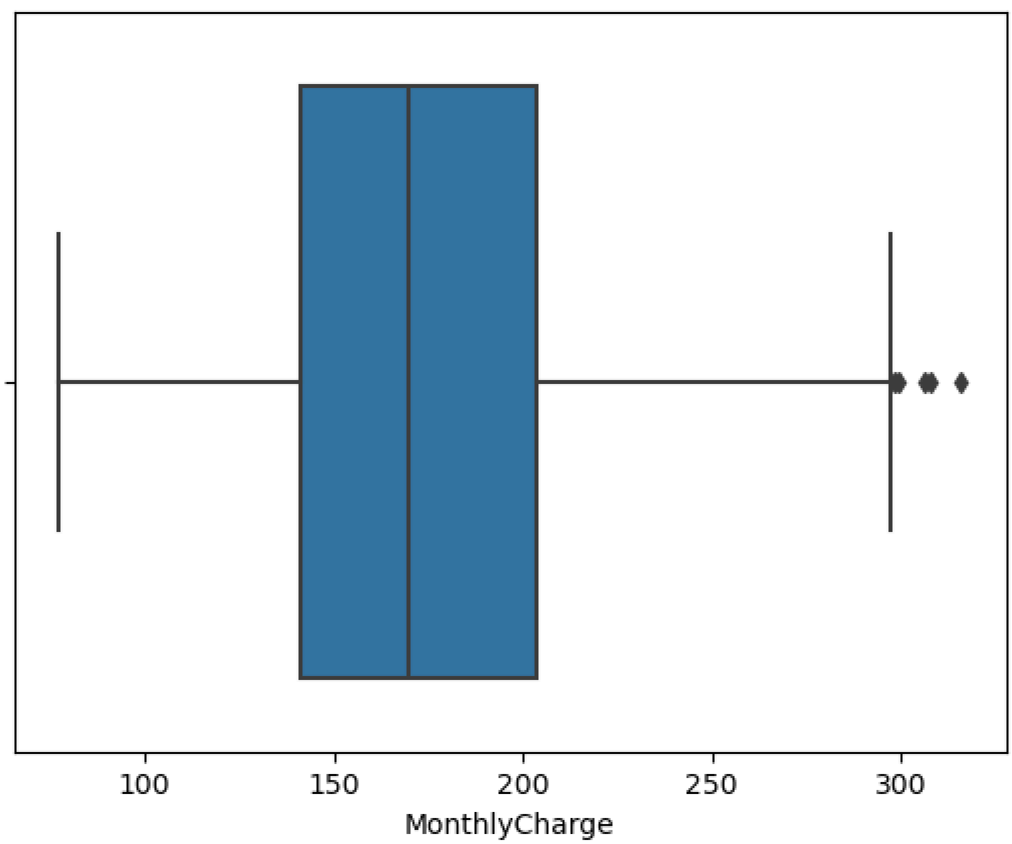
In [ ]:
```
boxplot = sns.boxplot(x="Outage_sec_perweek", data=churn)
```



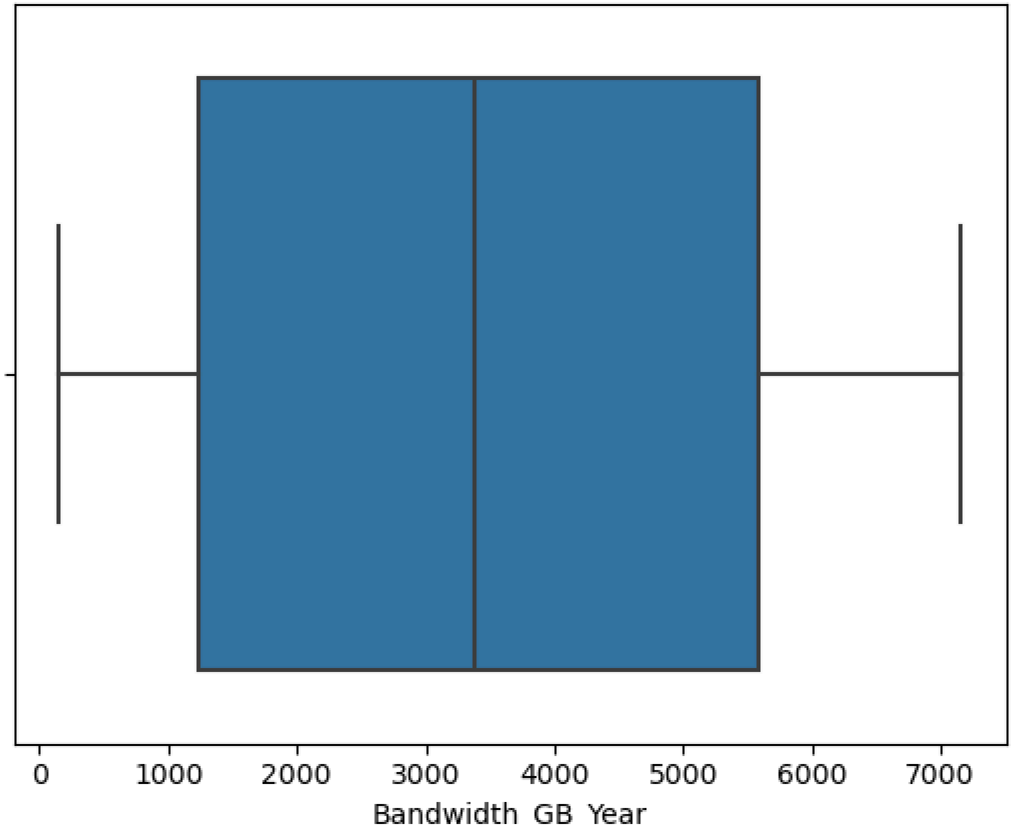
In [ ]:
```
boxplot = sns.boxplot(x="Tenure", data=churn)
```



In [ ]:
```
boxplot = sns.boxplot(x="MonthlyCharge", data=churn)
```



In [ ]:
```
boxplot = sns.boxplot(x="Bandwidth_GB_Year", data=churn)
```



Further exploration

In [ ]:
```
churn.Population.describe()
```

Out[ ]:
```
count     10000.000000
mean       9756.562400
std       14432.698671
min           0.000000
25%         738.000000
50%        2910.500000
75%       13168.000000
max      111850.000000
Name: Population, dtype: float64
```

In [ ]:
```
churn.Children.describe()
```

Out[ ]:
```
count    7505.000000
mean        2.095936
std         2.154758
min         0.000000
25%         0.000000
50%         1.000000
75%         3.000000
max        10.000000
Name: Children, dtype: float64
```

In [ ]:
```
churn.Age.describe()
```

Out[ ]:
```
count    7525.000000
mean       53.275748
std        20.753928
min        18.000000
25%        35.000000
50%        53.000000
75%        71.000000
max        89.000000
Name: Age, dtype: float64
```

In [ ]:
```
churn.Income.describe()
```

Out[ ]:
```
count      7510.000000
mean      39936.762226
std       28358.469482
min         740.660000
25%       19285.522500
50%       33186.785000
75%       53472.395000
max      258900.700000
Name: Income, dtype: float64
```

In [ ]:
```
churn.Outage_sec_perweek.describe()
```

Out[ ]:
```
count    10000.000000
mean        11.452955
std          7.025921
min         -1.348571
25%          8.054362
50%         10.202896
75%         12.487644
max         47.049280
Name: Outage_sec_perweek, dtype: float64
```

In [ ]:
```
churn.Tenure.describe()
```

Out[ ]:
```
count    9069.000000
mean       34.498858
std        26.438904
min         1.000259
25%         7.890442
50%        36.196030
75%        61.426670
max        71.999280
Name: Tenure, dtype: float64
```

In [ ]:
```
churn.MonthlyCharge.describe()
```

Out[ ]:
```
count    10000.000000
mean       174.076305
std         43.335473
min         77.505230
25%        141.071078
50%        169.915400
75%        203.777441
max        315.878600
Name: MonthlyCharge, dtype: float64
```

In [ ]:
```
churn.Bandwidth_GB_Year.describe()
```

Out[ ]:
```
count    8979.000000
mean     3398.842752
std      2187.396807
min       155.506715
25%      1234.110529
50%      3382.424000
75%      5587.096500
max      7158.982000
Name: Bandwidth_GB_Year, dtype: float64
```

Part III: Data Cleaning

D. Data Cleaning Process

1. Description of findings from data cleaning plan(Part C)

- Columns Zip, Lat, Lng currently are floats or ints. There are not quantitative so will change datatype to object.
- Outage_sec_perweek has negative numbers in it. There should not be negative time for an outage. Imputation will be used.
- Education column could be standardized.
- Survey item columns to be renamed based on data dictionary
- First column is not named and matches CaseOrder. Drop first column and re-index
- Numerical columns missing values. Children, Age, Income, Tenure & Bandwidth_GB_Year. Imputation will be used to deal with missing data. Type of imputation will be based on column data.
- Categorical columns missing values. Techie, Phone, TechSupport, InternetService

2. Imputation method will be based on variable type as stated above for numerical columns missing values.

- Categorical columns Techie, Phone and TechSupport will be set to 'No' as that is normally the default value. InternetService will be set to 'No Internet Service' instead of none or nan.
- Outliers identified will not be treated at this time. Most of the outliers are on columns that seem to be reported by the customer and there is not a good way to

infer the values. Outage_sec_perweek is the only exception and will be treated with the appropriate imputation method.

Each step in the cleaning process will include a justification and summary to satisfy parts D3 & D4 of the rubric.

A copy of the data set will be made in order to preserve the original data set.

In [ ]:
```
# copy of data set
churn_clean = churn.copy()
```

In [ ]:
```
# inspect copy
churn_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 52 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            10000 non-null  int64  
 1   CaseOrder             10000 non-null  int64  
 2   Customer_id           10000 non-null  object 
 3   Interaction           10000 non-null  object 
 4   City                  10000 non-null  object 
 5   State                 10000 non-null  object 
 6   County                10000 non-null  object 
 7   Zip                   10000 non-null  int64  
 8   Lat                   10000 non-null  float64
 9   Lng                   10000 non-null  float64
 10  Population            10000 non-null  int64  
 11  Area                  10000 non-null  object 
 12  Timezone              10000 non-null  object 
 13  Job                   10000 non-null  object 
 14  Children              7505 non-null   float64
 15  Age                   7525 non-null   float64
 16  Education             10000 non-null  object 
 17  Employment            10000 non-null  object 
 18  Income                7510 non-null   float64
 19  Marital               10000 non-null  object 
 20  Gender                10000 non-null  object 
 21  Churn                 10000 non-null  object 
 22  Outage_sec_perweek    10000 non-null  float64
 23  Email                 10000 non-null  int64  
 24  Contacts              10000 non-null  int64  
 25  Yearly_equip_failure  10000 non-null  int64  
 26  Techie                7523 non-null   object 
 27  Contract              10000 non-null  object 
 28  Port_modem            10000 non-null  object 
 29  Tablet                10000 non-null  object 
 30  InternetService       7871 non-null   object 
 31  Phone                 8974 non-null   object 
 32  Multiple              10000 non-null  object 
 33  OnlineSecurity        10000 non-null  object 
 34  OnlineBackup          10000 non-null  object 
 35  DeviceProtection      10000 non-null  object 
 36  TechSupport           9009 non-null   object 
 37  StreamingTV           10000 non-null  object 
 38  StreamingMovies       10000 non-null  object 
 39  PaperlessBilling      10000 non-null  object 
 40  PaymentMethod         10000 non-null  object 
 41  Tenure                9069 non-null   float64
 42  MonthlyCharge         10000 non-null  float64
 43  Bandwidth_GB_Year     8979 non-null   float64
 44  item1                 10000 non-null  int64  
 45  item2                 10000 non-null  int64  
 46  item3                 10000 non-null  int64  
 47  item4                 10000 non-null  int64  
 48  item5                 10000 non-null  int64  
 49  item6                 10000 non-null  int64  
 50  item7                 10000 non-null  int64  
```

```
 51  item8                  10000 non-null  int64
dtypes: float64(9), int64(15), object(28)
memory usage: 4.0+ MB
```

In [ ]:
```
churn_clean.shape
```

Out[ ]:
```
(10000, 52)
```

In [ ]:
```
churn_clean.columns
```

Out[ ]:
```
Index(['Unnamed: 0', 'CaseOrder', 'Customer_id', 'Interaction', 'City',
       'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area',
       'Timezone', 'Job', 'Children', 'Age', 'Education', 'Employment',
       'Income', 'Marital', 'Gender', 'Churn', 'Outage_sec_perweek', 'Email',
       'Contacts', 'Yearly_equip_failure', 'Techie', 'Contract', 'Port_modem',
       'Tablet', 'InternetService', 'Phone', 'Multiple', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'PaperlessBilling', 'PaymentMethod', 'Tenure',
       'MonthlyCharge', 'Bandwidth_GB_Year', 'item1', 'item2', 'item3',
       'item4', 'item5', 'item6', 'item7', 'item8'],
      dtype='object')
```

Columns Zip, Lat, Lng currently are floats or ints. There are not quantitative so will change datatype to object. These values are not meant for any type of calculation. Since the rest of the data types for categorical are set to object, these will aslo.

In [ ]:
```
obj_columns = ["Zip", "Lat", "Lng"]
churn_clean[obj_columns] = churn_clean[obj_columns].astype(object)
```

In [ ]:
```
# inspect copy after type change
churn_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 52 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            10000 non-null  int64  
 1   CaseOrder             10000 non-null  int64  
 2   Customer_id           10000 non-null  object 
 3   Interaction           10000 non-null  object 
 4   City                  10000 non-null  object 
 5   State                 10000 non-null  object 
 6   County                10000 non-null  object 
 7   Zip                   10000 non-null  object 
 8   Lat                   10000 non-null  object 
 9   Lng                   10000 non-null  object 
 10  Population            10000 non-null  int64  
 11  Area                  10000 non-null  object 
 12  Timezone              10000 non-null  object 
 13  Job                   10000 non-null  object 
 14  Children              7505 non-null   float64
 15  Age                   7525 non-null   float64
 16  Education             10000 non-null  object 
 17  Employment            10000 non-null  object 
 18  Income                7510 non-null   float64
 19  Marital               10000 non-null  object 
 20  Gender                10000 non-null  object 
 21  Churn                 10000 non-null  object 
 22  Outage_sec_perweek    10000 non-null  float64
 23  Email                 10000 non-null  int64  
 24  Contacts              10000 non-null  int64  
 25  Yearly_equip_failure  10000 non-null  int64  
 26  Techie                7523 non-null   object 
 27  Contract              10000 non-null  object 
 28  Port_modem            10000 non-null  object 
 29  Tablet                10000 non-null  object 
 30  InternetService       7871 non-null   object 
 31  Phone                 8974 non-null   object 
 32  Multiple              10000 non-null  object 
 33  OnlineSecurity        10000 non-null  object 
 34  OnlineBackup          10000 non-null  object 
 35  DeviceProtection      10000 non-null  object 
 36  TechSupport           9009 non-null   object 
 37  StreamingTV           10000 non-null  object 
 38  StreamingMovies       10000 non-null  object 
 39  PaperlessBilling      10000 non-null  object 
 40  PaymentMethod         10000 non-null  object 
 41  Tenure                9069 non-null   float64
 42  MonthlyCharge         10000 non-null  float64
 43  Bandwidth_GB_Year     8979 non-null   float64
 44  item1                 10000 non-null  int64  
 45  item2                 10000 non-null  int64  
 46  item3                 10000 non-null  int64  
 47  item4                 10000 non-null  int64  
 48  item5                 10000 non-null  int64  
 49  item6                 10000 non-null  int64  
 50  item7                 10000 non-null  int64  
```

```
 51  item8                  10000 non-null  int64
dtypes: float64(7), int64(14), object(31)
memory usage: 4.0+ MB
```

Outage_sec_perweek has negative numbers in it. There should not be negative time for an outage. We are going to change the negative numbers to 0.

In [ ]:
```
# descriptive stats for Outage_sec_perweek
churn_clean["Outage_sec_perweek"].describe()
```

Out[ ]:
```
count    10000.000000
mean        11.452955
std          7.025921
min         -1.348571
25%          8.054362
50%         10.202896
75%         12.487644
max         47.049280
Name: Outage_sec_perweek, dtype: float64
```

In [ ]:
```
# will use .shape to determine how many negative values
churn_clean[churn_clean["Outage_sec_perweek"] < 0].shape
```

Out[ ]:
```
(11, 52)
```

In [ ]:
```
# 11 values are negative.  Will set them to 0 instead of deleting them.  Deleting t
churn_clean["Outage_sec_perweek"][churn_clean["Outage_sec_perweek"] < 0] = 0
```

```
C:\Users\jjord\AppData\Local\Temp\ipykernel_15200\628072134.py:2: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  churn_clean["Outage_sec_perweek"][churn_clean["Outage_sec_perweek"] < 0] = 0
```

In [ ]:
```
# verify with descriptive stats for Outage_sec_perweek
# min should be 0 now
churn_clean["Outage_sec_perweek"].describe()
```

Out[ ]:
```
count    10000.000000
mean        11.453592
std          7.024842
min          0.000000
25%          8.054362
50%         10.202896
75%         12.487644
max         47.049280
Name: Outage_sec_perweek, dtype: float64
```

Education column could be standardized. Will create a dictionary to standardize values in the Education column as the current values are too granular.

In [ ]:
```
# Education current value counts
```

```
churn_clean["Education"].value_counts()
```

Out[ ]:

```
Education
Regular High School Diploma                  2421
Bachelor's Degree                            1703
Some College, 1 or More Years, No Degree    1562
9th Grade to 12th Grade, No Diploma          870
Master's Degree                              764
Associate's Degree                           760
Some College, Less than 1 Year               652
Nursery School to 8th Grade                  449
GED or Alternative Credential                387
Professional School Degree                   198
No Schooling Completed                       118
Doctorate Degree                             116
Name: count, dtype: int64
```

In [ ]:

```
# create dictionary mapping
education_levels = {
    "Master's Degree": "Graduate Degree",
    "Regular High School Diploma": "Highschool",
    "Doctorate Degree": "Graduate Degree",
    "No Schooling Completed": "No Diploma",
    "Associate's Degree": "Associate's/Some College",
    "Bachelor's Degree": "Bachelor's Degree",
    "Some College, Less than 1 Year": "Associate's/Some College",
    "GED or Alternative Credential": "Graduate Degree",
    "Some College, 1 or More Years, No Degree": "Associate's/Some College",
    "9th Grade to 12th Grade, No Diploma": "No Diploma",
    "Nursery School to 8th Grade": "No Diploma",
    "Professional School Degree": "Graduate Degree",
}
# map to dictionary
churn_clean["Education"] = churn_clean["Education"].map(education_levels)
```

In [ ]:

```
# Education value counts after mapping
churn_clean["Education"].value_counts()
```

Out[ ]:

```
Education
Associate's/Some College    2974
Highschool                  2421
Bachelor's Degree           1703
Graduate Degree             1465
No Diploma                  1437
Name: count, dtype: int64
```

Survey item columns to be renamed based on data dictionary.

In [ ]:

```
# rename item columns
churn_clean.rename(
    columns={
        "item1": "Timely_response",
        "item2": "Timely_fixes",
        "item3": "Timely_replacements",
        "item4": "Reliability",
```

```
        "item5": "Options",
        "item6": "Respectful_response",
        "item7": "Courteous_exchange",
        "item8": "Active_listening",
    },
    inplace=True,
)
```

In [ ]:
```
# verify columns after rename
churn_clean.columns
```

Out[ ]:
```
Index(['Unnamed: 0', 'CaseOrder', 'Customer_id', 'Interaction', 'City',
       'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area',
       'Timezone', 'Job', 'Children', 'Age', 'Education', 'Employment',
       'Income', 'Marital', 'Gender', 'Churn', 'Outage_sec_perweek', 'Email',
       'Contacts', 'Yearly_equip_failure', 'Techie', 'Contract', 'Port_modem',
       'Tablet', 'InternetService', 'Phone', 'Multiple', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'PaperlessBilling', 'PaymentMethod', 'Tenure',
       'MonthlyCharge', 'Bandwidth_GB_Year', 'Timely_response', 'Timely_fixes',
       'Timely_replacements', 'Reliability', 'Options', 'Respectful_response',
       'Courteous_exchange', 'Active_listening'],
      dtype='object')
```

First column is not named and matches CaseOrder. Drop first column and re-index will ensure correct indexing.

In [ ]:
```
churn_clean.iloc[:, 0:1].head()
```

Out[ ]:

	Unnamed: 0
0	1
1	2
2	3
3	4
4	5

In [ ]:
```
# inplace drop of column
churn_clean.drop(churn_clean.columns[0], axis=1, inplace=True)
```

In [ ]:
```
# index reset
churn_clean = churn_clean.reset_index(drop=True)
churn_clean.set_index("CaseOrder", inplace=True)
```

In [ ]:
```
# verify after column drop and re-index
churn_clean.head()
```

Out[ ]:

CaseOrder	Customer_id	Interaction	City	State	County	Zip	Lat	
1	K409198	aa90260b-4141-4a24-8e36-b04ce1f4f77b	Point Baker	AK	Prince of Wales-Hyder	99927	56.251	-133.37
2	S120509	fb76459f-c047-4a9d-8af9-e0f7d4ac2524	West Branch	MI	Ogemaw	48661	44.32893	-84.2
3	K191035	344d114c-3736-4be5-98f7-c72c281e2d35	Yamhill	OR	Yamhill	97148	45.35589	-123.24
4	D90850	abfa2b40-2d43-4994-b15a-989b8c79e311	Del Mar	CA	San Diego	92014	32.96687	-117.24
5	K662701	68a861fd-0d20-4e51-a587-8a90407ee574	Needville	TX	Fort Bend	77461	29.38012	-95.80

5 rows × 50 columns

Numerical columns missing values. Children, Age, Income, Tenure & Bandwidth_GB_Year. Imputation will be used to deal with missing data. Type of imputation will be based on column data.

In [ ]:
```
# create list of numeric columns
numeric_col = ["Children", "Age", "Income", "Tenure", "Bandwidth_GB_Year"]
```

In [ ]:
```
# descriptive stats for numeric_col
churn_clean[numeric_col].describe()
```

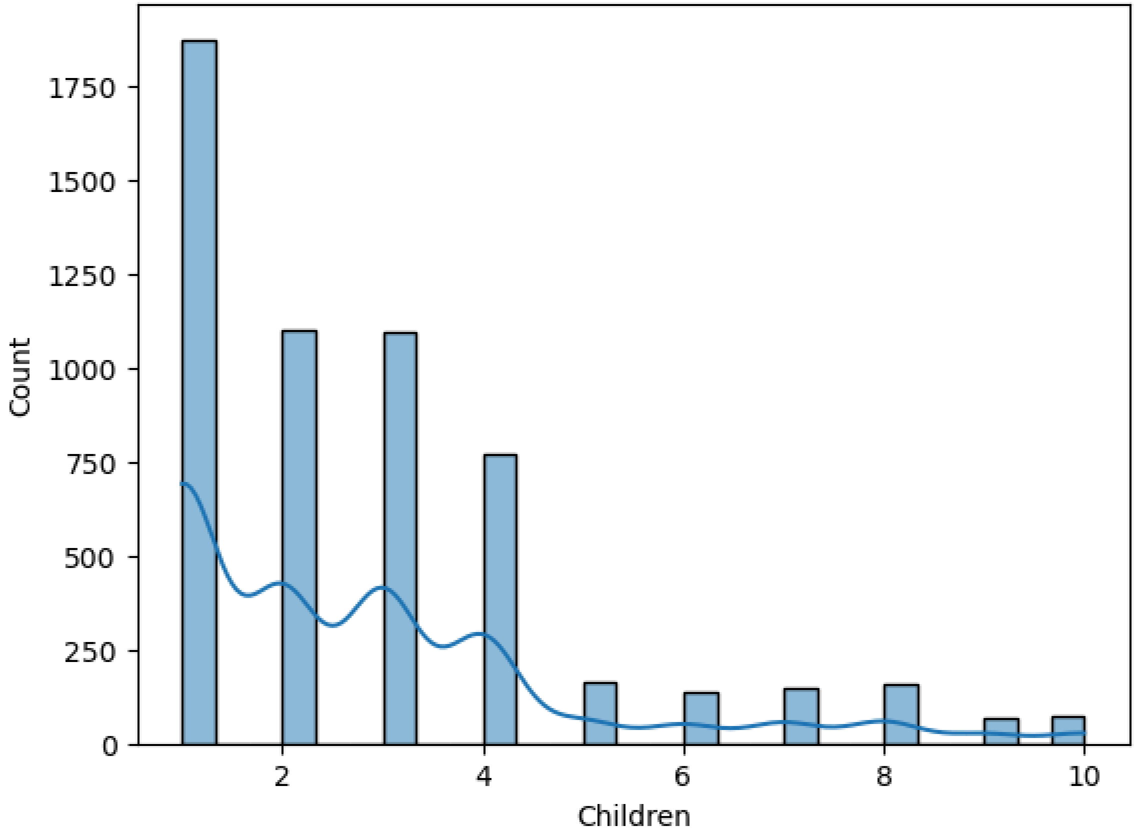
Out[ ]:

	Children	Age	Income	Tenure	Bandwidth_GB_Year
count	7505.000000	7525.000000	7510.000000	9069.000000	8979.000000
mean	2.095936	53.275748	39936.762226	34.498858	3398.842752
std	2.154758	20.753928	28358.469482	26.438904	2187.396807
min	0.000000	18.000000	740.660000	1.000259	155.506715
25%	0.000000	35.000000	19285.522500	7.890442	1234.110529
50%	1.000000	53.000000	33186.785000	36.196030	3382.424000
75%	3.000000	71.000000	53472.395000	61.426670	5587.096500
max	10.000000	89.000000	258900.700000	71.999280	7158.982000

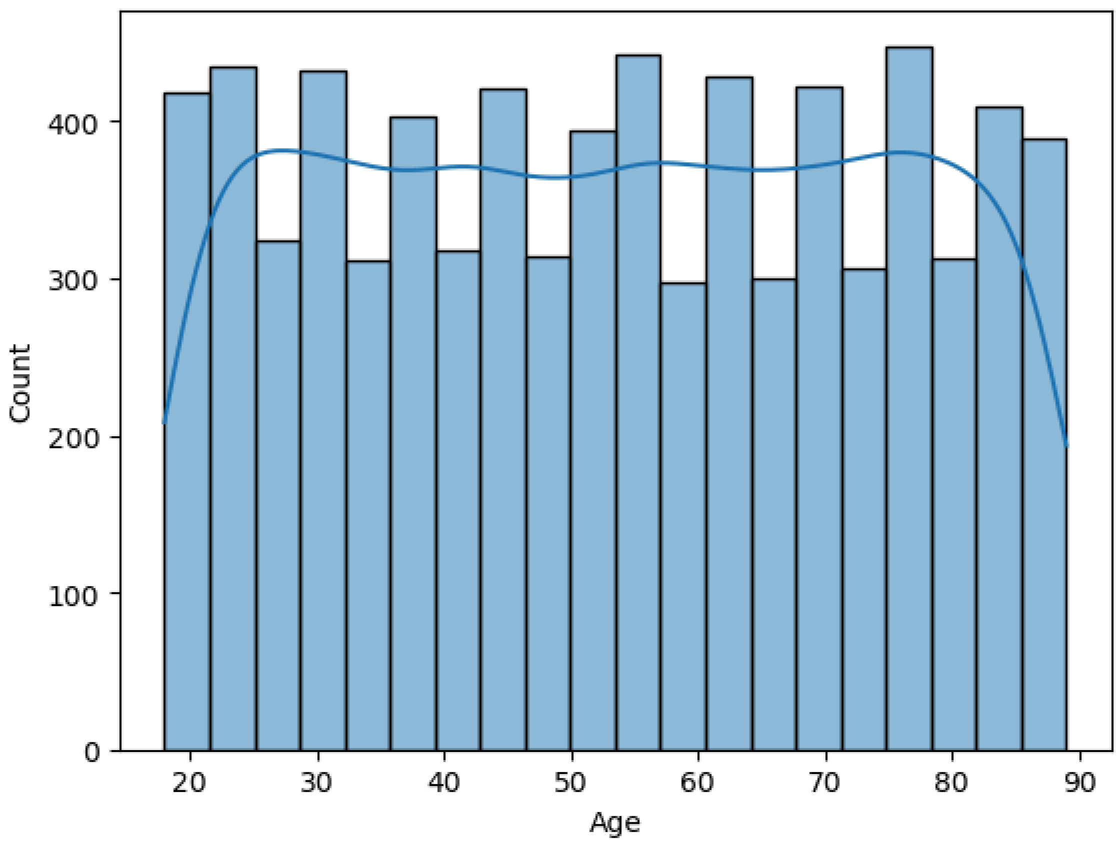
In [ ]:

```python
# histogram visualizations
for n, col in enumerate(churn[numeric_col]):
    try:
        plt.figure(n)
        sns.histplot(churn[col][churn[col] > 0], kde=True)
    except Exception:
        pass
```

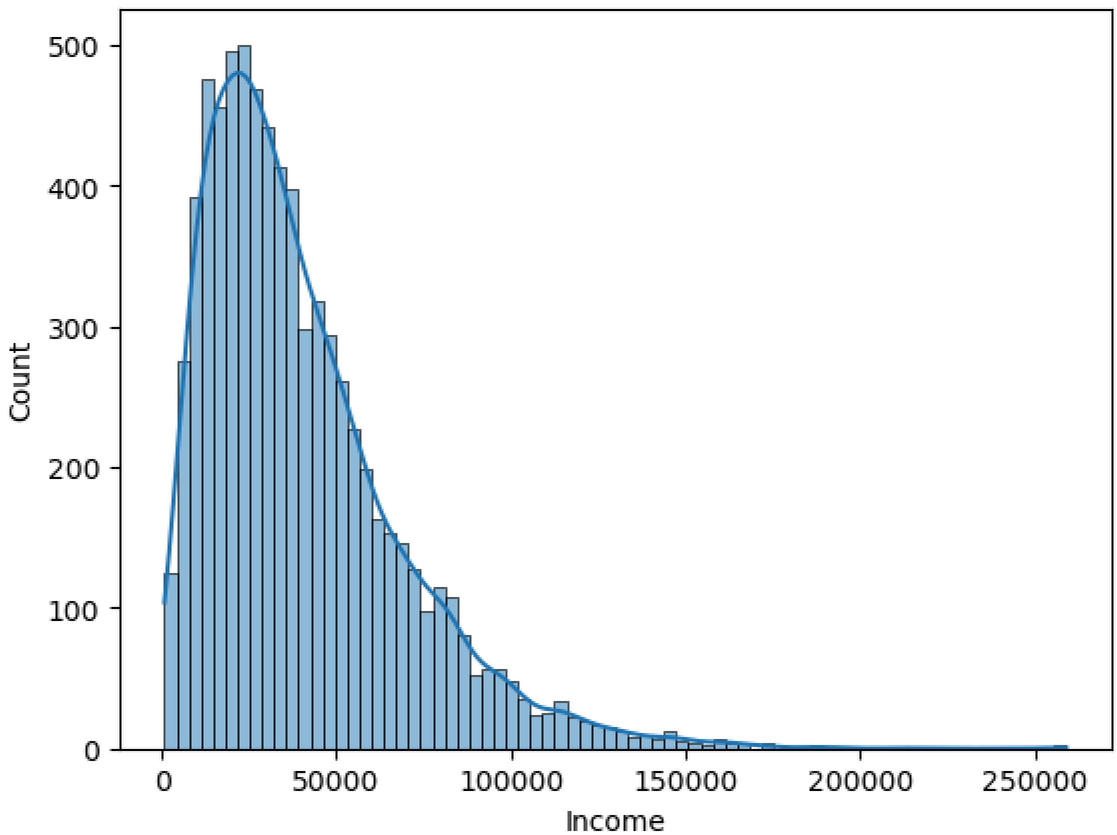
```
c:\Users\jjord\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
c:\Users\jjord\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
c:\Users\jjord\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
c:\Users\jjord\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
c:\Users\jjord\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

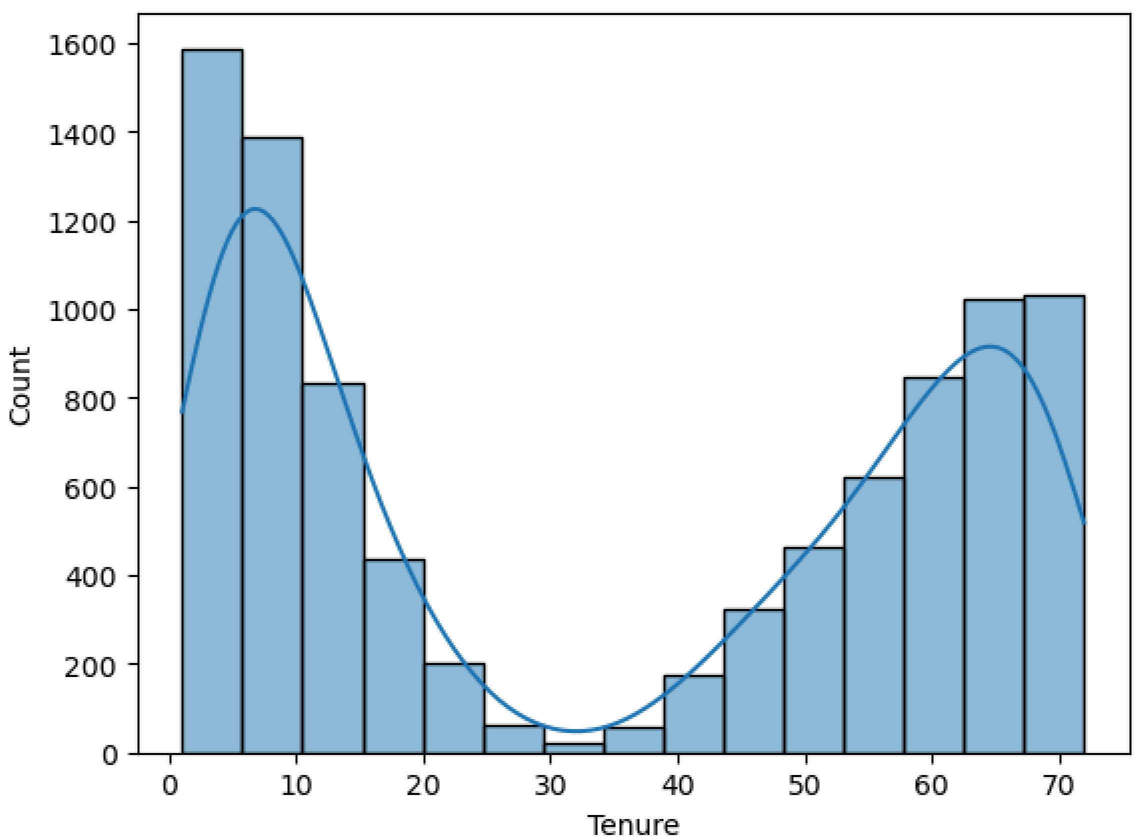


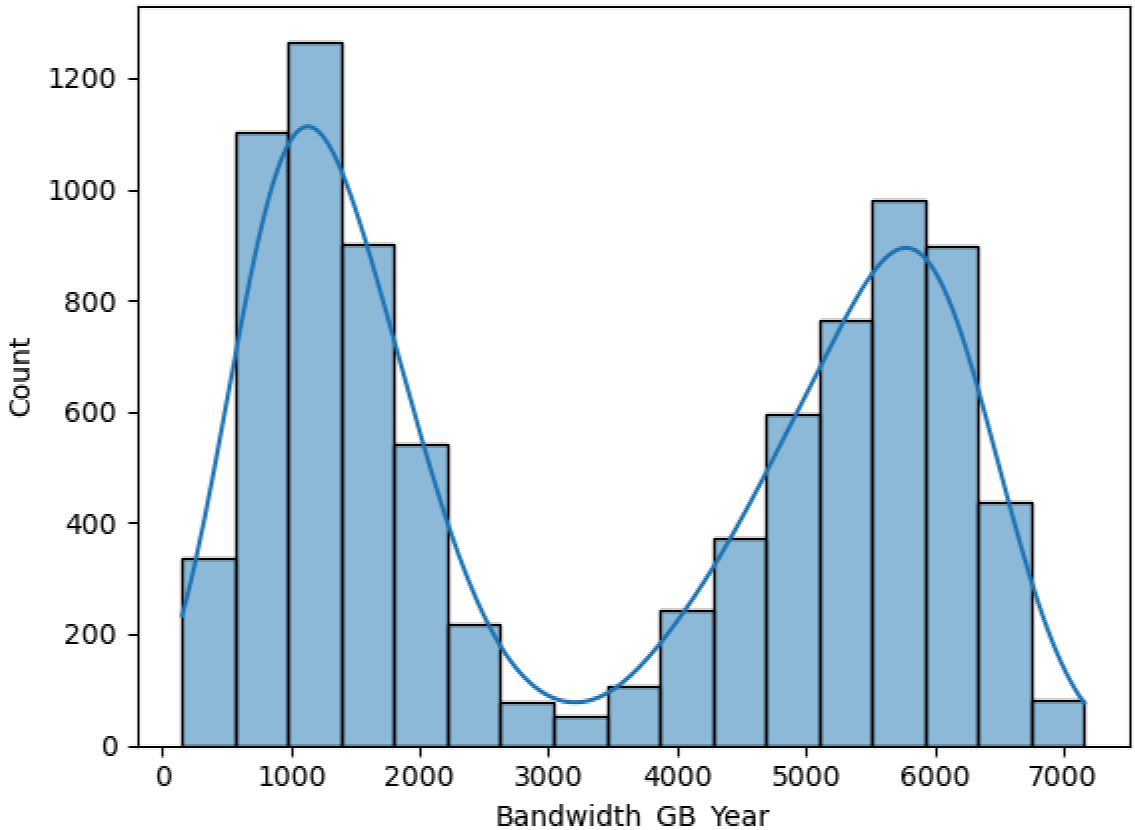
Count
Children
1750
1500
1250
1000
750
500
250
0
2
4
6
8
10



Count
Age
400
300
200
100
0
20
30
40
50
60
70
80
90


500
400
300
200
100
0
Count
0
50000
100000
150000
200000
250000
Income


1600
1400
1200
1000
800
600
400
200
0
Count
0
10
20
30
40
50
60
70
Tenure



Income and Children have distributions that are skewed normal. Skewed distributions should be imputed with the median function to reduce variance(Middleton, 2023)

In [ ]:
```
# create list for imputation
median_col = ["Income", "Children"]
# impute null values
for col in median_col:
    churn_clean[col].fillna(churn_clean[col].median(), inplace=True)
```

In [ ]:
```
# verify Income and Children no longer have null values
churn_clean[median_col].isnull().sum()
```

Out[ ]:
```
Income      0
Children    0
dtype: int64
```

Age has a distribution that is even. Since the size of the data is fairly large and there is no real way of knowing age, we will use random imputation.

In [ ]:
```
# set random seed
random.seed(36297)

# create list of current ages
ages_current = churn_clean["Age"][churn_clean["Age"].notna()].tolist()
```

In [ ]:
```
# asign ages at random
churn_clean["Age"].fillna(
```

```
    pd.Series(np.random.choice(ages_current, size=len(churn_clean.index))), inplace
)
```

In [ ]:
```
# verify Age no longer has nulls
churn_clean["Age"].isnull().sum()
```

Out[ ]:
```
0
```

Tenure and Bandwidth_GB_Year have a distribution that is bimodal. Since the size of the data is fairly large, we will use random imputation.

In [ ]:
```
# create list of Tenure and Bandwidth_GB_Year
bimod_col = ["Tenure", "Bandwidth_GB_Year"]
```

In [ ]:
```
for col in bimod_col:
    churn_clean[col].fillna(
        pd.Series(
            np.random.choice(
                churn_clean[col][churn_clean[col] >
                                0], size=len(churn_clean.index)
            )
        ),
        inplace=True,
    )
```

In [ ]:
```
# verify Tenure and Bandwidth_GB_Year no longer have null
churn_clean[bimod_col].isnull().sum()
```

Out[ ]:
```
Tenure               0
Bandwidth_GB_Year    0
dtype: int64
```

Categorical columns Techie, Phone and TechSupport will be set to 'No' as that is normally the default value. InternetService will be set to 'No Internet Service' instead of none or nan.

In [ ]:
```
# create list for Techie, Phone and TechSupport
catnulls = ["Techie", "Phone", "TechSupport"]
```

In [ ]:
```
# Impute nulls for Techie, Phone and TechSupport
for col in catnulls:
    churn_clean[col].fillna("No", inplace=True)
```

In [ ]:
```
# verify no nulls for Techie, Phone and TechSupport
churn_clean[catnulls].isnull().sum()
```

Out[ ]:
```
Techie         0
Phone          0
TechSupport    0
dtype: int64
```

In [ ]:
```
# Impute nulls for InternetService
churn_clean["InternetService"].fillna("No Internet Service", inplace=True)
```

In [ ]:
```
# verify InternetService no longer has nulls
churn_clean["InternetService"].isnull().sum()
```

Out[ ]: 0

In [ ]:
```
# export clean copy
churn_clean.to_csv("churn_clean.csv")
```

D. Continued

6. Summary of limitations of the data cleaning process.

   Most of the limitations arise from issues with the data source itself. Columns such as Education and income rely on the accuracy of the customer input which makes it difficult as human error comes into play. Also, its suggested that if columns such as these are going to be used for the purpose of analysis, the company should mark them as required. This will greatly reduce the number of null values within the data set. The process of data cleaning requires significant amounts of time and resources to properly identify missing values and clean data(Larose & Larose, 2019). There is no one-size-fits-all method for cleaning data since every data set is different. Imputation and the removal of outliers also have thier limitations. Imputation for replacing missing values can cause distortion or distribution issues within the data(Larose & Larose, 2019). Outlier removal can also significantly reduce the sample size of the data.

7. Discuss how the limitations summarized in part D6 could affect the analysis of the question or decision from part A.

   With the limitations discussed above, it is possible the analysis of the data set could lead to false predictions. It is also possible that accuracy could be questioned as well. Human error can play a large part of that due to so many columns being entered by the customer which could lead to more inconsistencies within the data. With a couple of the columns missing more than 20% of their data, it is possible that the imputation techniques introduced bias. All of these should be taken into consideration in regards to the question from part A.

PCA

1. PCA is based on an orthogonal mathematical concept that converts a set of observations of possibly correlated variables into a set called principal components(Middleton, 2023). It involves the process of finding linear combinations of variables taht best explain the connection among variables(Middleton, 2023). PCA can only be completed with quantitative continuous variables.

Below will be the process and code for PCA including normalization of the variables used. Variables used will include: ['Tenure', 'Contacts', 'Outage_sec_perweek', 'Yearly_equip_failure', 'MonthlyCharge', 'Bandwidth_GB_Year']

In [ ]:
```
# copy of clean data set
churn_scaled = churn_clean.copy()
```

In [ ]:
```
# columns for PCA
cc = churn_scaled[
    [
        "Tenure",
        "Contacts",
        "Outage_sec_perweek",
        "Yearly_equip_failure",
        "MonthlyCharge",
        "Bandwidth_GB_Year",
    ]
]
```

In [ ]:
```
# normalization
churn_norm = (cc - cc.mean()) / cc.std()
```

In [ ]:
```
pca = PCA(n_components=cc.shape[1])
```

In [ ]:
```
pca.fit(churn_norm)
```

Out[ ]:
```
▾ PCA
PCA(n_components=6)
```

In [ ]:
```
churn_pca = pd.DataFrame(
    pca.transform(churn_norm), columns=["PC1", "PC2", "PC3", "PC4", "PC5", "PC6"]
)
```

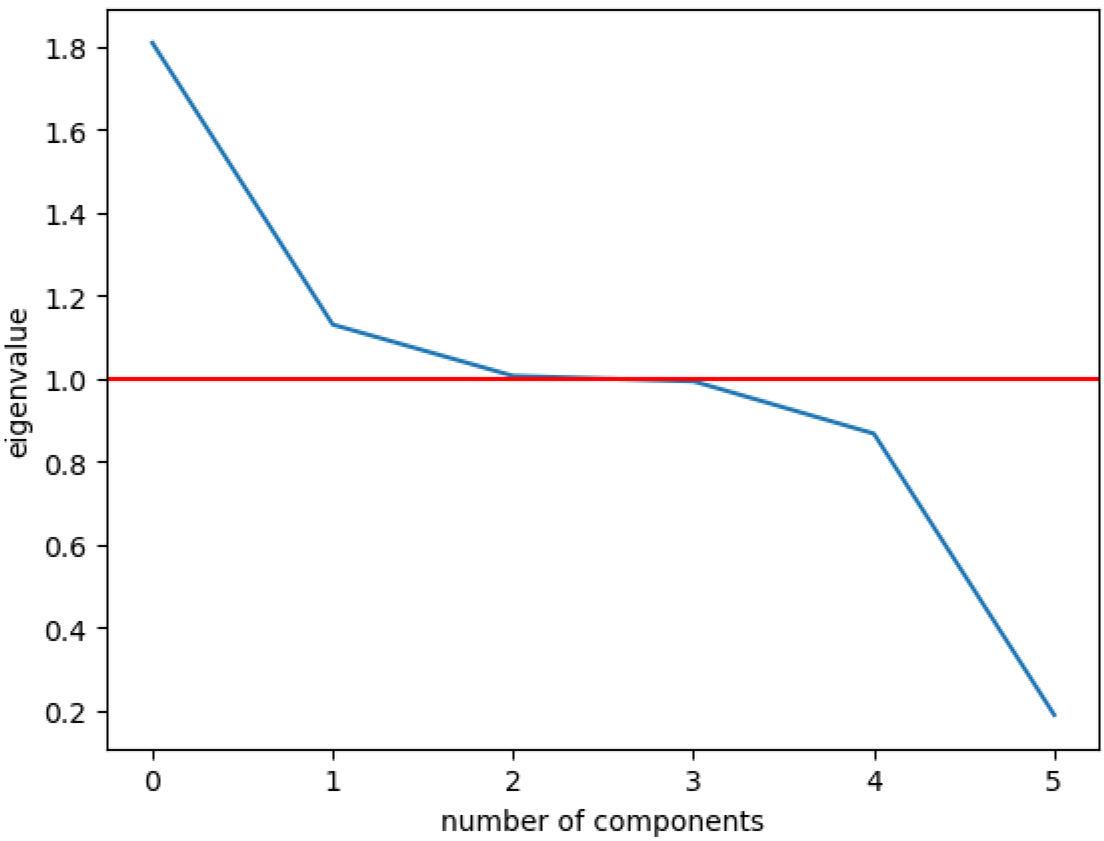
In [ ]:
```
# pca loadings
loadings = pd.DataFrame(
    pca.components_.T,
    columns=["PC1", "PC2", "PC3", "PC4", "PC5", "PC6"],
    index=cc.columns,
)
loadings
```

Out[ ]:

	PC1	PC2	PC3	PC4	PC5	PC6
Tenure	0.705093	-0.062149	0.001288	-0.004969	0.037895	0.705351
Contacts	-0.000792	0.004832	-0.677791	0.734910	0.020980	0.006506
Outage_sec_perweek	0.024033	0.706608	0.039253	0.011425	0.706014	0.000313
Yearly_equip_failure	0.013539	0.063653	0.727171	0.673532	-0.115497	0.001698
MonthlyCharge	0.047530	0.701897	-0.100839	-0.078115	-0.697257	0.051426
Bandwidth_GB_Year	0.706982	-0.010439	-0.010524	-0.002255	-0.012683	-0.706959

2. Justify the reduced number of the principal components and include a screenshot of a scree plot. Explanation of the reduced number of principal components will be below code and visualzation.

In [ ]:
```
# PC selection and visualization
cov_matrix = np.dot(churn_norm.T, churn_norm) / cc.shape[0]
eigenvalues = [
    np.dot(eigenvector.T, np.dot(cov_matrix, eigenvector))
    for eigenvector in pca.components_
]
plt.plot(eigenvalues)
plt.xlabel("number of components")
plt.ylabel("eigenvalue")
plt.axhline(y=1, color="red")
plt.show()
```



According to the Kaiser Rule, the number of components to retain from PCA is equal to the number of eigenvalues greater than one. An eigenvalue greater than one suggests that the corresponding principal component captures meaningful information from the data. The larger the PCA the more significant it is in explaining variations in the dataset(Lacrose & Lacrose, 2019). As we can see from the scree plot above PC's 0,1 & 2 are above the eigenvalue of 1 so they should be kept.

3. PCA Benefits

When PCA is performed, patterns in the data are identified and variables are grouped based on the relationship that exists among them(Middleton, 2023). PCA can also reduce the dimensionality of a data set making interpretation and visualization less complex. By making it less complex, it also reduces time and resources needed for analysis without sacrificing information in the variable.

Part IV. Supporting Documents

F. Panopto Video Link

- https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f031af29-be6a-488f-be64-b12b0021abef

Sources and References

Middleton, Dr. K. (2023). Getting Started with D206 Missing Values [Review of Getting Started with D206 Missing Values]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5cb19ab2-993f-4d04-af47-b07d011b3916

Middleton, Dr. K. (2023). Getting Started with D206 Data Cleaning Essentials [Review of Getting Started with D206 Data Cleaning Essentials]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=cf0de116-4cdf-4306-9284-b07d00ee8fcb

Middleton, Dr. K. (2023). D206 - Getting Started with D206 Outliers [Review of D206 - Getting Started with D206 Outliers]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=d2ce42e1-c2db-4e67-99bc-b07d00ee8bf0

Middleton, Dr. K. (2023). D206 - Getting Started with D206 Missing Values [Review of D206 - Getting Started with D206 Missing Values]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5cb19ab2-993f-4d04-af47-b07d011b3916

Middleton, Dr. K. (2023). Getting Started with D206 PCA [Review of Getting Started with D206 PCA]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=a371ff38-6c4f-403a-8018-b07d00ee8ba3

Larose, C. D., & Larose, D. T. (2019). Data science using Python and R. Wiley.