Performance Assessment D206 Data Cleaning

Ali Zaheer azaheer@wgu.edu

Part I: Research Question

A. Describe one question or decision that you will address using the data set you chose. The summarized question or decision must be relevant to a realistic organizational need or situation.

Which 'contract' type has high 'churn' and what type of correlation exists in respect to the customer's 'area'?

B. Describe the variables in the data set and indicate the specific type of data being described. Use examples from the data set that support your claims.

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

In [2]:
```
# Load data set
df = pd.read_csv('dataSet/churn_raw_data.csv')
```

In [3]:
```
# display data set
df.head()
```

Out[3]:

	Unnamed: 0	CaseOrder	Customer_id	Interaction	City	State	County	Zip	
0	1	1	K409198	aa90260b-4141-4a24-8e36-b04ce1f4f77b	Point Baker	AK	Prince of Wales-Hyder	99927	56.25
1	2	2	S120509	fb76459f-c047-4a9d-8af9-e0f7d4ac2524	West Branch	MI	Ogemaw	48661	44.32
2	3	3	K191035	344d114c-3736-4be5-98f7-c72c281e2d35	Yamhill	OR	Yamhill	97148	45.35
3	4	4	D90850	abfa2b40-2d43-4994-b15a-989b8c79e311	Del Mar	CA	San Diego	92014	32.96
4	5	5	K662701	68a861fd-0d20-4e51-a587-8a90407ee574	Needville	TX	Fort Bend	77461	29.38

5 rows × 52 columns

In [4]:
```
# Number of records in the data set
df.shape
```

Out[4]:
```
(10000, 52)
```

In [5]:
```
# Column names and their data types
df.dtypes
```

Out[5]:
```
Unnamed: 0               int64
CaseOrder                int64
Customer_id             object
Interaction             object
City                    object
State                   object
County                  object
Zip                      int64
Lat                    float64
Lng                    float64
Population               int64
Area                    object
Timezone                object
Job                     object
Children               float64
Age                    float64
Education               object
Employment              object
Income                 float64
Marital                 object
Gender                  object
Churn                   object
Outage_sec_perweek     float64
Email                    int64
Contacts                 int64
Yearly_equip_failure     int64
Techie                  object
Contract                object
Port_modem              object
Tablet                  object
InternetService         object
Phone                   object
Multiple                object
OnlineSecurity          object
OnlineBackup            object
DeviceProtection        object
TechSupport             object
StreamingTV             object
StreamingMovies         object
PaperlessBilling        object
PaymentMethod           object
Tenure                 float64
MonthlyCharge          float64
Bandwidth_GB_Year      float64
item1                    int64
item2                    int64
item3                    int64
item4                    int64
item5                    int64
item6                    int64
item7                    int64
item8                    int64
dtype: object
```

Part II: Data-Cleaning Plan

C. Explain the plan for cleaning the data by doing the following:

1. Propose a plan that includes the relevant techniques and specific steps needed to identify anomalies in the data set.

   A. Use Pandas to import the CSV file in the data frame.
   B. Examine and ensure data type consistency in the columns.
   C. Validate that each column has the same data type.
   D. Identify and resolve spelling mistakes in column headers or row level data.
   E. Identify and remove outliers
      - Outliers are identified using Z-score and boxplot graphs.
      - Validate if the outliers are to be removed or kept
   F. Identify, Standardize and replaced missing values using central tendency (Mean, Mode or Median)

   (Larose, 2019, p.29-43)

2. Justify your approach for assessing the quality of the data, include:

characteristics of the data being assessed:

There are 10,000 customer related records with 52 related variables in this data set. The 'Churn' column describes and defines whether the customer has cancelled their service(s) in last month.

Other variables that are related to each customer are categorically captured below:

- Services that each customer has signed up for (phone, multiple lines, internet, online security, online backup, device protection, technical support, and streaming TV and movies)
- Customer account related information (how long they've been a customer, contracts, payment methods, paperless billing, monthly charges, GB usage over a year, etc.)
- Customer demographics (gender, age, job, income, etc.)

Approach used to assess the quality:

- Validate each column to ensure its data is consistent with its data type.
- Identify and resolve spelling mistakes in column headers.
- Identify and remove outliers.
  - Outliers are identified using Z-score and/or boxplot graphs.
- Identify and replace missing values using central tendency (Median)

3. Justify your selected programming language and any libraries and packages that will support the data-cleaning process.

```
A.I will utilize Python due to my previous interaction with it
and its Pandas, matplotlib and Scipy modules. Additionally, I
will be using Jupyter notebook as the IDE because it provides a
user-friendly experience.
Pandas is an excellent package for working with data set as it
makes it easy to load and manipulate columns and/or rows to
replace null values.
Matplotlib plot is an easy way to create graphs for identifying
outliers using histograms and/or boxplot.
```

4. Provide the code you will use to identify the anomalies in the data.

In [6]:
```python
import pandas as pd
import numpy as np
from scipy import stats
```

```
%matplotlib inline
from sklearn.svm import OneClassSVM
from sklearn.preprocessing import scale
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
#from icecream import ic
```

In [7]:

```
# Load data set
df = pd.read_csv('dataSet/churn_raw_data.csv', dtype={'CaseOrder':np.int64})
```

In [8]:

```
# display data set with all the columns
pd.set_option('display.max_columns', None)
df.head(n=5)
```

Out[8]:

	Unnamed: 0	CaseOrder	Customer_id	Interaction	City	State	County	Zip	
0	1	1	K409198	aa90260b-4141-4a24-8e36-b04ce1f4f77b	Point Baker	AK	Prince of Wales-Hyder	99927	56.25
1	2	2	S120509	fb76459f-c047-4a9d-8af9-e0f7d4ac2524	West Branch	MI	Ogemaw	48661	44.32
2	3	3	K191035	344d114c-3736-4be5-98f7-c72c281e2d35	Yamhill	OR	Yamhill	97148	45.35
3	4	4	D90850	abfa2b40-2d43-4994-b15a-989b8c79e311	Del Mar	CA	San Diego	92014	32.96
4	5	5	K662701	68a861fd-0d20-4e51-a587-8a90407ee574	Needville	TX	Fort Bend	77461	29.38

In [9]:

```
# Number of records in the data set
df.shape
```

Out[9]:

```
(10000, 52)
```

In [10]:

```
# Column names and their data types
df.dtypes
```

Out[10]:

```
Unnamed: 0              int64
CaseOrder               int64
Customer_id            object
Interaction            object
City                   object
State                  object
```

```
County                   object
Zip                       int64
Lat                     float64
Lng                     float64
Population                int64
Area                     object
Timezone                 object
Job                      object
Children                float64
Age                     float64
Education                object
Employment               object
Income                  float64
Marital                  object
Gender                   object
Churn                    object
Outage_sec_perweek      float64
Email                     int64
Contacts                  int64
Yearly_equip_failure      int64
Techie                   object
Contract                 object
Port_modem               object
Tablet                   object
InternetService          object
Phone                    object
Multiple                 object
OnlineSecurity           object
OnlineBackup             object
DeviceProtection         object
TechSupport              object
StreamingTV              object
StreamingMovies          object
PaperlessBilling         object
PaymentMethod            object
Tenure                  float64
MonthlyCharge           float64
Bandwidth_GB_Year       float64
item1                     int64
item2                     int64
item3                     int64
item4                     int64
item5                     int64
item6                     int64
item7                     int64
item8                     int64
dtype: object
```

In [11]:

```
# Remove column with no headers
df = df.drop(df.columns[[0]], axis=1)
```

In [12]:

```
# Amend columns with no names
df = df.rename(columns=({ 'item1': 'Timely response', 'item2':'Timely fixes', '
                         'item4':'Reliability', 'item5':'Options', 'item6':'Res
                         'item7':'Courteous exchange', 'item8':'Evidence of act
```

Identify spelling mistakes in the rows

In [14]:

```
# Review unique data in Area column
df['Area'].unique()
```

Out[14]:
```
array(['Urban', 'Suburban', 'Rural'], dtype=object)
```

In [15]:
```
# Review unique data in Employment column
df['Employment'].unique()
```

Out[15]:
```
array(['Part Time', 'Retired', 'Student', 'Full Time', 'Unemployed'],
      dtype=object)
```

In [16]:
```
# Review unique data in Gender column
df['Gender'].unique()
```

Out[16]:
```
array(['Male', 'Female', 'Prefer not to answer'], dtype=object)
```

In [17]:
```
# Review unique data in Marital column
df['Marital'].unique()
```

Out[17]:
```
array(['Widowed', 'Married', 'Separated', 'Never Married', 'Divorced'],
      dtype=object)
```

In [18]:
```
# Review unique data in PaymentMethod column
df['PaymentMethod'].unique()
```

Out[18]:
```
array(['Credit Card (automatic)', 'Bank Transfer(automatic)',
       'Mailed Check', 'Electronic Check'], dtype=object)
```

In [19]:
```
# Review unique data in InternetService column
df['InternetService'].unique()
```

Out[19]:
```
array(['Fiber Optic', 'DSL', 'None'], dtype=object)
```

In [20]:
```
# Review unique data in Job column
df['Job'].unique()
```

Out[20]:
```
array(['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'],
      dtype=object)
```

Reexpression of categorical data as numerical data

Education

In [21]:
```
# Capture unique values from the 'Education' column for Re-Expression
df['Education'].unique().tolist()
```

Out[21]:
```
["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']
```

In [22]:
```
# Re-expression categorial data in 'Education' columns
dict_edu= {'Education': {
     'No Schooling Completed': 0,
     'Nursery School to 8th Grade': 8,
     '9th Grade to 12th Grade, No Diploma':11,
     'Regular High School Diploma': 12,
     'GED or Alternative Credential': 12,
     'Some College, Less than 1 Year': 12,
     'Some College, 1 or More Years, No Degree': 12,
     'Professional School Degree':13,
     "Associate's Degree": 14,
     "Bachelor's Degree": 16,
     "Master's Degree": 18,
     'Doctorate Degree': 20,
}}
```

In [23]:
```
# Apply the Reexpression values
df.replace(dict_edu, inplace = True)
```

In [24]:
```
# display data set with Re-Expressioned 'Education' column
df.head()
```

Out[24]:

	CaseOrder	Customer_id	Interaction	City	State	County	Zip	Lat	
0	1	K409198	aa90260b-4141-4a24-8e36-b04ce1f4f77b	Point Baker	AK	Prince of Wales-Hyder	99927	56.25100	-133.37
1	2	S120509	fb76459f-c047-4a9d-8af9-e0f7d4ac2524	West Branch	MI	Ogemaw	48661	44.32893	-84.24

	CaseOrder	Customer_id	Interaction	City	State	County	Zip	Lat	
2	3	K191035	344d114c-3736-4be5-98f7-c72c281e2d35	Yamhill	OR	Yamhill	97148	45.35589	-123.24
3	4	D90850	abfa2b40-2d43-4994-b15a-989b8c79e311	Del Mar	CA	San Diego	92014	32.96687	-117.24
4	5	K662701	68a861fd-0d20-4e51-a587-8a90407ee574	Needville	TX	Fort Bend	77461	29.38012	-95.80

Identify Missing Values

In [25]:

```
# Identify and isolate the columns with null
df.loc[:,df.isnull().any()]
```

Out[25]:

	Children	Age	Income	Techie	Phone	TechSupport	Tenure	Bandwidth_GB_Year
0	NaN	68.0	28561.99	No	Yes	No	6.795513	904.536110
1	1.0	27.0	21704.77	Yes	Yes	No	1.156681	800.982766
2	4.0	50.0	NaN	Yes	Yes	No	15.754144	2054.706961
3	1.0	48.0	18925.23	Yes	Yes	No	17.087227	2164.579412
4	0.0	83.0	40074.19	No	No	Yes	1.670972	271.493436
...	...	...	...	...	...	...	...	...
9995	3.0	NaN	55723.74	NaN	NaN	No	68.197130	6511.253000
9996	4.0	48.0	NaN	NaN	NaN	No	61.040370	5695.952000
9997	NaN	NaN	NaN	No	Yes	No	NaN	4159.306000
9998	1.0	39.0	16667.58	No	No	Yes	71.095600	6468.457000
9999	1.0	28.0	NaN	NaN	Yes	No	63.350860	5857.586000

10000 rows × 8 columns

In [26]:

```
# Count of missing values per columns
df.isna().sum()
```

Out[26]:

```
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                 0
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
Timely response                 0
Timely fixes                    0
Timely replacements             0
Reliability                     0
Options                         0
Respectful response             0
Courteous exchange              0
Evidence of active listening    0
dtype: int64
```

Change Misleading Field Values

Limitations: replacing missing value can cause the data set to be inflated as I am trying to impose what could be the accurate value

- Children: Customer might have chosen not to tell the actual number of children they have due to privacy concerns
- Phone: Customer might have chosen not to list their phone number due to privacy concerns.
- Techie: This could have been left out a human error.
- TechSupport: This could a human error, someone might not have entered appropriate values assuming 'No' and '' are the same.

In [27]:

```
# Replace the NAN in Childern column with 0 as it already has 0 value for peopl
df['Children']=df['Children'].replace({np.NaN:0})
```

In [28]:

```
# Replace the NAN in Phone column with No, as either a person has a phone or th
df['Phone']=df['Phone'].replace({np.NaN:"No"})
```

In [29]:
```
# Replace the NAN in Techie column with No
df['Techie']=df['Techie'].replace({np.NaN:"No"})
```

In [30]:
```
# Replace the NAN in TechSupport column with No
df['TechSupport']=df['TechSupport'].replace({np.NaN:"No"})
```

Identify Missing Numeric Values

In [31]:
```
# Identify missing data in Age column
df["Age"].isnull().sum()
```

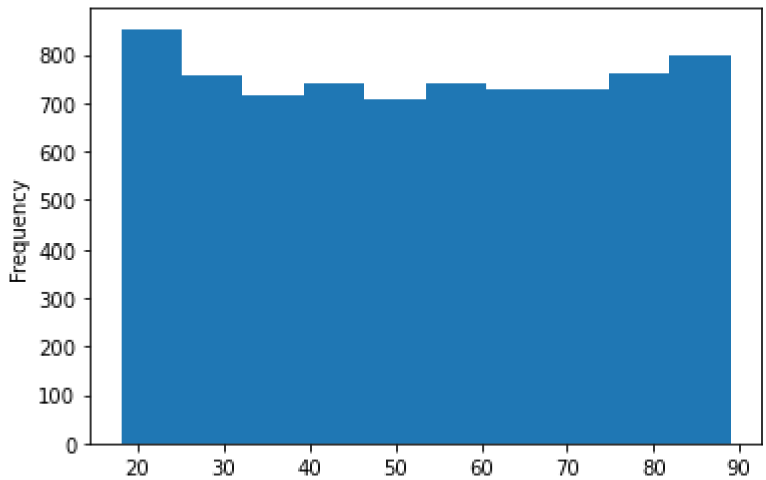
Out[31]:
```
2475
```

In [32]:
```
#list out all values including null
df["Age"]
```

Out[32]:
```
0       68.0
1       27.0
2       50.0
3       48.0
4       83.0
        ...
9995     NaN
9996    48.0
9997     NaN
9998    39.0
9999    28.0
Name: Age, Length: 10000, dtype: float64
```

In [33]:
```
#Plot Age distribution
df["Age"].plot.hist()
```

Out[33]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [34]:
```
# Identify missing data in Income column
df["Income"].isnull().sum()
```

Out[34]:
```
2490
```

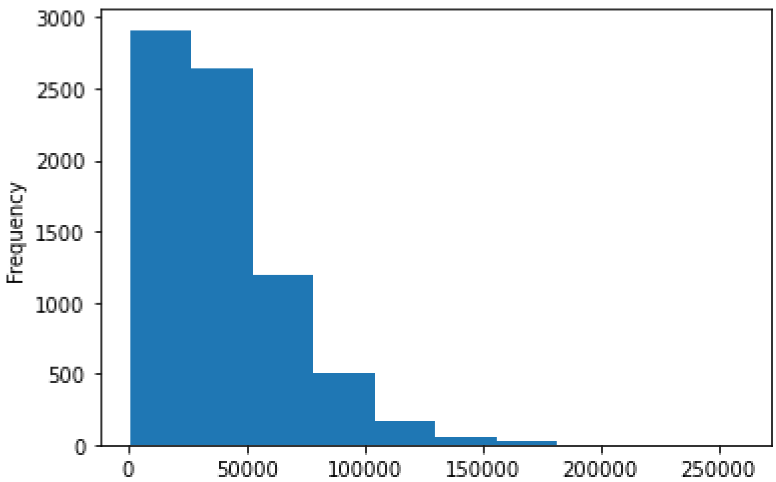
In [35]:
```
#list out all values including null
```

```
df["Income"]
```

Out[35]:
```
0       28561.99
1       21704.77
2            NaN
3       18925.23
4       40074.19
          ...
9995    55723.74
9996         NaN
9997         NaN
9998    16667.58
9999         NaN
Name: Income, Length: 10000, dtype: float64
```

In [36]:
```
#Plot Income distribution
df["Income"].plot.hist()
```

Out[36]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [37]:
```
# Identify missing data in Tenure column
df["Tenure"].isnull().sum()
```

Out[37]:
```
931
```

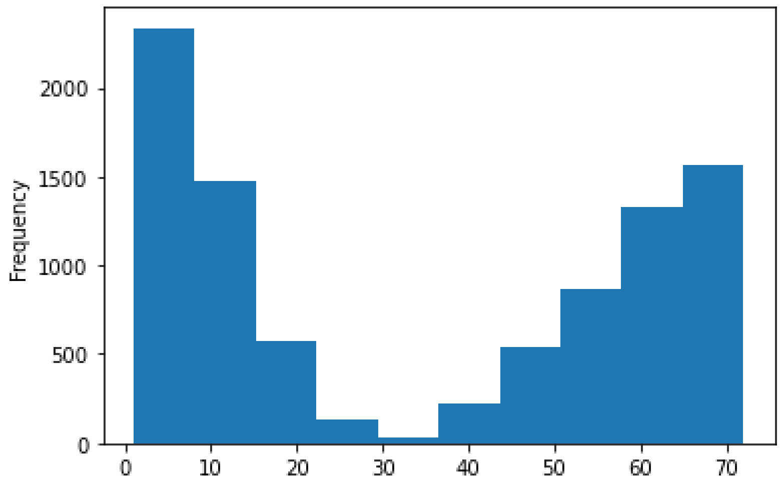
In [38]:
```
# Identify missing data in Tenure column
df["Tenure"]
```

Out[38]:
```
0        6.795513
1        1.156681
2       15.754144
3       17.087227
4        1.670972
          ...
9995    68.197130
9996    61.040370
9997          NaN
9998    71.095600
9999    63.350860
Name: Tenure, Length: 10000, dtype: float64
```

In [39]:
```
#Plot Tenure distribution
df["Tenure"].plot.hist()
```

```
<AxesSubplot:ylabel='Frequency'>
```

Out[39]:



In [40]:
```
# Identify missing data in Bandwidth_GB_Year column
df["Bandwidth_GB_Year"].isnull().sum()
```

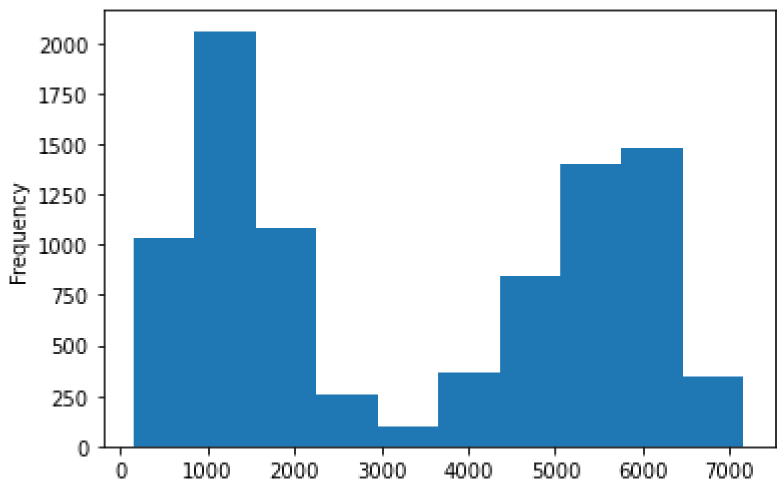
Out[40]:
```
1021
```

In [41]:
```
# Identify missing data in Bandwidth_GB_Year column
df["Bandwidth_GB_Year"]
```

Out[41]:
```
0        904.536110
1        800.982766
2       2054.706961
3       2164.579412
4        271.493436
            ...
9995    6511.253000
9996    5695.952000
9997    4159.306000
9998    6468.457000
9999    5857.586000
Name: Bandwidth_GB_Year, Length: 10000, dtype: float64
```

In [42]:
```
#Plot Bandwidth_GB_Year distribution
df["Bandwidth_GB_Year"].plot.hist()
```

Out[42]:
```
<AxesSubplot:ylabel='Frequency'>
```



Replace Missing Numeric Values with Median becuase the distrution of

data is skewed as displayed above.

This is a robust measure that is not strongly influenced by the outliers

In [43]:
```
# Fill in the NAN in age with median
df["Age"].fillna(df["Age"].median(), inplace=True)
```

In [44]:
```
# Fill in the NAN in income with median
df["Income"].fillna(df["Income"].median(), inplace=True)
```

In [45]:
```
#Fill in the NAN in Tenure with median
df["Tenure"].fillna(df["Tenure"].median(), inplace=True)
```

In [46]:
```
# Fill in the NAN in Bandwidth_GB_Year with median
df["Bandwidth_GB_Year"].fillna(df["Bandwidth_GB_Year"].median(), inplace=True)
```

In [47]:
```
# Validate all the null values have been replaced
df.isnull().any()
```

Out[47]:
```
CaseOrder                False
Customer_id              False
Interaction              False
City                     False
State                    False
County                   False
Zip                      False
Lat                      False
Lng                      False
Population               False
Area                     False
Timezone                 False
Job                      False
Children                 False
Age                      False
Education                False
Employment               False
Income                   False
Marital                  False
Gender                   False
Churn                    False
Outage_sec_perweek       False
Email                    False
Contacts                 False
Yearly_equip_failure     False
Techie                   False
Contract                 False
Port_modem               False
Tablet                   False
InternetService          False
Phone                    False
Multiple                 False
OnlineSecurity           False
OnlineBackup             False
DeviceProtection         False
TechSupport              False
StreamingTV              False
StreamingMovies          False
PaperlessBilling         False
PaymentMethod            False
```

```
Tenure                            False
MonthlyCharge                     False
Bandwidth_GB_Year                 False
Timely response                   False
Timely fixes                      False
Timely replacements               False
Reliability                       False
Options                           False
Respectful response               False
Courteous exchange                False
Evidence of active listening      False
dtype: bool
```

Cleaned data set

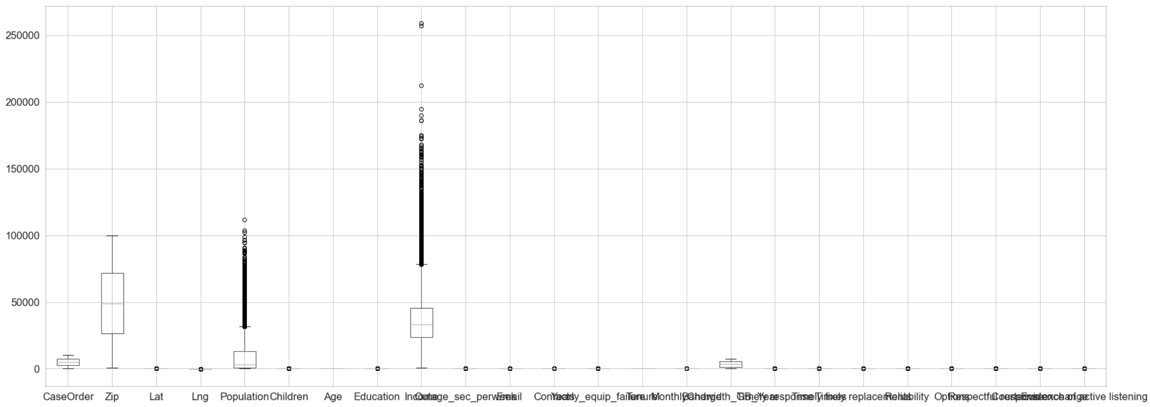
In [48]:
```python
# Cleaned data set
df.to_csv('Cleaned_Data_set.csv')
```

Outliers

In [49]:
```python
# Change sns settings
sns.set(rc={'figure.figsize':(30,11)}, font_scale=1.5, style='whitegrid')
```

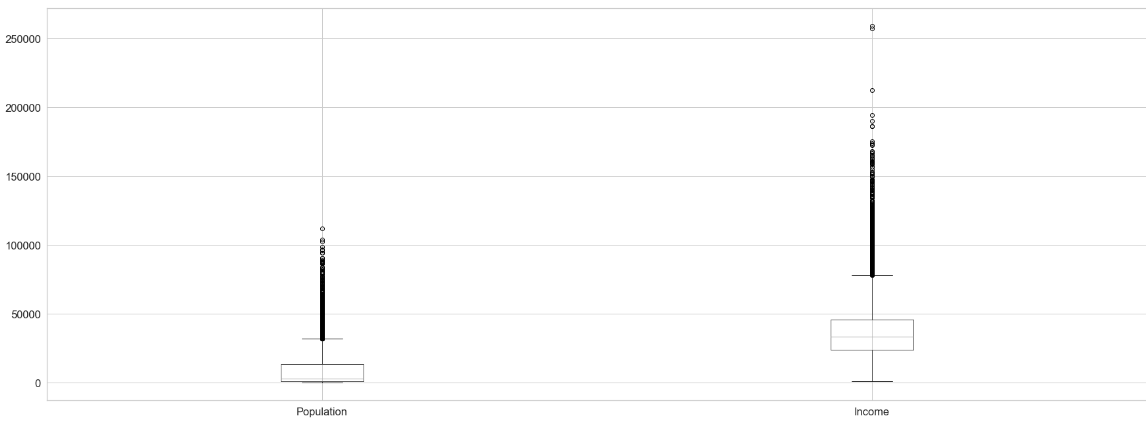
In [50]:
```python
# Quick look to see which columns have outliers
df.boxplot()
```

Out[50]: <AxesSubplot:>



In [51]:
```python
# Box plot of all the columns with outliers
df.boxplot(['Population', 'Income'])
```

Out[51]: <AxesSubplot:>

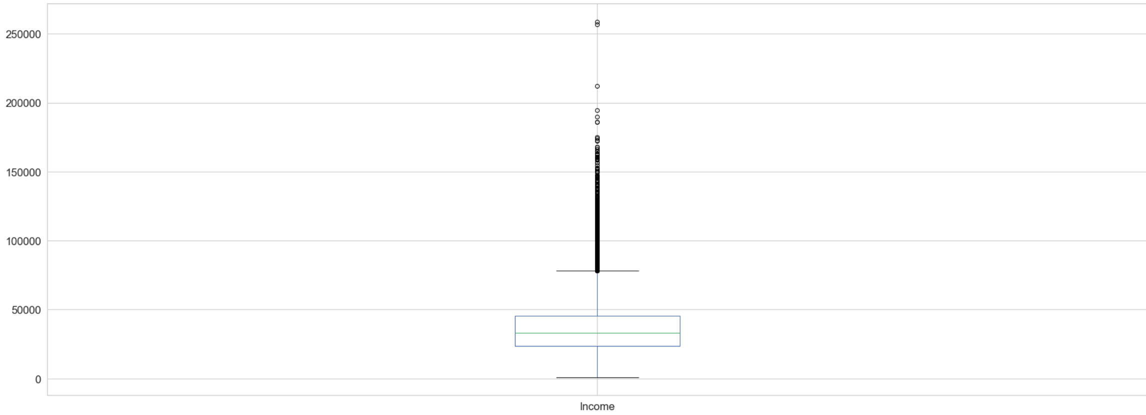


Investigate Outliers in the Income column

In [52]:

```
# Using box plot plot to identify outliers
Income = df['Income']
Income.plot.box()
```

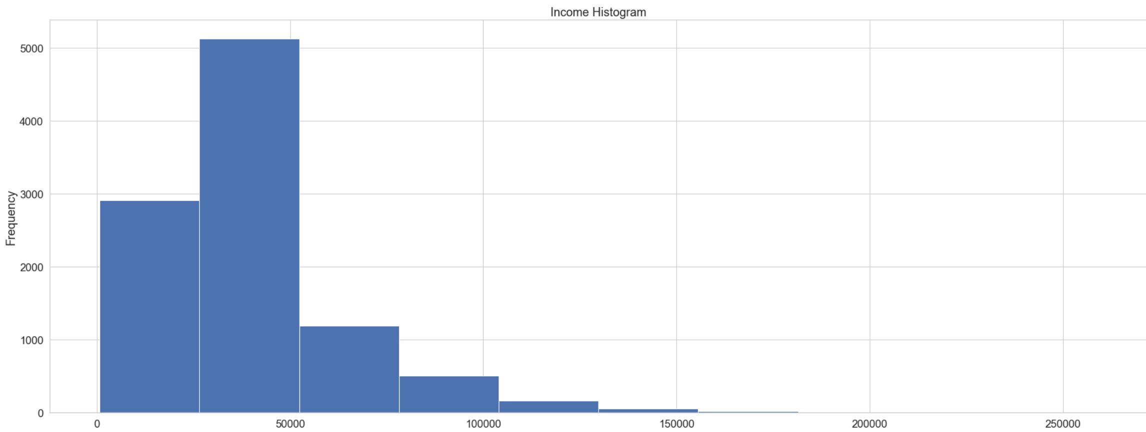
Out[52]: <AxesSubplot:>



In [53]:

```
# Investigate distribution of Income column using histogram
df["Income"].plot(kind = "hist", title = 'Income Histogram')
```

Out[53]: <AxesSubplot:title={'center':'Income Histogram'}, ylabel='Frequency'>



In [54]:

```
# Create a new column with standarized Income values
df["Income_z"] = stats.zscore(df["Income"])
```

In [56]:

```
# Based on the z score isolate the outliers
df_income_outliers = df.query('Income_z > 3 | Income_z < -3')
```

In [57]:

```
# Create a new data set for the outliers and sort it in descending order
df_income_outliers_sort_values = df_income_outliers.sort_values(['Income_z'], a
```

In [58]:

```
# List out the outliers
df_income_outliers_sort_values['Income'].head()
```

Out[58]:

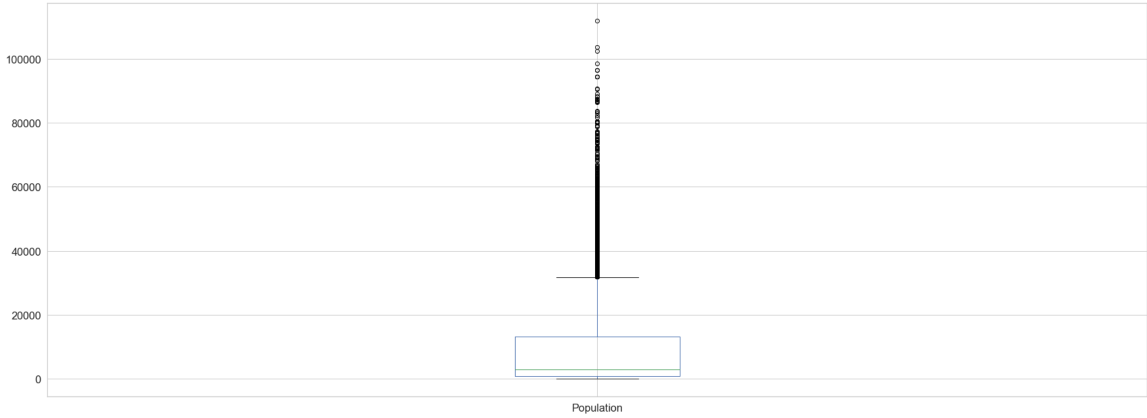
```
4249    258900.7
9180    256998.4
5801    212255.3
6837    194550.7
```

```
3985    189938.4
Name: Income, dtype: float64
```

Investigate Outliers in the Population column

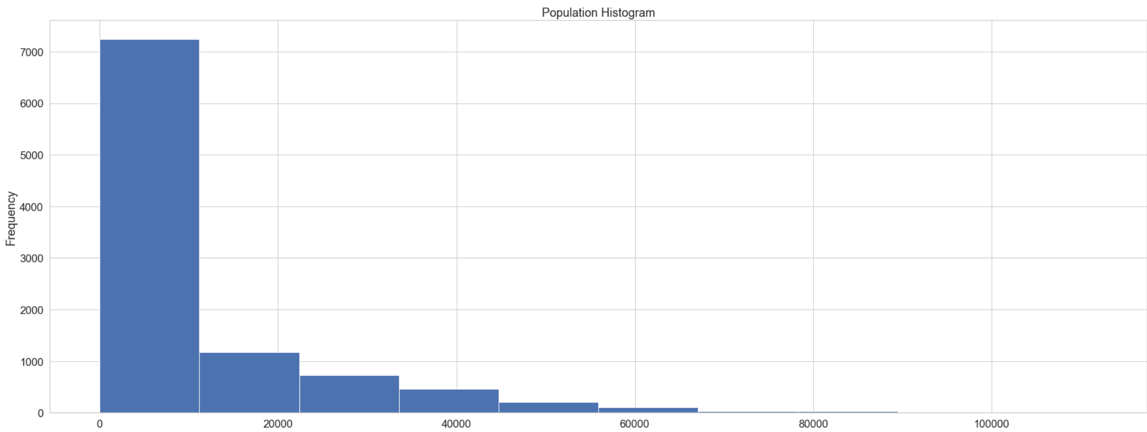
In [59]:
```
# Using box plot plot to identify outliers
Population = df['Population']
Population.plot.box()
```

Out[59]:
```
<AxesSubplot:>
```



In [60]:
```
# Investigate distribution of Population column using histogram
df["Population"].plot(kind = "hist", title = 'Population Histogram')
```

Out[60]:
```
<AxesSubplot:title={'center':'Population Histogram'}, ylabel='Frequency'>
```



In [61]:
```
# Create a new column with standarized median values
df["Population_z"] = stats.zscore(df["Population"])
```

In [62]:
```
# Based on the z score isolate the outliers
df_Population_outliers = df.query('Population_z > 3 | Population_z < -3')
```

In [63]:
```
# Create a new data set for the outliers and sort it in descending order
df_Population_outliers_sort_values = df_Population_outliers.sort_values(['Popul
```

In [64]:
```
# List out the outliers
df_Population_outliers_sort_values['Population'].head()
```

Out[64]:
```
8139    111850
8320    103732
```

```
6288    102433
1775     98660
6610     96575
Name: Population, dtype: int64
```

PCA Analysis

In [65]:

```
# Load data frame
df = pd.read_csv('dataSet/churn_raw_data.csv', index_col=0)
```

In [66]:

```
# Quick view of the data-set
df.head()
```

Out[66]:

	CaseOrder	Customer_id	Interaction	City	State	County	Zip	Lat	
1	1	K409198	aa90260b-4141-4a24-8e36-b04ce1f4f77b	Point Baker	AK	Prince of Wales-Hyder	99927	56.25100	-133.37
2	2	S120509	fb76459f-c047-4a9d-8af9-e0f7d4ac2524	West Branch	MI	Ogemaw	48661	44.32893	-84.24
3	3	K191035	344d114c-3736-4be5-98f7-c72c281e2d35	Yamhill	OR	Yamhill	97148	45.35589	-123.24
4	4	D90850	abfa2b40-2d43-4994-b15a-989b8c79e311	Del Mar	CA	San Diego	92014	32.96687	-117.24
5	5	K662701	68a861fd-0d20-4e51-a587-8a90407ee574	Needville	TX	Fort Bend	77461	29.38012	-95.80

In [67]:

```
# Add names to the customer feedback columns
df = df.rename(columns=({ 'item1': 'Timely response', 'item2':'Timely fixes', '
                         'item4':'Reliability', 'item5':'Options', 'item6':'Res
                         'item7':'Courteous exchange', 'item8':'Evidence of act
```

In [68]:

```
# Create PCA analysis data-set with feedback response
customer_data = df[['Timely response', 'Timely fixes', 'Timely replacements', '
                         'Courteous exchange','Evidence of active listening']]
```

In [69]:

```
# Normalize the data frame
customer_data_norm = (customer_data-customer_data.mean())/customer_data.std()
```

In [70]:

```
# Component extraction
pca= PCA(n_components=customer_data.shape[1])
```

In [72]:

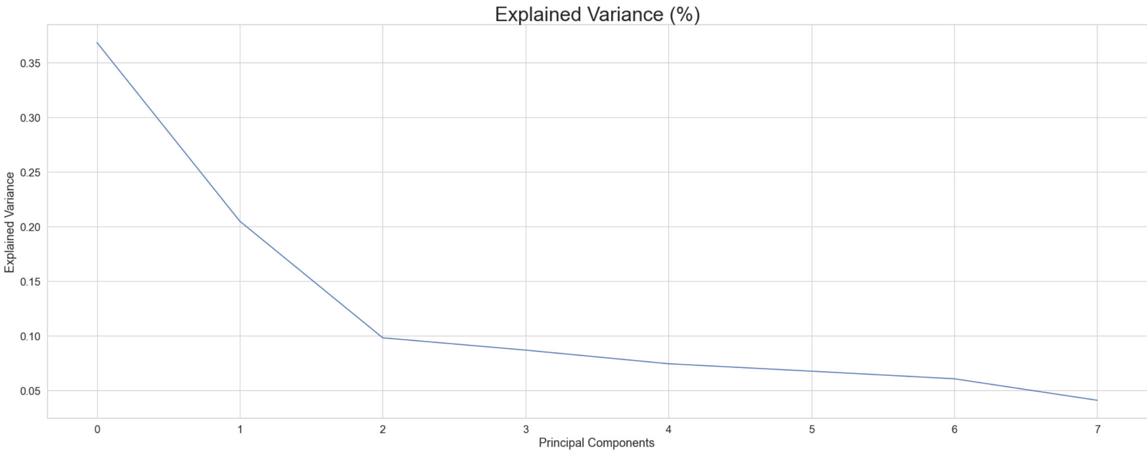
```
# PCA fitting
pca.fit(customer_data_norm)
```

Out[72]: PCA(n_components=8)

In [73]:
```
# PCA transform and normalization
customer_pca = pd.DataFrame(pca.transform(customer_data_norm))
```

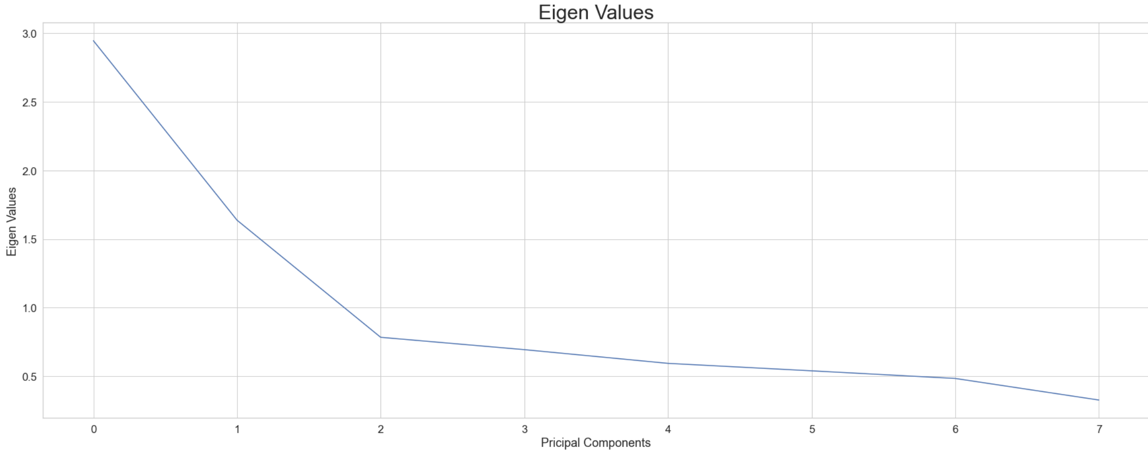
In [74]:
```
# Principle Component for the Scree plot
columns = ['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8']
```

In [75]:
```
# Scree plot showing the PCs
# Below show the 60 percent of the variance is explained by 2 component
plt.plot(pca.explained_variance_ratio_)
plt.xlabel('Principal Components')
plt.ylabel('Explained Variance')
plt.title('Explained Variance (%)',  fontsize=30)
plt.show()
```



In [76]:
```
# Eigenvalues
cov_matrix = np.dot(customer_data_norm.T, customer_data_norm) / customer_data.s
EigenV = [np.dot(eigenvector.T, np.dot(cov_matrix, eigenvector)) for eigenvecto
```

In [77]:
```
# Scree plot show Eigen Values
# PC0 and PC1 has Eigenvalues greated than 1.
plt.plot(EigenV)
plt.xlabel('Pricipal Components')
plt.ylabel('Eigen Values')
plt.title('Eigen Values',  fontsize=30)
plt.show()
```



In [78]:

```
# Loading and identifying the PC from the Customer dataframe
loading = pd.DataFrame(pca.components_.T, columns = ['PC1','PC2','PC3','PC4','P
loading
```

Out[78]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	
Timely response	0.459030	0.282717	-0.069221	0.120013	-0.044752	0.025032	-0.241688	0.79
Timely fixes	0.434342	0.285321	-0.106259	0.170235	-0.064900	0.074672	-0.591586	-0.57
Timely replacements	0.400775	0.282950	-0.173885	0.254689	-0.148134	-0.396761	0.673403	-0.17
Reliability	0.145686	-0.569898	-0.171525	0.482754	-0.444692	0.431115	0.086961	0.01
Options	-0.175385	0.591292	0.135315	-0.060906	-0.211030	0.693537	0.265272	-0.04
Respectful response	0.405197	-0.183897	-0.061988	-0.063712	0.757170	0.403694	0.231751	-0.06
Courteous exchange	0.358413	-0.181067	-0.184917	-0.806749	-0.378391	0.067449	0.066043	-0.04
Evidence of active listening	0.308851	-0.132624	0.931619	0.009229	-0.114326	-0.044789	0.046267	-0.04

In [79]:

```
# Isolate and show values of the PC1
load = loading['PC1'] > .4
loading[load]['PC1']
```

Out[79]:

```
Timely response          0.459030
Timely fixes             0.434342
Timely replacements      0.400775
Respectful response      0.405197
Name: PC1, dtype: float64
```

Part III: Data Cleaning

D. Summarize the data-cleaning process by doing the following:

D1. I was able to find 8 columns with anomalies. Children, Phone, Techie and TechSupport were categorical, and their ‘null’ values were replaced with ‘No’. I used the 'Reexpression of Categorial column' to create the Education column. The limitations are as follow:

D2. Categorical data imputation limitation can distort the data if the assumptions are not confirmed.

- Children: The customer might have chosen not to tell the actual number of children they have due to privacy concerns.
- Phone: Customer might have chosen not to list their phone number due to privacy concerns.
- Techie: This could have been left out as a human error.
- TechSupport: This could a human error, someone might not have entered appropriate values assuming 'No' and '' are the same.

D2a. Numerical data The continuous type columns (Age, Income, Tenure, Bandwidth_GB_Year) data were replaced using python's median functions because these are continuous and the data was either skewed to left or Bimodal. I chose this because it is simple, easy to apply method and does not reduce the sample size. on the limitation side, it is possible to distort data / distribution of the data. The rest of the columns were not part of the process as they did not have any null values.

D3. All the missing categorical values were imputed to 'No' and numerical values were imputed using median central tendency. Age and Tenure were left alone as they did not have any outliers.

D4. Code is available above and in the Panopto recording

D5. Attached file 'Cleaned_Data_set.csv')

D6 & D7. The data cleaning process assumes that replacing categorical null values with 'No' is the right approach however this can lead to inflated data that will lean toward replaced values and can lead to inaccurate decision making. Similarly, using statical central tendencies is an appropriate approach but can lead to inflated data and imbalanced decision making. Imputing data values using the above steps can give us a picture but cannot replace true values which were missed due to human error/system errors.

E. Apply principal component analysis (PCA) to identify the significant features of the data set by doing the following:

1. List the principal components in the data set.

- Timely response
- Timely fixes
- Timely replacements
- Respectful response

1. Describe how you identified the principal components of the data set.

- PC0 and PC1 should be kept as they have Eigenvalues greater than 1.

1. Describe how the organization can benefit from the results of the PCA

- The four identified scores should be reviewed carefully to understand customer's feedback. This will help the company to keep their customer for a longer time hence increasing profits.

Part IV. Supporting Documents

F. Provide a Panopto recording that demonstrates the warning- and error-free functionality of the code used to support the discovery of anomalies and the data cleaning process and summarizes the programming environment.

Note: For instructions on how to access and use Panopto, use the "Panopto How-To Videos" web link provided below. To access Panopto's website, navigate to the web link titled "Panopto Access", and then choose to log in using the “WGU” option. If prompted, log in using your WGU student portal credentials, and then it will forward you to Panopto’s website.

To submit your recording, upload it to the Panopto drop box titled “Data Cleaning – NUM2 \ D206” Once the recording has been uploaded and processed in Panopto's system, retrieve the URL of the recording from Panopto and copy and paste it into the Links option. Upload the remaining task requirements using the Attachments option.

G. Reference the web sources used to acquire segments of third-party code to support the application. Be sure the web sources are reliable.


```
{bibliography}
Pandas. (2021). Pandas DataFrames.
https://pandas.pydata.org/pandas-
docs/stable/reference/api/pandas.DataFrame.dtypes.html

Get started with references. (2021). Jupyterbook.
https://jupyterbook.org/tutorials/references.html#tutorials-
references

Marques, A. M. (2020, March 11). How to show all columns / rows
of a Pandas Dataframe? Towards Data Science.
https://towardsdatascience.com/how-to-show-all-columns-rows-of-a-
pandas-dataframe-c49d4507fcf
```


H. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.


```
{bibliography}
Chantal D. Larose, & Daniel T. Larose. (2019). Data Science Using
Python and R. Wiley.
```