D206 – Data Cleaning

Task 1

William Stults

02/16/2022

Table of Contents

Contents

Research Question

Description

My dataset for this data cleaning exercise includes data on an internet service provider's current and former subscribers, with an emphasis on customer churn (whether customers are maintaining or discontinuing their subscription to the ISP's service). Data preparation performed on the dataset will be aimed with this research question in mind: is there a relationship between customer lifestyle, or "social" factors, and customer churn? Lifestyle and social factors might include variables such as age, number of children, and level of education, among others.

Description of Variables

The table below describes the variables included in the customer churn dataset.

Variable	**Data Type**	**Quantitative / Qualitative**	**Description**
CaseOrder	int64	Qualitative	Numerical ID for each row in the dataset
Customer_id	object	Qualitative	Customer ID number
Interaction	object	Qualitative	A unique identifier relevant to customer transactions
City	object	Qualitative	City of residence per billing info
State	object	Qualitative	State of residence per billing info
County	object	Qualitative	County of residence per billing info
Zip	int64	Qualitative	Zip code of residence per billing info
Lat	float64	Qualitative	GPS coordinate of the customer's residence

Lng	float64	Qualitative	GPS coordinate of the customer’s residence
Population	int64	Quantitative	Population within 1 mile of customer
Area	object	Qualitative	Type of area customer lives in (urban, suburban, rural)
Timezone	object	Qualitative	Time zone the customer resides in
Job	object	Qualitative	Customer’s occupation
Children	float64	Quantitative	How many children live in the customer’s household
Age	float64	Quantitative	Customer’s age
Education	object	Qualitative	Highest education level of customer
Employment	object	Qualitative	Status of employment
Income	float64		Customer’s income annually
Marital	object	Qualitative	Customer’s marital status
Gender	object	Qualitative	Self-identified gender of the customer
Churn	object	Qualitative	Yes/No if customer canceled service
Outage_sec_perweek	float64	Quantitative	Amount of time in seconds that outages were experienced in the area of the customer’s residence
Email	int64	Quantitative	Number of emails the customer was sent over the past year
Contacts	int64	Quantitative	How many times technical support was contacted by the customer
Yearly_equip_failure	int64	Quantitative	Number of times in the past year the customer’s equipment had to be replaced or reset
Techie	object	Qualitative	Yes/No does the customer feel they are technically inclined, based on a questionnaire
Contract	object	Qualitative	Length of the contract term the customer has purchased
Port_modem	object	Qualitative	Yes/No is the customer’s modem portable

Tablet	object	Qualitative	Yes/No does the customer own a tablet
InternetService	object	Qualitative	Type of internet service (cable, DSL, fiber, etc.)
Phone	object	Qualitative	Yes/No does the customer have phone service
Multiple	object	Qualitative	Yes/No does the customer have multiple lines
OnlineSecurity	object	Qualitative	Yes/No does the customer have an add-on for online security
OnlineBackup	object	Qualitative	Yes/No does the customer have an add-on for online backup
DeviceProtection	object	Qualitative	Yes/No does the customer have an add-on for device protection
TechSupport	object	Qualitative	Yes/No does the customer have an add-on for tech support
StreamingTV	object	Qualitative	Yes/No does the customer have streaming tv service
StreamingMovies	object	Qualitative	Yes/No does the customer have streaming movies service
PaperlessBilling	object	Qualitative	Yes/No does the customer have paperless billing
PaymentMethod	object	Qualitative	Type of payment method the customer uses
Tenure	float64	Quantitative	How long in months the customer has subscribed
MonthlyCharge	float64	Quantitative	How much the customer is charged monthly
Bandwidth_GB_Year	float64	Quantitative	Average data used per year by the customer in GB
item1	int64	Qualitative	Survey response, scale of 1-8, timely response
item2	int64	Qualitative	Survey response, scale of 1-8, timely fixes

item3	int64	Qualitative	Survey response, scale of 1-8, timely replacement
item4	int64	Qualitative	Survey response, scale of 1-8, reliability
item5	int64	Qualitative	Survey response, scale of 1-8, options
item6	int64	Qualitative	Survey response, scale of 1-8, respectful response
item7	int64	Qualitative	Survey response, scale of 1-8, courteous exchange
item8	int64	Qualitative	Survey response, scale of 1-8, active listening

Data Cleaning Plan

Plan Overview

My first steps taken toward cleaning my data set will involve evaluating the quality of the data. Specifically, actions will be taken to identify duplicate rows, null data, and outliers. This will help me identify any values that are in need of being manipulated or wholly excluded to improve the health of the data set for analysis purposes. Functions utilized for detecting duplicate rows in the dataset are duplicated() and sum(). The functions being used for the detection of missing or null values will be info(), isnull(), and sum(). For detecting outliers, I will initially use the z-scoring method by way of the stats.zscore() function from the scipy library. I can then further analyze which numerical variables may need treatment by using the hist() function from the matplotlib library and the boxplot() function from seaborn. As I will only be evaluating quantitative data for outliers, numerical variables that are qualitative (geographical information, survey response, etc.) will be excluded.

Justification of Approach

By using the info() function on the data frame, I gain an overview of variable names, data types, and how many non-null values exist for each variable. This particular dataset is comprised of data types object (categorical variables) int64, and float64 (numerical variables). By further using the isnull(), and sum() functions on the data frame I am given a list of variables along with the total number of missing or null values for that variable within the dataset. The benefit of using a z-scoring method for the detection of outliers is that it gives you a numerical representation of how far a particular value differs from the mean value of all values for that variable, making the process of identifying variables for which outliers may exist much more efficient. By viewing these z-score values in a histogram using the hist() function, I will be able to identify which variables exhibit a higher degree of variance, meaning the likelihood of outliers existing for that variable would be higher. Boxplots can then further illustrate visually how far potential outlier values fall outside of a variable's interquartile range.

Justification of Language and Libraries

I will be using Python as my selected programming language due to prior familiarity and broader applications when considering programming in general. R is a very strong and robust language tool for data analysis and statistics but finds itself somewhat limited to that niche role (Insights for Professionals, 2019). I will be utilizing the Numpy, Pandas, and Matplotlib libraries to perform many of my data cleaning tasks, as they are among the most popular Python libraries employed for this purpose and see widespread use, and Seaborn is included primarily for its better-looking boxplots (Parra, 2021). Beyond these libraries, SciPy builds upon NumPy, incorporating NumPy's many mathematical functions and including numerous others (Johari, 2019). It will be used here primarily for its stats module. Lastly, Scikit-learn, or sklearn, is valued for its statistical modeling capabilities and will be utilized for principal component analysis (Jain, 2015).

Code

```
##Observe general information about the dataset
df.info()

##Print data frame
print (df)
```

```
##Detect number of duplicate rows in the dataset

df.duplicated().sum()


##Observe specifically which variables in the dataset contain missing
values

df.isnull().sum()


##Generate histograms to observe variable distributions

plt.hist(df['Children'])

plt.hist(df['Age'])

plt.hist(df['Income'])

plt.hist(df['Tenure'])

plt.hist(df['Bandwidth_GB_Year'])


##Using len function to determine number of zscores

##Exceeding 3 standard deviations from the mean

len(df.query('Children_z > 3 | Children_z < -3'))

len(df.query('Age_z > 3 | Age_z < -3'))

len(df.query('Income_z > 3 | Income_z < -3'))

len(df.query('Outage_sec_perweek_z > 3 | Outage_sec_perweek_z < -3'))

len(df.query('Email_z > 3 | Email_z < -3'))

len(df.query('Contacts_z > 3 | Contacts_z < -3'))

len(df.query('Yearly_equip_failure_z > 3 | Yearly_equip_failure_z < -
3'))

len(df.query('Tenure_z > 3 | Tenure_z < -3'))

len(df.query('MonthlyCharge_z > 3 | MonthlyCharge_z < -3'))

len(df.query('Bandwidth_GB_Year_z > 3 | Bandwidth_GB_Year_z < -3'))


##Generate histograms from the new zscore columns

##to analyze zscore distributions

plt.hist(df['Children_z'])
plt.hist(df['Age_z'])
plt.hist(df['Income_z'])
plt.hist(df['Outage_sec_perweek_z'])
plt.hist(df['Email_z'])
plt.hist(df['Contacts_z'])
plt.hist(df['Yearly_equip_failure_z'])
plt.hist(df['Tenure_z'])
plt.hist(df['MonthlyCharge_z'])
plt.hist(df['Bandwidth_GB_Year_z'])


##Generate boxplots to visually illustrate variable
##values for variables exhibiting abnormal zscore distributions
boxplot=seaborn.boxplot(x='MonthlyCharge',data=df)
boxplot=seaborn.boxplot(x='Yearly_equip_failure',data=df)
boxplot=seaborn.boxplot(x='Email',data=df)
boxplot=seaborn.boxplot(x='Income',data=df)
boxplot=seaborn.boxplot(x='Children',data=df)
boxplot=seaborn.boxplot(x='Contacts',data=df)
boxplot=seaborn.boxplot(x='Outage_sec_perweek',data=df)
```

Data Cleaning

Findings

No duplicate rows were detected in the dataset.

Numerical variables found to contain missing or null values were as follows:

Variable Name	Number of Missing or Null Values
Children	2495
Age	2475
Income	2490
Tenure	931
Bandwidth_GB_Year	1021

Categorical variables found to contain missing or null values were as follows:

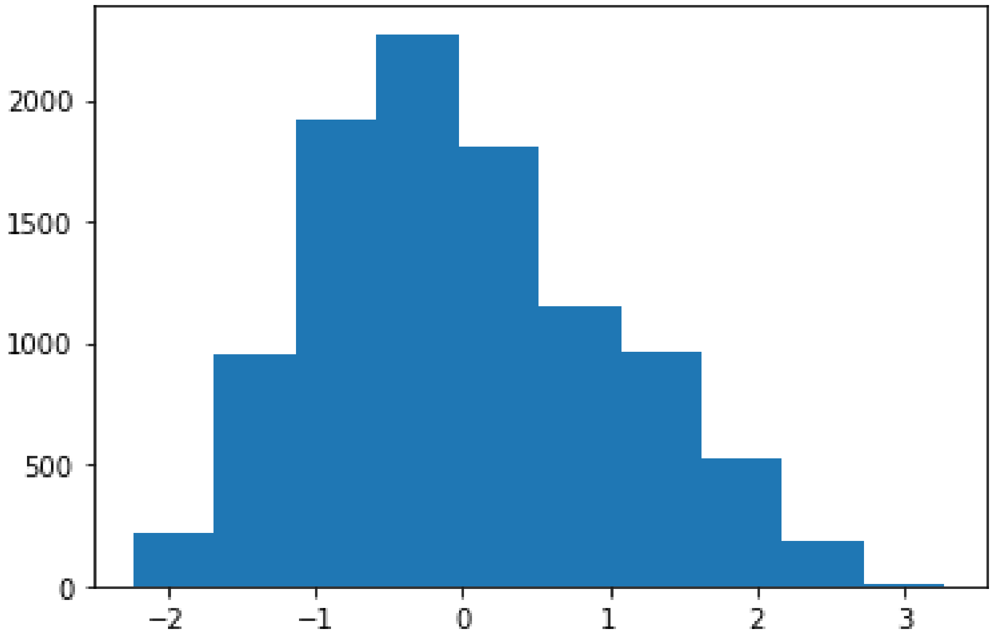
Variable Name	Number of Missing or Null Values
Techie	2477
Phone	1026
TechSupport	991

Variables containing values with a zscore greater than 3 or less than -3:

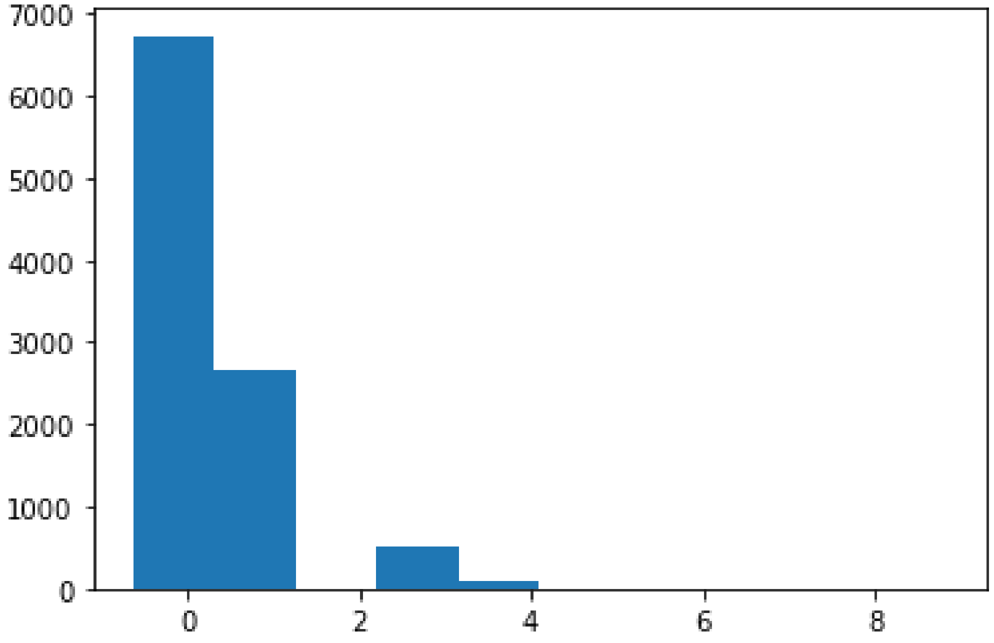
Variable Name	Number of Potential Outliers
MonthlyCharge	3
Yearly_equip_failure	94
Email	12
Income	193
Children	302
Contacts	165
Outage_sec_perweek	491

Histogram outputs showing distribution for zscored variables containing values with a zscore greater than 3 or less than -3:

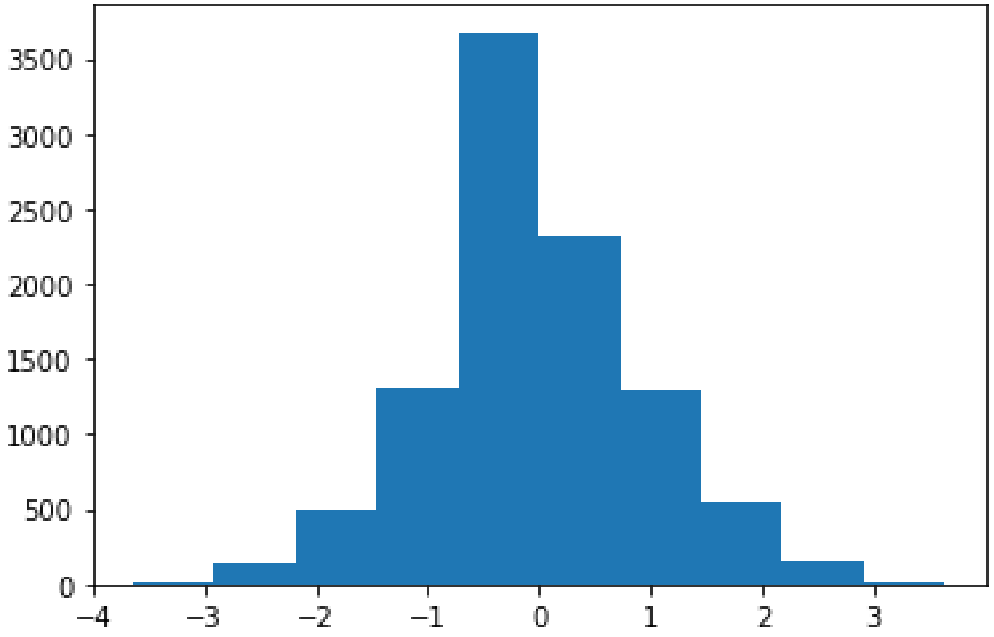
MonthlyCharge:



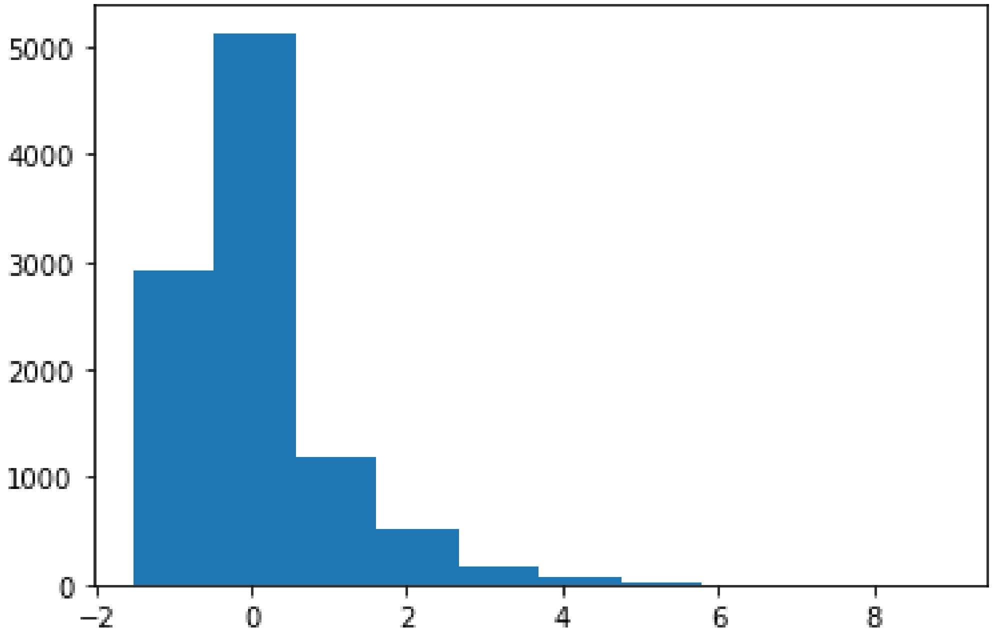
Yearly_equip_failure:



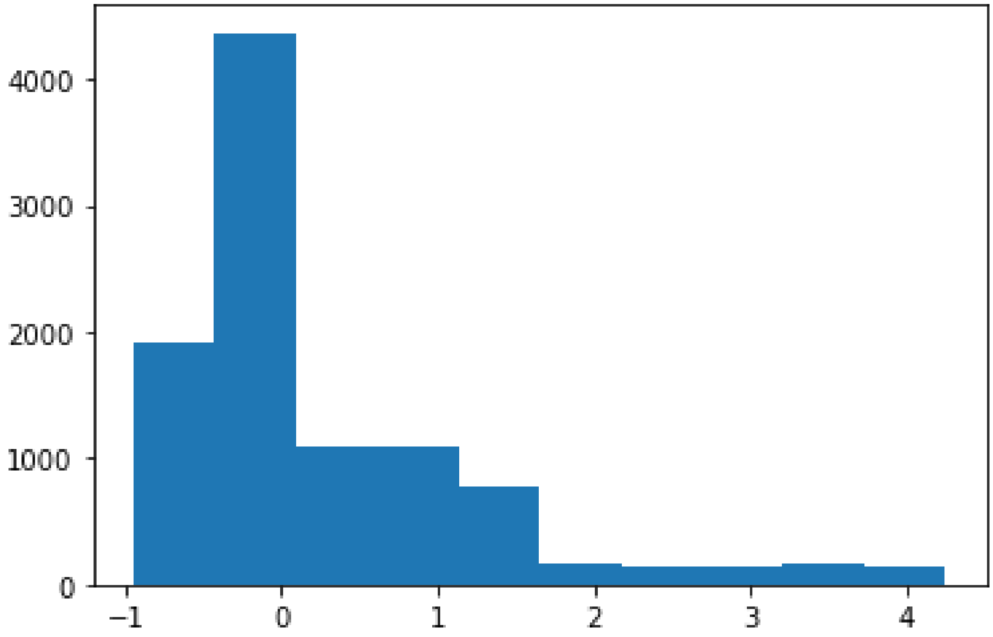
Email:



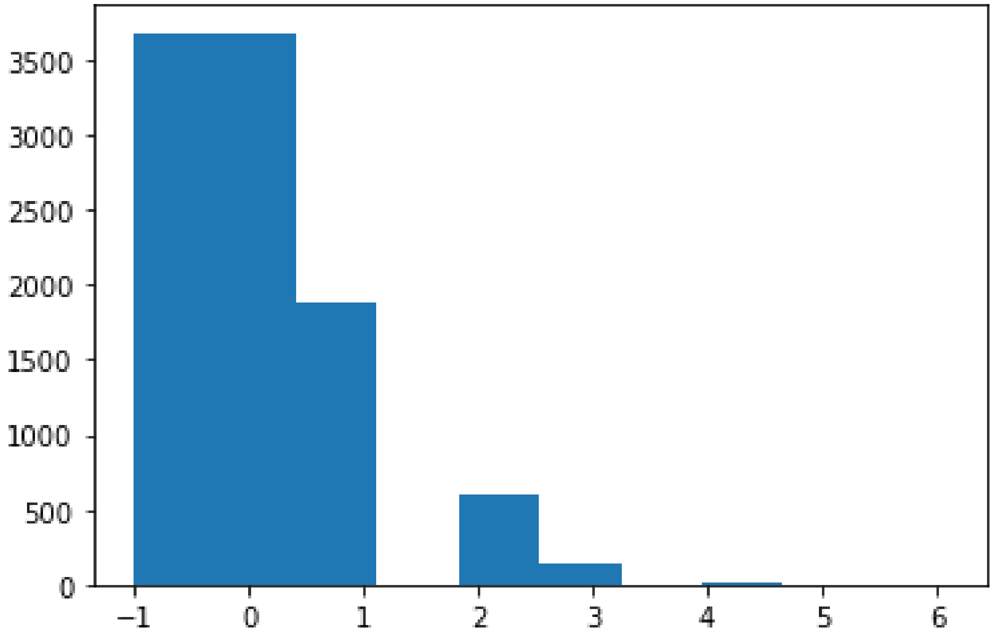
Income:



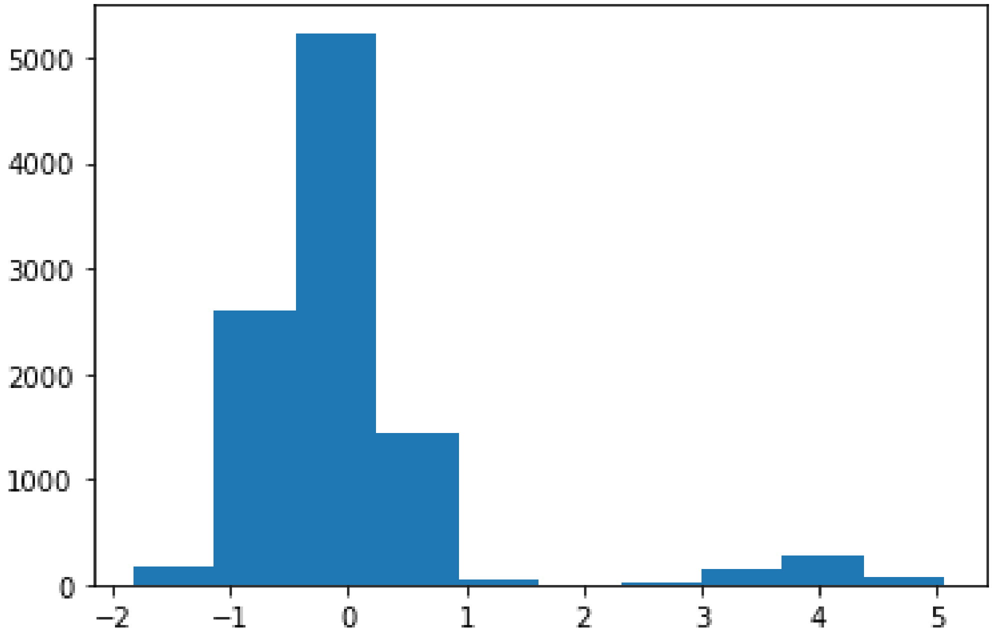
Children:



Contacts:

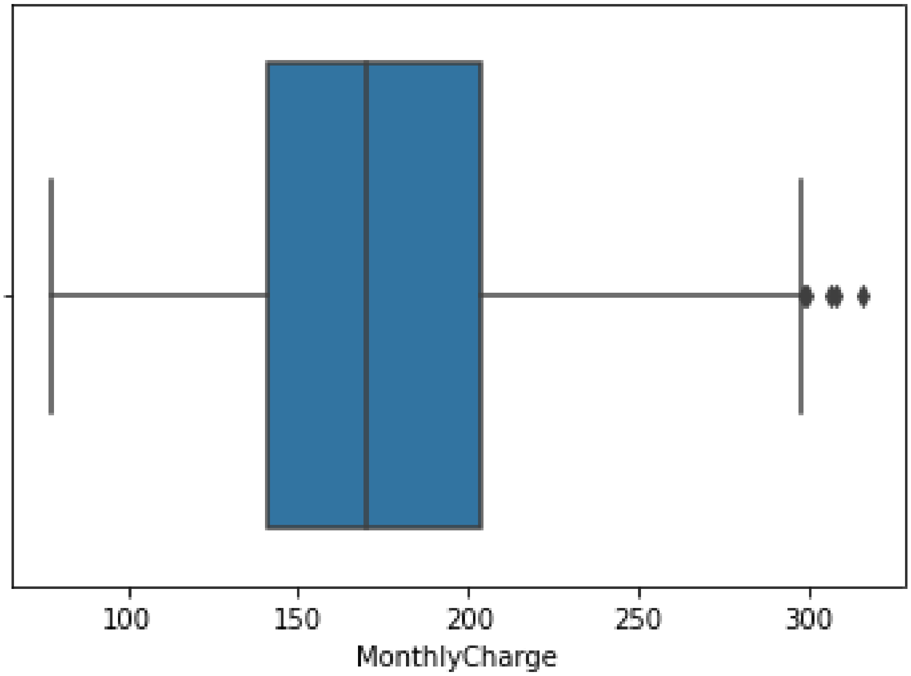


Outage_sec_perweek:

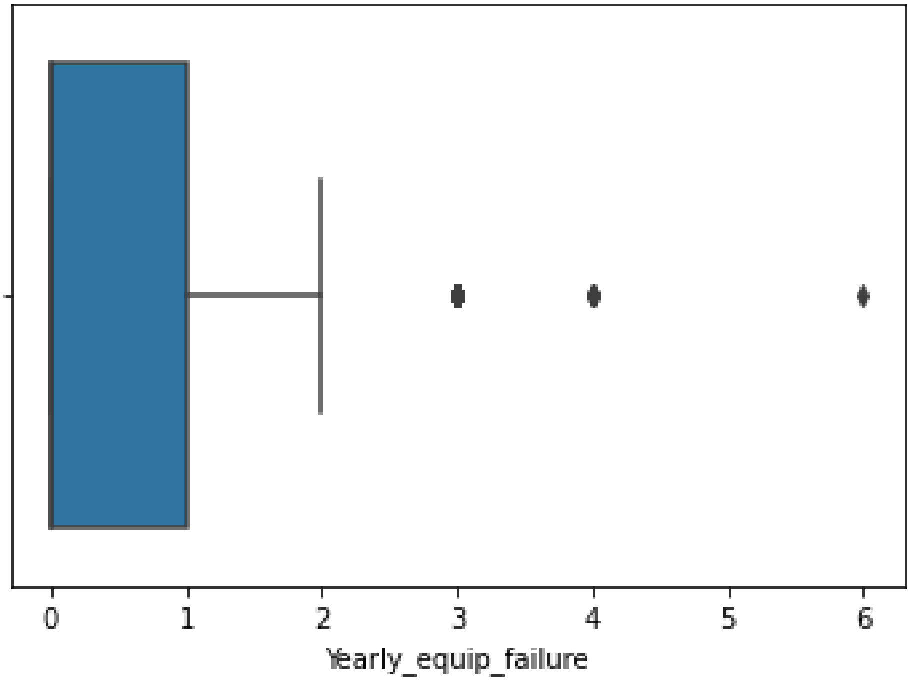


Boxplots for variables containing values with a zscore greater than 3 or less than -3:

MonthlyCharge:

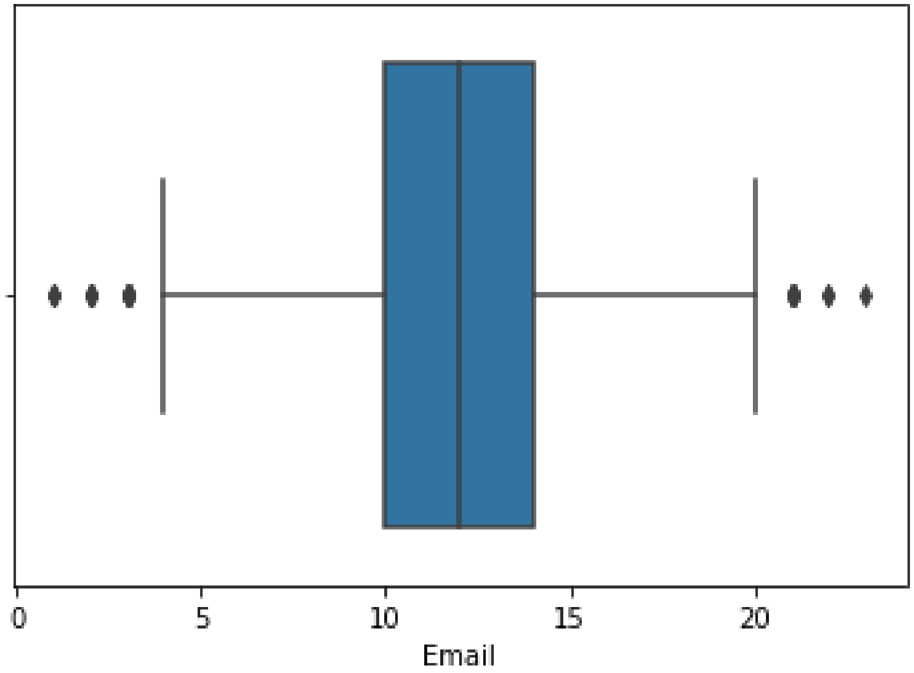


Yearly_equip_failure:

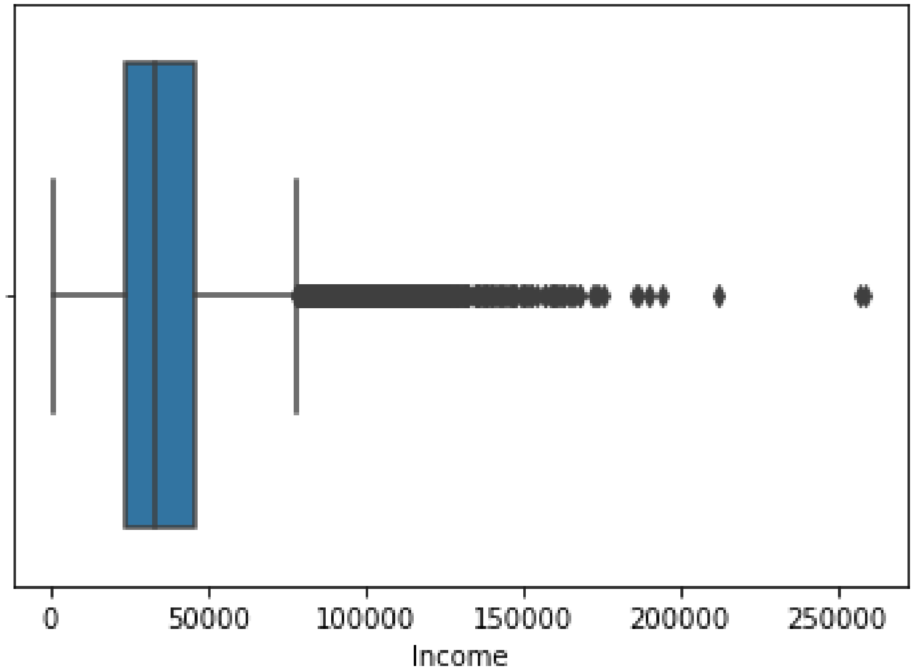


William Stults – D206

Email:

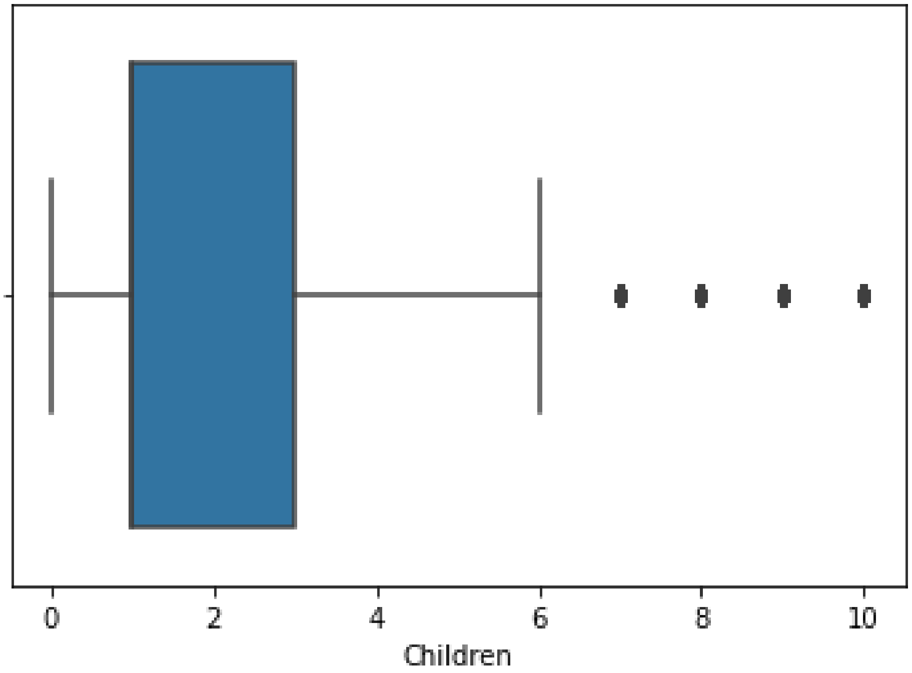


Income:

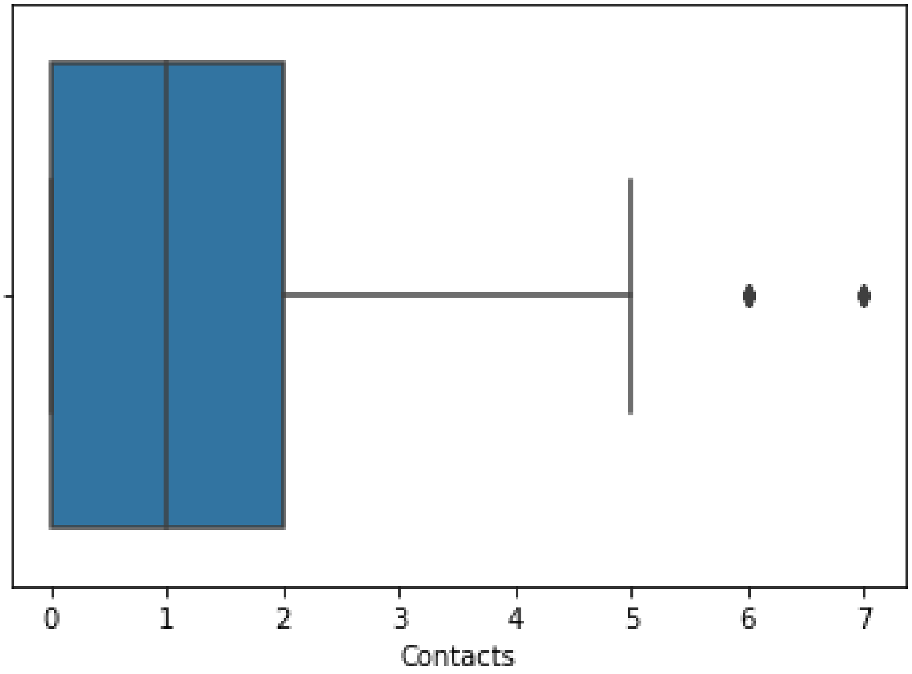


William Stults – D206

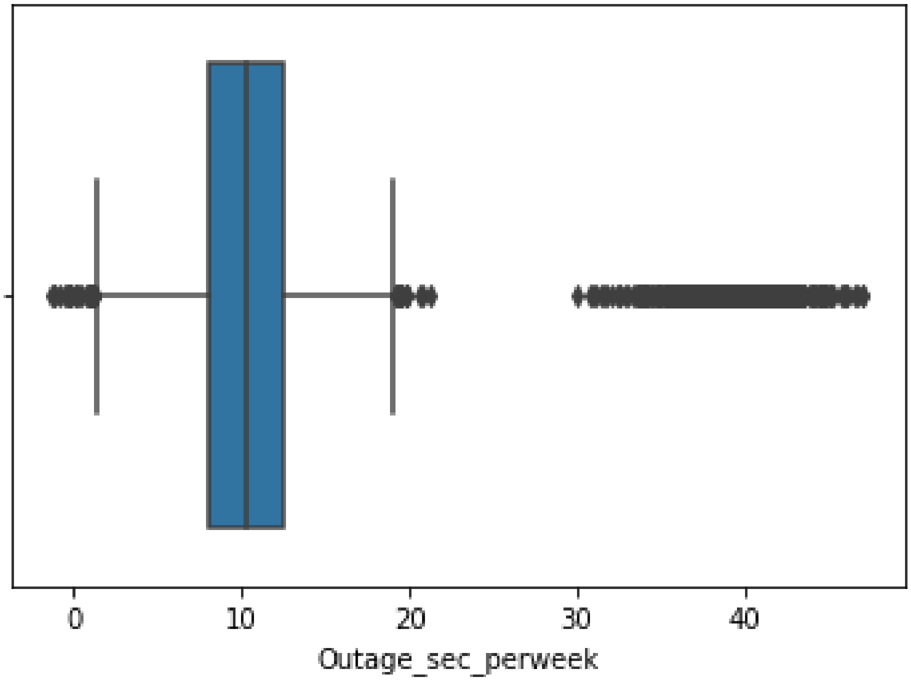
Children:



Contacts:



Outage_sec_perweek:

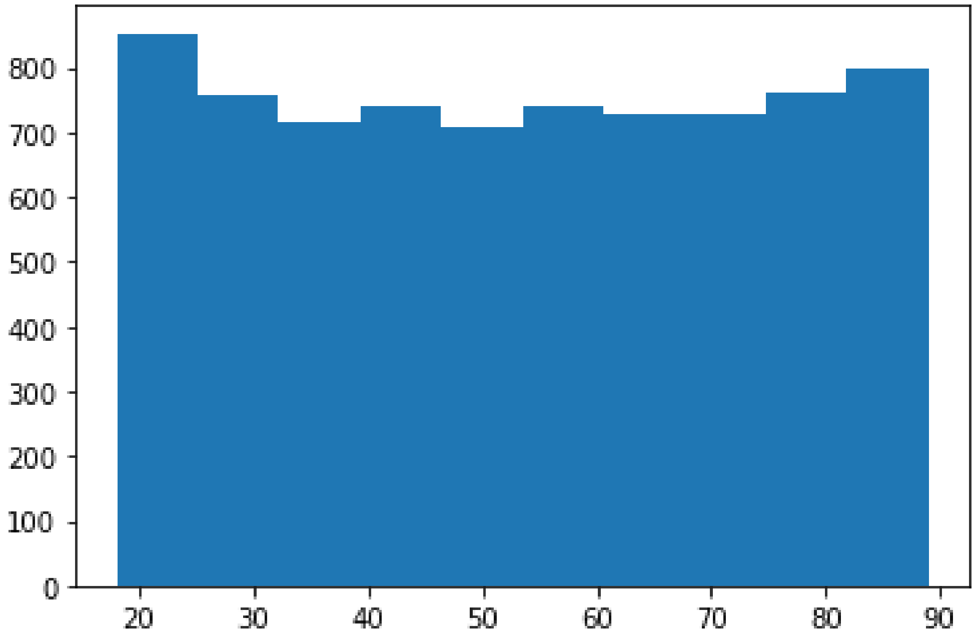


Mitigation Methods

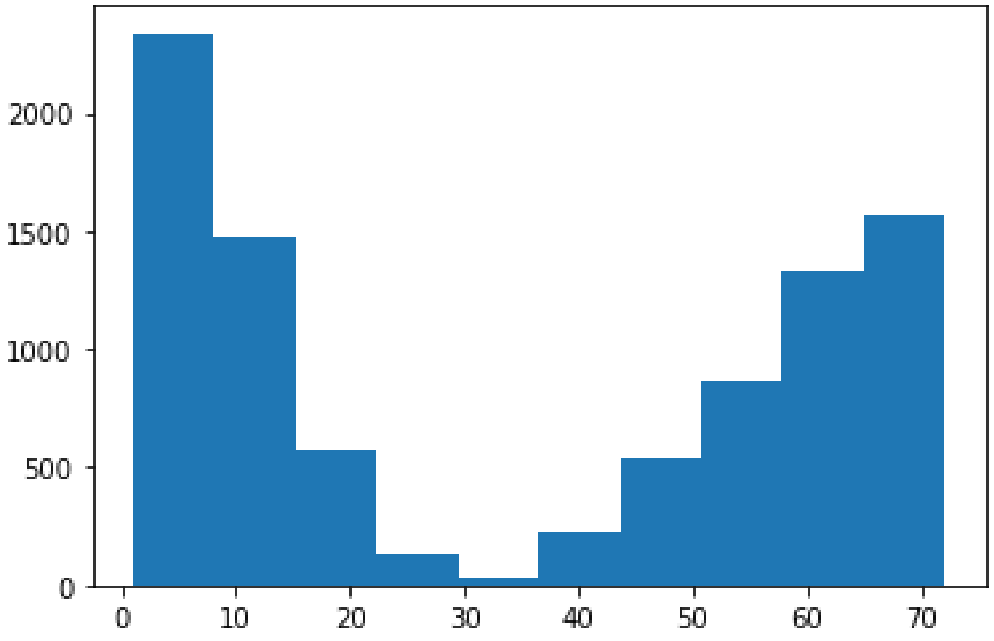
Of the variables evaluated for missing or null values, Age, Tenure, and Bandwidth_GB_Year all display a bimodal, uniform, or normal distribution.

William Stults – D206

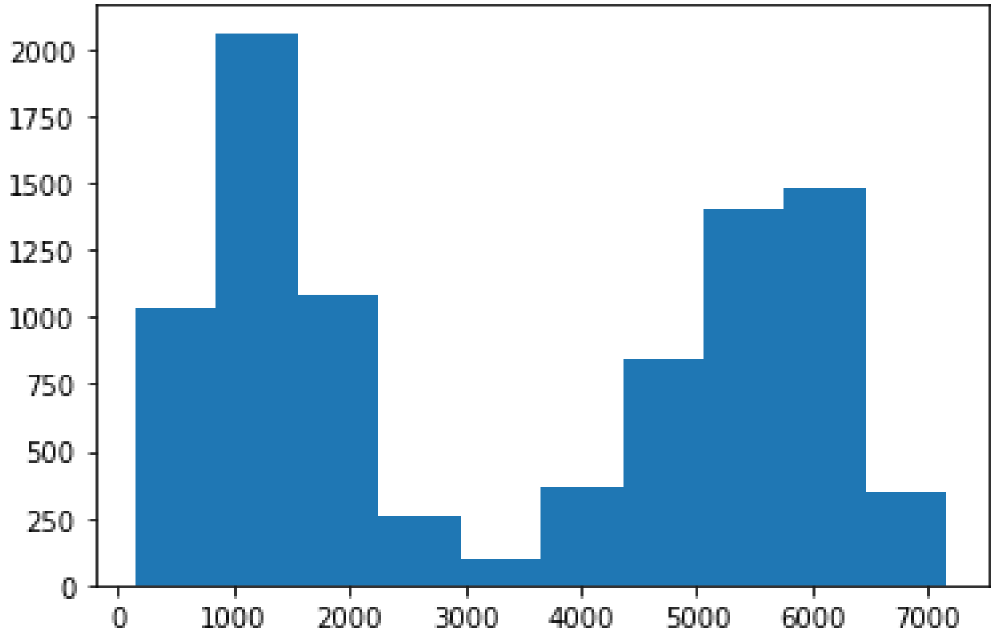
Age:



Tenure:



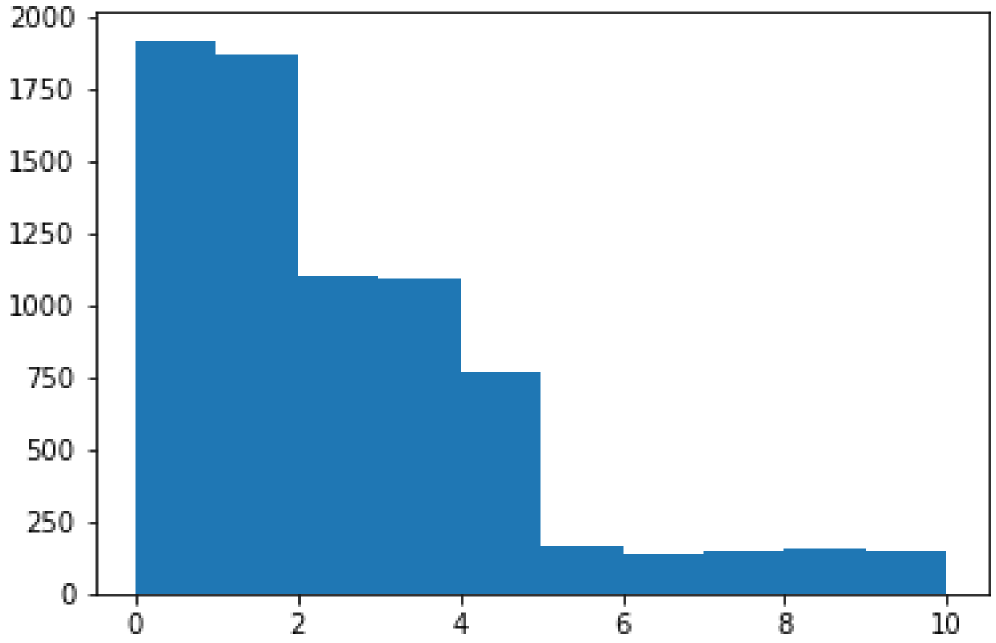
Bandwidth_GB_Year:



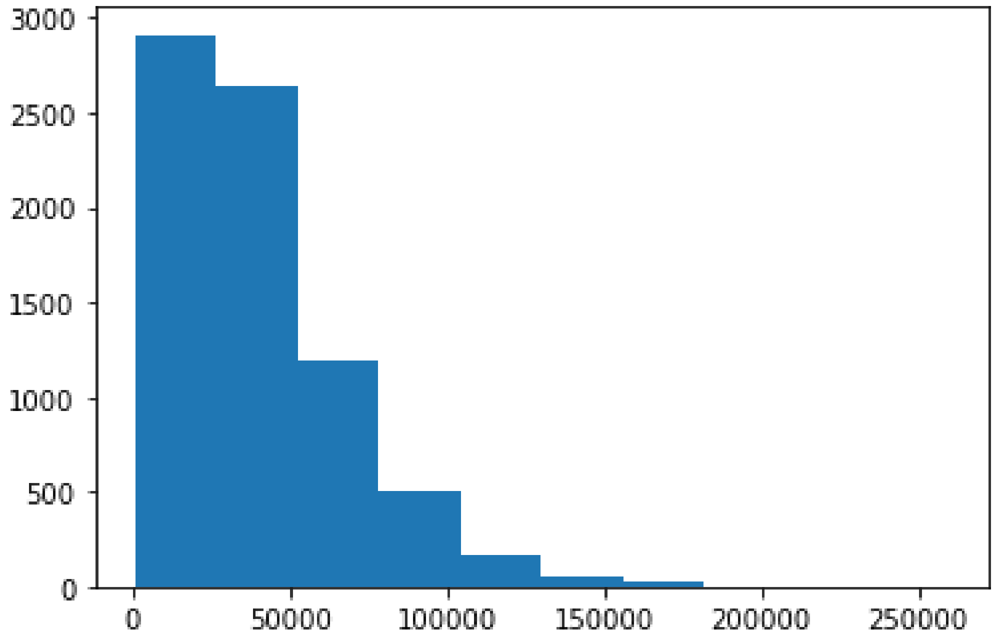
These types of distributions, when seen in quantitative numerical variables, are suitable for using the mean value of the variable to replace missing or null values. This was implemented via the fillna() and mean() functions.

For variables Children and Income we see a skewed distribution, indicating that using the median value via fillna() and median() for replacement is more appropriate.

Children:



Income:



For categorical variables not quantitative in nature, such as Techie, Phone, and TechSupport, I have replaced null and missing values with the generic value “NA” using the fillna() and mode() functions.

When evaluating outliers for potential treatment actions to be taken, I found that, in my judgment, there were no values suitable for outright exclusion. After examining the histogram and boxplot output images, I resolved to inspect the outlier values directly in the dataset. Based on their relative position compared to other values in the same variable, I found no values I could conclude were factual errors or illegitimate entries. For this reason, no outlier values have been excluded from the dataset and no rows have been removed.

It seemed beneficial to me for these outlier values to be contained and stored in their own datasets, however, should the need arise in the future for those values to be more closely examined. Because of this, I have saved them to separate files while still retaining them in the main dataset.

To further prepare the dataset for forthcoming data analysis I elected to perform a re-expression of 3 categorical variables; Marital, Contract, and Education. The values in the Marital variable are nominal, meaning they cannot be ranked in order of greatest to least, worst to best, etc. I employed one-hot encoding to transform the Marital variable into multiple variables containing a value of either 1 or 0 via the get_dummies() function from the Pandas library, then added these new variables to the existing data frame before dropping the original Marital variable.

The values in Contract and Education can be considered ordinal data as there is an easily defined ranking that can be applied. These variable values were re-expressed as numerical rankings in their own new variables, Contract_Duration and Education_Level, utilizing Python dictionaries and the replace() function.

Outcome

After treating missing values, null values, and outliers, the dataset remains intact with no rows meriting outright removal. A total of 13,906 missing or null values have been replaced with values appropriate to their data types and distributions, maintaining the integrity and shape of the data as much as possible. Outliers have been retained but also saved separately for later review should the need arise. Three categorical variables have been re-expressed numerically, further prepping the data for data mining and machine learning/algorithm consumption.

Code

```
##Treat quantitative null values for numerical variables

df['Age'].fillna(df['Age'].mean(), inplace = True)

df['Children'].fillna(df['Children'].median(), inplace = True)

df['Income'].fillna(df['Income'].median(), inplace = True)

df['Tenure'].fillna(df['Tenure'].mean(), inplace = True)

df['Bandwidth_GB_Year'].fillna(df['Bandwidth_GB_Year'].mean(), inplace
= True)


##Treat qualitative null values for categorical variables

df['Techie'] = df['Techie'].fillna(df['Techie'].mode()[0])

df['Phone'] = df['Phone'].fillna(df['Phone'].mode()[0])

df['TechSupport'] =
df['TechSupport'].fillna(df['TechSupport'].mode()[0])


##Output csv files for outlier retention

churn_MonthlyCharge_z = df.query('MonthlyCharge_z > 3 |
MonthlyCharge_z < -3')

churn_MonthlyCharge_z_sort =
churn_MonthlyCharge_z.sort_values(['MonthlyCharge'], ascending =
False)

churn_MonthlyCharge_z_sort.to_csv(r'C:\Users\wstul\d206\churn_MonthlyC
harge_z_sort.csv')

churn_Yearly_equip_failure_z = df.query('Yearly_equip_failure_z > 3 |
Yearly_equip_failure_z < -3')

churn_Yearly_equip_failure_z_sort =
churn_Yearly_equip_failure_z.sort_values(['Yearly_equip_failure'],
ascending = False)
```

```
churn_Yearly_equip_failure_z_sort.to_csv(r'C:\Users\wstul\d206\churn_Y
early_equip_failure_z_sort.csv')

churn_Email_z = df.query('Email_z > 3 | Email_z < -3')

churn_Email_z_sort = churn_Email_z.sort_values(['Email'], ascending =
False)

churn_Email_z_sort.to_csv(r'C:\Users\wstul\d206\churn_Email_z_sort.csv
')

churn_Income_z = df.query('Income_z > 3 | Income_z < -3')

churn_Income_z_sort = churn_Income_z.sort_values(['Income'], ascending
= False)

churn_Income_z_sort.to_csv(r'C:\Users\wstul\d206\churn_Income_z_sort.c
sv')

churn_Children_z = df.query('Children_z > 3 | Children_z < -3')

churn_Children_z_sort = churn_Children_z.sort_values(['Children'],
ascending = False)

churn_Children_z_sort.to_csv(r'C:\Users\wstul\d206\churn_Children_z_so
rt.csv')

churn_Contacts_z = df.query('Contacts_z > 3 | Contacts_z < -3')

churn_Contacts_z_sort = churn_Contacts_z.sort_values(['Contacts'],
ascending = False)

churn_Contacts_z_sort.to_csv(r'C:\Users\wstul\d206\churn_Contacts_z_so
rt.csv')

churn_outage_sec_z = df.query('Outage_sec_perweek_z > 3 |
Outage_sec_perweek_z < -3')

churn_outage_sec_z_sort =
churn_outage_sec_z.sort_values(['Outage_sec_perweek_z'], ascending =
False)

churn_outage_sec_z_sort.to_csv(r'C:\Users\wstul\d206\churn_outage_sec_
z_sort.csv')


##Drop zscore columns from dataset

df.drop(['Children_z', 'Age_z', 'Income_z', 'Outage_sec_perweek_z',
'Email_z', 'Contacts_z', 'Yearly_equip_failure_z', 'Tenure_z',
'MonthlyCharge_z', 'Bandwidth_GB_Year_z'], axis=1, inplace=True)


##Implement one-hot encoding on Marital column values
```

```
df_marital_ohe = pd.get_dummies(df['Marital'], prefix = 'Marital',
drop_first = False)

df_marital_ohe

df = pd.concat([df, df_marital_ohe], axis = 1)

df.drop(['Marital'], axis=1, inplace=True)


##Implement label encoding for ordinal variables Contract and
##Education

scale_mapper = {'Month-to-month' : 1, 'One year' : 2, 'Two Year' : 3}

df['Contract_Duration'] = df['Contract'].replace(scale_mapper)

scale_mapper = {'No Schooling Completed' : 1, 'Nursery School to 8th
Grade' : 2, '9th Grade to 12th Grade, No Diploma' : 3, 'GED or
Alternative Credential' : 4, 'Regular High School Diploma' : 5, 'Some
College, Less than 1 Year' : 6, 'Some College, 1 or More Years, No
Degree' : 7, 'Professional School Degree' : 8, "Associate's Degree" :
9, "Bachelor's Degree" : 10, "Master's Degree" : 11, 'Doctorate
Degree' : 12}

df['Education_Level'] = df['Education'].replace(scale_mapper)
```

Summary of Limitations

Any imputations performed to treat missing or null values were univariate, as multivariate imputation methods have not been used. In addition to this, no rows were excluded from the original dataset due to missing values or outliers, so the dataset has retained its original size. Re-expression of categorical values, however, has resulted in an increase in the number of columns in the dataset.

Effect on Analysis

While using mean, median, and mode for missing and null value treatment is appropriate, it is essentially replacing one inaccurate value with another. This can potentially result in some distortion of data or its distribution. Retaining outliers and re-expressing some categorical variables increases the resulting size of the dataset. While the impact in this specific situation is likely minimal, these types of expansion can sometimes result in additional computational power being required in future steps such as data mining.

Principal Component Analysis

Principal Components

The following variables were included in principal component analysis:

Children

Age

Income

Outage_sec_perweek

Email

Contacts

Yearly_equip_failure

Tenure

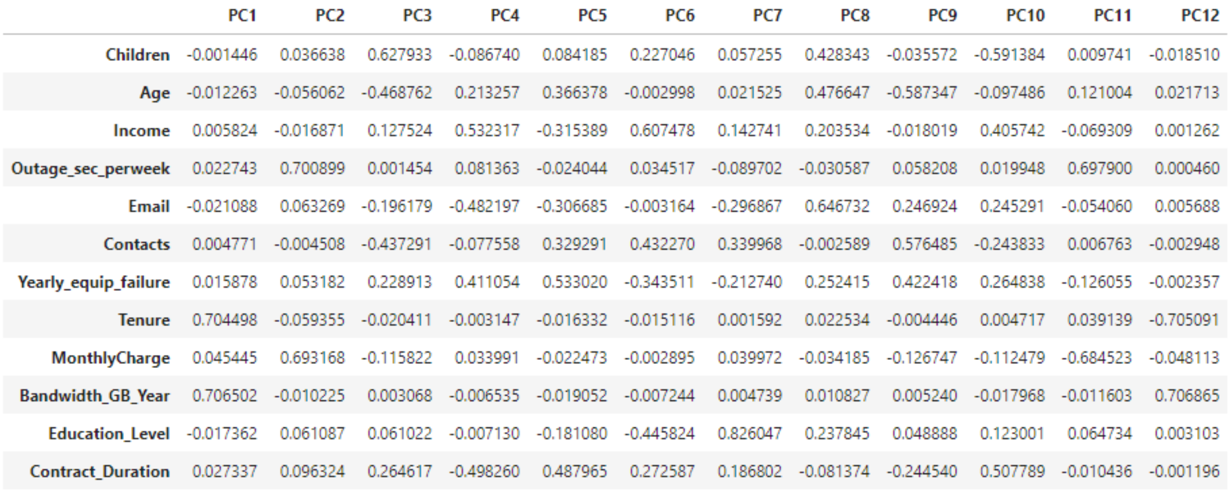
MonthlyCharge

Bandwidth_GB_Year

Education_Level

Contract_Duration

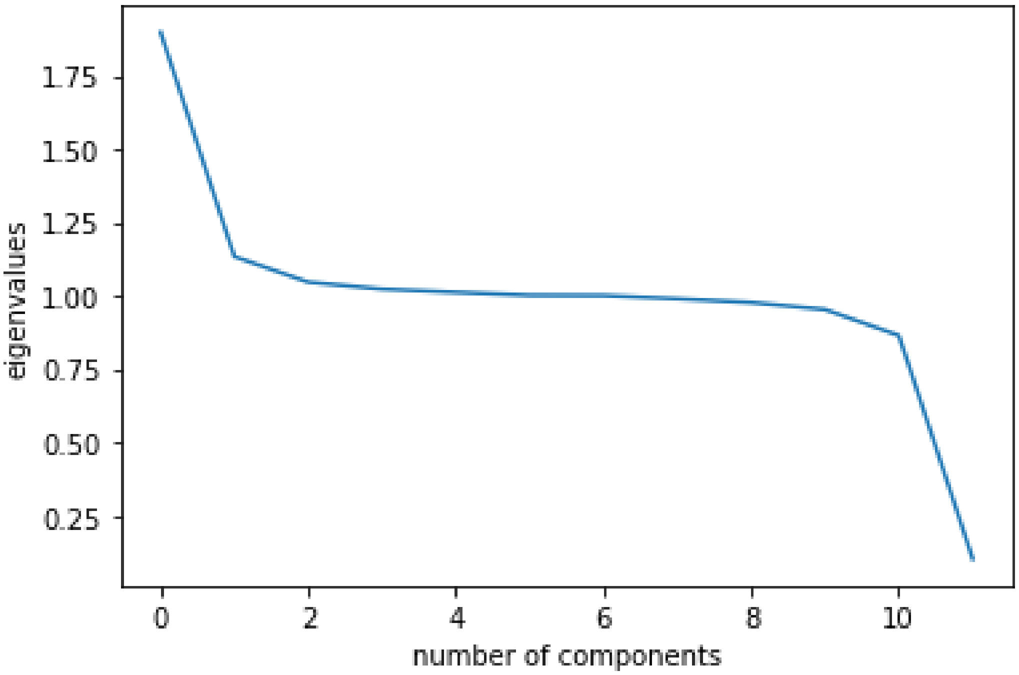
Taking 12 variables as input resulted in 12 principal components. The below image contains the loadings resulting from principal component analysis:



	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Children	-0.001446	0.036638	0.627933	-0.086740	0.084185	0.227046	0.057255	0.428343	-0.035572	-0.591384	0.009741	-0.018510
Age	-0.012263	-0.056062	-0.468762	0.213257	0.366378	-0.002998	0.021525	0.476647	-0.587347	-0.097486	0.121004	0.021713
Income	0.005824	-0.016871	0.127524	0.532317	-0.315389	0.607478	0.142741	0.203534	-0.018019	0.405742	-0.069309	0.001262
Outage_sec_perweek	0.022743	0.700899	0.001454	0.081363	-0.024044	0.034517	-0.089702	-0.030587	0.058208	0.019948	0.697900	0.000460
Email	-0.021088	0.063269	-0.196179	-0.482197	-0.306685	-0.003164	-0.296867	0.646732	0.246924	0.245291	-0.054060	0.005688
Contacts	0.004771	-0.004508	-0.437291	-0.077558	0.329291	0.432270	0.339968	-0.002589	0.576485	-0.243833	0.006763	-0.002948
Yearly_equip_failure	0.015878	0.053182	0.228913	0.411054	0.533020	-0.343511	-0.212740	0.252415	0.422418	0.264838	-0.126055	-0.002357
Tenure	0.704498	-0.059355	-0.020411	-0.003147	-0.016332	-0.015116	0.001592	0.022534	-0.004446	0.004717	0.039139	-0.705091
MonthlyCharge	0.045445	0.693168	-0.115822	0.033991	-0.022473	-0.002895	0.039972	-0.034185	-0.126747	-0.112479	-0.684523	-0.048113
Bandwidth_GB_Year	0.706502	-0.010225	0.003068	-0.006535	-0.019052	-0.007244	0.004739	0.010827	0.005240	-0.017968	-0.011603	0.706865
Education_Level	-0.017362	0.061087	0.061022	-0.007130	-0.181080	-0.445824	0.826047	0.237845	0.048888	0.123001	0.064734	0.003103
Contract_Duration	0.027337	0.096324	0.264617	-0.498260	0.487965	0.272587	0.186802	-0.081374	-0.244540	0.507789	-0.010436	-0.001196

Identification Process

The Kaiser Rule was utilized to help decide which principal components should be kept. My loadings from the previous section were processed via the dot() function from the NumPy library to determine covariance and eigenvectors, and ultimately assign eigenvalues for each principal component. Of all principal components, eigenvalues for principal components 1 and 2 were clearly above 1, and appear to be the best candidates for retention. The scree plot below illustrates these results.



Benefits

While PCA would likely not be used on a dataset this small, it can provide great benefit when working with big data and larger data sets in general. The ability to group variables that share a high degree of covariance can significantly reduce the dimensionality of a dataset, reducing the time and computational power necessary for more intensive data analysis operations. PCA can also provide a strategic advantage when making business decisions by highlighting these variables which share high degrees of covariance.

William Stults – D206

Web Source References

https://towardsdatascience.com/finding-and-removing-duplicate-rows-in-pandas-dataframe-c6117668631f

https://www.projectpro.io/recipes/encode-ordinal-categorical-features-in-python

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html

https://thispointer.com/pandas-get-unique-values-in-single-or-multiple-columns-of-a-dataframe-in-python

https://www.mygreatlearning.com/blog/label-encoding-in-python/

References

Larose, C. D., & Larose, D. T. (2019). Data science using Python and R. ISBN-13: 978-1-119-52684-1

Insights for Professionals. (2019, February 26). *5 Niche Programming Languages (And Why They're Underrated)*. **https://www.insightsforprofessionals.com/it/software/niche-programming-languages**

Johari, A. (2019, September 4). *What is Python SciPy and How to use it?*. Medium. **https://medium.com/edureka/scipy-tutorial-38723361ba4b**

Jain, K. (2015, January 5). *Scikit-learn(sklearn) in Python – the most important Machine Learning tool I learnt last year!*. Analytics Vidhya. **https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/**

Parra, H. (2021, April 20). *The Data Science Trilogy*. Towards Data Science. **https://towardsdatascience.com/the-data-science-trilogy-numpy-pandas-and-matplotlib-basics-42192b89e26**