Predictive Modeling D208
Task 2 – Logistic Regression
Eric D. Otten
Student ID: 011399183

A1. Describe the purpose of this data analysis by doing the following: Summarize one research question that is relevant to a real-world organizational situation captured in the data set you have selected and that you will answer using logistic regression.

My research question is: “Which customer factors contribute most to a customer’s decision to cancel their subscription with the service provider (i.e churn?)” This question is relevant to an organization because decision-makers use customer behavioral patterns to decrease churn.

A2. Define the goals of the data analysis.

My goal with the analysis in D208 Task 2 is to build upon the model created in D208 Task 1 by employing logistic regression rather than multiple linear regression.

This model will provide insights as to which customer factors contribute most to churn. This information will be used by stakeholders of an organization to identify the signs that a customer is likely to churn, which can be counteracted with loyalty benefits and customer retention strategies.

In economics, the Pareto principle is often observed, which states that for many outcomes, roughly 80% of consequences come from 20% of causes. The goal with my research analysis is to identify the 20% (or fewer) of factors that predict 80% (or more) of the churn. This can be achieved by looking at the statistical significance and the magnitude of the coefficients in the logistic regression model.

This will be accomplished by creating an initial logistic regression model and reducing it to a few key features (i.e. the 20%) which predict the majority (i.e. the 80%) of churn. Please note, the Pareto principle is not a hard rule; rather a conceptual model which helps decision makers understand the difference between inputs and outputs and is useful for storytelling with data. This principle is widely understood within business contexts as a useful heuristic. It is likely that more than 80% of the churn can be explained by fewer than 20% of the customer factors, and these two figures need not add up to 100%.

In Part F2, I will describe applications of my data analysis and how these results can be applied within the context of the Pareto principle.

B1. Describe logistic regression methods by doing the following: Summarize four assumptions of a logistic regression model.

The first assumption of logistic regression is that the result is binary. Therefore, my research question seeks to predict a binary output: customer churn.

The second assumption of logistic regression is that observations are independent. This means that the outcome of one observation should not influence, nor be influenced by, the outcomes of other observations in the data set. Within the context of the churn dataset, if customers discuss with each other about canceling their subscription with the service provider, this shared behavior would violate the assumption of independence.

The third assumption of logistic regression is that there is no multicollinearity among explanatory variables. Multicollinearity occurs when two or more predictor variables in a regression model are highly correlated, meaning one can be linearly predicted from the others with a substantial degree of accuracy.

The fourth assumption of logistic regression is that there are no extreme outliers. Extreme outliers can be corrected for by either removing them, replacing them with the mean or median, or by simply making a note of them when reporting the regression results.

B2. Describe two benefits of using Python or R in support of various phases of the analysis.

I chose to use Python for my data cleaning and analysis due to its flexibility, simplicity, and numerous powerful libraries.

I will import the following Python libraries:

- pandas, which allows for handling large datasets and importing .csv files.
- numpy which allows for mathematical operations on the dataset.
- scikit-learn/sklearn for machine learning, linear regression, and model evaluation.
- matplotlib for graphing functionality.
- statsmodels which allows for statistical modeling, including regression analysis.
- seaborn which allows for informative statistical graphics.

B3. Explain why logistic regression is an appropriate technique to analyze the research question summarized in part I.

As previously discussed in Part B1, one assumption of logistic regression is that the output must be binary. Therefore, it is an appropriate technique for answering my research question, “Which customer factors contribute most to a customer’s decision to cancel their subscription with the service provider (i.e churn?),” as we aim to predict churn, which is a binary output: 1) Yes, the customer churned this month or 0) No, the customer did not churn this month.

C1. Summarize the data preparation process for logistic regression by doing the following: Describe your data cleaning goals and the steps used to clean the data to achieve the goals that align with your research question including the annotated code.

My goal with regards to cleaning the sample data is to create a uniform DataFrame to which logistic regression can be applied and from which useful business insights can be drawn.

Firstly, I will analyze only columns that are relevant to customer churn. This excludes variables such as latitude and longitude which likely have minimal to no effect on customer churn.

- **Area**: Classification of the area (rural, urban, suburban) based on census data.
- **Children**: Number of children in the customer's household as reported.
- **Age**: Age of the customer as reported in sign-up information.
- **Income**: Annual income of the customer as reported at time of sign-up.
- **Marital**: Marital status of the customer as reported in sign-up information.
- **Gender**: Gender identity of the customer as self-reported.

- **Contract**: Type of contract the customer has (month-to-month, one year, two year).
- **Port_modem**: Indicates whether the customer has a portable modem (yes, no).
- **Tablet**: Indicates whether the customer owns a tablet (yes, no).
- **InternetService**: Type of internet service (DSL, fiber optic, none).
- **Phone**: Indicates whether the customer has a phone service (yes, no).
- **Multiple**: Indicates whether the customer has multiple lines (yes, no).
- **OnlineSecurity**: Whether the customer subscribes to an online security service.
- **OnlineBackup**: Whether the customer uses an online backup service.
- **DeviceProtection**: Whether the customer has a device protection service.
- **TechSupport**: Whether the customer has technical support service.
- **StreamingTV**: Whether the customer subscribes to streaming TV service.
- **StreamingMovies**: Whether the customer subscribes to streaming movies service.
- **PaperlessBilling**: Indicates whether the customer uses paperless billing (yes, no).
- **PaymentMethod**: Customer's payment method (e.g., electronic check, bank transfer, etc.).
- **Tenure**: Number of months the customer has been with the provider.
- **MonthlyCharge**: Average monthly charge billed to the customer.
- **Bandwidth_GB_Year**: Average annual data usage in gigabytes.
- **Outage_sec_perweek**: Average number of seconds per week of system outages.
- **Email**: Number of emails sent to the customer in the last year.
- **Contacts**: Number of times customer contacted technical support.
- **Yearly_equip_failure**: Number of equipment failures experienced in a year.
- **Techie**: Indicates if the customer considers themselves technically inclined.
- **Item1**: Rating for timely response.
- **Item2**: Rating for timely fixes.
- **Item3**: Rating for timely replacements.
- **Item4**: Rating for reliability.
- **Item5**: Rating for options.
- **Item6**: Rating for respectful response.
- **Item7**: Rating for courteous exchange.
- **Item8**: Rating for evidence of active listening.

For nominal categorical data, one hot encoding will be used as it is the most widespread approach. This approach involves creating a new column for each category, which contains a binary encoding of 0 or 1 to denote whether a particular row belongs to this category. This can be achieved using the get_dummies() method within the pandas library. Additionally, the `drop_first=True` argument will be used to avoid the dummy variable trap.

The following code below implements my data cleaning strategy and includes comments for each step of this process:

Code

```
from matplotlib.colors import ListedColormap
from sklearn import linear_model
import matplotlib.pyplot as plt
import pandas as pd
import re
import seaborn as sns
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,
precision_score, recall_score


def to_snake_case(df):
    new_df = df.copy()

    def convert(name):
        # Handle the internal capital letters and add an underscore before capitals
        name = re.sub(r'(?<!^)(?=[A-Z])', '_', name).lower()
        # Replace spaces and other special characters with underscores
        name = re.sub(r'\s+|\W', '_', name)
        # Remove any double underscores caused by previous replacements
        name = re.sub(r'__+', '_', name)
        # Remove any trailing underscore
        name = re.sub(r'_+$', '', name)
        return name

    # Apply the convert function to each column name
    new_df.columns = [convert(col) for col in new_df.columns]
    return new_df

# Path to the CSV file
file_path = 'churn_clean.csv'

# Load the entire dataset
df = pd.read_csv(file_path, keep_default_na = False, na_values=["NA"])



print("Press Enter to begin Part C1...")
# C1. Summarize the data preparation process for logistic regression by doing the
following: Describe your data cleaning goals and the steps used to clean the data to
achieve the goals that align with your research question including the annotated code.

# List of columns to select
selected_columns = [
    'Area', 'Children', 'Age', 'Income', 'Marital', 'Gender', 'Contract',
    'Port_modem', 'Tablet', 'InternetService', 'Phone', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
    'StreamingTV', 'StreamingMovies', 'PaperlessBilling', 'PaymentMethod',
    'Tenure', 'MonthlyCharge', 'Bandwidth_GB_Year', 'Outage_sec_perweek',
    'Email', 'Contacts', 'Yearly_equip_failure', 'Techie', 'Item1', 'Item2',
```

```
    'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8', 'Churn'
]

# Select the specified columns
df = df[selected_columns]

# Rename all columns using snake_case
df = to_snake_case(df)

# Columns that are expected to be boolean
boolean_columns = [
    'port_modem', 'tablet', 'phone', 'multiple', 'online_security', 'online_backup',
    'device_protection', 'tech_support', 'streaming_t_v', 'streaming_movies',
    'paperless_billing', 'techie', 'churn'
]

# Convert "yes" to True and "no" to False
for column in boolean_columns:
    df[column] = df[column].map({'Yes': True, 'No': False})

# Print summary statistics before creating dummy variables
categorical_columns = [
    'marital',
    'area',
    'gender',
    'contract',
    'internet_service',
    'payment_method'
]
# Convert specified columns to 'category' dtype
for column in categorical_columns:
    df[column] = df[column].astype('category')
```

The annotated code describes the following steps of the data preparation process:

- Only the specified columns will be selected to construct the initial logistic regression model
- Columns names are made Pythonic using snake case
- Boolean columns are converted to real Booleans instead of strings containing “Yes” or “No”
- Categorical data are converted to real categories instead of strings
- In Part C4, following data visualization, dummy categories will be generated. This was done after visualization in order to visualize categorical data properly.

C2. Describe the dependent variable and all independent variables using summary statistics that are required to answer the research question, including a screenshot of the summary statistics output for each of these variables.

My research question is “Which customer factors contribute most to a customer’s decision to cancel their subscription with the service provider (i.e churn?)” Therefore, the dependent variable in my analysis is customer churn. The remaining customer factors are independent variables. To generate summary

statistics, the `value_counts()` method will be run on all categorical data and the `describe()` method will be run on all numerical data. The following Python code below implements this functionality.

Code

```
# C2. Describe the dependent variable and all independent variables using summary
statistics that are required to answer the research question, including a screenshot of
the summary statistics output for each of these variables.

numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns

# Generate summary statistics for numerical data
for column in numerical_columns:
    print(f"\nSummary statistics for column: {column}")
    print(df[column].describe())

print("\nFrequency Distribution for Categorical Data:")
for column in categorical_columns:
    print(f"\nFrequency count for column: {column}")
    print(df[column].value_counts())
```

Result

```
Summary statistics for column: children
count    10000.0000
mean         2.0877
std          2.1472
min          0.0000
25%          0.0000
50%          1.0000
75%          3.0000
max         10.0000
Name: children, dtype: float64

Summary statistics for column: age
count    10000.000000
mean        53.078400
std         20.698882
min         18.000000
25%         35.000000
50%         53.000000
75%         71.000000
max         89.000000
Name: age, dtype: float64

Summary statistics for column: income
count     10000.000000
mean      39806.926771
std       28199.916702
min         348.670000
25%       19224.717500
50%       33170.605000
```

```
75%       53246.170000
max      258900.700000
Name: income, dtype: float64

Summary statistics for column: tenure
count    10000.000000
mean        34.526188
std         26.443063
min          1.000259
25%          7.917694
50%         35.430507
75%         61.479795
max         71.999280
Name: tenure, dtype: float64

Summary statistics for column: monthly_charge
count    10000.000000
mean       172.624816
std         42.943094
min         79.978860
25%        139.979239
50%        167.484700
75%        200.734725
max        290.160419
Name: monthly_charge, dtype: float64

Summary statistics for column: bandwidth_g_b_year
count    10000.000000
mean      3392.341550
std       2185.294852
min        155.506715
25%       1236.470827
50%       3279.536903
75%       5586.141370
max       7158.981530
Name: bandwidth_g_b_year, dtype: float64

Summary statistics for column: outage_sec_perweek
count    10000.000000
mean        10.001848
std          2.976019
min          0.099747
25%          8.018214
50%         10.018560
75%         11.969485
max         21.207230
Name: outage_sec_perweek, dtype: float64

Summary statistics for column: email
count    10000.000000
mean        12.016000
std          3.025898
min          1.000000
```

```
25%         10.000000
50%         12.000000
75%         14.000000
max         23.000000
Name: email, dtype: float64

Summary statistics for column: contacts
count    10000.000000
mean         0.994200
std          0.988466
min          0.000000
25%          0.000000
50%          1.000000
75%          2.000000
max          7.000000
Name: contacts, dtype: float64

Summary statistics for column: yearly_equip_failure
count    10000.000000
mean         0.398000
std          0.635953
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          6.000000
Name: yearly_equip_failure, dtype: float64

Summary statistics for column: item1
count    10000.000000
mean         3.490800
std          1.037797
min          1.000000
25%          3.000000
50%          3.000000
75%          4.000000
max          7.000000
Name: item1, dtype: float64

Summary statistics for column: item2
count    10000.000000
mean         3.505100
std          1.034641
min          1.000000
25%          3.000000
50%          4.000000
75%          4.000000
max          7.000000
Name: item2, dtype: float64

Summary statistics for column: item3
count    10000.000000
mean         3.487000
```

```
std          1.027977
min          1.000000
25%          3.000000
50%          3.000000
75%          4.000000
max          8.000000
Name: item3, dtype: float64

Summary statistics for column: item4
count    10000.000000
mean         3.497500
std          1.025816
min          1.000000
25%          3.000000
50%          3.000000
75%          4.000000
max          7.000000
Name: item4, dtype: float64

Summary statistics for column: item5
count    10000.000000
mean         3.492900
std          1.024819
min          1.000000
25%          3.000000
50%          3.000000
75%          4.000000
max          7.000000
Name: item5, dtype: float64

Summary statistics for column: item6
count    10000.000000
mean         3.497300
std          1.033586
min          1.000000
25%          3.000000
50%          3.000000
75%          4.000000
max          8.000000
Name: item6, dtype: float64

Summary statistics for column: item7
count    10000.000000
mean         3.509500
std          1.028502
min          1.000000
25%          3.000000
50%          4.000000
75%          4.000000
max          7.000000
Name: item7, dtype: float64

Summary statistics for column: item8
```

```
count    10000.000000
mean         3.495600
std          1.028633
min          1.000000
25%          3.000000
50%          3.000000
75%          4.000000
max          8.000000
Name: item8, dtype: float64

Frequency Distribution for Categorical Data:

Frequency count for column: marital
marital
Divorced         2092
Widowed          2027
Separated        2014
Never Married    1956
Married          1911
Name: count, dtype: int64

Frequency count for column: area
area
Suburban    3346
Rural       3327
Urban       3327
Name: count, dtype: int64

Frequency count for column: gender
gender
Female       5025
Male         4744
Nonbinary     231
Name: count, dtype: int64

Frequency count for column: contract
contract
Month-to-month    5456
Two Year          2442
One year          2102
Name: count, dtype: int64

Frequency count for column: internet_service
internet_service
Fiber Optic    4408
DSL            3463
None           2129
Name: count, dtype: int64

Frequency count for column: payment_method
payment_method
Electronic Check            3398
Mailed Check                2290
```

```
Bank Transfer(automatic)    2229
Credit Card (automatic)     2083
Name: count, dtype: int64
```

C3. Generate univariate and bivariate visualizations of the distributions of the dependent and independent variables, including the dependent variable in your bivariate visualizations.

This code generates univariate and bivariate visualizations for both numerical and categorical/boolean variables in a dataset to analyze their distributions and relationships with a dependent variable (churn).

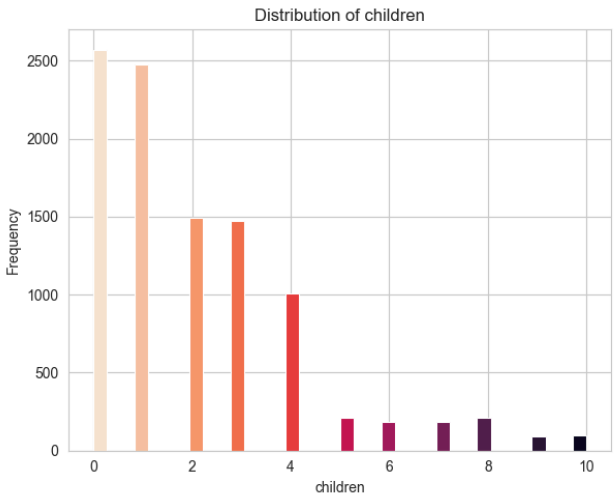
Firstly, numerical variables are analyzed. The code uses Seaborn to generate a plot with two subplots: one containing a univariate visualization and the other, a bivariate visualization. A histogram is used for univariate analysis of numerical data and a box plot is used for bivariate visualization of numerical data and the dependent variable churn.

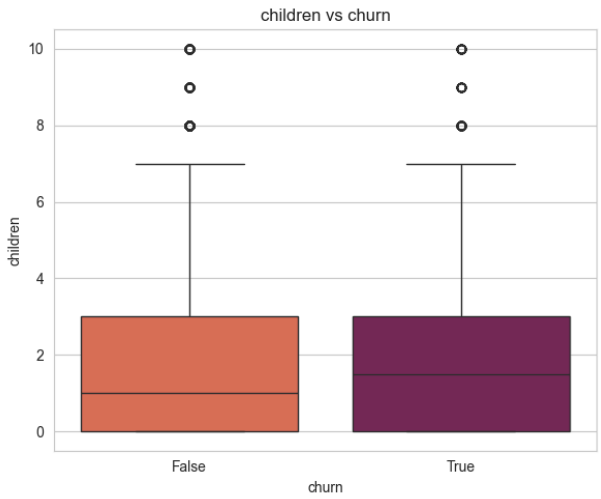
Then, categorical/boolean variables are analyzed. The code uses Seaborn to generate a plot with two subplots: one containing a univariate visualization and the other, a bivariate visualization. A pie chart is used for univariate analysis of categorical/boolean data and a contingency table is used for bivariate visualization of categorical/boolean data and the dependent variable churn.

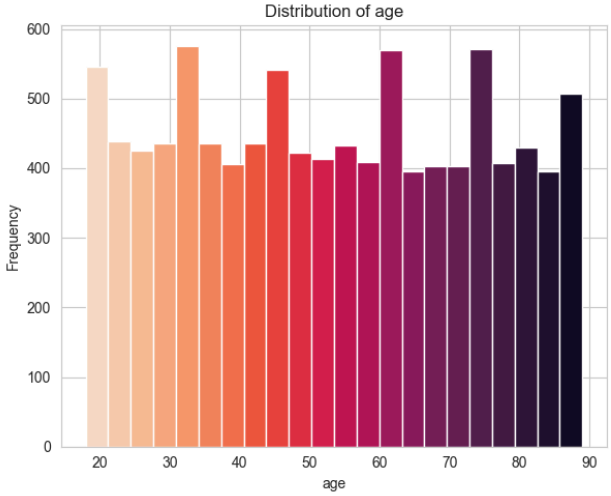
Code

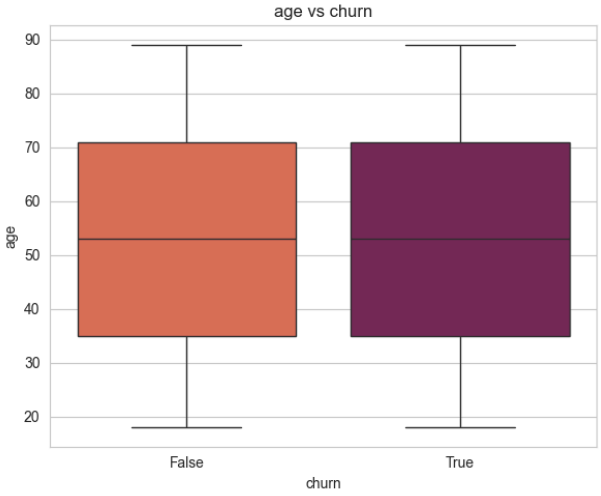
```
# C3. Generate univariate and bivariate visualizations of the distributions of the
dependent and independent variables, including the dependent variable in your bivariate
visualizations.
palette = "rocket_r"
for column in numerical_columns:
    with sns.axes_style("whitegrid"):
        # Create a figure for each column with 1 row and 2 columns
        fig, axes = plt.subplots(1, 2, figsize=(12, 5))

        # Histogram on the left
        histplot = sns.histplot(df[column], kde=False, ax=axes[0])
        axes[0].set_title(f'Distribution of {column}')
        axes[0].set_xlabel(column)
        axes[0].set_ylabel('Frequency')

        cm = sns.color_palette(palette, len(histplot.patches))
        for bin_, i in zip(histplot.patches, cm):
            bin_.set_facecolor(i)

        # Box plot on the right comparing with 'Churn'
        sns.boxplot(x='churn', y=column, data=df, ax=axes[1], palette=palette)
        axes[1].set_title(f'{column} vs churn')
        axes[1].set_xlabel('churn')
        axes[1].set_ylabel(column)

        # Display the plot
        plt.tight_layout()
        plt.show()
```

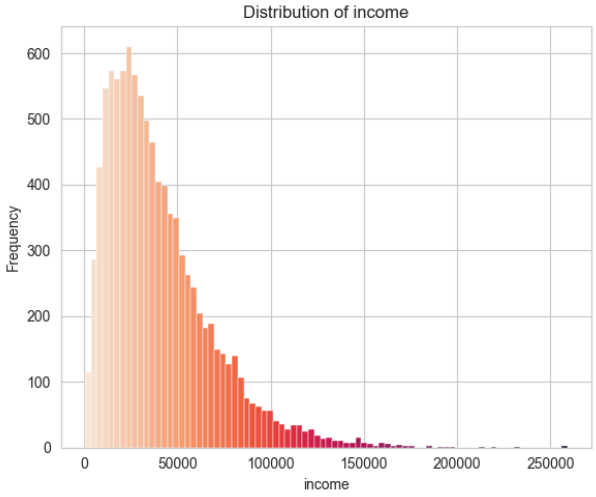
```
palette = "crest"
for column in (categorical_columns + boolean_columns):
    with sns.axes_style("whitegrid"):
        fig, axes = plt.subplots(1, 2, figsize=(14, 6))

        # Pie chart on the left
        df[column].value_counts().plot.pie(ax=axes[0], autopct='%1.1f%%', startangle=90,
colormap=palette, explode=[0.1]*df[column].nunique())
        axes[0].set_title(f'Distribution of {column}')
        axes[0].set_ylabel('')  # Hide the y-label

        # Contingency table on the right
        contingency_table = pd.crosstab(df[column], df['churn'])
        sns.heatmap(contingency_table, annot=True, fmt="d", cmap=palette, ax=axes[1],
cbar=False)
        axes[1].set_title(f'{column} vs churn')
        axes[1].set_xlabel('churn')
        axes[1].set_ylabel(column)

        # Adjust layout and display the plot
        plt.tight_layout()
        plt.show()
```

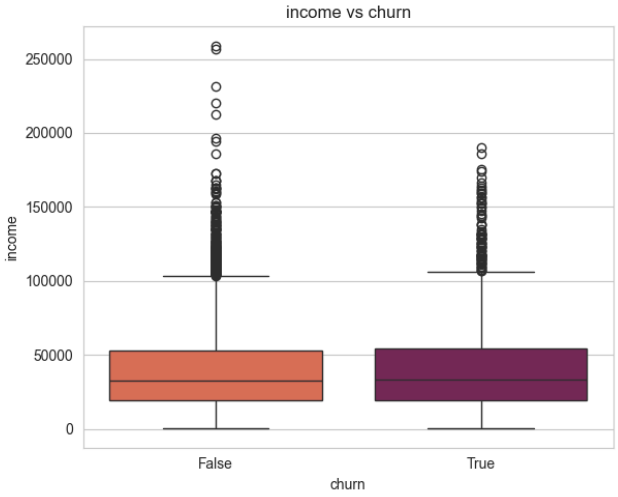

Distribution of children
Frequency
children
0
500
1000
1500
2000
2500
0
2
4
6
8
10

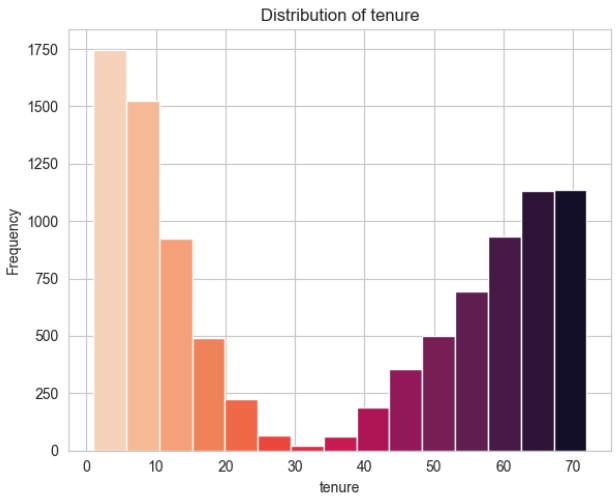

children vs churn
children
churn
False
True
0
2
4
6
8
10

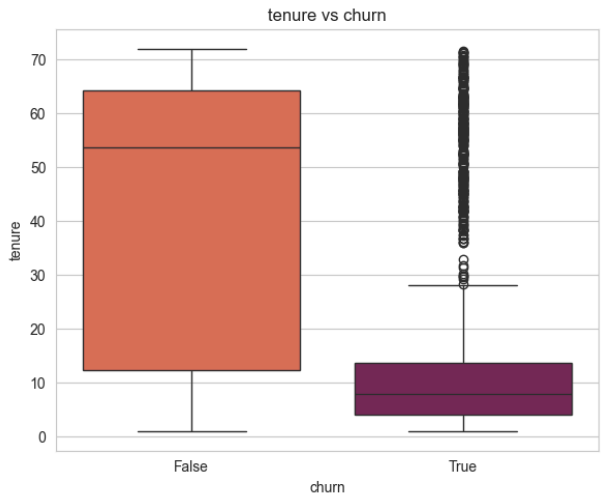

Distribution of age
Frequency
age
0
100
200
300
400
500
600
20
30
40
50
60
70
80
90

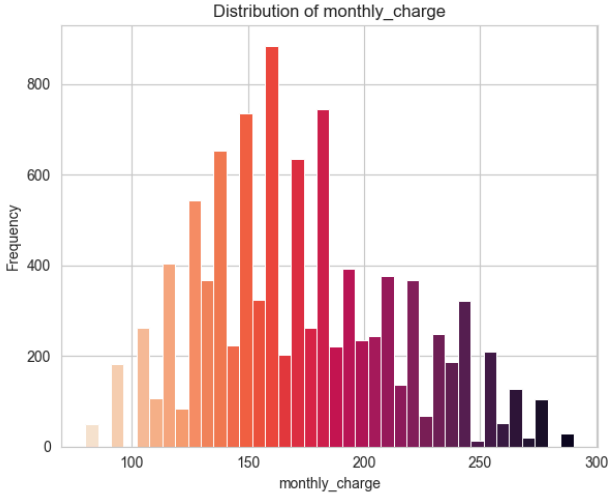

age vs churn
age
churn
False
True
20
30
40
50
60
70
80
90

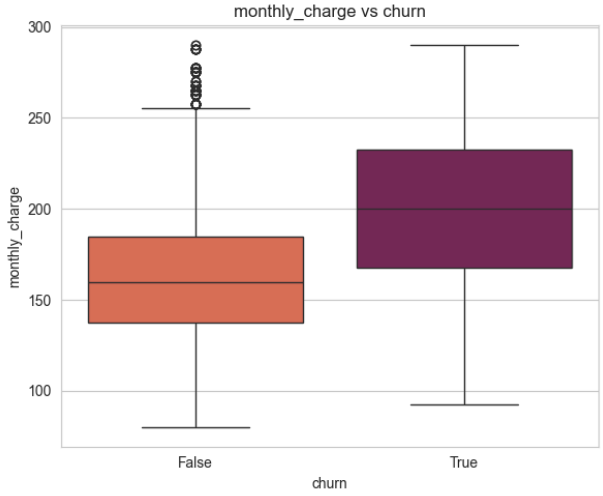

Distribution of income
Frequency
income
0
100
200
300
400
500
600
0
50000
100000
150000
200000
250000

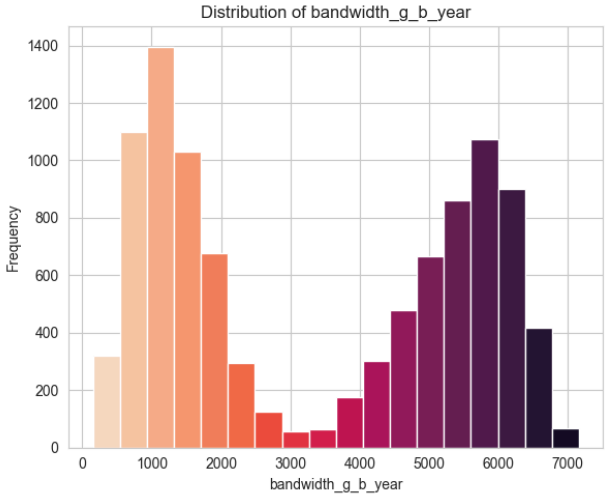

income vs churn
income
churn
False
True
0
50000
100000
150000
200000
250000

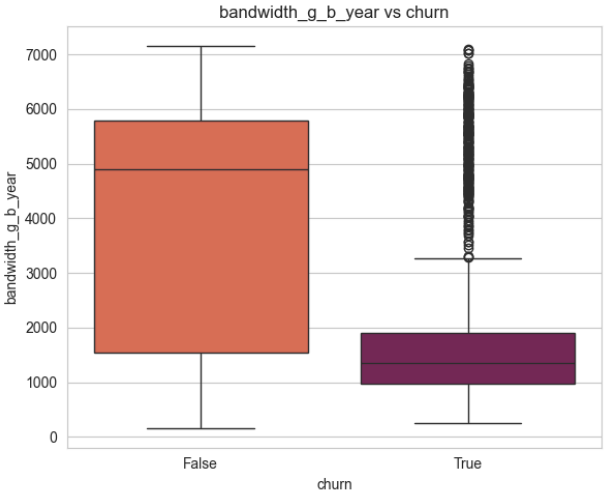

Distribution of tenure
Frequency
tenure
0
250
500
750
1000
1250
1500
1750
0
10
20
30
40
50
60
70

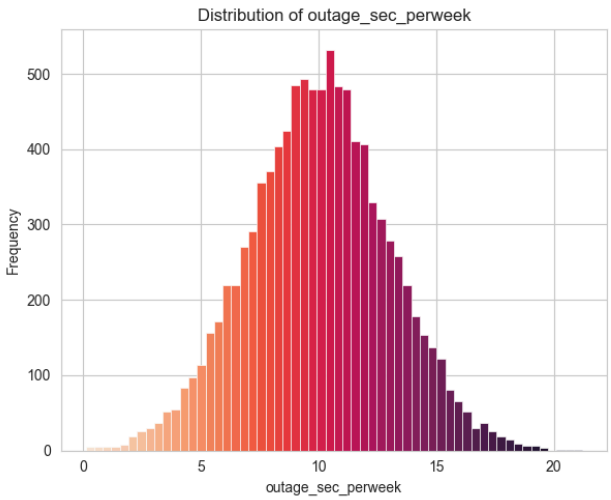

tenure vs churn
tenure
churn
False
True
0
10
20
30
40
50
60
70

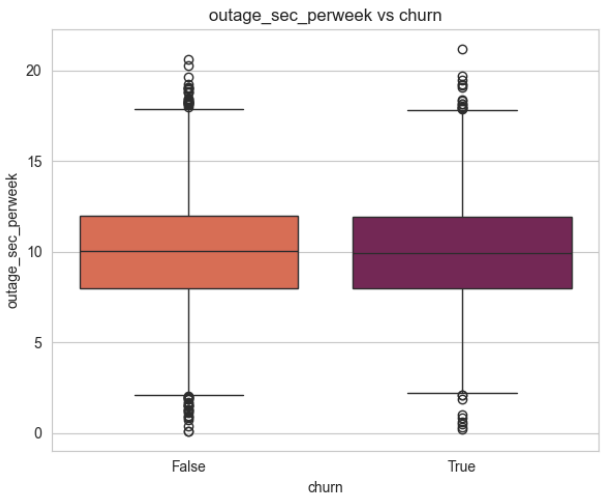

Distribution of monthly_charge
Frequency
monthly_charge
0
200
400
600
800
100
150
200
250
300

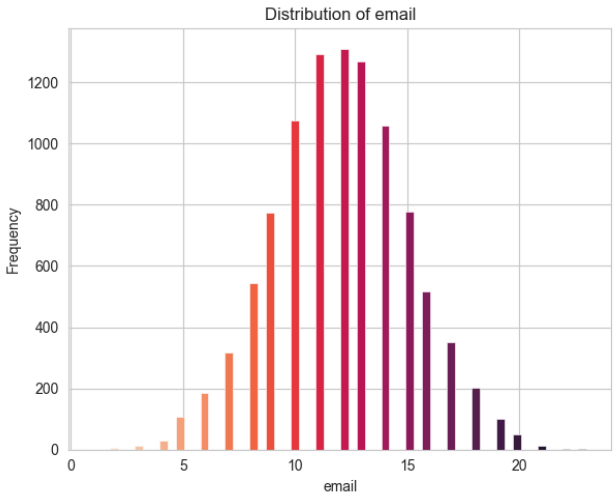

monthly_charge vs churn
monthly_charge
churn
False
True
100
150
200
250
300

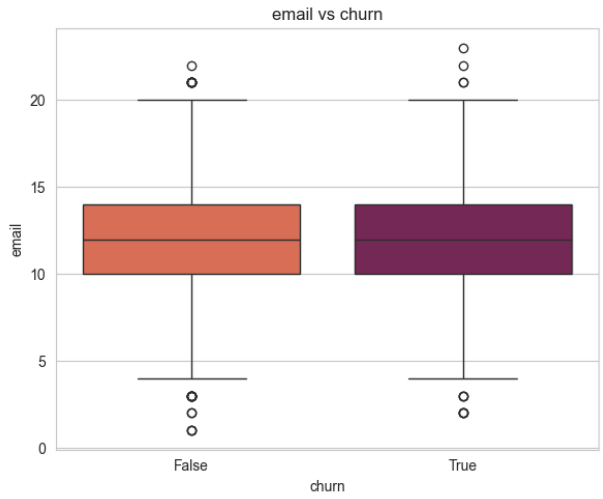

Distribution of bandwidth_g_b_year
Frequency
bandwidth_g_b_year
0
200
400
600
800
1000
1200
1400
0
1000
2000
3000
4000
5000
6000
7000

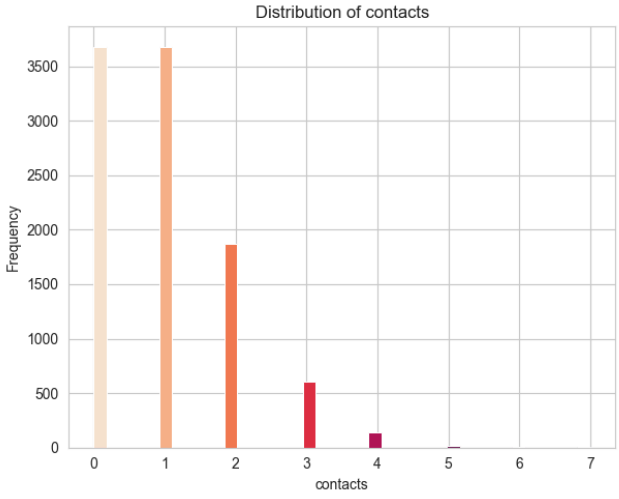

bandwidth_g_b_year vs churn
bandwidth_g_b_year
churn
False
True
0
1000
2000
3000
4000
5000
6000
7000

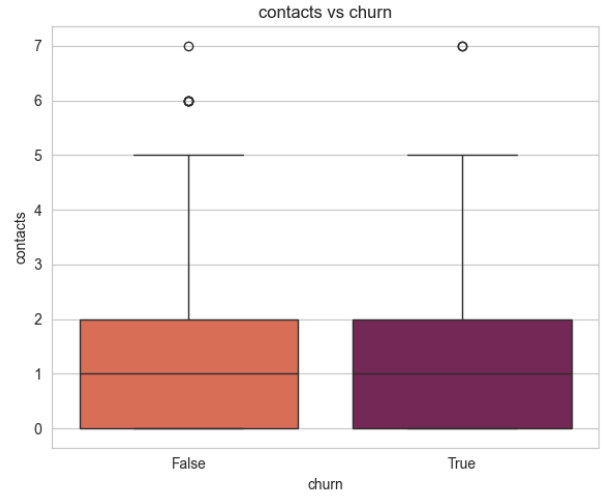

Distribution of outage_sec_perweek
Frequency
outage_sec_perweek
0
100
200
300
400
500
0
5
10
15
20

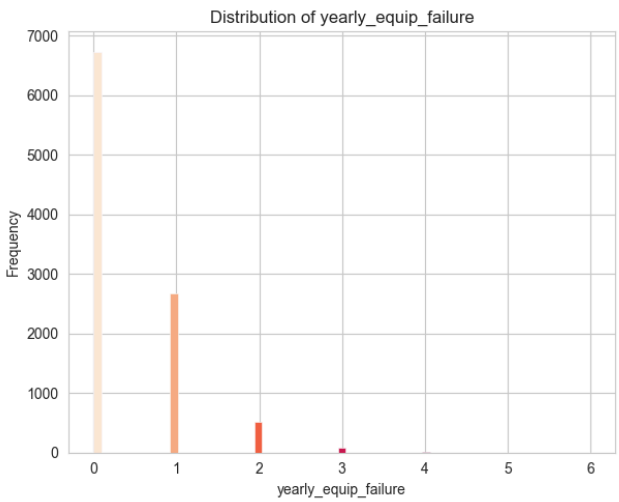

outage_sec_perweek vs churn
outage_sec_perweek
churn
False
True
0
5
10
15
20

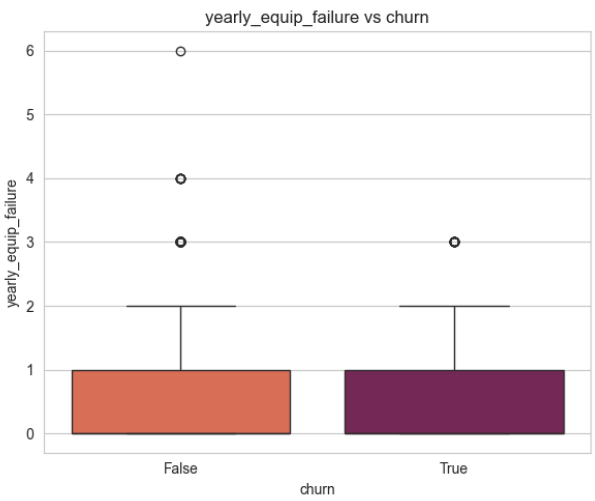

Distribution of email
Frequency
email
0
200
400
600
800
1000
1200
0
5
10
15
20

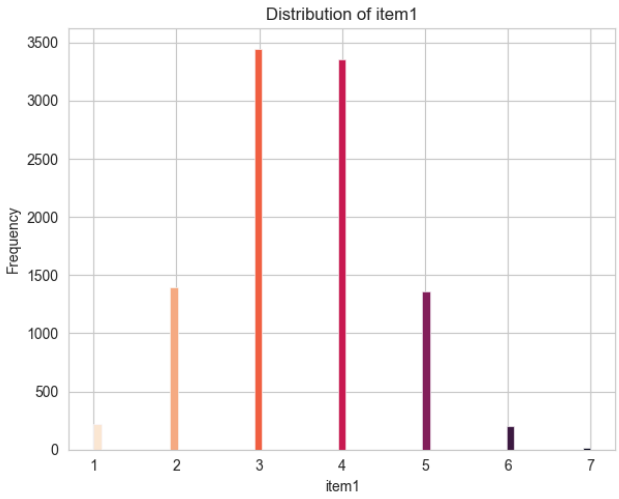

email vs churn
email
churn
False
True
0
5
10
15
20

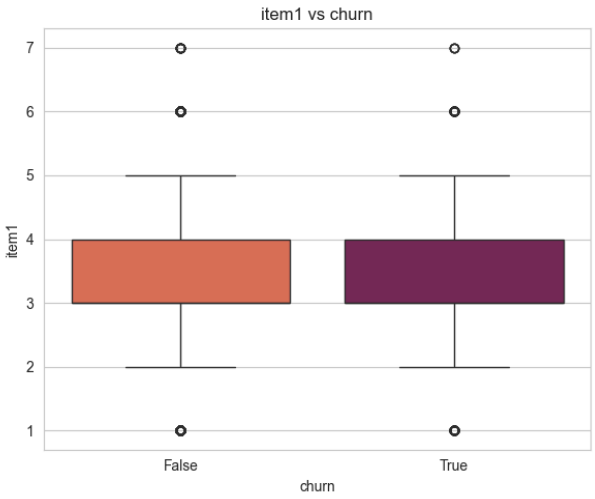

Distribution of contacts
Frequency
contacts
0
500
1000
1500
2000
2500
3000
3500
0
1
2
3
4
5
6
7

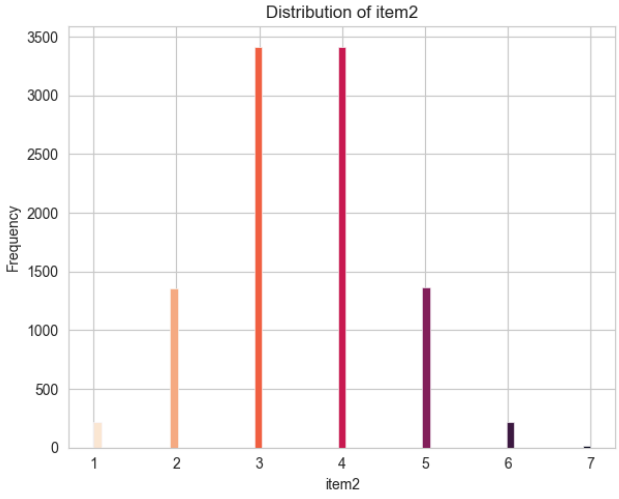

contacts vs churn
contacts
churn
False
True
0
1
2
3
4
5
6
7

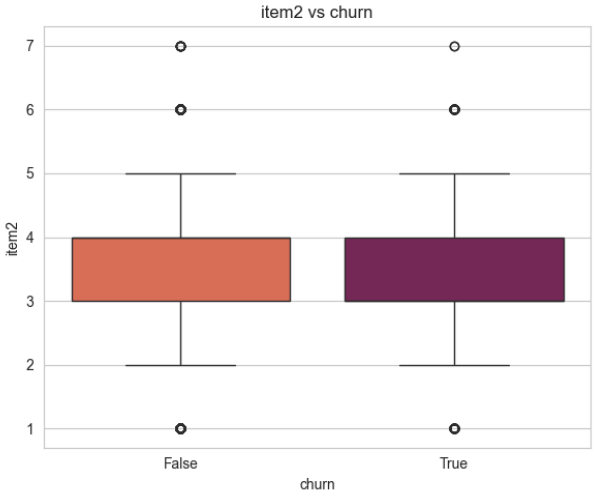

Distribution of yearly_equip_failure
Frequency
yearly_equip_failure
0
1000
2000
3000
4000
5000
6000
7000
0
1
2
3
4
5
6

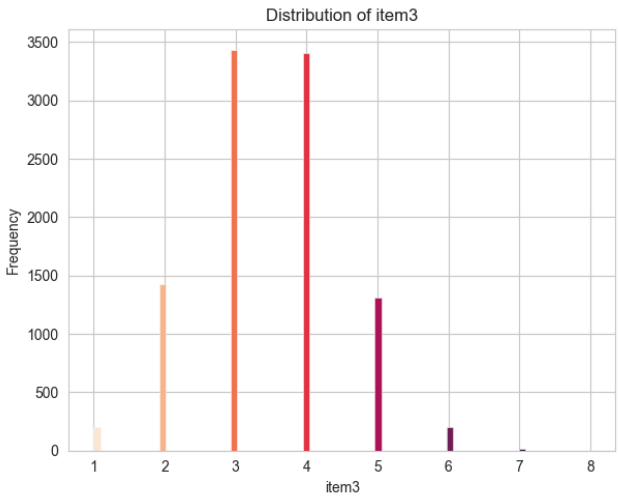

yearly_equip_failure vs churn
yearly_equip_failure
churn
False
True
0
1
2
3
4
5
6

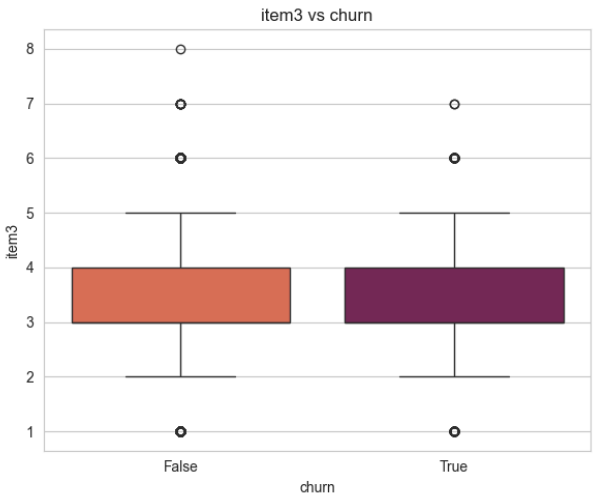

Distribution of item1
Frequency
item1
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7

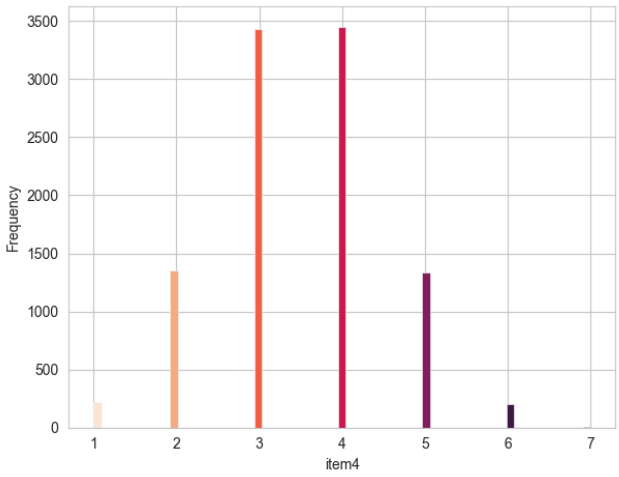

item1 vs churn
item1
churn
False
True
1
2
3
4
5
6
7

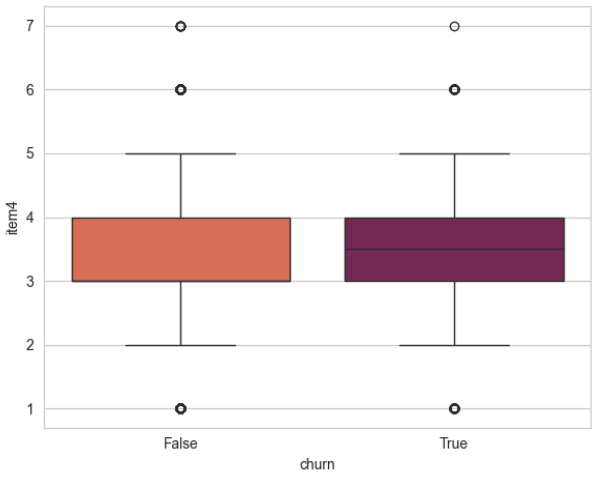

Distribution of item2
Frequency
item2
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7

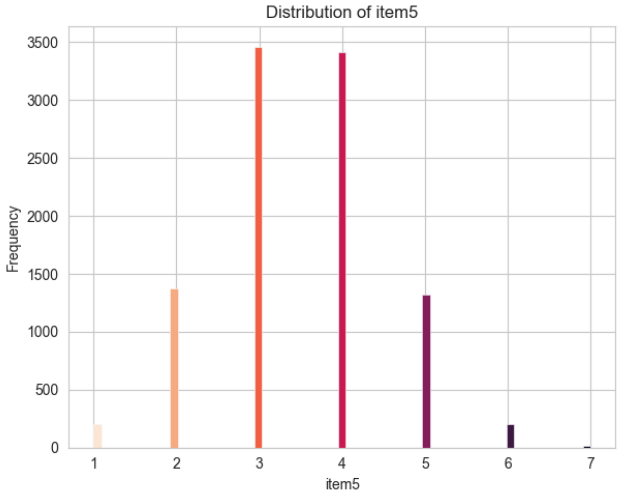

item2 vs churn
item2
churn
False
True
1
2
3
4
5
6
7

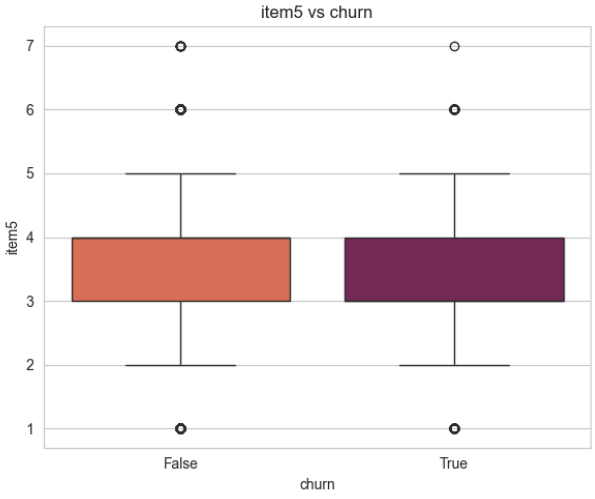

Distribution of item3
Frequency
item3
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7
8

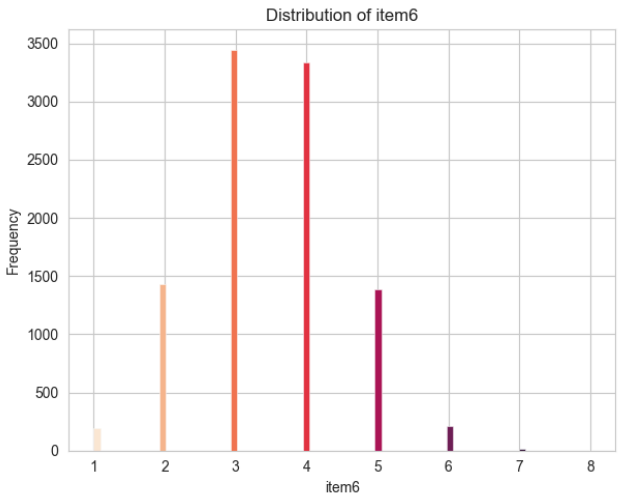

item3 vs churn
item3
churn
False
True
1
2
3
4
5
6
7
8

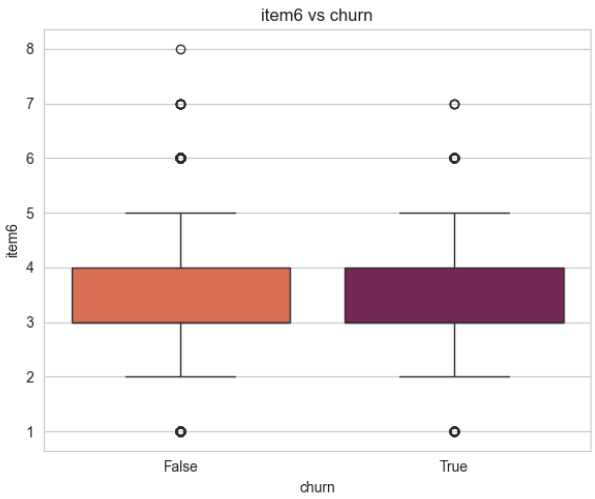

Distribution of item4
Frequency
item4
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7

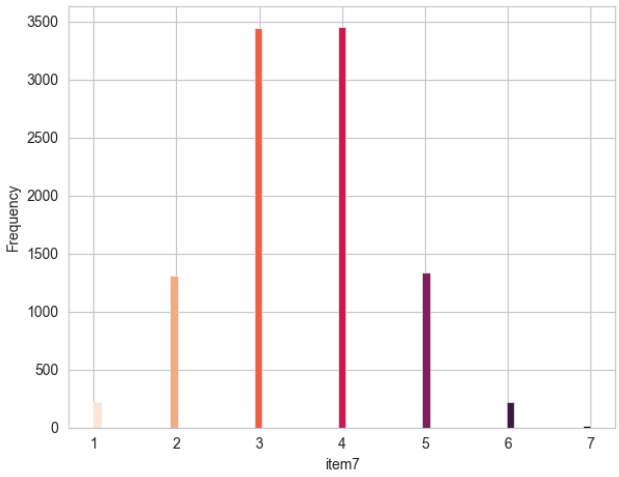

item4 vs churn
item4
churn
False
True
1
2
3
4
5
6
7

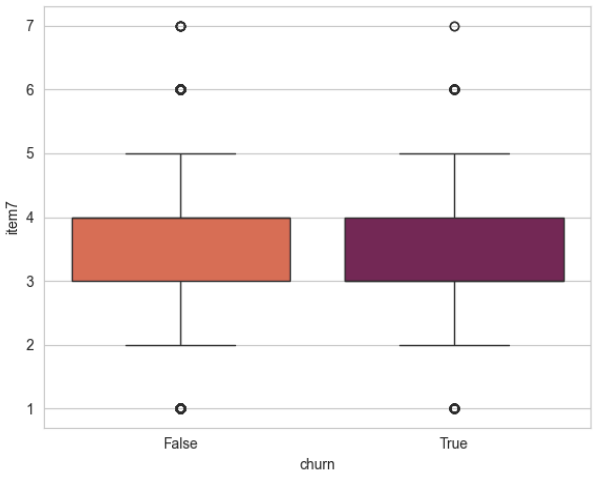

Distribution of item5
Frequency
item5
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7

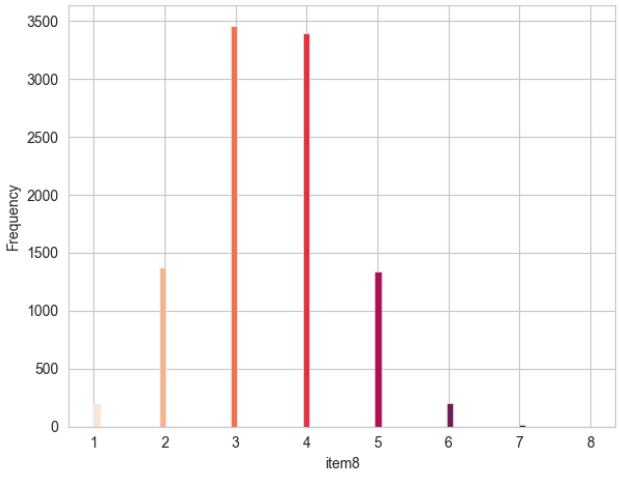

item5 vs churn
item5
churn
False
True
1
2
3
4
5
6
7

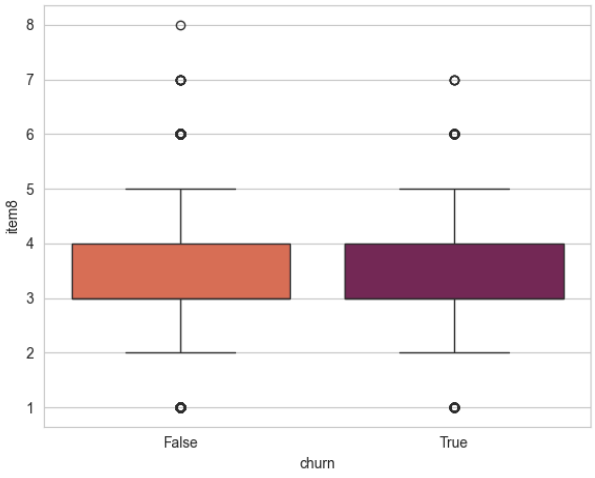

Distribution of item6
Frequency
item6
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7
8

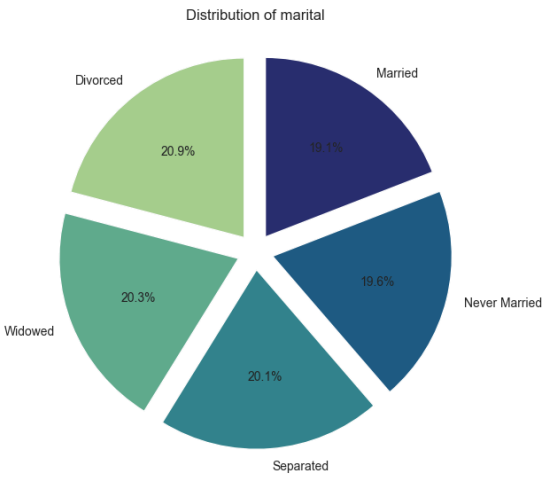

item6 vs churn
item6
churn
False
True
1
2
3
4
5
6
7
8

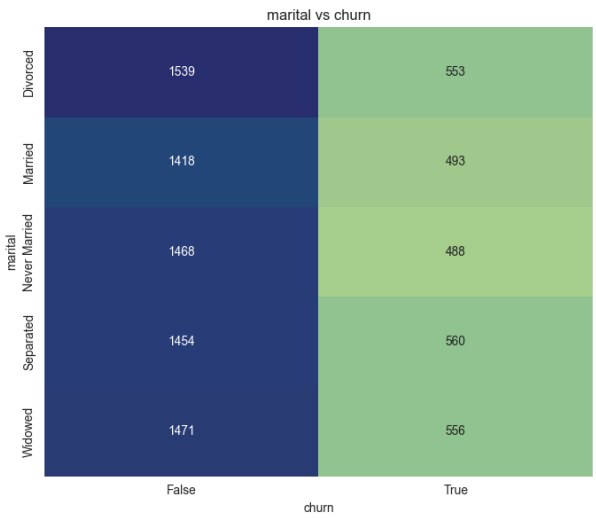

Distribution of item7
Frequency
item7
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7

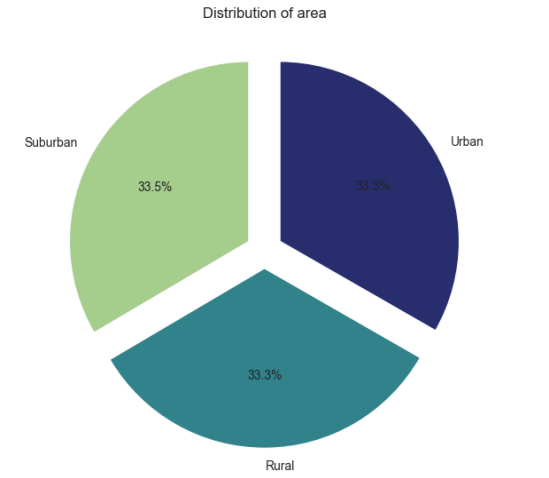

item7 vs churn
item7
churn
False
True
1
2
3
4
5
6
7

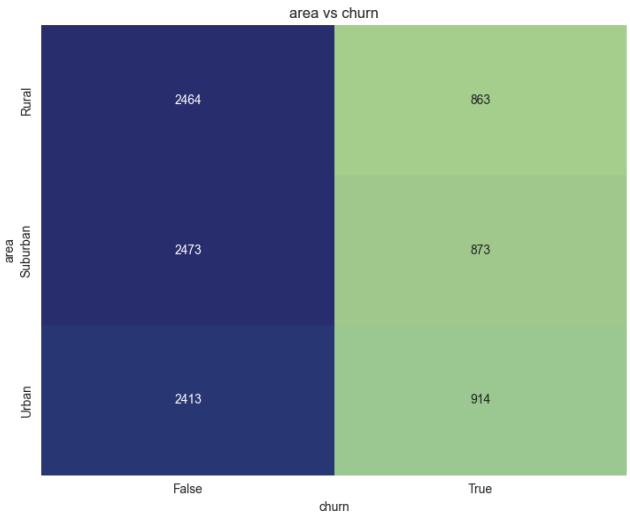

Distribution of item8
Frequency
item8
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7
8

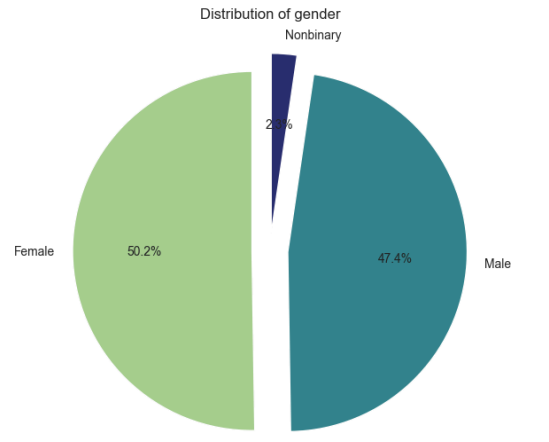

item8 vs churn
item8
churn
False
True
1
2
3
4
5
6
7
8

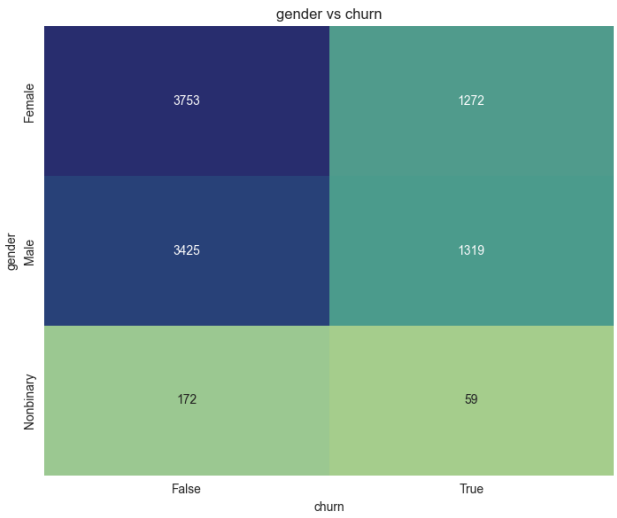

Distribution of marital
Divorced
Married
20.9%
19.1%
20.3%
19.6%
Widowed
Never Married
20.1%
Separated

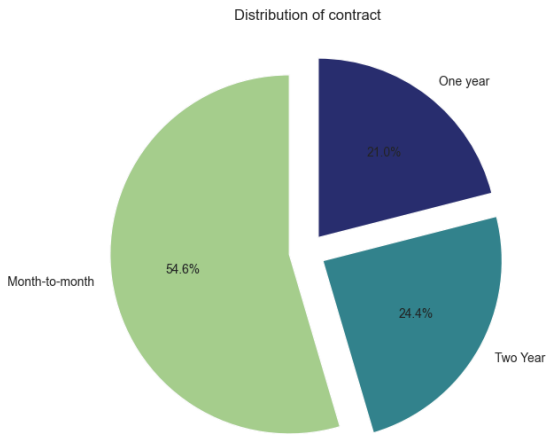

marital vs churn
Divorced
Married
Never Married
Separated
Widowed
marital
1539
553
1418
493
1468
488
1454
560
1471
556
False
True
churn

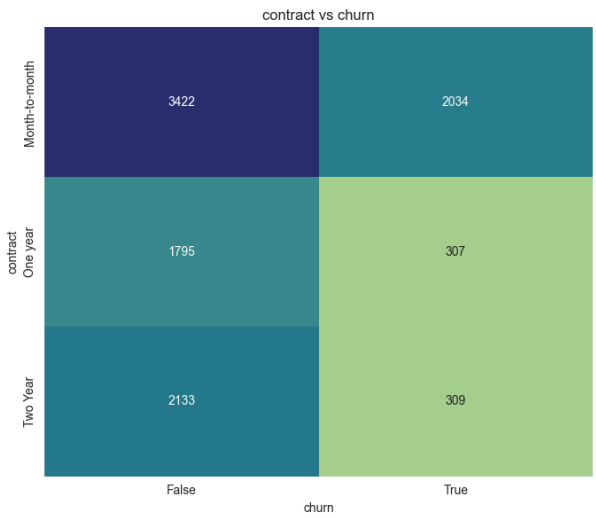

Distribution of area
Suburban
Urban
33.5%
33.3%
33.3%
Rural

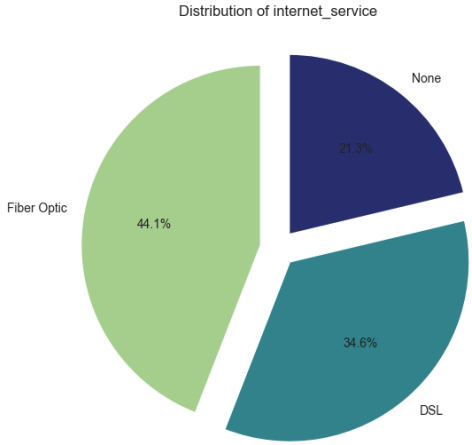

area vs churn
Rural
Suburban
Urban
area
2464
863
2473
873
2413
914
False
True
churn

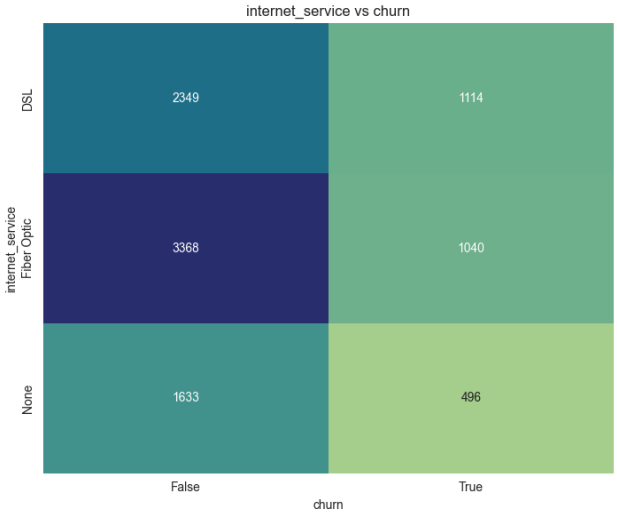

Distribution of gender
Nonbinary
2.4%
Female
50.2%
47.4%
Male

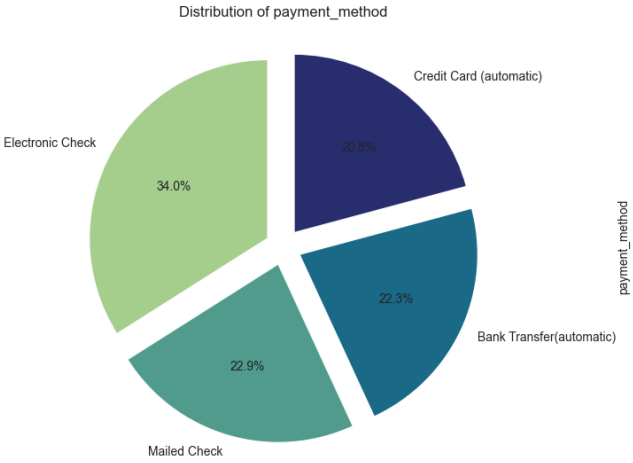

gender vs churn
Female
Male
Nonbinary
gender
3753
1272
3425
1319
172
59
False
True
churn

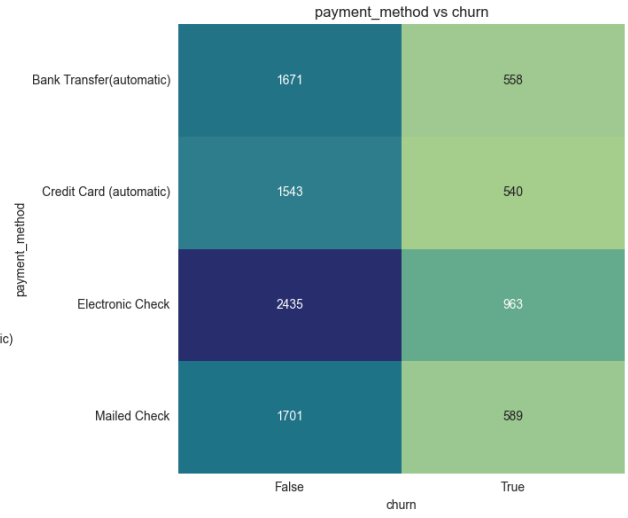

Distribution of contract
One year
21.0%
Month-to-month
54.6%
24.4%
Two Year

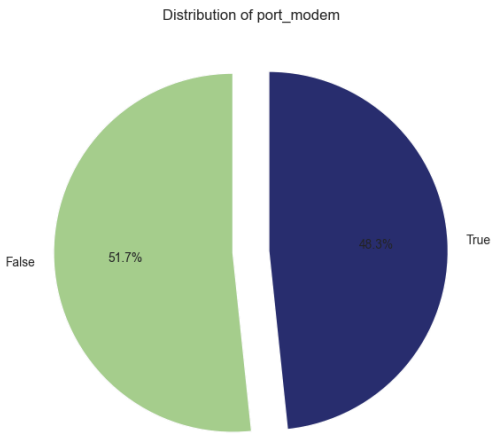

contract vs churn
Month-to-month
3422
2034
contract
One year
1795
307
Two Year
2133
309
False
True
churn

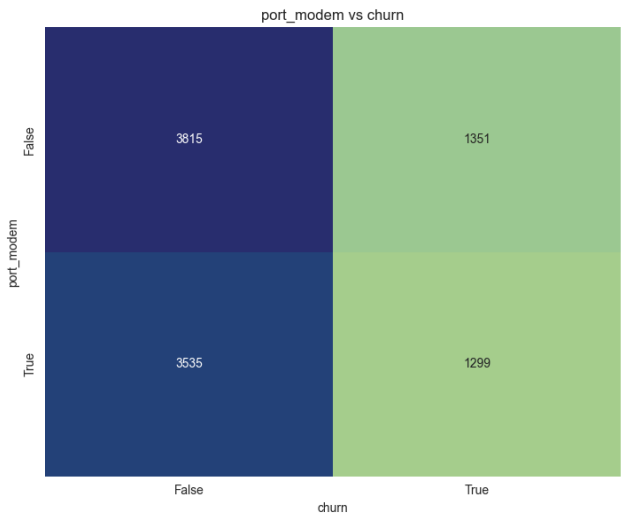

Distribution of internet_service
None
21.3%
Fiber Optic
44.1%
34.6%
DSL

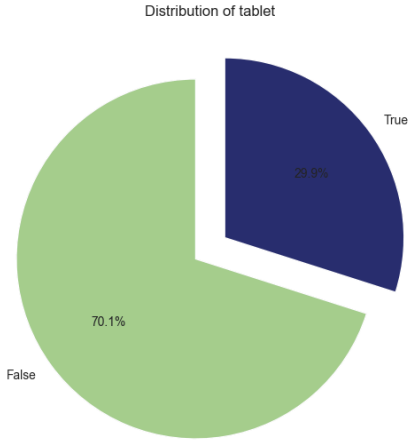

internet_service vs churn
DSL
2349
1114
internet_service
Fiber Optic
3368
1040
None
1633
496
False
True
churn

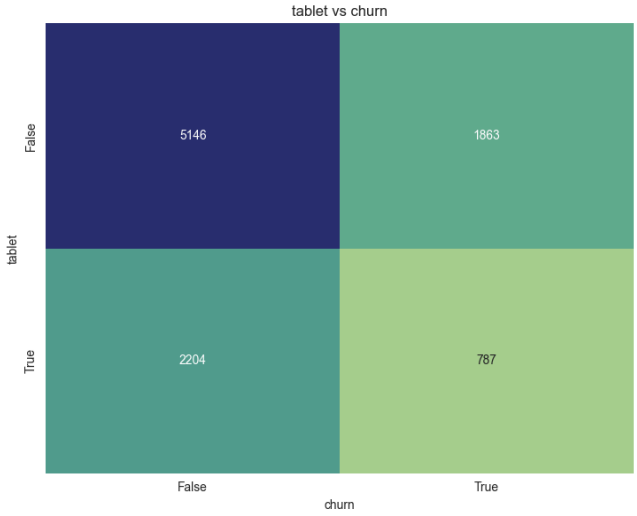

Distribution of payment_method
Credit Card (automatic)
20.8%
Electronic Check
34.0%
22.3%
Bank Transfer(automatic)
22.9%
Mailed Check

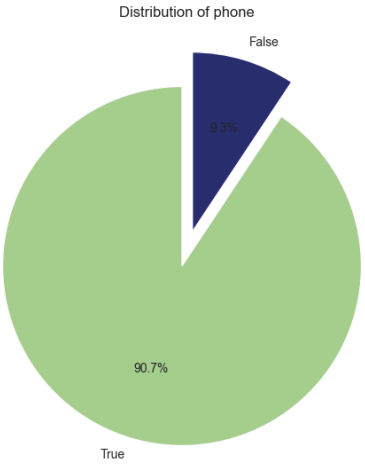

payment_method vs churn
Bank Transfer(automatic)
1671
558
Credit Card (automatic)
1543
540
payment_method
Electronic Check
2435
963
Mailed Check
1701
589
False
True
churn

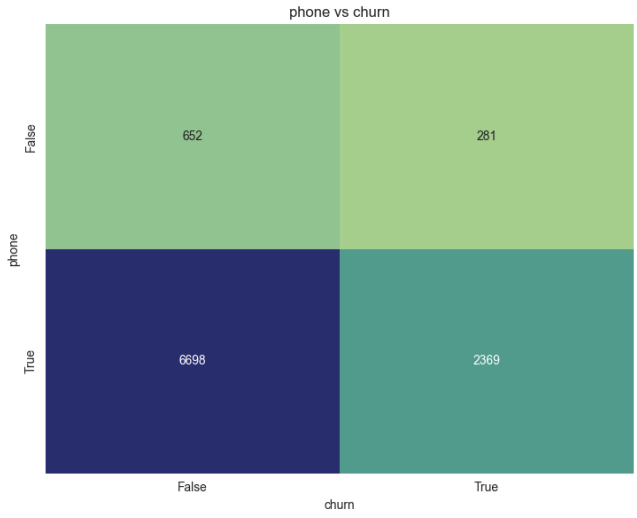

Distribution of port_modem
False
51.7%
48.3%
True

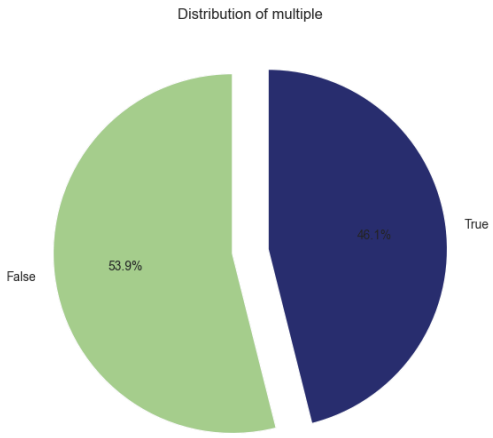

port_modem vs churn
port_modem
False
True
3815
1351
3535
1299
False
True
churn

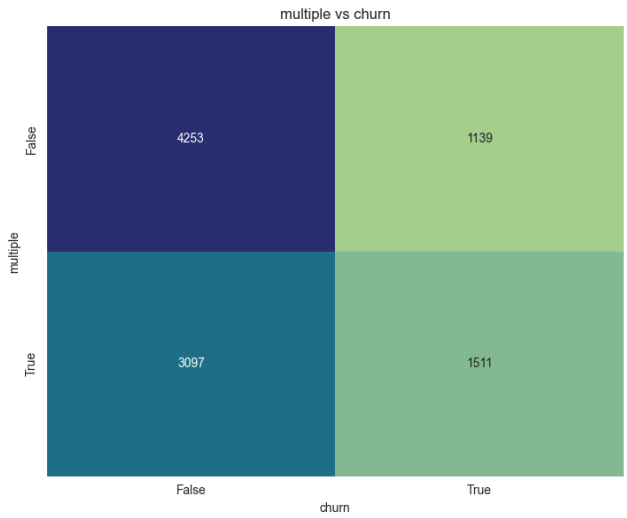

Distribution of tablet
True
29.9%
70.1%
False

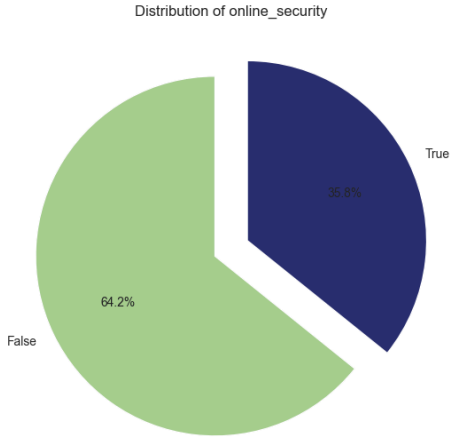

tablet vs churn
tablet
False
True
5146
1863
2204
787
False
True
churn

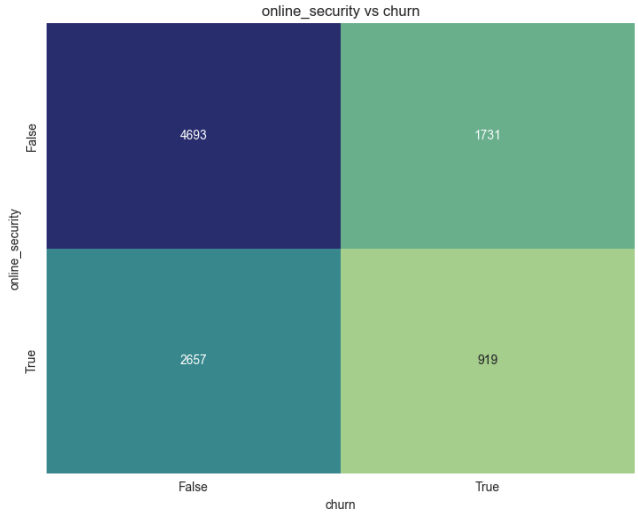

Distribution of phone
False
9.3%
90.7%
True

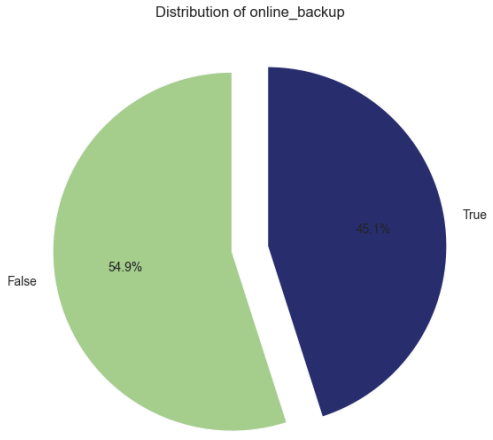

phone vs churn
phone
False
True
652
281
6698
2369
False
True
churn

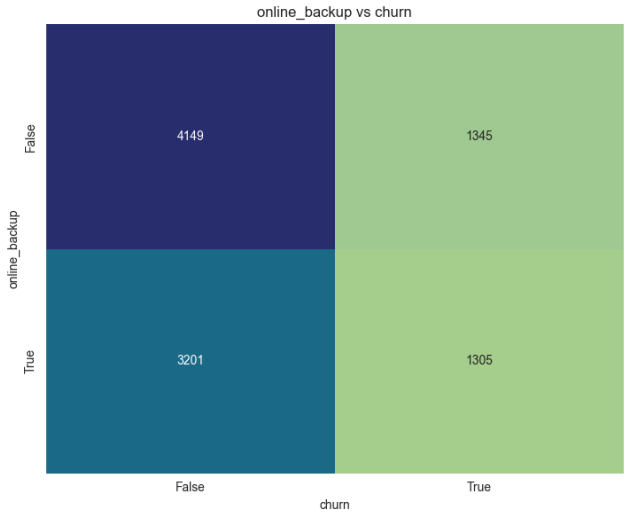

Distribution of multiple
True
False
46.1%
53.9%

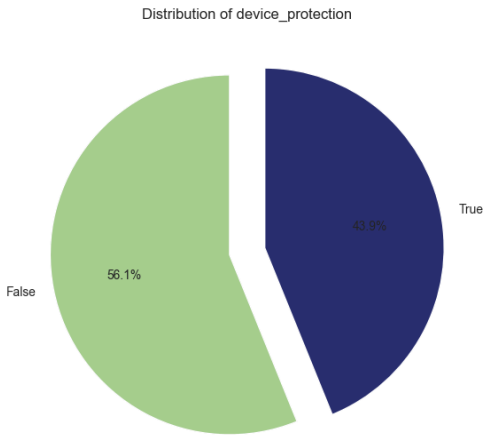

multiple vs churn
multiple
False
True
4253
1139
3097
1511
False
True
churn

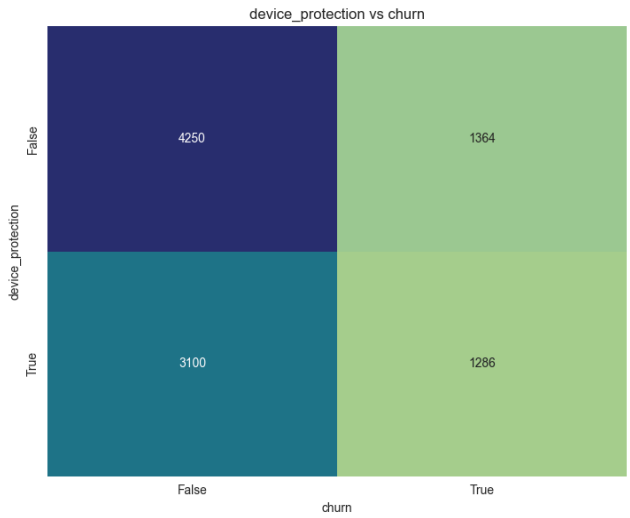

Distribution of online_security
True
False
35.8%
64.2%

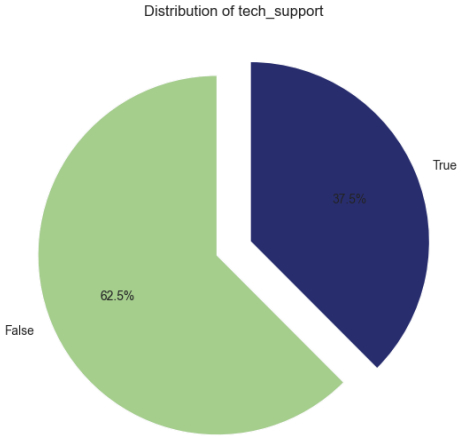

online_security vs churn
online_security
False
True
4693
1731
2657
919
False
True
churn

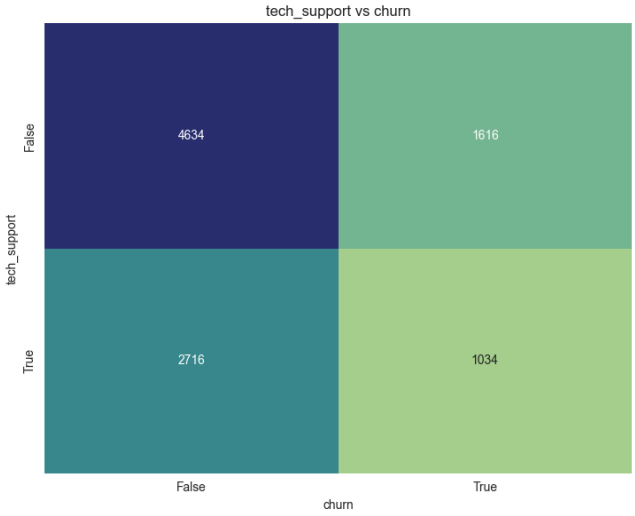

Distribution of online_backup
True
False
45.1%
54.9%

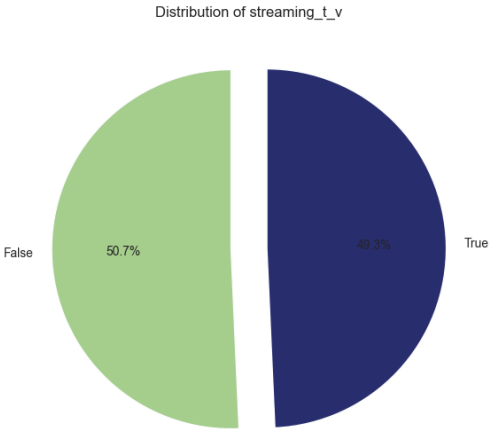

online_backup vs churn
online_backup
False
True
4149
1345
3201
1305
False
True
churn

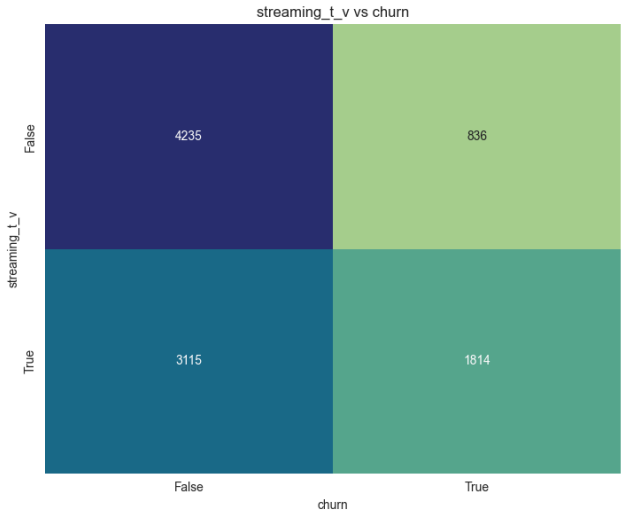

Distribution of device_protection
True
False
43.9%
56.1%

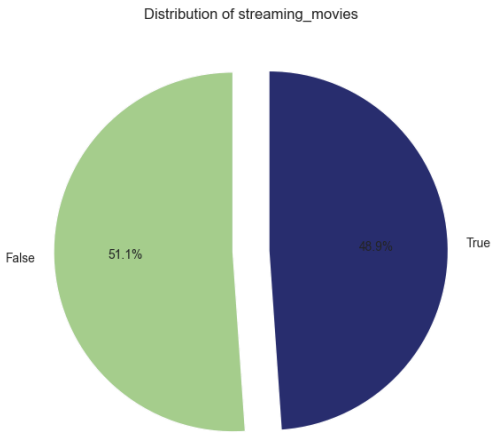

device_protection vs churn
False
True
device_protection
4250
1364
3100
1286
False
True
churn

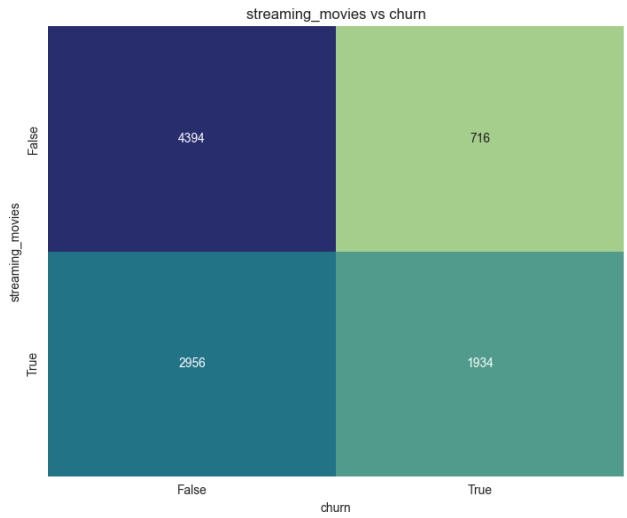

Distribution of tech_support
True
False
37.5%
62.5%

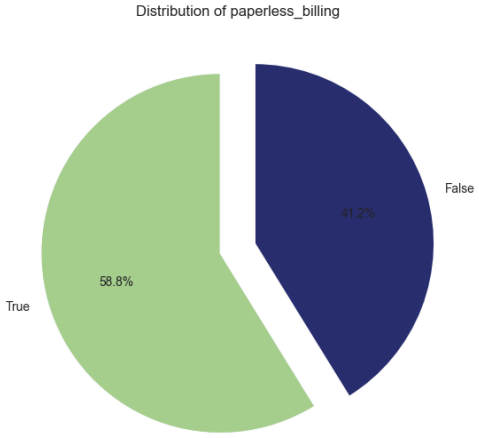

tech_support vs churn
False
True
tech_support
4634
1616
2716
1034
False
True
churn

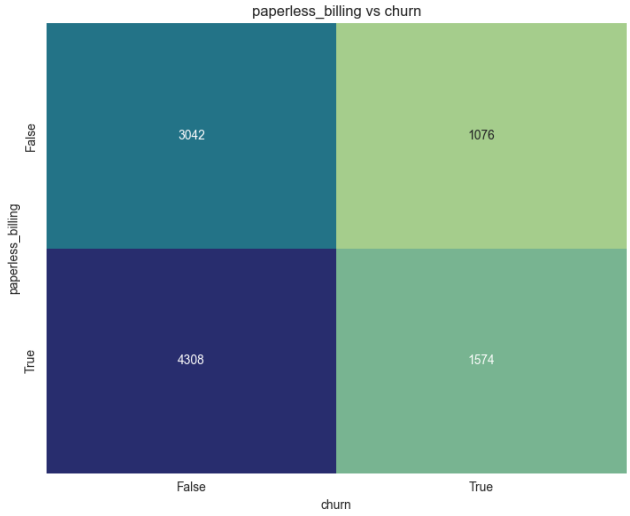

Distribution of streaming_t_v
False
True
50.7%
49.3%

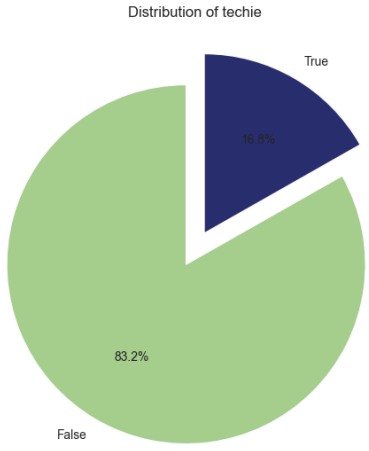

streaming_t_v vs churn
False
True
streaming_t_v
4235
836
3115
1814
False
True
churn

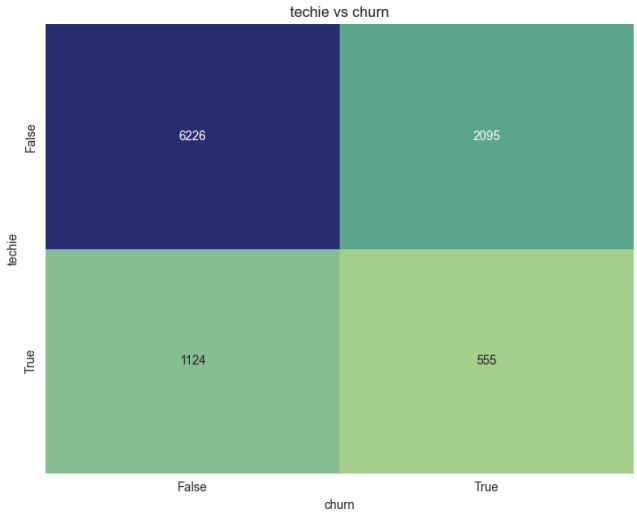

Distribution of streaming_movies
False
51.1%
48.9%
True


streaming_movies vs churn
streaming_movies
False
True
4394
716
2956
1934
False
True
churn


Distribution of paperless_billing
False
41.2%
58.8%
True


paperless_billing vs churn
paperless_billing
False
True
3042
1076
4308
1574
False
True
churn


Distribution of techie
True
16.8%
83.2%
False


techie vs churn
techie
False
True
6226
2095
1124
555
False
True
churn

C4. Describe your data transformation goals that align with your research question and the steps used to transform the data to achieve the goals, including the annotated code.

My goal with regards to cleaning the sample data is to create a uniform DataFrame to which logistic regression can be applied and from which useful business insights can be drawn.

After data visualization and prior to constructing an initial logistic regression model, one final step of data preparation is required: creating dummy columns for categorical variables. As previously described in Part C1, this was done after visualization in order to visualize categorical data properly.

Code

```
# C4. Describe your data transformation goals that align with your research question and
the steps used to transform the data to achieve the goals, including the annotated code.

# Create dummy variables for specified columns and drop the first category of each
df = pd.get_dummies(df, columns=categorical_columns, drop_first=True)

# Rename all columns using snake_case
df = to_snake_case(df)

print(df.head())
df.to_csv("churn_encoded.csv")
```

Result

```
   children  age  ...  payment_method_electronic_check  payment_method_mailed_check
0         0   68  ...                            False                        False
1         1   27  ...                            False                        False
2         4   50  ...                            False                        False
3         1   48  ...                            False                         True
4         0   83  ...                            False                         True

[5 rows x 46 columns]
```

All steps of the data preparation process, including annotations, can be found within the attached file main.py.

C5. Provide the prepared data set as a CSV file.

See churn_encoded.csv.

D1. Compare an initial and a reduced logistic regression model by doing the following: Construct an initial logistic regression model from all independent variables that were identified in part C2.

The following code below constructs an initial logistic regression model using the independent variables identified in part C2 with the creation of dummy variable columns. The output of the initial logistic regression model along with its evaluation metrics are shown below.

Code

```
# D1. Compare an initial and a reduced logistic regression model by doing the following:
Construct an initial logistic regression model from all independent variables that were
identified in part C2.
X = df.drop(columns=['churn']).astype(int)
y = df['churn']

X = sm.add_constant(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state =
23)

initial_model = sm.Logit(y_train, X_train).fit()

y_pred = initial_model.predict(X_test)
predicted_classes = (y_pred > 0.5).astype(int)

# Evaluate the model
accuracy = accuracy_score(y_test, predicted_classes)
precision = precision_score(y_test, predicted_classes)
recall = recall_score(y_test, predicted_classes)
conf_matrix = confusion_matrix(y_test, predicted_classes)

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print('Confusion Matrix:')
print(conf_matrix)

print(initial_model.summary())

# Save training and test variables to CSV
X_train.to_csv('churn_X_train.csv')
X_test.to_csv('churn_X_test.csv')
y_train.to_csv('churn_y_train.csv')
y_test.to_csv('churn_y_test.csv')
```

Result

```
Optimization terminated successfully.
         Current function value: 0.216889
         Iterations 9
Accuracy: 0.904
Precision: 0.8348968105065666
Recall: 0.8105646630236795
Confusion Matrix:
[[1363   88]
 [ 104  445]]
                           Logit Regression Results
==============================================================================
Dep. Variable:                  churn   No. Observations:                 8000
Model:                          Logit   Df Residuals:                     7954
Method:                           MLE   Df Model:                           45
Date:                Wed, 19 Jun 2024   Pseudo R-squ.:                 0.6233
Time:                        15:42:09   Log-Likelihood:                -1735.1
converged:                       True   LL-Null:                       -4606.3
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
```

const	-4.3289	0.896	-4.830	0.000	-6.085	-2.572
children	0.0616	0.056	1.095	0.273	-0.049	0.172
age	-0.0026	0.006	-0.439	0.661	-0.014	0.009
income	1.927e-07	1.55e-06	0.124	0.901	-2.85e-06	3.23e-06
port_modem	0.1359	0.087	1.566	0.117	-0.034	0.306
tablet	-0.0484	0.095	-0.512	0.608	-0.234	0.137
phone	-0.2615	0.148	-1.768	0.077	-0.551	0.028
multiple	0.3795	0.197	1.926	0.054	-0.007	0.766
online_security	-0.1950	0.149	-1.313	0.189	-0.486	0.096
online_backup	-0.1039	0.152	-0.683	0.495	-0.402	0.194
device_protection	-0.0432	0.137	-0.315	0.753	-0.312	0.226
tech_support	-0.2704	0.124	-2.173	0.030	-0.514	-0.026
streaming_t_v	1.3820	0.305	4.537	0.000	0.785	1.979
streaming_movies	1.4534	0.311	4.678	0.000	0.844	2.062
paperless_billing	0.1359	0.088	1.544	0.123	-0.037	0.308
tenure	-0.0037	0.140	-0.027	0.979	-0.277	0.270
monthly_charge	0.0443	0.007	6.061	0.000	0.030	0.059
bandwidth_g_b_year	-0.0014	0.002	-0.814	0.416	-0.005	0.002
outage_sec_perweek	-0.0096	0.015	-0.656	0.512	-0.038	0.019
email	-0.0051	0.014	-0.357	0.721	-0.033	0.023
contacts	0.0751	0.043	1.726	0.084	-0.010	0.160
yearly_equip_failure	-0.0047	0.068	-0.069	0.945	-0.138	0.129
techie	1.0517	0.116	9.087	0.000	0.825	1.278
item1	-0.0344	0.061	-0.561	0.575	-0.155	0.086
item2	0.0033	0.058	0.057	0.954	-0.111	0.118
item3	0.0096	0.053	0.183	0.855	-0.093	0.113
item4	-0.0475	0.047	-1.010	0.312	-0.140	0.045
item5	-0.0350	0.050	-0.706	0.480	-0.132	0.062
item6	-0.0190	0.050	-0.379	0.704	-0.117	0.079
item7	-0.0141	0.049	-0.290	0.772	-0.109	0.081
item8	-0.0187	0.045	-0.415	0.678	-0.107	0.070
marital_married	0.2450	0.138	1.772	0.076	-0.026	0.516
marital_never_married	0.1287	0.136	0.944	0.345	-0.139	0.396
marital_separated	0.1659	0.135	1.226	0.220	-0.099	0.431
marital_widowed	0.3186	0.134	2.371	0.018	0.055	0.582
area_suburban	-0.0369	0.108	-0.342	0.732	-0.248	0.174
area_urban	0.0468	0.105	0.444	0.657	-0.160	0.253
gender_male	0.3736	0.137	2.728	0.006	0.105	0.642
gender_nonbinary	-0.0905	0.302	-0.300	0.764	-0.682	0.501
contract_one_year	-3.3347	0.142	-23.418	0.000	-3.614	-3.056
contract_two_year	-3.3930	0.140	-24.247	0.000	-3.667	-3.119
internet_service_fiber_optic	-2.8200	0.822	-3.432	0.001	-4.430	-1.210
internet_service_none	-1.5263	0.659	-2.317	0.020	-2.817	-0.235
payment_method_credit_card_automatic	0.2311	0.132	1.751	0.080	-0.028	0.490
payment_method_electronic_check	0.6060	0.118	5.116	0.000	0.374	0.838
payment_method_mailed_check	0.2438	0.131	1.867	0.062	-0.012	0.500

D2. Justify a statistically based feature selection procedure or a model evaluation metric to reduce the initial model in a way that aligns with the research question.

One way an organization may reduce an initial model is via expert judgment. This process is not statistical but rather involves an expert selecting features based on what typically influences the dependent variable. This process ensures the model includes variables that are contextually relevant and important but potentially overlooks important predictors that are not obvious to humans.

Therefore, a statistically based feature selection procedure is needed. For my analysis, I will be using backward elimination to remove insignificant features. The code below fits a logistic regression model and identifies the p-values of all features as defined in the documentation for statsmodels.discrete.discrete_model.LogitResults.pvalues. If there are any p-values above the threshold

of 0.05, the feature with the largest p-value is removed and the model is fitted again. This process is repeated until the p-values of all features are within the threshold.

A significance level of 0.05 means there is a 5% risk of concluding that an effect exists when there is actually no effect (i.e., a 5% chance of a Type I error). Using 0.05 as a threshold is conventional and strikes a balance between being too lenient (e.g., α = 0.10) and too stringent (e.g., α = 0.01).

As Dunkler et al. (2014) notes with regards to backward elimination in their paper *Augmented backward elimination: a pragmatic and purposeful way to develop statistical models*, “The threshold value t could be set to, say, 0.05 but can be adopted to the specific modeling situation” (Dunkler et al., 2014).

The following output is a logistic regression model using the reduced feature set.

Code

```
# D2. Justify a statistically based feature selection procedure or a model evaluation
metric to reduce the initial model in a way that aligns with the research question.

target = "churn"
significance_level = 0.05
X_reduced = X_train

round_count = 0
while True:
    round_count += 1
    print(f"\nRound {round_count} of backward elimination:")

    # Fit the model
    model = sm.Logit(y_train, X_reduced).fit()
    p_values = model.pvalues
    print(p_values)

    if p_values.max() > significance_level:
        feature_to_remove = p_values.idxmax()

        if feature_to_remove == "const":
            break

        print("Removing feature:", feature_to_remove)
        X_reduced = X_reduced.drop(columns=[feature_to_remove])
    else:
        break

print(model.summary())
```

Result

```
<29 rounds of backward elimination truncated>

Round 30 of backward elimination:
Optimization terminated successfully.
         Current function value: 0.218427
         Iterations 9
```

```
const                                5.327527e-49
children                             2.477262e-03
multiple                             1.799872e-04
online_security                      4.425392e-02
tech_support                         1.315827e-02
streaming_t_v                        2.033520e-22
streaming_movies                     4.176392e-20
monthly_charge                       2.402196e-43
bandwidth_g_b_year                  6.087203e-270
techie                               9.832033e-20
marital_widowed                      8.823944e-02
gender_male                          5.639524e-06
contract_one_year                   4.690736e-122
contract_two_year                   2.408896e-130
internet_service_fiber_optic         4.143685e-98
internet_service_none                3.111589e-33
payment_method_electronic_check      1.249693e-06
dtype: float64
```
Removing feature: marital_widowed

```
Round 31 of backward elimination:
Optimization terminated successfully.
         Current function value: 0.218608
         Iterations 9
const                                1.430042e-48
children                             2.590363e-03
multiple                             1.682156e-04
online_security                      4.560543e-02
tech_support                         1.225697e-02
streaming_t_v                        1.778620e-22
streaming_movies                     3.308490e-20
monthly_charge                       2.855350e-43
bandwidth_g_b_year                  4.791129e-270
techie                               6.775562e-20
gender_male                          7.145956e-06
contract_one_year                   4.346902e-122
contract_two_year                   2.291151e-130
internet_service_fiber_optic         4.969169e-98
internet_service_none                4.511569e-33
payment_method_electronic_check      1.375005e-06
dtype: float64
```

Logit Regression Results

Dep. Variable:	churn	No. Observations:	8000
Model:	Logit	Df Residuals:	7984
Method:	MLE	Df Model:	15
Date:	Wed, 19 Jun 2024	Pseudo R-squ.:	0.6203
Time:	15:39:16	Log-Likelihood:	-1748.9
converged:	True	LL-Null:	-4606.3
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-4.6039	0.314	-14.646	0.000	-5.220	-3.988
children	0.0618	0.021	3.013	0.003	0.022	0.102
multiple	0.4723	0.126	3.763	0.000	0.226	0.718
online_security	-0.1796	0.090	-1.999	0.046	-0.356	-0.004
tech_support	-0.2381	0.095	-2.505	0.012	-0.425	-0.052
streaming_t_v	1.5125	0.155	9.754	0.000	1.209	1.816
streaming_movies	1.6245	0.176	9.208	0.000	1.279	1.970

```
monthly_charge                    0.0409      0.003     13.792      0.000       0.035       0.047
bandwidth_g_b_year               -0.0014   4.04e-05    -35.110      0.000      -0.001      -0.001
techie                            1.0481      0.115      9.131      0.000       0.823       1.273
gender_male                       0.3882      0.086      4.489      0.000       0.219       0.558
contract_one_year                -3.3051      0.141    -23.497      0.000      -3.581      -3.029
contract_two_year                -3.3552      0.138    -24.294      0.000      -3.626      -3.084
internet_service_fiber_optic     -2.7511      0.131    -21.013      0.000      -3.008      -2.494
internet_service_none            -1.5583      0.130    -11.980      0.000      -1.813      -1.303
payment_method_electronic_check   0.4389      0.091      4.829      0.000       0.261       0.617
==============================================================================================================
```

D3. Provide a reduced logistic regression model that follows the feature selection or model evaluation process in part D2, including a screenshot of the output for each model.

The following code evaluates the reduced model by calculating its accuracy, precision, recall, and confusion matrix.

Code

```
# D3. Provide a reduced logistic regression model that follows the feature selection or
model evaluation process in part D2, including a screenshot of the output for each model.

# Evaluate the final model
X_test_reduced = X_test[X_reduced.columns] # X_test need to be adjusted to match the
features selected in the reduced model.
y_pred = model.predict(X_test_reduced)
predicted_classes = (y_pred > 0.5).astype(int)

accuracy = accuracy_score(y_test, predicted_classes)
precision = precision_score(y_test, predicted_classes)
recall = recall_score(y_test, predicted_classes)
conf_matrix = confusion_matrix(y_test, predicted_classes)

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print('Confusion Matrix:')
print(conf_matrix)
```

Result

```
Accuracy: 0.9065
Precision: 0.8376865671641791
Recall: 0.8178506375227687
Confusion Matrix:
[[1364   87]
 [ 100  449]]
```

The initial logistic model and its evaluations generated in Part D1 is shown below:

Initial model output and evaluation

```
Optimization terminated successfully.
```

```
         Current function value: 0.216889
         Iterations 9
```
Accuracy: 0.904
Precision: 0.8348968105065666
Recall: 0.8105646630236795
```
Confusion Matrix:
[[1363   88]
 [ 104  445]]
```
Logit Regression Results

Dep. Variable:	churn	No. Observations:	8000
Model:	Logit	Df Residuals:	7954
Method:	MLE	Df Model:	45
Date:	Wed, 19 Jun 2024	Pseudo R-squ.:	0.6233
Time:	15:42:09	Log-Likelihood:	-1735.1
converged:	True	LL-Null:	-4606.3
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-4.3289	0.896	-4.830	0.000	-6.085	-2.572
children	0.0616	0.056	1.095	0.273	-0.049	0.172
age	-0.0026	0.006	-0.439	0.661	-0.014	0.009
income	1.927e-07	1.55e-06	0.124	0.901	-2.85e-06	3.23e-06
port_modem	0.1359	0.087	1.566	0.117	-0.034	0.306
tablet	-0.0484	0.095	-0.512	0.608	-0.234	0.137
phone	-0.2615	0.148	-1.768	0.077	-0.551	0.028
multiple	0.3795	0.197	1.926	0.054	-0.007	0.766
online_security	-0.1950	0.149	-1.313	0.189	-0.486	0.096
online_backup	-0.1039	0.152	-0.683	0.495	-0.402	0.194
device_protection	-0.0432	0.137	-0.315	0.753	-0.312	0.226
tech_support	-0.2704	0.124	-2.173	0.030	-0.514	-0.026
streaming_t_v	1.3820	0.305	4.537	0.000	0.785	1.979
streaming_movies	1.4534	0.311	4.678	0.000	0.844	2.062
paperless_billing	0.1359	0.088	1.544	0.123	-0.037	0.308
tenure	-0.0037	0.140	-0.027	0.979	-0.277	0.270
monthly_charge	0.0443	0.007	6.061	0.000	0.030	0.059
bandwidth_g_b_year	-0.0014	0.002	-0.814	0.416	-0.005	0.002
outage_sec_perweek	-0.0096	0.015	-0.656	0.512	-0.038	0.019
email	-0.0051	0.014	-0.357	0.721	-0.033	0.023
contacts	0.0751	0.043	1.726	0.084	-0.010	0.160
yearly_equip_failure	-0.0047	0.068	-0.069	0.945	-0.138	0.129
techie	1.0517	0.116	9.087	0.000	0.825	1.278
item1	-0.0344	0.061	-0.561	0.575	-0.155	0.086
item2	0.0033	0.058	0.057	0.954	-0.111	0.118
item3	0.0096	0.053	0.183	0.855	-0.093	0.113
item4	-0.0475	0.047	-1.010	0.312	-0.140	0.045
item5	-0.0350	0.050	-0.706	0.480	-0.132	0.062
item6	-0.0190	0.050	-0.379	0.704	-0.117	0.079
item7	-0.0141	0.049	-0.290	0.772	-0.109	0.081
item8	-0.0187	0.045	-0.415	0.678	-0.107	0.070
marital_married	0.2450	0.138	1.772	0.076	-0.026	0.516
marital_never_married	0.1287	0.136	0.944	0.345	-0.139	0.396
marital_separated	0.1659	0.135	1.226	0.220	-0.099	0.431
marital_widowed	0.3186	0.134	2.371	0.018	0.055	0.582
area_suburban	-0.0369	0.108	-0.342	0.732	-0.248	0.174
area_urban	0.0468	0.105	0.444	0.657	-0.160	0.253
gender_male	0.3736	0.137	2.728	0.006	0.105	0.642
gender_nonbinary	-0.0905	0.302	-0.300	0.764	-0.682	0.501
contract_one_year	-3.3347	0.142	-23.418	0.000	-3.614	-3.056
contract_two_year	-3.3930	0.140	-24.247	0.000	-3.667	-3.119
internet_service_fiber_optic	-2.8200	0.822	-3.432	0.001	-4.430	-1.210
internet_service_none	-1.5263	0.659	-2.317	0.020	-2.817	-0.235
payment_method_credit_card_automatic	0.2311	0.132	1.751	0.080	-0.028	0.490
payment_method_electronic_check	0.6060	0.118	5.116	0.000	0.374	0.838
payment_method_mailed_check	0.2438	0.131	1.867	0.062	-0.012	0.500

The final logistic model generated in Part D2 is shown below

Final model output

Logit Regression Results

Dep. Variable:	churn	No. Observations:	8000
Model:	Logit	Df Residuals:	7984
Method:	MLE	Df Model:	15
Date:	Wed, 19 Jun 2024	Pseudo R-squ.:	0.6203
Time:	15:39:16	Log-Likelihood:	-1748.9
converged:	True	LL-Null:	-4606.3
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-4.6039	0.314	-14.646	0.000	-5.220	-3.988
children	0.0618	0.021	3.013	0.003	0.022	0.102
multiple	0.4723	0.126	3.763	0.000	0.226	0.718
online_security	-0.1796	0.090	-1.999	0.046	-0.356	-0.004
tech_support	-0.2381	0.095	-2.505	0.012	-0.425	-0.052
streaming_t_v	1.5125	0.155	9.754	0.000	1.209	1.816
streaming_movies	1.6245	0.176	9.208	0.000	1.279	1.970
monthly_charge	0.0409	0.003	13.792	0.000	0.035	0.047
bandwidth_g_b_year	-0.0014	4.04e-05	-35.110	0.000	-0.001	-0.001
techie	1.0481	0.115	9.131	0.000	0.823	1.273
gender_male	0.3882	0.086	4.489	0.000	0.219	0.558
contract_one_year	-3.3051	0.141	-23.497	0.000	-3.581	-3.029
contract_two_year	-3.3552	0.138	-24.294	0.000	-3.626	-3.084
internet_service_fiber_optic	-2.7511	0.131	-21.013	0.000	-3.008	-2.494
internet_service_none	-1.5583	0.130	-11.980	0.000	-1.813	-1.303
payment_method_electronic_check	0.4389	0.091	4.829	0.000	0.261	0.617

E1. Analyze the data set using your reduced logistic regression model by doing the following: Explain your data analysis process by comparing the initial logistic regression model and reduced logistic regression model, including the following element, a model evaluation metric.

The initial and reduced (final) models vary is several significant regards:

Complexity

The initial model includes 46 features whereas the reduced model includes just 16.

Accuracy

Initial Model: 0.904

Reduced Model: 0.9065

Accuracy is one of the most important evaluation metrics for logistic regression models. The initial model has an accuracy of 0.904 whereas the reduced model has an slightly increased accuracy of 0.906. While it is common to expect that reducing features might lead to a decrease in accuracy, this is not always the case. In fact, in many scenarios, reducing features can lead to better or at least comparable performance.

Removing features that do not contribute significantly to the model or are redundant can reduce noise and improve the model's performance. Irrelevant features can sometimes confuse the model and lead to overfitting.

Precision

Initial Model: 0.8349

Reduced Model: 0.8377

Between the initial model and the reduced model, the precision has slightly increased, indicating that the reduced model is a bit better at predicting true positives among the positive predictions.

Recall

Initial Model: 0.8106

Reduced Model: 0.8179

Recall, also known as Sensitivity or True Positive Rate, is the ratio of correctly predicted positive instances to the total actual positive instances. It measures how well the model captures all the actual positives.

The recall has also improved slightly with the reduced model, meaning it is a bit better at capturing actual churn cases.

Confusion Matrix

The confusion matrix shows a slight improvement in both true negatives and true positives, with a slight decrease in both false positives and false negatives for the reduced model.

Model Fit (Pseudo R-squared):

The pseudo R-squared values are very close, indicating that the reduced model still explains a similar proportion of the variance in the data.

E2. Provide the output and all calculations of the analysis you performed, including the following elements for your reduced logistic regression model, confusion matrix, accuracy calculation.

The following code was included in Part D3 and is used to evaluate the reduced model using a confusion matrix, accuracy calculation, and other evaluation metrics.

Code

```
# D3. Provide a reduced logistic regression model that follows the feature selection or
model evaluation process in part D2, including a screenshot of the output for each model.

# Evaluate the final model
X_test_reduced = X_test[X_reduced.columns] # X_test need to be adjusted to match the
features selected in the reduced model.
```

```
y_pred = model.predict(X_test_reduced)
predicted_classes = (y_pred > 0.5).astype(int)

accuracy = accuracy_score(y_test, predicted_classes)
precision = precision_score(y_test, predicted_classes)
recall = recall_score(y_test, predicted_classes)
conf_matrix = confusion_matrix(y_test, predicted_classes)

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print('Confusion Matrix:')
print(conf_matrix)
```

Result

```
Accuracy: 0.9065
Precision: 0.8376865671641791
Recall: 0.8178506375227687
Confusion Matrix:
[[1364   87]
 [ 100  449]]
```

E3. Provide an executable error-free copy of the code used to support the implementation of the logistic regression models using a Python or R file.

See main.py

F1. Summarize your findings and assumptions by doing the following: Discuss the results of your data analysis, including the following elements, a regression equation for the reduced model, an interpretation of the coefficients of the reduced model, the statistical and practical significance of the reduced model, the limitations of the data analysis.

Using the coefficients of the reduced logistic regression equation, we can construct the following regression equation to calculate the log odds customer churn based on the features of a specific customer:

log(p/1-p) = −4.6039 + 0.0618 * children + 0.4723 * multiple−0.1796 * online_security−0.2381 * tech_support + 1.5125 * streaming_t_v + 1.6245 * streaming_movies + 0.0409 * monthly_charge−0.0014 * bandwidth_g_b_year + 1.0481 * techie + 0.3882 * gender_male−3.3051 * contract_one_year−3.3552 * contract_two_year−2.7511 * internet_service_fiber_optic−1.5583 * internet_service_none + 0.4389 * payment_method_electronic_check

In the context of logistic regression, both log-odds and probability are used to describe the likelihood of a particular outcome. Probability is a measure of the likelihood that a particular event will occur, expressed as a number between 0 and 1. Log odds of an event are the ratio of the probability that the event occurs to the probability that it does not occur. Log-odds provide a linear relationship between the predictors and the outcome, making the logistic regression model easier to fit.

To convert log odds to probability, one simply has to use the following equation: p=1/(1+e^-θ) where θ represents the log odds.

Therefore, the coefficients of the reduced model can be interpreted as follows:

- **Intercept (const)**: -4.6039
    - This is the baseline log-odds of churn when all independent variables are 0.
- **children**: 0.0618
    - Each additional child slightly increases the log-odds of churn.
- **multiple**: 0.4723
    - Having multiple devices increases the log-odds of churn.
- **online_security**: -0.1796
    - Having online security decreases the log-odds of churn.
- **tech_support**: -0.2381
    - Having tech support decreases the log-odds of churn.
- **streaming_t_v**: 1.5125
    - Having streaming TV increases the log-odds of churn significantly.
- **streaming_movies**: 1.6245
    - Having streaming movies increases the log-odds of churn significantly.
- **monthly_charge**: 0.0409
    - Higher monthly charges increase the log-odds of churn.
- **bandwidth_g_b_year**: -0.0014
    - Higher bandwidth usage per year slightly decreases the log-odds of churn.
- **techie**: 1.0481
    - Being tech-savvy increases the log-odds of churn.
- **gender_male**: 0.3882
    - Being male increases the log-odds of churn.
- **contract_one_year**: -3.3051
    - Having a one-year contract decreases the log-odds of churn significantly.
- **contract_two_year**: -3.3552

    - Having a two-year contract decreases the log-odds of churn significantly.

- **internet_service_fiber_optic**: -2.7511
    - Using fiber optic internet service decreases the log-odds of churn significantly.

- **internet_service_none**: -1.5583
    - Having no internet service decreases the log-odds of churn.

- **payment_method_electronic_check**: 0.4389
    - Paying via electronic check increases the log-odds of churn.

F2. Recommend a course of action based on your results.

My research question described in Part A1 is “Which customer factors contribute most to a customer’s decision to cancel their subscription with the service provider (i.e churn?)” A for-profit organization would aim to reduce customer churn as it’s typically cheaper and easier to retain an existing customer than it is to acquire a new customer.

As log-odds of churn increase, the probability of churn increases predictably, and vice versa. Therefore, in order for an organization to minimize churn, it should minimize the factors that increase the log-odds of churn.

The logistic regression model identifies the 16 customer factors that significantly contribute to the likelihood of churn. In order to minimize churn, an organization can implement the following plan:

- **Multiple Devices** (Coefficient: 0.4723):
    - **Recommendation**: Customers with multiple devices are more likely to churn. Investigate why these customers might be dissatisfied through anonymous polls and train customer service representatives to check in with customers. Perhaps they face technical difficulties or higher costs. Providing better support and possibly discounts for multiple devices could mitigate churn.

- **Streaming Services (TV and Movies)** (Coefficients: 1.5125 and 1.6245):
    - **Recommendation**: Customers using streaming services are significantly more likely to churn. Ensure that the streaming quality is high, and consider offering exclusive content or bundled packages to enhance perceived value. Monitoring streaming service satisfaction and addressing any issues proactively could help retain these customers.

- **Monthly Charge** (Coefficient: 0.0409):
    - **Recommendation**: Higher monthly charges are associated with higher churn. Review pricing strategies and consider offering loyalty discounts or personalized pricing plans. Regularly reassess pricing against competitors to ensure competitive rates without sacrificing profitability.

- **Tech-Savvy Customers (Techie)** (Coefficient: 1.0481):

    - **Recommendation**: Tech-savvy customers are more prone to churn, possibly because they are more aware of alternatives. Offering advanced features, early access to new technologies, or exclusive tech support could help retain these customers. Tech-savvy customers make up just 16.8% of the customer base (See Part C3 for a visualization,) but likely have an impact on the opinions of non tech-savvy customers as well. Therefore, if an organization aims to focus its marketing campaigns towards non tech-savvy customers, it may inadvertently lose out on those customers.

- **Gender (Male)** (Coefficient: 0.3882):
    - **Recommendation**: Males are slightly more likely to churn. As men make up a substantial portion of the general population and 47.4% of the telecom company's customer base (See Part C3 for a visualization,) it's unreasonable to focus on moving towards a female-only customer base. Therefore, rather than aiming to acquire a greater percentage of female customers, an organization should aim to identify what causes churn among male customers.

- **Payment Method - Electronic Check** (Coefficient: 0.4389):
    - **Recommendation**: Customers using electronic checks are more likely to churn. This is likely due to the fact that manual payments cause the customer to consciously evaluate their purchase. "Every recall is a reframe," as the heuristic goes. Therefore, encouraging customers to pay via debit or credit card automatically would likely reduce churn.

Web Sources

The 6 Assumptions of Logistic Regression (With Examples) – https://www.statology.org/assumptions-of-logistic-regression/

statsmodels.discrete.discrete_model.LogitResults.pvalues Documentation

https://www.statsmodels.org/dev/generated/statsmodels.discrete.discrete_model.LogitResults.pvalues.html

Works Consulted

Dunkler, D., Plischke, M., Leffondré, K., & Heinze, G. (2014). Augmented backward elimination: a pragmatic and purposeful way to develop statistical models. *PloS one*, *9*(11), e113677.