```
pip install numpy
```

```
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (1.25.2)
```

```
pip install pandas
```

```
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: numpy>=1.21.0 in
/usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1-
>pandas) (1.16.0)
```

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in
/usr/local/lib/python3.10/dist-packages (0.13.1)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (1.25.2)
Requirement already satisfied: pandas>=1.2 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (1.5.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (4.49.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (23.2)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (3.1.1)
```

```
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn)
(2023.4)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
```

```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.49.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.25.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
```

```python
import numpy as np
import pandas as pd
import datetime
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import colors
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt, numpy as np
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import AgglomerativeClustering
from matplotlib.colors import ListedColormap
from sklearn import metrics
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
np.random.seed(42)
```

```python
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving bigml_59c28831336c6604c800002a.csv to
bigml_59c28831336c6604c800002a.csv
```

```python
data = pd.read_csv(r"bigml_59c28831336c6604c800002a.csv")
print("Number of datapoints:", len(data))
data.head()
```

```
Number of datapoints: 3333
```

{"type":"dataframe","variable_name":"data"}

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   state                   3333 non-null   object
 1   account length          3333 non-null   int64
 2   area code               3333 non-null   int64
 3   phone number            3333 non-null   object
 4   international plan      3333 non-null   object
 5   voice mail plan         3333 non-null   object
 6   number vmail messages   3333 non-null   int64
 7   total day minutes       3333 non-null   float64
 8   total day calls         3333 non-null   int64
 9   total day charge        3333 non-null   float64
 10  total eve minutes       3333 non-null   float64
 11  total eve calls         3333 non-null   int64
 12  total eve charge        3333 non-null   float64
 13  total night minutes     3333 non-null   float64
 14  total night calls       3333 non-null   int64
 15  total night charge      3333 non-null   float64
 16  total intl minutes      3333 non-null   float64
 17  total intl calls        3333 non-null   int64
 18  total intl charge       3333 non-null   float64
 19  customer service calls  3333 non-null   int64
 20  churn                   3333 non-null   bool
dtypes: bool(1), float64(8), int64(8), object(4)
memory usage: 524.2+ KB
```

```
print("Total categories in the feature state :\n",
data["state"].value_counts(), "\n")
print("Total categories in the feature international plan:\n",
data["international plan"].value_counts(), "\n")
print("Total categories in the feature voice mail plan  :\n",
data["voice mail plan"].value_counts())
```

```
Total categories in the feature state :
 WV    106
MN     84
NY     83
AL     80
WI     78
OH     78
OR     78
WY     77
VA     77
CT     74
MI     73
ID     73
VT     73
TX     72
UT     72
IN     71
MD     70
KS     70
NC     68
NJ     68
MT     68
CO     66
NV     66
WA     66
RI     65
MA     65
MS     65
AZ     64
FL     63
MO     63
NM     62
ME     62
ND     62
NE     61
OK     61
DE     61
SC     60
```

```
SD     60
KY     59
IL     58
NH     56
AR     55
GA     54
DC     54
HI     53
TN     53
AK     52
LA     51
PA     45
IA     44
CA     34
Name: state, dtype: int64

Total categories in the feature international plan:
 no     3010
yes     323
Name: international plan, dtype: int64

Total categories in the feature voice mail plan  :
 no     2411
yes     922
Name: voice mail plan, dtype: int64
```

```
data.describe()
```

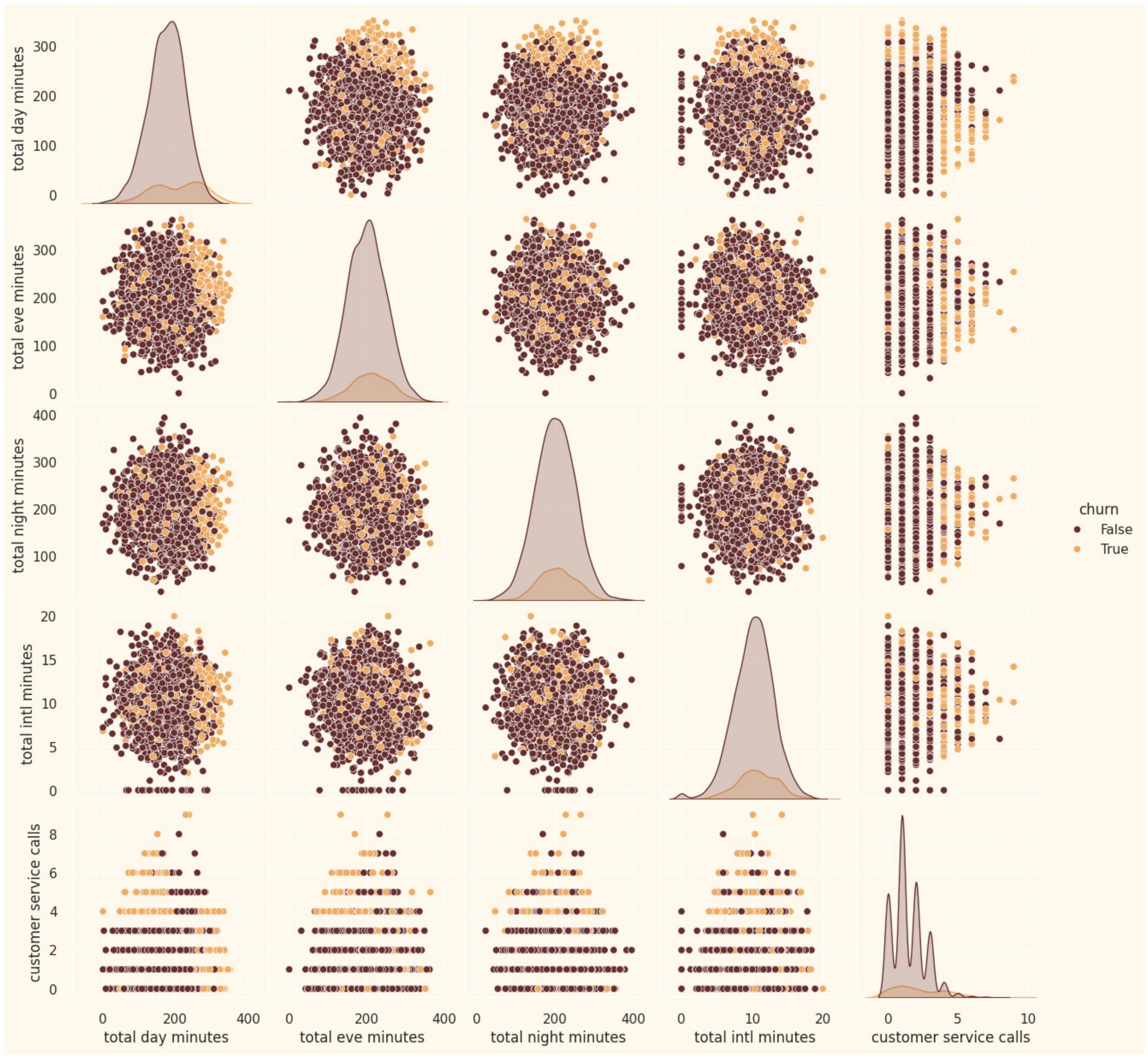
{"summary":"{\n \"name\": \"data\",\n \"rows\": 8,\n \"fields\": [\n   {\n   \"column\": \"account length\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1145.9000546457337,\n   \"min\": 1.0,\n   \"max\": 3333.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   101.06480648064806,\n   101.0,\n   3333.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"area code\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1050.9404187100654,\n   \"min\": 42.371290485606615,\n   \"max\": 3333.0,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   3333.0,\n   437.18241824182417,\n   510.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"number vmail messages\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1173.8301515451412,\n   \"min\": 0.0,\n   \"max\": 3333.0,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   3333.0,\n   8.099009900990099,\n   51.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"total day minutes\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\":

```
1126.5109095005125,\n        \"min\": 0.0,\n        \"max\": 3333.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
179.77509750975094,\n          179.4,\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total day calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1149.911513463978,\n        \"min\": 0.0,\n        \"max\": 3333.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
100.43564356435644,\n          101.0,\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total day charge\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1168.8739502927278,\n        \"min\": 0.0,\n        \"max\": 3333.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
30.562307230723075,\n          30.5,\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total eve minutes\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1122.3747331695974,\n        \"min\": 0.0,\n        \"max\": 3333.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
200.98034803480348,\n          201.4,\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total eve calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1149.7835542430264,\n        \"min\": 0.0,\n        \"max\": 3333.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
100.11431143114311,\n          100.0,\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total eve charge\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1173.1998112594604,\n        \"min\": 0.0,\n        \"max\": 3333.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
17.083540354035403,\n          17.12,\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total night minutes\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1119.9504817494353,\n        \"min\": 23.2,\n        \"max\": 3333.0,\
n        \"num_unique_values\": 8,\n        \"samples\": [\n
200.87203720372037,\n          201.2,\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total night calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1147.6980607245025,\n        \"min\": 19.568609346058558,\n
\"max\": 3333.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          100.10771077107711,\n          100.0,\n
3333.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total night charge\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1175.5113859334936,\n        \"min\":
```

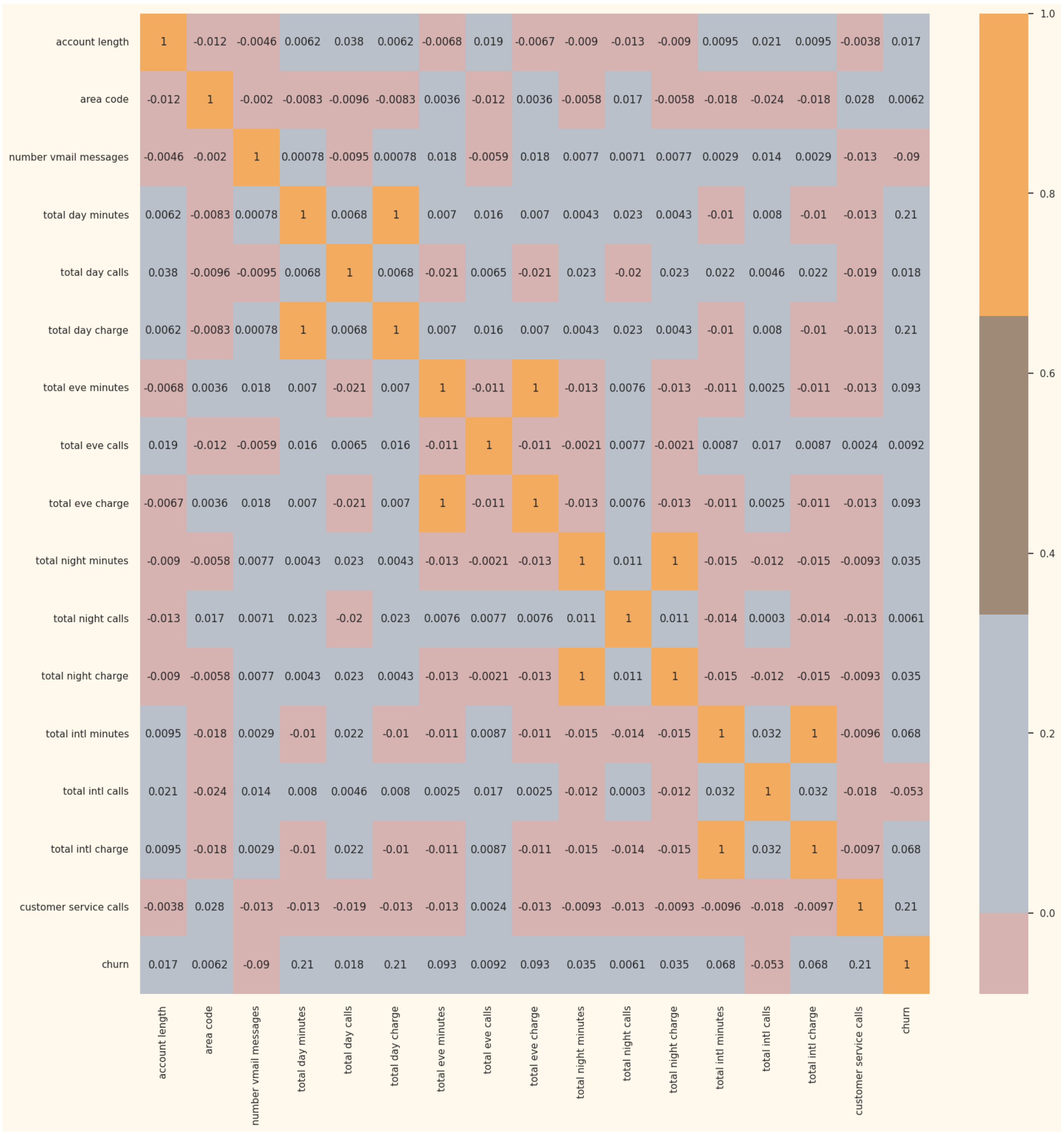
```
1.04,\n        \"max\": 3333.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          9.03932493249325,\n          9.05,\n
3333.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total intl minutes\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1175.1799500190264,\n        \"min\":
0.0,\n        \"max\": 3333.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          10.237293729372938,\n          10.3,\n
3333.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total intl calls\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1176.3918828456763,\n        \"min\":
0.0,\n        \"max\": 3333.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          4.4794479447944795,\n          4.0,\n
3333.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total intl charge\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1177.522388434277,\n        \"min\":
0.0,\n        \"max\": 3333.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          2.7645814581458144,\n          2.78,\n
3333.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"customer service calls\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1177.5948187412753,\n
\"min\": 0.0,\n        \"max\": 3333.0,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          3333.0,\n
1.5628562856285628,\n          2.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

```python
# Setting up colors preferences
sns.set(rc={"axes.facecolor":"#FFF9ED","figure.facecolor":"#FFF9ED"})
pallet = ["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78",
"#F3AB60"]
cmap = colors.ListedColormap(["#682F2F", "#9E726F", "#D6B2B1",
"#B9C0C9", "#9F8A78", "#F3AB60"])

# Selecting features to plot
To_Plot = ["total day minutes", "total eve minutes", "total night
minutes", "total intl minutes", "customer service calls", "churn"]

# Plotting pair plot
plt.figure()
sns.pairplot(data[To_Plot], hue="churn", palette=["#682F2F",
"#F3AB60"])
plt.show()
```

```
<Figure size 800x550 with 0 Axes>
```



```
#correlation matrix
corrmat= data.corr()
plt.figure(figsize=(20,20))
sns.heatmap(corrmat,annot=True, cmap=cmap, center=0)

<ipython-input-12-dd18bcecb909>:2: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  corrmat= data.corr()

<Axes: >
```



```
# DATA PREPROCESSING

#Get list of categorical variables
s = (data.dtypes == 'object')
object_cols = list(s[s].index)

print("Categorical variables in the dataset:", object_cols)

Categorical variables in the dataset: ['state', 'phone number',
'international plan', 'voice mail plan']
```

```python
#Label Encoding the object dtypes.
LE=LabelEncoder()
for i in object_cols:
    data[i]=data[[i]].apply(LE.fit_transform)

print("All features are now numerical")
```

```
All features are now numerical
```

```python
#Creating a copy of data
ds = data.copy()
# creating a subset of dataframe by dropping the features on deals
accepted and promotions
cols_del = ['phone number', 'state', 'area code' , 'account length']
ds = ds.drop(cols_del, axis=1)
#Scaling
scaler = StandardScaler()
scaler.fit(ds)
scaled_ds = pd.DataFrame(scaler.transform(ds),columns= ds.columns )
print("All features are now scaled")
```

```
All features are now scaled
```

```python
#Scaled data to be used for reducing the dimensionality
print("Dataframe to be used for further modelling:")
scaled_ds.head()
```

```
Dataframe to be used for further modelling:
```

```
{"summary":"{\n  \"name\": \"scaled_ds\",\n  \"rows\": 3333,\n
\"fields\": [\n    {\n      \"column\": \"international plan\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666932,\n        \"min\": -0.32758047881347 84,\n
\"max\": 3.0526849573639936,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          3.0526849573639936,\n          -
0.3275804788134784\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"voice mail plan\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001500487666932,\n        \"min\": -
0.6183962614101809,\n        \"max\": 1.6170861022342147,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          -
0.6183962614101809,\n          1.6170861022342147\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"number vmail messages\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666934,\n        \"min\": -0.5917598578997325,\n
\"max\": 3.134591032126578,\n        \"num_unique_values\": 46,\n
\"samples\": [\n          2.6231311060445357,\n
2.91539392094856\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total day minutes\",\n      \"properties\": {\n        \"dtype\":
```

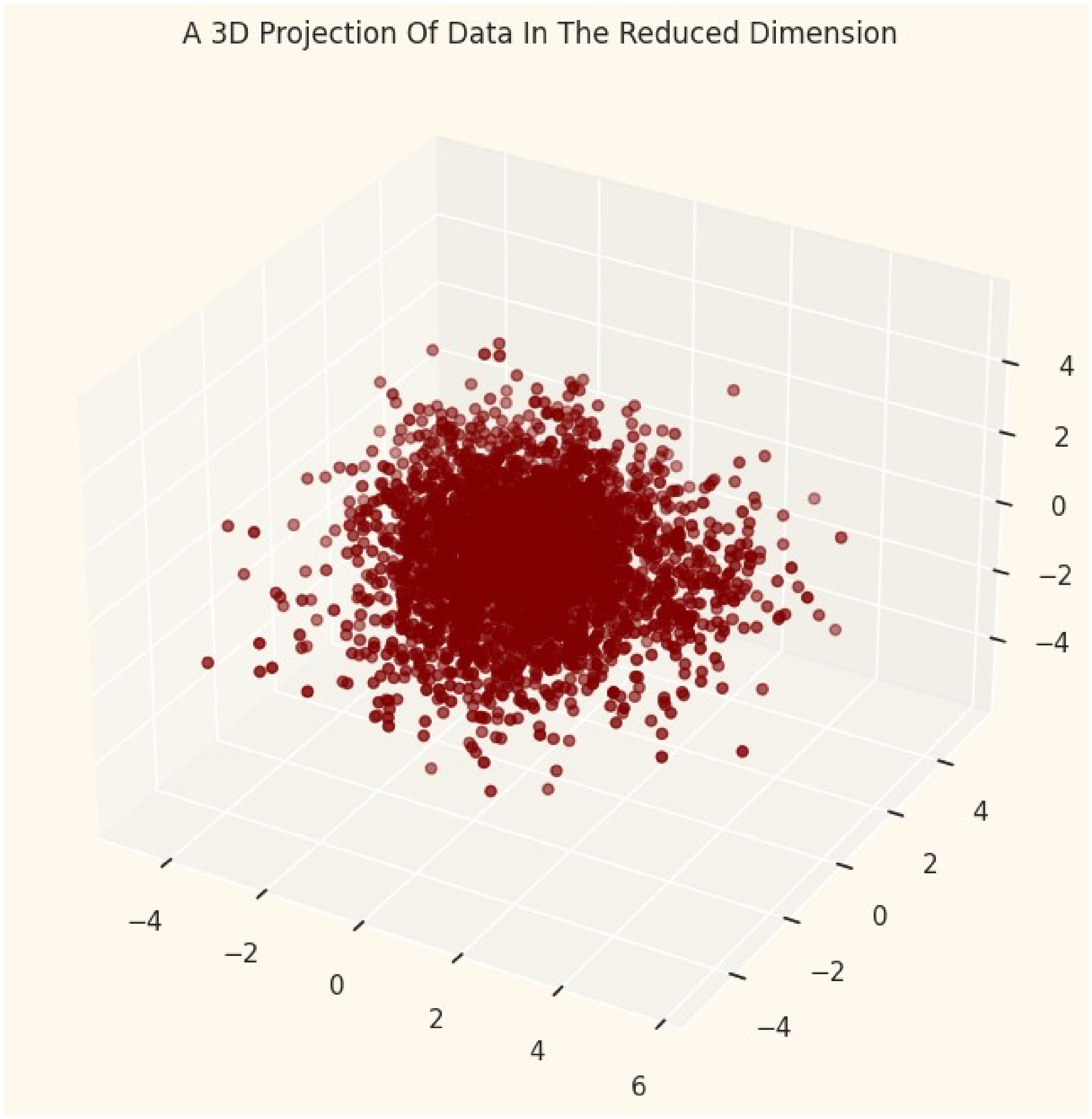
```
\"number\",\n        \"std\": 1.0001500487666934,\n        \"min\": -
3.301095851562981,\n        \"max\": 3.1404215819930377,\n
\"num_unique_values\": 1667,\n        \"samples\": [\n          -
1.692552729614557,\n          -1.1765702898114665\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total day calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666932,\n        \"min\": -5.00524651604494,\n
\"max\": 3.2175879861128793,\n        \"num_unique_values\": 119,\n
\"samples\": [\n          1.2241735613473474,\n          -
2.0649602395157802\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total day charge\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001500487666934,\n        \"min\": -
3.3011619542421338,\n        \"max\": 3.1408025697282262,\n
\"num_unique_values\": 1667,\n        \"samples\": [\n          -
1.6928311064567774,\n          -1.1765234199279657\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total eve minutes\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666932,\n        \"min\": -3.963621909638677,\n
\"max\": 3.2090658811395154,\n        \"num_unique_values\": 1611,\n
\"samples\": [\n          0.2784598714896957,\n
0.357345632180135\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total eve calls\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001500487666932,\n        \"min\": -
5.025910590736409,\n        \"max\": 3.5083817418122134,\n
\"num_unique_values\": 123,\n        \"samples\": [\n          -
0.35715066763628023,\n          2.1027335929218522\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total eve charge\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666934,\n        \"min\": -3.96367920999539,\n
\"max\": 3.207979699219934,\n        \"num_unique_values\": 1440,\n
\"samples\": [\n          1.8019540444206827,\n          -
0.6713518103912023\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total night minutes\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001500487666934,\n        \"min\": -
3.513648005179199,\n        \"max\": 3.839080926654489,\n
\"num_unique_values\": 1591,\n        \"samples\": [\n
2.17987178899756,\n          -0.4305645181623298\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total night calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666934,\n        \"min\": -3.4298696965828364,\n
\"max\": 3.827738875049051,\n        \"num_unique_values\": 120,\n
\"samples\": [\n          -2.049901869582407,\n          -
1.3854729158414596\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total night charge\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001500487666932,\n        \"min\": -
3.5153656605698598,\n        \"max\": 3.8367631750071522,\n
\"num_unique_values\": 933,\n        \"samples\": [\n          -
1.2477634614319408,\n          -0.9577213196817419\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total intl minutes\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666932,\n        \"min\": -3.6674134186931715,\n
\"max\": 3.4973969611645344,\n        \"num_unique_values\": 162,\n
\"samples\": [\n          2.279379196588725,\n
1.634546262401531\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total intl calls\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.0001500487666934,\n        \"min\": -
1.820288519390071,\n        \"max\": 6.307001011925544,\n
\"num_unique_values\": 21,\n        \"samples\": [\n          -
0.6011950896927287,\n          3.8688141525308595\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total intl charge\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0001500487666934,\n        \"min\": -3.668210059284882,\n
\"max\": 3.496829068707996,\n        \"num_unique_values\": 162,\n
\"samples\": [\n          2.276118698753654,\n
1.639226331820953\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"customer service calls\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1.0001500487666932,\n
\"min\": -1.1882184955849664,\n        \"max\": 5.654359775112334,\n
\"num_unique_values\": 10,\n        \"samples\": [\n
3.3735003515465674,\n          -1.1882184955849664\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"churn\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 1.0001500487666934,\
n        \"min\": -0.41167181614791937,\n        \"max\":
2.4291194120529402,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          2.4291194120529402,\n          -
0.41167181614791937\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"scaled_ds"}
```

```
#Initiating PCA to reduce dimentions aka features to 3
pca = PCA(n_components=3)
pca.fit(scaled_ds)
PCA_ds = pd.DataFrame(pca.transform(scaled_ds),
columns=(["col1","col2", "col3"]))
PCA_ds.describe().T
```

```
{"summary":"{\n  \"name\": \"PCA_ds\",\n  \"rows\": 3,\n  \"fields\":
[\n    {\n      \"column\": \"count\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.0,\n        \"min\":
3333.0,\n        \"max\": 3333.0,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          3333.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"mean\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 4.704563889191423e-17,\n
\"min\": -4.263682782838885e-17,\n        \"max\": 5.116419339406662e-
17,\n        \"num_unique_values\": 3,\n        \"samples\": [\n
-4.263682782838885e-17\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"std\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.01856466428662679,\n        \"min\": 1.425712048328782,\n
\"max\": 1.4597082764051543,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1.4597082764051543\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"min\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.33562905097668394,\n
\"min\": -5.383642231931511,\n        \"max\": -4.717949271829641,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          -
4.717949271829641\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"25%\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.02459964510661601,\n        \"min\": -0.9734825474689532,\n
\"max\": -0.9282501623202806,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          -0.9341047583586072\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"50%\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.03833646260304601,\n
\"min\": -0.03519786188139147,\n        \"max\":
0.03801210362651O245,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          -0.03519786188139147\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"75%\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.10251275002042369,\n
\"min\": 0.7997123881547982,\n        \"max\": 0.9921593385378198,\n
\"num_unique_values\": 3,\n        \"samples\": [\n
0.7997123881547982\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"max\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.1838552216480599,\n        \"min\": 5.274062588379545,\n
\"max\": 5.640679527802284,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          5.640679527802284\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

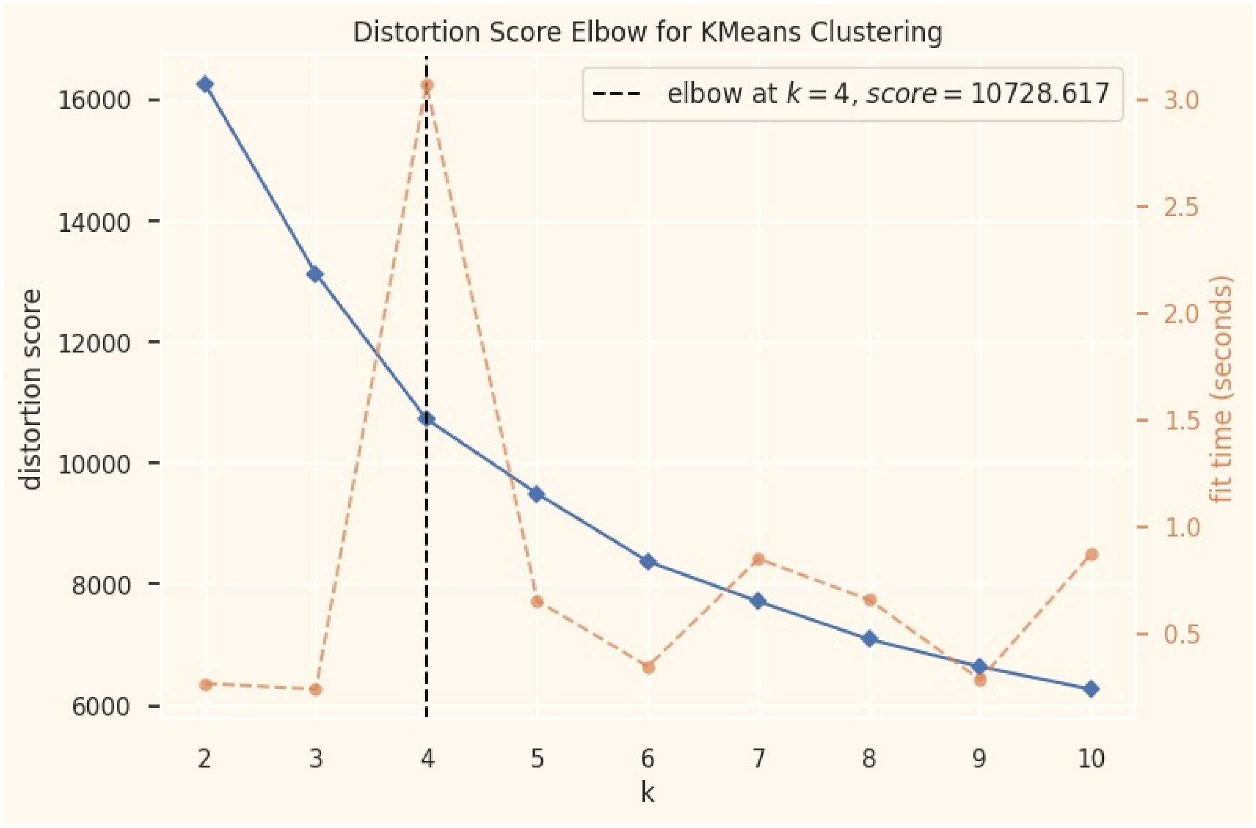
```
#A 3D Projection Of Data In The Reduced Dimension
x =PCA_ds["col1"]
y =PCA_ds["col2"]
```

```
z =PCA_ds["col3"]
#To plot
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection="3d")
ax.scatter(x,y,z, c="maroon", marker="o" )
ax.set_title("A 3D Projection Of Data In The Reduced Dimension")
plt.show()
```



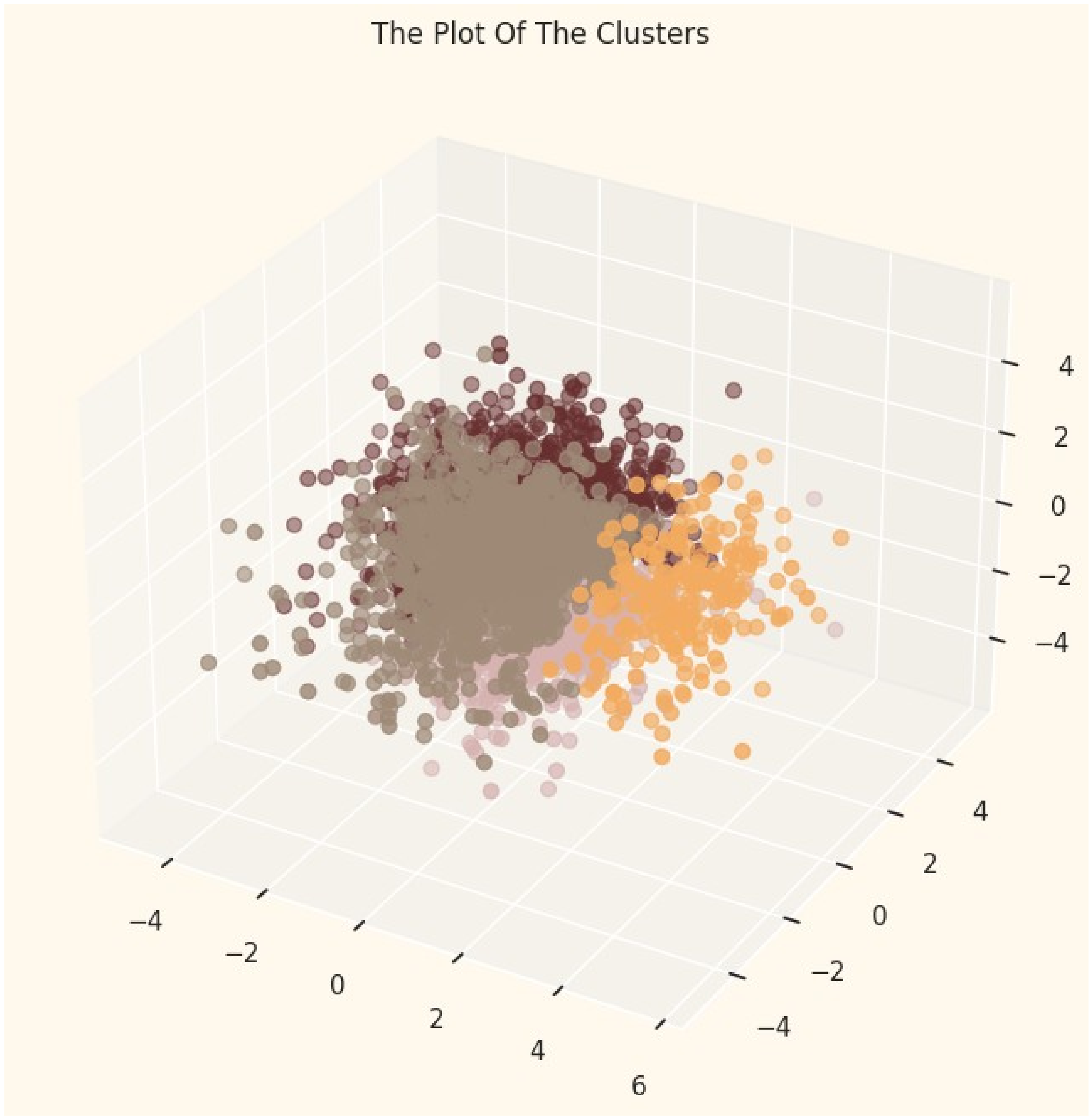
```
# Quick examination of elbow method to find numbers of clusters to
make.
print('Elbow Method to determine the number of clusters to be
formed:')
Elbow_M = KElbowVisualizer(KMeans(), k=10)
Elbow_M.fit(PCA_ds)
Elbow_M.show()

Elbow Method to determine the number of clusters to be formed:

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/
_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870
: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
```

```
warning
  warnings.warn(
```



```
<Axes: title={'center': 'Distortion Score Elbow for KMeans
Clustering'}, xlabel='k', ylabel='distortion score'>
```

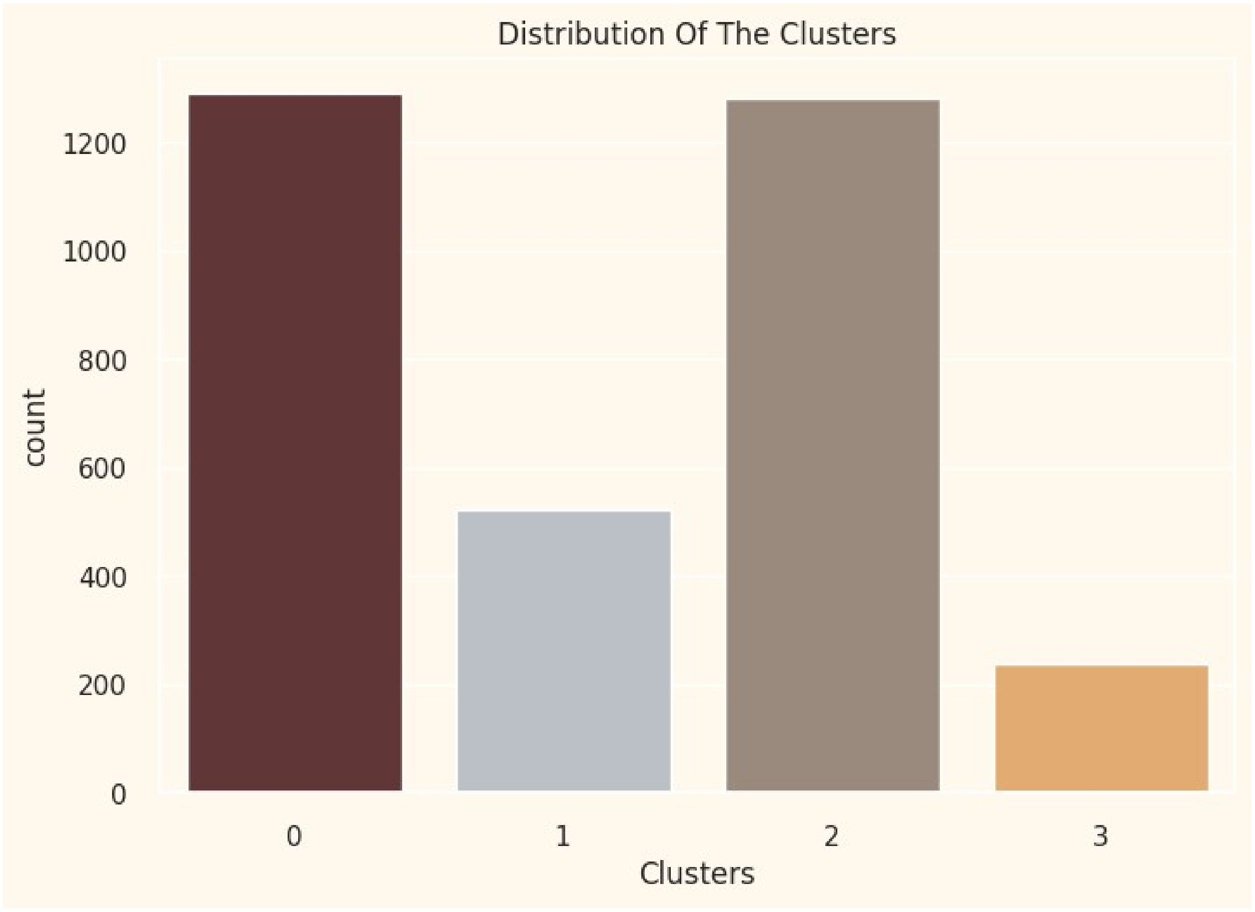
```
#Initiating the Agglomerative Clustering model
AC = AgglomerativeClustering(n_clusters=4)
# fit model and predict clusters
yhat_AC = AC.fit_predict(PCA_ds)
PCA_ds["Clusters"] = yhat_AC
#Adding the Clusters feature to the orignal dataframe.
data["Clusters"]= yhat_AC

#Plotting the clusters
fig = plt.figure(figsize=(10,8))
ax = plt.subplot(111, projection='3d', label="bla")
ax.scatter(x, y, z, s=40, c=PCA_ds["Clusters"], marker='o', cmap =
cmap )
ax.set_title("The Plot Of The Clusters")
plt.show()
```



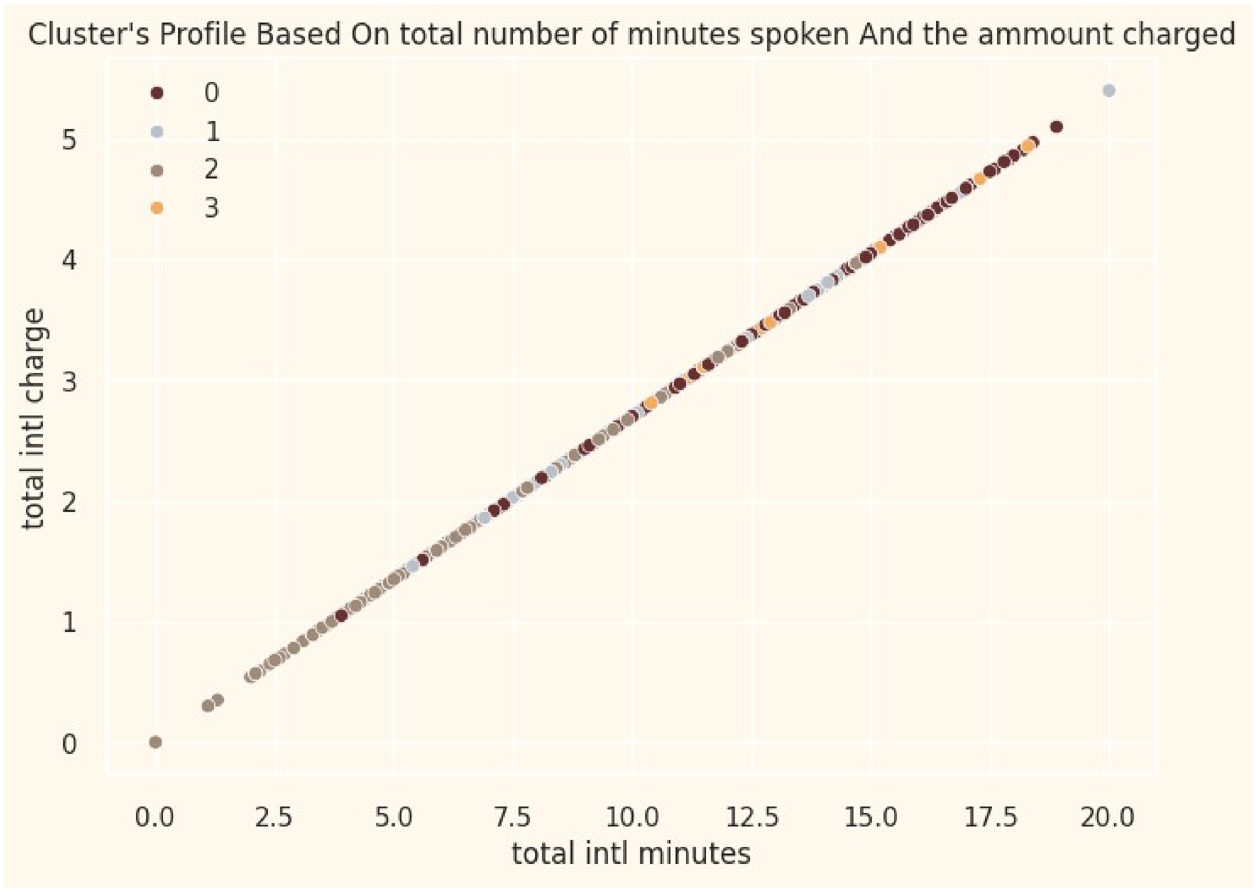
```
#Plotting countplot of clusters
pal = ["#682F2F","#B9C0C9", "#9F8A78","#F3AB60"]
pl = sns.countplot(x=data["Clusters"], palette= pal)
pl.set_title("Distribution Of The Clusters")
plt.show()
```

```
<ipython-input-24-2f63248a592e>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.
```

```
pl = sns.countplot(x=data["Clusters"], palette= pal)
```

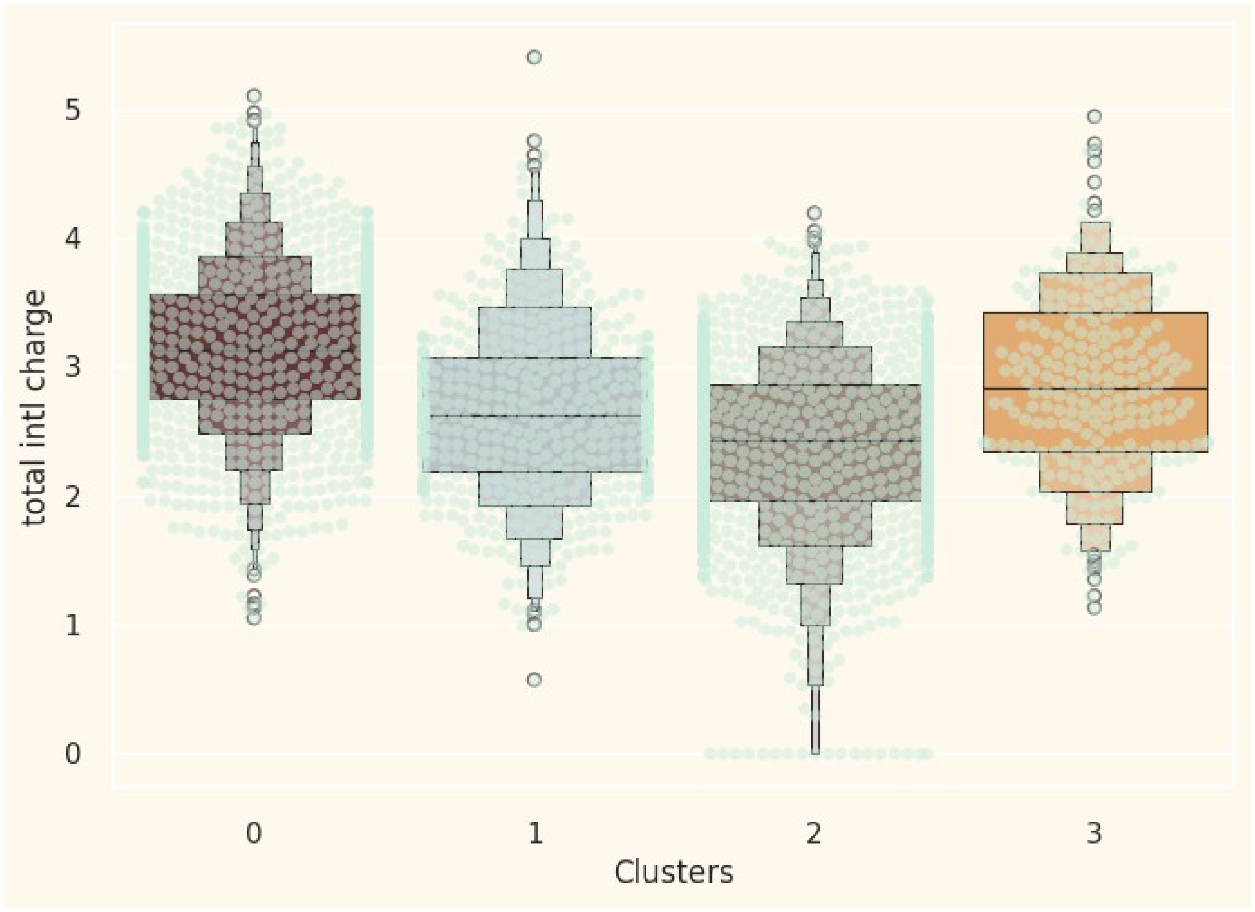


```
pl = sns.scatterplot(data = data,x=data["total intl minutes"],
y=data["total intl charge"],hue=data["Clusters"], palette= pal)
pl.set_title("Cluster's Profile Based On total number of minutes
spoken And the ammount charged")
plt.legend()
plt.show()
```

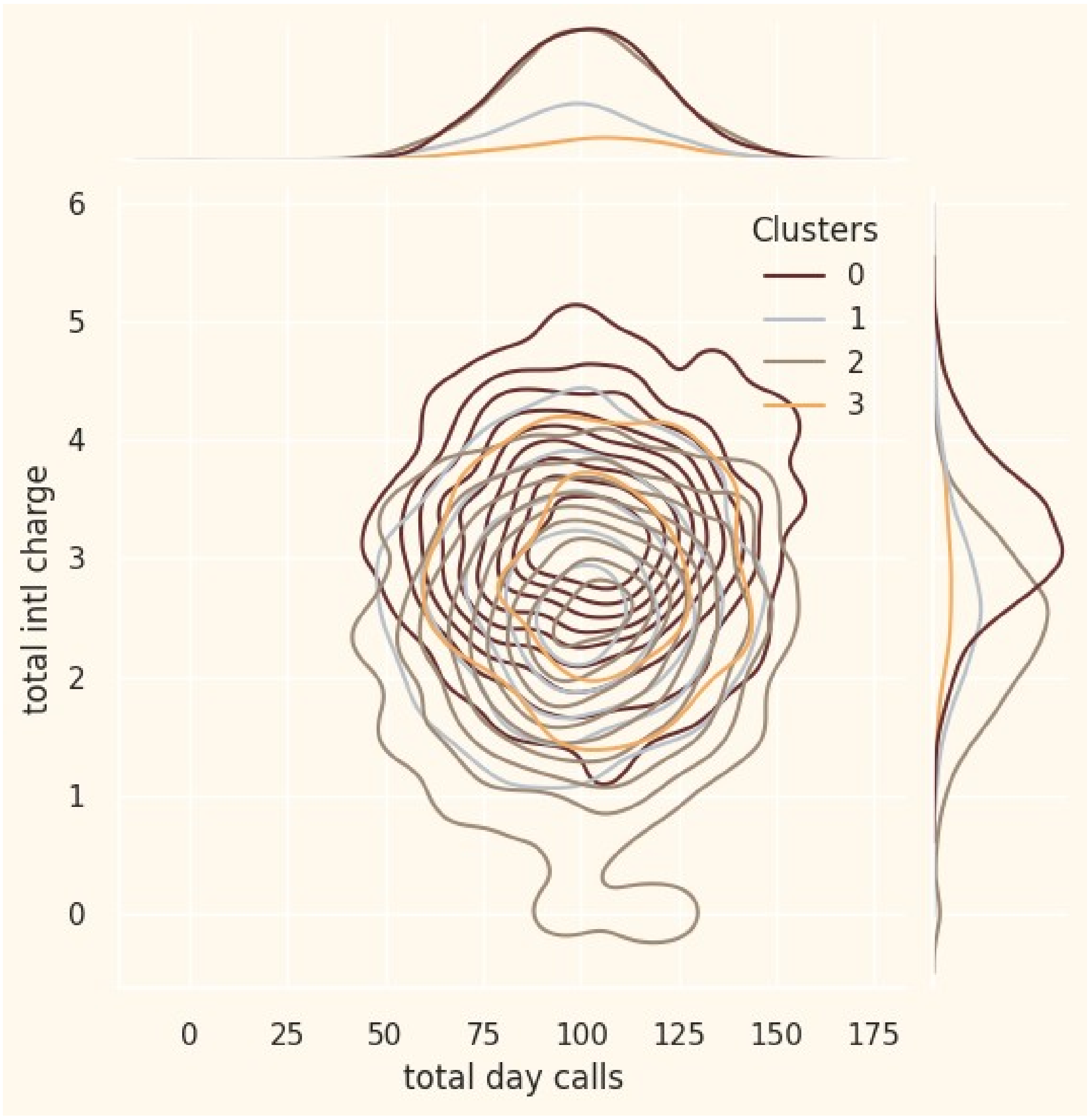


```
plt.figure()
pl=sns.swarmplot(x=data["Clusters"], y=data["total intl charge"],
color= "#CBEDDD", alpha=0.5 )
pl=sns.boxenplot(x=data["Clusters"], y=data["total intl charge"],
palette=pal)
plt.show()

/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 58.1% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 20.8% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 56.3% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
<ipython-input-33-df89a73409c0>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
```

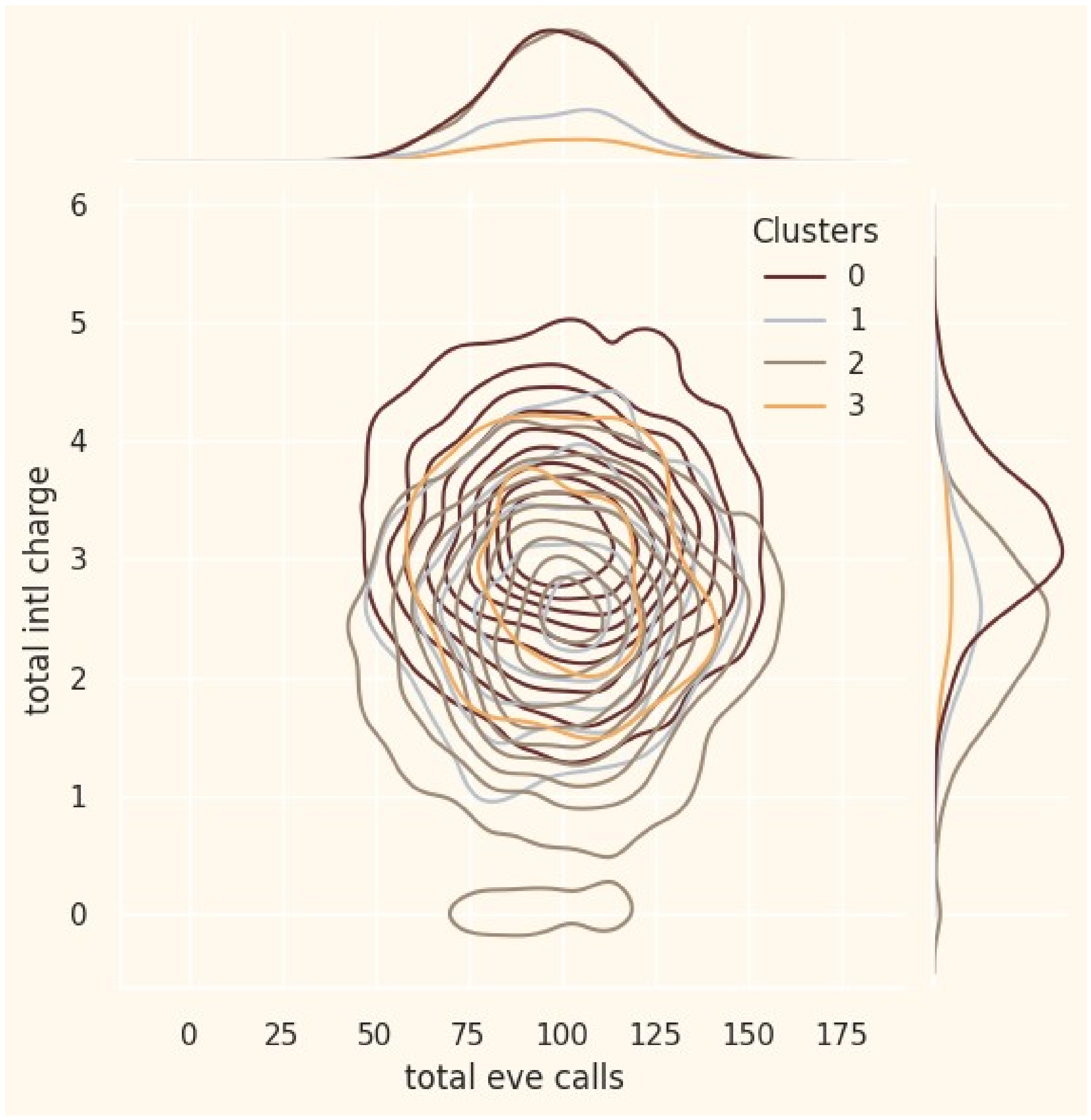
```
`legend=False` for the same effect.

  pl=sns.boxenplot(x=data["Clusters"], y=data["total intl charge"],
palette=pal)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 63.8% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 28.9% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 62.2% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



```
Personal = [ "total day calls","total eve calls","total night calls",
"total intl calls", "customer service calls", "number vmail messages",
"total day charge", "total eve charge","total night charge"]

for i in Personal:
    plt.figure()
    sns.jointplot(x=data[i], y=data["total intl charge"], hue
=data["Clusters"], kind="kde", palette=pal)
    plt.show()

<Figure size 800x550 with 0 Axes>
```

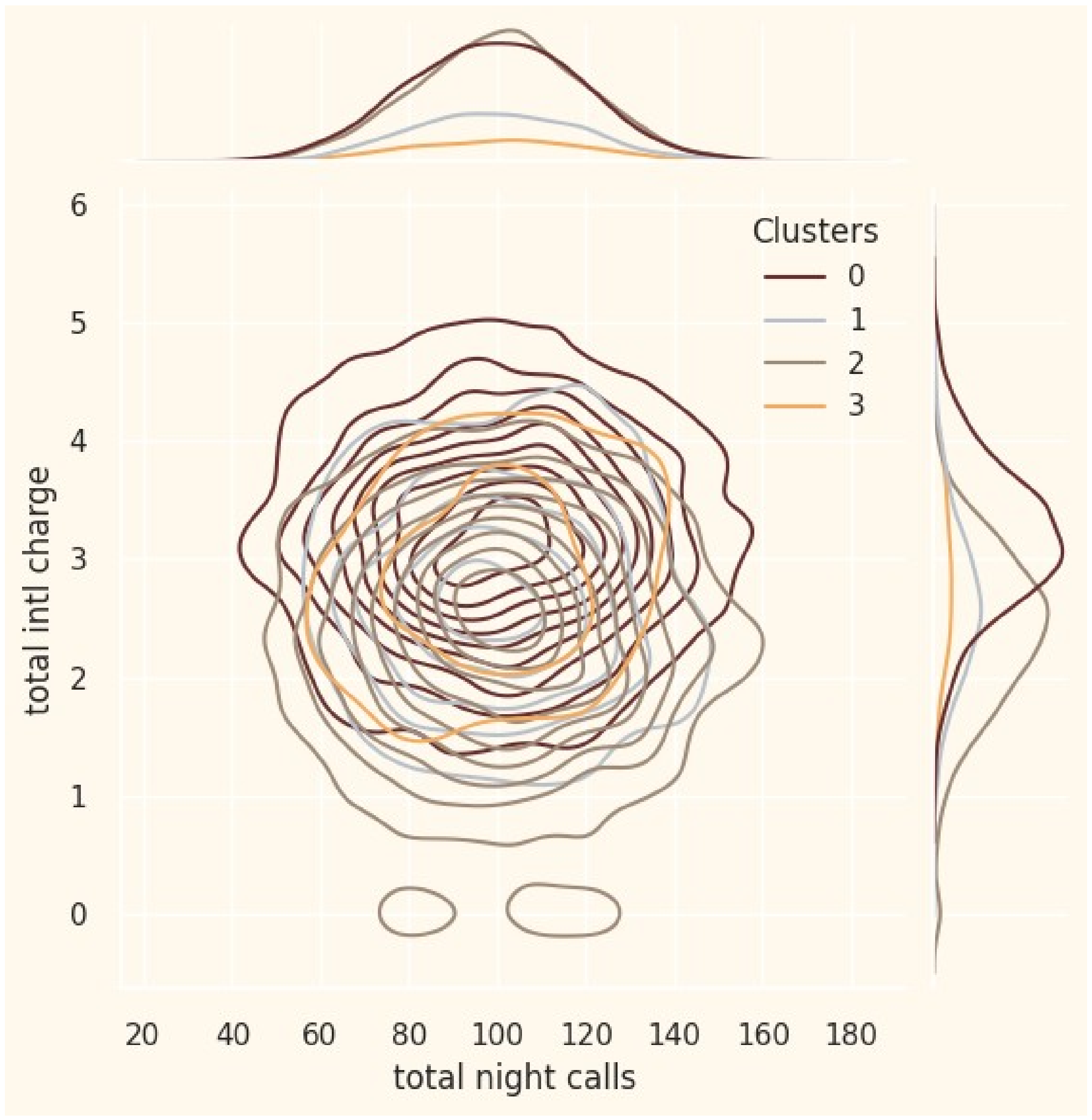


```
<Figure size 800x550 with 0 Axes>
```

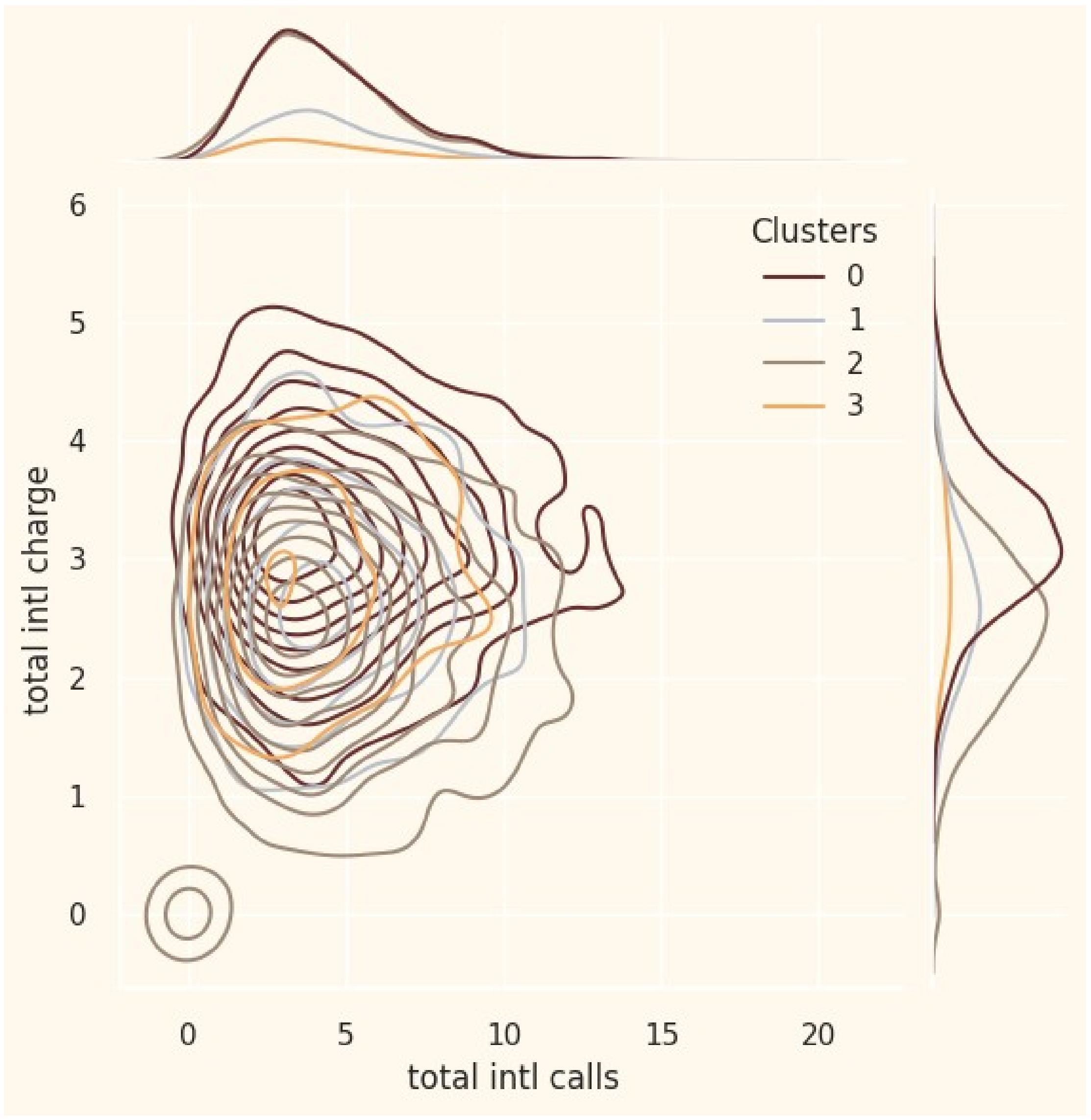


Clusters
0
1
2
3
total intl charge
total eve calls
0
1
2
3
4
5
6
0
25
50
75
100
125
150
175

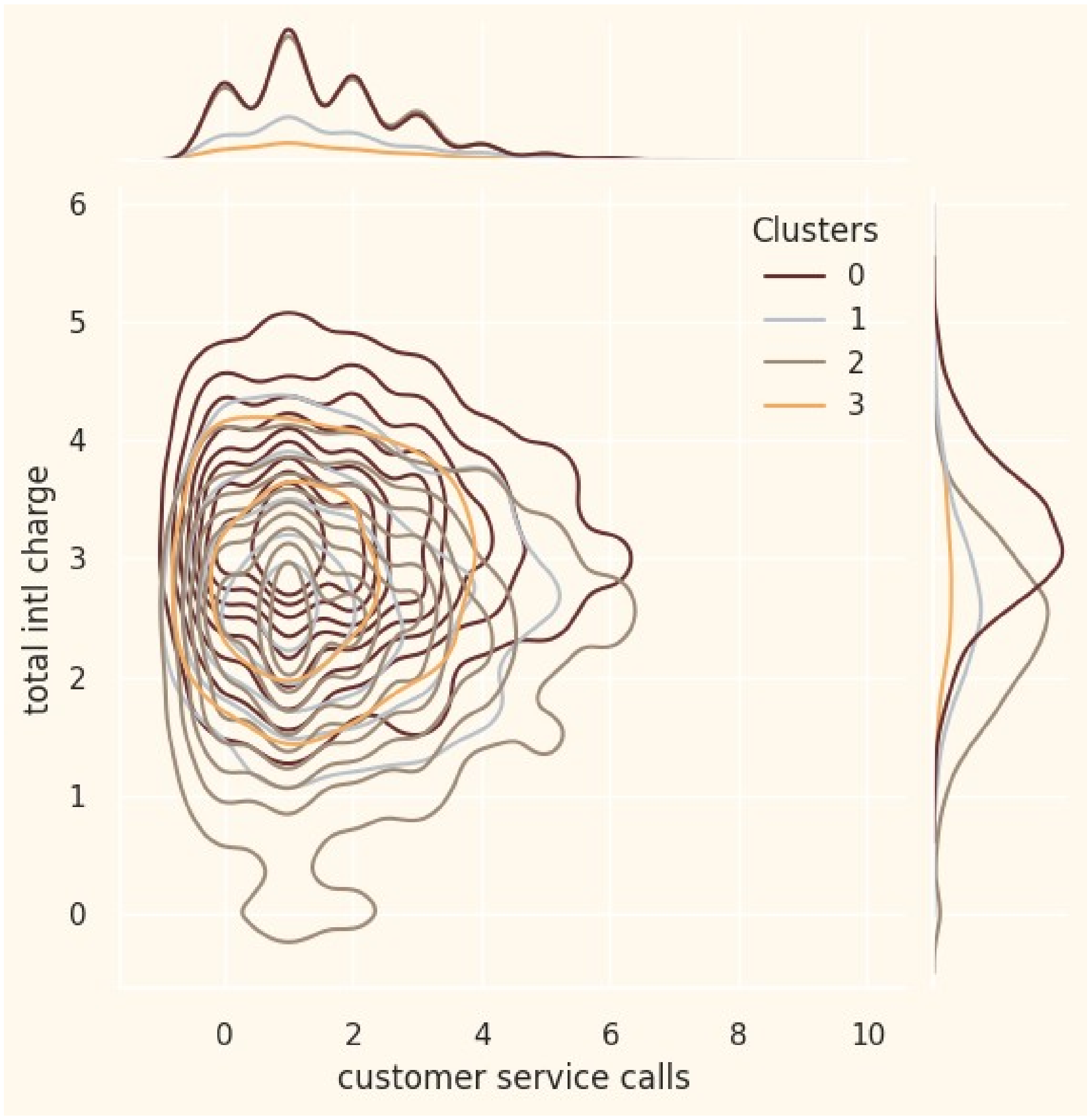
```
<Figure size 800x550 with 0 Axes>
```


Clusters
0
1
2
3
total intl charge
total night calls
6
5
4
3
2
1
0
20
40
60
80
100
120
140
160
180

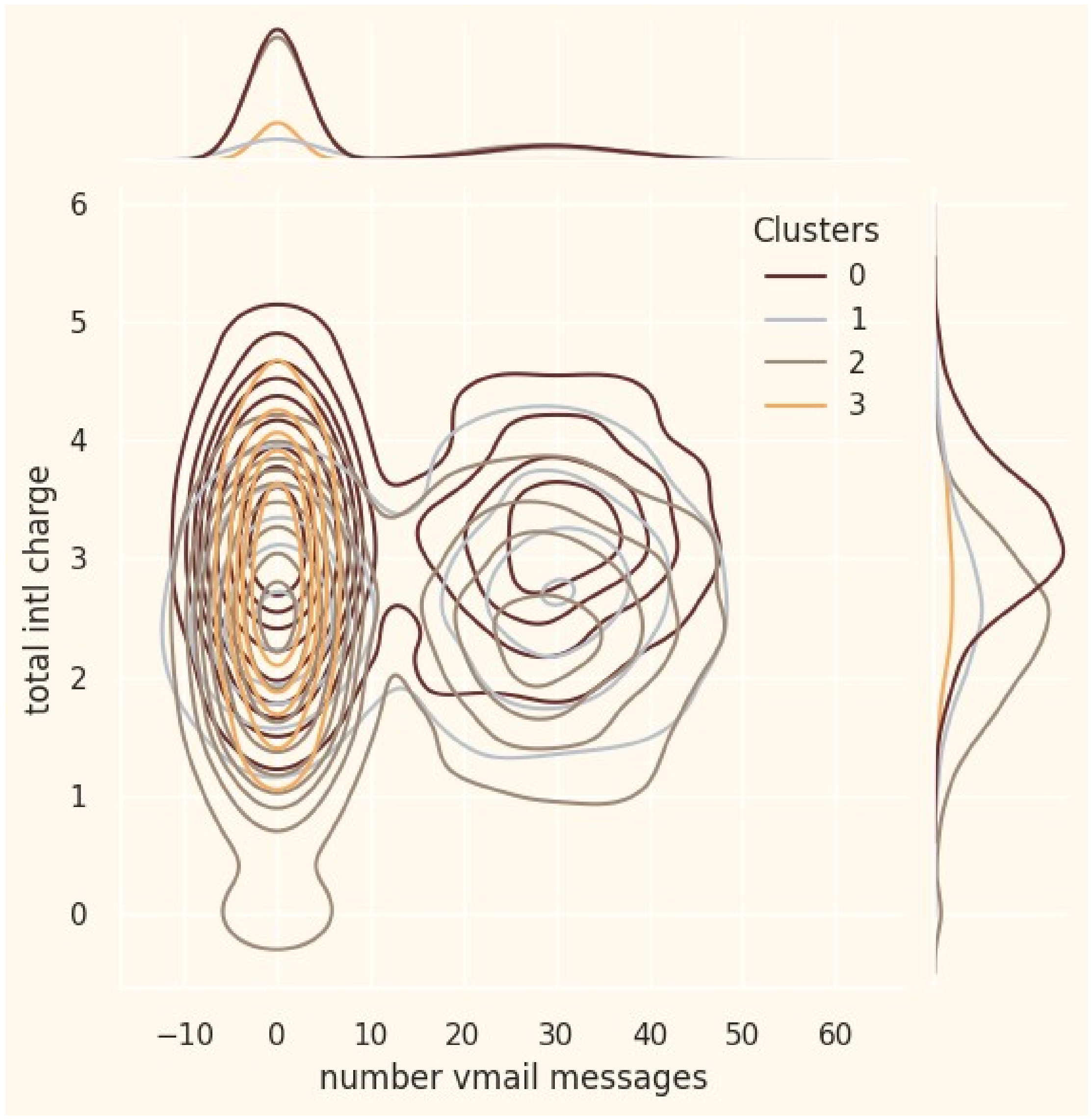
```
<Figure size 800x550 with 0 Axes>
```


Clusters
0
1
2
3
total intl charge
total intl calls
0
1
2
3
4
5
6
0
5
10
15
20

```
<Figure size 800x550 with 0 Axes>
```

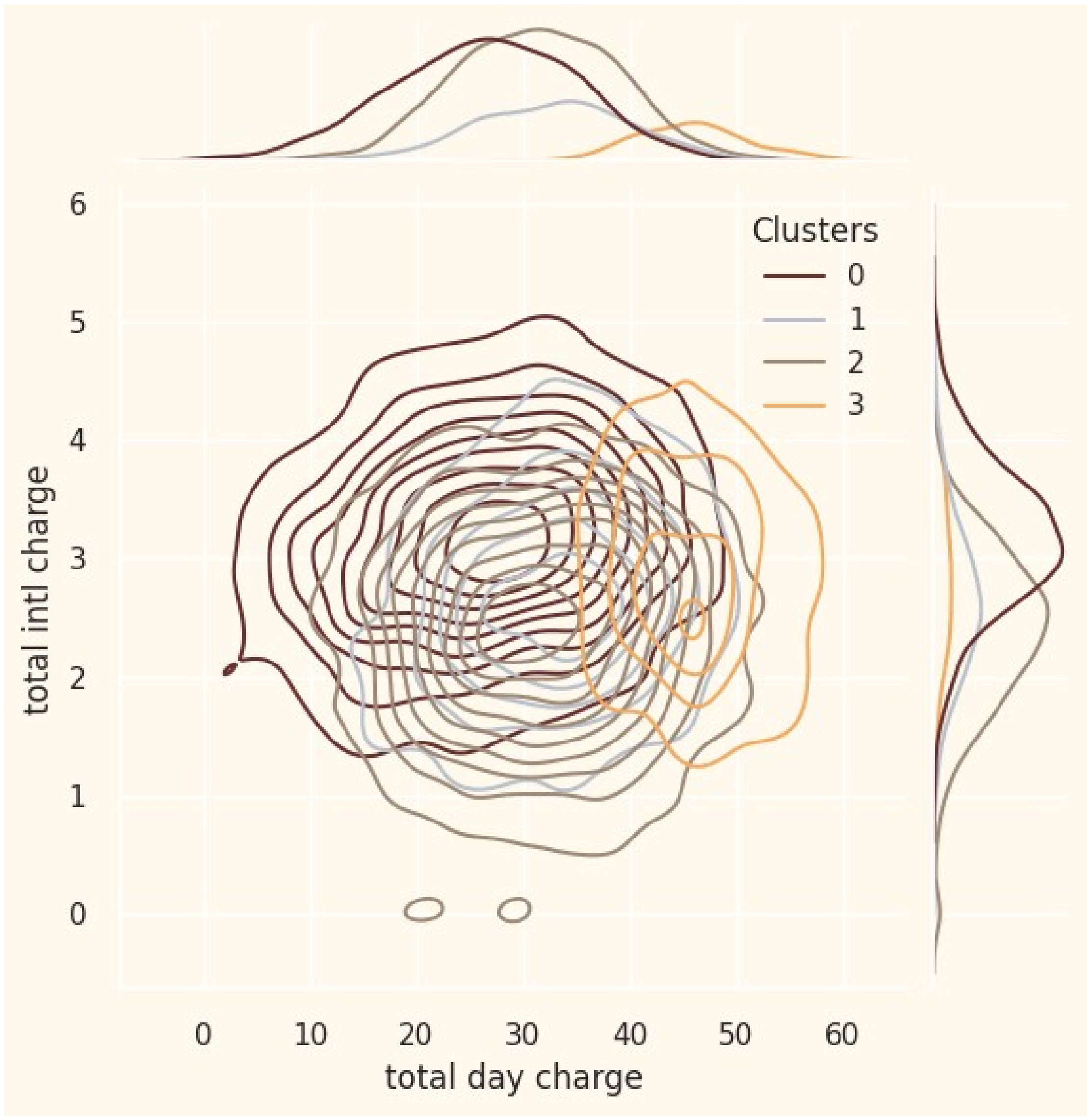

Clusters
0
1
2
3
total intl charge
customer service calls
0
1
2
3
4
5
6
0
2
4
6
8
10

```
<Figure size 800x550 with 0 Axes>
```



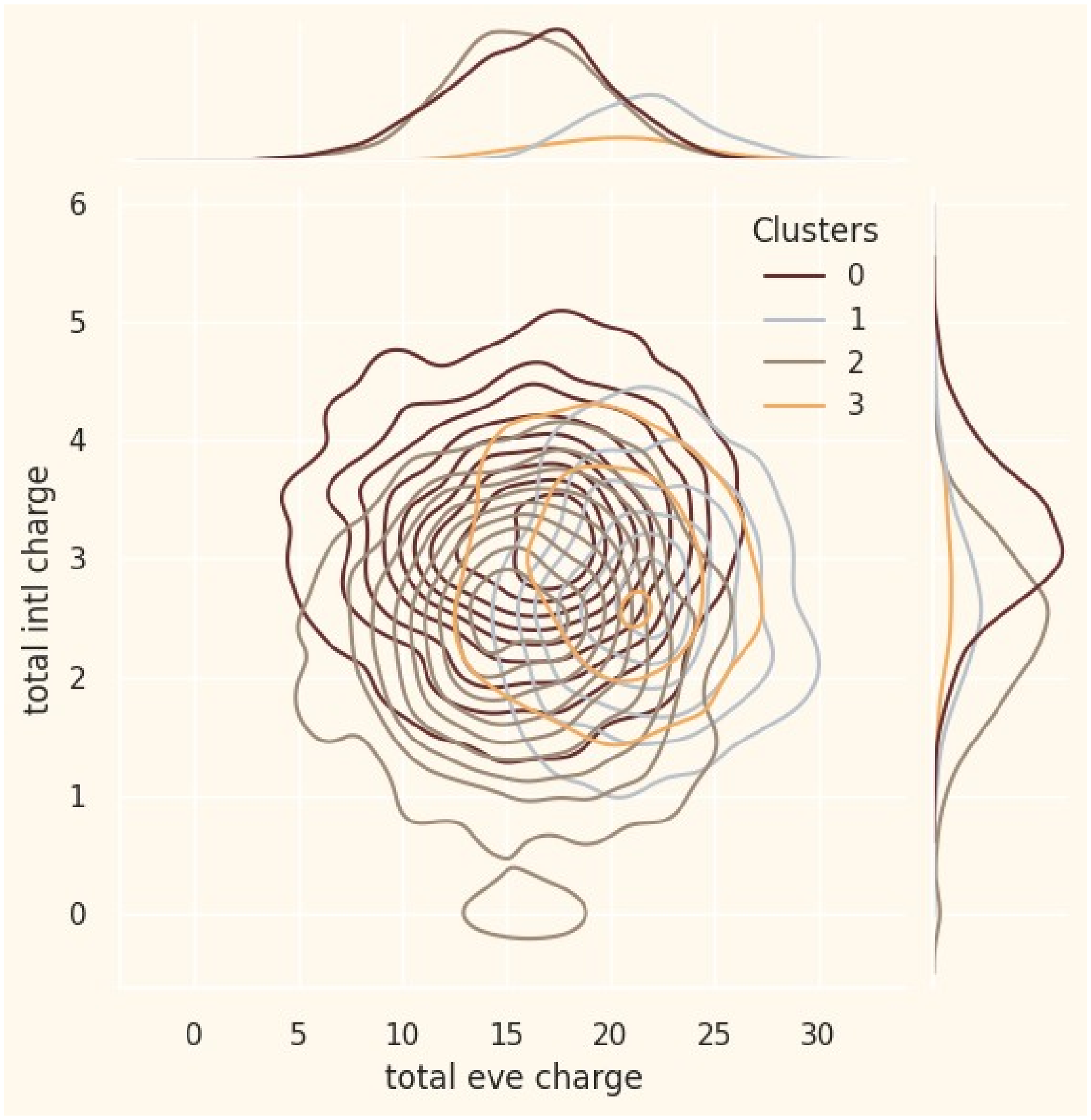
Clusters
0
1
2
3
total intl charge
number vmail messages
6
5
4
3
2
1
0
−10
0
10
20
30
40
50
60

```
<Figure size 800x550 with 0 Axes>
```

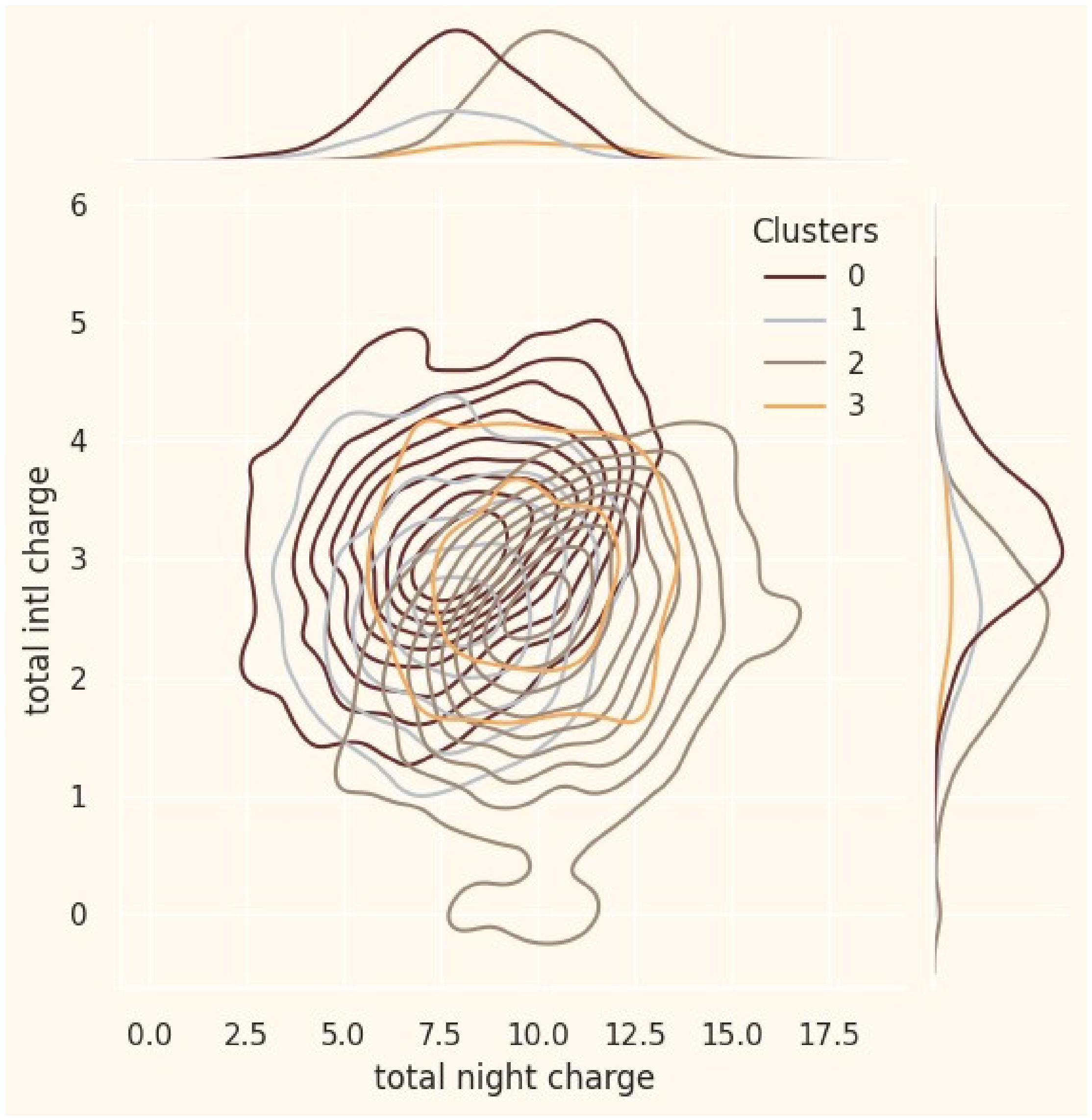


Clusters
0
1
2
3
total intl charge
total day charge
0
10
20
30
40
50
60
0
1
2
3
4
5
6

```
<Figure size 800x550 with 0 Axes>
```


Clusters
0
1
2
3
6
5
4
3
2
1
0
total intl charge
0
5
10
15
20
25
30
total eve charge

```
<Figure size 800x550 with 0 Axes>
```



Clusters
0
1
2
3
total intl charge
6
5
4
3
2
1
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
total night charge