```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 79,
   "id": "ae5bb75b",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Requirement already satisfied: matplotlib in c:\\users\\samsung\\anaconda3\\lib\\site-packages (3.8.4)\n",
      "Requirement already satisfied: contourpy>=1.0.1 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (1.2.0)\n",
      "Requirement already satisfied: cycler>=0.10 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (0.11.0)\n",
      "Requirement already satisfied: fonttools>=4.22.0 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (4.25.0)\n",
      "Requirement already satisfied: kiwisolver>=1.3.1 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (1.4.4)\n",
      "Requirement already satisfied: numpy>=1.21 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (1.26.4)\n",
      "Requirement already satisfied: packaging>=20.0 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (23.1)\n",
      "Requirement already satisfied: pillow>=8 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (10.2.0)\n",
      "Requirement already satisfied: pyparsing>=2.3.1 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (3.0.9)\n",
      "Requirement already satisfied: python-dateutil>=2.7 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from matplotlib) (2.8.2)\n",
      "Requirement already satisfied: six>=1.5 in c:\\users\\samsung\\anaconda3\\lib\\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)\n",
      "Note: you may need to restart the kernel to use updated packages.\n"
     ]
    },
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "\n",
      "[notice] A new release of pip is available: 24.0 -> 24.3.1\n",
      "[notice] To update, run: python.exe -m pip install --upgrade pip\n"
     ]
    }
   ],
   "source": [
    "pip install matplotlib"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 80,
   "id": "671f76f9",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import matplotlib.pyplot as plt\n",
    "import numpy as np\n",
    "import seaborn as sns\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 81,
   "id": "64765d7f",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>customerID</th>\n",
       "      <th>gender</th>\n",
       "      <th>SeniorCitizen</th>\n",
       "      <th>Partner</th>\n",
       "      <th>Dependents</th>\n",
       "      <th>tenure</th>\n",
       "      <th>PhoneService</th>\n",
       "      <th>MultipleLines</th>\n",
       "      <th>InternetService</th>\n",
       "      <th>OnlineSecurity</th>\n",
       "      <th>...</th>\n",
       "      <th>DeviceProtection</th>\n",
       "      <th>TechSupport</th>\n",
       "      <th>StreamingTV</th>\n",
       "      <th>StreamingMovies</th>\n",
       "      <th>Contract</th>\n",
       "      <th>PaperlessBilling</th>\n",
       "      <th>PaymentMethod</th>\n",
       "      <th>MonthlyCharges</th>\n",
       "      <th>TotalCharges</th>\n",
       "      <th>Churn</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>0</th>\n",
```

```
"      <td>7590-VHVEG</td>\n",
"      <td>Female</td>\n",
"      <td>0</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>1</td>\n",
"      <td>No</td>\n",
"      <td>No phone service</td>\n",
"      <td>DSL</td>\n",
"      <td>No</td>\n",
"      <td>...</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>Month-to-month</td>\n",
"      <td>Yes</td>\n",
"      <td>Electronic check</td>\n",
"      <td>29.85</td>\n",
"      <td>29.85</td>\n",
"      <td>No</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>5575-GNVDE</td>\n",
"      <td>Male</td>\n",
"      <td>0</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>34</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>DSL</td>\n",
"      <td>Yes</td>\n",
"      <td>...</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>One year</td>\n",
"      <td>No</td>\n",
"      <td>Mailed check</td>\n",
"      <td>56.95</td>\n",
"      <td>1889.5</td>\n",
"      <td>No</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>3668-QPYBK</td>\n",
"      <td>Male</td>\n",
"      <td>0</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>2</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>DSL</td>\n",
"      <td>Yes</td>\n",
"      <td>...</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>Month-to-month</td>\n",
"      <td>Yes</td>\n",
"      <td>Mailed check</td>\n",
"      <td>53.85</td>\n",
"      <td>108.15</td>\n",
"      <td>Yes</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>7795-CFOCW</td>\n",
"      <td>Male</td>\n",
"      <td>0</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>45</td>\n",
"      <td>No</td>\n",
"      <td>No phone service</td>\n",
"      <td>DSL</td>\n",
"      <td>Yes</td>\n",
"      <td>...</td>\n",
"      <td>Yes</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>One year</td>\n",
"      <td>No</td>\n",
"      <td>Bank transfer (automatic)</td>\n",
"      <td>42.30</td>\n",
"      <td>1840.75</td>\n",
"      <td>No</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>9237-HQITU</td>\n",
"      <td>Female</td>\n",
"      <td>0</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>2</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>Fiber optic</td>\n",
"      <td>No</td>\n",
```

```
      "      <td>...</td>\n",
      "      <td>No</td>\n",
      "      <td>No</td>\n",
      "      <td>No</td>\n",
      "      <td>No</td>\n",
      "      <td>Month-to-month</td>\n",
      "      <td>Yes</td>\n",
      "      <td>Electronic check</td>\n",
      "      <td>70.70</td>\n",
      "      <td>151.65</td>\n",
      "      <td>Yes</td>\n",
      "    </tr>\n",
      "  </tbody>\n",
      "</table>\n",
      "<p>5 rows × 21 columns</p>\n",
      "</div>"
     ],
     "text/plain": [
      "   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \\\n",
      "0  7590-VHVEG  Female              0     Yes         No       1           No   \n",
      "1  5575-GNVDE    Male              0      No         No      34          Yes   \n",
      "2  3668-QPYBK    Male              0      No         No       2          Yes   \n",
      "3  7795-CFOCW    Male              0      No         No      45           No   \n",
      "4  9237-HQITU  Female              0      No         No       2          Yes   \n",
      "\n",
      "      MultipleLines InternetService OnlineSecurity  ... DeviceProtection  \\\n",
      "0  No phone service             DSL             No  ...               No   \n",
      "1                No             DSL            Yes  ...              Yes   \n",
      "2                No             DSL            Yes  ...               No   \n",
      "3  No phone service             DSL            Yes  ...              Yes   \n",
      "4                No     Fiber optic             No  ...               No   \n",
      "\n",
      "  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \\\n",
      "0          No          No              No  Month-to-month              Yes   \n",
      "1          No          No              No        One year               No   \n",
      "2          No          No              No  Month-to-month              Yes   \n",
      "3         Yes          No              No        One year               No   \n",
      "4          No          No              No  Month-to-month              Yes   \n",
      "\n",
      "               PaymentMethod MonthlyCharges  TotalCharges Churn  \n",
      "0           Electronic check          29.85         29.85    No  \n",
      "1               Mailed check          56.95        1889.5    No  \n",
      "2               Mailed check          53.85        108.15   Yes  \n",
      "3  Bank transfer (automatic)          42.30       1840.75    No  \n",
      "4           Electronic check          70.70        151.65   Yes  \n",
      "\n",
      "[5 rows x 21 columns]"
     ]
    },
    "execution_count": 81,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "df = pd.read_csv('Customer Churn.csv')\n",
   "df.head()"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "faa777c2",
  "metadata": {},
  "source": [
   "##Replacing balnks with 0 as a tenure is 0 and no total charges are recorded"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 82,
  "id": "ab00aa75",
  "metadata": {},
  "outputs": [],
  "source": [
   "df[\"TotalCharges\"] = df[\"TotalCharges\"].replace(\" \",\"0\")\n",
   "df[\"TotalCharges\"] = df[\"TotalCharges\"].astype(\"float\")\n"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 83,
  "id": "ee85bb3b",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "<class 'pandas.core.frame.DataFrame'>\n",
     "RangeIndex: 7043 entries, 0 to 7042\n",
     "Data columns (total 21 columns):\n",
     " #   Column            Non-Null Count  Dtype  \n",
     "---  ------            --------------  -----  \n",
     " 0   customerID        7043 non-null   object \n",
     " 1   gender            7043 non-null   object \n",
     " 2   SeniorCitizen     7043 non-null   int64  \n",
     " 3   Partner           7043 non-null   object \n",
     " 4   Dependents        7043 non-null   object \n",
     " 5   tenure            7043 non-null   int64  \n",
     " 6   PhoneService      7043 non-null   object \n",
     " 7   MultipleLines     7043 non-null   object \n",
     " 8   InternetService   7043 non-null   object \n",
     " 9   OnlineSecurity    7043 non-null   object \n",
     " 10  OnlineBackup      7043 non-null   object \n",
     " 11  DeviceProtection  7043 non-null   object \n",
     " 12  TechSupport       7043 non-null   object \n",
```

```
    " 13  StreamingTV       7043 non-null   object \n",
    " 14  StreamingMovies   7043 non-null   object \n",
    " 15  Contract          7043 non-null   object \n",
    " 16  PaperlessBilling  7043 non-null   object \n",
    " 17  PaymentMethod     7043 non-null   object \n",
    " 18  MonthlyCharges    7043 non-null   float64\n",
    " 19  TotalCharges      7043 non-null   float64\n",
    " 20  Churn             7043 non-null   object \n",
    "dtypes: float64(2), int64(2), object(17)\n",
    "memory usage: 1.1+ MB\n"
   ]
  }
 ],
 "source": [
  "df.info()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 84,
 "id": "5ab9e8eb",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "<class 'pandas.core.frame.DataFrame'>\n",
    "RangeIndex: 7043 entries, 0 to 7042\n",
    "Data columns (total 21 columns):\n",
    " #   Column            Non-Null Count  Dtype  \n",
    "---  ------            --------------  -----  \n",
    " 0   customerID        7043 non-null   object \n",
    " 1   gender            7043 non-null   object \n",
    " 2   SeniorCitizen     7043 non-null   int64  \n",
    " 3   Partner           7043 non-null   object \n",
    " 4   Dependents        7043 non-null   object \n",
    " 5   tenure            7043 non-null   int64  \n",
    " 6   PhoneService      7043 non-null   object \n",
    " 7   MultipleLines     7043 non-null   object \n",
    " 8   InternetService   7043 non-null   object \n",
    " 9   OnlineSecurity    7043 non-null   object \n",
    " 10  OnlineBackup      7043 non-null   object \n",
    " 11  DeviceProtection  7043 non-null   object \n",
    " 12  TechSupport       7043 non-null   object \n",
    " 13  StreamingTV       7043 non-null   object \n",
    " 14  StreamingMovies   7043 non-null   object \n",
    " 15  Contract          7043 non-null   object \n",
    " 16  PaperlessBilling  7043 non-null   object \n",
    " 17  PaymentMethod     7043 non-null   object \n",
    " 18  MonthlyCharges    7043 non-null   float64\n",
    " 19  TotalCharges      7043 non-null   float64\n",
    " 20  Churn             7043 non-null   object \n",
    "dtypes: float64(2), int64(2), object(17)\n",
    "memory usage: 1.1+ MB\n"
   ]
  }
 ],
 "source": [
  "df.info()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 85,
 "id": "5a8e1195",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "0"
    ]
   },
   "execution_count": 85,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.isnull().sum().sum()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 86,
 "id": "417bce79",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
```

```
      "<table border=\"1\" class=\"dataframe\">\n",
      "  <thead>\n",
      "    <tr style=\"text-align: right;\">\n",
      "      <th></th>\n",
      "      <th>SeniorCitizen</th>\n",
      "      <th>tenure</th>\n",
      "      <th>MonthlyCharges</th>\n",
      "      <th>TotalCharges</th>\n",
      "    </tr>\n",
      "  </thead>\n",
      "  <tbody>\n",
      "    <tr>\n",
      "      <th>count</th>\n",
      "      <td>7043.000000</td>\n",
      "      <td>7043.000000</td>\n",
      "      <td>7043.000000</td>\n",
      "      <td>7043.000000</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>mean</th>\n",
      "      <td>0.162147</td>\n",
      "      <td>32.371149</td>\n",
      "      <td>64.761692</td>\n",
      "      <td>2279.734304</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>std</th>\n",
      "      <td>0.368612</td>\n",
      "      <td>24.559481</td>\n",
      "      <td>30.090047</td>\n",
      "      <td>2266.794470</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>min</th>\n",
      "      <td>0.000000</td>\n",
      "      <td>0.000000</td>\n",
      "      <td>18.250000</td>\n",
      "      <td>0.000000</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>25%</th>\n",
      "      <td>0.000000</td>\n",
      "      <td>9.000000</td>\n",
      "      <td>35.500000</td>\n",
      "      <td>398.550000</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>50%</th>\n",
      "      <td>0.000000</td>\n",
      "      <td>29.000000</td>\n",
      "      <td>70.350000</td>\n",
      "      <td>1394.550000</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>75%</th>\n",
      "      <td>0.000000</td>\n",
      "      <td>55.000000</td>\n",
      "      <td>89.850000</td>\n",
      "      <td>3786.600000</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>max</th>\n",
      "      <td>1.000000</td>\n",
      "      <td>72.000000</td>\n",
      "      <td>118.750000</td>\n",
      "      <td>8684.800000</td>\n",
      "    </tr>\n",
      "  </tbody>\n",
      "</table>\n",
      "</div>"
     ],
     "text/plain": [
      "       SeniorCitizen       tenure  MonthlyCharges  TotalCharges\n",
      "count    7043.000000  7043.000000     7043.000000   7043.000000\n",
      "mean        0.162147    32.371149       64.761692   2279.734304\n",
      "std         0.368612    24.559481       30.090047   2266.794470\n",
      "min         0.000000     0.000000       18.250000      0.000000\n",
      "25%         0.000000     9.000000       35.500000    398.550000\n",
      "50%         0.000000    29.000000       70.350000   1394.550000\n",
      "75%         0.000000    55.000000       89.850000   3786.600000\n",
      "max         1.000000    72.000000      118.750000   8684.800000"
     ]
    },
    "execution_count": 86,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "df.describe()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 87,
  "id": "d64b8a7f",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "0"
     ]
    },
    "execution_count": 87,
    "metadata": {},
```

```
    "output_type": "execute_result"
   }
  ],
  "source": [
   "df[\"customerID\"].duplicated().sum()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 88,
  "id": "52889332",
  "metadata": {},
  "outputs": [],
  "source": [
   "def conv(value):\n",
   "    if value == 1:\n",
   "        return \"yes\"\n",
   "    else:\n",
   "        return \"no\"\n",
   "    \n",
   "df['SeniorCitizen'] = df[\"SeniorCitizen\"].apply(conv)"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "7c3e5bd9",
  "metadata": {},
  "source": [
   "#Converted 0 and 1 value to yes and no value to make it easier to understand"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 89,
  "id": "79e7d474",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/html": [
      "<div>\n",
      "<style scoped>\n",
      "    .dataframe tbody tr th:only-of-type {\n",
      "        vertical-align: middle;\n",
      "    }\n",
      "\n",
      "    .dataframe tbody tr th {\n",
      "        vertical-align: top;\n",
      "    }\n",
      "\n",
      "    .dataframe thead th {\n",
      "        text-align: right;\n",
      "    }\n",
      "</style>\n",
      "<table border=\"1\" class=\"dataframe\">\n",
      "  <thead>\n",
      "    <tr style=\"text-align: right;\">\n",
      "      <th></th>\n",
      "      <th>customerID</th>\n",
      "      <th>gender</th>\n",
      "      <th>SeniorCitizen</th>\n",
      "      <th>Partner</th>\n",
      "      <th>Dependents</th>\n",
      "      <th>tenure</th>\n",
      "      <th>PhoneService</th>\n",
      "      <th>MultipleLines</th>\n",
      "      <th>InternetService</th>\n",
      "      <th>OnlineSecurity</th>\n",
      "      <th>...</th>\n",
      "      <th>DeviceProtection</th>\n",
      "      <th>TechSupport</th>\n",
      "      <th>StreamingTV</th>\n",
      "      <th>StreamingMovies</th>\n",
      "      <th>Contract</th>\n",
      "      <th>PaperlessBilling</th>\n",
      "      <th>PaymentMethod</th>\n",
      "      <th>MonthlyCharges</th>\n",
      "      <th>TotalCharges</th>\n",
      "      <th>Churn</th>\n",
      "    </tr>\n",
      "  </thead>\n",
      "  <tbody>\n",
      "    <tr>\n",
      "      <th>0</th>\n",
      "      <td>7590-VHVEG</td>\n",
      "      <td>Female</td>\n",
      "      <td>no</td>\n",
      "      <td>Yes</td>\n",
      "      <td>No</td>\n",
      "      <td>1</td>\n",
      "      <td>No</td>\n",
      "      <td>No phone service</td>\n",
      "      <td>DSL</td>\n",
      "      <td>No</td>\n",
      "      <td>...</td>\n",
      "      <td>No</td>\n",
      "      <td>No</td>\n",
      "      <td>No</td>\n",
      "      <td>No</td>\n",
      "      <td>Month-to-month</td>\n",
      "      <td>Yes</td>\n",
      "      <td>Electronic check</td>\n",
      "      <td>29.85</td>\n",
      "      <td>29.85</td>\n",
      "      <td>No</td>\n",
      "    </tr>\n",
```

```
"    <tr>\n",
"      <th>1</th>\n",
"      <td>5575-GNVDE</td>\n",
"      <td>Male</td>\n",
"      <td>no</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>34</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>DSL</td>\n",
"      <td>Yes</td>\n",
"      <td>...</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>One year</td>\n",
"      <td>No</td>\n",
"      <td>Mailed check</td>\n",
"      <td>56.95</td>\n",
"      <td>1889.50</td>\n",
"      <td>No</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>3668-QPYBK</td>\n",
"      <td>Male</td>\n",
"      <td>no</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>2</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>DSL</td>\n",
"      <td>Yes</td>\n",
"      <td>...</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>Month-to-month</td>\n",
"      <td>Yes</td>\n",
"      <td>Mailed check</td>\n",
"      <td>53.85</td>\n",
"      <td>108.15</td>\n",
"      <td>Yes</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>7795-CFOCW</td>\n",
"      <td>Male</td>\n",
"      <td>no</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>45</td>\n",
"      <td>No</td>\n",
"      <td>No phone service</td>\n",
"      <td>DSL</td>\n",
"      <td>Yes</td>\n",
"      <td>...</td>\n",
"      <td>Yes</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>One year</td>\n",
"      <td>No</td>\n",
"      <td>Bank transfer (automatic)</td>\n",
"      <td>42.30</td>\n",
"      <td>1840.75</td>\n",
"      <td>No</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>9237-HQITU</td>\n",
"      <td>Female</td>\n",
"      <td>no</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>2</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>Fiber optic</td>\n",
"      <td>No</td>\n",
"      <td>...</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>No</td>\n",
"      <td>Month-to-month</td>\n",
"      <td>Yes</td>\n",
"      <td>Electronic check</td>\n",
"      <td>70.70</td>\n",
"      <td>151.65</td>\n",
"      <td>Yes</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"<p>5 rows × 21 columns</p>\n",
"</div>"
],
"text/plain": [
"   customerID  gender SeniorCitizen Partner Dependents  tenure PhoneService  \\\n",
"0  7590-VHVEG  Female            no     Yes         No       1           No   \n",
"1  5575-GNVDE    Male            no      No         No      34          Yes   \n",
"2  3668-QPYBK    Male            no      No         No       2          Yes   \n",
```

```
      "3  7795-CFOCW    Male            no      No         No      45           No   \n",
      "4  9237-HQITU  Female            no      No         No       2          Yes   \n",
      "\n",
      "      MultipleLines InternetService OnlineSecurity  ... DeviceProtection  \\\n",
      "0  No phone service             DSL             No  ...               No   \n",
      "1                No             DSL            Yes  ...              Yes   \n",
      "2                No             DSL            Yes  ...               No   \n",
      "3  No phone service             DSL            Yes  ...              Yes   \n",
      "4                No     Fiber optic             No  ...               No   \n",
      "\n",
      "  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \\\n",
      "0          No          No              No  Month-to-month              Yes   \n",
      "1          No          No              No        One year               No   \n",
      "2          No          No              No  Month-to-month              Yes   \n",
      "3         Yes          No              No        One year               No   \n",
      "4          No          No              No  Month-to-month              Yes   \n",
      "\n",
      "              PaymentMethod MonthlyCharges  TotalCharges  Churn  \n",
      "0           Electronic check          29.85         29.85     No  \n",
      "1               Mailed check          56.95       1889.50     No  \n",
      "2               Mailed check          53.85        108.15    Yes  \n",
      "3  Bank transfer (automatic)          42.30       1840.75     No  \n",
      "4           Electronic check          70.70        151.65    Yes  \n",
      "\n",
      "[5 rows x 21 columns]"
     ]
    },
    "execution_count": 89,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "df.head()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 90,
  "id": "a818a134",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAAAkQAAAHICAYAAAACvevVkAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
     "text/plain": [
      "<Figure size 640x480 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   }
  ],
  "source": [
   "ax = sns.countplot(x = 'Churn', data = df)\n",
   "    \n",
   "ax.bar_label(ax.containers[0])\n",
   "plt.title(\"Count of customers churned\", fontsize = 14)\n",
   "plt.show()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 91,
  "id": "74b49910",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAAAVwAAAETCAYAAAB6GL6VAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
     "text/plain": [
      "<Figure size 300x500 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   }
  ],
  "source": [
   "plt.figure(figsize = (3,5))\n",
   "gb = df.groupby('Churn').agg({'Churn':\"count\"})\n",
   "plt.pie(gb['Churn'], labels = gb.index, autopct = \"%1.2f%%\")\n",
   "plt.title(\"Percentage of churned customers\", fontsize = 14)\n",
   "plt.show()"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "754cae16",
  "metadata": {},
  "source": [
   "#From the given pie chart we can conclude that 26.54% customers have churned out.\n",
   "#NOW LETS EXPLORE THE REASON BEHIND IT "
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 92,
  "id": "c35702f6",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAAAYoAAAGHCAYAAACu4BXOAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
     "text/plain": [
      "<Figure size 400x400 with 1 Axes>"
```

```
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "plt.figure(figsize = (4,4))\n",
  "sns.countplot(x = \"gender\", data = df, hue = \"Churn\")\n",
  "plt.title(\"Churn by Gender\")\n",
  "plt.show()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 93,
 "id": "6b32ea9f",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAAYoAAAGHCAYAAACu4BXBAAAACu4BXOAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
    "text/plain": [
     "<Figure size 400x400 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "plt.figure(figsize = (4,4))\n",
  "sns.countplot(x = \"SeniorCitizen\", data = df, hue = \"Churn\")\n",
  "plt.title(\"Churn by SeniorCitizen\")\n",
  "plt.show()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 94,
 "id": "3526cc8d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAAYoAAAGHCAYAAACu4BXBAAAACu4BXOAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
    "text/plain": [
     "<Figure size 400x400 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "plt.figure(figsize = (4,4))\n",
  "sns.countplot(x = \"SeniorCitizen\", data = df)\n",
  "plt.title(\"Count of Customers by SeniorCitizen\")\n",
  "ax.bar_label(ax.containers[0])\n",
  "plt.show()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 95,
 "id": "373be62c",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAAYoAAAGHCAYAAACu4BXBAAAACu4BXOAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
    "text/plain": [
     "<Figure size 600x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "data = df.groupby(['SeniorCitizen', 'Churn']).size().reset_index(name='Count')\n",
  "total_counts = data.groupby('SeniorCitizen')['Count'].transform('sum')\n",
  "data['Percentage'] = (data['Count'] / total_counts) * 100\n",
  "\n",
  "# Pivot the data for stacked bar chart\n",
  "pivot_data = data.pivot(index='SeniorCitizen', columns='Churn', values='Percentage').fillna(0)\n",
  "\n",
  "# Plot the stacked bar chart\n",
  "ax = pivot_data.plot(kind='bar', stacked=True, figsize=(6, 6), color=['blue', 'darkorange'])\n",
  "\n",
  "# Add percentage labels\n",
  "for p in ax.patches:\n",
  "    width, height = p.get_width(), p.get_height()\n",
  "    x, y = p.get_xy()\n",
  "    if height > 0:  # Only annotate non-zero bars\n",
  "        ax.text(x + width / 2, y + height / 2, f'{height:.1f}%', \n",
  "                ha='center', va='center', fontsize=10, color='black')\n",
  "\n",
  "# Add labels and title\n",
  "plt.title('Churn by SeniorCitizen (as % of Total)', fontsize=14)\n",
  "plt.xlabel('SeniorCitizen', fontsize=12)\n",
  "plt.ylabel('Percentage', fontsize=12)\n",
  "plt.legend(title='Churn', bbox_to_anchor=(1.05, 1), loc='upper left')\n",
  "plt.tight_layout()\n",
  "plt.show()"
 ]
```

```
},
{
 "cell_type": "markdown",
 "id": "252a7e7e",
 "metadata": {},
 "source": [
  "#Compartivel of a larger number have been churned out in Senior citizen Section"
 ]
},
{
 "cell_type": "code",
 "execution_count": 96,
 "id": "d3af5745",
 "metadata": {},
 "outputs": [
  {
   "name": "stderr",
   "output_type": "stream",
   "text": [
    "C:\\Users\\SAMSUNG\\anaconda3\\Lib\\site-packages\\seaborn\\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a
    "  with pd.option_context('mode.use_inf_as_na', True):\n"
   ]
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAAwUAAAINCAYAAABvQdOgAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
    "text/plain": [
     "<Figure size 900x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "plt.figure(figsize = (9,6))\n",
  "sns.histplot(x = \"tenure\", data = df, bins = 40, hue = \"Churn\")\n",
  "plt.show()"
 ]
},
{
 "cell_type": "markdown",
 "id": "9b4c7214",
 "metadata": {},
 "source": [
  "#People who have used our services for a long time have stayed and people who have used for 1 or 2 months have churned"
 ]
},
{
 "cell_type": "code",
 "execution_count": 97,
 "id": "980e13af",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAAwUAAAYoAAAGHCAYAAAACu4BXOAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjguNCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8fJSN1AAA
    "text/plain": [
     "<Figure size 400x400 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "plt.figure(figsize = (4,4))\n",
  "sns.countplot(x = \"Contract\", data = df, hue = \"Churn\")\n",
  "plt.title(\"Count of Customers by Contract\")\n",
  "ax.bar_label(ax.containers[0])\n",
  "plt.show()"
 ]
},
{
 "cell_type": "markdown",
 "id": "faa4e104",
 "metadata": {},
 "source": [
  "#People who have month to month contract are likely to churn as compared to people who are having yearly contract "
 ]
},
{
 "cell_type": "code",
 "execution_count": 98,
 "id": "8136f28a",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',\n",
     "       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',\n",
     "       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',\n",
     "       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',\n",
     "       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',\n",
     "       'TotalCharges', 'Churn'], dtype=object)"
    ]
   },
   "execution_count": 98,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.columns.values"
 ]
```

```
},
{
 "cell_type": "code",
 "execution_count": 99,
 "id": "daca2f73",
 "metadata": {
  "scrolled": false
 },
 "outputs": [
  {
   "data": {
    "image/png":
    "text/plain": [
     "<Figure size 1500x1200 with 9 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "# Example DataFrame\n",
  "data = {\n",
  "    'PhoneService': ['Yes', 'No', 'Yes', 'Yes', 'No'],\n",
  "    'MultipleLines': ['No', 'Yes', 'No', 'Yes', 'No'],\n",
  "    'InternetService': ['DSL', 'Fiber optic', 'DSL', 'No', 'Fiber optic'],\n",
  "    'OnlineSecurity': ['Yes', 'No', 'No', 'Yes', 'No'],\n",
  "    'OnlineBackup': ['No', 'Yes', 'Yes', 'No', 'Yes'],\n",
  "    'DeviceProtection': ['Yes', 'No', 'No', 'Yes', 'No'],\n",
  "    'TechSupport': ['No', 'Yes', 'No', 'No', 'Yes'],\n",
  "    'StreamingTV': ['Yes', 'No', 'Yes', 'No', 'Yes'],\n",
  "    'StreamingMovies': ['No', 'Yes', 'No', 'Yes', 'No'],\n",
  "}\n",
  "\n",
  "df = pd.DataFrame(data)\n",
  "\n",
  "# List of columns\n",
  "columns = [\n",
  "    'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',\n",
  "    'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies'\n",
  "]\n",
  "\n",
  "# Create subplots\n",
  "n_cols = 3  # Number of columns per row\n",
  "n_rows = -(-len(columns) // n_cols)  # Calculate number of rows using ceiling division\n",
  "\n",
  "fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 4))\n",
  "axes = axes.flatten()\n",
  "\n",
  "for i, col in enumerate(columns):\n",
  "    sns.countplot(data=df, x=col, ax=axes[i])\n",
  "    axes[i].set_title(col)\n",
  "    axes[i].set_xlabel('')\n",
  "    axes[i].set_ylabel('Count')\n",
  "\n",
  "# Remove empty subplots if the number of columns is not a multiple of n_cols\n",
  "for j in range(i + 1, len(axes)):\n",
  "    fig.delaxes(axes[j])\n",
  "\n",
  "plt.tight_layout()\n",
  "plt.show()"
 ]
},
{
 "cell_type": "markdown",
 "id": "ee7c1d4e",
 "metadata": {},
 "source": [
  "#From the provided image, each subplot visualizes the distribution of categorical responses for services like PhoneService, MultipleLines, InternetService
 ]
},
{
 "cell_type": "code",
 "execution_count": 102,
 "id": "00f35a61",
 "metadata": {},
 "outputs": [
  {
   "ename": "ValueError",
   "evalue": "Could not interpret input 'PaymentMethod'",
   "output_type": "error",
   "traceback": [
    "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
    "\u001b[1;31mValueError\u001b[0m                                Traceback (most recent call last)",
    "Cell \u001b[1;32mIn[102], line 2\u001b[0m\n\u001b[0;32m      1\u001b[0m plt\u001b[38;5;241m.\u001b[39mfigure(figsize \u001b[38;5;241m=\u001b[39m (\u001b
    "File \u001b[1;32m~\\anaconda3\\Lib\\site-packages\\seaborn\\categorical.py:2943\u001b[0m, in \u001b[0;36mcountplot\u001b[1;34m(data, x, y, hue, order, h
    "File \u001b[1;32m~\\anaconda3\\Lib\\site-packages\\seaborn\\categorical.py:1530\u001b[0m, in \u001b[0;36m_BarPlotter.__init__\u001b[1;34m(self, x, y, hu
    "File \u001b[1;32m~\\anaconda3\\Lib\\site-packages\\seaborn\\categorical.py:541\u001b[0m, in \u001b[0;36m_CategoricalPlotter.establish_variables\u001b[1;
    "\u001b[1;31mValueError\u001b[0m: Could not interpret input 'PaymentMethod'"
   ]
  },
  {
   "data": {
    "text/plain": [
     "<Figure size 500x600 with 0 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  }
 ],
 "source": [
  "plt.figure(figsize = (5,6))\n",
  "ax = sns.countplot(x = \"PaymentMethod\", data = df, hue = \"Churn\")\n",
  "plt.title(\"Churn Customers by Payment Method\")\n",
  "ax.bar_label(ax.containers[0])\n",
```

```
    "plt.show()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "4b495489",
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.11.7"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```