```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "ajXrC9XDfi1Q"
   },
   "source": [
    "##Step 1 : Downloading the dataset of chum_modelling(csv format)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "94VuTf5HfxkQ"
   },
   "source": [
    "##Step 2 : Importing the libraries and loading the dataset"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 25,
   "metadata": {
    "id": "2LqTZHCQUZgw"
   },
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "import pandas as pd"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 26,
   "metadata": {
    "id": "pvT9ibsTf8f8"
   },
   "outputs": [],
   "source": [
    "data = pd.read_csv('/content/Churn_Modelling.csv')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 27,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 270
    },
    "id": "AXtFMOoTgCLS",
    "outputId": "42c21063-5f17-44f6-eee5-91f5c2b72551"
   },
   "outputs": [
    {
     "data": {
      "text/html": [
       "\n",
       "  <div id=\"df-47df7257-a0b7-4341-a855-60328ad07fee\">\n",
```

```
"    <div class=\"colab-df-container\">\n",
"      <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>RowNumber</th>\n",
"      <th>CustomerId</th>\n",
"      <th>Surname</th>\n",
"      <th>CreditScore</th>\n",
"      <th>Geography</th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Tenure</th>\n",
"      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>EstimatedSalary</th>\n",
"      <th>Exited</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1</td>\n",
"      <td>15634602</td>\n",
"      <td>Hargrave</td>\n",
"      <td>619</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>42</td>\n",
"      <td>2</td>\n",
"      <td>0.00</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101348.88</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>2</td>\n",
"      <td>15647311</td>\n",
"      <td>Hill</td>\n",
"      <td>608</td>\n",
"      <td>Spain</td>\n",
```

```
"    <td>Female</td>\n",
"    <td>41</td>\n",
"    <td>1</td>\n",
"    <td>83807.86</td>\n",
"    <td>1</td>\n",
"    <td>0</td>\n",
"    <td>1</td>\n",
"    <td>112542.58</td>\n",
"    <td>0</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>2</th>\n",
"    <td>3</td>\n",
"    <td>15619304</td>\n",
"    <td>Onio</td>\n",
"    <td>502</td>\n",
"    <td>France</td>\n",
"    <td>Female</td>\n",
"    <td>42</td>\n",
"    <td>8</td>\n",
"    <td>159660.80</td>\n",
"    <td>3</td>\n",
"    <td>1</td>\n",
"    <td>0</td>\n",
"    <td>113931.57</td>\n",
"    <td>1</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>3</th>\n",
"    <td>4</td>\n",
"    <td>15701354</td>\n",
"    <td>Boni</td>\n",
"    <td>699</td>\n",
"    <td>France</td>\n",
"    <td>Female</td>\n",
"    <td>39</td>\n",
"    <td>1</td>\n",
"    <td>0.00</td>\n",
"    <td>2</td>\n",
"    <td>0</td>\n",
"    <td>0</td>\n",
"    <td>93826.63</td>\n",
"    <td>0</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>4</th>\n",
"    <td>5</td>\n",
"    <td>15737888</td>\n",
"    <td>Mitchell</td>\n",
"    <td>850</td>\n",
"    <td>Spain</td>\n",
"    <td>Female</td>\n",
"    <td>43</td>\n",
"    <td>2</td>\n",
"    <td>125510.82</td>\n",
"    <td>1</td>\n",
"    <td>1</td>\n",
"    <td>1</td>\n",
"    <td>79084.10</td>\n",
"    <td>0</td>\n",
```

```
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-47df7257-a0b7-4341-a855-60328ad07f
"              title=\"Convert this dataframe to an interactive table.\"\n",
"              style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.0
-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.0
.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47
  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-47df7257-a0b7-4341-a855-60328ad07fee button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
```

```
    "\n",
    "        async function convertToInteractive(key) {\n",
    "          const element = document.querySelector('#df-47df7257-a0b7-4341-a855-60328ad07fee');\n",
    "          const dataTable =\n",
    "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
    "                                                     [key], {});\n",
    "          if (!dataTable) return;\n",
    "\n",
    "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
    "            '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table not
    "            + ' to learn more about interactive tables.';\n",
    "          element.innerHTML = '';\n",
    "          dataTable['output_type'] = 'display_data';\n",
    "          await google.colab.output.renderOutput(dataTable, element);\n",
    "          const docLink = document.createElement('div');\n",
    "          docLink.innerHTML = docLinkHtml;\n",
    "          element.appendChild(docLink);\n",
    "        }\n",
    "      </script>\n",
    "    </div>\n",
    "  </div>\n",
    "  "
   ],
   "text/plain": [
    "   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \\\\n",
    "0          1    15634602  Hargrave          619    France  Female   42   \n",
    "1          2    15647311      Hill          608     Spain  Female   41   \n",
    "2          3    15619304      Onio          502    France  Female   42   \n",
    "3          4    15701354      Boni          699    France  Female   39   \n",
    "4          5    15737888  Mitchell          850     Spain  Female   43   \n",
    "\n",
    "   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \\\n",
    "0       2       0.00              1          1               1   \n",
    "1       1   83807.86              1          0               1   \n",
    "2       8  159660.80              3          1               0   \n",
    "3       1       0.00              2          0               0   \n",
    "4       2  125510.82              1          1               1   \n",
    "\n",
    "   EstimatedSalary  Exited  \n",
    "0        101348.88       1  \n",
    "1        112542.58       0  \n",
    "2        113931.57       1  \n",
    "3         93826.63       0  \n",
    "4         79084.10       0  "
   ]
  },
  "execution_count": 27,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "file = pd.DataFrame(data)\n",
 "file\n",
 "file.head()"
]
},
{
 "cell_type": "code",
 "execution_count": 28,
```

```
 "metadata": {
  "id": "FaB9225fgQQz"
 },
 "outputs": [],
 "source": [
  "file['HasCrCard'] = file['HasCrCard'].astype('category')"
 ]
},
{
 "cell_type": "code",
 "execution_count": 29,
 "metadata": {
  "id": "9bEcRqXxiTpi"
 },
 "outputs": [],
 "source": [
  "file['IsActiveMember'] = file['IsActiveMember'].astype('category')\n",
  "file['Exited'] = file['Exited'].astype('category')"
 ]
},
{
 "cell_type": "code",
 "execution_count": 30,
 "metadata": {
  "id": "Prq2dJp3i_cP"
 },
 "outputs": [],
 "source": [
  "file = file.drop(columns=['RowNumber', 'CustomerId', 'Surname'])\n",
  "#Removing the RowNumer,CustomerId and Surname column labels from the dataset"
 ]
},
{
 "cell_type": "code",
 "execution_count": 31,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 206
  },
  "id": "68A6e0MykATx",
  "outputId": "090b3b46-fb60-439b-eaf8-341c89bc737c"
 },
 "outputs": [
  {
   "data": {
    "text/html": [
     "\n",
     "  <div id=\"df-37c875f5-fb98-4f59-a278-75a500bfe1c0\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <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>CreditScore</th>\n",
"      <th>Geography</th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Tenure</th>\n",
"      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>EstimatedSalary</th>\n",
"      <th>Exited</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>619</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>42</td>\n",
"      <td>2</td>\n",
"      <td>0.00</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101348.88</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>608</td>\n",
"      <td>Spain</td>\n",
"      <td>Female</td>\n",
"      <td>41</td>\n",
"      <td>1</td>\n",
"      <td>83807.86</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>112542.58</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>502</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>42</td>\n",
"      <td>8</td>\n",
"      <td>159660.80</td>\n",
"      <td>3</td>\n",
"      <td>1</td>\n",
```

```
"      <td>0</td>\n",
"      <td>113931.57</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>699</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>39</td>\n",
"      <td>1</td>\n",
"      <td>0.00</td>\n",
"      <td>2</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>93826.63</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>850</td>\n",
"      <td>Spain</td>\n",
"      <td>Female</td>\n",
"      <td>43</td>\n",
"      <td>2</td>\n",
"      <td>125510.82</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>79084.10</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-37c875f5-fb98-4f59-a278-75a500bfe1c
"              title=\"Convert this dataframe to an interactive table.\"\n",
"              style=\"display:none;\">\n",
"        \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.0
-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.0
.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47
  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
```

```
 "      display: none;\n",
 "      fill: #1967D2;\n",
 "      height: 32px;\n",
 "      padding: 0 0 0 0;\n",
 "      width: 32px;\n",
 "    }\n",
 "\n",
 "    .colab-df-convert:hover {\n",
 "      background-color: #E2EBFA;\n",
 "      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
 "      fill: #174EA6;\n",
 "    }\n",
 "\n",
 "    [theme=dark] .colab-df-convert {\n",
 "      background-color: #3B4455;\n",
 "      fill: #D2E3FC;\n",
 "    }\n",
 "\n",
 "    [theme=dark] .colab-df-convert:hover {\n",
 "      background-color: #434B5C;\n",
 "      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
 "      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
 "      fill: #FFFFFF;\n",
 "    }\n",
 "  </style>\n",
 "\n",
 "      <script>\n",
 "        const buttonEl =\n",
 "          document.querySelector('#df-37c875f5-fb98-4f59-a278-75a500bfe1c0 button.colab-df-convert');\n",
 "        buttonEl.style.display =\n",
 "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
 "\n",
 "        async function convertToInteractive(key) {\n",
 "          const element = document.querySelector('#df-37c875f5-fb98-4f59-a278-75a500bfe1c0');\n",
 "          const dataTable =\n",
 "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
 "                                                     [key], {});\n",
 "          if (!dataTable) return;\n",
 "\n",
 "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
 "            '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table not
 "            + ' to learn more about interactive tables.';\n",
 "          element.innerHTML = '';\n",
 "          dataTable['output_type'] = 'display_data';\n",
 "          await google.colab.output.renderOutput(dataTable, element);\n",
 "          const docLink = document.createElement('div');\n",
 "          docLink.innerHTML = docLinkHtml;\n",
 "          element.appendChild(docLink);\n",
 "        }\n",
 "      </script>\n",
 "    </div>\n",
 "  </div>\n",
 "  "
],
"text/plain": [
 "   CreditScore Geography  Gender  Age  Tenure    Balance  NumOfProducts  \\\n",
 "0        619    France  Female   42       2       0.00              1   \n",
 "1        608     Spain  Female   41       1   83807.86              1   \n",
 "2        502    France  Female   42       8  159660.80              3   \n",
 "3        699    France  Female   39       1       0.00              2   \n",
```

```
    "4         850     Spain  Female   43       2  125510.82            1   \n",
    "\n",
    "   HasCrCard IsActiveMember  EstimatedSalary Exited  \n",
    "0         1              1        101348.88      1  \n",
    "1         0              1        112542.58      0  \n",
    "2         1              0        113931.57      1  \n",
    "3         0              0         93826.63      0  \n",
    "4         1              1         79084.10      0  "
   ]
  },
  "execution_count": 31,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "file.head()\n",
 "#Displaying the data without first three columns"
]
},
{
"cell_type": "markdown",
"metadata": {
 "id": "LbuB9TeamG-k"
},
"source": [
 "##3.Performing the visualizations\n",
 " - Uni-variate Analysis\n",
 " - Bi-variate Analysis\n",
 " - Multi-variate Analysis\n",
 "\n",
 "\n",
 "\n"
]
},
{
"cell_type": "code",
"execution_count": 34,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/",
  "height": 296
 },
 "id": "QtXUXlkZlzJw",
 "outputId": "93211e08-5c9f-4272-dbcc-fba6b2ac0792"
},
"outputs": [
 {
  "data": {
   "text/plain": [
    "<matplotlib.axes._subplots.AxesSubplot at 0x7fdc53fb4890>"
   ]
  },
  "execution_count": 34,
  "metadata": {},
  "output_type": "execute_result"
 },
 {
  "data": {
   "image/png": "iVBORw0KGgoAAAANSUhEUgAAAYIAAAEGCAYAAABo25JHAAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAA
```

```
XRwbG90bGliLm9yZy+WH4yJAAAQ90lEQVR4nO3df5BdZ13H8ffHQKwigtrVwSRtMhh+REALS1AZpQNUU9QEBTRhUKq
d5znPvfjez2c+e85z7nFQVkqR2fd+oC5AkjZZBIEmNMwgkqXEGgSQ1ziCQpMY9YNQFnK4LL7yw1q5dO+oyJOmccuONN3
8JMlNSW5J8owu65EknaqzIEiyAtgDXA5sALYl2TAw7OXA/qq6BNgK/F1X9UiShuvyiGAjMF1VR6vqBLAP2DIwpoAf7m0/BLi
nf5BVbUX2AswPj5+v++k84SXvPP+voXOQze+9ndHXYI0El0eERwH1vS1V/f6+l0J7Aeoqk8AFwAXdliTJGlAl0FwkEFifZF2SlcxN
1BVk8xNAvf37erbPgw8ucsaJEkL85PFktQ4g0CSGmcQSFLjDAJJapxBIEmNMwgkqXEGgSQ1ziCQpMYZBJLUOINAkhpnEEh
APcBkwAxxMMtG7PSUAVfWnfeNfCFzSVT2SpOG6PCLYCExX1dGqOgHsA7YsMH4bczewlyQtoy6DYBVwrK+K890+s7RZKLg
dBsFBYH2SdUlWMvfLfmJwUJHAT8CfKLDWiRJ8+gsCKrqJLADOADcBuyvqkNJdifZPJ3Dd0K7CvqqqrWiRJ8+vs8lGAqpoEJgf6
ziCQpMYZBJLUOINAkhpnEEhS4wwCSWqcQSBJjes0CJJsSnIkyXSSnfOM+a0kh5McSnJdl/VIkk7V2a0qk6wA9gCXATPAwSQ
f0eATwiyceTfDLJpmFvlGR7kqkkU7Ozsx2VK0ltGvVk8QOA9cClwDbgLUkeOjioqvZW1XhVjY+NjS1ziZJ0fusyCI4Da/raq3t9/
wG7K+qQ0l2J9ncG3YAuDPJYeAjwEuq6s6uapIknaqzy0cBqmoSmBzo29W3XTDVzYckaQRGPVksSRoxg0CSGmcQSFLjDA
JpiRHkkwn2Tlk/xVJZpPc3Hv8QZf1SJJO1dmtKpOsAPYAwlwEzwMEkE1V1eGDoe6tqR1d1SJIW1uURwUZguqqOVtUJYB+wpv
VY2PjY0ta4GSdL7rMgiOA+/1/4a/u9/XVd1ZVd/sNd8KPKHHG6O
kbMIJCkxhkEktS4BecIkly90K+90P6qesOZLUeStNwWWmjOIZLUeStNwWmyx+cO/5kcATuffyz18HbuiqKEnS8lkwCKrqLwGSfAx4
7n8tEfBO6uqjcBM0nWdVSTJGkZLSkIkrwCeCn3LgfxQODdXVUHhoEJ6qqqgAJI8qD
Jks5hSz01dNmQvsvPZCGSpNFYMAiS/HGSW4FHZFHJrml7/F54JbF3jzJpiRHkkwn2bnAuGclqSTjp/8tSJLuj8VODV0HfAD4a6D/
TgD7gC1Dxr0KeDXwjdOoW5J0hiwWBNf1mm8EpnrPN/a1F7IKONbXnun1fVuSJwBrquqfF3qjJNuTTCWZmp2dXeRjJUmnY7Egq
DDhhLEkLa+lXj76tCSTSR6R5gCSTSR6W5DHAJ1l8raGDwPok65KsuqaqvVNWFVbW2qtb23nNzVS12ykmSdAYt9ZPFy28
0z9tKl1CJJOrOWemrofcfVf8PcBwdeQ5A7uv6KooSdL
lsdhaQyuWqxBJ0mgs9XMEkqTzlEEgSY0zCCSpcQaBJDXOIJCkxhkEktQ4g0CSGmcQSFLjDAJJalynQZBkU5IjSaaT7Byy/4+S
P+Ceu571PAhXNJWkZbfUW1XeF6uAY33tGeBJg4OSvAC4GlgJPHXYGyZDmwHuOiii854oZLUspFPFlfVnqp6OPBS4OXzjN
YjO5giq6mSSHcABYVAwbVUdSrIbmKqqCWBHkqcD3wT2Ay9YsSWCQaBJDXOIJCkxnhUaBEk2JTmSZDrJziyAHuBzYAGxLsmFg2E3AeFU9DrgeeeE1X9UiShuvyns
XOIJCkxnUaBEk2JTmSZDrJziH7r05yOMktSf41ycvd1iNJOlVnQZBkBbAHuBzYAGxLsmFg2E3AeFU9DrgeeeE1X9UiShuvyns
tSaaSTM3Ozp7BEiVJZ8VkcZLnAePAa4ftr6q9VTVeVeNjY2PLW5wknee6PDV0HFjT117d6/seSZ4OvAx4SlV9s8N6JElDdHlE
O0lSR7o8NURVTQKTA327+raf3uXXlyQt7qyYLJYkjY5BIEmNMwgkqXEGgSQ1ziCQpMYZBJLUOINAkhpnEEhS4wwCSWqc
UGSFcAe4HJgA7AtyYaBYV8ErgCu66oOSdLCurxn8UZguqqOAiTZB2wBDt8zoKq+0Nv3nQ7rkCQpMtTQ6uAY33tmV7fa
tJku6DzoKgqk4CO4ADwG3A/qo6lGR3ks29YU8ASZ6YZAZ4Dv
QSFLjDAJJalynQZBkU5IjSaaT7Byy//uTvLe3/1NJ1nZZjyTpVJ0FQZIVwB7gcmADsC3JhoFhFhVwJ3VdVPAW8EXt1VPZKk4bo8It
WpIjnVTcpgsZ+PduVV73/FGXoO/lz+Y9XnFG/j6+eL4dXQbBGVNVe4G9o67jfJRkqqrGR12HHNMifzeXT5amh48CavvbqXt/Q
1NVNQG8DXhXkmngy8yFZhy8yFhZaXp9x0tvJnc5nEP8AlqW1+sliSGcqQSa5NckeS/xOU/pFF
okcM/yH7cB+6vq0GirkuYkeQ/wCeCRSWaSvHDUNUm9Dnqms53LjEhSY3ziECSGmcQSFLjDAJJapxBIEmNMwgkqXEGgTSPJP9xmuM
S7Opt/0qSjyXx/6FGqrOb10vnvnmUk+BFwDHEzyb8DfAM+oqu

INAkhpnEEhS4wwCSWqcQSBJjft/MclTBktvBQYAAAAASUVORK5CYII=\n",
     "text/plain": [
      "<Figure size 432x288 with 1 Axes>"
     ]
    },
    "metadata": {
     "needs_background": "light"
    },
    "output_type": "display_data"
   }
  ],
  "source": [
   "#Importing seabron library\n",
   "import seaborn as sns\n",
   "depth = file['Exited'].value_counts(normalize=True).reset_index()\n",
   "sns.barplot(data=depth,x='index',y='Exited')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 35,
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 112
   },
   "id": "lGKO-vwWmrNk",
   "outputId": "d186299a-99ef-4fdf-bc4d-0ec273eccf4c"
```

```
  },
  "outputs": [
   {
    "data": {
     "text/html": [
      "\n",
      "  <div id=\"df-fb780043-474b-4f00-960b-f2d445aeb438\">\n",
      "    <div class=\"colab-df-container\">\n",
      "      <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>index</th>\n",
      "      <th>Exited</th>\n",
      "    </tr>\n",
      "  </thead>\n",
      "  <tbody>\n",
      "    <tr>\n",
      "      <th>0</th>\n",
      "      <td>0</td>\n",
      "      <td>0.7963</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>1</th>\n",
      "      <td>1</td>\n",
      "      <td>0.2037</td>\n",
      "    </tr>\n",
      "  </tbody>\n",
      "</table>\n",
      "</div>\n",
      "      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-fb780043-474b-4f00-960b-f2d445aeb4
      "              title=\"Convert this dataframe to an interactive table.\"\n",
      "              style=\"display:none;\">\n",
      "        \n",
      "  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
      "       width=\"24px\">\n",
      "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
      "    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.0
-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.0
.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47
  </svg>\n",
      "      </button>\n",
      "      \n",
      "  <style>\n",
      "    .colab-df-container {\n",
      "      display:flex;\n",
```

```
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-fb780043-474b-4f00-960b-f2d445aeb438 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-fb780043-474b-4f00-960b-f2d445aeb438');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"            '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table not
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
"    </div>\n",
```

" </div>\n",
" "
],
"text/plain": [
" index Exited\n",
"0   0 0.7963\n",
"1   1 0.2037"
]
},
"execution_count": 35,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"depth"
]
},
{
"cell_type": "markdown",
"metadata": {
"id": "1cbbc8i4nTz1"
},
"source": [
"From the above relation analysis, it can be said as the data is imbalanced\n",
"For to correct this, we are processing and visualizing the data using matplotlib libary below"
]
},
{
"cell_type": "code",
"execution_count": 36,
"metadata": {
"id": "vr48d74XnNtJ"
},
"outputs": [],
"source": [
"import matplotlib.pyplot as plt"
]
},
{
"cell_type": "code",
"execution_count": 37,
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 441
},
"id": "Q42zIiyfn9t3",
"outputId": "7266b78c-4d79-4e74-8681-e3086057c216"
},
"outputs": [
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAAsgAAAGoCAYAAAABbtxOxAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzde7xcdXno/89jCAYkCoRAMTs0WEC5iCEElOiVBKLpqan
9Z67su88zsybOerPmu9Y3MRJIkSVLNm1odgCRJkjSYWCBLkiRJFRbIkiRJUoUFsiRJklRhgSxJkiRVbNPqABphl112yXHjxrU6D
lSZKkCgtkSZIkqaIj+yBLGhpeffVVuru7eemlI1odSq9GjBhBV1cXw4cPb3UokjTg2iEPQ99zsQWypLbV3d3NyJEjGTduHBHR
IX0G+eBzrm11CJu18NIzWh2CNOgN5qQMgz8+aagYM7MfgdTs+qEd8lxeFxyyySbEwaduHBX
PKTn2xUaJI0ZLRzLvYMsqQhYc2aNbzzne/kscceA+D000/nG9/4BlAbEvnZZ5/l3HPP5ec//nnHPOAeDSSy/lkEMO4cA
CwCjg1M58s+5gJTAPWA3+embc3Om5J7evFF19k/PjxG+ZnzpzpJqaeeyuWXX87UqVM5++yzWb16NZ/4xCdet90lI1zCI488s

a5r8AXJaZcyLi69QK3yvL8+rM3CsiTivrnRoR+wGnAfsDbwf+JSL2ycz1TYhdUhvq7We9Y445hhtvvJEZM2bw4IMPbnE/d9xx
4OQyfVKZpyw/uqx/EjAnM1/OzCeAZcChjYxbUmf6zW9+w5IlS9h+++1ZvXr1FtfPTGbOnMmiRYtYtGgRy5YtY9q0aU2IVJI6V
777ct1113HmmWfy6quvvm75yJEjWbt27Yb54447jlmzZvHCCy8AsHz5clasWMERRxzBTTfdxIsvvsjatWv54Q9/2NT3IUntrI
TRRVxwwQUb1hs1ahSHH344BxxwACeccAKXXnopS5Ys4bDDDDgNghx124Dvf+Q4TJkzg1FNP5T3veQ+77rorhxxySNPfoy
xFyxYsMUYB/uoOo6kJ23ekiVL2HfffVsdxhZtKs6IWJiZE1sUUlPUm4ulZhjsx/yB1Mz6oV3yMPQtFzesi0VmzszMrswcR+0iu
R8Qy4DlqRTWZ+WhE3AAspnY2eoZ3sJAkSVKjNKVAzsx7gHvK9ONs4i4UmfkS8OFetr8YuLhxEUqSJEk1jqQnSZIkVVggS5I
SHwscx8JSLeDFwLHAysAk7NzCfLPmYC04D1wJ9n5u3NfyeSBrtW5GEY+FzsGWRJ6qf169czY8YMfvSjH7F48WKuv/56Fi9e
vYdtttOe2007j55pu3vGETRUQX8AfA1WU+gKOA75VVZgMnl+mTyjxl+dFl/ZOAOZn5cmY+ASxjE3cjkqRWaEQutkCWpH5a
Avg9J2qRG5GILZEnqUBFxIrAiMxc26zUz86rMnJiZE0ePHt2sl5WkAeVFepLUT2PGjOHpp187sdrd3c2YMZs8sdoqhwOTI2I
MGfOHCZPntzqsDbIzJmZ2ZWZ46hdZHdXZn4UuBs4paw2BejjprDe3zFOW35WZWdpPi4g3lztg7A3Mb9LbkKTNakQu9gy
gEzMnN988OWNNg1Ow9DY3KxBbIkbYVJkyYxadKkVoexRZl5D3BPmX6cTdyFIjNfAj7cy/YXAxc3LkJJ6r+BzsV2sZAkSZIqLJ
WRRx4Z0NeXpHbTKXnYM8iStBWmTp3Kbbfd1uowJGnIakQetkCWpK1wxBFHsPPOO7c6DEkashqRhy2QJUmSpAoLZEm
6BEka0hqRhy2QJUmDzsHnXNvqEJpm4aVntDoESRuxD7IkSZJU0bACOSJGRMT8iHh6NiAtK+54R8bOIWBYR342IbUv7
RT0NWF3aLyvrERH7AacB+wPHA1dExLAGxi2pYWMYMYJVq1YN2uScmaxatYoRI0a0OhRJaojBnoehf7m5YX2QM/OViPggcxHmph
JLUEO2Qh6HvubihF+mVM70Lgb2ArwE/B57PzHVllW5gTJkeAzwNkO031vvZbXW6mtNB6YD7HHHgP+XiQNfw
Scue4Clpfp5cBYgLL8bcCqavsmtpEkSZIGVCPvYjE6InYs09tT09sBxwBLqBXKp5W/Kp5TVpgA3l+m5ZZ6y/K7Sj3kucFq5y8WewN7A/Eb
JkiQNYY28i8VDwEGbaH+cWn/kjdtfAj7cy74uBi4e6BglSZKkjTmSniR1MAdtkqS0CWpM7moE2S1EcWyJLUwbKmt0GbvI
pN5YIEtSh3PQJkqngwtkSRoiHLRJkqioskCVJkqQKC2RJkiSpwgJZkiRJqrBAliRJkioskCVJ
DXflu6D/CfAp4G3Awt5LTH/B3B5A+OSJNWYhyWpyTZbIGfml4EvR8SnvxOrTYpJklSYhyWp+eoaSS8Mq26Tmc
naLPOwJDVJvWeQdwEWR8R84Owexsyc3JCoJEkbMw9LUp9WCf38ggJElbdH6rA5CkoaLeu1j8a6MDkST1zjwsSc1T710
Ykhqr3i4WH6zMvona/ThfakhEkqQ3MA9LUvPUexeLP6xMwOepPbzN2oUlV2XmZ
H+SI6IqIH0TEivL4fkR0bWGzdcBfZuZ+C4C4kmRER+wHnAndm5t7AnWUe4ARg7/KYDlxzvcChwHkRsVOf3qUktb
FwNzAOOrzNuSeoUfc7DkqT+qbdAHp2Z38zMdeXxLWB0vS5vS8SEeOAg4CfAbtl5WW/vGrzE9PI
1mSOo4/c7DkqS+qbdAPgv4CLUzvs8ApwBTt7RRRAynVhz/U2b+c2n+Vek6QXleUdqXA2Mrm3eVtt7aJWko6W8eHhsRd0
AJZn5j5VFc4GexDoFuLnSfsZ8D1pSuGLcDx0bETtkfkZJzu8D1pSuGLcDx0bETiWBH1vaJkGo6XMeLrxgWpL6qN77IbIB9YLpADIDOfi4iDtrDN4cDHgIcjYlFp+xvgEmBORJwFPAmcWpbdSu0Wb8up3ebtTIDMXB0RFwH3l/UuzMzVdcYtSYWfyuM
mw3cA3yWygXTwL0R0XPB9JGUC6YBIqLngunrB+TdSdIgUm+B/KaI2KnnOMwOZzcCzczN/AkQvi4/exPoJzOhXX7OAWXXGK
DVdXX2QM/Na4IPAr8rjg5n57UYGdk4bM0dhSR1grp/nsvMxciSYiSdqM/uThOi6YvoQ3XjD9ZxExh9oFeWs
7egUIkSZKkIcECWZIkSaqwQJYkSZIqLJAlSZIqLJAlSZIqLJAlSZIqLJAlSZIqLJAlSZIqLJAlSZIqLJAlSZIqLJAlSZIq
kCgtkSZIkqcICWZIkSaqwQlYgR8R8SsiFgREY9U2naWEKHX9pREexpVLYsSbEkSw
U+3xuYdbHYK0RQ07g5yPy3PZwe6ZOBfGgZvZ7e26j5j5RGH6ZwpvFdHfxhmzpAcG6NRlZBNB6B3
vjLW093t72BO8s8vL6723Rq3d2odHd7L3AocF5PUS1JnajaZBfKvvStcJyvOK0r4cGFtZr6u09dYuSaqD3d0kqe+aXSDPBXp+m+m
DjwDLgG8AnATLzOeDwH3lwPNhHnj9xdskdN6uuIGn3KMPiU4WU/s4BZAeyiju4/Z7m
r4EyodXeLiJ7ubmB3N2lQG+x5cqA1Iu9aIEtSH7SB2Sp7xo5UgGzrl9Emq9sEKgCAlmSJEmqsECWJmSKiyQJUmSpAoLZEmSJKmS
Zu3xuYdbHYIkSR3DM8iSJEmSRL2iSJElShQWyJEmSVGGBLEmSJFVYIEuSJEkVFsiSJElShQWyJEmSVGGBLEmSJFVYIEuSJEkVFsiSJElShQ
qgkaegwD0saStqiQAYOBZZl5uOZ+QowBzipqTFJ0lBiHpY0Z0GZGzT6gDqNAZ4ujLfLDby3ukKxKJsXWML8NuwDI
k5sdVxxyL/FYOLfYnDrxFzcaH6nNdD8Tm29dulsiRwYW5nvKm2SpOYWD0saMtqlQL4P2PDsihYLjkLtjkmShhLzKQh
KC2RJkiSpwgK5gSJTom1UTE30TEoxHxUPnbvOF+rOF+rnXsY7LD7W5eROwWEddSRxxOhh2h2h1XNJAaWacrnxrMRsTcAifxOMRsTAifhoRH2h2xCw
cLbt634ycy5exd+r8h2/CZidmX9U2n4Fzn9ttk5roGhigNhF7zvDTA/hM4ICK2y8WxWxRWcCzw
5n/t3zm/1A+2/kRsRdARPxhRPwsIh6iIH+JiN1K+9SIuLxMfysivhIR/9z+W97DANT21bjVD4Sr1Un0kLTOfysyvRsSwiLg0Iu4hG3ly+Zr4AxQ1vHGRjmr1x1Iu4Gf7L4NnjwHRxmAw2pP3SuzuZjp13T8wvEh8Z8T/cBiwgxkIpV6xlkTz4XdgpHz8mXzsUz
Z+OOpcEXFw+d4ujIjbI2L30n5PRFxWcv6bdIBeoqBdVMkIVEaV7W8q2z2z4atZEJN/Uaf1z+HSyKiP8ZEcOa9f9f7UUrcCf1CmTwe
mTgCuBP6qtP13YE1mvjzDwTuKmdA/xb4/bL+2z8n8Ntz8A5cCXStppGPdI5kHAHOAzv
ccChwHkRMRyYBZwBEBFvonZP3+8g9d3r8rNJ6D6b2Xbu4sv4rJed/HbgZmAeEEPthnbPKthOBP6+0AxAR+
dOeysx7K+t8pJwh2IagGf8/8FBZ9k9s9s/leSHwwTL5uqAIOLFs928VDfTKLzhYGg4GPv1BLEd+O1vsTX
Awtb8fbP5v9L/KrwUvR8QKYLfMfMfDPVdvnFvH6A37KGho3z/AHUCt55JR8PA56prN/TLeth4NHKd/txaiMVrqW
bmAVy8/1nz0Q5Y/ZXwCGl0P0WtS9rj5fL83o2/7cKYF5mnj7AsXasaszFwP3APcx6N6fBf4xM+dGbZ7u
qBW+h/WyvPr93fi7vEub4vcd7zYs5fQqx1DRaDebsNFacyvScGGF1d+3H7s7MD9Qi9hZPNzBxvBvaxyd7p
ogpqF53ASMi4k8rbkJ/JRtB+XuxPnxRM6AnMX5Tg1VjhYmMfwzdLXdc1m
HxIPB7mbmS2j+G6yPiIWqFSBeWyr3E3VH2IHaz1CLy+e1H6+dcyptJ0N/H+l7XgzxohYCDzb5FjVtaG6TyZWt/6JyjiPjAb+Cy1s
1J3eUfDWPWhc9DQGZ2Z2ZX9nEon8A/j4iHqD3FhUUmRJOAd2ZqRfAQUC7Oux/4cGYubxU8kqTE
WzZPREyNiMtbHYcKktVKj8nVEHBkRtwxEAHY2C2LSZ1rXXkGvNxOLowhriyyQ1c
/hwRHyi7PfBiPh+RGxf1vtWRsW9HZExP+JiBMrugLnIr5XcvM/VUaA
gNLf0eYBfgvoj4MXA4MI7aqHq7AkuoDefDeZ1VmTgClFFF2GgjMqyjUfe6ZI/udK2n
kG1NnskBWx4iIrwHvB14BngDkLOTG2M+r3L8uvL0MK/ioh/BQ4p7Tdm5m5VhqADyiF8Y7Uhq2+vbLshrKP
MDki/qrMjwD2KNPzLI5VLwktbNHgQ/1zGTmjIjYBVgA/AL4K2K2n
BGxT0S8BfjfwKmIz9to4AhgPrWf7jV/9j5U+iLvxuuL3o2NBJ4p4WFZv8EXD9Ft+xJLWn/uTr3
QrgZOBHwCHAQ9SO7v7mcz8Z8ZUR8/HzaVxXvq/+Az8r+/wZtCYK5xy+oFdxvBf5bZr5U8vTm3ACMz8V9b97SW
ckpnf6+Pr3gJclpleWS1J0iDgGWTpNbEq7BBT7K4P3qusAYetDiWJGnw8Ayy9ayyJEmSVKFBbIkSZJUYYEsSZIkVVggS
ZuboVsfRSOZiSYNdb7m4IwvkcePGsWDoaGI0yL2KJiRRJRkbIkiRJUoUFsiRjIkTRkX2QNbS9+
d3dzZ577tnqcBqRJ6f/BfsyF/h137WKhjvPSSy8xWkhUJoWxatSoUiuOjOhvtq7QO71i2vvUnQtJOpeQLcL4IRUBFeP7OzR6ehsL
55rrjiij6/3vnnn88Xv/jF/jF/oYrSVJba+fjrl912LFq0OYOx93b731Vb0y1c2o3KjRBRJkjpRRdeMAyHUHHgbmX

/iy++mH322Yf3v//9G/YpSZJq2uW46xlkDRkvvvgi48eP3zA/c+ZMTj31VC6//HKmTp3K2WefzerVq/nEJz7xuu0uueQSHnnnkl
w4AFwPLMPDEi9gTmAKOAhcDHMvOViHgzcC1wMLAKODUznyz7mAlMA9YDf56Ztzc6bnWe3n7qOeaYY7jxxhuZMWMC
KZW+FKeV5f2y8p6RMR+wGnA/sDxwBWl6JYGxG9+8xuWLFnC9ttvz+rVq7e4fmYyc+ZMFi1axKJFi1i2bBnTpk3b4naSJKk9j
kmr7766uuWjxw5krVr126YP+6445g1axYvvPACAMuXL2fFihUcccQR3HTTTbz44ousXbuWH/7wh019H5H5IktYN2OO42uov
66quZP38+I0eO5IgjjuCiiy7iggsu2LDeqFGjOPzwwwznggAM44YQTuPTSS1myZAmHHXYYADvssAPf+c53mDBhAqeeeeirvec
4L3A+WWb75T2a8o236MXEydOzAULFjTkfQ0l7TqS3pIlS9h3330HKJrGaZc41RgRsTAzJ7Y6jkYyF2swcSS9xminY9mmYu0
LWL7O7KzI8CdwOnlNWmADeX6blInrL8rqyd3p4LnBYRby53wNgbmN+ouCVJkjS0teI+yJ8F5kTERcADwDWl/Rrg2xGxDHi
6MiIcjYlFELChtO0fEvIhYWp53Ku0REV+JiGUR8VBETKjsZ0pZf2lETOnt9SSp3VkgS9LQ8HuZOb5ytfa5wJ2ZuTdwZ5kHOIHa
a5yTgyDI9m1o3uM+W9mvLRdL3RsSOEbF7WXdeZj4HEBHzqI1uen1zw5Y0mHXKMdczyFIDrF+/nhkzZvCjH/2IxYsXc/311
nz+fvfbai3e84x1su+22nHbaadx8881b3lBqjPdn5gRq3SdmRMQR1YXlbPGAjBqVmVdl5sTMnDh69OiB2KUkbVYjjrkWyFI
8CxwCO8fnCmjQdtOqPczeJ9wJrSFeN24NiI2KlcnHdsaZOkjjMkL9Jr1XjszRwbXa01ZswYnn76te6a3d3djBnzhu6aUjPsBvw
B06VKeeOlJxowZw5w5w5c7juuutaHZaGoDI403s20b4KOHoT7QnM6GVfs4BZAx2jJG2NRhxzLZDV8Vpx5n6bbbbh8ssv57jjj
CWZIkSaqwQJYkSZIqLJAlSZKkCm/zpo73iwvfPaD72+NzD29xnbPOOotbbrmFXXfdlUceeWRAX1+SpMGqU465nkGWGmD
3tTx6rlFzEA7/fTTueeee3j22Wfp6uriggsYNq0aU2PQ5KkZuqUY64FstQA119/fatDkCRpSGJEMdcuFpIkSVKFBbIkSZJUYY
X2NsWEFckSMiIj5EfFgRDwaEReUdhOzGmrEiBGsWrVq0P6DzUrVrVrFiBEjWh2KhKho6zgSWV+S8Al2XmXsBqoOdqlmnA6
q1xHzqRon57cC/RMQ+mbm+gbGrjXV1ddHd3c3KlStbHUZdRowYQVdXV6vDkCRpSHUqvRowYQVdXV6vD0BAQEV3AHwAXA38REQEcBfxRWWU2cD
LwGeAkWV+FPB8Zq4r893AmDI9Bngal1Jmagmpifrux2k4k5Id
iHgSmEPtF7wvAztGRM8viF3A8BjK9HBgLUJa/DVhVbd/ENq9jLpbUCZpyF4vMfB64GziMBiZmSdJrMnNmZnZl5jhq13LclZkf
q0r2zWl/RpgVGn/C+BcgMx8FLgRWAzcBsywOwCJUmtpVp70fH33v5X0i2UyFeI/r9QEsvA/HfqUBAWldS3gxBeJpnmDEMaH6WyfYsR3lE8A0bGIv
mSKiyQJUmSpAoLZEmSJKkCAlmSJEmqsECWJEmSpAoLZEmSJKkCAlmSJEmqsECWJEmSpAoLZEmSJKkCAlmSJEmqsECWJEmSpAoLZEmSJKkCAlmSJEmSJ
YlySpxjwsSU222QI5M78MfDkiPpWWDZTX21STJKkwjwsSc23pTPlAGTmVyPid4Fx1W0y89HXqqjAPS1Lz1FUgR8S3gd8BF
fc+I+FIELIuI70bEtqX9zWV+WV+Vk+rrKvmaX9sYg4boDelyQNOvWeQd4FWBwR84GXexozc3JDopIkbay/efhI4KjMfCEihgM/i
P5sVLpkvFBmh5dHAkcBf1TaZwOfpzYi6snlOcDvlxbHGcPFSaE8sXwH3Dqpc9cCLBeZXj2+gK3jxxTPYV6wB8C19lhBjYy6IO81Y+j9cP4Zc5Z/sF6xEPj5STVFnIkTTE8kaoUncyrLMvR57lTj25g4JMrR2VN0kN2GSw/yZ+fEGnCIqQ9cZgdaBdyOyjSQoMLFM+bKMa4VjNEigWlqtEaYoLdcKP2M0kHC5jyzQJFxumd1e4fjaHrYuHSWqZuFWa3KMGnxNjv4qE3IJ7SDiH0xQjyFrGd88c3w07o8xGZG3D6RJOHABm3eh7GT+3ku6g8+XY7nCNQhCEGTtNq4DCX/YJ8aSmPxZDQFzc3lAfmlJ0GGcSlUDI9m+E0JvaUL2bZ8fzDzF0aAAlHPagK2I/OmBNzWVE7ZmaOAuOoVFgKSHoQAXsa7NqAwdXk9RPiuHmwxqSLRMtHXIMZxBvTbyO8i7ZFU2ZqMNmVV3r1yaxTpfe3Njkg

/iy++mH322Yf3v//9G/YpSZJq2uW46xlkDRkvvvgi48eP3zA/c+ZMTj31VC6//HKmTp3K2WefzerVq/nEJz7xuu0uueQSHnnkl
w4AFwPLMPDEi9gTmAKOAhcDHMvOViHgzcC1wMLAKODUznyz7mAlMA9YDf56Ztzc6bnWe3n7qOeaYY7jxxhuZMWMC
KZW+FKeV5f2y8p6RMR+wGnA/sDxwBWl6JYGxG9+8xuWLFnC9ttvz+rVq7e4fmYyc+ZMFi1axKJFi1i2bBnTpk3b4naSJKk9j
kmr7766uuWjxw5krVr126YP+6445g1axYvvPACAMuXL2fFihUcccQR3HTTTbz44ousXbuWH/7wh019H5IktYN2OO42uov
66quZP38+I0eO5IgjjuCiiy7iggsu2LDeqFGjOPzwwwznggAM44YQTuPTSS1myZAmHHXYYADvssAPf+c53mDBhAqeeeeirvec
4L3A+WWb75T2a8o236MXEydOzAULFjTkfQ0l7TqS3pIlS9h3330HKJrGaZc41RgRsTAzJ7Y6jkYyF2swcSS9xminY9mmYu0
LWL7O7KzI8CdwOnlNWmADeX6bllnrL8rqyd3p4LnBYRby53wNgbmN+ouCVJkjS0teI+yJ8F5kTERcADwDWl/Rrg2xGxDHi
6MiIcjYlFELChtO0fEvIhYWp53Ku0REV+JiGUR8VBETKjsZ0pZf2lETOnt9SSp3VkgS9LQ8HuZOb5ytfa5wJ2ZuTdwZ5kHOIHa
a5yTgyDI9m1o3uM+W9mvLRdL3RsSOEbF7WXdeZj4HEBHzqI1uen1zw5Y0mHXKMdczyFIDrF+/nhkzZvCjH/2IxYsXc/311
nz+fvfbai3e84x1su+22nHbaadx8881b3lBqjPdn5gRq3SdmRMQR1YXlbPGAjBqVmVdl5sTMnDh69OiB2KUkbVYjjrkWyFI
8CxwCO8fnCmjQdtOqPczeJ9wJrSFeN24NiI2KlcnHdsaZOkjjMkL9Jr1XjszRwbXa01ZswYnn76te6a3d3djBnzhu6aUjPsBvw
B06VKeeOlJxowZw5w5c7juuutaHZaGoDI403s20b4KOHoT7QnM6GVfs4BZAx2jJG2NRhxzLZDV8Vpx5n6bbbbh8ssv57jjj
CWZIkSaqwQJYkSZIqLJAlSZKkCm/zpo73iwvfPaD72+NzD29xnbPOOotbbrmFXXfdlUceeWRAX1+SpMGqU465nkGWGmD
3tTx6rlFzEA7/fTTueeee3j22Wfp6uriggsuYNq0aU2PQ5KkZuqUY64FstQA119/fatDkCRpSGjEMdcuFpIkSVKFBbIkSZJUYY
X2NsWEFckSMiIj5EfFgRDwaEReUdhOzGmrEiBGsWrVq0P6DzUxWrVrFiBEjWh2KhKho6zgSWV+S8Al2XmXsBqoOdqlmnA6
q1xHzqRon57cC/RMQ+mbm+gbGrjXV1ddHd3c3KlStbHUqvRowYQVdXV6vD0BAQEV3AHwAXA38REQEcBfxRWWU2cD
LwGeAkWV+FPB8Zq4r893AmDI9BngaIDPXRcSasv4Y4N7KPqvbvSEiHA7KPqvbSBrCOvWY29A+yKXf2yYBgTAP+Dl1Jmagmpifrux2k4k5Io
iHgSmEPtF7wvAztGRM8viF3A8jK9HBgLUJa/DVhVbd/ENq9jLpbUCZpyF4vMfB64GziMBiZmSdJrMnNmZnZl5jhq13LclZkf
q0r2zWl/RpgVGn/C+BcgMx8FLgBWAzcBszwQmlJnayRd7HYHZhdbgX0JuCGzLwlIhYDXYBXp+Yv10S83PUznaQmY9
mSKiyQJUmSpAoLZEmSJKkCAlmSJEmqsECWJEmSpAoLZEmSJKkCAlmSJEmqsECWJEmSpAoLZEmSJKkCAlmSJEmSJ
YlySpxjwsSU222QI5M78MfDkiPpWZX21STJKkwjwsSc23pTPIAGTmVyPid4Fx1W0y89oGxSVJqjAPS1Lz1FUgR8S3gd8BFgh
fc+I+FlELIuI70bEtqX9zWV+WVk+rrKvmaX9sYg4boDelyQNOvWeQd4FWBwR84GXexozc3JDopIkbay/efhI4KjMfCEihgM/i
P5sVLpkvFBmh5dHAkcBf1TaZ5f9XwmcVHmt7wGXR0SU9ZqWUtsH3ujLbqOeYMSBFttO

```
  },
  "metadata": {
   "needs_background": "light"
  },
  "output_type": "display_data"
 }
],
"source": [
 "categorical = file.drop(columns=['CreditScore', 'Age', 'Tenure', 'Balance', 'EstimatedSalary'])\n",
 "rows = int(np.ceil(categorical.shape[1] / 2)) - 1\n",
 "\n",
 "fig, axes = plt.subplots(nrows=rows, ncols=2, figsize=(10,6))\n",
 "axes = axes.flatten()\n",
 "\n",
 "#Generating subplots\n",
 "for row in range(rows):\n",
 "    cols = min(2, categorical.shape[1] - row*2)\n",
 "    for col in range(cols):\n",
 "        col_name = categorical.columns[2 * row + col]\n",
 "        ax = axes[row*2 + col]       \n",
 "\n",
 "        sns.countplot(data=categorical, x=col_name, hue=\"Exited\", ax=ax);\n",
 "        \n",
 "plt.tight_layout()"
]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "-OpbcL8MpdJv"
 },
 "source": [
  "##4.Performing Descriptive Statistics method for to explain the features on the dataset"
 ]
},
{
 "cell_type": "code",
 "execution_count": 38,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "_CR1_PwmoPbD",
  "outputId": "dd6329ab-3926-4aa5-c34e-3a9940224cf2"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "<class 'pandas.core.frame.DataFrame'>\n",
    "RangeIndex: 10000 entries, 0 to 9999\n",
    "Data columns (total 11 columns):\n",
    " #   Column           Non-Null Count  Dtype   \n",
    "---  ------           --------------  -----   \n",
    " 0   CreditScore      10000 non-null  int64   \n",
    " 1   Geography        10000 non-null  object  \n",
    " 2   Gender           10000 non-null  object  \n",
    " 3   Age              10000 non-null  int64   \n",
    " 4   Tenure           10000 non-null  int64   \n",
```

```
    " 5   Balance          10000 non-null  float64 \n",
    " 6   NumOfProducts    10000 non-null  int64   \n",
    " 7   HasCrCard        10000 non-null  category\n",
    " 8   IsActiveMember   10000 non-null  category\n",
    " 9   EstimatedSalary  10000 non-null  float64 \n",
    " 10  Exited           10000 non-null  category\n",
    "dtypes: category(3), float64(2), int64(4), object(2)\n",
    "memory usage: 654.8+ KB\n"
   ]
  }
 ],
 "source": [
  "file.info()\n",
  "#To print a concise summary of a DataFrame"
 ]
},
{
 "cell_type": "code",
 "execution_count": 39,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 300
  },
  "id": "XdviRVpep_ic",
  "outputId": "8ee8be91-488d-4f8d-8760-b9492808efea"
 },
 "outputs": [
  {
   "data": {
    "text/html": [
     "\n",
     "  <div id=\"df-ff051609-facc-4535-8af6-39e9921a8010\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <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>CreditScore</th>\n",
     "      <th>Age</th>\n",
     "      <th>Tenure</th>\n",
     "      <th>Balance</th>\n",
     "      <th>NumOfProducts</th>\n",
     "      <th>EstimatedSalary</th>\n",
     "    </tr>\n",
     "  </thead>\n",
```

```
"  <tbody>\n",
"    <tr>\n",
"      <th>count</th>\n",
"      <td>10000.000000</td>\n",
"      <td>10000.000000</td>\n",
"      <td>10000.000000</td>\n",
"      <td>10000.000000</td>\n",
"      <td>10000.000000</td>\n",
"      <td>10000.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>mean</th>\n",
"      <td>650.528800</td>\n",
"      <td>38.921800</td>\n",
"      <td>5.012800</td>\n",
"      <td>76485.889288</td>\n",
"      <td>1.530200</td>\n",
"      <td>100090.239881</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>std</th>\n",
"      <td>96.653299</td>\n",
"      <td>10.487806</td>\n",
"      <td>2.892174</td>\n",
"      <td>62397.405202</td>\n",
"      <td>0.581654</td>\n",
"      <td>57510.492818</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>min</th>\n",
"      <td>350.000000</td>\n",
"      <td>18.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>1.000000</td>\n",
"      <td>11.580000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25%</th>\n",
"      <td>584.000000</td>\n",
"      <td>32.000000</td>\n",
"      <td>3.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>1.000000</td>\n",
"      <td>51002.110000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50%</th>\n",
"      <td>652.000000</td>\n",
"      <td>37.000000</td>\n",
"      <td>5.000000</td>\n",
"      <td>97198.540000</td>\n",
"      <td>1.000000</td>\n",
"      <td>100193.915000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>75%</th>\n",
"      <td>718.000000</td>\n",
"      <td>44.000000</td>\n",
"      <td>7.000000</td>\n",
```

```
"      <td>127644.240000</td>\n",
"      <td>2.000000</td>\n",
"      <td>149388.247500</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>max</th>\n",
"      <td>850.000000</td>\n",
"      <td>92.000000</td>\n",
"      <td>10.000000</td>\n",
"      <td>250898.090000</td>\n",
"      <td>4.000000</td>\n",
"      <td>199992.480000</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-ff051609-facc-4535-8af6-39e9921a801
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.0
-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.0
.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47
  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
```

```
 "      background-color: #434B5C;\n",
 "      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
 "      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
 "      fill: #FFFFFF;\n",
 "    }\n",
 "  </style>\n",
 "\n",
 "      <script>\n",
 "        const buttonEl =\n",
 "          document.querySelector('#df-ff051609-facc-4535-8af6-39e9921a8010 button.colab-df-convert');\n",
 "        buttonEl.style.display =\n",
 "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
 "\n",
 "        async function convertToInteractive(key) {\n",
 "          const element = document.querySelector('#df-ff051609-facc-4535-8af6-39e9921a8010');\n",
 "          const dataTable =\n",
 "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
 "                                                     [key], {});\n",
 "          if (!dataTable) return;\n",
 "\n",
 "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
 "            '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table not
 "            + ' to learn more about interactive tables.';\n",
 "          element.innerHTML = '';\n",
 "          dataTable['output_type'] = 'display_data';\n",
 "          await google.colab.output.renderOutput(dataTable, element);\n",
 "          const docLink = document.createElement('div');\n",
 "          docLink.innerHTML = docLinkHtml;\n",
 "          element.appendChild(docLink);\n",
 "        }\n",
 "      </script>\n",
 "    </div>\n",
 "  </div>\n",
 "  "
],
"text/plain": [
 "       CreditScore          Age       Tenure        Balance  NumOfProducts  \\\n",
 "count  10000.000000  10000.000000  10000.000000   10000.000000   10000.000000   \n",
 "mean     650.528800     38.921800      5.012800   76485.889288       1.530200   \n",
 "std       96.653299     10.487806      2.892174   62397.405202       0.581654   \n",
 "min      350.000000     18.000000      0.000000       0.000000       1.000000   \n",
 "25%      584.000000     32.000000      3.000000       0.000000       1.000000   \n",
 "50%      652.000000     37.000000      5.000000   97198.540000       1.000000   \n",
 "75%      718.000000     44.000000      7.000000  127644.240000       2.000000   \n",
 "max      850.000000     92.000000     10.000000  250898.090000       4.000000   \n",
 "\n",
 "       EstimatedSalary  \n",
 "count     10000.000000  \n",
 "mean     100090.239881  \n",
 "std       57510.492818  \n",
 "min          11.580000  \n",
 "25%       51002.110000  \n",
 "50%      100193.915000  \n",
 "75%      149388.247500  \n",
 "max      199992.480000  "
]
},
"execution_count": 39,
"metadata": {},
"output_type": "execute_result"
```

```
 }
],
 "source": [
  "file.describe()\n",
  "#This function performs descriptive statistics that summarize the central tendency, dispersion and shape of a dat
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "Z2Y_9iJ9q1jY"
 },
 "source": [
  "##5. Handling and checking for any missing values"
 ]
},
{
 "cell_type": "code",
 "execution_count": 40,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "ZQTVSSYvqR0Z",
  "outputId": "71b2a6d3-622b-48c3-8526-c70af39afa5f"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "CreditScore       0\n",
     "Geography         0\n",
     "Gender            0\n",
     "Age               0\n",
     "Tenure            0\n",
     "Balance           0\n",
     "NumOfProducts     0\n",
     "HasCrCard         0\n",
     "IsActiveMember    0\n",
     "EstimatedSalary   0\n",
     "Exited            0\n",
     "dtype: int64"
    ]
   },
   "execution_count": 40,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "file.isnull().sum()"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "KhrH9kp8rAnt"
 },
 "source": [
  "The above result shows that there is no missing values in the dataset"
```

```
 ]
},
{
 "cell_type": "code",
 "execution_count": 42,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "lSS9rsBmq-Ml",
  "outputId": "0474b8e5-6aa6-47a9-beb9-23a8fae2d512"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "unique of Geography is 3 they are {'France', 'Germany', 'Spain'}\n",
    "unique of Gender is 2 they are {'Female', 'Male'}\n",
    "unique of HasCrCard is 2 they are {0, 1}\n",
    "unique of IsActiveMember is 2 they are {0, 1}\n",
    "unique of Exited is 2 they are {0, 1}\n"
   ]
  }
 ],
 "source": [
  "for i in file:\n",
  "  if file[i].dtype=='object' or file[i].dtype=='category':\n",
  "    print(\"unique of \"+i+\" is \"+str(len(set(file[i])))+\" they are \"+str(set(file[i])))"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "rZR4R3HoryMo"
 },
 "source": [
  "##6. Finding the outliers and replacing the outliers"
 ]
},
{
 "cell_type": "code",
 "execution_count": 43,
 "metadata": {
  "id": "QpwhzuyHrhQp"
 },
 "outputs": [],
 "source": [
  "def box_scatter(data, x, y):    \n",
  "    fig, (ax1, ax2) = plt.subplots(nrows=2, ncols=1, figsize=(16,6))\n",
  "    sns.boxplot(data=data, x=x, ax=ax1)\n",
  "    sns.scatterplot(data=data, x=x,y=y,ax=ax2)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 44,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
```

"height": 447
},
"id": "HNuepepesfU_",
"outputId": "155877db-97ea-408c-d00b-9aa5dc187f82"
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"# of Bivariate Outliers: 19\n"
]
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzdeZgl510f+u/b6+xaRqMFy/KYO/ImGWwzGPKQOEZerjCKRB7ASxJslsQ
kePdNVp+qtd6/qn6rPW2qtAQAAAKC7xtY7AwAAAAB8dQR4AAAAADpOgAcAAACg4wR4AAAAADpOgAcAAACg4ybWOw
gAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg
eEwJ8AAAnbF37958/NN3ZmHTheudFR5nxo8eSJLcP+Px9lw3fvTh9c4CwLpwBwQAOmVh04U59rSXrHc2eJzZ+Nl3J4m+C
AAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAA
APAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8LAu3vSmN+VNb3rTemcDAACAc8C58DvoxHpngH
CPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAA
PAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPOvgXe96V57//Ofn7W9/e2688cbs37//tM7fv39/brzxxuzdu3fZ85v
rfmU5/6VG699dZlz2+Oaz5faXu11x08/tZbb82RI0dy6623nlYaDzzwwKqOBwAAAFYmwPMYe9e73pVaa3+71pr3vOc9p/U
MT652Bc03z9s6ghYWFvO1tb8uP//iPr3j+W9/61iwuLq54/uBxCwsLufXWW5fdXun67fTe9ra35ROf+MRJx9x666359V//9V
yZ9++feudna/K4Ns7jfn5+bzvfe9b1fnvf//7Mz8/v+L5g8fNz8/n7rvvXnZ7peu303vf+97Xf3un0d5eKe/tcgAAAABnphNv8NI
b6f3ohe9KJ/4xCdOCurs3LnztPK+ffv2vPGNb1xVuQEgSV73utfl9rt8jxsw2uKGbdn19Zd4zgROci681deJN3jOJj/2Yz92yr7x8
wcAAABWR4DnMXbDDTeklNLfLqXk2muvzfbt21d1/vbt23PttdemlJKdO3eOPH/wuGuvvTa7du1adnul67fT2759e3bt2tV/
rPV9pe7XUHj3/DG96QzZs3r/j2TjsNb+8AAADA104Z9qW/j2e7d++ue/bsWe9s8FVq/v7R30YDcDqa7+A59rSXrHdWeJzZ+N
AxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnw
xwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHTcxHpngHPTrl271jsLAAAAnCPOhd9BBXhYF6997
AAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAH
AAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHTcxHpnAADgdIwffTgbP/vu9c4GjzPjR/cnib5Bxo8+nOSS9c
ADwAAAEDHCfAAAAAAdJwADwAAAEDHCfAAAAAAdJwADwAAAEDHCfAAAAAAdJwADwAAAEDHCfAAAAAAdJwADwAAA
WAtPqrXuaO/sXIAHvtZKKXtqrbvXOx/weGNswHDGBgxnbMCpjAseS/5ECwAAAKDjBHgAAAAAOk6AB5K3rHcG4HHK2ID
73tvb3Pd65nvmEtlVLuLqV8qpTy8VLKnt6+C0sp7yulfKH37wW9/aWU8ou9sfHJUspz1jf3sHZKKeeXUn63lPLZUsqdpZS/Z
9Qkkd6+/9j7zg4m317rfVZtdbdve3XJ/lArfXKJB/obSfJdyS5svffq5P8p8c8p/DYeWOS99Ran5bkG7N0/zA2OKfVWj/Xu188K
62+l9/oLe8XCuGBwD7bHxtrrkI0nOL6Vcth4ZhLVUSjkvyfOS/JckqbXO1IofjbEBg16Q5C9rrV+KsQFJMpFkYylIlsmmJF+J3zd
k1zS+7k/NnoGxw2cTZ6cZF+S/9r7095fLaVsjrEBg16e5O29n40Nzmm11vuS/Psk92QpsHMgye3x+wbrQICHs1op5bokD9
/W+d+qGLP0Pgq9LsjnJteuaKc5ZAjyc7b4tyfWllLuT/HaWXpV8Y5ZeE57oHXN5kvt6P9+X5IlJ0vv8vCT7H8sMw2Ol93+cUm
tqrXNJ/t8s/Q7i9w0ecwI8nNVqrT9da7281rozS68Tf7DW+neTfCjJ9/QOe1WSd/V+vq23nd7nH+z93yg4q5RSNpdStjY/J3lxk
7Bo+5oi9xriiIPD/JT9ZaryulfH2W3ui5MMnHkvy9WutMKWVDkt9I8uwkDyd5ea31rvXKM6yV3hj4vd7mRJL/Vmv9V6WU7U
+9w3OOeVUm5O8rIk81n63eLvZ+m7dvy+wWNKgAcAAACg4/yJFgAAAEDHCfAAAAAAdJwADwAAAEDHCfAAAAAAdJwAD
ADO3MR6ZwAA4EyVUkqS30vy1lrry3v7vjHJJUk+39ueqLXOn27atda/P7D5z5L86156fy3JdUmeU2udKaVclGTqqyzHGeUR
nlw6WU3aWU/yfJxt6bOr+V5LIkD9VaZ3rXe6jW+le9c765lPInpZRPlFL+vJSytZSyoZTyX0spnyqlfKyU8u29Y7+/lHJbKeWDST
y1CTPyNJbP59J8muDidRaX19K+dFa67OSpJSyJcnPlFI+n+T9Sd5Ra/3/SilTSd6R5GW11o+WUrYlOZbkdUvJ1GeWUp6W5I8
yfOSvL3WupDkr3pv0yyr1nq4lPJNSf5Glt4gekcp5fVZCjR9pdb60d5xB5OklPLXk7ypt++zpZQvJWkCPO+rtT48kMfrSyk/2dv
99da/3cmeQJADi3+Q4eAKDLPphkuvcnWEmSUso3ZOntmkHvTfKPSimTvWOeUkrZnOSPk7ys9x09l2XpjZxh5gbOfWop5
TyU0mOJ7k7ye+3Dv3VJDuT/EUvgLIvyXdlaQWua7L03Tv3JPnTEZd6S5JPlIL+IsnPJ3lT73ty5pPsTfLqWutsKeVlvc82Zun7d1
QcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8
AAAAAHTcmgZ4SinXllI+V0rZW0p5/ZDPp0sp7+h9/mellJ1rmR8AAACAs9GafQdPKWU8yZuTvCjJvUk+Wkq5rdb6mYHDfijJ
2vNWCk5MjufzVMTmZ1fzIbJ8Rybn8vU+ET2HZrJjq3TuWDTeB45upAHD83k4q1L522YnEgpNbPzNfsOzWT7IqkcmZ3Plq
sZK+M5eHwu2zZM9v999OhsLtg0lW0bx3Pg2EIePTaX8zdOZv/hmWzfMt0vz/mbxnNstubIzEI2T08kqZmZX+yXq5TF1DrW
4Zj5HZxf6ZT9wbLF/7OGZuWwZKPOjR+dy/qbJ/v6mbo7Nzmfj1ES/jpo8tOus2d8/r5f383p1NjleM7dQBsq+lNcmP8lCkhN
4Nhszts4lUeOzuWCTZM5b+NYDhxb7Oen6RuP9PrMQl3I+ECfaveFrRvGcuj4Yv86x+fns2FioI/vs/PzmZqY6NdLMzaadJq+
JLxPHR4Nls2TGTbxomkLmZmPv08HJmdy+apE32pGb/NNZo+ODM/n+mJiTx6dDbnbzrRNk2d7D8yk+2bT+S9aZv2OG/+
O3zfnHe/NJc9ymqeTobPrlaOqr6Svt9E+Ue6m+TvT5pTF9dHY+m6Ym+vNg0yf6faU11pr2a/pMU/6m3E39tLeb/DT52ziZH
4aMrajMOmLZuyPHx0Nhdumurvb+aZpoz9PtBLr6mrpu81123arOl7TTrNfNfUUdOmTZs36W7dOJZDx07MLymLST3RR
p0/MUe3rNWOjab+mPzTlb7dHU+5LzhvPAwcWTqnH5ryDx2azbeNUDs/OZcfm6RydXXrOuHjbdOYXFzIxcE9rj+/mnt1cq

+T7xYbJ5Phc+vlu2uTw7Fy2TJ1IrxlLTXpN/TXHt+eIdr6ae2NTL005m/pv6vXA8bmct+HEM1jzbz/dXp9r9jfXb7abcjTla8Zmc
nL9jxw4pIgqS+d6GMnP5f0+0KvTg4cnc15m6b6+WqPmaavN2U6MR8s1cGJsbFU58080MwTzbzRlGPU/N6k0+7LI2wbzw
zcMx7e8f8aSlIlsn9SXbUZTK1e/fu6kuWeawsLta854778xPv/HiOzy1mw+RYfv6lz8qLn35J/ujOB07af8sNV+edH/1Srnnapf
lz+8dFxz3n//i3vy0t1Pyjv3nHq9W66/Km8eOP6W66/KO/fckz1fOnBSui/bfUU+/uX9ecHTL8svf3hvXrb7iv7+dnrv3HPPKde
m685sp+fm9qpff+O7+Sb/u/Ls7RuYW88QMnzvnZG67OL33oC/26GKzT3+mVqSnzqH+HteVgGQfT/YfP29VrmysyPz+biY
e7HDdff1W2TNUcmS0npdv0pW9/6mVD02v36RuvuTJzc7O57IItp/Stg0ePZeumjf3yvfgZF51UX4NjqMn/YHnf/OHhfa89F
+/bLdV+Thw0fzlEvPP6Wu3zxkXvre3VfkA3cutWW7DzZtPLi/Gb/PeuL2kW2+0vz4gd75pzN/DZsnP3//o7lwy6ahbf+a5+/K+
rMwrpdX0jWuedmk++Nn789LdV5wyf4yab9vz6rB72Wuevyt77n4o37TzolPy1dTZ4HgfvM5yY2RwLA321T/6zEMn3csG5/
MvfekYWUnLzHwy/I4661w57jrrmaZdmamwh2zZtPKkN2/fI1YyX9j1qWFsP5qMZS8s9f41KZ7BuRs17g/lpI39UuoP1stI9e
zUfi4Z9Xw2rM/cfvfJ96phzxCDdTwsv8vVYXt+H/UssFJ9tX+XGJbPH3n+rvzNK7fIj79wsDWWrs6Lr97R2SBPKeX2WuvuU/av
Ib/nIKfv/7fd8Y/7p735i1ft/6K9/fd78ob15zbfvyn/533cN/fy//O+7+se101sp3cHjb3z7x05J91e+75vyw79x+0nXGZaPUddpz
dTOqbVZKv10H7Tperi+dTl8bdVxz/Xb9rbae2vXSTndUeu12fOsPPDevGpJOO/23/uA3D+0zg/kfLO9y119N/Yzqb7/+A8/N
wd//aOnPYZH9YeVyn+m9fvWH3huHjg0c1IZVirzqLTa96xR8+9K8+qoe1m7Ttt1Nmq+Ot1776983zflVb/20f52e75ZKb2V-
XacrmxvlJ5V/p8NeVfbsydbjlXOn9Yv7jxBbuGPpeOmsube9Bq6mhwnJzu89Nqy76a+WA188xqn5tH3atWqq+V0m/+HfX
B4/rpzc6vKt3B44el++iRuVOuMzT/s8Ov05zfaOdnVHoPHDo+NP8r5eOR1vX6+Tg6l8U6ug6Hlb0pU/uaK9bFMnU+mG5T
Z1QZR80Lp5PHYX121HGj8j5qXho5TkfsH3X8asuxUt9ZrnyD28uVc1g+V5p/R33+0OGZMxrDo/rDSuU/0/p98NDxU8pwp
y6n3A+HHHdkhXNP57nodNtwNZ8vl/dhbX2mz1+n27arzd/p1t9KfXalZ63TLdeoOWRYfleTvxXLucrzh+Vj1HPPpqLm8uQe
dv6DG9P9E6L8n+dkK11rfUWnfXWnfv2LFjjbILp7pk24ZsmDx5mGyYHMtl520cun/T9MRp7R98gW7U58OO2zg1cVrpbpy
MOq/dNqPqelibtetsue0z7Wvt45rrD5bndPrMqDE1rC8ul/9Ltg5Pp71/VJ9p2mlYeb+a+hnV3y4ekd+xkpHnjKqrUXk5nTyC
vax/zxpy71rNuGvOH3Uv27F1eJ0288FK89Wo67bLdf6myZPTHzFGR6V3pn3mTOerYfe5Yddttkf1zeXmltPJR1O+Vd8Phxy
qU6GNy+8DSfn1bbxttXMR+spi1W+9w86llg1Bi9eMj8vlw+R6Wz9N1IZ5exlQ85Yx9NcmUp5cmlIKkkL09yW+uY25K8qvfz
X+8JNLMc8/+MR9ueX6q0457w8/eV9uvv7Ecc15b/uTu3LL9cOvd0vr+FuuXzq+ne6N11yZ3/rIF3Pz9VflDz6xtN38205v2HV
m8roXnHzOz95w9Ul1MVinTZnaZV+uLpq2bOehSfdE2yyVsV037bq4udU2o+p6MP12nbfruKnD5fraqPwP9p1RfbK5ftN2
+dkbrj7p+Ne94Mrs2rFl6Dk3XXdV/vQLp5Zx8PNh57XbsN0XBsd/8+9yffrGa67M++4YPo6HzUs3X39VfrPXlqPaeNj4Xa7N
aOHlXmltJq+0fS1YfPHcuNucF4ddi+75fqlOh2Wrz/t1dmo+Wq5MTKsj/zWR754OnZ7nI+urw6Wf1T9jeozTfnbY3u5eht89lj
XVPHMMu177GWa5/LTLPyrdpm1Wcw9eri3bY35YX1muvKP67HLprtT3hl2n3c7D+sIqn+uH9YuLtkyPnJeGXft9d9y3Yp9
p5pxt1uw7eJKklPKSJL+QZDzJr9Va/1Up5ZYke2qtt5VSNiT5jszCStxqvGr7jjSrNZSs5ix3koq0xMTp6xC015
krq0CsIFm6cyMVYzv4pVtNorSh06PpetQ1bROm/TeB49utA/b9QqWkdnF7JpanzgzzqGr7jSrNZSs5ix3koq0xMTp6xC015
mtmV7GKVrsso1bRavrQSqtoNauU9FeM6OWrKUezylU7/1umSw4vs4pWszpDexWt9qo07ZXZmhXS2qtoNX2iWf2hyWe
f04fYqWv2Vf3p18vCRmVw4sIpWc1zTJs14b/5tr6LVXGfUKlrN6itNXTRt1tR9e5w22+1VptqraD3cXw1w+CpP7ZXfRq2i1fT1
18PGcv/HECjjtcdyUqbnnNNvtlcPaK+g0q7X0+3BrFa3+aIb9/UvzXtNm7VWamrI3bXHqKlpLfaRJp73qVLO/3+a9dJtVtJpyN
b+5p8tH0h/aqgk167VW0TqxS2F5Fa+kZ5ujsXC7cNJ1jc4vZd3ipzzUrEDV9o32vbd+z23XSX8W0d+32ymbN+GzqeNQqWs
YpW80xyykqlrb7bfgZrr6LV9LImf3P99ipaTfmasdlcd6VVtJoVRpu+1l9Fq3Xfavp6M4eNWkXr8MzSdRcWc8p4b67V9NFmu
cqaXP99ipazTzVzE/NcWfbKlqP+ZcsrxUBHgAAAOBc1ekvWQYAAABgNAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAAAI
AgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI5b0wBPK
TfJv1yo/AAAAAGertXyD57lJ9tZa76q1zib57SQ3DB5Qa/1QrfVob/MjSS5fw/wAAAAAnJXWMsDzhCRfHti+t7dvlB9K8+HfVE
74kTxzYvry37ySlIBcm+edJ/matdWYN8wMAAABwVlrLN3g+muTKUsqTSylTSV7TSV6e5LbBA0opz7yK0mur7U+uIZ5AQAAAD
mFa3l9AAAAgHPB4+JLIgEAAAA4cwI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQ
AAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB23pgGeUsq1pZTlVL2lIJev/
SX4uyfE1zAsAAADAWWstAzxPSPPSLIge17e/v6z2yEw/vN/fKJbq7u1ZpDa
yk/sdzntdafX+bjjya5spTy5CwFdl6e5O8MnHsgyUUD1/pwkp+ste5ZOdsAAAAANJYN8GTpLZskeWqSb5w8O0smhGfF
2zx0s+AAAAwLmnlHJ7rfWsDTNXKeXVtdZH/z/fZ7ZG/9a9+GfX/nzIu5bWmjLSA5bUAjB4bdWPtWZk8TsLTjXQ/
AAGD1VvsGz+ZHtCHm7PTRo+ls1A3LXGN+Jh6k/Y6HZTYyWNM1yK2p9
tcwlAAAAAGdm2Td4D9XeVttkW4zO5J9a4g3x+9Kz9dmjlAAAACA07+KSIBSQ7cy3SCpOI9MiOxTfwl6K6dHGWgR7UX
aJVSvmh1vZZVzlLpE3Z4KeWqmm5D8kYv8iZGdPV7Dv7ccf6h+OcF77Kq3brtZpxDZTrzRtvYbzpKvDvyCZSQ1UbNY2tdk
l8t9Z6MEIqrTXJfyil/MHZyil/MHaZQsAAACA1Vr2DZ5zrL3Hu8ag8rFLsrXxxUUWFzvzR4XWBdKmRtBb6F1qI9pFwL
i52Hbbc9NsrfWeletdTbJbye5oXMP0jy5lrIrI0lSa31xkbAAAAQWoPw1r3GBUBpWVCs5lRLe5qpJDVZQRiqq22j
QbXWtyR5SSLs3r17pcASAAAwDlI2QBPPrXX8j5qXPHN+vLdj0ohJ/qz+QS8mcb8lL1V/qzWOpfkYmbqaSR1y/l
ntvkv2llM8k+VCSf1Jr3b3WeVXQWv3Ffki/9a8vp5jbcmvOFl/kz0bXBeBvn2e8pT5gmQOIZ1JXnV4dF8
AAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AE
rSinPaB32hiTvrLU+O8nLk/zyWuUHHAAAA4Gy1Im/wPDfJ3lrrXbXWuSS3Jnnl4AG11t+rtR5pbn4kyeVrmB8AAACAs9JaBnieKeKSPL
Wutbaq27a627d+zY8ZhnEgAAAODxbC0DPD9PCeleLA9uW9+fN+KMK6k6k6TW+qdJNiKo3K3jAAAAAHDWWcsAz0mU0z8
b6/1IrXXKk8AAAAAZ6OJtUy81TUy81vruLH158uC+nxn4+TNJNAAAABwtlvzkAGKkswkwAAMAAADQcQI8AAAAAB0nwA
I8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAA
zxsbKiucvLtbcvf9IHjh4PJumJjK7sJDtm6dPOX/wuEu2bgvjLizV3fOVAvnLgeC47b2Ouumxb
2zi16rZsp/d1Wzfk0/cfzP0Hj+eSrdOZmhjL/ELNgeOz2b55Q55y0ebc8cChZPl
Zke2g4XbZrKw8fm8uDB49m+ZToz8/PZODmZ2fnFXHreqW0yP7+YO75yII8enc3GqYkcPD6XbRsm8/CR2ezYMp3NG8Zy
7GyTxydDYXbJrKWFlMrWP982YX5jM1PpFHj87lws1T/TY4fHw+UxNjeeTYXLZvmsrTL9mSzzzw+KQ2u2jLVEpZzGId61/nW
xhTx4aCYXb53O+FjJ+NhYJsaTIzMLefDgTC7eNp25hYVMjo/n4LG5bNs42U9nYxq9Ok0xw/WF+Ldaw/j9YyspmQsF
bZrq72+uu23jWA4eW+yXq2YhJeN5+OhsLtw0tdsszPG5hew3k3n/E3jeeToQr+PNGlNT9TM

NW/TrvOm/SbdNp115w/Mz+f6YmJHDg2m/M2TuXRY3M5f+NSH5scn8hDvT7U5Ku5XlOOY7Nz2Th1onxHZueyeWqyP6
77/PjcfDZMTuTo7Hw2TZ2or6bczZjt9+n+2F0qT3PcYl3I2EC6TfmbftDU94HjczIvw2Qme/P0g725rmapn+/YOt3vm+dtHM
YI5umkqOz6R/XbJ/ou0vjtLnO1g0lh47XU/LTtEEzFps+1ZzXHjtNn2jSabab6zZ9r2mLZj5rtvcdnsmOLdM5PDOXLdOTp5zf
pT+Wetc5MjOXzdOTp/TtifGa1LEcmpnPsbmFXLxladxMjJ+Ye5v5qbl2vy56ZWvadtR81cw3GyaT43Mn+s7WjWM5dGyxP
Plv2rRpi+b6zXz2tEs357P3HzIxby81C7X056FmPmvy1Zzfvvc2bdt83vTxpv6bZ4Jmnmry31x36/R4Ds0snJjve2OvSa+5n/Tv
t5+MhsLjtvQ0pZeoZvnu33H5nJlukTz0/bt4znocMLJ34/HC+ZWagn5sHJ8RybO/E8+oxLtubzDx3O4ZnZTI6fGHePtp4Pm2f
u3Za/OnS8/zvPzPxSHT9ydDbnbTzRtvt6bTs9OZYt05P9Z/vmufro3Fw2Tk5mcbFmbKzk0aOzS8+NrXI+cPB4Lu79bjO3UP
JG+stX7LSunu3r277tmz52ud3cfEsWNz+YNP35+fue3TOT63mA2TY7n5+qvygTu/ku969hW59qpLVwyyvOeO+/MT7/x4//
vEJQ39ZXynvL376JfmjOx846fPl8nQ6dTSYzkp10xzf5Ofn3nNnXrb7ivziB79wUjnf9MEv5Ev7j/XL8cHP3p+XP/dJI+u0ue6w
d8Kd/9nP+/vXMPsquq0vi3+p103ulO0jySgOkE7QaS0ESsUSoExvIRO0yJBErlaUVHNIxTlqNOkyMoiB
v17RizY7DWfM2v2lIju+1tXXhked3ZY3Y3T0tyAFWacuN62n8/PqQ/6TUdHNx7e+Aa+9pu/eM+L6/2FK6ZiSHkJvvXfL2XaLprb
qyI53X4mjR2Cz86egkWPbMqywfb9R9R9A0udY7Z1buOedNyBlv0tghuGX2FHztkU1Bua1Pu/0unFOf8ZU7V72SVw9r//Wt+
7evfAVTxw3Dte+ehMWPxmVoxMu7D+Id40blyPby7kOonzAqa+4XzW3Ar9b1zu0ts6dkZPXpaG3n6hKPDZ+PxW3u+kw+
m+6vbbNGkkPtY0MWd+n9oc5YxC5LF/P3PpFDy1eRcuf2eddz598rpyVpWX4NbL67H0jztw8ERHRqYr3lmX4wM2v7m+6
8EvKReD4KyZkvZuJ2sDH359cOeO2QJoYWNzdg98FjqBs9zLsehGLajf3r3jMZu4+047YnXs60+ca8RtxhrpdCtr/L8cm9hyN
S9I73n88z/t89pbZU4L2X76mFZ+dPSUzb2n08+XPbN8f7b12Ca1HvhiyuSN0jWTXh/KyEnzl189njn39Iw1Yt2M/ms6pxZ1F
d9x4tc83vFjeobyqW23Zvu+HB9vaW7E+xtri/Ymj4isVdWmnP2n6gaPGXQygMcCN3h+BGCVqi4z25sBzFbVXfn6LOYbPM9
AAFa9JREFUf9ga3Lb9hMYPJbvbg0nNx+5NbsvYtX3AJLjx7dMGyL19wCebf80xqmQq1UVrbxPWYf88zGfl8ct/51JbM9revuh
C57l6+3zSbWNl3fDawaw5Ds2Nq/vPbn43miaPSe1LVvekcaycaeP1/hsuxk0/eS7vnH3pwQ0546XtP9Svq0+SL1o5k+ya1g
V9+QD4Z8zvbb15hwYzifXX3H3X5D/uDmy7Ty2L+hfNtX/8iXw9OuSUnxHZqbtHNa6Fy6a1/lJm67kHxJcoT0SJlvdH6SD4fn
80nae1g5UjyAbffQq+PQz6TNOe2v7T6heydlGPSzkdSLrP9+65HQ/HrjpU2npLWSuurlV3yreHxtT9pjpLWB/f8pOu4QuYm
092Hu0LeH8Nu/5Isg6321nj4e2bT+h8UPj9jj3Bts6e7D7cGF9WFl2HfYfT7JJ2nHS2sa2t/KEjotkb59s70o1btJ4SeMcOt6Zqp
/1p44Z77x0vZvz0/SJ6SH3T5wzO8Trl3T+mDouB1nz1G/zyTZKe3+nNj3xAYA73yHZNgXkDm0351b229Qx8D+kM+5uoW0
LztrX8gmbru0sZBWj1C7kFw2zxXis/nkzMh1NJyn439DdijkOjNNTvPpYdv1aGFraMgWexN8LiTroROdWeOEzrdrUaHx6M
PUh7doX9MEEW7sxGWoXynN7jrRjoFEUL1lW1XtUtUlVm2pra/tbZ/VV6CccPzPx42fkSV93xVZ
+zx0N6DKkoy9q28+b63oSQj+eRO5/f1CX4jau3zyfdNlbWul3yzY2r+/gRhflSXPdCfCfqp39phlXnnbGhFWd7xkrbj5yfpk0+P
om+DbtQGfi9uzLzHh5r18dk1jh9Bxm5cLlcf+HR3It4XGgPWPfH2mXZNGJcR30tykmdN8tsqR21n73OM2xnzt3oocNQE93
L23e9a1h+cYJ5Ye0/ae5FknKb4XoF7J3bcJ8p50PmxOS8r/vejQeZ33x4Xw6+WRwfTVtPvCt/fnGCX7HCOSL4LVQyAcTbO2L
qnHYxt3Zp3vtnt0w05888rG4HZVeUne8X3j/sfHpqN+3LCsfd+8shENdSP7JHtD3cic42lskmacQmxj21t5Ht2wEwvn1Ofo+
+dXzcyZ5yW2Di++c/nN+efMRIt8xqD58X1/sIVU1FTXZHVdtHchhzbWFkb6kbgm1c2ZubENzctju4t8xpxwRkjg77k2s7qvm
bEn1wyeqtOf21NPeOs7F1f87xkD5xO6XdH9fP9Y/FnnmM2+k/n34Viz7iytCIFZv8vvPEpty5XzQ3GtPGQVzWkO18usRjw+
FCJP3D4PPLPNq39IXl9M3Xp5PX697vUsmXw+4MoaygMP5InvRXOjuWnxzI3t76F1ralixJfffPkne+3zx1hc7p8Zm/rks76VZ
F0vhWzv+qSVJTSmT1br83EdQz77+oEjWb5TaF5O8n039nz5xpefXTssNXnI5wO9+aExZ27TXAPE83zoWiTfnMfnLY1+i+ZC
nX6LW5Hd8X1761z/qg+x0nnn+S9AzlU9vO5+MtzY2YOqGw75rFQH++g+fDAD6H3pcs366qs5L6LOZ38AC5VbSqyqO3hv
tXj2SbBrqL6R/vIpWzbBKlJUAI2JVtNLKb/e7VbQqy0vQ0aU42taBMdWVmFozDJv2HMXx9s5M9ZiaYZWoLBP0aFQFqaKsJF
rQkjqnBBoIrWniNtKC8tQXtXV1ZFoWAVLVO5xlbssFW0bGUHW3Hj0lIOjK6u6J2Dti6Ul5Xg0MlOjB5agXeZ6gHxORtbHa6i
OoucXulqqJlqtJYfWyFIqu3rdTUq0dUXStYRcupDmSrClm9eqvzmKoypkJJporWsXacNpU0TJV6kabKlq2monty62i5VZhylT
l7q0xFVVZsVRY7124VLVuBxNrBHrfyWf3cSnJudRSb9zK+aiq+uVW03GpmtopWry9HVV5sNSzX7lY/K487r9ZHbezY/ZnqX
cbevJkcetotWbh6LzrG/YOcxU0cpUEcyuqmV9z/qijQ23cpH1NetjbgUfmwPcKlp2znOqaDkV9aw9MznFxGZSFS07HtADoG
zFTRMnnIraJlbZtTRctcC7h52fp+ThUts9Zmqmg5MTGsUnAsTRUtxw52Tu1c2PFt/rFVtKz8SVW07La79lo72ePWx639bftM
qtoTRhRhVJzDZ9TRet4O2qqK1EzrBT7jvVWG7VVtHrzYClOdvRejzZkqmh1ZlUZPXSyE2Ni14f2mn9MdSkOHI/1b6pNdfd0
5VlIjg8MmOTMXKuJ62ipb9bnPlVOuKqmhVYlrt8AFZReu0v2RZRJYBmA2gBsAeAlsAlAOAqt5tyqTfAeDAiDiAMqk36iqiXduiv0
QQgghBBCCCFFzil9yfKpQET2AdjR33KAUP39LQhQhf4UwNgwxw9ggxA9jg5BcGBfkVDBJVWvdnUV3g4eQtxsRWeN
fhgbhPhhbBCSC+OCnDb4Dh5CCCGEEIIIYSQIodP8BBBCCCGEEIIIYQUObzBQ0obzBQwghhBBCCCGEEFLk8AYPGRSISKmIrBeR
4eIjBKRB0XXkJRF5UUTew9gggw9ggggg8oAqaHoB4AeUoNXODEEBEvB8AvBmlBr2R4
EGNaq62awX0wFcBOAEgIfA2CCDCDHBE5E8BCAE2q2gigFMA14PcN0g/wBg8Z8IjWQA+DOBesy0A5gB40AzRbe2
u8PsG6Qd4g4cMBr4H4EswH4EsAesz2WACHVLXLbL8O4Ezz+UwArwGA2X/IlnP2x66ZaUMIIYQQQgghhBQ5vMFDihoR+T
8lfIe1V1JqLH6G8RkUvjB1VVEd0mQZQfR5MgFR+5MTIAwMfR+5MTAIwNMrgx7xFpBvBL9XhxjGzHun5iH6D4IzAFQD+
hF683evaZ6JDUM8bggZSLwO4HeAiAF+9VOB2hQBDR/GBiEhhRDbReW4HVV/ZPZfhBiERHwSwTlX3F3AFgm6oSKmdZ4iM
QTMmBQ1d0AXhORaDaWbXUGZBUiEVcGuG/y+QU47QlNPq4ZENRmQ3gl6ovA6BDVvRNsZowrw0tNmDaNSvBuWH3Xw8tYkKzX
RCTnliMh0RC/mrwCwFcCNiP5TjLFBBBBjXmPwRaAZyrqofNPq4bZNAjIosBZCVcjgNcQvLxqyCRFADzAQ+aghm8Qk
CCGEEEIIaTI4Q0eQgghBBQ1IjBRH4uIq+KyKWEAYtFAtoTtLx3x5MkEOEvGKKzKOI/FlE3v3WNSGEEEIIKSnGj+PgFCdh6AIFQgf
JOMhBBCCCEWPsFDCCGEEEkGLmMgCdqnq33aGGwGECIRqjReQRAC+EZGff/ISfhjTf8Wk/VKvEEIIIKSnhf9cmnqPCCEMhjhqzFQ
85b72IzDs9JiSEEELIQBP8BBBCCCGkmDwZwBAwarlXdJiILLABxW1YtFpBLA/4nI8U9U9csi8jDtJv8ecsi8jl\
ERgF4VkRWqOrxt8dUhBBBCCBnI8AYPIYQQQgYqz6rqNvP5/QAusiLQ/QAusO/XATASQD2ASwEuAG+Yp2nyoqrHROQOQiAO9D9
AK4KHIs/+SIAPq+qv4s3EJEP9WVQc0NoFYBVIvI8gOSRfDH66MH1XVzX2RiRBCCCGDG76DhxxCCCHxCCHFzEoAleYnWAAAE
b6bOL8D8Hnz4miIyIy0RiCEEEII4RM8hBBCBDF5VVhBBCCCClaVFVF5O8AfE9E/glAG4Gyj4AVyKqwDUH0bt3WgH
GEEEISEVXtbxkIIYQQQQgghhBBCyFuAP9EihBBCCCGEEIIKXJ4g4cQQgghhBBCCCCGEEFLk8AYPIYQQQgghhBBCCCGEEFLk
ASUVORK5CYII=\n",
"text/plain": [

"<Figure size 1152x432 with 2 Axes>"
]
},
"metadata": {
"needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
"#Scatter and box plot\n",
"box_scatter(file,'CreditScore','Exited');\n",
"plt.tight_layout()\n",
"print(f\"# of Bivariate Outliers: {len(file.loc[file['CreditScore'] < 400])}\")"
]
},
{
"cell_type": "markdown",
"metadata": {
"id": "xAxltNBNuCE4"
},
"source": [
"#19 outliers in the analysis view"
]
},
{
"cell_type": "code",
"execution_count": 45,
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 447
},
"id": "qepopdsysppL",
"outputId": "648c947a-5658-43d6-bab9-5cfd7df0bfce"
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"# of Bivariate Outliers: 3\n"
]
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3dfbRdZX0v+u+PBEh4SVCIJAI1evClEBQl12q1NvWlgi8Eq5dKj63t8NQ7
t/PmMwsudc67ef3177GWvN/eWZc1ZrLQAAAAB01yHDbgAAAACAR0bAAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAA
AAKDjBDwAAAAAHSfgAQAAAOg4AQ8AAABAxwl4AAAAAADpOwAMAAADQcQIeAAAAgI4T8AAAAAB0nIAHAAAAoOMEPA
AAADI+ABBmrjxo259oab8+ARjx52K0O34N6tSZLN93nrnUsL7v3BsFsAAICB81cGMHAPHvHobH/KS4bdxtAt/sYVSeK1m
B0n4AEAAADoOAEPAAAAQMcJeAAAAAA6TsADAAAA0HECHgAAAICOE/AAAAAAdJyABwAAAKDjBDwAAAAAHSfgAQAAA
gAAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8Q/De9743733ve4fdBgBAJzh2AoD9WzjsBuajjjRs3D
A6DgBDwAAAEDHCXgAAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8AAAAAB0n4AEAAADoOAEPA
wAMAAADQcQIeAAAAgI4T8AAAAAB0nIAHAICxs2bNmt3/DbruvvPPOy5o1a/Ka17xm4GP1U3Puuz3dmzZo1Oe+88w6ov7
FmzJn/4h38465q3ve1tWbNmTd7xjnjfMumb9+vVZs2ZN3vWudx1Qf5dddlWrFmTT3/607Ou+eIXv5g1a9bk6quvPqCxN
23DrrmT6d6d1xxxx1JHupztrZu3Zok+eEPPfzjrjmn5fi/vuuuy9Jsm PHjlnXTPZ1oH+EXnfddUmSr371q7Ou6fnfn+uxnP5skBxSg/MM
99fngBz840JoLL7ww99xzTy688MKB9zeXRr2/fvX7+++oSAQ8AAGNl79Uts13h0juV2Xg2lVUvV5xFFrVsVBB85uiWGUe5bMd
bMOQyy67LK21JErbVYh1Be/+MXs3LkzSbbzZ5Zr6zsGFD7r777iTJJ3XffPlatKK69nbK59vFZTlmo2b
mzLxo13ec+FDtu4cWMWL1487DamtfGqbXdFqkxJcGNAzHzVazHTAN+uBamDsAW0ePZ4Ml...

JzT8+Zpy7f5xtIP2Pdf/+D+dR13887Lnuo5oK1q3LOUx+7z5DnR9t35PM3bMk7Lp9Sd/aq/OyqZTMeFN27/f585obbH1bzs
wupeuesyMwcb27Q/k0zdsfljNy1ctnzHk6ecYKOnvOHLHjp25/PpNDxvr7NNW7DPk6ee16Pf4uJ+6fsdivJkXjKqBnaLVWvt
131/6+5wZ7LmHZfdkOu+P3NNknxr8z27P5R3111+Q761eeYeb9h817Q1N2y+66COM9dj9VM36mONen/jOtao9zeuY41
6nrdyzGm3nBqBrmNXhOSPK9Kdu39vY9TFW9vqo2VNWGLVu2zElzg7Bp647dbwKTdjywK5u37pix5vZt09fccdfMNf2OtX
OqExdZbq1d0lpb3VpbvWzZsmG307cVSxdn0aF7vuSLDj0ky5fOvIT3+CWLpq15zNH7Pre7n7FWzDDW8Uv2PdbxSw6foW
E3r6xdeqKJbnwnFW73wwmz2c+dcXSGWtWHntkLj739D1qLj739Kw89siDPtZpj12aC9buWXPB2lV56mNnrkmSJy0/Mhed
UdPWnbr8qBlrTlu+ZNqa05YvmbGmn2OgpM/jyBVLp+9vHzVJf69Fv8fH/dT1OxbjzbxgVFVrbXDfvGplks/McBetlyb59UzcF
Retsbvbxzj2N65jjXp/4zrWqPdnrPHvb1zHGvX++q0b97toHchx5ORdtHb3Nwd30TrQ4+N+6vodi/FmXjBMVXVNa231w/YI
GNDwAAADBfdOIiywAAAADMTMADAAAA0HEDvcjyIFTVliTfGXYfHXNckn8fdhOMHPOC6ZgXzMTcYDrmBdMxL5iJucF0zI
2A4wk84LpmBfMxNxgOuYF0zEvmIm5wXTMi4PENXgAAAAAOs4KHgAAAICOE/AAAAAAdJyAZ4xU1UlVdXVV3VRVN1bV+
VddW1YbePp8l81xVHVNVH6+qb1TVzVX1bPOCqnpy771i8r9tVfUmc4Oq+o3esecNVfWR3jGpY4x5rqrO782JG6vqTb193
hmu85PcPGXb3CBJfqa1dnprbXVv22cJ70lyZWvtKUmelon3DfNinmutfbP3XnF6kjOS3JvkkzE35rWqOiHJG5Osbq2tSrIgya
Defy3J85N8vLffvJinqurEJC9N8me97Yq5wfR8lsxjVbU0yfOSvD9JWmv3t9Z+FPOCPb0gyb+21r4Tc4NkYZLFVbUwyRFJNsU
j3htrktyaiTCQ+efdSX4rya7e9rExN5gIgT9fVddU1et7+3yWzG+PT7IlyX/rndL5Z1V1ZMwL9vTqJB/pfW1uzGOttduS/EGS72
iSWRTxlyS4yAqnpZkjtaa9cMuxdGznNba89IclYmTvd93tQHfZbMSwuTPCPJn7bWnp7knuy1hN68mN9611I5O8nH9n7M
MQqDuah3nL6q5M8O8kxvSWzyUTwc9vQGmNYnpPk7Kq6JclHM7Fs+j0xN0iSXu+TtzTbQ8VkOrRF8Vky392a5NbW2la5NbW2ld6
1Jbm6tXTzlocuTvLb39WuTXDbXvTE8VbWsqo7pfb04yYsycX2mq5O8qvc082Ieaq39TmvtvxNbaykwsq/9ia+0/xtyY16rqyK
PZeL6Ox+O94uDpiZWQDEOquq5Sb6c5Po8dD2Nt2XiOjyXJvmxJN9Jcm5r7Qd9aZI5V1VPzcTFyhZk4hOrl7bW3llVj8/Ev8Ae9mrYe8ThoGrkzHWjqU2CP3RX7JKhZV1i+3KTHT9fu23XlxMeMAHbpJtKmhgT9b4W6HJ7OefJbJVD6Nmzfp4vXJYxL7VGnu1x+XLa6IYeW3OPN9FtjxR4n3sq6b9SNrmGk+VVyrhzkk3E
tV4Y/N0kT2itbe3t854xz1XV+iQ/n4k7/X4tyX/KxDV3HGPMY1X15Uxc8/GBJG9urX3B+8XBI+ABAAAA6DinaAEAAAB0nIAH/
AA9VXVUkucmeV2SV/f2HVJVf1JV36iqq6rqiqp6Ve+xM6rq76vqmqr6XFWtGGL7AMA8JuABAHjI2iRXtta+leTOqojojyc8lWZr
VJrqqqJFmQZNPctgsAMEHAAwCQpKoeneT5SU6rqpaJwKYl+eRMJUlubK09e45aBACYkVO0AAAmvCrjrmcrJX7bWHtdaW9laOy
qrye5NslPzl27AAAPqdbasHsAABhpVXVUa+3uqjo2T8eZqjo2yT8leU5rbfOw+wIAmOQaPAAAGit4AAAA/
      "text/plain": [
       "<Figure size 1152x432 with 2 Axes>"
      ]
     },
     "metadata": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
    "box_scatter(file,'Age','Exited');\n",
    "plt.tight_layout()\n",
    "print(f\"# of Bivariate Outliers: {len(file.loc[file['Age'] > 87])}\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "0TZdBBGRuF0o"
   },
   "source": [
    "#3 outliers in the above analysis view"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 46,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 447
    },
    "id": "IkvHx4OOtJgH",
    "outputId": "7bc49fd2-671f-4a9a-b753-85da28afea19"
   },
   "outputs": [
    {

"name": "stdout",
"output_type": "stream",
"text": [
"# of Bivariate Outliers: 4\n"
]
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzde5wcV3nn/+/TXX2fi6TRWJbli2xLAiyZcFEcQrJAfMPws2QnEGJI1mwCI
RK+rxfL72mu+pU1VNV55yqflTdx5xzAgAAAAAAQHhF+h0AAAAAAAAAAXhgSPAAAAAAAACFHggcAAAAAACDkSPAAAAAA
7203nK1oAAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAAgJAjwQMAAAAAABByJHgAAAAAABCjgQPAAAA
AELO63cAp6JPfOlTGhsb63cYAIA+2LdvnyRp1apVfY4EvbRmzRpdd911/Q4DAACcQkjw9MHY2Ji+/+hjjqqSX9TsUAMAii+Yr
QAAAAAACDkSPAAAAAAAAACFHggcAAAAAACDkSPAAAAAAAACEHAkeAAAAAACAkCPBAwAAAAAAEHIkeAAAAAAAAEKO
AAAAAAQo4EDwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAgJAjwQ
AAACAkCPBAwAAAAAAEHJevwM4Fe3bt0+RfK7fYQAAAAAAcEr4xcC+Ium67rrr+hxJ75Dg6YPZMZ2Vdn3tvTvMAAAAAAAOCW
DwAAAAAAQMiR4AEAAAAAAAAg5EjwAAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAgJAjwQMAAAAAAABBy
wAAAAAAEHIkeAAAAAAAAEKOBA8AAAAAAAEDIkeABAAAAAAAIORI8AAAAAAAIUeCBwAAAAAAIORI8AAAAAAAAIQcCR4
AExsbGlEql+h1GT4XiCR4ze4+Z7TSznePj4/0OBwAAAAAAA4IQSiid4nHOfkvQpSdq4caPrczgAAAAdVZNDWnPeCt122239D
AACFHggcAAAAAACDkSPAAAAAAAACEHAkeAAAAAACAkCPBAwAAAAAAEHIkeAAAAAAAAEKOBA8AAAAAAEDIkeABAAA
QMiR4AEAAAAAAAg5EjwAAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAgJAjwQMAAAAAABByJHgAAAAAA
DglrFmzpt8h9BwJnj5YtWqVqDVHhSe73cYAAAAAACcEq677rp+h9h9BzfEULAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAgJAjwQ
keAAAAAACAkCPBAwAAAAAAEHIkeAAAAAAAAEKOBA8AAAAAAEDIkeABAAAAAAAIORI8AAAAAAAAIUeCBwAAAAAIOR
HAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAAgJAjwQMAAAAAABByJHgAAAAAABCjgQPAAAAAABAyHn9DuBUFc1NKPW
AQAAAAAnET4DR4AAAAAAICQI8EDAAAAAAAQciR4AAAAAAAAQo4EDwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAAh
AAAAAAACFHggcAAAAAACDkzDnX7xiOi5mNS9rb0AF0ar2b7zheBMslHep3EMAhJhYtbtF02abyXGbJ4abJ4YuwX
0ArWgXQCvaBdDqlGoX/AYPAAAAAABAyPEEDwAAAAAAAg5EjyLzMyuMLPHHzWzMzG7odxb0AL5jZzG7odxzAL5jZ02b2
2gBNBh3Zxk5ntC64Z3zezNzfM+/2gjj9uZm9smN72fsrMzjWz7wTTP2dm8WB6Ing/Fsxfx/h7/+a2Q/NbJeZb
+qTk33BI2irppyRdJGlrw03HJyX9esNyV/R/R+d4Dj9ldqrZuL0QQ46bQM4EfyV2vfZHw+uGa9wznZ1ZkoJ7pGskrQ+W+W+TMzi85zz
Kekq/ocE7BYrpJ0R/D6DklXN0z/rPPN9W9ISM1sp6Y2S7nnjnjnRzCS9ri8y3fnf8r8Z9tWBdwwwnDO3S9pomnnyYrS
sGTmFd2kUnXDNEgmexrZL0bMP759S9kgJh5SR93cweMrP3NNNWOOf2B66PrP7ULrpNf67NdCAMFqaSGNCSzf2tMSBkFrnoGcG
/Hrm+RAacIBajHdDWEBKflHS+pFdI2/pv/U3HKA/zGxA0j9zZ3FTjPp3n3FOTTW1aRdcM7O9x+kk+q4RA6/Nms
s65g/nBQHjghmhFlM/ofYv3XO/e9gMtcMnNLatQuGd2R4Fd/xad1z+uSY/Kr+u10RzeN7SysTvaH/YXjGEQkb6
Nfla4LRTM6V/8Ow/6oO91PB0wfflPTWYPnmNZHNsZrF2+u0VYp2WVqk3fjyAtNSyQn3RTFSYUhNdeEt/8/0t6zDl3a8Msrhk4ZXVqF1wzurOQxHnSMH8Yt/8
7inPufwfSN8kdiSUn6iqTrwtLp4tRhZn8v6Q2SIiZC8ItnaSnJf1G7TdBzOowPjf2a/NFUft
zcErq0i7eLq4ZHZHgAQAAAAACDm+ogUAAAAAAABByJHgAAAAAABCjgQPAAAAAABAyAAAAAAAHIkeAZAAAAAACDkSPAAAAAAAAAA
qhmSdESSSzGzAzO4Lnup5xMyuai7cq7cqYyZrTazx8zs2s02a2y8y+bmapYN4aM7u34Ymh84Ppv2tmD5rZw4Lh4Z28yuMwAAOM
NI3JP1X59wXZCwp/z/VflbSWyx9hiSTTtF3SHzn7zzl7nSfHo7Vj5zlcKjk1r+i5m9Tl0V0ipJKYQdaF12Ul
05+Ptq51zJzJ6W6W/5RPo1/uUqbbQUK4iKdVl0ybpo865v3jewfwcl5Z2f0pUnkRSYcIDUs6GRrufk7k+166aLaRM
/arPsQso0+/eS/sLMtkqSfpF59zxXzkv7ZmcDnhdq/2nrNnLkvSSVDr4QnyMAFpQvSkcCf2AuqsBGkI4FRpVgeGuqYSI
AAAAhBwJHgAAAAAAgJAjwQMAAAAAABAyAABByXr8DOF7nbk8Yn7H6nWtTXnIc8vHi6eOm5HABVtmczTzGSGKWqPXeZVLn7ncH6/P
JyMevYbPGYWlXSfM9skpSSRTS0kkoH9SG19qeD5OQr5wpHQ3F7LbuFK1Nm0J8vK1RJP0Ywr2c42mcAAAAAA0GZpmK2rzFGqkwnrmv2KMotpLE
IaIWFsIOuWNbyIElV55SMSRVWuVHVopqh0PKqRgZhikcJQTXEP0CqRsvzK26aAAdmMLjVxqTfnRuO1Z4fPDqqYWeKQNVfObS6LqGxQGdL
tKKBOP6tmJnEYGEqqqNHOPIYxu6ep7MiS1w5sKEvmvpK22hQmiW5YmCj7Sr6RiO5egbMT3o2nsO7TR0z2AcL5
BoH/MhpIxFSpVTc2WtVFJj94oWSvVdXvS55PT3r4gpNbGJK0uaMF7r3NJAH4k4M3nMLZ11q1NW1c1cnzd2bSKQCCsmOpNRi1KzU
LMvMNJDwlElENFusarrgt5HMHzrN8OOsb2nO1haMzTgRXyPLzQ7qcv3ZkLCXLLZIwqqAkE9V4MNzCkMW1VQ2bZJs3nRNZ
X/ekZzfx+VLZaUTnxLlUvmsr7bX2mRUtQHXwkeHtGR8R3NBJVKhqgqOAGJoIUW/W4wwWxRpwwf90kyhyaEpTb
J03m8rIwMxTecrej5oL8ImusSnr6PclZ3TFt2nLA+4ygr4BdtoHEiPWqzK8lcqqEKqRwkYgHEpSRR9ObqgVpGfTwbo4I
LGgnqz0TQ1pKgxA7nilqSmLOJWrXvu87xvdV/zoixxtbqScmlOWHmzNd5IYgrS7UdMdWMp5w0+dbfgpGhpvAvL6R2vdMSvKi
jEeV9Pzr9kS2pNHBuMi4nn0AawaecgYdtmgNmmyttRXxmEOUKPXJyx3pxarx30XVVtIhkYsDi6x
WjOb/OTs35dKZSr9fM7MuBpMlfRRK4U1PGSUrGoBhNRFSpO49N+uXTMbo9LMjHNFisany5kTQ7MzwvnpMvmQUeNavlW
zf3kYH44pFl/XzlSuWFfciSnn+NfaMpUlVKv79x1DSC/q7ikYyiYi73gLV7xLOXpvXskZwOzuRVKjKcdIwBKxeOV
xrHg82pfPDWH4rAI06+WPLF8kacwVtOR5e5jcN5V2pxnCtGmZkP1qVwu7Tnb3JDL/7wZpHeEX1fE5HvBp4nM/8B4+vEuH3gisiYr5hbvRdxL7S9BKOyf
xrP1uZ3P6Jc2nq0//cbuevx38UjoptvamRvM3fv4wXr5g7t5NxNTFq5fX+aXneHrlVJb7dcY/Geh3yWUsXNgl8KmU/YibvV92MqZeyG
bxt83pFlv6Hxtvua96XZ3TNT56tz/7LXh3JFbV103r9+3e9+T+Nae/hWSVjEX30Fy7U+HRBX9+1X2951dm6e4xg0vc3iJ2KmYpU1d
WbY9965Xr9w3ef0SUvO10jmZiyxo+9tXv1Zf9ze9VKm4pw81nKfm9f39d/d/bqiYMz+uO03aRe7K66oiPZRpWb0
fTmj/0aI+9KVHO56DYjmn//KVH9WnfewtF2oo5Wnv4Zxub5bNjGBy5dp1QsMfkzhaCUvf3XaRRXSB0YKVahrqzYrtLz

DscHPERERERERERERERERERERE...
oi9HPH+G0e7auPNGBpsz0Nv1tytuH1fYtNIcN2KtosFs7M4oi/Y/9XWx+7u32r7xnWH92SwabTUtG203c5iBYaqoGR7GMi
vr+nhmyzi2jFs4tCeNlKZie6ESW6boc21ZGreLrntKU+H7EiXHhecDjutjds7EhGXDUNW6OKNrvrOwq3wLeuuve9QmbNdD
Gdek40BQNO4sVzErrSOsqbM9H2fZRdjzkUxrSugrXl9g+aaEvZ0JVBFQhMFKykTU1+NJHWtNQqLgwdRWOF5wzY2jYXqig
aQ1lx0VaU6GpCkaKNvJpHbbjIW1oKFQcmKoKU1eq5yk5HooVF4d0p5EzNWydsDC/J4VtEza2TliYkzeR1gVKjsSOQgV9OR
U/6yBkathds9OZ1mIqKCctFoeKiL2cgpQftz/EkHN9D3tRhOT6Ktot8SoMAYGoqSo6LrK7BB1CsODA1DaMlG7OzUd3qcD
foowICk5aDrrSOSctBSlPRldZhOS7GSi66MzoKFQeaoiBnakjpCsZKNjRVxbbw/JDA9kIFOVODqSsoVBz0pE1YroexkoOsEb
lwlAVbBopY063CQAYLzmYldEhALhSwnYlCpaL3pwBy3GRMYLylB0fjutjsuIiravIGCryKRUTZQ/DkxX05wyYqoJxy0Ha0DBZ
A1NXwn7rQUoF2yYrmJM3YWhBeW1PomQH9deVUjFZ9jBSstGTNeCH5SyGbdP1PahCxc6ijbypIZfSMGm5KNkuerMmxs
lzkTA2GLqArKiYrLgphP9WUcOwydOwsVjA7ayCtqxgr2ehO67A9WS2PpgApXYWhKXA9iaLtYWfBDvqOCPq7hA9DVcMx2
AAHklJREFUqNiwo4hNI0WkjaDvd5k6RksOulIaTE3F8ISFXHhNovG67HiYkzfh+BKTloPerImS7WGkaKM/b4bX3USXqWLzu
mcPScPF4eL2NnMbgnFSwPO4o2ukwNaUNF1lQwXgr6ydwuEwNdJl4Zq59zHdqdxp+2TmDzWBl9+aC/qIqCtKZCCKAYzk
UHE9vDJWRl/OhOP7GC06OGRWGksHdh1rIJ+CgMSfdxSDe31Gx9I5XVAUUb3OWVNDxfVwSHemOl9Imsc1zlvamT/GzZs
Trtd5/pdqCU/0Apx1SElHLvHFiIkwEUANyc8IDnrQA+guABz4kAviylPHGq4y5fvlw++uij0x3uPjFWtnDX09vx6dufhuX4SOk
PF4QK+fM+LdeU+8+g5uOvZ4T2uj92tz1bbA6hbt6A3jY+cNoh/+vnTsTH/66+exTuWH46v/HZXma5YMYibH9yI0ZI95bZXr
IjTvL1c8/fHgjTls6txpfXNkvP20Qv31uK95z0kJsHbdaxnnK4ADue3EbXhkto2h7sdf9vhe34cXhAn74yKbYepzfk8ZE2cFn/9+
Z6rHWH3OEL55/7pqXTV+vuqMJTBVBZ//1XPVY1x26mBTHIfMSuG7D75UV8cpXcE//e3R0BQFn6k559Urh/D1+9bBdmXT
1CouHVxXLNqGe7+0ys44Yh+fOGu56vLP3HWUejLGfiH256qOz4g8bX7/ox3n7AAX/rNC+jJGPhfb17U1DauWbUMZdvFD
1dsWIQWUPFE5tGcdKRffh0TZ3Vtq0rT1+CtK7gn3/5XFMMbzv+8Lr2VNtP5s9KI2OquOyWPyaeP2dqkFLWHfuKFYNY1JfB
PbcJFbzwCIwUb//Kr+ljnz0rh5gdfwqMbx5HSFXzqLUtRdnzc8vDGpjivPf+1KNteXfusreu49n/l6Uvw2MYdWHH0PHz9vnXx
Ch6C7Sw41T4Qd83nz0rhi3e/UD3uFSsGkdFV3PjABrznxAX4j/9+CYYm8L9PWVytpwW9aXz09CVN43Lj+BTFnTUVWHZzjF
OFWyfbxsZ+sbTrfu09YgFse3oh3vuHwavv41FuWwtTVuvbzybOXYnZOR7HiN/XFqI9+83fN7aa6Powt6f78xF924pzXzsfml
59fgtOXTqvqf9GbTO6J3313hfryrigN40Pn7w4sWxrVi3Djx/ZWO0zHz55Mb55f3z/iuo/6ec1q5bha/e+WNema7db1J+F6w
oOLKubhrHnrj9kpbtydx4On436bTffabbgVL+A6UctYQQj0kplzct31sPeMKTLgRwZ8IDnm8BuE9K+YPw8/MATpFSbml1zE
3Ou3F/DzJzbj+vvXNx3nR5echHdc/9Ae18fu1mer7QHUrbv01MW44ffJMUd1EFc3X7t3XVvbfuH8Y/HxW4PJa9L5on0vOXk
+9fn1g3l5y8CADwlXvWNcVQGysAXL5icWJM0bW45ORF+Mo963DpqYtx9Nw8XmgoR7t11fg5Om6rY1xy8iIsHsjvVtwAmc
Qlvm1ccvlieD5argeA5Qt6MFZ2m+q0sY20astTbXPJyYtw0qLelnWWVOa4dtrYT5Yv6MFFNz7S8vxAcx9Iur5R3bXbbq6/8H
GqfpT4/W6/sLjcUnCtV2+oCd23Xc+cAIe+POOluNPu/eR2vEwqZ984fxj0ZXWYmNpHO+mah9J5Y3OndRu2hkjWo2RjXFee
U7v38ahvJrW1pPZcO+eZqv/X9tepxsSk/ZKW7cnceDp+N+m0332m24FS/gOlHLWSHvDM5Dt45gP4S83nl8NlTYQQlwghH
WVMijlpvRDtb1u03eryVsezHL/6Vd6pyrK1ZvlUMUafyxW3rbKXbTfxWtbGuWU8uOatrnu0LuIc0f5NMTTECiSfp/ZaRMcSA
TrfQmMlpzYttDYRlq15am28SWmrLOkMifFVttPRkvOlOePrc+EY0d11267GS06iccaK7V/fVuNC1Hbb4xld9t3q/VRW5yq/w
nfjVjRKsxsjHOsu3G9rekco6F40OrMiT106hOGmOINI7HSeVvdYzGc7Xzc9y6pDEoaZ4ZzX2S4mk1f0yaN7V7H5+qrSW159
trFvKnbfgXx77++ZLvtLHO1KilfK9uOe05WCKhB7nHnd6VdVH7tbn622T1rXHPOu7ZLqJq58cdtmw3dXtDpfVNeKQF25ku
b7232wq/nr25+PH0LgypnQFPVk98ZyzsnpiWRuXZQ2tqTxJx432b7VeEUBPRk9st43Lk9ryVNsoAlPWWVKZk2Kr7Sc9GX3E
t3qv5Uu6ynRd+dnbCuP598H49ia/c+oghUt03aJ2toibE0jndTXbek8taee0/HiNktxsjGODOGFtvfkso5KxwfovPsTj9tbENJ8b
ffabbgVL+A6Uc7ZjJBzybARxW8/nQcNkBa8ncLNasXFY3sK9ZuQxL5man3HdhbxbXXXBc3b7XXXBc9UVk+8r+Eke74uK9/
/OWm8sVte9UZS6AoqNbLbY+93FRHUV1fsWIQrz20u65cSWV57SHd1bjjjvnZ85bhzic3132+6YH1dfHFlf3y0wZx0wPrsbA
s1Yfe5Q3TFWnzNUV1eNn686YwI6M0bdMeLiOLI/21THKT144PKZhnNevTI4R9y1m50xgndYxNR97TUoWU7TdledsQQ/
9r4I61mzE707z+qjOWoC9r4Mj+LMZKFdz0wHqsPmeoqT5uemB9tZxx9bj6nKG663HbYy831dcVKwbRlzXwk0c2YU1Dnd
uXjnUFM9nzh3CzQ+sx5EDOXzy7KXN/bg/i5sfWF9d1ps1cOXpS2LjPHIg19Q+a+s6rv1fefoSfP+hDbh65VDyWFITwx1rN2N
WIKfPv4y7li7ua6e7li7OXZcbhyforjHynZsjH3Z4AXqje1hzcpl+M0zQdmSxp/V5w7F3kcajxX14Wgcj+vDq88dgqIAnu/ji28/N
ThWqjRdj6tXBn2w9jxrGu4tq88Zwvcf2hDbf6M2FN2TGst4x9rNLcu2ZtWyuj6z+pzk/hXVf9LPa1Yta2rTtdvpqsAX397+PD
r5D5RytGMm38HztwAuw66XLH9FSnnCVMfs5HfwANOTRWvbZPDG+5nOojXTcbSrPouWCsfzMftVZNEq2S4Oj8mitaf1sk
OohVknDI1BdLHrixano+ejIFCxYE+RRYt2/WbrntzFi0HKV0JsIFlNOyctFG0PRQrHgbyBoQQ2DphoTfKojVhoS9rImMmZ9H
aQ2W7WN7oYKsqcHUBLrT9Vm0guxNbWTR6krBcn0UoixatoeUrkBTFIyWHORSWlh3QRYtI8yiVbAcdKcNlJ2g7uZ2m8ibO
md1mGqQRSvIyGLA1BWUwyxaru8H2VQcHyXbQ84MMsVEWbQyeIB/RduBqdZk0QozM7I+kCkrMYuWoqJgN2fR2jZRQo
u0EWTgirJoFcMsWj1RFq2ciYrnY7zsoKuaRSvIPmZoCiqOjwnLRdpQYWgCWT3IVqOEWbRGijYG8iZMTcGm0XLw7ViJarYgI
WLUODhAc/zKIVlG9XFq1yxUNPVg+yaFn1WbQcV6JQCdqmJz0oIsj0IzM15E0NkxUXJdsLsphZNtKaGI4fH74PjJSCbXVNwFI
u4IsWj0ZAxIDxXjJRleYRSsqj6oAaU2Foddk0QrbgyKC/i7hQ1dUlJ0wi1YqaD+aqiBvaCi7Hkq2i5wZZNGanTUhwixaJTvIQpU
+ZxC044jhcqQd8phVm0fN9HSg/6eV0WrbSOcas+i1b0J3hzu4J+GlwJQT0cPacLL4+XMVIM7kkFK6jfnKkh0zKL1q45V10Wr
CbNeBKH2NFF/O6TSwNj7Vt0kJ/ro0sWmHmsHltZNFqnLfsThat2nkTgKa538aREjaNFJGpyaJVO7+r3ac/t2v+N7cryKK1vdA
/GaZJ/yqAsxGkSf+AlHLKJzed/oCHiIiIiIiiGhPJT3g0eI2ng5SyndNsV4CuHRvnZ+IiIiIiIiI6GAxk+/gISIiIiiiKiacAHPERERERERE
wo57wHOgEEI8GvdSJKKDGfsFUTP2C6Jm7BdEzdgvijodbP2Cf6JFRERERERERERERNTh+ICHiHiIiIiiKjD8QHPzLl+pgMg2g+xXA1
kI8L4RYJ4T45EzHQ7Q7CCFeEkI8JYR4JR4QgjxaLhsthDibiHEi+G/PeFyIYT4StgnnnhRCvL7mOO8xt39Lt39RCPG+muXRCPG+muXHh8dfF+4r9n
JCGOFyM/y8Ly/cN+UmGhqQojDhBD3CiH+JlR4RghxRbic9wJ4kHFC9xm6aLXoF7xntMAHPPuQEEIF8DUAbwRwHIB3CiGOm9moiGbEGiHlsprh7ItB3CtADVGhggYGaUAHWJFDV8zw8HxTKdLxmajItpT
/SDoH0f7gO4gfs/b3BxdM57PCOHpRXkRcrl/61/V2KAgDCOdK71/BYiwGMAvhbgDEAvhbgDEuguPyDEbD5V8KtyPaX2AkwBcC
UspRAHcDOdtc1yWlfEBKKQFcV3MsojnDfZE9t6tc1yWlfEGb4m4MFYFAbQDqIZ5AvAkpnodAHOgRzqdj5VPlNhNMA3Br
MvmSCm3hD9vBTAn/DmpxX7Ra/nLMCK6DcvrjhWuHw+K8joAfwCDvzjIPDzJAGBqQT27joAfwDv
TIAbgNwEellBO163jPoINVTL/gPaMFPuDZtzYDOKm86Hmq1DipRyc/OVEr
zWuMDnn3rEQCD4du6DQQvgbp9hmMimlZCiKwQIh/9DOBMAE8jaOcCB4+FXhXwM4UwjP

fnE+gN+G2xPNuHAcvwHAs1LK62pW8Z5BB62kfsF7RmuiQ+I8YIggjdu/AVAB3Cil/NwMh0Q0rYQQixB8awcANAC3SCk/J4
AMwjCCzyc+xl/tBUl/b6wUmakNCvzgFwVftJYCXAHwoeieIEOL/ALgYQTaVj0opfxkuj51PhfegHwKYDeCPAN4rpawIlVIAvo
ERERERdTg+4CEiIiIiIiIi6nB8wENERERERERE1OH4gIeIiIiIiIiIqMPxAQ8RERERERERUYfjAx4iIiI64AkhPCHEE0KItUKIx4UQb2
yQoh7wm/1PCWEWNW4cdI2QoiFQohnhRDfFkI8I4S4S4SwiRDtctFkL8puYbQ0eGyz8hhHhECPGkEOLqfVhmIiIiOsAJKeVM
IcQfAPyLlPJnQogUgv+p9iYA5wP4EAAB4HYA10op79+XdUFEREQHJv6JFhERER0Mav9E668A3CyEWIbgQcs/CyFOBuAD
4bfxOkH8Nbw3+OlI4Q4iUE3/Kp9Z6W4W21RqtvMApFucWgD4vJTyW6++FEREREERET1+A4eIiIiOqgIlZYCUAHsBNANYFv44OZU
Di+wDuEEI8BeBRAM/F7NvONo0uBPAtIcQaAA6At0sp7xJCHA3gQSEEAvfAtldTMCIiKAL1kmIiIiIiIiIup4/BMtIiIiIiIiIq
ERETU4fiAh4iIiIiIiow/1/IB+Dzb9+iFAAAAAASUVORK5CYII=\n",
     "text/plain": [
      "<Figure size 1152x432 with 2 Axes>"
     ]
    },
    "metadata": {
     "needs_background": "light"
    },
    "output_type": "display_data"
   }
  ],
  "source": [
   "box_scatter(file,'Balance','Exited');\n",
   "plt.tight_layout()\n",
   "print(f\"# of Bivariate Outliers: {len(file.loc[file['Balance'] > 220000])}\")"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
   "id": "MbKKNqFot6rA"
  },
  "source": [
   "We can see that there are four outliers above"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 47,
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 461
   },
   "id": "m3I6pfwStjfT",
   "outputId": "d3d58045-af41-400a-961d-02e6c3b73163"
  },
  "outputs": [
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzdeZwcZ33v+++vl+p1ZrTMSHYElmxLNiCFVRBykrDZZguSycE3MSEEEs6
f9evVruqufrnqqnqWqf9PdP3POCQAAAAAAAOmVWe4KAAAAAAAA4CdDgAcAAAAAACDlCPAAAAAAAAACkHAEeAAAAAA
g5XxFCwAAAAAAIOUI8AAAAAAAAKQcAR4AAAAAAICUI8ADAAAAAACQcgR4AAAAAAAAUo4ADwAAAAAAQMoR4AEAAA
yuWWuwIAcLZ797vfrb179y53NQA8TPbv3y9J2rBhwzLXBMDDYfPmzXrta1+73Nu8UNAAgJ8YAR4AWGJ79+7+7Vt759l4LymuWuC
DkCPAAAAAAAClHgAcAAAAAACDlCPAAAAAAAACkHAEeAAAAAACAlCPAAwAAAAAkHIEeAAAAAAAFKOAA8AAAAAAE
4ADwAAAAAAQMoR4AEAAAAAAEg5AjwAAAAAAApR4AHAAAAAAAg5QjwAAAAAAApBwBHgAAAAAAgJQjwAMAAAA
AAwAAAAAAkHIEeAAAAAAAAFKOAM8yePe73613v/vdy10NAAAAAAWhJXwPjy33BVYifbu3bvcVQ
eAAAAAAAAFKOAA8AAAAAAEDKEeABAAAAAABIOQI8AAAAAAAAKUeABwAAAAAAIOUI8AAAAAAAAKQcAR4AAAAAAICU
gAAAAAAgJQjwAMAAAAAAAJByBHgAAAAAAABSjgAPAAAAAABAyhHgAQAAAAAASDkCPAAAAAAAAClHgAcAAAAAACDlCP
AAfTKNae3dO8N1ObAC7N27V6VSabmrsaRS8bmsaRRS8QkeM3uIme0xz2HDh1a7uoaAAAAAAACcUVLxCR7n3PskvU+Stm/fnvp/g
4AAAAAAAAAUo4ADwAAAAAAAQMoR4AEAAAAAAEg5AjwAAAAAAAApR4AHAAAAAAAg5QjwAAAAAAApBwBHgAAAAAAgJQjwAMAAAA
AAAAAACAlCPAAwAAAAAAkHIEeAAAAAAAAFKOAA8AAAAAAEDKEeABAAAAAABIOQI8AAAAAAAAKUeABwAAAAAAIOUI

+yYj6LVoJBMnGthHL1odSVPJe9Rj5rsnY1YVYn7hLRfHinF/5O5fy5kzVABOBBFayapu2Ezjoiye6ZBMYjnIWHrQBStmVqI58
tqNilno2glkWSKQSaKlmvi57z5aKFDRY/ZJIpWNemr4mh0MFLyafSJojVbD6lko2jVQ9aNtEfRSseOop/j4f11SgWPYpBLIl7la
xX5CNojU+FNBsxtGp0mhQ0/WQ6bm4vg4X4k/EPDwVv55G0UojAJWDuO+b6IiitboSUK0nEY2SOcKemYixih/3FY2IPTNx
nvHTyRrHJlG0dk2n/b7P1FyDwPPmo/7UknggWijUA1bRM14PlZNxoTxSp6pesjqcjzm75ltUAtbyZzX5ud6bVG0Ap+ZRsh
BFFazaZc87UIwIvro9zUTxnGCn55MhE0RoK2D/XI4pW0afg5XhoKh5vS5koWun7h/1zDdZUCnFfnczNS4HHVC1kpt5kuO
EUrcb8PKhS8Mj7SRStoQKzYZOZZN9qEs13phYxMVyg0SeK1lwYR7vdO9uYn4ceiKLVpOB5zCRRj/tH0YooJfWkLYqWZww
htDDLEUYtZupNjh4pcNq60RX9A8uwDD+ybGafAM4G1gCPAFcAAYBz7pokTPrfAOcTh0l/hXNuwZWblb7AIyIiIiIiLyaB3y
WAPsXu5EyCGj8j6yqLyPPCrzI4vK+8ii8j6yqLyPPCrzI8vhUt4nOOcmOjeuuAWew4WZbe31o0hyeFJ5H1lU3kcelfmRReV9Z
I8sKu8ji8r7yKMyP7Ic1uWt3+AREREREREREVnh9AkeEREREREREZEVTgs8IiIiIiIiIrnBZ4DjEzO9/M7jazbWb2luVOjyyemR1
6kXLcm21ab2Q1m9qPk31XJdjOz9yVld4eZPSVznt9O9v+Rmf12ZvtTk/NvS461Q3+XkjKzUzPt+LtmNmVmv6c2fvgwsw+b
jW5/y/lSmrH9sZt9NtquNr3DW/72YxvEs55weh+gBeMC9wMlAHrgdOG2506XHostvHfCU5PkwcA9wGvB24I099j8tKeMC
OfaC5b5nPebL2QMeBk5QGz98HsBZwFOA72W2LXmb7ncNPZalvJ8D+Mnz92TK+8Tsfh3nOahy7Vd39Fi2Ml/yPhx4DXB
Rwz4JCLgU865+rOufuAbcR1oGc9SFaInw18Njn+o8ALluZu5GdwMXHZQHsZXQxc62I3A2Nmtg54LnCDc26vc24fcANwfv
Vp4/38PPvwbF34LHBO+j//snQGlXeS/78OfGLQOdTGV44B78U0jmdogefQOgb4SebvBxm8QCCPUclHb58MfCvZ9Lrko38
5vZL4f2hTJ5nZd8zs62b2K8m2R1Oumu899ix1Hz5/TPL6/mR/WT6/AjzinPtRZpva+GGi472YxvEMLfCIHCQzGwI+B/yec24
poeA451zTwYuBz5uZiOLPZ/K9TFNffiR6SW0/0eN2vhhosd7sXkax7XAc6jtAI7L/H1ssk1WCDMLiDuUjznn/hnAOfeIc67pn
gQ4FG263zVkGZjZy4ELgd9MJuokX9PZkzy/lfg3WDby6MpV873HkEPUh88fk7w+muwvyyApg/8OfCrdpjZ+eOj1XgyN4220
f/NvD5pbwn6c/MKmY2nD4n/mHO7xGXa/pr+9kyug54WfKL/U8D9icf5fwy8BwzW5V8LPw5wJeT16bM7GlJ3XoZKu/Hirb
N2vjK1++9GBrH27nHwC89H0kP4l/zvod41fiPlzs9ehxU2T2D+ON4dwDfTR7PA/4RuDPZfh2wLnPMHydlfTeZCEn96gFxxI
3/eR/gAqxP/rOprZpjZ+mDyIF+4eAkLi79a/6lC06X7X0GNZynsb8W8vpON4GvnohUlf/13gNuCiR1uug+qOHstS5kvehwPF
IiIiIiIrHBa4BEREREREREREWeG0wCMiIiIiIiIissJpgUdEREREREREZIXTAo+IiIg8pphZ08y+m3m8zcC+LzCz0zJ/X2Vm5/4c0jBm
DOubf9nNIwBrwGeP/PcI6PAr/unLvdzDzg1J9LyhJm5jvnop/nOUVERGTl0id4RERERERERERZEUwsz81sx+Y2R1Y2R1m9l4zezefBjfjz5FM
rj83sHjP7Bu2LOGuBhwCcc03n3A+S/c80s2HQmSuyIH6P0S1+Qv0BRERERERERERERE5
1gA3J+d8C3B6+kkiM3sOsAE4EzDgOjM7C5gFXgw8ihXXhedRtwa3LtvwLuNrOvAf8GfNQ5VwN+CPHeketM
szqZjZGvEDzrmSxpgUcAxzV4zzPSR7fSf4eIl7wGQb+xTlXTdKZXg/n3FVm9Nm9rHkuN8AXgKcDYwcCzDYwCHzWzDYADgh7XG7TPDc6
w6GfBu59yTksd659yHFpHue51zfwucA5xhZuPAO4CvOudOBy7B8+Fhv9nw8sGaqaXpgUdEREROTxpgUdEREQe8
wmVxzR7L95QPStdA+qY8AvweQ/s6PiIiIHHn0FS0RERF5rOn8DZ5/A/4a+LyZFYk/TXN58tongQ+a2euBSx5euBSx7FtT4GfMHM7g
Ffg/fdK1mH1I0vqImd0F/OujuH8RERERE5TJhzbrnTICIiIiIiKPkpmViX9j6CnOuf3LnR4RERERFZHvqKloiIiMgKUTYugv4/7W4IyIicr
iIrnBZ4RERERERERERWuP8H5Un1bmDVYIgAAAAASUVORK5CYII=\n",
     "text/plain": [
      "<Figure size 1152x432 with 2 Axes>"
     ]
    },
    "metadata": {
     "needs_background": "light"
    },
    "output_type": "display_data"
   }
  ],
  "source": [
   "box_scatter(file,'EstimatedSalary','Exited');\n",
   "plt.tight_layout()"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
   "id": "xrEiRm_6uL6t"
  },
  "source": [
   "### Removing the outliers"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "SPyz86j-t0_w"
  },
  "outputs": [],

"source": [
 "for i in df:\n",
 "   if df[i].dtype=='int64' or df[i].dtypes=='float64':\n",
 "   q1=df[i].quantile(0.25)\n",
 "   q3=df[i].quantile(0.75)\n",
 "   iqr=q3-q1\n",
 "   upper=q3+1.5*iqr\n",
 "   lower=q1-1.5*iqr\n",
 "   df[i]=np.where(df[i] >upper, upper, df[i])\n",
 "   df[i]=np.where(df[i] <lower, lower, df[i])"
]
},
{
"cell_type": "code",
"execution_count": 52,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/",
  "height": 447
 },
 "id": "FMcWZiddufC5",
 "outputId": "9d38b728-914d-44c3-a067-5257ea2ffeb6"
},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "# of Bivariate Outliers: 19\n"
  ]
 },
 {
  "data": {
   "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzdeZgl510f+u/b6+xaRqMFy/KYO/ImGWwzGPKQOEZerjCKRB7ASxJslsQ
kePdNVp+qtd6/qn6rPW2qtAQAAAKC7xtY7AwAAAAB8dQR4AAAAADpOgAcAAACg4wR4AAAAADpOgAcAAACg4ybWOw
gAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg
eEwJ8AAAnbF37958/NN3ZmHTheudFR5nxo8eSJLcP+Px9lw3fvTh9c4CwLpwBwQAOmVh04U59rSXrHc2eJzZ+Nl3J4m+C
AAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAA
APAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8LAu3vSmN+VNb3rTemcDAACAc8C58DvounodsKRr
CPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAA
PAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPOvgXe96V57//Ofn7W2688cbs37//tM7fv39/brzxxuzdu3fZ85v
rfmU5/6VG699dZIz2+Oaz5faXu11x08f/tZbb82RIz0dy6623nIYaDzzzwwKqOBwAAAFYmwPPYe973pVaa3+71pr3vOc9p/U
MT652Bc03z9sghYWFvO1tb8uP//Pr3j+W9/61iwiwuLq54/uBxxCwsLufXXWW5RfdXun677fTe9ra35ROf+RJx9966+9V//9V
yZ9++fuedna/K4Ns7jfn5+bzvfe9b1fnvf//K//7Mz8/Nz8vvu+7vvXnZ7peu303vf+97Xf3un0d5eKe/tcgAAAABnphNv8N
b6f3ohe9KJ/4xCdOCurs3LnztPK+ffv3bRt8jxs2uKGbdn19Zd4ZgROci681deJN3jOJj/+vvbt23ttdemlJKdO3eOPH/wuGbuttvTV/
rPV9pe7XUHj3vz+9ZV+pe7XUHj3/DG96QzZs3r/j2TjsNfsmbYsbqTsb+A59sNb1+Z9o0zHB6uXB6B6JcwjW2aUgB+CPw8+rmNVlcqVr+
AxwnwAAAAAHScAA8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAAAQMcJ8AAAAAB0nAAPAAA
AAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHTcxHpngHPTrl271jsLAAAAnCPOhd9BA
xwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHTcxHpngHPTrl271jsLAAAAnCPOhd9BBXhYF699
AAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHS
AAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHTcxHpnAADgdIwffTgbP/vu9c4GjzPjRx9KEn0RgCQr3ln
ADwAAAEDHCfAAAAAAdJwADwAAAEDHCfAAAAAAdJwADwAAAEDHCfAAAAAAdJwADwAAAEDHCfAAAAAAdJwADwAAA
WAtPqrXuaO/sXIAHvtZKKXtqrbvXOx/weGNsHBbMIhCpjAseSeS/5ECwAAAKDjBHgAAAAAAOk6AB5K3K2ID
73tvb3Pd65nvmEtlV8fHJUspPZAMwnOZPgQAAAAAAAAAAAAAAAAA
9Qkkd6+/9j7zg4m317rfVZtdbdve3XJ/IArfXKB5svffq5P8c8p/DYeWOS99Ran5bu5bG7N1/Xu188KG8K
62+l9/oLe8XCuGBwD7bHxtrrkI0nOL6Vcth4EBg16Q5C9VfrrV+fjfvSklSvOShbInS1ofjbEBg16Q5C9VfrrV+KsQFJMpFkYYlmJF+J3zd
k1zS+7k/NnoGxw2cTZ6cZF+S9pZlk11vuuS/Psk92OmmlvuuS/PsK92QpsHMgye3x3x+wbrQCICHs1op1ZokD9s
/W+d+qGLP0PgqLqLjaAhAsrzWPyLP+iZQ6vSaWX8mSrJ0vvv8vCT7H8SMw2Ol93+cUm
tqrXNJ/t8s/Q7iw0ecwNVqrT9da7281rozTF5YRX6vq7nd7HnH+z93ygk5q5RSNdpYY/J3lxk

QQgghhBBCCCFFzil9yfKpQET2AdjR33KQAUUNgP39LQQhf4UwNgjxw9ggxA9jg5BcGBfkVDBJVWvdnUV3g4eQtxsRWeN
fhgbhPhhbBCSC+OCnDb4Dh5CCCGEEEIIIYSQIodP8BBCCCGEEEIIIYQUObzBQwghhBBCCCGEEFLk8AYPGRSISKmIrBeR
4eIjBKRB0XkJRF5UUTew9gggx0RmWbWC/vviIj8A2ODEEBEviAim0TkLyKyTESq+H2D9Ae8wUMGC7cCeDG2/W8AvquqU
EGNaq62awX0wFcBOAEgIfA2CCDHBE5E8BCAE2q2gigFMA14PcN0g/wBg8Z8IjIWQA+DOBesy0A5gB40DRZAuBK83me2
u8PsG6Qd4g4cMBr4H4EsAesz2WACHVLXLbL8O4Ezz+UwArwGAOX7YtCdkIKIAHheRtSKywbr6q7zOfdAMabz5nYMI
8lfIe1V1JqLH6G8RkUvjB1VVEd0EImQwUQZgJoAfquoMAMfR+5MTAIwNMrgx7xFpBvBL9xhjgwxGzHun5iH6D4IzAFQD+
hF683evaZ6JDUM8bggZSLwO4HVV/ZPZfhDRDR/GBiERHwSwTlX3mG3GBhnsXAFgm6ruU9VOAL9G9B2E3zfiIaYc3eMiAF
QTMmBQ1d0AXhORaWbX5QBeAGODEMu16P15FsDYIKQVwCUiMtS8SeuG/y+QU47Q3gi6o6V0TORfREzx
RCTnliMh0RC/mrwCwFcCNiP5TjLFBBjXmPwRaAZyrqofNPq4bZNAJAjIosBzAfQhei7xacQvWuH3zfIaYU3eAghhBBCCCGEE
CCGEEEIIIaTI4Q0eQgghBQ1IjJBRH4uIq+KyFoR+a2ITO1jXz8RPkRkEQ7zmGEEEII6TtI
JOMhBBCCCEWPsFDCCGEkGLmMgCdqnq33aGqGwCUisjvReQAC+ISKmIfEdEnjNP3XwaiG4QicgdIrJZRFYAGGf7EZFVIt
85b72IzDs9JiSEEELIQIBP8BBCCCGkmGkEsDZwbCaARlXdJiILLABxW1YtFpBLA/4nI4wBmAJgG4F2Invp5AcD98U5U9csi8jlV
ERgF4VkRWqOrxt8dUhBBCCBnI8AYPIYQQQQgYqz6rqNvP5/QAusO/XATASQD2ASwEsU9VuAG+IYn2nyoqrHROQiAO9D9
AK4KHIs/+SIAPq+qv4s3EJEP9WVQc0NoFYBVlvI8gOsRfpIoH66MH1XVzX2RiRBCCCGDG76DhxBCCCGEEEIIIYQQQgghhJAihzd4CCGEEFJM
b6bOL8D8Hnz4miyIy0RiCEEEII4RM8hBBBCCClaVFVF5O8AfE9E/gIAG4DtAB52mt4Lyjdey8AVyKqwDUH0bt3WgH
GEEEISEVXtbxkIIYYQQQgghhBBCyFuAP9EihBBCCCGEEEIIIKXJ4g4cQQQQQgghhBBCCGkyOENHkIIIYQQQgghhBBCChhJAihzd4CCGEE
ASUVORK5CYII=\n",
     "text/plain": [
      "<Figure size 1152x432 with 2 Axes>"
     ]
    },
    "metadata": {
     "needs_background": "light"
    },
    "output_type": "display_data"
   }
  ],
  "source": [
   "box_scatter(file,'CreditScore','Exited');\n",
   "plt.tight_layout()\n",
   "print(f\"# of Bivariate Outliers: {len(file.loc[file['CreditScore'] < 400])}\")"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
   "id": "tTwXFuDXvDjo"
  },
  "source": [
   "Removing the outliers"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 53,
  "metadata": {
   "id": "823mThKxvGbI"
  },
  "outputs": [],
  "source": [
   "for i in file:\n",
   "   if file[i].dtype=='int64' or file[i].dtypes=='float64':\n",
   "   q1=file[i].quantile(0.25)\n",
   "   q3=file[i].quantile(0.75)\n",
   "   iqr=q3-q1\n",
   "   upper=q3+1.5*iqr\n",
   "   lower=q1-1.5*iqr\n",
   "   file[i]=np.where(file[i] >upper, upper, file[i])\n",
   "   file[i]=np.where(file[i] <lower, lower, file[i])"
  ]

},
{
"cell_type": "markdown",
"metadata": {
"id": "gGdfUY_EuWWp"
},
"source": [
"##After removing outliers, boxplot can visualized as below"
]
},
{
"cell_type": "code",
"execution_count": 54,
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 447
},
"id": "Aw5XzYXgu-kP",
"outputId": "ebe27ba9-e0d6-4c3d-cc86-0738eed53925"
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"# of Bivariate Outliers: 19\n"
]
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzde7xlZ10f/s9z7nPmmsxMLgZCwEwQEpXLgPrSIjctKib2J+XSVkVpqVU
x12Wt913Nba3+z935KrTUAAAAtNfQuQ4AAAAAgC+OBA8AAABAy0nwAAAAALScBA8AAABAy0nwAAAAALTcyLkO4IHa
BAAAAaDkJHgAAAICWk+ABAAAAaDkJHgAAAICWk+ABAAAAaDkJHgAAAICWk+ABAAAAaDkJHgAAAICWk+ABAAAAaDkJH
JHgAgL4OHz6cv/r4J7MweeG5DoUWGp46niS5a8bjJpszPHXfuQ4BoNXccQGAgRYmL8yZL/vmcx0GLbTtb96eJNoPm9a0G
QAAAGg5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGg5C
AAAGg5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGg5CZ5z4A1veEPe8IY3nOswAAAA4LxwPrwPHznXAZyPDh+
AAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4A
AACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5UbOdQDnozvuuCNnzpJq1/96n
Km1vinJm5Lk4MGDrf9fiZdddlmS5PWvf/05jgQAABnv1q1+dm265+1yHAZwnFid25crHXewZGdgS58On0OnA1vxCR4AAAAABp
g5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGg5CR4AAAC
5CR4AAACAlpPgAQAAAGg5CR4AAACAlpPgAQAAAGi5kXMdwPnoyiuvPChAAAAwHnjfHgfLsFzDrzyla881yEAAADAeeN
AAAALSfBAwAAANByEjwAAAAALSfBAwAAANByEjwAAAAALSfBAwAAANByEjwAAAAALSfBAwAAANByEjwAAAAALSfBAw
AAALSfBAwAAANByEjwAAAAALSfBAwAAANByI+c6AADg4Wt46r5zn6swaKHHhqXuTRPth04AA6C1JHgAg
nwAAAAALScBA8AAABAy0nwAAAAALScBA8AAABAy0nwAAAAALScBA8AAABAy0nwAAAAALScBA8AAABAy0nwAAAAAL
7e1e2LsHD1iiIHKq1HjzXccAjhT4FDy59Ch5c+hQ8ePQnHi58RQsAAACg5SR4AAAAAFpOgofGm851APAIo0/Bg0ufggeXPg
3NI+xbmMGx6OSimfK6V8rJTyV6WUQ511F5ZS3llK+Uzn3vo6feqpjpZZzWnSnnNvo4eGnlLKnlPK7pZS/KaV8spTyNf
UmOddb/185+wFrPqrU+qdZ6sLP8mi6LP8miTvqrUeSPKuznKSFfOSA53PTR
I625/T2R9YX3ff6e1Tv1aXvD/JnlLKpeciQHg4KqXsTvKMJLvUW2Vrr/dGn4MHwnCR/W2u9NfoUfDFGkmwrpYwkmUxy
x5+/lPtXR3d+A5LFJjib5lc5XiX+xlLI9+hQ8GF6c5C2dv/UpOAu11n/7Ockt2UpsXM8yU3yU3xfoqHGQme80Ap5flJjtRabzrXscAj
VddI6X9IfEmS7Umed06Dgj4kePX5vk2L38dqSzz6OS3NH5+44j/T2
Uvy+1I6j+X9FA8rEjzngVrj9ZaH1VrvTijfR/N9N9d631HyZx7VbT0/[illegible]
M8rBTt7PxSSnlmkh+qtT6/lPK4LH2i58IkH0nyj2qtM6WUiSS/nuTJSe5L8uJa6y3nKmZ4uOn0nT/oLI4k+a1a60+UUvYmeV
zn4KzVEq5PsmLksxn6b3TP87Sb+14XDhgQPAAAAAQMv5ihYAAABAy0nwAAAAALScBA8AAA
OHsj5zoAAICzVUopSf4f4gyZtrrS/urPvKJBcn+XRneaTWVO9Aj11r/7BAw
eW9pZSDpZT/mGRb55nk0iT7n6hc5rnlZK+YtSyl+XUj5YSl+XUj5YStIZSpkopfxKKeEVjpZSPlFKe1dn3paWUG0pskvXUG0sp
cneWKWPvXziSS/3H2QWutrSimvqLU+KUlKUrTuS/Hgp5deDLfUvCwbkvDEzh5e7r8UxhgEidlh72lulpJ1K4Zm4ZJ23ejVR0Qw9lV0Wc2pJWqHpT0A5CrbMVlf/fo9e2qgYLSydqpt7sNiluZ9Gvu6zFBycBa1YVh6Ltlo4ahMv4Sg9yBAx90JJwoCgsPl/ylntZhCJr7HIl7V4dVLkEykxvswaI0uWa3
8I8lbaq0LSb7Q+TTNumqt03yd7L0CaK3llJek6Vh919juR8spQWQt3ZmJwO52q1vymI3JqkSfC8s9Z6X1eM15ZSfqizPb
7bXWT51NTADA+c1v8AAbfuJOOdr2AlSUopXw5aspKgZKGe3sc1UpZXUs9yV5Uec3Ouaks8ei7Nx0idy+pnreu3jSykHur

zxt68j9G3YgI89brx/u0Xqxzce2Xqzx2Shozkb+sXSuYt1ydbO8FB9hevHPwxZ/a3q68Kctu612un/DDtcvgnzaxMOtnjXJZh/v
JNud9W04PtOj/Ix9+nYyvML7w2teW+ebPq8ZULjq/Q6a8tOc7kjXhUsDZk3KD12x97tz21LXTd9Pq2de3T/tmYilZcN9t0arN
vxKz6RPg8PHLbdFl+/dBttZm5Dz/SOJS/wfMxAF9E6UeWr1NKnVmtz6n8GzxA9Spae4bTmN14cKpohfXlP8b8ern/HO9x7
OKsBG3cPY1RXONg/mkVHUwrT651qC/FoBKMZp0qSqZih3kYVrWyugEQ86o7f1dE84SpaEFRUM/Pax1tFa+9IGjPrS1W0
7tVLgbGcphRn0AyJigqYDSdRzwWwcB4DjOmJXCi51f358ychkINVbSMXQQFFFTEragwOJ5Da0MSBU8VLTPnnK7IYKpomf
4gY2QzFYeMHsayebQ3JTGULrjVwUyFrcAqWp4KP6aKVjQSxXjWqfowu6yyjlP9ZVxXUHGraOl5jefyaKlPYjxbWUXLVF0xuv
YWbUufxUto2tj02w+j0Qshv6RDI5tdiof7RxM67nmEYvG3Oooxj/MuEamsWwOTakEIiLoH82gOVWqfmIqphi9+6tomSog
EQH2jWXRlCr5tKkkY85xq3jp8ZUqQHTfbY1OhZvduoqX0aWxg+nLX0XLxOces1bEKqtomUoy/rw5oSpavspwE6qi5TvX5E
yuqaOmqXaaio7+K1mg2h9b6pK4caa8SZWQxOjbtJrYHxnJob0oi49Glv4qWGRcoAohUVJkaTjtrEVDKD8Zn/RXy/H5p2vtH
Xs4q+itWfYybUFVVlFy8R4RRUtLX9ZFa1sriyWGpKCkUlX0XLsbmxl5mHypKmiZWSBUphZH8eBsTwG9DH+Klpm238N4P
rtU5DMo6IAIWiQs5jF+99ibWK1mgGs+qTmNUQRf9IqSqqqaJldAwA7c1OFS1T8bfLraKVK6uGOqDzkfe6tq0phdlNCYyk8x
TxWtsnsEXxWtwfGsWznTL7epyJXNK+wfzaK9OYkInIpo7c0pHN/aeERW0XrXf2RZRG4HcC6AWQB2A1gNIA4ASqmf6TLpF
gghhBBCCCHksIcPeAghhBBCCCGEEEKmOO/ojyy/E4hIP4Dth3oeRyCzAOw91JMg5AiCMUXIgYdxRciBhzFFyIGHMUXIgYdxRcjBhTFFyMGD8UTeb
QQMsXhAx5iuOFQT4CQIwzGFCEHF8YUIQcXxhQhBw/GEzks4G/wEEIIIYQQQgghExx+AYPIYQQQgghhBBCyBSHD3gIIY
ir+rPGbpdROQ6HVMbROS0Qzt7Qg4/RGS6iNwpIi+LyCYR+QBjipDJISLH6/XJ/BsSkb9jTBEyeUTkSyKyUUUReFJHbRSTF+ylyu
CeCn7/pMCTn8+SGA3ymlTgBwCpz1ijFFyCRQSm3W69NiAKcDGANwNxhThEwKETkWWwCoAS5RS3QCiAC4D76fIYQYf8B
sCYIuRgsAzAa0qp7WBMEfJ2iAGoE5EYgGkAdoL3U+Qwgw94jh5+AOCrAIp6uwXAgFIqr7ffBHCs/n4McTQkc
A82BkEsB68nyKHGXzAcxQgIssB7FFKrT/UcyHkCOJspdRpcF5r/4KInOPdqZRZCF5r/4KInOPdqZRScB4CEUKqEwNwGoCfKqVOBTCK0p+OAGE
fAeA4AND7mwHsezcnTMjhjv6fHCil9sD5XYMzAew2r7Trzz36cDemNN54I4Q4/+v5plLqab19J5wHPowpQt4eHwHwrFJqt
f0T/Lw8hBICI1ItIo/kO4AIAL6l8dvwx9WldpeT9AAY9r8gTctSjlNoF4A0ROV43LSxdPoHhEhThTThLxdPoHSn2cBjClCJksfgPeLyDT9
tabMQD/Vyn1TyLSAuAOAHMAbAdwiVJqv74QD/Fyul0o9BFMn5LBFRBbDKQSQALAVwOVw/iOKMUXIJND/Ad
IIIYQQQqY4fMBDCCGEEEIIIYQQMsXhAx5CCCGEEEIIIYSQKQ4f8BBCCCGEEEIIIYRMcfiAhxBCCCFTGhFpF5HfiMhrIrJeRH4
02ykA2gC8ordjSqn8RPtWSv21Z/MbAP5Z9/cBAMsBnKaUyojIlLACJtynHpOZICCGEEGLgGzyEEEIImcqcByCnlPqZaVZaVBKvQ/
1Fv6nDNE5EkReUFE/iwijSKSEpFfiMhfROQ5ETlPH/tEbIXRB4B8LCl1IvIzfq850RkxbujQkIIIYQcCfANHkIIIYRMZboBrA/Yd
E4p9f9f9EJAFgHYBLlVLPiEgTgHEAVzvdqJNE5AQAD3j+hOw0ACcrpfKFJq9OCoihBCCCFHMnz
AQkbMB/Ei3vSwi2wGYBzwPKqX2e+bYlyJf0dspAHMAbJqc+IQQQgg5muADHkIIIYRMZTYCuDhgn/fDn/fNFwFwlVLq994+OmZs7Evh8
i+piFIlP4HEAI+rf6OmA80aOjZzn3ONFpNOzbzGA7QA2A+gQkTP0cY0iEgSvH9hDn9wCu0j8cDRE5tVYlEE
AN8D8CP9OzI5AFsAXKmUyorIpXpfHZzf3zkfwE/6ZMH8Fldfcs/xjcB/ECPEwGwDU61LkIIIYSQqohS6lDPgRBCCCCGEE
jh8wEMIIYQQQgghhBAyxeEDHkIIYQQQgghhJApDh/wEEIIYQQQgghhExx/j/c4vCNhc7TPgAAAABJRU5ErkJggg==\n",
     "text/plain": [
      "<Figure size 1152x432 with 2 Axes>"
     ]
    },
    "metadata": {
     "needs_background": "light"
    },
    "output_type": "display_data"
   }
  ],
  "source": [
   "box_scatter(file,'CreditScore','Exited');\n",
   "plt.tight_layout()\n",
   "print(f\"# of Bivariate Outliers: {len(file.loc[file['CreditScore'] < 400])}\")"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 56,
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 447
   },
   "id": "oEvxBoL3vhd9",
   "outputId": "0a70fef8-9c94-4bba-b296-861fe4ab0a93"
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",

"text": [
"# of Bivariate Outliers: 0\n"
]
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3df7DndX0f+ueLXXAR+VFgu5Bd4uKAenFV1HOpVmutqRk0XuA21EKaj
/vee95PGaY8/382Nc+53De7HeffD6fb3V3AAAAABjXYfMOAAAAAMDBUfAAAAAADE7BAwAAADA4BQ8AAADA4BQ8AAA
AAINT8AAAAAAMTsEDAAAAMDgFDwAAAMDgFDwAAAAAg1PwAAAAAAxOwQMAAAAwOAUPAAAAwOAUPAAAAACDU
8AAwNdu2bcsNN96S77zw+HlH4Tla89B9SZKdj3jLwOyseeieeUcAgOF4twbAVH3nhcfn4Ze/Y4/xeI6O/No1SeLfITP1M1xM8
wAAAGBwCh4AAACAwSl4AAAAAAan4AEAAAAYnIlHAAAAYHAKHgAAAIDBKXgAAAAABqfgAQAAABicggcAAABgcAoeAAA
fgAQAAABicggcAAABgcAoeAAAAgMEpeAAAAAAGp+CZg8svvzyXX375vGMAAADAqrAa/h6dt27bbNOwIAAACsG
AGJyCBwAAAGBwCh4AAACAwSl4AAAAAan4AEAAAAYnIlHAAAAYHAKHgAAAIDBKXgAAAAABqfgAQAAABicggcAAABgc
AAAABqfgAQAAABicggcAAABgcAoeAAAAgMEpeAAAAAAGt3beAVajHTt25OGHH84lI1wy7ygAU7Vt27bbNOwIAAACsG
LnsssvmnARgui655ARy6Jsve3OeccABvP4umnNy2ks2eK8EwPNmVPVAAAAAADsm4lHAAAAAYnIlHAAAAYHAKHgAAAIDBKXgAAAA
IHAAAAYHAKHgAAAIDBKXgAAAAABqfgAQAAABicggcAAABgcAoeAAAAgMEpeAAAAAAGp+ABAAAAGJyCBwAAAGBwCh4
Gt3beAVaj0047bd4RAAAAYNVYDX8PV/DMwcUCXXXzzvCAAAALBqrIa/h7tFCwAAAGBwCh4AAACAwSl4AAAAAan4AEAAAA
Ch4AAACAwSl4AAAAAan4AEAAAAYnIlHAAAAYHAKHgAAAIDBKXgAAAAABqfgAQAAABicggcAAABgcAoeAAAAgMEpeA
u2aecfgOVrz0N1J4t8hM7XmoXuSbJh3DAAYioIHgKk57bTT5h2Bg7Rjx2NJko0b/WwWNw0b/WWbWdrgvx8A8CwpeACYmosvvjnjeEQ
AAgMEpeAAAAAAGp+ABAAAAGJyCBwAAAGBwCh4AAACAwSl4AAAAAan4AEAAAAYXHX3vDM8K1W1K8K35p1jlToxyZ
gFDw8G1fMOwBgHcIKYB3C/FmHMH/W4QrjGTwAAAAg3MFDwAAAMDgFDwAAAAAg1Pw8DRVdUpVXVd1fVTV1y
qKrFyT7vS2GGquq4qvp4VX2tqm6pqjdYhyuPgoe9e9eSzUIen+QnQuqjMJO9L8rnuPj3J5ybbbwHQ8kuSt3f3qJGcmOb
9HdX5y8fiBLi3djknOTfGxy2seSnDefhHDo6yV/Ptk8fPJPJ/Ju1ksM1ECIxVR2b5M1Jfi1Ju
NLSX42yeOT7RNiHcKsdZLfr6qtVXXXRZJ/3pTA7pybZlefTG5Z/KhVHRXrcMVR8LBPVfWiJ9LPPdff/yY93dWfrDFpiS7v5Od
4Mih42KuqOjxL5c6c6/6+7/MNl9Z1WdPDI+cpauKgCmbHIJ7HVJ3pDkuKpaOzm0KcmOuQQWuQWtQWDQ98bk5xTVbcnuTJL
1lfVcZPXXZRyZ5W5aeh3vdkvMnp1mHMEXd/XPdvam7Nye5IMkfy+9Q9aqzDFUP
7J0JcHxSb6U5Ie7+5HoeSy5Ij7+5H5JYXVoarekuRnuvd1iHMzmS9fxCjNTVWdm6QMjkhyW5Ify+Q9aqzDFUP
D4bhcm+e+TrwAAQ1DwAABMVNWLkrwpyfSxAh2qrq2qa2qa6rq/Mmx11XVf62qrVX12qaeo6eY7xAYBVTMEDl
p7seT7Kyq6ybHX5ZkS5JrqypJ1iS5Y7Y7ZxAQCWKHgAAJJU1fFJ3prkprknlVXVWpsOskn9/VLktzU3W+YUQAgH1yixYAwLz
X0xyX3c/mqVS6INV9eUkNyT5y5y7OLCwDwlOruueWcAAJ3gGDwDA/n26qo5LSkSSX
Q8AAADA4BQ8AAAAAINbO+8Az9aJjZymdvnnncMAAAAgJnbunXrn33n33n33D1fwbN68OYuOYuLi/OOAQAQAAADBzVfWNve1
q6u29edtp7kvxZd59WV5Rck+WCSvzWtCvBvQ/vzq07H8yd9z+S6F+S896agcd+S6FTt31Nkyjz9b9b5tnMHjHzNGfLPP5sm
uyvuvvvvjG7v/141h1+WC49Z0u+f8v6ZS/xve+/5Mkfqifs/ZR5/tszj29t/izZR5/tszj5Z5/NkyNz9b5tnKXmnmmWfB8IcnpVXVqR5V5IIkV+9ztVJ3Vn
f7bM48+WeXazV5ohHrJcVRdV1WJIcfVb6fpm3xnGCxTnzz+bJlnM3vEzNOcLFP4s2Uef7bM4
V93+GHZcMwLVuTcUWfLPP5smWcze8TM05wt8/izZR5/tszjz5Z5/Nkyjz9b5tnNXmmmWfB8lcnpVXVqVR2R5lIkV+9xztVJ
l2s1eammafUlXvSPJLSdYk+Wh3f6CqLk2y2N1XV9W6WJ+R5DVJ7klywMPFZd6XhYWFcxcrnFrmafPE8dnMlnn82TLPZvaIm
MTsEDAAAMDgFDwAAAMDgFDwAAAMDgFDwAAAAg1PwAAAAAxOwQMAAAAwOAUPAAAAwOAUPAAAAACDU/AAAAAAADE7BA
AACDU/AAAAAADG6qBU9VnvqvmVjPqmqmvqVlJVn6uqF08zdPMOBD1E0+BQ8MQPqLnEtzzp2dLzZHm+J9LzNdl
Wx7+2Tfvrwnye/t7/6C0VjwDHTvDbz3Xr70P1vbvXlaBhl1MFqtfgCSXvmi5SjqZLF1tZ+gBThlZk/q
U7Zp7j1+nZ5Nc7DKgm9QBP6yn3OS9rgbn2GdntmMkicDvWGZBz/o2Uy2DhQvlncDP07O+sHJm
9xXdvdDdC+vXr595SAAAlCVbJoFz44kpyzb3j7TZt7x5klyVJN3N9P0Os9S3LiFDMBAAAAHHKmWfB8IcnpVXVqVR2RpYcoX73
gEAAAAcitZOc3h3h3X5Olhyv3/f+Za9vTvLGaWaWGaWYAAAAONTN+yHLAAAAAAAANTsEDAAAAMDgFDwAAAMDgFDwAA
DA4BQ8AAAAAINT8AAAAAAMTsEDAAAAMDgFDwAAAMDgFDwAAAAAg1PwAAAAAxOwQMAAAAwOAUPAAAAwOAUPAAAAwOD
l8wlH5bDD6nlIPL3MSfLoo9/JV751X3bevzsnH7Mur/yeY3PEEWsOeu69D+/OrTsfzJ33P5INx7wgLzx0ZwAGzVhXHy0MO
Lyf59X0cf3uS0yf//KUkvzr5esh67LHH86kv78jPf+/rG7P7241l3+GH5p+dtyvmVhOfnnyjc78jPf+/rG7P7241l3+GH5p+dtyXmv3nhnQcnjjj3c+c9PO/NRVNzw590PvOjNnv+
U+b/c4tGw6qMBkx8zRzT/P7IfPYmac5W+bxZ8s8/myZx58t8/izZR5/tsyzm73STO0Wre7+wyT3PMMp6Oq6

nWwc2+deeDTy7eJ+defWNu3Xnw348bdz6w19k37nxg1WVOppd7mt8PmcfOPM3ZMo8/W+bxZ8s8/myZx58t8/izZZ7d7
uO+8/2O/HI/uY+8hBzZ3m7BEzT3O2zLOZPWLmac6WefzZMo8/W+bxZ8s8/myZx58t8+xmrzRDPGS5u6/o7oXuXli/fv284
bGaPmHmas2Uef7bM48+WefzZMo8/W+bxZ8s8u9krzTwLnh1JTlm2vWmy75D1ipOPyT89b8uTP1zrDl96ns0rTj72oOZu
enoVZc5mV7uaX4/ZB478zRnyzz+bJnHny3z+LNlHn+2zOPPlnl2s1ea6u7pDa/anOTT+/gUrR9I8t4sfYrWX0ryr7r7rP3NXFh
nTnD1i5mnOlnn82TKPP1vm8WfLPP5smcefLfPsZs9DVW3t7oWn7Z9WwVNV/z7JW5KcmOTOJL+Q5PAk6e7/d/Ix6b+c5C
tSvKNeedYpU5M8qfzDgGrnHUI82cdwvxZhzB/1uH8vLi71++5c7iCh/mpqsW9PcgJmB3rEObPOoT5sw5h/qzDlAAA
VXVzVd1UVZdM9h9fVddW1R9Pvv6FeWeFQ1VVrauqz1fVlyfr8J9M9p9aVddX1baq+u2qOmLeWeFQV1VrqupLVfXpybZ1
g1MxyNJ3trdr05yZpKzq+r1ST6Y5F9292lJ/izJe+aYEVaLS5LcsmzbOoTZ+2vdfWZ3L0y2vS+F2bosyWe6++VJXp2lPxetw4JM
hTVlWbkvxAko9MtivWIawE3pfCjFTVsUnenOTXkqS7H+3ue2MdrjgKHp5RVW1O8pok1yfZ0N13TA7tTLJhTrFgVZjcFnJDkr
XHAUP+1RVL0ryiSQ/2d33Lz/W3Z2lP2yBKenu73T3mUk2JTkrycvnHAlWlap6Z5K7unvrvrLPAKvem7n5tkrdn6dEBb15+0P
Q5uS7JhbMDj0vTHJOVV1e5Irs3Rr1mWxDmGmunvH5OtdST6Zpf/p4X0pzM72JNu7+/rJQqXjbZ9X1Jbo51uOLU0pVU8JSqelOS/5bkq3nqmQP/MEvP4bkqyfcm+UaSd3X3PM
6I96UwM1V1ZpY+cOCIJLcl+bFM3qPGOlwxFDwAAAAg3OLFgAAAMDgFDwAAAAg1PwAAAAAxOwQMAAAAwOAUP
fHHtdVf3XqtpaVZ+tqpPnGB8AWMUUPAAATzk3yWe6+9Ykd1fV65L8jSSSbk5yR5EeSvCFJqurwJJcnOb+7X5fko0k+MI/QAA
wqaTfHJfvyTJTd39hhlFBADYJ7doAQAsOT/Jb3T3i7t7c3efkuRPktyT5Acnz+LZkOQtk/O/nmR9VT15y1ZVvWIewQEAFDwA
JI3dvfOeecCAHiCZ/AAAOzfp6vquCRHJPlF5Q4AsNK4ggcAAABgcJ7BAwAAADA4BQ8AAADA4BQ8AAAAAINT8AAAAAAMT
"text/plain": [
"<Figure size 1152x432 with 2 Axes>"
]
},
"metadata": {
"needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
"box_scatter(file,'Age','Exited');\n",
"plt.tight_layout()\n",
"print(f\"# of Bivariate Outliers: {len(file.loc[file['Age'] > 87])}\")"
]
},
{
"cell_type": "code",
"execution_count": 57,
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 447
},
"id": "eJiFJvJAvurd",
"outputId": "24f4fda6-a1db-41cb-a9a8-6f3e9fa3cd3d"
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"# of Bivariate Outliers: 4\n"
]
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzde5wcV3nn/+/TXX2fi6TRWJbli2xLAiyZcFEcQrJAfMPws2QnEGJI1mwCIZ
RK+rxfL72mu+pU1VNV55yqflTdx5xzAgAAAAAAQHhF+h0AAAAAAAAAAXhgSPAAAAAAAAACFHggcAAAAAAACDkSPAAAAAAA
7203nK1oAAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAAgJAjwQMAAAAAAABByJHgAAAAAAABCjgQPAAAA
AELO63cAp6JPfOITGhsb63cYAIA+2LdvnyRp1apVfY4EvbRmzRpdd911/Q4DAACcQkjw9MHY2Ji+/hjqqSX9TsUAMAii+Yr
QAAAAAACDkSPAAAAAAAACFHggcAAAAAACDkSPAAAAAAAACEHAkeAAAAAACAkCPBAwAAAAAAEHIkeAAAAAAAAEKO
AAAAAAQo4EDwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAAhBwJHgAAAAAAgJAjwQ

IjTvL1c8/fHgjTls6txpfXNkvP20Qv31uK95z0kJsHbdaxnnK4ADue3EbXhkto2h7sdf9vhe34cXhAn74yKbYepzfk8ZE2cFn/9+
Z6rHWH3OEL55/7pqXTV+vuqMJTBVBZ//1XPVY1x26mBTHIfMSuG7D75UV8cpXcE//e3R0BQFn6k559Urh/D1+9bBdmXT
1CouHVxXLNqGe7+0ys44Yh+fOGu56vLP3HWUejLGfiH256qOz4g8bX7/ox3n7AAX/rNC+jJGPhfb17U1DauWbUMZdvFD
1dsWIQWUPFE5tGcdKRffh0TZ3Vtq0rT1+CtK7gn3/5XFMMbzv+8Lr2VNtP5s9KI2OquOyWPyaeP2dqkFLWHfuKFYNY1JfB
PbcJFbzwCIwUb//Kr+ljnz0rh5gdfwqMbx5HSFXzqLUtRdnzc8vDGpjivPf+1KNteXfusreu49n/l6Uvw2MYdWHH0PHz9vnXx
Ch6C7Sw41T4Qd83nz0rhi3e/UD3uFSsGkdFV3PjABrznxAX4j/9+CYYm8L9PWVytpwW9aXz09CVN43Lj+BTFnTUVWHZzjF
OFWyfbxsZ+sbTrfu09YgFse3oh3vuHwavv41FuWwtTVuvbzybOXYnZOR7HiN/XFqI9+83fN7aa6Powt6f78xF924pzXzsfml
59fgtOXTqvqf9GbTO6J3313hfryrigN40Pn7w4sWxrVi3Djx/ZWO0zHz55Mb55f3z/iuo/6ec1q5bha/e+WNema7db1J+F6w
oOLKubhrHnrj9kpbtydx4On436bTffabbgVL+A6UctYQQj0kplzct31sPeMKTLgRwZ8IDnm8BuE9K+YPw8/MATpFSbml1zE
3Ou3F/DzJzbj+vvXNx3nR5echHdc/9Ae18fu1mer7QHUrbv01MW44ffJMUd1EFc3X7t3XVvbfuH8Y/HxW4PJa9L5on0vOXk
+9fn1g3l5y8CADwlXvWNcVQGysAXL5icWJM0bW45ORF+Mo963DpqYtx9Nw8XmgoR7t11fg5Om6rY1xy8iIsHsjvVtwAmc
Qlvm1ccvIieD5argeA5Qt6MFZ2m+q0sY20astTbXPJyYtw0qLelnWWVOa4dtrYT5Yv6MFFNz7S8vxAcx9Iur5R3bXbbq6/8H
GqfpT4/W6/sLjcUnCtV2+oCd23Xc+cAIe+POOluNPu/eR2vEwqZ984fxj0ZXWYmNpHO+mah9J5Y3OndRu2hkjWo2RjXFee
U7v38ahvJrW1pPZcO+eZqv/X9tepxsSk/ZKW7cnceDp+N+m0332m24FS/gOlHLWSHvDM5Dt45gP4S83nl8NlTYQQlwghH
WVMijlpvRDtb1u03eryVsezHL/6Vd6pyrK1ZvlUMUafyxW3rbKXbTfxWtbGuWU8uOatrnu0LuIc0f5NMTTECiSfp/ZaRMcSA
TrfQmMlpzYttDYRlq15am28SWmrLOkMifFVttPRkvOlOePrc+EY0d11267GS06iccaK7V/fVuNC1Hbb4xld9t3q/VRW5yq/w
nfjVjRKsxsjHOsu3G9rekco6F40OrMiT106hOGmOINI7HSeVvdYzGc7Xzc9y6pDEoaZ4ZzX2S4mk1f0yaN7V7H5+qrSW15
trFvKnbfgXx77++ZLvtLHO1KilfK9uOe05WCKhB7nHnd6VdVH7tbn622T1rXHPOu7ZLqJq58cdtmw3dXtDpfVNeKQF25ku
b7232wq/nr25+PH0LgypnQFPVk98ZyzsnpiWRuXZQ2tqTxJx432b7VeEUBPRk9st43Lk9ryVNsoAlPWWVKZk2Kr7Sc9GX3
t3qv5Uu6ynRd+dnbCuP598H49ia/c+oghUt03aJ2toibE0jndTXbek8taee0/HiNktxsjGODOGFtvfkso5KxwfovPsTj9tbENJ8b
ffabbgVL+A6Uc7ZjJBzybARxW8/nQcNkBa8ncLNasXFY3sK9ZuQxL5man3HdhbxbXXXBc3b7XXXBc9UVk+8r+Eke74uK9/
/OWm8sVte9UZS6AoqNbLbY+93FRHUV1fsWIQrz20u65cSWV57SHd1bjjjvnZ85bhzic3132+6YH1dfHFlf3y0wZx0wPrsbA
s1Yfe5Q3TFWnzNUV1eNn686Ywl6M0bdMeLiOLI/21THKT144PKZhnNevTI4R9y1m50xgndYxNR97TUoWU7TdledsQQ/
9r4I61mzE707z+qjOWoC9r4Mj+LMZKFdz0wHqsPmeoqT5uemB9tZxx9bj6nKG663HbYy831dcVKwbRlzXwk0c2YU1Dnd
uXjnUFM9nzh3CzQ+sx5EDOXzy7KXN/bg/i5sfWF9d1ps1cOXpS2LjPHIg19Q+a+s6rv1fefoSfP+hDbh65VDyWFITwx1rN2N
WIKfPv4y7li7ua6e7li7OXZcbhyforjHynZsjH3Z4AXqje1hzcpl+M0zQdmSxp/V5w7F3kcajxX14Wgcj+vDq88dgqIAnu/ji28/N
ThWqjRdj6tXBn2w9jxrGu4tq88Zwvcf2hDbf6M2FN2TGst4x9rNLcu2ZtWyuj6z+pzk/hXVf9LPa1Yta2rTtdvpqsAX397+PD
r5D5RytGMm38HztwAuw66XLH9FSnnCVMfs5HfwANOTRWvbZPDG+5nOojXTcbSrPouWCsfzMftVZNEq2S4Oj8mitaf1sl
OohVknDI1BdLHrixano+ejIFCxYE+RRYt2/WbrntzFi0HKV0JslFlNOyctFG0PRQrHgbyBoQQ2DphoTfKojVhoS9rImMmZ9H
aQ2W7WN7oYKsqcHUBLrT9Vm0guxNbWTR6krBcn0UoixatoeUrkBTFIyWHORSWlh3QRYtI8yiVbAcdKcNlJ2g7uZ2m8ibC
md1mGqQRSvIyGLA1BWUwyxaru8H2VQcHyXbQ84MMsVEWbQyelB/RduBqdZk0QozM7l+kCkrMYuWoqJgN2fR2jZRQ
u0EWTgirJoFcMsWj1RFq2ciYrnY7zsoKuaRSvIPmZoCiqOjwnLRdpQYWgCWT3IVqOEWbRGijYG8iZMTcGm0XLw7ViJarYgI
WLUODhAc/zKIVlG9XFq1yxUNPVg+yaFn1WbQcV6JQCdqmJz0oIsj0lzM15E0NkxUXJdsLsphZNtKaGl4fH74PjJSCbXVNwF
u4IsWj0ZAxlDxXjJRleYRSsqj6oAaU2Foddk0QrbgyKC/i7hQ1dUlJ0wi1YqaD+aqiBvaCi7Hkq2i5wZZNGanTUhwixaJTvIQpU
+ZxC044jhcqQd8phVm0fN9HSg/6eV0WrbSOcas+i1b0J3hzu4J+GIwJQT0cPacLL4+XMVIM7kkFK6jfnKkh0zKL1q45V10Wr
CbNeBKH2NFF/O6TSwNj7Vt0kJ/ro0sWmHmsHltZNFqnLfsThat2nkTgKa538aREjaNFJGpyaJVO7+r3ac/t2v+N7cryKK1vdA
/GaZJ/yqAsxGkSf+AlHLKJzed/oCHiIiIiIiGhPJT3g0eI2ng5SyndNsV4CuHRvnZ+IiIiIiI6GAxk+/gISIiIiIiKiacAHPERERERERE
wo57wHOgEEI8GvdSJKKDGfsFUTP2C6Jm7BdEzdgviJodbP2Cf6JFRERERERNTh+ICHiIiIiIiKjD8QHPzLl+pgMg2g+xXxA1
kI8L4RYJ4T45EzHQ7Q3CCFeEkI8JYR4QgjxaLhsthDibiHEi+G/PeFyIYT4StgnnnRCvL7mOO8Lt39RCPG+muxXHh8dfF+4r9n
JCGOFyM/y8LIy/cN+UmGhqQojDhBD3CiH+JIR4RghxRbic9ww6aLXoF7xntMAHPPuQEEIF8DUAbwHwGgDvEkK8ZmajItp
/SDoH0f7gO4gfs78U3jOOk1L+AgDCOdI7AQyF+3xdCKFOMZ/61/BYiwGMAvhguPyDAEbD5V8Kt
UspRAHcDODtc1yWlfEgGb4m/ueZYRPsNKeX9AEYaFu+Lfd0Dqdj5VPiNhNMA3Br
MvmSCm3hD9vBTAn/DmpX7Ra/nLMcqJOsC/6QdI5iPZnl4V/anJjzTcOdrdf9AIYk1K6DcvrjhWuHw+3J9qvhH8K8joAfwDv
TIAbgNwEeIlBO163jPoINVTL/gPaMFPuDZtzYDOKzm86HhMqIDipRyc/jvNgA/Q/DVyOHwK8II/90Wbp7UL1otPzRmOVEr
zWuMDnn3rEQCD4du6DQQvgbp9hmMimlZCiKwQIh/9DOBMAE8jaOtRNof3Afj8OfbAVwUZoQ4CcB4++FXhXwM4UwjF
fnE+gN+G2xPNuHAcvwHAs1LK62pW8Z5BB62kfsF7RmQ+I8YIggjdu/AVAB3Cil/NwMh0Q0rYQCs/CyFOBuADm4ANAC3SCk/J4
AMwjCCzyc+xl/tBUl/b6wUmakNCvzgFwVftJYCXAHwoeieIEOL/ALgYQTaVj0opfxkuPhegYWDeCPAN4rpawIlIVIAvo
ERERERdTg+4CEiIiIiIiIi6nB8wENERERERERERE1OH4gIeIiIiIiIiIqMPxAQ8RERERERERERERUYfjAx4iIiIiI64AkhPCHEE0KItUKIx4UQb2
yQoh7wm/1PCWEWENWNW4cdI2QoiFQohnRhRDfFkI8I8S4SwiRDtcuOqvCEGkOEOLqtUbYEJPcXQY8UlMzYKI
IcQfAPyPyLlPPgBgUgUEGNSRBAM/F7NvONo0uBPAtIcQAADm0asHs9BNHMg/JmsEMFJpfu1kqrRDtUmahbmlMLZWkxL8Ajh9GSLMPQ8ZuPZ0TktMCyhz+I
ERETU4fiAh4iIiIiIiow/1/IB+Dzb9+iFAAAAAASUVORK5CYII=\n",
    "text/plain": [
     "<Figure size 1152x432 with 2 Axes>"
    ]
   },
   "metadata": {
    "needs_background": "light"
   },

```
   "output_type": "display_data"
  }
 ],
 "source": [
  "box_scatter(file,'Balance','Exited');\n",
  "plt.tight_layout()\n",
  "print(f\"# of Bivariate Outliers: {len(file.loc[file['Balance'] > 220000])}\")"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "mJ4Mtggyv9li"
 },
 "source": [
  "##7. Checking for categorical columns and performing label encoding\n",
  "Label encoding is performed to convert the labels into numerical(binary digits) values"
 ]
},
{
 "cell_type": "code",
 "execution_count": 58,
 "metadata": {
  "id": "5tURFqMxv39M"
 },
 "outputs": [],
 "source": [
  "from sklearn.preprocessing import LabelEncoder\n",
  "encoder=LabelEncoder()\n",
  "for i in file:\n",
  "    if file[i].dtype=='object' or file[i].dtype=='category':\n",
  "        file[i]=encoder.fit_transform(file[i])"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "eqYY_luRwjyf"
 },
 "source": [
  "##8. Splitting the data into dependent and independent variables"
 ]
},
{
 "cell_type": "code",
 "execution_count": 59,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 206
  },
  "id": "St9RQU8XwfiU",
  "outputId": "4a3aebae-574e-4b4a-d8b2-cffef974b582"
 },
 "outputs": [
  {
   "data": {
    "text/html": [
     "\n",
     "  <div id=\"df-20085f35-b34a-4423-a0e0-2d943da3fe30\">\n",
```

```
"    <div class=\"colab-df-container\">\n",
"      <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>CreditScore</th>\n",
"      <th>Geography</th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Tenure</th>\n",
"      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>EstimatedSalary</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>619.0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>42.0</td>\n",
"      <td>2.0</td>\n",
"      <td>0.00</td>\n",
"      <td>1.0</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101348.88</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>608.0</td>\n",
"      <td>2</td>\n",
"      <td>0</td>\n",
"      <td>41.0</td>\n",
"      <td>1.0</td>\n",
"      <td>83807.86</td>\n",
"      <td>1.0</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>112542.58</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
```

```
      "      <td>502.0</td>\n",
      "      <td>0</td>\n",
      "      <td>0</td>\n",
      "      <td>42.0</td>\n",
      "      <td>8.0</td>\n",
      "      <td>159660.80</td>\n",
      "      <td>3.0</td>\n",
      "      <td>1</td>\n",
      "      <td>0</td>\n",
      "      <td>113931.57</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>3</th>\n",
      "      <td>699.0</td>\n",
      "      <td>0</td>\n",
      "      <td>0</td>\n",
      "      <td>39.0</td>\n",
      "      <td>1.0</td>\n",
      "      <td>0.00</td>\n",
      "      <td>2.0</td>\n",
      "      <td>0</td>\n",
      "      <td>0</td>\n",
      "      <td>93826.63</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>4</th>\n",
      "      <td>850.0</td>\n",
      "      <td>2</td>\n",
      "      <td>0</td>\n",
      "      <td>43.0</td>\n",
      "      <td>2.0</td>\n",
      "      <td>125510.82</td>\n",
      "      <td>1.0</td>\n",
      "      <td>1</td>\n",
      "      <td>1</td>\n",
      "      <td>79084.10</td>\n",
      "    </tr>\n",
      "  </tbody>\n",
      "</table>\n",
      "</div>\n",
      "      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-20085f35-b34a-4423-a0e0-2d943da3fe
      "              title=\"Convert this dataframe to an interactive table.\"\n",
      "              style=\"display:none;\">\n",
      "        \n",
      "  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
      "       width=\"24px\">\n",
      "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
      "    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.0
-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.0
.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47
  </svg>\n",
      "      </button>\n",
      "      \n",
  <style>\n",
    .colab-df-container {\n",
      display:flex;\n",
      flex-wrap:wrap;\n",
      gap: 12px;\n",
    }\n",
  "\n",
```

```
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-20085f35-b34a-4423-a0e0-2d943da3fe30 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-20085f35-b34a-4423-a0e0-2d943da3fe30');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"            '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table not
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
"    </div>\n",
"  </div>\n",
"  "
],
"text/plain": [
```

```
    "   CreditScore  Geography  Gender   Age  Tenure    Balance  NumOfProducts  \\\n",
    "0       619.0          0       0  42.0     2.0       0.00            1.0   \n",
    "1       608.0          2       0  41.0     1.0   83807.86            1.0   \n",
    "2       502.0          0       0  42.0     8.0  159660.80            3.0   \n",
    "3       699.0          0       0  39.0     1.0       0.00            2.0   \n",
    "4       850.0          2       0  43.0     2.0  125510.82            1.0   \n",
    "\n",
    "   HasCrCard  IsActiveMember  EstimatedSalary  \n",
    "0          1               1        101348.88  \n",
    "1          0               1        112542.58  \n",
    "2          1               0        113931.57  \n",
    "3          0               0         93826.63  \n",
    "4          1               1         79084.10  "
   ]
  },
  "execution_count": 59,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "x =  file.iloc[:,:-1]\n",
 "x.head()"
]
},
{
 "cell_type": "code",
 "execution_count": 61,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 206
  },
  "id": "-ibQ_hq9w0T4",
  "outputId": "72b917af-fb7e-4a39-bb39-c922189324da"
 },
 "outputs": [
  {
   "data": {
    "text/html": [
     "\n",
     "  <div id=\"df-bed6d97f-0fa2-4be1-b21e-5b37b85302c1\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <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>CreditScore</th>\n",
"      <th>Geography</th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Tenure</th>\n",
"      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>EstimatedSalary</th>\n",
"      <th>Exited</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>619.0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>42.0</td>\n",
"      <td>2.0</td>\n",
"      <td>0.00</td>\n",
"      <td>1.0</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101348.88</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>608.0</td>\n",
"      <td>2</td>\n",
"      <td>0</td>\n",
"      <td>41.0</td>\n",
"      <td>1.0</td>\n",
"      <td>83807.86</td>\n",
"      <td>1.0</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>112542.58</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>502.0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>42.0</td>\n",
"      <td>8.0</td>\n",
"      <td>159660.80</td>\n",
"      <td>3.0</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>113931.57</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>699.0</td>\n",
```

```
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>39.0</td>\n",
"      <td>1.0</td>\n",
"      <td>0.00</td>\n",
"      <td>2.0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>93826.63</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>850.0</td>\n",
"      <td>2</td>\n",
"      <td>0</td>\n",
"      <td>43.0</td>\n",
"      <td>2.0</td>\n",
"      <td>125510.82</td>\n",
"      <td>1.0</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>79084.10</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-bed6d97f-0fa2-4be1-b21e-5b37b85302
"              title=\"Convert this dataframe to an interactive table.\"\n",
"              style=\"display:none;\">\n",
"        \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.0
-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.0
.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47
  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
```

```
 "    .colab-df-convert:hover {\n",
 "      background-color: #E2EBFA;\n",
 "      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
 "      fill: #174EA6;\n",
 "    }\n",
 "\n",
 "    [theme=dark] .colab-df-convert {\n",
 "      background-color: #3B4455;\n",
 "      fill: #D2E3FC;\n",
 "    }\n",
 "\n",
 "    [theme=dark] .colab-df-convert:hover {\n",
 "      background-color: #434B5C;\n",
 "      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
 "      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
 "      fill: #FFFFFF;\n",
 "    }\n",
 "  </style>\n",
 "\n",
 "      <script>\n",
 "        const buttonEl =\n",
 "          document.querySelector('#df-bed6d97f-0fa2-4be1-b21e-5b37b85302c1 button.colab-df-convert');\n",
 "        buttonEl.style.display =\n",
 "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
 "\n",
 "        async function convertToInteractive(key) {\n",
 "          const element = document.querySelector('#df-bed6d97f-0fa2-4be1-b21e-5b37b85302c1');\n",
 "          const dataTable =\n",
 "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
 "                                                     [key], {});\n",
 "          if (!dataTable) return;\n",
 "\n",
 "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
 "            '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table not
 "            + ' to learn more about interactive tables.';\n",
 "          element.innerHTML = '';\n",
 "          dataTable['output_type'] = 'display_data';\n",
 "          await google.colab.output.renderOutput(dataTable, element);\n",
 "          const docLink = document.createElement('div');\n",
 "          docLink.innerHTML = docLinkHtml;\n",
 "          element.appendChild(docLink);\n",
 "        }\n",
 "      </script>\n",
 "    </div>\n",
 "  </div>\n",
 "  "
],
"text/plain": [
 "   CreditScore  Geography  Gender   Age  Tenure    Balance  NumOfProducts  \\\n",
 "0        619.0          0       0  42.0     2.0       0.00            1.0   \n",
 "1        608.0          2       0  41.0     1.0   83807.86            1.0   \n",
 "2        502.0          0       0  42.0     8.0  159660.80            3.0   \n",
 "3        699.0          0       0  39.0     1.0       0.00            2.0   \n",
 "4        850.0          2       0  43.0     2.0  125510.82            1.0   \n",
 "\n",
 "   HasCrCard  IsActiveMember  EstimatedSalary  Exited  \n",
 "0          1               1        101348.88       1  \n",
 "1          0               1        112542.58       0  \n",
 "2          1               0        113931.57       1  \n",
 "3          0               0         93826.63       0  \n",
```

```
    "4         1              1        79084.10       0  "
   ]
  },
  "execution_count": 61,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "y = file.iloc[:-1]\n",
 "y.head()"
]
},
{
"cell_type": "markdown",
"metadata": {
 "id": "EiuLiNYtxPVM"
},
"source": [
 "##9. Scaling the independent variables\n",
 "It can done using StandardScaler from **sciket-learn** framework"
]
},
{
"cell_type": "code",
"execution_count": 69,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/"
 },
 "id": "jegp71flzfdC",
 "outputId": "8b0ed65c-326c-4424-ed20-063222f1a29c"
},
"outputs": [
 {
  "data": {
   "text/plain": [
    "array([[-0.32674984, -0.90196803, -1.09610816, ...,  0.9701425 ,\n",
    "         0.02177902,  1.97704053],\n",
    "       [-0.4406824 ,  1.51496305, -1.09610816, ...,  0.9701425 ,\n",
    "         0.21642782, -0.50580653],\n",
    "       [-1.53857802, -0.90196803, -1.09610816, ..., -1.03077641,\n",
    "         0.24058116,  1.97704053],\n",
    "       ...,\n",
    "       [-1.39357294, -0.90196803,  0.91231872, ...,  0.9701425 ,\n",
    "         0.02788069, -0.50580653],\n",
    "       [ 0.60542568, -0.90196803, -1.09610816, ...,  0.9701425 ,\n",
    "        -1.00875873,  1.97704053],\n",
    "       [ 1.25794855,  0.30649751,  0.91231872, ..., -1.03077641,\n",
    "        -0.12533935,  1.97704053]])"
   ]
  },
  "execution_count": 69,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "from sklearn.preprocessing import StandardScaler\n",
 "Scaling = StandardScaler()\n",
```

```
  "x = Scaling.fit_transform(x)\n",
  "x"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "iURc3XI00SEg"
 },
 "source": [
  "##10. Splitting the data into training and testing"
 ]
},
{
 "cell_type": "code",
 "execution_count": 70,
 "metadata": {
  "id": "_s2R5mKcz3YU"
 },
 "outputs": [],
 "source": [
  "from sklearn.model_selection import train_test_split\n",
  "x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 73,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "zYr5DfPG0rSa",
  "outputId": "bddaf063-b7b0-408c-a13a-1b684210376a"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(6699, 11)"
    ]
   },
   "execution_count": 73,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "x_train.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 74,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "erFRqg-60tlA",
  "outputId": "624fb285-82fd-4cda-9c57-52da492ac76b"
```

```
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(3300, 11)"
    ]
   },
   "execution_count": 74,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "x_test.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 75,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "1siWL3Zb040q",
  "outputId": "2408604c-db0a-48b5-da9a-89181d8763b0"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(6699, 11)"
    ]
   },
   "execution_count": 75,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "y_train.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 76,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "7iXQlG7906L7",
  "outputId": "16eaeda5-a3dc-4bbd-ba0d-bad37cbc4e73"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(3300, 11)"
    ]
   },
```

```
    "execution_count": 76,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "y_test.shape"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "wf_6NFji0-tW"
  },
  "outputs": [],
  "source": []
 }
],
"metadata": {
 "colab": {
  "provenance": []
 },
 "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": "3.9.7"
 }
},
"nbformat": 4,
"nbformat_minor": 1
}
```