

```

{
  "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.\"\"\",
" style=\"display:none;\">\n",
" \n",
" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\" viewBox=\"0 0 24 24\"\",
" width=\"24px\">\n",
" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
" <path d=\"M18.56 5.44l.94 2.06.94-2.06-.94-2.06-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06
-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.39.39.59 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\n",
"            + ' 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 RowNumber,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...)\n",
"    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-.94-2.06-.94-2.06.94-2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06\n",
"-2.06 2.06-.94-2.06-.94-2.06-.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\n",
".41c.39.39.959 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\n",
"  </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\n",
" + ' 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": "QtXUXIkZlZjw",
"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+WH4yJAAQ90IEQVR4nO3df5BdZ13H8ffHqKwigtrVwSRtMhh+REALS1AZpQNUU9QEBTRhUKq
d5znPvfjez2c+e85z7nFQVqR2fd+oC5AkjZZBIEmNMwgkqXEGgSQ1ziCQpMY9YNQFnK4LL7yw1q5dO+oyjOmccuONN3
8JMINSW5J8owu65EknazlEiyAtgDXA5sALYl2Taw7OXA/qq6BNgK/F1X9UiShuvyigAajMF1VR6vqBLAP2DlwpoAf7m0/BLi
nf5BVbUX2AswPj5+v++k84SXvPP+voXOQze+9ndHXYl0El0eERwH1vS1V/f6+l0J7Aeoqk8AFwAXdliTjGIA0FwEFifZF2SlcxN
1BVk8xNAvf37erbPgW8ucsajEkL85PFktQ4g0CSGmcQSFLjDAJJapxBIEmNMwgkqXEGgSQ1ziCQpMYZBJLUOINAKhpNEEh
APcBkwAxxMMtG7PSUAVfWnfeNfCFzSVT2SpOG6PCLYCEX1dGqOgHsA7YsMH4bczewlyQtoy6DYBVwrK890+s7RZKLg
dBsFBYH2SdUIWMvflfmjwUJJHAT8CfKLDWiRj8+gsCKrqJLADOADcBuyvqkNjdifZ3Dd0K7CvqqqrWiRj8+vs8IGAqpoEjg6
ziCQpMYZBJLUOINAKhpNEEhS4wwCSWqcQSBjjes0CjJsSnikyXSSnfOM+a0kh5McSnjdl/Vlkk7V2a0qk6wA9gCXATPAwSQ
f0eATwiyceTfDLjpmFvIGR7kqkkU7Ozxs2VK0ltGvV8kQOA9cClwDbgLUkeOjioqvZW1XhVjY+NjS1ziZJ0fussyCI4Da/raq3t9
wG7K+qQ0l2J9ncG3YAuDPJYeAjwEuq6s6uaplknaqzy0cBqmoSmBzo29W3XcDVvYckaQRGPVksSRoxg0CSGmcQSFLjDA
JpiRHkkwn2Tlk/xVJZpPc3Hv8QZf1SJJ01dmtKpOsAPYAlwEzwMEkE1V1eGDoe6tqR1d1SJIW1uURwUZguqqOVtUjYB+wp
VY2PjY0ta4GSdL7rMgiOA/1/4a/u9X1XVd1Zvd/sNd8KPKHDeiRJQ3QZBAeB9UnWJVkjBAUm+gckeVhfczNwW4f1SJKG6O
kbMlJCKxhkEktS4Beckly90P6qesOZLUeStNwWmyx+cO/5kcATuffyz18HbuiqKENs8lkwCKrqLwGSfAx4fV9tdd+jfD+zqu
7n8tEfBO6uqjcBM0nWdVSTJGkZLSklkrwCeCn3LgFXQODdXRUISVo+Sz0i+A3mVgf9P4Cqup17Ly2VJJ3DlhoEj6qqgAJl8qD
Jks5hS0z1dNmQvsvPZCGSpNFYMAiS/HGSW4FHjrm17/F54JbF3zjpiRHkkwn2bnAuGclqSTjp/8tSjLuj8VODV0HfAD4a6D/
TgD7gC1Dxr0KeDXwdOoW5J0hiwWBNf1nm8EpnrPN/a1F7IKONbXnun1fVeSxwNrqrBja2TbE8ylWRqdnZ2kS8rSTod
DDhhLEkLa+IXj76tCSTSR6W5DHAJ1l8raGDwPok65KsBLZy760uqaqvVNWFvBw2qtb23nNzVS12ykmSdAYt9ZPFz03y28
0z9tKl1CJJOrOWemrofcBfVNUfAk8BPsfqR9J0luqfcs3lhVd8PcBwuA1yd5X3dISZKWy2JLTPw5QFXdneQ5A7uv6KooSdL
lsdhaQyuWqxBJ0mgs9XMEkqTzIEEgSY0zCCSpcQaBJDXOIJCkxhkEktQ4g0CSGmcQSFLjDAJJalynQZBkU5ljSaaT7Byy/4+S
P+Ceu571PAhXNJWkZbfUW1XeF6uAY33tGeBJg4OSvAC4GlgjPHXYGyXZDmwHuOiii854oZLUspFPFifVnqp6OPBS4OXzjN
YjO5giq6mSSHcABYAVwbVudSrlbmKqQCBWHkqcD3wLuAp7fVT2SpOG6nCymqiaByYG+XX3bL+ry60uSfjfyWJJ0mgZB
XOIJCkxnUaBEk2JtmSZDrjzIH7r05yOMktSf41ycVd1iNJOiVnQZBkBBaHuBzYAGxLsmFg2E3AeFU9DrgeeE1X9UiShuvyns
tSaaSTM3Ozp7BEiVJZ8VkcZLnAePAa4ftr6q9VTveVeNjY2PLW5wknee6PDV0HFJT117d6/seSZ4OvAx4SIV9s8N6JEIDdHIE
O0ISR7o8NURVTQKTA327+raf3uXXlyQt7qyYlYkjY5BIEmNMwgkqXEGgSQ1ziCQpMYZBJLUOINAKhpNEEhS4wwCSWqc
UGSFcAe4HjgA7AtyYaBYV8ErgCu66oOSdLCurxn8UZguqqOAiTZB2wBDt8zoKq+0Nv3nQ7rkCQtoMtTQ6uAY33tmV7fa
tjku6DzoKqgk4CO4ADwG3A/qo6lGR3ks0ASZ6YZAZ4DvDmJle6qkeSNFyXVw1RVZPA5EDfrr7tg8ydMplkjc5MVksSeqO
QSFLjDAJJalynQZBkU5ljSaaT7Byy//uTvLe3/1Nj1nZZjyTpVJ0FQZIVwB7gcmADsC3JhoFhVwJ3VdVPAW8EXt1VPZKk4bo8It
WpljnVTcpgsZ+PduVV73/FGXoO/lz+Y9XnFG/j6+eL4dXQbBGVNve4G9o67jfjRkqqrGR12HNmifzeXT5amh48CavvbqXt/C
1NVNQG8DXhXkmngy8yFhZaXp9x0tvJnc5nEP8AlqW1+slisGmcQSFLjDIJGLbb8hzQqSa5NckeS/xp1La0wCBq0xOU/pFF
okcM/yH7cB+6vq0GirkuYkeQ/wCeCRSWaXDNqms53LjEhSY3ziECsGmcQSFLjDAJJapxBIEmNMwgkqXEGgTSPJP9xmuM
S7Opt/0qSjYx/6FGqrOb10vnmUuAnwZuBz4OPDnJFPAW4KnANPDewEvAz5cVb+f5KHADUK+BFwDHEzyb8DfAM+oqu
luabfdvZvH/OwGeVVHhux7LHAN8JNnqDbpfvGIQLrvPgOsTfLwXntb374DwAv75hlu6T1fDLyYuVNNIy50jLWKw1IEEj
INAKhpNEEhS4wwCSWqcQSBjft/MclTBktvBQYAAAAASUVORK5CYII=\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs_background": "light"

},

"output_type": "display_data"

}

],

"source": [

"#Importing seaborn 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": "IGKO-vwWmrNk",

"outputId": "d186299a-99ef-4fdf-bc4d-0ec273eccf4c"


```

" 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\n",
" + ' 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 library 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": "Q42zliyfn9t3",
"outputId": "7266b78c-4d79-4e74-8681-e3086057c216"
},
"outputs": [
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAASgAAAGoCAYAAABbtXOxAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAEIEQVR4nOzde7xcdXno/89jCAYkCoRAMTs0WEC5iCEEIOLhUCi3IAZUFKiVBKLpqa
n9Z67su88zsybOerPmu9Y3MRJlkSVLnm1odgCRJkjSYWCBLkiRjFRbikiRjUoUFsiRjklRhgSxJkiRVbNPqABphl112yXHjxrU6D
ISZKkCgtkSZIkqal+jyBLGhpeffVVuru7eemll1odSq9GjBhBV1cXw4cPb3UokjTg2iEPQ99zsQWypLbV3d3NyJEjGTduHBHR
IX0G+eBzrm11CJu18NlzWh2CNOgN5qQMgz8+aagY7Mf8gdTs+qEd8lxfY/QMsiRjklRhgSyp4wwwbNozx48dveFxyySWbXX
PKTn2xUajl0ZLRzLvYMsqQhYc2aNbzzne/kscceA+D000/nG9/4BIAbEvnZZ5/I3HPP5ec//znjx4/nnHPOAeDSSy/lkEMO4cA
CwCjg1M58s+5gJTAPWA3+embc3Om5J7evFF19k/PjxG+ZnzpzJqaeeyuWXX87UqVM5++yzWb16NZ/4xCdet90ll1zCI488s

```

a5r8AXJaZcyLi69QK3yvL8+rM3CsiTivrnRoR+wGnAfsDbwf+JSL2ycz1TYhdUhvq7We9Y445hhtvwJ EZM2bw4IMPbnE/d9xx
4OQyFVKZpyw/uxq/EjAnM1/OzCeAZcChjYxbUmf6zW9+w5IIS9h+++1ZvXr1FtfPTGbOnMmiRYtYtGgRy5YtY9q0aU2IVJl6
777ct1113HmmWfy6quvwm75yJjEjWbt27Yb54447JlmzZvHCCy8AsHz5clasWMERRxzBTTfdxIsvsjstWv54Q9/2NT3lUntr
TRRVxwwQUb1hs1ahSHH344BxxwACeccAKXXnopS5Ys4bDDdGNgHx124Dvf+Q4TJkzg1FNP5T3veQ+77rorhxyxSNPfoy
xFyxYsMUyB/uoOo6k123ekiVL2HffVsdhZtKs6lWjZE1sUUIPUm4ulZhsx/yB1Mz6oV3yMPQtFzesi0VmzszMrsWCR+0iu
R8Qy4DlqRTWZ+WhE3AAspnY2eoZ3sJkSVKjNKNVAzxs7gHvK9ONs4i4UmfkS8OFetr8YUlhxEUqSJek1jqQnSZlkVvggS5l
SHwscx8JSLedFwLHAysAk7NzCfLPmYC04D1wJ9n5u3NfyEsbRtW5GEY+FzsGWRJ6qf169czY8YmfVsjH7F48WKuv/56Fi9e
vYdtttOe2007j55pu3vGETRUQX8AfA1WU+gKOA75VVZgMnl+mTyjxl+dFl/ZOAOZn5cmY+ASxjE3cjKqRWaEQutkCWpH5
Avg9J2qRG5GILZENqUBfxlrAiMxc26zUz86rMnJiZE0ePHt2sI5WkAeVFepLUT2PGjOHpp187sdrd3c2YMZs8sdoqhwOTI2l
MGfOHCZPntzqsDbIzJmZ2ZWZ46hdZHDxZn4UuBs4paw2BejprDe3zFOW35WZWdpPi4g3lztg7A3Mb9LbkKTNakQu9gy
gEzMnN988OWNNg1Ow9DY3KxBbIkbyYVjkyXadKkVoexRZl5D3BPmX6cTdyFijNfAj7cy/YXAXc3Lkjj6r+BzsV2sZakSZlqLJ
WRRx4Z0NeXpHbTKXnYM8iStBwMtp3Kbbfd1uowjGnlakQetkCWpK1wxBFHsPPOO7c6DEkashqRhy2QJUmSpAoLZEm
6BEka0hqRhy2QJUmDzsHnXNvqEjpm4aVntDoESRuxD7lKsZJUObACOSJGRMT8iHgwlh6NiAtK+54R8bOIWBRY342IbUv7
RT0NWF3aLyvrERH7AacB+wPHA1dExLAGxi2pTywYMYJvq1YN2uScmaxatYoRI0a0OhRjaojBnoehf7m4YX2Qs/ZjvVBmh
JLUEO2Qh6HvubihF+mVM70Lgb2ArwE/B57PzHVllW5gTjkeAzWnKjNrlmINMKq031vZbXWb6mtNB6YD7LHHHGP+XiQN
Scue4Clpfp5cBYgLL8bcCqavsmtpEkSZIGVCPvYjE6InYs09sBxwBLqBXKp5TVpgA3l+m5ZZ6y/K7Sj3kucFq5y8WewN7A/EL
JkiQNYy28i8VDwEGbaH+cWn/kjdtfAj7cy74uBi4e6BgISZKkTmSniR1MAdtkqS+s0CWpM7moE2S1EcWyjLUwbKmt0Gbv
pN5YIEtSh3PQJknqGwtKSRoiHLRjkupjgSxJHcxBmySp7xo5UlgkqfUctEmS+sgCWZl6mIM2SVL2f2VCkiRjqrBALiRjkioskCVJ
DXflu6D/CfAp4G3Awt5LTH/B3B5A+OSJNWYhyWpyTZblGfml4EvR8SnMvOrTyPjKlSYhyWp+eoaSS8zvxoRvwuMq26Tmc
naLPOwJdVjvWeQdwEWR8R84OWexsyc3JCojEkbMw9LUpPUWyCf38ggJElbdH6rA5CkoaLeu1j8a6MDkST1zjwsSc1T7lC
Ykhqr3i4WH6zMvona/ThfakhEkqQ3MA9LUvPUexELP6xMrwOepPbznSpOczDktQk9fZBPProV046IsdSGQN2N2oUIV2X
H+Sl6lqIH0TEivL4fkR0bWGzdcBfZuZ+1C4kmRER+wHnAndm5t7AnWUe4ARg7/KYDlxZXntn4DzgvCchwHkRsVOF3qUkt
FwNzAOOrzNuSeoUfc7DkqT+qbdAHp2Z38zMdeXxLWB0vS8SEeOAg4CfAbtl5jNI0S+pdCGAWvH8dGWz7tLWW/vGrzE9
1mSOo4/c7DkqS+qbdAPgv4CLUzvs8ApwBT7RRRAynVhz/U2b+c2n+Vek6QXleUdqXA2Mrm3eVtt7ajWko6W8eHhsRd0
AJZn5j5VFc4GexDoFuLnSfkZJzu8D1pSuGLcDx0bETiWBH1vajGko6XMeLrxgWpL6qN77lB9YLpADIDOfi4iDtrDN4cDHglc
mw3cA3yWygXTwL0R0XPB9JGUC6YBlqLNgunrB+TdSdlgUm+B/Kal2KknOZczCZvdNjN/AkQvi4/exPoJzOhIX7OAWXXGKl
DvdXX2QM/Na4IPAr8rjg5n57UYGjkl6zdbk4WZeMO0dhSR1grp/nsvMxcDiBsYiSdqM/uThOi6YvoQ3XjD9ZxExh9oFeWsy
7egUlKSZKklcECWZlKsaqwQJYkSZlqLJAISZKkCgtkSZlkqclCWZlKsaqwQJYkSZlqLJAISZKkCgtkSZlkqclCWZlKsaqwQJYkSZlq
kCgtkSZlkqclCWZlKsaqwQJYkSZlqGlygR8SsiFgREY9U2naOiHkRsbQ871TalyK+EhHLLuKhijhQ2WZKWX9pRExpVlySJekSw
U+3xuYdbHYK0RQ07g5yZPwae26j5JGB2mZ4NnFxpvrZr7gV2jljdgEOAeZn5XCmK5wHHNypmSZlkqdl9kHfLzGfK9C+B3
vjLW093t72BO8s8vL6723Rq3d2odHd7L3AocF5PUS1JnajZBfKvStcjyVOK0r4cGfTzr6u09dYuSaqD3d0kqe+aXSDPBXp+m
DjwDLgG8AnATLzOeDzwH3lcWFpkyQNALu7SdlbNew2b5l5ei+Ljt7EugnM6GU/s4BZAxiaJA11v4ql3TPzmT50dztyo/Z7m
r4EyodXeLij7ubmB3N2lQG+x5cqA1lu9aEtSB7O7myT1nV0sJEmSpAoLZEmSJKnCAImSJEmqsECWJEmSKiyQJUmSpAoLZ
Zu3xuYdbHYIKSR3DM8iSjEiShQWyJEmSVGGBLEmSJFVYIEuSJekVfSiSjEiShQWyJEmSVGGBLEmSJFVYIEuSJekVfSiSjEiShQ
qgkaegwD0saStqiQAYOBZZl5uOZ+QowBzipxTFJ0lBiHpY0ZGzT6gDqNAZ4ujLfDby3ukJETAeml9kXlUkXsXWML8NuWDP
k5sdVxyL/FYOLfYnDrxFzcaH6nNdD8Tm29dulisRwYW5nvKm2SpOYWd0saMtqlQL4P2Dsi9oylbYHTgLktjkmShhLzsKQH
KC2RjkiSpwgK5gSjifUQsqjzGtTom1UTE30TEoxHxUPnbvOF+rnXsY7LD7W5eRowWEddFxoMRsTAifhoRH2h1XNJAAWaa
cLbt634ycy5exd+r8h2/CZidmX9U2n4bmFzn9ttk5roGhigNhf7zvDTA/hM4ICK2y8wXgWPwVosN4xnkJJoqlcRHxWERcZv
5n/t3zm/1A+2/kRsRdARPxhRPwslh6liH+jIN1K+9SluLxMfysivhIR/385W3pKy97d4HEU8Epmfr2nITOfysyvRsSwiLg0lu4r3
Z+OOpEXFw+d4ujljbI2L30n5PRFxWcv6SiDgklv45lpZGxEWV7W8qz24atZEJN/Uaf1z+HSyKiP8ZEcoA9f7UURcCf1CmTw
mTgCuBP6qtP13YE1mvjszDwTuKmdA/xb4/bL+AuAvmvD+2tUd1P5z8n8i4oql+K+VZWsY893A5cCXSttPgPdI5kHAHOAz
ccChwHkRMRYBYBZwBEBFvonZP3+8g9d3r8nz5fn0VOCUzD6b2Xbu4sv4rjed/HbgZmAeCAEYniFFlnbPKthOBP6+0AXar+
dOeysx7K+t8pjwh2lZagbUf8FBZ9s/leSHwwTL9+9QO5gBk5uqIOLFs928RABXuAj8d6DfTKTLzhYg4GPgv1BLEd+O1vsTX
Awtb8fbP5v9L/KrwUvR8QKYLfMfDiVxKEQcBuwAOZuWqA37KGho3z/AHUct55JR8PA56prN/TLeth4NHKd/txaiMVRqJW
bmAVy8/1nz0Q5Y/ZXwCGlOP0WtS9rj5fl83o2/7cKYF5mnj7AsXaszFwP3APcExEPA1N6FVIXK89fBf4xM+dGxJHA+b3s9uX
qBW+h/WyvPr93fi7vU35/v4+cFhm/joi7uH1x4We15idmTMHLGq1k7nAf4EjgeqvC58H7s7MD5Qi+p5NbBvAhzLzscaG2F
ogpqF53ASMi4k8rbduX59uBPY0/RxMR+2zlz3z9+Rv9ADieWheP27ewrlSvx4DRUbsYmIgyHhH792H7twGrS3H8LuB9m1
HxIPB7mbmS2j+G6yPiWqF3bsaE3VH2IHaz1CLy+e1H6+dcdyptJ0N/H+I7XzgxohYCDzb5FjbVtaG6TyZwt/6JyJiPjAb+Cy1
1J3eUfDWPWhc9DQGZ2Z2ZX9nEon8A/j4iHqD3fPp5al0vHoqlR8u8NsGhpqUiIp4EJmamRfAQUC7Oux/4cGYubXU8kqT
WzZPREyNiMtbHYcktVKj8nVEHBkRtwxEjFIPC2S1rXKj85uAH2fmOzLzYGrDcHc18DWHNWrftkSpWpGvNxOLOWhriyyQ1
/hwRHyi7PfBiPh+RGxf1vtWRHw9lhZEXp+JiBMrbs49Im6LiKUR8Q9l/bMi4ks9K5T9XtbwT02Smq8/+frliLgnlr5XcvM/VUaA
gNlf0eYBfgoj4MXA4MI7aqHq7AkuoDefZY1VmTgCIIFGZ+Y0yfVF5ra+W9cYBhwK/A9zdM/x3ec2DgleBxyLiq8ANwN9Ex
kG1NnskBWx4ilrwHvB14BnglO7DkLTG2M+r3L8uvL0MK/loh/BQ4p7Tdm5m+AX0bE3RvtvpqADyiF8Y7Uhq2+vbLshrKP
MDki/qRMjwD2KNPzLi5VLwtktbNHgQ/1zGTmjIjYBVgA/AL4VGZWi1ci4oR+vtZ/Vqa/BZycmQ9GxFTgyMqyjUfe6Zl/udK2n
BGxTOS8BfjfwKmlz9to4AhgPrWf7j5U+iLvxuuL3o2NBj4p+/7oRss+XPbxO8A7gMfesHVFZv4MGAv8EXD9Ft+xjLWn/uTr3
QrgZOBHwCHAQ9SO7v7mcz8ZUR8HzgaWaw8Ta2v3jpeXvq/Az8r+/wZtYK5xy+oFdxvBf5bZr5U8vTm3ACMz8zV9b97SV
ckpnf6+Pr3gJclpleWS1J0iDgGWTpNbeUq563BT7fl+K4P3qusAYetDiWJGnw8AyyJEmSVOFFepIkSVKFBbIkSZJUYYEsSZIkV
ZuboVsfRSOZiSYNdb7m4lwwkcePGsWDBglAHlUm9ioinWh1Do5mLJQ12veViu1hIkIRJFRbIkIRJuoUfSiRjKlTRkX2QNbS9+
d3dzZ577tnqcCRj6rfBfsyF/h137WKHjvPSSy8xatSoQfsPNSIYNwRuoP/ftiRJWzLjY7nQv+OuZ5DVq4PPubYlr7vw0jO2eh+
55rrjij6/3vnnn88Xv/jf/oYrSVJba+fjrl0sNGRst912LFq0qO71b731VgCeFpJrrjiCj75yU82KjRjkjpoOox93PYOsIW3NmjW885

/iy++mH322Yf3v//9G/YpSZJq2uW46xIkDRkvvvgi48eP3zA/c+ZMTJ31VC6//HKmTp3K2WefzerVq/nEjz7xuu0uueQSHnnkl
w4AFwPLMPDEi9gTmAkoAhcDHMvOViHgzcC1wMLAKODUznz7mAlMA9YDf56Ztzc6bnWe3n7qOeaYY7jxxhuZMWM
KZW+FKev5f2y8p6RMR+wGnA/sDxwBWL6JYGxG9+8xuWLFnC9ttvz+rVq7e4fmYyc+ZMFi1axKjFi1i2bBnTpk3b4naSjKk9
kmr7766uuWjxw5krVr126YP+6445g1axYvPACAMuXL2fFihUcccQR3HTTTbz44ousXbuWH/7wh019H5iktYN2OO42uov
66quZP38+I0eO5IgjjuCiij7iggsu2LDeqFGjOPzwznggAM44YQTuPTSS1myZAmHHXYAdvssAPf+c53mDBhAqeeirveo
4L3A+WWb75T2a8o236MXEyDzAULFjTfKfQ0l7TqS3pIIS9h3330HKJrGaZc41RgRsTAzJ7Y6jkYyF2swcSS9xminY9mmYu0
LWL7O7KzI8CdwOnlNWmAdEx6bllnrL8rqyd3p4LnBYRby53wNgbmN+ouCVJkJS0tel+yJ8F5kTERcAdwDWl/Rrg2xGxDH
6MilcjYIFELChTO0fEvIhYWP53Ku0REV+JiGUR8VBETKjsZ0pZf2IETOnt9SSp3VkgS9LQ8HuZO5ytf5wj2ZuTdwZ5kHOIH
a5yTgyDI9m1o3uM+W9mvLRdL3RsSOEBf7WXdeZJ4HEBHqzJ1uen1zw5Y0mHXKMdczyFIDrF+/nhkzZvCjH/2IxYsXc/311
nz+fvfbai3e84x1su+22nHbaadx8881b3lBqjPdn5gRq3SdmRMQR1YXlBPGAjBqVmVdl5sTMnDh69OiB2KUkbVYjrkWyFI
8CxcwCO8fnCmjQdtOqPczeJ9wJrSFeN24Nil2KlcnHdsAZOkjJmKl9r1XjszRwbXa01ZswYnn76te6a3d3djBnzhu6aUjPsBw
B06VKeeOlJxowZw5w5c7juuutaHZaGoDI403s20b4KOHOt7QnM6GVfs4BZAx2jJG2NRhxzLZDV8Vpx5n6bbbbb8ssv57jji
CWZIkSaqwQJYkSZlqLJAlSZKkCm/zpo73iwwfPaD72+NzD29xnbPOOotbbrmFXXfdlUceeWRAX1+SpMGqU465nkGWGm
3tTx6rIfzEA7/FTTueeee3j22Wfp6uriggSUYNq0aU2PQ5KkZuqUY64FstQA119/fatDkCRpSGJEMdcuFpIkSVKFBbIkSZJUYY
X2NsWEFckSMilj5EfFgRDwaEReUdhOzGmrEiBGsWrVq0P6DzUxWrVrFiBEjWh2Kho6zgSWV+S8Al2XmXsBqoOdqlmnA6
q1xHzqRon57cC/RMQ+mbm+gbGrjXV1ddHd3c3KlStbHUqvRowYQVdXV6vD0BAQEV3AHwAXA38REqECbfXRWWU2cD
LwGeAkWV+FPB8Zq4r893AmDI9BngalDPXRcSasv4Y4N7KPqvbSBrCOvWY29A+yKxf2yJgBTAP+DI1JmagmpifruX2k4k5l
iHgSmEPtF7wvAztGRM8viF3A8jK9HBGLUja/DVhVbd/ENq9JlpbUCZpyF4vMfB64GziMBiZmSdJrMnNmZnZl5jhq13LclZk
qOr2zWI/RpgVGn/C+BcgMx8FLgBWAzcBszwQmlJnayRd7HYHhZhdBgX0JuCGzLwlhYDcyLiluABXp+Yv10S83PUznaQmY9
mSKiyQJUmSpAoLZEmSJKnCAImSJEmqsECWJEmSKiyQJUmSpAoLZEmSJKnCAImSJEmqsECWJEmSKiyQJUmSpAoLZEmSJ
YlySpjxwsSU222QI5M78MfDkiPpWZX21STJkKwjwsSc23pTPIAGTmVypid4Fx1W0y89oGxSVJqjAPS1Lz1FUgR8S3gd8BFG
fc+l+FIellul70bEtqX9zWV+WVw+rrKvmaX9sYg4boDelyQNOvWeQd4FWBwR84GXexozc3JDoplkbay/efh4KjMfCEihgm/i
P5sVLpkvFBmh5dHAKcBf1TaZ5f9XwmcVHmt7wGXR0SU9jmZ+TLwREQsAw4FftqfCRpMKv3Lhb/2uhAJEm925o8HBH
fEfzQ6OEISzdbk4cxn5njgS5qZ33f1ag4M/OqzJyYmRNHjx7dqeRplaq9wzyyJ7pyk9t72tUUJkK1xulPjyZz0fE3cBhw14RsU
ltSnAzWV6bpmnLL+r9GOeC5xW7nKxj7A3MH+A3plkDSr1DhTywcrsm6jdj/OlhkQkSXqDrcjDuwOzSz/kNwE3ZOYtEbEYr
qfVH3rj9JeDDvezrYuDi+sKVpPZVbx/kMxsdiCSpd+ZhSWqeeu9i0RURP4iFeXx/Yjo2sI2YyPi7ohYXEZvOru07xwR8yJiaXne
hLTucCdmbk3cGeZBi2kUfe1O7h+aVUCuogfOA91L7OfC8nqJakoAQ/uRShVI/1Fsgj87Mb2bmuvL4FrDZG1xm5jOZeX+
Hl8cfU7otZl4gYR+0ikZ8Bu2XmM2XRL4HdyvSG0ZuKnlGaemvf+DWmR8SCiFiwcuXKekOTpHaxVXIYklS/egvks4CpUCton
M/OfS/KvSdYLvYK09zZKk6M3SVI/87Akqe/qLZAPL1/AcjM59jEfTWrylCo1wBLMvMfK4uqozRtPhrTGeVuFu8D1pSuGL
LAN+DZwjtYNARHweuK+sd2E5MEjSUNKfPCXj6od6k+v/AH4aETeW+Q+zhdGUMvMnQPSy+OhNrJ/AjF72NQuYVWeskS
jqEpIl46RmtDkHSRjyDLEmSJFVYIEuSJekVFsIjEiShQWYjEmSVGGBLEmSJFVYIEuSJekVFsI1MEiYmxE3B0RiyPi0Yg4u7Tv
buLPMAJwB7l8d04EqoFdTAcB7gUOB83qKaknqNBbktTBMvOZzLy/TK8FlgBjgOA2WW12cDZJfok4NqsuRfYMSJ2B44D
XpGaxQJakiSAidG+C+D3w6M/+juiwzE8iBej3MvCoZj2bmxNGjRw/ELiWp6SyQJanDRcRwasXp2XmP5fmX5WuE5TnFaV9C
mYi4Hfi7yoV5xwlmz/EejKnZLJAlqbMdDnwMeDgiFpW2v6ZWGN8QEdOAp4CPIGW3ApOAZcCvgTMBMvO5iPg8cF9Z78L
EmSJFVYIEuSJekVFsIjEiShQWYjEmSVGGBLEmSJFVYIEuSJekVDRtqOijmAScCKzLzGNK2M/BdYBzwJPCRzFwdEQF8GZgE/
OBK2FDQX0e8F7gUOC8iNipgTFLkiRpiGtYgZyZPwae26j5JKDnDPBs4ORK+7VZcy+wY0TsDhwHzMvM5zjzNTPCNxbdkRj
06lHarSRJkoaYZhflvypdJyPK0r7cmBsZb2u0tZbuyRjktQZs6Q5WjTyvQU4OZK+xlR8z5gTemKcTtwbETsVC7OO7a0SZIk
ViBn5um9LDp6E+smMKOX/cwCZg1gaJlSVKvHElPKjYRMYKiBUR8UilbeelMbcRS8vzTqU9lulrEbEslh6KiAmVbaaU9Zev
KHHQZskaTMadh9kSVLrOWiTJPWdBbIkdtAHbZKkvrOLhSRjklRhgsXjkiRVWCBLkiRjFRbikiRjUoUfSiRjklThXSwkSWqhX1
7HAlmSJEmqsECWJEmSKuyDLLUp+2pLktQYnkGWJEmSKiyQJUmSpAoLZEmSJKnCAImSJEmqsECWJEmSKtqmQl6l4yPisY
FlmXm45n5CjAHOKnFMUnSUGlelJrRkRga2OoYtiohTgOMz8+NI/mPAezPzyrrTAeml9l3Ao81PdAt2wV4ttVBtAE/p/r4Od
n1XztEMuHmz8fmqg+Z3aeu3SxWI5MLYy31XajEnNYR6WNGS0S4F8H7B3ROwZEdsCpwFzWxyTJA0l5mFJQ0ZbdLHlZHU
CA3WETMiogVEfFlq2MZ7CjibETChRGLI+LRiDi71TENRhExliLmR8SD5XO6oNUxDWYRMSwiHoilW1odi7Qxh+/+f+3dfcxW
LmNFcUk5g1sBET+PTH+d1yvLq5ubm54Tro57Vd27l+1++c8z3X7vt3fa/vOef6RW9KztF7kiAfebOBS9odxDFiD/Al28OAsc
qe3VZ3kmV1Axob1TN48pr5Wmf8sJdtp2QNBC4DJv7lgiOpHpu6NXJefoPUmQo5EkDQLGAM+0N5JmKpcNrAW2AYtt53
OLYkQY5GkdSHKjmea/vhdsfTdLb/DTxJrjnrzIXAFekvU526nizp5+0NKejNMn13REMIQY7GkCTgJ8Dztu9tdzxNjekUSe8py
UE10BnAk5LWUX3ALradnzCLOMbY3gN0TN/9PDA/03fH4UjO0Xsy1XRERERERE0qyBERERERNUmQilyliljQkiBHRERERN
Kk9/dgvUGSPtPT/UZEHA2SXmt5fr2kmT3c1hBJj0p6QdJqSfMlnXa4fbu579mSrurp+vHWkQQ5mmY3cGVPksmDKVMPfx
XORjD4xiVBDmaZg/wl2Ba6wut3+w7KhaSjkn6g6Sfkl6SNEPS1FihXi9pcFnyl8AdtrcD2F4N/Ay4SdKNwKeAb0maW99v+TH
w8txrS37POuw3/GliCNI0scIPSNpTRnjTivtE2uTQq2RdBJVQWCF7d92rG/797Y3tl6/B+k7SNKyMpauljSu7HNSaV8EbFJlpq
A7go8D60nQuCjXticBdwBrbl4GvAXNKn28AT9keDiwApTCNY5gLPGB7FDAO2Ap8BVhme7Tt75Zj/F6pbj8PbOnGdiMijrR
NvBDM9EI1je4ekOcAtVANmdzxreyuApBeBx0v7euAjPQXlcBnoDSy0/Zik66mmdn619Lkl+GSJe0mpHL8bmABcWdofkfs
Mea3q429X+gAzJY0G9glDaq+ttN0RwwTgldt7gX+U6nREKsjRWPcBNwDvrLXtozfNlhvmTqi9tru2vK/2fB/7vwhuAs5r2c95
koZSjct43Rdfztqu804BWqs4vn8+bPi8MZw+NtlglyNFKpEMynSpi7vMz+wXAKVYXgUHwHuFvSyQCIsnA91U0dPbWM6
l6Wr+tolDty9nrbdwPPAkOBecA4SZfV+k2QNKK7XbVtz+w1fy+qsvujtAbEuBqyUdVyrPT3jGG8xSZCjye4B6r9m8WNg
AAAKZJREFUXQvcUra1HDgdWafSLTfuTaO6kXBDufRjBPuveY6laKrpwK8krQK219pvk7ShjHmvA4/Z3kVVOlhz1U+3bQK
SIEderERE1CRBjoililioSYIcEREREVGTBDkililouZ/cd/Ncm9mxM4AAAAASUVORK5CYII="

"text/plain": [

"<Figure size 720x432 with 4 Axes>"

]


```

},
"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-4535-8af6-39e9921a8010)\" title=\"Convert this dataframe to an interactive table.\" style=\"display:none;>\n",
"    \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\" viewBox=\"0 0 24 24\" width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"    <path d=\"M18.5 5.4l.94 2.06-.94 2.06-.94 2.06-.94 2.06-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94 2.06 2.06 2.06-.94 2.06-.94 2.06-.94 2.06-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04-.41c-.39 39.9.59 1.41.59 51 0 1.02-.2 1.41-.59 17.78-7.78 2.81-2.81 c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59 7.72-7.72 1.47\n",
"  </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\n",
" + ' 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": "ISS9rsBmq-MI",
    "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": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLLAAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAELEQVR4nOzdeZgl510f+u/b6+xaRqMFy/KYO/ImGWwzGPKQOEZerjCKRB7ASxjsC
kePdNVp+qtd6/qn6rPW2qtAQAAAKC7xtY7AwAAAAAB8dQR4AAAAADpOgAcAAACg4wR4AAAAADpOgAcAAACg4ybWOW
gAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg
eEwj8AAANbF37958/NN3ZmHTHeudFR5nxo8eSJLcP+Px9lw3fvTh9c4CwLpwBwQAOMvh04U59rSXrHc2ejZ+NI3J4m+C
AAPAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8
APAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8LAu3vSmN+VNb3rTemcDAACAc8C58DvoxHpngk
CPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAA
PAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAAAdJ8ADAAAA0HECPAAAAA
rfmU5/6VG699dZl2+Oaz5faXu11x08/tZbb82RI0dy6623nlYaDzzwwKqOBwAAAFYmwPMYe9e73pVaa3+71pr3vOc9p/U
MT652Bc03z9s6ghYWFvO1tb8uP//iPr3j+W9/61iwuLq54/uBxCwsLufXWW5fdXun67fTe9ra35ROf+MRJx9x666359V//9V
yZ9++feudna/K4Ns7jfn5+bzvf9b1fnvf//7Mz8/v+L5g8fNz8/n7rvvXnZ7peu303vf+97Xf3un0d5eKe/tcgAAAAABnphNv8N
b6f3ohe9KJ/4xCdOCurs3LnztPK+ffv2vPGNb1xVuQEgSV73utfl9rt8jxsw2uKGbdn19Zd4zgROci681dejN3jOJj/2Yz92yr7x
wcAAABWR4DnMXbDDTeklNLfLqXk2muvzfbt21d1/vbt23PttdemlJkdO3eOPH/wuGuvvTa7du1adnul67fT2759e3bt2tV/
rPV9pe7XUHj3/DG96QzZs3r/j2TjsNb+8AADA104Z9qW/j2e7d++ue/bsWe9s8FVq/v7R30YDcDqa7+A59rSXrHdWejzZ+l
AxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnw
xwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAH
AAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAH
AAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAH
ADwAAAEHDHcFAAAAAAdJwADwAAAEHDHcFAAAAAAdJwADwAAAEHDHcFAAAAAAdJwADwAAAEHDHcFAAAAAAdJwADwAA
WAtPqrXuaO/sXIAHvtZKKXtqrbvXOX/weGNswHDGBgxbnMCpjAseS/5ECwAAAKDjBHgAAAAAOk6AB5K3rHcG4HHK2ID
73tvb3Pd65nvmEtIVLuLqV8qpTy8VLKnt6+C0sp7yulFKH37wW9/aWU8ou9sfHJUszp1jf3sHZKKEeXUn63IPLZUsqdpZS/Z
9Qkkd6+/9j7zg4m317rfVZtdbdve3XJ/lArFXKJB/obSfjdY55svffq5P8p8c8p/DYEWOS99Ran5bkG7N0/zA2OKfVWj/Xu188K
62+I9/oLe8XCuGBwD7bHxtrkl0nOL6Vcth4ZHLVUSjkvyfOS/JckqbXO1lofjBEbg16Q5C9rrV+KsQFJMpFkYyllsimmJF+J3zd
k1zS+7k/NnoGxw2cT26cZF+S/9r7095fLaVsjsrEBg16e5O29n40Nzmm11vuS/Psk92QpsHMGye3x+wbrQIChs1op5bokD9
/W+d+qGLP0Pgq9LsjnjteuaKc5ZAJyc7b4tyfWILLuT/HaWXPv8Y5ZeE57oHXN5kvt6P9+X5IlJ0w8vCT7H8sMw2OI93+cUm
tqrXNJ/t8s/Q7i9w0ecwl8nNVqrT9da7281rozS68Tf7DW+neTfCj9/QOe1WSd/V+Vq23nd7nH+z93yg4q5RSNpdStjY/J3lXk
7Bo+5oi9xriilPD/JT9ZaryulFH2W3ui5MMnHkvy9WutMKWVDkt9I8uwkDyd5ea31rvXKM6yV3hj4vd7mRJL/Vmv9V6WU7U
+9w3OOeVUm5O8rlk81n63eLvZ+m7dvy+wWNKgAcAAACg4/yJFgAAAEHDHcFAAAAAAdJwADwAAAEHDHcFAAAAAAdJwAD
ADO3MR6ZwAA4EyVUkqS30vy1lrry3v7vjHJJUk+39ueqLXOn27atda/P7D5z5L86156fy3JdUmeU2udKaVclGTqqyzHGeUF
nlw6WU3aWU/yfjxt6bOr+V5LlkD9VaZ3rXe6jW+le9c765lPlnpZRPIFL+vJSytZSyoZTYX0spnyqlfKyU8u29Y7+/IHJbKeWDST
y1CTPyNjbP59J8muDidRaX19K+dFa67OSpJSyJcnPIFI+n+T9Sd5Ra/3/SilTSd6R5GW11o+WUrYIOZbkduVj1GeWUp6W5I8
yfOSvL3WupDkr3pv0yyr1nq4lPJNSf5Glt4gekcp5fVZCjR9pdb60d5xB5OkIPLXk7ypt++zpZQvJWkCPO+rtT48kMfrSyk/2dv
99da/3cmeQJADI3+Q4eAKDLPhkuvcnWEmSUSo3ZOntmkHvTfKPSimTvWOeUkrZnOSPk7ys9x09I2XpjZxh5gbOfWop5
TyU0mOJ7k7ye+3Dv3VJDuT/EUvgLlvyXdlaQWua7L03Tv3JPnTEZd6S5JPIIL+IsnPJ3IT73ty5pPsTfLqWutsKeVlvc82Zun7d1
QcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8
AAAAAHTcmgZ4SinXIII+V0rZW0p5/ZDPp0sp7+h9/mellJ1rmR8AAACAs9GafQdPKWU8yZuTvCjvUk+Wkq5rdb6mYHDFij
2vNWck5MjufzVMTmZ1fzlbJ8Rybn8vU+ET2HZrJq3TuWDTTeB45upAHD83k4q1L522YnEgpNbPzNfsOzWT7lqkcmZ3Plq
sZK+M5eHwu2zZM9v999OhsLtg0lW0bx3Pg2ElePTax8zdOzv/hmWzfMt0vz/mbxnNstublzEI2T08kqZmZX+yXq5TF1DrV
4Zj5HZxf6ZT9wblF/7OGZuWwZKPOjR+dy/qbj/v6mbo7Nzmfj1ES/jpo8tOus2d8/r5f383p1NjleM7dQBsq+INcmP8lCkN
4Nhszts4lUeOzuWCTZM5b+NYDhxb7Oen6RuP9PrMQl3l+ECfaveFrRvGcuJ4Yv86x+fns2Fiol/vs/PzmZqY6NdLMzaadjq+
JLxPHR4Nls2TGTbxomkLmZmPv08HJmJdy+apE32pGb/NNZo+ODM/n+mJiTx6dDbnbzrRNk2d7D8yk+2bT+S9aZv2OG/+
O3zfnHe/NJc9ymqeTobPrLaOqr6Svt9E+Ue6m+TvT5pTF9dHY+m6Ym+vNg0yf6faU11pr2a/pMU/6m3E39tLeb/DT52ziZ
4aMrjMOMLZuyPHx0NhdumurvbaZpoz9PtBLr6mrpu81123arOI7TTrNfNfUUDomTZs36W7dOJZDx07MLymLST3RR
p0/MUe3rNWOjab+mPzTlb7dHU+5LzhvPAwcWTqnH5ryDx2azbeNUds/OZcfm6RydXXrOuHjbdOYXFzIxcE9rj+/mnt1cq
```


+T7xYbj5Phc+vlu2uTw7Fy2Tj1lrxlLTxpN/TXHt+eldr6ae2NTL005m/pv6vXA8bmct+HEM1jzbz/dXp9r9jfXb7abcjTla8Zmc
nL9jxw4plgqS+d6GMnP5f0+0KvTg4cnc15m6b6+WqPmaavN2U6MR8s1cGjsbFU58080MwTzbzRlGPU/N6k0+7Ll2wbzw
zcMx7e8f8aSlIln9SXbUZTK1e/fu6kuWeawsLta854778xPv/HiOzy1mw+RYfv6lz8qLn35J/ujOB07af8sNV+edH/1Srnnapf
lz+8dFxz3n//i3vy0t1Pyjv3nHq9W66/Km8eOP6W66/KO/fckz1fOnBSui/bfUu+/uX9ecHTL8svf3hvxrb7iv7+dnrv3HPPKde
m685sp+fm9qpff+O7+Sb/u/Ls7RuYW88QMnzvnZG67OL33oC/26GkzT3+mVqSznqH+HteVgGQfT/YfP29VrmysyPz+biY
e7HDDff1W2TNUcmS0npdv0pW9/6mVD02v36RuvuTjzc7O57lItP/Stg0ePZeumjf3yvfGZF51UX4NjqMn/YHnf/OHhfa89F
+/bLdV+Thw0fzlEvPP6Wu3zxkXvre3VfkA3cutWW7DzZtPLi/Gb/PeuL2kW2+0vz4gd75pzN/DZsnP3//o7lwy6ahbf+a5+/K-
rMwrpdX0jWuedmk++Nn789LdV5wyf4yab9vz6rB72Wuevyt77n4o37TzolPy1dTZ4HgfvM5yY2RwLA321T/6zEMn3csG5
MvfekYWUnLzHwy/l4661w57jrrmaZdmamwh2zZtPKkN2/fl1YyX9j1qWFsP5qMZS8s9f41KZ7BuRs17g/lpl39UuoP1stl9e
zUfi4Z9Xw2rM/cfvfj96phzxCDdTwsv8vVYXt+H/UssFJ9tX+XGJbPH3n+rvzNK7flj79wsDWWrs6Lr97R2SBPKeX2WuvuU/av
lb/nlKfv/7fd8Y/7p735i1ft/6K9/fd78ob15zbfvyn/533cN/fy//O+7+se101sp3cHjb3z7x05J91e+75vyw79x+0nXGZaPUddpz
dT0qbVZKv10H7Tperi+dTl8bdVxz/Xb9rbae2vXSTndUeu12fOsPPDevGpJOO/23/uA3D+0zg/kfLO9y119N/Yzqb7/+A8/N
wd//aOnPYZH9YeVyn+m9fvWH3huHjg0c1IZVirzqLTa96xR8+9K8+qoe1m7Ttt1NmQ+Ot1776983zfIvB/20f52e75ZKb2V-
XacrmxvlJ5V/p8NeVfbsydbjXOn9Yv7jxBbuGPpeOmsube9Bq6mhwjnzu89Nqy76a+WA188xqn5tH3atWqq+V0m/+HfX8
B4/rpzc6vKt3B44el++iRuVOuMzT/s8Ov05zfaOdnVHoPHDO+NP8r5eOR1vX6+Tg6l8U6ug6Hlb0pU/uaK9bFMNu+mG5T
Z1QZR80Lp5PHYX121HGj8j5qXho5TkfsH3X8asuxUt9ZrnyD28uVc1g+V5p/R33+0OGZMxrDo/rDSuU/0/p98NDxU8pwp
y6n3A+HHHdkhXNP57nodNtwNZ8vl/dhbX2mz1+n27arzd/p1t9KfXalZ63TLdeoOWRYfleTvxXLucrzH+Vj1HPpqLm8uQe
dv6DG9P9E6L8n+dkK11rfUWnfXWnf2LFjjbLp7pk24ZsmDx5mGyYHMTl520cun/T9MRp7R98gW7U58OO2zg1cVrpbpy
MOq/dNqPqelibtetsue0z7Wvt45rrD5bndPrMqDE1rC8ul/9Ltg5Pp71/VJ9p2mlYeb+a+hnV3y4ekd+xkpHnjKqrUXk5nTyC
vax/zxpy71rNuGvOH3Uv27F1ej0288FK89Wo67bLdf6myPPTHZfGR6V3pn3mTOerYfe5Yddttkf1zeXmltPJR1O+Vd8Phxy
qU6GNY+8DSfn1bbxttXMR+spi1W+9w86llg1Bi9eMj8vbw+R6Wz9N1IZ5exlQ85Yx9NcmUp5cmllKkkL09yW+uY25K8qvzf
X+8JNLMc8/+MR9ueX6q0457w8/eV9uuv7Ecc15b/uTu3LL9cOvd0vr+FuuXzq+ne6N11yZ3/rIF3Pz9VfIDz6xtN38205v2HV
m8roXnHzOz95w9U11MvinTZnaZV+uLpq2bOehSfdE2yyVsV037bq4udU2o+p6MP12nbfruKnD5fracPwP9p1RfbK5ftN2
+dkbrj7p+Ne94Mrs2rFl6Dk3XXdV/vQLp5Zx8PNh57XbsN0XBsd/8+9yffrGa67M++4YPo6HzUs3X39VfrPXlqPaeNj4Xa7N
aOHlXmItjq+0fS1YfPHcuNucF4ddi+75fqLOh2Wrz/t1dmo+Wq5MTKsj/zWR7540nZ7nl+urw6Wf1T9jeozTfnbY3u5eht89J
XVPHMMu177Gwa5/LTLPyrdpm1Wcw9eri3bY35YX1muvKP67HLprtT3hl2n3c7D+slqn+uH9YuLtkyPnJeGXft9d9y3Yp9c
p5pxt1uw7eJkklPKSJL+QZDzlr9Va/1Up5ZYke2qtt5VSNi5jSTPTvjwkpXWu9aLk3fwcNjrVkt68FDx3Px1INX0Wr2N6toPX
krq0CslFm6cyMVYzv4pVtNorSh06PpetQ1bROm/TeB49utA/b9QqWkdnF7JpanzgzzqGr7jSrNZS5ix3koq0xMTp6xC015
mtmV7GKVrsso1bRavRQSqtoNauU9FeM6OWrKUezyLU7/1umSw4vs4pWszpDexWt9qo07ZXZmhXS2qtoNX2iWf2hyWe
f04fYqWv2Vf3p18vCRmVw4slpWc1zTJs14b/5tr6LVXGfUKlRNgitNXTTrt1tR9e5w22+1VptqraD3cXw1w+CpP7ZXfRq2i1ft1
18PGcv/HECjttcdyUqbnnNNvtlcPaK+g0q7X0+3BrFa3+alb9/UvzXtNm7VWamr13bXHqKlpLfaRjp73qVLO/3+a9dJtVtjpyN
b+5p8tH0h/aqgk167VW0TqxS2F5Fa+kZ5ujsXC7cNj1j4cVzd3ipzzUrEDV9o32vbd+z23XSX8W0d+32ymbN+GzqeNQqWs
YpW80xyyqqlrb7bfgZrr6LV9Llmf3P99ipaTfmasdlcd6VvtJoVRpu+1l9Fq3Xfap6M4eNwKXr8MzSdRcWc8p4b67V9NFmu
cqaXP99ipazTzVzE/NcWfbKlqP+ZcsrxUBHgAAAOBc1ekvWQYAAABgNAEeAAAAAgI4T4AEAAADoOAEeAAAAAgI4T4AEAAAD
AgI4T4AEAAADoOAEeAAAAAgI4T4AEAAADoOAEeAAAAAgI4T4AEAAADoOAEeAAAAAgI4T4AEAAADoOAEeAAAAAgI5b0wBPK
Tfjv1yo/AAAAAGertXyD57lJ9tZa76q1zib57SQ3DB5Qa/1QrfVob/MjSS5fw/wAAAAANjXWMSDzhCRfHti+t7dvlB9K8j+HfV
74kTxxYvry37ySllBcm+edj/matdWYN8wMAAABwVlrLN3g+muTKUsqTSylTSV6e5LbBA0opz07yK0mur7U+ulZ5AQAAD
mFa3l9AAAAAgHPB4+JlIgEAAAA4cwl8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8
AAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB23pgGeUsq1pZTPIVL2lljev8xx311KqaWU
SX4uyfE1zAsAADAWWstAzxPSPLlge17e/v6SinPsfLEWuv/WC6hUsqrSyl7Sil79u3b97XPKQAAAECHrduXLjdSxpL8fj/vN
yk/sdzntdafX+bjya5spTy5CwFdl6e5O8MnHsgyUUD1/pwkp+ste5ZodsAAAAANjYN8GTpLZskeWqSb05yW2/7byX58+V
2zx0s+AAAAwLmnlHJ7rfWUr7dZ7XfwXjJkdmB7trcPAAAgHW20p9oNd6W5M9LKb/X2/6uJG9dmywBAAAcDpWFeCpt
AAGD1VvsGz99Ocn2Si0lSa/2rnFhCHQAAAIb1tNoAz2xdWk+9JkpkZfPaZQkAAACA07HaAM87Sym/kuT8Uso/SPL+JL+6d
tcwIAAAAGdm2Td4Sin/KMmPJpN6UsonBz7amuT/rGXGAAAAAFidlf5E678l+Z9J/k2S1w/sP1RrfXjNcgUAAADAqQ0U4Kn
ajVSvmh1vZ4KeWmtckSAAAAAKdjtcukv6CU8u5SymWllKuTfCRLK2kBAAAAAsM5W+g6eJEmt9e+UUI6W5FNjjiT5O7VWY6
l8t9Z6MElqrTXJfyil/MHaZQsAAACA1Vr2DZ5Sj9NklrrwVLK97Y+/v6VEi+IXfTk+VwpZW8p5fUjjnlPeUzpZQ7Sin/bbUZBw
i52Hbbc9NsrFWeletdTbjbye5oXXMP0jy5lrrl0lSa31whTQBAAAAAfKpWFnH/Dxsu+QJsb48sH1vb9+gpyR5Sinl/5RSPJlKGF
QbXWtyR5S5Ls3r17pcASAAAwDII2QBPrXX8q0j7viRPHNi+vLdv0L1J/qzWOpfki6WUz2cp4GMJdgAAABVWulPtL4aH01y
ntkv2lIM8k+VCSf1Jr3b9WeQIAAAA4G5Vau/WVNrt376579uxZ72wAAAAAPOZKKbfXWne396/ln2gBAAAA8BgQ4AEAAA
AAgl4T4AEAAADoOAEeAAAAAgI4T4AEAAADoOAEeAAAAAgI4T4AEAAADoOAEeAAAAAgI4T4AEAAADoOAEeAAAAAgI4T4AEAA
rSinPaB32hiTvrLU+O8nLk/zyWuUHAAAA4Gy1lm/wPdfj3lrrXbXW2SS/neSG1jE1ybbez+cl+as1za8AAADAWWktAzxPSPL
Wutbaq27a627d+zY8ZhneGAAAOdxbC0DPPcleLA9uW9fYN+KMk7k6TW+qdJNi5SaA3zBAAAAHDWWcsAz0eTXFlKeXlp
b6/1lrxKk8AAAAAZ6OjtUy81vruLH158uC+nxn4+TNjvm0t8wAAAAABwtlVl1kGAAAA4KskwAMAAADQcQI8AAAAAB0nw
l8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nw
zxsBkiucvLtbcvf9lHjh4PJumJk7sJDTm6dPOX/wuEu2bcgVF2zKPY8cHbm90vXb6e3cvjmLizV3fOAvnLgeC47b2Ouumxb
2zi16rZsp/d1Wzfk0/cfzP0Hj+eSrdOZmhJL/ELNgeOz2b55Q55y0ebc8cChHJ6Zy6apiew/PJuLt05ncqKk1uTl8YVMT47l0M
Zke2g4XbZrKw8fm8uDB49m+ZToz8/PZODmZ2fnFXHreqW0yP7+YO75yll8enc3GqYkcPD6XbRsm8/CR2ezYMP3NG8Zy
7GyTxydDYXbjrKWFIrWP982YX5jM1PpFHj87lws1T/TY4fHw+UxNjeeTYXLZvmsrTL9mSzzxw+KQ2u2jLVePzGld61/nW
xhTx4aCYXb53O+Fjj+NhYjsaTlzMLeDgTC7eNp25hYVMjo/n4LG5bNs42U9nYqxmfrH089Ok0xw/WF+Ldaw/j9YspmQsF
bZrq72+uu23jWA4eW+yXq2YhJeN5+OhsLtw0tdTnpyayWBdTyJl2HZrJ5RdszPG5hew7NjLtK3n/E3jeeToQr+PNGINT9TM

NW/TrvOm/SbdNp115w/Mz+f6YmJHDg2m/M2TuXRY3M5f+NSH5scn8hDvT7U5Ku5XlOOY7Nz2Th1onxHZueyeWqyP6
77/PjcfDZMTuTo7Hw2TZ2or6bczZjt9+n+2F0qT3PcYl3l2EC6TfmbftDU94Hjczlw2Qme/P0g725rmapn+/YOt3vm+dtHM
Yl5umkqOz6R/Xbj/ou0vjtLnO1g0lh47XU/LTtEEzFps+1ZzXHjtNn2jSabab6zZ9r2mLZj5rtvcdnsmlDm5PDOXLdOTp5zsf
pT+WetC5MJOXzdOTp/TtifGa1LEcmprnPsmbFXLxladxMjj+Ye5v5qbl2vy56ZWvadR81cw3GyaT43Mn+s7WjWM5dGyxP4
Plv2rRpi+b6zXz2tEs357P3HzlxbY81C7X056FmPmvy1Zzfvc2bdt83vTxpv6bZ4Jmmnry31x36/R4Ds0snjve2OvSa+5n/Tv
t5+MhsLjtvQ0pZeoZvnu33H5nJlukTz0/bt4znocMLJ34/HC+ZWagn5sHJ8RybO/E8+oxLtubzDx3O4ZnZTI6fGHePtp4Pm2f
u3Za/OnS8/zvPzPxSHT9ydDbnbTzRvt6bTs9OZYt05P9Z/vmufro3Fw2Tk5mcbFmbKzk0aOzS8+NrXI+cPB4Lu79bjO3UP
JG+stX7LSunu3r277tmz52ud3cfEsWNz+YNP35+fue3TOT63mA2TY7n5+qvygTu/ku969hW59qpLVwyyvOeO+/MT7/x4//
vEJQ39ZXynvL376JfmjOx846fPl8nQ6dTSYzKp10xzf5Ofn3nNnXrb7ivziB79wUjnf9MEv5Ev7j/XL8cHP3p+XP/dJl+u0ue6w
d8Kd/9nP+/vXMPsquq0vi3+p103ulO0jySgOkE7QaS0ESsUSoExvIRO0yJBErlaUVHNlXtLqNOITGtY83oDCoClgNookyMoIE
v17RizY7DWfM2v2liju+1tXXhked3ZY3T0tyAFWacuN62n8/PqQ/6TUdHNx7e+Aa+9pu/eM+L6/2FK6ZiSHKjwXfL2XaLprb
qyl53X4mjR2Cz86egkWPbMqywfB9R9A0udY7Z1buOedNyBlv0tghuGX2FHztKUI1Bua1Pu/0unFOF8ZU7V72SVw9r//Wt+5
7evfAVTxw3Dte+ehMWPxmVoxMu7D+ld40blyPby7kOonzAqa+4XzW3Ar9b1zu0ts6dkZPXpaG3n6hKPDZ+PxW3u+kw+
m+6vbbNGkkPtY0MWD+n9oc5YxC5LF/P3PpFDy1eRcuf2eddz598rpyVpWX4NbL67H0jztw8ERHRqYr3lMx4wM2v7m+6
8EvKReD4KyZkvZuj2sDH359cOeO2QJoYWNzdg98FjqBs9zLsehGLajf3r3jMzu4+047YnXs60+ca8RtxhrpdCtr/L8cm9hyN
S9l73n88z/t89pbZU4L2X76mFZ+dPSUzb2n08+XPbN8f7b12Ca1HvhhyuSN0jWTXh/KyEnzl189njin39lw1Yt2M/ms6pxZ1F
d9x4tc83vFjeobyqW23Zvu+HB9vaW7E+xtri/Ymj4isVdWmnP2n6gaPGXQygMcCN3h+BGCvqi4z25sBzFbVXfn6LOYbPM
AAFa9JREFUf9ga3Lb9hMYPjbvg0nNx+5NbsvYtX3AJLjx7dMGyL19wCebf80xqmQq1UVrbxPWYf88zGfl8ct/51JbM9revuh
C57l6+3zSbWNI3fDawaw5Ds2Nq/vPbn43miaPSe1LVvekaycaeP1/hsuxk0/eS7vnH3pwQ0546XtP9Svq0+SL1o5k+ya1g
V9+QD4Z8zvbb15hwYzifX3H3X5D/uDmy7Ty2L+hfnTx/8iXw9OuSUnxHZqbtHNa6Fy6a1/Ijm67kHxJcoT0SJlvdH6SD4fn
80nae1g5UjyAbffQq+PQz6TNOe2v7T6heydlGPSzkdSLrP9+65HQ/HrjpU2npLWSuurIV3yreHxtT9pjpLWB/f8pOu4QuYm
092Hu0LeH8Nu/5lsg6321nj4e2bT+h8UPj9jj3Bts6e7D7cGF9WFI2HfYfT7JJ2nHS2sa2t/KEjotkb59s70o1btJ4SeMcOt6Zqp
/1p44Z77x0vZvz0/SJ6SH3T5wzO8Trl3T+mDouB1nz1G/zyTZKe3+nNj3xAYA73yHZNgXkDm0351b229Qx8D+kM+5uoW
LztrX8gmbru0sZBWj1C7kFw2zxXis/nkzMH1Njyn439DdijkOjNNTvPpYdv1aGFraMgWexN8LiTroROdWeOEzdrUaHx6M
PUh7doX9MEEW7sxGWOxynN7jrRjoFEUL1IW1XtUtUlvM2pra/tbnD4zfkQlqsqzTV5VXojRQ8tRVV6CccPzPx42fkSV93xVZ
+zx0N6DKkoy9q28+b63oSqj+eRO5/f1CX4jau3zyfdNlbWul3yzY2r+/gRhflSXpdcCfCqP39phlXnnbGhFWd7xkrbj5yfpk0+P
om+DbtQGfi9uzLzHh5r18dk1jh9Bxm5cLlcf+HR3lt4XGgPWPfH2mXZNGJcR30tykmdN8tsqR21n73OM2xnzt3oocNQE93
L23e9a1h+cYJ5Ye0/ae5FknKb4XoF7J3bcJ8p50PmxOS8r/vejQeZ33x4Xw6+WRwfTVtPvCt/fnGCX7HCOsL4LVQyAcTbO2L
qnHYxt3Zp3vtnt0w05888rG4HZVeUne8X3j/sfHppN+3LCsfd+8shENdSP7JHtD3cic42lskmacQmxj21t5Ht2wEwn1Ofo+
+dXzcyZ5yW2Di++c/nN+efMRlt8xqD58X1/slVU1FTXZHVdtHchhzbWfkb6kbgm1c2ZubENzctju4t8xpxwRkjg77k2s7qvm
bEn1wyeqtOf21NPeOs7F1f87xkD5xO6XdH9fP9Y/FnnmM2+k/n34Viz7iytCIFZv8vvPEpty5XzQ3GtPGQVzWkO18usRjw+
FCJP3D4PPLPNq39IXl9M3Xp5PX697vUsmXw+4MoaygMP5InvRXOjuWnxzl3t76F1ralixJffPpkne+3zx1hc7p8Zm/rks76VZ
F0vhWzv+qSVJTSmT1br83EdQz77+oEjWb5TaF5O8n039nz5xpefXTssNXnl5wO9+aExZ27TXAPE83zoWitfnMfnLY1+i+ZC
nX6LW5Hd8X1761z/qg+x0nnn+S9AzlU9vO5+MtZ2Y0QgW75rFQH++g+fDAD6H3pcs366qs5L6LOZ38AC5VbSqqyQ03hv
tXj2SbBrqL6R/vlpWzbBKlJUA12JvtNLKb/e7VbQqy0vQ0aU42taBmDWVmFozDjv2HMXx9s5M9ZiaYZWoLBP0aFQFqaKsJF
rQkjqnBBolrWniNtKC8tQxtXV1ZFoWAVLVO5xlbssFW0bGUHW3Hj0lloJk6u6j2Dti6UI5Xg0MIOjB5agXeZ6gHxORtbHa6i
OoucXulqqlJqtYfWyFlqu3rdTUq0dUXStYRcupDmSrClm9eqvzmKoypkJlporWsXacNcpU0TJV6kabKlq2monty62i5VZhYlT
l7q0xVVVZsVRY7124VLvUBxNrBhrfyWf3cSnJudRSb9zK+aiq+uVW03GpmtopWry9HVV5sNSzX7lY/K487r9ZHbezY/ZnqX
cbevJkctotWbh6LzrG/YOcxU0cpUEcyuqmV9z/qijQ23cpH1NetjbgUfmwPcKlp2znOqaDKv9aw9MznFxGZSFS07HtADoC
zFTRMnnlrajlbZtTRctC7h52fp+ThUts9Zmqmg5MTGsUnAsTRUtxw52Tu1c2PFt/rFvtKz8SVW07La79lo72ePWx639bftM
qtoTRhRhVjZDZ9TRet4O2qqK1EzrBT7jvVWG7VvtHrzYClOdvRejzZkqmh1ZIUZPXSyE2Ni14f2mn9MdSkOHI/1b6pNdf0
5Vlljg8MmOTMXKuj62ipb9bnPIVOuKqmhVYlrt8AFZReu0v2RZRjYBmA2gBsAeAlaAOAqt5tyqTfAeADiMqk36iqiXduiv0
QQgghhBBCCCFZil9yfKpQET2AdjR33KQAUUNgP39LQqhf4UwNgjxw9ggxA9jg5BcGBfkVDBJVVwdnUV3g4eQtxsRWeN
fhgbhPhhbBCSC+OCnDb4Dh5CCCGEEEEIIYSQlodP8BBCCCGEEEEIIYQUObzBQwghhBBCCCGEEFLk8AYPGRSISKmlrBeR
4eljBKRBOXkjRF5UUTew9gggx0RmWbWC/vvllj8A2ODEEBEviAim0TkLyKyTESq+H2D9Ae8wUMGC7cCeDG2/W8AvquqU
EGNaq62awX0wFcBOAEglfA2CCDHBE5E8BCAE2q2gigFMA14PcN0g/wBg8Z8ljIWQA+DOBesy0A5gB40DRZAuBK83me2
u8PsG6Qd4g4cMBr4H4EsAesz2WACHVLXLbL8O4Ezz+UwArwGAOX7YtCdkIKIAHheRtSKyWOWbr6q7zOfdAMabz5nYm
8lfle1V1JqLH6G8RkUvjB1VVEd0ElmQwUQZgloAfquoMAMfR+5MTAlwNMrgx7xPbVbL9xhjgwxGzHun5iH6D4lZAFQD+
hF683evaZ6JDUM8bggZSLwO4HVV/ZPZfhDRDR/GBiERHwSwTIX3mG3GBhnsXAFgm6ruU9VOAL9G9B2E3zflaYc3eMiA
QTMmBQ1d0AXhORaWbX5QBeAGODEMu16P15FsDYIKQVwCUiMtS8S8euG/y+QU47Ql8igwURmQ3gi6o6V0TORfREzx
RCTnliMh0RC/mrwCwFcCNIp5TjLFBjXmPwRaAZyrqofNPq4bZNAjlosBzAfQhei7xacQvWuH3zflaYU3eAghhBBCCCGEE
CCGEEEEIIaTi4Q0eQgghhBQ1ljjBRH4ulq+KyFoR+a2ITO1jXz8RkavM53tF5F3m81eddv8slptEZKOI/FIE3v3WNSGEEEEII6Ttl
JOMhBBCCCEWPsFDCCGEGGLmMgCdqnq33aGqGwCUisjvReQRAC+ISKmlfEdEnjNP3XwaiG4QicgdIrlZRFYAGGF7EZfVlt
85b72lzDs9jiSEELIqIBP8BBCCCGkmGkEsDZwbCaARlXDJlLABxW1YtFpBLA/4n14wBmAjG4F2Invp5AcD98U5U9csi8jlv
ERgF4VkrWqOrxt8dUhBBCCBnl8AYPIYQQQgYqz6rqNvP5/QAusO/XATASQD2ASwEsU9VuAG+Yp2nyoqrHROQiAO9D9
AK4KHIs/+SIAPq+qv4s3EJEP9WVQC0NoFYBVlvi8gOsRfplO66MH1XVzX2RiRBCCCGDG76DhxBCCCHfZEOAleYnWAAAE
b6bOL8D8Hnz4milyl0RiEEEEII4RM8hBBCCCLaVVFV5O8AfE9E/glAG4DtAB52mt4LYDKAdeYgyj4AVyKqWDUH0bt3WgH
GEEEEISEVxtbXkIIYQQQgghhBBCyFuAP9EihBBCCCGEEEEIKXJ4g4cQqgghhBBCCCGkyOENHkIIYQQQgghhJaihzd4CCGEE
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])})\n")
  ],
  },
  {
    "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": "iVBORw0KGgoAAAANSUUhEUgAABHgAAAGoCAYAAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAABAAQAAABIAAAABG90bGllIm9yZy+WH4yJAAAgAEIEQVR4nO3dfbRdZX0v+u+PBEh4SVCIJAI1evCIEBQI12q1NvWlgi8Eq5dKj63t8NQ7t/PmMwsudc67ef3177GWvN/eWZc1ZrLQAAAAAB01yHDbgAAAACAR0bAAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAAAAKDjBDwAAAAAHScgAQAAAOg4AQ8AAABAxw4AAAAADpOwAMAAADQcQleAAAAAgI4T8AAAAAB0nIAHAAAAAoOMEPAADi+ABBMrjxo259oab8+ARjx52K0O34N6tSZLN93nrmUsL7v3BsFsAAICB81cGMHAPHvHobH/KS4bdxtAt/sYVSeK1mB0n4AEAAADoOAEPAQAQAQMjCjEAAAAAA6TsADAAAAOHECHgAAAIcOE/AAAAAADJyABwAAAKDjBDwAAAAAHScgAQAAAgAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8Q/De9743733ve4fdBgBAJzh2AoD9WzjsBuajjRs3DA6DgBDwAAAEHCHXgAAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8AAAAAB0n4AEAAADoOAEPAwAMAAADQcQleAAAAAgI4T8AAAAAB0nIAHAICxs2bNmt3/DbruvPPOy5o1a/Ka17xm4GP1U3PuuedmzZo1Oe+88w6ov7FmzJn/4h38465q3ve1tWbNmTd7xjnfMumb9+vVZs2ZN3vWudx1Qf5dddlnWrFmTT3/607Ou+elXv5g1a9bk6quvPqCxBN23DrmT6d1xxx1JHupztrZu3Zok+eEPfzjrmn5fi/vuuy9JsmPHJlnXTPZ1oH+EXnfddUmSr371q7Ou6ffn+uxnP5skBxSg/MM99fngBz840JoLL7ww99xzTy688MKB9zeXRr2/fvX7++oSAQ8AAGNI79Uts13t0k/d3qtiZrvCo5+x+qk599xz99ie7SqetWvX7bMOQyy67LK21JElrbVYh1Be/+MXs3LkzSbjz585Zr6zZsGFD7r777iTJ3XffPatVPP2Odeedd+bKK69May1XXnnlrALAfmo2bmzLxo13ec+FDtu4cWMWL1487DamtfegmFFbxTO5emfSbFfxTK7emTSbVTz9vhaTq3cmzWYVz979zPaP0MnVO5Nms4"
        }
      }
    ]
  }
]

```

x+x1q3bt0e2+985zvzmc98ZiBjfeADH8iuXbuSJA8++GA++MEP5jd+4zcOes3eq0AuvPDC/MVf/MVA+ptLo95fv/r9fXVJ1b
rLo16f/3q9/fVNZ1YwdNauyTJJUmyevXqzi8FOOGEE5lk73nPe4bcCQze+eefn2u+ffuw22Ae27VoSU5+wwHec6HDrMCDg
cuXA+ptLo95fv/r9fXVJ1bwAADAKFqxYsUe2yeeeOKQOpneYx7zmD229+53JkuXLt1j+1GPetR+a/p9LQ4//PA9thctWrTfr
ZvrEe19jaDZnAqWPPWurb2vG3Qwx3rta1+bQw6Z+DN3wYIF+aVf+qWB1Lz97W/f5/bBHGsujXp//er399U1Ah4AAMbK3
QnZjXWH/3RH+2xPZu7R/X7c/32b//2Htv7u/5O8vDrzlxwwQX7rdn7rLSzDULWrl27e9VOVc0qgHr+85+/eyXNwoULZ3VNr
XCHgAAOARmFzhMWqrdyZNRuKZ7eqdSZOreGazemdSv6/F5Cqe2azemTTZ14H+ATq5imc2q3cm9ftzTa7imU14MmlyF
xppLo95fv/r9fXVJzeZiWqNk9erVbcOGDcNu4xGZPI/c9SCYDyavwbP9KS8ZditDt/gbVySJ12KOLF7GFTnDNXig0xw7AcBD
AAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8AAAAAB0n4AEAAADoOAEPAQAAAAQMjEAAAAA6TsAD
4T8AAAAAB0nIAHAAAAoOMWDruB+ejkk08edgsAAJ3h2AkA9k/AMwRveMMbht0CAEBnOHYCGP1zihYAAABAxw4AAAA
xAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8AAAAAB0n4AEAAADoOAEPAQAAAAQMjEAAAAA6TsADAAAA0HECHgAAAI
AAAAoOMWDrsBYPwtuPcHWfyNK4bdxtAtuPFOJPFazLEF9/4gyfHDbgMAAAZKwAMM1MknnzsfkbGbbftTJKcclKwYW4
+ABAAAA6DgBDwAAAEDHCXgAAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8AAAAAB0n4AEAAAD
W8YDrmBTMxN5iOecF0zAtmYm4wHfPi4HGKfGAAAEDHCXgAAAAAOk7AMz9cMuwGGEEnmBdMxL5iJucF0zAumY14wE
q+npvXqzv7X98VX2lqjZW1V9V1WHD7pXhqKoFvFW1qvpMb9vcmOeq6paqur6qrq2qDb19Pkvmuao6pqp+XlXfqKqbq
krw1yRdaa09M8oXeNvPHfUme31p7WpLTK5xZVc9KclGS/9JaOznJD5O8bog9Mlznj7l5yra5QZL8TGvt9Nba6t62zxLek+T
XVvF8cNAKeMdJa29Ra+2rv67syceB1QpK1ST7Qe9oHkpwznA4Zhjbh7t7mob3/WpLnj/l4b795MU9V1YlXprkz3rbFXOD
PxfnHQCHjGVFWtTPL0JF9jcnxrbVPvoc1Jjh9SWwxj7xSca5PckeSjP+a5Ee9N9YkuTUTYSdZz7uT/FaSXb3tY2NuMBECf76
YqjoiyUuSnBTvFweNgGcMvdVRST6R5E2ttW1TH2uttUwcuDOPtNYe7C2dPjETsYKfMuSWGAFV9bikd7TWrhI2L4yc57bW
1m7OxGl6n09yZZJrkzy413O8XzwCap4xU1WHZiLc+e+ttb/u7b69qlb0HI+RiVUczEO95fRXJ3l2kmN6S2aTieDntqE1xrA8J
PJvrbUtrbUHkvx1Jo47HGPMc62197fWzmitPS8T12H6VrxfHDQCnjHSu3bG+5Pc3Fq7eMpDlyd5be/r1ya5bK57Y3iqallVH
5XVU/u7XpBkptiXvCQ8/LQ6VmJuTHfftjs6rqiN7fKjPvGY4x5rmqekzv3x/LxPV3PhzvFwdNTayAYhxU1XOTfDnJ9Xnoehp
tvzczmN96v/9P9jYXjvlwa+1dVXVsfjBma1V1eiYuyH5Ykm8n+ZX0PldiXsrxvTD4u0me0Frb2tvnPWoeq6r1SX4+E3f6/VqS/
cQleAAAAgl4T8AAATFFV51RVq6qnDLsXAIDZEvAAAOzpvCT/o/cvAEAnCHgAAHq6qgkz03yuiSv7u07pKr+pKq+UVVXVd
Z2R5M+TvGsYtQMALBx2AwAAI+S8JO/pff3R3vbCJB9rre1Ksrmqru49/uQkq5jCvVvJsiDjPrIltFwBggoAHACBJVT06yfOTnFz
X7IK2qOnUYjQMACHgAACl4ev1vIEkuVjbk1yU5IPjflqkq2ttfszEQpdVFVfT3Jtkp+cu3YBAB5SrbVh9wAAMNKq6qjW2t1V
AAAAAdj+ABAAAA6DgBDwAAAEDHCXgAAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHTcwmE3cKCOO+64tnLlymG3AC
fnhV/Vx8a9U1cpB9gMAAAAwjZ2DZ6qWpDkfUlelOTWJP9cVZe31m6a8rTXJflha+3kqnp1kouS/PygehoFP9q+I9/afE9u
Qj9XOP6undhrLu278jNU+p+fPmROXo/dVu378g3p9Q8efmRWTqgObh9+wO5fvO23XWnLV+SxYsP3WfNjh07c/2mrdm
HUzZVR769fO3fuyo2btmbT1h1ZsXRxtI2xjAsXjt8jTYO8yPlzk2xsrX07Sarqo0nWjpkA8KxNsQ739ceT/HFVWwutDbCvofn
312/dtu07cuU0NWeuWrbPP7C97uM/1l3bd+Sz09SdtWrZjCHP1u078rlpal68atmMIU+/c3D79gfy6Rs2P6zu5auWzxjy7N
5qrmkdTNIVHvr187d+7Kp75+W97+qYfm04XnrMo5Tzth7EkeQf40JyT53pTtW3v7pn1Oa21nkq1Jjh1gT0P1rc337H6TSpl
rqb91H3zRlqvjmAOXj95m3T1l2/edvMNZu2TI+zaes+x7ru+1t3hzu76y67ldd9f+a6fse65c57docTk3VvvTa3HLnzK/HXNU
3UmnEda9T7G9exRr2/cR1r1Psb17FGvT9jX9/4zrWqPc3rmONen/G6lZ/m/sca/O2HTPU7TjoY90+w1h33DXzWHNV80jq
oni/5okMPyFLDj+oNeM61qj3N65jjXp/4zrWqPc3rmONen/GGv/+xnWsUe9vXMca9f6M1a3+lvc51ooli2aom/kUsn7HOn
MuU5S5P87Lp16z6zf36c5Msaq19bF/f95jLLIn3+te/fiA9D9qRiw7Jjz3qqHx545bs3NWY6NCJc0mf+YRjsujQ6c8l7admXM
46aYayfGMAcPSiQ3PCo454WN1PpEHYHHro9NfFOXbxYXnsNDXP+w/H7fM6jscdcVhWPGpxvwwU+rWsrpPn7wsCxZM
79+k3r1q27ZO/9NcjrGVfVS5K8O8mCjH/eWntXV2QZENr7fKqWpTkL5M8PckPkrx68qLMM1m9enXbsGHDwHoetHG9
jD+yOWIOueSR1c2XU++vX5F20Nm/dkeVLF+XUFUs7fYHlqrqmtbb6Yfu7dsOqrgc8AAAAAP2aKeDpbmQFAAAAQBIbDw
AAA6TsADAAAA0HECHgAAAIcOE/AAAAAADjYABwAAAKDjBDwAAAAAHsfGAAQAAOg4AQ8AAABAxw4AAAAADpOwAM
XjeVZ2y19O+lmR1a+2pST6e5P8ZVD8AAAAA42qQK3iemWRja+3brbX7k3w0ydpT2itXd1au7e3+Y9JThxgPwAAAAABjaZ
1tcwAAAAAjbueAv/dtSU6asn1ib98equqFSX43yU+31u4bYD8AAAAAY2mQK3j+OckTq+rxVXVYklcnuXzqE6rq6Un+a5KzV
VSa7Ya987pnz9wkGODwAAADAFjMRfIgEAAADon4AHAAAAoOMEPAAAAAADj+ABAAAA6DgBDwAAAEDHCXgAAAAAOk
AICOE/AAAAAADjYABwAAAKDjBDwAAAAAHsfGAAQAAOg4AQ8AAABAxw4AAAAADpuoAFPVZ1ZVd+sqo1V9dZ9PO+VV
5fkoiQ7BtgLAAAAwNgaZMBzQpLvTdm+tbdtv6p6RpKTWmt/s69vVFWvr6oNVbVhy5YtB79TAAAgA4b2kWWq+qQJBcn
093lq7eB8P/3OSJ1bV4zMR7Lw6yS9Mqd2a5LgpY/1dkt9srW3Yf9sAAAAATNpnwJOJVTZJ8uQk/0uSy3vbL0/yT/sqbK3trKp
AAABg/qmqA1prD7u8zWyvwXN8kvunbN/f2wcAAADAKO3vFK1JH0zyT1X1yd72OUk+MjiWAAAAADgQswp4WmvmvqqrPJ
VyQ5O8k9Sdja+34eUoU6AAAAAEM024Dn/jZxP/WWJFV15OBaAgAAAOBAzDbgubSq/muSY6rqV5P8bZI/G1xbAAAAAMz
FX1vyf5P5I8oaqum/LQ0Un+5yAbAwAAAGB29neK1oeTfDbj/53krVP239Va+8HAuGLAAABg1vYX8LTW2i1V9Wt7P1BVjxb
Vvr9tre0FvVxMwLQEAABWlGZ7m/QXVNUVVbWiqlYI+cdM3EkLAAAgCHb3zV4kiSttV+oqp9Pcn2Se5L8QmvNbdIBAA
StNZakj+sqk8Pri0AAAAAZmufK3iq6reSpLW2rar+170e/uX9ffOqOrOqvlIVG6vqrTM859yquqmqqbqyqD8+2cQAAAAAm7
rX27tXZ/ko8mWbvXc341yftaaz9MktbaHfv5ngAAADsZX8BT5vh6+m293ZCku9N2b61t2+qjyV5UIX9z6r6x6qadIVQVb2-
vX+giUAAACAEWWfAU9rbCej+N63JtlpyvajvX1T3ZrkK621B5L8W1V9KxObj1uwAwAAAMzS/k7ReiT+OckTq+rxVXVYj7Yf
bW7hxUTwAAADJqFrr1ivTq9e3TZs2DDsNgAAADmXFvd01pbvff+QZ6iBQAAAMAcEPAAAAAADjYABwAAAKDjBDwAA
A6DgBDwAAAEDHCXgAAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAHScgAcAAACg4wQ8AAAAAB0n4AEAAADouIEGPF
AAAxtUgV/A8M8nG1tq3W2v3j/lokrV7PacIwDl7emmS7w+wHwAAAIcXNMiA54QK35uyfWtv31Trkrymqm5NckWSN0z
U6asn1ib99Ur0tyaZK01v6/JluSHDfAngAAAAADGziADnn9O8sSqnVHZAjyihfvtDzvpvkBUISVT+eiYDHOvgAAAAAB2BgA
VNSZ4zyB4AAAAAxt2wL7IMAAAAwCMk4AEAAADoOAEPAQAAAAQMjEAAAAA6TsADAAAA0HECHgAAAIcOE/AAAAAADjY
AAAAAdj+ABAAAA6DgBDwAAAEDHCXgAAAAAOk7AAwAAANBxAh4AAACAjhPwAAAAAJs4qG9cVX+e5GVJ7mitrZrm8UryniQvSXjvkI9
tiMrIizKaY9dmsMOW7DfsX60fUe+tfme3L7tvhy/5PA8afmROWbXon3W3L3v9/tyw+a7dNauWH50jFh920MeZ67H6qRv1s
eZtu2tOW74kixcfus+af06Bkv6OI3fs2JnrN23N5m33ZfmSw3PaiqVZtGj/f27081r0e3zcT12/YzHezAtG0cACniR/keSPk3xw

JzT8+Zpy7f5xtIP2Pdf/+D+dR13887Lnuo5oK1q3LOUx+7z5DnR9t35PM3bMk7Lp9Sd/aq/OyqZTMeFN27/f585obbH1bzs
wupeuesyMwcb27Q/k0zdsfIjNy1ctnzHk6ecYKOnvOHLHjp25/PpNDxvr7NNW7DPk6ee16Pf4uj+6fsdivJkXjKqBnaLVWvt
131/6+5wZ7LmHZfdkOu+P3NNknxr8z27P5R3111+Q761eeYeb9h817Q1N2y+66COM9dj9VM36mONen/jOtao9zeuY41
6nrDyzGm3nBqBrmNXhOSPK9Kdu39vY9TFW9vqo2VNWGLVu2zElzg7Bp647dbwKTdjyWk5u37pix5vZt09fccdfMNf2Ot
OqExdZbq1d0lPb3VpbvWzZsmG307cVSxdn0aF7vuSLDj0ky5fOvIT3+CWLPq15zNH7Pre7n7FWzDDW8Uv2PdbxSw6foV
E3r6xdeqKjbnwnFW73wwmz2c+dcXSGWtWHntkLj739D1qLj739Kw89siDPtZpj12aC9buWXPB2IV56mNnrkmSjy0/Mhec
UdPWnBr8qBlrTlu+ZNqa05YymbGmn2OgpM/jyBVLp+9vHzVJf69Fv8fH/dT1OxbjzbXgVFVrbXDfvGplks/McBetlyb59Uzc
RetsbvbXzj2N65jjXp/4zrWqPdnrPHvb1zHGvX++q0b97toHchx5ORdtHb3Nwd30TrQ4+N+6vodi/FmXjBMVXVNa231w/Y
GNDwAAADBfdOliywAAAADMTMADAAAA0HEDvcjyIFTVliTFGXyfhXNckn8fdhOMHPOC6ZgXzMTcYDrmBdMxL5ijucF0zl
2A4wk84LpmBfMxNxxOuYF0zEvmlm5wXTMi4PENXgAAAAA0s4KHgAAAIcOE/AAAAAAdjyAZ4xU1UIvdXVV3VRVN1bV+
VddW1YbePp8l81xVHVNVH6+qb1TVzVX1bPOCqnpY771i8r9tVfUmc4Oq+o3esecNVfWR3jGpY4x5rqrO782JG6vqTb193
hmu85PcPGXb3CBJfqa1dnprbXVv22cJ70lyZWvtKUmelon3DfNinmutfbP3XnF6kjOS3JvkkzE35rWqOiHJG50sbq2tSrlgya
Defy3J85N8vLffvJinjurEJC9N8me97Yq5wfR8lsxjVbU0yfOSvD9JWmv3t9Z+FPOCPb0gyb+21r4Tc4NkYZLFVbUwyRFJNsU
j3htrktyaiTCQ+efdSX4rya7e9rExN5glgT9fVddU1et7+3yWzG+PT7IlyX/rndL5Z1V1ZMwL9vTqJB/pfW1uzGOTtduS/EGS72
iSWRTxlyS4yAqnpZktaa9cMuxdGznNba89lclYmTvd93tQHfZbMSwuTPCPJn7bWnp7knuy1hN68mN9611I5O8nH9n7M
MQqDuah3nL6q5M8O8kxvSWzyUTwc9vQGmNYnpPk7Kq6JclHM7Fs+j0xN+a93v95TWvtjkxcS+OZ8Vky392a5NbW2ld6
1Jbm6tXTzlocuTvLb39WuTXDbXvTE8VbWsso7pfb04yYsycX2mq5O8qvc082leaq39TmvtXNbaykwsq/9ia+0/xyY16rqyK
PZeL6Ox+O94uDpiZWQDEOquq5Sb6c5Po8dD2Nt2XiOjyXJvmtxJN9Jcm5r7QdDaZi5V1VPzcTFyhZkltS9tLV2QVU9IROrN
tV4Y/N0kT2itbe3t854xz1XV+iQ/n4k7/X4tyX/KxDV3HGPMY1X15Uxc8/GBJG9urX3B+8XBI+ABAAAA6DinaAEAAAB0nIAH
AA9VXVUkucmeV2SV/f2HVJVf1JV36iqq6rqiQp6Ve+xM6rq76vqmqr6XFWtGGL7AMA8JuABAHjI2iRXtta+leTOqjojc8IWZr
VJrqqqJFmQZNPctgsAMEHAAwCQpKoenet5SU6rqpajwKYl+eRMJUlubK09e45aBACYkVO0AAAmvCrjX7bWHtdaW9laOy
qrye5NslPzI27AAAPqdbasHsAABhpVXVUa+3uqjo2yT8leU5rbfOw+wIAmOQaPAAA+/eZqjomyWFJfk+4AwCMGit4AAAA

"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'\n# of Bivariate Outliers: {len(file.loc[file['Age'] > 87])}')\n"]

]

},

{

"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": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLLAAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAEIEQVR4nOzde5wcV3nn+/TXX2fi6TRWJbli2xLaiyZcFEcQrJAfMPws2QnEGJl1mwCL
RK+rxfl72mu+pU1VNV55yqfITdx5xzAgAAAAAAQHhF+h0AAAAAAAXhgSPAAAAAAACFHggcAAAAAACDkSPAAAAAA
7203nK1oAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgIAjwQMAAAAAABByJHgAAAAAAABCjgQPAAAA
AELO63cAp6JPfOITGhsb63cYAlA+2LdvnyRp1apVfy4EvbRmzRpdd911/Q4DAACcQkjw9MHY2Ji+/+hjqqSX9TsUAMaii+Yr
QAAAAAACDkSPAAAAAAACFHggcAAAAAACDkSPAAAAAAACEHakeAAAAAACakCPBAwAAAAAAEHikeAAAAAAAEKO
AAAAAAQo4EDwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgIAjwQ
AAACakCPBAwAAAAAAEHJevwM4Fe3bt0+RfK7fYQAAAAAAcEr4xCc+IUm67rrr+hxJ75Dg6YPZ2VIZtdTvMAAAAAAOCW
DwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgIAjwQMAAAAAABBy
wAAAAAAEHikeAAAAAAAEKOBA8AAAAAAEDikeABAAAAAAIORI8AAAAAAAIUeCBwAAAAAAIORI8AAAAAAAIQcCR4
AExsbGIEql+h1GT4XiCR4ze4+Z7TSznePj4/00BwAAAAAA4IQSiid4nHOfkvQpSdq4caPrczAAAAAdVZNDWnPeCt122239D
AACFHggcAAAAAACDkSPAAAAAAACEHakeAAAAAACakCPBAwAAAAAAEHikeAAAAAAAEKOBA8AAAAAAEDikeABAAA
QMiR4AEAAAAAAAg5EjwAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgIAjwQMAAAAAABByJHgAAAA
DglrFmzpt8h9Bwjnj5YtWqVDhSe73cYAAAAAAcEq677rp+h9BzfEULAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgIAjwQ
keAAAAAACakCPBAwAAAAAAEHikeAAAAAAAEKOBA8AAAAAAEDikeABAAAAAAIORI8AAAAAAAIUeCBwAAAAAAIOR
HAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgIAjwQMAAAAAABByJHgAAAAAAABCjgQPAAAAABayHn9DuBUFc1NKPW
AQAAAAAAnET4DR4AAAAAAICQI8EDAAAAAAQciR4AAAAAAAQo4EDwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAh
AAAAAAACFHggcAAAAAACDkZDnX7xiOi5mNS9rb7zheBMSlHep3EMAJhnYBtKjDAK1oF0Ar2gXQ6mRtF+c450abj4Yuw
0ArWgXQCvaBdDqlGoX/AYPAAAAABayPEEDwAAAAAAQMiR4AEAAAAAAAg5EjyLzMyuMLPHzWzWzG7odzxAL5jZ02b2
2gBNBh3Zxk5ntC64Z3zezNzfM+/2gjj9uZm9smN72fsrMzjWz7wTTP2dm8WB6Ing/FsxvfTh7DMzPzM4ys2+a2Q/NbjeZb
+qTk33BI2irppyRdJGlrw03HjyX9esNyV/R+d4Dj9ldqrZuL0Q46bQM4EfyV2vfZHw+uGa9wzn1Zkoj7pGskRQ+W+TMzi85z
Kekq/ocE7BYrpj0R/D6DkIXN0z/rPN9W9ISM1sp6Y2S7nHOTTjnjki6R9IVwbwh59y3nf8r8Z9tWBdwwnDO3S9pomnyYrSD
sGTmFd2kUnXDNEgmexrZL0bMP759S9kgJh5SR93cweMrP3BNNWOO2F68PSFoRvO7ULRpNf67NdCAMFqMddNoGcC
/Hrm+RAacIBajHdDWEBKfHS+pFdI2i/pv/U3HKA/zGxA0j9l+m3n3FTjPK4ZOFW1aRdcM7ogwbO49kk6q+H9mcE04KTin
s65g/nBQHjghmFIM/ofYv3XO/e9gMtcMnNLatQuuGd2R4FlcD0paG/xad1z+j0Bt73NMwlvKzDjMnIh7LelySY/Kr+u10Rze
Nfla4LRTM6V/8Ow/6oO91PB0wfflPTWYPnmNlZrF2+V9I2gPNB3QT/+0t6zDl3a8MSrhk4ZXVqF1wzurOQxHnSMH8Yt/8
7inPufwfSN8kdiSU6iqTrwtLp4tRhZn8v6Q2Slt6Xv7lJl9Uj9tBp7bW8x0GFqBDu3iD/EftnaSnjf1G7TdBzOwPjf2a/NFUfts
zcErq0i7eLq4ZHZHgAQAAAAACDm+ogUAAAAABByJHgAAAAAAABCjgQPAAAAABayJHgAQAAAAACDkSPAAAAAA
qhmSdESSzGzAzO4Lnup5xMyuai7cqYyZrTazx8zs02a2y8y+bmapYN4aM7u34Ymh84Ppv2tmD5rZw2Z28yLuMwAAOM
Nl3JP1X59wXzCwp/z/VflBSWyX9hiSTtF3SHzn7l/MYwEAAE5OfEULAACcChq/ovXTKj5rZhvkj1r+i5m9TIJv0ipJkYQdaFi2U
05+Ptq51zjz6W/5RPo1/uUqbQUK4iKdVl0ybpo865v3jhewEAADAXv8EDAABOKWb2UklRSYclDU56GCRufk7+166aLaRM
/arPsQso0+/eS/sLmtkkqSfpF59zXzexlkv7FzCRpRtKvSDr4QnYMAABA4keWAQAAAAAAQo+vaAEAAAAAAIQcCR4AAAA
AAAAhBwjHgAAAAAAgIAjwQMAAAAAABByXr8DOF7Lly93q1ev7ncYAAAAAAAI+6hhx465JwbbZ4eugTP6tWrtXPnzn6H
JyMevYbPGYWIxS7pMskPSfpQTPb7pZ7YUOxd0k64pxbY2bXSPqYpF/qVUwngqOzeT1xIkvnwpwpaMZTQutMzWpJk9jusF
TNV5SIRTSZL2okk9T6IUPyvEhL+VyxohVDSZ05nNIPD0xp39FZjQzEIY7ms6XNTqQ0GyprCOzjeVLFZ02mFasGtF0vjwnnr
laWFsIOuWNbyTEIV55SMRVWuVHVopqh0PKqRgZhikajGZwpKxz1Vnb//R2dLkpxi0agOTu0OpTQklRUhZJTrlRRqeKUL
EM9AlqpcsaKZQlnDqZjiwbxkLKrhpKdSVXp+Kq/IgwkVy2WZlkrETLGl1zS8ZiO5opKxqlaSHgqVSsyRTQ+XdBosP0fT+Y1O
OudKZWViEZ1KFjXUMLTUMp0aKaia1MFnTaY0FAqqnyxqtLSRTOFigaTnvLlspamYipWpEMzBS0fSGi2WFbci2q2VFYyFlix
tKKBOP6tmJnEYGEqqqlNnRHZop6PQhvy/dP5XX6EBcAwlp04WyDs0UNZKJKx2PKh41zRQrmi1W5jxTOu7pcLaopemYf6
BoH/MhplxFSpVTc2WtSwTu8VVF94milWJFUVj3o6NOPXi1KlrLjnqViuKlsoa2kmpohFdGi6oHTCUzzq9yFxL6J82d+vdMx
LMvMNJDwlEIENFusarrgt5HRgbjKVafJ2ZKWZWlyZzqaL2koGdNUvqTBRExTsyUNp2PKFUtKx2MqVcoaiMc0W67Wz1kml
X/ekZzfx+VLZaUTnlxVmsr7bX2mUjIXiWgw6WkgHtGR2YqmC6XgOJe0NB1XtlhSNBJVjh5R3ItqOI/W4WxRpwf90kyhplTnb
J03m8rlwMxTecrej5oL8vSUR3jVXQkv9KydExHckWl4568iEkmbj2I8uWqpmZLOmckpanZY+seTHiazBc1kdjWXkcHEipW
LGgnqz0TQ1pKxqA7nilqSismLOJWrVu87yxv/G0dnSxpKecqXyviIUUUjUiwakcnvjyrOqVj2+/wVgwnJqpLzrzFLUn7/PjyOq
jEeV9Pzr9kS2pNHBuGZLZQ0n48qXqzoc9M9LUP6m8uXguuMpHvX7ulTMUyxqGgrm1/oXp6piFtVMSaxc0b/Ol6pVFUoV
WjOb/OTs35dKZSr9fM7MuBpMlFRRK4U1PGSUrGoBhNRFSpO49N+uXTMb09LMjHNFisany5qOO1pIO737zPBvKj55ykT
zf3kYH44pFI/XzLSuWFFciSnn+NfaMpUIVKv79x1DSC/q7ikYyi73gLV7xLOXpvXskZwOzuRVKjvliHUtSceUjkd1JFdUPBrVaY
xrHg82pfPDWH4rAI06+WPLF8kacw5t0eSzOxOSVdJjakzWXCXppuD15yX9DzMz55zrYVx9c3Q2r68/Oq4btz+qfKmqZCyib
xrP1uZ3P6Jc2nq0//cbuevn3X7xWn9v5Q1138VptvAM3fv4wXr5WrmN5wzrbT95jm780rE6tnXTev39d/ZqMI/Sf3z9Gt28
bxt83pFlv6Hxtvua96XZ3TNT56tz/7LXh3JfVb103r9+T+Nae/hWSVjEX30Fy7U+HRBX9+1X2951dm6+e5jcX/4qg0ycy3rvXr
WbY9965Xr9w3ef0SUuO10jmZiyxao+9tXv1Zf9gze9VKm4pw81nKfm9f39d/bqiYmz+uO3Xqh8ybWUXZLYvKo4ff6h5/TW
fTmj/0al+9KVHO56DYjmn//KVH9WnfewtF2oo5Wnv4ZxubNjGBY5dp1QsMqfszZvX66uP7Nd0oarF3Hj2njhjqx3HVkkTLv
```

Wqngj8zW61YyFtEfveVCHZiabiabqj363P++jPX6ijXFZ//LXHTTQd16/+zOo529lyyVqNZGjyMv3Zt8Za2uAHLI2nv/vXvfqtN6zx
G+6cKXufLbZp1E7Vr/2M+fpmYnCnHa2ddN6Dac8/fm3ntRkvqQPXLpO+yendNt97c9yYzu06aufjmyxar++ttPdZ3vv3/F5
J/DC6/YLkufdnKOXVo2+YNUv1bu7X38Gzbvnmnb5vW6/Vt+f1Rb/vZvjekdf52jLfc+OXI+JskSMT35/JqMZ5e29DXL0jEVmtp
5Hu32v/FcZOJRLQ+S5hO5cstxumvnM7rsgpVzrnONMd5y9QZ94hu75/Ttf37/rvr75n6idk2Le9b1mrHlkrU6fTipUrmij987
v/0dr9Shpnlfe8uFKpSruvFLHWLddIEOZ4stx//ex/brFWeN1NtSu9i2blqvrz+6Xz+9Znm9TjQem079YiYe1dJMXDP5kv78/j
MZ7xFuu3qBqtarD2VJLv5qlRvSZB56q1/FP/tMexT3TdcE9WuP21q4Y0MUvWdFxo7V7v68/9vyc6bVl37D2tDnz2tXLW67
31xIFsPbkjSfISVTduf1RPHMj2ObIXx9OHs/VOUPL37/q7vq+nD7fu3/GU7Zd2Md56zxPafXDmuOOeb39r8698+Sr96Td21
e5hg+9KVHde1rz5sTa7uYb9y+S0nPq9+cNK/ntvt217d/845duvLlq+plnjqU1a33PKFrX3tePblTm/ehLz3adr1bt+/S1a869q
aX3vft35ypeqGk7F25bdP1XQoWxR737d+TqULdb/Ne9v4zn+r1/9kcaa6m+7WD70pUe1f6owZ9r4TKF+XjvPY+3vrfc8oaT
xib9+1QtHlP7jRu6+B0Qe9+3fn1DxeNy1/72vNa9uemHbvm1K18qaqx8c51/anDfnKnFmfzdmP1bOv2XW3b4Mfv9fd16/
OOXapWVe/D9hzK1rfT7rw1tqPG1/unCvqTrz8+73k/nCu2PZe1/fzgFx/t2j/fdt9uPTme1ZUvX1Vff/M227XTm3b4fcZNO/w
fE0ut3nXr59KJWMv8du3ptvt263Cu2HI82u1/47k4IC2qWpVKIda6deN2/5i3Oy61GGvnu14f757b1zf3E43Hpds147b7duu
Pf6//Jpz57SlDrHdvMPv7xvrROOx6dQvHsoWtfvgjPZPFz97X61Xgua22xsb/ZP1Xo2uc01oNa39dcdnzGj6fdPWBj2Q9+0
rQtiH9ZNrP3mNIOM9s5Pj7e73D+zZ5v+MBUky9V9fxUoU8Rvbien8q33b+D0/kXVLZfOsVYbXq+bCFxz7e/tflmmvO3uXx
LSGmOvHfNOcdB2rdv6ZotlSeoYa9X565ktlOuvF7Le5vrbaZmFlms+/9liWePT85+Ddus/OL3wNnc0V+pYN8w6t4UjwFfSjC
t+rV2y3er5wvtG2vHqN0+d6tL9T4jOAEs6vW10z7Odwza1ffGco19x3iX9teuDTbuo9Ta5hZyfhR3tX4cOtSDqms9Hgs5F93
Qtpuc32Zrx7XyJtAzaX7XT/1LiO5v6mXdn5trN/snPbb57XaTsHJl/Y/Xun2i7d97b/DNXLzw1h+KwCtNPLBM8+SWc1vD8zn
5W5kLjn29/G+c1/G8s75/89vU15SUonvLbLpeJe13XWXq8cTilqC4uhcVvdymfi7WOqradx+41fzqzF0WmfMsn200cH5raz
0tScc61g3nOreFpQ3Hc74YG/et23FKd6jjinaZ3OrbtYm7e7nXtrNt+ZOJex3ZTK1Ord932t7ned6t76TaxdSrXKbZM3OsY10L
PR7tt1JbpVu+79SHNr7u973RN67beWmy1aZ1i7LSObu2IUaf61vx+6TzXmnZ1ar5tzBfrkvTCzud85ear2wst3+56tpC221x
a2bnmlc0jWstjeV2S7pncHrt0r6xsn6+zuStO70jLzt3jDnZmbv5g1ad3qzm5G9OFaPZHTr214xZ/9ufdsrtHqkdf+Op2y/tlv
f1PascP9mnprvUt+/O/v/tcPZ71K4d04ZnD2nKJH8MHLI3XNoYPX7VBdzywZ06s7WLetnm98uWytLzSfl+2XLK2vv2tm9br7
5rSFK0O3bdYt965Xp99oE92nLjWuUKJf3eFS+ds+xlJq4PN52n5vX95f1PKhmLaHK22LbsyqGElmfi+vT9T2okE6//a97fxnN
uXddS9ubN6/VX//cp3fHAnpaYasex3bytm9arXK3q+svWzamv7Wjs3reRTFy/c/ILWrZ12mBCn77/SV1/WWtbueOBPS37c
vVueidfrU2250cGE1q4YbFv3Pn3/k/X+pFvfdscDexSRWtrZ1k3+73jV+rBzl2fq22l33hrbUePr04cS+p3LXzLveV+Wjrc9l7X9v
v157Xwuz8R12mBCX31kX9tzvmo42dlGa8d2xw/21eP79P1Pzollxw/2dWyftXObK5Ra5rdrT1suWauRdLzleLTb/8ZzsTzj/9
4495ieMZzUtFOcl22b16tYqbY9X7VrTW27pTbLzhvNaNtVXWLNl9oe/7/99lNz2IK72LZu8v7xjrReGw69YvLM3GtPW1AK4
463Zq937N02vLNs9rVy9vuXqD1q8c1gvRLrba8dx21QZduHJ40T83hOGzCtCO9TKfYmZvlvTfjUUIfcY59xEz2yZpp3Nuu5k
FhRrHEWrUFHci2g6X9SyTElRvVw7PGUWvR7bKFozBX/0qtooWoVStT76w3yjaGXinj86STKmYqWilQ2jaE1kCxpM+qODXc
WVpOfkjHQ0lY6q6uaNoTWSL9VGsDkzINTKQULZQCKaZiehw1o8vE4yilS1UNJTyglG0ikrGlnNH0RplqFjxR9GKe1YfCSwd
/iHnj39+BpKdCuallKa/tKFr5UkUjL6JopHUUrUMzRS1rGUXLH80kW6xotlDRQCqqTNMoWuMz/rnOl4+NolWLLxOP6tkj/i
hLKdi2Wn/VF5nDDeMopXyR4oaWMAoWkvTMVVVtwSnTuKvTbf/1KlorgXDUBRqmhpXlNE/jlNjajzRtEqLKrKlcpKxTzN5P
krSnijRtBL+ie8DtVG0UjHlpgHSSpVKMlpWpX7OaqNoHZopamkmrnLFHw2oPorWUEJO0viMP8KSuzByVzquUrWqqdmk
yjb1GLDCzmkmUJZCS+qqdmSlg/E5UUi/ghAwQhLx0bRqsiLRmTy28tAlqplmfajaNXa8NL6KFqVel9aH0VrMKFlGX8UraPB
VaUTQS1US2oJFgFK19k3l/xKymUbRqT182j6l1kPCUikU1kStquHkUrYfJl0ldG0Wrli8SUTQiedGllpo7ilatZ49YVW7OKFp+X
ZSCaufLhFK1aH5P0+7Dm604qHIUs6tfjWh83OpCQs7mjal0OJFR2VRXLFSW81IG0jmSLyrQZRetlzu+/4l4wipbaj6JO8+ye
fEXDGa8+itZAwo8vFo0oGYvU+7o5o2gNJSWT348Go2hIG0fRSvn3CI5kYaNoTcwUlamNohWNKBULRtFaklSl6rR/sqCBZF
R9FazKvWNR0dNbvX1/eNirWYn5uCMNnFZy6zOwh59zGlulhe2Am7AkeAAAAAACAF6tOCZ5efkULAAAAAAAi4AEDwAA
AAAAABCjgQAAAAAABayJHgAQAAAAAACDKSPAAAAAAACAFHggcAAAAAACDKSPAAAAAAACEHAkeAAAAAACAKCPBAV
FmT1uZmNmmdkOb+deb2Q/N7GEzu8/MzullPAAAAAAACejniV4zCwq6XZJb5J0gaS3m9kFTcW+J2mjc+7lkj4v6Y96FQ8AA
boaZvcfMdrZzvHx8RcxRAAAAAAAGPA7IX5k2cx+RdJGSX/cbr5z7lPOuY3OuY2jo6OLGxwAAAAAAMAJzuvhuvdJOqvh/Zn
bMEj3OuLOl9kr4m6TFjdznndpnZNjPbHBT7Y0kDkv6XmX3fzLZ3WB0AAAAAAA66OVXtOSc+7KkLzdNu7Hh9aW93D4A
PBAwAAAAAAEHikeAAAAAAAEKOBA8AAAAAAEDikeABAAAAAAAIORI8AAAAAAAIUECBwAAAAAAAIORI8AAAAAAAIQ
wAAAAAAAHBwjHgAAAAAAGlDraYLHzK4ws8fNbMzMbuhS7i1m5sxsYy/jAQAAAAAAOBn1LMFjZlFjt0t6k6QLJL3dzC5o
dWb2KklNOf+sduKzOw9ZrbTzHaOj4+/+JECAAAAAACEWN9+ZNNMlpjulfSf5ivrnPuUc26jc27j6Oho74MDAAAAAAAIKv4
yUtb9jWtyT9jNu5/xhAwAAAAAAoKZrgkf+UzaS9BJJPylpe/B+k6R/7bagc65sZu+T9DVJUUmfc7tMrNtknY657Z3Wx4AA
4Xg2kAAAAAADos/m+olXzWUn/amZfCN5fLemO3oQEAAAAAACAA47GgBI9z7iNm9hVj/y6Y9KvOue/1LiWAAAAAAAAs1F
6AAAAAAA+mihCZ6i88dtd5JkZpnehQQAAAAAAIDjsdAEz11m9heSlpjZr0u6V9Jf9i4sAAAAAAALNRCR9H6EzO7TNKU
U9jvSTrPzB5umDUo6Z97GRgAAAAAAAWZr6vaP2dpK9I+qikGxqmTzvnJnoWFQAAAAAAABZsvgSPc849bWbvbZ5hZstl
3sTEgAAAAAAAI7HQodjv8TMvmxmK81sg6Rvyx9JcWAAAAAAAH0232/wSJkcc+8ws1+S9likrKR3OOcYJh0AAAAAAOAEsN
45j+m/mdmO3oUFAAAAAACAher6BI+Z/Wdjcs5NmndkvNs3+D/Ot3MyuMLPHzWzMzG7oUOZtZvZDM9tIzn+30MABAAA
n3B7nXFHsNZKuairz65Jud84dkSTn3MF51gkAAAAAAIAm8yV4XlFX7d43WyXp2Yb3zwXTGq2TtM7M/tnMvm1mbZ8KMrf
8yWWAAAAAAATildEzzOuegLWPc+SWc1vD8zmNboOUnfcc6VJD1lZk/IT/gwBDsAAAAAAMACzfcVrRfiQUlRzexcM4vL/8
5w73KuYAAAAAAATkbmXLh+0mbjxo1u586d/Q4DAAAAAABg0ZnZQ865jc3Te/kVLQAAAAAAACwCEjwAAAAAAAHr4I
AAAAAAACAFHggcAAAAAACDKSPAAAAAAACEHAkeAAAAAACAKCPBAwAAAAAAAEHIkeAAAAAAAEKOBA8AAAAAAEDike
PzB42szf3Mh4AAAAAAICTUC8SPGYWIXS7pDdJukDS283sgqZiH5R0l3PulZKukfRnvYoHAAAAAADgZNXlJ3gukjTmnNvjnc
vYgVAAAAAAAGtPr9I8tvl/RXzrkzJb1Z0l+bWUtMzrIPOec2Ouc2jo6OLnQAAAAAAAJ7Jejn2STqr4f2ZwbRG75J0lyQ55/5
+1Op5ckBEgl6RDsICgRGMdBIGw6kPC+Iq6g4jPo+4IggDOO77xGgo4zPlqj44oPjKDiBuoArwuIlOMasklYzIsjYsks/R6l7q1
QQK8PNPgbg74Qq

DscHPEREREREREHY4PelilililililOhwf8BARERERERERdTg+4CEililililil6nB8wENERERERERE1OG0vXVglcSNAM4BsE1Ku
oi9HPH+G0e7auPNGBpsz0Nv1tytuH1fYtNlcN2KtosFs7M4oi/Y/9XWx+7u32r7xnWH92SwabTUtG203c5iBYaqoGR7GMI
vr+nhmyzi2JfS4tCeNlKZie6ESW6boc21ZGreLrntKU+H7EiXHhecDjutjds7EhGXDUNW6OKNrvrOwq3wLeuuve9QmbNdD
Gdek40BQNO4sVzErrSOSqbm9H2fZRDjzkUxrSugrXl9g+aaEvZ0JVBfQhMFkYkTU1+NJHWtNQqLgwdRWO5wzY2jYXqig
aQ1lx0VaU6GpCkaKNvpJHbbjW1oKFQcmKoKU1eq5yk5HooVF4d0p5EzNWYdsDC/J4VtEza2TliYkzeR1gVKjsSOQgV9OR
U/6yBkathds9OZ1mlqKCctFoeKiL2cgpQftz/EkHN9D3tRhOT6Ktot8SoMAYGoqSo6LrK7BB1CsODA1DaMIG7OzUd3qcD
foowlCk5aDrrSOSctBSIPRldZhOS7GSi66MzoKFQeaoiBnakjpCsZKNjRVxbbw/JDA9klFOVODqSsoVBz0pE1YroexkoOsEbl
lwlAVbBopY063CQAYLzmYldEhAlhSwnYlCpaL3pwBy3GRMYLylB0fjutsuliravIGCryKRUTZQ/DkxX05wyYqojxy0Ha0DBZ
A1NXwn7rQUoF2yYrmJM3YWhBeW1PomQH9deVUjFZ9jBSstGTNeCH5SyGbdP1PahCxc6ijbypIZfSMGm5KNkuerMmxs
lzkTA2GLqArKiYrLgphP9WUcOwydOwsVJA7ayCtqxr2ehO67A9WS2PpgApXYWhKXA9iaLtYWFbDvqOCPq7hA9DVCmX2
AAHklJREFUqNiwo4hNI0WkjaDvd5k6RksOullaTE3F8ISFXHhNovG67HiYkzfH+BKTloPerImS7WGkaKM/b4bX3USXqWLzu
mcPScPF4eL2NnMbgnFSwPO4o2ukwNaUNF1lQwXgr6ydwuEwNdJl4Zq59zHdqdpx+2TmDzWBI9+aC/qlqCtKZCCKAYzk
UHE9vDJWRI/OhOP7GC06OGRWGsHdh1rlJ+CgMSfdxSDe31Gx9l5XVAUUb3OWVNDxfVwSHemOI9Imsc1zlvamT/GzZ
Trtd5/pdqCU/0ApX1SElHLvHFilkwEUANyc8lDnrQA+guABz4kAviylPHGq4y5fvlw++uij0x3uPjFWtnDX09vx6dufhuX4SOk
PF4QK+fM+LdeU+8+g5uOvZ4T2uj92tz1bbA6hbt6A3jY+cNoh/+vnTsTH/66+exTuWH46v/HZXma5YMYibH9y1OZI95bZXr
ljTvL1c8/fHgjTls6txpfXNkvP20Qv31uK95z0KjsHbdaxnnK4ADue3EbXhkt02h7sdf9vhe34cXhAn74yKbYepzfk8ZE2cFn/9+
Z6rHWH3OEL55/7pqXTV+vuqMJTBVBZ//1XPVY1x26mBTHlfMSuG7D75UV8cpXcE//e3R0BQFn6k559Urh/D1+9bBdmXT
1CouHVxXLNqGe7+0ys44Yh+fOGu56vLP3HWUejLGfiH256qOz4g8bX7/ox3n7AAX/rNC+jjGPhb17U1DauWbUMZdvFD
1dsWIQWUPFE5tGcdKRffh0TZ3Vtq0rT1+CtK7gn3/5XFMmbzv+8Lr2VntP5s9KI2OquOyWPyaeP2dqkFLWHfuKFYNY1jFb
PbcJFbwClwUb//Kr+ljnz0rh5gdfwqMbx5HSFXzqLutRdnzc8vDGpjiVpf+1KNteXfusreu49n/l6Uvw2MYdWHH0PHz9vnX
Ch6C7Sw41T4Qd83nz0rhi3e/UD3uFSsGkdFV3PjABrznAX4j/9+CYym8L9PWVypwW9aXz09CVN43Lj+BTfTUVVWHZjF
OFWyfbxsZ+sbTrfu09YgFse3oh3vuHwv41FuWwtTVuvbzybOXYnZOR7HiN/XFqI9+83fN7aa6Powt6f78x924pzXzsfm
59fgtOXTqvqf9GbTO6j3313hfryrigN40Pn7w4sWxrVi3Djx/ZWO0zHz55Mb55f3z/iuo/6ec1q5bha/e+WNema7db1J+F6w
oOLKubhrHnrj9kpbtydx4On436bTffabbgVL+A6UctYQQj0kplzct31sPeMKTlgRwZ8IDnm8BuE9K+YPw8/MATpFSbml1z
3Ou3F/DzJzbj+vvXNx3nR5echHdc/9Ae18fu1mer7QHUrbbv01MW44ffjMUd1EFc3X7t3XVvbfuH8Y/HxW4Pja9L5on0vOXk
+9fn1g3l5y8CADwLXvWNcVQGysAXL5icWJM0bW45ORF+Mo963DpqYtx9Nw8XmgoR7t11fg5Om6rY1xy8ilsHsjvVtwAm
Qlvm1ccvlieD5argeA5Qt6MFZ2m+q0sY20astTbXPjYtW0qLelnWWVOa4dtrYT5Yv6MFFNz7S8vxAcx9lur5R3bXbbq6/8H
GqfpT4/W6/sLjcUnCtV2+oCd23Xc+cAle+POOlunPu/eR2vEwqZ984fxj0ZXWYmNpHO+mah9j5Y3OndRu2hkjWo2RjXFee
U7v38ahvjrW1pPZCo+eZqv/X9tepxsSk/ZKW7cnceDp+N+m0332m24FS/gOIHLWSHvDM5Dt45gP4S83nl8NITYQQlwgH
WVMijlpvRDtb1u03eryVsezHL/6Vd6pyrK1ZvlUMUafyxW3rbKXbTfxWtbGuWU8uOatrnu0LulcOf5NMTTECiSfp/ZaRMcSA
TrfQmMlpzYttDYRlq15am28SWmrLOkMiffVttPRkvOIOPrc+EY0d11267GS06iccaK7V/fVuNC1Hbb4xld9t3q/VRW5yq/w
nfjVjKsxsjHOSu3G9rekco6F40OrMit106hOGmOINI7HSeVvdYzGc7Xzc9y6pDEoaZ4ZzX2S4mk1f0yaN7V7H5+qrSW15
trFvKnbfGx77++ZLvtLHO1KilfK9uOe05WCKhB7nHnd6VdVH7tbn622T1rXHPou7ZLqj58cdtmw3dXtDpfVNeKQF25ku
b7232wq/nr25+PH0LgypnQFPV98ZyzsnpiWRuXZQ2tqTxjX432b7VeEUBPRk9st43Lk9ryVNsoAlPWWVKZk2Kr7Sc9GX3
t3qv5Uu6ynRd+dnbCuP598H49ia/c+oghUt03aj2toibE0jndTXbek8taee0/HiNktxsjGODOGFtvfkso5KxwfovPsTj9tbENJ8
ffabbgVL+A6Uc7ZjJBzybARxW8/nQcNkBa8ncLNasXFY3sK9ZuQxL5man3HdhbxbXXXBc3b7XXXBc9UVk+8r+Eke74uK9/
/OWm8sVte9UZS6AoqNbLbY+93FRHUV1fsWlQrz20u65cSWV57SHd1bjjjvnZ85bhzc132+6YH1dfHFf3y0wZx0wPrsBA
s1Yfe5Q3TFWnzNUV1eNn686Ywl6M0bdMeLiOLI/21THKT144PKZhnNevTi4R9y1m50xgndYxNR97TUoWU7TdledsQQ/
9r4l61mzE707z+qjOWoC9r4Mj+LMZKFdz0wHqsPmeoqT5uemB9tZxx9bj6nKG663HbYy831dcVKwbRlZxwK0c2YU1Dnd
uXjnUFM9nzh3CzQ+sx5EDOXzy7KXN/bg/i5sfWF9d1ps1cOXpS2LjPHlg19Q+a+s6rv1fefoSfP+hDbh65VDyWfITwx1rN2N
WIKfPv4y7li7ua6e7li7OXZcbhyforjHynZsjH3Z4AXqje1hzcpl+M0zQdmSxp/V5w7F3kcajX14Wgcj+vDQ88dgqlAnu/ji28/
ThWqjRdj6tXbn2w9jxrGu4tq88Zwvcf2hDbf6M2FN2TGst4x9rNLcu2ZtWyuj6z+pzK/hXVf9Lpa1Yta2rTtdvpqsAX397+PD
r5D5RytGMm38HztwAuW66XLH9FSnnCVMfs5HfwaNOTRWvbZPDG+5nOojXTcbSrPouWCsfzMftVZNEq2S4Oj8mitaf1sl
OohVknDI1BdLHrixano+ejlFCxYE+RRyt2/WbrntzFi0HKV0jslFINOyctFG0PRQrHgbyBoQQ2DphoTfKojVhoS9rlmMmZ9H
aQ2W7WN7oYKsqcHUBlrT9Vm0guxNbWTR6krBcn0UoixatoeUrkBTfLyWHORSWlh3QRYtI8yiVbAcDKcNlJ2g7uZ2m8ibC
md1mGqQRsvlyGLA1BWUwyxaru8H2VQcHyXbQ84MMsVEWbQyElB/RduBqdZk0QozM7l+kCkrMYuWoqJgN2fR2jZRQ
u0EWTgirJoFcMsWj1RFq2ciYrnY7zsoKuaRSvlpMzocQjOjwnLRdpQYWgCWT3IVqOEwbRGijYG8iZMTcGm0XLw7VijarYgl
WLUODhAc/zKIVIG9XFq1yxUNPVg+yaFn1WbQcV6JQCdqmjz0olsj0lzM15E0NkxUXJdsLsphZNtKaGl4fH74PjJScbXVNWf
u4lsWj0ZAxIDxXjJrleYRSsqj6oAaU2FoddK0QrbgyKC/i7hQ1dUlj0wi1YqaD+aqiBvaCi7Hkq2i5wZZNGanTUhwixajTvlQpL
+ZxC044jhqQd8phVm0fN9HSg/6eV0WrbSOcas+i1b0j3hzu4J+GlwjQT0cPacLL4+XVMIM7kkFK6jfnKkh0zKL1q45V10W
CbNeBKH2NFF/O6TSwnj7Vt0Kj/ro0sWmHmsHltZNFqnLfsThat2nkTgKa538aREjaNFJGpyaJVO7+r3ac/t2v+N7cryKK1vd
/GaZj/yqAsxGkSf+AlHLKJzed/oCHilililililGhPJT3g0el2ng5SyndNsV4CuHRvnZ+lililililil6Gaxk+/glSilililililKiacAHPERERERERE
wo57wHOGEEI8GvdSJKKDGfsFUTP2C6Jm7BdEzdgviJodbP2Cf6JFRERERERERERTh+lCHilililililKjD8QHPZLl+pgMg2g+xxA
kl8L4RYJ4T45EzHQ7Q3CCFeEkl8JYR4QgjaLhsthDibiHEi+G/PeFylYT4StgnnhRCvL7mOO8L39RCPG+muXHh8dfF+4r9n
JCGOFyM/y8Lly/cN+UmGhqQojDhBD3CiH+JIR4RghxRbic9ww6aLXoF7xntMAHPPuQEIEF8DUAbwHwGgDvEkK8Zmajlt
/SDoH0f7gO4gfs78U3jOok1L+AgDCOdI7AQYF+3xdCKFOMZ/61/BYiwGMAvhguPyDAEbD5V8KtyPaX7gAPialfA2AkwBc
UspRAHcDODtc1yWlFegGb4m/ueZYRPsNkeX9AEYaFu+LfpB0DqIzI9AvkqwC8EMpZUVKuQHAOGrzqdj5VPiNhNMA3Br
MvmSCm3hD9vBTAn/DmpX7Ra/nLMcqJOsc/6QdI5iPZnl4V/anJjZtCodrd9AIYk1K6DcvrjhWuHw+3J9qvhH8K8joAfwDv
TIAbgNwEellBO163jPolNVTl/gPaMFPuDtZtYDOKzm86HhMqlDipRyc/jvNgA/Q/DVyoHwK8II/90Wbp7UL1otPzRmOVER
zWuMDnn3rEQCD4du6DQqvgbp9hmMimlZCiKwQlh/9DOBMAE8jaOtRNof3AfjP8OfbAVwUZOQ4CcB4+FXhXwM4Uwjf


```
fnE+gN+G2xPNuHAcvwHAS1LK62pW8Z5BB62kfsF7RmuiQ+I8Ylggjdu/AVAB3Cil/NwMh0Q0rYQQixB8awcANAC3Sck/J4
AMWjCCzyc+xl/tBUl/b6wUmakNCvzgFwVftJYXAHwoeielEOL/ALgYQTaVj0opfxkuj51PhfegHwKYDeCPAN4rpawllVIAvov
ERERERdTg+4CEililili6nB8wENERERERERE1OH4glelilililqMPxAQ8RERERERERUYfjAx4iilil64AkhPCHEE0KitUKIx4UQb2
yQoh7wm/1PCWEWNW4cdI2QoiFQohnhRDfFkI8I4S4SwiRDtctFkL8puYbQ0eGyz8hhHhECPGkEOLqfVhmliliOsAJKeVMz
lcQfAPyLlPjNqogUgv+p9iYA5wP4EAAB4HYA10op79+XdUFEREQHjv6JfHERER0Mav9E668A3CyEWlbgQcs/CyFOBuADm
4bfxOkH8Nbw3+OllI4Q4iUE3/Kp9Z4W21RqtxMApFucWgD4vjTyW6++FERERET1+A4eliliOqglIZYCUAHsBNANYFv44OZU
Di+wDuEEI8BeBRAM/F7NvONo0uBPAtlcQaAA6At0sp7xJCHA3gQSEEABQAvBfAtldTMCililiKAL1kmlililililup4/BMtlililililq
ERETU4fiAh4ililililiow/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": "m3l6pfwStjft",
    "outputId": "d3d58045-af41-400a-961d-02e6c3b73163"
  },
  "outputs": [
    {
      "data": {
        "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAEIEQVR4nOzdeZwcZ33v++vlp1ZrTMSHYElmxLNiCFVRBykrDZZguSycE3MSEEEs6
f9evVruqufrnqqnqWqf9PdP3POCQAAAAAAAAOmVWe4KAAAAAAAA4CdDgAcAAAAAACDIPAAAAAAAAACkHAEeAAAAAA
g5XxXCwAAAAAAIOUI8AAAAAAAKQcAR4AAAAAAICUI8ADAAAAAACQcgR4AAAAAAAUo4ADwAAAAAAQMoR4AEAAA
yuWWuwlAcLZ797vfrb179y53NQA8TPbv3y9J2rBhwzLXBMDDYfPmzXrta1+73NUAgJ8YAR4AWGJ79+7Vt759l4LymuWu
DkCPAAAAAAACIHgAcAAAAAACDIPAAAAAAAAACkHAEeAAAAAAACIPAAwAAAAAAkHIEeAAAAAAAFKOA8AAAAAA
4ADwAAAAAAQMoR4AEAAAAAAEg5AjwAAAAAAApR4AHAAAAAAAg5QjwAAAAAAApBwBHgAAAAAAgJQjwAMAAAAA
AAwAAAAAAkHIEeAAAAAAAFKOA8yPe73613v/vdy10NAAAAABWjhXwPjy33BVYifbu3bvcVQAAAAAYMVYCe/D+C
eAAAAAAAFKOA8AAAAAAEDKEeABAAAAABIOQI8AAAAAAAKUeABwAAAAAAIOUI8AAAAAAAKQcAR4AAAAAAICU
gAAAAAAgJQjwAMAAAAAAJByBHgAAAAAAABsjgAPAAAAABAYhHgAQAAAAASDkCPAAAAAAACIHgAcAAAAAACDIP
AAFTKNae3dO8N1ObAC7N27V6VSabmrSaRS8QkeM3ulme0xsZ2HDh1a7uoAAAAAACcUvLxCR7n3PskvU+Stm/fnvp/g
4AAAAAAAUo4ADwAAAAAAQMoR4AEAAAAAAEg5AjwAAAAAAApR4AHAAAAAAAg5QjwAAAAAAApBwBHgAAAAAAg
AAAAAACIPAAwAAAAAAkHIEeAAAAAAAFKOA8AAAAAAEDKEeABAAAAABIOQI8AAAAAAAKUeABwAAAAAAIOUI8
```

AAAAAG5QjwAAAAAaPfxuuSuwEm3evHm5qwAAAAAAwIqxEt6HE+BZBq997WuXuwoAAAAAAKwYK+F9OF/RAgAA
DKEeABAAAAABIOQI8AAAAAAAKUeABwAAAAAAIOUI8AAAAAAAKQcAR4AAAAAAACUI8ADAAAAAACQcgR4AAAAAA
yBHgAAAAAAABsJgAPAAAAABAYhHgAQAAAAASDkCPAAAAAAACIHgAcAAAAAACDIPAAAAAAACkHAEEAAAAAAC
SdqwgTeFQPqt5zwN4KxBgAcAlthrX/va5a4CAAAAGLMcv8EDAAAAAACQcgR4AAAAAAAUo4ADwAAAAAAQMoR4AEAA
ABAYhHgAQAAAAASDkCPAAAAAAACIHgAcAAAAAACDIPAAAAAAACkNdnlnrsOp8TMDkn64XLX42EwKenwclCpww
vloc1XlrO9vfmKfGAAAAAAQMoR4AEAAAAAAEG5AjzL533LXQGcVrT3ykj7rzy0+cpCe68stPfKQnuvPLT5ynjWtze/wQMAA
vZt+Kb89PvOYP4ra+x8yek1g+sB+Y2flm9rV4+cfMzDu9e4kkM/uBmd0Zt+ueeNkaM/usmX0//rs6Xm5m9hdx291hZk9M
J81sVbx8k5nVE+P8vYnXnFK7juo7WFpD2nzJ53AzK8SP98bPbzo9e7yyDWnvjyXa+gdm9q14OWM85Wz4ezHO40nOOW6
V7JN0k/kDTZt+xtkt4Y33+jpD+N7z9f0qclmaSnSvpavHyNpH3x39Xx/dXxc1+Py1r82uct9z5z67ZzVtIBSRsZ42fPTdLTJD1R
/5/1fSm+P7jPGU3zT8vRjn8cSNT/CcXk+RtNc5t88515LOUUmXL3OdsEjOuQecc9+I789IukvShhEuvzSR51zTefcfZL2KuoD
0tviwaO8zySWS7nXO/XBEGcZ4yjjnvizpaN/i0zGmh20DS2hQezvn/s05144f3iLpEaPW8RDbdVjfwRlbMsaHeTjn8GRf+ISk
1t7Dlq+VdDxx4Un/WH5O0r+Z2W1m9sp42Xrn3APx/QOS1sf3T7W9N8T3+5fjzHClei8KGeNnr9MxpodtA8vrNxX9h7bjfD
mtJF0p6vKQHFH0cFGeHn3fOPVHS8yS92syelnwju67ZakZlkz8mw07JX08XsQYXyFOx5hm3jgzmnmbJLUlFSRe9ICk85xz
wLnHOhpPcr+mivNLy9hy0/ouijg7m+5Vgmzrn98d+Dkj6pqG0f7HwMN/57MC5+qu29X71fDaC9zxzPk/QN59yDEmN8B
xHu9BgOf0ulXSFot+gd9T9BWA3ctcJyxS/F3eD0i6yzn3zsTy5Hdud0IS55f8d0u60qLMCudL2qLoh7sG9oP4lvOLkq6IX/8yS
Xxftfabo+a8fy/ysdzrG9LBt4DQzs+dK+j1JO51ztcTyKTPLxvcvUDSe9z3Edh3Wd7AMTtMcnuwLV0j6Qid4iGVxqaS7nXPdr
9+jRlakyf1AUcaGryv6ob+PSyos936v1FvcFrFht+902knRd+o/L+n7k4naU283CS9J27TOyVtT6zrN+M23SvpNXLlyu60Lx
Fpx0Nt11F9h9uytPmSz+GSivHjvfHzFyz3sVgtJ0HtHS//kKRX9ZVljKf8puHvxTiPj26dCgMAAAAAACCI+loWAAAAABAYhHg
ZpQ9DhVaZ2W8/hNdda2a/G99/qpl9Ld6Hu8zs2gVe+wwzu+khVhKAAKxwueWuAAAAAQJ+6c+7xiyz7Qkk3SfquJDNn3vw
f0rmQjP7kjlDEzf/gZn9SfzchjN7opl9xszuNbNXxWWqZvZ5M/uGmd1pZpfHm/ufki6MX/v2uOwbzOzWePvXJr1jP7npl9
ABwpimZ2bcSj/9EOuck/ZKkRznrrnJmtcs4dN7Pdkm5yzn1Cksysf10/cs493sz+TNKHJP2cpKKkb0t6r6SGpF9yzk2b2aSkW+
WyP5b0or56jyrzREmPdc4dNbOnS3q9pH8yswlj/0XSyxZ5nAEAwFmEAA8AADjTzPuKlpnIFAVjPhD/Ts1if6tmd/z3TklV59y
z2yLjScoP2N6oMp91zh2Nt/HvZvZXZjalKAB0A1/bAgBgZelrWgAA4lwXBy2elukTkl6g6BMxi9GM/4aj+53HOukvkTQl6Ul
1QkAAjydCPAAAAAznpIvJU045z6l6CtJj4ufmlH0KZqHakLSQeeCb2bPlLRxyHo/I+k343rlzDaY2TjPjX5b0QjMrmdmYomBM
NxHjb3zf4n6YqHsK2PSLRzO6UtEfRb9/IOXFzP6PmX1b0qedc28ws0dL+mocs5mV9GvOuW+Y2cck3S7poKRbE+tt+qa
4PAABYHnxFcWAAIKXiDfT3SXo3wR0AAFY2PsEDAAAAAACQcnyCBwAAAAAAIOUI8AAAAAAAKQcAR4AAAAAAACUI8AD
c1QAAAAAADjtbrvtsPOuan+5akL8GzatEI79uxZ7moAAAAAAACcdmb2w0HL+YoWAAAAABAYhHgAQAAAAAASLklDf
KelPjF3KUtXpTFCv+7r74lxmmm0FLITFy+toraU1ZU91v61SPqeZpq/xQl5H4uXOOR2r+1pXLajuB6o1A42VcprptjVryqvRD
+damhorqN5qK5PjyMuZ8pmMjtZamijmdWSupYqX06pyTqFzenC6qVl+Jy9nqhRyqvuB6q2gu5/tuF6zrbYafqipqqaH2i6
XKq9akJlGtFb2+5OWUz5qqhZxmGm0dmWvpEauKCp10cKapsUJOJS+rgzNNral4ylgouYwOz7U0UcorclEK2ZyO13yNI3L
bq8t5+WGoE/W21o0XFIZOx2q+Jkp5nai1NF7Kq+a3lbWM1IQ81VqBjy2NFbMycubMpKcMx2O69gM2vlyOR2Za2pqrKC
FbRVyOV0cKapn5qI2ungTFPrxgrKZU0/Pt7QT00U1Q6jvrNuvKQDBcooq2P1lqpeToV8RtMNx5OVgmp+oDAMVSnkNdc
w9CNduBKl5ex2otlb2cVpXyqrcDtYNQQRgdx8IKQU5OdT9UvRX116yZLCMVslk559Roh/LbTnW/rXVjBTXboY7XfZW9nM
7PtjRRzqmUy+qB6YY2rCqp7gfdtsyYVPdDzTSifiZjtdbWlvxNNds7yYV82PxmY1kFO5kFFGptlmoLIWw6vLnJlmtdqhWk
W1UJOftHwGGbUbEf9+icr/FiThmzaM6teMpnTMMNv7svc81A46Wcaq1o//wg1l4fmr40TzY8NuqeHkdr/laXcmrHTodr
jRWjdptr+fjyURtPIPIqxcchdKGqXqdsXtONaM6ot3xlm1IVC1INN3xVC3kdm2qll+OWdakmZavQjanw7MtVys5VQoZhY
tHUWNQmD0w3NVn11A4DOZdRqx1orBjPUa1A504UFITSiYavipfT4dmo/9RavspeXsdqvtaxPfPYiYYesbqodigdmmlqohy
NzTY2XovYte/Ionijk1GyHmmsGWIPx5OWcXJhRzY/Ow525/3jd19qyp0DReWm6EZ3njtV8FPRXODlouXtUKrH4zBQolyL+
2lcs45bM5HZyO+veqclYn6oEOzUTjPgids752jRZUqMvaqYZak4Z9YFmOzp3tMNQtVZnHmxrVTmn8WJOJ+qBDSbr8YNA
PxOlre7UAj0Yr6fsZfTj4w2tqXhqtUPNxNdn2Yxpthlt/8hcdG4eK0Rz4WyzHZ3DW22NFaLrFy+bVTE+71W8nCqFnMlWVDO
7U0Xsopn4muY9aNRddnE0VPc62oDc+dKMopPnfHz5fzOc21To5li6+f6n6o2fj82A5DHZvzNVn1uvNVp3zZy+pEw1fDj87
PNaNy3o7Hdud4aL0Xz/WTvUxCGOjLnq1rlqeplDXC2FV1n50yryjnnNkKdqEfn3NlmfO5oh91+n89Kc61orK8q55U1qdkC
pslJQ04/6/9RYQSfqlax8wqd01zcD4/OtVQp5DRRzGsmcW1XLWY0U4/mu/PWIOQHoQ5MR+ezaiGruh/o6Jyv9eOenDPV
q+5HfWeqWIA+azow09SacrRvZS8qN9vyVY7fc5W9nKbrbZW9bLcfrkrUc91YQYE7OR8dq0X1nG74KnnZ7nVU8jzUOfd1r
n1rtnWs5mt1xVPOJD908gOnllyO4+HZpqaqBxk5yQ+i89JYMa92GL1POxrvX+Ccaq3ovel0va1SPqOxYk4mqd4O5cfz0uS
rn5WUzOjzb0qpyXtN1X+VCVqV8Vs12qOm6r2Jnzshm5JxTEErTjZaK+ehaqlzIKmumkpFvXVnJkhZT95PECzLn3NKs2OxnJ
iPqNrXrBVN3zjR7rsMefqzz73ve7yqy7Zok/f+YCe99Pn6l2fP1n+dc/aoo/t+ZFe88wt+ux3f6xnXnyuvnjPA3rmo87VdTd+p1
kn77GzT71nXVJVu0YXVJx+da+sD/uU+/sv08feHuA3rjz2zSgemG3vX572t12dOv/+zGnv256plt+vBXf6hjtZZef+lf+vuv/1C/+
lc/Y4vevPvb3X191dM267qbvrNg/Ye1WX+ZznYGHcP+csdqLf3h8x6luh/29I9O+7z6GZv1ni/t1Q+P1LvrOGeiqD//3Pe6yW
sVOOpL7t2btOeHxzSEzZO6robBx+ra3Zs1b99+wE95YK187b1V337MFHKqeTi9Ok79+u/PvGROIzr92zv2h1b1Q6jNI5d9
XXaRV8T4cm2vNK/+G51ys/+8/7tOxWqvbX9//H/vmzw9xv++sNzkuz50oystmdPXHb+/W7zXP3KL//s/f7ulTayt5Odm8vvb
17y6DZuDrr7slq0fL+idn/3eyG1ffdlFWlXM6UjNHzi3Xvkn87pjtH9fBo3V/rkydKZrb5w/rjesKvTMwYP24ZodW3XDbfP7ZW
DON9sBj+ZKf2aibbv9xd05MjtVk+bdcvk2fuUph3fnm43t+pB2P3aCa31vurS/cOvRclTxvXLSuqpc8dePI+fWnVhX1L3fs1+M
uFDauKaviB3rz7uz3n+7/60uC++1Orivrbr/5Ae354lj6u21RrtvG8dWXXaQfH2/Mm8OT/SJ53t99+3696Inn6bqbkieFrfrx
2lOkc69c+a4vyWdPv33Bnz+u/f+C4Lj53dU+/T46ZL9x9QL+8/byede7auVX/sOdH3eO2a+dW3fXj41o3Ue7Zl/5zdv/8NWh
2rlVkuV2u2l+o7dd8Vgdmnmq7Z+5Z+Br+vtQcrw3g7BnTHT69oe/+oN5c+JbLt+meqvdc5147Y6t+usB89apHJPO/H+s1tfl
WBOL/6ZjT19+JodW5U1p0/feUA7n7BBh2aa847VWDGnQi56Y94/50tO//2fe8fEF+4+MHTuGnbufsNzLtaaiqc/+Mc7e455
Pddz195LqdW/X5ux7QpY8+V+OlnFqB617Tbt84oV958sbuXLZxbUm/c+IFOnCiMfCaeNfObXrPI77f7Y+d9zj9vmqS7Z04
zev9/7Js3hq6+7CIVcxmtqnh69sXrUxvkMbPbnHPb5y1fwgDPFZKe65z7b/Hjl0r6GefcaxJlvh2XuT9+fG9c5vCw9aY5wPP1+
Ewau630vfZK62397ojtvOOKx/U8/+pnbh64H6982gWSpCCUPcvfXrbFY/T3oMz3XoMe90rfv4CveeLe3u2+cqnXaC/+P

UufpJd98NZ5x+7R54wN7C8ffPmT9ZsfunXBdfb31f76JfvSkzau1syQ/pmsUzajkWNsUH8Y1lbvuOJxuuvATHedyb6Qjb/gC
LvyNDJm2i9i5mDksfxVI9JckxJGjnHDJq3Rq334nPGEvrbsLotNO+/44rHzWvzUXXsPN/fnxdzLab1u2T9kmNv2LYXWz453/9
2HTRXb143NrBunXbtPO6M+2F98a4DM4sa6+976ZP06/F6r//NJw891sOuJz70GO/Ry//X1xfVb5PPv+2Kx/Uct0Hr6e8To64
0l+/ha4TF3o+eUx+kn1Ots+weWWxY/yDL3+yDs+2hl4vjZVyuU2Hx4a230XrBl8z9ved5H4P6+PD9meha5PksmHXWqey7v
qud9YfS0DEkSb+weVJPPn+t0mhYgCcV4Soze6WkV0rSeeedt8y1eegenG4qdFLDD2Wmno4mRY/rzfbg5a3ByzvrOV7ze/4
056Bj072f/suTjuUW2y/Ga37POTtm5lft4eKa58Drn5vfYfsgScfmFDWH9M9knRYaY4P6w7DxNddq96yzo7OsU26hfZhrtl
q54PkfNy/jlF17PwDNo/0z+/9+9e/ruS4GbXeUy2fnG/6zx0dC7Vbp88sZh4OnRacjYzBobuz4h1n+q+JY/LsDol+0Wn3Khl
vQutL9mXRpVdql0Xer5/Plrsa/r7UL01eH+7fXVEvx+2nYd6TE51//vrt9D5YrHn+Z90nzv3R83vix3jR2abqjCHX4fNtdpqh25f
l86jgutOznHLbYoi51Xu+fDEX1Sit6fn22W8keW90t6ZOLxl+JlA8vEX9GakHSkf0XOufc557Y757ZPTU0tUXWX3vrxgrlWRQ
Df/rqMWRZQPYbt60KvG3ac+/dxMevq3O8/hsnnk+WS+9m/LPm4Ulxcu6wq53vWmXz9oPJTY4WF11mZ31eH7UPGpDUj
qt9C2F5rzhs01/WX7XzdqvcPmhv7Ho+aQTr8atl6F9mfYcR42vw/rd/3jZqH5abHlk/PNOhOHNHZu6T9vLGyEzpgWnl8W0
+4/boPUM279h20rW07nB5+j+9Y6cl/r60qn0w2QbL+bc1T8fLeY1/X2o7A3e384xGdXvh23noR6TU93/QfVbzLhZ6Pmfjd8
dB2FzNXJB8vZp+HHft55frmyEHR7p/jFIOHxc6rnfPhqDGUsej9+dkms3CRh+xWSVvM7Hwz8yRdKWl3X5ndkl4W379C0hc
+mj9xyn655wdbo746tPeXWVjxdT/Pksutv3tfzujlPvht6fK6pht+e9/rOdq66ZltqLV+/++yLu8/fePv+eeu66plt2ryuqsmK193
EG4r56Du6F05VFqzHjbfv166d23r29ZoXbF1U/Ye1WX+ZznYGHcP+cp227O8fnfbZtTM6Lsl1nD9Z6Vl2zQt6y2RMOn9ydL
8Q233a+1FW/oGLtu51ZtXlcdOO7mjccdw1Vr+rrpjv3zjuvNt+/XBVOV7rZG7cPVI12kWsvX5nXVgeXf8JyLu+3b6a8D54cdv
107t+qc8cLaug2bg66+7CjDOFVZcNtXX3aRag1/6NyaHKP9+zJorPbMIU1f1+4YPK775+BB+3DNjsH9srONa3dsndfm1wyp
33f/neBefXC6cq+sgt9w08JyfpNcPmnc6yyWphYDsOmus6dZ2sFqLfAkg8l8/YvH1L1rlzXAZdK2yeqqjW8rvLO31tWN+9cK
a3WV/d8t988p0jvVbX7ite+5Pvw5z35k/XybHzPU375u3zl07t/Yct107t+ozd+6fty+Dzh3D+m1y/lhb9nTTHfuVz0ibJucf/856
2+Zdj147ZN46lWOSrOuWoiXnkf76DTt3duo1bN7vuWYbch4YdW08rH61pj9vH5Lt/vPfdS8sZFc366d2/RP3/iR3v/le+f14
OPXfx1w/c375p3/+9+7DFvP5qlqdxwcrzXn9Zhr4r10186tymXVc017/c37euayG2/f350/Bu3rrp3bevpij/7w9aj+vumSLMqa
rbLNLv8EiSmT1f0p9Lyr6oHPuf5jZLkl7nHO7zawo6W8lPUHSUUIXOuf2jVpnmn+DR+rNohWGTuVC9AvrqOtePBByUTaf
Ztl7VWhpPZNGaKOXknNODM1E2gf4sWqFzKnu5nixaTT/UZNVTPc6iFWXQCTTTCLSmnNfxRl8WrXag2UagSiH6hftOlo/1
rOIsWpWT2abGi3EWzrMg1o8V5RRn0Sp7ChVl0Zoa8xSG0gNxoheLFrIKPvBTD263+5k0RorKHTJLfp+NxtLxizOWBPo8
al1XITgRmTRipd3smhFWaWibClzDV9rk1m0vHw3u8xUmN3Gy2XlnEVZtMYKkkVZtDq/zH94Ls6i5eUkG5JFqx1uW4WrX
vloXKA67VVjZj3SxadT863ofijD6yUFmLMuRE2QTilFpeTo12GGVMibPSHZhu6KcmSqq3T2b/yWaij7DOxH1SMP2ot+IsQ
vo6qXk+8ChaGp2T6Z9WqslFNGFs25lSjrw0wii1Znbq3H2UT8MNR0pDbKncrsY5mMmn6gihdl0lsVZx85NiCLVjZjasVZt17
eLalYj46ZtlMdKw7mbkqcbu6UDoyFx2z2TjTtyeLvr0VIZ9ttDVVjea4Wiv6CPfqODtk2cspiDMLrh8rqOa3NV0P4jnlZBatyao
VwzyqLSzaLVjrKHHJ5t6tzxaK5KZtGaKhlq+WH3HDAoi5bM6dhsW2uq0RjqZCtp+oFKXk4Nv61yladWO1StGWWompdFK
63tuLFWfeyasfbLObj/p/IonVoJpo//CCQl8sqk8iitbqcU60VZS6aqhZU86OsNhOlrfpt131u3fjj+b/iSQ0/6ufjA7NoOQWbdd
UsZtFK6fpuq+JctT/V5cGZdEkTG7MU8MPuxm8jft8+FMXcRatYj7e55zy2czgLFqzTZ0zXpQSWbT8lKpDj4vWZNVtJuPknHW
iiezaHWuYzpZTTtZzprtQOPFFjTBtpNFqx7VOQhDHZ1rq1LlqupldWi2pWlni1Ypp9lmqBONlsrxuXsqPt8dmm1q/ViU0Wm
aejc74qhezJLFrXHDQ2KouWl1W9HwxRwJcWZQzsZOire1mV81n5YaiGH83lc3Fm26mKp6lXZ9HqZBWO26taiN5fHK/5a
R9FXPZ7nXU2k4WrXy+e+471t1mLEUreh/Wn0Uruo6vejIVSxkdrwWaaafaXfRkmSiLVud6j9TgLFoPTjd1TiKLVnSN4imXSW
ZtDiZtcLeLFpBGL1njPpAIOGzFYTxV8MCTZRz8jZHa61tKoUZQntZNqLsmhFQaWSl5WYxcgpyql10/BViM8l5UJW2YypIMv
ACDICPAAAAAAACkHAEeAAAAAAAClCPAAwAAAAAAkHIEeAAAAAAAFKOA8AAAAAAEDKEeABAAAAAABIOQI8AAAA
Eg5AjwAAAAAAApR4AHAAAAAAAg5QjwAAAAAAApBwBHgAAAAAAgJrb0gCPmT3XzO4xs71m9sYBz19tZt81szvM7PM
dakj4q6fJkAefcF51ztzfJLZiesYT1AQAAAAAAOCstZYBng6T/TDy+P142zCskfXrQE2b2SjPbY2Z7Dh069DBWEQAAAAAIP3
sDwAAAAAAwFlpKT/Bc6ukLWZ2vpl5kq6UtDtZwMyelOlVJO10zh1cwroAAAAAAACctZYswoOca0t6jaTPSLpL0j84575jZrv
HjoCPAAAAAAAClHgAcAAAAAACDICPAAAAAAACkHAEeAAAAAAAClCPAAwAAAAAAkHIEeAAAAAAAFKOA8AAAAAA
o4ADwAAAAAAQMoR4AEAAAAAAEg5AjwAAAAAAApR4AHAAAAAAAg5QjwAAAAAAApBwBHgAAAAAAgJqjwAAAAAA
AAnM2W8hM8T5G01zm3zznXkvRRSZcPKPcWSX8qqbGEdQEAAAAAADhrLWWAZ4Ok/0w8vj9e1mVmT5T0SOfcv4xakZ
yjjFJ2yR9ycx+I0mpknbzQ8sAAAAAAACnJjfQSTO7etTzzrl3jn6VklbzOx8RYGdKyX9auK1JyRNjrb1JUUm/65zbs3C1AQAAAA
Cxcy+7KkX3TOzcSPxyT9i3PuaUtcv3m2b9/u9uzhQz4AAAAAGDlMbPbnHPzft5msb/Bs15SK/G4FS8DAAAAAADAMlvoK
Pvj7FgAAAAAABYZosK8jJZNZj+X9lfxlvkyv5uqSoFAAAAAACaxVvsj3h+SdJOSXOS5jZsf7/9u49TPk6wvf/69NV1de57c7O
WQFdj7F7x6uruq+3v+qKrZ6uvMorPD7L6fj0c/tqe6Lt/63vu73f05EEJdRERERERERESW0WlXeBoujqfuAMyssnRJEhERERE
ZIEUt8jjZe5xzfWdC0GObilililililgso8V+Reu8Htsu+HkmREREREREREREHP2Bn+Axs98FXgOcbGZ3ZF4aBv5rKRMmlilililil
bzGf4LkQuBVwgGVec8DJ55QuERERERERERERFzPIELPM65C5N/Tzo0yERERERERERERkyO1qChaZvaqjr89M7tiaZikilililililH
NivaG0ALgM+B9wPvNTMykuZMBERERERERERWZzFfkXrC8Cfoof+j/BM4EfALUuWkHERERERERERERwBfFfUULONM5Nw
AAIABJREFUi024ilililililjEFvqK1oszz/+w47XzBx1oZh5wNXABcBrwEjM7rWofDcl5f9k5twn4vcUkWKEREREREREREDlhogcf
m27I2AhvN7L/M7GYz6/mpID071My2mtnWXbt2LXBZEREREREREZEjy0l/snyGmU0Rf1qnlDwn+bv4c7r+BuBs4FjgRjN7c
REREREQWaaGvaP0sbgE2mNljZpYn/sHm6zr2+VfiT+9gZmulv7K1fQnTJClililililij2FmyBR7nXAS8DvgycBfwaefc983skjPb
25fyq9oiYililililililaAFHhERERERERGRFU4LPCilililililK5wWeEREREREREREVjgt8ililililililrHBa4BERERERERERERWeG0wCMilili
ucFnhERERERERERFY4LfClilililililKxwWuAREREREREREVnhtMAjilililililrLCaYFHRERERERERERGSF0wKPIilililililMgKpwUeERER
e0uP1483sq2b2HTO7w8yet5TpERERERERERE5HC3ZAO+ZecDVwAXAacBLzOy0jt3eCnzaOfdk4MXA+5cqPSilililililil6ul/A
ncjMLjWzrWa2ddeuXUuRVhERERERERGRFWu5f2T5JcBHnHPHAS8D/tHMutLknPuAc26zc27zxMTEIU+kililililililMhj2Vlu8C
+9giYilililililgchCVb4HHORcDrgC8DdxFH/y/q+mV1ZluS3X4f+B0zux34BPBy55xbqjSjililililililByO/KU8uXPui8Q/npzd9rbM8
ililililililrHBa4BERERERERERERWeG0wCMilililililissJpgUdEREREREREZIXTAo+ililililililYaqnBR4ERERERERERERKRVOCzwililililililucFnl
ERERERGSF0wKPIililililililMgKpwUeEREREREREZEVTgs8ililililililrNBZ4ERERERERERERWOC3wililililililiscFrgERERERERERERFZ4bT
rSpRDZvsnmmdwrhAvRlhLsdw0We2HhJ4HntmGxwzWqQWtXhkqs6a4TjyRZ/9tYjjashoySfveTwyXWPtclFKwWOu0WLn

c9g53Sc9vGhgEemGtTCiFLeZ3+1wWg5z3QtZLgYMNeIKOd9aIFEOQiYC5vMNIiLGSgGBlyNstmhEjrkwyu1wAQMcUi8cu6
DT/TVOGi8TNltM1yOqjWaSpzkmq012z9RZM1wgakb4OY+5sEkl7zMXRgSeRyNqMIQI2FdtMFT0KfoeM/WlaiNiVtNpVC2
va6beZLQUMFMPKeV9wmaTsu8z24gojOcZKQZM1UKGiz6lwGP3bj3hQsB0LWS0FAC0ZsvYNVPnqOECZo65qEnFD5hpl
TedotojzteBTDHI0W8yncXluZLySp9ly7Eqz82wjYqWYMFz02DMb0ogc1TBivJlnZ8b+WkgIqUNrhgpEzSa5XI7Rks/UXFw2lb
jp31EiBwDMenKyxbrSAZ7mk/kUMJ/Vocq7BcNEHZ+xM2oMBE2bjPiDwcxv0ssbakQJhM6Lg+zgHu2fuijhS8KmGlX7Om6
PwjGqjydRcXBfLQ4wds7UWZX0Abtm6gwVfAqBxXlQCynnffZV4/o5F0aMFvPUori9NVtNCp7PTCNUq2uG8hT8HGyW12
qSXv1c8zWm8zUI0ZKPmsqAdP1Fg/trzExlGeoEPdFkWsxnA/YNVPn6JeiUavF/rkD7XFiKE/UcuyeaVDOe5TzHn7OMDOma
xr7ZOqPIpJPVKjFSUsZdCqBz0w9opT3qDbidISRY9dMnKfFIK5jY+WASsGL20U9YqjgM9toMl2LWDOUp9qlqBR8Ws6xrxp
pmuc/RlgYKfY/9cXP7lFxfAy/H3tm4/eAcuVzcltcOF+brWLXRZKTKy9Yin/OZDZvM1iPKgUcp71EpeDTCjmHL2D1zoP1ErSk
uOapiOAz4O5vtZ5+L+tJz3ma6FrCnaTSblAKfXcn5h4s+YbPFXKPJaCkeywMvRy2KKAVxXzFU9Gm6OO375klmKgX8HEwr
oBjImanHfFrWa+J7HI1N11o4UGC7EedqIXNw3F3z8XNw3jxR96qHj4am4P62H8bxgYrjATD1kqBC3pX2zTaZqletGi8zUu8
9YjRpj575hF4UA9h53R8n81WE8gxk+w3WQ2T/siYrXj/nE4lpg10w8JlCkdIPzcVtOe7X4aGp+PVS3qNab7JrpsFY2afke+y
Gi5NJ0FHE3ynk8jcuyeqbO6kifvGTmLz7G6UqAWNRRkredQjqlUtqkmdWpXMC6qNuExmG02qjYixcn5+HI+aTbycNz+v3D
lmDmP3bj1y3me2HjFSiufNpbzPdD1kuBCwdzbOx2Yr7jPXDOUBmAtb1BpNVIUCKnmPaqPFTJIHR40U8MyYrkXUoibDx
zXj+Xo9qlyPvxxPPE8TLHjZb4/iPT7j2tM1rKM9MIGSkETNdDCp7HUDEek6bm4nlWoz9cyHlgsfEcMD+apNH0vcGZuyZaT
8dwjARUYKQbsq4YU/ByjyYBa2IzHpmT/3TMNRko+Xs6ojfPEVZUANdFdCxl3ysV4rn7/r4jMx7OfbN1SnnAyaT9xWBI2Pfi
qYY+OyZTdqIbwzIPsaTsnncabFHxHdHS3H9XV3JkzPYOxy9Gjc76bzdckcrnqPOxO958r4RePE4sHu2wWjlp+B5TNej+TIWJ
4X8QLPLwF/7Zz7pYXOu3nzZrd169afd3PiUajyb/e8VPe9vnvsXHtEC/5pRO48gvfpxa2KAY5rrhwE5+77QHOecLRPG6syF
tN+9nX7XB5edt5NhVjXZP1+M31c7a03vRjj7xrfu5Z+cM77j4dD51y/08+/FH877/PHDN1z97A5/a+gCvfuZ6vvrDh7jg9GPY
Pp6rv7aN3zjzBP7qP+6ZP8ern7m+7R7S9L34F4/n29v3sOXjX/Dw/tp8WjafMMqvbT6+Lb/T65132jo+/u37efVZ6/ncbQ/M3
OHDvPCpx3PN17vzL1tGHo6rv34vL9p8PN/9yR7OecK6tnu/6uJNDBV87t9T5ZO3PMArn34S9WaLv7zhnrZrHz1a5H//xz3
xx4RMwy/H2jrZ3785jNh49xqe3PsALn3p82zUgIcM7Lj6detTknf/nrvltf3TB4ykE3sB0Z8voios28e/fe4hnPX4ttag7P4cKPs
9XHbOBO4ZK/IXN8T1IzSxq8p5XvHLj3bd73AxXqD6/754V1eaXvq0E/n2fbt4/i8cw08na2358N5fO4Na2OSt//q9ruN+48w
K3c9xJOOG5/vKzrL8g3nbmSk6HHI9Xe15ckj42VmalFbv50t9zecu5GPf/t+fv+8jTSarmcevPaZp9CkvR/P1t9e6cn2D6NFn/
OnP/6LNx/OfP3yYX9t8PO/vMTakaX/ZfzsR34x3/1t3++k3plbyHn/79e3kfePVZ63nyuvbx75rvn6gXN958enUOvqAtB8sBx6
be9vvnvte0/XgnYV434+Lfv50Wbj+dTW7vbxRUXbuKaG9vTcc3Xt9GIHP/jV07qSsPjxorC+ZO9/Mjx4+yYnGtrd519WloGx4w
DvPAPx3eVVTOPuuyCDYyXayZrER/71v1t849+fWd2XtA5j+vc53XP2sDdD+1j49FjbW2ks/6mfcFrzl7P+7+2rec8snMOMNb
ndT1euT4+29s59KzTuGE1UX2zIzDy8Jbn/8EnIOZetSzX3vN2ev5zNYHuuauV1y0iVu27+YXT17TdQ/pPONVZ63nq3c/xLNC
vRk699w0adS8LvmOmeuKnH/nmpX++xVH7Jp+8PzH0+92ejj37q/7R569TdpOrj9er954IVbTufqr/2IRuR6zh+yeZO2zau2b
9zX3Xx6YwWPT74f7d39V9t6btwU1yfH7+ua8wdrwRgxtVfjd9XpWNC5xyplPd482fv6Fuvs33EmSePt43Dnf1eOgYAfPbW
X91ng+TvgA865TyR/3w2c7Zx7aNA5V/ICz9Yf7+W3PvQtamGL973kybz5s3HFSxWDHH92yRm8+bO3c+IZJ9NswalHD7ft9
6/37+MCNB9LojW0q3ZY97tKzTuZ9X9nWdt/puQbl2/q1w/P3kd13UJ70Sh+wYDmlx2bvPb2vXveU7v/eS87omZ/Z85ua7
ub156zvmeaPvOJMXv4P3+5bjweVQ+f1O68xqO6IzDsr/mXPD/C+r2xb8Fz97u9g69ZC5dvrXGnbG1QfsukfVj7p/XamqV9
m9+vDLf5Gbt+9ZVD1abLvJnn+xfVu/e1zoGtC/fWTLtV+b2Lh2mNd94jtd29evHeb1n/jOgu15UD73qo8tYNSCY8ji8zn1x9
X/uOtfceG7DyoM88W07az+dsv39J90nFqsFOavZ3j5EXPI9N8zHs5XvGRW/rOexbTx03PRT3L6gMvfSqX9iinfv+Ny95cs/zp
JHSPu/3GnoXqQ3pPBzsX75xPH+x7mc65b/Y9yt1JufU7ZqE5fzYfD6Y/zjZVv3QvNB4ujt/7nTsdGwaVV+f8pFcdWGgsWOw
CFJ3FJ4eKo2XwHn6lFbZYT4o6VzXh7yxF/DLxjPzN6Hmd24PnB7Nfr2tI9ZhsRLQezA9I7/7zPPmlajmdDWq49j53XPz2ztNp
f0u3aa5s48a7XcgnV0rhGBo2e963dMr3zrlzf90rxrujawHg8qh87rd15jMe1xstpd/3qdf6FzLeYci7mnhcq317aF8q4z/YvNz
n7+yL+6W93z0udI30717nHXQv6bbZZIzs3J6OnYupa4utW+m1Ftp/sfk825G3i20Xg/Kt5WDXdK1rLjCoDObqg+chLUff8W
fD+QL737mUFzzF7zt+zru2fqfa+dndf2yttBffFco/t6/drkyUO2bF10DUPdnycv98F5sH90pZe72Dn4p3z6YOpL52vtb1HyZ
fD+Gmccx2FnRXwmyTn3AefcZufc5omJieVOzqO2bqRIMYizPP2+dIYxyFHkX9tZBs7136/z7/SDWNnXftrf57Wz+1TyPp5Bp
/LZ6GyXihP5tOV99vyo9810+2Dyi774cA0zzrP2S9vK8XB9a5fHi0mb/qlee1wcWA9HIQOndfvd43Ov7NINFburn/Z82evMe
vT4upW533O+ie0/IuBjkmhgqLrkeLbTfZ85fzi+vbFrrHftcYdN5B95Juq+T9nttLme0LnX+xdauS9ykvYkxYbD5XeuTtYtlxKN
/7ZQj9cH5VX277FK0LesVpV7I1O/e+93nkreb+9nhnv3M4PmmP3mb+nrE0O98yE7TxxrU+/Sa/fMx3z39Qa1ycX0+wudZ6
PYeOudBg86Xzi37nWvVgDQvL5ikOPo0SKHo9zCuyyZHUB2zezYZNth64mPG+Wqio+nGOT44I33csVFm9oayhUXbuLan
t5qvWwO70XbeLvb7yXYhB/t/GjN23n9c9uv+brn72B6+/YwRUXbeJjN9/HieOV+DwwSRrHK3nGK/mutL77V5/IKROVtvu+a
p7d7T++q8p+y1qvWQNz7n1LbXLz9vlyetqXrtGy/ne973lVs2UW2EPa+VLcFXP3sDH71pO1dc1Dv/smVUrYdcf0e8zz/dfF/
rKnku84TNqOeeXP0SIG392h7N3w/rmNpXi22HN5x8emszUyli0GO8Up+wXRny+iKizbxkW/cx3ild35ODBxm73FQ2y4GG
a582HDUMO98wek9j+vtV9703FPxPebPk+Ztv/aWrdNXXLSJlPntj8/d+iATw4Wu9HQec+WWuB/M9hWdx7zh3I2sGyl05U
rj5lIK2z32azfp+dO+7co+Y0Oa9jc991TGy73bT78xdU0IPz9eXnFh99iXzYd39ugD0n4w8Kz4wkHCAAAUgUIEQVRHexji2pu
u+P4OTpkY6mp3/crgozdt55S1Q9TCqGf9XT9R6T1ebdnEZXe1VetqeT7zrGy9SYdDyZn61yZjg09yiqdB6Vj++Xnbeyaf/Tr
evb5822t4xg/R/d4ndT1iaEC19+xgz+58DSarWbPMWFiuMCaoe6+Oc3TfY76vFm/jMLQ/0vld0nnTfHXF/nq0T6T2k86N
fn9TL+61Nk2r0rmoL3PdyAP+81h0nv96E3bOXG80nee0NkeepVvV3G/s2+YGC7wB+c/flH5PujcV118OoFHz/6rLX1pfe4x
tOZ9O6UQ5Hy/kbPM8HXseBH1l+n3PuzIXOuZJ/gwcORNHaOV3j2LH2KFqNZgQuX3DBZ7YRR9HaO9tg3Wj8i+iPTMfRM
/9f+5MKIc+DSiiGISRavaaM5HADoQRavjXHCeHJkoWskvwbDF0QriX04fr8QRkDCj5HvsqS4uitbumTojpYBG1GSkFNBqxR
yn5bFK1G1KQqMeZU4igQs42Q4XxA5Fo4DkTRGi0FNjPNZuvxvc3WQ4pBHLmllmiNVOLGO6lorVnts5QPmCmEZebWR
RQtsxxj5R5RtlbiKFdtUbSKcd3ujKKVjcLTL4rWTD1iYiiOtPHTqRpHDCfn3zFZ4+iRII7O5uvNUMGjGMRrtIYKcfsqNHJkjjgSw
0qBY9K3os/2p8zVpUCGs0Wu2fia5cDj8BPo2jFEVvSKFq7ZuqZSBP1OLqXb/i5A1G0Jqtxtj5aGDFczFOPlvKeR9M1yecORN
z9fgjwONDeYaTKFrNImOoENe/o4aLRK4jilYIT+TiSEGlFByRyc+lkwMIqvWI8aECLddishpH8NpbDVIVClhV8aiF8Q+QTs41G
s1G0JqsRxSDHWCakVhoBrOXiSDWT1ZBy3qMUeOyvNRjKH6ifLdek5XI0uqJoxZEE4yhaDSpfRjPjJomjtn17HlklQP4pWpeC

+yYj6LVojBMnGthHL1odSVPJe9Rj5rsnY1YVYn7hLRfHinF/5O5fy5kzVABOBBFayapu2Ezjoie6ZBMYjnIWHrQBStmVqI58
tqNiIlo2gIkWSKQSaKImvi57z5aKFDry/ZJlpWNemr4mh0MFLyafSjojVbD6lko2jVQ9aNtEfRSseOop/j4f11SgWPyPBLI7la
xX5CNojU+FNBsxtGp0mhQ0/WQ6bm4vg4X4k/EPDwVv55G0UojAJWduO+b6liitboSUK0nEY2SOcKemYixih/3FY2IPTNx
nvHTyRrHJlG0dk2n/b7P1FyDwPPmo/7UkznggWijUA1bRM14PlZNxoTxSp6pesjqczm75ltUatbyZzX5ud6bVG0Ap+ZRsh
BFFazaZc87Ulwlvro9zUTxnGCn55MhE0RoK2D/XI4pW0afg5XhoKh5vS5koWun7h/1zDdZUCnFfnczNS4HHVC1kpt5kuO
EUrcb8PKhS8Mj7SRStoQKzYZOZZN9qEs13phYxMVyg0SeK1lwYR7vdO9uYn4ceiKLVpOB5zCRRj/tH0YooJfWkLYqWZww
htDDLEUYtZupNjh4pcNq60RX9A8uwDD+ybGafAM4G1gCPAFcAAYBz7pokTPrfAOcTh0l/hXNuWZWblb7AlylilililiLyaB3y
WAPsXu5EyCGj8j6yqLyPPCrzI4vK+8ii8j6yqLyPPCrzI8vhUt4nOOcmOjeuuAWew4WZbe31o0hyeFJ5H1IU3kcelfmRReV9Z
l8sKu8ji8r7yKMyP7lc1uWt3+AREREREREREVnh9AkeEREREREREZEVTgs8lilililililrnBZ4DjEzO9/M7jazbWb2luVOjyyemR1
6kXLcm21ab2Q1m9qPk31XjdjOz9yVld4eZPSVznt9O9v+Rmf12ZvtTk/NvS461Q3+XkjKzUzPt+LtmNmVmv6c2fvgwsw+b
jW5/y/ISmrH9sZt9NtquNr3DW/72YxvEs55weh+gBeMC9wMIAHrgdOG2506XHostvHfCU5PkwcA9wGvB24I099j8tKeMC
OfaC5b5nPebL2QMeBk5QGz98HsBZWFOA72W2LXmb7ncNPZalvj8D+Mnz92TK+8Tsfh3nOahy7Vd39Fi2Ml/yPhx4DXB
Rwz4jCLgU865+rOufuAbcR1oGc9SFalnW18Njn+o8ALLuZu5GdwMXHZQHsZXQxc62I3A2Nmtg54LnCDc26vc24fcANwfv
Vp4/38PPvwbF34LHBO+j//snQGIxES/78OfGLQOdTGV44B78U0jmdogefQOgb4SebvBxm8QCCPUclHb58MfCvZ9Lrko3
5vZL4f2htJ5nZd8zs62b2K8m2R1Oumu899ix1Hz5/TPL6/mR/WT6/AjzinPtRZpva+GGi472YxvEMLfCIHCQzGwl+B/yec24
poeA451zTwYuBz5uZiOLPZ/K9TFNffir6SW0/0eN2vhhosd7sXkax7XAc6jtAI7L/H1ssk1WCDMLiDuUjznn/hnAOfelc67pn
gQ4FG263zVkgZjZy4ELgd9MJuokX9PZkzy/lfg3WDby6MpV873HkEPUh88fk7w+muwvvyApg/8OfCrdpjZ+eOj1XgyN422
f/NvD5pbwn6c/MKmy2nD4n/mHO7xGXa/pr+9kyug54WfKL/U8D9icf5fwy8BwzW5V8LPw5wjeT16bM7GlJ3XoZKu/Hirb
N2vjK1++9GBrH27nHwC89H0kP4l/zvod41fiPlzs9ehxU2T2D+ON4dwDfTR7PA/4RuDPZfh2wLnPMHydlfTeZCEn96gFxxl
3/eR/gAqxP/rOprZpjZ+mDylF+4eAkLi79a/6lC06X7X0GNZynsb8W8vpON4GvnohUlf/13gNuCiR1uug+qOHstS5kvehwPP
lilililrHBa4BERERERERERWeG0wCMililililissjpgUdEREREREREZIXTAo+lilg8pphZ08y+m3m8ZcC+LzCz0zj/X2Vm5/4c0jBm
DOubf9nNlwBrwGeP/Pcl6Par/unLvdzDzg1J9LyHjm5jvnop/nOUVERGTI0id4REREZEUwsz81sx+Y2R1m9l4zezqwbFjz5FM
rj83sHjP7Bu2LOGuBhwCcc03n3A+S/c80s2+a2XfM7CYz61r46bePmb3czK4zs/8EvmJm15rZCzLHfSxzDyliInIE0Sd4RERE
1gA3j+d8C3B6+kkiM3sOsAE4EzDgOjM7C5gFXgw8iXhedRtwa3LtwLuNrOvAf8GfNQ5VwN+CPyKcy5Kvk72LuCFHeketM
szqZjZGvEDzrmSxpgUcAxzV4zzPSR7fSf4ell7wGQb+xTIXtdKZXg/n3FVm9rHkuN8AXgKcDYwCHzWzDYADgh7XG7TPDc
w6GfBu59yTksd659yHfPhue51zfwucA5xhZuPAO4CvOudOBy7qc71B+8x27Hst8FvAK4APL5QmEREROTxpgUdEREQe8
wmVxzR7L95QPStdA+qY8AvweQ/s6PililHHn0FS0RERF5rOn8DZ5/A/4a+LyZFYk/TXN58tongQ+a2euBSx7FtT4GfMHM7g
Ffg/fdK1mH1I0vqlmd0F/OujuH8RERE5TJhzbbrnTICililiKPkpmViX9j6CnOuf3LnR4RERFZHvqKloiliMgKIUTYugv4/7W4Ilylcr
ilrnBZ4RERERERERERERWuP8H5Un1bmDVYlgAAAAASUVORK5CYII=\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].dtype=='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": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAGAEIEQVR4nOzdeZgl510f+u/b6+xaRqMFy/KYO/ImGWwzGPKQOEZerjCKRB7ASxjsC
kePdNVp+qtd6/qn6rPW2qtAQAAAKC7xtY7AwAAAAAB8dQR4AAAAADpOgAcAAACg4wR4AAAAADpOgAcAAACg4ybWOW
gAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg4AR4AAACAjhPgAQAAAOg
eEwj8AAANbF37958/NN3ZmHTHeudFR5nxo8eSJLcP+Px9lw3fvTh9c4CwLpwBwQAOMvh04U59rSXrHc2ejzZ+NI3J4m+C
AAPAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAA
APAAAAQMj8AAAAAB0nAAPAAAAQMj8AAAAAB0nAAPAAAAQMj8LAu3vSmN+VNb3rTemcDAACAc8C58DvoxHpngH
CPAAAAAdJ8ADAAAA0HECPAAAAAdJ8ADAAAA0HECPAAAAAdJ8ADAAAA0HECPAAAAAdJ8ADAAAA0HECPAAAAAdJ8ADAAAA
PAAAAAdJ8ADAAAA0HECPAAAAAdJ8ADAAAA0HECPovgXe96V57//Ofn7W9/e2688cbs37//tM7fv39/brzxxuzdu3fZ85
rfmU5/6VG699dZl2+Oaz5faXu11x08/tZbb82RI0dy6623nIYaDzzwwKqOBwAAAFYmwPMYe9e73pVaa3+71pr3vOc9p/L
MT652Bc03z9s6ghYWFvO1tb8uP//iPr3j+W9/61iwuLq54/uBxCwsLufXWW5fdXun67fTe9ra35ROf+MRJx9x666359V//9V
yZ9++feudna/K4Ns7jfn5+bzvfe9b1fnvf//7Mz8/v+L5g8fNz8/n7rvvXnZ7peu303vf+97Xf3un0d5eKe/tcgAAAAABnphNv8N
b6f3ohe9KJ/4xCdOCurs3LnztPK+ffv2vPGNb1xVuQEgSV73utfl9rt8jxsw2uKGbdn19Zd4zgROci681dejN3jOJj/2Yz92yr7x
wcAAABWR4DnMXbDDTeklNLfLqXk2muvzfbt21d1/vbt23PttdemlJKdO3eOPH/wuGuvvTa7du1adnul67ft2759e3bt2tV/
rPV9pe7XUHj3/DG96QzZs3r/j2TjsNb+8AAADA104Z9qW/j2e7d++ue/bsWe9s8FVq/v7R30YDcDqa7+A59rSXrHdWejzZ+
AxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAH
xwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAH
AAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8
AAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AAABAxwnwAAAAAHScAA8AA
ADwAAAEDHcFAAAAAAdJwADwAAAEDHcFAAAAAAdJwADwAAAEDHcFAAAAAAdJwADwAAAEDHcFAAAAAAdJwADwAAAEDHc
WAtPqrXuaO/sXIAHvtZKXtqrbvXOX/weGNswHDGBgxbnMCpjAseS/5ECwAAAKDjBHgAAAAAOk6AB5K3rHcG4HHK2ID
73tvb3Pd65nvmEtlVLuLqV8qpTy8VLKnt6+C0sp7yulfKH37wW9/aWU8ou9sfHJUszp1jf3sHZKKEeXUn63IPLZUsqdpZS/Z
9Qkkd6+/9j7zg4m317rfVZtdbdve3XJ/lArFXKJB/obSfjdyS5svffq5P8p8c8p/DYeWOS99Ran5bkG7N0/zA2OKfVWj/Xu188K
62+I9/oLe8XCuGBwD7bHxtrrkIOnOL6Vcth4ZHLVUSjkvyfOS/JckqbXO1lofjbeBg16Q5C9rrV+KsQFJMpFkYyllsimmJF+J3zd
k1zS+7k/NnoGxw2cTZ6cZF+S/9r7095fLaVsjeBg16e5O29n40Nzmm11vuS/Psk92QpsHMGye3x+wbrQIChs1op5bokD9
/W+d+qGLP0Pgq9LsjnjteuaKc5ZAJyc7b4tyfWILLuT/HaWXPv8Y5ZeE57oHXN5kvt6P9+X5IlJ0w8vCT7H8sMw2OI93+cUm
tqrXNJ/t8s/Q7i9w0ecwl8nNVqrT9da7281rozS68Tf7DW+neTfCj9/QOe1WSd/V+Vq23nd7nH+z93yg4q5RSNpdStjY/J3lXk
```

7Bo+5oi9xriilPD/JT9ZaryulfH2W3ui5MMnHkvy9WutMKWVDkt9I8uwkDyd5ea31rvXKM6yV3hj4vd7mRJL/Vmv9V6WU7U
+9w3OOeVUm5O8rlk81n63eLvZ+m7dvy+wWNKgAcAAACg4/yJFgAAAEDHcFAAAAAAdJwADwAAAEDHcFAAAAAAdJwAD
ADO3MR6ZwAA4EyVUkqS30vy1Irry3v7vjHJJUk+39ueqLXOn27atda/P7D5z5L86156fy3JdUmeU2udKaVcIGTqqyzHGeUR
nlw6WU3aWU/yfjxt6bOr+V5LlkD9VaZ3rXe6jW+le9c765lPInpZRPIFL+vJSytZSyoZTYX0spnyqlfKyU8u29Y7+/IHJbKeWDST
y1CTPyNjbP59j8muDidRaX19K+dFa67O5pJSyJcnPIFI+n+T9Sd5Ra/3/SilTSd6R5GW11o+WuRYIOZbkduVj1GeWUp6W518
yfOSvL3WupDkr3pv0yyr1nq4lPJNSf5Glt4gekcp5fVZCjR9pdb60d5xB5OkIPLXk7ypt++zpZQvJWkCPO+rtT48kMfrSyk/2dv
99da/3cmeQJADi3+Q4eAKDLPhkuvcnWEmSUSo3ZOntmkHvTfKPSimTvWOeUkrZnOSPk7ys9x09I2XpjZxh5gbOfWop5
TyU0mOJ7k7ye+3Dv3VJDuT/EUvgLlvyXdlaQWua7L03Tv3JPnTEZd6S5JPIIL+IsnPJ3IT73ty5pPsTfLqWutsKeVlvc82Zun7d
QcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8
AAAAAHTcmgZ4SinXIII+V0rZW0p5/ZDPp0sp7+h9/mellJ1rmR8AAACAs9GafQdPKWU8yZuTvCjjvUk+Wkq5rdb6mYHdfij
2vNWck5MjufzVMTmZ1fzlbj8Rybn8vU+ET2HZrJjq3TuWDTTeB45upAHD83k4q1L522YnEgpNbPzNfsOzWT7lqkcmZ3Plq
sZK+M5eHwu2zZM9v999OhsLtg0lW0bx3Pg2ElePTaX8zdOZv/hmWzfMt0vz/mbxnNstublzEI2T08kqZmZX+yXq5TF1DrV
4Zj5HZxf6ZT9wblF/7OGZuWwZKPOjR+dy/qbj/v6mbo7Nzmfj1ES/jpo8tOus2d8/r5f383p1NjleM7dQBSq+INcmP8lCkN
4Nhszts4lUeOzuWCTZM5b+NYDhxb7Oen6RuP9PrMQl3l+ECfaveFrRvGcuJ4Yv86x+fns2Fiol/vs/PzmZqY6NdLMzaadjq+
JLxPHR4Nls2TGTbxomkLmZmPv08HJmdy+apE32pGb/NNZo+ODM/n+mJiTxd6DbnbzrRNk2d7D8yk+2bT+S9aZv2OG/+
O3zfnHe/NJc9ymqeTobPrLaOqr6Svt9E+Ue6m+TvT5pTF9dHY+m6Ym+vNg0yf6faU11pr2a/pMU/6m3E39tLeb/DT52ziZ
4aMrjMOMLZuyPHx0NhdumurvbaZpoz9PtBLr6mrpu81123arOI7TTrNfNFUUDOmTZs36W7dOJZDx07MLymLST3RR
p0/MUe3rNWOjab+mPzTlb7dHU+5LzhvPAwcWTqnH5ryDx2azbeNUds/OZcfm6RydXXRouHjbdOYXFzIxcE9rj+/mnt1cq
+T7xYbj5Phc+vlu2uTw7Fy2TJ1lrxlLTxPn/TXHT+eldr6ae2NTL005m/pv6vXA8bmct+HEM1jzbz/dXp9r9jfXb7abcjTla8Zmc
nL9jxw4plgqS+d6GMnP5f0+0KvTg4cnc15m6b6+WqPmaavN2U6MR8s1cGjsbFU58080MwTzbzRIGPU/N6k0+7Ll2wbzw
zcMx7e8f8aSlIln9SxbUZTK1e/fu6kuWeawsLta854778xPv/HiOzy1mw+RYfv6lZ8qLn35j/ujOB07af8sNV+edH/1Srnnapf
lz+8dFzx3n//i3vy0t1Pyjv3nHq9W66/Km8eOP6W66/KO/fckz1fOnBSui/bfUU+/uX9ecHTL8svf3hvxrb7iv7+dnrv3HPPKde
m685sp+fm9qpff+O7+Sb/u/Ls7RuYW88QMnzvnZG67OL33oC/26GkzT3+mVqSznqH+HteVgGQfT/YfP29VrmysyPz+biY
e7HDdff1W2TNUcmS0npdv0pW9/6mVD02v36RuvuTJzc7O57lItP/Stg0ePZeumjf3yvfgZF51UX4NjqMn/YHnf/OHhfa89F
+/bLdV+Thw0fzIEvPP6Wu3zxkXvre3VfkA3cutWW7DzZtPLi/Gb/PeuL2kW2+0vz4gd75pzN/DZsnP3//o7lwY6ahbf+a5+/K
rMwrpdX0jWuedmk++Nn789LdV5wyf4yab9vz6rB72Wuevyt77n4o37TzolPy1dTZ4HgfV5yY2RwLA321T/6zEMn3csG5
MvfekYWUnLzHwy/l4661w57jrrmaZdmamwh2zZtPKkN2/fl1YyX9j1qWFsP5qMZS8s9f41KZ7BuRs17g/lpl39UuoP1stl9e
zUfi4Z9Xw2rM/cfvfj96phzxCDdTwsv8vVYXt+H/UssFJ9tX+XGjBPH3n+rvzNK7flj79wsDWWrs6Lr97R2SBPKeX2WuvuU/av
lb/nlKfv/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/p8NeVfbsydbjXOn9Yv7jxBbuGPpeOmsube9Bq6mhwjnZu89Nqy76a+WA188xqn5tH3atWqq+V0m/+HfX8
B4/rpzc6vkt3B44el++iRuVOuMzT/s8Ov05zfaOdnVHoPHDO+NP8r5eOR1vX6+Tg6l8U6ug6Hlb0pU/uaK9bFMNu+mG5T
Z1QZR80Lp5PHYX121HGj8j5qXho5TkfsH3X8asuxU9ZrnyD28uVc1g+V5p/R33+0OGZMxrDo/rDSuU/0/p98NDxU8pwp
y6n3A+HHHdkhXNP57nodNtwNZ8vl/dhbX2mz1+n27arzd/p1t9KfXalZ63TLdeoOWRYfleTxvXLucrzh+Vj1HPpqLm8uQe
dv6DG9P9E6L8n+dkk11rfUWnfXWnf2LFjblLp7pk24ZsmDx5mGyYHMTl520cun/T9MRp7R98gW7U58OO2zg1cVrbpy
MOq/dNqPqelibtetsue0z7Wvt45rrD5bndPrMqDE1rC8ul/9Ltg5Pp71/VJ9p2mlYeb+a+hnV3y4ekd+xkpHnjKqrUXk5nTyC
vax/zxpy71rNuGvOH3Uv27F1ej0288FK89Wo67bLdf6myZPTHZfGR6V3pn3mTOerYfe5Yddttkf1zeXmltPJR1O+Vd8Phxy
qU6GNy+8DSfn1bbxttXMR+spi1W+9w86llg1Bi9eMj8vllw+R6Wz9N1IZ5exlQ85Yx9NcmUp5cmllKkkL09yW+uY25K8qvfz
X+8JNLMc8/+MR9ueX6q0457w8/eV9uw7Ecc15b/uTu3LL9cOvd0vr+FuuXzq+ne6N11yZ3/rIF3Pz9VfIdz6xtN38205v2HV
m8roXnHzOz95w9U1lMvinTZnaZV+uLpq2bOehSfdE2yyVsV037bq4udU2o+p6MP12nbfruKnD5fracPwP9p1RfbK5ftN2
+dkbrj7p+Ne94Mrs2rFl6Dk3XXdV/vQLp5Zx8PNh57XbsN0XBsd/8+9yffrGa67M++4YPo6HzUs3X39VfrPXLqPaeNj4Xa7N
aOHlXmItJq+0fS1YfPHcuNucF4ddi+75fqIoh2Wrz/t1dmo+Wq5MTKsj/zWR7540nZ7nl+urw6Wf1T9jeozTfnbY3u5eht89J
XVPHMMu177Gwa5/LTLPyrdpm1Wcw9eri3bY35YX1muvKP67HLprtT3hl2n3c7D+slqn+uH9YuLtkyPnJeGXft9d9y3Yp9c
p5pxt1uw7eJkklPKSJL+QZDzjr9Va/1Up5ZYke2qtt5VSNi5jSTPTvjwkpXWu9aLk3fwcNjrVkt68FDx3Px1INX0Wr2N6toPX
krq0CslFm6cyMVYzv4pVtNorSh06PpetQ1bROm/TeB49utA/b9QqWkdnF7JpanzgzzqGr7jSrNZS5s3koq0xMTp6xC015
mtmV7GKVrsso1bRavRSqtoNauU9FeM6OWrKUezyLU7/1umSw4vs4pWszpDexWt9qo07ZXZmhXS2qtoNX2iWf2hyWe
f04fyQwv2Vf3p18vCRmVw4slpWc1zTJs14b/5tr6LVXGfUKlRn6itNXTTr1tR9e5w22+1VptqraD3cXw1w+CpP7ZXfRq2i1ft1
18PGcv/HECjctdyUqbnnNNvtlcPaK+g0q7X0+3BrFa3+alb9/UvzXtNm7VWamr13bXHqKlpLfaRjp73qVLO/3+a9dJtVtjpyN
b+5p8tH0h/aqgk167VW0TqxS2F5Fa+kZ5ujsXC7cNj1j4cVzd3ipzzUrEDV9o32vbd+z23XSX8W0d+32ymbN+GzqeNQqWs
YpW80xyyqqlrb7bfgZrr6LV9Llmf3P99ipaTfmasdlcd6VvtJoVRpu+1I9Fq3Xfapv6M4eNwKXr8MzSdRcWc8p4b67V9NFmu
cqaXP99ipazTzVzE/NcWfbKlqP+ZcsrxUBHgAAAOBc1ekvWQYAAABgNAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAAD
Agl4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI4T4AEAAADoOAEeAAAAgI5b0wBPK
Tfjv1yo/AAAAAGertXyD57lJ9tZa76q1zib57SQ3DB5Qa/1QrfVob/MjSS5fw/wAAAAANjXWMSDzhCRfHti+t7dvlB9K8j+HfV
74kTzxYvry37ySllBcm+edj/matdWYN8wMAAABwVlrLN3g+muTKUsqTSylTSV6e5LbBA0opz07yK0mur7U+ulZ5AQAAD
mFa3l9AAAAgHPB4+JlIgEAAAA4cwl8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8
AADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB23pgGeUsq1pZTPIVL2lljev8xx311KqaWU
SX4uyfE1zAsAADAWWstAzzPSPLlge17e/v6SinPSfLEWuv/WC6hUsqrSyl7Sil79u3b97XPKQAAAECHrduXLjdSxpL8fj/vN
yk/sdzntdafX+bjja5spTy5CwFdl6e5O8MnHsgyUUD1/pwkp+ste5ZodsAAAAANjYN8GTPZLskeWqSb05yW2/7byX58+V
2zx0s+AAAAwLmnlHj7rfWUr7dZ7XfwXjJkdmB7trcPAAAgHW20p9oNd6W5M9LKb/X2/6uJG9dmywBAAAACDpWFeCpt

AAGD1VvsGz99Ocn2SI0ISa/2rnFhCHQAAAIB1tNoAz2xdWk+9JkkpZfPaZQkAAACA07HaAM87Sym/kuT8Uso/SPL+JL+6d
tcwIAAAAAGdm2Td4Sin/KMmPJpN6UsonBz7amuT/rGXGAAAAAFidlf5E678l+Z9J/k2S1w/sP1RrfXjNcgUAAADAqQ0U4Kn
ajVSvmh1vZ4KeWmtckSAAAAAKdjtcukv6CU8u5SymWllKuTfCRLK2kBAAAAAsM5W+g6eJEmt9e+UUI6W5FNJjiT5O7VWY6
l8t9Z6MElqrTXJfyil/MHaZQsAAACA1Vr2DZ5Sjy9NklrrwVLK97Y+/v6VEi+IXfTk+VwpZW8p5fUjjnlpKeUzpZQ7Sin/bbUZBw
i52Hbbc9NsrfWeletdTbjbye5oXXMP0jy5lrrl0ISa31whTQBAAAAAfkpWFNH/Dxsu+0JSb48sH1vb9+gpyR5Sinl/5RSPJJKGf
QbXWtyR5S5Ls3r17pcASAAAawDII2QBPrXX8q0j7viRPHNi+vLdv0L1J/qzWOpfki6WUz2cp4GMJdgAAABVWulPtL4aH01y
ntvkV2lIM8k+VCSf1Jr3b9WeQIAAAA4G5Vau/WVNrt376579uxZ72wAAAAAPOZKKbfXWne396/ln2gBAAAA8BgQ4AEAAA
AAgl4T4AEAAADoOAEeAAAAgl4T4AEAAADoOAEeAAAAgl4T4AEAAADoOAEeAAAAgl4T4AEAAADoOAEeAAAAgl4T4AEAA
rSinPaB32hiTvrLU+O8nLk/zyWuUHAAAA4Gy1lm/wPDfj3lrrXbXW2SS/neSG1jE1ybbez+cl+as1zA8AADAWWktAzxPSP
Wutbaq27a627d+zY8ZhnEgAAAOdxbC0DPPcleLA9uW9fYN+KMk7k6TW+qdJNiS5aA3zBAAAAHDWWcsAz0eTXfKKeXl
b6/1lrXk8AAAAAZ6OjtUy81vruLH158uC+nxn4+TNjvm0t8wAAAAABwtlvL1kGAAAA4KskwAMAAADQcQI8AAAAAB0nw
l8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAAAB0nwAMAAADQcQI8AAAA
zxsBkiucvLtbcvf9IHjh4PjumJkK7sJDTm6dPOX/wuEu2bcgVF2zKPY8cHbm90vXb6e3cvjmLizV3fOvAvnLgeC47b2Ouumxb
2zi16rZsp/d1Wzfk0/cfzP0Hj+eSrdOZmhjL/ELNgeOz2b55Q55y0ebc8cChHJ6Zy6apiew/PJuLt05ncqKk1uTi8YVMT47l0Mx
Zke2g4XbZrKw8fm8uDB49m+ZToz8/PZODmZ2fnFXHreqW0yP7+YO75yll8enc3GqYkcPD6XbRsm8/CR2ezYMP3NG8Zy
7GyTxydDYXbJrKWFlMrWP982YX5jM1PpFHj87lws1T/TY4fHw+UxNjeeTYXLZvmsrTL9mSzzxw+KQ2u2jLVePzZGld61/nW
xhTx4aCYXb53O+Fjj+NhYjsaTlzMLefDgTC7eNp25hYVMjo/n4LG5bNs42U9nYqxmfrH089Ok0xw/WF+Ldaw/j9YspmQsF
bZrq72+uu23jWA4eW+yXq2YhJeN5+OhsLtw0tdTnpyayWBdTylj2HZrJ5RdszPG5hew7NjLtK3n/E3jeeToQr+PNGINT9TM
NW/TrvOm/SbdNp115w/Mz+f6YmJHDg2m/M2TuXRY3M5f+NSH5scn8hDvT7U5Ku5XlOOY7Nz2Th1onxHZueyeWqyP6
77/PjcfDZMTuTo7Hw2T2Zor6bczZjt9+n+2F0qT3PcYl3l2EC6TfmbftDU94Hjczlw2Qme/P0g725rmapn+YOt3vm+dtHM
Yl5umkqOz6R/Xbj/ou0vjtLnO1g0lh47XU/LTtEEzFps+1ZzXHjtNn2jSabab6zZ9r2mLZj5rtvcndsmOLdM5PDOXLdOTp5zsf
pT+WetC5MjOXzOdTt/TifGa1LEcmPnPsbmFXLxladxMjj+Ye5v5qbl2vy56ZWvadR81cw3GyaT43Mn+s7WjWM5dGyxP4
Plv2rRpi+b6zXz2tEs357P3HzlxbY81C7X056FmPmvy1Zzfvc2bdt83vTxpv6bZ4Jmnrmry31x36/R4Ds0snjve2OvSa+5n/Tv
t5+MhsLjtvQ0pZeoZvnu33H5nJlukTz0/bt4znocMLj34/HC+ZWagn5SHJ8RybO/E8+oxltubzDx3O4ZnZTI6fGHePtp4Pm2
u3Za/OnS8/zvPzPxSHT9ydDbnbTzRtvt6bTs9OZYt05P9Z/vmufro3Fw2Tk5mcbFmbKzk0aOzS8+NrXI+cPB4Lu79bjO3UP
JG+stX7LSunu3r277tmz52ud3cfEsWNz+YNP35+fue3TOT63mA2TY7n5+qvygTu/ku969hW59qpLVwyyvOeO+/MT7/x4//
vEJQ39ZXynvL376JfmjOx846fPl8nQ6dTSYzKp10xzf5Ofn3nNnXrb7ivziB79wUjnf9MEv5Ev7j/XL8cHP3p+XP/djl+u0ue6w
d8Kd/9nP+/vXMPsquq0vi3+p103ulO0jySgOkE7QaS0ESsUSoExvIRO0yJBERlaUVHNixTlqNOITGtY83oDCoCIGNookyMoiE
v17RizY7DWfM2v2Iju+1tXXhked3ZY3T0tyAFWacuN62n8/PqQ/6TUdHNx7e+Aa+9pu/eM+L6/2FK6ZiSHKjwXfL2XaLprb
qyl53X4mjR2Cz86egkWPbMqywfB9R9A0udY7Z1buOedNyBlv0tghuGX2FHtzkU1Bua1Pu/0unFOF8ZU7V72SVw9r//Wt+9
7evfAVTxw3Dte+ehMWPxmVoxMu7D+ld40blyPby7kOonzAqa+4XzW3Ar9b1zu0ts6dkZPXpaG3n6hKPDZ+PxW3u+kw+
m+6vbbNGkkPtY0MWd+n9oc5YxC5LF/P3PpFDy1eRcuf2eddz598rpyVpWX4NbL67H0jztw8ERHRqYr3lmX4wM2v7m+6
8EvKReD4KyZkvZuj2sDH359cOeO2QJoYWNzdg98FjqBs9zLsehGLajf3r3jMZu4+047YnXs60+ca8RtxhrpdCtr/L8cm9hyN
S9I73n88z/t89pbZU4L2X76mFZ+dPSUzb2n08+XPbN8f7b12Ca1HvhiyuSN0jWTXh/KyEnzl189njin39lw1Yt2M/ms6pxZ1F
d9x4tc83vFjeobyqW23Zvu+HB9vaW7E+xtri/Ymj4isVdWmnP2n6gaPGXQygMcCN3h+BGCvqi4z25sBzFbVXfn6LOYbPM
AAFa9JREFUf9ga3Lb9hMYPjbvg0nNx+5NbsvYtX3AJLjx7dMGyL19wCebf80xqmQq1UVrbxPWYf88zGfl8ct/51JbM9revuh
C57l6+3zSbWNI3fDawaw5Ds2Nq/vPbn43miaPSe1LVvekaycaeP1/hsuxk0/eS7vnH3pwQ0546XtP9Svq0+SL1o5k+ya1g
V9+QD4Z8zvbb15hwYzifXX3H3X5D/uDmy7Ty2L+hfnTx/8iXw9OuSUnxHZqbtHNa6Fy6a1/Ijm67kHxjcoT0SJlvdH6SD4fn
80nae1g5UjyAbffQq+PQz6TNOe2v7T6heydlGPSzkdSLrP9+65HQ/HrjpU2npLWSuurIV3yreHxtT9pjpLWB/f8pOu4QuYm
092Hu0LeH8Nu/5lsg6321nj4e2bT+h8UPj9jj3Bts6e7D7cGF9WFI2HfYft7JJ2nHS2sa2t/KEjotkb59s70o1btJ4SeMcOt6Zqp
/1p44Z77x0vZvz0/Sj6SH3T5wzO8Trl3T+mDouB1nz1G/zyTZKe3+nNj3xAYA73yHZNgXkDm0351b229Qx8D+kM+5uoW
LztrX8gmbru0sZBWj1C7kFw2zxXis/nkzMh1Njyn439DdijKojNNTvPpYdv1aGFraMgWexN8LiTroROdWeOEzrduUaHx6M
PUh7doX9MEEW7sxGwoXynN7jrRjoFEUL1IW1XtUtUlvM2pra/tbnD4zfkQlqsqzTV5VXojRQ8tRVV6CccPzPx42fkSV93xVZ
+zx0N6DKkoy9q28+b63oSQj+eRO5/f1CX4jau3zyfdNlbWul3yzY2r+/gRhflSXpdcCfCqP39phlXnnbGhFWd7xkrbj5yfpk0+P
om+DbtQGfi9uzLzHh5r18dk1jh9Bxm5cLlcf+HR3lt4XGgPWPfH2mXZNGJcR30tykmdN8tsqR21n73OM2xnzt3oocNQe93
L23e9a1h+cYj5Ye0/ae5FknKb4XoF7J3bcJ8p50PmxOS8r/vejQeZ33x4Xw6+WRwfTVtPvCt/fnGCX7HCOsL4LVQyAcTbO2L
qnHYxt3Zp3vtnt0w05888rG4HZVeUne8X3j/sfHpqN+3LCsfd+8shENdSP7JHtD3cic42lskmacQmxj21t5Ht2wEwn1Ofo+
+dXzcyZ5yW2Di++c/nN+efMRlt8xqD58X1/sIVU1FTXZHVdtHchhzbWfkb6kbgm1c2ZubENzctju4t8xpxwRkjg77k2s7qvm
bEn1wyeqtOf21NPeOs7F1f87xkD5xO6XdH9fP9Y/FnnmM2+k/n34Viz7iytCIFZv8vvPEpty5XzQ3GtPGQVzWkO18usRjw+
FCJP3D4PPLPNq39IXl9M3Xp5PX697vUsmXw+4MoaygMP5InvRXOjuWnxzl3t76F1ralixJfffPkne+3zx1hc7p8Zm/rks76VZ
F0vhWzv+qSVJTSmT1br83EdQz77+oEjWb5TaF5O8n039nz5xpefXTssNXnl5wO9+aExZ27TXAPE83zoWiTfnMfnLY1+i+ZC
nX6LW5Hd8X1761z/qg+x0nnn+S9AzlU9vO5+MtZ2Y0QgW75rFQH++g+fDAD6H3pcs366qs5L6LOZ38AC5VbSqqyQ3hv
tXj2SbBrqL6R/vlpWzbBKlJUA12JVtNLKb/e7VbQqy0vQ0aU42taBmDWVmFozDjv2HMXx9s5M9ZiaYZWoLBP0aFQFqaKsJF
rQkjqnBBolrWniNtKC8tQxtXV1ZFoWAVLVO5xlbssFW0bGUHW3Hj0lloJk6u6j2Dti6U15Xg0MlOjB5agXeZ6gHxORtbHa6i
OoucXulqqlJqtYfWfFlqu3rdTUq0dUXStYRcupDmSrCIm9eqvzmKoypkJjporWsXacNcpU0TjV6kabKlq2monty62i5VZhyIT
l7q0xVVZsVRY7124VLvUBxNrBHRfyWf3cSnJudRSb9zK+aiq+uVW03GpmtopWry9HVV5sNSzX7lY/K487r9ZHbezY/ZnqX
cbevJkctotWbh6LzrG/YOcxU0cpUEcyuqmV9z/qijQ23cpH1NetjbgUfmwPcKlp2znOqaDkV9aw9MznFxGZSFS07HtADoC
zFTRMnnlrajlBzTTrctC7h52fp+ThUts9Zmqmg5MTGsUnAsTRUtxw52Tu1c2PFt/rFvTkz8SVW07La79lo72ePWx639bftM
qtoTRhRhVjZDZ9TRet4O2qqK1EzrBT7jvVWG7VvtHrzYClOdvRejzZkqmh1ZIUZPXSyE2Ni14f2mn9MdSkOHI/1b6pNdf0
5Vlljg8MmOTMXKuj62ipb9bnPIVOuKqmhVYlrt8AFZReu0v2RZRjYBmA2gBsAeAlaAOAqt5tyqTfAeADiMqk36iqiXduiv0

QQgghhBBCCCFZil9yfKpQET2AdjR33KQAUUNgP39LQqhf4UwNgjxw9ggxA9Jg5BcGBfkVDBJVVWvdnUV3g4eQtxsRWeNfhgbhPhhbBCSC+OCnDb4Dh5CCCGEEIIIIYSQIodP8BBCCCGEEIIIIYQUObzBQwghhBBCCCGEEFLk8AYPGRSISKmlrBeR4eIjBKRBOXkJRF5UUTew9gggx0RmWbWC/vvIlj8A2ODEEBEviAim0TkLyKyTESq+H2D9Ae8wUMGC7cCeDG2/W8AvquqUEGNaq62awX0wFcBOAEglfA2CCDHBE5E8BCAE2q2gigFMA14PcN0g/wBg8Z8ljIWQA+DOBesy0A5gB40DRZAuBK83me2u8PsG6Qd4g4cMBr4H4EsAesz2WACHVLXLbL8O4Ezz+UwArwGAOX7YtCdKlIAHheRtSKywOwbr6q7zOfdAMabz5nYMI8Ifle1V1JqLH6G8RkUvjB1VVEd0ElmQwUQZgJoAfquoMAMfR+5MTAlwNMrgx7xFpBvBL9xhjgwxGzHun5iH6D4IzAFQD+hF683evaZ6JDUM8bggZSLwO4HVW/ZPZfhDRDR/GBiERHwSwTIX3mG3GBhnsXAFgm6ruU9VOAL9G9B2E3zflaYc3eMiAFQTMmBQ1d0AXhORaWbX5QBeAGODEMu16P15FsDYIKQVwCUiMtS8S8euG/y+QU47Ql8igwURmQ3gi6o6V0TORfREzRCTNliMh0RC/mrwCwFcCNIp5TjLFBBjXmPwRaAZyrqofNPq4bZNAjlosBzAfQhei7xacQvWuH3zflaYU3eAghhBBCCCGEECCGEEIIIIaTi4Q0eQgghhBQ1IjjBRH4ulq+KyFoR+a2ITO1jXz8RkavM53tF5F3m81eddv8slptEZKOl/FIE3v3WNSGEEII6TtJOMhBBCCCEWPsFDCCGEkGLmMgCdqnq33aGqGwCUisjvReQRAC+ISKmlfEdEnjNP3XwaiG4QicgdIjZRFYAGGf7EZFVIt85b72IzDs9jISEEELIqIBP8BBCCCGkmGkEsDZwbCaARIXdJilLABxW1YtFpBLA/4nl4wBmAjG4F2Invp5AcD98U5U9csi8jVlERgF4VkrWqOrxt8dUhBBCCBNl8AYPIYQQQgYqz6rqNvP5/QAusO/XATASQD2ASwEsU9VuAG+Yp2nyoqrHROQiAO9D9AK4KHIs/+SIAPq+qv4s3EjEP9WVQc0NoFYBVlVl8gOsRfplO66MH1XVzX2RiRBCCCGDG76DhxBCCCHfzEoAleYnWAAAEb6bOL8D8Hnz4milylY0RiCEEEII4RM8hBBCCCLaVFVF5O8AfE9E/glAG4DtAB52mt4LYDKAdeYGyj4AVyKqWDUH0bt3WgHGEEIIEVXtbxkIIYQQQgghhBBCyFuAP9EihBBCCCGEEIIKXJ4g4cQQgghhBBCCGkyOENHkIIYQQQgghhJAihzd4CCGEEASUVORK5CYII=\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])}')\n"
],
{
  "cell_type": "markdown",
  "metadata": {
    "id": "tTwXFuDxvDjo"
  },
  "source": [
    "Removing the outliers"
  ]
},
{
  "cell_type": "code",
  "execution_count": 53,
  "metadata": {
    "id": "823mThKxvGbl"
  },
  "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": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
XRwbG90bGliLm9yZy+WH4yJAAAgAEIEQVR4nOzde7xlZ10f/s9z7nPmmsxMLgZCwEwQEpXLgPrSljctKib2J+XSVkVpqVUu
x12Wt913Nba3+z935KrTUAaaaaAtNfQuQ4AAAAAgC+OBA8AAABay0nwAAAAALScBA8AAABay0nwAAAAALtCyLkO4IH
BAAAAaDkJHgAAAIcWk+ABAAAAaDkJHgAAAIcWk+ABAAAAaDkJHgAAAIcWk+ABAAAAaDkJHgAAAIcWk+ABAAAAaDkJHg
JHgAgL4OHZ6cv/r4J7MweeG5DoUWGp46niS5a8bjpsZPHXfuQ4BoNXccQGAgrYmL8yZL/vmcx0GLbTtb96eJNoPm9a0C
QAAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAAC
AAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAAC
AAAGg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4A
AACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4A
AACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4AAAAAFpOggcAAACg5SR4A
Km1vinJm5Lk4MGDrf9fiZdddImS5PWvf/05jgQABnv1q1+dm265+1yHAZwnFid25crHXewZGdg5S8OnA1vxCR4AAAAAB
g5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGg5CR4AAACAlpPg
5CR4AAACAlpPgAQAAGg5CR4AAACAlpPgAQAAGi5kXMdwPnoyiuVPnchAAAawHnjfHgfLsFzDrzyla881yEAAADAeeN
AAAAALSfBAwAAANByEjwAAAAALSfBAwAAANByEjwAAAAALSfBAwAAANByEjwAAAAALSfBAwAAANByEjwAAAAALSfBAw
AAALSfBAwAAANByEjwAAAAALSfBAwAAANByl+c6AADg4Wt46r5s+5u3n+swaKHhqXuTRPth04an7kty8bkOA6C1JHgAg
nwAAAAALScBA8AAABay0nwAAAAALScBA8AAABay0nwAAAAALScBA8AAABay0nwAAAAALScBA8AAABay0nwAAAAAL
7e1e2LsHD1iilHKq1HjzXccAjhT4FDy59Ch5c+hQ8ePQnHi58RQsAAACg5SR4AAAAAFpOgofGm851APAlO0/Bg0ufggeXPg
3Nl+xbmMGx6OSimfK6V8rJTjyV6WUQ511F5ZS3llK+Uzn3ws660sp5Wc7feqjpZSnnNvo4eGnllKnlPK7pZS/KaV8spTyNfo
UmOddb/185+wFrPqrU+qdZ6sLP8miTvqrUeSPKuznKSfFOSA53/Xp7kvz/kkCLD3+uT/J9a65cl+cos3a/0KTgLtdZPde5PT0
I625/T2R9YX3ff6e1Tv1aXvD/JnlLKpeciQHg4KqXsTvKMJL+UJLXW2Vrr/dGn4MHwnCR/W2u9NfoUfDFGkmwrpYwkmUxy
x5+/IPtXR3d+A5LFJjib5lc5XiX+xlLi9+hQ8GF6c5C2dv/UpOAu11juS/Ockt2UpsXM8yU3xfoqHGQme80Ap5flJjtRabzrXscAj
Vddl6X9IfEmS7Umed06Dgj4keM4PX5vk2ILK55L8dpY+Svj6LH38dqSzz6OS3NH5+44kj06SzvbdSe59KAOGH7vO/8lJrfVll
Uvy+1I6j+X9FA8rEjzngVrrj9ZaH1VrvSJLH9N9d631HyZ5T5IXdHb7riT/q/P3jZ3ldLa/u/N/eYAkPZTtpZSdzd9JvjHjx7O67/T2
M8rBTt7PxSSnlmkh+qtT6/IPK4LH2i58lkH0nyj2qtM6WUISS/nuTJSe5L8uJa6y3nKmZ4uOnOnT/oLI4k+a1a60+UUvYmeV
zn4KzVEq5PsmLksxn6b3TP87Sb+14P8XDhgQPAAAAQMv5ihYAAABay0nwAAAAALScBA8AAABay0nwAAAAALScBA8AA
OHsj5zoAAICzVUopSf4gyZtrrS/urPvKJBcn+XRneaTWOv9Aj11r/cddi/8qyb/vHO9rkjw/yVNqrTOlIH1Jxr7I6zirGAEAGj7BA
eW9pZSDpZT/mGRb55M6v5nk0iT31FpnOue7p9b6hc5rnlZK+YtSyl+XUj5YStlZSpkopfxKKeVjpZSPfKe1dn3paWUG0sp7
cneWKWPvXziSS/3H2QWutrSimvqLU+KUIKKTuS/Hgp5dNJ/jTJW2ut/7eUMpbkrUleVGv9UCIIV5IzSV69dJj65aWUL0vyJ11
8I8lbaq0LSb7Q+TTNumqtp0opT03yd7L0CaK3llJek6VE05211g919juRJKWUr0vyhs66vymI3JqkSfC8s9Z6X1eM15ZSfqizPJ
7bXWT51NTADA+c1v8AAAbfujOodr2AISUopX5GIT9d0e0eSf1ZKGe3sc1UpZXuS9yV5Uec3ei7N0idy+pnreu3jSykHurY
```

iNjppN8Lsn/7Nn1F5NckeTDnQTKOSTflqUZuj6dpg/euS3JXw441ZuSfLSU8uEkP53kDZ3fyZlPcjJy2uts6WUF3W2bcvS7+8
ISPAAAAAAtJ8EDAAAAA0HISPAAAAAAtJ8EDAAAAA0HISPAAAAAAtJ8EDAAAAA0HISPAAAAAAtJ8EDAAAAA0HISPAAAAAAtN3Ku
AAACg5bY0wVNKeV4p5VOIIMOIInF02T5eSnlrZ/sHSilXbGU8AAAAA9EW/YbPKWU4SRvTPINSW5P8qFSyo211k907fayJM
cjR3H5/J6EjJ6dn5bB8byez8YiZGh3Nmfi5jwyM5enlm+3eO54LJ4RybWsiRkzO5aOd4aq3Zt2Miw0PJsanzLCwmR0/OZO+
ezeNpqTM3PZOT6axbqQoTKcE9Nz2TUXuvzvfadnc9HO8ezNPrZxYWcOLOQ+8/MZc+20dx7aiZ7d4wwX9ueyeGcma05
dGc8+pmeszMZ6p2blMjo1mfmeHl8PDy2U3PlIzM1+Wr/HoyZl8ye6JDJWSO09MZ+/2sezaNpTjZxaXX3NqZi47usrh/qm5
TybGRVXV04fax5fM09dG0g6aNTc3O5+Kd4zk5s9LO5hbmMzo8kntPz2bv9rHlGjvYfYXmJy0ci33n5nLBdtGMzEyllm6kF
XePZNjqcW++byv6d4zkzO59tYyOZnZ/P2Mjl8rU1fWrl+pfay7GpuVy5fyL3nl5YU+6zCwsZ62pLe7cPr9pven4+EyMjuf/MX
Lp7PKPDQ7n92Jns37ly/ib207Nz2T620u4mx0qmZuvyuU5ML/Xj0aGhHDk1k93bRnP/1Gz2TK7UXVPu956eyd7tK9dybG
5+1AXbMrew1Eb27xxPzWJkhpjadr48NBiFhZxXqC5hYVcMDmWUzMLq9r5nm0r5d4cY7ozPjX7TY4IU7PJ3SdmctmeiZS
cF0wO59b7prNrYjQzC/MZ7yrvp8119bE3rSbE9NzuXbYLCXJ0FBd1R57y318NjmZWznvzomSk9N1uSyT5MT0XCbHRTaM
cA4fnc7ubaPLbbU5VjNeN2U31anzpl00x941sdT+kpXrTfIM6kodbhtNzszlzf1ks5S5k+9hopucWI+/B3eNgdx9fPnYwK672M
5rXNmV3emYu28dXxrymjO85NZPLdm/L3OJSOXT38abNNOXRW09NOezcNpSRUnjsau19o3ntiToz2bVtLKdm57JjbKWt
MeTZ02dd3E09Txsam5Xlp76d53X+e6e+95zRjX3R+bsXZydCSLSY5Pz2bH2OpnrP07xzM0tjJFxaGuZ5SlsW/tvWk4952eW
0XHZPJc4fo/l3+did9nlqdi57JpbGyyOdOJp9m2tqrrfp28357zm1dC0jQ0M5cnl6eybHlsef5fvj4kJGhoaX22VTPs05jpxcapf
PAs042Kz/eTmXPZNjq/q28ttplNWx6dms3tybLlt9/a1E9NzuWTXRM7MLuRi13uPpbFlabxc6VNL9dGMY82Yc2xqLhfvXhC
m5ddcn27Nk28dC8kX4lVrr1hy4lK9J8m9rrX+3s/yjSVJr/Q9d+7yjs89fllJGktyVZH9dJ6iDBw9WP7L80FtrPk/N9+VH3zbX2
+6eXnba59/dX7+fYdz671n8pi92/L9z7wyP37jyvZXPftA3nrotrzn11yRCyfH8qN/8LFvr/29D9+WFx58TN52aO15b7j26rzu
eT92Z5z7h0lXXcMO3PjELKbn+D5fW9bvOG669OvPzsxkeGcvvHLot3/6Uy1eV048+78sys7CY3/zArWvO+6pnH1iO/bU9x
8PJV8TTH/t5nXLcb/t3jebPP30kV12yZ7k8Dj5md1548PJV5XP9tVfndw7dlkO3Hs/E6FBed901GR8pfeuh+zz96uM//n9fnvr
mO/9luzls+cGuOT8/le59xZX7+fYfXlMegOr7h2ifm7hNz+fEbP75q3xNTZ7Jzctvy9X7jE/etKcOm733DEy/NttGh/Ps//ptV1
9+1WMyPjyU//B//mbNtuaamvbQ216a/V78tMuX49ioH7zo4OW579TUqrY/MTqUf/dt12R0uORHfm/tWPF3D16ed33yzjz
k3/bximcdyLaxofzqn3921Th48DG715RldzvpN240/WNspKza1ttPB43JuydH8j/ed8ua8bi7T6/X1v/FNxl9FzW9LHuttTwj
xb6e+K7J/p0jeeN7/jbf/pTL83sfXn3e5nzn9a1XPOtAxkeH8sO/+9E1bftPPnHPqvvoyMjYquv/yW//8szO175t+0UHL18uj95
qm7Pm647prsnhj9/6V6tifVuf+1Z3P+ltj8Op+fe//MS69/zu+0RT1s0zyU/8vavzhWMzq/pw7316o35x/bVXZ3xkKK/5/Y+ti
5P3LmmvruP1a+99j4T9rv+QW28t06bOH7rg/2fc3vb6aD3Gb3tYL22tjN3Khs91/W2k20jX/0sS+seSa//tqrl59f1nuf0ds/ut
at3zg1nXLsN97pX7PYtdfe3W+/sCuvO8j3rG+Gvyjdfsb22Sp5RyU6314jr1W5jgeUGS59Va/3Fn+TuSfFWt9Rvd+3y8s8/t
9N32xvcczvc/68qB23/p/92Slz/jcfnZdx3ue771zvzVg9xxeXv7PL/jkV0ltH+I7/F/4jqfmm/76TcvLzb+9x3z5Mx6XhcX+19lco3v
/2m73hqxkaG89Jf+eDyvj/7kif3fe1PveAr86pO3W1UD7112q8sN6qPs2mXg/Zt4ugul0H7Diq7pn302/9Td51ctz761fEzDu
44NuoHg65lvbFuUF9/2dc9Lo+/ZGff8633ms3UQ7/x8WzG+GZ5vTps6uHKI1Zfy6BxoTn2euNGklXbeutwUJ3+ykufIqOnZ
73VvWq4f1+ucvfMdT812//KHI5X7j5EZtezPPD7338e5yba6lu/wfSFludtx+03c8NR/47LFNjT/94tiofHvbxaby6fccNWjfQW
TRwbPT98Me8zNIM+vcfY6Lmud/2bvUOpqUnfbc29b6N206+PbXTPOdu23Bz7pluPrTtOrvfeoDeOjZ7NN3qe2UwZbvR8
QzePj23mMWenOT03GLOzM5v6rzN8unZ+YHHv//03KrlQfEs1sGxNsfodrontkGwvfk9MBrWazrx3Ssz3mn5xZz/9TK+vX
Cym+1//U376Lf/RvXR46PDDhe73kGta0mzn7Xv1GZdcfTWx/rxTw9t5ijA+LuHmMGvXbQNa83xvTgVnIrW68uN7qW9ca
VD7Ac1htXuv/uPvZmx49Vr+0Zj5vX3nNyZsOy7D1v42yeAe6fmlu13G+c3Khtb+b5YVBddt8nu6/hgZTlZstfY1Nzmx5/+sW
s9e/me0bPedutr9uqi09gGM80OfuY1NzqQPqbn3bVru/frYsQ3uOWfblptxdKNxsrf8z+Y90pr3AF9EGW70PqO7Lgc9P9
46P9F3ffGBsvW3rbZ8YHcpQyZpt28ZGNnXeZnnp+/L9j79n++jyMXr/7d6/iaPftu5jNLZP9l+td7kp637XmIzWj+nCPuedGB
tbdmP9r39QX2zKZb366FfHveW+fj6e9YPaVIOHg65/vTLrjmdQfQxql4Pi7h5jBr12UBmuV5f9Yt7Mta1Xlxtdy3pj3aD6qHV
3ODHZ57652bpt2mHveNzU3f6d4xuWZb/zrhf7ev1zz+ToquV+fXyjr2ZMh1UI0159Cv/jcpy0D223/knRpfu+vtu5k41i2Hr
7PVv1H82es7dbH/dzHWfzTF6lwe17QsmRwdu677nr3et/frYhRvcc862LTfj6EbjZL/y30zb7nes5fcAX0QZbvQ+o7suB40XS
X/DddfkzX9xS1717AO1r/6OQfy+x++PX/413fkct+/etW21z7/6vzRR5fyf3/413fkhtbXb3/Vsw/kjz56R/7I3318Hrt3+5rX/tp
/5/s+uuYap6bm89ltX1vW7zhuuvTqH7zqW66+9Om/+i1vWlNOFk2P5wW+4qu95X/XsA/mNTuy9x/zN9392ef/fu+n2vPo5
bnn/3bdesqo+N2uWga/l337YUx3p127StG3rqtI99djd77/qjJ06tOfZrv/Xq/OL7/nb5GvqVx6A6np6byw3XXrNm349+/p5V
jvXd1n3Mfu2l2a87jo36wauefWBN258YXfoNni/d33+su/7aq/Mb7//swPb6P973t6vGmn7jwKB28aUX7cjrrrtmYJ01x1ivjW
eXTIOvmymGjYzbl0N0uu8ej9cbz7ja9MnavHo+7+8kffPi2PvFdk+m5ueXz9Z53vbG4u11cedGOvm27e/nwXcfWXP/j9m8
9TM3Or2nS/e3739Tdl3cQ7uzC/pg8/OH5x/bVX53H7tq+JeTPxNM91m3kW6j3W6667JpfuGl/3tb3133v+phy+dP/2vu1yM
+z+Tdzy/r1X1v/2hibu79/V7T+yzWnG8z7aC5v1+8a3zdcBk3/NeLY6Nxxqhk3+z+bLL032KgMB92be5/Frr/26lx+wXCfMf6a
OhftXDuLVrO+mUXr2NTs0owfp2Zy8c6jXDA5mrtPzGR0uCz/4vns/GLGR4czMz+/alaUC7cP577TK7PppNbs7cyidf/UbOY
8uvHTSL1tTsQibHhJ03sJh9A2bRamZBaWZSqVnMUGdmk/GRnl+Gn1ya7eD+6dIVs1Y1Mzc0v8y/MivM6lm0mpk8hvwM
OMjsbmmpRzrbpBNuzyVm0mtkOemfRONpyJpfunkjNyixavbO4NbOobWoWrU7MzSxavTPT9c4AdP/UXL50wCxayzNd
LpmZ6ZtGaGM3o8NpZtjq6a8r9vtMzubBrFq3u/S7cPpaRsjKLvjOG9M6i1fSPzcyitVxGnTpt6qW3r3cvN7NoNTNp9M6idd
I+OAWbRWZvEZPltW0x7XzqK19JreWbROTi/NTjNUVmbRal7bzHjVjP0To8l0n1m0jpyYzD0ZtE6OT23NPtMz3jQtl/mXtcs
9i7JoZzYmZueRaki3atlENTtpOjydRc1txPFutCjkdHMzO/dhat5T6+bSgnz3TNotWZHadh92zzDXn651Fq5k9qBkfmjHg3l
NoLT1f9c68c6RzLd2zaDVtqPee3/v80JTt8eW2NpTRkeTk9OjYHL0znzXnbdrS8ixaj2dyyc7xjl6szKLV3Gua8uidoa2Jp3t2t0
e8yywe9tYak3uOTWbPZOjmv2YXzWzbe8sWs2zUINOx6dms3/nROa7ZtHqnk1w1zqzaDxt8/TsXHZNjGUog2fRasaBpm9
i1Z3PTWzaN11YiZ7JldmJGti7R2vm+2nZuZy4eR45hcXOzMNrZOLVs/MX/1m0TraM1tY05dXxrqlNt+sn5qZy+SAWbR6x/K
yu/ogUAAADAQ0CCBwAAAKDIJHgAAAAAWk6CBwAAAKDIJHgAAAAAWk6CBwAAAKDIJHgAAAAAWk6CBwAAAKDIJHgAA
DIJHgAAAAAWk6CBwAAAKDIJHgAAAAAWk6CBwAAAKDIJHgAAAAAWk6CBwAAAKDIJHgAAAAAWk6CBwAAAKDIJHgAA
3qLW+p9Y61Vl8f5JHbWE8AAAAA9IW5nguSzJ57uWb++sG+RISf6434ZSystLKYdKKYEOHj36lIYIAAAA0H4Pix9ZLqX8oyQ
J3g+IORAKeWxpZSxjC9OcmP3DqWUjyf5hSTX1lqPbGesaAAAAA9YW5bgqbXOJ3lFknck+WSst9Vaby6l3FBKubaz239KsiPj

[illegible]

zxt68j9G3Ygl89brx/u0Xqxzce2Xqzx2Shozkb+sXSuYt1ydbO8FB9hevHPwxZ/a3q68Kctu612un/DDtcvgznaxMOtnjXJZh/v
JNud9W04PtOj/lx9+nYyvML7w2teW+ebPq8ZULjq/Q6a8tOc7kjXhUsDZk3KD12x97tz21LXTd9Pq2de3T/tmYilZcN9t0arN
vxKz6RPG8PHLbdFl+/dBttZm5Dz/SOJS/wfMxAF9E6UeWr1NKnVmtz6n8GzxA9Spae4bTmN14cKpohfXIP8b8ern/HO9x7
OKsBG3cPY1RXONg/mkVHUwrT651qC/FoBKMZp0qSqZih3kYvWyugEQ86o7f1dE84SpaEFRUM/Pax1tFa+9IGjPrS1W0
7tVLgbGcphRn0AyljigqYDSdRzwWwcB4DjOmJXCI51f358ychkINVbSMXQQFFTERagwOJ5Da0MSBU8VLTpnnK7IYKpomf
4gY2QzFYeMHsayebQ3JTGULrjVwUyFrcAqWp4KP6aKvJQsXxJWqfowu6yyjIP9ZVxXUHGraOI5jefyaKlPYjxbWUXLVF0xuv
YWbUufxUto2tj02w+j0Qshv6RDI5tdiof7RxM67nmEYvG3Oooxj/MuEamsWwOTakEliLoH82gOVWqfmlqphi9+6tomSog
EQH2jWXRICr5tKkkY85xq3jp8ZUqQHTfbY1OhZvduoqX0aWxg+nLX0XLxOces1bEKqtomUoy/rw5oSpavspwE6qi5TvX5B
yuqaOmQXaaio7+K1mg2h9b6pK4caa8SZWQxOjbtJrYHxnJob0oi49Glv4qWGRCoAohUVJkaTjtrEVDKD8Zn/RXy/H5p2vtH
Xs4q+itWfYybUFVVIYf8R4RRUtLX9ZFa1sriyWGpKCKUIX0XLsbmxi5mHypKmiZWSBUphZH8eBsTwG9DH+Klpm238N4P
rtU5DMo6lAIWiQs5jF+99ibWK1mgGs+qTmNUQRf9lqSqqaJldAwA7c1OFS1T8bflRaKVK6uGOqDzkfe6tq0phdINCYYk8
TxWtsnsEXxWtwfGsWznTL7epyjXNK+wfaK9OYklnlpo7c0pHN/aeERW0XrXf2RZRG4HcC6AWQB2A1gNIA4ASqmf6TLpP
gghhBBCCCHksclPeAghhBBCCCGEEEEKmOO/ojyy/E4hIP4Dth3oeRyCzAOw91JMg5AiCMUXlwYUxRcjBhTFFyMGD8UTeb
QQMsXhAx5iuOFQT4CQlwzGFCEHF8YUIQcXxhQhBw/GEzks4G/wEIIIIYQQQgghhExx+AYPIYQQQgghhBBCyBSHD3gIIY
ir+rPGbpdROQ6HVMbROS0Qzt7Qg4/RGS6iNwpli+LyCYR+QBjipDJISLH6/XJ/BsSkb9jTBEyeUTkSyKyUUREfJHbRSTF+yl
CeCn7/pMCTn8+SGA3ymITgBwCpz1ijFFyCRQSm3W69NiAKcDGANwNxxThEwKETkWwCoAS5RS3QCIAc4D76fIYQYf8B
sCYluRgsAzAa0qp7WBMEfj2iAGoE5EYgGkAdoL3U+Qwggw94jh5+AOCrAlp6uwXAgFlqr7ffBHCs/n4sgDcAQO8f1McTQk
A82BkEsB68nyKHGXzAcxQgIssB7FFKrT/UcyHkCOJspdrpcF5r/4KInOPdqZRSb4CEUKqEwNwGoCfKqVOBTCK0p+OAG
fAeA4AND7mwHsezcTmJhvjv6fHCil9sD5XYMzAew2r7Trzz36cDemNN54I4Q4/+v5pILqab19J5wHPowpQt4eHwHwrFJq
f0T/Lw8hBICI1Itlo/kO4AIAL6I8dvw9WldpeT9AAY9r8gTctSjInoF4A0ROV43LQpWEhhThLxdPoHsn2cBjClCksfgPeLyDT
tabMQD/Vyn1TyLSAuAOAHMAbAdwiVJqv74Q+DGCv3nHAFyulOo9BFMn5LBFRBbDKQSQALAVwOVw/iOKMUXIJND/Ad
IIYQQQqY4fMBDCCGEEIIYQQMsXhAx5CCCGEEIIYSQKQ4f8BBCCCGEEIIYRMcfiAhxBCCCFtGHfP5HfiMhrIJeRH4
02ykA2gC8ordjSqn8RptWSv21Z/MbAP5Z9/cBAMsBnKaUyojILACJtynHpOZICCGEEGLgGzyEEIImcqcByCnlPqZaVBKvQ
1Fv6nDNE5EkReUFE/iwijSKSEpFfiMhfROQ5ETIPH/tZEblXRB4B8LCI1lvzf850RkxbujQkIIYQcCfANHkIIYRMZboBrA/Yd
E4p9f9EJAFgHYBLIVLPiEgTgHEAVzdqJNE5AQAD3j+hOw0ACcrpfaLyD8DeEQp9TkRmQ7gzyLykFjq9OCoihBCCCFHMnz
AQkbMB/Ei3vSwi2wGYBzwPKqX2e+bYlyf0dspAHMAbjqc+IQQQgg5muADHkIIYRMZTYCuDhgn/fNFwFwVLq994DROS
i+piFIIP4HEAL+rf6OmA80aOjZzn3ONFpNOzbzGA7QA2A+gQkTP0cY0iEgPwBwCfNOPCesVh9hDn9wCu0j8cDRE5tVYIE
AN8D8CP9Ozl5AFsAXKmUyorlpXpfHZzf3zkfWE8A/FS/6ZMH8Fldfcs/xjcB/ECPEwGwDU61LkIIYSQqohS6IDPgRBCCCGE
jh8wEMIIYQQQgghhBAyxeEDHkIIYQQQgghhJApDh/wEIIYQQQgghhExx/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])})\n")"

]

},

{

"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": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLLAAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA  
XRwbG90bGliLm9yZy+WH4yJAAAgAELEQVR4nO3df7DndX0f+ueLXXAR+VFgu5Bd4uKAenFV1HOPVmutqRk0XuA21EKaj  
/vee95PGaY8/382Nc+53De7HeffD6fb3V3AAAAABjXYfMOAAAAAMDBUfAAAAAADE7BAwAAADA4BQ8AAADA4BQ8AA  
AAINT8AAAAAAMTsEDAAAAAMDgFDwAAAMDgFDwAAAAAg1PwAAAAAAxOwQMAAAAwOAUPAAAAAwOAUPAAAAACDU  
8AAwNdu2bcsNN96S77zw+HIH4Tla89B9SZKdj3jLwOyseeieeUcAgOF4twbAVH3nhcfn4Ze/Y94xel6O/No1SeLfiTP1xM8  
wAAAGBwCh4AAACAwSl4AAAAAAan4AEAAAAAYnIIHAAAAAYHAKHgAAAIIDBKXgAAAAABqfgAQAABicggcAAABgcAoeAAA  
fgAQAABicggcAAABgcAoeAAAAAgMEpeAAAAAAAgp+CZg8svvzyXX375vGMAAADaqrAa/h6+dt4BVqNt27bNOWIAAACsG  
AGJyCBwAAAGBwCh4AAACAwSl4AAAAAAan4AEAAAAAYnIIHAAAAAYHAKHgAAAIIDBKXgAAAAABqfgAQAABicggcAAABgc  
AAAAABqfgAQAABicggcAAABgcAoeAAAAAgMEpeAAAAAAAgT3beAVajHTt25OGHH84lI1wy7ygAU7Vt27Yc9mjPOwYwmM  
LnsssvmnARgui655Jjsve3OeccABvP4umNy2ks2eK8EwPNmNVwVOsQVPAAAAADsm4IHAAAAAYHAKHgAAAIIDBKXgAAAA  
IHAAAAAYHAKHgAAAIIDBKXgAAAAABqfgAQAABicggcAAABgcAoeAAAAAgMEpeAAAAAAAgp+ABAAAAGJyCBwAAAGBwCh  
Gt3beAVaj0047bd4RAAAAYNVYDX8PV/DMwcUXxzvCAAAALBqrla/h7tFCwAAAGBwCh4AAACAwSl4AAAAAAan4AEAAA  
Ch4AAACAwSl4AAAAAAan4AEAAAAAYnIIHAAAAAYHAKHgAAAIIDBKXgAAAAABqfgAQAABicggcAAABgcAoeAAAAAgMEpeA  
u2aecfgOVrz0N1J4t8hM7XmoXuSbjh3DAAYioHhGkK57bTT5h2Bg7Rjx2Njko0b/WWbWdrgvx8A8CwpeACYmosvvnjeEQ  
AAgMEpeAAAAAAAgp+ABAAAAGJyCBwAAAGBwCh4AAACAwSl4AAAAAAan4AEAAAAAYHX3vDM8K1W1K8k35p1jIToxyZ  
gFDw8G1fMOWBgHclKYB3C/FmHMH/W4QrjGTwAAAAAg3MFDwAAAMDgFDwAAAAAg1Pw8DRVdUpVXVdVN1fVTVV1y  
qKrFyT7vS2GGquq4qvp4VX2tqm6pqjdYhyuPgoe9eSjZT3f3GUlen+QnquqMJO9L8rnuPj3J5ybbwHQ8kuSt3f3qJGcmObu  
9HdX5y8fBLi3djknOTfGxy2seSnDefhHDO6yV/Ptk8fPJ3lrko9P9luHMGVvtSnjDyT5yGS7Yh3CSuB9KcxIVR2b5M1JfiJu  
NLSX42yeOT7RNIhCksdZLfr6qtVXXRZj/3pTA7pybZleTfTG5Z/khVHRXrcMVR8LBPVfWijj918pPdff/yY93dWfrDFpiS7v5Od  
4Mih42KuqOjxL5c6/6+7/MNI9Z1WdPDI+cpauKgCmbHlJ7HVJ3pDkuKpaOzm0KcmOuQWDQ98bk5xTVbcnuTjLt2ZdFus  
1IfvcZPXRYZ5W5aeh3VdkvMnp1mHMEXd/XPdvam7Nye5IMkfdPffjnUIM1NVR1XV0U+8TvL9SW6M96UwM929M8k3q+  
7J0jChXsb6U5le7+5H5JYXVoarekuRnuvud1iHMzmS9fXKyUtbJb3X3B6rqhHhfcjNTVWdm6QMhJkhyW5Ify+Q9aqzDFUP  
D4bhcm+e+TrwAAQ1DwAABMVNWlkrwpyXuSXDdZd1hV/UpVfa2qrq2qa6rq/Mmx11XVf62qrVX12ao6eY7xAYBVTME  
p7seT7Kyq6ybHX5ZkS5Jrqpyj1iS5Y7ZxAQCWKHgAAJjU1fFj3prklVXVWSpsOskn9/VLktzU3W+YUUQAGH1yixYAwJLzk/xG  
X0xyX3c/mqVS6INV9eUkNyT5y7OLCwDwlOrueWcAAfJRqupF3f3nVXVCks8neWN375x3LgCAJ3gGDwDA/n26qo5LckSS  
Q8AAADA4BQ8AAAAAINbO+8Az9ajJ57YmzdvnncMAAAAgJnbunXrn3b3+j33D1fwbN68OYUli/OOAQAAADBzVfWNve13  
q6u29edtp7kvxZd59WVRck+WCSvzWtTCvBvQ/vzq07H8yd9z+SDce8IC896agcd+S6FTt31Nkyjz9b5tnMHjHzNGfLPP5sm  
uyvuvvjG7v/141h1+WC49Z0u+f8v6g/rhmtbcUWfLPP5smWWex2yZx58t8/izZR5/tszjz5Z5/Nkyz272SjPNW7Q2jvnmsu3t  
f7bM48+WeXazV5ohHrjCVRdV1WjVLe7atWvecZ6zO+9/5MkfqiFs/vbjufP+R1bk3FFnyzz+bJlnM3vEzNOclFp4s2Uef7bM4  
V93+GHZcMwLVuTcUWfLPP5smWcze8TM05wt8/izZR5/tszjz5Z5/Nkyjz9b5tnNXmmmWfB8lcnPvXVqVR2R5IlkV+9xztVJ  
l2s1eammafUIxvSPJLsdYk+Wh3f6CqLk2y2N1XV9W6JL+R5DVJ7klywRMPZd6XhYWFxlxcnFrmfPE8dnMlnn82TLPZvalm  
MTsEDAAAAAMDgFDwAAAMDgFDwAAAAAg1PwAAAAAAxOwQMAAAAwOAUPAAAAAwOAUPAAAAACDU/AAAAAADE7BA  
AACDU/AAAAAADG6qBU9VnV1VX6+qbVX1vr0c/6mqurmqljVn6uqF08zDwAAAMChaGoFT1WtSfLhJG9PckaSC6vqjD1C  
Wx7+2Tfvrwnye/t7UBVXVRvi1W1uGvXrucXlgAAAMD4VsRDIqvqh5MsjPnnezve3Vd090j3L6xfv3624QAAAABWuLVTnL0  
CAQ9bUCp7ufizJe5N8NsktSa7q7puq6tKqOmdy2j9P8q1kv1NVN1TV1fsYBwAAAMA+TPMWRXT3NUmu2WPf+5e9/uvT/P  
OwQMAAAAwOAUPAAAAAwOAUPAAAAACDU/AAAAAADE7BAwAAADA4BQ8AAADA4BQ8AAAAAINT8AAAAAAMTsEDAA  
p6rWjPlwkrnOSPjhVv1xl7OOzrJJUmun1YWAAAAGEPZNK/gOSvJtu6+rbsfTXJlknP3ct4vjvlgkt1TzAIAAABwyJpmwbMxyT  
z/9cAAAAAADmWbBsyPJKcu2N032PeHojFuS/Jequj3j65Nc7UHLAAAAAM/O2mc6WfU/9UzHu/tDz3D4C0lOr6pTs1TsXJ  
uq6tli9199TP9egAAAAOTHX3/k+q+sMkP9DdD0y2j07yn7r7zVPO9zQLCwu9uOgiHwAAAGD1qaqt3f20x9sc6DN4NiR  
2H5P+wiT3d/dlSbZPPh0LAAAAAgDk7oIKnqn4hyT9l8nOTXYcn+c1phQIAAADgwB3oFTz/Z5JzkjyYJN39rTz1EeoAAAAAzNG  
u/vaqSYDAAAA4IAcUMFTVR/s7n+Q5Nq97AMAAABgig70Fq237WXf25/PIAAAAA8N894BU9V/d0kfy/J56rqK8sOHZ3kf0  
qST1LjJneQIU8oFAAAAwAF6xoKnu985+XrbOIAAAAA8Gwd0KdoVdV79theU1W/MJ1IAAAADwbB/ox6d9XVddU1clVtS  
DAHOGtWv8xyf/V3X8nyV9N8sdjvJC1VAAAAAcsAO6RSvJWd19f5J0dyf5F1X1H6cXCwAAAIAD9YxX8FTVzyZjd99fVX9zj8M  
sqjP2OOf0ydw3dvcrkvzkgYQAAAA4Cn7K3hqH6/3tr2ns5Js6+7buwvRJFcmOXePc348yYe7+8+SpLvv2s9MAAAAAAPawv4  
jpy8zmR73fP0+5+e5C1JNiX5w6p6Zffu/yk7r4iyRVJsrCwsL9iCQAAAGBVecaCp7vXHMTsHUIOWba9abvue1Jru/ubyf5k6  
yV5b5LPJrklyVXdVNVXvpV50xO+2ySu6vq5iTXJfn73X33tDIBAAAAHlqqe6xH2iwsLPTi4uK8YwAAAAADMxFvt7e6FPfDP8x  
AMTsEDAAAAAMDgFDwAAAMDgFDwAAAAAg1PwAAAAAAxOwQMAAAAwOAUPAAAAAwOAUPAAAAACDU/AAAAAADE7BA  
kb09yRpILq+qMPU77+SRXdfdrklyQ5FemlQcAAADgUDXNK3jOSrKtu2/r7keTXJnk3D3O6STHTF4fm+RbU8wDAAAAcEiaZ  
9xXdvdDdC+vXr595SAAAIcVbJoFz44kpyzb3jTz9x7klyVJN39P5OsS3LiFDMBAAAAHHKmwFb8lcnPvXVqVR2RpYcoX73  
gEAAAacitZoc3h3X5OIhycv3/f+Za9vTvLGaWYAAAAAONTN+yHLAAAAABwbKBQ8AAADA4BQ8AAAAAINT8AAAAAAMTsE  
DA4BQ8AAAAAINT8AAAAAAMTsEDAAAAAMDgFDwAAAMDgFDwAAAAAg1PwAAAAAAxOwQMAAAAwOAUPAAAAAwODW  
l8wIH5bDD6nllPL3MSfLoo9/JV751X3bevzsnH7Mur/yeY3PEEWsOeu69D+/OrTsfzj33P5INx7wgLz3pqBx35LrnIXHy0MO  
Lyf59X0cf3uS0yf//KUKvzr5esh67LHH86kv78jPf+rG7P7241I3+GH5p+dyXmv3nhQhcnjj3c+c9PO/NRVNzw590PvOjNnv+  
U+b/c4tGw6qMBkx8zRzT/P7lFPYmac5W+bxZ8s8/myZx58t8/izZR5/tsyzm73STO0Wre7+wyT3PMMp5yb59V7yR0mOq6
```

```
nWwc2+deeDTy7ej+defWNU3Xnw348bdz6w19k37nxg1WVOppd7mt8PmcfOPM3ZMo8/W+bxZ8s8/myZx58t8/izZZ7d7
uO+8/2O/HI/uY+8hBzZ3m7BEzT3O2zLOZPWLmac6WefzZMo8/W+bxZ8s8/myZx58t8+xmrzRDPGS5u6/o7oXuXli/fv284
bGaPmHmas2Uef7bM48+WefzZMo8/W+bxZ8s8u9krzTwLnh1JTlm2vWmy75D1ipOPyT89b8uTP1zrDI96ns0rTj72oOZu
enoVZc5mV7uaX4/ZB478zRnyzz+bJnHny3z+LNIHn+2zOPPlnl2s1ea6u7pDa/anOTT+/gUrR9I8t4sfYrWX0ryr7r7rP3NXFH
nTnD1i5mnOlnn82TKPP1vm8WfLPP5smcefLfPsZs9DVW3t7oWn7Z9WwVNV/z7JW5KcmOTQJL+Q5PAk6e7/d/lx6b+c5C
tSvKNeedYpU5M8qfzDgGrnHUI82cdwvxZhzb/1uH8vLi71++5c7iCh/mpqsW9PcgJmB3rEOBPoOT5sw5h/qzDlcctWgAAA
VXVzVd1UVZdM9h9fvddW1R9Pvv6FeWeFQ1VVrauqz1fVlyfr8J9M9p9aVddX1baq+u2qOmLeWeFQV1VrqupLVfXpybZ1
g1MxyNJ3trdr05yZpKzq+r1ST6Y5F9292Ij/izJe+aYEVaLS5LcsmzbOoTZ+2vdfWZ3L0y2vS+F2bosyWe6++VJXp2lPxetw4JM
hTVIWbkvxAko9MtivWlawE3pfCjFTVsUnenOTXkqS7H+3ue2MdrjgKHp5RVW1O8pok1yfZ0N13TA7tTLJhTrFgVZjcFnJDkr
XHAUP+1RVL0ryiSQ/2d33Lz/W3Z2IP2yBKenu73T3mUk2JTkrycvnHAIWlap6Z5K7unvrvLPAKvem7n5tkrdn6dEBb15+0P
Q5uS7JhbMDj0vTHJOVV1e5Irs3Rr1mWxDmGmunvH5OtdST6Zpf/p4X0pzM72JNu7+/rj9sezVPhYhyuMgoenmTxf4NeS3
XVUVR39xOsk35/kxnhfCjPT3TuTfLOqXjbZ9X1Jbo51uOLU0pVU8JSqelOS/5bkq3nqmqP/MEvP4bkqyfcM+UaSd3X3PXM
6I96UwM1V1ZpY+cOCIJLcl+bFM3qPGOlwxFDwAAAAAag3OLFGAAAMDgFDwAAAAAag1PwAAAAAAxOwQMAAAAwOAUPA
fHHtdVf3XqtpaVZ+tpqPnGB8AWMUUPAAATzk3yWe6+9Ykd1fV65L8JSSbk5yR5EeSvCFJqurwJJcnOb+7X5fko0k+MI/QAA
wqaTfHJfvyTJTd39hhIFBADYJ7doAQASOT/Jb3T3i7t7c3efkuRPktyT5Acnz+LZkOQtk/O/nmR9VT15y1ZVvWlewQEAFDwAA
Jl3dvfOeecCAHiCZ/AAAOzfp6vquCRHJPIF5Q4AsNK4ggcAAABgcj7BAwAAADA4BQ8AAADA4BQ8AAAAAINT8AAAAAAMT
"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": "iVBORw0KGgoAAAANSUHEUgAABHgAAAGoCAYAAAAA99FLLAAAABHNCSVQICAgIfAhkiAAAAAlwSFlZA
XRwbG90bGliLm9yZy+WH4yJAAAgAEIEQVR4nOzde5wcV3nn+/TXx2fi6TRWJbli2xLAiyZcFEcQrJAfMPws2QnEGJl1mwCI
RK+rxfl72mu+pU1VNV55yqfITdx5xzAgAAAAAAQHhF+h0AAAAAAAAXhgSPAAAAAAACFHggcAAAAAACDkSPAAAAAA
7203nK1oAAAAAAAHR4IHAAAAAAAG5EjwAAAAAAAABWjHgAAAAAAAGJawQMAAAAAAABByJHgAAAAAAABCjgQAAAA
AELO63cAp6JPfOITGhsb63cYAlA+2LdvnyRp1apVfy4EvbRmzRpdd911/Q4DAACcQkjw9MHY2ji+/+hjqqSX9TsUAMaii+Yr
QAAAAAACDkSPAAAAAAACFHggcAAAAAACDkSPAAAAAAACEHAkeAAAAAACAKCPBAwAAAAAAEHikeAAAAAAAEKO
AAAAAQo4EDwAAAAAAQMIR4AEAAAAAAAG5EjwAAAAAAAHR4IHAAAAAAAG5EjwAAAAAAAABWjHgAAAAAAAGJawQ"

```


AAACakCPBAwAAAAAAEHJevwM4Fe3bt0+RfK7fYQAAAAAAcEr4xCc+IUm67rrr+hxJ75Dg6YPZ2VIZtdTvMAAAAAAOCW
DwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgAjwQMAAAAAABBy
wAAAAAAEHikeAAAAAAAEKOBa8AAAAAAEDikeABAAAAAAAIORi8AAAAAAAIUeCBwAAAAAAAIORi8AAAAAAAIQcCR4
AExsbGIEql+h1GT4XiCR4ze4+Z7TSznePj4/0OBwAAAAAA4IQSiid4nHOfkvQpSdq4caPrczgAAAAAdVZNDWnPeCt122239D
AACFHHggcAAAAAACDkSPAAAAAAACEHakeAAAAAACakCPBAwAAAAAAEHikeAAAAAAAEKOBa8AAAAAAEDikeABAAAA
QMiR4AEAAAAAAAg5EjwAAAAAAAhR4IHAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgAjwQMAAAAAABByJHgAAAAAA
DglrFmzpt8h9Bwjnj5YtWqVDhSe73cYAAAAAACcEq677rp+h9BzfEULAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgAjwQ
keAAAAAACakCPBAwAAAAAAEHikeAAAAAAAEKOBa8AAAAAAEDikeABAAAAAAAIORi8AAAAAAAIUeCBwAAAAAAAIOR
HAAAAAAAg5EjwAAAAAAAhBwjHgAAAAAAgAjwQMAAAAAABByJHgAAAAAAABCjgQPAAAAABayHn9DuBUFc1NKPW
AQAAAAAAnET4DR4AAAAAAICQI8EDAAAAAAQciR4AAAAAAQo4EDwAAAAAAQMiR4AEAAAAAAAg5EjwAAAAAAAh
AAAAAAACFHHggcAAAAAACDkzDnX7xiOi5mNS9rb7zheBMslHep3EMAJhnYBtKjDAK1oF0Ar2gXQ6mRtF+c450abJ4Yuw
0ArWgXQCvaBdDqlGoX/AYPAAAAABayPEEDwAAAAAAQMiR4AEAAAAAAAg5EjyLzMyuMLPHzWzMzG7odzXAL5jZ02b2
2gBNBh3Zxk5ntC64Z3zezNzfM+/2gjj9uZm9smN72fsrMzjWz7wTTP2dm8WB6Ing/FsxvTh7DMzPzM4ys2+a2Q/NbjeZb
+qTk33BI2irppyRdJGlRw03HjyX9esNyV/R+d4Dj9ldqrZuL0Q46bQM4EfyV2vfZHW+uGa9wzn1Zkoj7pGskRQ+W+TMzi85z
Kekq/ocE7BYrpJ0R/D6DkIXN0z/rPN9W9ISM1sp6Y2S7nHOTTjnjk6R9IVwbwh59y3nf8r8Z9tWBdwwnDO3S9pomnyYrSD
sGTmFd2kUnXDNegmexrZL0bMP759S9kgJh5SR93cweMrP3BNNWOO2F2B68PSFoRvO7ULRpNf67NdCAMFqMddNoGcC
/Hrm+RAacIBajHdDWEBKfHS+pFdI2i/pv/U3HKA/zGxA0j9l+m3n3FTjPK4ZOFW1aRdcM7ogwbO49kk6q+H9mcE04KTin
s65g/nBQHjghmFIM/ofYv3XO/e9gMtcMnNLatQuuGd2R4FlcD0paG/xad1z+j0Bt73NMwlvKzDjMNIh7LelySY/Kr+u10Rze
Nfla4LRTM6V/8Ow/6oO91PB0wfflPTWYPnmNIzrF2+V9I2gPNB3QT/+0t6zDI3a8MSrhk4ZXVqF1wzurOQxHnSMH8Yt/8
7inPufwfSN8kdiSU6iqTrwtLp4tRhZn8v6Q2Slkt6Xv7Ij9Uj9tBp7bW8x0GFqBDu3iD/EftnaSnjf1G7TdBzOwPjf2a/NFUfts
zcErq0i7eLq4ZHZHgAQAAAAACDm+ogUAAAAABByJHgAAAAAAABCjgQPAAAAABayJHgAQAAAAACDkSPAAAAAA
qhmSdESSzGzAzO4Lnup5xMyuai7cqYyZrTazx8zs02a2y8y+bmapYN4aM7u34Ymh84Ppv2tmD5rZw2Z28yLuMwAAOM
NI3JP1X59wXzCwp/z/VflBSWyX9hiSTtF3SHzn7l/MYwEAAE5OfEULAACcChq/ovXTkj5rZhvkj1r+i5m9TlJV0ipJkYqdaFi2U
05+Ptq51zJz6W/5RPo1/uUqbQUK4iKdVl0ybp0865v3jhewEAADAXv8EDAABOKWb2UklRSYclDU56GCRufk7+166aLaRM
/arPsQso0+/eS/sLMtkkqSfpF59zXzexlkv7FzCRpRtKvSDr4QnYMAABA4keWAQAAAAAAQo+vaEAAAAAAAIQcCR4AAAAAA
AAAAAhBwjHgAAAAAAgAjwQMAAAAAABByXr8DOF7Lly93q1ev7ncYAAAAAAAI+6hhx465JwbbZ4eugTP6tWrtXPnz6H
JyMevYbPGYWX57pMskPSfpQTPb7pz7YUOxd0k64pxbY2bXSPqYpF/qVUwngqOzeT1xlKvnpwpaMZTQutMzWpJK9jusF
TNV5SIRTSZL2okk9T6IUPyvEhL+VyxohVDSZ05nNIPD0xp39FZjQzEIY7ms6XNTqQ0GyprCOzjeVLFZ02mFasGtF0vjwnnr
laWFsIOuWNbyTEIV55SMRVWuVHVopqh0PKqRgZhikajGZwpKxz1Vnb//R2dLkpxi0agOTuU0OpTQklRUhZJTrlRRqeKUL
EM9AlqpcsaKZQlnDqZjiwbxkLKrhpKdSVXp+Kq/IgwkVy2WZlkrETLGl1zS8ZiO5opKxqlaSHgqVSsyRTQ+XdBosP0ft+Y1O
OudKZWViEZ1KFjXUMLTUMp0aKaiA1MFnTaY0FAqqnyxqtLSRTOFigaTnvLlspamYipWpEMzBS0fSGi2WFbci2q2VFYyFlXU
tKKBOP6tmJnEYGEqqqlNnRHZop6PQHvy/dP5XX6EBcAwlP04WyDs0UNZKJkx2PKh41zRQrmi1W5jxTOu7pcLaopemYf6
BoH/MhplxFSPVtC2WtSwTu8VVF194milWJFUVj3o6NOPXi1KlrLjnqViuKlsoa2kmpohFdGi6oHTCUzzq9yFxL6J82d+vdMx
LMvMNJDwlEIENFusarrgt5HRgbjKVafj2ZKWZWlyZzqaL2koGdNUvqTBRExTsyUNp2PKFUtKx2MqVcoaiMc0W67Wz1kml
X/ekZzfx+VLZaUTnlxVmsr7bX2mUjIXiWgw6WkgHtGR2YqmC6XgOJe0NB1XtlhSNBJVjh5R3ItqOI/W4WxRpwf90kyhplTn
J03m8rlwMxTecrej5oL8vSUR3JVXQkv9KydExHckWl4568iEkmbJ2I8uWqpmZLOmckpanZY+seTHiazBc1kdjWXkcHEipW
LGgnqz0TQ1pKxqA7nilqSismLOJWrVu87yxv/G0dnSxpKecqXyviUuuUjUiwakcnvjyrOqVj2+/wVgwnJqpLzrzFLUn7/PjyOq
jEeV9Pzr9kS2pNHBuGZLZQ0n48qXqzoc9M9LUP6m8uXguuMpHvX7ulTMUyxqGgrm1/oXp6piFtVMsaxc0b/Ol6pVFUoV
WjOb/OTs35dKZSr9fM7MuBpMlFRRK4U1PGSUrGoBhNRFSpO49N+uXTMb09LMjHNFisany5qOO1pIO737zPBvKj55ykT
zf3kYH44pFI/XzLSuWffciSnn+NfaMpUIVKv79x1DSC/q7ikYyi73gLV7xLOXpvXskZwOzuRVKjvliHUtSceUjkd1JfDuPBvay
xrHg82pfPDWH4rAI06+WPLF8kacw5t0eSzOxOSVdJkzWXCXppuD15yX9DzMz55zrYVx9c3Q2r68/Oq4btz+qfKmqZCyib
xrP1uZ3P6Jc2nq0//cbuevn3X7xWn9v5Q1138VptwAM3fv4wXr5WrmN5wzrbT95jm780rE6tnXTev39d/ZqMI/Sf3z9Gt28
bxt83pFlv6Hxtvua96XZ3TNT56tz/7LXh3JfbV103r9+T+Nae/hWSVjEX30Fy7U+HRBX9+1X2951dm6+e5jcX/4qg0ycy3rvXr
WbY9965Xr9w3ef0SUUvO10jmZiyxao+9tXv1Zf9gze9VKm4pw81nKfm9f39d/bqiYmz+uO3Xqh8ybWUXZLYvK04ff6h5/TW
fTmj/Oal+9KVHO56DYjmn//KVH9WnfewtF2oo5Wnv4ZxubNjGBY5dp1QsMqfszZvX66uP7Nd0oarF3Hj2nJhgx3HVkkTLv
Wq9nj8zW61YyFtEfveVCHZiabqjr363P++jPX6ijXFZ//LXHtTQd16/+zOo529lyyVqNZGJyMv3Zt8Za2uAHLI2nv/vXvftN6zx
G+6cKXufLBzP1E7Vr/2M+fpMynCnHa2ddN6Dac8/fm3ntRkvqQPXLpO+yendNt97c9bYzua06aufjmyxar++ttPdZ3wv3/FS
J/DC6/YLkufdnKOXVo2+YNUv1bu7X38Gzbvmnb5vW6/Vt+f1Rb/vZvjekdF52jLfc+OXI+JskSMT35/JQMz5e29DXL0jEVmtp
5Hu32v/FcZOJRLQ+S5hO5cstxumvnM7rsgpVzrnONMd5y9QZ94hu75/Ttf37/rvr75n6idk2Le9b1mrHlkrU6fTipUrmij987
v/0dr9Shpnlfe8uFKpSruvFLHWLddIEOZ4stx//ex/brFWeN1NtSu9i2blqvrz+6Xz+9Znm9TjQem079YiYe1dJMXDP5kv78/j
MZ7xFuU3qBqtarD2VJLv5qIRvSZB56q1/FP/tMexT3TdcE9WuP21q4Y0MUvWdFxo7V7v68/9vyc6bVI37D2tDnz2tXLW67e
31xIFsPbkjSfISVTduf1RPHMj2ObIXx9OHs/VOUPL37/q7vq+nD7fu3/GU7Zd2Md56zxPafXDmuOOeb39r8698+Sr96Td21
e5hg+9KVHde1rz5sTa7uYb9y+S0nPq9+cNK/ntvt217d/845duvLlq+plnjqU1a33PKFRx3tePblTm/ehLz3adr1bt+/S1a869q
aX3vft35ypeqGk7F25bdP1XQoWxR737d+TqULdb/Ne9v4zn+r1/9kcaa6m+7WD70pUe1f6owZ9r4TKF+XjvPY+3vrfc8oaT
xib9+1QtH7PjRu6+B0Qe9+3fn1DxeNy1/72vNa9uemHbvm1K18qaqx8c51/anDfnKnFmfzdmP1bOv2XW3b4Mfv9fd16/
OOXapWVe/D9hzK1rft7rw1tqPG1/unCvqTrz8+73k/nCu2PZe1/fzgFx/t2j/fdt9uPTme1ZUvX1Vff/M227XTm3b4fcZNO/w
fE0ut3nXr59KJWMv8du3ptvt263Cu2HI82u1/47k4IC2qWpVKIda6deN2/5i3Oy61GGvnu14f757b1zf3E43Hpds147b7duu
Pf6//Jpz57SldrHdvMPv7xvrROOx6dQvHsoWtfvgjPZPFeZt97X61Xgua22xsb/ZP1Xo2uc01oNa39dcdnzGj6fdPWBj2Q9+0
rQTih9ZNRp3mNIOM9s5Pj7e73D+zZ5v+MBUky9V9fxUoU8Rvbien8q33b+D0/kXVLZfOsVYbXq+bCFxz7e/tflmmvO3uXx
LsGmOvHfNOcDb2rdv6ZotlSeoYa9X565ktlOuvF7Le5vrbaZmFlms+/9liWePT85+Ddus/OL3wNnc0V+pYN8w6t4UjwfFsjG

Qtpuc32Zrx7XyJtFAzaX7XT/1LiO5v6mXdn5trN/snPbb57XaTsHJl/Y/Xun2i7d97b/DNXLzw1h+KwCtNPLBM8+SWc1vD8zn
5W5kLjn29/G+c1/G8s75/89vU15SUonvLbLpeJe13XWXq8cTilqC4uhcVvdymfi7WOqradx+41fzqzF0WmfMsn200cH5razo
0tSc61g3nOreFpQ3Hc74YG/et23FKd6jjinaZ3OrbtYm7e7nxtRnt+ZOJex3ZTK1Ord932t7ned6t76TaxdSrXKbZM3OsY10L
PR7tt1JbpVu+79SHNr7u973RN67beWmy1aZ1i7LSObu2IUaf61vx+6TzXmnZ1ar5tzBfrkvTCzud85ear2wst3+56tpC221xl
a2bnmlc0jWStjeV2S7pncHrt0r6xsn6+zuStO70JLzt3jDnZmbv5g1ad3qzm5G9OFaPZHTr214xZ/9ufdsrtHqkdf+Op2y/tlv
f1PascP9mnprvUt+/O/v/tcPZ71K4d04ZnD2nKJH8MHLI3XNoYPX7VBdzywZ06s7WLetnm98uWytIzSfl+2XLK2vv2tm9br7
5rSFK0O3bdYt965Xp99oE92nLjWuUKJf3eFS+ds+xlJq4PN52n5vX95f1PKhmLaHK22LbsyqGElmfi+vT9T2okE6//a97fxnN
uXddS9ubN6/VX//cp3fHAnpaYasex3bytm9arXK3q+svWzamv7WJs3reRTFy/c/ILWrZ12mBCn77/SV1/WWtbueOBPS37c9
vVueidfrU2250cGE1q4YbFv3Pn3/k/X+pFvfdscDexSRWtrZ1k3+73jV+rBzl2fq22133hrbUePr04cS+p3LXzLveV+Wjrc9l7X9v
v157Xwuz8R12mBCX31kX9tzvmo42dlGa8d2xw/21eP79P1Pzollxw/2dWyftXObK5Ra5rdrT1suWauRdLzleLTb/8ZzsTzj/9
4495ieMZzUTfOcl22b16tYqybY9X7VrTW27pTbIzhvNaNtVXWLNl9oe/7/99INz2IK72LZu8vv7xjrReGw69YvLM3GtPW1AK4
463Zq937N02vLNs9rVy9vuXqD1q8c1gvRLrba8dx21QZduHJ40T83hOGzCtCO9TKfYmZvlvTfJlUULfcY59xEz2yZpp3Nuu5k
FhRrHEWrUFHci2g6X9SyTElRvVw7PGUWvR7bKFozBX/0qtooWoVStT76w3yjaGXinj86STKmYqWilQ2jaE1kCxpM+qODXc
WVpOfkjHQ0lY6q6uaNoTWSL9VGsDkzINTKQULZQCKaZiehw1o8vE4yilS1UNJTyglG0ikrGlnNH0RplqFjxR9GKe1YfCSwd
j/iHnJ39+BpKdCuallKa/tKFr5UkUjL6JopHUUrUMzRS1rGUXLH80kW6xotlDRQCqqTNMowuMz/rnOI4+NoIWLLxOP6tkj/i
hLKdi2Wn/VF5nDDeMopXyR4oaWMAoWkvTMVVVtwSnTuKVtbf/1KlorgXDUBRqmhpXlNE/jlNJajzRtEqIKrKlcpKxTzN5P
krSnijlRtBL+iE8DtVG0UjHlpgHSSpVKMlpWpX7OaqNoHZopamkmrnLFHw2oPorWUEJO0viMP8KSuzByVzquUrWqqdm
yjb1GLDCZmkmUJZCS+qqdmSlg/E5UUi/ghAwQhLx0bRqsiLRmTy28tAlqplmfajaNXa8NL6KFqVel9aH0VrMKFlGX8UraPB
VaUTQS1US2ojFgFK19k3l/xKymUbRqT182j6l1kPCUiK1kStquHkUrYfJl0ldG0Wrli8SUTQiedGllpo7ilatZ49YVW7OKFp+X
ZSCaufLhFK1aH5P0+7Dm604qHlUs6tfjWh83OpCQs7mjal0OJFR2VRXLFSW81IG0jmSLyrQZRetlzu+/4l4wipbaj6JO8+Y
fEXDGa8+itZAwo8vFo0oGYvU+7o5o2gNjSWT348Go2hIG0fRSvn3CI5kYaNoTcwUlamNohWNKBULRtFaklSl6rR/sqCBZF
R9FazKvWNR0dNbvX1/eNlrWYn5uCMNnFZy6zOwh59zGlulhe2Am7AkeAAAAAACAF6tOCZ5efkULAAAAAAAi4AEDwAA
AAAAABCjgQAAAAAABayJHgAQAAAAAACDkSPAAAAAAAACFHggcAAAAAACDkSPAAAAAAAACEHAkeAAAAAACakCPBAw
FmT1uZmNmmdkOb+deb2Q/N7GEzu8/MzullPAAAAAAAACejniV4zCwq6XJB5J0gaS3m9kFTcW+J2mjc+7lkj4v6Y96FQ8AA
boaZvcfMdpRZzvHx8RcxRAAAAAAAGPA7IX5k2cx+RdJGSX/cbr5z7lPOuY3OuY2jo6OLGxwAAAAAAMAJzuvhuvdJOqvh/Zn
bMEj3OuLOl9kr4m6TFJdzndpnZNjPbHBT7Y0kDkv6XmX3fzLZ3WB0AAAAAAA66OVXtOSc+7KkLzdNu7Hh9aW93D4A
PBaWAAAAAAEHlikeAAAAAAAEEKOBA8AAAAAAEDikeABAAAAAAAIORI8AAAAAAAIIUECBwAAAAAAAIORI8AAAAAAAIIQ
wAAAAAAAhhBwjHgAAAAAAGlDraYLHzK4ws8fNbMzMbuhS7i1m5sxsYy/jAQAAAAAAOBn1LMFjZlFjt0t6k6QLJL3dzC5o
dWb2KklnoEf+sduKzOw9ZrbTzHaOj4+/+JECAAAAAAACEWN9+ZNnMlPjulfSf5ivrnPuUc26jc27j6Oho74MDAAAAAAAAlKv4
YUtb9jWtyT9jnNu5/xhAwAAAAAAoKZrgkf+UzaS9BJJPylpe/B+k6R/7bagc65sZu+T9DVJUUmfc7tMrNtknY657Z3Wx4AA
4Xg2kAAAAAADos/m+olXzWUn/amZfCN5fLemO3oQEAAAAAAACA47GgBI9z7iNm9hVJ/y6Y9KvOue/1LiwAAAAAAAAs1F
6AAAAAAA+mihCZ6i88dtd5JkZpnehQQAAAAAAIDjsdAEz11m9heSlpjZr0u6V9Jf9i4sAAAAAAAALNRCR9H6EzO7TNKU
U9jvSTRpZB5umDUo6Z97GRgAAAAAAAAWZr6vaP2dpK9I+qikGxqmTzvnJnoWFQAAAAAAAABZsvgSPc849bWbvbZ5hZstl
3sTEgAAAAAAAI7HQodjv8TMvmxmK81sg6Rvyx9JCwAAAAAAAHO232/wSJkcc+8ws1+S9likrKR3OOcYJhOAAAAAAOAESN
45J+m/mdmO3oUFAAAAAACAher6BI+Z/Wdjcs5NmmdkvNs3+D/Ot3MyuMLPHzWzMzG7oUOZtZvZDM9tlZn+30MABAA
n3B7nXFHsNZKuairz65Jud84dkSTn3MF51gkAAAAAAIAm8yV4XlF7d43WyXp2Yb3zwXTGq2TtM7M/tmMvm1mbZ8KMrf
8yWWAAAAAAAATildEzzOuegLWPc+SWc1vD8zmNboOUnfcc6VJD1lZk/IT/gwBDsAAAAAAAMACzfcVrRfiQUlRzexcM4vL/8
5w73KuYAAAAAAAATkbmXLh+0mbjxo1u586d/Q4DAAAAAABG0ZnZQ865jc3Te/kVLQAAAAAAAACwCEjwAAAAAAAhhR4l
AAAAAAACFHggcAAAAAACDkSPAAAAAAAACEHAkeAAAAAACakCPBAwAAAAAAAEHlikeAAAAAAAEEKOBA8AAAAAAEDike
PzB42szf3Mh4AAAAAAAICTUC8SPGYWIXS7pDdJukDS283sgqZiH5R0I3PulZKukfRnvYoHAAAAAADgZNLJ3gukjTmnNvjnc
vYgVAAAAAAAGtPr9I8tvl/RXzrkzjb1Z0l+bWUtMzrIPOec2Ouc2jo6OLnqQAAAAAAAj7JeJnj2STqr4f2ZwbRG75J0lyQ55/5F
+1Op5ckBEgl6RDsICgRGMdBIGw6kPC+lq6g4jPo+4lgqDOO77xGgo4zPlqj44oPjKDiBuoArwullOMAskIYzIsjYsks/R6l7q1
QQK8PNPgbg74QqawH8AMD7pZryb8VERERERERERHQg0vbmwaWUv0Dw8uTaZZ+u+fIPAP56b8ZARERERERERHSgm
ERNTh+ICHilililililKjD8QEPEREREREREVGH4wMelililililqIoxwc8REREREREREREQdjg94ilililililg6HB/wEBERERERERERF1OD7gI
DscHPEREREREREREHY4PelilililililOHWf8BARERERERERERdTg+4CEililililil6NBwENERERERERERE1OG0vXVglcSNAM4BsE1Ku
oi9HPH+G0e7auPNGBpsZ0Nv1tytuH1fYtNlcN2ktosFs7M4oi/Y/9XWx+7u32r7xnWH92SwabTUtG203c5iBYaqoGR7GMi
vr+nhmyzi2jFs4tCeNlKZie6ESW6boc21ZGreLrntKU+H7EiXHhecDjutds7EhGXDUNW6OKNrvrOwq3wLeuuve9QmbNdL
Gdek40BQNO4sVzErrSOSqbM9H2fZrdjzkUxrSugrXl9g+aaEvZ0JVBFQhMFkykTU1+NJHWtNQqLGwdRWOFF5wzY2jYXqig
aQ1lx0VaU6GpCkaKNvjPbHbbjW1oKFQcmKoKU1eq5yk5HooVF4d0p5EzNWYdsDC/J4VtEza2TliYkzeR1gVKjsSOQgV9OR
U/6yBkathds9OZ1mlqKCctFoeKiL2cgpQftz/EkHN9D3tRhOT6Ktot8SoMAYGoqSo6LrK7BB1CsODA1DaMIG7OzUd3qcDv
foowlCk5aDrrSOSctBSIPRldZhOS7G5i66MzoKFQeaoiBnakjPsCZKNjRVxbbw/JDA9kIFOVODqSsoVBz0pE1YroexkoOsEbl
lwlAVbBopY063CQAYLzmYldEhALhSwNYlCpal3pwBy3GRMYLylB0fjutsuliravIGCryKRUTZQ/DkxX05wyYqojxy0Ha0DBZ
A1NXwn7rQUoF2yYrmJM3YWhBeW1PomQH9deVUjFZ9jBSstGTNeCH5SyGbdP1PahCxc6ijbypIzFSMGm5KNkuerMmxs
lzkTA2GLqArKiYrLgphP9WUcOwydOwsVjA7ayCtqxgr2ehO67A9WS2PpgApXYWhKXA9iaLtYwFBDvqOCPq7hA9DVCmX2
AAHklJREFUqNiwo4hNI0WkjaDvd5k6RksOullaTE3F8ISFXHhNovG67HiYkzfh+BKTloPerlms7WGkaKM/b4bX3USXqWLzu
mcPScPF4el2NnMbgnFSwPO4o2ukwNaUNF1lQwXgr6ydwuEwNdJl4Zq59zHdQdpx+2TmDzWBI9+aC/qlqCtKZCCKAYzk
UHE9vDJWRI/OhOP7GC06OGRWGksHdh1rlj+CgMSfdxSDe31Gx9I5XVAUUb3OWVNDxfVwSHemOI9lmsc1zlvamT/GzZ
Trtd5/pdqCU/0ApX1SEIHLvHFilkwEUANyc8IDnrQA+guABz4kAviylPHGq4y5fvlw++uij0x3uPjFWtnDX09vx6dufhuX4SOk
PF4QK+fM+LdeU+8+g5uOvZ4T2ui92tz1bbA6hbt6A3jY+cNoh/+vnTsTH/66+exTuWH46v/HZXma5YMYibH9yI0ZI95bZXn

```
"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'\n# of Bivariate Outliers: {len(file.loc[file['Balance'] > 220000])}\n')\n"
],
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "mj4Mtggv9li"
  },
  "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-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06
-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 1.41-.59 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",
" \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\n",
"       + ' 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 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06
-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-.52 0-1.0
.41c.39.39.9.59 1.41.59 1.41.59 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\n",
"       + ' 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 **scikit-learn** framework"
  ]
},
{
  "cell_type": "code",
  "execution_count": 69,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "jegg71flzfdC",
    "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-60tIA",
"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": "7iXQIG7906L7",
    "outputId": "16eada5-a3dc-4bbd-ba0d-bad37cbc4e73"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "(3300, 11)"
        ]
      },
      "execution_count": 76,
      "metadata": {},
      "output_type": "execute_result"
    },
  ],
  "source": [
    "x_test.shape"
  ],
}
```

```
    "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  
}
```