

```

{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "Assignment 2\n",
        "\n",
        "Phthon Programming\n",
        "\n",
        "1. Assignment date: 26-09-2022\n",
        "2. Student Name: M.AARTHI\n",
        "3. Student Roll Number: 923819106028\n",
        "4. Maximum Marks:2Marks "
      ],
      "metadata": {
        "id": "enJlsqwbMjqA"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "1. Importing Requiried Package"
      ],
      "metadata": {
        "id": "xXjEePZhAqHC"
      }
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "DFtZfmlg-Y9r"
      },
      "outputs": [],
      "source": [
        "import pandas as pd \n",
        "import seaborn as sns\n",
        "import numpy as np \n",
        "import matplotlib.pyplot as plt"
      ]
    },
    {
      "cell_type": "markdown",

```

```

"source": [
  "# 2. Loading the Dataset"
],
"metadata": {
  "id": "C68QWd1HB3s6"
}
},
{
  "cell_type": "code",
  "source": [
    "df=pd.read_csv(\"/content/Churn_Modelling.csv\")\n",
    "df"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 488
    },
    "id": "6bIeDd4DCCWP",
    "outputId": "e6ca071c-3f8d-44c7-fad6-31458a0bd00c"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "   RowNumber  CustomerId  Surname  CreditScore
Geography Gender  Age
0      1      15634602  Hargrave          619
France Female  42
1      2      15647311    Hill          608
Spain Female  41
2      3      15619304    Onio          502
France Female  42
3      4      15701354    Boni          699
France Female  39
4      5      15737888  Mitchell          850
Spain Female  43
...      ...      ...      ...      ...
...      ...      ...      ...      ...
9995      9996      15606229  Obijiaku          771
France Male  39
9996      9997      15569892  Johnstone          516
France Male  35
9997      9998      15584532    Liu          709
France Female  36
9998      9999      15682355  Sabbatini          772
Germany Male  42
9999      10000      15628319  Walker          792
France Female  28
Tenure  Balance  NumOfProducts  HasCrCard
IsActiveMember
0      2      0.00      1      1
1
1
1      1      83807.86      1      0
1

```

```

0  \n",      "2          8  159660.80          3          1
0  \n",      "3          1          0.00          2          0
1  \n",      "4          2  125510.82          1          1
... \n",      "...          ...          ...          ...
0  \n",      "9995         5          0.00          2          1
1  \n",      "9996        10   57369.61          1          1
1  \n",      "9997         7          0.00          1          0
0  \n",      "9998         3   75075.31          2          1
0  \n",      "9999         4  130142.79          1          1

```

```

\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  \n",
"...          ...          ...  \n",
"9995         96270.64          0  \n",
"9996        101699.77          0  \n",
"9997         42085.58          1  \n",
"9998         92888.52          1  \n",
"9999         38190.78          0  \n",
\n",
"[10000 rows x 14 columns]"
],
"text/html": [
\n",
"  <div id=\"df-ecfc8a2f-3a09-43e5-a5a6-81502f66851c\">\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",

```

[illegible]

```

"      </tr>\n",
"      <tr>\n",
"          <th>9995</th>\n",
"          <td>9996</td>\n",
"          <td>15606229</td>\n",
"          <td>Obijiaku</td>\n",
"          <td>771</td>\n",
"          <td>France</td>\n",
"          <td>Male</td>\n",
"          <td>39</td>\n",
"          <td>5</td>\n",
"          <td>0.00</td>\n",
"          <td>2</td>\n",
"          <td>1</td>\n",
"          <td>0</td>\n",
"          <td>96270.64</td>\n",
"          <td>0</td>\n",
"      </tr>\n",
"      <tr>\n",
"          <th>9996</th>\n",
"          <td>9997</td>\n",
"          <td>15569892</td>\n",
"          <td>Johnstone</td>\n",
"          <td>516</td>\n",
"          <td>France</td>\n",
"          <td>Male</td>\n",
"          <td>35</td>\n",
"          <td>10</td>\n",
"          <td>57369.61</td>\n",
"          <td>1</td>\n",
"          <td>1</td>\n",
"          <td>1</td>\n",
"          <td>101699.77</td>\n",
"          <td>0</td>\n",
"      </tr>\n",
"      <tr>\n",
"          <th>9997</th>\n",
"          <td>9998</td>\n",
"          <td>15584532</td>\n",
"          <td>Liu</td>\n",
"          <td>709</td>\n",
"          <td>France</td>\n",
"          <td>Female</td>\n",
"          <td>36</td>\n",
"          <td>7</td>\n",
"          <td>0.00</td>\n",
"          <td>1</td>\n",
"          <td>0</td>\n",
"          <td>1</td>\n",
"          <td>42085.58</td>\n",
"          <td>1</td>\n",
"      </tr>\n",
"      <tr>\n",
"          <th>9998</th>\n",
"          <td>9999</td>\n",
"          <td>15682355</td>\n",
"          <td>Sabbatini</td>\n",
"          <td>772</td>\n",

```

```

"      <td>Germany</td>\n",
"      <td>Male</td>\n",
"      <td>42</td>\n",
"      <td>3</td>\n",
"      <td>75075.31</td>\n",
"      <td>2</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>92888.52</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>9999</th>\n",
"      <td>10000</td>\n",
"      <td>15628319</td>\n",
"      <td>Walker</td>\n",
"      <td>792</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>28</td>\n",
"      <td>4</td>\n",
"      <td>130142.79</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>38190.78</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"<p>10000 rows × 14 columns</p>\n",
"</div>\n",
"    <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-ecfc8a2f-3a09-43e5-a5a6-
81502f66851c')\">\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 2.06-.94-2.06-
.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-
.94L8.5 2.51-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 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 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78
2.05 0 2.83L4 21.41c.39.39.95.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78
2.81-2.81c.8-.78.8-2.07 0-2.86z\"M5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41
20z\"/>\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-ecfc8a2f-3a09-43e5-
a5a6-81502f66851c 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-
ecfc8a2f-3a09-43e5-a5a6-81502f66851c');\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 notebook</a>'\\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",
        "        "
    ]
    },
    "metadata": {},
    "execution_count": 19
  }
]
},
{
  "cell_type": "markdown",
  "source": [
    "# 3. Visualizations"
  ],
  "metadata": {
    "id": "aW8u6bWkG63P"
  }
},
{
  "cell_type": "markdown",
  "source": [
    "### 3.1 Univariate Analysis"
  ],
  "metadata": {
    "id": "iiQs2YbbHLmS"
  }
},
{
  "cell_type": "code",
  "source": [
    "sns.displot(df.Tenure)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 386
    },
    "id": "-E6ImBXmG577",
    "outputId": "b2296c3f-e13d-4146-f9d7-98a519c621a0"
  },
  "execution_count": null,
  "outputs": [

```

```
{
  "output_type": "execute_result",
  "data": {
    "text/plain": [
      "<seaborn.axisgrid.FacetGrid at 0x7f25a7da9e90>"
    ]
  },
  "metadata": {},
  "execution_count": 20
},
{
  "output_type": "display_data",
  "data": {
    "text/plain": [
      "<Figure size 360x360 with 1 Axes>"
    ],
    "image/png":
"iVBORwOKGgoAAAANSUUhEUgAAAWAAAAFgCAYAAACFYaNMAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAATdeLEQVR4nO3df6ymZX3n8fdHRvyBWwfo7CzODAuNxJaIseIUEXYb13EbtG6H3adYWJ01dKdJqdXStGJ3Eze72UQTt/5oG1YWqENKUJeYgXYNlgLabFupA7oqjF0mKMwMvw4WsbWxOPjdP55r4nE6ca5znud8Oee8X8mt576v+7rv5/tkZj5zP9f9K1WfJGn5Pau7AElaqxgSWpiAEtSEwNYkpoYwJLUZF13AbNw7rnnl0033thdhiQdkimlrso94EcceaS7BEla0KoMYElacQxgSWpiaEtSEwNYkpoYwJLUxAcWpCYGsCQ1MYAlqYkBlelNDGBJamIAS1ITAliSmhjAktTEAF5lNm05mSQLvjZtObm7VGnNW5X3A17L7t+/jws++ucL9vvEL569DNVieiruAUtSEwNYkpoYWDPIwKykhTgGPCOOxUpaiHvA0irS8cvLX3tHzzlgaRXp+OXlr72j5x6wJDUXgCWpycwCOMmVSR508pV5bSckuSnJ3eP9+NgeJB9JsJfJl5KcMW+dHaP/3U12zKpeaTFjmY5jappmQQb8MeB3gKvmtV0C3FxV70tyyZh/N/A64LTxeiVwKfDKJCcA7ww2AgXcnusGqnp0hnXradq05WTu379vwX4v2ryFA/vuW4aKjs5ixjIdx9Q0zSyAq+pPk5xyWPN24NVjehfwGSYBvB24qqoK+FyS9ULOGnlvqqq/BkhyE3AucM2s6tbT50EY6egs9xjwxqp6YEw/CGwc05uA+btQ+0fbk7X/A012JtmdZPfC3Nx0q5akGWg7CDf2dmuK27usqrZWldYNGzZMa7OSNDPLHCAPjaEFxvvDo/0ASGVev82j7cnaJWnFW+4AvgE4dCbDDuD6eelvG2dDnAU8NoYqPg38dJLjxxkTPz3aJGnFm+VpaNcAfWG8JMn+JBcC7wP+VZK7gdeOeyBPafCa7wSwDj49l+Azz4/Xfz50QE6rl5e2aq2Y5VkQP/cki7YdoW8BFz3Jdq4ErpxiaXqG86wKLcZqOP3Re0FIWpFWw3/UXoosSU0MYelqYgBLUHMDWGrkGR9rmwfhpEar4UCSjp57wJLUxAcWpCYGsCQ1MYAlqYkBlelNDGBJamIAS1ITAliSmhjAktTEAJakJgawJDUXgCWpiQEsSU0MYelqYgBLUHMDWJKaGMCS1MQAlqQmBrAkNTGAJamJASxJTQxgSWpiaEtSEwNYkpoYwJLUxAcWpCYGsCQ1MYAlqYkBlelNDGBJamIAS1ITAliSmhjAktTEAJakJgawJDUXgCWpiQEsSU0MYelqYgBLUHMDWJKaGMCS1KQlgJP8apI7k3wlyTVJnpvk1CS3Jdmb5BNJjh19nzPm947lp3TULEnttuwBnGQT8CvAlqp6KXAM8GbG/cAHq+rFwKPAHWOVc4FHR/shRz9JWvG6hiDWAc9Lsg54PvAA8BrG2rf8F3DemN4+5hnLtyXJMtYqSTox7AFcVQeAdWd3MQnex4DbgW9W1cHRBT+wauXvAvANdQ+O/icevt0ko5PsTrJ7bm5utl9CkqagYwjieCZ7tacCLwKOa85d6nar6rkQ2lpVWzds2LDUZUnSzHUMQbwW+FpVzVXXvd4HrgHOA9WNIAmAzcGBMHWC2AIzllWs+sbwlS9L0dQTWfcBZSZ4/xnK3AXCBtwLnjz47gOvh9A1jnRh8lqqqZaxXkmaiYwz4NiYH0+4AvjxquAx4N3Bxkr1MxnivGKtcAZw42i8GLlnumiVpFtYt3GX6quq9whSPa74HOPMIfb8DvHE56pkK5eSVcJLUxAcWpCYGsCQ1MYAlqYkBlelNDGBJamIAS1ITAliSmhjAktTEAJakJgawJDUXgCWpiQEsSU0MYelqYgBLUHMDWJKaGMCS1MQAlqQmBrAkNTGAJamJASxJTQxgSWpiaEtSEwNYkpoYwJLUxAcWpCYGsCQ1MYAlqYkBlelNDGBJamIAS1ITAliSmhjAktTEAJakJgawJDUXgCWpiQEsSU0MYelqYgBLUHMDWJKaGMCS1MQAlqQmBrAkNTGAJamJASxJTQxgSWpiaEtSk5YatRI+ybvJVppk25JXJTKhyUlJ7h7vx4++sfKRJHuTfCnJGR01S9K0de0Bfxi4sap+fHgZsaE4BLi5qk4Dbh7zak8DThuvncCly1+uJE3fsfgdwkhCPwVcAVBVj1fvN4HtwK7RbRdw3pjeDlxVE58D1ic5aZnLlqSp69gDPHWYA34vyResXJ7kogBJVT0w+jwIbBzTm4B989bfP9p+QJKdSXyn2T03NzfD8iVpOjoCeBlwBnBpVb0c+Dbfh24AoKoKqKez0aq6rkQ2VtXWDRs2TK1YSZqVRQVwnMMW07ZI+4H9VXXbmL+WSSA/dGhoYbw/PJYfALbMW3/zaJokFW2xe8C/vci2BVXVg8C+JC8ZTduAu4AbgB2jbQdw/Zi+AXjbOBviLOCxeUMVkrRirXuqhUleBZwNbEhy8bxFPwQcs4TPfQdwdZJjgXuAtzP5z+CTSS4E7gXeNPp+Cng9sBf4u9FXkla8pwxg4FjjGaBfpP5rX/i3g/KP90Kr6IrD1CIu2HaFvARcd7WdJ0jPVUwZwVX0W+GySjlXVvctUkyStCQvtAR/ynCSXAafMX6eqXJOloiRpLVhsAP9P4L8DLwNPzk4cSVo7FhvAB6vKS4AlaYoWexraHyb5pSQnjZvmnJDkhJlWJkmr3GL3gA+dn/vr89oK+JHpliniJa8eiAriqTp11IZK01iwggJO87UiTvXXVdMuRPLVjsUMOPzlv+rLMLpi4
```

AzCAJekoLXYI4h3z55OsBz4+k4okaY042ttRfpvJfX01SUdpsWPAf8j37897DPBjwCdnVZQkr  
QWLHQP+wLzpg8C9VbV/BvVI0pqxqCGICvOerzK5I9rxwOOzLEqS1oLFFhHjTcBfAm9kcp/e25  
Ic9e0oJUmLH4L4D8BPvtXDAEk2AH/C5HFCkqSjsNizIJ51KHvHbzyNdSVJR7DYPEAbk3wauGb  
MX8DkUUGSpKO00DPHXgxsRkpft/JvgX8+Fv0FcPWSi5Ok1WYhPeAPAE8BqKrrgOsAkVz4WPav  
Z1qdJK1iC43jbqyqLx/eONpOmUlFkrRGLBTA659i2fOmWYgkrTULBfDuJP/+8MYkvwDcPpuSJ  
GltWGgM+F3A/0ryFr4fuFuBY4F/M8vCJGmle8oArqqHgLOT/EvgpaP5f1fVLT0vTJJWucXeD/  
hW4NYZ1yJLTZtOZn79+9bsN+LNm/hwL77pva5i70QQ5JWrfv37+OCj/75gv0+8YtnT/VzvZx  
YkpoYwJLUxACWpCYGsCQ1MYAlqYkBLElNDGBJamIAS1ITAlisMhjAktTEAJakJgawJDUxgCWp  
iQEsSU0MYElqYgBLUhmDWJKaGMCS1MQAlqQmBvA8m7actJKnfG3acnJ3mZJWibaHciY5BtgNH  
KiqNyQ5Ffg4cCJwO/DWqno8yXOAq4BXAN8ALqiqR8+ipsU8mG/aD+WTtHZ17gG/E9gzb/79wA  
er6sXAo8CFo/1C4NHR/sHRT5JWvJYATrIZ+Bng8jEf4DXAtaPLLuC8Mb19zDOWbXv9JWlF69o  
D/hDwG8D3xvyJwDer6uCY3w9sGtObgh0AY/ljo/8PSLIzye4ku+fm5mZzuyRNxbIHcJI3AA9X  
1e3T3G5VXVZVW6tq64YNG6a5aUmaiY6DcOcAP5vk9cBzgR8CPgysT7Ju7OVuBg6M/geALcd+J  
OuAfzI5GcdJK9qy7wFX1XuqanNVnQK8Gbilqt4C3AqCP7rtAK4f0zeMecbyW6qqLrFkSZqJZ9  
J5w08GLk6y18kY7xWj/QrgxNF+MXBJU32SNFvt5wEDVNvngM+M6XuAM4/Q5zvAG5e1MElaBs+  
kPWBjWlMMYElqYgBLUhmDWJKaGMCS1MQAlqQmBrAkNTGAJamJASxJTQxgSWpiAEtSEwNYkpoY  
wJLUxACWpCYGsCQ1MYAlqYkBLElNDGBJamIAS1ITAlisMhjAktTEAJakJgawJDUxgCWpiQEsS  
U0MYElqYgBLUhmDWJKaGMCS1MQAlqQmBrAkNTGAJamJASxJTQxgSWpiAEtSEwNYkpoYwJLUxA  
CWpCYGsCQ1MYAlqYkBLElNDGBJamIAS1ITAlisMhjAktTEAJakJgawJDVZ9gBosiXJrUnuSnJ  
nkneO9hOS3JTk7vF+/GhPko8k2ZvkS0nOWO6aJWkWOvaADwK/VlWnA2cBFyU5HbgEuLmqTgNu  
HvMArwnOG6+dwKXLX7IkTd+yB3BVPVBvd4zpvwH2AJuA7cCu0W0XcN6Y3g5cVROfA9YnOWmZy  
5akqWsdA05yCvBy4DZgY1U9MBY9CGwc05uAffNW2z/aDt/WziS7k+yem5ubWc2SNC1tAZzkBc  
AfAO+qqm/NX1ZVBdTT2V5VXVZVW6tq64YNG6ZyqSTNRksAJ3k2k/C9uqquG80PHRpaG08Pj/Y  
DwJZ5q28ebZK0onWcBRHgCmBPvf3WvEU3ADvg9A7g+nntbxtnQ5wFPDZvqEKSvqx1DZ95DvBW  
4MtJvjjaFhN4H/DJJBcC9wJvGss+Bbwe2Av8HfD25S1XkmZj2QO4qv4PkCdZvO0I/Qu4aKZFS  
VIDr4STpCYGsCQ1MYAlqYkBLElNDGBJamIAS1ITAlisMhjAktTEAJakJgawJDUxgCWpiQEsSU  
0MYElqYgBLUhmDWJKaGMCS1MQAlqQmBrAkNTGAJamJASxJTQxgSWpiAEtSEwNYkpoYwJLUxAC  
WpCYGsCQ1MYAlqYkBLElNDGBJamIAS1ITAlisMhjAktTEAJakJgawJDUxgCWpiQEsSU0MYElq  
YgBLUhmDWJKaGMCS1MQAlqQmBrAkNTGAJamJASxJTQxgSWpiAEtSEwNYkpqsmABOcm6Sv0qyN  
8kl3fVI0lKtiABOcgzWu8DrgNOBn0tyem9VkrQ0KyKAgtOBvVV1T1U9Dnwc2N5ckyQtSaqqu4  
YFJTkfOLEqfmHMvxV4ZVX98rw+O4GdY/YlwF8dxUf9MPDIEst9pvE7rQx+p5XhaL/TI1V17uG  
N65ZezzNDVV0GXLaUbSTZXVbplTSM4LfaWXwO60M0/5OK2UI4gCwZd785tEmSSvWSgngzwOn  
JTk1ybHAM4EbmmuSpCVZEUMQVXUwyS8DnwaOAA6sqjtn8FFLGsJ4hvI7rQx+p5Vhqt9pRRyEk  
6TVaKUMQUjSqmmas1ITA5jVd5lzkilJbklyV5I7k7yzu6ZpSXJMki8k+aPuWqYlyfoklyb5ap  
I9SV7VXdNSJfnV8XfV0KmuSfLc7pqeriRXJnk4yVfmtZ2Q5KYkd4/345fyGWs+gFfpZc4HgV+  
rqtOBs4CLVsF3OuSdwJ7uIqbsw8CNvfWjwMtY4d8vySbgV4CtVfVSJgf039xb1VH5GHD4xROX  
ADdX1WnAzWP+qK35AGYVXuZcVQ9U1R1j+m+Y/IPe1FvV0iXZDPwMcHl3LdOS5IXATwFXAFTV4  
1X1zd6qpmId8Lwk64DnA/c31/O0VdWfAn99WPN2YNeY3gWct5TPMIAnwbRv3vx+VkfYHZLkFO  
DlwG29lUzFh4DfAL7XXcgUnQrMab83hlYuT3Jcd1FLUVUHgA8A9wEPAI9V1R/3VjU1G6vqgTH  
9ILBxKRszgFexJC8A/gB4V1V9q7uepUjyBuDhqrq9u5YpWwecAVxaVS8Hvs0Sf9Z2G+Oi25n8  
5/Ii4LgkP99b1fTV5BzeJZ3HawCv0suckzybSfheVXXddczBecAP5vk60yGiV6T5Pd7S5qK/  
cd+qjR0C+VaJoG8kr0W+FpVzVXVd4HrgLOba5qWh5KcBDDeH17KxgzgVXiZc5IwGVPcU1W/1V  
3PNFTVe6pqc1WdwuTP6JaqWvF7Vvx1ILAvyUtG0zbgrsaSpuE+4Kwkzx9/F7exwg8sznMDsGN  
M7wCuX8rGVsSlyLO0jJc5L6dzgLCcX07yxdH2m1X1qcaa90TeAVw9dgDuAd7eXM+SVNVtSa4F  
7mByRs4XWIGXJSe5Bng18MNJ9gPvBd4HfDLJhcC9wJuW9BleiixJPRyCkKQmBrAkNTGAJamJA  
SxJTQxgSWqy5k9D0+qV5EQmN0wB+CfAE0wu+wU4c9z7Q2rjaWhaE5L8J+Bvq+oDM9r+MVX1xC  
y2rdXLIQitKULEkeSzSW5P8ul515V+Jsn7k/xlkv+X5F+M9n+X5HfmrF9HSV49pv82yX9L8n+  
BVyX5+bH+F5N8dNzqVHPSBrDWkgC/DZxfVa8ArgT+67z166rqTOBdTK56WshxwG1V9TLgG8AF  
wDlV9RNMhjveMs3itfo4Bqyl5DnAS4GbJrco4Bgmt0s85NBNI24HTlnE9p5gcsMjmNzv4BXA5  
8e2n8cSb9Silc8A1loS4M6qerJH/vz9eH+C7//bOMgP/lKc/2id78wb9w2wq6reM6litfo5BK  
G150+BDYeeuZbk2Un+2QLrfB34iSTPSrKfYRNUjuRm4Pwk/3hs+4Qk/3RKdWuVcg9Ya8n3gPO  
Bj4xHaa1j8pSNp7r73Z8BX2Nyi8g9TO7w9Q9U1V1J/iPwx0meBXwXuIjJHbOkI/I0NElq4hCE  
JDUxgCWpiQEsSU0MYElqYgBLUhmDWJKaGMCS1OT/A41BrgMAjLkkAAAAAE1FtKSuQmCC\n"

```
},
  "metadata": {
    "needs_background": "light"
  }
}
```

```

    ],
    {
      "cell_type": "markdown",
      "source": [
        "# 3.2 Bi-Varient Analysis"
      ],
      "metadata": {
        "id": "g7n7kMgfHcQn"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "df.plot.line()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 293
        },
        "id": "BIzPGmabHkaU",
        "outputId": "4efe7fc8-6f45-436e-d82c-6c7d4f10f1f0"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at 0x7f25a542c250>"
            ]
          },
          "metadata": {},
          "execution_count": 21
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x288 with 1 Axes>"
            ],
            "image/png":
              "iVBORw0KGgoAAAANSUhEUgAAAXQAAAEEDCAYAAAAAlRP8qAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjlsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAELEQVR4nO3dd3gU1frA8e+bnkAooUtlQEoCCQuEKB3pIgYRFRaVlCIqgr2A/kT02rgWLGdhKoKCIEQpIl4RBAFBgWBAekBDEwgEElp6zu+P3axJSNmEDSHL+3mePNmZOXpMzMum8nZmfeIMQallFLln1tZN0AppZRzaEBXSikXoQFdKaVchAZ0pZRyERrQlVLKRWhAV0opFlGmAV1EZopIvIhsd6DsVBGJsfsFZHEy9FGpZQqL6Qs70MXkS7AOeAzY0zLYqw3HmhtjBlVao1TSqlypkYv0I0xa4BTOeeJSGMR+Z+IRivIWhFpns+qQ4F5l6WRSilVTniUdQPyMQO43xgTKyLXAe8D3bMXikhDIAj4qYzap5RSV6QrKqCLSEWgA7BARLJne+cpNgSImSzkXs62KaXUle6KCuhYu4ASjTGWQsoMAR68TO1RSqly44q6bdEYcwb4S0RuBxCrVtnLbf3pVYENZdREpZS6YpX1bYvzsAbnZiJyWETuAyyB94jIVmAHMCDHKkOAL42miFRKqYuU6W2LSimlnOeK6nJRSilVcmX2pWj16tVNYGBgWWleKaXKpejo6JPGmBr5LSuzgB4YGMjnzZvLavNKKVUuiciBgpZpl4tSSrkIDehKKeUiNKArpZSLKDKgO5LiVks62dLa7hCRn53bRKWUuo5w5Ap9FtC3oIUiuUgVrAq1IY0wL4HbnNEOppVRxFBnQ80txm8edwDfGmIO28vFOaptSSqlcIEYfelOgqoistuUwH15QQREZiYKbRWTziRMnnLBppZRS2ZxxH7oH0BboAfgCG0TkV2PM3rWfjTEzsOY7Jzw8vOQ5BxIPwtlj8PlASDsHgZ0hIwVqh8GxbZCwD9IuQL1w6096Cmz8CPq8AtvmQ+cn4Pc54F"
          }
        }
      ]
    }
  ]
}

```

cN/vgKsjJy13/bp3AkGjZMg6Y3WufVbwcrX4S7v4ff50KdMKjSEI7/AT+9BBVqQJUGcO4EeHj  
B7bNg5xJY8x+ofz0ENILd38E1Fuj9Evw4CUIGWH8HdYgKtcC3CmRlWrdbqyUcjYE6FrjjMzix  
G/athE0fg7sn9HkZju+EzZ9A1SCodi007QPbv4ZWQyD1LHhXgvQLEngJ/o6BZv3gjjwQfBril  
kFAEOxaAo9sh8QDsOBUqFwPaoZAZWD47UPr75QkaNQNDqyHC6fg4Hp4dCcs/z9IiIVz8dZ9Ce  
wMZ4/C90/1Pp4RY6B5f+t2E/YBxnrMAG79GEym9VhVrAk3vw37VsDicda2HN4EjW6AGs2hbhv  
reU9JAhHY8z20vgt+fB6qN4UWtjYkJ8Lfv0PMF1ClPoTeDv614c+fITPVWpebO2xfCH5V4boH  
4Ngf8Ot0a3v2fg/rpkJwJKSdh86PWc9ToxugWmPYMB18q8K1Pax1Nush3z8J3pWh0jXg6Qs+l  
azn4PgOa9mzR63ncsB02PYlrH/Puv/NboILCRC/E1LPWM9116fg5/9AxL3w52rrub/+QUg6BK  
fjRH9HWM9bhH3QcP21vdNzFzw9AOvCpCZDn9vgbrhCp4E9J8K3z5sXde/DlSqa21n9KfQ+2X  
r52jnYug2wXpuTu6FEUshPRl2fwut7oSF91nfJ9Wuta7bsJP1/WS5E84dt34GaofCX2us53rP  
MujxPPz6PnQYb23b0a3Q8lbrZxdA3KFhB4hbC417QOhtsPYt6/uqxa3Q7Eb430RocL31vZB8G  
m78j3XdNa9bP/OlQqBGMGS1w/ZvrO//mLlgGQarXobK9eEv21d79a+3HhcPX0hNss5re7f1GG  
emQauhlm07938Q1BmuHwtR90D6eajXDrJTpQz8EPb+AIvut763D22EEed9aj/X+ldaYc+Yw7Pr  
2n8/nvh+t6/acDJ0eLXH4K4xDuVxEJBBYmt8wcSiYAfAlxjxvm/4E+J8xZkFhdYaHh5sSPVh0  
5ii8ld8gRkopVU4M/Mj6h6cERCTaGBOe3zJndLksBjqJiIeI+AHXAbucUG/+Vr1calUrpdRls  
fC+Uqm2yC4XW4rbbkB1ETkMPA94AhhjPjTG7BKR/wHbgCzgY2NMgbc4XrKtOpSoUkrlp8iAbo  
wZ6kCZl4HXndKioozbBO+2viybUkqp8uRKG4KuaAGNYHKS9cuJf8YdLZgxtp8scPe4eJnIP8v  
F7Z/p7LqzsQy/Raxl3Nz/WT8ry/qFqofXP/My0nJPg/ULKwTc3KzruOXT05Vzm9nfa5isH0tl  
Wue7uTu23/nJb9vZxye/NhWnrow065e1IpCZcfGxZrVujuNRVfTyHp0FPDwth5zcct9LuzHr  
IB9yXmu8x4/R99L+W0L2znJyvznd/Z8d8+L18vKtG4r+71W1Pazt5V3efbxzpaVcfH2CpNlG5  
I35zHM2UZ3j38+F/mdq5z1Czt2WVnW5dl1OtQ222cuv21mL8v+LGZl2t4L+byXcn6GRRw/x3k  
/i/mt1/c9nD2d33Zy1lHUZ+MSlb+Ans3Rk2M/wPm8ObLrELF+255f3TnfKJLnze/mBm55gnfe  
YA65PzQFFtBybtPerpZr5dl2SeS37eK80QurK+d+F/WGLWhf8mtLzmlPH1v9+QSunOeyoLoLW  
l7c/c+vruX9Kuw8ieR/bArbfkHL8r7PihPMofBzkn3GvJ+L4rTPvh3b+6Q4Qaywiwv7siLqLe  
gz7Ij8PosXtSNvLCjivGcrxWAomstFKAvcHgZ0pZRYERqLVLKRWhAV0opF6EBXSmlXIQGDkW  
UchEa0JVSykVoQFdKKRehAV0ppVxEx+X1SVC11156ezuHDh0lJSSnrpign8fHxoV69enh60v4U  
sAZ0pVzA4cOH8ff3JzAwEClprh91xTDGkJCQwOHDhwkKcNj4PelyUcoFpKsUKl1aQ3mLkJEq  
FatWrH/49KarpSL0GDuWkpyPjWgK6WUiygyoIvITBGJF5FCRYESkXYikiEitzmveUqp8sLd3R  
2LxULLli25+eabSUxMvKT6Jk+ejJ+fH/Hx8fZ5FStWvNRmAhAXF0fLlhcNkVzuOXKFPgvoW1g  
BEXEHpgDLndAmpVQ550vrS0xMDNu3bycgIIDp06dfcp3Vq1fnzTffdeLrncsjI60sm5CvIgO6  
MWYNcKqIYuOBr4H4Isoppa4C7du358iRIwDExMRw/fXXExYWxsCBazl9+jTx8fG0bdsWgK1bt  
yIiHDx4EIDGjRtz4cIFAEaNGsX8+fM5dSp3CMp7hf3GG28wefJkALp168ajjz5KeHg4wcHBbN  
q0iVtVvZUmTZrwf//3f/Z1MjIyGDZsGMHBwdx22232bUZHR901a1fatm1Lnz590Hr0qL3eRx5  
5hPDwcN55551SOGqX7pJvWxSRusBA4AagXRfLxwBjABo0aHCpm1ZK5eOFb3ew8+8zTq0z5JpK  
PH9zC4fKZmZmsnL1Su655x4Ahg8fznvvvUfXr12ZNGkSL7zwAm+//TYpKSmcOXOGtWvXEh4ez  
tqla+nUqRM1a9bEz88PsHaxjBolinfeeYcXXnjB4fZ6eXmxfNm3nnnHQYMGEb0dDQBAQE0bt  
yYRx99FIA9e/bwySef0LFjR0aNGsX777/Pww8/zPjx41m8eDElatRg/vz5PPvss8ycOROAtLQ  
ONm/eXJxDd1k54z70t4GnjTFZRX0ra4yZAcwACA8PN07Yt1LqCpGcnIzFYuHIkSMEBwftQ1cv  
kpKSSExMpGvXrgCMGDGC22+/HYAOhTrwy+/sGbNGp555hn+97//YYyhc+fOuep96KGHSFgsP  
PHEEW63JTIYeoDQ0FBatGhBnTp1AGjUqBGHDh2iSpUq1K9fn44dOwJw11138e6779K3b1+2b9  
9Or169Aosfp+x1AQYPhlzCo3N5OCoghWnf2oJ5daCfiGQYYxY5oW61VDE5eixtbN196BcuXKB  
Pnz5Mnz6dESNGFFi+S5curF27lgMHDjBgwACmTJmCiHDTTf1KlelShXuvPPOXH3yHh4eZGUP  
GA0X3a/t7e0NgJubm/119nR2/3fec1ARwRhDixYt2LBhQ75trlChQmGHoMxd8m2LxpggY0ygM  
SYQiaLGajBX6url5+fHu+++y5tvvkmFChWoWrUqa9euBeDzzz+3X6137tyZOXPm0KRJE9zc3A  
gICGDZsmV06tTpojofof+wXpVroI3swrlWrFvHx8SQkJJCamsrSpUuL3c6DBw/aA/cXX3xBp06  
daNasGSdOnLDPT09PZ8eOHSU6DmXBkdsW5wEbgGYiclhE7hGR+0Xk/tJvnlKqPGgrdujVhYWHM  
mzeP2bNn8+STTxIWFkZMTAyTJk0CIDAWEgMMXbp0AaBTp05UqVKFqlWrXlRf9erVGThwIKmpq  
QB4enoyadIkIiI6NwRf82bNy92G5s1a8b06dMJDg7m9OnTPPDAA3h5eREVFcxTTz9Nq1atsF  
gsrF+//hKOxOulxpRNV3Z4eLi5kr9cUKo82bVrF8HBWXdDOVK+Z1XEYk2xoTnV16FFVKKRe  
hAV0ppVyEBnSl1HIRGtCVUspFaEBXSikXoQFdKaVchAZ0pZTTHDt2jCFDhtC4cWPatm1Lv379  
2Lt3b7HqWLRoEtT37iylFhasW7du+eZpmTVrFuPGjBvs7SkJDehKKacwxjBw4EC6devG/v37i  
Y605tVXX+X48ePFqqsAnpmZuZl3V5p0YCulHKKVatW4enpyf33//MQeatWrcjMzKR///72ee  
PGjWPWrFkATJgwgZCQEMLCwnjiiSdYv349S5Ys4cknn8RisbB///580++C421y58yZQ0REBBa  
LhfvuU88evCtWrMjjjz9OqlatLsrd8umnn9K0aVMiIiL45ZdfSuuQOZ0zknMppa4k30+AY384  
t87aoXDja4UW2b59uz3HuSMSEhJYuHAhu3fvRkRiTEykSpUqREZG0r9/f267zTr4WVhYWL7pd  
6HoNLnx8fHMnz+fX375BU9PT8aOHcvCuXmZPnw458+f57rrrrtoAI2jR4/y/PPPEX0dTeXKlb  
nhhht03bp1MQ9Y2dCArpQqE5UrV8bHx4d77rmH/v3757qKz1ZY+l0oOk3uunXriI6Op10761A  
NycnJ1KxZE7AomTdo0KCLtVnbb7/RrVs3atSoAVhT5hb3e4CyogFdKvDTxJV0aWnRogVRUVEX

zS8o1a2HhwcbN25k5cqVREVMW3aNH766adibbOoNLnGGEaMGMGrr7560bo+Pj64u7sXa3tXO  
u1DV0o5Rffu3U1NTWXGjBn2edu2bcMYw86d001NTSUXMZGVK1cCc07cOZKSkuJXrx9Tp05169  
atAPj7+3P27FnAehVfUPpdR/To0Y0oqCj7QNOnTp3iwiEDha5z3XXX8fPPP5OQKEB6ejoLFix  
w/CCUMb1CV0o5hYiwcOFCHnnkEaZMmYKPjw+BgyG8/fbb3HHHHbRs2ZKgoCB7f/TZs2cZMGAA  
KSkpGGN46623ABgyZAJ33nsv7777LlFRUcyePZv777+fCxcu0KhRiz799FOH2xQSEsJLL71E7  
969ycrKwtPTk+nTp9OwYcMC161Tpw6TJ0+mffv2VKLSBYvFcmkH5jLS9LlKuQBNn+uaNH2uUk  
pdpRwZsWimiMSLyPYClg8TkW0i8oeIrBeRVs5vplJKqaI4coU+C+hbyPK/gK7GmFDg38CMQso  
qpZQqJUV+KWqMWSMigYUszzng3q9AvUtv1lJKqeJydh/6PcD3BS0UkTEisl1ENp84ccLJm1ZK  
qaub0wK6iNyANaA/XVAZY8wMY0y4MSY8+ykspZRSzuGUgC4iYcDHWABjTIIz6lRKlT/OSJ+bb  
eTikfYnT0ePHm3PwPjKK6/kKvfyYy/TokULwsLCsFgs/Pbbb5e2E+XYJT9YJCINGg+Afxljyk  
fCA6WU02Wnzx0xYgRffvklAFu3buX48eM0bdoUgIyMDDw8ih92Pv74Y/vrV155hWeeeQaADRs  
2sHTpUrZs2YK3tzcNt54kLS3tkvajpG28Ejhy2+I8YAPQTEQ0i8g9InK/iGTnyJwEVApeF5EY  
EdGnhZS6ChWWPrdz585ERkYSEhJCZmYmTz75J03atSMsLIyPPvoIsP5BGDduHM2aNaNnz572x  
/Xhn8EnJkyQHJyMhaLhWHDhnH06FGqV69uz+NSvXplrnmGgA2bdpEhw4daNWqFREREZw9e5  
aUlBTuvvtuQkNDad26NatWrQKsglHERkbSvXt3evTowfnz5xklahQREERG0bt2axYsXX67DeEk  
cuctlaBHLRwOjndYipdQlmbJxCrtP7XZqnc0DmvN0RIffjwGFp8/dsmUL27dvJygoiBkzZlC5  
cmU2bdpEamoqHTt2pHfv3vz+++/s2b0HnTt3cvz4cUJCQhg1a1Suel577TWmTztGTEwMYM0H8  
+KLL9K0aVN69uzJ4MGD6dq1K2lpaQwePJj58+fTrl07zpw5g6+vL++88w4iwh9//MHu3bvp3b  
u3vUtoy5YtbNu2jYCAAJ555hm6d+/OzJkzSUXMJCIigp49e1KhQgUnHM3SUZ7/r1BKlSsRERE  
EBQUBsHz5crZt22bvH09KSiI2NpYla9YwdOhQ3N3dueaaa+jevXuR9VasWJHo6GjWr13Lq1Wr  
GDx4MK+99hpt27a1Tp069rS5lSpVAMddunWMH8zegObNm9OwYUN7QO/VqxcBAQH2Ni5ZsoQ33  
ngDsGaIPHjw4BWFxkEDulIupqgr6dJSUPpcINEvRTGG9957jz59+uQqs2zZshJt193dnW7dut  
GtWzdCQ0OZPXt2sQbaKKiNX3/9Nc2aNSTRm8qK5nJRSjlFQelzs1PfZuvTpW8ffPAB6enpAOz  
du5fz58/TpUsX5s+ft2ZmJkePHrX3b+f16elpX3fPnj3Exsbal8XExNCwYUOaNwVG0aNH2bRp  
E2DN7JiRkUHNzp2Z03eufbsHDx7MN2j36dOH9957jz+khb//ntJD8tlpVfoSimnKCh97i233  
JKr3OjRo4mLi6NNmzYYY6hRowaLfiLi4MCB/PTTT4SEhNCgQQPat2+f73bGjBlDWFgYbdq04b  
HHHmP8+PEkJibi4eHBtddey4wZM/Dy8mL+/PmMHZ+e5ORkfh19WbFiBWPPhjuWBBx4gNDQUdW8  
PZs2a1WtgjGzPPfccjzzyCGFhYWRlZREUFMTSpUtL5bg5k6bPVcoFaPpc16Tpc5VS6iq1AV0p  
pVyEBnS1lHIRGtCVUspFaEBXSikXoQFdKaVchAZ0pZTTLFq0CBFh927n5pJRjtGArpRymnz5  
tGpUyfmzZtX1k25KmlAV0o5xblz51i3bh2ffPKJPR96VlYWY8eOpXnz5vTq1Yt+/frZ871ER0  
fTtWtX2rZtS58+fTh69GhZnt8l6KP/SrmYY6+8Quou53Z5eAc3p7ZtUImCLF68mL59+9K0aVO  
qVatGdHQ0f/31F3FxczczuZP4+HiCg4MZNWou6enpjB8/nsWLF1OjRg3mz5/Ps88+y8yZM53a  
7quNBnS1lFPMmzePhx9+GIAhQ4Ywb948MjIyuP3223Fzc6N27drccMMNgDWplvbt2+nVqxcAm  
ZmZlKlTp8za7iqKDOgiMhPoD8QbYlrmSlyAd4B+wAVgpDFmi7MbqpRyTFFX0qXh1KlT/PTTT/  
zxxx+ICJmZmYgIAwcOzLe8MYYLWVqWYcOGY9xS1+ZIH/osoG8hy28Emth+XgAfXHQz1FLlSVR  
UFP/61784cOAAcXFxHDp0iKCGIAICAvj666/Jysri+PHjrF69GoBmzZpx4sQJe0BPT09nx44d  
ZbgHrqHIG6MWQOcKqTIAOAzy/UrUEVE9H8npa4i8+bNu+hqfNCgQRw7dox69eoreHLcXXfdR  
Zs2bahcuTJex15ERUXx9NNP06pVKyWC+vXry+jlrsOZ/Sh1wU05Zg+bJt30VfwiJIG61U8DR  
o0cMKmlVJXgVwGo3jooYcA690vFstWJCEhgYiICEJDQwGwWCysWbPmsrbt1V3WL0WNMTOAGWD  
Nh345t62UKhv9+/cnMTGRtLQ0nnvuOWrXrl3WtXJZzgjoR4D6Oabr2eYppZS931yVPmc8WLQE  
GC5W1wNjxhh9QkApps4zR25bnAd0A6qLyGHgeCATWbjzIbAM6y2L+7Detnh3aTVWKAUVuYoM6  
MaYoUUsN8CDTmuRUkqpEtFcLkop5SL00X+l1CVLSEigR48eABw7dgx3d3dq1KgBwMaNG/Hy8i  
rL5l01NKArpS5ZtWrViImJAWdy5MlUrFiRJ554o1S2lZmZibu7e6nUXd5pl4tSqlQUlB63W7d  
uPP3000RERNc0aVPWrl0LwKxZsXg3bpx9/f79+9tveaxYsSKPP/44rVq1YsOGDcyZM4eIiAgs  
Fgv33XcfmZmZl33/rkR6ha6Ui1n71V5OHjrn1Dqr169I5zuaOlzeGFNoetyMjAw2btzIsmXLe  
OGFF1ixYkWh9Z0/f57rrruON998k127djFlyhR++eUXPD09GTt2LHPnzMX48OGXtI+uQA06Us  
rpU1NTC02Pe+uttwLQtm1b4uLiigzP3d2dQYMGAbBy5Uqio6Np164dAMnJydSsWdPJelA+aUB  
XysUU50q6tBSVhtfb2xuwBuqMjAwAPDw8yMrKspdJSUmxv/bx8bH3mxtjGDFiBK+++mppNb/c  
0j50pZTTExT7Fzs9bmBgIDExMWRlZXHo0CE2btYb7kePXoQFRVffHw8YM3FfuDAAefuQDmlV  
+hKKadz3MjKiqKhx56iKSKJDIyMnjkkUdo0aJFget07NiRoKAgQkJCCA4Opk2bNvmWCwkJ4a  
WXXqJ3795kZWxh6enJ9OnTadiwYWntTrkh1gc9L7/w8HCzefPmMtm2Uq5m165dBACHl3Uz1JP  
ld15FJNoYE55feelyUUpF6EBXSmlXIQGdKWUchEa0JVSykVoQFdKKRehAV0ppVyEQwFDRPqK  
yB4R2SciE/JZ3kBEVonI7yKyTUT6Ob+pSqrmbu7OxaLhVatWtGmTrvWr19f5DoVK1a8DC27e  
jgyBJ07MB3oBRwGNNonIEmPMzhzF/g/4yhjzgyiEYB2WLRaU2quUukL5+vraU+j+8MMPTJw4kZ  
9//rmMW3V1ceQKPQLYZ4z50xiTBnwJDMhTxgCVbK8rA387r4lKqfLmzJkzVK1aFYBz587Ro0c  
P2rRpQ2hoKIsXL76ofEF14uLiCA405t5776Vfixb07t2b5ORkAPbt20fPnj3t/xHs378fgNdf  
f5127doRFhbG888/f5n2+MrgyKP/dYFDOaYPA9f1KTMZWC4i44EKQM/8KhKRMcAYgAYNGhS3r  
UopB6yaNYP4A386tc6aDRtxw8gxhZZJTk7GYrGQkpLC0aNH+emnnwBrYq2FCxdSqVilTp48yf

XXX09kZCQiYl+3oDIAsbGxzJs3j//+97/ccccdfP3119x1110MGzaMCRMmMHDgQFJSUsjKymL  
58uXExsayceNGjDFERkayZs0aunTp4tTjcaVyVi6XocAsY8ybItIe+FxEWHPjsnIWMsbMAGaA  
9dF/J21bKXUFyNnlsmHDBoYPH8727dsxxvDMM8+wZs0a3NzcOHLkCMePH6d27dr2dQsqAxAUf  
ITFYgH+Sbd79uxZjhw5wsCBAwHrHwSA5cuXs3z5clq3bg1Yr/xjY2M1oOdwBKifY7qebV509w  
B9AYwxG0TEB6gOxDujkUopxxV1JX05tG/fnpMnT3LixAmWLVvGiRMniI6OxtPTk8DAwFypcQH  
mzp1bYJnsVLtg/eI1u8slP8YYJk6cyH333Vc603aFc6QPfRPQRESCRMQLGAIsyVPmINADQESC  
AR/ghDMbqpQqP3bv3klmZibVqlUjKSmJmjVr4unpyapVq/JNdetImZz8/f2pV68eixYtAqwDa  
ly4cIE+ffowc+ZMzp2zjth05MgRe5rdq0GRV+jGmAwRGQf8ALgDM40x00TkRWCzMWYJ8DjwXx  
F5FOsXpCNNWaVxVEqView+dLBeKc+ePrT3d3eGDRvGzTffTGhoKOHh4TRv3vyidR0pk9fnn3/  
Offfdx6RJk/D09GTBggX07t2bXbt20b59e8B6W+ScOXOumhGNNH2uUi5A0+e6Jk2fq5RSVyKN  
6Eop5SI0oCullIvQgK6UUi5CA7pSSrkIDehKKeUiNKArpZxCRHj88cft02+88QaTJ092Wv0zZ  
sygefPmNG/enIiICNatW2dftnbtWlq0aIHfYmHXrl34+vpisVgICQnh/vvvJysrq5CaCxcYGM  
jJkyeLvV5cXBxfFFfFibdbEhrQlVJO4e3tzTffffFOi4FeUpUuX8tFHH7Fu3Tp2797Nhx9+yJ1  
33smxY8cAa+qAiRMnEhMTg6+vL40bNyYmJoZt27axc+dO+xOl2TiYmpzexrw0oCulyi0PDw/G  
jBnD1K1TLl02cuRIoqKi7NPZAlusXr2arl27MmDAABolasSECROYO3cuERERhIaG21PiTpkYh  
ddff53qlasD0KZNG0aMGMH06dP5+OOP+eqrr3juuecYNmzYRW3q0KED+/btY9asWURGRtK9e3  
d690jBqVOnuOWWWwgLC+P6669n27ZtACQkJNC7d29atGjB6NGjyX74Mi4ujpYtW9rrzvfkSH6  
pfCdMmMDatWuxWCxMnTqVHTt2EBERgcViISwsjNjYWCcd+Rz76/QalVJlKvHb/aT9fd6pdXpd  
U4EqNzcustyDDz5IWFgYtZ311MN1b9261V27dhEQEECjRo0YPXo0Gzdu5J133uG9997j7bfffZ  
seOHbRt2zbXeuHh4cyePZt///vfrFu3jv79+3PbbbcRFxdnL3PhwgVWrlzJiy++yPHjx9myZQ  
vbtm0jICCA8ePH07plaxYtWsRPP/3E8OHDiYmJ4YUXXqBTp05MmjSJ7777jk8++aTIfcgvle9  
rr73GG2+8wdKlSWEYP348D31104AAAB9ZSURBVD/8MMOGDSMTLY3MzEyHj5Gj9ApdKeU0lSpV  
Yvjw4bz77rsOr9OuXTvq1KmDt7c3jRs3pnfv3gCEhobmCs7FsX//fiwWCx07duSmm27ixhtvB  
KBXr14EBAQAsG7d0v71r38B0L17dxISEjhz5gxr1qzhrrvuAuCmm26yD9RRkPxS+fr5+V1Urn  
379rzyyitMmTKFAwc04OvrW6J9K4xeoSvLYhy5ki5NjzzyCG3atOHuu++2z/Pw8LB/MZmVlUV  
aWpp9Wc70uG5ubvZpNzc3e193SEgI0dHRd0/e3V420jqafila5NuG7D70vCpUqFDi/cq5D8BF  
KYCLcuedd3Ldddfx3Xff0a9fPz766KNc++MMeoWulHKqgIAA7rjjjlxdfYGBgURHRwOwZMkS0  
tPTi1XnU089xdNPP01CQgIAMTExzJo1i7Fjx5a4nZ07d2bu3LmAtS+/evXqVKpUiS5duti/zP  
z+++85ffo0ALVq1SI+Pp6EhARSU1PtXSkFpfL19/fn7Nmz9u39+eefNGrUiIceeogBAwbY++y  
dSa/QlVJO9/jjjzNt2jT79L333suAAQNo1aoVffv2LfaVcmRkJEEOHKFDhw6ICP7+/syZM4c6  
deqUuI2TJ09mlKhRhIWF4efnx+zZswF4/vnnGTp0KC1atKBDhw724TI9PT2ZNGkSERER1K1bN  
leK3/xS+YaFheHu7k6rVq0YOXIkqampfP7553h6elK7dm2eeeaZEre9IJo+VyKXoOlzXZOmz1  
VKqauUQwFdRPqKyB4R2SciEwooc4eI7BSRHSJyee+mV0opVXQfuoi4A9OBXsBhYJOILDHG7Mx  
RpgkwEehojDktIlfHeE9KKXUFceQKPQLYz4z50xiTBnwJDMhT5l5gujHmNIAx5uoZlVUppa4Q  
jgT0usChHNOHbfNyago0FZFfRORXEembX0UiMkZENovI5hMnTpSsxUoppfLlrc9FPYAmQDdgK  
PBfEamSt5AxZoYxJtwYE16jRg0nbVoppRQ4FtCPAPVzTNezzcvpMLDEGJNuJpKl2Is1wCulrh  
LZCbeyzZo1i3HjxpWorr1799KvXz+aNg1CmzZtu0000zh+/Pg113VE3kRi5YkjaX0T0EREgkT  
ECxgCLMlTzhHWq3NEpDrWLpg/ndhOpdRVIiUlhzTtuokHHniA2NhYtmzZwtixY8nbTzUrkeFw  
2YJcjjs6l1ORd7kYYzJEZBzWA+AOzDTG7BCRF4HNxpgltmW9RWQnKak8aYxJKM2GK6XKj2+//  
ZaXXnqJtLQ0qlWrxtY5c6lVqxY//wzDz/8MGAdIGPNmjUsWLCA9u3bc/PNN9vX79atG2C96v  
/mm284d+4cmZmZjBgxosCycXFx/Otf/+L8eWwmyWnTptGhQwdWr17Nc889R9WqVdm9ezd79ux  
h/Pjx/Pjjj9SvXx8vL6/LclBKguOP/htjlgHL8sybl001AR6z/SilytD3339vH/jBWWrXrm3P  
WFiQ5ORkLBA1ffrUqVNERkYC0KlTJ3799VdEhI8//pj//Oc/vPnm7zxxhtMnz6djh07cu7cO  
Xx8fNi+fftFqXJzypkC97HHHiuwbM2anfnxxx/x8fEhNjaWoUOHkv10+pYtW9i+ftTBQUF888  
037Nmzh507d3L8+HFCQkIYNWpUcQ/RFUFzuSilnMLX1zdXhsNZs2bZA+jhw4cZPHgWR48eJS0  
tjaCgIAA6duzIY489xrBhw7j111upV69ekdvJmQK3MOnp6YwbN46YmBjc3d3Zu3evfV1ERIS9  
DWvWrGHo0KG4u7tzzTXXOD0D4uWkAV0pF1PulXRZGD9+PI899hiRkZGsXr3aPtLPhAkTuOmmm  
1i2bBkd03bkx9+oEWLFvz8888F1pUzsVdhZadOnUqtWrXYunUrWVlZ+Pj45FuHK9FcLkqpUp  
eU1ETdutbHV7KzGoJ1IIRQ0FCefvpp2rVrx+7du7nzzjtZv3493333nb3cmjVr2L59+0X1F1Y  
2KSmJonXq4Obmxueff17gCEFdunRh/vz5ZGZmcvToUVatWuWs3b7sNKArpUrd5MmTuf3222nb  
tq19XFCA99+5YtWxIWFOanpyc33ngjvr6+LF261Pfee48mTzoQEhLC+++T37PrhRWduzYs  
cyePZtWrVqxe/fuAQ/KBw4caF93+PDhtG/fvtSOQ2nt9LlKuQBnn+uaNH2uUkdpdTsgK6WUi9  
CArpRSLkIDulJKuQgN6Eop5SI0oCullIvQgK6Ucoq86XPzExMTg4jwv//9r8iys2bn4u+//7Z  
Pjx49mp07dxayRSECAwPp3LlZrnkWi4WWLVuWqL68rpSUuxrQlVKXzbx58+jUqRPz5s0rsmze  
gP7xxx8TEhJS4m2fPXuWQ4esg6/t2rWrxPU4mzNT+GpAV0o51dGjR+nSpYv9Cnjt2rUAGGNys  
GABS2bn4scffYqLJcW+zpQpUwGNDaVVq1ZMmDCBqKgoNm/ezLBhw7BYLCQnJ9OtWzc2b97Mhx  
9+yJNPPmlfN+dAGnPMzCEiIgKLxcJ9992X63H/O+64g/nz5wPWPYxDhw61L8vMzOTJJ5+kXbt  
2hIWF8dFHHWgwevVqunbtyoABA2jUqBETJkxg7ty5REREBoayv79++11rFixgvDwcJo2bcrS

pUuLrLdz585ERkZe0h+pvDQ5l1IuZu/ef3P2nHOvQP0rBtO06XM0lf3i1y/o06cPzz77LJmZm  
Vy4cAGA9evXExQUROPGjenWrRvfffcgdgWYN4vvvv2fx4sX89ttv+Pn5cerUKQICAgp2bRpvvP  
EG4eG5H4ocNGGQ7du35/XXXwdg/vz5PPvss+zatYv58+fzyy+/4OnpydixY5k7dy7Dhw+3r3f  
33XfzxBNP8O233zJ37lw+//xzAD755BMqV67Mpk2bSE1NpWPHjvTu3RuArVu3smvXLgICAmjU  
qBGjR49m48aNVPP007z33nu8/fbbgDX/+saNG9m/fz833HAD+/bt47PPPiuw3pwpfJ1FA7pSy  
qnatWvHqFGjSE9P55ZbbrHnSJ83bx5DhgwBYMiQIXz22WcMGjSIFStWcPfdd+Pn5wdQZGrcGj  
Vq0KhRI3799VeaNGnC7t276dixI9OnTyc6Opp27doBlvzsNWvWtK9XrVolqlatypdffklwcLB  
9ewDLly9n27Zt9n7wpKQkYmNj8fLyol27dtSpUweAxo0b2wNyaGhorkRed9xxB25ubjRp0oRG  
jRqxe/fuQuvNmcLXWRwK6CLSF3gH64hFHxtjXiug3CAGcmhnjNFELUqVAUevpEtLly5dWLNmD  
d999x0jR4605zv/+uuvWbx4MS+//DLGGBISEjh79myJtjFkyBC++uormjdvsCBAXERjDGMGD  
GCV199tcD1Bg8ezIMPPsisWbNyzTfG8N5779GnT59c81evXo23t7d92s3NzT7t5uaWq/9bRHK  
tm92mquotjRS+Rfahi4g7MB24EQgBhorIRZ0+IuIPPAz85uxGKqXKjwMHDlCrVi3uvfdeRo8e  
zZYtWli5ciVhYWEcOnSIuLg4Dhw4wKBGgli4cCG9evXi008/tXfNnDplCgB/f/8CA/7AgQNzV  
Hhxrqv+Hj16EBUVRXx8vL2eAwcOXLTeU089dVGA7dOnDx988AHp6emAdeDp7KhrHLVgWQKysr  
LYv38/f/75J82aNXNKvcXhyBV6BLDPGPMngIh8CQwA8t4/9G9gCvAkSqmrLurVq3n99dfx9PS  
kYsWKfPbZZ7z44osMHDgwV7lBgwbxwQcf8P333xMTE0N4eDheXl7069ePVl55hZEjR3L//ffj  
6+vLhg0bcqlbtWpVgoOD2blzJxEREQCEhITw0ksv0bt3b7KysvD09GT69Ok0bNjQvp6/vz9PP  
/30RW0ePXo0cXfxtGnTBmMMNWruYNGiRcXa7wYNGhAREcGZM2f48MMP8fHxcUq9xVfk+lwRuQ  
3oa4wZbZv+F3CdMWZcjjJtgGeNMYNEZDXWRH5dLiIyBhgD0KBBg7Z5/3oqpUpG0+e6psuePlD  
E3IC3gMeLKmuMmWGMCTfGhOeXrF4ppVTJORLQjwDl1c0zXs83L5g+0BFaLSBxwPbBERPL9C6KU  
Uqp00BLQNwFNRCRIRLyAlcCS7IXGmCRjTHVjTKAxJhD4FYjUu1yUUuryKjKgG2MygHHAD8Au4  
CtjzA4ReVFEiku7gUoppRzj0H3oxphlWLi88yYVULbbpTdLKaVUCWkuF6WUchEa0JVSTuHu7o  
7FYrH/vPZavg+UA7Bo0aJcqXANtZrEihUrLrkNiYmJvP/++8Veb/LkybzxhsA/Prrr1x33XV  
YLBaCg4OZPHlyoeuXr2a/v37l6S5Tqe5XJRSTuHr60tMTIxDZRctWkT//v3tmQZffPFFp7Qh  
O6CPHTu2xHWMGDGCr776ilatWpGZmcmePXuc0rZsGRkZeHiUTujVK3SlVKmaMGECISEhhIWF8  
cQTT7B+/XqWLFnck08+icViYf/+//bkGiAgMDGTixIlyLBbCw8PZsmULffr0oXHjxnx44YcAnD  
t3jh49etCmTrtCQ0NZvHixfVv79+/HYrHYU+y+/vrr9vSlzz//vLldL7/8Mk2bNqVtP065gnZ  
8fLw9GZe7u7v9j87GjRtp3749rVu3pkOHDvkG+oLKzJoli8jISLp3706PHj0YPnx4ridGhw0b  
Zt+HS6FX6Eq5m0diD7P9XLJT62xZ0Zd/N6lXaJnk5GR7ZkWAiRMn0rNnTxYuXMju3bsRERITE  
6lSpQqRkZH079+f2267Ld+6GjRoQEXMDI8++igjR47kl19+ISUlhzYTW3L//ffj4+PDwoULqV  
SpEidPnuT6668nMjKS1157je3bt9v/Uli+fDmxsbFs3LgRYwyRkZGsWbOGChUq8OWXXxITE0N  
GRgZt2rShbdu2ADz66KM0a9aMbt260bdvX0aMGIGPjw/Nmzdn7dqleHh4sGLFCp555hm+/vrr  
XO0urMyWLVvYtm0bAQEB/Pzzz0yDopVbbrmFpKQklq9fz+zZs0t8frJpQFdKOUV+XS4ZGRn4+  
Phwzz330L9/f4f7miMjrXdEh4AGcu7cOfz9/fH398fb25vExEQqVKjAM888w5o1a3Bzc+PIkS  
McP378onqWL1/O8uXLad26NWC9so+NjeXs2bMMHDjQnkI3e3tg7c8fNmWYy5cv54svvmDevHm  
sXr2apKQkRowYQWxsLCJiT7iVU2FlevXqZU8N3LVrV8aOHcuJEyf4+uuvGTRokFO6YTSgK+Vi  
irQsVpw8PDzYuHEjKleuJCoqimnTpvHTTz8VuV7OFLV509dmZGQwd+5cTpW4QXR0NJ6engQGB  
uYaASmbMYaJEYdy33335ZqfPShFQr03bswDDzZAvffeS40aNUhISOC5557jhhtuYOHChcTFxd  
GtW7eLliustN50ucOHD2fOnDl8+eWxfPrpp0UdEodoH7pSqtSc03eOpKQk+vXrx9SpU9m6dSt  
QeGpcRyQlJVGzZk08PT1ZtWqVPu1u3nr79OnDzJkzOXfuHABHjhwhPj6eLl26sGjRIpKtkz17  
9izffvutfZ3vvvu07KSFsbGxuLu7U6VKFZKSkqhbtY7ARfnUc7arqDLZRo4caf/D4qxh6PQKX  
SnlfHn70Pv27cvDDz/MgAEDSELJwRjDW2+9BVgHqLj33nt599137V+GFsewYc04+eabCQ0NJT  
w8nObNmwpWUYk6duxIy5YtufHGG3n99dfZtWsX7du3B6BixYrMmTOHNM3aMHjwYFq1akXNmjX  
toxwBfP755zz66KP4+fnh4eHB3LlzcXd356mnmLEiBG89NjL3HTTTFm2y5Ey2WrVqkVwcDC3  
3HJLsfe/IEWmzy0t4eHhZvNmTfeildNo+tzY58KFC4SGhrJlyxYqV66cb5nLnj5XKaVU8axYs  
YLg4GDGjx9fYDAvCelyUUqpy6xnx54XDY/nDHqFrPRLkIDulJKuQgN6Eop5SI0oCullItwKK  
CLSF8R2SMi+0rkQj7LHxORnSKyTURWikhD5zdVKXUlK076XIB+/fqRmJjolJS3yqrIulxExB2  
YDvQCDgObRGSJMWZnjmK/A+HGmAsi8gDwH2BwaTRYKXVlKk76XIBly6yDoMXFvVlyl1t15cgV  
egSwzXjzpzEmDfgSGJCzgDFmlTHmgm3yV+DKSSahlCozSULJNGvWzJ5GdujQofz3v/8FrGlyT  
5486ZSut8rKkfvQ6wKHckwfBq4rpPw9pF5LRCRMcAYsKbHVEo53wvf7mDn32ecWmfINZV4/u  
YWhZbJL33u4MGDMtZtGiNHjuThhx/m9OnT3HvvvbnWc0bKW2Xl1AeLROQuIBzomt9yY8wMYAZ  
YH/135raVumWroC6XXr16sWDBAh588EF7cq7C1CtLrbJyJKAfAernmK5nm5eLiPQEngW6GmNS  
ndM8pVRxFXUlfbllZWWxa9cu/Pz8OH36NPXqFd4jW9KUT8qxPvRNQBMRCRIRL2AIsCRnARFpD  
XwERBpj4p3fTKVUeTVl6lSCg4P54osvPvuuy8aGMIZKW+VVZFX6MaYDBEZB/wAuAMzjTE7RO  
RFYLMxZgnwOLARWCAiAAeNMfr/kFJXkfzS59599918/PHhNy4EX9/f7p06cJLL73ECy+8YC/  
njJS3ykrT5yrlAjR9rmvS9LlKKXWV0oCulFIuQgO6Ui6irLpPvekoyfnUgK6UC/Dx8SEhIUGD  
uoswxpCqkICPj0+xltMRI5RyAfXqlPw4c0cOHGirJuinMTHx6fIe/bz0oCulAvw9PQkKCiOr  
JuhypH2uSillIvQgK6UU5iCA7pSSrkIDehKKeUiNKArpZSL0ICulFIuQgO6Ukq5CA3oSinlIj



SgK6WUiyiXT4rGp6YTc/YCJ9MycDdZxO/dRXBQEBE8/ch0cyPu3AVq+PrQr4ofXx07RZdq1Ti  
UlsnBlHRSki/Q9uxJfK5tTrOKPuw+m8zvx+K57Zrq7N+zG2nWguTkZDLFjeiTp7k1sC7bT5yi  
ggcn15o0Fmd6EpTwNxe8/Zh9PouR/h5gDKEN6rP7fAoNvdzx8vBgS9wB4t08aVenJtdVrsiKh  
CTOJZzENY2FDV7+jKhdlZVnUuhbvTK7zifzx9lk2lWuwMHkVIwIYRV9SUjPoImfD8dTUjmdlk  
HDir58unQpdfZso/ujE111+Cgd/H3ZmAF/p2ZwbUVfgny9Sc/KJMjLgZQ3dzKB46nprD99jv6  
VvPg7Pp4K/pU4lZrGtqRzDKxemV/EiwA36F2tMuuOHMWvWglqenny6ZETtKzgQxV3N8TdneTM  
LEhPp92ZeColac6mxLP4HDtM1J79vHzLzfxw9ASeGeksPpNKiwrehFSphI+7G0dT0kjatoXAl  
mGkennTys+bzafP4ObhQQ1vb46npVPH25NT6Rn4HdzPnkolaBZQmUZ+PpzNyGDjqTO09Pflr/  
OpfB9/mnsa1KaCpzs1fLzZGX+S+gFVycrIwM/Tg2927uHaBg1xy8zg3IE/CQsNo6GPF55uwrF  
z59mQeAE3L0/6VanAydQ0MpIv4FapCicTTuJTPsQn0zOp7uVBcmYWe86nICaLBm6G+HPniahf  
l62JZzh8/DidGtSjgo8PNb08STeG5NRULqSkcNTNkxYVfPnl72Ngsjhu3PBYPExpWqoSvu1DPw  
w0PN3c8PNz5M/4Ea08m0cjLnfnVAvBxc6OShzN3LJw9/Vjz/kUMo3hBn9vlh8+Roq3H82Sk8  
ioVRcPgZQsQ1pWfKdS08lMTaVKEgp9GtZj04GDJAFUZPWxE3Rwy+B0UhIXKlWlfs2a1PH2pJK  
b8PvhI6QnJl07kj+JFSvTJ8CfvedTSAGOpKQT7OdN7PlkrjEZ1KxQgYOJiaR6+eBzKp5zaem4  
17yGoCr+1Pby5K1tu7m9WWMuZGQiIny37y+GBtalUUAVlp0+i4+bG9f4eHIGYOY3UzEwS4/Zzb  
dPm7E9OpUOViqt+FUv8hWR2Va5BtzolCRM4GX+Cw94V6FOzKlvOXKCiuxtpxuAhQoCnBx4Ylp  
1IonEFHxLSM6jqJtT08qSSu3DwyBF2x+7lfIVKeNWui3fSKYKaNsOkJPNXehbxmXBTRhKb3Xz  
pVKs68RlZVPPyIOZUEmnuHtT38cLXzY1anm58d+BvEj19GFmvBrsupBBRuQLxARkcPp9Mgwq+  
HEtNp001P9INpGVlMe1gPNW9POhXozK/Jp7nVMJJ2vv7csDdm/jflnJzz15sT06nf53q2AYDc  
iqHBrqQkb7AOlhHLPrYGPnanuXewGdAWyABGGyMiSuszpIOcHH61CmCtx4s9npKKXWlGHP2CC  
9G3lSidS9pgAsRcQemAzccIcBQEQnJU+we4LQx5lpgKjClRC11QOdNe0qraqWUuixm+NctlXo  
d6XKJAPYZY/4EEJEvgQHAzhxlBgCTba+jgGkiIqYUcnm22PMHG5IaOrtapZS6bDLr+ZVKvY4E  
9LrAoRzTh4HrCipjG1Q6CagGnMxZSETGAGMAGjRoUKIGH6cksL5i4wKXe5gMMuTi3RIMBrnod  
X58s1JIdnM8D7GXSSNNvBwun5OPSSVFvPndJlJ/HlbOPEuieyUA3EwWWVK877LdTSaZ415omS  
qZSSS6V75ovrdJJBWA9nmbNFKL2O/C9i9bfuejQtYFzruVzpv+U1TJTOKMmz9Z4kblzL0kuHm  
RKt5FvqcK4oYhqwTrAdTMOEm8R/WL5vtlJXPBzbdEdTqqYtZ5zrlVAKzvETdjivWZuVSOHG8P  
k4HBDQO5PjN5j3lJ3muFfs7+KVPw5+O2k78CnYu1TUdc1i9FjTEzqBlg7UMvSR1PPjWRJ53aK  
qWUutx61EqztJlZqHQHq55iuZ5uXbXkR8QAqY/1yVCml1GXiSEdFBDQRkSAR8QKGAEvylFkCjL  
C9vg34qTT6z5VSShWsyC4XW5/40OAHrLctzjTG7BCRF4HNxpglwCfA5yKyDziFNegrpZS6jBz  
qQzfGLAOW5Zk3KcfrFOB25zZNKaVUceij/0op5SI0oCullIvQgK6UU5iCA7pSSrkIh5JzlcqG  
RU4AB0q4enXyPIV6FdB9vjroPl8dLmWfGxpjauS3oMwC+qUQkc0FZRtzVbrPVwfd56tDae2zd  
rkoPZSL0ICulFIuorwG9Bll3YAyoPt8ddB9vjQYy6Xyz50pZRSFyuvV+hKKAxy0ICulFIuot  
wFdBHpkYJ7RGSfiEwo6/aUliJUF5FViRjTRHaIyMO2+QEi8qOIxNp+V7XNFxF517bf20skTY6  
6RtjKx4rIiIK2eaUQEXcR+VlEltqmg0TkN9u+zbelaUZEvg3T+2zLA3PUMde2f4+I9CmbPXGM  
iFQRkSgR2S0iu0SkvaufZxF51Pa+3i4i80Tex9XOs4jMFJF4EdmeY57TzquItBWRP2zrvCsiR  
Q9tZYwpNz9Y0/fuBxoBXsBWIksS21XCfakDtLG99gf2Yh2E+z/ABNv8CcAU2+t+wPeANcDv9  
nmBwB/2n5Xtb2uWtb7V8S+PwZ8ASy1TX8FDLG9/hB4wPZ6LPCh7fUQYL7tdYjt3HsDQbb3hHt  
Z71ch+zsbGG177QVUceXzjHVIyr8A3xznd6SrnWegC9AG2J5jntPOK7DRV1Zs695YZJvK+qAU  
8wC2B37IMT0RmFjW7XLSvi0GegF7gDq2eXWAPbbXHwFdc5TfY1s+FPgox/xc5a60H6wjXq0Eu  
gNLbW/Wk4BH3nOMNQd/e9trD1s5yXvec5a70n6wj71F7YbEPKePlcz8/wzxnCA7bwtBfq44n  
kGAVMedKecV9uy3Tnm5ypX0E9563LJb8DqumXUFqex/YvZGvgNqGWMOWpbdAyoZXtd0L6Xt2P  
yNvAUkGWbrgYkGmMybNM5259r8HEge/Dx8rTPQcAJ4FNbN9PHI1IBFz7PxpjgWbVaQeAo1vMW  
jWuf52zOOq91ba/zzi9UeQvoLkdEKgJfA48YY87kXGasf5pd5r5SEekPxBtjosu6LZeRB9Z/y  
z8wxrQGzmP9V9zOBc9zVWAA1j9mlwAVgL5l2qgyUBntbwFdeGrC43RMQTazCfa4z5xjb7uI  
jUsS2vA8Tb5he07+XpmHQEIkuKdvgSa7fLO0AVsQ4uDrnbX9Dg4+Vpnw8Dh40xv9mmo7AGeFc  
+zz2Bv4wxJ4wx6cA3WM+9K5/nbM46r0dsr/POL1R5C+iODFhdLti+sf4E2GWMeSvHopwDbo/A  
2reePX+47dvy64Ek2792PwC9RaSq7cqot23eFccYM9EYU88YE4j13P1kjBkGrMI6uDhcvM/5D  
T6+BBhiuzsiCGiC9QukK44x5hhwSESa2Wb1AHbiwucZa1fL9SLiZ3ufZ++zy57nHJxyXm3Lzo  
jI9bzjODxHXQUR6y8VSvAlRD+sd4TsB54t6/Zcwn50wvrv2DYgxvbTD2vf4UogFlgBBNjKCzD  
dt9/AOE56hoF7LP93F3W++bg/nfjn7tcGmH9oO4DFgDetvk+tul9tuWNcqz/rO1Y7MGBb//L  
eF8twGbbuV6E9W4G1z7PwAvAbmA78DnW01Vc6jwD87B+R5CO9T+xe5x5XoFw2/HbD0wjzxf+  
f3oo/9KKeUiyluXi1JKqQJoQFdKKRehAV0ppvYEBnS1lHIRGtCVUspFaEBXSikXoQFdKaVcxP  
8DpFzw1D66tHYAAAAASUVORK5CYII=\n"

```
    },  
    "metadata": {  
      "needs_background": "light"  
    }  
  }  
}
```

```

    ],
    {
      "cell_type": "markdown",
      "source": [
        "# 3.3 Multi-Variant Analysis"
      ],
      "metadata": {
        "id": "K_tvzmkFHpq3"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "sns.lmplot(\"Age\", \"NumOfProducts\", df, hue=\"NumOfProducts\",
fit_reg=False);\n"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 424
        },
        "id": "_EpgOdbbyHw_M",
        "outputId": "0727a483-2e22-49a1-c8f9-dd1ae3e6556a"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stderr",
          "text": [
            "/usr/local/lib/python3.7/dist-
packages/seaborn/_decorators.py:43: FutureWarning: Pass the following
variables as keyword args: x, y, data. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.\n",
            "FutureWarning\n"
          ]
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 446x360 with 1 Axes>"
            ],
            "image/png":
      "iVBORw0KGgoAAAANSUhEUgAAABEAAAFgCAYAAAAA1skc7AAAABHNCSVQICAgIFAhkiAAAAAlw
      SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
      jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAE1EQVR4nO3de5hddX3v8fdn75nJZD
      IJuQ2KSSAgKAgijyIijSl9RS5aZVWqFJQLFZpxXrpsZ5HRTz2oBxb8YLCAQripVqviIhSxQa
      OFTpEFELQoxAkMZKEXMhkkszM3t/zx1oz2bPntmdm39bk8+KZZ/a67LW+e++V+bDW+u3fTxGB
      mZlZFuUaXYCZmdl0OcTMzCyzHGJmZpZDZjEzM8ssh5iZmWVWS6MLmKrTTz897rjjjkaXYWY2H
      jW6gANJ5s7Etm7d2ugSzMysSWQuxMzMzIY4xMzMLLMcYmZml1kOMTMzyyyHmJmZZZZDzMzMMs
      shZmZmmeUQMzOzzHKImZlZztW82ylJeaAH2BgRZ5UtmwN8HjgReAp4XUSsr3VNzWTX6tVsu+F
      GBjZsQPPmIYliby+ty5cz96ST2HPffQxs2EDr8uUUIth3//1QKEA+T+cZZ7Diqo+N2s5Yz118
      8ZuYf+qpADzxnr+n9/bbk+1I5BcvJtfePvk+y7Q85zm0LVo0vg7/9u0M/upX+1fI5UAiN28ed
      HZS/N3v9i961rOgt5fi7t2Qz0MEFivk5s2j4w/+gOLmzRXVQGcnuVyO4u7dqK2N3KJF5KSpvQ
      dl7yXA5s9cw/abb062m88TJfUtuvBCDr70bW0vWlZD6T6rqfzzrtV+zJqdaJ2ys6R3At3AgjF

```

C7G3A8RHx15LOA/40I1430fa6u7ujp6endgXX0a7Vq3nyig+jtjaKAwMMbtoEEbQsexaxr5/C  
1q3ku5bSsngJ+x5/HHbvHrWNzrPPZuHZZw1vR+3tDD711IjnxT69RH8/z/jA+9nxdvo/c53R  
m1HCxaQ65hLYctW8kuX0rJkCfvWr4e+volfxJw5zHn2s9n3m9/Avn1TfxPy+f3h1IYehQJaeB  
Btz1o27usepbUFBgaTzSxdilryI15LJe9B59lNs+Kqj7H5M9fw1DXXJLVEjKwvl4MIlrztbRx  
86dtGritG1NCyYMHwPqsZMKXHjdrbR7w2B1lTcN+JdVTTY4mSlgNnAtePs8qrgJvTx18D/kjS  
AXMAbLvhuT/30fOpbhtG+RyKJ+n+NQ2orcXJGJXL5JG/iGXkh+g9/bbR2xH0qjn5ubORW1tb  
LvhuTsY2gbJeLpp4ld6fN6031OFmAA+/Y1604nwGDk2VWxmIQGEE/vGv26J9xOcfglFbdtG/  
Vaxn0Pyt5LgO033wwSamkZVZ9aWkBK1hm17sgaSvdZTeWfd632Y5YFtb4n9gng74HiOMuXAU8  
ARMQgsBNYUr6SpEsk9Ujq2bJlS61qrBuBDRtQezsAxf7+5A9gLkf09yft+TzR3z/xRgqFEdsZ  
3lbZc9XezsCGDWNfkkfT+rziZPuspaErA8XxDplJnpC+d6zXMul7kM4fvsQ5nnw+Wad83bIaR  
uyziso/71rtxywLahZiks4CNkfe/TPdVkrCfXhEdHd1dVvhegaQ+vy5cTevQDk2tqG7wuprS  
2ZLhRQW9vEG8nnR2xneFtlz429e2ldvznCP85Kn5ebbJ+1NHSgmJvioVl6ZpnLjflaJn0P0vm  
5efMmDHsKhWSd8nXLahixzyoq/7xrtR+zLKjlmDjLgHMkrQf+FThN0hfK1tkIrACQ1A1cRNLA  
44Cw+OI3JWdde/aQW7w4OYMoFMgtWYw6OyECze9MGhWkfzSBJOzS/+vvPOOMEduJiFHPLe7ZQ  
/T3s/jiN9F5xhn7t1FCCxag+enzOtN9dnRM/iLmzEnWnTNnem9CaaAM3RMDtGD+6Nc94XZyw6  
8pt3jxqNcy7ntQ914CLLrwQoggBgDh1ReDgxCRrDNq3ZE1lO6zmsO/71rtxywLat6wA0DSKuD  
dYzTsuBR4fknDjtdExJ9PtK3Z1LAD3DrRrRonx60Tm9oBc1+/GdQ9xCRdAfREXK2S2oFbgBcC  
24DzIuLRibY120LMzGYdhlgdlSXEqsKhZmZnziFWR+6xw8zMMsshZmZmmeUQMzOzzHKImZlZZ  
jnEzMwssxxiZmaWWQ4xMzPLLIeYmZl1lkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlmOcTMzCyzHG  
JmZpZDZjEzM8ssh5iZmWWWQ8zMzDLLIWZmZpnLEDmzs8xyiJmZWWY5xMzMLLMcYmZml1kOMTM  
zyyyHmJmZzzZDzMzMMsshZmZmmeUQMzOzzHKImZlZZjnEzMwssxxiZmaWWQ4xMzPLLIeYmZl1  
lkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlmOcTMzCyzHGJmZpZZNQsxSe2S7pP0c0lrJX1ojHUuk  
rRF0gPpz5trVY+Zmc0+LTxc9j7gtIjoldQK3CPpexHx07L1vhIRf1PDOszMbJaqWYhFRAC96W  
Rr+hO12p+ZmR14anpPTfJe0gPAZuD0iLh3jNveK+kXkr4macU4271EUo+kn1lbtTSyZDMzy5C  
ahlhEFCLiBGA5cJKk48pW+Q6wMiKOB+4Ebh5n09dFRHdEdHd1ddWyZDMzy5C6tE6MiB3AXcDp  
ZfOfioh96eTlwIn1qMfMzGaHWrZ07JK0MH08F3gF8EjZOoeUTJ4DrKtVPWZmNvvUsnXiIcDNk  
vIkYfnViLhN0hVAT0TcCrxd0jnAILANuKiG9ZiZ2SyjpBFhdnR3d0dPT0+jyzAzG48aXcCBxD  
12mJlZZjnEzMwssxxiZmaWWQ4xMzPLLIeYmZl1lkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlmOcT  
MzCyzHGJmZpZDZjEzM8ssh5iZmWWWQ8zMzDLLIWZmZpnLEDmzs8xyiJmZWWY5xMzMLLMcYmZm  
l1kOMTMzyyyHmJmZzzZDzMzMMsshZmZmmeUQMzOzzHKImZlZZjnEzMwssxxiZmaWWQ4xMzPLLI  
eYmZl1lkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlmOcTMzCyzHGJmZpZDZjEzM8ssh5iZmVWwZU  
JMUruk+yT9XNJASR8aY505kr4i6deS7pW0slb1mJnZ7NNSw23vA06LiF5JrcA9kr4XET8tWed  
iYHtEHCnpPOCjwOtqWNOE7t5wNzetvYmNvRvZM7CHHF07KEaRvPKcvvJ0rjz1yuF1P/vAZ7l1  
3S30DfQRErQpDi876qCjWDR3UbKd/j3sGni/HYUYZLCienLkRmx3PHnlOWLBefx+z+/pG+iJo  
7WDXf27pv4GNIG88ixOxcD2/u0z3taitkUctfgoNvZuZFnnMrrmdrF642r6BvooRGFK2xIipx  
wdrRlccMwFvPWEtw4vKz0W2vJtLJqzCEks61xGRLBm85rh/bWohSDoaO3g6EVH88j2R4Y/s/L  
p0v28d/V7uWP9HRSiMOp4LDlul3Uuo/sZ3fQ82TPu9EXHXsTL1798Ru9t+T6rsU2z6VBE1H4n  
UgdwD/DWiLi3ZP73gcsj4j8ltQC/B7pigqK6u7ujp6en6jXeVeFu/vHef6Q138rW3VvZNTg6B  
M48/EyuPPVKPvVAZ7n2F9ciicHi2IE0JzeH1lwrvY09Va91InNyc9hX3FfXfTazHDmeu/i5bN  
q9iR37dlT8Pwbjacm1EBG85fi38NYT3jriWCBgMJLjYemcpfQN9tFX6BulDaX/FSkiRFuujYH  
iWHBdc3JzKFAY3s/jTz/Odx/77qjtnHn4mZx5xJnDx217vp1te7exdc9Wls5dyuL2xaOm9xb2  
MlAY4H0vft+0Q6f030p7vr0q25x11ogCDiQ1vScmKS/pAWAzCGdpgKWWAU8ARMQgsBNYUsuax  
nPT2ptozbcyt2XuqABTekzesf4OAG5ZdwuSaNH4J7L7iivvqHmBA8sfUhhUpIomn+5+uyvZa1I  
Ikb113CzDyWChGcfhY2da/bcWAAwj2n7kHgaQRwTq0vaH9DB13KvkPkuOx9LiVxK7+XRP+nts  
yl9Z8KzetvWna70H5PquxTbPpqqmIRUqhIk4AlgMnSTpuOtuRdImkHkk9W7ZsqW6RqY29G2nP  
t0+4ztAlOb6BPvLka1KH1UYxkpCYyVnYkDx5+gaSgCo9FoL9FxCg91eN/Yx36bMQhVHH7UBxg  
Dx5BooDY04DtOfb2di7cdpljfVvZabbNJuuurROjIgdwF3A6WWLNgIrANLLiQcBT43x/Osioj  
siuru6umps47LOZewt7J1wnbySP1YdrR0UmNo9FWusnJJDPVeFQ75AgY7WDMdksaCSq0hD+6v  
GfoaOu3J55Ucdt625VgoUaM21jjkNsLewl2Wdy6Zdl1j/Vma6TbPpqqMxRxC5JC9PHc4FXAI+U  
rXYrcGH6+FzgRxPdD6uli469iIHCAHsG9zC/Zf6IZUP/h336yiSDLzjmAiJi+P7HWObk5tDZ0  
lm7gsfRoLevaeXIERESAftQle0NxiARwQXHXACMPBZyyg0fK4vbFtOR7xhzG0LDYSperIwI16  
HtDe1n6LiLkv8gOR5Lj9uIYH7b/Al/7xncw0BhgIuOvWja70H5PquxTbPpqlnDDknHAzcDeZK  
w/GpEXCHpCqAnIm6V1A7cArwQ2AacFxpGPTTrTdWjXsALdObAZunejWibOAb0zXUV1aJ1ZTLUPM  
zKwKHGJ15B47zMWssxxiZmaWWQ4xMzPLLIeYmZl1lkPMzMwyyq6IQk/RsSXPSx6skvX3oO2BmZ  
maNUumZ2NeBgqQjgetIetn4Us2qMjMzq0ClIVZMO+j9U+BTEfEe4JDalWVmZja5SkNsQNL5JF  
1E3ZbOa51gftMzs5qrNMTECLwE+EhEPCbpcJLuoszMzBqm0pGdXxERbx+aSINs4i7fzczMaqz  
SM7ELx5h3URXrMDMzm7Ijz8TS+2B/ARwu6daSRfNJep03MzNrmMkuJ/4E2AQsBT5eMn8X8Ita  
FWVmZlaJCUMsih4HHpf0euB3EbEXhge5XA6sr3mFZmZm46j0nthXYcTojAXg36pfjpmZweUqD

bGWiOgfmkgft9WmJDMzs8pUGmJbJJ0zNCHpVcDW2pRkZmZWmUq/J/bXwBclfZpk600ngL+sWV  
VmZmYVqCjEiU13wMmSotPp3ppWZWmVoGKQkzSB8qmAYiIK2pQk5mZWUuqvZy4u+RxO3AWsK7  
65ZiZmVWuooYdEfHxkp+PAKuAI2pamZnZLCIpJH28ZPrdki6v4vYvkfRI+nOfpFNKlrlc0lpJ  
D0g6RtKe9PHDkj4nqdJGfmPtd72kpdN43kpJfzHd/Q6ZbuEdJF92NjOzyuWDXjOdP/iTkXQW8  
BbglIg4mqQx3pckPTNd5fXA/4qIE4A9wG/Sx8cDzwNeXba9Sq/SzcRkkm4NZ6SiEJP0oKRfpD  
9rgV8Cn5jpsz3MDiCDwHXA35UvkHSTpHNLpntV36sk/Yekb0t6VNKVkl6fnmk9KOnZ6VP+O/C  
eiNgKEBFrgJuBSyW9Gfhz4MOSvli633Sw458AR0q6SNKtkn4E/FDSYknfSv/u/1TS8WlNSyT9  
ID2zu56kxfrQmdVDJa9h+Ext0pGS/13SzyWtSeu+Enh5ekb4d5KOTV/XA+k+j6rkTa00bc8qe  
TwIPJm+eDMzq9xngF9I+tgUnvMC4BiSttcfBa6PiJMkXQb8LFAO4Fjg/rLn9QAXRsT700uLt0  
XElyStHFpBUgfwR8AHGcALwKOj4htkj4F/CwiXi3pNODzwAnAB4F7IuIKSWcCF1fwGr4IXBk  
R35TUTnIC9V7g3RFxVlrLp4CrI+KLktqAfCvVzmS92C9OH+4qW7RAEHhHnuzNzCoUEU9L+jzw  
dpLLepX4r4jYBCDPn8AP0vkPAN84zVKeLekBIIBvR8T3JF0E3Fnyd/0U4LVp3T9Kz8AWAKcCr  
0nnf1fS9o12JGk+sCwivpk+Z6gP3vJV/xP4H5KWA9+IiP9XyQuZ7Ezs/vRFCjgU2J4+Xgj8Fj  
i8kp2YmdmwTwBrgh8pmTdIensnbWRR2q3fvpLHxZLpIvv/hj8MnAj8qGTdE4G149QwdE+s3O4  
x5lVq+DWk2qfy5Ij4kqR7gTOB2yW9JSJ+NNnzJrwnFhGHR8QRwL8DZ0fE0ohYQnJ58QcTPdfM  
zEZLz3S+ysjLcOtJQgfgHKBlipv9GPBRsUsAJJlAMnDxNTMo9W6SBiFIWgVsJYingdWkDTIkV  
RJYlK7/JHBwesY2h/Q2VETsAjZiEnX6nDnpZcxdJGNTks4/Ang0Ij4JfJuk0cmkKr0ndnJE/N  
XQRHrqOZVrumZmtt/Hgb8pmf4/wLcl/Ry4gyMEUXERZKWAT+RFCQB8Yahy5DTdDlwo6RfAH3  
Ahen8DwFfThv5/YTkqhwRMSDpCuA+YCPwSMm2LgCuTzCpAH9GmiZlIX3NNwFzgAskDQC/B/6x  
kiIVEZOvJH2fJJW/kM56PXBqRPxJJTuppu7u7ujp6an3bs3MKjXqZo/VTqXfEzs6fAK+mf4cn  
M4zMzNrmEo7AN4GXJa2Mgl3AGxmZs2g0i87P1/Sz4CHgLWS7pd0XG1LMzMzm1illxOvBd4ZEY  
dFxGHAu0i+eW5mZtYwlyYbYvIi4a2giIn4MzKtJRWZmZhWqtIn9o5LeD9ySTr+BpPsTMzOzhqn  
0TOxNJK0TvwF8HViazjMzswOIpbSlbS7t7LeRjJ0Tk5Qn6cdqSn10SVpB0mHkM0i6rrouIq4u  
W2cVyTezH0tnfcOjRZuZnNbWbgE+T/H1vuElDLClKkoqSDoqInVPY9iDwrohYkzbNv1/SnRHxc  
Nl6dw/1YmxmZtWz8r3fPR14D0k/t48BV62/8sw7ZrLNIhFd2hN+o1V6ObEXeFSDSDZI+OfQz0R  
MiYlM6pslQ3l1nrgGUzK9fMzCqRBthngENIhnE5BPhMOn/WqLRhxzfsn2lJU/uFwL1jLH5J2nf  
W70jG1hnV67Kks4BLAA499NDplmFmdiB5D0mP933pdF/J/BmdjTWTsu6JvZqkUceDefH9qe5A  
UidJY5B3pd0gl1oDHBvYrvZLOAL4FjBrNMyKuI/leWnd39+SdPZqZ2eEkZ2Cl+phlQ2hNeDlR0  
jUkQ2kvIRna+v1T2bikVpIA+2JEjDqTi4inh7qwiojbgVZJS6eyDzMzG9NjQEfZvA72N6SbFS  
a7J3YqcFpE/AOWCnhlpRtWmmznDcC6iPincdZ5Zroekk5K63mq0n2Ymdm4riIZ3mQoyDrS6at  
mslFJXyYzhfm5kjZiuniy59TSZJcT+yOiABARfUObU6GXkYwh82A6DDbA+0hGiCYiPgeC7xV  
0iDJUN3nRSVjw5iZ2YTWX3nmHSvf+9lLqX7rxKYawWTC8cQk9QG/HpoEnp1oi6Q3+4pG3qwmj  
ydmZk3044nV0WRnYi8iOUMyMzNrOpOF2Jci4kWSbomIC+pSkZmZWYUmC7E2SX8BvFTSa8oXjt  
Xi0MzMrF4mC7G/Bl4PLATOLlsWzOALOGZmZjM1YyHfXDX3APZJ6IuKGOTvKZmZWkUp67DgYOEz  
S19JZa4HPRMTmmlZmZmZn5IRSuppsh47Xgb8F0mhn2d/1/v3pcvMzOzAMjRCyfOAK4FLJT2v  
UcVMdib2ceDVEfGzknm3SvomcC3w4ppVZmZmM3P5QaOGYuHynTP9svMmYFP6eJekoRFKyofZq  
ovJuplaUBZgAETEA8D82pRkZmYzlgTYqKFY0v1VMckIjXUxWYhJ0qIxZi6u4LlmZtY4Yw3Fsi  
+dP2OTjFBSN5MF0T8DP5D0B5Lmpz+rgO+ly8zMrDkdzv4AG1KVoVgmG6GkniZrYn+dpN8BHwa  
OTWevBf5nRHyn1sWZmdm0PUZyCbE0yGY8FESlI5TU06RN7CPiNuC2OtRiZmbVcxXJPTFIgqwq  
Q7Ewzgg16ZiQdTdpIAFIohz4W2Bl6Xmi4pZalGVmZjNy+c47uPygUuOXVKF14j00UU/9FYUY8  
C2S08fvAMXalWNmZlWTBNAmqvZVRpieyPikzWtxMzMbIoqDbGrJX0Q+AFJE00AImJNTaoyMz  
OrQKUhn9ySG3mnsf9yYqTTZmZmDVFPiP0ZcERE9NeyGDMzs6motNeNh0jGfDMZM2salZ6JLQQ  
ekfRfjLwn5ib2ZmYHEEntwGqS75y1AF+LiA82qp5KQ6xhBZqZWVPZB5wWEb1p91P3SPpeRPY0  
EcVUFGIR8R+1LsTMzKrr+Tc/f9RQLA9e+OBmv+wcQG862Zr+xEy2ORMV3ROTtEvS0+nPXkkFS  
Q3rtdjMzCaWBtiOOvJjS+TMiKZ92ObUZuDmimnYoFgAiYn5ELiIiBcBc4LXANTWtzmZmZqJmQ7  
FERCEiTgCWAydJom6m25yuKY8JFolVAX9Sg3rMzKw6ajYUy5CI2AHcBVRtoM2pqrQD4NeUTOa  
AbmBvTSoyM7NqqNVQLF3AQETskDQXeAXw0ZlscyYqbZ14dsnjQWA98KqqV2NmZtVSq6FYDgFu  
lpQnOan5ajpkV0MoaWiShd3d3dHT09PoMsZMxtM0w5TUonVis5kwXCR9YILnRkR8uPolTcwhZ  
mZNRmlC7EAw2eXE3WPMmwdcDCwB6h5iZmZmQyYMsYj4+NBjSfOBy4A3Av8KfHy855mZmdXDpA  
07JC0G3gm8HrgZeFFebK91YWZmZpOZMMQkXQW8BrgOeh5E9E60vpmZWt1NlrcjSPIN70FG9o0  
lkoYdC2pb3mhu2GFmTc4NO+posntiU+7Rw8zMZrf002I9wMaIOKuRtTikzMxsqi4D1jW6CKhh  
iElaIEkuSQ9LWivpsjHWkaRPSvqlpF9IelGt6jEzO9CsO/qY09cdfcwPlx19zKPP72r0YL8cO  
BO4fuYVzlyl3U5NxyDwrohYkzbPv1/SnRHxcMk6rwsOSn9eDHw2/V09ly8CilXdpNkBaW4X7N  
sGxCLYy6If9vXCnE446nTY9TvY8XhyN73vKRjcmYybvwKeWpdsJ5eHFS8DRbLuwsPgpZfBc14  
xdg2/uhN+cvXk61a63kz20eTSwPoMSbuG4aFY1h19zKXHPLJuJr12fAL4e2D+zKucuzqdiUXE  
pohYkz7eRXLquaxstVcBn097xv8psFDSIVUrwgFmVj17towdYEPL9u6EXGsSZA9+BTathf59s  
PO3MLAblEvW2fLQ/u0UC/D4atjwM2hfBLuehO+9OwmScr+6M1m268mJ1610vbHM5LnNp+pdSU  
g6C9gcEffPvLzqqMs9MUKrgRcC5QonLQOeKJnewOigmwEHmFlD5cRwQ+b+nbBna7pAEGUBqJJ

GfIO9yXRbB+TakjOhcj+50lnWljHxupWuN5aZPLf51GIolpcB50haT9LpxWmSvjCD7c1YzUNM  
UifwdeAdETGt0aAlXSKpRlLPli1bqlugmVXX0Nd2ojAyuCrtbLx1Luz47ej5Ox5Plk22bqXrj  
WUmz20+j5H0XF9qRkOxRMQ/RMTyiFgJnAf8KCLeMP0SZ66mISaplSTAvhgR3xhj1Y3AipLp5e  
m8ESLiuojojojurq6u2hRrZtUxdIalfPJTPn8yA3tg4aGj5y88LFk22bqVrjeWmTy3+VxFMvT  
KUJBVayiWp1LL1okCbgDWRCQ/jbParcBfpq0UTwZ2RsSm6lXhbxCY1VUxGP6ub9tBMHdpuiBG  
BhqMPDNr6Uym+/ug2J80pij30suSZf19E69b6Xpjmclzm0zaeONSyBOWOP0900YdwYLiX43+j  
hjUcDwxSacAdwMPsv/m1PuAQWEi4nNp0H2aZGjrPuCNETFhdxxT7rHDjTvMqmNarRN/m4RBRa  
0Tf5uc8VTUOnGSdStdbyb7GJ977KgJd4ppZlZdDrE68vU2MzPLLIeYmZl11kPMzMwyYyFmZma  
Z5RAzM7PMcoiZmVlmOcTMzCyzHGJmZpZzDjEzM8ssh5iZmWWWQ8zMzDILLIWZmZpnlEDMzs8xy  
iJmZWWY5xMzMMLLMcYmZml1kOMTMzyYyHmJmZzzZDzMzMMsshZmZmmeUQMzOzzHKImZlZZjnEz  
MwssxxiZmaWWQ4xMzPLLIeYmZl11kPMzMwyYyFmZmaZ5RAzM7PMcoiZmVlmOcTMzCyzHGJmZp  
ZzDjEzM8ssh5iZmWWWQ8zMzDILLIWZmZpnlEDMzs8xyiJmZWWbVLMQk3Shps6SHxlm+StJOSQ+  
kPx+oVS1mZjY7tdRw2zcBnwY+P8E6d0fEWTswczMzRganYlFxGpgW622b2Zmluh7Yi+R9HNJ  
35N07HgrSbpEUo+kn1lbtSszPjMza2KNDLE1wGER8QLgU8C3x1sxiQ6LiO6I607q6qpbgWZm1  
twaFmIR8XRE9KaPbwdaJS1tVD1mZpY9DQsxSc+UpPTxSWktTzWqHjMzy56atU6U9GVgFbBU0g  
bgg0ArQER8DjgXeKukQWAPcF5ERK3qMTOz2admIRYR50+y/NMkTfDNzMypmDgtE83MzKbNIWZ  
mZpnlEDMzs8xyiJmZWWY5xMzMMLLMcYmZml1kOMTMzyYyHmJmZzzZDzMzMMsshZmZmmeUQMzOz  
zHKImZlZZjnEzMwssxxiZmaWWQ4xMzPLLIeYmZl11kPMzMwyYyFmZmaZ5RAzM7PMcoiZmVlmO  
cTMzCyzHGJmZpZzDjEzM8ssh5iZmWWWQ8zMzDILLIWZmZpnlEDMzs8xyiJmZWWY5xMzMMLLMcYm  
Zml1kOMTMzyYyHmJmZzzZDzMzMMsshZmZmmeUQMzOzzHKImZlZZjnEzMwssxxiZmaWWS212rC  
kG4GzgM0RcdwYyVcDZWb9AEXRcSaatdxxHu/S7HaGzWzEzZ0tLBjb4FCMcjnxChz29i0q59C  
MRCwpLON9tY8KxZ18JiJfVofj27jie19rFjUwVtOPYJVRx8MwI8f2cy1qx8dXvbMBW388JEt7  
04vMK8tz5tPOZy3//Fzavpaymsorc+ajyKiNhuWTgV6gc+PE2JnAH9LEmIvBq60iBdPtt3u7u  
7o6empqAYHmFnzOKi9hY45eTbv6qers42lnXPYm1BgoBBccc6xAHzglrW05sXcljy/27GHbX0  
D5AsteVEMKAZcdtqRNQuYH9ekKnaAAhRSURBVD+yeUQNpfVNIchUk+JsTDW7nBgRq4FtE6zy  
KpKai4j4KbBQ0iHVRMEBZlZfmuDP9869gzy9Z5CcYNfeQSTR0dZCa15cu/pRr139KK35ZJ4kd  
uwZGH5uTj lacjlyguvveaxm9ZfXUFqfNadG3hNbBjxRMr0hnTeKpEsk9Ujq2bJlS12KM7Pq6y  
8UySn5PWRua54N2/t4Ynsfclvzw/OL6UWi0otFOcHu/kLN6iuvobQ+a06ZaNgREddFRHdEdHd  
ldTW6HDObprZ8jmIkv4fsGSiwfFEHKxZ1sGdgf0Dl0rO60r07YsC8tpEhU03lNZTWZ82pkSG2  
EVhRMr08nVclmUhos11kolvsB7W3sGBuC8WA+e0tRAR9/YMMFIK3nHoEbzn1CAYKybyIYOHcl  
uHnFqPIYLFIMEdNpxxes/rLayitz5pTI//O3wr8pRInAzsJYlMld/DolWc6yMzqYElHC/n01C  
mfE8sPmjM8LWBpZxsL5rayckkn1512Jicv7WTnngEOnt8+3Ghi1dEHc8U5x3Lw/HZ27hnguc9  
cwJ+ecAidc1oYLCaX9WrZqAMYVUNpfdacatk68cvAKmAp8CTwQaAVICI+lzax/zRwOkkt+zdG  
xKTNDqfSotHMRaHcOrGOavY9sYg4f5LlAVxaq/2bmdns56ttZmaWWQ4xMzPLLIeYmZl11kPMz  
MwyYyFmZmaZ5RAzM7PMcoiZmVlmOcTMzCyzatZjR6lI2gI8XuXNLgW2VnmbM9Fs9UDz1dRs9U  
Dz1eR6JleLmrZGxOlV3qaNI3MhVguSeiKiu9F1dGm2eqD5amq2eqD5anI9k2vGmmxqfDnRzMw  
yyyFmZmaZ5RBLXNfoAso0Wz3QfDU1Wz3QfDW5nslY002Bb4nZmZmmeUzMTMzyYyHmJmZzDYB  
FWKSVki6S9LDktZKuiydv1jSnZL+X/p7UR1rapd0n6SfpzV9KJ1/uKR7Jf1a0lcktdWrpnt/e  
Uk/k3Rbk9SzxTKDkh6Q1JPOa+Tnt1DS1yQ9ImmdpJc0qh5Jz03f16GfpyW9o5HvT1rX36XH9E  
OSvpwe6w07jiRdltayVtI70nknfY9s5g6oEAMGgXdfXPOak4FLJT0PeC/ww4g4Cvhh010v+4D  
TIuIFwAna6ZJOBj4K/HNEHALsBy6uY00AlwHrSqYbXQ/AH0bECSXf62nk53Y1cEdEHA28gOS9  
akg9EfHL9H05ATgr6AO+2ah6ACQtA94OdEfEcUAeOI8GHUesjgP+CjiJ5PM6S9KRNPYYsmqIi  
AP2B/g28Arg18Ah6bxDgF82qJ4OYA3wYpJeBFRs+S8Bvl/HopaT/IM+DbgNUCPrSfe5H1haNq  
8hnxtwEPAYacOoRtdTVsN/A/5vo+sBlgFPAIuBlvQ4+pNGHUfAnWE31Ey/H/j7ZvjM/DOznwP  
tTGyYpJXAC4F7gWdExKZ00e+BZ9S5lrykB4DNwJ3Ab4AdETGYrrKB5I9CvXyC5B94MZ1e0uB6  
AAL4gaT7JV2SzmVU53Y4sAX41/SS6/WS5jWwnlLnAV9OHzesnojYCPxv4LfAJmAncD+NO44eA  
l4uaYmkDuAMYAXN8ZnZDBYQISape/g68I6IeLp0WST/S1bX7x1ERCGSS0HLSS53HF3P/ZeSdB  
awOSLub1QN4zgl1l4EvJLkMvCppQvr/Lm1AC8CPHSRLwR2U3YZqhHHUXp/6Rzg38qX1bue9N7  
Sq0gC/1nAPKBh/Q1GxDqSS5k/AO4AHgAKZevU/TOzmTvgQkxSK0mAfTEivpHoflLSIenyQ0jO  
iOouInYAd5FcZlkoqSVdtBzYWKcyXgacI2k98K8klxSvbmA9wPD/2RMRmOnu95x4z63DcCGi  
Lg3nf4aSag1+jh6JbAmIp5MpxtZzx8Dj0XElogYAL5Bcmw17DiKiBsi4sSIOJXkftyvaPxnZj  
N0QIWYJAE3Aosi4p9Kft0KXJg+vpDkXlm9auqStDB9PjfkHt06kjA7t941RcQ/RMTyiFhJcmn  
qRxHx+kbVayBpnqT5Q49J7vs8RIM+t4j4PfCEPoems/4IeLhR9ZQ4n/2XEmlwPb8FTpbUkf67  
G3qPGnkchZz+PhR4DfAlGv+Z2QwdUD12SDoFuBt4kP33e95Hcl/sq8ChJMO8/H1EbKtTTccDN  
5003soBX42IKyQdQXImtBj4GfCGiNhXj5pKalsFvDsizmpkPem+v51otgBfioiPSFpC4z63E4  
DrgTbgUeCnJP9fg+qZRxIcR0TEznRew96fdP8fAl5H0ir4Z8CbSe6BNeo4upvk/u4A8M6I+GG  
j3yObuQMqxMzMbHY5oC4nmpnZ7OIQMzOzzHKImZlZZjnEzMwssxxiZmaWWQ4xm/UkvVpSSGpY  
TyhmVhsOMTsQnA/cK/42s1nEIWazWtpP5ikkQ36cl87LSbomHQvsTkm3Szo3XXaipP9IOxr+/

```
lCXRGbWnBxiNtu9imTcr18BT0k6kaTLoZXA84ALSPqqHOpX81PAuRFxInAj8JFGFG1mlWmZfB
WzTDufpANjSLo7Op/kuP+3iCgCv5d0V7r8ucBxwJ1Jd3/kSYyRMbMm5RCzWUvSYpJe+J8vKUh
CKdjfD+OopwBrI+IldSrRzGbIlxNtNjsXuCUiDouIlRGxgmRE5m3Aa9N7Y88AVqXr/xLokjR8
eVHSsY0o3Mwq4xCz2ex8Rp91fR14JsmYYA8DXwDWADsjop8k+D4q6eckAye+tH7lmtlUuRd7O
yBJ6oyI3nQojvuAl6XjhJlZhviemB2obksHI20DPuwAM8smn4mZmVlm+Z6YmZl1lkPMzMwyyy
FmZmaZ5RAzM7PMcoiZmVlm/X9MncyKP0xsSAAAAABJRU5ErkJggg==\n"
```

```
    },
    "metadata": {
      "needs_background": "light"
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "4. Perform descriptive statistics on the dataset"
  ],
  "metadata": {
    "id": "ak-SIXanIEl3"
  }
},
{
  "cell_type": "code",
  "source": [
    "df.describe()"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 364
    },
    "id": "Wp7ucqk-IO23",
    "outputId": "5e958dab-8649-4aae-cdfb-6b4f2c03e595"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "      "      RowNumber      CustomerId      CreditScore
Age      Tenure  \\\n",
10000.000000      "count      10000.000000      1.000000e+04      10000.000000
10000.000000      10000.000000      \n",
38.921800      "mean      5000.500000      1.569094e+07      650.528800
10.487806      "std      2886.89568      7.193619e+04      96.653299
18.000000      "min      1.000000      1.556570e+07      350.000000
32.000000      "25%      2500.750000      1.562853e+07      584.000000
37.000000      "50%      5000.500000      1.569074e+07      652.000000
44.000000      "75%      7500.250000      1.575323e+07      718.000000
      7.000000      \n",
```

```

92.000000    "max    10000.00000    1.581569e+07    850.000000
10.000000    \n",
"\n",
"          Balance    NumOfProducts    HasCrCard
IsActiveMember  \\n",
"count    10000.000000    10000.000000    10000.000000
10000.000000    \n",
"mean     76485.889288          1.530200          0.70550
0.515100    \n",
"std      62397.405202          0.581654          0.45584
0.499797    \n",
"min       0.000000          1.000000          0.00000
0.000000    \n",
"25%       0.000000          1.000000          0.00000
0.000000    \n",
"50%       97198.540000          1.000000          1.00000
1.000000    \n",
"75%      127644.240000          2.000000          1.00000
1.000000    \n",
"max      250898.090000          4.000000          1.00000
1.000000    \n",
"\n",
"          EstimatedSalary          Exited    \n",
"count    10000.000000    10000.000000    \n",
"mean     100090.239881          0.203700    \n",
"std      57510.492818          0.402769    \n",
"min       11.580000          0.000000    \n",
"25%       51002.110000          0.000000    \n",
"50%      100193.915000          0.000000    \n",
"75%      149388.247500          0.000000    \n",
"max      199992.480000          1.000000    "
],
"text/html": [
"\n",
"  <div id=\"df-2dbf72e4-9e1b-41af-b79f-149ac692cb0b\">\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>CreditScore</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>count</th>\n",
"      <td>10000.00000</td>\n",
"      <td>1.000000e+04</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",
"      <td>10000.00000</td>\n",
"      <td>10000.000000</td>\n",
"      <td>10000.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>mean</th>\n",
"      <td>5000.50000</td>\n",
"      <td>1.569094e+07</td>\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>0.70550</td>\n",
"      <td>0.515100</td>\n",
"      <td>100090.239881</td>\n",
"      <td>0.203700</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>std</th>\n",
"      <td>2886.89568</td>\n",
"      <td>7.193619e+04</td>\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>0.45584</td>\n",
"      <td>0.499797</td>\n",
"      <td>57510.492818</td>\n",
"      <td>0.402769</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>min</th>\n",
"      <td>1.00000</td>\n",
"      <td>1.556570e+07</td>\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>0.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>11.580000</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25%</th>\n",
"      <td>2500.75000</td>\n",
"      <td>1.562853e+07</td>\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>0.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>51002.110000</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50%</th>\n",
"      <td>5000.50000</td>\n",
"      <td>1.569074e+07</td>\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>1.000000</td>\n",
"      <td>1.000000</td>\n",
"      <td>100193.915000</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>75%</th>\n",
"      <td>7500.25000</td>\n",
"      <td>1.575323e+07</td>\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>1.000000</td>\n",
"      <td>1.000000</td>\n",
"      <td>149388.247500</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>max</th>\n",
"      <td>10000.00000</td>\n",
"      <td>1.581569e+07</td>\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>1.000000</td>\n",
"      <td>1.000000</td>\n",
"      <td>199992.480000</td>\n",
"      <td>1.000000</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"    <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-2dbf72e4-9e1b-41af-b79f-
149ac692cb0b')\">\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 2.06-.94-2.06-
.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-
.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 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 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-
2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78
2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41
20z\"/>\n",
"        </svg>\n",
"      </button>\n",
"    </div>\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-2dbf72e4-9e1b-41af-
b79f-149ac692cb0b 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-
2dbf72e4-9e1b-41af-b79f-149ac692cb0b');\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 notebook</a>'\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",
    "
    ]
  },
  "metadata": {},
  "execution_count": 23
}
]
},

```

```

{
  "cell_type": "markdown",
  "source": [
    "5. Handle the missing value"
  ],
  "metadata": {
    "id": "F8SAIMtpIUz"
  }
},
{
  "cell_type": "code",
  "source": [
    "data = pd.read_csv(\"Churn_Modelling.csv\")\n",
    "pd.isnull(data[\"Gender\"])"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "4hM5tbR5Ifmt",
    "outputId": "c3670f67-78bb-4590-b6a5-0fb15df126c3"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "0      False\n",
          "1      False\n",
          "2      False\n",
          "3      False\n",
          "4      False\n",
          "...   \n",
          "9995   False\n",
          "9996   False\n",
          "9997   False\n",
          "9998   False\n",
          "9999   False\n",
          "Name: Gender, Length: 10000, dtype: bool"
        ]
      },
      "metadata": {},
      "execution_count": 24
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "6. Find the outlier and replace the outlier"
  ],
  "metadata": {
    "id": "LEw6n-74ImNK"
  }
},
{
  "cell_type": "code",

```

```

"source": [
  "df[\"Tenure\"] = np.where(df[\"Tenure\"] >10,
np.median(df[\"Tenure\"])\n",
  "df[\"Tenure\"]"
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "wxAeY8NJIsjg",
  "outputId": "34dfb479-9c26-4310-81bd-b95d570465a2"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "0      2\n",
        "1      1\n",
        "2      8\n",
        "3      1\n",
        "4      2\n",
        "      ..\n",
        "9995    5\n",
        "9996   10\n",
        "9997    7\n",
        "9998    3\n",
        "9999    4\n",
        "Name: Tenure, Length: 10000, dtype: object"
      ]
    },
    "metadata": {},
    "execution_count": 25
  }
]
},
{
  "cell_type": "markdown",
  "source": [
    "7. Check for categorical columns and perform encoding"
  ],
  "metadata": {
    "id": "kOLJNGLvIORG"
  }
},
{
  "cell_type": "code",
  "source": [
    "pd.get_dummies(df, columns=[\"Gender\", \"Age\"],
prefix=[\"Age\", \"Gender\"]).head()"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 300
    },
    "id": "1gUWe06VJCWH",
  }
}

```

```

    "outputId": "a39fa651-dd7c-4a0f-bf12-75ba97c8e2ee"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "      RowNumber  CustomerId  Surname  CreditScore  Geography
Tenure  Balance  \\\n",
2      0.00  \n",
1      83807.86  \n",
8      159660.80  \n",
1      0.00  \n",
2      125510.82  \n",
          "      NumOfProducts  HasCrCard  IsActiveMember  ...
Gender_78  Gender_79  \\\n",
0      0  \n",
0      0  \n",
0      0  \n",
0      0  \n",
0      0  \n",
0      0  \n",
          "      Gender_80  Gender_81  Gender_82  Gender_83  Gender_84
Gender_85  \\\n",
0  \n",
0  \n",
0  \n",
0  \n",
0  \n",
0  \n",
          "      Gender_88  Gender_92  \n",
          "0      0      0  \n",
          "1      0      0  \n",
          "2      0      0  \n",
          "3      0      0  \n",
          "4      0      0  \n",
          "\n",
          "[5 rows x 84 columns]"
        ],
        "text/html": [
          "\n",

```

```

" <div id=\"df-436055e6-5c75-440d-b31e-955708e2dbb3\">\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>Tenure</th>\n",
"      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>...</th>\n",
"      <th>Gender_78</th>\n",
"      <th>Gender_79</th>\n",
"      <th>Gender_80</th>\n",
"      <th>Gender_81</th>\n",
"      <th>Gender_82</th>\n",
"      <th>Gender_83</th>\n",
"      <th>Gender_84</th>\n",
"      <th>Gender_85</th>\n",
"      <th>Gender_88</th>\n",
"      <th>Gender_92</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>2</td>\n",
"      <td>0.00</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>...</td>\n",
"      <td>0</td>

```

[illegible]



```

"      <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>1</td>\n",
"      <td>0.00</td>\n",
"      <td>2</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>...</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</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>2</td>\n",
"      <td>125510.82</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>...</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"<p>5 rows × 84 columns</p>\n",
"</div>\n",
"    <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-436055e6-5c75-440d-b31e-
955708e2dbb3')\">\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 5.44l1.94 2.06.94-2.06 2.06-.94-2.06-
.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-
.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 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 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78
2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78
2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41
20z\"/>\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-436055e6-5c75-440d-
b31e-955708e2dbb3 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-
436055e6-5c75-440d-b31e-955708e2dbb3');\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 notebook</a>\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",
        "
    ]
  },
  "metadata": {},
  "execution_count": 27
}
],
},
{
  "cell_type": "markdown",
  "source": [
    "8. Split the data in to dependent and independent variables"
  ],
  "metadata": {
    "id": "M4FytxJAJWom"
  }
},
{
  "cell_type": "markdown",
  "source": [
    "8.1 split the data in to independent variables."
  ],
  "metadata": {

```

```

    "id": "2GvXrNh3JoS8"
  }
},
{
  "cell_type": "code",
  "source": [
    "x = df.iloc[:, :-2].values\n",
    "print(x)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "KZzPm2UkJ09P",
    "outputId": "5452d768-bd55-4cab-bed2-29ebf319753b"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[[1 15634602 'Hargrave' ... 1 1 1]\n",
        " [2 15647311 'Hill' ... 1 0 1]\n",
        " [3 15619304 'Onio' ... 3 1 0]\n",
        "... \n",
        " [9998 15584532 'Liu' ... 1 0 1]\n",
        " [9999 15682355 'Sabbatini' ... 2 1 0]\n",
        " [10000 15628319 'Walker' ... 1 1 0]]\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "8.2 Split the data into depent vaiables"
  ],
  "metadata": {
    "id": "GJANbD0KKTW2"
  }
},
{
  "cell_type": "code",
  "source": [
    "Y = df.iloc[:, -1].values\n",
    "print(Y)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "7wYWd8uFKbw4",
    "outputId": "b68ce27a-e065-4cdf-9bf7-6391a0d45132"
  },
  "execution_count": null,
  "outputs": [
    {

```

```

        "output_type": "stream",
        "name": "stdout",
        "text": [
            "[1 0 1 ... 1 1 0]\n"
        ]
    }
]
},
{
    "cell_type": "markdown",
    "source": [
        "9. Scale the independent variables"
    ],
    "metadata": {
        "id": "MeffK6J2KgZR"
    }
},
{
    "cell_type": "code",
    "source": [
        "import pandas as pd\n",
        "from sklearn.preprocessing import MinMaxScaler\n",
        "scaler = MinMaxScaler()\n",
        "df[["RowNumber\"]] =
scaler.fit_transform(df[["RowNumber\"]])\n",
        "print(df)"
    ],
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "id": "myR-aFRqKsbJ",
        "outputId": "0521a61f-5497-4450-8a48-973ea1ee1b87"
    },
    "execution_count": null,
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "
                RowNumber  CustomerId  Surname  CreditScore
Geography  Gender  Age  \\\n",
                "0          0.0000    15634602   Hargrave          619
France  Female  42  \n",
                "1          0.0001    15647311     Hill          608
Spain  Female  41  \n",
                "2          0.0002    15619304     Onio          502
France  Female  42  \n",
                "3          0.0003    15701354     Boni          699
France  Female  39  \n",
                "4          0.0004    15737888  Mitchell          850
Spain  Female  43  \n",
                "...          ...          ...          ...
...          ...  \n",
                "9995      0.9996    15606229  Obijiaku          771
France  Male  39  \n",
                "9996      0.9997    15569892  Johnstone          516
France  Male  35  \n",
            ]
        }
    ]
}

```

		"9997	0.9998	15584532	Liu	709
France	Female	36	\n",			
		"9998	0.9999	15682355	Sabbatini	772
Germany	Male	42	\n",			
		"9999	1.0000	15628319	Walker	792
France	Female	28	\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",				
	"9995	5	0.00	2	1
0	\n",				
	"9996	10	57369.61	1	1
1	\n",				
	"9997	7	0.00	1	0
1	\n",				
	"9998	3	75075.31	2	1
0	\n",				
	"9999	4	130142.79	1	1
0	\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	\n",
"...	...	...	\n",
"9995	96270.64	0	\n",
"9996	101699.77	0	\n",
"9997	42085.58	1	\n",
"9998	92888.52	1	\n",
"9999	38190.78	0	\n",
"\n",			
"[10000 rows x 14 columns]\n"			

```

    ]
  }
]
},
{
  "cell_type": "markdown",
  "source": [
    "10. Split the data into training and testing"
  ],
  "metadata": {
    "id": "8G_FtHsfLErP"
  }
}

```

```

    },
    {
      "cell_type": "code",
      "source": [
        "from sklearn.model_selection import train_test_split\n",
        "train_size=0.8\n",
        "X = df.drop(columns = ['Tenure']).copy()\n",
        "y = df['Tenure']\n",
        "X_train, X_rem, y_train, y_rem = train_test_split(X,y,
train_size=0.8)\n",
        "test_size = 0.5\n",
        "X_valid, X_test, y_valid, y_test = train_test_split(X_rem,y_rem,
test_size=0.5)\n",
        "print(X_train.shape), print(y_train.shape)\n",
        "print(X_valid.shape), print(y_valid.shape)\n",
        "print(X_test.shape), print(y_test.shape)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "hRSjcya8LN3v",
        "outputId": "c136c8b7-a2ff-4408-c53b-c70255d9e54a"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "(8000, 13)\n",
            "(8000,)\n",
            "(1000, 13)\n",
            "(1000,)\n",
            "(1000, 13)\n",
            "(1000,)\n"
          ]
        },
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "(None, None)"
            ]
          },
          "metadata": {},
          "execution_count": 34
        }
      ]
    }
  ]
}

```