```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "ETAZ_x0QWpG-",
```

```
    "outputId": "67904d6f-f9e0-4b9c-cdb1-ae82d03a6c75"
},
"outputs": [
    {
        "output_type": "stream",
        "name": "stdout",
        "text": [
            "Mounted at /content/drive\n"
        ]
    }
    ],
    "source": [
        "from google.colab import drive\n",
        "drive.mount('/content/drive')"
    ]
},
{
    "cell_type": "code",
    "source": [
        "import numpy as np\n",
        "import pandas as pd\n",
        "import seaborn as sns\n",
        "from matplotlib import pyplot as pt\n"
],
"metadata": {
```

```
        "id": "MQvsGvrHkHqN"
      },
      "execution_count": 2,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "%matplotlib inline"
      ],
      "metadata": {
        "id": "7sawSmNYkRSG"
      },
      "execution_count": 3,
      "outputs": []
    },
    {
      "cell_type": "code",
          "source": [
            "df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/downloads/Churn_Modelling.csv')\n",
            "df"
          ],
          "metadata": {
            "colab": {
              "base_uri": "https://localhost:8080/",
```

```
        "height": 488
      },
      "id": "GA41Hn-4kUxl",
      "outputId": "2a7e4277-a90d-4b3f-b700-fc5c91d9726b"
    },
    "execution_count": 4,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "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",
            "9995    9996    15606229  Obijiaku          771    France
Male   39   \n",
            "9996    9997    15569892  Johnstone         516    France
Male   35   \n",
```

```
"9997 9998 15584532 Liu 709 France Female 36 \n",
"9998 9999 15682355 Sabbatini 772 Germany Male 42 \n",
"9999 10000 15628319 Walker 792 France Female 28 \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",
"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",
"\n",
```

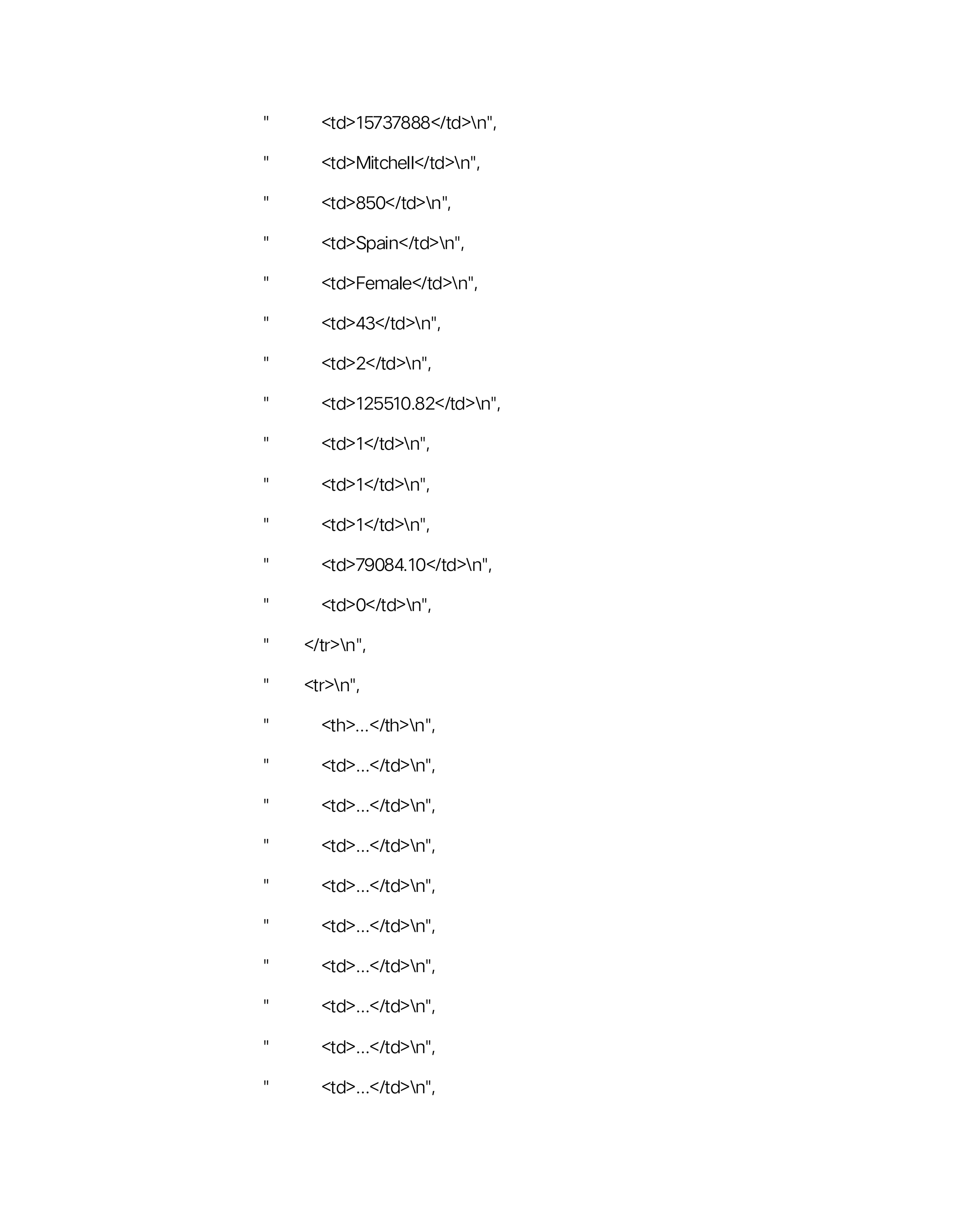
```
"          EstimatedSalary    Exited    \n",
"0               101348.88            1    \n",
"1               112542.58            0    \n",
"2               113931.57            1    \n",
"3                93826.63            0    \n",
"4                79084.10            0    \n",
"...                   ...          ...    \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-e9eb92fb-3b89-4a44-bbf1-ffc9461e2083\">\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>...</td>\n",
"            <td>...</td>\n",
"            <td>...</td>\n",
"            <td>...</td>\n",
"            <td>...</td>\n",
"        </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-e9eb92fb-3b89-4a44-bbf1-ffc9461e2083')\"\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 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-e9eb92fb-3b89-4a44-bbf1-ffc9461e2083
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-e9eb92fb-3b89-4a44-bbf1-ffc9461e2083');\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": 4
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "sns.displot(df.Tenure)"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 386
      },
      "id": "AkMU2HRDkoOL",
      "outputId": "4d3f9aca-7b4c-446d-dfbd-190df6086673"
    },
    "execution_count": 5,
```

"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"<seaborn.axisgrid.FacetGrid at 0x7fdebb761250>"
]
},
"metadata": {},
"execution_count": 5
},
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 360x360 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAWAAAAFgCAYAAACFYaNMAAAABHNCSVQICAgIfAh
kiAAAAAlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2
ZXJzaW9uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAATdElEQVR4nO3df6
ymZX3n8fdHRvyBWwfo7CzODAuNxJa1seIUEXYb13EbtG6H3aDYWJ01dKdJqdXStGJZ3Eze
72UQTt/5oG1YWqENKUJeygXYNlgLabFupA7oqjF0mKMwMvw4WsbWxOPjdP55r4nE6cA
5znud8Oee8X8mT576v+7rv5/tkZj5zP9f9f9K1WFJGn5Pau7AEIaqwxgSWpiAEtSEwNYkpoYw
JLUZF13AbNw7rnn1o033thdhiQdkiM1rso94EceaS7BEIa0KoMoMYElaCQxgSWpiAEtSEwNY
kpoYwJLUxACWpCYGsCQ1MYAIqYkBLElNDGBJamIAS1ITA1iSmhjAktTEAF5lNm05mSQLv
jZtObm7VGnNX3A17L7L7t+/jws+++vEL569DNVPVIeiruAUtSEwNYkpoYwDPiWKykhTgG
PCOOxUpaiHvA0irS8cvLX3tHzz1gaZ+OXMlogrGz6zKpeaTFjmY5jappmOQb8MeB3gKvmtV
0C3FxV70tyyyZh/N/A64LTxeiVwKfDKJCcA7wW2AgXcnnqnSGqnp0hnXradq05WTu379vwX4v"

2ryFA/vuW4aKjs5ixjIdx9Q0zSyAq+pPk5xyWPN24NVjehfwGSYBvB24qqoK+FyS9UlOGn1v
qqq/BkhyE3AucM2s6tbT50EY6egs9xjwxqp6YEw/CGwc05uA+btQ+0fbk7X/A0I2JtmdZPfc
3Nx0q5akGWg7CDf2dmuK27usqrZW1dYNGzZMa7OSNDPLHcAPjaEFxvvDo/0AsGVev82j
7cnaJWnFW+4AvgE4dCbDDuD6ee1vG2dDnAU8NoYqPg38dJLjxxkTPz3aJGnFm+VpaNc
AfwG8JMn+JBcC7wP+VZK7gdeOeYBPAfcAe4H/AfwSwDj49l+Az4/Xfz50QE6rI5e2aq2Y5
VkQP/cki7YdoW8BFz3Jdq4ErpxiaXqG86wKLcZqOP3Re0FIWpFWw3/UXoosSU0MYElqYg
BLUhMDWGrkGR9rmwfhpEar4UCSjp57wJLUxACWpCYGsCQ1MYAIqYkBLEINDGBJamIA
S1ITA1iSmhjAktTEAJakJgawJDUxgCWpiQEsSU0MYElqYgBLUhMDWJKaGMCS1MQAIqQ
mBrAkNTGAJamJASxJTQxgSWpiAEtSEwNYkpoYwJLUxACWpCYGsCQ1MYAIqYkBLEIND
GBJamIAS1ITA1iSmhjAktTEAJakJgawJDUxgCWpiQEsSU0MYEIqYgBLUhMDWJKaGMCS1
KQIgJP8apI7k3wIyTVJnpvk1CS3Jdmb5BNJjh19nzPm947Ip3TULEnTtuwBnGQT8CvA1qp6
KXAM8Gbg/cAHq+rFwKPAhWOVC4FWlWWAc9Lsg54PvAA8Brg2rF8F3D
emN4+5hnLtyXJMtYqSTOx7AFcVQeADwDW3MQnex4DbgW9W1cHRbT+waUxvAvaNdQ+O/
icevt0kO5PsTrJ7bm5utl9CkqagYwjieCZ7tacCLwCOA85d7d4Wzds2LDUzUnSzH
UMQbwW+FpVzVXVd4HrgHOA9WNIAmAzcGBMHwC2AIzlzwf+s2+sbwlS9L0dQTwfcBZSZ4/
xnK3AXcBtwLnjz47gOHqqqZaxXkmaiYwz4NiYH0+4AvjxquAx4N3Bxkr1MxnivGK
tcAZw42i8GLInumiVpFtYt3GX6quq9wHsPOPMIfb8DvHE56pkKk5eSVcJLUxACWpcY
GsCQ1MYAIqYkBLEIND GBJamIAS1ITA1iSmhjAktTEAJakJgawJDUxgCWpiQEsSU0MYEIqY
gBLUhMDWJKaGMCS1MQAIqQmBrAkNTGAJamJASxJTQxgSWpiAEtSEwNYkpoYwJLUxA
CWpCYGsCQ1MYAIqYkBLEINDGBJamIAS1ITA1iSmhjAktTEAJakJgawJDUxgCWpiQEsSU0
MYEIqYgBLUhMDWJKaGMCS1MQAIqQmBrAkNTGAJamJASxJTQxgSWpiAEtSk5YATrI+yb
VJvppkT5JXJTkhyU1J7h7vx4++SXJ5kr1J7khy1lKgsap+FHgZsTfCnJGR01S9K0de0Bfxi4sap+FHgZsAe4BLi5qk4D
Dbh7zAK8DThuvncCly1+uJE3fsgdwkhcrVj1+uJE3fsgdwkhcCPwVcAVBVj1fVN4HtwkVz7wCVS9Os+jjvDGKHCwlpnjeDIxVE58D1ic
5aZnLIqSp69gDPhWYA34vyReSfCTJccBJVT0w+jwInDSmNwH7Fq2/f7Q9RZKdSXYn2T03NzfCP7vy6HhUJVT0w+jwIbBzTm4B989bfP9p+QJKdSXYn2T03
NzfD8iVpOjoCeB1wFnBZVb0c+DbwaA0Ob9q2H24AoKoKqKez0aq6oqq2VtXWDRs2TGlWSZqVwA8MgBA8GPRg8D4Q+vUbjZMn...

V9RNMhjveMs3itfo4Bqy15DnAS4GbJrco4Bgmt0s85NBNi24HTInE9p5gcsMjmNzv4BXA58
e2n8cSb9Si1c8A1IoS4M6qerJH/vz9eH+C7//bOMgP/IKc/2id78wb9w2wq6reM61itfo5BKG
15O+BDYeeuZbk2Un+2QLrfB34iSTPSrKFyRNUjuRm4Pwk/3hs+4Qk/3RKdWuVcg9Ya8n3g
POBj4xHAa1j8pSNp7r73Z8BX2Nyi8g9TO7w9Q9U1V1J/iPwx0meBXwXbHbOkJHbOkIl/I0NEIq4h
CEJDUxgCWpiQEsSU0MYEIqYgBLUhMDWJKaGMCS1OTA41BrgMAjLkkAAAAAElFTkSuQ
mCC\n"

```
        },
        "metadata": {
          "needs_background": "light"
        }
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "df.plot.line()"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 293
      },
      "id": "5I8ATmo9ksCB",
      "outputId": "3f106a5f-d08f-437b-c8e2-5324726ca39d"
    },
  "execution_count": 6,
```

"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"<matplotlib.axes._subplots.AxesSubplot at 0x7fdea91aacd0>"
]
},
"metadata": {},
"execution_count": 6
},
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 432x288 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAXQAAAEDCAYAAAAIRP8qAAAABHNCSVQICAgIfAhki
AAAAAlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2Z
XJzaW9uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3dd3g
U1frA8e+bnkAooUtLQEoCCQuEKB3pIgYRFRAvICIqgr2A/kT02rgWLGDhKoKCiEQpII4RBAF
BgWBAekBDEwgEElp6zu+P3axJSNmEDSHL+3mePNmZOmum8nflMMallFLLn1tZN0AppZRkXoQF
1tZN0AppZRzaEBXSikXoQFdKaVchAZ0pZRyERyERrQlVLKRWhAV0opF1GmAV1EZopIvlhsd6D
sVBGJsf3sFZHEy9FGpZQqL6Qs70MXkS7AoXAzY0zLYqw43xgTKyLXAe8D3bMXikhdD
BTOeeJSGMR+Z+IRlvlWhFpns+qQ4F5l6WdVTniUdQPyMQO43xgTKyLXAe8D3bMXikhD
lAj4qYzap5RSV6QrKqCLSEWgA7PYZ50QAyy4AvcmuC+rEsz6Ce8Uwu/ZKmjjSudQAb/TWQ
soMAR68TO1RSqIy44q6bdEYcwb4S0RuBxCrVtnLRaQZUBXYUEZtVEqpK1KZBnQRmYc1MDcTkcMi
WETuAYYB94jIVmAHMCDHJkOBb4zOpKGUUrlc9j50EZkJ9AfijDEF3skiIm2B34HBxpiIy9U+pZ
QqrH0B4YGYMr5LaXKpejUikKcCuQiciBgpp4K0S0KaiBf1iLLavxfk9ZS8AbnZiJyNOeK6nJRSi
lVcmX2Kr7N2Gcl0Z2Ssc4zRBJ7Lx4Z3NmVUuieBgpZpI4tSSrkIDehKKeNK
YGBgWW1eKaXKpejo6JjGcMZaWsiwKqTxUGsTlpo4NzCBGsMBhyQaB4ymzZvLavNKUuiciBgpZpl4tSSrklDehKKe

UiNKArpZSLKDKgO5LiVkS62dLa7hCRn53bRKWUUo5w5Ap9FtC3oIUiUgVrAq1IY0wL4Hb
nNE0ppVRxFBnQ80txm8edwDfGmIO28vFOaptSSqlicEYfelOgqoistuUwH15QQREZIyKbR
WTziRMnnLBppZRS2ZxxH7oH0BboAfgCG0TkV2PM3rwFjTEzsOY7Jzw8vOQ5BxIPwtIj8PlA
SDsHgZ0hIwVqh8GxbZCwD9IuQL1w6096Cmz8CPq8AtvmQ+cn4Pc54FcN/vgKsjJy13/bp
3AkGjZMg6Y3WufVbwcrX4S7v4ff50KdMKjSEI7/AT+9BBVqQJUGcO4EeHjB7bNg5xJY8x+
ofz0ENILd38E1Fuj9Evw4CUIGWH8HdYGKtcC3CmRlWrdbqyUcjYE6FrjjMzixG/athE0fg7sn
9HkZju+EzZ9A1SCodi007QPbv4ZWQyD1LHhXgvQLENgJ/o6BZv3gjwWQfBri1kFAEOxaAo
9sh8QDsOBuqFwPaoZAzWD47UPr75QkaNQNDqyHC6fg4Hp4dCcs/z9IiIVz8dZ9CewMZ4
/C90/IPp4RY6B5f+t2E/YBxnrMAG79GEym9VhVrAk3vw37VsDicda2HN4EjW6AGs2hbhvre
U9JAhHY8z20vgt+fB6qN4UWtjYkJ8Lfv0PMF1ClPoTeDv614c+fITPVWpebO2xfCH5V4boH
4Ngf8Ot0a3v2fg/rpkJwJKSdh86PWc9ToxugWmPYMB18q8K1Pax1NusH3z8J3pWh0jXg6
Qs+lazn4PgOa9mzR63ncsB02PYIrH/Puv/NboILCRC/E1LPWM9I16fg5/9AxL3w52rrub/+QU
g6BKfjrHX9HWM9bhH3QcP21vdNzFzw9AOvCpCZDn9vgbrhcP4E9J8K3z5sXde/DISqa21
n9KfQ+2Xr52jnYug2wXpuTu6FEUshPRl2fwut7oSF91nfJ9Wuta7bsJP1/WS5E84dt34GaofC
X2us53rPMujxPPz6PnQYb23b0a3Q8IbrZxdA3KFhB4hbC417QOhtsPYt6/uqxa3Q7Eb430Ro
cL31vZB8Gm78j3XdNa9bP/O1QqBGMGSIw/ZvrO//mLIgGQarXobK9eEv21d79a+3HhcPX0
hNss5re7f1GGemQauh1mO7938Q1BmuHwtR90D6eajXDrJTpQz8EPb+AIvut763D22EEd9
aj/X+IdaYc+Yw7Pr2n8/nvh+t6/acDJ0eLXH4K4xDuVxEJBBYmt8wcSIyAfA1xjxvm/4E+J8xZk
FhdYaHh5sSPVh05ii8ld8gRkopVU4M/Mj6h6cERCTaGBOe3zJndLksBjqJiIeI+AHXAbucUG
/+Vr1caIUrpdRlsfC+Uqm2yC4XW4rbbkB1ETkMPA94AhhjPjTG7BKR/wHbgCzY2Y2NMgbc4X
rKtOpSoUkrIp8iAbowZ6kCZ14HXndKioozbBO+2viybUkqp8uRKG4KuaAGNYHKS9cuJf8Y
dLZgxtp8scPe4eJnIP8vF7Z/p7Lqzsqy/Raxl3Nz/WT8ry/qFqofXP/My0nJPg/ULKwTc3Kzru
OXT05Vzm9nfa5isHOtlWue7uTu23/nJb9vZxye/NhWnrow065e1IpCZcfGxzrVujuNRVFtyHp
f0FPDwth5zcct9LuzHrIB9yXmu8x4/R99L+W0L2znJyvznd/Z8d8+L18vKtG4r+71W1Pazt5V
3efbxzpaVcfH2CpNlG5I35zHM2UZ3j38+F/mdq5zlCzt2WVnW5dl1OtQ222cuv21mL8v+LG
ZI2t4L+byXcn6GRRw/x3k/i/mtl/c9nD2d33Zy1IHUZ+MSIb+Ans3Rk2M/wPm8ObLrELF+255
f3TnfKJLnze/mBm55gnfeYA65PzQFfTBybtPerpzr5dI2SeS37eK80QurK+d+F/WGLWhf8mt
LzmlPH1v9+QSunOeyoLoLWl7c/c+vrux9Kuw8ieR/bArbfkHL8r7PihPMofBzkN3GvJ+L4rTP
vh3b+6Q4Qaywiwv7siLqLegz7Ij8PosXtSNvLCjivGcrxWAOmstFKaVchgZ0pZRyERrQlVLKR
WhAV0opF6EBXSmlXIQGdKWUchEa0JVSykvoQFdKKRehAV0ppVxE+X1SVClII56ezuHDh0
IJSSnrpign8fHxoV69enh4cOv4UsAZ0pVzA4cOH8ff3JzAwEKIprh91xTDGkJCQwOHDh2tKK
CnJ4Pe1yUcoFpKSkUK1aNQ3mSSkw/49KArpSL0GDuWkyPjWgK6WUiygyolT
BGJF5FCRyESkXYikiEitzmveUqp8sLd3R2R2LxULLi25+eabSUxMLHFd06dPZ86cOQAMGTKE
WvNRmAhAXF0fLhcNkVzuOXKFPgvoW1BEgDXpm1DNu3byTycsgJ4bJ+ZqIs
Dp06dfcp3Vq1fnzTffLPC5zp07pg5plgbKlSaQsN67vJZpoBBkBr94ODpMW
ExMRw/fXXXj+GtJbhG/wq8AcU5pFn4lUNgKPbvlGHgBbUhbDmHA05lqUBAsvRlvGBa8Bq4ivb
ExMRw/fXXExYWxCBAzl9+jTvcH4tEyOfAsWBMI9OwclzDSFq3
CMp7hf3GG28wefJkJn1ap168Ohz5X5d9tK9EyhILnzYhPDwcN5nqj3rAtTTHfSXvC3LSlDb5gsWcvZlRrS
Bwdx22232bUZHR9O1af1atm1nz59ZuOgqL3eRx55hPDwcN5nqj4MYG05dGstQ2Bw2Bgtmi
A4AagXRFIxwBjABo0aHCpm1ZK5ZK5e0Fb3ew8+8zTq0z5JpKPH9zC4fKZmZmsnsmnLsSu655x4
Ahg8fznvvvUfkXrI2ZNGkSL7zwb3zwAm+//TTq1a+nUqRM1a9bEz88P39+fMSnnHz3TPx79
HaxjBo1infeeYcXXXnjB4fZ6eXmxYp9tnm3HHQYMGEDB0dBQAE0btyRx99fFA9e/bwySef
HaxjBo1infeeYcXXXnjB4fZ6eXmxYp9tnm3HHQYMGEDB0dBQAE0btyRx99fFA9e/bwySef

0LFjR0aNGsX777/Pww8/zPjx41m8eDE1atRg/vz5PPvss8ycOROAtLQ0Nm/eXJxDd1k54z70
t4GnjTFZRX0ra4yZAcwACA8PN07YtILqCpGcnIzFYuHIkSMEBwfTq1cvkpKSSExMpGvXrgC
MGDGC22+/HYAOHTrwyy+/sGbNGp555hn+97//YYyhc+fOuep96KGHsFgsPPHEEw63JTI
yEoDQ0FBatGhBnTp1AGjUqBGHDh2iSpUq1K9fn44dOwJw11138e6779K3b1+2b99Or169A
Osfp+x1AQYPHIzCo3N5OCOghwNf2oJ5daCfiGQYYxY5oW6lVDE5eiXtbNl96BcuXKBPnz5
Mnz6dESNGFFi+S5curF27IgMHDjBgwACmTJmCiHDTTTfIKIelShXuvPPXXH3yHh4eZGUP
GA0X3a/t7e0NgJubm/119nR2/3feC1ARwRhDixYt75trlChQmGHoMxd8m2LxpggY0y
gMSYQiALGajBX6url5+fHu+++y5tvvkmFChWoWrUqa9euBeDzzz+3X6137tyyZOXPm0KRJE
9zc3AglCGDZsmV06tTpojofe+wxPvrol3swrlWrFvHx8SQkJJCamsrSpSuL3c6DBw/aA/cXX
3xBp06daSdOnLDPT09PZ8eOHSU6DmxXBkdsW5wEbgGYiclhE7hGR+0Xk/tJvnlKqP
GrdujVhYWHMmzeP2bNN8+STTxlWFkZMTAyJk0CIDAwEGMMXbp0AaBTp05UqVKFqlW
rXlRf9erVGThwlKmpqQB4enoyadlklkIi6NWrF82bNy92G5G5s1a8b06dMJDg7m9OnTPPDAA3
h5eREVFcXTTz9Nq1atsFgsrF+//hKOBOxOUlxpRNV3Z4eLi5kr9cUKo82bVrF8HBwWXdDOVOk+
Z1XEYk2xoTnV16fFFVKKRehAV0ppyVyEBnSllHIRGtCtGtCVUspFaEBXSikXoQFdKaVchAZ0pZTT
HDt2jCFDhtC4cWPatm1Lt3792Lt3b7HqWFu7du/eZpmTVrFuPGjbvs7SkJ
DehKKacwxjBwjBw4EC6devG/v37iY2N5bXXXrv32uyNfrb8x/dyBNS79oPcdmgmy6ZlvMMS+T/vMQHj
sbg0Y4p8ap9gfjjkNHMAgAQ6xEn4tS9AEu1m3smzt/7wdZnufQrZQiD3fu5sI7ymNmY/8kbNE2gG
kEaMtAQEvp01GEqQdlfz3XNjHOJLuapmKvcUf8EasNHeP0yqNRngA0uNyi5nRZZNlmGEiKZx
...

w4iwh9//MHu3bvp3bu3vUtoy5YtbNu2jYCAAJ555hm6d+/OzJkzSUxMJCligp49e1KhQgUn
HM3SUz7/r1BKISsREREEBQUBsHz5crZt22bvH09KSiI2NpY1a9YwdOhQ3N3dueaaa+jevXu
R9VasWJHo6GjWrl3LqIWrGDx4MK+99hpt27aITp069rS5ISpVAmDdunWMHz8egObNm9O
wYUN7QO/VqxcBAQH2Ni5ZsoQ33ngDsGaIPHjw4BWfXkEDulIupqgr6dJSUPPpcIUeVrTGG9
957jz59+uQqs2zZhJt193dnW7dutGtWzdCQ0OZPXt2sQbaKKiNX3/9Nc2aNSmRm8ns1PfZuvTpw8fPnj3
RSjIFQeIzs1PfZuvTpw8ffPAB6enpAOzdu5fz58/TpUsX5s+fT2ZmJkePHrX3b+fI6fHxpL6elpX3fPnj3
ExsbaI8XExNCwYUOaNWvG0aNH2bRpE2DN7JiRkUHnzp2ZO3eufbsHDx7MN2j36d8dOH99
57j+zkhb///ntJD8tIpVfoSimnKCh97i233JKr3OjRo4mLi6NNmzYYY6hRowaLFi1i4MCB/PTTT
4SEhNCgQQPat2+f73bGjBldq04bHHHmP8+PEkJibi4eHBtddey4wZM/Dy8mL+/Pl+8mL+/P
mMHz+e5ORkfH19WbFiBWPHjuWBBx4gNDQUDw8PZs2aalWtgjGzPPfccjzzzyCGFhYWRIZ
REUFMTSpUtL5bg5k6bPVcoFaPpc16Tpc5VS6iqIAV0ppVyEBnSlIlHIRGtCVUspFaEBXSikXo
QFdKaVchAZ0pZTTLFq0CBFh927n5pJRjtGArpQRymnnz5tGpUyfmzZtX1k1RxYhAV0o5o5xbklz
51i3bh2ffPKJPaBnZmby1FNPERISQsGDB9uTr23atIkOHToQFhbG9dlrrrmpq1zxJ2j9ffP9Y6
/SrmYY6+8Quou53Z5eAc3p7ZtUlmCLF68mL59+9G0KP31F3FxcWzcuJGvv/6adu3a/+Pv/vTk6bbq7q
+HiCg4MZNWoU6enpjB8P/nsWLF1OjRg3mz5/Ps88+y8WTE28yZM53a7quNBnSlIIFPMmzePhx/+4x
IAhQ4Ywb948MqaUebX44wbS8uzqYsSoP/kP2hP8+G
XbaAhcjNPOWhLk6WaO5gK0VkRcDmDJnfaoJHDQwGwWCysWbPmsrbT1V3WL0WNMTOA
GWDNh345t62UKdKgaiLVDdvZPmFt7mRRFJhVOXRkzg+kidSgAMLkxnB6bnHWMaiDMysqK8tMVU5KqUo8yl2qeM9JD8Ucqr1QbVzVzFtAfVWMpOPaCjXOHsQbLDDm
mc8WLQEGC5W1wNJxhh9QkAppS4zR25bnAd0A6qLyGHgecAWT7inuh3L+7Detnh3
aTVWKaVUwYoM6MaYUsN8CDTmuRUkqpEtFcLkop5SL00X+I1CVLSEigR48Bw7dgx
3d3dq1KgBwMaNG/Hy8irL5SIArpS5ZtWrVimmJAWDy5MlUrFiRJ554ws73oSS2lZmZibu7e
UXd5pl4tSqlQUlB63W7duPP3000RERNC0aVpVPWrI0LwKxZsg3bpx9/f79+9tveaxYSKPP/4
4rVq1YsOGDcOMaeliAgsFgv33zuzcN7zuzHjrn1DqrI69l5zuaQlzeG
FNoetyMjAw2btzftzClsmXLeOGF1ixYkhh9Z0/f57rrruRHR++9/nUXPD09GTt2LH
PnzmX48OFFqqqzP3m1eodvt+tgPSSs2ApiGjmRuoPRRZgoQWEwdFSWEiSzgVzmyzMOUnMU3ZEG...
164dAMnJydSsWdPJe1A+aUBXysUU50q6tBSVHtfb2xuwBuqMjAwAPDw8yMrKspdJSUm
xv/bx8bH3mxtjGDFiBK+++mppNb/c0j50pZTTeXt7Fzs9bmBgIDExMWRIZXHo0CE2btyYb7
kePXoQFRVFfHw8YM3F+hW0QFRVFfHw8YM3FfuDAAefuQDmlV+hKKadzc3MjKiqKhx56iKSkJDIyMnjkkUdo0aJFg
et07NiRRo0a0ukC7JrHg+BGGMaKrEFf1+mG7fGiV2Q7+L3MrYzNl7f8lmTkkWGgvqcP7ntAR8f+xSCOs7ChEl0g
gc9L7/w8HCzefPmMtm2Uq5m165dNBMdmsUlVBbKD9L2VcjiFXIoqPi+cjgHMbO63EI5DG6RQkp4sYe9n7hRvtZpbXkJj+lpdnOGr5RpHFYXyMUE6JwyHzh3GYiwTAP2adt+EyuwXpdtvGm1CNVfuNdBvm2/pkv7/Pa2b9TKjN7t2ZJTJbDyaqPv4hJ77w2NFqAHC4ZjwiIo+BSRIW9Dvbe3HjprHg8kECQDFqohaiGrhRAP4N1Q7jWgmx4Vqk0IAt3cKdnP3ajwtbzNz3R3Mwk85h4bMQz4FLK8Vx2xqs91gCUdd7iJzngKpNPdQJVokPAdJQBwcWYKOGxl
ml+vRgTH91aJjnCGjnoh4cQkEixvQVfCh4XgzJfgECMOW8yk+qVLqZtp/6vFOUu2hUoyF/Uw4zq8xyrD8+hyRY3kRsIlqHAAyE8PMZz+k65v/p6Qq1Wz1zIiSKY66e8y+AbJxMHzTp1gZxFTEOfiKrvhHe
gc9L7/w8HCzefPmMtm2Uq5m165dBAcHl3UzIJPld15FJNoYE55fee1yUUopF6EBXSmlXIQ
GdKWUchEa0JVSykVoQFdKKRehAV0ppVyEQwFdRPqKyB4R2SUGjJ3R2SciE/JZ3kBEVonI7yKyTUT6
Ob+pSqmrmbu7O+Hh4Tz55JM0b96c8PBw+0CSW7du5dtvvy2xfefNVljW7c/hMRvfqLsLPkI0MMDbp9WQbJfZYMB2KeXkdBY5dVmVLKAAonyUnQknaDyeuX/12Rp4SiSW6HnqggwZbPiB4o0b9nmx6PmqGcFI4o5P
4z50xiTBnwJDMhTxgCVbK8rA387r4lKqfLmzJkVkrlT7zWz7Z6oVm5K1aaCDdX5hsB3okKk0Hc4J85TIHK8zX76ggP7K7eOC7zWmcbrqqxjbC1y3pu4eijcR0ShxjXKf12q7P1q
EFI4uLiCA4O5t5776Vfh2sqt0dkvqEU92b1xyQ9FFPe6kRk2DmWsKcx85zBr4VGYwTRv2vwuH1j6bXTxZB78bnTFY5NwNCSHTY5G9wF9g+fCrpeXn6ukA43/F+
MrgyKP/dYFDOaYPA9flKTMZWC4i44EKQM/8KhKRMcAYgAYNGhCHS3rUopB6yaNYP4A386

tc6aDRtxw8gxhZZJTk7GYrGQkpLC0aNH+emnnwBrYq2FCxdSqVIlTp48yfXXX09kZCQiYl+
3oDIAsbGxzJs3j//+97/ccccdfP3119x1110MGzaMCRMmMHDgQFJSUsjKymL58uXExsayce
NGjDFERkayZs0aunTp4tTjcaVyVi6XocAsY8yblItle+FxEWhpjsnIWMsbMAGaA9dF/J21bKXU
FyNnIsmHDBoYPH8727dsxxvDMM8+wZs0a3NzcOHLkCMePH6d27dr2dQsqAxAUFITFYg
H+Sbd79uxZjhw5wsCBAwHrHwSA5cuXs3z5clq3bg1Yr/xjY2M1oOdwBKifY7qebV5OccAyEY
AYwxG0TEB6gOxDujkUopxxV1JX05tG/fnpMnT3LixAmWLVvGiRMniI6OMniI6OxtPTk8DAwFypcQ
HmzpqbYJnsVLtg/el1u8sIP8YYJk6cyH333Vc6O3aFc6QPfRPQRESCRMQLADQESCRMQLGAIsyVPmINAD
QESCAR/ghDMbqpQqP3bv3k1mZibVqIUjKSmJmjVr4unpyapVq/JNdetImZ8fP2pV69evnqV6/
AqwDaly4cIE+ffowc+ZMzp2zjth05MgRe5rRrJGI0yxMSGdRkElAxGmAwRGQf8ALgDM40xO0TkRWCz
MWYJ8DjwXF5FOsXpCNNWaVxVEqView+dLBeKc+ePRt3d3eGDRvGDRvGzTffTfTGGhoKOHh4TR
v3vyidR0pk9n3/Offfdd3/Rk/Dx6RJk/D09BXTkHNaSrklvQQgK6UUi5CA7pSSrkDehKKeUiNKArpZxxCRHj88cft
02+88QZTp061T48fP56PP/4Ynd3Q3X28/UGqffnQ3/Nn/v9bKnavbRlHb6qa3/Y+6Q+Ue/vyGQc4
vvJysrq5CaCxcYGMjJkyeLvV5cXBJbdbEhrQlVJO4e3tzueFFibjhHKL5a2SPGhtEaYBPHROmpdhbvtPm7bQtZiowNMoMaxpNYSsMo9zJ2ijbhdwNhZl3JCzpSSEOhYLs96Ee+eJrg9p6iZ/EvTymLHmfu1dZW6Nqfkd4AmSu8FZZYZ0t5fr2C+lTmA6NjpLRL6hhmBdYnQcNQSTcdAh6
exrw0oCulyi0PDw/S0Ol1KrlrQicL75+k1uk1usXsrLg18pN7h9cASSR5r2CZVMB7HqvBaRqrKF3IZQxV1dSHg6dQXRXCexCdt9aEKI2lZKr53nzcAr55+mdVnmXTsx4KCGdEcF
uERERhIaG2lPiTpkyhdff53q1asD0D0KZNG0aMGH06dP5+q+PqE+eqrr3juecYNmzYRW3q
0KED+/btM9GAz9JPeOWnFqqEOpnl1nq9LzgECQkwjmh7zy+24gOQGZD+zqVdbg/5r5JCz7r3nLC8bTNebPuHuZH5Hf2g7tnSb9Kz2gmuWMSbGZtlOgkqqf6xg+D1G2kZVhVijlqsGCtBvYKsjV7kHvgSGEZuzyrGnkPRSzZrAkOjLgEAEJktxWcYHnNu/HBHHC
8MMOGDSMtLY3MzEyHj5Gj9ApdKeUKx/Sp3vxgF77Z0q3XN7mFJ1RNPvr+2hKO1GAp4d7HGIWQjxDhFNSS6/o+7ycesv3X1IeE13Bz67BjW6+Z2AxnHyZxQz/mA6g
CEhobmCs7FsX//fiwWC07udSmm27ixhtvBKXr14EAQAsG7dOv71r38EJhz5g
xr1qzhrrvuAuCmm26Fm0zzXF2S+r5+V1Urn379rzyyitMmTKFAwcOvW6PEYJKzexeoSvlYh
y5ki5NjzyCG3atOHuHu2Hm++2z/Pw8LBLGmbMD7bdtKd3H
RdO/e3V42OjaFie1a5NuG7D70vCpUqFDi/9cq5QZa0wGZwQ8UKWeadWaPpTdY7FHLdEF0a9fs/B7dddzXdzfRzz766KKwo766KN
c++MMeoWWuIHKqgIAA7CP7+syPlr6Wq5duSVn7Ev0x9zqMYp4Mgy1BQ0uACUGB1
Jo1i7Fjx5a4d2bu3LmAtS2nTZNmzS+bXUNZSi1yd1/fyxyWzn0cyd1oxDd2h3xqjUTtCqgJAMTExz
XSkFpfL19/fn7Nmz9u39+eefNGrUgiXolpcPooeMeogBAwbY++2bbzQaS9+ia/+3FraJzrKgdURJ20/iUlnjizUh7+zJ+svDzc1wGNnjdzVMT7zAuuAD08yhR0QWK0Hx+b+ALTs1yA9hHaN1tsfuyAAQN
o1aoVffv2LfaVcmRkJEeo8iEs6eqPUqJ09m1KhRhRhIWF4Xf++nx/+szlSw
F4/vnnGTp0K1atKBDhw72I3K1bRxbx6K4e6Y0Y0zokqq0YOXIkqSjWi8wwKEvWqZEWyE2EcMnDQSePy6
ampfP7553h6elK7dm2m7dmBMO7dm2eeeaZEre9JlJ+o6VkqauQpQFzXdRqRZkXBK2RSciIEwo
oc4el7BSRHSJyee+mV0opVXOLuoi4A9BXsBhYJOILLDDHG7MxRxvpKpjEXXwPqYVqEmJHYy
UFceQKPQLYZ4z50wDPLi+YM50hT5usScRmSFRxF5tubnsbeUhGZmNNPbfNYAg0
FZFfRORXEmbX0UiMkZENvi5lrC5AKqFPvapPaN6H0J00TppnpNGgl42AkHd0rzPf2VzOB4ZPmjNbNmPzRtUoZ0YGJxSdL
twYE16jRg0nbVoppRQ4PCPF8FoNPNzvuXPaIHz5kmoZWrRIJJEG9vECdVNLaeAmzRZXPMtiU3fDLhCbeyzzZo1i3Hjxp
Worr1799KvXz+aNGlCzZtuOOOzh+/PgII3VE3kRi5YkjjAX0T0EREgkTECxgCLMITZhHWq
3NEpDrWLpg/ndhOpdRVIiUlhZtuokHHniA2NhYtmzZtiXY8nbTZuRkeFw2YJcjjS6l1ORd7
kYYzJEZBzwA+AOzDTG7BCRl4BNWzxB9hwYmhxWNpgmtW9RWQnAK8aYXxJKM2K6Xr+Kj2+//ZaXXnqqJtL

Q0qlWrxty5c6lVqxY///wzDz/8MGAdIGPNmjUsWLCA9u3bc/PNN9vX79atG2C96v/mm284
d+4cmZmZjBgxosCycXFx/Otf/+L8eWvmyWnTptGhQwdWr17Nc889R9WqVdm9ezd79uxh
/Pjx/Pjjj9SvXx8vL6/Lc1BKgUOP/htjIgHL8syblOO1AR6z/SilytD3339vH/jBWWrXrm3PWFiQ5
ORkLBaLffrUqVNERkYC0KlTJ3799VdEhI8//pj//Oc/vPnmm7zxxhtMnz6djh07cu7cOXx8fNi+
fftFqXJzypkC97HHHHiuwbM2aNfnxxx/x8fEhNjaWoUOHkv10+pYtW9i+fTtBQUF888037Nmz
h507d3L8+HFCQkIYNWpUcQ/RFUFzuSilnMLX1zdXhsNs2bZA+jhw4Zjmw4cGgwR48eJS0tja
CgIAA6duzIY489xrBhw7j11lupV69ekdvmQK3MOnp6YwbN46YmBjc3d3Zu3evfVIERIS9D
WvWrGHHo0KG4u7tzzzTXXOD0D4uWkAV0pF1PUlXRZGD9+Pl899hiRkZGsXr3aPtLPhAkTuO
mmm1i2bBkdO3Onqq2WrXYunUrWVIZ+Pj45FuHK9F
cLkqpUpeUlETdutbHV7Kz3GoJ1IIrQ0FCefvpp2rVrx+7nzzzjtZv3493333cmjVr2L59+0
X1FlY2KSmJOnXq4Obmb99mbtq19gCEFdunRh/vz5ZGdmcvToRaNKArpUrd5M
mTuf3222nbtq19XFCAt99+m5YtWxlWFoanpyc33ngjvr6+LF26lPfee48mTZoQEhLC+++/T3
7PrhRWduzYscyePZtWrVqxe/fuAq/KBw4caF
kpdpTSgK6WUi9CArpRSLkIDulQJKuQgN6Eop5SI0oCullIvQgK6Ucoq86PzXxPzExMTg2jwv//9r8i
ys2bN4u+//7ZPjx49JWi4WWLVuWL68rpSUuxrQIVKXzbx58+j
UqRPz5s0rsmzegP7xLRyxTUHM4esg6t2rWrxPU4mzNT+GpAV0o51dGJR+dHSp
Yv9Cnjt2rUAGGNYsGAbNYsYycWzbdbKGoNm9N4scffyQ9Qj4PwBEQMTJKFkK5+uaqQ
YBBoayv79++11ARfIxa++CMi0oVHrGzEq2fAsj6gagqcKczb4ytrzb2hcx4gdEuvQy5mGBN4UN+C19gL+P+2vm76dv0/L/4Fg
rBtO06XMOIf3iiy/o06cPzz77LJm//1fO6Bh
wPWPyxDhw61L8vMzOTKeu8ecOGmPm0Tgs+Y3GI+5
BERPL9C6KUUqp0OBLQNwFNRCRIRyAlcS7lXGmCRjTHVjTKAxJhD4h7jUUu1TWMdJsH0S7DJwPiKD5L7FEoVyM
CglAA6duzIY489xrBhw7j11lupV69ekdvmQK3MOnp6YwbN46YmBjc3d3Zu3evfVIERIS9D
WvWrGHHo0KG4u7tzzzTXXOD0D4uWkAV0pF1PUlXRZGD9+Pl899hiRkZGsXr3aPtLPhAkTuO
mmm1i2bBkdO3Onqq2WrXYunUrWVIZ+Pj45FuHK9F
rBtO06XMOIf3iiy/o06cPzz77LJm//1fO6Bh
BERPL9C6KUUqp0OBLQNwFNRCRIRyAlcS7lXGmCRjTHVjTKAxJhD4h7jUUu1TWMdJsH0S7DJwPiKD5L7FEoVyM

G2MygHHAD8Au4CtjzA4ReVFEIku7gUoppRzj0H3oxphlwLI88yYVULbbpTdLKaVUcWkuF6
WUchEa0JVSTuHu7o7FYrH/vPZavg+UA7Bo0aJcqXAnTZrEihUrLrkNiYmJvP/++8Veb/Lkyb
zxxhsA/Prrr1x33XVYLBaCg4OZPHIyoeuuXr2a/v37I6S5Tqe5XJRSTuHr60tMTIxDZRctWkT//
v3tmQZffPFFp7QhO6CPHTu2xHWMGDGCr776iIatWpGZmcmePXuc0rZsGRkZeHiUTujVK
3SIVKmaMGECISEhhIWF8cQTT7B+/XqLFnCk08+icViYf/+/bkGiAgMDGTixxIlYLBbCw8PZs
mULffr0oXHjxnz44YcAnDt3jh49etCmTRtCQ0NZvHixxfxfVv79+/vrr9vS1zz//vL1dL
7/8Mk2bNqVTp065gnZ8fLw9GZe7u7v9gZ3pDVCGZ1oOHDvkG+oLKzJo1i8jISLp370
6PHj0YPnx4ridGhw0bZt+HS6FX6Eq5Jnk5GR7ZkWoQExMDI8GjRoQExMDI8+ISUIhZ
YtW3L//ffj4+PDwoULqVSpEid7XS1157je3bt9/U/1i+fDmxsbFs3LgRYwyRkZGGs
WbOGChUq8OWXXxITE0NGRgZt2rShbdu2ADz66M0a9aMbt260t260bdvdX0aMGIGPjz/Nmzd
n7dq1eHh4sGLFCp555hfz+/vrrXO0urMyWLVvYtm0bAQEBzzzqzm0OdZ/9F
vExEQqVKjAM888w5o1a3Bzc+Pk4+PlKSMcP378otqWWL1/O8uXLad26dYhzCmdi/PHtfPpZBOK4dfnnNiw5uqStR2n5w6Wts2dbp9Q1UyMKbcmm1S+LN
wWGtW3zlYCubNL0vvWPp7nr0cbMuBq9g1m4KTKbMBpSZ3e8Hd31OuSwB7l6qC62I/OQC7O90I7rD+ELhmfyu+OY7q4rXvF6yTnH6+eR7s4stumbbuFXzH/30d+Kqm69qe5fNDX8trI/DE7L0Yg+6zxbA36y+FuRsIGiX+e3bL6Nz7DWn0HHxYZ+7T0cQThZTxQw/1v6SEnLz9o5y1xc6+vH0nzqRBuvyTuWtr3N9i+u+q0e3Kq+Ly3Ne4ir+bOzxuIvdPKGhaaH5nb3tyKbt6G3F9WPQJFj/WtxFNw5FpGNdddttgRCNjMz69GprHjL8LnxVe0B4LxB5d3NptNSx1fVCiaSWaCnsXy0hMPmVhAfY0m3yTs4f5W60DxxXBDjg8XtG9H8S6oVHv5PNKU1ZmZTTTtrP7sK5h5rx9J4vfgmtG0wr6dzcw9yC9Gsfx+3J98Qo3VbKAE29IMafvm1Ux0y70F6Kwmr3C0KtoymXkxRjpE87dZOY6lIp7t+BuaqVkT+kRy1E8ZgjWK+iNC0p+nxNzgLOFj9E23sPz13wNKAAnfx4TN/bCGkH+rprE4s7JUK33bIbLvS20OiqWBtmgfSg1/vJ+8tfsy8vdbXdRnWd9EEWgk6yOQcYJ4HVf8LHwfkvxw6i+KHMEqfrf5+7bS4ds/ZOR9f/f+/tT+cPvJDsfpTsHMv2bydk0jiBsDqtJkttd5PePiDbfkmkm8lVT77Ji8+fSrVpmXx65f9ZPN4s81Nrl3Ho+9a8sY+G77obn8pGIpkr+1ePhqIMJ4gaRI2obtCf48N5LTaZEBUaz1Tw/JmoeK+0q82z/0W0LbiqDxqxUUPf19hWsxOhQ21tDxBVqaFPhl4SvH/PkWXWTixY3fNHF1k3LAt2v7S6FNbHEaP3fz3u3YGbJYgFiU8iV7qy3qoJM05ZFpC6d48smcQAGPdUbUzPz1uLFgKmH1rXOiy+GdX1hOIC6RJIUk/yGuhQT/Yr7mE7iR0i6vL/GZcb0+Ua7Sbf9DmTDd+dFC2qE5gJaaLpQ2ZPWRHE1rR3U5GZT83Wj1TX2wcSlZdpDqBf8vq/WMmJ5tMw4J3F1Y5kP+bl1j+dB9+bd2vKEW29WuA/n7Ju6vQpv4+LZkY+4B8FtnFB6wgUnmw+6Kd7A6aAZkxGjd1bjTpRc5Ep4sFvG/yIQ7Pvpm+Bu5GDAk8vb4Rk5dxz38X2MAYBIFxK67VE4d0t8wlbWIrnaYVCvlRxdPkD5D3XoVUp5RwIqrfKu9vylY0O2NoqcBbpPXUAA35eK5TUzXpgj9ccLuMUZOCUgmm6WuIUDz2NmHy9f6f2B3LmYvL39y2RTpvpLM5SwEGfN4JwcvT1c7qRzRLqH8tYU4G6DMJ9gcmOF2E53Epdm/XcTdzHfJ1lZqc9QGHj6m/bvSvOxx7JZNyWjXDIYRUP8uJh1x7p/LqmLSIbhMCN8DFEyy0FqgkXnojHiR0aBt7j/E6Vm+4Fe/oqgW2Dm8gmIsDF02XbMx6/H6dA9+Iw8i1VVUWbhAuuKzHSUWQrpQl7DPmgfYa7gsVVPOvyA1zXMtLX4SGTxGXD7oKkRWnYSdJuUrKDMhGsKzjiSZoOH0gqNEA6dH+UYD8hDqq2pqxvwHYyGrAUkOgdKdNNxSi7SvIy0oDm3wh5fdfbFF3xL1M7gKKTQpYLhi8W6jO2aE/NhOpBuNyzgWyD0tq6ZNhdDZLDiwdGvJwFRs1tH/hTWkNvwEVSUX02nSsMQYEFYOMIyuXc76qQZ/HObWkFwD2kVTChu9A3vdFBbI/jwLoEfXs1AZMGbm0yxJmE9X8z/ZBeSsPd0p8Y/oXF6OgfNNPSkxGaKoCxGcExmTR8oHIrGNXmq7rjVAOTQ9kZ7gM8hgBnB5q2YT5+FZGGjRGYw9eSgRXi/jAaLkI2ZR6Bc8XnJzw/4o/s0t+0RHiI9vvRrKajL5/y6eh9eg1+LvqDmuGtiJvzvTYhUIqyRgR0+OHI/2gJl3vGHzhcA9aYBxbZy5Z6p9fH8sTuP+ZdTrtM9S6hm8y0pRlC2i7JeyFbqY0NlNqzECw8nMNzpCJ4W2YRDjMBwFU/lZdoQBMUz/Sc4OTq1kr3X4SS4+DljWcGuA/z9wcFKyTvrYu3u3+PR9v4+/6HOLjkZRYN4UxWa6f8YU6yX4uZ8KbYyJpJ7/+mMbW7w2sHTuwBVWGoAMXI+D8k0MXr/g0LLIC0w4m5r3bEeqD7RMhUjM2nOgL8Lve1CWHb0Zgr4Ag4jRiX6VRJHlnvIW8/HcBWN7oJfAYmMSB1rYgWcn3pZn8mYm2VjVMv5XySbwS5g9m0kpL3E0e2gR9x1CNyw4aYRbpVk/C+oWuaoIpNgHmpn2mNBqCdC+BjrZmKzUBbLHA0C3yXtTvQr3t3cvwA2JCrF7BZxpOu1zPtlm3jnEAKuLyOz3G+R1RrG1L7X5jlHYBU4Owf5S23K4oC1iUkD6SC+6fTSuKxUDHyH1M+h8gvZQ57x6RVrP9QVcDXOkQvEFQGVLm6tGBxoaWUkK4OTVBPs60W5kLXMR46E6cBiu1jBPKh5pOSPqnnVbLPWH1B8/eRRVj73SuE1kXjhlmoVmQP4e8GQ7EWGpsJdQY9Ru3NVT6OwZ2GyBZXYd1dWOGlVNG+vzIrtIVS+hZDfVqfmAa3SlsEC1Eo0NSPwLXTK8ZwJHNxd4MXGgJ8ZcIEyCwFHR3Z1dGJZYgFy3tBqxoW7URR6ePyh8o1OAVy3C0uRWPuvdzbyJp+0iZCG0+d0xvoUHtkJZTFf87Kc3LQJxsyZHp3XdD9yCcPtWf6k6XH1i/3WXRc/PqDUYzpbjm8/kurv0o8I6wJZKHwX2RRMTDAEe8Gm5FbGLmMUOSxjyR1T4oFb9o03uyhQsYoWHkh9NPNRKKcQ2N3IZuH3vVY9dAoAdRl2RcT1j3SW9XCBEVPlEQLcFK74Fj4yPFGNm1E4i7RGjgzAnsXRrPvUyIaKPEoiYmM+4I5q6aVYVoTBl1Ms63F7MAENyRt5GmjXlQZa0e+Dt0mtbmrIIO01HSpDrygjRZ0L2gc2vEBrLaeNNv9o2zdb1X1uTDOSGiRQIxZMKbxQ2DbFcOlvwMmNTbwrMnjMJeNO0Od8gGpzzSN0h3ulwOCRZfkBBXzD5q4FV+SGRrYGkD0CfAHVgB5kk8DfGPq+fR9/O0Q4tIhFGxjg8bpcaIj2ONV8uTs1ThUJv3ZU2wH6Zw+zRmCA3CdHqahZ5Edj5dOWgWjvkUAvcOFgmZvw2/w9ZVWF7CN0J2yaCeTLHUSf+H1xFLaOX5gECWIHH3mhaRkxXeIAVmR5uD5mYLY+xYlR1XOfQIPbJU1B6EXvpo5SiMY+2pnPbLCq0ScHexbnwYeLPZ4V40+7yNgqr/DrCRS0udH8GVTJoKuJYTpSJ9K9PtMFHeH1YiY9o8nGAp1PxKKBsvOlD+Sn8iYC+rIG3K7tJ5rOS2lY1YuUi1lhV8TGmeZH5LyKAj+xLJMcPQwk51RUWjdQHPe+2YMcs4lMiECvMkBq7Vqrj2zs4BN3dSLSG9zbzbFCB7nAYUc2K6sr/30B+QKxOzm+nAj9bWoYy2ofjTlXZcaUqBCn0eP2FvDfHHl/t2EYbBdmjTHB06z5DCYR9Sh+WozBHSlqPq5hMRa1sNOIvTd7W1SL2s0cYV/u4i7b5dY3+UTKEnZfq9WqLhGYO6cjx+fj46PKOv9gVVmiMrx4ObF9Dgm3Y1yo4mcuAq9WC4v4ucnTYh1MxuCF+9/i4RIG6dQycbAOJkGOEmFAi/wIQDhTdDZuUeLxl2Lhz4YSVsbTqgVuqXKKQwZRbGxfjvxFmDxJjwKRh4ee8cCeNb7W5tq6iwfYZo8PAy4CSu3ZgpkeIC1dsvqD5MnXxJ7+AXY6ZIpJh7gqOK3cWsiPOZBLOvh1K80mCRhz93iIUCYw1+Ep9d2Mb/1BfRRdICHHR3hXGRoCv5H2XAzR0CXmeGZ2/iRy1jhQHz59kvPYNeAGaVV1pSdqUmQHiPi1HuPNULHm4WaqCFAVYIq2c0A1kcjNWW9WIHrrWj3jJVyJbBzaoXTk0n9d+oXf55wqO8aTpqcwr7cZuDk8mWRX1UA+WB+rqfk8/0+bRFTGVx5JMQLQ6IiWvSIIH5xjOvKGiIH6MoVgkSRiE0+XVRhPbXKp+yDdsKC4jfWmXwYMdsGTqnGYdNqr8zuzkVDfPMbQNOUsEVN7Htx1uJl8WVHsMM8B1k5xNhuBkswqf4Cjg56ocF9LeEQmVNaqN9pGh0xHBb8pZGHMfA8V3C6B2rHxuGYJvEh4TTNwsbT67C6+PHvlKxOQFDkXRrGDGhU4jYPpCPD+jM+rMVX+96N5FD2ZgTQGlqgF2fHVQe2pXDXCo2WfWw+Z3vlXR9zEwhx0EYH3IyY9Vmp7nv8VpSHMIuZ4V9hBzN7W2xnjv4GS/kaX5b9k0CTGEfM2VUZpYKlfO1Ssf8+AlIY7kRbX1lc1dtDeGQC7RgMLZ8Pe1iWEIxCLiPB3gmKdT2cO3cDmhv4Mzf7D9FuXJkO5pXfsLMPEv9gmFwsnJHv7VXptvb9HjHFsOf+6G7wuaxfhKP/mCsqkOaHJiYqO8j8bQm3O40tU/Nd5XJFl6c9TN0sRS+Zmt8KWNDyBxsK4zcS3dDzQhL3CHV2zBQK2AT8Vz2FR7W4BIiZQVt3dz3xV8ejL/mLo/JjN/Zer2FxGN4yK2Fm9dJhtPlfNy2cVN/UD/T/d3BYPLuY44s/zI8cymJoX8ArALZVyGFuzwtCT3D7Hq6fdVZ0k89G4uu6nA8Uqc2b0/SqVEdR5G5gnwFx1x0Dvs9yPsOrpaiodxxvC9uSMOHtx+1Mtg71ypXyZtjgzUXKmSUBWRt3tAObLHiPmJeTufAtkPs4eNKUWBmZADtx3vv7AcFhJmqj3pB5k4tRSwSvLrVoXWwr6IFTcrJVkF8FDyK6JrBmWm8AXTqJLSK7i7Pwa7Ti2zEi4rH4GjMMrZDyr6OP5sWhvGf2H1fRSP5R3RQMcB5q+8U5nBG8rCO5RdZ6PcnuJqTiWajcFJtXo6H3mZcZVQHAkRaZeUXT5tWpWRGVA2m4YQi9zE7yZUIH9ETvvrOSdcvGsuTcdd2FCZgOc+9dnhpnmTb8XuD2gXdBUzMqwkYkqQ87CewXM3f48X2avtNwp8PTQnezdgXqiM6ddWqYzIY1XjZhTVZtfR1ohxEzlmLDGUlTLjzgAoTaJ7o8TAlJvgkyl6PL30VMo2qcuaUvJGmlX8dL6Rb2W+r/NWxHpeSxETrT0jnM8kpLAkoKAa+7S2TCZkn5eGJtq/NqMMmhf7HSTvEAuzf4WgpV1x8S4F/gnrZnh0h8+cTCxoXJF5BdD0RvRy4wB4Zt3u1SZbhKJT9Kx4mWVEd0oT88L/HxJTsmzRWNzqoUNLAAzuYNA+S2cX/BLjGjLmRDiKK+r9lwzLbGeykLLPm3iJRX12hEE9pgNXSGMrkJuzuT3pyKL/Z6mS2BYdN0cRPOEjMn9Bb7mNK2BEhFqkgdh8O4iPQQd6QiJufvP9sXPNVWlrV8VL8ZbW1XqZnN7+Wr8XzzQOf/1hJ1dyEcBwvXN8FM1BqtvnM8bOLE6S6u2tZ/gQx4sOuZzqXuC5oFPTNrLbG+pjFhvaDwoFXTLYEqJ+Ch5vPyvSIRGh8Q3vwi8zXVOnQALo9Hf6rn3fUJh0J6AD/P9+GVfm/8f9K+Bmev6AwJtRYw3z3YXsSFOvFb9HROFk3M6/oIJoRkbe6wThnTG1sKmT9+wzHgTGwdGaW6vzdjwntuwaT9uFP7T3NwDGp5mRIc/5t0vAR6tgnJ4ZQmB7qC3fgH4RyM2q5y6V47NxSK4hOCuGeAnm6HLwtsGK/t+GRhKSKTt27qHsBw2j6FTqXHr9w+cZ/JlM8yC8l+oC7H3/97ZlOXLJN+pQwfXMzz6XzVdU8eVaEIWVz0mRbUmmgR1vQqWQ6ZDRYuEdX2g4uG4+gfF/0L5bmdrJ9K8Bsup6c/p9zcLNvbqZoQyUwm0Us+C8YqXI/SmU2uw3hhPbqG9/iwH6LQQOrfB9SO0MZF5rJz1L/cQm91PoU7sWkS5mU5ZPrJw2mJiuvv2Yz/GHn3vWC6Hqyu5MAQX/K4IuCZ0upjWnWOYl7QzE0uf3eMCjoWBdRQpH0ND0cHoPa6CwBcWDdJPdFEeN7zD9CAGEhxiMrRWxYvd7ziZUOR4t4aaMoUD4HWlUiy9vlOMM8C3/Fag3LBMW3xK0hAB+mF8RMbRQcAGnW9wWbrbvnzj6K5TQfjNdC/xQ1BH2Q8+4g6gmdRj3fhzg7/cEEBRN/Xh/fNPD/AIpl+BTp4Y1qBGPQUNK2g4SSxmXPgr+eX7pt1xffrXz3tW8F0rLk9IZMNTZ+I/aTsBWDF6cWq5bKH4oGAFk6Ay5b2tdtbZnU1rhwa1Jz2N+OhSeJ4qZWxVMO/HJGbJZxPx9QK5u0PjTTWRUzMj/qTU9TWIkY9pTnrQWujUeHqU8DvyfG/WrFiZdRZ13GWI5AT2c/nh1dz8NFNcaPhvBX/YKc0mSbKi/B0XaxLYs2VSdUI4uAn3dFfx3sWcRvWa0mRGC/QprjzTz9tQvK5fI4AKCjCp4yWzJlUwETaZaKVJSmGY/HJJfHaRyFAUPdsW+cmvzzO6tWqmU4ShPsvgcKK+i2GVpgYNYdC4sAWiLgvfXSeeT7CDx2RibyiThEnWDhBbjhHXp7oU2X7Bvju9+rPX7Odq6/K3Du+6TpPbwbjmzTuJ0/jdYfk4X6nQp0LSs2j28fyNbQzmvPtL4T5ER+GAV29t1BEvV9PVnXiQzoqhPqTwVt2vvpm2B0oN8hEKXycJKaVQXmGzXtfKX8TEfLe5MxjI5R2p1dQzLwldUnsvyI/Z5bW5xV+RfY3Hr3VXUn3l8+PuHr8x8cbRc23FNFZTWnLZDLkKyO3Rxo8wHM5Jbmw5x5KfaoLs3PnWTVHyMSv+XV3tmS2Iq3Bl7tLg7d0bMdgHKuMYVjwnE3EP1cg8lzy/SkfBP/oTQmeX8mPdYGxMBZ5gn7Bx3ciLx8mg6xptAdFy8OGg1+DHLv8hD5krS5cEXqWhKiGNpX4/7NRRA5XeE4H2owI20m9B1iB7jq9rnyB4T0i8KOJqMPXNo3sbdd/7VFXoSDrUYvgkrmv0TYqU/wRejS7qJjxjLxoL99e5c3Stb1w2Ai4pk3aZYhMzgv3bB9uzLzGJF1DBf7yyrL4Bk/2NPqK0H+wKRCH7gxhVfWWCDb6Efdh+RZ9X9RR9lHwsn5yl47MhADQ0ODOIsThyOoelpk4V9I4e7jMxUbRaqlGEkg4oKtVnwUt/r2X/QfG9vWb8p7D/dFxRN3ZgrDbIxw3O4b+lNOpyYfRvm3iV3d4i4w0s3Yq/xHXr2OXEpFf7VuPnSglTNXiDsLvkWSiMWfLsnE8QUFu0IEMWl1J1M5kx6r+CSQJvWv9w7cYlCdvlE2YvzvDn0cQhuGe9HzwXUAW0ocD09gXM+94uwQxSQIZ9Wy1dxkEu3YdR+JTmRCe1ESvYlcPlbc3uZlQVT7jWYebPJKcrlaALmNHWMcZR6pyTfc8Q/6D5afBw+D/XFFYfQa4n+OHDqhc5y7lZ8NPHHVk5qS1z4T2pKc/k6nCk3w0e0bvl7DVfKYQqNuwhPTqbaYL45LEN9IMPtmEd8tVN0EB4ngpBtgRGTMQWwsoRY/ZQzRhCbTeUKmvuUJ5b/1hd7MSUBwl3NYMwEKXJ8xMZQo4zAR1bM5ieKNC6uAhLRPCcE5h1bADzyE2JFlN3DaKkyR6i/LZRiIl8ZDcYTcRQfXh2RhxqK0aN8Q99ThkIlAI4HJFKtIaZlkDTNkiudIMgGbJ9L7MRWcUuvQu7eW1djEN4ju0DyHNqxCjqDbJTVv2V0vB/SeAtJxmMJpQpFk6c4Cat61tRhN1jmMJtvLBqkdCTW7GmmyUmKLqEPJkXfNMtz2SfDmmmdEkZfyIsWVAk8VRdBLPVsMYJQZsrNbVcZ3n9dNRfRD6QDyx9b+nVc7Y3BpMu4DYaS9BLFaPTmfkk0iXr6Yc05cSaY+IkU=

dzx8vBgS9wB4t08aVenJtdVrsiKhCTOJZzENy2FDV7+jKhdIZVnUuhbvTK7zifzx9Ik2lWuwM
HkVIwlYRV9SUjPoImfD8dTUjmdIkHDir58unQpdfZso/ujE1I1+Cgd/H3ZmAF/p2ZwbUVfgny
9Sc/KJMjLgzQ3dzKB46nprD99jv6VvPg7Pp4K/pU4lZrGtqRzDKxemV/EiwA36F2tMuuOHM
WvWg1qenny6ZETtKzgQxV3N8TdneTMLEhPp92ZeCo1ac6mxLP4HDtM1J79vHzLzfxw9A
SeGeksPpNKiwrehFSphI+7G0dT0kjatoXAlmGkennTys+bzafP4ObhQQ1vb46npVPH25NT6
Rn4HdzPnko1aBZQmUZ+PpzNyGDjqTO09Pflr/OpfB9/mnsa1KaCpzs1fLzZGX+S+gFVycrI
wM/Tg2927uHaBg1xy8zg3IE/CQsNo6GPF55uwrFz59mQeAE3L0/6VanAydQ0MpIv4FapCi
cTTuJTpSqn0zOp7uVBcmYWe86nICaLBm6G+HPniahfI62JZzh8/DidGtSjgo8PNb08STeG
5NRULqSkcNTNkxYVfPnI72Ngsjhu3PByExpWqoSvu1DPww0PN3c8PNz5M/4Ea08m0cjLn
fNVAvBxc6OShzuN3LJw9/Vjz/kUMo3hBn9vlh8+Roq3H82Sk8ioVRcPgZQsQ1pWFkdS08I
MTaVKegp9GtZj04GDJAfUZPWxE3Rwy+B0UhIXKIWlfs2a1PH2pJKb8PvhI6QnJ1O7kj+JFS
vTJ8CfvedTSAGOpKQT7OdN7PlkrjEZ1KxQgYOJiaR6+eBzKp5zaem417yGoCr+1Pby5K1tu7
m9WWMuZGQiIny37y+GBtaIUUAV1p0+i4+bG9f4eHIgOY3UzEwS4/ZzbdPm7E9OpUOViqT
+FUv8hWR2Va5Btzo1cRM4GX+Cw94V6FOzKlvOXKCiuxtpxuAhQoCnBx4Yip1IonEFHxLSM
6jqJtT08qSSu3DwyBF2x+7IfIVKeNWui3fSKYKaNsOkJPNXehbxmXBTRhKb3XzpVKs68RIZ
VPPyIOZUEmnuHtT38cLXzY1anm58d+BvEj19GFmvBrsupBBRuQLxaRkcPp9Mgwq+HEtN
p00IP9INpGVIMe1gPNW9POhXozK/Jp7nVMJJ2vv7csDdm/jf1nJzz15sT06nf53q2AYDciqH
BrgQkb7AO1hHLPrYGPNanuXewGdAWyABGGyMiSuszpIOcHH61CmCtx4s9npKKXWlGHP
2CC9G3ISidS9pgAsRcQemAzcCIcBQEQnJU+we4LQx5IpgKjCIRC11QOdNe0qraqWUuixm
+NctIXod6XKJAPYZY/4EEJEvgQHAzhxIBgCTba+jgGkiIqYUcnm22PMHG5IaOrtapZS6bDLr
+ZVKvY4E9LrAoRzTh4HrCipjG1Q6CagGnMxZSETGAGMAGjRoUKIGh6cksL5i4wKXe5gM
MuTi3RIMBrnodX58s1JIdnM8D7GXSSNNvBwun5OPSSVFvPNdJlj/HlbOPEuieyUA3EwWW
VK877LdTSaZ4l5omSqZSSS6V75ovrdJJbWA9nmbNFKL2O/C9i9bfuejQtYFzruVzpv+UITJT
OKMmz9Z4kblzLOkuHmRKt5FvqcK4oYhqwTrAdTMOEm8R/WL5vtIJXPBzbdEdTqqYtZ5zrI
VAKzvETdjivWZuVSOHG8Pk4HBDQO5PjN5j3IJ3muFfS7+KVPw5+O2k78CnYu1TUdc1i9FjT
EzgBIg7UMvSR1PPjWRJ53aKqWUutx6IEqtjIzqHQHq55iuZ5uXbxkR8QAqY/1yVCmI1GXiSE
DfBDQRkSAR8QKGAEvyIFkCjLC9vg34qTT6z5VSShWsyC4XW5/4OOAHrLctzjTG7BCRF4H
NxpgIwCfA5yKyDziFNegrpZS6jBzqQzfGLAOW5Zk3KcfrFOB25zZNKaVUceij/0op5P0Ogdo
OOIxNp+V7XNFxF517bf20SkTY66RtjKx4rIiIK2eaUQEXcR+V1EItqmg0TkN9u+zbeIaUZEvG3T
+2zLA3PUMdE2f4+I9CmbPXGMiFQRkSgR2S0iu5FdmeY57TzqultBWRP2zrvCsiRQ9tZYwpNz9Y0/fuBxoBXsBWIKSs21XCfakDtLG99gf2Yh2E
+z/ABNv8CcAU2+t+wPeAANcDv9nmBwB/2n5Xtb2uWtb7V8S+PwZ8ASy1TX8FDLG9/hB4
wPZ6LPCh7fUQYL7tdYjt3HsDQbb3hHtZ71ch+zsbGG177QVUceXzjHVlyr8A3xznd6SrnWeg
C9AG2J5jntPOK7DRVlZs695YZJvK+qAU8wC2B37IMT0RmFjW7XLSvi0GegF7gDq2J5jntPOK7DRVlZs695YZJvK+qAU8wC2B37IMT0RmFjW7XLSvi0GegF7gDq2eXWA
PbbXHwFDc5TfY1s+FPgox/xc5a60H6wjXq0EugNLbW/Wk4BH3nOMNQd/e9trD1s5yXvec
5a70n6wjt71F7YbEPKeP1c8z/wzxnCA7bwtBfq44nkGAvMEdKecV9uy3Tnm5ypX0E9563LJ
b8DqumXUFqex/YvZGvgNqGWMOWpbdAyoZXtd0L6Xt2PyNvAUkGWbrgYkGmMybNM5
259r8HEge/Dx8rTPQcAJ4FNbN9PHIIIBFz7PxpgjwBvAQcBFz7PxpgjwBvAQeAo1vMWjWuf52zOOq91ba/zzi9U

eQvoLkdEKgJfA48YY87kXGasf5pd5r5SEekPxBtjosu6LZeRB9Z/yz8wxrQGzmP9V9zOBc9
zVWAA1j9m1wAVgL5I2qgyUBbntbwFdEcGrC43RMQTazCfa4z5xjb7uIjUsS2vA8Tb5he07+
XpmHQEIkUkDvgSa7fLO0AVsQ4uDrnbX9Dg4+Vpnw8Dh40xv9mmo7AGeFc+zz2Bv4wxJ4
wx6cA3WM+9K5/nbM46r0dsr/POL1R5C+iODFhdLti+sf4E2GWMeSvHopwDbo/A2reePX+
47dvy64Ek2792PwC9RaSq7cqot23eFccYM9EYU88YE4j13P1kjBkGrMI6uDhcvM/5DT6+BB
hiuzsiCGiC9QukK44x5hhwSESa2Wb1AHbiwucZa1fL9SLiZ3ufZ++zy57nHJxyXm3LzojI9bZj
ODxHXQUr6y8VSvAIRD+sd4TsB54t6/Zcwn50wvrv2DYgxvbTD2vf4UogFIgBBNjKCzDdtt9/
AOE56hoF7LP93D+sd4TsB54t6/Zcwn50wvrv2DYgxvbTD2vf4UogFIgBBNjKCzDdtt9/
P8DpFzwID66tHYAAAAASUVORK5CYII=\n"

```
            },
            "metadata": {
                "needs_background": "light"
            }
        }
    ]
},
{
    "cell_type": "code",
    "source": [
        "sns.lmplot(\"Age\",\"NumOfProducts\",df,hue=\"NumOfProducts\",
fit_reg=False);"
            ],
            "metadata": {
                "colab": {
                    "base_uri": "https://localhost:8080/",
                    "height": 424
                },
                "id": "9-DiMJeakxqr",
```

"outputId": "c2d3e9ff-781b-4be8-97a3-12474c30e70a"
},
"execution_count": 7,
"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":
"iVBORw0KGgoAAAANSUhEUgAAAbEAAAFgCAYAAAA1skc7AAAABHNCSVQICAgIfAhkiA
AAAAlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZX
JzaW9uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3de5hd
dX3v8fdn75nJZDIJuQ2KSSAgKAgiyjyIijSl9RS5aZVWqFJQLFZpxXrpsZ5HRTz2oBxB8YLCA
QripVqviIhSxQaOFTpEFELQoxAkMZKEXMhkkszM3t/zz1oz2bPntmdmdm39bPntmdm39bk8+KZZ/a67LW+
e++V+bDW+u3fTxGBmZIZFuUaXYCZmdl0OcTMzCyshHGJmZpZZDjEzM8ssh5iZmWWVS6
MLmKrTTz897jrjjkaXYGjW6gANJ5s7Etm7d2ugSzMysSWQuxMzMzIY4xMzMLLMcYm
ZmlIkOMTMzyyyHmJmZZZZDzMzMMsshZmZmeUQMzOzzHKImZIZZtW82ylJeaAH2Bg

RZ5UtmwN8HjgReAp4XUSsr3VNzWTX6tVsu+FGBjZsQPPmIYliby+ty5cz96ST2HPffQxs2E
Dr8uUUlth3//1QKEA+T+cZZ7Diqo+N2s5Yz1188ZuYf+qpADzxnr+n9/bbk+1I5BcvJtfePvk+y
7Q85zm0LVo0vG7/9u0M/upX+1fI5UAiN28edHZS/N3v9i961rOgt5fi7t2Qz0MEFIvk5s2j4w/
+gOLmzRXVQGcnuVyO4u7dqK2N3KJF5KSpvQdl7yXA5s9cw/abb062m88TJfUtuvBCDr70
bWOvW1ZD6T6rqfzzrtV+zJqdaj2ys6R3At3AgjFC7G3A8RHx15LOA/40lI430fa6u7ujp6endg
XX0a7Vq3nyig+jtjaKAwMMbtoEEbQsexaxr5/C1q3ku5bSsngJ+x5/HHbvHrWNzrPPZuHZZ
w1vR+3tDD711Ijnxt69RH8/z/jA+9nxndvo/c53Rm1HCxaQ65hLYctW8kuX0rJkCfvWr4e+volf
xJw5zHn2s9n3m9/Avn1TfxPy+f3hIIYehQJaeBBtz1o27usepbUFBgaTzSxdilryI15LJe9B59In
s+Kqj7H5M9fw1DXXJLVEjKwvI4MIIrztbRx86dtGritG1NCyYMHwPqsZMKXHjdrbR7w2B1ITc
N+JdVTTy4mSIgNnAtePs8qrgJvTx18D/kjSAXMAbLvhxuT/3OfOpbhtG+RyKJ+n+NQ2orcXJ
GJXL5JG/iGXkh+g9/bbR2xH0qjn5ubORW1tbLvhxuTsY2gbJeLpp4ld6fN6031OFmAA+/YI6
04nwGDk2VWxmIQGEE/vGv26J9xOcfg1FbdtG/Vaxn0Pyt5LgO033wwSamkZVZ9aWkBK1
hm17sgaSvdZTeWfd632Y5YFtb4n9gng74HiOMuXAU8ARMQgsBNYUr6SpEsk9Ujq2bJIS6
1qrbuBDRtQezsAxf7+5A9gLkf09yfT+TzR3z/xRgqFEdsZ3lbZc9XezsCGDWNfkktF+rziZPus
paErA8XxDpIJnpc+d6zXMul7kM4fvsQ5nnw+Wad83bIaRuyziso/71rtxywLahZiks4CNkfE/TP
dVkRcFxHdEdHd1dWheqaQ+vy5cTevQDk2tqG7wuprS2ZLhRQW9vEG8nnR2xneFtIz429e
2ldvnzCP85Kn5ebbJ+1NHSGmJvioVl6ZpnLjflaJn0P0vm5efMmDHsKhWSd8nXLahixzyoq/
7xrtR+zLKjlmdjLgHMkrQf+FThN0hfK1tkIrACQ1AlcRNLA44Cw+OI3JWdde/aQW7w4OYMoF
MgtWYw6OyECze9MGhWkfzSBJOzS/+vvPOOMEduJiFHPLe7ZQ/T3s/jiN9F5xhn7t1FCCxa
g+enzOtN9dnRM/iLmzEnWnTNnem9CaaAM3RMDtGD+6Nc94XZyw68pt3jxqNcy7ntQ9I4
CLLrwQoggBgdH1ReDgxCRrDNq3ZE1IO6zmso/71rtxywLat6wA0DSKuDdYzTsuBR4fknDjt
dExJ9PtK3Z1LAD3DrRrROnx60Tm9oBc1+/GdQ9xCRdAfRExK2S2oFbgBcC24DzIuLRibY12
0LMzGYdh1gd1SXEqskhZmZNziFWR+6xw8zMMsshZmZmmeUQMzOzzHKImZIZZjnEzM
wssxxiZmaWWQ4xMzPLLIeYmZIIlkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVImOcTMzCyzH
GJmZpZZDjEzM8ssh5iZmWWWQ8zMzDLLIWZmZpnIEDMzs8xyiJmZWWY5xMzMLLMcY
mZmllkOMTMzyyyHmJmZZZZDzMzMMsshZmZmmeUQMzOzzHKImZIZZjnEzMwssxxiZm
aWWQ4xMzPLLIeYmZIIIkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVImOcTMzCyzHGJmZpZ
ZNQsxSe2S7pP0c0lrJX1ojHUukrRF0gPpz5trVY+Zmc0+LTXc9hHgjyNik6QW4B5JdyS
IRf1PDOszMbJaqWYhFRAC96WRL+hP16anpPTFJe0gPAZuDOiLh3jNVeK+kc4GHgHRHx
macU427IEUo+kni1bttSyZDMzy5CahlhEFCLiBGBhbGA5cJKk48e0G8H5cJKk48pW+Q6wMiKOB+4Ebh5nO9
RHdEdHd1ddWyZDMzy5C6tE6MiB3AXcDpZfOfiOh96eT1wIn1qMfMfMzGaHWrZO7JK0MH0
8F3gF8EjZOoeUTJ4DrKtVPWZmrKtVPWZmNvUsnXiIcDNkvIkYfnViLhN0hVATAT0TcCrxd0jnAILANuKiG9ZiZ
9ZiZ2SyjpBFhdnR3d0dPT0+jyzAzG48aXcCBxD12mJIZZjnEzMwssxxiZmaWWQ4xMzPLLIe
YmZIIIkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVImOcTMzCyzHGJmZpZZDjEzM8ssh5iZm
WWWQ8zMzDLLIWZmZpnIEDMzs8xyiJmZWWY5xMzMLLMcYmZmllkOMTMzyyyHmJmZZ
ZZZDzMzMMsshZmZmmeUQMzOzzHKImZIZZjnEzMwssxxiZmaWWQ4xMzPLLIeYmZIIIkP
MzMwyyyFmZmaZ5RAzM7PMcoiZmVImOcTMzCyzHGJmZpZZDjEzM8ssh5iZmWWzUJ
MUruk+yT9XNJaSR8aY505kr4ibqijR7vN8kr4i6deS7pW0sIb1mJnZ7NNSw23vtWe
diYHtEHCnpPOCjwOtqWNOE7t5wNzetvYmNvRvZM7CHHf07KEaRvPKcvvJ0rjz1yuF1P/vA
Z7Il3S30DfQRERQpDi876qCjWDR3UbKd/j3sGNi/HYUYZLCienLkRmx3PHnIOWLBEfx+z+/
pG+ijo7WDXf27pv4GNIG88ixoXcD2/u0z3taitkUctfgoNvZuZFnnMrrmdrF642r6BvooRGFK2

xIipxwdrR1ccMwFvPWEtw4vKz0W2vJtLJqzCEks61xGRLBm85rh/bWohSDoaO3g6EVH88j
2R4Y/s/Lp0v28d/V7uWP9HRSiMOp4LD1uI3Uuo/sZ3fQ82TPu9EXHXsTLI798Ru9t+T6rsU2
z6VBE1H4nUgdwD/DWiLi3ZP73gcsj4j8ItQC/B7pigqK6u7ujp6en6jXeveFu/vHef6Q138rW3
VvZNTg6BM48/EyuPPVKPwAZ7n2F9ciicHi2IE0JzeH1IwrvYO9Va91InNyc9hX3FfXfTazHD
meu/i5bNq9iR37dIT8Pwbjacm1EBG85fi38NYT3jriWCBgMJLjYemcpfQN9tFX6Bu1DaX/FSk
iRFuujYHiwHBdc3JzKFAY3s/jTz/Odx/77qjtnHn4mZx5xJnDx217vp1te7exdc9WIs5dyuL2xa
Om9xb2MIAY4H0vft+0Q6f030p7vr0q25xI1OgCDiQ1vScmKS/pAWAzcGdpgKWWAU8AR
MQgsBNYUsuaxnPT2ptozbcyt2XuqABTekzesf4OAG5ZdwuSaNH4J7L7ivvqHmBA8sfUhh
UpIomn+5+uyvZa1IIkbII3CzDyWChGcfhY2da/bcwAAwj2n7kHgaQRwTq0vaH9DB13KvkPk
uOx9LiVxK7+XRP+ntsyl9Z8KzetvWna70H5PquxTbPpqmmIRUQhIk4AIgMnSTpuOtuRdImk
Hkk9W7ZsqW6RqY29G2nPt0+4ztAIob6BPvLka1KH1UYxkpCYyVnYkDx5+gaSgCo9FoL9F
xCG9IeN/Yx36bMQhVHH7UBxgDx5BooDY04DtOfb2di7cdp1jfVvZabbNJuuurROjIgdwF3A
6WWLNgIrANLLiQcBT43x/Osiojsiuru6umpS47LOZewt7J1wnbySP1YdrR0UmNo9FWusnJJ
DPVeFQ75AgY7WDmDksaCSq0hD+6vGfoaOu3J55Ucdt625VgoUaM21jjkNsLewI2Wdy6Z
d11j/Vma6TbPpqmXrxC5JC9PHc4FXAI+UrXYrcGH6+FzgRxPdD6uli469iIHCAHsG9zC/Zf6I
ZUP/h336yiSDLzjmAiJi+P7HWObk5tDZ0Im7gsfRoLevaeXIEREsaFtQIe0NxiARwQXHXACM
PBZyyg0fK4vbFtOR7xhzG0LDYSpERIwI16HtDe1n6LiLkv8gOR5Lj9uIYH7b/AI/7xncw0BhgIu
OvWja70H5PquxTbPpqInDDknHAzcDeZKw/GpEXCHpCqAnIm6V1A7cArwQ2AacFxGPTrT
dWjXsALdObAZunejWibOAb0zXUV1aJ1ZTLUPMzKwKHGJ15B47zMwssxxiZmaWWQ4xMz
PLLIeYmZlIIkPMzMwyq6IQk/RsSXPSx6skvX3oO2BmZmaNUumZ2NeBgqQjgetIetn4Us2q
MjMzq0CIIVZMO+j9U+BTEfEe4JDalWWmZja5SkNsQNL5JF1E3E3ZbOa51gfTMzs5qrNMTeCL
wE+EhEPCbpcJLuoszMzBqm0pGdXxERbx+aSINs4i7fzczMaqzSM7Ex5h3URXrMDMzm7I
Jz8TS+2B/ARwu6daSRfNJep03MzNrmMkuJ/4E2AQsBT5eMn8Mn8X8ItaFWVmZIaJCUMsIh4
HHpf0euB3EbEXhge5XA6sr3mFZmZm46j0nthXXYciOgfmkgf
t9WmJDMzs8pUGmJbJJ0zNCHpVcDW2pRkZmZmZWmUq/J/bXwBcIfZpk6O0ngL+sWVVm
ZmYVqCjEIuI3wMmSOtPp3ppWZWZmZWmVoGKQkzSB8qmAYiIK2pQk5mZWUUqvZy4u+RxO
3AWsK765ZiZmVWuooYdEfHxkseHgZcDR0x1oZKmSfq+pNb/pFN
Klr1c0IpJD0g6RtKe9PHDkj4nqdJGfmPtd72kpjO0zyuwDXjOd
P/iTkXQW8BbgIIg4mqQx3pckPTd5fXAY6Sq/SzcRKkm4NZ6
SiEJP0oKRfpD9rgV8Cn5jpzs3MDiCDwHXA35A35UvkHSFN6fn
mk9KOnZ6VP+O/CeiNgKEBFvkyQ1B2ATU2VAy5DrmAOpUTM+3bm
nfSv/u/1TS8WINSyT9ID2zu56kxfrQmdVDJa9h+ExT8KOTV/
XA+k+j6rkTa00bc8qeTwIPJm+eDMzq9xngbsi4ktl04cBHjI7
/rLn9QAXRsT7gX2GBgfwAHhgGcALwKOj4htkj4Tbh/dNODzwAnAB4F
7luIKSWcCF1fwGr4XkLp4Crl+KLktqAfCVzmS92C9OH+4qW7R
AEhHhnuzNzCoUEUL+jzwdpLLepPkt4nkpCqjRmpLsBVvR8T3JF0E3E3F
nyd/0U4LVp3T9Kz8AWACKCr0nnf1fS9+Z6Kwivpk+Z6gP9XyQ
uZ7Ezs/vRFCjgU2J4+Xgj8Fji8kp2kMY2ymdmwTwTwBrgH8pmTdIeHnsbWRR2q3fvpLHxZLpIvv/hj8
MnAj8qGTdE4G149QwdE+s3O4x5x4kxq5lj4kqR7gTOB2yW9JSJ+NNnJrhwFhGH
R8QRwL8DDZ0fE0ohYYhQnJ58QcTPdfE0ohYQnJ58QcTPdfMzEZLz3S+ysjLcOtJQgfgHKB1ipv9GPBRSUsAJJ1AM
nDxNTMo9W6SBiFIWgVsjYingdwKDTIkvRJYIK7/JHBwesY

GNTks4/Ang0Ij4JfJuk0cmkKr0ndnJE/NXQRHrqOZVrumZmtt/Hgb8pmf4/wLcI/Ry4gymeE
UXErZKWAT+RFCQB8Yahy5DTdDlwo6RfAH3Ahen8DwFfThv5/YTkqhwRMSDpCuA+YCP
wSMm2LgCuTZcPAH9GMiZIIX3NNwFzgAskDQC/B/6xkiIVEZOvJH2fJJW/kM56PXBqRPxJ
JTuppu7u7ujp6an3bs3MKjXqZo/VTqXfEzsf6AK+mf4cnM4zMzNrmEo7AN4GXJa2MgI3AG
xmZs2g0i87P1/Sz4CHgLWS7pd0XG1MZmZm1iIlxOvBd4ZEYdFxGHAu0i+eW5mZtYwlYbY
vIi4a2giIn4MzKtJRWZmZhWqtIn9o5LeHpYBsIbS7Tk5Qn6cdqSn10SVpB0mHkM0i6rrouIq4uW2cVyTezH0tnfcOjRZuZN
bWbgE+T/H1vuEIDLCIKkoqSDoqInVPY9iDwrohYkzbNv1/SnRHxcNl6dw/1YmxmZtWz8r3fP
R14D0k/t48BV62/8sw7ZrLNiFhd2hN+o1V6ObEXeFDSDZI+OfQz0RMiYIM6ps1Q31nrgGUz
K9fMzCqRBthngENIhnE5BPhMOn/WqLRhxzfSn2lJU/uFwL1jLH5J2nfW70jGIhnV67KkS4BL
AA499NDplmFmdiB5D0mP933pdF/J/1eWd3+SdUzQkjbJV+1we9tqpdZbF2GqoNv+5Ajq5YUidJY5B3pD0gl
1oDHBYRvZLOAL4FjBrNMyKuI/1eWnd39+SdPZqZ2eEkZ2Cl+phIQ2hNeDIR0jUkQ2kvIRna
+v1T2bikVpIA+2JEjDqTi4inh7qwiojbgVZJS6eyDzMzG9NjQEfZvA72N6Sba7J3YqcFpE/A
OwCnh1pRtWMmznDcC6iPincdZ5Zroekk5K63mq0n2Ymdm4riIZ3mQoyDrS6atmsIFJXyY
Zhfm5kjZIuniy59TSZJcT+yOiABARfUOBU6GXkYwh82A6DDbA+0hGiCYiPgecC7xV0iDJUN
3nRSVjw5iZ2YTWX3nmHSvf+91LqX7rxKYawWTC8cQk9QG/HpoEnp1Oi6Q3+4pG3qwmjy
dmZk3O44nV0WRnYi8iOUMyMzNrOpOF2Jci4kWSbomIC+pSkZmZWYUmC7E2SX8BvFT
Sa8oXjtXi0MzMrF4mC7G/BI4PLATOLIsWzOAL0GZmZjM1YYhFxD3APZJ6IuKGOtVkZmZW
kUp67DgYOEzS19JZa4HPRMTmmZmmIZmZmZNp5IRSuppsh47Xgb8F0mhn2d/1/v3pcvMzOz
AMjRCyfOAk4FLJT2vUcVMdib2ceDVEfGzknm3SvoZ3UEhGPtCMrvPwapmtQ+5OGn8jVWg
TP9svMmYFP6eJekoRFKyofZqovJup1aUBZgAETEA8D82pRkZmYzIgTYqKFY0vlVMckIJXU
xWYhJ0qIxZi6u4LImZtY4Yw3Fsi+dP2OTjFBSN5MF0T8DP5D0B5Lmpz8bAO5lI
4AG1KVoVgmG6GkniZrYn+dpN8BHwaOYn1sWZmdm0PUZyCbE0yGY8FEsII
5TU06RN7CPiNuC2OtRiZmbVcxXJPTFIgqwqQ7Ewzggl6ZiQdTdpiAFIOhz4W2Bl6XMi4pzaI
GVmZjNy+c47uPygUUOxVKF14j00UU/9FYUY8C2S08fvAMXalWNmZlVRpi
eyPikzWtxMzMbloqDbGrJX0Q+AFJE00AlmJmNTaoyMzOrQKUh9nysf9yMmnsf9yYqTTZmZm
DVFpiP0ZcERE9NeyGDMzs6motJZ6JLQQekfRjEntw
GqS75y1AF+LiA82qp5KQ6xhZqZWVPZB5wWEb1p91P3SPpeRPSPpeRPy0EcVUFGIR8R+1LsTM
zKrr+Tc/f9RQLA9e+OBMv+wcQG862Zr+xEy2ORMV3ROTtEvS0+nPXkkFSQ3rtdjMzCaWBt
iooVjS+TMiKZ92ObUZuDMimnYoFgAiYn5ELBIcBc4LXANTWtzMzMZqJmQ7FERCEi
WAydJOm6m25yuKY8JFoIvAX9Sg3rMzKw6ajYU5CI2AHcBQ4NeAbm
BvTSoyM7NqqNVQLF3AQETskqxeAXw0ZIscsyYqbZ14dsnjQWA6FYD
gFuIpQnOan5ajpkV0MoaWiSHd3d3dHT09PoMszMxtM0w5TUonVis5kwxCR9YILnRkR8uP
oITcwhZmZNrmIC7EAw2eXE3WPMmwdcDCwB6h3Av8K
fHy855mZmdXDpA07JC0G3keE5E9E6
0vpmZWT1N1rCjSPIN70FG9o0IkoYdC2pb3mhu2GFmTc4NO+posntncRw8zMZrf0O2I9
wMaIOKuRtTikzMxsqi4D1jW6H9jW6CKhhiEIaIekuSQ9LWivpsjHWvsO9CsO/
qY09cdfcwP1x19zKPp72r0YL8cOBO4fuYVzlyI3U5NxyDwrohYkzbPv1/SnRHxcMk6rwSOSn
9eDHw2/V09Iy8CiIXdpNkBaW4X7NsGxcLYy6If9vXCnE446nTY9TvY8XhyN73vKRjcmyybvw
KeWpdsJ5eHFS8DRbLuwsPgpZfBc14xdg2/uhN+cvXk61a63kz20eTSwPoMSbuG4aFY1h1
9zKXHPLJuJr12fAL4e2D+zKucuZqdiUXEpohYkz7eRXLquxstVcBn097xv8psFDSIVUrwgF

mVj17towdYEPL9u6EXGsSZA9+BTathf59sPO3MLAbIEvW2fLQ/u0UC/D4atjwM2hfBLuehO
+9OwmScr+6M1m268mJ1610vbHM5LnNp+pDsUg6C9gcEffPvLzqqMs9MUkrgRcC5QOn
LQOeKJnewOigmwEHmFId5cRwQ+b+nbBna7pAEGUBqJJGfIO9yXRbB+TakjOhcj+5OInW
1jHxupWuN5aZPLf51GIolpcB50haT9LpxWmSvjCD7c1YzUNMUifwdeAdETGt0aAlXSKpR1L
PIi1bqlugmVXX0Nd2ojAyuCrtbLx1Luz47ej5Ox5Plk22bqXrjWUmz20+j5H0XF9qRkOxRMQ/R
MTyiFgJnAf8KCLeMP0SZ66mISapISTAvhgR3xhjIY3AipLp5em8ESLiuojojurq6u2hRrZtUx
dIalfPJTPn8yA3tg4aGj5y88LFk22bqVrjeWmTy3+VxFMvTKUJBVayiWplLL1okCbgDWRcQ/j
bParcBfpq0UTwZ2RsSm6lXhbxCY1VUxGP6ub9tBMHdpuiBGBhqMPDNr6Uym+/ug2J80pij
30suSZf19E69b6XpjmcIzm0zaeONSYBOwOP0900YdwyLix43+jhjUcDwxSacAdwMPsv/m1
PuAQwEi4nNp0H2aZGjrPuCNETFhdxxT7rHDjTvMqmNarRN/m4RBRa0Tf5uc8VTUOnGSdS
tdbyb7GJ977KgjD4ppZIZdDrE68vU2MzPLLIeYmZIIlkPMzMwyyyFmZmaZ5RAzM7PMcoiZ
mVlmOcTMzCyzHGJmZpZZDjEzM8ssh5iZmWWWQ8zMzDLLIWZmZpnIEDMzs8xyiJmZ
WWY5xMzMLLMcYmZmllkOMTMzyyyHmJmZZZZDzMzMMsshZmZmmeUQMzOzzHKIm
ZIZZjnEzMwssxxiZmaWWQ4xMzPLLIeYmZIIlkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlmO
cTMzCyzHGJmZpZZDjEzM8ssh5iZmWWWQ8zMzDLLIWZmZpnIEDMzs8xyiJmZWWbVL
MQk3Shps6SHxlm+StJOSQ+kPx+oVS1mZjY7tdRw2zcBnwY+P8E6d0fEWTWswczMZrGa
nYIFxGpgW622b2Zm1uh7Yi+R9HNJ35N07fSzPjMza2KNDLE1wGER4STgfwPfHmtFSZdJ6pPU
LgU8C3xlsxIq6LiO6I6O7q6qpbgWZm1twaFmIR8XRE9KaPbwdaJS1tVD1mZpY9DQsxSc+U
pPTxSWktTzWqHjMzy56atU6UdCtwBrBU0gbgb4EswNgEAQQEcxWSuOjxu24a3YYw/WoJFtY
z2admIRYR50+y/tDNzMympdGtE83MzKbNIWZmzKbNIWZmZpnIEDMzs8xyiJmZWWY5xMzML
LMcYmZmllkOMTMzyyyHmJmZZZZDzMzMMsshZmZmmeUQMzOzzzHKImZIZZjnEzMwss
xxiZmaWWQ4xMzPLLIeYmZIIlkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlmOcTMzCyzHGJ
mZpZZDjEzM8ssh5iZmWWWQ8zMzDLLIWZmZpnIEDMzs8xyiJmZWWY5xMzMLLMcYm
ZmllkOMTMzyyyHmJmZZZZDzMzMMsshZmZmmeUQMzOzzHKImZIZZjnEzMwssxxiZma
WWS212rCkG4GzgM0RcdwYywVcDZwB9AEXRcSaatdxxHu/S7HaGzWzEZZ0tLBjb4FCMc
jnxCHz29i0q59CMRCwpLON9tY8KxZ18JIjFvOfj27jie19rFjUwVtOPYJVRx8MwI8f2cy1qx8dX
vbMBW388JEt7O4vMK8tz5tPOZy3//Fzavpaymsorc+ajyKiNhuWTgV6gc+PE2JnAH9LEmIv
Bq6OiBdPtt3u7u7o6empqAYHmFnzOKi9hY45eTbv6qers42InXPYM1BgoBBccc6xAHzg1r
W05sXc1jy/27GHbX0D5ASteVEMKAZcdtqRNQuyH9ekKnwAAAhRSURBVD+yeUQNpfVNI
chUk+JsTDW7nBgRq4FtE6zyKpKAi4j4KbBQ0iHVrMEBZIZfmuDP9869gzy9Z5CcYNfeQST
R0dZCa15cu/pRrI39KK35ZJ4kduwZGH5uTjIacjIyguveaxm9ZfXUFqfNadG3hNbBjxRMr0h
nTeKpEsk9Ujq2bJKM7Pq6y8UySn5PWRua54N2/t4Ynsfc1vzw/OL6UWi0otr8YsWGBuC8WA+e
0tRAR9/YMMFIK3nHoEbzn1CAYKybyIYOHc1uHnFqPlYLFIMeDNpxxes/rLayitz5pTl//O3wr8
pRInAzsjYIM1d/DoIWc6yMzqYEIHC/n01CmfE8sPmjM8LWBpZxckyknI512JIcv7WTnn
gEOnt8+3Ghi1dEHc8U5x3Lw/HZ27hnguc9cwJ+ecAidc1oYLCaX9WrZqAMYVUNpfdacatk
68cvAKmAp8CTwQaAVICI+I+kzwIHrbGOavY9sYg4f5LIA
Vxaq/2bmdns56ttZmaWWQ4xMzPLLIeYmZIIlkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlm
OcTMzCyzatZjR61I2gI8XuLNLgW2W2VnmbM9Fs9UDz1dRs9UDz1eR6JIeLmrZGxOIV3qaNI3
MhVguSeiKiu9F1DGm2eqD5amq2eqD5anI9k2vGmmxqfDnRzMwyyyFmZmaZ5RBLXNfoA

so0Wz3QfDU1Wz3QfDW5nsk1Y002Bb4nZmZmmeUzMTMzyyyHmJmZZdYBFWKSVki6S
9LDktZKuiydv1jSnZL+X/p7UR1rapd0n6SfpzV9KJ1/uKR7Jf1a0IcktdWrpnT/eUk/k3Rbk9SzX
tKDkh6Q1JPOa+TntIDS1yQ9ImmdpJc0qh5Jz03fI6GfpyW9o5HvT1rX36XH9EOSvpwe6w0
7jiRdItayVtI70nkNfY9s5g6oEAMGgXdFxPOAk4FLJT0PeC/ww4g4CvhhOl0v+4DTIuIFwAnA
6ZJOBj4K/HNEHAIsBy6uY00AIwHrSqYbXQ/AH0bECSXf62iOSXf62nk53Y1cEdEHA28gOS9akg9Ef
HL9H05ATgR6AO+2ah6ACQtA94OdEfEcUAeOI8GHUeSjgP+CjiJ5PM6S9KRNPYYsmqIiA
P2B/g28ArgI8Ah6bxDgF82qJ4OYA3wYpJeBFrS+S8BvI/HOpaT/IM+DbgNUCPrSfe5HIhaN
q8hnxtwEPAYacOoRtdTVsN/A/5vo+sBIgFPAIuBIvQ4+pNGHUfAnwE3IEy/H/j7ZvjM/DOznw
PtTGyYpJXAC4F7gWdExKZ00e+BZ9S5lrykB4DNwB3wJ3Ab4AdETGYrrKB5I9CvXyC5B94MZ1
e0uB6AAL4gaT7JV2SzmvU53Y4sAX4I/SS6/WS5jWwnlLnAV9OHzesnojYCPxv4LfAJmAnc
D+NO44eAI4uaYmkDuAMYAXN8ZnZDByQISapE/g68I6IeLp0WST/S1bX7x1ERCGSS0HLS
S53HF3P/ZeSdBawOSLub1QN4zgIll4EvJLkMvCppQvr/Lm1AC8CPhsRLwR2U3YZpEH/RrcB
p/6Rzg38qX1bue9N7Sq0gC/1nAPKBh/QIGxDqSS5k/AO4AHgAKZevU/TOzmTvgQkxSK0m
AfTEivpHOfILSIenyQ0jOiOouInYAd5FcZIkoqSVdtBzYWKcyXgacI2k98K8k8K8klxSvbmA9wPD/2
RMRm0nu95xE4z63DcCGiLg3nf4aSRPS6PfQK4EfR8QT6XSj6nkF8FhEbImI/cA3SA6tRh5HN0TEiRFxKsn
9LDkth6Q1JPOa+TntIDS1yQ9ImmdpJc0qh5Jz03fI6GfpyW9o5HvT1rX36XH9EOSvpwe6w0
Bsi4sSIOJXkftyvaPxnZjN0QIWYJAE3AOsi4p9KFt0KXJg+vpDkDkj9PJfkHt06kjA
7t941RcQ/RMTyiFhJcmnqRxHx+kbVAEnq0Ems/4IeLhR9ZQ
4n/2XEmIwPb8FTpbUkf67G3qPGnkcHZz+PhR4DfAlIGv+Z2QwdUD12SDoFuBt4kP33e95H
cI/sq8ChJMO8/HlEbKtTTccDnyf5hm8OuCkirpL0bJIz0MXAT4A3RMS+etSU1rUG+LuIOKuRtaT
m+v5lOtgBfioiPizzbSAWQ6PMqMZ1ez3E4HK3yObuQMqxMzMbHY5oC4nmpnZ7OjQMzOzzHKIm
EzMwssxxiZmaWVVQ4xm/UkvVpSSGpYThmVhsOMTsH3ck1ckQ36nW1nEI37LSbomHQvsTkm3SzaipP9IOxr+/ICXRGbWnBxiNtu9imTcr18A84jS84A
LSPqqHOpX81PAuRFxInAj1JFG1mlWmZfAj8JFGFG1mIWmZfBWzTDufpANjSLo7Op/kuP+3iCgCv5d0V7r8r
cBxwJ1Jd3/kSYYRvRMbMm5RCzWUvSYpJe+J8vKUhCKdjfD+OopwBrI+IldSRzGbIxNtNjsXu
CUiDouIIRGxgmRE5m3Aa9N7Y88A1qTrPwIslTR8SVHSsRNt2GyqOcRstjufkWdd3yT5Mc8NwFrgZpIbBn4ZEfsj+uBfGxfG1aF0+EBoZfw01dYDOnYk8mkJYTjlyH4JKF90Hz71zfMisZBkj
bWdwFJEm4SiqAoiqzSYVD0bFu4JsdmCk8gYCxH7Mas8FKXUSTRVT9MpO5gIHo7SRXmzyTEHbNE
YYA8DXwDWADsjop8k+D4q6r3H7mtlUuRd7OyBJ6oyI3nQojvuAI6XjhJlZhviemB2o
bksHI20DPuwAM8smn4mZmVlm+Z6YmZlIkPMzMwyyyFmZmaZ5RAzM7PMcoiZmVlm/X
9MncyKP0xsSAAAAABJRU5ErkJggg==\n"

},

"metadata": {

"needs_background": "light"

}

}

]

},

```
{
   "cell_type": "code",
   "source": [
      "df.describe()"
   ],
   "metadata": {
      "colab": {
         "base_uri": "https://localhost:8080/",
         "height": 364
      },
      "id": "5uRUfxrTk0An",
      "outputId": "bf082450-c2ab-41b2-dbde-2478d0dd865f"
   },
   "execution_count": 8,
   "outputs": [
      {
         "output_type": "execute_result",
         "data": {
            "text/plain": [
               "            RowNumber      CustomerId     CreditScore             Age
Tenure    \\\n",
               "count    10000.00000    1.000000e+04    10000.000000    10000.000000
10000.000000     \n",
               "mean      5000.50000    1.569094e+07      650.528800       38.921800
5.012800      \n",
               "std       2886.89568    7.193619e+04       96.653299       10.487806
2.892174      \n",
```

```
"min         1.00000  1.556570e+07      350.000000       18.000000
0.000000     \n",
"25%       2500.75000  1.562853e+07      584.000000       32.000000
3.000000     \n",
"50%       5000.50000  1.569074e+07      652.000000       37.000000
5.000000     \n",
"75%       7500.25000  1.575323e+07      718.000000       44.000000
7.000000     \n",
"max      10000.00000  1.581569e+07      850.000000       92.000000
10.000000     \n",
"\n",
"                  Balance  NumOfProducts      HasCrCard
IsActiveMember    \\\n",
"count     10000.000000     10000.000000   10000.00000
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-5e3f3305-d16c-4e6b-8b55-1b484509a1a6\">\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",
"          <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>0.00000</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>0.00000</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>1.00000</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>1.00000</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>1.00000</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-5e3f3305-d16c-4e6b-8b55-1b484509a1a6')\"\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 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-5e3f3305-d16c-4e6b-8b55-1b484509a1a6
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-5e3f3305-d16c-4e6b-8b55-1b484509a1a6');\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",
                  "    "
                ]
            },
            "metadata": {},
            "execution_count": 8
        }
    ]
```

```
    },
    {
      "cell_type": "code",
      "source": [
        "data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/downloads/Churn_Modelling.csv')\n",
        "pd.isnull(data[\"Gender\"])"
      ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "4_jOo0CDk-Am",
    "outputId": "9e47b12d-5450-43f2-ec68-728c636b332e"
  },
  "execution_count": 11,
  "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": 11
    }
  ]
},
{
  "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": "kuLsHda8IrxZ",
```

```
    "outputId": "12e61b92-5065-41cf-f84d-0a6bc427e8d6"
},
"execution_count": 12,
"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": 12
    }
```

```
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "pd.get_dummies(df, columns=[\"Gender\", \"Age\"], prefix=[\"Age\", \"Gender\"]).head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 299
        },
        "id": "x33HLmmXI6kS",
        "outputId": "3ae8911a-30bd-4c89-c9ea-09dbb39d9e03"
      },
      "execution_count": 13,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   RowNumber  CustomerId   Surname  CreditScore Geography  Tenure  Balance  \\\n",
              "0          1    15634602  Hargrave          619    France       2     0.00   \n",
              "1          2    15647311      Hill          608     Spain       1
```

```
83807.86     \n",
                    "2            3      15619304          Onio                 502       France
8   159660.80     \n",
                    "3            4      15701354          Boni                 699       France
1          0.00     \n",
                    "4            5      15737888      Mitchell                 850       Spain
2   125510.82     \n",
                    "\n",
                    "     NumOfProducts  HasCrCard  IsActiveMember  ...  Gender_78
Gender_79   \\\n",
                    "0                1          1               1  ...          0
0     \n",
                    "1                1          0               1  ...          0
0     \n",
                    "2                3          1               0  ...          0
0     \n",
                    "3                2          0               0  ...          0
0     \n",
                    "4                1          1               1  ...          0
0     \n",
                    "\n",
                    "     Gender_80  Gender_81  Gender_82  Gender_83  Gender_84
Gender_85   \\\n",
                    "0            0          0          0          0          0
0     \n",
                    "1            0          0          0          0          0
0     \n",
                    "2            0          0          0          0          0
0     \n",
                    "3            0          0          0          0          0
0     \n",
```

```
                "4                0                0                0                0                0
0     \n",
                "\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-a9c64075-2157-409f-82ea-a825a6427114\">\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",
```

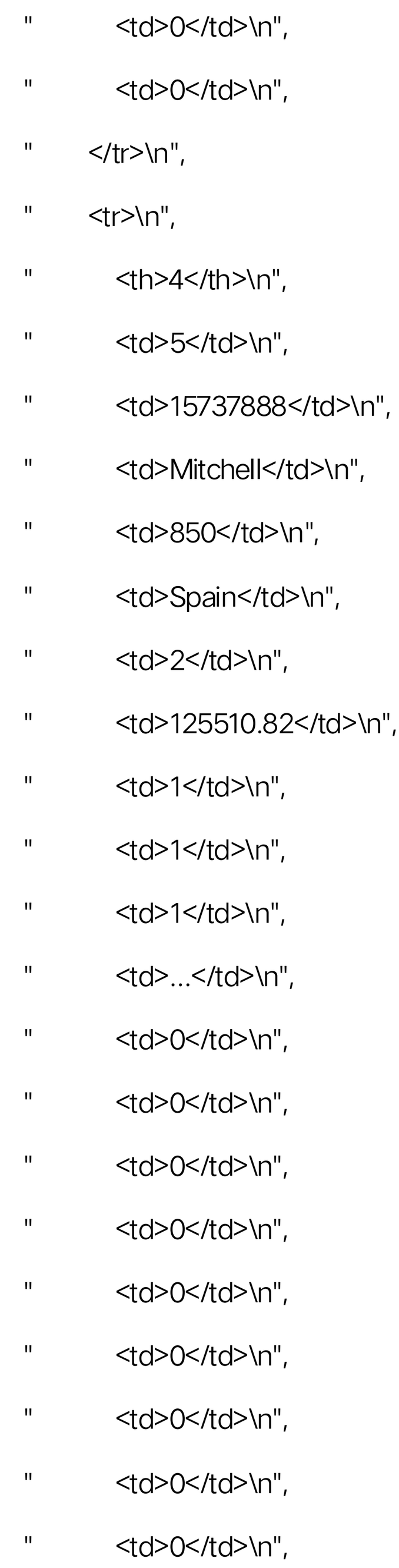
```
"        .dataframe thead th {\n",
"            text-align: right;\n",
"        }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"    <thead>\n",
"        <tr style=\"text-align: right;\">\n",
"            <th></th>\n",
"            <th>RowNumber</th>\n",
"            <th>CustomerId</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>\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>1</th>\n",
"          <td>2</td>\n",
"          <td>15647311</td>\n",
"          <td>Hill</td>\n",
"          <td>608</td>\n",
"          <td>Spain</td>\n",
"          <td>1</td>\n",
"          <td>83807.86</td>\n",
"          <td>1</td>\n",
"          <td>0</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",
"            <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>8</td>\n",
"            <td>159660.80</td>\n",
"            <td>3</td>\n",
"            <td>1</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",
"          <td>0</td>\n",
"          <td>0</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>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",
"    </tbody>\n",
"</table>\n",
"<p>5 rows × 84 columns</p>\n",
"</div>\n",
"            <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-a9c64075-2157-409f-82ea-a825a6427114')\"\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 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-a9c64075-2157-409f-82ea-a825a6427114 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-a9c64075-2157-409f-82ea-a825a6427114');\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": 13
    }
  ]
},
{
  "cell_type": "code",
  "source": [
```

```
    "X = df.iloc[:, :-2].values\n",
    "print(X)"
],
"metadata": {
    "colab": {
        "base_uri": "https://localhost:8080/"
    },
    "id": "de8amCwFmAfS",
    "outputId": "9281cc13-81eb-4ace-e8ee-0b5ef7b16c8f"
},
"execution_count": 14,
"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": "code",
        "source": [
          "import pandas as pd\n",
          "df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/downloads/Churn_Modelling.csv')\n",
          "Y = df.iloc[:, -1].values\n",
          "print(Y)"
        ],
        "metadata": {
          "colab": {
            "base_uri": "https://localhost:8080/"
          },
          "id": "ExCvgw-QmFKp",
          "outputId": "612b517b-9792-4aae-e0b9-56027f6f2d45"
        },
        "execution_count": 16,
        "outputs": [
          {
            "output_type": "stream",
            "name": "stdout",
            "text": [
              "[1 0 1 ... 1 1 0]\n"
            ]
```

```
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "from sklearn.preprocessing import MinMaxScaler\n",
      "scaler=MinMaxScaler()"
    ],
    "metadata": {
      "id": "k5upG866mGjh"
    },
    "execution_count": 17,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "df[[\"RowNumber\"]]=scaler.fit_transform(df[[\"RowNumber\"]])\n",
      "print(df)"
    ],
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
```

```
      "id": "NpHzd2tOmafJ",
      "outputId": "0257c367-71b5-4fe9-962f-6387e7ff5a62"
    },
    "execution_count": 18,
    "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",
"\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",
"\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": "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": "sFseXZHumd4C",
      "outputId": "c2f3700a-770f-49c2-87e1-1bdd31c0c368"
    },
    "execution_count": 19,
    "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": 19
}
    ]
}
```