```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "_uuid": "fd0fc89fe26333add074845d0629b52ae828584c",
    "id": "MoN9mX3FYHNs"
   },
   "source": [
    "## Importing Libraries."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {
    "_cell_guid": "b1076dfc-b9ad-4769-8c92-a6c4dae69d19",
    "_kg_hide-input": false,
    "_uuid": "8f2839f25d086af736a60e9eeb907d3b93b6e0e5",
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 17
    },
    "id": "2BseupTpYHNt",
    "outputId": "ac7b1479-f26f-40d0-fd71-3f24d315aaf8"
   },
   "outputs": [
    {
     "output_type": "display_data",
     "data": {
      "text/html": [
       "        <script type=\"text/javascript\">\n",
       "        window.PlotlyConfig = {MathJaxConfig: 'local'};\n",
       "        if (window.MathJax) {MathJax.Hub.Config({SVG: {font: \"STIX-Web\"}});}\n",
       "        if (typeof require !== 'undefined') {\n",
       "        require.undef(\"plotly\");\n",
       "        requirejs.config({\n",
       "            paths: {\n",
       "                'plotly': ['https://cdn.plot.ly/plotly-2.8.3.min']\n",
       "            }\n",
       "        });\n",
       "        require(['plotly'], function(Plotly) {\n",
       "            window._Plotly = Plotly;\n",
       "        });\n",
       "        }\n",
       "        </script>\n",
       "        "
      ]
     },
     "metadata": {}
    }
   ],
   "source": [
    "import numpy as np # linear algebra\n",
    "import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)\n",
    "import matplotlib.pyplot as plt \n",
    "import seaborn as sns \n",
    "import plotly as py\n",
    "import plotly.graph_objs as go\n",
    "from sklearn.cluster import KMeans\n",
    "import warnings\n",
    "import os\n",
    "warnings.filterwarnings(\"ignore\")\n",
    "py.offline.init_notebook_mode(connected = True)\n",
    "#print(os.listdir(\"../input\"))"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "_uuid": "5b838f778af8cb2740c6d54e5bb6608223ab54f4",
    "id": "YVH1b5bhYHNw"
   },
   "source": [
    "## Data Exploration"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "_cell_guid": "79c7e3d0-c299-4dcb-8224-4455121ee9b0",
    "_uuid": "d629ff2d2480ee46fbb7e2d37f6b5fab8052498a",
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 206
    },
    "id": "EYoJW3XGYHNw",
  "outputId": "09d9b33f-ece4-40f7-9dc1-2a5c0516f3ff"
 },
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)\n",
     "0           1    Male   19                  15                      39\n",
     "1           2    Male   21                  15                      81\n",
     "2           3  Female   20                  16                       6\n",
     "3           4  Female   23                  16                      77\n",
     "4           5  Female   31                  17                      40"
    ],
    "text/html": [
     "\n",
     "  <div id=\"df-fb30e0d0-3212-4a95-b578-c0f82a0276e9\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>CustomerID</th>\n",
     "      <th>Gender</th>\n",
     "      <th>Age</th>\n",
     "      <th>Annual Income (k$)</th>\n",
     "      <th>Spending Score (1-100)</th>\n",
     "    </tr>\n",
     "  </thead>\n",
     "  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1</td>\n",
"      <td>Male</td>\n",
"      <td>19</td>\n",
"      <td>15</td>\n",
"      <td>39</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>2</td>\n",
"      <td>Male</td>\n",
"      <td>21</td>\n",
"      <td>15</td>\n",
"      <td>81</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>3</td>\n",
"      <td>Female</td>\n",
"      <td>20</td>\n",
"      <td>16</td>\n",
"      <td>6</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>4</td>\n",
"      <td>Female</td>\n",
"      <td>23</td>\n",
"      <td>16</td>\n",
"      <td>77</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>5</td>\n",
"      <td>Female</td>\n",
"      <td>31</td>\n",
"      <td>17</td>\n",
"      <td>40</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
       "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-fb30e0d0-3212-4a95-b578-c0f82a0276e9')\"\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-fb30e0d0-3212-4a95-b578-c0f82a0276e9
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-fb30e0d0-3212-4a95-b578-c0f82a0276e9');\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": 2
    }
   ],
   "source": [
    "df = pd.read_csv('Mall_Customers.csv')\n",
    "df.head()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {
    "_uuid": "067b51b55e3daed31cfed664a1eacde23b932ec1",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "8nQvoDDrYHNx",
    "outputId": "0d6705da-2c0a-4ccf-dc9b-220e80bba1d1"
   },
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "(200, 5)"
      ]
     },
     "metadata": {},
     "execution_count": 3
    }
   ],
   "source": [
    "df.shape"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {
    "_uuid": "0e352ff53fe883942afd06b2e830ff7fa7cfdabd",
    "colab": {
     "base_uri": "https://localhost:8080/",
   "height": 300
  },
  "id": "MvBaa4f4YHNx",
  "outputId": "a1ba84f8-4ce8-4629-8c06-77b953211b04"
 },
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)\n",
     "count  200.000000  200.000000          200.000000              200.000000\n",
     "mean   100.500000   38.850000           60.560000               50.200000\n",
     "std     57.879185   13.969007           26.264721               25.823522\n",
     "min      1.000000   18.000000           15.000000                1.000000\n",
     "25%     50.750000   28.750000           41.500000               34.750000\n",
     "50%    100.500000   36.000000           61.500000               50.000000\n",
     "75%    150.250000   49.000000           78.000000               73.000000\n",
     "max    200.000000   70.000000          137.000000               99.000000"
    ],
    "text/html": [
     "\n",
     "  <div id=\"df-6b8817df-b38c-45f6-9e3b-7aff85953ce1\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>CustomerID</th>\n",
     "      <th>Age</th>\n",
"      <th>Annual Income (k$)</th>\n",
"      <th>Spending Score (1-100)</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>count</th>\n",
"      <td>200.000000</td>\n",
"      <td>200.000000</td>\n",
"      <td>200.000000</td>\n",
"      <td>200.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>mean</th>\n",
"      <td>100.500000</td>\n",
"      <td>38.850000</td>\n",
"      <td>60.560000</td>\n",
"      <td>50.200000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>std</th>\n",
"      <td>57.879185</td>\n",
"      <td>13.969007</td>\n",
"      <td>26.264721</td>\n",
"      <td>25.823522</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>min</th>\n",
"      <td>1.000000</td>\n",
"      <td>18.000000</td>\n",
"      <td>15.000000</td>\n",
"      <td>1.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25%</th>\n",
"      <td>50.750000</td>\n",
"      <td>28.750000</td>\n",
"      <td>41.500000</td>\n",
"      <td>34.750000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50%</th>\n",
"      <td>100.500000</td>\n",
"      <td>36.000000</td>\n",
          "      <td>61.500000</td>\n",
          "      <td>50.000000</td>\n",
          "    </tr>\n",
          "    <tr>\n",
          "      <th>75%</th>\n",
          "      <td>150.250000</td>\n",
          "      <td>49.000000</td>\n",
          "      <td>78.000000</td>\n",
          "      <td>73.000000</td>\n",
          "    </tr>\n",
          "    <tr>\n",
          "      <th>max</th>\n",
          "      <td>200.000000</td>\n",
          "      <td>70.000000</td>\n",
          "      <td>137.000000</td>\n",
          "      <td>99.000000</td>\n",
          "    </tr>\n",
          "  </tbody>\n",
          "</table>\n",
          "</div>\n",
          "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-6b8817df-b38c-45f6-9e3b-7aff85953ce1')\"\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-6b8817df-b38c-45f6-9e3b-7aff85953ce1
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-6b8817df-b38c-45f6-9e3b-7aff85953ce1');\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
        }
      ],
      "source": [
        "df.describe()"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 5,
      "metadata": {
        "_uuid": "57e39b6ed035b3e1b9e3dda9efd273fe552f48a7",
        "id": "XxUcixvPYHNy",
        "outputId": "8ea23b8f-1f95-425a-fb77-88725f9677fb",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
```

```
      "text/plain": [
       "CustomerID                 int64\n",
       "Gender                    object\n",
       "Age                        int64\n",
       "Annual Income (k$)         int64\n",
       "Spending Score (1-100)     int64\n",
       "dtype: object"
      ]
     },
     "metadata": {},
     "execution_count": 5
    }
   ],
   "source": [
    "df.dtypes"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "metadata": {
    "_uuid": "e57f14ca0de9ce6d08a1dd1fb6470707db3eb85a",
    "id": "pFle9JtcYHNy",
    "outputId": "695d4889-c01e-4e27-9857-6c923dc6b3a6",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "CustomerID                0\n",
       "Gender                    0\n",
       "Age                       0\n",
       "Annual Income (k$)        0\n",
       "Spending Score (1-100)    0\n",
       "dtype: int64"
      ]
     },
     "metadata": {},
     "execution_count": 6
    }
```

```
      ],
      "source": [
        "df.isnull().sum()"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "_uuid": "2459a03e13d3bae3d946715b17c151f0615b3dcf",
        "id": "p4Ay6QYBYHNy"
      },
      "source": [
        "## Data Visualization"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 7,
      "metadata": {
        "_kg_hide-input": true,
        "_uuid": "25f06dcc79fbf5f5b83d3dc7a337865aacb7657e",
        "id": "Gz08NWqwYHNz"
      },
      "outputs": [],
      "source": [
        "plt.style.use('fivethirtyeight')"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "_uuid": "0ee494fa3103f3aead92791fd958ac8e97b7beed",
        "id": "wDIZR2izYHNz"
      },
      "source": [
        "### Histograms"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 8,
      "metadata": {
        "_kg_hide-input": true,
        "_uuid": "7e72632388acada5fd66e2323f0544af06cac54d",
    "id": "IgKCSmswYHNz",
    "outputId": "a45ed8b9-3cb1-4a26-ffa4-2882ec8b4d13",
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 389
    }
   },
   "outputs": [
    {
     "output_type": "display_data",
     "data": {
      "text/plain": [
       "<Figure size 1080x432 with 3 Axes>"
      ],
     },
     "metadata": {}
    }
   ],
   "source": [
    "plt.figure(1 , figsize = (15 , 6))\n",
    "n = 0 \n",
    "for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:\n",
    "    n += 1\n",
    "    plt.subplot(1 , 3 , n)\n",
    "    plt.subplots_adjust(hspace =0.5 , wspace = 0.5)\n",
    "    sns.distplot(df[x] , bins = 20)\n",
    "    plt.title('Distplot of {}'.format(x))\n",
    "plt.show()"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "_uuid": "f3b491e34170ac1ee8e670bf1292fb8e451d5053",
    "id": "_BQ_WdaNYHN0"
   },
   "source": [
    "### Count Plot of Gender"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 9,
   "metadata": {
        "_kg_hide-input": true,
        "_uuid": "befa7a57e4e37f96c8644b24e6982f354e964864",
        "id": "iBZBC2tEYHN0",
        "outputId": "14359a02-7e45-4c56-e2e2-681c89a8dd34",
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 333
        }
      },
      "outputs": [
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 1080x360 with 1 Axes>"
            ],
            "image/png": 
"iVBORw0KGgoAAAANSUhEUgAABAYAAAFZCAYAAAALui1cAAAABHNCSVQICAgIfAhkiAAA
AAlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3de7SXc97/8dfW
QRFtU9q5VUihmhgiOUc3OYQcWjmTIcK6xRjHGUxyyIxDczfcIG6ZGoScFwbDyKImGjfdjjEYZtU
usaIoqPbvD7/Z99pTDIt799W+Ho+19Ir6XNe+9vvb+qxWPV3f61tWVVVVHQAAAKCQ1ij1AAAAAEDpC
AEDpCAMAAABQYMIAAAAAFJgwAAAAAAAUmDAAAAAECBCQMAAABQYMIAAAAAAFJgwAAA
AAAUmDEAjMWvWrFKPQIHZf5SS/Ucp2X+Uir1HfRIGAAAAoMCEAQAAACgwYQAAAAAKTBg
AAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwY
QAAAAAKrKyqqqq61EM0Nj2um17qEQAAoMG9fErvUo9QWLNmzUrXrI1LPQaNhDsGAAAAoM
CEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoM
CEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAA
hMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAA
KTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAA
CgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAA
oMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAA
CAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAA
AAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQ
AAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEgX/Rs2fP/Od//mepwAAAIBVYrUPA8OGD
Ut5eXXOPtEFKS8vz+1+205Y5ddNFFKS8vz+DBg0swGQAAAHz/rfZhEk6dOiQ+6e65J4e65J4sWLapZW7W7JkbS677
bZ06NChhJMBAADA91ujCAPdu3dPhw4dMnXq1Pq1qVOnZuONN85WW21Vv7ZkyZKceuqpadu2bZo1a5Ydd9wxzzzzTJ1
rb6wbbGzJmyzwwFfJmjJXB1e/DQ+6/rc06u4XrS8Ndyv/XPH+W0w5+Vs6WJbfZpyyrUQOvt2Hz
Y8eO2XvvvTNTN9+vSvbfbHH3c008/PV26dEuIw2y77R7xmEzviAX6oUXwvXD1wIAAA0mDZuydqHtRn
Z9KkSTW/njnxxRfu3RAEOmbzz9kDhBHfUffRDAwakEFjaamCpdsRFS7R/WzHsOAQEAZWyyomFNW3MQ3
jXU+Fwlm7Z+gQ1Ib1gbjmbq7bNjwhhAqCgqYmGjcidQdsv85dosQ2YTvXG5vnVXOGjJHPjcZz5mhgaFAzq9NmdYmxUWjr9Xx+6gQ1nCa6NrDS/iwwjN5NPomV8m0L4mGXx6IrLxaQmXmrzpbcDA3GyQeF9thdsvbJVZDM/i6R2vvNOD6h9qoVZOMI+PkKC52OK3/pRa2rqXixHbeZRfiLkj9jYBHrxwiMmLYn/zB7+EmjavV
cemImTp1aubNm5eIs5fms88+y/vvv/C7744o59NNPO6VLI1rrixcvzttvv91grwUAAABWIUYT
BpLkqOOrBhw7L22mvn/PPP3+FXX4tX74zvrxFx22WWPx1v9ZcuW3sj7x2w2WXp1KIT1IxzzzRxwwAH5/PPPPP
pV27dnnooYeZWO7bOOuvvt+Cwqj+Ww28LLDbbruWb7NmT9/fvbbb7/+Pe54TJ3/PyP9V6k07/OnOO54oor0r9//yTJ3
```

LIzU1IZ+ZXX22qrrTJ37tysscYa2XjjjRtqbAAAACiZRhUGysrK8swzz6S6ujprrrnmcsc33XTTTJ48
Odtuu20+/fTTXHjhhWnevPIXXq9v377p06dPjjjiiPziF79I165dM3fu3Dz22GPp27dvdtxxx4Z8OQA
AANDgGs3DB/9pnXXWybrrrrvCY2PGjMmiRYvSt2/fHH/88TnqqKPSqVOnr7xWWVIZJk+enF122
SWnn356tttuuwwZMiRvvvImNthgg4Z6CQAAALDKIFVVVVWXeojGpsd100s9AgAANLiXT+Id6h
EKa9asWenatWupx6CRaHR3DAAAAADfnjAAAAAABSYMAAAAQIEJAwAAAFBgwgAAAAAUm
DAAAAAABSYMAAAAQIEJAwAAAFBgwgAAAAAUmDAAAAAABSYMAAAAQIEJAwAAAFBgw
gAAAAAUmDAAAAAABSYMAAAAQIEJAwAAAFBgwgAAAAAUmDAAAAAABSYMAAAAQIEJA
wAAAFBgwgAAAAAUmDAAAAAABSYMAAAAQIEJAwAAAFBgwgAAAAAUmDAAAAAABSYM
AAAAQIEJAwAAAFBgwgAAAAAUmDAAAAAABSYMAAAAQIEJAwAAAFBgwgAAAAAUmDAA
AAAABSYMAAAAQIEJAwAAAFBgwgAAAAAU2LcOA4sXL86tt96aGTNmNOQ8AAAAwCr0rcN
AixYtcvrpp2fmzJkNOQ8AAACwCtXprQRdunRJZWVIQ80CAAAArGJ1CgM//eIPM3bs2Lz88ssN
NQ8AAACwCjWty8lPP/102rZtm1133TW9e/fOJptskpYLYtW9Y6p6ysLL/61a/qdUgAAACgYZRVVV
VVf9uT11tvvW++YFIZPvzww5UaanXX47rppR4BAAAa3Mun9C71CIU1a9asdO3atdRj0EjU6Y6B
jz76qKHmAAAAAEqgTs8YAAAAABqXOt0x8E9PPvlkPPkIkpk6dmnnnz5uW0007LZpttIoULF+bFF19Mjx
49UI5eXt9zAgAAAA2gTncMfPbZZZznkkENy8MEH55rrsnEiRMze/bsJEnz5s1z7LLHH5oYbbmiQ
QQEAAID6V6cwcMkII+Tpp5/OjTfemJkzZ6a6+v+eW9i8efMMHDgwDz/8cL0PCQAAADSMOoW
Be+65JyeccElOPfTQ5T6mEm6du2ad955p75mAwAAABpYncLA/Pnzs/nmm3/I8bKysixevIhw
IAAABWjTqFgQ4dOuT111//yuPPP/98OnfuvNJDAQAAAKtGnT6VYNCgQRkzZkwGDBhQc+dA
WVIZkuSmm27KPffckxEjRtT/IKuZI0/pXeoRKKBZs2a0a1a9eupR6DgrL/KCX7j1Ky/4DGoE5h4Mwzz
8yMGTMyYMCAdOnSJWVIZTn33HPz4YcfprKyMnvvvvXdOOeWUhpoVAAAAqGd1CgPNmzfPH
XfckTvuuuCP33HNPysrKsmTJkmTJkmy11VY56KCDMnjw4Jo7CAAAAIDvvzqFgX8aNdN0aVN+z
AAAAAKtYnR4+CAAAADQuX3vHwP7771/nC5aVleCyFly5Yt98yAv//9
73nnnXfSunXrbLTRRkmSd999Nx9//HE22WSTbLjhhg03LQAAAFCvvjYMPPjgg7V+/dxz
ImDFjcthhh6VJkyZJkqVLI+Z3v/tdLrzwwIx33XUNNy0AAABQr+r0jIjGf//znOeqoo3LkkUfWRIEkad
KkSY4++ugceeSRueCCC+p9SAAAAAKBh1CkMvPzzyy+nYeNXHu/u/UqVNeeWVIR4KAAAAWD
XqFAbat2+fu+++O0gu/WLFnu2JIISzJIypS0b9++++3oYDAAAAGtbXPmgX51++uk544ZCUOYP5ZNZ9
hj07Iz5yTJW2+9lQkTJmTmzJm56qqrGmRQAAAAoP7VKQwA7cwcd9xx9xWWONNTJy5MiceeaaZNZ9Y
UF1dnbZt2+aaa67Jscce2yCDAgAAAPWvTmEgSY8455pgcccdcHOmYL9xYThAAAPWvTmEgSY455pgcccQReeGFF/Lee+8IST
W+XIAAABACX2nf8k3bdo02223Xbbbbrv6ngcAAABYhb5TGHjjttdfyzjvvpKqqKtXV1csdP/zww1d
6MAAAAKDh1SkMvP322xk6d6dGhmzJixwiCQJGVIZcIAAAArCbqFAaGDx+eV155JdffnI22GH
lJeXN9RcAAAAwCpQpzAwbdbqJMaah4AAABgFVqjLie3adMmVIq9MsSXbaaacfvjh2XDDDdOkOSZPIAgAAAK
xidQoDxx9/9/fCZPnpylS5c/c21DwAAADAKlSntxJxJsvPJPPHG43sOHzaZPIAAA8dQpCOJxxwV5/9hzMmXxvvNPXbfuONjyqgqOCUgAA
PqbUAAAACg4dQpDJxwwgk1/33xx3xRev8JyysjJhAAAAAFYTdQoD999999b9f0PNAQAAAAJRAncLAzjv
v3FBzAAAAACVQpzDwT5988OP5988kImzJiRefPmpW/fvmnrwRefPmpW/fvmnXrI19zwUAAACsAnX6V
PDBOfnkk/Pqq68Pn58NNtgm48ePr/chAQAAgIZRpzAwbfz4jBw5qbjB4zuV1NE9f/kmJ+FgYZ9V+ThR72O8o0i6Y7Uif1S2/D
T5C0Pl+x7Xeh7Tf33wzSZqMv5t9Zdc0BTk+5y4JBd0tX7+q4kv5M03vFwvp2Rpx49bqcVfpR/s7m
2rRpk3333Tf33HNPvQ8JAAAANIW6hYEEbbrghAwcOzOjRo7Prrrrsud3zLLbbfMG2+8UW/DAQAA
A2rTs8YeOeddzJs2LCvPF5eXp6ePPvopopYda3S044/BSj0ABtU+yoNRDUFj2H6Vk/1FK9h+IYu+V
1jrX3FrqEepVne4YKC8vz7x7x7x5877y+KuKioqVHgoAAAABYNeoUBvbaaafbcba69MmDBhhXcF/O///m
9uueWW7LvvvvvU2HAAAANCwyqqqqqq/+bQvVVZWpl+/flmyZEn22muvjqTJw4MYceemiWLFmS4oX9+Xp9XAIKqqqpzRMy4H/nQKcNCjnMyw/K/MatzzEneppVWpT8XrS
Bx98MP/2b/+Wxx9/PD/4wQ8acubvPW8IAAAaLwK/VaCioqKPPNkk+nfv3/uv//+VFdX54477shj
z2WQYMG5dFHHHy18FAAAAIDVSZ3uGPhXH3zwQQZ2bdtmjTXq1bgdamNCgaNXcMAAAANF6N7
Y6BOn0Oqwb9q27tfc0BAAAAIMC3CgOPP/541I577fTp0ydJsmjRopx99uytndLnLndexY8ece+659TshA
AAA0GC+MQxnMo1gwYNyi233Gk9o9//CO/+93v//dxJh4dvg8mdm3zkzz0/+2p0MsPzTcOnSmdKzizHG
mBQAAAOrVNz4YYNbYSKbSenevXsGD3bbrst77//9fs3XFItskUmTJmTWJmvIIkIAAEAD9+8Yw8Pzzz

2efffb5Vhfbf//9M23atJUeCgAAAFg1vjEMzJkzJxtttFGttebNm+fggw9ORUVFrfUNN9wwc+bMqd8
JAQAAgAbzjc8YaNasWf7xj3/UWmvVqlVuuumm5c794osvaj1zAAAAAPh++8Y7Bjp16pQZM2Z8
q4vNmDEjnTp1WumhAAAAgFXjG8NA//79c/fdd+ett9762vNmzZqVKVOmZO+996634QAAAICG
9Y1h4NRTT02rVq0yYMCA3HvvvVm6dGmt40uXLs2UKVNywAEHZJ111smpp57aYMMCAAAA9
esbnzHQpk2bTJ48OUceeWSGDBmSli1bpkuXLlI77bWzcOHCvPXWW/nss89SUVGR22+/PW3a
tFkVcwMAAAD14BvDQJJsvfXWee655zJ+/Pg88sgjef3117NgwYK0atUqP/zhD7P33ntnyJAhKS8
vb+h5AQAAgHpUVlVVVV3qlRqbBWccXuoRAAAAaCDrXHNrqUeoV9/4jAEAAACg8RIGAAAAo
MCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAAC
AAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAA
AAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQA
AACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAA
AAoMCEAQAAACgwYQAAAAAKTBgAAACAAhMGAAAAoMCEgX9x+eWXZ4cddij1GAAAAALB
KIDQMDBs2LOXI5ct9vfTSS6UcCwAAAAqjaakH6Nu3b2644YZaa23atCnRNAAAAFAsJX8rwZpr
rpmKiopaX02bNs1DDz2U3XbbLRUVFdIyyy1zySWX5PPPP6/5vp49e2bUqFEZNmxYOnTokB49
emTKICmpqqrK8ccfnw033DDbbLNN/vCHP9R8z9KIS3PaaadIyy23TPv27bPNNttk9OjRWbZs2df
OOHHixGy//fapqKhIr1698pvf/OYbvwcAAABWByUPAyvy+OOPZ+jQoTnxxBPz/PPPPZ8yYMbn33
nszYsSIWuddf/316dWrV/74xz9m4MCBGTZsWE488cTsueeemTp1ancccMHTo0ixcvTpIsW7Y
sG2ywQW6++eZMmzYtP//5z3PVVdI4sSJXznLhAkTcsklI+T888/PtGnTMnLkyIwePTrjxo1r0N8D
AAAAWBXKqqqqkv1w4cNG5bJkyenRYsWNWs77LBDFi1aIN133z1nn312zfoDDDzyQk046Ke+//
37KysrSs2fP9O7dO7zfddFOSZOOChenQoUOGDh2aK6+8Mkny7rvvvvZVZuttsoTTzyRbfeeeoUzXHz
xxXnhhRdy7733Jvny4YP33XdfnnvuuSTJD3/4w/zsZz/LYYcdVdVvM91113XSZMmJBp06at8JoLzjh
8JX5XAAAA+D5b55pbSz1CvSr5MwZz23HHHjB49uubXLVq2f//OUWtNcsbXLVVq0yLbbbbpu//OUvtdaWzz77LJW
VIWnfvn2SpEePHPHjXHW7VqlbXWWqvWWrt27ZIk8+bNq1kbP358brrlpkyevrlz33ZvHhxvvjii3Ts2HGF
s33wwQd5//QnP/IJzfqSJUtSXV2yngIAAAD1puRhYK211+krnzm/t0vrrS1btiztiznpwnHHOBg4cu
Nz5bdu2rfnvZs2a1WVIaWpk2b1vr1P6+XJFOmTMl5552Xsy65JL179866666b66+/PjlrzZvH79XPH5oEHHlj
hbP/8vquvvjo77LBDrt+G3PxU2j/xTHSXAgmSIzKiiNgQhxVH9kPfBi2jr6kTQBwPUk2Et25xs8tT6N1FpfW5WHrGqswAmEwCr2nQNwtqzhh6tx8jNvCdfpGPsxU8WgX7X36u/XfNvFhLpxtp1m3d1QPXSA8N3wfyb8qRk5Q0PSvWHpxe3NeF0p/XQUg5Zw/AMqAs/OOvBnJe0N04/c7jdbpsjw1ep/mNxzuj7TSQ4vQ7/t7jiEOezPQ3aKsl0GRZ/DsYB/q2Ub+oACrDhPzJ9SCnIe0nCvLgFH7B5mR/wWnVtH3hyEwX4pHlCtWXjcWbOA5UnnomgiPWNpnzeyEw/sRvSv95ZgMNvcA3w4kVdmk0+0gDyeWRi6ipEq7/M58rNUGzRQ5vZnMmtS1qAAAAIAQQr2vvDtJ+lSvWTkPDzLyuzBqdJDJ9FxDIJ3Q1V5DWAQQT+Y2gERr96lKEWZUeKkfTXDczbx+ZwQ2ZbZeLkK1cOWbb/wP8hbT6gyYlMwkIPxMdbNDykRBC2bDn1SaZbqU3YnRjN/KOycYJAlq1y8o+aMtuBhmxgUhfUPKzG8hfMcvYwt4zxMdK3pHCsLvTQlWPJeDDRCoQNEbdq8lKDP58p1Uu4gNbzWpMg5LczBsJhI4fcFuV73uRDBWGCNMmhPlcD4qbAzJJDh3bxRq0VRcPJ82L+8oyL+M3pJ4q4sFtO+Rw8POpLnCR/Dba+3NWtqFpNdfmKm/xCBR1r58p0xgxG0sTOsHzIoNwXezFxE7fdk/aEqljRHCLi9w0sWqHdhwbA1ATdm6tXtRBqPr7c+dcagg0Ad4AbdwfRnwvdCI2XaUfTNFR8ziuyZd6YL3PPv7uNFo9nfUzTJMYPrbmc7NdCHb6tsJSgLmyqyjUcbP1RGgMGFFd9qjSjhe9uyIaCG4BfEOWKdYvzBPJMbPlcsrKm1bPKvvd99RC9TF

1/NvstaqqqgwdOjSdOnVKp06dMnTo0FRVVa3Kl8FqSBioB1OmTMm5556bn/zkJ3nqqafSu3fvD
Bo0KO+9916pR6MRefrpp/PjH/84jzzySO677740bdo0AwcOzEcffVRzzujRo/Ob3/wmo0aNyh/+8I
esv/76Oeigg7JgwYISTk5j86c//Sk333xzevToUWvd/qOhVFVVpX///qmurs7kyZMzbdq0XHnlIVI//fV
rzrH/aCjXXnttxo0bI1GjRmX69Om54oorMnbs2Fx99dU159h/1JdFixale/fuueKK9KyZcvljn+bvXb
CCSfkpZdeyp133k777wzL730Uk466aRV+TJYDZVVVVVVl3qI1V2/fv3So0eP/PrXv65Z22abbX
LggQfmoosuKuFkNGYLFy5Mp06dMmnSpOyzzzz6prq7OFltskRNPPDFnnXVWkuSzzz5L165dc8
kIl2TIkCElnpjG4OOPP85uu+2WX//61xX/61xk1alS6d++eX/7yl/YfDWrEiBF55pln8sgjj6zwuP1HQxo8e
HDWW2+9/Nd//VfN2sknn5yPPvvot99+u/1Hg9lwww1z5Z7ZVX5sgjj0zy7f6se/3117P99tvn4YcfTp8+
fZIkzz33XPbZZ5/86U9/SteUv2evh+c8fASvr888/zP//zP9Ijjz1qre+xxx6ZNm1aiaaiCBYuXJhly5
alvLw8SfLuu++msrKy1l5s2bJldR7uRejN8+PAceOCB2XXXXXWut2380pAcffDC9evXKkCFD0q
VLI+y888658cYbU1395f/bsP9oSH369MnTTz+dN954l0ny2muvZerUqzUqdlzzz2T2H+sOt9mr02fPj2t
WrXK9ttvX3NOnz59svbaa9uPfK2mpR5gdTd//vwsXbuu2MSbL++utn7ty5JZqKljj33HPTs2fP9
O7dO0lSWVmZJCvci7/Nnz17IISWVmZJCvci7Nnz17I89H4TJgwlX/9619z4403LnfM/qMhvfPOO7npppZPjw4Zk5c
2bOOeecJMnQoUPtPxrU8OHDs3Dhwmy//fZp0qRJlixZkrPOOisnnHBCEn/+sep8m702d+7ctGn
TJmVIZTXHy8rK0K0rZtW/824WsJA7AaOv/88/P888/n4YcfTpMmTUo9DgUwa9asjBgxBgxIg8//HCaNW
tW6nEomGXLImXrrbeueXveVItb/+9a8ZN25chg4dWuLpmyyRmTN
n5txzz02nTp1yzDHHlHo165diaaiMTvvvPNy11135b7
77svGG29cs15RUZEk9iINYvr06Zk/f3769OmTNm3apE2bNnnmmWdWcybty4tGnTJj/4wQ+S2H80jI
qKimy++ea11jbbbLO8//77NccT+a0007LIYcckh49euSwww7LqaeemmuuuSaJ/ceq8
232Wrt27TJ//vyat1olXz6b4lMPPrAf+VrCwEpq3rx5fvSjH+WJJ56otf7EE0/Uem8P1IdzzzjmnJgpsttI
mtY5ttNFGqaioqLUXFy9enOeee85eZKXtt99+efbZZZN16tSar6233jqHHHJIpk6dmi5duth/NJg+ffr
kzTffrLX25ptTT5e7O69JkyZZtmxZEvuPVefb7LXevXt4cKFmT59es0506d
Pz6JFi+xHvIaTc8899+JSD7G6W2eddXL55Zenffv2adGiRX75y5PX
Z+W2227LzTffnA4dOmTRokVZtGhRki8DVVlZWZYuXZZoLLrgglZWVufbaa7
PmmmuW+BWwOmvRokXWX3/9Wl933HFHOnXqlCOPPNL+o0F16NAho0aNyhprrJH27dvnj3/8P
Y0aOHJkzzjgjvXr1sv9oUK+//npuv/32dOnSJc2aNcvUqVNzySWX5OCDD06/fv3sP+rVwoUL89pr
r6WysjK//e1v071796y33nqqqqhsr5/PPP07p162/ca23bts2f//zn3HnnnnenZs2f+/ve/54wzzssg222zjIwv5Wj6
usJ6MGzcuo0ePTmVlZbp165bLLrssO+20U6nZp165bLLrssO+20U6nHohH556cP/Ktzzjkn5513XpIvbxW74oorcvPNN6e
qqiq9evXXKr371q3Tv3n1VjkpB7LfffjUfV5jYfzSsRx55JCNGjMibb76ZDh065PMQTT8xJJ51U84At+4
+GsmDBgIx66aV54IEH8sEHH6SioiKHHHJIzj777LRo0SKJ/Uf9eekll/l1g8//PBcf/3132qvV
VVV5eyzz85DDz2UJNInn31y5ZVXXfuXfJSERBgAAAKDQPGMAAAAACkwYAAAAgAITBgAAAK
DAhAEAAAAoMGEAAAAACkwYAAAAgAITBgAAAKDAhAEAoFGbPXt2Lr/88rz00kulHgUAvpeE
AQCgUZszZ05GJkzJ8OHD0/37t3TrI279OzZM//xH
/+RBQsWJEnefffdDBkyJJtssknat2f3XffPQ888ECta0yaNCnI5eV59913a61PnTo15eXImTp1as3
afvvtI+222y6vvfZa9t9//2ywQbp1q1bRo8eXev7dt999yTJqaeemvLy8pSXI+fyyy9vqN8GAFjtNC3
1AADA6q+ysjL9+vXL/Pnzc+yxx6Zbt2tZMkmzZtMnny
5Bx99NEZO3ZsDj300O/0cz/55JMceuihHIh77703F110Ubp3754999wzm2++ec4///MdS/n
MX4mGQAYOHJh77703F110Ubp3754999wzm2++ec4///x
cdtllOe6447LDDjskSXr06FGfLx8AVmvCAACw0i6++++O/+LMnj07v//977/RPttvWrJ933nmprq7OBRdck
DIz5uT+++/PLrvskiQZMmRI+vbtmwsuuCAHHnhgmjVrVuefW/UtFIZmeuvvz6
HH354kuToo49O49Oz54989vf/jZ77rIn2rVrlz333DOXXXZZttuuwwePLh+XjAANCLeSgAArJRly5bIw
QcfzJ577IkrCvxTWVIZfv/732errbaqiQJJ0rJy/z4xz9/0s1u2bFnrH/vNnrH/vNmzfPNttk3fe
eec7XQ8AikgYAAB
BWygcffJBFixZsyYMGHCcuuH+akfWR9H9Au05BJf0H7/8peoHx/ANLlnPKzbHQcO+x/6EMRBPRQcIRLG+Td5MPKNcAjTT8ocr7i/VYegb/mtUGQlcZOkDSuwX7F4q1bKZGfjG8LX1zYnwTQ5KJ8bdDNF4XVbxqTBB3Uq82ofoZNUMhR3IBHv95APmXVbsjRHnmGQoaDgT2mZYkrjvB9MXx4n/fWeoFvBp10n7/3BPSBiN1vfn34kIZmmRnfqPPe+sDBHKBeJ9V2Wne1V7gmxyZ4sS+2nMPnMGUtVD6JIHxPrbaM+7WOynXl09zZXtsJvPKsfGRR8kLAT7Oe5UIwqdlRlxjAeRJDtRNd/wFUhB/WKbV8HjX/h/VCuHtHqIq2XOVWjhVf1Ou2Ua5/1MTWTYxDMhTPp6x4Q9cAWfhtpXtRBdIl+4/KP1FXl1rkcv8d2AhB8BWOs2UK37mb8TpLSaWQ6e8XEmwVUX5BQjKn7hZTd0bBj6I1AbLnTW33jIKUUBcwH0p6yO4m1uUM1O1fG3AvnFHAJCqAPpIZtlmR6sVyZPxNW9UHfgIJHDxCaFWaShUBpXRBDO5P9KBiEJxfXzLjWz+W2Cc3yeJ95gn2Yl6aOMw15XP5TCcuHzXYyCZ+JSeGB+V1Xz9YtfCLq8T77n1WfpJ/IxZBmhkdIMnm1fkYuNgZibNkapjzY6LnkdF6mk59eVqNLu1aW9ccJA3gQ59JvaGiZDv7RXvoj2FRDd10Z8WwyVhFo5jQBGaRRJT7Do1TLPcl3TVw1ezOgPfkm6BR/aBtD8a67r+FLyAkHFEoPYVh4N0P9p7AA+P8i+H0TsC18LHvXOdqU2XMJnbmDzrHwGkoHXk5SFBO3g1W6qVGBw1cLYCmQNGrRoV3SmyIK7xAeScHv0NYAF/IkMm4fcBOXV3WOz32qRY9mAEfwvzI6Cbv7CeqP2eZ8tvNENK1gW41V8FjD4XcLi+4Ccp9TIBBtgIvQo7jk5CbOEIO7wgTMJx7JX0xINHb2nKcTfa/L8cMmVVFQFtVjPF0SFYwF4njnaxkrjQPpNgLZqPcPj3POVGGmc2RENHa4dJuBrlXYHDNNcq40brGxUzkvfGO5bmoSS4eXb+X6yfTw1ZW9FTyWcXNTsbFaAr+zuafbbPM5LCYiqOJnKK+NWnS1GvwlWLeLkFYvV0srnWYYkD6w7Zp2SwNSKm/gd8J4QrhwQIQLB52TC6L88TGaDqIgrw0XcXhA

```
EAYKWsscYa2W+//fLoo4/mz3/+83LHq6ur079//7z44ot59tlna9YXL16c8ePHp6KiIj/60Y+SJJtssk
mS1Dpv6dKImTBhwneeb6211koSby8AgK/gUwkAgJV20UUX5cknn8yAAQNy3HHHZYsttsjcuXN
z//33Z+LEiRk+fHjuuuuuDB48uNbHFb722msZO3Zsmjb98q8k3bp1y3bbbZcRI0bko48+ynrrrZcp
U6ZkyZII33m2TTbZJOXI5Rk/fnxatWqVVq1apVu3bunevXt9vXwAWK0JAwDASmvfvn0ee+yxXHr
ppbnrrrvy8ccfp3379tljz3Spk2btGrVKg8//HAuvvjijBs3Lp999Im6deuWW265Jfvvv3+ta40dOzbDh
w/Ptddem9atW+foo4/OLrvskoEDB36n2Zo1a5Ybbrghv/jFL3LWWWfliy++yDnnnCMMAMD/V1ZV
VeXpPAAAAFBQnjEAAAAABSYMAAAAQIEJAwAAAFBgwgAAAAAUmDAAAAAABSYMAAAAA
QIEJAwAAAFBgwgAAAAAUmDAAAAAABSYMAAAAQIH9P4iqbRkEYxsYAAAAAElFTkSuQmC
C\n"
     },
     "metadata": {}
    }
   ],
   "source": [
    "plt.figure(1 , figsize = (15 , 5))\n",
    "sns.countplot(y = 'Gender' , data = df)\n",
    "plt.show()"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "_uuid": "fcc79189184015bc626be9264fbd2836b2ebad8c",
    "id": "cl-esshkYHN0"
   },
   "source": [
    "### Ploting the Relation between Age , Annual Income and Spending Score"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 10,
   "metadata": {
    "_kg_hide-input": true,
    "_uuid": "89e99c64c4e69e1141090185da444c8991e5ce8f",
    "id": "hN1mxD3nYHN1",
    "outputId": "dd97d371-5d69-4306-a63a-22badef33bb0",
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 465
    }
   },
   "outputs": [
    {
     "output_type": "display_data",
     "data": {
      "text/plain": [
       "<Figure size 1080x504 with 9 Axes>"
      ],
     },
     "metadata": {}
    }
   ],
   "source": [
    "plt.figure(1 , figsize = (15 , 7))\n",
    "n = 0 \n",
    "for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:\n",
    "    for y in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:\n",
    "        n += 1\n",
    "        plt.subplot(3 , 3 , n)\n",
    "        plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)\n",
```