```
{
  "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": 1,
      "metadata": {
        "id": "drX6Hcn_I8aK",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "b0bbad4b-dec7-4a54-ce5a-5ecfb01f2cf9"
      },
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/\n",
            "Requirement already satisfied: nltk in
/usr/local/lib/python3.7/dist-packages (3.7)\n",
            "Requirement already satisfied: joblib in
/usr/local/lib/python3.7/dist-packages (from nltk) (1.2.0)\n",
            "Requirement already satisfied: click in
/usr/local/lib/python3.7/dist-packages (from nltk) (7.1.2)\n",
            "Requirement already satisfied: tqdm in
/usr/local/lib/python3.7/dist-packages (from nltk) (4.64.1)\n",
            "Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.7/dist-packages (from nltk) (2022.6.2)\n"
          ]
        }
      ],
      "source": [
        "!pip install nltk\n",
        "\n",
        "import numpy as np\n",
        "import pandas as pd\n",
        "import matplotlib.pyplot as plt\n",
        "import seaborn as sns"
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df = pd.read_csv('/content/spam.csv', encoding='latin-1')\n",
        "df.sample(5)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "S2oSme6QW14t",
        "outputId": "9ff2bb99-c409-4fc4-b69d-2d92bcc4a63b"
      },
      "execution_count": 2,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "       v1
v2 Unnamed: 2  \\\n",
              "4447  ham                          I sent them. Do you
like?        NaN   \n",
              "1774  ham  Its ok chikku, and its my 1 of favourite
song....        NaN   \n",
              "4151  ham                             What's nannys
address?        NaN   \n",
              "4180  ham          Ok da, i already planned. I wil pick
you.        NaN   \n",
              "926   ham  But I'm on a diet. And I ate 1 too many
slices...        NaN   \n",
              "\n",
              "     Unnamed: 3 Unnamed: 4  \n",
              "4447        NaN        NaN  \n",
              "1774        NaN        NaN  \n",
              "4151        NaN        NaN  \n",
              "4180        NaN        NaN  \n",
              "926         NaN        NaN  "
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-38dbe6b4-5ea0-487d-8924-
83edac74c609\">\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>v1</th>\n",
              "      <th>v2</th>\n",
              "      <th>Unnamed: 2</th>\n",
              "      <th>Unnamed: 3</th>\n",
              "      <th>Unnamed: 4</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>4447</th>\n",
              "      <td>ham</td>\n",
              "      <td>I sent them. Do you like?</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1774</th>\n",
              "      <td>ham</td>\n",
              "      <td>Its ok chikku, and its my 1 of favourite
song....</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4151</th>\n",
              "      <td>ham</td>\n",
              "      <td>What's nannys address?</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4180</th>\n",
              "      <td>ham</td>\n",
              "      <td>Ok da, i already planned. I wil pick
you.</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>926</th>\n",
              "      <td>ham</td>\n",
              "      <td>But I'm on a diet. And I ate 1 too many
slices...</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "      <td>NaN</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-38dbe6b4-5ea0-487d-8924-
83edac74c609')\"\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-38dbe6b4-5ea0-
487d-8924-83edac74c609 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-
38dbe6b4-5ea0-487d-8924-83edac74c609');\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
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df.shape"
      ],
      "metadata": {
        "id": "e4Ckhe53OqqD",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "9fe4a87a-3545-4fdf-96d0-fc5c753baddb"
      },
      "execution_count": 3,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "(5572, 5)"
            ]
          },
          "metadata": {},
          "execution_count": 3
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df.info()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "Bl8uCIdzlWT2",
        "outputId": "ca051d49-19eb-482b-cbd1-40475d40d36c"
      },
      "execution_count": 4,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "<class 'pandas.core.frame.DataFrame'>\n",
            "RangeIndex: 5572 entries, 0 to 5571\n",
            "Data columns (total 5 columns):\n",
            " #   Column      Non-Null Count  Dtype \n",
            "---  ------      --------------  ----- \n",
            " 0   v1          5572 non-null   object\n",
            " 1   v2          5572 non-null   object\n",
            " 2   Unnamed: 2  50 non-null     object\n",
            " 3   Unnamed: 3  12 non-null     object\n",
            " 4   Unnamed: 4  6 non-null      object\n",
            "dtypes: object(5)\n",
            "memory usage: 217.8+ KB\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed:
4'],inplace=True)\n",
        "df.sample(5)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "6c7Bum1ZlZMy",
        "outputId": "3beb1e4f-7ae5-407f-c1f5-a7232a037b89"
      },
      "execution_count": 5,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "       v1                                          
v2\n",
              "4782  ham  Especially since i talk about boston all up 
in...\n",
              "1981  ham  Sorry, I'll call later in meeting any thing 
re...\n",
              "4133  ham  Is it your yahoo boys that bring in the 
perf? ...\n",
              "1523  ham                                    Yup ok 
thanx...\n",
              "1626  ham                          Dear how you. Are 
you ok?"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-3cacb929-063f-4944-8299-
2114122a6858\">\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>v1</th>\n",
              "      <th>v2</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>4782</th>\n",
              "      <td>ham</td>\n",
              "      <td>Especially since i talk about boston all up 
in...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1981</th>\n",
              "      <td>ham</td>\n",
              "      <td>Sorry, I'll call later in meeting any thing
re...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4133</th>\n",
              "      <td>ham</td>\n",
              "      <td>Is it your yahoo boys that bring in the perf?
...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1523</th>\n",
              "      <td>ham</td>\n",
              "      <td>Yup ok thanx...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1626</th>\n",
              "      <td>ham</td>\n",
              "      <td>Dear how you. Are you ok?</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-3cacb929-063f-4944-8299-
2114122a6858')\"\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-3cacb929-063f-
4944-8299-2114122a6858 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-
3cacb929-063f-4944-8299-2114122a6858');\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": 5
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df.rename(columns={'v1':'target','v2':'text'},
inplace=True)\n",
        "df.sample(5)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "dVyhBcu1ld0d",
        "outputId": "5560a4e4-74a7-4efa-a7ce-85fb8cb7f095"
      },
      "execution_count": 6,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "     target
text\n",
              "850     ham  U reach orchard already? U wan 2 go buy
ticket...\n",
              "4804    ham                     Sorry for the delay. Yes
masters\n",
              "5397    ham                        That's necessarily
respectful\n",
              "1744   spam  Someone has conacted our dating service
and en...\n",
              "2643    ham  Hi! You just spoke to MANEESHA V. We'd
like to..."
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-4c176029-f88b-4e66-a00e-
ed722c69a40f\">\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>target</th>\n",
              "      <th>text</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>850</th>\n",
              "      <td>ham</td>\n",
              "      <td>U reach orchard already? U wan 2 go buy
ticket...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4804</th>\n",
              "      <td>ham</td>\n",
              "      <td>Sorry for the delay. Yes masters</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>5397</th>\n",
              "      <td>ham</td>\n",
              "      <td>That's necessarily respectful</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1744</th>\n",
              "      <td>spam</td>\n",
              "      <td>Someone has conacted our dating service and
en...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2643</th>\n",
              "      <td>ham</td>\n",
              "      <td>Hi! You just spoke to MANEESHA V. We'd like
to...</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-4c176029-f88b-4e66-a00e-
ed722c69a40f')\"\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-4c176029-f88b-
4e66-a00e-ed722c69a40f 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-
4c176029-f88b-4e66-a00e-ed722c69a40f');\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": 6
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "from sklearn.preprocessing import LabelEncoder\n",
        "encoder = LabelEncoder()\n",
        "df['target'] = encoder.fit_transform(df['target'])\n",
        "df.head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "lHQOVFOxlhIb",
        "outputId": "d40e7af3-1932-4cef-8fa8-86e4145bace7"
      },
      "execution_count": 7,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   target
text\n",
              "0       0  Go until jurong point, crazy.. Available
only ...\n",
              "1       0                      Ok lar... Joking wif u
oni...\n",
              "2       1  Free entry in 2 a wkly comp to win FA Cup
fina...\n",
              "3       0  U dun say so early hor... U c already then
say...\n",
              "4       0  Nah I don't think he goes to usf, he lives
aro..."
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-662eb83c-c2f8-4ba3-9882-
9611b2bb38b0\">\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>target</th>\n",
              "      <th>text</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>0</td>\n",
              "      <td>Go until jurong point, crazy.. Available only
...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>0</td>\n",
              "      <td>Ok lar... Joking wif u oni...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>1</td>\n",
              "      <td>Free entry in 2 a wkly comp to win FA Cup
fina...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>0</td>\n",
              "      <td>U dun say so early hor... U c already then
say...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4</th>\n",
              "      <td>0</td>\n",
              "      <td>Nah I don't think he goes to usf, he lives
aro...</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-662eb83c-c2f8-4ba3-9882-
9611b2bb38b0')\"\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-662eb83c-c2f8-
4ba3-9882-9611b2bb38b0 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-
662eb83c-c2f8-4ba3-9882-9611b2bb38b0');\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": 7
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df.isnull().sum()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "iscZMrFxlsK-",
        "outputId": "dbe5138f-ddae-4a53-f811-2b7583e87a4e"
      },
      "execution_count": 8,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "target    0\n",
              "text      0\n",
              "dtype: int64"
            ]
          },
          "metadata": {},
          "execution_count": 8
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df.duplicated().sum()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "_o6gQOealvKd",
        "outputId": "001c31c9-7056-4b4a-fa66-9e7e6b57c949"
      },
      "execution_count": 9,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "403"
            ]
          },
          "metadata": {},
          "execution_count": 9
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df = df.drop_duplicates(keep='first')\n",
        "df.duplicated().sum()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "Xk19JaZDl0dg",
        "outputId": "2b537c98-8c71-4778-91c6-0c09c7ad3556"
      },
      "execution_count": 11,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "0"
            ]
          },
          "metadata": {},
          "execution_count": 11
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df.shape"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "AQiDzko0l5Wc",
        "outputId": "f93a4073-5f44-44f3-9f12-25edb5b31bf4"
      },
      "execution_count": 12,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "(5169, 2)"
            ]
          },
          "metadata": {},
          "execution_count": 12
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df.head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "hf20ls40l8S5",
        "outputId": "643b9d96-f953-46a9-a73e-53f501e5df96"
      },
      "execution_count": 13,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   target
text\n",
              "0       0  Go until jurong point, crazy.. Available
only ...\n",
              "1       0                      Ok lar... Joking wif u
oni...\n",
              "2       1  Free entry in 2 a wkly comp to win FA Cup
fina...\n",
              "3       0  U dun say so early hor... U c already then
say...\n",
              "4       0  Nah I don't think he goes to usf, he lives
aro..."
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-53cbe0c7-b167-430c-9756-
51fa54aebb6b\">\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>target</th>\n",
              "      <th>text</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>0</td>\n",
              "      <td>Go until jurong point, crazy.. Available only
...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>0</td>\n",
              "      <td>Ok lar... Joking wif u oni...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>1</td>\n",
              "      <td>Free entry in 2 a wkly comp to win FA Cup
fina...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>0</td>\n",
              "      <td>U dun say so early hor... U c already then
say...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4</th>\n",
              "      <td>0</td>\n",
              "      <td>Nah I don't think he goes to usf, he lives
aro...</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-53cbe0c7-b167-430c-9756-
51fa54aebb6b')\"\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-53cbe0c7-b167-
430c-9756-51fa54aebb6b 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-
53cbe0c7-b167-430c-9756-51fa54aebb6b');\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": [
        "df['target'].value_counts()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "DXLsAfWel_tI",
        "outputId": "f0c4061f-91ad-4541-85da-287c1464f015"
      },
      "execution_count": 15,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "0    4516\n",
              "1     653\n",
              "Name: target, dtype: int64"
            ]
          },
          "metadata": {},
          "execution_count": 15
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "plt.pie(df['target'].value_counts(),
labels=['ham','spam'],autopct=\"%0.2f\")\n",
        "plt.show()\n",
        "\n",
        "# Conclusion: Data is imbalanced"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 248
        },
        "id": "W2sRE9O7mBPL",
        "outputId": "231240c3-1849-4c1f-e110-44a56af9c158"
      },
      "execution_count": 16,
      "outputs": [
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x288 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAPAAAADnCAYAAAAghtuxAAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAYWElEQVR4nO3deZQU1d3G8
e+d6WEYhkU2QYRQBiQIouIGKgrGJWi5RU1wyxu3JGrc1/JNoqPRWJLELZrk1aBGYxI1Joq
WgsagiPsGogKuZZQdhRaYpae77/tHFTDAwPTMdM+t6v59zunDMNNNd9XDOPNxabymtNUKIe
CozHUAI0XZSYCFiTAosRIxJgYWIMSmwEDEmBRYixqxTAQsSYFFiIGJMCCxFjUmAhYkwKLES
MSYGFiDEpsBAxJgUWIsakwELEmBRYiBiTAgsRY1JgIWJMCixEjEmBhYgxKbAQMSYFFiLGp
MBCxJgUWIgYkwILEWNSYCFiTAosRIwlTAcQLbMcrxvwTWAHwAL6Aj228KoEUkBDk9e6v9c
BS4DPN335rr2mw/5BIm+UPNwsOizHKwd2A8OYAI4GdgBFZgNvTPdd+0OKJ
1G5MlKLMHPOm79tuG85Q8KXAHsBxvd+B4guIOMxwnnnxYBTwJTgWm+azcazlNypMAFYjneS
OBUgtLuYDZDZNh1gK3AdM8V17gekwpUpUIKnEfhQaiJqNAQw3HMMAw3HMelF4E/Aw75rrzUd
AcrzewJnA2cBgw3GiZDXwF2Cy79q4q+4SxFSRpDpbjDpbjDQOuAE4Xgh8HDceJkwzwAr
WA5Xj/gaoLN5QrDceIsC/wDuEo
OeLWPFDgH4QUXlwCXEa0LLeKuEbgduMZ37aTpMHEkBW6B5XhHAtcDg2ksksRWwZcCVwj+/a8
gvZClLgLbAcry9wK3Ci6Swl5AXgLN+13zcdJC6kwkM2UGagj+ksJagRuFgszptOkz
USYGbsBzvG8AfCa5TFma9Cpzku/YnpoNEmdzQH7Ic7zhgLLeqBgDvG053kmmg0RZyY/Al
uMlAJfgKLOIpvuAn8qkA5sr6QJbjtcfeMBvfeQso
bFzsCsyzH+57pIFFSkgW2HO8C4D/AdqaziFbpDDxoOd4VRcltQluQluOb7S7
BeX9KmmkilweK/uncDpprOIvJkGfN937dWmg5SEgW2HK8S+DtwjOksIu/eAq9a8
mFH2BwzmVpwITDEcRhfMBMN537SWmg3S0oi5wOFPGj87wkI5UtAS4H2H
K87wZFmKW/pmAscWEpzVxflaSTL8aqAx5HylppRwDOW0RwDOW421jOkhHKboCh5OF/
Twy2wold0BQb+D7BNhxBG7Q08Zjle0U97VFQFthyvBjnPKwITCGZSKWpFcxArvO3sAdM5R
ORc6rv2b02HKJSiKHD4Gf2NXgWrTWUTkZIHDfdeebjpIpIiS4cS+wJbjfR+4Eyg3nUVE0lp
O7lEM+8B/Qsortq4n8C/L8bqAZNCvQNxCEs9Vm1nl2sG07ok5cC/OVmv4pVmNz
4iVicW0PxzLAodPSphLaTx3V+TXQ6SD3HdhL4WKa9om+0JHudSFGI3AluOtyf
wCvJUQNE+x/uu/QvRUEm9JLTQdpj7+KklFkRx+K4Kh0b
EZgy/GGEhy46mw6iygaGu/ZrpoO0VZsRVFfBgEm/xBFaV/
L8b5vOkRbxaLAluMNAJ4GepjOIopSIzDJd+03TAdprcgX2HK8/YC5SHlF4XQBnrIc73DTQVor
0gW2HO8k4GlkoLkovDLgj6ZDtFZkD2JZjncR8DvAMZ1FlIzVwFDftZeaDpKrSG6BLce7FLg
RKa/oWNsAt5sO0RqRK7DleBcCk03nECXrZMvxjjYdIleRmkLbcrzzgDtM5xAl73Nz2JZXN
Xx53j0qKjSi6j0L5lz1rPi+mlk00jk15rk7bwdG4j5k91N3UJQDTnYJtiC9ndgj+b+vVHZc7
xkAzWWdhS6TRS6XwCjj1pd5vxB1vvPLNk+qFdM

ANG
```

2DL8cqA60znECXhf8NZTGMl1wKXK6XuUkq9p5R6WilVpZT6kVLqdaXUHKXUI0qpLgBKqXu
VUn9QSr2ilPpEKTVBKXW3UmqeUureVuY7Cdi5lZ8Roi26AeeaDtFauRZ4R+AOrfVIYBVwH
PBPrfVeWutdCebkPaPJ+3sC+xBslkwlONI3EhillNotlxWG8xnV5JhPiHw4N5zRNDZyLfC
nWuvZ4ddvEuwz7KyUekEpNRc4maCg6zyugxPMc4GlWuu5Wuss8F742VycAAzJ8b1C5EMf4
FTTIVoj1wI3NPk6AySAe4FztdajgGvY+BztuvdnN/lsNvxsLi7M8X1C5NPF4bGXWGhP0G7
AYqVUBcEInDeW440Dds/nMoXI0VDgu6ZD5Ko9Bf4FwURyLwLz8xNnvfPzvDwhWiM2z4+O3
LXQluP1Az4Hin5SbhFp+/qu/bLpEC2J4rb+6Uh5hXmnmg6Qi0iNwJbjKeBjZLYNYV4S6O+
7dr3pIFsTtRF4P6S8Ihp6AMeYDtGSqBX4WNMBhGgir2dXCiFqBY7N4XtREg6N+rOGI1Ngy
/H2IPertIToCJ2I+KASmQIjm88imo43HWBrpMBCbN34KD8oPNfrkgvKcrwRFPABZV+//ih
r5jwNCir6WvQ5/EKWPvhzsqk6ALK1STptN4xtj/35Rp9LJ5ex/F/Xo3UWMhm67XEE3UYfT
rahliV/vWL9+zKrv6R6xAR6HfzjQv0ThDnVBBNKzDIdpDmRKDDwnUItOL16BV+/+TgDzvg
9ZRWVLH/UZe28mfQ/efL69yz/16+o2nHzST/Ku/ak/ym/QSUqyKbqWDTlp1QNHUOiW28Gn
Pa79e9bfO8FdPnWvoX6JwjzDiaiBY7KJvR+BV16NoNOp9DZDDrdQHnXXht+1FBL/Wdz6LL
jPpt9TJVXoBLB1pPONEIzF700frWQTG2SyoEjN/uZKBoHmQ6wJVEZgQs2fCW69aH73t9l4
R9OQyU60XmH0VTtsOFGp9oPX6bz4F0pq2z+0bHpr5ez7B/XkF65mJ4Hnkai28azrqydN5P
q4fujlCrUP0GYN8ZyvK6+a68xHWRTxkdgy/F2ALYr1PIz9Wuo/fBVtj9rCgN/eh+6sYE17
81Y//O178+ky4jxW/x8ontfBpx+OwN+fCdr3n2WzNqVG/28dt7WPy+KQgVwgOkQzTFeYAq
8+VzvzybRox/lXXqgyhN0GbYPDQvnAZCpTZJa/AFdhuzV4nIS3XpT0Wcw9Z+/t/57qWWfo
LMZKvsPLVh+ERkHmg7QnCgUuKBHfxLd+5JatIBsYz1aa+o/m0NF70EA1C54kaqhe6ESzT8
jPP31CrKNwYQimfo1NHzxPhW9B67/+dr3Z1K9k4y+JWK06QDNicI+cEFH4MoB36LLt/Zj8
b0XosrK6NRvCN12nQgE+689xn5vo/c3LP6QNbOfovdh59P45eesnDFl/c+6730snfpa6/9
eO/8Ftv1eTSHji+iI5OyoRm8ntByvM7CWaGwJCNGSvr5rrzAdoinTxRkSgQxC5Cpyo7Dp8
sjRHxEno0wH2JQUWIjcyQi8CSmwiBMp8CakwCJOBpsOsCkpsBC562s6wKaMnUYK77GsA8q
NBBCibXr5rr2y5bd1DJMjcG+kvCJ+tjUdoCmTBe5pcN1CtJUUOCQFFnEkBQ5FerpOIbZAC
hyqNrhuIdqqV8tv6TgmC9z8FBhCRFukZqiUAgvROpE6c2KywKYvIhGiLaJwD/16JsM0GFx
3UevHV8tmVF6ysjOpyF05FHcNVKRguekY65kscKSfuxpnS+m17ZGp6+qe7nR5qlzp/qbzF
JMqUpHacjQZRgpcQB/r7QcfnrqhLqPVMtNZikzadICmpMBFbIH+xg5HpH61OqtVdLb54q/
RdICmpMBFbp4ePOTI1HXJrFaRmsspxiL1e2uywHUG111S3tM7DP1u6pqvslp9ZTpLEVhiO
kBTJgtca3DdJWeOHjrs+NTVy7KaVaazxNxi0wGaMlngRQbXXZLe0sOGT0pdtVhrkqazxFi
kfm9NFngZEdufKAWv6+E7ndj4sy+05mvTWWJKRmAA37U18Lmp9ZeyV7IjR/6g8crPtGa16
Swxs5aaZKT+4zN9Uvq/htdfsmZlR406rfHyT7VmreksMRKp0RekwCXtuexuu5zZeMmHWss
BxRxFav8XpMAl79nsHrud1XjhAq3ltF4OZATexGeG1y+A6dm9R5/beP77WstBxRYsMB1gU
6YL/L7h9YuQlx27x0WN57yrtdwlthVvmg6wKdMFnk3Eri0tZY9mx+15Wfonc7QmZTpLRL1
hOsCmjD4fGMByvLeI6NPPS9UJ5f959YbEn3ZXKj/Tx5z+WB1PfJBm22rFu+d0BeCyp+t5/
IM0ncphSK8y7jm6im06q80+u6pec+bUOt5dlkUpuPuozuwzKMEv/lPPYwvSlCnYtlpx7zF
VDOhW0PFoMTXJAYVcQVuYHoEBXjcdQGzs75lvj7kqfeqbWufn1rlTd6tg2ikbz6B0yJAE7
55TzTtnd2VYrzJueKH5LfcLptUzcWiC+ed2Zc5Z1ezUN5jR5rL9Knnn7K7MPqsrRwxLcO3
zBd/yj9zmM0iBxRbcnzl07LXpH7yuNZn2LuuAwQl6VW08uh46JEGiLPje2IHlfLE6u9nnk
vWamZ+lOWN0sCHQqVytH6W7V25Y3toUbD52510kCxyF+X2kwBF1T+awfRJkX/rfxANjlCr
cZG53z25k0sjNfxU/XZWlbxfFaY/VM2dphj22K+fWiZ2p7hTU9WfP1nPfO430qFTM+GHB5
0iMZIGjMAK/h9xaGFl3Zex9f52e9IrWbD5E5sH1MxtIlMHJozbf3U5n4a3FWc7es4K3f9K
V6gqFO2vDpvL1B3Xm84u6cfKoCm5/reDH3SJ3AAsiUGDftdPAa6ZziC37febo/W5JH/eS1
uT1iOe9s1M88WGaB46tQqnNN4IHdlcM7K4YMzAYnY8fkeCtJZv/P3LyLhU8Mq+gM918Tk0
ychdxQAQKHHradACxdbdmjht3R+boWfkq8bSP0kx+McXUE6roUtH8Hmz/rmUM6lHGghXBb
vizn6YZ0Sf4lf3wyw275o/NTzO8T0F/lZ8o5MLbw/hpJADL8XYnovsYYmNO4q8zz0o8cUB
rPnPiI7U852dYUavpV624ZkIlN8xqoCEDvas2HMj64xFVLFqd5cyp9Tx5crBPO3tJhjOn1
pHKwDd7BqebelYpjnuolgUrspQpGLxNGX+0O7N994KVeCI1yemFWnh7RKXAiuA6036ms4i
W/SJx//NnJJ4abzpHB1kN9KEmGcmLWyKxCR3eG+yZziFy88v0D8bflz7kedM5Osi0qJYXI
lLg0KOmA4jcXZU+bfzf0geWQokfMx1ga6JU4GeANaZDiNxdmf7R+H9kDnjOdI4CSgNPmg6
xNZEpsO/a9chmdOxc2njWhKmZfZ4znaNAXqAmudJ0iK2JTIFDd5sOIFrv/MbzJjyV2asYN
6cjv1sXtQI/A/imQ4jWO7vxovHPZHZ/znSOPEoBfzUdoiWRKnB4NHqK6RyibX7UeOmEGZl
dnzOdI0/+SU0y8o+jiVSBQ/dA+++AEWac1njFhBczI4thc/pO0wFyEbkC+669EHjKdA7Rd

```
ic3/mz8q9nhcS7xB9QkZ5gOkYvIFTh0l+kAon0mpa4a/0Z22EzTOdrod6YD5CqqBfaQGSt
j7/jU1fvPzg55wXSOVlpFsBu3VUqpaqWUp5Sao5R6Vyk1SSnlK6UmK6XmKqVeU0oNDd97p
FLqVaXU20qpfyul+oXfr1FK/Vkp9YJS6jOl1LFNPj9NKdXilEaRLLDv2hngBtM5RHspdUz
q2nHvZq04lXgKNclcnlYxEViktd5Va70zMC38flJrPQq4Hbgl/N4sYKzWejTwd+DyJssZA
nwbOAr4CzAj/HwdYLcUIpIFDt2DPDupCCh1ZOq6/eZnB80ynSQHjeS++TwXOEQpdaNSan+
t9bonPv6tyZ/7hF8PBKYrpeYClwEjmyznKa11Y7i8cjb8RzAXsFoKEdkC+66dQkbhoqApK
zssdcO+H2S3f9F0lhbcRU0yp103rfUHwO4ERbtOKXXVuh81fVv45++A28OR9SdA5ybvaQi
XlwUa9YbbA7PkMOVVZAscmgJ8YTqEaD9NWdnE1I1jP8lu95LpLFuwBrgm1zcrpQYAtVrrv
wC/JigzwKQmf74cft0DWBh+/cP2R90g0gUOR2HXdA6RH1nKyg9JTd7bz/Z7ueV3d7jfUpN
c1or3jwJeU0rNBq4Grgu/31Mp9Q5wAXi8OicQN/VtjOV4l8DGwveksI
j8SpBtndLr4rUFlK8aYzhJaCgylJtmuu+GUUj6wp9a6w67givQIDOC7dgPwc9M5RP6kSVQ
cmLpp94W6d1QmM/xle8trSuQLHPozwaF4USTSJComNNy82xLd0/S84B+Rp8smtdZWR46+E
JMChzc5nAP5edSHiIZGEp0OaLhll2W6h8kJDX9GTTK2D9iLRYEBfNeeC9xmOofIrxQVlQc
03DJihe7+loHVPw88bGC9eRObAoeuZsPheFEk6qms2r/h1uFf6W6zO3C1a4DTqElG+yhuC
2JVYN+11wAXm84h8q+Oyi7jGm7dcZWuntNBq7yUmuSnHbSugon8aaTmWI7nAYebziHyr5q
6NS9VnvdpD1U7qoCreZqa5HcKuPwOE6sRuIkzgOWmQ4j8W0tV1/0abhu8Wle9V6BVJAl+f
4pCLAvsu/YS4HTTOURhrKFL930bbhu4Rnd+vwCLv4CaZNFcnhvLAgP4rv0EcIfpHKIwVlP
dY7+G2wbU6sr5eVzsVGqSf87j8oyLbYFDlwAmTj+IDpCk6zb7NtzWr1Z3WpCHxS0CfpyH5
URKrAscXmb5feBr01lEYayiW89xDbf2qdcVH7ZjMXXA0dQkl+YrV1TEusAAvmt/THCLVkG
eIC/M+4oevcc13LpNva74uA0f18Cp1CTfyHeuKIh9gQF8136UYKYDUaRWsE3f8Q03d2vQi
daeu72WmuRDBQkVAUVRYADftW8iRrMJitZbSq9tJzTcXJXSCT/HjzxEK27Sj6OiKXDoQiL
+OEjRPovp3X9Cw02Vjbq8palv3iDYdI7flUqtEMsrsbbGcrwqYAYQlZvFRQEMVMsXzeh0c
aZCZQY18+NFwF7UJBd1dK6OVmwjML5r1wFHAp+YziIK5wvdd8BBqd+otC7b9OaWFcDEUig
vFGGBAXzXXg4cjDzpsKj9V/cbeEhqciatyxaH3/oKOJia5FyTuTpS0W1CN2U53iDgWWBH0
1lE4QxRCz+b1smpr1CZE6lJvm06T0cq6gIDWI7Xn6DEI0xnEQXzZRUNh8xzjy2p8kIJFBj
Acry+wNPAbqaziLxbAhzsu3ah7l6KtKLcB95UuE/8bSAqsyCK/PCB/Uu1vFAiBQbwXXslc
AjwpOksIi9eAPb2Xfsj00FMKpkCA/iu/TXBKSZ52kO8TSHYbC75SR1KYh+4OZbjnUDwi9D
FdBaRswxwqe/at7T4zhJRsgUGsBxvNPAo8A3TWUSLksAk37Wnmw4SJSW1Cb0p37XfBvYkm
B9YRNc7wBgp7+ZKusCw/gj1QQRzTsuTH6IlC9wI7OW7dj5m5Sg6Jb0JvSnL8fYG7geGmc4
i+BT4H9+15ZlYW1HyI3BTvmu/BowGbkFm+DBpCrCLlLdlMgJvgeV4+wB3A8NNZykhi4Czf
deeajpIXMgIvAW+a79McOnllcikeYVWR/CE+2FS3taRETgHluNtC9QQTEtabjZN0XkQuNx
37f+aDhJHUuBWsBxvJ+DXgG06SxF4A7jQd+0XTQeJMylwG1iOdxAwGdjddJYYWkCwufxA+
OB20Q5S4HawHO9g4FKgKJ50V2BzgOuBR3zXliP8eSIFzgPL8XYmeMzLSUAnw3GiZjpwk+/
aT5sOUoykwHlkOd52wHkEB7t6G45j0kqCg1O3l/K9uh1BClwAluNVEGxWnwQcBVSbTdQhU
oBHcCWb57t2ynCekiAFLjDL8aqBownKfChQYTZR3r1EUNqHfNf+ynSYUiMF7kCW4/UmKPN
3CG6giONm9nKCSQKfAZ7xXftzw3lKmhTYEMvxyghOQx0IHACMA7YxGqp5dQTT1/yboLRz5
PRPdEiBIyIs9AhgFDCyyWsIHXfJ6xcE996ue80F5vuuLbdZRpQUOOIsx+tMcEPFSGAQ0Af
o2+S17u/NHSjT4StLMKPFivC1nODGgYXh61Ngbjjxn4gRKXCRsBxv3TXaWdnELR1SYCFiT
G4nFCLGpMBCxJgUWIgYkwILEWNSYCFiTAosRIxJgYWIMSmwEDEmBRYixqTAQsSYFFiIGJM
CCxFjUmAhYkwKLESMSYGFiDEpsBAxJgUWIsakwELEmBRYiBiTAgsRY1JgIWJMCixEjEmBh
YgxKbAQMSYFFiLGpMBCxJgUWIgY+39fUkrh4uNRDQAAAABJRU5ErkJggg==\n"
          },
          "metadata": {}
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "import nltk\n",
        "nltk.download('punkt')"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "1mhnZibzmbxp",
        "outputId": "a89bba45-37dd-484e-9604-4fc8b8812fd6"
      },
      "execution_count": 17,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stderr",
          "text": [
            "[nltk_data] Downloading package punkt to
/root/nltk_data...\n",
            "[nltk_data]   Unzipping tokenizers/punkt.zip.\n"
          ]
        },
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "True"
            ]
          },
          "metadata": {},
          "execution_count": 17
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "\n",
        "df['num_characters'] = df['text'].apply(len)\n",
        "df.head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "yZQyW92xmfVL",
        "outputId": "607ce6e4-894a-4e89-f76b-e11d460030ac"
      },
      "execution_count": 18,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   target
text  num_characters\n",
              "0       0  Go until jurong point, crazy.. Available
only ...             111\n",
              "1       0                      Ok lar... Joking wif u
oni...              29\n",
              "2       1  Free entry in 2 a wkly comp to win FA Cup
fina...             155\n",
              "3       0  U dun say so early hor... U c already then
say...              49\n",
              "4       0  Nah I don't think he goes to usf, he lives
aro...              61"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-002f615f-9b75-4a53-924e-
ff945286640d\">\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>target</th>\n",
              "      <th>text</th>\n",
              "      <th>num_characters</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>0</td>\n",
              "      <td>Go until jurong point, crazy.. Available only
...</td>\n",
              "      <td>111</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>0</td>\n",
              "      <td>Ok lar... Joking wif u oni...</td>\n",
              "      <td>29</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>1</td>\n",
              "      <td>Free entry in 2 a wkly comp to win FA Cup
fina...</td>\n",
              "      <td>155</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>0</td>\n",
              "      <td>U dun say so early hor... U c already then
say...</td>\n",
              "      <td>49</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4</th>\n",
              "      <td>0</td>\n",
              "      <td>Nah I don't think he goes to usf, he lives
aro...</td>\n",
              "      <td>61</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-002f615f-9b75-4a53-924e-
ff945286640d')\"\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-002f615f-9b75-
4a53-924e-ff945286640d 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-
002f615f-9b75-4a53-924e-ff945286640d');\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": 18
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df['num_words'] = df['text'].apply(lambda
x:len(nltk.word_tokenize(x)))\n",
        "df.head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "JoaXgpLDmqRY",
        "outputId": "1ff9c994-e165-4855-efc2-a1c347139e76"
      },
      "execution_count": 19,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   target
text  num_characters  \\\n",
              "0       0  Go until jurong point, crazy.. Available
only ...             111   \n",
              "1       0                      Ok lar... Joking wif u
oni...              29   \n",
              "2       1  Free entry in 2 a wkly comp to win FA Cup
fina...             155   \n",
              "3       0  U dun say so early hor... U c already then
say...              49   \n",
              "4       0  Nah I don't think he goes to usf, he lives
aro...              61   \n",
              "\n",
              "   num_words  \n",
              "0         24  \n",
              "1          8  \n",
              "2         37  \n",
              "3         13  \n",
              "4         15  "
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-36b00ba4-38f5-4856-b286-
2e1687fe528c\">\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>target</th>\n",
              "      <th>text</th>\n",
              "      <th>num_characters</th>\n",
              "      <th>num_words</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>0</td>\n",
              "      <td>Go until jurong point, crazy.. Available only
...</td>\n",
              "      <td>111</td>\n",
              "      <td>24</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>0</td>\n",
              "      <td>Ok lar... Joking wif u oni...</td>\n",
              "      <td>29</td>\n",
              "      <td>8</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>1</td>\n",
              "      <td>Free entry in 2 a wkly comp to win FA Cup
fina...</td>\n",
              "      <td>155</td>\n",
              "      <td>37</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>0</td>\n",
              "      <td>U dun say so early hor... U c already then
say...</td>\n",
              "      <td>49</td>\n",
              "      <td>13</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4</th>\n",
              "      <td>0</td>\n",
              "      <td>Nah I don't think he goes to usf, he lives
aro...</td>\n",
              "      <td>61</td>\n",
              "      <td>15</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-36b00ba4-38f5-4856-b286-
2e1687fe528c')\"\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-36b00ba4-38f5-
4856-b286-2e1687fe528c 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-
36b00ba4-38f5-4856-b286-2e1687fe528c');\n",
              "          const dataTable =\n",
              "            await
google.colab.kernel.invokeFunction('convertToInteractive',\n",
              "
[key], {});\n",
              "          if (!dataTable) return;\n",
              "\n",
              "          const docLinkHtml = 'Like what you see? Visit
the ' +\n",
              "            '<a target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data
table notebook</a>'\n",
              "            + ' to learn more about interactive
tables.';\n",
              "          element.innerHTML = '';\n",
              "          dataTable['output_type'] =
'display_data';\n",
              "          await
google.colab.output.renderOutput(dataTable, element);\n",
              "          const docLink =
document.createElement('div');\n",
              "          docLink.innerHTML = docLinkHtml;\n",
              "          element.appendChild(docLink);\n",
              "        }\n",
              "      </script>\n",
              "    </div>\n",
              "  </div>\n",
              "  "
            ]
          },
          "metadata": {},
          "execution_count": 19
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df['num_sentences'] = df['text'].apply(lambda
x:len(nltk.sent_tokenize(x)))\n",
        "df.head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "aZ9ti7zAmxxZ",
        "outputId": "2e893526-7b0f-4bf2-a3a4-88aec9105a56"
      },
      "execution_count": 20,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   target
text  num_characters  \\\n",
              "0       0  Go until jurong point, crazy.. Available
only ...             111   \n",
              "1       0                      Ok lar... Joking wif u
oni...              29   \n",
              "2       1  Free entry in 2 a wkly comp to win FA Cup
fina...             155   \n",
              "3       0  U dun say so early hor... U c already then
say...              49   \n",
              "4       0  Nah I don't think he goes to usf, he lives
aro...              61   \n",
              "\n",
              "   num_words  num_sentences  \n",
              "0          24              2  \n",
              "1           8              2  \n",
              "2          37              2  \n",
              "3          13              1  \n",
              "4          15              1  "
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-526449ad-2d3b-4d77-98da-
b2b2d002d62e\">\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>target</th>\n",
              "      <th>text</th>\n",
              "      <th>num_characters</th>\n",
              "      <th>num_words</th>\n",
              "      <th>num_sentences</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>0</td>\n",
              "      <td>Go until jurong point, crazy.. Available only
...</td>\n",
              "      <td>111</td>\n",
              "      <td>24</td>\n",
              "      <td>2</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>0</td>\n",
              "      <td>Ok lar... Joking wif u oni...</td>\n",
              "      <td>29</td>\n",
              "      <td>8</td>\n",
              "      <td>2</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>1</td>\n",
              "      <td>Free entry in 2 a wkly comp to win FA Cup
fina...</td>\n",
              "      <td>155</td>\n",
              "      <td>37</td>\n",
              "      <td>2</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>0</td>\n",
              "      <td>U dun say so early hor... U c already then
say...</td>\n",
              "      <td>49</td>\n",
              "      <td>13</td>\n",
              "      <td>1</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4</th>\n",
              "      <td>0</td>\n",
              "      <td>Nah I don't think he goes to usf, he lives
aro...</td>\n",
              "      <td>61</td>\n",
              "      <td>15</td>\n",
              "      <td>1</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-526449ad-2d3b-4d77-98da-
b2b2d002d62e')\"\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-526449ad-2d3b-
4d77-98da-b2b2d002d62e 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-
526449ad-2d3b-4d77-98da-b2b2d002d62e');\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": 20
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [

"df[['num_characters','num_words','num_sentences']].describe()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 300
        },
        "id": "5WU8IMt_nDFe",
        "outputId": "c80cbfbf-7c52-4a0b-b493-31907b31d044"
      },
      "execution_count": 21,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "       num_characters    num_words  num_sentences\n",
              "count     5169.000000  5169.000000    5169.000000\n",
              "mean        78.977945    18.453279       1.947185\n",
              "std         58.236293    13.324793       1.362406\n",
              "min          2.000000     1.000000       1.000000\n",
              "25%         36.000000     9.000000       1.000000\n",
              "50%         60.000000    15.000000       1.000000\n",
              "75%        117.000000    26.000000       2.000000\n",
              "max        910.000000   220.000000      28.000000"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-c30a59f7-7373-459c-be70-
a23623f47f58\">\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>num_characters</th>\n",
"      <th>num_words</th>\n",
"      <th>num_sentences</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>count</th>\n",
"      <td>5169.000000</td>\n",
"      <td>5169.000000</td>\n",
"      <td>5169.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>mean</th>\n",
"      <td>78.977945</td>\n",
"      <td>18.453279</td>\n",
"      <td>1.947185</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>std</th>\n",
"      <td>58.236293</td>\n",
"      <td>13.324793</td>\n",
"      <td>1.362406</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>min</th>\n",
"      <td>2.000000</td>\n",
"      <td>1.000000</td>\n",
"      <td>1.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25%</th>\n",
"      <td>36.000000</td>\n",
              "      <td>9.000000</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>50%</th>\n",
              "      <td>60.000000</td>\n",
              "      <td>15.000000</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>75%</th>\n",
              "      <td>117.000000</td>\n",
              "      <td>26.000000</td>\n",
              "      <td>2.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>max</th>\n",
              "      <td>910.000000</td>\n",
              "      <td>220.000000</td>\n",
              "      <td>28.000000</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-c30a59f7-7373-459c-be70-
a23623f47f58')\"\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-c30a59f7-7373-
459c-be70-a23623f47f58 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-
c30a59f7-7373-459c-be70-a23623f47f58');\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": 21
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df[df['target'] ==
0][['num_characters','num_words','num_sentences']].describe()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 300
        },
        "id": "s_Q-9JmbnGN_",
        "outputId": "94274a98-3b7f-47bb-a3d9-56caa38d52dd"
      },
      "execution_count": 22,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "       num_characters    num_words  num_sentences\n",
              "count     4516.000000  4516.000000    4516.000000\n",
              "mean        70.459256    17.120903       1.799601\n",
              "std         56.358207    13.493725       1.278465\n",
              "min          2.000000     1.000000       1.000000\n",
              "25%         34.000000     8.000000       1.000000\n",
              "50%         52.000000    13.000000       1.000000\n",
              "75%         90.000000    22.000000       2.000000\n",
              "max        910.000000   220.000000      28.000000"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-2c1b211d-c23c-4925-b7b7-
01c7fbc5d3d2\">\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>num_characters</th>\n",
              "      <th>num_words</th>\n",
              "      <th>num_sentences</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>count</th>\n",
              "      <td>4516.000000</td>\n",
              "      <td>4516.000000</td>\n",
              "      <td>4516.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>mean</th>\n",
              "      <td>70.459256</td>\n",
              "      <td>17.120903</td>\n",
              "      <td>1.799601</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>std</th>\n",
              "      <td>56.358207</td>\n",
              "      <td>13.493725</td>\n",
              "      <td>1.278465</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>min</th>\n",
              "      <td>2.000000</td>\n",
              "      <td>1.000000</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>25%</th>\n",
              "      <td>34.000000</td>\n",
              "      <td>8.000000</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>50%</th>\n",
              "      <td>52.000000</td>\n",
              "      <td>13.000000</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>75%</th>\n",
              "      <td>90.000000</td>\n",
              "      <td>22.000000</td>\n",
              "      <td>2.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>max</th>\n",
              "      <td>910.000000</td>\n",
              "      <td>220.000000</td>\n",
              "      <td>28.000000</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-2c1b211d-c23c-4925-b7b7-
01c7fbc5d3d2')\"\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-2c1b211d-c23c-
4925-b7b7-01c7fbc5d3d2 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-
2c1b211d-c23c-4925-b7b7-01c7fbc5d3d2');\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": 22
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "df[df['target'] ==
1][['num_characters','num_words','num_sentences']].describe()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 300
        },
        "id": "D4yni3iBnJ2O",
        "outputId": "d211b009-9910-4934-b0ad-75ba87a65675"
      },
      "execution_count": 23,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "       num_characters   num_words  num_sentences\n",
              "count      653.000000  653.000000     653.000000\n",
              "mean       137.891271   27.667688       2.967841\n",
              "std         30.137753    7.008418       1.483201\n",
              "min         13.000000    2.000000       1.000000\n",
              "25%        132.000000   25.000000       2.000000\n",
              "50%        149.000000   29.000000       3.000000\n",
              "75%        157.000000   32.000000       4.000000\n",
              "max        224.000000   46.000000       8.000000"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-ad79fbff-2540-4d92-a4b9-
fa91808772a6\">\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>num_characters</th>\n",
              "      <th>num_words</th>\n",
              "      <th>num_sentences</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>count</th>\n",
              "      <td>653.000000</td>\n",
              "      <td>653.000000</td>\n",
              "      <td>653.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>mean</th>\n",
              "      <td>137.891271</td>\n",
              "      <td>27.667688</td>\n",
              "      <td>2.967841</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>std</th>\n",
              "      <td>30.137753</td>\n",
              "      <td>7.008418</td>\n",
              "      <td>1.483201</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>min</th>\n",
              "      <td>13.000000</td>\n",
              "      <td>2.000000</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>25%</th>\n",
              "      <td>132.000000</td>\n",
              "      <td>25.000000</td>\n",
              "      <td>2.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>50%</th>\n",
              "      <td>149.000000</td>\n",
              "      <td>29.000000</td>\n",
              "      <td>3.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>75%</th>\n",
              "      <td>157.000000</td>\n",
              "      <td>32.000000</td>\n",
              "      <td>4.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>max</th>\n",
              "      <td>224.000000</td>\n",
              "      <td>46.000000</td>\n",
              "      <td>8.000000</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-ad79fbff-2540-4d92-a4b9-
fa91808772a6')\"\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-ad79fbff-2540-
4d92-a4b9-fa91808772a6 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-
ad79fbff-2540-4d92-a4b9-fa91808772a6');\n",
              "          const dataTable =\n",
              "            await
google.colab.kernel.invokeFunction('convertToInteractive',\n",
              "
[key], {});\n",
              "          if (!dataTable) return;\n",
              "\n",
              "          const docLinkHtml = 'Like what you see? Visit
the ' +\n",
              "            '<a target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data
table notebook</a>'\n",
              "            + ' to learn more about interactive
tables.';\n",
              "          element.innerHTML = '';\n",
              "          dataTable['output_type'] =
'display_data';\n",
              "          await
google.colab.output.renderOutput(dataTable, element);\n",
              "          const docLink =
document.createElement('div');\n",
              "          docLink.innerHTML = docLinkHtml;\n",
              "          element.appendChild(docLink);\n",
              "        }\n",
              "      </script>\n",
              "    </div>\n",
              "  </div>\n",
              "  "
            ]
          },
          "metadata": {},
          "execution_count": 23
        }
```

```
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "plt.figure(figsize=(12,6))\n",
        "sns.histplot(df[df['target'] == 0]['num_characters'],
color='green')\n",
        "sns.histplot(df[df['target'] == 1]['num_characters'],
color='red')"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 406
        },
        "id": "ESV0JrkVntRf",
        "outputId": "341f6737-c77f-4aba-84cf-5ac553c6872c"
      },
      "execution_count": 24,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at
0x7f5f8ed69950>"
            ]
          },
          "metadata": {},
          "execution_count": 24
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 864x432 with 1 Axes>"
            ],
            "image/png":
```

"iVBORw0KGgoAAAANSUhEUgAAAtMAAAF0CAYAAADhBbX6AAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3df5RdZX3v8
fd3MslMGEIIhiIkwPgjghaqmGgxeu9Sae9Sa4utaO2iys3FpqvVW621lvb2trbLrEVXbf3
R9nqbSlPsovUH6oV6uVbkRy2LgmRQGAQqaSBmIoEEISQzzmzvf+cfYMZyaH5MzOnDlnZ
t6vtWbNfp69z97fmdnr5HOePHvvvyEwkSZIkzVxHqwuQJEmS5ivDtCRJklSSYVqSJEkqyTA
tSZIklWSYliRJkkoyTEuSJEkldba6gOOxevXq7O3tbXUZiRJWuD6+vr2Z+ap0/vndZju7
e1l+/btrS5DkiRJC1xE7KrXbjMPSZIkqSTDtCRJklSSYVqSJEkqyTAtSZIkldDtCRJklSSYVqSJEkqyTAtSZIklWSYliRJkko
yTEuSJEklGaYlSZKkkgzTkiRJUkmGaUmSJKkkw7QkSZJUkmFakiRJKskwLUmSJJXU2eoCF
rORkRH6+vqm9K1fv56urq4WVSRJkqSZMEy3UF9fH3/8V5ey+pzlAOz/4SC8D5773Oe2uDJJkiQ1wODbNu8jY0Mkm
SJDXCMN1iPWt7WLluZavLkCRJUgnOmZYkSZJKMkxLkyRJkmgzTkiRJUkmSpJIM05IkSVJJhmlJk
iSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSjJMS5IkSSUZpiVJkqSSDNOSJElSSYZ
pSZIkqSTDtCRJklSSYVqSJEkqqalhOiIeiYj+iPhuRGj+iPhuRGwv+k6JiBsj4qHi0xGxI

```
yLujYhXNrM2SZIk6XjNxcj0GzLzFZm5oWhfAdyUmeuAm4o2wJuBdcXXZuAzc1CbJEmSVFo
rpnlcDFxdLF8NvK2m/3NZdQdwckSc3oL6JEmSpIY0O0wn8I2I6IuIzUXfaZn5aLG8FzitW
F4D7K557UDRN0VEbI6I7RGxfd++fc2qW5IkSTqmzibv/3WZuScifgy4MSIerF2ZmRkROZM
dZuZWYCvAhg0bZvRaSZIkaTY1dWQ6M/cU3x8Hvgq8GnhsYvpG8f3xYvM9wJk1L19b9EmSJ
EltqWlhOiJ6ImLFxDLwX4D7gOuBy4rNLgOuK5avB95T3NXjQuBAzXQQSZIkqe00c5rHacB
XI2LiOP+QmV+PiLuAL0bE5cAu4J3F9jcAbwF2AEPApibWZpiibWJkmSJB23poXpzNwJvLxO/xPAR
XX6E3hfs+qRJEmSZptPQJQkSZJKMkxLkiRJJRmmJUmSpJIM05IkSVJJzX5oiwojIyP09fV
N6evv76eSlcl2ZaxCf3//Ea9dv349XvdTa9RkiRJM2OYniN9fX1s2rqJNta9XkbS9MR2fX1s2rqJnrU9k337795P9
zndrGIVAEN7h9jy8BZW7149uc3gwCDbNm9j48aNc16zJEmSjs4wPYd61vawct3KyfahgUN
HbNN9RveUbSRJktS+nDMtSZIklWSYliRJkkoyTEuSJEklGaYlSZKkkgzTkiRJUkmGaUmSJ
Kkkw7QkSZJUkmFakiRJKskwLUmSJJVkmJYkSZJKMkxLkiRJJRmmJUmSpJIM05IkSVJJhml
JkiSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSjJMS5IkSSUZpiVJkqSSDNOSJElSS
YZpSZIkqSTDtCRJklSSYVqSJEkqyTAtSZIklWSYliRJkkoyTEuSJEklGaYlSZKkkgzTkiR
JUkmGaUmSJKkkw7QkSZJUkmFakiRJKskwLUmSJJVkmJYkSZJKMkxLkiRJJRmmJUmSpJIM0
5IkSVJJhmlJkiSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSmp6mI6IJRHxnYj4WtF
+QUTcGRE7IuILEbbGs6O8q2juK9b3Nrk2SJEk6HnMxMv0B4IGa9p8An8jcPZcX/ZcDT
xb9nyi2kyRJktpWU8N0RKwFgb4bNEO4I3AtcUmVwNvK5YvLtoU6y8qtpckSZLaUrNHpj8
JfASoF3nAU9l5ljRHgDWFMtrgN0AxfoDxfZTRMtmiNgeEdv37xvXzXzNolSZKko2pamI6It
wKPZ2bfbO43M7dm5q4lSZKkGels4r5fC/xcRLwF6AZj4FnBwRncXo81p
gT7H9HuBMYCAiOoGVwBNNrE+SJEk6Lk0bmc7M383MtZnZC7wLuDkzLwVuAS4pNrsMuK5Yv
r5oU6y/OTOzWfVJkiRJx6sV95n+n+HeBDEbGD6pzoq4r+q4DnFf0fAq5oQW2SJElSw5o5zWN
SZt4K3Fos7wReXebYeAdc1GPJEmSNBt8AqIkSZJUkmFakiRJKskwLUmSJJVkmJYkSZJKm
pMLEFVeZaxCf3//lL7169fj9tdXVooSZI0wTDd5ob2DrH5a5tZtXoVAEMD/A9xndrFy3stVlSJIkaRrnTEuSJEklGaYlSZKkkgzTkiRJUkmGaUmSJKkkw7QkSZJUkmFakiRJKskwLUmSJJVkmJYkSZJKMkxLkiRJJRmmJUmSpJIM05IkSVJJhWpIkSSrJMC1JkiSVZJiWJEmSjJMS5IkSSUZpiVJkqSSDNOSJElSSYZpSZIkqSTDtCRJklSSYVqSJEkqyTAtSZIklWSYliRJkkoyTEuSJEklGaYlSZKkkgzTkiRJUkmGaUmSJKkkw7QkSZJUkmFakiRJKskwLUmSJJVkmJYkSZJKMkxLkiRJJRmmJUmSpJIM05IkS
VJJhmlJkiSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSjJMS5IkSSUZpiVJkqSSmha
mI6I7Ir4dEfdExPci4/K+rqK9o1jf26zaJEmSpNnQzJHpEeBNmflK4HzgzRFxIfBx4BOZeTbwJHB5sf3lwJNF/ycWSm1iEmmSpJbVaR0YjubB2C7MH5FhLCGG2JnnUvGd4uOo5ul95HLz7ZmtQEjkVXyr2
mI6I7Ir4dEfdExPci4o+K/hdExJ0RsSMivhARy4r+rqK9o1jf26zaJEmSpNnQzJHpEeCNm
fly4BXAmyLiQuBPgE9k5ouBJ4HLi+0vB54s+j9s+j9RbCdJkiS1rYbCdES8tpG+ebJJvaSBvaSbjszb5S8k2p0JwrBv7w9MbwSDYBc3DKBRIJ1ekFaHCrrX
SeCNwbdF/NfC2Yvniok2x/qKIbiEbqkyRJklqh0ZHpv2iwb4qtS3ZmpfFHRjPq8fiqHmo9Bvh8g0vXW9Z+rKlf8gDkvpUdXbGhWiVn4mYeacWtwSaFkvYEh3BxL/R8mpZ93RBZk4WjyqeVCPVIPIaJ0mOS
k0GgDXF8hpgN0Cx/gDwvDr73BwR2yNi+759+xosX5IkSZp9nUdbGRIXAC2BWI2pFEsjwvAtBLkbScXp6M5VYmtR7Q1TVXAh5X6gAjc4TnO9VcRFdpQpwBxQBE2
sOdbOM3MceEVEnAx8FTj3OGqd6Gqd2OdWYCvAhg0b8nj3J0mSJJV1rJHpZcCJVEP3ipqvvcm4FLG
j1IZj4F3AK8Bjg5IiZC/FpgT7G8BzgToFi/Enii0WNIkiRJc+2oy5wXBuPtO/NTIt2RNL92TnOsE/z1Tpw2dWCu5oMM/pfW19/yc9xAg1YMTNp6BWPL1SqWZb4QAQ+xnOqmmtO2G9eIxoy4kN7ZRIp1s+YOYMPbM/4eD57Q72A8XQmUzrOMgcA5C4Zwx1FUzIHx
sOCJOBQ5n5lMRsRz4aaoXFXd5CNYh/HrgMuK54yfVF+9K9TdnpiPPkiRJalzfMq9KdPcH8oRvGiJL2sPUdU5xAA/sA3d+XR0wSRrEFxdBP6fhlfQR2EkBGdlwc3KIi7LVWU5eDZGLoPiRB0mXfTfqfixWo08xPCBMhOaCpHTxZRN4eelmUgb2/3N9ZdN9XvPB4HNs4bbyGyXbm1fXkmSiPpKJyAVyWc6cbA0jTKcSaARn7oq9K5mMW7TAOuFzT1PYZfaqtlnHhX9pTDZPUHXjajVkvyuzJmtSIKi
9Bb+5rMfONRXnM6cHVELKE6Av7FzPxaRjYgfwI+BdM/pJJEm
SpDnWaJj+EvC/gc8C4428IWxOYXQSOwYLPCEqBRBYuyHwHYRVsYr2l58fYEu2JkAVBj8fvIImXKkFeE67fXXthcXTbXEuHaYNEG1b9QkbBcxBe0wDnLd4tqmNTC6W5zcIbNWvTAeQqnaqUeIjinOrz4
Hmm0Vvj/VNE/HpEnB4Rp0x8NbUySZIkqc01OjJ9WfH1eHjen7v5CjcYpRYfHGlZqzQ56sdV5PFCHVfeRkOWKNRdDsSCeCDkiHmLrdMEjZFiNhf7bP6PCp1ysDj
zC5EkSZLmm4bCdES8p15/mfbKAZ4oZKQRBgbLgIDXxqfTiY6ZXv22aO3IcKEVnrTA4e3gJNVPv+3c6v+rVGtvFBA3A4PF5lONPvBLsK/5ZdVzk1x6/8TqMD
RePB/98UyqSJEmS5olGR6anGwSc9CbEK/f39R/SvX7+erq6ZPs1ZW1KSZ+PM0GrN3WLWgH0bZr1nqHOyD6npFdNR62X
9A2AJ8FLgi80qSs9taO8QWx7ewurdqyf7BwcGuvLNDUPPd4fBn1MCU12qwYc1R8Mzeurz5PiJX2rpcsN3vGlAErFgZJMb2mshGDzHEzQ3A
eZAE+pRA7rP6GblupWtLkOSJGnRa+ihLZn5L8CDwh4B4k0X1R7wS+DbwJ5mL6qwN0dIBmWRbg3b3BfjmHtyCb9tU2AvSgNy9pl5c3wFdQ0LzBhIfnWCvnUjd9kIXgwiTF3ePkR7xdDs/pIWb3mGbWZdyxgxWP
DeCdwZ0Rc0szCJEmSpHbX6DSP/wD4KjMfAi4ITpE4G3JkcYGtB23QVvj4j9ABqWW1UQkNwu5vTmBjUX4CSfZWDRJtxevJsmxmbTdVY0oXkqHeUvFeNCWuHi
K0kSZK0IDU6Mv31iPhPn4B+L9i8CNzSnJEmSpmXEmSJG6Ku6t+nqXG79Sr+mOswq7Q4mi+X8E4PfRv6Kcv8yK4y+5zPUqtEGptpW/MaBT7wt/4i9VFjlUwX
LkyRJktrZsUamPn8LkBmfgX+s+9mmVidEz4tM4943F7R19uUiiRJk
qR54lhh+uSjM4VIkiRJ882Z2xwvT2iPiV6Z0R8V6grzklSZIkSfPDcIPlKzOM1yrP
heQOwDPj5ZhYmSZIktbujhunMfAzsYGBFvAM4ruv9vZt7c9MokSZKkNtfQQ1sy8xbgliZXIkmSJM0L3v2jaSSSSpK0pIW
kmSJM0rPsVQkiRJKskwLUmSJJVkmJYkSZJKMkxLkiRJJRmmJUmSpJIM05IkSVJJhmlJkiSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSupsdQE6fpWxCv39/VP6
iRJWhwM0wvA0N4htj29hVe7Vk/2Dw4MdOWaGoeeOw4bQBSBvz5CiuZzjRTXmWuhjwPqfRBfsPlmmwIOr/Ffxz3ZfJPo3PRrTJOYMq3CVvb9y4UZx94xJR9qbEWrMO0AWtgm0wE9xndrFy3stVlSJI
kLSrOmZYkSZJKMkxLkiRJJRmmJUmSpJIM05IkSVJJhmlJkiSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSupsdQE6fpWxCv39/VP6
```

```
ZIkSc3nnGlJkiSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSjJMS5IkSSUZpiVJkqS
SmhamI+LMiLglIu6PiO9FxAeK/lMi4saIeKj4vqroj4j4dETsiIh7I+KVzapNkiRJmg3NH
JkeA34rM18GXAi8LyJeBlwB3JSZ64CbijbAm4F1xddm4DNNrE2SJEk6bk0L05n5aGbeXSw
fBB4A1gAXA1cXm10NvK1Yvhj4XFbdAZwcEac3qz5JkiTpeM3JnOmI6AUuAO4ETsvMR4tVe
4HTiuU1wO6alw0UfZIkSVJbanqYjogTgS8DH8zMp2vXZWYCOcP9bY6I7RGxfd++fbNYqSR
JkjQzTQ3TEbGUapC+JjO/UnQ/NjF9o/j+eNG/Bziz5uVri74dhcCBmukgkiRJUtvpbOK+Xwu8G+i
PiO8Wfb8HXAl8MSIuB3YB7yz6vVtW2T+B9z
apHkiRJmm0+AVGSJEkqyTAtSZIklWSYliRJkkoyTEuSJEklNfNuHmojlbEK/f39U/rWr19
PV1dXiyqSJEma/wzTi8TQ3iG2PLyF1btWA3Bg9wG2sX
Ley1WVIkiQtGIbpRaretA9w6ockSdJMGKYXqenTPsCpH5IkSTNlmF7EnPYhSZJ0fLw1niR
JklSSI9Oa5O3zJEmSZsYwrUnePk+SJGlmDNNNMjIyQl9f32S7v7+fSlZaWFFj5vM86sOHD
3PfffdN6TvvvPNYunRpiyqSJEkLnWG6Sfr6+ti0dRM9a3sA2H/3frrP6WYVq1pc2cJ1333
38eCll3LuCScA8ODQEFxzDRdccEGLK5MkSQuVYbqJetb2TI7yHho41OJqFodzTziBC049tdVR
HUZkiRpkfBuHpIkSVJJhmlJkiSpJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSjJMS5I
kSSUZpiVJkqSSDNOSJElSSYZpSZIkqSTDtCRJklSSYVqSJEkqyTAtSZIklWSYliRJkkoyT
EuSJEklGaYlSZKkkgzTkiRJUkmGaUmSJKkkw7QkSZJUkmFakiRJKskwLUmSJJVkmJYkSZJ
KMkxLkiRJJRmmJUmSpJIM05IkSVJJhmlJkiSpJMO0JEmSVJJhWpIkSSqps9UFqH1Vxir09
/cf0b9+/Xq6urpaUJEkSVJ7MUzrOQ3tHWLLw1tYvWv1ZN+B3QfYxjY2bNjQwsokSZLaxMkiSpPRi
mdVTdZ3Szct3KVpchSZLUlpwzLUmSJJVkmJYkSZJKalqYjoi/jYjHI+K/jYjHI+K+mr5TIuLGiHio+L6y6I+I+I+HRE7IiIeyPilc2qSZIKSZotzRyZ/jvgGgTdP6rgBuysxziCaszPwmcbV8A3gLsAIaATc2qS3PDJkK
qTFoJl38/il51h1UZ1tE3hfs2qRJEmSmqFlFyBKzZAJhwYHATg4NMN0e+7hmWeembMLIiV
J0uJimNaCkmMVdj25MS1VVdj25i6UjS/nB6DhX3nYleWs6Z1uSJDWFYVoLTkdXB53LO1kZJ1IZr
bS6JEmStEB5V15JkiSpJEem1RZa/fAXSZKkMgzTarMTarrKWIX+/v4j3h/VCuPwh3/VCuCPViSpnRimNemP8emP8Rle1Du7++nkl
MvFJwewh2pl
iRJ7cYwrZaYHpT3372f7nO6WcWqFlclSZLW/Rl5JEVJktTODNOakUZGlBtG5b6fdQ9vI7xZUmFjX6L4UxNt7WrJh8DhatEqvKxZf8QavqMiM1yX8gVuE2etCx3qR3VnPXIOBmvJ28vV7xvtbe9d7eOVk9efOO4NZmKhQi6QzMqAZM9xDTrt2/QjW1L/e4hsMbgx7OO
hjaO8TH/uNj9Nz17INd9t21jz8fN0xLkqTWMExrXollwWX/bze9y6ozlL61f5hcGi2uSpI
kLVaGac07vcs6Le7euo+tMQgLUmSWscLECVJkqSSDNOSJElSSYZpSZIkqSTnTM+CkZER0ZER+
vr6pvQdz2O2JUmSJUmSND8YpmdBX18fm7Zuomdtz2Tf8TxmW5IkSfODYXqW9Kzt8VHZkiRJi4x
SmzBMz5KetT2T7HDx7Tj9DvMmGWKwusbdmb7XYnPZswTKEqmnzfYTHoGGXOLHUUjvSnKUVUwTKvtTNxH2vAsSZKLanXf
zkCRJkkoyTEuSJEklGaYlSZKkkgzTkiRJUkmGaUmSJKkkw7QkSZJUkmFakiRJKskwLUmSJ
JVkmJYkSZJKMkxLkiRJJfk4cS1YY5kMPTpPEXtro0SZK0QBi
mtWANjFa49NqdnN0ZnHrjH3J3BPd89KO85CUvYf369XR1dbW6REmSEmSNM8ZpksYGRmhr69vB
t3f308lKy2sM+ld2kHL0dfOy2K8lbk22bt7Fx48ZWlydJkuY5w3QJfX19bNq6iZ61PQDsv3s/3ed0
6iZ61PQDsv3s/3ed0s4pVLa5Mz6Wjq4MlnUs48awTqYz6wUeSJM0Ow3RJPWt7Jp9yeGjgUIurWTCAQCHB
g61uJr5qzJWYXDPIMNPDMNh6BzuhGx1VZIkSY0xTGvTGvOVcYqk+F53/Z9vPu6RzhjtAJLYHv
uo9IZsNxTU5IktT8Ti+bc4J5B3vuvezm7q4PtuY+zOoMXLe2ATvjheKurkyRJapxhWxWrMuK
8nI4yMc3HUQgOEnhuk6aeqdM87u7OCcZR2GZ0mSNK8Zpjn4Xd7U8C8OCPhvn
rsedxcHU1XA89OtR286Kn36EF8PZ5kiTpmAzTmrEczymjzmjSPL3zaWJJADB6YJSzl3Rwb
nf19Hoogl+7/Qle8uAgALcPjpHjFVr1AM7KWIX+/v4pff39/Xz89o+z4swz4swVAAwODHr7PEm
SdEyG6UWikakXjRp1MHrwcMHHcmftN/fxwhOrTxb81v5hcmlMec3ZS2IyX
D8yMg6jx/HDHKehvUNSeXgLq3evnuybLq3evnuybuL3hxB1aJEmSGmGYXiSmT73Y/vbp1vu7+Z0uOa6eQAKb7Ky9fzvZ7iWNkNGRE4LySTb1AT1oU7hhlFtcUNv5B0JCdhIjlz6O0ZZc/ctHZ/S8svbZc8cvR0f7T5Kg4XcXJd7GUelyYhr
ZdeR5ZHicXZUKvUtrRqKXxDFe3XrdZ0wNztNvb1hv9Hp0tPoJYNmyZXXbE5weIknS4mGYX
```

qBqp2LAkVMvRjsPt6q0eeG5Rq9ZAavXra7bBji46yAfft2HOf/88yf7yoRr53BLkjQ/GKb
bWO39mA/uOkiOJ8P7n20D9Kzpqfva2qkYcOTUi7FMRn80Ormf6fue3u5Z00NHZ2vmOM+Ws
UyGHh2a8js92s9Vd/R6FVMf1lPTnujbcvOzIbzs3OvpT9ls5hxug7skSeW1VZiOiDcBnwK
WAJ/NzCtbXFJL1d6PedntT3L74Birnxmj94QlLLv9SR4ZrfDlX33pc75+YioGHDn1Ys94H
nFRYO2+a9u7b9nPP1zcywmnn8DwE8OMjo223d04GjEwWuHSa3eyNoElHPFzcbj6AeZ4PzR
MD+HHUi/M9vf3s3zN8sn91Jt2ArMTeqcH93qj67N1LEmSFpq2CdMRsQT4K+CngQHgroi4P
jPvb21lzTP9osBn9j1DjMRke+jRIc5eUr0fc1d3J4+MjHP6kphsw9iz+5p2h41GAu/0iwJ
r913b/uE4XHrtTl544tLJCw4rPUvn5VMKe5d28KJk8gExtT/Xzso4XzrpxFkP19NND8/T7
yQCz14QuYpVQP1pJ9ND7/HM4e5Z2zNlxL12dL3RY9XrWwhTXOrVA364kCRVtVMaejWwIzN
3AkTE54GLgXkVpitjFQb3DB4xlWD6eoCDjxzk1+/70eRFgd/aP8zzO+Ald1TbM7mFXL07b
Mxm4J24wHDigsOFYsrPNVKZEq53ZYXrz/kxVpy94ojXTZ+CA/Wn3DzXbfj+9LY/pbOr+rd
56oGnWLF+xVEviIT6005qQ28jc7jrBd7+/n4qOfVvWuZY0/saGeF+rlH52g8XszVSXu9Yj
XwoqPdhp8y0m0aPP/3nauTDxfRtGvm5yn74arcPO+2mzO+n7Ac2/xbzix/MZ26+nOPtFKb
XALtr2gPAT7aolmMaHBicXB5+fBiegQM9BxjaO8R/unYnp44lBOxbspcbX38Gw/uGoQdGH
x7lp2/9Ic/v7ODeg4epLA3GR6qPAcyxJJcw2R4/PM4joxU4NEbnOOx+ZoyRmvYjhyvsv3s
/w/uGOTx+mPHRccYzycMVcizZ+czY5OD0D0cqVJZA54GRuu3p+65t76Y6Bp7A2NAYPxyrE
DlWtz3TY03s+3DxmeF4jlXvZ6i376Mda82y6u8/D1fITPbfvZ9DP6gG26cfehp6oONgB8P
7h/mpb/6Q53cGS7oeZ+9YZcrfueNg9QPQUw88xRXfuIKeU58N2oO7B+k4qYNfePAgz+/sY
M/gGDf0wIGTDtQ9p+q1J/uOzPlTjDwxwu9f8/ucdMZJ1Z9hx9NwApPtib7lL1pOR3Qc17G
OdWyA4f3D/ME7/mAyGPf39/PHX/pjuld3H1HPiuKAjeynEc91rHq/j9q+6fXU7m8mGjl+v
Z9r+usa2abRn2v6No38XhupZzEr8/upd274t1h4yv6dF7N65/iX/vhLbfcMiMhsj8mvEXE
J8KbMfG/Rfjfwk5n5/mnbbQY2F81zgH+f00JhNbB/jo+p9ud5oXo8L1SP54Xq8bxof2dn5
qnTO9tpZHoPcGZNe23RN0VmbgW2zlVR00XE9szc0Krjqz15XqgezwvV43mhejwv5q92utf
ZXcC6iHhBRCwD3gVc3+KaJEmSpOfUNiPTmTkWEe8H/pnqrfH+NjO/1+KyJEmSpOfUNmEaI
DNvAG5odR3H0LIpJmprnheqx/NC9XheqB7PC9Xjedl6LCRJkuabdpozLUmSJM0phmlJkiSpIIO6ISfVQpOO0xGbOHr6UOT89RwI+qyV6/fUXBMzP2DHVGmi+d9nBo6WrtvMxBJkeqw==
fuzP1cCJt3V3ksnDe2+k5ShtPu8bMnoUuaoEpZ0x7g2WIr6RphB9RTmeVMTsakK9IwgVRyn1rmtHQKUnhvi3/tEm+YAD6a9GnuubjS6q2M8LypmMqD/H+bMRV7wY/yVrBzp1jqjBEoHa+8gj32/q/ngyswFu3Y2LrHcSAAAAAElFTkSuQmCC" width="100%" height="100%" />

```
lkOPF6sGwe+XCwfAIaBq4o53nM6z1uSZoMj05LUHCM1y+NUBy/GePZ9t/so21dq2hXKDXw
ccfyI6Ab+F3BJZp4P/M20OgZrlv+C6ij2+cCv1qn3aAK4uhipfkVmnpOZHy3WDWfmOEBmj
gGvBq6lOkL+9RkcQ5LagmFakubOI1RHbAEuOd6dFSPZAxHxNoCI6IqIE47ykolAvD8iTjx
GDSuBPcXyZTX9NwLvm2hExKpi8XBELC2WbwIuiYgfK7Y5JSLOnn6AooaVmXkD8JvAy49Sj
yS1JcO0JM2djwO/FhHfAVbP0j7fDfxGRNwL3A48/7k2zMynqI5G3wf8M3DXUfb7UeBLEdE
H7K/p/xiwqriA8R7gDUX/VuDeiLgmM+8HmM+8Hfh/4RlHXjcDpdY6xAvhasc1twIeO9cNKUruJz
Gx1DZIkSdK85Mi0JEmSVJJ385CkeSAi/opnb2834VOZua0V9UiSqpzmIUmSJJXkNA9JkiS
pJMO0JEmSVJJhWpIkSSrJMC1JkiSVZJiWJEmSSvr/nISDA7L8ty4AAAAASUVORK5CYII=\
n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "plt.figure(figsize=(12,6))\n",
        "sns.histplot(df[df['target'] == 0]['num_words'],
color='green')\n",
        "sns.histplot(df[df['target'] == 1]['num_words'],color='red')"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 406
        },
        "id": "GuVvWlcbnv-U",
        "outputId": "28d666c6-f6a6-4112-ecee-d2f69869cd14"
      },
      "execution_count": 25,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at
0x7f5f8ec93990>"
            ]
          },
          "metadata": {},
          "execution_count": 25
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 864x432 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAtMAAAF0CAYAAADhBbX6AAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3de3Cd9X3n8
ffXFpYUQZCJHQ8XOybcUkoJpGpKGyVDw7SbsJkl7aY03S6QmKx3dkknHTrd0m6n7cx212m
nLU06WboucUM66YWmyUJTJm2WkFtLLiIhIgSYOBRHdh3M1dgyEpbdOgyEpbdOd/84j8XhWLKOf9bRO
ZLerxmNnuf3XM5X1jPHn/Po9/x+kZliRiVJkqRChmlJkiSpkGF
akiRJKmSYliRJkgoZpiVJkqRCPZ0u4ESsW7cuN2/e3OkJtMzdd999T2bm+ub2R2mN
2/ezMjISKfLkCRJ0jIXEbtma7ebhyRJklTIMC1JkiRJUiHDtCRJklTIMC1JkiRJUiHDtCRJklTIMC1JkiRJUqGeThewk
k1MTDAyMnJU+9DQEH19fR2oSJIkScejrWE6Ih4FDgBHgOnMHIqIM4G/BjYDjwLXZOazEREAHwauAg4B783Mr7ezvk5r
AVwKHgHdl5tfbWV+nj
YyMsGX7FgY2Dsy0jY+Ns+2mhg0b4ZgzQBORZrSZIkdbe2hemIOBGTw4v+AHw6M18Dv
BZ4CLgJGsmMd58Wt2Qy0T9+fzR3heRF8R+s+Z7MU3N3LBbI1LqA4hwHJrlmEpt4sLm3VQOQ4cBUbnT8vnuIyyHDdUiCxEGwRhQ4bsV1hVWn1jmLgZE3k5m4DhE41WX6z2h9e8uG5TSQGvHvn+w5C3iHrYKRWm+Xnlv3Lv+bMm5NiV0rGUHtsFwV4cyPswOHi+K+APQ7Y2Qh3l5Ep5RpAjqGsq6Npj4d2XrJmDEm1tXbM7+fL7fOWL4ua96pP4D6iYnyaEtMhgsQB/FwE2oyUvpo0kvSlnEHJUAlTp74hnXqjsyrACatNEgOKSamsnp2YZPkPK2bGRKJWN2pfoVRJ1I2tVPTeNdEZ7IqG7Cxm5HJcgsnA/9u+8wMrmbcE/8pwU6UNFybkUgtaK8zX4ozvbP5xZP6Ghiay8/4uGYtaCqSg4MoT5A92Q4EehTN8QgjnCw6AFvkCX1/MkoaLdE7Cy5D8WtUo9ZLlXfz8nO2Xx
Ro6SahLEhtp3HZ8k9D0ncMBjUqLKA9BcI8hkIKyUC3VI7F2D7WgA8rvpCCzqMddQ5S+
QCObYAp13UcEdsP+z+gWsW3V66aJn9MnWv+MWwfl0DnnORIDv0/HOcDn80RPL6M2A2MRw
gNtbqZJWGRP+L7a/Phaq+2aCW6u9ogQS/dsKW+uMv2+7pPKsSo6v/T0UhB5KnbDLo/fDAd4Jy7kuZsQUCXbgG8b/qbCuPtD4
u4X2ns0l+x3MD6t7MUVFd5Nj+wVGmVoqdyjxoVNQJ0OOgtZ2T0bSdrHlWkZZWkTUOYNoHOrMAROzXOWJ6Zw3aKt0cA5GxTC5L2FA
rkh9yW7qxJuF3qBSUrCkkpFCYQtGE6kCvAqSiESpQJu9ukBBxMHtiSpXQA9xgRXp
SuNtAxTaPBKqOcZ2aC9+zIADWtm7akVtuEsYuvTBzUmKvI4YK/YYJsBnSmaSWs1TSmYKSmB/IF6PlWlI7k/lFIVjJ
nc23HQu8n0EQdeZdmmdrzkVn7FwcZtYwlzmGBBjMnTWrdvcMp7KuyTRmYx4QPYQIqkt9dpF3EXtIIwi6vzVk9wxLrB0XKrfKq
KPpUhDf1N4oRobz3GZVWxTRrKXIwQHeWHnaoQAE6DGhAmcFX9JJ5YY1N4nJ1CsCFYKHtuN1/cLjIjZIBvSi/IwPbtJlvXwcuPgyXbUiHTg
ZhD9+jYpOf6hzq4n+RJ0zoWPZd5XMzmZ31iM2gOTrp6iU5tDkbHYE4AmbSgYJIsPkB+Gin/MTtcrMphr5FbpvtknRbf8aLB
iFicOVkMJi/9m6ZIdJhnZRO29VVZTHXT4KtLjtaaqxYTtfaVspdVT0BgaWjBc/8qI8eDBtMu9V0+Kv/wU9RnL2P4EVGQkEbe
DUnZyzYQm5uUI7uyAmtQNbrJJpqIwpMjqbUJ75HiijESdsRpOaPlnTJGhadrwiu3ETlrVLMcjvWTyQPVvxx7Mu4uRWDeCfP9
c1tm0Nhn9KU8Jdfpvp0bF/s56DRrvWOl1DLPuzZXc2w/h3d94ah+mRx9PG0YRoDYSMhpR/7Dwmt8h+lgR0/0jG+BsdiZpKEE
Qu4Wr2qsQ+Aq7eV/c6iO4rlHwQqSY3ePr2rQ3wRiGdAL41nPA4qiBtKgOhfEOBmyP4xmVK4+tLXXQlHTSUTeNn6/kfSNL4xRnM
cVO3Es0MJH9zGqhP56I1qWmaWafI7RjuBu89X2hA9vLuhXsrT/SGVr+ti6RJ7wK8lLrlCtRf31gTNqjT3Rvc/a6WrLcNarXbZt2TVC
RWDt2fMPwNROiXC4iMnfumO+WRPBbvtWXJ7ZlUu3pSFxuIuD4QxYKx3PSGa1t/SPPGmb+ZHm2NuzTr5TPCfRiXG1tV4LsgXSoGmo
CLe0HdOFcsqkdaVcCSd0Lp1Y4XYTR9PhxZHfTPIELUvcXXh2AqJ8Bhz4V7Q0iNJSKsHpN/SqnXxEnEkf2sTwvWYRdHHkw9zwN
d92xlKAZyPsMytiZLW0bpurCp/3Xsui8h7BhXHQ2dyT0cwlQn3wqRZGn4vK0Ydpcv6IfHOH9zhXBM7EPrSHrOt7Vqr3TRx3xmanAj
sbSDEsBj1LvDRVHqk3WJc4hlNUUx5E3ZnRIwzJCnh0WdDr9Xv7+Ph99AX8R7A6G1W90j+36JJZPyy/7n/wR+5rvYtZaFZKEKAO3kc
5ShCjXdzkT0Bk1yIpi0rPQnasuQ9bLqhe9i6nHcB/PaNJ8kWTVMoj61wqM9yAIqqL9O5lM9PBaawEaKWD6vtSAqkuhsdavnCsdA
wsiqPw8Zu0nRu5cpPIagHPVH0GEZi3Rv8I5PUXUd/DjkIcvdhHZB7VpiVbp6Zjn/BJ4kZFvL89f
UcLUbDy5NEFcyRbYXZFjpjZ9Xu5I6a4OqnRf6ZI4ns8Ejx13+ahRPx8SCk+Zt60ZBKYtQ4N
AS0WFbM6Gy2SUUAwhQB9MUpCSqaSggxjiOP+pFYBoI5PqWp3GjlM6xd9iMb1tMMmS2A4VH
JjrFAxkPbgvtMEg/hUz2vuBdUYzqcAG04ILL42ts1MF+fz5/uoGGpIH+oJnCqaadRX/CL
CrLB8nmQJ9ZjXSSxe7k4nkP3mJ3gZj1sbzpLHRUNq2ZNb3PW3iXLMufG0qEgMo9Txl+XbJ
e9YB/kaRkhIt6E/UyxnBWVTdmcz5L1VaUxgQuc5/MoqjHSi3rhd/ZgqyZB03O0fpFfsNzX
sVMpJ9U2XpQjc1JU5/8iS+qHapz0tWi+ZoxmqcGx6RrVKKa7jRDRUu/a4ISM4PkK1RS7k1
7Stsl/CcvgtKLbzXU2z6DBNSBtw3g2d1oHCnYcsnx3g7YNhDsoAx96R8m3rRG1JyC5vHb6j
yCHQNFuK35sR48qBF3O0gTXs5U8jLupVJ1KOYi+r1mWV4wwStgV2q/Y0K/u2qM0JeD20ku
PTDsKTakqEpFROkuNUHY4YRiljAxn0IMpUNDZFlUrHdn4D7lk/4AfmpfK8E/WIa5aPfN0FP
lhpw4Y1/7SbkRG9O/JZaXzEHy4Hr8gMcc4bM6eySk8kRk76tfzlOe4KDd5ZBc/r7NvzoI16
Y+23pLnpXZc/WUfpWcOd9b19l+2bNcHqw3Zm1VEM4xcxOczYRLsYwPpLprpKSYNVZzGwwk
6nN9mznnxDZ6WHO+YlLnBeu0fk0I2xYF5X7t0FZ2gKvU7Th7WbE2sEX7ad4fxW1i2a8xz
gt1AHeCt3DyeO6veWvyaaRjbLsYtyz1Ryyjs+yM1fh08eOZS3OXL9tTJEA0l3LqDjBd+x3
sqvLkuZ2mJtSCkW4F0BtCKiRpNo+1R8p9mf7iZmp5NQMjyQ+7Sz5l7ffF40NcmGVNoEoO
dvWp+jCevd1ozmNy9xxPG2MLqtPL4Lp/1wwjnVu0PZNMrWT3Ww22+GCvBvrZrWY4Qbsx7
HG8Pqi+zycQw3o/TWeOzbWsDDFjmP5z/4pgJVqu2qYa5uHyU33+hJHOKMqOFaFqI1TPN
PqQqoJg+jXDPL/lQRtqBWZrgjZm/6Ee0ZeJfr5wmu2ZhCAJl+Td+OsIGwAa01Z91HZ0Gj
W7ev2/ok4Dcr+N/wOQQdKzd6DYaaZ+YxXMlmWUcGRLIOXgmVl8rBFZ8oZEzMQAsoe7CzG
XGWNvtWSj8IMt2AMxyW0ob6uVdcC3xdYGHzPsQFEyPdnEPcKTT7TFJpx1ztHGwfDetG6
J87l6HQD1d2q5fB4ycwIqp9WJ8PwmMmuGJdpCYKk3AWPXNdqL+wXKtcs0Yhdf37KfbFS4u
WBYG1Af6S5WDnedJAsMuRapMQkV4B7HkyCw1fHNFPsS5qoyM6PUx05Kbi+fsLbQzvmdn
RPRtA1oRUJb8JP/LEFmTGRl3iknRzQXCCNqFXIH3oQhk+ANu/b4dIdQxoLdhUnMuhkS+
M/dg1BqdVOuk3mhm7/Q9xq1p6Ig4EZu1Zk4D7lmNv33bpr1fUeyn66t1WcJRhQz7wm4M
WDlbo1zJ8ch8pdGDxXe2F/9+mFvSKpTuUtrDr2NpiVTVSE5uwK2cB9kAR5BHbHCePsu3U6
1r9bwJuQpgv/gTs/JGn61Kqr1nTSM06+F77oPtvyt+x+JQSOGeVeo8Vx8MX9Z2dPz/f1/J
/d+W3tzN1c6b8z9vFqQr5Rad9ibOHCK8xn6Ecy25hrcPkdVXhqrRyWbP4uVDtX6FHR4QLm
EkJvPYj+gbrgIi55aBiWQFaaGtC5QWh8CldP+P1N8q7k4+ev47LaP2dBvuWx91ek5l+E
bWQMu/ZjzOaNzs8f06a+Ie6a0zaMzzNXyp1YMq9EnuPxfWXgcA06fGpm4Ze8mrLGpUdpB
x+rXvbb0KU/sf3dN5KPPG9D9mxqevfNv75V+NZpufs9Zzzp+VKrRdsKv5n0aWeyTaW+iUk
0Zn39jRc3nfT84e9pbR91ywGNuWNwnT2VuV8r2OH8YvBg9w+dDsK9FWFZnKo7ukkVHD+nn
mOXO7TrUe/9MxejtGcGDuiu6ydfSVhlZMVLt24QsDn12bX8/ip0wBbhIfnx29h3fVYZ68f
/IP7H6jOn64X4s1GU9TuQwVQdBDcz8UDEkWN8GuFNRR+tcYaV6+d0mcjD/PHgv+7Z8xmkh
Ck90CebHVQjWzWFYQ9kqEvW5kTRLSzWe7MfcyF7NOSUc9ugamFq9jsaksAZsRAXiGnJAek
vaTrxUjFQIYoFFm6tqzcc8Oib05Fs9jFTBlOKA8rhqwjcxgTcsumgnV3NJZ5eJyrJfDe9u
R30b6vqS6w5k8FCRPsBmEHmbb/zUzCH8/xzz9g4UWZu1kc1DpUqPqAThv8W7M0AYoXVgAj
YQtkxEmn4R9Ms6y7IRO6ZuwbVV4AFNRgqr+ayeD4gJrHy2ctPPjLhVgsQvkA6vTy3zVnX
B3SSkXqKGTKnNXpaU/g9i/kFZB/jM3+NmzAHCJazbDmvrj24I0jsT3T6mWeUWQPM2H3JW
3S3YvRhWJkiYXYzP1e4Q4GhKdvSn7b8FGdJiiR3XOAp2yr4ZmgtaIwYfVSG4pEQSZ+Zh
uC9qEtjFpzxgSMHIRk81KYMCzg3BeO1jU1uW/xO1xF7EpzeQlOVuKjhmL+qpyYqKtyQSEH
n9WoWCy+FUTvYNH9SEHvxaxEY0ZRQW6oF4cWhZ+6VWJvzbE1uuWvXUTYO3VuTJ+BdNX1/
9nNdmmp71YyPR5U+LqEnEPtWT7nLj8W0rMwZ4a4kJG1W2d1rGdXM7tgqY8NQqa25rs8RVJVHz
A5/YM3kjdH9iNJBBh/D9m9zx4KaNw/ERwJPjojMJqe1e2KV0w7VRDMJYxc/qLuW9m3ukr0
dm3uoTCcgE9gL4Fz5GF3bNYWKUo8ZpHr73m2GsJoy+4Pcs1sZuxPNrmc+GRkmqkQczSgs
q2Rqw8Jw2ClMnqmeeGPjqdSeDBl0S+D1smmmRaKtPZ0BZ6OZZbP0T+Okvu4jUxsw9Mfwl
ScB3y+Ahcxaqwe8tjK83kRS+/KvX42uwQo21lwVOSkGAM2z1NCRVm/kWxF5m82H2sY2Uw4
PZLJ4YfyNHPZ6FKSeD9SxwtWOxUgmE2SkpL+Y1ynIbRF08M/3MZDQoWW6u/PScM3nOHZ+k
kmhDqtrvfDL86BI3Ko9qFC5QZ4TU1sj5NvQqhS1mm9RRzVpBBEvNAq9PY18cM9i4Oy6A3C
N4gjmFVM9vmmzLc2dhy1SjaBBdnQANvbjrNM+cAYJAG5mWxKWsekU4BYNA8Xm/IIhkt7IE
dUjSCKAfVMtoV+QRoHoNUhILLdRtBHCt7x6iCF3rRzQaJjrXgn7lvjGVh7jC6ZAdbs2UfSQ
D5gnb4y8fsJzcLNUEuAFdR58WU9v1O5gyrLNbBqOVJhF/sM06c5uSLpkX+rs3QS5/lpT+qjj
FxKXpl0N4oFy4A5DJHz+4LpDPTn+SrXpkR3a8wSUSlRFXI4xHWWnwZKNsDN2tCTbsrtzeNf
GWvLPIHUwmd7n0bCFjx7n9i00Ui5jVVqPt2LtR8YHbmY8jtpYDsltntyTwW/ShDRlHi7Wv4
+JOd04DKeAMgUmm3Fao5yC43Rm6PZ48NnVKdS4EXSLU/KFP8ZKhJW8kI4nsrWcRcbX+a6z
3u0cbZuwqiTjxy4o7hdN8b7CGBpfkK3dL7YfF3aHJoSrGpuIytb7BHnGG8MD1GLqpdfvxFY
R0+K3Z6JgeoTLH5IHDX6+bSoPBMkHWcSVG3x0frzyp4IjbiC01vM8kkRvzTdyXn89xXwmPP
tX2npo+vmHX+CVKcn3KyJaLhSkkPgLptY3zB7Z7v4cUufoS1WGQBe2t4ipw9RCz8Z8c1Yq4q
ji0zjn/GPrEpl/w+81RHvz0T5i1E8OVFbfIzKHo9ZzqtrHqt36mAvrC5Ji5xMnTY5LrbVjk
PX2RcbbOX5L8bpWKZ13GJ5s1LPsqtSUhmLrGaQMhHqlr3npKcBKjJv/LTfg3R5hJxuiPT6M
kQlJJeD4kw7WOf4rAlfhsq1GS0lLSaxhWy3SU5ifX5v3pCfjRb/t2qpVhPpUsAKgaOCHUWR
q/Xeq+RBlzjbg8A6aSGvzA6rbzTAYrtaWjB22CTTNYJOJZR1YVeXpfoXcDmjHq/Vl0VVVNz
Fa8q/ZTsT/hFsy4mQJrkqgK7G95yvOkIGV+o9fGLYKHL+xFqb1ESiMRg+M2pW58AmMMt5m
Bqdvpd9Fm/1ZRDt8iddXK5sP4QTe4ZyMnSuuH2aDfGgdsGMJi3l/o4+11k3mp1y9XV5dWLC
4e0U9Fo69N6bmNfcz3qoUE+pDGFzOZRKwXwIoKn8Jib4OLk2q7lpM6yQ+r0uG1smK+yu7e
rsi2GR6yuk7R6UO0G+mxOQGWQp6ybT7fWVDqbqlurn9Ktg9YRRuL2Xo5gkjrVC1ZwU8uiNA
BOaJ9C3mp3ZXtb4zx13PbXGr5OvJ0Eq97IoXyz04ACdxPpnGEVxnGvrK8tAT1uPZ6xfT0Wn
mhuVPFj4d9wUjxLoMhcmHGrJfQmkYc6eMpM5XYq00gxIpIB0NyrB1GX24vBnqr8Ng9wZmh
ucXr2OW1OLJL/01yhytFKPmf7UmUcXYOdsNsnD/eZHoYZQyvC1Ib5b+NMMzLtDMDFCZh2J
JO7D6R2gRb7wL2r3ZJHwPu4XJMq6FyZfzwMqEf0e8SCC6uBRNDdU8p7eSZm5tvdrMvkfFr
KDdBcLVunWMvCoqi1nPAa3qrQdj5D9dXVxM3AnpI4JnO5wQ31Y1fNuvmt0vdjyGLhEtyY
sUFBmNaJ10b1KmV3VLSBxYf11rrxWLE68Jt33GPEaPOtWwhLtpqoeIwi0TUQs9XlfwwNG
rT0fpHrIYOeXRLXTvWzNTdlPGeUIYEPJP/j/oD2Jt0aNvP1n1AU4cNyOrNs+nAaNV1cK0
OzMRL4nyvdY2h8BxkufzpPhHbPz8qUJYfHJAkMEzG7ZLp6vrfZ5+P2t5yn3pwZ7IX/XV1FP
Qz3Mey9AWF6bQWwUNsmSH5DLemN9N3Pnqe/SK8TVkN7gJwXUhkBgOydhYB2P2G1DcTZ1N7
UA+DNXD9QhvS9+J8fYN/8WjbjGU7CkUJNhsZKDgh6R6+T1iRlDZTjb89wHeDywpYnT5+S4g
O6Tma1TMbp3hq8R4uz+y5DN4AYh4+PnE5I8QsANxbcyPBYy2DLX0Zv+vXhR03zeCRk68qh
hSdR+jqe0Vd2mKz8uNfYGEgFmZdsrq9Ix+kUw1HlmVTYhoj8OOvyE6MBexgRVfO7jl5f/K
CF0yhDO/Hl67A4qE3Tg5gTpH4JEvP2SAr+7BJIzrcNG+JaX4Zzqy/f6RVfE0bYz/bD4l+dvr
tjZTdUBIjRcMe6QgE5mtupGEJCk2oPnWHD1iV/bL38Vgh/MMH+pr0gZv/9bwJG5KoI80G1L
ujNWeYXLUmzoHRrH4j0lVBXSXkOGS6JMOnX9B1eJn8ypT5lh6mo7LhxyR2yQcY6E9+ZJtYj4V
jXw+qsTHKx6jvUIC1swo9yr01aZrkBY1ZY69ZrQtgUcWJ1lBu/9o9p0nFGZ/SuIOdC+ZS9
eqwPOrmpyQBkD9qVCDtfhp4oFZlHXXxxaXPo6mmuhG5x8wT6m17uOB0qfpH7ZSI/3lhO9
AuWB7oE/4xW4mWRzYQYrEm5dD+dEbiPjOpqyCZZfjcB4KEtDOd1npsTm0T6TnwCnKKQaDV
SdWUqH8Wkez+8a/2h6CJ+UtZ1c0eJJJiMbH8mu2AX39KuLgyHT4XqrZ6sQcg7qQbNuCLRpK
lV+MGaGGmkHZ+kzE6DTIxOmKIPXbsEfbn7dQpmRHTP6l2zXTBq5iEANERz5SdTQXNPuDfH
ZEvbXM2cSZkLaOMH03Bb7EiANFgoCfZgn9+j9Mmo9pRNmVdmIZJyQXyoVNBJ2ZC9mXG6tpf
J2bX8KGCZqYuxH4dxEGj5ifgKnhkR3JrowiCvpCJ8HcbqiJ/ZP/Oe4Qfx+Y1WMxnNa8tdq+e
yhcgJW/FeaQCEzejvr+9C7MEu+eQeTMFC4xcwJZVzALazrr0RlW1GqRZ4izHdy7uo3rJ8WQRS
```

```
qcy85a2ViJJkiQtMa0Ojfd3EfFfI+L0iDjtyFdbK5MkSZK6XKt3pq+rvv9KQ1sCr17YciR
JkqSlo6UwnZlnt7sQSZIkaalpKUxHxLWztWfmRxe2HEmSJGnpaLWbx480LPcBVwBfBwzTk
iRJWrFa7ebxi43rETEI/FVbKpIkSZKWiFZH82g2DtiPWpIkSStaq32m/4766B0Aq4EfAG5
vV1HLVW2qxujo6Mz66OgoWctjHCFJkqRu1mqf6d9vWJ4CdmXm7jbUs6wd2nuIbY9uY92ed
QA8ed+T9F/Qz1rWdrgySZIklWi1z/TnI2IDLz6I+J32lbS89Z/Rz+D5gwAcHDvY4WokSZJ
0IlrqMx0RVwNfBX4WuBr4SkS8o52FSZIkSd2u1W4e/x34kczcBxAR64H/B3x8rgMiog/4A
tBbvc7HM/O3IuJs6iAK4D7gmM1+IiF7qQ+39MPAU8HOZ+VjRTyVJkiQtglZH81h1JEh
Xnmrh2EngzZn5WuAS4C0RcRnwu8DNmXku8AxwfbX/9cAzVfvN1X6SJElS12o1TH86Iv4hI
t4VEe8C/h6461gHZN2RTsEnVV8JvJkX72jfBry9Wr6qWqWqfafkVERIv1SZIkSYvumN08IuJ
cYENm/kpE/AwwXG26F/jYfCePiNXUu3KcC3wI+C7wbGZOVbvsBs6sls8ExgAycyoi9lPvC
vJk0zm3AlsBNm3aNF8JkiRJUtvMd2f6j4DnADLzLzE5l5Y2beCHyy2nZMmTmdmZcAZwGvB15
zgvWSmdszcygzh9avX3+ip5MkSZKKzRemN2TmA82NVdvmVl8kM58F7gZ+AFgTDBiMiCN3xM8C9
lTLe4CNANX2U6n3zZYkSZK60nxhevAY2/qPdWBErI+IwWq5H/hJ4CHqofrIsHrXAXdUy3d
W61TbP5uZTg8oSZKkrjVfmB6JiP/U3BgR76HeF/pYTgfuiYhR4GvAZzLzU8CvAjdGxE7qf
aI/XO3/YeAVVfuNwE2t/xiSJEnS4ptvnOlfAj4ZxP/Ai+F5CFgFgD/PSxDszMUeDSWdofpd5
/url9gvqkMJIkSdKScMwwnZmPAz8eET8BXFg1/31mfraVF8jMUeCSN1ERv8F1SlfJt3+b27/2n7q6o7YEwfVtkrcZF/X1+5V4ZBfY2rT8dR4dE4TzKtL/h5ZECUIsq1cQ8J3+bY39ZR9DvSp3l4WzZkY06Yh5N8/I2fmYnMf0AfqbLjeyU/fKIKT8+YZ09LXR9hWU0Zvx7RWEY0DNw1OXpPRTAqtIZbsRcV+ZUaG4e0QvCqKY8yOO9D7ZeN6Qt6RgIJTW3oCHgMk8D7i3DtVD1p7WlXt7ptJWUoC63M0Z7nU8GvfcVJrUsOv3DjMA9uFZ3PnmNZfpd2PDyN9jpdvN5M2Ul5ZTVQ2WnJ5UlztfxEaqkHmJIZT0UeaMeQEjLQA9OEaZ0qY2ZmDhBJjPo+wvNdZhuKK0hQYXzCtH0/pH1nZcvS0YnRlhzQHOjdjXTn3ZTsbhj+dhZtZ1bbHq9Yid/cvSGc1O9KkJkpbbKKGnj7iM2I2m39KdK0M5X3Rpp/fFjIFPmSJNUbkTtJMVu5HHFqrtxSVaMjfa9B5J5iH8wFe4yZ7RXUULcj6pVNTzJWGJdiAb1ZSG6BMgwDu9BYk9QJL/6dzMtmOQbzrZcd3HNPSYGZXajfJkb1ghT/nppKRvydVHY/F/P8eLjNaWRJPuVaxZbJQDjYQzEc8AgDjyPK5yG+ECDKAhwSKbzV2sF3C20CxajGsBjgaEzTfhDXW46gZhQ1dz8cncYbdyb53U4Tb5sJGBDpN0fuAnVtR0Mkra93lt5txUlmcnk3NgBpbw0Bn1s4MQIiPVNKO1S+Iqla0vNC6dRT7lc8eTWSnYRoNZu0LfZQZGBGZgA1AYx3ylgt39Sw4zoxR6lPOK6gc2P4pEMWq0DtwC2LiL5HjbsMfcmMOUJ6n1VB4AkaS6GoKr7MZpYgDStR3GD8n73XvqWbZT84Yh7QhK4HdQl9/lAl5jKuA6iZiYoPr3nCY3cV/4eBxjzRR7aRmfbfvMJZCGkZjq4k2kRWFS0SCMmFGhiUB/DMxE7bZx3vcVxSdQdSJxR06shf4OjyrxVZQjV0aPSaN9YYoEx0cdkx0SwgWSdmlPTVbO4x1gyIv8+TKN3kEm7kL8htvQ0L6j2P0OXRGJbTEyuXRexaIIaQIbJz5zqdeL1cpcJ2xzdJE7BSXQ1HG7aQm3UftoyhbyA1eVWGafH1Jxw5Dz+FuyBjJoKYkPYt4UzU9PaUGbwA3ytbhQvETHMXsjrKLUw4pcEp4bIrV+jq5sj3Dz9P1xN9u5ySmr+XrsI4eS+VBqNp9HFEYuXW5Y1jbT/DtLbmoM+1hoqCZg3ai2n0sX3hsxZfcp6NEIM/UFZqz0HwHaBMLd9NAbXWOoXf0nKbCBmWixzCm5Kj5R2tuJvzPhPFl4P9EZz7QhPYy7MP6nbfBL1GOZ7zE7zZhkNSbzTxfMvPKqPS2rbyrzY/Yd+S+SqBwWhG+ogZwXyKEhPAfWx1uDNKt5kcDYnDnOvDjHJDcLrsvqF2l6tnHZQ0r3qO35O2fN3DuIUmHQyB3mvtkX9ESRkbn6v4r9y6pC1N5WM0g2kK5Ne6p4R90TC0s6jx+UT42O1CFtfi26jrXOwtXcp4H6wHjUNAOjc3vb6qqR4DmUCDzmmqHBrtsPuL/vNhbbZJiIKh0JU20VXuSXHN8SBlGQTF/kxQqkDY7KdH4F7ES6wN/e2bYvkyDlHM+X3IqEM6NxEbUFftFwOxX3QJQGOWpgZQXvq5EHgW6WwfNBYfrVYTdBIP1uREXNHa1xBkAbUnKL+pSv7bE4gp1V/9EBrp3QBRuqU3DK4T+c3ni9sHxoJj/FSjmc7mqDO9sePkcyZtBUyP5c5fGfSUz9y6V2ooynYGXMIVKzL7yx7LjJp9rs9SndgIbZoqfyTV4Qu8fj6xXDYdxzcMV0jYWqjPqGYGSKCtnJE9xcb2C2Ehc1n4F7PfRBZB7P1G0Nt8ojqg9IbiwBkwlTSsFMH+6LUvGR0GJ5wy+rkm1+D3nXrcKA3tu/8YdR1kCKmy7qrJDxt7aFgJG5HVvD7bEKqhXv4h2E8HXY6W0z1SFU2Hl/IcFfi9Ll7Rwzpn7XEEENXC1+OvXcn0jC2Kx+bLYuPFWaKpyDpmUTqz8V4M9iZJYi0l/WXZSvQBQkdeYWiPc9LMBH1JtC7dRv4SdqkFz8j6+zlH+GZX8mbfmZuJAkWgCx3Lf6SvEXbDGzMw8i5xEwGHLGz1vMCtfaM+S3lAUeXvs9UJmZs/sSFwV/hZbXaERxL/nRS46VzRNbR3YtgYN2PJbjuTeb0vA0XQrWVL68GMQGDnaqvK4s0j+pNsc7ebSMiRvuTwBprQMLRijNGiKZTBjnCQuDAFPNUNW5/5VqXcuqFnH1R8BH4NaJlIfxBSR3Pc0OdBzL6OrPqkAhYwuC3jdHi03zcVD7XW1TQbQ4ZWWHUhbjXRWpubbJHtpLk3Gg3xrbG1L+PvOrzTEKZLJfoOgJaxUhWqHOvlwQpbWSWvAKNMG2iHLzGmVm/UjIrkUuSpRBxTmqYFA8d==
wc7WM2Jae7D3MoHg3N6V3OhIViSJK0whmnNqv+M/pkPB0v5g4EkSVI7LfY405IkSdKy4Z1
pLVmzPfgJjlEtSZIWj2FaS8JcI6bcfO/NnLzp5Jk2x6iWJEmLyTCtJeFYI6Y0PvgpSZK0m
AzTWjKW04gpkiRpefABREmSJKmQYVqSJEkqZJiWJEmSChmmJUmSpEI+gKiFk8n4wYMciOD
Rb36T6elpspadrkqSJKltDNNaMNMv1Ng18T3GVq3iD+79A57//vP0X9DPWtZ2ujRJkqS2M
ExrQa1as4rVPas5eePJ3pWWJEnLnn2mJUmSpEKGaUmSJKmQ3Ty0rE1MTDAyMnJU+9DQEH1
9fR2oSJIkLSeGaS24w5mM7x5n4okJmID9O/czvnscssU+1E2jgkxNTTE6OlrUB3tkZIQt2
7cwsHFgpm18bJwd7GB4ePi4zydJktTIMK3jUjtc48CuA8cMyrteqPGOO3dxdgI9cNJXn+X
zBw4zvaa1XkXNo4IMfG+AJ+97snhkkIGNAwyeP3jcx0mSJM3HMK3jcmDXAd5268NsmqwdM
yifs2YVF1Zhure/h52T08f1Oo2jgpx67qkcHDtIbarG/p37Z/aZeGKC3lN6F+LHkiRJKmK
Y1nE7p3c1F5xAUIYXu4JAPRRPvjA5bzeQyScmedsnHuac3tUAPPL0JH9+yobjfm1JkqSFY
phWRxzpCnL+KSdx+LnDfPHwFNMnr4GXHfu4c3pXc3F//bI93HN4ESqVJEmam2FaHXPOmlV
c3N/D5MQ0301HaZQkSUuPCUaSJEkqZJiWJEmSCtnNQ8tKbarG6OjozHrp+NSSJEmtMExrW
Tm09xDbHt3Guj3rAE5ofGpJkqT5GKbVtUqGzwPoP6N/ZpKWg2MH21qjJEla2QzTBSYmJhg
ZGZlZX05dCWqHay+Z3RAgp5MgFr2W0uHzJEmSFothusDIyAhbtm9hYOMAsLy6EhzYdYBr/
ulxXt23ipO++izfnZzmzy55Bf3r+jtSTzuGz2vuV33E0NAQfX19C/IakiRpZTBMFxrYOLB
suxK8umcVF61ZRW//8rw8mvtVA4yPjbODHQwPD3ewMkmStNQsz7SkFeFwJpNPTc50R4F63
+reU3rnPbaxXzXMfrfaO9WSJGk+hullIqeTib0TRcfnFyqfredLLLq8/wA999fqbtkacn+fN
TNhz3uZrvVnunWpIktcIwvUxMPj3JQd07ujd+e6d3NAd5XDP4eJzNd+tl
iRJmo9hehl59Umrkh4ksZ7t7Pd9IHYczmXx2EvbWR/cY3z3e0vB0S50PKUqSpFYYppeo
tJaF48tn5x2Buvnd6kFFn7Yw9IybXhggadtdjmXx8Om+ZfTqwjlRRsQS7qs8SsRAFp2Rq1sQ
jZ5z1SopoeBx89Yzo0.rutYfQCTPs8GlyXj6eAb/eEpp9nQqLw1HJmQL8Hk+u+zerVRwWFe+7
lUcP9Z1QGGlkbpIhTNjQTDUrQW6P5vOzXMv8d0R4M6fePd4JlMX1O8g2O8a2fXSwDG0LVFp
jm5dldXkB0OK+q4/Q8/1eqVzuDpSL4aZFthkqyfi+O9DiXzNLc2jh0OUFEDqiSVjF+R1zF2
S3v6nmTzqK9I9jJ2FeX9pd8VGPc2JnCPWNjsHXPN7zyaOPI9ojxEZEoRm6DuZTpNBGJ9O3
```

```
pajbhIRJwN/C/xSZj7XuM33lsVlmJ5fS9Oca+XKzD3V933AJ6n/ufXxI39Gq77v61yF6jJ
zXRu+1+glMvPxzJzOzBrwp7zYlcNrZYWLiJOoB+mPZeYnqmbfWzrEMD0/pznXnCJiICJOO
bIM/BTwLerXyHXVbtcBd3SmQnWhua6NO4FrqyfvLwP2N/zJVitQU7/Wn6b+3gL1a+WdEdE
bEWdTf7Dsq4tdnzojIgL4MPBQZv5hwybfWzqkq2ZA7EZOc655bAA+WX9vo94sALMAAAMUS
URBVAf4i8z8dER8Dbg9Iq4HdgFXd7BGdUhE/CVwObAuInYDvwW8n9mvjbuAK6k/THYIePe
iF6yOmeNauTwiLqH+5/rHgP8MkJkPRsTtwLepj+xwQ2ZOd6JudcQbgGuAByLi/qrt1/G9p
WOcAVGSJEkqZDcPSZIkqZBhWpIkSSpkmJYkSZIKGaYlSZKkQoZpSZIkqZBhWpIkSSpkmJY
knbCIONjpGiSpEwzTkqTjEhFO+CVJFcO0JHVYRGyOiIci4k8j4k0I6I+Iz0XEULXPu
oh4rFp+V0T834j4TEQ8FhHvjYgbI+IbEfHliDhtjd5ZUTcVy2/NiIyIjZV69+NiJdVtXw
2IkYj4u6G7R+JiD+JiK8AvxcRZ0fEvRHxQET8TsNrnB4RX4iI+yPiWxHxvb+60lSZxmmJ
ak7nAd8KDN/EHgW+Pfz7H8R8DPAjwD/EziUmZcC9wLXznZAZu4D+iLi5cAbgRHgjRHxKmB
fZh4C/hi4LTMvBj4GfLDhFGcBP56ZNwIfAG7JzB8C9jbs8x+Af8jMS4DXAvcjScuYYVqSu
sO/ZOaR4HkfsHme/e/JzAOZ+QSwH/i7qv2BeY79Z+ANwJuA/1V9fyPwF/US3/OTD
ccOzfZOZ0tfwG4C8b9jvia8C7I+K3gR/KzAPz/BySSStKQZpiWpO0w2LE8DPcAUL75P9x1j/
1rDeq06di5foB6eXwXcQf3u8TAvhuljGW9az+YdMvML1AP6HuAjETHrXXJWi4M05LUvR4
DfrhafscCnfOLwH8EvpOZNeBp4ErgS9X2fwbeWS3/AnOH9H9q2q+AqsvI45n5p8CtwW
G5J6kqGaUnqXr8P/JeI+AawbiFOmJmPUH9H9DjXXUQ/SzmflMtf6L1LtpjALXAO+b41TvA26
IiAeAMxvaLwe+Wdx8c9T7VkvSshWZR/2VTpIkSVILvDMtSZIkFXLgfUlahiLijcCdTpIkVZX
cz8s07UI0nLld08JEmSpEJ285AkSZIKGaYlSZKkQoZpSZIkqZBhWpIkSSpkmJYkSZIK/X9
TLpak0YV6AAAAAABJRU5ErkJggg==\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sns.pairplot(df, hue='target')"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 567
        },
        "id": "N-wcJafQnxcu",
        "outputId": "1a245af7-34bc-45cd-eab6-544a1bb4574d"
      },
      "execution_count": 26,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<seaborn.axisgrid.PairGrid at 0x7f5f8e8f27d0>"
            ]
          },
          "metadata": {},
          "execution_count": 26
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 582.375x540 with 12 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAkQAAAIVCAYAAAAwMuAIAAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzdd5ibxbX48
e+ot+2977r3jistYMD0mkAILZBAEgIXkptA2iXtl0LuDem5IZBAEnqvl9DBNOPe29pee3v
vklZtfn9Ili3vYltm5S06n+fZZ1ej99WO7Helo5kzZ5TWGiGEEEKIZGYY6g4IIYQQQgw1C
YiEEEIIkfQkIBJCCCFE0pOASghhBBJTwIiIYQQQiQ9CYiEEEIIkfRGfEC0bNkyDciXfCX
q67Dk+pOvBH8dllx/8pXgr6Qy4gOilpaWoe6CSGJy/YmhJNefEINnxAdEQgghhBDHSgIiI
yCEEEOpvsNDSGsK0uwYDGqou5m5NQ9R0eglpTmATPVYh4SUAkhEhKHW4fT62p4p4Tev7cQfCnH
jyWO5akEpEUpuam2oe7aoOv0+HhmTS2/fm0HfYEQXz6pzC6pqsXlZM3Cp+rEMdKpsyEEElpxZ42f
vLiVrr7Anj9IX73xk7e2NY41N1KiJV72vnhC1vo8gboC4T4w1u7eG3z6NYvrr3TVPrz+0fdchThWSR0QhUKa8jtfwu0LDHVXBD
bgcV7+2SQUpmA2j72VxTLazX9vEvBTMxtH3XIU4Vkn919Dp8QOwfKcsXRUi2SydkkduqjV
6O9Vm4sr5ZSg1+pKNPzMph4K0A/lCLquJaxeXY5TEaiGikjqput3tA+C1zY2cNTV/iHsjh
DieJuSl8PiNi9hS30UopJlUkMK43JSh7lZCjMtN4dEbF7KlvotAUDMpP4XxzuQpxrJI
8IAqPEO1t6x3ingghhkJ5tpPyAaaTRqOyLCdlWcnxXIU4Fkk9Zdbh9pHhMNPlkRwiIYQQI
pkldUDU7vaTn4aRl2qjyysBkRCCJHMkjogauv1ke2y0ucPEQi
Ghro7QgghBgiSR8QpdpMOK1GurySRySEEEIkq+QOiNw+XFYTLqtJ8oiEEEKIJJbUAVFvX
wCb2YjTZqIUpNICCGEEMknqQOivkAIs9GAjREIIUQyS+6AyB/CYjREIIIUQyS+6AyB/CYjLgsBolIBJCCCG
SWHIHRIEgZqMBh0WSqoUQQohklvCASCl1u1jRm5XRu4vJiLtPAAiLtPAiIhhBAiWSU0IFJKFQG3AvO0kA
duCFyyg1Ae6T9nshxCeMLhLAYFVaTgV4JqV72qnr8AxAxv1d0atHq+fTbUdb
KnrxO2T12ZxfByPzV1NgF0p5QccQD1wGnBl5P5P4H+P6WU0lrrRHRs/5S
ZzWykRwIiIcRhBIIhXtnUhOnUwB1PbaDPE4XFyTbZeEPV85h4Zisoe7aqLKv1c0Pn9/Em9ubhkT
hHfOnMiBen2Ie6ZGO0SOkKta4F/hvYRzNCi/pVxMtCkU+aPxmQqLmE8UbPuKxDpHWh1eh1
b7z+I4Ujfbr76WN9dFgCODpNbW8t6tlCHskkWip8wyCI/6VACFgBNY9mkfV2t9r9Z6ntZ
6Xk5OzjE/Tl8ghMVowG420tMq8zE0Rms60+MLHUdHgKh2MHq5u4+mrr6jms/RvP15w+G+
Pfmhn7ty3eMvsBPDD+JTqpeCuzREWjdrf3A08SIF0ptX+X6oT0bFRF0ptX+6rhiojfxcC5QARO5PA1oT0bF
AMEQopDEaFFazgV4ZIRJCHEZuig2lYttS7SYnZah6dAoZDYaWDK2/xTkvLLMIeiNSDaJD
oj2AQuVUg6llAJOB7YApuYAbwGXRY65FnkuMDoUdHPvUUKA2mAUgq72ShJ1UKIwxq
f6+K/zpuCIRIUWYwG/vuymZRkOOaoa2Y6PMxXOKGZ/rit6eW5bOKRNH10iYGJ4SmlSttV6hl
HoSWAMEgLXAvcBLwKNqZ9G2u6PnHI/8E+lVCXQRnhFWkKAFWkJ4/eHpMgCbb2UivrGQQhyG1Wz
kyvmlzCnNoN3tozDNRnwvlsBcdZIGlgavvMtNZ3A
Xcd0rwbmD/AsV7gs4nuE4RXmFlMBwVEMmUmhDgCq9nI
rVceMEJkMeHwSEAkhhBDJKmkDooNHiOwWmTITghklnSBkRefwhzJCCymox4/UESlL8th
BBCiGEuaQOiPn8wOmVmNCjMRgMev0yRCoaQMib+BADhEQKc4o02ZCCCFEMkragKj
PH8RsPFBlzWYxSmK1EKIkaSSNiDyBg7kEAGR4owSEAkhhBDJKGkDovAIKMbW1A5AuGMBkOmjIzG6K7WAshhBAiusSRtQOQPhGJHiExG3JJULYQQQiSl5A2IzdQ
oikFM8I0Zla6y7gPKAKGAd8KxGdOh78hwRENpNs3yGEEEIkq3gCInPk+7nAE1rrzgT057j
xBUKYDAeEemevsVkoFemzIQQQQoikZIrj2ZIrj2OeVUtsAD9E1MtxKvL3DoDoKjOjBERCCCFEMkp
jqqESKllAF4AVgMzNa+wE3cAGMzNNa+wE3cGEC+5ZQvmAotlK12UiPFGYUQgghkt
6GGnr1Vo3JLR3CeQPhDAaghkRLKKZ8rsDaXUpcDTWgZXYRxmVqmHXc09OK2kLJpzsjw9VLT2kOStM
kIkxGiyq6mHHwOVuJujjkkl9o9VVnmjwGjxHcMNHF4LHUDgaOSnokrpvPOaLG
m4BtAUCnlARSgtdapCelZ5GvkPzSGySA6REKPJ6r3tXHnPMrxRsL6a+tdUVnKT+lYaFM4//BSDA9eoP0C
```

```
N6vbOX6B1biC4YAuHBmIT84fwrZLglwxeA56lVmWusUrbVBa23WWqdGbo/IYAigLxi7ysw
mq8yEGDV6vQF++cq2mNpiy3e2sKFmRC+OTUqtPX1879mN0WAI4Ln1dWyp6xrCXonRKJ7Cj
EopdZVS6geR2yVKqfmJ61pi+QMhTAdNmdllhEiIUaPb52dbQ/83zPrOoVsY29bbx46Gbpq
6Ruzi3CHR0xdgb6u7X3trT98Q9GZk6/H62dnYTW27Z6i7MizFU4foT8Ai4MrI7R7gj4Peo
+Pk0FVmdrNRtu4QYpTIclo5Z1pBv/YJea4h6A2s3dfOZ//3Q878zbtc+Mf3ea+yhVGQinl
cZKdYOWVCTr/28mznEPRm5NrZ1M0ND67ijHve5ZzfLeeZtTX0+SVv9mDxBEQLtNY3E6k9p
LVuB46Y1aaUSldKPamU2qaU2qqUWqSUylRKvaaU2hn5nhE5VimlfqeUqlRKbVBKzTmmZ3U
U/EEdM2Vmt0hStRCjhdlo4KZTxnDqxPAbqcNi5IcXTGFGUfpx70tDl4ev/msNu5p7gfAo1
Y3/WEVVS+9x78tI5LSY+N65k5lXlgFAqs3Erz83kykFIzZj47jz+IL8/OWtrNjTBkCnx8/
tj61nS71MOx4snqRqv1LKCGiASGHG0OFPAeC3wCta68uUUhbAAXwXeENr/Qul1J3AncAdw
NnA+MjXAuDPke+D7tCkaovRQDCo8QdDmI1Ju+etEKNGRbaLP145h9oOD1aTgdJMB0qpI58
4yOravTQcMk3m9gXZ1+6mImdoRqxGmgl5Kfz9uhOo6/TgsJgoyXQMdZdGlKZuL29ua+7Xv
qell9mlGUPQo+EpnoDod8AzQK5S6v8BlwE/ONwJSqk04GTgOgCttQ/wKaUuBE6NHPYg8Db
hgOhC4B+RZf0fRUaXCrTW9XH086j4g7E5REopHNZwHlG6Q5ZzCjHS9PmDbKnvYndLL5kOM
9OK0slJsTIhL2VI+5XuMGM1GegLHPj8qFR4Wk8cvRS7mYl285EPPAY93gCb6jqpaXdTkGZ
namHqqHofcFlNlGU5+uViyTUY66gDIq31Q0qp1cDphJfcX6S13nqE0yqAZuDvSqmZwGrgP
4C8g4KcBiAv8nMRUH3Q+TWRtpiASCl1I3AjQGlp6dE+hRjh3e5jR4LsZiPdXgmIxOENxvU
nBt+rWxq55ZG10dunTcrh7ktnDnntofIsJz+6YCp3Pr0x2nbb6eMZl3tsOTBy/Q0ufyDEP
z+q4pevbI+2Xb+kgm+eOQGnNZ4xg+Ery2Xl/100jS8+sBJ/MJy7ds70fKYVybTjwY76f1s
p9U+t9dXAtgHaDvf4c4BbtNYrlFK/JTw9FqpuLILtdb3AvcCzJs375gyE8M5RLHD5
w6LJFaLIxuM608MrvpOD3c9vzmm7c1tzWxt6OKklP4JuceTwaC4eHYRUwpTqWnzkJdmY2J
+Cjbzsb3ZyvU3uPa09vI/r+6Iafvb+3u4aHYhM4qPf85Zoiwem80Lt5zInpZe0u1mJuank
CkjRDHi+YucevCNSD7R3COcUwPUaK1XRG4/STggatw/FaaUKgCaIvfXAiUHnV8caRt0A+U
K2S0mWXovxAjk8QVp6/X1a+/0+IegN/1ZzUZmFKePqjfY0aLHGyAQ6h9Xdg2Ta2ewGAyKS
fmpTMqXUaFPcsTsYaXUd5RS3cAMpVRX5KubcBDz3OHOjex1Vq2UmhhpOh3YAjwPXBtpu
gx3keuCay2mwh0JmI/CHon0MEYDcb6JaASIgRJz/Nxmcmxo4EmQyKsZK0LI6gJNNOaWZs9
fJ0h5myLFnWn2yOGBBprX+utU4BfhWpUL2/SnWW1vo7R/E7bgEeUkptAGYBPwN+AZyhl
JLI3cBngZ2A1UAn8Fvhb/Uzo6A02Z2cxGerwSEAkx0jgsJr5/7hTOnpaPUlCe7eDvXzyBi
UOcUC2Gv5wUG3++ai4LKjIBmFGcyt+vO0FWsiWheKbMPlZKpWmtOyFcXwg4VWv97OFO0lq
vA+YNcNfpAxyrgZvj6NMx8x+ydQeEAyKZMhNiZBqb6+I3l8+iqbsPh8VIluxzJY7S1MI07
r92Hm1uP2l2M2kJWs0mhrd4Cu7ctT8YAtBadwB3DX6Xjo+Bp8zCq8yEECOT1WykJNMhwZC
Im8tmpjTTIcFQEosnIBro2BG7JnHgKTMDXd7RlUgnhBBCiCOLJyBapZT6tVJqbOTr14TrC
o04oZAmGNIYDwmInFYTHW4JiIQQQohkE09AdAvgAx6NfHmBmwZzUK31Wq31PK31DK31RHwdt
3/6W7QgghBjd4qlU3cshRRVHqkM3dt3PaTXRKTlEQgghRNKJp1J1DvBtwanUabfvbtdanJ
aBfCeUP9E+oBnBajHTJlJkQQgiRdOKZMnuI8LYdFcCPgCpgZQL6lHCftKN9eIRIAiIhhBA
i2cQTEGVpre8H/Frrd7TW1wMjbnQIwhu7HrrCDMIB0Wgr1y6EEEKII4tn2fz+SKFeKXUu
AdkDn6XEs8f1JgHmjKzSh0iIYQQIhnFExD9VCmVBnwV+rfH+D2QCtyekfH4l2EBVqgEsRgMhrfH
6g9jMxiHomRCjm9aanr4ADoupX9kLMfK5fQGMSmGV108xAh1VQBTZ2X681vpFoBP4TEJ7l
WC+T0iqVkqRYgtPm0lAJMTg2t3cw2Mrq3l9axMnjs/iqgVljJe9xkaFTreft7Y38dflu0m
zm/naZ8YxvzwTiymerAwhhtZRBURa66BS6vPAPQnuz3HxSSNEAC6biQ6Pn9xU24D3CyHi1
+728c3H17O2ugOAXc09vLezhUe+vFD+1kaBN7c3cftj66K3P9zdyhM3LWJe+YjMqhBJKp7
w/X2l1B+UUicppebs/0/0pYzxLIH9QDjhABpNrMtPVKcUYhBlNVS280GNpvV3Mvu1t6h6hHY
rD09gX46/LdMW1awzs7moeoR0Icm3hyiGZFvv/4oDbNCFxp5g+GPjF/IdVmprVHAiIh4uH
xBdhQ08m2hm6yXVbG5DiwmU0Up9sxmwxYBihzAXxiuxg+Gjq9rK9up76rjwm5LmYUp+OyH
XjrMCpFqrX/W4lrgDYhhrN4KlWP6Lyhg/mCIcyfMGWWYjPR1tt3nHskxMj28sYGvvnE+uj
tqYWpzC5Jx2hQfOWUsZTnOPns3GKeWF0TPeb0ybkUZ9rRWvfbRkcMD609fdzx5Hre2dkSb
bvr/Clct7g8+n9msxi58ZQxrKhqQ+vwMU6LkYVjwtNlbl8Aq8koSfRi2IsrhI8stz+0UvW
PP/mM4ckfCGH8hCkzp9VEq0yZCXHU6js8/OSlLTFtm+u6OHNqPve8toMxOS6uXVzOt86aw
AWzCqlq6cVumoIBrrn/Y06fnMelc4c4sYk+0aomcgPsm2hu6YYAjg7le2c9qkXMqynNG2mnY
3dy6bxO6WXuxmI7kpVvoCIe59dxfPrq1jenEq1ywqEq1ywqZ2phWsL77PEF6QsESXdYEv67xOgSz
9Yd/ws4CK8wuw+4DPg4Qf1KqPBeZp88ZdbSLSNEQhytvkBowIKmwVAIgKfX1HD+jHerq2z
lT2/vwmYycOncYp5WW8u/nbdCfImNsy4ff3rsnn8Qbz+UPR2l8fPgx/sY
```

```
1dzD3mp4UCoIsvJzqYenllbC8CW+i5e3dzIszcviQmkBpPWmlV72/ndGzvZ2+rmihNKuGh
2EYXp9oT8PjH6xDOBv1hrfQ3QrrX+EbAImJCYbiWW/xMqVQOk2k20SA6REEetIN3GBTMLY
9qsJgNGQzh36NK5RbxX2co3Hl9PZVMPm+q6+NELWzhnekH0+DX7OtgjCdbDzpgcF05LbAm
Sk8dnU5xuxxcI4vYFsJuNTC1MBaCxq48Ot58l47N5bl1tzHntbj87GroT1tet9V1cdd8Kl
u9sYV+bm7v/vZ1/flRFKKQT9jvF6BJPQOSJfHcrpQoJV64uOMzxw5Yv+MlTZqk2M62SQyT
EUbOajNx+xgSuW1ROmt3M9KI07jx7Euv2tfOfZ02ksqmHBz/c2++8zbWdjMk+MFpgkgTrY
Wdsjot/3rCABRWZpNpNXH5CCXedP4UNtR186cFVfO4vH/LChjq+fFI5qfYDEw4uq3HA/SI
T+X+8raGbvkAopu3v71fR0OVN2O8Uo0s8OUQvKqXSgV8BawivMLsvIb1KsMOPEMkqMyHiV
Zbl5AfnT+ErnxmDP6DZ2djNJXOKueWRtcwuSSfNYe53jstmojcyJXPBzIKY4EgMH3PKMvj
btSfQ7fOT5bSyoaaDL9y3gv0DL994fD13XzqD529ewvaGHiwmA5PzU+gLhLjntZ3RxxmX4
2JifuIKcdoHKKabYjMPuE2TEAOJZ5XZTyI/PqWUehGwaa07E9OtxPIHPrkwY7rdLEnVQhw
Do0GRnxrO1yjJdPCVf64CYG11Bz84bzLv7Wwh1W7mygWlGBTMKE6nJMNOfpqdRWOycMoy7
WHLaTPhjCy1/2hXG4fOQv3l3d0sm76Ys6YdSIy/emEZE/NSeHdHCxPzUzh5Qs6nyufZ09x
DVaubdIeZcTkuUuyxQfbUojRKM+3sa/NE2757ziRyUqTwpzg68a4yWwyU7z9PKYXW+h8J6
FdC+YP6E5eAumwmevsC9AWCWE2yfYdITg1dHqpbPaTYTFRkO49pb6rUg96w/vZeFXecPYm
8FCt3Pr0Rty8IwLWLyrhyQZkkU48gNouBRWOzWFiRSUhDXYeHXc09/UbdM51Wlk0rYNm0T
59Z8fGeVr7495X0Rq6bqxaW8p9nToy5bkozHTx4/QJW7G6lvtPL/IpMZpWkf+rfLZJHPKv
M/gmMBdYBwUizBo4YEEX2QlsF1Gqtz1NKVQCPAlnAauBqrbVPKWWNPN5coBW4XGtddfRP5
+j4DlOY0aAUGU4LTV19lGQ6BvtXCzHsbazp5MZ/rqK+04tBwa2nj+f6JeWk2uMLWj43r4R
n1tbiD2pqOzxsb+ji0Y87osEQwIMf7uWsafksHps92E9DJMj8ikxWV7Vzz+vh6bDxuS5+d
OFUHJbEjPC19/r47jMbo8EQwL8+2sfZUwtYMj72uqnIdlIhU6/iGMVzBc8DpmitjyVl/z+
ArUBq5PYvgXu01o9GlvPfAPw58r1daz1OKXVF5LjLj+H3HdbhKlUDZDosNHVLQCSST7fXz
49e3Ex9ZzgRNaThN6/vZEFFJoviDFrmlGbwxE2LeWt7EwCnTMjhydW1/Y5r7JJFDCNJdZu
Hlzc1RG/vbOrhpQ31zC/PTEjSdJfXT2VT/xWITd2SLC0GVzxX7yYgP95foJQqBs4lkoCtw
uVNTwOejBzyIHBR5OcLI7eJ3H+6SkAJW99hcogA0h1mmmRlgkhC7W4/q6ra+7XXdsT/92A
wKGaVpnP7GRO4/YwJjM9zMa88o99xZfLBY0TZVNs/dfSdHc109/WvWTQYslwWFlT03yRWP
rCKwXbEESKl1AuEp8ZSgC1KqY+B6Ec6rfUFR3iI3wDfjpwP4WmyDq31/r+eGqAo8nMRUB1
53IBSqjNyfGyp1E/Jd5hVZhAOiBolIBJJKMNuZk5pOmv2xW7EWpB2IDG10+Onpt2Nw2zEZ
FT0BUJsb+jGaDAwpSCF0k8ovJdiM/PjC6ZyyyNr2dXci81s4HvnTGZyQeJWHonBN7UwlXS
HmXOnF+CwmHhjayMnVGQOuHdZa08fdR1eUu2mYy7I6LKaueuCqdz6yBoqm3qxmgx879zJT
ClMPfLJQsThaKbM/vtYH1wpdR7QpLVerZQ69VgfZ4DHvRG4EaC0tDTu8/2B0Cfudg/hWkR
Su0J8kk97/Q1nKXYzd50/lS89uIrmnj6Ugq+cPIZpkS0XKpt6+O4zG1k6OY+3tjXxcVUbk
/NTuHRuMfe8vgOX1cQ/rl/AuNyBt+GYUpjG4zctorbDg8tqojzLiUH2uIrLUF9/Y3NcfO3
Usdy3fA/d3gCXzC7ikjlF/eoObazp4JZH11LV4sZpMfKTi6Zx7vSCARP06zs89AVCFKTbB
lzMMqUglcdulOtGJNYRAyKt9TsAkUToeq21N3LbDuQd4fQlwAVKqXMI73+WCvwWSFdKmSK
jRMXA/sSCWqAEqFFKmYA0wsnVh/bpXuBegHnz5sWd0+QLhgasWbFflstC/TFMEYjk8Gmvv
+FuZkk6z319CXtb3TgsRrKcZiwmA75AkD++VUlJhp3HV1VT2dQDwKa6Lva17eALC8r48zu
7eHt70ycGRABZLitZLuvxejqjzlBff5VNPfzs5W3R2w99vI80h5lpRWnRYKfD7ePbT23Aa
TFx3eJyajs83PHUBsbnpjC9+MB+Zm5fgJc21PP/Xt5Kl8fPhbMKuW3phAFHk+S6EYkWTw7
RE8DBZUCDkbZPpLX+jta6WGtdDlwBvKm1/gLwFuG90ACuBZ6L/Px85DaR+988xiSpDT
lGm00pdp+cT7xditCtMt5NmN/P7N3dy1m+Wc/PDq9lc18Xync0UptujwdB+Xd4AJZl2bji
xIpqQLUanTXX9c4he2lhPU7eXTbWdbG/ooqHTy5lT8inLcvDU6hoaOr1895zJNBzyurqhp
oNvPbmBDrefkIZn1tbxwAdVBGW7DTEE4gmITFrraMXCyM/HWjzkDuAbSqlKwjlC90fa7we
yIu3fAO48xsc/LH9QH3bKLNNpkZUvIqk1dXm58Z+reH1rE72+IG9ua+brD63h+hMr0IBlg
NVELT0+nlpTw+ySdBLwOUYMExkD1IwqzrDz4Ad7Oe/373H2b5dT0+5m9d42Xt7YQHdfgI2
1nfzylW0xtakAttb339vs2bW1tPTI6684/uIJiJqVUtEEaqXUhcSR7Ky1fltrfV7k591a6
/la63Fa689qrfsi7d7I7XGR+3fH0b+jdqSk6kyHhcYur7yoi6S1t81NTXvsp/naTi/F6Q6
W72jm6kVlMfctm5bPqqo2Otx+vv3UBqpa3cezu+I4Kkq3U5xxoOK01WTgs3OLuf/9PUC4V
MOW+i7eq4zNdvD6Q7QeEujkpvSfAhub48JplaK44viLpw7RV4CHlFJ/iNyuAa4e/C4l3uG
27gCwW8KrZzo9fqmgK5KSwzLwG1KGw8yJ47IpSLdxz+Uz6XT76e4LsLmui3d3hj8fuX1B6
jo8UiBvlMpLtYVXmFlNhEKawnQb//poLwd/fuz2BnFZTfQcshT/0BGiWSXpzCpJZ111eFW
j1WTg28sm4rL23/tOiESLZy+zXcBCpZQrcjsmiUApda3W+sEBTx5mDrfb/X7ZLit1HV4Ji
```

```
ERSGpPt5PolFfwt8qkf4NrFZcwty8BqNrB2XwedngCT8lx87eG1MbuMmwyKbJf83YwW/mC
I3c29tLt9FKXbmVGcRpfXz/+8up1OT4D/PHMCY3NcrNp7oFTDixvquG3peH760tZo2+KxW
UzMj10qX5Th4H+vmsvW+k56fUHG57qYkCdlGMTQiLvW+qGB0EH+gwNFFYe1w+12v1+200p
jl1dqXYhRLxTSePxBzAZFQGscFhM2s5FTJ2ZTke2gyxsg22VhTLYTm9nI/IosZhSn0+31k
2Yz87OLp/PtpzYQDGkMCn584VTG5HzyKjMxcnj8AR5fWc1PXtxKIKRJs5v5y9VzOX1yHvP
LM/EFQ2S5rEwuSGVjbQefmZRHb1+AyuYezpySx7SiNHY2dpObamNGcRqpNhNb6rpo7vFSm
GZnbI6L/DQb+WmyAasYeoO5+cyIKQoRTqo+fPpUhtMsK83EqLejsZuHV+zjw12tLBqTSVG
GA5fNxOySdN7a1szG2k66vH5OmZBDh9vP1vpuxua6yHJa2NrQhUIxsziNF285kYZOL3mpN
sblOvvVpBEj046GHtbXdHLr6eOj5UruX76bMdlOclMPBDFpdjNXLSzn0ZXVuKwmrl1cTk6
KldIsJwvHZAHQ5w/y6Mp9/OiFLYR0ODH/91fO5qypcW+AIERCDGZANGIykMIaCHIAACAAS
URBVMPL7g8fv6U7pBaRGJ32NPewo7GHdKeJbz+xgb1t4cB/e2M388oyyHJZaO/1UZ7loDD
Dzpa6Tva1uslyWXlqTQ3razq5elEZq6ra2FrfTardxCNfXshnJuUO8TMT8drZ2E1lUw92i
5HJBankpcaO1HR7/TR0enl6zYE96L526lg6Pb6YgOjNbU1879lN0dsf7W7l/mtPiLkmdjX
38MMXtkRzjXzBEN96cj2TC1IpPcI2HK29fXxQ2cpz62qZlJ/KeTMKmFQgo/dicA3mx7gRM
0LkC4SO+Ak202GhpkNWyojRZWNtJxf+6X1u+tdqPqhsiwZD+63a286k/FT+9v4eyrKd3P3
KNp5ZW8e/tzTyi//bxtnTC7AYDTy8Yh9nTAnXZe3yBHhhfd1QPB3xKazZ286Ff3yfrz60h
uv+vpIb/7GKmvbY1zyvP8gHu2JXiz34QRUHlwlq7vby8Mf7Yo4JaXivMnYRcmNXH4cu3O3
yBGg7whJ7rTWPrazmlkfW8vrWJv7wViVfuG8Fe1r6b/gqxKcxmAHR+4P4WAnVFwhiPkJSt
VSrFqNNIBji/uW76fKEV/4MtG3y/jab2UhrTx8OS+wg8qubG1g0NotgSMe8ucmb08ji9gX
4n9e24/YFo23razpZszd2Y9/AAAUSe31BAkHN5rrw8b5gcMDK/zZz7NtLYbq938h8jssaM
9I0kLoOD394szKmrbXXx9b6rsOeJ0S8jnrKTCmVDlwDlB98ntb61sj3rw925xLFF9RHHiF
yWmQ/MzGqeP1BNtcdeBPZ3dzbbyPXi2cV4fUHuX3pBNy+ABfMLCQ/zcZ7lS18uKuVkA4PB
VdkO2P+Pi6aXYQYOXr7guxo7L8+pvaQD4Hjcl3YzUY8/gOB06IxmWyp72BPiweTUdHR6+O
LS8pZva89GiQ7LEYWRXKH9hub4+Q3V8zijic30OsLkuW08PvPz6Yw3c7haOg3sgRInTgx6
OLJIXoZ+AjYSOwWHiOO/yhyiMLVqsPFGdVAH6WFGGFcNjMXzy7i7n9vB+D59XVct7icC2Y
Vsrm2i8J0G9XtHtZWtzMxP4UfPLc5eu5XThnDvlY3y6bmsaa6ne+fO5nvPL2RbJeF/1g6v
t+bnxjespwWLppVyF+X74lpn1qYyordrexrc5ObYmVqURp/vmoOv/r3diqbejh1Yg7XL6n
gxQ31PLayGl8wxInjsjl/ZgG/u2I2H+1qxWk1MacsnQynhaZuL3XtHlLtZsqznJw7vYBph
Wm09frIT7MdMRgCKEyz89VTx/Lr13ZE29IdZiZLDpEYZPEERDat9TcS1pPjyBcMYTYdfoT
IYTFhVIouT4A0hxQJE6PDhbMKqW738PiqahxmI4sqMgADf6+qor7Diy8Y/qyT4bAwoziND
TXhfav+9dE+fnvFLLKcZq5dXIHdYuSFW9JQcMQpDzH8GAyKqxaW0djVxwsb6nBaTNyxbCI
dbh+3ProuetyFswo5aWwW/3nmBHr6gqRYText6+WfH+2NHvNeZQtjcpxMyk8h3WFGo9nX5
qGuw83jK2vZ1tiNzWzgR+dP5aLZRZRnOymPo2inwaC4cn4pxRl2nlpTw5SCVC6eXSylHcS
giycg+qdS6svAi0A0C05r3TbovUowXyCE+TCVqvfLdlmp7/JIQCRGpI21HazY3UZIaxZWZ
DG9OI1gSHPJnEI+O7eYYCjEjsZuqts97D1kq433K1u5bG5xNCDq9YX3ozpvRiH2SBXrQ1c
kieFtS10XH+1ppc8fYuGYTGYUp/Ory2Zw29LxWEwGQiHNGfe8G3NOZVMPk/JTuPOZTQRCm
rE5zgFXE765rYlOj5/n1oWT660mA7+4dDrtnvD2l15/iDue3sjkwlRmFKcfsa9N3V52NfV
gNBgYl+skO8XKJXOKuWRO8SD8SwgxsHgCIh/wK+B7HFhir4Exg92pRPMFQ0dMqoYDidWT8
mVoVowMe1t7aenxEQiGuOZvH0crSFuMBu6/bh43P7yGLk+AaUWpLKzIot3t4+TxOXz9tHF
UNvXw+pZGAiHN+FwX+9oOBElnT8tnYl4K2U4zG6o72N7YjctqYlpRGiVHWDItht6m2k4u/
8uH9EaSqE0GxcNfXsD8iqzoSMvm2k6sJgNfWFAa3ZYj3WHmZy9viz5OdZuHFGv/t42J+Sl
UH3S99AVC7Grq5fvnTkYphQ7Bij2t7G1l9wuIqtvcbKrtpKcvwKT8FOwWEzc/tJrtkRyne
eUZ/PqzsyjNkutMJFY8AdE3gXFa66Pe0HU4CoY0oZDGeIQcIghPG9R3SmK1GP5CIc0b2xq
5/bH1ZLsszC7JiNlOwxcM8djKasZkuzhxfDbjcly8tKGOZdPy+e0bO9nd0su0olS+e85k/
vhWJbctHc/97+2mIM3G6ZNyOX9mIXe/sg0N3PLIWoKR1Ufjclz87boT5M1qmHtja2M0GIL
w6rG/Lt/DnNKMaJFal83EnWdP4n9e3UFrrw+72ch3zpkU8zi+YIh2tz8mGT/DYWbp5Fy+8
/SBOkQGBVMKU3nggyo+3tNGUXo4D6ggNXYz132tbq5/YCWVzeHgpyTTzjnT8qPBEMCqqnb
e3tHENYvKB/XfRIhDxRMQVQIjvjCPLxDOHzqaROkMp4V6qVYtRoCq1l5ueWQtXn+IsiwHX
X3+fsd0evx8cUk5//XcZkozHXxhQQl3Pb8lugHnptouOj17+P3nZ9Pu7uPiOSUYFDy7poa
XN9Zz4ewifvfGzmgwBFDZ3MO66nYJiIa59t7+10NrTx/BkMYUWTHf5fXz2zd20trrw6DA4
w/S5fFz7vQCJuanRCtVf7y7lR9fOJVOTwCvP8i4XFd0c9b9lk7O4+/v7WFlZBl/bYeHHz6
/mb9eMy/muHXV7dFgCKA8y8mKPbFL/wFWVbVJQCQSLp6AqBdYp5R6i9gcolsHvVcJFC7Ke
```

HSrxjIdFmraJSASw199pxevPzwiVNPm5tbTxzGnNJ1AUFPV6uaZtbVcMLOQTo+Pbq+fjbW
d2M0V/XYjr27zsDEyffH7Nyuxmgz87OLp3P/ebmYUp9E4QCmKdnf/N1sxvJw5NY8HPqyKa
btucTk9fQE213VhMxsJhTSLxmSxdEoerT0+0h1mjEqxqbaL372xE4fVSJcnwF3nT8Ef1Cw
Zlx19rDS7md9dMYvHVlaTYjNz1cIyrrp/RczvC4Q0DYd8wDz02tlW3825Mwr6BVgnjsv59
P8IQhxBPAHRs5GvEa0vGMRylPssZbksrKvpOPKBQgyxnBQrZqNCA7/67Ew21XXS5QmQl2r
FYIDfXTGLhz/eR1NXH19cUsGLG+oGLMxoNRliivX1BUI8sbqaJeOyeb+yhcvmFg+4VFsMb
7PL0rnv2nn87o2duH1BvnrKWCbkpfD5v37EjsYelIKvnTKWBWOyuPWRtdFK1NcuKuPMqXn
Mr8jA6w9hNxv4cFcr88oyYh4/3WHhgllFnDujEIOCva1uMhzmfgFPdkpsIv6h105zTx+zS
9Opasnh7R3NAFw0q5ATx2cjRKIddUCktR4RO9kfydFs27FftstKfYeMEInhb0y2k59fMoM
ut4+fvrQ1JiH6m2dOoNPjp7KxhzSHhXnlGRhUeJ+qaxaW8Y+DllDfePIYphWm8otXDiTS7
mt1c/Op47CYDBSk2QDFQyv2kuWy8INzpzC9KO14PlVxDOxmE/kpNq5fUkEgGGJivou/Lt8
dLc6odbhg7c9e2hqzLYfbF6S1p4+2Xj/1nV7G5Dg5fVJ4tObp1TV4A0Em5acwuzQDpVQ0N
7M828kPzpvCN59YHy2qeM70fOaWxSZUTy9K4y9XzeUnL22hrdfHFxaUMa8sg9Mm5VLV0ov
BoKjIdvarmC5EIsRTqXoPA2zgqrUeUavM+uIIiLJcFhq7+giFNIajSMIWYqiYjAYumFHAC
xvqYoIhgH99tJfvLJvELy+ZThC46V+ro29SZ0zO5defm8mu5l5KM+x0evzc994eclKs7Go
Ob8dx2bwSTppwYMrizrMn8cUl5dhMBjJdsUmyYnjaWNPBZ//yYXRa9YoTSnh3R+z6GKvZQ
PchU6gnjsviT2/vZntjd7TtqgWl5KbYqW53E9Lh/fH8gSBWs4kdjT1YTAYm5adw3owCKrK
d7GruIdtlZXpRGplO6yG/08hZ0/IZm+vE6wtRkePEGVnFNv0olucLMZjiCbsPzoazAZ8FM
ge3O4kXTw6R1WTEbjHS2usjJ0Ve+MXwVt/lHXBVZIfbT3aKlb2tbl7YUBezDcJrW5sozHD
w/Lpaur0Bbjx5DPPKM3l9ayNmo+ILC8q4fF5s7RejQR1VhWExfLy2tTEaDAHsaOxhZkkar
29tiralWE2UZNqpPmjDX08gFBMMATy+qoZzphdQ3+nFaTVR1+EhoOGmB1bSEZkim5jn4pe
XzmB2aQazS2On1w7W2xfguXW1/PzlbfT4Aiybms8dyybFVbhRiMESz5RZ6yFNv1FKrQb+a
3C7lFjxTJlBODejrsMjAZEYngI+aNkOHdUE7NNwWU2YjQp/8EDU8/kTSvjHB1Uopej2Bvo
9hMcXxGoyYnUZsZgMXDCzkGsWleEPhshPtUWXZYuRqyfy/z4m24nZaGBtdTs/v3g6W+q6q
IsE0ekOC9cvqeDBD6qoanWT5bRgG6Civz8Uwh8M8c6OZrq9AT47t5gt9V3RYAhge2MPK6v
amHWYYAhgfU0H333mwHL9/9vUQI7Lyn+dP0WuO3HcxTNlNuegmwbCI0YjbmI3nikzgGxXe
KXZzBIZvhXDTCgEm56E527GUzCf6uk/ZnujjV9cMoNHV+6jpt3DxbOL8PiCvLq1CYfFyE0
nj4nZ4FWp8Kabj6+q5p7LZ3LK+ByZBhuFzpqUQUG6nY01nfiCIS6dW0y2Hc6alk+KLbxNU
YbDzI9f2MzZ0ws4b2YhPd4AIa3JdFpo6/VFH+ucafn8+Z1d0dHIv39QxZdPGkOOy0pzT3Q
BMrubewkEQ7T2+HBZTTht/d8utjd092t7fkMdN582Tiqhi+MunoDmfziQQxQAqghPm40ov
kAI01FOmQHkuKzsbetNYI+EOEZtu+DF28Fo5r2J3+PLz4VzQh5fVcvPL5lGfqqNbQ3d/On
tXUA4QXbNvg5uXzqeV7c04rKZuHJ+KaFQiKe+uphphalYzcahfEYiQQy+Xn75fzsJRDKmX
9nUwN8uH8eHu5rZ1hB+ffv7dfM4d0Yhj6+qjp43vzyD339+No+vqmZzXRdnTcljfF4Ktz2
2LubxX93SwKKxWTy/vi7aNrcsg588v5lnN9QzIS+FO5ZNZF55bJbFQEHP+NwUXJJELYZAP
Ffd2cClQPlB510B/HiQ+5RQvmAwrhGi3FQbu5slIBLDT7fHx44l/4s/pZg/ruhjRnEqn59
fhscfxKgUy3c0M6UwdgXYOzua2VzXya8/N4uO3j6mFqYxNtdFbXu4/lCGw0JFtlMWEYwyb
1R2RIOh/f62spkvLizioVWNdHsDGJTCFwhy+xkTqG5zk5dqo6cvQFtvHxfMKOScaflYTAb
2tPSvz1uUYacs24FS4W1irltSTorNxIMr9gGwsqqNq+//mOe/voTxeSnR82aVpHNCeQYrq
8LFGG1mA3csmzjgaJIQiRZvHaIOYA1wVPtZKKVKgH8AeYRHl+7VWv9WKZUJPEY4uKoCPqe
1blfh8tG/Bc4hXBX7Oq31mjj6eETxJFUD5KfaWF8ttYjE8NLh9nHfZkV6IIuTC50sqHAyN
ieFVzY34LQYmVGSTnm2E7NRccbkXF47KHn2CwtK2VzXyS9f2c7VC0u5dE4xNzy4itZeH1a
TgZ9cOI0LZxXKaNEo4jf0ryQ+Nj8du81BtsvKlIJULCYDc8rS+elL28h0WGh3+7hucTk17
R5++cp2ILwH2gNfPIHCNFs098hiNHDNwjJOKM/gpHHZmA0Gsl0WTvrV2zG/z+MPsqu5JyY
gKky388cvzGFbfTfuvgBjc10x9wtxPMUTEBVrrZfF+fgB4Jta6zVKqRRgtVLqNeA64A2t9
S+UUncCdwJ3KKVmAFcCM4E84HWl1BStdWs/XXOPqKpvYsWGYn7+0kbqGlsMJEamDeHuC2dw1vRRZPXwVJoIj+nUEYKDPZP9Kjdbam+qtlYmdYQQ/tfS2mYWGHuqqvmcDwu545k9zM5L5+2RhCVHhhfSDdKK9cVcMr2hXQ+MDWhRDkc+ueYdHlxSUOFIBtx+pPz4WeBLgLjKsiMhsWpPTmYUNY/yJwo+nA9jYr/dXl7
x/PdrO6JL7y0mA52HrDzTOrz3lAREo8ee5h5+fskM/vnhXvoCIW5bOp7Ve2P3DPP4g2yq7

aS6zU2qw8LW+m7ml4TLK5iNCpfVRCCkmViQwhOrqqPTXDkpVm5fOp7W3j6+/vA60uxmvnH
GBO5YNpEfPLc5+vgLx2QyqUCqmovhK553+BOB6yIFGvsABWit9YyjOVkpVQ7MBlYAeQcFO
Q2Ep9QgHCxVH3RaTaQtJiBSSt0I3AhQWloax1OIP6naYFCUZjrY3tDF3LIByi5pDVueg4l
ng0lW5ySDT3P9Haut9d184b4V0R3s7369B//i87j4rLnkZ6Rw8RQTd72yl9ZeH/PKMvj2s
kmAjqkz0+728+rmBpaMzea9ynAC9q2njefljXX9fl9+mnxiH66O5forzXTy9UfWRm9/95l
N3Hn2JCblO7l0nIlmr+aJLR7Kspw8vbaWrfXdFKTZWDRmIj89LZ3Flt2kBdppsJawIRCIB
kMAzd19vF/ZypySND4/vxSzUbG9sZvpRWk8/OUF7GjoJi/NxszidClfIoa1eJOqj4lSygU
8Bdymte46eKd5rbVWSvWrgH04Wut7gXsB5s2bF9e54RGi+OpblGU52FL3CQFR0xZ44lq49
H6YfllcjytGpk9z/R2rbQ1d0WBov/tWd3HmDYvZ0e7hV29XkZNipdcXYF11B1azkZPG9d/
/aUtDFz+9cBrZLiszS9JYOimXKYWpfOVfq3H7ghgUfO/cyUzIcx2PpyWOwbFcfx/ubunX9
tqWBh4+oYrMd74Hjkwuv+Q+bnxlG/WdXqYVpVLX4eXWxzby2OfL+HBfKds6yvlMiWJfS1e
/x1pf08EjK2t48MPwNjDlWQ5+eMFUTp2Yy+Kxsg+ZGBniKcy498hH9aeUMhMOhh7SWj8da
W7cPxWmlCoA9n+MrQVKDjq9ONI2aHxxTpkBlGQ42FDTOfCdNavC3+vWSUAkBp+vF/Z9hMX
Tvw5WptOCy2bF6TLxpRMrMBkMpNqNNPf4aO/1U5LZv5r0woosHvpoL7WdXp5fX0tFtpNTJ
uTw0q0nUdvuJstlZUy2UxKqR5kUm7lfW5bDhC+o+fiUB7Hjw+juYk5ZLvlpNrbUdrGwIgu
r2cgHjYqN9QYmZcJv1/j53An9g+VTJuQwP7OHy87oJKDMvN+dQmVTD93eAC9trGd6URpnT
c1nXK4E2mL4SmhSTGTV2P3AVq31rw+663ngWuAXke/PHdT+daXUo4STqTsHM38IwvPkFlN
8L/Zjc13c/96ege9s3gaZY6Ft9yD0TiQtdxvUr4eeJsgoh/zp0NMYrjXU28xpuTaWXxJAp
RSgAm4KzB5U1y50czcppnTsNjv/tdzD0qkFTMh18uSqGqYVpnLd4nIeWrEXf1AzrSiVmSX
p0YRqgPZeH0qFN9CskO0SRq1TxmUQ6knjlMw2DARY3ZvD/KljueJpN1Wt4UUjl8/NZnK+i
9e2NTO7JJ2q1l72tvTypwuK+Kp9E8rbztdPLGW9JY8/XlROTl8VFvxUG4oYV5JObm8TeB1
gNFGa2cPuYDu1PSa+O9NLQ1+Q7z3TxM/On8DG2k62N7mZVeRiblkW2RmSVySGh0RnCS8Br
gY2KqX2L2f4LuFA6HGl1A3AXuBzkfteJrzkvpLwsvsvDnaHevsCWAcoR384FdlO2nt91HZ
4KDp0D6e23ZA/DTr2DWIvRVLxdsLrP4I1D4RvF8+Hs36O9rTC5mdReVNxvPQtHKULoWgeI
asTw6vfB8KJfGkzr6Ro8jX8v2VFrG0xYjIZ+PGFU/hgVxvrq9v57jmTSbOZqelw8z+vHgi
GlIIxOfKJPRnk+Oq4PmcbO0MF+LBwfmEPv125NxoMATy5tp77rynEsL2Zv7y7m4l5KXzn1
HyKVv8ctekJAIzAzHN/w8yaDzFuegyAmanF+Iv/Ss/ud9nnmIJZuSlyb2Pc1IuZ17oG1Vx
HqdXFH88+k9tf3snyyrbo77x+UQl3nD0FqxRiFMNAQq9CrfV7hF+zB3L6AMdr4OZE9snjC
2I1xxcQGZRiZkk6r25u4ItLKmLv7NgHky84MHUmRLyath4IhgpmwtSLoGkLqnETVJxCKOh
DX3IfymxD6xDGx66MOV2tfxjvmM9xyePb8Qc1BWk2fnbxdDIdFqYXp/OTF7fw5y/MZUyOi
5PH5/D2jmZyUqz8+MKpTJZVP0nBYNA82jWdOa5WzNrPC/XZTCvK4Ol1DdFjzp1ewN3/3s6
W+vB2Glvqu7C1t2OKBEP7Gd+4C2ZfHb2tumqobnPzrb2nsaamB4DLpy3l5h43GW/8BDzhb
TB3nTWT5ZWxRW4f+KiaS2flMbUsDyGGWtLtntfri3+ECODkCTk8+EEVoUOqvdJVC1njwlM
eQhwLz0GFP0/4MhhMYLbB7jfh2Zsw/Ps7GDv3YnjiGoz7PgC/p99DNDc3RTd0re/08t//3
k55tpMMh4VZJek8tqqa/3xiPWdPz+fV207ixVtO5OxpBViO4W9BjDzugIEvt97Nye9+nhO
XX8VN+/6T2c4Wxh40QliW5YgGQ/vZAv33GsPbCeYDI+WhjLE8usceDYYAHtvUzfouB2SPj
7b19fZPxg5paPcel3UJQhxR0r0a7t/ZO16T81Owmow8dtA+P/jc4TcnVx7oUDgBVoh4ZY0
FswMcmeDvheZd4aBo6iVw6p1wyrfgg9+F34g6ayB7Quz5Fid7dOwn7M31XTT39GE3G/jaq
WPx+APctnQCpRkOJuSnysaZSSa/fTXOva9Hb5ubN1FR8xxnTM6JthWn2/p9WKxWhWC0xLT
ponnQfGDqtSd/Pm/u9XOotU1Az4ERqLGBneSnxi67n1OagesTtunocPvodIcLhnr9QVZVt
fGvj/byfxvrqW2XYrli8CXdxG04qTr+OFApxReXlPOLV7ZxQnkG43JToKsOnDnhZAxbGnj
awSKJqSJO2ePhyidgz9vgyg8H2B/8AWoj07Dn3YMvazKGlFJMm56CpXfB5mehZiVkjaN76
d387MnYN6SxuU6au7wsLEvh5S0thEIwuzSd2SUZx//5iSGX2rIalAF/yWK00Ypl33LS695
l8tzruPX0cRiVotzcxneWpPDDdw6sqP24zc6Zl9yP6fX/gvY9ULYE/6nfw73uWdKVAXQIC
0EWlqex65A9H6fl22DzgTUxRWt/y68vXs6Dq5rZUt/F/PJMFlRkkR4bb9Hh9vHq5gb+8NY
uzCbFN5ZOwG42cv2DB9ISZhan8b9XzaXg0JxOIT6FpAuI3L4gtmOcJijLcnLl/FKuuu9jH
rj+BEo6a2ixjKcMwJYanjZLKx7U/ookUXEiuApw71uNrXE1hkgw1DnmfN71zeGBnglkWuG
G03ycsOIbGHMnELj2JTZ1WHlgvZFl08w8FNlIM9Vu4o6zJpFnC7KhwcOuFjcr9rThtJhki
ixJ9Zafxcrcy7l3qwlPH9xw0m3MM1eRk55CVWcXVpOBVFcKl/Xcy5Rl51LlSyXP7GFqdg+
mZ28K57VNvxTq1mJ+7mu8MuOvpC5Zhhk/bzY6WDoliw92d7AnkqT9mQnZVGTZ6Bp7Pqm7X
8KfOYG6+d+nxWti4Zgs5pVnYlKQYtGU5mXF9PXdHS18+6kDGyLc/PBavnvOJJQiWmF9fU0
nW+q7JCASgyrpAiKPL/5l9wc7aXwOSsFn//dD/H4/Jn0d39/axxXWlPAIkRDHoMcb4J1qA

```
+uqx/C9ht+HGy0u3iq6iduePlCK681diifO/hkF3l1U9ZXxZl03xZlGjMA3z5hAtsvCxHw
X4/q28433zSwak0UwqFkyNovSzP4bfIrksMYyly89sSEaUHy8F/7wuZP5yys7mFqYTrc3w
OrdXr7/mW9wwqtfYn79OkgrwXPqXeB3w7qHo4+lgBML4Ev/NtPtVZw/M4ccO3xmUi4X2Mw
YVLiyenvQwYNZ3yA7+wY6gzbsfRkUpRnJd5jo9AbIcVmYVJyL4aC6cIFgiH99VNWv/2v3d
TAux8XOpgN5St2HbDkjxKeVfAGRP3hMSdUHO3FcDovGZBPa8Dh7G9r587rZXF7gRHk7jny
yEAP4aFczP3y5kgfPtqPN81ENG+gZczZ/2RiMOS4Y0rzrruCDXSn4duyjNNPO8+vD0xI5K
VZ+eP4UCnc8RGf5yXxmQgZuX4CZJelcNLuIVHv/4nwiOby7qzMaDO33wMd1XDK7hD+9s4s
Mh5nL5pawscvKrvn3UmBop127GGPppN8YTGohTaFUKrKNOK0mnlxdw97WDOo6vKw/qIDtm
Gw7H1d1MjbXRW27hw93b+GvV89j/sSSQx8xyqDUgNvG5KbaWLvvwOur2agYL9XUxSBLuvF
z7yAERABGg8LsaWJsOrgDUKmLZIRIHLOHP97Hr06EKf93KcqZjc6bhkGpAStGu4OKiUXpb
Kn7/+zdoMr/PgAAIABJREFUd3ic1ZX48e+dXjSj3qstyUWu2MY2xqYX0yGEhJYQlhJaEnY
32STsL9lsshCyaRuSkIQUAoTeQjMQIHRjMC64F7nIltXraDSafn9/zFiWLLmMLXlk6XyeZ
x577rzvzB0Yv3Pm3nPv8XD13FLuPKuSO8+q5CsnlVJhbEAXz+W8vzVS7/Fz/vR87jyrUjZ
dHOPMg9RvtJuNvLq+geauAFsavfz41Y1EoppJoQ3YunZT4tvALmMR9fO/Dyp+zbS62bLgZ
9RH3Ly6roFnVtTS4g2ydo+Hsv0+Y5b4Z/fJ5btZV9fJt86dREXOwT+HBoPiugXjsBj3XaN
TrCYump7P4ql5uG0mpha4eehf5jI5T7aMEENr7I0QBSNDV5bA24TKmcK0LAOf+Iuo9B+gv
IcQBxPoIc9tZUJoI4T97LaUs/nUZ+joDnLDJBNfe3x176EOi5E0uxmb2cApE7Pp8AUpy3B
AuIcQJu5808uZk3OJRDtYPCWPwjSZJhOwsCKTR5bt6q2HpxQsnpLHpnoP88dnYFSKT3a2s
bvdx9Nbs/hgh4fSDBvfv8DEE5zLpIUzSNUedkcy6YmW4Gvtv8rrzEk5vLd1X700m9lAZba
TE8syqG7qJsNpZkpBKiWZhx7VmVWSxrO3ncSKmg7MBsWs0nQm57uZWZzGV08dj9NiktFOM
SzGXEDkDw/NCBEA3mYoSaPQZWBDcxb0DCygKMRBdbeyvdnD6ZNycepJfPal9by6uYsHXlp
JVMOJZWn87ppZvLWpCavJQHG6gz+8v4Tq/gvCm5+MNR3trUyEtrYsubU6wmJuW5eOqr8
5lamJrkNydGCq013zp3ItVNXoKRKJPz3IzLcvC7d7dR2x7b1+riGflUZqdwZk43zlOsRJW
JT9q72doSwJhfSjAaxWwy0NHSTVG6HZNBEY5qFpRnsqgik8L0WBFsh8VISYaD+pZWLpgxg
5Mrsg/Ru/6UUkwrTGNaYf/6fSajgIWw2dMBUTRqCYQig7dShtfC9jTKHEZeHV3GvR
sGZrnFWODt5nozg/I7WwmPXcmz9UWkZGm+P17++riLd/ZQX3nRn78uWnUtvv48aub+NzMQ
maXpNHuC5HusLB2T2zDu8I0Oz+7YjonSXVxsZ+l29v54/s7yHPbs7q7Wb6zrTcYAnj
xs3rOnpTNkh1eluyIMCfPyHUzTYyfl8uTq1vZ3tLN7NJ0ghHNjOJ0vn3eRMxGIw3tXrSvl
XuW1FCS4cAfitDZE+Kxq8uT+I6FSNyYCogC4dgvHINKrNr9oILe2BpQs50SN2z1OdE9nQe
sUyJEP34PvPXfGFY9giMlj+2XvcG9b67nu+dPGnBobXsP7d1B7n5lE/E/ddORoryUCLN0huq
pWtDV5+eMkU0h0WclNt5Lhkw0Ux0N4VWQ0ePwALyrP458amAcdtaurm0bU9dPSE2NYMb+8
M8MTVTpbtaKUo3cGTy3dTnp3CVScauPuVTb3nPXdNKQ9fmsnzW8Ok1cUJFGR3xTRSGOF
2MqqdoXDGNLsI7ZAXkbYjsLK4XbqlAoWrz+oXluMfq1bIFVjwCw/rQHaO8OkJ1iZWKua8B
ntDTTwa42H95AmDZfkKauAC3eIB3dYb717FoeeG8H04rSJBgSB3TKhNio4gbSB3TKhNio4bgsJ5PzXGyq9
zB3XMaA4wrS7Hx+dhF3nFHBraeWNY1CKXR1BUm1mVta0s6A8k387u5XPSJ7v7BzteXNJGamc8
XJ8BphYpffOylOSpJz+L4MqZGiGJL7ocoodoTD4ji8h0RarwGEpstF2NV2NuCCdCuAhrsl
WQ6THzxRPjzBzv5t7MnUN3k5alPa8lwWvjWuRP55tOfxc6LaixmAmAwWpdr7y10+IROGk8QO
/2IToy+sP8+BX5vCP9Y34w1HOqcrFbTOxod7DjpbYDtMXzyigKN3Or97cSlNXAKfFyG2nV
WCzWynK8HFSeRYratpZX+dh/H4ryk6dkM2z6z08vCy2sOTsqlwOVKGsJxihyx+b7jXLRqF
iBBlTAVF3IIJ9qFaYddWDfd8XUa4DdvqszBmaZxejnDIYwZnNhkW/wWW38Pe1DaTazUzIT
eHJ5bs5oTiNH10yhRyXjVW72W72/GHopRkOOj2h5lW4OaRj3YxoyiN7kCY86blJ/vtiBEu123
jxoc/7S0A/PyqPfzu2ll8d/FE/OEoJqV2x9bRVNXAIDuYIRfvLmFx244kdfW7atJt
qWxi/uvOQGDihVnnVOaTlWBm55AmK+fWYFBKT7d2T7otXbtnk5++cZmVu/u5IxJajk
VObKfkBgZxlRA1NkTIsU6G+RG+5s7ZfQJTtMLGjQ/5hi8FFojr++TPS2NTMitZCTvzCa/ii6
QRCUdbWdrJyVwdWk4Fr5pWwtdFLAXLeNW04dj0EpfvzaJr52egWr/kVV5Kf
apVCrOKS3Nzf1BkN7tXqDLFlbz9JtrViMBr566g9rmt8wT6necNhPGHItxxR
gUAWxq9RKOana0+/hYvH3NOVe6AKbnadh9f+csntMan255ZUcu2Ji8PXn8iaY79CpoJkQR
jKiDq8AVJsQ7RCFHnHig7ufdunsvMtlBGLNF6KJK2xaixrdnLw0t38ubGi6ZkY9WcEp5N
nuiBt7a0EhDV4CV8V14A+Eof/lwJ988ZyLBSASL0UCXP8wjH9XQHYyQ57bR6AnwLwvmCm
FWsVhCob7B0MTc12sqGlj6bbW2OORKL/+ZzXfXjyRd7Y09zvWaen/NTGrJI0Pqlt5+KOa3
rYmj5/7rpzJRTMKMJsMlGenkLrfXkHbTwIiMSKMrYCoJ4RRpwhRjNe3JqR9M65A2
ile7j8lwm/hnNhVAPWGQzPBHT6QvyH8+sYUVNO2ajYlFlNpFoFIfNQIMnyOSCVJ54cf2A8
```

```
/yhMHmpNp74ZDeLp+bTHYygFHz11PEUptpZUJE5yKsJMbjTJ2Xz5sZGzpuWj9moMCjF86v
2DDiuOxAh02mhtTuIUvAvJ4/DajIwITeF6iYvp0/M4csLyrj+wU/6ndfZE8JgUMwbf+DPp
d0y8MeoyaCwDVUagxBHaUwFRJ2+EA7LELzlgAeiIbC6epvynAZqdA7a14aSgEjE1bT5WFE
TK+ny5M3zWba9jap8F3WdQe5dsomyLAeT8lx8vKOt33llmU7e3dzE186spNsf4vsXVjGzO
I0JeSmkWGWXXpEYl9XETYvGc/871QTCUb573iSq8t2829V/NKgwzcaX5pcS1hqL0UAgHMZ
lM/HtcyaS6jSzuaGLuvaeAQnTl84sJCvFetA+VOakcN60PF5duy8f6bbTKyjLlLIyYmQYU
wFRR08QxyC/UhLWvhNc+f2mxlIsCpPSNLe2kJNWdPSvIUYFi8mAQcFvrz4BDZRlOUixmfl
sWwuXnFBAS1eQCbkpbGzw4OmJ7RVz2sRsxmU5sZnzUFpT1xmgIsfJrFKZIhNHZltzN3cv2
dh7/3svrOfP181mYl5smwejwUCnL8C4LCcWk4Hd7T2kOyxkpjgJhCPc8MiK3nPLs1P47dW
z+MO72+jyh7lgWj6nT8rGcogVvGkOCz+4qIpLZxZS09rNhFwXM4vTZKWZGDHGVEDU3j1ES
dVtOyElZ0BzobmLHQ1t5MgGrSJuXJaTW04tJyvFSnt3CKvRwG/frubtzc29j5dk2Ll2Xin
l2SmkWE28+Fkd72xpoiTdwdaObn711laumF3E4qmymkwcmRW7BhaebvEGWbKmntqO2G7V5
03Nw+MP8/DSnYzLdvHu5mZW7e7gvitn9juvrqOHwjQb/33xFIKRKIVpdgrTD29UPNdt59w
pUn5DjExjKjRv9wWHJoeodSuk5A1ozjP3sL3Jc/TPL0YNq8nI4qos/KEwmxs81HsCvcEQw
I6WbjY2dLFiZxslGXaeX72bV9bWM7M4nV+8sRVvIDZqNKM47UAvIcQhjdtvWmpWSTpvbmz
sDYYAXl0Xq3xf3xng76v3sGp3LNHfaOi/SOTW08YzOc/NzJJ05o7LPOxgSIiRbkyNEHX4h
miEqGULVF02oLnQFmBDU2CQE8RYtqe1i7K8DCpyory2vmHA4+v2ePjWuRMwGRUrdnbwk8u
n4TQbue7kMn75xhZmFqexqFLqk4kjN7M4lWyXleb4HkNTCly8tq5xwHGNXX56QpHe+4VpN
jJTLPz4smk0evxU5KZwQnEaFkmEFqPQiAuIlFKLgV8BRuBPWut7h+q5O3qCpNiO8i37O8H
XCq6BI0SlzhCvtMmSe9Gf25XCtiYv33x6DTefOn7A44sqMslxWWjuCnD/tbPoDoQozLATi
moevP5EJuS4yHIdPGFViIPZ2eLjitlFWE0GohqyXRY6fCFeWlPf77gJOS6uX1jG25uamVG
cRmmGHaXhqnklSeq5EMfOiAqIlFJG4LfA2UAtsFwp9aLWesNQPH9bdxDX0Y4Q1a2CjHFgG
DjbWOZSbKm1Eo1qDAYJjATUtHj4eEcbNpMRfzhKXYefc6fk8vr62K/zaYVuLphRwIqadgr
T7Czf0cYJJekUZzgpzpDVN2JorNrdwXP7LbP/0SVT2dzYxZZGL0rBNfNKKEiz8fSK3WQ6L
VQ3ephblo5Fkp7FGDGiAiJgLlCttd4OoJR6ArgEOOqAKBiO0tYdJPMQS0MPafs7kFM16EM
pTidpRj+bG7uYnC+FDQU0dAbxh6KY4gH0MytqOak8kzvPqsRkMDC7NI1OX5BJeW58wQinT
cyhqiA1yb0Wo83EPNeAtvV1Hfz6qplsberGbjaS6TDjshpx2cxU5qbQ0OmnwePn7KrcJPR
YiGNvpAVEhcDuPvdrgXlD8cR7OnrITLEOSBBMSOtWaNkMky4c/HGbmxmmXfxzU6MERAKAY
ESTZjezvq6Tc6tyeX1DIx9ta+Xj7a388oszsRogJ8tJec7ALywhhsr0olTKMh3sbPUBkO4
wc8G0AlLMRqYUpBKNRGjyBinPdvKNMyupbe/hzElWKnJSZONEMWaMtIDosCilbgZuBigpO
by57d1tPnKPJg+juxnevhsmnA+mA2wzb09nZvQtnv9sHredVoGSEh6jUiKfv+IMG3/7uIb
ZpRkoNAsrszAaFEVpdjz+MK09EU4okxp44vAdyfUvqjU/umQqO1u7CUc147OcRKJR8tMd7
Gj14bCZOCk39iPO5bBSmSsBuhh7Rtrk8B6guM/9onhbP1rrB7TWc7TWc7Kzs/d/eFC72nx
kH2lAFPDA63dB4VwomHHg48x2php30RMI8o8NA1dwiNEhkc9fWZaLW04rp8MXZOm2Vtp9I
XLcNiwmA9OLUjm7KlcCZ5GQI7n+lWc68AUjrKvtZGOdhxZvgKI0GwZDrO5YfqosnRdipI0
QLQcqlVLjiAVCVwJXD8UTb2v2kn0k+UORIPzzfyCzEsYtPPixSmFw5fLl8ZpvP7uGFKuJk
ytkufRYd0JxOpNz3bR6/dgtRjJSpDq9OLby0p3kpTuZVZJKVEOO2yaBuBD7GVEBkdY6rJS
6A3id2LL7v2itB1a+PAIfbG3hS/NLD31g0AsrHoK2HZBaCK3VYE+DynMO74XchVRFt3L7a
Yv5xhOrOKcqj7sumDw0+x+J45bNYqRQVo2JJMt2yy7RQhzIiPuW1lovAZYM5XM2efzUd/o
Znz1IrkbYDzs/BF8zWFJg3fOxZfVlC2N5Q+NOheyJ/eqWHVRmBex8n6nTv8i9n5vOY5/Uc
MbP3uG6BaXkue20eAOEo5qJuS4WVmb1JixqrWnrDmI0KFLtZvn1JoQQQhxDIy4gGmpaa37
11lYWlGf2WWGmoasBdrwPG18AVwE4cyDcExsJyp0cOyxz4CZ6h5RZAZuXwJbXcE44l5sWl
VPd5GXpthY+3t6G227GoOD19Q3c+eRqpha6iUQ1mxq6UEA4qrGZDJxckcUpE7IpSLPjD0U
wGBTjs5yUZDh6gyV/KNK782y2yyqrQYQQQogjpLTWye7DUVFKNQM1gz2We81PKm1FU9wAs
9QWXMo34BhfSEV7wkSHsk82EwanWR8yYX1ZizPUk1JkQEcJdzb5o4HuCCiMjlSzObNIxra
Pka6VrzS0vfG7Acn7cS1a68UHOvdgn79BZAEtifbvOCXvdWgc7efveP//IP1Pnixg08E+f
6PNcR8QJUIp9anWek6y+7HXSOrPSOoLjLz+DIXR+J4ORN7ryDCS+3Y4pP/Jczz3/UiNtGX
3QgghhBDHnAREQgghhBjzxlpA9ECyO7CfkdSfkdQXGHn9GQqj8T0diLzXkWEk9+1wSP+T5
3ju+xEZUzlEQgghhBCDGWsjREIIIYQQA0hAJIQQQogxTwIiIYQQQox5x31AtHjxYg3ITW7
DdTso+fzJbZhvByWfP7kN821MOe4DopaW43UTUDEayOdPJJN8/oQYOsd9QCSEEEIIcbQkI
BJCCCHEmDfqq92L4dMTDPPJjjYe+3gXDquJq+aWMKskDZNR4mwhhBhJWrwB3tvSzHMr9zA
538VlJxRSVZCa7G6NKBIQiSO2bHsbl/9ee/9Fz+r46mvzmd2aUYSeyWEEKIvrTVPfLKLn
```

```
/1jCwAfVLfw9Ipanrt1AeOzU5Lcu5FDfsqLIxKMRPjT+9v7tUWimtfXNyapR0IIIQZT1+n
nt29v69fW4Quxsb4rST0amSQgEkdMKTVIWxI6chz78ZKNtHcHk90NIcQopjjAtVmu1/1IQ
CSOiMVo5MZF4/q1mQyKc6vyktSj49Mf3tvO6+sbkt0NIcQolp9q4+tnVvZry3BaqMp3J6l
HI5PkEIkjNn98Jo/cMJenP92N02ri8llFzChOS3a3jhv+UASADfWeJPdECDGaKaW4Yk4RB
Wl2Xli1h0n5bi6cns+4LGeyuzaiSEAkjpjNbGRRZTaLKrOT3ZXjUpMnAMDynW1J7okQYrT
LdFq5eEYBF88oSHZXRiyZMhMiSZq6/GSnWGnuCiS7K0IIMeZJQCREkjR6ApRkOOjsCaH1m
CsbJIQQI4oEREIkSVOXn2yXBaNB4QtGkt0dIYQY0yQgEiJJGj1+3HYLLpuZdp8svRdCiGS
SgEiIJPH4wzgtRtw2Ex2+ULK7I4QQY5oEREIkSU8wgsVkIMVqkhEiIYRIsmENiJRSxUqpt
5VSG5RS65VS34i3Zyil3lBKY3/mR5vV0qp+5RS1UqpNUqpWcPZPyGSqScUwWoykGIz0S4
jREIIkVTDPUIUBv5da10FzAduV0pVAd8B3tJaVwJvxe8DnAdUxm83A78b5JkTSxESIjK
RYTHTJCJIQQSTWsAZHWul5rvTL+9y5gI1AIXAI8I8FD/sIeDS+N8vAR7WMcuANKVU/nD2URw
fdrf5WLenkzbv6NmzZ+8IkcNqor1bRoiE2KsnGGFzg4etjV0EwrICUxwbx2ynaqVUGXAC8
DGQq7Wujuz/UAOTG/14I7O5zWm28rb5PG0qpm4mNIFFSUjJsfRbJFwxHeXVdPf/v+XV0BcK
UZzv5vy/OZFpR8kqEDNXnzx/PIXJaTHT2yAiRODyj/fpX2+7jp69v5oXVdRgUfGl+KbedX
kGu25bsrolR7pgkVSulUoBngTu11v0v0KN+nYmEN5Aaz1HaozvKRoxmWxq7uPP
J1XQFwgBsa+7mW8+sSeoU01B9/vaOENktRjz+8BD2UIxmo/3699q6Bl5YXQdAVMNDH9Wwt
Lolyb0SY8GwB0RKKTOxYOhRrfVz8ebGvVNh8T+b4u17gOI+pxfF28QYtavNx/6bOG9q6KL
Rc/xPnflDsREiu9lIl1+mzIQIRaK8vKZ+QPvbm5sGOVqIoTXcq8wU8Gdgo9b6F30eehG4L
v7364AX+rR/Ob7abD7Q2WdqTYxB2S7roG2pjuO/LrE/FMVqMmK3GOmSESIhMBsNnFiWPqB
9RvHANiGG2nCPEJ0MfAk4Qym1On47H7gXOFsptRU4K34fAmwHagG/gjcNsz9EyPcpFwXN
y0a13vfbFT85PJp5LntSezV0PCHYyNEDosRrwREQgBwxexiitL3/fuenOfmjEmjb2pQjDz
D+jNba/0BoA7w8JmDHK+B24ezT+L44rKb+caZEzhvaj6t3QFKM51UZKcku1tDwh+KYDHGp
sy8AQmIhACYkOfiqa+exNbGLgwGxcRcFzmSUC2OgeN/3kGMeik2E7NKR9eQeSSqCUc0ZqO
KjRBJQCREr4I0OwVpx/8osDi+SOkOIZLAH4pgMxtQSmGXgEgIIZJOAiIhkiC25N4IgM1kx
B+KEI0mtPuEEEKIISQBkRBJ0BOMYDXH/vkZDAqryUh3UEaJhBAiWSQgEiJJ/H1GiCC25N4IgM1kx
2E0KIZJKASIgk2LtL9V5Oi0mW3gshRBJJQCREEvTE65jtZTdL+Q4hhEgmWXYvxoTW7gAb6
jw0dwUozXRQle/Gbknexz8QjmI29gmILEa6ZcpMDKOa1m421HkIRzWT8lxU5rqS3SUhRhQ
JiMSo1+kL8eMlm3hmRW1v2z2XTeOqucXEqssce8FwFJNh32vbzAbJIRLDprqpi2v//DENn
bEagClWE4/dNI/pRWlJ7pkQI4dMmYlRb0tTV79gCOB/XtnArjZfknoUK2JpMu4LiGS3ajG
c3t7U3BsMAXgDYR5aupOIbPUgRC8JiMSo19kzsJK8LxhJ6hRVMNJ/ysxmlikzMXy2t3QPa
Nvc2EUoEk1Cb4QYmSQgEqNeWaYDm7n/R31qgZvCJJYGCOw3ZWYxGWSVmRg2Z07KGdD2xTn
F2MzGQY4WYmySgEiMehU5Lv56/VwqcpwAnDIhi59/YQapDkvS+hQMD5wy65IRIjFMThyXz
n9fXIXbZsJqMnDHGRWcMyUv2d0SYkSRpGoxJswfn8nTX12Axx8iy2XFmcQVZhDPITL0X3b
f5R84tSfEUEi1W7huwTjOqcojojUFqXYMhuQsKBBipJKASIwZ6U4L6c7kjQr1FQxHMfb5Q
rJbjLR4Awc5Q4ijly8V5IU4IJkyEyIJBi67l1VmQgiRTDJCJIZMbbuPrY1eTEbFxFwXOW5
bsrs0YgXCUUzG/abMJCASYlBaa6qbvOxs9ZFmNzEh102qw5zsbolRRgIiMSQ21nv48l8+o
bkrNu0zrTCV31x9AqWZziT3bGQKRvqPEMlO1UIc2LLtrXzlweUEwrFtAj4/u4i7zp9EhtO
a5J6J0USmzMRRi0Q1j3xU0xsMAazd08nS6tYk9mpkC4Yjg+xDFElij4QYmdq6A/zn82t7g
yGAZ1bUsn6PJ4m9EqORBETiqPlDEVbUtA9o31DfmYTeHB8Cgyy7lxwiIQbq8ofZ3jJwV3l
ZhCCGmgRE4ogEQhGi8W3/nVYTF87IH3DMSeVZx7pbx41AOIq5z7J7h8WILygBkRD7y0qxs
rBi4LVkOKbj/aEIWks5k7FKAiKRkLqOHv78wQ4+//uP+N4L69hQFxsFunRmIedPi230ZjI
objl1PHPHZSSzqyNaaL8RIqvJQDAcJSylFITox2k18b0LJzO1wB27bzHyk89NY3KBa8heo
6alm1+9uYXP/34pP3ltE9VNXUP23OL4IUnV4rAFwxF+9041jyzbBcTyhF5d18Bzty2gLNP
Jz6+YyTfO7MZkMFCS6eiXIyP6C+y3MaNSCofFhDcQJi2JO2gLMRJNzHPztxvnUdfRg9Nqo
iTDgVJDs7FkZ0+I7z6/lqXbYjmP6/Z4+OemJh69cR7ZLlkpO5YM6zeWUuovSqkmpdS6Pm0
/UErtUUqtjt/O7/PYd5VS25RSm5VS55bxRRK3vGOy6/rN5LGzLsfQGmqAcTpjrLmtkcLcZQTKDo9xVRmxqK1R4tgTPyNvHHKMBD
6Dfe31l+BxYO0/1JrPTN+WwKglKoCrgSmxM+5XykllrQdHEKNB9Vsq3rddJCa43z5EEJsak
IBIiGPrQNcvo1Gua2PNsAZEWuv3gLbDPPwS4AmtdUBrvQOoBuYOW+dEworSHdx6Wnm/tnF
ZTibnu5PUo+PX/vsQgaw0EyIZyrKcXHZCYb+2k8ozqchOSVKPRLIcdg6RUupkYLXWulspd
S0wC/iV1rrmCF77mFmrtZGE0KL50UikTcl28vamJSfluTp+
YTYHUR0pYKBwdMK3osBjxBqTAqxDHktNi4j8WT2RhRSYfbW9jVkk6iyqzRkzdQ3HsJJJU/
```

```
TtghlJqBvDvwJ+Ah4FTE3zN3wE/AnT8z58D/5LIEyilbgZuBigpKUnw5cXRyHRaOX9aPud
PG7jMfqwYis9fcL9VZhDLwZIpM3Eocv0bevmpdi6fXczls4uT3RWRRIlMmYV1bIOGS4Dfa
K1/CyS87lFr3ai1jmito8Af2Tcttgfo+2ksircN9hwPaK3naK3nZGdnJ9oFIY7KUHz+gpG
BI0R2swRE4tDk+ifE8EgkIOpSSn0XuBZ4RSllABKurqeU6ju0cBmwdwXai8CVSimrUmocU
Al8kujzC3E8CEU05v1yiKwmg+QQCSFEkiQyZfZF4GrgBq11g1KqBPjpwU5QSj0OnAZkKaV
qgf8CTlNKzSQ2ZbYT+CqA1nq9UuopYAMQBm7XWktxJzEqDTZlZjMb8fRIDpEQQiTDYQdEW
usG4Bd97u8ilkN0sHOuGqT5zwc5/m7g7sPtkxDHq2Bk4LJ7h8UkAZEQQiTJIQMipVQXsdG
cQWmtZc21EAkKDbLs3mExsqcjmKQeCSHE2HbIgEhr7QJQSv0IqAceARRwDTB2lxoJcRRCg
yRVO60mOmSESAghkiKRpOqLtdb3a627tNYerfXviK04E0IkKBTRA3KInBbJIRJCiGRJJCD
qVkpdo5QyKqUMSqlrACn2IkSColFNJKox7lePyWmVHCIhhEiWRAKiq4EvAI3x2xXXxNiFEA
kLRWGHX/QtUSnFXIYRInsNaZRYZZFGGAAAgAElEQVQvsnqH1lqmyIQ4SqGIxmQY+FvEaTX
h8csIkRBCJMNhjRDF9wNaOMx9EWJMiNUxG1hJ22oyEI5qAmHZfksIIY61RDZmXKWUehF4m
j65Q1rr54a8V0KMYqFB9iACUErhsprw9ITJdhmT0DMhhBi7EgmIbEArcEafNg1IQCREAgI
HGCGC2LRZZ0+IbJf1GPdKCCHGtkR2qr5+ODsixFgR25Rx8NnqlHhAJIQQ4tg67FVmSqkip
dTzSqmm+O1ZpVTRcHZOiNFosD2I9nJYZS8iIYRIhkSW3T9IrCJ9Qfz2UrxNCJGAwcp27JV
iNdHRI+U7hBDiWEskhyhba903APqrUurOoe6QGHqNnX42NXYRiUSpzHVRnOFIdpfGtMEKu
+6VYjXR1i0jROL4F4lqtjZ2sbO1m0ynhYl5Ltx2S7K7JcQBJRIQtSqlrgUej9+/iliStRj
BdrZ0c8vfPmVTgxeA7BQrD//LXCYXSE3eZAmFDzxC5LCYaO+WESJx/Ht7cxO3PLKCcDRWG
/xL80v45rmTSLWbk9wzIQaXyJTZvxDbqbqBWJHXzwOSaD3C1HX08NbGRl5bV8+2Zi8fbWv
pDYYAmr0B/vZxDdH4RUoce6GIHlDYdS+3zUSbBETiONfg6eG7z67tDYYAHlm2i031niN6v
khUs6Guk5fX1PH+1mZavYGh6qoQvRIZIWrUWl88bD0RR21nSzc3PvQp1c2xAMhlNXH3ZVM
HHLe+rpNmr59Wb5BUu4XCdPux7uqYdtAcIpuJ+kbvoI8JMZL5gmF2t/li08Fa0zxI0NJyh
IHM+1ubueGhT4nEA6yzq3L58WVTyXLZjqrPQvSVSEC0TinVCLwfv32gte4cnm6JI/FhdUt
vMATQFQjz+voG7rlsGvWdPVhNRp5dsZtbT6vgqgeWsb3FR5rDzE8+N50zJ+ccMK9FDK1AO
HrAVWYum5k2n4wQiePLrtZu7l6ykdfXN2I2Km4+ZTw3nFzGnz/c2XuMQUFZljPh5271Bvh
/f1/XGwwBvLGhkWvnlXDqxP4BUUOnn/V1nXh6QpTnpFCV75brmjhsiexDVKGUKgEWARcAv
1VKdWitZw5b70RCtrd0D2jb3OgXX0GVvPK1hUz
Kl7yiYyEUiWI82CozCYjEcURrzTMra3l9fSMQmxL+7dq9d+PVDjN3XzaVM6fndTEmP/HZ
hxXws/fHQhT294zoL19v38nDZ09fO3xVSzf2Q7EArC/fOVETpuYcwTvSgwHpVQacLXW+v5
hfp1LgS1a6w2JnJfCTibyYGF4LyjMHtC2syGJlTewCEY5qfvPPrUzM7
X9RikQ1u9p8x6SP4uAbM7ptJjp8sspMHD+6/GFeWVM/oH317k4evWker35jEa98bREXTC/
AYkp8tCbbZePsybn92pSCcVkp/drW7unsDYYAohp+8NJ62rol32gESQNuO9yDVcyRDPFdC
lQlelIiL7QLuBN4VWt9ktb6Aq31jxN9QTF8Zpemc9f5k3FYjmv+peE3e0
9FA2y7D4rRUpFHCsHHSGKB0RaS9K7GLl8wTDr93SyoqadYDjCCSXpA46ZlOci1W5hcr77q
PIU7RYj3z5vEqdOyAYg02nhN1edwOT8/j/sBtvhfU97D76gFEseQe4FypVSq5VSv1RKvaW
UWqmUWquUugRAKVWmlNqslHRoYWAcUK6W+F2/7QCn1uFLqm/Fjy5VSi3l3SqdYSr1Vu9ylbDTwPe4t
QAuBn4af53yw+1cIjlEJxCreH+1Uuo7wFbgXa31nxN4DjGM0hwv4leNZxn1FtcHtlmDu9PEbfoa
/nl3T75jLZxUyf3wGD7y3vXdO/vbTK5iQl/hQtjgywYg+YFK11WTEYABfMILTmsg/TyGOj
aYuP798YwuPf7IbgKp8Nz+8ZArvbG6ixRubxppZnMb88QNHrI9URU4K9187i8ZOPw6Lkbz
UgQFWRU4KSkHf3xKXziwkR+oCjiTfAaZqrWcqpUyAQ2vtUUplAcuUUlXjh5xPPBlrfVrSqm/
pcDMwAzsBJYET/uAeWdxVWpdQ84H6H6t9fIoIfoM6XUNmAbsUNmAbsWmza4FTAQm
IRhClFN5AmBse/pSJuS7+49yJ/HXpTlq7g3zuhEJuXDSewjQ7L39tIbvbfGSlWJmQ5yJFv
nyPmdBBkqoB0uxmWr1BCYjEiLRiZ3tvMASwod7DkrX1PHvrArY1ebGYjEzISyFniFeAOS0
mxmenHPDxqvxU/nDtbL7/wnqauvxcMrOA20+vwGIyDmk/xJBRwD1KqVOAKFAI7J0brdFaL
4v//WTgBa21H/ArpV4C/QDAqFO5RDFa98A+mJdzKbU61nWr3+AIOd5Pz1x++KWtTUHQ
bQeyzEXUS9HpwDr9XYXHaTTyR+uzy4RLQBc9KF8buyyjDEfhgFSRr1GEx1NR5crW1I+sDCyDyMTCQYCUrTGNyzgBoEZmo+TCf5fyp7BGBkj
Bh51u4ZuLj4Hxsa+doZFVQVuDEbDWQmYQreYjJwzpQ8ZpWk0xOKkOO2YpVgaCS7BsgGZmu
tQ0qpncDeKHrgCqGBDMCQLuxK5CfoeVrr5gM9qJS6Tmv90BD0SRyl/NR9v8w8PWEe/XgXu
W4rBgV/XVrDVXNLuG5BKUXp8oWbDKGDlO4ASLWZZOM5MWJVDfJD6qTyTJasbeB/X9+MM57
zc9bk3KSMcmbJFNlI1gXszc9IBZriwdDpQOkBzvkQ+INS6sfEYpYLgQfiU207lFJV
rFhoula68/2e53DdthJ1QcLhuK+sX+DUuovSqkmpdS6Pm0ZSqk3lFJb4/+mx9uVUuo+pdQ2pdQapdSsPufcEH/8VqXUDX2OnxivhL4tfq6UBBvj5pZlcFJZ
```

```
C+iVtmtWoxQJ5alc9H0/N77ZZkOFk/J4z//vo7OnhB1nX6+8cRqVu3qSGIvxUiktW4FPoz
HBDOBOUqptcCXgU0HOGc5scLya4BXgbXA3mHKa4AblFKfEVv5fkm8/QngW0qpVcOVVH0og
13h/wr8Bni4T9t3gLe01vfGk7O/A3wbOI9YIlUlMA/4XfxPkSCb2cSX5peyoDyLtu4g/lC
EX76xhRsXjSMU1ViMBlbvbqfZ6yfXLbtUH2vBg9QyA3DJCJEYQbz+EJsbu2j1BinJcFCZ6
+Kez03jhoXjCISjFKbZuPHhFQPOe39rMwsrs5LQYzGSaa2vPozD9i+x8DOt9Q+UUg7gPeJ
J1VrrHcDiQV7jQ45g2f1QBkQDhhu01u8ppcr2a74EOC3+94eAd4gFRJcAD+vYeuNlSqk0p
VS+1nrgBhdjXHcwfNDVFgAWk7E3R+j1dQ1cOKOAn72+ube20BfnFA3yf0wcC8GD7EMEsRG
i5i4JiETyef0h7vtnNQ+8tx0Ak0Hxu2tnc3ZVLjPjS+0D4QiFaXYsJgOLKrMJRaIsWVtPf
pqU1RBD5gGlVBWxHKOHtNYrh+NFhnuEaDC5fYKcBvZllRcCu/scVxtvGxAQKaVuBm4GKCk
pOaLOHq+qm7z88OX1vLelhawUC//vgipcNiN2s4lJ+S4ynAPnz8uyHHz3+f6FFp/8tJYr5
hSTe4CAShzY0X7+QuHoAYu7AqTazWxrlnpmYnDH8vq3qaGrNxiC2Oau3352DWWZ89nT3oM
/HKEy18Vtp5Xz4NKd/P7dbVhNBq6dV8qCIVx2L8a2wxxVOmpDWeTlw0RPiI8GJTxOobV+Q
Gs9R2s9Jzs7O9HTj1s9wQg/eXUT721pAaDFG+Rfn1rNxnovV//pY77/wvpBp1qimkErqB9
pocWx7mg/f8FIFONBlt2n2s2DFsYUAo7t9W+wa0Rbd5B3Njfzlb8u55a/reSmhz7lg+oWX
l5TTySq8QUjPPD+9kHLbQgxkiWy7D6NWOJTWd/ztNZfj/95x2E+VePeqTClVD7QFG/fAxT
3Oa4o3ibimrv8vLGxsV/b5Dw3OS4LbruJl9fUc+WJxSys7H+RzHVbqchxUt20byWjQUHJI
DtWi+EXPEi1e4A0h5kWmTITI0BxhgODiv2o2qsyJ4WN9R5mlaThtJpId5h58bOBmQ1Lt7d
yxn4lN4QYyRIZIVpCLBhaSyyhae8tUS8C18X/fh3wQp/2L8dXm80HOiV/CJo8ftbUdrCrt
Run1URRfAt8t83Ef14wmZJMB48s28VVJ5Zw+azCQVcnZTit/PILMynPjlWaTrWb+c3Vs6j
Mld2pkyGWVH3gf3ppdkvvjr9CJNOEXBe/uXoWqXYzAOXZTu46bxJTC924bGa6A2GqClI5d
8rAwGf8EVS2FyKZEskhsmmt/y2RJ1dKPU4sgTpLKVUL/BexWiZPKaVuAGqAL8QPXwKcD1Q
DPuD6RF7reBYKR/GFwqTaLf3aV9a0c8djK6nr9OOymvjfK6bzo0umcuPDn3L9yeP41Ztb8
QbCQGyztKvnljAhZ/CdXKcVpfHUV0+i0ePHbTfLHkRJFDxEDpHTaiQQjuAPRbCZZWM5kTx
mo4Hzp+UzvSgVT0+IXLeNNXs6uXvJpt7SPyt3dXDPZVPJc1tp8MRGNsdnO5lVGku69vSEs
JuNmA9S2NXbE6InHCF7iHe3FiIRiQREjyilbgJeBnrH87XWbQc6QWt91QEeOnOYzZ9Yqw
9GRXW7enkj+9vZ01tJxfNyOfzs4opyXTQ1OXn60+soq7TD0BXIMwdj63i1a8v5LEb57K1q
bs3GNrr2ZW13HLq+AO+VmaKNSk7yIr+Aoco3aGUIt1hobkrQLFMa4oRoCjdAfH6rWtqO3q
Dob0e/HAn3zx3IrvafBiUosMXoqUrwK/WbeGF1XVMKXBz8ynjmVaU1u+8cDjKu1ub+dP72
2nxBrl8VhGLp+ZSlnXgEh1CHA6l1GLgV4AR+JPW+t5DnZNIQBQEfgr8J/sSoTVw4cVVA
1rd186c8f0+6LVWm+761qtjV5+dkVM2nyBAYkJUaiml1tPjz+8KDV0C1Gg6ykPw4caoQIi
AVEXgmIxMijBllQbDUbeG1dA29ujKWEzilNp8Ub4OU1sayH7S3dvLu1mRduX8i4PlNpn+x
s46uPrOhdAXvva5sIhCN846wJx+CdiNFKKWUEfgucTWzF+nKl1Ita6w0HOy+RgOjfgQqtd
cuRd1P0tbXR2xsM7bVkXQN3nuUjzWFmXJaTxVPyMBgUJoPi5TX1ZLus1LT6yHXbyHVbafT
sS769el4JPcHIsX4bIkHBcBTzQZKqAVIdZpo8klgtki8UjrK9pZu27gCFaXYqc5w4LEZ8f
a41V55Ywo9e3vddc1J5Jr99u7rf83h6wmxt7OoXEK3d09lvOxCARz/exSUzCymTHKQxo+w
7r1wN3AOUALuAu3bee8FjR/GUc4FqrfV2AKXUE8T2OhyygGhvbo8YIpZB5tTNBgMmo4Gid
AffXjyRbz29hq5AGJNB8e/nTKAsy8medj/bmr1cPbeEjp4QjR4/k/PdrKxpx3CIL1qRfMF
D1DKDWMX7pi7/MeqREIPrCYZ5/JPd3L1kI5Goxm038esrT+DOsyqpafXR5Q8zpcBNabqdd
KeZhs5YEJ/utGAxGfCHov2eb/9r3mA5cilWExaTXMfGingw9Edg73B4KfDHsu+8wlEERYP
ta3jIyheJBETdwGql1Nv0zyH6egLPIfqYlOdielEqa2r3VY++7fRySjIc1Hf08L2/r8dpN
WE2GWjrDvK/r29mUWU204tT2dTg4YPqFna0dJPhtPD6+kauX1BGqVRIH/GCh8ghgthKwEa
PBEQiubY0evlhn5EfT0+Y7z63lnsvn86Lq+uIRDWlmQ4q89w8d+vJVDd5sZgMTMhJIRCK8
JPXNveeOznPzaS8/itbZxSnkum09Fsde/vp5RSkyXVsDLmHfcHQXo54+9GMEiUskYDo7/G
bGCI5bhs/u2IGS7e1srWxi5nFacwsTiMYjtLg6eHGRePY1tyNzWwgP9XG79/dTm27j6mF+
Vw+u4iTK7Jo6PSzanc1IyaqkkS4YiWI+yLJ7iOUQ1XVIQCSSq75z4OaKdZ1
+sl1WfnP1LEKRCGWZKb0ryArS9u18f/GMAibmuuiIrzIrSbcTjmie+nQ36+s6mVuWwdxxG
fzxy3NYuq2FFm+QkysymTsufcBrenqCrKjp4P2tzZRmOllUmcX4bEm8HiUOtN360WzDfkT
7Gh52QKS1fugIOiUOIhLVvLaunsc+3kVFTgrPr6rlrvOq+Ev9Dk6tzObe1zaxN3faaTFy+
+kVgGbZ9hbuWbKRtXs8LKrM4q7zJjMpXrdMjHyHM0KU7jSzrq7zoMcIMdzyByntU5hmp8n
j55tPr8HjD/GVk8u47qSyfsFQNKpZX+fhsY938d7WZsZlOfnaGRqB19bHNpd9aGkNX
zmpjO9eMKl3if6BvLC6ju+9sL73flG6ncdvmi+LDkaHXcSmyQZrP1LLgUql1DhigdCVwCH
Lfxz2xoxKqR1Kqe373468v2J3m49wVHPB9ALsFhPfv3AyualWzEYDr29o4K7zJ1MZ31eoO
```

```
xihOximyRPg5odXsKbWg9bw3pYW7nh8ZW/Jjj3tPmpauwlFogd7aZFEocjhrTKTKTORbA6
LkW+dOwFzPIBPc5j5wUVV/OCl9Vw8s4AbF41naXUr725p4oOtzXz7mc+4+5UNfFrTxlOf7
uadLc1ENWxr7uY/nlnbW3B6r4eX7WRHS/dgL92rvqOHn76+uV9bbXsPG+o8Q/tmRbLcxcD
8ZF+8/YhorcPAHcDrwEbgKa31+oOfldiU2Zw+f7cBVwAZiXRS7OMPhdnT0YPJoJic76Iyx
0m608ptj64kFIkNC/19dR13nT+ZHy/ZSFRDR3cIt82Ex99//6Hqpm7qOnp46bM6fvaPLfh
DEa6aW8ytp1X0+9UmRoZQRB/GCJFFKt6LpGvuCuDxhfjJ5dPx+MOkO8y0egNcNKOQBz/Yg
S8U4fypeeS6bVz75096z3v4oxruPKuydxk+xKaK919RFtUQjhx8s5Cw1gOSs/c+nzj+7bz
3gsfKvvMKDO0qM7TWS4ht+HzYEpkya92v6f+UUiuA7yfygiI2QvDk8t3814v7khX/9axKV
u/u7A2GALSGdzY3MasknRW72inOdOC2mQc8n9VkoK7Dzw9e2vd8jyzbRUGanVtPqxjeNyM
SFjqMHCKX1YQ/FJXdqkVSBSIR6j0B/vD+Z0CsBuJPPz+D+97a2nvMS2vqyUqxkp9qoz6+k
WwgHKXB4yfdYe63tUieu/9O1Isqsg65EKQg1c5XTi7jgff2TUg4LUYm5knpodEiHvwc0wT
qwSRS3HVWn7sGYiNGiYwwibidLd38zysb+7X9ffUe5o/LHHBsJKKZXpTKJTMLaPcF6Q7CF
+YU8dSntb3H/NdFVayoGbhh+DMr9nDNvFLc9oFBlEie2MaMBx8hUkqRmWKhvtPfb98WIY6
lYFjz4md1vfcznVY+HeRa89amJuaOy+CF1fuOtRiNpDksTC1MZU97D7mpNsqznXxpfgkfb
W/jvKl5XD67CNcgP/L6MhoU159cRq7bylPLa6nMTeHGReOZILUYxRBLJKD5Oft2qA4DO4l
Nm4kEtfuC/UaCAHa0+Pj24klEgWyXlWA4yt9X7eGOMypwWI0EQlECoVitn0tnFnHR9AIau
/wUpzuYUpjKC6sGJtBPzEvBZk6kfq8YbtGoJhzVGA9jv6isFAv1HT0SEImkCYb7T0t5/CH
SHRYm57s4a3IuBqVYu6eTaFST47LyjTMriWrN25uamF7kxm0z9QZLc8dl0O4L8oOLp9ITD
OO0mlDq8PYbyk+1c8PC8Vx5YjEW48HroglxpBIJiM4DLidW8X7veVcCPxziPo0Ku9t8rNv
TiT8UYWKei8n57t5//IVpDuaPyyDVYSE7xUKO24ZG47CYWLatlZo2H26bie9dWMXLa+p4Y
nktJoPixkXjuGnReDJTrCxyZfd7vQUVmZRnOdkWT1BMsZq4+ZRyLLIMf0QJRmKjQ4fzRZD
htLKnY+CyZyGOxI4WLxvqPIQimsn57t4pp+5AGLNR9V4rtNZ4A2HsZiOF6XaMBtVbuywQj
lKR7cTjT+d372wjHNXMH5/Bv541gfvf2ca7W5oxGRRfPqmUcCSKUoqFlVmYjQZW1rRzxZx
ijAZFyiFGhQB8wTAGpfpNGTutMtothk+i+xB1ACsBWf5yEDtavFz/4HJ2tsYS5y1GA4/eO
I8Tx8Vy0Dt7QhSmO3BajWQ4LIBmemEaz62spaYtdo7HH+Y7z63hZ5+fQY7bhkLxQXULs0r
aOWdK3oDXHJeVwsM3zGNjvYdgJMqE3BQqcmRIeaQJHsYKs73SHGbqJCASQ2BLYxfX/vnj3
nIwDouRx2+az+aGLh5cuoMcl5XbTqsgP9XGMytrWbK2gdml6Vw+q4BvL57IA+/Fiq9OKXA
Rjmr+tmzfiuhl29t4ZU0dJRl2vn5mBQrFh9UtVOW7ae7y47KbCYajdPlDeP2hA3Wxl9cf5
v2tzfzhvW1YTAZuPa2Ck8ZnSi6dGHaJBERFWuvFw9aTUWT5jvbeYAhiX4L/99YW/vTlE2n
xBrj/7WrMJgMLK7L40wc7+GRHbE7+3Cm5fGFOMU99Gttx/PbTK3hkWQ0rd3WgFFw0vYAW7
4FXHhWm2ylMl1VlI9nhFHbdK9NpHVDgV4gj8c7mpn618WxmI+9uaeIXb8SSozfWd/FhdSv
/c+lU7nsrVoOsusnLO5ua+M55k/j5FTNAAVrzRp+VY3vlp9n5x4ZGVsWvVZfOKCDVYeb19
Y00ePxYTQZuXDSeQPjQK8M+3NbCrY+u7L3/yY7lPHrjPE6uyDrK/wpCHFwiE7FLlVLThq0
no0jDIPvH1LT68IfCbGnsosHjZ0eLl09r2nuDIYDX1zeS7bJiNRkoy3RQ3+Fn5a4OILbi7
MXP6qRW2XHucBKq98pMsciUmRgSfX+gAZw2MbvfwgyAcFSzrbkbe5+RmFMnZBOOav70wQ6
+++xaXl7TwKkT+k/Xl2Y62N3Ww6o+1yplUNz76qbea2EgHOW3b1cT1QdfYh+JRHlo6c4B7
S/1SewW4nAopf6ilGpSSq073HMSCYgWAiuUUpuVUmuUUmuVUmsS7+boN6ds4K6rV51YQl2
Hn1v/tpJPa9pJc1hYuq1lwHHVTV6KMxxMKUhlRU37gMe3NHQNS5/FsZHICFF2iuQQiaFx1
qScfvf9oQhOy8ApKLNR9QtaTp2YzY9e2cD7W1uo6/Tz9IpaHvt4F9cvKOs9Zl5ZBh/v6L8
rS0GanW3NAzdc3H8Ptf0ppXDbBk5cpMpKWZG4vwIJzWolmlQtDsPM4jT+74szuWfJRuwWI
xdNz+eyWYV8tL2VS08oJD/VRqrdxPbmbtbt8ex3bipTCmJFEFOsRrbvt4vrtMLUY/lWxBB
LJIcox22lvsNP5DBXpQlxIHPKMrj7sqn87PXNBMJRphS4uWh6Qb+pqVS7mYl5rn7TWp09Y
YrTHZyzMI9QJIrW8OTyXdx6WjnnTMkjEIpQmZvCfW9V9wuAWr0BCtPsAwL60kOU2jAYFNe
fPI43Njb1JnJbTQbOn5Y/FP8ZxEj1g9Sr2W9jRn7QebQbM76nlCpL5JxENmasSbRDY00oH
KWpy4/DYuKMSTk4rSY+2d5KfqodfyhChsPC6t3tPPWpF4D/vngK47OdbI9fSGaXpOG2mWn
pDmIyKsqzUyhKt/fmkZw2IZt54wfuVSSOH4lMmVlNRlw2E40ev+w4Lo6K227mmnmlnDU5l
0hUk59qY1NDF9+7YDKbGrpIsZmoynczvTiV+6+exbIdrVTlu8lPtTF3XAb/9+YWtI4lY3/
znAlYjQY+qmmlsyeM3WLkugWlLN3ewu622LWq1RvgnsumcuujK/EFIwB848xKJuYduubi7
NJ0nv7qSby9uQmL0cCpE7Plh+BoFguG/si+ivelwB/5QSpHGxQlSjZWHCI1rd3c/041z6+
sY0ZxKudMyePuPpsvnliWzqySdLY0envbfvjyBu67ciYaCIairNnTwd9X7+GK2cU8uqyG9
7a2cNXcEtIdFgwKzpqcI1+Mx7lAOIrpMEeIAHLdNna1+eT/uxgSufGdogPhCL/+51aWrG0
gP9WGLxjhwZ6d3HflTF5b28CGBg8b6z3ccXoFD364s/d8XzDCH97bztVzi/nlm7Hk69+/t
```

```
40HvjSbp285ie1N3VhMBsqzU0h3Wnjl64vY1dZNhsNKRY4Tu+XQXzkmo4FZpemHLPgqRo1
72BcM7eWIt0tAdLwJhiP8+p/VPLMilqQ4vSiN+97c2u+YGUWpLN/Zf4fXSFTz0md17Grro
cUb4OyqXK6aW4LLZuKtTc1ArCbQXlUFbqYUDvObEcMqFIliSmD6K9tlZVerj/kyMiiGkKc
nzPIdsRzFveU2ANbWdqIMcOGMAuo7egbNYWv0BOhbWkxr+O3b1Tx203wW7LcSbFyWUzYWF
YdSkmD7sJGAaAg0egI832enaKvJgD8cGyZWCn508RScVhN2i6l31dhe88szeX/rZrqDEd7
d0kxUa3Ldtn51gSBWQ0hGCY5/iSRVA+S6rWxr9h76QCESkGY3c8qELJ5due+6ZTYq5pVnc
v/b1byytoGyTAcXTh+Yu1OcYSfPbeVfz6okqmNB/ifb29jZ0k1dpx+L0cDEPFfvaNShtHQ
F2NzYhS8QpjwnhfHZKUP2PsVxYRexabLB2o8p2f98CER1bNt6iBUdNBsV91w2jUtnFvL5W
YV0+sP829OfEQhHmRvfnBFg8ZRcUiwmbjm1nDy3lRsWjuPvq+qobfPxyy/OJNNpAWIB1v9
+fjqVOXKhON4Fw4mNEBWlOdgkKwvFEDMZFWdOzmVCbuyaohTcdf4kfvGPzRSm2bnjjApOr
sji9+9u478uqsIaL5WRnWLlR5dM5YPqFnzBCBGtWV/XycKdPCUAACAASURBVNfPquTeVze
yencHS7e1cNdza6hpHbjKbH/1nT3865OrueZPH3PTIyu46NcfsHLXwNW1YlS7C/Dt1+aLt
x8xpdTjwEf8f/buO77t6lz8+OdoWcOW995J7OxBNiRhj5SyKZSUAj+gUNpSRmkpLb2dt+t
23FtuaQsUWvaGMi6zYY8kZJK9bcd7b22d3x+SHTu2Mywncqzn/Xr5RfzSVvtKj8I386JznP
AcmKqUqlVLXH+ocGSGK0JqyZlaXN/Pt0yfw3//ewU2njOehj8qoanVRkhHP3V+czzG1PrUd
ruP+DPZw9JZNbzyhhel4iKXYzX/rbp2Q5rfzs/Kk8vXofTquJry4sZFZBMq98ezE1baEdo
4tSHdKDaAw4klVmAHnJNp4KN+oUIlLBYJBgEFpdPn7xyhZOnJDK0mlZBTxFhNfmhNqDPv
K5zVkOa18bUkxRSl2Hrh6Lu1uHxkJcTR2eslLtvPIp+W4/QHOmpzJ3s2d2ExG
Vg2v4CNVW0UpjoIBIIYh7jm11e08uGu/e1HurwBfvfGNv5+lrKSb8tO0JfpoII7/
KbNmRnhO1K04pVQZ0AAHAr7Weq5RKAZ4mtF9aGXC51nnUfl2obXNR0dxNWWM3v3tzBz84
xI/v2Aa33lmAy5faLpsZ30nP3xhI+fNyOaxuaPn0CWU4rZ0zKYFZBMp/
saaLV5efGk8eRER5qzkmyyTTZGOP2BQ57lRmEimCbOj10efzyC0IMm8vr59M9TTy1ah8uX
4Bl8/M5ZWI6e3cf9Ucrn5oFRBaCXvpnDz+/uHe3vP
e2hJqKBvqYQRuX5B/fFzG+PR4Xv28in98XM+zJ+Ul8aU4ek7L7rzLrWxbQY2ttB51uud5jS
ij5OaYF1IOJ9hV3mta6b3fCu4DlWuvfKKXuCt/OhNVr
kpLrN1ZsM9ahuc/cmOD3S4i2cNC6VunYPF52Qy5uba3l5Qw0Aayta2N3Ywi+
naXj30t3dx48jiWb63n9U21zCtKYXpuIhXN3UyTvcrEKBCtEaJM4EUVKpQxAU9ord9QSn0
GPBNuoFQOXB6l+Ppp6PCQ5LBw65Icxje8zJ6JX+Ox5bt7//EDtHb72FjZhsNiZE5BEluq2
3n4k3IunZ3LQx+X4fYF6PT4eX9HA3+4bCYr94Y2fc1JsvLdZzfQ7vZDi4vTJqYj7YbGLpc
vgOMw9nM60JKSdO5+cSN3f3Ey9mGcL2Lbhso2ku1mlk7LwmIy8tbmWuo73NxxdinbaztwW
s1My3VS1eLmtY01TM5OZEt1I4FgkJl5Sfz+re0EdWgq7Xdfms6DH5X1PvfcwhQe/LSy3+d
hS7eP3fWd3P3Fyeyo7SQ9IY7pOUnc8ewGmrpCU3Bvbq6lqrWbs6ZkHBiuEFERlU9WrfUeY
OYgx5uAM459REMLBjXPrankv97czj8uzuKjpKtoqumkvt0z4LFtLg+ry7o5e2omdouBDre
vt0CwKNXOb97Yjj8QpCDRxLj5BRgNiq88sJJOjx8Ao0Hx9VPGY5Hl9mOWyxvAknjkI0Rp8
XFMynby3OpKrj6paOQDE2PaWaWJLCnNoKbNjT+oufOcUowGA7c8tb8QOi3ewq8vmY7JYOC
RT8uYmJVAVpKNu57f2LtCrdPj52evbOXCWTm8tbkOq8WA02airPHARsNQ3+Hh9U011IU/K
/9w2czeZKjHpqp2ur2BAecKEQ3yVfMQKpq7+dPynRgU+G1pfF7VweMry7nl9BL+87Wt/R6
7uMDO0vwgq5p9pCdYuWhGBpur2rhuURGbqtvo8vj5y1kO5pXdB2f+BIBnvn4iK/Y04fIFO
HF8KjNk6HhMG+6UGcAXp2eHdhpfUCDF1eKIJMQn8Jc3d7Cxqg0IdUC/YXFxv8csHJfKPct
3sjG8fP7zyjbufPZzrj6piHvf3dX7OJcvwEnjU1FK0eXxMz03kcXFifzHq/2X1M/OtfPPT
/Z/cRxs5azJoAa0AxhMm8vLZ2UtvLutnnFpDk6ZmM6EjITDfftCHBZJiA6hw+3DGwgyPSe
R8hYP7+2oBwUp8Wa+cco4sgytzLPXkhVvwuPowN6+m42WE8hNsnFukY1dnXFk2wOUdZq4I
rue0nevh6Ilvc8/JcfJlBznQSIQY4nbFzziZfc9SjMTSLaH6swumZ03wpGJsWx9ZVtvMgR
Q2eJiU3U7xWmO3kaLxWkOXv28pt95HR4/JYlBHjsHEgKt1BqyaY8vZvvuPVyY2IBZe6gJF
DDbXEbwlALu3+Al3qK4Y46JKcGNOK1x5KfYqWt34w8GObkkjQ927u/F++V5+Tgshx4Rf3F
tFT/t067koY/38uQNCymQ1WliBElCNIgOl4+VZc28vL6K82fmcGppBiXpduzWODISQl2n3
95Sz8/nukn7922oxh0A6MQ89AX3MjvFSUdLPcrv56z3vowh4Ob1wvuZmbgL1VUHs66I5ts
TURTJCBHAF2fk8Lf3d3PxCbkoaVYlDtOu+s4Bx7bUtFOUuj8h0loTZzLg8Qd7H3P2eDtfa
HqUuM/uBWCmwYT7wgeg/XGsG/4NwCx7Cpz+I67Z9P84r/QSzP4OnB+9iGfmNbxy5mwyyh6
hbdxUWixJbM1wMKcohbZuLymOONq7XOTYDz5lVt3q4g9v7eh3rKrVzZaaDkmIxIiScfdBv
L21jq89vJqXN9TQ2u3jukUFTMtLJMVu4bYzxuPxBThnchqplct7kyEA1VaJ2v4ak+wd5Gd
lYNvxEoaW3XQXn0OD10yqtwq+/BgUnBTFdyeiKdKEaGZeIm5foLeZnhCHY8EgqxNPn5jB/
KJkzp+RzUWzcihMdXD76f2n0e6Y4elNhgAI+rG++V2sGX2ahHY3E9y5HJILSf38PpxbngC
fi7a0Eyhc/Utsu14la/VvmbT8em6e0ERldS0f72qiramGZfZVODv3HDT2YFDjDQQHHPcHB
x4TIhIyQnSA5i4P//3vUJJzwYxsClNsdHoDNHf7aOnq4sGPm7hoVi7z86wYdm4fcL6qXsv
qBiOPr9nDV8efjHNKNdUTr+ZKRz7xWd+DuIF794jY4fYPf8oMQCnF6ZMyeXRFGSeOlyX44
vCMT7Nzy6nFFBrrMCvY5kqiKM1OXbuH5i4vqfEW2lw+ilPiePILBsrc8WRaXIwzNwx8su4
miOtfv2Oo24R/xlcwVa4GZaBl2rWUm4rJ6G7qfYxqKye17hN+1fUWXXnTcdZ8gmF7Bcz86
KCxZyfZuG5xEX99b3/iFB9nYmKm1BCJkSUJUR+dHj/bajq4dHYeEzPjsZpNWM1G3P4AW6r
bqW/3sLWmgxlJHpyrH4GMKbDt1X7PoUu/wJNra3ljSwNvbIHfXfp9Ug02Ti9Mi9K7EqNJp
CNENGxnUUEmt79YSXOXlxSHbIIpDq27tY7rfU+QuOpvoAOcO/UyNppv5kfv1DCvKJl9zS7
e3FzHPZdP50Tfak7c8yY6cwo651JQBtD7R2N0+iRU675+z++feB5v2b+IaclsvNrIa1U2r
rMlgbf/VJ22pWKuXUtS7drQgUW3guHgv4aMBsX/O7GILKeNZ1bvozQznmsXFVMiCZEYYZI
QhfkCQR5fUc6vX9/GudMyyU60srO+k0SbhcJUO81dXgpSbVwyO5ciUzPmTz+DtCKY8WXY9
BzoIHrqlwimT2Lrqv1LUN/Y0sB/XTojem9MjCoef3D4CVHtRnjrR8SnlnBC/k38a10V1x2
wUkiIwUzyfI61di3bF/2RgDJSWPU6xfVvc/tZF/PaxlqS7GbuXDqRpi4vTxu/wN6cc3Gag
5xqSsd22r0Uf/pDcLUQSC2l+uTfkVH/CXFGCwS8uIrPYUf2JTS54mkMFmFQkJsW4NPyTuY
60qArVESt47NoSp9P2uk/Ar8bTDbY+SZMufCQ8Wcm2rjmpCIum5uHxWiQVZbiqJCEKKy8q
QuPP8h3zy4lPSGOrERraKNBi5Flf1/JNScWsbGqjUc+rcCg4MoZP+LbwVfIaFoFi28HWyp
uZwFxlv1FigA2s5HU+LgovjMxmgxnc9deO96AiV+AihWcXOzhqc8qJCEShyWoFb+Iv5vH/
92G1nDmhBv5cVwV63bVsH5faPXZx7sa+d8rZmFy5uJI1RiVotKt+OEH6Xxj1iNkxblZ12L
jkac6ue/Kq2lasBCrIcAr5Wa2vNXOiePMPLO6svc1bzmtmLoFd5O562mak2eyr+BCzO3dp
L3zn/sDSy4C5+Cd+bs8fpSiXyNSaUoqjia5usLq2z3c/8Eeur1+bj2zlJV7m1lT3sK8whR
+8IVJ7Kzr5JPdofnwoIZHN7Qz/8wvcP6Wf0D1OvTc67Al5hMs+5j0+AU0dHpQCr66oDDK7
0yMJh5/8Ij3MgvRULMe5t0Ang6muNbS5prGpqo22fZAHNJqNY3HNpT33v73rk5m5JTS4tv
fJ8gX0GileHtrAyv3NjExK4FzpmYxryiZX3zUs+2HjxSHhQ6v5rvv9Hzxc3PupERy7fsbC
lmMBhYWJfFu2xmo6WfgD2iKEmwssOyBxHxo2weZ0+DCP0NC/07VHS4f7+1o4K/v7ybOaOD
m0ydw0oRUbGb5dSWOrpi+wnyBIK3dPhJtJl7eUE2nx89Fs3J5c1MtW2pCzcnKm7oxGfPZU
```

```
Nk64PyP60ycn5QP3m7U+DMItFejjGbOn+ykI2jh0jn5zC5IPtZvS4xSvkAQrTXG4ezN0tU
Y2m3TlgzJ4zBUfMqSktN4clUFv7x4+sgHK8aUjyvcA469ucvV20YE4LSJGTy+opyPdoW++
JU1dbO2vJXfXjodt6ubySnwTkWQL88vZE9De+95SXYz35gagJRUtjX5cdpMnD/JyazqZ5h
x4reoaXOTYDOR5bQBmfC1d8DVDPGZYB/4+fjx7ka+/eS63tvXP7yax66fz+KS9BH8GxFio
JhNiHbVd/Dgh3v597Z6fnXxNPY2dmFQMCk7nlc/rybOZOD7SyeRYDWRbDdjNCi21vTvxDq
zIAUmfhuad8OGJzEWnwr58/ix63k482dgtg7+4iImdXn82C2m4fUPat4DzpxQu9+kfNjwO
KcsSeHul7bwH+dNwWqW7V7E0CblJMK6un7HTihIItVuYXqeE6NSZDmtfP+Fjf0e09DpIej
p5IG4P2Hau5FbJl5IdepXcce7mXu2xh00MkHvxq5ySW54m7/GvQ9drfDce7Qv/V+cVjMl1
gN2sk/IGDAq1CMQCPLwJ2UDjr/6eY0kROKoi8mEqK3by3ef3cD6faHNWNfva+W6RcWcNCG
Nhg4P3zptAjPzEvmPlzZT1eoC4A+Xz+T9HQ1UtoRuz8m1sajrbWjfBzvfAk8HzL0O3O0wc
5kkQ2KADrcfR9wwE5fWcogP/xIxW8GaRFqgjnHp8by1pY4LZuaMXKBizJmQmcjMvEQ2VO7
fuuOL07P5/vOfU9Ec+ky7dHYuF83K4V/rq/ud62jdhmlPqAmjdc395LaU0bD456RVvIHZ0
0JNzpno5BLSN/wfbPkXAO68RTSlzOZIe/ArpUi2D1w5mWQ3D/JoIUZWTCZE5c3dbKhs4/K
5+YxPd1CaGc/yrfW0dPsozYwnPSGOR1eU9yZDAHc9/zl/++pszPWbsAS6KGl9g9QV94E9F
aZeDE274dN70YvvQOXMiuK7E6NVh9s//KLQlgpw9PmGnJANzWUsmjCdZz7bJwmROKhOb4D
J2U5Om5SB1mA1KZ5YVdGbDAE8v7aKX188jcJUOyajAYOCsvoOSpue7PdccXveIvHku9hZe
Al+DOQEa4jrKqN69h1QcD5Bo4XPOjPYuRO+X3pkcRoMiv+3qIi3ttThD29yZjUbOHdadsR
/B0IcSkwmRFazkW+dOoG3t9QxIy8RfyD0Dy/Fbu5tW7+lur3fOb6AZnVZKze3P4Nj69P77
7Alh3p0lJ4NFifKKjVDYnAdbh/2w9i3aVBtFZA+cf/t+Axo3s3cWSfzz4/30trtJWmQb9Z
CAKwua2H5tnrOm5GNyWDA64cVuwd2O69td/PYigqaurwoBXeeXYKj+YCd7M02XFiZ2LIcg
6eNztwlPFyWwd7WDv5vowJ8QBVP3LBg0FjaXF7aXX5S4i04BvmCMKcwhWdvOpEPdzZiNhp
YUpImCwfEMRGTCVFRqp2CZBs/v3AKG6vaibeaWFySxr5mFx1uH8+s3sdtZ5Xy53d29RslK
kixYavp35CMk+8ER0ZoyizohXiZ5xaD63D7sQ2r1kdDRw04+jT3jM+C2g1YzUam5yXy1pY
6Lp87+PJlIdLiLVw2J48nVlXg8QW5Yl4+N51STGOnD7NJYVSK3Q1dWEwGmru9AGgNv31zJ
/P+33eZu/Gx3udqmHsHzcZ0znx3InEmIw2dXfzvslIykt2sLm8hPs7Ed8+ZyOyCpAFxrC5
r5qcvb2ZzTTuLx6dx9xcnMym7/8Sa0aA4oSCZE2RBijjGYi4haun2snFfC3mpdm56bC2nl
KazuaqNF8Pz5jazkTuXTuQXr27htjNL+dVrW1EqNL8+M8mFofBEgiVn0e0De7wTgyUelv8
01KCxYTv4PXDCldF9k2JU6vT4sQ1nhMjVBiiw9NnIMiETtlUAcEJ+Mm9tloRIDC0n0catb
6zvvf2PT8p44Oo53PPO57S5fACcOC6FWXmJaN3/3H1disST7yHBVU1jwiTaU2fgbnPT7vY
DfgAe+GAP/7x2Hl+YnovZqAYdrSxr7OLaf3xGhyd0zoe7QqvJnrpxofRqE6NCzCVEr6yvY
nJWAvd9uJeWbh+nTUrn1qc29N7v8gV4bEUFZ03JxGEx8j9fnoVBgdUQZPLbl0HrHj4//TH
+sC2BB0/uxtxegVpyB6x9GHb9Gxq3w4wrwCirfkR/HW7f8EaI2isHjjzaksDXDd5OZuUn8
ciKMjz+AHEmue7EQB/uaux3e+G4FB7+pKw3GQL4dE8ziyekYVChXmsQGq1JdcSx2ncy7aY
gmQ4j420m7ny2/+7z3b4ALp+f3OShd58vb+rqTYZ67KzvpKrFJQmRGBViqv95WWMnVrORZ
pePiZnxPH3jQmrbPAMet7uhk7xkO2ajgdueXs+TK3Yz3uGieur1LF/0BDe8Y2DZvBws/i7
U6r/DM1eFkiGA9MmSDIlBdXj8WM3D+CfXVgX2AxIiZQhNm7WU47SZyU2ys6asZWQCFWNOf
rKt3+3CVAfb6zoHPM7tD1KQYgfAbjFyx1mlxFvNTE30siDLwARTI4HuVqr7lBIALJtfQGr
8wVfWJtgGrhSLMxlwxMXc93IxSsXUlej2BfAHNTaTgUUl6dS2u8lNsg143MJxKfgDAUpTT
Tz8lUkUJypqgon8z65puLwBfnp+MUtKM6C9Ezpr958Y54Q51xzDdySOJ+0u3/C67bZWhFY
zHighE1rLIHMq03OdvLejgZMmyCbCYqB5hcnccvoEDAaF1qGaIoCnP+tfE1mamcCtp0+gt
sNDit0CaJo7ulmy80/E1W2gu/QCdmWdxwNXz+XBj/bS5vJx6Zw8ZucnHbIXVklGPMvm5/P
kqv2veec5EylKG3pUSYhjKaYSoqCG5Vtq+eZpE9hZ30Vlq4uKpm6+ddoEHvpoLy5fgCnZC
Xzz1AmYjYomt4/xrSspeP1uApe+xt8vzsOQmLN/2sM2Fa59E+o2hlaaZU7rvxJIiD7aXf7
hrTJrLYOsQTYIdmSEGjYCM/KSeHxlOT88d3JkQYoxqaHTy/NrKqlqC3WsdlpN3HfVHJJsZ
uLCo5ZWkwFHnJFfL9/JuHQHZY3ddHn9/PPyYr7ruo4TJlp4YVMbm94u46kbM7nznFK0Vti
tRiZmHrrjUILVzPfOmci507Kp63BTkGJnSk7i8Dq3C3EUxExCtK+5G6PSfP3UCfzh7Z18s
rsJk0Fx2dw8GjrcXLeoiJLMeNZVtGJSYOuoYOHbV6C66sCWTLHdC2mDFK2mTQj9CHEIHW4
fSeHpiCPSUg4Tzhp43JkNe98HYHx6PLVtbmrb3GQlSlNQ0d/uhs7eZAig3e2nssXF2vIWP
itvwWxUXLmgkMJUP0unZvLRriZm5CdSkGKnqSvAKxvreCXcxHpyVgKTspzDavOQ4ohjSam
sxBWjU0zUEJU3dfGNR1bjCWhe3lDdu0mrP6h5ctU+8lPs/PX93TR2ejl7aia5hkZm/eu0U
DIEcMqdkDMziu9AjAXt7mGsMnO1QsAT6nd1oPjsUMNGHcRoUMzMT+Ld7fUjE6wYU2ra+tf
8TMpK4KNdjXxWHqo78wU0//ykjC5vkPe3N+KIM7FiTxN/eGsHTT5z7xTb1Gwnv710hvS8E
```

```
mNSTIwQba1p5xeXTOeNTXW8t71hwP1VLS5m5iUytzAZHQxi83bgn/8NjC17ULOWQfEpUYh
ajDUNHR6SBiksPajG7aHdwQfb/8xiCy3Fb6+GxDxm5iXx+sYals0vGJmAxZixYFwqz66p6
r09Mz+Jd7YOTJ6rWlzsbeqiozq0GsxqNpCU4ODVby+m3eUn0xlHoiRDYoyKiRGiJJuJf62
r4tPdTUzMShhwf36KnbuWTsLl87Opup2yQBp64bdQVz4T2pbDnhKFqMVY09DhGXSfpoOq3
RhKiIbizA1tLgzMLkhmTXkLbd2+oR8vYpJRKb62pBiHxYjFaCA/2cbU3IF1P4UpdiZnhz4
ji1Lt/PKi6TitRrISbZRmJUgyJMa0UZcQKaWWKqW2K6V2KaXuivT5Kho7sVpMvLejgXX7W
lk4LpVUx/5/1AvHpTA7P4mqVjcur5/i9HjmTCzGnCJN7sTICQQ1Ld1eEo90k8rKzyCtZOj
7ndnQEOoJY7MYmZGXyGubaiKIVIxFH+5s5I1NtXxlQSHXLy7m5Q3VnDM1i+Q+1+PJpWkUp
dn52pJifn3JdL57dinZiXFYhrMyUojj0Ki60pVSRuBe4CygEvhMKfWy1nrLcJ+z1eWjutX
F9NxEypu6+eNbO/jqwkJsFgPj0uKxWwyUNXURZzLS6fGzYmsjC8elArLyQYycxk4PCVYTJ
sMRfAdp3g3ebkjMG/oxzjwo/7j35uKSdJ5YWSHTZqKfmfmJvLCuigc+3NN7rKyxO7cqQwA
AIABJREFUi4eumUtZUzdmo4FEmwmtNW9ursFps9DQ7uHqk4oozRw4qi7EWDTaRojmA7u01
nu01l7gKeDCSJ6wy+NnW20H03MTyUu24fIFeODDPWysbMNiUhgMBqZkJ1Ccbuc7z2zkg50
NdLj9h35iIY5AXbv7yLvxbn4R8uaGmjAOJTEPWvZCIDRNNjMvidp2N5uq2iKIVow1OUk2F
hbvn/ofn+5gdkESjjgTyqDQhBqHpjji+M+LZnL94nH88pLpzC8epP+VEGPUqBohAnKBvp3
CKoEBWyYrpW4EbgQoKDj4N+Eku4VVe5opzUrgjMmZJNpMxMeZsBgNpIV/QTV1efnao2sIB
mFJSRrx1tH21yJGkyO5/nrUtrn7TU8cUvNuqFwDi287+OPMVnCkQ/MuSJ+M0aA4c3IG93+
wh3uWnXD4ryeOG8O5/gDyUuz8aEomSsHehi68/iBWU6jG0mQwMCEznixnqFFtnmUY7SGEO
M6NthGiw6K1vl9rPVdrPTc9/eA9LUoyE7hucTHNnR7S4i00dXqxmY1Mz03E7epmxe4mrns
4lAxNyIjn2kXFmI3H5V+LOEaO5PrrUdXqOvyC6tYKePdXMHEpmAd2Uh8gpRiq9m/cecakT
N7bUU9FU/fhvZ44rgzn+stLtDG3MJn6Dg9VLS5KMxMYl+6gIM3JqRMzWVyS3psMCRGrRtt
QSBXQt5o5L3xs2ExGA2dMziTDGUd1i4tF41PJSLCQlxIPpDBnnJ9TJ2Xg8QcpTnWQliCbD
IqRt2Ffa+8eUUNq3gOf3AMddTD+NMg5zBGetFLY/R7MWgaAI87E0qlZ/Of/beH+q+dGFrg
YEybnJmIyKXKT4ghqRYbTyuTsQ3eXFiKWjLaE6DOgRClVTCgRugL4SqRPajQoZuUnMyt/Y
HM7h9XEjLykSF9CiIPaUNnGjSePG/zOgA8qPoEVf4XSpTB7FhxJ8XXyOOh+LlRLlFwMwBe
n53D3vzby+IpyrlxYOALvQBzvSjKdlBzGFhtCxKpRlRBprf1KqZuBNwEj8JDWenOUwxIiI
m3dPmraXOQdsOM47VWw8j6o/TzUT2j21QdfUTYUgwEKF8Fnf4ezfg7KiMVk4PYzS/mvN7f
xzvZ6Ti1NZ1Z+MlNznBhk7yghhBhgVCVEAFrr14DXoh2HECPl0RVlzC9O3b/kXgdg97vw2
YNQvAQmXwCmCBveFZ4Ea/4Zqj1acBM40slJsvGbS2awYk8T721v4L4P9uD1BzljcgZzClO
YmJlAhjOOiuZu3thUy/p9rSTbLZw9JZMTx6diMRmoa3fj9gXJclrJT7GhBuuYLYQQY8CoS
4iEGCte3lDNL17ZTEOnlwscW3n/0SfA16fQOfUsaE+CLRUj84LmxVC5Fcr+GLptsoHVibI
mUjLuVEoyc6hudfHxriaeXLWv36np8RYmZTmpa3dz5/OfRxRGnMmAw2IizmzAZFQoFEGt8
Qc03V4/nR4/QR2aynZYjDhtZhKsZuwWIyaDQilQR9AHTKPROtT80uMP4vEHQs+vQs8VCGo
0YDYasJoNmI0GFKA1eANB3L4A/qDGZFDEmY3EGQ2hGPrE0fMa/qDG7Qvg9QdRCuJModG4o
eLue543HJvWYDEZsJpD7/e8mTlcJdOaQkSd0lpHO4aIKKUagPLDfHga0HgUwzlSoyme0RQ
LjJ54GrXWS4e682DXX/KZN+U655yXNUXv0iXGWgXgD6JbPSrQ7SdwlOLFqFCpVm2ymvavI
n27wtje6FIRNdgyOpLNpqRsmzIaVcDV4Qu62v3KaFZGe5LFYHWYMBiVMhze5rVaB1EH668
UQ1x7VrfUP/vTPUPcPezrL2y0/DsaLok/etKAbQe7/saa4z4hOhJKqdVa61Gz7GY0xTOaY
oHRF89IGIvvaSjyXkeH0Rzb4ZD4o+d4jn245CuaEEIIIWKeJERCCCGEiHmxlhDdH+0ADjC
a4hlNscDoi2ckjMX3NBR5r6PDaI7tcEj80XM8xz4sMVVDJIQQQggxmFgbIRJCCCGEGEASI
iGEEELEvOM+IVq6dKkG5Ed+jtbPQcn1Jz9H+eeg5PqTn6P8E1OO+4SosfF47XklxgK5/kQ
0yfUnxMg57hMiIYQQQohISUIkhBBCiJgnm7uKiGyv7WBbbTsmg2JKjpPitPhohyTEqNPp
rGpqp3K1m4ynVam5SSS7LBEOywhRB+SEIlh27CvlWUPrKDbG9qnNMMZx+PXL6AkMyHKkQk
xevgCQR5bWcFvXt/We+yaEwv53tJJxMfJR7AQo4VMmYlhCQQ1//ykrDcZAqhv9/DOtvooR
iXE6FPW2MUf3tre79jDn5azq74jShEJIQYjCZEYFl8gyK76zgHHy5q6oxCNEKNXp8ePLzB
wBXOH2x+FaGDV3mZkhwIhBpKESAyL1Wxk2fz8AcfPnJwRhWiEGL3yk+0Up9n7HUu0mSlMd
UQlnq89/BmNnd6ovLYQo5kkRGLYzpySye1nlWC3GEm2m/nlRdOYV5Qc7bCEGFXSEuK498o
5LJ6QhlIwKz+Jf147j4IU+6FPPgo8/iDeQDAqry3EaCYVfWLYMhKs3HJ6CZfNycdoUGQ6r
dEOSYhRaUq2k/uumkNLtxen1YzTZo5KHFprvIEgXr8kREIcSBIiERGlFDlJtmiHIcSo54g
z4YjyqjJ/UKM1khAJMQiZMhNCiBjhCSdCkhAJMZAkREIIESN6EiFvoReyQRYsT+ESKZMhPiQJIQCSFEjOiZKpMaIiEGkoRICCFihNsnRdVCDEU
SIiGEiBE9U2VSVC3EQJIQCSFEjPDKsnshhhSVhEgpla+UelcptUVVKq
```

```
fXhn3OjEZ8QQoxFkhAJMbRotU31A3dordcqpRKANUqpt8P3/bfW+vdRiksIIcasnkTIIwm
REANEJSHSWtcANeE/dyiltgK50YhFCCFihUeW3QsxpKjXECmlioATgJXhQzcrpT5XSj2kl
Bp063Sl1I1KqdVKqdUNDQ3HKFIhQuT6E9EUyfUny+6FGFpUEyKlVDzwPHCb1rod+CswHph
FaATpD4Odp7W+X2s9V2s9Nz09/ZjFKwTI9SeiK5LrT2qIhBha1BIipZSZUDL0uNb6BQCtd
Z3WOqC1DgIPAPOjFZ8QQow1UkMkxNCitcpMAQ8CW7XWf+xzPLvPwy4GNh3r2IQQYqzy+IM
YFHh80odIiANFa5XZIuAqYKNSan342A+BZUqpWYAGyoCvRyc8IYQYezz+IDaLUUaIhBhEt
FaZfQSoQe567VjHIoQQscLrD2IzG6WGSIhBRH2VmRBCiGPD4w9gt5gt5hk2b0Qg5CESAghYoS
MEAkxNEmIhBAiRvTUEMkIkRADSUIkhBAxwusPYjUbpKhaiEFIQiSEEDHCH9TEmYz4JCESY
gBJiIQQIkYEgxqTQRHQOtqhCDHqSEIkhBAxwh/UmI0GAkFJiIQ4kCREQggRIwJaYzIqgpI
QCTGAJERCCBEjQlNmBpkyE2IQkhAJIUSMCE2ZKZkyE2IQkhAJIUSMCPQUVUtCJMQAkhAJI
USMCAQ1JimqFmJQkhAJIUSMCPRMmUk+JMQAESVESqlblVJOFfKgUmqtUurskQpOCCHEyAn
oUFG1rDITYqBIR4iu01q3A2cDycBVCSIC02ZSQ2REIOJNCFS4f+eCzyqtd7c5
5gQQohRJNDTmFGW3QsxQKQJ0Rql1FuEEqI3lVIJgGySI4QQo1AwKI0ZhRhKpAnR9cBdwDy
tdTdgAa491ElKqXyl1LtKqS1Kqc1KqVvDx1OUUm8rpXaG/5scYXxCCCHC/EGNWRozCjGoS
BMiDUwBbgnfdgDWwzjPD9yhtZ4CLAS+pZSaQii5Wq61LgGWh28LIYQYAUEtNURCDCXShOg
vwInAsvDtDuDeQ52kta7RWq8N/9pjUDCD4cf9jBwUYTxCSGECAvI5q5CDCnShGiB1
vpbgBtAa91CaNrssCmlioATgJVApta6JnxXLZA5xDk3KqVWK6VWNzQ0DDN0IYZHrj8RTZF
cfz2dqoMyZSbEAJEmRD6llJHQ1BlKqXSOoKhaKRUPPA/cFl6+30trrXue90Ba6/u11nO11
nPT09OHHbwQwyHXn4imSK6/0G73BoKy9EWIASJNiO4BXgQylFK/BD4CfnU4J6qUzRSoce
11i+ED9cppbLD92cD9RHGJ4QQIiwY1JgNioDWaBklEqIfUyQna60fV0qtqtAc4g1H/oIq311
kOdp5RSwIPAVq31H/vc9TJwDaHmjtcAL0USnxBCiP38QY3ZaMCgIKjBqA59jhDjRUQJkVJ
YrLW+N3zbqZRaoLVecYhTFwFXARuVUuvDx35IKBF6Ril1PbAbuDyS+IQQQuzXM0JkNChs2z
vHeaJCMSIgeESVEwF+B2X1udw5ybACt9UcM3dH6jAhjEkIIMYhgEIwGhcMIfOFDEW3f
oPhPRWusgkSdZQgghjgJ/MIhBhabKpDmjEP1FmhDtUUrdopQyh39uBfaMRGBCCCFGVlCDQ
YVHiAKSEAnRV6QJ0U3ASUAVUAksAG6MNCghhBAjLxAuqjYqJSNEQhwg0lVm9cAVIxSLEEK
IoyhUVE1vUbUQYr9IV5mlAzcARX2fS2t9XWRhCSGEGGk9K8sMShIiIQ4UaQH0S8CHwL+BQ
OThCCGEOFoCwdCye4NMmQkxQKQJkV1r/f0RiUQIIcRR1bcPUVBGiIToJ9Ki6leVUueOUCR
CCCGOGq11eJVZ6McvCZEQ/USaEN1KKClyK6XlVIdSqn2Q54lhBDimNNaozUoDEqhlBDimApNl4FS0phRiMFEu
sosYaQCEUIIcfQEwtNlQGjZvSREQvQT0QiRCvmqUuo/wrfzlVLzLzRyY0IYQQI6Vn2w5ARoi
EGESkU2Z/AU4EvhK+3QncG+FzCiGEGGGhbTvCCZEKFVgLIfaLdJXZAq31bKXUOgCtdYtSy
jICcYlRrLHTQ02rC6fNTEGKHaVkx2wxOjR3eqhudZNgNVGQKtdmX8EgGMJfgQ0GJUXVQhw
g0oTIp5QyAhp6GzUGI45KjFobK9u45am17G3sxmEx8vMLp3HejGzizMZohyZi3KaqNm59a
h27G7qwW4z87IKpXDAzR67NsIDW+6fMpIZIiAEinTK7B3gRyFBK/RL4CPh1xFGJUamt28t
dL3zO3sZuALq8Ae54ZgPb6zqiHJmIdW0uL3e/uJHnnX8S9jlSm6zqNlKm1yzthnAMFBEuSPEaysI3+OaNbEPJWR0dhZBpEvilomYCh4KvdTLuAnnABCE8yqRS1SaRdUOZgEiFBhLYyBxSsgrBu1S9KgkjKmWNzBLOWFAz4oTMxnZkPdfSOgtI+WmxBIp2IDvvhpzaCz6DbR2NnCfGJfHQNgmLcLWOYJQTIXRSOPMklNCIWW3BeQXLsGyQsV8f81wKnCYZTk5urmu75P1hTGJbSJPxQ7ZJZxgiHCeMS8JPM3ilYAgUZY/CRC8BDIJkhfQwhxRUaTx8chqXSbNvlBNUoSsoKpiypp54LrWvxyHvQDPd2zqDDQxKV1ZrNk3+QwkD7O2vdXoD3O5RzsZm8vURXOT2UmZM3RR1USwaLZ7yaTSfzbgJbixqpGZWSMrDATKsGdbDo8z1c4TDCZTTkE3Q8vB84NQ/CyDGB3vlv5pzimn9ekA8bbQqe+gvYr7EotatWnbrJeW8g+Z9NPBQXtmx1sTIvW22ZVlHRAoA8jEcgD7NV5tz+PFyWH54zZePXKEOsg6wXedNA2ok0r1TjnSXgXEDcq3LffZgfR1aUcRlzp9SDP1rWM0YOLSYNngpmwTd2ynBWNSZlnfX2vFfD1CgxRZw0Ob6Cfhg6OoFkmhE7vjFl9eBGxsbBmqZ6oxtzHrSl9PNIuFCFGJ2aZxmgalNH1cPm48NNx7nLuuJJmYeKkd2FHKYbtg8KF9efw/J2K8/1t/3UCH9z/wyDGlOtXqaHdzJkbC5tuP1gYlMqldX6Omh6V6rEl7XIbSkQH0NP2gj7tq3rMt+uoQvkSCCxIIdrVwk/tzDRKFagTJKm+c6BSLAbJHF2WaKt3zhjO4mx4LA5bYb0ko6mjVEXBWlnXBjbp1lIitcDBaXKfJTiG6nIc1FFwaSZM5yoMvl/HH2pSxlmsbQlVjmt7QFmj0hlK6xzU3Mrqsy4xpy0HpnyNLSmDaj9lSiN7HbjmmFxUHpVJj1QWWKJq0ILhT/YYPFQFLC2lBjKiE8JPFaWZdvaqNgQCm0W5LZl5Wq5tjuRw/p/5ZRqLdRPtOjkHYhXwV1IC93eNvyYn1ZtpR5qyqxwaMw2+1vZdtW0Kw7n6jiUE+TqnoFG+6AjXalmYzqBCv48QaSNVIFhKZsMaE3J1pbAJpAsGy+K/u/Lm9qDbwybnc2q2NvZUUcfpY6F+0tQAdLEsmdb4eTQD9crUCSUhqPIA6KdVEaHi6/2fkC6X+ZLnvz4QWg+nuq42nNVuvWVPVaGHIzMCeRPfoLzfdpW97GTOaKWV0EwvjLOKUy/pu+NISp1Uzc7Acc9QgO3Y4FeO+bo57aQvCC3CoOPCANjTcqsxVKZK9sS0YxzKAO8GsLpqbsMAhUymE9RtMN+3BwqC9QBPQpHlK7zMlNqNjKbTFv3VbbVbCHPqbT5C+zcf+yClw4pG1kYPhkI4zkrK0aSD+UgTOKEzAGZrK1zK2gfAj0g4GIbsCrLTRWAm5JNELCd2+Ie5fQt/8FmiMACWpeVbM3JK6rSgvMLMYEjDyJDmG4tbxyZAWKSm1ErcYBmBfRGrmnJzHT5GDcMRRMAXGGB3mNHs+jnzEWF1GO1Yae9VIyk2AOMzt93zUwtfXK3UawzmN+sqGHHvXz4fB0EfBnyhuxnLNd2unuzBKq4rJRHxqmDxqyZjsF0F/XV4X6T2nrHKGHOsD1J0sfHaQ0TqMlsYgSFJTMxVK0cqTjgJ/wfemZgRXUx4I0iTBfTsH8YkYDJm4f2HdpfWy7ZaFtALUq5h24Xvq1NdVnASvdSB0JHnKqBgkCVGnPIVU/M9ZTg5CvOFvaD/O7BbCZ+o6kz7Rx/n0iVE6Gg2PF7KY+ZXG+SVIMCjcFCJk4OXZw9nwLPm7KMvFZqTS0fi9A/CJ1Sf8xGPvbKeYuD0t0RB6U0+O0Ynyd+KhWx8+O7jj+Gr1D9bwZ9T7Yfe5jWXHYaQdnPwZ9nQ1/gXAxoVQZsMzoQ98TWGq9O3ZCiePKNvmXjM2zz1ZzwWOuwIy9ZNvmR3hTxEjl+bHSLiF2nSgEbZzVCmqgzZ4EXYGAkSjf3/Q6vh1pc8m6Oo3ICb9XMIdDT8+aS8wZfE6LE11VQSBbmMGdp0iKYz4ygy8pfAI9NR3Zb5B5Lvhl0+1MNF9bwGASALHPBtvIEomvYbYB1LvcYbzfTNsfx3Rsa5b5xgDJB8wwnnPTbM/WpVcU1SxqFVg5r2NI9nbMdKsp0Kp8b8Ey8lMn+Ymu2OgUOChfgDP6gT5mxP4XT68sAGE7j5nAn9+VrTXUONRNwyW8P8tjxW+CQmExBnHAp7UsTo4OkrU1V0gcG0v1rOK7ZzTRFoV2MrsL5S2k0O72u6ns+VT9Lb8GjqXr3VNxdUk/XdYcNhnngyJVRwDrCWS9Mw2n01Gz6o9OPf6nVGeprUpsM+mHcUpuWbNjEfVZTvI0mbGcwkOUcbhKe05VrNCD9yqDi0g5lQd4c6uGbhAj8nkIj+n2D5fSnpA6PRlhYDthHZN4X9I6Xsl4ZNQmN3YiqDrs/xv5EV6nKeyMNDWb3cCuXs4J/YRUoNhgWYEE2J8G2zTzlyKUFi0XYLwiGYJlDNf4YnP6c7bcrM4KwjjsN5zkrRe4N8TZg6Lb+X6Vnr8E7ZqvrBwTDyXMpOAGCJJm79czMzs1K/hWIGPMPsRE3y6H/7OaWbg0gCkH2EKjL1Ttlm1O/mC/iDshCh0Y0Ky0Zhkj0EY5PqJYm2vAGTDJcbZYZ2hqGH4Ymji9IGQkNcdexbpUYL2Dh1g9y8IPM+FgMh5uPM9/rZnF+BXgbT1tnJhf2URRgvDTPIqxCg5cAFYCVGGwq2nL7G0FmeOPQ6Y6ws7pAVRV1K2HSKpyMqPKJfdCk+IkGlJTDlwJ1xxpWXN+2ZS0BMlyNMgcZL8I3ZgGzJnkfJN0q4LHDvPYxEfmfVsj9b0d4iE2TxZAPwALqrIYhbsNx3Q9L7PXD5nk+J3cgs/2bT3tN1+Npn6WFm7fWl5YFrW2nh0GnG9DmsqF2WEfV1wzDyXmuLyLxlJ7dWSOVmQf3sVwt6lMLRXaDnEqRTmUZt1y37GhRHCMDSntHh0X/pdcTZyH5hB2tjZ0Jy9S8Hn8iHP4ZlwvbO1k3DfB8u6cqA0u63UdfgZWxqCaQl4x+Avy+D35H6DQ76SxI/VtIyqRMBKjZrDTQqzbUwEgsv8xNEY6qu0SK5tGB2WqRp5hvqK9Esk+iqFzdPvbKpbXD1Fh7e8Zc54DFpKGqN5c89Q22BwvUnFiCBPg8NrN6ZEvbV/TISWClSXnjBgGw2w5Ozr4l+ScHXqR4bFN5ZqyHH29qgDh7bWsHFsMnj1y2AS33WJoOjdVpCgBASlCJYf3XKYP8fLkC3+8PbwpUcEy6PY8pXhm5UjGn8ltZydDv2+nxbRP8ggTz8t6ZFSf0YEMdS8G7RixeKYupLpY7N5hvgqdP+6zW+I1tDbIgPuiwJOYeVPWbfBDw7kfCuHrTQEHOVwQCw6ROwAMd1Zph0Bj9hD4qIpS7tJ+WbR0Zq4RB5wGbaTBZtTC4B+7CvtqQdTQwXWtHFD7zs8SL3MLlw8JFsKCFZ2pfrUJBY3vqRdNvo8JK4dpQaTJPQCOqnTMi+0kLqi6xR7s/20kVKMEAvnyURYnDmfDEZ8KBD9F4MKJxQGA+gprrY7S4rGrlE4yc/COtUpFMKcmgcmRLP4bUJeSn5VC5T1rGIuFdrR1+yTT7yoUCm7jOhl8yIYhKpQ0e6Mg9qomRnjjUwBxEW5k0xtzoopzcm3bw7ezlGSBIqHmcRzW6SK7SNvkQaEx/bVZrGt4X/Vi3yjIeG8XaK4sEvkPE2EoZQjfoPn2NXkeU3yCFpj8iqDFjmRCKnqE6hbSBhKh4NeKqKSoSSqCtBoXBBsjUsaPjJLPEkq8/UZcOKeZ3ENrUgzY3wEq8vMeJ4JT84XO13M+ndzxOkyzHFwBhVLXV0kY0nnpXzQPLRa8AQ0V5aQVRWHlSyv3QUDPGm2H1ubXmaDAsDsJYrPG0VR9cvVjyfdvBy2F6NxW2FGyIqn+tEGcFhQG0TlfRPZTeYmBUL3nXvD2qEgoPGcjT9M1mMz3zGM3ODPGYzzZkxlKj3YgrPW3C6XPcQRfrOo61JhSLVgzy3yIhwDQ2FdXNbgVz0pAGiyIiE5jEqO3t2L7OU8pOmN4JPC/TZyiY8Sp5uCLzXCAaO7m7h/ttWgz2Mrbtbpm9R7HMDK1OjmeCUw2x3FrCI6Fg+BVOqwcAsLdaJVqO+P43F8M6+2iwDh1OCb8TcCfsQOtDeWr0xSLy0qYpmQkw+DhKqxdTD3rSHdpwB4FHmQe5Ijc6rUz1xwZuwv0Sww3gjKtCTFG0kUV1SLhkuhGUEg1aKKfJg1CTlc9o9Cl4YEhkQPi0Ge/FGQgO66KWdyw00sB3Rbnhr3nrsEQKqoKF80o6X2+tgE5d8YKSRCuxDCbkVMuzrFuh2E9WpNBvzlS+ID8WPKlOB+jdBQd2Xc0D9NPJqXNuwyEP5w5gvCnuSjYWqN/vLuJOHFtHnRW5Y8OZXtJSVIoXXcnomtvhxRvKk8dSQtPPfuHuI4JnBCD2zxZNJ4vDNETahGAUNDdVTzYiBmCW/2hEIgpyCoePDU04IPMRCwaEmcYGDyL91PQ8hDRgY2KRF6sExrG8rhPlxbFaLpMbpCJTtHQXqE+MM7Az9jQY2v8yGYjFYgEQQqDUpJ7NQMdLwR8iewixCSMbJxqWQfrCNJ2IaXq8PSKRgzzvCfvyZuESm0YRPV3SOWSBk1vbQYrRjntWfpBZDqYHbHXxlWLrC0QEmNK5IXAMpC0pn7dF5u4D9j0X3UdrFwoftZtjeDDqRhtM/qXpm3BgNEOjcSWmjw1FIMQgNSrKGiuwL0GMH/w1NbOddOjOiZXSpGI+zqJrFXeEg6QMgWf5x2OtpPHRnkcyfsLcuxJzjnNbKqf4z+y8AGzEqcFPzhB4i3g5xVI7fGWIeDYNehVZYMxGMa8zk6Qe4OAFC1wuGBv0o3eyS3UrOYIkNqRDXdUvKfT9dCOdWhhArqoH5rGRmTw8hFNf+KLMQgxv8XfIAbUrKiEtxTQN3W8Ym/fw+VGLYiIFl7vLEuwPmA0TqwGBmAvyEmmN1yAqgdNVUOUqtINCK0BiLlHMi3v+KsBBOn31cxtN0uE1vWxR6ER/XLMEA1dlTeVFTYPPTK8jzwCVg8w/zM2mtBCUOJP/TNxHl/cnl0A5KfCpFjJ/lOFWz0tzi+ZMu8gK4k28rCCXYEdZbAbUUYE7LGLFqEOQ0kFz1R8m23sZlqNTiKQp7VyjeWqzc0EmPR2IEtHA/ZdPKpkMPpbHyYe+MxWbuSlNlaFbHYy1yEqRSQEZt+eYoCAhNwVkAAQgrV4KRVBSCAMU9xTTqhj8NmW+aLIKXkQ8KoOMK4GX+Bfy9M4T9iYm8qw6RSTCrThF/0Ki0Rn/Aq/ydSEqu7jNMzU8cEXuIDYjhHr3GBqwWf0dCX2DEiPVm4m7Nj6S9H/FaCiI6rOhVBKMB1Ud8W6T3lsaSpOCQyXqvnJ0wQmybQrFBVxOiR8xNNXnKVB14kVfp5E7vA5C1PFShwxaUP3BuBhQkK0UWWB0GV9XHmVeqxRlg7J+LIZKq5wPtsi4Ex97fh1JhXNdgO82s8D8fxAq9AJc0O47S3mnCpVU1kfwsJaC+Ogo37INiYCWxS+i1jVajDBmqpYkgUg4MR2pIL8RBDZeGtBqjEkQu8P/VTghT0pa+zn0p0f7Kx0EYrjN4SdwtDcOpuOg58cZGmO7f8MJQ9Hgt3cNFPTVi+Nh4l7wS9cKJi5knyEIyzQ70EyvL3nRg3qYJf9TmgG/3Kg1Xz7u1x4DeP/BCUVUrW0qdWrQIDCyUBZCZeKaoH9AEQrJg0GsaPhbv6NyZ7GgEH9Aq6b/aqlPHqPQyB8GCpN+I2kZqV3kjcuYJO+uvYYSUGhFzz1E2Lt/EC0kS1YZqxLdDXfqFb2dH3xKOd0sWX8hqpx7l/0Aet4+fC+2hISB3nkZpS4+rlLk7u9KhWqpGqJLx2U0ZJPVUTPRPwRj5C60SbEeq0p3WJlIh8pdUqJr1O4OsKtHjTOOZsq8hQUI9I0zDvBmBHP8jwxpEz7Q/JzNk3UU8mD5xAoD9y/aQkwYRKsVzNeHdVLSqoUWpX/7mp7wGiPm7P4LiqlhcaJNCu9dsTtpt/CxrLZq4cAmPLQsT7SF2jj6jyeYQeBIOhDjAb4UqcUkkvbWjW0MSn+9YJ4uIebePNSy6IbeEXl6B/M1BWbnBjMfKPi5YDpTkPwgTUlROABPjLXOSb6wPrqjLX+K8F5cMJq+CQvRqqtw3yO2KfrjYm6pmQYgEHUkZyTp+wTt8VFfnTt8gJUjjm/DrSIDhkXUQLgL5sVfv3JdV0IqAUpIFY0Ci29kE0J/tmUPzQVWFGSjZv+UhPmUQ2bP3bzV4pWOwTtpWHQH9KpNxqm6rywZb6EIgClTnbx7ODhbiGFg1qGY8g5c72Kc9P/LLE0Njpm9/OgwJtLD2K1bbqhQ6izqxm1Z4AvyPhrzl1FLfAHNrO3ZGXpgsr6HqLsRQhYSpqdxp3T0VS2pFC9+g1LjF1Z1kx6ItQFbKXtKYNS8KdUuw5fAwV1ftmVbqhIsaNyHE/Bd5T4G7nhJJZqdl9bP4Gks0G6k/S0vvgFXF+5KZ8Ip3FGt+O7uw9pV+e+Dp7VoUm3sRnsZsHGg1KjAj5WeFn9IKFdSeHFmOk4nAXWG/YEPDA/rImgr6vIPyrV5EgpT5ZtmVzD8f5RvIHPBhjjOgCmUyxxdK0GtX+qfV8H6h2RGSYRihe92SdR8s0nvDZbJRpJgrLlXfAoSFJuxglhsDuAP8ID/TRXdOxaU4pWMRK8aHkJORJcBwWBDI+YLNGfdCuyiR0eCk5ggy5+AmUNb4PoWMW/OOq9CqmQDs/eMoS9GTOJo5mVjdmeZyazlQgu9eP5zVeHU7xl9IVU+pfMbW4AOeqJQrxDpkXQDDbbkKGiZAsM/9ZCkOdnkNXAA7Z0oeFiI3WzlcKz3jmR7Q5DyfHUmh9qmeV5zFuOTeVUbt9nZrhbh5S/Cz+vaJdFzOFzaU1p/gsg1F3C7XgBjq5QqG6Vvx4UmCcRRcBO1OrtGXIp5PMnHIwH8EXHZjQf7x7MQbKxGkMMmJZwY6UDTJsOG12vjCvBFDXF4kPsWqJhJr2WpUqiE6hHwxB/R3aJkZeKvBcmCk9EiHMYIYLGVqRbmTvq1XqaPmTAT62GDLQ0JLS7jllXLm2G3pcbyMEvDAMwNDp9W+DrRyqIzI8g9NuBXBZnmMfRcnG/EE3HR+JdSY8ghAcW3nbTjBeQEEA86+LsNlN51X5X+EB6uKOZfzPDvBnDfnvIvXeYaEOV1WjyRAlS8IDhbk2z/SjMaA5ddx6glcq8/n+MuQ6CwspFLVTH2zHJwxD06pVIKrAzEN7Y+bmXaI3I2z51c6lIsGTQsYrxhyiF5AVjbWSK9nGe2dIyfCtK8S2N+S+JdVy/PxMtqsMDmu0J6LeEt/EfhVTsfQCylrJJZYdR+X8DOWzAUtt9dKYCzN4hKvtEQhF1DAbI5Yl2HqGzJ7XSuY6gsLYNahDvQtldU/lSBpnDUUWCa5CNg4HRAd74I7oJ6HRdJrlAU/rmHbp4XN4vxAjdtxkkwkVb4qzRXSJJKlIZNLUYkk3ucYv5Za86WgmP8hWoUeENEQx4a6YSuoYuZxdtTPzd+jjNDrnxm8Ac2SoQ+njZqQ==
OO9B3gSaAduizQoMTJq21wDjm2obKXd7YtCNCJWef0BPtndNOD42grZmedYCgSDffoQ0Tt
C1Nbt47/e2BbN0IQYFSJKiLTW3Vrru7XW84AFwG+11u6RCU1EarDpiFNK00lxWKIQjYhVF
pORpdOyBhxfUpIWhWhil//AKbNAqLqhrKmLF9dVHexUIWJCpKvMnlBKOZVSDmAjsEUp9b2
RCU1EanpuIt87ZyJmY+hDcHJ2At85ayJxZmOUIxOx5rwZ2ZwxKQMIFfR+eV4+i8anRjmq2
BIIaox9Vpn1jBB1uP10yDS6EBEXVU/RWrcrpa4EXgfuAtYAv4s4MhExp83MjSeP4+wpmbi
8AfJS7DI6JKKiMNXBn66YRXlTN0aDoijNjtUc6cePOBL+QP8RIl94hKjd7aPL4w8VXRvUw
Z5CiDEt0k8ks1LKDFwE/Flr7VNKSaXeKGI2GijJTIh2GEIQbzUzNTcx2mHErL4jRKZ+I0Q
+NNDp9eO0jsjWlEIclyItqr6P0L5jDuADpVQhocJqIYQQo4i/3273+xsztrtC02WdMm0mY
lykRdX3aK1ztdbnaq01UAGc1nO/UuqaSAMUQggRuUAw2NuY0WjoP2UGSB2RiHmRjhD1o0P
6/qu6dSSfXwjRn5YNOsVh8h+wdYc/EF527+pJiKQdh4htR7uqccxV6JU1dvH6plre217PG
ZMzOGdq1lHpttvl8bNybxOPraggyW7mygUFnJCffNCix2BQs7aihSdWVtDq8nHlggIWjkv
FEXf0i1e7PH5W7W3m8ZXlJFhNfGVBIbMLkntrFsTIau7y8NHOJp5ZvY/x6Q6+NDeP6blJI
/oagaBmXfh6anP5uHJhIQuKUyK6nho63Ly/o5EX1lYyNcfJxSfkMiXn6NQVtXZ7+XhXE09
9VkFBip3L5+YzM39k/46OJ313uzf2KaruTYg8MkIkYtvR/k05pr6+Nnd5ueOZ9aypaAVg5
d5m3tvewF+/OptE28iu3vpwZwM3Pba29/bL66t59qYTOaFg6G03NlS2csX9K3qLJd/ZVs9
9V83mnKnZIxrbYD7Z3cQNj6zeH++GGp79+onMLpRtQo6Gf62r5uevbgHgo12NvLC2iue/e
RKlI1hAf+D1tHxbPfddNYdzpg7sKXQ4gkHN4ysr+J9/7wRC18xzayp54ZuLKE4b+S8Vr2+
q4QcvbOq9/eK6Kp7/xklMznaO+GsdD/rWEPVddt/u8mE2KpkyEzFvRKfMBjGmhgf2NHT2J
kM9PtndxN6GriHOGJ5ur5+/vb+73zF/UPPBjoaDnvf+joYB+xP99b09dHuP7ged2xfg/g/
6xxsIat7ZVn9UXzdW1bW7+d93dvY71uHxs6W6bURf573t9QOup/ve343bFxjW81W3ugZc1
y3dPrbVjPw6jKZOD/cs39XvWLc3wMbKkf07Op4cuNu9v7eGyE9qfJxMmYmYd7QToo+P8vM
fU2qI9E4NdcewX0cN+pyHeh3DIPcb1MjHdyAFqEFyX5ktOzoUg/+/Hun/z4O9hjGS11CDX
ydH42vTUH9HY+sr2pHxB/bvdt93c9cOt480h0VGiETMi7RTdZJS6hal1B+VUvf0/PTcr7W
+OfIQR4/x6fGcOK5/d91TS9NHfLjfZjbyjVPG9ztmMRo4pTT9oOedUpqOxdj/f+k3Tp2A7
Sh3po4zG7nplHH9jpmNitPCnYnFyMpwWrn1zJJ+x5w2E1NGeCro1NL03i7nPb5+6nisw7y
ecpNsfOu0/td1enzcUZnCSomPG/B3FB9nYkYM90Hy99nLzNRn2X23N0CS3SIjRCLmRVpD9
BqwgtC2HcEjOVEp9RBwHlCvtZ4WPpYCPA0UEepvdLnWetTsAJlkt3Dn0olsrGrD4wtSlOZ
gUlYCTltkzczKG7vYXNOOPxBkYpaTiVkJLClJ4+Hr5vPcmn0k2cxcMjuPGXkH/zCfkZfI0
19fyAtrK2l1+fjSnHzmFx2bGp6TJqTy6PXzeXb1PpzWULwz80amgLWhw83m6naaOr0Upzm
YkuMc9i/l44nXH2BLdTu7G7pIdpiZmpNIptMKwAUzc0hPiOPFtVWMT3dw3sycYTfg3NPQy
daadoJaMznbyYSM0PPMyEvimRtP5IV1oevpsjn5zIvgelJKsWx+AQUpdl7eUM3kbCfnzci
m6CgsSgBYOjWTJNscXlhXRX6yjQtn5TIpwuSrucvD5qp26jo8FKTYmZrjPCaLFkZCvxGiP
svuvf4gjjgTbt8RfYQLMeZE+i/ZqrX+zjDP/SfwZ+CRPsfuApZrrX+jlLorfPv7kYU4ctb
va+XKv6+k2xuqochOtPLo9Qsies6ddR1c9dBKats8ADgsRp64YSEz85M4pTT9kKNCfSmlO
KEg+aCF10eL1WxiSUk6S0oOP97D0dTl4ccvbeb1TbW9x/5w2UwunZM3oq8zGi3fWs83n1h
Lz8r6k0vS+P1lM8lwWkmyW/jCtGy+MC2ygvltNe1c+feVNHV5AXBaTTx540Km5iRiMChOK
EzmhBEsjE+Nj+OCWblcMCt3xJ5zKE6bhbOnZnH2MIvAD9Th9vG7N7fz5Kp9vcd+fN4Urjm
p6LhYTTlg2X14hMgXCGIzG/H6JSESsS3SGqJHlVI3KKWylVIpPT+Hc6LW+gOg+YDDFwIPh
//8MKEtQUYFfyDIQx/t6U2GAGra3Lx/iELnQ3l/R0NvMgTQ5Q3wj4/39hY8xrrttR39kiG
An726mcqW7ihFdGzUtbv58Uub6dtm6IOdjWwd4QLk1zfV9iZDECqwfeazfQc5I3btqOvol
wwB/PaNbZQ3jeyiiqPF37cxo9rfh8jjD2K3GIddLC/EWBFpQuQltJHrp4Q2dV0DrD7oGQe
XqbWuCf+5Fsgc7EFKqRuVUquVUqsbGiJLSA6XNxBkV/3AD77yxsg+DPcOcv7O+s4Bq3tiV
btrYF1Du8vfLzE91o7F9efyBmjs8gw43jbCdR676jsHHNte10FQrr8BBis69viDdB3j/j3
Dvf78QU1PiWFohCj0pcsXCCVEMkIkYl2kCdEdwAStdZHWujj8M+6QZx2G8FYgg34qa63v1
1rP1VrPTU8f2SmaodgtJq6Ylz/geKSFw6cPcv4V8/NjokbmcBSnxQ8oFF9YnEJOkjVKER2
b6y/TaeXsyf2/DxgNivFp8SP6OufNGDjldtmcfNn1fBCFqQ7iD6gXKsmIJzfZdkzjGO71F
wj0nzILBDXBoMYX0FjNRjwyKi1iXKQJ0S5gJOcu6pRS2QDh/46qRjbnTMvk26dPwGo2kGg
z88uLpjHnCOor9jV3s7qsmbLGrt4tF+YUJPOT86eQaDNjNRu45fQJnD3l0DUPWmv2NnSyp
rxlTE8flWbG849r5zE+3YFScPaUTH5x8TTi48b2rtw2i5Hvf2ES58/IxqCgIMXGg9fMHVZ
RcDCo2d3QwZryFmpaXf3uWzgulf/44mQS4kzYLUa+e3Ypp0w8+C/ZyuZu3ttez2d7m+iMo
ZVJxWkO/nHtvN7VfIsnpHHPshNIccRFObLD4z9gLzN/QOMNBLEYDViMBhkhEjEv0qLqLmC
9UupdoHd8X2t9yzCf72XgGuA34f+frHHPxfX++R2pe6lyO9OrGHXdi8GaUsK03M5Vo6mN+PWG3gHpUJQX28ceDsJ+3YtXrXVrUxEvxlbLPoxbaCMLFmFGEGiL5ewbdufsEoeShNAz0kbXWWuvpbVnn3Yq9XX7mxqAPI31rzb7Bq5p8VkFs0IGhdj1MuPLxNndyu9SKvQsi5b85R2DuvMMizpNapajBCaVkYlYaBQ9CGSgcVhcKl7MuG+zCh+nDSx9GzsutvUtbfhU2fgKmZaGN5bvV+9BqbfHuJS91wRjc2W0uMX36oYuy7q4gjVxhpC4AF0gnSqt76nM5jr2x5XJIi1dnmAnIM3cQVWuoBQ8/PrNbnFfiJRlJ6n7r5HwyeZq1GmmFOIVCu31Df8hjJBY2hGeHuDiQRVdvGyiN6I9+h7gahqj6wwX56IGh3pBRq5q4VSvpF4Z2hOsttWk1sRpWdeyLexLxjxbBprfVLaeY9ArKYmGQJ7o/0TUtWhT3cLaJEQZCH5RTuFNgTNvS2gGp33uZ4n3gpWBEnFQbqkyhO3Lpd5STNQbRfD4S2pnlBZdO2bhCoD4mq9lhfJRu2bFGVRGV1tFbrtX/mxwMX9pgR7e5UZW5M+9f/yHxrhk23n6/rmF7WmOnf24nMV4ZQZ6EcmqZS7IbYHXBXxo1gAPd9M5t6OqMWnoQvJgQb9pZGlEXp1wSgHS6j+yD7tUNQqQKWqW4s8LCtS7Bz8DpE16IhJTXyhV5CxTi+O7N+q7GYbVtORO3GGKL1nHzyuLFHYGNNR4Ebf36ibgZ+rLaVXb6n2B9fvPRUxDdHNBn5EV2aB0qPfIqzUf8zAVVs3ZSFGeGFOnoa3SlTC1ggN8HI1SX5ZNpzCRU9NS3ziILpOx6xnrZdJCTrhNaBzhhffLXzHDIwtnd8Rc9c7Ph/EvtxGJwNqtzW1dBQYdZnuF0CBcWz2vCKYdRH8P51V8tJwVxxOYX4IpXTBqJIYJ3PnAm9sW1wEvaAAPMwNl5jrYQ4wftuZZrRvZSTHdK3ZnwXbrMT5gaFZOLYrpsjpxhC/Dd3aFOVhqgI0DGM5GJ23J9I5MTjiDLWtxLvn7FRq/A2/LDFBgoWDZmyiA4uS4c93Kj23RcHOm3aQrkLZ20TBLhUQZzf9htRs8fxlqp9eVkvptHcGRzhrDqZc5SSfIZqHgGYbILR0EGdMvpEB0WFDsupSeS/9F7tyHyQHNujh6F0xaXRKwHHnSeIPA4J5iC5d2ikr9pgK8QjXwjT2bd/QCBgKTvOZQQXYHtPJPOKWuU99VkXS4KdNoD0ai3p6zCIFsdxpDYNc5RXvbMZEHvqn/BtoiE6C7xNrs0UvXPJKlKS7LPUxTT4JEXxd5lVcqDl8F7z7ZHS5sAvgBdv9bwJ9X08BAGd82DB4x0o2j1XRDSvxF58fR4Fi7rmnPbUJa/c7tNBBv06ny2VJ0ufH7BfoqIc9U8lgBN98E7+aWgbCIB8d4ZCIkvQhjkfLbJtiBvOfCg9lGB0MfHnMaA4pmU3Cx1ScI1otImA9IZS0F2KKUGVAXvJxMDfVGSPgnxY6F94yOD6dZBzZc3TEhFZSgpn8S6OLnXnX7GXQqGgPBO8AzPGbLe87fmyxbYd5bhr0T4LYqJrEYLTqY3D+vkOPR4O9ZL7sTGGPU0cs1M82lMAdM3y+EZC4nHqlmj3e7sjT2oG+H0E4OVrB2jkq23PYK14aECyIZLs69C9oYG3EZGTH0S6DF7eTCUM3v2P7TG1Vm17+xSATHnblZ/ikQKzMLm53O72U4dBVYDxW4V0Ld4GXZjCqsfDl10HyKnLdkUqrQwjbxjlCZ8hBLFe3Cej6Lmm6ezFiE1FWq1uo6OQ7xJPXm1sQ0g7kXjTdAGBLB9sJXNtCPgtEUThBwxHfTO3WDnBqq3rVWI1W4+0KsThTUxwxCu1pqZoSK8pYbVdyfNIqBU39jRmM8ehmUDTkqZ1HAqYUUXjyqqdBr7I1GMzbwNuwJnmm9xWbvkT9rwLIhTy6BwJbK7DUhPiCFgIE5yvUkjlfHB7XS3dMFunFaWw1Ryzbj5K2sIz7FmCD4l+OxRaV8pqL3sEV28DRn3uhT/EoOh0ynbZCM0wOVhGbHbmLpTPrmjl9KmI2aNnV1EFA9RlvtLq8jCFcc76gzKO94bsPzaRnWKPpCUhE3WiZYr+PyJHdN/J+5nvZUNnvyQtKZaUPwxZcAOGjFBGrMFw+VbSdH1KmYYg9iZSIa1Ey1Zzg9fOeC9xrUcqBYk+OY0TcDzBaYC3uG8tr1yL1nQ7UyHRDhrEhFpyxzWzeqnY4RBSpk6JEL1Pu3c0fcxTyY3FHHIkYRx+J+TmGHx+zlOIvjOpGVObmOKtqzyHMJzoJwHWUM+wy67h6TSGHLUWXpf7A9kMmZ9GmmlR4QIvszTyXzJ2m3XQjRQdx4T6tAlNWglKyAfSUQZsI0rxJWZeyjOwnk/qStkBzSGsvWAijRtNHqphqUvaQcLu6BS1cojoJa5L5xYdm56YY8NxDPMSSl7EHD8MgoD4tsOYpxvIx4qSKaP3RsBsQNPJ6mEhckHjcutoKbsFuI6Hn9OaX9/lbYTzM6m6r7yrIKcbpyRlpuHqcrkk+Ou5hAtWbGMMUJbAhzRJmFxG0pOyFM7SR5BGeZ6oaVdD0rHC7wh4IQD6Uqft6BqSs3ABO/gNXOzAbAXVR9rF3Qw4iAaqJbBwzyXuXIx9cxl9TNH9Jcgm8FcZi6DBuTOkkIv5eCVhkFrRcCm3a3yTlD3/YF+VVHKRG67a3Q4b4uRzRjKJ8Ocwo6DPjF6sNnSj2UNAEVkrnvZ3KS4/u0O+VHfpbk6t7d6qk7ch2FQnNgeFPWd3z1DK1Y61QQ4GpslobhQb7xkqg9N3HrZWBwT7h4uPr20gVmVQV6w7OypDNt8ioQkDaZG7Vh+ZMtj2SHrHjKRnAuIIctHVR6qHvH2LDMRF1ExR3XUUNaF6FszBE7Y7zl3hCZe6XJ6dqDdlk6Ah6OJzCyA0xaxjXbNMGn2mTvxcXhWHbtK0kFJ4cNmJe6SGWXCNxNjUqvOvuwt23ZdTIJhzGXYf8
```

```
naXH7vFyO8vm8lppem8sbmWBz/cyyWzc3FazZw+Kb13JdFQfIEgr22s4a7nN+LyBUiym/n
LV2Zz0oS0SN/iqKOUYtGENJ676SQ6PX7S4uOwWWJj9Gxcejy/v3wmdy6dhMNiJCX+yH/xe
nwB/rW+mh+/tAmPP0havIW/fXUOc4tCpX7JDgvXLxnHudOz0YQWChysn9G6iha+++zn7G7
oxKDgqoWF3HDyOPKS7cN9m8eVeUUpPHnDAtrdflIcluNmhRkMXlTt8QcxGxUmo8LjlxoiE
dsiHSH6F/BL4BP21xCtOZwTlVJPEqo9mqiUqlRKXU8oETpLKbUTODN8e1QxGBS5yfYjSob
2NXf3JkMQ6vtx+9Pr2VzTzl0vbKSy1cU/Pi7jT8t38u0n19HYMbB2pK/dDZ1855kNuMJFk
K3dPm55ah3VB3z7H0uSHRbyU+wxkwz1iDMZyU+xDysZglA92l0vfI4n/O2/sdPL7c+sp6H
D3e9x2Uk2cpJsB02GOj0+7n13F7sbQnVHQQ0Pf1rO+gO6t491ifbQtXg8JUMQmjLr35gxi
NcfxGIyYJYRIiEiGyHSWj986EcNee6yIe46Y7jPOVrVd3h6k6EeHn+QqpaBCUxFs4uGTg9
pCUP/Aqxpdfc2VevR2OmlocNDTtKxrWcQo1tli6vfSjWAfc0uGju9pCccWR1WfbuHVWUHL
gyFvcfJKqtYFyqq7t+Y0eMPYDIaMBkVPr8U0ovYFlFCpJTayyCFzyNVWD1WpDosOCxGuvq
sjDIZ1KCjTJl1rV2NAAAgAElEQVTOOFIcB98oNiMhDqXo94su0WY+5Hki9mQNMv2akXDoa
2wwqQ4LM3KT+GhX4/9n77zD2yrPBf77tJflvfeMM52ETAJhhz1aSqFAoYPSXm4LnbfQcUs
HndB721ugpaVltuwNZRPCyF6O48R2vPeQrb2lc/+QoliREwjxin1+z6PH0qfvnPOd41PcUs
94ZN16UNjvcZSc6wXA41s5EEXWZ+aMuM7VSgS8ku8xkZjfH6zJbBiyPPk4F/gg8cryLmiq
GXT5GRtVk+SgkSaLfYqW9f4Q+y5ELahenG/jdFTVoVZHLrVII7vjUAublJfH98+bEeqQZN
EruuqImIYbI5QsyYD9kFarIMvHTS+bHfu1pVQruvKKGQvnGdELj8YcYsHtjFYSPiHMAPLb
IwzmALxhi0OGlx+rB6vYzYPfG4kGqckzcdn51rK+cQaPkrs8mytjRsLn9WJw+kg0abj6rg
qxR1ssLF+ZS8xEV1GWmB6FRFqKDdYh8wUhQtUohZJeZzKzneF1mlsOG/lcIsR347+PZ72R
jcwd4dW8v//f2AZQKwTfPquTsudkHaUlR/fAEI1DAXrsfh7e1M6wy881Kwq4ckCk2ORnxm
WdCCM6dn8PLN59Cr9VLpllLeaYJtVLB9aUsrYqk2GnnYn4XAXY23tg1
7/gwBuw4HLY+QiNC7/DNrGQXd0O3t7fT0Gqnk8vKaCh3851q0uZk5PEF04u4ZTKjCPK2JH
w+IOsbxzkztcbsRcNm68vxg2n3EQiRnmWdfUtctcbcPtCfHVtGRctyuyORL6+kedCJSauiOjfpmF1vMlNDJKg68vxg2n3EQiRnm
cnIwPG7zJaOeqkgYjE6sSINgfcPDPL9p/fEXn/rid387bplnD1vzLqQhMISmztcDHtC3PH
Kvpjr6n/eakaBxDfOSUzFVyoEFVlJsT5RB9GqlczPZG/sXdmO/g2vv3xzrMXT/+60MOnz87
jOL0KqVkaaYx9e5QWYa0G5x8fn7t2CLFqF8fGsnLYMu7r9+WXyfvF3/grd/BmffDv/+HpZ
FX+MPLTmEGY5V8x5y+tnX6+CmMyq46zz3dHtPH7jajaKStEUsq4ndnr6uwdHWrgCqM1gqo
G5FEVU5n6xvmszUMbqX2cHCjL6oQqRSKvDLdYhkZjnH6zK7C7gz+vglsBS44ngXNZlIkpR
Qjh/g2Z3dR9ymf8ROt81Pv92XELD68JZuBu3jk+11oN+Z0HDxxdqeGZ1NNhtpGXTFlKGDb
G0bjq8v5RyArfeBSgfuYZAk2s1LKM4w83p9f9y2vmCYUDhM86CL9uFPHvC8viGxDNhDG9s
mvTKzzPgQ5zIbZSFSKQVqpewyk5E5XoXofOB+4C3gA6AbuOp4FzWZCCEoErqrVa
rRqBTp14uXLMGnQqsfHSDZWinmSToVGNbtSz2c6Y/2fNUoF2tHVylU6MGZBOAjqiItKL4K
4/CHMukR5UwqBEKA/DlnMGsMVlmvWo1bKVaxPREYHVSvFwTpEoUgMkVJBICRnmcnMbsajD
tHFQABwRh8nXA7uVcsL45Qbo0bJxWO0NDhIutlAdZaeHLOO3FGZYgoB3zurNN7NcRzMyzO
zuDAlbuy28+dOeqsAmYmlKjspoYXLzWdVUDw6SF5nhrN/AlIIVAbIWURp9wuk4OQLJ5fEb
TsnO4khp58vrC6hLPOTB9qfWZ1JVpKGglQ9GqUCtVLw1dPKZIX8BCUYOmQhUihItBDJLjO
ZWc7xmjIKJEk6b1xWMoUsLkrlmf84mZ2dVhRhZ2dVhRCsKQwZcwWCS5fkF0dVvZ0WSnLNLIuc4TlF
5nZY03HFtKyIC8JjVaLJxBEr1Zh8/jZ3jbC+wcslGYYSYXWFORQdmoQNlgKMymFgvrGwZRKGB
FaTpDDi8uf4i1lZlUZidx99VL2NVppdfmZmCZxbCny8bcXDOhhUJg0k5blJ
aljBrjWddt4t2EQTyDE6XMyqSlMQa08Xl1YZrxIM2r45acWsrvLSsewi+psM4/16Qk
yFpzCvs/vI9/bTEaFE60U4GvFHraGCvn9Z2votXlJNagxalVIEuSl6Pjn5g5snmDk/16Qk
tCjrGvEzQcHhqjtslGRZSLVoEajUqJUgMsX4ptnz2FPt5Wq7CTm5ZqZk53EQqFxfd0hwbdGdwmeuMbHTUGmHJJHDzHo
Zx6lVmczJPvZ4oh6rh80tFnZ1WVlalMqjMqq0jSykLVrfVAIVbAk5eLv2zKLtdTAhAubZ/Tf0SR93IN5Iw
TllZGYyx6sQfSiEWChJ0p6Pnjq9mZeXzLyPCDzd0mrhFy/vZ01mRGSzNqs2mFGEvSUkJHSqmzm2
QqJHO/hl7dEvZ4Axx7vwcntnRzU9frI9tX5Ju4JEbVsbaHHzYbOEL/9jCmodEaqg2bmtlXnl
sj977fx53dbePzGVZRlmsgf1RZh2OXne0/VsrH5Uvj++dVoR/2C39Nl5Yq
/bIzFIt2z/gCPfHnljGzzcSKTk6wjJ/noPey2d9pIt+4l69WrIBSJOdJvugfjGY/w1ddAC
FhSlEKaUcvb+/q56YxI370/vt3E3e8c4LEbV8VadpUtq8lAqBCl6NWW
ZJn78fF3svdOrMrl2dRE3PHiaogGHzr+/a+3lM87zOqhGSBo2HzBLj9hb2x2KcHP2zn00vy+fmnF
mDUnHA5GScMwbA0qg5RpGHL1QQuRQiGi1asl2SUqM2s5XjPBKcB2IUSDEGKPEKJWCLFnvBc2HVCr
FBRkmtipRQE8fNVEKZTCFB4z0aNjYNXKRWkGjjMbt/W6+fMaqmO/pptOPS5bX95/V6+tLEqtn
dtMrEfNW2/t4Gxfz3wxoJ9FqoxdxzefjwwxPn5OJvnp2hWMfLxB08PqafuxmTOeNRdpupDzWsr2u+EnW8KmJSTFl8/Qu42fdd9
m+L0zg2dJg93P6QU9nXm0BeSaqzrFbJWurDNqchAsRQoW6wtd6L5GWlhd9g0JqX5Cnh
bbgw5fNT3OmgedJJp0lDq2xcJbgUIhyDohW1/o9rg4K7XGmi3uLjr9ca4u9fe6/LcdhF
```

```
9A/PmxjdMFpbyDM5tZhPrUkn0GHj7pue8I6mvodccoQwIu7ezilMoMHPmyjbSi+0esb+/r
jArPDEvz1vRYCcgDlCYXN7afL6iO/9emYMgRAKEBlz/MUpOpx+0N8cMBCRaYRlz/EkNPHl
tZhTipKJRiW+OeWjrh9HhhwxilDEAnaL0zTk2JQM+T0MdpYsL5xkAMD8R7xYZefXZ3H1rq
jedCZEAj+zM5uWgdPOG/7CcXhdYhCUiSGSB3NxVer5MBqmdnN8f4cO39cVnECEJLCBKM+d
gmBKpzYb0z4nQgphMsfUXjGapboi+4jjDRmp3C3P4Q2Gs801vZjFe07qFSFpUjg5Gic3sS
MIJsngJRYYFxmGhMKSwhA7bclvKfxW+MCs0PR1EdvIEQgGI69Z3PHy9tYsnQwazIkSYTCk
Z5XoVGplMExtvEd4030SIUnP7IgpcxxEQmqjjwXQqAUApc/hCpqEVIrFfiCYYyfrG2ejMw
Jz3FZiCRJah/rMV6Lm05kJemoyDKRYlDjD4Xp1FeDMj54Orz4GvqkFG5cW0pRupFrVxbHv
W/SqqiO1m/RKJVcuyr+fYDV5Wm8Ud+PVqVg3hhxTGWZJvJT4uOElpeksr/XwWlVmfGBuMC
6+YlumC+dUioHxp5gpJm0pOhVdJYmVrVoLfoMdd0RRakk3YDF6UcIKE43sqQ4lS2tEUvmN
YfJW1mGkcLDAvSXFqXSM+LFHwyTYdLE9cyrzDIltOlQKwWL8o+txlFphpGKrPiikzUFKZS
ky8VFJ5LRFiKIpN67fcGYhUguzigz2xHS4YV0TjCWLVsmbdu2bVz32T3ioWXQiVolcHv9F
Ot9dNsDqPVJDHvCKIVETrIOrX+EbieoCFJmCmDWquhyK+kNJZNq1GDWqVjfOMiT27o4fU4
G6+bn0jbkQq9WUp2TRLZZx+v1fTzwYTtatYLrVhWjVIhIMK0kUZllxOYN0TLoRKNSMjc3i
fJMEzs7rDy+tZNt7SOcUpFObrIOly/E5csKEm4qvkCIza0W7lnfjMsX5NpVxSwvTsUXlOg
ccZORpKUqKwnTGKnbMgAcNaBiIuQPIv+3xn4HFqudBSYH6pEDWFMX0OzUEvA6KdG5kEIBM
lUeVH4rTcpyOgKpCCFI1qvx+AOEJIHHH8SoVeMLhkk3acgwaanOSYoFbNf32nj4w3a2tI1
wZnUmayszCEmR+lxuf4gBh49gWKLf5mFtVRalGUbeaxrkoY3t5CTruG51MdXZSVhcfnptX
nKSdWQYtRwYdBIKS1Rmm2Ixc75AiKYBJz1WD0k6Fbs6rDy1o5uz5mZy1fKiuIQDmRjjJn+
n/uZtvnVOFbnR4PUvPbCVa1YWMez2c8VJhXzr8V088dXVFH3MSuYys4JZFVAm3wUPo7HPw
Zce3MKXTynjr++18ueLMrn5uQEuXlLC/S/tZdARcZX94rIF/M8bbViivc8W5er4w5I+8jI
L2NLay1MjJhbmJ3PBwlyuXlHErk4rX35gK/aoC2t5SSo/uXgel59UyKmVGby8p5dvP7GbY
FhCpRB879w5qJUKvvtULcPRY8zLNfPTS+Zz7f2bKU43UJltIi9Zx6VL8sk268bMDtGqlcz
JSeKk4lT29Tr441tNfH5VMb97vTH26/+m08u56YwKTFpZHKYDobDEMzu6+cFze/jg+gwyn
riKljW/4+c7h3incYjPrSgkTRniS8GnSN/zV0aqPsuT4Wt5vC5SYFSjVPCHqxbT3O9gW/s
IG5oizVhNWhXfWVdFx7CLCxbm4Q0Eeat+gO0dI1Rmm3h1bx8mrYoXd/fw9TMr+c6TuwlFg
2zvvWYpa6syAVhZmkZtp5UWi5tfvLSPi2vy+OPbTTF3202nl/Pmvn4a+53kJGt58IsrqMh
K4tmd3dz27J7YvB9dOJenvraKZL0mIftNZvwJhg9VqoaIhcjhDcYyTiOp93KDV5nZi5x7f
RhP7eiiIjOJl2p7+eIiLXu6bfglFXt7bDFlaH6emU3NlpgyBFDb62WLJx+GW1mXPsD6hkG
yzTqe29WNNxDinvXNMWUIYGvbSCxoel+vg5++uI9gVEEJhiU6hl08taMrpgwB1Pfa2dVpR
atS0Njv5NW6Pn71agMHBpxHTZWt7bJz9zvNvL1/gPMW5HLvuy1xrpB71jfT1O8Ynwsoc9y
0Djn5yQt7uWppDqm7/gzuYeo0i3mncQghICdZzxrzAOl7/gpAfe5lPF7njG3vD4W545V91
BSmxJQhAKcvyEu1vbxU20uP1cOBARe/f7MxJkudwx7+/G4L58zL4cEP2zgtqgAFQhI/em4
vA3YvAM/t6uGxbV1saR3mwkW53Lu+Oa5i+182tHDBwkgdrz6bj8e3dsbOafS8X76yjz67T
1aGJonDXWYqhcDmDaCPFgBVRWOIZGRmK7JCNIpgKMy2tmEK0ww0DTgoSxbstUgUpRs4MHj
ohlOYZmBfX6ICsX9EgdrVg0kRIM2oweYJsLV1GIvbT+MYCsfB1gyDzsQA7YIUPft7E7dpG
XImNHI9qKgdiT7boVYfeo0yoU3Ex9mHzOQx7PLjD4WpyVajH9gFwJAncqPSqZTYPQHMoZH
Y/KFgYu2prhEPnkDir/2mAQfpRi02TwCLM7H1jCcQQqkUNA044+KF+uzemNxsbRuOjUuQU
NAvFJZiyj3AtvYRLC5/ws02LIFlDNmXmRhCYSlO+VQqBDZ3AF1UIdLICpHMLEdWiEahUiq
4uCaPHR0jrCnP4L2uMGvyldR22VhRmh6bV9tp5ZSK9ITtV2UFCKRWMhAw0G/3kpmk5eKaP
ApT9Jw+JzNh/pycSNB0SbqRw38k7+9zcEZ14jaLC1PY2xOfaVTyEZ3uy0cFsPbbvQndzlU
KkRAsKzN15KXoSTWoeabegaPiEgCKzZGPqicQIs2ooU+ZCyIyVqxOzDxbWpQypgt0TXkGA
w4POWYd+akGtKr4r4CsJC12T4A1Fens6DikdNUUJJNljiheFy/Ki437AmFSDfHJBSativA
oheiSmjwKUgykGzVx8wwaZSy+SGbiCR3mMlMpBSNuf6xKv0Yl8I6hRMvIzBamZdCIEKINc
AAhIChJ0rKJPma7xUW/3cfyklTS9Goyk3UIJAqVVi63DIFa8KML5mL1BrC7/Vw0L5VBu4d
X9g5EWhqsymFR6gBBXR67gxn84wuppBk1eAORwNTrVhUzPy+Zpn4nr9b1cvXKYmoKItk5N
YUp3HlFDXe8vI+Ty9OZn59MWYYRjUpBy6CLdxoGUSsF160uYX6emfl5yezuslGSbuDHF81
DkqBz2E1+ip7OYTfBsERBqp4Rtx+7J0h5honbL57Lr19t4IVdPdxx2QLufbeZ5kEXqQY1v
/70ooSsH5nJp9/moWvEg1IhePwrKzkw6IL0z+GuWkeNNsybN1Sy16oihJJF6TkEc/+Fqm0
Dc5vv567zfsLt74zg8AWZn2fmBxdUo1cp+Nkl87C4AhSnG1ArBEqlIN2goXXIRUWWkbuvW
cqtT9cy5PRTkKrntvOrqeu28cWTS9jSNszykjTqe2x8/7xqkqMtac6ozuKalUV4/CFMOiX
/e9VifvFSPU0DLrKStNx6fjW/+vd+hICbTitnVVkagVCIe65dyrce20WPzUtWkpa7Plvzk
```

```
cq8zPgRkuItRAaNin67N+Yy0yiV+AKyhUhm9jItFaIoZ0iSNPTR046PSBbWMHe+3kBtl40
Ug5qfXzKfJ7d28kJtLzqVkvuurCSoMvKj5/fSOezh5LJUugqMXFKp5RvVOrS2Foqaf4cwn
UtT0Wf5/fpWem0HKMswcv3Jxaj7Ffzt/VZahlzkJeu441ML0SoF7x8Y4nydmlSjhk8vLaA
ozcB/P7+XF2t7yTRpueXsCm45q5KvnlaORimYn2dGq1Zx99VLqeu20mf38a0ndmH3BFlen
MqXTy3llsd24Q+FuXhRHkVpev70TjPz88z85vJFvHbLWjzBEAUpes6ozqLf7iNZr5Z7o00
x4bDE9vZhnt7RzVPbu4BIf70vVnmxOkC4ndgDLm7dpKZh0MtjF2nJfOWHiMF6SMrFf9Zvq
DJl8Z11mVhcfsKhAAZXF+/3gFdh4G/vteDyh1hTkc6Zc7KwOP387KV65ucl8+tPLeSpr62
mvtfBxuYhvvtkLStL08hP1fP7N5ow61XcfGYlf3qniRvXlrOsOI2mAQdLi1K4Z30zz+zsJ
ses47vnVjFk95KXamBNRTo1BavwBkM8s6OLy+7+EICvnFrKIzesxBsIk2ZSk2OW5W4yOdx
CZNQoaR0a5TJTKcZ0s8rIzBZmvcuscSCSeVXbFXE7SBLUdtt5dlcPobCEyx9k10CImx/bT
edwJBZndUUmd77ZypKh55n78qcpe/+7qHp3oHz3V5gHd9AfDT5tGXLxl3dbaB920TIUqcL
bY/PynSd3s63DisXlZ1t7xC0x4vLzo2frqO+NBFoPOn3c/kI9G5qGSDWoWVqchjbaufyD5
iGah9zc/mI9dk8kUHtr+wgPbWznpOJUJAle2N2D1RMkN1nH3h47331yN2a9muocMyadmnS
Tlnl5ZlkZmga0DjnZ3j7CY1s7CUbjb4atIxS1PonVK2Fqf4P/2qSmttfNDUtM1Gz+dkQZA
nD0kvTSV2hra8XlD/F/bx/g7nfb+cUGK+VZSfz+jUZc/shN7oMDFnZ1WnmvaZALF+axq9P
Kj1+oo3vEy02P7uDhTR14AiHWNw7yYbOF+Xlm7J4gv3h5H4sL0/jKQ9vZ22Pjzfp+/ufNJ
pqjlaX77F5+9FwdrkCYmx/bxebWEcqzTOzttnP/+22xc7r33Ra2t48wL88sK0NTQDAsoRj
1jW+IulQPKkRq5SGXmccfikvokJGZDUxXhUgCXhdCbBdC3DiRB7I4DyklAAvzk9k2KmgUI
NOkjQtE9gXDrCnQkN32QsL+0oc2k2M+FOTaY/OiPqwIotsfQqEQBEJh9kcVoD67l/2HBV4
HwxL+UJh2S3w7jpdqe8f8Jfdhs4Wlxamx1xubh1hSlAJEYpJ6bd6xL4LMlOIPSrzbGG8MP
atAwpU6h6C1m76kBdT1RmRgkdmJYqQ1fgdBH/n0x8nE9m43LcOJN7QNTUNkmfUUpEUUkp0
dVhrGCPh/r3GI5aP6nnmDIWyeAB3DbpINGrpGPHHzvYFwrPb5u40DQKQNyOG8NMaYzMQjS
RKBYDhWhBEiFiIAnepgDJEi1ubn8a0d/PTFvZO/UBmZKWS6KkSnSJK0lEhrkP8UQqwd/aY
Q4kYhxDYhxLbBwcHjOpBRoyQ3+ZAC0231JMQ1BA9LV9UoFRywhnGmzU/Yn8dcwcioFgk6t
QJxWCaPEMQ6TOdEj23WqWMxGqPRKBWkHRaMurgwBc0Y3eqL0gz0jVJ6SjOMsRtXsl6NWT+
dPaQnDuMpfxBptFmVE98ctdWhwODsQGFIxRwYiMmAJaQHTWK8l0MZLxMZRg1ZpkR5KsswM
uL24YlajTJMWrLNib0aSjONsSxIiMihUiFIN2kIh6VY3MloDh5/Xm4kNm5xYUrCnLHGZI6
NTyJ/gVDkO2x0DNHBli46zUELkSJmIWoacNIi95aTmWVMS4VIkqTu6N8B4FlgxWHv3ydJ0
jJJkpZlZiZmYh0LFRkmbju/GlX0iyJZp+KalYVcVpPLV04t4c7PLKQqy8h/nVvFmop0rlp
eSJpBwQXzMqkvuR4MGVC8BkpPQ8qtoTN1Rdwv9ZtOryBplCKiVgp+dskCVhankKpTsaQwB
bfPj1mn4o7LFsRlm11Sk0eGUUN5phF/IIQvut+La/Lot3tZW5kRt9+vrC3l5dreyHno1aw
qS6e2y4YQcMenFlCQasAbCMVlAMkcO+MpfwB5Zj0XL8qNKSbnzM1i9fxy/IuvJzO/HE1qP
neclc45c7MY0eQxdNovISkXKtdBegWDNTexyZGFQkQUcJVC8D+XFFGSYeK0qkMyolcruWR
xHjX5KfRa3awoSeWuKxZRU5DCunlZsXk6tYJPL81nfUPkZntqRQbNg07uuHQ+VVlJZJjUf
PW0srhzuHpFEdvahrlqeQFnVkf2dXFNHnmj2szkp+i4cFSGmswn45PInzcYQnNYRuFBpVa
nOpR27432T2wedNIxHG+ZlpGZ6Uw7k4EQwggoJElyRJ+vA3423scJhSU2tVj415YOXL4Ad
1+zBJNWzUu1PfzkhXpOrczgnLnZNA44eWZnD0uLUrnipHwWSgfIa7gLvbMDz8qbCVz0R1T
b7oNQgPCab6FSafjr1QsiLRMMRnrtkYyaR768ArsnQFCCRze1oRAKLlmcR6vFxXO7uqkpT
OH1+j5+fukCFApBmkFNhknL5tZhrr1/C/NyzdQUppCkVbK2KouvnVZO54ibzy4vxOsPkZe
iJz9FR+G1S/EHw5RkGLF7/NxzzVKK0gyYdSru29DM87t6WFqUwjUri6keo1eazOThC4TY0
DTEAx+2srbEyGNXFeFVGPAFQuQMf4j+/b+h1xgZWngjqcYcKux2Xqnr55S1C0hbej2Khlc
Il56Bau5nMLUn8db+AW6/eB6lGQYe2NhB53A/ly3O44ZTS+m1+Ug3akgxqGgbcjPg8GPSq
eixeXnwwzaWFqdxzcoimgddVGQlMeT08u1zqsg260jWqzBoVLy9f4CH/76Fc+Zlc1pVBos
KkrG6A2QladGoFOzssPJibQ9/eqeJz68qYUF+Mo/fuDpWg2tOTpKcZj9F+ALhBKuyQaNCq
RCoo81dVUoF3qjlsN3ixuYJYPMExrRcy8jMRKadQgRkA89GKy+rgH9KkvTqeB9kR/swX3t
4Ow5fJCj5ptMruPmxXbE4m8Z+BzlmHb98ZT+3nF1Jn93HAmU75e99DoKRYnJ6Zxe8/qPYP
pXtH1B94V081DiHcwtD6Nx+fvRKxH32xTUl1BQk883Hd8fmb2yx8MML5pJq1PC71xqoKUz
hh8/VIQT86XNLuGd9M2/tj8Rj7O2x817TEBcuysXlD3H1ymIKx6gdVJyR6E7xBULc9swen
tnZHdvXq3X9PHPTyWPuQ2Zy2NY2wlce2oZKAfestDJgT6ZN6Jjj2ETOm4dC57Jb3sJ90ZP
cu97P+VVJVNT+FkVL5COh6KvF3Pgaqsp72NFhZ0eHlW+fU8m7DYO4/CH2dNv46toytrcP4
/AGuf7kEn7wbF1s3xsaB/mv86r59b/3c+2qYs6ck8GXHtgaV1jx1vOrebm2lz3RBrJ7e+x
saR3mz9eehFmvJhgK8+t/7+dv70dim+q67fy7ro/nblpDWaZJlrFpgC8YQn2YhcioVaJXK
```

```
2NV7jUqBe5ACG8ghMXppyTdQIfFzcKCY2veKyNzojLtXGaSJLVIklQTfcyXJOmOiTjO/j5
HTBkC6Lf74oKOl5Wk8UpdHytK01jfMIAvFKIscCCmDJE5B7q3J+5434tcXqXkoSY1FYruW
NG6/b12nt7RnTB9V6cVvVpB25CbvGjTRUmCjc2WhGDXPrsXk1bFPeubjykDpGPYzbO74o8
96PTJ7TqmmBd2RwKML1ucj6n+cbxJ+fgCQYoaHoifKEmktL9KYZqeS4p8GFvifx8o7R0sN
QzEXr9c28epVYdcKY9saueaVcWcOz+Hf23pjNs2LEHbkItss5ant3fhDoTjlCGAx7d2JlR
H/7DZQpslEmPSbfXw4Ma2uPftniANY1Rzl5kafMFwQhFOg0YVK8oIUZeZP8SQ00eyQU1Wk
pYem+fwXcnIzFimnUI0WagOMx+rlPGlov3BiInZHwqjVipQIgiLUQa1oB+U8cHOAKj1BCU
FJjVIQhnrGSZJh9JbR6NVKxAIwpIUFz+kVSsIH95XgUhAtk6tZIyY6iOiVIhYjFTc+LHsR
GbcORjU6g2ECat0IEmEURBWJrbiCKn0BIISQRSxCtWjCYtDsqVVK/CPasGgVSsJhiS8gcS
bIkSCaYMhCY1KMWZPPJ1KkdCeA0AVzViKuF0S96uUe5RNG3yBcML/yKhRxn0nHaxDZHUHM
OtUGLWqMdv8yMjMVGbtHXFeThKlqQYKkjWkGhQk6VQszDOjVERK2O/sGOFTi/PY0THMmdV
ZSECDsgL00bT2kVbIWQjKUf51oUCqvk30OToAACAASURBVIQH9/i4tsJPXSAv1tB1VXk6l
y/Nj1N61NFii1ZPgGUlqdRHe5dpVQpWlqaztjI+YHJOdhJ9tkhsR7J+DGXsCBSmGbjh1NL
D9mViTrZcnXoqubgmlxyTgo0tQzjnXYXJdgCzVklH9Vcimu9B1Absxevos3t5uEGJdf51c
fvxZS9l/fChFPlLa/J4f1RT1xtPLeMfH7SwudXCF04uidtWp1aQl6LD4vJz7coidCqR0PL
jcyuLcBx2Y/zM0nxKMiKusPwUPbecVRn3fkGqnrlyjNq0wRcMJcQQ5acaWFN+KOheG027t
7oDGLUq9BoVdlkhkplFCGkMK8SJxLJly6Rt27Z9rLnhsERtlxW3P0goLNFt9eL0BanKNqF
SCNz+ED02L5IEpek6BoatpGslytPUNHmS8ARCzBGdZPW+g8HTi2f+55BCATQtb6AIB/BWn
E+TsoIWi48Cs5I+u4/NHQ7OnpeDQqFEKSAMvBvN3llekkaKQQ0ShJB4c98AWpWCZcVpmHV
K8pL17Omxs75hkLIMI9lmLTnJOpYWp2LQHFv415DTx+YWCxsaB5mXl8xpVZly24SPx1HNH
McifxApwNky6CAYlugdceF0e5iTbWLIFSA9yUCfw0+f3UdBsoYag4VuJzS4jWg0OvJSdHx
wYIjT88IsCe5E0/E+gZyltKau5n2LiRS9GrNeTYZJw3tNQ7Rb3KypyCDHrKWx30mWWYdSA
UoheKO+n1SDhoWFyWxqtjA/L5mCVD12bwClQkFdt41+u4+TilNQKxVkJ2lptbip7bKyuiy
d1eUZsZIRAFa3ny2tw7y9f4DKLBOnz8mK66En84kZF/nb2Gzhjpfr+eGF8444Z2vrMLu7r
HxqaT7/3NxBXoqebLOW751bfeyrlpkpzCoz73QMqp4wNrVaGLB76Rz28K8tHfTYvGSYNPz
skgW0WVz848O2WNd3jVLBredX870n67lmoZkvnuTlxXYPi8oVOKs+g2n7nTzeEOKO9+zkJ
p+GUiEo6dVi1vWysiyd6x+p49vnVLF2TjZ9Dj9N/U42tVjQq5WsKU9nW4eVl/f0MuT0c9n
iPMoyjXx4YIgbTi3la49sR6tS8NiNq7i4Jo+La44/VTnDpOXCRXly2vMU4g0EeW53NwUpe
v74VhN7uu18+ZRS/r75ALecWcEr9UM88GFbbP6FC3PxBEK8vb8JgAX5Zr55VhUdln5WmHX
sL/08177sweKMxAWlGNTceGoZgw41lVlJrG8Y5Nmd3eQm67l0cR4/e6me2y6Yy33vtvDji
+ayvWOEglQDA/YeXt/bSL/Dh0Gj5F9fWcW583MS1r+mEqB4zHNLMWhYNz+HdWNsJzP1jBV
UfTjqUS4zk1aFUaPC6pYtRDKzh1njMvP4QzyxtYPaLhtqpaAnGkD96SUFNPQ7sHoCMWUIw
B8K8+a+flaVpvNIrR2H28tje2xYh/sw9W+k3byEOzdGqkz32rx0jXh4/8AQc3LNPLO9k9P
nZHHfhhYCIYm7Xm8k3RT5lX76nCzu3dDCltZhhpyRwOjndvVQnG6kedDFa3v7WVKYgi8Y5
h8ftMZikGROfJoHXZSlR4pl7umOyI5ZF/lN4g+HeXRze9z8l/f0srL0kCusrttOQ7+d/Ix
UpLpnuK9OYHEeCq63ugP0WD282ziI0x9kUUEKYSkS9KxQCJQKwVPbulhemsqmlmF2tlt5c
XcPr9X30x+Vfbc/xONbOyb6UshMMr5gYgzR4UQqVYewuv0YNEpMOlkhkpldzBqFyB8K4Q1
KDDl9cdllKQY1nkBwzODBfruPVGMkRsjl9ZFh0uIJSqh8tki9mGBioGkoHKbX7iPVoMHqC
aBWKrB5AoTCkblhSWIsL+XBCrGDDh+p0arEHcPu2HYyJz4uXwiFELGAZyEgEJZIMWhw+UI
EQomCcfiINxBGqxSEELQ7EudbXH78wTACMOkOGYDdviAapYJBp480gwaLy0dBqp5ua2IWU
euQSy7eOcPwBRPrEB2ORhmJIRqJxhAZNUqsHrmfmczsYdYoRMl6DScVpXByeQaZpkOtCt5
pGKA43UhJemKtlHPmZbGx2UJWkpa8dDPDLj9ZRjX2tAUUDL3PsoL4GByTVkUoDJcuzuP9p
kHOrM5i0OHjvPk5aJRKlArBoMNH0WF1WbKStBiiGUenVWWytTXSS+3alcVoVImZaTInJiX
pBtqHXaQZNagUAkmKVAuu77WRY9YyNze+fUeKQY3NfeiGpFEqKEjRs6vXjSoln6srE5XlB
fnJVGUlIYBNzZbYeGaSFpc/xHnzc3i3aZBFBSm80zDABQtyE/bxuRVFcS0eZE58fIFQrAD
jkdCqFPiCkaauRm00y0y2EMnMImZVUHWnxYkkhXH4wjGXWZI2UoXX4vQx7PbTMezhqe2df
HZ5IUsKUrC4/GSbVBgUIQa9EaVHo1KwwLudTqufu5vSeP2Ak3m5Zv7j9DJ6bW4a+z34giH
OX5gTSU2WJN6o76M43cRTOzq5fGkhGxoH2dI2zJKiFL5yahl3vtbAWXOzGXJ6eW1vP986u
5JlJWlYXH7SjRpKM0wJpfdlJoVxC6p2egJ0Wd0oBPTafPiCYUw6JQohQIq4vIZcfvZ226j
ONZNt1uL2BbnzjSZSDWq+fkYFOSY1Pp8nUq4hHGRbp5M/f9iDRqnghlPL0GsUZCfpCIQl/
uupWtIMaq5dVcwrdb3MyU7CrFeTk6wjSasmzahhbk4SdT02GvudhJGozDKxoiQ9ZqWUmXL
GRf4e3tTOe42DfHFN6RHn9Nu9/Pa1BqqyTZxUlEpeip4/vNXEhv8649hXLTNTmFW/jGZNU
```

```
PWA3YPbH2TYHeQHz+6hdSjSp+fc+dmcWpnJj5+vQ5Igw6Th/z63BIvTz23P1sVcClecVID
TF2Rnh5Ub15bRrS5Bpwxw+coMcrNs7O9zcOMjO/jS6kK+lNdOUJfOBQ9txx8K87W15VywK
BerO8AXTi5Br1aRYdJw5fJCuoYjN8h7rlmCUasCBN88u4qGPieX37sRpy+ISiH474vnceW
yQrRj1DKSmf5YnD4a+yJxQw9sbOe1vf0AVGSZuPX8am5/fi9dUVm7fGk+W1otvLynD7Nex
f98djEv7+7l5sd2olRE5OPF3d1ctbwIc1ISFyzMxRsIc+/6Zr57bhWPbW0nxaDlwS8u59H
NHRSl6fnq2jKc3iBatZL7329lR4cVgJ9dOp/X6vr4IGpNWlqUQlV2kqwQzTB8gVBCrbXDi
dQd8jPs9JOkU2PUymn3MrOLWWNyaB504Q2EeWZHd0wZAnhtbz8Wpw9ltO7LkNPPr/+9n4Z
+R1x8xZPbu1hSlEKf3ctre/t4bp+dgbCZO99s4Z71zby9f4BgSOK+9ztoFwWUbvg2n1tgQ
pLg3nebaep3cePDO+iz+/j6v3by1I5u/vFBG2/sG+B7T9WiUStJN+lIN2kJhSW+/cQunNF
Yp2BY4icv7KVpwDm5F01m3Dgw4MDi8tE+7I4pQ5FxJ6/v7UM7qmLw0zu6Y33m7J4gt7+4F
5VKQSBaXPFPbx9gbWUWL9X28vb+AR7a2M4T2zoZdPr48XN7WVyYxhPbutjTbadl0MG7jUP
c+PAO/vROpBXMQWXIrFfRNeKJKUMAOzqsvLKnb5Kuisxk4QuGUSuO/nVv1CiRJDgw6CTDp
MGoVeLwBeV4MplZw6xRiIKhMN5gmI0tloT3em3eSD2gKLu7bGNWdnZ4g2hVCnZ3WSnNiNQ
u2t1lTZjX7dOhGKxnYfqhrveeaND0WIHY/XZfXLbQkNPPwKiMN4hUuu6Vy+ifsDi8IZAU1
HXZEt7b3m6lMjs+fsjmCcSqSncOe8gyH4p7c/qCKJWCnR1WitLj49j8oXDMtVrfY+PaVcX
U9UQsU1U5SezuPHT8ojQDjWO013hn/4B8E5xh+ILhj7QQCSHITNLiD4ZJNWhQKRTRwGrZS
iQzO5g1CpFKqUCvidQAOpzcFD0jo4IHlxSmJPRzAkjSqfAFwywpTKVl0EkwHGZJYUrCvHy
tl3DWAnYNHXJv6aOuLq1KweHdEXLMOjJGBXpnJmnJHnUDhEhGUm6015nMiYdZp0QIacxGm
ctLUhP6fiXr1THluSjNQL/9UJ+9JK2KYCjMScUptA+54rbTKBWx7ebnJ/PQh60syI8cs6H
PweLCQ8fvsLiZkxOviAGcNTdbDqqeYXgDiZWqxyLDpCXLrI39/5MNaizOyI+zn79Uz6t1s
vVQZuYyaxSiklQtZp2KK5cXUjmqgu6li/PIMmk4GFyebdZy6/nVVGYZY9lgQsA1KwtpGXT
yjTMr+OKaEq5dUcCqTD93nZfNrefNIcesQyHgprXFlIba2H/W/bzTHkAh4OYzK1iSq+f+K
8qp77Fz2/nVsYyPZL2aP1y5GJ1aSbc1Eoydbdbxv1cuxqyPhHiplYI7LlsQt26ZY8fhCUS
ucSD00ZPHmZJ0HVWZJhYWJHPxokOZXfNykzhvQU7MIiMEXL2iiPqoVSfVoObnl84jw6The
2eX89tz0vnTp8vY1GzhwoW5fOakfG46vZylRano1Ap+cdkCHN4An16Sz4I8MzVF6awsTeW
s6ixah1wUpRmYG1WC7N4gBal6ThvVCHZlSRoXLJxdxRUHHN44hXMm4g18dGFGgDSjhqykQ
xXIk3Vqhpx+XL4gj25u56GNbRO3SBmZKWZWBFVvb7PgD0m81zTEo5s7uGp5Id9ZV0WP1cs
7+wfY22vn4S+vpHnAQavFzS9equfmsyv5xWXzqe2yIYCSDCON/U7+8UErWrWS75yWx8rOu
zB1vMWna26i+rwLSU1Nx+rycumzRiSpgxtOLWNpYQqPbe3gsvtaOK0imStXlPB/b7dw81m
VFKYaWFyQzIDTxzV/3URjv5Nz52dzy9lVrC7P4OVvnEqPzUOaQUNphjGhIa3Mx2db2zA/f
7mefT0OzpqbxbfXVVGZlWgdmQh2d47Q0Ocg1ajhrtcbmJ+XzJ+vXUrbkJuWQSev1vXy+8/
WsKPDitsfwu7xc9qSPBYUmFlSkMp7TYM8vKkDk1bFbat1LGv6I/9cdzk7hJFfvrKP5kEXp
8/J5P51y3hhZzfrm4b41tmVtPTbeGF3D8/v6uZrp5Xz1dPK0KqUXLGskF6bB7VSQXmGiZV
laaybl82g04fHH2TQ4aM4fea3dbF7Ary8p4e7Xm8kEJK46fRyLj+pIM5aO1PwBsIk6T766
z7jsGB6s16NxeXjvaZByjNN7Oq04vAGSNKpj7AHGZkTlxmvELUOOtjX52DI4eee9c0AKIT
g+0/viRVjfO8AbG4ZZlFBMk/v6Abgpkd38LvP1HDn642YdSquWVnMve9Gtscb5NaXWsk5/
2pOb3yarE134D4llV2upVSKXoZdEevP715r4DvnVPHq3n78oTBvNIzQZQ9SlZ3EXa83AvD
Il1dww0Pb8AYibo4Xa3uxuPzcd91JFKYZKExLrI8kc2y0DDq5/u9bcPkjlqF/1/XRb/fyj
y+uIFk/sV/s+3vt7OiwIkkSP35+L75gmIsW5fOf/9wZV4V8xB1gwO5jZ2ckJm1Ta0Qeh50
B7n+/FYjEDt3ympd/XnAxJz9xJfaTH6e+N+IyiyQH+Ll6ZRFP7ujm1mfq+P55cxh2RWLTf
vXv/fzm8oVcubwIgLyUiPvV7gnwg2fq2NY+ElvLw5s6eOkbp1CWObMtklvahrntmbrY61/
9ez/pJg2fOalwClc1MfgCIdI+RubgOfNyCI4qBmvWqRhy+Bh2+anKTsIfDHNgwMmSotSJX
K6MzJQw400OvTYfhakGXtjdExtTKUVCZeqmAWfsJgEQCEmMRIviLStJY0PTYMK+3+zRQmq
krkdR6xPsHVGRG+4ld1TTy40tllgMB8C+XkdcYcb9fY6YMnSQD5stdI/MbBP+ZNI65IopQ
wfZ0WGle8R9hC3GjxGXnzaLG51aGYvt8QVDCS1Z3qjvZ/Wo+La9PXZOKk7lnf0DCfv8cEg
PumSKwt1x49vaR/AGQpRHFZmWQRc55kOy+PyuHg6ne8QTpwxBpH1Hy6ArYe5M49U9vQlj/
9zSgT80+S7ViWbEE8Co+eiSHXqNMs76Y4q6zPb22ilMNZCXoqepX852lZmZzHiFSK9REAp
L5KUcujGM5XpSK0VCm4SDHeUtTh/Zo24sBykyhcETuZm4kkoxqgU+hR6H91BrkGyzLhaUC
JGg6tH3wpQxLBQmrSpWuVrm+DFpEw2hOrUi9v+dSFQKQZJWFZfyPJb8ZZi0cUq6NtpXaiy
5y9VL4HfiVsRbDw8WDT3YfyrFoImVbgAoGcMNptcoYwH/o/k47pUTnbHcgmUZJlRi5n0tD
ti9H7+21HALvPxtePEWzGEbAw4fDX2RH3J5yToa+hMzE2VkZgIz75N/GOUZJgYdPj6/qji
WZbGzfYTz5mfHzfvaaeW8NiqDYnlxaqyPz+4uG6dUZKAbVSsmw6ThdFNnRCHSGGko/Cxnp
```

```
I+wK1AYuwklaVWcXJ5Om+WQJeK61cW8Whf5ZZpiUDMvz8zZc7Pi1vKDC6plV9k4MicniQs
PCxS+9bzqhBYqE0Fppp7q3CTqe22cWpkBQL/Ny4J8c2yOEHDT6eW8WHvIgnPd6hLuf6+Vi
2ty47KD8pI1rNI04S1Yw34p3rXzpVNKcHiDDDp9ZJu1mPWqOFm8dHFewvqK0w3cev6cuLE
LFuZQlT058VVTyTnzs0kdVW7DoFHy+VXFMzLDbtAR6a+YQDgE9c/BGz+GrX+D3f+C134AO
QshbzH5DQ+yqamPEbefnGQdBakG9vVGAv6DoTC/fHmfXA5EZsYw7Vp3CCHOA/4AKIG/SZL
066PN/zil65sHbAy7gviD4YhyIqA03cCIO0C/3Ue6SUNhqpYhZ5DOYTcpBg3F6XrcvjB9D
i8eX5DyTAMKIWi1eFApBXPToNy2mXDQh8NciVOVRprCRbuUQ5PFT0iSyDHrCIXD+IJhHE4
nxUmCZLOZxuFI9en5ecmUZ5kYsHvZ021jyOGjNNPI/LzkaNVqmfFi0OFlT7edQbuXkgwj8
/OTx7QcjcFxt07Y12Nj2O1HIQRDTh9DTj/lmUaGnH4c3gDpJi2ZRi0DTi8Wp5/cFD1eXxB
PMExxmoFQOETPsAODMsw8k4tkbzed2nLCxmy6RjwMOnzkJOvIM6tp7rcTEkrmZGjwhQUtw
wHCkkR1ThInFaeNuT6nL0hdt432IReZZh0L881kJiVapmYizQNO9vbYCIUl5uaaYwUxpxH
HLX/BUJg5P36VB7+4AqXfDvUvQNALSTnQ9AYolFCwApx9EPBAwbLIe0CwfStfra1gSUaYr
y9WYjXP4fuvdLL7J+u4d30zf9nQwqL8ZB6+YeX4nbHMdGLm/To4CtPqriuEUAJ3A+cAXcB
WIcQLkiTVH89+y7OSKY8+X1N5fGs8OW77SIDq6N/SGcBJ5RyVBSXxr7PMOs4awzUiM35kJ
uk4s3pqrvHcvMTaQ8fOaIvmEg5GGy0tjp+1pCQz7vXHuU2ZtCpWlaWzqiyxRtdMpzzLRPk
ML2cx5PSTrFdHlKGXvw2pJaBLht5dULAcchdFzJTMT9hWVbycUwcHWUIjNHSQMvwn9NI3a
Om38cimdr67rorfvd5A14ibglTZqi1zYjOtFCJgBXBAkqQWACHEY8ClwHEpRDIyMjKzlX6
7lzSDGt79DWTMgTnnHdP2X1iWCWQCa8Bjo/KDHn59z1/QUkj5/jdZmbaSZ7d38I2zq3m/a
Ygnt3dyy1mVMz5LUWbmMd1iiPKBzlGvu6JjcQghbhRCbBNCbBscTMz+kpGZSGT5k5lKjlX
+BuxeUvy94HdC5brjO7g+mQuWz2FDcD5XlXgQ+hTO8r7Jg2/v4t1XHuc/H91GOCxx7f2bs
UazdP3BMHav3P5DZvoz3SxEHwtJku4D7oOID32KlyMzy5DlT2Yq+djy5xyAA28RXL8Bt2M
Rw3MvAtvxZ4glA79aBVpFOsOkkzR3DhV1bq7fkMyNyueoqW9mRFrHGb8YZol+gLddJQAsN
Nq4stSDxmCm369DoVRRmKwiWa/GHVTg8INWpycltwRBpJ+fxx/CqFXF2ib12734g2HSTRo
yk7To1EoEkT5skiQRliIKWDAcRiEEGpUCpULE5hzheiIBgWCYQEhCQkKjVKBWKY663YnEw
XMMhSX8wTBhSUKtVKBWKkg3aijJmPmFWD8O0yqoWgixGrhdkqRzo69vA5Ak6VdH2WYQaP+
Yh8gAho53nePIdFrPdFoLTJ/1DEmSdEQfwwkufxOJfK7jwyeWv4cu0xV9vkaTuT1cyVX+H
5EkTU5tKSEikbhDHOrzaMaFHfmmO13p+P1ndkoB7+GdxzOA/UeTv5nGdFOIVEAjcBbQDWw
FrpYkae847X+bJEnLxmNf48F0Ws90WgtMv/WMBzPxnI6EfK7Tg+m8to+DvP6p40Re+ydlW
rnMJEkKCiG+DrxGJO3+7+OlDMnIyMjIyMjIHIlppRABSJL0CvDKVK9DRkZGRkZGZvYw3bL
MJpr7pnoBhzGd1jOd1gLTbz3jwUw8pyMhn+v0YDqv7eMgr3/qOJHX/omYVjFEMjIyMjIyM
jJTwWyzEMnIyMjIyMjIJDBrFCIhxHlCiAYhxAEhxK2TcLxCIcQ7Qoh6IcReIcQt0fHbhRD
dQohd0ccFo7a5Lbq+BiHEuROwpjYhxJ7ocbdFx9KEEG8IIZqif1Oj40II8cfoemqFEEvHc
R1zRp3/LiGEXQjxzam8NhPJZMveZCCE+LsQYkAIUTdqbNJlaaI5yud42p/riSZ3xyJT041
jlZPphhBCJ4TYIoTYHV3/T6PjpUKIzVEZelwIMUaH4BmEJEkz/kEkY60ZKAM0wG5g3gQfM
xdYGn2eRKScwDzgduC7Y8yfF12XFiiNrlc5zmtqAzIOG/stcGv0+a3Ab6LPLwD+TaSkyCp
g8wT+b/qA4qm8NjNJ9ibpvNYCS4G66SJLE3SeR/ocT+tzPRHl7lhkaro9jlVOptsjKq+m6
HM1sDkqv08AV0XH/wz8x1SvdSIfs8VCFOuRJkmSHzjYI23CkCSpV5KkHdHnDmAfY7QhGcW
lwGOSJPkoFdCjwYff4gcNmo8YekCJuAFCFE7gQc/yygWZKkoxU3nKprMx5Mu
uxNBpIkbQCGDxuealkad47yOZ7u53rCyd0xytS04hPIybQiKq/O6Et19CEBZwJPRcen7fr
Hi9miEH2sHmkThRCiBFhCROsG+HrUnP73USbUyVijBLwuhNguhLgxOpYtSVJv9Hkfh9qqT
9Y1uwr416jXU3VtJooTee3HylTL0oRy2Od4up/rdFnH8XKk6zxt+ZhyMu0QQiiFELuAAeA
NIhZGqyRJweiUE1WGPjazRSGaMoQQJuBp4JuSJNmBe4FyYDHQC/x+0hYy8b/gz4EPcIOe94VehWqoHWnRPEFkIemKqvmPRpG9MDlCYpcC2rGoI2WV2Dm8iXMV8bO/J/7aA35+yr/+fqLHPYgw2PmtkRQMTuHPJHKaHVFsHRd4Qhh+AQuoBp3WUhVODBEPHzghaoM8CSbITRjQ0xIdqZcQEnzD8EbqBzFHb0+Tz6QmNMbqBBuKH9ocLBsAEOd+k9BRVLDnyYD7+0I57GEPp+3BbAD
daOflOK2EUnLe0w6o++BHgyOjSV10ZmHJlsWZpoxvgcx5hp5zpdORGu84ksJ5IkhSRJWgw
UELEwVk/xkiad2aIQdQOFo14XRMcmFCGEmsiH41FJkp4BkCSpPyp4YeCvHHL9TPgaJUnqj
v4dAJ6NHrv/oEk/+ndgstZDRDHbIUlSf3RdU3ZtJpATee3HylTK0oQx1ueY6X+u02Udx8u
RrvO04xjlZNoiSZIVeAdYTcTle7CA84kqQx+b2aIQbQUqoxHzGiJumhcm8oBCCAHcD+yTJ
On3o8ZHxxN8CjiYUfECcJUQQiuEKAUqgS3juB6jECLp4HNgXfTYLwBXR6VgRdqHzw/aj3XRbN
mVgG2Uabf8eJzjHKXTdW1mWAmXfamkKmUpQnhSJ9jpv+5zhS5O9J1nlZ8AjmZVgghMoUQK
dHneuAcInFQ7wCfiU6btusfN6Y6qnuyHkSyPxqJ+EV/OAnHO4WIebQW2BV9XAA8DOyJjr8
A5I7a5ofR9TUA54/zesqIZJrsBvYevAZAOvAW0AS8CaRFxwVwd3Q9e4Bl47weI2ABkkeNT
cm1mWmyN0nn9C8ibs0AkdiCL0+VLE3weR7pczztz/VEk7tjkanp9jhWOZluD2ARsDO6/jr
gv6PjZUR+fB4gEtqgneq1TuRDrlQtIyMjIyMjM+uZLS4zGRkZGRkZGZkjIitEMjIyMjIyM
```

```
rMeWSGSkZGRkZGRmfXICpGMjIyMjIzMrEdWiGRkZGRkZGRmPbJCJCMjIyMz6xFC/GCq1yA
ztcgK0QxECHG6EOKlST6m/GUiMyUIIZwfPUtG5iORv8NmObJCJJPAqFLtx8Ixf5kIIZSf4
Dgys5hPKJsyk4wQokQIsU8I8VchxF4hxOtCCL0QYr0QYll0ToYQoi36/AtCiOeEEG8IIdq
EEF8XQnxbCLFTCLFJCJF2lGPdLISojzaFfiw6Zow2iN4S3celo47zjBDiVSFEkxDit9HxX
wN6IcQuIcSj0bFro9vvEkL85eD3lRDCKYS4QwixO7q27Oh4thDi2ej4biHEyUfaT/TxgBC
iTgixRwjxrYn6X8h8fGSFaAKY5C+DCiHEm9EP4A4hRHn0LZMQ4ikhxH4hxKPR0vIIIf5bC
LE1+kG8b9T4eiHE/wohtgG3CCEuFkJsjq7hzVEfepMQ4h/RD3GtEOLyT/BlcpcQYjewWgj
x61FfZndOyD9E5ohMlqwKIbKEENujz2uEEJIQoij6ulkIYYiu5e2oLLw16v0HhBB/FkJsB
n4rIu0oNkZl8BejjpErhNgQlbk6IcSpE3v1ZD6CSuBuSZLmA1bg8o+YvwD4NLAcuANwS5K
0BNgIXHeU7W4FlkiStAj4WnTsh8DbkiStAM4AficiLYsg0jz6SmAhcKUQolCSpFsBjyRJi
yVJukYIMTc6Z40UaXgaAq6Jbm8ENkmSVANsAL4SHf8j8G50fCmw9yj7WQzkS5K0QJKkhcA
/PuLayEwCskI0cUzWl8Gj0ePUACcTKX0PsAT4JjCPSPn1NdHxIhqMzwAABOdJREFUP0mSt
FySpAWAHrho1L40kiQtkyTpLuB94P/bO7cQK6sojv/+SjWINSQhFNjvQSFWQRdMSqjhIq
CrIcyoh4KEZEEtcjALgSFLzaY1HQxciLTbkSRhRecSSi7yGhGBDmBQWakZtbUXFYPa39zP
k9nztHhnDMjZ/1gYF/X3h+zz9rrW3t/rMvTHN4EFqc2j+HxmaYlBbRpBMrk8zTf7/CYZRc
kWUObW1BXar5WzQMKN0k6DZgBfAnMkNQK/GpmfwFtwGtpLXTgG0xCC3ClmS0EVgCr0kaSj
xN2F7AhrbnpePiEYPTYY2bZ/+ArYGqF9pvN7LCZ7QcoAR+k8p8nYDAbYUk
7gC1lAE3B2qttoZofMrBfYDbSWkDkTuATYnmTM
ARk
dQmaRbwR5nnC+pEuJ9rx4iUAXBYUrEyuLBUB3mw1ilm9i5A+oGTnD5fmNnelN+Rxu8CrpW
0GJgATMLjmmVjrc2JbwHWygOungzsSeXX44EiSWMeKDG1vBIAN7yyKM8DeERocKXXC7wsv
/NU13tPwRA1X6uJbbhhfjXwNDALj/3Vh2z+b6rjOzgZS+ioLR9jrwTEpvB16
RRx1/L/dMwejwTy49gOuBfgov4k1l2g/m8oOU36tuwtfULcCjkqbh6+b+DSZSXmV
Uq2cOP8kRJ1fVaIeTVc/4pyJE0HbsS9WncC95eRE9SB8BDVjlI/uloog2MeX1IT8DwwO71
dtxfN40gu3YZ7k6YBD5aYbzqfr4OMYIqke91upW3
DvUikfNno4Hxews0yfjSFH+f0EYzWwrvjH+DKyWVM6zGowOPfjLEhSiqI8YSeOAs8xsM7A
EaAYmAhuA+bkrARCfg7i+ZEyDB2SdLemM1S8maWYMwN7UfL6L6l5O
BaMMmEQ1ZceqqgM0lv6Xkm3AUg6RdKEMrzk3SRMrzKEZ31gA7s2VfwrMyzKST
maQ7NZvYR8BC+QQZjgx6quFYTncAc4AczGwR+xyOCd6X6bS8b22dm7
cBLxCYzFlkOzJX0DTC5CvLGA2sk7TnZgeBJ4GTgZ36sMbDduqHwiqRv
Xe2dW6L8A98DvxD2t55eRMwXYkrz3a4BSnqigzsSRWX1ZDrwl6QHgwyrJvAd4QdITQB9wx
3ANzeygpHZgF/ALfsQwHMuAdZIOAJuAcE1L5U8BKSbtwb8LjwDsUlMnX6R5RpgTGpXnNA34
qGuNU4P3kuRKw8NgfO6gxVV+rZtaT3ti3pqIuoCV37DofeFXSImA/cN8qwHYAb0hagnuAM
q4BFknqA/6k/N27oIaYB+rLk31q4HVufZTTc+n9l1DhKDQIgiygI+5lDhKDQIgiAIgqAxCQ9R
ARBMOaRtJLC17IZK8wsPlkPqkJ4iE4QQhkEJwqxVoMgOBEJgygIgiAIgoYnvjILgiAIgqD
hCYMoCIIgCIKGJwyiIAiCIAganjCIgiAIgiBoeMIgCoIgCIKg4fkPz1sPfd6oBhEAAAAAS
UVORK5CYII=\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sns.heatmap(df.corr(), annot=True)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 357
        },
        "id": "77ZkUfBdnz0c",
        "outputId": "d21b6b95-1651-432e-b88f-b5fcc6aafa9b"
      },
      "execution_count": 27,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at
0x7f5f8e7e2c90>"
            ]
          },
          "metadata": {},
          "execution_count": 27
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x288 with 2 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAaQAAAFDCAYAAAB1DXoPAAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3dd3wVVfrH8
c83jQgoIkgLVUWxgHQEUVEEO2DvfWXXta1d17Zr7/tTl1Wx69rdXURFARFFRSWAFAFRBIS
EjtJBkpvn98cdwk2AFEkyN8zz3te8vDNz7txn7ob7zDlz5zYUsIOwDnnnANPS
M4555KEJyTnnHNJwROSc865pOAJyTnnXFLwhOSccy4peEJyzjlXLpKel7RE0nfb2C9Jj0u
aJWmKpI5lOa4nJOecc+X1InB0CfuPAVoHy1c1Du0ybIc1BOSc865cjGzMcAvJRTpD7xscV8
0pqXNpxPSE555yraFnA/IT1nGBbidqcJqBP+4Fhh5A0vlgyI+wQkkZmWkb
YISSVNevmaHveX57fm4zd9/wj8aa2TQab2eDt+fyy8ITknHNRUBArc9Eg+WxPAsoFmiWsN
w22lcib7JxzLgqsoOzL9hsKnBf0tjsIWGlmC0t7k9eQnHMuCgoqJNEAIOl1oBdQX1IOcAe
QDmBmTwHDgGOBWcA64KyHNcTknPORYBVTM0nOJadWcp+Ay4r73E9ITnnXBTE8sOOoFFSek
JxzLgrK0akhLJ6QnHMuCiqwya6ieEJyzrkooMBUnpCccy4KvMnOnOedcUvAmO+ecc0nBa0jOOeeeSgteQnHPOJQMr8E4Nzjnnk
oHXkJxzziUFv4fknHMuKfjgqs4555KC15Ccc84lhWpwDykl7ACqiqRdJf25Cj5n
zttet9z7KocedwYBz/hR2KFWia68uvPzZC7z6xUucddkZW+zvd87xPP/xMzw7/Cme+O//0
aJ1cwBS01K56R838PzHz/DS6Oc467ISJ8qsFo7q24tp343h++lLuvw2fbBeJbzo8gV19xkN5RmrUN
mThjJiI/epHnzrMJ9zZo14cMPXmPqlE+AZmWeRBjJl8mg
uVXXMz4CSP4+psPef+Df9Os2ebv4q67byJ7/HAmjTBzJQw/fUZVh/z6x/LIvIYlMQgJ2Bcq
ckBT3e76fAUDSJ6QBx/bhqUfvDjuMKpGSksJVd1/Bjef+lfMP/yNH9O9Ni9bNi5T56+Y2+F
CX84ag/8fqTb3LZHZcC0Ov4w8jwUf/pt85x9D8Cef
Q9sDDOf30Aey7b+siZSZN+o5uWKCIGHq/+GgV5x059+M9/P+P+DD++24t3PfiLzRdL7r3OI4lS
5ZV9SlUmJSUFB79naQ3nNNOACOnfsy6mn9qNNm72KlJkyeRqH9OzH7nM3HPnGGpdL2X
tIwd170S3rsfQpfNRdOzUjkMO6Yci6IaJWmipKmS+gNIa
ilppqSXge+AZpJuC7Z9IellSdcFZfeU9JGkCZI+l9RGUg+gH/BQ8Dl7hna2pejcvi5l1dtk
57DCqRJv2+5A7dwEL5y0kPy+fT7P7lo16wpfZ9bMxMwAMDMya2QCULNWJgvTwn9z9X79iNs
OXlszahbHXTtUsHfvppLnPmzCMvL4lB+33fnLuCq5Dz3w8jeZcKUlyzXJS0vj
41GfA7B27brCctw8HMvunn5k7dz55eXkMHTqcfv2OKlKmcxGZ4Z7dz55eXm88857HHd8nyJ
lxoz5uvAcs437liZZj
```

```
YDg7yKzBhkZ6dSokUF6elrSJ2ezWJmXsETpHtJNwAFm1l5SGlDTzFZJqg98LWloUK41cL6
ZfS2pC3AycCCQDkwEJgTlBgN/MrMfJXUD/mVmRwTHed/M3qnKk3Pbtnvj+ixduKRwfemip
ezXoc0W5Qac349TLzmF9Iw0rj79egA++2AMPfv24D8T36LGTjUY9PenWL1idZXFXtGaZDV
ifs6CwvWc3IV07dJhm+UvvOBMPho+GoDWrfdgxYpVvP3WM7Rs2ZxPRn3OzbfcS0E1uDexN
U2aNCInd2Hhem7uIrp0ab/N8ueffzojR3wGxJPTmM++ZtbscUgw+OlXmDnzp0qPebtUg/+
folRDSiTgXklTgI+BLGBTO8zPZvZ18Ppg4F0z22Bmq4H3ACTVBnoAb0uaBDwNNK7KE3AVb
8hLQzm753k8fe+znHvl2QDs274NsYICTu50Omd2P5fTBp5C4+bR+L/6rLNOonOnA3n4kSc
BSEtLo2fPrtxw410c1P1YWu3RnPPPOy3kKKvG6WcMoEPHtvzfPwYDsMceLdinzZ7s07o7e
+/VnUMP606PHl1CjrIUVlD2pQwkHR20Hs2SdNNW9rcIWqKmSPpUUqk3HKOakM4Gdgc6mVl
7YDGQGexbW4b3pwArzKx9wrJvWT5Y0kBJ4yWNf/bl139X8K58li5cxu6NGxSu795od5YuX
L7N8p+8O5qeR8Wb9HoPOIJxn2YTy4+xYvkKvsuexj7t9q70mCvLgtxFNGvapHC9aVZjFix
YtEW53kccws03XcmAky5g48aNAOTmLGTy5GnMmTOPWCzGu0OH06FD2yqLvaItWLCosDkSI
Cur0Va/i16HH8wNN1zG6adeUvhdnNDvKLLHTWLt2nWsXbuOkSM+pWu3jlUW++9SgfeQJKU
Cg4BjiN8zP3MrnbkeBl42s3bAncB9pR03SglpNbDppkkdYImZ5Uk6HGixjfd8CZwgKTOoF
R0PYGargDmSToXCDhAHbuVztmBmg82ss5l1/sN51b/HVnUwc/JMmrbKolGzRqSlp3FE/16
MHTm2SJmsVpt7Tx3Uuxu5c3IAWLJgCR17xJtxMnfKZL+O+zLvp3lVF3wFyx4/ib32akXLl
s1IT0/ntNP68977I4qUad9+f/416H5OPOlCli5dXuS9dXatQ/36uwFweK+DmTHjhyqNvyJ
NmDCFPfdqSYsWTUlPT+eUU05g2AcfFynT7sD9ePyJezjt1EuKfBc583Pp2bMrqampQc2xG
zNnzqrqUyifiu1l1xWYZWazzWwj8AbQv1iZ/YBPgtejt7J/C5G5h2RmyyV9Kek6I2k
qMB74fhvvyQ7uCU0hXouaCqwMdp8NPCnpVuL3l94AJgf/fUbSlcApZpaUDcvX33E/2d9OY
cWKVfQecA5/vvhcTi52c3tHEYsV8NhtT/DQ/x9E/eTkpLCh29wKeYNZ1rJrqhxbldUVtkXW+hsHrtyq2LA9gIAIUA3EaZyPDmwVFr+VX/s6QLCVA6B3T5jb1sBTfe5z2S4PKBjTxN
RPuDKb/iIjjDuZwEuEw2GA6oT6nhaowHsH1+YzjZ1pJFxMBrhzJjZwNDz/1t9lUz2/dV5zJ
ot7ZZTmWEIXXzh6w/jVtZSVQcsL+CdTRZS+jzTEnfq4EVyAHk/zfDUVvc7NqJ4bYkfP/s7/b
y8YGQLGlhvrTLbavIsCWjhQZyd3yaYvVPeYvXOJm4smL/KfjPtMIkF/h7OCCPSsfcrTYwsX
FRaDeXf/XRxI83j9JLR6HnV+yMK9DmxJmA5ZvGvPkF5HDy9L8dUwT+wYkmeS8R3wO55lhzW
EJkq6NbRq+f4SB/k14D6JEIWbO9ElnL3j+54NjLSQdF9+Wih1fMuFSAUk7jNyHAnqA+XKDR
BWnHOadExr0k9HS1uE8lhuS2BnIXSeexDekdz6tnOuH/9fvpN1PZlLP0i1GjzBldEv95x+/x
LbNc/FWkVVmGpU8zNNDXMrQZoY6LOobxUrPBAbrlUQrdb2hAr/ygzZIi6CfwGCzN6YXK9Ahv
Wv6H8JqS9NOiIjTWM0bgtzXFQp9x0nhXRsxwZqfTF6lNu6u+/A/z2xwWj0Ze7bJGnD77YN/b
k5Zf7Tx1B1gm0LFYxLqzCk3VZS+lnWQEnfQ9KGl/f4WxB65BmGlqzhF/CnQNv/k63uPZq1tU
ZbN1NiiCY8PcH1l0dPqVqhRmFwBbb7wXQJr4rPyGm1s8lCY9qz8gCvbyBj31WuC3t4LBEKVo
3+82U0+jFa9kkwuxOrVcKvMNphm9FWPkjMq8ZpRxn3cKTW9SCNJU8OgrdMWdBZM1cQx8Izyr
```

```
tQZYyW1lDRD0jOSpkkaIWknSZ9K6hyUqS9pbvD6AklDJI2UNFfS5ZKukfStpK8l7VbCZ+0
l6WNJkyVNlLRnsKu2pHckfS/pVUkKyt8uKVvSd5IGJ2z/VNL/SRoPXCXpBEnfBDF8LKlhU
K62pBckTZU0RdLJku4HdpI0SdKrQblzJI0Ltj0tKTXYvkbSI5ImA90l3S9penCsUH/d2rT
fh9y5C1g4byH5efl88u6nHNy36NQk69asK3ydWTOz8ME5MyOzZiapqSnUyKxBXl4+axPK7
mg6t29LnV12DjuMKvHdD7Np3rgBTRs1ID09jaMP7cror78tUmb2/AV0a7cvAF3b7bvFfoC
RX46nZ6e27JRZo0rirgw127fmt7mL2DhvMZaXz6/vfU6dvl2LlKl3Vl+WvTyM2Mq1AOQvX
1m4b6e2e5Jef1dWj5lUpXH/XpZvZV7CUtYpzFsDg8xsf2AFcHIp5Q8ATgK6APcA68ysA/A
VcF4J73s1+JwDgR7AwmB7B+AvwH7AHmye9OmfZtbFzA4AdgISm/UyzKyzmT0CfAEcFMTwB
nBDUOY2YKWZtTWzdsAnZnYTsN7M2pvZ2cHUu6cDB5tZeyAGnB28vxbwTRDvDOBEYP/gWHe
X8h1Vqt0b12fpws1T1y9dtJTdG9fbotyA8/vx6hcv86dbLuHx2wcB8NkHY9iwbgP/mfgWb
457lTeffpvVK1ZXWeyu8ixe/isNd998Tdiw/m4sWf5rkTJ7t2rGx2MnADDqqwmsXb+BFav
WFCnz4ZhvOOawbpUfcCXKaFSPjQuWFa5vXLic9IZF/43UaNWEGns0ofV/72fvIQ+y82Ed4
jsksm69kNy7X6jKkLdPQTmWkJQ1Ic0xs02XAROAlqWUH21mq81sKbASeC/YPnVb75W0M5B
lZv8DMLMNZrbpsnycmeWYWYQEwKeEYhwc1n6nAEUBi+8GbCa+bAsODctcnlDsSGLSpkJkV/
ZcZ1xvoBGRLmhSs7xHsiwGbGp1XAhuA5ySdBGy1SiFpoKTxksYvWJu7tSJVashLQzm753k
8fe+znHtlPM/u274NsYICTu50Omd2P5fTBp5C4+aNQ47UVZVrLzqdCd/N5LQr72D81Jk0q
FeXlJTNPxVLf1nBrLk59Oh4QIhRVg2lpVKjZRN+PO0W5l7xMM0fuJzUXWpR/7xjWDV6Anm
LlocdYplZQdmXsJT1HtJvCa9jxGsj+WxOaJkllC9IWC8ox2eW9PlpkjKBfwGdzWy+pL8Vi
2NtwusngEfNbGjQSeJv5fhsAS+Z2c1b2bdh032jYI75rsQT1inA5cSTZBFmNhgYDNCr6ZG
VVjdeunAZuzduULi+e6PdWbpw2/94Pnl3NFffexUAvQccwbhPs4nlx1ixfAXfZU9jn3Z7s
3Dewm2+31UPDevVZfHSXwrXFy/7hQb16hYp06BeXf5xyxUArFu/gY/HTmCX2jUL9w//fBx
HdO9Eelr1vgW9cdFyMprUL1zPaFyFyPvMVF/43kLVzO2m9/gPwYG+cv4bc5udRo2ZhaHdtQu
+t+1D/3GFJr7YTS0yhYt4EF979c1adRdtWgU0NZa0hbM5d4zQHiP8DbxcxWAzmSBgBIqiG
pZglv2ZR8lkmqXUoMdYBN1ZHzE7aPBC7btCJp07/MPEnpwetRwCmSGgRldpPUovgHBDHUC
eaZvxo4sIR4Kt3MyTNp2iqLRs0akZaexhH9ezF25NgiZbJabb6RfVDvbuTOyQFgyYIldOz
RHoDMnTLZr+O+zPtpHq7623/vVvy8YAk5i5aSl5fPR2PG0atbhyJlfl25moKC+K/Xs29/w
Il9Dimyf0dorgNYN/lHarRqTEazBig9jbonHMLKkeOKlFkx/Gtqd4/XBFPr7kyNVln8Nm8
xP1/1KNO6/4HpBw8k9+4X+OU/o5M7GbFj1ZC25mHgLUkDgQ8qKJ5zgacl3QnkAaduq6CZr
ZD0DPAdsAjILuG4fwPelvQr8AnQKth+NzBI0/5LvAYzRdLE4DSrcAISSlBXJc
BPxf7jJ2Bd4Oam4Bryn7aFS8WK+Cx257goVfvJyUlhQ/f/Ii5P/zMzBz5Fece
EF/OvXsSCw/n9Ur13Df1Q8CMOTFd7nx0et5YdSzE+nUl1/x/1kfzuFFSt
W0XvAOfz54nM5+YSjwg6rUqSlpvLXP53Npbc/QqyggAF9DmGvFlkM+vf/2K91Sw7v1oHsq
d/z+EvvIImOB+zNLZeeW/j+3MXLWLz0FzofsE+IZ1FBYgXk3DaYPV/5W7zb95uj2PDDfBp
dcxbrps5i1chxrP7sW3Y5tANtRv0TYjFy73mRWDW9n1ruvsYhUHUYknxHVZlNdtXJyElJ/
vxGFSr4dVHYISSN6b0fCDuEpNJh3rvanvcv6X1YmX9vGoz6rNTPknQ08BiQCjxrZvcX2/8
P4PBgtSbQwMx2LemY1bsR2DnnXJlUFNc8OjLIKAPkEO809dQM5te+HlmVyeUv4J4b+kSb
c89pN9N0qDgmZ7E5cIwYnHOuUgwlX0pXVdglpnNNrONxB+n6V9C+TOBUod7CaWGZGaXlV7
KOedcRSlPDSnoGzAwIfwN1HGCrPV2CTmCtiN+/L5E32TnnXARYQdlvQSU+n
lIBzgDeKcvQap6QnHMuAgpi29UnorhcoFnCelawLbkV1DuITnnnKtaFfwcUjb
QWlIrSRnEk87QQHqEh82rlReQ3LOuQgoT5NdaoqTCceLftv5sWvAM6Xgz2
5SczgDesDI+X+QJyTnnIqCiHzkNRqUZVmzb7cXW8vCKjIGlJl8YTknHMRUMG
dGiqFJyTnnIsAryE555xLCla2ERhC5QnJOQnJOeciIMxpJcrKE5JzzkVAgdeQnHPOJQNvsnPOOZcUyvKX2mDeQnHPOJQNvsnPOO
ZcUvJedc865pOC97JxzziUFv4fknHMuKfg9JOeccxSKPB
ODa4kXyyZEXYISaHg10Vhh5A0Uuo2CjuEpFGvwZqwQ9iheA3JOedcUvBODc4555KC15Ccc
84lhWrQyc4TknPORUGsIsFUlCdSklXkLZjUmtDhDXyEEk55yKgGswEFFTgTSRJqcAgoA+QA2RLGmpm0xPKtAZuBg42s2s18lNSjtu
F5Dcs65CChAZV7KoCswy8xmm9lG4A2gf7EylwCDzOxXADNbUtpBPSE551wEVHCTXRYwP2E
9J9iWaG9gb0lfSvpa0tGlHdSb7JxzLgJiZbw3BCBpIDAwYdNgMxtczo9olDvYCmwBhJb
c1sRUlvcM45t4MrTy+7IPmUlIBygWYJ602DbYlygG/MLA+YI+kH4gkqe1sH9SY755yLgIJ
yLGWQDbSW1EpSBnAGMLRYmSHEa0dIqk+8YZvmSLgeGA6nA8
2Y2TdKdwHgzGxrs6ytpOhADrjez5SUd1xOSc85FQEXPPmFmw4BhxbbdnvDagGuDpUw8ITnnXARU
nXASUsTt3qDwhOedcBMTCDqAMPCE551wEFMhrSM4555KATz/hnHMuKVSDuVU9ITnnXARUd
```

C+7yuAJyTnnIqA8QweFxROSc85FgNeQXKiO6tuLRx+9k9SUFJ5/4XUefGhQkf1/uWogF11
0Jvn5+Sxb+gt/GHgN8+bFh6Nq1qwJg596mKbNmmBmnNDvXH7+OSeM06gQX0yYygODX6Ogo
ICT+h7KxaceV2T/giXLuP3/nufXVaupU7sW9143kEb1d2PclBk89MzrheXm5CzkwRsu5Yj
uHav6FKrErfc+ypgvx7Fb3V0Z8u+nwg6n0mV270Ld6y6DlBTWDhnGqpfe2KJMzSMPo87A8
zEz8n78ieW33gvArlcOJPPgbihFbPhmAr8+PGiL9yYTv4cUAZLWmFntsOMoLiUlhccfu4e
jjz2TnJyFfP3VMN57fwQzZvxYWGbSpO/odtAxrF+/gT8OPI/777uVs86+FIAXn3+M++5/n
I9HfU6tWjUpKKgOf85bF4sVcO+TrzD47utoWG83zrz6Tnp1a8+ezTePlv/Ic29yQu8e9O/
dk28mT+fxl97h3msH0rXdvrz9xJ0ArFy9huMuuYnuHfYP61Qq3YBj+3DWyf3461OPhx1K5
UtJoe6NV7LkshuILV5Ko5f/xboxX5E/5+fCImnNstjlwjNZdPGV2Oo1pNTdFYCMdvtR48D
9WXTmJQA0fPYxanQ6kN8mTA7lVMqiOvSy88FVy0FStUngXbt04Kef5jJnzjzy8vJ466136
XfCUUXKfPrZWNav3wDAN+Mm0DSrMQD77tuatLQ0Ph71OQBr164rLFcdfffDbJo3bkDTRg1
IT0/j6EO7Mvrrb4uUmT1/Ad3a7QtA13b7brEfYOSX4+nZqS07ZdaokrjD0Ll9W+rssnPYY
VSJjP3bkD8/l1juQsjPZ92I0dQ8rEeRMrVPPI7Vbw3FVq8BoODXYOYEA2VkoPQ0lJ4Oaan
Elv9a1adQLgUq+xKWUBOSpJaSZkh6JphzfYSknSR9KqlzUKa+pLnB6wskDZE0UtJcSZdLu
kbSt8EEULtt43MaSJoQvD5QkklqHqz/JKlmEMsnkqZIGpWw/0VJT0n6BngwGN32K0lTJd2
d8BmNJY2RNEnSd5IOqdxvr2RNshoxP2dB4XpO7kKaNGm0zfIXXnAmHw0fDUDr1nuwYsUq3
n7rGbLHDeeB+24lJaX6Xrssv4rDXff/KfRsP5uLCn247F3q2Z8PHYCAKO+msDa9RtYsWp
NkTIfjvmGYw7rVvkBuyqR2qA+scVLC9fzlywltUH9ImXSmjclvUVTGj73GA1feILM7l0A2
Dh1OhvGTyLro7fJGv4WG74eT/7ceVUaf3lV8GjflSIZfmVaE5/mdn9gBXByKeUPAE4CugD
3AOvMrAPwFXDe1t4QTJ2bKWkX4BBgPHCIpBbAEjNbBzwBvGRm7YBXgccTDtEU6GFm1wCPA
U+aWVtgYUKZs4DhZtYeOBCYVNYvIGxnnXUSnTsdyMOPPAlAlAWloaPXt25YYb7+Kg7sfSao/
mnH/eaSFHWbmuveh0Jnw3k9OuvIPxU2fSoF7dIkl46S8rmDU3hx4dDwgxSlfVlJpKWrMsF
g+8hmW33MNut1yDatcirWkT0ls1J/fY08k95nQyO3egRvu2YYdbopjKvoQlGZqg5pjZph/
vCUDLUsqPNrPVwGpJK4H3gu1TgXYlvG8scDBwKHAvcDQg4PNgf3fiiQ7gFeDBhPe+bWabh
oI6mM1J8xXggeB1NvC8pHRgSMI5FZE4E6NS65CSUqvks/2dFuQuolnTJoXrTbMas2DBoi3
K9T7iEG6+6UqO6H0yGzduBCA3ZyGTJ09jzpz4Fd+7Q4fTrWtHXnhxyxu+1UHDenVZvPSXw
vXFy36hQb26Rco0qFeXf9xyBQDr1m/g47ET2KV2zcL9wz8fxxHdO5Gelgz/ZFxFiC1ZRmr
D3QvX0xrsTmzJsiJl8pcsZeN3MyAWI7ZgEfnzckhv3jR+v2jqDCxoyl4/drB2YzwAAB44S
URBVBwZ7fbjt0lTq/QcyqM63AVOhhrSbwmvY8STZD6bY8ssoXxBwnoBJSfYMcRrRy2Ad4n
XYnqyOSGVZG2x9S3uD5rZGOLJLhd4UdK2amuDzayzmXWurGQEkD1+Envt1YqWLZuRnp7Oa
af15733RxQp0779/vxr0P2ceNKFLF26vMh76+xah/r1481ch/c6mBkzfqi0WCvb/nu34uc
FS8hZtJS8vHw+GjOOXt06FCnz68rVhR03nn37A07sU7TF1Zvrdjwbp39PerMsUps0grQ0a
vY9nPVjxhYps/7TL6nRqT0AKXV2Ia15U/JzFxJbtITMju0gNQVSU8ns2I78Od5kt72S9XJ
vLtAJGAecUkHH/Jx4E98YMyuQ9AtwLHBzsH8s8VkPXwHOZtuJ6sug3L+DcgAEzX85ZvaMp
BpAR+DlCoq93GKxGFf95VaGffAqaSkpvPTilEnv7hPjSm0yf/gN/u+M6xk+YzPvvj+SB+26jdu1avPH
60wDMm5fDSSdfQCwW48qr/o8Rn7yFJP77byn+/fdXQ/nwzc9Pj0NjbsftCiXm5jVd5MBCYCyf7Kr9aMmh
ldWzf6c2AGqgwVSIOWgcVW11WiD5lqhUp8gt6x1iLX/jN7OVEZfIJ6hF/2r+Ri1L/GKXhXXmUFyBBNWd37N4
P+NG12lxiCewlpZ/GJB+S5J4vA2G0s9PSOO0UqoUQFfXrmqhnHyZi0tzunDk9vEqIdrq+YTmrrM7xwvRkDdv

PzpyBOSc85Fgnf7ds45lxS8l51zzrmkkO8JyTnnXDLwGpJzzrmk4N2+nXPOJQWrBt2+U8I
OwDnnXOUrwMq8lIWkoyXNlDRL0k0llDtZkknqXNoxvYbknHMRUJFDB0lKBQYBfYAcIFvSU
DObXqzczsBVwDdlOa7XkJxzLgIquIbUFZhlZrPNbCPwBtB/K+XuAh4ANpTloJ6QnHMuAsy
szEsZZAHzE9Zzgm2FJHUEmpnZB2WN0ROSc85FQEE5FkkDJY1PWAaW57MkpQCPAteW531+D
8k55yKgPM8hmdlgYHAJRXKBZgnrTYNtm+wMHAB8KgmgETBUUj8zG7+tg3pCcs65CKjgsey
ygdaSWhFPRGcAZ23aaWYrgfqb1iV9ClxXUjICT0jOORcJMau4R2PNLF/S5cBwIBV43symS
boTGG9mQ3/PcT0hOedcBFT00EFmNgwYVmzb7dso26ssx/SE5JxzEeAT9DnnnEsKyZ+OPCE
551wkVIcJ+nbo55Ak/TXsGMJ0ZJ9DmThpFJOnjuaaa/+0xf7Lr7iY8RNG8PU3H/L+B/+mW
bPNz7XddfdNZI8fzoSJI3no4TuqMuxKsfNhHdh39L/Yb8xTNPzzyVsts+vxB9Nm1D9p8/E
TtHj8miL7UmrvxP7fPEfTO8v1OEZSyuzehcb/eZHG/3uZXc4/Y6tlah55GI3fep5Gbz5Hv
bs3/zPa9cqBNHrzORq//Tx1r7usqkIOxa33Psqhx53BgHO2/LdTHVX0WHaVYYdOSEBkE1J
KSgqP/uNOThpwAZ079uXUU/vRps1eRcpMmTyNQ3r246BuxzBkyIfcfU98fMRu3TpyUPdOd
Ot6DF06H0XHTu045JBuYZxGxUhJodndf+Sn8//OjN6XU7ffIWS2blakSI2WjWn451P48aQ
b+f7IK8j9+3NF9je+7mzWfDOtKqOuHCkp1L3xSpZceTMLT72ImkcdQVqrFkWKpDXLYpcLz
2TRxVey6PSL+fWRfwGQ0W4/ahy4P4vOvISFp/+BjP3aUKPTgWGcRZUYcGwfnnr07rDDqDA
xKyjzEpYyJyRJLSXNkPSMpGmSRkjaSdKnm0ZxlVRf0tzg9QWShkgaKWmupMslXSPpW0lfS
9qthM+6UtJ0SVMkvRFsqyXpeUnjgmP0T/ic/0r6SNKPkh4Mtt8P7CRpkqRXg23nBO+fJOn
pYIBAJK2RdI+kyUFsDYPtDSX9L9g+WVKPbR0nWF6U9J2kqZKuLv//HRWnc+cDmf3Tz8ydO
5+8vDzeeec9jju+T5EyY8Z8zfr18SGmxo37liZZjYD4ECOZmTXIyEinRo0M0tPTWLJkWZW
fQ0Wp2b41v81dxMZ5i7G8fH5973Pq9O1apEy9s/qy7OVhxFauBSB/+crCfTu13ZP0+ruye
sykKo27MmTs34b8+bnEchdCfj7rRoym5mE9ipSpfeJxrH5rKLZ6DQAFv66I7zBQRgZKT0P
p6ZCWSmz5r1V9ClWmc/u21Nll57DDqDBWjv+Fpbw1pNbAIDPbH1gBbL3tY7MDgJOALsA9w
Doz6wB8BZxXwvtuAjqYWTtgU335FuAhdKJKlWsK89cDrQFjhdUjMzuwlYb2btzex
sSfsGZQ42s/ZADDg7eH8t4GszOxAYA1wSbH8c+CzY3hGYVsJx2gNNnaAmbUFXijlu6lUT
Zo0Iid3YeF6bu4imjRptM3y559/OiNHfAbEk9OYz75m1uxxzJr9DaM+/pyZM3+q9JgrS0a
jemxcsDmhbly4nPSG9YqUqdGqCTX2aELr/97P3kMeZOfDOsR3SGTdeiG5d4f6f2eFSW1Qn
9jipYXr+UuWktqgfpEyac2bkt6iKQ2fe4yGLzxBZvcuAGycOp0N4yeR9dHbZA1/iw1fjyd
/7rwqjd/9fhU8ll2lKG+nhjlmtukycQLQspTyo81sNbBa0krgvWD7VKBe+bArwqaQgwJ
NjWF+gn6bpgPRNoHrweFTwZjKTpQAuKDvwH0BvoRHyYdICdgCXBvo3A+wnntakqcQRB4jS
zGLBS0rnbOM57wB6SngA+AEaUcH5J5fQzBtChY1uO7hu/n7DHHi3e7NO6OwBD33+FH
j26MHZsdphhViqlpVKjZRN+PO0WMhrXo/Xb9/F93yupe+JhrBo9gbxFy8MOscooNZW0Zlk
sHngNqQ13p+Hgf7DwjD+Qumsd0ls1J/fY0wFoMOghNrRvy2+TpoYcsSuL6tCpobwJ6beE1
zHiP8b5bK5pZZZQviBhvaCUzz4OOBQ4AbhFUltAwMlmNjOxoKRuW4lra8cW8JKZ3byVfXm
2+bJgW+8v9TiSDgSOIl6rOw24aCtlBgIDATLS65GeVjlNAgsWLKJpVuPC9aysRixYsGiLc
r0OP5gbbriMo486g40bNwJwQr+jyB43ibVr1wEwcsSndO3WsdompI2LlpPRZHMtMtIKNxPfI
WF00weQuXs/bbHyA/xsb5S/htTi41WjamVsc21O66H/XPPYbUWjuh9DQK1m1gwf0vV/VpV
IjYkmWkNty9cD2twe7EijXH5i9ZysbvZkAsRmzBIvLn5ZDevCk1Oh3Ib1NnYEEz7/qx48h
ot58npGoiKjPGziVeYwA4ZXsPFowS28zMRgM3AnWA2sSHqLhCQbVEUocyHC5PUnrwehRwi
qQGwft3k9Ri228tfM+lQflUSXW2dRxJ9YEUM/sPcCvxJr4tmNlgM+tsZp0rKxBTJgwhT3
3akmLFk1JT0/nlFNOYNgHHxcxcp0+7A/Xj8iXs47dRLWLp08w90zvxcevbsSmpqKmlpafTs2
Y2ZM2dVWqyVbd3kmmRqjEZzRqg9DTqnnAIK0eOK1JmxfCvqd39AABS6+5MjVZZ/DZvMT9
f9SjTuv+B6QcPJPfuF/jlP6OrbTIC2Dj9e9KbZZHapBGkpVGz7+GsHzO2SJn1n35JjU7tA
UipswtpzZuSn7uQ2KIlZHZsB6kpkJpKZsd25M/xJrvqIkZBmZewVMRzSA8DbwVX/mWe96I
EqcC/gx9/AY+b2QpJdwH/BH0wJktYc4PhShIeCwNmnvcI4jL1Oli4
jWnS83sq20cZz3wQrANYGs1sSoTi8W49po7GDL0ZVJTU3jl5beZ5MeNHbr3taiZOnMqwDz7
mnntupnatWrzy6iAA5s9fwOmnXsL//9TL4IwICc2waz5
yt/Q6kpLH9zFBt+mE+ja85i3dRZrBo5jtWffcsuh3agz/RLG85jtWffcsuh3agzah/QixG7j0vEluxOuzIK16sgF8
eeoIGTzwAqSmsHfohebN/ps4fL2DjjJmsH/MVG77KJvOgzjR+63msIMaKxwdTsHIV60/aNo
UaXDjR+41kwWP9VNus//yrsM6o0199xP9nfTmHFilX8OI3XWWIbI6q8Ny+4aiww/rdqsNIDao
O1bgdVe2arfzLBz6vX9LtxGip12BN2CEkjcYfPRN2CEklvf4e2p7379+wW5l/b6Yt/ma7P
uv38pEanHMuAqpDDSnUhCRpEHBwsc2PmdmO0cfWOeeSRJjPF5VVqAnJzHbssUeccy5JeA3
JOedcUghzSKCy8oTknHMR4E12zjnnkoLnZ551wy2BGHDnLOOVcNVYdnTj0hOedcBFSHG

```
tKOPkGfc845IFZQUOalLCQdLWmmpFmSbtrK/j8Fc8NNkvSFpP1KO6YnJOeci4CKnKAvmNx
0EHAMsB9w5lYSzmtm1jaYN+5B4NHSjusJyTnnIqCCJ+jrCswys9lmthF4A+hf7PNWJazWg
tIznd9Dcs65CCjPPaTEedsCg81scMJ6FkUnQc0Bum3lOJcB1wAZxCc8LZEnJOeci4Dy9LI
Lks/gUguWfpxBwCBJZxGfJ+78ksp7QnLOuQgoa2eFMsoFmiWsNw22bcsbwJOlHdTvITnnX
AQUYGVeyiAbaC2plaQM4AxgaGIBSa0TVo8DfiztoF5Dcs65CKjIB2PNLF/S5cBw4rN8P29
m0yTdCYw3s6HA5ZKOJD6r9q+U0lwHnpCccy4SKnr6CTMbBgwrtu32hNdXlfeYnpCccy4Cf
LRv55xzScEn6HPOOZcUCnz6Ceecc8nAR/t2zjmXFKpDQlJ1CNJVHkkDiw0JEln+XWzm38V
m/l1UHX8w1g0svUhk+HexmX8Xm/l3UUU8ITnnnEsKnpCcc84lBU9IztvGN/PvYjP/Ljbz7
6KKeKcG55xzScFrSM4555KCJyTnnHNJwROSc865pOAJKWIktSrLNuecq2qekKLnP1vZ9k6
VR5EkJD0oaRdJ6ZJGSVoq6Zyw4wqDpIMl1QpenyPpUUktwo6rqkm6KvibkKTnJE2U1Dfsu
KLAE1JESGoj6WSgjqSTEpYLgMyQwwtTXzNbBRwPzAX2Aq4PNaLwPAmsk3QgcC3wE/ByuCG
F4qLgb6IvUBc4F7g/3JCiwQdXjY59iP/o7gqckLB9NXBJKBElh/Tgv8cBb5vZSklhxhOmf
DMzSf2Bf5rZc5IuDjuoEGz6AzgWeCWYmjuyfxRVyRNSRJjZu8C7krqb2Vdhx5NEhkr6Hlg
PXCppd2BDyDGFZbWkm4FzgEMlpbA5YUfJBEkjgFbAzZJ2BpJ/MqEdgD8YGzGS9ibeNNPQz
A6Q1A7oZ2Z3hxxalQt+cA8CvgdWmlksuIeys5ktCje6qiepEXAWkG1mn0tqDvQys0g12wV
/F+2B2Wa2QlI9IMvMpoQc2g7PE1LESPqM+D2Sp82sQ7DtOzM7INzIwiHp203fg3MAQfPc2
cAeZnZnkJgbmdm4kEPb4XmnhuipuZV/WPmhRJIcRkk6Ocr3CCStlrRqW0vY8YXgX0B34Mx
gfTUwKLxwosPvIUXPMkl7AgYg6RRgYbghheqPwDVATNJ64je0zcx2CTesqmNmOwNIuov43
8IrxL+Hs4HGIYYWlm5m1lHStwBm9qukjLCDigJPSNFzGfHRi9tIygXmEL+JHUmbfowdEL+
XeGDC+pOSJgO3hxVQSPIkpbL5om13vFNDlfAmu4gxs9lmdiSwO9DGzHqa2dyQwwpN8PDjO
ZJuC9abSeoadlwhWSvpbEmpklIknQ2sDTuoEDwO/A9oIOke4Avg3nBDigbv1BAxkq7Zyua
VwAQzm1TV8YRN0pPEr36PMLN9JdUFRphZl5BDq3KSWgKPAQcTrx18Cfwk+lihcsktoAvYk3X
Y4ysxkhhxQJ3mQXPZ2D5b1g/XhgCvAnSW+b2YOhRRYOv18ABE1Ul5tZ/7BjCZukg4BpZjY
oWN9FUjcz+ybk0HZ43mQXPU2BjmZ2rZldC3QCGgCHAheEGVhI/H4BYGYxoGfYcSSJJ4E1C
etrgm2uknkNKXoaAL8lrOcRf0h2vaTftvGeHVnx+wWnALeFG1JovpU0FHibhHtHZvbf8EI
KhSzhXoaZFUjy38oq4F9y9LwKfCPp3WD9BOC1YISC6eGFFQ4ze1XSBDbfLxgQ4fsFmcBy4
IiEbQZELSHNlnQlm2tFfwZmhxhPZHinhggJHv5sCjQkfuMa4EszGx9eVOGS9IqZnVvaNhc
dkhoQrzkfQTwhjyLeuWNJqIFFgCekiJE01czahh1HspA00cw6JqynAlPNbL8QwwqFpKbAE
2y+WPkcuMrMcsKLykWJN9lFz0RJXcwsO+xAwhSMav1XYKeE4XEEbCT+4HAUvQC8BpwarJ8
TbOsTWkQhCDq2XAK0JOE30swuCiumqPAaUsQEUy3sBfxM/Mb1pqFy2oUaWEgk3WdmN4cdR
zKQNMnM2pe2bUcnaSzx2uEEILZpu5ltbbZlV4G8hhQ9R4UdQJIZJ6mOma0EkLQr8SkXhoQ
cVxiWB9O3vx6sn0m8k0PU1DSzG8MOIor8OaSIMbOfzexn4hPSWcISVXdsSkYAZrYCuCPEe
MJ0EXAasIj4IKunABeGGlE43pd0bNhBRJE32UWMpH7AI0ATYAnQAphhhZvuHGlhIJE0p3lw
Z1Y4fkjLNLKqz5RaStBqoRfx+4kYiOAJ8WLzJLnruIj5L6sdm1kHS4UR4tG9gvKRH2Tzfz
WXE7x1E0XeSFhO/f/I58EVi7TEqfAT48HiTXfTkmdlyIEVSipmNJj62XVRdQfwq+M1g+Y1
4UoocM9uL+H2jqcBxwGRJURxw10eAD4nXkKJnhaTawBjgVUlLKDpuV6SY2VrgprDjSAbBc
0gHA4cABwLTiE+9EDX/IhgBnniLwhriNejIjQBf1TwhRc9kYB1wNfEZQesAtUONKETBMyc
3APsTHzoHADM7Yptv2nHNA7KBe83sT2EHEyIfAT4k3mQXPYebWYGZ5ZvZS2b2ONG+8nsV+
B5oBfwdmEv8RzmKOgAvA2dJ+krSy5I4IuDjuoEPgI8CHxXnYRIelS4oNE7gnMSti1M/Hx7CL
ZsUHSBDPrlNjbTlJ2FCfoAwiac3sSb7Y7B8DMWoQaVBULZso9HegIvEQwAryZvRVqYBHgT
XbR8RrwIXAfRe+ZrDazX8IJKSnkBf9dKOk4YAGwW4jxhEbSeKAGsGmkgkODZ9YixUeAD4/
XkFykSTqe+I9vM+IDi+4C/N3MhoYaWAgk7W5mzSvVSVMYXBR4APj9eQXGQF9wlam
9n7wErg8JBDClVJyShwFfEmrB1dkYfEg7+TTiHFEineqcFFVjBt95lhx1GNKOwAKpOkm4N
RGtpJWiVpdbC+BHi3lLe7CuBNdi7SJP0DSCf+UGzqqwQYSRWfO2pH5SPAh8cTkos0S
aO3stki+hxSiSR9a2Ydwo6jKkjKIj7OY+J8SGPCiyga/B6Sizha/R9o3L6MuwAqoKk+4E
zgOlsng/JiI9u4iqR15Bc5AXdvYuP1HBneBGFI5gL6jy2nCn1yrBiCoOkmUA7M/st7Fiix
mtILtIkPQXUJN7D7lniD0GOCzWo8AwDviY+uGqURyaYTfy+oiekKuY1JBdpm0ZoSPhvbeB
DMzsk7NiqWlQ6LZRG0n+IDy47ioSkFLWaYhi8huSibn3w33WSmhCfsrtxiPGE6RVJlwDvU
/SHOGojeQwNFlfFPCG5qHs/uHfyEDCR+M3rZ8MNKTQbiX8Pt7B5WnsD9ggtohCY2UuSdgK
am9nMsOOJEm+ycy4gqQaQGcVZUgEkzQa6mtmysGMJk6QTgIeBDDNrJak9cKeZnRhyaFXi
5Bc5EnqQULPMkmY2cuhBhWOWcTnyoq6vwFdgU8BzGySpEjVEsPiCclFmqRXiE/JMiz5x
EMSGtBSYFDwtH+WZ+npmtlIqMlBTlXodVxhOSi7rOwH7mbdcAQ4Il6qZJOgtIldQauJL4l
```

```
Byukvk9JBdpkt4GrjSzhWHH4pKDpJrEO3b0JT6g7HDgLjPbEGpgEeAJyUWSpPeIN83tDLQ
n/jBsYjNV5G5gS5rD5t51hcwssvdPgqknapnZqrBjiQJvsnNR9XDYASShzgmvM4FTieDsu
ZJeA/5E/J5iNrCLpMfM7KFwI9vxeQ3JRZqkVsDCTc0xwfMnDc1sbqiBJQlJE8wsUpPTSZp
kZu0lnQ10BG4CJphZu5BD2+F5DclF3dtAj4T1WLCtSzjhhEdS4rBBKcRrTFH8jUiXlA4MA
P5pZnmS/Mq9CkTxj825RGlmtnHTipltlJQRZkAheoTN95DygbnEm+2i5mni5z4ZGCOpBeD
3kKqAN9m5SJM0EnjCzIYG6/2J97rrHW5kVU9SJnAyRaefsChOxZFI8QeSUs0sP1g/38xeC
jmsHZInJBdpkvYEXgWaBJtygHPN7KfwogqHpI+AFcTH9Nv0kDBm9khoQSUhHxW98niTnYu
0IPEcFEw7gZmtSdwfsavhpmZ2dNhBVAMqvYj7PVLCDsC5ZGBma4ono8BVVR5MeMZKaht2E
NWANytVEq8hOVeyKF0N9wQuCB6Q/Y34uZt3d95ClP4mqpQnJOdKFqWr4WPCDqCa+DLsAHZ
U3qnBuRJI+tbMOoQdh6s6wYSN51G0t2EURz2vcl5Dcq5kfjUcPcOAr4Gp+LQTVcprSC7S/
GrYFefdusPjNSQXdX417Ip7RdIlwPsUHQH+l/BCigavIblI86thV5yky4B7iD8kvOkH0qI
8DUdV8YTkIk3S1cAa/GrYBSTNBrqa2bKwY4kab7JzUbcReIj4DKGFV8OAXw1H1yxgXdhBR
JEnJBd11wJ7+dWwS7AWmCRpNEVrzd7RpZJ5QnJR51fDrrghweKqmN9DcpEm6X/A/oBfDTs
XMq8huajzq2FXRDCW3xZX6t7LrvJ5QnKRFqGpJVzZdU54nUl81tzdQoolUrzJzkWaXw27s
pA0wcw6hR3Hjs5rSC7q/GrYFSEp8UHpFOJ/I/5bWQW8huRcMX41HG1Bd+9NP4z5wFzgYTP
7IbSgIsKzvos0vxp2W3EMcDJFB9w9A7gzrICiwv/huah7hC2vhk8NLRqXDIYQH8duIrAh5
FgixZvsXKRJymTLq2EzM78ajihJ35nZAWHHEUUpYQfgXMiGACcAecQHWV1DfOgYF11jJbU
NO4go8hqSizS/GnbFSZoO7AXMIT56h4jXmtuFGlgE+D0kF3VjJbU1s6lhB+KSxjFhBxBVX
kNykeZXw84lD09ILtIktdjadjP7uapjcS7qPCE555xLCt7LzjnnXFLwhOSccy4peEJyzjm
XFDwhOeecSwqekJxzziWF/wdQuqwsShxSnQAAAABJRU5ErkJggg==\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "from nltk.stem.porter import PorterStemmer\n",
        "ps = PorterStemmer()\n",
        "print(ps.stem('played'))\n",
        "print(ps.stem('playing'))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "BfwauAmUtozp",
        "outputId": "74a09438-701c-4521-db9e-437e6251c22d"
      },
      "execution_count": 28,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "play\n",
            "play\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "import string\n",
        "string.punctuation"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 35
        },
        "id": "MSsHUDJeudwo",
        "outputId": "9c070bf7-ba08-4d2c-8ecf-bf5251ef9aa7"
      },
      "execution_count": 29,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "'!\"#$%&\\'()*+,-./:;<=>?@[\\\\]^_`{|}~'"
            ],
            "application/vnd.google.colaboratory.intrinsic+json": {
              "type": "string"
            }
          },
          "metadata": {},
          "execution_count": 29
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "nltk.download('stopwords')\n",
        "from nltk.corpus import stopwords\n",
        "stopwords.words('english')"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "QSkExn8puitC",
        "outputId": "dded287d-c593-4b2d-b571-898bde79b021"
      },
      "execution_count": 30,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stderr",
          "text": [
            "[nltk_data] Downloading package stopwords to
/root/nltk_data...\n",
            "[nltk_data]   Unzipping corpora/stopwords.zip.\n"
          ]
        },
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "['i',\n",
              " 'me',\n",
              " 'my',\n",
              " 'myself',\n",
              " 'we',\n",
              " 'our',\n",
              " 'ours',\n",
              " 'ourselves',\n",
              " 'you',\n",
              " \"you're\",\n",
              " \"you've\",\n",
              " \"you'll\",\n",
              " \"you'd\",\n",
              " 'your',\n",
              " 'yours',\n",
              " 'yourself',\n",
              " 'yourselves',\n",
              " 'he',\n",
              " 'him',\n",
              " 'his',\n",
              " 'himself',\n",
              " 'she',\n",
              " \"she's\",\n",
              " 'her',\n",
              " 'hers',\n",
              " 'herself',\n",
              " 'it',\n",
              " \"it's\",\n",
              " 'its',\n",
              " 'itself',\n",
              " 'they',\n",
              " 'them',\n",
              " 'their',\n",
              " 'theirs',\n",
              " 'themselves',\n",
              " 'what',\n",
              " 'which',\n",
              " 'who',\n",
              " 'whom',\n",
              " 'this',\n",
              " 'that',\n",
              " \"that'll\",\n",
" 'these',\n",
" 'those',\n",
" 'am',\n",
" 'is',\n",
" 'are',\n",
" 'was',\n",
" 'were',\n",
" 'be',\n",
" 'been',\n",
" 'being',\n",
" 'have',\n",
" 'has',\n",
" 'had',\n",
" 'having',\n",
" 'do',\n",
" 'does',\n",
" 'did',\n",
" 'doing',\n",
" 'a',\n",
" 'an',\n",
" 'the',\n",
" 'and',\n",
" 'but',\n",
" 'if',\n",
" 'or',\n",
" 'because',\n",
" 'as',\n",
" 'until',\n",
" 'while',\n",
" 'of',\n",
" 'at',\n",
" 'by',\n",
" 'for',\n",
" 'with',\n",
" 'about',\n",
" 'against',\n",
" 'between',\n",
" 'into',\n",
" 'through',\n",
" 'during',\n",
" 'before',\n",
" 'after',\n",
" 'above',\n",
" 'below',\n",
" 'to',\n",
" 'from',\n",
" 'up',\n",
" 'down',\n",
" 'in',\n",
" 'out',\n",
" 'on',\n",
" 'off',\n",
" 'over',\n",
" 'under',\n",
" 'again',\n",
" 'further',\n",
" 'then',\n",
" 'once',\n",
" 'here',\n",
" 'there',\n",
" 'when',\n",
" 'where',\n",
" 'why',\n",
" 'how',\n",
" 'all',\n",
" 'any',\n",
" 'both',\n",
" 'each',\n",
" 'few',\n",
" 'more',\n",
" 'most',\n",
" 'other',\n",
" 'some',\n",
" 'such',\n",
" 'no',\n",
" 'nor',\n",
" 'not',\n",
" 'only',\n",
" 'own',\n",
" 'same',\n",
" 'so',\n",
" 'than',\n",
" 'too',\n",
" 'very',\n",
" 's',\n",
" 't',\n",
" 'can',\n",
" 'will',\n",
" 'just',\n",
" 'don',\n",
" \"don't\",\n",
" 'should',\n",
" \"should've\",\n",
" 'now',\n",
" 'd',\n",
" 'll',\n",
" 'm',\n",
" 'o',\n",
" 're',\n",
" 've',\n",
" 'y',\n",
" 'ain',\n",
" 'aren',\n",
" \"aren't\",\n",
          " 'couldn',\n",
          " \"couldn't\",\n",
          " 'didn',\n",
          " \"didn't\",\n",
          " 'doesn',\n",
          " \"doesn't\",\n",
          " 'hadn',\n",
          " \"hadn't\",\n",
          " 'hasn',\n",
          " \"hasn't\",\n",
          " 'haven',\n",
          " \"haven't\",\n",
          " 'isn',\n",
          " \"isn't\",\n",
          " 'ma',\n",
          " 'mightn',\n",
          " \"mightn't\",\n",
          " 'mustn',\n",
          " \"mustn't\",\n",
          " 'needn',\n",
          " \"needn't\",\n",
          " 'shan',\n",
          " \"shan't\",\n",
          " 'shouldn',\n",
          " \"shouldn't\",\n",
          " 'wasn',\n",
          " \"wasn't\",\n",
          " 'weren',\n",
          " \"weren't\",\n",
          " 'won',\n",
          " \"won't\",\n",
          " 'wouldn',\n",
          " \"wouldn't\"]"
        ]
      },
      "metadata": {},
      "execution_count": 30
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "def transform_text(text):\n",
    "    text = text.lower()\n",
    "    text = nltk.word_tokenize(text)\n",
    "    \n",
    "    y = []\n",
    "    for i in text:\n",
    "        if i.isalnum():\n",
    "            y.append(i)\n",
    "    \n",
        "    text = y[:]\n",
        "    y.clear()\n",
        "    \n",
        "    for i in text:\n",
        "        if i not in stopwords.words('english') and i not in
string.punctuation:\n",
        "            y.append(i)\n",
        "            \n",
        "    text = y[:]\n",
        "    y.clear()\n",
        "    \n",
        "    for i in text:\n",
        "        y.append(ps.stem(i))\n",
        "    \n",
        "            \n",
        "    return \" \".join(y)"
      ],
      "metadata": {
        "id": "mpYezhNwpdAs"
      },
      "execution_count": 31,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "df['text'][23]"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 35
        },
        "id": "8z-2c79Vtc5c",
        "outputId": "e15c21c7-acd8-46c9-a1eb-a79460880757"
      },
      "execution_count": 32,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "'Aft i finish my lunch then i go str down lor. Ard 3
smth lor. U finish ur lunch already?'"
            ],
            "application/vnd.google.colaboratory.intrinsic+json": {
              "type": "string"
            }
          },
          "metadata": {},
          "execution_count": 32
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "transform_text(df['text'][23])"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 35
        },
        "id": "YgoiwWFMt0M0",
        "outputId": "04c27a46-977a-450e-c811-aaf9ef55a934"
      },
      "execution_count": 33,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "'aft finish lunch go str lor ard 3 smth lor u finish ur lunch alreadi'"
            ],
            "application/vnd.google.colaboratory.intrinsic+json": {
              "type": "string"
            }
          },
          "metadata": {},
          "execution_count": 33
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df['transformed_text'] = df['text'].apply(transform_text)\n",
        "df.head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 320
        },
        "id": "seUesFnmtkME",
        "outputId": "73fee866-975f-40ab-d229-b9e398087927"
      },
      "execution_count": 34,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   target
text  num_characters  \\\n",
              "0       0  Go until jurong point, crazy.. Available
only ...             111   \n",
              "1       0                      Ok lar... Joking wif u
oni...              29   \n",
              "2       1  Free entry in 2 a wkly comp to win FA Cup
fina...             155   \n",
              "3       0  U dun say so early hor... U c already then
say...              49   \n",
              "4       0  Nah I don't think he goes to usf, he lives
aro...              61   \n",
              "\n",
              "   num_words  num_sentences
transformed_text  \n",
              "0         24              2  go jurong point crazi
avail bugi n great world...  \n",
              "1          8              2
ok lar joke wif u oni  \n",
              "2         37              2  free entri 2 wkli comp win
fa cup final tkt 21...  \n",
              "3         13              1                u dun say
earli hor u c alreadi say  \n",
              "4         15              1               nah think goe
usf live around though  "
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-9e646c32-3929-4359-a6ec-
e64bdcffe7a5\">\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>target</th>\n",
              "      <th>text</th>\n",
              "      <th>num_characters</th>\n",
              "      <th>num_words</th>\n",
              "      <th>num_sentences</th>\n",
              "      <th>transformed_text</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>0</td>\n",
              "      <td>Go until jurong point, crazy.. Available only
...</td>\n",
              "      <td>111</td>\n",
              "      <td>24</td>\n",
              "      <td>2</td>\n",
              "      <td>go jurong point crazi avail bugi n great
world...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>0</td>\n",
              "      <td>Ok lar... Joking wif u oni...</td>\n",
              "      <td>29</td>\n",
              "      <td>8</td>\n",
              "      <td>2</td>\n",
              "      <td>ok lar joke wif u oni</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>1</td>\n",
              "      <td>Free entry in 2 a wkly comp to win FA Cup
fina...</td>\n",
              "      <td>155</td>\n",
              "      <td>37</td>\n",
              "      <td>2</td>\n",
              "      <td>free entri 2 wkli comp win fa cup final tkt
21...</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>0</td>\n",
              "      <td>U dun say so early hor... U c already then
say...</td>\n",
              "      <td>49</td>\n",
              "      <td>13</td>\n",
              "      <td>1</td>\n",
              "      <td>u dun say earli hor u c alreadi say</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4</th>\n",
              "      <td>0</td>\n",
              "      <td>Nah I don't think he goes to usf, he lives
aro...</td>\n",
              "      <td>61</td>\n",
              "      <td>15</td>\n",
              "      <td>1</td>\n",
              "      <td>nah think goe usf live around though</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-9e646c32-3929-4359-a6ec-
e64bdcffe7a5')\"\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-9e646c32-3929-
4359-a6ec-e64bdcffe7a5 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-
9e646c32-3929-4359-a6ec-e64bdcffe7a5');\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": 34
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "from wordcloud import WordCloud\n",
        "wc =
WordCloud(width=500,height=500,min_font_size=10,background_color='whit
e')"
      ],
      "metadata": {
        "id": "Y3OblhJFvmoY"
      },
      "execution_count": 35,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "spam_corpus = []\n",
        "for msg in df[df['target'] ==
1]['transformed_text'].tolist():\n",
        "    for word in msg.split():\n",
        "        spam_corpus.append(word)\n",
        "len(spam_corpus)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "GYENps4iv5aB",
        "outputId": "e01c9aaf-c8c6-44eb-ddf5-39eed0718f11"
      },
      "execution_count": 37,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "9939"
            ]
          },
          "metadata": {},
          "execution_count": 37
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "from collections import Counter\n",
        "from collections import Counter\n",

"sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.
DataFrame(Counter(spam_corpus).most_common(30))[1])\n",
        "plt.xticks(rotation='vertical')\n",
        "plt.show()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 362
        },
        "id": "EANqfrRJwLtY",
        "outputId": "8e826823-fa63-48f3-ee4d-4dad75a5ba32"
      },
      "execution_count": 38,
      "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. 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 432x288 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYUAAAEiCAYAAAAWOs4eAAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
```

uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3dedxtc93/8
dfbMc/kGHJwJHFLMpzMFamo6EQypJKU6j7CLyq6K5JKbuWOupVCR5GMN0mGZMx4TMdcJxE
nw6mEEoXP74/vd69rXftaaw/Xde1rONf7+Xjsx9577c9e+7un9VnrOy1FBGZmZgALjHYBz
Mxs7HBSMDOzgpOCmZkVnBTMzKzgpGBmZgUnBTMzKyw42gUYihVWWCGmTp062sUwMxtXbr3
11j9HxOSqx8Z1Upg6dSqzZs0a7WKYmY0rkh6ue8zVR2ZmVnBSMDOzgpOCmZkVnBSMDOzgpOCmZkVnBTMzKzgp
GBmZgUnBTMzKzgpmJlZwUnBzMwK43rwGsC8E3/SNmbyJz8wAiUxfKRgZmYFJwUzMys
4KZiZWcFJwczMCk4KZmZWcFIwM7OCk4KZmRWcFMzMrNCzpCBpUUk3S7pGj2SvpyXrynpJ
klzJP1M0sJ5+SL5/pz8+NRelc3MzKr18kjhBeAtEfF6YENgB0mbA98AjouI+0SL5/pz8+H2
Bp/Ly43KcmZmNoJ4lhUj+nu8ulC8BvAU4Jy+fCbwn356e75Mf306SelU+MzMbqKdtCpImS
boDeBK4HPg98LeIeIeDGHPAqsmm+vCjwCkB9/GnhFL8tnZmb99TQpRMRLEbEXA0gP74M8JeKdZ0UE
dMiYtrkyZN7XnYzs4mkl72PJJi5hJkbZNt9eDHgbcB8pOeyaw/YGLsi3L8z3yY//+Yb3nbGdTUFyyN
11j9HxOSqx8Z1Upg6dSqzZs0a7WKYmY0rkh6ue8zVR2ZmVnBSMDOzgpOCmZkVnBTMzKzgp
GBmZgUnBTMzKzgpmJlZwUnBzMwK43rwGsC8E3/SNmbyJz8wAiUxMxv/fKRgZmYFJwUzMys
4KZiZWcFJwczMCk4KZmZWcFIwM7OCk4KZmRWcFMzMrNCzpCBpUUk3S7pGj2SvpyXrynpJ
klzJP1M0sJ5+SL5/pz8+NRelc3MzKr18kjhBeAtEfF6YENgB0mbA98AjouIVwNPAfvm+H2
Bp/Ly43KcmZmNoJ4lhUj+nu8ulC8BvAU4Jy+fCbwn356e75Mf306SelU+MzMbqKdtCpImS
boDeBK4HPg98LeIeDGHPAqsmm+vCjwCkB9/GnhFL8tnZmb99TQpRMRLEbEhMAXYFh3qOu
UtJ+kWZJmzZs3b8hlNDOzPiPS+ygi/gYnzqz+jZT0rrLiwCRn/NNW4CLZcxOd6A0gZ7mZU0
dMiYtrkyZN7XnYzs4mkl72PJktaNt9eDHgbcB8pOeyaw/YGLsi3L8z3yY//+Yb3nbGdTUfc
Benk+hVWAmZImkZLPWRFxkaR7gTMlHQXcDpyc408GfizpfuAvYI9xDpyc408GfixpDvBXYI8els3
4rlD5LaF5qXPw+8r1flMTOz9jyi2czMCk4KZmZWcFIwM7OCk4KZmRWcFMzMrOCkYGZmBSc
FMzMrOCmYmVnBScHMzApOCmZmVnBSMDOzgpOCmZkVnBTMzKzgpGBmZgUnBTMzKzgpmJlZw
UnBzMwKTgpmZlZwUjAzs4KTgpmZFZwUzMys4KRgZmYFJwUzMys4KZiZWaFnSUHSapKulHS
vpHskHZiXHyZiXHyFprqQ78uWdpeccJmmOpAckHfxWETclm8/C9wHrNriKdOBMyPihYj
4AzAH2HzeuQe+rspmZWbUFe7PnJ/KiC6ZnKmKbBCiOEC7ERgPEBd5EpEtEqV4ZLYU
EQcWw6WtB6B0lRHjlyP+e+E3TLRgwKgKxnRl8v2G8pKPhvDWGL5tWatnu77HvgTGZ
4AzAH2LRX5TMzs4FGpE1lB0lRgI+CmvOgASbMlnSJpubxsVeDR0tMeozqJzJA0S9KsefPm9
eBcS4zA4QCKz4lSiCyfVIUrc5b9kWrMyE88R0RWLMwzz5xbDgAjQsZhEUnG12U1bvW0DoMfsV
Zcuv4YuwW5HCzN8fq+aNZb8zI7vi4jqIPc+qNyHdoctq9ZcL58V9wHeMPZiaG5rKY6Hn7MaTsr
LwTmAM8B+zTw7Sm4qzSEGH+Q0xSdRj8XdGScqhaNoMO9eGtyH1MfsgRrnJT9pLpbYM7JiZLctzR4L6TsXvgsbYuTm7
9BqATFh8vbPN7YwGxwPSwJOoWAjkNr6KTdIs7lPyJsw0G7PXSrSG3Zj1JvbmZVwE3eKZeLhbT3
CmMBUULLN2pyFnTdGjYVWOJd4SZPN1iglNzHm7PdzNrPMtdUzNsWxK7uqMUQKoLXvFRWblA5
bA1Y9OeeVg1vSVaRnZ6hBkmnu9wWMH5AOuyLwXtbfbLS1EfmzjKjezOhR3SHkh5Ql3tMg/NEJmwN3
8xX2D5iU6YMaL7xRfN4l4jg1dA1yLCKgW6vRH6tqmaSVGbIxq7zfqMcwQ/pLDBAV2cu9ZfYnqJMSN0
ZJ7x1ZBgO7kFvvMZFnDYxVuMPE4mINIgIf1vcqTNn5v+5ThhmhUlThH1mWAhn5RnfcoNj7lknrNvcC
ZpVbr1d8G+Iw6FdiHLZ4U7hEnnrx1oGOa+B2sRB5HvcmYTVRo7d94PmaSm7WTlnrjrL3wxWaM5
wxsMxzhyxW0Xm6ZgCCqJZ7PH4MjNAPIMMjLR3J4g1dJxGnUq7cDG+FJAdrZW8GgAVpv6RE6BMe
cXU8BRyAfCVHRHnAKIOeAFEYSa8ZU4EUTKo1+B7hxgJCSPmSLNbUJ+0Fr5UuXhnfD6NKZ2yEj
7VOJOmZgO7Hs3AOt0aBBLMQb3ysBMyxuOcj2LjMBPe/JyR0A0tB7wMhLGVRHzzVVDUR4Ep
VAuvd2M9TpEdtu6Hm3Oju6TYtZEBGQ+FFWx1rF8bWR4nvJD/c3ix9kyKgpRnsyOHd+N1L9Ob0eA2sR
dLodPS6kAZfaMHEsdJ4SdIMg6HOLaSsJAnFr+7JmmDeasQK+YjDdaM0nvt7R1FTQ9ORJZbh+j9W5DVd7
CM92xDgnQvTtfxEgsnV+u1rnQyKqBfeovKQ1C2Gy5vOwe4I61CqsQwlWHmY9MIBY9Ol9LLTh0p
wYuyDHcVWSgcSNmr1t9wy5fi6ClOQy5t2IWp0rnL8AQXZIN9HDvp3YI6T9a3nNIFHOV43wjDr5J5
x/YwE6jVIeMqWmxJyOZM5B4p27DtGvBAkqDqmGpeIHKvPlb0YqEgmDaIW1Y9qSwJPwDUrKQdBTHyk1
Ug6iy9HdDApeT4k06OIYaB5tamQqVrBhcoFP4eFOvaZvXSb4iO93FAsZbZ7eRvWYp98DbfMAi4
gdXH6FsQ/EN/nvrO9b5UXNhUGGpIlx+1A4wMLwO2DNtfjyDVzu3aCKwqFQw6DYSZ9SdRuqmc
4HFmeoEgxvb2rk6kTwCTq6XSMrnDrbw4s9gC7ueEWoCzhUhXSmSE7Sc6Mke3BwRhGU9Ldf
mzr57+yW9Zuz1Dx7eGhwt2HSMbcPBxH41g2SYEHzxVgGQ8RIxqb5mpW8ezYKu+uSRHRuz
z9SCuYIiSHR+D8YNRLTSbWhfaTadkBG+UbL8W8dhyPo9GZuVkvs2pfBHP9uz8jbjmJHwZzmJq
+Pf9lW1xh7tvnxGNm7bfjSz3TgNsYg

spzyi2czMCk4KZmZWcFIwM7OCk4KZmRWcFMzMrOCkYGZmBZ+j2eZL7zz/qLYxF+/8hREoi
dn44iMFMzMrOCmYmVnBScHMzApOCmZmVnBSMDOzgnsf2YT3rvOObxvzi10OGIGSmI0+Hym
YmVnBScHMzApOCmZmVnBSMDOzgpOCmZkV3PvIrEvvOveHLR//xXs/Wtze8ZzT267vol33G
nKZzIaLk4LZGLHTOee1jfn5rruMQElsInP1kZmZFXykYDYOTT/nkrYxF+y6wwiUxOY3PlI
wM7PCqBwpSHoIeBZ4CXgxIqZJWh74GTAVeAjYLSKeGo3ymc1Pdj73urYx5793awDed+7st
rFnv3eD4vYB5z/SMvb4nVdruz4bW0az+mjbiPhz6f6hwBURcbSkQ/P9z41O0cxsuH33/Cf
axszYeaURKIm1Mpaqj6YDM/PtmcB7RrEsZmYT0mgdKQRwmaQAvh8RJwErRcRj+fHHAe8ym
E1Q553z57Yxu+y6wgiUZOIZraSwdUTMlbQicLmk+8sPRkTkhDGApP2A/QBWX3313pfUzMa
0K0+f1zZm270mA3D7D59sG7vRR1cccpnGs1FJChExN18/Kel8YFPgCUmrRMRjklYBKr+9f
FRxEsC0adMqE4eZ2VA99D+Pt42ZetDKADx+7Jy2sSsf8uohl2kkjHibgqQlJC3VuA28Hbg
buBDYO4ftDVww0mUzM5voRuNIYSXgfEmN1z8jIi6RdAtwlqR9gYeB3UahbGZmPffE/9zaN
malgzZJscdf1T72gG0AePK7P28bu+KMnVo+PuJJISIeBF5fsfwvvwHYjXR4zM+szlrqkmpn
ZKHNSMDOzgpOCmZkVnBTMzKzgpGBmZgUnBTMzKzgpmJlZwUnBzMwKTgpmZlZwUjAzs4KTg
pmZFZwUzMys4KRgZmYFJwUzMys4KZiZWcFJwczMCk4KZmZWcFIwM7OCk4KZmRWcFMzMrOC
kYGZmBScFMzMrOCmYmVnBScHMzApjLilI2kHSA5LmSDp0tMtjZjaRjKmkIGkS8F3gHcB6w
J6S1hvdUpmZTRxjKikAmwJzIuLBiPgXcCYwfZTLZGY2YYy1pLAa8EjL/Ny8zMzM+mCs9T2k
XoSBpV2CHiPhovv9BYLOI2L8Usx+wX767Dvb7AxapWAP7c4cuOt9ixUo6xEDtWyjEWYsdKO
cZb7Fgpx0jHrhERkyufERFj5gJsAVxaun8YcNgg1jNrfo0dK+UYC7FjpRxjIXaslGO8xY6
VcoyF2MZlrFUf3QKsLWlNSQsDewAXjnKZzMwmjAVHuwBLPP2BS4FJwP2BlEfGipP2P2BS4FJwCkRcc8oF8vMb
MIYU0kBICIuBi4e4mpOmOo9xkLsWCnHWIgdK+UYC7FjpRxjIRYYYw3NZmY2usam4KZam4K
ZmY0iJwUzMytMyKQgaTFJ63QQ66oSVrQ8c8tb/7JQ3FSvcXkLR496UceyQt0smysSp/F0uP0mvPlLRs6f5ykk6piT2
wk2XDUq7x3KYg6dOtHo+Ib1U8ZyfgWGDhiFhT0obAkRGxc4s4SRH9qAa4A4AZwH3AhsCBEXGBfu9Vor5OVD75Td/Hvk5rwCOALbKz72O9Nn
9pRSzdEQ8I2n5mvX+taY8HxkZp13xwRm1Y9Vor5OVD75Td/Hvk5rwCOALbKz72O9Nn
9pRSzdEQ8I2n5mvX+taY8ywGrUer8EBG3lR5v7posYFvg13Xlzc97DXAisFJErC9pA+DdE
XFUU1zV93FrRGxSs96rgSmkbtTXAtdExF01sTcCb42bR78h
zf20RESeXYip/Uw3lz61p3VdExHbtlpUeWw5Ym/Sba6z7mqaYTv8Ld9H697ZBxesL2At4V
UQcKWl1YOWIuLmmvFsCU+n/Gxqw4yDpDOATwEuk729p4NsR8d8VsUdGxJdK9ycBp0XXEXXhW
xiwDvrSjDkTXlvT0iNmq3LC+v+kwrY4dqzPU+6tJSg3jOEaQ5lq4CiIg7avYIPwZsEhF/l
zQVOEfS1Ij4NmvhNmmDBICkdwDvBFaVdHzp+0nnAXXeULFzavtNv4XMZvNq3jWFK33Tq/kfQd4GfAP0r
rKW8sjm3x/DpnAteQfviQ/rA/A95aijmDlLRuJW0EVHosgFc1r1TSV4APA7+nb8MRwFv+SwKXRIwt ...

```
2SpviXkUawLMlKeH+AfhARDzUFNdR20MpfllS28qbSHvULwM3RMQXa+IXIh2VADwQEf+ui
dsYOAFYn/R5TAZ2jYgBe4WSfkr6/H+SF+0FLBkRe1atOz+n03r384F9SAntLaTPbqGIeGd
T3NeAYyLib/n+csDBEfGFFu/vVGCZvOhvwEeqNrJ5Q7g7sDEwk7QT8oWIOLsU02gbW4p0h
HwzHf6Gml5rwfJ/apC/43uBV5N+Yy/QegerKzl5bNnY+chHTL+JiKqqtG+TEuT/0f+zOK+
j1xrPSWEoJC0eEc8N4/pWIW18ngdWIjVQH1yuSx5K/BDLtghposFtah5fmdTOEsAtVYmwl
yTdQ9oTu4vSnuAgq7jqXmMJUpXgszWPX01ue2hUb0i6OyLWb7HO/wDeTKpC2hL4Y0S8ufT
4WyLi1+rfVlGo+5NKWpCUQETrBLIo8ElSYoLU3nNi8w5OKb6y3r3miLr8vDeTNuKXNB8Rq
bqxtK7DxCTggIg4TtIy+cXrjoIaz1kX2I70WVzRvAecy1ar7jeUdypeS/9G9FZH3211s4M
1iHXfEREbNi27MyJeXxF7anUxqjumNBvX1UcN+bDyc6SztZW/5KpsviWpsXJJYHVJrwc+H
hH/OZQy5L3uS0gzu74MHNpqA99t/BDVtkEoNSh+idSLR8AJSj0u6rrGvZu+jdBVEXFRTdy
iwH8CW5OSzbXA92o2WM9FRMuqwMHK9fH/TZptt1ENUbXRatf20LzeB4H7ST2wTgT2qahCe
jPpc92JgQKo23PblL69+Y0lVe7N58/yuHzpRNt6d1X3HmtUhyzJwMbNSbl66YX8/MWAyi6
pEfGSUsPpce2SQcnvSFU7C+b1rx4Rfyyt8+q8fAngnxHxcj7qW5fUiFz1Hr9H+k9sS9oW7
Eo6wqiK7aQ32NIR8Qypnr9X5kl6d0RcmF9zOjXTZ0fEPkN5ofkiKQCnk3povIvU1WxvYF5
N7HGkRrILIdXxS3pTTWzHJF1Oqs9cn9S18mRJV0fEZ2rif0VqdC7HXxMRhwxDWbppg/gMs
FHkbqW5Pvt6UttJ83qPJlWVnJ4XHZir0g6rWO9ppD9Jow77/aQeJ1VjMa6V9HXSd1I+3K3
sWtmle0jVXJdJ2j1Sl1hVxLVse6jw6kjtHLUi4vB886MR8VKr2Ia6vXkqetFI2orUsLwG/
auDBvTwyjqpd6/qPVa+bl736cAVpb3TfUhVPXU66RkHgKRPAYcDT5A+i0YZqqpjrgHemKu
vLiN1Nd2d6t5NW0bEBpJmR8SXJX2TmgRCZ73Buu5xNwifIDV4fzev81HgQ+UASZ+NiGMkn
UBF55l2R4TlwHF/AW7N17NLy26pib0pX99eWnbnMJTht033FwReahH/nqb7k0htDMPxeax
RuqwKLNgi9nrSmI3G/YWB62tiZ5OqYMplnl0Te28ny/LyKysuvx6mz+K2fL07KUFs0ljWF
Pcq4FfAc8Bc0hHAGi3W+xrgCuDufH8DUn13VewfSVWF25GrbFus9752MaXY+0nnM18ReEX
j0iL+SlLbwKWkBHwhcGEXn2VluXIZjs2X7duso+PvGpjT6v3UfM+fAj6bb1f+r0vbgBtJP
bIWIZ0GuCr2lnxd3l7cURP7E1KPtXWH47db8xpLktqNqh7bKV/vTUoY/S6dvsb8cqTQqHN
9LNcV/gmoHDwFPJKrkCI3/h1I+iMOVb+qn0hdCe+tC46I/1Pq1rh2RJwKLEdfg+GQRAd1m
Oob+DcHuEnSBaS9i+nUd3WD1EuoUYWwTIu42yRtHhE35tfbjNQVt6q827Yr7xAov8bPctv
FGcDqFXHvIc3OeyV9DehvVRrAdkdFfMvxD03WJe1JziAdEV4EnBkR11XEdtOL5umIqNvDr
XJEp4EaOGhrAdKR3oA971yGjsrR5Xf9CPU9r5pJ0hak8u2bl9X1aroodxT4b+A20u/+hzW
x3fQGO5nUvnRCPuq8Dbg2Uu+vIemkGisifp5v3kvqFjyVviPIujEbA18rZ5ZxTamr6LWka
pgTSANojih9SOXYFUhdzN5K2mBcRhqt/Jfm2A5f+5OkuvNXkQZfNSxF6h3wgZrnHU6q410
nIl4j6ZXA2RGx1WDK0a38+rWiYgxIrg8+mrThFKlt4bCIOLMi9j5SY2mj/nd10qlTXyT3y
JD0gYj4iWpGpkfFiPRuSdokIm4t3V8GmB5NdfR5gz6NtPcs0kZ8NumPdXZEHNMUf0tEvKH
c0FrVGFhRnuVIv7+9ImJSxeNX0mEvmlydN4nUNtFRtVtuDF07In6l1BNvUlQ0vufqoN9G0
6CtiDiiKW5z0n/uP0hHmZOAf0SLgVKdNvJKOpn0G/pF0/urmqngTcAhpP/cN5R6nR0U7Rv
RFyENLqzc0KuL3mA5fhKpinVbUpXPPyNi3VZl6ISkX5KrsSLi9bkzwu0R8bqK2AdIOyzNH
Tc6avCeX44U3gdcFxF3A9vmRrJjgX5JIX9h346KIepDcAZpL+nr9B+A82zUTOmQ7QxsRNq
bICL+JGkwI7QHpXmjrzy/U7RuHP+ppKvoG1H6uajvqdTJIJ4l8vWwv2/lnj/AGhW9Qqre4
xTSQLDGVBSHkzZGbyLVFR/TFN9VG0TuJbM76XOZRRqwV+WIFm+r2Wb5elppWb9Bik1l+Bj
p/ObLk9otVgW+R6rWatbpoK3vkM6QeHYux4dIVWuVumnkJe1Q/JGUbOoGrQHF9BvXlO4/S
BqDVFWGxYGDgdUj4mOSVpf0xqjoNBERt+XvrpPeYFeQftM3kHZS3xARtWNvutR2UGPJvMg
N0oMxvySFDSL3kwaIiL8qDb3vJ1LvhzUkLRytBxt1LO9hPA3U9g2v8a+ICOWRlLn3xIhTG
gX7Y3J1m6Q/k+ofB5zxTn3z5FxYsayfiHi4XD2Wj9CWiog/lGIaVS+DGZneTrc9f1ak/9Q
E/yaNWfinpKopC2aQ2gnWlTSX1De9cmdD0kOkqRHOIo3G/kdVHKTeNLmqoJF4b67bsAyi2
m0GqWfTTfn5v5O0YlNZux60FRFzJE2K1Jh+qtII5KrOB9BFI2/jd9HJDotSD8TPMvAIpCp
BnkpK9I3pQOaSktqApJATyKdJ7Usfk7S2pHWqEgjpyHIT0lHF08DfJN0QEXWj17vRTTXW4
ZJ+SGrz6nqcwvySFNrOIVLyIKkHxIX07/0w5KqKLp0l6fvAsnkP7iOkeuqRdhLw6Yi4EkB
pOo8fkPrdk5cNZu6conqM9CdcmNRmMqB6LK9/Xwb+oTvqV10lIg7P9eC/jDxAsI3T6Wtbg
ZRMzsjJuqptaC7pfV1JSqjPkBr4+lWD5KPTU6qqR6pI2o1U130VfV2EPxMR51TEtq1nbvJ
CRPxLudttroJorj9unkvqKVJX729SfRTynNJAqjskHUM6Wmo1+3JjA/lcrjL9C7BKVWA3O
yz09UDckfY9ENeKiN1zdSgR8Zykqh5p0EUCiYj/l8u5FGkKkFNJ7UPDMWvsp0k7Y2tJ+g2
5Gqsmdh9SO9ZC9FUfteoC3c/8khTaziEi6ccR8UHSnEPHkX64I1ZdU2EycA5pY7IOaazAW
1s+ozeWaCQEgIi4quKopWrunKB/l9Nm3VSP/ZjUk2Z70kZ1L4ah8T9Sn/XP0jctR6vYr+R
```

```
620bS+kRENBrGq44ALiCNxr2N1LGhbr0v5TavTgdG/Relaoe8B/wr0m+l2QSURhgAAAlrS
URBVI9o312y7GpJnydNYvc2UltYvyrWQRx9fJD0X9qfNHfXakDlYL2sqpG3bmeo7Q5LySs
i4mRJB0Yau3C1pFtq1vsvpfEUjb3utaifwK7jBKJ0fvk3ko4WHiJ1666bD6sr3VRjkX4/b
U8N0OrF5osLaW9m/3xZr+Lxe0kbtdmkPY9+l1Eob1W3yMrunT0ux/nAF0kNqlOBLwDn18R
+CVg63/5ifu7GNbE3l98nqa61rvvq7eX3T9rDuXGY3t/RpAbI1Ybz+yZ3Re0w9jhS3fsbS
VM2bNzic7ur6f4CzctKj3XcXbK0ro+R9nTPybfrupkuA3yL1P4xi7TjtUxF3IGdLKt5jUW
q1ll6fECX0qplefmN+fpS0niljYDf18S+DbiadCRxOmkDvk1N7PXAYqXf8VqN33ZF7CGkd
p7aLuCD+J3t0upS85xTqdgGdnqZX44UiDSRVG0XUFKD2hXAmvTvGlk3KKcnVOqtJKncg2E
pUt3tSPsIaTbVxqRm15IOP6vsGmle+61J1QjHkkbzblYOyntSF3VRPdbY4/lbrjJ4nFTHP
xx2J32/zSPWh/p9dzP5WaNHUvlooa5B+JeSLgV+mu/vTuoqW6WbemYiDbb7AZ1VU55C6m3
TaBD/IGlj03wUsDep7aHswxXLyGW8jrRBvpbUU6hVl9MHJX2Rvmm2P0Cq/q1ylFLPsoPp6
4F4UFVgRFyuNDPr5qT//4ERUTk6mDR47hLSFNqnk44kP1yz3sFMR99OVZtY8ZJUVwltTqr
O+wODmINpvuiS2g1JJ0bEJ0fx9ZchjUnotrdSr8ozjVT9MJVSn+aqH5By90ul0cd3RcQZq
j8pyF2ketC3k36Ul0bE5TVl+CgpKW1A2vAsCXwpIr43DO9vMaqn2xhS4596NPmZpG+QGoK
3zouuBTaPiM9VxHbbXbLqRDdPk3aSjor+J0uqmmunWKa+ef63pn8VyVLAy1F/4p416Ztyf
HPSZ3dt5Pr4ptjlSDss5c/iiMhth02xM0kb98bEfMsDx0b/SSC7nsFX0k9ItQv/JCWkm1o
kkDFBQ5yDacIlBetPqU/zIaSNSss+zUqDruaSDr83Jv1Rbo7qSblmkuaqr6vXHRGSziK12
zSm5ng/qcqirktop+vt+I/XTYOwqs+wNbsmSb+PVF2yGunER5uRRsXXnXntGNJ0EWfkRXu
QOhA8DmwdETuVYm8g9ZS6Lt/firSR3aL0/tekYueGVA3Yat6oVeibSHBb0kSC3ZyHoGqdb
c9ipv4zn5Y3fI2EXjVX2rb0JbG1SL3IrolhGJDWCQ3i7JJDfk0nhYlN0nURsXX7yKJ73g6
ko4Tf5T/36yLisorY+0l70g/Tv5fXBqWYnv/gJd0bTdORVy3rJXUw8EiDGATZSBa5Ou8rp
Oq8L0XEZs2xOb721JmS7moqz4akOYwao9afAvauOwrplKTfkyZyO4O0539H1MwhpTSx3SE
MnOq7auN9J6ldoNwD8eqoHtxVdfTYanbZngxI64QGMch0qOabNgUbtI77NEeaavy80v3Hq
B+wtX0Hr93ojRQwYJK64dpb6Xi6jR7qZODRYAZBNtbxLuAHEfELSZXnf8gmSdo08jmOJb2
BNAIZBs4Iex9pwN5apKlNniZNBdIvKaj7Uz8eT9oY70lqDL5aaSLI31fEnk1qC/xh6b3W6
eQsZg0zSUePjZl530+aAmLA0aN6OyCtrV5s9NvxkcIEl+tM1yVNFlf0aY4hjBEYRBma64O
XA745HGVQB9NtDPU1OijDVaTqncvzXvnmwDeidO6FQa634+q8HD+NvjYbSFU9+5I6aLwrS
uM5lKZ1b3S5LTbIMfB84XMYxKkflQak7UM6EpgS1VN+3BoRm3SxzvXoa7z/dTSdxawU1/H
Ro6TjSF1MXyAP4iOdTGk4BqR1TOmkXyfQ12X6WtJ/5tH6Zw3ytZwUJjZJD8RQ+jQPTxna1
gcPYd2Vdf8NnTa+DbEMXTUId7HebqrzujrBjdqcYKgU95voYr4upRHMbyTtfV9Pmo322kj
TUjRiGpNZHkDqNto8t9OQOmTkHaHvNB09zoiID7V4TmNA2iHAyhExHAPSOqY0Nf8Z9O+Jt
VdEvG24X8vVR3a9pPXq9qpGSDcj0rsyEhv9DqxFml663CA85PfXTXVedH+Cm0673M6S9DM
6P/XjDaS2j9XpG+k7hf5dTZvPS1A+vSwMvTvxJqT31+/osdE7q6ndq2cD0ro0OdJsyg0/k
lTZ5XaonBRsSH2ah0k39cHj0Rcj4uxcLbYtNeM7RkDHJ7gh1ft/uIPfxdKkc1C8vbSs1ZQ
KjZPgTCGdSGhzUqIoGo8jYk2o707cyRtto5ueTouSBvHd2qpH1Qj4i6QP0Dd+ZU/SFCHDz
tVHE9xQ+zQPYzk6qg8ej9TF+I4el6PRJbPxp2/VFbMnv4u8N/4G0gjkDZXOwfy1iBgwNUa
vuhOPR/n7OIE0B1OQqt4+FRGPDPdr+Uhhghsj1SudjEgfz+Yqje5+G/ANpTn8W00a1ytXV
Syr3Cvs9HcxiAbQ5yPieUkondv5fkl1bVrrNzX+XqkWJ66azx1J6hJcrmI9ljRTwLAajR+
m2USzG2mQ2fa5h9XypJOgjLS/ly4vkqpRpg5xnaeSZu98Zb78PC+r86jShHj/B1yuNCttX
QK6LffUAkatO/FYsUGURnLnxvaeHGm6+shsgspHLJdGxDZDWEfL6TDaPPfNpMFxl0TF+U3
GQnfisaKbwXlD5eojs4lrcVKD71AMugE00hTXrQxp6ov5zIh1xvCRgtkEof4T4k0ijZc4M
iK+M4R1jlgD6EQ3Up0xnBTMJoimHkUvAk8MtZtlHo1+UHMD6EiOiLfh5eojswmiRz3NBjS
AquL86DZ+uPeRmQ3FAnlQHjC8o9FtdPjLM7OhmN9Ho084blMwsyGZn0ejT0ROCmZmVnCbg
pmZFZwUzMys4KRgNswk7SDpAUlzJB3a/hlmY4fbFMyGUT7D2W9JM6I+CtwC7OnGVxsvfKR
gNrw2BeZExIN5krczgemjXCazjjkpmA2vVYHyvD+P5mVm44KTgpmZFZwUzIbXXGC10v0pe
ZnZuOCkYDa8bgHWlrSmpMXZHcdkwiogXJe1POv3mJOCUiLhnlItl1jF3STUzs4JrCmZm
VnBScHMzApOCmZmVnBSMDOzgpOCmZkV/j/scQAYp
NqfngAAAABJRU5ErkJggg==\n"
         },
         "metadata": {
           "needs_background": "light"
         }
```

```
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "ham_corpus = []\n",
        "for msg in df[df['target'] ==
0]['transformed_text'].tolist():\n",
        "    for word in msg.split():\n",
        "        ham_corpus.append(word)\n",
        "len(ham_corpus)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "--6Y_dDywS_4",
        "outputId": "81d713b7-e13d-4ae2-aaf6-9a15a3c5c193"
      },
      "execution_count": 39,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "35394"
            ]
          },
          "metadata": {},
          "execution_count": 39
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "from collections import Counter\n",

"sns.barplot(pd.DataFrame(Counter(ham_corpus).most_common(30))[0],pd.D
ataFrame(Counter(ham_corpus).most_common(30))[1])\n",
        "plt.xticks(rotation='vertical')\n",
        "plt.show()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 352
        },
        "id": "n0VdBo8dwY82",
        "outputId": "9e0fc4d6-6f9f-448b-f7df-18830398d1a6"
      },
      "execution_count": 40,
      "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. 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 432x288 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYUAAAEYCAYAAACz2+rVAAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3deZxcVZ338
c8XAig7SEBJAkFAGAdZI4KoD4goq2EJCCJgRHkcUVSUgXEZUMdRZBMQcZBlAgYVAjxhFRX
ZRSAhyM4YQSTIEhEQEFnG3/PHOXVzu/rW1t3VXd35vl+vfnXdW+eePre66v7uWUsRgZmZG
cASI10AMzPrHQ4KZmZWcFAwM7OCg4KZmRUcFMzMrDBupAswGKuttlpMnjx5pIthZjaqzJ0
7988RMb7quVEdFCZPnsycOXOXNGuhhmZqOKpEcaPefmIzMzKzgomJlZwUHBzMwKDgpmZlZwU
DAzs4KDgpmZFRwUzMys4KBgZmYFBwUzMyuM6hnNAAtP/1HLNOP/5SPzeTMzMs9HPNQUzMys
4KJiZWcFBwczMCg4KZmZWcFAwM7OCg4KZmRUcFMzMrOCgYGZmBQcFMzMrOCiYmVnBQcHMz
AoOCmZmVnBQMDOzgoOCmZkVHBTMzKzQ1aAg6fOS7pV0j6Qfs3qdpHUk3SppvqvqSfSlo6p10
mb8/Pz0/uZtnMzKy/rgUFSROAw4ApEbERsCSwL3AscFJErAc8AxycYbvPymnMzOzY
dTt5qNxwOsljQOWAR4H3gvMzM/PAPbIj6fmbfLzO0haHLk4zhnWCxD3fJ4scMxbQD2cGlvsqj
lFtvHp9fmbfLz20tSl8tnZmYlXQsKEfEYcDwR4AVoFQGI/QUQ1HqsjmZDZhWH9jvMys/OsCaw
HLAjoPNNyLOiIgpETx8g+3j3cXwZ8sRCaW9wYS7NMB9o/NxY8A
kgPz8SsDTXSyfmZnV6WZQ+COI9Wz8zM6
nSzT+FWUofxHcDd+W+dARwJHC5pPqnP4Kx8yFnAG/L+w4GjuIU2MzOrNq51koGLiKOBo+t2
2PwRsWZH278De3SyPmZk15xnNZmZWcFAwM7OCg4KZmRUcFMzMrOCiYmVnBQcHMzAoOCmZmV
nBQcHMzAoOCmZmVnBQMDOzgoOCmZkVHBTMzKzgoGBmZgUHBTMzKzgomJlZwUHBzMwKDgp
mZlZwUDAzs4KDgpmZFRwUzMys4KBgZmYFBwUzMys4KJiZWcFBwczMCg4KZmZWcFAwM7OCg
4KZmRUcFMzMrOCgYGZmBQcFMzMrOCiYmVnBQcHMzAoOCmZmVnBQMDOzgoOCmZkVHBTMzKz
goGBmZgUHBTMzK3Q1KEhaWdIsSQ9Iul/S1pJWlfQLSb/Lv1fJaSXpFEnzJd0lafNuls3Mz
Prrdk3hZOBnEbEbEhsAlwP3AUcE1ErA9ck7cBdgLWzT+HAKcPf/TXxmZlana0FB0krAe8
ilYh4FpgKzMjpgIzQ3NW/PIgEGWVkvcF4jVBBzlnQ9B8Ha5I9lynnOZxcvRH9CpmdKl
Lk7ZWNhQ0a2xzjxSyB7hbeXj7xhhsM1f1pgd7+m2sN9Hjr9j4VpA+fFHZ/jkRTR4o8hzzT
ik0wj4oyImBIRU8aPHZ9khTUzs+4GhQXAgwi4NW/PIgWJJ2vVQfn3U/n5x4BJpeMn5n1mZjZ
jZMuhYUIuIJ4FFJG+Rd2wD8D2wz2wD6e6V9BwV9BwGz8+NLgQ96Qz0+RLgtg8Zmnm3Hsrxqk
nu8qa3wg0H5pRKk7lc8L6zFVdGPiZMtFhwWMQ3+jkiNfz5uYjlR5y6+4BQcE11FrEWKeMXjXBR4cmhwd
5YmH7kx/A5qZ57c8tpf0eUx4FFUbaBKbRkkBDRpLyhR9bg4bJHcMfuGcgWkHr+wGlGrqkZHPPp0gqi
tDTwETCcFogskHQw8AuLjaqUmdVvBQdrzn4nRTT0YXfRENbuqSZZZpI+p4beMz
ZZGtuc2HPk0fobKtYjbGjoMrGM4WPW7cexY9vlEuM3byNfUAyI7yrX9hSyAS7cJnT0T+Ho7u8
jyOsfs1QaAt3nuRaDcyjxs5vQP2uSZnlCDf6G44Eb8IoBRqpnFOTsvcJE0f2KG9WuY4WqW9Z0A
WJtJmNvq0PD3AwpbaS6wwAgqpyzmd6bX8E2aCmJbT3hVgK0D2tnGZV+vqhD0zHfwQNyIhgMDGgw
BOM6nOPdJ5cTbx9eaqfGnuTlDAyuNE0oHUeIgs9XrDWnzsvMqTMyyxNrfhpNY5ePObkyz8DoWaM
LuMzUUNx3LwxNVuvNn6T8IWs7xCoLaNnZ5E3jETZ+Z3ynATgtvmUkozQvcczm/YKNB6yeZVtQvK
4YPHbt5jy3dzkTovnC2mnjIfNSIBMjHaiE2QsvXqzH6KpDBfYJqBAULgS4RRxI+SF7FpAbSchdxnN
```

```
QlcLMzHrCuGZPSrqr0VPAGkNfHDMzG0lNgwLpwv8B4Jm6/QJ+3ZUSmZnZiGkVFC4Hlo+IO
+ufkHRdV0pkZmYjpmlQiIiDmzz34aEvjpmZjSQPSTUzs4KDgpmZFRwUzMys4KBgZmYFBwU
zMys4KJiZWaHrQUHSkpLmSbo8b68j6VZJ8yX9VNLSef8yeXt+fn5yt8tmZmZ9DUdN4bPA/
aXtY4GTImI90kzp2lyIg4Fn8v6TcjozMxtGrWY0D4qkicAuwDeBwyUJeC9Qm/g2AzgGOB2
Ymh8DzAK+J0kREUNVnqd+cErLNKt/8rCh+nNmZqNOt2sK3wX+FfhH3n4D8GxEvJa3FwAT8
uMJwKMA+fnncvo+JB0iaY6kOQsXLuxm2c3MFjtdCwqSdgWeioi5Q5lvRJwREzJwREVMiYsr48eO
HMmszs8VeN5uPtgE+KGln4HXAisDJwMqSxuXawETgsZz+MWASsEDSOGAl4Okuls/MzOp0r
aYQEf8WERMjYjKwL/CriNgfuBaYlpMdBMzOjy/N2+TnfzWU/QlmZtbaSMxTOJLU6Tyf1Gd
wVt5/FvCGvP9w4KgRKJuZ2WKtq6OPaiLiOuC6/PghYMuKNH8H9h6O8piZWTXPaDYzs4KDg
pmZFRwUzMys4KBgZmYFBwUzMys4KJiZWWFYhqSOVn867fCmz6956InDVBIzs+HhmoKZmRU
cFMzMrOCgYGZmBQcFMzMrOCiYmVnBQcHMzAoOCmZmVnBQMDOzgoOCmZkVHBTMzKzgoGBmZ
gUHBTMzKzgomJlZwaukDpEHTpvaMs2Gh84G4JYzdm2ZdutDLh90mczMOuWg0OOuPmvnlmk
+cPCVw1ASM1scuPnIzMwKrimMIbPO2bFlmmnTfzYMJTGz0co1BTMzKzgomJlZwUHBzMwKD
gpmZlZwUDAzs4KDgpmZFRwUzMys4KBgZmYFT15bTJ0z4/0t00w/6OcAnDLzAy3THrb/1YM
uk5mNPAcFG3LHXNA6iByzTwoi0y9pPQv7nD08C9tsuLj5yMzMCg4KZmZWcPORjRo7zT60Z
Zqrpp42DCUxG7scFGxM2vmS/2iZ5so9vgLALhef0jLtFXseNugymY0GXQsKkiYB5wJrAAG
cEREnS1oV+CkwGfgDsE9EPCNJwMnAzsDfgI9GxB3dKp/ZQOxy0Zkt01yx18eHoSRm3dHNm
sJrwBci4g5JKwBzJf0C+ChwTUR8W9JRwFHAkcBOwPr55x3A6fm32ai166yZLdNcPm3/YSi
JWXu61tEcEY/X7vQj4nngfmACMBWYkZPNAHbPj6cC50byG2BlSW/qVvnMzKy/YelTkDQZ2
Ay4FVgjIh7PTz1Bal6CFDAeLR22IO97vLQPSYcAhwCstdZaXSuz2XDbbdbFLdNcNm3PYSi
JLc66PiRV0vLARcDnIuKv5eciIkj9DW2LiDMiYkpETBk/fvwQltTMzLpaU5C0FCkgzIyI2
m3Qk5LeFBGP5+ahly4BJpcMn5n1mVmfqrNazvGdPaz1b3Kxe12oKeTTRWcD9EXFi6al
LgYPy44OA2aX9ByrZCniu1MxkZmbDoJs1hW2AA4C7Jd2Z930J+DZwgaSDgUeAffJzV5KGo
84nDUmd3sWymZlZha4FhYi4CVCDp7evVg7ZmbWNV77yMzMCl7mwmyM2+Oim1qmuWS
vdw1DSWw0cFAws8LeF93VMs2Fe208DCWxkeKgYGYDctglj7ZMc8oek1qmsd7ioGBmw+K0S
55s+vyhe6zR9HkbHg4KZtZzLp7155Zp9py22jCUZPHjoGBmo9q1Mxe2TLPd/mlJnHlnPtU
iJWz28dUHXabRzEHBzKzCH777RMs0kz/3RgCeOH5+y7RvPGq9VonDHBQMDMbZk9+7tWWaNT63JQB/OuXalml3/cx2LPx+84/e8fu19i2ve6SVgTdrx2T3TXFGSKYyvHXzGKtehnpzkRbCvNr7dt6ngE95LqAg3j2eenPMqgNqqnD0AHCo8VoX0vNUKzOQtAW
9f43BbDVJL+HBTMzHrYk6dc1zLNGod2V8ksY5VYgF+GOjprMIq+VyOv3aTIK5KFHyJmpcW6mXT72y5NIKsa79oXs7mzZcV2KDm+lSH3x3WnHtqYTwqGeE5J9Z8SzBTMzKzgom
KDgpmZlZwUDAzs4KDgpmZFRwUzMys4KBgZmYFBwUzMyv0VFCQtKOkByXyXNl3TUSJfHzGxx0
zNBQdKSwGnATsBbf0kvXVkS2VmtnjpmaAAbAnMj4iHIuIV4HxgaiG4pTnIjmoSYT+ziDYFkRZzgoeffBmmDV3jkZLTFZpZUyIkaCgAKbPVAzbfDsJW
SNGDHiPh43j4AeEdEfLou3SHAIXlzA+DB0rOrM8o1cLVNPd4a4Sl6ZXO9VCVCHfCWFNUmAQWLQtLY0Oc7LyItuOj7tLDv7SScWTXPfoXwH13CqRPPqLRhCVHGrK4XZ5Lj08HeQFJUqQ+Vi2RgTtp04Skmqo4
zNBQdKSwGnATsBbgf0kvXVkS2VmtnjpmaAAbAnMj4iHIuIV4CfA1BEuk5nZYkURMdJlAED
SNGDHiPh43j4AeEdEfLou3SHAIXlzA+DBiuxWA/7c5p8ey2l7pRy9kLZXytELaXulHKMtb
a+UYyjSrh0R4yuPiIie+AGmAWeWtg8ATh3A8XOGMq3TOm03046WcvZCWpez99IOR1kH+sHzUvq1yOlbY0oB5eeyCF93SZfZrdnp8/zSv2Lnmo/w
ioqeajx4DJpW2J+Z9ZmY2THopKNwOrC9pHUlLA/sCl45wmczMFivjRroANRHxmqTPAFeRhqSeHEnO3G67yYm6mMJgtUuBnuSeVUmM30Nrz4rpbKTjJG9vQJ+H9wJ67BJGVD5cbSKBL8Jo2/bO6QffmaNDkVtbaOlz9rB8CLOyV
wJnR8S9A8zuDKftqXL0QtpeKUcvpO2Vcoy2tL1Sjm6eX+90NJuZ2cjrpeYjMzMbYQ4KZmZ
WcFAwM7PCmAkKkv696meQtSoulmH2bcTmAZXfEdsLXSodd8jWvHyGm0o6zXz6Nj5nYqfpYUmtdu8aobo4t09U5IhbVzi/56QZp/mVu9SqF9WFOvKnYzmdDGLZVxMxYkpNWJ8xaTJW4wpThtSgmDLuhM27tM8bN+MhBOMNqn0QrQp2+U1JZcK1pLgO9NyFzGAWqVtxmlxaADTWY/wAzZLlmkdv/X3n7M+NVvb7eKYu6e38/2Tnoc0a/tt1xSHe4JeYdJyeUg/qOlO2E2ND+uDWtkJmMwNb9QLqVkumDnSLdWGd7fkr/mS8xjZ0wzBtrVeEnDPNeDUmPZGtD+k1lp89dL2oPtzsY+sxoSZSNLzeuErfK7CBzLjhb4+yv2N9H3Rl1dkzx1bJ
vbtOsxl2KadfXn/kpI+30HeSw6mbCNJ0rlDnJ8q9o2Z9/JwGjMdzZK+UNp8HbArcH9EfKw
u3UrAMcC7867rga9HxHMVed4ZPFtUtWLA
N0i32WBLwBrRcQnJK0PbBARlw+mvPm5m0iv143AzRHxfFW6AxGGoFDuWqH8KpI/J6EaEsBFpyvRHzRQ9a52NnZRzadk4x1s8hgwHs7ScmW9d8aIYBcsKgbsqGkd3a4AWE0CeHlg1gGAOOKMsk41iAmP7Bv8hF4S24U1/eOgATxMOXqBn0V3cTk+Fxq2Pv1zjH9dsyVhPGmi9zABDsb/48Oz8WS0zYuvMm3nUgk5H+ijcqjq0p9+/8ycR4ESoEYjpaP6RR5o
YIyI2krQx8MGI+I+q8wMOjIh78vZ+wOci4h1Nyr02sH5E/FLS64BxEfF8XZorgf2AJ4EvARtHxJ83e/3LOV8UdRNyhYvMlJGgHWYlHpPAk98Tr7E9L2Kk+6C7CmNJL0FjGuMxoXMSYK0y8AXIlkEu/wCk6YIA8Jd40rfRGG0M6lR8n4jRs8AoWt2U5oDz/Qlw68fHBJhAShMxnYXkWNxF9TcMJIBrJ5hTw6kRzRmiOp33nUOvOx5Ew0u1+1WnEFK3TqoR+M96vkKcdLr6w63XrS93/HsnxcDdDE4xnvzS0IWHexjrEHrgZkkXqRQEKq6bXRgOFbLXkLxGhHLfcUYdgqQp2Ly+w78Kb8F3BEGUftMgX6SaryMVkxCW2o7PX6a+/a2/fZzejq8X0y2tCRobiWgjdODTlkZ3PzpWyxZSqqKAn4Jgbz72Wi89/wAnKsbTYaRgsXDUz5SPfjsh03FBLJ/n4+QCf8l1u6hz4tD+DzR8OA5vv6mtXe8Pq4YLgL/gPOFBL8B2+xqo5NBKBCHmUKA5VcYBg0SYrRl4Lg/sunLBxNY2SXgM8fsxKDBaVtSxKTOeTBvZoIdJe
aHD1/i2iNiy0d+oS/sQcBFwTkTc1yDNqs3yiIi/VBzTyWu8JHBvRGzYpJz8sFbAf8Kpf
hgw2OE7A/8OaI+LqktYA3RsRtFWnPLn/W803U7IjYfpDn1zJtk/eF0ulVvi9eBxwM/DPpe
kV+Lcrn0HG++bi2rwFVemZI6mBFxAnlbUnHk4a31jsbuAFqjsbuAfYJ28fAJwD7Fk6dkPSP2slSXu
Wjl2R0j+wlP6NwATg9ZI2r0u/bF3a/YAPA+vUfVhWoJ8AtoiIFyRNBmZJmhwRJ5fLlMJiwaqfK/DMp0ueoZ59Gw7ZdMuS1Y2y3IMwQmd8+mHZgThbx4n8tz0xpKxhZ9jOoLm3xqLqpBobbsOcJT1MFNGmDCb5bWRGWULXq/MdJyAAhRXaqhdfpyGeW1tuzmBaaIMYd5N8+AkuPYaxZQ6ge9mq8H4C7FO3NC7KNfrjQ27ey8IBc0hkA/Go22k58QJv+bzwJewjuyeB6AHDZ1aRTUy2CfHc+5tBn6sBdWd7evLbGvmTGD+KsZgTLuqbjvdbv+r2CvUyBbXsFWPnpHkJzWjWg==
FXOAeYCW+ftx4ALgfLFuFzezUp59StvnQNIwXQv4DhJLwM3RkTV3W3LcuRz7tQPgSOA/wK
IiLsknQ/0+0CTZsnPkvThXO4Dgfc3yljSJ0hLqKwKrEuaQPkDoHUyDs2/z8u/96/Ia
2vgncD4uuC3ImnIdSM3S/oe8FPgxdrOi0LijIu0mpPk8Z0pagvTe/klE/LWUZi7pwiJgLeC
Z/Hhl4I9AVc237dc4Iv43r2y8VkT8scE5TQTuA84slWUKcEKD9DXfB/4BvBf4Bf4OvA8KQi+v
SLtAknfj4hPSVoFoBuCKfR5VO3kPtpB1I7fM84AHgA6Rz259081c20FptO5+9hsZMUKiwLOn
NWG/diNirtP01SXfWpdmA9A9ZGdittP950oW63geAj+a/d3xd+n+rS/tr4HHSmiQn1KW9q
y7tEhHxAkBE/EHSTstqSL3No0DgrrRsSHcvAhIv5WUbUul7d8N/488KUG+RIRD0v6O/BK/tk
```

O+KeBliMivtbobzWxbETcVpfVaw3K+5CkfYH/R7oAvj8iXmqS96GkhRlvzcf/TtLqpfweA
ZC0Q7lGBxyVayXl5d6XBpYnfcbKwe+vpGDVyKb5d+21EelC+t6K83uedNH6oaT/A5wPnCR
pFvCNiJgfEevkMv8QuCQirszbOwG7NyhD269xtgpwr6Tb6BvIajWAKcBngS8DR0TEnZJei
ojrm+QJae2zzSXNy/k9ozSxtZ+I+HdJ35H0A2AL4NsRcdEQnF/LtLX3RYfWi4i9JU2NiBk
50Nw4BPlCe9eAhsZMUKirai0JjCdF4HovSXpXRNyUj9sG6HOhyE00syVtHRG3tPrbETEDm
JGbsGp3QuTHq0raNCLuzGkfAR5hURRv5sm6Y19QahM/G3hbg2Neyc0ekc9vXeDlBuXdq8k
Hpx9JvyctrnU+cBbwmYj4x0DLIemUZn8vIg6r2P3nnFct32mkIFvOt77avSrpPXGrpMpmn
uzliHil9vmRNK4un9Kf0DYRcXPeeCd1/XP5gne9pP/u8MN9XcW+yiar3HSzCzAdmEy6yZh
JqhVdCbyllHyriChuaCLiKknfaVCGlq9xna82eY78HjlJ0oX595O0d+15NZ9jrRzjSTWHQ
l1N/tZcltuAkLRnRFxckW8n59fO++15GrxPSM08K1adW/79rKSNgCeA1csJ6vItX1Oa5Qt
tfPaaGTNBgb5VrdeAJyOiKvp/EjhXqW8BUnX6oAZ5Pi3pGtpoe8y2IN0VXUr6x+1Kuvv/p
KQLI+I7Hb6BDqT/XclrwIGS/qtBGY4GfgZMkjQT2IZUK6hyjaQTgffk7Yb9K9kpwLuA/YD
NSBe9GyLi9wMsx9wGf6eZQ0mzNDeU9BjwMPCRujTl98IqpItkkO7Enm2S9/WSvkRqVtsB+
BRwWUW6g4Gz83tIpPfQxyrSASwj6QzSRbv4vEVEvzv/7IXS46JvrEHa3wHXAsdFxK9L+2d
Jek9d2j9J+grwo7y9P/CnBvm28xoXIuJ6SWuwqFnntoh4qiLdAmBvSbuQakytnAJcAqwu6
ZukGtZX6tLsVrc9D1gq7w+gKih0cn5Vafs0F0bEQJpBz8jNXF8hXS+Wpy64DjBf6Owa0M+
Y6WhuV6l9tzaa5wXgOWBu7Y68lPZ6cntirblA0j0RsVGDvG8Adq41+Sh1dl0B7JjzH5YvD
ZL0BmAr0gXrNxFRucyupItI/Ssz8q4DgE0iYs+q9KXjlifdnX4RmBgR/drIlTo4VS4HsEJ
EPDygk+qf/3Kk5rVmnd2HkZr7Ls5l2B34YUSc2iD9EqQL/vtz+qtJK/c2ulNfCaBJEEXSb
0n9EnOB/63tj4i2AqLSCJqrI2LbiueWr73X2shnVdLF4j2ki+UNpBuAfh3NpWNavsY53T7
AcaRajkhB+IiImNVO2VrkvSGpT0fANRHRKEAOJO923kNbRMTcclpJu0bfARMrRsRf1aBTv
/waq3pwhRYl7T+4Ih/3LtIAiHMkrUaTz9JgP3uLY1A4n+q7+cnAhRHxnVLa2yPi7SqNCpJ
0Z0Rs2j9nkPQA8LaIeDVvLwP8NiI2VN3Iom7KNZrJ9L0z7XfHVHUuLc7vBFJNYXngFtKd9
40R8VBF2puBnWqdnpL+ifT69guouVngSNI37pVHYvS7m5a0MqkGVX9+/ZqaJN0FbB0RL+b
t5YBbGjUf5aaIKyKiaVU7B4PaBRaaj2BrOFqtHflu8vaIWK/iuZYjWCqOWa72ejRJswxpM
MFk+r7GVc2xtcC3Q612kP+fv4yITZr9nXbk859UV45+ne75b36iosz9Xotcq/lPYM2I2En
pGx63joizKtLWj2DbF/h8lEawSbo8InaV9DB9m49zEeLNpbRH54cbkGpWtcEmu5FqWP1qL
PmYKaQRRG+RtCbps9RoaHPbn70qY6n5qF0Tgc1Ld/NHk+7m30O6myu3s3batjqT1G49O2/
vBpyfL0aVQwaHmqSzgY2Be1nU/tqoGt2yf6XOLcB3IuLJNoryn8BlknYGNgTOpWKUTjaTN
NpmF1Lz3kHAwgZpryTd+dxNXftyBVG6O8+Pm3W47UZq874hl+dnDZogW45gK7lM0qdIzSB
FsGl0h672+8agvREstXzfSRr9szywlqRNgP8bEZ+qSD6bXHumvbboJeqai55mCOZASfoGq
dnj9yx6TSo73UllvhH4JX3/51X+m/T/+nLe/h/S/7tfUKCNEWwRsWv+3XAOUynt16BoVdg
8Fg13PoZ0HaqyB6m59o6cx58kNWta6uSzV1nIxeqH9CFaqrS9DPBAfjyvLu2bSW+yv5GGd
d1E+saiZvnXRlp8FpgyAud3XwdpNwF+C/wh/8wDNm5xzAdJI6yOB3ZrkXZ30miru4G3NEk
3N/++q7Tv9gZp7+jg/A7P53dM/rmTNE+h2TFL5XOcSRoQcGZFmjvb2Zf3P1zx81CTv7926
WcCaZ5Eo7Tzyq9bLvtvGqS9lXTHPa+0754GaSv3NynHcaSmto/mn6uAY4fgvfwgsHSbaSt
f/wZpby+/fq2OJ3XY30dqp399k3TXtLOvdG7LlLaXAR5skPa2/PuO/Hu58melwTFtffaqf
hbHmkInd/O7k+5MryXd+bwIvC83CdQPYwUgIuYAc7pS8vbcIumt0WAyU53tSf0J5f6Vt0t
aour8JH2LNGRzZt51mNIIrS+V0pxK3470lUh3ep9WGvlTNaKoNhLj8dwJ+SfSiKEq5ynNJ
7icFnfeEXGipOtITV4A0yNiXoN8a8e8KumqfA6vJ70HPl6XrO0aVrRx91iXvpORSi1HsNT
l/aj6jkxsdEf9a0lvi4i72ylERBwhaS9ShxTvAPaVMyNz0ZWKdDfj7qMnM2N4jn0NsWX
sz9brUWgK1INaOCOhjBlpvxlgVWy81d5Xk/ExqU4VzgNkm112l3Ug2mygVKA0tWzu/9j1E
xB2OAn71+Frs+BQBJU1j0Br45X8ir0rXd/9ArlMarX0q6QLzMolFNVbMqOzq/3Ea/aeRhq
ErDBefVfUAajeQCiuGw9eXYlVT1nwScSvowHRMR/Ub+SDoU+CZpFFHRpBCldtuBUhq7/yF
gW1Kn6QXAz6OuCUnSpqRgWht99BfgoIion2dSS78R/ftLBr3Mg6SPkyZzvY10QVke+GpE9
BuZpjR34UTge8A7WFST3bci7X3A+sBDtHgPdVP+nM4mBYfyDUC/GdBKo/qWy+lepcmwTaU
JpqcCG+W8xwPTyv8/pblADZWDt9LyNJ8D1iS1KNTmljxPCpCnNTi/zVm0ssINjW5YJB1La
rEoD4B4X0QcWZeu489e5d9bHINCu9Qjo4k6IWk+qdmkT5t71R1op+eXg8K2tbvyPMrhusF
eLCTNIC3h8Wwp3+OjupPwIWDLaDCiapDl+DGpbfmqaNHZnNOvCBB9ZxDXpzmaFGTeSqp17
gTcFBHNJrC1+rsdj2BRGrFyMvC+nPbnpNf86Yq0a7NoKC+kkUrP1r+HNLDx+W2TdC9pJnH
9e7nVpLdW+e5NurBOInWov4MUTKtmjdeOWZ2+Qb3f7G2ltda+G2kk0leBzUmTCBvm22Z5q
5ZKuatbQXpxbD7qxOr07Wh7lTRn4SWlJR560cKIaPdrTDs9v28B8yRdS/rgv4e+s3iRdEF

```
E7FNR/QagwRt541pAyGn+orT8RpX5pD6eIRcR++WRKTvkZpbK8fb1o4+Uhi43mt8xjdR3M
y8ipuf8f1SRrhO1TsbKESxVB+Qg2m5nY63JrDaU9zxSc0Wfobwx8HH07fpbRDSd4Chpw4h
4QH2Xlyk0uCB/NSIuzE0925H6x04nBYf6/D9ImhS4JqkZa21SZ/4/V+Q7LdIaTe8idYY3z
Lcdkv6FNFfmzfmGrGYF4OYmx61P+qzW107bqk07KDQ34qOJBmBebha6jL5V7qrRRx2dX0T
8OLfR1yYpHRkRT9Qlq6302sm6LUtIWiUinoGiptDovfkicGcOTOXza6u9tJl8B3k8i8bbn
yqparx9J6OP/h4R/5D0Wq5ZPEW6Qx2wGMAIFnWwCBxpmOtWsWgo77GkkWeV8zu66Mbcj3U
pff/X5Qv94aT1qqrWUWo0UqnWl7ILad7KFZIaTUj9Bmm8/y8jYjNJ29F4olsn+bbjfFKn/
bfoe/P1fFUfWsk5pJuWk0hBbzodjAZz81EL7fY/9ApJ51TsjqqmmJy+o/OTNIF0t1QeC37
DAItby/NA0ppLF+ZdewPfjIjzKtJWtpu2217aohxtjbdXB/M7JH2fdG77klaufIE00mX6E
JT3QVIt6+W8vQxpVMoGFWnbnoiZa3lvj4i/5+3XkUbsNFpapSty4K8X0Xg2eLv5Xk5q+9+
B1MTzEqlW2G9ehaQ5ETElvzc2ywH+tw3Stp1vNynPjZF0d+1/pg7my7im0EKM/GiijnR6s
enk/PId44foPwfihlKajtuZI+JcSXNYdFe3ZzQYPTUUF/8m2h1v38n8jhVJQe460pDGFRt
1SA9AJyNYOlkE7hxSDbKcb9UY/q6KiO06Sa80F2MyfW9Yqjr09yH1mx0fEc9KehMpYFZ5N
ve13QDMlPQUfZciGWi+3fSy0uz830n6NClQLd/imIJrCmOMpImkan7t7v9GUofigiHIu8+
d6XBq0VcRQ3E3Juk40sS/H+ddHyLdedeP8tiEdEHus35W1cU+Nze8O/+sS5oLckOk5c8Hr
YMRLFcBnyaNKttcaSLmwRGxU5N8a0N5b2yUbzfV993QfOb4eaTX904WNePEYJsVlWbxH0G
6Odif9D/fJCIOHky+3STp7aR+j5VJzV8rkiad3trW8Q4KY4ukX5DaImtNLx8B9o+IHYYg7
6uAvaPN9XaGkqQ3RcTjki6g792XSG/4fRoc2unfKY+3vzEqxturg/WzcvolSf0w25FmbL8
UTb6UphskvZm0sNs7SUHsYdL7YqDLM3edOlibS9L9wFtjiC9owz3yZyjkJuEvk5p5l8q7o
90yOyiMMZ20dw8g74tII2muYYg7eTsow4h/SNXZ+lnXkMbP19aKuqlqRNMwlHkZ0kioyaS
JWH8lXSgaLaEx4jrsu7kQOCwimi1D08nfro38WZc04q1mBVLfW8NVY0dartEfQRvD0qu4T
2HseVrSR1jUBLIfqW18KNzCouGPNd0elggMfHhem3l32g/SyfpZd5GWVN+IVJt4VtIt0fz
LfrphNmnC3x00XjK717Tsu5F0Gel/twJwn9IX/TSd6NamgY786QWdDEvvxzWFMUZp4tGpp
C/xCdL6J5+JiEeHIO+Ov/N4qOT25VXogQ+pBrAartICZh8lLTf+xogY1i+VbzTSqJep78x
xqOi7UZrBL+BY4F/Lh5PWX+r6e7PXSNqedDNYX6OvGpbej2sKY8/XSR+c8jvkOvQAAAINS
URBVJj/42n8JTCd6Og7j4dS7lx8jvRmH2ltz+/Ioz/eTaot/IE0x6HP1y4Ok47WM+oR95N
qXeuSOk2fI42EKoJC5NnNkpaKupnOSt8+tjiaTloddSlar5Tcj2sKY0zVnWqju9cB5v8WF
n3n8R4j0AzSE9qd3yHpi6QgMDeql+EeFkrrGa1H6mAesfWMOiHpZyxq8ip/QdEJpTRFsyJ
p8beanm/77xZJD1bNVWn7eAeFsSVPstm2rqZw/WAmHlUMA12ddNf2MjRcusJ6iBos8Nbjo
49aNnn1UrNir8gTWI9rNNenFTcfjT0nkJbP7jM7eJB5drJkhfWgXr74N9GyyavHmhV7xVa
kpWAGVCt0TWEMUvp6wdrs4F8N9I7BbCSUaqbj6IElvEebwdYKHRTMrKc0uqjVjNJaz6jho
GBmZoVBf7m2mZmNHQ4KZmZWcFAwG2KSdpT0oKT5ko5qfYRZ73CfgtkQyiui/g/pi1YWALc
D+3kEmI0WrimYDa0tgfkR8VBEvAL8BJg6wmUya5uDgtnQmgCUFx9ckPeZjQoOCmZmVnBQM
BtajwGTStsT8z6zUcFBwWxo3Q6sL2kdSUsD+9L/i4nMepYXxDMbQhHxWv4OhauBJYGzI+L
eES6WWds8JNXMzApuPjIzs4KDgpmZFRwUzMys4KBgZmYFBwUzMys4KJiZWcFBwczMCv8fk
3rQdE++LtIAAAAASUVORK5CYII=\n"
     },
     "metadata": {
      "needs_background": "light"
     }
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "from sklearn.feature_extraction.text import 
TfidfVectorizer\n",
    "\n",
    "tfidf = TfidfVectorizer(max_features=3000) # 3000 is giving 
good accuracy and precision\n",
    "\n",
        "X = tfidf.fit_transform(df['transformed_text']).toarray()\n",
        "X.shape"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "f3mqHgTQzOEL",
        "outputId": "dd7cf9a9-407c-4d0c-d9e4-e2139682e79f"
      },
      "execution_count": 41,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "(5169, 3000)"
            ]
          },
          "metadata": {},
          "execution_count": 41
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "y = df['target'].values\n",
        "\n",
        "from sklearn.model_selection import train_test_split\n",
        "X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=2)\n",
        "\n",
        "from sklearn.naive_bayes import
GaussianNB,MultinomialNB,BernoulliNB\n",
        "from sklearn.metrics import
accuracy_score,confusion_matrix,precision_score\n",
        "\n",
        "gnb = GaussianNB()\n",
        "mnb = MultinomialNB()\n",
        "bnb = BernoulliNB()\n",
        "gnb.fit(X_train,y_train)\n",
        "\n",
        "y_pred1 = gnb.predict(X_test)\n",
        "\n",
        "print(accuracy_score(y_test,y_pred1))\n",
        "print(confusion_matrix(y_test,y_pred1))\n",
        "print(precision_score(y_test,y_pred1))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "WgqL6zOWAINA",
        "outputId": "e34be648-ee03-4587-b6ea-d24e85f3bc04"
      },
      "execution_count": 42,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "0.8694390715667312\n",
            "[[788 108]\n",
            " [ 27 111]]\n",
            "0.5068493150684932\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "mnb.fit(X_train,y_train)\n",
        "y_pred2 = mnb.predict(X_test)\n",
        "print(accuracy_score(y_test,y_pred2))\n",
        "print(confusion_matrix(y_test,y_pred2))\n",
        "print(precision_score(y_test,y_pred2))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "sd2nm9lcAP0c",
        "outputId": "dacd4cab-e0d3-4d4e-edf7-420b629cc613"
      },
      "execution_count": 43,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "0.9709864603481625\n",
            "[[896   0]\n",
            " [ 30 108]]\n",
            "1.0\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "bnb.fit(X_train,y_train)\n",
        "y_pred3 = bnb.predict(X_test)\n",
        "print(accuracy_score(y_test,y_pred3))\n",
        "print(confusion_matrix(y_test,y_pred3))\n",
        "print(precision_score(y_test,y_pred3))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "3-3BK7_1ARPB",
        "outputId": "d5698b05-c335-4a68-e4ba-085fdf110313"
      },
      "execution_count": 44,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "0.9835589941972921\n",
            "[[895   1]\n",
            " [ 16 122]]\n",
            "0.991869918699187\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "from sklearn.linear_model import LogisticRegression\n",
        "from sklearn.svm import SVC\n",
        "from sklearn.naive_bayes import MultinomialNB\n",
        "from sklearn.tree import DecisionTreeClassifier\n",
        "from sklearn.neighbors import KNeighborsClassifier\n",
        "from sklearn.ensemble import RandomForestClassifier\n",
        "from sklearn.ensemble import AdaBoostClassifier\n",
        "from sklearn.ensemble import BaggingClassifier\n",
        "from sklearn.ensemble import ExtraTreesClassifier\n",
        "from sklearn.ensemble import GradientBoostingClassifier\n",
        "from xgboost import XGBClassifier\n",
        "\n",
        "\n",
        "svc = SVC(kernel='sigmoid', gamma=1.0)\n",
        "knc = KNeighborsClassifier()\n",
        "mnb = MultinomialNB()\n",
        "dtc = DecisionTreeClassifier(max_depth=5)\n",
        "lrc = LogisticRegression(solver='liblinear',
penalty='l1')\n",
        "rfc = RandomForestClassifier(n_estimators=50,
random_state=2)\n",
        "abc = AdaBoostClassifier(n_estimators=50, random_state=2)\n",
        "bc = BaggingClassifier(n_estimators=50, random_state=2)\n",
        "etc = ExtraTreesClassifier(n_estimators=50,
random_state=2)\n",
        "gbdt =
GradientBoostingClassifier(n_estimators=50,random_state=2)\n",
        "xgb = XGBClassifier(n_estimators=50,random_state=2)\n",
        "\n",
        "\n",
        "clfs = {\n",
        "    'SVC' : svc,\n",
        "    'KN' : knc, \n",
        "    'NB': mnb, \n",
        "    'DT': dtc, \n",
        "    'LR': lrc, \n",
        "    'RF': rfc, \n",
        "    'AdaBoost': abc, \n",
        "    'BgC': bc, \n",
        "    'ETC': etc,\n",
        "    'GBDT':gbdt,\n",
        "    'xgb':xgb\n",
        "}"
      ],
      "metadata": {
        "id": "CpPtZz2YAWTj"
      },
      "execution_count": 45,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "def train_classifier(clf,X_train,y_train,X_test,y_test):\n",
        "    clf.fit(X_train,y_train)\n",
        "    y_pred = clf.predict(X_test)\n",
        "    accuracy = accuracy_score(y_test,y_pred)\n",
        "    precision = precision_score(y_test,y_pred)\n",
        "    \n",
        "    return accuracy,precision"
      ],
      "metadata": {
        "id": "p-OEAS9NAb7h"
      },
      "execution_count": 46,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "accuracy_scores = []\n",
        "precision_scores = []\n",
        "\n",
        "for name,clf in clfs.items():\n",
        "    \n",
        "    current_accuracy,current_precision =
train_classifier(clf, X_train,y_train,X_test,y_test)\n",
        "    \n",
        "    print(\"For \",name)\n",
        "    print(\"Accuracy - \",current_accuracy)\n",
        "    print(\"Precision - \",current_precision)\n",
        "    \n",
        "    accuracy_scores.append(current_accuracy)\n",
        "    precision_scores.append(current_precision)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "ttY6J4moAiGn",
        "outputId": "21b769ea-f0ab-4f11-d3d3-102f68c9c489"
      },
      "execution_count": 47,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "For  SVC\n",
            "Accuracy -  0.9758220502901354\n",
            "Precision -  0.9747899159663865\n",
            "For  KN\n",
            "Accuracy -  0.9052224371373307\n",
            "Precision -  1.0\n",
            "For  NB\n",
            "Accuracy -  0.9709864603481625\n",
            "Precision -  1.0\n",
            "For  DT\n",
            "Accuracy -  0.9274661508704062\n",
            "Precision -  0.8118811881188119\n",
            "For  LR\n",
            "Accuracy -  0.9584139264990329\n",
            "Precision -  0.9702970297029703\n",
            "For  RF\n",
            "Accuracy -  0.9748549323017408\n",
            "Precision -  0.9827586206896551\n",
            "For  AdaBoost\n",
            "Accuracy -  0.960348162475822\n",
            "Precision -  0.9292035398230089\n",
            "For  BgC\n",
            "Accuracy -  0.9574468085106383\n",
            "Precision -  0.8671875\n",
            "For  ETC\n",
            "Accuracy -  0.9748549323017408\n",
            "Precision -  0.9745762711864406\n",
            "For  GBDT\n",
            "Accuracy -  0.9477756286266924\n",
            "Precision -  0.92\n",
            "For  xgb\n",
            "Accuracy -  0.9439071566731141\n",
            "Precision -  0.9347826086956522\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df =
pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Prec
ision':precision_scores}).sort_values('Precision',ascending=False)\n",
        "df"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 394
        },
        "id": "niCBefuhAz4G",
        "outputId": "a3cbc4e4-c16a-453b-9f5f-b0046f032a54"
      },
      "execution_count": 48,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   Algorithm  Accuracy  Precision\n",
              "1         KN  0.905222   1.000000\n",
              "2         NB  0.970986   1.000000\n",
              "5         RF  0.974855   0.982759\n",
              "0        SVC  0.975822   0.974790\n",
              "8        ETC  0.974855   0.974576\n",
              "4         LR  0.958414   0.970297\n",
              "10       xgb  0.943907   0.934783\n",
              "6   AdaBoost  0.960348   0.929204\n",
              "9       GBDT  0.947776   0.920000\n",
              "7        BgC  0.957447   0.867188\n",
              "3         DT  0.927466   0.811881"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-e8d83d41-73bf-4703-a65c-
c9450a6133ea\">\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>Algorithm</th>\n",
"      <th>Accuracy</th>\n",
"      <th>Precision</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>KN</td>\n",
"      <td>0.905222</td>\n",
"      <td>1.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>NB</td>\n",
"      <td>0.970986</td>\n",
"      <td>1.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>5</th>\n",
"      <td>RF</td>\n",
"      <td>0.974855</td>\n",
"      <td>0.982759</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>SVC</td>\n",
"      <td>0.975822</td>\n",
"      <td>0.974790</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>8</th>\n",
"      <td>ETC</td>\n",
"      <td>0.974855</td>\n",
"      <td>0.974576</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
              "      <td>LR</td>\n",
              "      <td>0.958414</td>\n",
              "      <td>0.970297</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>10</th>\n",
              "      <td>xgb</td>\n",
              "      <td>0.943907</td>\n",
              "      <td>0.934783</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>6</th>\n",
              "      <td>AdaBoost</td>\n",
              "      <td>0.960348</td>\n",
              "      <td>0.929204</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>9</th>\n",
              "      <td>GBDT</td>\n",
              "      <td>0.947776</td>\n",
              "      <td>0.920000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>7</th>\n",
              "      <td>BgC</td>\n",
              "      <td>0.957447</td>\n",
              "      <td>0.867188</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>DT</td>\n",
              "      <td>0.927466</td>\n",
              "      <td>0.811881</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "       <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-e8d83d41-73bf-4703-a65c-
c9450a6133ea')\"\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-e8d83d41-73bf-
4703-a65c-c9450a6133ea 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-
e8d83d41-73bf-4703-a65c-c9450a6133ea');\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": 48
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "performance_df1 = pd.melt(performance_df, id_vars =
\"Algorithm\")\n",
        "performance_df1"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 739
        },
        "id": "YhkqKldCBE70",
        "outputId": "fd672770-1823-416c-ed3c-367b817e80fe"
      },
      "execution_count": 49,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   Algorithm   variable     value\n",
              "0         KN   Accuracy  0.905222\n",
              "1         NB   Accuracy  0.970986\n",
              "2         RF   Accuracy  0.974855\n",
              "3        SVC   Accuracy  0.975822\n",
              "4        ETC   Accuracy  0.974855\n",
              "5         LR   Accuracy  0.958414\n",
              "6        xgb   Accuracy  0.943907\n",
              "7   AdaBoost   Accuracy  0.960348\n",
              "8       GBDT   Accuracy  0.947776\n",
              "9        BgC   Accuracy  0.957447\n",
              "10        DT   Accuracy  0.927466\n",
              "11        KN  Precision  1.000000\n",
              "12        NB  Precision  1.000000\n",
              "13        RF  Precision  0.982759\n",
              "14       SVC  Precision  0.974790\n",
              "15       ETC  Precision  0.974576\n",
              "16        LR  Precision  0.970297\n",
              "17       xgb  Precision  0.934783\n",
              "18  AdaBoost  Precision  0.929204\n",
              "19      GBDT  Precision  0.920000\n",
              "20       BgC  Precision  0.867188\n",
              "21        DT  Precision  0.811881"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-b5d6d22e-1660-4219-8862-
74a10101c9fd\">\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>Algorithm</th>\n",
"      <th>variable</th>\n",
"      <th>value</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>KN</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.905222</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>NB</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.970986</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>RF</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.974855</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>SVC</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.975822</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>ETC</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.974855</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>5</th>\n",
"      <td>LR</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.958414</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>6</th>\n",
"      <td>xgb</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.943907</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>7</th>\n",
"      <td>AdaBoost</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.960348</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>8</th>\n",
"      <td>GBDT</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.947776</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>9</th>\n",
"      <td>BgC</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.957447</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>10</th>\n",
"      <td>DT</td>\n",
"      <td>Accuracy</td>\n",
"      <td>0.927466</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>11</th>\n",
"      <td>KN</td>\n",
"      <td>Precision</td>\n",
"      <td>1.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>12</th>\n",
"      <td>NB</td>\n",
"      <td>Precision</td>\n",
"      <td>1.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>13</th>\n",
"      <td>RF</td>\n",
"      <td>Precision</td>\n",
"      <td>0.982759</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>14</th>\n",
"      <td>SVC</td>\n",
"      <td>Precision</td>\n",
"      <td>0.974790</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>15</th>\n",
              "      <td>ETC</td>\n",
              "      <td>Precision</td>\n",
              "      <td>0.974576</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>16</th>\n",
              "      <td>LR</td>\n",
              "      <td>Precision</td>\n",
              "      <td>0.970297</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>17</th>\n",
              "      <td>xgb</td>\n",
              "      <td>Precision</td>\n",
              "      <td>0.934783</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>18</th>\n",
              "      <td>AdaBoost</td>\n",
              "      <td>Precision</td>\n",
              "      <td>0.929204</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>19</th>\n",
              "      <td>GBDT</td>\n",
              "      <td>Precision</td>\n",
              "      <td>0.920000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>20</th>\n",
              "      <td>BgC</td>\n",
              "      <td>Precision</td>\n",
              "      <td>0.867188</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>21</th>\n",
              "      <td>DT</td>\n",
              "      <td>Precision</td>\n",
              "      <td>0.811881</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-b5d6d22e-1660-4219-8862-
74a10101c9fd')\"\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-b5d6d22e-1660-
4219-8862-74a10101c9fd 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-
b5d6d22e-1660-4219-8862-74a10101c9fd');\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": 49
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sns.catplot(x = 'Algorithm', y='value', hue =
'variable',data=performance_df1, kind='bar',height=5)\n",
        "plt.ylim(0.5,1.0)\n",
        "plt.xticks(rotation='vertical')\n",
        "plt.show()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 411
        },
        "id": "E_S0059pBOb9",
        "outputId": "351b7165-6c48-4071-a4d9-dc9e954dc0cb"
      },
      "execution_count": 50,
      "outputs": [
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 441.75x360 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAa0AAAGKCAYAAAC//x65AAAABHNCSVQICAgIfAhkiAAAA
AlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3debglVX3u8
e9LI4KAQ6A1BhpBxQFFpgblxihOEb2JoBKlNSF6VUwijolGvQYJ0QSixhsVY3DGEEGDRJI
QMSrgEAcGmRoCtiBTSAQE44QI/O4fVQc2u083p+mqPr1Ofz/P00/vWlW71jo97HevqlVrp
aqQJKkFG813AyRJmitDS5LUDENLktQMQ0uS1AxDS5LUDENLktSM0UIryUeSfD/JBavYnyT
vSbIiyXlJdh+rLZKkhWHMntbHgH1Xs/8/8ZwI79r4OBvxmxLZKkBWC00KqqLwM/WM0h+wHHV
OcbwH2TPHCs9kiS2jef97S2Aa6c2L6qL5MkaVYbz3cD5iLJwXSXSXENlpp532WL58+Rqf44r
Dd57Tcdsdev4an3s+65K03sl8N2Ahm8+e1tXAkontbfuylVTV0VW1tKqWbrbZZuukcZKk9
c98htZJwEH9KMnLAT+sqmvnuZ3SpPXcaJcHk3wS2AfYOslVwFuAewBU1YeAk4GnAiuAnwI
vHqstkqqSFYbTQqpld7G/gFeMVb8G/gFeMVb8kaeFxRgxJUjMMLUlSMwwtSVIzDC1JUjMML
UlSMwwtSVIzDC1JUjMMLUlSMwwtS
VIzmpjGSavmlFGSNiT2tCRJzTC0JEnNMLQkSc3wntYGbo/XHzPnY896x0tSS7pqhtR5
akyA5ccsRGzJlbQd9rNnP9Y61qkvSwuTlQUlSM+xpzdH62vvR2vOvkSc0wtCRJz
TC0JEnNMLQkSc0wtCRJzTC0JEnNcMi71BBnMNGGztDSguRzDW0xjDVXXh6UJDXD0JIkNcP
QkiQ1w9CSJDXD0JIkNcPRg5Jm5Yg6SaUmSmmFoSZKa4eVBaS15GU1ad+xpSZKaYWhJkpphaEmSmmFoSZKa4UAMaR264vZd53Tcdo
eeP3JLpDbZ05IkNcPQkiQ1w9CSJDXDe1rSAuX
9My1EhpakDYozmLTNy4OSpGYYWpKkZhhakqRmeE9L0lpz0IfWFXtakqRmGFqSpGYYWpKkZ
hhakqRmGFqSpGYYWpKkZhhakqRmGFqSpGaM+nBxkn2BvwYWAR+qqiOm9j8I+AiwGPgB8Nt
VddVcz78mE1+euOWD5UkradG62klWQQcBTwD2AlYlmSnqcPqeCRxTVY8GDgf+Yqz2SJLaN
+blwb2AFVV1aVXdDBwH7Dd1zE7AF/vXp86yX5Kk242VMXTT7+tnAlkm
2GrFNkqSGzfdAjD8D7SnPnJwdttdeu6zZZkTYTY4bW1cCSie1t+
7LbVdV/VtVzqmo34P2ZZTdenYvVPHixSM2WZK0PhsztM4jNgAOAy4j9VTXcbWyPYHGVbSvKmhRDZPqQI7bhkxSymZBkTjHntHMLwaO+TnJrZmB/4cJrJvVNVDD
SLJ1kpk2vJHlJkiTNarRQquqbsF1+P2Dea4i8eAGYsyYRyWzvG5xrhAV4BtCF5mgG3FcGl7CsOqVWGoYWJmtZ1BXxiHQVbqKlW3lqSpGYYWpKkZhhakqRmGFqSpGYYWpKkZhhakqRmGFqSpGYYWpKkZhhakqRmGFqSpGYYWpKkZhhakqRmGFqSpGYYWpKkZhhakqRmGFqSpGYYWpKkZoy6n
pYkDe2Kw3ee03Hvzm0j+cvV+0UIsesslgxVSOhJJUTTSVS0JOhCSVJlUqRmGFgSpMRoUXdxuhaAr
qhqElSWqGoSVJaoahJUlqhvSVJaoahJUlqhfSWqGoSVJaoahJUlqhfSWqGoSVJaoahJUlqhfSWqGoSVJaoahJUlqhfSWqGoSVJaoahJUlqhfSWqGoSVJaoahJUlqhfSWqGo
SVJaoahJUlqhqElSWqGoSVJaoahJUlqhqElSWqGoSVJaoahJUlqhqElSWqGoSVJaoahJUl
qhqElSWqGoSVJaoahJUlqhqElSWqGoSVJasbG890ASVpfXXH4H4zmM6brtDx+5JZZT0uSmmFo
```

AxDS5LUjFFDK8m+SS5OsiLJG2fZv12SU5N8O8l5SZ45ZnskSW0bLbSSLAKOAp4B7AQsS7L
T1GFvAT5VVbsBBwLvH6s9kqT2jdnT2gtYUVWXVtXNwHHAflPHFHDv/vV9gP8csT2SpMaNO
XpwG+DKie2rgMdOHXMY8PkkrwQ2B546YnskSY2b74EYy4CPVdW2wDOBTyRZqU1JDk5yZpI
zr7322nXeSEnS+mHM0LoaWDKxvW1fNuklwKcAqurrwKbA1tMnqqjq2ppVS1dvHjxSM2VJ
K3vxgytM4Adk+yQZBO6gRYnTR1zBfAUgCSPpAstu1KSpFmNFlpVdQtwCHAKcBHdKMHlSQ5
P8qz+sD8EXpbkXOCTwIuqqsZqkySpbaNO41RVJwMnT5UdOvH6QuBXx2yDJGnhmO+BGJIkz
ZmhJUlqhqElSWqGoSVJaoahJUlqhqElSWqGoSVJaoahJUlqhqElSWqGoSVJaoahJUlqhqE
lSWqGoSVJaoahJUlqhqElSWrGXYZWkgck+XCSf+23d0ryku6ffFtmlp/UxutWHf6Xfv
gR4zVgNkiRpVeYSWltX1aeA2wCq6hbg1lFbJUnSLOYSWj9JshVQAEmeANw8aqskSZrFXEYPPk
wzOuAk4CHJPkasBg4YNRWSZI0i7sMrao6O6yjJkqbcZWglOWiqaLckVNXHR2qTJElrNPckV
NUxI7VJkqRZzeXy4J4rKB1ai6XB185uZ3kvsBxo7VIkqRVuDszYvwE2GH
ohkiSdFfmck/rn+iHu9OF3E7Ap8ZslCRJs5nLPa13Try+Bbi8qvZ43kVSXK3SXO5pnb4uG
iJJ0l1ZZWgl+RF3XBa80y6gqureo7VKkqRZrDK0qmrLddkQSZLuylzuaQGQ5P7ApjPbVXXF
XjNIiSZJWYS7raT0ryXeAy4DTge8B/zpyuyRJWslcntP6M+BxwjfGLVVkiTNY
i6h9Yuquh7YKMlGVXUag6DTge8B/zpyuyRJWslcntP6M+BxwCVVtQPdjBjfGLVVkiTNY
i6h9Yuquh7YKMlGVXUqsHTkdkmStJK53NO6MckWwFeAY5N8n25WDEmS1qm59LROBe4DvBr
4HPBd4DfHbJQkSbOZS2htvwA3YEji+v1woSdI6dZehVVV/WlWPAl4ckxBPBA4PckXRm+ZJ
ElT1mSW9+8D/wVcD9x/nOZIksaS5OR+eanVHfPjPR+VHfPjVZR/LMkB47Rs7ubynNNYfJDkN+CKwFfC
yqnrM2A2TJA0jnY2Aq6vqWnpGVdeN8t2dtzKWntQR4TVU9qqoOq6oLx26UJGllSY5I8sqJ7bckeV2Skyb2vj7D04Sv
CXJF5OcneT8JPv1+7ZPcnGSj4HpbFsr/QFuyelvvdk637/PyY5K8nyJAdP1fPuvvyLSRbP0o49kpz
ev/+UJA8c9ye/w1zuab2pqs5ZF42RJK3W8cDzJrafB5wx8vkEWRvPRmDefdP6LUq9WkwCw2CzSqdD9iFw
8R4O2D/YR3Yc5zD0qS5ld
VfTvJ/ZP8CrAYuIFurMG7kzwBuA3YBnhA/5bLq2pVMxc87pKGXd8ZHTCmwTOz+zNQMiQeGNLgzMnhMwYd5yxa8XKKHvIxZlQ/x07+KuNFe8E25mCN2XfjBXr2U8h/Ss8a5HuORbcDx/9ceBw8EAnF/E/32lXw
up9DwceDfxbn42LgGvW9mebK0NLktryaeAA4JfpwuFfdAG2R1X9Isn3uGNy2HJvjLWKV5P
UYO+q+mk/bmHT2Y5l5SWqAiyvqr3X4me429Zk9KAkaf4dDxxIF1yfpx+38pv84ft9YD0JeNAcz
nEf4IY+sB5BN5Bs/sjI36cwO8APjq1HsvBhYn2Ru6y4VJHnW3f5o1ZGhJUkOqajndRA9XV9U
1wLHA0iTnAwcBGZ2OjJRcAR3HkS9J8AeyW5AHgycPhU/TfThdqRSc4FzgH+19r9V
HPn5UFJakxV7Tzx+jpgpVVzfqHj31vu0nNp+xinNvsYryF028Pgd4wtxaOyx7WpKkZhhakqR
mGFqSpGYYWpKkZhhakqRmGFqSpGaMGlpJ9u0nbFyR5I2z7H93knP6X6X5ckaXr2YUla3yTZP
0n1DxE3b7TntJIsAo4CngVcCBZyR5KTx3a7n9s9kjSfNrj9cdMT4e0Vs56x0G
566MAWEY3q8Uypia/HUqSRVV16xjnnjZmT2svYEVVXdo/QX0csN9qjl8GfHLkNkqjSBiXJF
sDjgZfQTf1EkkVJ3pnkgiTn9R0GkuyZ5Nn+TnJvkW0m2TPKiJO+bbdO8/MWkuTHSd7Vz4
xd5JDk5zRn/fomZnkzmzw0yRf6856j51ZUuWujBla2wBBXTmxf1ZetJMLNw+wIo5+u4tzbw58s6p
B2AL43YHkna0OwHfK6qLguCDCwPbBrv6DVTQvZ2oZvT8NVVtQvdZ/2s1xjC1uzx57kg6y2s5N6p
2qaqvAu+rqj2r6r6tHAZsBv9McdCxzVn/d/0c0I/2HgRQBJ7tOX8T05H4rKp7m
eTgJGcmOfPaa69dx20xdpGYto7vKRf/7MrpA+tuqugWgqn5At9zIZB9BJE7tOX/I/yHIU2BY5wL1+
J7Scl+WY/B+KTgUcl2RLYpqpO/Olls77M97U1X9tKpOB3bsbF5hcBwHAcecevJpujZYZZ2/Zls
zkQeMUq9lFVRwNHAyxdunTQmAezMTSD7SW6B2Ve5SdGte1XAGWtwmlu4c+dmcvmSm2Y6Gkk
2Bd4PLK2qK5McqxOPqJOp94Z8q3XoAKu4dXHD8rGqcqF+r55nnA4Tu/zNAfPIc2xOKDhbe6eWBS
A3NAcAnqupBVbV9VMLsgPHOvpvWO5HaTQ1OlOb+DSxAm2c+bMTLSmiwGHBBAUcv+rdRHAo6bwvwHRXBnYm64n
9fbQDAKrqR8BVM/evktwzyb36bQ9TfGr3rHlrbFzwJb7I8F6QqHyY1m6D2HNGSM4P6X
I6rKntQkjScZcCJU2UnAA8ErgdDO6wdRvKAfLPd84L192b/RBdHX6ILuQuA9wNmzVVRVNwI
fBC6g+Dt0qyecB/063eCVV9d902fDRNfmhRl2apKpOBk6eKjt0avuugMdsgSeuDN
RiiPoiqetIsZe+Z2Hzd1L4zuPNikDNeuIrzbzG1/RbgLbMc9x26y5R30ve4dmQNR42vLwM
xJEkbiCRPpetlvbeqbm4QmfrgmN73U3RSEnSOlVVXwAsAedHfea09LktQMQ0uS1AxDS5LU
0uSFqgkt/ZLP12Q5NMTD/auzTkP70f/rWr/7yU5aG3rWVHD0rSOnDF4TsPOoHCdoeeP5f
nvn5WVbtCN5M68HvAX83sTLLxXOf8mzH9rO0s+z+wJudbU/a0JGnD8BXgoUn2TvKVJCcBF/
/ZLlbyjX1bkvCQvn3lDkj9Ocn6/6r9XCQQ9S9Bg7k4LNnPhHXQEqBb6Ujfz/HU/1PH6P2/2nY+A/
k9+vLT0tyZL8EyiVJfm2uRg4IdVtWeSewJfS/J54JC5p9
BF0S5s8tqp+2s9ROHnOrYBnWdnuRwukUoX9Pv27iq9kryzL58lZc
cJ9nTkqSFa7Mk5wBwBn0s03+OG+/FtVdVn/+teBg/rjvglrvglsRTe90lOBj1bVT+H2JUm4C
fhwkucAP53mhTkuANPNucUSC5G2C2BdsR+G3GwTc1zK3bRtBwR6RWBzWVkuR3ED7Qrn0
ZOu7pqztxVd2SZC/gKXUjYuXuOJ1nsmhzYt/+kDe2+zBR9N/P8TvGGp11RI8lAmA0RPIv

```
3hmxOEslwe3AO7TT4z+WmCXyf39nII3TNyv+h3gdNaSPS1J2rB9iO7y3NnpumHXAvtX1ee
S7AqcmeRmuhU73jzxvi2Bz/YLQIapWeN7vwt8oA++S1mDxR5XxdCSpHVgjkPUBzW9fEhfd
hpw2sT2bXRh9OZZjj0COGKq7EUTmystCjm53FRVncMsy51U1T4Tr69jDe5peXlQktQMQ0u
S1AxDS5LUDENLktQMQ0uS1AxDS5LUDENLktQMQ0uS1AxDS5LUDENLktQMQ0uS1AxDS5LUD
ENLktQMQ0uS1AxDS5LUDENLktQMQ0uS1AxDS5LUDENLktQMQ0uS1AxDS5LUDENLktQMQ0u
S1AxDS5LUDENLktQMQ0uS1AxDS5LUDENLktQMQ0uS1AxDS5LUDENLktQMQ0uS1AxDS5LUD
ENLktQMQ0uS1AxDS5LUDENLktSMUUMryb5JLk6yIskbV3HM85JcmGR5kr8fsz2SpLZtPNa
JkywCjgKeBlwFnJHkpKq6cOKYHYE3Ab9aVTckuf9Y7ZEktW/MntZewIqqurSqbgaOA/abO
uZlwFFVdQNAVX1/xPZIkho3ZmhtA1w5sX1VXzbpYCDDknwtyTS7DtieS7DtieyRJjRvt8uAa1L8
jsA+wLfDlJDtX1Y2TByU5GDgYYLvttlvXbZQkrSfG7GldDSyZ2N62L5t0FXBSVf2iqi4DL
qELsTupqqOramlVLV28ePFoDZYkrd/GDK0zgB2T7JBkE+BA4KSpY/6RrpdFk3q3pLhdeOmK
bJEkNGy20quoW4BDgFOAi4FNVtTzJ4Ume1R92CnB9kouBLn4Ume1R92CnB9kguBU4HXV9X1Y7VJktS2Ue9pVdXJw
MlTZYdOvC7gdf0vSZJWyxkxJEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQ
kSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJz
TC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQ
kSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJz
TC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQkSc0wtCRJzTC0JEnNMLQ
kSc0wtCRJzTC0JEnNMLQkSc0YNbSS7Jvk4iQrkrxxlv0vSnJtknP6P6XyXFQHsMCYNSDXh5M85JtPcT
ywCjgKeBlwFnJHkpKq6cOrQ46vqkLHaIUlaOMbsae0FrKiqS6vqZuA4YL8R65MkBjhtY
2wJUT21f1ZdOem+S8JP+QZMmI7ZEkNS5VNc6JkwOAfavqpf327wCPnbwUmGQr4MdV9fMkL
weeX1VPnuVcBwMH95w3aGrjubrzv7rAu67KuDbeu66pq3zEao3FDa2/gsKp6yer/
9JoCq+otVHL8I+EFV3Wek9pxZVZB7JhkhySbAAcCyKNgXTBMTCvCvBJqeqwPHWGUm6HdxSlObmYvnlmrQ2n0tE5jZMkqRmGliSpGYZWQ
5KMeg9SbUtyz7mUrWUdzxnyfOuTJB+b7zborhlaA0jyxCSP6V8/L8n7jq6kFsTdb5
34HPfLsmeSZ4xS/kzk+wxQn3v6B8uny5/zJjHjRqhvnJOqbY3SXJwksEfuMi5FaDI2b
N9P8tkkDx66nt7X51i2NnT+4kKpssPSPK0gat7zMDn0wgW5Df3JD8CZkaYpP+
96H7eTapqsJ87yVF0/9jvmeQSYAvgc8CvAh8BXjhUXdzxs9CffyxHAi+epXw58FFgpVlL1
tKTgTfMUv5B4DxgpRUC7q4kBwJ/C/wkyXeAE4AP0/nF0zbXXM/6ebuLoZ/fbBwKfBB47VAV
JfpluirTNkuzGHf9O7g3ca6h65smhwP6zlJ8G/BPwbwPWda+pP787qaqzB6xLd9OCDK2q2
nJyO8kWwCuAlwMnDlzdk6pqpySbAlcD96+qW5N8EDh3wDo3zGzO9u9pKe64+8e5S1v9Zn1i3QFP3nUiDt0XWCujJ/J8rJqmBv0X2r
a3x0X9vLPx+Y6uEBqCPrcWs4pKFBfx7V5nhO4NlEaK0gdPn8H0lkQOBqZ3Z5tcmc8P32lxN3N0+pfP5EFbb+uDtB13/EbbA9Ok4eqPkR/jnWkVrWDnzQMKYLvrvBVdwfOswadxVPPS0p0ovDqqzB6XlLKdKlhq
WLMNYq+q2JLN+gKyFtwB7VNWKJLvT9UQOqKp/GrieG6ehPAz8HMVk8i96uTjR8kuQdJd0yOaJMLFXLyv1KcnfyxDfbJ9NpDpJLsZcYuBXGf3WehzzHzMq+/o20VlOvtrQWvDmqG8FvMD2bl4XEdl1yAkaXFDF
3d86P4IePPAdT0iyWz/rgNUVQ3dW7lnVV07XVhV1yXZfOC6tuHOf353qpLhv6zpbliQoTU
jyX2B1wAH0X3j3bOqqrh+4mpsAquqmJJdX1a39diX5xcB1zBHm/Bj3tCB1DZ3WWqi9s41
f8syY5V9Z3JwiQ7Aj8buK6sFzN+ck3xnjMBK8kv9y3y3/t15I7ju6D7/lMPbu4tqrq48D
Hkzy3qk4Y8tyuzAz4zZHrmHTvJBtBRtX1S2cF0rZpv7VOuXBRlafW/gD+k+ID4C7
FZVPxypuvsneR1dcIfKvyhSRvB94ykSA+0wTqezx/CnxphOpPo4pftwfxtwVl+
2lO7B89cMXNfk3xHAfSe3q+qvBqrnLLQmvnWPnnPrhjnofptk9ybrof1QWB34I1V9fkB6
7h5tl74iD4DfDDJVX9/u0gVmQDxcn+QlwLd39lx9N7x/wg4kkb13/qr6r606H
qWk0bNgKWVdWxA51vc+DDWJ7AOX3xLsCZWsDlDNR3+Po/p7eADy6L74AeGdVnT9wX
av7+6qqOnzI9tI+taLJOdW1S5Jng78PvAXwDFVtd1I9e0I7AfQx2xT8QH1H8E9uHkmjGLyY1K0KsQkLQMWL5DXPNyq/ZSfavF2qu
7+6qqOnzI+talJOdW1S5Jng78Ht2l0E9U1e4D1vG+dbloaz9i9m3AS4HL6b4ELKH79/9/k
nVTXY1Ywkvz4T8EkWA8x2aVLza6GG1mGs+v5Psx9M/bfoV9Bdez+J7ib0IXS9ynOratBFN
vtRbo/qN5dX1aVDnn+inrOH/GBdi3a8pqr+38DnnG2I+A+B86vq+wPXdXFVP0nNdW
JSb5dVbsNXM/D6ZYKerq6O0sGzbXOzYCH9psrqmroy8JYeSvd/aiO6gLwFe
G+rnxkL0UINrSVVdeUq9v1GGf3zgHUduprVVV/NmBdTBs88BTg/nrAfg8KlBGmtb10fpvtDsQ
NczXkQXXoM9ppBuyaHP0I3A/Dbdv7/dgJcBz6mqbwVxV11S9o4d/f5n4GcDBVXVZX/Zg4G+
Az1XVu4eoR2tnoYbWP9A4/BBfQLYBWHF4ojHP12PQcgU+6W+AUW6IBIAXGlkNvLsLjN0
tiuqm4aqo7+3LvQDc1+HnA93RDt46vqP4esZ6G4Evr2p/VT1T11Ix6lvWhXpsJOT2
Oqqr/7rcfABwDLAO+XFWPXt3717CujYBmjBdgUur6sZ0i6uDWJWmiKj+N8q/AkV121lT5E+nun
630fN9A9Y4e/km+DTytqq6bK6oHqvvungU5EAN4HfD5JP7ZkRaukUoxXwA8cciKup
dM6+TbAm8muY4Zp+Pohs8O6fbr9/2w+quGDqz+3OcC5wJ6wJv6v83PR/4RpLvAn9fVR8cuPRPLvAn9fVR8cuMprG
```

```
f7P6u4Y4xvcdjOB1fs+sKSqfjD06NL+EYFtgRf0TwqcPsLIyIdMB1Zf9+lJxpwRfWPgkbO
E/2PpvvAM0WO9x3RgQXdfqx+tqPXAggytqjo5yc/pRqTtT3cTdy/gCVV1w9D19cObX0f3c
OrHgd3HqAfYJcn/zFRL9zDp/3DHkPd7D11hf7nnG/2lyXcD76MbmTakH1fV6QOfc1ZTD57
faRfDD6EGODXJPwOf7rcPAE7rB7vcOGRF6WYP2ZPu0i5069XtXVVDPqu10sCmCT8ZsJ5pS
9ZB+N98N/dpHVqQoQVQVV/sLweeBvw78OQxeiVJ3gE8h27dnZ2HHlk3qaoWjXXu2STZk+4
y1nPpns/5W+748B3SZSOcc1bTD56vA+8HHg48vt/+OHBdP3z7SQPX9Uxg16q6DSDJx+nuO
w0ZWkuSvGeW8tDdTxvLaVPh/1yGD//JL4WTAmw6UB1aSwv1ntbMt+kA96S7rHYrI/RIktw
G/JxulNHkH+ZovZ+xJflzuvtZN9Bd5jy+qq4asb43VNVf9q9/q6o+PbHvzwfuKaxTSS6gu
3T1l3Q9uSOBpVW19wh1nQfsU1U/6Ld/iW4gxmCzVCT53dXt7x90Hlw/su853BH+XwNOqIX
4AabVWpChpbXTj4g8taq+0m8fRPfN9nLgsJkPxQHru3304PRIwvVlZOHd1fcEjgT2ALaku
3R35ExvaOC6lgFH0A1WCPAEusERxw9d13zqJw+43sDaMDnLu2azP93kuCR5At0H4TF0Q4z
HuNmeVbyebbs1v6CbimozuktMl40RWABV9UngcXRD0k8A9h46sJJsneStSV6VZIskf5Pkg
nSz1z/0rs+wxvU9LslpST6TZLe+53oB8N9J9h26Pq3/DC3NZqOJ3tTzgaOr6oSq+hPueMB
zSLWK17Ntt+YMutDaE/g1YFmSMe4LztiTrof1hP710P6e7pL7jnRL5VxKN7jkn4EPjVDf+
4A/p3vs4kt0M7L8Mt3P9xcj1Kf1nJcHtZL+2+yuVXVL/8zbwVX15Zl9Qz5b1J/zVrqRZzM
j+H46swvYtKqaHW6cZGlVnTlV9jtDPlQ8cd7p0YPLgDOGvCc4MVVUgMsnH8ZOck5V7TpUX
dPnTHJRVT1yYt/gs31o/bdgRw9qrXwSOD3JdXS9hJl7Ww+lu0Q4qHU9KnJdmg6svmzwwOq
ti9GDk6sYTD/TNMZlz8lzTk/d5DfuDZChpZVU1duTfBF4IN1MADMfDhsBr5y/lmkO7gvMX
Nq9zwjnf3CSk+h6wTOv6bd3GKG+mWHok88lztTnMPQNkJcHpQViXYwe7Kdrmnb7KuHr6iF
xbbgMLWkBSfJA7hiA8a2q+q+Bz78fsG1VHdVvf4tu3bgC/njyGTtpDI4elBaAJBsnSVVdQ
3cfaxO6y7tDewPdsjgzNqFbsHMfujW8pFEZWlLjkryMbi6+y/vXX6Qbhn5ckj8euLpN6s7
L/ny1qq6vqivoVjeQRuXlQalxSZbTTW+0Jd2CjA+qquuS3ItuyPujVnuCNatrRVXN+qxek
u8OueyPNBt7WlL7bq6qG/rezoqZ5TWq6qcMPzv5N/ve3J0keTndw8bSqBzyLrVvsyS70X0
J3aR/HcYZFv5a4B+TvAA4uy/bg26WjP0HrktaiZcHpcYlOXV1+6tq6CVQSPJkYOay4/Kq+
tLQdUizMbQkSc3w8qC0gCR5NLATE5cFq+qY+WuRNCx7WtICkeStdM9L7QScDDyDbkj6AfP
ZLmlIjh6UFo4DgKcA/1VVLwZ2YZz5B6V5Y2hJC8fP+hneb0lyb7oHjpfMc4SiIXgAAAOPS
URBVJukQXlPS1o4zkxyX+CDwFnAj4Gvz2+TpGF5T0tagJJsD9y7qs6b56ZIgzK0pMYl2X1
1+6vq7NXtl1piaEmNm3i4eFO6GdfPpZsN4zHAmVW193y1TRqaAzGkxlXVk/pZL64Bdq+qp
VW1B7AbcPX8tk4alqElLRwPr6rzZzaq6gLgkfPYHmlwjh6UFo7zknwI+Lt++4V0lwqlBcN
7WtICkWRT4PeBX6O7p3UWsENVvWReGyYNyMuD0gJRVTcBpwHfBXanmx3j2/PZJmloXh6UG
pfkYcCy/td1wPEwzpIk0nzz8qDUuCS3AV8BXlJVK/qyS6vqIfOAImrAr9ruOhbbpkc9qNuh
5YJKn0N3TkhYce1rSApFkc2A/usuETwaOAU6sqs/Pa8OkARla0gKU5H7AbwHPr6qnzHd7p
KEYWpKkZnhPS5LUDENLktQMQ0tNSLJ/kkryiH7/kxyiH57+yQXDHj+DyXZqX/95onyQeuRtHYMLbV
iGfDV/vdBJVlUVS+tqgv7ojev9g2S5o2hpfVeki2A3qQXDHj+DyXZqX/95onyQeuRtHYMLbV
iGfDV/vdBJVlUVS+tqgv7ojev9g2S5o2hpfVeki2AxwMvAQ6cZf+9knwqyYVJTkzyzSRL+
33Lkpyf5IIkR06858dJ3pXkXGDvJKClWZrkCGCzJOckPazZGB0A/fFH//NPyJJ9Psll/jtOSvDv
JmUkuSrJnks8k+U6St4395yJtiAwtfWAZRpSFFBBsk8OS3Hvb+/HHwE/N0/95GOp
HtmaVdgzyT79+/ZHPhmVe1SVV+dOVFVvRH4WVXtWlUv7It3AZR6aqkcBNwLPnaj75qpaCnw
A+CzwCuDRwIuSbDXQzy+pZ2ipBcuA4/rXx7HyJcLHz+zv15A6ry/fEzitqq6tqluAY4En9
PtuBU6YY/2nBFxSDHzLhnqp/7Lz4e1pa3x0AfKKqHlRV21fVEuAy7
tyL+RrwPIB+BODOffm3gCcm2TrJEIrsoe2ulzqPMXSe4x2E8gaTCGltZ3y4ATp8pOAN40sf1
+YHGSC4G3AcuBH1bVNcAbgVPpVvA9q6r+uS2Lt5dSv6j86o4cjr1yVxYVNo1jz6rd7ZjbVnt+NukDtV3ZTkI
cAXgIdX1c3z3DRJA/OauixaCe9Ety3EPuiU5/sDAkhYme1qSpS1AxDS5LUDENLktQ
MQ0uS1AxDS5LUDENLktSM/w/VRskaK89f0AAAAABJRU5ErkJggg==\n"
         },
         "metadata": {
           "needs_background": "light"
         }
       }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "\n",
        "svc = SVC(kernel='sigmoid', gamma=1.0,probability=True)\n",
        "mnb = MultinomialNB()\n",
        "etc = ExtraTreesClassifier(n_estimators=50, 
random_state=2)\n",
        "\n",
        "from sklearn.ensemble import VotingClassifier\n",
        "\n",
        "voting = VotingClassifier(estimators=[('svm', svc), ('nb', 
mnb), ('et', etc)],voting='soft')\n",
        "voting.fit(X_train,y_train)\n",
        "y_pred = voting.predict(X_test)\n",
        "\n",
        "print(\"Accuracy\",accuracy_score(y_test,y_pred))\n",
        "print(\"Precision\",precision_score(y_test,y_pred))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "A_GF3xnkB8F1",
        "outputId": "12f1fa64-734d-4f76-f378-df5196241949"
      },
      "execution_count": 51,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Accuracy 0.9825918762088974\n",
            "Precision 0.9918032786885246\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "estimators=[('svm', svc), ('nb', mnb), ('et', etc)]\n",
        "final_estimator=RandomForestClassifier()\n",
        "from sklearn.ensemble import StackingClassifier\n",
        "clf = StackingClassifier(estimators=estimators, 
final_estimator=final_estimator)\n",
        "clf.fit(X_train,y_train)\n",
        "y_pred = clf.predict(X_test)\n",
        "print(\"Accuracy\",accuracy_score(y_test,y_pred))\n",
        "print(\"Precision\",precision_score(y_test,y_pred))"
      ],
      "metadata": {
        "id": "wFF6DJ7cCBFq"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "import pickle\n",
        "pickle.dump(tfidf,open('vectorizer.pkl','wb'))\n",
        "pickle.dump(mnb,open('model.pkl','wb'))"
      ],
      "metadata": {
        "id": "We9O6hfeCEyh"
      },
      "execution_count": 55,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [],
      "metadata": {
        "id": "XgwO5rZUiwaz"
      },
      "execution_count": null,
      "outputs": []
    }
  ]
}
```