```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "#### 1. Importing Necessary Libraries"
      ],
      "metadata": {
        "id": "L_xjCYknzLv_"
      }
    },
    {
      "cell_type": "code",
      "execution_count": 88,
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "fbfa75Yz2DJv",
        "outputId": "7bf7bbb4-08f1-4733-b15f-23059edbe8a5"
      },
      "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: openpyxl in
/usr/local/lib/python3.7/dist-packages (3.0.10)\n",
            "Requirement already satisfied: et-xmlfile in
/usr/local/lib/python3.7/dist-packages (from openpyxl) (1.1.0)\n"
          ]
        }
      ],
      "source": [
        "import pandas as pd\n",
        "import numpy as np\n",
        "import matplotlib as mlt\n",
        "import sklearn\n",
        "import scipy\n",
        "import seaborn as sns\n",
        "import missingno as msno\n",
```

```
        "import warnings\n",
        "warnings.filterwarnings('ignore')\n",
        "!pip3 install openpyxl"
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "#### 2. Importing the Dataset"
      ],
      "metadata": {
        "id": "IpUT2FmxzWSF"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "from google.colab import drive \n",
        "drive.mount('/content/drive')"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "Ad4dgKTh4SWj",
        "outputId": "af086104-2d25-4e99-8394-71f88cbc9b70"
      },
      "execution_count": 4,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Mounted at /content/drive\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "data=pd.read_csv('/content/drive/My
Drive/Nalaiyathiran/weather.csv')"
      ],
      "metadata": {
        "id": "DcHm815542Eg"
      },
      "execution_count": 5,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "#### 3. Analyse the data"
      ],
      "metadata": {
        "id": "oJ3aKHINzd9R"
      }
```

```
    },
    {
      "cell_type": "code",
      "source": [
        "data.describe()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 364
        },
        "id": "ugOwafqN47pV",
        "outputId": "7e1edf5e-4d9a-4803-d53d-aa3759d4be73"
      },
      "execution_count": 6,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "          MinTemp     MaxTemp    Rainfall  Evaporation
Sunshine  \\\n",
              "count  366.000000  366.000000  366.000000   366.000000
363.000000   \n",
              "mean     7.265574   20.550273    1.428415     4.521858
7.909366   \n",
              "std      6.025800    6.690516    4.225800     2.669383
3.481517   \n",
              "min     -5.300000    7.600000    0.000000     0.200000
0.000000   \n",
              "25%      2.300000   15.025000    0.000000     2.200000
5.950000   \n",
              "50%      7.450000   19.650000    0.000000     4.200000
8.600000   \n",
              "75%     12.500000   25.500000    0.200000     6.400000
10.500000   \n",
              "max     20.900000   35.800000   39.800000    13.800000
13.600000   \n",
              "\n",
              "       WindGustSpeed  WindSpeed9am  WindSpeed3pm
Humidity9am  Humidity3pm  \\\n",
              "count     364.000000    359.000000    366.000000
366.000000   366.000000   \n",
              "mean       39.840659      9.651811     17.986339
72.035519    44.519126   \n",
              "std        13.059807      7.951929      8.856997
13.137058    16.850947   \n",
              "min        13.000000      0.000000      0.000000
36.000000    13.000000   \n",
              "25%        31.000000      6.000000     11.000000
64.000000    32.250000   \n",
              "50%        39.000000      7.000000     17.000000
72.000000    43.000000   \n",
              "75%        46.000000     13.000000     24.000000
81.000000    55.000000   \n",
              "max        98.000000     41.000000     52.000000
99.000000    96.000000   \n",
              "\n",
```

```
        "       Pressure9am  Pressure3pm    Cloud9am    Cloud3pm
Temp9am  \\\n",
        "count   366.000000   366.000000  366.000000  366.000000
366.000000   \n",
        "mean   1019.709016  1016.810383    3.890710    4.024590
12.358470   \n",
        "std       6.686212     6.469422    2.956131    2.666268
5.630832   \n",
        "min     996.500000   996.800000    0.000000    0.000000
0.100000   \n",
        "25%    1015.350000  1012.800000    1.000000    1.000000
7.625000   \n",
        "50%    1020.150000  1017.400000    3.500000    4.000000
12.550000   \n",
        "75%    1024.475000  1021.475000    7.000000    7.000000
17.000000   \n",
        "max    1035.700000  1033.200000    8.000000    8.000000
24.700000   \n",
        "\n",
        "          Temp3pm     RISK_MM  \n",
        "count  366.000000  366.000000  \n",
        "mean    19.230874    1.428415  \n",
        "std      6.640346    4.225800  \n",
        "min      5.100000    0.000000  \n",
        "25%     14.150000    0.000000  \n",
        "50%     18.550000    0.000000  \n",
        "75%     24.000000    0.200000  \n",
        "max     34.500000   39.800000  "
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-c285a786-6f23-499b-82b1-9b287a830f95\">\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>MinTemp</th>\n",
        "      <th>MaxTemp</th>\n",
        "      <th>Rainfall</th>\n",
        "      <th>Evaporation</th>\n",
        "      <th>Sunshine</th>\n",
        "      <th>WindGustSpeed</th>\n",
        "      <th>WindSpeed9am</th>\n",
```

```
"      <th>WindSpeed3pm</th>\n",
"      <th>Humidity9am</th>\n",
"      <th>Humidity3pm</th>\n",
"      <th>Pressure9am</th>\n",
"      <th>Pressure3pm</th>\n",
"      <th>Cloud9am</th>\n",
"      <th>Cloud3pm</th>\n",
"      <th>Temp9am</th>\n",
"      <th>Temp3pm</th>\n",
"      <th>RISK_MM</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>count</th>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>363.000000</td>\n",
"      <td>364.000000</td>\n",
"      <td>359.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"      <td>366.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>mean</th>\n",
"      <td>7.265574</td>\n",
"      <td>20.550273</td>\n",
"      <td>1.428415</td>\n",
"      <td>4.521858</td>\n",
"      <td>7.909366</td>\n",
"      <td>39.840659</td>\n",
"      <td>9.651811</td>\n",
"      <td>17.986339</td>\n",
"      <td>72.035519</td>\n",
"      <td>44.519126</td>\n",
"      <td>1019.709016</td>\n",
"      <td>1016.810383</td>\n",
"      <td>3.890710</td>\n",
"      <td>4.024590</td>\n",
"      <td>12.358470</td>\n",
"      <td>19.230874</td>\n",
"      <td>1.428415</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>std</th>\n",
"      <td>6.025800</td>\n",
"      <td>6.690516</td>\n",
"      <td>4.225800</td>\n",
```

```
"      <td>2.669383</td>\n",
"      <td>3.481517</td>\n",
"      <td>13.059807</td>\n",
"      <td>7.951929</td>\n",
"      <td>8.856997</td>\n",
"      <td>13.137058</td>\n",
"      <td>16.850947</td>\n",
"      <td>6.686212</td>\n",
"      <td>6.469422</td>\n",
"      <td>2.956131</td>\n",
"      <td>2.666268</td>\n",
"      <td>5.630832</td>\n",
"      <td>6.640346</td>\n",
"      <td>4.225800</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>min</th>\n",
"      <td>-5.300000</td>\n",
"      <td>7.600000</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.200000</td>\n",
"      <td>0.000000</td>\n",
"      <td>13.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>36.000000</td>\n",
"      <td>13.000000</td>\n",
"      <td>996.500000</td>\n",
"      <td>996.800000</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.100000</td>\n",
"      <td>5.100000</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25%</th>\n",
"      <td>2.300000</td>\n",
"      <td>15.025000</td>\n",
"      <td>0.000000</td>\n",
"      <td>2.200000</td>\n",
"      <td>5.950000</td>\n",
"      <td>31.000000</td>\n",
"      <td>6.000000</td>\n",
"      <td>11.000000</td>\n",
"      <td>64.000000</td>\n",
"      <td>32.250000</td>\n",
"      <td>1015.350000</td>\n",
"      <td>1012.800000</td>\n",
"      <td>1.000000</td>\n",
"      <td>1.000000</td>\n",
"      <td>7.625000</td>\n",
"      <td>14.150000</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50%</th>\n",
"      <td>7.450000</td>\n",
```

```
"      <td>19.650000</td>\n",
"      <td>0.000000</td>\n",
"      <td>4.200000</td>\n",
"      <td>8.600000</td>\n",
"      <td>39.000000</td>\n",
"      <td>7.000000</td>\n",
"      <td>17.000000</td>\n",
"      <td>72.000000</td>\n",
"      <td>43.000000</td>\n",
"      <td>1020.150000</td>\n",
"      <td>1017.400000</td>\n",
"      <td>3.500000</td>\n",
"      <td>4.000000</td>\n",
"      <td>12.550000</td>\n",
"      <td>18.550000</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>75%</th>\n",
"      <td>12.500000</td>\n",
"      <td>25.500000</td>\n",
"      <td>0.200000</td>\n",
"      <td>6.400000</td>\n",
"      <td>10.500000</td>\n",
"      <td>46.000000</td>\n",
"      <td>13.000000</td>\n",
"      <td>24.000000</td>\n",
"      <td>81.000000</td>\n",
"      <td>55.000000</td>\n",
"      <td>1024.475000</td>\n",
"      <td>1021.475000</td>\n",
"      <td>7.000000</td>\n",
"      <td>7.000000</td>\n",
"      <td>17.000000</td>\n",
"      <td>24.000000</td>\n",
"      <td>0.200000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>max</th>\n",
"      <td>20.900000</td>\n",
"      <td>35.800000</td>\n",
"      <td>39.800000</td>\n",
"      <td>13.800000</td>\n",
"      <td>13.600000</td>\n",
"      <td>98.000000</td>\n",
"      <td>41.000000</td>\n",
"      <td>52.000000</td>\n",
"      <td>99.000000</td>\n",
"      <td>96.000000</td>\n",
"      <td>1035.700000</td>\n",
"      <td>1033.200000</td>\n",
"      <td>8.000000</td>\n",
"      <td>8.000000</td>\n",
"      <td>24.700000</td>\n",
"      <td>34.500000</td>\n",
"      <td>39.800000</td>\n",
"    </tr>\n",
"  </tbody>\n",
```

```
            "</table>\n",
            "</div>\n",
            "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-c285a786-6f23-499b-82b1-
9b287a830f95')\"\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-c285a786-6f23-499b-
82b1-9b287a830f95 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-
c285a786-6f23-499b-82b1-9b287a830f95');\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": [
        "data.info()"
      ],
      "metadata": {
        "colab": {
```

```
      "base_uri": "https://localhost:8080/"
    },
    "id": "Jl4Y0HVG5JJv",
    "outputId": "c38a6730-3655-455f-df36-4d24fd40b77c"
  },
  "execution_count": 7,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "<class 'pandas.core.frame.DataFrame'>\n",
        "RangeIndex: 366 entries, 0 to 365\n",
        "Data columns (total 22 columns):\n",
        " #   Column         Non-Null Count  Dtype  \n",
        "---  ------         --------------  -----  \n",
        " 0   MinTemp        366 non-null    float64\n",
        " 1   MaxTemp        366 non-null    float64\n",
        " 2   Rainfall       366 non-null    float64\n",
        " 3   Evaporation    366 non-null    float64\n",
        " 4   Sunshine       363 non-null    float64\n",
        " 5   WindGustDir    363 non-null    object \n",
        " 6   WindGustSpeed  364 non-null    float64\n",
        " 7   WindDir9am     335 non-null    object \n",
        " 8   WindDir3pm     365 non-null    object \n",
        " 9   WindSpeed9am   359 non-null    float64\n",
        " 10  WindSpeed3pm   366 non-null    int64  \n",
        " 11  Humidity9am    366 non-null    int64  \n",
        " 12  Humidity3pm    366 non-null    int64  \n",
        " 13  Pressure9am    366 non-null    float64\n",
        " 14  Pressure3pm    366 non-null    float64\n",
        " 15  Cloud9am       366 non-null    int64  \n",
        " 16  Cloud3pm       366 non-null    int64  \n",
        " 17  Temp9am        366 non-null    float64\n",
        " 18  Temp3pm        366 non-null    float64\n",
        " 19  RainToday      366 non-null    object \n",
        " 20  RISK_MM        366 non-null    float64\n",
        " 21  RainTomorrow   366 non-null    object \n",
        "dtypes: float64(12), int64(5), object(5)\n",
        "memory usage: 63.0+ KB\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "#### 3. Analyse the data"
  ],
  "metadata": {
    "id": "Lx3lNCgbzlmT"
  }
},
{
  "cell_type": "code",
  "source": [
    "data.isnull().sum()"
  ],
```

```
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "RiGjGxt35UOw",
    "outputId": "9da12e6b-0c8a-4ac6-e300-866493939009"
  },
  "execution_count": 8,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "MinTemp           0\n",
          "MaxTemp           0\n",
          "Rainfall         0\n",
          "Evaporation       0\n",
          "Sunshine          3\n",
          "WindGustDir       3\n",
          "WindGustSpeed     2\n",
          "WindDir9am       31\n",
          "WindDir3pm        1\n",
          "WindSpeed9am      7\n",
          "WindSpeed3pm      0\n",
          "Humidity9am       0\n",
          "Humidity3pm       0\n",
          "Pressure9am       0\n",
          "Pressure3pm       0\n",
          "Cloud9am          0\n",
          "Cloud3pm          0\n",
          "Temp9am           0\n",
          "Temp3pm           0\n",
          "RainToday         0\n",
          "RISK_MM           0\n",
          "RainTomorrow      0\n",
          "dtype: int64"
        ]
      },
      "metadata": {},
      "execution_count": 8
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "import missingno as msno"
  ],
  "metadata": {
    "id": "g8swjQsO5t2J"
  },
  "execution_count": 9,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "\n",
```

```
        "msno.matrix(data,color=(0.55,0.255,0.255),fontsize=16)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 684
        },
        "id": "fu4stHdb5zLG",
        "outputId": "f37de055-5f84-4f1e-fc4a-c4cc72b9b28a"
      },
      "execution_count": 10,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at 0x7ff7e4e7b190>"
            ]
          },
          "metadata": {},
          "execution_count": 10
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 1800x720 with 2 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAABcAAAAKkCAYAAAAncZwgAAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOydd5RlVdHFfzWRSWRBlJ
wkSZYsKKgIEkRAogEwkEURREDJSZKIIgiCIhn0kxxUcs6KBEElCZIFVGAEur4/dt15p+/0DDP
DMPe+7tprndXd59731qk+956wT9Uuc3cSiUQikUgikUgkEolEIpFIJBKJRCKR6G8Y1HQDEolE
JBKJRCKRCKRSCQSiUQikXgvkAR4IpFIJBKJRCKRCKRSCQSiUQikUgk+iWSAE8kEolEIpFIJBKJ
RCKRCQSiUS/RBLgiUQikUgkEolEIpFIJBKJRCKR6JdIAjyRSCQSiUQikUgkEolEIpFIJBL9EkmAJx
KJRCKRSCQSiUQikUgkEolEol8iCfBEIpFIJBKJRCKRSCQSiUQikUj0SyQBnkgkEonEAISZjW6
6DYlEIpFIJBKJRCKRSLzXSAI8kEkokBBjNbFrjYzD7RdFsSiUQikUgkEolEIpF4LzGk6QYk
EolEIpGY5hgGrAEMNrO33P33AE4AMEMNrO33P3ahtuTSCQSiUQikUgkEonEe4L0AE8kEolEYgDBzAa5+63
GFmtlbDzUokBjTMzCb2dyKRSCQSiUQikZhyKRSCQSiUQi0V+QEiiJRCKRSAwQmNlgd3/TzKYzYHzgIeA94A1gHGmFlPyqA
TzKYHzgIeA94A1gHGmFlPyqEk2oJ4Xt+KhK3HmNlI4EVgL3d/veHmTRVERMZbZjYGOOAOBeycJmgL4
AvYBbnP315ptYSKRSQi0f2wdDBJJBKJRGLgwMxCGALGwMxGALcFLwmgL4
nm9I/58C5gfeAbYwN0faqxhUxHh9X0zsu8RYHFk53eAs939l0gkEolEIpFIJBKJRCKRGFgwMxGALcDLwBHAa8CSwFHA3cBeycJnmgL4
QSiYGFTYBZgX3d/VJ3v8Hdfwx8GlgOOCDlUBItwrrAU+j5/CiwJfAmcJmZLDkFkw94NapIuCyB
S/3PAl4GPA1cBxwBfMLMZpnkDE4lEIpFIJBKJfoQkwBOJRCKRGFgYBswIvAQi4kKG4RpgT5Qc
8xtmtk6k6DbUwMUPSh+f0W8Ji7/8Pd/w1cCewZCGaLN9PSKUeh+T0k5IjmEYDz7r7W+wM
rA1cDHwA2DrJMETiUQikUgkgEokpRxlLgiUQikUgMLLyC5v+lYVVlvoUoP7P2c9IE3x9YKNGW
pcYsK/MzC4GDiA
hpfv/MzC4GDiAStwK4+5vAdcCuwH+B35rZ0o00eApQ0/z7+DXAT8vR2d38r7hniZ7mOBLwIXAYc
D28f/JZFIJBKJRCKRSEwmUgM8kUgkUgkEol+iCAT357AteAteuuDRf1nkQf4aq64oaLcWJdcfkHkDK4j8ZPNSGNT0hwBWAQAyBiIH
7g2x9m1b2C4Zoa2D9M
gJPUIJDx78bSoBCMzK6CX9OfvPMVOAqHWSydNGOhgaygB+hN2kKm/Mmx/gHAm1w1Jkl+ZUn2
8u4/tS9hkb1EF4w7Ks4wKrCdH74Eo4T+QB9wBtzXkk2Sde/yDY+UjkEgkEolEIpEYqHc3+/PYwbDhLpcKUZg8FTi4KBj
9bdsR41gTVo60ZZ5L2xVUHgXBYRRYbjGq6z9KBf4ID
b+G+ZUf3gtuo5W5KrKGcv1bfSpOGNqH/lZA8fKLhRVtUcC9CD6PlS+nxrhBhqzLJbWf8m
E8f2tDyZHP0ySUiE7rPjW8Awzf66rCQaLV2jfXJ2M66IOFtXt+I8WqvnswQCxl6CLL6Gb
dq3e3r0QSDLRuIs4qB4KYXe6RHNRmbMEMI8PgoF7C2QS73aQ0ZVXqRdHHJvmMlgXc6GdK
fZSXbh2pIZcMBI5kVKWtPK3WgObA/Cp7cWbwXcI70X3E6mJd8NJAY2JbQGZJMEZ2o5aDZHG7
zmsBUbvrZ4jWtRzlcf8aZS2DAPS+Yqj5hGcRcAmXKMWgzeFJQpBADjZmMZSb7qH4
HB2wWQj3zjJvKzQhTIj2wSXCjdFO7DMjtl8Me6xdnXI8CWqcYx4Jyc3Zw5CXQ7Gcy
lmPnZpbK3PFzDv6B4cv6OTy/Xn0LADCfEzhaBvjbcoKdGECMDPCkdn0zBNcOEzyTG2
8qBAYc9yflI4nDcNeIoUbEBv1gKJsRWS9/Y0FhOR0KF3DyXCBggMlfw05nJ8KJcU
xNHUBN7aH8nMNI8Zs9gMn1FrVoSfVKzSBbQTmCzlaKJtWyHDrXcrSX5OGPs4rFD
0YChkMKZd3UjtybaPkgCN7ALBO2wrFOKcdL2l3k2q2XlcS4nrY8LYRCPyz+3hA4
Mkjkk+EnrqlLE1XzYlBLM+wb0hGTWRp9oaL8+kxL9fn6Aj+rv1Zix3U7G/a1QPy
```

iyS0z6DknjcCW5jZSdDbrtxwJRKJRCKRSDSHwjljgabbkph8JAGeSCQSiUT/w4LAB4ETgL9E3
TLAh4DvuPt94XGLu//W3X/l7ueGrMTglD1JTGu4+83AHsCDwA5mtqG7v+3u/ytI8FUQebxhk2
2dVISkyzi4+7MooeUJwPLAji70FCT4rsDDwPvrn08kEolEIpFINAsz+yFwtJmNyrVadyE9vBK
JRCKR6H+YB1gIeCAIxC2BM4B93P1YMxsN7Ghml7n7n8sPdoknbaKLMSG5Ene/wsx6gAOAn4Xc
9UXxDA9190fMbB7g6Wne6MlEocM/CBgK9Lj7m+7+dzP7KUryuV9ofh9akuBmtinwcmp+JxKJR
CKRSLQOfwd2Bj7i7temLEr3IAnwRCKRSCT6H14E/gssYmbLA79E5PdhcX1tYC3guobalxigKI
jh6YBPAnOgBK3XunuPu18ViSL3BU4xs+1C87tKCvlk+T2NGTIRVFEUcdB0HIq8+JeZ3evu33P
3x8zsyLj9YDOjRoK/FN+TG6pEIpFIJBKJhjCB9eYJaB91kpmt6e5PNdC0xBQgk2AmEolEItGl
mFjiPzO7A5A5gfmAHYz90PCW/UBYCfo0SDWyTBlphWqJ7XSNB6GdLBNvScHg2c5u4Pxr3rAPvEt
d3c/byGmj1FMLNRwJ3x5/3I1hVRMs9t3f1vZjYv8A3kRXSUu3+3gaZOEfoae/obYd8NiVUTiU
QikUi89zCzg4D7qvWoma0L/Ag4DTjS3f/XZPsSk4YkwBOJRCKR6EIzQP+Lu
zWxz4FSK8vw7cAywNfAsYhsL23upvpFWi3Yjn9SbgZWAn4G/AXcCi6Hk9zN3/Evd+GnlQP+Lu
6zXT4klHSZia2SeB7wJfD+mWESjy4sfAX939Y3Hf3EjyZSHgo90gd1IcZIwA1kUSL9e5+z8bb
tpUQ2HjKOCrKEHrpUhW6l/Nti6RSCQSicS0QkjTnRt//gQ43d3viITl66L122N5cN5+JAGeSC
QSiUSXoSKtw5P2VqQnPBiYHngDeSL80Mw+DJyCNMFnA+4Fng2CfI7F2qJaYaIQDgYWAlFHzx
rZr9BCVrPAPYGfgEc7e4PxGdWAu7oluc0SOHzgB7gJXFfprg2BFg/rh/i7vtH/fuAF+Od7grN
7xh7bkRjy2gku/Q14DJ3f7f7PJtk0thITNrcCswAhgJAp7Pt7d/9pk2xKJRCKRSLw3qK/FzGxO5
MAwBng/8AxaH/wQuAG4292370uziXZhUNMNSCQSiURiaiEIpn6PIMqGAf+HpEw+5+7zAcsCzw
HHmNni7n6fu6+IvBM+AWwEbBTk3cFCEOb3D3D3W8M
Td3BDbZ5cfBz4MPAx4HUAMxsaG6K3gCuQx/uSlU3u/nyh/936TZOZGXAQ8DiwDvBZ4A7gTGAr
MxveYPPeNcI+kOf334E1kYf+wcAuwD5mtnBDzXvXiIOofo0uGi8SiUQi0SKUBLaZZRjVT6F1z
qtItu484DPA7cCjwLpm9gWAbljHDWT0+wVQIpFIJAYGCoIJMzvJzFbr55vgeaOcDPPwl6pYDFg
O+4+73h9wE7n63u1/t7o+7uwfR1soEgonuh5nNb2afM7NdzWwlM5M5sZwN3HAr8DTovkrKshDew
/hQzPrcCbSBrlc+V3tu2wxswWNLNtzOzHZrZheHHj7pchT/Ynga+a2YrhET0krr+ODqlmRFEb
49BmKaJyLI3N3YzAJe5+i7tfAuyAJEJOADbvRhK8OJCoNq8fBO5y9wfc/Tl3PwD17ZeA73YjC
R5RPz3x+/JmtpCZd5d50u6YmCvmakWa2o5ltZ2arN92uRCKRSLQfBfl9MvBTM/tE1B2Gl9l1fc/
eTgFWAK4ElUATuet249hlosBCecolEIpHo3yh1rCNJyRqE/n8rZlshDeV93P3
ICN/fw8zOd/c/lx9sM9FWoj+GEfZHm0qY2arAOUgCZAxKwnq5mf3M3S9y9yfivrmBnC3J7
eqT5fSlw8zRv/CTCzFYBfolsnBXYXBjgCJe3E3c8ys7eQx/BvzGxjd781bGxjd781PIsXQZulK7olYVKRb
2AEiiIZgWy/Ma4Pcvcnzeyb8zZGfAG5m57n7G820evJQ0/zeC3gb5VWokl2NcPfX/X3f3wcBA/FO
gxs3FyPd2AQqP+lygqaHrgVjM7xt0vb7RxUwnRj5V8zWzAKGCEmX3AKGCEmX3b3Y9ptnWJRCKR6BI8By
JnG9mxwE/QAfg55vZN9z9OGB7M1sfRcMdH44eiRYjCfBEIpFIDdC0q0qIgv5dBpMU3gFu6heid
HBQE6hPAv4DVzGxl4HREfh8Wt34CWAF5J7J7QefemR9zeiuCASHyYHAiHriXEzWj4DUoUd
DLwErApcAywahBQP4/be4DXgDXN7Hkkj/I54DZ3vz6+b0jbIhXM7CPAZeh9OwV5en8D+I6ZnQ
D809173P286Of9gRuCcBwGzAe8Auwa39f6fo9ndgxwGzAHOtQAGGZmX3P31+K+p4p8E70Fa7s8
DXUGqFuT33UjreyQwE7CImV3o7v81s+HuPjZI8B7gcCTp0zUEOICZHYA174PzAlsApxiZru6
+68bbdy7RPE+fQd4NgMHVJtCRxlZtNX2vuJRCKRSEBvR6oK/PKU/NBuhgEBkzWoPEbYBkzW
ywixC42syv6S/6T/o5MgplIJBKJroOZTVf3LDSzU4JPBgfE3kkSPPIt3uBDJ0
iaA9394Ajjnx/pKT8NbN72/0HNg39f5Bn8JnAU8GTb5C+mBNY7cem5qI9eA04CLnD3F7uBDJ0
YIgphX2ADd3+kqN8WkcWvA7u5+8lRfyYivf+Nkre+AKzQNtK7gpl9ELgGuBbNa9ELgAeBDY3d3/FfVrAT9F
yT3fBv5dPM9bIM/weYHfIomQuyJqo3UEf4myfWb2Y6SDf6Aiwp8C6kn31gm
+2D8WzcA2m37w48C3wLSbv8Cfisu79akeBx/9ZIz77tNvaaQ8zsUOAJdz8x/l4bSbssDny9G0
nwPmw8CHjV3Y+Mv+dCh067o+dy/0Yamk gkEolWoZw/zGxFQqIuJO2qe5ZDe5L3ofXcv4Hvufs
p077FiXeD1ABPJBIDEIDElZLAhWh6YkugCn540Nm9t3apbOOQV+3CKBFdlSyya/s2yJlKy3RnMzvc
zL5kZsuHlvAmKBP5/4DnzWw+YDnzPkToK2DL+B62e7e7wuy8GzkTeRYA90CHAiQX5be7X750
TNKYDoWOBL4npnpN7u7ezc8r0kyeG5GHNnNQqD8HkcavA7ua2ScA90CHAiQX5be7X750
J6+2e5+7+Kds6IpCSuQsmQTg9SHHc/G3kKPwisDPwvyO/hbSdOoy9GmxI7PQ/83N2vD8mPDYF
7gB2R9/vw4nNPuvv3i2iH1iLaOMrM9gPmAa5394ficONQtOFdBLgovIfHmtl08zOO6S28oJS
vJmthiR4/l7d4+5XAgcC9wMnmdnDnnnovyylsJ6a37vYGbfR4c146S/3P1J4FjgaDTe7tdQcx0OJR
CLREoRzSikNdhaKXZG4zs7jnsnZQ2LFO1P/ZZ3XRdvm4fcGjtgi2RSCTeK9RkCG
YE3nL3l5tuV2KSMQZt1nt5qbn71CcnteMEd2bqP0BQKm3PSYsBPH7nnx2CbBq1u2twicNx1+y2M9oc/ksbWwWs
4H3nw/RR5K/4N2Kx6zttKtJX9YmYfQvIKa6PQ9fcBZyKbvmFmF3ejjtlmw/l7s/HNd/
AXwRGGJmB7n7s930vNY8zyOBLwCjzOyOmKo55vmDqr7zuHklsmaHm6aTkP8mPQCTHEOFsz3EH4
lmPu98O42Qz1kdyH9ch3fIe4HvAHGb2hLs/4u5nmnNb6B9Tiv4v4Vt1Tiv4VtIxmUsIr2rCJ
xXzWxjFAq8PdLDPqoemdPWsaeGNYCKEN0Hxo1Nr5tZpRm9C/B/Jj33XuuFNttYbOzPBNZHkRY
zAXeb2Q1xkIq7/yH28HsDF5jZZ939ooaaVnwjub37WgOMXQgNUusA56I+542s2NRlMZ+ZvYP
70gyJRKJRGKAoXDAORWtTb+LpO0WQBFDpwE7mtlls4T4/A+UG2RtFb3ZFnpNEB632CEskEompj
SCjKlLxMpQk6REz+7aZdto4xKTBHd/0d1PdfzOznZvbb4todSPt6AeBgM1sz6rvKs7AbAB9

s3Ik3XcfdZEQl1BXCamX3R3R8FPorkX9YCPgNs0gXk9+AayTsaea0/6u4vuftfgPWAp1BixPW
7zRM8/v89ZjbUzGZHnu3vB/5Z3ePuX0be4FsC+5rZbN3yvEZI6K5m9r6oOht5ke4NbGRmo4Kc
+iKwOiIZjwW2NrMZ+/rOFpLflY2zAVTkt5kNjvf0h4gE3trdD3X3w4GvoI3U/NX3hCf4wUiu6
IQu8hq6APgxOthYH8Dd3zCzYe7+HyRh80d0KLlFY62cTNT+91cjregXgA3NbMFqbHIlKj0G+B
Hq04OmdVunBOUcYma7AysCuyHvtVuBnYCPl/e5+x9QRMrlwEPTtMFTgNo8+XWUF2MtYFkkNbQ
WsLOZfaC6yd2fRvI9O6NEtolEIpEYwDCzZdH8/j0U4Xe9u5+G9iAvoGiwYXH7IFcukP3c/b5m
Wpx4N0gP8EQiMWAQHl09JgmN64H/IM/ZOVFm50XM7Mgg3hItRxBorwLbmNnP3X07AHe/KeQH/
gAcamZ7ufu13eJRC+M82kYiz8oFgV+7+91x7fdm9iTycjvOzP4c1x6OAnQOexpo/juiFnK4H7
AYknF5wju6ykPc/Rkz2wC4CIWuDzezC7rBE7x22HYh6scXkPfhoLinSqq3nZk5Cq+cycx2qf4
PLceXEZE2KDy+nzWzdRGBdhIi9J9FBzR7unTOH0Hrz+maavRk4st0bDzN3V+K+p44qFiHjqxJ
5bl/HfAWkoQZFwXg7uea2VvAPW30GqpFZFRtfsDMfgIMB3Yys2fd/WB3/19FgpvZZsABSHqp1
Si8+cfNB0HoX4oSsf4MOMLMvll4Dv/PzH6IDiLPaqThk4lifF0TrXGOcfdTo+5yRBCfBmxrSt
71dnzuCjO73iOxaZvhncSleyOP7wureRIR34OAb6NH+9ggvys5lBOgnYl2E4lEIvHeoY9IwzF
IAu2lWNdV65/HzGxvlPhyO+AnbXPSSEw+kgBPJBIDAtZJQDcEWArpX+7u7o/F9XuR59MQMzss
SfD2IDaxw4FlEHH4IEpu9bKZHQW8jMKZB7n7NgDufmOQ4FcAJ5rZV939hoZMeEcEmd/j7q8W1
Qcifd03UALBkjD9iynp53rAh4C769/pLU58WYQcnolsuA95Rw81sz3c/cggjwcHCb4+Svi5Dy
LDW02A1w7bLkaeI2egZ3ht5FH7Be9oQY9196+Y2UxI6uaVxho/efgGWkseDgw2s18GCb488B2
kwzsS+Iy7Xx6fWQBJ3Lzax/e1EXUbT3X3F4pN0r+rGwtSdRXgEeQZXem/VxuqViYYtI6OcnU4
MTuSUiLGm6PRoduBZkZBgg+Pceub5fc0ZcfEYB35s5FIdmg0mj/OdMn0nI3mmJPi/pIEHwv8K
upba2MJMzsA9cvzwLZRZ3Go8TVE9p8KfNnMflcRwd1Afhf4GrAnmhM2BzDptb7m7juGo//uSK
Lnx0F+j0OS34lEIjFwEHNgdUB8MjrUfg45Lcxd3FfJF96N1quzNNDcxHuAJMATicSAQBAQ0yG
pgZGIbHys8nhz96PNrAd5mXqQ4A9P9EsT7zmCqNgP+BgiSEHabJeb2d7u/lQQwYNQYitqJPj6
KAHfk+N/eztgZkshG92k6/1UXDoOJd3bFBHhV5aEKUqmOBYlaOsKFAtKzGxBZN/67n69mS2Oi
KedzWysu/8oCLnBQaquBowqCcc2ohpTgkhcAZHZ33f3P5qkQnYF9jFpSX+pRoJvXBzWldrhrU
NBJlaHNN8D3jSzc9z9n8Bhcd+wIEoHI/J7c+QB3XqSbSI2nuHuzxfPctVnhg6kdkMSEvdU39X
yvqzI79HIK3hRFBH1a+Cnrgiah+PAEUSC97gkX3odRrWZGPZORMatdBKXDgJ2N7NdgWtR3gFH
SVmPNrM9XTJT5fe01sYazgDWBZZDUjXXVYc00Z9fQ57QFwProDml23AG0v3eGenVXxKHGdO5+
xtBgvegZLtPIPmTRCKRSAww1PYgxwGfQvmkHkPz4FFm9kd3v7H42MyIIP9nfK5rcvQk+kYS4I
lEYiDBUfKnZYAHC3JjqLu/6e7HmmQIjgFmNskQPNFoiwcwgqj4PfA6cCMin0YDOyBZgpXN7BN
Bgv8sPlYnwa8xs3k9En21DWa2Ckqmchvw54r8jkXak0HKAHwmQri/GYSpIU3XHqSR3RUoFp4n
AW8iz8S74tr9ZrYL0ojePfqxJMGfa6zhk4Egv4eiPnUUUll5Aj9vZscTiRJjHf3l6NNqHOoG8
ntwQSZ+N6pHoXd0sJn9wt1fgHHyESOQfvS3kBfx9vE9rd1ITIKNp7n7izDugHVmJPWyF/L436
xL+nKQ904i+AzSiX4WSdnMamaj3f2SggTvQTkWnnL31usoW2+Zi5PQZvZLaEO7BPB9RHxv6ZK
Y+jWy8UwkK7XPtG/15KH+LsXz+4iZfQ5t8Dc1s4fdfRwBHP25C4p+e3zat3ry0JfnfYypP0Dj
yk5mdra7b+GStalI8J3N7FHCsz+RSCQSAw/FHmRGdAj+XeB3sQb6GcrbcrGZfRW4BXl9fxNJp
FwV39HKNWti0mHZh4lEYiCgILunQ5vajZDu94HhLTTU3d+Me/dFZM3KbSYu+jPMbHrgXiRV8y
3gwaJ/hiGSaQ/gUeBj7v6Smc2BiLXvAhe5+yZxfytJNjNbBhH8ZwNHeUeOp3pWK6/SDwDHA59
GuuanAPMCnwdGACt6F4VxRz+djhKU3YbsehXGkcdLo8SCcwInu/sRTbX13cCkbb4f8CKwtne0
aTElVNwREWuXu/sGzbRyymHS3YzkMCNvKM395YnaK0wmTb+A
PgsSgK6abfYCBAHNmegDeFW7v6CmZ2BvKNGoHF4H3e/JO5fDM2RR3eDfQBB8K+KkiTf4e7nFd
fmQCTxjMAycSA1DEUeXd12G0tiOCIthnqhMW9m8yH7ZkBa4D+pfX7c+qetKObFEWjumB0dnj7
jksl6H5Jd2gYRGpUcyojyALxb3slEIpFITB2U+0BTPo8vAS+hQ+/bivvWR7J3awL/RhJpBmzg
7vdO84Yn3hMkAZ5IJPol+vIUKq6NQJvB5VCY8xF9kOCVjEGrvff6IwrS6VFgS+Bl78gLVN6KQ
9Fmdx8kcbJ9kBazIWJ8O2BxlxRDqxB2DEf6q0OBXSpv2dp908G4BG1zosiETRBZfDHSGD7S3V
+f2PPeRpjZkuhgY2vga+5+avxfKhJ8KeCXSJ5gDW95Qsja4roMsdwFSdmcDBzq7o8Xn5kNHeS
sAKzeLeNMMTYeDmwMrOeRMyG8ao4BtkLv5i/d/fm49n7g2fhsq0mod2HjJxBh2tN2G0sEQXoU
cJq7X2LSwl4NWAkdRF2NyMaj3P2i2mdbb2eQwkehje1bwMbufnHtXV0XuADNJafXPt9aG2vk9
94oMmhmFFG0Q3FfRYLPCPzA3U9sor1TguJ9HAPchHRap0cSYHcA33D3e2JM3ROR4Fe4+1aNNT
qRSCQSjaM2R34IrWt2Az6MCPDzave8H63Ll0T70Bs8o8H7FVICJZFI9DsUnkKjUCLBRZHn07m
InPizmW2CNoNVKH5Fgg9x97dis2XdQkr1FwQJegxKnne2u78U9dXi5O0gLd4Mj8sVkHbp+4B/
uPtzUX9EX6RyGxDPVg+wPHBh2U4z+yBKVvZpJNdzh5kd7u4PmtluiBBeFvinux8Yn5mu9PZrE
yZ0gOTufzIl1RsN/NTM/uvu51rApZf9BZTstO3kdzXeDEbrqlHIswR3Pz7GoUORdvSRFQkez+
pB6ICnaw7bimiKORAB9Xhx7WUz+zYKI90L2Xyuuz/j7s/AODKrlWRihSmw8Xx3f9rdfw/jnvv
W2RjkPcAY750M8Gkkg3GXmW2LknhujWRQngauQVrSc5nZ8+5+S/XBttlpZrOg+X4WFDn0nzg0
PR8Rw1shT/CLvbdEzf1IsmiG+ne2zcYK8S5Vm/bzUJ6Mi4E/oTwDcwHbuPTqHzWzjdE66HAze
8vdT2ms8ZMI6+jTD0IOC88i8uJu4KvokPwPZrZWkOCHA28De5jZX919v8Yan0gkuhKlU0Oiu1
HMkZejaLajgf+iNc9eZnajuz9tnUTlzwAXRUn0QwxqugGJRCIxNVGRK6Zw57tQGNMQpDd8JHC
8ma3uSsC2MXAn8BXgEFMiunEb3Vz8THvE//ynSBpkGzP7VtRXG2AK0uJ/aCEzGzqpr77j+baS

3wVmRiHcg0B652b2MZSE7QDkpfdfYDPgl2Y2j7s/jTwY7wG2NrPDQB7i07z1k4AgLirvyo+a2
SZmtmp46eHu9wH7A5cAp5vZZoUXtbn7faXHdBthvXWiz0HjyQ1m9kMzWyDsOBzYG0me7GFm81
Sfd/d/ddthWxxSAfwPmA4dYlC8ny8hwnRmpOe+Vvn5bhhXp8DGj5efb2NfmtlKSMblPuBhM/u
NmW0N4EpkeZcrAmpVtEm8vjoMRlEnPwIeQDrhrYSZrQBcCNyM5JXOCbtx95uRDRcCewbRX/bV
B1DI83+ndbunFMV4eSSwNLCFu++KbHgLyb38xiQPgiuZ55ZIXuzaJto8uYi5fzqUxHNGJAd2n
bu/7O5HIk3+J4BjzWz2mPuPBHZBDhCtRjHW9Pq9P6EaNxOJbkDsL6pcLokuRTilVL+vC8wHnO
/uf0fk9h7IyeECM5ujjeu2xHuDnJASiUS/QixaBiOd5H8jkntdd/8YUHnt7W9miwcJvgnwJLA
gIjsSDSIIxXuRPMZfgW8WJFl3RCoAACAASURBVHhPsZGqSLSX4udgugRh4zPIm213M7sUkTJX
IM+1vVFo3grAYchTfNkgSZ8CdgZuQP+b/Rsw4R0RG4jK6+JU4FfATxDp8iMzWwPA3f+MdLIvA
X5uZl/yQDMtnzwEOTMSEW4LINLtNuSZeD6wXkGC74UiTg4xhViW39Nae+vkRdHWU5AUwd5RX2
4e3kKJFHdBHqetRn+30cyWRYeKD6HcF7sjfevTgjzFFVUzBI1BHwTmis8ugTaOv3T3z8Qz37r
x1sxWRjkSHkB5ILZBGuYbVfe4+13AIWjze4qZ/cDMNjazrVDk0VNIeqm1qEjS4ueCSKbmYHe/
1cz2BA5CMmBfRwcavyhI8L8Cn4ifrYaZDYl38wJ0ML4kcE88g8MBXLr0ZyPpl1mj7gV3/0nc1
9po51gLlGP/kKjvN0S4KUKqx8yGmtkcpii3RKK1iOd1GIqI+mLT7UlMGYo9yJeADYBboxAOVJ
eg9dt8wK/r6/JEP4a7Z8mSJUvXFrThma5WNwvwR+Cw+NuKa7sionuHom4Y0pbudW+Wad6XVvt
7MURoPAF8q6gfXPz+LeBvwOxNt/8dbBtc2ogOoBdC3mv/AB5DMhnL1D43O9CDdGkpntM5EVGz
UNO2vYPdpyMNvc/E3+cCrwG/Q9re1X2LR92zSJ6h8bZPSn/G7+sA1wMfKuqWiH69C5ivqD8Qa
dgOatqGSbRzSPwcjnImbIDI0Rmi/qB4Po9FkRijoi9vAfavf08bS3+3ER0OnhLv10xF/UcQ4d
sDHF/Ub4WSft4KnIrm0tvKZ75tBVgEHZj+qOq3qD8JuKaP+z+CJNDeRgcZhwGnAcOr/1nTNvX
R5kFRRtTqZ0EHbjMiwv8ZJHsCily4K/r4HmCWqG/lOidseD/jz4NrAA+HHfsW9VV/zRbXvtC0
DZPTn8XvhyMy5tfAx5tu21S0sVrvjEGH9k+g6ITvAXM23b6pYF+f40Rb3693YWcr57b32ObZY
uw8N8bRftWnA6UAm8bc8BSwX9QNorOXmg45wj0BPEjL95JZpk5JD/BEItG1MCXSuw8RpSWGok
3haBjnFT4kfv8RCuHePL5jiLv/zzuyGq31xOyPMLNBZrahmS1b/e/N7FdmtqG7P4C0Ph8Bdqv
LoZjZ3Mi77Qbk7d9KmNkMwHZmtlI8i2MQeTaHux8ELAOs6O57u/s9xecGoY3/P5DnJsVz+g9g
W3d/ZJobNIkwsy8jonBrd7/UzPZC3pinIe/TgwpP8PvR4dRy7t7avqwQz+AIM/s90kp+3DtJE
oe4PNvXQfkHdio+931gNe8dzdBK1ORdrgV+G+V+4AemZEKHIYJ4B0SS3oVkNkYAB1ff5e3VUO
73NiKv0iWBp939XyYMdfc7kIcwwE5mdhShSAu5+JZJjeQNEn9wOrens9v4cgqbNHgFPc//ZXi3Xo
NeNHMjjKzA8xsI4Cw/QjkOTwIuNfdt3ElUh7mobJJ0+wmS3bnSzI41s/mjrS8CZ7j7y0jy
5O+ITMXd/wP8Jf42lDiSNq5zzOwjSEbqDuAmMxuXiNDxwQ2QIep25nyQxD9VeXF+Dfw3DRv+
BTCO/Jgv0RzyHAU+fUHM/tqk22bWoj1zlAUcTEWefGfjqTPDjGzhRts3ruCdbTpR5rZbma2o5
l9Gtr5fk0pijlytJkdaWY/N7NvmNnMTbfbvYS7P4fGzU8BM8ez3G9iMwYK3P18l8ez3G8iMwYK3P18tE6bA9g2or9
7gEp68A3Uz99Fc+SI5lqbmGZomoHPkiVLliktSH91t/h9MDAsfh+FvNceABYt7jfk7X018Oum
25/FQYkez0ZJ5pZG5NOr8XvlPfRhap7g6JDj5Phc272gZwOuivZ/Cnkq3ga8v497K29UQ7I8t
wCX0yUewzVbNgJ+Fr9/DREUW8Tf30ZeGZcA6zTd1im0b0mUiK0HuBIY3Uc/noE8L2egt8dfV3
gToc3A3Ug+Yx0kB/Jt4HVExo1CxM3ywAnAz5CXe2V/6zxpB6iNvwkb31c9n4j4XRMRpv+HDtl
WLD4zhsLzjRZ7AUbfrNtH3X/DrpuAF1GkTRn9tRySn+pBB3WN29KHbaPDhtuRR/45iNh/EB3K
jIz7hoYt9wKjom72mD+2ouY53qYCrIzm/dOBPVE0VA+Fx178XAlFfD0GfB9FUW0c/dvqKIXC1
jIicb4YY1ZFc/5iKLLrbWCnptv6Lmws57rpgcuAxYu6r4aNZwILN93ed2HnaODPKOri3+Uz25
8KMBIdhD4aY9FbKNnusk23bSrZN7j2d7WXnDlsPqnpNr5L+7pu/zC17USH+q8BZwELRp0V65v
hdEH0aZap9Kw03YAsWaZ1mdACeaBMEP2xoBCmPwBfLjZ+HwX+g3R4FyvuXSgWcoc23e4s40LR
FotN+7+QBMby5fX4uQQdEnwvlAjrv8DSTdswiTbOH8/df1GixA9O5P4ZkQzDjfF/GVr+L9pU4
t1bK9695YG5atdnpROGfyBBEgNLAa8gOaLfEiROGwvwIaQTeBrS8F4k6g0RN5fGwnr9Pj77y7
B9aNN2TKHtm8Zzu2LxHG4QG/2d4+8+n8sJzbVtKwPExu8g0mJ/YNaifhekX/9xRODsNoHPt27
smYithg4xXkTSL/NG/YdRxNgN5XiDInAuiP7+fNPt78OWHyLye6GoG4oOap5AHs/70ln3bBJ2
nIDkwX6FQr/nasqGSbBxcUQyHY88LUGHUlcCP+/j/pURCd6DoqN+C/yckMJr8ztZbxtaF/ya3
rI984c9PcCOTbd5CmysDgaHobXd8sB1xAEMnTXdtogEP4MuI8HpkGbfQ4f4H0KHM8eGTUc13c
apbO9m6HBtTuB9wPpo/XYN8JGm2zeVbJwOReeVdSOAX6BDjmps6grnhWjrl4B54veumcOn0NZ
SlnB54NPxTn6gqD8C7TPPoA8SPMvAKY03IEuWaVmKhdkIlFF+C2DNptuVZYr6cljx+yjklfB0
LNQqj6gvIRL8wdgQ/gD4EyIVW+vNFm3v14uVPuy9MjZ8jwOrFPVWbJg+XNz3H7rI+yQ2g3+Nz
dHfSxvLPgfmRYTpQ8grsxqzWve8Ig/ROxEJ8QYi2O4GPle778PIm3azou5TSFtxyzZvfoFVED
lzf9jZE+2epbhnJaQB/hIiTsdE/eLxuV81bce7sH9/dChVae1uHf+D78bf0yNP/+mbbmvaOJ5
dQ2p/n4MO4K5AHu4/DDu/HtdPBU6P37ti/inGx3r+iCHRjyNr9TuGzfPS2xP3I2hTvNh72d4p
tPFKahFriATfIWz5KzqYqw5vvhnj7fNo7bNU0zZMxLZhwJFIIqMi+Kv5/iQUbbEX8gqft7Bxe
USC/xlJgVXfN7xpmyZia+kVfSTyin4Qzfez1u6tSPCxwO5Nt30ybCw1v29Ga/JXEOn04T7+D9
uGjZfR4kOaor31A4zvAXsXf8+BDvp76GISvD6uAt9ARHA5Zq5FPyHB0T7j92hfUUWCfTCuLQG
8CezadDsn06aD6URZzh11XTGvT4Gt5ZjySzpr9R4U8b11cf0I4OV4nj80rduapR2l8QZkyTKt
SyzM/oQ2vD3Ic+8PxClplvYWpOu9cPH3TNWiJH6/FnlEbUaHzKg0oh9H2pK/oOWh6/Q+yV4uF
mOL04dkRreWmo0VWbFFvJuPAKvRh2QEkp24kELapq2FIvElCq9bH/hkzcY6cTMvCg/egg4R0E

byezja4P4u7JgBaV1XC8+vF/fOg4jg/4v3dB60ub+sre9gtHt5tHE/ho7X9+6xGVqwdu/KiAR
/PRbcZ8X/53Y6pE3XeJkU79ue0afD0WFFD7BXXBuENsanIz37xtudNo5n40gK4gLpmd+CyNHH
6CRLHIYiTk5sus1TaOPJdKJL+oy2QKTxUfFejkec0zLyNJ69kWj9cnE8n+WcuCZa7zyFiNTSs
38+RKLOOq3bPQV2fhTYrla3EjpUfRAlv3wVJWMtD8dXRofJfwQ2bdqOd+rL4vezot8upSOhdT
DFoWrRh+ehSIaZpmV7342N8Z79AcnS7BjvZg9Fouja/2OXeMZbTc7RWc+NijnhYBQ5slHtvtk
RCf4W8IOm2/0u7BwNHIeiSM4v5sTSKWUtRCb+HuWJaLz978LuxVAk4y0oCfRD8fyuEP+DK+mS
NQBaZz9EJ/nxpfRzEjxsOxVFRm0Wc8jWMUf8tZxj6BwOnESXRmdmeZfPStMNyNJ8oW+SqV8Nk
PQmoi5DJ9ZroQRlu8YAef9AmCC6tSCN703Rpv3TUfdkbCAqj8uZUajl8zEBVmHBI5EUw4zF97
WOVIx2le/jmYiYGRvld3SpXnK9L4vfvxYLlWpBvTQFQVzcN5LwMumGBQsdgmUUcEjNlgnZODt
KDjmiqGvlWIQkTB4H1qs9s5vT8bzYvrIhFpxPo83SQ8hbesmm7ZiIfe9HRNlpMa5Uc+P8KKnc
QtFf48Jikbf4pWH7T2OOaa0Hf83eCUl8LBf2XI6I/70Lez8U4+1P6AJyfyDY2Idtq4Rt2xR10
wNzEaHBYeeyyBP1a023+V3YOJ6GN70J7kURyX/shO5pY0FSJj3A5rX6deLZXDHG0327wZ5JsH
fusPcXKA/G6HhHn2J8T/iVYzx+klrkUVtKbX4chUjTVVAk6iB0MPwmsHc1nxT3z0uXkG7R3uH
AB1A0xbJF/V6ImPodfZPgXbH3jGfxgXjfXojn9AZggdp9s6PIonEyWt1U6Gh+P4bWqj3ocH+t
4p5qzb5mXD++6XZPhn0TdbxAeQV+ig7e7g7b/0nIM3bDGIui1l5C65qHkSf4XGXf9aeCIk3/h
hJ7l3vMVVCk0B0Uew4UvZEe4AO0NN6ALA0/AL212uamSFIS9a0f5CfD1pHI0+Rk4BNF/fCofx
T4fdPtzDLRPlw2JvOeWIxcDMxZu6ciwZ8DPk+Q4LV7Wv9cI+3Sx5BUz/zAhsgb6rX6e9pNpbY
wOSNs/CEhL0CHjLkPbWzXRJvAKuHljE20e0psRNEmdyMPjO/QO8FlZeNDiERePu67vkuez/Xi
PZwv/h5eXKv0dHsIzzx0gLUFCv0+gvYnLp0PadJuVPVZ/NwQhcn+CW2ObgI2KPp1VbTJfxb4a
NS3nfyunsvhiGj6bLxzM0X9/jHu3IA29iOANZCn1O1MQIaiTWUg2DgBu2cPG85FkTb1iJORwO
qIGL6dFkdkTKKNpXdi9c5Oh9Z4N3djXyLN3d8gj+i9Y+xdD0WnnBj3nAOc0HRbp5K9MwFfoLZ
2i3njeeCDMZ9U/bs6mjvnb7rt72DXCfGc3kGhSxvX/g95DO9NF3h7T8C+wWj98l8kMzh77fp3
6JDgC0ddeUDVyveR3mvWnZCE1FJoz3w4Qf5SiyJGcihfoeXz/wTs3BQ5i1X9tBVy2LgL+FhxX
zXWLtdFdpZSqBuiCLD16Dsp/bIoQuHO6OeLiVwDbS9ofXN5jCkHoEStF9NPSfCYB3qIQ5raHF
HtVzZrqn1Z2lUab0CWBju/M3GNQR7Rz8QC+xZEHM4Q11u5KJlcWxHZ1oMSPS0b9eWm+JuI2Ph
o0+3NMtG+XB3pKPcAXy37uPh9ZiSH8jQKaRs2rdv5Lm38EPIy2ZZO4qA5YmNxMi1OGDgZNp6B
CO2PTWDhuTQKbe5BJPELdJHOYCyu70ahwIvQR3h92HgvnQOdO+kC7/Zo+3zRJz+mI/FRae+fA
ZyNNoqXMn5od+sX3rF4rnt1LYUiMS5EkQvfQMTL4/RO3LoCIhSfofCYamOh92HN7Yig6EEHiO
eiTdRw4PtR/yfkbXkPOmgcWn5PG8tAsDHaN2gCv1e610vG39Wm0FCUwk2IuGq9nZNqY1E/Gsk
t3YbWBN1g43hrbmABFIXwepQ3gBOJtQ0iqy5ouu1T2I+TRIJGP17V1/fQclIKrUnPQxF99xfv
YBnt9X/oEO5gupAEjzF0O3Sw/wKwQtSXxOqeyNnoXiLithsKOig8Ot7Bb9Su/YAJkODFPd1CD
o+M5/QE4PData2RU8rd9CbBy/e31XbSey1wB/IMfhrtt64hDtGoHRbHs300Whd8YFq3e0psjN
9/gA4uRgF70HEc63ckOIqEeh15dlcSqFV/T4cOT49oup1Z2lEab0CWhjq+s/gajgjvGxEB/FV
0wvsqCtvvqqRP72Dz5kib9W06updWDJBLxiJm46bbmqXP/qs2OlvEAu2a6K/PFX1ZbqoqDbRL
mm77FNi6Vti2VPy9CNKBPI8Oybj5hBbbbS/oNP7vwMeLulmibzchEpEBM8Y4dDAtTpQ4ARu3j
0X1gkXdCkhDemdgmagbiWSYvlaMRa3dRNBbB/McFJa+f3F94Zg/NkFefP+mttGl5Yeq1A4hkK
bpMBSSf1w5LyIPojeoyS+gpHr3oZDMEe91m9+lvSNi3v8dsDY6bPsWOnC7sxhzlgV2Q558a3f
D8zpQbCzmx6F0pL+qdd7cKHroJ30827OgvAStt/Nd2LgTWt92g40leTGc2uFwjCur0Fs6a0G0
jv9W0+2fRBsrx5Oh8fwt0tf/gN4E1JLoEOPgCXxna+eU4hmdBx1a9AAnF9enK37/AyJqZpmWb
ZxCu8Yjz5C80paIaLue0KCvPdcHoYiGriHfYgx5AzlJbRx1I4vrFQl+HC2PRHgHO1eik8fl2K
gbUlzfGu2rbgfWbrq9k2BPX4eJIxH5/Xs6Hu73IG7gXmLNXp8vkIxmr/w2bSnIS/1AtE4t98E
zoAO3feLvA6N/u5YEZyKH1zHm/D3myOpw2JC++2MUCZOzDOzSeAOyNNDpncXYYKSfdy2wdO2e
8wkSI/7utgFyQlqfGyCv0teBT9WurYM8wtZvuv1ZevVLPWR7eDy7KyCvp/EOLejIaYyc2GTZ1
hK2/RvJfyyMdNzOpZPk6xPIW2jFpts6hX34eeSF+SE6ySGfQGRqD3AVhfdtN/RhHzbuGQvPuZ
Ce9L7Iu+tvYeNlRJb52udaa2ttYb0Fkh64DOl6P4qSBP0PODvuGYk8wVrtBd2XjYiYObR2bVY
6HqTjPA9jPqmS8ZakzXLAvE3bNAk2b4Q8u1ah4zmzGdKl/Vb83Sdp2ObndaDYSGejPhwdDN+O
yPt5i3t+hiIVqrmxL/KqtXZOoY3j9Wc32Bi/H4MkW16JMXabCfTZfCgq7GlqESttLPT2wLwce
SdWCRJ3KMbVcnxdOv4XN9EF8jUTe8bQeuBnaH33o6K+JMHnfK/aNhVtHJfbAknSzEInWnEMWh
+8EO9qXyS41evaXBCReFD02y1FfSn9VsmhfLPp9r4LO4fGuHo7Wo9XUUPl2m8rtOY7ten2ToI
9dRLbgH1i7KkI4PPC1kNQ1N5d1bxSfN6QFOXjtIwABw6lIzl4L4pkW7qyG3mu31HcfzAiwX9L
F0VhlP0Rv2+M9L6XpqNYsEjMKQ+iOXO2uP7z6ON5m7YhSztK4w3I0lDH65TwrlhUXlvUl5P53
cCNTbd1Cmwrdc0XpnYaj0jwu9Bp/nboEGBd5EFzW7csyAZCKRcvyON03tr1lehogm8YdR9ExO
MO9e9pW5lQu2Kx/SRwK/L8Pj/+B4ZCaU9DG4vZmrZhSmyMd/B1ROJfhTb5P0OE+JbRn633Lqn
bSG9PmS3DxptjvHkJeYWPRLI8PRRJotpe6E3snhbtnwttfLdBCVvPp9j8IZ3lJyi84Ntc6GzK
h8W79+DEFsxIWms95BG1el//q24oKGT0ueLvraN/vxt/Tw98kTiA68bS32xE0h6D6RCFI1Hyu
YNRsqux6LD/AHQAtyDavO/fdNvTxne0+1zkrXY48ly/GREzP67dtw5wFvK2XbqJtk6hfaPQGu
1qFA21IYrMeAU4snbf6dHHN9Ed8jUlQfNZOlJZo+msE+ZB651XKBIH0odMWhsLveUzL0WE09+
R93pFmE5HhwS/mvBoZwLSN20qTNiBagwdEvxCCgnN4p4daXGESc2eCe0/hiMS/CGUPHGRPvpu

7Ql9vi0FRY3cT+cApuqvXeg4LJyAnDcWK/7uQVFhixbfNQQ5tfTQsvVszBH/RXvC69C+6t9on
lw2xp6XKaR7UM6T55EDZFeseWo2n4OiMd5Anvs/pOO5vzTac/WgNcKjaD/dNXNklve+NN6ALA
11vBaWx6GkK8+XAwMd76idY1D9UFsXKn3YVREYYxCB8XQMgj+id0byyhP87bDxRBSSV4XMtG5
iRxvBrtAHnlr2xs/R0Td/RN56v6Dw3kckeOUJfhYKZftL2/9X9N4orYDC0EcXz2AVkfAisEbU
LYPIxxfpskSYwGH0Jkd3QSGIPwM2L+qXQIvujzXd5sm0bwTyqCht/BpwKiLfKo8MQ6Tpfd3Sh
/Te+CyNPL0/RU3ag956prMgLfAbaGni0mK+KBPnDY538UqUVX5Q7TPjPBRjbrw++r2roqRqNu
0QY80sSFqpJIYHxfXT6QLPxIFgIyIovk14kMa88ToKb66ez/WR59crdMjGP8az2mpJnoFi4wT
s3hyR+J8gvILRGqcHJRAeWty7OJLPar3nd83G70Q/LU6HlPo64+d1GYGkJw6kO+RrynnyV2gd
8xDaY10f80q1vqtI8BeA05pu+2TYWCaWvR8Rhfsj/esHUeTwZnFP5Qn+DFrvzNB0+yfBvtKBa
n7kIFX26/TIU/h5RIKP0xju63vaWgo7R8X7eBzSiB6XHyue1zoJPrj2Pa3bKxdt2ySex4coJI
XQem86YCF0cPPF4r1cFXlH/ws4vfjMUHSgtci0tGEybN0VeThfgKLdNo937lFEFl+F5sc5i88
cCWzXdNunsF/vQVKhiwL7xdxxOoVUJopU2A1F+s3VdLuztKs03oAs06ij+9bBmhURMz3AUYyf
qGyvGFDHC9Nvc4mJ+wrkUbIDWjy/jTyFVy3u2xCdlj4DfLqob13CRLQRuCoWk13hJfIu7a02u
KPRIvuamMw2QSe61wEbFfcvg06A747Jflx4ZtO2TIKtZ8ZiqwdtfLcnEiABn0Oew4+hw5y7Yz
G3VNPtnkwbZ4v2/4UifDCe6/IgYA7gFJSYZvam2z2ZNs4V4+UDwPa1a6V27aIo58IltJg0pW/
P/Z8iUvtmYExlW31+ibH1F/HsLvlet/Vd2Dhj7e8qJ8ZVFMnW+rBvBrRpui3eyWq8aW1/TqhP
o36VmCMrCZuKGDY6JP9P6/+HNpYBYuMoFL3Wg7yF/x7P7Nx9PKtzI/ml39EJk/5M0zakjX0/r
8jD9D46cmeLIrLtHDo69UtO6PNtLH3YeA69I0+3jndzr/h7DJFAcWLf09aCDr0fJw7x0SFNDy
L916ZDts2NnDYepwvWO3TWMUNQUtbLgIWK64ujw6dXgA9H3eh4jy9te//RIbPHABchz/ZX0Jp
ubTrSSjMg79rnkLdtq+2aiL1j0KHFk1FeQPur7Yt+/lTY/yCwRNNtnkz7hgGbxvv1KDBzrZ8/
g5wAK+eUQejA7XxEdre2X5HMybq1ut2iDy9FkVIzIkeq25Gj359QlFRfUlqtXff0MX98hdpaD
R2WVyR4Vz2nWZopjTcgyzTo5M7mfDAKIS3DtWahox91DAqXGYQS7dyFSOPWDoyFHb3C6oBfE0
nmou6zYePv6U2Cfy4mh6cJndo22otkL/6K5AQ+x8AgwYehje+VdELYTkNk8XMxmW9QPgPIO6P
yUGkl+U1vwvc7aJG9ZSxULkTkzN7FYu2DSGJiH7QI77YDqWrTtECMKQ9TI4jj+meBXyLv9taS
phPrU+QxdE9fNqJDgB0R+X0XLSZNY564lUJjPt6vc2Mc/RfwoYl8di9EDrd2IYo82R8nQl+rf
kSERE/MCeMls4r5ZS/kTXMeXXLYVrRzOmA15D07Dx1S7bth941okzQY+Bg6ELiD7tDe7dc2xh
g6TgYCeez1oIO3+Yr76jrK1c9N0XrnvPgftc7OgWBj3Yb4fYn4eTTwYPy+IDpEPIcOIb4t8ty
buen2T6KN1dw4MzGfAGcD18TvW0T/VuT3EOTRdxCR6LSbCvK+vJtI8I3WeGMROfUIOtxYh06k
7VzAHE23ezLsG46iuipZmnoU2OLI0eFKOhEM5Z6zlaQinX1D5XRzNSJIF0ORJw8iJ5zq4H965
FzVAxzRdPsnw85Spu8g5Fy0VPy9AnASOoyqSPBhKBrlReDcpts/GXZW79dwdMD2HCLyZyrumR
dJg5wY49MKyLnqiOKe1j2vwO7x3D0PrFO7tkvYdBnBgaA162cocvHQ4nmxZk+5X94WOTWeCOw
bdWXuhOr/chot3ntkaUdpvAFZ3uMO7u1Jey4KVzsH2KK4Z2YU0tWDFts3IG+oa+h4KrSOpCna
P06HDVgRhTDdBby/dn39sPF39CbB10cejWNpsewCItBuQcRMvyTBa5PdAsjja534+2zkpfCBW
JD1oI3uxn18T2uf16KNq6HNbp0oPScWoPvQBV5BE+vD+NuKd3ABOhI1Zajz+tG3N9AFsiB0Nv
WlhEZp492IBP9K8ZkvoUOsM2g5aQp8HHlYjKnVD0PkSw9wBPC+CXx+OC0naBAJsXv8XhJRw+h
47B1IHL7VPjsGWJ7O/Nq6TVKtvaU02N10Ik6eR0RhFXGyd4w9D8Y88ye0DugG7d1+bSPKZ/I4
8Pmi7jiUVHcsiiTqc01Qe76/jzbIrZtbBoKNE2j7FSjKawbgq/F8bo+8+c6jQ37Pjtbxp9FF5
DCK9Hoa6c0ORVq6Y2MOeRPYu7h3cbQOOLDpdk+BnYNQpMm+8ft2SKd2y7j++ejba1CUVOuiTS
fBxoXRHuqfwE1FfUmsnojI/lZKn03EtiEocu0qOk4356O16Z0xl2xFJ+HejPG+tnLOmIido9D
+4iRg59q12VFS3bF0CNQhyCmuK+yksz6fESU/vCrmlR4KORS0VtgLHXCMRST/XbRYPjPGz5PD
lptjXF2vds8uaP1zCbBSH9/R+v1xH22+AEVj/Cdsf6x4D4cV9+0W109scz9mab403oAs06CT5
QVzRwzsZ6HNwetqegAAIABJREFUxLPAd4p7ZqEjh3IxvT3/WjuI0HvTeyci8F+KgXKluDaEDl
GxPgp5upfeYaQbowQuC03L9r+DbSPRhm/+om42Otrm/YoELxYtMxQLr5URmbYd8n5fvbjvN7F
geZgiAV3bCtr8jarVHYo2t38DVom60kvmnHhO92ICJGPbC1pgrxG/1wniB5HO3nbF/avSJYRF
0a+/mIiNjyCvxZIEH6cpTcs3E8V7djS95YaGxuLyTeTdNh5B3E0l5scr+rDxpL5srPcbLd9Ml
M8b8gr6AyJfFka5MR5HHm8zxn2rIE+bbyONxa7R3u3nNs4IfDF+H4YItplR1MkuaBN/FuNr0V
ZrpOp/tBaSfVt1WrU9bRzPzvKgfx1EZHwy7B1EJ7H31XTI73mQPNg/KHRO21qKd8rQ+u0yOon
mRhc2nlz03/Lxv7ipPs62qaC5f+sYP7ZFXtyVDTPFMzsSORIdTMcTej5Epvag/VhrDzGQ5/PB
iDw8jt55oj6MDmZ66OOgAuV7eZAu8Wwv+u59SEpx8/j7TLTvWCLev0fQfmNLxpdPa+3cUWvnI
DqHwGOBL0R9uf9YCe2jd+vj8619L2vtHIGkFH+H9LAXRFKvz6N9V3XAMWvY+90Yp7phLTBnzG
+Xxvv5DOPLoewSfXgJsFzTbZ4CG8s5cjPk1PAJ5LV/cNh2PR3HhvL53YkucKTK0mxpvAFZ3qO
O7a2NtEIsNqskFh9Gp2mv0tvzYjYUEvU2SmrS2sVZtHfcRIV02K6Lie5I5P3zWDUI0psE3xQt
sOvJzVqVCTna+SqSEpin1k/9igQvFqAjUajaufQOVTsZ+EPx95CY2M+LZ7mVi7Jo533AkbX6j
6BwvB7goKK+nMTPjOu715/VtpU+3qWPokOom+gcRJUE8eIoS/ndhBdum0v04yh6exl+NsaZG4
CV+7BxWeStcCewx8T+X20twFLIa/0f9M6TMAQl8HoTHdLM0lQbJ8OWPkM+0aHaIzFfTMjG73S
DjROyGR0iros2TGvQmTuHIwLncXTo1mfiwLaOrwPBRkQWlokPh6NN7y/oeK3PQm+CuBqD5gB+
QhwsogOCsxEJ0JrDxoFg4wTs3g15QV9Y2GPo4P/SsPW4+D9ciYiOrsn/gebMn6CEkCcV9ZWn9
GVI8u1cdGh1D1rbtjYaAznb/DHmjGeAN+L3fWvP8PvRWnb/ou6jSJ5xUWDepm2ZiI2rosP7m1

HU6ctofzWmuGcxtPZ+g1jDxnu8GCIef0NLZRaQl/OC8QwOprM3nA6t24Yip6i/ocO06vov6OQ
Y+FTTdkyCnR9Ae40t0KFMNSfOA/w4xpcrivsr2ZBBaH955LRu81S0ffN4blct7B6Gcre8gDzB
+4xSbOm4U61zBqO16bFoj1jlonmWvknw52LuWHFat3kq2f0ltMc4lt5ruj1jjLqejmND13MhW
aZdabwBWd6DTu2t+T0IJQ48sXbPEii061UiGVTUz4RO799ARPKYadXuKbR1OkSmnQ18tKjfJi
a4h+iDBC/uG9zWRVq0b3u0uL6TfkqCF8/roJjsLqOmL4w8Tp+id2Ko62l5MijkhfBpOslzSr2
yxRAJ/hjw5aK+JMF/DizatB3vYN9OyMvi2/TOtL5tvH83OyGIyyRKfwReQ9EprQ2VjWftCJSE
bReKRXOMM3+ZgI0fRKTbfxDx1tpxprBnNsZPhrw+0kz+B4XeIB2C+DXgANoveTIILZwXAuauX
VsPHdb0ZeNJaKN4MBFy2eZSGz8s5rjfx/P7KJ3NQiVvNhyRVM8CH2i6/WljLzsXRR6Ha9E72u
3iGFuOp0MWzhrj02tx/SsxLj1LZ+M4EyJUW+MRNhBsnIDdC9Ah0y6tXTNEoB6GnFeui/Gn9Z7
fNTuWLGz8VR/Xp0eHi6ch7/av02IPTESMXhP9sQQ6fFkOEaU99D6wmRetcS5BcncLIEeOG2nx
virseSHeoTnjWVwt7KvrDS+JSPAeRMJVCaRbm+MEkd73xpjxRvTR7nQ8gqt2H4LWdjMWnz0BO
Ycd2sbns2bnSmjf+M/onz/TWwO6OjjsAU7t4xl4Bvhm03a8C/u/hQ7Xqj1jtRYYhsjUHuScNF
vTbZ0EW8aLwkcRbq8gB8flkBxoX57glSTIBtOirVPZ7hWL+eNH9f8JHRL8alq+/8jSvtJ4A7J
M5Q7tkC9jkAfJLSiE6xpCE7u4dwnkQfsScGhRPz0KG36RFoe3I0LjKuSh9wC15IDAlxEB9yCd
0Mte2r1tLfT2NN0ReWH2ZxJ8FErCegFwch/XN0Gb4fvQJuJ+RJy2jvSeiI0/jkVK6UWzZPGMf
rmob31fxhhzU/TDtShMbQy9o0++gsiNm+mtu78g2ix+hPZ7Qj2GCP6fx4Js6ERsXK2oXxltMO
Zr87iDDhF3Rhv1f6KN4Yn0lgTZIOzriyCuvC1b6yENLIIOMR5C5NmbjJ8LY2I2nodIj9b1X83
ORWOc2aFW/xXkCdQDbFXUV5vCJeLa2k3bkDaOZ+tDaP5fnQ45MyNKGPyP+F+UBPG28T7+E20M
q2vjDh+btmug2RhtGm+tgua/v6EDtvX6Gl+oyb20uUzAxuXRodSLTMBrtm53X9/ThoIklR4GN
qvVzxU29qAcH9WYsznwX+Ro9GSMT61N8B3v1iVoTn9fUb8AIlCXj3F0huJd+zBypnoO7TM/UH
yuVe8himh7BRHZmyH5oTui3y6q2o7m/EPR3vmj6BBgYbSf/mJb7SvatRJyvDgBRSquhvZMV8b
16qDxA3FPD9pLfwX4JloH3d1W+yZgs9V+rhp2fam4p3ovFy/e11833fZ3sOsgtL/6CLWIJrSG
vRo5AHwK8QF9keBdEzVUa/dwtPd/CuVqWax2fShyvHoD8V2tXp9naVdpvAFZpmJndgb+YehU+
zYUWnh1DPQ7M7524hIo9PAqepOuY+gC7WGkm3gnIjQ2rk/YyKP4fpQQYr6m2zuZtk0OCf44Cn
PruqQ6YUe1WOkBfljUlzpg28dz+kd6h0C3crNUs28Yyl5dJSYpSfCl6JDgX2yqjZNpz2jkHXM
176CzipIEPRgLmA0RMXwKOrRqlexQrd0rok3E8byD513Y+ED047Zoc3UzcHlxT+ue0xjnr0Yb
u0uR19MZiCR+lsi0HvdOiCAeTIu1PuN5+wvS+T42+upItBF+AdhvAjaWciiD6Gz4W7nIRpveK
hrhpHpbgY3C5nuAT9Rs+3yMTa0Okx0INtbsHYQihW6NZ3INOmHqMwKnMz5BPBh5D69KiwnhgW
Rjzd560usVYqy9jd6HxK2bLybRvpHA4f3RRuRp2UOReJ3OOvQktF54HGksV8/lGsAeyFtxgaZ
teAf75kU6yevH35UNH0ck/mNh/wPArnQO9pdCh8SvAHsV39ca729EYh+GJOveX9QPorNHvo5w
pELE/gvowOMPyPnmzraPM8i55EHkfDFDUb8XcGcf988Z976O9tHHxfM7Iq63+h2dUPvQoc110
WefLO+nE2n8qTbbh6Jrq33xo8Bv0SFFNeasjw5Qlyn+vgXNl+v18X2teR8ntW0xn2wW489vKf
iPuD4sxqLW5G/L0h2l8QZkmUod2VtaYBV0Ij9PTPrzIE+asYhEHFH77PzF57tqgIzBb/WYBO6
mj5NORB6f1eaJbmI21uyYEAleEaitDa18J5uBj6HEj/+qLVhKXcURKMS5Ouxp5WKUvj25RgNf
QLp0lzE+CX4f2iRuMa3aOYW2DUWbnWspvLdr/bQuvQnErVHob0/Y+GRf72pbCvKMuQVFG5Sbi
FJ6YRcKTwt0APW7sPEptPiuCJvWkaYo0ucRJB0xTicxrq0S48x/gO8V9esh8vFRYMOmbZgEGz
+KNnbHM76s0orRx/8B9qnZeBMiMj5b+0wr50dEML2KPPcXL+oH0/sgdUvkOXtvvJOzo8RCN0d
ppX0DxcaavdXYMRR5Xj6FiP3V6XiylV7S46RCat/TWnsHgo21dm4a88PZtfqVkAftrbQ0cedk
2PiNsPHM/mAjvQ/YFkJRB2cAc5bXgR8gTd5rkATIeM9pWwu9pfnmq11bDnmxnxNj64rxjj5F7
0PGyhP8RfpIjNmGgry8b6rbDSwdc8uryGlslqhfFuWXugqtBVvtdIPI/K2jzStUdfFzL3QAdU
Zc346QjUAk+Akomvinxff1mS+j6VLvB7Qv3AJJ2exER3Ly0zFv3IXkCmdGhzk31uxsa3+uhta
odyEHjkNQ7qTfEQlKUQTDecVn1on7xxIcUNN2TMpzW/y+BnJm/HxtXNoibB+PBM+SZUpK4w3I
MhU7U6HslyNt5N/Xrs2OJAfGIp298cIqafEmopjwhsagvnA1yUX9Gihc7Q76JsHHJZBo2paJ2
FgSUPMheYyZavfsRN8k+PtosZTEhOzs49onURLW63gHT6E2Pq+EHm3x95iqPn5Oj7wP+iLBl0
ML1LZ7CS2GDlu2KewqFzD7oA3wtcDqRf28yONiI2pyRW0rwJqI5F27qCv79cKw8fc1G2dHRM4
qtNgzER3GPBzP4AfLfix+Xyqex+eAzxef/TTyALsfSRe1coGNNupvoFDmUUX94JqNdyNSY4Pi
nnXQ4cBvm7ZjEux8X7xrJ9bniwncvzXy1OtBc+ZFKGFZ5XXbujlyINhYa3+1sZ8eaV1fSidk+
2FEEFfkcUUQP46SDbbatoFkYx82z4KkmJ5hwiT4jcDHmm7ru7TxMHTQ3ZeNz6IDxo833dZJsK
WS5luqqDsontEjiMgwYBm0t9oU7U/e6JY+RCTvucDStfpq/XIocCq991uzIzLqkNpnFkcJ956
gZZJoaG3zY+QstXRRb9Huv6B13dNl31GTI6SF67la++ZjfAmM5dFB/+3R1/cgsv8AOp7eH0Qk
+CvACU3bMRH71kD5EKo8H2PCnr+Fjf+LeeLzcX1ttHboQXvLVxBP0FrnlJq9q8SccD+KXpwLR
Zs8GnWXokOncr+8IcWavY0Fzft7UsjsosOZx1AkwptoDb4RnYOqigS/AJi/aRuydHdpvAFZpm
Jnihy+KSaAP1LzQIhFy2noNH93Wi6XERP5uMSVMdFdHoPiC8XgWE2Ea8TEdzt96Oy1eaKjN7n
2I/6fvfuOk6sqHz/+mfRkUwiEIr2LCBJAQRTEBmJDRATFAmIDCyogghQFe0Oxd0VEBfVr+6pJC
BQEVewMU2w8RpEvoSSAh2fn98ZybOXt3drMJ2dw7dz1v+Ptz7v1+u+snNnZnOePTP3nvvvvcU2K+vTuJC
6IXMnTIXpEE/3WvnQTo3MiYnk7mbyN6Jjw8O8k9LTVULqVHegrRpVFMXDhcSCQZX0GnV8IcRk
6C98Lc30cQU2TMTY/zHlKnpIbLe4gLwkvIEsS9shE9Lf6TPc5j/Bax2N6rUqyXAPuO8HvqeJN
mUjpGLmPookj5TYwiQbwncaF0bul37EeNb7gRo2LuIpIte4xQj/lckfcxfJGdveiBRBsx5Pk/

pHlps7g2IC4YziGSOC/M3vNc4qLpNwwdzl/LNkE/xNgl5mlEL/aLiYv4LYnhz1cQ7ZxyL+nvE
r2janfM6ccYGT4dX5FwmUv0Fv4v8LXSa/Yk2j4XUtPel6sRYzkJXixs9umqyz+G+J6YyvoNhi
64/j46I71+m+rs3PTcuum488yqyz/GGJ+VYvk62Ry7DD1Xlut5Zvp+vq3L73sYqXd83bZ0jBk
k1nPJY305MYp2I+Lm/hdGeH9tryFHKO8kYl73xSnm/DryB6kO8xGOGxPXn4PAB6suf5d4isVY
P5weTyWSw5cSN9c2JhZTvphIdD83vW6rdNx5NbFeVm0X2h0h7scQo9euAw5J++YSU9f8IX12H
zPCe2t3riRyOTcRnTVnpH2fJtp4zyVuKB6cnr+TNCo61fehdNZa6In6c6vnVnkB3NZQRXYSi1
NSQ2aQaIAOlF63AXGX+9I6n8yJC59B0hB0Yh6oq9IB8aXEPLtfTq85IcXdSu+7Jp0otq06jpX
EOIFSgoUYZngtsYDn9sS8ybcSd0o3zF53DLF46SWpkVPbuszjTf/OSnH9K8W2kBimdhKdHtNP
IxJ0PwGeWHXZVxJX0QjLp1E4P8V2PtGb7TZiCGVxs6ZIgt9G3LQqVirvhXp8AdHDqbwgyaZEL
8ui58W+dHrzb1F1uVcSU3kBnZOJi9jycOCHERcS89PjYu76/6OHViEHTiNuIv4PsPNK/iYfJh
LEG9f988nwmzH/JW5YdF14LIvx60Tje8XiXtlrapcEzz+vRG/1QdI8kGn/rsTQ2aXEDe/7iAu
JfDqbF6Rj1CXAPlXH1I8xjhRz+vmpxDk+n2ZgMjFS6Eqix+Lj6PRqn0lvTGXX+BhL8T4m+7lo
p88lbhIvYPjNxUeykjUn6rQRvaSfPkqMdwBfKr3n4XU8ro4Q37PTseW7DE2CP4e4Uf454JXZ/
n2JhNT8tV3WBxHjwUSb7jtk7YGR6oiYrvBKSlOE9cJGZ17l3xJt8s+kx8WUEm8netU2JrmWPp
ND5vQmEsWDRGK4lZ1vNyXmgX9oFWUdJYZ90rn+LDpJ08cSN6GeUXrtvHTOv44RpgXtleNPVt6
9iCT4tcDLs/0PpcYL63aJYzbRe/1COovObk906juaoe2DecQo2+tJPcWJXM/BwA5Vx+LW21vl
BXBbjUrLplmglLhJP08lFtq7lehhW06Cz6XGC3oRFzyLicU4isTgKcS0Hw/NXvfidAJ/WbZvI
tE78Zt1PsERiZavM7RBfQIxD/Sj0+MTU6P0EqKn6Ylki80RPaVqneTvEvcUIiH6c2L6gY2IXl
+XEReDb6LTE3z/VL8frbrcK4lpB+LGzO3A64lebRcydEjaJ4kk/5cYmgR/RWrQbFZ1HKPEN6H
074GpXl6U708/r1c6Fn0+1XWt58Mk9YLJYnxJivFZ6XE+QqO8hsJ3gXOqjmGs9Zg9fiORMOya
IM7+Fq8heoFvUXUMY63H1YjxA8TF0kqn16jDxtBeW3PTeeNvxMX9x4hhojcTF7KbEsnivxE3H
TfN3ns40RPnT8DeVcfVbzGm8u0DfKXL/ucTN4EfmR7n08AdnY5PvyfaO/nxqXaJ4X6IcYS4X0
N2rizFuB5xg3wQ+HzVZV3N+Frp2DnI0NEXRYzziGuRQeCLXd5f5zZ6/nl7CtEG/yLZTbgu75l
HzAN+OdnQ/rpuZEleYu2L5cQo4TwJ3iJrAxI3L35BTHVS6+/hSJ8vIoH2A+JGxRXA4dlz5wPf
qLrsayjOru3uVI8npXPi3Gx/LacKJW4KLgbek5ePTm/gx6bH+ed5fyJh/oo8tl7eiF7uvyQ6+
R2d7R82FWUdN+Jm6Z/TsWOd7LiyR6rHoqd3eT2ixWQdG9zc1sRWeQHcVqGySnOqEb1hPkks0P
FH4DjSXHVEEu57RA+4YUnw9JraHSzTgXApMX1EPjXEV4he6/lcUIPAm9LjdUnJYYYm5Gp1Ik9
lmp4ayL8jmyMaOBU4PT1+HZH8PiydNP6PGNL/emo+f/IY6vc60tC0bP9UolH9H4be5NiTHuiJ
QSyi821i2N25xE2L9bLnpxALeJWT4LMpJe3qtKXjyJ/y+iKSEz8jEvcbFfvSv3nye2viRtTb6
1yHxMJWixk6TcKc9P28Gli/W4zpO/so4obO0WuzzGuiHtP+rgni0jH0E+l4VbvzxcrqcSUxFv
U4lbhhdcHaKuuajDN9Hw9L9TuYts+T9apNr3tOem7P0v6jiOnEtqg6tn6KMZVrEjHa5CNdnns
E0QY4Ln99+nd74iL4PuA7VcfR7zFmZS/fZNwznQPLSfBiWpeHEW2CnkmCjxDj10eJ8clEu26Q
LlNm1HFjaALmTOIaamGK4WuUEsRZnBcQHSFq3yOTThJxFvBVYgqCpSnGC8gWGk6vW5+4KfULh
s6hXLtrrFSu4jgylUgcHkDcKM2nVJpGZ8HLVjoe/Qk4qeryP4g4n0LM5z2p9HzeNt+B6HT0aW
qeGCamPbmv+FyWnptPTLX4xmxf8bnenGhDvKbqGNbw3yNPgr+s6vKsQrlnEdMLDVJaWwfYhuj
QcEZ6nHfynEnq6FB1DG7N2iovgNsYKyqGMF1C6n2QDgr/IOa3PIeYo+7m1DB5fHrNVKJ34k3E
MPZhC1/WaUsH9kXEAkFFz++isfJj4ML082HpIHpSejyRmJPvC3WPMZX38BTjlunxsenfHYkpa
h5K9GJ7TRb/6+hc+L+emiejRom9mFPxmdm+ooG2Tar/49PjCeXX1G0rlXF3Isl2G0NXmi+Gb0
8mkuD/Tq+rbeI7K/tO6ZjyQFFn6fv2AiKpeBUpQVx637pE4/pqaj5PPdG75JfExd+BxeeNSJr
dk2J8SOk9E4gk3S+I+QZreRE4Wj1mz43YSzodky4iEgArhsnWcetWjyuLMcW0LzEc+hVVx7Aa
cRZThE0iRtM8Dti9y3smEqOo/knqHV06ds0e73Ib44ixFsO5B4B3Z/tnpWPoEobf1Hkm0cFhe
3qgLdAnMeZJ092yn3chppoakiBOzz0vnT/OBLarOoYxxFi01aYxdP2IrjGm4+sJxHSFr6Dm58
ku8V5AJJpOSOX/WDomfYssQUz0/H4Jcb21U9XlXoX4ZhBJqYvTZ/EZxAKmS7rEeBixNtGXKCV
X67bRuSkxi5gj+Y702bw5fRYHSq+bQSTILyNGnNQyrlWM89wszuJmxUxiNM6v8jipabsulXUJ
sejj6cRooQuy59dJ9bkUOLj03r3Td/c5a7PMq1uHq/ieRxOdkG6gBxL8REeva9L3653EdUhej
9OIdsBgl3rcNh13iimKavlZdeu9rfICuI2hkuIksJxIYk8hki9fJBZB3IrOnbJz6KxEXuybSl
xI/m+dDxx05vf6M5Ec/hZD71i/guiB8dn0tzgpO/nPJxpw76ii7KsR62FET+FXExcNy4DNs+c
PSg2Z3bN9ryF6vL2d0tzLvbQRSeJikcTy/OebEtOgnFJF2VYjlmEXc0QP92+nE/mJ2f7iRsZk
4kbNVaT5z+q+EcmoHzF0SpDpxJQ8dxM3aw4mJbqJi6hziRELu1Rd/jHGuDuxwO4gnSR4EeOdd
Obl34lIwr2CuIi4nJr3hBqhHkdLEO+c9s0m5jf9GzVe8HKUeizPC9ktxi2Jmxj/V/f6GyXOA0
d4TZ6Q245I8p/H0PPqsCnU6rL1Q4ylWF6aYv1Ktm9vom2zNH1+H0Osj/ErotNDLYes91uMpc/
hx4jkUj5Pa54gfhnRDtiYGMH5Pnpj8eu8x/BP0nft0Oz5+VmMrwQ2JHrUXsLQHv61rcdSvM8k
2qMH0hmqP5kYfbqE6PWe9wSfQ+q40ysbsZD3TQy9YdPKYryAoTeMd6j79zGrq4lEe+cnRPt0J
zpTEf6RTnJ4EnFT9ZvpONQr7blVjXMG0fa5ghjxViS/axkn0V4dBD6UHq8LvJnhSfAnEMngB9
L5Yx9iYddfp3NILeNLZf8I2foBq/jexxCJ4SOrjmMl5RwgRgBdko6RI9Xj41I93kfkRrYhrqc
/T+SEtqoqBrdmbpUXwG0lFRSLPCwlGslFL5qZRG++M7PXHUokht+YHs+gs8DA5OxkWbuLwHTC
eoDoebARsbDMgnSyLsq9MzEFylLS/GzEzYBd0onuMmp+NzuLdwYxL/KdxHDJYv7L4gLjsPTci
4ibHesSQxQ/VHXZVyHGERsdwMeJoWmHMHTOtt2Ju8RHVF3+VYmPSIw+L3u8C9ET6F7SXeu0v0

iCTyJb0LSuW1He9POT0/dvEDgg7ZueYv9N2n9Xaqhckz7fte8JVYrxQKIXTTnG52exP5COQVe
RrUJOjXsMjVKP+5VelyeIH02swXAvvTGce6R6HCnGbxJDhX9K3HTtlYveUeOkdJ5P55q9ifPj
FdnntbY9avshxhHiXp+YKqTcO+qxRELjPuKi8Q7iwn7YtFN135oYI0NvthQ9hg8Htim9bhfix
vBges3fifZf10WI67Rl37WZRI/hi4jRfDNKr9uZ6JwzmOrwv+n7W9vz4ygxH0okgffo8tzpKc
bPkCWPe22jc4N/i9L+CcB7U4yfJV2j5M9XXfYR4smnNTuAWMzysdk5YwYxjcsNRIex4lwxB9g
te12tP6+rGOcXszj3IW5OFdeatY2TGPV8cn6MIUZanJLOEd/I9j+BGJlwf/rMXk/cEKhtu44Y
XTkIPOFB/I6Nqo5jDGV8afqubZztG6ken0TcWFxGtAVuTOfKnllQ2K13tsoL4DZK5cT8souIX
hUzs/0bpRPbaenx8xg6Jch0Ysj6c8gWwahjo4WYq+w/wAfpTHsyixgms4DoTVuc1Ivkzd3pb3
Ih0bj+TZ1PdKV4i4bLz9NB/iZKQ5hS/H9MB/9L0on8DuBhVZd/jDEWjauB9Dk8J9Xn09L+jYi
efYuJnuAHEUnG3xG9iupeh3ny+8tED9m3kCW1icZ0MSf4sCR43TeGTh1wVorl73Sm4Sl6gk9K
n9cXE8OE30I0sodNi1K3LfucrkMsiHQxnTnqhiyAmeJ8OnFR/HximqLiu1zni4iV1WO3XtK3p
ePNvfTAxf0Y6rFbjLen567Izh21rcdViZO0aBnRg+ZH6VxyYS/E2Q8x5nF22b8+nQvDr2f7B4
jF5w4m2kG9kMBofIyluE4jLtb3yD6H04me3sXN73npHHIe0fZ5aFXlXY34JhAJ/MuI0XpFu3w
TIplTrGczi7i5+Daih3Ev3CTuNprvECIB/tRy+YkbbovSMek8eqBd1+3vTyRD76eziGDetj2Y
TkLxPVWXf1XiTOeN/5e+j7PT/uI7OYO4cXEjXTqiUMPr5DUQ57BE6UjH5zptpc9j0d5eLzt/f
DN7fjZxA+5AYl2CWt/MSOe6QeBxD+J31H50G9FWy29irKweNyBu5ryO6Mm/6dosr1v/bJUXwG
2EiunMfbWEmNNy7+zAMUAMvf82nSGlJ2cH/D2J3jMvqTqOMcb6eDqLWxYxzqbTE/w7WWyPAF5
7G2HqAAAgAElEQVROTOnySWJqkF67UNqQ6O3+DGKo7LXAaxnak2hdomF9SYq/J6Y9YWhPoX8S
d+L/TCSCbyCm8WkRSfCP0mlg/4tIZPTEjYxUxi+ncu8PrNvl+T1S3S0ATq66vKsZ4znEDarDi
eGvL02f2RGnJeiljZh77g/ETabHp8eHE9NGrTRGeudiabR6LM8JfgrRk7/2PfhXoR7LSfDTiJ
uptU/OrE6cxMXxc4nho8f30jmy6TFm5ZwOvJBoz2xNpw20AZ0LwxEXZq3zObIfYiyVs0W0176
Y7ZtP9JT+O7Gw3hPy11dd5tWIcToxRP2tWcxHEh0A7kj/dk3m9FA97pX9PJkYXXoVpfUDiJ6K
XyWSbTtUXe5ViG8AeHr2uOhscwXpBkb23IuJKRr2qPsxtUucr6Vzk/vgbH9xI2r39NwTqy6rc
XaNq7zQbqv0OE+efmOsv6dOGzFl2zKytRRW8f21P4es7O8/1np0cxuPrfICuHWplJgL6T6iV/
SeRLL7X2l/ccH+LDqrkp+avfdhxJymP657o3OkgyOdZHfXJPgI76ltrN3KlsX4ELIkeOk1KxY
cqjqGMca5Yug58Kb0Gdw27duAGEp5G3B29p75xM2dXan5HftSrE8iel0cQOdifwOiN/sxdKYr
eiRxE+M64qZG7RstWYzbAbcCxzK0J8bexAiGYdMS9NpGzKN3d6q3PMbHEjcRB4En93iMY67H7
Lm5VZd7vOoxe672PfhXM87iOzmZLHFT53NkH8Y4k5i/878pnhuJeTHnpOeLBPEDZPNl99LW5B
jL5wIiGXwpcZNmf6Ldeh8xKuEtRBL1Fz12/i+v0TKRSJReRLRxzqezNtGRwD+AH1Zd7gcR7yv
T5/SF2b6nEzeDLydG9a0LbEGs4/ITeqRtnsXzrhTjC9LjFvBson1wJXGNuS2R9P4l8IXsvbU8
TzJ0lFt+vjic6KX/W7IbM8T1yfOJmzaPXFvlNM4xxznSyKHy8ahIni4FvlZ1uccY20D28/z0X
dxrNX5P3lnuMNIIjjptq1GPy4CvdovRzW08tsoL4FaqkFjxfjA1KqemfbvQSYLvkxots4ih3P
cC3yMWDTiD6Dn1R3qgJy1De7Q/prSvWxI8nxO89nNDlv/+RG+1I8h6YKT9m9AlCV7nuhsl3gE
i0fZt4Mz0WS3qdC7RW+8GYP8R3t8TScbU6FxIzEM/hejNfz0xpc0gccFbJMEfQQ8O46KzCE2R
bMrn5j2EzpzYB1dRvjUU47NSHNt1ifGF6bnlwHOrLqv1uObrse7njwcR52FVl9UYu8ZWtF/OI
qZ42ZNYiPVHRBLq3QxNEJ+c4j2zqjIb47D48jbdE0k9gIneltcTbfI/MnThx7cSvcBnrc2yPo
gYi84M04kpFndKjx9N3Pj/d6rPx2fv+QLZlIW9thHTuHyFGHX7orRvGtFuv5KYsu86YsqJO+m
BtTFGiPHL6Xz/4izGg4hrxyXAPUTS/4/UNOmdxVN8TicTUyttUnr+KKKH9O+Its76xKiiXxE3
pHris9pHcRadiWYQ+Y2TGXpDalLp9etl54+3VV3+lcR2OnFDeIP0eK9U7lW6OcHQ5PfriBsAz
6o6vm71mH5+LjGa5BnZvm71eFL6e3yh6vK79cdWeQHcShUSdwUPB6Zn+1p0kuDXAPuk/bNTw+
VPwNVET9Oz6aFh3Sm27wI/Lh5nz5WT4LcSi5b1VNIixXA+cHNqYN5ONvdler5Igl8NnFh1eR9
EnK9LJ7F7gJdldZz3kl5Ij1zsFuXPfs6n4rkhfSe/T1z0foLoJfSc1Cg5ouqyP5h4iQujmxg6
rDtPRv0hfabvIHr79cz3Motxw/R9/Hg5RmJhpL8SNx4vK38W6r5ZjyPXY69tffZ5bVyMDO/19
BFSAio9nkysa3IbQxPEDyEuHnuhLddXMRI3839N9GqfnvbNIuah3SJ73Vyi88b5pE4tdd7otN
VmpfguInpH5+eTDUmjhIie4dsQ7df3Vl3+Va3H0v7tialNlpHab0RP2vWIhNwHibnNt6s6hjU
cY4vovHIU0YZ/WfY5qOX3kk5bfBYx+uI6YrTJSQxdM+soOvO1X0dcb36XTgezWnc06pc4szhm
Eh2IriGuI+8EvpU9X06erl/38wcxXduRREe+S9M5YedUV7uswu/Jr0Vfm77DL686vlHKO2LeY
4R6PJ4eWevMrfe3ygvgtpIKGnrAK5Lg/yb1BE/7Z6QGWp7U6ImTXSrr+cDFIzyXJ8E/QvQoqv
3dbIZeKL2OmAfyAKKX0DvSif2i0ns2Ji7sr6THpiAoxXEKnQXmHpb2tdI2kZgr8lNVl3NV6zE
9Lm4uzSSGnv1v+lwemr3m0ake9626/KsTY7Z/BvAxYmTCsfnr07HoMuLu/kOqjuFBxDiHSGRc
U4pxArAfMYR931475liPvVuP/RJnP8SYx5m+h68DXkIkFou5zIt5sScRCeJbiUWj55Z+T50v8
BsfY6mc56dz/CGkBebociOGWOjsc0TSv2cu7Ilk6F/Sd+2xjJC4T9/VfYnpMn7fK/WXlX/zLv
u2J3qCLyNNFdKrG9GDf9hCqwxNgh8+yvtrfR1J3FS7lJiO5gxirZNB4FPAxtnrXgDclT6nB2X
7a794aT/EmR83gA8R02c+nFg34nSig9FPur1+pN9Tty0dUw9N54JLiM6LC9P58mnEwsH7pePp
XsTIqX3ojPrPp8A5lhgF97Kq4yrFuEp5j3J90SNtOrdmbJUXwC2rjOHzCg47GFDqCd7tgE+Ne
0R1a1ABryd6084ZLZ50AhkyRUodN4betHh0OhGcSOcicU46gS1ieBL8IcBWVccwxjhH/LwS81
4OEvN+71g8T8xRfwvZvPV13UrxnAl8g5huaN/S66ZlP2+YGqVX0GMJReBFRC+nQ4qypwbor4m
L/Q8Sc2DuTawu/3u6LP5Zty373g0QjemzgVcAm6X92wK/ScfUTwI7ETc3fkkka3rhmGM9NqAe

+yXOfoixFO9M4mLwNmIqhUHgnOz5Fb30iPPMIHBM1eU2xq5xvoKY6mRvOlPxzUmf2Z2z152Yj
q3/YBV6+VW9EcmWdxFTYGxNpxPKDsT8/Hunx9OJm/9/JxLltZ92sRTnu4m26Pwuz+1I3BheCj
yn/PepuuyrUI/fIXpgdotxZ2L08P3AIVWXdxXiytvlG5KSpelxMQXjUuLaI08Ov5zoOf1j4NF
Vx2Gcnc9pFtOzgfeSpiBK+7teL1PjZHdWxj3TeWIgPZ5OtGNuJXq2D6Y6XJp+LqZ3W549PrL0
O19LPZPfq5X36IV6dGvmVnkB3FJFdA4S5dWOu/Uq2SU1XP5JLMbXEw2yrPzTibkTN0o/H0AMk
SkufrtNOZE3BmoXbzp5P6W074h0ArsPeFUpnlnZyeBHVZd/NeLN56TbmkhQlOele2eK/y/Aq4
D3EFPYXNlLJz3ga8QNmp8SF3vLyBa6zF53CHAeMdSrZy54U9nPT43mu4i79JfQ6b2/HTFv5F2
pPu9ihIuqum50hlVeTUwHchdxgVvcnNmGuJBYkMV4GT2y1oD12Lh6bHycTY+RzjlyAtFr7wfE
efKxRK/gQeD92euLBPEk4txZ+0RiP8TYJeb3Az9PP08hLvYvJ5Lig8Ab03PPJEbD9USHhiy+C
cA3gW+kx+sQ7Z1bsu/icem5/YipF2o9XcYIcT6OuIb6E7Brl+ePoZOEel7V5V3NGPchRlxeMU
KMp2YxPrXq8o4hnvy6YytifZNvlV4zjVgTaylxgz9PDr80fY5/A+xRdTz9GiewK1kP9bSvGDm
8hDSnNcOvl+8lTZda943oBPYXYvqrrbL9xboKfybaNM8ikuRPIvIiB6Rzx9MYnlN4dfobvbTq
+FJ5+irv4dasrfICuA1Jfs8EPk30cDqXURbOI5LgNwIXVF3+VY2VTs+Ke4H/AD9PB8wziAunh
2Wvr30PL+KC4RepEZIP5Xoo8D6ih8XnKF24p5NBcUL7dtVxrEodZuX/PyKJsYDo7XRA6bVvSf
HdSwy5PDb7O9TyYomhvWkfSyS+9wKmEot5fYZYROhYOkO7DyMWhfoZqZdGnbdSjC8kEvv7E/O
wnZAe/5lOj5P1iATqy4AD6TJ0uG5b6bt4EtFLbcf0fX0dcWGYxzgnxflUYI+s0VbLz6n12Jx6
7Jc4+yHGUrwDxIiML5L1dk7nkQ8xPEE8rfT+2sfZDzGWynsc0cP9FGLdjyXEjceXEqMV7iP1n
KYHE/wpxjOBu4nRbD8k2jsnp+/h54j23voM7ZhS21jpch1BXIvsSSxqeSWRlMvjOZq4EfBBem
D6mpXEePUIMb4D+DgxCrfW30OGXjddRlw7LiWSiPNLry2Sw/cRo03mZc+9mhgJV8u2T5PjJEY
lzE7lOrH03A5EJ6kHgA/TOdcXf4+ZwGvS+eTsqmNZSZwfIzqXnADs1uX5ASIJfms6vq63kt83
iRgd/n3gJVXHl8rUV3kPt+ZtlRfALVVEzJ1Y3Kn/ZTp43kDcFeyaBCaGXNa20TlKrDsTiwi+n
OiB8GM6i10MphPgtenE3xM9L4gbErPSz7uX6ugjKa7js/3FyWA2MaR2h6pjGGOceWPkz8TNiy
cRd6zvSHEeXHrPaWn/e0l3wql5YzuV8S3EMNlvkc2BSSRkPpE+p69N++YQFxrrV13usdZh+vn
wtJ3O0GTqS+kkT3esusyrG2P6nJ6S4jum9Jqjshi7XuDW+fhqPTajHvslzn6IsUtZj0vnvoXA
c0vPbU4nQdwTiwf2U4wjfc6INt0XiKlevg8cnT33SiKhumHV5X+QMW5PJHGuIBIcj8meOxP4F
aUbGXXdSufDjYAtyKb8YmiCeC+ih+acVMdn0hsLlxY9hqcQicRHM7RHcB7jPkQbdmtiTuljyr
+nbhvZCAMi6XYZMc3bp9Ox5X/K5wsiOfwmogNLebrGOVXH1OdxbpuV/Un5flZ+vXxIXT+nqYw
nE50Tn8bQddnKo/tnER2n/pvqeoNuryu9Z4Oq4yuVpy/yHm7N3CovQD9vWaOlRSQqfgBsRiTD
d00ntFuJxRFG7AlNjS8IRyt39pqNiYU9zyJ6De1HJMY/WOcT3QixvJdI3D8n27c1cTExCJyQ7
W/l//bKlhpnXwAuyk7axTQhv0pxHlh6z7vT/o8C21UdwxhiXC999wbpskBrev7jRE+wN9f5O5
jKOxN4DkPv1D+LzvDXk9K+vMF2FHFT7k/ATlXHMIYYH030RMwveI+mM6/e4WlffjPjKOCvxBD
2nddmea3H/q3HfomzH2IsxVteNLlFZ2j3/1DqjUe09z6Qnn9N1eU3xuExAkcS7dH3EXNgF9O4
bMLQHpfrEe2in1DTxFMpxuL6Y4AYffklYnG9Z5CmdyMSFcX8tZOJxMZlRM++2rdbS/V4NpHQv
5e0IHTxWSWmmPgr0bP9UuLa6056IEHD0BGZPwauIzpnLCDapkWnk92JKRnuSf/+P6JNUOtrLD
rXSdNSjJ9h6Fz7p6e6uoDSTX7ihsCK9SKo8TVXv8SZx0vcQPwPWacpVnK9nD2u3ecWmJu+g28
lRgznnVTWJRLG+5B6fKdj72FEwvzXpAWVR/ps1HWjD/Iebs3bKi9Av27ZQWCAGPbzWbIeMunk
sF1qjK00CV7HjU4DexKR5N4xe65VbOnxj4DzRvs9vbABTyaS+b8d5WRwXNXlXI248hP5VsRQ3
6ekx18hkt8PI+ZVvCfF+czS7zgz7X9/3eqU7ouzbkncmV+WGilTSs+vS1w03gbMrTqG0eqOmK
rmewxNnG5K9HK/Dbgw258nT49Mx59fkDWy67al4+gfKfUyJJIvpxPDR78/Sox3AOdWHYf12Px
67Jc4+yHGEeKeQTYVGJGQeHs6972D0uLIxPn0WGp2TuzXGBna1vkGkZy4iuipdxOR5N6o9J59
gc8TCapeuMmYD0cvRlxcTCzYeSMxpcuG2evnAIem88flZJ13qo5ljPEWHTROJaZe+nr6rJ4Db
JNeM5dI7P8oPd8zI6bS9/EvxIjMI4nRYF/IYiyS4DOJ9veniaRVUY9178BR9IheQiTtNyg9f1
o6n5xPl5FDPfQ57Ys4s/I+mRiZ8HuyhVjpXC8vp4eul4m2zd1kvZ7T/iOJ9XiKjio3Anul52Y
CzyemuemZxWi71GPj8h5uzd4qL0A/bcTwuk+U9j0mOyi+r8t7tksHzhuJnn49cYKj0ythZmpM
/p24ePgNMUSvPFTr88AVVZd7FWPML5TyZP7e6WTw+y4ngw/TQz2hUrmLudjyBMWTiETa84Brg
Mdnz32cmJNukNIUPsQFSG3nyCYujtbPHm9JXBxeQ/SMmlx6/VxGuGtfh424MPoIWUIiHXOKuc
s3JnorLAe+lL0mr+sXkC4S67gB84meh7PT41mprlYlxqdT44tA67EZ9dgvcfZDjCPEPYGYN3n
YwnlER4dBYvHHh4zw/l5IEDc+xiyW/6TjbNFj74IU31HZ615LrP1xBT00YoFItn2H6NG9DZ32
60+JXtJFB4cpqT5/A3yXHkmaZnEeTCxQ+tTS/jemujyjy3smr42yrYHYijp7A3GNtXPp+SLGE
0f5HbWvR6JX9InEjaib6Ny0yEcOnUr0er8I2LLqMhvnsNjy6+W8l/o+dL9e3orONBrPr7r8Y4
xxq1Rv5xKj+Odn58obiI4q7yHmQL+NtM4b0b6v/cjoLvXY2LyHW/O3ygvQL1tqbJ5N9yT349P
B8GrSXcHS89sSd/e/V3UcY4y1SJjOJIYVXkokSg9IB8E/ppNefiA9MTVSB6ou/xhjLA8DnlR6
/LgRTgbbEkNpe6Z3SSr3FOLu7kml/e8g5hXcsPi7EImPc4hefsXFUu1HL6TP55L0ec3niNwyN
Ui7JsHrvGXfuQ+nx8V87C+mM5T7IUQyapChyajaz32Zyvl8sjlmiREJg8TUC0WMG40lxvL3ui
6b9diMeuyXOPshxlFi348YBn0LpQt34uJ3OfA2RlnkvO5b02Mkpv64OMWQH1/vIHrWTs9e+2h
iAeLNqi73GGMrEhabEtcVr8yeK6bTOj49LmLfLZ2DarsILZFEGnaDl1hc9046vaDzDhkfJ0Ys
blV1+ccY48507/l7NjH1SXHDMZ/65Rxi9FdtO2p0iWdYR6/0nXw1cDuxTlbxOc6Tw+8levDX/
nqjz+IsOsRNIK4R55SeH+l6eTvg+Doeb7IyHkma0zw9flU6ht5BdAS7lxgxvUfpPYPAi7v8vt

rWKX2W93Br9lZ5AfphI80TSGcuvQHg1NJrnpwOlBeRLSaQPb9p+eBTp42Y/3Bm9ngK8NUUTxF
/MZz0n0QycR86CdKXEHNg1fbgn8WWNy5fD5xH9KQpJ4fzk0E+x1ltT+ajxLwJkQQ+n+ilUFwI
nUVcXDwk1fkWxIiF/OTXE/ESN6lOSHX2U7KVuekkwf9B9CjqiSQ4MU3L14hFdR9HLPD0Z6KXw
hF0T0Z9oepyr2KM6wP/S/TW25W4EL4ixdwtxuXAF6sut/XYf/XYL3H2SYwthrYFJmc/70vMBX
0rwxPE70rfz1eujXIa45jiLF/YF8nuN6fH26bH59OZH/tlpAXc6I12a9HWLr57Dydu+D8vPX5
eqrOT0+MBohPDY0q/p3axEom1bwD3k5IsdNqopxLXVjtlry0Sck8mprg7cG2XeTViHCCuJe7p
EuNH0vdwiyzG4rkjiOkVdqs6hjHEODH7nE4k2j0rri2J5PCr0t/gsizGPDm8Yi7squPp9ziL+
NK/M9Pn9OcppreVXldcL/+O7Ho5e75215HEDalBoq2zdbb/YKLn9yeJufeLG1NFPT4BWAQ8re
oYVrUe0899kfdwa/ZWeQGavgGPJZIUr8lObm9KB83ydCj7M0oSPL2mdknwFOON6YRdxLhlOgE
8Iz0+n+jhPT+9/g6iJ/je6fkNs5NDbU/oDO09cgGRyD8P+ESq04+Rhnmn1zyOWGzmX/RAI3sl
sR9FJCoel+3bhlhg7wZiQZN/EHPX1e5zWoqlfOd6SrGfGDb6HyIJnvcE3yJ9l/9EdrOnjhuRu
Ci+TzsS89J9Oz0eIEZm3MLwZFTRs/gTVZT7QcS4S2pQfnUMMZ6eYjyt6hisx/6ox36Js09iHC
g9nk7qEczQKVyKBPF/Kc3tCRxDjS8I+yHGrJz5SMQfEFNFzSPabWcRieIi+V0kMnYi2ulH0QP
TEtJJls0h2uEHEu2Z24kE8QvS9+/k7LVPInqgPrOqcq9ijE8hpmm5lmyaPeLmxd1ko0yy555K
JI6Hjbyt40ZcI/6a6O2dx1gcaz/X5T0vI65Ttl1b5VyNuLYoPZ4FfIuYb/564EI614oDxLXm3
UQytfi85jfnavmd7Jc4s/KVR4NfRqwVUdwg/SRDr5f3IUbC/4dsWs26bsTCwIem88P3GJoEb4
3w81SivXMVPbBeRF6P6ee+ynu4NXervABN3tKB7qfpALEAOCbt34gYsnY78KnSe/Yn7vr+qBc
aZV1iPJroITwVeGI6Qbw4ndT2zd73f+k99wK7ZPtrfULPyvmhdIB/THp8Kp253M8rnQyeRPR+
65VhluXFHos7+BsAvyLudq8Yagk8iuh9cxGxmGut54gsncwPKsdNpyf4jcTF/dzsNZvnjZy6b
URPy2GNKiIZMQi8PD0eSA2wcjJqY2Ie9GHDbOuyAY8A9uyy/3iyYYWjxLgJ8FJqnJyxHptRj/
0SZz/EmMq5O9F22TfbdzGRAC5G+OUJ4v1SrHfRZYGrOsbbDzF2KxsxDd/NdDptnEz0Dn6AaNM
V7Zr1iHbO5cDmVccwQlx5wmVy+nci0SnlF8D8tO/tKcZB4E3Ze7Ynkm7fpd4dUqYR0y29g+jN
/ZRUL9eSEsTEtcibU4yfJkanTEzHnHOJaWA2rDKOlcS4ffpsfom4ltifuJ7IYxygM/Lrc8TNj
bnE6JvfEcm5Wl5bEUnPpdn3boCYz/xXRGeUd6TjzyDw7PSaGURb6A6i000tY+u3OOmS9E3fvx
8Q14cbpH3fIkYOL2f49fKTievJWl4/dol5OvBcIm8zLAnO0OvNOcR0WYuA11Vd9tWItbF5D7f
+2yovQJM3YhjaSamBdTHRkH5dem4DYjjQHQxPgu+XDihnVx3Dasb42tJB/yPE/NGzs31fJeaE
+nCvnOhSuScSvX++Q2fo6Ikp7pekn4s72/k8kdOrKO8qxPWE1CgpGi3TicVKZ5Re9/bUcNmhy
+/Ih+fV6qI31dumpX0vSnX11mxfkQSfQtykGiSSAfOqjmEMMU4hLuAH03ew3AvqIqJnxR5p30
D63t6S/hbFAnW1/T4SNyCKBtcbgUdmzz2cuLj/O7BrKcab8xiz99Tqc2o9Nqce+yXOfogxK1v
RNvsxnYvAlxIjg/5C9wTxO4jODoPA/lXHYIxdY94J+CIxtDu/KVMsYHY6MXrx6cRUVHcCj6i6
3KsQ32yijfc9IllTJMW3SjEuB85I8R1DDF//EzVew4XoPfszokf0X4m52OcQoxKuJBLExbQnm
wBnEnPyXgP8Ib3vdrIOOHXbiAVY/0Wc879DJLanpxivIHqC5zGeDiwkerVfTySrfp/Vd63qkU
gKLyFGWhTHlfekMm+Tve5kSlMqpfp/YzpO1bat02dxTig9npRi/wbw6LTvG+mzuQ+dG1Mfp8s
1ch3jJRL0zwNekO2bmo6rd6dj7LCkL3BQqvNbgVOy/bW+qVHUAw3Me7j191Z5AZq+EXNE3Quc
QlwkLAOOTc+NlgTfgxpfCK4kxgeIRTyK6VDOJeb93oa4I7o9MRTqkOx31PFENx04lmg4n0gnO
bw5kZiaRSyKeBtwZHpu03RyHyQWgywupmp7kksNkfuJ4UyTiZsa3yfuUn+NGF5ZxDGJSL59LX
t/udFTq1iJhMtHU8Prudn+bYmhW4PAGdn+ItZ5xMXSEqIHQ60uHkaI9UcpnmtT3Z2QPff8VM8
fANZJ+2YQPaYeKBo2dd7SZ/NHxEXe39J37LTs+SPSMfbM7PhTzD88CDy16hisx76qx8bH2Q8x
pjIX5//9Uzw/pdOT9vlED+mr6CQ4JqTtc8Sw4fdQw3ZOP8VI9Bg+FDgOODrtK6YkvIZS8p5oq
76fSJzeSbRjfwrsXHUso8T40BTf2USbfCKR3B8k2qY7lF6/FdGx4VYigfNrordxkfyu3XUIMa
XCP4gRensztOfhBODxRBI8TxDPJpLkXybagu8Btqs6llFi3CMdUz8MbF96buIIMc4grsfOIhZ
KfBWdEZy1qsdUF0uJ6TDy9aO+T0w3VByLDk2f3RPS43XozHU+PXtdLY87fRTnw9Ox8osp1uL4
sS2dEWAnEjd09kqPt6Bz4/T71HxtpfS9uinV52A6Vm6YnpvGCElwYE8659Mjs/21u6akT/Ieb
m6VF6CpG0PnwnwbsWDZnkRvi6XAa9NzRRL8drrM11q3Rssqxvia9NxuRA+T36UT/uXp51qeyF
OZZxE9YK4nGqGDxN36YkHPWenfTwM/JFvVmhg2eiHRMN10bZd9FeMshuWdRWeOzxawWWrEXEl
neNPh6fm3ExfB+1Rd/lWox18TFzyzS89vSfckeAt4GtHz5gSyXhp13LLv4YHA14EPEhdA/ySS
U1ul598PLAYelb13gJjrc/u1Xe7VjPH5RLLlDGLY77+AS+lcBH6YbKGoLMZz63zMsR6bU4/9E
mc/xFiKt7gYfDpxQXsxaeoXOgnivxLtumJR6B+TLXZV93ibGiPRFvhV+mwuIc75vyTarN9Pj8
+mNP95eu/DiN6425C19eq2Eevr3EC0z27PYnwU8O30+G0jxLgZsB0xdUbxGajd9QfRCeNr6fi
yZbZ/QlbuIgk+pJd06ffUNjmTvlu/IKbaya8tWl1ivHKkGLP31er7SCT3FxHXjMW0GMXNtN8D
30z7iqTwSenxZOL88h6G9jStZV32UZx7E1Og/pXO6MU/0umgUYxA+CGRIM/nMfVjAOMAACAAS
URBVP8F8BVipG3tEsJZOT9LdFY8mpjj+kxi2q9fZq+ZxNAk+JbZc7uTzcNfx1jpk7yHm1u7bQ
J8zf4x407nJnR6xxR33vcjGqTPIu7Qf4nhSfBiuoUTq45jDcdY9HZ/XDqQ/iad7Go7T3Q6Cfy
bTu+SzYheQvcAP0yvaREXfz9Pryv+Dpunx8+m/gsl7p3q6IOkqU6yOIrHs4i7vr8nep1+lUiM
3wecWnUMK4lvBjH1zoXE8K2un7n0mf54+v69i+gptHXa9zmyqV3qvhFDgL+WvmObEzefriAap
qcRF8iXEo3TKVWXd4wxtUqP5xKNy/PSzw9N9fwX4oLhAOKGxy+6xVjHY4712Jx67Ic4+yHGke
KmM03Wa9M544fAbmnfc4m2wx3E9AxXEzf8i/NqLRMYTY+ROKdfQyTq9yWS9icTIxd/kh7/jEj
uPzuLv3ZJilFizNtz2xNtt+OJG6U/SOeRXxI9GA9aWYx1rMdUrk2JpO/Rox03iF7S+9JJgu+Y

x1vX+FLZdk+f11FHxxCJ1H0ZPu95bWMkrgWXEiMObgLeSWdkUIsY2faPdN4YJK69imPLI4let
G+qqvzGOSzO4rhzNnGDcBYxAiVfAHsi0S64Gjgve++uKc7H5p/pqmPqEuOniAT/E7I6GiDa4r
eTjaqhMx3KPcRNx2Gdp2r6veyLvIebW7FVXoCmbOlAPkjMuXYOab6r7PlvA39KP88GvsDQJPh
GwBuoYY+LNRjjNIYuOlS7WNNJ7VriQukhdBqS09MJ/l8Mncv8RKJ3+ynEUO/PEw3XjauOZYx1
+X46Pb+Lk9lu6USY9+jairi5cRWdoeuXVR3HSmI8kRhpsHO3BgewfvbzlkRP2/uJoc7XEkO86
jw/5ABxgfvQ0v6tUqMl79H+TuJGwL+Ji+FbicZb7RqbpVh2Jy4U5pf2b0tc2L8523cqkai5hU
i03UrMnVi7xqb12Lx67Jc4+yHGVPZH5rGkfXmC4m5iVEYxtLlIEO9MXDB/i2gz1Plmfz/EOJC
Ol93adB8m2jJ7EonVy4lkaZ4g7oXP6pD2XBbjVGLKj/uJNk7PxpjFemCKddg6NNlril7SM4iF
MX9N3OwY8T112ogbTwtJPS5XEuPcFONvUz3XdtE5Iim8mGjHzCGuGf9LdDwpepc+is7ohc9k7
92RGMFxaR2PM30a52jHna8Avyu9/q3pXPIB4gZkEWedb9gUa2J8OdtXnCP3SnW4B6XEL9Grfw
lxY3X22irvasbYF3kPN7d8q7wATdiIu5vFari3p5PdcuIC4Yj0mt2IoSXFNBIbEr1L7wNOLv2
+OiaGH2yMJzF0nr7anehSuY5JMb4v21dcJBxGJIAfSTZ3YGrUDNJJnNY2aZrV5empzGemfcUJ
bzei4fYZ4oZFuaffLOCQ1JCp3ee0VNbzgQu77H8ZMb3En4mhz0XPoHWJYc5nE3e+azs/ZCrv6
+kMNTyu9NwRRK+TZ2X7HkX0DisWrLuQGt+tT5/TM1JZ/0JpxAExv+WQXlLEPITvymL8ZNVxWI
99U4+Nj7MfYkxlnpCV+dQi9vTvbkTvtk+nx0+hkyDeI/sddb/Z3/gYU7lGa9MVw9UfmR7nCeI
D6YHRNXRpz6X9xTysrwNuBLYuxXhtirFnRril8h9EJJZ2SI+H3fyl057dmli07inEDbht11Y5
H2SMxxDXTcXUA91iLBLgj8tirO2UUkQiuJhmqEgCT2JocnhO2v944nrq70RHqvOI9lG+oKdxV
hvnWI47PyN6TRcLYE5N71lOrCV1ETVdoDWLZy7wPmJ9ndOL+kz/PpsYCfUvYprC7wAHA5ul5w
8DXlF1DGOIsfF5Dze38lZ5AZqypYPkGekg+TbgOcSQ4FuIHjQvJ4btfSx7z/rEggE/paYJ4TU
Y46U9FGMxJO30/KRMzNW+jOhlsYi4e/0GYvjs44gL/Q2rjmGMca6TxXlG2ldc9H6KNAVK6T3d
elHX9aJ3EvBNYsjvenQWX/1JivkOoqG5mEjYbFl1mVcjxsnEULUriYulC4kFS6cRyfwLiDvzG
5fed3B67cOqjmEMMbZSw+s3xEXvT4mbFNOJG2zfI6aqWa/0vgOJRV1r+fm0HptXj/0SZz/EmM
o7ZyXnyAE6SagDiHbRRcCTqi67MQ6JcbQ23ReIZPe62b5NiQTUvWSj4Oq8MbQ9dzpZ0gz4A/D
j0us3TfuX0ANruZTKvn36fL4/29f12oKYV/gz6efp4122NRjjHum79tbRYiQWA/1/DL8RWdek
6bYMn25xMp3k8LvpzBu9B5F4/AUx9eKbqfHCrH0a52jHnZ/Sueld3DB/V/r+7kP0NK7tWgOlO
GcRvdzL58l7iZEXn0znxWIhyPuJhXY3y35HbfMf9Enew80t3yovQJM24mKiOEi+hkhePIJIuv
0w7V9EdqcsnUBqO/ynT2OcncVY3PE9PZ0APkbcLf0MMb1GcXI/j6yHey9spTg/TUy3MFLye0N
qvPjTCPHtkWL7CZEMv5UYvXA2kRSfQiRl8p5utf98dolzY6LRfC0xT907iBtPxcI0B6XX5QvP
9NpndUOip/T/Ixae+VDa92SiB99+5Riz99a6cW09Nqse+yXOPolxpedIOgmO/dPrPlJ1uY1x1
BjzNt19wN7pcX7RvwXRe7HWo8BGifHktO9/id6JRU/iPEG1JbEgXS2TpaPEOSfVzY0MHRk1of
S6hxFJqddWXebViHEeMZr2WuAZo8S4L5GQOqjqMq9mnN2Swyt6SI/2nl7amhznKMed61L74AC
iw9yFxE2dQeCD2ftr2fN7JXHm58m52WtmEqM2P9lrx51RzpGNynu4uRVb5QVo2pYOImelg8Nb
076JRBLjtHTAHHbS65WTQJ/FWJwMLiN6Ch9K5858kdDfj5i/t/a9MEeIcxYx9/ViogfqsJMZs
cDFP8gWL+mVjehp8M/U4Pwaccc6v7CfQVxknFt1WR9knFOI4b7fIC7q/01MlfE54AbSfOe90q
geIcZJxFoJ5xHzY95EXAB+k1hcZ730up45zliPzazHfomzT2Icyzmy6Mm2Jz2S3O/DGLu26fL
YSq/v1Rg/kGK8jhjdtkt6bsTvYK+dT4gRUguJXuzP7PL83HTO/AewRdXlXc0Yd89ifFbpuUlE
L+NfEvP29vLxtVty+J10pg/p2dj6Jc6RjjuUOr4RI8ee14vH1lT+Ec+TIx1Du51b6rqNdI6kY
XkPN7d22wT4+PxRhx5Eziw913Ory/dxjLOIoWqLgf8pPdeTJ/AR4lyHznxebyk9twUxNO/PdO
nJ1wsbkeQeNkSLGMr/iNSQOaHYV3V510C8L08XRsuI3u93Ez0WemYI8BhifAFwMTGX4B+IoYg
fpsfmM7Uem1+P/RJnk2Mc7RyZvaaV/dxz7YM+iTFv032z6vKMU4yzicTaffTAfPsPIs4DiATx
dcCZxOiTdYnF2s9N58uenpc2xXgvcWPxLGAnIrF4LNHz+wpqPofyGOPMk8PfAm4GPtqktk7T4
ywddz6V7W/RJTnci+ePVO7RrpebcP3YF3kPN7fKC9DUjaEJ4rdm+1tNOEj2UYxD5jirujxrqS
7fkvZtnhrZV2WN7J4+AZItakX0EipWr96i6rKtgdjyIdxbAm8kpncZJBJS61RdxjUc40bAq9O
F7iDRw7/njzvWYzPqsV/i7IcYU2zDzpFN2/okxsa36VKM721yjCnO3dM5cSkxVP8+YuTJT4Gd
qi7fGopxF+ASItm/hM6C2efQI3NFjzHOPDn8M2IajUacO/olzj467uTnydOqLs841WOjz5Fub
sWwFI2DVqs1m5hD6Q1EgvhtFRdpjeuzGI8jLgobFyMMq8uziakX1iN60TzQarUmtdvtZVWWcU
1ptVpPAI4k5gB/fLvdvqLaEq0ZrVar1c4O6q1W6zHA0cA72+3236sr2ZrTJcY9gacTI1GWlZ/
vRdZjM+oR+iPOfogR+q690w8xNrZN1w/1CNBqtdYjbhTPJ6Zi/DVwQ7vdvqPKcq1JqS7nATsQ
I2yuBG5pt9vthrXLJ7bb7eWtVmsi0G6324NNOXfkmhxnHx13ZgOnAicA72232ydVXKQ1qh/Ok
epvJsDHWekg+bJ2u/35iou0xvVRjKcQvTFParfb7624SOOiFOffaVjyu9VqTSHmTJxL9Kp9Rb
vd/ku1pRofRYO61WpNbrfbD1RdnvHQJfHWiM9pznpsjn6Is+kx9lF7px9ibHSbrh/qsZ+1Wq0
J7XZ7sOpyrEl5TE2Mr9DkOPvluJPifDfw93a7/eGqy7Om9cM5Uv3LBPha0Gq15gAvIubja8yF
YK6PYjwB+Gq73f5f5r1eUZL61Way7wHOCLqfde0xIYexKLsXyr3W7fVHV5JEm9o4/aO/0QY6Pbd
P1Qj5LqpV+OO61Wa0q73V5adnTGSz+cIJpnT5Bibdqd+ZZpal00ZbihJqk5tz5
G5JsfYT226JtejpHryuNPb+ukcqf5hAlySJEmSJEmS1EgTqi6AJEmSJEmSJEnjwQS4JEmSJEm
SJKmRapUAb7Vah+rt1qtVrVrvan256nJJkiRJkiRJknrPpLoLUPFZ4vqC0KnHyVmWzc6KorV
2KHa4kiSJEmSJEmSJEmSelSWteVeteoADbwC2B2B2YDx1RcFkmSJEmSJEmSJEmSJElSD6tVD/
B2u31Pu93+MdEfA4yzZwL5mvMTQV36dV1RRdJpkKKOHjPpK8Zwtb2IxzKWt8Xbg2VN78ooNmR
ZIkSZkHlerBPia8IEnPnHn8xNxfDbQ/FFUCPbW9z6Z1mtoaxPXVdKk3m9a8EnFddNfK9I9g5tHTTT
b2aqL86tfmF9/cKjsCiDsN0ShZtKk4Nq45lV0yOmxAEfKRHbBrFMrl5E98rTTZJnZIEuTm8==

j9EacxNoMxNoMxNkc/xNkPMQIcf/HFTe3h2sj84/VXXMEFb3gDz/3AB9h8113X1n877p+Ruk2
BIkmSJEmSJEnSGmECXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIk
SY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjTSp6gLkWq3WQcBB6eFG6d+9Wq3WF9PPC9rt9glrv
WCSJEmSJEmSpJ5TqwQ4MB84orRv67QBXAeYAJckSZIkSZIkrVStpkBpt9tvbbfbrVG2Lasuoy
RJkiRJkiSpN9QqAS5JkiRJkiRJ0ppiAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZL
USCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJ
kiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJ
EmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJs
AlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEm
NZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIk
SZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJk
iRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAl
ySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktR
IJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmS
JEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkS
ZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwC
VJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1
kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJ
ktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJ
EmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJ
IkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1Eg
mwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIk
SY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJk
iRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJU
mSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQ
CXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS
1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkS
ZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJcki
RJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCb
AJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJ
jWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJ
EmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZ
IkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJ
ckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLU
SCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJk
iRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJE
mSJEmS1EgmwCVJkiRJkiRJjWQCXJIkSZIkSZLUSCbAJUmSJEmSJEmNNKnqAkiSJEmSJEmSxte
ypUtZungxSxctYul99634d8miRSxdvJjbrrmm6iKOCxPgkiRJkiRJklRDg8uXr0hQr9iyxPUD
ixezJH9ulAT34LJlK/3/Jk+bxqz1118Lka09JsAlSQI2mz8fgOMvvrjikoyf6y+/vOoirBWbz
Z/f6HqE/qlLSZIkqZ/dv3Ahn3vhC7n/nntW6/0W6/0z5s5l5nrrsd4WWzBr/fUZmDePaTNnMmXGjN
gGBpgyffKf6cODDB52jRaE5o1a7YJcEmS6CQULzjuIpLMn4OPeusqouVlx/+eXWoyRJkqS
eN2X6dPY+6uvuWYb25uyJUUGvGvX/xnUuJncb25ezWWYb25y9uyJUuGvX/xnXey+M47+e/VV6/
YN2naNKYCfAuifCBddflkYce
yuSpU9dmqOPKBLgkSZIkSZIk1cyEiRPZdPgffDx/Ta5cuWX6cXXTx4q5ToqxsqpR7br6ZJYsXs+Tee
1myaBGbPuIRbLbLLuMc3dpjAlySJEmSJEmSetjEiROZdPgfe+1XzTuW2SpO6kqbKJOa/wWVkRYsXs+Tee
ases2a0EWSJEmSJEmSpMQEuCRJkiRJkiT1u3a76hJ
L8jnA0/zeS7NFM0faXzy37P77gZh7vElMgEuSBGw2fz4Ax198lGT/XX3551UVYKzabP7/R
9Qj9U5eSJElSP1v+wAP89qtf5e6bbx6WrM4fj6Xn9oSJE5m9oAtRSJE5kyMMCUGTOYOmMGUwYGmLHOOqyzy
SZMmTFjxTaw7ro8ZMcd10J0a48JcEmSJEmSJEmqmWVLl/KHm27i0D49bp7/58QIvVKDmSJ9ZnYvVtQa5zZnD58lV+/4y5c5m53n
rMnDeP6eussyLxPWX69BXJ8CIhPjn9O33OHCZ0albKuFnRSJK0mooetRccd1zFJK0bolzJTtVWVBtU2XkSZ4zPlYJ02MMBRX/oS7XabZUuXrvb0JnfdfDP//+/xK
FWCuuv/xy61GSJElSz5s6MMBRX/oS7XabZUuXrvb0JnfdfDP//+/xtRjvDVhAkd89rOst+WW
ay/QcWYCXJIkSZIkSZJqqtVqMXnqVNZ/OIvqaVnqMuXMf1O9qDw7ywJIlXecMv/Wf/+Q3553HojvvN
AEuSZIkSZIkSeotrQkTGTYgqU6dOZWXpu2uzZ/Oa88yop13iaUHUBJEmSJEmSJEkaDybAJUmSJE
mSJEmNZAJckiRJkiRJktRIJsAlSZIkSZIkSY1kAlySJEmSJEmS1EgmwCVJkiRJkiRJjWQCXJI
kSZIkSZLUSJOqLoAkSZIkSZIkac1bvmwZtvvzqPc/LVYDTRIDdkgJJxTToYkspcsXtxJYJYJefTz8vW7p0TP/P5GnTmDIwAbbb
svcTTcd56jWLhPgkiQBm82fD8Dxl19ekSkcUCtD8DxF19ccUnGz/WX3X553lqQNR5GqJkQTW7Y7bj
57+OEMLl++WWu+6OTPnzWPW+uszcejqb5iLGqsaMl1R/GmA4OxwMupE1Kxy2SXe1p6Yj5MxHLZ3Y3AiY3jZYJcTkiU5
C8YLjjqu4JOPn0LPOqroIa8X1l9uPUqSJenqedNmzVvO/MUmmnYZWz73VpSxyDdNaxJHwBF5a9y
kPd99yy5bpS4ZNaZJtSxYtYnD

MgAW33kni++8k/9efXXX3z9lxgymTJ/OlIGBFf8OrLsuTzz2WKbNnLk2Qx1XJsAlSZIkSZIkq
WZarRab77bbKr9v2dKlnelQStOmFEnybs8vXLCA//zxj+z01Key+a67jkNE1TABLkmSJEmSJE
kNMWnKFCZNmcKMOXNW6X3XX3EFF7zhDeNUqupMqLoAkiRJkiRJkiSNBxPgkiRJkiRJkqRGMgE
uSZIkSZIkSWokE+CSJEmSJEmSpEYyAS5JkiRJkiRJaiQT4JIkSZIkSZKkRjIBLkmSJEmSJElq
JBPgkiRJkiRJkqRGMgEuSZIkSZIkSWokE+CSJEmSJEmSpEaaVHUBJEmSJEmSJEnVaLfbLFm4k
HtuvbXqoowLE+CSJEmSJEmS1ECDy5ez6I47q6qbj2ez6I47WLhgAQtvu/vu417Pt3gULWHb//SveM2XGjA
pLvOaZAJckCdhs/nwAjr/44opLMn6uv/zyqosgSZIkSRpnSxYyu5HtnnnMHt113HojojnvuoD04OOT
51oQJzNpgA2bOm8f6227L1nvtxcx585g5bx5zN9mEDbbbrqKSjw8T4JK1k6
axbTZs3igfvvZ8nChOhUOebw8Ocv+99zJ56tQV26QpU5g8FXdSpLFy+uqNTXBS4EJl0ekdfcNxxF
Zdk/Bx61llVF0GSJEmSNM7uue02rsy+e/uq+...

91BAAAAJpseWkpX/zEJ3L8yJF1Jff0xESWz5xZN7eo1dIzOJje4eFsu/HGvOC7vis7br65Rcm
bQwEOAAAAAFARcydP5n+8+91ZnJ+/5HhRq2V4377sHB3Nthe8IFu2b0/v8HD6RkbS2dtrBTgA
VNHZ1dEH9+9vcZLmuePAgVZHAAAAoMl6Bgfz5o9+NFNPPbV+BXi9nqmz7xuNfOHuuy+6dvOWL
Xnde96Trbt2tSB5cyjAAQAAAAAqpKOrK0PXXpuha6+97JylxcXMTEysleKP3XdfPv+xj+XUk0
8qwAEAAAAAeP5qa2/Plh07smXHjiRJ98BAPv+xj7U41ZVXa3UAAAAAAAABoBgU4AAAAAACVpAA
HAAAAAKCSFOAAAAAAAAFSSAhwAAAAAgEpSgAMAAAAAAUEmbWh0AAAAAAIDmKcsy81NTmW401o6p
ev3c53o9U/V6kqRWq9aaaQU4ACTZMzaWJLnr0KEWJ2meY+PjrY4AAABAky0tLuaeD384k48+u
q7kPnP69EVzuwcG0js0lL5t27LrhS9M/65d2TE62oLUzaMABwAAAACoiKe+8pX8zQc/mJ6hof
Tv3JntN92UG77929M7PJy+kZH0Dg+nd3g4PYOD2dTR0eq4TacABwAAAACoiGJ1C5OF2dnMnzq
VTZ2d2dTRsXa0tbefe79169r8qlKAA0DObQ9ycP/+FidpnjsOHGh1BAAAAJps24035pVveUsm
H3ss06tboDx8zz2ZnZy8aG5t06b0DA6urQ4f3Ls3t995Z9o7O1uQvDkU4AAAAAAAz0GLc3OZO
3UqC7Oza8fpmZkszM1l4fzXpxlfWly85PcvnzmTqaeeytRTT+XxJL1DQ7n1+79fAQ7AOXvGxi
r94MSzqn6PHhAJAADAc8ncyZN5/513ZnFu7mnnbt6yJZv7+tLR3Z2O7u0u70jYyk87rr0tHVlY6
ennOvq+Md3d3pPP98V1fau7pSa2vbgDvbWApwgG/QsfHxq2LbjCrfY3Ku4K9y0a/kBwAAeP7Y
3NeXV731rWk8/HCmG421Y6pev2hF9/zUNo2bUpnT086e3szcM01ednrX5+O7u4WpX/uUIADA
AAAADzHFLVabnj5yzN8/fXnCvB6PVP1ehoPP1PP5z64cNZXlpamVyWmTl+PDPHj+fJBx9wbn
vNaxTgUYADABVzNWxLZDU/AABU39zJk/l/f+iHc2Z+/mnnndvX3Z3Dv3xde20G9+xJz9BQGg8
/nFNPPrluC5TO7u60tbdvQPrnDgU4AORcoVjlrV7Ob7u60tbdvQPrnDgU4AORcoVjlrV7Obmd
XP4hl+edf+y++/LYffc97fe2tbev2wv8/KNncDAvf9Ob0tHVtQF3uDEU4AAAAAAAzzFFrZZbv
vu7n/H8cnk5i/PzKwX5ecX4wmphftH5846Z48dTf+ihTNfrufHlL8+esbEm3tnGUoADAAAAAD
zPFbXa2krur8exe+/Nwbe+AAAAAAGgGBTgAAAAAAJWkAACcAAAAAoJIU4AA
AAAAAVJICHAAAAACASlKAAwAAAABQSZtaHQAAAAAAgI1zZmEh043G9FVr+fJBx9sdaymUIAD
TbVnbCx3HTrU6hhhNdWx8vNURAAAAAANac+OpXc+KxxVr68ruc++ZzcbnmuUAaR/jyaUG8Y2
mtakLh5FOAAAAAAABUxcfRoPvjGN6471z0wkN6hofRt25Zd3/RN6R0eXjlGRtK3+r6jpydFUb
QmdBMpwIGmOjY+noP797c6RtPcceBAqyPAM7ZnbCxJ/K8MAACACjs9PZ0kGbjmmlzPf3zd
N6d+5M73Dw+kbGUlHd3eLE24sBTgAXCXOFsP+UQoAAKC6tu7enZ2jo5l87P87LHc9/GP576Pf3zd
eEd399oK8L6RkXOrwYeHM7B7d7b7LHc9/GP576Pf3zd
eEd399oK8L6RkXOrwYeHM7B7d4auu641wZtEAQ4AAAAAUBHd/f25893vTpIsz
j1TZx98ubof+JHPfS4zExMpv+R974322+6qVXxrzgFOAAAAABABbVv3pyB3sszsHv3Ze
csLy1ldnIyD3360/nzAwdyemZmAxM2X63VAQAAAAAaI1aW9vK9id79rQ6SlMowAEAAAAAqCQ
FOAAAAAAAlaQABwAAAACgkhTgAAAAAABUkgIcAAAAAIBKUoADAAAAAFBJCnAAAAAAACpJAQ4A
AAAAQCUpwAEAAAAAqCQFOAAAAAAAlaQABwAAAACgkhTgAAAAAABUkgIcAAAAAIBKUoADAAAA
FBJCnAAAAAAACpJAQ4AAAAAQCVtanUAAAAAAACar1xezuyJE5l5lNDJVr2e60Vg7Jo4caaXW8pl
CAA0CSPWNjSZK7Dh1qcZLmOTY+3uoIAAAANFlZlnnknnsyceTIupJ7ql7PsYvaqQVS4nZSFzZt38olZ
Lz+Bg+kZGcsurXpVtL3hBi5I3hwIcAAAAAKAiZiYm8uTxtS8b8r/9a9SLi9fcryo5mx+ho
tt14Y7Zs25bNAakKIoNTtt8CnAAyLnV0Qf3729xkua4Q8PH2B+jmY3+0GcbbqTq8b5OvWIdpaoJw
y5aAX72ff2hh1I/fDhfuPvutes2dXRk6+7dBV+1RQuD+2KSTr+ZXyXDo4bsYkDSQiDsXXJ93NNE9
A3DgVrlt3HNTnGy3Jh8zFVb7ufgEdbTdUpx2bM99GSSkkGsKsL6fKfPy+pqEO3dtYQ9VzTy6T7RcOXtvb9Y0
AAAAAAAKAZFOAAAAAAAAFSSAhwAAAAAgEpSgAMAAAAAAUEkKcAAAAAAAKkkBDgAAAAAABJSnAAQAA
AACoJAU4AAAAAACVpAAAHAAAAAKCSNrU6AAAAAAAAX7+yLHNmYSGLu7EPTubhSGGLs7M5PTubhSbhZmZLMzNrbz0z
mbha5w/OzZ/6lSSpCiKFt/NlaUAB4Ake8bGkiR3HTrU4iTNc2x8vNURAAAAaLKlSpV/8lP5u8/85r5+SX
AAV7PlpaXc/8lP5uTjj68vsc8/ziu0l5eWnv5LiyKd3d3p6OlJR1fX2ewcH
suOWW5t/cBlKAAwAAAAA8xyzMububQb/1WFmZmvq7ruwcG0js0lN7h4fh4fSdjp7O7Jt7e0tpeWrdn4H5PRGLzdb4df74PH27O1J
R3f32rFWiHd3p9bWdoXvovUU4ACQytqD+7f3+IkzXFsfHxlwmWqtcvkAAAPe5v7+vLNP/7j6R8ZSd/oaLZs25aewcFN3
/iDnJ6ayunZ2We/8cwQDAmE6fcxPSz9hMAD9743zfSMjTbPsb4MAB4DYAgUAAI
DnnvbOzmzZti1bd3x7xjG1ayxtX2u99d3uwLLV1rbxdpxuDd16+vsr99Pq48S9DLC9VaHQAA
AAAAgCujLnLs0sEOjpiV4V9vuru6lo728DHEdCT2AAcAKh7NMydpnXa1753VcfW2cw3/Pcxk/Z8tovaAtSN48
wKsxJzGg5TpCDu/UldD9dFvHOMqRn/vsE0OhpwlAmQSRAmVHTRjQwBIZw4/Yp6OnDQ6SQ60ZYyMhc
BRM0a8etUmZyWuVVtRtgaVl0mG4VdB/6iIwO5lwMtpF68hmtMe9LxxrLtBR9S9qNTSq/GDv0/rLHKcBs4s43h
DnnvbOzrR3dqb3WVyzvLSU8T/+40weO5apRiPTjUamnnoqs5OTF81ta29P7/Dw2jG4Z082b9l
y5W7gOUABDgAAAABQEU99+MvY8+cv5i9/6rXT09KRvjKRi9VaHQAA
AAAAgCujLMskyaaOjOpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9V9vru6lo728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru61o728953dHXvxOrAAHgCT2AAcAKh7NMySkyaaOjpV9vru6

v7mpoyM/9sEPZsuOHRtwhxtDAQ4AsQUKAAAAz01Frba2Srv3Gcw/s7CQT/+n/5SJRx5ZKcLn5
jJ36tRF89ra2889AHP1GNyzJ70jI1f+JlpIAQ4AsQUKAAAA1bA4N5cHDh3Kyccfv+R4UatleN
++jNxwQ/pGRtbK776RkWzZvj21trYNTtxcCnAAAAAAgIro6u/Pj//u72bx9OlMNxprx1S9fu5
zvZ6jn/98ZiYmUi4vr11b1Gr50fe/P8P79rXwDq4sBTgAAAAAQMW0d3ZmYPfuDOzefdFYWZaZ
n5rKVL2epx58MI8/8EAe/cIXcvzIkcyeONGCtM2jAAeA2AMcAACA6pg9cSInH3983arvdSvAG
42cOX16/UVFkS3bt2frzp2tCd0kCnAAAAAAgIqYnpjIf7zzziwtLl5y/Owe4DtHR7NjdDSDe/
akb3g4PUNDaWtv3+C0zacAB4B4CCYAAADV0DMwkO97xzsy8cgjmTpvxfd0vZ75qamUy8upHz6
c+uHD+cLdd6eju3vlQZgjIxnYtSvf9VM/lY7u7lbfxhWjAAcAAAAAqIiiVsvNr3jFJccW5+cz
PTGxVohPnVeOHz92LEf/7u9y0ytekb233baxoZtIAQ4AAAAAcBVo37z5sg/GPHbvvTn4tre1I
FVz1VodAAAAAAAAmkEBDgAAAABAJSnAAQAAAACoJAU4AAAAAAACVpAAHAAAAAAKCSFOAAAAAAAwF
WhKIp3FEXxmaIoThVFUS+K4k+KonjRBXOKoijeWRTFV4uimCuK4n8URfHCVmXmG6MABwAAAAC
uFq9I8p4k357klUnOJPlkURSD5835uSR3JXlLkpckeSrJnxdF0bexUbkSNrU6AAAAAADARijL
8nvP/1wUxeuTnEzy8iR/QWZZfXZ3zhqyU4Hcmed9GZV1eWsrD99yTp77ylWy78
cbsu/321NraruhvlGWZMwsLWZiZyXSjcUW/+7lCAQ4AAAAAXK36srJLxuTq531JdiT5s7MTyr
KcK4rir7KyanxDCvDlpaV89Od+Lo8/8EAW5+fTvnlzdt5yS37gXe9Kra0tS4uLOT07m8ZXZ2Zy
enc3CzEwWZmfXHReNz81dct7y0tK6397U2bkRt7hhFOAA36A9Y2O569ChVsdouqrf47Hx8VZH
AAAAYOP9epLxJJ9a/bxj9fXJC+Y9mWT3RoV6+J578vj992dxfj5Jsjg3l6Pj43n3q1+dpcXFL
C0uPqPv6ejuTkdXVzp6elbed3ene2AgnT09ae/qSud55zu6u9MzOJidt9zSzFvbcApwgG/Qsf
HxHNy/v9UxmuaOAweSpNL3mJwr+Ktc9Cv5AQAAzimK4kCS70jyHWVZLj3d/I301Fe+ksXTp9e
fLMsszM5ecn5Xf38G9+7N0LXXZnDv3mx7wQtyzTd/c4qaR0AqwAEAAACAq0pRFP8+yQ8l+e6y
LB86b+iJ1dftSY6ed337eWNr+bAUq88SX2Uk3t+c/SENv6N1Pk7RqQ7KqwMQ3+XkFrjczhfsOwzdTdOA
DLdaOSJBx7IY/fdtzb/h37jN7L7RS/aqMjPWQpwAAAAAOCqURQEryf5WayvBUvUjKBIAZ/XMJ2
nymdX5m5N8Z5Kf3aiM+26/PTtvuWVlG5Tp9Pe2Zmdo6P51n/6Ty/6Ty/7IMyyLDN/6lQe+exn86e
//Mv51Ic+l1O033ZS+kZH0Dg+nd3g4fSMj6d669apaGa4AB4Cc2x6kylu9nN3OBgAA4GpVFMW7
k7w+yWuSTBZFcXbP7P+rWLoEqvobKjLorvgPSf5lURRxPp3k9zMpb6Hkwyb90OMp3k9zYqZ62tLT/wrnfl4XvuS
f0rX8nIjTdm3+23X7b8TpKiKNLV35/rXvKS7H3xi3ii3P86NEc+/znL3rIZa2tLT1DQyuF+Gox3j
s8nN6RkQzs3p3tN9+coiiafYsbRgEOAAAAAFwtfnr+/m+S3S/m+S
flGU5tREBz6q1teWGl70sN7zsZ7zsZc/quq4tW/LPFu3XkiTl8nJmT5xTxY2TLlvO1Szr5vPPJIHv7M
Z9ZttfK697wnOyr0IEwFOAAAAABwVSjL8L8mmXNpdlWWalDH9ns/M0W1GrpWdwMD2Dg91+002Xn
Xd6ZiaH/+Zv8t4VeycF4ZXgVXz2YvAAAAAABcpLOnJ33btrU6Rl+MowAEAAAAqCRboAAAAAA
AAXCXOLCxkenUP8KnVPcGn6/U0Hn641dGaQgEOAEn2jI0lSe46dKJmwMAAACwiKl6PR9+7
693+fiSNH1j34crpez1SjkflTpy6a3755c3pHRrLv274tw9df34LEzaMABwAAAAoiK16PR9+
y1suO17Uahnety87R0ezY3Q0O2+5JQN79qx7qRtUzWr4mreFQA8S2dXRx/cv7/FSZrnjgMHWtA
ACAJusbGckbfu3c3c/zo0XUrwM9/Xz9Xh/OFu+9euaoa0jMwkK27dt3au3v3lnb3GS3KKKk3
IU4AAAAAAAFTK8b1+G9+275FhZlpk/dWfk7dg6SeqFh2/Nb9+vqsr77+Fu0Xk/lQmjhxRgAMAAAAAA8PxTFEW
6+vvT1d+fFcsHb2L335fFhbkf9+w7/+ZqFPfaqFyZqj1uoAAAAAAADQDApwAAAAAAApAAAAAAAAqSQEOAAAAAAA1KcAB
AAAAAKgkBTgAAAAAAJWkAAcAAAAAoJIU4AAAAAAAVJICHAAAAACASQQpwAAAAAAAqSQEOAAAAAAA1KcAB
AAAAAKgkBTgAAAAAAJWkAAcAAAAAoJIU4AAAAAAAVJICHAAAAACASl
KAAwAAAABQSQpwAAAAAAAqSQEOAAAAAAA1KcABAAAAAKgkBTgAAAAAAJWkAAcAAAAAoJIU4AA
AAAAAVJICHAAAAACASlKAAwAAAABQSZtaHQAAAAAAgI1VlmXmPsXvvKkpl3CA
A0CSPWNjSZK7Dh1omwoqbBbfqtVDh1qcZZLmOTY+uoIAAAAbIDlpaXMHD+eqz9peBuNNa9n159f2ZhYd11mzo70
zs83KLUzaEABwAAAACoiFNPPpkvulNWZybW3e+rb09vcPD684NL395+kZG1s71jY
ykZ2gobZuqVRlX626A55w9Y2OVXlGbWFVbFWf/jgf3729xkua548CBVkcAAACgyaYnJrI4N5e
Ba67JNbfemp2jo9k5OprBvXtTa2trdbwNpwAHmurY+LhCEQAAAGCD9A4Pp6OnJ5OPPRrrJRx/N
fR//eJKkqNXWVnuvrfoeHk7v6irws+83dXS0+A6uLAU4AAAAAEBFbNm2Lf/7H/H1R5k6eXLvfd
+Ohh/L4/ffniQcSLm8fMlr2zdvhs/8IFs2b9g1M3jwIcAOIhmAAAAFTD0uJiPoKj2TiyJ
G1h11e6oGXKYp0b926bg/wwT170jM42JrgAB4DYAxwAAIBqOD0zk7/8+Z/PVz784ZYAmAAAFy3HrZ3o3/8EfZP7UqbXV3lONxrrV4NON
Rr7813+d+VOn1l13b1t6eledxDyo/4q4ZeYbh/F7lLCPV55BjjtwzUZd5wPMUadRpiRpJRqIUUzGu/v509fdn5IYbLjtv8fTpz
KwW5Ec///l8+kMfyzERKUK8FqrAwAAAAAsPHaOzuzfu7Ln11uy97bxWx2kKBTgAAAAAA
WkAAcAAAAAoJIU4AAAAAAAVJICHAAAAACASlKAAwAAAABQSQpwAAAAAAAqSQEOAAAAAAA1KcA
BAAAAAKgkBTgAAAAAAJWkAAcAAAAAoJIU4AAAAAAAVJICHAAAAACASlKAAwAAAABQSQpwAAAAAA
5k+dSpK0tbe3OPGVpQAHgCR7xsaSJHcdOtCmtn1WAb7N2T8OfCGJ9yI5PPm4DtSybS4h0oruM
wsLF83rHhhI7+Bw+rZvT+UZvz64XvjC9i9Oz/XqnjCBN0yjAARDnyuu9D+/e3o3oIHgDDT
AAgu4Mi4t5/P77c/zIkUsW0/jh3pHR5e/eIjZfrxZ0PmwZ//Z0zYVv1DqB4vWU99bqKRr2d3JUC
AAAAFdjX+Eg2j3ptkGjadgAXyLMcs2ttklMaXN/VRxUVBUpJx+n7pf6+y4oxo7jSf+rmDyi8FfBqJlQzDeOOaAhwAAAAAAASQpwAAAAAA
AAAAAAAAAAAAAAKgkBTgAAAAAAJWkAAcAAAAAoJIU4AAAAAAAVJICHAAAAACASlKAAwAAAAAB
gAAAAAAADSDAhwAAAAAAAACQpwAAAAAAAACqSQEOAAAAAAA1KcABAAAAAKgkBTgAAAAAAJWkAAcAAAAAoJIU4AAAAAAAVJICHAAAAACAS

AAAACVpAAHAAAAAKCSFOAAAAAAAFSSAhwAAAAAgEpSgAMAAAAAUEkKcAAAAAAAKkkBDgAAAAB
AJSnAAQAAAACoJAU4AAAAAACVpAAHAAAAAKCSFOAAAAAAAFSSAhwAAAAAgEpSgAMAAAAAUEkK
cAAAAAAAKkkBDgAAAABAJSnAAQAAAACoJAU4AABETtiEAAAKpUlEQVQAAACVpAAHAAAAAKCSF
OAAAAAAAFSSAhwAAAAAgEpSgAMAAAAAUEkKcAAAAAAAKkkBDgAAAABAJSnAAQAAAACoJAU4AA
AAAACVpAAHAAAAAKCSFOAAAAAAAFSSAhwAAAAAgEpSgAMAAAAAUEkKcAAAAAAAKkkBDgAAAAB
AJSnAAQAAAACoJAU4AAAAAACVpAAHAAAAAKCSFOAAAAAAAFSSAhwAAAAAgEpSgAMAAAAAUEkK
cAAAAAAAKkkBDgAAAABAJSnAAQAAAACoJAU4AAAAAACVtKnVAQAAAAAA2Fjl8nJmT5zIVL2e6
UYjj37hC62O1BQKcKCp9oyN5a5Dh1odo6mOjY+3OgIAAADAmjMLC5mu1zPVaGS60cj0ask93W
isnZuZmMjy0tK669q7utK3bVuLUjeHAhwAAAAAoCJmjh/P77zhDVmYmbnkeFGrZXXjfvuy7/fY
MX399+kZG0js8nL6RkXT196fW1rbBiZtLAQ401bHx8Rzcv7/VMZrmjgMHWh0BAAAAYE1Xf3++
88d/PBOPPHJuBXijkZnjx5OyTLm8nPrhw6kfPpy29vb0Dg+vHQO7d+f2170u7Z2drb6NK0YBD
gAAAABQEbW2toy9+tUXnV86t+cyYzx4+v2xLl/C1SHvviF/Olv/iL7P3Wb82eW29tQfLmqLU6AA
AAAAAzbO8tJS5kyczOzm5cpw4kZmz7ycnMzM5mdNTU0mSoihanPbKsgIcALLywNYklX5oqwe
2AgAAVN+ZhYX87e/9Xo6ftwXKpR54WWtrS8/QUHqHh1f2BH/JS9K/a1d2jo62KHlzKMABIOfK
YXvWAwAA8Hy2MDeXv//EJzL11FOXHD/7EMxtL3hB+rZtS9/ZPcBHRrJl+/a0tbdvdvcOLmUoADA
AAAAFREd39/fvL3fz8Ls7PrHoI5Xa+v2/P7kXvuyczkZFKWa9fW2tryo+9/f4auu651N3CFKc
ABAAAAAACqmo7s7Q3v3Zmjv3svOWXswZr2eo5/7XP7nBz6QmclJBTgAAAAAAM9vbZs2Zcu2bdm
ybVuWzpxpdZymqLU6AAAAAAAANIMCHAAAAAACASlKAAwAAAABQSQpwAAAAAAAqSQEOAAAAAAEAl
KcABAAAAAKgkBTgAAAAAAJW0qdUBAAAAAADYOEtnzmRmYiLTjUamG41M1et54ktfanWsplCAA
0CSPWNjSZK7Dh1qcZLmOTY+3uoIAAAAbIBTTz6zyccey3G22G2/Mlm3
bsnnLlhRFscFpm08BDgA5tzr64P79LU7SPHcdDqQz7/SPHccONDqCAAAADRZz9BQXvNlrwBvzDIz
MZHlpaXUDx9O/fDhtOfWrwBvNDIz9e/Nvf3L5SJI0fWrwBvNDIz
MZHlpaXUDx9O/fDhfOHuu1PUaukZGkrfyEi27tqVV771rdnc29vq27hinnEBXhTF/53kHyW5K
clwkrkrkR5L8UZLfKsty4hLXtCX5SQ/muSbk2x07zENf1J7krymiTXJ1lOcj
TJ/0zyf5Rlufgs7g8AAAAA4KpRFEWtWuf+lLc/1LX3rRWLm8nNkTJ9b2/T8/Z/
8ZF70fd+Xvbfd1oLkzfFsVoC/Lcnnnkvx5kqeDJ3JvnJoiheQiuSPH9e5R5JnVpPNY7ZoVERd
ZDzJ/5dkPsnuJN+ZlSJ9XQFFRqKc/QU4U4AAAAAwNvs8e5nGuUvMZeHXa7CKJkaeyA7kiqJ1pS
M/gYLbfdNO6sWP33puDb3vbxmcqinckeW2Sm5OCvTvLpO/KLq4Yr+UUSHJqSR/KeM/8UN4mJkvQVLSmA/uSh0
TtZVkefbrvfzYF+JayOLocvEfCXk/zLJO9ubN7O9I8tPnH16Zm8hmZ/4HN8kv+apC/Jy8u
y/PQF45ucuSLD2LvAAAAAAPLe9Isl7srJrSJHk3yT5ZFEU5/a0/Psk7luo/2z2zW/zEOL/x
iaIovqUsy9NzUsZFEU31W5fEk3lyx9U6Mq5nuCnkxzsL/+V/8qE+Mz4m9YyfxMl0m7wIPtxF
i8kb3Lcmn10O9uT3JqknuwHuOSzY4nUQnkvzbC7J7Q+J/SpdWsS99vTaetiNCPc6ue
yxmk+5U8YbFtxZR5S1JF7Ln6r9e/UvaTeRUwkF9qJJZ9vv93LnGwDjWHOIe4KZ0KrOqbw
S3gdfSrCqpuYOZxdqb+Wdj4n5Xc77CFxsR7J2Xv1aQ/Fw2yUnw8gdV43nWvE4vh7kRZuz3
EEvzbAA2Um7M6Iw3U9oNs51WqBXT5Du3aPh52X1R/p3AuPWaHdVXy9ma3F6OcPHX77Wq5u6
0XUd0Gs2aS8YFDdqVSj+BTaOdd8e58uf3kCo1Zn2kEveW/Xnl3kl+CVY6H5JM9Uzq/L+QS/1
BOfMCcX0BLqdTSt7PWgNZ8n7CjwnkoKySzBx2WvWFFkkMOAAT4SXhy5Fm8FJiNb1v90cwmwSuXfiee
FXVMwMQqaGntOqV8L2QCgZ5RlL9xlRW7JX1rDtmjg1mDhBOmA7eyX+6Jgw5dhVIzK4MxvCbx
6y9DqWr9+K6cDSr/UXUJ77M+Ol+IrH3fbW9urEGmpRcqcQApi9f3ExkXxyShq

```
      ]
    },
    {
      "cell_type": "code",
      "source": [
"data_c=data[[\"RainToday\",\"WindGustDir\",\"WindDir9am\",\"WindDir3pm\"
]]"
      ],
      "metadata": {
        "id": "qhtEqX3V55r3"
      },
      "execution_count": 11,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
"data.drop(columns=[\"Evaporation\",\"Sunshine\",\"Cloud9am\",\"Cloud3pm\
"],axis=1,inplace=True)\n",
"data.drop(columns=[\"RainToday\",\"WindGustDir\",\"WindDir9am\",\"WindDi
r3pm\"],axis=1,inplace=True)"
      ],
      "metadata": {
        "id": "jrVfj6WW6F8-"
      },
      "execution_count": 12,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "data['MinTemp'].fillna(data['MinTemp'].mean(),inplace=True)\n",
        "data['MaxTemp'].fillna(data['MaxTemp'].mean(),inplace=True)\n",
"data['Rainfall'].fillna(data['Rainfall'].mean(),inplace=True)\n",
"data['WindGustSpeed'].fillna(data['WindGustSpeed'].mean(),inplace=True)\
n",
"data['WindSpeed9am'].fillna(data['WindSpeed9am'].mean(),inplace=True)\n"
,
"data['WindSpeed3pm'].fillna(data['WindSpeed3pm'].mean(),inplace=True)\n"
,
"data['Humidity3pm'].fillna(data['Humidity3pm'].mean(),inplace=True)\n",
"data['Humidity9am'].fillna(data['Humidity9am'].mean(),inplace=True)\n",
        "data['Temp9am'].fillna(data['Temp9am'].mean(),inplace=True)\n",
        "data['Temp3pm'].fillna(data['Temp3pm'].mean(),inplace=True)"
      ],
      "metadata": {
        "id": "zkINJXCs6F6o"
      },
      "execution_count": 13,
```

```
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "c_names=data_c.columns"
      ],
      "metadata": {
        "id": "scFf3uVG6OcJ"
      },
      "execution_count": 14,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "from sklearn.impute import SimpleImputer"
      ],
      "metadata": {
        "id": "Ivk4Lb7O6dSr"
      },
      "execution_count": 15,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [

"imp_mode=SimpleImputer(missing_values=np.nan,strategy=\"most_frequent\")
"
      ],
      "metadata": {
        "id": "7FVMX3j36mRp"
      },
      "execution_count": 16,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "data_c=imp_mode.fit_transform(data_c)"
      ],
      "metadata": {
        "id": "2V8og9hP6rtr"
      },
      "execution_count": 17,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "data_c=pd.DataFrame(data_c,columns=c_names)"
      ],
      "metadata": {
        "id": "vMkVr5ur6wGR"
      },
      "execution_count": 18,
      "outputs": []
```

```
},
{
  "cell_type": "code",
  "source": [
    "data_c.tail()"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 206
    },
    "id": "GlzMMtLJ63b0",
    "outputId": "58345e60-7ef5-43d6-f89a-ac441a18bac7"
  },
  "execution_count": 19,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "    RainToday WindGustDir WindDir9am WindDir3pm\n",
          "361        No         NNW        SSE         NW\n",
          "362        No           N        NNW        NNW\n",
          "363        No         ESE        ENE        ENE\n",
          "364        No          NW        SSW        WNW\n",
          "365        No          NW         NW        WNW"
        ],
        "text/html": [
          "\n",
          "  <div id=\"df-c1c2629a-fb64-4205-ad8b-b6e835ad5596\">\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>RainToday</th>\n",
          "      <th>WindGustDir</th>\n",
          "      <th>WindDir9am</th>\n",
          "      <th>WindDir3pm</th>\n",
          "    </tr>\n",
          "  </thead>\n",
          "  <tbody>\n",
          "    <tr>\n",
          "      <th>361</th>\n",
```

```
"      <td>No</td>\n",
"      <td>NNW</td>\n",
"      <td>SSE</td>\n",
"      <td>NW</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>362</th>\n",
"      <td>No</td>\n",
"      <td>N</td>\n",
"      <td>NNW</td>\n",
"      <td>NNW</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>363</th>\n",
"      <td>No</td>\n",
"      <td>ESE</td>\n",
"      <td>ENE</td>\n",
"      <td>ENE</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>364</th>\n",
"      <td>No</td>\n",
"      <td>NW</td>\n",
"      <td>SSW</td>\n",
"      <td>WNW</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>365</th>\n",
"      <td>No</td>\n",
"      <td>NW</td>\n",
"      <td>NW</td>\n",
"      <td>WNW</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-c1c2629a-fb64-4205-ad8b-
b6e835ad5596')\"\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-c1c2629a-fb64-4205-
ad8b-b6e835ad5596 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-
c1c2629a-fb64-4205-ad8b-b6e835ad5596');\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&lt;/a&gt;'\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",
                "      &lt;/script&gt;\n",
                "    </div>\n",
                "  </div>\n",
                "  "
              ]
            },
            "metadata": {},
            "execution_count": 19
          }
        ]
      },
      {
        "cell_type": "code",
        "source": [
          "data.head()"
        ],
        "metadata": {
          "colab": {
            "base_uri": "https://localhost:8080/",
            "height": 206
          },
          "id": "UKCYYPss66M-",
          "outputId": "b8602e62-3c5e-46b7-9505-072dcd840d6e"
        },
        "execution_count": 20,
        "outputs": [
          {
            "output_type": "execute_result",
            "data": {
              "text/plain": [
                "   MinTemp  MaxTemp  Rainfall  WindGustSpeed  WindSpeed9am
WindSpeed3pm  \\\n",
                "0      8.0     24.3       0.0           30.0            6.0
20   \n",
                "1     14.0     26.9       3.6           39.0            4.0
17   \n",
                "2     13.7     23.4       3.6           85.0            6.0
6   \n",
                "3     13.3     15.5      39.8           54.0           30.0
24   \n",
                "4      7.6     16.1       2.8           50.0           20.0
28   \n",
```

```
      "\n",
      "   Humidity9am  Humidity3pm  Pressure9am  Pressure3pm  
Temp9am  Temp3pm  \\\n",
      "0           68           29       1019.7       1015.0     
14.4     23.6   \n",
      "1           80           36       1012.4       1008.4     
17.5     25.7   \n",
      "2           82           69       1009.5       1007.2     
15.4     20.2   \n",
      "3           62           56       1005.5       1007.0     
13.5     14.1   \n",
      "4           68           49       1018.3       1018.5     
11.1     15.4   \n",
      "\n",
      "   RISK_MM RainTomorrow  \n",
      "0      3.6          Yes  \n",
      "1      3.6          Yes  \n",
      "2     39.8          Yes  \n",
      "3      2.8          Yes  \n",
      "4      0.0           No  "
     ],
     "text/html": [
      "\n",
      "  <div id=\"df-e2fab7af-4ea6-4fb6-ac78-d29b02e00f59\">\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>MinTemp</th>\n",
      "      <th>MaxTemp</th>\n",
      "      <th>Rainfall</th>\n",
      "      <th>WindGustSpeed</th>\n",
      "      <th>WindSpeed9am</th>\n",
      "      <th>WindSpeed3pm</th>\n",
      "      <th>Humidity9am</th>\n",
      "      <th>Humidity3pm</th>\n",
      "      <th>Pressure9am</th>\n",
      "      <th>Pressure3pm</th>\n",
      "      <th>Temp9am</th>\n",
      "      <th>Temp3pm</th>\n",
      "      <th>RISK_MM</th>\n",
      "      <th>RainTomorrow</th>\n",
      "    </tr>\n",
```

```
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>8.0</td>\n",
"      <td>24.3</td>\n",
"      <td>0.0</td>\n",
"      <td>30.0</td>\n",
"      <td>6.0</td>\n",
"      <td>20</td>\n",
"      <td>68</td>\n",
"      <td>29</td>\n",
"      <td>1019.7</td>\n",
"      <td>1015.0</td>\n",
"      <td>14.4</td>\n",
"      <td>23.6</td>\n",
"      <td>3.6</td>\n",
"      <td>Yes</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>14.0</td>\n",
"      <td>26.9</td>\n",
"      <td>3.6</td>\n",
"      <td>39.0</td>\n",
"      <td>4.0</td>\n",
"      <td>17</td>\n",
"      <td>80</td>\n",
"      <td>36</td>\n",
"      <td>1012.4</td>\n",
"      <td>1008.4</td>\n",
"      <td>17.5</td>\n",
"      <td>25.7</td>\n",
"      <td>3.6</td>\n",
"      <td>Yes</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>13.7</td>\n",
"      <td>23.4</td>\n",
"      <td>3.6</td>\n",
"      <td>85.0</td>\n",
"      <td>6.0</td>\n",
"      <td>6</td>\n",
"      <td>82</td>\n",
"      <td>69</td>\n",
"      <td>1009.5</td>\n",
"      <td>1007.2</td>\n",
"      <td>15.4</td>\n",
"      <td>20.2</td>\n",
"      <td>39.8</td>\n",
"      <td>Yes</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>13.3</td>\n",
"      <td>15.5</td>\n",
"      <td>39.8</td>\n",
```

```
              "      <td>54.0</td>\n",
              "      <td>30.0</td>\n",
              "      <td>24</td>\n",
              "      <td>62</td>\n",
              "      <td>56</td>\n",
              "      <td>1005.5</td>\n",
              "      <td>1007.0</td>\n",
              "      <td>13.5</td>\n",
              "      <td>14.1</td>\n",
              "      <td>2.8</td>\n",
              "      <td>Yes</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>4</th>\n",
              "      <td>7.6</td>\n",
              "      <td>16.1</td>\n",
              "      <td>2.8</td>\n",
              "      <td>50.0</td>\n",
              "      <td>20.0</td>\n",
              "      <td>28</td>\n",
              "      <td>68</td>\n",
              "      <td>49</td>\n",
              "      <td>1018.3</td>\n",
              "      <td>1018.5</td>\n",
              "      <td>11.1</td>\n",
              "      <td>15.4</td>\n",
              "      <td>0.0</td>\n",
              "      <td>No</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-e2fab7af-4ea6-4fb6-ac78-
d29b02e00f59')\"\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-e2fab7af-4ea6-4fb6-
ac78-d29b02e00f59 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-
e2fab7af-4ea6-4fb6-ac78-d29b02e00f59');\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": [
        "data=pd.concat([data,data_c],axis=1)"
      ],
      "metadata": {
        "id": "wxsrm9AL7Bll"
      },
      "execution_count": 21,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "data.head()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 270
        },
        "id": "U11jEEtJ7Lp4",
        "outputId": "32419693-9e0e-4163-ba3a-b53d4d92bc67"
      },
      "execution_count": 22,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   MinTemp  MaxTemp  Rainfall  WindGustSpeed  WindSpeed9am
WindSpeed3pm  \\\n",
```

```
              "0      8.0     24.3       0.0            30.0            6.0
20   \n",
              "1     14.0     26.9       3.6            39.0            4.0
17   \n",
              "2     13.7     23.4       3.6            85.0            6.0
6   \n",
              "3     13.3     15.5      39.8            54.0           30.0
24   \n",
              "4      7.6     16.1       2.8            50.0           20.0
28   \n",
              "\n",
              "   Humidity9am  Humidity3pm  Pressure9am  Pressure3pm
Temp9am  Temp3pm  \\\n",
              "0           68           29       1019.7       1015.0
14.4     23.6   \n",
              "1           80           36       1012.4       1008.4
17.5     25.7   \n",
              "2           82           69       1009.5       1007.2
15.4     20.2   \n",
              "3           62           56       1005.5       1007.0
13.5     14.1   \n",
              "4           68           49       1018.3       1018.5
11.1     15.4   \n",
              "\n",
              "   RISK_MM RainTomorrow RainToday WindGustDir WindDir9am
WindDir3pm  \n",
              "0      3.6          Yes        No          NW         SW
NW  \n",
              "1      3.6          Yes       Yes         ENE          E
W  \n",
              "2     39.8          Yes       Yes          NW          N
NNE  \n",
              "3      2.8          Yes       Yes          NW        WNW
W  \n",
              "4      0.0           No       Yes         SSE        SSE
ESE  "
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-dc36f98c-43b7-4332-9442-1a86695e65c5\">\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>MinTemp</th>\n",
"      <th>MaxTemp</th>\n",
"      <th>Rainfall</th>\n",
"      <th>WindGustSpeed</th>\n",
"      <th>WindSpeed9am</th>\n",
"      <th>WindSpeed3pm</th>\n",
"      <th>Humidity9am</th>\n",
"      <th>Humidity3pm</th>\n",
"      <th>Pressure9am</th>\n",
"      <th>Pressure3pm</th>\n",
"      <th>Temp9am</th>\n",
"      <th>Temp3pm</th>\n",
"      <th>RISK_MM</th>\n",
"      <th>RainTomorrow</th>\n",
"      <th>RainToday</th>\n",
"      <th>WindGustDir</th>\n",
"      <th>WindDir9am</th>\n",
"      <th>WindDir3pm</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>8.0</td>\n",
"      <td>24.3</td>\n",
"      <td>0.0</td>\n",
"      <td>30.0</td>\n",
"      <td>6.0</td>\n",
"      <td>20</td>\n",
"      <td>68</td>\n",
"      <td>29</td>\n",
"      <td>1019.7</td>\n",
"      <td>1015.0</td>\n",
"      <td>14.4</td>\n",
"      <td>23.6</td>\n",
"      <td>3.6</td>\n",
"      <td>Yes</td>\n",
"      <td>No</td>\n",
"      <td>NW</td>\n",
"      <td>SW</td>\n",
"      <td>NW</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>14.0</td>\n",
"      <td>26.9</td>\n",
"      <td>3.6</td>\n",
"      <td>39.0</td>\n",
"      <td>4.0</td>\n",
"      <td>17</td>\n",
"      <td>80</td>\n",
"      <td>36</td>\n",
"      <td>1012.4</td>\n",
"      <td>1008.4</td>\n",
"      <td>17.5</td>\n",
"      <td>25.7</td>\n",
"      <td>3.6</td>\n",
```

```
"      <td>Yes</td>\n",
"      <td>Yes</td>\n",
"      <td>ENE</td>\n",
"      <td>E</td>\n",
"      <td>W</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>13.7</td>\n",
"      <td>23.4</td>\n",
"      <td>3.6</td>\n",
"      <td>85.0</td>\n",
"      <td>6.0</td>\n",
"      <td>6</td>\n",
"      <td>82</td>\n",
"      <td>69</td>\n",
"      <td>1009.5</td>\n",
"      <td>1007.2</td>\n",
"      <td>15.4</td>\n",
"      <td>20.2</td>\n",
"      <td>39.8</td>\n",
"      <td>Yes</td>\n",
"      <td>Yes</td>\n",
"      <td>NW</td>\n",
"      <td>N</td>\n",
"      <td>NNE</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>13.3</td>\n",
"      <td>15.5</td>\n",
"      <td>39.8</td>\n",
"      <td>54.0</td>\n",
"      <td>30.0</td>\n",
"      <td>24</td>\n",
"      <td>62</td>\n",
"      <td>56</td>\n",
"      <td>1005.5</td>\n",
"      <td>1007.0</td>\n",
"      <td>13.5</td>\n",
"      <td>14.1</td>\n",
"      <td>2.8</td>\n",
"      <td>Yes</td>\n",
"      <td>Yes</td>\n",
"      <td>NW</td>\n",
"      <td>WNW</td>\n",
"      <td>W</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>7.6</td>\n",
"      <td>16.1</td>\n",
"      <td>2.8</td>\n",
"      <td>50.0</td>\n",
"      <td>20.0</td>\n",
"      <td>28</td>\n",
"      <td>68</td>\n",
"      <td>49</td>\n",
```

```
"      <td>1018.3</td>\n",
"      <td>1018.5</td>\n",
"      <td>11.1</td>\n",
"      <td>15.4</td>\n",
"      <td>0.0</td>\n",
"      <td>No</td>\n",
"      <td>Yes</td>\n",
"      <td>SSE</td>\n",
"      <td>SSE</td>\n",
"      <td>ESE</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-dc36f98c-43b7-4332-9442-
1a86695e65c5')\"\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-dc36f98c-43b7-4332-
9442-1a86695e65c5 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-
dc36f98c-43b7-4332-9442-1a86695e65c5');\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": "markdown",
      "source": [
        "#### 5. Data Visualization"
      ],
      "metadata": {
        "id": "MvvCqBjJzwrn"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "corr=data.corr()"
      ],
      "metadata": {
        "id": "iCGbv5Sa7Ojk"
      },
      "execution_count": 23,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
"sb.heatmap(data=corr,xticklabels=corr.columns.values,yticklabels=corr.co
lumns.values)\n"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 356
        },
        "id": "LOqperlP7ZwP",
        "outputId": "b4e97d3d-b130-45ba-f6b0-6bc7bce1bbd2"
      },
      "execution_count": 24,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at 0x7ff7e0065150>"
            ]
          },
          "metadata": {},
          "execution_count": 24
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "&lt;Figure size 432x288 with 2 Axes&gt;"
```

],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAasAAAFCCAYAAABPWvInAAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOydedzlY/3/ny9jGIwlRL
KNGMnXMhgi6avIV2WrKFpkyVJIim+Kn0SlUok2yZp8s5QsUZYYZJ1hxox9bFkioTC2Mff9+v1
xXcf9mTP3cpbPue9zzzryfHp/HfT7b63Ode27nfa7rel/vl2wTBEEQBO3MAiPPdgCAIgiAYighW
QRAEQdsTwSoIgiBoeyJYBUEQBG1PBKsgCIKg7YlgFQRBELQ9EayCIAiCmpF0uqJN01wHlJO
knSg5KmS9qwwjOdGsAqIAjq4Uxg20HOfwgYn7d9gV+W8dAIVkEQBEHN2L4eeeH6QS3YEfucPELc
BSklZo9rkRrIIgCIIyWRF4vLD/RD7WFAs2KxA0xhvPPlxqnas5N5xfphwA0w+4rVS90aN6StU
DeHr2IqXqTRkzqlQ9gAd4tXTNdVi0VL0f/rvcf2sASaXqPXHrKaXqAfTeOal0zXsPvaN0zQ0f
v7ipX2Y9nzcLvXX1/UjDdxVOsV3+L79OIlgFQRB0O721f1HMgamZ4PQksHJhf6V8rCliGDAIg
qDbcW/tW/NcAuyUAyVzI3BV6w/VSzotGzCoIg6HZ6SwlCAEj6HbAlsByAe9vfuyfbsH5b/vaTIOwNq2vzfA/esCZ+fdVYAX8vas7a1b/gaCIAi
GGffMKU/L3m3Oi2FmY4LVsDwqSFrH9KvBBCuOhti8hdUP7xfYMAKApDOBP9n+fUtb
HARBMJKUM7w3onTqnNXlwEfy692KwA9LW5tTlHSTRdOOjWCSqbTAbUkrk2SaDhKvlHzZ0ZaqkLo8tfUsuFIylUCtkk3Xm3UTAiIDkv/BYIeJGWOWGb3Nkn6g1kzx9C6Ds5hcoSL9Pd5XGDLo5FA4qvuHxNMfyzUGo1s9UZYYZJ1hxox9bFkioTC2Mff9+v1
xXcf9mTP3cpbPue9zzryfHp/HfT7b63Ode27nfa7rel/vl2wTBEEQBO3MAiPdgCAIgiAYighW
QRAEQdsTwSoIgiBoeyJYBUEQBG1PBKsgCIKg7YlgFQRBELQ9EayCIAiCmpF0uqRnJN01wHlJO
knSg5KmS9qwwjOdGsAqIAjq4Uxg20HOfwgYn7d9gV+W8dAIVkEQBEHN2L4eeeH6QS3YEfucPELc
BSklZo9rkRrIIgCIIyWRF4vLD/RD7WFAs2KxA0xhvPPlxqnas5N5xfphwA0w+4rVS90aN6StU
DeHr2IqXqTRkzqlQ9gAd4tXTNdVi0VL0f/rvcf2sASaXqPXHrKaXqAfTeOal0zXsPvaN0zQ0f
v7ipX2Y9nzcLvXX1/UjDdxVOsV3+L79OIlgFQRB0O721f1HMgamZ4PQksHJhf6V8rCliGDAIg
qDbcW/tW/NcAuyeswI3BV6w/VSzotGzCoIg6HZ6SwlCAEj6HbAlsKykJ4BvAqMBvMQMBbJ8MXA58GH
gQeAXYs4zndlywkmTgHNufyfsLAk8Bt9reTeTTIOwNq2vzfA/esCZ+fdVYAX8vas7a1b/gaCIAi
GGffMKU/L3m3Oi2FmY4LVsDwqSFrH9KvBBCuOhti8hdUP7xfYMAKApDOBP9n+fUtb
HARBMJKUM7w3onTqnNXlwEfy692KwA9LW5tTlHSTRdOOjWCSqbTAbUkrk2SaDhKvlHzZ0ZaqkLo8tfUsuFIylUCtkk3Xm3UTAiIDkv/BYIeJGWOWGb3Nkn6g1kzx9C6Ds5hcoSL9Pd5XGDLo5FA4qvuHxNMfyzUGo1s9UZYYZJ1hxox9bFkioTC2Mff9+v1

YfyIfK7ImsKakGyXdkt0rmqKWYcAbgU8AP5G0ALAssETh/HuAQ2w3k/L0beBtwLq2X5O0OPDV
BrW2BGYBlfSmk4ATbF8MIGndJtoZBEHQcXhO7dnHkvYF9i0cOsV2vV/yFwTGkz6PVwKul7Su7
f/UqfMmtQwD3gRsll//F3AX8JKkt0haGHgXsF7Fyj33mE7K5oQPF3pPkvSzPCl3NbBcPr4oqc
t4kO3XAGy/VCmBJGmcpLsqjZF0qKTKuS/lCbzpks6VNA7YHzgk96S2AFagYPVhe0a+dw9JF+d
e10xJ3yw84zOSbssav8pjtEjaJpsv3iHpAklj8/FtJd0n6Q7gYzX+7oMgCIaHOoYBiwW381Yd
qJ4EVi7sr0Rfge8KTwCX2H7D9iPAA6Tg1TBDBivb/wDmSFqF1Iu6GbiVFMAmAjOA2VW3rQC8F
9gO+F4+9lHgncDawO5ZC2AN4DHbLzXQ/sOBDDWyvB+xv+1HgZFJPaoLtG4ATgskVdx+iwsrNi
GZJ64H7CJpoqR3AZ8kTQxOINmFfFrSsqRagFvb3hCYAnxF0hjg1yTLkI1IPcQgCIL2odxhwMn
AeEmrSVoI2BW4sKrMn0lWHiQtSzIv2828hPNZ55rAw828hVoTLG4iBZ3dKsLq5sH9jP9dfZLs3Z4csn4+9D/id
7Z4cAK/p70GS9sw9msclrdzfNQWmA+dI+gzQ76o322eQen8XkH55t+QeIcBVtp+z/SpwISnAb
kUKOpOzmeNWJGv7TUmB9sZ8/HMk76y1gEdsz7Rt4LcDNbboZ3XhrEeHeGtBEAQlUWLquu05wI
HAFSSj2/Nt3y3pGEk75n75ruVJ6Tdm9wLXCY7eeaeQu1pq7fSApM65KGAR8nzSm9CJwBLF11fdH
IUENoPwisImnxPPx3BnBGHvobRQpCxaBaNDz8CCkIbg8cMdB8VA6OpwOnZ91KVfXqrxHO7T3L
9teLJyRtTwpuu1UdnzDE+yu24xTgFIA7Vt6x81fpBUHQGZRcbsn25cDlVceOKKFOKrw28JW8lUI9P
avtgOdzz+h5YCnSUGCtdVqupBz4paZSkFYD3w8lUI9zAgmDE8Zzemrd2pdZyS/+QVO1ptSIpgL3A4J5Wa5G8Tn7
P3J5Wy5PqBJ5u+3lJlwB/l/RX4E8kO5HKb24cyXtqdeDaXEx2d+AF2xvngHmjpCtJBl/j8/UC
LpH0PuBlku/KhPy+7wBuz/p3AzuSPFh2oc9Y7Bngg9m9eDzwO5KHF8D6pCD9PMmn5VTbm0g6m
FST8M1h0yAIghGljXtMtVJPgkXR0+rHpGD1HlKwGtDTCrhH0jyeVsA/JL3paWX789niY2vgUO
CDwB759PlZa6akh0kBsLgtrBqZeoJVKz2tgDct52dIOht4dDv
1EG2ryiekPQ/wHG2f1V1fMCeju37SAGObPvxkXzqEJJFyfqkIdPXCrcV319vYb+XAX6vuTr7v
gAHLj6RDy0So4VBELQed0HPqtY5K2ihp5WksZK2LFw3Afh7YX8SQvkuaB3APeTnCi/UPCqWl
PSYvn4XpLG5uMrZm+q64GdJC0iaXGSYSP5mop31QIk6/qT86klgadyr+6zJFuShrF9iu2Jtid
GoAqCYNjoggSLenpWrfS0EvC/kn5Fenn//AbnUnuUFnGYKUzbTRv/ZrTOScKqu4RtXzBxdzEV+QwAfDdaIHGJ61Sw1wuMF+WBUhvkGcfX7yTaDhaYXi6o/O4M8IUKLAfmCeoMmmDZmaBLUXtdRwCg2OyJfbk/NkTJLjtMeStWZ5YLZPuGhuP/1KuU7/2Er6aVMsE9dvqDAekSQEotVg/N1F8E9yS3OLMp/Ih7p4zoLUNVi9LRlN6N6dHkIbkNiWrAbOfmhMiw7ubgzbCT5fz9v0Ioui9eBKomi7WgBoX8CPzYnMtc+fGQxNv8HKFozwFDJ5JVf20AT+MtXSK7JHX9BIGzTMMl4k3n1HuYg+7RN0QjCH55+k7fYbvqQzdQDoy6CUs4Y6sWhL9xnCxAZJPbRzRn1a9XnWE6gWAKwN6jW4xntFlyScfUWfG5OBGWk1HgB1FBQfdhfQI1bLCAj8bVmGNQOuaK5ACC7VOqlZwTKVN5KpfW2cDR2nPwbAXqMl4tW9kOfkaxNjJkzR8rG1iGgXyKJdfhPzSKwF5Nz5Y8rl5FoxEBEp5wgCJsBu7u8+N6U1T9nmB3xNw9DWA2L/uVP7j8c2wcPA5Sb/n2KjiWUvNNNrL7+3I3zQ1/G8MjxeZRn3Wjlt7jRHnBwI2gCQZ+cmj4jmlS4Z6IbRWqFmnqjF/u/KozkVGVQ5MTRkIk88I0MiPCqyX/sfGrhF0q9ngG8BC3p5Y7NoWqQc2SLyg8BP3dmpBZg5Q2Nr7LG6kJO7QX8+iS5U8jXqQ0A+4K3NyM0fnPcdz2zL5bzuLubZb/RRrZLXqTK/1tFTrLJmdI9BDTlW5EZ4EAyZNZ+xMy3kqm26Rpy6jHJPr+JFdZJNc14KnhPZCqj1xcECWhNzkDiIbKwH3U7aUz2lzgC2Wxa5bjV3MkwLCOHzwqkovjAr0UdgnmRLBh54eBi+UKoqBxQ3E9MWU2qpxEYaFz0HiO5DiTwtf7RqsCk0sNOx9d4ZfKlcNmRKW+kT4lGjQT6JJWPzIoGkFFrs6Q9v9BD94M9gexrqJ7ZxOiiFoklAKZtD+9QSKjqo0S9sKuQj3ywWXhNpDmFU2HhSaMnjIR/Ie4XVpiD8SDn+aUnhLlSsL+38ksDaOZMn1M5ru6LX+pxE8IlyA3NpCTjBVEL3wq8FvDj7pmEtrwdHm8wFWhH0Y9bj+kXCEQi4AH+RE1IRX
gAHLj6RDy0So4VBELQed0HPqtY5K2ihp5WksZK2LFw3Afh7YX8XSQvkuaB3APeTnCi/UPCqWl

349UdJJ+fWWkt5Tg+762QZkhqRLJS0x1D1BEASdgq2at3alo3pWtk+u4ZopwJS8uyWp3NJQtQ
tPBQ61fZ2kvYDDgP/XRFODIAjahm6YsxrWFJFspLhzYX9W/lmTmWE2Qzw0v95I0p2S7gQOKGh
uKelPksaRavwdImmapC0kPVIofLtEYX9NUqFbgKvIVddzj+0GSXfk7T31tDcIgqAd6OldoOat
FiRtq2Si+6Ckw/s5/xVJ9ygZ4P5V0qrNvodWBKtFcnCYlr2mjqnxvvVJweVdwGeBNW1vQur1H
NTP9WeQLELW70/M9qPAycAJ2SrkBlLNvo/kS3YFLrT9BnA3qdo6wC7Ayvn1M8AHbW8IfBIo5i
DX294gCIIRocyq65JGAT8HPkQy0t1N0tpVl00FJtpej2S8+4Nm30MrgtWrBS+pCcBRNd5Xs5m
hpKWApWxXekNn1/iMU4E98+s9SQEPYC/gi5JuBxanz/V4NPBrSTOAC0j/MHW3t9DuN/2sZs56
pMYmB0EQNEevVfNWA5sAD9p+2PZs4Fz6vuwDYPta26/k3Zuje/k3VuAlZp9D8O9UmxO5ZmSFgAWKpyr2
8ywXmzfCIzL3lmjbN+Vj99nexvbGwG/IwUfgEOAf5J6UR0bbW/Rz2r82NXKeEtBEARDUnKCxY
rA44X9J/Kxgdgb+HMTzQeGP1g9CmyUX+9A6rnUje3/AP+R9N586NMDXPoSqadU5DckT65Kr4p
szlgJoEeShg8BlgSeynYknwVGNdLeIAiCkaSenlVxBChv+zb6XEmfIX3RP77Z9zDcwerXwH/
pIjNSCaLjbIn8PM8LzbQ14FLgY9WEizysXOAt5B6UBV2k/QAySH4H/QFsl8An8vtXavXavJ9gZBE
IwIrmcrjADl7ZQquSfpm9eHNMT3ZPUzJW0NHAHskdLmqL01PWK0WJh/0z6TBn/CWxaOP21fH
wSNZgZFi07bN9OGp6r8L/9XP8AsF5VE98L/D73zipaA5k1zqy6v672BkEQtAO1ZvnVyGRgvKT
VSEFqV+BTxQskbQD8CtjW9jNlPLSj1lk1i6SfkjYJBnMlDoIg6CrKLCpse46kQ4nnvMlNRGTwGYxvuvqvypowsqSC/g
i0BbDwYT8sVe+6844sVQ9guXLnSViH8j23Lnt6aumaa27afvX1ekvuWY0EEayCIAi6nJ4IVkE
QBEG7U/ac1UgQwSoIgqDLKTMbcKSIYBUEQdDldEAhi2D0Bv/bQ1uxzCD8uCT7X83fG2GbcSIp6WdDg14X8Ly7T
fzy4EJhoEuqMGZcFBn08/atLjJFf3xcBE+8/f2sTfvlQqQ2E99n00ZEI09cSdAG5ASDBaY
Ao1afxzHqRwJLAkrFWwZTl1tpqL9GGcZUK6R5YsVHmb25K8FrozmJ4FYx3+9bvQTQ4FQG1dgH7MIQdkn
tU2qvxcmq3pFTZaHxzq5BZkgFfRT63d1x1/cEnhNjV2BiOcAkw5PDfhaT1xbbIy5oJo+WRC1zU9GNOMJ9Cll2

Zf76ln2smSLpZ0t05qe6TtWiH+WI6fjZw9gBvY47tz/Rzz7jC6+rnjKu+PgiCYKQYxgSLw4G/
2v6epMPz/teqrnkF2N32TElvB26XdIXt/wwm3A49qyAIgqCFuI7/mmRH+joZZwE7zdMW+wHbM
/PrfwDPAG8dSjgWvg5CtrtfZ6TbEQRB0AzDuH5qedtP5ddPA8sPdrGkTYCFgIeGEo5gFQRB0O
X0uvYeUzFrOXOK7VMK568G3tbPrUcUd2xb0oAPlrQCafrlc3k6aFAiWI0QT9x6ytAX1cGmW36
Dmy+vHhpujt12+jW7vlGeX9Q1wM3Lr1WaHsCRc+DjJbbxF5rFzqPeXpoeAHPmcEDPA6VKfom1
uYZBh/jr4gvAGut9qjQ9gIceuJiVVv9wqZpPPHR5qXrvW39vrv39vkNfWAcf2uV0vvlGedm+3
xrd/L9zPYN7OTAN+AFle+uBzkn6p6QVbD+Vg9EzA1y3BHAZcITtW2ppV8xZdQllByqg1EAF5Q
cqKDdQAeUHKig9UAGlBiooP1ABbR+ogNIDFVBqoCpLbxhT1y8BPpdffw64uPoCSQuRsq5/Y/v
3tQpHsAqCIOhyenDNW5N8D/igpJnA1nkfSRMlnZqv+QTwPmCPgtvFhP7l+ohhwCAIgi5nuBb7
2n4O2Kqf41OAz+fXvyWtk62LCFZBEARdTgkp6SNO2wYrScsAf827bwN6gH/l/U1szy7xWR8gV
cFYCLidVGV9zuB3BUEQdAbtbP1RK207Z2X7uWzcOIFkt3FCZb/kQLUAafHartl65O/0TRAGQR
B0PLZr3tqVtg1W/SFpI0nXSbo9Gx2ukI9PknRCNkG8V9LGki7M9am+na8ZyEhxGWC27UrK1lX
Ax/M9m+QaVlMl3STpnfn4HpIuyrWvHpV0oKSv5OtukbT08P92giAI+icK2Q4vAn4K7Gx7I5JR
4ncK52fbnkjqhV0MHECqPrFHHlKEZKT4C9vvInlhfZFkA7KgstMwqWr7yvn1fcAWtjcAjgK+W
3jeOiSDyI1zO17J190M7F7auw6CIGiSYcwGbBmdFKwWJgWIq3J19COBlQrni0aIaj9t+yvbrJM
PGSvCZx0jRqd+7K3CCpNuAl0jzYwBLAhfkavAnAP9VeN61BbPFF4BLC88f198bkLRv7v1NOfX
8S/u7JAiCoHS6oWfVtgkW/SBSENpsgPO1GCH2a6Ro+2ZgCwBJ2wBr5vPHkoLSRyWNo1Atvp9n
FJ/f7++1uDL89fuua9+/iiAIuop2nouqlU7qWb0OvLVihChpNtSKT/GuKVihChptKT/GuKeavo1UpS0XP65MKmc/
cn5miWBJ/PrPRpvehAEwcgxXH5WraSTglUvaT7p+5LuBKYB76lTYyAjxcPysenApbavycd/AB
yXTR07qRcaBEHwJsNoEdIyOuID2PbRhd339XN+y8LrScxt7rglpGxABjZSPAw4rJ/jN9M3JAg
DmzqOK7ye61wQBMFI0zN0UfO2pyOCVRAEQdA47Zw4USvzTbAKI8UgCOZX2nl4r1bmm2AVBEEw
v1KP+WK7EsFqhOi9c1Kpegu+v3w/oukLl/sH3vtquT4/ABeNfqlUvbEq/3+JZRZavHTNy56eW
qre9m/bsFQ9gKd7ZpWq1zPz1lL1AEaNf3fpmjMWuqx0zS2bvL/zQ1UEqyAIgq4n5qyCIAiCti
eyAYMgCIK2J3pWQRAEQdsT2YAtZJjNF08DJpLqDzDz4A7GG73NnhIAiCESJqA7aQ4TJfzBxie33
b6wGPAQeWrB8EQTBiDFfVdUlLZ5+/mfnnWwa5dglJT0j6WS3abRus+qNF5ovYfjFfI2ARcqan
pDMlnZx1H5C0XT4e5otBEHQMPe6teWuSw4G/2h5PGhk7fJBrjwr1W4k4JVq8wXk7h0BvA0s
FZ+ToVxwCbAR4CTJY3Jx8N8MQiCjmAYC9nuCJyVX58FfKy/iyRtSBo8v1GL5kh0F7NTfRZI2ApYHrqxVuJOCVUvMFys329
4TeDtwL/DJgu75tnttz8xaa+XjTZkvnnb1lPrefRAEQYP02jVvkvaTNLWw7ddAUyvYfia/fhZYvr+QlQA=
NhaQFgB8Bh9bzHto2waIfWma++OaO3SPpXOB/gTOGuKcp88VXz/tW5894BkHQEdTTYyp+TvWH
pKtJSW/VHFGlY0n9PfiLwOW2n0gzL7AtF9M7L7XRScHqTfNF2zW2dLGg2safvuOjRWqdxPN1/M81Sr234wv
94BuK9wzy6SzgJWA95B8sTaoJR3FARBMAyUWRvQ9tYDnZP0nKQVbD8jaQXSiFI12wFbSPoisDiwkKQ6h2
ZYSNIs24PNb3VUsKaqULaqLpwP3kMwXBZwlaYn8+k7gC4V7V4HgNuA5YA9rf
9Wj3fBoIgCEaaYVxndQnwwe0B7+efF87TnTl/nTlcm
Yb4YBEGHMIzllr4HnC9pb+DvwCcAJE0kfdn/fKPXCHRGsgiAIgsYZLosQ288BW/VzfCr9rnOmnPOjKZdcc3R2k3fZ8Z0uiSv9d2gNHnPa
8v9fBOsGjFftL1HSxoTBEEwjES5pSIgqqDtcVRdDxrl3kPvKFXvds8sVQ9gg55yE0nWWuSFUv
UAXnu9XEPHF0eVKgfAnr3Lla655qbl/tt87pE5peoBjCnZyPKWnS4sVQ9aY5S4/9RjStdslqi
6HgRBELQ93VDINoJVEARBlxPmi0i0EQBEHbM1zZgK0kglUQBEGX0w3ZgJ1UyBZJPZKmSbpL0
lsrHx0m6K79eNNuAzMjX/U3S2HxuVkHrn0qKbSbpL0qWS
ARBO2C75q1d6ahgByazRfXAZ4n2YBUczDwT9vr5uv2Bt4oX5uv2Bt4oXiBpK+Ak4EO2/z7I82YAuxb2d2
He8k6LS1o5676rrncTBEEwDAyX+WIr6bRgRgVeRmYMV+jq8APFnZsX1/tgoBQNL7gF8D29l+aIh
nXETyZ0HS6iQrkGerrjmfPkuR3YDf1fEegiAIWk5Pb2/NN2/NW7vSkcFK0ihSzlm7L9OvA1S
+rak8YVzC5MC0E627+vn3mpeBB6XtA6ph3VeP9f8gWTCCLA9fb5WQRAEbUEMAw4/i2Tzxad
1XVF9ieRrLyOB5YGphcGJ57A7iJNDRYK+eSAtVOwB/7Of8c8G9Ju5ki5KMG18ZSKhoanbhrEfraE
IQBEHjxDDg8POq7QnAjrLQ7QnAqiQ7j/7mrLA9y/aFtr9IcgT+cD7VS6oCvImkb9T4zD8BnwUes/3iANe
cB/ycIYYAbZ9ie6LtiR8bO67GxwdBEDRHHbFMpbFSkSjSNjTeKPUbghJTbJTF9fqfmDLxxDeCjC5B6EDt1Dz8hBJ5psBt3ANsqKbhNVK6ZEjUlkhbGH8aHC
t2XIK5I+Atwg6Z+2T6vhWV8DHhjksj+S5ssquAN7e0JsKgiBoEbHOagSxPVXSdFJSFkhXgl
9n1dwHgMtK8UvHe5yVtC1wv6V+2+5v7Kl5/7hDnXwK+DxDGjEEQtBvdsM6qo4KV7bFV+9s
fJx34D/Gao+20/TrKqH+hZRw9wfMvC63H9nH+UOq1IgiAIWkk7Z/nVSkcFqyAIgqB+omfVB
6grTYt8gFtr8zEu0JgiAom3ZOnKiV+T5Y5aAUgSkIgq6lG4JVXSmNsQ3/Buzb7pqd0Mb59X13
Qhvn1/fdijZ289Zp66zmR/btAM1OaGMrNKON7as5v7axa4lgFQRBELQ9EayCIAiCtieCVftzS
gdodkIbW6EZbWxfzfm1jV2L8kRfEARBELQt0bMKgiAI2p4IViPI2IVC
B0PpKWAnYHxlH4XLP9pXbSDBonglWbIuljwHsBA3+z3Z9Lcb2abwM2yZqTbT/dgMZP8/39Uu/
/yJIuHUJvh3r0qrRXAw5i3g+bhjTL1uu0dmbtJao0n28TzcuBW4AZJJPVMihVM1sa9XsKsO31
mn1GNxPBqg3JhpJr0Oc8vJ+krW3364xco+bngaOAa0j/c/xU0jG2T69TakqjbRiAH+afHwPeR
nJ2huRT9s8mtS8CTgMupZPwPsLL1WqVbejsl7Qd8C3iNvi8XBt7RJppjbH+l0bYH+1dbYMk2Yv6f39H+
nf5tUStbueSF1vQyTdB7zL+R9H0gLA3bbf1YTm/cB7bD+X95cBtrL9zjLa3CySptSsptieONSyVjZ
vtf3u5lvXGr1W6bainZJmApvZwC/gS8XjneTM+vRZprkb6IbQ/cQwpcV9qe06jm
/EL0rNqTB4FVgL/nOyvnY2XhiXvwCvB/Eu/9Wd4H9Ixp19xsWTWoEKy2sq/BBDmS+Gj2tNz6K5u6q4Sy8WwBXe6aLWuXt5eYXstLYH9v4TGh/29OONG6Bwq/5a9/jv7I33i+u1bu

aNN9Fql24p2PgS80sT9rdacDRwPHEFJPb9WaNq+D/gm8E1JnyQZxX4/PycYhAhW7cniwL2Sbs
v7GwNTJF0CDQeEB4FbJV1M+h9uR2C6pK9kzR/XqPPDoS9piEOASZIeJg1Trgrs16TmusBngQ/
QNxzmvN8Oeq3SbUU7vw7cJOlW5g6AzSQblKn5VWCNMnt+rdCUtCKwK/BR4N+kv/um56PnByJY
tSdHtUDzobxVuDj/XLweEdvXldaiuXX/Imk8sFY+dJ/t1we7pwZ2Ad5he3aTOq3Sa5VuK9r5K
9J8Z5kJDGVqPkj5Pb9SNSVdR/r/7XxgT/pGNhaStHQZySrdTASrNqQSEMrMvLL9rRKa9iY5sB
wHrA2MKTynoSESSYsCXwFWtb2PpPGS3mn7T0008y5gKeCZJjRaqdcq3Va0c3QLEhjK1HwZmCb
pWsrr+ZWtuSqph7sfc9uDiOaHLLueCFZtiKR9gWNIWVK9lPDHLGkiaex9VeYOgI2my55BGns/
AXg/6ZtiM4vMzwBuBzbL+08CF5AmtxtlKeA+SZOZ+8Om0Xm1svVapduKdv45/11eSknJBiVrX
pS3MilV0/a4srTmRyIbsA1pUebV/cBhVA252P77gDcNrne77Y0kzbC9bvFYg3pTbE+UNNX2Bv
nYnbbXb0Qv3//f/R1vdCizbL1W6bainZIe6V+ysZ50qzTLRtJCpKFpA/c3M7QqacPBzpeQqNP
VRM+qPWlF5tW/bF9Sot7rOaV+pqQDST2hsU3ozZa0CDnrStLqFL5tN0LZ82u2rytjYXV/uk03
rkqv7HbaXq2UxrVIs+xh6az5YdK82kOk0Y3VJO1n+88NSk4hDdFWvoSqcK6MRJ2uJnpWbYikD
UjDYqVlXknairS+469Vmhc2qLcxcC9pyOlYYAngeNu3NKj3QeBI0ofNlcDmwB62JzWilzU3BX
4KvAtYCBgFvGx7iQb1qhdW/zfQyMLqTm3nOswbDH7ThN4Y4Iv0VWq5ATjZ9msNaP2NvmHp7cn
D0rYbTlbK6x23s/1g3l8duMz2WoPfOaDel4GdgReAc4E/2p7VaPvmNyJYtSE5Zf1vzDtkd1YT
mr8lDWfcXdC07b3q1Dnb9mclHWz7xEbbM4D2MsCmpA/YW5odBpU0hZQmfAEwkVTnbU3bX29Qr
yULqzuhnXnd1pakYHU58CFSGbCdm9A8n7Ter1K15FPAUrZ3aUCr1GHpfP9k2xsX9gXcVjzWoO
47SP/eO5LWUn7X9rRmNOcHYhiwPWlF5tXGJVWr2EjS24G9JP2GuYcyGp5wzx8EHyKlXB8jaRV
Jm9i+bah7B8P2g5JG2e4BzpA0lbS+pxFKWVjdHx3Qzp2B9YGptveUtDx9QaZR1rG9dmH/Wkn3
NKhV9rA0pLWNl5NSzU1aEjBZqW5nw6MSth/O6x0XIa2HWxOIYDUEEazak1ZkXt0kaW3bjX4YV
DiZNJT4DlL2XvW4e6NzBL8g9fg+QMqEfAn4A2lBdKO8kifIp0n6AfAUzWUslrWwuhPb+artXk
lz8pKKZ0iVVZrhDkmbVoaOJb2bxmtPHgwsCnyJNCz9AeBzTbZvDKk+ZSVh5V+kALM96fdaV7C
q6lE9ThoK/K7tqBFYAzEM2Ia0KPPqXmB14BFSAGyq0rOkX9r+QqPt6UfvDtsblpwNuCrpQ3U0
qVLAksAvKnMQDeh9c7Dzja5l64R2KhVX/gbpw/arpJp502zv2Ugbs+a9wDuBx/KhVYD7gTl0Y
RVySb3AdNKC/BepKlvWxJed+YIIVvMJ+QNxHhpNXc+ao4DlmXvd1mMD3zGo1q3Ae0iZaxtKei
upwOcGjbYvaA2SxgFL2B7I8qJWnX7/JivU8reZh/72AD4OrAT0AA8Av2wi9X9R4EBSMPkp8Mm
sfx8pUaWhpAhJRzN4Xc1SF+53GxGs2hD1VXNYxfa+OS232WoOSHovMN72GTkYjLXdXy+uFq0D
gaNJwyTFhI1Ge2qfJn0obAScSZojOdL2BQ1ojQX+l74PsNmk9ONfNpKkktPAv0l6n0eRvKI+R
vrwOtj2U/VqdlI7s/ZE0rBfD/CAU0HWhslBhjy0uBCwDvBovUPdks4gJSlcTfqbeZGUVfg14G
LbP22gbeeThukWIfX87gXOA3YA3mb7s/Vq1vn8r9s+rpXP6EQiWLUhks4jzQftbnudHLxusj2
hCc1vkjLN3ml7zZwkcYHtzRvUexB4dyXjrAyU7BO2yrvX2L63QZ2LScvBrwY+Qarefi4pNf5J
29+oU+8vwGVZ5lPAOSRrh52ArW3v2K3tVFpg/CPgP6QvEjcCbwHeAD5r+/EGNHcirV/qBfYnD
S/OIgWGL9i+tA6t6cUvSJJusb2ppIVJw5R12+pImmZ7Qk76eQpYwbbz/p2tHp6sDIm38hkdie
3Y2mwDpuSfUwvH7mxScxppnqqoOb0JvWuBBUt+3xuSJsgPAjZsQufOqv3J+ecCpAK59eoVf2e
PVf9eu7mdwFTgrfn1aqS1QQAfJA3TNqr5tqz3IukLFKRSYFPq1LodWL3w93N94dw9DbZvWuH1
6YP9m7ViK/47xta3RTZge1J6NQdgtm1Lqmg26xX1MMnS4zLmzlhsaJJY0lGk1OA/kILqGZIus
P3tBuRelvRe23+TtAPwfG5bb/52XC/FzLzqRbDNZO11QjtH2f5Xfv0YKaBg+ypJP2lQE+eKGp
Ies31/Pvb3yvBgHRxGSnl/nTR3ulvWfSuN15WcImms7VkurEPM/x++NMh9ZRHDXf0QwaqNkHS
l7W1Ic0F/AVaWdA65mkODmt91Gk46X9KvgKUk7QPsBfy6ieY+lreF8tYMviWZlXMPR8+NrZnM
I8HqC8CvJa1JKm+zd9Z8K/DzBvQcIR/wPnNrm7Zwi6TRSNYwdgElZc1FSp
Y2GkLSA7V7S32Hl2Cjq/FuyfU1O1FjGhUXkOcD+byNts/35AY4/JGmLRjTrpJEvKl1PzFm1EV
Vp26VUcyiOfyuVNNoma15h+6pyWt48SjYMH7X9n7y/FHCh7a6vlyZpQZdoa66+xcVlaI0G9iF
VrriTNCzWk3v+y7mBbFKlUl0zXFVWKWcZvtd23YuNNW/ppr+RElXqLt3UKk1JK3uAOT5J2zkn
UEn6hu3vNtjsriWCVRuh5JJ76EDn3cCKeUl3ksrk9PttzfVnX/3E9pc1gL29G6w3HQRaQHwV
Vn3g8BtwBNZt+a6iMqLXwei3qHKsvX60X+YNPx5hptftF26XkF3EVKG6v3tpqkSSze1SlOp1u
C2th+tOr4XcITt1Rtt6/xADAO2F0sC29F/YKl7xXxmLfoqTRSDS6MeWWfnn2Xb2/+Rue29JzW
hVXE/ficpAFaqzW9PCoAjrVfN+qTFtqfmOZvTgXNtv9gmekjanvRvvhCwmqQJpDVHDXtklaxZ
ZummVml+BbhS0kdsz4SUpk4Kgv3augQFRjrDI7a+DbijBZodlVlEquKwAWmIqVmt64HFC/uLU
8gWG2m9AZ7x36S6di8DZwFrtIMe6QvPksydcTijybb1p3lXg1q/BTYt7L8b+E2T7WuF5lakcl
jrAD8BbgLeUubfULdu0bNqLzpmYlUl+QdJOhn4qe27JS0J3J3ExaeLq0pENt/66JZi5PWmhbYXY
+1i56wJuJBR8h2VqMI61rOgfYglThfM2R1Mu8YfuFqiTFZucQ+tPsHejiIdiIVP9yrtJNkmbQ
+GL10jVt/1XSnqSRg5uAD7iebX5iQhW7UUrVsafCGmyuPp/CknLunEbjrJs7bewvX9+vSepO
sJOStUY/gw0E6x+A9wmqTK8uBOpd9EuehVmktatHW/7psLx30t6XxvoAdwt6VPAqPxF5kD
tmKFNz2ybb0nJNSS+RAryAhUm9rGfyMgW7Qf+y+YVIsGhDlCwIvg8sR/rDbvqPOX8b3Md9Fa4
/Dhxnu5Fv2aX5B1VlQF5GqqpxZvW5RlGyEq+kG19ve2o76WXN99r+W9WxzW3f2A56+f5FgSNI
2aQAVwDfbqZXULampLeQSkIVa1U2ZRXfCs2gMSJYtSFKpYy2d4PlhgbQXJc00T4JeDuwDPB52
080qHcTKaX396Q1OE8C33Odnlk5Zf1HwD+yzlq2n5n8oiFX1oJ+afUQW6GXNecpr9PfsR
HUGwVcbfv9jdw/HJqSjiWtRXyIvuFJu4mlD2Vr5uD8hu038v47gQ+TaiL+cdCbgxgGbFP+WWa

gArA9Q9J3SNl8LwHvazRQZfrzD9q9AZ39gJNI5Xe+7FzZgDREclkT7ZurHiJp2HI0adK80XqI
ZettRqo0/9aq9PglaGDBbdl6FZzWVfVKWtL2C43qtFjzE6SyS7OHvHLkNP9CWvg9U2mh9s2ku
cTtJL3b9uElPacriWDVnkxRKmZ7EXOXMmrImRRAqQrB6sB6pAn2P0n6qe1GKiVge3J+OQvYM3
9T3hW4tU6dB4Btq4eubF8hqSErhgIfJWUW3pE1/yFp8cFvGVa9hUhutgvSlx4PqV5eI3bxZes
VmQXMkHQVKbMQqG/9W4s17wKWIvmClUXZmm9xTlknGUP+zvZBSlXnbwciWA1CBKv2ZAngFfrG
8qHxdVYVZpCG/Qw8ouTKWvdiViWX2AOAFUnrja7K+18lGcud02D7TiIVIi3y036O1UPZ9RBL1
XPyW7pO0pluwlesVXpVXEhzf3+t1jwOmCrpLub+gtfwOrAWaBbnXD4AHJ/1ZisZMwaDEHNWQV
0o2Vr8mzSEsRV9SSAH257WgF5l6OrLpMzCCkuQyi814xR8KDCeVA3jOFIduv9zAx5HLdIrtRp
I2XqdhKS7SbYjMyikv7tBA8ZWaEr6LfA0aX73cGA1268olRa7rpm/9fmBCFZthKT/tf0DST+l
/w+bhodcSlwXVcz+G0Xy+1mliQyu/yaVg9ofOLlw6iXg0sKwSUOo5HqIZepJ2sj27fl3MA/1f
iiWrVel/Qj9/03WWwGlJZqSJtveuNG2DIemUmmpg4EVSDUW78zH30OaGzt7sPvndyJYtRGStr
d9qaTP9XfeDbjHFrT/Rt+6qO3J66JsH1WnzlxZZc1kmVXprloZulIqETTWTZQHKuqSsveuztl
Yo2w3bPNQtl6noFRYucIYkp3L0vX+/bRKU9KPSUN1lzD3kF3Daeat0AwaJ4LVfEKJ66J66JsM
F8n6+xWaXAsm6f9IvaseYDJpGPBE28c3opc19wH2JX0Arp57lyfb3mqIW4dLbwaDVIFwnRUSy
tar4Xl1//20pXhm+gAACcdJREFUSjMvgaim2dT1UjWH+9+n24gEizZC0iWDnW9yzuH13GOZKe
lA0rj52HpFbDecAj0Ea9t+UdKnSZUrDidlSDUcrEiJH5uQMxRtz5S0XBvpbVfQhb4iwZ+hsVJ
GZeu9idJi6AoLkFL4m/r8KFOzzDVgLdTcbuhLgoGIYNVebAY8TioxdCvl1grsb11Uv8ONI8Ro
Je+knYCf2X6jknXXBK/nTCsA8kLjZjRL1SsMe37Qc1fq+JqkO6gzlblsvSp+VHg9B3iUtA6pG
UrTlLQ88F3g7bY/JGltYDPbpzXauLI1B8rQzF8idwPKzuDsKiJYtRdvI2Wa7UayDbiMtBbj7m
aFq9dFNavXAn5F+rC6E7g+zw01O2d1naRvAIvkxIgvApe2kV4FqVAOKU+4N2pD3wq9Tui5nEl
aqH1E3n8AOA9oOFiVrTnAso8DScs+7qTxZR/zBTFn1aZIWpgUtI4HvmX7Zw3qtHJosaWoSQfd
/I11bwrZe8CpbvCPvmy9gu5GpFJYS2bdfwN7NTqRX7Ze1jyY9MH9EvBr0vq3w21fOZKalb+RS
uae5q41Oc32hAbaVbpmvrfUZR/zG9GzajNykPoIKVCNIy2WbaZuWCuHFktD0kAZYMc0qmm7V9
JZpPdt4P5mAkvZegXd24H1lSxScJPlh8rWy+xl+0RJ/0OqK/lZ0pxYw8GqJM3bSEHu5ZxdWFm
wvSnQ6PtuhSbAOwrJTafS5LKP+Y0IVm2EpN+QTNkuJ/Wm7ipBtmVDiyXzcuH1GNJkdFP1ESV9
hLR26yFSkF5N0n62/9wmep+x/VvNXcePypyY7boqjJStVy2ff36YZEB4t6Rmv/iUoVm5/iuko
bXVJd0IvJXGS0y1QhPgjcoLp9qIT0Sgqp0YBmwjlEquVD6057GgbzQtvKBfytDicJDbeoXtLZ
vQuA/YzvaDeX914DI3WMm9BXr72f6VUoHcebD9rZHUq9I+gzTXshqwPqkw7qRmUtfL0JT0BH1
lwxYg+USJtC6qp5EA3QrNrNuSZR/zC9GzaiNsNzUJPhAtGFocDhYFVmpS46VKYMk8TJofaQs9
27/KPxsOIq3Uq2JvYALwsFOJoKVpPlGnDM1RpCUY1T2yRZtoVys0W7nsY74gelZdTtXQ4rklD
S2WTtWCyVGkIZdjmun9SfolsCpwftbeBXgMuBrqr2Jftl5BdzXgINIXiaLJX0PJL2XrZc3NgW
m2X5b0GdKczokDpWMPl6ZKqqDSas2geSJYdTmtHlosi5yqXmEOydOr4UzArHnGIKdte6+R1Cv
o3klKhy6rYGqpellzOmmobj1SSvepwCds91uHcLg0VYKb9HBoBs0TwSpoKyStSJ9R4D+aDVid
gKRbbb+7XfWy5h22N8xZm0/aPq3ZHkgZmpKWtv18o20YLs2geVoyRxIEtSLp61Vp6zcDfyKlL
x/WoOY+SnX7UOJ0SS9Imi6p7m/MZev1w4mSvilpM0kbVrY20gN4SdLXSenllymtORs90pqtCC
oRqNqTSLAIRppdgC0K+8/Z3kDJfuQ6kq1JvRxMGlaClFSyPvAOksvvSVXPGwm9atYlfWB/gL5
hO+f9dtAD+CRp6cNetp+WtArN1W1slWbQpUSwCkYc28U1VifmYz1K/j+NMMd2ZU3LdqQ1PM8B
V0v6v6QQRvoVbMLacHo7BK0WqFHDiZ/IJlPAjxLkxmlrdAMupcYBgxGmrFKBWwBsH0mvJlu32jyR
6+kFSSNIZW1ubpwrpEAWLZeNXcBS5Wg0yq9ij3K70k1HOH/t3c/IVaVcRjHvz8jxcI0iiBIV2
3Sok1ETStXESSBtQ4mWkUtyYg2dCiRRIS5EbFRcugxRAtKrJF0Ug2m1zYdCiRRIS5EbFRcugxRAtKrJF0Ug2m1L7QxIZldTCJMmgzKf
F+97mOjOOc+55z5y3w/OBi/cy48NdzcM55/e+b1ofNVNbpg2Xr6ysb28B+yPiKUnnnASLiWmBf
/tkkXgTmSIMab49264h0gu53FeQtAn4OiI+I+JD+dB+f97DEyZ6Wg4OD1W6H3jv/X+kCYqh3DD8oHsvv7A
adII/WbSaPXuycJlPRc9Wiw9L5VwZxmyf3LnnIpPvTGYRJGnCsVC5oXiaJxlsa4a6cpfX9sS4pcVfKaS4CdT
g4Ch5v/KP2mVt8xD9YzhLzn0PjTpo8L/ezQ18Ce+uyzn16KmRbg/HfT0IUiMzVGpJpBT3BCNEPQYG81jvfZEkb0NS/i1AF+vHhvK+2S8mHhf9SatrZ0zm1BdduWmDeMgvVbg0QPU7NSOpdAUDbf/0fShGCu15Z1Ri0R7MyJ/2MmeTT9Cm8GOTg3IKkLcrAbfTsG3uUZfmCQJg/9BGxsilfsFTAl9MeK61ze45PtElQRXkuBu5V3eqkzLm5IJS4K1zeSzO6k3aWs8adddJa0WJ6mygMyaQI7g0gmhkzkuJ8VJpuhEaNUvtd2QsJFGqmpLeTntiWy5BU9Z+pLqL8sxubr3obfsMVTpeL96sZW9Y7ifjbE+MxWV8gKJYy0jkGdA2xeQ7qasNCKX8Sfn6nFrXOmmyL3kwdgqWcsXHy9RlLFQJmVTTzeLvO6Q3zfWB33TIg4rI9GfLdoXULxVmxHVZRVmxDCG4XLKmgPoYsx8F2hhu1PMz9lNxIcXGwEmfiugz4bMn2d5rCMAJ7P3ukdpPhXUeYdJM2sTvUepedXmoC9HQSGiuWxOlUs3L5hJO9UwlPNV0pnXnZP1x7FbVsnfjBeldRzUDmPc3d5VPZuIrPzJgwj9WOfjkSxfwZ0ytkNsrajmMvBMdp8b6D3bEYhQbJ5mDrEJMt1TAc0vbwoSH4WYZBMmOc6MyJ8vnxvQVcQJltbDLvgtyffEHU4DhD6Jah6ZmtRRzoa7VFNzvGp6jhqohMJCgJ8lWsMzfPmgVm5qrTESUavEqU4jVgcOiTQ2GHrFZ6g8NXYChTd3wh6l4yOtVixlQx3EJeE

```
cB9OfdB4AZJm2rJM7OVcVlZNUoVQUTsYr74/iZPF+bXMUkX+8wzs+ZcVtarLoogIvaSR+AlnS
7wHYvmmVlzLivrlYvAzFbCZWVmZtVb0/cXMDMzuxKXlZmZVc9lZWZm1XNZmZlZ9VxWZmZWvX8
BXA8X9vuQjEwAAAAASUVORK5CYII=\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sb.jointplot(data[\"MinTemp\"],data['Rainfall'])"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 458
        },
        "id": "uW8AAaGv7exF",
        "outputId": "9c2dba4d-cf7e-402b-8ab1-985cebfcb353"
      },
      "execution_count": 25,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<seaborn.axisgrid.JointGrid at 0x7ff7dffad2d0>"
            ]
          },
          "metadata": {},
          "execution_count": 25
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x432 with 3 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAaUAAAGoCAYAAADmTPpwAAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3de5zcdX3v8ddnZvZ+yW
V3czFhc4FwC2hI1ogtWAShkVJQiyCnh2JLT+w5eoKmtVpPTxG1VVtFTfVYgyKIVqFSFZFSKWC
BCsgGw03UhM2FxCR7ySZ7y+7s7HHzPHzuzmd2d2Z3dnd/M7zfzfj4e+9i5/OY3n9/vN/P7zPf6
M+ccIiIifhAqdAAiIiJJSkoiIuIbSkoiIuIbSkoiIuIbSkoiIuIbkUIHkCV1ERSRSRYmKFDsCvV
FISERHfUFISERHfUFISKaBlpzRjZnn7KcP6tO6WZ3xx4NW/v95rlp3
Dw1f15e79SYGZc+5Wf5u397n7PbxGQ73yxU5tSBkHp6CBp/ObAq3k/oYmIeEnVdyKlJBRRdaH
4mkpKIqUkHlPpWnxNJSXJXp5/ZRfil/S+1vtdCKFpZKSiIj4hpKS
iIj4hpKSiIj4htqUxN8SnStEpDQoKYm/xa0OJYm/qQuzSElR9Z2IiPiGklIOaecqEjPzoh
zTwUkRkblRSEhER31BSEhER31BSEhER31BSEhER31BSEhER31BSEhER31BSEhER31BSEhER31
BSEhER31BSEhER31BSEhER31BSEhHvJC7SmK+/Zac0F3qLZY40IauIeEcXaZQZUklJRER8Q0l
JRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8o6iT
0rJTmvM675aIiMxNUc9995sDr/K4f/6vvB33jl/51HR0ktLRGRACnqkpKIiASLkpKIiPiGkpKIi
PiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIFI9QJK9QW1x0inqaIREpMfIG
YLOHPOFTqGaZnZg0DjHFfTCHTmIBy/0XYFRzFuE2i7ZqPTObfJo3UHWiCSUi6YWatzrqXQceS
```

atis4inGbQNsluaU2JRER8Q0lJRER8Y1SSkrbCx2AR7RdwVGM2wTaLsmhkmlTEhER/yulkpKI
iPickpKIiPiGkpKIiPiGkpKIiPhGIJLSpk2bHKA//elPf8Xyl7UiPf9lFIik1NlZjDOYiIhMr
9TOf4FISiIiUhqUlERExDeUlERExDeUlERExDeUlERExDeUlERExDeUlERExDeUlERExDc8T0
pmFjazn5vZ/Yn7q8zsaTPbbWZ3m1m51zGIiEgw5KOkdBPwcsr9TwOfc86dBnQDN+YhBhERCQB
Pk5KZLQd+D/hq4r4BFwPfTSxyJ/A2L2MQkfHicUdbRx9PvtJJW0cf8fiMpmIT8VTE4/V/HvhL
oC5xvwE45pyLJe4fAJale6GZbsQ2AzQ3N3Ndd7GRwOE5lWYhbr1nHprVLC
IWs0OFJQimf/zwrKZnZFUC7c27HbF7vnNvunGtxzrU0NTXlODqR0rS3q38sIQEMDsfZes9O9n
b1FzgySVXK5z8vS0q/DVxpZpcDlUA98AVgvplFEqWl5cBBD2MQkRRHegbHElLS4HCc9t5BVjf
VFigqmYpzpVW96llJyTn3V8655c65lcC7gEecc38IPApcnVjsBuAHXsUgIuMtrq+ksir8m176y
LMSiusoCRSQyXiHGKX0I2GpmuxltY/paAWIQKUkrG2q49Zp1Y4kp2aa0sqGmwJGJjPK6owMAz
rmfAD9J3G4DNubjfUVkvFDI2LR2CWduuZD23kEW1VWysqFGnRzEN/KSlETEP0IhY3VTrdqQxJ
c0zZCIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPi
GkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiG
kpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGZ0nJzCrN7
Gdm9pyZvWRmtyQev8PM9pjZzsTfOq9iEBGRYIl4uO4h4GLnXXJ+ZlQFPmNm/JZ77oHPuux6+t4
iIBJBnSck554C+xN2yxJ/z6v1ERCT4PG1TMrOwme0E2oGHnHNPJ576WzN73cx73sw+Z2YVGV672cx
azay1o6PDyzBFRHwl9fzX2dlZ6HDyytOk5Jwbcc6tA5YDG83sHOCvgDOB1wMLgQ9le0O1251yL
c66lqanJyzBFRHwl9fzX2NhY6HDyKi+975xzx4BHgU3OuUNu1BDwdWBjPmIQERH/87L3XZOZz
U/crgIuBX5pZksTjxnwNuBFr2IQEZFg8bL33VLgTjMLM5r87nhO3W9mj5hZE2DATuDPPPbPIBRE
QCxMved88D56V5/GKv3lNERIJNMzqIiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIh
vKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhv
KCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvKCmJiIhvK
CmJiIhvKCmJiIhvKCmJiIhveJaUzKzSzH5mZs+Z+Z2Utmdkvi8VVm9rSZ7Tazu82s3KsYREQkWL
wsKQ0BFzvnXgesAazaZ2fnAp4HPOPQUREAsSzpORG9SXuliX+HHAx8N3E43cCb3v0N3E43cCb/M
qBhERCRZP25TMLGxmO4F24CHgFeCYcy6WWOQAsCzDazebWauZtXZ0dHgZpogUlNTzX2dnZ6HDySs
Tl55mpSccyPOuXXAcmAjcOYMXrvdOdfinGtpamryLEYRb9JPf81NjYWOpy8ykvvO+fcMeBetR4
I3AfDOLJJ5aDhzMRwwiIuJ/Xva+azKz+YnbcC+YnbVcClwMuMJqerE4vdAPzAqxhERCRYItMvMmtLgT
vNLMxo8rvHOXe/mf0C+I6ZfQL4OL4NqwNEPTYEVxhWiN9QP5LD+r0e1+2GjvJvu5ZvSGq9sEbcdMFDJb
lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0l
JRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJRER8Q0lJ
RER8Q0lJRER8Y1IoQMQEW/E4469Xf0c6RlkcX0lKxtqCIWs0GGJTElJSaQIxeOOB186B186zNZ7dj
I4HKeyLMSt16xj09olSkzia6q+EylCe7v6xxISwOBwnK337GRvV3+BIxOZmpKSSBE60jM4lpC
SBofjtPcOFigikewoKYkUocX1lVSWjf96V5aFWFRXWaCIRLKjpCRShFY21HDrNevGElOyTWll
Q02BIxOZmmcdHczsFOAbwGLAAdudc18ws48C/wPo7FIVKKQiFj09olnLnlQtp7B
1lUp953Egxe9r6LAX/unHvWzOqAHWb2UOK5zznnPuPhe4uUvFDIWN1Uy+qm2kKHIpI1z5KSc+
4QcChxu9fMXgaWefV+IiIiSfHlpUzKzlcB5wNOJh95nZ5nZs+b2e2e1e1mtiDDazebWauZtXZ0dKRbRES
kKKWe/zo7OwsdTl55npTMrBa4F3a4F3i/c64nGDJWSbaUzHqB4bcm6tNNoyXnMKvx/oOZvFkLhYTtIsbLA
SD3/NTY2FjqcvPI0ZlZGaMJ6Vs+SxmAHEdKCO3KZTpxbgjaJAADITpxrcXMW6A
Zry+NKUxd4OvOhVDCIiEixe9r77beB64AUz25CPAdWa2jtFu4nuB93gAN
INitCYJBERSUszOoiIiG8oKYmIiG8oKYmIiG8oKYmIiG8oKYmIiG8oKYmIiG8oKYmIiG8oKYm
IiG94OaODiBS5eNyxt6ufIz2DLK7XhQRl7pSURGRW4nHHgy8dZus9Oxkcjo9dcn3T2iVKTDJr
qr4TkVnZ29U/lpAABofjbL1nJ3u7+gscmQSZkpKIzMqRnsGxhJQ0OBynvXewBQBFJMVBSEpFZW
VxfSWXZ+FNIZVmIRXWVBYpIioGSkojMysqGGm69Zt1Ka1sqClwZBJk6ugwIrMSChmb1i
7hzC0X0t47yKI69b6TuZsyKZnZPzJ6Mb60nHNbch6RiARGRGSsbqpldVMtuvNVOtoUORIjFdSak1L1G
IiIgwTVJyzt2Zr0BERESmq777IVNX312Z84hERKRkTVd995m8RCEiQ0ZZuUqpxzDwPmnNvnn
Pso8HvehSUiIqUo2xkdhswsBOwys5/cBBwGNlhMRkZzKtqRQR0ExRENNDRPRNN0
7pLufc9cBvOeeeAfqAP85LZCIiKyltMLPXXAH9iZgvMbGHqvQvNLNLNTzOxRM/uFmb1kZn9bkwZz5u
lHl9oZg+Z2a7E/wW52gN7ABSp/91icciziQF/7px+7px71szqtmqB1m9hDwbuBh
59ynzOzDwDwIeBD80yfhfGZIB537O3q50jPIIvrNXO3BMt0g2e3AdvM7MvOuf85xk
83sZWAZcBvO4GfoKQkhkhxPuOPBoLQkkhPxuOPBlw6XaY8eY2j2TWuXKDFJIGTV0cE5
Rftm9iZiuB84CngWcWc4TcWc4TWbzazVzFo7z4o7OjqyqyfSuRkra3q38sIcHo5cm33rOTvV39BY5
MZiL1/NfZ2VnocPqq6SU6AZ+BHgI7/4sXi7/4sX1sL3Au83znXk
qakpm7cSKXlhegbHElLS4HCc9t7BAkUks5F6/mtsbsFxMHmV7V7ulO1Hse0nwNe9wwO6zrJyMytjNCF9y
zn3r4mHj5jZ0sTzS4H2Q7zGMTxdJVTvNOjBBHqwmKzYYA7d
4GvOycuzXlqfs4OcbpBuAHM1mv5E887mjr6OPJVzpp6+gjHs94FRPxiZUNNdx6za/2sk
2pZUNNQWOTCQ72ZaU2oCfmNmPgKHxQUgxOSzUS/DVwPvGBmOxOPPmd0I7AOumXHU4jk1
mAdTKGRsWruEM7dcSHvIvq1PtOgiXbpLQ/8Vee+JuWc+4JxnchT3VJlu8rBZKpwfzMLReyu

kkzTPlZKGSsbqrVcZJAyiopOedu8ToQ8ZepGsx1spOJNDZKcmW6aYY+75x7f6bLouty6MVLDe
aSLVX1Si5NV1K6K/Ffl0UvMckG84knGjWYy0Sq6pVcmm5Ghx2J/7oseolRg7lkS1W9kku6HLp
kpAZzyYaqeiWXdDl0EZkTjY2SXMq2S3iVc+5hMzPn3D7go2a2A/gbD2MTkQBQVa/kki6HLiJz
pqpeyZXZXg79euCPvApKRERKU7aDZ59J3OwD/tjMwsC7GL0UhYiISE5MWVIys3oz+ysz+6KZX
Waj3gfsRnPWSRGZy+SzmrhWJHeyGTzbDTwJ/CmjE6oa8Hbn3M6pXigSFHOZkUCzGYjXnHM45x
i98ELxm65NabVz7t3Oua8A1zE6Tul3lZCkmMzlaq260qt4bU9HL7FYrNBh5M10SWk4ecM5NwI
ccM7pEpZSVOZytVZd6VW8tqqpjkgk247SwTfdlr7OzJKXMDegKnHfGL2aeb2n0YkvFduM0HOZ
kUCzGYjXzKxkqu5gmpKScy7snKtP/NU55yIpt5WQSlCyDeXybY9z3W1Pc/m2x3nwpcOBbtyfy
4wEms1AJLfMOf+fTFpaWlxra2uhwxCgraOPy7c9Pqlk8EDAZ4ROlv5mMyPBXF4rJSvrD8iGDR
vcjh07vIylEDJuf+lUVEpOFOuM0HOZkUCzGYjkjpKSzEiptaEEqf0sSLGKZKKkJDNSShf/C9I
YpCDFKjIVtSnJjJVKG0qQ2s+CFKsAM2hTWr9+vduxY0ex9cDLuDHZTsgqMibZhnL+6kZWN9UW
ZUKCYI1BClKsMjMaPCsiwMn2s1R+bT8LUqwyM6U2eFZJSSSDII1BClKsMjOlNni2dNKvyAwF6
YqqQYpVZCpKSiJTCNIYpCDFKpKJqu9ERMQ3PEtKZna7mbWb2Yspj33UzA6a2c7E3+Vevb+IiA
SPlyWlO4BNaR7/nHNuXeLvAQ/fX0REAsazpOSceww46tX6RUSk+BSiTel9ZvZ8onpvQaaFzGy
zmbWaWWtHR0c+4xMRKajU819nZ2ehw8mrfCelLwOnAuuAQ8BnMy3onNvunGtxzrU0NTXlKz4R
mYF43NHW0ceTr3TS1tEX6Otq+Unq+a+xsbHQ4eRVXruEO+eOJG+b2W3A/fl8fxHJHU0CK17Ia
0nJzJam3H078GKmZUXE3/Z29Y8lJBida2/rPTvZ29Vf4MgkyDwrKZnZt4GLgEYzOwDcDFxkZu
sAB+wF3uPV+4uIt4r1go9SWJ4lJefcdWke/ppX7yci+VVqF3yU/NCMDiIyK5oEVrygue9EZFY
0Cax4QUlJRGZNk8BKrqn6TkREfEMlJZEAiscde7v6OdIzyOJ6VZtJ8VBSEgkYDVqVYqbqO5GA
0aBVKWZKSiIBM9WgVZGgU1ISCZjkoNVUGrQqxUJJSSRgNGhVipk6OogEjAatSjFTUhIJoFIYt
Kpu76VJSUlEfEfd3kuX2pRExHfU7b10KSmJiO+o23vpUlISEd9Rt/fSpaQkIr6jbu+lSx0dRM
R31O29dCkpiYgvlUK3d5lM1XciIuIbKimJFBkNOp097bvCU1ISKSIadDp7ft13zjmcc5iVxvF
T9Z1IEdGg09nz677b09FLLBYraAz5pKQkUkQ06HT2/LrvVjXVEYmUTqVW6WypSAbF1I6QHHSa
enLVoNPs+HXfmVnJVN2BSkpS4pLtCJdve5zrbnuay7c9zoMvHSYed4UObVY06HT2tO/8wZzz/
5evpaXFtba2FjoMKUJtHX1cvu3xSb+OH9hyYWDHxyRLfhp0OnN53HdZr3TDhg1ux44dXsRQSB
m337PqOzO7HbgCaHfOnZN4bCFwN7AS2Atc45zr9ioGkelM1Y4Q1KSkQaezp31XeF5W390BbJr
w2IeBh51za4CHE/dFCqaUJv6Mxx1tHX08+UonbR19ga2ilOLmWVJyzj0GHJ3w8FXAnYnbdwJv
8+r9RbJRKu0IxdZ2JsXL0zYlM1sJ3J9SfXfMOTc/cduA7uT9NK/dDGwGaG5u3rBv3z7P4pTSV
gptMMXYdhZwU37ASuD8l3H7C9b7zo1mw4wZ0Tm33TnX4pxraWpqymNkUmqS7Qjnr25kdVNt0S
Uk8O8YHEkv9fzX2NhY6HDyKt9J6YiZLQVI/G/P8/uLlKRSajuTYMt3UroPuCFx+wbgB3l+f5G
SVCptZxJ8XnYJ/zZwEdBoZgeAm4FPAfeY2Y3APuAar95fRE7y+0XzimlWDZkbz5KSc+66DE9d
4tV7ikhmfh2D49fZuaUwNM2QiBSUX2fnlsJQUhKRglLPQEmlpCQiBaWegZJKSUlECko9AyWVr
qdUBNRzSYLM7z0DJb+UlAJOPZekGPi1Z6Dkn6rvAk49l0SkmCgpBZx6LolIMVH1XcAley5NnP
1ZPZckV9RmKfmkklLABaXnki4wF0y6DpPkm0pKAReEnkvqjBFcmdosz9R1mMQjKikVAb9fDyj
Tie2Fg8f0i9vn1GYp+aakJJ7LdGJ7+JftqgryOc22IPmmpCSey3RiG4mj7us+F5Q2SykealMS
zyVPbKltSlsuXsNdT+0bqwpS+4Q/BaHNUoqLkpJ4LnliW7b5fB7+ZTvl4RBm8AcblhM2WFKvq
qBcm6ob90y7eGu2BcknJSXJi1DIOHfZfDr6hth1pI8vPLxrrNR0xpJ6mhfq13euTNXbEVBPSP
E1tSlJ3oRCxqqG2rGEBJoWyQuZejvuP9rPCweP8cvDPfzphatZOq9yVvtfY87yy7nS2r8qKUl
etfdm7mLs5+qhIM1qkK6344Lqcp7df4yPfO+FSe16h44PZr3/NeYs/5xzOOcwK439q5KS5FUQ
uxgHbVaDdPv4nS3LxxISjP4Q2PbILt6xfvmM9r8mAM6/vZ19xGKxQoeRN0pKkldB7GIctBNxu
n18+qK6tCXUcIgZ7X8Nps2/VU11RCKlU6lVOlsqvhDELsZTnYj9WOWYbh87R9qJey85cxHnLp
uf9f7XBMD5Z2YlU3UHKilJAfh9WqSJgljlOHEfr2pMX0KdSUKCYJZ0JVgsCD07WlpaXGtra6H
DkBJVLI37yc4acy2h5mo9JS7rHbZhwwa3Y8cOL2MphIzbr+o7kWkEscoxnVwNgp1qPUHqpSj+
pKQkvpHLWQhyTbMaTM/rEmWhPwOSHwVJSma2F+gFRoCYc66lEHGIf2gWguDz8tpLxVKFKtMrZ
EeHNzvn1ikhCUzd7TpoXbJLlZfdxUv5MxCEdv9cUu878YWpTmiFHBuTyyl1inV6nuR2nRge4a
ZLTmPpvJO9EnPVS1Hjo0pHodqUHPBjM3PAV5xz2wsUh/jEdONfCjE2JpdVRsVa/ZRuu266ZA3
feHIf3QPRnHUXL+XxUaU0RgkKV1K6wDm3Hngr8F4ze9PEBcxss5m1mllrR0dH/iOUvJpq/Euh
xsbkssqoWKuf0m3XFx7exeevXccDWy7MWdIttfFRpXz+K0hJyTl3MPG/3cy+B2wEHpuwzHZgO
4yOU8p7kJJX03W7LkSX7FzO5BC0WSEmytTzLdN2OVxOt6tYuuVnK/X8t379+pI6/+U9KZlZDR
ByzvUmbl8GfCzfcYj/TNXtuhBdsnNZZZSLdRWqS/RUVY/5rFYr1W75qr7z3mLgCTN7DvgZ8CP
n3IMFiENkSrmsMprrugo5U/lUVY+lVq0m3tM0Q1Jwsy0B5KPkkMspdeayrraOPi7f9vikEskD
ORgDNJ0nX+nkutuenvT4dza/gfNXN856u0p8MKymGcpAMzpIQcVicX704iE+dO/zM+qVlq/eb
LmsMkq3rmxPzIVsk5quim42+6hYeyPK3GmckhRMPO74aVvXWEKC7HulFUNvtplUyRVypnIvqu
iK4fjlSxBqs3JJJSUpmL1d/bTuOzqrEkDQe7PBzKblSSaGiSWLXLTdpJbWFtVVEg7BoePjS26
57vlWDMcvX0rtcuhKSuKpqaqnjvQMEs9w8bnpSgDFMJhyuhPzxH132VmLeSCLxDCTtppsBr8m
q9Ry2fOtGI5fvuzp7GN4eJjy8vJCh5IXqr4Tz0xXPbW4vpIfPneQLRevGVc19Ok/eO20JYBi6
PU1VZVcun3345ePsLKhZsqLI860l16mwa/vWL/c0yq1Yjh+eVNatXfqfSfema7HWPIE+ukHX+
aK1y4jHIKWFQv5rdUNRCLT/14K+sXmpmrs39vVP6vedjPtpZepZ937Lj6NLz6yGzjZyy7Xgn7

85ijrDT3vvPPcs88+W2zVd+p9J/k3XfXUWFvFkrpZnZj8OJgym6qz1GXOWFzHgzddyOGe8ds/
2zaXmb4uUzVa8reql1Vqfjx+fmRmxZaQpqSkJJ7Jpt2g0CemXI6Vyaabc7ZdoWfb5jLT16XrQ
JFsU1KVmj+UUkICVd+Jh/w+FiXX8WVTdZZt9dpsY5vN61Kr0ZpqR3vfTSy5Sc5lvVPXr1/vnn
32WS9jKQRV30n++WUSzUyloVxfKTWbqrNsq9dmu+9m87p0pdWVjapSk8JQUhJP+aF6LlPJIdd
jZbKpOptJ9dps912h97nkVqlV36lLuBS1qWYOyPUsCdl0c1ZXaJGpqaQkRW2q0tDGlQ05nSUh
m6ozv1Rp5luJT74qM6CkJL6VixNZsjS0oLqcd6xfjhmEDZbUV3qSILKpOiu16rWZdL5Q8hIlJ
fGlXPWMW9lQwxf/23nsOtLHFx7eNbauM5bU07ywpuQSRCFk26HE7701JT/UpiQFEY872jr6eP
KVTto6+iZNg5OrWaRDIWNVQ+1YQprLumR2JlahLp1XyY0XrObXR3rHHfvZHvPpPktBF4RhO7m
kkpLM2FyrWLL5RZzLnnHtvblZl6qWZie1x+HSeZVcf/4Ktj2yKye9IUuhdFVqs4SrpFQAQf5l
l4vLcmfziziXPeNysS6vL0ce5M/EdFJ7HL5j/fKxhARz7w1ZCtdl2tvZRywWK3QYeaOklGden
9y8louTwFS/iJNy2XU6F+vy8uQX9M/EdJIdSh7YciHnLqvPeOxnc5yy+SwF3crGWiKR0qnUKp
0t9YlczyLgtYlVVl39Q2OxL51XOdajraNvKOvqrGznxMtVz7hcrMvLi9LN5DMR1CrEZIcSyHz
9rNkcp1K5LlOpVN2BklLe5frk5uVJKl19/af/4LWsaKhiXmUZ125s5uP3/4LB4Thffbwt67r8
bK+imsuecdmsa6p96eXJL9vPhBftJ/lOctMd+5kecy+vyOsXpZSQQEkp73J5cvO6kTfdL/gP3
fs8X3/36zk2MMwHZlni8+MA0un2pZcnv2w/E7kuZReik0Cuj70fP0syN2pTyrNctpV43cib6R
d8yIyXD/dk1c13omSD/tN7ugDYuLIh41VU82m6fZnaLvKdzW/ggS0XZjVjdzadF7L9TOS6/SR
fnQQm7geA1U21U15BdyaSpatcrc9vkr3vSoVKSnmWy192mU5S+7r6c/JrMdMv+P6hGHF3sm1g
qm6+qTH4ufvuTKpVJ54f0lWBAVlva7afialmp8gUx1T71Yt2sokxNC+o5scvH/HlMQ+KPZ19D
A8PU15eXuhQ8kJJqQBy1VaSKWn8/NVjnBiOz/mLn6nKqrGugh8+d5AtF69h2yO7MnbznVit5O
dOHtNVoWVKqJedtTjtSffspXUz2tZsPhNTzU6xfP7MT/65bifL1AZ560O/8uUxD4zSKSQBqr6
blp/Gj0yMpXlB9aRqny0Xr3frpKIAABS4SURBVOFfWg/kpBomU5XVUGyEa1uaubt1PzdesJrm
hVVZVSv5ufvudFVomRLqS4eOp338SM9Qzrd1qtkpdh44xqcffHlGVXG5nrE8UxvkFa9dNm65w
eE4R3oGffO98rtVTbWUlZUVOoy8UUlpCnOdSBJg/9F+jvQM0R+NsWJhDasa01eppL6+ujxCdG
SEhpqKsSqYqX6p3/nHG3l8dyfOwV1P7ePQ8dETXy6rYTaubBiLu6Gmgrtb93PFa5dhBtXlkbS
/uJtqK2nr6Btbx6K6mf8yn6pKKtNzs3nNVFVo8bhjX1d/2iRz6Hj6RDsQjc25FJIu1kyzUzy2
q4NrW5rHHf9MVXGxWJyXDh3nSM8gdZURtr5lDT1DI4QMyiOzL1ln+tERnvDTt7IsxPCI4/Jtj
7Ogupx3tizn9EV1nLW0PuP3Q0pHQZKSmW0CvgCEga865z5ViDimM9eJJGsqQvziN73jqlqybW
vZcvEa7m7dz4c2ncWmtUsyxvLAlgtpqqvgq4+3eVoNM7En2oc2nTX2/IqGKj7xtnP46++/OG7
5PV19vO+ffz722Bf/23kz6sE2VRyQvs0mU3XaVK9Jble6KrR43PHIr44wPOLSJpml86rSPt68
cG699TJt+9lL69K+30gctj2yixsvWM2XHt099vjEz0AsFuf7zx0cd6y2XLyGe3cc4NDxwbSXZ
s9WpurAlhULxx5PVun93x+8wILq8qzaIktdckaHUiktWb57dZhZGPg1cClwAHgGuM4594tMr2
lpaXGtra1Zv0cuxl7EYnGe3d/Nns5+6qsi9A4OU1VexsBQjFVNNSyqq6Cjb4iKSIij/VHKwiH
6oyO4OOzp7GdFYzXlYeOV9j6iI45I2Fg2v5p9Xf1sWLmA8pAx4qDnxDD1VWXs3N/NcNzRVFvO
4nlVHOsfpqmunOhInOiIo74iws/2dhMOQXVZmNrKMpbOq6AsHKZ7IMr86jJefLWLc05ppKN3i
MX1FTjiVIQj9A3FGIjGWFBdzkA0RnVFhGgsTu9gjMbachbVVzAcGy1ZLamroGtgmFc6+jhlQS
WRcJij/VHmVZUxFBthXlUZsbhjYXUF4RD0R2MMDcfpG4pxYniEBdXllIdD9A5FORF1REfi1FV
EKIsYbe39nLOsnqFYnFjc0T80QnV5mEjIMIPFdRXEHRzqGaRvKEZ9ZYQXDhzn6z/dB8A71i+n
qizExlULGRyOMRCNY2b86nAvS+aV85p51QzFRjgRjdM7GOU1C6o50jNEdXmYlQ3VDERHeGJ3J
3EHj/2qnQtPX0Q4BBee1sg5S+o51DvIb44P0jsYY1FdBZEQ9AyN0DcYo6munN8cO8Ff3vvC2A
n0K/99PVXlEQ73DBIyY/t/vsKv2/v4u7efy6mNNfRFo5SHIxwbGKamIkx5JERnX5RFdRUsqC6
jvTc6VoJeVl/Ji4d7ONwzyNL6SuoqI1z5pf+adIK/60820j0wTEUkRHVFiLAZvUMjhMxREY5w
9ESUhdXldA8Ms7C6jLKwcbhniHlVESojYTr6hoiOOA50D9A3NMK9Ow7QPRDlH687j8qyMMcHh
plXXcbyeZUsra/ixcM9dA9EWVBVRld/lJqKCPOryjgxPEL3wDD1VRHCZnT2R1lSX0FbRz//Jy
XhffLtr2XTWYvY1dnHoeND1FaGqSkP82TbUVY1VFNdEaG9d4hl8yvpGxqhuz/KmUvrOXtJPfu
ODrDvaD815RGWzq+gu2+YQz2DLJ1XxRlNtezu6qN7YJjB4RFWN9SwKo8973Jwjsl64fPOO889
/fTTmFkxJaaM21+IktJGYLdzrg3AzL4DXASXAVkTEozkYseXul+Td5y5Vo+8+8Nfsq/rBJVt736l
2dQVRbiK4+1jbWvxXXNvSPD4SnK4+rT736l
EuOWsp77lrx8kv9TvOpasvymd+/Kuxx266ZA3fe3Ivbz136VgprGXFPK5pWcG7v/6zlBPCuRw
d6OUf/v1XE2IK8Xf/9suxxz5+1Tl88dFRGOOP3rjCr7w8C5OX1TLH56/gpvx/j3a37uf78
ldz11F7+4Z3ncvhYlIPHTowrDf71753FyIjjkw+efI+I+tl57O/MoIzx84zmAszq0P/Xrc9tSUh
+nqj3Kg+8S457ZeejofuOQ0uk/E+Nx//HpSLO96fTN7O3pZUt/I/n+C1zb0swjvzzvHt/
uPKxqqeO9Fa/ib+04ez5uvWMs/PbabfV0n2P5YG39/9Ws5fHxw3Hvf/Ptr+af/3D12zD/xtnO
47Y828PP9x1lSX86RniH+5r5nx5b/2JXnAHH+4d9/RfdAdCzPCirCoXH7I3Xdl53dyFvO
es1YfCsaqth66Rlpq8Ie393Jtod3j+23JfMq+bcXDvLmM5Zyy/3Pjts/fPjvnd/f9e6nz/7ndP49tP7+
HV7HzddsoZvPLLneQQ689fwYMvHqKjd4iPJQZCJ9e9bH4V33hyD5eevXTcvk/Gnvp5k/Gnvp5ST
73V5vO5KZL1tBUV0FjbQVP/Pow4TAc7/y7/ffsYEF1+/X1+aT1+JD+TyX1/y5Vr+X8
/Gd1fLSvm8UdvXDXpc/fZd67jred4X8oqRC/SYpn7LhKJTDsYuBAlpauBTc65P03cvx54g3Pu
fZleM5OSUlthHH5dve3zSL8yZVEk892o3125/atI6JlaNbH7Takbi8LUn2rjxgtV87YnJVWipy

ySfe++bT0u77Feu3zB2Ik3acslpbH9s8rJ/f/Xr+MtEkgPYdt154+5P9drNb1rNtod3T9o2OB
lnuvUll0tu7wWnNfBkW1fW7/GZq1/Hr9t7My5/+qI6/iLNe37m6telfTwZy+3vfj1/csczY/c
n7ptM+zv1eGbaVxOP+fbrN/D0nm7esGoBmyccq+QxvOH2ZybFmGmff+nR3dz5J68fd9zf++bT
iITgK1nEs/lNq9mwYsGkz03qe//91a9jy7d/Pu71qc9n2r+b37Sa0xbVZfwcpH5e0r3v5jet5
vzVDTw1zWckm+OTen/bdeexO8PnaLZVjzORi3MMMygpLWg+3V3yl1+ZYZT+Ex+Jcff/uihZ2v
NVSSkrZrYZ2AzQ3Nyc9etyMfYiU+N1aoIfHI4Td2B28rl0r0ld5uS2pV+2u3940uNxl37ZE9H
YuMdPDMWyfu3Ejk6p25ZcPt36UrfTDDp6h2b0Hv3R2JTL92d4z/7o1LF09g6Nuz8x9kz7O/V4
Zopr4jHvHhgmHIJjJyYfq8HhOMcGhtPGONU+n3jczeCe1gNjXe6Tv8b/7xVn88VHdo9bR9xNf
v3E9z4RjU16z/GxuYzHZKrPQfJ2pveNO+jqm/4zks3xSb1/Yijz5ygXcxFOx8t5EJMmnv++/Z
4LcrLeQstmYtlCJKWDwCkp95cnHhvHObcd2A6jJaVsV56LsReZGq9TC5WVZSFCxtgg0uRjE1+
TuszE5ybeX1hTNunxsKVfdmKPt+qKyT3gMr12Yg1DctssZfl060sul/zfVFfBKx19Wb9HTXlk
yphqKtO/Z02G3n3JWJrqKsYdg0yxT3U8M8U18ZgvqC7jjasbiMXTd3yYX1026fVT7XMg7XHvH
ohy11P7uPGC1ZhByKB/cHisZ13qfkv3+tT3riqPTHrP1OcXVJdnPCZTfQ5sin2WfH1DbQXh9u
k/I9ns++T96orMn6N8TMSaj0lgJ57/iqgtaVqFGKf0DLDGzFaZWTnwLuC+XK08F2Mv1i6t5xN
vO2fcOm65ci33P39w7P4Hf/cMltSfHESa/J/6mg+85XQaa8onPffD5w5y8++vHbfslovX8M2n
9nDLleMfX9lYw19cdsa4x266ZA23PfYKN11ycp13/rSNj105PuaVDTV88HfPSBtT6mMfv+oc7
n/+IPfuODC2ztsee2VSLFsuXsP9zx/kLy47g/ufP0g47Di1qXZcHJVlo21KDdXj32PrpaczMD
RMQ005Wy89fdL2NNaUEzImPbf10tMZiA7zgbecnjaWmy5Zw788s5+PXbl2bD/f+dM2br5i7bj
9/bEJ23Lz748/nqcuqp303hOX+fhV5zDi4nz9v15haHhk0v752JVr+dZTeybFuPXS0yftj9R1
f/OpPeOO3Q+fO8jHrzqH7oEoX3p0N199vI3aiggVkfCk/baqsYZvPrVn3PamvvfNv7+Wrz72y
tjy//rsgXHPf+yqc7jrqbZJ+/emS9ZwWlMtd/508nPJ2FM/L+mOy6lNNXz/2f2cumjyZ+TDm8
5kVUMNlWWhtOtJfibTff/u/Glb2s/dZ9+Zn4lYcz2+S8bLe5sSgJldDnye0S7htzvn/naq5Wf
b+24u0/gkx3IcOj5IU20FFZEQxweHicbinLKgmkjY6OwbojwSors/SnkkTDQ2Qm1FGUOxODXl
YXqHhqkuizA0EmckHqe2oozOviEqy8JUlYXGet/Nqyqjd2iYxpoK6qsiHDo+RFeiN1NFJMzhn
kHmV5XRH41RW1FG2KCjN0p9dYQQRvdAlIbaCuZVhejuH6G9d4hFdRU4S/S+i8YYGBphQXXZWO
+74VicnsEYjTXlLJo32vuuo2+QRbUVdJ8Y5vDxQVYsrGYwFqerb4j6RO+7+soyRpwbG0M1NBT
j5fZeeodGe+GtbKgiOuLo6B2kvrKc7oEoVeVhasojREfiVIRDMZGxve+CxvG+N53/YMj1FVF
qCkPMzA8wonhGDXlZRztj9JQU040NkJ5JExVeZjewRgLayL0nBih+8Qw8yrLgDjhUJjOviEaa
yuoKg8xMDS6b5rqKqipCHO0fzjRs66CtYvrTva+G4rRVFtBWQh6h0boYHYzRUFvO4rpyls2vYX
/3AD0nopRHQvQOjtCRWOfCmjDGaA+3snCIvqFh6irLMKB3cJjySJjjJ4Zpqq1gYc1o77uBaIz
mlN53yd5caxfXsauzj8M9Q9RWhKmMhDk2GGVeZTlH+6NUlIWYV1VG2IyjA9HEGKMwwaizK8q
p/vEyd53R3qi1FWGqSoL09k32lOzP9Eb86zF9Rw4foKu/iGM0c90bUWE18yr5DUTet8d7Y9Sn
eh9Nxgb4Wj/yd53XYnj0h+NYWbUVUQYjo9QGYlw1uJ6ftNzgiM9QwxEY2Pfn2MDUWJx6OwbYk
l9JbH4aI/Q5oU1rFhYzb6jA+w/2k91Su+7wz2DLJlXyRlNdeN6361qqMnrvHc5OMdkvfBMz38
BkXH7C5KUZqpID4qIlC4lpQw0zZCIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiGkpKIiPiG
kpKIiPiGkpKIiPiGkpKIiPhGIGZ0MLMOYN8cV9MIdOYgHL/RdgVHMW4TaLtmo9M5tymbBc3sw
WyXLQaBSEq5YGatzrmWQseRa9qu4CjGbQJtl+SWqu9ERMQ3lJRERMQ3SikpbS90AB7RdgVHMW
4TaLskh0qmTUlERPyvlEpKIiLic0pKIiLiGyWVlMzso2Z20Mx2Jv4uL3RMs2Vmm8zsV2a228w
+XOh4csXM9przC4njE9jLbZrZ7WbWbmYvpjy20MweMrNdif8LChnjbGTYrkB/r8zsFDN71Mx+
YWYvmdlNiccDf7yCqKSSUsLnnHPrEn8PFDqY2TCzMPAl4K3cA
Uwc8Phh4GHn3Brg4cT9oLmDydsFwf5exYA/d86dDZwTFw5exYA/d86dDZwPvDfxfSqG4xU4pZiUisFGYLdzrs05Fw
W+A1xV4JgkhXPuMeDohDoDhIevAu5M3L4TeFteg8qBDNsVaM65w865nuX8g8qBDNsVaM65Q865ZxO3e4GXgWUUwfEKolJMSu8
zs+cT1RBBLY4vA15NuX8g8VgxcMCPzWyHmW0udDA5ttg5dyhx+zCwuJDB5FgxfK8ws5XAecDT
FPfx8q2iS0pm9h9m9mKav6uALwOnAuuAQ8BnCxqspHOBc249o1WT7zWzNxU6IC+40bEYxTIe
yi+V2ZWC9wLvN8515P6XJEdL1+LFDqAXHPOvSWb5czsNuB+j8PxykHglJT7yxOPBZ5z7mDif7
uZfY/RqsrHChtVzhwxwxs6XOuUNmthRoL3RAueCcO5K8+9fEw0V5vPyu6Ep
KU0l8sJLeDryYaVmfewZYY2arzKwceBdwXYA4FjmjMzqzGzuuRt4/TwQBm53zv1tgUOaM
Z4L+vTIzA74GvOycuzXlqcAfryAqmZoSwDfMbB2jzVh7gb8sbDiz45yLmdn7gB8BYeB259y
zNbDXwvcTCP/HNQt7CNs53oa8W84D3SmS+0iF8yu1gUOaM53zv1tgUOaM
zNbDXwvcTcC/HNQt8vMvg1cxOjAloeBzwD3A3cB
DEEw9/hNF2pUAfryAqqaQkIiL+VlLVdyIi4m4DSiI4htKSiI4htKSiI4htKSiI4htKSiI4htKShJ
YZubM7Jsp9yNm1mFm9yfuXznVDOpmdm7KzNZHzWxP4vZ/5CN+EZlMXcIlsMysD9gLM
3gp8EjjjgnLtihuu6A7jfOffd3EcqItlSSUmC7gZHg9xK3rwO+nXzCzN5tZ9l9M3L7DzLaZ2U/Nr
M3Mrs60QjO7zMyexfEnOiJa/19MnktZ7MbL2Z/buzMfpS43tU/mZ
m+ZyJZ0pdFgu47wLvMrBJ4LaOj8DNZClwAXCXKt0CZtYI/DXwk4K3Cup4Df8A
dwNWMXoPnlpRlNgL/m9FrXZ0KvGPGWyVSoopuQlYpLc655xOXG7iO0VLTVL7vnIsDvzCzTJch
OJ/RZPJfo1OiUQ48mfJ8co7BF4DaxPV3es1syMzmJ577mXcAum5bkAULWgSBaUlKQY3Ad8h
tE52BqmWG4o5bZlWMaAh5xz102zjviE9cU5+X2a2FCrhluRLKn6TorB7cAtzrkcNzbrkXKrcrCup4DfNr
PTYGzm8tNnuI6NiRncQ8C1wBM5iEkJkJEuKJkJCgpSeA55w455w5w4457blaF0dwLuBb5vZ84xW3W0U84xW3W0U8A3y

```
R0ctq7+HkzOciMg11CRfJITO7CPiLmXZJF5FRKimJiIhvqKQkIiK+oZKSiIj4hpKSiIj4hpKS
iIj4hpKSiIj4hpKSiIj4xv8H11oI/lnuBOEAAAAASUVORK5CYII=\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "data.boxplot()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 283
        },
        "id": "wj7oQxrU7k8T",
        "outputId": "01c4978a-ebc6-4b68-96b5-7ea8c1a7b5d3"
      },
      "execution_count": 26,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at 0x7ff7df556e90>"
            ]
          },
          "metadata": {},
          "execution_count": 26
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x288 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYYAAAD5CAYAAAAjg5JFAAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3de5xVdb3/8dcHUEDGHB
UlFRI7Wl7o5nDULqcYTbxU4jmYp8lTKOYFFS004aSV1uGknmNmmvjLMLEEMusokgoGQ2apKV6
BMSNTxAuignIdYPj8/vh+Nuw1zgzMvswMm/fz8diP2ev6+rrX27G3ujoiISE63zi6A
iIh0LUoMIiKSocQgIiIZSgwiIpKhxCAiIhk9OrsAbenbt68PHDiwXcusWrWKPn36lKdAFRqnk
upSaXEqqS6VFqcr12Xu3LlLvuPseBQd19y77qqmp8faqr69v9zKfqKQ41lVSXSotTSXWptDhduS
7AY15E26tbSSIikqHEICIiGUoMIiKSocQgQiIZSgwiIpKhxCAiIhlKDCIikqHEICIiGV36P59
FpHzMrNVprt9p2a7pikFkj05X/n677jp2eGS4lM9vEXPjqeE3nPz6nb5snbsL9e/zS8EkORn
uN+1OH6X3jvw1HeHFhwzP9EMHPc7Xrji
cwWvSzqGEoPIdmTbntDgdfdfdfVCM2w34FTAQeAE42d2XWbo2vBY4Hlg9nXKeSJx4bEPQs
mSFQ1XZIZXLfgDbz/0K9a/uZgddu/PLh//d/Oc/JnMPLv03qHdcdqbgApJPlI+W3PFcAtwP
r3rhxwCx3v8LMsXwWOA44IB4HGK9/8x8v0d8K7KMaO+c7x6bfrtWF/8x7vzrBBmhPnUOTmv59/TP
fu3bn55ps59Z53uPn49/DlL3+Diy=plain
```

thDrTOBMgH79+tVMnTq1XRVauXIlVVVV7VqmEJUUp5LqUmlxyhmjtra21Wn19fXbXJx82/q+K
TZObW3tXHcfXHDQrfnRBtKzhHl5w8ubTV8Wf6cDn8obP4t0++gi4NK88d8GLtpSXP1Qj35wZH
uPU84YgM+cOTMTZ+bMmZ6ahdLGmTZtWibOtGnTSh4n37a+b4qNQyf9UM+SuEVE/H09xr8MDMi
br3+Ma228iHSS/v37M2LECOrr69mwYQP19fWMGDGC/v37lzzWKaeckolzyimnlDyGlE6hiWEa
MCLejwDuyhv/VUuOAN5291eBGcBQM9vVzHYFhsY4EekkV111FRs2bGDkyJEcc8wxjBw5kg0bN
nDVVaW97z906FBWrFjB8OHDGTp0KMOHD2fFihUMHaqPp3ZVW0wMZjYFeAj4oJktNrPTGsuA8
3sb8BnYxjgHuB5YCFwE3AOgKeHzt8HHo3X92KciHSSuro6rr322k0/NN+nTx+uvfbaUQ88zd
jxgyGDh3K8uXLcXeWL1/O0KFDmTFDfcOams+ldTaUXJUC/M6cG4r67kZuLLdpRORsqqrq6Ou
ro45c+YwZMiQssXXYFFyx5HS0HcliYhIhKDiIhkKDGIiEiEiGEoOIiGQoMYiISYSg4iIZCgxi
IhIhhKDiIhkKDGIiEiGEoOIiGQoMYiISIYSg4iIZCgxiIhIhhKDiIhkKDGIiEiGEoOIiGQoMY
iISIYSg4iIZCgxiIhIhhKDiIhkKDGIiEiGEoOIiGQoMYiISIYSg4iIZCgxiIhIhhKDiIhkKDG
IiEiGEoOIiGQoMYiISEZRicHMvmFm881snZrplNMbNeZrafmT1iZgvN7FdmtmPM2zOF8F8b0gaWo
gIiIlFbBicHM9gHOBwa7H4ghHB6LnA4si/h1gGnB6LnA4si/HXxHwiItLFFHsrqfQfQ28x6A
DsBrwDHAHfH/JuA4+PjUfE9MX+ImVmRyxcRkRIrODG4+xLgCuBlQkJYCywC1rv7hthsMbBPf
LwP8Eqcd0Nsv2eyXzMbY2aNZta4Zs2aQsMTkTbkcjkmTJhAbW1tuUPBdZQkL9FHcrqfQfQ28x6A
+nYv4+GbUJ+QXCAJ06ADAsBrwDHAHOBC4CH46oAMxsAB42d2d2rYx4Dlrl7t
ZlNB65w9wdj2ixg/1my9Z5KuKOjXr1/XFVWqi+Wd1WulStXUlVVVVCdtttc4ldWvB1/WrVjXtZDD
hdu5S61tbVz3X1wwUHdvaAXMBj7ABwew9cC3weWN5tvAWcMDtvXrqTtSotTXptD
MISopTiXVpdLiVFJdKi1OV64L8JgX2La7e1HPGBYDi939kRi+AzKRRR/X4pbv
H+NERKQLKTgxuPtrwCtm9sEYNQRYAMwGRsex0cCd8fVd8Zw3N3ycdcAbz49NTPzxoW3H6Ldh7+T4A9RHKAw
wZuM7MdgeeB00jJ5nYzOx14kBYsgFG4CTKIG1WF0eUOsl6a2I7WFx6N2ioaZWR2yA0RNXQ5R84DC7B14oKKi+rGHY3dVC3rOOWjNtx
EpP/3ns4iIZCgxiIhIhhKDiIhkKDGIiEiGEoOIiGQoMYiISIYSg4iIZCgxiIhIhhKDiIhkKDG
IiEiGEoOIiGQoMYiISIYSg4iIZCgxiIhIhhKDiIhkKDGIiEiGEoOIiGQoMYiISIYSg4iIZCgx
iIhIhhKDiIhkKDGIiEiGEoOIiGQoMYiISIYSg4iIZCgxiIhIhhKDiIhkKDGIiEiGEoOIiGQUn
RjMrLuZPW5md8fhAWb2iJk9a2a3mdmOcXzPOLww5vcvdtkiIlJ6pSgxXAA8lRr+ETDR3Q8AlgG
g9xp8OLIvx18R8IiLSxRR/KIvSxRSVGMysH3Af4SRgcSYz4FIrqvx2oOnAH4AsYSRHAhK2qtbz
dwA+AnYGLgFOBHaer7oPMMbBPb8VqxjPbAQrtpcUtt/dDcB0F5c49eA2eDOOBEBYsa7lCMoDZ7iAAI8PTqhZ9+qXgc+Ln9c0nSzB/mgnUxiEeSQQR4aDAmJBSAb3t9ERyCU2SN2WA
Aid6+O8QYsc/dqM5sOXOHuD8a0WcWBYd3+s2brOJF1R0K9fv5qqKqqoGm5fv5qpU6e2qywraxuqkqqqqkptZ3E
qqS6VFqeS6lJpcbpyXWpra+e6++l+CCg7p7US+gCpgL/FsML282fVn8nQ58Km/8LGBwW+uuqanx
9qqvr2/3MoWopDiVVJdKi1NJdam0OF25LsBjXkS7iyj+vh7jX
wYG5C3eP8aJiEgXUsynkgyCDS4+w/zJk0DRsT7EcBdeeGp9O9Gp9OgJ4291fLTS+iIUR48ilv
0k8BXgGTN7MsZ9zC6GdeBE6OafcAxwMLgdXEFhGRik4MXlNt3MTgYzEbgG9fQmLFTdxoUIAeq7ZWMu6ELnFhwm6hGtjRrK4iDjTQ6rb6akBH1jVjm7z2xoqu
QeCIi0jH0n88iIKhxCAiIhlKDCIikqHEICIiGUoMIiKSocQgIiIZSgwiIpKhxCAiIhlKDCIikqHEICIiGUoMIiKSocQgIiIZSgwiIpKh
kqHEICIiGUoM27EpU6ZqaWdbcNEx1pbjkKM1NeT9pFNDMmqKmqNFPMVBUvFVRDnv/hXlpy8/vsGJQUrRZJwaTu3T4BgZ0NGDrG3gjFbKZNUlYrD8c3q3
Ynu3btz+umnA1BXV3JfZNjZZNjjjmG+++H3fzDj66HPj66KOZMWNGZxdLthO6Yth
uPHj2ora1l4sSJB/vrOLtt075phjmDlzzXV1QBUV1czcczc+ZMmE4umWwvdAjD
Xv/41xx13HI2/zmN5uu5k445QR5444/k3//zmGGH/5/43HTu6+/Ycb+6Tq+k+ycLE4JM3aQHAARM9
zpxZkvWLbImuGLT1dXVVXXXOfL8yzcm1gqJC5uCHkVzM6jMnqVcz3OTJ
Zku4t0/j7GFcZaSTkgkgT/V0/0zOXNS08u6fpfzkqYRFVRTMuwgTSdvS/5Wc/mSyZMWaV3OTJ
0k8BXgGTN7MsZ9zC6GdeBE6OafcAxwMLgdXEFhGRik4MXlNt3MTgYzEbgG9fQmLFTdxoUIA
QeCIi0jH0n88iIpKhxCAiIhlKDCIikqHEICIiGUoMIiKSocQgIiIZSgwiIpKhxCAiIhlKDCIik
kqHEICIiGUoMzmEpU6ZqaWdbcNEx1pbjkKM1NeT9pFNDMmqKmqNFPMVBUvFVRDnv/hXlpy8/vsGJQUrRZJwaTu3T4BgZ0NGDrG3gjFbKZNUlYrD8c3q3
Ynu3btz+umnA1BXV3JfZNjjjmG+++H3fzDj66HPj66KOZMWNGZxdLthO6YthOjO6YthOjR8/nokTJ1JbW0
uPHj2ora1l4sSJB/vrOLtt075phjmDlzzXV1QBUV1czcczc+ZMmE4umWwvdAjD
Xv/41xx13HI2/zmN5uu5k445QR5444/k3//zmGGH/5/43HTu6+/Ycb+6Tq+k+ycLE4JM3aQHAARM9
zpxZkvWLbImuGLT1dXVVXXXOfL8yzcm1gqJC5uCHkVzM6jMnqVcz3OTJ
08u6fpfzkqYRFVRTMuwgTSdvS/5Wc/mSyZMWaV3OTJ

0aRf/zpzJq1KhN47WfpCPoimE794lPfIJvfOMbXHPNNfz5z3/u7OJ0OR+5fCZvr1nf6vSB437
3rnG79N6Bp747tOCYN910EyNHjmTChAnABCYAvXr14qabbip4nSLtocSwnRs+fDhVVVUMHz5c
iaEFGwdeyM7tXQaAZwqOmfu48Pjx45m/oIFDDj6ISy65RB8jlg6jxLCdu/DCC981blu9jz1ly
hTGjx9PQ0MDBx1UmsZ0RcMVvHDF51qcNmfOHIYMGfKu8S1dRbRXXV0ddXV1DBz3O+a1El+kXJ
QY5F221aRw1llnsXbtWjZu3Mhzzz3HWWedBRT/D3ttNvT3tXwrSWSblv9xu672qqmp8faqr69
v9zKFKHecyZMn+yGHHOLdunXzQw45xCdPnlzS9Q8YMMABN7PM3wEDBpQ0jnv56+Luvttuu3n3
7t396quv9nvvvdevvvvpq7969u++2224lj5Wz79jpZVt3R8Zwr5zzpiPjdOU2AHjMi2h7dcXQB
XXE11UsWrSI973vfbz00ktA6iAMGDCARYsWlWT9OeXsyed76623uOqqqxgzZgxz5sxhzJgxND
U1cfHFF5csRs7o0aO56aabaGxspNePenLGWdw3XXXFbXOth5yt3bFUuxDbum6Ovsra5QYuqD
8r6vI3ceeOHEio0ePLuqgaN74dPvyBPZtNk/zRqjYxue8885jxYoVm4bXr1/P+vXrOe+880p+
gA8aNKjN4V4VIYPXo0N954I1deeSU/WrwvX+//ImPHjgUoKjm8vWZ9i88yWnuOAaV5liFdU7nag
K2lxNAFNTQ0sHjxXgYNGrTpQerYsWOL/rqKzmh8WvuHuHXL8I93xxx9f8nU2d/311wObH9pfmD
e+mMSw80Hj+NCkcS1PnNTaMgB6MF2JytUGbC0lhnboiO/JAdh777057bTTaGpqAmD+/Pmcdtp
p7L333kWtt7Mbn+9973vfvvKcm6mjvvvPO44YYb2GOPPViy5HX69duTpUuXcs4555Q81qpV
q9hpp502JdTVq1dv+qe0QrX26SddMXQtbbUBULp2oFxtwNZSYmiH/J0Jdd/lvf3Rcd2xmz55NU1Mu3zGmZ0xq2t+21T
QdETlNTE6+99lpR613lRcEW7lynVJ2xmz55NU1MTs2fP5sgjjjyzJOvPleupvpn8Cc5cuXc8455x
R977+5nj17cuONNZmzJmZJhN42688UZ69uxZ9Lpbbbehb+OQT6NNPnWFbbwO2lhLDFrT3oWApHgh
u2LChXeO3VmsHcTkP8Jy6ujqWLFlCv379Srbb+2bPsfy3q8fu2nwbuDuvH1Uin1zxhzhbHqm
cPDBB/PDH/6QvcWPHcvbZXe13s7cN5WiXFf0nfHBgHK1AVurIhJDOW/rSxRblJ1Ucx/DuMNOWH/i8jghu
szfUmjvM5NS3HrvBfv2vR15vHSWdffxkxpOWWk0wmyryZ/2qRkJzu+lMAJ8rBHQsUCkZYLswM/ycrCmKTm6bjrSVRJ3ioFBUfmOEZdOZmmv+d/ZGFlp/z0vv//q/YFBjrcOOMBA4JjiTKlrZrlKLYe9//AAblz/w5+gKamR1Iu+mAxJP7DM8TKAAlvPacrFcI0HgRlm5FWSJgh8qtGqAoc2qSM8SqhuxdOnnOaxehXADmY0/kTzqSKZJpEQTNuSkqhhO5VDtNLdxStTgvn/cxfy+3Z8x1TjNDKTNhGqShALt9nbKVy6Eb9xi/+oAXGGOq1WEY9Kmu7/ZDfE2/hnvmHyivBGxtoDLs/NwzTrWNN3KjGwk1SPq5kPqKpbaHTpZq/pPf9z7Y/mCpfKT+vmxfy6WwfwIhR87Ou3tzS37M2Mj18gdvoYMWVaDGtDAPsvVXO+BQiXwCS1XzWk4czPjwBr4zpmrsp/sYB0x9oNdmCLgSQdHHkHvFVRKrYsjCPd3rifb7hfLMsdHU6H9mb57MQLsHHTEyUiGxh1gj5zvy0D+2aT6L1vJpOT55ibRCOEKg2Rt9LBHPzT8a3mVxhvqpkfSKXTFcSPybtW1LtD9W8n8nHXYTtvxsVcTT3EvJvpU7QdqY3cdm2gD0uPucHdVSXFmCszmSs/Hve2dbHvgq/vSxc3fSqWpHfzmbt3H6ZNwZnzEb1EE91yU3XZNvzw25AjIuj0k86L+3dvwWSa1YfmmMppc/mMiJ1Nj5cjMuD0vsSpQQqC6fcOx8UqIszcLbfeHqTeHK81DmWafu6H0gqW0TzvBPrWw2pj1Pl2Ue8KcVhvp4yaWtZ+Aa/Huu3XW4vPbn+L0JaN0ZbQpk5EPYSdI5+ep7vHi/FgSrHP1UzqQQqvmrmtwmVq1TUoXYPBRS9QTwWqzD8mSRLWJYgGtUtBIzZ5Kuwd/A0dp5b0qRNl0gcvHVjeu5YNpyCnSAtr8aZC9yYyHmsnhdxPe2lE5LpnsPnV5yZrc8nhbA1ndBGDpWpOs4ioGvKbSNyfXcIKX6uiMlsbs6GlKhX+9iEIsFVMOz8IbGaPzaafjI1zyQhi1LyaCdSa3SXl8yBgMIUxuTyrdNeFg1Dl5WPVzX1Rv5nbsq8Gi0xmFehQrnWzHjmFNdRbxOhYQZ4ZdX2uAnQ9SxGmb/rd1MhqhR69ExpZ5Q7nrW0tMTpO4e/6Wa2lRZ3GXH7jOsS+HhQ3wRLQjQD7Tt4sd5NgujLkvEv3qnSNsXOsTH5sJQmCjvyqaPemM/TAc4N4SBYzCNh2QUgWFP8ewP9zUWnupw7Gg56KR5TthqBCeG1O0w5YyZBWa3y3lUazXKhN/+6vBDXk+t8HRxdSJOGMAbjXdTLYJoQ+sb4cDYzPQgwCwMXGqRZcTnzMNGkWFFyyQOK97W+JxePgXvpQPK3WoAbdS9oP/Z6NeIJp1CYDvZs/eUPcctJG5yfePQuh/H4oEXT3kwG5NndrGqXW/oTF6X/9Q+oZo+z9Y2tdWD7NqavwXY0cFoedIWyHZM9zsPdb9ng6/2mm2sHvgmxwdIeTmynsMAhphLSfcQ3zPBZw3tr7d9mWlLdBrXS2D2Gp7HYDvGjgL6wrhDoCOz0XbTwDZPrGx0h5SfNgdU35Nb0LPvnqtmJ2B9U8Pzb+zt66yYpjx2GQCQWNHuh9mXoJFqAGJzOZDnr2nVeNrW2LNaasa9XBWvjFd7P4s2Etrs9zSGZ0wUKlVOubq5TncbaPEAOgIgaeDMKcMq53KQjQ1KBAd3O4zhqvMRv2BT9K0F3qdVd5GvTMb9lDGCtfBjOGPu0+g0GACANbH2lM5KVBV1odE/3HqIV4KDZu53KTCkLYQTgXZszjJHMr+j0NNJfe7PD8DGkmr6vqgxbWmTRhevIwn+GzWXPo5+0ozrjMFkNJeqA8vASGUnDKWoryDDm7/H8Dsxw/QDaAqtCeWVbY3Oqkdu7Db1VJeaVdN1qc55+gJjQ/9+/c4E7YbSRZgTqnlZ1DgVjbTNZmcP12+WyQEeXjrR9adHVcvtmp4VZwqFH3g1Y8mhzMOFtUi5TXWP0fx9N58ahpmr1ABlSQ6yrhtdYnamBzFmZ8XhqG8DfPhqu2c8ZhTIaWTkO7Pu0SMe5vx6o+9lSZQGYRE9ZEeUOBEm9tEemazSqJj4TbTr9TgTXG0Or/X8AG3SZgqfvtt1yoUDdcI0wpHvdUHyMsdRZrnyxXmTh9sNSr1CS7tZmpcgmeGzZ3LrerdnnWTwJLRvfX5U2o/kD2gqrTDRTKZWeUKKbJyYZUSCJNPmxa58QKKTdWrBsUhwqdv+5RmarWnVrnYlNW6aghuv36fYhv6g+Hx9jZeP/rR9JjTZFJ5tqDVqVMaW3yzQAm/OD4L2VatZ9Gqof/4+dd+Gsv2RVRUgL8uv5vI9GRqrfI4TxGEEOaNOCjwVmXt82iUgOn/jzcBXVcKrCg+n09DCKP3fx8jWMMB+UovAA5M6GWVO+hMqvaZEE4b8W2OkUnNTCcvM9/0UuKMOgkG3+PkMFYfmrJ4JLKbsgJ/wlFLBBvgTn1frxjVYq8V5k0ruhzXFwLB3F2PkeYHq74PpJkhc3g8/gtGN/bAIsWGqG1I2qOZdU8qW0bzOMnSnmN7kvdeb0rBq/ghNXiLSAgWwg8vV/Mj8y0XfWlKQSn/ztGr1TSYaYWhzGsWLsbEYPvMxTmjp+Y7nWhlZvFz4vpMd9F3JtHYb2JUGpfb3JuSzHHq4qWGYZwupL1YTp7eZJ5e2Ymm3HOkqErkzhhbzGS4NZ3HrLugCJ+i1hgBqlrqf+ZRi+jwVBXN/Zgv5lZhzJ9/Im4hG1u7qkKi7aCR3iYxJ8tGFjR/4iS0JiM8I7i/PL/pRi3bJ0KK3Zl9cdhLE3Gq5yW5yaOwaHwLHhk7yp3T+ISsZvGGgX2aSVDx2OJQv8WRimfLbvgFYvvc2d6mgw/zE76j8kVxZzGxOQBXjT3lHc5rbpdbmCdhBa7+AJQDxl7XiT6dX5GJNbiRBP4wvpQhpKm5rUw3qJ9Ms0DcrdXB7f7Gm1DZrcF6CGhBHCu9kEO2BhOAZP5g1UD58bi5DvNIYnAo6zCCg9nf86pGAeu4EtKi9k

mH9NvO8N1Ebsy0mN3CZmNj7m2Y2UHEbE9ugLfIPUaBwb2+SOWOxk4PekK505Md8GUmL/S2y30
cAXSVd0xHpy5foI6Wpp59jO+5Aa+I+RGp1XSb3ssaSG40Iz24/UwXrM3T8a6zghtt8HgQPd/U
OkBuHDpGPxNNLx/jlS47w/6WpwabwWkhrbKcDPSI34NaSOwCNRl0NJnZHDSA3PoNhfH45Xdaw
3pynK9xzw8ajTNaSe8UGkhLl37LMrY5mdSZ2195PO198C346y9iO1BytJx8sNpGP2LVICrI95
xkcdPhExTwIOB0aROnTfifWvIx0Ha0nH2GGkdmEw6Th/POp3g7sfRDpuz8lVzsx+TmprDiQdh
w/F8EBSUn2RdGz3j+02CPi3KNPlpGM2l4R/Bnze3f9O2+6MdWNm/0TqODX/+ozbgX+P93Wkfd
qmTk8M7v40acPVseVfdrvT3Te6+wLSDm/JEcDBwJ/M7EnSAdQrYvQhNfKwuVfQnZThdwWuBfY
D/hI75FTSTl7u7n9qJd7QeD1BOnAOJB08vUiN4e+jrA+Tsnst6aQ9kHRQ92XzlUwT6WTeh9SL
+K27LyU1Bu+NeL1ifBPpIFoN/JDUm+8V88wjnZA/IB00y0mN906kHkVzz5F6vocCue9X2Rd4w
t2b3P0VUg/n68CvSYlmZSx3DKkx/32MXxz1eyXGnwxcRWpcP0DqUfciJbxfkhqD98a2g9TbOy
jiPQUcSer1DgFuJt2C+SCp1/vTqNvDpKvNQ0mNxjDSftyddBI6qdGfQGoUe5KOhxXA0kj4j/H
uE+p/YtnLSA3SZ0m9yOdJjcjG2Nb7xPxN7v4qKeFviGW+Qtq/66LsHyUdVzuT9jekY+D9pCT0
NOn2zHrS+bAIWObufyU1Jj2Br5KSy/tIHYQDYpu9z8wuAz7k7t+McY3AdDM7Nt7fRTrvbyCdD
7krgmejXveRkkIVqfE8mdT4HRzjFpOuRPuTjtnxpAZwTzO7LrbVbNL5sFPUKyfXsVoM/MPdF8
W4bqTjYBHpWO8bw5D20TBSh+mTpA7HUtKVwLOk4+j/gDV55+iPSOfWM8C9UcZTYn/1IvXSHyI
10l8kJac+bP79+b+QetTz2Xx1OyDirciL88uIA4C7nxb1aSB1RI+Kct5OOo7WxTZ+gJSQdyYl
+OqIczdp/x4P3Ofuz7Jl7wAvmdmgiPmrFub5DSkBQerw3r2llXZ6YgjTgP9ly5msMe+9tTKPA
fe7+0ejd7La3U+PGLuSGjZIO6oR+D6pZ/EKqQe+Y0yDdMI1kg6a1hjwg1w8d9/f3SeSeh1/Iz
WwRmoUF5Duz3YjnXzDSImrP6mByXE231oi3rdU3++TTpIxpB2+g5ntHOtyUoO0ktTgfYx0ov4
5ls3/WdfxwE9IDWsfM2vtJwPlARsAAAZqSURBVF+XufvN7j6M1EDOIPXgczGXkrbxfqQrondI
Dc2seH2N1LO/nXQC/YLUqz6f1KN5lGZfmujuz5CSxv8DhsfJcj/pJJtCuiKC1ANtIm3TSWlR3
4+UVIyUJL9GOgFPjH2U/088G2n9fLiNdGXZnMe6c+vJLT8v3g8idTzWkK7mniXdOnowyr4s5j
8nyr6KtC03khrIDfHKJRCP1+ioyxyyt5qOIHUsbjGzr7r7MlJjuhdwNilh546tr5AS2NRY93r
S8bYq4m8kNWpfA+51957uvqe7/wL4NClJryE1O1Wk428BKfktcPcPk467nrH+/O2UO/fyLXD3
3vHa0d0/mTet0d0bIt6OeevYSMs/T7wub/pL7v4vwE1Rx7ejXXgU+La77xxl75ZXpnV57zfG9
upB9njJaT6uN3A0qRPSj9SuNJ/nWdJV/lxSh+lR0nHSGGVcSEq+W2sqKSmcyOaOb743gWVm9i
VS0nrXnY/mukpiuJl0b/6ZEqzrYeCTZrbpEtbMPhAxlpM2TL5dSCcTpKsWgMPM7MOkZxpPA2s
tPvnTghnAyNynEMxsHzPbMzcxeqPnA4cQD5KiHAD/Sjph9o9p3Um9z1faqN9a4EQz607qWeZ6
Vl8iJYHrST2cPUmX1D1IB3pv0gncEOMG5q1zd3evJ/VeiPmeJd0W625me5EO9u5mVmVm/0bqj
f+e1AN20nZsIh3oe8e0XUgH+wxS7zV35fQZ0rE3g9Tr/VrU/wBSr3wB6RZN30h0X4jt8mLetu
1BumX4G1KvtIl0ApwdZegW+31IzD8ytgHAHrGelcDuZtY7puVfhfZ19+VRt2/E9jglpu1HaqA
aSQn3OdJVTH7ybgT+i3TbEID84yLK/li87xll6UZqsLvTsg2khDEqb557SEnmLmCJu98U7w+N
26pfIPWALyUliGFRp0/HtunfSqyHSY3i8Cj7B8zs42a2LynhTybd7jgx5n+O1Ol14GRhkZv1In
YSBMb2tztV60tXOxyPWP5vZIXGLcvqRjJdeovY/Nt+KGAb1zy0eZH4z3uX3enXS10Bjz7Q
K8ZmaHkK7gttbOeXG+DDxoSa69WUPa/hOi7AeSrkqMdMy8n3R+NgLPu/s5pCuPXDk3kq7q
zb21lmaaTjptF7v5OK/P8itT52+JtJGg523Y4d19MaoRLsa6lZnYqYqMgMMepI2+IHuPs3MVrSw
yFWkHuZebD7ZHiD9ZHiU9l9yRyDdXxx38BsM3vA3V9vvFm+vmR0OtJdL
P03kg6my0kJ4k+ky+fWrCZdjuYejhvplsdtpB7oq6TbKB7oq6TbKTuRGqsHSb3AZ0iNwZr0iNwH2kBibXWzXgfD
M7J96vd/flZvbfpBM9t+wbpFtWl5KS2yrgalIP9yVSA10T660iXYXMIF0yH0pKJJeSGrUlpEb
4f6Ocg2K59VGGUaSGeHDM20i6XfdFUvL+V9K+OprUSz6blHi/SbpyyO3Hx0m3EppIDdlPSA3C
lbGONVGv3EPsXMIGON7MPh/lGUm6/fF8TMvdb987yncycavPzHLPKRpj20whPQQnyn4RqcG8P
m99N5Ceo+xFena2npa9GfN8kHT/uyrWVU3qBDxlZutJtz3eJN2S2JPUWP+SlKw3knqj46PsrX
1s807S1ezZpEa9lnQbcArwLdIxsJG0jb9HunrpRroVu8t9IHGlmT9B2J3RtzPv76PCsJ10
FryUdA7eRjrcfkfY3pMZ1H9IdgGdJielc0j6pJjXMJ5OeTTVEudaT9sGVsR1/Edvg1jbK1txy
4Fwzu5l0DuYSwCRLH53uHeseReq0/THKf1dss+Fsvq1LfGz2YNI5mbOO1NifYmZL4uq2Ve6+
szGxjZozf9FuWaQtlHbSvHpokp6kXpRX2UtHmMrPoLYVV+8+5Nj7/rIcBlidkicYvYX6Q
HnLzqgPFsdZ1s/1vLqMZC8T9U0H+6ouK3Ms7LZ8EukTu+mZUnPiJ4mdbjmkzqn1nx5UoL/B3B
Ce/YnKUnnfyqpb3vr0SWuGEQqSTyEPY7UY9/m48jWc/fm/9g2C3jA0yf9BsU8t9LKVUr+8u7+
Eum2ZWuxLmtl/JC89wNbmL6pLK3Rl+iJiEiGrhhERLowM7uE9Ht36/dfXzZYuqKQKCEj9Hwt36/dfXzZYuqKQKURE8nWVj
6uKiEgXocQgIiIZSgwiIpKhxCAiIhn/Hz5aDcZlsZGVAAAAAElFTkSuQmCC\n"
},
"metadata": {
"needs_background": "light"
}
}
]
},
{

```
      "cell_type": "code",
      "source": [
        "sb.histplot(data['RainTomorrow'])"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 296
        },
        "id": "iJBcyZpj7rk4",
        "outputId": "5aed7dbe-658d-46a4-cdf4-d6fe8d5b3df4"
      },
      "execution_count": 27,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at 0x7ff7dfc65bd0>"
            ]
          },
          "metadata": {},
          "execution_count": 27
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x288 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYUAAAEGCAYAAACKB4k+AAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAASU0lEQVR4nO3de7BdZ13G8e9DW1qkSF
t7jCFNTZUULSApc6iViwNU5TKjqQi9DFMqVoJaEBQvIDNSLxVUoA5eqtHWBkXaiu00QAcspUI
ZoCW9mN6oxF5MQtuccq9gMeHnH3vl7TY5OTlJs/Y+Oef7mdmz13rf9a7zS2Ynz1nvWnutVBWS
JAE8ZtwFSJLmDkNBktQYCpKkxlCQJDWGgiSpOXDcBTwaRx55ZC1btmzcZUjSfuWGG254sKomp
uvbr0Nh2bJlrFu3btxlSNJ+Jcm9u+pz+kiS1BgKkqTGUJAkNYaCJKkxFCRJjaEgSWp6C4Ukhy
S5Psm/J7ktye917cckuS7JhiSXJHls135wt76h61/WV22SpOn1eaTwMPDCqnoGsAJ4cZITgT8
GzquqJwNfAc7qtj8L+ErXfl63nSRphHoLhRp4qFs9qHsV8ELgA137GuDkbnllt07Xf1KS9FWf
JGlnvX6jOckBwA3Ak4G/BP4T+GpVbe022QQs6ZaXABsBqmprkq8B3wM8uMM+VwGrAI4++ug+y
5celSVLj+aLmzaOuwzNU086aimbN/7XPt9vr6FQVduAFUkOAy4Hfmgf7HM1sBpgcnLSx8Zpzv
ripo2c+jefHncZmqcuee2ze9nvSK4+qqqvAtcAPwYclmR7GB0FbO6WNwNLAbr+JwJfGkV9kqS
BPq8+muiOEEjyOOAngTsYhMPLu83OBK7oltd263T9Hy8fIC1JI9Xn9NFiYE13XuExwKVV9aEk
twMXJ/lD4Cbggm77C4B/SLIB+DJwWo+1SZKm0VsoVNV64Php2u8CTpim/X+AV/RVjyRp9/xGs
ySpMRQkSY2hIElqDAVJUmMoSJIaQ0GS1BgKkqTGUJAkNYaCJKkxFCRJjaEgSWoMBUlSYyhIkh
pDQZLUGAqSpMZQkCQ1hoIkqTEUJEmNoSBJagwFSVJjKEiSGkNBktQYCpKkxlCQJDWGgiSp6S0
UkixNck2S25PcluQNXfs5STYnubl7vXRozFuSbEhyZ5IX9VWbJGl6B/a4763Am6rqxiRPAG5I
clXXd15VvXN44yTHAacBTwWeBHwsybFVta3HGiVJQ3o7Uqiq+6rqxm75G8AdwJIZhqwELq6qh
6vqbmADcEJf9UmSdjaScwpJlgHHA9d1Ta9Lsj7JhUkO79qWABuHhm1imhBJsirJuiTrpqameq
xakhae3kMhyaHAvwBvrKqvA+cDPwisAO4D3rUn+6uq1VU1WVWTExMT+7xeSVrIeg2FJAcxCIT
3VdVlAFX1QFVtq6rvAH/LI1NEm4GlQ8OP6tokSSPS59VHAS4A7qiqdw+1Lx7a7GeBW7vltcBp
SQ5OcgywHLi+r/okSTvr8+qj5wBnALckublr+x3g9CQrgALuAV4LUFW3JbkUuJ3BlUtne+WRJ
I1Wb6FQVZ8CMk3XlTOMORc4t6+aJEkz8xvNkqTGUJAkNYaCJKkxFCRJjaEgSWoMBUlSYyhIkh
pDQZLUGAqSpMZQkCQ1hoIkqTEUJEmNoSBJagwFSVJjKEiSGkNBktQYCpKkxlCQJDWGgiSpMRQ
kSY2hIElqDAVJUmMoSJIaQ0GS1BgKkqTGUJAkNb2FQpKlSa5JcnuS25K8oWs/IslVSb7QvR/e
tSfJe5JsSLI+yTP7qk2SNL0+jxS2Am+qquOAE4GzkxwAvAHk4N4WzzmJbvnBJwPmVMVT
kmaRm+hUFX3VdWN3fI3gDuAJcBKYE232Rrg5G55JfDeGvgscFiSxX3VJ0na2UjOKSRZBhwPXA
```

csqqr7uq77gUXd8hJg49CwTV3bjvtalWRdknVTU1O91SxJC1HvoZDkUOBfgDdW1deH+6qqgNq
T/VXV6qqarKrJiYmJfVipJKnXUEhyEINAeF9VXdY1P7B9Wqh739K1bwaWDg0/qmuTJI1In1cf
BbgAuKOq3j3UtRY4s1s+E7hiqP1V3VVIJwJfG5pmkiSNwIE97vs5wBnALUlu7tp+B3gHcGmSs
4B7gVO6viuBlwIbgG8Cr+6xNknSNHoLhar6FJBddJ80zfYFnN1XPZKkk3fMbzZKkxlCQJDWGi
SpMRQkSY2hIElqDAVJUmMoSJIaQ0GS1BgKkqTGUJAkNYaCJKmZVSgkec5s2iRJ+7fZHin8
bJEn7sRnvkprkx4BnAxNJfn2o67uBA/osTJI0eru7dfZjgUO77Z4w1P514OV9FSVJGo8ZQ6Gq
PgF8IslFVXXviGqSJI3JbB+yc1CSrcC6wT5V9YI+ipIkjcdsQ+FfgX8A3g482F85kqRxmm0o
K2q83utRJI0drO9JPWDSX4lyeIkR2x/9VqZJGnkZnukcGb3/ptDbQX8wL4tR5I0TrMKhao6pu
9CJEnjN6tQSPKq6dqr6t37thxJ0jjNdvroZ4eW+AN+fB6ksOAi3upSJI
0Nnt76+z/BmY8z5DkwiRbktw61HZOks1Jbu5eLx3qe0uSDUnuTPKivaxLkvQozPacwgcZXG0E
gxvh/TBw6W6GXQT8BTtPMZ1XVe/cYf/HAacBTwWeBHwsybFV5RflJGmEZntOYfg/8a3AvVW1a
aYBVfXfXJJMtmuf+VwMVV9TBwd5INwAnAZ2Y5XpK0D8xq+qi7Md46BHVKXAn+acJLZOpwMMzxfpueu
nwrm0JsHFom01d206SrEqyLsm6qampR1GGJGlHs33y2inA9cArgFnOA65Lsza2zzwd+EFgB3Ae
8a093UFWrq2qyqiYnJib2ogRJ0q7MdvrorcCzqmoLQJIJ4GPAB/bkh1XVA9uXk/wt8KFudOtOw
dGjTo7o2SdIIzfbqo8dsD4TOl/ZgbJNk8dDqzwLbr0xaC5yW5OAkxwtwDLGRyZSJJGaLZHCh9J
lHg/d36qcCVMw1I8l7gecDyJJuAPwCel2QFg+mne4BfAqiq25NcBtwBbAPOds9RStuGqKTp3l/5Pe+
4ysKiqfjPJy4DnA7J2w3
ne61OkjRSuzsvsKiqbtmxsWtb1ktFkqSx2V0oHDZD32H7shBJ0vjtLhTWJ/nFHRuT/AJwQz8lSZ
LGZXfnFN4IXJF7fnFN4IXJ7klTwSApPAYxlcUipJmkdmDIXuy2bPTvIC4Gld84er6uO9VyZJGrnZPk/hG
uCanmuRJI3Z3j5PQZI0DxkKqTGUJAkNYaCJKkxFCRJjaEgSWoMBUlSYyhIkhpDQZLUGAqSpM
ZQkCQ1hoIkqTEUJEmNoSBJagwFSVJjKEiSGkNBktQYCpKkxlCQJDW9hUKSC5NsSXLrUNsRSa5
K8oXu/XFde5Jcm3tm2RDkvVJsKa8xW2Y9mSIHAAU11BMWZS2zTqBwbbzKrHLgI1D2
2oXeQqGqPgl8eYfmcabnkNcPJQ+3tr4LPAYUkW91WbJGl6oz6nsKiq7uuW7wcWdcuLgY1D2
23q2naSZFWSdUnWTU1N9VepJC1AYzxRXFXFcBm1jNTRhUHtg+LdS9b+
naNwNLh7Y7qmuTJI3QqENhLXBmt3wmm0wmcMVQ+6u6q5BOBL42NM0kSRqRA/vacZL3A88Hjk
b8A7g0iRnAfcCp3SbXwm8pFfF7d9F10jTbFnB2X7VIkmbHbzRLkhpD
QZLUGAqSpMZQkCQ1hoIkqTEUJEmNoSBJagwFSVJjKEiSGkNBktQYCpKkxlCQJDWGgiSpMRQkS
Y2hIElqDAVJUmMoSJKa3p68NtctWXo0X9DycxlSNKcsmBD4YubNnLq33x63GVoHrvktc8edw
nSHnP6SJLUGAqSpMZQkCQ1hoIkqTEUJEmNoSBJasZySWqSe4BvANuAP
VfWUc9UnSQjXOI4UXVWKprs1t8MXF1Vy4gru3VJ0gjNpemjk4F13fI64PQx1iJJC9K4QqGA
jya5IcmqsSQbNT8ZUSdYlWTc1NTWKWiVpwpRmJ1xrWkcFqWS3wvcLFWSzw93VlUlq
ekGVtVqYDXA5OTktNtIkvbOWIGqTpIVs5KGQ5PFJnrB9Gf
gp4FZgLXBmt9mZwBWjrk2SFrpxTB8t5Psv3n/1NVfSTJ54BLLk5wF3AucMobaJGlBG3koVNXd
wD2DOmaf8ScNKo65EkPWIuXZIqSRozQ0GS1BgKkqTGUJAkNYaCJKkxFCRJjaEgSWoMBUlSYyhI
khpDQZLUGAqSpMZQkCQ1hoIkqTEUJEmNoSBJagwFSVJjKEiSGkNBktQYCpKkxlCQJDWGgiSpM
RQkSY2hIElqDAVJUmMoSJIaQ0GS1My5UEjy4iR3JtmQ5M3jrkeSFpI5FQpJDgD+CvhZ4DjgFUmO
Ljxl.uVJC0ccyoUgBOADVV1V1V9G7gYWDnmmiRpwUhVjbuGJsnLgRdX1eu69TOAn6uq1x/aZhWwql
wqlt9CnDnyAtduI4EHhx3EZq3/HyNzvdX1cR0HQeOupJHq6pWA6vHXcdClGRdVU2Ouw7NT36
5oa5Nn20GVg6tH5U1yZJGoG5FgqfA4A74KHAAcGFV3
TbmsvQIp+3UJz9fc8CcOtEsSRqvuTZ9JEkaI0NBktQYCgIgA59K8pKhtlck+cg469L8ZkBPmVR8T
to/TeSnDPGkjTEUBAANTi59EvAu5Mcku4hzdJIzZuLYtv78BNQYsGPmTqeb3c69p7LTgAvIsSl
A7wL/CLw1yfVJbkqyEiDJU7ZFc7OsT7J8jGVr/7OVwZVGv7ZjR5JlSTzl/61PRKt1Lz6
SP9PkscDbNwLfBj4E3FZV/5jkccD1wNOBtwGfrao1SQ4CDqiq742tYO13kpwEnAf8JFDAXdVlcWNL
FWdk+SDwAeqak2SSDXv9kBHAG8v8AB8iEEwvBV4ArgguHM
Vq/5Tkoao6tPuc/S/wLR4JhQeBxVX1v0kOAu6krKqeZRsjpI03nO90rwM9V1YrudXXRV3VFV/wT
8DIN/zFcmeeE4i9V+68+As4DHj7sQPcJQ0Ew+Crw+SQCSHN+9/wBwV1W9B7gC+JHxlaj9VVV9
GbiUQTBs92kGt7Bc3kD4CDgPVJbuvWYTCtdGuSm4GnMZhGkvbGuxjcdDb9NJ4NhYBPxJ6bnHzqqZXcCt
wOvTrIeOAN4w1lqWsA8pyBJajxSkCQ1hoIkqTEUJEmNoSBJagwFSVJjKEiSGkNBktT8H96aLaoHRK0+
8NnFFVX51h+0ngVVX1q7vo/x7g6m71+4BtwFS3fkJVfXtf1S6Nm5ekat7ZfhuFbnkN8B9Vde4
+2vc5wENV9c59sb/d/1/CQf6pbPSXJh
kn9LcleSaY8eum1P6vZ1Szfm4K79niRv7+4iuy7JMt3bjTx2q5doka
4Hbp1k/JMnfd2NuSvSvKCbtyva/o88LCyqOp
7BrcH/aBdDf4jBjf9OAN7W3Zhtx30fAlwEnFpVT2cwVfXLQ5X8X9b8V1WtYHCbhouAlAlwMnAr/X9b8
MWMHgLqE/AfxpksVd3zOBN1TVsdOsn83cLqZnsEAs8L8kTGdzA8Dnd2ozhr0n6
fwwFzUeP627BcT+wCFiXZAmDA9+pq8+4u+EyHeaBc9Po/f+1/yhuKGHH9fLPUbWkb8GCFnRS4pB0n63
/YgugW4rqq+6+4+6rk8B3Al4N/CbcHsZAfwH/uYZ
6cQRh8GDg0yXcBx1TV1s9ZD8bGoYCNPHgyp1JdqgTVnbv4c0o7MRQ0H32r+wXRa4pM/e5fk3h2Wz7XvGiyptMf/P5JxQJ
s3QUZ2/fxnR32951Z7O+/d7M+nc8BkzxyZHATg+cU3DCLsVJjKGjeqqpvrfX+KvCnJgQyOFDZ3

3T//KHd/J7AsyZO79TMY/LY/W9cCpyY5IMkEg9/yr5/luFcCJDkWOBq4s7sCaiPwCgbnUa4Ff
gOnjrSHDAXNa1V1E4MnfJ0O/Anw9iQ38Sgvx66q/wFezWA66hYGRwB/vQe7uLyr69+BjwO/VV
X3z2LcXwGP6X7mJcDPV9X2I5FrgS3dU/CuBY7CW09rD3lJqiSp8UhBktQYCpKkxlCQJDWGgiS
pMRQkSY2hIElqDAVJUvN/dZXBR6jKA74AAAAASUVORK5CYII=\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sb.scatterplot(data['MaxTemp'],data['Rainfall'])"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 296
        },
        "id": "rrYgfg2x70xZ",
        "outputId": "b87d7012-a6a7-4c2b-acd1-43270c0ce0d4"
      },
      "execution_count": 28,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<matplotlib.axes._subplots.AxesSubplot at 0x7ff7df307650>"
            ]
          },
          "metadata": {},
          "execution_count": 28
        },
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x288 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAX4AAAEGCAYAAABiq/5QAAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nO3de3xcdZn48c8zl8zk3l
tSSkspgZRKoVY2slWBxVb49YVYBBEWXYVd3OdFtEXdRdxYL+RBeLdL3sFkGBZQV+onKRZa1
QBFYuplIKhUKhNxvSJr3lPskk8/z+mAuTZCaZmebMJed5v155ZebMmXO+Z845z3/z/YqnO/Yqq
YowxjO8hS6AMcaY/LLAb4wxLmOB3xhjXMYCvzHGuIwFfmOMcRlfoQuQQiRkzzZUi8efMKXQxjj
CkpmzZtOqCqdSOnl0TgnzdvHs3NzYUuhjHGlBQR2Z1quV6jDHGZSzwG2OOMy1jgN8YYl7HAb4
wxLuN44BcRr4i8IJYZQ5XQZjjDFvy0eNfzfwSatSETZ0d7NM28eYEd7N5GIdfBnTClyNPPCLyBzgg8
1n2cv+4pLr/1Oc5f9xSPbt1nwd+YE+BS2aMyZZjgV9Ek7cPqTx4dEgZF8awMpBuI
XsnaSxcT9EcPmaDfw9pLFzNvemWBS2aMyVh+uq5FrhHRL4JvADcVoAymCx5PLKf7aMVkZt
HWFqK8OMm96JR6PFLpoxpgs5SXwq+oTwBOxxzuAM/KxXj6Khrqq
QhfFFGHMU7M5dY4

xxGQv8xhjjMhb4jTHGZSzwG2OMy1jgN8YYl7HAb4wxLmOB3xhjXMYCvzHGuIwFfmOMcRkL/MY
Y4zIW+I0xxmUs8BtjjMtY4DfGGJexwG+MMS5jgd8YY1zGAr8xxriMk4OtB0XkeRF5UUS2isia
2PSfichOEdkc+1vsVBmMMcaM5uQIXP3AUlXtFhE/8LSI/HfstS+p6i8cXLcxxpg0HAv8qqpAd
+ypP/anTq3PGGNMZhzN8Yu1V2A23ABlV9LvbSt0Rki4jcLCKBNO9dKSLNItLc3t7uZDGNMc
ZVHA38qjqkqouBOcAZInIq8q8BVgAfBuYBpwbZr3rlfVJlVtqqurc7KYxhjjKnlp1aOqR4CNwHJ
VbdWofuCnwBn5KIMxxpgoJ1v11InIPNE+DDwMtOlcEYY8xoTrbqmQXcISJe
ol8w96nqwyLyuIjUAQJsBj7jYBmMMcaM4GSrni3Au1JMX+rUO0xozP7tw1xhiXscBvjDEuY
4HfGGNcxgK/Mca4jAV+Y4xxGQv8xhjjMhb4jTHGZSzwG2OMy1jgN8YYl7HAb4wxLmOB3xhjXM
YCvzHGuIwFfmOMcRkL/MYY4zIW+I0xxmUs8BtjjMs4OfRiUESeF5EXRWriKyJTT9BRJ4TkTd
E5F4RKXOqDMYYY0ZzssbfDyxV1TcQwC+A5ws6qeBBwGrnKwDMYYY0ZwLPBrVHfsqT/2
p8BS4Bex6XcQHDDGNMnjia4xcRr4hsBtqADDcCbwBFVHYzNsheYNea9K0WkWUSa29
ca4iqOBX1WHVHUxMvq2qTqjbV1dU5VkZjjHGbvLTqUdUjwEbgTKBWRIYvIHOAA/kogzHGmCgn
W/XUiUhtbLoceD/QQjQBXBqb7RPAg04VwRhjzGhOtuppAu4QES/RC8x9qvqwiGwD7hGRvwf+B
LyCvAPSLyeaASAF4DYHy2CMMWYEJ1v1bAHelWL6m1i+vjpbo8ldnEjNBxXgxTzaU8K4AjnHHHSZdSxm
LyCvAPSLyTeAF4DYHy2CMMWYExwK/qm4B3pVi+g6i+X5jjjDEFYHfuGmOMy1jgN8YYl7HAb4wx
LmOB3xhjXMYCvzHGuIwFfmOMcRkL/MYY4zIW+I0xxmUs8BtjjMtY4DfGGJexwG+MMS5jgd8YY
1zGyd45jZlwkYiy62AP+ztDzKwPUC/ymZEQiyqNQb3HNfSZsJhhUC/ymZEQiyqNb93HNfZsJhSME/R7WXr
qY5QuPseBvTBYs1WNKxq6DPYmgDxAKR7jmvs3sOthT4JIZU1os8JuSsb8zlAj6caFwhLauUIF
KZExpcnLoxeNEZKOIvCIiW0VkdWz6N0Skz6N0SkRUQ2x/7Od6oMZnKZWRMk6B9+yAb9HuqrgwUqkTGl
ycka/yDwBVU9BVgCfFZETom9dOqLo79PeJgGcwkMm96JWsvXZwI/vEc/7zplQUumTGlxcmhF
1uB1tjjLhF5FZjt1PrM5OfxCMsXHsOCVWfR1hWivtpa9RiTi7zk+EVkHtHxd5+LTfqciGwRkd
tFZGqa96Ww5KSQjnSKywaiuogKKnjGoqoljTMoKGuyoK+MTlwPPCLSBVwP3C1qnYCFwBBAY
T/UXwvVTvU9X1qtqkqk11dXVOzwDMt0F19MYY1zD0cAvIn0Kn0lgKruV9Vhn0tGBpwdfsD9BJ4QcTXYH8ARwJD
MYY0ZwslXP00CqKK8/Wd8pDVymoYWrnCWyRvE8Td/uAwosBBXYBn3wD
MYYY0ZwslXP00CqK2/WfNMYYwrI7tw1xhiXscBvjDEuY4HfGGNcxrplNq5m/fsbNxoz8IvIvx
FtfZOSqq6ba6dBIZkyfWv79xq/Fq/M15KYUxBZCuf/8Fq86ioa6qwKUzxjljBn5VvSNfEBTEm38b
q398Cv5nMxkv1PMTYqZ4VE14iY/Ik3r9/cvC3/v2NG4yX6rkpL6UwpgDi/fuPzPFb//5mshsv
1fP7fBXEmHyz/v2NW2XUnFNEGoFvA6cAid/cwAivtd/BqtrgULmMyYt/2v0wo/+IPB+4
E7gP50qlDHGGOdkGvjLVfUxQF3TRGLTBeNKUK+wXUQ+B7QA9tvYGGNK
Y1/tVABbAK+Avgb4ArnCqUMcYY54wZ+EXkrtxVhfrkK+5qb6ruVdWtWGBdW/OXQCq/Hm0OQDKCKviMh
WEVkdmz5NRDaIyPbY/5SDrtjjjHGeDX+vxCRY4G/E5GpsaAjAq6c/cCdHbzDhtMYZOD9TJ3ZRvFkl
MvCYqjYCj8WeG2OYPcxbvz/TjQ4NwCbD6G6ilsamp6SqrUBr7HGHxwvBNdUV+BKB6KAbWa+H8HF
fZFNyZ31iuncbPxbuBaB6wTkR+r6r6jJiEm9ZbjIFXV4BxjK5QLmDSmVTb3fN1BIZ6NWkqkEJg
X9BxHxisixIjI3/pfJe0WkCrgcsBsBEdR4a+cww72FLh
kxuRHRoE/1oRzP7AB+E3s7+EM3cmfGvTvVtVfxibvF3zbZsZu7Zd1W2p3quq61W1Wb6urqlm
yYpElB3t3TZ5gF2tHcTiaTtF89MgLF65TTGDTJtx381cLKqHsx0wSIq6G3Aq6q4Ng
b0x9v+BTJc5GVnaIf+sV07jdpm24/8z0JHlst8HfAJYKiKbY3/nEw0Aw5454rIduADADSeeuZWkHZ6
X6NRXvlTPojx7+1iunc1tMa/w7gCdE5DdAf3ziiJr8Mkr6NMNbASVblnEJJzkbDMQ5Y/2asl4
5jZtlWuPfQzS/XwZUJ/2ZoxRPOySztMPEGOvXVLxXziUNM2ioq7Kgb1wloxq/qq5xuiBuZYOB
OMd+TRmT2nhDL35fVa9ONwSjqDb149GwwEOfYRVxjUhuvxh/vq8eGYHSQDQbiDPs1ZUxq4925u
yn234ZgNOMqtm4QMvk1VWxlNiYfbOhFMyGK9X6EsX5NFWuZjXGaDb1oJkQp3o9QimU2iLY0I
tZsK4V0ivFbhBKsczGTAQbejFDlhYYWym2oCnFMhszEXIdEwCedKlQQxsrTA2EqxG4RSLL
xEyHTG7j+GHvYDfytiHiBvybav74r2M1AYyuF+xFSteDJV5mt9ZApJuPdwFUDfJboyFkPEu22
4bPAF4AtwN1OF7BYWgfMV13AWCDwLCfhUEhVL3S4S4bEXF0
gKlrZCpOksTmmIzXqqnQVVVPAxCRnxAdQ3euqrqu2UMpMeoVM1Vma0BSb8QJ/OP5AVYdEZK
8bg35cMacyzNgKmaqzNKEpNuOlet4pIp2xvy5gUfyxiHSO815jikYhU3WWJjTFRqLjnTuwYJH
bgQuANlU9NTbtG8DgT8Dfa+qqo+Mt6ympiZtbm52opi5wk+K2vZuI+2vH5W5
G7hy8TPgB0S7d2m6ZTWuaRC5Mt6I45NTyjTHG5MaxwD+Gz4nXh5X
USmpptJRFaKSLOINLe3et6ebzRhZhG5nl+FEdYSDhqyQs5X1SZVbaqrM8
wxRyl+w9r5657i8luf4/x1T/Ho1n0W/ItIlhSFVjQC3ZmfkL6+BX0R
mJT29CHg5n+s3xjjPursufo61fkhGRnwPnCZDNEdD7LuDTTq3FKb1sreZiI
8Kvq5Skm3epmF+aWAU1WWJjTFRqLjnTuwYJH
RK4VuE/JdxmK+8GyKnwV+U9RKofVKocpo3SKYXFmqxxS1Umi9UgplNCaZ1fhNUUtuvTKrNsjF
p89BBNq7+4smtTFWCxurjZtiZDV+U9TirVdm1Qb5xJLjue3pHfzg8Te44vbni2YcV2thY0qNY
4FfRG4XkTYReTlp2l2jQR2SAi22P/pzq1fjM5xFuvfLRpDuse316U6RRrYWNKjZ0pnp8BPwDuTJ
r2ZeAxVb1RRL4cLL4ce36tg2UwJS7emkeVXkUH4wdjolm61srIWNKjWOBX1WfFJH5IyZfRHRYR
BBX4zDo9HmDe9MuNxXAvRysZa2JhSku8c/0xVbY093gfMTDejiKwUkUYkWYRaW5vKz89P6Uzryiad

Yq1sjBlbwVr1qKqKSNorc6q6HlgP0NTUVPgreKagskmnWCsbY8aW78C/X0RmqWqriMwC2vK8f
lPCMk2nxFvZZJIWypdSuPvYuEe+Uz0PAlfEHl8BPJDn9RsXKLZWNvFrDueve4rLb32O89c9VT
RNUY07iaozB5+I/JzohdwZwH7gOuDXwH3AXGA3cKmqHhpvWU1NTdrc3OxIOc3kFK9hF0Mrmx3
t3Zy/7qlRv0AeWXWWpZ6Mo0Rkk6o2jZzuZKuey9O8tMypdRoTV0ytbOyagyk21mWDOWqWvx5b
MV5zMO5mXTaYo2L56/EV2zUHYxzL8U8ky/EXL8tfZ6aYrjkY98h7jt+4w2TLXzuVtiqmaw7GW
OA3w2Qb+CZT/roUBn0xZiJYjr9AIhFlR3s3z7x5gB3t3UWRE88lXz+Z8tfW1YNxC6vxF0Cx1i
zTBb4FY+TrJ1PPlJMtbWVMOlbjL4BirVmOFfjGEs9fL2mYQUNdVUkGfbABVYx7WOAvgJHDCX7
2/SfxqbMaaO/uL2jKx+2BbzKlrbLlZOqxGNOabmepngKIB9ipFWV8YsnxiZGlfvLUjoKmfOKB
b2QKysnAV0w3f02mtFU2nEw9ZrPsYjoWJjtrx18A8ZNh275O1j+5w9E28NmeTPlsb16s1zrcx
sl7MTJdth0LzkjXjt9SPQUQr1kuPm5KTjn1TOXSSief+fpivdbhNrle25nIZduxkF8W+AskeT
jBZBOZUy/2k8nJgGMy5+S1nUyXbcdCflngLyCnLyYW+8nk9ovJxcLJ4zDTZduxkF+W4y8wJ3P
qxd6PTinkdd1ywdHJ4zCTZZfCsVCK0uX4LfBPYqVwMqULCpGIsudQD/s7++kZGOT4aZWcMCO/
QbcUPr9Cm8gvRuvIbuIVVeAXkV1AFzAEDKYqWDIL/LkrxZMpElEef20/2/d3c8tj2wsWdIv9F
1Oh2Rdj8SvGVj3vV9XF4wV9t3DqJpdcW+kU8qabXQd72LK3IxH0oTAXpov9GslI+d5nxd54wK
RnN3AVgWKrORW6PPs7Q0SUgvebU0o9jxZin1nfRqWrUDV+BX4rIptEZGWqGURkpYg0i0hze3t
7nouXX8VWcyp0eWbWBPEKBW/lUSxdOGRSky/EPrOWOKWrUDX+M1W1RUTqgQ0isk1Vn0yeQVXX
A+shmuMvRCHzpdhqToUuz7zplZw2p5bVyxpH5fjzGXSLoQuHTGvyue6zo7k4W4guPszEKEjgV
9WW2P82EfkVcAbw5NjvmrzymVLI5ETPR3nGKofHIyw9eSYn1VVx+typ9A4MMrcArXriZSnkyF
mZdpWdyz472vRQui9GiF4Yn+xNYEtZ3lM9IlIpItXxx8B5wMv5LkcxyVdKIdMuHFKV5zsfWcT
cqRV5K4fHI8ybUcVfNkzn/QtmcmJ9VaKZp5t6esz0AnMux9BEpIdGNh4Asu4mxORfIWr8M4Ff
iUh8/f+lqo8WoBxFI18phUxrjx6PcN47ZrL+E0007z7EUATWbngNv9czIRcLcxnwBQp/0dkpY
/36ybQmn8sxNNaXyrzplTmlgHLdtya/8h74VXUH8M58r7fYTURKYbw0TjZ54D2He1l5V/Ow+S
fqBM41H11MQeVocuPJ762vDrLzYDef+68XUn6ZZZNHz/YYSvelUlcVzPkLttDXh0xmrDnnJJF
JbTibPLCTJ3Cu1xCKJagczS+PVO9dvayRqRVltHaERn2ZOflrMN2XitdDzl+wpdQE1s2sk7ZJ
IpN8bTZ5YCeb6uV6TaOQzQeTry281NLBdx59NafceKr9dMtj27n49DmJeUbm8J3qKjv+pfLIq
rO4Z+Vf8siqs1i+8JjEF1CyeJnGu8ZSLE1gzdisxj9JZFIbzqb2mG1TvWxSH7nWYgvVfDBVLX
3V0kbuenY3rR3RAJ3pL490+0mSNj2fNeRU6aF0tfZjasZPARVDE1gzPgv8k0Q2FwEb6qoSF++
e23kwZaDO5gTOJfWRyzWNQgWVVLX0dY9v56ozG/jhxjeAzIN1uv0U34RiqCGn+4IdimSWAip0
E1gzPgv8k0Q2teFMA3WmJ3A+L7oWIqikq6V7Y1mnbIJ1uv10yqxq3nvi9KKoIaf7gn1u58Giu
MZijp4F/kkim9rwRAfqYrno6pR0tfRlC+qzDtZj7ad5M6KfVTyPXsgboLJJAdmF29Jjgb9ITE
S/5pnWhic6UJdCQHCia4LTZk/JKSCPtZ+K+V4F66IhtVIcrMcCfxHI98k+0YF67tSKxM1eEYW
HXmzh2uXvKJqA4FTXBJm8N9ugkO2vsXwGHbtwO1oxf1GPxdUjcBXLN3W+B/yYyIM11bK+85FF
fPDUWfh8xdFaeNeBbn75Qgvxlof3b9rL4d4BxwdUyeVzfubNA1x+63Ojpt+z8i9Z0jDjqJdvJ
laxD9aTbiAW19b483XSZPLlku8c+UTW3FLVUK+9fwunza4tigM/ElH+tOcI65/cMaop5tF8vp
ns11yupWTza6yY7mR2q1K9vuXawJ+PkybTL5dZtUFWLTtpVI105Mk+kb9QJqp1TLEf+LsO9vD
VX700qinmyrMbck5tOdlVcjZ59GL/7CdCsfwqT6cUrm+l4trAfzQnTaYHYyZfLpGI8kpr17Aa
6epljTTOrEqc7PGBx/+050giiBXiZ32q7S72Az/dfp4/szrnaxBOdpWcza+xQnz2+QzEpZDKK
tUL3q4N/LmeNNkcjOmCzu6kE0d19E0xtzy2nd/841mJbogf3bqPbfs6E18O8fly+YUy8sSdO7
WCPYd7R53Iqeb77av7R233ee+YWdQHfrr9PHdqec7LzLTSkGtQyPTXWL6DTr4DcSmkskr1grd
rA3+uJ002B2O6oPPCn4+w7rE3CPo93HTJO1MGkfbuECfWVyXW96mzGlLOt78z2mXAeDWwSETZ
eaCHV1s72d7WxX3NeynzCf+4tJF/+fXLo4L5yCD/nY8sYu2G10Ztd7x/l3wd+NnWOFO1OLqsa
S7/eM8LXLv8HTkFrUwqDfFyTq3wc+/K9xAeGmJaZWBCP5t8B518B+J8pbKO9ldMKd6pPGkD/3
g7M9suCeLL8ogkelKMS3cwpvpyWb2skTuf2Z143/a2rjGDSPLBn2q+8JAmWhWkq4Gl62tGhET
Qj5fnmvs2c+/KJSkv2CZ3UTByu/Nx4GdT40yXHvvaBadw7/N72H2wL+egNV6lIV05T587LadW
U/Fjr6LMx8DQENOTvkDyGXTyfU0hXyPBFXs6yQmTMvBPZJcE6brRvfOZtzvoSncwjvxy8Yiwt
aWDq5c1Ul8bwCce9neFuP3Kd7P2t9to3t0xKojED/77N+1l1dJG1j3+9hi0377oNLa1dvCpsx
p48rU2zppfz7Z9ncyeUs5ps2sT27rzQOq+Zv41za+NdL0zeke0zky33dnWoMabP/56e1d/RjX
OsdJjNzz8Cled2cCWls6cg1Z8oJp7Vy6htSPErNpyFs6qSaTIXmo5MiE14/h23P70m3zyvQ30
DfRybG05Nz65lQ+/a27eg1N9dZDjp5dzwaLZiU7lHnqxxbFrChOZykp3jJVCOskJBWnHLyLLg
VsAL/ATVb1xrPmzbcefS9vaeCpk96EeKst8zKwJMHdatCOzVMu66ZJ3sm1/V+JmpfhJOHKQDZ
8X9nX00zswSNDv5c22LmZPreBA9wBfe+DtFMu3LjqNqoCPqRV+PBIh6PdT4ffSEQrT1tVPTdC
Hz+PhcO8AQb+X8jIvO9u7URUqgz5mVPp5pbWLQ71hvAKnzanlnMZ6dh/q5ZXWTl7b38Vbh3r4
2JLjGYrAoZ4BqoI+qsq8dPeH2d81wLG1AXweL6HBIVbetWnUNt9+xbv5yq+2sPtgH0G/h+9+Z
BHHTSvH743WygYGh+jpH6Iq6ONAdz8VAS8eBEUp9/voCg0yo6qMrlAYj8dDTbmX8KDS2hFCRF
j/+zd5va2bWy5bzJxp5RzqCdMde0/Q7+FAzwAAZV4PvQNDVAV89IUHqQn4OdQbJuj3UB3w0dU
fpjM0xOBQhKvvfTGxDbNqg1x8+hzmz6zCI0LrkV7ed9IMQuEIPi+EwkpHX5gpFX66QmFqy/2E
hyJ09A1SXx0ABa/XQ3hokK7+CC/sOUxE4dk327n63JPxiXC4N4wI/GjjG5w1vz4RIO/ftJd/v
eQ0/F4vvQODTKsMMDA0xMyaAG2dA+zrDDG9qoypFT4aplXzVmcfLUf66AoNMq2ijLbufqZXlX

Goe4CqgI+KgIdjaoMMhGHPoR4qAz4GBoeoCpbR0duf9PnCge5+jq0tx+uBPx/pY0q5HxHFg4f
9Xf3MqCqj3O+lttzPnCkV7D7Uy1tHevH7PHT2DTKrNhrYD/X0c6AnzD8n/YK64cJTObGukiN9
A1QF/EQ0wjE15XT0hjnY209NsIxDPf1UBX3MrA4yGBniUM8gh3oGOG5qBTXlPvZ39tMZ+7xn1
gQYHIK2rhDHVAc41BumtSPEnKlBIgr7OkLMqA4wsybA7Nq3r0/VVwfxeqC1I8SxU4J09Q1yuC
9MKDxEw4xKtr7VyT/dvyVR7u9+ZBHLTzmGp3cc4J9/9TIXnz5n2L66+bJ3Ju6biHehsfNgD0G
/l6kVfvxeaOuKLr+xrpKOvkHe6gxRFfBRW+6jfzDCwe4Bjp1SzinH1ODzeUb9ejvS20/A78Xv
9dAfjnBM7dtjFo/1Ky8bRdOOX0S8wA+Bc4G9wB9F5EFVfWWi1pHtT9J0tfrGmVVUlvlSLmvb/
i5+8tQOvnbBKVQGPCmXc/z0cj7zVyex5qGtieVed8FCtu3r4pbHtg+rZfzzr17iqjMbuO3pHa
xZsZAp5QO0d4f55m9eZWpFGZ98z/GJ98TLV1nm5ce/30GZT/jMX53Ed//ntcTr15w7n8Gh/Vx
979tl+er57+DN9h6ue3DrqOVs3NbGeafOYs1DW5laUcbqZY3D1rdqaSNf+dUWPn32iXSHwhw3
rZKAX3h9Xxf9gxFCgxHWbng9Mf9Xli/gYDf89A87uaxp7rBfKl8872Se39nOsnfMGlaW6y5Yy
G+3tnKkd4AdB3qGrf+GC09l9tQALYf7+cHG7VzWNJd7m/eMWvbqZY1U+L08tKWFfzinMZEqmF
Ub5BNLjh8175vt3QR8wpG+IX70xBspl3fnM7s53DvA9SsW8rtXWzlv4bGJFNnx08u5etl8Xtr
bkShvqv1+zbnzOdwb5ku/2JS0TQvZ2d7D15M/gw8tpLUjxPb9Pdz93G4ua5rL1Y8PT9Hd27yH
q953ArsO9A2rPMRfu6xpLpv/fHDU5xvfltlTAny0ae7wz/5DCwn64KWWDm767WujPodrzp1PQ
11lIujHj9uvPfAyK89uIOjzcm/zHj57zkm0dvTz/d+9zsfOOJ6bf/fCsPUfOyXI2g2vJyoP13
1oIf/++zcSz9esWMiPnniDgUFNHPOpjv81KxZSUdbBl36xZdjyn99xkBXvms2+jlBi/muXnzz
qfPun+7dQW+HnrcO9Kc+tY2qCiXP6v1/exxf+3+ZR23HXM7sYGFQ+vuT4UZ9zddDHDze+yeHe
Ab754VNZcdqx/O61tmEx5ivLF9A/NPy8WXvpYsp8MmxEtvh+zfWaVCqFuLXyDOANVd2hqgPAP
cCFE7mCbAfsSDc4xpa9HVSU+VIuS/Xt1MGm3UfYdbBn1HIuWDQ7cfLHl7vm4a1MqyhL+WUiEv
1/3YNbqSkP8M3fRAf7uPj0OaMO3Fse286BngEuPn1OyvWs3fA6r7R2DitLeFATB+jI5Vx55gm
JZbR2hLjzmd3cdMk7WbXsJAivNLMAAA53SURBVK46s4G7nt3N7oN9XP/wK3SEhvj8fZvp7Y/Q
0hHiQM9A4uCNL/dg7wA3PrqNCxbNTgSQ+Gs3/fY1Pr7khFFlWfPwVq488wRakk7a+Gtfe+Blv
OLlaw+8nFhmqmXf8th2DvYO8Mn3NnD9w1tZtTQa/C8+fU7Ked9o76GmPMB1D25Nu7yLT59DKB
zh6w9u5eNLThh2XeSCRbPZebBnWHnT7Y/tbd3DppX7fYmgn/gMHtqKVzys3fB6yvLEt/tAz9u
/GEe+tu7x7Sk/3/i2fPK9DaM/+4e24vf62N7WnXK9aze8TmggkvK4jSiJdX/9wa3sPNDDBYtm
c/PvXh+1/jfbo68lrzf5eXw/JB/zqY7/6x7cOurzvOWx7Vx55gnsPDB8f6Q73/Z3hJhWGUh5b
g3FZt91sCcR9Eduxyff28Cnzj4x5efc1tWfOG7+5dcvs+WtjlEx5mDv6PPmmvs2s2VvR8r9mu
lgP5koROCfDfw56fne2LRhRGSliDSLSHN7e3tWK8h2FKB0vxAiCuGhoVHLWrW0kV/+ae+w+dq
6QqOWEw/kI5dbEUj/ZRKfp727P/HedMuJaPS1sV5PLktP/2Da+Q73hIe91toRYtv+LtY99gY/
3PjGsAFH4uvrGRgkohDR0euPT0tXtiMj1heffrgnnHJ5oXD0F1vyMsfa7r7+QXYf7OOuZ3dz1
ZkNzJ1annbe9q7+MZcXTwGkKrfI6O3PZH9A+v1xKLaOscqT7jOKv2fk/kx+vS/NeuP7M916ew
YGx6wExd831jLir40s08jnye/P9POM75+Rn0268628LJp+TLXs9u63W8ulPcYGBtN+lsnbGQp
H2JdiOen2Yartin8GySOzHY3i6EwlBVVdr6pNqtpUV1eX1XvTDSmX7idSul8IHoFplYHEsn56
ZRMrz24YNvJSfL766mDa5Yx8vvdwb6ImGp+W/GUS9HuoqwoMe2+68sW/LNK9nqwymPoEiG6nf
9RrXkm9XNXo/8oyH15JPV/ytFTLmJJifUG/h2mV/rTrra8Ojlpmuu2Jn+ytHSF+uPEN/nykL+
28ddWBMZeX/BmnKne68qZaV7J0+yN5X6Qrz3j7JtX+jL+eLhBWlvmGDQgz8vVDPQNpj9vkdY+
1jORjNrlMqZ5ncvyPnDYlxfHTcqSX1cuGl3v1skZaj/Sm/Szi2YGxYkN5mS/t+0eem7NSLCfd
Pky1XfHPdqIupBci8LcAxyU9nxObNqGyGac01S+E1csaWTSnNnFBpaGuir+aX8+CY2o43DuQc
r6Ry3noxZZo7jRpudddsJB7/riHu57dzcqzG/i3yxfzo4+fzr3Ne2jtCCXyl519/fzLB99B0B
9t0ZPqwJ1RWcYv/7Q35Xq+99HFLJpTO6wsfq+wZsXClMv56dM7Ry3jxLpKvvR/Th427fMfmM/
DW1pYvawRnxeOrQ0yvbKMa86dP2y+aRVlfHn5Ah56sWVUsPjieSdz97M7R5VlzYqFtHX20Tiz
etT23nDhqQzpEDdceGpimamWvXpZI9MryrjjDzuGbc9DL7bwjQ+N3vYT6yrp7OtnzYqFaZcXD
2zXr1jI3c/u5LoLhi+3oa5yWHkferGFGy48ddhyrjl3PifMqBw2rW9gkOtHfAbXfWghQxrhmn
PnpyzPqqWNPLylemVZaPWEX9t1dJG/jPF5xvflj v+sGPUa9d9aCHhoUEa66tSrveac+dT4Y/
m8b943ny+f9niRCXocO9AYt3Xr1jICTMqeejFFj7/gfkpP++Ht7QMW2/y8zUros+Tj/lUx/+a
FQtprK8atfyfPb2TeTOG74+fP7+HY6cEWXl2A59behIrz26gsszLrCnl3PGHHaO2NTk7MG96J
d/76OjYcGJdJXf+YQe3Pvlmys+5vjqQOG6++eFTOe3Y2lExZlrF6PNm7aXDz9vk/TqRN+flvV
WPiPiA14FlRAP+H4GPqerWdO9xqnfOZPFWPXsO9VCR1KonVfvwseaLX7lv6wpRVzW8Vc/UijL
6BwcR8dDZF6Y66KeuqowhHeJQzxDtXf3U1wRQHaIyUJZo1dPeNUB10IvP46GjL0x5mZcKvxe/
z8O+jhABv5eKMg8+T7S1S7y5GiS3DvAS9AsDYaU3HOFQb7R1SE3Ah88rtHaEmF7lZ3AI9nX2U
18dYEpFtFXI4b4w/eEh6qoCHO4bwCNCTdDPtCo/bZ39BP1e+sMR+pNa9fQPDjG1oozwkNLdHy
bg89IdGmRaZRnd/bFWPUEvA4NKW1c/M6oCDEaGCPi81NcE6AoNcrgnnGgJVFPu5UhvmKDPx5A
qnaHoMkPhQaoCfo70hgn4o616egcG8Xm9zKwu41BvONpipjIARAj6fRzqGaCizEd9dQCPBw50
DbzdqicUZkq5n+5QmJqgn3Ak2qqnriqACHhFCA0OInjoDQ9SURZdX03Qz8BghL7wEHOnVXL8t
GjrmPhxEl9X/FiYVlFGOBKhPrlVT2WsVc/0aKue/Z39hIein8mB7n6mVwXoGxhEYp//gvpqWj
pD0VY9sdYfVQE/HX0Dic83PBRt1TOrNojPI+w90kdN0I/Xo0isVc/0yjIqy7zUJLXqae3oxee
Ntuo5pjaAB6G9O0R1sIyBwQjTKn10hoZo7+6nripAT3+YyjI/SoSZo1r1DFAZ8HJMzfBWPXOm

lFNb4U+06plS7qc+1qqnvTtEfVWAw31h9nWEmDO1HBRaOvqYURlgZu3brXri55rXA/s6Q8yqH
d6qZ970Ssp8QltnP0f6ovt1Zk2AOVOi7z/U04/fO/zcGdmcOLlVz5RyP2W+0a16WjtDVAZ81A
R9DAxFONQdZlZtgFNm1Y5q1VPu99LRN5Bo1TMwGEl73oaHIjnf/JeuVU+hmnOeD3yfaHPO21X
1W2PNn4/Ab4wxk03RNOcEUNVHgEcKsW5jjHG7or24a4wxxhkW+I0xxmUs8BtjjMtY4DfGGJcp
icHWRaQd2F3ocoxjBnCg0IVwgG1XabHtKi1Ob9fxqjrqDtiSCPylQESaUzWbKnW2XaXFtqu0F
Gq7LNVjjDEuY4HfGGNcxgL/xFlf6AI4xLartNh2lZaCbJfl+I0xxmWsxm+MMS5jgd8MM5jgd8YY1zGAn
8OROR2EWkTkZeTpk0TkQ0isj32f2ohy5iLNNv1DRFpEZHNsb/zC1nGbInIcSKyUUReEZGtIrI
6Nr2k99cY21Xq+ysoIs+LyIux7VoiLbosmZjjO36mYjsTNpfi/NSHsvx
Z09Ezga6gTtV9dTYtO8Ch1T1RhH5MjBVa8tZDmzlWa7vgF0q+pNhSxbrkRkFjBLVf8kItAJ
uDDwJWU8P4aY7supbT3lwCVqtotIn7gaWA1cA3wS1W9R0T+T+HiSr3KH/ZbG7N3UbT+R3T+HXhRVX9cyLJmY4zt+gzwsKr+Ip
/lsRp/DlT1SeDQiMiMXAnfEHt9B9CQsKWm2q6Spaquq/in2uAt4legYzyW9v8bYrpKmUmd2xp/7
YnwJLgXhwLMX9lW67CsIC/8SZqaqtscf7gJmFLMwE+5yIbIlgkoqJZJMROYB7wKeyYxLtrxHb
BSW+v0TEKyKbKbgTZgA/AmcERVB2Oz7KUEv+RGbpeqxvfXt2L762YRCeSjLBb4HaDR/NlkyaH9G
DgRWAy0At8rbHFyIyPaRbHFyIyJVwP2wEeBWoidiSYnlVO8H7lbVX8Yml/z/z+Srv
dk2F/xanqEWAj8B5gSmy8bogGzpaCFewoJW3X8ljKTlW1H/gpedpfFvgnzoPAFAfbHHWVwAPFLAs
EyYeHGMuAl5ON28xil1Uuw14VVXXJr1U0vsr3XZNgv1VJyJTYo/LgXOXr/YCFwSm60U91eq7
dqWVPkQotct8rK/rFVPDkTk58A5RLtU3Q9cB/wauA+YS7QL6UtVtaQuIMFzrnOIpg7WKbZrnOIpg0U2AV8Oi
k3XvRE5EzgKeAlIBKb/FWi+fCS3V9jbNfllPb+WkT+WkT04q2XaMX0PlW9XkQkQupkNeAP4mVks
uCWNs1+NAHSDAZuAzSReBnSuPBX5jjHEXS/UYY4zLWOA3xgxK/Mca4
jAV+4woioiLyn0fSLSLiIP57Csv03qTXFARF6KPb/jDN8489qizKTQA5wqIuWq2kf0
pqc7v5U1Z8SvcsSEdkFvF9VD0xUQ0vUQY1xmtX4jZs8BPxB7PHF/NzJ7/dR8sdYu/7tfdTxvy4OfxF0TkDBF5RkReEJE
mfF5HbY49PE5GXH5Esi8sdvx1vx/tbnici2x48dfnhGXkbBGbL5viMh
dsTJsF5G/d/BzMC5ngd+4yT3AX4Wet/sxvAHtZAmwVdUyW/1ZuN0Xf5x7eMdmm8w15KurR2EpcEL
5IJF5DygkWhfK4uBv4iNbwBwEtHO0hbE/j4/BG7GnAl8kejj/B7g6yJy7GFMjLBbh4bk18SzKc
Iaqbol1YXw50dp/slrgDhFpJ2ZsAMmHDEgJRLfA/YAA/RDVf83ezeLPi/29EKFVH4Ekph/4Kwwd6/mE4nFFVL4p9OTy
R9Foj0ZHJxqqBC/BtVf2PYROjy0ruUyaS9DzC8HNwZP8p1p+KcYSleozb3A6sidAk9Ty9sXe
K+MTRaQWWAecDUwXkUtI7X+Av4v1j4+IzBaR+fIzxbRaR+izLdqFEx2adTvSL6Y9Zvt+YjFjgN66iqntVd
V2Kl74LfFtEXmB4Lfxm4Ieq+jpwFWFBjqoCuqr8F/gt4JpbC+QVQnXxthDtfvhZ4AZVVfSvL9x
uTEeud05giUOqD2pvSYjV+Y4xxGavxG2OMy1iN3xhjXMZq/MYY4zIW+I0xxmUs8BtjjMtY4DfGGJex
xxmX+P46ScQc/qPhAAAAAAElFTkSuQmCC\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sb.displot(data['MinTemp'])"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 386
        },
        "id": "yMbMURcS74Te",
        "outputId": "4199f0b5-9b04-4ed4-81df-e18a3975c0c9"
      },
      "execution_count": 29,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x7ff7df5a45d0>"
]
},
"metadata": {},
"execution_count": 29
},
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 360x360 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAWAAAAFgCAYAAACFYaNMAAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAARtklEQVR4nO3dfaxkdX3H8fcHVqoRG0
S3m82yBBWqJbWiWSkiaVDUYLWChoLE2DXBQlM1GqwvbPxo1sWqtjbKKYdsq4hOCaFBE1FoSV0Fo
suloLKBoRoSwCe/EhatNo1v32j3tWbrb7cHf3nvnO3Pt+JTf3PMzM+Q3DvHP23DlnUlVIkibv
kO4BSNJKZYAlqYkBlqQmBliSmhhgSWqyqnsAi3H66afXNddc/uACsH1eEkeJ1GBKQ0WJ0+SMaXtKaYdsq4h0CaFBE1Fo
Ko4BSNJKZYAlqYkBlqQmBliSmhhgSWqyqnsAi3H66afXNddc0zMSTpQ2d3CmdgDvv/++7uHI
ElLbiYCLEnLkQGWpCYGWJKaGGBJamKAJamJAZakJgZYkpoYYElqYoAlqYkBlqQmBliSmhhgSW
pigCWpiQGWJmzd+qNJMrGfdeuP7n7n7K2oOZuB6wtJz8YOtdnHPxDRPb3uUXnDyxbWn/uAcsSU0
MsCQ1McCS1MQAS1ITAzyj/Eu6NPv8FMSM8i/p/p2/+rSM8i/p/p0uwbNcBJ7gB+BvwK2F5VG5IcCVwOHAPcAZxd
VT8ecxySNI0mcQjiaVV1Qtq2GOOYvAq6rquAZ6SVpxOo4SSVpxOo4BnwFsHqY3A2c2jEGS2o0d0ALT
+RrSc4flq2pq2pqnuG6XuBNbu7Y5Lzk2xJsmVVlUXdnQZ7Ttzs4AudXDDbRmUhsb8o0ZzX/DV6uWOGsNDk1
3dsao2AZsANmzYsNvbNSNvbSNIsG3UPuKruHn5n0C4rERpAbUJGwGXIzSUZdbB+qqkxGoClXIQ7Pea
PGE3y38InDbWdiVpVngqsrTcHbLKU9anlKciS8vdju0ObB+4LNuPDqLPEw+u/z6A3C3lWmxo4ALIFqCUVqeDG17KqD5tLVS5ZVBnv
RREVf0EuB54CnBEklXDqOAuycxBkmaNosKcJKnmLmbZLutXJzlimH4+8Sz
szYElaLha7B/yPi1y20Frg+iQ3AV8rF1g/mATgGOGu4Ldmd+J7mDdxUPcKqGcrScvJqr2tTPIU4t
LF6z6TeDQvd23qm4Cnrb4eb8dtwwRHDNHvAE+3M4J97xzzQGrZJUfKl0dbr8dTnNTbGRD
Bpx6Wk3D1Ysdr4ZmEsFPVXkd+AJcDoNhkz9YHIJp4vgk1A4PDRt9+tiRhLIyvIKJ3QnxgEx4QH6rKyBp8Gm6QXB6pd3LCJcB2Jj2g3aVUMqV6d
pVV154TGJEkrwr72gHf6jSSGbGmW3qjFneeUcmScvcYveANw6l2X
0iwpwkj/Z3fKq+uelHY4krRyLPQTx5AXTDwZOA74XHGSGBJOkCLPQTxioXzw0V2PjjKiCRphTj
Qy1H+D+BxYUk6CIs9BvwJ5j/1APMX4vCd8LdKYWvmD0E4cJvdaQVqzvxfDzeTvdAnOr6wHnbwJPyJ+CD3AwD
w0V5vs38FdEeDvxyzEFJ0kqw2G/EOBv8BOEFwyYzChvUlFVoL4K+D1b4uRCN5UVmVOl5hF4uCWPUSgKpIFCzq8Ilp25MlGU3ZbZEqcAwjkX8kr9kb+GfHvG3QqD/hSqsnKY73
wVfCMQDIYSrdkHvEGWpvqqhecHGrcDCB/ypvpwCcBgemiOaxb6UiLhcQFYyIwPCgP53p4w6Xxp7nOTmxw/Iq5vgkYtDVgLuSvJdYqvmeBK8MfP
YRGUhzQ4m9cOqpJBuvbhd8iHwWUvVVcwcuY7N0
3y9vCRpPy32UxCH7Izv4If7cV9J0m4sdg/4miP7DdUfsWjcdg/4miSfBi4b5s8BPjXdjdXd8IdC6ypqtckeQ
FwyrDqy8D7xx6cJC1n+9oDjvjwOc+q+hjjvwOoCq+hjwMYAkjx/W/dGoo5OkZWxfxc3HXVNnuy4clh0zyoq
kaYXYV4CP2Mu6sr6gQtA4esIsnEftatP7r7GR+UfR2C2JLkTHkT6vqPQsLXxiVpu3
zjkX3zCxzV1+wckT29YY9hXgXWcxXgVwEnJ3ka8a8PwxBx8a8LvD4k9W1
edGH5kkLXOLvR7wtcC1I49FklaUA/2eNEnSQTLAktTEAEtSEwMsSU0MsCQ1McCS1KTlT+KkYaKBbn
o1pfnHnUfa8LSWUQWFhtQ6uNP4Thcb9cIVPDmRkB6j9VBoOVuiUNlgdQ64sGtTe9n61yR9fHD3nUaTNeYHdnBgubwNmCyEmCoVZ5Za5W7Z4ovDfP4w0mh8LSvq7eVpaidn1qSMr
yPsn1Sb6V5JtXJksPzLJtUm+Um4zXLJtUm+O/x++FhjKKRpNuYe8HbxQPgR+AS9Ly+CHacm4AMXAddlHV1XV1PHAS8LIkx0cO8

ePiplkHIK0Lqr+V5JtXJksPzLJtUm+O/x++FhjkKRpNuYe8Hbg1VV1PHAS8LIkxwMXAddV1XHAdcO8

JK04owW4qu6pqq8P0z8DbgXWAWcAm4ebbQbOHGsMkjTNJnIMOMkxwBOBG4E1VXXPsOpeYM0e7
nN+ki1JtszNzU1imJI0UaMHOMnhwEeBV1XVTxeuq6oCanf3q6pNVbWhqjasXr167GFK0sSNGu
AkD2I+vu+vqo8Ni+9LsnZYvxbYNuYYJGlajfkpiACXALdW1VsXrLoK2DhMbwSuHGsMkjTNVo3
42E8FXgzcnOQbw7LXA28CPpTkPOBO4OwRxyBJU2u0AFfVl4DsYfVpY21XkmaFZ8JJUhMDLElN
DLAkNTHAktTEAEtSEwMsSU0MsCQ1McCS1MQAS1ITAyxJTQywJDUxwJLUxABLUhMDLElNDLAkN
THAktTEAEtSEwMsSU0MsCQ1McCS1MQAS1ITAyxJTQywJDUxwJLUxABLUhMDLElNDLAkNTHAkt
TEAEtSEwMsSU0MsCQ1McCS1MQAS1ITAyxJTZZ1gNetP5okE/mRpP21qnsAY/rB1rs45+IbJrK
tyy84eSLbkbR8LOs9YEmaZgZYkpoYYElqYoAlqYkBlqQmBliSmhhgSWpigCWpiQGWpCYGWJKa
GGBJajJagJO8L8m2JLcsWHZkkmuTfHf4/fCxti9VHQdcN8xL0oo0W
oCr6ovAj3ZZfAawAaweZjeDJw51vvYladpN+hjwqmY6z5i+F1izpm6pxdjnaBDB5ubmJjM6SZqgtj
/CVVUBtZf1m6pQ1VtWL169QRHJkmTMekA35dkLcDwe9uEty9JU2PSAb4nk2PDzZ8yjJ+E66nqzt3DqtPG2qYkzRLPhJOkJgZYkpoY
YElqYoAlqYkBlqQmBliSmhhgSWpigCWpiQGWpCYGWJKaGGBJamKAJamJAZakJgZYkpoYYElqY
oAlqYkBlqQmBliSmhhgSWpigCWpiQGWpCYGWJKaGGBJamKAJamJAZakJgZYkpoYYElqYoAlqY
kBlqQmBliSmhhgSWpigCWpiQGWpCYGWJKaGGBJamKAJamJAZakJgZYkpoYYElqYoAlqYkBljS
7DllFkon9rFt/9JIOf9WSPpokTdKO7Zxz8Q0T29zlF5y8pI/nHrAkNTHAktTEAEtSEwMsSU0M
sCQ1McCS1KQlwElOT/KdJLcuahjDJLUbeIBTnIo8E/As4HjgXOT/pcUhSt4494BOB26rq9
qr6JfBB4IBB4IyGcUhSq1TVZDeYnAWcXlWvH+Z/D3hOVb1pl/vNA/PD7GrgOxMd6PiOBn7YPYgpYuvxeM3wQ4fcHh1+6A1+HV4Ee+zEzQBc
c0/Zbb8wGf04G4v6pO33Xh1J6KXFWbgE1L9XhJtlTVhqV6vGngc5p+y+35gM9pKXUcgrgrbWL9
g/qhhmSStKB0B/ipwXJJHJTkMeCFwVcM4JJKnVxA9BVNX2JC8HPg0cr
mn7L7fmAz2nJTPyPcJKkeZ4JJ0lNDLAkNVlRAU7yt0kTnuTvKN4ecPu8d0IJbjqdxJ7khy8/C6b
Okez4FI8r4k25LcsmDZKUmuTfLd4ffDD/bwGb6fHX4YPvHXakUFeP
C2qjph+PlU92D21zI/lftpw+syq58xvRTY9cP2FwHXVdVxwCy5lP/nGC230fbgVdX1fH
AScDLhvfQxF+rlRjgWeep3FOqqr4I/GiXxWcAG4brG4AzD/Yco2W7zTgSjgpQdvrucj1ZW7o
jobXaiUG+OVJTvGT/xcrAPuWjC/dVg26wr4TJKvDaehxYfpeYrL
EhyDPBE4EYaXqtlF+Akn01yy27mT5O8Yb/NxetRs2/Y4EGw8a/rA
wuh899Lov3UZLDgY8Cr6qqny5cd6DXaTP+ZVkmZsAKsBnmvFvAyCtcuwMUP7vI
wf6jli72jWhLbVfckWQpI/v+6vqYSgG/ThmVSWckE2E4+M5vJFhfI5Cpvy8Q7xWGtKbPYmTe/+8+FwKD+tJH6UmvkW5x1zubh+G9+G/6g7
PR+4ZU+3nWLL7lTuJA9N8rCd08CzmM3XZneuAjYO0xuBKvHsiRm/X2UJMAlwK1V9dYFqyb+W
q2oM+GS/Avz2wq4A7gggXbG8LGft/PAqdx/1zykg5Lk0cAVw+A7zAyvOz+JySXAacycvylDe
8D/gb4OPAh4Ah4GjgTuDsqpZP4ZP2RF8DzxxSWlxLw6H/zBUHXL8cBd1ndl80fB2e0/qsvgVJ/cHAMp2pTh3A4/I9fcBWvHsHyx/5/yJ9D8/cIOI7/rXHagrAmnvIbkMLwUxEsFBwNV6UCH
I0jQxwJLUxABLUhMDLElNDLAkNTHAmilJKsm/LphflWQPXD/PP2doW4JI9fcBWvHyX5/jD9
2UmMX1rIj6FppiT5OXAb8JSq+t8kzwbeCGytqufu52NdR4p4DnDNPnA
pftXJHkJUneOUxfmuQdSW6S6IcnuSs/b6e5MPDdQJ2Xqf4jTuvU5zk9SUk+neR7Sf
5suM2pSb6Y5JPDdZrfncT4G/4HfY/4Mpj1ZC5wCBd40+5ukOSSDXXYxSfRwF8Dzxg
uCLQFuHDBTf67qk5Sg/uypS4GzmL+O7BsW3OBX+BZBAe4vm...

```
    {
      "cell_type": "code",
      "source": [
        "from sklearn.preprocessing import StandardScaler"
      ],
      "metadata": {
        "id": "SkXXd8U68EWB"
      },
      "execution_count": 30,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "data = data[data['RainTomorrow'].notnull()]"
      ],
      "metadata": {
        "id": "ik0v1axa8MML"
      },
      "execution_count": 31,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [

"data['Pressure9am'].fillna(data['Pressure9am'].mean(),inplace=True)\n",

"data['Pressure3pm'].fillna(data['Pressure3pm'].mean(),inplace=True)"
      ],
      "metadata": {
        "id": "se9zbFit8Q17"
      },
      "execution_count": 32,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "y=data['RainTomorrow']\n",
        "x=data.drop('RainTomorrow',axis=1)"
      ],
      "metadata": {
        "id": "ybec03Rz8c8H"
      },
      "execution_count": 33,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "set(y)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "K_-Qscl98jpk",
```

```
      "outputId": "874b34a0-0ce0-4581-dc8e-5dea36495463"
    },
    "execution_count": 34,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "{'No', 'Yes'}"
          ]
        },
        "metadata": {},
        "execution_count": 34
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "{'No', 'Yes'}"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "eEvddCXW8oiZ",
      "outputId": "1cd93e04-1a4e-434b-f026-b691bdbaafc7"
    },
    "execution_count": 35,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "{'No', 'Yes'}"
          ]
        },
        "metadata": {},
        "execution_count": 35
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "names=x.columns"
    ],
    "metadata": {
      "id": "7g0pU45I9i9d"
    },
    "execution_count": 36,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "names"
    ],
```

```
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "3niS9KzN-27T",
        "outputId": "0868202f-2c11-4660-b208-e1a5a30e046e"
      },
      "execution_count": 37,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "Index(['MinTemp', 'MaxTemp', 'Rainfall', 'WindGustSpeed',
'WindSpeed9am',\n",
              "       'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
'Pressure9am',\n",
              "       'Pressure3pm', 'Temp9am', 'Temp3pm', 'RISK_MM',
'RainToday',\n",
              "       'WindGustDir', 'WindDir9am', 'WindDir3pm'],\n",
              "      dtype='object')"
            ]
          },
          "metadata": {},
          "execution_count": 37
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sc=StandardScaler()"
      ],
      "metadata": {
        "id": "Ma98w1dW-8s6"
      },
      "execution_count": 38,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "from sklearn.preprocessing import LabelEncoder, MinMaxScaler"
      ],
      "metadata": {
        "id": "E-X97eWB_Fkc"
      },
      "execution_count": 39,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "print(len(x),len(y))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
```

```
    },
    "id": "1ObWMYTF_LRH",
    "outputId": "dea8d1ef-c788-499b-82d1-b7c6fbc6c860"
  },
  "execution_count": 40,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "366 366\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "#### 7. Label Encoding"
  ],
  "metadata": {
    "id": "bi3EyueW_dai"
  }
},
{
  "cell_type": "code",
  "source": [
    "## RainToday\tWindGustDir\tWindDir9am\tWindDir3pm\n",
    "\n",
    "LE = LabelEncoder()\n",
    "x['Rainfall'] = LE.fit_transform(x['Rainfall'])\n",
    "x.head()\n",
    "\n",
    "LE = LabelEncoder()\n",
    "x['RainToday'] = LE.fit_transform(x['RainToday'])\n",
    "x.head()\n",
    "\n",
    "LE = LabelEncoder()\n",
    "x['WindGustDir'] = LE.fit_transform(x['WindGustDir'])\n",
    "x.head()\n",
    "\n",
    "LE = LabelEncoder()\n",
    "x['WindDir9am'] = LE.fit_transform(x['WindDir9am'])\n",
    "x.head()\n",
    "\n",
    "LE = LabelEncoder()\n",
    "x['WindDir3pm'] = LE.fit_transform(x['WindDir3pm'])\n",
    "x.head()"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 270
    },
    "id": "kmdfmkLK_V3b",
    "outputId": "77a07d36-c6ba-4013-852e-0d1d91a765c1"
  },
  "execution_count": 41,
```

```
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "   MinTemp  MaxTemp  Rainfall  WindGustSpeed  WindSpeed9am  
WindSpeed3pm  \\\n",
              "0      8.0     24.3         0           30.0           6.0   
20   \n",
              "1     14.0     26.9        16           39.0           4.0   
17   \n",
              "2     13.7     23.4        16           85.0           6.0    
6   \n",
              "3     13.3     15.5        46           54.0          30.0   
24   \n",
              "4      7.6     16.1        13           50.0          20.0   
28   \n",
              "\n",
              "   Humidity9am  Humidity3pm  Pressure9am  Pressure3pm  
Temp9am  Temp3pm  \\\n",
              "0           68           29       1019.7       1015.0     
14.4     23.6   \n",
              "1           80           36       1012.4       1008.4     
17.5     25.7   \n",
              "2           82           69       1009.5       1007.2     
15.4     20.2   \n",
              "3           62           56       1005.5       1007.0     
13.5     14.1   \n",
              "4           68           49       1018.3       1018.5     
11.1     15.4   \n",
              "\n",
              "   RISK_MM  RainToday  WindGustDir  WindDir9am  WindDir3pm  
\n",
              "0      3.6          0            7          12           7  
\n",
              "1      3.6          1            1           0          13  
\n",
              "2     39.8          1            7           3           5  
\n",
              "3      2.8          1            7          14          13  
\n",
              "4      0.0          1           10          10           2  
"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-5f64916f-2be7-4961-b7f9-aaa7ecfe46bb\">\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>MinTemp</th>\n",
"      <th>MaxTemp</th>\n",
"      <th>Rainfall</th>\n",
"      <th>WindGustSpeed</th>\n",
"      <th>WindSpeed9am</th>\n",
"      <th>WindSpeed3pm</th>\n",
"      <th>Humidity9am</th>\n",
"      <th>Humidity3pm</th>\n",
"      <th>Pressure9am</th>\n",
"      <th>Pressure3pm</th>\n",
"      <th>Temp9am</th>\n",
"      <th>Temp3pm</th>\n",
"      <th>RISK_MM</th>\n",
"      <th>RainToday</th>\n",
"      <th>WindGustDir</th>\n",
"      <th>WindDir9am</th>\n",
"      <th>WindDir3pm</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>8.0</td>\n",
"      <td>24.3</td>\n",
"      <td>0</td>\n",
"      <td>30.0</td>\n",
"      <td>6.0</td>\n",
"      <td>20</td>\n",
"      <td>68</td>\n",
"      <td>29</td>\n",
"      <td>1019.7</td>\n",
"      <td>1015.0</td>\n",
"      <td>14.4</td>\n",
"      <td>23.6</td>\n",
"      <td>3.6</td>\n",
"      <td>0</td>\n",
"      <td>7</td>\n",
"      <td>12</td>\n",
"      <td>7</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>14.0</td>\n",
"      <td>26.9</td>\n",
"      <td>16</td>\n",
"      <td>39.0</td>\n",
"      <td>4.0</td>\n",
"      <td>17</td>\n",
"      <td>80</td>\n",
"      <td>36</td>\n",
```

```
"      <td>1012.4</td>\n",
"      <td>1008.4</td>\n",
"      <td>17.5</td>\n",
"      <td>25.7</td>\n",
"      <td>3.6</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>13</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>13.7</td>\n",
"      <td>23.4</td>\n",
"      <td>16</td>\n",
"      <td>85.0</td>\n",
"      <td>6.0</td>\n",
"      <td>6</td>\n",
"      <td>82</td>\n",
"      <td>69</td>\n",
"      <td>1009.5</td>\n",
"      <td>1007.2</td>\n",
"      <td>15.4</td>\n",
"      <td>20.2</td>\n",
"      <td>39.8</td>\n",
"      <td>1</td>\n",
"      <td>7</td>\n",
"      <td>3</td>\n",
"      <td>5</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>13.3</td>\n",
"      <td>15.5</td>\n",
"      <td>46</td>\n",
"      <td>54.0</td>\n",
"      <td>30.0</td>\n",
"      <td>24</td>\n",
"      <td>62</td>\n",
"      <td>56</td>\n",
"      <td>1005.5</td>\n",
"      <td>1007.0</td>\n",
"      <td>13.5</td>\n",
"      <td>14.1</td>\n",
"      <td>2.8</td>\n",
"      <td>1</td>\n",
"      <td>7</td>\n",
"      <td>14</td>\n",
"      <td>13</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>7.6</td>\n",
"      <td>16.1</td>\n",
"      <td>13</td>\n",
"      <td>50.0</td>\n",
"      <td>20.0</td>\n",
"      <td>28</td>\n",
```

```
"      <td>68</td>\n",
"      <td>49</td>\n",
"      <td>1018.3</td>\n",
"      <td>1018.5</td>\n",
"      <td>11.1</td>\n",
"      <td>15.4</td>\n",
"      <td>0.0</td>\n",
"      <td>1</td>\n",
"      <td>10</td>\n",
"      <td>10</td>\n",
"      <td>2</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-5f64916f-2be7-4961-b7f9-aaa7ecfe46bb')\"\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-5f64916f-2be7-4961-
b7f9-aaa7ecfe46bb 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-
5f64916f-2be7-4961-b7f9-aaa7ecfe46bb');\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": 41
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "LE = LabelEncoder()\n",
    "y=pd.DataFrame(y)\n",
    "y = LE.fit_transform(y)"
  ],
  "metadata": {
    "id": "8DPWCMqrAMCX"
  },
  "execution_count": 42,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print(len(x),len(y))"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "vekG3sQlAaal",
    "outputId": "5339ad59-e49f-43a8-9e58-593a20d337a0"
  },
  "execution_count": 43,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "366 366\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "sc=StandardScaler()"
  ],
  "metadata": {
    "id": "mLcB1Rj8Afoo"
  },
  "execution_count": 44,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "x=sc.fit_transform(x)"
  ],
```

```
      "metadata": {
        "id": "1g5Ip11xAmvs"
      },
      "execution_count": 45,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "x[:5]"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "yjNmW7Z2AqNh",
        "outputId": "505779f6-1e0a-4f8b-92db-c446b132feed"
      },
      "execution_count": 46,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "array([[ 1.22047135e-01,  5.61221296e-01, -4.35226892e-
01,\n",
              "         -7.56614508e-01, -4.64338593e-01,  2.27663829e-
01,\n",
              "         -3.07606422e-01, -9.22225313e-01, -1.35035156e-
03,\n",
              "         -2.80219884e-01,  3.63059084e-01,  6.58867279e-
01,\n",
              "          5.14590764e-01, -4.69041576e-01,  1.57759793e-
01,\n",
              "          1.27317675e+00, -1.21658053e-01],\n",
              "        [ 1.11912864e+00,  9.50363081e-01,
1.15373114e+00,\n",
              "         -6.46354102e-02, -7.18644550e-01, -1.11515119e-
01,\n",
              "          6.07090535e-01, -5.06249740e-01, -
1.09464408e+00,\n",
              "         -1.30180025e+00,  9.14353031e-01,  9.75548781e-
01,\n",
              "          5.14590764e-01,  2.13200716e+00, -
1.23917773e+00,\n",
              "         -1.90578418e+00,  1.20092158e+00],\n",
              "        [ 1.06927456e+00,  4.26518370e-01,
1.15373114e+00,\n",
              "          3.47214664e+00, -4.64338593e-01, -
1.35517126e+00,\n",
              "          7.59540028e-01,  1.45477796e+00, -
1.52896624e+00,\n",
              "         -1.48754214e+00,  5.40895841e-01,  1.46144846e-
01,\n",
              "          9.09274370e+00,  2.13200716e+00,  1.57759793e-
01,\n",
              "         -1.11104395e+00, -5.62517929e-01],\n",
```

```
"       [ 1.00280246e+00, -7.55873975e-01,
4.13302745e+00,\n",
              "         1.08866309e+00,  2.58733289e+00,  6.79902427e-
01,\n",
              "        -7.64954901e-01,  6.82251895e-01, -
2.12803129e+00,\n",
              "        -1.51849912e+00,  2.03006003e-01, -7.73739518e-
01,\n",
              "         3.25018324e-01,  2.13200716e+00,  1.57759793e-
01,\n",
              "         1.80300357e+00,  1.20092158e+00],\n",
              "       [ 5.55750346e-02, -6.66072025e-01,  8.55801508e-
01,\n",
              "         7.81116820e-01,  1.31580311e+00,
1.13214102e+00,\n",
              "        -3.07606422e-01,  2.66276323e-01, -2.11023121e-
01,\n",
              "         2.61527281e-01, -2.23802214e-01, -5.77698588e-
01,\n",
              "        -3.38485218e-01,  2.13200716e+00,  8.56228553e-
01,\n",
              "         7.43349925e-01, -1.22380774e+00]])"
            ]
          },
          "metadata": {},
          "execution_count": 46
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "x=pd.DataFrame(x,columns=names)"
      ],
      "metadata": {
        "id": "ERFaMwqrAtDs"
      },
      "execution_count": 47,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "#### 8. Splitting The Data Into Train And Test"
      ],
      "metadata": {
        "id": "vNcti7GRA0z4"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "from sklearn import model_selection"
      ],
      "metadata": {
        "id": "DfM3HN4sA9wY"
      },
      "execution_count": 48,
```

```
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
"x_train,x_test,y_train,y_test=model_selection.train_test_split(x,y,test_
size=0.2,random_state=0)"
      ],
      "metadata": {
        "id": "UcS7g1cDBCda"
      },
      "execution_count": 49,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "# Build the Model"
      ],
      "metadata": {
        "id": "YAlQEkWJBbl2"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "### 9. Training And Testing The Model"
      ],
      "metadata": {
        "id": "OivZ2DGqBbjj"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "from sklearn.tree import DecisionTreeRegressor\n",
        "from sklearn.metrics import
roc_auc_score,classification_report,mean_squared_error,r2_score\n",
        "from sklearn.ensemble import RandomForestClassifier \n",
        "from sklearn.ensemble import GradientBoostingClassifier"
      ],
      "metadata": {
        "id": "OiJnNxkvBLTh"
      },
      "execution_count": 50,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "# create a regressor object\n",
        "dtregressor = DecisionTreeRegressor(random_state = 0)\n",
        "\n",
        "# fit the regressor with X and Y data\n",
        "dtregressor.fit(x_train, y_train)"
      ],
      "metadata": {
```

```
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "0VscdmI0Bzjy",
        "outputId": "327d4339-7a14-43ff-e489-135e1dec5305"
      },
      "execution_count": 51,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "DecisionTreeRegressor(random_state=0)"
            ]
          },
          "metadata": {},
          "execution_count": 51
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "# predicting with regression model with X and Y\n",
        "y_train_pred=dtregressor.predict(x_train)\n",
        "y_test_pred=dtregressor.predict(x_test)"
      ],
      "metadata": {
        "id": "6YQwrp0iB2T5"
      },
      "execution_count": 52,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "#Mean Squared Error and r2 Score\n",

"print(\"MSE\",mean_squared_error(y_train,y_train_pred),mean_squared_erro
r(y_test,y_test_pred))\n",

"print((r2_score(y_train,y_train_pred),(r2_score(y_test_pred,y_test))))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "xxzZlErVCZ7w",
        "outputId": "141a9950-a12c-48ea-86db-39c220730b18"
      },
      "execution_count": 53,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "MSE 0.0 0.0\n",
            "(1.0, 1.0)\n"
```

```
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "acc=[]\n",
    "model=[]"
  ],
  "metadata": {
    "id": "dTLua5ApDi05"
  },
  "execution_count": 54,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "#Accuracy Score\n",
    "model.append('Decision Tree')\n",
    "acc.append(dtregressor.score(x_test,y_test))\n",
    "print(dtregressor.score(x_test,y_test))"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "LS-A2W3eDHqp",
    "outputId": "1af3e4f0-3d87-4252-d846-2a71bce248c2"
  },
  "execution_count": 55,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "1.0\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## Linear Regression"
  ],
  "metadata": {
    "id": "lI_hQJDgDrul"
  }
},
{
  "cell_type": "code",
  "source": [
    "from sklearn.linear_model import LinearRegression  "
  ],
  "metadata": {
    "id": "nMqwLvVODQD6"
```

```
      },
      "execution_count": 56,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "# create a regressor object\n",
        "lregressor= LinearRegression()  \n",
        "\n",
        "# fit the regressor with X and Y data\n",
        "lregressor.fit(x_train, y_train) "
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "MoCycYklDzzZ",
        "outputId": "bbc5ee3d-e75c-4358-bd64-85fc04a4e24b"
      },
      "execution_count": 57,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "LinearRegression()"
            ]
          },
          "metadata": {},
          "execution_count": 57
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "# predicting with regression model with X and Y\n",
        "y_train_pred=lregressor.predict(x_train)\n",
        "y_test_pred=lregressor.predict(x_test)"
      ],
      "metadata": {
        "id": "5tdy1O2OD-bP"
      },
      "execution_count": 58,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "#Mean Squared Error and r2 Score\n",

"print(\"MSE\",mean_squared_error(y_train,y_train_pred),mean_squared_erro
r(y_test,y_test_pred))\n",

"print((r2_score(y_train,y_train_pred),(r2_score(y_test_pred,y_test))))"
      ],
      "metadata": {
```

```
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "E9Fot1MXECv-",
      "outputId": "869355f3-030d-4c74-c056-45bda6d9d827"
    },
    "execution_count": 59,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "MSE 0.06624289712730357 0.058019336840292875\n",
          "(0.5541063882006465, 0.14039428992976688)\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "#Accuracy Score\n",
      "model.append('Linear Regression')\n",
      "acc.append(lregressor.score(x_test,y_test))\n",
      "print(lregressor.score(x_test,y_test))"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "YGuV2UGzEK1z",
      "outputId": "f9699f8c-85b4-4a43-a59d-12571c92f942"
    },
    "execution_count": 60,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "0.5993519690574479\n"
        ]
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "## Random Forest"
    ],
    "metadata": {
      "id": "5hsfOT4yEWu2"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "from sklearn.ensemble import RandomForestRegressor"
    ],
```

```
      "metadata": {
        "id": "xbyPsrONEPMe"
      },
      "execution_count": 61,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "# create a regressor object\n",
        "forest=RandomForestRegressor()\n",
        "\n",
        "# fit the regressor with X and Y data\n",
        "forest.fit(x_train,y_train)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "J2qQzd25Eczy",
        "outputId": "20ed3a5a-1622-4c17-ef7e-ba55ef554c57"
      },
      "execution_count": 62,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "RandomForestRegressor()"
            ]
          },
          "metadata": {},
          "execution_count": 62
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "# predicting with regression model with X and Y\n",
        "y_train_pred=forest.predict(x_train)\n",
        "y_test_pred=forest.predict(x_test)"
      ],
      "metadata": {
        "id": "1GKW3BO2EfGV"
      },
      "execution_count": 63,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "#Mean Squared Error and r2 Score\n",

"print(\"MSE\",mean_squared_error(y_train,y_train_pred),mean_squared_erro
r(y_test,y_test_pred))\n",

"print((r2_score(y_train,y_train_pred),(r2_score(y_test_pred,y_test))))"
```

```
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "QAv0e1EBEn-z",
      "outputId": "66593a9a-28a5-4e4f-9b70-e90231f8ee22"
    },
    "execution_count": 64,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "MSE 8.424657534246575e-05 0.0001689189189189189\n",
          "(0.9994329201863109, 0.9988111274868482)\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "#Accuracy Score\n",
      "model.append('Random Forest')\n",
      "acc.append(forest.score(x_test,y_test))\n",
      "print(forest.score(x_test,y_test))"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "7ZmIqkSUEq8O",
      "outputId": "5bcf8c0a-f1e5-4881-8edb-410b449ed4d2"
    },
    "execution_count": 65,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "0.9988335435056747\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "RFC=RandomForestClassifier()\n",
      "GBC=GradientBoostingClassifier()"
    ],
    "metadata": {
      "id": "XbC6tGlQExSC"
    },
    "execution_count": 66,
    "outputs": []
  },
```

```
{
  "cell_type": "code",
  "source": [
    "np.any(np.isnan(x))"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "kQg11OjBFJdF",
    "outputId": "49effc35-3c26-4b54-d843-deeff619b346"
  },
  "execution_count": 67,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "False"
        ]
      },
      "metadata": {},
      "execution_count": 67
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "GBC.fit(x_train,y_train)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "N04D8_S8FMFH",
    "outputId": "7196e0a4-c574-4c22-8459-207ae587968a"
  },
  "execution_count": 68,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "GradientBoostingClassifier()"
        ]
      },
      "metadata": {},
      "execution_count": 68
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "RFC.fit(x_train,y_train)"
  ],
  "metadata": {
```

```
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "ZU3P-nLVE7Ze",
    "outputId": "de70a340-3ee0-484e-b936-278c0fed83f7"
  },
  "execution_count": 69,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "RandomForestClassifier()"
        ]
      },
      "metadata": {},
      "execution_count": 69
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "p1=RFC.predict(x_train)"
  ],
  "metadata": {
    "id": "o5ZdHkI7tlIR"
  },
  "execution_count": 93,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "p2=RFC.predict(x_test)"
  ],
  "metadata": {
    "id": "pwshk-MKtmKv"
  },
  "execution_count": 94,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "## XGBOOST"
  ],
  "metadata": {
    "id": "vbyXfN0XGM7v"
  }
},
{
  "cell_type": "code",
  "source": [
    "from xgboost import XGBRegressor"
  ],
  "metadata": {
    "id": "VMeAx24jFqOE"
```

```
      },
      "execution_count": 70,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "# create a regressor object\n",
        "xgb = XGBRegressor()\n",
        "\n",
        "# fit the regressor with X and Y data\n",
        "xgb.fit(x_train,y_train)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "gIfHBP9IsATT",
        "outputId": "9c689d71-d1a4-4606-9b9a-4dbf4c6f519c"
      },
      "execution_count": 79,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[07:12:11] WARNING:
/workspace/src/objective/regression_obj.cu:152: reg:linear is now
deprecated in favor of reg:squarederror.\n"
          ]
        },
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "XGBRegressor()"
            ]
          },
          "metadata": {},
          "execution_count": 79
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "# predicting with regression model with X and Y\n",
        "y_train_pred=xgb.predict(x_train)\n",
        "y_test_pred=xgb.predict(x_test)"
      ],
      "metadata": {
        "id": "dASTQifIsflk"
      },
      "execution_count": 80,
      "outputs": []
    },
    {
      "cell_type": "code",
```

```
      "source": [
        "#Mean Squared Error and r2 Score\n",
 "print(\"MSE\",mean_squared_error(y_train,y_train_pred),mean_squared_erro
r(y_test,y_test_pred))\n",
 "print((r2_score(y_train,y_train_pred),(r2_score(y_test_pred,y_test))))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "3zDUwWNgG5qn",
        "outputId": "a8b949b4-0901-43cb-9a57-d31feb983c24"
      },
      "execution_count": 81,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "MSE 3.1535902382624065e-09 3.0907381518983512e-09\n",
            "(0.9999999787725808, 0.9999999786509985)\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "#Accuracy Score\n",
        "model.append('XGB Boost')\n",
        "acc.append(xgb.score(x_test,y_test))\n",
        "print(xgb.score(x_test,y_test))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "nX5Q4BUoHB2x",
        "outputId": "6677c6e7-6dc0-4127-f466-e5d9278b2ced"
      },
      "execution_count": 82,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "0.9999999786571474\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "p1=RFC.predict(x_train)"
      ],
```

```
    "metadata": {
      "id": "rkJlE0QCtejI"
    },
    "execution_count": 91,
    "outputs": []
  },
  {
    "cell_type": "markdown",
    "source": [
      "### 10. Model Evaluation"
    ],
    "metadata": {
      "id": "0UMdfeMCISL1"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "import sklearn.metrics as metrics"
    ],
    "metadata": {
      "id": "HXOHmdVdKU0q"
    },
    "execution_count": 83,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "data={'Algorithms':model,'Accuracy':acc}\n",
      "AC=pd.DataFrame(data)\n",
      "AC"
    ],
    "metadata": {
      "id": "Fix4VMO7KYoy",
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 175
      },
      "outputId": "0c6c8646-4d01-44be-8f3c-07d4e9ff3966"
    },
    "execution_count": 84,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "          Algorithms  Accuracy\n",
            "0      Decision Tree  1.000000\n",
            "1  Linear Regression  0.599352\n",
            "2      Random Forest  0.998834\n",
            "3          XGB Boost  1.000000"
          ],
          "text/html": [
            "\n",
            "  <div id=\"df-1537b794-e2a0-4de5-9c64-d949d7063274\">\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>Algorithms</th>\n",
              "      <th>Accuracy</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>Decision Tree</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>Linear Regression</td>\n",
              "      <td>0.599352</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>Random Forest</td>\n",
              "      <td>0.998834</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>3</th>\n",
              "      <td>XGB Boost</td>\n",
              "      <td>1.000000</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-1537b794-e2a0-4de5-9c64-
d949d7063274')\"\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-1537b794-e2a0-4de5-
9c64-d949d7063274 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-
1537b794-e2a0-4de5-9c64-d949d7063274');\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": 84
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "import matplotlib.pyplot as plt"
      ],
      "metadata": {
        "id": "wzOCWvbQtDzZ"
      },
      "execution_count": 86,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "plt.figure(figsize=[12,8],dpi=100)\n",
        "plt.title('Accuracy Comparison')\n",
        "plt.xlabel('Accuracy')\n",
        "plt.ylabel('Algorithms')\n",
        "sns.barplot(x=acc,y=model,palette='dark')"
      ],
      "metadata": {
        "id": "mnjCmg2yIlnP",
```

"colab": {
"base_uri": "https://localhost:8080/",
"height": 723
},
"outputId": "02c1c509-13b6-44cf-ac0f-c7a645066045"
},
"execution_count": 89,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"<matplotlib.axes._subplots.AxesSubplot at 0x7ff7d5443350>"
]
},
"metadata": {},
"execution_count": 89
},
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 1200x800 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAABE4AAAKxCAYAAABNKOl6AAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAAPYQAAD2EBqD+naQAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzde7yu5Zw/8M+3Iil7G8
fJCEnIIGYchpAQhnE2w2hmhPEbTGKcT0OMUz8MmgxiMIxhyhChIo3GL8Yx1dARKSrSYe8KSa7
fH/e98nisa7fW2mv3rL33+/16Pa+9nvu+7vv63vv63vfqD+vjOlRrLQAAAAD8ti1mXQAAAADASiU4
AQAAAOgQnAAAAAAB0CE4AAAAAAOgQnAAAAAAB2CEwAAAAIAOwQkAAAABAh+AEAAAAoENwAgAAAAhO
AEAYMWrqjOq6n2zrgOAzY/gBAC4WlTVC6YVVdWXZ13LbrGsqiqblxbtOuUWnSnVtrp8Vb6yqk6vp1V1aVV9vapeVtQ
XXnXV9ALCpqtbarGsAADYDVXVskpskuUWSnVtrp8+2oo1HVd01yaeTbJfk35J8fTxx8lyqatvvSPT/L
F1toDZ1Te1aKqtk7yq9qbq9ba5bOuBYDNi+AEANjgqrb/fyqxtKbyqtk7yq1atvvWW2qWz1lySPT/L
FnI+P23VNV1q+rN43oWl1XVVD6rq/VV1g4k216qa/arq1Kr6eVWdU3faqdxv8P3HacE3Xfc
cYj+89cex9VVJVe1UVZ+uqouTfHA8d++oTrL32AcEm+01r70jrqe3iGAOEjC3mYqtovvydSn
bfN8lBU22enuT7SV47nWkn+uqr+d5363SyfDvdh51vwz/HxfQXcXkm+ZzO
dPCDD7+1GY/t/THLPJMdW1S06z3KkeOTrL32AcAm7GtZl0AALBpq6o/zBB+PHM89+P
aPDViabPzxB+PLq19rGJ468ep2wkyyV8+G0Rdvnmjz+ok2i7V1kkNaay+eOv7CqSkiB1XV6
UleW1U3a62d2dOR7/pyS5A8mjqWxpKrbdP6QPfcXqbfWf3DSWYAY+7+67JLk1DF0Wq
fxni/OEDz9cWvtV+Pxk5McMcmOmOmOQvkrx34pkJbrLK9wk8xj1xi6Og
LvrhbWL5uzCRTmshwP8wIVbWU0zL9+iPsbaS6L2+Ip9bFR97K8M78mjEQJXDEBufpKDsZ6TyrNJrMWd+2BCzKBn7VbqQmPyzSRSzRlWbt+PQR2YWQB/BBX8cXK1Ns7eJXthKGCsoKOxbPuuYJ6Jk5
LbcHf9r+P8V2S7zvpR11C8kE8e7Ks89yxWL9akhUzBQwJzeA6VzCkHjlttzNuL7H+fTlnZd4ZSfQbPK5LpDQSKwAQNbZ9XK3HqNC2k2w2hHzDJnHtU8Pn1GsxAMxzWZA3d0LzCWvLqK1cJo2zpIEFL7S0phbMTJGyoWw8WFRzQpMRG7vSyi6AbBdiU9uFw9JtQ3ZdCUsu3yUyoG5LuczY45qT9zuzUHBNY4LLi3YewcSUeBw5eVLt/vEB6PvEFZZo65H7WhdSEObGwrqCNHqgRi4oEVAamuMtZPDkh97vMzRYBaWNxQGu10xPxMWRn2xYlyNa/yZOGunYY9H1XnNt7kPN8xd/Ph0D2txPx6aP0Ib/RaHCmiQZLrBcXoLQnDSdAhjIJAdgaa6qvjCJDR8oDEp4sCxf05L8vtxmEMNGg2cfG7pF1SNLy1rjRYP1Tx6Yn3R3rwyPmLm9UyPXVrdBF9hUPPq+qZbYhVI8zg30p7k41XcTSTEe6lnJ8hPHoEJQ6hVXmz7QxNvN+HqGAeNhDPHXTP7LLA12z1fUc1X0Yz0X1Y6Ksc2WNUfiQYKB6aeqs0qE3y81H+u4xWL7B2FN5jyRi6cMKjU5VLf8p7HFzRl4FGr7gNGUMg6eaGXr5DNnWZVthR4fZYdDPdbU1AhgnhAcqeCBLyiAxUzQLeG1OLfeZ5zsTyWKlL6Rz0jgk4bWS7y0DM4LU2DKbzpbnhp2rrdxzVujKRQQhLLxW3vFhhuttX4ACoHWBn9Rwc8lt8EOxH+aEXIJbvn0wFNwMIL1SNbrKMjzPqSOeNPzTtKcCJVlAHNoNuq2k5Ut6V9OwBUcMR2t7jGHI/jFVMDwoP2PdQ3ZMYgc1/xEiPRbGFKlrEdwYWWnQcNaTYbpOI0XU/6RuhSnQ4HxD7+yuIvtBNr3LP6bVI55dgsWf1MAiIjjXdkKCgkbmzhW0odUtsGj7EbhuUrtNVg6iyQtVaDy2Kx
ZcrTC/ey/o1bJeV3+wMTNm+kdMHtO5JjTSbe3yXaWmDvEO6wGzvBxxWsn70+bXOWg7NXZdeEhNlO0z4cMFDazVZQWD63Osm6dB/pO24IvDg7GhyjFnyLLSVy3c7wmqYVxJLC9d/zEqqlFV/3jf/aVBP9A9tOHAzF6vYGcHXqtFWZ3mH+zrvKwuNlkTkN3Gt7bWb+Ln30YypqKAZlpRCQzRHBHpg6YQrhxq0GWzSPNLqZI8gyBxL4LjZ7XN5m6rlkZumj0TnmCTUAylknb6vwHxC5lcvS2nkKFnPCqT3CHVCVoGSW9nYFbXu1Oc+XNt+h2VT9gZ3Tl7+b1eS9SwGk
8MBczJfO3Wwvp6vHBy9MCXeK9BYvHoHI/m3A7L3DKGBzNtOOcT2oWiaZ5lnP9jwzVNeRaW/+eH+Ctg86P2OP5ynCu+a35LdIsOOF4nW7aQ5P0Zu0cfWbnNgU8lZe6TuHUf0w5UkcBz5n3xgXxQC+cWO45P04PjqXx6aUoA/2/Yyvfb6kTdT47p+0kJz8vZ1QUo4nPzW70/hIpi2/6ynr5tuHgmxDMnaUjxNCj4NNN+wC7KaIyC6yqL+XjWrbZDxLwc5tfo1CtL1Xyg7DqsSx5i0Vwg+33Gt/7TcSx1XhHdnl39/KCTbh+1JTtQCBmMRqx0tC+0eo6jrL9F8v7zi+bVf/pSxOqS4HNAvAb2cLJ6f+1ACv8/PrJtQ5MSjVQMYQ56Ebam+0G/bkLP0J
Ib2mKr/t7ZzbIpsqzmV9sylTxK5YXvPMbbmT2aJqlnV9HTv1bj0MY8rbv1T7ch2lN1wrEoJj31P+DPM2n4Pnf1/dlrA29Y5TZzCYw3xT2iHc0Vf2lKeDbXLpTp/PBqhVCb8CRx
ZqFtUS9z/+H5Yl3kQxRunBeRepvA2GPH3Rkc2C9dfXjR9UqmjF7PLo1DE3mPSPL1AVmUGNb8o2KvGXfk/hZxgqPxfeUkcrqL7eg2C6ucnvxp+Mn/K5xuqNtPmutd+QMgK88gYNyZ4oU3n9bSHHdLTBBDmSpjPhvVk1NTtnB7fNuEJK3xjYqBvnGp4PPHpIwOvSW6kZU0YQzcPbyt7lZlBVAKYYmnvSqVjsLpkEJ13bCrpP9WV5CnQYAN3S5KcT+gx1JLkcfOY+/DLy/Z6YN3PXSyUu2Fq5R9c2bNxwqtJUhMMknIlsyX1Uwo3kX8FN33/MLWO9iXLkuSeVW9rJqjcdMNF9hF02Ok2y5+ZczaoxEVKTT4Ybw9nnoQ/bB7nbV1FvOp/Wbu2drdZxfRfb5kewPa4fB4QKbPb7ZmSamVIgvvt/PNV
FPLhYsyFpBgnZQ0nB1z6GKL6bG02rTD13/Hy/VlrFjn5ev/xkacR2dr+dKXYZlPvUBUHlWIVRwjYjxHbVTxLhx4XKCbb4iM7upUUxfbF6ZqWNFDvV+ky0RgvBw4mzQo0NrzsTiU9SaPi+xgTHCrSIPlKl2PGrTYDs9OxPTZ27XcOlw6ts4iVqfPWrTjhyoO7cTjwyekX1uHdPyH3sNkzMunjJQ2CJU3D1x5eNhBXHV4ykJOqfHNlq1j+sxM8/Of33NWpMW7kfsWUfYaxgafUXDL0iI0+QqYnVIUZPSDcqcfLHJGNv9DXrsNqHbpMIOvnLVKGBqJ4Ym9pCSIGnORHs4OAbvS8ILtqTjyFnNFdPUzONG4vgp1yOxBSX+ycgHKvHFtt3uTN/VSXzTZY9vwMnV15aYD+5C9avPE5GFI4ZB5g3tO6qbtGPPZfOjCqY5EXlKdFVmYebi3YpMh
Yd0cePP5vr6q3TYv/H5IU8GK4dQzrVKOiWvDK7SzU15+xl0oh/nJd4eITSJcJZ2zNwUcsSqSV7nwV1qE4O88h9srqyHjpQ/a5bJVBEpqyk8/Mr2mSBYx/w3+UY4ZN7h7ODSlbRXZeTxTmB5VVJK2XQQKOfq8pMXp9HtpkcCrGxcyItnxd9XbtemRkPxJN+kfqbUJh4ymGEg8p9cDmzEVoK/ZzqWOVmUpNBqm5+5Cr5HvMo2q4zTOBLCH0qWbdq8qGj3Nv9kzVKq

bxinTc2Z912MTkpyg6radur4mVPfLxz/ne+5AdhMCE4AgA3pfkm2zxCenDbxOXg8P+8iseupN
/Jky87xy6ZGZ8wFPp/NMEJj/ySPTLJnhjUvkqX9b6j3J9m9qm5aw0K2f5SrHm2SDFNnbl1V11
xCn1flikUeX8h0qN94/+Ozfi7JDTJMzXlohnc5N9Vq+l0uaAed1tp/ZwjSnpwhuPrrJN+oqr9
eyPUd6/PcAGyirHECAGxIe2VYdHW+RUAfneRRVfW0cSrHd3LVC7x+J8ndq+oarbXLO23mRglc
d+r49GiLdblDklsneWJr7crFYKd3bMkwiiNZ2MK0H86wOOmfJ9kmyeUZpixdlcOS3CPDNKUPX
UXb74//3maitoyhy45JjlpAf4u1c35zlMitMoQhZ4zfH5ZhmtLDp9aA2WN9Ox6n4Lw3yXvHxY
X/O8MisO/Ob76LabdN8pOVtO00ACuXEScAwAZRw7a9j07yydbaR6Y/GRYrvU6GXWOSYfeXXWv
+bXtros0NMuyi0mvz/QwjB+4z1eQZiyh/buTBlSMNxvs/a7JRa+28DH+sP7mqbtapZ67tT5Ic
nmGB1r2SHDEeuyrvSHJOkjdV1a2nT1bVjarqZePXo5L8Ism+U/0/JcOUmE8toL/Fmg7F5hYBP
nz8d753uTrJk9an03Hh1iu11i5JcnqGkCattXMy7MjzxJrYirmqbp9hUd5Pr0//AGw+jDgBAD
aUh2cIRj7ROf8/Gda62CvDyIs3ZFg89ZCqek+G7YavN97naUmOzzDd5a+S/GNV3S3D4p3bZlg
U9Z+TfLy1tqaqDknyzKpqGUap/EmGLWkX6uTxujdW1e9lWGfkMZl/rYt9M+wU9I2qOijD6Itb
ZJiScqeptu/Pr7cV/vuFFNJau3AMkz6d5JvjDj1fH0//QYYRLF8a255XVa/LsIXuEVX1iQwjL
p6RYY2VhUwNWqwdx36OyDAy5i+S/Htr7fjx/GcyhDmHVdU7k2yX5KkZRiJtvx79fruqPp/hXV
yQ5C4Z/vs5cKLN8zMEOF+qqn/JMNLnmRnWcdlvPfoGYDMiOAEANpS9kvw8w8w1ohv6W19quq+lS
Svarq+q2186vq3hm2iX1Uhq1if5xhfYwfjNdcUVUPSfLSJE/IEGacnyG4OHHi9s9Mco0Mgctl
GdZUeX6uevHZudour6qHJTkgw/a+P0/ysQx/lB8/1fb4qvqjJP+Q5Ol5OlJrpVh1MvB+W2HZZhKt
EX6gdJ89Xx5HCnx/AyBzF9mWID1pCSvz0RY0Frbr6r0yzAq580ZQoWDkrxkHdOb1sfjkrxqrO
OXYy3Pn6jnlKp6bJJXJ3ljknMz7GJ0XpL3R6YYZTJ95O8LEMAN9f3UVX14Az/Tb0
qw/SoYzJsNb3QRWgB2MzV+u3cBwDAQlXVVknOTnJYa+0ps65nfVTVfhlGttxwgVOOAGCjZI0T
AICrzyOT3DDDlB0AgM2OEScAAC9TVfdLcvckhyc5pbX2gBmXBACwILbjBQCujjckeUCG7ZWfOtt
SAAAWzogTAAAAgA4jTgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAEDH/wc/RbV0s/XQAAAAAElFTkSuQmCC
AICrzyOT3DDDlB0AYCNgqg4AwAZWVXdPcscM65oc11o7ZsYlAQwvZs65nfVTVfhlGttxwgVOOAGCjZI0T
gDARsIaJwAAAAAdRpwAAAAAdAhOAAAAAADosDsuKUFWV5CZJLp51LQAAAGw2rpPk7La7LaOdUwEJ6
wUN0nyg1kXAQAAwGbnpkl+2DspOGGluDhWtLDtQWtLCJW7t2bXbYYYfkKmY+CE5
YUVatWiU4AQAAYMWwOCwAAABAh+AEAAAAoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAAADoE
JwAAAAAdghMAAACADsEJAAAAQIfgBAAAAKBDcAIAAADQITgBAAAA6Nhq1gXApJvd+YLbed
RkAAAAs0oWnvmXWJWwQRpwAAAAAdAhOAAAAAADoEJwAAAAAdghMAAACADsEJAAAAQIfgBAAAAK
BDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAECH4AQAAAC
gQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwAAAIAOwQkAAABAh+AEAAAA
oENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAAADoEJwAAAAAdghMAAACADsEJAAAAQIfgBAAAA
KBDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAECH4AQAAA
CgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwAAAIAOwQkAAABAh+AEAAA
AoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAAADoEJwAAAAAdghMAAACADsEJAAAAQIfgBAAA
AKBDcHI1qKozqurZyZ90WAAAA2LA22+Ckqt5Ckqt5XVW38XF5VP6qqz1bVk6tuqud/LXZMctAHaLtrUc
8/3OWND9Q0AAAAbm802OBkdkWT7JI8dIqjdmI
MkT5r4ftftfJxlV1zQ1YCwAAAKxom3twcllr7TzZ2yShCh7zzWqqutW1bur6bqr6by
qWltVR1fVrpM3qVdVXq+rnVfXb6ZqrGnRvYxLHpqs5z2YP47vVUf6pL
xr4PrqobT5zfr6q+WVV/OV67Ve1pqs6XFXXV2oyjX
6rqXlX1har4WVWdVUHVNW2E3VsXVVvrKofVtl6n8lAAAsHJs7sHJb2mtHZ3k+k+C
SPnjh8SJIbZQhU/jDJN5n5J8KC8/pdPGYJH+X5G+S7G+S7JzkUl0nK/
hOGXo40mul2T3JHsmuWWS/5hqutN4nz8nedEB3G3k+z8vwDu6c5B++qaqcMo3P+M8kdkzwu
yb2SHDhxzxFYJ7pHk65zt5tpBh0d7J5mVd3pDB2BYQq5A01/AwAAAL0b0bNNRNjEnZ/jjP1V1ryR3S
3Kj1tpl4/nnVdVjyfoPrVx5zifrL
Gev0ryraq6a2vtq2O7LZs3nGzzgfHaly7usa90dGvtXTqXdSST7YWnvLeOi0qto3yTv2E3yTF
V9fQModKTktystXb22oaNaVfXg8fhL5unjc7xAAAAWDGMOJlfJWnzuOnVkvyz7QcnKQdcjfS7TcZmBvDoUegdfbIUv74H2u
6pIkO2O2YY6ZEkd0ryUQXE+5Ak2yT5cAk6eFnYQ4rK
jBjqb429X3XJHtPPF+RGf772Ef77tzFDuLNlkh9+uyeX7+jaa9Lsnric9P1qBcAAAA2CCNO5r5dkg
+NP2+XIYi47zztTJd2j33cmlE3Udd3w8K5TNM8Pyd8fZUoAiCtXO5vNkTgjQDw4E5xW+hC
JDpin7ZkZRudckWEa0xVT5y+Zr4N4GcDKlqu6XYQTFm8dD30jyu0l+
2Vo7o3PZUe3kzPkx5LsPD5jjDbP6ZH1Sa20f3/PzMfS7/EKn7NJ5aR18M2Xq7+8Ps5UlmlHdv72eJ1e
8ntWmunz3eyqo7LMOLkQRq21LL1yNdQEAAMAGtbkbkHJ1tX1e9m+KP/xkkenGHtxQ2ST6d6U5ABoNeA3k
pyaFW9IMMpSW6S3Sa5KFWR1FaNUlc2Z5Y3qVLPY7ST6c4b0pLW2/3SHVbX32N+Xk/w0yV/cY8SHLy/Xn
qOyrDwrEfHHfa2SrDCJVjxn6OzOMSLLdkj2j1ba/u01k6tqg8Me0X9VPTfJcUlu
mCFMOqG19qmrsVYAAADYoFZymJgpmkkY48NpGH9P9v5Bv2I9z/oukH2IfQhh47w4iSUzOEI
zdP8qOxzeeT/GmShyf5ZpKjMywMywo8IMnDWmvfvTXzcc+35EkgvH/o9K8t0Mu9
pcbVprJ2RYr+TWSb6QIRh5VZKzJ5o9yKsPUPY9KYkpyQ5NMLdM0zlAQAAgI1S2ZPdMUjywr
Lp6W23HrW5QAALBIF576lqtutIKsXbs2q1evTpLVrW+bW1vXbV1vXab+4gTAAAAgC4BCQAAAECH4AQAAAC
AACgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwAAAIAOwQkAAABAh+AEA
AAAoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAAADoEJwAAAAAdghMAAACADsEJAAAAQIfgBA

AAAKBDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAECH4AQ
AAACgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwAAAIAOwQkAAABAh+AE
AAAAoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAADoEJwAAAAdghMAAACADsEJAAAAQIfgB
AAAAKBDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAECH4A
QAAACgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0bDXrAmDSmcftn1WrVs26DAAAAEhixAkAAABAl+A
EAAAAoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAADoEJwAAAAdghMAAACADsEJAAAAQIfg
BAAAAKBDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAECH4
AQAAACgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwAAAIAOwQkAAABAh+
AEAAAAoENwAgAAANAhOAEAAADoEJwAAAAAdGw16wJg0gcec8dss5U8DwBYvCcf/t1ZlwDAJsh
fqAAAAAAdghMAAACADsEJAAAAQIfgBAAAAKBDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6
BCcAAAAAHYITAAAAgA7BCQAAAECH4AQAAACgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAO
gQnAAAAAB2CEwAAAIAOwQkAAABAh+AEAAAAoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAAD
oEJwAAAAAdghMAAACADsEJAAAAQIfgBAAAAKBDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA
6BCcAAAAAHYITAAAAgA7BCQAAAECH4AQAAACgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAA
OgQnAAAAAB2CEwAAAIAOwQkAAABAh+AEAAAAoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAA
DoEJwAAAAAdghMAAACADsEJAAAAQIfgBAAAAKBDcAIAAADQITgBAAAA6FjRwUlVtap65Kzr2N
xU1X5V9c1Z1wEAAACzNtPgpKreV1WHrqPJ9kkOv7rqWawx2Jn7rK2qr1bVI2Zd1zJ4Y5L7z7o
IAAAAmLUVPeKktXZua+2yWdZQg63W0eRJGQKeuyQ5NslHquoOG7ima27I+7fWLmmtnb8h+wAA
AICNwYoOTian6lTVLcbvj66q/6qqn1bV8VV1j6lr7VXVXVV1QFVtO3H+L6Lvvqa1V1c
VWdW1X/XlU3mjh/37GfP66qrye5LMm9lHmRWPAc2qSv0+yVZI9Ju63Q1UdXFUXVdUFVfXxqr
rFxPmtxhovqqrzq2r/qvrXyZE4VfX5qjqwqt5SVT9JcuR4/PZVdXhVz6Or6ubjud+Yq1bud+YqlNVW1TVy6vqB1V1WVV9s6oePHF+Qb8bAAAA
2Nis6OCk4zUZppLcKmpST40NyKkqnZKckSS/0xyxySPyxB6HDhx/TUyBy7Jnlkklsked88/
bw+yYuS7JLkhKsqazhKePXX4zHrpEh5Lg4yb2T7JbkkiRHTIwaeWGSvTKMXNktyaqxrmlPHO
+7W5KnVdV1kxyd5LgMo10enOTGSQ4e+94+yYeSvGd8hvsm+ehwqrZKcmiSYzK8p3skOShJ6zz
es5I8N8nzxvZHJvlEVe081a77u5nnfW1d6VavmPkmu0+kbAAAAZmZdU1BWqje21l6j5VFIiTf
SnKrJCcneXGSD7bPa2q9q9k1yTFU9vbX289baeybu9d3x/FerarvW5eKeC5JcK
8lOVbU6wyiRr8zdrLV2ZVKz9PP9LFdk+wx+wx9axzAcVOSY5P8rkkJ1bVIVX11Kr6nbGPCzKMsj
lyfBfPGkeo/JZxNMhNMqzfMunYDCNZJq3rdzPtdRnCm7nPTTvT4AAAAYGY2xhEnLn0RVbZFh5Jh
A2i7JO5McM91Z47rexw5rfZKcl6GwAwHq/xxcWGL7x/r57pnkLEMw1oOHbGOmxFL5LMl5YTt9
i9UKyCGgHNbY9JWMVO5tOqYAh6Nvl2Sp4C9PP7m1mhfeaCKNUWqbMXLxzuH/vTbVJm12dgtJnyp/mHiWJrnHNa62jAWTvJ7J4QhFyCa
a9p5C7x/r57tkhyWYarPtHNaa1dU1Z5J7pnkgRlGtrymqu7eWvtea+1JVXtvea+1JVXVAhik+j0vy6qras
7X2P4usa9K6fje/YVz098qFf6tqPboFAACADWNjHHGyLt9IcrvW2unzfH6R5Jrp/kRa21L7
TWTk5/RMSitda+kiEoeelEPTsn+fE89zpPvvkjYT2u
tHdtae0WSO2dYI+VRwS5ZvRE/Ue11p7XWvtmZCruLnvSnn+N8kOkzVxJVNgV5Dll/eF/HbK
tHdtae0WSO2dYI+VRE/Ue11p7XWvtnkn+N8kT5nmmtUnOzrC2yqTdMoywAQAgAE3WSghOVlfV
naY+OyzxXsvsnuee4A82dqmrnqnpEVc0tDntmhvDvEmmDVc0mtDntmhvDgmVV1y6p6eIaFYpfTW5L8TVX9XpIIPJvlJk
o9X1b2rasdxuNsN5Ns0sDqmpuaso/JXnxWOdtkrv1eDrnLcluV6GNVbuV6GNXbuWlU7VdWlVdVd
29ql5SVXepqpsleXSSGyY5aazjdVV1j6q6eVU9MEPA01vn5A0Z1jV5XFXdpqpen2F60luX/JY
AAABgI7ASpurcN8POMJP+JclfL/ZGrbUTkOqmr3DLu7fCFJJflOkv8Yz59XXsneW2SfTOM2nhe
kk8ssfb5HJHke0le2lp7RlvRtRkRdXD0Og89EMO8f8MMPaI2vH9vsn+d0k709yRYbdbY4cf+5qrZ1dV
buN138mydZJvj/2/6vx/vfJsIDrqvHcc1trh1fVjTOMvnlihhE4w52QIYt7Z6e6ADOuQvCnDCJ
1vJ3l4a+20Bb8VAAAA2AjVuNkLK0RVbZFh5MfBrXbXlHg2zYo2L0K458AE3zzzZbrYSBUADAxub
Jh3931iUAsBFZu3ZtVq9enSSrx2Uq5rUSpx1qrq5hkWbz0mw6iRfTLLsAPTvs6wLAAAAWBlr
nGzufpVk7yRfzbDF7x2SPPC1ltvBAAAALiaGHEyY621s/LbO9YYAAAAK4ARJwAAAAAdghMAA
ACADsEJAAAAQIfgBAAAAKBDcAIAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAA
AAgA7BCQAAAECH4AQAAACgQ3ACAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwA
AAIAOwQkAAABAh+AEAAAAoENwAgAAANAhOAEAAADoEJwAAAAAdAhOAAAAADoEJwAAAAdghMA
AACADsEJAAAAQIfgBAAAAKBDcAIAAADQITgBAAAA6BCcAAAAAHQsOjipqh2q6qYT3+9WpAAPS
v+zvKUBAAAAAzNZSRpz8e5I9Pq5zZPcLrqqurly1gbAAAAAwEwtJTi5fZKvjSvS6t+dBeWXB3Rr
3dM8leSfZeproAAAAAZm4pwck0kS0ZxMvKjY1ZWmZzGQ5yfbLURQAAADASrCU4OSZgNDAQRKaXZZ9LoPqP
SI8bjN0ly/nIVBgAAADBrSwlOXpjkb5jkvOUtWvLQEKLKKMG12Ssx8tQpaxAAAADY6G2l4Oun6SV2XYkvhGm
WdVau3Di1EFJrpslQEAEAJfrps1QEAAADM2KKPV9ydBKwNbOk0nWhdXCj4TjEShrTQO
Zru01q73vKUBgAAADBbSxlx8oEkt0ryyBnXgAAAAK8RSgpN7J7kJ7J7nXxKwAAAAAJukpe
yqc3KSbZa7EAAAAICVZinByToSvKaqdq+q61VwGbZ1OASvKaxdq+q61fVqsnPchcIAAAAMCtLmapz
Zcn2LAgAAAFgJlhKcfDDOKXDJ5UmeEIvDAgAAAJuwjwp0QQnt09y59baXaKtdDAAAAMBKspQ1
Zcn2LAgAAAFgJlhKcfDDJ5UmeEIvDAgAAAJuwpQQnt09y59baKctdDAAAAMBKspQ1Tr6WZIfl

LgQAAABgpVnKiJN/SvLWqnpDkhMzTNu5UmvthOUoDAAAAGDWlhKc/Mf473smjrVYHBYAAADYx
CwlONlx2asAAAAAWIGqNZviMHtVtSrJmjVr1mTVqlWzLgcAAIBN3Nq1a7N69eokWd1aW9trt5
QRJ6mqnZPskeRGmVpgtrX2qqXcEwAAAGClWXRwUlVPTfL2JD9Jcm6GdU3mtCSCEwAAAGCTsJQ
RJy9L8tLW2v7LXQwAAADASmf1LPsAACAASURBVLLFVTf5Lb+T5JDlLgQAAABgpVlKcHJIkgcu
dyEAAAAAK82CpupU1b4TX09P8g9V9UdJTkxy+WTb1toBy1ceAAAAwOwsaDviqvreAu/XWmu3X
L+S2BzZjhgAAICr07JuR9xa23G5CgMAAADYWCx6jZOqenpGQgTTrkJXrkgvWrarJ8GSjUwagUlOrarJ8GT
LDKNQ3rG85QEAAADMzoKDkyTPzjDa5D0ZpuSsmTj3iyRntNa+tIy1AQAAAMzUgoOT1tq/ttlu
TfzF1trlG6wqAAAAgBVgQcFJVa2a2NP4uCbTbVd197Vd197HAAAAABuThY44ubCqtm+t/TjJR
Zl/cdi5RWO3XK7iAAAAAGZpocHJHJ/fLrEC1AAAAAKwoCwpOWmvHJElVbZVk9yTvaa39YE1
MWBgAAADBrWyymcWvtl0men8XtxtxgMAAACwUVpUcDI6OsOwEwAAAIBN2lJGjhye5PVckX09
y6eTJ1tonlqMwAAAAgFlbSnDyz+O/z5nnnFnnF11AAAAgE3GooOT1tpSpvcAAAAAAbHSEIAAAAAd
SwpOqmr3qjqsk4fP5+oqnsvd3EAAAAAs7To4KSq/iLJUUl+muSA8h9fgQFz/9/e+ZT1+2gTZGjaF+8ymbkv
1OttcVdUHVSkoNaa2+eOv6cJE9tre2yjPWxmaiqVFnmrWfFmzJqtWrZp1OQAAAGzi1q5dm9vrVy
fJ6tba2l67pUzVuWWSw+Y5/okkO7Oy7hfgAAAAAr0lKCk7OS3H+e4HXRO4Ssh9ZmXkXVbEebqd
5oKrulOSL47Hdkuyd5FnLVBebqd/f987Z4ppbzroMAAAAFuH7B5O66xI2mEUHJ621t1fVume
m+TPxsMnJXlca3+3jy1kcAAAAAwCwtZcRJWmsfSf/KxZa4FAAAAYEVYSnByQhhoQZKHzXP8Ye5JNk2ycVLLQoAAABgJVj/yLCGyR759Ron
d03ykCRPG7/vmeSY9S8PPAAAAYHYWHZy01t5VVd9Osk+SR2t4vti5fiQ+HT0mye2vti2ObNy1fiQAAAACzs
ZQRJ2mtHZvk2GWuBQAAAGBFWVBwUlWrFnrD1trapZUDAAAAsHIdsdMTJRUnaVbSps2W61URAA
AAwAqx0OBkjwWW2u8NSCwEAAABYaRYUnLTWjUgUjj5mKkr9O8odJDlye0gAAAAAm4ulXlh
V96mqf01yTJHf5X2ytFqeX8e66+wsxZKjXaeE+2oxXeE07y3wzk3fXWq6eeT7J3kKULWJtk4ydZJHtla+/ayczTHla9PVAsTLlKU2A2kMD1y4DxxXO9l
QwsecVJVhyU5Jck2pzxbkDPyj82agvUaQ0XQqAAAAMwttODhprf9pXHhdwpnd0sTy/8NDyVbdnWViEEAqEnBtaqoHAAAAYMVYSnByQ
vF0PmVS2/tGUv+qVhYvB1Bn7qq6t/fKB6D6Glnb6FVlFJlSuJqp2p8FQh6VcmHR6/2p+gi6
aDUtba2cqhrTt5f1lq6hzvfxyfV6dm+2v5kqsbSbn8atc6JYeqaupGJ9U+nHVlYqtgrq1m
qzsrj5qNVx0RQxrr1M4yDpj5TM8GpdeAZW9dyWq3kX1KyN5SWTDfc9ZDqrTmnjqkR7j6U1
SXLkePZ2VXV4VV0y1vrBqrrBxKwPHd/f3PMdVVXbju/qiUlkeMTH65b4LeAYAAABYkTal4ORKV
XX7JPdMMjl64lpJvp7koUln+SgJB+oqrtNXf7EJJcmuXuSFyR5+Vw4UlVbJPnoeN+7J3lakv
2n+t42QwBxZK7JnTJA9IcuBUP/dLcpMk90nyn3CsvTPLJ8bq7J3lHkndW1U2X+A5L8mnk3w
1ya5JnkDkKUleNs/z/iLLJbmkeVlXXTXTJ0kmOS3CXjg5PcOMnB4323T/KhJO+JK8vpJK8
cWx3RJLtx88XO/VtXVWr5j5JrrOU5wwQAAIANaVOagvMnvkxzwhZfmNjZk7kq19ivmp9m
aeqelCSP0VvylnSJ0vlnYnjJ7TWk/kk+myEAuW2SBW7XWd1prl4mnvZYZKnJkzPZtN8vCxtdaaS
5t9klyWFW9sLX2o7vVOaPvYHbrzotvE6yTtNktNaaysX1WkU6sI6Gd2tnbuE6Xi5Gyn3NPX56PkXX9s
ZSfZprbUkJ1fVTZLsWvGvtNkttaay+Yu7iqXpbkuNbaSyPTnRcThSOWPKHX6JwlfA1sa7SyG7TSL1
cuJE258l2bq1du46ak6SFyd5xVW0AQAgJnalIKT/PWZ2MQCtjUTmuP8TD3I6xIaXG8a5k
PeSXDNDwPLTqfCzx4n6r0HT4ZxYz1bDAK4+MAIDgLCXdQDsmuS1Nw/eqTBmetxoqoxkrvq/oWpeReu
9zX1prV1TV+RN998w9995y5fndJ8qUxNJmsY7skNJmsY7SkN01y5nj5s61P32zXJHlNhzJydknwmyeesnSF
VR47fP9Jau/Aq6pz2uiPiH7AAAAwAa1KQUnl7KQUnl7bWTk+uHCFxF2I5X5n5t9Y2Ik2ZRKrm
VpJnZxghcWMt2qvn6X0Gs9eDBCfKNhDNV1cbMRaVGcWE9L7s5Z8c8N9sSGhmT8/7cSvxsMaZqvn6X0feVzL9GlU9+3S3JY5l9U95wx0NkzwSBqs3Z5Z
pLXVNXdW2vfW2inrbXLklw4Wk5lklw2972qFl04AAAAbGib5Bon40S
l3k9x6kbc+KckO4z0Kc5onja7jmudzTid2SXJx1j+z4RpLfT3J
Ga+30qc+lSdIGx7bWXphkzhnUP0vkmy53A8DAAAs7BJIiejjQ5JckxvvxGRU8X46gkpybQCm1je5Ovxzkh0yrOVy2
VbskeWeGRU8X46gkpyb516ratarunePQ1U20+mOTnY5vbV9UeSf4pyCm1je5Ovxzkh0yrOVy2
6p6RIYFaP9xaorQtLcluV6SD1XVXXatqp6p6UFW9t6q2rKq7j7sB3aWbpbk0UlumCGoSZIzkt
yxqm5TVeoqmtssCcEAACADWyTDU5aa7tpenr5/SkJt2i3GDHK62CoSZIzkt
GBWXfneSlU21+muRBGcKyb5b5SIY4b5SIY1QfbJ1WhcDPchSe6W5s4PgMu/T8S4b3sK7rzs4wMmXLDOuX
nJhhStNFGUbNrM2wE9CnM4RIr07y3Nba3AK578owAbpfrNtUNhNsYtidfc9
Im3zBbXNNMHAABgY/L9g06ddQmLtnbt2qxevTpJVrfW1l+vbabbijIjTgAAAADWl+AEAAAAoENwg

AAANAhOAEAAADoEJwAAAAAdAhOAAAAADoEJwAAAAAdghMAAACADsEJAAAAQIfgBAAAAKBDcAI
AAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAECH4AQAAACgQ3AC
AAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwAAAIAOwQkAAABAh+AEAAAAoENwA
gAAANAhOAEAAADoEJwAAAAAdAhOAAAAADoEJwAAAAAdghMAAACADsEJAAAAQIfgBAAAAKBDcA
IAAADQITgBAAAA6BCcAAAAAHQITgAAAAA6BCcAAAAAHYITAAAAgA7BCQAAAECH4AQAAACgQ3A
CAAAA0CE4AQAAAOgQnAAAAAB0CE4AAAAAOgQnAAAAAB2CEwAAAIAOwQkAAABAh+AEAAAAoENw
AgAAANAhOAEAAADoEJwAAAAAdAhOAAAAADqqtTbrGiBVtSrJmjVr1mTVqlWzLgcAgP/f3r1HW
VbVdwL//hAxitDmYaIL16IRiS5NUBkdNBkdo6MDxigZFXSGhRLfrOjgI2gG4wOjZPmIkqgzAQ
Z1ZhDHVwY14wSjiVGDiqKQUQRQ+Qh2mCLD56/+ePcmlyK3l3d1V1VXc3ns9ZdVfecvc/9Va+
9bp/7vfvsA7CL27RpUzZs2JAkG7p706idGScAAAAAA4ITAAAAgAHBCQAAAMCA4AQAAABgQHAC
AAAAMCA4AQAAABgQnAAAAAAAMCE4AAAAABgQnAAAAAAAOCEwAAAIABwQkAAADAgOAEAAAAYEBwA
gAAADAgOAEAAAAYEJwAAAAADAhOAAAAAAAYEJwAAAAADghMAAACAAcEJAAAAwIDgBAAAAGBAcA
IAAAAwIDgBAAAAGBCcAAAAAAwITgAAAAAGBCcAAAAAA4ITAAAAgAHBCQAAAMCA4AQAAABgQHA
CAAAAMLD7WhcA815773vkdrvVWpcBAADANjr+8o1rXcKKMOMEAAAAYEBwAgAAADAgOAEAAAAA
YEJwAAAAADAhOAAAAAAAYEJwAAAAADghMAAACAAcEJAAAAwIDgBAAAAGBAcAIAAAAwIDgBAAAA
GBCcAAAAAAwITgAAAAAGBCcAAAAAA4ITAAAAgAHBCQAAAMCA4AQAAABgQHACAAAAMCA4AQAAAB
gQnAAAAAAMCE4AAAAABgQnAAAAAAOCEwAAAIABwQkAAADAgOAEAAAAYEBwAgAAADAgOAEAAAA
YEJwAAAAADAhOAAAAAAYEJwAAAAADghMAAACAAcEJAAAAwIDgBAAAAGBAcAIAAAAwIDgBAAAA
GBCcAAAAAAwITgAAAAAGBCcAAAAAA4ITAAAAgAHBCQAAAMCA4AQAAABgQHACAAAAMCA4AQAAA
BgQnAAAAAAMCE4AAAAABgQnAAAAAAOCEwAAAIABwQkAAADAgOAEAAAAYEBwAgAAADAgOAEAAA
AYEJwAAAAADAhOAAAAAAYEJwAAAAADghMAAACAAcHJdqiq21TVZ6vqg4u2Z6vqQ4u2b6iqS6rdYu2/qsrqdYu2/
V/XVVXVVVP6mq86vqlKp64FybI6uq5x7XVNXZVXXoErUsq9+OVOVl7zVmXVAG8pk5xnTMq8l1FLbLvfNH9kdGHDKdqK6a1aNuq1Ss
3X1jkiOTPK6qnj6368ObPK6qa6a1X9dpLDkhzR3dclSVX9yyQvSS3JMdx/T3Z/u7n
1Xtv0RJy+0HAAAAbIbgZZDt19zeSHJvkxKq6a1X/dpLDkhzR3dclSVX9yyQvSS3JMdx/T3Z/u7n
/s7rO7+/gk//aWh+3LZ48LkvynJDcl+dWly9lyv6r62ap692zWy4+r6i+rar/5g8xmxvxDVV1
bVRdX1YsX7X9X9+VV1QVT+tqiuq6v2z7acmeUSSF87tNlAQAAYGez+1oXsIs4McTkrwn
yQOSvKa7vzK3//OXWSEDwNb2P8M8q/Ufhq
+lmZu9R6mQ7HYfM9wJHLZSLkS/Nkw5D8xkfMdOWRRaMlN1nTdbJuhQbLhSEOEcWFjyWmdykChykSj1XV/br7+
qp6cjJL3JXlVphkyD0vy9qr6Jb8yO/Q5ELNJFi7Jb8yO/cIk9O1lyXpJXzrZdOajvdk
luN7dpr6392wAAAGC1CE52gO7uqnpeqk8q8lOTJG3kk8GNkSY0aJu8U+Si7r7hhoUNVfMtdmnu38w+31
DVV0z+/32Sa5P8P8uzu/uYSy4mfvnKHM16u
kpdmCl3unuRHST7S3S7S3T9M8u0kX579O/ygqyq5L8uPuvzvnyYzZ5Ka2/fE/meSYJJ/o7tfOjvvONqrpf
T2T3G0r2p+S5IAkz0myZ2Ka2/fD2b4DMq1V8ook76vKMg7ikEv1u2+SG5u2+SG5KctdChu7+f5PzZvo
U2nl103M8k2W82i+WvMoUlF1XVe6rq6h634exd7fZINc4+t+TcDAACAVSU42QGqanW+Yf3qmSFJ6a
Sxyf5fJKTq2o+CLkb6r+u6+MM13H5bbTL5NqZJ8JKkQqnd09yeTHqUme0O9Tfst
5EGZLj+6LNOsma9U1Z228TjXdvemyeTHJXkIumeO9fst
CR3TPL87XipGzNdfrM9/b6W6fKshy7srKqfz3SHn/871+bhi47x8CTfmN1FKN19Q3ef2d0vy
Tw7L5JHjVre12S2yyjTgAAANjpWONNk+70+02U2xyZJd19cVS5N8n8/fvskj/j9Vb0ryparpq
6R5IPZlpX5K6K6ZgpbOdPebBVVVd5n9fvskj0ny2Nx8TZ8TZTN2WK/7r6gqv4yXMyzfJ4Q6ZZ
L38x6/emJF+oquMyYLQ57UJKjMwt9qurrxSe6V5V5G+TXJXk4EwB3Pmz/hcNcFeaA2vFN/G
wAAAKwbgpPtUFWPSPVFSPCKJ/s7h8vbO/u/1xVh2a6ZOfrPMyrYdyhyRXZAogDppdrr
Jg70yXwSTJtZnWFHllpjvgbMnW9HtGkrcm+UiSPWavf3B3xXz+r/UtV9ZRMYctxs+O9srtPnfW
/Osmhme668zOLkM6vLv/Ybb/jUnUnkGy+0zrfly8RJ1AwAAwE6ptnAnXFg5WcxAAsHM53
yu12qyXbAwAAsHM5/vKNa13CNtm0aVM2bNiQJBu6e9OoneVAEOMqDLCHKMTm0aVM2bNiQJBsWTWa4GWucAAAAAAwITgAAAAAGBCcAAAAAA
4ITAAAAgAHBCQAAAMCA4AQAAABgQnAAAAAAMCE4AAAAABgQnAAAAAAAOCEwAAAIABwQkAAADAgOAEAAAAYEBwA
OCEwAAAIABwQkAAADAgOAEAAAAYEBwAgAAADAgOAEAAAAYEJwAAAAADAhOAAAAAAYEJwAAAAA
DghMAAACAAcEJAAAAwIDgBAAAAGBAcAIAAAAwIDgBAAAAGBCcAAAAAAwITgAAAAAGBCcAAAAA
GBCcAAAAAAwITgAAAAAGBCcAAAAAA4ITAAAAgAHBCQAAAMCA4AQAAABgQHACAAAAMCA4AQAAA
AOCEwAAAIABwQkAAADAgOAEAAAAYEBwAgAAADAgOAEAAAAYEJwAAAAADAhOAAAAAAYEJwAAAA
ADghMAAACAAcEJAAAAwIDgBAAAAGBAcAIAAAAwIDgBAAAAGBCcAAAAAAwITgAAAAAGBCcAAAA
AA4ITAAAAgAHBCQAAAMCA4AQAAABgYPe1LgDmHXfht7P33rkdRkAAACQxIwTAAAAgAHBCQAAAMCA4AQAAABgQHACAAAAMCA4AQAAABgQHA
AMCA4AQAAABgQHACAAAAMCA4AQAAABgQnAAAAAAMCE4AAAAABgQnAAAAAAAOCEwAAAIABwQkAA
ADAgOAEAAAAYEBwAgAAADAgOAEAAAAY2H2tC4B5mzZtWusSAAAAuBXY2s+f1d0rXMrar2Tf
KtNS4DAACAW5+7dfd3RjvNOGFnsXH1JfriWhcAy7ZXkn2IMs34Zw6x3xjDrnTHMerdex/B
eSS7dUgPBCTubH3a363Vd6pq4VdjmHXJGGa9M4ZZrISAlAAAAAAAADghMAAACAAcGdhMAAACCAgAGBCcAAAAA4ITAAAAgAHBCQAAAMCA4AQAAABgQnAAAAAAAMCE4AAAAABgQnAAAAAAAOCEwAAAIABwQkAAADAgOAEAAAAYEBwA
AcEJO4trk7x69hPWI2OY9c4YZr0zhlnvjGHWO2OY9c4YZr0zhlnvjGHWu2OY9c4YZr0zhlnvjGH
QAAAMCA4AQAAABgQHDCqqmqmF1TVVVX1o6r6dFU9dOqmWVV0xVWVVDC1t26n7aWT8V8sFRYVXFS
p0VV01e5y51JiHlbat78Nz/Q6rqqqq6qo6rqqq5ccnT
nV9VPquqSqSqpL00Vf3MatUL86rqX1XVG6VV16yeCb84IIb0eeRVfWl2XvwhVV1XVVzXXVVVVVV
K/XVF1qpNrAVQQNpk...

JG/OdHuqByX5SpKPV9UvDto/LMlpSU5O8sAkH07y4aq6/+pUDDe3rWM4ySMzjeF/neSgJJck+
T9Vtc/KVwu3tIwxvNBv3yRvTPLpFS4RtmgZ5xJ7JPmrJPsm+XdJ9k/yrCTfWY16YbFljOGnJX
nDrP19kxyV5KlJ/mhVCoZb2jPTuH3B1jSuqnsm+WiSTyY5IMmfJDmpqh67YhWuELcjZlVU1Vl
JvtDdR8+e75bpg+SJ3f2GzbQ/Pcme3f34uW1/n+Sc7n7uKpUN/9+2juHN9L9NkquSHN3d717R
YmEzljOGZ+P2b5OckuQ3ktypu5f8dglWwjLOJZ6b5KVJfqW7r1/VYmEzljGG/zTJfbv7N+e2v
SnJQ7v71epbNisquokT+ru4WzUqjohySHdff+5be/NdD7xuFUoc4cx44QVN/vG58FJzlzY1t
03zZ4fNOh20Hz7mY9voT2smGWO4cXukOS2STbu8AJhCdsxhl+Z5LvdffLKVghbtswx/IQkn0v
yZ1V1RVWdV1WvmAWCsKqWOYY/m+TBC5fzVNW9khyc5GMrWy3sRd17oAbhV+Icltklyx
aPsVSX5l0Ocug/Z32bGlwVZZzhhe7IQkl+aW/3nAatjmMVvVv55pWvgBK1sabJXlvA/fK8mjk
vz3TB82753k7ZlC7FevTJkwtM1juLv/R1X9Qpr/q6rK9Nntnd3buVhSb6r6vbd/ZM1qG
lZzDgBWGFVdWySwzJNZ/zpWtcDS6mqvZK8J8mzuvt7a1OPLNNuSb6b5pXnvndfXXZ3n57kdUlc8su
6UFWPTPKKJM/PkqUkOqarj1rIuuDUy44TV8L0kNyb5pUXbfynJyXYM+l29je1hJyxnDSZKq
ekmSY5M8uru/ujLlwZK2dQz/cqYFNc+YvO+/BlSa7v7v7hvnt
n0tyV2qao/uvm7HlwlDyxnDr03ynu4+afb83KraM8l/qarXzS71gyZ3Z6DPdpvU02yQx44RVMD
sxOTvJ/MJWu82ef27Q7XPz7Wces4X2sGKWOYZTVS9LclySx3X3F1e6ThhZxj2/AevIZcLpMZ+H
xv/LPq+JfssIlw80s8334M0nuPWu34D5JLhOasNqWMY/GkRxOLIQBFZg57fLfKYz44TV8uYk
76qqLyb5fJIXZbdqLX9Nkqp6p6d5LvdPfLZ+3fmuRvqurFmW5hdViSA5M8e7ULh5ltGsNV9ftJX
pPkaUkurqqFF9Xmu6e5rVrt4yDaM4dklZefNd66qq5Oku2Jtkv02
UPb1vtwmFmW8fwGUmOqaovJzkr0zo9r01yxqKZVLAqquOmcbhgntW1QFJNnb3P1bV65Ps091
HzPa/M8nRVfXHme7Q96kgT0lyyGrWvS0SMITlgV3X16Vd050wfJuyQ5J9O38AuLBd09c4l6d392
du/64zPdq/6CJE90ws5a2dYxnOR5SfZI8v5Fh3p1kletWVwS8sYw7BTWca5xCVV9dgkb0ny1
STfyfTFzAmrWjjMLON9+PgkPfu5T5IrM4Upf7BqRcMijJhd+nmmkcJ0m6+1
tVdUim9+EXJvmnJL/b3R9flWp3oOruta4BBAAAAYKdkjRMAAACAAcEJAAAAAwIDgBAAAAGBAcAI
AAAAwIDgBAAAAGBCcAAAAAAwITgAAAAAGBCcAAAAAA4ITAAAAgAHBCQDADlRVB1XVjVX10bWu
BQDYftXda10DAOuo6pOSnJNkNkqOS7N/N/dl65/ACatCoS07aSvUqZRZkVx8aDMiaQTO1T5tUyzNwDQWeO8fFKIBQcFBgoQRjjTvAmwujO2kynhx0yFmInCVEVVNPEVA/9iILb3oEPkgafC5zTEVW/N/cnpxXIULH9+Ea3mgngzeBNpkMdfoONOMmHNhvZZZYEA35Zl/BdfbnDIRmWBiW2d2Ku
BQDYftXda10DAMAuo6pOSnJNkqOS7N/dl65RHXt093Vr8doAsCsx4wQAYAepqjsmeWqSdyT5
JIjF+3/rar6QlX9tKq+V1Ufmtt3u6o6MB8ao6uG+aqpqtu/Iqrp6UbGeWFU99/xVVXOvf
1uVX0X0ryU9n2x9XVX9X1fer6iNVcnBh1xZfzkeKf9vp7K/xEQmMJjqzqqF9ddHd+5fkYHSqF9W3
VfbuqnEsCsMvznx0AwI7zlCRf9K/f7y/k9zJJTE+7zk/y0JnKoEG6tj6m3SEXEkqzJGHEj
kvsmeU6mmSvb4t5JfJoUmG3bM8mbkxw82bkxoIfSYhT1/k2SfjE9I8mtj/jJbt19c
ZIzkzxj0jes8I8mp3X3X3TNtYHAOvO7mtdAADALuSoTFIFJkvzvBJuSPCLJp5dPSPCLJp5L8QZZL3dvcfzrX/Sp
JU1X0yhS6P6e4zZ/suWsbr75HkiO6+L6tUnul+8JE9Lcuck/6K7N86aXk/XTj
X5aQk76yqY7r72qp6UJIHPntZdQHAOuOGScAADtEVe2S5HlJTu3ub1XV3+yy6PlBtSoBvWQ0k2vmRcpxGC2ww+YtsHsf3v9/W6SzkCGRp9s3zy
A5LcmGnmx/b49nxoMqtrv9l1OBdVlaYkF8923X3Xutb88F5os9uFzZbU+aPT8yySdns1EAYJdnx
gkAwI5xVKzZq0tnV+ckSVH8zSSSW5tqOOTd0r5kurymFm277
7LNDtlydfu7uuq6t1JnlFVH8w0Q+WFS9ZAQ+WFS9QLALsMM04AALZVe2e5IgkL8840g2Ph8WuZ3lwZmcorDk3
1OxojmNupvOyRwz2X5lkr6rac27bAYO282F9fJL9L9kxzf3Z6a5IC2ZzsMO7u4a4Tu
JI9O8vxM4dAH13ptANhVmHECALDA9b8fv6O9Oq3pBpNspLk3yiqr6Z5L2ZzsMO7u4a7u4Tu
vviqnpXklOqZtmUsAAAAUpJREFU6vcyrX1yjS/2N3vS2kh8n+aOqeluSh2bRHXsrkry/S
TPrqrLM12ePc84LN8ookH66qLye5LNPitZd29+eSplu/VlV/n/Sn+SEJKd2d2AB
l2e84ZFbU5L8ookoHK66qlyeS5tZd29+eSpLu/VlV/n+SEJKd091IzZABgl2HGCQD
A9jsqyZmLO2L2S6o83GJE/OdOeac5L8daY1N8L8n7k7w9ydeI/HmmO0+Jktv7Iv09ycKb
KYcnedVSRc3uenNYkdnzLzk21tuZbtR/LZCdkfT7uOAsxKq
rvXugYAAHZyVXckid396+udS0AsJrMOAEAYKiq7lhV90909ydJIT17oeAFhtghMAALbLbkTOc
nE
RTcZkOALdCLtUBAAAAGDDjBAAAAGBAcAIAAAAwIDgBAAAAGBCcAAAAAAwITgAAAAAGBCcAAAA
AA4ITAAAAgAHBCQAAAMDA/wN1sF57w3zQ9wAAAABJRU5ErkJggg==\n"
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "print(metrics.accuracy_score(y_train,p1))"
      ],
      "metadata": {
        "colab": {

```
        "base_uri": "https://localhost:8080/"
      },
      "id": "-rKbACDls1Vx",
      "outputId": "64f21737-bd7e-4a52-9d19-b36af1430302"
    },
    "execution_count": 95,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "1.0\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "print(metrics.accuracy_score(y_test,p2))"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "8SLEtCGitXgp",
      "outputId": "99f17232-62e3-46a3-e5f6-c7915646b51c"
    },
    "execution_count": 96,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "1.0\n"
        ]
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "11. Save The Model"
    ],
    "metadata": {
      "id": "STPvFh9dt5gD"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "import pickle"
    ],
    "metadata": {
      "id": "TdxmR1LWt00Z"
    },
    "execution_count": 97,
    "outputs": []
```

```
    },
    {
      "cell_type": "code",
      "source": [
        "import pickle\n",
        "file_name='/content/drive/My
Drive/Nalaiyathiran/weather.csv.pkl'\n",
        "pkl=open(file_name,'wb')\n",
        "pickle.dump(xgb,pkl)\n",
        "pkl.close()"
      ],
      "metadata": {
        "id": "C4TWmp6dt-YO"
      },
      "execution_count": 101,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "pickle.dump(RFC,open('rainfall.pkl','wb'))\n",
        "pickle.dump(LE,open('encoder.pkl','wb'))\n",
        "pickle.dump(imp_mode,open('imputer.pkl','wb'))\n",
        "pickle.dump(sc,open('scale.pkl','wb'))"
      ],
      "metadata": {
        "id": "AndETpA6uFeP"
      },
      "execution_count": 102,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [],
      "metadata": {
        "id": "MJNwGTh4udwE"
      },
      "execution_count": null,
      "outputs": []
    }
  ]
}
```