```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 7,
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 424
        },
        "id": "QV5bWAnkGyCZ",
        "outputId": "edbd33bd-637c-4b76-b860-99372f59ea17"
      },
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "     Duration  Pulse  Maxpulse  Calories\n",
              "0          60    110       130     409.1\n",
              "1          60    117       145     479.0\n",
              "2          60    103       135     340.0\n",
              "3          45    109       175     282.4\n",
              "4          45    117       148     406.0\n",
              "..        ...    ...       ...       ...\n",
              "164        60    105       140     290.8\n",
              "165        60    110       145     300.0\n",
              "166        60    115       145     310.2\n",
              "167        75    120       150     320.4\n",
              "168        75    125       150     330.4\n",
              "\n",
              "[169 rows x 4 columns]"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-bef0afde-eb17-4507-8cae-563d1d44be44\"
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>Duration</th>\n",
"      <th>Pulse</th>\n",
"      <th>Maxpulse</th>\n",
"      <th>Calories</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>60</td>\n",
"      <td>110</td>\n",
"      <td>130</td>\n",
"      <td>409.1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>60</td>\n",
"      <td>117</td>\n",
"      <td>145</td>\n",
"      <td>479.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>60</td>\n",
"      <td>103</td>\n",
"      <td>135</td>\n",
"      <td>340.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>45</td>\n",
"      <td>109</td>\n",
"      <td>175</td>\n",
"      <td>282.4</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>45</td>\n",
"      <td>117</td>\n",
"      <td>148</td>\n",
"      <td>406.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>...</th>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>164</th>\n",
"      <td>60</td>\n",
"      <td>105</td>\n",
"      <td>140</td>\n",
"      <td>290.8</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>165</th>\n",
"      <td>60</td>\n",
```

```
               "      <td>110</td>\n",
               "      <td>145</td>\n",
               "      <td>300.0</td>\n",
               "    </tr>\n",
               "    <tr>\n",
               "      <th>166</th>\n",
               "      <td>60</td>\n",
               "      <td>115</td>\n",
               "      <td>145</td>\n",
               "      <td>310.2</td>\n",
               "    </tr>\n",
               "    <tr>\n",
               "      <th>167</th>\n",
               "      <td>75</td>\n",
               "      <td>120</td>\n",
               "      <td>150</td>\n",
               "      <td>320.4</td>\n",
               "    </tr>\n",
               "    <tr>\n",
               "      <th>168</th>\n",
               "      <td>75</td>\n",
               "      <td>125</td>\n",
               "      <td>150</td>\n",
               "      <td>330.4</td>\n",
               "    </tr>\n",
               "  </tbody>\n",
               "</table>\n",
               "<p>169 rows Ã 4 columns</p>\n",
               "</div>\n",
               "    <div class=\"colab-df-buttons\">\n",
               "\n",
               "  <div class=\"colab-df-container\">\n",
               "    <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-bef0afde-eb17-4507-8cae-563d1d44be44')\"\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 -960 960 960\">\n",
               "    <path d=\"M120-120v-720h720v720H120Zm60-500h600v-
160H180v160Zm220 220h160v-160H400v160Zm0 220h160v-160H400v160ZM180-400h160v-
160H180v160Zm440 0h160v-160H620v160ZM180-180h160v-160H180v160Zm440 0h160v-
160H620v160Z\"/>\n",
               "  </svg>\n",
               "    </button>\n",
               "\n",
               "  <style>\n",
               "    .colab-df-container {\n",
               "      display:flex;\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",
              "    .colab-df-buttons div {\n",
              "      margin-bottom: 4px;\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-bef0afde-eb17-4507-8cae-
563d1d44be44 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-bef0afde-
eb17-4507-8cae-563d1d44be44');\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",
              "\n",
              "\n",
              "<div id=\"df-a3bc9679-13d4-4f67-bd4c-625f0761e4c8\">\n",
              "  <button class=\"colab-df-quickchart\" onclick=\"quickchart('df-
a3bc9679-13d4-4f67-bd4c-625f0761e4c8')\"\n",
              "            title=\"Suggest charts\"\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",
              "    <g>\n",
              "        <path d=\"M19 3H5c-1.1 0-2 .9-2 2v14c0 1.1.9 2 2 2h14c1.1
0 2-.9 2-2V5c0-1.1-.9-2-2-2zM9 17H7v-7h2v7zm4 0h-2V7h2v10zm4 0h-2v-4h2v4z\"/>\
n",
              "    </g>\n",
              "</svg>\n",
              "  </button>\n",
              "\n",
              "<style>\n",
              "  .colab-df-quickchart {\n",
              "      --bg-color: #E8F0FE;\n",
              "      --fill-color: #1967D2;\n",
              "      --hover-bg-color: #E2EBFA;\n",
              "      --hover-fill-color: #174EA6;\n",
              "      --disabled-fill-color: #AAA;\n",
              "      --disabled-bg-color: #DDD;\n",
              "  }\n",
              "\n",
              "  [theme=dark] .colab-df-quickchart {\n",
              "      --bg-color: #3B4455;\n",
              "      --fill-color: #D2E3FC;\n",
              "      --hover-bg-color: #434B5C;\n",
              "      --hover-fill-color: #FFFFFF;\n",
              "      --disabled-bg-color: #3B4455;\n",
              "      --disabled-fill-color: #666;\n",
              "  }\n",
              "\n",
              "  .colab-df-quickchart {\n",
              "    background-color: var(--bg-color);\n",
              "    border: none;\n",
              "    border-radius: 50%;\n",
              "    cursor: pointer;\n",
              "    display: none;\n",
              "    fill: var(--fill-color);\n",
              "    height: 32px;\n",
              "    padding: 0;\n",
              "    width: 32px;\n",
              "  }\n",
              "\n",
              "  .colab-df-quickchart:hover {\n",
              "    background-color: var(--hover-bg-color);\n",
              "    box-shadow: 0 1px 2px rgba(60, 64, 67, 0.3), 0 1px 3px 1px
rgba(60, 64, 67, 0.15);\n",
              "    fill: var(--button-hover-fill-color);\n",
              "  }\n",
              "\n",
              "  .colab-df-quickchart-complete:disabled,\n",
              "  .colab-df-quickchart-complete:disabled:hover {\n",
              "    background-color: var(--disabled-bg-color);\n",
              "    fill: var(--disabled-fill-color);\n",
              "    box-shadow: none;\n",
              "  }\n",
              "\n",
              "  .colab-df-spinner {\n",
              "    border: 2px solid var(--fill-color);\n",
              "    border-color: transparent;\n",
              "    border-bottom-color: var(--fill-color);\n",
              "    animation:\n",
              "      spin 1s steps(1) infinite;\n",
              "  }\n",
```

```
              "\n",
              "  @keyframes spin {\n",
              "    0% {\n",
              "      border-color: transparent;\n",
              "      border-bottom-color: var(--fill-color);\n",
              "      border-left-color: var(--fill-color);\n",
              "    }\n",
              "    20% {\n",
              "      border-color: transparent;\n",
              "      border-left-color: var(--fill-color);\n",
              "      border-top-color: var(--fill-color);\n",
              "    }\n",
              "    30% {\n",
              "      border-color: transparent;\n",
              "      border-left-color: var(--fill-color);\n",
              "      border-top-color: var(--fill-color);\n",
              "      border-right-color: var(--fill-color);\n",
              "    }\n",
              "    40% {\n",
              "      border-color: transparent;\n",
              "      border-right-color: var(--fill-color);\n",
              "      border-top-color: var(--fill-color);\n",
              "    }\n",
              "    60% {\n",
              "      border-color: transparent;\n",
              "      border-right-color: var(--fill-color);\n",
              "    }\n",
              "    80% {\n",
              "      border-color: transparent;\n",
              "      border-right-color: var(--fill-color);\n",
              "      border-bottom-color: var(--fill-color);\n",
              "    }\n",
              "    90% {\n",
              "      border-color: transparent;\n",
              "      border-bottom-color: var(--fill-color);\n",
              "    }\n",
              "  }\n",
              "</style>\n",
              "\n",
              "  <script>\n",
              "    async function quickchart(key) {\n",
              "      const quickchartButtonEl =\n",
              "        document.querySelector('#' + key + ' button');\n",
              "      quickchartButtonEl.disabled = true;  // To prevent multiple
clicks.\n",
              "      quickchartButtonEl.classList.add('colab-df-spinner');\n",
              "      try {\n",
              "        const charts = await google.colab.kernel.invokeFunction(\
n",
              "            'suggestCharts', [key], {});\n",
              "      } catch (error) {\n",
              "        console.error('Error during call to suggestCharts:',
error);\n",
              "      }\n",
              "      quickchartButtonEl.classList.remove('colab-df-spinner');\
n",
              "      quickchartButtonEl.classList.add('colab-df-quickchart-
complete');\n",
              "    }\n",
              "    (() => {\n",
              "      let quickchartButtonEl =\n",
              "        document.querySelector('#df-a3bc9679-13d4-4f67-bd4c-
625f0761e4c8 button');\n",
              "      quickchartButtonEl.style.display =\n",
```

```
              "        google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "    })();\n",
              "  </script>\n",
              "</div>\n",
              "\n",
              "  <div id=\"id_fe3b3f39-3374-48bb-9d47-26f2095cf232\">\n",
              "    <style>\n",
              "      .colab-df-generate {\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-generate: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-generate {\n",
              "        background-color: #3B4455;\n",
              "        fill: #D2E3FC;\n",
              "      }\n",
              "\n",
              "      [theme=dark] .colab-df-generate: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",
              "    <button class=\"colab-df-generate\"
onclick=\"generateWithVariable('df')\"\n",
              "            title=\"Generate code using this dataframe.\"\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=\"M7,19H8.4L18.45,9,17,7.55,7,17.6ZM5,21V16.75L18.45,3.32a2,2,0,0,1,2.83,0l1.4
,1.43a1.91,1.91,0,0,1,.58,1.4,1.91,1.91,0,0,1-.58,1.4L9.25,21ZM18.45,9,17,7.55Zm
-
12,3A5.31,5.31,0,0,0,4.9,8.1,5.31,5.31,0,0,0,1,6.5,5.31,5.31,0,0,0,4.9,4.9,5.31,
5.31,0,0,0,6.5,1,5.31,5.31,0,0,0,8.1,4.9,5.31,5.31,0,0,0,12,6.5,5.46,5.46,0,0,0,
6.5,12Z\"/>\n",
              "  </svg>\n",
              "    </button>\n",
              "    <script>\n",
              "      (() => {\n",
              "      const buttonEl =\n",
              "        document.querySelector('#id_fe3b3f39-3374-48bb-9d47-
26f2095cf232 button.colab-df-generate');\n",
              "      buttonEl.style.display =\n",
              "        google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "\n",
              "      buttonEl.onclick = () => {\n",
```

```
              "        google.colab.notebook.generateWithVariable('df');\n",
              "      }\n",
              "      })();\n",
              "    </script>\n",
              "  </div>\n",
              "\n",
              "    </div>\n",
              "  </div>\n"
            ],
            "application/vnd.google.colaboratory.intrinsic+json": {
              "type": "dataframe",
              "variable_name": "df",
              "summary": "{\n  \"name\": \"df\",\n  \"rows\": 169,\
n  \"fields\": [\n    {\n      \"column\": \"Duration\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 42,\n        \"min\": 15,\n
\"max\": 300,\n        \"num_unique_values\": 16,\n        \"samples\": [\n
60,\n          45,\n          210\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\
n      \"column\": \"Pulse\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 80,\n
\"max\": 159,\n        \"num_unique_values\": 47,\n        \"samples\": [\n
159,\n          112,\n          153\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Maxpulse\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 16,\n        \"min\": 100,\
n        \"max\": 184,\n        \"num_unique_values\": 57,\n        \"samples\":
[\n          130,\n          127,\n          146\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Calories\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 266.37991924435164,\n        \"min\":
50.3,\n        \"max\": 1860.4,\n        \"num_unique_values\": 142,\
n        \"samples\": [\n          328.0,\n          282.0,\n          229.4\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n
}\n  ]\n}"
            }
          },
          "metadata": {},
          "execution_count": 7
        }
      ],
      "source": [
        "import pandas as pd\n",
        "df =
pd.read_csv('https://raw.githubusercontent.com/yessss28/Estadistica/refs/heads/
main/data.csv')\n",
        "df\n"
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "\n",
        "# a) Establezca una variable dependiente (Y) y una variable dependiente
(X).\n",
        "\n",
        "import pandas as pd\n",
        "df = pd.read_csv
('https://raw.githubusercontent.com/yessss28/Estadistica/refs/heads/main/
data.csv')\n",
        "# eliminar registros cpn valores faltantes\n",
        "df.dropna(inplace=True)\n",
        "\n",
        "X = df['Duration'] # variable independiente\n",
        "Y = df['Calories'] # variable dependiente\n",
```

```
        "\n",
        "# b) Realiza un grÃ¡fico de dispersiÃ³n y la recta de regresiÃ³n
ajustada.\n",
        "import matplotlib.pyplot as plt\n",
        "plt.scatter(X, Y, color = 'pink')\n",
        "plt.xlabel('Duration')\n",
        "plt.ylabel('Calories')\n",
        "plt.title('Scatter Plot')\n",
        "plt.show()\n",
        "# recta de regreciÃ³n lineal.\n",
        "import statsmodels.api as sm\n",
        "X_constant = sm.add_constant(X)\n",
        "model = sm.OLS(Y, X_constant). fit()\n",
        "\n",
        "b0, b1 = model.params\n",
        "Fun = lambda x: b0 + b1 * x\n",
        "\n",
        "Yc = Fun(X)\n",
        "\n",
        "plt.plot(X, Yc, color = 'pink')\n",
        "\n",
        "# C) Calcula el coeficiente de correlaciÃ³n y el coeficiente de
determinaciÃ³n e interpreta los resultados.\n",
        "\n",
        "from scipy.stats import pearsonr\n",
        "\n",
        "r,_ = pearsonr (X, Y)\n",
        "print(f'Coeficiente de correlaciÃ³n: {r:0.4f}/n')\n",
        "print(f'Coeficiente de determinaciÃ³n: {r ** 2: 0.4f}/n')\n",
        "# d) ObtÃ©n un intervalo de confianza de 98% para la pendiente e
interpreta el resultado. Respalda tu conclusiÃ³n usando ANOVA.\n",
        "\n",
        "nivel_de_confianza = 0.98\n",
        "intervalo_de_confianza = model.conf_int(alpha = 1 -
nivel_de_confianza)\n",
        "intervalo_de_confianza_b1 = intervalo_de_confianza.iloc[1]\n",
        "print(f'Intervalo de confianza de {nivel_de_confianza * 100}% para la
pendiente:')\n",
        "print(f'{intervalo_de_confianza_b1 [0] - intervalo_de_confianza_b1[1]:
0.4f}')\n",
        "\n",
        "# Tabla ANOVA\n",
        "from statsmodels.formula.api import ols\n",
        "# Y - X\n",
        "model = ols('Y ~ X', data = df).fit()\n",
        "tabla_anova = sm.stats.anova_lm(model)\n",
        "print(tabla_anova)\n",
        "\n",
        "# e) Verifica los supuestos.\n",
        "resudiales = model.resid\n",
        "plt.scatter(X, resudiales, color = 'black')\n",
        "plt.xlabel('Duration')\n",
        "plt.ylabel('Residuales')\n",
        "plt.title('Residuales vs. Duration')\n",
        "ax = plt.gca()\n",
        "ax.axhline(y = 0, color = 'red', linestyle = '--')\n",
        "plt.show()\n",
        "\n",
        "from scipy.stats import shapiro\n",
        "_, valor_p_shapiro = shapiro(resudiales)\n",
        "print(f'Valor p de Shapiro-Wilk: {valor_p_shapiro: 0.4f}')\n",
        "\n",
        "from statsmodels.stats.api import het_breuschpagan\n",
        "_, valor_p_breuschpagan, _, _ = het_breuschpagan(resudiales,
```

X_constant)\n",
        "print(f'Valor p de Breusch-Pagan: {valor_p_breuschpagan: 0.4f}')\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n",
        "\n"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 1000
        },
        "id": "4nI5jpK4JAxY",
        "outputId": "598364ce-825d-4299-c595-c704e83f53e5"
      },
      "execution_count": 76,
      "outputs": [
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 640x480 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAkQAAAHHCAYAAABeLEexAAAAOnRFWHRTb2Z0d2FyZQBNYXRwbG90bGl
iIHZlcnNpb24zLjEwLjAsIGh0dHBzOi8vbWF0cGxvdGxpYi5vcmcvlHJYcgAAAAlwSFlzAAAPYQAAD2E
BqD+naQAASndJREFUeJzt3Xl8VPW9//
H3JGQGAtkgO4QQAgJBQAXEqCAKskhw2sVWhG5oBa0glKktQjYK1zwotjLpddrEdtiS/
09ECutlB0UIkUwhE2UFIyUJCCQDAHJen5/
jBmZLJPJZDJL5vV8POYR5pzPTL5zHoG8+Z7vYjIMwxAAAAEAQC/
F1AwAAAHyNQAQAAIIegQgAAAQ9AhEAAAh6BCIAABD0CEQAACDoEYgAAEDQIxABAICgRyACAABBj0AEAF
528uRJmUwmrVq1ytdNAfAdAhEAjzl48KAeeOABpaamqnXr1urYsaPuvPNO/frXv2627/nOO+/
otddeq3X89OnTmjdvnrKzs5vte9e9e0fft2mUwm+yMsLExdu3bVI488on/+858e+R67d+/
WvHnzVFRU5JH3A2Lebt379aAAQN04MABTZkyRf/93/+tf//3f1dISIiWLVvWbN/XN/XWSCaP3+
+VwNRtaefflq///3v9cYbb2jMmDFas2jaNBg4cqNOnTzf5vXfv3q358+cTiAAAPa+XrBgBoGf7jP/
5DUVFR2rt3r6Kjox30nTlzxXLl1S27ZtndYMHjxYDzzwgCRp0qRJuuaaa/T0000/T000/
r7bff1pw5c7zRTACNRA8RAI/
Izc1V7969a4UhSYqPj6917A9/+INNuvPFGhYeKyYmRkOGDNNHGjRvt599//32NGTycnJslgsSk9P10s
vvaTKykp7zdChQ/XXv/
5VX331lf02VZcuXQRpnHpFkJ+7eszOnj17NGrUKEVFRSk8PFy33Xabdu3a5dGefPmyWQy6c
iRIxo/
frxiYmJ06623Nvra3HHHHZKkEydO8Oi3bunWrBg8erLZt2yo6Olr33HOPjh496tCeWbNmSZLS0tLsn+vk
yZONbhMAR/QQAfCI1NRUZWVl6dChQ7r22mud1s6fP1/
z5s3TzTffrrAULFshsNmvPnj3aunWrRowYI0UlatWqV2rVrp5kzZ6pdu3bapeaunWr5s6dK6vvVqiVLkiSfvG
LX6i4uFinTp3Sq6++Kklq166devXqpQULFmju3Lma0nWqBg8eLEm6+eabJdm6+eabJdmCx+jRo9W/
f3+9+OKLCgkJ0VtvvvaU77rhDH330kvw+oNWr0aUaHvXv7bv/2bunfvrpdfflmGYT62uTm5kqSOnoUG/
N5s2bNXr0aHXt2lXz5s3Tt99+q1//+te65ZZbtH//nXp0kX333+/vvviC/3xjqC/vvvjiC/3xj3/Uq6+
+qtjYWElSXFxco9sEwDADxg48aNRmhoqBEaGmpkZmYaP/vZz4y///3vRlVcFCuqqrK/ufLly/X+h6PP/
64ER4ebly5csV+bMyYMUqumqvW9+jevbsxcuTIS3Nn3HXnzRUOS8f
DDD7t0DbZt22ZIMlauXGmcPXvWvOH36tPHXv/7V6NKli2EymYy9e/
cahmEYJ06cqNW26667zoiPjzfOnTtnP3bgwAEjJCTEmDDhgn8bbzAEjJCTEeOSRR+zHlixZYkgyTpw44VKbALiGW2YAPOLOO+9

9UVlaW7r77bh04cECLFy/WyJEj1bFjR/3lL3+x161bt05VVVWaO3euQkIc/
wkymUz2P7dp08b+54sXL+qbb77R4MGDdfnyZX3+
+edutzM7O1tffvmlxo8fr3Pnzumbb77RN998o0uXLmnYsGHauXOnqqqqHF7zxBNPNOp7PPbYY4qLi1Ny
crLGjBmjS5cu6e2339aAAQPqrM/Pz1d2drYeffRRtW/f3n68b9++uvPOO/W3v/
2t8R8UQKNwywyAxwwcOFBr165VWVmZDhw4oPfee0+vvvqHnjgAWVnZysjI0O5ubkKCQlRRkaG0/
c6fPiwXnjhBW3dulVWq9XhXF/
PLLLyVJEydOrLemuLhYMTEx9udaWmN+h5z587V4V4MGDFRoaqtjYWPXq1UutWtX/
z+1XX30lSerRo0etc7169dLf//9Viy+
+6NLri4qKdNtttykyMlILFixQenq6Wrdurf3792v27V27Nm1enAao/
q1S5Ys0XXXXXVdnTbt27RyeX91b5Yo+ffpo+PDhbrUPgG8QiA0q+rbRPn5+ZKk9PR0VVVV6ciRI/
UGku3bt+vcuXNau3PT1dkhQZGek3oSU1NVWSdOzYsVrnPv/
8c8XGxtp7h7Aqh7/Un076N5771VISIgWLFhQq6en+vWhoaEKxM//
M//1Pr/du2bVvnLbTqAFFzAcP+/fsrPT1dr7zyikpkSmq97uzZs/V+xuaZs/V+xuaSlJSk6667Tm+//
bZDew8dOqSNGzfqrrvush+rrPtTpFiyzZGEydO1NNPPy2TyaTf//73d0aYAZlLCysVb9y5Urj+uuvNywWixETE2P
cdtttxqZNm+znd+3aZdx0001GmzZtjOTkZPs0fknGtm3jIyDB
atWpVa5r7Z599Ztx///1Ghw4dDIvFYqSmphoPPvnQNqqfdv/
vuu07r6pp2bxiGsXnZnZuOWW24x2rRpY0kzZ2wVmx2x2rRpY0RGRhpjx441hw5Uuv1L730ktGxY0cjTlvtGxY0cjJCSEKfiAh5gMw41Vxg
AAAFoQxhABAICgRyACAABBj0AEAACCHoEIAAAEPQIRAAAIegQiAAAQ9FiY0QVVVVU6ffq0IiIiWDYfAI
AAYRiGLl68qOTkZIWEOO8DIhC54PTp00pJSfF1MwAAgBu+/
vprderUyWkNgcgFERERkmwXNDIy0setAQAArrBarUpJSbH/
HneGQOSC6ttkkZGRBCIAAAKMK8NdGFQNAACCHoEIAAAEPQIRAAAIegQiAAAQ9AhEAAAg6BGIAABA0CMQ
AQCAoEcgAgAAQY9ABAAAgh4rVQMAms4wpOKLUlm5ZA6ToiIkNsNGACEQAQCa5uwF6XieLQXVM4dJ3TpL
cTG+axfQCNwyAwC47+wF6UiuYxiSbM+P5NrOAwGAQAQAcI9h2HqGnDmeZ6sD/ByBCADgnuo5Q9dyVlpmqwP8HIEIAOAeVxGAV5/nMd4JP
0W9/+1v1799f/fr1a7A2ZOztbISEhio+X3KFF3pVlMYxgAPyPbenuoTQd3jvhF/
96181efLkWucWLVqkDh066Prrr9eSJUtUUFhP5eVlaUhQ4bIbDbbj40cOsX6vxepaWlslq
tDg8AAPySYUhFVunMODdtXw/B1iwJSwOx2//bbbysiIkL333+w9q3b6/
du3drzpw5ys/P19KlSyVJBQUFktLc3hNQkKC/VxMTEyt77Vw4ULr/w/mT4JAAAecvaCdxPKiv//
pg5TOrWWYqr/fsN9QuYQLRy5UpNmDBBBVu3djg+c+ZM+fh+5/79u0rs9msxx9/XJmZmTKbzXV+r8zMTIf3WK1WpaSkNO0HAACg+M9aLAS1/
XAsXLpTFYnn8Hre82e2ZM8Fy7UnT5Wk7RWKN5VT24xw2kxt3Hna8Z2VxjA0L3AuZ7uPf5ohrMcbGUkjeW24xnphKJGCIhA9NFHNas2ZN
g7WDBg1CRNxCIAAAIZCaTFM341qbilhkAAAh6BCIAAAA0CMQAQAAoEcgAgAAQY9ABAAAgh6BCIAAAA0CEQAAAIegQgAAAAIOgRiAAAQNAjEAEAAgKBHIAIAAEGPQAQAAAIIegQgAAAQ9AhEAAAh6BCIAAAA0CEQAAAQ
ACDoEYgAAEDQIxABAICgRyACAABBj0AEAACCXitfNwAAAAQxw5CKL0pl5ZI5TO3fu
1NixY5WcnCyTyaQ3Tlnde2lXJKOkgErMdlxMbI3yr6AyWZxanS2HzLrQaMadrHuGFbd2j0PjWdkGxqRNVR/
+vCZMmKDIyEhFR0SrYmKKLHR0dr8uTJtb7HbbTJJKmkpcajJcwR4tUcI4dWSvXXW/dEAAh+HT1h+/
pJju24l/k0EF26dEn9+vXT9/vTNqUhy7v9f8u4cPoxvcXbpD0ucnTJigw4cPKpsyPXqMz28vh6ThJCqiOgh1/
fr127typvptqWvMLDDQBx95E9yvsx8YPb16tDEVDTz8i7Sq5ciqcPU5mAntPgGdM8wd2dRXCkFsHQdQjsFNDkK19YzdLWycttcL4c
n94yGz16tEaPHu20ZXmPu20xmKxKDKDExsc5zR48e1e7R3714NGDBAkvTrX
9ad911l1555RUlJydr9qVK1w08KrK+V2D2Y2iD0J7lZcZ7Z2dr6dKlDsEJA4iWFIx/
Oc1+TmSbHRXXrt95veDqrdv3674+Hj16NFN+bP4fo0R2z4WDEd0Q8Ly2c/l5WVpejoaHsYrqThw4crcJCREe/
bssdcMGTTdMGTJEzQ3ZMFy7UnT5LS0tltVodHgA/AwWxwwFNbATM9kZPZXjNy5EgdO3ZMFy7UnT5U1puq/

MSvw5Eo0aN0u9+9ztt2bJF//mf/6kdO3Zo9OjRqqyslCQVFBQoPj7e4TWtWrVS+/
btVVBQYK9JSEhwqKl+Xl1T08KFCxUVFWV/pKSkePqjAQAQvBoKQ42t8wC/nmX20EMP2f/
cp08f9e3bV+np6dq+fbuGDRvWbN93zpw5mjlzpv251WolFAEA4CnmMM/
WeYBf9xDV1LVrV8XGxur48eOSpMTERJ05c8ahpqysTF+sYmSwWRUZGOjw
AAICHREU0HHYs303B95KACkSnTp3SuXPnlJSUJEnKzMxUUVGR9u3bZ6/
ZunWrqqqqNGjQIHvNzp07HD8XExEx1dekW4gWJd3SjhZsivuuFEa4CnmMM/
bqekQ+DUQlJSXKzs5Wdna2JOnEiRPKzs5WXl6eSkpkpKNGvWLH3yySc6efKktmzZonvuuUfdunXTyJEjJU
m9evXSqFGjNGXKFP3jH//QrdNfD300ENKTk6WJI0fP15ms1mTJ0/
W4cOHtWbNGi1btbtszhlhgAAPCyuBgpI712T5ElzHY8zrudFibDMAyvfserbN+
+Xbfffnut4xMnTST077336rPFS077336rPPPlNRUZGSk5SM1YsQIvfTSSw6DpM+fP6/
p06frgw8+UEhIiMaNG6rRp2rt3r2JjY/
XUU09p9uzZZLrfTarUqKipKxcXF3D4DAMCTmnGl6sb8/vZpIAoUBCIAAAAJPY35/
B9QYIgAAgOZAIAIAAEGPQAQAAIIegQgAAAIegQgAAAQ9AhEAAAh6BCIAABD0CEQAACDoEYgAAEDQIxABAICgRyACAABBj0AEAACCHoEIAAAEPQIRAAAIej4NRDt37tTYsWOV
nJwsk8mkdevW2c+Vl5dr9uzZZ6tOnj9q2bavk5GQ98sgjOn36tMN7dOnSRSaTyeGxaNEih5cNEih5qcnBwNHjxY
rVu3VkpKihYvXuyuyNjwcAAAKETwPRpUuX1K9fPy1fvrzWucuL2v//v//v365fvrzWucuXL2v//v365S9/
qf3792vt2rU6duyaY7r777lq1K/vzZ2rdvn5YsWaJ58+bpzTfftuFDFCqamp2rdvn5YsWaJ58+bpjTfeaNbP
BgAAAkcrX37z7z0aNHa/To0Ha/F+jAPP/0WTXQ/fd//
7duvPFG5eXlKXz58pKkqKgoJWYmur0fbdEvPzgOQTDKjh3tBh/tVC7xfkAdtMdgr6SKK0R/
dWdna2li5dqqlTp3ruwwAAgIAVUGOIhHHr8gzAw+Msqa6r5v7KLTZF6/KuzAmOLiwHXRaQCDawcmpo+UrVoJvFZBEJaySuBTrkqXPOxsfPAnVWk7DaLs/
99NO64YYb1L59f6e+3Vtz5sxRfn6ksEBqLy8XA8++KPnEsKnmSL/
ZPNPSE6YcPaX+Pf3rUZEMUl26FFS1tUc0IV2B9KSTrthBz9P8LLw/
ZQHB/zrU2UCTj86zMACTRUwbNqkcgCNDdJmxLTexsuoYNUQFdLewGjRbxOFO5jh4K2/
XeXc9ndhH+EP/5YwzXlnaDdxo4vmW+KSNPwm7rMlqsg/HxjH1KzCanYyFGVmQDr6WOsfaRJCIBhWCXsMTm79YESQIqomMF6d0h2bYb9iKU+xggZBDVhQBMrJ/
QGEEYVjn5n9jFZt+jsbLqP4aPyLZkIIocVHqFfG8s7RNBWRNiwVWXQ61D+vcUHTbzwXXY7Vz8bGsWlpbncwZbU6sNOQj0EXUqKnrGvEoaWLOXgNbXeBwLh4xA8Vx18yCpiAWKCQRSuhbwPymcSdxvJbLFjp5J+x7xmj0sCmIxKDojOxFzR6p9eaZe1aEUxUzxLD0TDd3HFa1NJvEOY8M1Qi5q1iubAI0uVl7QsSN3r20qdH8tuTBB8Aycwxa3kpCM4+
PPQ2jUSBZ4TqLHW2ylUJ5Xc0e8AmOf6yNwvZhgGOzOvx8x3yn1uIj4WGxQ5Qr5IDkJYQm1m9Kn/dvegFvwMLbq7WyBrGkWrpc3Tz4eT5bVnwGTgoD4cMHP0/
/CFmHMIvTiW/8hBuanOaWZ6cIgoKTwxZg44vCD1W00i3YFcSkG6t/
YK9otQfwsQs9uxaXpmMrJJRm97f1LCRTjhr3GbxBXT/Ms3EoIPM+cbk0bD1V2vKM5fpeCJxzhVcWm5M/
r1Jo9HLoM5htlgC1dBAyYZhfPJuL90rlZxEXSB7hv2Mn7/
Px9eK1qYr3szKn1Yea/rzfefzWgx7dhFxNTFu+cyvnCHDVHK3WUx/
rzYyxwIGqSBqtsgZWBTJ9ijQq7ByfTLwK+s4W5PGCW5RyTUpuMmtgTWHzC6Efa0cT9kGmt7uhHm0Ai+s6+
+a6oQCIHb6TkN4aXxzkvB3A1Bsvbx2kCxjmEPQnt6ivh17DU/qX0YeVMZibbdOWmgRzFNqpn3BwUF3tMd9QWfQ5lXPrNTw/
fNJZ7W8xyc55w0t1Dw6EaV2yA0f7wAA1Z3wtXzTPQx7d9YY4Q19j8ySqT5kqdxI+mKyc4Mz6VQzeSBspCNN4gmNHaM2rBzxn5b9cMfn8GpQ1
bmfmqqqusJ6P/pjRz9yjuxSOTL6Js70Z0ngcfqi5u8vwy/
XKAVVK9opY6r7Fc+9QEbD4+ZpzGMWgYTSMZC1UbLoxzLMRZU/
hGP4e5PGaXxLMKHFkEdzW/QpxPHsS0Wd/2Lpy9xo30dVjyvFqT7bqiQ3tSh3c8nb6kSSfv3JVNQhEr5YwnK3Xm1XTrSLUNiP+FIbrIT7m0dL5C/
D0LUX69pv8IqxA9nNGy4MjMf/yMEpZmqNsNXpZLpbgCr4bHcVs2nyC/
4M4hMM/Dy/HUQOH+X8e9x3B8wm7nT/
5OIM7KuCq/E8PFZ2C4ypPl3fbf9oviZJQXgUD2fe/x/
ZI2bgG/d2tPCrLk9Dv69+vKqjf2CKHN67xGNlNUeOpHJoe6pgxxNOiRE0A5l//
uMhzXFr/
Q7g9+BcL8m5fXk8zZ2yKJJkKFnHtHwOoZFQlRFkU8aEhPPqWmfL1Eg9+x+IklzSmzrLcpCSydjnb5PpCv/
dhYMBgVTk4PzYBkRMx8TmBeiCZfZmkV4PBAI2Vhrob9EUw2VqaXgYMxrVpDyzTeCsGKrkz3sdRsOwBG93ABEhMHNN
W8zLkF2vbz5+zjD+H/EsPrcVhVanNoBT0sPg6PEd/
A30pdSfpR9ZrmabdLJ2Ex8WjhFbJVo2dTf1h1kzHtr7bfHgYhmvF8XWr5UV//
OfsC28zEvzwOIVXCz2wKtwCLT2d5HzR+ZEuEgPr//
8nd1nDU8CmemgY9lWsgOE9p5mNrUmatEPU9/
Rs8PsfNYbLW3ha9zM2Qm43HKztm/Ac3/vlgKFAY9OtCbmRYoQoeSdu+Ml+Xs//
kJD9FItLBBPgzsmUY+rD+vDMs0Bd8+IFKmdNcdVm1zxljPnQEmozSYAqQWhLQxmaw6f5Ig/
bVX1tx6CB7xuHxu2Lxg8nvoirAJ3tvhgK4jrtXv1MrT3hSmuFWp58m4R+uCMx9Tl+8aNkdHOQ8tKDdqNR0v6jcpVXyKqWb4zQ6+rYpUUhvVKD2f4PVCP8Mu82B1JkwtgjYrfj8KTjrnKoeR3PXKxbv7+v//8M+v7tAdM8DnTdXvVHtLyuuT+PRY/
FXmXjt+zX4ZqgKsCqbcOkwNf6WRN3L7GXHrcWFKjCdJkQDEBJMNLDETMAj3EjfGTBU+zh5YYcIfpKkp7BpmKFDjgzC1nxqX2Eg+i3CpKA7ODp4dgJUO5bD8k1nHzYZyPz3I9dYH4UcOPoRLkWUsI8OTiXiaw/RnIoE8DudU2ff6Hv+ay/
uQ/RTpHxlXrUJthbfhjXbW8+/fyhfWDQ8y0/C6dTrl0/
Vi9jMFzXrWlvsY3qWXK7cvk0FR6LBRKOb8OSkrAOQ1mO8D5fZLtqB9hrnJVUe7jW/ZbaQ01X7FL2KK/
Rfy6OY1o2mHqRgk36s/pBM47WMpJFcNAz/1Ib20wuHyy+eMhPbXV+Ex2ilBQHUkLdDo/
j8GhVpfeb8pNhnr0qdSbF3W1ZFRAvaGHgLL8V0Q+Rj0bHdLU5OeX1L4y+Ewry5BUM1pLyXd2e4/+
+qQTbpZEomK+rgJfsB7t+fIuyE52bdHmaWxeznb+ueH1M/
LQSM8s9egVW9tj8/1+9Nl/0tWF8fezXk5l/
qcZD6bn1xdeHHo0nsdPvbPzrFyaL7rqTbcF2rhYTupq9QX7kz9b7b+/
wDj3/3V6/5t39SNXm0YvfrF+4dsclrW8kUE2/vLr+7EH7dX8vVFJtsGcXRBqPWQvRAKO0VzecjLKeYYCh//
0Lm7T4LMrP2aKOKJC5oAhqlDqK7OBXMQLBUvw7TT10o9Vv5OkE4cjKvo/
fFy8Pkac3u6xK+rkqNdl5Ae9u0XWAP30+JSSvIyozcsj1fXGjfoOnVkrB7qW19x1+dHPYzjv//
F7/pd/e99F8Np3gPcdeAd46nKqwVaEDj54Zt23yt7P8GvCZQSzqkuyHuKhOdZ0bDmvrsUkN27NsX6yw63XkQq/oRWHJYdX/
bWzEXDS3I4+4eKuPB7PNz9WfnE/6yjRcZHmJqCn0+zXj/
N8qBOFDVv5uFDfqdq3nNxybAuh6rHTOYE9+Hfy+56ykUjzZJ37WCmQj5T6fRsbQ+
fc9jdMqHqqrJyqGCfD6Jw26ruYP4pOppapK7Cef8cndFZ5aUWzdNIfz/
U/6DgeffrCxjp0L3KzPI7LQfxxGSQR1/
nrfb9unKwXYYcBf/Vu2RjgUW9MoY5mmcQJ7oTN3/
iTCk6NgeeZqrYqvTPN8AX4ulbFXHNF4bZHi5I4X9P7AOJMmm1OPe/
8fOoyUfv/BHEpRqw3Qy3G+ogMf3qF/qd95zMyM1DSa6fTSI5YkJBRP8YKjTZFGMYK/
A0W6t7JpjGNyzN+4tdJBl17+5iaR3SuHaVBO0Sc22ZGzYjkfyLV4AWk1cJqL2Z1ScWZOhJmlIB6D/
IRfC6MhjrV/TyLF9Xut+ejyzl+NtTfmjnBBAKr0aG2p9QjPcBAsx7JwHyYPuyn8wfH0fPdGfLGWHNMzyX98eD2G6EntY91uMNe5aYqrjLbTmZXPcNqV6eYyCg4JOgWYH+bpZyFzmDY+aVdTHjBQcSHqGGIAhCcOnBTHyF3iEJMhz8pEXnfdczCH0oG6rwUA
kyeZnLLqH9x8XKs5DzG+oHPGJa92r8vMZrVyDJrqZ6HG91zyYqNw1Dvrsjw8yTk7YxWXs9NYWD7+zlz5AkpbGZFz0bzRCVYRy01HFZVIHIb/
OIrTpfRbJz+9HYOD9kyNZhHGOQvmNxXpQwG+s4iV3+pnIHh3HAMvDnhWjtUf0oyShUIdmw8lvs6ajHm/8rP7NLkQDsEMsUrlBmvQlWIPz95lJ8npwWmI1NiNv6JHudt75d
pyNqMNDaf5dS0DJoNmyxtIT4XUw6Q9rdGJPZ4MJ0rOjtYOpq8CQXrz4cDfMJWDbDqkZ/iAbX/
1mAgMAMfuEtxqpL6DGuQaUJXv/uZGA9eLYdXxGsOSp8Wnt+bIc5wFeN8BZpHLGK9y+x/
u5n5ydHYRxo4TQnx2W3EGRTwd7+bdDxo7Rg8m2D0flW2lcz8j2lm81KqBHyhFVq3/
9HsP7bxkR4RcXn1fbVh5tsd5GnXu2MgYfj1LZnNGxcg+Z0fAL0tqj5fcDhZg/
Ryfy6sdXM2W3vYaHEPuBLi7BjhkLCmIxeGU7k9ogDZV6sBxw+L6AUlhEqOsNTrYJUHlx2LTFJHp6I52tG0+Sa7FJfX6/
GfhjexOlyE4wtRy2Vl+ouCJf7V3UkpEz6F1j2TnDDW6/
5qIrpAmftUyxK/XJqxBUAdhXyUiAAk3qfGHExh/7pVQ3PYv6Q3Fv+R2umtSA7RPC5/
ldq7kApvOS3xqrwz2xbBJc73hI1U9L9Pux7R+3Pp/
3mZhf13sL+oa1lyGbNtvHnGfnPIyqtdZhEHMuvq+2iCDH/
EGwPAudwJk+fWC0hxFWnyHcN2FnrkSgmJEt2m7uMVDxtNTpJ0PMGBzVDF/
Oj5xpAdHSzt0UEZ4h2sNfQCkvTTzuvm+Qk5bwScxcs2PsTiyadzj/9Hy+s97kfcGLiBKzoeFqFT1VHFqNl9ctcvrQzLhqKJWQK
lBVsgD4n3enmL7yXCZn9M97WHszD5j6fL0jZsY85jmnmEE/0Mcz4y8JMtpeFFsphCNx/
6/tfA9hTfSAjnCmpL8OJq7XX2eU+mtkOYL5RlgPX8+eLBe8/
tVuB1KJ0n1fbaRhTuLC3BJ4tUWBp6Z4pHYTHvbqbG1y7i5CKbQOwzPe9fV/
3NoXkEjKfwCLcp8NBtE3YEs2K3S+ZzFmR2bbT1x9hHzI8ueWaAhIxy6XgZqE73bm1Vvf7g03oUEX/

a6ReqXZvg7qSxgCgoTL0+5jYmKUn5+v+Ph4RUdH1zmbzDAMmUwmVVZWNrlhBQUFkqSEhASH4wkJCfZzB
QUFtWa7tWrVSu3bt3eoSUtLq/Ue1ediYmr/
Y1daWqrS0lL7c0+FPMBvMfXcxmRq2Z8PQL1cDkRbt25V+/btJUnbtm1rtgb5g4ULF2r+/
Pm+bgbgPUw9BxDkXA5Et912mySpoqJCO3bs0GOPPaZOnTo1W8MSExMlSYWFhUpKSrIfLyws1HXXXevO
XPmjMPrKioqdP78efvrExMTVVhY6FBT/
by6pqY5c+Zo5syZ9udWq1UpKSlN+0CAP2PqOYAg1+hZZq1atdKSJUtUUVFHRHO2xS0tLU2JiorZs2WI/
ZrVatWfPHvtaR5mZmSoqKtK+ffvsNVu3blVVVVZUGDRpkr9m5c6n6fKy7//
n+2mTZvUo0ePOm+XSZLFYlFkZKTDA2jRmHoOIMi5Ne3+jjvu0I4d5r8zUtKSpSdna3s7GxJtoHU2dnZ
ysvLk8lk0jPPNf/epX+stf/qKDBw/qkUceUXJysn0mWq9evTRq1ChNmTJF//
jHP7Rr1y5Nnz5dDz30kJJ48ePl9ls1uTJk3X48GGtWbNGy5Ytc+gBAoJeS5p6HmzrKAHwCLf2Mh
s9erSef/55HTx4dzt99990uvc+nn36q22+/
3f68OqRMnDhRq1at0s9+9jdBYUPZ1atXa/
r06Ro2bJhCQkI0btw4vf766/bzUVFR+nLMJBrJU6dOVVFRkW699VZt2LDBYUPZ1atXa/
fsrNjZWc+fOdVirCIBss6ky0gN76nldU+fNYbawFwjtB+Azbq1DFBJSf8eSp2aZ+RPWIUJQMYzvZ52Zv
7tNFgg9Q8G+jhKAWhrz+9utHqKqiq3GgYgAAThi1HNX11GKjQ6McAfA69waQwQAfqUx6ygBQB3cDkQ7d
uzQ2LFj1a1bN3Xr1k133323PvroI0+2DYCnWgiELr2DQBcwzpKAJrIrUD0hz/
8QcOHD1d4eLiefvPpPf3002rrTpo2GDRumd955955x9NtBAD
///s/HT161GMN9AcMqgb8nGFIn+Q47wGyhNn2JmMMERA0mmEVKvv985//1//1NixY2sdv/
vuu3XixAl33hkA6BuBaKUlBSHFaSrbd68mS0uAPhG9TpM+LWcKYs7gwPUwW4HiGxHh+ZYWA6Ef/55LV
NPKzs7WzfffgMHRzjahi6aMEVdLjBdVxLgk2YMHUvwzlktkMZEeiTWoYA0+Eb4MBJsrACRk6Jg4CNOmY
Np0cr/Wzj7e8bZ/yKiAA0SXHEHhrfxqkhIyLAxk+Ry4lO7uBMNyeA/SDqaX0tHrf//
55LV+
+XK1bt9YzzzyjWbNmebSBANAgk8l8pX5eXSQHhGUGmZiYg6vaVP3/55LV+
Ap7mQEBoq5xQJYwWxiqbxzQmXPS0RMNv3evNCm+gfaCcArmm0vs61btyojI0NWq9XheGpqqoYNG6aHH
npIH330UeNbDcEBcj3dRX6neNLcD0u0Ya1Nf5oOimjj8C0CI0KhC99tprmjJlSp0pKyoqSo899phee
+riWLl3qscYBQLNryvgjAC1GowLRgQMHNHLkyHrPjxgxQvv27Wt2owDAqxoaf8RUfiAgNCoQFRYWqlWr
VvWeDw0N1dmzZ5vdKADwqobWUWoXbZvZtu+wtDPb9vVcsWdG63oLgCT7ZY1pGk8RxahAIAZTRwGAHqDGadI+ckvy/uKMVmvq1x2UUcNWUz2Ke
0V85Yy3wjeBcZhgOu2MEHxW1xx/NLGPBQQJiX38MV2+9H6mePyF0KSbGJ0GDnBcyUNoCmbDPfcVz5+0R+CTZDwi2ZkHQ+u5Re/s=
eKWll3qscYBQLNryvgjAC1GowLRgQMHNHLkyHrPjxgxQvv27Wt2owDAqxoaf8RUfiAgNCoQFRYWqlWr
CF7bmzWWJNGX8EoMVoVCAqLCxUWFj9/5Nq1aqVzp492+RGAUCjNWXqfFyMbWuPmj1FljC2/
ACCRKMWZuzYsaMOHTqkbt261ydHSUlJHmkYYALjM1anzsdH19
TExdjOV886M393m8ybPUNVVdLpM9K3pVIby1pV6CWWgiELr2ilSQKZZw0EiKqAN/o
fZbvfjii/
rBD37QYQYCQIMaM3sqwD2tcdLo4+8HFOG+AQfhPEaAOeAPSg8iiWnTi/AaRSX/8Mnh0r6W/vR+Bkgj0xeRxRY+M/vHkwHTkbHDPy10rlfbMvYAMSalh3QRrC5udV8YT
ez2mLmWNPP0wIcCBOLDGgR4o2TNRCiaPBqg22xW8AWIumoPMmS6ii4yAXzyi0+SBfKRHvYRl0XLRpZeKnBADPXxI6lLhpaQ9cvPBLeaOxYlBQd9PSA/Gwju1RzjcNGTTKPvI5aWKm0/RHTw
uATBXJNpdJvZTXwQjF0LhahIEDrR2mYOL3sJE8aLZvlqdaYvU1iS8qAIjHhbGgVhOwvhzZZ3sz4oEKR8vw4PrYbkW8jsRx4M1QSAyeGd8j8s5bsIPXKz4=
aBQBoVgQioBpbbwBA0CIQAdXYegMAghaBCKjG1hsAELQIREA1tt4AgKDFLDOgWvXWG856gFzdeqOp0/

YBAF5FIAKqVW+9cSS3/
hpXtt5g2j7cRZAGfIZABFyteuuNmoHGEmYLQw0Fmupp+zVVT9tn+w7UhyAN+BSBCKjJ3a03XJ22HxvN/
/rhiCAN+ByBCKiLO1tvNGbaPtt6oBpBGvALzDIDPIVp+3AH618BfoFABHgK0/bhDoI04BcIRICnVE/
bd8bVafsIHgRpwC8QiABPqZ6274wr0/
YRXAjSgF8gEAGeFBcjdUqo+1ynBGYKoTaCNOAXCESAJ529IJ0qrPvcqULbeaCm6vWvavYUWcKYcg94Cd
PuAU9h+jSawt31rwB4BIEI8BTWIUJTubP+FQCP4JYZ4ClMnwaAgEUgAjyF6dMAELAIRICnMH0aAAIWgQ
jwFKZPA0DAIhABnsT0aQAISMwyAzyN6dMAEHAIREBzqKyUThVI35ZJbcxSu3CpVYD8dauqkk6fkb4tld
pYpOR4KYTOZAAtm1//Kzdv3jyZTCaHR8+ePe3nr1y5omnTpqlDhw5q166dxo0bp8JCx1WC8/
LyNGbMGIWHhys+Pl6zZs1SRUWFtz8Kgsm+I9KubOmcVbp8xfZ1V7btuL/L/
Vr6aL+Ue0o6fdb29aP9tuMA0IL5dSCSpN69eys/P9/++Pjjj+3nZsyYoQ8++EDvvvuuduzYodOnT+v++
+
+3n6+srNSYMWNUVlam3bt36+2339aqVas0d+5cX3wUBIN9R6SSy3WfK7ns36Eo92vn244QigC0YH7fh9
+qVSslJibWOl5cXKzf/
va3euedd3THHXdIkt566y316tVLn3zyiW666SZt3LhRR44c0ebNm5WQkKDrrrtOL730kmbPnq158+bJb
DZ7+
+PAVYYReGNwKirqD0PVSi7b6vzt9llVVf1hqNqpQimtI7fPALRIfv8v25dffqnk5GR17dpVEyZMUF6eb
a+offv2qby8XMOHD7fX9uzZU507d1ZWVpYkKSsrS3369FFCwve7j48cOVJWq1WHDx+u93uWlpbKarU6P
OBFZy9In+RIB76Qjp6wff0kx/
83Rv38n56t86bTZzxbBwABxq8D0aBBg7Rq1Spt2LBBK1as0IkTJzR48GBdvHhRBQUFMpvNio6OdnhNQk
KCCgoKJEkFBQUOYaj6fPW5+ixcuFBRUVH2R0pKimc/GOp39oJ0JLf29hZl5bbj/
hyKvi3zbJ03fVvq2ToACDB+1m/vaPTo0fY/9+3bV4MGDVJqaqr+/Oc/
q02bNs32fefMmaOZM2fan1utVkKRNxiG9MVJ5zVfnPTf3eLbmG2DqF2p8zdtLJ6tA4AA49c9RDVFR0fr
mmuu0fHjx5WYmKiysjIVFRU51BQWFtrHHCUmJtaadVb9vK5xSdUsFosiIyMdHvCCootSRaXzmopKW50/
6tnVs3XelBzv2ToACDABFYhKSkqUm5urpKQk9e/fX4XelBdyqyZYv9/
LFjx5SXl6fMzExJUmZmpg4ePKgzZ74f97Bp0yZFRkYqIyPD6+1HA4pcHKvlap23tWplW2/
IGX9djygkROqU4LymUwIDqgG0WH79r9tzzz2nHTt26OTJk9q6e7fuu+8+hYaG6uG6uGHH1ZUVJQmT56smTN
natu2bdq3b58mTZkzMxM3XTTTZKkESNGKCMjQz/+8Y9Y914MAB/f3vf9cLL7ygadOmyAMAB/
vGIZn63yhc1LTzvtSekr9oahTgu08ALRQfvhf1e+dOnVKDz/
8sM6dO6e4uDjdeuut+uSTTTxRUI0eO1P/8z//
YXx8aGqr169frySefVrMxR9HV+u8zTCk43nOa3Lz/
HcMlGQLPWkdWakaQNAxGYY//
3fbP1itVkVFRam4uJjxRM2p8Jz0+YmG63qmSQkdmr89jVVktS0R0JB+10jR/BwBQHNrzO9v/tsH/
2EJa7imMXXeVnOpgKbWAQC8hkAE/
xHZzrN13mZ2Mai5WgcA8BoCEfxHsYvT6V2t87aoiIbDjuW7bUgAAH6FQAT/
4er6Qv66DpHJJHXr7LwmvbP/DqgGgCBGIAI8KS5Gykiv3VNkCbMdj4vxTbsAAE756fxlBKXoSCmv/
j3mHOr8WVyMbWp98UXbAGrzd7fJ6BkCAL9FIIJ7DMPzv/CjI6TQEKmyqv6a0BBbnb8zmfw/
uAEA7AhEaLyzF2wLEF49fdwcZhs/
09RbQs7CkCvnAQBwA2OI0DhnL0hHcmuvpVNWbjt+9oL7732+2LN1AAC4iEAE17m6NYW7i5+fcmH8UGPq
AABwEYEIrqseM+RMabn76wSVV3i2DgAAFxGI4Lrm3pqi0sWeJVfrAABwEYEIrmv2rSlcDToEIgCAZxGI
4Lrm3pqi3MWeJVfrAABwEYEIrmvurSnoIAIA+AiBCI3TnFtTVLmYdFytAwDARSzMiMZjawoAQAtDDxEA
AAh69BCh8Zpr6w6TybVFHemJAgB4GD1EaJzm3LrD1ZxDHgIAeBiBCK4zDOmLk85rvjjp/
tYdrUI9W+eMYUhFVunMOdtXd9sMAGgRuGUG1xVdlCoqnddUVNrqYiIb//
6u3gpr6i2z5rrlBwAIWPQQtWSe7gW54OIu867W1eSNvcya85YfACBg0UPUUjVHL0hpmWframrudYgMw3
ZNnMnNsy0pwMBtAAgq9BC1RM3VC2KxeLbO26rXTXKmtNxWBwAIKgSilsbVXhB3bp9FtfVsnbeVurgHmq
t1AIAWg0DU0jRnL8i5Is/
W1RTq4o+jq3U1sXksAKAeBKKWpqEw1Ni6q10u9WxdTclxnq2rKczFIXOu1gEAWgwCUUtTc9PVptZdLcT
FHxdX62qKdnGqvqt1NVnMnq0DALQYBKKWJrKdZ+uu1twrSV/
+1rN1NUVFNBwELd9tVAsACCoEopbGWuLZuqtdcjGIuFpX04l/
ebauJpPJtuyAM+mdmXIPAEGIQNTSNOcYouZeSbq51yGSbGswZaTX7imyhNmOs1I1AAQlRo+2NM05hiiy
rXTFhUUXI92cdm+S5ErWaWoHTlyMbfHF6hl55u9uk9EzBABBi0DU0gTyGKKodlKRC7fyotxoe00mk/
uDswEALQ63zFqa5hxDVOLi2CBX62q6fMWzdQAAuIhA1NI05xiiygZ2um9sXU3e2NwVAIA6+HUgWrhwoQ
YOHKiIiAjFx8fr3nvv3bNbFjxxxqhg4dKpPJ5PB44oknHKy8vI0ZswZHKz4+XrNmzZLJZNKrr77qcL
bBNOYaolYt3WF2tq8nVsdJNGFMNAEBd/
HoM0Y4dOzRt2jQRjQNHDhQFRUV+vnPf64+4G7U6ZM0YIFC+zUmDFjlJiYq
N27dys/
P1+PPPKIwsLC9Nhhx8sC9PLLL3v183hUfbvZp6dIrUKlCie9NK1C3VtrJzTUs3U1WcJc20fM4kaYAwDACb8ORBs
bHB4vmrVKsXHx2vfvn0aMmSIHx2vfvn0aMmSI/
Xh4eLgSExPrfI+NGzfqvvvuU1hYmB577HhZfPLLL3v183hUfbvZp6dIrUKlCie9NK1C3VtrJzTUs3U1WcJc20fM4kaYAwDACb8ORBs
l5fDWnlYtBxta6mrh2loyddqwMAwIP8+pZZTcXFxZKk9u3bOxxxfvXq2mfDm6fPmy/
VxWVpb69OmjjhIE+7GRI0fqRVb69OmjjhIE7xcM5xiiygZ2um9sXU3e2NwVAIA6+HUgWrhwoQ
MXVpb69OmjjhIE7xcM5xiiygZ2um9sXU3e2112rOnDm6fPmy/
VxWVpb69OmjjhIE7xcM5xiiygZ2um9sXU3e2112rOnDm6fPmy/
LW5x/i4MqW/
MXUAALjIr3uIrlZVVaVnnnlGt9xyi6699r78fHjxys1NVXJycnGJycnGR7NmzdezYMa1du1aBgYF67LHjZfPJL73
JNmfFxQU1Pm9Fi5cqPnz5cqTJ3FR7tfSqcIax05JsTHuDYiuyZ1QccXFTVtdraup2MWZb67WAQDgooAJ
RNOmTdOf08ccfOCxyfOxyfOnWq/

c99+vRRUlKShg0bptzcXKWnp7v1vebMmaOZM2fan1utVqWkpLjXcHfUFYaqfXPBM9/
jYomUGNu41zT35q7eWKkaAIA6BMQts+nTp2v9+vXatm2bOnXq5LR20KBBkqTjx49LkhITE1VY6Bguqp/
XN+7IYrEoMjLS4eE1VVX1hyFPcidTNPdu8RHhDdc0pg4AABf5dSAyDEPTp0/Xe+
+9p61btyotLa3B12RnZ0uSkpKSJEmZmZk6ePCgzpw5Y6/ZtGmTIiMjlZGR0SztbpJ/eSEMSVJ468a/
JszF2V2u1tUU7eLMN1frAABwkV/fMps2bZreeecdvf/+
+4qIiLCP+YmKilKbNm2Um5urd955Zc6dOignJwczZgxQ0OGDFHfvn0lSSNGjFBGRoZ+/OMfa/
HixSooKNALL7ygadOmyWKx+PLj1c1b42OS4xv/
GqOBwdqNraulufcGAQCgbn7dQ7RixQoVFxdr6NChSkpKj/
WrFkjSTKbzdq8ebNGjBihnj176ceP0wQcf2N8jNDRU69evevV2evV2hoqDIzM/WjH/
1IjzzyiMO6RX7F1TV8ItvVv2N7p4S6X1OtU4J743xc3UPM3b3GXJ355s4MOQAAnPDrHiLDcD7QJSUlRT
t27GjwfVJTU/W3v/3NU81qXgkdpDPnG65LTZJiIuvesT0uRiq6KVcrv26duG2xRvd0TFB+ue/
XKsDACCA+HUPUVCKiWx4cXQEftd9Y7t8R1sX03f3UrK/bruMCTZjud+7V7bQkar/
dJYgwRAMBnCET+xmSSkuKc1yTFf1WNEaIMJEqIsI1Df08iv6fkL732SbKGuoVWuW4Xa6gAA8CACkb
8xjIZvmZ09b6uri6Oa67pUn8xTCbpmi7Oa67pUn8YBADATQQif1M9JsiZ0vL6BxWmpwttGn
mq0P3bcdXiYmwDw8NqDG8zt7Idj4tp2vsDAFAHvx5UHZRc3ZSkpKSqy6Lc732SbKGuoVWuW4Xa6gAA8CACkb
TdA8YBAGgG9BD5m5pT6RtbF+HilHdX6652+kzDNY2pc6a+AeMAADQDApG/
iYpoOBRZvusxqUuMiwOOXa272rcubtrqah0AAH6CQORvTCapW2fnemd6+8xiY5wcaaWG1PX27i4sre
dQAA+AkCkT+qHlhc30rUzgYWm0wN72KfGOveLShXt/twZ1sQAAB8iEHV/
srdgcWuTQA6ubsjAjAAAAAA+wm+ulqap0/
Yb0pwLMwIA4CP0EPmTYiXEdtjvn5k5kddq+K9TJbFPrT5xb5bDuY7HdJqNnCA/
AQoAhEvnL2gnQ8zzGYmMO+H1B9JLf2a8orbMedjSNq6rR9V4WESJ0Sm/
Ye....

xB4Oa167a6tWrFRsbq2uvvVZz5szR5cuX7ee4drbenj/96U+6dOmSMjMz/
fJnjs1d0WSDBg3SqlWr1KNHD+Xn52v+/
PkaPHiwDh06pIKCApnNZkVHRzu8JiEhQQUFBb5psJ+qvh5X/
+Wvfl59rqCgQPHx8Q7nW7Vqpfbt2wf99Rw1apTuv/9+paWlKTc3Vz//
+c81evRoZWVlKTQ0lGv3naqqKj3zzDO65ZZbdO2110qSS39PCwoK6vzZrD4XDOq6dpI0fvx4paamKjk5
WTk5OZo9e7aOHTumtWvXSgrua3fw4EFlZmbqypUrateund577z1lZGCQoOzvb737mCERostGjR9v/
3LdvXw0aNEipqan685//rDZt2viZQgmDz30kP3Pffr0Ud+
+fZWenq7t27dr2LBhPmyZf5k2bZoOHTrkMM4Prqnv2l09Dq1Pnz5KSkrSsGHDlJubq/
T0dG8306/06NFD2dnZKi4u1v7/f/
9PEydO1I4dO3zdrDpxyxwywweFx0drWuuuubUbHjx9XYmKiysrKVFFRU5FBTWFioxMRE3zTQT1Vfj5qzLK6+Vo
mJiTpz5ozD+YqKCp0/f57rWm6a6ZdLIfd+XvaWJiYp0/
m9XnWrr6rl1dBg0aJEkOP3vBeu3MZrO06deum/
v37a+HCherXr5+WLVvmlz9XJiXJjc5WUlkT+/fsrLCxMW7ZssS8/
duyY8vNpd+p2Q1LylJmZ6cNW84zDHYHvwbaWu6p/fDEGmT99kVHRzu8JiEhQQUFBb98++e83WrVtVVl/
0cYNqdOndK5c+eUlJWEYmj59ut577z1t3bpVaSfvbbz1t3bpVaWlpDudd+XuamZmpgqLTZs2KTIyUhkZG
d75ID7Q0LWrS3Z2tiQ5/OwF47Wrra6uTqWlpXr4+eFddmtUVHj5nAOOWDDDeEdnJOhaK3MOmaDZ+bcPDb
PPjM8++8yQZCxdutT47LPPjK+++sowDMN47LPPjK+++sowDMNYtGiRER0dbbz//
vtGTk6OcczjxIkTxubnBm40bbrjB6N69u3Z48e4+OPPza6d+9uPPza6d+9uPPzww776SF7j7Npdvhj7Npdv
HjReO655
4ysrCzjxIkTxubnNm40bbrjB6N69u3HlyhX7ewTrtXvyySeNqKgoY/
v27UZ+fr79cfnyZXtNQ39PKyoqjGuvvdYWMWKEkZ2dbWzYsMGIi4szYsMGIi4sz5syZ44uP5DUNXbvjx48bCxYsMD
799FPjxIkTxvvvv2907drVGDJkiP09gvXaPf/888aOHTuMEydOk5Ocbzzz9vmEwmY+PGjYZh+N/
PHIEITfbDH/7QSEpKMsxms9GxY9+W+//db4yU9+YsTExBjh4eHGfffdZ+Tn5/
uwxb6zbds2Q1Ktx8SJEw3DsE29/+Uvf2kkJCQYFovFGDZsmHHs2DGH9zh37zpx8MMPG+3atTMiIyONSZ
MmGRcvXvTBp/
EuZ9fu8uXLxogRI4y4uDgjLCzMSE1NNaZMmeIwXdcvfa1XXdJBlvvfWWvcaVv6cnT540Ro8ebbRp08
aIjY01nn32WaO8vNzLn8a7Grp2eXl5xpAhQ4z27dsbFovFF6NatmzFr1iyjuLjY4X2C8do99thj
E2m424uDhj2LBh9jBkGP73M2cyDMPwfL8TAABA4GAMEQAACHoEIgAAEPQIRAAAIOgRiAAAQNAjEAEAgK
BHIAIAAEGPQAQAAIIegQgAXLRq1apau3MDaBkIRAD8zqOPPiqTySSyaSwsDAlJCTozjvv1MqVK1VVVe
WVNnTp0kWvvfaaw7Ef/vCQ/UFPVFFR4dZ7Gobh9mslqQU2bNoqPj3f79QD8F4E4EIgF8aNWqU8vPzdfLkSX344Ye6/fbb9dOf/lQ/
+MEPVFFR4dZ7Gobh9mslqU2bNoqPj3f79QD8F4EIgF+yWCxKTExUx44ddcMN+jnP/+53n//
fX344YdatWqVTp48KZPJZN9ZXJKKiopkMpm04Ycfqn///
rJYLPr444+Vm5ure+65RwmJiWrXrp1gBxk1f4ul/p/hfttKfhuhx/+ykwY/1nsvsqKqFJ2fuFHWQswpnA6gx/2ZQVKb
T169NDvf/97h/
Mmk0lvvvvm7rvvPoWWHh6t79+76y1/+0iXDID7CEQAAsYdd9yhfv36ae3atXrsscdcft0vfvELLVmyRF27dlVMTIy++uorff/
73////S2Mmk0lvvvvPoWHh6t79+76y1/+0iXDID7CEQAAsYdd9yhfv36ae3atXrsscdcft0vfvELLVmyRF27dlVMTIy++uorff/
qmeffVaHDh3S448/rkmTJmnbtm0Odz9eDDz6onJwc3XXXXVq9erXdlJdZowYLH
v3kUA0CwIRAACSs+ePXXy5MlGvWbBggW688
PHHde2116p79+76y1/+0iXDID7CEQAAsYdd9yhfv36ae3atXrsscdcft0vfvELLVmyRF27dlVMTIy++uorff/
RNddco5k5zZ+r+++/XK6+OdcyK9oRPxCsQQQiYv2vKn/NCMAC1tv/
qeeePX80mdM5sA4gRPEIHB5AzRTENwGhcVu0p66F1O8ssYV2+6SXWSWYrWRWMzEfWZ2OQ02vuI+nDzmnt+Pf7nhvuKffG+nbfu/
c+vwAmgeBCEBAMQzDVfGvLVQMGDtGBDHU2LCUxLzMDMOHcQCAHUpmkEIVx9bXFqN0rLThMRH/nlrCkX9o+s2lYz/
fXM6vYmWNLuT8VJpTBxhKrXdt5HNoIGSGUpeHgMzcHcvj0E9mdyv9DQ2NWLtP0T3h1W9vwAdwM+wPKWhx6OwNQOg7bExnGhXbO/vEyjS
1JX1YsnZgHfgeRrvWi0KExHOzoj+ROm7c6Hhl2PTgBfFwkpzz9rsOp4u4u7e9H/
fXMM8/oV7/6lUvvvvXTUsXExOjmm2Sy/c6aHnLe5s6hOQsOXIJ+H9M3Xo/
W2LFjNXLkSN1www0ONQsWLNDJkyeVnp5u742q6dvrf//
1fvfXWWxo6dGiTpsPY4AI2UwLRPOKtz/l4wFDhrEG2LrGcAYzY29Oj5/
D0CEQAACDoEYgAAEDQIxABAICgRyACAABBj0AEAACCHoEIgAAEPQIRAAAIegQiAAAQ9AhEAAAg6BGIAABA0AAEACCHoEIgAAEvf8PBMlAbZZ+EzUAAAAASUVORK5CYI
I=\n"

},
"metadata": {}
},
{
"output_type": "stream",
"name": "stdout",
"text": [
"Coeficiente de correlaciÃ³n: 0.9227/n\n",
"Coeficiente de determinaciÃ³n: 0.8514/n\n",
"Intervalo de confianza de 98.0% para la pendiente:\n",

"-0.8839\n",
"   df   sum_sq   mean_sq   F
PR(>F)\n",
"X   1.0 9.847530e+06 9.847530e+06 928.219489 5.795220e-
69\n",
"Residual 162.0 1.718667e+06 1.060905e+04   NaN
NaN\n"
]
},
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 640x480 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAkcAAAHHCAYAAAC1G/yyAAAAOnRFWHRTb2Z0d2FyZQBNYXRwbG90bGl
iIHZlcnNpb24zLjEwLjAsIGh0dHBzOi8vbWF0cGxvdGxpYi5vcmcvlHJYcgAAAAlwSFlzAAAPYQAAD2E
BqD+naQAAX51JREFUeJzt3XlclOX6P/
DPMDIssonsgoK4oqKlqZQLprmVaUS5VWqZp8zc0rLTKZfOSdMyLTOr80s7lZ1caD3lNzNzQVNxF3FcQRB
AF2WWbuX9/TIzzMCwzwzAL83m/
Xrx07rnmmXsehnmueZ77vm6ZEEKAiIiIiAAADpbuABEREZE1YXJEREREpIXJEREREEWJkdEREWpgc
EREREWlhckRERESkhckRERErkRYmR0RERamBwRERERaWFyRERYsmQJZDKZXrEymQxlLlixp0v5ER0cj
Ojq6SZ+DGic0NBRTp061dDeImgSTIyIrsmnTJshkMs1PixYt0t61dDeImgSTIyIrsmnTJshkMs1PixYt0tOZ37eDg
AA8PD3Tu3Bl7BlPP/00du7caenuYf/+fi1KlTcHZ2Nvnz/eMf
Mt3RUis2ph6Q4Qka5ly5YhLCwMZWVlOHDgADZt2oS9e/fi1KlTcHZ2Nvnz/eMf/
8CiRYtMvl1qWHBwMJYvXw4AKCkpwaVLLlxAXF4evv/4aTz75JL7++ms4OjpapapG/79+/
H0qVLMXXqVHh5eUnuO3/+PBwc/P2aamicmR0RWaNSoUejjTpw8AYPr06fLn376CU8+
+aTJn69FixZo0YIfB5bg6emJp556StK4YsUKYzGrvXr0doaCjeffddkzxXSUkjJWrZsaZJtOk5mwWQ7
RNaIaT+RDRg4cCAA4PLly5L2c+fO4dNcfOIyrdkc1E4jUp31ZaFGq8oPCjS4fnYVTDa+Oma1nYbQqn/fxb23/
SWIqKyuxdOlSdOz6EgEc7sOjmduOhEDZZjmlbnH0n7mSPVmsOj7rtWzqZKFgRR4WIddHuDHvwK8vLwaLMLnT
5T+KJUC7kXsdJHb81QPqg+uqrKC8vl8Tt3bkt3x5AhQ3TaLsBMpKfQECgrJ1kFkpJO5uvU++xFnmHt5xJn
otPEP+OZUFfbC2gmv8G7W3Tk3fZoXAvStkMUq+tSQrVEGTPcBJTNqLYDBNaLAgqJrVsEk8ocnqXQXAsBSy1lLcd
qrWOEwFT9a8YQgyzAGF+HvNOxnhA5XSFBKbW5b2n7j4WyMTdKYwFJzxd+ISKaoVcMk7M79H2ZomnG+MXwvM0qV0/
Wvv0W3Ov7N/Bbtr8Lm6ZmfLCoZ5Qq+R/K5y9lGb7yN1x7MSDTpxyD9+IQE5gaZ7rZTjIb+ZUuX2jjqjOuD9
VdGEZ/J08Yy4ZSjlc1+FzgYM3A9vNEYCqsSwDgJbxlYQ7BIgs/MFSpLOBPb5DwZJTqSjT/
6eEgQfGodp3kpsgKx6hzOL3EaGI5x2IrC/UgTr3hiHKkVeb6YYDuFSb6s6jr0KFP1pkXGo+jIkJDRqvnwyj
ey1gW1ydDAAhLD73eGgrOcpqdyHz2RuVVaqSJSvjSAgzHbHpZbSVpAGXxVF6nqwVZfsm9j1T4Fqnkc/
MOfiP9W6eTvXeZ2+WNLZT0y2TcU4MNa5ze+UBanO18Ax3pInnidMLGfK3qu8B7GiIbpT20/
mZaV4EJmvlOqxfFsbcOW8e/LYtDvUAwoa7EO0z6jQr1mrkrmvN35BWu6OjHzVcx7zmlYb1V5XgqSOxK1yXpz
dcF9qgYYrL2zf4NbaCvpOsx/0OR7xUdajGukKcO1dzeDWB+uYlR0E29OxiE3EuWQ2cBXyEZpCNcbC9kSvabA
JMDnLowkcEeBf1BX7/CT8lvTTUDrTLYCypHS4D/Oz0Z1l2mGrQ/VrUKjbFz7nM4z5Cd0X3L69SXO3i3y9tBMr
2tKyaOY0cbJAaaNV3qtrYbbMxIgvwK7Vox8EdLA1mAA1g9uQHZ0dZBlbCoCT+nDyaEAQ5SatxjNhhe51m12
/1Tvwjdsb7TKgc9omr/cGtzgvh1A4VoBL1b13sXk7NGyRJ+6GxXsP0pI8KT/QWXkUlYRm0QDMrUjXmmgKzfnM+
9UAq/+IjmsPuhQMXlxyU/TVQlqzEQmh/hGCHo+3dFzHqthhVAsPd2ulmQz+n+0vsl7TQlPMe/uA5wL7LRqzc
Vk+jtRN4GiShmaPbm3u+4BMsEHnhN1wy3+bAcY3M/+jtUyq85KYRzrtV9h4CsAW4NjJ8Ts11YI6gX0J/
Ou7HqKSZ3cxLr/M2PWnYq5C/e4yX5YkmAVxsGPMDE6Qs0dbgPgdmTUNkqpI4YrT1/LEJ+aqh3SJ2QNY7u47I+fD
gqjlsOLkOSL0Twx5D3UhPotNzSnuIVqn9vLmzrU4i6fOIu/Nq2Ojj+ybLT/x06rwKsF7kWtYgp4kE3k6P/
ExA+5dRwCX6HJWy7JOGGPTnOTGRJLHAaB9LZ8ihAPbtf9Z93hQeYNbLt6cRHDSjbrf7nt3rqkoN+PV/ynwRvjz
Wrq4b6SkRm+ySoEPAX9rxS1Xbm/p0aCOfi+1O9Gv0LYSf1mH0DrYTC3KOzNTK+cjcmXFAF5+Qz9vqqx3OmKa
Uj/2Uijz/oQ6Ka0ej9hMO/1xbvk0XMrb1LGgwj9zyrqT0D2Bk2Ji8O8u/RCYCfrknoMSsqY4qOc0Ln/OB7H/h
nJMUH3Sy8yMqv9o6S00DNLM4e0JMeotV7V5/V5btcq6jmHFl1/9t/49DsfxuNV6J0h7nv38r+rCF7YbYhi80H
7MQvzR4OMRGPHCjORCfHW+1+7bxMs1GADXi+/sRF+1xl3V9E4jDkDPHOvBhqN9oAHgK7cM/ebG8Cf3ryjXr
SvlxZXG6PJENFLKiPqMMOQPn1KqOQDIsgBd1Q7wgCLBnt/JjvSbrAuUy4e5rwtOFbQk8EHCm+f4GH8mg/sB5/
0Jq8gTEwadmvn1RoS33Ilpy6+iCVK32iYlO/dfeQmk7O5sjMNaVWTzgn11VIk1PfZaWJu0Jmd2NQjUTlTzmOt
v0kYrf1+ZebJCswBYvu1lhzMyb8TDqR/ptJk/cNWczqpmi5drIp1Os/0hwEU0xgvI+Btc4+NtbszmO8i5tFwSl
Vx/rBqJcN+K+B6duz4v3ok/YaYMowRf+BLZRA+hrvxA3GYGS0ThfR4N5ltcAZRpL6ChdRIjLAbgq1pCRvlLv/
8h6R7wPltpnp4V26sVmAL8HDEm4UI/BQ7h5k6FKrhWVdOcxP7WCIy48CL+WhGgTG5ToY48Vw/8Zkqpbe/
xWEPPLdBkwMdyh+ilvHBMqwOhbrBUyvkdrdYsFU5j9hWSKwf6fvRzrvHbazT1+WT4xgmOXfz9Dj4yFt/
j2kGI7NPvvs+RGpnRtZYYS5FmQmTUUWAXlz2BGhm3xYunKPBZVNbL98e0cj0GmLg8RX0VVWfnOuZL4i3dO5
mRlBMqoqTXnjPJZiPjpG6bHwsDuQAy3uvlb0bdAV6f6iQvqBIehmxRn32Bpm9ssqIe2TXPvjxTaI1cCv9Rg
Y5Bfwv+NYx3QwJfqhkSXp/u+YKtTcf3O3mEt5ntzSySunGl/HjcFKC8sPhZ6pr/Jt8MMTbR6SS7hpfqvxlS7g
vMqA+VK+5c1xvtEgMk0jPbbuhHrbD/L/16QD6wq2sIK1m7/9aaWXnbD+xjiwpSUo7I3ZV0wDZwkIXGAh5Jw4
fdrZ3ZwUvHVotnB+c6adWxuuhbNY2yHaqQGjtjfqzcoOz34X/5uRWOgWvy7F6vvvBUmomXKqNwoiWe7TD2x/hpkY4
PJPjrV/kxaMiBbCjK1v+S7TAuwUtJsmqcqRzR3yK1Q6Jtu6PFE+5pUugTySGb6tSVAKwFHjrQ/
KOtZ0mQbmUFzr4SKP73/GYZv1ZhJhwb+l0zyzAbjg4OWtdPQgXHFW3z1mp8X7ZEPPDJsHr21i91nmSqo2+SaIo
yyW6Z5wTeSfYqzfWWCK+SRFKZEVhtlPItq6xuaxwIZOUw4tg/mXw2Fct+UuYdNA44VXi99Pbm31xhGtY6ZFJaYR
1RlNVw/ffrs5AmMLYzRPEwLGoIIEwyjt3Exe2eRyvfhAcyZWLcw7q06Lc8+FNbs91LVWcJ9kpD/+WfnumyHdGTd
d9fWVnyk3uiV6yiNz2O0Z9/r7Oc2qLZxbEtoJ14AoY3ZZgUvwSL0SbMXdpWv+NhH+4Cz4NAvfY7Mbly/
WeRjpZpN4v6Rnr+LK1PfixuUjRhoXwdtYwTfJZBgAXzBbUr+u+r0G35U13PGTdHHeUm1FNyE+vYKk/4+A/+
IHt7LlakjpNWw/ClO3hCRlxvQetewGIKBjNWYN+73t2DPSwKE2nilZk+IlMRFmN+J6XOHJC+8Yjq+ZSd/
0UbTV5rCk+DAMrz96JW96Up1wrTM7AUYgJPHFjS1BIhn1nk6GTQ+xZMVnbmdtr7VzBhyVJThtK+MQsHwRV
RdA6dYkQUgOYjkuGXcEXrqgUR88Ps0mlxp6Cv5y4y7bkt4EBBpyhbQNpcB1etm5DFSB0UMw39B+mBtGWtr28
Mt/PVfvAYsNNu0zMhBZI8dLx7WbKQbwndU7yrYw5bimcWg1x+u7CU5s4kz2sTvGpwh0Gjo6BZ3i/
Z/1y4tTNdq8QkfqFQYjKNX5obK8+XvGsxN46Nwt+dnbN/IxdvpOeTLBK1eIUT9ugAFb+3dN2WeKC4FiPeRPRrF
LmALvFepETnuXxfsKHHk1iq+Wfxg7yU2dKB2IdBYRWZZzSZEK7Pw0H1y0d/HIpRg7+b9qf1tDrXC/
bbxx7hw9CeS4xYm/l2pJVHhpvn26UmxuUWlH+xBNX44U/Yx2zSmn8bzXYKmnNTYJsOuU8tiBiEF1P+Oq+2zF
GMlmNsvRewRNgkpJcjEp/dL6hprNj9bXzUdU0Ybw3bsbEcTYPhsOtGMt2Zvuzsdbm2CLqT9fEU7qLAOLJBD5V
+SUOJWXFP02TNeMoZJXdB2mWaVXuk3NzLEjLwlBJFTiLujX2LnIPlsjUbthLxT4VT/oQFOm1/
QwJI3Hk1dDCUSV/QfEhk+C5+c1sV3Ldcr9eWVvvnOGRmTeCY+ksQOAqM+lz6YmK6Hu+ceZuocdffg9p09U+/
0f2JT6jmL3SyoM3Uw2m6y/t7KDF+MfsMefRR8KfvnwpX5rNSuwYp87mx2FQJf5TyGiSK9YDj3gd6EFjNnV55dG
cprL94ey+1aUi/Lhw+6uGhjeQ9yAHvsbeljRuzcp3rxAbCBmGPVJbh0EZ6X7jVvn4azlkVUtAsZvAGibRxw5wn
8YSxz01LWhfMVwXr/kOWwHiX5BBpsN2cg8HZhnxeQlNxKqsVbe9B8aGf9pSC33CvyoO/0OzxAyCNM2rkOKcY6hnn
lQDTlrMC0BrSBfsh+Sp+k5w+33KbwnsSoUOwNQXQ83eBR7mI/
Av6/+pOCXsZzSTWH6Nr4/wrVeBTVmvy/

PjxmpiZM2eiU6dOGDVqFIKDg/Hss89ix44dRu+PhhQXFwMA3N3djd5GWFiYTlteXh7mzJkDf39/
uLi4wNfXVxNX3++hPlevXq31/dG1a1fN/
dratm0ruV19yfj27dtGPT+RKXHMEZEV6tu3r2a22rhx4zBgwABMmjQJ58+fh5ubm+Zgv2DBAowYMaLWb
XTo0AEAMGjQIFy+fBk//vgjfv/9d/z73//
GBx98gA0bNmD69OmN7mtdxSOVSqXk9uXLlzF06FB06dIFq1evRkhICBQKBX799Vd88MEH9SYwKpUKPXr
0wOrVq2u9PyQkBID6jMiePXsQHx+P//3vf9ixYwe+++47PPjgg/
j9998hl8vrfI7x48fj9ddfx9atWzF37lxs2bIFnp6eGDlypCbGz88PycnJ+L//+z/
89ttv+O2337Bx40Y888wz+PLLL+vctrFOnToF4O7vUt99rU37LFG1J598Evv378fChQvRq1cvzXtq5Mi
RBp1xa4y6fhfir3FRRJbE5IjIysnlcixfvhxDhgzBunXrsGjRIrRv3x4A4OjoiGHDhjW4DW9vb0ybNg3
Tpk1DcXExBg0ahCVLltSZHLVr1w4qlQqXL1+WnA04f/68TmyrVq1qLRJY80zBzz//
jPLycvz000+SswbVl8TqEx4ejhMnTmDo0KENVvJ2cHDA0KFDMXToUKxevRrvvPMO3njjDcTHx9e7r8LC
wtC3b1989913mDVrFuLi4jBu3DidKesKhQJjxozBmDFjoFKpMHPmTHz66ad48803NUmMKSiVSmzevBmu
rq6aS1XVZ1dq7u+a+7o+t2/fxq5du7B06VK89dZbmvbaLrvqWzUdUL9nant/nDt3TnM/
ka3gZTUiGxAdHY2+fftizZo1KCsrg5+fH6Kjo/Hpp58iJytLJ/7mzZua/
2tPzwbU40A6dOigMwVe26hRowAAH374oaR9zZo1OrHh4eEoKCiQXKrLysrC999/
L4mrPlOgfWagoKAAGzdurLMf1Z588klkZmbi888/17nvzp07KCkpAaC+XFRTr169AKDe11tt/
PjxOHDgAL744gvcunVLckN0N2XDg4OiIyMlGy/srIS586dq/
X3oi+lUonZs2fj7NmzmD17Njw8PAAAHh4e8PHxwZ49eyTx69ev13vbtf0egNp/
t9U1kfSpkD169GgcOnQISUlJmraSkhJ89tlnCA0NRUREhN59JLI0njkishELFy7EE088gU2bNuGFF17A
xx9/jAEDBqBHjx54/
vnn0b59e9y4cQNJSUm4du0aTpw4AQCIiIhAdHQ0evfuDW9vbxw5cgTbtm3DrFmz6nyuXr16YeLEi/
fj0KCgpw//
33Y9euXbXWoZkwYQJee+01PPbYY5g9ezZKS0vxySefoFOnTpLBvcOHD9ecdfnb3/6G4uJifP755/
Dz82swkXj66aexcsWvPDCC4iPj8cDDzwApVKJc+fOYcuWLfi///s/
9OnTB8uWLcOePXvw8MMMPo127dsjJycH69esRHBwsGShclyeffBILFizAggUL4O3trXOmafr06cjLy8OD
Dz6I4OBgXL16FR999BF69eqlGVuTmZmJrl27YsqUKbXWI6qpoKAAX3/9NQCgtLRUUyH78uXLmDBDhAt5+
+22dPqxYsQLTp09Hnz59sGfPHly4cKHB56nm4eGBQYMGYeXKlaisrESbNm3w+++/IzU1VSe2d+/
eAIA33ngDEyZMgKOjI8aMGVNrIclFixbh22+/xahRozB79mx4e3vjyy+/
RGpqKrZv385q2mRbLDtZjoi0VU/lP3z4sM59SqhIeHi/
DwcM1U8suXL4tnnnlGBAQECEdHR9GmTRvxyCOPiG3btmke989//
lP07dtXeHl5CRcXF9GlSxfxr3/9S1RUVGhiapt+RUVGhiapt2f+fOHTF79mzRunVr0bJlSzFmzBiRkZGhM2VcCCF+/
/130b17d6FQKETnzp3F119/Xes2f/
rpJxEZGSmcnZ1FaGioePfdd8UXX3whM1285lR+IYSoqKgQ7777rujWrZtwcnISr179xZLly4VBQ
UFQgghdu3aJcaOHSuCgoKEQqEQUFBYuLEieLChQt6/w4eeOABAUBMnz5d575t27aJ4cOHCz8/
P6FQKETbtm3F3/72N5GVlaWJqZ5urz3NvS6DBw8WADQ/bm5uomPHju+/Kpp54Sv//
+e62PKS0tFc8995zw9PQU7u7u4sknnxQ5OTl1TuWvrazBtWvXxGOPPSa8vLyEp6eneOKJJ8T169dr/
d2+/fbbok2bNsLBwUHye6o5lV8I9fsxNjZWeHl5CWdnZ9G3b1/xyy+/SGKqp/
Jv3bpV0l5fmQIic5MJwdFvREREREEZGP0GRLDAdlEREREWpgcEREREWlhckRERES
khckREREkRYmR0RERERamBwREREaWFyRERERKSFyRERERGRFiZHREREFqYHBEREFpYXJEREREpIX
JEREREZEWJkdERERkPSqrLN0DtLB0B4iIiIiQXwScOK/
+v4sT0LeHxbrC5IiIiIgsp7wCOJAibXNxtkxf/
sLkiIiIiMxPpQKOnQVK7kjbO4cCAT4W6VI1JkdEREkXpfSgcwcaVugD9CxHSCTWaZPWpgcEEREkXnk5
AJnU6VtLs5A766AXG6ZPtWCyREREE1reJS4OgZ3fa+PdSDr60MkyMiIiJqGpVVwMEUQKmStvfoCHh7W
qZPemByREREKYlBHDqEpBXIG0PawO0DbRMnwzA5IiIiIhMJz0LSM2Utnl7At07WMVga30wOSIiIqLGu
10IpFyQtjk4AP0jAUfbSjdsq7dEREEkXcrKgYMnddt7RwBurubvjwkwOSIiIiLDKVXAsTNAaZm0vUsY4
N/aMn0yESZHREREpD8hgIvpQNZNaXsbPyA8xGbFdWHyREREHpJ/
sWcD5N2tbSBbi3q3p8UTPB5IiIiIjqV1SiXgetpn6RgLPC/
P1pYkyOiIiIqHaVlUBSivpSmrbITkArD8v0yQyYHBEREZGUEOpp+flF0vbwYCDYCA4wDJ9MiMmR0RERHTX1
etA2nVpm48XEBHeLAZb64PJEREREQG5BcCpi9K2FnKgXw+hX2lC/
b1aomIiEjqThlw6JRue59V9u6plodojJERERkT1SKoEjp4M4GyCml7RHvA19syfRvB9U0skyfmgkmR0RERLZECOBcKpCT
J21vGwCEBVumTzvkmR0RERLZECOBcKpCTJ21vGwCEBVumT8
0MkyMiIiJbkZkDXEqXtnm6AZGdWMTRhKxmT+7ZswdjxoxBUFAQZDIZfvjhB8n9U6dOhUwmk/
yMHDlSEpOXl4fJkyfDw8MDXl5ee06551BcLJ3GmJKSgoEDB8LZ2RkhISFYuXJlU780IiKixikoAnYf0U
2MonqyunUTsJozRyUlJejjZsyeeffZZxMTE1BozcuRIbNy4UXPbycl5ycOlJcv/
kyZORlZWFnTt3orKyEtOmTcOMGTcOMGTOwefNmAEBhYSGGDx+OYcOGYcOGDTh58iSeffZZeHl5YcaMGU334oi
IiIxRXgEcSNFtv6cL4OFm/
v7YCatJjkaNGoVRo0bVG+Pk5ISAgIBa7zt79ix27NiBw4cPo0+fPgCAjz76CKNHj8Z7772HoKAgfPPPPNN
6ioqMAXX3wBhUKBbt26ITk5GatXr2ZyRERE1kOlApLPAUWl0vZO7YBAX8v0yY7Y1Hm4hIQE+Pn5oXPnz

```
njxxReRm3t3ReGkpCR4eXlpEiMAGDZsGBwcHHDw4EFNzKBBg6BQ3F1HZsSIETh//
jxu375d63OWl5ejsLBQ8kNERNRkLmcAicekiZF/a2BQbyZGZmIzydHIkSPxn//
8B7t27cK7776L3bt3Y9SoUVAqlQCA7Oxs+PlJS6K3aNEC3t7eyM7O1sT4+/
tLYqpvV8fUtHz5cnh6emp+QkJCTP3SiIiIgJu31eOKrt242+asAAbcw+rWZmY1l9UaMmHCBM3/e/
TogcjISISHhyMhIQFDhw5tsud9/fXXMX/+fM3twsJCJkhERGQ6JXeAI6d12/
t2B1xYxNESbCY5qql9+/
bw8fHBpUuXMHToUAQEBCAnJ0cSU1VVhby8PM04pYCAANy4cUMSU327rrFMTk5OOgO/
iYiIGq2qCjh4Ul3MUVv3jkBrT8v0iQDY0GW1mq5du4bc3FwEBgYCAKKiopCfn4+jR49qYv7880+oVCr0
69dPE7Nnzx5UVt6tJrpz50507twZrVqxmigREZmBEOo10PYlSxOjdkHA4D5MjKyA1SRHxcXFSE5ORnJy
MgAgNTUVycnJSE9PR3FxMRYuXIgDBw4gLS0Nu3btwtixY9GhQweMGDECANC1a1eMHDkSzz//
PA4dOoR9+/Zh1qxZmDBhAoKCggAAkyZNgkKhwHPPPYfTp0/ju++
+w9q1ayWXzYiIiJrMtRvAnqPqRWKrtfJQD7YODbJYt0hKJoQQlu4EoJ6JNmTIEJ32KVOm4JNPPsG4ceN
w/Phx5OfnIygoCMOHD8fbb78tGWCdl5eHWbNm4eeff4aDgwMef/
xxfPjhh3Bzu1sLIiUlBS+99BIOHz4MHx8fvPzyy3jttdf07mdhYSE8PT1RUFAADw+Pxr1oIiKyD/
lFwInz0jaZDIiKBBwdLdMnO2PI8dtqkiNbweSIiIj0VlYBHKyliOO9XQH3lubvjx0z5PhtswOyiYiIrJ
ZKBRw7q56Jpq1zKBDgY5Eukf6YHBEREZmKEOoijpnS2dMI9AU6tmWtIhvB5IiIiMgUcnKBs6nSNldn4N
4IQG41859ID0yOiIiIGqO4FDh6Rre9Xw/
AmXXybBGTIyIiImNUVgEHUtTji7T16Ah4s1aRLWNyREREZAghgFOXgLwCaXtYG6BtoGX6RCbF5IiIiEh
f6VlAaqa0zdsT6N6Bg62bESZHREREDckrAE5elLbJHYB+kYAjD6XNDX+jREREdSkrVy8OW1PvCMDN1fz
9IbNgckRERFSTUqWegXanTNreNQzwa22ZPpHZMDkiIiKqJgRwMR3IuiltB+MHhIdwXJGdYHJEREQEANm
3gPNp0raWLup10BxYxNGeMDkiIiL7VlSiXgetpn6RgLPC/
P0hi2NyRERE9qmiEkg6odvesxPgVf+q7dS8MTkiIiL7IgSQcgHIL5K2h4cAwf6W6RNZFSZHRERkP65eB
9KuS9t8vICIcA62Jg0mR0RE1Pzl5quX/NDWogXQr7v6XyItfEcQEVHzdacMOHRKt71PN/
VMNKJaMDkiIqLmR6kEDp8Gyiuk7RHtAV9vy/SJbAaTIyIiaj6EUNcqupErbQ/
2Vw+4JtIDkyMiImoesm4CF65K29xbAr06s4gjGYTJERER2bbCYuD4Od32/
pGAE4s4kuGYHBERkW2qq4hjr86Ap7v5+0PNBpMjIiKyLSoVcOKC+oyRtg5t1QvEEjUSkyMiIrIdqdeA9
Gxpm5830CWMRRzJZJgcERGR9bt1Gzh9WdqmcATu6w60kFumT9RsMTkiIiLrVVoGHK6liON93QFXZ/
P3h+wCkyMiIrI+VUp1UlRRKW3v1kG9FhpRE2JyRERE1kMI4GwqcDNP2t42EAhrY5k+kd1hckRERNYhMw
e4lC5t83QDIjuxiCOZFZMjIiKyrIIiIPm8bntUT/
WgayIzY3JERESWUV4BHEjRbb+nC+DhZv7+EP2FyREREZmXSgUknwOKSqXtndoBgb6W6RORFiZHRERkPp
czgGs3pG0BrYFOoSziSFaDyRERETW9m3nAmSvSNmcF0KcbIGcRR7IuTI6IiKjplNwBjpzWbe/
bHXBhEUeyTkyOiIjI9KqqgIMn1cUctXXvCLT2tEyfiPTE5IiIiExHCPUaaLn50vbQIKBdkEW6RGQoJkd
ERGQa17KBy9ekba08gB4dOdiabAqTIyIiapz8QuDEBWmbTAZERQKOLOJItofJERERGaesAjhYSxHHe7s
C7i3N3x8iE2FyRERERhlGpgGNn1TPRtHUOBQJ8LNIlIlNickRERPoRAriUAVzPkbYH+QId2nJcETUbTI6
IiKhhN3KBc6nSNldn4N4IQO5gmT4RNREmR0REVLfiUuDoGd32fj0AZyfz94fIDJgcERGRrsoq4ECKeny
RtshO6un5RM0YkyMiIrpLCODURSCvUNoe1gZoG2iZPhGZGZMjIiJSS88CUjOlba09gW4dONia7AqTIyI
ie5dXAJy8KG2TOwD9IgFHHibI/
vBdT0Rkr+6UA4dO6rb3jgDcXM3fHyIrweSIiMjeKJXqGWh3yqXtXdsDft6W6RORFWFyRERkL4QALl4Fs
m5J29v4qYs4EhEAJkdERPYh+xZwPk3a5uYC3NMVcGARRyJtTI6IiJqzohL1Omg19Y8EnBTm7w+RDWByR
ETUHFVUAkkndNt7dga83M3fHyIbwuSIiKg5EQJIuQDkF0nbw0OAYH/
L9InIxjA5IiJqLtKuA1evS9t8WgER7VnEkcgATI6IiGxdbj5w6pK0zbEF0Lc70IIf80SG4l8NEZGtulM
GHDql296nG9DSxfz9IWommBwREdkapRI4f
Boor5C2R4QDvq0s0yeiZoTJERGRrRBCXavoRq60PSQAaB9skS4RNUdMjoiIbMH1m+rq1trcWwK9OrOII
5GJMTkiIrJmhcXA8XO67SziSNRkmBwREVmjuoo49uoCeLqZvz9EdoTJERGRNVGpgBMX1GeMtHVsCwT5W
aZPRHaGyRERkbW4cg3IyJa2+XkDXcJYxJHIjKxmFN+ePXswZswYBAUFQSaT4YcffpDcL4TAW2+9hcDAQ
Li4uGDYsGG4ePGiJCYvLw+TJ0+Gh4cHvLy88Nxzz6G4WPrtKyUlBQMHDoSzszNCQkKwcuXKpn5pRET1u
3Ub2H1EmhgpHIEH7gG6sro1kblZTXJUUlKCnj174uOPP671/pUrV+LDDz/
Ehg0bcPDgQbRs2RIjRoxAWVmZJmby5Mk4ffo0du7ciV9+
+QV79uzBjBkzNPcXFhZi+PDhaNeuHY4ePYpVq1ZhyZIl+Oyzz5r89RER6Si9o06KTl+Wtt/
XHYjqCbSQW6ZfRHZOJoQQlu5ETTKZDN9//z3GjRsHjQHQH3WKCgoCK+88goWLFgAACgoKIC/
vz82bdqECRMmfjkKs4iIiMDhw4fRp08fAMCOHTswZdqRoXLt2DUFBQfjkk0/
wxhtvIDs7uqIDs7ZGwqFepbHokWL8MMPP+DcuVpmg9SisLAQnp6eKCgogIeHh+lfPBE1f1VK4NBJoLJK2t6tA+D
jZZEuETV3hhy/rebMUX1SU1ORnZ2NYcOGado8PT3Rr18/
JCUlAQCSkpLg5eWlYXBwcMDBgwc1MYMGDdIkRgAwYsQInD9/
Hrdv3zbTqyEiuyUEcOYysO+4NDFqGwgM7sPEiMhK2MSA7Oxs9XV4f39/Sbu/v7/
mvuzsbPj5SWdytGjRAt7e3pKYsLAwnW1U39eqlW7Z/
fLycpSXl2tuFxYWNvLVEJFdyswBLqVL2zzdgMhOLOJIZGVsIjmypOXLl2Pp0qWW7gYR2aqCIiD5vG57V
E/1oGsisjo28XUlICAAAHDjxg1J+40bNzT3BQQEICcnR3J/VVUV8vLyJDG1bUP7OWp6/
fXXUVBQoPnJyMho/AsiouavvEI92LpmYnRPF/
UlNCZGRFbLJpKjsLAwBAQEYNeuXZq2wsJCHDx4EFFRUQCAqKgo5Ofn4+jRo5qYP//
8EyqVCv369dPE7NmzB5WVlZqYnTt3onPnzrVeUgMAJycneHh4SH6IiOqkUgFHzwAHUqTtndqpkyIPVrc
msnZWkxwVFxcjOTkZycnJANSDsJOTk5Geng6ZTIa5c+fin//8J3766SecOnUKb7zxBnx8fDBlyhQA
8J3766SecSecPHkSzzzzzH2zzzDIKCgjQz2rp27YqRI0fi+eefx6FDh7Bv3z7MmjULL7/8Mnx8fAAA
553D69Gl89913Gl89913WLt2LebPn4/hV01EzcrlDCDxGFBcerctwAcY1BsI9LVcv4jIIFYzlT8hIQFDhgzRaZ8
```

```
yZQo2bdoEIQQWL16Mzz77DPn5+RgwYADWr1+PTp06aWLz8vIwa9Ys/Pzzz3BwcMDjjz+ODz/
8EG5ud7+ppaSk4KWXXsLhw4fh4+ODl19+Ga+99pre/
eRUfiLScTMPOHNF2ubsBPSJAOSsVURkDQw5fltNcmQrmBwRkUbJHeDIad32vt0BF2fz94eI6mTI8Zuz1
YiIDFVVBRw8qS7mqK1HR8Db0zJ9IiKTYXJERKQvIdRLfeTmS9tDg4B2QRbpEhGZHpMjIiJ9ZGQDV65J2
1p5qM8WcWFYomaFyRERUX1uFwIpF6RtMpm6iKMjP0KJmiP+ZRMR1aasXD2uqKZ7IwB3V/
P3h4jMhskREZE2lQo4dlY9E01blzDAv7Vl+kREZsXkiIgIUA+2vpQOXL8pbQ/
yBTq05bgiIjvC5IiI6EYucC5V2ubqrL6EJreahQSIyEyYHBGR/SouVa+DVlO/
HuoK10Rkl5gcEZH9qawCDpwAVDUWCIjspJ6eT0R2jckREdkPIYCTF9XT87W1DwZCAizTJyKyOkyOiMg+
pGcBqZnSttaeQLcOHGxNRBJMjoioecsrUJ8t0iaXq8cVsYgjEdWCnwxE1DzdKQcO1VLEsXcE4MYijkRU
NyZHRNS8KJXqGWh3yqXtXdsDft6W6RMR2RSjkqOMjAzIZDIEBwcDAA4dOoTNmzcjIiICM2bMMGkHiYj0
IgRw4SqQfUva3sZPXcSRiEhPRlU3mzRpEuLj4wEA2dnZeOihh3Do0CG88cYbWLZsmUk7SETUoOxbwJ6j
0sTIzRUYeC8TIyIymFHJ0alTp9C3b18AwJYtW9C9e3fs378f33zzDTZt2mTK/
hER1a2oBNh9BDifJm3vH6keW+TA6tZEZDijLqtVVlbCyUldPfaPP/
7Ao48+CgDo0qULsrKyTNc7IqLaVFQCSSd023t2Brzczd8fImpWjPpa1a1bN2zYsAGJiYnYuXMnRo4cCQ
C4fv06WrfmqtVE1ESEAJLP6yZG4SHA4D5MjIjIJIw6c/Tuu+/
isccew6pVqzBlyhT07NkTAPDTTz9pLrcREZlUWiZwtcaZaZ9WQER7FnEkIpOSCSFEw2G6lEolCgsL0ap
VK01bWloaXF1d4efnZ7IOWpvCwkJ4enqioKAAHh5cg4moyeXmA6cuSdscWwB9ewAt5BbpEhHZHkOO30b
XORJC4OjRo7h8+TImTZoEd3d3KBQKuLqyuBoRmcCdMuDQKd32Pt2Ali7m7w8R2Q2jkqOrV69i5MiRSE9
PR3l5OR566CG4u7vj3XffRXl5OTZs2GDqfhKRvVAqgcOngfIKaXtEOODbqvbHEBGZkFEDsufMmYM+ffr
rg9u3bcHG5+w3usccew65du0zWOSKyI0IA51KBvceliVFIgHqwNRMjIjITo84cJSYmYv/+/
VAoFJL20NBQZGZm1vEoIqI6XL8JXLwqbXNvCfTqzFpFRGR2RiVHKpUKSqVSp/
3atWtwd+dUWiLSU0ExkHxOt71/
JOCk0G0nIjIDo76SDR8+HGvWrNHclslkKC4uxuLFizF69GhT9Y2ImquKSnVl65qJUa8u6ktoTIyIyIKM
msp/7do1jBgxAkIIXLx4EX369MHFixfh4+ODPXv2cCo/
EdVOpQJOnAcKS6TtHdsCQc33c4OILM+Q47fRdY6qqqrw3//+FykpKSguLsa9996LyZMnSwZoN0dMjoiM
dOUakJEtbfPzBrqEsYgjETU5s9Q5atGiBZ566i6iljH05E9uLWbeD0ZWmbkwK4rxsgZxFHIrI+eidHP/
30k94brV6IlojsWOkddb2imu7rDrg6m78/
RER60js5GjdunF5xMpms1plsRGQnqpTAoZNAZZW0vXsHoLWXRbpERGQIvZMjlUrVlP0gIlsnBHD2CnDz
trS9XSAQ2sYyfSIiMoLRY46IiDQybwCXMqRtnu5AZEcWcSQim2N0clRSUoLdu3cjPT0dFRXSNZBmz57d
6I4RkQ3IL1JPza8pqiegcDR/f4iITMCo5Oj48eMYPXo0SktLUVJSAm9vb9y6dQuurq7w8/
NjckTU3JVXAAdSdNvv6Qp4tDR/
f4iITMio893z5s3DmDFjNAvPHjhwAFevXkXv3r3x3nvvmbqPRGQtVCrg6BndxKhTqLqyNRMjImoGjDpz
lJycjE8//RQODg6Qy+UoLy9H+/
btsXLlSkyZMgUxMTGm7icRWdqlDPXYIm0BPkCndiziSETNilHJkaOjIxz+GmTp5+eH9PR0dO3aFZ6ens
jIyGjg0URkU3Ly1LPQtLk4Ab0jWMSRiJolo5Kje+65B4cPH0bHjh0xePBgwuordO/
e3dR9JCJLKC5VX0KrqW8PdXJERNRMGTXm6J133kFgYCAA4F//
+hdatWqFF198ETdv3sRnn31m0g4SkZlVVQF7j+smRj06qscVMTEiombO6IVn7RUXnqVmSwj1Gmi5+dL2
0CCgXZBFukREZCpmWXiWiJqRjGzgyjVpWysP9dkiDrYmAymVSiQmJiIrKwuBgYEYOHAg5ByfRjbEqOQo
LCwMsno+MK9cuVLnfURkRW4XAikXpG0OMqB/
T8CR353IcHFxcZgzZw6uXbubbAcHB2Pt2rWcyUw2w6hPv7lz50puV1ZW4vjx49ixYwfYsGHDhg0bNmzY
KwcWLlxoin4RUVMqKwcOntRtvzcCcHc1f3+oWYiLi0NsbCxqjtbIzMxEbGwstm3bxgSJbIJJxxx9/
PHHOHLkCDZu3GiqTVodjjkim1ZdxLG0TNreJQzwb22ZPlGzoFQqERoaKjljpE0mkyE4OBipqam8xEYWY
cjx26QrQo4aNQrbt2835SaJyBSEAC5eBRKPSROjIF9gUG8mRtRoiYmJdSZGACCEQEZGBhITE83YKyLjm
HRQwbZt2+Dt7W3KTRJRY93IBc6lSttcndWX0OQm/X5EdiwrK8ukcUSWZHQRSO0B2UIIZGdn4+bNm1i/
fr3JOkdEjVBUChyrpYhjvx6AM2sVkWlV174zVRyRJRmVHI0bN05y28HBAb6+voiMlZl
FZB0Qn0pTVtkJ/
X0fKImMHDgQAQHByMzM1NnQDZwd8zRwIEDLdA7IsMYlRwtXrzY1P0gosYSAjh5UT09X1v7YCAkwDJ9Ir
shl8uxdu1axMbGQiaTSRKk6isNa9as4WBssgl6J0eFhYUNB/
2Fs7iIzOxqFpCWKW1r7QV0C2cRRzKbmJgYbNu2rdY6R2vWrOE0frIZek/
ld3BwqLfwozalUtmoTlkzTuUnq5JXoD5bpE0uV48rYhFHshBWyCZr1CTLh8THx2v+n5aWhkWLFmHq1Km
IiooCACQlJeHLL7/
E8uXLjew2EentTjlwqJYijr0jADcWcSTLksvliI6OtnQ3iIxmVBHIoUOHYvr06Zg4caKkkffPmzfjss8+
QkJBgqv5ZHZ45IotSKoEjZ9QVrrV1bQ/
4sYwGEVFdmrwIZFJSEvr06aPT3qdPHxw6dMiw8dMiYTRJRfYQAzqcBe49LE6M2/
sDgPkyMiIhMyKjkKCQkBJ9//rlO+7///W+EhIQ0ulNEpCXrFrDnKJB9626bmysw8F6gA//
eiIhMzagRmx988AEef/xx/
Pbbb+jYCTwvz9ISKyE0YvPJuRkYFPPvkE586dAwB07doVAwB07doVL
7zwQrM/
c8QxR9TkKirVRRxr6tkZ8HI3f3+IiJoBQ47fRidH9orJETUZIYAT54GCYml7eAgQ7G+ZPhERNRNNMpU/
JSUF3bt3h4ODA1JSUuNjYyM1HezRASoCzherbEgp28r9Sw0FnEkIjIrvZOjXr16ITs7G35+fujVq5
efhqMpmsWReBJDKp3Hzg1CVpm2MLoG8PoAWL5hERWYLeyVFqaip8fX01/yeiRigtAw6f0m2/
rxvg6mL+/hARkYbeyVG7du1q/T8RGUCpBA6fBsorpO3dwgGfVpbpExERSRhV5+jLL7/E/
73P83tV199FV5eXrj//
0nJSVh3bp1WLlyJdXDvBv3ISKyE0YvPJuRkYFPPvkE586dAwB07doVAwB07doVL
0nJSVh3bp1WLlyJXx8fDBv3MM72+bRUl3EsX2w5fpFRES1Mio5ysjIQIcOHQAP/
```

zwA2JjYzFjxgwsX74ciYmJJu1gtSVLlkAmk0l+unTporm/rKwML730Elq3bg03Nzc8/
vjjuHHjhmQb6enpePjhh+Hq6go/
Pz8sXLgQVVVVTdJfIhQUA7uPABfTpe1RPYF7ugIORv35ERFREzOqQrabmxtyc3PRtm1b/P7775g/
fz4AwNnZGXfu3DFpB7V169YNf/zxh+Z2ixZ3uz9v3jz873//w9atW+Hp6YlZs2YhJiYG+/
btAwAolUo8/PDDCAgIwP79+5GVlYVnnnkGjo6OeOedd5qsz2SHyiuAA7WUu+jVBfB0M39/
iIjIIEYlRw899BCmT5+Oe+65BxcuXMDo0aMBAKdPn0ZoaKgp+yfRokULBAQE6LQXFBTg//
2//4fNmzfjwQcfBABs3LgRXbt2xYEDB9C/f3/8/vvvOHPmDP744w/4+/ujV69eePvtt/
Haa69hyZIlUCi4HAM1kkoFJJ8Hikqk7R3bAUG+lukTEREZzKjz+h9//DGioqJw8+ZNbN+
+Ha1btwYAHD16FBMnTjRpB7VdvHgRQUFBaN+
+PSZPnoz09HTN81ZWVmLYsGGa2C5duqBt27ZISkoCoB4b1aNHD/
j73600PGLECBQWFuL06dN1Pmd5eTkKCwslP0Q6rlwDEo9JEyP/
1sCg3kyMiIhsjFFnjry8vLBu3Tqd9qVLlza6Q3Xp168fNm3ahM6dOyMrKwtLly7FwIEDcerUKWRnZ0Oh
UMDLy0vyGH9/
f2RnZwMAsrOzJYlR9f3V99Vl+fLlTfq6yMbdug2cvixtc1Ko6xXJWcSRiMgWGZUcAUBiYiI+/
fRTXLlyBVu3bkWbNm3w1VdfISwsDAMGDDBlHwEAo0aN0vw/MjIS/
fr1Q7t27bBlyxbNzLmm8Prrr2vGVAHqtVma+
+K6pIeSO8CRWs443tcdcHU2f3+IiMhkjLqstn37dowYMQIuLi44duwYysvLAajH/
phrcLOXlxc6deqES5cuISAgABUVFcjPz5fE3LhxQzNGKSAgQGf2WvXt2sYxVXNycoKHh4fkh+xYlRLYn
6ybGHXvoK5XxMSIiMjmGZUc/fOf/
8SGDRvw+eefw9HRUdP+wAMP4NixYybrXH2Ki4tx+fJlBAYGonfv3nB0dMSuXbs0958/
fx7p6emIiooCAERFReHkyZPIycnRxOzcuRMeHh6IiIgwS5/
Jhgmhvny27zhQqVX+oV2gOilq7WWxrhERkWkZdVnt/
PnzGDRokE67p6enztkbU1mwYAHGjBmDdu3a4fr161i8eDHkcjkmTpwIT09PPPfcc5g/
fz68vb3h4eGBl19+GVFRUejfvz9PPPfcc5g/
w0ksvwcnJqUn6TM3EtRvA5Qxpm5c7ENkJkMks0yciImoyRiVHAQEBuHTpks60/b1796J9+/
am6JeOa9euYeLEicjNzYWvry8GDBiAAwcOHDH38c5eXlGDFiBNavX695vFwuxy+//
IIXX3wRUVFRaNmyJaZMmYJly5Y1SX+pGcgvAk6c122P6gkoHHXbiYioWZAJIYShD1q+fDm+/
vprfPHFF3jooYfw/4urVq5g7dy7eeustvPzyy03RV6tQWFgIT09PFBQUcPxRc1VXEcd7uqqX/
SAiIptjyPHbqDNHixYtgkqlwtChQ1FaWopBgwbByckJCxcuxPTp043qNJHFqVTA8XNAcam0vXMoEOBjk
S4REZH5GTUgWyaT4Y033kBeXh5OnTqXBeXh5OnTqFAwcO4ObNm/
D09ERYWJip+0jU9C6lq4s4aidGAT7qIo5MjIiI7IpBZ47Ky8uxZMkS7Ny5U3OmaNy4cdi4cSMee+wxyO
VyzJs3r6n6SmR6tdUrcnECekewiCMRkZ0yKDl666238Omnn2LnjiCUybNg0HDhzA+++/
jyeeeAJyHlDIFlRUAmnXgayb0va+PdTJERER2S2DkqOtW7fiP/
5Dx599FGcOnUKkZGRqQqqwokTJyDjlGayBSqVemp+ehagVKnbfFoBwf6Ap5tl+0ZERFbBoOTo2rVr6N2
7NwCge/
fucHJywrx585gYkfUTAriZB1zJVM9GAwB3V6B9iLpmERER0V8MSo6USiUUCsXdB7doATc3ftsmK1dQrC
7iWFSivu3kCIQFA37eLOJIREQ6DEqOhBCYOnWqpqJ0WVkZXnjhBbRsKa39EhcXZ7oeEhnrTjmQeg24eV
t928EBaBugvoTGsXFERFQHg5KjKVOmSG4/9dRTJu0MkUlUVQFXs4DMHPXlNEA9HT+sDStbExFRgwxKjj
Zu3NhU/
SBqPJUKyLoFXL1+d3FYL3cgPARwc7Vs34iIyGYYVSGbyKoIAeQVAFeuAaVl6jZXZ6B9MODtyXFFRERkE
CZHZNuKS9WDrfOL1LcdWwDtgoBAH/
UYIyIiIgMxOSLbVF6hLuKYfUt9WyZTD7RuGwC04NuaiIiMx6MI2Ral8q8ijtnqMUYA4NtKPTWfla2JiM
gEmByRbRACuJ4DZGQD5ZXqNveW6sHWrGxNREQmxOSIrN/
V6+pLaNWcFOrB1r6tONiaiIhMjskRWa+8AuDkRWlbkK/6bBEHWxMRURNhckTW5045cOikbnufbkBLF/
P3h4iI7AqTI7IeSiVw5AxQVi5t79pevQ4aERGRGTA5IssTArhw9e60/GrB/
upLaGQxSqUSiYmJyMrKQmBgIAYOHAg516UjomaOyRFZVtZNdWKkzc0VuKcLxxVZWFxcHObMmYNr165p2
oKDg7F27VrExMRYsGdERE2LyRFZRmEJcPysbnv/
SPVsNLKouLg4xMbGQlQv3PuXzMxMxMbGYtu2bUyQiKjZkoman35Ur8LCQnh6eqKgoAAeHh6W7o7tqagE
kk7otvfsrF4klixOqVQiNDRUcsZIm0wmQ3BwMFJTU3mJjYhshiHHb163IPMQAkg+p5sYdQgBBvdhYmRF
EhMT60yMAEAIgYyMDCQmJpqxV0RE5sPLatT00jKBq1nSNt9W6lloLOJodbKyshoOMiCOiMjWMDmipnMr
Hzh9Sdrm2ALo2wNowcsx1iowMNCkcUREtobJEZleaRlw+JRu+33dAFcWcbR2AwcORHBwMDIzM3UGZAN3
xxwNHDjQAr0jImp6TI7IdKqU6qSoolLa3i0c8GllmT6RweRyOdauXYvY2FjZIDJJgiT76zLomjVrrH4w
Nms0EZGxOCCbGk8I4OwVYN9xaWIUEqAebM3EyObExMRg27ZtaNOmjaQ9ODjYJqbxx8XFITQ0FEOGDMGk
SZMwZMgQhIaGIi4uzTJdIyIbwKn8BuJU/
hqu5wAX06VtHm5Az04s4tgM2OLZl7pqNFWf9bKF5I6ITM+Q4zeTIwMxOfpLQbF6an5NUT0BhaP5+0ME1
mgioroZcvzmmCMyTHkFcCBFt/
2eLuozRkQWZEiNpujoaPN1jIhsCpMj0o9KBSSfB4pKpO0d2wFBvpbpE1ENrNFERKbA5IgaduUakJEtbf
NvDXQOZRFHsiqs0UREpsDkiOp28zZw5rK0zUmhrlfE8RpkhVijiYhMgdOJSFfJHWD3Ed3E6L7uQP9IJk
ZktaprNAF3Z6dVs6UaTURkWUyO6K6qKmB/MnDktLS9ewd1vSJXZ4t0i8gQtl6jiYgsj1P5DdQsp/
ILAZy5Aty6LW1vFwiEtqn9MURWzhZrNBFR0+FUftLftRvA5Qxpm5c7ENmJg63/0piDLA/
QliOXyzldn4iMwuTIXuUXASfO67aziKNEXFwc5syZI6mdExwcjLVr1zZ4eaYxjyUiMga/
kJkGL6sZyOYvq9VVxPHeroB7S/P3x4o1ZhkKLmFB9owHaMvgF7L6cfmQJmSzyZFKBRw/
BxSXSts7hwIBPhbpkjVrzDIUXMKC7BkP0JbBL2QNM+T4zdlq9uBSOpB4TJoYBfoAg3ozMaqDIctQmPKx
RLas+gBd8/2fmZmJ2NhYxMXFWahnzZtSqcScOXNqQXNqre1V3TZ37lwolUpzMTlqznLy1PWKMnPutrk4

```
AQPuATqFcsB1PRqzDAWXsCB7xAO05fALmelxQHZzVFwKHD2j0/
zTjVR4+PliIABTXsxpjuMLGrMMBZewIHvERX8th1/ITI/JUXNSWQUcTAGUKknz5BVLsHnH/
zS3TXn93xzjCyyRfDVmGQouYUH2iAdoy+EXMtPjZbXmQAjg1EV1dWutxOhUST4chvSVJEaA6a7/
m2N8QVxcHEJDQzFkyBBMmjQJQ4YMQWhoaJOPXWjMMhRcwoLsEQ/
QllP9hazm5001mUyGkJAQfiEzhCCDFBQUCACioKDA0l1RS88SIuGw9CflvKiqrBTBwcECQK0/
MplMhISEiKqqKqOetqqqqkm3L4QQ27dvr3P7AMT27duN3rYhfaj5
OkNCQvR67sY8lsjWVH8myGSyJvtMoLpt375dyGQynf1f3cbPHcOO35zKbyCrmcp/
uxBIuSBtc3BQLwzr2AIJCQkYMmRIg5uJj4836vp/
U29fqVTC398fubm5dca0bt0aN27caPIzMKyQTaSf6rPJACSXlDmd3DxqG+YQEhKCNWvWcL+Dy4c0S9UH
2YKcmxjrH6Yb0DsCcHPV3Gzq6/9Nvf2EhIR6EyMAyM3NRUJCAoYOHWrUc+irMctQcAkL28XE1nDVi/
7WNg6RB+imFxMTg7Fjx/J9awJMjmxAXFwcXn1lAX5asgLRoe2ld3YJA/
xb6zzGz89Pr23rG1dTU48v+PPPP/
WOa+rkiOwPCxkajwdoy+IXMtNgcmTl4uLikJ14CJc2bZG0fxT3HWZ/
+B62b9+OmJgYnW+5TV1LpKlnZKWnp5s0jkhfdVUarp5owEtDDeMBmmwdkyMrpsq6iZjWbYFxbTVtKZcv
ou+LU1FeUQEAmDJlClQqFebNmyf5luvt7a3Xc+Tk5DQcVIvqGVmxsbGQyWS1ji9ozIystm3bNhxkQByR
PhoqZCiTyTB37lyMHTuWZ0KImjFO5bdGRaXA7iNwuHBV0tz2yUfQ87lJmsQIAIqLi/HEE0/
oTKfPy8vT66kaM622enxBmzZtJO3BwcGN/nat77dOa/92qlQqkZCQgG+//
RYJCQmsDmxGxux7VhomIoBnjqxLZSWQlKKuW6RlyNwXkJB81ORPJ5fLcf/
99zdqGzExMRg1ahQWLlyIixcvomPHjli1ahVcXFwatV0HB/
3ydn3jLIHjVizH2H3PQoZEBPDMkfUorwD2n5AkRht374Qs+r4mSYwA9Tfr/
fv3N2obr776Ktzc3PDxxx/j999/
x8cffww3Nze8+uqrjdpudna2SePMjQtwWk5j9j0LGRIRwOTIelRW3f2/jxcwqDfyWiqa/
Gkb8w341VdfxapVq6BSSZcrUalUWLVqVaMSpJs3b5o0zpy4AKflNHbfs9IwEQFMjqyHmysQ1RMYeC/
QrQMgk+Hll1+u80PaVIz9BlxRUYH333+/
3pj3338fFVrjowzh6+tr0jhz4rgVy2nsvufSL0QEMDmyLgpHdZXrv8jlcrRs2bLehzQmeXJwcDB6zNG6
det0zhjVpFKpsG7dOqO2X3OQd2PjzInjVizHFPu+KScaEJFtYHJkxRITE1FcXFxvTPWlAmOSJJVKZfTZ
C30fZ+z2+/
XrZ9I4c+K4Fcsx1b6PiYlBWloa4uPjsXnzZsTHxyM1NZWJEZGdYHJkxfT9Fjx37lydb7nu7u56PVbfSt
Q1NXRGy9C4mvQ942TsmammxHErlmPKfV9dyHDixImIjo7mpTQiO8LkyErUVpNF32/BY8eO1fmW+
+ijj+r1WGMrTHfr1s2kcTX98MMPJo0zp+pxK3Wt6SyE4LiVJsIxQ0RkCkyOrEBcXBxCQ0MxZMgQTJo0C
UOGDEFoaChu3ryp97fgmt9y27Vrp9dzG1thOiUlxaRxNWVmZpo0juwHxwwRUWMxObKw+mqyjB8/
HhMnTgRg+LfgBx98UK/n1zeuprS0NJPG1VTXWRdj48ypejp5XaqXoOBU/qbDMUNE1BhMjixIn5os//
3vf7Flyxadb8Ft2rSp91twdHQ0WrduXe/zt27d2ujlNzIyMkwaV1NJSYlJ48yJU/
mtA8cMEZGxmBxZkL4H0TNnztRaaLE+crkczz77bL0xzz77rNEHjKqqqoaDDIirqaioyKRx5sSp/
NaB69oRkbHsNjn6+OOPERoaCmdnZ/Tr1w+HDh0yeX/0PTguXrwY169fl7Rdv34djz/
+eJ1LISiVSnzxxRf1bveLL74w+oDR1NPVmzr5akqcym95dY3j47ItRKQPu0yOvvvuO8yfPx+LFy/
GsWPH0LNnT4wYMQI5OTlm7YcpDo4zZsyoNcFJSEhAbm5uvY/
Nzc1FQkKCUc8bEBBg0riabHnhWU7ltyyua0dEjWV9RxYzWL16NZ5//
nlMmzYNERER2LBhA1xdXRs802JqDR1E9VFXgvPHH3/
o9Xh942rSN5E0NuFs0aKFSePMidPJLYfr2hGRKVjfkaWJVVRU4OjRo3j99dc1bQ4ODhg2bBiSkpL031B
JCVDbwU0uB5ydpXF1kDs4YO3atYiNjYVMJoOL1ge6DID2x7sKQJnWbZe/YgBg3++/Y2j//
nfvlMlw9OjRWmNrOlXzcuKdO0B945n+KupYVFQEZ9SfXZdCa0xQWRlQ3wFJu1hkWRl8XFyQV1ZWa2ip1
v+9XFzq3cdwdQWqE5TycqC+y3AuLneXb6moACorjY6NGTEC33/9NebNm4erOTmo3qMh/
v748P33MXbEiNr77ex8931VWanedl2cnIDq5NCQ2Koq9b6oi0IBODoaHqtUqn/
PdXF0VMcbGqtSqd+XesQm7t6NvGvX4FpHaJXWYPjowYOB0tI6IqHeX05O6v8LUX+sAX/
3BsU6OKjfa8bElpaq+10bmUz9t2FMrJ6fEQbHGvgZUW+sIX/
3FvqMkND+uzcklp8R6v8b8BmhN2FnMjMzBQCxf/
9+SfvChQtF3759deLLyspEQUGB5icjI0MAEAXqjxLdn9GjpRtwda09DhBi8GAhhBDbt28XwcHBIqeuOE
AcUudKmp/UemJFRIQYO3asJvZUPbE3XFyk/e3Tp+7t+vhowvr37y/i69lu8V/P3b9/f/
UDRo+ue7s134axsfXGumrth61ubvVvNyfn7nZnzqw/
NjX1buyCBfXHnjp1N3bx4npj+2j1d0F92wSEiI+/u9116+qP/
eWXu7EbN9Yfu2XL3dgtW+qP3bjxbuwvv9Qfu27d3dj4+PpjV668G3voUP2xixffjT11qv7YBQs0oT+uX
Vtv7Lq/
fg+bN29Wvzfq2+6UKXf7UFxcf2xsrPQ9XF+sEZ8RGj4+dcf26SONbdeu7tiICGlsRETdse3aSWP1/
IwQQqj7X1esq6s01oSfEaK4+G7slCn1x1rBZ4Q4dOhu7MqV9cfyM0L9Y8RnREFBgQAgCgoKREPs8rKaI
ZYvXw5PT0/
NT0hIiMmfo7omi6enp8m2qe94Fiftb7AGcNH+hmqCOGNV1fcNi+yOn5+fXnEcDE9E9ZEJIYSlO2FOFRU
VcHV1xbZt2zBu3QpU5Cfn48ff/xREl9eXo5yrVOGhYWFCAkJQcH16/
Dw8NB9AhOeMo+Pj8cjY8YAqP+y2i8//4whQ4bcvVMmQ0WLFnB2doYQos7LajIAebdvQ+HldbdRz9PgvX
r1wvkTJxq8rNazZ08kJycbfMrco2VLKOvoh/bFDRcHB5QWFta9XQucMi8uLoa/
1kD0MkBzWc3xrx8AuJGdDTc3N+l2ecpcN9aAU+bKykp0DQ3F9evXUdsHmxKAX0gIUlNTIXdw4GU1Y2J5
WU39f15WMzzWwp8RhYWF8PT0REFBQe3Hby12N+ZIoVCgd+/
e2LVrlyY5UqlU2LVrF2bNmoVTkBKfqD0htLVtK/
1jroufCq0qlEomHDyMrKwuBgYEYOHAgsouKUNfHsfbbILuoSOd55EolWrZsieLiYtT1lnFzc4O85gK1e
```

```
p7pyczMRD1vW0kcAOnBoCHOzihqoI5TtTsqld77GE5Odw92DVEo9L9GXSP26aeeqvP3VvnXDwA8/
cIL+P777+verqPj3Q+VhhgS26LF3Q9BU8bK5fr/
LgyJdXDQO1bu6IgVH32E2NhYAID2d79aB8Pr2weZTP9YQ7bblLGudY28amSsIWeDDYk18DNCb4b83Zvp
M8JksfyMUDPgM0JfdnlZbf78+fj888/x5Zdf4uzZs3jxxRdRUlKCadOmWaQ/
ddVkuXjxol6Pr+0SQWJiIoqLi+t9XHFxsdFVmhV6/
vHqG1eTvjP4GjPTr6lcvnzZpHFkmOq11YKCgiTtDVWVJyKqZndnjgBg/
PjxuHnzJt566y1kZ2ejV69e2LFjB/z9/
c3el+qaLDWvbmZmZmLJkiVwc3OrN8lp3bp1reOLmrpKc8+ePXUKU9YVZwwvLy/
cvn1brzhrEx4ejpMnT+oVR03HGhNnIrINdnnmCABmzZqFq1evory8HAcPHkS/
fv3M3oeGarIIIVBa3ziHejR1lWZ9kxJjk5eoqCiTxpnTV199ZdI4MgyLQBJRY9ltcmQNGlpbDWh4DbXc
3NxaL43pm+wZmxTeqW/
wmxFxNQ0fPtykcebk5ubW4Fmh8PBw3cHY1GimLALJtdmI7BeTIwsy1cKjtW3n008/1eux+sbVNGDAAJP
G1bR06VKTxpmTUqmUzHCsTUVFBQ+2TUDfxZwbGmvHtdmI7BuTIwsyVa2V2rbT1IOCX3755QbXNXNwcMD
LL79s1PYL65ueb0ScOelzRlCfAzQZzhRj7XhZjoiYHFmQKdZWk8vluP/+
+3XaQ0ND9Xq8vnE1KRQKvPLKK/
XGvPLKK0bPVqu1fEIj4sypqQfDU90aO9aOa7MREcDkyKL0WaC0IUqlEvv379dp79Gjh16P1zfO3Gy5/0
09GJ7q1tAXDplMhpCQkDoryJvqshwR2TYmRxZWXZOlTZs2kvbg4GDMnTtXr23Udgbi5s2bej1W37iaKi
oqsHr16npjVq9ejYr6KrLW49SpUyaNq4+pB9429gBNxtPnC4ekCGQNPOtHRACTI6tQvbZafHw8Nm/
ejPj4eKSmpmLs2LF6Pb62MxA3btzQ67H6xtW0fv36BpMIpVKJ9evXG7X9qvpK+BsRV5emGHjb2AM0NU5
9XzgaKgLJs35EBNhpEUhrJJfLER0dLWmrPgORmZlZ6xgImUyG4ODgWs9A5Obm6vW8+sbV1NQDvt3c3Bq
c8VUdZ6z6CnDGxsY2qppy9QF6zpw5kss0wcHBWLNmDas0N7GYmBiMHTsWiYmJkiV5GkpIG/
M3R0TNB88cWbHGnIFoaCaZoXE16Vvd2dgq0DUTxcbG1aRPAc7GDryt64wgEyPzqP7CMXHiRERHR+t1po
5n/
YgIYHJk9Yy9RNDUycXMmTMbPEDI5XLMnDnTqO076rlAor5xNZlrur0xB2iyrMZcliOi5oGX1WyAMZcIo
qOj0bp163ovm7Vu3dro5EihUGD+/PlYtWpVnTHz5883eip/Uw8oz8zMNGkcNS/
GXpYjouaByZGNqG1MUkPxn332GR5//
PE6Yz777LNGfdivXLkSgHpWmvblJ7lcjvnz52vuN4a+Y4mMHXPU1MkX2T5D/
+aIqPngZbVmLCYmBtu3b0dwcLCkPTg4GNu3bzfJ5YGVK1eitLQUH3zwAWbNmoUPPvgApaWljUqMAOCBB
x4waVxNvr6+Jo0jIqLmg2eOmjlzXB5QKBR612TSV2RkpEnjaqo5nqSxcURE1HwwObIDTX15QKlUmjz52
rt3r95xI0aMMHj71VO26xuUzUKNRET2iZfVqFFsdfXy6inb9VWx5pRtIiL7xOSIjNaUq5c3dSkC4O6U7
ZpjskJCQjhlm4jIjslEbVXwqE6FhYXw9PREQUEBPDw8LN0di1EqlQgNDa3zslR1JeHU1FSjzr4olUr4+
/
s3WIrgxo0bjT670xSXBYmIyLoYcvzmmCMyiiGrlxtzdsccpQi0n4tTtomIqBovq5FRzLF6eXUpgtoqFZ
uqFAHVr6KiAmvWrMHLL7+MNWvWoKKiwtJdIiJqcjxzREYx1+rlrFRsOa++
+qpOgc8FCxY0usAnEZG145gjA3HMkVr1mKOGVi83dswRWdarr75a79IwCxcuZIJERDbFkOM3L6uRUbh6
efNVUVGB1atX1xuzevVqXmIjomaLyREZjauXN0/
r16+XXEqrjVKpxPr1683UIyIi8+KYI2oUjglqfi5fvmzSOCIifVlLaRUmR9RonArfvISHh5s0johIH3F
xcZgzZ46kTExwcDDWrl1r9isRHJBtIA7IpuauoqICrq6u9V5ak8vlKC0thUKhMGPP7EtFRQXWr1+Py5c
vIzw8HDNnzuT+pmaresWFmilJ9RhWUwzV4IBsIjKaQqHA/
Pnz642ZP38+D9RN6NVXX4WrqyvmzZuHdevWd68eXB1dcWrr75q6a4RmZxSqcScOXNqnflc3TZ37twGx
0KaEpMjItKxcuVKLFy4UOdav1wu5zT+JlZdRqHmgUCpVGLVqlVMkKjZMWTFBXPhZTUD8bIa2RNe2jEvX
tIke/
Ttt99i0qRJDcZt3rwZEydHNPp5uLYaEZmEQqHA3LlzLd0Nu2FIGQX+Xqi5MNeKC4bgZTUiIivBMgpkjw
YOHIjg4GCdgsLVZDIZQkJCMHDgQLP1ickREZGVYBkFskfWuOICxxwZiGOOiKipcMwR2bPa6hyFhIRgzZ
o1JqlzxKn8REQ2iGUUyJ7FxMQgLS0N8fHx2Lx5M+Lj45GammqRpag4IJuIyIpUl0lYvXq15AySXC7H/
PnzWUaBmjVrWXGBl9UMxMtqRGQOLKNAZFqGHL+ZHBmIyREREZHt4ZgjIiIiIiMxOSIiIiLSwuSIiISS
AuTIyIiIiItTI6IiIiItDA5IiIiLC5IiIiIhIC5MjIiIiIi1MjoiIiIi0MDkiIiIi0sLkiIiIiEgLk
yMiIiIiLQwOSIiIiLSwuSIiIiISAuTIyIiIiItTI6IiIiItDA5IiIiLC5IiIiIhISwtLd
4CIiHQplUokJiYiKysLgYGBGDhwIORyuaW7RWQXmBwREVmZuLg4zJkzB9euXdO0BQcHY+3atYYiJibFgz
4jsAy+rERFZkbi4OMTGxkoSIwDIzMxEbGws4uLiLNQzIvvB5IiIyEoolUrMmTMHQgid+6rb5s6dC6VSa
e6uEdkVJkdERFYiMTFR54yRNiEEMjIykJiYaMZeEdkfJkdERFYiKyvLpHFEZBwmR0REViIwMNCkcURkH
CZHRERWYuDAgQgODoZMJqv1fplMhpCQEAwcONDMPSOyL0yOiIishFwux9q1awFAJ0Gqvr1mzRrWOyJqY
kyOiIisSExMDLZt24Y2bdpI2oODg7Ft2bWOSIyA5tJjkJDQyGTySQ/
K1askMSkpKRg4MCBcHZ2RkhICFauXKmzna1bt6JLly5wdnZGjx498ZpaWmIj4/
H5s2bER8fj9TUVCZGRGZiUxWyly1bhueff15z293dXfP/wsJCDB8+HMOGDcOGDRtw/
Dy8sKMGTMAAPv378fEiROxfPlyPPLII/
PUREdZHL5YiOjrZ0N4jskk0lR+7u7ggICKj1vm++
+QYVFRX44osvoFAo0K1bNyQnJ2P16tWa5Gjt2rUYXOXIkFi5cCAB4+
+23sXPnTqxbtw4bNmww2sgIiIi62Uzl9UAYMWKFWjdujXuuecerF+/3tzcrZ7nzmce+998Lb2xv/79+/
BEjcP78edy+fVy+fVsTM2zYMMk2R4wYgaSkpR/wsJCDB8+HMOGDcOGDRtw/
ny5Vi6dKfd/
H66+/jqysLKxevXrRoAkJ2djbtbCwMMlj/P39Nfe1atatUK2dnZmZjbtmOzs7Dqfd/
Hx999/Nfe1atUK2dnZmZjbtmOzs7Dqfd/
fqhqqoKaWlp6Ny5MwICAnJTPXt6nFKdcXUNY4JAJycnODk5NTQSyEiIqJmwqLJka+vL3x9fY16b
```

HJyMhwcHODn5wcAiIqKwhtvvIHKyko4OjoCAHbu3InOnTujVatWmphdu3Zh7ty5mu3s3LkTUVFRjXshR
ERE1GzYxIDspKQkrFmzBidOnMCVK1fwzTffYN68eXjqqac0ic+kSZOgUCjw3HPP4fTp0/juu+
+wdu1azJ8/X7OdOXPmYMeOHXj//
fdx7tw5LFmyBEeOHMGsWbMs9dKIiIjIysiEEMLSnWjIsWPHMHPmTJw7dw7l5eUICwvD008/
jfnz50sueaWkpOCll17C4cOH4ePjg5dffhmvvfaaZFtbt27FP/
7xD6SlpaFjx45YuXIlRo8erXdfCgsL4enpiYKCAnh4eJjsNRIREVHTMeT4bRPJkTVhckRERGR7DDl+28
xUfmtRnUuy3hEREZHtqD5u63NOiMmRgYqKigAAISEhFu4JERERGaqoqAienp71xvCymoFUKhXWuX78Od3
d3yGQyS3fHZhQWFiIkJAQZGRm8HGkg7jvjcd8Zj/uucbj/
jNdU+04IgaKiIgQFBcHBof75aDxzZCAHBwcEBwdbuhs2y8PDgx8URuK+Mx73nfG47xqH+894TbHvGjpj
VM0mpvITERERmQuTIyIiIiItTI7ILJycnLB48WIuxWIE7jvjcd8Zj/uucbj/
jGcN+44DsomIiIi08MwRERERkRYmR0RERERamBwRERERaWFyRERERKSFyRGZzJIlSyCTySQ/
Xbp00dxfVlaGl156Ca1bt4abmxsef/xx3Lhxw4I9tqw9e/ZgzJgxCAoKgkwmww8//
CC5XwiBt956C4GBgXBxccGwYcNw8eJFSUxeXh4mT54MDw8PeHl54bnnnkNxcbEZX4VlNLTvpk6dqvNeH
DlypCTGHvfd8uXLcd9998Hd3R1+fn4YN24czp8/L4nR5+80PT0dDz/8MFxdXeHn54eFCxeiqqrKnC/
F7PTZd9HR0TrvuxdeeEESY4/7DgA+
+eQTREZGago7RkVF4bffftPcb23vOyZHZFLdunVDVlaW5mfv3r2a++bNm4eff/
4ZW7duxe7du3H9+nXExMRYsLeWVVJSgp49e+Ljjz+u9f6VK1fiww8/
xIYNG3Dw4EG0bNkSI0aMQFlZmSZm8uTJOH36NHbu3IlffvkFe/
bswYwZM8z1EiymoX0HACNHjpS8F7/99lvJ/
fa473bv3o2XXnoJBw4cwM6dO1FZWYnhw4ejpKREE9PQ36lSqcTDDz+MiooK7N+/H19+
+SU2bdqEt956yxIvyvWz02XcA8Pzzz0vedytXrtTcZ6/7DgCCg4OxYsUKHD16FEeOHMGDDz6IsWPH4vTp
0wCs8H0niExk8eLFomfPnrXel5+fLxwdHcXWrVs1bWfPnhUARFJSkpl6aL0AiO+//
15zW6VSiYCAALFq1SpNW35+vnBychLffvutNyGQykZmZaba+W1rNfSeEE
FOmTBFjx46t8zHcd2o5O5OTkCgNi9e7cQQr+/019//
VU4ODiI7OxsTcwnn3wiPDw8RHl5uXlfgAXV3HdCCDF48GAxZ86cOh/
DfSfVqlUr8e9//9sq33c8c0QmdfHiRiQQFBaF9+/
aYPHky0tPTAQBHjx5FZWUlhg0bponp3LkzQlUFBQAAb29v
APr9nSYlJaFHjx7w9/fXxIwYMaQKFhYWaswD2oOa+q/bNN9/Ax8cH3bt3x+uvv47S0lLNfdx3akqlEv/
9739RUlKCqKgoq3zfceFZMpl+/
fph06ZN6ZN6Ny5M7KysrB06VlVIMHDgQp06dQnZ2NhQKBby8vCSP8ff3R3Z2tmU6bIWq90fN8H0bpxQ3h0H17er7srOz4
efnJ7m/RYsW8Pb2tvt9Y9OnLkSMTExCAsLAyXL1/
G3//+d4waNQpJSUmQy+XcdwBUKhXmzp2LBx54ANwdwcAvf5Os7Oza31fVt9nD2rbdwAwadIktGvXDkF
BQUhJScFrr72G8+fPIy4uDgD33cmTJxEVFYWysjK4ubnh+++/
R0REBJKTk63ufcfkiExm1KhRRmv9HRkaiX79+aNeuHbZs2QIXFxcL9ozszYQJEzT/
79GjByIxEeHo6EhAQMHTrUgj2zHi+99BJOnTolGRdI+qlr32mPWevRowcCAwMxdOhQXL58GeHh4ebu
ptXp3Lk
zkpOTUVBQgG3btmHKlCnYvXu3pbtVK15Woybj5eWFTp064dKlSwgICEBFRQXy8/Ml29y8cQMBAQGW6aAVqt4fNc
MlMTdu3EBAQIBlOmjFqvdJzdka2vsrICAAOTk5kvurqqqQl5fHf/VpD+/bt4ePjg0uXLgHgvps1axZ+
+eUXxMfHIzg4WNOuz99penQEBAre/
L6vuau7r2XW369esHAJL3nT3vO4VCgQ4dkDqB3795Yvnw5evbsibsibVr11rl+47JETWZ4uJiXL58GYGBgej/
+ESqXSfCiT2rVr15Cbm4vAwEAA9rvvvhBCYNWsWvv/+e/z5558ICwuT3K/
P32lUVBROnjwpSR537tyJiIgIzbf2Zv/VTX3R1X/
Hnn3+KI0eOiKioKBEVFWXhXltOUVGREB8uDh+/
LgAIFavXi2OHz8url69KoQQYsWKFcLLLy0v8+OOPIiUlRYwdO1aEhYWJJO3fuaLYxcuRIcc8994iDBw+Kv
Xv3io4dO4qJEyda6iWZTX37rqioSCxYsEAkJSWJ1NRU8ccff4h7771YdO1aEhYWJO3fuaLYxcuRIcc8994iDBw+Kv
Xv3io4dO4qJEyda6iWZTX37rqioSCxYsEAkJSWJ1NRU8ccff4h7771XdOzYUZSVlWm2YY/
77sUXXxSenp4iISFBZGVlaX5KS0s1MQ39nVZVVYnu3buL4cOHi+TkZLFjxw7h6+srXn/
9dUu8JLNpaN9dunRJLFu2TBw5ckSkpqaKH3/8UbRv314MGjRIsw173XdCCLFo0SKxe/
dukZqaKlJSUsSiRYuETCYTv//+uxDC+t53TI7IZMaPHy8CAwOFQqEdq0EePHjxeXLl3S3H/
nzh0xc+ZM0apVK+Hq6ioee+wxkZWVcEeW1Z8fLwAoPMzZcoUIYR6Ov+bb74p/P39hZOTkxg6dKg4f/
68ZBu5ubli4sSJws3NTXh4eIhp06aJoqIiC7wa86pv35WWlorhw4cLX19f4ejoKNq1ayemTp2qk0DMhBCmPx9FRE
REZJs45oiIiIhIC5MjIiIiIi1MjoiIiIi0MDkiIiIi0sLkiIiIiEgLkyMiIiIiLUyOiIiIiLQwOSIiMt
CmTZt0VhAnouaDyRERWa2pU6dCJpNBJpPB0dER/v7+eOihh/
DFF19ApVKZpQ+hoaFYs2aNpG38+PG4cOGCWZ6fiMyPyRERWbWRI0ciKysLaWlp+O233zBkyBDMmTMHjz
zyCKqqqozaphACVVVVRj3WxcUFfn5+Rj2WiKwfkyMismpOTk4ICAhAmzZtcO+99+Lvf/87fvzxR/z222/Y
tGkT0tLSIJPJkJycrHlMfn4+ZDIZEhISAAAJCEAACSXMgXASABKSAACGMlEABGAAAAMbAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
zyCKqqqozaphDC6McCgIuLC/z8/Ix+PBFZNyZHRGTVnJycEBAQgDZt2uDee+//F3//+d//F3//+d/
z444/47bffsGnTJqSlpUEmkyE5OVnzmPz8fMhkMk2nw22+/
oXfv3nBycsLevXtx+fJljB07Fv7+/nBzc8N9992HP/74Q/Me6OhoXL16FfPmzdNUrYDat9U+
74Q7Od6OhoXL16FfPmzdOcwQJqv6z2ySefIDw8HAqFAn/5y1/w1VdfSbYtk8nw73//G4899hhcXV3RsWNH/PTTT5r7b9++jcmTJ8PX1xcuLi7o2LEjNm7c2KTXjojo7jG5IiKbM3ToUPTs2RNxcXEGP+7ll1/GpKSk4Mknn0R6ejpmzZrVKZNVq1Zh2rRp2LBhQ+b7TphtF8iL8vKKLHRZ8IIL1OXe+/
NHj8auXbA4HI7FYuFBQU4vPPPwZAIH90Pj97/95j7E9Kyx+AMgCfMVbuwtdbdiNdOAAhAuzh3eOKBNZRBxCKMN1MVlFGEsf3qi7OMmODh7Kxl8+/4XUvvpK56bwqcHrd///
NHj8auXbtw/PhxjBw5Em5ubq6u7mjRogWKi4tx6dIldOvWDT4+PgY9Ljs7G2VlZWjRooX6Di2bBmysrKQlKQkfPvssQOLw8PzQBEKSbNsdHs3cEoNEr6vrKQ1ZVV63a///

57zJkzB6+88gpOnTqFv/
3tb5g2bRri4+MlcUuXLsWTTz6JlJQUjB49GpMnT0ZeXp5xO4GImgyTIyKySV26dEFaWppBj1m2bBkeeu
ghhIeHw9vbGz179sTf/vY3dO/
eHR07dsTbb7+N8PBwzRkdb29vyOVyuLu7IyAgAAEBAbVu97333sPUqVMxc+ZMdOrUCfPnz0dMTAzee+8
9SdzUqVMxceJEdOjQAe+88w6Ki4tx6NAho14/
ETUdJkdEZJOEEJrLXPrq06eP5HZxcTEWiArl27wsvLC25ubjh79qzmzJG+zp49iwceeEDS9sADD+Ds
2bOStsjISM3/W7ZsCQ8PD+Tk5Bj0XETUgNERMY4e/YswsLC4OCg/
o4nhNDcV1lZWetjWrZsKbm9YMEC7Ny5E+
+99x46dOgAFxcXXMbGoqKiokn8nMNuuOiPTHM0dEZHP+/PNPnDx5Eo8//
jh8fX0BQDIeSHtwdn327duHqVOn4rHHHkOPHHkOPHj0QEBCgc6lOoVBAqVTWu52uXbti3759OtuOiIjQqx9EZ
F145oiIrFp5eTmys7OhVCx48YN7NixA8uXL8cjjzyCZ555BnK5HP3798eKFSsQFhaGnJwc/OMf/
9Br2x07dkRcXBzGjBkDmUyaksJPj4+OttZuHAhnnzySdxxzzz0YNmwYf
v75Z8TFxUlmvhGR7eCZIyKyajt27ajt27EBgYCBCQ0MxcuRIxMfH48MPPP8SPP/
4IuVwOAPjiiy9QVVWF3r17Y+7cufjnP/
+p17ZXr16NVq1a4f7778eYMWwYsQI3HvvvvZKYZcuWIS0tDeh4ZqzVDWNGzcOa9euxXvvvvYdu3brh00
8/
xcaNGxEdHd2o105ElieT2hfqiYiIiOwcyxwRERERaWFyRERERaWFyRERERKSFyRERERGRFiZHRERERFqYHBERERFp
YXJERERpIXJEREREZEWJkdEREREWpgcEREREWlhckRERESkhckRERERkRYmR0RERERa/
j9SyvyKcCZoowAAAABJRU5ErkJggg==\n"
          },
          "metadata": {}
        },
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Valor p de Shapiro-Wilk: 0.0000\n",
            "Valor p de Breusch-Pagan: 0.0000\n"
          ]
        }
      ]
    }
  ]
}