

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "id": "755c232d",
      "metadata": {},
      "source": [
        "# Assignment 4"
      ]
    },
    {
      "cell_type": "markdown",
      "id": "3ae24645",
      "metadata": {},
      "source": [
        "# 1.Download the dataset"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 1,
      "id": "1b9fe6dd",
      "metadata": {},
      "outputs": [],
      "source": [
        "import pandas as pd\n",
        "import numpy as np\n",
        "import matplotlib.pyplot as plt\n",
        "import seaborn as sns"
      ]
    },
    {
      "cell_type": "markdown",
      "id": "4253df83",
      "metadata": {},
      "source": [
        "# 2. Load the dataset"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "id": "23f5edf3",
      "metadata": {},
      "outputs": [],
      "source": [
        "data = pd.read_csv(r\"Mall_Customers.csv\")"
      ]
    },
    {
      "cell_type": "code",

```

```
"execution_count": 3,
"id": "f8a79274",
"metadata": {},
"outputs": [],
"source": [
  "data.head();"
]
},
{
  "cell_type": "markdown",
  "id": "e57b2d3d",
  "metadata": {},
  "source": [
    "# 3. Perform Below Visualizations."
  ]
},
{
  "cell_type": "markdown",
  "id": "8dc94c33",
  "metadata": {},
  "source": [
    "### · Univariate Analysis"
  ]
},
{
  "cell_type": "code",
  "execution_count": 4,
  "id": "16dee427",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "(array([31., 19., 34., 29., 16., 26., 15., 10.,  6., 14.]),\n",
          " array([18. , 23.2, 28.4, 33.6, 38.8, 44. , 49.2, 54.4, 59.6, 64.8, 70. ]),\n",
          " <BarContainer object of 10 artists>)"
        ]
      },
      "execution_count": 4,
      "metadata": {},
      "output_type": "execute_result"
    },
    {
      "data": {
        "image/png": "iVBORw0KGgoAAAANSUhEUgAAAXAAAAD4CAYAAAD1jb0+AAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcNnpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfk9AAAACXBIWXMAAAsTAAALEwEAmpwYAAAOQ0IEQVR4nO3dYYhl9XnH8e8vampiLK51lCG6nSaITZA6psPWYgmJJmGNleoLIULTpUjXFwoKlrL1TUyhYCEm7YsirNFmaU2KVYOiNs2yVawlmM7aja5dxZBuE5Pt7qRpqrZgUZ++uGebYZzx3pl770z+d74fuJxz/vfcOc/D7P4487/n3JuqQpLUnndsAGSpLUxwCWpUQa4JDXKAJekRhngktSoE9fzYGeccUbNzMys5yElqXn79+//cVVNLR3vG+BJTgaeAH6h2/++qvpckluB3wMWul1vqapH3+5nzcMMD8/v9raJWITS/Jvy40Pcgb+GnBJVb2a5CTgySR/2z33par6wqiKlCQNrm+AV+9On1e7zZO6h3f/SNIGG+hNzCQnJDkAHAP2VtVT3VM3JHkmyd1JtoyrSEnSWw0U4FX1RlXNAmcD25KcD9wBvB+YBY4Aty/32iQ7k8wnmV9YWFhuF0nSGqzqMsKq+inwOLC9qo52wf4mcCewbYXX7K6quaqam5p6y5uokqQ16hvgSaaSnNatvwv4GPB8kulFu10FHBxLhZKkZQ1yFco0sCfJCfQC/96qeJJXyaZpfeG5mHgurfVKU16i0GuQnkGuHCZ8c+OpSJJ0kC8IV6SGrWut9JrdWZ2PbIhxz182+UbclxJq+MZuCQ1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNcoAl6RGG
```

eCS1CgDXJlaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An  
OTnJt5N8J8lzST7fjZ+eZG+SF7vllvGXXK0k6bpAz8NeAS6rqAmAW2J7kImAXsK+qzgX2dduSpHXS8Cr59Vu86  
TuUcAVwJ5ufA9w5TgKlCQtb6A58CQnJDkAHAP2VtVTwFIVdQSgW565wmt3JplPMr+wsDCisiVJAwwV4Vb1R  
VbPA2cC2JOCpeoCq21lVc1U1NzU1teYyJUILLreoqlKr6KfA4sB04mmQaoFseG3VxkqSVDXIVylSS07r1dwEfA54  
HHgJ2dLvtAB4cU42SpGUMcifmNLAnyQn0Av/eqno4ybeAe5NcC3wfuHqMdUqSlugb4FX1DHDhMuP/AVw6jqI  
kSf0181koWj9+7ozUBm+ll6RGGeCS1CgDXJlaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An  
lySGmWAS1KjDHBjapQBLkmNMsAlqVEGuCQ1ygCXpEb5jTwSfguR2uQZuCQ1ygCXpEb1DfAk5yR5LMmh  
JM8lubEbvzXJD5Mc6B6fHH+5kqTjBpkDfx24uaqeTnIqsD/J3u65L1XVF8ZXniRpJX0DvKqOAEe69VeSHALeO  
+7CJElvb1Vz4ElmgAuBp7qhG5I8k+TuJFtGXZwkaWUDB3iS9wD3AzdV1cvAHcD7gVl6Z+i3r/C6nUnmk8wvL  
CwMX7EkCRgwwJOcRC+876mqBwCq6mhVvVFVbwJ3AtuWe21V7a6quaqam5qaGlXdkrTpDXIVSoC7gENV9c  
VF49OLdrsKODj68iRJKxnkKpSLgc8CzyY50I3dAlYTZBYo4DBw3RjqkyStYJCrUJ4EssxTj46+HEnSoLwTU5IaZ  
YBLUqMMcElqlAEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An  
Q1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrGkNapvgCc5J8ljSQ4leS7Jd346Un2JnmXW24Zf  
7mSpOMGOQN/Hbi5qj4AXARcn+SDwC5gX1WdC+zrtiVJ66RvgFfVkap6ult/BTgEvBe4AtjT7bYHuHJMNUqSlr  
GqOfAkM8CFwFPAWV1BHohD5y5wmt2JplPMr+wsDBkuZKk4wYO8CTvAe4Hbqqqlwd9XVXtrqq5qpqbmp  
a542SpGUMFOBJTqIX3vdU1QPd8NEk093z08Cx8ZQoSvRoiFehBLgLOFRVX1z01EPajm59B/Dg6MuTJK3kx  
AH2uRj4LPBskgPd2C3AbcC9Sa4Fvg9cPZYKJUUnL6hvgVfUkkBWevnS05UiSBjXIGfimN7PrkY0uQZLewlvJal  
RBrGkNcoAl6RGGeCS1CgDXJlaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An  
KjDHBjapQBLkmNMsAlqVF+I49+rvjtR9LgPAOXpEYZ4JLUqL4BnuTuJMeSHFw0dmuSHyY50D0+Od4yJUILL  
DXIG/hVg+zLjX6qq2e7x6GjLkiT10zfAq+oJ4CfrUiskaRWGuQrIhiS/A8wDN1fvfy63U5Kdwe6ArVu3DnE4aTJt  
1JU3h2+7fEOOq9FZ65uYdwDvB2aBI8DtK+1YVburaq6q5qamptZ4OEnSUmS8K8Ko6WlVvVNWbwJ3AttGWJUn  
qZ00BnmR60eZVwMGV9pUkjUffOfAkXwM+ApyR5CXgc8BHkswCBRwGrhtfiZKk5fQN8Kq6Zpnhu8ZQiyRpF  
bwTU5Ia5YdZSdo0NvLD0sZx2aZn4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrGkNcoAl6RGGeCS1CgDX  
JlaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An  
yd4kL3bLLeMtU5K01CBn4F8Bti8Z2wXsq6pZgX3dtiRpHfUN8Kp6AvjJkuErgD3d+h7gytGWJUnq58Q1vu6sqjoC  
UFVHkpy50o5JdgI7AbZu3brGw0katZldj2zYsQ/fdvmGHXuSjP1NzKraXVVzVTU3NTU17sNJ0qaxlgA/mmQaoF  
seG11JkqRBrDXAHwJ2dOs7gAdHU44kaVCDXeb4NeBbwHlJXkpyLXAb8PEkLwIf77YlSeuo75uYVXXNck9d  
OuJaJEmr4J2YktQoA1ySGmWAS1KjDHBjapQBLkmNMsAlqVEGuCQ1ygCXpEYZ4JLUKANckhplgEtSowxwS  
WqUAS5JjTLAJalRBrGkNcoAl6RGGeCS1CgDXJla1fcr1SRp1GZ2PbLRJUwEz8AlqVEGuCQ1aqgplCSHgVeAN  
4DXq2puFEVJkvobxRz4R6vqxyP4OZKkVXAKRZlaNWYAF/DNJPUt7FxyuhyQ7k8wnmV9YWBjycJKk44YN8lu  
r6kPAZcD1ST68dleq21lVc1U1NzU1NeThJEnHDRXgVfWjbnkM+DqwbRRFSZL6W3OAjzklyanH14FPAAdHV  
Zgk6e0NcxXKWcDXkxz/OV+tmq+MpCpJU19rDvCq+h5wwQhrkSStgpcRSIKjDHBjapQBLkmNMsAlqVEGuCQ  
1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrGkNcoAl6RGGeCS1CgDXJlaZYBLUqMMcElql  
AEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An  
G/DdqypeVf0v8NfAFaMps5LUz4lDvPa9wA8Wbb8E/MbSnZLSBH22m68meWGIY67GGcCP1+IYG2mz9Ambp  
9fN0idsol7zJ0P1+svLDQ4T4FlmrN4yULUb2D3EcdYkyXxVza33cdfbZukTNk+vm6VPsNdhDToF8hJwzqLts4Ef  
DVeOJGIQwwT4PwHnJmVJO8EPgM8NJqyJEn9rHkKpapeT3ID8HfACcDdVfXcyCob3rpP22yQzdInbJ5eN0ufY  
K9DSdVbqp0lSQ3wTkxJapQBLkmNaj7Ak5yT5LEkh5I8l+TGbvz0JHuTvNggt2x0rcNKcnKSbyf5Ttfr57vxiesVenf  
7JvnnJA9325Pa5+EkzyY5kGS+G5u4XpOcluS+JM93/19/c0L7PK/7XR5/vJzkpnH02nyAA68DN1fVB4CLgOu7W/  
p3Afuq6lXgX7fduteAS6rqAmAW2J7kliazV4AbgUOLtie1T4CPvtXsouuEJ7HXPwO+UVW/ClxA73c7cX1W1Qvd  
73IW+HXgf4CvM45eq2qiHsCDwMeBF4DpbmwaeGGjaxtxn+8GnqZ39+vE9UrvvoJ9wCXAaw93YxPXZ9XIYOG  
PJ2ET1Cvwi8K90F05Map/L9P0J4B/H1esknIH/vyQzwIXAU8BZVXUEoFueuYGljUw3rXAAOAbsrapJ7fVPgT8  
A3lw0Nol9Qu8O5m8m2d999ARMXq/vAxaAv+imxb6c5BQmr8+IPgN8rVsfea8TE+BJ3gPcD9xUVS9vdD3jUIVv  
VO9Ps7OBbUnO3+CSRI7Jp4BjVbV/o2tZJxdX1YfofbLn9Uk+vNEFjcGJwleAO6rqQuC/mYDpkrfT3eD4aeBvxn  
WMiQjwJcFRC+97quqBbvHokunu+Wl6Z6wTo6p+CjwObGfyer0Y+HSSw/Q+5fKSJH/F5PUJQFX9qFseozdXuo3  
J6/UI4KXuL0aA++gF+qT1udhlwNNVdbTbHnmvzQd4kgB3AYeq6ouLnnoI2NGt76A3N960JFNJTuvW3wV8DHi  
eCeulqv6wqs6uqhl6f4L+fVX9NhPWJ0CSU5Kcenyd3pZpQsas16r6d+AHSc7rhi4F/oUJ63OJa/jZ9AmModfm78R  
M8lvAPwDP8rP50lvozYPfC2wFvg9cXVU/2ZAiRyTJrwF76H10wTuAe6vqj5L8EhPW63FJPgL8flV9ahL7TPI+e  
mfd0Jtm+GpV/fGE9joLfbI4J/A94Hfp/h0zQX0CJHk3vY/bfI9V/Vc3NvLfafMBLkmbVfNTKJK0WRngktQoA1yS  
GmWAS1KjDHBjapQBLkmNMsAlqVH/ByG7xWFHCR5SAAAAAEIFTkSuQmCC",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

```
"needs_background": "light"
},
"output_type": "display_data"
},
"source": [
  "plt.hist(data['Age'])"
],
{
  "cell_type": "markdown",
  "id": "8a08b8d5",
  "metadata": {},
  "source": [
    "### · Bi- Variate Analysis"
  ]
},
{
  "cell_type": "code",
  "execution_count": 5,
  "id": "788a5b44",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "<matplotlib.collections.PathCollection at 0x233dba0ed90>"
        ]
      },
      "execution_count": 5,
      "metadata": {},
      "output_type": "execute_result"
    },
    {
      "data": {
        "image/png": "iVBORw0KGgoAAAANSUhEUgAAAXoAAAD4CAYAAADiry33AAAAOXRFWHRTb2Z0dD
2FyZQBhbnV4RwB90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAACX
BIWXMAAAAsTAAALEwEAmpwYAAAVT0IEQVR4nO3dfYxU133G8efxsLEX5wVcFgvjdXArhGOBNCQbY8d
S5MihWEIqNmmiguKKNJZQJSsvTUNiCoqjBsduiZJWldIWJW5QSddyEmeNUicE4bhWX1hnCbbXGFPcugEWA
psXktTe8LL8+sdeYFjPsDN7Z2bnzH4/Epq5v3vv3CP+ePbOmXPPcUQIANC6LprsBgAA6ougB4AWR9ADQIsj6
AGgxRH0ANDipk12AyRp1qxZMW/evMluBgAkZefOnT+NiI7xjmuKoJ83b576+/snuxkAkBTbP67kuHG7bmw/Y
Puo7WdL7Puk7bA9q6i2xvYlTvfaXlpdswEAtVZJH/3XJN02tmi7U9ISSfuLatdKW7i7puuycL9su1KSlAIAJGTfol+I
JST8vsetLkj4lqfjR2mWSHoyI4xHxoqQXJN1Qi4YCACZmQqNubN8uaTAinh6za66kA0XbB7Naqc9YZbvfdv/Q0
NBEmgEAqEDVQW97uqS1kj5TaneJWsnJdCJiY0R0RURXR8e4PxoDACZoIqNufkfS1ZKet1JV0r6ke0bNH0H31
l07JWSDuVtZDnregfU03dAIXEq2FqxuFPruxfW63IAkKSq7+gjYiAiZkfEvliYp9Fwf3NE/ETSFknLbV9s+2pJ8yU
9WdMWZ9b1Dmjzjv0ayWbfHInQ5h37ta53oB6XA4BkVTK8skfSf0paYPug7TvLHRsRuyU9JOk5Sd+TdFdEjNSq
scV6+g5UVQeAqWrcrpUWDHO/nljtu+VdG++Zo1vpMw8+uXqADBvJTvXTcGlFvctXweAqSrZoF+XuLOqOgB
MVU0x181EnBldw6gbALgwN8OasV1dXcGkZgBQHds7I6JrvOOS7boBAFSGoAeAFkfQA0CLI+gBoMUR9AD
Q4gh6AGhxBD0AtDiCHgBaHEEPAC2OoAeAFpfsXDeS1LtrUBu27tWhY8O6Yka7Vi9doO5FJZeoBYCm0sj8Sjb
oe3cNas3Daxo+ObquyeCxYa15eHR1KcIeQDNrdH4l23WzYeves/9JZwyfHNGGrXsnqUUAUJIG51eyQX/o2HBV
dQBoFo3Or2SD/ooZ7VXVAaBZNDq/kg361UsXqL2tcF6tva2g1UsXTFKLAKAyjc6vZH+MPfODBaNuAKSm0fn
FCIMAKChWmAIASCLoAaDljdtHb/sBSe+RdDQirs9qGyT9vqQTKv5b0h9HxLFs3xpJd0oakfTRiNhan6YDQLrW
9Q6op++ARiJUsLVicafWdy+sy7UquaP/mqTbxtS2Sbo+It4o6b8krZEK29dKW7i7puuycL9suCABw1rreAW3esV8j2
W+kIxHavGO/1vUO1OV64wZ9RDwh6edjat+PiFPZ5g5JV2bv10l6MCKOR8SLkl6QdEMN2wsAyevpO1BVPa9a9
NF/WNJ3s/dzJRW39GBWewXbq2z32+4fGhqqQTMAIA0jZUY7lqvnIsvoba+VdErS18+UShxWsuURsTEiuiKiq6
```

OjI08zACApBZeKyvL1vCYc9LZXavRH2g/GucH4ByV1Fh12paRDE28eALSeFYs7q6rnNaGgt32bpE9Luj0iXi7at  
UXSctsX275a0nxJT+ZvJgC0jvXdC3XHjVedvYMv2LrjxqvqNupm3CdjbfldukXSLElHJN2j0VE2F0v6WXbYjoj4k  
+z4tRrtz8l6eMR8d2xnxzkWT8YCQPUqfTKWKRAAIFFMgQAaKETQA0DLS3aaYonFwQGki8XBK8Di4ABSxe  
LgFWJxcACpYnHwCrE4OIBUsTh4hVgcHECqWBy8QiwODiBVLA5eIRYHB5AqFgcHAFSEJ2MBAJIegBoeQ  
Q9ALQ4gh4AWhxBDwAtjqAHgBZH0ANAIyPoAaDFJftKLACkbMkXH9e+oy+d3Z4/+1Jt+8QtdbkWd/QA0GBj  
Q16S9h19SUu++HhdrkfQA0CDjQ358ep5EfQA0OKS7qNnzVgAGN+4d/S2H7B91PazRbXLbG+zvS97nVm0b43t  
F2zvTB20Xg0/s+bi4LFhhc6tudi7a7BelwSAmrj8Na+qpp5XJV03X5N025ja3ZK2R8R8Sduzbdm+VtJySddl53zZdkF  
1wJqxAFI1rVA6FsvV8xo36CPiCuk/H1NeJmlT9n6TpO6i+oMRcTwiXpT0gqQbatPU87FmLIBUpbJm7OURcViS  
stfZXW2upANFxx3Maq9ge5Xtftv9Q0NDVTdgxvS2quoA0CwanV+1HnXjErWSS1hFxMa16lqIro6OjqovdHxMt8  
14dQBoFo3Or4kG/RHbcyQpez2a1Q9K6iw67kpJhybevPJePnm6qjoANItG59dEg36LpJXZ+5WSHimqL7d9se2rJc2  
X9GS+JgIA8hh3HL3tHkm3SJpl+6CkeyTdL+kh23dK2i/pA5IUEbttPyTpOUmnJN0VEXX5LjKjvU3Hhk+WAMA  
zhk36CNiRZldt5Y5/15J9+ZpVCU+e/t1Wv2Np3Xy9LmfANousj57+3X1vjQAJCXZJ2PPPAHLk7EAUIOWNRKvH  
KdScKnxLPklG/TSaNgt7ABSs2Jxpzbv2F+yXg9JBz0ApGh990JJUk/fAY1EqGBrxeLOs/Vac5T4+tBoXV1d0d/fX  
/V563oHGvYfBQDNxvbOiOga77hk7+jX9Q6c99VnJOLsNmEPAOckOx99T9+BquoAMFUIG/SlfrG+UB0Apqpkg  
77cMKR6DU8CgFQlG/TlhiHVa3gSAKQq2R9jGz08CQBSltWsgCYylp+eKXEOHoA6WpkfiUb9IyjB5CqRudXs  
j/Glpon4kJ1AGgW/9xXOqfK1fNKNugBIFWny/w0Wq6F0EPAC2OoAeAFkfQA0CLI+gBoMUIG/TlfgFncXAAz  
a7R+ZVs0H/29uvUdtH5E5ixODiAFDQ6v5J9YIrFwQGkqTH5lewdPQCgMrmC3vaf2t5t+1nbPbYvsX2Z7W2292W  
vM2vV2GK9uwa15uEBDR4bVkgapDasNQ8PqHfXYD0uBwA10+j8mnDQ254r6aOSuiLiekkfScsl3S1pe0TMl7Q  
92665DVv3avjkyHm14ZMj2rB1bz0uBwA10+j8ytt1M01Su+1pkqZLOiRpmaRN2f5NkrpzXqOkQ8eGq6oDQLNod  
H5NOOgjYIDSfyTtl3RY0i8j4vuSLo+Iw9kxhyXNrkVDx7piRntVdQBoFtNfVaiqnleerpuZGr17v1rSFZlutX1HFee  
vst1vu39oakjq669euqCqOga0i5dPjFRVzytP1807Jb0YEUMRcVLSw5LeJum17TmSIL0eLXVyRGyMiK6l6Oro6K  
j64vc9+lxVdQBoFuUmqazXen95gn6/pBttT7dtSbdK2iNpi6SV2TerJT2Sr4mIHfn1iarqADBVTfiBqYjos/1NST+Sd  
ErSLkkbJb1a0kO279ToH4MP1KKhAICJyfVkbETcl+meMeXjGr27BwA0AZ6MBYAWl2zQz599aVV1AJiqkg36n  
/5f6R9dy9UBYKpKNuh/8fLJquoAMFUIG/QAgMoQ9ADQ4gh6AGhxBD0AtDiCHgBaHEEPAA3mKut5EfQA0G  
ApzV4JAEhAskHf6K8+AJCqZIOepQQBpKq9rXT0lqvnlWzQv+Oa0qtSlasDQLO4cmbpG9Jy9bySDfofPF96ndlyd  
QBoFvuOvIRVPA9kg/7QseGq6gAwVSUB9PTRA0Blkg361UsXqK1w/hibtoK1eumCSWoRADSnZINekk6OxAW3  
AQAJB/0nv/F0VXUAmKqSDfpTp0vfvZerA8BUlWzQAwAqQ9ADQIvLffs2Z9j+pu3nbe+xfZpty2xvs70ve51Zq8  
YCAKqX947+byR9LyKufS7kvZlulvS9oiYL2l7tg0AmCQTDnrbr5X0dklflaIOBERxyQt7QpO2yTpO58TQQA  
5JHnjv63JQ1J+kfbu2x/xfalki6PiMOSIL3OLnWy7VW2+233Dw0xPw0A1EueoJ8m6c2S/i4iFkl6SVV000TExojoioi  
ujg5mnASAeskT9AclHYyIvmz7mxoN/iO250hS9no0XxMBAHIMOogj4ieSDtg+M7nMrZKek7RF0sqstlLSI7laW  
MbcMpOXlasDwFSVd9TNRyR93fYzkt4k6fOS7pe0xPY+SUuy7ZpbvXSB2tsK59Xa2wpMagYAY0zLc3JEPCWp  
q8SuW/N8biW6F82VJG3YuleHjg3rihntWr10wdk6AGBUrqCfbN2L5hLsAJJTsDUSr5yXq2CXODq/pIP+mrWP6j  
dFUxNfUrCev/ddk9giABjfsWd2rxjf8l6PSQ7183YkJek34yErlN76CS1CAAqs233T6qq55Vs0I8N+fHqANAsjvz6R  
FX1vJINegBAZQh6AGhxyQb9JYXSv06XqwnAs7j8Na+qpp5XskH//L3vekWoM+oGQAr61i4pmV99a5fU5XpJD  
68k1AGkaF3vQMlRg+t6B7S+e2HNr5fsHb0k9e4a1M33P6ar7/4X3Xz/Y+rdNTjZTQKAcfX0Haiqnleyd/S9uwbliY  
ee0unsj+LgsWF94qGnJlmmZQE0tVJPxV6onleyd/Sf/tYzZ0P+jNMxWgcAnJNs0B8/dbqqOgBMVckGPQCgMgQ9  
ALS4ZIP+tRcXqqoDwFSVbND/6vhIVXUAmKqSDXoAQGUlegBosJnT26qq55Vs0LeVaXm5OgA0i3LPRdXpeal  
0g/5kmeHy5eoA0CyODZ+sqp5XskEPAKgmQQ8ALS530Nsu2N5l+zvZ9mW2t9nel73OzN9MAMBE1eKO/mOS9  
hRt3y1pe0TMl7Q926659jK/uparA8BUlSsVbV8p6d2SvlJUXiZpU/Z+k6TuPNco5773vbGqOgBMVXlvf/9a0qckFY  
91uTwIDktS9jq71Im2V9nut90/NDQ0oYu3XeQLbgMAcgS97fdIOhoROydyfkRsjliuiOjq6Oio+vwNW/fq5JgJ6U+e  
Dm3YuncizQGA1pVnhambJd1u+12SLpH0WtubJR2xPSciDtuel+loLRo61uCx4arqADBVTfiOPiLWRMSVETFP0  
nJJj0XEHZK2SFqZHbZS0iO5W1lCwaW7acrVAWCqqscQlfsllbG9T9KSbLvmGr3mlgCkqiaLg0fE45Iez97/TNKt  
tfjcC5k7o71kN83cGe31vjQAJCXZQeerly5Qe9v5i4y0txW0eumCSWoRAFSmXAdzvTqekw367kVzdd/7FmrujHZ  
Zo3fy971voboXzZ3spgHABX3wxququddVk66bydK9aC7BDgDjSDro1/UoQkfvgEYiVLC1YnGn1ncvnOxmAcA  
F9fQdKFuvR4YlG/Trege0ecf+s9sJEWec3XsAzazRowaT7aO/0F9EAMA5yQY94+gBoDLJBj1PvgJlVaPzK9mgX  
7G4s6o6ADSLRudXskHf9frLXtH4i7I6ADSzcjlv/xKNug3bN173iT40uik+ExTDKdZr32QFX1vJIN+kNlpiMuV  
weAZvHSiZGq6nklG/TTX1Woqg4AU1WyQf9ymb985eoAMFUIG/TlRsszih4Azpds0DOOHkCqGEdfcbRA0gV4  
+gr1OhxqABQK41+DijZoG/00FQAqJVGpweUBNA3ehwqANRko58DSjboASBVM6a3VVXPI6AHgAYrN5t6v  
WZZJ+gBoMF+OXyyqnpeyQb9zDJfcerVAaBZJNN1Y7vT9g9s77G92/bHsvpltrfZ3pe9zqxdc89p9FcfAKiVILpuTk  
n6s4h4g6QbJd1l+1pJd0vaHhHzJW3PtmvuWJmvOOXqANAsGp1fEw76iDgcET/K3v9a0h5JcyUtk7QpO2yTpO6c  
bQQA5FCTPnrb8yQtktQn6fKIOCyN/jGQNLvMOats99vuHxoaqkUzAAA15A5626+W9C1JH4+IX1V6XkRsjliuiO  
jq6OjI2wwAQBM5gt52m0ZD/usR8XBWPmJ7TrZ/jqSj+ZoIAMgiz6gbS/qppD0R8cWiXVskrezer5T0yMSbV94dN  
15VVR0ApqppOc69WdIfSRqw/VRW+3NJ90t6yPadkvZL+kCuFpaxvnuhJKmn74BGIlSwtWJx59k6AGDUhIM+I  
v5NurlZ8m+d6OcCAGorzx39pFrXO6DNO/af3R6JOLvNXT0AnJPsfAg9fQeqqgPAVJVVs0I+UeVa4XB0AmkWj  
B5Mk23VTsEuGOouDA2h2jR5MkmzQr1jceV4ffXEdAJrd+u6FDfs9MdmgZ3glAFTG0QR921ldXdhf3z/ZzQCAp

NjeGRFd4x2X7I+xAIDKEPQA0OIlegBocQQ9ALQ4gh4AWlxTjLqxPSTpxzk+Ypakn9aoOQDQSHny6/URMe7K  
TU0R9HnZ7q9kiBEANJtG5BddNwDQ4gh6AGhxrRL0Gye7AQAwQXXPr5boowcAlNcqD/QAgDIlegBocU0b9Lb  
D9j8VbU+zPWT7O+Ocd8t4xwBALdgesf1U0b95dbzW/9qeNZFzm3k++pckXW+7PSKGJS2RNDjJbQKAYsMR8  
abJbsR4mvaOPvNdSe/O3q+Q1HNmh+0bbP+H7V3Z64KxJ9u+1PYDtn+YHbesQe0GMEXZfovtf7W90/ZW23Oy  
+uO2v2T7Cdt7bL/V9sO299leX3R+b3bubturyljzDttPZt8i/sF24UJtavagf1DSctuXSHqipL6ifc9LentELJL0GUmfL3  
H+WkmPRcRbJb1D0gbb19a5zQCmjvaibptv226T9LeS3h8Rb5H0gKR7i44/ERFvI/T3kh6RdJek6yV9yPZvZcd8OD  
u3S9JHi+qSJNtvkPSHkm7Ovk2MSPrghRrZzF03iohnsj6vFZleHbP7dZI22Z4vKSS1lfiI35N0u+1PZtuXSLpK0p76t  
BjAFHNe143t6zUa3NtsS1JB0uGi47dkrwOSdkfE4ey8/5HUKelnGg3392bHdUqan9XPuFXSWyT9MLtGu6SjF2pk  
Uwd9ZoukL0i6RVLxX7bPSfpBRLw3+2PweIlzLekPImJvndsIANJo5uyOiJvK7D+evZ4uen9me5rtWyS9U9JNEfG  
y7cc1eoM69hqbImJNpY1q9q4bafSrZ19ExMCY+ut07sfZD5U5d6ukjzj7s2d7UV1aCACj9krqsH2TJNlus31dFee/TtI  
vspC/RtKNJY7ZLun9tmdn17jM9usv9KFNH/QRcTAi/qbErr+SdJ/tf9fo16NSPqfRLp1nbD+bbQNAXUTECUnvl/S  
Xtp+W9JSkt1XxED/T6J39MxrNqx0lrVgcpHWSvp8dt03SnAt9KFMgAECLa/o7egBAPgQ9ALQ4gh4AWhxBDwA  
tjqAHgBZH0ANAiyPoAaDF/T+ohYG32Po1HgAAAABJRU5ErkJggg==",

```
"text/plain": [
  "<Figure size 432x288 with 1 Axes>"
],
"metadata": {
  "needs_background": "light"
},
"output_type": "display_data"
},
"source": [
  "plt.scatter(data['Gender'],data['Annual Income (k$)'])"
],
},
{
  "cell_type": "markdown",
  "id": "1aabf1d0",
  "metadata": {},
  "source": [
    "### · Multi-Variate Analysis"
  ]
},
{
  "cell_type": "code",
  "execution_count": 6,
  "id": "226e50da",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array([[<AxesSubplot:xlabel='Age', ylabel='Age'>,\n",
          "      <AxesSubplot:xlabel='Annual Income (k$)', ylabel='Age'>],\n",
          "      [<AxesSubplot:xlabel='Age', ylabel='Annual Income (k$)'>,\n",
          "      <AxesSubplot:xlabel='Annual Income (k$)', ylabel='Annual Income (k$)'>]],\n",
          "      dtype=object)"
        ]
      },
      "execution_count": 6,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
}
```

```
{
  "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAAmUAAAJSCAYAAACC8LIyAAAAOXRFWHRTb2Z0
d2FyZQBhYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfk9AAAAC
XBIWXMAAAAsTAAALEwEAmpwYAAC6wkIEQVR4nOzdd5ibV5n38e9RmdEUTE/FMx733ksc1/TeICQEgKE
UBbYpS4sLEtbXlgWlrosoYZAeu/FSRzbce+9jqf3XtSl8/4xM46TuEyR9Kjcn+uayzOy9Ognaywdnec+91Faa4QQQ
gghhLFMRgcQQgghhBAyKBNCCCCGEiAgyKBNCCCCGEiAAyKBNCCCCGEiAAyKBNCCCCGEiAAyKBNCCCCG
EiAAWowOMVE5Oji4vLzc6hhAixIRVVSgVLUKIYNu5c2eb1jp3ONeNukFZeXk5O3bsMDqGECLGLFy4UF5bh
BBBp5SqHu515fSIEEIIIUQECOtMmVLqKuAbgz9OAT4LTAJuBKqBu7XW3nBmEkIIYSIBGGdKdNav6y1Xq2
1Xg3UADuBNVrr5cA+4KZw5hFCCDF8Hf0e3D7/+y53+/x09HsMSCREbDHk9KVSqgJoBmYD6wYvXgssNSKP
EEKI89ta2c79m6p4YHM1Lu87AzOX188Dm6u5f1MVWyrBduwoRPQzqqbsFuApIAPoGbysG8g825WVUvcqpX
YopXa0traGJ6EQQojT6rucAPS6fPS43qky6XF56XX5AGgYvI4QYnSMGpRdDzwLdAFpg5elDf78Plrr+7TWC7X
WC3Nzh7WqVAghRBBdPDGHkswkFpZnkme3nb48z25jYXkmJZIJXDwx8CEQkS/sLfEUeOVAB6tdbtSajvwOe
C/gMuALeHOI4QQ4sLy02zcurD0rH+3YpJ8WBYiGIyYKbsReAZAa90CrFdKbQTmAk8bkEcIIcR7eHwBWnpca
K2NjiJE3Aj7TJnW+vfv+fknwE/CnUMIICZ+QOah7fX0N7nYUZRGI fMKDA6khBxIeo6+ovha+1lsflkOydb+3G4
fWSlJjCnJIMFZZnYrGaj4wkhlPThF6C9b6DFRVOPy+A0QsQPGZTFoIMN3fxy7XHWHM4moEEpsFnMOAeXs
WcmW7lzaRmfXjWBIEt5FRBCvFtSgplVU3I5ldrPovIso+MIETfkHTmGuLx+fvLyEe7fVEVakpXPrJrA1TMLm
VZox2I20evysulUB49sr+VXb5zg0R11/OxDe2TFIBDifeaPy2T+uLN2KRJChlgMymJEQ5eTz/59J3vruvno0nF87Y
qppCdb33Udu83KpdPpyXRaPjurO/nXJ/Zx55+28m/XTOOEFRUGJRdCRKselxevL0B2amJlJt/t8BLQmsyUhHNep
6XHRarNqnKcVJ2J6Ce/xTGgur2fO/6wlW6nl9/fuYArhlGUu6Ask2f+6WK++thefvjCYfrdfr546USUUMFILISIdi2
9Lh7ZVosvoLlmViFTCuxBPX5DI5PHd9YR0Jrr5xQxITf1fdZUtnO5pPtJCWYUxNpmZRjiKhNVPNYESRVbf3c
9vstODw+Hr536bAGZENSEi385o75fGB+Cf+z9hh/3HAqhEmFELGkvc+DLzDQLqOIN/iLAdr7PPgDGq2hpcd91
uu09A5c7vT4T+8qIEQ0k48VUayj38Ndf9mGxx/goXuXMrUg7ci3eg+zSfHTD87G6fXxo5cOU5qVzFUzZfm7EO
L8JuWl0ICSjssbCEnt2dRCO009Lnz+AHNLM856nYsnZKO1JteeSEG67azXESKayKAsSrl9fj79wA4au1089KnRD
ciGmEyKn39oLgldW/jyo3uYnL+cirOcKhBCiCEWs4ILp+WH7PhWs4nLp5//+Nmpidw4tzhkGYQINzI9GaV++Px
htld18rNb57CgbOyfUm1WM7/76HwSLCa+8NBu3D5/EFIKEftae9100TxGxwirsTzmHpeX5jH2PvMHNPVdTLxe
eZ0SsUUGZVHopf2NPLClmk+tGM/1c4qCdtzC9CT+6wOzOdjQw3+/cjRoxxiUiVh1r7uXvW6r52+ZqGrudRscJiy
NNPacF80gHV539Hv62qYoHt9awp7Zr1Ble3N/Io9treXRHRWwDJWKKDMqiTG2Hg68/sY85pRl87cqpQT/+FTM
KuGPJOP608RR7x/CiKUQ8aBssNPcHNB398TFb1tY78DhH85i7nF68/oFBVGvv2Yv3h2Poth397yw2ECIWSE1Z
FNfa8/XH96E1/Pr2eSRYQjOm/sbVU3n9cDPfeHI/z37+YqxmGbsLcTbzxmxS4/KRaDExJT+4LSEi1fyyDPrCXhIt
ZiaP8DGXZyezsDyTHqePJRwj3yng8un57KrpZFKExV6fREyRQVkuEh7LZsr2/nRzbMYI50cstvJs1n53g0z+czf
d/KXt09x78oJlsvIaJZUoI57IYrJydYuGpm4ahuq5RixaTcMWcozUqmNCt0r4FCGEU+YkSJpm4XP3rhMBdVZP
PhxaUhv7+rZhZwydQ8fv36Cdr7Rn+aQQghQiUQCCLCjqoPGrvio5xOxTwZIUeLfnzmANxDgxx+YFbau+/92zTQc
Xj+/WHs8LPenhBAj8aeNp/jpK0f51yf3jXlFpxCRQAZIUeDNly28dqiZL102mbLslLDd78S8VO5YPI4Ht9VwoqU3
bPerhBDDUd81MBBzewNjWjggRKSQQVmE8/gCfP/5Q1TkpvDxi8eH/f7/5bJJFvN/PglaZEhhIgsH7+4nFkl6Vw7
u5CZxelGxxFizKTQP8L95e1TnGrr568fXxSy1Zbnk52ayL0rkj/Za8fYX9fNrBJ54RNCRIay7BS+fe10o2MIETQyU
xbBWnpC/Or141w2LY/VU/IMy3HXxeWkJ1n55evHDMsgbBi9XpeX6vZ+AjHW00trTW2Hg26n1+goQgSFDmoi
2M9ePYbXrw3/JJhms3LP8vGsPdZC/rpuQ7MIUbg5fXz4NYantxVzxtHWoyOE1Rvn2jn8Z11/H1LNb0uGZiJ6Ce
DsghlvLmXx3bWcudFZZTnhK+4/1xktkyI6OT2BxB4Bval7IyxPTqHhO/HF8DpkX0wRfSTmrII9dNXjpKSYOGf
1kw0OgrwzmzZz147xoH6bimqFSJKpCdZuWxaPnWdDhaPH30X/Ui0clluVrOJXHsieWk2o+MIMWYyUxaBdlZ3
8uqhZu5dWUFWSoLRcU772LJyUhLM/HFDpdFRhBAjMKsknatnFZKdmmh0IKBKT7Zy1cwCFpRlGh1FiKCQQ
VmE0VrzK5ePkJOayCdXhL8FxmKJ1m5ffe4ntvXSL100BYCgLPoBy3SuFQIEQQyKIsW6462su1UB/986USSEy
Lv7PInlg8MFP+y8ZTBSYQw3sGGbh7bUceD22qo63QYHUcIEeVkuBZBtNb89JWjlGUnc/vicUbHOavijCSum13
IQ9tqZBm6iHs9Th8AWkOvy2dwGiFEtJNBWQR59VAzhxp7+OdLJ2E1R+5Tc+/KCvo9fh7aVmN0FCEMNb8sg7
njMlgyPosp+Xaj4wgholzKvPHGa01v1x7nPLsZG6YU2R0nPOaUZTO8ok5/PXtKnz+gNFxhDBMosXMmil5LJuY
g8mkjI4jhIhyMiiLEK8NzpJ94ZJJWCJ4lmzI3cvKaepx8dqhZqOjCCEM0tzjoqY9smrpWnpCvLf3Gx1DiFGJ/Hf/O
KC15pevD8yS3Tg3smfJhgyZmkdJZhL3b64yOooQwgD1XU4e2lbDE7vq2FfXZXQcAJq6XTy4bWD3gt01nUbHE
WLEZFAWAdYebuFgQw+fj5JZMgCzSXHN0jK2VHZwtKnX6DhCiDDrd/vQg1tp9kXIIoe+MzLJwgsRjaJjBBDDt
Nb8Yu0xyrKTuSlKZsmGfGhhKYkWE3+T2TIh4s6kvFSWTchmflkmC8ojo3nrhNwUlK/KYd64jJjbvUDEh8hrhBV
nXh+cJfvpB2dHzSzZkMyUBG6cW8STu+r5+IVTSU+yGh1JCBEmSimWVGQbHeNdlfIsKpfBmlhe0TUKiDFDt
WRl2cncPK/Y6Dij8rGLynF6/Tyxs87oKEIIIURUk0GZgdYfb2N/fTefWz0h6mbJhswsTmdBWSYPbKkmENBGxx
FCCCGiVnSOBGLE/755gsJ0GzfPKZe6ypjCubSMU239bK5sNzqKEEIIebVkuGaQndWdbD3VwT0rKkiwRPfTc
NXMAjKSrTwoHf6FEEKIUyvu0UAU+926E2QkW7I9UanRUcbMZjXzgfklvHqwibY+t9FxbBBCiKkgzIDHG3q
Ze3hFu5eV5k5KYmwsGp3w4IK8fi0F/0IIICQoyaDMAP/31kmSE8zcdVG50VGCZmKencXIWTy0rUYK/oUQQohR
keFZmNV2OHh2bwMfXjyOzJQEo+ME1YeXIFLV7mCLFPwLIYQQIyaDsjC7b30IJgX3rBhvdJSgu3pmIelJUvAv
```

hBBCjEbYB2VKqY8ppV5XSq1TShUrb6mlNqolPqHUiqmW8K39rp5dEctt8wroTA9yeg4QWezmrllfjGvHGyiX  
Qr+hRBCiBEJ66BMKVUMrNJaX6q1Xg14gDva6+XAPuCmcOYJt7+8fQqPP8C9qqmJhIydyweN1Dwv0sK/oU  
YK601R5t6qe1wGB1IWGo7HTy9u562PpfRUYSISuGeKbsSMA/OIP0aWaysG/y7tcDSMOcJmx6Xlwc2V3P1zAI  
m5KYaHSdkJuXbWVseyUPbatFaCv6FGIud1Z28uL+Rx3fWUd/INDrOeQUCAb737EE21bDD58/YnQcIaJSuAdl  
+UCC1vpSwAFkAD2Df9cNZJ7tRkqpe5VSO5RSO1pbW8MSNNj+VqWaXrePz66aaHSUkPvw4nHS4V+IIHD7Au  
987/UbmOTCAoF38rp9kZ1ViEgV7kFZN/DW4PdvAOVA2uDPaUDX2W6ktb5Pa71Qa70wNzc31BmDzuX18+eN  
VayYIMOsKnSj44TcNbMKSbNZeHCrFPwLMRaLx2dx0YRsLp2WR0WEz7BbLCa+fPlkVk3J5UuXTzY6jhBRKd  
yDsk3A7MHv5wK1wKrBny8DtoQ5T1g8vrOOTj43n109wegoYTFQ8F/CqwebpeBfiDGwmk0srchmdkmG0VGGZd  
64TD63eiKT8+1GRxEiKoV1UKa13gM4IVLrgEXAw8B6pdRGGbZpT4czTzj4A5o/bTzF7JJ0LqrINjpO2NyxZBw  
ef0AK/oWIMnWdDg7Ud+PzBy585VGo7XBwsKEbfwQ1ma7tGHjMkZRpJKra+jnc2CN1vDEg7Hv8aK2/+p6LfjL  
4FZNeO9TMqbZ+fnPHPJRSRscJm8n5dhaWDRT8f2pFRVw9diGiVVufm8d31qE1tPd7WDU5uOUizT0untg1cPxu  
h5dlE3OCevzRaO11n87U6fCwYIJ0lcjUdj4anc9AP1uHwvLswxOJMZCmseG2B82VFKSmeRVMwqMjhJ2QwX/  
Wyo7jI4ihBgGf0AzNNni9QV/pszrD5w+vidEM3Ej5Qu8k8kbIZIG4szMXr/MIEW72NgNO0LtrO5gZ3Un371+OhZ  
z/11/r51dyPeeO8iD22q4aEL8nLoVIlrlp9m4dnYhHf0e5pZmBP34JZnJXDWzgB6nl3njzrYPuwK05O4Z1YhnQ4P8  
8ZlGB1nxCpyU7l8ej5Or595IXjORHjJoCyE7ltfSXqSIVsXlhodxRBDf8Pbq2hvc9NdmqioZGEEBcQ6iL9aYVpF7  
5SmE0piO6FCTOLY39Vf7yIv+mbMDnV1s+rh5q5c2kZKYnxO/b98OKBgV8nd9UbHUWIkDrZ2seB+m4CUVos  
LiJLbYeDvbVdUXIKVYyeDMpC5I8bKrGaTHxsWZnRUQw1pcDOgrJMHtpWlyuDRMyq7XDw7J4GXjvUzM6a  
TqPjiCjX3jew+OCNIy1sPN5mdBwRRjJoC4H2wRVMt8wvJs9uMzqO4e5YPI5KKfgXMezMVgo+KbYWY+TX7  
yy4iNY2HWJ0ZFAWAn/bXI3bF+CeFeONjIRp090OH/oW3S4V/EpvKcFK6cUcDKybsKo+MANyRvflsNq6f  
U8TFE3NYMdn4tiEifOK32CIEnB4/D2yp5tKpeUzMi+7i0WA5s+C/o99DVkqC0ZGECrRZFXwC6i18S8yN5W  
S4SGzJQF2RO76ujo93Dvygqjo0SUoYL/J3ZKh38h4llax97artCtmNAKPgDmn11XRxv7jU6iohxMigLIn9A88cN1  
cwpSWfxeOmqqCYp+BdCNHY7eWZPA28eaYmqGtOd1Z28friF5/c1Ut3eb3QcEcNkUBZErX1qpqrdbw0rJ8i2Qmf  
x4cGC/62nouffWAgRPGd+HtNEz4ezMz9IymdKEUpSUxZEf9xQSWIWEIfOyDc6SkS6bnYh33/uIA9urWFpHG3  
OLOQYUJSRXPVzCul2+phdEj0NTxeWZ5FgMZGcYKE8J8XoOCKGyaAsSPbWdrGjupPvXBefWyoNhxT8CyGic  
QGU2aQiZlsoEdtk9BAkf3n7FKmJFm5dWGJ0llg2VPD/215ao6MIUUTQ9Li8bDvVQVO3y+goIorJoCwImntcPL+v  
kQ8tLMVusxodJ6JNKbCzZHwWf9tcLU0RhRAx46X9jbx9oo0ndtXh8UXPyllRWWRFgR/31KNX2vuXIZudJSo  
cPeycuq7nKw93Gx0FCGECARFO4u7ZJ2XGC2pKRsj9fPP7bWcNm0fMZIJxsdJypcPj2fonQbf327iitnFBgdRwghx  
uya2YUcbeqhNDMZq9QVi1GS35wxemZPPR39Hj5xsWypNFwWs4k7Lypnc2U7R5p6jI4jhBBjlpPoYUFZFnlpst+  
xGD0ZlI2B1pq/vF3F1AI7SyukWexl3L6olESLifs3VRkdRQgRARweH1sq26lpdxgdRQjDyKBSDDafbOdIuy+fWD  
5emsWOUGZKAjFPK+ap3fV0OTxGxxFCGOy1Q81sPtnO03vq6XP7j14jhCFkUDYGF377FNkpCdwwp8joKFHpr  
mXluLwBHtku7TGEiHdm08AHWzX4JUQ8kkHZKFW19fP6kRY+srQMm9VsdJyoNK0wjaUVWdy/qQpvFG1OL  
IQIvsum5bNmah63LiwlJVHwOln4JIOyUbp/cxUWk+KjS8cZHSWqfWpFBQ3dLp7b22B0FCGEgWxWM3NLMy  
hIl0J5Eb9kUDYKDo+Px3fWcfXMQvLs8gLyFmum5DEl387/vXWSgDSTFXGurc/N2yfaaO6J3K7wPn+A7VUD7K  
/rNjoKTo+fzSfbOdHSA3SUqBeNv3vxQAZlo/Dc3gZ6XT7uvKjM6ChRz2RSfHpVBcea+3jzaIvRcYQw1LN7Gth2q  
oMnd9WjdWR+SNlZ3cnG422sPdzMiZY+Q7O8dayFLZXtPL+vkfY+tf6Zot0zg797T+2uNzpKXJNB2QhprXlgSz  
VT8u0sLJMNaoPh+jlFFGck8bt1J42OIoshrGZ1+s9IXdFtOaMx6lBeowwlaTUpdXqhgBidoefSIv+OhpJqyhHaW9f  
NgfoefnDTzlh90Yw2VrOJe1aM53vPHWJ7VQeLyqXnm4hPN80rprK1n/LsFKOjnNP8cRkkJ5ixWc2UGZxz5eRc8  
tNsZKUkkJGcYGiWaHdzFPzuxQOZKRuhBzZXk5Jg5uZ5xUZHiSm3LSolM9nK/755wugoQhjGbrMypzSD9GSr0  
VHOSSnFtMI0xucY/+ZtNZuYWZxOUUaS0VGiXjT87sUDGZSNQGe/h+f2NXDz/GJSZcl2UCUnWPjk8vG8ebS  
V3TWdRscRIuLVdTpYf6w1bLVU/W4fG4+3cbxZiuqFCBUZlI3A4zvr8PgCfHSPPiHwt0XjycrJYGfv3bM6ChCR  
DSfP8DTu+vZWd3Ji/sbw3Kfbx5tYXtVBy/sb6SzX3bhECIUZFA2TIGA5u9bq1lUnsnUgjSj48Sk1EQLn101gQ3H2  
9ha2W50HCEilkpEiwDL9+JlvA0r04YLKo3K4XF4AJ/IWKVDMqGacOJNqrbHTJLFmIfXVpGnj2Rn712LGJbA  
ghhNjNj8aGFpVw+PZ/rw7TN2yVT87hiRj63LSrFbpO6lyFCQQZlw/TQ1hqyUhK4amaB0VFiWIKCmc9fMpFtpzr  
YcLzN6DhCRKyM5ARmFqeTlBCemTKL2cSMonTy0qRhthChIoOyYWjrc7P2cDO3zCsO26mCeHbbolJKMpP40  
YuH8cmemCLKdTk8rDvaQmWrsY1WRXQ52NDNhuOtODw+o6OIMJJB2TA8taseX0Bz26JS06PEhUSLmX+7Z  
hpHmnp5eHut0XGEGJNXDjaxu6aL5/c14vT4jY4jokBzj4tXDzazo6pTzhjEGRmUXYDWmkd21DJvXAaT8u1Gx4  
kbV88sYPH4LH7+2jG6nV6j4wgxajbrwOy61WySrvNiWBLO+F1JssrZmXgig7IL2FXTxYmWPm5bKLNk4aSU4j  
vXTafT4eGXa48bHUEiUbtqZgFXzjgw4tLT6+YFOJ8MIMSuH1RKdfMKmT5xByj44gwkleIC3h0ey3JCWauC9M  
KJ/GOmCxp3L6olPs3V3GgvtvoOEKMSqLfzPSiNNkGSIXpqNKQV2TDK7GldkUHYe/W4fz+9r4NpZhdLB3y  
DfuGoamckJ/OsT+6ToX4gwOtnaxxtHmsO2Y0Co1HY4eONIM43dTqOjCHFBMig7jxf2NdLv8UuBv4Hsk618/8Y  
ZHGzo4Y8bTxkdR4i44PL6eWffl3tru3nlYLPRcUZNa82zexvYW9vNC/vCs/OBEGMhg7LzeGRHLRW5KSwoyz  
Q6Sly7emYBV87I5+evHeNIU4/RcYSIeWaTInGw/i0IMXoLzZVSpAz2cUtOkLMdIvLJoOwcTrX1s7O6kw8tLEU  
pOadvJKUUP7xpFmk2K198aLe0FRAixKxmE7cvHsc1swq5emah0XHG5NaFAwXzn88rNjqKEBcU1kGZUqpcK  
dWslFqnlHp18LKvKaU2KqX+oZSKmL07nt5dj1Jw41wp8I8EufZEfv6hORxr7uOHLxwyOo4QMS89ycqUANvUrx  
hNSbQwpcAetp0PhBgLI/63vaa1Xq21vkIplQus0VovB/YBNxmQ53201jy9p55IE7IpTE8yOo4YtHJyLp9eWcE/ttbw



2A5pKivEaHX0e1h7qJmjTb3v+7s9tV28caSZfrcxneRdXj/rjrawvaojKPvflrQ7eO1QM/VdkVfo3zn4PEhZhhhxEn2  
NUqpDcCTwDFg3eDla4E7gMcMyPQuu2u7qG538Pk1E42OI7jq1dO4UBDN//21H7Kc1JYVJ5ldCQhos7aw83Ud  
zo50NBNCwBS6dXIDV1O3jzSAoDXr7lyRvj3+t16qoPdNV0AZKckUJGbOupjaa15bl8DHI+AqrZ+PrWyIkpgp2Pt  
4Wbqhp6HjCTZ6F2EfaasEZgMrAEuAxYCQx8RuoGzVtQrpe5VSu1QSu1obW0NecindtWTaDHIJ5uMRyGo28b93  
LKA0M51PP7CTU239RkcSIurYBwdhNqsZq/mdmtnkBDOWwb5YdoPaAA0NEE1KkTLGDEqp08dLtUveob99M  
FOixYzVHN2niUVwqGBMD4/qjpX6LAMDSmKt9X8ppeYDH9Vaf/18t1u4cKHesWNHyHJ5fAGW/Ggtyyfl8usPz  
wvZ/YixqWzt44P/t5IEi4IHP30RpVnJRkcSUW7hwoWE8rUlkvj8AaraHeTaE0IPevfsTFufmx6nl/E5KYYtcqpq6yc5  
wUxemm3Mx3J4fDR0OSnJTD695VWkON/zlGKHUmqn1nrhcK4b7kL/MzePvBg4Aawa/PkyYEs485zN+mOtdDq  
83DxPCvwjWUVuKn//5BlcHj8f/sMWajscRkcSImpYzCYm5qWedSCQk5pIRW6qoavOy3NSgIgg4FWGGBPz7BE  
3IIPzPw8iPoV7vnSFUmqnUmoT0KC13gqsV0ptBOYCT4c5z/s8taee7JQEVkzKNTqKuIDpRWn8/ZNL6HF6ufl/N  
7GvrsvoSEKEzYmWPI4+0DSiTvXV7f28fKDpfr9iP4AG4638taxVrwh3DIDa832qg5eP9yMwzP6hQT76rr4fy8e5  
o3B+rczj7+1sp03jjTj8krrHBF9wjoo01q/qLVeoLVeprX++uBLP9FaL9da36G19oQzz3v1uLysPdTM9XOK5Px+1JhV  
ks6Tn1uGzWritt9vka7dli74/AFe3N/I4cYeXjnQNOzbPb9v4DYv7H/3/5NDjT3sqOpkV3VnSD/c1HQ42Hi8jX113  
Ww+2T7q4/xu3Un21Hbxxw2V7xrcnWztZ/PJdvbWdrPtVEcwIgsRVjLyOMPQ824fQFukN5kUWVinp0nP7eMqY  
V2/unBXXzzyf3SYFbENNMZBexpIzj1NXSa7L2ny9LOWPWXFsiVgKmJFsyDCwnGcsouKyXh9PESTO+8jdlTF  
kyDp11H8u8iRKQwrNB/tEJZ6P/Jv27nSFMvG/91jXTxj0Jef4D/fvUov3+rknFZyXzvhmsmZpndCwRJaKt0N/h8d  
HY7aI0M3nYDV5dXj/1XU6KM5LeV2PV1O0ioDVFGaHtzdje56bf7Wdc9ugX5/S5fOyo7mBmcRo5qe+uPWvtde  
Py+mXxj4gYEVvoH8m6nV7WH2/lmlkFmiCLUlaziW9ePY0HP7UEq1nx8b9u5577d3C4URozitiTnGBhQm7qiDr  
u26xmJuSmnrXovSdDfVIBGUB2auKYBmQw0N5i9ZS89w3IYGD3DxmQiWgl7JBx1qxuvXXDtTl1Gu2UTcn  
jpn1fy9aumsLWynat/uYFPP7CDPbVdRkcTwiCn2vp5cX8j1e3G9PcbKsR/9WDTiHYM2FndwcsHmthR3cGL+xtH  
tLghVHbVdPLygUa6HlaWQosYE3nd9Azywr4GijOSmFOSbnQUEQQJFhOfWz2Rjywu489vn+LPb5/iLYPNzCpO  
586lZVw/p0j2whNx58X9jXh8AarbHXx29YSw3391u+N0gb/ZpLh0Wv4Fb9PU7WL9sTb8Ac2ptn4m5qXS2uvmr  
mXIIU57bi29L46OtDI3OPX3DBHPsyL4JCZMqDb4WXD8Taum10opy5jTHqylS9dPpIN37iE7984A7fpz9ef2MeS  
H63lu88e5Hjz+/f+EyJWZSQPFL9nJhtTBG+3WU7vGJAxxAzJiWYSLCZM6p0C/+HeNIRSEiynTtxnyICEURS6  
A88uqOWrz++j2f+6WLmlGYE9dgismit2Xaqg79vreHIA414/ZpF5ZncsWQcV88sjMgGkyI8oq3QfzRcXj9N3S4K  
M2wkWoz5Xe/s99Dv8VGSOfy6r26Hlx6Xl+yUBFp63ZRkJmExuG1Rt9NLt8NLaVaSfJgX5zWSQn85fQm8sK+R  
kswkZsupy5inlGJJRTZLKrJp75vO4zvrgHbDV96ZC/fe+4QH5xfwieWjw9LwbMQ4WazminPSTE0Q2ZKApmD  
M17DIz5sJX1wdqzcoD053ys9ySqd+EXQxf3py85+D2+faONaOXUZd7JTE/n0qgm88ZXV/OOeJvW8IYe/bqpi9U/  
X8e9PH4iYmIhwm1LZTsv7W+k2+kd9m12Vnfywr5GOvpDU/S+7dRAGX8oiuq11mw+OfCYe1zDf8xDjjb18tzeB  
tnqLQycHj+vHGxi/bFW/IHoOss3XJHxkcNArx1qxhfQXDur0OgowiAmk+LiiTlcPDGHuk4Hv33zJA9tq+GxnbV8f  
s1EPrWywrBTPUKEU13nO4X4SsFVMy/8utjW52b9sYGid68/wE3zioOaqbHbydsn2gAlaM11QV4hX9fpZEvlwG  
M2mRRXzigY9m19/gAvH2gioDUtvW4+uXx8ULOJd9tR3cGhhoEWR3lpiUwtSDM4UfDF/UzZq4eaKM5IYlaxn  
LoUUJKZzP+7ZRZvfnU110zn479fPcZVv9jAzmrZskXEPvNerqAPSslcVi3SU4wn67FzBrhacnhSE20nJEp+MdP  
O+MxZ4/w+GaTijPFOqrbiPebv7NjKvmcmz+e8d1ob/D42Pe91/jw4vH8d0bZgTlmCK2vHWslW8/vZ/6TidfvHQS  
n18z0fACYx8aEaVDoPw9L199Lt+I6ir73D66HB6KM0JT9N7j8tLr8IEcolrPbqeXfvfIHvMQI9dPa6+bwnSbvDaEQ  
XOPiwSzacR1iUaSjv7DtP5YG25fgCumX7hXjohPqybn8uIXV3DT3GJ+sfY4n7h/x6jqToSIFmk264gHJ6mJFkoyk0  
NW15tms4ZsQAYDRfujXdxjs5opzUqWAVmY5KfZompANlJx/Vv06qEm0pOsLBqfZXQUEcHsNis/v20uP75Ifpt  
OtHHL/26ipl2KesXouH1+XjvUzGuHmnH7/EbHCSutNRuOt/LCvtEV1Yuxcfv8vHqwidcONePxByyOI84ibgdIPn  
+AN460cOnUPKzyCUcMw+2Lx/HAJ5fQ2uvml1t9v4kRLn9GRRBQ6UN99xld87cta1e5gR1Unx5p72VopdZrhdqC  
+m4MNPQO/ew3dRscRzXG3o5HtVZ10ObxcLqcuxQhcNCGbRz69FH9Ac/t9WzjaJDsCiJHJSU1EqYHVjTmpsXs  
a5mwykqxYzQONOHptw1tIIInO2Xgd8+kFDnDXMghwituW2K8eqjJBiuJlZNzjY4ioszUgJevvci7vjDFj76p608+  
dlIIgYNvzu5iG9L2S187KJyIDSrCSNZZkoCH1tWjtPjJz/NZnScuFOek8KdS8tQsXd7160iMuZMq01rx5sZsXEhFI  
ipDu0iC4T81L5+z1LcHv9fOzP22jvcxsdsUSRrJSEuH1TTLNZZUBmoOzUxLj93YsGoxqUKaXylVJ/Ukq9NPjzd  
KXUJ4MbLXQONfZQ3+Xkihly6lKM3uR8O3++exENXU4+ef+OuCvaFsF3oqWpp3bXcaQpNmvNAGHNuqMtP  
Lu3YUQ7BgxWi6vn5cPNPLKwaaWNww2pmyvwKvAEotlY8B/xKEPGHx2qfmlIJLp8mgTlzNwvIsfnHbXPb  
UdvGdpw8SbX3/RGRZe7iZqjYHrx5sjsnfpeoOB7trujjZ0se2U1LoL0JvX103hxt7OdTQw6HGyP+wM9pBWY7W+  
IEgAKC19gFRM03w5pEW5pZmkJMqhY5i7K6eVcjin10zkkR21PLitxug4Iorlpw28JuXZE2NyL97M5He65w89ViF  
CKdd+xuKGGKfHYM9qCqn6lVDagAZRSS4GoWF/b2utmb103X7l8stFRRaz50uWTOdDQzXefPciMonTmlmYY  
HUIEoetnF9HW5yE7Ct48RiMjOYg7BgV9ZfWlCifxOSncdVE5JqVIT7YaHeeCRjt9mXgWWCCUupt4G/AF4K  
WKOteGtw4d83UPIOTiFhiNil+eds88uw2/uXh3fS7fUZHEIHijYjZrK6G6L6d6JqYkWGZCJsMpMSYiKARmMcIC  
mtd4FrAKWAZ8GZmit9wUzWKi8ebSFPHsiM4pib3d5Yaz0ZCs//9AcqjsefP+5Q0bHEUKM0omWXh7fWcfBYT  
RY3V/XzeM766hslWbSYuxGdfpSKXXLey6arJTqBvZrrVvGHIs0vP4A64+1cvXMgpis1xDGW1KRzedWT+C3b5  
5k9ZRcrp5VaHQKlcQlvX64BYfHT32nk2kFaZhMZ3+/8PkDvH6kGa2hy+Ghljc1zElFrBntHPkngT8CHxn8+gMDp  
zTfVkrdGaRsQberupNel481U+TUPQidf7lsMrNL0vn20wfo7PcYHUcIMUIF6bbBPxPPOSCDgbKFPPvAdQvTQ7  
dhuogfoY30DwDTtNbNMNC3DPgdsARYDzwQnHjB9ebRviwmxflJOUZHETHMajbxkw/M5vpfb+QHLxzi5x+aa

3QkIcQIXDe7iPZ+N1nJ519woZTi1oUldDo8sm2RCIrRzpSVDw3IBrUAK7XWHUDEdgrcd7SFRvVZ2G3RUfAn  
ote0wjQ+u3oCT+6qZ93RiD2jL4Q4i6EZMMswFlxYzSby7LbzzqgJMVyjHZRtUEo9r5S6Syl1F/AMsF4plQJ0BS1d  
ENV3OTnS1MuaqBLXpQiPz18ykQm5KXzrqQOyGlPEHK8/wMsHGnl6dz29roj9LH5BgYDmjSPNPLGzjo5RIBv  
sqe3i0e21nGjpDUE6EW9GOyj7J+AvwNzBr22A1lr3a63XBCdacA3NVlwirTBEmCRazPzXB2dT3+Xk12+cMDq  
OEEF1vLmPw429nGrrZ3dN19FxrQ2208He2m5qOhxsO9U+otv6/AHWHW2hvsVJuqOtIUoo4sloW2Jo4CQDPypv  
Bi4FDgexV9C9eaSVkswkJsJqGBFGC8qyuHVBCX/aWMIJWtIvYkheWiIJFhNKQVFG9G4wnpWSQFKCGYCij  
EV61vMptObqxeP8LZCnM2ICv2VUpOB24EPA+3AI4CK1NmxIW6fn7dPtPHBBSXSCkOE3devmsrLB5r47rMH  
+dsnfsvvoIgJOamJfPzierx+TXpS9Nbp2m1W7l5WjsvrJ+MChf1nc+uCErqc3gsuChBiOEY6U3aEgVmx67XWY7X  
WvyYK9rzcWd2J0+tn1WSpJxPhl2tP5EuXT2bD8TZePdR84RsIESWSEyxRPSAbYrOaRzUgg4HZspzU87fOEGK  
4Rjoo+wDQBLyplPqDUupSIOJ/Ezccb8NiUiydkG10FBGnPNZRGVPy7fzg+UO4vBH/OUZEKRMtfTy8rYYdVR1  
GRxGj1Nnv4bEdtbx8oAmfP2B0HGGgEQ3KtNZPaa1vA6YC64AvAfIKqd8ppa4IQb6g2Hi8jfnjMklNHG1bNiHG  
xml28R83TKeu08mf3z5ldBwRQzYcb6Wx28WG420y4I9SO6s7qet0crixh6p2h9FxlHFGW+jfr7X+h9b6OqAE2AN  
8I5jBgqWj38OBhm5pGCsMt2xCDpdNy+N3b56kvc9tdBwRI0ozk4GBLvSJltjdyDyWFWcmoDTAadTcVGLCG8/G  
PHU02DD294NfEeftE21oDStkUCYiwDeunsqVv9jAr984wXdvmGF0HBEDLP2Wx4KyTOw2iywiiVLTCTMozkw  
iwWzCZjUbHUcYKOY/Vm043kqazcLskgyjowjBxDw7ty0q5e9bqjnV1m90HBEDlFJkpiQMq/u8iFxpNqsMyERsD  
8q01mw83sbFE3Mwy8oYESH+5bJJFhM/NfLR4yOIloSIQ/6A5uUDjTyyvYa2c5RSdPZ7eHRHLc/va8Ariw/CJqY  
HZZVt/TR0u6SeTESUPLuNT6+cwEsHmthZLSvmhBDhVd3ez+HGxhq6XOyo6jzrdfbUdlHf6eR4cx+VrTKrHy4x  
PSjbcGxg24uV6kQ/mYgsn1o5njx7Iv/vxSMMbJAhhBDhkWtPJCXRjFJQlp181uuUZiVhUgqb1Ux+miw+CJeY7h  
Gx8UQbZdnJlGad/ZdOCKMkJ1j44qWT+PbTB1h3rJU1U2RPVfEEAzsYjAejz9wzLZRE/Ps3LMiCYtZkWiRWrd  
wMWSmTCn1ZaXUxsHvv6aU2qiU+odSKmitob3+AJtPtsuqSxGxPrSwlNKsJP77laMEAjJbJoQInwSL6YK9O1MS  
LTlgC7OwD8qUUonAnMHvc4E1WuvlwD7gpmDdz+6aLvo9fpZPIFOXijlWEx86bLJHGzo4eWDTUbHERGkpt3  
BA1uqeeNlc0yd3vb5Azy/r4EHt9bQ2ju8Xn2BgObLA038fUs1jd3OMWdo6XXx4NYaXtjXiM8fwB/QvLi/kX9sraal  
xzXm44+F0+PnyV11PLqjll6X19AswhhGzJtDA9w/+P1iBnYGAfGLLA3WnWw43orZpLhltlYSEezGucVMYkv1Z  
68exS+zZWLQ11PttPW62VvbTXu/x+g4QVPd4eB4cx/NPS521Zy9wPy9GntcHG7sobXXzfZFKWPxK7qLpp7XB  
xr7qW200lth4OjTb209LjZWT3244/FkaYeqtsd1Hc6OdjQY2gWYYyWdsoGT0+u0lq/MXhRBjD0m9cNZJ7jdvcqp  
XYopXa0trYO6742nmhjTkl6TGyWK2KX2aT4yhWTOdnaz1O7642OIyLE+JwUALJTE0izxc5rWN5ggblJKcqzU4  
Z1m+yUBNKSrCgF5ecoSh+J8pxklILURAU59kRy7YmDjXehbJiZQqU4M4kEiwmLSVGSmWRoFmGMcBf63wk  
8eMbPXUDx4Pdpgz+/j9b6PuA+gIULF15wOqHP7WNfXTefXTVhLFmFCIsrZxQwqzidX6w9xg1zikiQrXLi3sLyL  
KYVpmGzmmOqx6LdZuXjF4/H59ckJQyvVslmNXPXRWV4/AGSE8b+lJw1II2yrBQsZoV1sOHuXcvK8Qbp+G  
ORZ7dxz4rxa100ko1T4X71nwJ8Vin1MjADWAisGvy7y4AtwbiT7ac68Ae0nLoUUUEpxVevnEJdp5NHttcYHUdE  
iJRES0wNyIZYzaZhD8iGWMymoa6YkhLMpwwdKQ5mMHPANsbSYZUAWx8I6KNNa/6vW+kt9VXAQa3194  
D1gysx5wJPB+N+Nle2k2A2saDsrgdDhYg4KyflsHh8Fr964wROj9/oOELErOYeFw9sqebZvQ34/AF8/gDP7W3gg  
S3VNHWPvNB/88l2/vr2KQ7Ud4cgrYg3hp0nGVxxidb6J1rr5VrrO7TWQalo3XSyjXnjMuTThogaSim+duUUWnv  
d3L+5yug4QsSs3TWdtPW6OdnSR22nk7pOJyda+mjrdbN7mIsPhvj8AbZUttPp8LKlsj1EiUU8ibni1W6HI4MNPSyb  
IP3JRHRZVJ7F6im5/N9bJ+mR5fBChERFbiompbDbLOTZE8ILGyj0NynF+NyRfFpbzCbKc5IHj2vsIgERGyLjJHo  
QbTnVjtZIPZmISI+9YgrX/Xojf9pwii9dPtnoOELEnMn5dsZIJWM1m07X7H384vF4/YFRnV25aW4xLm9gxHVyQ  
pxNzM2UbT7Zjs1qYk5putFRhBixmcXpXDORGd9uqKQjhvpTCRFJ3ruq1WxSoy53UUrJgEwETcwNyrZUtrOoPE  
u2hhBR68uXT8bp9fn/b500OooQqEXxBXh6dz1/21xFcxC753f2e/j7lmoe3V6Lw+N7199trWznTtxtPnbMxrNcf4Jk9  
9dy/qWpUOwa8faKNP208xb66rtFEjzvHmnv588ZTvH44tnarCJaYgPs197k50tTL0go5dSmi18Q8OzfPK+H+TcF9  
4xLCaLWdDk619dPe52FPbVfQjntosON/fddA0f4QrTVbKjvocXrZeurshfj1nU4qW/vp6Pewp2Zkmbz+ANtODR6/s  
mMsDyFubK/qoNvpZV9dNz0u34VvEGdialC2ZfA/hdSTiWj3L5dNIqA1v37juNFRhAiagjQbdtTA/7UJualBO255Tg  
pW88BpxJLMD7r+K6WYmDdwP5Py7Ge9bX6ajfQkKyalmJA3skxWs+l0gf+k/OA9nlg29DwUZdguuCF6PIqpf5H  
NIW2k1JqYXSz1ZCK6lWYlc9uiUh7eVsU9KyYwLgJbywhhtJREC5+4eDy+gA7qzhXFGU18etUETEeq9r+HutbMLu  
dSbd86asaQEM3cvKx91phvmFOH2jW6RQDxaPD6LOaXpJhNKBV7zZHHKqZmyjadbGdReSYWc0w9LBGnvn  
DJJMwmXs9eP2Z0FCGCxmRSIdlK7MzVIO91oQHTWDIpnfFAvEq0WKWAdk5xMzopbnHRWVrv5y6FDEjP8  
3GXcvKeXp3Pcebe42OI87D5fXz6Pza7t9URUuv1AHGkw3HW/nD+sqglsgFW32Xkz9vPMVTu+vw+gNGxxHn  
ETODss0nB4o4pWmsiCWfWTWB5AQLP39NZssi2am2fuq7nHT0ezjY0GN0HBEmXn+AHVWd9L19bD8VuYX+  
e2u76HZ6qWpzUN858hWmInxialCWZrMwrTDN6ChCBE1WSgKfXD6elw40sb9O9taLVMWZSdhtFqxmxcQgFr  
CLyGY1m04vDphScPaFBjFgU14qZpMiM9lKfprN6DjiPGKm0H9zZtTLKrLPWVMgRLS6Z8V47t9cxX+/epT7P7  
HY6DjiLJNsVj65fDwBjbWgXZkb5hTh9QewRnAt86R8O+NzUjCbInRyRbjl/S0agbpOBzUdDpZJPZmlQXable+u  
msBbx1rZFsGnSOKdOsvKpXeflInAnSqiQx2jQuT/Jg3DUD2ZFPmLWPWxi8rJtSfy368clS7YYlhcXj+PbK/hTtP  
0dQtiw8i0ZGmHu5bf5KX9jeiteZoUw/X/3oD1/xyPXtqzr4DQbg097j488ZTPLytBpFXb2iWeBITg7Kq9n5yUhOYfi  
7mgEJEu6QEM1+8ZCLbqipYf7zN6DgiCtR0OGjoctHj9HKgXuoRI9Humi763X6ONPXS7fTy9J4GOvo9dDu9PL  
GrztBsBxu66XZ6aex2Ud3uMDRLPImJQdnXrpzKhq9fgklOHYgYdtuicZRkJslsmRiW4owk0pOsWM2KyfnygTUS  
TStMQykoyUzCbrNy5fR8kqxmEixmrpxRYGi2SXI2Eiwm0pKsFGcmGZolnqhoe3FfuHCh3rFjh9ExhDDE4zvr+O

pje/nNHfO4bnaR0XFiyS KFC4nF15Z AQMsH1gj23ufH7x84VWg2G9+QVn53gkMptVNrvXA4142JmTlh4sXN84q  
ZWmDnJy8fwe2TOg9xYfK mGtne+/yYzealGJCB/O4YIWZaYggRD8wmxbuncadf9rG3zZV86mVFUZHiksnWvp  
Ye7gZm8WExx8gyWrm5vklJFnNPL+vgYYuF5dMzWNKgZ39dd1sONGKPDGCw+MnMzmBG+cVkh5/xvv5p  
Pt7K7tZEZROqsm5xrwyMSZjjX38vrhFooybFw3uygiV9c6PD6e3FWPy+vn+jlFMd2HrLnHxXN7G7BZzdw8r5iU  
GNzQXGbKhIgyKyblsnPKLr9+4zid/R6j48Sl/fvdoD1+9tR20dzjoq3Pw6nWfjr6PVS29uPy+tlX1wXAntpO3N4Am  
yvb6XJ4qO9y0tB19tWQuwevu7umU+oGI8De2i5cXj+Vrf2097uNjnNWVW0OWnvd9Lp8HGqM7d0kDjF20Ovy0d  
rrpq93+g4ISGDMiGi0L9dM40+t49fvn7c6ChxaXphOhaTYlphGpkpCaQlWRmXnUxm8kBRtNmkmF40sLvIjOJO  
TEoxblwmyYkWcuyJfKaffTzJzTHAdWcUpUtPqQgwwSgNs0lRnJIEVnKC0XHOalx2MhnJVhKtpphf0DE5306i1  
UR6kpVxWclGxwkJKfQXIKr921P7eXR7La9+aSUvSrXpml200F9rjVLq9J9n+7vhXPdcxxWRIVqej2jJOvBR+Dil  
0F+IOPClyyaTaDHx45eOGB0ILg29MZztDeK9l53vuhe6rTBWtDwf0ZJzrGL9ccqgTlgolWtP5LorJ/DqoWY2nZC  
GsiI2lHc5+f1bJ3lgSzVOT+SvMP7cP3ax6r/e4PvPHTQ6SlA1nPE8ODw+o+PEDRmUCRHF7llRQWlWet959iBef  
8DoOEKM2ZHGHHweP229bmo7I7uTfLftW97aTnwBzfpjsfXB6EjTgc9Dh9PoOHFDBmVCRDGB1cx3r5/BiZY+  
/vL2KaPjCDFmUwrs2KxmclITKInwTvLpSQnMLsnAbFlsmxbey9PKUgjKWHgeSjNiuznIZZlob8QMeCe+7ez6  
WQ7r39lFYXp8gl6GrHa0V8IYSwp9BcizvzH9TPwBzT/+cJho6MIlYQYJRmUCREDSrOS+ac1E3l+XyNvS9G/G  
KHOfg9/3FDJfetP0tobmU1SY1lth4PfvnmCBzZXRCxihrHYW9vFr14/zjN76gkEoutMXTjJoEyIGHHvygrKs5P51l  
P7Y/6FXQRXZVs/vS4f/W4/J1v7jI4Td4409eLxBWjr81DfFdmLG8bqQEM3/oCmsrWfXres6nwwGZQJESNsVjM/  
umUWVe0O/mftMaPjiCgyMTeVzGQr6UnWmO8KH4mmFdpJSTSTn2ajJDM2O9UPmVOSQYLFxKT8VNJssbd  
35VjJv4gQMWTZhBw+vHgef9xQyTWzCplbmmF0JBEF0pOt3H3xeKNjxK2SzGTuXTnB6BhhMbM4nZnF6UbH  
iFgyUyZEjPnmNVPJT7Px9cf34vbJaUwRWuFawR/uTgHvvb8zfx76PhSZtNYhO244bhOJQvVveubxg0VmyoSIM  
Wk2Kz+6eRYf/+t2fvvGCb58xRSjl4kY1OPy8tiOOlxePzfNK6Y4IzStWNw+P4/vrKO9z8NVMwvCcnpl/bFWdtV0  
Mq0wjStnFFDX6eCZPQ2n/z6gNVazwuPTXD49n2mFaUG5385+D4/vrMOvNbfMLyBPfvaN60fqZGsfL+5rJCPZy  
q0LS7FZzRe8TW2Hg2f3NpBkNfOhRaWkJkbncKG1l8WTu+oxKfjggIKyUoK7sXxjt5OndteTYDZx68JS0pOsYzq  
ezJQJEYPWTM3jlnnF/HbdSXZWdxodR8Sg2g4HPU4vHl+AEy2hWxzQ2uumpceNP6A53NgTsvs506HGHRSGw4  
09BAKa4y19eHwB6jocNHQ56er3crSpF39Ac6QpeJlOtfT5/bh9Pg52dlftOMeberFF9C09Xlo6nYN6zbHWwYWH  
3Q7vdR3Rm9H/8rWfpweP/1uP1Xtws3HXKypR+3N0Cvy0dN+9gXacigTlgy9d0bZ1CYbuNfHtlNr8trdBwRY8bn  
pJCXlkhakpVpBaGbvSpIs1GWnUxKopnZJRkhu58zzR+Xic1qZv64TEwmXyzCNNKSrEwusDMxL5XCDBvzxm  
WSnBDcTBPzUslJTSaZ2cqUIP6bzipOJzXRQklmEkXDnNGcXpiO3WahIN3GuKzoXXwwJd9OZrKVnNQEJuS  
mBv/4BXbSk6zk2hMzn5sy5uNJR38hYtiOgg4+9PvN3DS3mJ/fNtfoOBFNOvoLIUJBOvoLIQBYWJ7FFy6ZxJO7  
63l6d73RcWKCLwgbvwcC2rDGmUP5z/U4tNb446Sp59meh2A8v+c7vhDnE52Ve0KIYfvCJRN5+0Qb33pqPzOK  
0pgkfahGbePxNrZXdTAXL5Xr5xSN6hitvW4e31kHwAcXlJBrTwmxmPN69WATBxt6SE4w4/T6Kc5l4pb5JZhNC  
gCX18+jO2rpcni5emZBTP+utPUNPA9awweWFJOTkshTu+up6XCwfFIOi8qzxnT85h4XT+yqw6wUty4MfoG5iE  
0yUyZEjLOYTfzmjvkkJZj59AM76ZH6slEbKuo+0dKHd5QzKlXt/bi8flxeP9UHKDw+nyNNvQDsrO5Ea6jrdNLne  
qerekuPm/Y+D/6A5lhzbHf2r2obKAB3ef1UtTlweP3UdAwUah8JwoKCytaBANCHJ/zPs4heMigTlg4UpNv4zR3zqe  
5w8JVH98oplVFauJZJSqKZ+WWZWm2je/mcnG8nx55Irx7DNRi8qzSEk0c/n0ffITLUwrTCMt6Z0TJoUZnspz  
kklLsjK7JLYbfe7Kt5NrTyTHnsiUfDupiRZmDhbEzy/LHPPpxpbYyUINIC8tkYl5wS8wF7EprIX+SqmZwH2AHzg  
BfAL4KnAjUA3crbU+78d4KfQXYvT+tPEUP3j+EF+5fDJfuHSS0XEiIH6CyFCIZIL/Y9qrZdprVcM/rwQWKO1  
Xg7sA24Kcx4h4sonLi7n5nnF/OylYzyzRwr/g8HnDxheGO/xBa84XYzcmAeAg0VrPepT5MMRCIT2+PEkmK8BY  
S30f88smBuYDKwb/HkteAfwWDgzCRFPiFL8+AOzaOx28tXH9pJrT2TZhByjY0Wtuk4HT++ux2o2cduiUjKSw1/  
MPbT4oDwnmZvmFqOUCnuGePadZw5wtKmXFZNY+PwlwZl9dnn9PLYthm6njytn5jO1IDg7BgxxeHw8vK2WPr  
ePa2YVMDEvdhd0hFqwXwPCXIOMlLpBKXUAYGNgUDhUUDkNnPEvILqXqXUDqXUjtbWljAlFSI2JVrM/P  
7OhZRnp/DpB3ZydLD4W4zcqbZ+vH6Nw+OntsoYrufHmgeev6o2B26ZMQsrh8d3+v/P7pquoB23tddNp8NLQGu  
Oh2DBRVO3i26nF39AcyKIOwfeO6o2R1BfA8I+KNNaP6u1ngnUAz5g6CNAGtB1jtvcP7VeQLVemJubG56gQsS  
w9CQrf/3EYpKsZj76p62cbI3tlXahMqMonRx7IsUZSUzIG3s379FYPD4Lu22gOH04exqK4ElOsLBMsh5pNgtXzy  
wl2nEL021U5KaQkwxl3riMoB13SElmMuU5yWQmW5lTGtsLokJtelFaUF8Dwl3on6i1dg9+/5/AUeA2rfW1Sqm  
vA1Va60fpdwwp9Bciel4393L7fVuwmbWP3HsR5TnGDCwigRT6CyFCIZIL/a9SSr2lHhOlyAf+DqxXSm0E5gJPh  
zmPEHFtUr6dBz+1FK9f8+E/bJF+ShHA5fUTjg/LLq9fWqOcx2ieh3A9d2Ph9QekwD+Cyd6XQggONfRwxx+3YD  
Wb+Mvdi5hZHH+nNCJhpuy1Q80cqO8e044Bw7HtVAdvn2ijIN3GhxaWnu7oLwasPdTM/vpuJuSlcsMwn4cNx1v  
ZUdVJaVYyH5gfmQsumntep3eTuHvHcXl2m8GJ4kMkz5QJISLQ9KI0Hvv0RVhNitvv28LbJ9qMjhSXjrcMFI2fb  
O0L6YzLiZaBGsKmbldI2jleuxODNZaVrX3Dnk0c+jet7YjcBRfV7Q48vgAeX4Dawd0LRGSRQZkQAhg4lfnk5y6  
mOCOJu/+yjScGP1GL8LmolpuMZCvLJuSEdKZl8fjM00Xe6cnWkN1PtFo6+DxcVJGNaziziEvGD9xmYXnkLri  
YWminMN1GcUYSU4LcZkMEh5y+FEK8S7fTy2f/vpNNJ9u566Iyv3d9FFvKRRNluH0pRAi9sjpSyHEqKUnWf  
nbJxbzqRXjuX9zNXf8YQvNPS6jYwkRETy+AC6vH39A4/T4Q3J8ty/4x70QrTX9bl/EL1SIdWHt6C+EiA4Ws4lv  
XTudWSUZ/Ovj+7jyF+v50c2zuGZWodHRhDBMR7+HR7bX4vL6gIHTmmum5jG3NCMox2/pdfHYjjq01tw8v4T  
ijKSgHHc4XjnYzOHGnhEtbbDBJzNIQohzumFOES98cTllWcl87h+7+PIje+h2eC98QyFiUEOXE5fXT5/LT03HQP  
uYU23Ba7xc3+nE4wvg9euwF+JXDj6OU639MltmIBmUCSHOqyI3lcc/u4x/vnQSz+xt4NKfr+Op3XXywi3izsS8V

MpzkpmQl8KKsbk2BNZVJ4VtONPKbAzLiuZ4owkpheFtxD/4gk5ZKcmsHxSdkS284gXUugvhBi2A/XdfOvpA+yt7eKiimy+d+MMJufHxmbGUugvhAgFKfQXQoTEzOJ0nvzsMn5400wONnRz1S/W840n9slCgCjU6xrYkHok+tw+fAZ0g/f6A/S5R9dPLRDQ9Li8IzVZDdXxvf4A/aN8zCJ6SaG/EGJEzCbFR5eWcc2sQn7zxgke2FLF03vquWd5BZ9eVYHdJn2vIt2mE21sPdVBj2RDY8qxTKMlic7qztZf6yVjGQrdyWZR6IIPL24nB4//9haTa/LxyVT85gzWqL6Z/c2cKqtnykF9pAsVHliVx11nU5mFqdz+fT8oBzT4fHx4NYa+tw+Lp2az6yS+NthI17JTJkQYISyUhL4zvXTef3Lq7liegG/efMEK//rTf6wvhKXN/xL+sXwVbUPFJG39brpdw/vuRraF7XL4Q3rYo8Oh4fewV0HqkdY/K61pnrwsVaFYF9Xnz9AXadZIFsQj9/eN/CYtYbqDtmPNp7IoEwIMSbjspP51Yfn8dznIzOrJIP/fPEwq3+6jge31sjGxxHqognZ5NoTWVCWOeyO/ovHZ5GXlsjsknRy7YkhTviOwjQbM4rSyE+zsXiErFVKKVZMziHHnsjKSblBz2Yxm1g+aeD4F0/MCdpxhwr989NsQV1IICKfFPoLIYJqS2U7//XyEXbVdFGencyXLp/M9bOLhr1djVGk0F8IEQpS6C+EMMzSimye+Owy/nTXQmxWM//88B6u+dUGXj/cLG00BG6fn17X2E5/aa3pcnhGvFBBiEgnhf5CiKBTsnHptHzWTMnjuX0N/Py1Y3zy/h0sKMvka1dOYWIFtERhQF6XF4e3FqDy+vnyhkFTcscXS+utYdbOFDfTVGGjQ8tLJW+WiJmyEyZECJkTCbFjXOLWfvlVfzo5lnUdzq5/b4t3Pmnreyv6zY6ngiz1l43To8frRITx/qawds2dLnw+mW2TMQOGZQJIULOajZxx5JxrPvaar51zTQO1Hdz/W828sWHdtPW5zY6ngiT8uwUphWmUZyZxIKyzFEfZ+WkHPLTbKyYIEOCRd7GROyQQn8hRNj1urzct76S379VSXKimf+4fjo3zS029DSUFPoLIUJBCv2FEBHNbrPylSum8MIX11ORk8KXHtnLp/62g85+j9HRQqrb4ZUebhHM6w/QMczfQbFPT5cjt9fRfjJoEwIYZhJ+XYe+8wyvn3tNNYfa+OaX21g26kOo2OFxP66bv789in+uqlqzKsPRfD5A5qHt9Vw/6Yq3jzSct7rurx+HthczV/ermJ7VWz+vgpjyKBMCGEos0lxx4oKnvzcMhltJm6/bzN/3FAZc+0z6rsGitOdHj/tfTLDemlcXj9tg89LXzfzVnftdnP7zJQ33n+6woxEjIoE0JEhJnF6Tz/xRVcMb2AH75wmH97an9M7QiwqDyL0qxkZhWnMy4r2eg44j1SEi1cPDGHogwbKyedvzt/nj2R+WWZFGemcdEEae8igkcK/YUQESUQ0PzstaP89s2TLJuQzX0fW0hqYuhbKkqhvxAiFKTQXwgRtUwmxdeunMrPbp3D1IMdfPSPW8O6AfZo9bt9EVv47fb5aetz4w9oWnvdUdcJ3+X10y6tU0QckI7+QoiI9IEFJdhtFj7/4G4+/IctPPDJxWSnhm8j7JFo73Pz8PZavP4AV88sZEeqB3ehlP3l8Af6xpYZu58DKT5vVTHlOMjfpKzE62rD0u338fUs1Do+flZNzWFAMG3SL2CUzZUKIiHXFjAL+cNdCKtv6+EgEz5i19Xnw+AJODY3dkVX47fD46HYO/LtVtfcDA53wo0W304vDM9BGJJpyCzEaMigTQks0VZNz+cPHFnKytY9P3r8dpyfy+nxNyElhRIEaFbkpZB9Dp/pQyEhOYGI FNqVZydyxeBwlmUlcOi3P6FjDVphuY35ZJuOykmXPVBH5PSIECLirZiUyy9vn8fnH9zFZ/+xk/vuXBhR2+tYzCaumFFgdIxzIUyVgkopVk3ONTqGEGEROa9qQghxHtfMKuRHN89i3dFWvvPMgbD2MfP4AJT3uAicUSDf0usK6axdS48rJrv/97t9Y97vNBDQNP48PjG1jKl2+GN2FPIj7JTJkQlMrcvngctZ0OfvVmSaYU2Pn4xeNDfp+BgoAR7TW09XmYVmjnqpmFbKlsZ/PJdpITzNx5URnJCcF9Kd14vI3tVR3YbRY+urQMm9Uc1OMbpcvh4R9ba/D4Alw6LY/ZJRmjOs7LB5s42tRLjj2Rjy4ZN6o9U2s7HDy5qx6Am+cVMY5bescJ48lMmRAiqnzl8ileMT2fHxz/iPXHWkN+fx5/gPbB/RCbe9yDfw4UnDs8/tOd3YOpafD4vS4f/e7gH98oHf2e07NbQ/+Wo9HUPfDv097nxjfk9h4tvW4CWhPQmpZeWUAgIoMMyoQQUCVkUvzPbXOZUpDG5x/cRW2HI6T3Z7OaWTMlj7LsZNZMGSIQXzYhh/E5KSwZn0V+mi3o97liUg7lOcksm5AdsW1ARqM8O4W54zKoyElh8fJrt7a4ZOrA83Hp1Hys5tG9jc0oSmNaoZ1phXZmfQePOosQwSQd/YUQUam2w8E1v9pARU4Kj31m2ZgL/6WjvxAiFKSjvxAi5pVmJfPTD85hb103P37pSFju0x/QNHQ5cfveX4Df7fTS2itd50dLa01jtzMiW54IES5S6C+ElFpXzSzg7mXl/PntUyppyOLKELeleGF/Iydb+shJTeCjS8tOF5i39rp5eFsNvoDmyhkFTC9KC2mOWPTWsVZ213SRmmjhY8vKSLTExuLGIUZCZsqEEFHtm9dMZXZJO197bO/pAvBQGZoJ6+j3vmv/yE6H53TBeavs0TgqLYP/tn1un8yWibglgzlhRFRltJj55e3z8Po1X39iX0j71102LY/xOSlcNj0Pyxkf5hNyU5lBmsGUAjsLI6yjf7RYPTmXitwUVk7OISM5weg4QhhCTI8KlaLe+JwUvnnNVL7zzEH+sbWGjy4tC8n9lGWnUJad8r7LzSbFmqnRs3VRJMpLs3Hj3GKjYwhhKJkpE0LEhI8uKWP5xBx+9OJhgq33h6rXpeXhi5jNxbv7nHR5fC873KX109thwOff2xd7cOt2+kd82lmnz9AbYcjJnc8EPFNBmVCiJhgMin+64OzMZsUX31s77u2RBqNXpeXv22u5pHttWypbA9SypE5UN/Ng1tr+Nvm6nc1ONVa88j2Wh7fWccL+xsNyTYanf0eHthcxUPbathd0znq47x4oInHd9bxyPbasG63JUSoyaBMCBEzijKS+PfrprO9qpOHt9eO6Vi9Lt/p7vPtfe+fqQqHoZ0E/AH9rj0afQFN1+DPRmUbjW6nF69/YBA1ltztg4spuhzeUXf0FyISSU2ZECKm3LqghKd21fP/XjrmZdPyyBtlx/2ijCQumpBNe5+HiydmBzn8Cwsy8Tp8ZGcYGFcburpy61mElfMyOd4Sx/zSjMMYTYaZdnJLB6fRY/Ty5KK0Xf0v2xaPrtru5iUlzrqjv5CRKKwdvRXSi0B/gfwAzu0119SSn0NuBGoBu7WwvPdwzpc6+EuJBTbflc+Yv1XD4tn99+ZP6wbiMd/YUQoRDJHf2rgUu01iuAPKXUCmCN1no5sA+4Kcx5hBAxaHxOCv986SRe2N/I2kPNYz6e0+Onpn14RfW9Li+1HY6lqHXq7PcYvIAhmEbyPIyFxxegpl0WEojwC+ugTGvdpLUeqbl1AbOBdYM/rwWWHjOPECJ2fWpFBVPy7XznmQP0uX2jPo4/oHloWw1P7KrpQNN572u0+PnH1treHxnHeuOtY76PoOhtdfNA1sGFirsqe0yNEswBM54Hl68wPMwVs/ubeCJXXU8trMupPcjxHsZcjJeKTUbyAG6gJ7Bi7uBs3ZdVERdq5TaoZTa0dpq7AudECI6JfHM/OiWWTR0u/jV68dHfRxfIECva2BQ13mW1hRncnr9p7vRd/YbW4Df43pn14EL5Y4GvoB+53ki8b/t0PG7+j1jXsUrxEiEfVCmlMoCfGn8koFB2dAmcWmDP7+P1vo+rfVCrFXC3NzccMQUQsSABWWZ3L6olD9vPMXx5tRHSPRYubKmfIMzrdz+fT88143KyWBNVPzmFJgZ9Vky1+rKnJSWDI+i+IFaSwuH31RfaRIsJiG/TyM1VUZC5icb+fqWYWYTCqk9yXEmcJd6G8BngW+p7XeqpTKA/6itb5WKfV1oEpr/ej5jiGF/kKlkejo97Dmv9cxvTCNBz+15PQm4u8lh5CiFCI5EL/W4FFwE+UUuuACcB6pdRGYC7wdJjzCCFiXFZKA1+7cgqbK9t5bt/IGq12OTzUtEdG0b7RAGfNdXs/3c53Fsi39rpN3AhgdaDmRwDOy+09Lpo63NT1+klyvEDAUIV27sf80g0dtp6Rnb7gUivoS1T5nW+iHgofdcvBn4SThzCCHiy4cXj+Ph7TU8u6eBG+YUDes23U4vf99SjdevWTI+i2UTc0KcMrKtP97K7pouEq0m7rqonB6Xl0e31xHQmsun5zOzOD3smTaeaGNHVSddTg/JVjPegEZrTWqilUum5JfnjD3c3jrWyp7aged897JykhOG/5Z5rLmXF/Y1ohTcPK/4rHumCvFe0jxWCBHzzCbFn+9eRHZK4rBv4/D4Tnef7xrl

TEksGZotcnsDOL1+epw+AoMziKdSQpWpj6XD6vJhNsXwB8IkJpoDcpz1uUcKPh3ewO4vAGSE0aeTWvocY5  
+9a+ILzIoE0LEhTz7yDr7F6YnsXJyLu19bpZOMKajfyRZNTkXm9VMQZqNnNREspITWDI+C5fPz4Kysy6cD7k  
Vk3Kxmk0sm5iNw+3HYllopxD7/EFZ3LB6ch5brR0UZdjIShnBiAyYU5JBn9uHxaSYVmgfcxYRH8Ja6B8MUug  
vhAgFKfQXQoRCJBf6CyGECIJ+t48TLX2nN003QkuPi9qO4BTvn01Tt2vERfstvS5q2kOXSyhQktOXQggRZQI  
BzcPba+lxeinLTuaW+SVhz9DQ5eTRHbVoPbBB+Kys4Bb617Q7eHJ3HVOP9A2bVph2wds097h4eFstAa1ZPSW  
XeeOMOA0qxGjJTJkQQkSZgNb0D24d1WNQkX2f28dQ9UuvK/gZelzeM44/vEL5Xtc7iw+GexshIonMlAkhRJSx  
mElcO7uQky19zC7JMCTDpLxULpqQjcvrZ0F58GekphWm0eP04gto5g6ztcWE3BSWTcjG4fWzeHz072Ig4o8My  
oQQIgpNyE1lQm6qYfevlGJpRehWpZpNasS94ZRSALhJfCTU5fCiGEEJEgJANypRSRUqpXUop1+Cel0OXf0  
ApVXvGzx9RSmlSSj2vILPwJacQQgghRAwK5UxZB3ApsOU9l38QqAVQSlmBzwArgQeAT4cwjxBCCCFExAr  
ZoExr7dJad555mVLqWuA1YKixzmRgv9baB6wFloYqjxBCCCFEJA3of9dwEeBTwz+nAH0DH7fDZx1CY9S6l7  
gXoCUIBQWLhxWY1whhBi2qqoqeW0RQoTC/OfEMWyDMqXUJcAmrbVHKTv0cRcwVEeWNvjz+2it7wPuA9  
lmSQgRGrLnkhAiFJRSu4Z73XCuvpwJ3KCUehmYoZT6IXAMmKmUMgOX8f76MyGEEEEKIuBCymbLBlv6Xg  
DnAK8C/aa1/Nfh3G7XW3x78/g/ABqATuCNUEYQQQgghlInlBmVaay8Ds19n+7vLZ3z/AAMrL8UIODw+qtoclG  
YlYbdZz3k9f0BzvKWXRJQE8uy2MCYUQgghxEhIR/8o9dTuelp63KQlWfink8vHnvN5bx1rYW9uNxaT42LJy0pP  
OPYATQgghhHGko3+UcnkHuoq4fX700K6957meL6Dx+gPnvJ4QQgghjCUzZVHq+tmFHGrsYVK+nTNWs77Pq  
sm5pCZayEtLJCc1MYwJhRBCCDESMiiLUnlpNvLSLlwljlpJoYeXk3DAkEkIilcRYyOILIYQQQogIIIMyIYQQQ  
ogIIKcvhRBCRD1/QLO9qoPtpzo42txLe58HpSAzOYFphXaWVmSzoCzzvDW4QhhNBmVCCCGiVme/hz9tPMV  
jO2tp7nGjFJRlJZNRt0RrONDQzQv7GwEozUrikxep58NLxpFoMRucXlJ3k0GZEEKIqOP1B7hvfSW/W3eSfo+PS  
6bk8e/XFbNiUu77+jF2O7y8fqSZh7bV8N3nDvHHjaf4rw/OZtmEHIPSC3F2MigTQggRVQ419PCVx/ZyuLGHK6  
bn89UrpzA5337O66cnW7llfgk3zytm44k2vvPMQe74w1Y+t3oCX7liCiaTnNIUkUEGZUIIIaLGU7vr+MYT+0lPsn  
LfnQu4YkbBsG+rlGLFpFxe+OJyvv/clf533UIOtfxZp7fNxxWaV05nCeDIOE0IIEfECAC2PXz7CfesrWTI+i99+ZP6o  
G2InJlJ4f7fMYmJeKj984TAu707+784FUmcMDCctMYQQQkQ0nz/A15/Yx33rK7lzaRl/v2fJmHcoUUpxz4oKfnT  
zLN482sq/PLyHQODcW9YJEQ4yKBNCCBGxP4AX3x4N4/vrONfLpvE92+cgdUcvLeuO5aM49vXTuOlA038z9  
pjQTuuEKMhpy+FEEJEpEBA8/Un9vHi/ia+fe007lIREZL7+eTy8Rxx7uPXb5xgSoGd62YXheR+hLgQmSkTQggR  
cbTWfP/5Qzy5q54vXz45ZAMyGDiV+YObZjJ/XABffGI/dZ2OkN2XEOcgzIhhBAR5771lfx1UxWfXD6eL1wyM  
eT3l2Ax8cvb56GBrzy6F7/UlwkDyKBMCCFERHn9cDM/fvkl184q5FvXTAvb1kilWcl894YZbD3VwV/ePhWW+  
xTiTDIoE0IIEtGONvXyxYd2M6Mojf++dU7YG7t+YH4xl07N4+evHaOx2xnW+xZCBmVCCCEiQme/h3v+tp3k  
RAt/+NhCkhLC3zdMKcV3b5hBQGt+8PyhsN+/iG8yKBNCCGG4QEDzlcF20tzt5r47F1CYnmRYltKsZL5wySRe3  
N/E+mOthuUQ8UcGZUIIIQz3x42VvHGkhX+7ZirzxmUaHYd7VoxnXFYy++lI9JUVoSNDMrigKwiGqC1ln8LIS  
LQzuoOfvLyUa6aUcBdy8qNjgNAosXMV66YzOHGHp7b12B0HBEEnZFAW4w419PCbN07w9y3VeHwBo+MYx  
uXlc/+mKn775glOtPQaHUcImaiz38MXHtxNUYANn3xwdthWWg7H9bOLmFGUxk9fOYrb5zc6jogDMiiLccea  
wloTWuvM/Z+t9FxDNPU7aLT4cUf0Bxv7jM6jhCCgdnrf3tqP619bn57x3zSk6xGR3oXk0nx9aumUtpf5lmd9UbHE  
XFABmUxbm5pBnabhQl5qeTZbUbHMUxxZhJl2cmkJVmZXZphdBwhBPDU7npeOtDEly6fzOySDKPjnNXKST  
nMKc3g/946ic8fv2cbRHjI3pcxrjwnJaTbk0QLq9nELfNLjI4hhBhU3+XkP545yMKyTD69coLRcc5JKcU/rZ7AvQ/s  
5Pl9jdW0r9joSCKGyUyZEEKIsAoENF99dC8Brfn5h+ZiDnOD2JG6bFo+U/Lt/PbNE7ISU4SUDMqEEKEIv82V  
bG5sp1/v24647KTjY5zQSaT4nNrJnC8pY+1h5uNjiNimAzKhBBChM2JlI5+8vIRLpuWx22LSo2OM2zXzqiqKN3  
GXzdVGR1FxDAlZlAkhAgLf0Dz1cf2kZJg5v/dElntLy7EYjZx50XlbDrZztEmaasjQkMGZUIIIcLir5uq2FPbxXdv  
mEGuPdHoOCN2+6JSEi0m7t9cZXQUEaNkUCAEECLkatod/PerR7lkah43zCkyOs6oZKYkcNPcYp7aVU+3w2t0H  
BGDZFAmhBAipLTWfOPJfZhNiv+8eWZUnbZ8r7uWleP0+nlsZ63RUUQMctmgTCIVpJTApZRYKaUsSqnXsqk  
NSqn1SsqHIVLmwet9RCm1SSn1vFIqLVR5hBBCGOOR7bVsOtnON6+ZSmF6ktFxxmR6URrx2Xw8PZatJb2G  
CK4QjIT1gFcCmwZ/LkLuF5rvRI4BVyjlLICnwFWAg8Anw5hHjEKLb0u6jodRscIq363j8rWPrzSvVuIMWvqdvGf  
LxxmaUUWH140zug4QXHbolJotPSxq6bL6CgixRsUKa1dmmtO8/4uVNr3TX4ow/wA5OB/VprH7AWWBqqPG  
LkGrqcPLi1hsd21HGgvtvoOGERCGge2lbDM3saeHF/o9FxfhIhqWmu+/fQBvIEAP75lNqYIbXl7XNfOLi5wcyj2+  
UUpgiusNeUKaWKgMuAV4EMoGfwr7qBzHPc5l6l1A6l1I7W1taw5BTQ4/IyNDvf44yPolZfQNPv9gPQHSePWY  
hQeX5fl2sPN/OVy6dQnpNidJygSU20cN3sQp7f10C/22d0HBFdWjooU0olAvcDnxqcHesChurI0gZ/fh+t9X1a64Va  
64W5ubnhiCqAyXl2lIRkMbc0g/llZx0vx5wEi4lrZxcwrTCNK2cUGB1HiKjV0e/hu88eZE5JOH+/uNzoOEF326JS+j  
1+XtgnM+oieMK9IfI9wP9qrQ8N/nwMmDIY9H8Z79SfiQhgMimWTcgxOkbYTcyzMzHPbnQMlaLaj148TLfTyZ8  
+tQSLOfYW+s8fl8mE3BQe2VHLh6JoZwIR2UK5+tKqfLoLzAFeUUqtAm4B/lkptU4pdpPW2gv8AdgA3AX8PIR  
5hBBChMfmk+08vrOOe1dWMLUGNhfvK6X44IJSdlZ3UtMeX4uhROiEbKZscMB12Xsuft/0g9b6AAQZXgohhI  
hybp+fbz29n9KsJL5wySSj44TU9XMK+cnLR3huXwP/tGai0XFEDI9OWUhhBCG+b91lVS29vODG2eSIGA2Ok  
5lIWQms6g8k6d310vPMhEUMigTQggRFJWtffz2zRNcP6eI1VPyjI4TFjfmLeZ4Sx9HZJNyEQQyKBNCCDFmQz  
3JEq0m/v26aUbHCZtrZxViMSme2dNgdBQRA2RQJoQQYsy2l3PppPt/OtVU8mz24yOEzZZKQmsmJTDc3sbCA  
TkFKYYGxmUCSGEGJPOfg8/fOEw88ZlcMfi2NhKaSRUnFtMfZeTnTWdF76yEOchgzlhhBBj8uOXjtdt9PKjm2f  
FzFZKI3H59HxsVhPPyilMMUYyKBNCCDFqWyvbeWRHLfesGM+0wtjsSXYhKYkWVv/O45WDTXIKU4yJD  
MqEEEEKMiscX4FtPH6AkM4l/vjS2e5JdyNWzCmjpdBo7Vk5hitGTQVkc8gc0++q6ONnaf9L7ae1ls7O6k16XbO



wtRCy6b/1JTrT08YObZpKcEO5d+yLLJVPzSDCbeGl/k9FRRBSTQVkc2nqqndcPt/DsngbqOkOzPYg/oHl8Zx3rj7  
XyvGzYK0TMqWl38Os3TnDtrELWxEIPsvOx26wsn5TDywebpJGsGDUZIMWhM18vQvXaobUmMHjwgLxAC  
RFzvvcfQSwmxb9fN93oKBHjqhkF1HU6OdjQY3QUEaXie745Ti0Zn4XNasZus1CalRyS+7CYTXxwQQIVbf1Mj  
dPiXyFi1WuHmnn9SAvfumYaBenx05PsQi6fno/5KcVLBxqZWZxudBwRhWSmLA5ZzCYWlGUyOf99+8MHVX  
6ajSUV2aQnWUN6P0KI8HF6/Hz32YNMzk/17ovLjY4TUTJTElhakeXLB6SuTIyODMqEEEIM2/+uO0F915Pv3zg  
Tq1neQt7rqhkFnGzt53iz7IUpRk7+RwkhbBiWU239/P6tSm6eV8zSimyj40SkK2cUoBQyWyZGRQZlQgghLkhrzX  
88e5BEi4lvXjPV6DgRKy/NxpySDNYeaTE6iohCMigTQghxQa8cbGL9sVa+fMXkuNpwfDQum5bH3touWnpdRk  
cRUUYGZUIIIc7L4fHx/ecOMa0wjTuXlhkdJ+JdMjUfgDdltkyMkAzKhBBCnNev3zhBQ7eLH9w4A4sU91/QtEI7  
Rek21h6WQZkYGfntJYQQ4pxOtPTxxw2VfHBBcQvLs4yOExWUUIw6LZ+Nx9tweflGxxFRRAZlQgghzmqguP  
8ASVYz37haivtH4tJpeTi9fjafbDc6iogiMigTQghxVi/sb+TtE+187cop5KQmGh0nqiytyCY5wczaw81GRxRRRAZl  
Qggh3qff7eOHxz9mZnEadyR4v6RslNnrJiUwxtHwMSDejFsMigTQgixPr998wRNPS6+d8NMzCZldJyodOm0fB  
q7XbJBuRg2GZQJIYR4l6q2fv644RS3zC9mQVmm0XGIlVT81AKXpdVmGKYZFAmhBDiXb7//CESLCYp7h+j  
nNRE5pZm8PoRqSsTwyODMiGEEKe9caSZN4608M+XTpLO/UFw6dQ89tV109rrNjqKiAlyKBNCAGAG2+fn+8  
8dYkJuCnctKzc6TkxYPSUPgPXHWg1OIqKBDMqEEEIA8McNp6hqD/DdG2aQYJG3h2CYXphGTmoC62RQJoZ  
B/tfFgaZuF/1u3wWv19HvobPfc8Hr9b19NPeEfQpd4eYWQoxdY7eT37xxgitn5LNUq7RcWKGyaRYOTmXDcdb8  
QekNYY4PxmUxbhNJ9t4aFsND2ypPu8Ap7K1j79truJvm6up7XCc83rdTi/3b6riwa017KzuCEVkaDadeCe3wyMD  
MyFC7T9fOEaAa7597XSJo8Sc1VPy6HJ42VvXZXQUEEfKUBbj2voGZr6cHj995xmUtdf70BoCWtN+ntmyHqcXj  
y8AENLC1da+gWM7PX76XDIoEyKUNp9s5/19jXxm1QRKs5KNjhNzVkzMwaRg3VE5hSnOz2J0ABFayfymAJB  
nTyQ/7dwrqWYVp9PR78GkFNML0855vZLMJBaPz6Kj38NFFTlBzztk+cQclFLk2xPJO09uIcTY+PwBvvcQYoz  
kvjs6glGx4lJmSkJzCnN4K2jLXz58slGxxERTAZIMS4rJYEb5hRd8Ho2q5krZxRc8HpKKS6eGLrB2JDs1MRh5RZ  
CjM3ft1RzpKmX//vofGxWs9FxYtbqyXn84vVjtPe5yZZ9RMU5hOz0pVKqSCm1SynlUkpZBi/7mlJqo1LqH0op6+  
BIH1FKbVJKPa+UOvcUTRzwBzRP7qrjV68fZ39dt9FxRARwef08sKwa3755gqq2fqPjiBjT3ufm568dY/nEnGF9K  
BOjt3pKLrD+uNyClOcWyhryjqAS4EtAEqpXGCN1no5sA+4aXBg9hlGJfAA8OkQ5ol4PU4v1e0O/AHNwQYZlA  
lo7HbR1uvG4wtwpEn2zxPB9dNXjuLw+PnuDdNRSva3DKVZxelkpSTwltSVifMI2aBma+3SWneecFiYN3g92u  
BpcBkYL/W2nfGZXErPcnKhLxUEq0mZpdkGB1HRICiDBuF6TaSE8zMKEo3Oo6IlfvqunhkRy0fv7iciXl2o+PEP  
JNJJsXJSDuuPtxGQ1hjiHMJZU5YBDH3U7wYyz3FZ3DKZINRRiXdJtJi5ffe4o2OIGBMiAL7zzEFyUhP54qWTjI4  
TN1ZPyePpPQ3sq+9mbmmG0XFEBApnS4wuYKhmLG3w57Nd9j5KqXuVUjuUUjtaW2XqF+BESy+P76yL2dOc  
Lb0untxVx9sn2oyOIkTMeWJXHxtqu/jGVVOx26xGx4kbKyfnohSsO9pidBQRocI5KNsOrBr8/jlGas2OATOVUu  
YzLnsfrfV9WuuFWuuFubnSaRrg9cMt1HY4eP1wC1rH31T4phPtVLc72HaqIyy7BwgRL7odXn7y8hHmj8vg5nnFR  
seJK1kpCcwuyeAt2XJJnEMoV19alVJrgTnAK8B4YL1SaiMwF3haa+0F/gBsAO4Cfh+qPNGssrWPVw820dT9zuC  
kIH2gd1d+WmJMFugOPb7URAtPfwkf7Chm9cONdPluPD2UkLEgv965Qgd/R5+cNNMTKbYe+2IdKsn57Kntmt  
YW9qJ+BOymrLBAddl77l4K/CT91zvAQZWXoqz8PkDPL+vEX9A09Dl5O6LxwNw3ewi2vvdZCUNGJwwNJZW  
ZDMxL5XUREvleid1OTy8erAZgD63l5vnlYTkfSIFhtqu3hwWw13LyuXhSMGWTUll1++fpz1x1u5ca7MVIp3u  
+BMmVLqIqXUb5VS+5RSrUqpGqXUi0qpflJKyf/qEDObFCmJA2PnM2s/zCZFnt2GxXz+p9Dl9bO1sp3K1r6Q5q  
xpd7CIsj2o+1TmPCaGtJllosVMonXg38+eKHU1Irb5A5pvPbWfPHuidJU30JySDDKTrdIaQ5zVeWfKlFvAQ3AM  
8B/Ai2AjYFWFmuAZ5RSP9daPxxvqoPFFKcXti0pp7HYxbhR70q072srhxh6Ugo9dVE5WSvBnlnpdXp7aXU9Aa5  
p7XFHz6S8pwcxHFpfr1u+mPDvF6DhChNQDm6s42NDDb+6YJ8X9BjKbFCsm5bL+eCuBgJZTyOjDlnT68k6t9  
XuXv/UBuwa/fqaUCv2eO3EuJdHCxLzUud3WPPgfXqEI1f99kxo4dkC/c3/Rlj3ZSnqyvEGJ2NbS4+JnrX5jxaQcrp1  
VaHScuLdqci7P7m3gUGMPM4v1hJN4x3kHZWcZkI3qOsI4qybknmdPJDs1gYwQ1Z+IJFq4dWEpTT0uphZIE0oh  
Is0PXjiM2x/gBzfOjMmFQdFm5eSBLgLrjrbIoEy8y3Bqym5TSIUMfj9bKXVCKdWglPpA6OOJsUqwmJhTmkFJ5  
shPfY5EQbqNuaUZsqGxEBFm/bFWntvbWgdXTaA8R07TR4JceyIzi9OkNYZ4n+G0xPgaUD/4/Q+AfwYWAP8R  
qlBCCCHGrT/t45tP7qciN4XPrp5gdBxxhtWT89hV00W302t0FBFBzjsO0r9B1AM/KtS6nvAcmaRcC+QrpT6jlJqZ  
ehjCiGEGKmfvnKUhm4n//WB2TKLHWFWTcnFH9Cya4l4l/MOyrTW32Og634d4AJe0Vp/d/DyBq3197XW68O  
QUwghxAsjsqOrg/s1V3HVROQvLs4yOI95jXmkGdptFWmOIdxnO6cvPAFcC04CvACilpgMvhDCXiBBa65jcxkm  
8XyAgz3OscHn9fP2JfRRnJPG1K6cYHUechcVsYsWkHN461iqvseK0C3b011ofVkp9TGvtPuOyQ0qpptBGE0bre  
Xl5dHstbl+Am+YVU5yRZHqKESIH6rt5/XAL+WmJfHBBYQWbEovI9svXj1PZ2s8Dn1x8uvm0iDyrJufy4v4mjjb3  
MrUgzeg4IglM95X3SaXU6f/ZSqliC4LXQRBKRqobdQa/Lh8cX4ERLaHcEEMy60tRLQGsaU110yD6gUW17VQe  
/f+skty0sZcWkXKPjiPNYNTkPGGjyLQQMf1D2NPC4UsqslCpnYIPxb4YqlIgMFbkp5KfZSE+yMq1Q+o/Fsrml6a  
QkmpmQl0p2SqLRccQo9bq8fOmRPZRkIvPv1083Oo64gIJ0G1ML7FJXJk4b1ry21voPSqkEBgZn5cCntdabQphL  
RIDkBAat3LBlnDAwRBhPz7EzMK4F3tPvec4do6HLy2GcuIlVOW0aFVVNy+fPGU/S5ffKciQu2xPjy0BcDe16WA  
nuApYOXCSGEiAAv7W/k8Z11fH7NRBaUyWrLaLFqci5ev2aTtMYQXPj0pf2Mr1TgKeDEGZcJIYQwWF2ng28  
+tZ85Jel84dJJRscRI7CwLIuUBLN09xfAhfe+/F64ggghhBg5t8/PP/1jF36/5pe3z8MqK2ejSoLFxLKJOaw7OtAaQ/Y  
mjW8XOn15n1Jq5jn+LkUp9Qml1EdCE00IICSF/OcLh9lb181Pb50je1tGqdVTcnvcnKytd/oKMJgF6oq/F/gO0qpW  
cABoJWB2rJJQBrwZ+AfIU0ohBDirJ7ZU8/fNldz78oKrppZYHqCMUorBluxrDvawsS8VIPTCCNd6PTIHuBDSq  
IUYCFQCDiBw1rro6GPJ4QQ4mwON/bwzSf3s6g8U7r2R7nSrGQm5Kbw1rFW7lIRYXQcYaDhtsToA9aFNkr8ae

hy0unwMLUgDbNpZHUEfW4fla19lGWnkJ5kDVFCESpaa4639GExKSpyw/vJ2Onxc6Klj+LMJLJSEsJ63yI4Wnv  
d3HP/Duw2C7+5Y77UkcWA1VPyeGBLNU6Pn6QE2Tw+Xsn/ZIO09715bEcdrx5sZsPxka+6eXp3Pa8fbuGxHbUh  
SCdC7UB9Dy/sa+SZPQ2cbA3vbgkv7G9k7eFmHtlei9cfCOt9i7Fzef3c+8AOOvo9/OmuReSn2YyOJlJgleRcPL4A  
WyrbjY4iDCSDMoP4A5rA4Ca0Xv87m9H2urxUt/dfcHNojy9w+rbn28xWa01Nu4NupzclqQfsr+viuT31QTueP6Cp  
auun3+0L2jEjncfvf+d7X3gHRkMDMZ8/cPp3UEQHrTVfe3wfu2u6+J/b5jCzON3oSCJIFo/PwmY1SWuMODEi9sF  
KqRSttSwPCYK8NBvXzCqkvD/N/HGZwMAn4Ae31uDw+JlZnM710/PPefsb5hZxpLGXiXmp511C/faJdrZXdZBg  
MXHXsvIxd4w+VN/N5x/chS+g2VnTyXdvOOvi3BF55WATR5t6sdss3L2sPC42w55bmonWYDGbmFoQ3pZ/V88  
s4GBDD+Oykkm0yGmSaKG15scvH+G5vQ18/aopXDWz0OhIlohsVjMXVWSz7mgLMMPPoOMlgw3r3U0otU0od  
Ag4P/jxHKfW/IU0WB6YU2Fk2IQebdeCN0e0L4PAMzKB0XWBT6JzURJZPyqEg/fynLjoHj+PxBXCcMRPV7fT  
i8vrPdbNzqu5w4BucxWvoco349ufL2Of2vWvW8GxGmzvSmE2KheVZzC3NCHtfoozkBC6emENpVnJY71eMzf+  
uO8nv36rk10vG8dlVE4yOI0Jg9ZQ8qtodVLXJ3Ee8Gu60yf8AVwLPAmiT9yqlVoYsVZxKT7Jy+fr86jodLCwPzjY  
pKyfIYjWbyEtLJG+w9uRIUw8vH2giwWLijsXjyEgefrH31bMK2VHdQX2ni69eMTkoGS+fls/O6k7Kc1LOW+B6  
qKGHVw8N5P7I4jLSk2WBg4gP92+q4qevHOWmuUX84MaZ0mA0Rq2aPNAaY/3xVuk5F6eGfS5La137nheC6J  
+uiEAzi9ODWieSnmX9X/+i+k4nWoPbG6C11z2iQRnAt6+djj+gg3aaMS/NxtWzLnwppqHrjNx97ogdlPn8AcwmJ  
W+cIge31nHfzx7kMun5/PTW+dgGuFKbRE9ynNSKM9OZt3RVj52UbnRcYQBhjsq1VKLQO0UioB+CKDpzJF  
9FIQlkmXw0uqzcL4EX4ac3n9PLYthm6nj6tnFTA5P3z1UAvLM+l2erGPIne4HKjvZu3hZnJSE7ltUam0KhBj8vKB  
Rr7++F4unpjNrZ8sWyjFg1WTc3l0Rx0ur/90aYuIH8P9H/4Z4J+AYqAOmDv4s4hCGckJfGBBCVfOKBjxbFdzj4tO  
h5eA1hxvDm8rh6HcV8woGHFft3A52tSL1gN9pDr6z18XKMT5rD/Wyhce2s3c0gzuu3OhvEHHiVVTenF6/Wyv6j  
A6ijDAcJvHtgGyx6WgKCOJ8TkpdPR7mFMqy/Hfa35ZJp0ODwXpNnJTE42OI6LUjqo07n1gBxPz7Pzl7sWkjHH  
VtlgeSyuySbCYeOtoKysGt18S8WNY/9OVUuOBLwDlZ95Ga31DaGKJSGU1m7hpXrHRMSLW+JwU2SZFjMm  
B+m4+/pftFKUn8bdPLI7Y2kkRGskJFpaMz+KtY6182+gwIuyG+/HraeBPwHOAtAAXQogQONnax11/3obdZuGB  
e5aQa5fZ1ni0anluP3zhMPVdToozkoyOI8JouAVFLq31r7TWb2qt3xr6CmkyIYSIIw1dTU7841aUgr/fs0TejOPY6ik  
Dpy3fOird/ePNcAdlv1RK/YdS6iK11Pyhr5AmE0KIONHe5+ajf9pKr8vHXz++OOyb1IvIMiE3leKMpMHu/iKeDPf  
05SzgTuAS3jl9qQd/FkIIMUq9Li93/WUb9Z1OHvjkEtnPUqCUYs3UXJ7cVS+tMeLMcGfKbgYqtNartNZrBr8iek  
BW2drH+mOtQd2IO9J1O7ysP9YqW3QIESV8/gCf+8cujiT28ruPzmfx+ODs5CGi32XT8nF4/GyubDc6igij4Q7K9g  
IZY70zpVSyUuoFpdQ6pdQzSqlEpdTXIFblVL/UEoFZZlRn9vHc3sb2VndydpDzcE4ZFR45WATO6s7eXZvA06P  
bLggRKT7/vOH2HC8jf+8eSaXTM03Oo6IIEsrsklOMMfVe5gY/qAsHzilHpFKfXs0Nco7u8qYKvWejWwDbgdW  
KO1Xg7sA24axTHfx2JSWMwDzUUTrfHTAXvosVrMCIP8PGwhotL9m6r42+Zq71lZwW2LxhkdR0QYm9XMyk  
m5vH64Ba210XFEmaY3puw/gnR/J4EFg99nDN7/usGflwJ3AI+N9U5sVjO3LyqlqcfFpLzwbQNktKtmFnCipY/C9  
CQSLVKDIESk2nyyne89d5DLpuXzr1dNNTqOiFCXTc/n5YNNHgzokVrDODGs+ZTB9hdHAPvg1+FRtsQ4Dix  
RSh0EFgIngJ7Bv+sGMs9216XUvUqpHUqpHa2tw1sinJ2ayIyidBlS0T915PD4ONnah8d3/hZxiRYzM4rSyUoZ2QbjI  
jb0uQd+T7x+aSUyYVp73Xzx4d2UZ6fwi9vnRuyWYcJ4a6bkohS8Jqew48awRixKqQ8xcLrxVuBDwFal1AdHcX9  
3Aa9orWcALzAwU5Y2+HdpQNfZbqS1vk9rvVBrvTA3N762ndBa8/C2Wp7d08BzexuMjiMilD+geXhbDc/uaeDF/  
Y1GxxHn4A9ovvTIHnqcXn77kfmkYvZJ4jyyUxNZMC6T14/IoCxeDHca6VvAIq31XVrrjwGLgX8xf0pYGiX1TY  
Gtm1aNFjzZcCWURwzajk9ftr73Oe9jj+g6XP7AOhxvboS1OMLcKixG5fHF7Q8hxq7OdrUc+ErnoXb56e1992Pxes  
P0NLrlhB4px7CH9C09LrwBXE2p9vppdf17lW2fW4f3Y7Rrbw9W+5I5wsE6HcPLO7oiaMVx9Hm/946ycYTbXzv  
hhlMK0y78A1E3Lt0Wj4H6nto7HYaHUWEwXA/ppm01md2sWtn+AO6Mz0IPKKUuhPwArcBn1JKbQRqgF+M4  
phRqc/t4+9bqnF6/KyaksV8cWc9c4vFbOLa2YWcaOljdsk7NQU/fOEQR5t6GZeVzE9vnTPmPE/tquNnrX1DAd+6dj  
pXzSwY9m09vgB/31JDj9PLwvJMVkzKRWvNoztqaelxM7XAZtWzCgF4fl8Dla39FGcm8aGFpWPOXDxWzzN7  
GlAKPrighKKMJFp6XDy6oxZfQHPd7CIm5g2/EafWmke219La62ZaYdql/h2MlGgxc+3sAk629jOvNMPoOOIsjbb  
18oulx7h2ViG3LRr7776ID5dPz+MnLx9h7eEW7lxaZnQcEWLDHZS9rJR6BXho8OfbgJdGemda6y7gyvdc/JPBr5j  
h9QdQDAyozqWz33O6bUVjlwsGF19prXH7Au9qFjghN5UJ7+nwXdPhAKCu00kgEMB0nuWWbp8fq8mE6Ty1K  
/vru9Fao4H99V0jGoz0u32nZ2cau10AeP369MzZ0GUADV0D3zd1u9Bao9TY6mmaelwEtAYNLb3ugUFZrxuvX5  
++n5EMyJz+AG19Q7nf/ck00ps4TsyZMzGOFrZEE68/wFcf20uazcr3b5wx5t97ET8m5KZSlp3M64ebZVAWB4Y1  
KNNaf00pdQuwnIFTkPdpRZ8KabIo1dzj4vGddadnbvLstrNeryQziXnjMujo97C0YqBhpM8f4LGddTR1u1gxKYeF5  
eduJPnRJeN49VAzF0/MOe+A7EB9N2sPN5OVksBti0rPuSrzk8srqGzrx2JSfOLi8SN4xJCZksBFE7Kp63Ry8cRsA  
BlSjtZMyeNYcy8Lyt6ZBbx0Wh57a7uYXpQWIDemOSUZtPW5MSvFtMKBAcmUAjt1nU48/gBzx2WM6HiJfJO  
rp+RxxLn3Xf+rxxs4IBDz7tm/YQYrvvWV7K/vpvffWQ+2amyybgYPqUUI03L54HN1fS7faRIHWJMG9azq5QaD  
7yotX5y8OckpVS51roqlOGiUXW74/QqydoOxzkHZUopVkJJe9dlvS4fTYOzSida+s47KLtsegGXTb/wbNaJlJ60hv  
Y+Dx39HgrTz77JcXFmEn/9+OILHu9cIlZkv++yOaUZzHnPqBTJ+XYm5wdvNipcwex1s4vedZnVbBrTace5pRnM  
fU/uEy19ABxv6ePqUR9ZxKPaDge/ev0418wqkAG9GJXLpuXzp42n2HC8latmyu9QLBtuXdhjvLPnJYCfIPQTi0V  
TC+0UpNsoyrAxpWBkhhxpNgtOj58D9d1kJgenrUWOPYGDDd100jxkB+mY8WhpRTbpSVYumvD+wacQ5/Ofl  
xzGpBTfvna60VFEIfPyNkmazcJrh2SD8lg33HIQi9baM/SD1tqIJJ3+LNI51n58OLRdefucflISjAzszidTofnwjcYhrZ  
eDzOKBhYItDvOPVMmzm9BWea7TsMKMRzrj7Xy8sEmvnbIFloy5P+eGB2r2cQIU/N4/UgzXn8A63nqlUV0G+  
4z26qUumHoB6XUjQy0tBBBlJZkpSw7GbNJMb0oOMvls1PPmCmTprJxZ29tFw9sqWZvbZfRUeKOzx/g+88foiw

7mXtWjKxOU4j3umpmIV0OL1srOy58ZRG1hjtT9hngH0qp3zBQ6F8LfCkqkeKU2aS4ZX5JUFYIDuno9zC9cKCo  
vv08NWUiNm043orXr1l/rPV99X0itJ7aXc+Jlj5+95H5su2ZGLNVk3NJsp56UAjyyflGB1HhMhwt1k6qbVeCkwH  
pmutl2mtT4Q22tjsrunkmT31NPe4LnzLCBPM5fJl2SkopUhPso64Ts0f0Lx5pIWXDzTiCGKtWHE+5Tkp7/pThIfb5  
+cXa48zuyQ9anrdiciWlGDMkql5vHKwCX8UNbYWIZPc1ZeJwAcY6MBvGRo0aK2/H7JkY9Dj8rLu6MAemS6v  
n9sWja7GKxbMLc1gYl4qNovpvH3TzuZYcy97Bk97pSza5dNZFLp2ViG9k33YZRl9WD20tYb6Lic//sAs6Ukmg  
aqmQW8sL+RHVUdLDnLincR/Yb7Lv0McCPgA/rP+IpINosZu23gTSjnjJ5AtR0O3jzSQkuUzZ7VtA/m7h157j63jx  
1VHRxr7hvxbbNSEk5vlpXjj556NK01O6o62HSiLe4351ZKkWazysAgjJweP7958yQXVWSzfKJ8kBHBs2ZqHgk  
WEy8daDI6igiR4X58LtFaXxXSJEGUYDHxkSVldDo8FKYP9AnzBzTP7KnH69dUt/dz9wgbpBrF5w/w7N6B3DU  
dDu5aVj6i2791tJVjzb0A5KULvmuQeiH5aTY+dlEZXR8m1x49DS+PNfex4fjAOHSTSZ21h5oQofLYzlra+tz89o55  
MhgWQZWaaGHlpFxeOdJed66bft5dWkR0Gu5M2Sal1KyQJgmyPAQzRRlJp18UTYrTW+QkJYSu6FZrzcGGbg4  
19KD12M/7m5Qadm6Hx8fO6o7TDWgBkgdvYzEpEiwjX0adkZwQVQMyeOcxv/d7IULN6w/w+7cqWVCWyeLx  
527+LMRoXTOrgMZuF3vruoyOIkJguDNly4G7IVKnADeDKzC11np2yJIFmVKK2xaVUtftpZHwli54PNvTw2qH  
mwfuEaYVja21hMik+tKiU+mHkfvIAE9XtDqmxT0rKrBZzaycnEtxZhKZyQmk2axjyhItSrOSuXVhCR5fgIrc4e9  
7KcRYPbe3gfoup+XvKULm0mn5WM2Klw40MW+c9E6MNCmdlMXEzjJ2m5VphaEdmJw5ORaEiTJgoCft2jByn  
21Bjtmkgrqt0Uj0uX00dDkpy04+3RLA5fVT0+GgJDOJ5ISBXz+vP0BVWz/56bZRDZRrOxyYTIrIM5pzlmQmjzp3  
r8tLU7eLcWfkjX+gOZUWx/ZKYlkSj+7UQkENL9bd5Ip+XbWvGcLNSGCJT3JyrIJOxb0oJFvXj1VBv8x5ryDM  
qXU0Px7bxiyxISZxQMzYwOzZOEdDF09s4CDDT2UZCadPuVpleBA8/C2GnpdPsZlJfOBBSUAPLlroE1JVkrC6  
fq4lw40cbKlj+QEM59YPn5E3aqPNPXw0v6Boteb5hWPeRbUH9A8sr2WXpePsuxkbplfMqbjRYt1R1vYV9dNgsX  
E3cvKZdPjUdhwoo3jLX38/ENzpNZHhNTVMwv4xpP7OdjQw8zidKPjiCC60CvvTkAzclryvTRQEfREeaylx0Wq  
zXJ6hudslFLMKjHmP0lygplxWclJBt/mtKvNU6PH4D+M3qc9bsHvu9z+043yR26zOUN4PNrRjKe7HO9/9hjh3  
QOIZyB+F40aJv8LF6fAE8vgAp0VVGGbH+tmKnNRErptdZHQUEeOunFHA58+wLN7G2RQFmPOOyjTWkf  
HEsUw2Haqg7dPtJGUyObOpWUROZPw+uEW9td3k5Fs5c6lZSPuSxZMVrOJ6+YUcbKlj9lnDFKvnV3IoYYeph  
TYT0+7XzE9nz21XZRIJ494Ecac0gzcvGAmpeZcvwcDK3evm11IZWs/s0vj58Vu9ZQ87LYO8tNscvpyFGraHbxxtl  
UvrJk4qgU1QoxEZkoCqybn8tzeBr5x1VSZmY0hkTeyiFBNg73NnB4/PS5vRA7Kmgf7mHU5vLh9AUMHZQDjc1  
LedzqxMN1GRrKVpDOmw7JTE7loQja2UdRvWc0mLg5yL6iK3NS4WyCQnmTlkqn5RseIWg9sqcKsFB9ZWmZ  
0FBEbphbxOtHwtgujWRjSuSNLCLUxROy0VqTm5oYsftHrp6Sx7ZT7ZRLp0TkoBHglYPNHG7sYWJeKtfPGTj  
N8/aJNrad6qAow8atC0rlU5+IKk6Pn0e213LlZALy02xGxxFx4rJp+disJp7d2yCDshgi8+zDIJ2ay1lzi1kWwR26izOS  
uHleCfmJeJl0ZdvAzgKVrf2n+7hVtg5c1tDlwuH1G5ZNI4Zk89PS4fd11UbnQUEUdSEi1cPr2AF/c3xv3OJbHkv  
IMypVTW+b7CFVLEj0sn5JCvksDySdmna8qWVmStlZLAgRJMUIN0hk+Ic3lkRy2T81NZVB65H4ZEbLphThGd  
Di8bB3cwEdFPVl+KsJpTmsGc0ox3XTYp384kg3qpCTEWJl1p62V3TxbeumSb9okTYrZqcS3qSlWf3NrBmqvTGi  
wWy+tlg/oDmLYNNtPd7uGxaXsTWqY3FppNtHG/uY1F5FtOLxr4yMhrUdjYd7SFvDQbV0zPRylFS6+LVw82k  
5Zk5eqZBSPqwyYi22M76rCYFDfNKzY6iohDCRYTV88s4Lm9DTg9/pBulSjCY9jvDkqpTKXUYqXUyqGvUAa  
LdQ1dTo429dLW62ZndafRcYLO4wuwtbKDjn4PmyvbY4TnTtOddDW5+FQQw+tvW4Adtd00drr5mRLHzUdDo  
MTimDx+gM8saueNVPzom5/WBE7bphTRL/Hz9rDzUZHEUEwrEGZUuoEYD3wCvC9wT+/G7pYsS8nNZG0Jct  
KEdK9OI1iNStKswa2Oqrljb3Hdy5DjzUrJYH0wSa+FTkpmJQiNdEiq/NiyFtHW2nrc/OhhaVGRxFxbEIFNoXpNp  
7YVWd0FBEEw62q/mdgEbBfa71GKTWVgcGZGKWkBDN3LyvH6w8YviVSKCil+MD8Ypxe/3l3QIg188ZlMr  
UgjQSLCfnga49J+XY+nZWMxaQM7x0nguexnbXkpCawekqu0VFEHDObFLfML+Z3607S3OOSD35RbrjvEC6tt  
QtAKZWotT4CTAldrJHZV9fFm0daTm8VA3C0qZfXDzfT2e8xMnN5mU0qJgdkQ5RSMT0ga+tzs/ZQMyda+t51e  
VKC+fSAbIjNao67Admumk7ePNpyerutWNLe5+b1wy3cMr9EagSF4T64oJSAhq21xsdRYzRcF9N6pRSGcDTw  
GtKqWeAhlCFGommbhevH25hT20XG4+3AtDr8vLSgUb21XXLeXYRMq8ebGZ/fTcv7GvEJf3V3qW2w8FbR1v  
ZU9PF5srYW67/4oEmfAHNzVLgLyLA+JwUFpZl8tiO2tP9H0V0GtagTGt9s9a6S2v9XeDfgT8BN4Uw17AIWc1Y  
BmclhrrYW80mXF4/9V1OrGZZpm6Ulh4X20510030Gh0lJFISB2Y5kxJM75sZi3dJCWZMgy0iUmJwvS5PQ1M  
yktlaoG0chGR4YMLSjjZ2s+e2i6jo4gxGNarpVJq3Bk/nhr8swCoCXqiEUppTvlHJePodnoZnz1QZB3QGrNSJFnNS  
KNjY/j8AR7bWYfHF+BESx93LB134RtFmatnFllD3k9+uk1OYb1HTmoiH15SSp/LF3MLWRq7nWyr6uArl0+W3  
mQiYlw7u5DvPneQx3bWMS+Cd3UR5zfc5IXgOcH/3wdqAReCIWokepJTWRCburpPRMVCs3A4OysbW9FyC  
mlTs+UxOr7VoLFxKR8O2k2a0jvp7nHxfHm3qg7LZfnt1GRm3rBgUu3w8vhxh7cvndOAabf0uHhxXyMtPa7z3tbn  
D3C0qZf2PndQMg/H83sbAU7v3SpEJLDbrFw9s5Dn9jZIOUUUG9ZMmdZ61pk/K6XmA58OSaIgUAo0Gq8/gEl  
GZYYwmXqfXFBCTuc/k6Vb/6i19b15eFstAa1ZUpHFsgmRu/fqaHj9AR7eXoPD46c8J5mb55UA8N3nDtLe5+HF  
A4385o7557z9G0daONjQQ4LFxN3Lyk+XMITSc/samF2STnmMzQCK6PfBBSU8tbueVw8l4cN8alhKozrnorXex  
UCLjlkC2hsFguF6Un4LzC74PMHqO9yvutT+rk0djvftcJzJLodXlp6z/+pP9bk2hNZUJaFPcQzSubx+QLsrO6kyxG  
6Fb4ur39gxhdichWjP6Bx+wZqDJyed2oNHIOP1eHxveu69V3Od80COAE/9/oD+Pyhn0msautnX10318+WNzwReS  
6qyKY4I4mHtxleWSRGabg1ZV8+48f/3959x8d1loke/z0zGmnUJavLkiVb7nZclRDHaSbFISRxOkkgENiQhWUvC  
yx14d67Nwu7dy+wsMDCAoFdQguBEEIqjklI7MROXGPLdtXjW5Jl9a5RmflEP871lh2rz8yZGT3fz0cfj47PzHnO0  
ZRn3vK8DmAF0BSWiEiGLSmB9y8ppKatlxWII/etP1tVx4lmD3npSXzo0rJh93vjWAtvvT0C2+Xk/1V141o4u7Gzj8  
e31+APGNYuKpwySw7Fu29uOMjumg6yU1x8794VJCaeflxZSXYK1yzlp6PXy8Xl00L++HZzu5zcvLSYEy09LC



3JOrP9766Zw6YjTVw992wNsPX76jnc0EV2iosPryrH4RDWzM8nM7mN4qzkM8V6w+nZPdak8/cvKQr7sZQaL4  
dDuPeSUv5tw2GON/fE3XjOqWCsmcXQ/icf1tiyJ0MfTujMzk9ndv653WaBgKF7wEd6UsKZcS7NXVYrR2vPAIG  
AOTMu7XzNwTErfV4/3X2+cSVlrZ4B/AFzzuOo2GKMoav/3OdOXYfV8tne66W7z8e0tMSwHHvJkGQIHs3MT  
X3Xh8fyGdnvGqw8+Npp7/XiDQRlCjjJcLu4el7kFmJ+ttqOS8qnUZwVf2vVqvhwd2Up33npCL/ZVs1Xblxgdzhqn  
MY6pixk1ftF5MPARwAn8EHgPmAdeBJ4wBgTtvoJf3jrFDWtHhYVZ3D9okIARltYwJ7aduYXZgybkAFcPtsay5O  
XnkRh5vgqJs/JT+f0jD76vX4qy3VWTCx6Ya/VSjMrL5V1y6zaVA9cNpOnd59ieWl22BIyddY1C/LZebKNirw0khl  
iX3T5SEMxhu6+eq6RRE/tlJlZ/h5rqFBTyxo4bPXTc3rguUx6Oxdl/OBT4PlA+9jzHmveM5mIhMB64yxlwT/D0P  
WGOMuVxEvoRV++yJ8TzmWPn8AWqCi0GfbDm7KHR5buqYBuxmpyZecLaVMWbU2WVOh7BmmG/zY7n/Z  
EXiGJMVbTGeH8+Jlh7AnPPcWVmWzcoyTbIjpSQ7hZLSFNuO/+L+ekTghuAXOqWi1QffU8af9tWzfl89t2qB45g  
y1j64J4AfAT8FJjPaeC3gFJGXgbeB9cCrwf97CavVLCxJWYLTwRVzcyjQ38XKENVwqWn18MyeOIITndxVWTq  
umV/9Pj+/31lW88AnywuYnZ+WkhiOt+h+i427K8nLz2JO1ZG55IwO0+28fqRZspzU7hlabHtydnzVac52tjNpbO  
m8Z5ZOQCKJDrZcbKV1RU5tsam7LN+fz3LS7PI17UFVZS7rCKH8pwUHttarUIZjBnrJ7TPGPNDY8w2Y8zOwZ  
8JHK8ASay2IHmALKAz+H8dwAWzJRF5SER2iMiOpqaJzy+oLJ/G/ZeWhWyg/ZHGLgZ8Ado8Xmrbesd138bOf  
ho7+/H6DQfrO0e/wwQdON2JL2A43dEXtePZ3q7rIGAM7zT10GPzDMM+r5/DDV0EjGF/3dm/i2fAz5LpWXgGt  
BrxVFTT6mHfqU5uWKytZCr6WQP+Z7DtrCuHG7rsDkeNw1iTsmdF5G9EpEhEpg3+TOB4HcDG4O2/YHWHd  
mZIGUD7he5kjHnEGFNpjKnMy8s75//8AcOze+p4dPNxats8F7p72CwoyiDdnUBBhpsZ08bXrVKQ4aYkO5mUR  
CeLizPDFCFcVJJJcqKTspwU8tKSwnacyVhamkWSy8GConRSE+0d/+B2OVk8PZMkl4NIM7LObF8ejHH5kG3Kf  
sebe/jZ5uOs31cf1uK6G9621tBdq12XKkbcubKERKeDx7ZqeYxYMtb+to8E//3CkG0GmDXO420BPh68vQyoAT4  
AfAO4FnhznI9HXXsvRxu7AdhV3R7RMSdFmck8eMV4L4ElMcHBXZWIIY7o3Sry0qi4Kjxdo6GypCQrqmYYXr  
ewgOsWFpyz7bLZuVw2O74Kt8aDHSdaafd4afd4WVmWTV56eL54vLivnvmF6ZTlaIkBFRty0pK48aJCntxZy+fX  
zhtXxQBlN7EuSD7zAj/jzkaMMbuBXhF5Fav47OPAJhF5HStJ++N4HzMvPYnsFBcOEWbnRXfyEc16B/w8V1XH  
C3tPj6mQrrqwxS4+ntxZy+ajzXaHMiXMzk9DxHofyApTnbKmrn62n2zVVjIVcx5YPZOUfh+/31FjdyhqjMacOovIZ  
bx79uUvxntAY8znz9v09eDPhLhdTj68qhxfwISleOdUsfdUB0carBbHoky3Lmg7QZuPNVPd6qG61cOcgjTy03VQe  
Dgtn5HNwuIMep2OsEOQeelAA8ag48lUzFIWmsWKGvK8uuXEmYLLKrqNkySRkV8C/wZcjtXCdTFQGca4xsX  
hkJAmZM3d/by4v54Dp8M3AD/aFGQk4RDB6RAKdHbZhBVmWEVF05ISwrpQ+b5THWzYXx/WJZ5iRVKCM  
6wzdl/cX09ZTgrzC3UNVxV7Prp6JidaPLxyqNHuUNQYjLWlrBJYaMI5kjaKvHyggbr2Pg6c7mTGtJILHJst7KcV  
B5YXY5DiNu1KiNhVUUOcwrSSEtKCFvRxraeAf4cHHjuGfDrlPcw6uzzsvloMx9dPdP2Ui1KTcQNiwspszHDzs80  
nuGZBweh3ULYaa/PSPmDKtN0PtnAku5xRWdcrXDKTXZqQHUBuWIJYq2gnuRwkuaznZUZy/H9hsNMrBxvx+  
o2OJ1Mxy+V0cP+qMl4/2qzIMWLAWN/Rc4G3RWQbcKbY1THmlrBEZbPrFhYwtzCd/PQkHaemok5KYglfvKS  
Mlp5+ynU2YFgda+qxxliWZtkdiltDt8lM/jey0f42ebj/OvtS+wOR4lgrEnZw+EMItokOB1U6ExOfcUyU1xkhmm2  
oTrrc9fN5W+urtAB0iqmZacmcvuK6fxhlym+uHY+2am6Vm+0GmtJj1DfwAfcHd4Q1NKKfvpgs4qHnx09Uz6fQ  
F+8cZJu0NRixhZ35yILBORb4jICeCfGQNhqyCvP4A1S0e+rwj1+YyxIDb5qGzzuh47T2DFDf0Teh+8aqPq+f6hY  
PPv/ISxP5A4aaVg+eAV9Y4+nweDnVfu5yWN39PmpaPaNWgx/wWc+ToTXc/AFDdYuHXpuXhhrJheK+EGOsV0  
HXBJ/fkeALvlaj+XorFa3mFqRzzfx8fv7GCX0NRbERuy9FZC5wD3Av0AL8FhBjzJoIxBYRz+yuo7rVQ25alvevK  
h92vzeOtbD1eCuJCQ4+cln5uKojN3T28fi2GgLGcP2iAhaFcVmlaBEIGB7fVvk2bx8usvFTWLRt+huCG/fUcrO8i3Z  
3AA5eVkvCGyRUdHi+/fPMEXr/hijm5VJZPo8/r59dvnsQz4GfZjCzWzMsf9v5/fOsUp9p7Kcx0c+8lMwCrVMKh+  
i4ykl18ZFVZWOKerD/squV0Rx9FmW7uCcZ9IZuONLPrZBvJiU4+sqqcZJuXu7qQP+2r52hjN5nJLh64TGsuKTV  
en7i6grt+9Aa/21HdRy4rtzscdQGjfyocBK4BbjbGXG6M+Q8grlLs1h6rzlObx0sgMHxrSWuwHtSAL0B33/hadNo9  
XgLB1pjB48U7vzF09FrXabRzHry23f0+BkZpVZuoZj4vXr/1N2gJxtM74McT/MbY2j1yjIP3GXoug9u6hix2tBm8ti2  
j/Q16rPk71jUJb4vIRLWeud4+vAFdGF6p8bq4fBory7L5yWvvjNqDoewXWnPPHVgtZa+IyHqsZZHi6uvp2kWFVJ  
1qZ35h+ojfvC+fnYtThNz0JAozx1dedU5+GpXl2fQO+Kksm8g67rHH5XRww+JCjjR2sWyUmWvXzC9g58k2ynJ  
SSEkMT4mHkuxkVIXk0O4ZYFVFDmANfr16Xh6nO/q4ZOBI5f3LS5kX10HC4oyzmy7dkE+u062U56bEpUtSw  
DvW1zE/roOfg6J+0KumptPUkILhZlucqJ04frrFhawq7qNirw0khiK83orFe0+cVUFH//FDp7fe3rEHgxlDxILPVgRS  
QVuxerGfC/wc+ApY8yGsEZ3AZWVlWbHjh2RPqxSKs5VVlai7y0q3gUChrXf2USC08ELn75ciyJHgIjsNMaMaR  
Wksc6+7DHG/NoYcxNQAUwG/mHiISqllFIq0hwO4aErZ3HgdCcbDzfZHY46z7hHJhtjWo0xPzbGvDccASmllFIqf  
NYtm05RppsfnrM7lDUeaJvuphSSimlwiYxwcGDV8xi6/FWtp9otTscNYQmZUoppdQUc98lM8hNS+K7Lx2xOx  
Q1hCZISqkLOtbUzZeerOKbLx5iwKft55WKJ8mJTv76ylm8frSZnSe1tSxaaFKmlLqg3++o5URzDztOtLL1eIvd4Si  
lQuyDI84gJzWR77581O5QVJAmZUqd51B9Fy/sPc3pj7Rd45j84vSAesb9ay8VJumZw+r5+XDzTw2pEm/CMUi  
VZqKklJTODjV85i0+Em3qpuszscxejFY5WaUgZ8AdbvqydgDE1d/VN6KZJ1y6azoiyb9KQEsllS7Q5nUnZVt1FV  
2wFATmoSC4tHLqar1FRx/6VIPLLPbH778hEe/egldocz5cVFS9mh+i62HG+S1HxE809bD7aHNULLEdC74CfLU  
ebOdLQZXcoYeEPGHacaOWt6rZRFxUfiwSHkO62vqtkpbgm/XixrjQ7ZcIj2dt1nbxxrGXUxdAno6ffx+ajzRxr6h5  
xv+zgOYhApv5dlTojNSmBB6+YyauHmthd0253OFNezLeUNXB18cLe04C1Jt7aRYX09Pt4encdAWOo7+jjjpUIN  
kdpnlcPNXKwvgsRuD81MWqX0JmoPbXtvHakGYCkBOekW0AcDuHeS2bQ0NIHSXZyKEKckmrBPLy4vx6Afp  
+fq0dY7H0yXj7YyLHGbkTgo6tnkpl84YRrQVEGWSkuXE4HuXH2GlBqsj68qpxHNr3D914+wn8/cLHd4UxpMd  
9S5nl4cASXiUhMsE7H6RASnOdum6oGz98hQolj/q5FovPsOSUmhGa5kOREJ+W5qSQ4o+N61bR62F/XEVNjoR

KdDgZXb0kM0XXsHfBTVDt+zqLwicHXuVME5whr1wIUZSZrQqbUBaQlJfDxK2bxl4ON7NKxZbaK+Zay7NRE  
7qosoc0zwLwCa2Cy2+Xk7spSGjr7mFOQZnOE9rpqbh4FGW5y05Listtm8fRMkhIcOB3CrLz4+1s3dvXx5K5ajIG2  
Hi+Xz8m1O6Qxyc9wc8eKErr6fMwvTA/JYz5bVceptl7cLicfv2ImCU4H751fQE12CvnpSaQlxfzbmVK2eeCycn62+  
Tj/9uIhHvv4pXaHM2VFR1PAJBVNjBoOPOclg23y0FaUkJUtw61ewaobvGEZCzUcBKcDhZPz6Qw0x22Y9htTk  
F63CRkjV191LWfnfXp8xsGnx7eQGzVCiudlsLC4gwco7RgjZXPb10IfyDA4CsmMcF6fudnjP78rmvvpamrPySxK  
BVvUpMS+NSa2Ww51sLmo812hzNlxeVXY36fn8e2VuMZ8LowOIO1iwrtDuldOjxefvXmSbx+w6WzclhVkWN3  
SMpmtW0efr/TahW7YXEhC4oyKM5K5n0XFdLu8bJ8RpbdlrpxosK2V/XSVlOCq5xdom+XdfJi/vrEYG7KkuZnq  
XjBZU6333vmcFPNr3DN148xB8rchAJzRcqNXbR24w0CQO+AL3BmZgdnuicfdkz4MMb/Obf0Tswyt5qKujo9Z5  
pFWsf8rydX5jBpbNySEpw2hRZdMhKSWT17FxKslPGfd+OXut6GhO97wlK2S0pwcInrp3Lnp2NrzdYHc4U1Jct  
pSlu11cu6CA2jYPlExT7A7ngoqzkrybi4t3QPaSqYAK/lq6/Hi9QdYUZZldzhxZfmMLPq8flxOR8jGuCkVj25fMZ0  
fbTzGtzYc4toFBaNOoFGhFZdJGVgDwBdPz7Q7jBGtLivOhFGFns8fwCEy4vgqp0NiZiB/tPP5Azgdcqb7xe1ysmZ  
+eMpyKBVPEpwOPnf9XP72sbd4Zs8pbbs+dUtK2SEuuy+ViibHmrr5z1eP8bMtJ+jp99kdTtw72thlXe/NJ+gdCF/h  
WqXi1Y2Li1hYIMG3/3yYAV9sTTCKdZqUKRVMrxu78QcMnb1eTnf02R1O3DvSYF3vj14v9Z16vZUaL4dD+M  
IN86hp7eXXW0/aHc6UoklZHNl3qoMfbTzG+n31YS2zEe9ePtDAjzYem9CSI0cbu3lk0zGe3n3qTLHXpSVZZKe4  
mDEthRnTxj9IPRKONnbx443nxh2rlpZmkZXiojw3RVdlUGqCrp6bx+rZOxz35SNnJsQo8NOkLI68VdNO74CfA6  
c76dZusgnp8/ppqu2gd8DPrpPjr2y9p6adnn4/7zT10Nxt1cQqzHTzwOqZ3LGyJGpXmNhd04FnwIq7pSe2a3kVZyX  
z0dUzuW15ybhLZyilLCLCV25cQEevlx+8ctTucKYMfcekIwuL0hGB8twUUhOHn8PR1jPA73bU8MLE0/j8OI5g  
KLfLyez8NESs9RJHcqypm8e2VvPmOy1nts0vSschQmGmm2mpE1vI2w7zC624izLdTJvgAuRKqfiyqDiTO1aU8O  
jmE9S0euwOZ0q129mXU9HKsmksL80etYL6ruo2TrVZVePn5Kcxp0BLBAx189JiAgEz6nXcflSZlu4BGjr7WFK  
SSUpiAouKM11QGLOq9pGyeHomC4tiL26lVHh9/vp5PFdVx9fXH+T7962wO5y4py1lcWYsH6ql01IQsVqF8tPjd/  
mlyRjrdQQoyHDjHILYNVYTm1iNWykVPoWZbh66YhbPVZ3WxcojwJaWMhH5HHC7MeZyEfKCsA44CTxgjN  
ERhWE2tyCd4qkXE6Z8IXiJ2PNvHxWzMgmLSlBExqlVNz666sq+M32Gv75ubd58pOX6fJLYRTxljIRSQKWB  
m/nAWuMMZcDVcCtkY7HTvvrOnj9SPOotZRqWj1sPNx0ZuD4cNo9A2w83MTx5p5Rj52WIKAJWQhkJrvOqXj  
d3e/jtSNNHG7osjEqpZQKndSkBP7+urnsqm7n+b2n7Q4nrtnRffkg8PPg7UuAV4O3XwIutSEeWzR09rFhfwPbT7T  
y2pGmYffzBwxP7z7FrpNtvDDKi2HD/gZ2nWzj2T11WjTTJq8cbGTHCetv1daja5oqpeLDXZWILCjK4P8+fwDPg  
M7uD5eIJmUi4gKuMsb8JbgpC+gM3u4Asoe530MiskNEdjQ1DZ/ARFogYNh3qoN3mrrHfV+X04FjyBIwwxE406  
LIHqVlyyFQ39FLv8+PQ0cLhLSHx8uu6jbaPWcTLc+Ajl3VbTR2nS1QOvi3dlQ4NQmfqVUfHA6hK+uW0RdRx/  
f/4uWyAiXSI8pux94bMjv7cD04O2M4O/vYox5BHgEoLKyMmoqW2470cobx6xyCHdVilCSPfbCoNNSE7nnkILa  
PAPMzR9+9qPDIdx9cSm1bR5m5aaN+JgGg9PhQBC0dmxoPfVWLW0eL7tOtvHgFbMAeGFvPTWtHhITHHz8il  
kkJjhYMy+PkuxkctISSXe7b15aKaVC5+Lyady+fDo/ee0d7lxZwqy8kT+T1PhFuj1lHvBJEVkPLAIqgauC/3ct8GaE4  
5mUQHAp154BH4EJIPsqyHAzfwzIEzKTXSwqziQ5ceSWMhEHeelJJEVpgdJY5g8muYEh2W4gWPneGIPBup3gd  
LCgKENntY5DV5+Xo43deLVmnlJR7x9unl87wcnDz76tK8eEQRbyowxXxq8LSKvG2P+SUS+JCKvA9XAdyIZ  
z2QlJjo6feRlOakyWV/IrR2UQH76zopyU4esUtUjd+ty4o53NDN7Pyz3wxvuKiQt+s6KZ2WopMmJsJnD/D4thq6+  
33Myktl3bLpo99JKWWb/HQ3n7luL1977m1e3N/ADYsL7Q4prtiWSQRnXGKM+box5nJzH3GmLCOjPb6AzR29  
p1p4ZisngE/xVIWV1U0LGuU7nZx6ayccXWjToRnwEfLKDNB401OWhKrKnLIS086sy0jeL2nZ+n6ihPICxg8wU  
kpXX32v4aUUqP7yKoy5hWk87Xn3tZJZSFmf/NOBD2xo5Zfb63mhX2hmdL7nnpTWDYji9Wzc5mVmxqSx4x2X  
X1efr7lJL944yRvaSFBNUlul5MbLypkUXEG1y8qsDscpdQYJDgdfHXdlk619/LtPx+yO5y4MmWWWfIHZJLcvU  
dfaPsPTZul5M18/JD8lixot3jpc9rfTMK1XVUU9ucgnRd6kupGPOeWTnce8kM/uv149y0pJilpVl2hxQXpkxLmdMh  
XLuggNjPKayZP7USqVAqyU5m+YwsZuam8p5ZOxHaHo5RSyZfVnE+eelJfOnJKp2oEYJTJikDa9HIO1eWUKHT  
eCdMRLh6Xj63Lp/OtNREu8NRSillkwy3i6+tW8zB+i5+vPGY3eHEhSmVlCmllFIqdK5fVMj7lxTxvZePcrRRl5eb  
LE3KlIFJKKTVhD9+8iNqkJ5/97R7txpwkTepi1L5THfzyjRPsPNlqdygR09zdz2+2VfNeVZ2+8JVSkkkrkpSfxr7cvY  
e+pDr770hG7w4lpmpTFqNeONNPcPcBrR5qnTFXlt6rbqe/o40hDNydbeuwORymlVNANiww5QGUp//nqUbafmD  
qNBaEWF0nZlqPNPLunjrae8deePd7cw9O7T3GwvnP0naPlzGBdtPKcVESmxsLX5TkpOERISXSSn2H/Mka1bR6  
e3n2Kqtr2kDyezX/g5QMNrN93WgsyKqVizj/evJDSaSl89re76erz2h1OTIr5OmV17b1sPW5l5Q4R3r+kaFz3f+ntBrr  
7fZxs8TA3P33UdSijxdpFBayenUNaUsz/CcDsTkE6D2Wn4HQliVGwvucrH5po7urneHMPc/LTR12bdDQH67uoqu  
0AICPZxWUVuaEIUymllil1KYF//8Ay7vrRG3zhiSp++KEVU6bRIFTs/2SbpHR3wpl1J4cugTNWg/fJSUuMmYQ  
MrNIU6W7XIhVcJyc6oyIhA8hLs547mcmukMSUk5a10yGInH1spZSKJStmZPPI981n/f56fvracbvDiTkx38yS7nbx  
4VXI9PT7KJhA19ZNS4po7OonVz8E1Thdv7CAJSWZTEu1kqnJKspM5iOryvEGAvp8VerFrL+6fCY7T7bx/9YfZ  
ElJphYAh4foaHKYpLSkhAkIzGCt4VWclRw1rS8qdjgcQnFWMm7X5Loth8pMcWlCpPSKaSLCN+5cQlIOCp967  
C1Od/TaHVLm0ExEKaWUUiGV7nbxow+tpM/r56M/264D/8doyidlFv4/h+q76On32R2KUIHFgMOxpm4aO3Xhe  
aXU+M0tSOcHH1zBkcZu/vaxt/BpfclRTfmk7Ondp3hh72l+u73G7lCUiipbj7fyzO46frOthubufvDUUrFoKvm5vEv  
ty5m4+Em/vfT+6ZMXc2JivmB/pPV3W/Vg+r1+gkEzlgZMBs7+0hzJ5CSOOuVW0gFAobGrn6yUlwhHZ+IJsczYL  
UeB4zRumlKqQm755lZnGrv5T/+chS3y8k/3rRwylUOGKspn128/6li9td1MGeUGmVb32lhy7EWkhOd3H9pGalTq  
D5YuL10oIH9dZ1kpb4/9lyEpxTvgE3KqyalYvT4SDDnUDptBS7w1FKxbDPXTeXnn4//735OC6ngy+/b74mZhcw  
5TOLwkw3hZmjz9xs7LK6b3oH/HT2eSOaLAUChl6v35ZEcMAXIGBMWFuwBq9tu8fLgD8QtqTM5w/g9ZtJf3m

NNLviTk50ctXcvIgeUykVn0SE/33TArz+AI9seodAwPCVGxfEVH3QSJjySdlYrZ6dS8AY8tKTKMpMjuixn3rrFN  
WtHpaUZHLNngoKIHbetZ4DHt9fg9Qe4ZWkx5cGlnUJtzfx8th9vpTw3NWxdwz39Pn6zrZrufh9rFxWyoCgiLMcJtZ  
5+H49traZnwMcNiwuZXxgbcSul1PIeHh+6ZREOGZ++fpzm7n6+cedSLUk1hCZIYzQtNZF1y6ZH/Lhef4DqVg9gr  
dMZSXUdvfR5rbFE1a2esCVI07OSmb48vNe2qaufrj5rjNSJ5p6YScoau/rp7j8btyZISqLY5nAID9+yiPwMN9988RA  
tPQN8/74VZCa77A4tKmh6GuVcTgerKnLISUtkVUVkqyJX5KVRnptCUaabJSWZET12qJvkJzO3IJ38jCRWIGXb  
Hc6YIWYnM6cgzYp7RuzErZRSwxERPrVmNt+8cwlvHGvhl+/zoHTnXaHFRUk1qanVIZWmh07dtgdhlIqzlRW  
VqLvLUpF1o4TrXzqsV109Hr52rrF3LmyJO4mAjiTmNM5Vj21ZYypZRSStmisnwaz/2PK1hWmsUXfl/FQ7/cSW  
PX1C1YrUmZUkoppWyTl57ErX+8lK/cOJ+Nh5u4/t838bsdNQQCsdWTFwqalMWR2jYPf9hVy86TbXaHEjO2vtP  
CU2/V0qBLCSmllG2cDuGhKyt44dNXUJGXhd/X8UtP3idbcd7Q4tojQpiyMbDzdxssXDpsNNZ2bsqeG19gyw5  
VgLJ5o9vHak2e5wlfJqypudn8bvP7GK796zjJbuAe7+8Rv81aPbeat6ajQ2aFIWYj5/gM1Hm9lytBl/hJteCzOsIrhZK  
S7claj7Yoxh6zstbDzcRL8v/pbZSU1yku62qsIUZCTZHI1SSimwZmeuWzadv/z91Xxh7Tx2Vbdx239u4UM/3cprR5  
riev1MrVMWY1WnOs40t6YmJbC0Nctix37v/Hwump5JZoorJFXxDzd0s+VYCwAuh3DZ7NxJP2Y0SUpw8qFLy+  
js85KfPvqqDkoppSInOdHJp9bM5oHLyvn11pM8suk49//XNmblpvKhS8u4Y2VJ3NU306QsxNKHliWU5o7s5RU  
R8jNC11ykuRNo7u7H5w+QEqrdfbpdTlsWQQ8EDHtPdeB0CIunx3YNOKWUCqfUpAQeurKCD68q50/7TvPLN0  
7y1efe5psvHuKGxYXcunw6qyty4mLd5Pj8pLXRnIJ07qp0IiJmZ4rsckyh1uf1405w4nc6GPAF7A4nruyubWfjoSbA  
KhA8rzDd5oiUUiq6uV1Obltewm3LS9h3qoPHtlXz3J46nnrrFHnpSdytJhbl01n8fSMmK11pkLZGJRkp9gdQkg4R  
M609sXBF5Co4hjyqhHXVimlxmfx9Ez+720X8X9uXsgrB5v441un+OUBJ/mv148zKzeVm5YWc8vSYmbnp9kd6r  
hoRX81oqONXfR5AywsysDhiM1vHtHIGMPbptxOR3MLdBWsmigFf2Vim3tngH+tK+eZ3bX8ebxFoyBhUUZ3  
LKsmJuWFNnWYDKeiv5TKinr9/lp6uqnMMMdF33PkdDuGWDAAH9CB8CruaVKmVPxo6Ozj+arTPLOnjt017QC  
sLMvmlqXF3HhREXnpkZtxH7VJmYi8B/h3wA/sMMZ8Vks+AKwDTgIPGG08Iz3GRJMyYwy/evMkzd0DzMX  
N5dbl0ydwBINLQ2cfv91egz9guGFxIQuKMuwOSamw0aRMqfhU3eLh2ao6nt1Tx8H6LhwC11XkcsvSYtYuKiQzJ  
bwzOKN57cuTwHuNMVcA+SJyBbDGGHM5UAXcGq4DBwy09lj5XnN3/6j7d/Z58flHH9ze0++LyxpeYBVXHay  
11tI9MOK+Pn+Azr4R82kVRTp6vRGvo6eUUnaYkZPCp9bMZv1nrmTDZ6/kb66eTXWrhy8+WcXF//ISD/58B8/s  
qcMzYH/R9YgO9DfG1A/51QcsAV4N/v4ScB/wRDiO7XQIaxeXcLihm2UIWSPuu/loM9uOt5KXnsS9l8zAOcxYq  
qONXTxfVU9igoP7LpkR9mw70uYWpFPf2Ue/N8DKsuxh9/P6A/xmWzUt3QOsqsjh0lk5EYxSjderhxrZXd1OYaa  
bD1SW6lhBpdSUMbcgnc+vncffXz+XqtoOntlTx3NVdbx0oIFkl5PrFhZw89JirpybS1JC5Msl2TL7UkSWALIAO1Z  
XJkAHcMFPfhF5CHgiYMaMGRM+7vzCDOYXjt4Fd6KIB4Cmrn56BnxkuC+cbFW3eggYQ5/XT0NXX8STMm  
NMWKf9Oh3Cmnn5ox6nq893piWtusWjSVmUO9lsPb/rO/ro9wVITozsG89knrfhs4rpaYGEWFpaRZLS7P4yo0L2  
Ha8lWer6vjTXmscWoY7gfcLuLmpcWsqsgZtnEm1CKellnINOD7wN3ASmBwcFcGvPl2LsaYR4BHwBpTFu4  
YL6vIZcuxZsqmpQ6bkAEsL82muWuAICQnM3NTwx3WOTYdbmJXdRsLizK4flFhWI7R2efliR219Pv83LpsOsX  
D1F2blprIirJsats0IYsF18/JZevxVmblpkU0IfP5Azy5q5aGzn6uXVDAwuLxjVHc+k4Lb7zTQkVeGjctKdLkTckVE  
k6HsKoih1UVOzfTLYt4/UjzmRa03+6oITctiZuWFHHZ0iJWzMgO63tPRJMyEUkAfgV8wRhTLyLbgb8BvgFcC7  
wZyXiGMzM3dUxJVnZqIndfXBqBiN5tf10nxsDbpzu5bmFBWJ4k1S0eOnutcWJHGruHTcoArpqbF/Ljq/CYnZ/O7  
PzIl+Fo7Rmgrr0PsJ63403KBp/zRxu76fcFbFmJQSkV31xOB2vm57NmFj59Xj9/OdjIM7vreGxbNY9uOUF5Tgp3X  
1zKnStKQrCqzqBlit5TdBVwMfD2YRHwZ2CQirwPVwHciHM+Y1bR6eOlAA3npSbxvcVHEmjKhs2JGFruq21IU  
HL7KxYUZbqpbPfr5/dx4UXha49TUkZOWxKy8VE539LGsdPxLS60oy+aNYy3Mzk/ThEwpFXZul5MbLyrixou  
K6Orz8uL+Bn63o4ZvrD/EtzYc5r3z87nn4lKumpsXsjJbU6pO2WQ8vfu7zRZY3Huvrg05pdQGot9pzz489sNgPW  
BqK1hKp5pSQyl1Fgca+rmdztqeHJnLc3dAxRkJPGx1TP566sqLrh/NJfEiAk1rR6e2FHDtuOtZ7bNzk9DBHLSesIN  
S7QxuuENxr39ROvoO49BaXYKKYIOXE5hVoTHzCmlIFLRqClvjS+/bwFvfPkafvShlSwsyqChc/RSW2Oha19ew  
GtHmno7KO2rZcFRemku10sKs5kTn46LqdE7QDjjYeBaOrqD8adQVrS5P68mSkuHrxifGfjCokKCEoppdQZLqe  
DGxYXcsPiQkLV6zilPmmPNXWzfl89de29I+5XnGUN3stOcZECHLvs2efl1UONZ5ZriEaDXarTUhNxxJ4TmT+t0i  
CZkSiml1AhC1VgzZVrK/AHD81Wn8QcMde29fOzymcPue9XcPBZPzyTD7TozeO+1w80cbugCrOQnHLMuJuvq  
eXlcVHJu3EoppZSKDXH7yV3d4mHLsWa6+61lExwC6W4rB81lPlt7rLVngC1Hre7KQsJCbloSiUNamzKD90IM  
cOCOCLHNSbpQ3EoppZSKDXHZUtbT7+OPu0/hDxjqO/q4fUUIsI9F8/gdEcVJdkpZ/Z9Zvcp2jxedte288mrKoZtg  
w9O4eS7GSyUlwjFpRVSimllJqluEzKHCI4xFq/aWg3XnKik115aefsO/jCaPUHRMRynUGolJJKKaXCJC6TsuREJ  
3dXlnK6o495hSNXLI+3rJhjTT2UTUuJ2lmVSimllIp/cZmUAeRnuMc0GD/d7WJZaVb4A1JKKaWUGoGOCfDK  
KaWUigKaCmllfJKRQFNypRSSimlokDMLUguIk3AyUk8RC7QHKJwokE8nY+eS3SaKueyAtgVxsePRrEUbyzF  
CrEVbyzFCrEVby6QaozJG8vOMZeUTZaI7Bjrau2xIJ7OR88lOum5RM/jh1osxRtLsUJsxRtLsUJsxTveWLX7Uiml  
lFIqCmhSppRSSikVBaZiUvaI3QGEWDydj55LdNJziZ7HD7VYijeWYoXYijeWYoXYindcsU65MWVKKaWUUt  
FoKraUKaWUUpkFHU3KIFJKKaWigCZlSimllfJRQJMyZRSSqkokGB3AOEmIk7gVuBSIBtoB94E/miM8dkX2  
fjpuUqnPZfoFO5zibVrFUvxxlKsEJPxpgGfwIo3i7Px/tgY02VfZO8WS7HC5OON+9mXlVJLoAp4GegAMoBrgaX  
GmA/ZGdt46blEjZ2X6BTuc4mlaxVL8cZSRBCT8T4D/JJ3x/thY8zNdsZ2vliKFUIQrZEmrn+A18azPZp/9Fyi80fPJ  
Tp/wn0usXatYineWIo1RuPdDDjO2+YANtsdWyzHGop44777EnhGRJ4DXgU6sbLWq4Bn7AxqgoY712ftDGqCzj  
+XTObK4uNcYvvn8rSey5jF2ntLLL3mYu15GGvvAT8AXhWRKS4+FxYB/2lrVBc2XKw/tDWq4U3q2sZ99yWA  
iFwOXITvt9sBbAdmGWO22hnXRIhJITAXcAJ+rIz8V/ZGNTEikgtcgvWkbQcqiTFfszWoCRpyLiuBo8BRY8x2e

6MaPxEPArxY55IFlAPVwOMmCsfGjEREbgF2Aks5O7ZjuzGmKYTHiKn3llh5zcXa81BEEoF7gDKs178LK+bvG  
WPa7YtseCKSgPVZMvhcOBKN1xbOiTULK9bD0RorTC7euE/KRORbQD5WApMDfMwY0yQifzHGvNfe6MZ  
HRP4reHMAyAPqsDLxfGPMQ7YFNgEi8how+OST4L8Lgf3GmCvtiWpiRGS9MeYGEfkM1tiB54DVwCljzD/Y  
Gtw4Db4uROR7QA/wCrAM68P7bluDgycRqQNOAg3AU8Azzpi2ED5+TL23xNjRLtaehyLyFLANa5D/SuB5oA  
W4zxiz1s7YLiSWJialIAA3Yr3ONhhjAsHt64wxT9sa3BiJyFeNMf84ln2nQvdlpTHmKgARWQI8ISJfsDmmiZo95F  
z2GmPuDN5+xd6wJuQpYAnwqDHmVQAR+ZMx5n22RjUxicF/bwPWB80fiQir9sY00QFgv8uNMZcG7y9IUaf  
Y4eMMWtEZCZwO/CUiPQDTxtjQtFNE2vvLbH0mou152GWMeZf4cx787eDtx+wNarhPQrsBX7DuYPRHwWib  
WLCL4ETgA/4nyLyODHmEPB3QNQLZSJSjdWqO/gcFmCRiFw9li8/UyEpSxCRRGPMgDGmSkRuA36F1ccba4b  
+vb4y5Lacv2O0M8Z8O9jk/6CIfAJ4zO6YJmGhiPwCqACSgN7gdrd9IU3Yz0Xkp0CNiPwK2Ij1Qb7D3rAmzhzhH  
PgW8C0RKQDWheihY+q9JcZec7H2POwRkf+F9fqvF5G/B1qBfnvDGla5Meb+87a9FWxNjTYIjIjIDVUQeAR4V  
R/YHNNIPgvcAWwAfmWM8Y3ny89U6L68BDhhjGkcss0J3GWMedy+yMZPRBYBB40x/iHbEoEbjDHRORh4V  
MH+9/uBebHW3QcgImVDfq0zxniDtWquMMb8ya64JkpEioG1QAHWt+gtxpg99kY1fiKy1hjzYhgfp2bfW2LhNR  
dLz0MRSQZuAI4BR4CPYH1ZfswY02FnbBciIp8HrubbExM2GWO+aV9k7yYim4CbjDGdwd8TgUeA24wxmbY  
GNwIRuRH4MLAFuGVli+/I94v3pEwppZRS5xoy6SOL4CQYrBa0qJqgJCLIQPv5EyZE5EpjzCZbghoHEVkdLAb  
eHMu11aRMKaWUmKJEZLglF180xlwX0WBGEEUuxwrDxCrB+LPFOhTF1SimllDqrG2u25VCCNW4v2gzGKpw7  
ezgaY4Vz4wUr5jHHq0mZilnBgdV/ABYYYw7aHY9SSsWIA1hjss4Z7yYif7YpnpHEUqwwyXi1+1LFLBH5HVA  
EvGyMedjmcJRSkiYei/O2GGMGztueEIVlymImVph8vJqUqZgUnN14CFiDVRROfrAv//tYs4iOY6039t/GmN+Ly  
Erg20Aa0Aw8YIw5bU/0Siml1LsNN4BOqWh3K9bAycNAq4iswCoQWo617M2DwCoAEXEB/wHcaYxZCfw38C  
82xKyUUKoNS8eUqVh1L/Cd4O3Hg7+7gCeCFfXrh1T/noc1JfnP1oodOAFtJVNKKRVVNCITMUdEcoD3AotFxG  
AIWQZrGZkL3gVrfb9VEQpRKAwUGjftvLSx6E7gF8aYMmNMuTGmFGsMWTNwh4g4gsvpXB3c/xCQJyJnujO  
DqyMopWwilreJiBGR+TYc+0SweOqYtkcLEVkeXH4KEXk4WJl/uH0fPu/3PBFZH+YQ1SRpUqZi0b28u1XsSaA  
YqAX2AT8GtgldwVkwDwJfF5E9wG7gsohFq5S6kHuB14F77A4khnwFa3zssERkYXBpok+KyC4RuRfAGNMEn  
BaR1RGIU02QJmUq5hhjrjbGrD9v2/eMMZ8EPm+MWQj8FTAX2Bv8/93GmCuNMUuNMYuMMT+JfORKKTg  
ze3o11uv0niHbrxaRV0Xk9yJyUER+LcGBBoMFWRH8KJhp7B1vYzm8xEPf9waV5EJE/ishOEdkvlg+NI75yETkgIj  
8J3ndDcH1LRGS2iLwkInuCsVSI5ZvBY+8VkJQ8MOZ+NIvI7ETksIv9PRD4oItuC+1UE98sTkSdFZHvw512Jk4ik  
A0sutP6niHxcRP4UjPFh4BfAD4PXeoJSPn8EPjjW66AiT5MyFW+eE5HdwGvA14wx9TbHo5R6t1t59+zpQcuBz  
wALgVlYicWgZmPMCqyEY9iuuyE+FpxXQl8OjgedazmAD8wxizCWhvyjuD2Xwe3L8VqcT+NNfn7GbAUuBb  
4ZrBeFcFtf4c1K/x+YK4x5hLgp8D/CO7zXeDfjTEXB4/z0wvEU4nVC3AOEflb4GbgVmNMLzAA5AMOY0yvMe  
bokN13AFEM4xqoCNOB/iquGGOutjsGpdSolJR7elfw923GmFqA4BescqXuTrBW8ADYiZUIjebTwZU/AEqxEq2  
WMcZ43Bize8jxyoOtVdONMU8BGGP6gnFeDvzGGOMHGkRkI3Ax0AlsH6yJKCLHgA3Bx9yLVWcRrERuYbB  
RECBDRNKNMV1D4ikCms6L8X6sIRu3GmO8wW1fAv4NWCsiy4H/NaR1rRfRmleKUpqUKaWUipjhZk+LyBe  
Du/QP2d3PuZ9T/RfY7uPcXh938DhXYyU7q4wxHhF5dfD/xuj8OJI5u57h+Ybbfv7jBlb8HuDsOTiCcfaO8Di9vDv  
+fVgtdCVYk50wxpwC7hWRr2J1Xf4BqAju7w4+jopS2n2plFIqkoabPX35BB/vBLACINgNOjO4PRNoCyZk84FLJ  
xc2GGM6gVoRuTV4vCQRSQE2AR8QEaeI5AFXAAtvG8dAbgL8d/EVEll1gnwPA7PO2vQX8NfCMiBQH7zs4szy  
A1cKXOmT/uVygC1RFD03KlFJKRdJws6fvm+DjPQlMC3Z1fhI4HNy+HkgQkSrga8CbE3z8892P1S1aBWwBCr  
HOpwrYA/wF+OI4x7N+GqgUkSoReRv4xPk7GGMOApnBLtSh21/HG1/3fLCcx+0i8ibwMaxk79NDdl8DPD+OuF  
SE6dqXSImlVAwQkc8CXcaYC00EOH/fh40xD5+3bROWzhjTFqYQ1SRpS5lSSikVG37luWPURvLq0F+C3arf1o  
QsumlLmVJKKaVUFNCWMqWUUKqpKKBJmVJKKaVUFNCkTCmllFIqCmhSppRSSikVBTQpU0oppZSKAv8f  
st6I5l0KciEAAAAASUVORK5CYII=",

"text/plain": [

"<Figure size 720x720 with 4 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"pd.plotting.scatter\_matrix(data.loc[:,\"Age\":\"Annual Income (k\$)\"] , diagonal =\"kde\", figsize=(10,10))"

]

},

{

"cell\_type": "markdown",

```

"id": "b7153f24",
"metadata": {},
"source": [
  "# 4. Perform descriptive statistics on the dataset."
]
},
{
  "cell_type": "code",
  "execution_count": 7,
  "id": "dd6ec149",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",
          "    vertical-align: middle;\n",
          "  }\n",
          "\n",
          "  .dataframe tbody tr th {\n",
          "    vertical-align: top;\n",
          "  }\n",
          "\n",
          "  .dataframe thead th {\n",
          "    text-align: right;\n",
          "  }\n",
          "</style>\n",
          "<table border='1' class='dataframe'>\n",
          "  <thead>\n",
          "    <tr style='text-align: right;'>\n",
          "      <th></th>\n",
          "      <th>CustomerID</th>\n",
          "      <th>Age</th>\n",
          "      <th>Annual Income (k$)</th>\n",
          "      <th>Spending Score (1-100)</th>\n",
          "    </tr>\n",
          "  </thead>\n",
          "  <tbody>\n",
          "    <tr>\n",
          "      <th>count</th>\n",
          "      <td>200.000000</td>\n",
          "      <td>200.000000</td>\n",
          "      <td>200.000000</td>\n",
          "      <td>200.000000</td>\n",
          "    </tr>\n",
          "    <tr>\n",
          "      <th>mean</th>\n",
          "      <td>100.500000</td>\n",
          "      <td>38.850000</td>\n",
          "      <td>60.560000</td>\n",
          "      <td>50.200000</td>\n",
          "    </tr>\n",
          "    <tr>\n
```

```

"    <th>std</th>\n",
"    <td>57.879185</td>\n",
"    <td>13.969007</td>\n",
"    <td>26.264721</td>\n",
"    <td>25.823522</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>min</th>\n",
"    <td>1.000000</td>\n",
"    <td>18.000000</td>\n",
"    <td>15.000000</td>\n",
"    <td>1.000000</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>25%</th>\n",
"    <td>50.750000</td>\n",
"    <td>28.750000</td>\n",
"    <td>41.500000</td>\n",
"    <td>34.750000</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>50%</th>\n",
"    <td>100.500000</td>\n",
"    <td>36.000000</td>\n",
"    <td>61.500000</td>\n",
"    <td>50.000000</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>75%</th>\n",
"    <td>150.250000</td>\n",
"    <td>49.000000</td>\n",
"    <td>78.000000</td>\n",
"    <td>73.000000</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>max</th>\n",
"    <td>200.000000</td>\n",
"    <td>70.000000</td>\n",
"    <td>137.000000</td>\n",
"    <td>99.000000</td>\n",
"  </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"

```

],

"text/plain": [

```

"  CustomerID      Age Annual Income (k$) Spending Score (1-100)\n",
"count 200.000000 200.000000      200.000000      200.000000\n",
"mean  100.500000  38.850000      60.560000      50.200000\n",
"std   57.879185  13.969007      26.264721      25.823522\n",
"min    1.000000  18.000000      15.000000      1.000000\n",
"25%   50.750000  28.750000      41.500000      34.750000\n",
"50%   100.500000  36.000000      61.500000      50.000000\n",
"75%   150.250000  49.000000      78.000000      73.000000\n",
"max   200.000000  70.000000     137.000000      99.000000"

```

```

]
},
"execution_count": 7,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"data.describe()"
]
},
{
"cell_type": "code",
"execution_count": 8,
"id": "fde09c99",
"metadata": {},
"outputs": [
{
"data": {
"text/html": [
"<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>count</th>\n",
"      <th>mean</th>\n",
"      <th>std</th>\n",
"      <th>min</th>\n",
"      <th>25%</th>\n",
"      <th>50%</th>\n",
"      <th>75%</th>\n",
"      <th>max</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>CustomerID</th>\n",
"      <td>200.0</td>\n",
"      <td>100.50</td>\n",
"      <td>57.879185</td>\n",
"      <td>1.0</td>\n",

```

```

"    <td>50.75</td>\n",
"    <td>100.5</td>\n",
"    <td>150.25</td>\n",
"    <td>200.0</td>\n",
"  </tr>\n",
" <tr>\n",
"   <th>Age</th>\n",
"   <td>200.0</td>\n",
"   <td>38.85</td>\n",
"   <td>13.969007</td>\n",
"   <td>18.0</td>\n",
"   <td>28.75</td>\n",
"   <td>36.0</td>\n",
"   <td>49.00</td>\n",
"   <td>70.0</td>\n",
" </tr>\n",
" <tr>\n",
"   <th>Annual Income (k$)</th>\n",
"   <td>200.0</td>\n",
"   <td>60.56</td>\n",
"   <td>26.264721</td>\n",
"   <td>15.0</td>\n",
"   <td>41.50</td>\n",
"   <td>61.5</td>\n",
"   <td>78.00</td>\n",
"   <td>137.0</td>\n",
" </tr>\n",
" <tr>\n",
"   <th>Spending Score (1-100)</th>\n",
"   <td>200.0</td>\n",
"   <td>50.20</td>\n",
"   <td>25.823522</td>\n",
"   <td>1.0</td>\n",
"   <td>34.75</td>\n",
"   <td>50.0</td>\n",
"   <td>73.00</td>\n",
"   <td>99.0</td>\n",
" </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"

```

```
],
```

```
"text/plain": [
```

```

"      count  mean   std  min  25%  50%  75% \\n",
"CustomerID    200.0 100.50 57.879185  1.0 50.75 100.5 150.25  \n",
"Age           200.0 38.85 13.969007 18.0 28.75 36.0 49.00  \n",
"Annual Income (k$)  200.0 60.56 26.264721 15.0 41.50 61.5 78.00  \n",
"Spending Score (1-100) 200.0 50.20 25.823522 1.0 34.75 50.0 73.00  \n",
"\n",
"      max  \n",
"CustomerID    200.0  \n",
"Age           70.0  \n",
"Annual Income (k$) 137.0  \n",
"Spending Score (1-100) 99.0  "

```

```
]

```



```

    },
    "execution_count": 8,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "data.describe().T"
]
},
{
  "cell_type": "markdown",
  "id": "97ed7025",
  "metadata": {},
  "source": [
    "# 5. Check for Missing values and deal with them."
  ]
},
{
  "cell_type": "code",
  "execution_count": 9,
  "id": "f8a6ad05",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "CustomerID      0\n",
          "Gender          0\n",
          "Age             0\n",
          "Annual Income (k$)  0\n",
          "Spending Score (1-100)  0\n",
          "dtype: int64"
        ]
      },
      "execution_count": 9,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "data.isna().sum()"
  ]
},
{
  "cell_type": "markdown",
  "id": "acad58cb",
  "metadata": {},
  "source": [
    "# 6. Find the outliers and replace them outliers"
  ]
},
{
  "cell_type": "code",
  "execution_count": 10,

```

```

{id": "45bcc770",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"<AxesSubplot:xlabel='CustomerID'>"
]
},
"execution_count": 10,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUUhEUgAABGUAAAE9CAYAAACiD99jAAAAOXRFWHRTb2Z0d
2FyZQBhbnRwG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACX
BIWXMAAAAsTAAALEwEAmpwYAAAhHUIEQVR4nO3de5QtZ1kn4N+bBGIIQEDAnjAQ4XASGEgATQ4S
Z3ISgOyVIGF47MeAIRB8VIFgZWZi0VRmCCIIgYdBhAgWAWIBeBgI4juZEbl2AjB02IIEhQLiGb/6o6mSn03
3O7qS7v9p9nmetXt27du3ab9Xuent/v11VXa21AAAAALC19uldAAAAAMDeSCgDAAAA0IFQBgAAAKADoQ
wAAABAB0IZAAAgA6EMgAAAAAd7LeemQ8++OC2Y8eOTSoF2EznnnvuVa21Q3rXcVPpP7DY9CCgl0XvP
4keBItTz1oXaHMjh07cs4559z8qoAtV1Wf713DzaH/wGLTg4BeFr3/JHoQLLI99SCnLwEAAAB0IJQBAAA6E
AoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIaOhDIAAAAAAHQhIAAAAAADoQygAAAAB0IJQBA
AAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIaOhDIAAAAAAHQhIAAAAAADoQygAAAAB
0IJQBAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIaOhDIAAAAAAHQhIAAAAAADrYr3c
Bi+TUU0/N0tJS7zl2xWWXXZYkOeywwzpXsjl27tyZ448/vncZsKqevcW+Dyyb2vucnv1J74Dtbyo9b7u9F9M/108
osw5LS0s5/+JP5dpbHdS7lA237zVfSZL867e236/Evtd8qXcJsFs9e4t9H1g2tfc5vfqT3gF7h6n0vO30Xkz/vGkW/5Xf
Ytfe6qB8495P6l3Ghjvg0+9Jkm29bjBlvXqLfR+YNaX3Ob36k94Be48p9Lzt9F5M/7xpXFMGAAAAoAOhDAAA
AEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM
AAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhD
AAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQy
AAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMo
AAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQ
```

AAAAAHQhIAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAF7bcZCd+7  
cuRmLBWbYz1Znu8DWsK+tznaB1dk3Np9tDfTjo/e1TQlIj/++M1YLDddfrY62wW2hn1tdbYLRm6+sflsY9gaG7  
2vOX0JAAAAoAOhDAAAAEAHQhKAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAAOhA  
KAMAAADQgVAGAAAAoAOhDAAAAEAHQhKAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUA  
QAAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhKAAACADoQyAAAAAB0IZQAAAAA6EMoAA  
AAAdCCUAQAAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhKAAACADoQyAAAAAB0IZQAAA  
AA6EMoAAAAAdCCUAQAAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhKAAACADoQyAAAA  
B0IZQAAAAA6EMoAAAAAdCCUAQAAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhKAAACADo  
QyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhK  
AAACADoQyAAAAAB3s17uArbPvNV/KAZ9+T+8yNty+1/xbkmtDftSkjv1LgN2q1dvse8Ds6b0PqdXf9I7YO8  
xhZ63nd6L6Z83jVBmHXbu3Nm7hE1z2WXfTZlcdth23InutK1fOxZfz99P+z6wbGr7S7/+pHfA3mAq+/n2ei+m94  
UQp11OP7443uXAGxDegswBXoRsDfR85gK15QBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCG  
QAAAIaOhDIAAAAAHQhIAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAA  
QAdCGQAAAIaOhDIAAAAAHQhIAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6E  
MAAAAQAdCGQAAAIaOhDIAAAAAHQhIAAAAADoQygAAAAB0IJQBAAAA6KBaa/PPXHvIks/PMevBSa  
66qUVN2HZdr8S6Lar1rNsPttYO2cxiNtM6+s+8FuX3Qp0bS50bSw+66ab2GqtnbVOqJVHPnqxWz0L3n2SuHjS11  
2Eat58i1Zvsng1z1PvbnvQukKZeVXVOa21ozZ8wZ1t1/VKrNui2s7rttkWZdupc2Opc2MtSp1TNL Vtp561TamWR  
D17MrV6tsoirreaN9+i1ZssXs0bUa/TlwAAAAA6EMoAAAAAdLBZoczrNmm5vW3X9Uqs26Lazuu22RZ126lzY6  
lzYy1KnVM0tW2nnrVNqZZEPXsytXq2yiKut5o336LVmyxezTe73k25pgwAAAAAu+f0JQAAAIaONjSUqaonV  
NUIVbVUVb+1kcvuoap2VdVfVXV+VZ0zTjuoqj5QVf84fr997zrnUVVvqKovVtXFM9PWXJeqOnF8HS+pqsf3q  
Xo+a6zbyVV12fjanV9VT5q5byHWrrarUIUfrqpPVdUnquqEcfq2eN22yk3Zjp3q/L6qOquqLhjrPEU61xWVftW1c  
er6l3j7cnVuSg9vKpuV1Vvq6pPj7+nR0+tzqq610w/Pb+qrq6qX51anVM01R40pX14avtAVf3a+FpdXfVvHvvzltW  
zxvuabn/716jnpPrdWfVnV5Vt9uKelarZea+51dVq6qDt6KWKamJ8Gm2gfnMaVeOY+p9dM96d1v56xx03vyho  
UyVbVvkj9M8sQk90lyXFXdZ6OW39GjWmtHzPybq99K8sHW2j2SfHC8vQhOS/KEFdNWXZfxdTsyX3Hx7x6f  
H2n6rTceN2S5OXja3dEa+09ycKt23eT/PfW2g8leWiSx7r3y6v21ZZ13bs6FtJHt1ae0CSI5I8oaoemunVueyEJJ+au  
T3VOheh78yyXtba/dO8oAM23VSdbbWLInup0kelOSaJKdnYnVO1FR70JT24cnsA1V1WJLnJtmqtXa/JpTm+Nu  
6lfWclmm9Z1utng8kuV9r7f5JPpPkxC2qZ7VaUIV3SfLjSf55Ztpe8b5oQcZgU+2D85hSr5zHZPrpnkyk387jtGx2T2  
6tbchXkqOTvG/m9oIJTtyo5ff4SrIrycErpl2S5NDx50OTXNK7znWsz44kF+9pXVa+dknel+To3vWvc91OTvL8VeZ  
buHWbqfWvMrzh2Dav2xS34xS+ktwqyXlJfmSKdSY5fPwD9Ogk7xqnTbHOyffwJN+f5HMZr/E21TpX1Pa4JP936  
nVO9WsKPWhK+/DU9oEkhyX5lyQHJdkvybvG3/ktrWeV9zVd//avrGffU9P8qatqme1WpK8LcMA9Lq+v1Xbpv  
dXFnAMNoU+OGedk+mVc9Y7qX46R72T6Ldz1rqPpXkjT19a3qjLLh2nLbKW5P1VdW5VPXecdqfW2uVJmN6/  
Y7fqbr611mW7vJa/Mh5W+4aZQ8oWct2qakeSByb5WLb/67Zp5tyO3YyHyJ6f5ItJPtBam2SdSV6R5DeTfG9m2hT  
rXIQefrckVyb50/Hw6NdX1YGZXp2zjk3y5vHnKdc5ORPqQa/IdPbhSe0DrbXLkrwsWxEXlyf5Smvt/b3qmTHlv/3P  
SfLXveqpqqcmuay1dsGKu6awbbbCQq3nhPrgPF6R6fTKeUyqn+7JhPvtPDa0J29kKFORTFv0f+308NbakRkOB/zlq  
vqx3gVtke3wWr4myd0znAZyeZL/OU5fuHWrqIsneXuSX22tXb27WVeZNul120rr2I7dtNaubcPpIYcneUhV3a9zS  
TdSVU9J8sXW2rm9a5nDIvTw/ZlCmeQ1rbUHJvl6+h+mu6aquMwSpby5y961LJqp9KAJ7sOT2gfGD3GeluSuSe6  
c5MCq+tle9cyh69/+qnpPhlNT3tSjnq6VZIXJnnRandvZS0dLcx6TqUPzmOCvXiek+qne7KA/XYeN2l/3MhQ5tIkD  
5m5fXiSL2zg8rdea+0L4/cvZjh3/iFJrjqQ5Nk/P7FfhXebGuty8K/lq21K8YB7veS/HGG1y5ZsHWrqItk+OP1ptbaO8  
bJ2/Z12yZr3I7dtda+nOTMDOeiTq3Ohyd5alXtSvKWJl+uqv+d6dW5KD380iSXjkdFJcMh+EdmenUue2KS81prV4  
y3p1rnpEysB01tH57aPvDYJJ9rrV3ZWvtOknckeVjHepZN7m9/VT07yVOS/Ewbj9PvUM/dMwzoLhh/pw9Pcl5V/  
UCHWnpZiPWcWB+cx9R65Tym1k/3ZKr9dh4b2pM3MpQ5O8k9ququ4ydpXY5YwOXv6Wq6sCqus3yzxnOb7s4  
wzo9e5zt2RnOiVxUa63LGUmOrar9q+quSe6R5KwO9d1kyzVJ6OkZXrtkgdatqirJnyT5VGvtD2bu2rav22a4Cduxi6  
o6pMb/XIFVB2T4Q/XpTKzO1tqJrbXDW2s7MvT5D7XWfjYTq3NRenhr7V+T/EtV3Wuc9Jgkn8zE6pxxXK4/dS  
mZbp2TMbUeNLV9eIL7wD8neWhV3Wp87R6T4UKZvX/XJ/W3v6qekOQFSZ7aWrtmRZ1bVkr7aLW2h1bazv  
G3+ILkxw5/17tLe+LJ8Gm1ofnMfUeuU8JthP92Sq/XYeG9uT93TRmfV8JXlShiufzbcJcZdy2Vv9leGcvAvGr08sr0  
+SO2S44NM/jt8P6l3mOvz5gyn8Xwnwx+sn9/dumQ4FPSzGS5i9MTe9d+EdfvzJBcluXDcOQ5dtHVL8ogMh7tdm  
OT88etJ2+V1m/J27FTn/ZN8fKzz4iQvGqdPqs4VNT8y11/4b1J1LIIPz3Ca5Tnja//OJLefaJ23SvJvSW47M21ydU7ta  
8o9aCr78NT2gSQvzhCKX5zh/cT+W1IPJvaebY16lJcN2H5d/qPtqKe1WpZcf+uzFzgfbo3zVS+MvEx2JT74Jz1T6J  
XzlnrpPrpHPV27bdz1rjpPbnGBwIAAACwhTby9CUAAAAA5iSUAQAAAAOhAKAMAAADQgVAGAAAAoAO  
hDAAAAEAHQpkJq6ofqKq3VNVnq+qTVfWeqrnOpdxTFXZdZ7Nq3M3znllVR40/76qqi8avT1bVKVW1/1bXBP  
RVVU+vqlZV9+5dCwAwfVX1wqr6RFVdWFXnV9WPbPLznVxVzx9/fkIVPXaDlvuccSx0YVVdXFVP24jlvncB  
1TVR6pq3/H2e6vqy1X1rj08btX5ququVfWxqvrHqnprVd1ynF5V9b+qamlczyPH6besqo9W1X6btY6LTigzUVVVS  
U5PcmZr7e6ttfsk+e0kd1rnoo5JsQWhzPIOv8KjWms/nOQhSe6W5HVbWRMwCccl+bskx/YuBNh4PYPX8QOgg+  
edPhVV9cCqev3483WDwTXmPXnF7UOq6r2bXCJ0U1VHJ3IKKiNba/dP8tgk/7JVz99ae1Fr7W9u7nKq6vAkL0zyi  
HE9Hprkwpu5zNXGW2t5TpJ3tNauHW+/NMmz5njcWvP9XpKXt9bukeTfk/z8OP2JSe4xfj03yWuSpLX27SQfTP

LMddS8VxHKTNejknyntfZHxNaa+cn2Xc2rayqV1XVz40//+54JMqFvFwYqnpYkqcmeemYLN+9qo6oqn8Y5zm  
9qm4/PvbMqnr5mGJ+qqoeXFXvGBPQU2ae72er6qxxea+dSVy/NqbJH0ty9For1Vr7WpL/luSYqjpoA7cXMGFVde  
skD8/wh/vYcdo+VfXq8ROwd41HAz5jvO9B46c651bV+6rq0I7lA/MRvK7fbyc5dXczVNV9quqjSX6xqs6rquOSpL  
V2ZZLLq+rhW1An9HBokqtaa99KktbaVa21LyTXBa6/N45LzqqqneP0Q6rq7VV19vj18HH6yVX1hnHM809V9bzl  
JxmPxrmkqv4myb1mpp82875kV1W9eNwHL1oOn8fn+8A4/bVV9fVguA7Jvlqkq+N6/G11trnxsfvKq/qaoLxmXcf  
Tzi5KU1HFFzUVU9c5z3kVX14ar6P0kuqqp9x/nOHsd2/3WN7fgzSf5q+UZr7YNjPbu12nzjgQOPTvK2cdlbMxwE  
kCRPS/JnbfAPSW438/7tnWMdrElO1M133S3LuvDOPAcfTk9x3TGBPa39fZlzkvxGa+2I1tpnk/xZkheM81yU5Hd  
mFvPt1tqPJfmjDDvuL491/FxV3aGqfihDwvwn1toRsa7N9TvXgUkubq39SGvt73ZXa2vt6iSfy5CiAnuHY5K8t7X2  
mSRfquGQ1p9MsiPJdyf5LxkD3aq6RYZByjNaaw9K8oYk/6NDzcCcVgtex+mPHAdBb6uqT1fVm8Y39bsb5Nzgi  
JFxYlJj/PmdY1j7iap67jrj2zF+6PTH42PfX1UHjPetd1D0kar6i6r6TA0fiP3MOCi8qKruPs636sBwRU23SXL/1toFq  
9z3C1X112ONJ2d4//aacRufPTPrO2Ogw/b1/iR3Gfe1V1fVf1xx/9WttYckeVWSV4zTXpnhK14HJ/mpJK+fmf/eSR6  
f4cj936mqW1TVgzL0rAdmeF/y4N3Uc1Vr7cgM++Jyj/qdJB8ap5+e5D+s8rgLklyR5HNV9adV9RMz970pyR+21h6  
Q5GFJLh/rOCLJAzlChfTSMXDjIUleOJ5F8fNjVjKu64OT/EJV3XX2iWs4tehurbVdu1mv9bhDki+31r473r40yWHj  
z4flhkeyzd53cXa/bfdqQpnt4+ok30zy+qr6ySTXrJyhqm6b5HattY+Mk96Y5MdmZjlj/H5Rkk+01i4fk+1/SnKXJI9J8q  
AkZ1fV+ePt42PuTbJ29dRb61jXmDxHZfkLePPbxlvPyLJX7bWvtda+9ckHx7vv1eGQPgDY685KcnhW1susE7H5  
MbB67IHjvnVDKdT3y1DsLbstUHO7jxnDGuPSvK8qrrDOmq8R4bBz32TfDnDgC1Z/6DoAUlOyBAoPyvJPceB4  
euTHD/Os7uB4bKjMgxUbqCqfiXJTyQ5prX2jSTfzvBJ+z6ttW+01pZmZj8nyY+uYxvAwhiPsH9QhlNhrkzy1hrPE  
Bi9eeb78pH6j03yqvH9wxlJvn8MQJPk3a21b7XWrkryxQyXhfJRJKe31q4ZPzg+I2t7x/j93AwfKiXDe5m3jPW+N8  
PpPCvX49okT0jyjCSfSfLYMXy+TZLDWmunj/N9s7V2zbjMN7fWrm2tXZHKI7k+0Dhr+SibJI9L8p/Hdf1YhsBk5  
YfeB2fodxtltTFc29N94zb49srxwQwX25muT2TYcVf6bm4Ypn1fkrTWvltVD8kQlByb5FcyHFq2Ht8av39v5uf12/tl  
2NHe2Fo7cZXHfnPmPMXdGnfGHRmaErDNjYOmRye5X1W1JPtm+CN9+loPyRAMr3kqJDA5x+X6T6qXg9fzxt  
tntdYuTZJx8LAjw2lOyQ0HOT85x/M8r6qePv58lwwDkH+bs8bPjaeCLz/fjtUGRWod1w2KklxRVcuDoquTnN1au  
3yc77MZPs1Phg+1HjX+/Ngk9xkPCKrGgWFrbfZUGEMzDDRnPSvDp8vHtNa+M057QZKXJXl8VT0wyUkzR9d8  
Mcmd51x/WDjjPnhmkjOr6qlkz05y2vLds7OO3/dJcvQYaf5n3BdnxfX5vqx8Oxydmf58bOPneuD5tZaS3JWkrOq6  
gNJ/jTJH6wx++6W+fUV8x3fWnvfbub/Rsbx4u7UcAHl1443X9RaWyucuirDaUn7jUfLHJ7kC+N9l2boy8tm70uS/  
TMcRMAKjpSZrg8l2b+qfmF5QIU9OMNg5j5Vt945MtjxvtuneS2rbX3ZPg06ojxYV9Ncpskaa19Jcm/V9XyJyrPyp  
C8zuuDSZ5RVXccn/OgqvrB9azUWOerk7yztXajJBnYlp6R4RzjH2yt7Wit3SXDkYxXJfmpGq4tc6ckjxznvyTJITV  
c4C/j4cX37VE4sGczwevrq2pXkt9I8sy6PpFYayA0e9/s9FU/gKqR2YIO44ej2r5eOYYbKzyXLPpt9bgZ3eDopUfX  
M1+qLW8DssDwyPGr8NWBDLJ6oOlizOEvtcdHdhau6y1ldlyGUzRel+uDrIyPv8HgE7aLqrpXVc0e+XFEks/P3H7  
mzPf/N/78/gwfTi8v44g9PM1Hkzy9hv9QdJsMR6mtx98l+U/jcz0uye1XzlBVd15x9OARST4/HplzaVUdM863f1Xd  
aqzpmTVcM+aQDgc2nLXKc78vw7WmbjE+/p5VdeDsDON4a9+q2m2vbK19bKZfrXm00BgufTjXHzzw7Fx/vZo  
zMhy5U1X10AynVi0H2HdIcuVM2MwMocxEjb/wT0/y4zX8S+xPZDin+AtJ/iLDFbvflOENSTIEL++qqgszBC2/N  
k5/S5LfQKqPj+c5PzvDIbgXZmgIL1HTZ/McBrB+8fHfyDDpzzz+HBVXZyhofxzkUuRAVsP8flxkfVd3Dp7uXZ  
hiEvDbDobdfGa/S/4wkv1dVFyQ5P8MpBcA0rRW8PuImLm9XkuV/pXpkkuVrJNw2yb+31q6p4fozD715ZV93nbu  
bMyhayzwDw08l2bli2sczvEc6o6ruPD52OZT+XoYjfGYHXffMKqdAwTZx6yRvrPEfmWQ4BfLkmfv3r+GfjJyQ6  
8c+z0tyVA0Xvv1khn8wsqbW2nIj3prhvcbbk/ztOmt8cZLHVdV5Gf770OW58UV0b5HkZTVcV+v8DCHSCeN9z8  
pwBOCFSf4+yQ9keM90YYZr0XwoyW+Op3mv9Pokn0xy3jjOem1WPxPm/Znpx1X1t0n+MsljqurSqnr8aiu2m/lek  
OTXq2opwylTfzJof0+Gy14sJfnjJL80s7hHjfezihrG/gCw9arq1q21r42foJyV4ULiq73xACAqq5M8rvj9RSWpz0vy  
Q9lGOw8v7X2lHH6q5Kc01o7bTyq5qjW2lVVdVSSI7XWHlnDxW3/KsN1VM7OMJhYHuy8M8OFIy9JckiSk1trZ  
84ua0VtuzJcu+XWSd7VWrvfOP35SW7dWjt5/CT+tRmuvfCdJD+dIVT6/fF5W4Z/oPDW8Wid2fU5c7x9zux9Nfz3  
lT8ct8F+ST7aWrvR4HA8HeNhrbWv1vAvr7/WWnvZOPj53SQ/nuQXkzw5Q5B9dZKXtNb+YmY9vtVa2+1/cILtZ  
q19vkMd+ye5dryUxNFJXtOGf4gyGeNpj7/eWpvn32BvVg3vSHJia+2SXjVMmVAGG7GAc3tktwyie+31k7rWQ/  
AVqqqX0vy1dbaahcCXjnvya21k1dM+2iSpzklN3NhEKZe2Q4i2GfDBfl/qXW2tm7f9TWq6rnZLg26FzXAN3g57  
5lkmNba3+21c+9KIQyAADQwXidh59urf35HPM+srV25sztQzIcXfjOzasQgM0mlAEAAADowIV+AQAADoQy  
gAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgg/8PnGQaPk97LYoAAAAASUVORK5CYII=",

"text/plain": [

"<Figure size 1800x360 with 4 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

```
"source": [
  "fig,ax=plt.subplots(figsize=(25,5))\n",
  "\n",
  "plt.subplot(1, 5, 2)\n",
  "sns.boxplot(x=data['Age'])\n",
  "\n",
  "plt.subplot(1, 5, 3)\n",
  "sns.boxplot(x=data['Annual Income (k$)'])\n",
  "\n",
  "plt.subplot(1, 5, 4)\n",
  "sns.boxplot(x=data['Spending Score (1-100)'])\n",
  "\n",
  "plt.subplot(1, 5, 1)\n",
  "sns.boxplot(x=data['CustomerID'])"
]
```

```
},
{
  "cell_type": "markdown",
  "id": "6718d9d0",
  "metadata": {},
  "source": [
    "## Handling outlier"
  ]
}
```

```
{
  "cell_type": "code",
  "execution_count": 11,
  "id": "cf142f0b",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",
          "    vertical-align: middle;\n",
          "  }\n",
          "\n",
          "  .dataframe tbody tr th {\n",
          "    vertical-align: top;\n",
          "  }\n",
          "\n",
          "  .dataframe thead th {\n",
          "    text-align: right;\n",
          "  }\n",
          "</style>\n",
          "<table border='1' class='dataframe'>\n",
          "  <thead>\n",
          "    <tr style='text-align: right;'>\n",
          "      <th></th>\n",
          "      <th>CustomerID</th>\n",
          "      <th>Age</th>\n",
          "      <th>Annual Income (k$)</th>\n",
          "      <th>Spending Score (1-100)</th>\n",

```

```

" </tr>\n",
" </thead>\n",
" <tbody>\n",
" <tr>\n",
" <th>0.25</th>\n",
" <td>50.75</td>\n",
" <td>28.75</td>\n",
" <td>41.5</td>\n",
" <td>34.75</td>\n",
" </tr>\n",
" <tr>\n",
" <th>0.75</th>\n",
" <td>150.25</td>\n",
" <td>49.00</td>\n",
" <td>78.0</td>\n",
" <td>73.00</td>\n",
" </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
" CustomerID Age Annual Income (k$) Spending Score (1-100)\n",
"0.25 50.75 28.75 41.5 34.75\n",
"0.75 150.25 49.00 78.0 73.00"
]
},
"execution_count": 11,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"quant=data.quantile(q=[0.25,0.75])\n",
"quant"
]
},
{
"cell_type": "code",
"execution_count": 12,
"id": "a3843246",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"CustomerID 150.25\n",
"Age 49.00\n",
"Annual Income (k$) 78.00\n",
"Spending Score (1-100) 73.00\n",
"Name: 0.75, dtype: float64"
]
}
},
"execution_count": 12,
"metadata": {},

```

```

"output_type": "execute_result"
},
"source": [
"quant.loc[0.75]"
],
{
"cell_type": "code",
"execution_count": 13,
"id": "0aac4869",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"CustomerID      50.75\n",
"Age              28.75\n",
"Annual Income (k$)  41.50\n",
"Spending Score (1-100)  34.75\n",
"Name: 0.25, dtype: float64"
]
},
"execution_count": 13,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"quant.loc[0.25]"
],
{
"cell_type": "code",
"execution_count": 14,
"id": "9dd9649c",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"CustomerID      99.50\n",
"Age              20.25\n",
"Annual Income (k$)  36.50\n",
"Spending Score (1-100)  38.25\n",
"dtype: float64"
]
},
"execution_count": 14,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"iqr=quant.loc[0.75]-quant.loc[0.25]\n",

```

```

    "iqr"
  ]
},
{
  "cell_type": "code",
  "execution_count": 15,
  "id": "87d6fd39",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "CustomerID      -98.500\n",
          "Age              -1.625\n",
          "Annual Income (k$)  -13.250\n",
          "Spending Score (1-100) -22.625\n",
          "dtype: float64"
        ]
      },
      "execution_count": 15,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "low=quant.loc[0.25]-(1.5 *iqr)\n",
    "low"
  ]
},
{
  "cell_type": "code",
  "execution_count": 16,
  "id": "da5d37d2",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "CustomerID      299.500\n",
          "Age              79.375\n",
          "Annual Income (k$)  132.750\n",
          "Spending Score (1-100)  130.375\n",
          "dtype: float64"
        ]
      },
      "execution_count": 16,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "up=quant.loc[0.75]+(1.5 *iqr)\n",
    "up"
  ]
},

```



```
{
  "cell_type": "code",
  "execution_count": 18,
  "id": "d3f2fda9",
  "metadata": {},
  "outputs": [],
  "source": [
    "data['Annual Income (k$)'] = np.where(data['Annual Income (k$)']>132,60,data['Annual Income (k$)'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 19,
  "id": "5aeb5451",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "<AxesSubplot: xlabel='Annual Income (k$)'>"
        ]
      },
      "execution_count": 19,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAAWAAAEGCAYAAAABbzE8LAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfk9AAAA CXBIWXMAAAAsTAAALEwEAmpwYAAANNEIEQVR4nO3df5BdZXnA8e+TrAOBihqT0rgwLrJYCihCoyOFdij aVh2rqJ2RiAwzdmrrlGV12qFS/4ntP52hY5vZtnQobR3bDJ2O/ChFpbFaZBxHIVEI0QS8LajZRgiFqtNEFPL0j3M2 XC6bkE1273Pv3e9nZmf3vvfsPe+bvfnm7Nncs5GZSJL6b0X1BCRpuTLAkITEAEtSEQMsSUUMsCQVGvVlxmv WrMmJiYklmookjaZt27Y9nplre8cXFOCJiQm2bt26eLOSpGUgIr4137inICSpiAGWpCIGWJKKGGBJKmKAJamI AZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkoos6Hf CaenMzMzQ6XSqp7FoZmdnARgfHy+eycJNTk4yNTVVPQ0tAwZ4QHq6He7bsZOnT1hdPZVFsXLf9wH47pPD 9RRbue+J6iloGRmuvx0j7ukTVrP/zLdUT2NRrNr1aYChW8/cvKV+8BywJBUXwJJUxABLUhEDLEIFDLakFTH AKlTEAEtSEQMsSUUMsCQVMcCSVMQAS1IRayxJRQywJBUXwJJUxABLUhEDLEIFDLakFTHAKlTEAEtSE QMsSUUMsCQVMcCSVMQAS1IRayxJRQywJBUXwJJUxABLUhEDLEIFDLakFTHAKlTEAEtSEQMsSUUMs CQVMcCSVMQAS1IRayxJRQywJBUXwJJUxABLUhEDLEIFDLakFTHAKlTEAEtSEQMsSUUMsCQVMcCS VMQAS1IRayxJRQywJBUXwJJUxABLUhEDLEIFDLakFTHAKlTEAEtSEQMsSUUMsCQVMcCSVMQAS1IR AyxJRQywJBUXwJJUZKwfO5mdnWX//v1MT0/3Y3dDqdPpsOLHWT2NZW/Fj35Ap/NDn6s6qNPpsGrVqiV57O c9Ao6I90fE1ojYunfv3iWZhCQtR897BJyZNwA3AKxfv/6oDtHGx8cB2LRp09F8+rIwPT3Ntv96tHoay96B409i8h Un+1zVQUv53ZDngCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpi gCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIG WJKKGGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKA JamIAZakImP92Mnk5GQ/diNji24p+9WXAE9NTfVjN5K06JayX56CkKQiBliSihhgSSpigCWpiAGWpCIGWJKK GGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIA ZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYEk qYoAlqYgBlqQiBliSihhgSSpigCWpyFj1BPSMlfueYNWuT1dPY1Gs3Pc/AEO3npX7ngBOrp6GlkdPCAmJyerp7 CoZmefAmB8fNhdivLifS00uAzwgJiamqqegqQ+8xywJBUXwJJUxABLUhEDLEIFDLakFTHAKlTEAEtSEQMsS UUMsCQVMcCSVMQAS1IRayxJRQywJBUXwJJUxABLUhEDLEIFDLakFTHAKlTEAEtSEQMsSUUM49844i 9wLeWbjrzWgm83ud99sOorgtc27BybUvn5Zm5tndwQQGuEBFbM3N99TwW26iuC1zbsHJt/ecpCEkqYoAlqcgw
```

BPiG6gkskVFdF7i2YeXa+mzgzwFL0qgahiNgSRpJBliSigxMgCPi1Ij4jYGRFfj4jpdnx1RHw2Ir7Zvn9J9VyPVkS  
sjlivRcQd7e2RWfEvDgiPhkRu9qv3wWjsLal+FD7XNwRETdFxpHDvK6I+LuLeCwidnSNHXI9EXFtRHQi4sGI+  
LWaWT+/Q6zruvb5uD0ibo2IF3fdNzDrGpgAA08Bv5eZPwe8HvjdiDgL+DDwucw8A/hce3tYtQM7u26Pyto2AXd  
m5pnAuTRrHOq1RcQ4cDWwPjPPAVYClzHc6/o48KaesXnX0/7duww4u/2cv4qIlf2b6oJ8nOeu67PAOZn5auAh4  
FoYwHV15kC+Af8C/ArwILCuHVsHPFg9t6Nczyk0T/BLgDvasaFfG3AS8DDtD3S7xod6bcA48B1gNTAG3AH86  
gisawLY8XxfJ5pgXdu13b8BF1TP/0jX1XPfO4DNg7iuQTToCPigiJoDzgK8AJ2fmHoD2/U8XTu1Y/DlwDXCga2w  
U1vYKYC/w9+3plRs4kSGfG2ZOQv8KfBtYA/w/czcwPcvax6HWs/cP0Bzdrdjw+h9wGfajwdqXQMX4Ij4KeBm4  
IOZ+YPq+SyGiHgr8FhmbqueyxIYA84Hrs/M84D/Y7i+LZ9Xey707cBpwMuAEyPivbWz6quYZ2zo/s9qRHyE5vT  
m5rmheTYrW9dABTgiXkAT382ZeUs7/GhErGvvXwc8VjW/Y3Ah8LaLeAT4J+CSiPhHRmNtu4HdmfmV9vYnaY  
I87Gt7I/BwZu7NzJ8AtwC/wPCvq9eh1rMbOLVru1OA/+7z3I5JRFwJvBW4PNvzDQzYugYmwBERwN8COzPzY  
1133Q5c2X58Jc254aGSmdm5imZOUHZA4DPZ+Z7GY21fRf4Tkt8bDv0BuAbDP/avg28PiJOaJ+bb6D54eKwr6  
vXodZzO3BZRBwXEacBZwD3FMzvqETEm4A/AN6Wmfu67hqsdVWfPO86GX4RzbcC24H72re3AC+I+eHVN9  
v3q6vneozrvJhnfgg3EmsDXgNsbb92twEvGYW1AR8FdgE7gH8AjhvmdQE30ZzP/gnNkeBvHm49wEeA/6T5Qd2  
bq+e/wHV1aM71zrXkrwdxXb4UWZKKDMwpCElabgywJBUXwJJUxABLuhEDLElFDLaoioh3RERGXJkF+34k  
ItYc6figiIjzLuLG9uONEfH7h9l2Y8/ttRFx5xJPUQPMakvbBuCLNC8W0ZH5Q2DmcBtExFkRcTfwgYj4akRsAM  
jMvCceiLiwD/PUADLAAG5eg+NCmv/EflnX+MURcVfX9X43t68Mmzs6/WgblQfmjpx7jwTb6+lOtB/fFhHb2uvsv  
n8B85torzX8N+3nbomIVe19kxHx7xXfxfuX06NxxbvvByLi3V3r+UJE/HNEPBQRfxIRl0fEPe12p7fbrY2ImyPi3v  
btOZGmiBcCr87M++e577ci4jPtHdcCnwCub/+M7+3a9Dbg8iP9c9BoMcCacynNNX0fAp6liPO77jsP+CBwFs3Vz  
7pj9Hhmnk8Tl0N++93lfZn588B64OqleOkC5ngG8JeZeTbwv8C72vHN7fi5NNdr2AO8k+YVeufSXNfhurlrHrRj08  
CrgCuAV2bm64Abgal2m03An2Xma9v93DjPfNbTvEruWSLiKuDXgUszcz/wY5qrjK3IzP2Z2enafCvwiwv4M9AI  
McCas4HmKqG07zd03XdpZu7OzAM0L+uc6Lpv7qJJ23rGD+XqiLgf+DLNRVHOWMAcH87M+7r31x6Fjmfmr  
QCZ+aNsXvt/EXBTZj6dmY8CXwBe237uvZm5JzOfpHlJ6pZ2/IGuNbwR+Iuul/m+gEntfvqto7mUpzdrgDeDLyrf  
XxorknwKuCqiPjXiDi3a/vHaK62pmVorHoCqtcehV4CnBMRSfPbHzlirmk3ebJr86d59vPmyXnGn+LZ/7gf3+7nYp  
qwXZCZ+yLirr7jIDvPFYx/+UFOcx47+Mc6Lp9gGfWsKKd5/7DPM5+njv/HTRH3qfQXKiebK4tvCEi/ojm9MMt  
wOnt9se3j6NlyCNgaFwG8InMfHlmTmTmqTTxuOgoH+8RmktS0p7KOK0dfxHwvTa+Z9L86qljks01o3dHxKXt/o  
6LiBOAu4F3R/N7+NYCv8TCrnq1Bbhq7kZEvgAebXYCkz1jXwN+G7g9I17Wfu7Z7X0HaI7cT+za/pXMcxpDy4  
MBFjSnG27tGbsZeM9RPt7NwOr22/cP0PxOLOA7gbGI2A78Mc1piMVwBc2pje3A14CfoVnPduB+4PPANDlcOvN  
IXQ2sj+aXOn4D+J3eDTJzF/Ci3lMTmflFmvPhn2r/C907I+LLNL+ZYUv72HN+GfjUAualEeLV0KRjEBEfAn6Y  
mfP9kK53242ZubFn7G7g7Zn5vSWaogaYR8DSsbmeZ59TPpy7um+0p0Y+ZnyXL4+AJamIR8CSVMQAS1IRAy  
xJRQyWJBUXwJJU5P8BGkL5HVhWiA0AAAAASUVORK5CYII=",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.boxplot(x=data['Annual Income (k\$)'])"

]

},

{

"cell\_type": "markdown",

"id": "f37589fc",

"metadata": {},

"source": [

"# 7. Check for Categorical columns and perform encoding."

]

},

{

"cell\_type": "code",

"execution\_count": 20,

```

"id": "26658b84",
"metadata": {},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 200 entries, 0 to 199\n",
      "Data columns (total 5 columns):\n",
      " #   Column                Non-Null Count  Dtype \n",
      "---  -
      0   CustomerID            200 non-null   int64 \n",
      " 1   Gender                200 non-null   object\n",
      " 2   Age                  200 non-null   int64 \n",
      " 3   Annual Income (k$)    200 non-null   int64 \n",
      " 4   Spending Score (1-100) 200 non-null   int64 \n",
      "dtypes: int64(4), object(1)\n",
      "memory usage: 7.9+ KB\n"
    ]
  }
],
"source": [
  "data.info()"
],
},
{
  "cell_type": "code",
  "execution_count": 21,
  "id": "a1e0d1fc",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array(['Male', 'Female'], dtype=object)"
        ]
      },
      "execution_count": 21,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "data['Gender'].unique()"
  ],
},
{
  "cell_type": "code",
  "execution_count": 22,
  "id": "7f2f8f16",
  "metadata": {},
  "outputs": [],
  "source": [
    "data['Gender'].replace({'Male':1,'Female':0},inplace=True)"
  ]
}

```

```

]
},
{
  "cell_type": "code",
  "execution_count": 23,
  "id": "c4e8fbc7",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",
          "    vertical-align: middle;\n",
          "  }\n",
          "\n",
          "  .dataframe tbody tr th {\n",
          "    vertical-align: top;\n",
          "  }\n",
          "\n",
          "  .dataframe thead th {\n",
          "    text-align: right;\n",
          "  }\n",
          "</style>\n",
          "<table border='1' class='dataframe'>\n",
          "  <thead>\n",
          "    <tr style='text-align: right;'>\n",
          "      <th></th>\n",
          "      <th>CustomerID</th>\n",
          "      <th>Gender</th>\n",
          "      <th>Age</th>\n",
          "      <th>Annual Income (k$)</th>\n",
          "      <th>Spending Score (1-100)</th>\n",
          "    </tr>\n",
          "  </thead>\n",
          "  <tbody>\n",
          "    <tr>\n",
          "      <th>0</th>\n",
          "      <td>1</td>\n",
          "      <td>1</td>\n",
          "      <td>19</td>\n",
          "      <td>15</td>\n",
          "      <td>39</td>\n",
          "    </tr>\n",
          "    <tr>\n",
          "      <th>1</th>\n",
          "      <td>2</td>\n",
          "      <td>1</td>\n",
          "      <td>21</td>\n",
          "      <td>15</td>\n",
          "      <td>81</td>\n",
          "    </tr>\n",
          "    <tr>\n",
          "      <th>2</th>\n",

```

```

"    <td>3</td>\n",
"    <td>0</td>\n",
"    <td>20</td>\n",
"    <td>16</td>\n",
"    <td>6</td>\n",
" </tr>\n",
" <tr>\n",
"    <th>3</th>\n",
"    <td>4</td>\n",
"    <td>0</td>\n",
"    <td>23</td>\n",
"    <td>16</td>\n",
"    <td>77</td>\n",
" </tr>\n",
" <tr>\n",
"    <th>4</th>\n",
"    <td>5</td>\n",
"    <td>0</td>\n",
"    <td>31</td>\n",
"    <td>17</td>\n",
"    <td>40</td>\n",
" </tr>\n",
" <tr>\n",
"    <th>...</th>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
" </tr>\n",
" <tr>\n",
"    <th>195</th>\n",
"    <td>196</td>\n",
"    <td>0</td>\n",
"    <td>35</td>\n",
"    <td>120</td>\n",
"    <td>79</td>\n",
" </tr>\n",
" <tr>\n",
"    <th>196</th>\n",
"    <td>197</td>\n",
"    <td>0</td>\n",
"    <td>45</td>\n",
"    <td>126</td>\n",
"    <td>28</td>\n",
" </tr>\n",
" <tr>\n",
"    <th>197</th>\n",
"    <td>198</td>\n",
"    <td>1</td>\n",
"    <td>32</td>\n",
"    <td>126</td>\n",
"    <td>74</td>\n",
" </tr>\n",
" <tr>\n",

```

```

"    <th>198</th>\n",
"    <td>199</td>\n",
"    <td>1</td>\n",
"    <td>32</td>\n",
"    <td>60</td>\n",
"    <td>18</td>\n",
"  </tr>\n",
" <tr>\n",
"   <th>199</th>\n",
"   <td>200</td>\n",
"   <td>1</td>\n",
"   <td>30</td>\n",
"   <td>60</td>\n",
"   <td>83</td>\n",
" </tr>\n",
" </tbody>\n",
"</table>\n",
"<p>200 rows × 5 columns</p>\n",
"</div>"
],
"text/plain": [
"  CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)\n",
"0           1      1   19              15          39\n",
"1           2      1   21              15          81\n",
"2           3      0   20              16           6\n",
"3           4      0   23              16          77\n",
"4           5      0   31              17          40\n",
"..          ...      ...   ...              ...          ...\n",
"195         196      0   35              120          79\n",
"196         197      0   45              126          28\n",
"197         198      1   32              126          74\n",
"198         199      1   32               60          18\n",
"199         200      1   30               60          83\n",
"\n",
"[200 rows x 5 columns]"
]
},
"execution_count": 23,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"data"
]
},
{
"cell_type": "markdown",
"id": "8a984913",
"metadata": {},
"source": [
"# 8. Scaling the data"
]
},
{

```

```

"cell_type": "code",
"execution_count": 24,
"id": "8735fb43",
"metadata": {},
"outputs": [],
"source": [
    "from sklearn.preprocessing import MinMaxScaler\n",
    "sc=MinMaxScaler()"
]
},
{
    "cell_type": "code",
    "execution_count": 25,
    "id": "a3284529",
    "metadata": {},
    "outputs": [],
    "source": [
        "df=sc.fit_transform(data.iloc[:,1:])"
    ]
},
{
    "cell_type": "code",
    "execution_count": 26,
    "id": "2f29a211",
    "metadata": {},
    "outputs": [
        {
            "data": {
                "text/plain": [
                    "array([[1.        , 0.01923077, 0.        , 0.3877551 ],\n",
                    "       [1.        , 0.05769231, 0.        , 0.81632653],\n",
                    "       [0.        , 0.03846154, 0.00900901, 0.05102041],\n",
                    "       [0.        , 0.09615385, 0.00900901, 0.7755102 ],\n",
                    "       [0.        , 0.25        , 0.01801802, 0.39795918],\n",
                    "       [0.        , 0.07692308, 0.01801802, 0.76530612],\n",
                    "       [0.        , 0.32692308, 0.02702703, 0.05102041],\n",
                    "       [0.        , 0.09615385, 0.02702703, 0.94897959],\n",
                    "       [1.        , 0.88461538, 0.03603604, 0.02040816],\n",
                    "       [0.        , 0.23076923, 0.03603604, 0.7244898 ],\n",
                    "       [1.        , 0.94230769, 0.03603604, 0.13265306],\n",
                    "       [0.        , 0.32692308, 0.03603604, 1.        ],\n",
                    "       [0.        , 0.76923077, 0.04504505, 0.14285714],\n",
                    "       [0.        , 0.11538462, 0.04504505, 0.7755102 ],\n",
                    "       [1.        , 0.36538462, 0.04504505, 0.12244898],\n",
                    "       [1.        , 0.07692308, 0.04504505, 0.79591837],\n",
                    "       [0.        , 0.32692308, 0.05405405, 0.34693878],\n",
                    "       [1.        , 0.03846154, 0.05405405, 0.66326531],\n",
                    "       [1.        , 0.65384615, 0.07207207, 0.28571429],\n",
                    "       [0.        , 0.32692308, 0.07207207, 0.98979592],\n",
                    "       [1.        , 0.32692308, 0.08108108, 0.34693878],\n",
                    "       [1.        , 0.13461538, 0.08108108, 0.73469388],\n",
                    "       [0.        , 0.53846154, 0.09009009, 0.04081633],\n",
                    "       [1.        , 0.25        , 0.09009009, 0.73469388],\n",
                    "       [0.        , 0.69230769, 0.11711712, 0.13265306],\n",
                    "       [1.        , 0.21153846, 0.11711712, 0.82653061],\n"
```

" [0. , 0.51923077, 0.11711712, 0.31632653],\n",  
 " [1. , 0.32692308, 0.11711712, 0.6122449 ],\n",  
 " [0. , 0.42307692, 0.12612613, 0.30612245],\n",  
 " [0. , 0.09615385, 0.12612613, 0.87755102],\n",  
 " [1. , 0.80769231, 0.13513514, 0.03061224],\n",  
 " [0. , 0.05769231, 0.13513514, 0.73469388],\n",  
 " [1. , 0.67307692, 0.16216216, 0.03061224],\n",  
 " [1. , 0. , 0.16216216, 0.92857143],\n",  
 " [0. , 0.59615385, 0.16216216, 0.13265306],\n",  
 " [0. , 0.05769231, 0.16216216, 0.81632653],\n",  
 " [0. , 0.46153846, 0.17117117, 0.16326531],\n",  
 " [0. , 0.23076923, 0.17117117, 0.73469388],\n",  
 " [0. , 0.34615385, 0.1981982 , 0.25510204],\n",  
 " [0. , 0.03846154, 0.1981982 , 0.75510204],\n",  
 " [0. , 0.90384615, 0.20720721, 0.34693878],\n",  
 " [1. , 0.11538462, 0.20720721, 0.92857143],\n",  
 " [1. , 0.57692308, 0.21621622, 0.35714286],\n",  
 " [0. , 0.25 , 0.21621622, 0.6122449 ],\n",  
 " [0. , 0.59615385, 0.21621622, 0.2755102 ],\n",  
 " [0. , 0.11538462, 0.21621622, 0.65306122],\n",  
 " [0. , 0.61538462, 0.22522523, 0.55102041],\n",  
 " [0. , 0.17307692, 0.22522523, 0.46938776],\n",  
 " [0. , 0.21153846, 0.22522523, 0.41836735],\n",  
 " [0. , 0.25 , 0.22522523, 0.41836735],\n",  
 " [0. , 0.59615385, 0.24324324, 0.52040816],\n",  
 " [1. , 0.28846154, 0.24324324, 0.60204082],\n",  
 " [0. , 0.25 , 0.25225225, 0.54081633],\n",  
 " [1. , 0.78846154, 0.25225225, 0.60204082],\n",  
 " [0. , 0.61538462, 0.25225225, 0.44897959],\n",  
 " [1. , 0.55769231, 0.25225225, 0.40816327],\n",  
 " [0. , 0.63461538, 0.26126126, 0.5 ],\n",  
 " [1. , 0.98076923, 0.26126126, 0.45918367],\n",  
 " [0. , 0.17307692, 0.27927928, 0.51020408],\n",  
 " [1. , 0.67307692, 0.27927928, 0.45918367],\n",  
 " [1. , 1. , 0.27927928, 0.56122449],\n",  
 " [1. , 0.01923077, 0.27927928, 0.55102041],\n",  
 " [0. , 0.94230769, 0.28828829, 0.52040816],\n",  
 " [0. , 0.69230769, 0.28828829, 0.59183673],\n",  
 " [1. , 0.86538462, 0.2972973 , 0.51020408],\n",  
 " [1. , 0. , 0.2972973 , 0.59183673],\n",  
 " [0. , 0.48076923, 0.2972973 , 0.5 ],\n",  
 " [0. , 0.96153846, 0.2972973 , 0.47959184],\n",  
 " [1. , 0.01923077, 0.2972973 , 0.59183673],\n",  
 " [0. , 0.26923077, 0.2972973 , 0.46938776],\n",  
 " [1. , 1. , 0.30630631, 0.55102041],\n",  
 " [0. , 0.55769231, 0.30630631, 0.41836735],\n",  
 " [0. , 0.80769231, 0.31531532, 0.48979592],\n",  
 " [0. , 0.80769231, 0.31531532, 0.56122449],\n",  
 " [1. , 0.78846154, 0.35135135, 0.46938776],\n",  
 " [1. , 0.15384615, 0.35135135, 0.54081633],\n",  
 " [0. , 0.51923077, 0.35135135, 0.53061224],\n",  
 " [1. , 0.42307692, 0.35135135, 0.47959184],\n",  
 " [0. , 0.09615385, 0.35135135, 0.52040816],\n",  
 " [0. , 0.59615385, 0.35135135, 0.41836735],\n",  
 " [1. , 0.75 , 0.35135135, 0.51020408],\n",



" [1. , 0.38461538, 0.35135135, 0.55102041],\n",  
 " [1. , 0.94230769, 0.35135135, 0.40816327],\n",  
 " [0. , 0.53846154, 0.35135135, 0.43877551],\n",  
 " [0. , 0.05769231, 0.35135135, 0.57142857],\n",  
 " [1. , 0.57692308, 0.35135135, 0.45918367],\n",  
 " [0. , 0.71153846, 0.37837838, 0.58163265],\n",  
 " [0. , 0.07692308, 0.37837838, 0.55102041],\n",  
 " [0. , 0.30769231, 0.38738739, 0.60204082],\n",  
 " [0. , 0.61538462, 0.38738739, 0.45918367],\n",  
 " [0. , 0.96153846, 0.3963964 , 0.55102041],\n",  
 " [1. , 0. , 0.3963964 , 0.40816327],\n",  
 " [1. , 0.57692308, 0.40540541, 0.48979592],\n",  
 " [0. , 0.42307692, 0.40540541, 0.39795918],\n",  
 " [0. , 0.26923077, 0.40540541, 0.41836735],\n",  
 " [1. , 0.11538462, 0.40540541, 0.52040816],\n",  
 " [0. , 0.55769231, 0.40540541, 0.46938776],\n",  
 " [0. , 0.17307692, 0.40540541, 0.5 ],\n",  
 " [1. , 0.57692308, 0.41441441, 0.41836735],\n",  
 " [1. , 0.03846154, 0.41441441, 0.48979592],\n",  
 " [0. , 0.09615385, 0.42342342, 0.40816327],\n",  
 " [0. , 0.59615385, 0.42342342, 0.47959184],\n",  
 " [1. , 0.94230769, 0.42342342, 0.59183673],\n",  
 " [1. , 0.15384615, 0.42342342, 0.55102041],\n",  
 " [1. , 0.59615385, 0.42342342, 0.56122449],\n",  
 " [0. , 0.05769231, 0.42342342, 0.41836735],\n",  
 " [0. , 0.92307692, 0.43243243, 0.5 ],\n",  
 " [1. , 0.69230769, 0.43243243, 0.45918367],\n",  
 " [1. , 0.96153846, 0.43243243, 0.42857143],\n",  
 " [1. , 0.92307692, 0.43243243, 0.47959184],\n",  
 " [1. , 0.90384615, 0.43243243, 0.52040816],\n",  
 " [0. , 0.01923077, 0.43243243, 0.54081633],\n",  
 " [0. , 0.38461538, 0.44144144, 0.41836735],\n",  
 " [1. , 0.01923077, 0.44144144, 0.45918367],\n",  
 " [0. , 0. , 0.45045045, 0.47959184],\n",  
 " [0. , 0.01923077, 0.45045045, 0.5 ],\n",  
 " [0. , 0.86538462, 0.45045045, 0.42857143],\n",  
 " [0. , 0.59615385, 0.45045045, 0.59183673],\n",  
 " [0. , 0.63461538, 0.46846847, 0.42857143],\n",  
 " [0. , 0.61538462, 0.46846847, 0.57142857],\n",  
 " [1. , 0.17307692, 0.46846847, 0.56122449],\n",  
 " [0. , 0.38461538, 0.46846847, 0.39795918],\n",  
 " [0. , 0.42307692, 0.48648649, 0.58163265],\n",  
 " [1. , 0.40384615, 0.48648649, 0.91836735],\n",  
 " [0. , 0.09615385, 0.4954955 , 0.28571429],\n",  
 " [0. , 0.25 , 0.4954955 , 0.7755102 ],\n",  
 " [1. , 0.48076923, 0.5045045 , 0.34693878],\n",  
 " [1. , 0.42307692, 0.5045045 , 0.95918367],\n",  
 " [1. , 0.78846154, 0.5045045 , 0.10204082],\n",  
 " [1. , 0.38461538, 0.5045045 , 0.75510204],\n",  
 " [1. , 0.55769231, 0.5045045 , 0.08163265],\n",  
 " [1. , 0.40384615, 0.5045045 , 0.75510204],\n",  
 " [0. , 0.13461538, 0.51351351, 0.33673469],\n",  
 " [0. , 0.25 , 0.51351351, 0.71428571],\n",  
 " [1. , 0.03846154, 0.52252252, 0.04081633],\n",  
 " [0. , 0.21153846, 0.52252252, 0.8877551 ],

" [0. , 0.5 , 0.52252252, 0.06122449],\n",  
" [1. , 0.26923077, 0.52252252, 0.73469388],\n",  
" [1. , 0.01923077, 0.53153153, 0.09183673],\n",  
" [0. , 0.32692308, 0.53153153, 0.7244898 ],\n",  
" [0. , 0.75 , 0.54054054, 0.04081633],\n",  
" [1. , 0.26923077, 0.54054054, 0.93877551],\n",  
" [0. , 0.19230769, 0.54954955, 0.39795918],\n",  
" [0. , 0.26923077, 0.54954955, 0.87755102],\n",  
" [1. , 0.13461538, 0.55855856, 0.1122449 ],\n",  
" [1. , 0.19230769, 0.55855856, 0.97959184],\n",  
" [1. , 0.57692308, 0.55855856, 0.35714286],\n",  
" [0. , 0.26923077, 0.55855856, 0.74489796],\n",  
" [0. , 0.30769231, 0.56756757, 0.21428571],\n",  
" [1. , 0.30769231, 0.56756757, 0.90816327],\n",  
" [1. , 0.48076923, 0.56756757, 0.16326531],\n",  
" [1. , 0.40384615, 0.56756757, 0.8877551 ],\n",  
" [0. , 0.5 , 0.56756757, 0.19387755],\n",  
" [0. , 0.38461538, 0.56756757, 0.76530612],\n",  
" [0. , 0.55769231, 0.56756757, 0.15306122],\n",  
" [0. , 0.17307692, 0.56756757, 0.89795918],\n",  
" [1. , 0.36538462, 0.56756757, 0. ],\n",  
" [0. , 0.23076923, 0.56756757, 0.78571429],\n",  
" [1. , 0.30769231, 0.56756757, 0. ],\n",  
" [0. , 0.23076923, 0.56756757, 0.73469388],\n",  
" [0. , 0.73076923, 0.57657658, 0.34693878],\n",  
" [0. , 0.21153846, 0.57657658, 0.83673469],\n",  
" [1. , 0.01923077, 0.59459459, 0.04081633],\n",  
" [0. , 0.25 , 0.59459459, 0.93877551],\n",  
" [1. , 0.61538462, 0.63063063, 0.25510204],\n",  
" [0. , 0.34615385, 0.63063063, 0.75510204],\n",  
" [1. , 0.46153846, 0.63963964, 0.19387755],\n",  
" [0. , 0.28846154, 0.63963964, 0.95918367],\n",  
" [0. , 0.34615385, 0.64864865, 0.26530612],\n",  
" [1. , 0.26923077, 0.64864865, 0.63265306],\n",  
" [1. , 0.42307692, 0.64864865, 0.12244898],\n",  
" [1. , 0.19230769, 0.64864865, 0.75510204],\n",  
" [1. , 0.34615385, 0.64864865, 0.09183673],\n",  
" [1. , 0.34615385, 0.64864865, 0.92857143],\n",  
" [0. , 0.65384615, 0.65765766, 0.12244898],\n",  
" [0. , 0.23076923, 0.65765766, 0.86734694],\n",  
" [1. , 0.76923077, 0.65765766, 0.14285714],\n",  
" [1. , 0.17307692, 0.65765766, 0.69387755],\n",  
" [1. , 0.78846154, 0.7027027 , 0.13265306],\n",  
" [1. , 0.32692308, 0.7027027 , 0.90816327],\n",  
" [0. , 0.36538462, 0.73873874, 0.31632653],\n",  
" [0. , 0.26923077, 0.73873874, 0.86734694],\n",  
" [1. , 0.53846154, 0.74774775, 0.14285714],\n",  
" [0. , 0.21153846, 0.74774775, 0.8877551 ],\n",  
" [0. , 0.44230769, 0.75675676, 0.3877551 ],\n",  
" [1. , 0.23076923, 0.75675676, 0.97959184],\n",  
" [0. , 0.69230769, 0.77477477, 0.23469388],\n",  
" [1. , 0.19230769, 0.77477477, 0.68367347],\n",  
" [0. , 0.44230769, 0.79279279, 0.16326531],\n",  
" [0. , 0.34615385, 0.79279279, 0.85714286],\n",  
" [0. , 0.30769231, 0.79279279, 0.2244898 ],\n",

```

"      [0.      , 0.26923077, 0.79279279, 0.69387755],\n",
"      [1.      , 0.28846154, 0.88288288, 0.07142857],\n",
"      [0.      , 0.38461538, 0.88288288, 0.91836735],\n",
"      [0.      , 0.55769231, 0.94594595, 0.15306122],\n",
"      [0.      , 0.32692308, 0.94594595, 0.79591837],\n",
"      [0.      , 0.51923077, 1.      , 0.2755102 ],\n",
"      [1.      , 0.26923077, 1.      , 0.74489796],\n",
"      [1.      , 0.26923077, 0.40540541, 0.17346939],\n",
"      [1.      , 0.23076923, 0.40540541, 0.83673469]])"
]
},
"execution_count": 26,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"df"
]
},
{
"cell_type": "markdown",
"id": "8395e4ec",
"metadata": {},
"source": [
"# 9. Perform any of the clustering algorithms"
]
},
{
"cell_type": "markdown",
"id": "494a1b9c",
"metadata": {},
"source": [
"##### Kmeans_clustering"
]
},
{
"cell_type": "code",
"execution_count": 27,
"id": "e53db14b",
"metadata": {},
"outputs": [],
"source": [
"from sklearn.cluster import KMeans"
]
},
{
"cell_type": "code",
"execution_count": 28,
"id": "bcad85b9",
"metadata": {},
"outputs": [],
"source": [
"TWSS=[]\n",
"k=list(range(2,9))\n",

```

```
"\n",
"for i in k:\n",
"    kmeans=KMeans(n_clusters=i,init='k-means++')\n",
"    kmeans.fit(data)\n",
"    TWSS.append(kmeans.inertia_)"
],
{
"cell_type": "code",
"execution_count": 29,
"id": "596f5476",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"[381550.6840684068,\n",
" 268082.56760639744,\n",
" 191612.56821803437,\n",
" 153394.66603206735,\n",
" 119223.63779954854,\n",
" 101364.2432178932,\n",
" 85819.89345888031]"
]
}
},
"execution_count": 29,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"TWSS"
]
},
{
"cell_type": "code",
"execution_count": 30,
"id": "a91c6ab3",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"Text(0, 0.5, 'TWSS')"
]
}
},
"execution_count": 30,
"metadata": {},
"output_type": "execute_result"
}
],
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAAZgAAAEgcAYAAABYV4NmAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAA CXBIWXMAAAAsTAAALEwEAmpwYAAAOU0lEOVR4nO3de5yVZbn/8c93QBE8ICq6URA0KVNTzAkt7eARr
```

DylJkqKShszM612qdkvPESZGSaZJqmJRgoeSrY7NdJMMxUGxQOKSSqCkrl3iiiKHK7fH/c9zYKGCzIZNc+sNd/36/W81rPu57Cup73lmmvw3LciAjMzs7ZWU3QAZmZWnXxgzMysLJxgzMysLJxgzMysLJxgzMysLLOWHUBHscUWW8SAAQOKDsPMrKLMmDHjfyOid2PHnGCyAQMGUfDXV3QYZmYVRdLctR1ze5mZmZWFE4yZmZWFE4yZmZWFE4yZmZWFE4yZmZWFE0xrTZwIAwZATU36nDix6IjMzDoED1NujYkTYdQoWLo0fZ87N30HG D68uLjMzDoA12Ba49xzG5JLvaVLU7mZWSfnBNMaL720buVmZp2IE0xrbLvtupWbmXUiTjCtMWYM9Oixeln 37qnczKyTc4JpjeHDYfx46N8fpLQNGADHHlt0ZGZmhXOCaa3hw+HFF2HVqpRs/vlPeO65oqMyMyucE0xbGjk Snn0WPvShoiMxMyucE0xbkqB371SbmTw5fZqZdVJOMOVwxxlwzDFwxRVR2JmVhgnmHI45BA4+GA46yz4 xz+KjsbMrBBOMOUgpQ7/9daDk092U5mZdUpOMOXSty9ceincfz9cfnnR0ZiZtTtPdIlOJ54I06bBrrsWHYmZWbt zgiknCa68sugozMwK4Say9rBiBXznOzBuXNGRmJm1GyeY9tClCzz9NJx9tt/yN7N0wwmmPdSPKuvWdu46CV auLDoiM7Oyc4JpL1tvDZddBg8+6KYyM+sUnGDa0/HHp5cwL7gAliwpOhozs7IqW4KRtlGkaZlelZRL0vm5/DxJ L0uambfPlloxjqQ5kp6VNKSkfA9JT+Zj4yQpl3eTNCmXPYJpQMk1IyQ9l7cR5XrOdSLBvVeiWszGGxcdjZlZWZ WzBrMM2C8idgMGAUMl7ZWPXRORg/L2BwBJOWHDgJ2BocAVkrk868ERgED8zY0l48EXo+IHYBLgR/ne20 GjAb2BAYDoyX1KuOzNI+fPrDTTmn/+eeLjCXMrlZKlmaieSt/XS9v0cQlhwE3RcSyiHgBmAMMltQH2CQiHoqI AK4HDi+5ZkLevwXYP9duhgBTI2JRRLwOTKUhKXUMl10GO++cpvc3M6tCZe2DkdRF0kzgnDI/+I/kQ1+T9IS ka0tqFtsA80oun5/Ltsn7a5avdk1ErAAWA5s3ca814xslqU5S3cKFC1v+oC1xzDFpueUTT/SoMjOrSmVNMBGxMiI GAX1JtZfDSM1dHyA1my0AfppPV2O3aKK8pdeUxjc+ImojorZ3795NPEkZ/Md/wM9/Dg8/DGPHtu9vm5mlg3Y ZRRYRbwD3AUMj4tWceFYBvyL1kUCqZfQruawv8Eou79tI+WrxSoOK9AQWNXGvjuXYY+GII+D//T945pmio zEzalPIHEXWW9Kmeb87cAAwO/ep1DsCeCrvTwGG5ZFh25E686dFxAJgiaS9cv/KCcDtJdfUjxA7Crg399PcDR wkqVdugjsol3Us9XOVbbstzJ1bdDRmZm2qnJnd9gEm5JFgNcDkiLhD0g2SBpGarF4ETgGliFmSJgNPAYuA0yKi vnPiVOA6oDtwZ94ArgFukDSHVHMZlu+1SNKFwPR83gURsaiMz9pyW22Vai9durz/uWZmFUTpD36rra2Nurq6 4gJYtSqTG3PAAQ3DmM3MOjhJMyKitrFjfpO/o1i0KL3hf+KJafZIM7MK5wTTUWyxBfziFzB9OlxySdHRmJm1 mhNMR/LFL8KRR8Lo0TBrVtHRmJm1ihNMRyLBFVfAJpvAySeD+8fMrIJ5yeSOZsst4brrYMMNU8IxM6tQTj Ad0ec+17C/fDmst15xsZiZtZCbyDqyH/019tknJRkzswrjBNORffCDMG0aXHxx0ZGYma0zJ5iO7Mgj06zL558PTz xRdDRmZuvEcaaju/xy6NUrvYDppjIzqyBOMB3dFlukCTFnzUovYZqZVQiPIqsEX/gC/OMf0Lfv+59rZtZBuAZT KeqTy113uanMzCqCE0wlmTYNDj44DV82M+vgnGAqyeDBcNxxcOGFMHNM0dGYmTXJCabSjBsHm2+eRpW 9917R0ZiZrZUTTKXZfHO46ip4/HH44Q+LjsbMbK2cYCrYYfB6afDjjsWHYmZ2Vp5mHKlGjeu6AjMzJrkGkw li0iJxk1lZtYBOcFUMglmzIDvfz99mpl1IE4wle5nP4OttkqjypYtKzoaM7N/cYKpdL16wfxj8NRT6f0YM7MOWgm mGnzuc6kGc/HFMH9+0dGYmQFOMNXj0kth6lRPiGlmHUbZEoykDSRNk/S4pFmSzs/lm0maKum5/Nmr5JpzJM 2R9KyKISXle0h6Mh8bJ0m5vJukSbn8EUkDSq4ZkX/jOUkjyvWcHcamm8KnP532X3650FDMzKC8NZhlwH4RsR swCBgqaS/gbOCeiBgI3JO/I2knYBiwMzAUuEJSI3yvK4FRwMC8Dc3Ii4HXI2IH4FLgx/lemwGjgT2BwcDo0kRW 1W6+Gbbf3mvHmFnhypZgInkrf10vbwEcBkzI5ROAw/P+YcBNEBesI14A5gCDJfUBNomlhyIlgOvXuKb+XrcA+ +fazRBgakQsiojXgak0JKXqduCB0L36pN5992iozGzTqysfTCSukiaCbxG+gf/EWCriFgAkD+3zKdvA8wruXx+Lt sm769Zvto1EbECWAXs3sS91oxvlKQ6SXULFy5sxZN2IJtuCr/6FTz9NJx/ftHRmFknVtYEEExErl2IQ0JdUG9mlidP V2C2aKG/pNaXxjY+I2oio7d27dxOhVZiDD4aTT06jyqZNKzoaM+uk2mUUWUS8AdxHaqZ6NTd7kt9fy6fNB/q VXYNYXcCWx922kfLvrJHUFegKLmrhX5zf2bOqLefrpoiMxs06qnKPIekvaNO93Bw4AZgNTgPpRXSOA2/P+F GBYHhm2Hakzfl1puRlsiaa/cv3LCGtfU3+so4N7cT3M3cJcKXrlz/6Bc1nn07AmzZqW+GDOzApRzNuU+wIQ8Eq wGmBwRd0h6CJgsaSTwEnA0QETMkjQZeBpYAZwWESvzvU4FrgO6A3fmDeAa4AZJc0g1l2H5XoskXQjUD6 W6ICIWlFfZO6b110+ft94K/fqIFTHNzNqJ0h/8VlthG3V1dUWH0fbeeSetG9OjBzz2GGywQdERmVkvkTQjImob O+Y3+atd9+5w9dUwe3aaddnMrJ04wXQGBx4Io0bBJZfAQw8VHY2ZdRJOMJ3FJZfAttt6Wn8zazdeMrmz2Hhj+ PWvYcGChs5/M7MycOLpTPbdt2F/5Uro0mXt55qZtZKbyDqj66+H2to0wszMrEycYDqjvn1h5kz43veKjsTMqpgT TGe0335w6qlpkbIHhYw6GjOrUk4wndXFF0P//mlU2dKlRUdjZlXICAaz2mgjuPZaeP55uOuuqMxsyrkUWSd2b77 wrPPwg47FB2JmVUh12A6u/rk8sgj8PbbxcZiZlXFCcbgxRfh4x+HPn2gpgYGDICJE4uOyswqnJvILl0k69IFliXJ3+f OTXOXAQwfXlxcZlBRXIMxOPdcWLFi9bKIS1O5mVklOcEYvPTSupWbmTWDE4ylWZbXpdzMrBmcYAzGj EkrXpbq0QNOOQXeeKOQkMys8jnBWOrIHZ8+vdkvpc+f/zxNJTNkCCxeXHSEZlaBPiRMkuHD/33E2Oabw1FH wdChcPfdsMkmxcRmZhXJNRhbu8MOg8mToa4uJZn6YcxmZs3gBGNNO+IImDQJpk1LE2SamTWTm8js/X3hC3 DfftB4cNGRmFkFcQ3GmmeffWD99WHhwjS67K23io7IzDq4siUYSf0k/VnSM5JmSTojl58n6WVJM/P22ZJrzpE 0R9KzkoaUI08h6cl8bJwk5fJukibl8ckDSi5ZoSk5/I2olzP2elMmwBXXAOf/7wnxzSzJpWzBrMC+FZEfBjYCzhN 0k752KURMShvfvDIx4YBOwNDgSskdcnnXwmMAGbmbWguHwm8HhE7AJcCP8732gwYDewJDAZGS+pVxm ftPD73OfjNb+CBBIKS8WJlZrYwa00wkg6R1L/k+/clPS5piqTt3u/GEbEgIh7N+0uAZ4BtmrjkMOCmiFgWES8Ac 4DBkvoAm0TEQxERwPXA4SXXTMj7twD759rNEGBqRCyKiNeBqTQkJWutYcPghhvg/vudZMxsRZqqwYwBFg JI+jzwJeBkYArwy3X5kdx0tTvWSC76mqQnJF1bUrPYBphXctn8XLZN3l+zfLvrImIFsBjYvIl7rRnXKEl1kuoWLl y4Lo9kxx0H118PCxb4RUwza1RTCSYioV5P0y8A10TEjli4Gujd3B+QtBFwK3BmRLxJau76ADAIWAD8tP7Uxm Jooryl1zQURlyPiNqIqO3du9mPZPWGD4fHH0/ryKxYae++W3REZtaBNJVgJGkjSTXA/sA9Jcc2aM7NJa1HSi4T I+I2gIh4NSJWRsQq4FekPhJItYx+JZf3BV7J5X0bKV/tGklDgZ7AoibuZW1t/fUhaK46CQ49FN55p+ilzKyDaCrB/ AyYCdQBz0REHYck3Uk1jyblvpBr8rVjs8r7Ijx2BPBU3p8CDMSjw7YjdeZPi4gFwBJJe+V7ngDeXnJN/Qixo4B7

cz/N3cBBknrlJriDcpmVgwT77Qd/+IN6MdM1GTOjiRctI+JaSXcDWwKPlxxaAJzUjHvvDRwPPClpZi77LnCspEG  
kJqsXgVPy782SNB14mjQC7bSIWJmvOxW4DugO3Jk3SAnsBklzSDWXYfleiyRdCEzP510QEYuaEbO11EknPZr  
MyJEpyfzud7BBSyq6ZlallP7gb+RAGkH2RkQszt/3JY3emgtcHhHvtVeQ7aG2tjbq6uqKDqPyXX01/Od/wtFHp3nM  
zKyqSZoREbWNHWtqqpjPcCasxbnGcTPwI2A34Argy20cp1WDL385NZkNHfH0JGZWskYSTPeIqO8Y/xJwbUT  
8NHf6zyx7ZFa5Ro5s2L/jDjjwQOjWrbh4zKwQTY4iK9nfjzyKLI/+Mnt/Tz0FhxwCX/wivFdVLapm1gxNJZg/S5os  
6TKgF3Av/GsUmP+1sPe3yy5w+eUwZYqTjFkn1FSCOQO4jTTSa5+IWJ7L/wM4t8xxWbU47TQYNw5uvz1NMb  
N8+ftfY2ZVoak+mDOAB4Fb8jQsAETeY2WPyqrL6afDqLVw5plw112p2czMql5TCaYvcBmwo6QngL+REs5Dfqf  
E1tkZZ8CnPgW77150JGbWTtbaRBYR/xURnyA1iX2X9CLjycBTkp5up/ismtQnl7/9LQ1nXrGi6fPnRKi1Zz2Y7sA  
mpHm+epLm9HqkySvMmjJ9elq07PjjnWTMqtham8gkjSct/rWEIFD+BozN66uYtdwZZ6QRZd/5DtTUpGn/u3R5/+  
vMrKI01QezLdANeA54mTRD8RvtEJN1Bt/+dur4P/vs9Ob/hAlOMmZVpqNJLofm2Yt3Bj4BfAvYRdliUkf/6HaK0  
arVWWelJPPYY+nTCcasqjRVgyFPff+UpDdlQ0UuBj5PXue+7NFZ9TvnJRcampg0SLo2dOJxqxKrLWTX9J/Sbp  
J0jzgfIjJieZa0uuVm7RSfdQY1NfDWW7D33ml02SrPRmRWDZqqwXwV+A7wjbz0l1n5bLQRHHMMnH9+Sji/+1  
X6NLOK1VSCWRwRt7RbJGajR6fay4UXpuRy1VvOMmYVrKkE01vSN9d2sHQZZLM2IaUaTAT84AfQrx98/t  
FR2VmLdRUgukCbNzegZgBKclccAH07p2azMysYjWVYBZExPntFolZPQm+/vW0v3w53HADnHRSKjezitHcBc  
fMijF5cl0h8/TTU9OZmVWMpmow+7dbFGZrc9xxMHMmXHJJ6vC/7DLXZMwqRFNv8ntKfiueBBdfnEaXjR2bk  
syllzrJmFUajwG1jk9KNZgzz4Rrr4UXXig6IjNrBicYqwxSqsHMnAnbb190NGbWDGVLMLJ6SfqzPcckzZJ0Ri7f  
TNJUSc/lz14l15wjaY6kZyUNKSnfQ9KT+di4PAknkrpJmpTLH5E0oOSaEf3npM0olzPae1IakguY8emGZnd8W/  
WYZWzBrMC+FZEfBjYCzhN0k7A2cA9ETEQuCd/Jx8bRpq9eShwhaT6WQ+vBEYBA/M2NJePBF6PiB2AS4Ef  
53ttRpqMc0/yxJylicwqXERqJrvkkjTdv5OMWYdUtgQTEQsi4tG8vwR4BtgGOAyYkE+bABye9w8DboqIZRHxAj  
AHGCypD7BJRDyUZ3e+fo1r6u91C7B/rt0MAaZGxKK8QNpUGpKSVToJxo2DU09NAwC++10nGbMOqMnp+tt  
KbrranbQy5lb1k2dGxAJJW+bTtgEeLrlsfi5bnvfXLK+/Zl6+1wpJi4HNS8sbucaqgQSXX54Sy0UXpSn+f/CDoqMys  
xJITzCSNgJuBc6MiDe19uGljR2IJspek1pbKNITW9su+22a4vLOqqaGvjFL1KS2XrroqMxsZwUdRSZpPVIyWVi  
RNyWi1/NzV7kz9dy+XygX8nlfYFXcnnfRspXu0ZSV6AnsKiJe60mIsZHRG1E1Pbu3bulj2lFqqmBK6+Er341fZ87  
t9h4zOxfymKTMA1wDNrzLw8Bagf1TUCuL2kffgeGbYdqTN/Wm5OWyJpr3zPE9a4pv5eRwH35n6au4GDJPX  
KnfsH5TKrRvW14qefhp12giOPhAEDUvIZMAAamTiwyOrNOq5xNZHsDxwNPSpqZy74LXARMLjQSeAk4GiAiZ  
kmaDDxNGoF2WkSszNedClwHdAfuzBukBHaDpDmkmsuwfK9Fki4EpufzLvDMBJ3AjjvCHnvAbbe1lM2dC6NG  
pf3hw4uJy6yTUnj0DQC1tbVRV1dXdBjWWv37w0svNV7+4ovtHo5ZtZM0IyJqGzvmN/mtusybl3h5Y0nHzMrKC  
caqy9pGA26+OfzsZ/DOO+0ajlln5grJlWXMGOjRY/WyHj1S/8w3vgEf+EB6SfPdd4uJz6wTcYKx6jJ8OIwfn/pcp  
PQ5fjw88ADcdx988INwxhkp0dx8c9HRmlU1JxirPsOHpw79VavSZ/3osU9/OiWZe+9NCaZbt1T+9tuwbFlBwZpV  
LycY63z23Rf+8hc45JD0/aKLYOBA+OUv4b33io3Nrlo4wVjnJDW8oLnfftC3b5o8c+DA1KTmRGPWak4wZvvuC  
w8+CHfdBX36wCmnpGRjZq3SLrMpm3V4EgwZAgcdBHffDdvkybfnzEnNaSecAOutV2yMZhXGNRizUhIMHQ  
of+Uj6/pvfwJe/nIY5//rXsGJfSfGZVRAnGLOmjB4N//3f0KsXnHxySjS//W3RUZIVBCcYs6ZI8PnPW/TpcPvtsMk  
mqb+m3qpVxcVm1sE5wZg1hwSHHgozZsBPfpLKHngAPvzhtBzAypVNX2/WCTnBmK0LafWpaDbYAL70Jdhlf  
7jxRicasxJOMGYt9clPwmOPpSlnunaF445LQ569BIYZ4ARj1jo1NXDUUfD44zBpUhoIIKWazJQp7qOxTs0Jxqwt  
1NTAF78IJ56Yvk+ZAocdBrvtBrfc4kRjnZITjFk5HHpoGs68fDkcfTTsvntaytmJxjoRjixzcujsBY49FmbNSi9rvvsun  
HOOE4x1Kk4wZuXUpUtaLmDWrDQFTdeu8NZbaUqa22/3gACrak4wZu2ha1cYMCDtv/giPP88HH441NammQK  
caKwKOcGYtbdddoHZs9PcZm+8kfprBg+GxYuLjsysTTnBmBWha9c04mz2bLjmGth5Z+jZMx2bPds1GqsKTjBm  
RVpvvfTuzHXXpe8vvwyDBsEnPpH6bCZOTE1rNTXpc+LE4mI1W0deD8asI+ndG8aNgzFj0rIBNTUNI8/mzoVR  
o9L+8OHFxWjWTGWrwUi6VtJrkp4qKTtP0suSZubtsyXHzpE0R9KzkoaU1O8h6cl8bJyU1rmV1E3SpFz+iKQBjd  
eMkPRc3kaU6xnN2tz666ck8ve/w2ab/fuw5qVL4dxzi4nNbB2Vs4nsOmBoI+WXRsgvP0BQNJOWDBg53zNFZK  
65POvBEYBA/NWf8+RwOsRsQNwKfdjK/NgNHAnsBgYLSkXm3/eGZl1K0bvP5648fmzoUIS9o3HrMWKFuCi  
Yj7gUXNPP0w4KaIWBRYLwBzgMGS+gCbRMRDERHA9cDhJddMyPu3APvn2s0QYGpELIqI14GpNJ7ozDq2b  
bdd+7F+/eC5s1KfjVkhVUQn/9ckPZGb0OprFtsA80rOmZ/Ltsn7a5avdk1ErAAWA5s3ca9/I2mUpDpJdQsXLmzdU  
5m1tTFjVl8aANL3885LL2pecknq+P/97wsIzuz9tXeCuRL4ADAIWAD8NJerkXOjifKWXrN6YcT4iKiNiNrevXs3  
EbZZAYYPH/HjoX//NENz//7p++jRMHkyzJkDp58O++yTzv/LX+CuuzzE2TqMdk0wEffqRKyMiFXAr0h9JJBqGf  
1KTu0LvJLL+zZSvto1kroCPUIncmu7l1nlGT48vfm/alX6LB09tt12MHYsblFF+j52LBx8MOy6a3qJc9myli2+5d2  
TTC5T6XeEUD9CLMpwLA8Mmw7Umf+tlhYACyRtFfuXzkBuL3kmvoRYkcB9+Z+mruBgyTlyk1wb+Uys+p28  
80wYUIa2nzzyan57Oqri47KOrGyvQcj6UbGM8AWkuaTRnZ9RtlgUpPVi8ApABExS9Jk4GlgBxBaRNSvPXsqau  
Rad+DOvAFcA9wgaQ6p5jIs32uRpAuB6fm8CyKiuYMNzCrX+uvDCSfA8cfDPfekPpr6JZyXLoVXXoEddig2Rut  
UFG6vBaC2tjbq6uqKDsOsUWk/psrroCvfS1NsPmtb6WZAtRYd6XZupE0IyJqGzvmqWLMqlI9EjniilQezX33pU  
EBH/94aLlZ5hWRk4wZp1Bnz5p2PO8eXD55bBwIVx6aUMCWrgi2PisKjnBmHUmG24Ip52WpqK55ZZUtnBh  
GgJ9zjimpn8asjTjBmHVGXbra11lun/bffTk1mF1+cRp6deCI88USR0VmVcIlx6+wGDEi1meeeg698Je0PGgTz57/fl  
WZNcoIxs2T77dNSAfPmwaRJ0De/43zWWWm9Gr+4aevICcbMVterFxx9dNpftgz++Ec46aRU0/nhD2GRXyuz5n  
GCMbO169YNHn00JZlidd01r0fTrB//zP0VHZhXACcbMmibBgQemJZwffxyOOQZq83t1Dz8MDz1UbHzWYTnB  
mFnz7borXHstbLVV+n7hhWIWge98Am69tWFqGjOcYMysNSZNgp//HF59FY46Cj74Qfjtb4uOyjoIjXgza7mNNk



```
]
},
"metadata": {
  "needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
  "plt.plot(k,TWSS,'ro--')\n",
  "plt.xlabel('no of cluster')\n",
  "plt.ylabel('TWSS')"
]
},
{
  "cell_type": "code",
  "execution_count": 31,
  "id": "134037f3",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "KMeans(n_clusters=4)"
        ]
      },
      "execution_count": 31,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "#selecting 4 clusters\n",
    "\n",
    "model=KMeans(n_clusters=4)\n",
    "\n",
    "model.fit(data)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 32,
  "id": "ec8455f6",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,\n",
          "       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,\n",
          "       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,\n",
          "       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
          "       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
          "       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 0, 3, 2, 3, 2, 3,\n",
          "       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,\n"
```



```

"    2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,\n",
"    2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,\n",
"    2, 3])"
]
},
"execution_count": 32,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"model.labels_"
]
},
{
"cell_type": "code",
"execution_count": 33,
"id": "053dd73a",
"metadata": {},
"outputs": [],
"source": [
"mb=pd.Series(model.labels_)"
]
},
{
"cell_type": "code",
"execution_count": 34,
"id": "ee476632",
"metadata": {},
"outputs": [
{
"data": {
"text/html": [
"<div>\n",
"<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border='1' class='dataframe'>\n",
"  <thead>\n",
"    <tr style='text-align: right;'>\n",
"      <th></th>\n",
"      <th>CustomerID</th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Annual Income (k$)</th>\n",

```

```

"    <th>Spending Score (1-100)</th>\n",
"  </tr>\n",
" </thead>\n",
" <tbody>\n",
"   <tr>\n",
"     <th>0</th>\n",
"     <td>1</td>\n",
"     <td>1</td>\n",
"     <td>19</td>\n",
"     <td>15</td>\n",
"     <td>39</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>1</th>\n",
"     <td>2</td>\n",
"     <td>1</td>\n",
"     <td>21</td>\n",
"     <td>15</td>\n",
"     <td>81</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>2</th>\n",
"     <td>3</td>\n",
"     <td>0</td>\n",
"     <td>20</td>\n",
"     <td>16</td>\n",
"     <td>6</td>\n",
"   </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
" CustomerID Gender Age Annual Income (k$) Spending Score (1-100)\n",
"0          1    1  19          15          39\n",
"1          2    1  21          15          81\n",
"2          3    0  20          16           6"
]
},
"execution_count": 34,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"data.head(3)"
]
},
{
"cell_type": "markdown",
"id": "49ccef30",
"metadata": {},
"source": [
"# 10. Add the cluster data with the primary dataset"
]
}

```

```

},
{
  "cell_type": "code",
  "execution_count": 35,
  "id": "80715661",
  "metadata": {},
  "outputs": [],
  "source": [
    "data['clust']=mb"
  ]
},
{
  "cell_type": "code",
  "execution_count": 36,
  "id": "0bc5b4b9",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",
          "    vertical-align: middle;\n",
          "  }\n",
          "\n",
          "  .dataframe tbody tr th {\n",
          "    vertical-align: top;\n",
          "  }\n",
          "\n",
          "  .dataframe thead th {\n",
          "    text-align: right;\n",
          "  }\n",
          "</style>\n",
          "<table border='1' class='dataframe'>\n",
          "  <thead>\n",
          "    <tr style='text-align: right;'>\n",
          "      <th></th>\n",
          "      <th>CustomerID</th>\n",
          "      <th>Gender</th>\n",
          "      <th>Age</th>\n",
          "      <th>Annual Income (k$)</th>\n",
          "      <th>Spending Score (1-100)</th>\n",
          "      <th>clust</th>\n",
          "    </tr>\n",
          "  </thead>\n",
          "  <tbody>\n",
          "    <tr>\n",
          "      <th>0</th>\n",
          "      <td>1</td>\n",
          "      <td>1</td>\n",
          "      <td>19</td>\n",
          "      <td>15</td>\n",
          "      <td>39</td>\n",
          "      <td>1</td>\n",

```

```

" </tr>\n",
" <tr>\n",
" <th>1</th>\n",
" <td>2</td>\n",
" <td>1</td>\n",
" <td>21</td>\n",
" <td>15</td>\n",
" <td>81</td>\n",
" <td>1</td>\n",
" </tr>\n",
" <tr>\n",
" <th>2</th>\n",
" <td>3</td>\n",
" <td>0</td>\n",
" <td>20</td>\n",
" <td>16</td>\n",
" <td>6</td>\n",
" <td>1</td>\n",
" </tr>\n",
" <tr>\n",
" <th>3</th>\n",
" <td>4</td>\n",
" <td>0</td>\n",
" <td>23</td>\n",
" <td>16</td>\n",
" <td>77</td>\n",
" <td>1</td>\n",
" </tr>\n",
" <tr>\n",
" <th>4</th>\n",
" <td>5</td>\n",
" <td>0</td>\n",
" <td>31</td>\n",
" <td>17</td>\n",
" <td>40</td>\n",
" <td>1</td>\n",
" </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
" CustomerID Gender Age Annual Income (k$) Spending Score (1-100) clust\n",
"0      1      1  19      15      39      1\n",
"1      2      1  21      15      81      1\n",
"2      3      0  20      16       6      1\n",
"3      4      0  23      16      77      1\n",
"4      5      0  31      17      40      1"
]
},
"execution_count": 36,
"metadata": {},
"output_type": "execute_result"
}
],

```

```

"source": [
  "data.head()"
],
{
  "cell_type": "code",
  "execution_count": 37,
  "id": "c919ea1d",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",
          "    vertical-align: middle;\n",
          "  }\n",
          "\n",
          "  .dataframe tbody tr th {\n",
          "    vertical-align: top;\n",
          "  }\n",
          "\n",
          "  .dataframe thead th {\n",
          "    text-align: right;\n",
          "  }\n",
          "</style>\n",
          "<table border='1' class='dataframe'>\n",
          "  <thead>\n",
          "    <tr style='text-align: right;'>\n",
          "      <th></th>\n",
          "      <th>CustomerID</th>\n",
          "      <th>Gender</th>\n",
          "      <th>Age</th>\n",
          "      <th>Annual Income (k$)</th>\n",
          "      <th>Spending Score (1-100)</th>\n",
          "      <th>clust</th>\n",
          "    </tr>\n",
          "  </thead>\n",
          "  <tbody>\n",
          "    <tr>\n",
          "      <th>195</th>\n",
          "      <td>196</td>\n",
          "      <td>0</td>\n",
          "      <td>35</td>\n",
          "      <td>120</td>\n",
          "      <td>79</td>\n",
          "      <td>3</td>\n",
          "    </tr>\n",
          "    <tr>\n",
          "      <th>196</th>\n",
          "      <td>197</td>\n",
          "      <td>0</td>\n",
          "      <td>45</td>\n",
          "      <td>126</td>\n",

```

```

"    <td>28</td>\n",
"    <td>2</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>197</th>\n",
"    <td>198</td>\n",
"    <td>1</td>\n",
"    <td>32</td>\n",
"    <td>126</td>\n",
"    <td>74</td>\n",
"    <td>3</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>198</th>\n",
"    <td>199</td>\n",
"    <td>1</td>\n",
"    <td>32</td>\n",
"    <td>60</td>\n",
"    <td>18</td>\n",
"    <td>2</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>199</th>\n",
"    <td>200</td>\n",
"    <td>1</td>\n",
"    <td>30</td>\n",
"    <td>60</td>\n",
"    <td>83</td>\n",
"    <td>3</td>\n",
"  </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"  CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)  \n",
"195      196      0  35           120           79  \n",
"196      197      0  45           126           28  \n",
"197      198      1  32           126           74  \n",
"198      199      1  32            60           18  \n",
"199      200      1  30            60           83  \n",
"\n",
"  clust  \n",
"195    3  \n",
"196    2  \n",
"197    3  \n",
"198    2  \n",
"199    3  "
]
},
"execution_count": 37,
"metadata": {},
"output_type": "execute_result"
}
],

```

```

"source": [
  "data.tail()"
],
{
  "cell_type": "markdown",
  "id": "bcf97bab",
  "metadata": {},
  "source": [
    "# 11. Split the data into dependent and independent variables"
  ],
},
{
  "cell_type": "code",
  "execution_count": 38,
  "id": "5ffaa6fa",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "0    1\n",
          "1    1\n",
          "2    1\n",
          "3    1\n",
          "4    1\n",
          "   ..\n",
          "195   3\n",
          "196   2\n",
          "197   3\n",
          "198   2\n",
          "199   3\n",
          "Name: clust, Length: 200, dtype: int32"
        ]
      },
      "execution_count": 38,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "#dependent\n",
    "y= data['clust']\n",
    "y"
  ],
},
{
  "cell_type": "code",
  "execution_count": 39,
  "id": "3b4a49f2",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/html": [

```

```

"<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>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Annual Income (k$)</th>\n",
"      <th>Spending Score (1-100)</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1</td>\n",
"      <td>19</td>\n",
"      <td>15</td>\n",
"      <td>39</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>1</td>\n",
"      <td>21</td>\n",
"      <td>15</td>\n",
"      <td>81</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>0</td>\n",
"      <td>20</td>\n",
"      <td>16</td>\n",
"      <td>6</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>0</td>\n",
"      <td>23</td>\n",
"      <td>16</td>\n",
"      <td>77</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",

```



```

"    <td>0</td>\n",
"    <td>31</td>\n",
"    <td>17</td>\n",
"    <td>40</td>\n",
"  </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"  Gender Age Annual Income (k$) Spending Score (1-100)\n",
"0    1  19         15         39\n",
"1    1  21         15         81\n",
"2    0  20         16          6\n",
"3    0  23         16        77\n",
"4    0  31         17        40"
]
},
"execution_count": 39,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"#independent\n",
"x= data.drop(columns=['CustomerID','clust'],axis=1)\n",
"x.head()"
]
},
{
"cell_type": "code",
"execution_count": 52,
"id": "8925a5cb",
"metadata": {},
"outputs": [
{
"data": {
"text/html": [
"<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>Gender</th>\n",
"    <th>Age</th>\n",
"    <th>Annual Income (k$)</th>\n",
"    <th>Spending Score (1-100)</th>\n",
"  </tr>\n",
" </thead>\n",
" <tbody>\n",
"   <tr>\n",
"     <th>195</th>\n",
"     <td>0</td>\n",
"     <td>35</td>\n",
"     <td>120</td>\n",
"     <td>79</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>196</th>\n",
"     <td>0</td>\n",
"     <td>45</td>\n",
"     <td>126</td>\n",
"     <td>28</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>197</th>\n",
"     <td>1</td>\n",
"     <td>32</td>\n",
"     <td>126</td>\n",
"     <td>74</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>198</th>\n",
"     <td>1</td>\n",
"     <td>32</td>\n",
"     <td>60</td>\n",
"     <td>18</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>199</th>\n",
"     <td>1</td>\n",
"     <td>30</td>\n",
"     <td>60</td>\n",
"     <td>83</td>\n",
"   </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"  Gender  Age  Annual Income (k$)  Spending Score (1-100)\n",
"195      0   35              120          79\n",
"196      0   45              126          28\n",
"197      1   32              126          74\n",
"198      1   32               60          18\n",
"199      1   30               60          83"
]

```

```

    },
    "execution_count": 52,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "x.tail()"
]
},
{
  "cell_type": "markdown",
  "id": "55027329",
  "metadata": {},
  "source": [
    "# 12. Split the data into training and testing"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "8c768efa",
  "metadata": {},
  "outputs": [],
  "source": [
    "from sklearn.model_selection import train_test_split"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "65a64d15",
  "metadata": {},
  "outputs": [],
  "source": [
    "x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)"
  ]
},
{
  "cell_type": "markdown",
  "id": "1f083fee",
  "metadata": {},
  "source": [
    "# 13. Build the Model"
  ]
},
{
  "cell_type": "code",
  "execution_count": 41,
  "id": "2e2f691a",
  "metadata": {},
  "outputs": [],
  "source": [
    "from sklearn.ensemble import RandomForestClassifier"
  ]
}

```

```
},
{
  "cell_type": "code",
  "execution_count": 42,
  "id": "97db8fd0",
  "metadata": {},
  "outputs": [],
  "source": [
    "rf=RandomForestClassifier()"
  ]
},
{
  "cell_type": "markdown",
  "id": "b939506d",
  "metadata": {},
  "source": [
    "# 14. Train the Model"
  ]
},
{
  "cell_type": "code",
  "execution_count": 117,
  "id": "f4011fa5",
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "RandomForestClassifier()"
        ]
      },
      "execution_count": 117,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "rf.fit(x_train,y_train)"
  ]
},
{
  "cell_type": "markdown",
  "id": "191ebf9e",
  "metadata": {},
  "source": [
    "# 15. Test the Model"
  ]
},
{
  "cell_type": "code",
  "execution_count": 118,
  "id": "c56c5f8b",
  "metadata": {},
  "outputs": [],
  "source": [
```

```

"#prediction\n",
"pred=rf.predict(x_test)"
],
},
{
"cell_type": "markdown",
"id": "0faa2234",
"metadata": {},
"source": [
"# 16. Measure the performance using Evaluation Metrics"
]
},
{
"cell_type": "code",
"execution_count": 119,
"id": "614c42c3",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"0.975"
]
}
},
"execution_count": 119,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"# Accuracy of DI model\n",
"from sklearn.metrics import accuracy_score\n",
"accuracy_score(y_test,pred)"
]
},
{
"cell_type": "code",
"execution_count": 120,
"id": "a569b1b6",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"array([[13, 0, 0, 0],\n",
"       [ 0, 10, 0, 0],\n",
"       [ 1, 0, 8, 0],\n",
"       [ 0, 0, 0, 8]], dtype=int64)"
]
}
},
"execution_count": 120,
"metadata": {},
"output_type": "execute_result"
}
],

```

```
"source": [  
    "#confusion matrix\n",  
    "from sklearn import metrics\n",  
    "metrics.confusion_matrix(y_test,pred)"  
]  
}  
],  
"metadata": {  
    "kernel_spec": {  
        "display_name": "Python 3 (ipykernel)",  
        "language": "python",  
        "name": "python3"  
    },  
    "language_info": {  
        "codemirror_mode": {  
            "name": "ipython",  
            "version": 3  
        },  
        "file_extension": ".py",  
        "mimetype": "text/x-python",  
        "name": "python",  
        "nbconvert_exporter": "python",  
        "pygments_lexer": "ipython3",  
        "version": "3.9.12"  
    },  
    "nbformat": 4,  
    "nbformat_minor": 5  
}
```