```
{
"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 \land "Mall\_Customers.csv \land")"
 \bar{}
  "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/YYfK9AAACXBIWXMAAAsTAAALEwEAmpwYAAAOQ0lEQVR4nO3dYYhl9XnH8e8vampiLK51lCG6nSaITZA6psPWYgmJJmGNIeoLIULTpUjXFwoKlrL1TUyhYCEm7YsirNFmaU2KVYOiNs2yVawlmM7aja5dxZBuE5Pt7qRpqrZgUZ++uGebYZzx3pl770z+d74fuJxz/vfcOc/D7P4487/n3JuqQpLUnndsdAGSpLUxwCWpUQa4JDXKAJekRhngktSoE9fzYGeccUbNzMys5yElqXn79+//cVVNLR3vG+BJTgaeAH6h2/++qvpckluB3wMWul1vqapH3+5nzczMMD8/v9raJWlTS/Jvy40Pcgb+GnBJVb2a5CTgySR/2z33par6wqiKlCQNrm+AV+9On1e7zZO6h3f/SNIGG+hNzCQnJDkAHAP2VtVT3VM3JHkmyd1JtoyrSEnSWw0U4FX1RlXNAmcD25KcD9wBvB+YBY4Aty/32iQ7k8wnmV9YWFhuF0nSGqzqMsKq+inwOLC9qo52wf4mcCewbYXX7K6quaqam5p6y5uokqQ16hvgSaaSnNatvwv4GPB8kulFu10FHBxLhZKkZQ1yFco0sCfJCfQC/96qejjJXyaZpfeG5mHgurFVKUl6i0GuQnkGuHCZ8c+OpSJJ0kC8lV6SGrWut9JrdWZ2Pblhxz182+UbclxJq+MZuCQ1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNcoAl6RGG

eCS1CgDXJIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An OTnJt5N8J8lzST7fjZ+eZG+SF7vllvGXK0k6bpAz8NeAS6rqAmAW2J7kImAXsK+qzgX2dduSpHXSN8Cr59Vu86 TuUcAVwJ5ufA9w5TgKlCQtb6A58CQnJDkAHAP2VtVTwFlVdQSgW565wmt3JplPMr+wsDCisiVJAwV4Vb1R VbPA2cC2JOcPeoCq2l1Vc1U1NzU1tcYyJUlLreoqlKr6KfA4sB04mmQaoFseG3VxkqSVDXIVylSS07r1dwEfA54 HHgJ2dLvtAB4cU42SpGUMcifmNLAnyQn0Av/eqno4ybeAe5NcC3wfuHqMdUqSlugb4FX1DHDhMuP/AVw6jqI kSf0181koWj9+7ozUBm+ll6RGGeCS1CgDXJIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA 1ySGmWAS1KjDHBJapQBLkmNMsAlqVEGuCQ1ygCXpEb5jTwSfguR2uQZuCQ1ygCXpEb1DfAk5yR5LMmh JM8lubEbvzXJD5Mc6B6fHH+5kqTjBpkDfx24uaqeTnIqsD/J3u65L1XVF8ZXniRpJX0DvKqOAEe69VeSHALeO +7CJElvb1Vz4ElmgAuBp7qhG5I8k+TuJFtGXZwkaWUDB3iS9wD3AzdV1cvAHcD7gVl6Z+i3r/C6nUnmk8wvL CwMX7EkCRgwwJOcRC+876mqBwCq6mhVvVFVbwJ3AtuWe21V7a6quaqam5qaGlXdkrTpDXIVSoC7gENV9c VF49OLdrsKODj68iRJKxnkKpSLgc8CzyY50I3dAlyTZBYo4DBw3RjqkyStYJCrUJ4EssxTj46+HEnSoLwTU5IaZ YBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA1ySGmWAS1KjDHBJapQBLkmNMsAlqVEGuC Q1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNapvgCc5J8ljSQ4leS7Jjd346Un2JnmxW24Zf 7mSpOMGOQN/Hbi5qj4AXARcn+SDwC5gX1WdC+zrtiVJ66RvgFfVkap6ult/BTgEvBe4AtjT7bYHuHJMNUqSlr GqOfAkM8CFwFPAWVV1BHohD5y5wmt2JplPMr+wsDBkuZKk4wYO8CTvAe4Hbqqqlwd9XVXtrqq5qpqbmpp aS42SpGUMFOBJTqIX3vdU1QPd8NEk093z08Cx8ZQoSVrOIFehBLgLOFRVX1z01EPAjm59B/Dg6MuTJK3kx AH2uRj4LPBskgPd2C3AbcC9Sa4Fvg9cPZYKJUnL6hvgVfUkkBWevnS05UiSBjXIGfimN7PrkY0uQZLewlvpJal RBrgkNcoAl6RGGeCS1CgDXJIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA1ySGmWAS1 KjDHBJapQBLkmNMsAlqVF+I49+rvjtR9LgPAOXpEYZ4JLUqL4BnuTuJMeSHFw0dmuSHyY50D0+Od4yJUlL DXIG/hVg+zLjX6qq2e7x6GjLkiT10zfAq+oJ4CfrUIskaRWGuQrlhiS/A8wDN1fVfy63U5KdwE6ArVu3DnE4aTJt 1JU3h2+7fEOOq9FZ65uYdwDvB2aBI8DtK+1YVburaq6q5qamptZ4OEnSUmsK8Ko6WlVvVNWbwJ3AttGWJUn qZ00BnmR60eZVwMGV9pUkjUffOfAkXwM+ApyR5CXgc8BHkswCBRwGrhtfiZKk5fQN8Kq6Zpnhu8ZQiyRpF bwTU5Ia5YdZSdo0NvLD0sZx2aZn4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNcoAl6RGGeCS1CgDX JIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA1ySGmWAS1KjDHBJalTfAE9yd5JjSQ4uGjs9 yd4kL3bLLeMtU5K01CBn4F8Bti8Z2wXsq6pzgX3dtiRpHfUN8Kp6AvjJkuErgD3d+h7gytGWJUnq58Q1vu6sqjoC UFVHkpy50o5JdgI7AbZu3brGw0katZldj2zYsQ/fdvmGHXuSjP1NzKraXVVzVTU3NTU17sNJ0qax1gA/mmQaoF seG11JkqRBrDXAHwJ2dOs7gAdHU44kaVCDXEb4NeBbwHlJXkpyLXAb8PEkLwIf77YlSeuo75uYVXXNCk9d OuJaJEmr4J2YktQoA1ySGmWAS1KjDHBJapQBLkmNMsAlqVEGuCQ1ygCXpEYZ4JLUKANckhplgEtSowxwS WqUAS5JjTLAJalRBrgkNcoAl6RGGeCS1CgDXJIa1fcr1SRp1GZ2PbLRJUwEz8AlqVEGuCQ1aqgplCSHgVeAN 4DXq2puFEVJkvobxRz4R6vqxyP4OZKkVXAKRZIaNWyAF/DNJPuT7FxuhyQ7k8wnmV9YWBjycJKk44YN8Iu r6kPAZcD1ST68dIeq2l1Vc1U1NzU1NeThJEnHDRXgVfWjbnkM+DqwbRRFSZL6W3OAJzklyanH14FPAAdHV Zgk6e0NcxXKWcDXkxz/OV+tqm+MpCpJUl9rDvCq+h5wwQhrkSStgpcRSlKjDHBJapQBLkmNMsAlqVEGuCQ 1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNcoAl6RGGeCS1CgDXJIaZYBLUqMMcElql AEuSY0ywCWpUQa4JDXKAJekRg0V4Em2J3khyXeT7BpVUZKk/tYc4ElOAP4cuAz4IHBNkg+OqjBJ0tsb5gx8 G/DdqvpeVf0v8NfAFaMpS5LUz4lDvPa9wA8Wbb8E/MbSnZLsBHZ2m68meWGIY67GGcCP1+lYG2mz9Ambp 9fN0idsol7zJ0P1+svLDQ4T4FlmrN4yULUb2D3EcdYkyXxVza33cdfbZukTNk+vm6VPsNdhDTOF8hJwzqLts4Ef DVeOJGlQwwT4PwHnJvmVJO8EPgM8NJqyJEn9rHkKpapeT3ID8HfACcDdVfXcyCob3rpP22yQzdInbJ5eN0ufY K9DSdVbpq0lSQ3wTkxJapQBLkmNaj7Ak5yT5LEkh5I8l+TGbvz0JHuTvNgtt2x0rcNKcnKSbyf5Ttfr57vxiesVenf 7JvnnJA9325Pa5+EkzyY5kGS+G5u4XpOcluS+JM93/19/c0L7PK/7XR5/vJzkpnH02nyAA68DN1fVB4CLgOu7W/ p3Afuq6lxgX7fduteAS6rqAmAW2J7kIiazV4AbgUOLtie1T4CPVtXsouuEJ7HXPwO+UVW/ClxA73c7cX1W1Qvd 73IW+HXgf4CvM45eq2qiHsCDwMeBF4DpbmwaeGGjaxtxn+8GnqZ39+vE9UrvvoJ9wCXAw93YxPXZ9XIYOG PJ2ET1Cvwi8K90F05Map/L9P0J4B/H1esknIH/vyQzwIXAU8BZVXUEoFueuYGljUw3rXAAOAbsrapJ7fVPgT8 A3lw0Nol9Qu8O5m8m2d999ARMXq/vAxaAv+imxb6c5BQmr8+lPgN8rVsfea8TE+BJ3gPcD9xUVS9vdD3jUlVv VO9Ps7OBbUnO3+CSRi7Jp4BjVbV/o2tZJxdX1YfofbLn9Uk+vNEFjcGJwIeAO6rqQuC/mYDpkrfT3eD4aeBvxn WMiQjwJCfRC+97quqBbvhokunu+Wl6Z6wTo6p+CjwObGfyer0Y+HSSw/Q+5fKSJH/F5PUJQFX9qFseozdXuo3 J6/Ul4KXuL0aA++gF+qT1udhlwNNVdbTbHnmvzQd4kgB3AYeq6ouLnnoI2NGt76A3N960JFNJTuvW3wV8DHi eCeu1qv6wqs6uqhl6f4L+fVX9NhPWJ0CSU5Kcenyd3pzpQSas16r6d+AHSc7rhi4F/oUJ63OJa/jZ9AmModfm78R M8lvAPwDP8rP50lvozYPfC2wFvg9cXVU/2ZAiRyTJrwF76H10wTuAe6vqj5L8EhPW63FJPgL8flV9ahL7TPI+e mfd0Jtm+GpV/fGE9joLfBl4J/A94Hfp/h0zQX0CJHk3vY/bfl9V/Vc3NvLfafMBLkmbVfNTKJK0WRngktQoA1yS GmWAS1KjDHBJapQBLkmNMsAlqVH/ByG7xWFHCR5SAAAAAElFTkSuQmCC",

```
"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": "iVBORw0KGgoAAAANSUhEUgAAAXoAAAD4CAYAAADiry33AAAAAOXRFWHRTb2Z0d 2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACX BIWXMAAAsTAAALEwEAmpwYAAAVT0lEQVR4nO3dfYxU133G8efxsLEX5wVcFgvjdXArhGObNCQbY8d S5MihWElqNmmiguKKNJZQJSsvTUNiCoqjBsduiZJWldIWJW5QSddyEmeNUicE4bhWX1hnCbbXGFPcugEWA psXktTe8LL8+sdeYFjPsDN7Z2bnzH4/Epq5v3vv3CP+ePbOmXPPcUQIANC6LprsBgAA6ougB4AWR9ADQIsj6 AGgxRH0ANDipk12AyRp1qxZMW/evMluBgAkZefOnT+NiI7xjmuKoJ83b576+/snuxkAkBTbP67kuHG7bmw/YakBTbP67kuHg7bmw/YakBTbP67kuHg7bmw/YakBTbP67kuHg7bmwPuo7WdL7Puk7bA9q6i2xvYLtvfaXlpdswEAtVZJH/3XJN02tmi7U9ISSfuLatdKWi7puuycL9su1KSlAIAJGTfoI+I JST8vsetLkj4lqfjR2mWSHoyI4xHxoqQXJN1Qi4YCACZmQqNubN8uaTAinh6za66kA0XbB7Naqc9YZbvfdv/Q0 NBEmgEAqEDVQW97uqS1kj5TaneJWsnJdCJiY0R0RURXR8e4PxoDACZoIqNufkfS1ZKeti1JV0r6ke0bNHoH31 9WdMWZ9b1Dmjzjv0ayWbfHInQ5h37ta53oB6XA4BkVTK8skfSf0paYPug7TvLHRsRuyU9JOk5Sd+TdFdEjNSq scV6+g5UVQeAqWrcrpuIWDHO/nljtu+VdG++Zo1vpMw8+uXqADBVJTvXTcGlfvctXweAqSrZoF+xuLOqOgB MVU0x181EnBldw6gbALgwN8OasV1dXcGkZgBQHds7I6JrvOOS7boBAFSGoAeAFkfQA0CLI+gBoMUR9AD Q4gh6AGhxBD0AtDiCHgBaHEEPAC2OoAeAFpfsXDeS1LtrUBu27tWhY8O6Yka7Vi9doO5FJZeoBYCm0sj8Sjb oe3cNas3DAxo+ObquyeCxYa15eHR1KcIeQDNrdH4l23WzYeves/9JZwyfHNGGrXsnqUUAUJlG51eyQX/o2HBV dQBoFo3Or2SD/ooZ7VXVAaBZNDq/kg361UsXqL2tcF6tva2g1UsXTFKLAKAyjc6vZH+MPfODBaNuAKSm0fn FClMAkChWmAIASCLoAaDljdtHb/sBSe+RdDQirs9qGyT9vqQTkv5b0h9HxLFs3xpJd0oakfTRiNhan6YDQLrW 9Q6op++ARiJUsLVicafWdy+sy7UquaP/mqTbxtS2Sbo+It4o6b8krZEk29dKWi7puuycL9suCABw1rreAW3esV8j2 W+kIxHavGO/1vUO1OV64wZ9RDwh6edjat+PiFPZ5g5JV2bvl0l6MCKOR8SLkl6QdEMN2wsAyevpO1BVPa9a9 NF/WNJ3s/dzJRW39GBWewXbq2z32+4fGhqqQTMAIA0jZUY7lqvnlSvoba+VdErS18+UShxWsuURsTEiuiKiq6

OjI08zACApBZeKyvL1vCYc9LZXavRH2g/GucH4ByV1Fh12paRDE28eALSeFYs7q6rnNaGgt32bpE9Luj0iXi7at UXSctsX275a0nxJT+ZvJgC0jvXdC3XHjVedvYMv2LrjxqvqNupm3CdjbfdIukXSLElHJN2j0VE2F0v6WXbYjoj4k +z4tRrttz8l6eMR8d2xnzkWT8YCQPUqfTKWKRAAIFFMgQAAkETQA0DLS3aaYonFwQGki8XBK8Di4ABSxe LgFWJxcACpYnHwCrE4OIBUsTh4hVgcHECqWBy8QiwODiBVLA5eIRYHB5AqFgcHAFSEJ2MBAJIIegBoeQ Q9ALQ4gh4AWhxBDwAtjqAHgBZH0ANAiyPoAaDFJftkLACkbMkXH9e+oy+d3Z4/+1Jt+8QtdbkWd/QA0GBj Q16S9h19SUu++HhdrkfQA0CDjQ358ep5EfQA0OKS7qNnzVgAGN+4d/S2H7B91PazRbXLbG+zvS97nVm0b43t F2zvtb20Xg0/s+bi4LFhhc6tudi7a7BelwSAmrj8Na+qqp5XJV03X5N025ja3ZK2R8R8Sduzbdm+VtJySddl53zZdkF 1wJqxAFI1rVA6FsvV8xo36CPiCUk/H1NeJmlT9n6TpO6i+oMRcTwiXpT0gqQbatPU87FmLIBUpbJm7OURcViS stfZWX2upANFxx3Maq9ge5Xtftv9Q0NDVTdgxvS2quoA0CwanV+1HnXjErWSS1hFxMaI6IqIro6OjqovdHxMt8 14dQBoFo3Or4kG/RHbcyQpez2a1Q9K6iw67kpJhybevPJePnm6qjoANItG59dEg36LpJXZ+5WSHimqL7d9se2rJc2 X9GS+JgIA8hh3HL3tHkm3SJpl+6CkeyTdL+kh23dK2i/pA5IUEbttPyTpOUmnJN0VEXX5LjKjvU3Hhk+WrAMA zhk36CNiRZldt5Y5/l5J9+ZpVCU+e/t1Wv2Np3Xy9LmfANousj57+3X1vjQAJCXZJ2PPPAHLk7EAUlOwNRKvH KdScKnxLPklG/TSaNgT7ABSs2Jxpzbv2F+yXg9JBz0ApGh990JJUk/fAY1EqGBrxeLOs/Vac5T4+tBoXV1d0d/fX /V563oHGvYfBQDNxvbOiOga77hk7+jX9Q6c99VnJOLsNmEPAOckOx99T9+BquoAMFUlG/SlfrG+UB0Apqpkg 77cMKR6DU8CgFQlG/TlhiHVa3gSAKQq2R9jGz08CQBSlfTwSgCYylp+eKXEOHoA6WpkfiUb9IyjB5CqRudXs j/Glpon4kJ1AGgW/9xXOqfK1fNKNugBIFWny/w0Wq6eF0EPAC2OoAeAFkfQA0CLI+gBoMUlG/TlFgFncXAAz a7R+ZVs0H/29uvUdtH5E5ixODiAFDQ6v5J9YIrFwQGkqtH5lewdPQCgMrmC3vaf2t5t+1nbPbYvsX2Z7W2292W vM2vV2GK9uwa15uEBDR4bVkgaPDasNQ8PqHfXYD0uBwA10+j8mnDQ254r6aOSuiLiekkFScsl3S1pe0TMl7Q 92665DVv3avjkyHm14ZMj2rB1bz0uBwA10+j8ytt1M01Su+1pkqZLOiRpmaRN2f5NkrpzXqOkQ8eGq6oDQLNod H5NOOgjYlDSFyTtl3RY0i8j4vuSLo+Iw9kxhyXNrkVDx7piRntVdQBoFtNfVaiqnleerpuZGr17v1rSFZIutX1HFee vst1vu39oaKjq669euqCqOgA0i5dPjFRVzytP1807Jb0YEUMRcVLSw5LeJumI7TmSlL0eLXVyRGyMiK6I6Oro6K j64vc9+lxVdQBoFuUmqazXen95gn6/pBttT7dtSbdK2iNpi6SV2TErJT2Sr4mlHfn1iarqADBVTfiBqYjos/1NST+Sd ErSLkkbJb1a0kO279ToH4MP1KKhAICJyfVkbETcI+meMeXjGr27BwA0AZ6MBYAWl2zQz599aVV1AJiqkg36n /5f6R9dy9UBYKpKNuh/8fLJquoAMFUIG/QAgMoQ9ADQ4gh6AGhxBD0AtDiCHgBaHEEPAA3mKut5EfQA0G ApzV4JAEhAskHf6K8+AJCqZIOepQQBpKq9rXT0lqvnlWzQv+Oa0qtSlasDQLO4cmbpG9Jy9bySDfofPF96ndlyd QBoFvuOvlRVPa9kg/7QseGq6gAwVSUb9PTRA0Blkg361UsXqK1w/hibtoK1eumCSWoRADSnZINekk6OxAW3 AQAJB/0nv/F0VXUAmKqSDfpTp0vfvZerA8BU1WzQAwAqQ9ADQIvLFfS2Z9j+pu3nbe+xfZPty2xvs70ve51Zq8 YCAKqX947+byR9LyKukfS7kvZIulvS9oiYL217tg0AmCQTDnrbr5X0dklflaSIOBERxyQtk7QpO2yTpO58TQQA 5JHnjv63JQ1J+kfbu2x/xfalki6PiMOSlL3OLnWy7VW2+233Dw0xPw0A1EueoJ8m6c2S/i4iFkl6SVV000TExojoioi ujg5mnASAeskT9AclHYyIvmz7mxoN/iO250hS9no0XxMBAHIMOOgj4ieSDtg+M7nMrZKek7RF0sqstlLSI7laW MbcMpOXlasDwFSVd9TNRyR93fYzkt4k6fOS7pe0xPY+SUuy7ZpbvXSB2tsK59Xa2wpMagYAY0zLc3JEPCWp q8SuW/N8biW6F82VJG3YuleHjg3rihntWr10wdk6AGBUrqCfbN2L5hLsAJJTsDUSr5yXq2CXODq/pIP+mrWP6j dFUxNfUrCev/ddk9giABjfisWd2rxjf8l6PSQ7183YkJek34yErln76CS1CAAqs233T6qq55Vs0I8N+fHqANAsjvz6R FX1vJINegBAZQh6AGhxyQb9JYXSv06XqwNAs7j8Na+qqp5XskH//L3vekWoM+oGQAr61i4pmV99a5fU5XpJD 68k1AGkaF3vQMlRg+t6B7S+e2HNr5fsHb0k9e4a1M33P6ar7/4X3Xz/Y+rdNTjZTQKAcfX0Haiqnleyd/S9uwb1iY ee0unsj+LgsWF94qGnJImnZQE0tVJPxV6onleyd/Sf/tYzZ0P+jNMxWgcAnJNs0B8/dbqqOgBMVckGPQCgMgQ9 ALS4ZIP+tRcXqqoDwFSVbND/6vhIVXUAmKqSDXoAQGUIegBosJnT26qq55Vs0LeVaXm5OgA0i3LPRdXpeal 0g/5kmeHy5eoA0CyODZ+sqp5XskEPAKgMQQ8ALS530Nsu2N51+zvZ9mW2t9nel73OzN9MAMBE1eKO/mOS9 hRt3y1pe0TMl7Q926659jK/uparA8BUlSsVbV8p6d2SvlJUXiZpU/Z+k6TuPNco5773vbGqOgBMVXlvf/9a0qckFY 91uTwiDktS9jq71Im2V9nut90/NDQ0oYu3XeQLbgMAcgS97fdIOhoROydyfkRsjIiuiOjq6Oio+vwNW/fq5JgJ6U+e Dm3YuncizQGAlpVnhambJd1u+12SLpH0WtubJR2xPSciDtueI+loLRo61uCx4arqADBVTfiOPiLWRMSVETFP0 nJJj0XEHZK2SFqZHbZS0iO5W1lCwaW7acrVAWCqqscQlfslLbG9T9KSbLvmGr3mIgCkqiaLg0fE45Iez97/TNKt tfjcC5k7o71kN83cGe31vjQAJCXZQeerly5Qe9v5i4y0txW0eumCSWoRAFSmXAdzvTqekw367kVzdd/7FmrujHZ Zo3fy971voboXzZ3spgHABX3wxquqqudVk66bydK9aC7BDgDjSDro1/UOqKfvgEYiVLC1YnGn1ncvnOxmAcA F9fQdKFuvR4YlG/Trege0ecf+s9sjEWe3CXsAzazRowaT7aO/0F9EAMA5yQY94+gBoDLJBj1PxgJIVaPzK9mgX 7G4s6o6ADSLRudXskHf9frLXtH4i7I6ADSzcjlVr/xKNug3bN173iT40uik+ExTDKDZrf32QFX1vJIN+kNlpiMuV weAZvHSiZGq6nklG/TTX1Woqg4AU1WyQf9ymb985eoAMFUlG/TlRsszih4Azpds0DOOHkCqGEdfIcbRA0gV4 +gr1OhxqABQK41+DijZoG/0OFQAqJVGPweUbNA3ehwqANRKo58DSjboASBVM6a3VVXPi6AHgAYrN5t6v WZZJ+gBoMF+OXyyqnpeyQb9zDJfccrVAaBZJNN1Y7vT9g9s77G92/bHsvpltrfZ3pe9zqxdc89p9FcfAKiVlLpuTk n6s4h4g6QbJd11+1pJd0vaHhHzJW3PtmvuWJmvOOXqANAsGp1fEw76iDgcET/K3v9a0h5JcyUtk7QpO2yTpO6c bQQA5FCTPnrb8yQtktQn6fKIOCyN/jGQNLvMOats99vuHxoaqkUzAAAl5A5626+W9C1JH4+IX1V6XkRsjIiuiO jq6OjI2wwAQBm5gt52m0ZD/usR8XBWPmJ7TrZ/jqSj+ZoIAMgjz6gbS/qqpD0R8cWiXVskrczer5T0yMSbV94dN 15VVR0ApqppOc69WdIfSRqw/VRW+3NJ90t6yPadkvZL+kCuFpaxvnuhJKmn74BGIlSwtWJx59k6AGDUhIM+I v5NUrlZ8m+d6OcCAGorzx39pFrXO6DNO/af3R6JOLvNXT0AnJPsFAg9fQeqqgPAVJVs0I+UeVa4XB0AmkWj B5Mk23VTsEuGOouDA2h2jR5MkmzQr1jceV4ffXEdAJrd+u6FDfs9MdmgZ3glAFTG0QR92l1dXdHf3z/ZzQCAp

NjeGRFd4x2X7I+xAIDKEPQA0OIIegBocQQ9ALQ4gh4AWlxTjLqxPSTpxzk+Ypakn9aoOQDQSHny6/URMe7K TU0R9HnZ7q9kiBEANJtG5BddNwDQ4gh6AGhxrRL0Gye7AQAwQXXPr5boowcAlNcqd/QAgDIIegBocU0b9Lb D9j8VbU+zPWT7O+Ocd8t4xwBALdgesf1U0b95dbzW/9qeNZFzm3k++pckXW+7PSKGJS2RNDjJbQKAYsMR8 abJbsR4mvaOPvNdSe/O3q+Q1HNmh+0bbP+H7V3Z64KxJ9u+1PYDtn+YHbesQe0GMEXZfovtf7W90/ZW23Oy +uO2v2T7Cdt7bL/V9sO299leX3R+b3bubturylzjDttPZt8i/sF24UJtavagf1DSctuXSHqjpL6ifc9LentELJL0GUmfL3 H+WkmPRcRbJb1D0gbbl9a5zQCmjvaibptv226T9LeS3h8Rb5H0gKR7i44/ERFvl/T3kh6RdJek6yV9yPZvZcd8OD u3S9JHi+qSJNtvkPSHkm7Ovk2MSPrghRrZzF03iohnsj6vFZIeHbP7dZI22Z4vKSS1lfil35N0u+1PZtuXSLpK0p76t BjAFHNe143t6zUa3NtsS1JB0uGi47dkrwOSdkfE4ey8/5HUKelnGg3392bHdUqan9XPuFXSWyT9MLtGu6SjF2pk Uwd9ZoukL0i6RVLxX7bPSfpBRLw3+2PweIlzLekPImJvndsIANJo5uyOiJvK7D+evZ4uen9me5rtWyS9U9JNEfG y7cc1eoM69hqbImJNpY1q9q4bafSrz19ExMCY+ut07sfZD5U5d6ukjzj7s2d7UV1aCACj9krqsH2TJNlus31dFee/TtI vspC/RtKNJY7ZLun9tmdn17jM9usv9KFNH/QRcTAi/qbErr+SdJ/tf9fo16NSPqfRLp1nbD+bbQNAXUTECUnvl/S Xtp+W9JSkt1XxEd/T6J39MxrNqx0lrvGcpHWSvp8dt03SnAt9KFMgAECLa/o7egBAPgQ9ALQ4gh4AWhxBDwA tiqAHgBZH0ANAiyPoAaDF/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 d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAAC XBIWXMAAAsTAAALEwEAmpwYAAC6wklEQVR4nOzdd5ibV5n38e9RmdEUTe/FMx733ksc1/TeICQEQgKE UBbYpS4sLEtbXlgWlrosoYZAeu/FSRzbce+9jqf3XtSl8/4xM46TuEyR9Kjcn+uayzOy9Ognaywdnec+91Faa4QQQ gghhLFMRgcQQgghhBAyKBNCCCGEiAgyKBNCCCGEiAAyKBNCCCGEiAAyKBNCCCGEiAAyKBNCCCG EiAAWowOMVE5Oji4vLzc6hhAixlRVVSGvLUKIYNu5c2eb1jp3ONeNukFZeXk5O3bsMDqGECLGLFy4UF5bh BBBp5SqHu515fSlEEIIIUQECOtMmVLqKuAbgz9OAT4LTAJuBKqBu7XW3nBmEkIIIYSIBGGdKdNav6y1Xq2 1Xg3UADuBNVrr5cA+4KZw5hFCCDF8Hf0e3D7/+y53+/x09HsMSCREbDHk9KVSqgJoBmYD6wYvXgssNSKP EEKI89ta2c79m6p4YHM1Lu87AzOX188Dm6u5f1MVWyrbDUwoRPQzqqbsFuApIAPoGbysG8g825WVUvcqpX YopXa0traGJ6EQQojT6rucAPS6fPS43qky6XF56XX5AGgYvI4QYnSMGpRdDzwLdAFpg5elDf78Plrr+7TWC7X WC3Nzh7WqVAghRBBdPDGHkswkFpZnkme3nb48z25jYXkmJZlJXDwxx8CEQkS/sLfEUEoVAB6tdbtSajvwOe C/gMuALeHOI4QQ4sLy02zcurD0rH+3YpJ8WBYiGIyYKbsReAZAa90CrFdKbQTmAk8bkEcIIcR7eHwBWnpca K2NjiJE3Aj7TJnW+vfv+fknwE/CnUMIIcTZ+QOah7fX0N7nYUZRGlfMKDA6khBxIeo6+ovha+l1sflkOydb+3G4 fWSlJjCnJIMFZZnYrGaj4wkhIpTHF6C9b6DFRVOPy+A0QsQPGZTFoIMN3fxy7XHWHm4moEEpsFnMOAeXs WcmW7lzaRmfXjWBlET5FRBCvFtSgplVU3I51drPovIso+MIETfkHTmGuLx+fvLyEe7fVEVakpXPrJrA1TMLm VZox2I20evysu1UB49sr+VXb5zg0R11/OxDc2TFlBDifeaPy2T+uLN2KRJChIgMymJEQ5eTz/59J3vruvno0nF87Y qppCdb33Udu83KpdPyuXRaPjurO/nXJ/Zx55+28m/XTOOeFRUGJRdCRKselxevL0B2amJIjt/t8BLQmsyUhHNep 6XHRarNQnKCvJ2J6Ce/xTGgur2fO/6wlW6nl9/fuYArh1GUu6Ask2f+6WK++thefvjCYfrdfr546USUUmFILISIdi2 9Lh7ZVosvoLlmViFTCuxBPX5Dl5PHd9YR0Jrr5xQxITf1fdfZUtnO5pPtJCWYuXNpmZRjiKhnVPNYESRVbf3c 9vstODw+Hr536bAGZENSEi385o75fGB+Cf+z9hh/3HAqhEmFELGkvc+DLzDQLqOlN/iLAdr7PPgDGq2hpcd91 uu09A5c7vT4T+8qIEQ0k48VUayj38Ndf9mGxx/goXuXMrUg7cI3eg+zSfHTD87G6fXxo5cOU5qVzFUzZfm7EO L8JuWl0lCSjssbCEnt2dRCO009Lnz+AHNLM856nYsnZKO1JteeSEG67azXESKayKAsSrl9fj79wA4au1089KnRD ciGmEyKn39oLg1dW/jyo3uYnL+cirOcKhBCiCEWs4lLp+WH7PhWs4nLp5//+Nmpidw4tzhkGYQINzl9GaV++Px htld18rNb57CgbOyfUm1WM7/76HwSLCa+8NBu3D5/EFIKEftae910OTxGxwirsTzmHpeX5jH2PvMHNPVdTlxe eZ0SsUUGZVHopf2NPLClmk+tGM/1c4qCdtzC9CT+6wOzOdjQw3+/cjRoxxUiVh1r7uXvW6r52+ZqGrudRscJiy NNPacf80gHV539Hv62qYoHt9awp7Zr1Ble3N/Io9treXRHrWwDJWKKDMqiTG2Hg68/sY85pR187cqpQT/+FTM KuGPJOP608RR7x/CiKUQ8aBssNPcHNB398TFb1tY78DhH85i7nF68/oFBVGvv2Yv3h2Poth397yw2ECIWSE1Z FNFa8/XH96E1/Pr2eSRYQjOm/sbVU3n9cDPfeHI/z37+YqxmGbsLcTbzxmXS4/KRaDExJT+4LSEi1fyyDPrcXhIt ZiaP8DGXZyezsDyTHqePJRWj3yng8un57KrpZFKeXV6fREyRQVkUeXh7LZsr2/nRzbMYl50csvtJs1n53g0z+czf d/KXt09x78oJIbsvIaJZUoI57lYrJydYuGpm4ahuq5RixaTcMWcozUqmNCt0r4FCGEU+YkSJpm4XP3rhMBdVZP PhxaUhv7+rZhZwydQ8fv36Cdr7Rn+aQQghQiUQCLCjqoPGrvio5xOxTwZlUeLfnzmANxDgxx+YFbau+/92zTQc Xj+/WHs8LPcnhBAj8aeNp/jpK0f51yf3jXlFpxCRQAZlUeDNIy28dqiZL102mbLslLDd78S8VO5YPI4Ht9VwoqU3 bPcrhBDDUd81MBBzewNjWjggRKSQQVmE8/gCfP/5Q1TkpvDxi8eH/f7/5bJJJFvN/PglaZEhhIgsH7+4nFkl6Vw7 u5CZxelGxxFizKTQP8L95e1TnGrr568fXxSy1Zbnk52ayL0rK/jZa8fYX9fNrBJ54RNCRIay7BS+fe10o2MIETQyU xbBWnpc/Or141w2LY/VU/IMy3HXxeWkJ1n55evHDMsghBi9XpeX6vZ+AjHW00trTW2Hg26n1+goQgSFDMoi 2M9ePYbXrw3/JJhms3LP8vGsPdzC/rpuQ7MIIUbG5fXz4NYantxVzxtHWoyOE1Rvn2jn8Z11/H1LNb0uGZiJ6Ce Dsgh1vLmXx3bWcudFZZTnhK+4/1xktkyI6OT2BXB4BvaI7IyxPTqHHo/HF8DpkX0wRfSTmrII9dNXjpKSYOGf 1kw0OgrwzmzZz147xoH6bimqFSJKpCdZuWxaPnWdDhaPH30X/Ui0clIuVrOJXHsieWk2o+MIMWYyUxaBdlZ3 8uqhZu5dWUFWSoLRcU772LJyUhLM/HFDpdFRhBAjMKsknatnFZKdmmh0lKBKT7Zy1cwCFpRlGh1FiKCQQ VmE0Vrzk5ePkJOayCdXhL8FxvmkJ1m5ffE4ntvXSL100BYCgLpOBy3SuFQIEQQyKIsw6462su1UB/986USSEy Lv7PInlg8MFP+y8ZTBSYQw3sGGbh7bUceD22qo63QYHUcIEeVkUBZBtNb89JWjlGUnc/vicUbHOavijCSum13 IQ9tqZBm6iHs9Th8AWkOvy2dwGiFEtJNBWQR59VAzhxp7+OdLJ2E1R+5Tc+/KCvo9fh7aVmN0FCEMNb8sg7 njMlgyPosp+Xaj4wgholzkvvPHGa01v1x7nPLsZG6YU2R0nPOaUZTO8ok5/PXtKnz+gNFxhDBMosXMmil5LJuY g8mkjI4jhIhyMiiLEK8NzpJ94ZJJWCJ4lmzI3cvKaepx8dqhZqOjCCEM0tzjoqY9smrpWnpcVLf3Gx1DiFGJ/Hf/O KC15pevD8yS3Tg3smfJhqyZmkdJZhL3b64yOooQwgD1XU4e2lbDE7vq2FfXZXQcAJq6XTy4bWD3gt01nUbHE WLEZFAWAdYebuFgQw+fj5JZMgCzSXHn0jK2VHZwtKnX6DhCiDDrd/vQg1tp9kXIIoe+MzLJwgsRjaJjBBDDt Nb8Yu0xyrKTuSlKZsmGfGhhKYkWE3+T2Tlh4s6kvFSWTchmflkmC8ojo3nrhNwUlk/KYd64jJjbvUDEh8hrhBV nXh+cJfvpB2dHzSzZkMyUBG6cW8STu+r5+lVTSU+yGh1JCBEmSimWVGQbHeNdlFIsKpfBmIhe0TUKiDFDt WRl2cncPK/Y6Dij8rGLynF6/Tyxs87oKEIIIURUk0GZgdYfb2N/fTefWz0h6mbJhswsTmdBWSYPbKkmENBGxx FCCCGiVnSOBGLE/755gsJ0GzfPKzE6ypjcubSMU239bK5sNzqKEEIIEbVkUGaQndWdbD3VwT0rKkiwRPfTc NXMAjKSrTwoHf6FEEKIUYvu0UAU+926E2QkW7l9UanRUcbMZjXzgfklvHqwibY+t9FxhBBCiKgkgzIDHG3q Ze3hFu5eVk5KYmwsgP3w4lK8fi0F/0IIIcQoyaDMAP/31kmSE8zcdVG50VGCZmKencXlWTy0rUYK/oUQQohR kEFZmNV2OHh2bwMfXjyOzJQEo+ME1YeXlFLV7mCLFPwLIYQQIyaDsjC7b30lJgX3rBhvdJSgu3pmIelJUvAv

hBBCjEbYB2VKqY8ppV5XSq1TShUrpb6mlNqolPqHUiqmW8K39rp5dEctt8wroTA9yeg4QWezmrllfjGvHGyiX Qr+hRBCiBEJ66BMKVUMrNJaX6q1Xg14gDVa6+XAPuCmcOYJt7+8fQqPP8C9qyqMjhIydyweN1Dwv0sK/oU YK601R5t6qe1wGB1lWGo7HTy9u562PpfRUYSISuGeKbsSMA/OlP0aWAysG/y7tcDSMOcJmx6Xlwc2V3P1zAI m5KYaHSdkJuXbWVSeyUPbatFaCv6FGIud1Z28uL+Rx3fWUd/lNDrOeQUCAb737EEe2lbDD58/YnQcIaJSuAdl +UCC1vpSwAFkAD2Df9cNZJ7tRkqpe5VSO5RSO1pbW8MSNNj+vqWaXrePz66aaHSUkPvw4nHS4V+IIHD7Au 987/UbmOTCAoF38rp9kZ1ViEgV7kFZN/DW4PdvAOVA2uDPaUDX2W6ktb5Pa71Qa70wNzc31BmDzuX18+eN VayYlMOsknSj44TcNbMKSbNZeHCrFPwLMRaLx2dx0YRsLp2WR0WEz7BbLCa+fPlkVk3J5UuXTzY6jhBRKd yDsk3A7MHv5wK1wKrBny8DtoQ5T1g8vrOOtj43n109wegoYTFQ8F/CqwebpeBfiDGwmk0srchmdkmG0VGGZd 64TD63eiKT8+1GRxEiKoV1UKa13gM4IVLrgEXAw8B6pdRGBgZpT4czTzj4A5o/bTzF7JJ0LqrINjpO2NyxZBw ef0AK/oWIMnWdDg7Ud+PzBy585VGo7XBwsKEbfwQ1ma7tGHjMkZRpJKra+jnc2CN1vDEg7Hv8aK2/+p6LfjL 4FZNeO9TMqbZ+fnPHPJRSRscJm8n5dhaWDRT8f2pFRVw9diGiVVufm8d31qE1tPd7WDU5uOUizT0untg1cPxu h5dlE3OCevzRaO11n87U6fCwYlJ0lcjUdjh4anc9AP1uHwvLswxOJMZCmseG2B82VFKSmcRVMwqMjhJ2QwX/ Wyo7jI4ihBgGf0AzNNni9QV/pszrD5w+vidEM3Ej5Qu8k8kbIZlG4szMXr/MlEW72NgNO0LtrO5gZ3Un371+OhZ z/I1/r51dyPeeO8iD22q4aEL8nLoVIIrlp9m4dnYhHf0e5pZmBP34JZnJXDWzgB6nl3njzrrYPuwK05O4ZlYhnQ4P8 8ZIGB1nxCpyU718ej5Or595IXjORHjJoCyE7ltfSXqSlVsXlhodxRBDBf8Pbq2hvc9Ndmqi0ZGEEBcQ6iL9aYVpF7 5SmE0piO6FCTOLY39Vf7yIv+mbMDnV1s+rh5q5c2kZKYnxO/b98OKBgv8nd9UbHUWIkDrZ2seB+m4CUVos LiJLbYeDvbVdUXlKVYyeDMpC5I8bKrGaTHxsWZnRUQw1pcDOgrJMHtpWIyuDRMyq7XDw7J4GXjvUzM6a TqPjiCjX3jew+OCNIy1sPN5mdBwRRjIoC4H2wRVMt8wvJs9uMzqO4e5YPI5KKfgXMezMVgo+KbYWY+TX7 yy4iNY2HWJ0ZFAWAn/bXI3bF+CeFeONjhIRrp090OH/oW3S4V/EpvKcFK6cUcDKybksKo+MAnYRvfLsNq6f U8TFE3NYMdn4tiEifOK32ClEnB4/D2yp5tKpeUzMi+7i0WA5s+C/o99DVkqC0ZGECLrpRZFXwC6i18S8yN5W S4SGzJQF2RO76ujo93Dvygqjo0SUoYL/J3ZKh38h4lllax97artCtmNAKPgDmn11XRxv7jU6iohxMigLIn9A88cNl cwpSWfxeOmqfCYp+BdCNHY7eWZPA28eaYmqGtOd1Z28friF5/c1Ut3eb3QcEcNkUBZErx1qpqrdwb0rJ8i2Qmf x4cGC/62noufFWAgRPGd+HtNEz4ezMz9IymdKEUpSUxZEf9xQSWIWElfOyDc6SkS6bnYh33/uIA9urWFpHG3 OLoQYUJSRxPVzCul2+phdEj0NTxeWZ5FgMZGcYKE8J8XoOCKGyaAsSPbWdrGjupPvXBefWyoNhxT8CyGic QGU2aQiZlsoEdtk9BAkf3n7FKmJFm5dWGJ0lIg2VPD/2I5ao6MIIUTQ9Li8bDvVQVO3y+goIorJoCwImntcPL+v kQ8tLMVusxodJ6JNKbCzZHwWf9tcLU0RhRAx46X9jbx9oo0ndtXh8UXPylIRWWRQFgR/31KNX2vuXlZudJSo cPeycuq7nKw93Gx0FCGECArFO4u7ZJ2XGC2pKRsjl9fPP7bWcNm0fMZlJxsdJypcPj2fonQbf327iitnFBgdRwghx uya2YUcbeqhNDMZq9QVi1GS35wxemZPPR39Hj5xsWypNFwWs4k7Lypnc2U7R5p6jI4jhBBjlppoYUFZFnlpst+ xGD0ZII2B1pq/vF3F1AI7SyukWexI3L6olESLifs3VRkdRQgRARweH1sq26lpdxgdRQjDyKBsDDafbOdIUy+fWD 5emsWOUGZKAjfPK+ap3fV0OTxGxxFCGOy1Q81sPtnO03vq6XP7jI4jhCFkUDYGf377FNkpCdwwp8joKFHpr mXluLwBHtku7TGEiHdm08AHWzX4JUQ8kkHZKFW19fP6kRY+srQMm9VsdJyoNK0wjaUVWdy/qQpvFG1OL IQIvsum5bNmah63LiwlJVHWoIn4JIOyUbp/cxUWk+KjS8cZHSWqfWpFBQ3dLp7b22B0FCGEgWxWM3NLMy hIl0J5Eb9kUDYKDo+Px3fWcfXMQvLs8gIyFmum5DEl387/vXWSgDSTFXGurc/N2yfaaO6J3K7wPn+A7VUd7K /rNjoKTo+fzSfbOdHSa3SUqBcNv3vxQAZlo/Dc3gZ6XT7uvKjM6ChRz2RSfHpVBcea+3jzaIvRcYQw1LN7Gth2q oMnd9WjdWR+SNlZ3cnG422sPdzMiZY+Q7O8dayFLZXtPL+vkfY+t6FZot0zg797T+2uNzpKXJNB2QhprXlgSz VT8u0sLJMNaoPh+jlFFGck8bt1J42OIoShrGZ1+s9IXdFtOaMx6lBeoww1aTUpdXqhgBidoefSIv+OhpJqyhHaW9f NgfoefnDTzIh90Yw2VrOJe1aM53vPHWJ7VQeLyqXnm4hPN80rprK1n/LsFKOjnNP8cRkkJ5ixWc2UGZxz5eRc8 tNsZKUkkJGcYGiWaHdzFPzuxQOZKRuhBzZXk5Jg5uZ5xUZHiSm3LSolM9nK/755wugoQhjGbrMypzSD9GSr0 VHOSSnFtMI0xucY/+ZtNZuYWZxOUUaS0VGiXjT87sUDGZSNQGe/h+f2NXDz/GJSZcl2UCUnWPjk8vG8ebS V3TWdRscRIuLVdTpYf6w1bLVU/W4fG4+3cbxZiuqFCBUZlI3A4zvr8PgCfHSpFPiHwt0XjycrJYGfv3bM6ChCR DSfP8DTu+vZWd3Ji/sbw3Kfbx5tYXtVBy/sb6SzX3bhECIUZFA2TIGA5u9bq1lUnsnUgjSj48Sk1EQLn101gQ3H2 9ha2W50HCEilkkpEiwDL9+JlvA0r04YLKo3K4XF4AJ/IWKVDMqGacOJNqrbHTJLFmIfXVpGnj2Rn712LGJbA ghhNJNJ8aGFpVw+PZ/rw7TN2yVT87hiRj63LSrFbpO6IyFCQQZlw/TQ1hqyUhK4amaB0VFiWlKCmc9fMpFtpzr YcLzN6DhCRKyM5ARmFqeTlBCemTKL2cSMonTy0qRhthChIoOyYWjrc7P2cDO3zCsO26mCeHbbolJKMpP40 YuH8cmemCLKdTk8rDvaQmWrsY1WRXQ52NDNhuOtODw+o6OIMJJB2TA8taseX0Bz26JSo6PEhUSLmX+7Z hpHmnp5eHut0XGEGJNXDjaxu6aL5/c14vT4jY4jokBzj4tXDzazo6pTzhjEGRmUXYDWmkd21DJvXAaT8u1Gx4 kbV88sYPH4LH7+2jG6nV6j4wgxajbrwOy61WySrvNiWBLO+F1JssrZmXgig7IL2FXTxYmWPm5bKLNk4aSU4j vXTafT4eGXa48bHUeIUbtqZgFXzijgw4tLT6+YFOJ8MlMSuH1RKdfMKmT5xByj44gwkleIC3h0ey3JCWauC9M KJ/GOmcXp3L6olPs3V3GgvtvoOEKMSqLFzPSiNNkGSIxIXpqNKQV2TDK7GldkUHYe/W4fz+9r4NpZhdLB3y DfuGoamckJ/OsT+6ToX4gwOtnaxxtHmsO2Y0Co1HY4eONIM43dTqOjCHFBMig7jxf2NdLv8UuBv4HSk618/8Y ZHGzo4Y8bTxkdR4i44PL6eWFfI3tru3nlYLPRcUZNa82zexvYW9vNC/vCs/OBEGMhg7LzeGRHLRW5KSwoyz Q6Sly7emYBV87I5+evHeNIU4/RcYSIeWaTInGw/i0lMXoLzZVSpAz2cUtOkLMdIvLJoOwcTrX1s7O6kw8tLEU pOadvJKUUP7xpFmk2K198aLe0FRAixKxmE7cvHsc1swq5emah0XHG5NaFAwXzN88rNjqKEBcU1kGZUqpcK dWslFqnlHp18LKvKaU2KqX+oZSKmL07nt5dj1Jw41wp8I8EufZEfv6hORxr7uOHLxwyOo4QMS89ycqUAnvUrx hNSbQwpcAetp0PhBgLI/63vaa1Xq21vkIplQus0VovB/YBNxmQ53201jy9p551E7IpTE8yOo4YtHJyLp9eWcE/ttbw

2A5pKivEaHX0e1h7qJmjTb3v+7s9tV28caSZfrcxneRdXj/rjrawvaojKPvf1rQ7eO1QM/VdkVfo3zn4PEhZhhhixEn2 NUqpDcCTwDFg3eDla4E7gMcMyPQuu2u7qG538Pk1E42OIt7jq1dO4UBDN//21H7Kc1JYVJ5ldCQhos7aw83Ud zo50NBNcWbS6dXlDV1O3jzSAoDXr7lyRvj3+t16qoPdNV0AZKckUJGbOupjaa15bl8DHl+AqrZ+PrWyIkgpg2Pt 4Wbqhp6HjCTZ6F2EfaasEZgMrAEuAxYCQx8RuoGzVtQrpe5VSu1QSu1obW0NecindtWTaDHJ5uMRyGo28b93 LKA0M5lPP7CTU239RkcSIurYBwdhNqsZq/mdmtnkBDOWwb5YdoPaAA0NEE1KkTLGDEqp08dLtUVeob99M FOixYzVHN2niUVwqGBMD4/qjpX6LAMDsmKt9X8ppeYDH9Vaf/l8t1u4cKHesWNHyHJ5fAGW/Ggtyyfl8usPz wvZ/YixqWzt44P/t5lEi4lHP30RpVnJRkcSUW7hwoWE8rUlkvj8AaraHeTaE0lPevfsTFufmx6nl/E5KYYtcqpq6yc5 wUxemm3Mx3J4fDR0OSnJTD695VWkON/zIGKHUmqn1nrhcK4b7kL/MzePvBg4Aawa/PkyYEs485zN+mOtdDq 83DxPCvwjWUVuKn//5BIcHj8f/sMWajscRkcSImpYzCYm5qWedSCQk5pIRW6qoavOy3NSgjIgg4FWGBPz7BE 3IIPzPw8iPoV7vnSFUmqnUmoT0KC13gqsV0ptBOYCT4c5z/s8taee7JQEVkzKNTqKuIDpRWn8/ZNL6HF6ufl/N 7GvrsvoSEKEzYmWPl4+0DSiTvXV7f28fKDpfR9ifP4AG4638taxVrwh3DlDa832qg5eP9yMwzP6hQT76rr4fy8e5 o3B+rczj7+lsp03jjTj8krrHBF9wjoo01q/qLVeoLVeprX++uBlP9FaL9da36G19oQzz3v1uLysPdTM9XOK5Px+lJhV ks6Tn1uGzWritt9vka7dIi74/AFe3N/I4cYeXjnQNOzbPb9v4DYv7H/3/5NDjT3sqOpkV3VnSD/c1HQ42Hi8jX113 Ww+2T7q4/xu3Un21Hbxxw2V7xrcnWztZ/PJdvbWdrPtVEcwIgsRVjLyOMPaQ824fQFukN5kUWVinp0nP7eMqY V2/unBXXzzyf3SYFbENNMZBexpIzj1NXSa7L2ny9LOWPWXFsIVgKmJFsyDCwnGcsouKyXh9PESTO+8jdltF kyDp11H8u8iRKQwrNB/tEJZ6P/Jv27nSFMvG/91jXTxj0Jef4D/fvUov3+rknFZyXzvhhmsmZpndCwRJaKt0N/h8d HY7aI0M3nYDV5dXj/1XU6KM5LeV2PV1O0ioDVFGaHtzdje56bf7Wdc9ugX5/S5fOyo7mBmcRo5qe+uPWvtde Py+mXxj4gYEVvoH8m6nV7WH2/lmlkFMiCLUlaziW9ePY0HP7UEq1nx8b9u5577d3C4URozitiTnGBhQm7qiDr u26xmJuSmnrXovSDdFvIBGUB2auKYBmQw0N5i9ZS89w3IYGD3DxmQiWglg7JBrx1qxuvXXDtbTl1Gu2UTcn jpn1fy9aumsLWynat/uYFPP7CDPbVdRkcTwlCn2vp5cX8j1e3G9PcbKsR/9WDTiHYM2FndwcsHmthR3cGL+xtH tLghVHbVdPLygUa6HIaWQosYE3nd9Azywr4GijOSmFOSbnQUEQQJFhOfWz2Rjywu489vn+LPb5/ilYPNzCpO 5861ZVw/p0j2whNx58X9jXh8AarbHXx29YSw3391u+N0gb/ZpLh0Wv4Fb9PU7WL9sTb8Ac2ptn4m5qXS2uvmr mXIIU57bi29Lt46OtDI3OPX3DBHPsyL4JCZMqDb4WXD8Taum10opy5jTHqylS9dPplN37iE7984A7fPz9ef2MeS H63lu88e5Hjz+/f+EyJWZSQPFL9nJhtTBG+3WU7vGJAxzAzJiWYSLCZM6p0C/+HeNlRSEiynTxtnyIICEURS6 A88uqOWrz++j2f+6WLmlGYE9dgismit2Xaqg79vreHlA414/ZpF5ZncsWQcV88sjMgGkyI8oq3QfzRcXj9N3S4K M2wkWoz5Xe/s99Dv8VGSOfy6r26Hlx6Xl+yUBFp63ZRkJmExuG1Rt9NLt8NLaVaSfJgX5zWSQn85fQm8sK+R kswkZsupy5inlGJJRTZLKrJp75vO4zvreGhbDV96ZC/fe+4QH5xfwieWjw9LwbMQ4WazminPSTE0Q2ZKApmD M17DlZ5sJX1wdqzcoD053ys9ySqd+EXQxf3py85+D2+faONaOXUZd7JTE/n0qgm88ZXV/OOeJVw8IYe/bqpi9U/ X8e9PH4iIYmIhwm1LZTsv7W+k2+kd9m12Vnfywr5GOvpDU/S+7dRAgX8oiuq11mw+OfCYe1zDf8xDjjb18tzeB tnqLQycHj+vHGxi/bFW/IHoOss3XJHxkcNArx1qxhfQXDur0OgowiAmk+LiiTlcPDGHuk4Hv33zJA9tq+GxnbV8f s1EPrWywrBTPUKEU13nO4X4SsFVMy/8utjW52b9sYGid68/wE3zioOaqbHbydsn2gAIaM11QV4hX9fpZEvlwG M2mRRXzigY9m19/gAvH2gioDUtvW4+uXx8ULOJd9tR3cGhhoEWR3lpiUwtSDM4UfDF/UzZq4eaKM5IYlaxn LoUUJKZzP+7ZRZvfnU110zN479fPcZVv9jAzmrZskXEPrvNerqAPSslcVi3SU4wn67FzBrhacnhSE20nJEp+MdP O+MxZ4/w+GaTIjPFOqrbipEbev7NJkVmcmz+e8d1ob/D42Pe91/jw4vH8d0bZgTlmCK2vHWslW8/vZ/6TidfvHQS n18z0fACYxEa8VDoPxw9Li99Lt+I6ir73D66HB6KM0JT9N7j8tLr8lEcolrPbqeXfvfIHvMQl9dPa6+bwnSbvDaEQ XOPiwSzacR1iUaSjv7DtP5YG25fgCumX7hXjohPqybn8uIXV3DT3GJ+sfY4n7h/x6jqToSIFmk264gHJ6mJFkoyk0 NWl5tms4ZsQAYDRfujXdxjs5opzUqWAVmY5KfZompANlJx/Vv06qEm0pOsLBqfZXQUEcHsNis/v20uP75lFpt OtHHL/26ipl2KesXouH1+XjvUzGuHmnH7/EbHCSutNRuOt/LCvtEV1Yuxcfv8vHqwidcONePxBYyOI84ibgdlPn +AN460cOnUPKzyCUcMw+2Lx/HAJ5fQ2uvm1t9v4kRLn9GRRBQ6UN99xld87cta1e5gR1Unx5p72VopdZrhdqC +m4MNPQO/ew3dRscRZxG3o5HtVZ10ObxcLqcuxQhcNCGbRz69FH9Ac/t9WzjaJDsCiJHJSU1EqYHVjTmpsXs a5mwykqxYzQOnOHPtw1tIIIInO2Xgd8+kFDnDXMghwituW2K8eqiJBIuJlZNzjY4ioszUgjQevvci7vjDFj76p608+ dlllGYNvzu5iG9l2Sl87KJyIDSrCSNZZkoCH1tWjtPjJz/NZnScuFOek8KdS8tQSsXd7160iMuZMq01rx5sZsXEHFI ipDu0iC4T81L5+z1LcHv9fOzP22jvcxsdSUSRrJSEuH1TTLNZZUBmoOzUxLj93YsGoxqUKaXylVJ/Ukq9NPjzd KXUJ4MbLXQONfZQ3+Xkihly6lKM3uR8O3++exENXU4+ef+OuCvaFsF3oqWPp3bXcaQpNmvNAgHNuqMtParticles (Control of the Control of the ContLu3YUQ7BggxWi6vn5cPNPLKwaaoWNww2pmyvwKvAEOtlY8B/xKEPGHx2qFmlIJLp8mgTlzNwvIsfnHbXPb UdvGdpw8SbX3/RGRZe7iZqjYHrx5sjsnfpeoOB7trujjZ0se2U1LoL0JvX103hxt7OdTQw6HGyP+wM9pBWY7W+ lEgAKC19gFRM03w5pEW5pZmkJMqhY5i7K6eVcjn10zkkR21PLitxug4Iorlpw28JuXZE2NyL97M5He65w89ViF CKdd+xuKGKFhYM9qCqn6lVDagAZRSS4GoWF/b2utmb103X7l8stFRRAz50uWTOdDQzXefPciMonTmlmYY HUlEoetnF9HW5yE7Ct48RiMjOYG7Bgv9ZfWlCIfxOSncdVE5JqVIT7YaHeeCRjtT9mXgWWCCUupt4G/AF4K WKoTeGtw4d83UPIOTiFhiNil+eds88uw2/uXh3fS7fUZHElHIYjZRkG6L6d6JqYkWGZCJsMpMSYiKARmMclC mtd4FrAKWAZ8GZmit9wUzWKi8ebSFPHsiM4pib3d5Yaz0ZCs//9AcqjscfP+5Q0bHEUKM0omWXh7fWcfBYT RY3V/XzeM766hslWbSYuxGdfpSKXXLey6arJTqBvZrrVvGHis0vP4A64+1cvXMgpis1xDGW1KRzedWT+C3b5 5k9ZRcrp5VaHQkIcQIvX64BYfHT32nk2kFaZhMZ3+/8PkDvH6kGa2hy+GhIjc1zElFrBntHPkngT8CHxn8+gMDp zTfVkrdGaRsQberupNel481U+TUpQidf7lsMrNL0vn20wfo7PcYHUcIMUIF6bbBPxPPOSCDgbKFPPvAdQvTQ7 dhuogfoy30DwDTtNbNMNC3DPgdsARYDzwQnHjB9ebRViwmxfJJOUZHETHMajbxkw/M5vpfb+QHLxzi5x+aa

3QkIcQIXDe7iPZ+N1nJ519woZTi1oUldDo8sm2RCIrRzpSVDw3IBrUAk7XWHUDEdgRcd7SFReVZ2G3RUfAn ote0wjQ+u3oCT+6qZ93RiD2jL4Q4i6EZMMswFlxYzSby7LbzzqgJMVyjHZRtUEo9r5S6Syl1F/AMsF4plQJ0BS1d ENV3OTnS1MuaqbLXpQiPz18ykQm5KXzrqQOyGlPEHK8/wMsHGnl6dz29roj9LH5BgYDmjSPNPLGzjo5RlBv sqe3i0e21nGjpDUE6EW9GOyj7J+AvwNzBr22A1lr3a63XBCdacA3NVlwirTBEmCRazPzXB2dT3+Xk12+cMDq OEEF1vLmPw429nGrrZ3dN19FxRq2208He2m5qOhxsO9U+otv6/AHWHW2hvsvJuqOtIUoo4sloW2Jo4CQDpypv Bi4FDgcxV9C9eaSVkswkJsjqGBFGC8qyuHVBCX/aWMlJWTIvYkheWilJFhNKQVFG9G4wnpWSQFKCGYCijJ EV61vMptObqxeP8LZCnM2ICv2VUpOB24EPA+3AI4CK1NmxIW6fn7dPtPHBBSXSCkOE3devmsrLB5r47rMH +dsnFsvvoIgJOamJfPzicrx+TXpS9Nbp2m1W7l5WjsvrJ+MChf1nc+uCErqc3gsuChBiOEY6U3aEgVmx67XWy7X WvyYK9rzcWd2J0+tn1WSpJxPhl2tP5EuXT2bD8TZePdR84RsIESWSEyxRPSAbYrOaRzUgg4HZspzU87fOEGK 4Rjoo+wDQBLyplPqDUupSIOJ/Ezccb8NiUiydkG10FBGnPnZRGVPy7fzg+UO4vBH/OUZEkRMtfTy8rYYdVR1 GRxGj1Nnv4bEdtbx8oAmfP2B0HGGgEQ3KtNZPaa1vA6YC64AvAflKqd8ppa4IQb6g2Hi8jfnjMklNHG1bNiHG xmI28R83TKeu08mf3z5ldBwRQzYcb6Wx28WG420y4I9SO6s7qet0crixh6p2h9FxhIFGW+jfr7X+h9b6OqAE2AN 8I5jBgqWj38OBhm5pGCsMt2xCDpdNy+N3b56kvc9tdBwRI0ozk4GBLvSJltjdyDyWFWcmodTAadTcVGlCG8/G PHU02DD294NfEeftE21oDStkUCYiwDeunsqVv9jAr984wXdvmGF0HBEDLp2Wx4KyTOw2iywiiVLTCtMozkw iwWzCZjUbHUcYKOY/Vm043kqazcLskgyjowjBxDw7ty0q5e9bqjnV1m90HBEDlFJkpiQMq/u8iFxpNqsMyERsD 8q01mw83sbFE3Mwy8oYESH+5bJJJFhM/NfLR4yOIoSIQ/6A5uUDjTyyvYa2c5RSdPZ7eHRHLc/va8Ariw/CJqY HZZVt/TR0u6SeTESUPLuNT6+cwEsHmthZLSvmhBDhVd3ez+HGXhq6XOyo6jzrdfbUdlHf6eR4cx+VrTKrHy4x PSjbcGxg24uVk6Q/mYgsn1o5njx7Iv/vxSMMbJAhhBDhkWtPJCXRjFJQlp181uuUZiVhUgqb1Ux+miw+CJeY7h Gx8UQbZdnJlGad/ZdOCKMkJ1j44qWT+PbTB1h3rJU1U2RPViFEeAzsYjAejz9wzlZRE/Ps3LMiCYtZkWiRWrd wMWSmTCn1ZaXUxsHvv6aU2qiU+odSKmitob3+AJtPtsuqSxGxPrSwlNKsJP77laMEAjJbJoQInwSL6YK9O1MS LTIgC7OwD8qUUonAnMHvc4E1WuvlwD7gpmDdz+6aLvo9fpZPlFOXIjIIWEx86bLJHGzo4eWDTUbHERGkpt3 BA1uqeeNIc0yd3vb5Azy/r4EHt9bQ2ju8Xn2BgOblA038fUs1jd3OMWdo6XXx4NYaXtjXiM8fwB/QvLi/kX9sraal xzXm44+F0+PnyV11PLqjll6X19AswhhGzJTdA9w/+P1iBnYGAFgLLA3WnWw43orZpLhItlYSEezGucVMykvlZ 68exS+zZWLQ1lPttPW62VvbTXu/x+g4QVPd4eB4cx/NPS521Zy9wPy9GntcHG7sobXXzfZzFKWPxK7qLpp7XB xr7qW200lth4OjTb209LjZWT3244/FkaYeqtsd1Hc6OdjQY2gWYYywDsoGT0+u0lq/MXhRBjD0m9cNZJ7jdvcqp XYopXa0trYO6742nmhjTkl6TGyWK2KX2aT4yhWTOdnaz1O7642OIyLE+JwUALJTE0izxc5rWN5ggblJKcqzU4 Z1m+yUBNKSrCgF5ecoSh+J8pxklILURAu59kRy7YmDjXehbJiZQqU4M4kEiwmLSVGSmWRoFmGMcBf63wk 8eMbPXUDx4Pdpgz+/j9b6PuA+gIULF15wOqHP7WNfXTefXTVhLFmFCIsrZxQwqzidX6w9xg1zikiQrXLi3sLyL KYVpmGzmmOqx6LdZuXjF4/H59ckJQyvVslmNXPXRWV4/AGSE8b+ljW1II2yrBQsZoV1sOHuXcvK8Qbp+G ORZ7dxz4rxaI00ko1T4X71nwJ8Vin1MjADWAisGvy7y4AtwbiT7ac68Ae0nLoUUUEpxVevnEJdp5NHttcYHUdE iJRES0wNyIZYzaZhD8iGWMymoA6YkhLMpwdkQ5mMHpANSbSYZUAWx8I6KNNa/6vW+kqt9VXAQa3194 D1gysx5wJPB+N+Nle2k2A2saDsrGdDhYg4KyflsHh8Fr964wROj9/oOELErOYeFw9sqebZvQ34/AF8/gDP7W3gg S3VNHWPvNB/88l2/vr2KQ7Ud4cgrYg3hp0nGVxxidb6J1rr5VrrO7TWQalo3XSyjXnjMuTThogaSim+duUUWnv d3L+5yug4QsSs3TWdtPW6OdnSR22nk7pOJyda+mjrdbN7mIsPhvj8AbZUttPp8LKlsj1EiUU8ibnilW6Hl4MNPSyb IP3JRHRZVJ7F6im5/N9bJ+mR5fBChERFbiompbDbLOTZE8lLGyj0NynF+NyRFfpbzCbKc5IHj2vsIgERGyLjJHo QbTnVjtZIPZmISl+9YgrX/Xojf9pwii9dPtnoOELEnMn5dsZlJWM1m07X7H384vF4/YFRnV25aW4xLm9gxHVyQ pxNzM2UbT7Zjs1qYk5putFRhBixmcXpXDOrgD9uqKQjhvpTCRFJ3ruq1WxSoy53UUrJgEwETcwNyrZUtrOoPE u2hhBR68uXT8bp9fN/b500OooQQeXxBXh6dz1/21xFcxC753f2e/j7lmoe3V6Lw+N7199trWznTxtPnbMxrNcf4Jk9 9dy/qWpUOwa8faKNP208xb66rtFEjzvHmnv588ZTvH44tnarCJaYGpS197k50tTL0go5dSmi18Q8OzfPK+H+TcF9 4xLCaLWdDk619dPe52FPbVfQjntosON/fddA0f4QrTVbKjvocXrZeurshfj1nU4qW/vp6Pewp2Zkmbz+ANtODR6/s mMsDyFubK/qoNvpZV9dNz0u34VvEGdialC2ZfA/hdSTiWj3L5dNIqA1v37juNFRhAiagjQbdttA/7UJualBO255Tg pW88BpxJLMd7r+K6WYmDdwP5Py7Ge9bX6ajfQkKyalmJA3skxWs+l0gf+k/OA9nlg29DwUZdguuCF6PIqpf5H NlW2kJlqYXSz1ZCK6lWYlc9uiUh7eVsu9KyYwLgjbywhhtJREC5+4eDy+gA7qzhXFGUl8etUETEq9r+HutbMLu dSbd86asaQEM3cvKx91phvmFOH2jW6RQDxaPD6LOaXpJJhNKBV7zZHHKqZmyjadbGdReSYWc0w9LBGnvn DJJMwmxS9eP2Z0FCGCxmRSIdlK7MzV1O91oQHTWDIpNfpFAvEq0WKWAdk5xMzopbnHRWVrv5y6FDEjP8 3GXcvKeXp3Pcebe42OI87D5fXz6PZa7t9URUuv1AHGkw3HW/nD+sqg1sgFW32Xkz9vPMVTu+vw+gNGxxHn ETODss0nB4o4pWmsiCWfWTWB5AQLP39NZssi2am2fuq7nHT0ezjY0GN0HBEmXn+AHVWd9Ll9bD8VuYX+ e2u76HZ6qWpzUN858hWmInxialCWZrMwrTDN6ChCBE1WSgKfXD6elw40sb9O9taLVMWZSdhtFqxmxcQgFr CLyGY1m04vDphScPaFBJFgUl4qZpMiM9lKfprN6DjiPGKm0H9zZTtLKrLPWVMgRLS6Z8V47t9cxX+/epT7P7 HY6DjiLNJsVj65fDwBjbwGxZkb5hTh9QewRnAt86R8O+NzUjCblNRyRbjI/S0agbpOBzUdDpZJPZmIQXablc+u msBbx1rZFsGnSOKdOsvKPxEfInlANsQiqx2jQuT/Jg3DUD2ZFPmLWPWxi8rJtSfy368clS7YYlhcXj+PbK/hTxtP 0dQtiw8i0ZGmHu5bf5KX9jeiteZoUw/X/3oD1/xyPXtqzr4DQbg097j488ZTPLytBpfXb2iWeBITg7Kq9n5yUhOYfI 7mgEJEu6QEM1+8ZCLbqjpYf7zN6DgiCtR0OGjoctHj9HKgXuoRI9Humi763X6ONPXS7fTy9J4GOvo9dDu9PL GrztBsBxu66XZ6aex2Ud3uMDRLPImJQdnXrpzKhq9fgklOHYgYdtuicZRkJslsmRiW4owk0pOsWM2KyfnygTUS TStMQykoyUzCbrNy5fR8kqxmEixmrpxRYGi2SXl2Eiwm0pKsFGcmGZolnqhoe3FfuHCh3rFjh9ExhDDE4zvr+O

pje/nNHfO4bnaR0XFiysKFC4nF15ZAQMsH1gj23ufH7x84VWg2G9+QVn53gkMptVNrvXA4142JmTIh4sXN84q ZWmDnJy8fwe2TOg9xYfKmGtne+/yYzeaIGJCB/O4YIWZaYggRD8wmxbeuncadf9rG3zZV86mVFUZHiksnWvp Ye7gZm8WExx8gyWrm5vklJFnNPL+vgYYuF5dMzWNKgZ39dd1sONGKPdGCw+MnMzmBG+cVkWh5/xvv5p Pt7K7tZEZROqsm5xrwyMSZjjX38vrhFooybFw3uygiV9c6PD6e3FWPy+vn+jlFMd2HrLnHxXN7G7BZzdw8r5iU GNzQXGbKhIgyKyblsnpKLr9+4zid/R6j48Sl/fVdOD1+9tR20dzjoq3Pw6nWfjr6PVS29uPy+tlX1wXAntpO3N4Am yvb6XJ4qO9y0tB19tWQuwevu7umU+oGI8De2i5cXj+Vrf2097uNjnNWVW0OWnvd9Lp8HGqM7d0kDjf20Ovy0d rrpqq93+g4ISGDMiGi0L9dM40+t49fvn7c6ChxaXphOhaTYlphGpkpCaQlWRmXnUxm8kBRtNmkmF40sLvIjOJ0 TEoxb1wmyYkWcuyJFKaffTZjZtHAdWcUpUtPqQgwvSgNs0lRnJlEVnKC0XHOalx2MhnJVhKtpphf0DE5306i1 UR6kpVxWclGxwkJKfQXIkr921P7eXR7La9+aSUVsrXPmI200F9rjVLq9J9n+7vhXPdcxxWRIVqej2jJOVbR+Dil 0F+IOPClyyaTaDHx45eOGB0lLg29MZztDeK9l53vuhe6rTBWtDwf0ZJzrGL9ccqgTIgolWtP5LOrJ/DqoWY2nZC GsiI21Hc5+f1bJ3lgSzVOT+SvMP7cP3ax6r/e4PvPHTQ6SlA1nPE8ODw+o+PEDRmUCRHF7llRQWlWEt959iBef 8DoOEKM2ZHGHhweP229bmo7I7uTfLfTw97aTnwBzfpjsfXB6EjTGc9Dh9PoOHFDBmVCRDGb1cx3r5/BiZY+ /vL2KaPjCDFmUwrs2KxmclITKInwTvLpSQnMLsnAbFIsmxhbey9PKUgjKWHgeSjNiuznIZZIob8QMeCe+7ez6 WQ7r39lFYXp8gI6GrHa0V8IYSwp9BcizvzH9TPwBzT/+cJho6MIIYQYJRmUCREDSrOS+ac1E3l+XyNvS9G/G KHOfg9/3FDJfetP0tobmU1SY1lth4PfvnmCBzZXRcXihrHYW9vFr14/zjN76gkEoutMXTjIoEyIGHHvygrKs5P511 P7Y/6FXQRXZVs/vS4f/W4/J1v7jI4Td4409eLxBWjr81DfFdmLG8bqQEM3/oCmsrWfXres6nwvGZQJESNsVjM/ umUWVe0O/mftMaPjiCgyMTeVzGQr6UnWmO8KH4mmFdpJSTSTn2ajJDM2O9UPmVOSQYLFxKT8VNJssbd 35VjJv4gQMWTZhBw+vHgcf9xQyTWzCplbmmF0JBEF0pOt3H3xeKNjxK2SzGTuXTnB6BhhMbM4nZnF6UbH iFgyUyZEjPnmNVPJT7Px9cf34vbJaUwRWuFawR/uTgHvvb8zfx76PhSZtNYhO244bhOJQvVveubxg0VmyoSIM Wk2Kz+6eRYf/+t2fvvGCb58xRSjI4kY1OPy8tiOOlxePzfNK6Y4IzStWNw+P4/vrKO9z8NVMwvCcnp1/bFWdtV0 Mq0wjStnFFDX6eCZPQ2n/z6gNVazwuPTXD49n2mFaUG5385+D4/vrMOvNbfMLybPfvaN60fqZGsfL+5rJCPZy q0LS7FZzRe8TW2Hg2f3NpBkNfOhRaWkJkbncKGl18WTu+oxKfjgglKyUoK7sXxjt5OndteTYDZx68JS0pOsYzq ezJQJEYPWTM3jlnnF/HbdSXZWdxodR8Sg2g4HPU4vHl+AEy2hWxzQ2uumpceNP6A53NgTsvs506HGHrSGw4 09BAKa4y19eHwB6jocNHQ56er3crSpF39Ac6QpeJlOtffT5/bh9Pg52dIftOMeberFF9C09Xlo6nYN6zbHWwYWH 3Q7vdR3Rm9H/8rWfpweP/1uP1Xtwfs3HXKypR+3N0Cvy0dN+9gXacigTIgY9d0bZ1CYbuNfHtlNr8trdBwRY8bn pJCXlkhakpVpBaGbvSpIs1GWnUxKopnZJRkhu58zzR+Xic1qZv64TEwmxYzCNNKSrEwusDMxL5XCDBvzxm WSnBDcTBPzUslJTSAz2cqUIP6bzipOJzXRQklmEkXDnNGcXpiO3WahIN3GuKzoXXwwJd9OZrKVnNQEJuS mBv/4BXbSk6zk2hMZn5sy5uNJR38hYtiOqg4+9PvN3DS3mJ/fNtfoOBFNOvoLIUJBOvoLIQBYWJ7FFy6ZxJO7 6316d73RcWKCLwgbvwcC2rDGmUP5z/U4tNb446Sp59meh2A8v+c7vhDnE52Ve0KIYfvCJRN5+0Qb33pqPzOK 0pgkfahGbePxNrZXdTAxL5Xr5xSN6hitvW4e31kHwAcXlJBrTwxmxPN69WATBxt6SE4w4/T6Kc5I4pb5JZhNC gCX18+jO2rpcni5emZBTP+utPUNPA9awwcWFJOTkshTu+up6XCwfFIOi8qzxnT85h4XT+yqw6wUty4MfoG5iE 0yUyZEjLOYTfzmjvkkJZj59AM76ZH6slEbKuo+0dKHd5QzKlXt/bi8flxeP9UhKDw+nyNNvQDsrO5Ea6jrdNLne qerekuPm/Y+D/6A5lhzbHf2r2obKAB3ef1UtTlweP3UdAwUah8JwoKCytaBAnCHJ/zPs4heMigTIg4UpNv4zR3zqe 5w8JVH98oplVFaUJZJSqKZ+WWZWM2je/mcnG8nx55Irj0x7DNRi8qzSEk0c/n0fFITLUwrTCMt6Z0TJoUZNspz kklLsjK7JLYbfE7Kt5NrTyTHnsiUfDupiRZmDhbEzy/LHPPxpxbYyUlNIC8tkYl5wS8wF7EprIX+SqmZwH2AHzg BfAL4KnAjUA3crbU+78d4KfQXYvT+tPEUP3j+EF+5fDJfuHSS0XEiihT6CyFCIZIL/Y9qrZdprVcM/rwQWKO1 Xg7sA24Kcx4h4sonLi7n5nnF/Oy1YzyzRwr/g8HnDxheGO/xBa84XYzcmaeAg0VrPepT5MMRCIT2+PEkmK8BY S30f88smBuYDKwb/HktcAfwWDgzCRFPlFL8+AOzaOx28tXH9pJrT2TZhByjY0Wtuk4HT++ux2o2cduiUjKSw1/ MPbT4oDwnmZvmFqOUCnuGePadZw5wtKmXFZNy+PwlwZl9dnn9PLythm6njytn5jO1IDg7BgxxeHw8vK2WPr ePa2YVMDEvdhd0hFqwXwPCXlOmlLpBKXUAyGNgUDhUUdkNnPVEvlLqXqXUDqXUjtbW1jAlFSI2JVrM/P 7OhZRnp/DpB3ZydLD4W4zcqbZ+vH6Nw+OntsOYrufHmgeev6o2B26ZMQsrh8d3+v/P7pquoB23tddNp8NLQGu Oh2DBRVO3i26nF39AcyKIOwfEo6o2R1BfA8I+KNNaP6u1ngnUAz5g6CNAGtB1jtvcp7VeqLVemJubG56gQsS w9CQrf/3EYpKsZj76p62cbI3tlXahMqMonRx7IsUZSUzIG3s379FYPD4Lu22gOH04exqK4ElOsLBmSh5pNgtXzy wI2nEL021U5KaQkWxl3riMoB13SElmMuU5yWQmW5ITGtsLOkJtelFaUF8Dwl3on6i1dg9+/5/AUeA2rfW1Sqm vA1Va60fPdwwp9BcieI4393L7fVuwmBWP3HsR5TnGDCwigRT6CyFCIZIL/a9SSr2llHoLyAf+DqxXSm0E5gJPh zmPEHFtUr6dBz+1FK9f8+E/bJF+ShHA5fUTjg/LLq9fWqOcx2ieh3A9d2Ph9QekwD+Cyd6XQggONfRwxx+3YD Wb+Mvdi5hZHH+nNCJhpuy1Q80cqO8e044Bw7HtVAdvn2ijIN3GhxaWnu7oLwasPdTM/vpuJuSlcsMwn4cNx1v ZUdVJaVYyH5gfmQsumntcp3eTuHVhCXl2m8GJ4kMkz5QJISLQ9KI0Hvv0RVhNitvv28LbJ9qMjhSXjrcMFI2fb O0L6YzLiZaBGsKmbldI2jlEuxODNZaVrX3Dnk0c+jet7YjcBRfV7Q48vgAeX4Dawd0LRGSRQZkQAhg4lfnk5y6 mOCOJu/+yjScGP1GL8LmoIpuMZCvLJuSEdKZl8fjM00Xe6cnWkN1PtF06+DxcVJGNaZiziEvGD9xmYXnkLri YWminMN1GcUYSU4LcZkMEh5y+FEK8S7fTy2f/vpNNJ9u566Iyvn3d9FFvKRRNIuH0pRAi9sjpSyHEqKUnWf nbJxbzqRXjuX9zNXf8YQvNPS6jYwkRETy+AC6vH39A4/T4Q3J8ty/4x70QrTX9bl/EL1SIdWHt6C+EiA4Ws4lv XTudWSUZ/Ovj+7jyF+v50c2zuGZWodHRhDBMR7+HR7bX4vL6gIHTmmum5jG3NCMox2/pdfHYjjq01tw8v4T ijKSgHHc4XjnYzOHGnhEtbhDBJzNlQohzumFOES98cTllWcl87h+7+PIje+h2eC98QyFiUEOXE5fXT5/LT03HQP uYU23Ba7xc3+nE4wvg9euwF+JXDj6OU639MltmIBmUCSHOqyI3lcc/u4x/vnQSz+xt4NKfr+Op3XXywi3izsS8V

MpzkpmQl8KKSbnk2BNZVJ4VtONPKbAzLiuZ4owkpheFtxD/4gk5ZKcmsHxSdkS284gXUugvhBi2A/XdfOvpA+ yt7eKiimy+d+MMJufHxmbGUugvhAgFKfQXQoTEzOJ0nvzsMn5400wONnRz1S/W840n9slCgCjU6xrYkHok+tw +fAZ0g/f6A/S5R9dPLRDQ9Li8IZvZDdXxvf4A/aN8zCJ6SaG/EGJEzCbFR5eWcc2sQn7zxgke2FLF03vquWd5BZ 9eVYHdJn2vIt2mE21sPdVBjj2RDy8qxTKMlic7qztZf6yVjGQrdywZR6IIPL24nB4//9haTa/LxyVT85gzwqL6Z/c2c KqtnykF9pAsVHliVx11nU5mFqdz+fT8oBzT4fHx4NYa+tw+Lp2az6yS+NthI17JTJkQYlSyUhL4zvXTef3Lq7lieg G/efMEK//rTf6wvhKXN/xL+sXwVbUPFJG39brpdw/vuRraF7XL4Q3rYo8Oh4fewV0HqkdY/K61pnrwsVaFYF9X nz9AXadzIFsQj9/eN/CYtYbqDtmPNp7IoEwIMSbjspP51Yfn8dznlzOrJIP/fPEwq3+6jge31sjGxxHqognZ5NoTWV CWOeyO/ovHZ5GXlsjsknRy7YkhTviOwjQbM4rSyE+zsXiERfVKKVZMziHHnsjKSblBz2Yxm1g+aeD4F0/MCd pxhwr989NsQV1IICKfFPoLIYJqS2U7//XyEXbVdFGencyXLp/M9bOLhr1djVGk0F8IEQpS6C+EMMzSimye+Ow y/nTXQmxWM//88B6u+dUGXj/cLG00BG6fn17X2E5/aq3pcnhGvFBBiEgnhf5CiKBTSnHptHzWTMnjuX0N/Py1 Y3zy/h0sKMvka1dOYWlFttERhQF6XF4e3FqDy+vnyhkFTCscXS+utYdbOFDfTVGGjQ8tLJW+WiJmyEyZECJk TCbFjXOLWfvIVfzo5lnUdzq5/b4t3Pmnreyv6zY6ngiz1143To8frRlTx/qawds2dLnw+mW2TMQOGZQJIULOajZx x5JxrPvaar51zTQO1Hdz/W828sWHdtPW5zY6ngiT8uwUphWmUZyZxIKyzFEfZ+WkHPLTbKyYlEOCRd7GRO yQQn8hRNj1urzct76S379VSXKimf+4fjo3zS029DSUFPoLIUJBCv2FEBHNbrPylSum8MIXl1ORk8KXHtnLp/62g 85+j9HRQqrb4ZUebhHM6w/QMczfQbfPT5cjtn9fRfjJoEwIYZhJ+XYe+8wyvn3tNNYfa+OaX21g26kOo2OFxP66 bv789in+uqlqzKsPRfD5A5qHt9Vw/6Yq3jzSct7rurx+HthczV/ermJ7VWz+vgpjyKBMCGEos0lxz4oKnvzcMhItJm6 /bzN/3FAZc+0z6rsGitOdHj/tfTLDEmlcXj9tg89LXZfzvNftdnpP7zJQ33n+6woxEjIoE0JEhJnF6Tz/xRVcMb2AH75 wmH97an9M7QiwqDyL0qxkZhWnMy4r2eg44j1SEi1cPDGHogwbKyedvzt/nj2R+WWZFGcmcdEEae8igkcK/YU QESUQ0PzstaP89s2TLJuQzX0fW0hqYuhbKkqhvxAiFKTQXwgRtUwmxdeunMrPbp3D1lMdfPSPW8O6AfZo9bt 9EVv47fb5aetz4w9oWnvdUdcJ3+X10y6tU0QckI7+QoiI9IEFJdhtFj7/4G4+/IctPPDJxWSnhm8j7JFo73Pz8PZavP4 AV88sZEqB3ehIp3l8Af6xpYZu58DKT5vVTHlOMjfPKzE62rD0u338fUs1Do+flZNzWFAmG3SL2CUzZUKIiHX FjAL+cNdCKtv6+EgEz5i19Xnw+AJoDY3dkVX47fD46HYO/LtVtfcDA53wo0W304vDM9BGJJpyCzEaMigTQkS 0VZNz+cPHFnKytY9P3r8dpyfy+nxNyE1hRlEaFbkpzB9Dp/pQyEhOYGlFNqVZydyxeBwlmUlcOi3P6FjDVphuY 35ZJuOykmXPVBHz5PSIECLirZiUyy9vn8fnH9zFZ/+xk/vuXBhR2+tYzCaumFFgdIxziuYVgkopVk3ONTqGEGE ROa9qQghxHtfMKuRHN89i3dFWvvPMgbD2MfP4AjT3uAicUSDf0usK6axdS48rJrv/97t9Y97vNBDQNPe48PjG1 iK12+GN2FPiIj7JTJkQImrcvngctZ0OfvvmSaYU2Pn4xeNDfp+BgOaR7TW09XmYVmjnqpmFbKlsZ/PJdpITzNx5 URnJCcF9Kd14vI3tVR3YbRY+urQMm9Uc1OMbpcvh4R9ba/D4Alw6LY/ZJRmjOs7LB5s42tRLjj2Rjy4ZN6o9U2 s7HDy5qx6Am+cVMy5bescJ48lMmRAiqnzl8ilcMT2fHzx/iPXHWkN+fx5/gPbB/RCbe9yDfw4UnDs8/tOd3YOpaf D4vS4f/e7gH98oHf2e07NbQ/+Wo9HUPfDv097nxjfK9h4tvW4CWhPQmpZeWUAgIoMMyoQQUcVkUvzPbXOZ UpDG5x/cRW2HI6T3Z7OaWTMlj7LsZNZMGSiQXzYhh/E5KSwZn0V+mi3o97liUg7lOcksm5AdsW1ARqM8O4 W54zKoyE1h8fjRt7a4ZOrA83Hp1Hys5tG9jc0oSmNaoZ1phXZmFqePOosQwSQd/YUQUam2w8E1v9pARU4Kj3 1m2ZgL/6WjvxAiFKSjvxAi5pVmJfPTD85hb103P37pSFju0x/QNHQ5cfveX4Df7fTS2itd50dLa01jtzMiW54IES5S 6C+EiFpXzSzg7mXl/PntUyypyOLKELeleGF/Iydb+shJTeCjS8tOF5i39rp5eFsNvoDmyhkFTC9KC2mOWPTWsV Z213SRmmjhY8vKSLTExuIGIUZCZsqEEFHtm9dMZXZJOl97bO/pAvBQGZoJ6+j3vmv/yE6H53TBeavs0TgqLY P/tn1un8yWibglgzIhRFRLtJj55e3z8Po1X39iX0j7l102LY/xOSlcNj0PyxkF5hNyU5lbmsGUAjsLI6yjf7RYPTmXit wUVk7OISM5weg4QhhCTl8KIaLe+JwUvnnNVL7zzEH+sbWGjy4tC8n9lGWnUJad8r7LzSbFmqnRs3VRJMpLs3 Hj3GKjYwhhKJkpE0LEhI8uKWP5xBx+9OJhqgc33h6rXpeXhi5jNxhv7nHR5fC873KX109thwOff2xd7cOt2+kd82 lmnz9AbYcjJnc8EPFNBmVCiJhgMin+64OzMZsUX31s77u2RBqNXpeXv22u5pHttWypbA9SypE5UN/Ng1tr+Nv m6nc1ONVa88j2Wh7fWccL+xsNyTYanf0eHthcxUPbathd0znq47x4oInHd9bxyPbasG63JUSoyaBMCBEzijKS+Pfr prO9qpOHt9eO6Vi9Lt/p7vPtfe+fqQqHoZ0E/AH9rj0afQFN1+DPRmUbjW6nF69/YBA1ltztg4spuhzeUXf0FyISSU 2ZECKm3LqghKd21fP/XjrMZdPyyBtlx/2ijCQumpBNe5+HiydmBznl8Cwsy8Tp8ZGcYGFCburpy61mE1fMyOd4 Sx/zSjMMyTYaZdnJLB6fRY/Ty5KK0Xf0v2xaPrtru5iUlzrqjv5CRKKwdvRXSi0B/gfwAzu0119SSn0NuBGoBu7W WnvPdwzp6C+EuJBTbf1c+Yv1XD4tn99+ZP6wbiMd/YUQoRDJHf2rgUu01iuAPKXUCmCN1no5sA+4Kcx5hBA xaHxOCv986SRe2N/I2kPNYz6e0+Onpn14RfW9Li+1HY6IqHXq7PcYvlAhmEbyPIyFxxegpl0WEojwC+ugTGvdp LUeqlb1AbOBdYM/rwWWhjOPECJ2fWpFBVPy7XznmQP0uX2jPo4/oHloWw1P7KrjpQNN572u0+PnH1treHxn HeuOtY76PoOhtdfNA1sGFirsqe0yNEswBM54Hl68wPMwVs/ubeCJXXU8trMupPcjxHsZcjJeKTUbyAG6gJ7Bi7u Bs3ZdVErdq5TaoZTa0dpq7AudECI6JFhM/OiWWTR0u/jV68dHfRxfIECva2BQ13mW1hRncnr9p7vRd/YbW4Df4 3pn14EL5Y4GvoB+53kI8b/t0PG7+j1jXsUrxEiEfVCmlMoCfgN8koFB2dAmcWmDP7+P1vo+rfVCrfXC3NzccM QUQsSABWWZ3L6olD9vPMXx5t5RHSPRYubKmflMzrdz+fT88143KyWBNVPzmFJgZ9VkY1+rKnJSWDI+i+l FaSwuH31RfaRIsJiG/TyM1VUzC5icb+fqWYWYTCqk9yXEmcJd6G8BngW+p7XeqpTKA/6itb5WKfV1oEpr/ej5j iGF/kKIkejo97Dmv9cxvTCNBz+15PQm4u8lhf5CiFCI5EL/W4FFwE+UUuuACcB6pdRGYC7wdJjzCCFiXFZKAl +7cgqbK9t5bt/IGq12OTzUtEdG0b7RAgFNdXs/3c53Fsi39rqpN3AhgdaDmRwDOy+09Lpo63NT1+kIyvEDAU1V 27sf80g0djtp6Rnb7gUivoS1T5nW+iHgofdcvBn4SThzCCHiy4cXj+Ph7TU8u6eBG+YUDes23U4vf99SjdevWTI+i 2UTc0KcMrKtP97K7pouEq0m7rqonB6Xl0e31xHQmsun5zOzOD3smTaeaGNHVSddTg/JVjPegEZrTWqilUum5jF njD3c3jrWyp7agcd897JykhOG/5Z5rLmXF/Y1ohTcPK/4rHumCvFe0jxWCBHzzCbFn+9eRHZK4rBv4/D4Tnef7xrl TEksGZotcnsDOL1+epw+AoMziKOdSQpWpj6XD6vJhNsXwB8IkJpoDcpz1uUcKPh3ewO4vAGSE0aeTWvocY5 +9a+ILzIoE0LEhTz7yDr7F6YnsXJyLu19bpZOMKajfyRZNTkXm9VMQZqNnNREspITWDI+C5fPz4Kysy6cD7k Vk3Kxmk0sm5iNw+3HYllopXD7/EFZ3LB6ch5brR0UZdjIShnBiAyYU5JBn9uHxaSYVmgfcxYRH8Ja6B8MUug vhAgFKfQXQoRCJBf6CyGECIJ+t48TLX2nN003QkuPi9qO4BTVn01Tt2vERfstvS5q2kOXSYhQktOXQggRZQI BzcPba+lxeinLTuaW+SVhz9DQ5eTRHbVoPbBB+KyS4Bb617Q7eHJ3HVoP9A2bVph2wds097h4eFstAa1ZPSW XeeOMOa0qxGjJTJkQQkSZgNb0D24d1WNQkX2f28dQ9UuvK/gZelzeM44/vEL5Xtc7iw+GexshIonMlAkhRJSx mE1cO7uQky19zC7JMCTDpLxULpqQjcvrZ0F58GekphWm0eP04gto5g6ztcWE3BSWTcjG4fWzeHz072Ig4o8My oQQIgpNyE1lQm6qYfevlGJpRehWpZpNasS94ZRSLAlhJiFCTU5fCiGEEEJEgJANypRSRUqpXUop1+Cel0OXf0 ApVXvGzx9RSm1SSj2vlLpwJacQQgghRAwK5UxZB3ApsOU9l38QqAVQSlmBzwArgQeAT4cwjxBCCCFExAr ZoExr7dJad555mVLqWuA1YKixzmRgv9baB6wFloYqjxBCCCFEJAt3of9dwEeBTwz+nAH0DH7fDZx1CY9S6l7 gXoCUlBQWLhxWY1whhBi2qqoqeW0RQoTC/OFeMWyDMqXUJcAmrbVHKTV0cRcwVEeWNvjz+2it7wPuA9 lmSQgRGrLNkhAiFJRSu4Z73XCuvpwJ3KCUehmYoZT6IXAMmKmUMgOX8f76MyGEEEKIuBCymbLBIv6Xg DnAK8C/aa1/Nfh3G7XW3x78/g/ABqATuCNUeYQQQgghIlnIBmVaay8Ds19n+7vlZ3z/AAMrL8UIODw+qtoclG YIYbdZz3k9f0BzvKWXrJQE8uy2MCYUQgghxEhIR/8o9dTuelp63KQlWfnk8vHnvN5bx1rYW9uNxaT42LJy0pP OPYATQgghhHGko3+UcnkHuoq4fX700K6957meL6Dx+gPnvJ4QQgghjCUzZVHq+tmFHGrsYVK+nTNWs77Pq sm5pCZayEtLJCc1MYwJhRBCCDESMiiLUnlpNvLSLlwjlpJoYeXk3DAkEkIIIcRYyOlLIYQQQogIIIMyIYQQQ ogIIKcvhRBCRD1/QLO9qoPtpzo42txLe58HpSAzOYFphXaWVmSzoCzzvDW4QhhNBmVCCCGiVme/hz9tPMV jO2tp7nGjFJRlJZNrT0RrONDQzQv7GwEozUrikxeP58NLxpFoMRucXIj3k0GZEEKIqOP1B7hvfSW/W3eSfo+PS 6bk8e/XFbNiUu77+jF2O7y8fqSZh7bV8N3nDvHHjaf4rw/OZtmEHIPSC3F2MigTQggRVQ419PCVx/ZyuLGHK6 bn89UrpzA5337O66cnW7llfgk3zytm44k2vvPMQe74w1Y+t3oCX71iCiaTnNIUkUEGZUIIIaLGU7vr+MYT+0lPsn LfnQu4YkbBsG+rlGLFpFxe+OJyvv/cIf533UlOtfXzP7fNxWaV05nCeDIoE0IIEfECAc2PXz7CfesrWTI+i99+ZP6o G2InJ1j4f7fMYmJeKj984TAu707+784FUmcmDCctMYQQQkQ0nz/A15/Yx33rK7lzaRl/v2fJmHcoUUpxz4oKfnT zLN482sq/PLyHQODcW9YJEQ4yKBNCCBGxfP4AX3x4N4/vrONfLpvE92+cgdUcvLeuO5aM49vXTuOlA038z9 pjQTuuEKMhpy+FEEJEpEBA8/Un9vHi/ia+fe007llREZL7+eTy8Rxv7uPXb5xgSoGd62YXheR+hLgQmSkTQggR cbTWfP/5Qzy5q54vXz45ZAMyGDiV+YObZjJ/XAbffGI/dZ2OkN2XEOcjgzIhhBAR5771lfx1UxWfXD6eL1wyM eT3l2Ax8cvb56GBrzy6F7/UlwkDyKBMCCFERHn9cDM/fvkI184q5FvXTAvb1kilWcl894YZbD3VwV/ePhWW+ xTiTDIoE0IIETGONvXyxYd2M6Mojf++dU7YG7t+YH4xl07N4+evHaOx2xnW+xZCBmVCCCEiQme/h3v+tp3k RAt/+NhCkhLC3zdMKcV3b5hBQGt+8PyhsN+/iG8yKBNCCGG4QEDzlcf20tzt5r47F1CYnmRYltKsZL5wySRe3 N/E+mOthuUQ8UcGZUIIIQz3x42VvHGkhX+7ZirzxmUaHYd7VoxnXFYy/++lI9JUVoSNDMrigKwiGqC1ln8LIS LQzuoOfvLyUa6aUcBdy8qNjgNAosXMV66YzOHGHp7b12B0HBEnZFAW4w419PCbN07w9y3VeHwBo+MYx uX1c/+mKn775glOtPQaHUcIMaiz38MXHtxNUYaNn3xwdthWWg7H9bOLmFGUxk9fOYrb5zc6jogDMiiLcceae wloTWuvm/Z+t9FxDNPU7aLT4cUf0Bxv7jM6jhCCgdnrf3tqP619bn57x3zSk6xGR3oXk0nx9aumUtfp5Imd9UbHE XFABmUxbm5pBnabhQl5qeTZbUbHMUxxZhJl2cmkJVmZXZphdBwhBPDU7npeOtDEly6fzOySDKPjnNXKST nMKc3g/946ic8fv2cbRHjI3pcxrjwnJaTbk0QLq9nELfNLjI4hhBhU3+XkP545yMKyTD69coLRcc5JKcU/rZ7AvQ/s 5Pl9jdw0r9joSCKGyUyZEEKIsAoENF99dC8Brfn5h+ZiDnOD2JG6bFo+U/Lt/PbNE7ISU4SUDMqEEEKE1V82V bG5sp1/v24647KTjY5zQSaT4nNrJnC8pY+1h5uNjiNimAzKhBBChM2Jll5+8vIRLpuWx22LSo2OM2zXziqkKN3 GXzdVGR1FxDAZlAkhhAgLf0Dz1cf2kZJg5v/dElntLy7EYjZx50XlbDrZztEmaasjQkMGZUIIIcLir5uq2FPbxXdv mEGuPdHoOCN2+6JSEi0m7t9cZXQUEaNkUCaEECLkatod/PcrR7lkah43zCkyOs6oZKYkcNPcYp7aVU+3w2t0H BGDZFAmhBAipLTWfOPJfZhNiv+8eWZUnbZ8r7uWleP0+nlsZ63RUUQMCtmgTClVpJTapZRyKaUsSqnxSqkAndSqlvAnNSqn1SqkHlVLmwet9RCm1SSn1vFlqLVR5hBBCGOOR7bVsOtnON6+ZSmF6ktFxxmR6URrzx2Xw8PZatJb2G CK4QjlT1gFcCmwZ/LkLuF5rvRI4BVyjlLICnwFWAg8Anw5hHjEKLb0u6jodRscIq363j8rWPrzSvVuIMWvqdvGf LxxmaUUWH140zug4QXHbolJOtPSxq6bL6CgixoRsUKa1dmmtO8/4uVNr3TX4ow/wA5OB/VprH7AWWBqqPG LkGrqcPLi1hsd21HGgvtvoOGERCGge2lbDM3saeHF/o9FxhIhqWmu+/fQBvIEAP75lNqYlbxI7XNfOLiI5wcyj2+ UUpgiusNeUKaWKgMuAV4EMoGfwr7qBzHPc5l6l1A6l1I7W1taw5BTQ4/IyNDvf44yPolZfQNPv9gPQHSePWY hQeX5fI2sPN/OVy6dQnpNidJygSU20cN3sQp7f10C/22d0HBFDwjooU0olAvcDnxqcHesChurI0gZ/fh+t9X1a64Va 64W5ubnhiCqAyXl2llRkMbc0g/llZx0vx5wEi4lrZxcwrTCNK2cUGB1HiKjV0e/hu88eZE5JOh+/uNzoOEF326JS+j 1+XtgnM+oieMK9Iff9wP9qrQ8N/nwMmDlY9H8Z79SfiQhgMimWTcgxOkbYTcyzMzHPbnQMIaLaj148TLfTyz8 +tQSLOfYW+s8fl8mE3BQe2VHLh6JoZwIR2UK5+tKqlFoLzAFeUUqtAm4B/lkptU4pdbPW2gv8AdgA3AX8PlR 5hBBChMfmk+08vrOOe1dWMLUgNhfVK6X44IJSdlZ3UtMeX4uhROiEbKZscMB12Xsuft/0g9b6AQZWXgohhI hybp+fbz29n9KsJL5wySSj44TU9XMK+cnLR3huXwP/tGai0XFEDIi9OWUhhBCG+b91lVS29vODG2eSlGA2Ok 5IIWQms6g8k6d310vPMhEUMigTQggRFJWtffz2zRNcP6eI1VPyjI4TFjfMLeZ4Sx9HZJNyEQQyKBNCCDFmQz 3JEq0m/v26aUbHCZtrZxViMSme2dNgdBQRA2RQJoQQYsye2l3PppPt/OtVU8mz24yOEzZZKQmsmJTDc3sbCA TkFKYYGxmUCSGEGJPOfg8/fOEw88ZlcMfi2NhKaSRunFtMfZeTnTWdF76yEOchgzIhhBBj8uOXjtDt9PKjm2f FzFZKI3H59HxsVhPPyilMMUYyKBNCCDFqWyvbeWRHLfesGM+0wtjsSXYhKYkWVk/O45WDTXIKU4yJD MqEEEKMiscX4FtPH6AkM4l/vjS2e5JdyNWzCmjpdbO7Vk5hitGTQVkc8gc0++q6ONnaF9L7ae11s7O6k16XbO

wtRCy6b/1JTrT08YObZpKcEO5d+yLLJVPzSDCbeGl/k9FRRBSTQVkc2nqqndcPt/DsngbqOkOzPYg/oHl8Zx3rj7 XyvGzYK0TMqWl38Os3TnDtrELWxElPsvOx26wsn5TDywebpJGsGDUZlMWhM18vQvXaobUmMHjwgLxAC RFzvvfcQSwmxb9fN93oKBHjqhkF1HU6OdjQY3QUEaXie745Ti0Zn4XNasZus1CalRyS+7CYTXxwQQlVbf1Mj dPiXyFi1WuHmnn9SAvfumYaBenx05PsQi6fno/5KcVLBxqZWZxudBwRhWSmLA5ZzCYWlGUyOf99+8MHVX 6ajSUV2aQnWUN6P0KI8HF6/Hz32YNMzk/l7ovLjY4TUTJTElhakcXLB6SuTIyODMqEEEIM2/+uO0F9l5Pv3zg Tq1neQt7rqhkFnGzt53iz7IUpRk7+RwkhhBiWU239/P6tSm6eV8zSimyj40SkK2cUoBQyWyZGRQZlQgghLkhrzX 88e5BEi4lvXjPV6DgRKy/NxpySDNYeaTE6iohCMigTQghxQa8cbGL9sVa+fMXkuNpwfDQum5bH3touWnpdRk cRUUYGZUIIIc7L4fHx/ecOMa0wjTuXlhkdJ+JdMjUfgDdltkyMkAzKhBBCnNev3zhBQ7eLH9w4A4sU91/QtEI7 Rek21h6WQZkYGfnfJYQQ4pxOtPTxxw2VfHBBCQvLs4yOExWUUlw6LZ+Nx9twef1GxxFRRAZlQgghzmqguP 8ASVYz37haivtH4tJpeTi9fjafbDc6iogiMigTQghxVi/sb+TtE+187cop5KQmGh0nqiytyCY5wczaw81GRxFRRAZl Qggh3qff7eOHzx9mZnEadyyR4v6RslnNrJiUwxtHWmSDcjFsMigTQgjxPr998wRNPS6+d8NMzCZldJyodOm0fB q7XbJBuRg2GZQJIYR4l6q2fv644RS3zC9mQVmm0XGi1iVT81AKXpdVmGKYZFAmhBDiXb7//CESLCYp7h+j nNRE5pZm8PoRqSsTwyODMiGEEKe9caSZN4608M+XTpLO/UFw6dQ89tV109rrNjqKiAIyKBNCCAGA2+fn+8 8dYkJuCnctKzc6TkxYPSUPgPXHWg1OIqKBDMqEEEIA8McNp6hqd/DdG2aQYJG3h2CYXphGTmoC62RQJoZ B/tfFgaZuF/1u3wWv19HvobPfc8Hr9bl9NPeEfqPd4eYWQoxdY7eT37xxgitn5LNiUq7RcWKGyaRYOTmXDcdb8 QekNYY4PxmUxbhNJ9t4aFsND2ypPu8Ap7K1j79truJvm6up7XCc83rdTi/3b6riwa017KzuCEVkADadeCe3wyMD MyFC7T9fOExAa7597XSjo8Sc1VPy6HJ42VvXZXQUEeFkUBbj2voGZr6cHj995xmUtfd70BoCWtN+ntmyHqcXj y8AENLC1da+gWM7PX76XDIoEyKUNp9s5/l9jXxm1QRKs5KNjhNzVkzMwaRg3VE5hSnOz2J0ABFayyfmAJB nTyQ/7dwrqWYVp9PR78GkFNML0855vZLMJBaPz6Kj38NFFTlBzztk+cQclFLk2xPJO09uIcTY+PwBvvfcQYoz kvjs6glGx4lJmSkJzCnN4K2jLXz58slGxxERTAZlMS4rJYEb5hRd8Ho2q5krZxRc8HpKKS6eGLrB2JDs1MRh5RZ CjM3ft1RzpKmX//vofGxWs9FxYtbqyXn84vVjtPe5yZZ9RMU5hOz0pVKqSCm1SynlUkpZBi/7mlJqo1LqH0op6+ BlH1FKbVJKPa+UOvcUTRzwBzRP7qrjV68fZ39dt9FxRARwef08sKWa3755gqq2fqPjiBjT3ufm568dY/nEnGF9K BOjt3pKLlrD+uNyClOcWyhryjqAS4EtAEqpXGCN1no5sA+4aXBg9hlgJfAA8OkQ5ol4PU4v1e0O/AHNwQYZlA lo7HbR1uvG4wtwpEn2zxPB9dNXjuLw+PnuDdNRSva3DKVZxelkpSTwltSVifMI2aBMa+3SWneecdFiYN3g92u BpcBkYL/W2nfGZXErPcnKhLxUEq0mZpdkGB1HRICiDBuF6TaSE8zMKEo3Oo6IIfvqunhkRy0fv7iciXl2o+PEP JNJsXJSDuuPtxGQ1hjiHMJZU5YBDH3U7wYyz3FZ3DKZlNRRiXdJtJi5ffE4o2OIGBMIaL7zzEFyUhP54qWTjI4 TN1ZPyePpPQ3sq+9mbmmG0XFEBApnS4wuYKhmLG3w57Nd9j5KqXuVUjuUUjtaW2XqF+BESy+P76yL2dOc Lb0untxVx9sn2oyOIkTMeWJXHXtqu/jGVVOx26xGx4kbKyfnohSsO9pidBQRocI5KNsOrBr8/jIGas2OATOVUu YzLnsfrfV9WuuFWuuFubnSaRrg9cMt1HY4eP1wC1rH3lT4phPtVLc72HaqIyy7BwgRL7odXn7y8hHmj8vg5nnFR seJK1kpCcwuyeAt2XJJnEMoV19alVJrgTnAK8B4YL1SaiMwF3haa+0F/gBsAO4Cfh+qPNGssrWPVw820dT9zuC kIH2gd1d+WmJMFugOPb7URAtpIfwkf7Chm9cONdPluPD2UkLEgv965Qgd/R5+cNNMTKbYe+2IdKsn57Kntmt YW9qJ+BOymrLBAddl77l4K/CT91zvAQZWXoqz8PkDPL+vEX9A09Dl5O6LxwNw3ewi2vvdZCUnGJwwNJZW ZDMxL5XUREvIeid1OTy8erAZgD63l5vnlYTkfoSIFHtqu3hwWw13LyuXhSMGWTUll1++fpz1x1u5ca7MVIp3u +BMmVLqIqXUb5VS+5RSrUqpGqXUi0qpf1JKyf/qEDObFCmJA2PnM2s/zCZFnt2GxXz+p9Dl9bO1sp3K1r6Q5q xpd7Clsj2o+1TmpCaGtJllosVMonXg38+eKHU1Irb5A5pvPbWfPHuidJU30JySDDKTrdIaQ5zVeWfK1FIvAQ3AM 8B/Ai2AjYFWFmuAZ5RSP9daPxvqoPFKKcXti0pp7HYxbhR70q072srhxh6Ugo9dVE5WSvBn1npdXp7aXU9Aa5 p7XFHz6S8pwcxHFpfR1u+mPDvF6DhChNQDm6s42NDDb+6YJ8X9BjKbFCsm5bL+eCuBgJZTyOJdLnT68k6t9 XuXv/UBuwa/fqaUCv2eO3EuJdHCxLzUUd3WPPgfXqEI1f99kxo4dkC/c3/RIj3ZSnqyvEGJ2NbS4+Jnrx5jxaQcrp1 VaHScuLdqci7P7m3gUGMPM4vlhJN4x3kHZWcZkI3qOsI4qybnkmdPJDs1gYwQ1Z+lJFq4dWEpTT0uphZIE0oh Is0PXjiM2x/gBzfOjMmFQdFm5eSBLgLrjrbIoEy8y3Bqym5TSlUMfj9bKXVCKdWglPpA6OOJsUqwmJhTmkFJ5 shPfY5EQbqNuaUZsqGxEBFm/bFWntvbwGdXTaA8R07TR4JceyIzi9OkNYZ4n+G0xPgaUD/4/Q+AfwYWAP8R qlBCCCHGrt/t45tP7qciN4XPrp5gdBxxhtWT89hV00W302t0FBFBzjsoU0r9B1AM/KtS6nvAcmARcC+QrpT6jlJqZ ehjCiGEGKmfvnKUhm4n//WB2TKLHWFWTcnFH9Cya4l4l/MOyrTW32Og634d4AJe0Vp/d/DyBq3197XW68O QUwghxAjsqOrg/s1V3HVROQvLs4yOI95jXmkGdptFWmOIdxnO6cvPAFcC04CvACilpgMvhDCXiBBa65jcxkm 8XyAgz3OscHn9fP2JfRRnJPG1K6cYHUechcVsYsWkHN461iqvseK0C3b011ofVkp9TGvtPuOyQ0qpptBGE0brc Xl5dHstbl+Am+YVU5yRZHQkESIH6rt5/XAL+WmJfHBByQWbEovI9svXj1PZ2s8Dn1x8uvm0iDyrJufy4v4mjjb3 MrUgzeg4IgIM95X3SaXU6f/ZSqlC4LXQRBKRoqbdQa/Lh8cX4ERLaHcEEMY60tRLQGsau110yD6gUW17VQe /f+skty0sZcWkXKPjiPNYNTkPGGjyLQQMf1D2NPC4UsqslCpnYIPxb4YqlIgMFbkp5KfZSE+yMq1Q+o/Fsrml6a QkmpmQl0p2SqLRccQo9bq8fOmRPZRkJvPv1083Oo64gIJ0G1ML7FJXJk4b1ry21voPSqkEBgZn5cCntdabQphL RIDkBAt3LBlndAwRBhPz7EzMk4F3tPvec4do6HLy2GcuIIVOW0aFVVNy+fPGU/S5ffKciQu2xPjy0BcDe16WA nuApYOXCSGEiAAv7W/k8Z11fH7NRBaUyWrLaLFqci5ev2aTtMYQXPj0pf2Mr1TgKeDEGZcJIYQwWF2ng28 +tZ85Jel84dJJRscRI7CwLIuUBLN09xfAhfe+/F64ggghhBg5t8/PP/1jF36/5pe3z8MqK2ejSoLFxLKJOaw7OtAaQ/Y mjW8XOn15n1Jq5jn+LkUp9Qml1EdCE00IIcSF/OcLh9lb181Pb50je1tGqdVTcqnvcnKytd/oKMJgF6oq/F/gO0qpW cABoJWB2rJJQBrwZ+AfIU0ohBDirJ7ZU8/fNldz78oKrppZYHQcMUorB1uXrDvawsS8VIPTCCNd6PTlHuBDSq lUYCFQCDiBw1rro6GPJ4QQ4mwON/bwzSf3s6g8U7r2R7nSrGQm5Kbw1rFW7llRYXQcYaDhtsToA9aFNkr8ae

hy0unwMLUgDbNpZHUEfW4fla19lGWnkJ5kDVFCESpaa4639GExKSpyw/vJ2Onxc6Klj+LMJLJSEsJ63yI4Wnv d3HP/Duw2C7+5Y77UkcWA1VPyeGBLNU6Pn6QE2Tw+Xsn/ZIO097l5bEcdrx5sZsPxka+6eXp3Pa8fbuGxHbUh SCdC7UB9Dy/sa+SZPQ2cbA3vbgkv7G9k7eFmHtlei9cfCOt9i7Fzef3c+8AOOvo9/OmuReSn2YyOJIJg1eRcPL4A WyrbjY4iDCSDMoP4A5rA4Ca0Xv87m9H2urxUt/dfcHNojy9w+rbn28xWa01Nu4NupzcIqQfsr+viuT31QTueP6Cp auun3+0L2jEjncfvf+d7X3gHRkMDMZ8/cPp3UEQHrTVfe3wfu2u6+J/b5jCzON3oSCJIFo/PwmY1SWuMODei9sF KqRSttSwPCYK8NBvXzCqkvd/N/HGZwMAn4Ae31uDw+JlZnM7l0/PPefsb5hZxpLGXiXmp511C/faJdrZXdZBg MXHXsvIxd4w+VN/N5x/chS+g2VnTyXdvOOvi3BF55WATR5t6sdss3L2sPC42w55bmonWYDGbmFoQ3pZ/V88 s4GBDD+Oykkm0yGmSaKG15scvH+G5vQ18/aopXDWz0OhIIohsVjMXVWSz7mgLMMPoOMIgw3r3U0otU0od Ag4P/jxHKfW/IU0WB6YU2Fk2IQebdeCN0e0L4PAMzKB0XWBT6JzURJZPyqEg/fynLjoHj+PxBXCcMRPV7fT i8vrPdbNzqu5w4BucxWvoco349ufL2Of2vWvW8GxGmzvSmE2KheVZzC3NCHtfoozkBC6emENpVnJY71eMzf+ uO8nv36rkI0vG8dlVE4yOI0Jg9ZQ8qtodVLXJ3Ee8Gu60yf8AVwLPAmit9yqlVoYsVZxKT7Jy+fR86jodLCwPzjY pKyflYjWbyEtLJG+w9uRIUw8vH2giwWLijsXjyEgefrH31bMK2VHdQX2ni69eMTkoGS+fls/O6k7Kc1LOW+B6 qKGHVw8N5P7I4jLSk2WBg4gP92+q4qevHOWmuUX84MaZ0mA0Rq2aPNAaY/3xVuk5F6eGfS5La137nheC6J +uiEAzi9ODWieSnmx9X/+i+k4nWoPbG6C11z2iQRnAt6+djj+gg3aaMS/NxtWzLnwqpqHrjNx97ogdlPn8AcwmJ W+clige31nHfzx7kMun5/PTW+dgGuFKbRE9ynNSKM9OZt3RVj52UbnRcYQBhjsoq1VKLQO0UioB+CKDpzJF 9FlQlkmXw0uqzcL4EX4ac3n9PLythm6nj6tnFTA5P3z1UAvLM+l2erGPIne4HKjvZu3hZnJSE7ltUam0KhBj8vKB Rr7++F4unpjNrz8sWyjFg1WTc3l0Rx0ur/90aYuIH8P9H/4Z4J+AYqAOmDv4s4hCGckJfGBBCVfOKBjxbFdzj4tO h5eA1hxvDm8rh6HcV8woGHFft3A52tSL1gN9pDr6z18XKMT5rD/Wyhce2s3c0gzuu3OhvEHHiVVTcnF6/Wyv6j A6ijDAcJvHtgGyx6WgKCOJ8TkpdPR7mFMqy/Hfa35ZJp0ODwXpNnJTE42OI6LUjqoO7n1gBxPz7Pzl7sWkjHH VtIgeSyuySbCYeOtoKysGt18S8WNY/9OVUuOBLwDlZ95Ga31DaGKJSGU1m7hpXrHRMSLW+JwU2SZFjMm B+m4+/pftFKUn8bdPLI7Y2kkRGskJFpaMz+KtY6182+gwIuyG+/HraeBPwHOAtAAXQogQONnax11/3obdZuGB e5aQa5fZ1ni0anIuP3zhMPVdToozkoyOI8JouAVFLq31r7TWb2qt3xr6CmkyIYSIIw1dTu7841aUgr/fs0TejOPY6ik Dpy3fOird/ePNcAdlv1RK/YdS6iKl1Pyhr5AmE0KIONHe5+ajf9pKr8vHXz++OOyb1IvIMiE3leKMpMHu/iKeDPf 05SzgTuAS3jl9qQd/FkIIMUq9Li93/WUb9Z1OHvjkEtnPUqCUYs3UXJ7cVS+tMeLMcGfKbgYqtNartNZrBr8iek BW2drH+mOtQd2IO9J1O7ysP9YqW3QIESV8/gCf+8cujjT28ruPzmfx+ODs5CGi32XT8nF4/GyubDc6igij4Q7K9g IZY70zpVSyUuoFpdQ6pdQzSqlEpdTXlFIblVL/UEoFZZlRn9vHc3sb2VndydpDzcE4ZFR45WATO6s7eXZvA06P bLggRKT7/vOH2HC8jf+8eSaXTM03Oo6IIEsrsklOMMfVe5gY/qAsHziilHpFKfXs0Nco7u8qYKvWejWwDbgdW KO1Xg7sA24axTHfx2JSWMwDzUUTrfHTAXvosVrMClP8PGwhotL9m6r42+Zq7l1ZwW2LxhkdR0QYm9XMyk m5vH64Ba210XFEmAy3puw/gnR/J4EFg99nDN7/usGf1wJ3AI+N9U5sVjO3LyqlqcfFpLzwbQNktKtmFnCipY/C9 CQSLVKDIESk2nyyne89d5DLpuXzr1dNNTqOiFCXTc/n5YNNHGzokVrDODGs+ZTB9hdHAPvg1+FRtsQ4Dix RSh0EFgIngJ7Bv+sGMs92I6XUvUqpHUqpHa2tw1sinJ2ayIyidBIs0T9I5PD4ONnah8d3/hZxiRYzM4rSyUoZ2QbjI jb0uQd+T7x+aSUYyVp73Xzx4d2UZ6fwi9vnRuyWYcJ4a6bkohS8Jqcw48awRixKqQ8xcLrxVuBDwFal1AdHcX9 3Aa9orWcALzAwU5Y2+HdpQNfZbqS1vk9rvVBrvTA3N762ndBa8/C2Wp7d08BzexuMjiMilD+geXhbDc/uaeDF/ Y1GxxHn4A9ovvTIHnqcXn77kfmkyvZJ4jyyUxNZMC6T14/IoCxeDHca6VvAIq31XVrrjwGLgX8fxf0pYGiX1TY Gtm1aNfjzZcCWURwzajk9ftr73Oe9jj+g6XP7AOhxvbOS1OMLcKixG5fHF7Q8hxq7OdrUc+ErnoXb56e1992Pxes P0NLrIhB4px7CH9C09LrwBXE2p9vppdf17lW2fW4f3Y7Rrbw9W+5I5wsE6HcPLO7oiaMVx9Hm/946ycYTbXzv hhlMK0y78A1E3Lt0Wj4H6nto7HYaHUWEwXA/ppm01md2sWtn+AO6Mz0IPKKUuhPwArcBn1JKbQRqgF+M4 phRqc/t4+9bqnF6/Kyaksv8cWc9c4vFbOLa2YWcaOljdsk7NQU/fOEQR5t6GZeVzE9vnTPmPE/tquNnrx1DAd+6dj pXzSwY9m09vgB/31JDj9PLwvJMVkzKRWvNoztqaelxM7XAztWzCgF4fl8Dla39FGcm8aGFpWPOXdXWzzN7 GlAKPrighKKMJFp6XDy6oxZfQHPd7CIm5g2/EafWmke219La62ZaYdqI/h2MlGgxc+3sAk629jOvNMPoOOIsjjb 18ou1x7h2ViG3LRr7776ID5dPz+MnLx9h7eEW7lxaZnQcEWLDHZS9rJR6BXho8OfbgJdGemda6y7gyvdc/JPBr5j h9QdQDAyozqWz33O6bUVjlwsGF19prXH7Au9qFjghN5UJ7+nwXdPhAKCu00kgEMB0nuWWbp8fq8mE6Ty1K /vru9Fao4H99V0jGoz0u32nZ2cau10AeP369MzZ0GUADV0D3zd1u9Bao9TY6mmaelwEtAYNLb3ugUFZrxuvX5 ++n5EMyjz+AG19Q7nf/ck00ps4TsyzMzGOFrZEE68/wFcf20uazcr3b5wx5t97ET8m5KZSlp3M64ebZVAWB4Y1 KNNaf00pdQuwnIFTkPdprZ8KabIo1dzj4vGddadnbvLstrNeryQziXnjMujo97C0YqBhpM8f4LGddTR1u1gxKYeF5 eduJPnRJeN49VAzF0/MOe+A7EB9N2sPN5OVksBti0rPuSrzk8srqGzrx2JSfOLi8SN4xJCZksBFE7Kp63Ry8cRsA BIsJtZMyeNYcy8Lyt6ZBbx0Wh57a7uYXpQWlDemOSUZtPW5MSvFtMKBAcmUAjt1nU48/gBzx2WM6HiJFjO rp+RxvLn3Xf/+rxxs4lBDz7tm/YQYrvvWV7K/vpvffWQ+2amyybgYPqUUl03L54HN1fS7faRIHWJMG9azq5QaD 7yotX5y8OckpVS51roqlOGiUXW74/QqydoOxzkHZUopVk/Je9dlvS4fTYOzSida+s47KLtsegGXTb/wbNaJlj60hv Y+Dx39HgrTz77JcXFmEn/9+OILHu9cllZkv++yOaUZzHnPqbTJ+XYm5wdvNicpwcx1s4vedZnVbBrTace5pRnM fU/uEy19ABxv6ePqUR9ZxKPaDge/ev0418wqkAG9GJXLpuXzp42n2HC8latmyu9QLBtuXdhjvLPnJYCfIPQTi0V TC+0UpNsoyrAxpWBkhbxpNgtOj58D9d1kJgenrUWOPYGDDd10OjxkB+mY8WhpRTbpSVYumvD+wacQ5/OfL xzGpBTfvna60VFElFpYnkmazcJrh2SD8lg33HlQi9baM/SD1tqjlJJ3+LNIs1n58OLRdefucflISjAzszidTofnwjcYhrZ eDzOKBhYItDvOPVMmzm9BWea7TsMKMRzrj7Xy8sEmvnblFloy5P+eGB2r2cQlU/N4/UgzXn8A63nqlUV0G+ 4z26qUumHoB6XUjQy0tBBBlJZkpSw7GbNJMb0oOMvls1PPmCmTprJxZ29tFw9sqWZvbZfRUeKOzx/g+88foiw

7mXtWjKxOU4j3umpmIV0OL1srOy58ZRG1hjtT9hngH0qp3zBQ6F8LfCxkqeKU2aS4ZX5JUFY1Duno9zC9cKCo vv08NWUiNm043orXr11/rPV99X0itJ7aXc+Jlj5+95H5su2ZGLNVk3NJspp56UAjyyflGB1HhMhwt1k6qbVeCkwH pmutl2mtT4Q22tjsrunkmT31NPe4LnzlCBPM5fJl2SkopUhPso64Ts0f0Lx5pIWXDzTiCGKTWhE+5Tkp7/pThIfb5 +cXa48zuyQ9anrdiciWlGDmkql5vHKwCX8UNbYWlzPc1ZeJwAcY6MBvGRo0aK2/H7JkY9Dj8rLu6MAemS6v n9sWja7GKxbMLc1gYl4qNovpvH3TzuZYcy97Bk97pSZa5dNZFLp2ViG9k33YZRl9WD20tYb6Lic//sAs6Ukmgu aqmQW8sL+RHVUdLDnLincR/Yb7Lv0McCPgA/rP+IpINosZu23gTSjnjJ5AtR0O3jzSQkuUzZ7VtA/m7h157j63jx 1VHRxr7hvxbbNSEk5vlpxjj556NK01O6o62HSiLe4351ZKkWazysAgjJweP7958yQXVWSzfKJ8kBHBs2ZqHgk WEy8daDI6igiR4X58LtFaXxXSJEGUYDHxkSVldDo8FKYP9AnzBzTP7KnH69dUt/dz9wgbpBrF5w/w7N6B3DU dDu5aVj6i2791tJVjzb0A5KUlvmuQeiH5aTY+dlEZXr8m1x49DS+PNfex4fjAOhSTSZ21h5oQofLYzlra+tz89o55 MhgWQZWaaGHlpFxeOdjEd66bft5dWkR0Gu5M2Sal1KyQJgmypAQzRRlJp18UTYrTW+QkJYSu6FZrzcGGbg4 19KD12M/7m5Qadm6Hx8fO6o7TDWgBkgdvYzEpEiwjX0adkZwQVQMyeOcxv/d7IULN6w/w+7cqWVCWyeLx 527+LMRoXTOrgMZuF3vruoyOIkJguDNly4G7lVKnADcDKzC11np2yJIFmVKK2xaVUtfpZHwIi54PNvTw2qH mwfuEaYVja21hMik+tKiU+mHkfvlAE9XtDqxmxT0rKrBZzaycnEtxZhKZyQmk2axjyhItSrOSuXVhCR5fgIrc4e9 7KcRYPbe3gfoup+xvKULm0mn5WM2Klw40MW+c9E6MNcMdlMXEzjJ2m5VphaEdmJw5ORaEiTJgoCFt2jByn 21Bjtmkgrqt0Uj0uX00dDkpy04+3RLA5fVT0+GgJDOJ5ISBXz+vP0BVWz/56bZRDRxrOxyYTIriM5pzlmQmjzp3 r8tLU7eLcWfkjnX+gOZUWx/ZKYlkSj+7UQkENL9bd5Ip+XbWvGcLNSGCJT3JyrIJObx0oJFvXj1VBv8x5ryDM qXU0Px7bxiyxISZxQMzYwOzZOEdDF09s4CDDT2UZCadPuVplEBA8/C2GnpdPsZlJfOBBSUAPLlroE1JVkrC6 fq4lw40cbKlj+QEM59YPn5E3aqPNPXw0v6Boteb5hWPeRbUH9A8sr2WXpePsuxkbplfMqbjRYt1R1vYV9dNgsX E3cvKZdPjUdhwoo3jLX38/ENzpNZHhNTVMwv4xpP7OdjQw8zidKPjiCC60CvvTkAzcLryvTRQEfREEaylx0Wq zXJ6hudslFLMKjHmP0lygplxWclkJBt/mtKvNU6PH4D+M3qc9bsHvu9z+043yR26zOUN4PNrRjKe7HO9/9hjyh3 QOIZyB+F40aJv8LF6fAE8vgAp0VVGGBH+tqmKnNRErptdZHQUEeOunFHAt58+wLN7G2RQFmPOOyjTWkf HEsUw2Haqg7dPtJGUYObOpWUROZPw+uEW9td3k5Fs5c6lZSPuSxZMVrOJ6+YUcbKlj9lnDFKvnV3IoYYeph TYT0+7XzE9nz21XZRlJ494Ecac0gzcvgAmpcZcvwcDK3evm11IZWs/s0vj58Vu9ZQ87LYO8tNscvpyFGraHbxxtI UvrJk4qgU1QoxEZkoCqybn8tzeBr5x1VSZmY0hkTeyiFBNg73NnB4/PS5vRA7Kmgf7mHU5vLh9AUMHZQDjc1 LedzqxMN1GRrKVpDOmw7JTE7loQja2UdRvWc0mLg5yL6iK3NS4WyCQnmTlkqn5RseIWg9sqcKsFB9ZWmZ 0FBEnbphbxOtHWtgujWRjSuSNLCLUxROy0VqTm5oYsftHrp6Sx7ZT7ZRlp0TkoBHglYPNHG7sYWJeKtfPGTj N8/aJNrad6qAow8atC0rlU5+IKk6Pn0e213LlzALy02xGxxFx4rJp+disJp7d2yCDshgi8+zDlJ2ayI1zi1kWwR26izOS uHleCfMjeJl0ZdvAzgKVrf2n+7hVtg5c1tDlwuH1G5ZNiNF4Zk89PS4fd11UbnQUEUdSEi1cPr2AF/c3xv3OJbHkv IMypVTW+b7CFVLEjosn5JCVksDySdmna8qWVmSTlZLAgrJMUiN0hk+Ic3lkRy2T81NZVB65H4ZEbLphThGd Di8bB3cwEdFPVl+KsJpTmsGc0ox3XTYp384kg3qpCTEWJ1p62V3TxbeumSb9okTYrZqcS3qSlWf3NrBmqvTGi wWy+tIg/oDmlYNNtPd7uGxaXsTWqY3FppNtHG/uY1F5FtOLxr4yMhrUdjhYd7SFvDQbV0zPRylFS6+LVw82k 5Zk5eqZBSPqwyYi22M76rCYFDfNKzY6iohDCRYTV88s4Lm9DTg9/pBuISjCY9jvDkqpTKXUYqXUyqGvUAa LdQ1dTo429dLW62ZndafRcYLO4wuwtbKDjn4PmyvbjY4TNttOddDW5+FQQw+tvW4Adtd00drr5mRLHzUdDo MTimDx+gM8saueNVPzom5/WBE7bphTRL/Hz9rDzUZHEUEwrEGZUuoeYD3wCvC9wT+/G7pYsS8nNZG0JCt KEdK9OI1iNStKswa2OqrIjb3Hdy5DjzUrJYH0wSa+FTkpmJQiNdEiq/NiyFtHW2nrc/OhhaVGRxFxbElFNoXpNp 7YVWd0FBEEw62q/mdgEbBFa71GKTWVgcGZGKWkBDN3LyvH6w8YviVSKCil+MD8Ypxe/3l3QIg188ZlMr UgjQSLCfNga49J+XY+nZWMxaQM7x0nguexnbXkpCawekqu0VFEHDObFLfML+Z3607S3OOSD35RbrjvEC6tt QtAKZWotT4CTAldrJHZV9fFm0daTm8VA3C0qZfXDzfT2e8xMNn5mU0qJgdkQ5RSMT0ga+tzs/ZQMyda+t51e VKC+fSAbIjNao67Admumk7ePNpyerutWNLe5+b1wy3cMr9EagSF4T64oJSAhqd21xsdRYzRcF9N6pRSGcDTw GtKqWeAhlCFGommbhevH25hT20XG4+3AtDr8vLSgUb21XXLeXYRMq8ebGZ/fTcv7GvEJf3V3qW2w8FbR1v ZU9PF5srYW67/4oEmfAHNzVLgLyLA+JwUFpZl8tiO2tP9H0V0GtagTGt9s9a6S2v9XeDfgT8BN4Uw17AlWc1Y BmclhrrYW80mXF4/9V1OrGZZpm6Ulh4X20510O30Gh0lJFISB2Y5kxJM75sZi3dJCWZMgy0iUmJwtvS5PQ1M yktlaoG0chGR4YMLSjjZ2s+e2i6jo4gxGNarpVJq3Bk/nhr8swCoCXqiEUpPtvLhJePodnoZnz1QZB3QGrNSJFnNS KNjY/j8AR7bWYfHF+BESx93LBl34RtFmatnFlLd3k9+uk1OYb1HTmoiH15SSp/LF3MLWRq7nWyr6uArl0+W3 mQiYlw7u5DvPneQx3bWMS+Cd3UR5zfcd5IXgOcH/3wdqAReClWokcpJTWRCburpPRMVCs3A4OysbW9FyC mlTs+UxOr7VoLFxKR8O2k2a0jvp7nHxfHm3qg7LZFnt1GRm3rBgUu3w8vhxh7cvndOAbf0uHhxXyMtPa7z3tbn D3C0qZf2PndQMg/H83sbAU7v3SpEJLDbrFw9s5Dn9jZIOUUUG9ZMmdZ61pk/K6XmA58OSaIgUAo0Gq8/gEl GZYYwmxQfXFBCTUc/k6Vb/6i19bl5eFstAa1ZUpHFsgmRu/fqaHj9AR7eXoPD46c8J5mb55UA8N3nDtLe5+HF A4385o7557z9G0daONjQQ4LFxN3Lyk+XMITSc/samF2STnmMzQCK6PfBBSU8tbueVw81c4N8aIhKozrnorXex UCLjIjkC2hsFguF6Un4LzC74PMHqO9yvutT+rk0djvftcJzJLodXlp6z/+pP9bk2hNZUJaFPcQzSUbx+QLsrO6kyxG 6Fb4ur39gxhdichWjP6Bx+wZqDJyed2oNHIOP1eHxveu69V3Od80COAe/9/oD+Pyhn0msautnX10318+WNzwReS 6qyKY4I4mHtxleWSRGabg1ZV8+48f/3959x8d1loke/z0zGmnUJavLkiVb7nZclRDHaSbFISRxOkkgENiQhWUvC yxl4d67Nwu7dy+wsMDCAoFdQguBEEIqjklI7MROXGPLdtxjW5Jl9a5RmfLeP87Ilh2rz8yZGT3fz0cfj47PzHnO0 ZRn3vK8DmAF0BSWiEIgLSmB9y8ppKatlxWlI/etP1tVx4lmD3npSXzo0rJh93vjWAtvvtOC2+Xk/lVl41o4u7Gzj8 e31+APGNYuKpwySw7Fu29uOMjumg6yU1x8794VJCaEflxZSXYK1yzIp6PXy8Xl00L++HZzu5zcvLSYEy09LC

3JOrP9766Zw6YjTVw992wNsPX76jnc0EV2iosPryrH4RDWzM8nM7mN4qzkM8V6w+nZPdak8/cvKQr7sZQaL4 dDuPeSUv5tw2GON/fE3XjOqWCsmcXQ/icf1tiyJ0MfTujMzk9ndv653WaBgKF7wEd6UsKZcS7NXVYrR2vPAIG AOTMu7XzNwTErfV4/3X2+cSVlrZ4B/AFzzuOo2GKMoav/3OdOXYfV8tne66W7z8e0tMSwHHvJkGQlHs3MT X3Xh8fyGdnvGqw8+Npp7/XiDQRIcjjJcLu4el7kFmJ+tqqOS8qnUZwVf2vVqvhwd2Up33npCL/ZVs1Xblxgdzhqn MY6pixk1ftF5MPARwAn8EHgPmAdcBJ4wBgTtvoJf3jrFDWtHhYVZ3D9okIArltYwJ7aduYXZgybkAFcPtsay5O XnkRh5vgqJs/JT+f0jD76vX4qy3VWTCx6Ya/VSjMrL5V1y6zaVA9cNpOnd59ieWl22BIyddY1C/LZebKNirw0khI iX3T5SEMXhxu6+eq6RRE/tlJjlZ/h5rqFBTyxo4bPXTc3rguUx6Oxdl/OBT4PlA+9jzHmveM5mIhMB64yxlwT/D0P WGOMuVxEvoRV++yJ8TzmWPn8AWqCi0GfbDm7KHR5buqYBuxmpyZecLaVMWbU2WVOh7BmmG/zY7n/Z EXiGJMVbTGeH8+Jlh7AnPPcWVmWzcoyTbIjpSQ7hZLsFNuO/+L+ekTghuAXOqWi1QffU8af9tWzfl89t2qB45g y1j64J4AfAT8FJjPaeC3gFJGXgbeB9cCrwf97CavVLCxJWYLTwRVzcjlQ38XKENVwqWn18MyeOllTndxVWTq umV/9Pj+/31lLW88ANywuYnZ+WkhiOt+h+i427K8nLz2JO1ZG55IwO0+28fqRZspzU7hlabHtydnzVac52tjNpbO m8Z5ZOQCkJDrZcbKV1RU5tsam7LN+fz3LS7PI17UFVZS7rCKH8pwUHttarUlZjBnrJ7TPGPNDY8w2Y8zOwZ 8JHK8ASAy2lHmALKAz+H8dwAWzJRF5SER2iMiOpqaJzy+oLJ/G/ZeWhWyg/ZHGLgZ8Ado8Xmrbesd138bOf ho7+/H6DQfrO0e/wwQdON2JL2A43dEXtePZ3q7rIGAM7zT10GPzDMM+r5/DDV0EjGF/3dm/i2fAz5LpWXgGt BrxVFTT6mHfqU5uWKytZCr6WQP+Z7DtRCuHG7rsDkeNw1iTsmdF5G9EpEhEpg3+TOB4HcDG4O2/YHWHD mZIGUD7he5kjHnEGFNpjKnMy8s75//8AcOze+p4dPNxats8F7p72CwoyiDdnUBBhpsZ08bXrVKQ4aYkO5mUR CeLizPDFCFcVJJJcqKTspwU8tKSwnacyVhamkWSy8GConRSE+0d/+B2OVk8PZMkl4NlM7LObF8ejHH5kG3Kf sebe/jZ5uOs31cf1uK6G9621tBdq12XKkbcubKERKeDx7ZqeYxYMtb+to8E//3CkG0GmDXO420BPh68vQyoAT4 AfAO4FnhznI9HXXsvRxu7AdhV3R7RMSdFmck8eMV4L4ElMcHBXZWlIY7o3Sry0qi4Kjxdo6GypCQrqmYYXr ewgOsWFpyz7bLZuVw2O74Kt8aDHSdaafd4afd4WVmWTV56eL54vLivnvmF6ZTlaIkBFRty0pK48aJCntxZy+fX zhtXxQBln7EuSD7zAj/jzkaMMbuBXhF5Fav47OPAJhF5HStJ++N4HzMvPYnsFBcOEWbnRXfyEc16B/w8V1XH C3tPj6mQrrqwxs4+ntxZy+ajzXaHMiXMzk9DxHofyApTnbKmrn62n2zVVjIVcx5YPZOufh+/31FjdyhqjMacOovIZ bx79uUvxntAY8znz9v09eDPhLhdTj68qhxfwISleOdUsfdUB0carBbHoky3Lmg7QZuPNVPd6qG61cOcgjTy03VQe Dgtn5HNwuIMEp2OsE0QeelAA8ag48lUzFlWmsWKGVk8uuXEmYLLKrqNKYsRkV8C/wZcjtXCdTFQGca4xsX hkJAmZM3d/by4v54Dp8M3AD/aFGQk4RDB6RAKdHbZhBVmWEVF05ISwrpQ+b5THWzYXx/WJZ5iRVKCM 6wzdl/cX09ZTgrzC3UNVxV7Prp6JidaPLxyqNHuUNQYjLWlrBJYaMI5kjaKvHyggbr2Pg6c7mTGtJSILHJst7KcV B5YXY5DiNu1KiNhVUUOcwrSSEtKCFvRxraeAf4cHHjuGfDrlPcw6uzzsvloMx9dPdP2Ui1KTcQNiwspzHDzs80 nuGZBweh3ULYaa/PSPmDKtN0PtnAku5xRWdcrXDKTXZqQhUBuWlJYq2gnuRwkuaznZUZy/H9hsNMrBxvx+ o2OJ1Mxy+V0cP+qMl4/2qzlMWLAWN/Rc4G3RWQbcKbYlTHmlrBEZbPrFhYwtzCd/PQkHaemok5KYgIfvKS Mlp5+ynU2YFgda+qxxliWZtkdilITdt8lM/jey0f42ebj/OvtS+wOR41grEnZw+EMItokOB1U6ExOFcUyU1xkhmm2 oTrrc9fN5W+urtAB0iqmZacmcvuK6fxh1ym+uHY+2am6Vm+0GmtJjI1DfwAfcHd4Q1NKKfvpgs4qHnx09Uz6fQ F+8cZJu0NRIxhz35yILBORb4jICeCfgQNhiyqCvP4A1S0e+rwj1+YyxlDb5qGzzzuh47T2DFDf0Teh+8aqPq+f6hY PPv/ISxP5A4aaVg+eAV9Y4+nweDnVfu5yWN39PmpaPaNWgx/wWc+ToTXc/AFDdYuHXpuXhhrJheK+EGOsv0 HXBJ/fkeALvlaj+XorFa3mFqRzzfx8fv7GCX0NRbERuy9FZC5wD3Av0AL8FhBjzJoIxBYRz+yuo7rVQ25aIvevK h92vzeOtbD1eCuJCQ4+cln5uKojN3T28fi2GgLGcP2iAhaFcVmlaBEIGB7fVk2bx8usvFTWLRt+huCG/fUcrO8i3Z 3AA5eVkxCGyRUdHi+/fPMEXr/hijm5VJZPo8/r59dvnsQz4GfZjCzWzMsf9v5/fOsUp9p7Kcx0c+8lMwCrVMKh+ i4ykl18ZFVZWOKerD/squV0Rx9FmW7uCcZ9IZuONLPrZBvJiU4+sqqcZJuXu7qQP+2r52hjN5nJLh64TGsuKTV en7i6grt+9Aa/21HDRy4rtzscdQGjfYocBK4BbjbGXG6M+Q8grlLs1h6rzlObx0sgMHxrSWuwHtSAL0B33/hadNo9 XgLBlpjB48U7vzF09FrXabRzHry23f0+BkZpVZuozj4vXr/1N2gJxtM74McT/MbY2j1yjIP3GXoug9u6hjx2tBm8ti2 j/Q16rPk71jUJb4vlRLWeud4+vAFdGF6p8bq4fBory7L5yWvvjNqDoewxWnPPHVgtZa+IyHqsZZHi6uvp2kWFVJ 1qZ35h+ojfvC+fnYtThNz0JAozx1dcdU5+GpXl2fQO+Kksm8g67rHH5XRww+JCjjR2sWyUmWvXzC9g58k2ynJ SSEkMT4mHkuxkVlXk0O4ZYFVFDmANfr16Xh6nO/q4ZObIf5f3LS5kX10HC4oyzmy7dkE+u062U56bEpUtSw DvW1zE/roOFg6J+0KumptPUkILhZlucqJ04frrFhawq7qNirw0khKi83orFe0+cVUFH//FDp7fe3rEHgxlDxlLPVgRS QVuxerGfC/wc+ApY8yGsEZ3AZWVlWbHjh2RPqxSKs5VVlai7y0q3gUChrXf2USC08ELn75ciyJHgIjsNMaMaR Wksc6+7DHG/NoYcxNQAuwG/mHiISqllFIq0hwO4aErZ3HgdCcbDzfZHY46z7hHJhtjWo0xPzbGvDccASmllFIqf NYtm05RppsfvnrM7lDUeaJvuphSSimlwiYxwcGDV8xi6/FWtp9otTscNYQmZUoppdQUc98lM8hNS+K7Lx2xOx Q1hCZlSqkLOtbUzZeerOKbLx5iwKfT55WKJ8mJTv76ylm8frSZnSe1tSxaaFKmlLqg3++o5URzDztOtLL1eIvd4Si lQuyDl84gJzWR77581O5QVJAmZUqd51B9Fy/sPc3pjt7Rd45j84vSAesb9ay8VJujmZw+r5+XDzTw2pEm/CMUi VZqKklJTODjV85i0+Em3qpuszscxejFY5WaUgZ8AdbvqydgDE1d/VN6KZJ1y6azoiyb9KQEslIS7Q5nUnZVt1FV 2wFATmoSC4tHLqar1FRx/6VIPLLpHb778hEe/egldocz5cVFS9mh+i62HGs+Z1HxE809bD7aHNULLEdC74CfLU ebOdLQZXcoYeEPGHacaOWt6rZRFxUfiwSHkO62vqtkpbgm/XixrjQ7ZcIJ2dt1nbxxrGXUxdAno6ffx+ajzRxr6h5 xv+zgOYhApv5dlTojNSmBB6+YyauHmthd0253OFNezLeUNXb18cLe04C1Jt7aRYX09Pt4encdAWOo7+jjjpUlN kdpn1cPNXKwvgsRuD81MWqX0JmoPbXtvHakGYCkBOekW0AcDuHeS2bQ0NlHSXZyKEKckmrbPLy4vx6Afp +fq0dY7H0yXj7YyLHGbkTgo6tnkpl84YRrQVEGWSkuXE4HuXH2GlBqsj68qpxHNr3D914+wn8/cLHd4UxpMd 9S5nI4cASXiUhMsE7H6RASnOdum6oGz98hQoIj/q5FovPsOSUmhGa5kOREJ+W5qSQ4o+N61bR62F/XEVNjoR

KdDgZXb0kM0XXsHfBTVdt+zqLwicHXuVME5whr1wIUZSZrQqbUBaQlJfDxK2bxl4ON7NKxZbaK+Zay7NRE 7qosoc0zwLwCa2Cy2+Xk7spSGjr7mFOQZnOE9rpqbh4FGW5y05Listtm8fRMkhIcOB3CrLz4+1s3dvXx5K5ajIG2 Hi+Xz8m1O6Qxyc9wc8eKErr6fMwvTA/JYz5bVceptl7cLicfv2ImCU4H751fQEl2CvnpSaQlxfzbmVK2eeCycn62+ Tj/9uIhHvv4pXaHM2VFR1PAJBVnJbOoOPOclg23y0FaUkJUtw61ewaobvGEZCzUcBKcDhZPz6Qw0x22Y9htTk F63CRkjV191LWfnfXp8xsGnx7eQGzVCiudlsLC4gwco7RgjZXPb10IfyDA4CsmMcF6fudnjP78rmvvpamrPySxK BVvUpMS+NSa2Ww51sLmo812hzNlxeVXy36fn8e2VuMZ8LOwOIO1iwrtDuldOjxefvXmSbx+w6WzclhVkWN3 SMpmtW0efr/TahW7YXEhC4oyKM5K5n0XFdLu8bJ8RpbdIdrqxosK2V/XSVlOCq5xdom+XdfJi/vrEYG7KkuZnq XjBZU6333vmcFPNr3DN148xB8rchAJzRcqNXbR24w0CQO+AL3BmZgdnuicfdkz4MMb/Obf0Tswyt5qKujo9Z5 pFWsf8rydX5jBpbNySEpw2hRZdMhKSWT17FxKslPGfd+OXut6GhO97wlK2S0pwclnrp3Lnpp2NrzdYHc4U1Jct pSlu11cu6CA2jYPleXT7A7ngoqzkrlybi4t3QPaSqYAK/lq6/Hi9QdYUZZldzhxZfmMLPq8flxOR8jGuCkVj25fMZ0 fbTzGtzYc4toFBaNOoFGhFZdJGVgDwBdPz7Q7jBGtLIvOhFGFns8fwCEy4vgqp0NiZiB/tPP5Azgdcqb7xe1ysmZ +eMpyKBVPEpwOPnf9XP72sbd4Zs8pbls+dUtK2SEuuy+ViibHmrr5z1eP8bMtJ+jp99kdTtw72thlXe/NJ+gdCF/h WqXi1Y2Li1hYlMG3/3yYAV9sTTCKdZqUKRVmRxu78QcMnb1eTnf02R1O3DvSYF3vjl4v9Z16vZUaL4dD+M IN86hp7eXXW0/aHc6UoklZHNl3qoMfbTzG+n31YS2zEe9ePtDAjzYem9CSI0cbu3lk0zGe3n3qTLHXpSVZZKe4 mDEthRnTxj9IPRKONnbx443nxh2rlpZmkZXiojw3RVdlUGqCrp6bx+rZOXz35SNnJsqo8NOkLI68VdNO74CfA6 c76dZusgnp8/qpqu2gd8DPrpPjr2y9p6adnn4/7zT10Nxt1cQqzHTzwOqZ3LGyJGpXmNhd04FnwIq7pSe2a3kVZyX z0dUzuW15ybhLZyilLCLCV25cQEevlx+8ctTucKYMfceKIwuL0hGB8twUUhOHn8PR1jPA73bU8MLe0/j8O15g KLfLyez8NESs9RJHcqypm8e2VvPmOy1nts0vSschQmGmm2mpE1vI2w7zC624izLdTJvgAuRKqfiyqDiTO1aU8O jmE9S0euwOZ0qI29mXU9HKsmksL80etYL6ruo2TrVZVePn5Kcxp0BLBAx189JiAgEz6nXcfLSZlu4BGjr7WFK SSUpiAouKM1lQGLoq9pGyeHomC4tiL26lVHh9/vp5PFdVx9fXH+T7962wO5y4py1lcWYsH6ql01lQsVqF8tPjd/ mlyRjrdQQoyHDjHlLYNVYTm1iNWykVPoWZbh66YhbPVZ3WxcojwJaWMhH5HHC7MeZyEfkCsA44CTxgjN ERhWE2tyCd4qxkXE6Z8lXiJ2PNvHxWzMgmLSlBExqlVNz666sq+M32Gv75ubd58pOX6fJLYRTxljIRSQKWB m/nAWuMMZcDVcCtkY7HTvvrOnj9SPOotZRqWj1sPNx0ZuD4cNo9A2w83MTx5p5Rj52WlKAJWQhkJrvOqXj d3e/jtSNNHG7osjEqpZQKndSkBP7+urnsqm7n+b2n7Q4nrtnRffkg8PPg7UuAV4O3XwIutSEeWzR09rFhfwPbT7T y2pGmYffzBwxP7z7FrpNtvDDKi2HD/gZ2nWzj2T11WjTTJq8cbGTHCetv1daja5oqpeLDXZWlLCjK4P8+fwDPg M7uD5eIJmUi4gKuMsb8JbgpC+gM3u4Asoe530MiskNEdjQ1DZ/ARFogYNh3qoN3mrrHfV+X04FjyBIwwxE406 LlHqVlyyFQ39FLv8+PQ0cLhlSHx8uu6jbaPWcTLc+Aj13VbTR2nS1QOvi3dIqQ4NQmfqVUfHA6hK+uW0RdRx/ f/4uWyAiXSI8pux94bMjv7cD04O2M4O/vYox5BHgEoLKyMmoqW2470cobx6xyCHdVllCSPfbCoNNSE7nnklLa PAPMzR9+9qPDIdx9cSm1bR5m5aaN+JgGg9PhQBC0dmxoPfVWLW0eL7tOtvHgFbMAeGFvPTWtHhITHHz8il kkJjhYMy+PkuxkctISSXe7bI5aKaVC5+Lyady+fDo/ee0d7lxZwqy8kT+T1PhFuj1lHvBJEVkPLAIqgauC/3ct8GaE4 5mUQHApl54BH4EJlPsqyHAzfwzlEzKTXSwqziQ5ceSWMhEHeelJJEVpgdJY5g8muYEh2W4gWPneGIPBup3gd LCgKENntY5DV5+Xo43deLVmnlJR7x9unI87wcnDz76tK8eEQURbyowxXxq8LSKvG2P+SUS+JCKvA9XAdyIZ z2QlJjjo6feRlOAkyWV/IrR2UQH76zopyU4esUtUjd+ty4o53NDN7Pyz3wxvuKiQt+s6KZ2WopMmJsjnD/D4thq6+ 33Myktl3bLpo99JKWWb/HQ3n7luLl977m1e3N/ADYsL7Q4prtiWSQRnXGKM+box5nJjzH3GmLCOjPb6AzR29 p1p4ZisngE/xVlWV1U0LGuU7nZx6ayccXWjToRnwEfLKDNB401OWhKrKnLIS086sy0jeL2nZ+n6ihPlCxg8wU kpXX32v4aUUqP7yKoy5hWk87Xn3tZJZSFmf/NOBD2xo5Zfb63mhX2hmdL7npnTWDYji9Wzc5mVmxqSx4x2X X1efr7lJL944yRvaSFBNUlul5MbLypkUXEG1y8qsDscpdQYJDgdfHXdIk619/LtPx+yO5y4MmWWWfIHzJlZcvU dfaPsPTZul5M18/JD8lixot3jpc9rfTMK1XVUU9ucgnRd6kupGPOeWTnce8kM/uv149y0pJilpVl2hxQXpkxLmdMh XLuggNJpKayZP7USqVAqyU5m+YwsZuam8p5ZOXaHo5RSyiZfvnE+eelJfOnJKp2oEyJTJikDa9HlO1eWUKHT eCdMRLh6Xj63Lp/OtNREu8NRSillkwy3i6+tW8zB+i5+vPGY3eHEhSmVlCmllFIqdK5fVMj7lxTxvZePcrRRl5eb LE3KlFJKKTVhD9+8iNQkJ5/97R7txpwkTcpi1L5THfzyjRPsPNlqdygR09zdz2+2VfNcVZ2+8JVSKkrkpSfxr7cvY e+pDr770hG7w4lpmpTFqNeONNPcPcBrR5qnTFXlt6rbqe/o40hDNydbeuwORymlVNANiwv5QGUp//nqUbafmD qNBaEWF0nZlqPNPLunjrae8deePd7cw9O7T3GwvnP0naPIzGBdtPKcVESmxsLX5TkpOERISXSSn2H/Mka1bR6 e3n2Kqtr2kDyezx/g5QMNrN93WgsyKqVizj/evJDSaSl89re76erz2h1OTIr5OmV17b1sPW5l5Q4R3r+kaFz3f+ntBrr 7fZxs8TA3P33UdSijxdpFBayenUNaUsz/CcdsTkE6D2Wn4HQIiVGwvucrh5po7urneHMPc/LTR12bdDQH67uoqu 0AICPZxWUVuaEIUymlIiI1KYF//8Ay7vrRG3zhiSp++KEVU6bRIFTs/2SbpHR3wpl1J4cugTNWg/fJSUuMmYQ MrNIU6W7XlHvCJyc6oylhA8hLs547mcmukMSUk5aI0yGInH1spZSKJStmZPPl981n/f56fvracbvDiTkx38yS7nbx 4VXl9PT7KJhAl9ZNS4po7OonVz8E1Thdv7CAJSWZTEu1kqnJKspM5iOryvEGAvp8VErFrL+6fCY7T7bx/9YfZ ElJphYaH4foaHKYpLSkhAklZGCt4VWclRw1rS8qdjgcQnFWMm7X5Loth8pMcWlCppSKaSLCN+5cQllOCp967 C1Od/TaHVLM0ExEKaWUUiGV7nbxow+tpM/r56M/264D/8doyidlfV4/h+q76On32R2KUlHFGMOxpm4aO3Xhe aXU+M0tSOcHH1zBkcZu/vaxt/BpfclRTfmk7Ondp3hh72l+u73G7lCUiipbj7fyzO46frOthubufrvDUUrFoKvm5vEv ty5m4+Em/vfT+6ZMXc2JivmB/pPV3W/Vg+r1+gkEzIgzMBs7+0hzJ5CSOOUvW0gFAobGrn6yUlwhHZ+lJsczYL UeB4zRumlKqQm755IZnGrv5T/+chS3y8k/3rRwylUOGKspn128/6Ii9td1MGeUGmVb32lhy7EWkhOd3H9pGalTq D5YuL10oIH9dZ1kpbi4/9IyEpxTvgE3KqyalYvT4SDDnUDptBS7w1FKxbDPXTeXnn4//735OC6ngy+/b74mZhcw 5TOLwkw3hZmjz9xs7LK6b3oH/HT2eSOalAUChl6v35ZEcMAXIGBMWFuwBq9tu8fLgD8QtqTM5w/g9ZtJF3m

NNLviTk50ctXcvIgeUykVn0SE/33TArz+AI9seodAwPCVGxfEVH3QSJjySdlYrZ6dS8AY8tKTKMpMjuixn3rrFN WtHpaUZHLNgoKIHbetZ4DHt9fg9Qe4ZWkx5cGlnUJtzfx8th9vpTw3NWxdwz39Pn6zrZrufh9rFxWyoCgjLMcJtZ 5+H49traZnwMcNiwuZXxgbcSul1PlEhH+6ZREOgZ++fpzm7n6+cedSLUk1hCZlYzQtNZF1y6ZH/Lhef4DqVg9gr dMZSXUdvfR5rbFE1a2esCVl07OSmb48vNe2qaufrj5rjNSJ5p6YScoau/rp7j8btyZlSqlY5nAID9+yiPwMN9988RAt PQN8/74VZCa77A4tKmh6GuVcTgerKnLISUtkVUVkqyJX5KVRnptCUaabJSWZET12qJVkJzO3IJ38jCRWlGXb Hc6YlWYnM6cgzYp7RuzErZRSwxERPrVmNt+8cwlvHGvhlu+/zoHTnXaHFRUk1qanVlZWmh07dtgdhlIqzlRW VqLvLUpF1o4TrXzqsV109Hr52rrF3LmyJO4mAIjITmNM5Vj21ZYypZRSStmisnwaz/2PK1hWmsUXfl/FQ7/cSW PX1C1YrUmZUkoppWyTl57Erx+8lK/cOJ+Nh5u4/t838bsdNQQCsdWTFwqalMWR2jYPf9hVy86TbXaHEjO2vtP CU2/V0qBLCSmllG2cDuGhKyt44dNXUJGXxhd/X8UtP3idbcdb7Q4tojQpiyMbDzdxssXDpsNNZ2bsqeG19gyw5 VgLJ5o9vHak2e5wlFJqypudn8bvP7GK796zjJbuAe7+8Rv81aPbeat6ajQ2aFIWYj5/gM1Hm9lytBl/hJteCzOsIrhZK S7cIaj7Yoxh6zstbDzcRL8v/pbZSU1yku62qsIUZCTZHI1SSimwZmeuWzadv/z91Xxh7Tx2Vbdx239u4UM/3cprR5 riev1MrVMWY1WnOs40t6YmJbC0NCtix37v/Hwump5JZoorJFXxDzd0s+VYCwAuh3DZ7NxJP2Y0SUpw8qFLy+ js85KfPvqqDkoppSInOdHJp9bM5oHLyvn11pM8suk49//XNmblpvKhS8u4Y2VJ3NU306QsxNKHLIWU5o7s5RU R8jNC11ykuRNo7u7H5w+QEqdrfbpdTlsWQQ8EDHtPdeB0CIunx3YNOKWUCqfUpAQeurKCD68q50/7TvPLN0 7y1efe5psvHuKGxYXcunw6qyty4mLd5Pj8pLXRnIJ07qp0IiJMz4rsckyh1uf1405w4nc6GPAF7A4nruyubWfjoSbA KhA8rzDd5oiUUiq6uV1Obltewm3LS9h3qoPHtlXz3J46nnrrFHnpSdyytJhbl01n8fSMmK11pklZGJRkp9gdQkg4R M609sXBF5Co4hjyhqHXVimlxmfx9Ez+720X8X9uXsgrB5v441un+OUbJ/mv148zKzeVm5YWc8vSYmbnp9kd6r hoRX81oqONXfR5AywsysDhiM1vHtHIGMPbpztxOR3MLdBWsmigFf2Vim3tngH+tK+eZ3bX8ebxFoyBhUUZ3 LKsmJuWFNnWYDKeiv5TKinr9/lp6uqnMMMdF33PkdDuGWDAH9CB8CruaVKmVPxo6Ozj+arTPLOnjt017QC sLMvmlqXF3HhREXnpkZtxH7VJmYi8B/h3wA/sMMZ8VkS+AKwDTgIPGGO8Iz3GRJMyYwy/evMkzd0DzMx N5dbl0ydwBlNLQ2cfv91egz9guGFxIQuKMuwOSamw0aRMqfhU3eLh2ao6nt1Tx8H6LhwCl1XkcsvSYtYuKiQzJ bwzOKN57cuTwHuNMVcA+SJyBbDGGHM5UAXcGq4DBwy09lj5XnN3/6j7d/Z58flHH9ze0++LyxpeYBVXHay 11tI9MOK+Pn+Azr4R82kVRTp6vRGvo6eUUnaYkZPCp9bMZv1nrmTDZ6/kb66eTXWrhy8+WcXF//ISD/58B8/s qcMzYH/R9YgO9DfG1A/51QcsAV4N/v4ScB/wRDiO7XQIaxcXcLihm2UlWSPuu/loM9uOt5KXnsS9l8zAOcxYq qONXTxfVU9igoP7LpkR9mw70uYWpFPf2Ue/N8DKsuxh9/P6A/xmWzUt3QOsqsjh0lk5EYxSjdcrhxrZXd1OYaa bD1SW6lhBpdSUMbcgnc+vncffXz+XqtoOntlTx3NVdbx0oIFkl5PrFhZw89JirpybS1JC5Msl2TL7UkSWALlAO1Z XJkAHcMFPfhF5CHgIYMaMGRM+7vzCDOYXjt4Fd6KlB4Cmrn56BnxkuC+cbFW3eggYQ5/XT0NXX8STMm NMWKf9Oh3Cmnn5ox6nq893piWtusWjSVmUO9lsPb/rO/ro9wVITozsG89knrfhfs4rpaYGEWFpaRZLS7P4yo0L2 Ha8lWer6vjTXmscWoY7gfctLuLmpcWsqsgZtnEm1CKelInINOD7wN3ASmBwcFcGVpL2LsaYR4BHwBpTFu4 YL6vIZcuxZsqmpQ6bkAEsL82muWuAlCQnM3NTwx3WOTYdbmJXdRsLizK4flFhWI7R2efliR219Pv83LpsOsX D1F2blprIirJsats0IYsF18/JZevxVmblpkU0IfP5Azy5q5aGzn6uXVDAwuLxjVHc+k4Lb7zTQkVeGjctKdLkTCkVE k6HsKoih1UVOfzTLYt4/UjzmRa03+6oITctiZuWFHHz0iJWzMgO63tPRJMyEUkAfgV8wRhTLyLbgb8BvgFcC7 wZyXiGMzM3dUxJVnZqIndfXBqBiN5tf10nxsDbpzu5bmFBWJ4k1S0eOnutcWJHGruHTcoArpqbF/Ljq/CYnZ/O7 PzII+Fo7Rmgrr0PsJ63403KBp/zRxu76fcFbFmJQSkV31xOB2vm57Nmfj59Xj9/OdjIM7vreGxbNY9uOUF5Tgp3X 1zKnStKQrqCzqBIt5TdBVwMfD2YRHwZ2CQirwPVwHciHM+Y1bR6eOlAA3npSbxvcVHEmjKHs2JGFruq211U HL7KxYUZbqpbPfR5/dx4UXha49TUkZOWxKy8VE539LGsdPxLS60oy+aNYy3Mzk/ThEwpFXZul5MbLyrixou K6Orz8uL+Bn63o4ZvrD/EtzYc5r3z87nn4lKumpsXsjJbU6pO2WQ8vfsU7zRZY3Huvrg05pdQGot9pzr489sNgPW BqK1hKp5pSQyl1Fgca+rmdztqeHJnLc3dAxRkJPGx1TP566sqLrh/NJfEiAk1rR6e2FHDtuOtZ7bNzk9DBHLSEslN S7QxuuENxr39ROvoO49BaXYKKYlOXE5hVoTHzCmllFLRqCIvjS+/bwFvfPkafvShlSwsyqChc/RSW2Oha19ew GtHmmno7KO2rZcFRemku10sKs5kTn46LqdE7QDjjYebaOrqD8adQVrS5P68mSkuHrxiFgFjcOkKCEoppdQZLqe DGxYXcsPiQkLV6zilPmmPNXWzfl89de29I+5XnGUN3stOcZEcHLvS2efl1UONZ5ZriEaDXarTUhNxJ4TmT+t0i CZkSiml1AhC1VgzZVrK/AHD81Wn8QcMde29fOzymcPue9XcPBZPzyTD7TozeO+1w80cbugCrOQnHLMuJuvq eXlcVHJu3EoppZSKDXH7yV3d4mHLsWa6+611ExwC6W4rB81IPlt7rLVngC1Hre7KQSJCbloSiUNamzKD90lM cOCOcLHNsbpQ3EoppZSKDXHZUtbT7+OPu0/hDxjqO/q4fUUJIsI9F8/gdEcvJdkpZ/Z9Zvcp2jxedte288mrKoZtgl w9O4eS7GSyUlwjFpRVSimllJqIuEzKHCI4xFq/aWg3XnKik1l5aefsO/j/CaPUHRMRynUGolJKKaXCJC6TsuREJ 3dXlnK6o495hSNXL1+3rJhjTT2UTUuJ2lmVSimllIp/cZmUAeRnuMc0GD/d7WJZaVb4A1JKKaWUGoGOCFdK KaWUigKalCmllFJKRQFNypRSSimlokDMLUguIk3AyUk8RC7QHKJwokE8nY+eS3SaKueyAtgVxsePRrEUbyzF CrEVbyzFCrEVby6QaozJG8vOMZeUTZaI7Bjrau2xIJ7OR88lOum5RM/jh1osxRtLsUJsxRtLsUJsxTveWLX7Uiml lFIqCmhSppRSSikVBaZiUvaI3QGEWDydj55LdNJziZ7HD7VYijeWYoXYijeWYoXYindcsU65MWVKKaWUUt FoKraUKaWUUkpFHU3KlFJKKaWigCZlSimllFJRQJMypZRSSqkokGB3AOEmIk7gVuBSIBtoB94E/miM8dkX2 fjpuUQnPZfoFO5zibVrFUvxxlKsEJPxpgGfwIo3i7Px/tgY02VfZO8WS7HC5OON+9mXIvJLoAp4GegAMoBrgaX GmA/ZGdt46blEJz2X6BTuc4m1axVL8cZSrBCT8T4D/JJ3x/thY8zNdsZ2vliKFUIQrzEmrn+A18azPZp/9Fyi80fPJ Tp/wn0usXatYineWIo1RuPdDDjO2+YANtsdWyzHGop44777EnhGRJ4DXgU6sbLWq4Bn7AxqgoY7l2ftDGqCzj +XTOBK4uNcYvnv8rSey5jF2ntLLL3mYu15GGvvAT8AXhWRKs4+FxYB/2lrVBc2XKw/tDWq4U3q2sZ99yWA iFwOXITVt9sBbAdmGWO22hnXRIhIJTAXcAJ+rIz8V/ZGNTEikgtcgvWkbQcqjTFfszWoCRpyLiuBo8BRY8x2e

6MaPxEpArxY55IFlAPVwOMmCsfGjEREbgF2Aks5O7ZjuzGmKYTHiKn3llh5zcXa81BEEoF7gDKs178LK+bvG WPa7YtseCKSgPVZMvhcOBKN1xbOiTULK9bD0RorTC7euE/KRORbQD5WApMDfMwY0yQifzHGvNfe6MZ HRP4reHMAyAPqsDLxfGPMQ7YFNgEi8how+OST4L8Lgf3GmCvtiWpiRGS9MeYGEfkM1tiB54DVwCljzD/Y Gtw4Db4uROR7QA/wCrAM68P7bluDGycRqQNOAg3AU8Azxpi2ED5+TL23xNJrLtaehyLyFLANa5D/SuB5oA W4zxiz1s7YLiSWJiaIiAA3Yr3ONhhjAsHt64wxT9sa3BiJyFeNMf84ln2nQvdlpTHmKgARWQI8ISJfsDmmiZo95F z2GmPuDN5+xd6wJuQpYAnwqDHmVQAR+ZMx5n22RjUxicF/bwPWBN80fiQir9sY00QFgv8uNMZcG7y9IUaf Y4eMMWtEZCZwO/CUiPQDTxtjQtFNE2vvLbH0mou152GWMeZf4cx787eDtx+wNarhPQrsBX7DuYPRHwWib WLCL4ETgA/4nyLyoDHmEPB3QNQlZSJSjdWqO/gcFmCRiFw9li8/UyEpSxCRRGPMgDGmSkRuA36F1ccba4b +vb4y5Lacv2O0M8Z8O9jk/6CIfAJ4zO6YJmGhiPwCqACSgN7gdrd9IU3Yz0Xkp0CNiPwK2Ij1Qb7D3rAmzhhzH PgW8C0RKQDWheihY+q9JcZec7H2POwRkf+F9fqvF5G/B1qBfnvDGla5Meb+87a9FWxNjTYlJjiDVUQeAR4Vk R/YHNNIPgvcAWwAfmWM8Y3ny89U6L68BDhhjGkcss0J3GWMedy+yMZPRBYBB40x/iHbEoEbjDHROrh4V MH+9/uBebHW3QcgImVDfq0zxniDtWquMMb8ya64JkpEioG1QAHWt+gtxpg99kY1fiKy1hjzYhgfP2bfW2LhNR dLz0MRSQZuAI4BR4CPYH1ZfswY02FnbBciIp8HrubdExM2GWO+aV9k7yYim4CbjDGdwd8TgUeA24wxmbY GNwIRuRH4MLAFuGVIi+/I94v3pEwppZRS5xoy6SOL4CQYrBa0qJqgJCLlQPv5EyZE5EpjzCZbghoHEVkDLAb eHMu11aRMKaWUmkJEZLglFl80xlwX0WBGEUuxwrDxCrB+LPFOhTFlSimllDqrG2u25VCCNW4v2gzGKpw7 ezgaY4Vz4wUr5jHHq0mZilnBgdV/ABYYYw7aHY9SSsWIA1hjss4Z7yYif7YpnpHEUqwwyXi1+1LFLBH5HVA EvGyMedjmcJRSKiYEi/O2GGMGztueEIV1ymImVph8vJqUqZgUnN14CFiDVRR0frAv//tYs4iOY6039t/GmN+Ly Erg20Aa0Aw8YIw5bU/0Siml1LsNN4BOqWh3K9bAycNAq4iswCoQWo617M2DwCoAEXEB/wHcaYxZCfw38C 82xKyUUkoNS8eUqVh1L/Cd4O3Hg7+7gCeCFfXrh1T/noc1JfnP1oodOAFtJVNKKRVVNClTMUdEcoD3AotFxG AlWQZrGZkL3gVrfb9VEQpRKaWUGjftvlSx6E7gF8aYMmNMuTGmFGsMWTNwh4g4gsvpXB3c/xCQJyJnujO DqyMopWwiIreJiBGR+TYc+0SweOqYtkcLEVkeXH4KEXk4WJl/uH0fPu/3PBFZH+YQ1SRpUqZi0b28u1XsSaA YqAX2AT8GtgIdwVkwdwJfF5E9wG7gsohFq5S6kHuB14F77A4khnwFa3zssERkYXBpok+KyC4RuRfAGNMEn BaR1RGIU02QJmUq5hhjrjbGrD9v2/eMMZ8EPm+MWQj8FTAX2Bv8/93GmCuNMUuNMYuMMT+JfORKKTg ze3o11uv0niHbrxaRV0Xk9yJyUER+LcGBoMFWrH8KJhp7B1vYzm8xEpF9waV5EJE/ishOEdkvIg+NI75yETkgIj 8J3ndDcH1LRGS2iLwkInuCsVSI5ZvBY+8VkQ8MOZ+NIvI7ETksIv9PRD4oItuC+1UE98sTkSdFZHvw512Jk4ik A0sutP6niHxcRP4UjPFh4BfAD4PXeOjSPn8EPjjW66AiT5MyFW+eE5HdwGvA14wx9TbHo5R6t1t59+zpQcuBz wALgVlYicWgZmPMCqyEY9iuuyE+FpxxXQl8OjgedazmAD8wxizCWhvyjuD2Xwe3L8VqcT+NNfN7GbAUuBb 4ZrBeFcFtf4c1K/x+YK4x5hLgp8D/CO7zXeDfjTEXB4/z0wvEU4nVC3AOEflb4GbgVmNMLzAA5AMOY0yvMe bokN13AFeM4xqoCNOB/iquGGOutjsGpdSoLjR7elfw923GmFqA4BescqxuTrBW8ADYiZUIjebTwZU/AEqxEq2 WMcZ43Bize8jxyoOtVdONMU8BGGP6gnFeDvzGGOMHGkRkI3Ax0AlsH6yJKCLHgA3Bx9yLVWcRrERuYbB RECBDRNKNMV1D4ikCms6L8X6sIRu3GmO8wW1fAv4NWCsiy4H/NaR1rRFrmIeKUpqUKaWUipjhZk+LyBe Du/QP2d3PuZ9T/RfY7uPcXh938DhXYyU7q4wxHhF5dfD/xuj8OJI5u57h+Ybbfv7jBIb8HuDsOTiCcfaO8Di9vDv +fVgtdCVYk50wxpwC7hWRr2J1Xf4BqAju7w4+jopS2n2plFlqkoabPX35BB/vBLACINgNOjO4PRNoCyZk84FLJ xc2GGM6gVoRuTV4vCQRSQE2AR8QEaeI5AFXAtvG8dAbgL8d/EVEll1gnwPA7PO2vQX8NfCMiBQH7zs4szy A1cKXOmT/uVygC1RFD03KlFJKRdJws6fvm+DjPQlMC3Z1fhI4HNy+HkgQkSrga8CbE3z8892P1S1aBWwBCr HOpwrYA/wF+OI4x7N+GqgUkSoReRv4xPk7GGMOApnBLtSh21/HGl/3fLCcx+0i8ibwMaxk79NDdl8DPD+OuF SE6dqXSimlVAwQkc8CXcaYC00EOH/fh40xD5+3bROwzhjTFqYQ1SRpS5lSSikVG37IuWPURvLq0F+C3arf1o QsumlLmVJKKaVUFNCWMqWUUkqpKKBJmVJKKaVUFNCkTCmllFlqCmhSppRSSikVBTQpU0oppZSKAv8f st6I5I0KciEAAAAASUVORK5CYII=",

```
"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 thody tr th:only-of-type \{\n'',
       vertical-align: middle;\n",
     n''
  "\n",
     .dataframe thody tr th \{\n'',
       vertical-align: top;\n",
  **
     }\n",
  "\n",
     .dataframe thead th \{\n'',\
       text-align: right;\n",
     n''
  "</style>\n",
  "\n",
  " <thead>\n",
     \n",
        \n'',
      CustomerID\n",
      <th>Age</th>\n'',
      <th>Annual Income (k$)</th>\n",
  "
      Spending Score (1-100)\n",
     \n",
  " </thead>\n",
  " <tbody>\n",
      \n'',
  "
      count\n",
       200.000000  n''
       200.000000  n''
       200.000000  n''
  "
       200.000000  n''
  **
     \n",
      n'',
      <th>mean\n",
       100.500000  \n''
       38.850000  n''
  **
       60.560000  n''
       50.200000  \n''
     \n",
      \n'',
```

```
<th>>std</th>>n",
"
     57.879185  n''
"
     13.969007  n''
"
     26.264721  n''
"
     25.823522  n''
"
   \n'',
"
    \n''
"
    <th>min</th>n",
"
     1.000000  n''
     18.000000  \n''
"
     15.000000  n''
"
     1.000000  \n'',
"
   \n",
"
    n''
"
    <th>25%\n".
**
     50.750000  n''
"
     28.750000  n''
"
    41.500000\n".
"
     34.750000  n''
"
   \n",
**
    \n''
"
    <th>>50%\n",
     100.500000  n''
**
     36.000000  n''
"
     61.500000  n''
**
     50.000000  n'',
"
   \n".
"
   \n",
"
    <th>>75%</th>>n",
**
     150.250000  n'',
"
     49.000000  n''
"
     78.000000  \n''
"
     73.000000  n''
"
   \n",
"
    n''
"
    <th>\max\n'',
"
     200.000000  n''
"
     70.000000  n''
"
     137.000000  n''
"
     99.000000  n''
   \n".
" \n",
"\n",
"</div>"
],
"text/plain": [
                  Age Annual Income (k$) Spending Score (1-100)\n",
    CustomerID
"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",
       1.000000 18.000000
                             15.000000
                                              1.000000\n",
"min
"25%
       50.750000 28.750000
                              41.500000
                                              34.750000\n".
"50%
      100.500000 36.000000
                               61.500000
                                               50.000000\n".
"75%
                                               73.000000\n",
      150.250000 49.000000
                               78.000000
"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 thody tr th:only-of-type \{\n'',
      vertical-align: middle;\n",
    n''
 "\n",
     .dataframe thody tr th \{\n'',
       vertical-align: top;\n",
    n''
 "\n",
    .dataframe thead th \{\n'',
      text-align: right;\n",
    }\n",
 "</style>\n",
 "\n",
 " <thead>\n",
    \n",
       \n'',
 "
     count\n",
     <th>mean\n",
     <th>>std</th>>n",
 "
     <th>min</th>n".
     <th>25%\n".
     <th>>50%</th>\n".
     <th>75\%n'',
     <th>\maxn",
    \n",
 " </thead>\n",
   <tbody>\n",
     n''
     CustomerID\n",
 "
      200.0  n''
      100.50  \n",
      57.879185  n''
       1.0  n''
```

```
 50.75  n''
"
    100.5  n''
**
     150.25  n''
"
    200.0  n''
"
   \n",
**
    \n''
"
   <th>Age</th>\n'',
"
    200.0  n''
"
     38.85  n''
     13.969007  \n'',
"
     18.0  n''
11
    28.75  n''
"
     36.0  n''
"
    49.00  n''
"
    70.0  n''
**
   \n",
"
    n''
"
   <th>Annual Income (k$)</th>\n",
"
     200.0  n''
**
    60.56  n''
**
     26.264721  n''
"
    15.0  n''
"
     41.50  n"
**
    61.5  n''
**
    78.00  n''
**
     137.0  \n''
"
   \n".
"
    \n''
"
   Spending Score (1-100)\n",
"
    200.0  n''
"
     50.20  n''
"
     25.823522  n''
"
    1.0  n''
**
    34.75  n''
"
    50.0  \n'',
**
    73.00  n''
"
    99.0  n''
   \n".
" \n",
"\n",
"</div>"
],
"text/plain": [
                          std min 25% 50%
                                              75% \\\n",
            count mean
"CustomerID
                 200.0 100.50 57.879185 1.0 50.75 100.5 150.25 \n",
"Age
              200.0 38.85 13.969007 18.0 28.75 36.0 49.00 \n",
"Annual Income (k$)
                   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,
```

"image/png": "iVBORw0KGgoAAAANSUhEUgAABGUAAAE9CAYAAACiD99jAAAAOXRFWHRTb2Z0d 2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACX BIWXMAAAsTAAALEwEAmpwYAAAhHUlEQVR4nO3de5QtZ1kn4N+bBGIICIQEDAnjAQ4XASGEgATQ4S Z3ISgOyVIGF47MeAlRB8VIFgZWZi0VRmCCIIgYdBhAgWAWIBeBgI4juZEbl2AjB02IIREhQLiGb/6o6mSn03 3O7qS7v9p9nmetXt27du3ab9Xuent/v11VXa21AAAAALC19uldAAAAAMDeSCgDAAAA0IFQBgAAAKADoQ wAAABAB0IZAAAAgA6EMgAAAAAd7LeemQ8++OC2Y8eOTSoF2EznnnvuVa21Q3rXcVPpP7DY9CCgl0XvP 4keBItsTz1oXaHMjh07cs4559z8qoAtV1Wf713DzaH/wGLTg4BeFr3/JHoQLLI99SCnLwEAAAB0IJQBAAAA6E AoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB0IJQBA AAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB 0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADrYr3c Bi+TUU0/N0tJS7zI2xWWXXZYkOeywwzpXsj127tyZ448/vncZsKqevcW+Dyyb2vucnv1J74Dtbyo9b7u9F9M/108 osw5LS0s5/+JP5dpbHdS7lA237zVfSZL867e236/Evtd8qXcJsFs9e4t9H1g2tfc5vfqT3gF7h6n0vO30Xkz/vGkW/5Xf2gF3gF3h6n0vO30Xkz/vGkW/5Xf2gF3h6n0vO30Xf2gF3h6n0Ytfe6qB8495P6l3Ghjvg0+9Jkm29bjBlvXqLfR+YNaX3Ob36k94Be48p9Lzt9F5M/7xpXFMGAAAAoAOhDAAA AEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAo AOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgV AGAAAAOAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAM AOhAKAMAAADQgVAGAAAAAOAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAAA6EMoAAAAAd CCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAAB0IZQAAAAA6E MoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB1sSi hz6qmn5tRTT92MRQMj+9nqbBfYGva11dkusDr7xuazjWFrbPS+tt+GLWnG0tLSZiwWmGE/W53tAlvDvrY62wV WZ9/YfLYxbI2N3tecvgQAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAAB0IZQAAAAA6E MoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZ QAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyA AAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAA ACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAA EAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoA OhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVA GAABYKEtLS3nyk5+cpaWl3qUA3CxCGQAAYKGccsop+frXv55TTjmldykAN4tQBgAAWBhLS0vZtWtXkmT Xrl2OlgEW2n6bsdDLLrss3/jGN3LCCSdsxuK7WVpayj7fbr3LYJ32+ebVWVr66rb8fTzggAN6lzE5i9h/9JbNsV33/ anQg1a3iD1omV400Ds2x0b2jJVHx5xyyik57bTTNmTZi2yR+08Pet7G21v650a/B9rjkTJV9dyqOqeqzrnyyis37IkB 9kT/AXrSg2Calo+SWev2dqEHwd5hj0fKtNZel+R1SXLUUUfNFSUedthhSZJXvvKVN6e2yTnhhBNy7j9d0bsM1u 173/f92Xm3O23L38ftbm/pP3rL5tiu+/5U6EGrW8QetEwvGugdm2Mje8aOHTtuEMTs2LFjw5Y9JevtQYvcf3rQ8zb e3tI/N/o9kGvKAAAAC+Okk07a7W2ARSKUAQAAFsbOnTuvOzpmx44d2blzZ9+CAG4GoQwAALBQTjrppBx4 4IGOkgEW3qb89yUAAIDNsnPnzrz73e/uXQbAzeZIGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB0IJQ BAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAA AB0IJQBAAAA6EAoAwAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADo QygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAA AAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAA HQhlAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDI

AAAAAHQhlAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAf7bcZCd+7 cuRmLBWbYz1Znu8DWsK+tznaB1dk3Np9tDFtjo/e1TQlljj/++M1YLDDDfrY62wW2hn1tdbYLrM6+sflsY9gaG7 2vOX0JAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAAA6EMoAAAAAdCCUAQAAAOhA KAMAAADQgVAGAAAAAOAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUA QAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAA AAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAA AA6EMoAAAAACCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAA B0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADo QyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhk AAACADoQyAAAAAB3s17uARbPvNV/KAZ9+T+8yNty+1/xbkmzTdftSkjv1LgN2q1dvse8Ds6b0PqdXf9I7YO8 xhZ63nd6L6Z83jVBmHXbu3Nm7hE1z2WXfTZIcdth23InutK1fOxZfz99P+z6wbGr7S7/+pHfA3mAq+/n2ei+mf94 UQpl1OP7443uXAGxDegswBXoRsDfR85gK15QBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCG QAAAIAOhDIAAAAAHQhlAAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAA QAdCGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6E MAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB0IJQBAAAA6KBaa/PPXHVlks/PMevBSa 66qUVN2HZdr8S6Lar1rNsPttYO2cxiNtM6+s+8FuX3Qp0bS50bSw+66ab2GqtnbVOqJVHPnqxWz0L3n2SuHjS11 2 Ee at 58i1 Zvsng 1z 1 PvbnvQukKZeVXVOa 21ozZ8wZ1t1/VKrNui 2s7rttkWZdupc 2 Opc 2 MtSp1TNLVtp561 TamWRAM 2 Strategy 2 SD17MrV6tsoirreaN9+i1ZssXs0bUa/TlwAAAAA6EMoAAAAdLBZoczrNmm5vW3X9Uqs26Lazuu22RZl26lzY6 lzYy1KnVM0tW2nnrVNqZZEPXsytXq2yiKut5o336LVmyxezTe73k25pgwAAAAAu+f0JQAAAIAONjSUqaonV NUIVbVUVb+1kcvuoap2VdVFVXV+VZ0zTjuoqj5QVf84fr997zrnUVVvqKovVtXFM9PWXJeqOnF8HS+pqsf3q Xo+a6zbyVV12fjanV9VT5q5byHWraruUlUfrqpPVdUnquqEcfq2eN22yk3Zjp3q/L6qOquqLhjrfPEU61xWVftW1c er6l3j7cnVuSg9vKpuV1Vvq6pPj7+nR0+tzqq610w/Pb+qrq6qX51anVM01R40pX14avtAVf3a+FpdXFVvHvvzltW zxvuabn/716jnpePrdWFVnV5Vt9uKelarZea+51dVq6qDt6KWKamJj8Gm2gfnMaVeOY+p9dM96d1v56xx03vyho UyVbVvkj9M8sQk90lyXFXdZ6OW39GjWmtHzPybq99K8sHW2j2SfHC8vQhOS/KEFdNWXZfxdTs2yX3Hx7x6f H2n6rTceN2S5OXja3dEa+09ycKt23eT/PfW2g8leWiSXx7r3y6v21ZZ13bs6FtJHt1ae0CSI5I8oaoemunVueyEJJ+au T3VOhehh78yyXtba/dO8oAM23VSdbbWLlnup0kelOSaJKdnYnVO1FR70JT24cnsA1V1WJLnJTmqtXa/JPtm+Nu 6lfWclmm9Z1utng8kuV9r7f5JPpPkxC2qZ7VaUlV3SfLjSf55Ztpe8b5oQcZgU+2D85hSr5zHZPrpnkyk387jtGx2T2 6tbchXkqOTvG/m9olJTtyo5ff4SrIrycErpl2S5NDx50OTXNK7znWsz44kF+9pXVa+dknel+To3vWvc91OTvL8VeZ buHWbqfWvMrzh2Dav2xS34xS+ktwqyXlJfmSKdSY5fPwD9Ogk7xqnTbHOyffwJN+f5HMZr/E21TpX1Pa4JP936 nVO9WsKPWhK+/DU9oEkhyX5lyQHJdkvybvG3/ktrWeV9zVd//avrGfFfU9P8qatqme1WpK8LcMA9Lq+v1Xbpv dXFnAMNoU+OGedk+mVc9Y7qX46R72T6Ldz1rqpPXkjT19a3qjLLh2nLbKW5P1VdW5VPXecdqfW2uVJMn6/ Y7fqbr611mW7vJa/Mh5W+4aZQ8oWct2qakeSByb5WLb/67Zp5tyO3YyHyJ6f5ItJPtBam2SdSV6R5DeTfG9m2hT rXIQefrckVyb50/Hw6NdX1YGZXp2zjk3y5vHnKdc5ORPqQa/IdPbhSe0DrbXLkrwswxEXlyf5Smvt/b3qmTHlv/3P SfLXveqpqqcmuay1dsGKu6awbbbCQq3nhPrgPF6R6fTKeUyqn+7JhPvtPDa0J29kKFOrTFv0f+308NbakRkOB/zlq vqx3gVtke3wWr4myd0znAZyeZL/OU5fuHWrqlsneXuSX22tXb27WVeZNul120rr2I7dtNaubcPpIYcneUhV3a9zS TdSVU9J8sXW2rm9a5nDIvTw/ZIcmeQ1rbUHJvl6+h+mu6aqumWSpyb5y961LJqp9KAJ7sOT2gfGD3GeluSuSe6 c5MCq+tle9cyh69/+qnphhlNT3tSjnqq6VZIXJnnRandvZS0dLcx6TqUPzmOCvXIek+qne7KA/XYeN2l/3MhQ5tIkd 5m5fXiSL2zg8rdca+0L4/cvZjh3/iFJrqiqQ5Nk/P7FfhXebGuty8K/lq21K8YB7veS/HGG1y5ZsHWrqltk+OP1ptbaO8 bJ2/Z12yzr3I7dtda+nOTMDOeiTq3Ohyd5alXtSvKWJI+uqv+d6dW5KD380iSXjkdFJcMh+EdmenUue2KS81prV4 y3p1rnpEysB01tH57aPvDYJJ9rrV3ZWvtOknckeVjHepZN7m9/VT07yVOS/Ewbj9PvUM/dMwzoLhh/pw9Pcl5V/ UCHWnpZiPWcWB+cx9R65Tym1k/3ZKr9dh4b2pM3MpQ5O8k9ququ4ydpxyY5YwOXv6Wq6sCqus3yzxnOb7s4 wzo9e5zt2RnOiVxUa63LGUmOrar9q+quSe6R5KwO9d1kyzvJ6OkZXrtkgdatqirJnyT5VGvtD2bu2rav22a4Cduxi6 o6pMb/XIFVB2T4Q/XpTKzO1tqJrbXDW2s7MvT5D7XWfjYTq3NRenhr7V+T/EtV3Wuc9Jgkn8zE6pxxXK4/dS mZbp2TMbUeNLV9eIL7wD8neWhV3Wp87R6T4UKZvX/XJ/W3v6qekOQFSZ7aWrtmRZ1bVk9r7aLW2h1bazv G3+lLkxw5/l7tLe+LJj8Gm1ofnMfUeuU8JthP92Sq/XYeG9uT93TRmfV8JXlShiuwfzbJCzdy2Vv9leGcvAvGr08sr0 +SO2S44NM/jt8P6l3rnOvz5gyn8Xwnwx+sn9/dumQ4FPSzGS5i9MTe9d+EdfvzJBcluXDcOQ5dtHVL8ogMh7tdm OT88etJ2+V1m/J27FTn/ZN8fKzz4iQvGqdPqs4VNT8y11/4blJ1LllPz3Ca5Tnja//OJLefaJ23SvJvSW47M21ydU7ta 8o9aCr78NT2gSQvzhCKX5zh/cT+W1lPJvaebY16ljJcN2H5d/qPtqKe1WpZcf+uzFzgfbO3zVS+MvEx2JT74Jz1T6J XzlnrpPrpHPV27bdz1rjpPbnGBwIAAACwhTby9CUAAAAA5iSUAQAAAOhAKAMAAADQgVAGAAAAoAO hDAAAAEAHQpkJq6ofqKq3VNVnq+qTVfWeqrrnOpdxTFXdZ7Nq3M3znllVR40/76qqi8avT1bVKVW1/1bXBP RVVU+vqlZV9+5dCwAwfVX1wqr6RFVdWFXnV9WPbPLznVxVzx9/fklVPXaDlvuccSx0YVVdXFVP24jlzvncB 1TVR6pq3/H2e6vqy1X1rj08btX5ququVfWxqvrHqnprVd1ynF5V9b+qamlczyPH6besqo9W1X6btY6LTigzUVVVS U5PcmZr7e6ttfsk+e0kd1rnoo5JsqWhzPIOv8KjWms/nOQhSe6W5HVbWRMwCccl+bskx/YuBNh4PYPX8QOgg+ edPhVV9cCqev3483WDwTXmPXnF7UOq6r2bXCJ0U1VHJ3lKkiNba/dP8tgk/7JVz99ae1Fr7W9u7nKq6vAkL0zyi HE9Hprkwpu5zNXGW2t5TpJ3tNauHW+/NMmz5njcWvP9XpKXt9bukeTfk/z8OP2JSe4xfj03yWuSpLX27SQfTP

LMddS8VxHKTNejknyntfZHyxNaa+cn2Xc2rayqV1XVz40//+54JMqFVfWyqnpYkqcmeemYLN+9qo6oqn8Y5zm 9qm4/PvbMqnr5mGJ+qqoeXFXvGBPQU2ae72er6qxxea+dSVy/NqbJH0ty9For1Vr7WpL/luSYqjpoA7cXMGFVde skD8/wh/vYcdo+VfXq8ROwd41HAz5jvO9B46c651bV+6rq0I7lA/MRvK7fbyc5dXczVNV9quqjSX6xqs6rquOSpL V2ZZLLq+rhW1An9HBokqtaa99KktbaVa21LyTXBa6/N45LzqqqneP0Q6rq7VV19vj18HH6yVX1hnHM809V9bzl JxmPxrmkqv4myb1mpp82875kV1W9eNwHL1oOn8fn+8A4/bVV9flVguA7Jvlqkq+N6/G11trnxsfvrKq/qaoLxmXcf Tzi5KU1HFFzUVU9c5z3kVX14ar6P0kuqqp9x/nOHsd2/3WN7fgzSf5q+UZr7YNjPbu12nzjgQOPTvK2cdIbMxwE kCRPS/JnbfAPSW438/7tnWMdrEIoM133S3LuvDOPAcfTk9x3TGBPaa39fZIzkvxGa+2I1tpnk/xZkheM81yU5Hd mFvPt1tqPJfmjDDvuL491/FxV3aGqfihDwvnw1toRSa7N9TvXgUkubq39SGvt73ZXa2vt6iSfy5CiAnuHY5K8t7X2 mSRfquGQ1p9MsiPJDyf5LxkD3aq6RYZByjNaaw9K8oYk/6NDzcCcVgtex+mPHAdBb6uqT1fVm8Y39bsb5Nzgi JFxYLJj/PmdY1j7iap67jrq2zF+6PTH42PfX1UHjPetd1D0kar6i6r6TA0fiP3MOCi8qKruPs636sBwRU23SXL/1toFq 9z3C1X112ONJ2d4//aacRufPTPrO2Ogw/b1/iR3Gfe1V1fVf1xx/9WttYckeVWSV4zTXpnhKI4HJ/mpJK+fmf/eSR6 f4cj936mqW1TVgzL0rAdmeF/y4N3Uc1Vr7cgM++Jyj/qdJB8ap5+e5D+s8rgLklyR5HNV9adV9RMz970pyR+21h6 Q5GFJLh/rOCLJAzIcHfTSmXDjIUleOJ5F8fNJvjKu64OT/EJV3XX2iWs4tehurbVdu1mv9bhDki+31r473r40yWHj z4flhkcyzd53cXa/bfdqQpnt4+ok30zy+qr6ySTXrJyhqm6b5HattY+Mk96Y5MdmZjlj/H5Rkk+01i4fk+l/SnKXJI9J8q AkZ1fV+ePtu42PuTbJ29dRb61jXmDxHZfkLePPbxlvPyLJX7bWvtda+9ckHx7vv1eGQPgDY685KcnhW1susE7H5 MbB67IHJvnVDKdT3y1DsLBstUHO7jxnDGuPSvK8qrrDOmq8R4bBz32TfDnDgC1Z/6DoAUlOyBAoPyvJPceB4 euTHD/Os7uB4bKjMgxUbqCqfiXJTyQ5prX2jSTfzvBJ+z6ttW+01pZmZj8nyY+uYxvAwhiPsH9QhlNhrkzy1hrPE Bi9eeb78pH6j03yqvH9wxlJvn8MQJPk3a21b7XWrkryxQyXhfjRJKe31q4ZPzg+I2t7x/j93AwfKiXDe5m3jPW+N8 PpPCvX49okT0jyjCSfSfLyMXy+TZLDWmunj/N9s7V2zbjMN7fWrm2tXZHkI7k+0Dhr+SibJI9L8p/Hdf1YhsBk5 YfeB2fodxtltTFc29N94zb49sxrwQwX25muT2TYcVf6bm4Ypn1fkrTWvltVD8kQlByb5FcyHFq2Ht8av39v5ufl2/tl 2NHe2Fo7cZXHfnPmPMXdGnfGHRmaErDNjYOmRye5X1W1JPtm+CN9+loPyRAMr3kqJDA5x+X6T6qXg9fzxt tntdYuTZJx8LAjw2lOyQ0HOT85x/M8r6qePv58lwwDkH+bs8bPjaeCLz/fjtUGRWOd1w2KklxRVcuDoquTnN1au 3yc77MZPs1Phg+1HjX+/Ngk9xkPCkrGgWFrbfZUgEMzDDRnPSvDp8vHtNa+M057QZKXJXl8VT0wyUkzR9d8 Mcmd51x/WDjjPnhmkjOr6qIkz05y2vLds7OO3/dJcvQYaF5n3BdnxzfX5vqx8Oxydmf58bOPneuD5tZaS3JWkrOq6 gNJ/jTJH6wx++6W+fUV8x3fWnvfbub/Rsbx4u7UcAHl1443X9RaWyucuirDaUn7jUfLHJ7kC+N9l2boy8tm70uS/ TMcRMAKjpSZrg8l2b+qfmF5QlU9OMNg5j5Vtf945MtjxvtuneS2rbX3ZPg06ojxYV9Ncpskaa19Jcm/V9XyJyrPyp C8zuuDSZ5RVXccn/OgqvrB9azUWOerk7yztXajJBnYlp6R4RzjH2yt7Wit3SXDKYxXJfmpGq4tc6ckjxznvyTJITV c4C/j4cX37VE4sGczwevrq2pXkt9I8sy6PpFYayA0e9/s9FU/gKqqR2YIO44ej2r5eOYYbKzyXLPPt9bgZ3eDopUfX M1+qLW8DssDwyPGr8NWBDLJ6oOlizOEVtcdHdhau6y1dlyGUzRel+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+c5PzvDIbgXZmgIL1lHTZ/McBrB+8fHfyDDpzzz+HBVXZyhofxzkrUuRAVsP8flxkfFvD3Dp7uXZ hiEvDbDobdfGa/S/4wkv1dVFyQ5P8MpBcA0rRW8PuImLm9XkuV/pXpkkuVrJNw2yb+31q6p4fozD715ZV93nbu bMyhayzwDw08l2bli2sczvEc6o6ruPD52OZT+XoYjfGYHXffMKqdAwTZx6yRvrPEfmWQ4BfLkmfv3r+GfjJyQ6 8c+z0tyVA0Xvv1khn8wsqbW2nlJ3prhvcbbk/ztOmt8cZLHVdV5Gf770OW58UV0b5HkZTVcV+v8DCHSCeN9z8 pwBOCFSf4+yQ9keM90YYZr0XwoyW+Op3mv9Pokn0xy3jjOem1WPxPm/Znpx1X1t0n+MsljqurSqnr8aiu2m/lek OTXq2opwylTfzJOf0+Gy14sJfnjJL80s7hHjfezihrG/gCw9arq1q21r42foJyV4ULiq73xACaqqs5M8rvj9RSWpz0vy Q9lGOw8v7X2lHH6q5Kc01o7bTyq5qjW2lVVdVSSl7XWHlnDxW3/KsN1VM7OMJhYHuy8M8OFIy9JckiSk1trZ 84ua0VtuzJcu+XWSd7VWrvfOP35SW7dWjt5/CT+tRmuvfCdJD+dIVT6/fF5W4Z/oPDW8Wid2fU5c7x9zux9Nfz3 lT8ct8F+ST7aWrvR4HA8HeNhrbWv1vAvr7/WWnvZOPj53SQ/nuQXkzw5Q5B9dZKXtNb+YmY9vtVa2+1/cILtZ q19vkMd+ye5dryUxNFJXtOGf4gyGeNpj7/eWpvn32BvVg3vSHJia+2SXjVMmVAGgG7GAc3tktwyye+31k7rWQ/ AVqqqX0vy1dbaahcCXjnvya21k1dM+2iSpzklnL3NhEKZe2Q4i2GfDBfl/qXW2tm7f9TWq6rnZLg26FzXAN3g57 5lkmNba3+21c+9KIQyAADQwXidh59urf35HPM+srV25sztQzIcXfjOzasQgM0mlAEAAADowIV+AQAAADoQy 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$)'])\n",
"\n".
"plt.subplot(1, 5, 4)\n",
"sns.boxplot(x=data['Spending Score (1-100)'])\n",
"\n",
"plt.subplot(1, 5, 1)\n",
"sns.boxplot(x=data['CustomerID'])"
},
"cell type": "markdown",
"id": "6718d9d0",
"metadata": {},
"source": [
"## Handling outlier"
"cell type": "code",
"execution count": 11,
"id": "cf142f0b",
"metadata": {},
"outputs": [
 "data": {
 "text/html": [
  "< div> n",
  "<style scoped>\n",
     .dataframe thody tr th:only-of-type \{\n'',\n''\}
        vertical-align: middle;\n",
     n''
  "\n",
     .dataframe thody tr th \{\n'',
        vertical-align: top;\n",
  "
     }\n",
  "\n",
     .dataframe thead th \{\n'',\
        text-align: right;\n",
     n''
  "</style>\n",
  "\n",
  " <thead>\n",
     \n",
        \n'',
  "
      CustomerID\n",
      <th>Age</th>\n",
      <th>Annual Income (k$)</th>\n",
       Spending Score (1-100)\n",
```

```
\n",
  " </thead>\n".
     n'',
      \n'',
      <th>>0.25</th>>\n",
  **
       50.75  n''
  "
       28.75  n''
       41.5  n''
       34.75  n''
     \n",
  "
      \n''
      <th><0.75</th><n",
  "
       150.25  n''
       49.00  n''
       78.0  n''
  **
       73.00  \n''
    \n",
  " \n",
  "\n",
  "</div>"
 ],
 "text/plain": [
      CustomerID Age Annual Income (k$) Spending Score (1-100)\n",
  "0.25
                              41.5
                                             34.75\n",
          50.75 28.75
  "0.75
          150.25 49.00
                               78.0
                                             73.00"
 ]
 "execution count": 11,
 "metadata": {},
 "output type": "execute result"
],
"source": [
"quant=data.quantile(q=[0.25,0.75])\n",
"quant"
"cell type": "code",
"execution count": 12,
"id": "a3843246",
"metadata": {},
"outputs": [
 "data": {
 "text/plain": [
  "CustomerID
                       150.25\n",
  "Age
                    49.00\n",
                          78.00\n",
  "Annual Income (k$)
  "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",
                           -13.250\n",
  "Annual Income (k$)
  "Spending Score (1-100) -22.625\n",
  "dtype: float64"
 "execution count": 15,
 "metadata": {},
 "output type": "execute result"
],
"source": [
"low=quant.loc[0.25]-(1.5 *iqr)\n",
"low"
"cell type": "code",
"execution count": 16,
"id": "da5d37d2",
"metadata": {},
"outputs": [
 "data": {
  "text/plain": [
  "CustomerID
                         299.500\n",
  "Age
                      79.375\n",
  "Annual Income (k$)
                            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": "iVBORw0KGgoAAAANSUhEUgAAAWAAAEGCAYAAABbzE8LAAAAOXRFWHRTb2Z 0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAA CXBIWXMAAAsTAAALEwEAmpwYAAANNElEQVR4nO3df5BdZXnA8e+TrAOBihqT0rgwLrJYCihCoyOFdij aVh2rgJ2RiAwzdmrrlGV12qFS/4ntP52hY5vZtnQobR3bDJ2O/ChFpbFaZBxHIVEI0QS8LajZRgjFqtNEFPL0j3M2 XC6bkE1273Pv3e9nZmf3vvfsPe+bvfnm7Nncs5GZSJL6b0X1BCRpuTLAklTEAEtSEQMsSUUMsCQVGVvIxmv WrMmJiYklmookjaZt27Y9nplre8cXFOCJiQm2bt26eLOSpGUgIr4137inICSpiAGWpCIGWJKKGGBJKmKAJamI AZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkoos6Hf CaenMzMzQ6XSqp7FoZmdnARgfHy+eycJNTk4yNTVVPQ0tAwZ4QHQ6He7bsZOnT1hdPZVFsXLf9wH47pPD 9RRbue+J6iloGRmuvx0j7ukTVrP/zLdUT2NRrNr1aYChW8/cvKV+8BywJBUxwJJUxABLUhEDLEIFDLAkFTH AkITEAEtSEQMsSUUMsCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLEIFDLAkFTHAkITEAEtSE QMsSUUMsCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLE1FDLAkFTHAk1TEAEtSEQMsSUUMs CQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLEIFDLAkFelLgGdmZpiZmenHriRpUS1lv8aW5FF7d DqdfuxGkhbdUvbLUxCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLE1FDLAkFTHAk1TEAEtSEQMsSUUM sCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLElFDLAkFTHAklTEAEtSEQMsSUUMsCQVMcCS VMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLEIFDLAkFTHAklTEAEtSEQMsSUUMsCQVMcCSVMQAS1IR AyxJRQywJBUxwJJUZKwfO5mdnWX//v1MT0/3Y3dDqdPpsOLHWT2NZW/Fj35Ap/NDn6s6qNPpsGrVqiV57O c9Ao6I90fE1ojYunfv3iWZhCQtR897BJyZNwA3AKxfv/6oDtHGx8cB2LRp09F8+rIwPT3Ntv96tHoay96B409i8h Un+1zVQUv53ZDngCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpi gCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIG WJKKGGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKA JamIAZakImP92Mnk5GQ/diNJi24p+9WXAE9NTfVjN5K06JayX56CkKQiBliSihhgSSpigCWpiAGWpCIGWJKK GGBJKmKAJamIAZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIA ZakIgZYkooYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYEkq YoAlqYgBlqQiBliSihhgSSpigCWpyFj1BPSMlfueYNWuT1dPY1Gs3Pc/AEO3npX7ngBOrp6GlgkDPCAmJyerp7 CoZmefAmB8fNhidvLIfS00uAzwgJiamqqegqQ+8xywJBUxwJJUxABLUhEDLEIFDLAkFTHAklTEAEtSEQMsS UUMsCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLEIFDLAkFTHAkITEAEtSEQMsSUUiM49844i 9wLeWbjrzWgM83ud99sOorgtc27BybUvn5Zm5tndwQQGuEBFbM3N99TwW26iuC1zbsHJt/ecpCEkqYoAlqcgw

BPiG6gkskVFdF7i2YeXa+mzgzwFL0qgahiNgSRpJBliSigxMgCPi1Ij4j4jYGRFfj4jpdnx1RHw2Ir7Zvn9J9VyPVkS sjIivRcQd7e2RWFtEvDgiPhkRu9qv3wWjsLaI+FD7XNwRETdFxPHDvK6I+LuIeCwidnSNHXI9EXFtRHQi4sGI+ LWaWT+/Q6zruvb5uD0ibo2IF3fdNzDrGpgAA08Bv5eZPwe8HvjdiDgL+DDwucw8A/hce3tYTQM7u26Pyto2AXd m5pnAuTRrHOq1RcQ4cDWwPjPPAVYClzHc6/o48KaesXnX0/7duww4u/2cv4qIlf2b6oJ8nOeu67PAOZn5auAh4 FoYwHV15kC+Af8C/ArwILCuHVsHPFg9t6Nczyk0T/BLgDvasaFfG3AS8DDtD3S7xod6bcA48B1gNTAG3AH86 gisawLY8XxfJ5pgXdu13b8BF1TP/0jX1XPfO4DNg7iuQToCPigiJoDzgK8AJ2fmHoD2/U8XTu1Y/DlwDXCga2w U1vYKYC/w9+3plRsj4kSGfG2ZOQv8KfBtYA/w/czcwpCvax6HWs/cP0Bzdrdjw+h9wGfajwdqXQMX4Ij4KeBm4 IOZ+YPq+SyGiHgr8FhmbqueyxIYA84Hrs/M84D/Y7i+LZ9Xey707cBpwMuAEyPivbWz6quYZ2zo/s9qRHyE5vT m5rmheTYrW9dABTgiXkAT382ZeUs7/GhErGvvXwc8VjW/Y3Ah8LaIeAT4J+CSiPhHRmNtu4HdmfmV9vYnaY I87Gt7I/BwZu7NzJ8AtwC/wPCvq9eh1rMbOLVru1OA/+7z3I5JRFwJvBW4PNvzDQzYugYmwBERwN8COzPzY 1133Q5c2X58Jc254aGSmddm5imZOUHzA4DPZ+Z7GY21fRf4TkT8bDv0BuAbDP/avg28PiJOaJ+bb6D54eKwr6 vXodZzO3BZRBwXEacBZwD3FMzvqETEm4A/AN6Wmfu67hqsdVWfPO86GX4RzbcC24H72re3AC+l+eHVN9 v3q6vneozrvJhnfgg3EmsDXgNsbb92twEvGYW1AR8FdgE7gH8AjhvmdQE30ZzP/gnNkeBvHm49wEeA/6T5Qd2 bq + e/wHV1aM71zrXkrwdxXb4UWZKKDMwpCElabgywJBUxwJJUxABLUhEDLElFDLAOioh3RERGxJkF + 34kBUhEDLElFDLAOioh3RERGxJkF + 34kBUhEDLETDLAOIOH3RERGxJkF + 34kBUhEDLAOIOH3RERGxJkF + 34kBUhEItYc6figiIjzIuLG9uONEfH7h9l2Y8/ttRFx5xJPUQPMAKvbBuCLNC8W0ZH5Q2DmcBtExFkRcTfwgYj4akRsAM jMvcCeiLiwD/PUADLAAg5eg+NCmv/EflnX+MURcVfX9X43t68Mmzs6/WgblQfmjpx7jwTb6+lOtB/fFhHb2uvsv n8B85torzX8N+3nbomIVe19kxHx7xFxfzuX06NxXbvvByLi3V3r+UJE/HNEPBQRfxIRl0fEPe12p7fbrY2ImyPi3v btOZGMiBcCr87M++e577ci4jPtHDcCnwCub/+M7+3a9Dbg8iP9c9BoMcCacynNNX0fAp6IiPO77jsP+CBwFs3Vz 7pj9Hhmnk8Tl0N++93lfZn588B64OqIeOkC5ngG8JeZeTbwv8C72vHN7fi5NNdr2AO8k+YVeufSXNfhurlrHrRj08 CrgCuAV2bm64Abgal2m03An2Xma9v93DjPfNbTvEruWSLiKuDXgUszcz/wY5qrjK3IzP2Z2enafCvwiwv4M9AI McCas4HmQkG07zd03XdPZu7OzAM0L+uc6Lpv7qJJ23rGD+XqiLgf+DLNRVHOWMAcH87M+7r31x6Fjmfmr QCZ+aNsXvt/EXBTZj6dmY8CXwBe237uvZm5JzOfpHlJ6pZ2/IGuNbwR+IuIuI/m+gEntfvqto7mUpzdrgDeDLyrf XxorknwKuCqiPjXiDi3a/vHaK62pmVorHoCqtcehV4CnBMRSfPbHzIirmk3ebJr86d59vPmyXnGn+LZ/7gf3+7nYp qwXZCZ+yLirrn7jlDvPFYx/+UFOcx47+Mc6Lp9gGfWsKKd5/7DPM5+njv/HTRH3qfQXKiebK4tvCEi/ojm9MMt wOnt9se3j6NlyCNgAfwG8InMfHlmTmTmqTTxuOgoH+8RmktS0p7KOK0dfxHwvTa+Z9L86qljks01o3dHxKXt/o 6LiBOAu4F3R/N7+NYCv8TCrnq1Bbhq7kZEvGaebXYCkz1jXwN+G7g9II7Wfu7Z7X0HaI7cT+za/pXMcxpDy4 MBFjSnG27tGbsZeM9RPt7NwOr22/cP0PxOLoA7gbGI2A78Mc1piMVwBc2pje3Al4CfoVnPduB+4PPANdlcOvN IXQ2sj+aXOn4D+J3eDTJzF/Ci3lMTmflFmvPhn2r/C907I+LLNL+ZYUv72HN+GfjUAualEeLV0KRjEBEfAn6Y mfP9kK53242ZubFn7G7g7Zn5vSWaogaYR8DSsbmeZ59TPpy7um+0p0Y+ZnyXL4+AJamIR8CSVMQAS1IRAy xJRQywJBUxwJJU5P8BGkL5HVhWiA0AAAASUVORK5CYII=",

```
"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",
  "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",
 "____
                      -----\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 thody tr th \{\n'',
      vertical-align: top;\n",
    n''
 "\n",
    .dataframe thead th \{\n'',
      text-align: right;\n",
 **
    }\n",
 "</style>\n",
 "\n",
 " <thead>\n",
    \n",
 "
     <th></th>n",
     CustomerID\n",
     Gender\n",
     <th>Age<math>\n'',
 "
     <th>Annual Income (k$)</th>\n",
     Spending Score (1-100)\n",
    \n",
 " </thead>\n",
   <tbody>\n",
     \n''
 "
     <th>0</th>n".
      1  n''.
      1  n''
 "
      19  n''
      15  n''
 "
     39\n",
    \n",
 "
     \n''
     <th>1\n",
      2  n''
      1  n''
 "
      21  n''
      15  n''
 "
      81  n''
    \n",
    \n",
     <th>2</th>n",
```

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

 $\n''$ ,

```
<th>198\n",
  "
       199  n''
  "
       1  n''
  "
       32  n''
  "
       60  n''
  **
       18  n''
  "
     \n",
      \n'',
  "
       199  \n''
       200  n''
  "
       1  n''
       30  n''
  "
       60  n''
       83  n''
     \n",
  " \n",
  "\n",
  "<p>200 rows × 5 columns<math></p>\n",
  "</div>"
 ],
 "text/plain": [
     CustomerID Gender Age Annual Income (k$) Spending Score (1-100)\n",
                                            39\n",
  "0
                1 19
                              15
  "1
           2
                  21
                              15
                                            81\n",
                1
  "2
           3
                  20
                                             6\n",
                0
                              16
  "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
                  1 32
                                               74\n",
           198
                                126
  "198
                  1 32
                                60
                                               18\n",
           199
  "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",
                0.03846154, 0.00900901, 0.05102041,\n",
        [0.
  "
        [0.
                , 0.09615385, 0.00900901, 0.7755102 ],\n",
        [0.
                         , 0.01801802, 0.39795918],\n",
               0.25
        [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.
  "
        [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",
  11
        [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",
                , 0.13461538, 0.08108108, 0.73469388],\n",
        [1.
  "
        [0.
               , 0.53846154, 0.09009009, 0.04081633],\n",
        [1.
               , 0.25
                         , 0.09009009, 0.73469388],\n",
                , 0.69230769, 0.11711712, 0.13265306],\n",
        [0.
        [1.
                , 0.21153846, 0.11711712, 0.82653061],\n",
```

```
, 0.51923077, 0.11711712, 0.31632653],\n",
     [0.
     [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.
                    , 0.16216216, 0.92857143],\n",
            , 0.59615385, 0.16216216, 0.13265306],\n",
     ſ0.
     [0.
            , 0.05769231, 0.16216216, 0.81632653],\n",
     [0.
            , 0.46153846, 0.17117117, 0.16326531],\n",
     [0.
            , 0.23076923, 0.17117117, 0.73469388],\n",
"
            , 0.34615385, 0.1981982, 0.25510204],\n",
     [0.
     [0.
            0.03846154, 0.1981982, 0.75510204,\n",
     [0.
            , 0.90384615, 0.20720721, 0.34693878],\n",
"
            , 0.11538462, 0.20720721, 0.92857143],\n",
     [1.
            0.57692308, 0.21621622, 0.35714286],\n",
     [1.
     [0.
                      , 0.21621622, 0.6122449 ],\n",
            , 0.25
"
     [0.
            , 0.59615385, 0.21621622, 0.2755102 ],\n",
            , 0.11538462, 0.21621622, 0.65306122],\n",
     [0.
     [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.
                     , 0.25225225, 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
"
            , 0.98076923, 0.26126126, 0.45918367],\n",
     [1.
"
     [0.
            , 0.17307692, 0.27927928, 0.51020408],\n",
     [1.
            0.67307692, 0.27927928, 0.45918367,\n",
                    0.27927928, 0.56122449,\n",
     [1.
     [1.
            , 0.01923077, 0.27927928, 0.55102041],\n",
"
     [0.
            , 0.94230769, 0.28828829, 0.52040816],\n",
     [0.
            , 0.69230769, 0.28828829, 0.59183673],\n",
"
            0.86538462, 0.2972973, 0.51020408,\n",
     [1.
     [1.
                    , 0.2972973, 0.59183673],\n",
     [0.
            , 0.48076923, 0.2972973, 0.5
"
     [0.
            , 0.96153846, 0.2972973, 0.47959184],\n",
     [1.
            , 0.01923077, 0.2972973, 0.59183673],\n",
     Γ0.
            , 0.26923077, 0.2972973, 0.46938776],\n",
"
     [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.
                     , 0.35135135, 0.51020408],\n",
            , 0.75
```

```
, 0.38461538, 0.35135135, 0.55102041],\n",
     [1.
     [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",
     ٢0.
            0.61538462, 0.38738739, 0.45918367,\n",
     [0.
            , 0.96153846, 0.3963964, 0.55102041],\n",
     [1.
                    0.3963964, 0.40816327, n'',
            , 0.
     [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
     [1.
            , 0.57692308, 0.41441441, 0.41836735],\n",
"
            0.03846154, 0.41441441, 0.48979592, n,
     [1.
            0.09615385, 0.42342342, 0.40816327,\n",
     [0.
     [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
     [1.
            , 0.69230769, 0.43243243, 0.45918367],\n",
     [1.
            , 0.96153846, 0.43243243, 0.42857143],\n",
"
            0.92307692, 0.43243243, 0.47959184,\n",
     [1.
     [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.
                    , 0.45045045, 0.47959184],\n",
     [0.
            , 0.01923077, 0.45045045, 0.5
     [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",
     Γ0.
            , 0.09615385, 0.4954955, 0.28571429],\n",
"
     [0.
                     , 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",
            , 0.40384615, 0.5045045 , 0.75510204],\n",
     [1.
"
     [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 ],\n",
```

```
, 0.52252252, 0.06122449],\n",
     [0.
             0.5
     [1.
             , 0.26923077, 0.52252252, 0.73469388],\n",
"
             , 0.01923077, 0.53153153, 0.09183673],\n",
     [1.
     [0.
             , 0.32692308, 0.53153153, 0.7244898 ],\n",
     ٢0.
             . 0.75
                      , 0.54054054, 0.04081633],\n",
"
     [1.
             , 0.26923077, 0.54054054, 0.93877551],\n",
"
             , 0.19230769, 0.54954955, 0.39795918],\n",
     [0.
     [0.
             , 0.26923077, 0.54954955, 0.87755102],\n",
             , 0.13461538, 0.55855856, 0.1122449 ],\n",
     ſ1.
     [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",
                     , 0.56756757, 0.19387755],\n",
     [0.
     [0.
             , 0.38461538, 0.56756757, 0.76530612],\n",
"
     [0.
             , 0.55769231, 0.56756757, 0.15306122],\n",
             , 0.17307692, 0.56756757, 0.89795918],\n",
     [0.
     [1.
             , 0.36538462, 0.56756757, 0.
                                               ],\n'',
     [0.
             , 0.23076923, 0.56756757, 0.78571429],\n",
     [1.
             , 0.30769231, 0.56756757, 0.
     [0.
             , 0.23076923, 0.56756757, 0.73469388],\n",
     [0.
             , 0.73076923, 0.57657658, 0.34693878],\n",
"
             , 0.21153846, 0.57657658, 0.83673469],\n",
     [0.
     [1.
             , 0.01923077, 0.59459459, 0.04081633],\n",
     [0.
                      , 0.59459459, 0.93877551],\n",
"
             , 0.61538462, 0.63063063, 0.25510204],\n",
     [1.
     [0.
             0.34615385, 0.63063063, 0.75510204,\n",
     [1.
             , 0.46153846, 0.63963964, 0.19387755],\n",
"
             , 0.28846154, 0.63963964, 0.95918367],\n",
     [0.
"
     [0.
             , 0.34615385, 0.64864865, 0.26530612],\n",
     [1.
             , 0.26923077, 0.64864865, 0.63265306],\n",
             , 0.42307692, 0.64864865, 0.12244898],\n",
     [1.
     [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",
             , 0.32692308, 0.7027027, 0.90816327],\n",
     ٢1.
"
             , 0.36538462, 0.73873874, 0.31632653],\n",
     [0.
     [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",
"
     ٢1.
             , 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",
               , 0.32692308, 0.94594595, 0.79591837],\n",
        [0.
        [0.
                , 0.51923077, 1.
                                     , 0.2755102 ],\n",
        [1.
               , 0.26923077, 1.
                                     , 0.74489796],\n",
        [1.
               , 0.26923077, 0.40540541, 0.17346939],\n",
                , 0.23076923, 0.40540541, 0.83673469]])"
        [1.
  "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": "iVBORw0KGgoAAAANSUhEUgAAAZgAAAEGCAYAAABYV4NmAAAAOXRFWHRTb2Z
0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAA
CXBIWXMAAAsTAAALEwEAmpwYAAAoU0lEQVR4nO3de5yVZbn/8c93QBE8ICq6URA0KVNTzAkt7eARr
```

DylJkqKShszM612qdkvPESZGSaZJqmJRgoeSrY7NdJMMxUGxQOKSSqCkrI3iiiKHK7fH/c9zYKGcZiZNc+sNd/ 36/W81rPu57Cup73lmvvw3LciAjMzs7ZWU3QAZmZWnZxgzMysLJxgzMysLJxgzMysLJxgzMysLLoWHUBHsc UWW8SAAQOKDsPMrKLMmDHjfyOid2PHnGCyAQMGUFdXV3QYZmYVRdLctR1zE5mZmZWFE4yZmZW FE4yZmZWFE4yZmZWFE4yZmZWFE0xrTZwIAwZATU36nDix6IjMzDoED1NujYkTYdQoWLo0fZ87N30HG D68uLjMzDoA12Ba49xzG5JLvaVLU7mZWSfnBNMaL720buVmZp2IE0xrbLvtupWbmXUiTjCtMWYM9Oixeln 37qnczKyTc4JpjeHDYfx46N8fpLQNGADHHlt0ZGZmhXOCaa3hw+HFF2HVqpRs/vlPeO65oqMyMyucE0xbGjk Snn0WPvShoiMxMyucE0xbkqB371SbmTw5fZqZdVJOMOVwxx1wzDFwxRVFR2JmVhgnmHI45BA4+GA46yz4 xz+KjsbMrBBOMOUgpQ7/9daDk092U5mZdUpOMOXSty9ceincfz9cfnnR0ZiZtTtPdllOJ54I06bBrrsWHYmZWbt zgiknCa68sugozMwK4Say9rBiBXznOzBuXNGRmJm1GyeY9tClCzz9NJx9tt/yN7NOwwmmPdSPKuvWDU46CV auLDoiM7Oyc4JpL1tvDZddBg8+6KYyM+sUnGDa0/HHp5cwL7gAliwpOhozs7IqW4KRtIGkaZIelzRL0vm5/DxJ L0uambfPllxzjqQ5kp6VNKSkfA9JT+Zj4yQpl3eTNCmXPyJpQMk1IyQ9l7cR5XrOdSLBVVelWszGGxcdjZlZWZ WzBrMM2C8idgMGAUMI7ZWPXRoRg/L2BwBJOwHDgJ2BocAVkrrk868ERgED8zY0I48EXo+IHYBLgR/ne20 GjAb2BAYDoyX1KuOzNl+fPrDTTmn/+eeLjcXMrIzKlmAieSt/XS9v0cQlhwE3RcSyiHgBmAMMltQH2CQiHoqI AK4HDi+5ZkLevwXYP9duhgBTI2JRRLwOTKUhKXUMl10GO++cpvc3M6tCZe2DkdRF0kzgNdI/+I/kQ1+T9IS ka0tqFtsA80oun5/Ltsn7a5avdk1ErAAWA5s3ca814xslqU5S3cKFC1v+oC1xzDFpueUTT/SoMjOrSmVNMBGxMiI GAX1JtZFdSM1dHyA1my0AfppPV2O3aKK8pdeUxjc+ImojorZ3795NPEkZ/Md/wM9/Dg8/DGPHtu9vm5m1g3Y ZRRYRbwD3AUMj4tWceFYBvyL1kUCqZfQruawv8Eou79tI+WrXSOoK9AQWNXGvjuXYY+GII+D//T945pmio zEza1PlHEXWW9Kmeb87cAAwO/ep1DsCeCrvTwGG5ZFh25E686dFxAJgiaS9cv/KCcDtJdfUjxA7Crg399PcDR wkqVdugjsol3Us9XOVbbstzJ1bdDRmZm2qnJNd9gEm5JFgNcDkiLhD0g2SBpGarF4ETgGIiFmSJgNPAyuA0yKi vnPiVOA6oDtwZ94ArgFukDSHVHMZlu+1SNKFwPR83gURsaiMz9pyW22Vai9durz/uWZmFUTpD36rra2Nurq6 4gJYtSqtG3PAAQ3DmM3MOjhJMyKitrFjfpO/o1i0KL3hf+KJafZlM7MK5wTTUWyxBfziFzB9OlxySdHRmJm1 mhNMR/LFL8KRR8Lo0TBrVtHRmJm1ihNMRyLBFVfAJpvAySeD+8fMrIJ5yeSOZsst4brrYMMNU8IxM6tQTj Ad0ec+17C/fDmst15xsZiZtZCbyDqyH/0I9tknJRkzswrjBNORffCDMG0aXHxx0ZGYma0zJ5iO7Mgj06zL558PTz xRdDRmZuvECaaju/xy6NUrvYDppjIzqyBOMB3dFlukCTFnzUovYZqZVQiPIqsEX/gC/OMf0Lfv+59rZtZBuAZT KeqTy113uanMzCqCE0wlmTYNDj44DV82M+vgnGAqyeDBcNxxcOGFMHNm0dGYmTXJCabSjBsHm2+eRpW 9917R0ZiZrZUTTKXZfHO46ip4/HH44Q+LjsbMbK2cYCrRYYfB6afDjjsWHYmZ2Vp5mHKlGjeu6AjMzJrkGkw li0iJxk1lZtYBOcFUMglmzIDvfz99mpl1IE4wle5nP4OttkqjypYtKzoaM7N/cYKpdL16wfjx8NRT6f0YM7MOwgm mGnzuc6kGc/HFMH9+0dGYmQFOMNXj0kth6lRPiGlmHUbZEoykDSRNk/S4pFmSzs/lm0maKum5/Nmr5JpzJM 2R9KykISXle0h6Mh8bJ0m5vJukSbn8EUkDSq4ZkX/jOUkjyvWcHcamm8KnP532X3650FDMzKC8NZhlwH4RsR swCBgqaS/gbOCeiBgI3JO/I2knYBiwMzAUuEJSl3yvK4FRwMC8Dc3lI4HXI2IH4FLgx/lemwGjgT2BwcDo0kRW 1W6+Gbbf3mvHmFnhypZgInkrf10vbwEcBkzI5ROAw/P+YcBNEbEsIl4A5gCDJfUBNomIhyIigOvXuKb+XrcA+ + faz RBgakQsiojXgak0JKXqduCB0Lt36pN5992iozGzTqysfTCSukiaCbxG+gf/EWCriFgAkD+3zKdvA8wruXx+LtAgrams + faz RBgakQsiojXgak0JKxqduCB0Lt36pN5992iozGzTqysfTCSukiaCbxG+gf/EWCriFgAkD+3zKdvA8wruXx+LtAgram + faz RBgakQsiojXgak0JKxqduCB0Lt36pN5992iozGzTqysfTCSukiaCbxG+gf/EWCriFgAkD+3zKdvA8wruXx+LtAgram + faz RBgakQsiojXgak0JKxqduCB0Lt36pN5992iozGzTqysfTCSukiaCbxG+gf/EWCriFgAkD+3zKdvA8wruXx+LtAgram + faz RBgakQsiojXgak0JKxqduCB0Lt36pN5992iozGzTqysfTCSukiaCbxG+gf/EWCriFgAkD+3zKdvA8wruXx+LtAgram + faz RBgakQsiojXgak0JKxqquadqs-gf/EWCriFgAkQsiojXgak0sm769Zvto1EbECWAxs3sS91oxvlKQ6SXULFy5sxZN2IJtuCr/6FTz9NJx/ftHRmFknVtYEExErI2IQ0JdUG9mlidP V2C2aKG/pNaXxjY+I2oio7d27dxOhVZiDD4aTT06jyqZNKzoaM+uk2mUUWUS8AdxHaqZ6NTd7kT9fy6fNB/q VXNYXeCWX922kfLVrJHUFegKLmrhX5zF2bOqLefrpoiMxs06qnKPIekvaNO93Bw4AZgNTgPpRXSOA2/P+F GBYHhm2Hakzf1puRlsiaa/cv3LCGtfU3+so4N7cT3M3cJCkXrlz/6Bc1nn07AmzZqW+GDOzApRzNuU+wIQ8Eq wGmBwRd0h6CJgsaSTwEnA0QETMkjQZeBpYAZwWESvzvU4FrgO6A3fmDeAa4AZJc0g112H5XoskXQjUD6 W6ICIWlfFZO6b110+ft94K/fqlFTHNzNqJ0h/8VltbG3V1dUWH0fbeeSetG9OjBzz2GGywQdERmVkVkTQjImob O+Y3+atd9+5w9dUwe3aaddnMrJ04wXQGBx4Io0bBJZfAQw8VHY2ZdRJOMJ3FJZfAttt6Wn8zazdeMrmz2Hhj+ PWvYcGChs5/M7MycoLpTPbdt2F/5Uro0mXt55qZtZKbyDqj66+H2to0wszMrEycYDqjvn1h5kz43veKjsTMqpgT TGe0335w6qlpkbIHHyw6GjOrUk4wndXFF0P//mlU2dKlRUdjZlXICaaz2mgjuPZaeP55uOuuoqMxsyrkUWSd2b77 wrPPwg47FB2JmVUh12A6u/rk8sgj8PbbxcZiZlXFCcbgxRfh4x+HPn2gpgYGDICJE4uOyswqnJvILI0k69IFlixJ3+f OTXOXAQwfXlxcZlbRXIMxOPdcWLFi9bKlS1O5mVkLOcEYvPTSupWbmTWDE4ylWZbXpdzMrBmcYAzGj EkrXpbq0QNOOQXeeKOQkMys8jnBWOrIHz8+vdkvpc+f/zxNJTNkCCxeXHSEZlaBPIrMkuHD/33E2Oabw1FH wdChcPfdsMkmxcRmZhXJNRhbu8MOg8mToa4uJZn6YcxmZs3gBGNNO+IImDQJpk1LE2SamTWTm8js/X3hC3 DffTB4cNGRmFkFcQ3GmmeffWD99WHhwjS67K23io7IzDq4siUYSf0k/VnSM5JmSTojl58n6WVJM/P22ZJrzpE 0R9KzkoaUlO8h6cl8bJwk5fJukibl8kckDSi5ZoSk5/I2olzP2elMmwbXXAOf/7wnxzSzJpWzBrMC+FZEfBjYCzhN 0k752KURMShvfwDIx4YBOwNDgSskdcnnXwmMAgbmbWguHwm8HhE7AJcCP8732gwYDewJDAZGS+pVxm ftPD73OfjNb+CBB1KS8WJlZrYWa00wkg6R1L/k+/clPS5piqTt3u/GEbEgIh7N+0uAZ4BtmrjkMOCmiFgWES8Ac 4DBkvoAm0TEQxERwPXA4SXXTMj7twD759rNEGBqRCyKiNeBqTQkJWutYcPghhvg/vudZMxsrZqqwYwBFg JI+jzwJeBkYArwy3X5kdx0tTvwSC76mqQnJF1bUrPYBphXctn8XLZN3l+zfLVrImIFsBjYvIl7rRnXKEl1kuoWLl y4Lo9kxx0H118PCxb4RUwza1RTCSYiov5P0y8A10TEjIi4Gujd3B+QtBFwK3BmRLxJau76ADAIWAD8tP7Uxm Jooryl1zQURIyPiNqIqO3du9mPZPWGD4fHH0/ryKxYAe++W3REZtaBNJVgJGkjSTXA/sA9Jcc2aM7NJa1HSi4T I+I2gIh4NSJWRsQq4FekPhJItYx+JZf3BV7J5X0bKV/tGkldgZ7AoibuZW1t/fUhAk46CQ49FN55p+iIzKyDaCrB/ AyYCdQBz0REHYCk3Uk1jyblvpBr8rVjS8r7lJx2BPBU3p8CDMsjw7YjdeZPi4gFwBJJe+V7ngDcXnJN/Qixo4B7

cz/N3cBBknrlJriDcpmVgwT77Qd/+lN6MdM1GTOjiRctl+JaSXcDWwKPlxxaAJzUjHvvDRwPPClpZi77LnCspEG kJqsXgVPy782SNBl4mjQC7bSIWJmvOxW4DugO3Jk3SAnsBklzSDWXYfleiyRdCEzP510QEYuaEbO11EknpZr MyJEpyfzud7BBsyq6ZlallP7gb+RAGkH2RkQszt/3JY3emgtcHhHvtVeQ7aG2tjbq6uqKDqPyXX01/Od/wtFHp3nM zKyqSZoREbWNHWtqqpjJpCasxbnGcTPwI2A34Argy20cp1WDL385NZkNHFh0JGZWsKYSTPeIqO8Y/xJwbUT 8NHf6zyx7ZFa5Ro5s2L/jDjjwQOjWrbh4zKwQTY4iK9nfjzyKLI/+Mnt/Tz0FhxwCX/wivFdVLapm1gxNJZg/S5os 6TKgF3Av/GsUmP+1sPe3yy5w+eUwZYqTjFkn1FSCOQO4jTTSa5+IWJ7L/wM4t8xxWbU47TQYNw5uvz1NMb N8+ftfY2ZVoak+mDOAB4Fb8jQsAETEY2WPyqrL6afDqlVw5plw112p2czMql5TCaYvcBmwo6QngL+REs5Dfqf E1tkZZ8CnPgW77150JGbWTtbaRBYR/xURnyA1iX2X9CLjycBTkp5up/ismtQnl7/9LQ1nXrGi6fPNrKI1Zz2Y7sA mpHm+epLm9HqkySvMmjJ9elq07PjjnWTMqtham8gkjSct/rWElFD+BozN66uYtdwZZ6QRZd/5DtTUpGn/u3R5/+ vMrKI01QezLdANeA54mTRD8RvtEJN1Bt/+dur4P/vs9Ob/hAlOMmZVpqnJLofm2Yt3Bj4BfAvYRdIiUkf/6HaK0 arVWWelJPPYY+nTCcasqjRVgyFPff+UpDdIq0UuBj5PXue+7NFZ9TvnnJRcampg0SLo2dOJxqxKrLWTX9J/Sbp J0jzgflJieZa0uuVm7RSfdQY1NfDWW7D33ml02SrPRmRWDZqqwXwV+A7wjbzol1n5bLQRHHMMnH9+Sji/+l X6NLOK1VSCWRwRt7RbJGajR6fay4UXpuRy1VVOMmYVrKkE01vSN9d2sHQZZLM2IaUaTAT84AfQrx98//t FR2VmLdRUgukCbNxegZgBKclccAH07p2azMysYjWVYBZExPntFolZPQm+/vW0v3w53HADnHRSKjezitHcBc fMijF5cloh8/TTU9OZmVWMpmow+7dbFGZrc9xxMHMmXHJJ6vC/7DLXZMwqRFNv8ntKfiueBBdfnEaXjR2bk syllzrJmFUAjwG1jk9KNZgzz4Rrr4UXXig6IjNrBicYqwxSqsHMnAnbb190NGbWDGVLMJL6SfqzpGckzZJ0Ri7f TNJUSc/lz14115wjaY6kZyUNKSnfQ9KT+di4PAknkrpJmpTLH5E0oOSaEfk3npM0olzPae1IakguY8emGZnd8W/ WYZWzBrMC+FZEfBjYCzhN0k7A2cA9ETEQuCd/Jx8bRpq9eShwhaT6WQ+vBEYBA/M2NJePBF6PiB2AS4Ef 53ttRpqMc0/yxJylicwqXERqJrvkkjTdv5OMWYdUtgQTEQsi4tG8vwR4BtgGOAyYkE+bABye9w8DboqIZRHxAj AHGCypD7BJRDyUZ3e+fo1r6u91C7B/rt0MAaZGxKK8QNpUGpKSVToJxo2DU09NAwC++10nGbMOqMnp+tt KbrranbQy5lb1k2dGxAJJW+bTtgEeLrlsfi5bnvfXLK+/Zl6+1wpJi4HNS8sbucaqgQSXX54Sy0UXpSn+f/CDoqMys xJlTzCSNgJuBc6MiDe19uGljR2IJspbek1pbKNITW9su+22a4vLOqqaGvjFL1KS2XrroqMxszWUdRSZpPVIyWVi RNyWi1/NzV7kz9dy+XygX8nlfYFXcnnfRspXu0ZSV6AnsKiJe60mIsZHRG1E1Pbu3bulj2lFqqmBK6+Er341fZ87 t9h4zOxfyjmKTMA1wDNrzLw8Bagf1TUCuL2kfFgeGbYdqTN/Wm5OWyJpr3zPE9a4pv5eRwH35n6au4GDJPX KnfsH5TKrRvW14qefhp12giOPhAEDUvIZMAAmTiwyOrNOq5xNZHsDxwNPSpqZy74LXARMljQSeAk4GiAiZ kmaDDxNGoF2WkSszNedClwHdAfuzBukBHaDpDmkmsuwfK9Fki4EpufzLvDMBJ3AjjvCHnvAbbc1lM2dC6NG pf3hw4uJy6yTUnj0DQC1tbVRV1dXdBjWWv37w0svNV7+4ovtHo5ZtZM0IyJqGzvmN/mtusyb13h5Y0nHzMrKC caqy9pGA26+OfzsZ/DOO+0ajlln5gRj1WXMGOjRY/WyHj1S/8w3vgEf+EB6SfPdd4uJz6wTcYKx6jJ8OIwfn/pcp PQ5fjw88ADcdx988INwxhkp0dx8c9HRmlU1JxirPsOHpw79VavSZ/3osU9/OiWZe+9NCaZbt1T+9tuwbFlBwZpV LycY63z23Rf+8hc45JD0/aKLYOBA+OUv4b33io3NrIo4wVjnJDW8oLnfftC3b5o8c+DA1KTmRGPWak4wZvvuC w8+CHfdBX36wCmnpGRjZq3SLrMpm3V4EgwZAgcdBHffDdvkybfnzEnNaSecAOutV2yMZhXGNRizUhIMHQ of+Uj6/pvfwJe/nIY5//rXsGJFsfGZVRAnGLOmjB4N//3f0KsXnHxySjS//W3RUZIVBCcYs6ZI8PnPw/TpcPvtsMk mqb+m3qpVxcVm1sE5wZg1hwSHHgozZsBPfpLKHngAPvzhtBzAypVNX2/WCTnBmK0LafWpaDbYAL70JdhlF 7jxRicasxJOMGYt9clPwmOPpSlnunaF445LQ569BIYZ4ARj1jo1NXDUUfD44zBpUhoIIKWazJQp7qOxTs0Jxqwt 1NTAF78IJ56Yvk+ZAocdBrvtBrfc4kRjnZITjFk5HHpoGs68fDkcfTTsvntaytmJxjoRJxizcujSBY49FmbNSi9rvvsun HOOE4x1Kk4wZuXUpUtaLmDWrDQFTdeu8NZbaUqa22/3gACrak4wZu2ha1cYMCDtv/giPP88HH441NammQK caKwKOcGYtbdddoHZs9PcZm+8kfprBg+GxYuLjsysTTnBmBWha9c04mz2bLjmGth5Z+jZMx2bPds1GqsKTjBm RVpvvfTuzHXXpe8vvwyDBsEnPpH6bCZOTE1rNTXpc+LE4mI1W0deD8asI+ndG8aNgzFj0rIBNTUNI8/mzoVR o9L+8OHFxWjWTGWrwUi6VtJrkp4qKTtP0suSZubtsyXHzpE0R9KzkoaUlO8h6cl8bJyU1rmV1E3SpFz+iKQBJd eMkPRc3kaU6xnN2tz666ck8ve/w2ab/fuw5qVL4dxzi4nNbB2Vs4nsOmBoI+WXRsSgvP0BQNJOwDBg53zNFZK 65POvBEYBA/NWf8+RwOsRsQNwKfDjfK/NgNHAnsBgYLSkXm3/eGZl1K0bvP5648fmzoUlS9o3HrMWKFuCi Yj7gUXNPP0w4KaIWBYRLwBzgMGS+gCbRMRDERHA9cDhJddMyPu3APvn2s0QYGpELIqI14GpNJ7ozDq2b bdd+7F+/eCss1KfjVkHVUQn/9ckPZGb0OprFtsA80rOmZ/Ltsn7a5avdk1ErAAWA5s3ca9/I2mUpDpJdQsXLmzdU 5m1tTFjVl8aANL3885LL2pecknq+P/97wsIzuz9tXeCuRL4ADAIWAD8NJerkXOjifKWXrN6YcT4iKiNiNrevXs3 EbZZAYYPh/HjoX//NENz//7p++jRMHkyzJkDp58O++yTzv/LX+CuuzzE2TqMdk0wEfFqRKyMiFXAr0h9JJBqGf 1KTu0LvJLL+zZSvto1kroCPUlNcmu7l1nlGT48vfm/alX6LB09tt12MHYsbLFF+j52LBx8MOy6a3qJc9myIiI2+5d2 TTC5T6XeEUD9CLMpwLA8Mmw7Umf+tIhYACyRtFfuXzkBuL3kmvoRYkcB9+Z+mruBgyT1yk1wB+Uys+p28 80wYUIa2nzyyan57Oqri47KOrGyvQcj6UbgM8AWkuaTRnZ9RtIgUpPVi8ApABExS9Jk4GlgBXBaRNSvPXsqaU Rad+DOvAFcA9wgaQ6p5jIs32uRpAuB6fm8CyKiuYMNzCrX+uvDCSfA8cfDPfekPpr6JZyXLoVXXoEddig2Rut UFG6vBaC2tjbq6uqKDsOsbUWk/psrroCvfS1NsPmtb6WZAtRYd6XZupE0IyJqGzvmqWLMqll9EjniiLQezX33pU EBH/94alLzH5hWRk4wZp1Bnz5p2PO8eXD55bBwIVx6aUMCWrGi2PisKjnBmHUmG24Ip52WpqK55ZZUtnBh GgJ9zjmpn8asjTjBmHVGXbrA1lun/bffTk1mF1+cRp6deCI88USR0VmVcIIx6+wGDEi1meeeg698Je0PGgTz57/fl WZNcoIxs2T77dNSAfPmwaRJ0De/43zWWWm9Gr+4aevICcbMVterFxx9dNpftgz++Ec46aRU0/nhD2GRXyuz5n GCMbO169YNHn00JZldd01r0fTrB//zP0VHZhXACcbMmibBgQemJZwffxyOOQZq83t1Dz8MDz1UbHzWYTnB mFnz7borXHstbLVV+n7hhWlWgE98Am69tWFqGjOcYMysNSZNgp//HF59FY46Cj74Qfjtb4uOyjoIJxgza7mNNk

pznNW/uLnVVvB//5eOvfsuLFhQbHxWKCcYM2u9Ll3gyCPhb39LMwUA3HBDmiHgpJPgySdh4sQ0Eq2mJn1On FhkxNYOyjZdv511UjX579b994dRo9LiZ9ddl8pXrUrH5s5Nx2D1RdSsqrgGY2blsf32aWLNefNg000bkku9pUvhm9 9MU9VYVXKCMbPy2mwzWLy48WOvvQY9e8LgwSnZeMhzVXGCMbPy23bbxsu33BLOPht69IArr4QHHkjl//u /adnnX/86zZHmdWsqkvtgzKz8xoxJfS5LlzaU9egBY8c29MG8917aAP7xD7j99pRgII1O22cfOO882GWXdg3dWs 41GDMrv+HDYfz4NKpMSp/jx6/ewb/++mnYM8Cee6Z1ambNgquuSjMJzJiRRqsB3HgjHHAAnH8+3HMPvPVW +z+TvS+Fq54A1NbWRl1dXdFhmFlz/Pa38JOfpKlrIlLi+ehH4S9/ge7dYflyWG+9oqPsFCTNiIjaxo65iczMKs9xx6 Vt8eI0MOCvf4Xnn0/JBdJ8abNmpWa1ffaBT34SPvCBhiWirV04wZhZ5erZE4YOTVupIUPSvGi//32aOw3SOXfe mfaffTYlnK7+J7Cc/L+umVWfU05J26pVMHt2quHU9++8915asbNL17RUdH0tZ889G86xNuEEY2bVq6YGdtopbf Ui0swCDzyQEs/556eyCy+E730P3nwT/vSnlHS23LKw0KtB2UaRSbpW0muSniop20zSVEnP5c9eJcfOkTRH0rOS hpSU7yHpyXxsnJQaUSV1kzQplz8iaUDJNSPybzwnaUS5ntHMK1C3bqmP5vLLYeZMeP311HR27LHp+F//muZV 22or+NCHYOTIIJDqV/L0nGrNVrZRZJI+BbwFXB8Ru+Syi4FFEXGRpLOBXhFxlqSdgBuBwcDWwJ+AD0bESk nTgDOAh4E/AOMi4k5JXwV2jYivSBoGHBERx0jaDKgDaoEAZgB7RMTrTcXrUWRmBqRloh99NCWaBx6ABx 9MyeXxx9OknSNHpnPq9ejx70OuO5GmRpGVrQYTEfcDay7efRgwIe9PAA4vKb8pIpZFxAvAHGCwpD7AJhHx UKRMeP0a19Tf6xZg/1y7GQJMjYhFOalMBdboATQzW4tu3VLfzLe/DVOmpPdxnnoqveB57rmrJxdIL49+5SuwY kUx8XZg7d0Hs1VELACIiAWS6hs4tyHVUOrNz2XL8/6a5fXXzMv3WiFpMbB5aXkj16xG0ihgFMC2a5vKwsw6t 5oa2HnntP/SS42f89ZbDSPSTjghzUTwsY+lpaU/9jEYOLBhlulOpKM8cWOD06OJ8pZes3phxPiIqI2I2t69ezcrUDP rxNb2h2hp+Y47pvdtxo+H449P3/fbr+H4Pfekd3Y6wUvu7Z1gXs3NXuTP13L5fKBfyXl9gVdyed9Gyle7RlJXoCep SW5t9zIza50xY1KfS6kePeCHP2z4/t3vpv6bN99MfTbXXpua0CC9m3PooekdnC22SO/rfO978PDDVKP2TjBTgPp RXSOA20vKh+WRYdsBA4FpuTltiaS9cv/KCWtcU3+vo4B7cz/N3cBBknrlUWoH5TIzs9Zpzpxq9bp2Tf02J50Ew4 alMgnuvz/NHH3EEfDPf8JFF8Hd+Z+oN95ICejCC9PItoUL2+3RyqFsfTCSbgQ+A2whaT4wGrgImCxpJPAScDR ARMySNB14GlgBnBYRK/OtTgWuA7oDd+YN4BrgBklzSDWXYfleiyRdCEzP510QEWsONjAza5nhw1s+Yqym BvbYI231li5tmEX6lVdgzhy4446GJrT+/dOEn0OGpMXZli9PC7hVAE92mXmYspl1GG++CY89BtOnQ11dGr32k Y+kWaSPOy4NGqitbRhEMHhwGv1WgKaGKTvBZE4wZtbhzZ4Nv/tdQ+KZlwfMvvQS9OsHf/wj/P3vKensthtssE HZQyrkPRgzM2tjO+4I55wDt92Wkso//wl/+AP0zWOhbr0VTj8d9toLNt44LWFw2mkNzW1rVijKPCuBazCZazBm VvEiUq2mri5t06enPpv77kvHDzkkLUddW5v6fa6/Ht59t+H6FsxK4CayZnCCMbOqd/75cO+9aSqcta0C2r8/vPhis2/ pBNMMTjBm1mmsXJIW/Gzs338pLXPQTO6DMTOzBl26NG9WglZygjEz64zWNivBmDFt9hNOMGZmndG6zE P/bRuEUqVqeA/wsHVW1PEu1PAe07ln6R0Sja847wbQRSXVrmy6hklTLc4CfpaOqlmeplueA8j2Lm8jMzKwsnG DMzKwsnGDazviiA2gj1fIc4GfpqKrlWarlOaBMz+I+GDMzKwvXYMzMrCycYMzMrCycYFpBUj9Jf5b0jKRZks 4oOqaWkrSBpGmSHs/Pcn7RMbWGpC6SHpN0R9GxtIakFyU9KWmmpIpeclXSppJukTQ7/zfz8aJjaglJH8r/96jf3p R0ZtFxtYSkb+T/3p+SdKOkDdr0/u6DaTlJfYA+EfGopI2BGcDhEfF0waGtM0kCNoyItyStB/wVOCMiHi44tBaR9E 2gFtgkIj5fdDwtJelFoDYiKv6FPkkTgAci4mpJ6wM9IuKNgsNqFUldgJeBPSOiNS9qtztJ25D+O98pIt6RNBn4Q0Rc 11a/4RpMK0TEgoh4NO8vAZ4Btik2qpaJ5K38db28VeRfH5L6Ap8Dri46FkskbQJ8CrgGICLeq/Tkku0P/KPSkkuJr kB3SV2BHsArbXlzJ5g2ImkAsDvwSMGhtFhuVpoJvAZMjYhKfZafAd8BVhUcR1sI4I+SZkgaVXQwrbA9sBD4d W66vFrShkUH1QaGATcWHURLRMTLwCXAS8ACYHFE/LEtf8MJpg1I2gi4FTgzIt4sOp6WioiVETEI6AsMlrR LwSGtM0mfB16LiBlFx9JG9o6IjwIHA6dJ+lTRAbVQV+CjwJURsTvwNnB2sSG1Tm7mOxS4uehYWkJSL+AwY Dtga2BDSV9qy99wgmml3F9xKzAxIm4rOp62kJsu7gOGFhtJi+wNHJr7Lm4C9pP0m2JDarmIeCV/vgb8DhhcbEQt Nh+YX1IrvoWUcCrZwcCjEfFq0YG00AHACxGxMCKWA7cBn2jLH3CCaYXcMX4N8ExEjC06ntaQ1FvSpnm/ O+n/+WYXGlQLRMQ5EdE3IgaQmi/ujYg2/ausvUjaMA8eITcnHQQ8VWxULRMR/wTmSfpQLtofqLjBMGs4lgp tHsteAvaS1CP/W7Y/qR+5zXRty5t1QnsDxwNP5r4LgO9GxB+KC6nF+gAT8qiYGmByRFT0EN8qsBXwu/TfPl2B 30bEXcWG1CqnAxNz09LzwEkFx9NiknoABwKnFB1LS0XEI5JuAR4FVgCP0cZTxniYspmZlYWbyMzMrCycY MzMrCycYMzMrCycYMzMrCycYMwKlt9BeiRPofLJZpz/mZbOEi3pzDzE1qzsnGDMirc/MDsido+IB 8r8W2eSJjVstvxulNk6c4IxayZJA/I6Jr/Ka2j8Mc96gKRBkh6W9ISk3+V5nta8vr+ke/I590jaVtIg4GLgs3ltke5rXPMx SX/L6/RMq3+zv+T4eZL+q+T7UznODSX9T77uKUnHSPo6ac6pP0v6cz7/IEkPSXpU0s15Xr36dWi+L+mvwNFt+ 7+kdRZOMGbrZiDwi4jYGXgDODKXXw+cFRG7Ak8Coxu59nLg+nzORGBcRMwEvg9MiohBEfFO/cn5jfdJpH V5diNN3/POv9+2UUOBVyJit4jYBbgrIsaRpmPfNyL2lbQF8D3ggDyhZh3wzZJ7vBsR+0TETc38TbPVOMGYrZs XclKAtMDcAEk9gU0j4i+5fAJp7ZM1fRz4bd6/AdjnfX7rQ8CCiJgOEBFvRsSKZsb5JHCApB9L+mRELG7knL2A nYAH81RHI4D+JccnNfO3zBrlucjM1s2ykv2VQPe1ndgM7zdPk5pxzgpW/0NxA4CI+LukPYDPAj+S9MeIuKCR+ 0+NiGPXcu+33+e3zZrkGoxZK+XaweslI8COB/7SyKl/I83wDDCctFxtU2YDW0v6GICkjfPKg6VeJE97L+mjpLU9 kLQ1sDQifkNaVKp+avwlQH0/zsPA3pJ2yNf0kPTB94nJrNlcgzFrGyOAX+YhwGubKfjrwLWSvk1a3bHJ2YQj4j1 JxwA/z53/75D6YUrdCpyQm7imA3/P5R8BfiJpFbAcODWXjwfulLQg98OcCNwoqVs+/r2Se5i1imdTNjOzsnATm ZmZlYUTjJmZlYUTjJmZlYUTjJmZlYUTjJmZlYUTjJmZlYUTjJmZlCX/B5VDraYFskFEAAAAAElFTkSuQmCC

<sup>&</sup>quot;text/plain": [

<sup>&</sup>quot;<Figure size 432x288 with 1 Axes>"

```
},
 "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": [
 "
     0, 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, 2, 3, \ln",
       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 thody tr th:only-of-type \{\n'',\n''\}
        vertical-align: middle;\n",
     n''
  "\n",
     .dataframe thody tr th \{\n'',
        vertical-align: top;\n",
  "
     n''
  "\n",
     .dataframe thead th \{\n'',
        text-align: right;\n",
     n''
  "</style>\n",
  "\n",
  " <thead>\n",
      \n",
       <th></th>\n",
  "
       CustomerID\n",
       Gender\n",
       <th>Age</th>\n'',
       <th>Annual Income (k$)</th>\n",
```

```
Spending Score (1-100)\n",
     \n",
  " </thead>\n",
    <tbody>\n",
      \n''
  **
      <th>0</th>\n",
       1  n''
       1  n''
       19  n''
  "
       15  n''
  "
       39  n''
     \n",
  "
      \n'',
  **
      <th>1\n",
       2  n''
  **
       1  n"
       21  n''
       15  n''
  "
       81  n''
     \n",
  "
      \n'',
      <th>2</th>n",
       3  n''
  "
       0  n'',
       20  n''
  **
       16  \n''
       6  n''
     \n",
  " \n",
  "\n",
  "</div>"
 ],
  "text/plain": [
  " CustomerID Gender Age Annual Income (k$) Spending Score (1-100)\n",
  "0
              1 19
                                        39\n",
         1
                           15
  "1
         2
              1 21
                           15
                                        81\n",
  "2
         3
                                         6"
              0 20
                           16
 "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 thody tr th:only-of-type \{\n'',
       vertical-align: middle;\n",
     }\n",
  "\n",
     .dataframe thody tr th \{\n'',
       vertical-align: top;\n",
     n''
  "\n",
     .dataframe thead th \{\n'',\
       text-align: right;\n",
     }\n",
  "</style>\n",
  "\n",
  " <thead>\n",
     \n",
        \n'',
  "
      CustomerID\n",
      Gender\n",
      <th>Age\n",
  "
      <th>Annual Income (k$)</th>\n",
      Spending Score (1-100)\n",
      <th>clust\n",
     \n",
  " </thead>\n",
    <tbody>\n",
      \n'',
      <th>0</th>n",
  "
       1  n''
       1  n''
  "
       19  n''
       15  n''
       39  n''
       1  n''
```

```
\n",
  "
     \n'',
  "
     <th>1\n",
  "
      2  n''
  "
      1  n''
  **
      21  n''
  "
      15  n''
  **
      81  n''
  "
      1  n''
  "
    \n",
  "
     \n''
  "
     <th>2</th>n",
  "
      3  n''.
  **
      0  n''
  "
      20  n"
  **
      16  n''
  "
      6  n''
  **
      1  n''
  "
    \n",
  **
     \n'',
  "
     <th>3</th>n'',
  "
      4  n''
  "
      0  n''
  "
      23  n''
  "
      16  n''
  "
      77  n''
  "
      1  n''
  **
    \n",
  "
     \n'',
  "
     <th>4</th>n",
  "
      5  n"
  "
      0  n''
  "
      31  n''
  **
      17  n''
  "
      40  n''
  "
      1  n''
  "
    \n",
  " \n",
  "\n",
  "</div>"
 ],
 "text/plain": [
    CustomerID Gender Age Annual Income (k$) Spending Score (1-100) clust\n",
  "0
         1
             1 19
                                      39
                                           1\n",
                          15
  "1
         2
             1 21
                          15
                                      81
                                           1\n'',
  "2
         3
             0 20
                                      6
                                          1\n'',
                          16
  "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 thody tr th \{\n'',\
       vertical-align: top;\n",
     }\n",
  "\n",
     .dataframe thead th \{\n'',
       text-align: right;\n",
     n''
  "</style>\n",
  "\n",
  " <thead>\n",
     \n",
      <th></th>\n",
  "
      CustomerID\n",
  "
      Gender\n",
      <th>Age</th>\n'',
      <th>Annual Income (k$)</th>\n",
      Spending Score (1-100)\n",
  "
      <th>clust\n",
     \n",
  " </thead>\n",
    <tbody>\n",
      \n'',
  "
      <th>195</th>\n",
       196  n''
  "
       0  n''
       35  n''
  "
       120  n''
  **
       79  n",
  "
       3  n''
     \n",
  "
      \n''
      <th>196</th>\n",
  "
       197  n''
       0  n''
       45  n''
      126\n",
```

```
 28  n''
 "
      2  n''
 "
    \n",
 "
     n''
 "
     <th>197\n",
 **
      198  n''
 "
      1  n''
 "
      32  n''
 "
      126  n''
 "
      74  n''
 "
      3  n''
 "
    \n",
 "
     \n'',
 "
      198  \n'',
 "
      199  \n''
 "
      1  n''
 "
      32  n''
 **
      60  n''
 "
      18  n''
 "
      2  n''
 "
    \n",
 "
     \n''
 "
     199\n",
 "
      200  n''
 "
      1  n''
 "
      30  n''
      60  n''
 **
      83  n''
 "
      3  n''
    \n",
 " \n",
 "\n",
 "</div>"
 ],
 "text/plain": [
     CustomerID Gender Age Annual Income (k$) Spending Score (1-100) \\\n",
 "195
                                          79 \n",
          196
                0 35
                            120
                                          28 \n",
 "196
         197
                0 45
                            126
 "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 thody tr th:only-of-type \{\n'',\n''\}
    vertical-align: middle;\n",
  }\n",
"\n",
  .dataframe thody tr th \{\n'',
    vertical-align: top;\n",
  }\n",
"\n",
  .dataframe thead th \{\n'',\
    text-align: right;\n",
  }\n",
"</style>\n",
"\n",
" <thead>\n",
  \n",
   <th></th>\n",
"
   Gender\n",
**
   <th>Age</th>\n'',
"
   <th>Annual Income (k$)</th>\n",
   Spending Score (1-100)\n",
"
  \n'',
" </thead>\n",
" <tbody>\n",
"
   \n''
   <th>0</th>n",
    1  n''
"
    19  n''
    15  n''
"
    39  n''
"
  \n",
"
   \n'',
"
   <th>1\n",
    1  n''
    21  n''
"
    15  n''
    81  n''
"
  \n",
"
   \n''
   <th>2</th>n",
"
    0  n''
    20  n''
"
    16  n''
"
    6  n''
"
  \n",
"
   \n'',
"
   <th>3</th>n",
    0  n''
"
    23  n''
"
    16  n''
"
    77  n''
  \n",
   \n'',
   <th>4</th>n",
```

```
 0  n''
  "
       31  n"
  "
       17  n''
       40  n''
     \n",
  " \n",
  "\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
                                     77\n".
                       16
  "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 thody tr th:only-of-type \{\n'',
       vertical-align: middle;\n",
     }\n",
 "\n",
     .dataframe thody tr th \{\n'',
       vertical-align: top;\n",
  "
     n''
  "\n",
     .dataframe thead th \{\n'',
       text-align: right;\n",
     }\n",
  "</style>\n",
  "\n",
  " <thead>\n",
     \n",
```

```
<th></th>\n",
"
   Gender\n",
"
   <th>Age</th>\n'',
"
   <th>Annual Income (k$)</th>\n",
    Spending Score (1-100)\n",
**
   \n",
"
 </thead>\n'',
 <tbody>\n",
    n''
"
   <th>195</th>\n",
"
    0  n''
**
    35  n"
"
    120  n''
"
   79\n",
"
   \n'',
"
    \n'',
"
   <th>196</th>\n",
    0  n"
"
    45  n''
**
    126  n''
"
    28  n''
"
   \n",
"
    \n'',
**
    197  \n''
    1  n''
"
    32  n''
    126  n''
**
    74  n''
"
   \n",
"
    n'',
"
   <th>198\n",
"
    1  n''
"
    32  n''
**
    60  n''
"
    18  n''
"
   \n",
"
    \n'',
    199  \n''
"
    1  n''
    30  n''
"
    60  n''
"
    83  n''
  \n",
" \n",
"\n",
"</div>"
],
"text/plain": [
   Gender Age Annual Income (k$) Spending Score (1-100)\n",
"195
       0 35
                   120
                                79\n",
"196
       0 45
                   126
                                28\n",
"197
                   126
       1 32
                                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 Evaluaation 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": {
"kernelspec": {
 "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
```