```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "755c232d",
   "metadata": {},
   "source": [
    "# Assignment 4"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "3ae24645",
   "metadata": {},
   "source": [
    "# 1.Download the dataset"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "1b9fe6dd",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "4253df83",
   "metadata": {},
   "source": [
    "# 2. Load the dataset"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "23f5edf3",
   "metadata": {},
   "outputs": [],
   "source": [
    "data = pd.read_csv(r\"Mall_Customers.csv\")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "f8a79274",
   "metadata": {},
   "outputs": [],
```

"source": [
"data.head();"
]
},
{
"cell_type": "markdown",
"id": "e57b2d3d",
"metadata": {},
"source": [
"# 3. Perform Below Visualizations."
]
},
{
"cell_type": "markdown",
"id": "8dc94c33",
"metadata": {},
"source": [
"### · Univariate Analysis"
]
},
{
"cell_type": "code",
"execution_count": 4,
"id": "16dee427",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"(array([31., 19., 34., 29., 16., 26., 15., 10., 6., 14.]),\n",
" array([18. , 23.2, 28.4, 33.6, 38.8, 44. , 49.2, 54.4, 59.6, 64.8, 70.]),\n",
" <BarContainer object of 10 artists>)"
]
},
"execution_count": 4,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAAXAAAAD4CAYAAAD1jb0+AAAAOXRFWHRTb2Z0
d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAAC
XBIWXMAAAsTAAALEwEAmpwYAAAOQ0lEQVR4nO3dYYhl9XnH8e8vampiLK51lCG6nSaITZA6psPWYg
mJJmGNIeoLIULTpUjXFwoKlrL1TUyhYCEmj7aja5dxZBuE5Pt7qRpqrZgU
Z++uGebYZzx3pl770z+d74fuJxz/vfcOc/D7P4487/n3JuqQpLUnndsdAGSpLUxwCWpUQa4JDXKAJekRhngktSoE
9fzYGeccUbNzMys5yElqXn79+//cVVNLR3vG+BJTgaeAH6/n+/qvpmlqapmH3+5nzzczMMD8/v9r
aJW1TS/Jvy40Pcgb+GnBJVb2a5CTgySR/2z33par6vqiKlCQNrm+AV+9+8M9Ze6n3j7h3f/SNIGG+hNzCQnJDkAH
AP2VtVT3VM3JHkmyd1JtoyrSEnSWw0U4FX1RlXNAmcD25OcB3wBuAF2g/yn1jNGZ2VpOjq5KcD9wBvB+YBY4Aty/32iQ7k8wnmV9YWFhuF
0nSGqzqMsKq+inwPLC9qo52wf4mcCewbYXX7K6quaqam5qaWmr6F7mJPz1RWUD95oSaaSnnNNatvwv4GPB8kulFu10FHB
xLhZKkZQ1yFco0cnhb4nBJ/BmUiLiiXsg3JjI/6SsIH9oy+9HEP7lDpfNcgTHfsqFy13jHFq6a+R7LKIimrVSqSqiicSGgWSLcJxB0QnGrnZ5rOmTc0z1ZXHn51BmzrdUl2TACjBoiIUhAd0axRPuiQIrLrfVGj/OvIy8d
WZ2PbIhxz182+UbclxJq+MZuCQ1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNcoAl6RGG
eCS1CgDXJIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDWqmW/k2ahvpwG/oUbSzyfPwCWpUQa4JDWqb4An
OTnJt5N8L8mzST7XPf5A5eJvmbwqiK/S+q1+GKhQi0wbmNtOeUmj7f4wOMcNwo9fYnRiMH6G4TN3IqA+dxSeeHPzmNmu1+qHe7QnrfcLdOX+nv0ZulSxAMR+SGwxTp0xGUj+nCvGeIwwLTVS7kAXDbfEi7wjTY4JTYtFEeKp9vXQ6t/c+20X3HT8vubmKGqnhgNN3N/Y38kmHFZcJZ6FuOyaS6Uu5UHK/w6
VbPA2cC2JOcPeoCq2l1Vc1tRypKA4BkmjRvuLyDcTUOUVnX1a1NzU1tcYyJUlLreoqlKr6KMdmQqJZPaJoAcZJIMDf4PGtL8m9OYWVcq44Zh0UDQS31STE5mmQUx2ynpQ9PF3HLBCSGX9ZeOBMq8zV5
xmJ4v9ALA54g

HHgJ2dLvtAB4cU42SpGUMcifmNLAnyQn0Av/eqno4ybeAe5NcC3wfuHqMdUqSlugb4FX1DHDhMuP/AVw6jqI
kSf0181koWj9+7ozUBm+ll6RGGeCS1CgDXJIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA
1ySGmWAS1KjDHBJapQBLkmNMsAlqVEGuCQ1ygCXpEb5jTwSfguR2uQZuCQ1ygCXpEb1DfAk5yR5LMmh
JM8lubEbvzXJD5Mc6B6fH+5kqTjBpkDfx24uaqeTnIqsD/J3u65L1XVF8ZXniRpJX0DvKqOAEe69VeSHALeO
+7CJElvb1Vz4ElmgAuBp7qhG5I8k+TuJFtGXZwkaWUDB3iS9wD3AzdV1cvAAHcD7gVl6Z+i3r/C6nUnmk8wvL
CwMX7EkCRgwwJOcRC+876mqBwCq6mhVvVFVbwJ3AtuWe21V7a6a6quaqam5qaGlXdkrTpDXIVSoC7gENV9c
VF49OLdrsKODj68iRJKxnkKpSLgc8czyY50IaZ+dBYo4DBw3RjqkyStYJCrUJ4EssxTj46+HEnSoLwTU5Ia
YBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA1ySGmWAS1KjDHBJapQBLkmNMsAlqVEGuC
Q1ygCXpEYZ4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNapvgCc5J8l8ljSQ4leS7Jjd346Un2JnmxW24Zf
7mSpOMGOQN/Hbi5qj4AXARcn+SDWwC5gX1WdC+zrtiVJ66ult/BTgEvBe4AtjT7bYHuHJMNUqSlr
GqOfAkM8CFwFPAWVV1BHohD5y5wmt2JplPMr+wsDDBkuZQkK4wYO8CTvAe4CTvAe4Hbqqqlwd9XVXtrqq5qpqbmpqbmpp
aS42SpGUMFOBJTqIX3vdU1U1QPd8NEk093z08Cx8ZQoSVrOIFehBLgLOFRVX1z01F5gO7ftkA1pa7VX1z01EPAjm59B/Dg6MuTJK3kx
AH2uRj4LPBskgPd2C3AbcC9Sa4Fvg98akkhBXNWBWevnS05UiSBjXIGfimN7PrkY0uQZLewlvpJal
RBrgkNcoAl6RGGeCS1CgDXJIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA1ySGmWAS1
KjDHBJapQBLkmNMsAlqVF+I49+rvjtR9LgPAOXpEYZ4JLUqL4BnuTuJMeSHUtmuSHyFw0dmuSHyY50D0+Od4yJUlL
DXIG/hVg+zLjX6qq2e7x6GjLkiT10zfAq4CfrUISkaRWGuQRlhiS/A8wDN6ArVu3DnE4aTJt
1JU3h2+7fEOOq9FZ65uYdwDvB2aaQKTEsYsZUsKkao6WlVvVvNWbwJ3AttGW5YkaTlDBniSbYs2Pwkc
qZ00BnmR60eZVwMGV9pUkfOfAkXwM+BpCBwVGrhtfiZKk5fnZnhu8ZQiyRpF
bwTU5Ia5YdZSdo0NvLD0sCx8aZn4JLUKANckhplgEtSowxwSWqUAS5JjTLAJalRBrgkNcoAl6RGGeCS1CgDX
JIaZYBLUqMMcElqlAEuSY0ywCWpUQa4JDXKAJekRhngktQoA1ySGmWAS1KjDHBJalTfAE9yd5JjSQ4uGjs9
yd4kL3bLLeMt8Bn4F8Bti8Z2wXsq6pzgX3dtiRpHUBgD3Td+h7gytGWJUnqZ1CBj/VVl7GtGVJUnq58Q1vu6sqjoC
UFVHkpy50o5JdgI7AbZu3TrE4aTJtLrDXrGS0k/mnFtGlwMKPA/cnGjFhTBuHDgJNxLaXJLXDFv4PyAxNTVk1zF1UXLxeU4t8fYLjNA2N8wlakcGomf11kMJF4vCaWVvVqZJTSD9G1Hg6
UhFVHkpy50o5JdgI7AbZu3brGw0katZldj2zYsQ/fbvUyxKpp0LtwSjEm0QhQFvFm1AmNebyAC7r9Yz3zywMKr1L9Zhc1hE9pH3vv
7Qkq9yRFasJ71s
4X6pp8UBmMa/5fnFz+Gv2mxxOWK69TvtiQOYzt0wcM8UpMH5T5q+zNR0azdB9OuJUCk2pvMnCLJlXrxSRwQkYtYV5d6rIt1SZ96lUc2PG
UUm4EoVdvK0ASoY7XpEhgI5jIy7AKjJvGupplNdpNVBStAR+vFaD+yOp3AHajGzGWE+WjSi6NNatxMi0ZAyFrbmjR3LKn+7PeJd2Z5zh3
ZXgb4gwRd/nQgwcDeHfBF7lIkKyGd1Gq7Z8VzGO+ZW1S5pMHFmzyDjoWXj/PHSUvrGyxUBNBERYGBGHNRnLO+/f2ATKB7J31OHpMLaxG8
K73kL1Ip0K0Sd6mYGkd/0DLW1p0MXDhjJEX8rPUAkqJSBrdd/lCo3JUvwM35zPBsOuqZbZ1L4ssd7b1BoKb2xTozxfJ6vUThx7zcUm4OP
ZMx9JEb80wIXkuf+9dvB+WIPaWyHeRnNz33ZkjDq/0XlRRHs5Vg5WmVGqdSDXE2Qf2cf9XkEvKmJN5ih8TdtGxuGnLX8E3Uu/xAjLwoZD
K7rEb+dnNVrvcwHd9g6aCWj83uKuTXj2m70OHB9Oc0SIn8SntvBNo27TV0M/nLxPLD3dC3qCpPvdy8A1W6eoEBcnO+mb6vV3bGS1rMO0E
eCeu1qv6wqs6uqhl6f4L+fVX9NhPWJ0CSU5Kcenyd3pzpQSas16r6d+AHSc7rhi4F/oUJ63OJa/jB7Ta/

],
"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": "iVBORw0KGgoAAAANSUhEUgAAAXoAAAD4CAYAAADiry33AAAAOXRFWHRTb2Z0d
2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACX
BIWXMAAAsTAAALEwEAmpwYAAAVT0lEQVR4nO3dfYxU133G8efxsLEX5wVcFgvjdXArhGObNCQbY8d
S5MihWElqNmmiguKKNJ0Sd4bUkIWBKuwKkgmEiT6kK7QdzeYGwjEJiwzybdbIFPgfvZi6djmKNmneFx9BdF/aSCWkNSjxs6CgzVHbWKVPwiZJRkp7Z0xqqaPs8AhEsncaGHGqd1prFhpV9gd2IxYfGAFdJMZwr1RckFRbB
psXktTe8LL8+sdeYFjPsDN7Z2bnzH4/Epq5v3vv3CP+ePbOmXPPcUQIANC6LprsBgAA6ougB4AWR9ADQIsj6
AGgxRH0ANDipk12AyRp1qxZMW/evluBgAkZefOnT+T+NiuI7xjmuKoJ83b576+/snuxkAkBTbP67kuHG7bmw/Y
Puo7WdL7Puk7bA9q6i2xvYLtvfaXlpdswEAtVZJH/3XJN02tmiL9uLatdKWi7puuycL9su1KSlAIAJGTfoI+I
JST8vsetLkj4lqHoyI4xHxoqQXJN1QiQ4YCACZmQqNubNubN7NaqnM9uuaTAinh6za66kA0XbB7Naqc9YZbvfdv/Q0
NBEmgEAqEDVQW97uqS1kj5TaneJWsnIoRXRXqR8vmR1JdSvCJiY0R0RURXR8e4PxoDCZoIqNufkfS1ZZKeti1JV0r6ke9diBHoH31
l07JWSDuVtZDnregfU03dAIxEq2FquxuFPruxfW63IAkKSq7+gjYiAiZkfEvIiYiYp9Fwf3NE/ETSFknLbV9s+2pJ8yV9r
9WdMWZ9b1DmjzjvOayWbfHInQ5h37ta53oB6XA4BkVTK8skfSf0paYPug7TvLHRsuU9JOSq+2gjYiAiZkfEvIiYp9Fwf3NE/ETSFknLbV9s+2pJ8y
scV6+g5UVQeAqWrcrpuIWDHO/nljtu+VdG+Zo1vpMw8+uXqADBVJTvXTcGlfvctXweAqSrZoF+xuLOqOgB
MVU0x181EnBldw6gbALGwN8OasV1dXcGkZgBQds7MwAVpFDgRLgGPdKHxG5AbALHEEPAC2OoeAeAFpfsXDeS1LtrUBu27tWhY8O6Yka7Vi9DoO5FJZeoBYCm0sj8Sjb
oe3cNas3D+ObzrQ7W...+ObquyeCxYa15eHR1KcIeQDNrdH4l23WzYeves/9JZwyfHNGGrXsnqUUAUJlG51eyQX/o2HBV
dQBoFo3Or2SD/ooZ7VXVAAaBZNDq/kg361UsuSOdbjzcFKsm9bUAqQEGTZ4hanvRMGNDGdf6s71qakXGM7szVnnrtoUOfb9IFGTghaW5sq
FClMAkChWmAIASCLoAaDljdtHb/sBSe+RdCwibsxqo3dLWiHppKF+ZXcbhVZWb2T1smdrY1/x7F2vtDe3
9Q6op++ARiJUsLVicagw7bZDk7tGtfLRY3B/RJZkjklPFMGRWaY
oe3cNas3DAxo+Obz9qaX1a15HH1R13EwLXSnb/DpbuqyeCxYa15eHR1KcIeQDNrdH4l23WzYeves/9JZwyfHNGGrXsnqUUAUJlG51eyQX/o2HBV
dQBoFo3Or2SD/oOZ7VXVAAaBZNDq/kg361UsXqL2tcF6tva2g1UsXTFKLAKAyjc6vZH+MPfODBaNmdn
FClMAkChWmAIASCLoAaDljdtHb/sBSe+RdDQirs9qGyT9vqQkv5b0h9hxHLs3xpd0akfTRiNhan6YDQLrW
9Q6op++ARiJUsLVicafWdy+sy7UquaP/mqTbTBcsBx5Swz4CDAhDGKPiV7mggJ6Uf8dpjtZJZeoBYCm0sj8Sjb
W+kIxHavGO/1vUO1OV64wZ9RDwh6edjat+PiFPZ5g5JV2bvl0l6MCKOR8SLkl6QdEPZo0hSvl6eFKKOR8SLKl6QdEMN2wsAyevpO1BVPa9a9
NF/WNJ3s/dzMH/dz/JRW39GBWewXbq+S0q+xWsuURsTEiuiKiq6
OjI08zACApBZeKyvvL1vCYc9LZXavRH2g/GuBvv5kjD2g2rNaGgt32bpE9uLoj0iXi7at1Xj0IXi7at
UXSctsX275nOTn0a1vBaYL7vE/HmPRD2T+/nZ6+g5dPJd/1bEZqm3CdQbcfBfvYIukvXoqbwdXcLv1m
+z4tRrtzs8l6eMR8d7xnkkWT8YCQPUqKKRAAIFFMsgQyFyU4CbAtEgOBUsTh4hVgcHECqWBy8QiwODiBVLA5eIRYABSxe
LgFWJxcACpYnHwCrE4OIBUsTh4hVgcHECqWBy8QiwODiBVLA5eIRYABSxe

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/29uvUdH5E5ixODiAFDQ6v5J9YIrFwQGkqtH5lewduPQCgMrmC3vaf2t5t5t+1nbPbYvsX2Z7W2292W
vM2vV2GK9uwa15uEBDR4bVkgaPDasNQ8PqHfXYD0uBwA10+j8mnDQ254r6aOSuiLiekkFScsl3S1pe0TMl7Q
92665DVv3avjkyHm14ZMj2rB1bz0uBwA10+j8ytt1M01Su+1pkqZLOiRpmaRN2f5NkrpzXqOkQ8eGq6oDQLNod
H5NOOgjYlDSFyTtl3RY0i8j4vuSLo+Iw9kkxhvVdQBoFtNfVaiqnleerpuZGr17ny/NL0i6xPaHqzj+Ttt9tvuOHDkynXaupjH6oY+0SsR6sRPXvfBGW/6r
Z1XgXgOn3e9TPq3Y4W2I7Mv4+b3KsyJvu+SLUdkXIIaEJHiyiouhnoLMXgR0YNLsYVyK0WU1/1wC7Fa2vLTK5jQQyXb9wk2xpzfaDFOF8NKgU1mlD9IOMlt
/V+qZf+sVJOU3OkT3OTHlfTOsjVQPX2ZK7vN7oo11Awg0NnEd+4ZdUTy2x2+Fg9nYh1G5SBbq0hoH8Wk1NSjTO2GpN0Pd5aW/Vy2A8ubaVt9kGcqhq9uW20+Ls+cUpXlnqiuq3UQnU0PptE9Mz2PrTRBtnE08dRC4zOXnEMnFRHE8QLm+rx1kq+QGEbaOQK21u0+Uc5QDgZHc/UMrx2n9NAzBoU0H7BAtLPtOJKDSjpuLndqMFPjV8Mo3Ey8mdi7Or3smKmWVB0D4L9Xp1ojo2XyK00Yn4bq4lMYGwJbqFjOiuY0qNq6i0W3nGzZ41eaT
BnCbaZZzvwG2D+Xy6nswGqiXqx1BAh9e1VrzOJXXi8PqCxeWLM1LvL9R6zBWcE+q4UjnG4WPdWMqlgpEYVrgbUgm4e9wnI1nxVEzJv7l+QPSnV/SMTyVUbLMdTRg4C9H/8qPrvaqE/N3R1Dl/L3rSmZhXCzXYDv0OzDj6HHuUXemXS4TmBcrXXzS+5Mu0HiRWxrkkbWTb0A8WI0kW3jVJi6pXqE+yiNmvrT6Yt0vq0WgOB1MwGrY7UYuHMKQphv9cTv2/HWuTtszc9ovdBKyaLy/4A7RU5E+RTQ+WmFZa/wO6aX+zUm5kzN9QrT7mGQ4Y3PZ+fm8X+dmZ2ZPlIwjH2RVb0Gsh1zdzmhFJn3mY9yGB9Vki5HzW8RqBoVqm99vv7GzF5W3yNbdl7D7bUFX9K8Gc5uTkW8eeavbFlw/Y1wRcBGgUYi1G6e2NUtl1NhTrg6HHV2QENjN1Ly5RDHSW2Y3jmRJ2FjeB1xFznD7ifcvnBM6e9YfJDK5l6JfhB4qqQzYAkAiQRKFXoJ6hhtaXKoB2tnpp+HrvQ0Ksl2m4aIMFVIpl6gQNrwA+0zxjV9dz9sczN3NPybaZjQoADhH7SAZkHA9+XkSDMx/fXu6Yh7WSk9HNS/XOj/t3U70T/AQNVezJt0+bZjv4hNiHnz6ZAI2EvYAoKJ7O1Mh5GvRrEHiPbtuN7TZMDp1dNFYT+e7Rv7d0PAh/8Ys+sv0ke+FqZCtzbOSCKjO4/ypfdDp4GDF1+Kq2Fx7nPmNq3UxpF7bWu+4lgP+sBP1FHsOqTjx7f1dhhYvVg/w1Mr1GPpG3Ur4xRfhjXbK2HsGDwgS1rLsYWI1zdJkPGE1cnnvwEcPd0r/c5M0BjJiXp8VOIoFHZm2D5MWKNBcBZWnFfmOYeA+JAtJSh2oJF4VvLl8SmKHsOwuq2k1O/xlxOwo8vxgDOaErTwWEb7v8rtKx1/ZXHdQcqaU+eE6vjqNfhk0a/Ks6p6yQ0uMVJbwhIQ+VeL9pD6+VKa4HOV5SbGr/fhHcPtnsw3CN2AXBO/Sy3OGoW/Cj6U2Xlv7LAd5/TZ63ye7d+ARcgD2QAz0VoX2nNY8fO+GwNTR/e4+Aq5Cj0Dh4+bwQIgVUZ5cGoXDvAGxHoZrR8SWO1UA0vbO/a/4rDx3g+S/SgjUvwnxJNDbHFgvfc7FXUjD0m4NrHZ0bp9wDZ/8FNI9GDe6SQlVLK5+UZVrQC5Mm2mJ9MK1e/VXWgeWNQ+GNbDAnQ==

+uO2v2T7Cdt7bL/V9sO299leX3R+b3bubturylzjDttPZt8i/sF24UJtavagf1DSctuXSHqjpL6ifc9LentELJL0GUmfL3
H+WkmPRcRbJb1D0gbbl9a5zQCmjvaibptv226T9LeS3h8Rb5H0gKR7i44/ERFvl/T3kh6RdJek6yV9yPZvZcd8OD
u3S9JHi+qSJNtvkPSHkm7Ovk2MSPrghRrZzF03iohnsj6vFZIeHbP7dZI22Z4vKSS1lfiI35N0u+1PZtuXSLpK0p76t
BjAFHNe143t6zUa3NtsS1JB0uGi47dkrwOSdkfE4ey8/5HUKelnGg3392bHdUqan9XPuFXSWyT9MLtGu6SjF2pk
Uwd9ZoukL0i6RVLxX7bPSfpBRLw3+2PweIlzLekPImJvndsIANJo5uyOiJvK7D+evZ4uen9me5rtWyS9U9JNEfG
y7cc1eoM69hqbImJNpY1q9q4bafSrz19ExMCY+ut07sfZD5U5d6d6ukjzj7s2d7UV1aCACj9krqsH2TJNlus31dFee/TtI
vspC/RtKNJY7ZLun9tmdn17jM9usv9KFNH/QRcTAi/qbErr+SdJ/tf9fo16NSPqfRLp1nbD+bbQNAXUTECUnvl/S
Xtp+W9JSkt1XxEd/T6J39MxrNqx0lrvGcpHWSvp8p8KFMgAECLa/o7egBAPgQ9ALQ4gh4AWhxBDwA
tjqAHgBZH0ANAiyPoAaDF/T+ohYG32Po1HgAAAABJRU5ErkJggg==",
"text/plain": [
"<Figure size 432x288 with 1 Axes>"
]
},
"metadata": {
"needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
"plt.scatter(data['Gender'],data['Annual Income (k$)'])"
]
},
{
"cell_type": "markdown",
"id": "1aabf1d0",
"metadata": {},
"source": [
"### · Multi-Variate Analysis"
]
},
{
"cell_type": "code",
"execution_count": 6,
"id": "226e50da",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"array([[<AxesSubplot:xlabel='Age', ylabel='Age'>,\n",
"   <AxesSubplot:xlabel='Annual Income (k$)', ylabel='Age'>],\n",
"   [<AxesSubplot:xlabel='Age', ylabel='Annual Income (k$)'>,\n",
"   <AxesSubplot:xlabel='Annual Income (k$)', ylabel='Annual Income (k$)'>]],\n",
"   dtype=object)"
]
},
"execution_count": 6,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAAmUAAAJSCAYAAACC8LIyAAAAOXRFWHRTb2Z0
d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAAC

XBIWXMAAAsTAAALEwEAmpwYAAC6wklEQVR4nOzdd5ibV5n38e9RmdEUTe/FMx733ksc1/TeICQEQgKE
UBbYpS4sLEtbXlgWlrosoYZAeu/+F+9jqf3XtSl8/4xM46TuEyR9Kjcn+uayzOy9Ognaywdnec+91Faa4QQQ
gghhLFMRgcQQgghhBAyKBNCCCCGEiAgyKBNCCCGEiAAyKBNCCCGEiAAyKBNCCCGEiAAyKBNCCCG
EiAAWowOMVE5Oji4vLzc6hhAixlRVSGvLUKIYNu5c2eb1jp3ONeukFZeXk5k5Oy5O3bsMDqGECLGFLy4UF5bh
BBBp5SqHu515fSlEEIIUQECOtMmVLqKuAbgz9OAT4LTAJuBKqBu7XW3nBmEkIIIYSIBGdKdx6y1Xq2
1Xg3UADuBNVrr5cA+4KZw5hFCE8/4KZw5hFCCDF8Hf0e3D7/+y+y53+aSCREbDHk9gxBqJoBmYD6wSssspP
EEKI89ta2c79m6Azm4AzOX188Dm6u5f1MVWyrbDUwoRPQzqqbsFuAApGIAPoGbysG8G8g825WVUvcqpX
YopXa0traGJ6EgBWtW4ReYUPEyhGIyYKbsReAZAa90CrFdKbBQTmAk8bkEcHwBWnpca
C/gMuALeHOI4QQ4sLy02zcurD0rH+3YpJ8nNukFZeXk4nLp5//+Nmpidw4tzhkGYQINzl9GaV++Px
htld18rNb57CgbOyfUm1WM7/76HwSLCa+8NBu3D5/EFIKEftae9100TxTxGxwirsTzmHpeX5jH2PvMHNPVdTlxe
eZ0SsUUGZVHopf2NPLClmk+tGM/0qCdtzC9CT+6wOzOdjQw3+/cjRoxxUiVh1r7uXvW6r52+ZqGrudRscJiy
NNPacf80gHV539Hv62qYoHt9awp7Zr1Ble3N/Io9treXRHrWwDJWKKDMqiTG2Hg68/sY/oFT
KuGPJOP608RR7x/CiKUQ8aBssNPcHNB398TFb1tY78DhH85i7nF68/oFBVGvv2YvV3dTlxe
FNFa8/XH96E1/Pr2eSRYQjzOm/sbVU3n9cDPfeHI/z53g0z+czf/fyyyDPrcXhIt
ZiaP8DGXZyezsDyTHqePJRWj7pZFKeXV6fsrzDrFsd1fNMYzaU+nHO8cS44n53g0z+czf
d/KXt09x78oJIbsvYYrJydGpm4ahuq5Riahuq5RixaTcMWcozUqmNCt0r4FCGEUhnzmzAnP3VZP
PhxaUhv7+rZhZWd0OfAUTC/Q92zzTQoPGrvio5xbau+/YF5lau+/92zTQc
Xj+/WHs8LPcnhBAj8aeNp/jqLOsWMlxGaPFNOwl2ioZE/jsp8iL2WoJ+mbLsLl0LmDsLLDd78S8VO5YPI4Ht9wVoqU3
bPcrhBDDDUd81MBBzewNjWjggRKSQQVmE8/gCfP/5Q1TkpvDxi8fJ3JHFvbk/6Vw7
u5CZxelGxxFizKTQP8L95e1TnGrr568zyayL0rK/jZa8fYX9fNrBJ54RNCRIay7BS+fe10QTQyU
xbBWnpc/Or141w2LY/VU/IMy3HXxeWkJ1n55evHDMsghQgSFDMoi
2M9ePYbXrw3w3ePKSGLP8vGsPsdzC/rpuQ7MIIUbG5nJjjnn2ok0MegmhwgoR
8uqhZu5dWUFfRUoWoPRcU772LJyUhLM/HFDpdFRhBAjMKsnatnFZfKOmdZYCbmSQmh1FimKCQQ
VmE0VrzK5ePkE4Zk+ffE4ntvVbBCgLpOBy3SuFQIEw6462su1BUSSy
Lv7PInlg8MFP+y8ZTBSYQw3sbGbh7bUHUceD22qo63QYHUcIEeVkUBZBtNb89+89yz/EF59VAJWjlGUNc89JWjlGUnc/vicUbHOavijCSum13
IQ9tqZBm6iHs9Th8AWiFEtJNBWQR59VAzhp7+R4dxoUF0FCEMNb8sg7
njMlgyPosp+Xaj4wgholzkvvPHGa0R0nPOaUZTO8ok5/PXtXRz5LJuY
g8mkjI4jhIyMiiLEK4eH4uHV+bhtDyuLzJebiepSxtWuaDdmt
KC15pevD8yS3Tg3smfJhqbGwmWf07kLUr+vn0nUbHE
WLEZFAWAdYebPYXdc4d+Wk4GrvzDbcbzjI3EeV0c4j5UtF+YBYbGGmIhVqkJDBpqYx/Gvc
Nb8Yu0xyrKTuSlKZsmGfGhhKYkWE3+T2TIh4s6kvFSWTchmlkmC8ojo3nrhNwUlk/KYd64jJjbvUDEh8hrhBV
nXh+cJfvpB2dHzSzZkMyUBG6cW8STu+r5yGh1JCBEmSimWWVGQbHeNdlFIsKpfBmIhe0TUKiDFDt
WRl2cncPK/Y6Dij8rGLyn/Tyxs87oKEIIIUR
FCCCGiVnSOBGLE/755gsJTRZHTzgJKT8SzXwTAj1nMm/
NXMAjKSrTwoHf6FEEKIUYvu0UAR
Ze3hFu5eVk5kYmwsgP3w4lK0/ollcQoyaDMcDMAP/3lkmSE8zcdVG50VGCZmKencXlWTy0rU/oUQQohR
kEFZmNV2V2cDPb0xh+gL+FPwlLYJgD2VKqY8pPV5V5XSq1TSHShUrAuxqAXfzeVFDCHyiX
Qr+hRBCiBEJ66gBMKVUMMrJ66BMKVUMrNJaX6q1Xg4gDVa6+1ypwjOsjuyK10gFXClW9WGo7HTy8u5Myi/H
YK601R5t6qe1wGB1lWGo7HTy9u5Yxh1tmG4lN5tJXs3uW9PSjZ0qHyRFGsolAsIlUvAv
m5KYaHSdkJuXbWVMVSeyUPbatFCv6FGIud1Z28Ad28SxzsWSMmtrSMSQoTRIjGKoPkLpAlNcPqYRbkKYMWHP

+UCC1vpSwAFkAD2Df9cNZJ7tRkqpe5VSO5RSO1pbW8MSNNj+vqWaXrePz66aaHSUkPvw4nHS4V+IIHD7Au
987/UbmOTCAoF38rp9kZ1ViEgV7kFZN/DW4PdvAOVA2uDPaUDX2W6ktb5Pa71Qa70wNzc31BmDzuX18+eN
VayYlMOsknSj44TcNbMKSbNZeHCrFPwLMRaLx2dx0YRsLp2WR0WEz7BbLCa+fPlkVk3J5UuXTzY6jhBRKd
yDsk3A7MHv5wK1wKrBny8DtoQ5T1g8vrOOtj43n109wegoYTFQ8F/CqwebpeBfiDGwmk0srchmdkmG0VGGZd
64TD63eiKT8+1GRxEiKoV1UKa13gM4lVLrgEXAw8B6pdRGBgZpT4czTzj4A5o/bTzF7JJ0LqrINjpO2NyxZBw
ef0AK/oWIMnWdDg7Ud+PzBy585VGo7XBwsKEbfwQ1ma7tGHjMkZRpJKra+jnc2CN1vDEg7Hv8aK2/+p6LfjL
4FZNeO9TMqbZ+fnPHPJRSRscJm8n5dhaWDRT8f2pFRVw9diGiVVufm8d31qE1tPd7WDU5uOUizT0untg1cPxu
h5dlE3OCevzRaO11n87U6fCwYlJ0lcjUdjh4anc9AP1uHwvLswxOJMZCmseG2B82VFKSmcRVMwqMjhJ2QwX/
Wyo7jI4ihBgGf0AzNNni9QV/pszrD5w+vidEM3Ej5Qu8k8kbIZlG4szMXr/MlEW72NgNO0LtrO5gZ3Un371+OhZ
z/I1/r51dyPeeO8iD22q4aEL8nLoVIlrlp9m4dnYhHf0e5pZmBP34JZnJXDWzgB6nl3njzrrYPuwK05O4ZlYhnQ4P8
8ZlGB1nxCpyU7l8ej5Or595IXjORHjJoCyE7ltfSXqSlVsXlhodxRBDBf8Pbq2hvc9Ndmqi0ZGEEBcQ6iL9aYVpF7
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+HtNEz4ezMz9IymdKEUpSUxZEf9xQSWlWElfOyDc6SkS6bnYh33/uIA9urWFpHG3
OLoQYUJSRxPVzCul2+phdEj0NTxeWZ5FgMZGcYKE8J8XoOCKGyaAsSPbWdrGjupPvXBefWyoNhxT8CyGic
QGU2aQiZlsoEdtk9BAkf3n7FKmJFm5dWGJ0llg2VPD/2I5ao6MIIUTQ9Li8bDvVQVO3y+goIorJoCwImntcPL+v
kQ8tLMVusxodJ6JNKbCzZHwWf9tcLU0RhRAx46X9jbx9oo0ndtXh8UXPylIRWWRQFgR/31KNX2vuXlZudJSo
cPeycuq7nKw93Gx0FCGECArFO4u7ZJ2XGC2pKRsjl9fPP7bWcNm0fMZlJxsdJypcPj2fonQbf327iitnFBgdRwghx
uya2YUcbeqhNDMZq9QVi1GS35wxemZPPR39Hj5xsWypNFwWs4k7Lypnc2U7R5p6jI4jhBBjlppoYUFZFnlpst+
xGD0Zll2B1pq/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+6ToX4gwOtnaxxtHmsO2Y0Co1HY4eONlM43dTqOjCHFBMig7jxf2NdLv8UuBv4HSk618/8Y
ZHGzo4Y8bTxkdR4i44PL6eWFfI3tru3nlYLPRcUZNa82zexvYW9vNC/vCs/OBEGMhg7LzeGRHLRW5KSwoyz
Q6Sly7emYBV87I5+evHeNIU4/RcYSIeWaTInGw/i0lMXoLzZVSpAz2cUtOkLMdIvLJoOwcTrX1s7O6kw8tLEU
pOadvJKUUP7xpFmk2K198aLe0FRAixKxmE7cvHsc1swq5emah0XHG5NaFAwXzN88rNjqKEBcU1kGZUqpcK
dWslFqnlHp18LKvKaU2KqX+oZSKmL07nt5dj1Jw41wp8I8EufZEfv6hORxr7uOHLxwyOo4QMS89ycqUAnvUrx
hNSbQwpcAetp0PhBgLI/63vaa1Xq21vkIplQus0VovB/YBNxmQ53201jy9p55lE7IpTE8yOo4YtHJyLp9eWcE/ttbw
2A5pKivEaHX0e1h7qJmjTb3v+7s9tV28caSZfrcxneRdXj/rjrawvaojKPvf1rQ7eO1QM/VdkVfo3zn4PEhZhhhixEn2
NUqpDcCTwDFg3eDla4E7gMcMyPQuu2u7qG538Pk1E42OIt7jq1dO4UBDN//21H7Kc1JYVJ5ldCQhos7aw83Ud
zo50NBNcWbS6dXlDV1O3jzSAoDXr7lyRvj3+t16qoPdNV0AZKckUJGbOupjaa15bl8DHl+AqrZ+PrWyIkgpg2Pt
4Wbqhp6HjCTZ6F2EfaasEZgMrAEuAxYCQx8RuoGzVtQrpe5VSu1QSu1obW0NecindtWTaDHJ5uMRyGo28b93

LKA0M5lPP7CTU239RkcSIurYBwdhNqsZq/mdmtnkBDOWwb5YdoPaAA0NEE1KkTLGDEqp08dLtUVeob99M
FOixYzVHN2niUVwqGBMD4/qjpX6LAMDsmKt9X8ppeYDH9Vaf/l8t1u4cKHesWNHyHJ5fAGW/GgtyyfI8usPz
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
586lZVw/p0j2whNx58X9jXh8AarbHXx29YSw3391u+N0gb/ZpLh0Wv4Fb9PU7WL9sTb8Ac2ptn4m5qXS2uvmr
mXlIU57bi29Lt46OtDI3OPX3DBHPsyL4JCZMqDb4WXD8Taum10opy5jTHqylS9dPplN37iE7984A7fPz9ef2MeS
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/lHoOss3XJHxkcNArx1qxhfQXDur0OgowiAmk+LiiTlcPDGHuk4Hv33zJA9tq+GxnbV8f
s1EPrWywrBTPUKEU13nO4X4SsFVMy/8utjW52b9sYGid68/wE3zioOaqbHbydsn2gAIaM11QV4hX9fpZEvlwG
M2mRRXzigY9m19/gAvH2gioDUtvW4+uXx8ULOJd9tR3cGhhoEWR3lpiUwtSDM4UfDF/UzZq4eaKM5IYlaxn
LoUUJKZzP+7ZRZvfnU1l0zN479fPcZVv9jAzmrZskXEPrvNerqAPSslcVi3SU4wn67FzBrhacnhSE20nJEp+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+OuCvaFsF3oqWPp3bXcaQpNmvNAgHNuqMtP
Lu3YUQ7BggxWi6vn5cPNPLKwaaoWNww2pmyvwKvAEOtlY8B/xKEPGHx2qFmlIJLp8mgTIzNwvIsfnHbXPb
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

OEEF1vLmPw429nGrrZ3dNl9FxRq2208He2m5qOhxsO9U+otv6/AHWHW2hvsvJuqOtIUoo4sloW2Jo4CQDpypv
Bi4FDgcxV9C9eaSVkswkJsjqGBFGC8qyuHVBCX/aWMlJWTIvYkheWiIJFhNKQVFG9G4wnpWSQFKCGYCijJ
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+6aLvo9fpZPlFOXIjIlWEx86bLJHGzo4eWDTUbHERGkpt3
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
DJJMwmxS9eP2Z0FCGCxmRSIdlK7MzVlO91oQHTWDIpNfpFAvEq0WKWAdk5xMzopbnHRWVrv5y6FDEjP8
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/zjN76gkEoutMXTjIoEyIGHHvygrKs5P51l
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
63l6d73RcWKCLwgbvwcC2rDGmUP5z/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
wI2nEL021U5KaQkWxl3riMoB13SElmMuU5yWQmW5lTGtsLOkJtelFaUF8Dwl3on6i1dg9+/5/AUeA2rfW1Sqm
vA1Va60fPdwwp9BcieI4393L7fVuwmBWP3HsR5TnGDCwigRT6CyFCIZIL/a9SSr2llHoLyAf+DqxXSm0E5gJPh
zmPEHFtUr6dBz+1FK9f8+E/bJF+ShHA5fUTjg/LLq9fWqOcx2ieh3A9d2Ph9QekwD+Cyd6XQggONfRwxx+3YD
Wb+Mvdi5hZHH+nNCJhpuy1Q80cqO8e044Bw7HtVAdvn2ijIN3GhxaWnu7oLwasPdTM/vpuJuSlcsMwn4cNx1v
ZUdVJaVYyH5gfmQsumntcp3eTuHVhCXl2m8GJ4kMkz5QJISLQ9KI0Hvv0RVhNitvv28LbJ9qMjhSXjrcMFI2fb
O0L6YzLiZaBGsKmbldI2jlEuxODNZaVrX3Dnk0c+jet7YjcBRfV7Q48vgAeX4Dawd0LRGSRQZkQAhg4lfnk5y6
mOCOJu/+yjScGP1GL8LmoIpuMZCvLJuSEdKZl8fjM00Xe6cnWkN1PtFo6+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
9eVYHdJn2vIt2mE21sPdVBjj2RDy8qxTKMlic7qztZf6yVjGQrdywZR6IlPL24nB4//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
TCbFjXOLWfvlVfzo5lnUdzq5/b4t3Pmnreyv6zY6ngiz1l43To8frRlTx/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
35ZJuOykmXPVBHz5PSlECLirZiUyy9vn8fnH9zFZ/+xk/vuXBhR2+tYzCaumFFgdIxziuYVgkopVk3ONTqGEGE
ROa9qQghxHtfMKuRHN89i3dFWvvPMgbD2MfP4AjT3uAicUSDf0usK6axdS48rJrv/97t9Y97vNBDQNPe48PjG1
jKl2+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/gfwAzu01l9SSn0NuBGoBu7W
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+UUuuACcB6dpdRGYC7wdJjzCCFiXFZKAl
+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+3HYlIopXD7/EFZ3LB6ch5brR0UZdjIShnBiAyYU5JBn9uHxaSYVmgfcxYRH8Ja6B8MUug
vhAgFKfQXQoRCJBf6CyGECIJ+t48TLX2nN003QkuPi9qO4BTVn01Tt2vERfstvS5q2kOXSYhQktOXQggRZQI

BzcPba+lxeinLTuaW+SVhz9DQ5eTRHbVoPbBB+KyS4Bb617Q7eHJ3HVoP9A2bVph2wds097h4eFstAa1ZPSW
XeeOMOa0qxGjJTJkQQkSZgNb0D24d1WNQkX2f28dQ9UuvK/gZelzeM44/vEL5Xtc7iw+GexshIonMlAkhRJSx
mE1cO7uQky19zC7JMCTDpLxULpqQjcvrZ0F58GekphWm0eP04gto5g6ztcWE3BSWTcjG4fWzeHz072Ig4o8My
oQQIgpNyE1lQm6qYfevlGJpRehWpZpNasS94ZRSLAlhJiFCTU5fCiGEEEJEgJANypRSRUqpXUop1+Cel0OXf0
ApVXvGzx9RSm1SSj2vlLpwJacQQgghRAwK5UxZB3ApsOU9l38QqAVQSlmBzwArgQeAT4cwjxBCCCFExAr
ZoExr7dJad555mVLqWuA1YKixzmRgv9baB6wFloYqjxBCCCFEJAt3of9dwEeBTwz+nH1ClZx1CY9S6l7
gXoCUlBQWLhxWY1whhBi2qqoqeW0RQoTC/OFeMWyDMqXUJcAmrbVHKTV0cRcwVEeWNvjz+2it7wPuA9
lmSQgRGrLNkhAiFJRSu4Z73XCuvpwJ3KCUehmYoZT6IXAMmKmUMmKmUMgOX8f76MyGEEEKIuBCymbLBIv6xXg
DnAK8C/aa1/Nfh3G7XW3x78/g/ABqATuCNUeYQQQgghIlnIBmVaay8Ds19n+7vlZ3z/AAMrL8UIODw+qtoclG
YlYbdZz3k9f0BzvKWXrJQE8uy2MCYUQgghxEhIR/8o9dTuelp63KQlWfnk8vHnvN5bx1rYW9uNxaT42LJy0pP
OPYATQgghhHGko3+UcnkHuoq4fX700K6957meL6Dx+gPnvJ4QQgghjCUzZVHq+tmFHGrsYVK+nTNWs77Pq
sm5pCZayEtLJCc1MYwJhRBCCDESMiiLUnlpNvLSLlwjlpJoYeXk3DAkEkIIIcRYyOlLIYQQQogIIIMyIYQQQ
ogIIKcvhRBCRD1/QLO9qoPtpzo42txLe58HpSAzOYFphXaWVmSzoCzzvDW4QhhNBmVCCCGiVme/hz9tPMV
jO2tp7nGjFJRlJZNrT0RrONDQzQv7GwEozUrikxeP58NLxpFoMRucXIj3k0GZEEKIqOP1B7hvfSW/W3eSfo+PS
6bk8e/XFbNiUu7ZF7y8fqSZh7bV8G9P7fPvQnPVCx/ZyuLGHK6
bn89UrpzA5337O66cnW7llfgk3zytm44k2vvPMQe74w1Y+t3oCX71iCiaTnNIUkUEGZUIIIaLGU7vr+MYT+0lPsn
LfnQu4YkbBsG+rlGLFpFxe+OJvyv/cIf533UlT7fNxWaV05nCeDIoE0IIEfECAc2c2PXz7fesrWTI+i99+ZP6o
G2InJ1j4f7fMYmJeKj984TAu707+784FUmcmDCctMYQQQkQ0nz/A15/Yx33rK7lzaRl/v2fJfmoHcoUUpxz4oKfnT
zLN482sq/PLyHQODcW9YJEQ4yKBNCCBGxfP4AX3x4N4/vrONUcgUcvLeuO5aM49vXTuOlA038kXzYsdeOgFsWhEXeR+hLgQmSkTQggR
cbTWfP/5Qzy5q54vXz45ZAMyGDiV+YObZXjJ/dZ2OkN2XEOcjgzIhhBAR5771lfx8pURM
eT3l2Ax8cvb56GBrzy6F7/UlwkDyKBMCCFERHn9cDM/fvkI184s5AvXTAvb1kilWcl894YZbD3VfvE/ePhWW+
xTiTDIoE0IIETGONvXyYxYd2M6Mojf++dU7YG7t+YH4xl07N4+evHaOx2xnW+xZCBmVCCCEiQme/h3v+tp3k
RAt/+NhCkhLC3zdMKcV3b5b5hBQGt+8PyhsN+/iG8yKBNCCGG4QEDzlcf20tzt5r47F1CYnmRYLWWtrkKsZL5wySRe3
N/E+mOthuUQ8UcGZUIIIQz3x42VvHGkhX+7egozSzOMjkOrK1t9j7TQ0qQVQpTUWvmkaxVw2l6o+8LaD1SnLJDKlNwMVw9gL7rxXzdsqIXbRMrlCZsbWQ0XEOoOMbAZbXzeu3ORBL7dzgt1c8iItIQhlP6Pc8Y5QD2dM3c7p1UJMmEXAYQu7
SKmj1bdk6jcKeP8UPY8cX1QP4aL4uD/ppdvHXX4PHHYHRmK0ud1OJ8VFM9pYEs9vvYkjbhQgtlkd6zkJtdYqKwPrsN+WhkkqM+vhqGNqzdPq3gFS84hijBxcXyw3XfpwHA8cdYPvI0Oc4rWQwLdFaRu5FsfV9bIuO1xppzpOd1cB9XzdOK8ds30HF9bMoH4WjXQ6yv3xFRGkhnYLk1HKXN2WzMZaaT2bBDOrLrk3AUWA+U3G7ocQbpF5xs8XZ2CPrKsvQyKY5NOAdPXb7p2mq1o17AWcqeF1r+c1tSn8DdwIiQO9tQLK0tmZ2CfpxXSrQbq6VzENF5tJWv0XvrdHZ1jpf7qvsBwMFqHm/Dq31q0ZbZvuNtUtXY8VpBRlRb4t6oWVeoV8cx8E/tT3PLWtmJSIDe7v1WzQ+fQz4m2bWnl8IaFHqvlnwNaYHCYZ6hQBYHyaaOjkElWAbczAbyZBpLNNoIMj1tM6Dd5CF/Q+WUVVWbf1m/pCUuwMYmJHWqfdSX8zhTOjpXtw8uCJXvbOiCu9xvkVYYDO2TS5lc4Obu1wgW9ymD8lsXmlXvgU9PuJkJ/ezfVJgKzjSDAs3Dj3xUx9fVFkOYB2gXlYpRHuQv47Ll8KAMk/dwpDU0OBRLuVFf2aGtDhOAk2Bsvgw1I8V7xJjpt+Xo8OR8vbr2bNdVyiCmEuXYJQgZOtqGnzPsTNEpNzDbChO/p/L0fZ5ZI/5LQ4G5IEcdKzDk8BlKYtiwYKbiEFkH+c4OC7JXpTOf0y1PukgtmWRSTpEp5YiJm6yxB55YW0gHuGHTHBKXgfMk93pkCrsfuGVZ/A/6MI7x6Um6GBYEzh6OtpE4IHi4N2Ij5FFTQc9JbR5Nf78Q+Rf3Y2xdTCTmSNuy8Jj1K/ZJl19ofKSmbpLo8dWDEllIZR8lvYfqchCVo6CdXUT7kIWUeYZEEBP5MPCmZ1e08KPfUTJqzuKBZQ

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+eKHU1Irb5A5pvPbWfPHuidJU30JySDDKTrdIaQ5zVeWfKlFIvAQ3AM
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
RIDkBAt3LBlndAwRBhPz7EzMk4F3tPvec4do6HLy2GcuIlVOW0aFVVNy+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+cIige31nHfzx7kMun5/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+sGMs92I6XUvUqpHUqpHa2tw1sinJ2ayIyidBIs0T9l5PD4ONnah8d3/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/ev0418wqkAG9GJXLpuXzp42n2HC8latmyu9QLBtuXdhjvLPnJYCflPQTi0V
TC+0UpNsoyrAxpWBkhbxpNgtOj58D9d1kJgenrUWOPYGDDd10OjxkB+mY8WhpRTbpSVYumvD+wacQ5/OfL
xzGpBTfvna60VFElFpYnkmazcJrh2SD8lg33HlQi9baM/SD1tqjlJJ3+LNIs1n58OLRdefucflISjAzszidTofnwjcYhrZ
eDzOKBhYItDvOPVMmzm9BWea7TsMKMRzrj7Xy8sEmvnblFIoy5P+eGB2r2cQlU/N4/UgzXn8A63nqlUV0G+
4z26qUumHoB6XUjQy0tBBBlJZkpSw7GbNJMb0oOMvls1PPmCmTprJxZ29tFw9sqWZvbZfRUeKOzx/g+88foiw
7mXtWjKxOU4j3umpmIV0OL1srOy58ZRG1hjtT9hngH0qp3zBQ6F8LfCxkqeKU2aS4ZX5JUFYlDuno9zC9cKCo
vv08NWUiNm043orXr1l/rPV99X0itJ7aXc+Jlj5+95H5su2ZGLNVk3NJspp56UAjyyflGB1HhMhwt1k6qbVeCkwH
pmutl2mtT4Q22tjsrunkmT31NPe4LnzlCBPM5fJl2SkopUhPso64Ts0f0Lx5pIWXDzTiCGKTWhE+5Tkp7/pThIfb5
+cXa48zuyQ9anrdiciWlGDmkql5vHKwCX8UNbYWIzPc1ZeJwAcY6MBvGRo0aK2/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+wsMDCAoFdQguBEEIqjklI7MROXGPLdtxjW5Jl9a5RmfLeP87ZGT3rz8yZGT3fz0cfj47Pz1Hln
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
y1j64J4AfAT8FJjPaeC3gFJGXgbeB9cCrwf97CavVLCxJWYLTwRVzcjlQ38XKENVwqWn18MyeOlITndxVWTq
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+Em3qpuszscxejFY5WaUgZ8Adbvqydg DE1d/VN6KZJ1y6azoiyb9KQEslIS7Q5nUnZVt1FV
2wFATmoSC4tHLqar1FRx/6VlPLLpHb778hEe/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/j8Ol5g
KLfLyez8NESs9RJHcqypm8e2VvPmOy1nts0vSschQmGmm2mpE1vI2w7zC624izLdTJvgAuRKqfiyqDiTO1aU8O
jmE9S0euwOZ0qI29mXU9HKsmksL80etYL6ruo2TrVZVePn5Kcxp0BLBAx189JiAgEz6nXcfLSZlu4BGjr7WFK
SSUpiAouKM1lQGLoq9pGyeHomC4tiL26lVHh9/vp5PFdVx9fXH+T7962wO5y4py1lcWYsH6ql01IQsVqF8tPjd/
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
MrNIU6W7XlHvCJyc6oyIhA8hLs547mcmukMSUk5aI0yGInH1spZSKJStmZPPl981n/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
VgLJ5o9vHak2e5wlFJqypudn8bvP7GK796VK81aPbeat6ajQ2aFIWYj5/gM1Hm9lytBl/hJteCzOsIrhZK
S7cIaj7Yoxh6zstbDzcRL8v/pbZSU1yku62qsIUZCTZHI1SSimwZmeuWzadv/z91Xxh7Tx2Vbdx239u4UM/3cprR5
riev1MrVMWYlWnOs40t6YmJbC0NCtix37v/Hwump5JZoorJFXxDzd0s+VYCwAuh3DZ7NxJP2Y0SUpw8qFLy+
js85KfPvqqDkoppSInOdHJp9bM5oHLyvn11pM8suk49/XNmblpvKhS8u4Y2VJ3NU306QsxNKHLIWU5o7s5RU
R8jNCl1ykuRNo7u7H5w+QEqdrfbpdTlsWQQ8EDHtPdeB0CIunx3YNOKWUCqfUpAQeurKCD68q50/7U
7y1efe5psvHuKGxYXcunw6qyty4mLd5Pj8pLXRnIJ07qp0IiJMz4rsckyh1uf1405w4nc6GPAF7A4nruyubWfjoSbA
KhA8rzDd5oiUUiq6uV1Obltewm3LS9h3qoPHtlXzX2Z3J46nnrrFHnpSdyytJhbl01n8fSMmK11pklZGJRkp9gdQkg4R
M609sXBF5Co4hjyhqHXVimlxmfx9Ez+720X8X9uXsgrB5v441un+OUbJ/mv148zPzWc8vSYmbnp9kd6r
hoRX81oqONXfR5AywsysDhiM1vHtHIGMPbpztxOR3MLdBWsmigFf2Vim3tngH+tK+eZ3bX8ebxFoyBhUUZ3
LKsmJuWFNnWYDKeiv5TKinr9/lp6uqnMMMdF33PkdDuGWDAH9CB8CruaVKmVPxo6Ozj+arTPLOnjt017QC
sLMvmlqXF3HhREXnpkZtxH7VJmYi8B/h3wA/sMMZ8VkS+AKwDTgIPGGPIPGGO8Iz3GRJMyYwy/evMkzd0DzMx
N5dbl0ydwBlNLQ2cfv91egz9guGFxIQuKMuwOSamw0aRMqfhU3eLh2ao6nt1Tx8H6LhwCl11XkcsvSYtYuKiQzJ
bwzOKN57cuTwHuNMVcA+SJyBbDGGHM5UAXcBwy09lj5XnN3/6j7d/Z58flHH9ze0++LyxpeYBVXHay
11tI9MOK+Pn+Azr4R82kVRTp6vRGvo6eUnaYZPCp9bMZv1nrmTDZ6/kb66eTXWrhy8+WcXF//ISD/58B8/s
qcMzYH/R9YgO9DfG1A/51QcsAV4N/v4ScB/wRDiO7XQIaxcXih2UlWSPuu/loM9uOt5KXnsS9l8zAOcxYq
qONXTxfVU9igoP7LpkR9mw70uYWpFPf2Ue/N8DKsuxh9/P6A/xmWtZUt3QOsqsjh0lk5EYxSjdcrhxrZVdOO
bD1SW6lhBpdSUMbcgnc+vncffXz+XqtoOntlTx3NVdbx0oIfkl5PrFhZw8Zw89JirpybS1JC5Msl2TL7UkSWALlAO1Z
XJkAHcMFPfhF5CHgIYMaMGRM+7vzCDOmn56BnxkuC+cbFW3eggYQ5/XT0NXX8STMm
NMWKf9Oh3Cmnn5ox6nq893piWtusWjSVmUO9lsPb/rO/ro9wVITozsG89knrfhfs4rpaYGEWFpaRZLS7P4yo0L2
Ha8lWer6vjXX8vjTXmscWoY7gfctLuLmpcWsqsgZtsZAm1CKellInINOD7wN3AmBwAwnSBPsGVsIM0RkBwmwBpTFu4
YL6vIZcuxZsqmpQ6bkAEsL82muWuAlCQnNM3NTwx3WOTYdbmJXdRsLizK4flFhWI7R2efliR219Pv83LpsOsX
D1F2blprIirJsats0IYsFl8/JZevxVmblpkU0IfP5Azy5q5aGzn6uXVDAwuLxjVHc+k4Lb7zTQkVeGjctKdLkTCkVE
k6HsKoih1UVOfzTLYvYfKSZZ3bX8dzeUzy2rYas1ETWLizg+kUFrKrIwe2K+FdPNetMX+Rf8HVb9BdtUJeBPL+QVe5xmJf8TXJ89hT/nCXlRyhJ42ozPPSp0Qlc8NkNaY/9LOhm7n+mfMNzx9W31q29BtPxw8PnuZl7nD9sJ/3LaQbzJkl0i
wZyXiGMzM3dUxJVnZqIndfXBqBiN5tf10nxsDbpzu5bmFBWJ4k1S0eOnutcWJHGruKoArpqbF/Ljq/CYnZ/O7
PzIl+Fo7Rmgrr0PsJ63403KBp/zRxu76fcFbFmJQSkV31xOB2vm57Nmfj4Dvnc2tpZWnuWFg2vmZ4dmWvhbi6E5Zhzrg04x3/NIIPsWqlLsazVl2P9DlAP2GR3i9EN5pVJQ8dpBr4p2eEIwQnWixzFzI1Ro7sQLPzdHFi7CbGnbpy6cNrYb69QGvdDTLXJnW/Re9bphPmgYF
1zKnStKQrqCzqBIt5TdBVwMfD2YRHwZ2CQirwPHMk+Y1bR6eOlAA3npSdS5JGFruq21lU
HL7KxYUZbqpbPfR5/dx4UXha49TUkZOWxKy8VE539LGsdPxLS60oy+aNYy3Mzk/ThEwpFXZul5Mbyrixou
K6Orz8uL+Bn63o4ZvrD/EtzYc5r3z87nn4lKumpsXgNgPW
BqK1hKp5pSQyl1Fgca+rmdztqeHJnLc3dLxRkJPGx1TP566zZ9DBHLSEslN
S7QxuuENxr39ROvoO49BaXYKKYlOXE5hVoTHzCmllFLRqCIvjSduy/bhPkafvShlSwsyqChc/RSW2Oha19ew
GtHmmno7KO2rZcFRemku10sKs5kTn46LqdE7QDjjYebaOrqD8adQVrS5P68mSkuHrxiFgFjcOkKCEoppdQZL
qeDGxYXcunw6qyty4mLd5Pj8pLXRnIJ07qp0IiJMz4rsckyh1uf1405w4nc6GPAF7A4nruyubWfjoSbA
DGxYXcsPiQkLV6zilPmmPNXWzfl89de29I+5XnGUN3stOcZEcHLvS2efl1UONZ5ZriEaDXarTUhNxJ4TmT+t0i
CZkSiml1AhC1VgzZVrK/AHD81Wn8QcMde29fOzymcPue9XcPBZPzyTD7TozeO+1w80cbugCrOQnHLMuJuvq
eXlcVHJu3EoppZSKDXH7yV3d4mHLsWa6+61lExwC6W4rB81IPlt7rLVngCrC1Hre7KQSJCbloSiUNamzKD90lM
cOCOcLHNsbpQ3EoppZSKDXHZUtbT7+OPu0/hDxjqO/q4fUUJIsI9F8/gdEcvJdkpZ/Z9vcp2jxedte288mrKoZtgl
w9O4eS7GSyUlwjFpRVSimllJqIuEzKHCI4XxKq/aWg3XnKik1l5aefsO/j/CaPUHRMRynUGolJKKaXCJC6TsuREmvXAQb
3dXlnK6o495hSNXLl+3rJhjTT2UTUuJ2lmVSimllIp/cZmUAeRnuMc0Gc0d/d7WJZaVb4A1JKKaWUGoGOCFdK
KaWUigKalCmllFJKRQFNypSsSimlokDMLUguIk3DML4pQ2jC7Jf4r/QE+HKJwokE8nY+eS3SaKueyAtgVxsePRrEUbyzF
CrEVbyzFCrEVby6QaozJG8vOMZeUTZaI7BjrauCrau2xIJ7OR88lOum5RM/jh1osxRtLsUJsxRtLsUJsxTveWLX7Uiml
lFIqCmhSppRSSikVBaZiUia3jjWwfpYJxUk6P3EW47d2iHQu/DQ2p6y/X2p8hu1HkvXVyHAE5dfgS9mbd4bB3LmScy3zyXi5HEyLiPwcE/NeQdERhBMF1SrbzN1ZZxBmHY2TdWoOmbz2WvRWqpSvJ4CpzHWoGyGMfhOGPXaxjhvSnbkRh6zsZMdntkXcd3qy4TLy78N3uLmFjE8c+bgPR8Ry0jznIyhbsz4uKfTBgWIbyqBLMTeuExjbybRPTBDU87gr0TgDSlEW4PGc2iH0lhJSvj1QZW3E+rCiA4G7rdfR1YSFxPYvPG9jHSCaQkp0qrnE2ayGhA8oT2FbVHeYEfXhXUyuUbuzoJA1Br2mLpYMXWAgjcGndY5cWZo6HMNrHIdULiuDe3wFJ7Dq2C4a8/l4VrRTnoHADljAJTUTqWbMh5fPx/Xb/KsRcF2VTV5EpH1wvkRlv/nMvRmXw/bZ6tTYOWpD6m5+zszHoi9RkJijQhpeYSRzWjKIRPJjk2PrdnpWJ0FmLO2VYy7DvqDDQPxMHHWY6hYr6L/jm2nl7FZdbzRFp9TJmJy97NcZfGV+lvJDXTHKkrJeB4qE/+ZxDz/wASC86ZJQ6AzXnwtJUvk6fPBy98ny4Q1KqBs3AJyB47l+4FTW8I/RHzIb8wIsZKSgo66i1cZAa1bv9gqtmDAn2W0H4k7SGpUZIFXVvF4gbW5i/CyZtcmmU9oLKHFz/hBhzZmUnKHd6yG2fmqf8N9mWqp3BIKyEtN3D75jRJN5gdfwnDrCxtlafytVK+h/Cm7P2xSWL9MEh23xCq3vANhzwvnKnyYIxSwzz/LwPC54zdQeJ0YOKlp0GnbmrYT4JJmYUrDRhBoLxFGWpjF+MlomOh2SGp2z6Jbf0D2s8fhWJ+gcrIkNnjOXd4iSfthqzMxtC4mtyh3zHb2DhOuPlV6lxxOD+L3pHT7BDdDpHR7CZ0jLtTwHBGqbnvTLDsjL07lxHoDt0AgtlZxNP/dVWt3v+Lw/xdZWU9Es/Sw1Zos/NULIxiJkZpcJxXUzd0Ihw/SqSkCVfOp7Gm+AkaI2ggQYZ6G/jKFR6kCTYvXkhChvQmSZXZxkxFa/QqhRpe88SDRvLC1cUIyEwbzQZWP7SmjRIjQ/ZSmC0oSwtBJI2k+QxDWbCOrSpA9yI0VepoUsHb7x8T+Pcm47sEp4EUzsTfJeqWt2EHyrX8aUy+Vw4B4/9v+1bWsa61XS5ZYt2OkL7+fKi2dRuF4w5rVFXpaXyEDzdPm2ewmKI4Te3eL6wOUeBcONRLtEOd/WfoiaRaWM6vc7mH2MBGxrkGwZo+XNvlKe98h8k3/OdtSxR+wHbgcC3QtO7aATLTEdjcXnPGW8bJvZ6TnCk7WfPtQ6O+oSZa9uO7DEg/b5AUqbWvk+dL1F9nlpc3bX9Rz/VhMLrcgHgf6+8+VrfmWfDF/bbkI0VVFsvPqzmJ76qLGl/yglTUMOwXpsFDqI+Sa3UjJFs7Vlpwk

W4zxiz1s7YLiSWJiaIiAA3Yr3ONhhjAsHt64wxT9sa3BiJyFeNMf84ln2nQvdlpTHmKgARWQI8ISJfsDmmiZo95F
z2GmPuDN5+xd6wJuQpYAnwqDHmVQAR+ZMx5n22RjUxicF/bwPWBN80fiQir9sY00QFgv8uNMZcG7y9IUaf
Y4eMMWtEZCZwO/CUiPQDTxtjQtFNE2vvLbH0mou152GWMeZf4cx787eDtx+wNarhPQrsBX7DuYPRHwWib
WLCL4ETgA/4nyLyoDHmEPB3QNQlZSJSjdWqO/gcFmCRiFw9li8/UyEpSxCRRGPMgDGmSkRuA36F1ccba4b
+vb4y5Lacv2O0M8Z8O9jk/6CIfAJ4zO6YJmGhiPwCqACSgN7gdrd9IU3Yz0Xkp0CNiPwK2Ij1Qb7D3rAmzhhzH
PgW8C0RKQDWheihY+q9JcZec7H2POwRkf+F9fqvF5G/B1qBfnvDGla5Meb+87a9FWxNjTYlJjiDVUQeAR4Vk
R/YHNNIPgvcAWwAfmWM8Y3ny89U6L68BDhhjGkcss0J3GWMedy+yMZPRBYBB40x/iHbEoEbjDHRh4V
MH+9/uBebHW3QcgImVDfq0zxniDtWquMMb8ya64JkpEioG1QAHWt+gtxpg99kY1fiKy1hjzYhgfP2bfW2LhNR
dLz0MRSQZuAI4BR4CPYH1ZfswY02FnbBciIp8HrubdExM2GWO+aV9k7yYim4CbjDGdwd8TgUeA24WRuRH4MLAFuGVIi+/I94v3pEwppZRS5xoy6EpJCLgLlFl80xoy6SOL4CQYrBa0qJqgJCLlQPv5EyZE5EpyGjzCZbghoHEVkDLAb
eHMu11aRMKaWUmkJEZLglFl80xlwX0WBGEUuxwrDxCrB+LPFOhTFlSimllDqrG2u25VCNW4v7zjEXCCNW4v2gzGKpw7
ezgaY4Vz4wUr5jHHq0mZilnBgdV/ABYYYw7aHY9SSsWIA1hjss4Z7yYif7YpnpHEUqwwyXi1+1LFLBH5HVA
EvGyMedjmcJRSKiYEi/O2GGMGztueIV1ymImVph8vJqUqZgUnN14CFiDVRR0frAv//tYs4iOY6039t/GmN+Ly
Erg20Aa0Aw8YIw5bU/0Siml1LsNN4BOqWh3K9bAycNAq4iswCoQWo617M2DwCoAEXEB/wHcaYxZCfw38C
82xKyUUkoNS8eUqVh1L/Cd4O3Hg7+7fXrh1T/noc1JfnP1oodOAFtJVNKKRVVNClTMUdEcoD3AotxFxG
AlWQZrGZkL3gVrfb9VEQpRKaWUGjftvlSx6E7gF8aYMmNMuTGmBaheM/RqEHsXmEAtYjJnujO
DqyMopWwiIreJiBGR+TYc+0VkeXH4KEXk4WJl/uH0fPu/3PBFZH+YQ1SRpUqZi0b28u1XsSaA
YqAX2AT8GtgIdwVkwdwJfF5E9wG7gsohFq5S6kHuB14F7hs9Fa5S6kHuB14F77A4khnwFa3zssERkYXBpok+KyC4RuRfAGNMEn
BaR1RGIU02QJmUq5hhjrbGrD9v2/eMM+MWQj8FTAX2Bv8/93GmCuNMUuNMYuMMT+JfORKGKlgM8jhOEdkvIg+NI75yE
ze3o11uv0niHbrxaRV0Xk9yJyUER+LcGBoMFWrH8KJhp1vYzm8xEpF9/ishOEdkvIg+NI75yETkgIj
8J3ndDcH1LRGS2iLwkInuC+1UE98SkIv9PRD4oItuC+1UE98sTkSdFZHvw512Jk4ik
A0sutP6niHxcRP4UjPFh4BfAD4PXeOjSPn8EPjjW6W6AiT5MyFW+eE5HdwGvA14wx9TbHo5R6t59
wALgVlYicWgZmPMCqyEY9iuuyE+FpxxXQl8OjgedazmAD8wxlQ9D2Xwe3L8VqcT+NNfN7GbAUuBb
4ZrBeFcFtf4c1K/x+YK4x5hLgp8D/CO7zXeBfjTEXAOEY0yvvMe
bokN13AFeM4xqoCNOB/iquGGOutjsGpdSoLjR7elfw923GmdFqA4BescqxuTrBW8ADYiZUIjebTwZU/AEqxEq2
WMcZ43Bize8jxyoOtVdONMU8BGGP6gnFeDvzmGDXkrI3Ax0hsH6yJKCLHgA3Bx9yLVWWxWcWrREuYbB
RECBDRNKNMV1D4ikCms6L8X6sIRu3GmO8wW1fAv4NWCsiy4H/NaR1rRFrmIeKUpqUqZg1Dv/QP2d3PuZ9T/RfY7uPcXh938DhXYyU7q4wxHhF5dfD/xuj8OJI5u57h+Ybbfv7jBIb8HuDsOTiCcfaO8Di9vDv
+fVgtdCVYk50wxpwC7hWRr2J1Xf4BqAju7w4+jopS2n2plFIqqoabPX2FIqqkoabPX35BB/vBLACINgNOjO4PRNoCyZk84FLJ
xc2GGM6gVoRuTV4vCQRSE2AR8QEaeI5AFXAtvGGG8dcMvG8dAbgL8K1IL1O
A1cKXOmT/uVygC1RFD03KlFJKRdJws6fvm+DjPQ1C3Z1fhI4HNy+HkgQkSrSga8CbE3zP1S1aBWwBCr
HOpwrYA/wF+OI4x7N+GqgUkSoReR4v4xPk7GGMOApnBLtSh21/HGl/3fLCcx+0i8DPxaxk79NDdl8DPD+OuQuF
SE6dqXSimlVAwcAkc8CXcaYC00EOEH5SSikVG37IuwPUw0F+C3arf1o+RvLq0F+C3arf1o
QsumlLmVJKKaVUFNCWMqWUUkNJmVJKKaVUFNCkTCmllFIqpRSSikVBTQpU0oppZZS8f
st6I5l0KciEAAAAASUVORK5CYII=",
"text/plain": [
"<Figure size 720x720 with 4 Axes>"
]
},
"metadata": {
"needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
"pd.plotting.scatter_matrix(data.loc[:,\"Age\":\"Annual Income (k$)\"] , diagonal =\"kde\", figsize=(10,10))"
]
},
{
"cell_type": "markdown",
"id": "b7153f24",
"metadata": {},
"source": [
"# 4. Perform descriptive statistics on the dataset."

```
]
},
{
 "cell_type": "code",
 "execution_count": 7,
 "id": "dd6ec149",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>CustomerID</th>\n",
     "      <th>Age</th>\n",
     "      <th>Annual Income (k$)</th>\n",
     "      <th>Spending Score (1-100)</th>\n",
     "    </tr>\n",
     "  </thead>\n",
     "  <tbody>\n",
     "    <tr>\n",
     "      <th>count</th>\n",
     "      <td>200.000000</td>\n",
     "      <td>200.000000</td>\n",
     "      <td>200.000000</td>\n",
     "      <td>200.000000</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>mean</th>\n",
     "      <td>100.500000</td>\n",
     "      <td>38.850000</td>\n",
     "      <td>60.560000</td>\n",
     "      <td>50.200000</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>std</th>\n",
     "      <td>57.879185</td>\n",
     "      <td>13.969007</td>\n",
     "      <td>26.264721</td>\n",
```

```
 "      <td>25.823522</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>min</th>\n",
 "      <td>1.000000</td>\n",
 "      <td>18.000000</td>\n",
 "      <td>15.000000</td>\n",
 "      <td>1.000000</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>25%</th>\n",
 "      <td>50.750000</td>\n",
 "      <td>28.750000</td>\n",
 "      <td>41.500000</td>\n",
 "      <td>34.750000</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>50%</th>\n",
 "      <td>100.500000</td>\n",
 "      <td>36.000000</td>\n",
 "      <td>61.500000</td>\n",
 "      <td>50.000000</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>75%</th>\n",
 "      <td>150.250000</td>\n",
 "      <td>49.000000</td>\n",
 "      <td>78.000000</td>\n",
 "      <td>73.000000</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>max</th>\n",
 "      <td>200.000000</td>\n",
 "      <td>70.000000</td>\n",
 "      <td>137.000000</td>\n",
 "      <td>99.000000</td>\n",
 "    </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)\n",
 "count  200.000000  200.000000          200.000000              200.000000\n",
 "mean   100.500000   38.850000           60.560000               50.200000\n",
 "std     57.879185   13.969007           26.264721               25.823522\n",
 "min      1.000000   18.000000           15.000000                1.000000\n",
 "25%     50.750000   28.750000           41.500000               34.750000\n",
 "50%    100.500000   36.000000           61.500000               50.000000\n",
 "75%    150.250000   49.000000           78.000000               73.000000\n",
 "max    200.000000   70.000000          137.000000               99.000000"
]
},
"execution_count": 7,
"metadata": {},
```

```
  "output_type": "execute_result"
 }
],
"source": [
 "data.describe()"
]
},
{
"cell_type": "code",
"execution_count": 8,
"id": "fde09c99",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/html": [
    "<div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
    "      <th></th>\n",
    "      <th>count</th>\n",
    "      <th>mean</th>\n",
    "      <th>std</th>\n",
    "      <th>min</th>\n",
    "      <th>25%</th>\n",
    "      <th>50%</th>\n",
    "      <th>75%</th>\n",
    "      <th>max</th>\n",
    "    </tr>\n",
    "  </thead>\n",
    "  <tbody>\n",
    "    <tr>\n",
    "      <th>CustomerID</th>\n",
    "      <td>200.0</td>\n",
    "      <td>100.50</td>\n",
    "      <td>57.879185</td>\n",
    "      <td>1.0</td>\n",
    "      <td>50.75</td>\n",
    "      <td>100.5</td>\n",
    "      <td>150.25</td>\n",
    "      <td>200.0</td>\n",
```

```
 "    </tr>\n",
 "    <tr>\n",
 "      <th>Age</th>\n",
 "      <td>200.0</td>\n",
 "      <td>38.85</td>\n",
 "      <td>13.969007</td>\n",
 "      <td>18.0</td>\n",
 "      <td>28.75</td>\n",
 "      <td>36.0</td>\n",
 "      <td>49.00</td>\n",
 "      <td>70.0</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>Annual Income (k$)</th>\n",
 "      <td>200.0</td>\n",
 "      <td>60.56</td>\n",
 "      <td>26.264721</td>\n",
 "      <td>15.0</td>\n",
 "      <td>41.50</td>\n",
 "      <td>61.5</td>\n",
 "      <td>78.00</td>\n",
 "      <td>137.0</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>Spending Score (1-100)</th>\n",
 "      <td>200.0</td>\n",
 "      <td>50.20</td>\n",
 "      <td>25.823522</td>\n",
 "      <td>1.0</td>\n",
 "      <td>34.75</td>\n",
 "      <td>50.0</td>\n",
 "      <td>73.00</td>\n",
 "      <td>99.0</td>\n",
 "    </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "                        count    mean       std   min    25%    50%     75%  \\\n",
 "CustomerID              200.0  100.50  57.879185   1.0  50.75  100.5  150.25   \n",
 "Age                     200.0   38.85  13.969007  18.0  28.75   36.0   49.00   \n",
 "Annual Income (k$)      200.0   60.56  26.264721  15.0  41.50   61.5   78.00   \n",
 "Spending Score (1-100)  200.0   50.20  25.823522   1.0  34.75   50.0   73.00   \n",
 "\n",
 "                          max  \n",
 "CustomerID              200.0  \n",
 "Age                      70.0  \n",
 "Annual Income (k$)      137.0  \n",
 "Spending Score (1-100)   99.0  "
]
},
"execution_count": 8,
"metadata": {},
"output_type": "execute_result"
```

```
 }
],
"source": [
 "data.describe().T"
]
},
{
"cell_type": "markdown",
"id": "97ed7025",
"metadata": {},
"source": [
 "# 5. Check for Missing values and deal with them."
]
},
{
"cell_type": "code",
"execution_count": 9,
"id": "f8a6ad05",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "CustomerID                0\n",
    "Gender                    0\n",
    "Age                       0\n",
    "Annual Income (k$)        0\n",
    "Spending Score (1-100)    0\n",
    "dtype: int64"
   ]
  },
  "execution_count": 9,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "data.isna().sum()"
]
},
{
"cell_type": "markdown",
"id": "acad58cb",
"metadata": {},
"source": [
 "# 6. Find the outliers and replace them outliers"
]
},
{
"cell_type": "code",
"execution_count": 10,
"id": "45bcc770",
"metadata": {},
"outputs": [
 {
```

"data": {
"text/plain": [
"<AxesSubplot:xlabel='CustomerID'>"
]
},
"execution_count": 10,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAABGUAAAE9CAYAAACiD99jAAAAOXRFWHRTb2Z0d
2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACX
BIWXMAAAsTAAALEwEAmpwYAAAhHUlEQVR4nO3de5QtZ1kn4N+bBGIICIQEDAnjAQ4XASGEgATQ4S
Z3ISgOyVIGF47MeAlRB8VIFgZWZi0VRmCCIgYdBhAgWAWIBeBgI4juZEbl2AjB02IIREhQLiGb/6o6mSn03
3O7qS7v9p9nmetXt27du3ab9Xuent/v11VXa21AAAAALC19uldAAAAAAAAAMC/kFv0pQsbrYvEgQSNnnnvvuVa21Q3rXcVPpP7DY9CCgl0XvP
wAAABAB0IZAAAAAgA6EMgAAAAAd7LeemQ8++OC2Y8eOTSoF2EznnvuVa21Q3rXcVPpP7DY9CCgl0XvP
4keBItsTz1oXaHMjh07cs4559z8qoAtV1Wf713DzRHLo/LQhDfr3/JHoQLLLI99SCnLwEAAAB0IJQBAAAA6E
AoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAAoQygAAAAB0IJQBAA
AAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAAoQygAAAAB
0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADrYr3c
Bi+TUU0/N0tJS7zI2xWWXXZYkOeywvnv1fSZKqevcW+Dyyb2Z2r4xTVm7Q4o1vncZsKqevcW+Dyyb2vucnv1J74Dtbyo9b7u9F9M/108
osw5LS0s5/+JP5dpbHHS7S7Evtd8vZ8fSbUmtd8qXcJsFs9e4t9H1g2tfc5vfqT3gF7h6n0vO3OXkz/vGkW/5Xf
Ytfe6qB8495P6l3Ghjvg0+9Jkm29bjBlvXqLfR+YNaX3Ob36k94Be4+B9Lzt9F5M/7xpXFMGAAAAoAOhDAAA
AEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZ
QAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyA
AAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAA
ACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAA
EAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoA
OhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVA
GAABYKEtLS3nyk5+cpaWl3qUA3CxCGQCYKGccop+frXv55TTjmldykAN4tQBgAAWBhLS0vZtWtXkmT
Xrl2OlgEW2n6bsdLLLrssb2rlnJpo/qJn8bXTTxV/0UvbNQr41OmsCr6Tr2rp2W9mxm1I+tTsQJl3tWKjT
BmzbKsoi/1XJW0xaPNKBH9vmrdKNF3ymuaqUYm2sGz6+ACoNvqFaZyxnOgoMp8+Jr38ZtT/2NgHiqOS+3HqANl8bBQNcdf7/jhBdymgprnFgnRG9kpWvz7a
e3tI/N/o9kGvKAAAAC+Okk07a7a7W2AR1TUOzp9mx44kZ9iZzE44LmHb9LwIlx4Ed5Xc7jP9t9hB9nX0h0oPWEQjFXgDHftR8dFHkPt9J/+J+
4IGOkgEW3qb89yUAAAIDNsnPnznZnzVTUXtoq7kiiMLkVYmJY4MsmaYbTPpNz9UqU51Pj6/VDuHxwOGvlBw4OZJxT2ebbHEHn7HuMm4QrlVPJAi
BAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADrYr3c
BAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADrYr3c
Q

QAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAAAA6EMoAA
AAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAAB0IZQAAA
AA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADoQyAAAAA
B0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhkAAACADo
QyAAAAAB0IZQAAAAA6EMoAAAAAdCCUAQAAAOhAKAMAAADQgVAGAAAAoAOhDAAAAEAHQhk
AAACADoQyAAAAAB3s17uARbPvNV/KAZ9+T+8yNty+1/xbkmzTdftSkjv1LgN2q1dvse8Ds6b0PqdXf9I7YO8
xhZ63nd6L6Z83jVBmHXbu3Nm7hE1z2WXfTZIcdth23InutK1fOxZfz99P+z6wbGr7S7/+pHfA3mAq+/n2ei+mf94
UQpl1OP7443uXAGxDegswBXoRsDfR85gK15QBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAQAdCG
QAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6EMAAAAA
QAdCGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB0IJQBAAAA6EAoAwAAANCBUAYAAACgA6E
MAAAAQAdCGQAAAIAOhDIAAAAAHQhlAAAAADoQygAAAAB0IJQBAAAA6KBaa/PPXHVlks/PMevBSa
66qUVN2HZdr8S6Lar1rNsPttYO2cxiNtM6+s+8FuX3Qp0bS50bSw+66ab2GqtnbVOqJVHPnqxWz0L3n2SuHjS11
2Eeat58i1Zvsng1z1PvbnvQukKZeVXVOa21ozZ8wZ1t1/VKrNui2s7rttkWZdupc2Opc2MtSp1TNLVtp561TamWR
D17MrV6tsoirreaN9+i1ZssXs0bUa/TlwAAAAA6EMoAAAAAdLBZoczrNmm5vW3X9Uqs26Lazuu22RZl26lzY6
lzYy1KnVM0tW2nnrVNqZZEPXsytXq2yiKut5o336LVmyxezTe73k25pgwAAAAAu+f0JQAAAIAONjSUqaonV
NUlVbVUVb+1kcvuoap2VdVFVXV+VZ0zTjuoqj5QVf84fr997zrnUVVvqKovVtXFM9PWXJeqOnF8HS+pqsf3q
Xo+a6zbyVV12fjanV9VT5q5byHWraruUlUfrqpPVdUnquqEcfq2eN22yk3Zjp3q/L6qOquqLhjrfPEU61xWVftW1c
er6l3j7cnVuSg9vKpuV1Vvq6pPj7+nR0+tzqq610w/Pb+avtAVf3a+FpdXFVvHvvzltW
zxvuabn/716jnpePrdWFVnV5Vt9uKelarZea+51dVq6qDt6KWKamJj8Gm2gfnMaVeOY+p9dM96d1v56xx03vyho
UyVbVvkj9M8sQk90lyXFXdZ6OW39GjWmtHzPybq99K8sHW2j2SfHC8vQvQhOS/KEFdNWXZfxdTs2yX3Hx7x6f
H2n6rTceN2S5OXja3dEa+09ycKt23eT/PfW2g8leWiSXx7r3y6v21ZZ13bs6FtJHt1ae0CSI5I8oaoemunVueyEJJ+au
T3VOhehh78yyXtba/dO8oAM23VSdbbWLhj76RFJHpzkm0n+LhOqs6oOTfIrSY5srd03yd5Jnp4J1bgDq9UmsaMTI6Gjk/3jtGx2T2
6tbchXkqOTvG/m9oaM3iGjbP99kKyxxqQQZ7VaLZ3JqagRwtBbmK7JPtm+Nu
6lfWclmm9Z1utng8kuV9r7f5JPpPkxC2qZ7VaUlV3SfLjSf55Ztpe8b5oQcZgU+2D85hSr5zHZPrpnkyk387jtGx2T2
6tbchXkqOTvG/m9olJTtyo5ff4SrIrycErpl86q44kF+9pXVa+dknel+To3vWc91OTvL8VeZ
buHWbqfWvMrzh2Dav2xS34xS+ktwqyXlJfmSKd5fPwD9Ogk7xqnTbHOyffwJN+f5HMZr/E21TpX1PpjSf73FOtMcmiSLyY5aKz3XWOVmxu2+Xg7l1l+pHkK0i4F9LeKXl1gpKsC7+eahlxhLi4zhmRnbk0bpHZR1nfe6GtGpWIq+1oKIWXBg8Md5zMk6NP3dxcdtL0G2P5b/YUY6jnDeYjA1VxYy9JxYTNWc6VMJnJL8fCAmUA0k/0xyHK7E6k5zepiYJP/rv
RtWUuuqdRrqxpOfqfDotWZfY5HpvmSqD7RmUbVxVq2xXs2aj9s67ZvXtYU99HVKuuo5AjB+cfjtWa6udUbqyCbXddJu59ojcPnmwVl0/6/4Wdfq0U1OnaAgVztqJXHbJtMunJ1xu51nHQaulGhO9Ofv4ZXh30RfbeLjtj1b8+JfsvBXf5TfbedGnVTlZfIOq1T6vgcDwZh+Wdk9mvtlBXTwRMfbH5L2rZbRV7fgnpTw3yz12VLwqQVpRB/hmYppXQYYzTjfwGJiG6nmu1ZpCYdSDLTeVWctd7TZfvCrsLSU9Fz12Yj9o0Ll1pe9Ot9BZ5HNwRmn2LXR7rFqmR2JVQjtBQdnMLNxh8aGtexPIq0Ybz6hYYfW
QXGl6/qz0S9TeulnlHTp+oC9Tge1ViiGRdJRJC3MYlI8YLoYSD4aRNNZTfZdPDQpF/Pq1R9Vfpp5qkC65EPMtOXaYjbtrujdvlYE3CZIa1cS2iwodHPCRhLoqeklxE7T/aO+hFH8DyE0+ppL6MYOiQdcZwtbyvibMmaqTu2e/qjjVRaeaUEV9o4TgJtP7X8xLhQcMtzWbhuSbNpp9SJCoHdl7vKs3l8eRnumXeDa3tS9Vq/vk5O3s7q6oJXpkfl5pK7/SZgfaCnAKBdGXjPBsf/3gs7Ys4wiSXaDyUaEbJQL41btjeOdm/OsvDxTQOO4D4dJlrNlaUl8ltu4b53R1iSiWLpk6q+6TqcGbpAZeY3HLpH7u9/3Q/v2P7QRcYJT4tDjJ7VtGzVP/QDR/NpMzLjLbLw0hm3M7bYXVcq/5C0/DTeUTUV/d/CMTXv8bEJhWVTAnYmWX/2lv4wmPK0xGaTmTeblJs6AyiVObNgcurj4T16Qj3PzA5EnGzA7xODeRR4DaUX7jfJx47m0xnoHJ8u6OIlqbU86KY8TkV8wpQq/0JznHfWBRcqu0vWfXvhOwwPf8dXB6yTHo2SI6yIgZmSl2qPuU1lRGMTutA+RG4/JUHdL9ZXq7aRLbZ7Kn8Fo+ltZdD3yv2RMuJbV4Yij0V0zrz2zgN6BhA5x37l+u+sKLAz+2Ve0vrs6vQYZ+3Gf1Nv9Ovo0/vjZ2tPQexaSHNxLGR8QzMkbRxnBgXC4Pu2nqtKH94xV+KL3aqa91p9G9O98L7Uq6t9x57+1dv7hdAYpqq7zbUJuVYSqYg3SXvWZ8p8Gszh1dsb8fb6+0MgcVdHy2CYIYiAnHA3NnLtJL1XAaWDCuN5qGUl0vQhXnBFLMyMFLL2jB6gq6Wf7SVS7hbSPt6QRl+H5SDmXU6KypK1rSpAPDHpDyOcW6xkTmQsw8ivM9ZVb5OV8qa4LpBrE8tezPPQkqLs8iHWKWfJZdoZpFKDDC6nHoWCMjYOcw2eTWmOo9sW03NL2cWW0nfI7F9PTZTAKk9TVfDSgzGUhdrVCO++cnoSJkhB2YjUgxqxGnXNxjh3Vy+b8ynO4AzX0MPVK0ePWfNsP55uaQUKMOq3cVOt+XJrK/dUnBy70IFaTNqvKEOs0U2qvBnkExhB9rFd0lSoGxbWKJ9jqnNNyMBwRS1xzoTu6EDjVQKoE73YCNiQvxmo3x5lyt6fQxbxpxLM1umVQm2AlDntbrvNBmOKR9qC8yiRt5dfHM0mNwsSp63XnNwJvhP03SOe7JVGpxUCjlYGaHx1Lfmv8LjYjwXB/LXaJVC0Mq6KDTLjBGlHpGYXEcl5oW+j1Y4uUDC1GJgWnD0lEJmDlgR3OxPRmUzbMLgPbVfYbbbVwKAo6xPXUhWLz3ZsPwyJrNG7qAHq8ndhUU5dkrlb4FVe8yb8nOjSfX6jHE1FHxf8lSRnblaN7NOCnO2R6pvK3Kp2Rfv3mrFjh3fhR8RZ9T1e3NmjcSfwTKSH4lm2lWj5Fs7PLA6/QRmwkOv6JuhHpmONLW3c58tWQ5mnAJTYUtaypTFqGYbgyGLZzqcTqnFK8+9J1A/0LEQZxVDd9WRKFyn72K/3gDD3RvrtpNVLubJDhXDQK7bb7wkmD9HYpD3lNfXGrSJwBD0g5vi9xMTUT7klWTDdXELn2Y9/T8rYhm4iTXWP7erehXZ8f7KebE9sGnL/Nnm3Hc72fTiUh78JRg7HJvlNnk5wC9B7Z0Xp7h+JbwdqZJUAfvLfJ36UKwnOp7PJQJE6ZjqPPIUaMwIU63RVtmV9Op1RGLMSN+HRx99d+7arfPyH3cWFQLhW8hnRN4JvPTgBfzPVGPqhuNrk5M+7SUgAVc1+WcbJGgCpC4l6MnjRrpeS9fNNS/9r3HGyfnehR+aFZvGt5GFMJJcm6KrMHkZRUkwGE/YASqtQp7n2l1qLJ0FTHjW/bVDCSMnAzY/Rkw1A3L1Sv7t/N1JaJsxhBi5rVyLxWNqSEZvkwHz22lV+lDq7cSGNbsxjBg/2+e6R0aqs4TgabZNq23Y0QZKcP5fYfp6q8vMdlMw6Sm8V3ppnSM8fAsdAB/WUEGt3aROSqiUrT4vWZT7YZVzI6RUJLX3w67fcWwkgfmvTjmdh7xr8+Wwhl9DN0sJNPmLeWDTnaSpPuNhI38Y3ayrmqEHXD1AbRsKlr7KmHkwaqZtK4p7AeUm4GGwqJ+vbk2A4HQtuk4Nf6LJTWOyyMRmyVdpP/qtHQIvR6Tny5ZXrvZN6TgbFTW6xQC3VKvRsvUn3AFvABiJDCmxMYhbOfABmeJKKbRkPq/v5bjqn4VL5mPnHf3HI5RrUtsdOhrO4L+HMaU7PDAUJ3vNHB9aU8E5K1mAfBRnz2Lhe9XW2rEE88ZO0hHVGFFF

JxmPxrmkqv4myb1mpp82875kV1W9eNwHL1oOn8fn+8A4/bVV9flVguA7Jvlqkq+N6/G11trnxsfvrKq/qaoLxmXcf
Tzi5KU1HFFzUVU9c5z3kVX14ar6P0kuqqp9x/nOHsd2/3WN7fgzSf5q+UZr7YNjPbu12nzjgQOPTvK2cdIbMxwE
kCRPS/JnbfAPSW438/7tnWMdrEIoM133S3LuvDOPAcfTk9x3TGBPaa39fZIzkvxGa+2I1tpnk/xZkheM81yU5Hd
mFvPt1tqPJfmjDDvuL491/FxV3anw1toRSa7N9TvXgUkubq5SXa7vt6iSfy5CiAnuHY5K8t7X2
mSRfquGQ1p9MsiPJDyf5LxkD3aq6RYZByjNaaw9K8oYk/6NDzccVgtex+mPHAdBb6uqT1fVm8Y39bsb5Nzgi
JFxYLJj/PmdY1j7iap67jrq2zF+6PTH42PfX1UHjPetd1D0kar6i6r6TA0fiP3MOCi8qKruPs636sBwRU23SXL/1toFq
9z3C1X112ONJ2d4//aacRufPTPrO2Ogw/b1/iR3Gfe1V1fVf1xx/9WttYckeVWSV4zTXpnhKI4/mpJK+fmf/eSR6
f4cj936mqW1TVgzL0rAdmeF/y4N3Uc1Vr7cgM++Jyj/qdJB8ap5+5+e5D+s8rgLklyR5HNV9adV9RMz970pyR+21h6
Q5GFJLh/rOCHfTSmXDjIUleOJ5F8fNJvjKu64OT/EJV3XX2iWs4tehurbVdu1mv4uGhKvRs4tehurbVdu1mv9bhDki+31r473r40yWHj
z4flhkcyzd53aQ+ok30zy+qr6AOmq8Uh6bvd3aSz+Mk96Y5MdmZjljlj/H5Rkk+01i4i4fk+l/SnKXJI9J8q
AkZ1fV+ePtMq8OGHD3W0/uPzBdVIWhrJ9hVrJUhqRx/tTWFQXWxX3ogWbbjwbn/y/3fvnw79SUpk/Gh22zeXL24i5fc
6oxd/Z4bcBYSHvtzGiBKgBUWFQ/eOsrn7MhGfQ/MrYNCL+R6WmWt2PcVYPd2nFbuRC0W/dkKiTmRV1aRimx5HGHI
RZnhrS9K/SUGXg6KLqEHHWyRTKwyVYFjWpaPJK/h+i5Y/1GrS/e+fdedV/BkgIoy7o2ZgbeXmEVgGlNF1u3u0pjgSC
fpTS/qbUFqoA9xwAdyPX3ebqvJqP+Y3oVqbN48NZzUG7rXMkXtLmm1+nPxg2OJOQ6Eed0uRQHf/7b+u3zaL6Vcu
YthamYsDYDZ+o0v62Ku3ZXSsuXPcv4nLzuCjJu0iJnDTsoaxycnXXtBb5iYzFHW5K+XFscSbhHbuz5JvaeIWsO5Li
ArA7kpbdY6dUeKexrz22cnTLK2AHdrmH/jZrWcazH4LFV/ttq/uyLqoRdPbV8xVHN9zzHJ8YTNlXOz3xYP5ELN5jJ9k
mHzsAtOEkeuAFmoCb1lcLuE4BkFrlVP5cQbNF6xzcS4GdTNgtgzA5aEHu8ev0qUzgmmTSsJRa/F0LqySGZ8tqW0KGAQ
Y7HrBpO0IN8L4SR4I6c9Sb7hI+nzp2FDHo2XaI7W02SZXcQXIxewHjLhWkqmScaFjIq7+qnLLjM01qOcXxtwTthLPEnyk1UWYR
wZ8DlPjYoLb+oHbfJ7rQP5rRJYKtFSGjn2KjL9qYDZdA0HsafuQIAyVVUI6ULd5WwnQ/ut+kHPG1TjUR00gsqhGpGZ2uOQtDd8Eh
+DDgl/D+UmX+CZOOHlbBvAzeLb8yg1TxlBUbiXk3QmaORAR/q5sm3b5YcHnGThG3K/Y6TtK93QSnPnxyoZ+T/UmK+C5ES5P
MNwD/ZJ+IOQmxUb/5QD4wZWkiNafuQKFRjoNTq7ImNbMqGSmj5/wSDSRTm3ZqJqc7rywH8oMr3/b9VFZMTQMtyQUF3tX9
fp+kDE61G8hvU/kqQg9QsVtxQtKfRh8JnmPtWUn95k+d8HFn6VpjNwmjAh8Gq9Fwe5F4o59Ngy3ti8iypiYNjcTnNmJ4n3o5c
3b8BFMW0c4A+uuIOzUe8lN2TB84utsYX0FIQVpc9rcHeMz8MNIwfXR39pYdlT4kzN7YRqb3UzlBBWOaBy/fEvqIWwbMFmyE
NeXq+AOTMRKsYS+1KCiXmdRlOkufZq+6IdT1mT/2ZIHcN6bGvjh2QO5fx7lzlVbxTDfy7UP+WFNeQKhcJBXKWv1LYk
8z6c5+Z8WfedJ1iGftKe9Tdcfnm8Tf/m+tU2k8Z6bN8A6xKDfT1zLrnbOyZKZlcK0sfZ75UmBtrP8Hcp6r/ZGs7wt1L1rH08Ba
ZrJNxNrGFdmqBtXiVRNt+dVW0fQ9Ypu8Ff8DbPdfF6kOZKm/oNfnS+MWHJZx/mVzfmSv6HqMTvyuF82GWrPmwHoaoDzO
t+aIQQl2fYIQ3rcz7VfmBz1U5xzUoZfgXQvs1b9OxnGSSsbYf4gQ3GbfZ9vtmNk8Cvz9T0rmXHyGSZDmQO9ZxmHmjOcdfYYY
C+DaAYvbeVzVp05cbmOcR4X3j2Sf2UHoiQWRp7JqFrDNFbgWsj6Y7Wwo/DzSbiq6bNo3ZTLGBIFnKJdgRfqtFwHTaZTb
uDs5OrDwTK6d0OEA/qpIcy+2Fvni3O6VLSV7iyMBUpfRrl4O3y/NxlFL1vAjlTuq+BgImyr7P/yegPx5vLovjc8t7l/Il2Gf
cD2PRNs4HUK6Vpj4CWfhZpFk8tzN1CePBNMmqi5MLhSJkIelYqAAAAAABJRU5ErkJggg==
sRVelqNq9oIVJDA+IR8a5mg4uQoTrqUAuKFPj3eRkLKFvygl88+jE9avHNGnpCqoakjcHqL3WnVyG7JGrU3dlJNX17
Gp2s4hO5yEkQkXwHZ7GeD3F6A3Z+jUjlU8hqqG3pLS1+x+RXZ91eu13E3HBiagKr5EykNQDE1zsPEDT8tAGM+G9mZ
T1sd6WHYRtyTz72sfbjLMthnfdaeHHXeY/+s/wkMLTqJbxTtcTnvY1LP5qDXZzHH1vpZg6SXD+qmnz6ZVfIX9cKzjhSGl
yG+H0N3HW+ta6r0N+G6S1cMxNSf1pCZZVyU5qYhJL8HH/Ke7K53yHuX1kqoSt+yu7hOZxJG+5gQz4VRMiLcMyiLOfdOWl
V2JKSX9Xyi2wEO2hPO1sJbXLB5Fb9d/fQzO3cBHOkVZqFJE9KB9eClDgFgrbZ6YQXg2dR2NsJYfXYYn8Cm2gVfGZYJXT
AAAA

"text/plain": [
"<Figure size 1800x360 with 4 Axes>"
]
},
"metadata": {
"needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
"fig,ax=plt.subplots(figsize=(25,5))\n",
"\n",
"plt.subplot(1, 5, 2)\n",

```
 "sns.boxplot(x=data['Age'])\n",
 "\n",
 "plt.subplot(1, 5, 3)\n",
 "sns.boxplot(x=data['Annual Income (k$)'])\n",
 "\n",
 "plt.subplot(1, 5, 4)\n",
 "sns.boxplot(x=data['Spending Score (1-100)'])\n",
 "\n",
 "plt.subplot(1, 5, 1)\n",
 "sns.boxplot(x=data['CustomerID'])"
]
},
{
"cell_type": "markdown",
"id": "6718d9d0",
"metadata": {},
"source": [
 "## Handling outlier"
]
},
{
"cell_type": "code",
"execution_count": 11,
"id": "cf142f0b",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/html": [
    "<div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
    "      <th></th>\n",
    "      <th>CustomerID</th>\n",
    "      <th>Age</th>\n",
    "      <th>Annual Income (k$)</th>\n",
    "      <th>Spending Score (1-100)</th>\n",
    "    </tr>\n",
    "  </thead>\n",
    "  <tbody>\n",
    "    <tr>\n",
```

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

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

```
"cell_type": "code",
"execution_count": 15,
"id": "87d6fd39",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "CustomerID              -98.500\n",
    "Age                    -1.625\n",
    "Annual Income (k$)      -13.250\n",
    "Spending Score (1-100)   -22.625\n",
    "dtype: float64"
   ]
  },
  "execution_count": 15,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "low=quant.loc[0.25]-(1.5 *iqr)\n",
 "low"
]
},
{
"cell_type": "code",
"execution_count": 16,
"id": "da5d37d2",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "CustomerID              299.500\n",
    "Age                    79.375\n",
    "Annual Income (k$)      132.750\n",
    "Spending Score (1-100)    130.375\n",
    "dtype: float64"
   ]
  },
  "execution_count": 16,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "up=quant.loc[0.75]+(1.5 *iqr)\n",
 "up"
]
},
{
"cell_type": "code",
"execution_count": 18,
"id": "d3f2fda9",
```

"metadata": {},
"outputs": [],
"source": [
"data['Annual Income (k$)']= np.where(data['Annual Income (k$)']>132,60,data['Annual Income (k$)'])"
]
},
{
"cell_type": "code",
"execution_count": 19,
"id": "5aeb5451",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"<AxesSubplot:xlabel='Annual Income (k$)'>"
]
},
"execution_count": 19,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAAWAAAAAEGCAYAAABbzE8LAAAAOXRFWHRTb2Z
0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAA
CXBIWXMAAAsTAAALEwEAmpwYAAANNElEQVR4nO3df5BdZXnA8e+TrAOBihqT0rgwLrJYCihCoyOFdij
aVh2rqJ2RiAwzdmrrlGV12qFS/4ntP52hY5vZtnQobR3bDJ2O/ChFpbFaZBxHIVEI0QS8LajS8FPPL0j3M2
XC6bkE1273Pv3e9nZmf3vvfsPe+bvfnm7Nncs5GZSJL6b0X1BCRpuTLAklTEAEtSEQMsSUUUMsCQVGVvIxmv
WrMmJiYklmookjaZt27Y9nplre8cXFOCJiQm2bt26eeLOSpGUgIr4137inICSpiAGWpCIGAJamIAZakIgBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkoos6HqaD
CaenMzMzQ6XSqp7FoZmdnARgfHy+eycJNTk4yNTVVPQ0tAwZ4QHQ6He7bsZOnT1hdPZVFsXLf9wH47pPD
9RRbue+J6iloGRmuvx0j7ukTVrP/p9+2/cUT2NRrNr1aYChW8/cvKV+8BywJBUxwJJUxABLUhgFTHLAkFTH
AklTEAEtSEQMsSUUMsCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLElFDLAkFTHAklTEAEtSEQMsSUUMsCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLElFDLAkFelLgGdmZiZmenHriRpUS1lv8aW5FF7d
DqdfuxGkhbdUvbLUxCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLElFDLAkFTHAklTEAEtSEQMsSUUM
sCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLElFDLAkFTHAklTEAEtSEQMsSUUMsCQVMcCS
VMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLElFDLAkFTHAklTEAEtSEQMsSUUMsCQVMcCSVMQAS1IR
AyxJRQywJBUxwJJUZKwfO5mdnWX//v3Y3dDqdPpsOLHWT2NZW/Fj35Ap/NDn6s6qNPpsGrVqiV57O
c9Ao6I90fE1oojYunfv3iWZZn3BJyZNwA3AKxfv/6fvEyQ2Mwo3Ykqxfv/6oDtHGx8cB2LRp09F8+rIwPT3Ntv96taaeSai3ZRtv96tHoay96B409i8h
Un+1zVQUv53ZDngCWpiAGWpCIGWJKKGGBJKmKAJamIAJamIAZakIgZYkooYYYEkqYoAlqYgBlqQiBliSihhgSSpi
gCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIG
WJKKGGBJKmKAJamIAZakIgZYkooYYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKA
JamIAZakImP92Mnk5GQ/diNJi24p+9WXAE9NTfVjN5K06JayX56CkKQiBlhSihhgSSpigCWpiAGWpCIGWJKK
GGBJKmKAJamIAZakIgZYkooYYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIA
ZakIgZYkooYYYEkqYoAlqYgBlqQiBliSihhgSSpigCWpiAGWpCIGWJKKGGBJKmKAJamIAZakIgZYkooYYYEkq
YoAlqYgBlqQiBliSihhgSSpigCWpyFj1BPSMlfueYNV9v1M9jUWxct/3ARia4xlYue8JYLx6GlomDPCAmJycrJ7Cotqz5wAA
CoZmefAmB8fNhidvLIf00uAzwgSYlvfjSyAzkgEBFbM3N99TwW26iuC1zbsHJt/ecpCEkqYoAlqcgw
UUMsCQVMcCSVMQAS1IRAyxJRQywJBUxwJJUxABLUhEDLElFDLAkFTHAklTEAEtSEQMsSUUiM49844i
9wLewbjrzWgM83Wg4zUd99ZUsOorgtc27BybUvn5Zm5tndwQQGuEBFbM3N99TwW26iuC1zbsHJt/ecpCEkqYoA
lqcgw
BPiG6gkskVFdF4zu2lzXcBnVdcGIrG0hb8KPA8CSNsj5OgPbDx0gqkK4uffjpMUT3vSiPR4j4YDeRyYVIVUpH
9fLghwVjBzS5D8qHKCLkYTb0eaasqYuS/qZ/aBvIxMeh6nYo8Zhd/4wGBtdHsHnt3ugHfTQLg2nOgLl4r9e
2vN7xz2uQGsfgJ8cN8xhkh6kQUWuw/cuhvbxcUswnLYyYNpKgaEs7aEZS7jMMO5jKXTe4eD+Gw6yRp/B8Xd4oRfLn
C4iXJkNWTk8PjAMHVzpX/cy3vpNfl38SMcVZ4rLX4aZVYdA9cQNXgMqQo+e5FQGDY1lHAWqPAv/qYRQGaRxsMGaKyQpR3QbVvkapRtSGY2F7y7
CLGxGlJDEObXmfrOBt/DM4MStJhPDwzkhBPGbdu5QMEKgZm5TH4TZ0nlwzfn2h3aA/F8A7IA2rMM7zpcQvH
ft6YM9rwqGLBoY6+hDsWbhZAMv+6R/7dFrFLCYxwjKBV9eGSFXu7IIq7a+2Sl5aKYwUeuAhXaXo/zVgAu8g+
fWSnGfgaDBxL4QbbPi0jMQ8ld0fYIE9aF5J8fLAvLZxm6yp0ubsKLZ0CN2CF/QNwZxu6zBpXHbSYNWyl3fL
fYtVTBv/u/2qUjoFdQGzeJ+KlM/DTvj0XyHkPNeIuIHmzYmNMx0qL5p7i6nLLAvjYzgGVFcBLOJxTgdbMN/bLFJ
y/s3EdvOrf5QXcwTGIa0uj9cncqBMUoInVNKj+RtW1dJxmL5JYW1glfX4GPbI4+aCrlSrpyLtNhuN+o
dXvYVXwNsbUzAL3lTf+FRg4jwsdoD9TgIOm8dxMIbUyP9rATq6Ktd3a0TLWl9/ZVLHhq/NmC+aUr8v8+0Mdn
4LuyqAkA4JQ0D7eRPMPIlePO6xrQyHWrsM/6zQNGsgoqFOeGN/8Q+IVy4Gi2U3BH6g3Q+wXRLgIL00sgv6kvk7s
wl2JQpr3Lhsp8BydLSqp4YxYaQz2/xzQg0p9cKbcS4NZlmh3D0WLQ7c4J28FcTDf3Km7IM0G7lThOT2vdx0
jxuo/WF3iIN36LrjUA5jtIN1lTpczXqEYCFrD3yIK/EwMCIKuq1j6kzuWYQJQUNfeTZE9DrfgH81RLbpJ/J4Xo
iEaRmKbGdPnxJtS/vDVy4tgFCHKpBbEVzvEGnSTxlbgFtFvUMzNtZ/8mU0bUdr0bNmgHBFs1xwBxKYAe4Iki1
ueYqkr5eGo61/qAmtpAyHa+b/2Zvmuk7vbMGoq/0ZuqrspYuvQN3dq6yOdwwaXqMQu9rBxL67+8ddC5J5aocjg
R+/BONsfqjgPp7GoDlcBKTp/O4fM0SwEZpmqJHjSiVt4uJ8QDMYxuwtnJivWWfBtxYEiXrj7yEsSGfUYTJlKfOTLUQ
gAMjq8dwFlzy0gOfTn8HXsqOwvgn6iQw9xtTY3q0IE56hThGNNFYxoTVLDzjuc6ZFPqAkG90Fs0nCH0mDOEL
AAb+DTtN5BmQhCnYcXtvrNj4VWuKmgU+h/+ygfiBgT7Be6iUvhu6hJvLT3VgPHZ6pNgGH6TBPbb6k2OR1
Wg7TLZfZb73Snmfr6SNzV43rkydkYeFp2u6qsHTCoWqSr30ofwsBlSnHOhqWnHYn2yfVeOpmDPtodB0zM0Ll
FPOOVqEAKsVmpBsRGKy3cJqVT9qHr0MS4ax2dyiWW2+p0JYbeNYaFCh60JPgCeiSf8BH0nZmLjEnCzxWKuffqAh+GwTBqLEAVDaUJ1zl3sY0LTMR1VrMphpZ9CnOGYCP3fw9gxqrq4w3HPBtt7GBK
FCePgfBR1ZlKqvXq2QAuIXF7YPFx8mevfQ0SklRO7x+J6HDpXDz2yFvELZYmb3eGKuovqNxpNxyxT7Jhc0XJT3
HbhKoxQOKUqBNB0FZFRwiVkqT1FeTDjbhcQ+V7TV9NLdpxbUXHB0sW9vCD+TiHvvANNtxwYspphPGXj1bMyCNxF
k0BszbJq/x1vCtb3KNf9Ymrgd72xtFMWk+KNdr20mZKnhN8UGXLfoUX+J12u3k9+uSDhVdVgFQsPzS1g5xrKLCPm
ouHXu4LXUWKHBiaDj9lpQj0VYCtb0eFodjPIGlqONzgVqhSaavSlfRM5fK43sqRKkbyHaH1hg3H6j7Z2CEbUPBt
KFJSobdTvhDsEWgMMV9hqrxyY2rKbn5jfcWuZZmjcaqL0+2vHhRmBYhG3vJXc29W6HfMRwwTyx4bH0bKUCmzf
Ytr/1V0KMHyrIBMmsdKi7d4OnDsPzxzrsCBfQSKvjXy/dQ0bGNOz8oDkSZhdD96T5fz1/vAf7HzrLzfr7CvFDp
XxV7nQxG2nLBJGG8Lnt9kFG06byBsMHLxdMIQYs0s+5CKdrOaIIIy8OIyLF7FHl2/BMMRyT5b8BZ4/6JAfvE
Bb7Hx8JPN+XNlRzYG26ZIH8MsrTnQQyf6uQbJAdTbQKqrvv0DLUzlqhI7QTrr2pY2BTLmn2P9upfD4Onvj4DIgt
bMsbb0rCd2mpeBFgbfSqQs3yJhEuvF0MdhA2R9/F8NFdzpGbi21u9Kxk0ohQGYXUQhr99pqlT36c3Fv5dqHM
0sGnqBu0oHrX2tM3gN1DH9oNbvUj+pYIcaU5Uaasw2s5EEJMYz9t5HDEIBbOkM1Izp0ZBDMV5bG1XXFHr86NQC
YaDxXXaQ48kJdn8k6WkAWAuIZfIp6lKDgKdZQ5RbYcZBGz7oL3Pyej7DaHP8dDFhAeu6DZcOv4gZPUKlvuM0
hBnKyL6Y04D8/3u0CCG4e4Pwi58K53JH8g1mzGv9kq7+KL1QlAPoTwljJT4DDaA3Gm5rAwPdNhWIVhqVu1e
QYtkU1JlESmyjdz7RKkK4u3+mRvmNcF4pOG5fJ3bgPlsUQFgQl3ug+nLJm88/0hk2LNOJ6mpycCHBwKUH7L
fnRkzi+ou8bL6tKXvqzK2nDlPl4MrQS5ufcHeSsS4u7QJ/DQTRhoKqnUwY3s01gkWaZq7RtvGAA+uo4K0PhaV1
GGvOE1u3VwoUIzRl5rsp5MH8FoS5vMqZyLaJpVnLuRdEmwnSEfnNO0T+k8ZfbhJYcuoH+kuEYKVUGJeK+v"

FoYwHVl5kC+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/+7z3I5JRFwJvBW4PNvzDQzYugmwBERwN8COzPzY
1133Q5c2X58Jc254aGSmddm5imZOUHzA4DPZ+Z7GY21fRf4TkT8bDv0BuAbDP/avg28PiJOaJ+bb6D54eKwr6
vXodZzO3BZRBwXEacBZwD3FMzvqETEm4m4A/AN6Wmfu67hqsdVWfPO86GX4RzbcC24H72re3AC+l+eHVN9
v3q6vneozrvJhnfgg3EmsDXgNsbb92twEvGYW1AR8FdgE7gH8AjhvmdQE30ZzP/gnNkeBvHm49wEeA/6T5Qd2
bq+e/wHV1aM71zrXkrwdxXb4UWZKKDMwpCElabgywJBUxwJJUxABLUhEDLElFDLAOioh3RERGxJkF+34k
ItYc6figiIjzIuLG9uONEfH7h9l2Y8/ttRFx5xJPUQPMAKvbBuCLNC8W0ZH5Q2DmcBtExFkRcTfwgYj4akRsAM
jMvcCeiLiwD/PUADLAAg5eg+NCmv/EflnX+MURcVfX9X43t68Mmzs6/WgblQfmjpx7jwTb6+lOtB/fFhHb2uvsv
n8B85torzX8N+3nbomIVe19kxHx7xFxfzuX06NxXbvvByLi3V3r+UJE/HNEPBQRfxIRl0fp7fbrY2ImyPi3v
btOZGMiBcCr87M++e577ci4jPtHDcCnwCub/+M7+e577vHN7fi5NNdr2AO8k+YVeufSXNfhurlrHrRj08
CrgCuAV2bm64Abgal2m03An2Xma9v93DjPfNbTvEruWSLiKuDXgUszcz/wY5qzcz/wY5qrjK3IzP2Z2enafCvwiwv4M9AI
McCas4HmQkG07zd03XdPZu7OzAM0L+uc6Lpv7qJJ23rGD+XqiLgf+DLNRVHOWMAcH87M+7r31x6Fjmfmr
QCZ+aNsXvt/EXBTZj6dmY8CXwBe237uvZm5JzOfpHlJ6pZ2/IGuNbwR+R+IuIuI/m+gEntfvqto7mUpzdrgDeDLyrf
XxorknwKuCqiPjXiDi3a/vHaK62pmVorHoCqtcehV4CnBMRSfPbHzIirmk3ebJr86eb86d59vPmyXnGn+LZ/7gf3+7nYp
qwXZCZ+yLirrn7jlDvPFYx/+UFOcx47+Mc6Lp9gGfWsKd5/7DPM5+njv/HTRH3qfQXKiebK4tvCEi/ojm9MMt
wOnt9se3j6NlyCNgAfwG8InMfHlmTmTmqTTxuOgoH7s0p7K/HxKt/o3dHxKXt/o
6LiBOAu4F3R/N7+NYCv8TCrnq1Bbhq7kZEvGaebpXMcxpDy4
MBFjSnG27tGbsZeM9RPt7NwOr22/cP0PxOLoA7gbGI2A78Mc1piV
IXQ2sj+aXOn4D+J3eDTJzF/Ci3g+LL0NL+ZYUv72HN+GfjUAualEeLV0KRjEBEfAn6Y
mfP9kK53242ZubnFn7G7g7Zn5vSWaogaYR8DSsbmeZ59TPPy7um+0p0Y+ZnyXL4+AJamIR8CSVMQAS1IRAy
xJRQywJBUxwJJU5P8BGkL5HVhWiA0AAAAASUVORK5CYII=",
     "text/plain": [
      "<Figure size 432x288 with 1 Axes>"
     ]
    },
    "metadata": {
     "needs_background": "light"
    },
    "output_type": "display_data"
   }
  ],
  "source": [
   "sns.boxplot(x=data['Annual Income (k$)'])"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "f37589fc",
  "metadata": {},
  "source": [
   "# 7. Check for Categorical columns and perform encoding."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 20,
  "id": "26658b84",
  "metadata": {},
  "outputs": [
   {

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

```
"execution_count": 23,
"id": "c4e8fbc7",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/html": [
    "<div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
    "      <th></th>\n",
    "      <th>CustomerID</th>\n",
    "      <th>Gender</th>\n",
    "      <th>Age</th>\n",
    "      <th>Annual Income (k$)</th>\n",
    "      <th>Spending Score (1-100)</th>\n",
    "    </tr>\n",
    "  </thead>\n",
    "  <tbody>\n",
    "    <tr>\n",
    "      <th>0</th>\n",
    "      <td>1</td>\n",
    "      <td>1</td>\n",
    "      <td>19</td>\n",
    "      <td>15</td>\n",
    "      <td>39</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>1</th>\n",
    "      <td>2</td>\n",
    "      <td>1</td>\n",
    "      <td>21</td>\n",
    "      <td>15</td>\n",
    "      <td>81</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>2</th>\n",
    "      <td>3</td>\n",
    "      <td>0</td>\n",
    "      <td>20</td>\n",
    "      <td>16</td>\n",
    "
```

```
"     <td>6</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>3</th>\n",
"     <td>4</td>\n",
"     <td>0</td>\n",
"     <td>23</td>\n",
"     <td>16</td>\n",
"     <td>77</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>4</th>\n",
"     <td>5</td>\n",
"     <td>0</td>\n",
"     <td>31</td>\n",
"     <td>17</td>\n",
"     <td>40</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>...</th>\n",
"     <td>...</td>\n",
"     <td>...</td>\n",
"     <td>...</td>\n",
"     <td>...</td>\n",
"     <td>...</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>195</th>\n",
"     <td>196</td>\n",
"     <td>0</td>\n",
"     <td>35</td>\n",
"     <td>120</td>\n",
"     <td>79</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>196</th>\n",
"     <td>197</td>\n",
"     <td>0</td>\n",
"     <td>45</td>\n",
"     <td>126</td>\n",
"     <td>28</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>197</th>\n",
"     <td>198</td>\n",
"     <td>1</td>\n",
"     <td>32</td>\n",
"     <td>126</td>\n",
"     <td>74</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>198</th>\n",
"     <td>199</td>\n",
"     <td>1</td>\n",
"     <td>32</td>\n",
"     <td>\n",
```

```
    "      <td>60</td>\n",
    "      <td>18</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>199</th>\n",
    "      <td>200</td>\n",
    "      <td>1</td>\n",
    "      <td>30</td>\n",
    "      <td>60</td>\n",
    "      <td>83</td>\n",
    "    </tr>\n",
    "  </tbody>\n",
    "</table>\n",
    "<p>200 rows × 5 columns</p>\n",
    "</div>"
   ],
   "text/plain": [
    "     CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)\n",
    "0             1       1   19                  15                      39\n",
    "1             2       1   21                  15                      81\n",
    "2             3       0   20                  16                       6\n",
    "3             4       0   23                  16                      77\n",
    "4             5       0   31                  17                      40\n",
    "..          ...     ...  ...                 ...                     ...\n",
    "195         196       0   35                 120                      79\n",
    "196         197       0   45                 126                      28\n",
    "197         198       1   32                 126                      74\n",
    "198         199       1   32                  60                      18\n",
    "199         200       1   30                  60                      83\n",
    "\n",
    "[200 rows x 5 columns]"
   ]
  },
  "execution_count": 23,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "data"
]
},
{
"cell_type": "markdown",
"id": "8a984913",
"metadata": {},
"source": [
 "# 8. Scaling the data"
]
},
{
"cell_type": "code",
"execution_count": 24,
"id": "8735fb43",
"metadata": {},
```

```
 "outputs": [],
 "source": [
  "from sklearn.preprocessing import MinMaxScaler\n",
  "sc=MinMaxScaler()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 25,
 "id": "a3284529",
 "metadata": {},
 "outputs": [],
 "source": [
  "df=sc.fit_transform(data.iloc[:,1:])"
 ]
},
{
 "cell_type": "code",
 "execution_count": 26,
 "id": "2f29a211",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([[1.        , 0.01923077, 0.        , 0.3877551 ],\n",
     "       [1.        , 0.05769231, 0.        , 0.81632653],\n",
     "       [0.        , 0.03846154, 0.00900901, 0.05102041],\n",
     "       [0.        , 0.09615385, 0.00900901, 0.7755102 ],\n",
     "       [0.        , 0.25      , 0.01801802, 0.39795918],\n",
     "       [0.        , 0.07692308, 0.01801802, 0.76530612],\n",
     "       [0.        , 0.32692308, 0.02702703, 0.05102041],\n",
     "       [0.        , 0.09615385, 0.02702703, 0.94897959],\n",
     "       [1.        , 0.88461538, 0.03603604, 0.02040816],\n",
     "       [0.        , 0.23076923, 0.03603604, 0.7244898 ],\n",
     "       [1.        , 0.94230769, 0.03603604, 0.13265306],\n",
     "       [0.        , 0.32692308, 0.03603604, 1.        ],\n",
     "       [0.        , 0.76923077, 0.04504505, 0.14285714],\n",
     "       [0.        , 0.11538462, 0.04504505, 0.7755102 ],\n",
     "       [1.        , 0.36538462, 0.04504505, 0.12244898],\n",
     "       [1.        , 0.07692308, 0.04504505, 0.79591837],\n",
     "       [0.        , 0.32692308, 0.05405405, 0.34693878],\n",
     "       [1.        , 0.03846154, 0.05405405, 0.66326531],\n",
     "       [1.        , 0.65384615, 0.07207207, 0.28571429],\n",
     "       [0.        , 0.32692308, 0.07207207, 0.98979592],\n",
     "       [1.        , 0.32692308, 0.08108108, 0.34693878],\n",
     "       [1.        , 0.13461538, 0.08108108, 0.73469388],\n",
     "       [0.        , 0.53846154, 0.09009009, 0.04081633],\n",
     "       [1.        , 0.25      , 0.09009009, 0.73469388],\n",
     "       [0.        , 0.69230769, 0.11711712, 0.13265306],\n",
     "       [1.        , 0.21153846, 0.11711712, 0.82653061],\n",
     "       [0.        , 0.51923077, 0.11711712, 0.31632653],\n",
     "       [1.        , 0.32692308, 0.11711712, 0.6122449 ],\n",
     "       [0.        , 0.42307692, 0.12612613, 0.30612245],\n",
     "       [0.        , 0.09615385, 0.12612613, 0.87755102],\n",
```

```
"      [1.        , 0.80769231, 0.13513514, 0.03061224],\n",
"      [0.        , 0.05769231, 0.13513514, 0.73469388],\n",
"      [1.        , 0.67307692, 0.16216216, 0.03061224],\n",
"      [1.        , 0.        , 0.16216216, 0.92857143],\n",
"      [0.        , 0.59615385, 0.16216216, 0.13265306],\n",
"      [0.        , 0.05769231, 0.16216216, 0.81632653],\n",
"      [0.        , 0.46153846, 0.17117117, 0.16326531],\n",
"      [0.        , 0.23076923, 0.17117117, 0.73469388],\n",
"      [0.        , 0.34615385, 0.1981982 , 0.25510204],\n",
"      [0.        , 0.03846154, 0.1981982 , 0.75510204],\n",
"      [0.        , 0.90384615, 0.20720721, 0.34693878],\n",
"      [1.        , 0.11538462, 0.20720721, 0.92857143],\n",
"      [1.        , 0.57692308, 0.21621622, 0.35714286],\n",
"      [0.        , 0.25      , 0.21621622, 0.6122449 ],\n",
"      [0.        , 0.59615385, 0.21621622, 0.2755102 ],\n",
"      [0.        , 0.11538462, 0.21621622, 0.65306122],\n",
"      [0.        , 0.61538462, 0.22522523, 0.55102041],\n",
"      [0.        , 0.17307692, 0.22522523, 0.46938776],\n",
"      [0.        , 0.21153846, 0.22522523, 0.41836735],\n",
"      [0.        , 0.25      , 0.22522523, 0.41836735],\n",
"      [0.        , 0.59615385, 0.24324324, 0.52040816],\n",
"      [1.        , 0.28846154, 0.24324324, 0.60204082],\n",
"      [0.        , 0.25      , 0.25225225, 0.54081633],\n",
"      [1.        , 0.78846154, 0.25225225, 0.60204082],\n",
"      [0.        , 0.61538462, 0.25225225, 0.44897959],\n",
"      [1.        , 0.55769231, 0.25225225, 0.40816327],\n",
"      [0.        , 0.63461538, 0.26126126, 0.5       ],\n",
"      [1.        , 0.98076923, 0.26126126, 0.45918367],\n",
"      [0.        , 0.17307692, 0.27927928, 0.51020408],\n",
"      [1.        , 0.67307692, 0.27927928, 0.45918367],\n",
"      [1.        , 1.        , 0.27927928, 0.56122449],\n",
"      [1.        , 0.01923077, 0.27927928, 0.55102041],\n",
"      [0.        , 0.94230769, 0.28828829, 0.52040816],\n",
"      [0.        , 0.69230769, 0.28828829, 0.59183673],\n",
"      [1.        , 0.86538462, 0.2972973 , 0.51020408],\n",
"      [1.        , 0.        , 0.2972973 , 0.59183673],\n",
"      [0.        , 0.48076923, 0.2972973 , 0.5       ],\n",
"      [0.        , 0.96153846, 0.2972973 , 0.47959184],\n",
"      [1.        , 0.01923077, 0.2972973 , 0.59183673],\n",
"      [0.        , 0.26923077, 0.2972973 , 0.46938776],\n",
"      [1.        , 1.        , 0.30630631, 0.55102041],\n",
"      [0.        , 0.55769231, 0.30630631, 0.41836735],\n",
"      [0.        , 0.80769231, 0.31531532, 0.48979592],\n",
"      [0.        , 0.80769231, 0.31531532, 0.56122449],\n",
"      [1.        , 0.78846154, 0.35135135, 0.46938776],\n",
"      [1.        , 0.15384615, 0.35135135, 0.54081633],\n",
"      [0.        , 0.51923077, 0.35135135, 0.53061224],\n",
"      [1.        , 0.42307692, 0.35135135, 0.47959184],\n",
"      [0.        , 0.09615385, 0.35135135, 0.52040816],\n",
"      [0.        , 0.59615385, 0.35135135, 0.41836735],\n",
"      [1.        , 0.75      , 0.35135135, 0.51020408],\n",
"      [1.        , 0.38461538, 0.35135135, 0.55102041],\n",
"      [1.        , 0.94230769, 0.35135135, 0.40816327],\n",
"      [0.        , 0.53846154, 0.35135135, 0.43877551],\n",
"      [0.        , 0.05769231, 0.35135135, 0.57142857],\n",
```

```
"       [1.        , 0.57692308, 0.35135135, 0.45918367],\n",
"       [0.        , 0.71153846, 0.37837838, 0.58163265],\n",
"       [0.        , 0.07692308, 0.37837838, 0.55102041],\n",
"       [0.        , 0.30769231, 0.38738739, 0.60204082],\n",
"       [0.        , 0.61538462, 0.38738739, 0.45918367],\n",
"       [0.        , 0.96153846, 0.3963964 , 0.55102041],\n",
"       [1.        , 0.        , 0.3963964 , 0.40816327],\n",
"       [1.        , 0.57692308, 0.40540541, 0.48979592],\n",
"       [0.        , 0.42307692, 0.40540541, 0.39795918],\n",
"       [0.        , 0.26923077, 0.40540541, 0.41836735],\n",
"       [1.        , 0.11538462, 0.40540541, 0.52040816],\n",
"       [0.        , 0.55769231, 0.40540541, 0.46938776],\n",
"       [0.        , 0.17307692, 0.40540541, 0.5       ],\n",
"       [1.        , 0.57692308, 0.41441441, 0.41836735],\n",
"       [1.        , 0.03846154, 0.41441441, 0.48979592],\n",
"       [0.        , 0.09615385, 0.42342342, 0.40816327],\n",
"       [0.        , 0.59615385, 0.42342342, 0.47959184],\n",
"       [1.        , 0.94230769, 0.42342342, 0.59183673],\n",
"       [1.        , 0.15384615, 0.42342342, 0.55102041],\n",
"       [1.        , 0.59615385, 0.42342342, 0.56122449],\n",
"       [0.        , 0.05769231, 0.42342342, 0.41836735],\n",
"       [0.        , 0.92307692, 0.43243243, 0.5       ],\n",
"       [1.        , 0.69230769, 0.43243243, 0.45918367],\n",
"       [1.        , 0.96153846, 0.43243243, 0.42857143],\n",
"       [1.        , 0.92307692, 0.43243243, 0.47959184],\n",
"       [1.        , 0.90384615, 0.43243243, 0.52040816],\n",
"       [0.        , 0.01923077, 0.43243243, 0.54081633],\n",
"       [0.        , 0.38461538, 0.44144144, 0.41836735],\n",
"       [1.        , 0.01923077, 0.44144144, 0.45918367],\n",
"       [0.        , 0.        , 0.45045045, 0.47959184],\n",
"       [0.        , 0.01923077, 0.45045045, 0.5       ],\n",
"       [0.        , 0.86538462, 0.45045045, 0.42857143],\n",
"       [0.        , 0.59615385, 0.45045045, 0.59183673],\n",
"       [0.        , 0.63461538, 0.46846847, 0.42857143],\n",
"       [0.        , 0.61538462, 0.46846847, 0.57142857],\n",
"       [1.        , 0.17307692, 0.46846847, 0.56122449],\n",
"       [0.        , 0.38461538, 0.46846847, 0.39795918],\n",
"       [0.        , 0.42307692, 0.48648649, 0.58163265],\n",
"       [1.        , 0.40384615, 0.48648649, 0.91836735],\n",
"       [0.        , 0.09615385, 0.4954955 , 0.28571429],\n",
"       [0.        , 0.25      , 0.4954955 , 0.7755102 ],\n",
"       [1.        , 0.48076923, 0.5045045 , 0.34693878],\n",
"       [1.        , 0.42307692, 0.5045045 , 0.95918367],\n",
"       [1.        , 0.78846154, 0.5045045 , 0.10204082],\n",
"       [1.        , 0.38461538, 0.5045045 , 0.75510204],\n",
"       [1.        , 0.55769231, 0.5045045 , 0.08163265],\n",
"       [1.        , 0.40384615, 0.5045045 , 0.75510204],\n",
"       [0.        , 0.13461538, 0.51351351, 0.33673469],\n",
"       [0.        , 0.25      , 0.51351351, 0.71428571],\n",
"       [1.        , 0.03846154, 0.52252252, 0.04081633],\n",
"       [0.        , 0.21153846, 0.52252252, 0.8877551 ],\n",
"       [0.        , 0.5       , 0.52252252, 0.06122449],\n",
"       [1.        , 0.26923077, 0.52252252, 0.73469388],\n",
"       [1.        , 0.01923077, 0.53153153, 0.09183673],\n",
"       [0.        , 0.32692308, 0.53153153, 0.7244898 ],\n",
```

```
"      [0.        , 0.75      , 0.54054054, 0.04081633],\n",
"      [1.        , 0.26923077, 0.54054054, 0.93877551],\n",
"      [0.        , 0.19230769, 0.54954955, 0.39795918],\n",
"      [0.        , 0.26923077, 0.54954955, 0.87755102],\n",
"      [1.        , 0.13461538, 0.55855856, 0.1122449 ],\n",
"      [1.        , 0.19230769, 0.55855856, 0.97959184],\n",
"      [1.        , 0.57692308, 0.55855856, 0.35714286],\n",
"      [0.        , 0.26923077, 0.55855856, 0.74489796],\n",
"      [0.        , 0.30769231, 0.56756757, 0.21428571],\n",
"      [1.        , 0.30769231, 0.56756757, 0.90816327],\n",
"      [1.        , 0.48076923, 0.56756757, 0.16326531],\n",
"      [1.        , 0.40384615, 0.56756757, 0.8877551 ],\n",
"      [0.        , 0.5       , 0.56756757, 0.19387755],\n",
"      [0.        , 0.38461538, 0.56756757, 0.76530612],\n",
"      [0.        , 0.55769231, 0.56756757, 0.15306122],\n",
"      [0.        , 0.17307692, 0.56756757, 0.89795918],\n",
"      [1.        , 0.36538462, 0.56756757, 0.        ],\n",
"      [0.        , 0.23076923, 0.56756757, 0.78571429],\n",
"      [1.        , 0.30769231, 0.56756757, 0.        ],\n",
"      [0.        , 0.23076923, 0.56756757, 0.73469388],\n",
"      [0.        , 0.73076923, 0.57657658, 0.34693878],\n",
"      [0.        , 0.21153846, 0.57657658, 0.83673469],\n",
"      [1.        , 0.01923077, 0.59459459, 0.04081633],\n",
"      [0.        , 0.25      , 0.59459459, 0.93877551],\n",
"      [1.        , 0.61538462, 0.63063063, 0.25510204],\n",
"      [0.        , 0.34615385, 0.63063063, 0.75510204],\n",
"      [1.        , 0.46153846, 0.63963964, 0.19387755],\n",
"      [0.        , 0.28846154, 0.63963964, 0.95918367],\n",
"      [0.        , 0.34615385, 0.64864865, 0.26530612],\n",
"      [1.        , 0.26923077, 0.64864865, 0.63265306],\n",
"      [1.        , 0.42307692, 0.64864865, 0.12244898],\n",
"      [1.        , 0.19230769, 0.64864865, 0.75510204],\n",
"      [1.        , 0.34615385, 0.64864865, 0.09183673],\n",
"      [1.        , 0.34615385, 0.64864865, 0.92857143],\n",
"      [0.        , 0.65384615, 0.65765766, 0.12244898],\n",
"      [0.        , 0.23076923, 0.65765766, 0.86734694],\n",
"      [1.        , 0.76923077, 0.65765766, 0.14285714],\n",
"      [1.        , 0.17307692, 0.65765766, 0.69387755],\n",
"      [1.        , 0.78846154, 0.7027027 , 0.13265306],\n",
"      [1.        , 0.32692308, 0.7027027 , 0.90816327],\n",
"      [0.        , 0.36538462, 0.73873874, 0.31632653],\n",
"      [0.        , 0.26923077, 0.73873874, 0.86734694],\n",
"      [1.        , 0.53846154, 0.74774775, 0.14285714],\n",
"      [0.        , 0.21153846, 0.74774775, 0.8877551 ],\n",
"      [0.        , 0.44230769, 0.75675676, 0.3877551 ],\n",
"      [1.        , 0.23076923, 0.75675676, 0.97959184],\n",
"      [0.        , 0.69230769, 0.77477477, 0.23469388],\n",
"      [1.        , 0.19230769, 0.77477477, 0.68367347],\n",
"      [0.        , 0.44230769, 0.79279279, 0.16326531],\n",
"      [0.        , 0.34615385, 0.79279279, 0.85714286],\n",
"      [0.        , 0.30769231, 0.79279279, 0.2244898 ],\n",
"      [0.        , 0.26923077, 0.79279279, 0.69387755],\n",
"      [1.        , 0.28846154, 0.88288288, 0.07142857],\n",
"      [0.        , 0.38461538, 0.88288288, 0.91836735],\n",
"      [0.        , 0.55769231, 0.94594595, 0.15306122],\n",
```

```
    "       [0.        , 0.32692308, 0.94594595, 0.79591837],\n",
    "       [0.        , 0.51923077, 1.        , 0.2755102 ],\n",
    "       [1.        , 0.26923077, 1.        , 0.74489796],\n",
    "       [1.        , 0.26923077, 0.40540541, 0.17346939],\n",
    "       [1.        , 0.23076923, 0.40540541, 0.83673469]])"
   ]
  },
  "execution_count": 26,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "df"
]
},
{
"cell_type": "markdown",
"id": "8395e4ec",
"metadata": {},
"source": [
 "# 9. Perform any of the clustering algorithms"
]
},
{
"cell_type": "markdown",
"id": "494a1b9c",
"metadata": {},
"source": [
 "#### Kmeans_clustering"
]
},
{
"cell_type": "code",
"execution_count": 27,
"id": "e53db14b",
"metadata": {},
"outputs": [],
"source": [
 "from sklearn.cluster import KMeans"
]
},
{
"cell_type": "code",
"execution_count": 28,
"id": "bcad85b9",
"metadata": {},
"outputs": [],
"source": [
 "TWSS=[]\n",
 "k=list(range(2,9))\n",
 "\n",
 "for i in k:\n",
 "    kmeans=KMeans(n_clusters=i,init='k-means++')\n",
 "    kmeans.fit(data)\n",
```

"   TWSS.append(kmeans.inertia_)"
]
},
{
"cell_type": "code",
"execution_count": 29,
"id": "596f5476",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"[381550.6840684068,\n",
" 268082.56760639744,\n",
" 191612.56821803437,\n",
" 153394.66603206735,\n",
" 119223.63779954854,\n",
" 101364.2432178932,\n",
" 85819.89345888031]"
]
},
"execution_count": 29,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"TWSS"
]
},
{
"cell_type": "code",
"execution_count": 30,
"id": "a91c6ab3",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"Text(0, 0.5, 'TWSS')"
]
},
"execution_count": 30,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAAZgAAAEGCAYAAABYV4NmAAAAOXRFWHRTb2Z
0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAA
CXBIWXMAAAsTAAALEwEAmpwYAAAoU0lEQVR4nO3de5yVZbn/8c93QBE8ICq6URA0KVNTzAkt7eARr
DylJkqKShszM612qdkvPESZGSaZJqmRgoeSrY7NdJMMxUGxQOKSSqCkrI3iiiKHK7fH/c9zYKGcZiZNc+sNd/
36/W81rPu57Cup73lmvvvw3LciAjMzs7ZWU3QAZmzgzMysLJxyzLJxgzZMysLJxgzMysLJxgzMysLLoWHUBHsc
UWW8SAAQOKDsPMrKLMmDHjzId2PHnGCyAQMGUFdXV3QYZmYVRdLctR1zE5mmZYVRdLctR1zE5mZmZWFE4yZmZW
FE4yZmZWFE4yZmZWFE4yZmZWFE0xrTZwIAwZATU36PXFi0RGZmXUIn
UWW8SAAQOKDsPMrKLMmDHjfyOid2PHnGCyAQMGUFdXV3QYZmYVRdLctR1zE5mZmZWFE4yZmZW
FE4yZmZWFE4yZmZWFE4yZmZWFE0xrTZwIAwZATU36nDix6IjMzDoED1NujYkTYdLo0fZ87N30HG

D68uLjMzDoA12Ba49xzG5JLvaVLU7mZWSfnBNMaL720buVmZp2IE0xrbLvtupWbmXUiTjCtMWYM9Oixeln
37qnczKyTc4JpjeHDYfx46N8fpLQNGADHHlt0ZGZmhXOCaa3hw+HFF2HVqpRs/vlPeO65oqMyMyucE0xbGjk
Snn0WPvShoiMxMyucE0xbkqB371SbmTw5fZqZdVJOMOVwxx1wzDFwxRVFR2JmVhgnmHI45BA4+GA46yz4
xz+KjsbMrBBOMOUgpQ7/9daDk092U5mZdUpOMOXSty9ceincfz9cfnnR0ZiZtTtPdllOJ54I06bBrrsWHYmZWbt
zgiknCa68sugozMwK4Say9rBiBXznOzBuXNGRmJm1GyeY9tClCzz9NJx9tt/yN7NOwwmmPdSPKuvWDU46CV
auLDoiM7Oyc4JpL1tvDZddBg8+6KYyM+sUnGDa0/HHp5cwL7gAliwpOhozs7IqW4KRtIGkaZIelzRL0vm5/DxJ
L0uambfPllxzjqQ5kp6VNKSkfA9JT+Zj4yQpl3eTNCmXPyJpQMk1IyQ9l7cR5XrOdSLBVVelWszGGxcdjZlZWZ
WzBrMM2C8idgMGAUMl7ZWPXRoRg/L2BwBJOwHDgJ2BocAVkrrk868ERgED8zY0l48EXo+IHYBLgR/ne20
GjAb2BAYDoyX1KuOzNl+fPrDTTmn/+eeLjcXMrIzKlmAieSt/XS9v0cQlhwE3RcSyiHgBmAMMltQH2CQiHoqI
AK4HDi+5ZkLevwXYP9duhgBTI2JRRLwOTKUhKXUMl10GO++cpvc3M6tCZe2DkdRF0kzgNdI/+T9IS
ka0tqFtsA80oun5/Ltsn7a5avdk1ErAAWA5s3ca814xslqU5S3cKFC1v+oC1xzDFpueUTT/SoMjOrSmVNMBGxMiI
GAX1JtZFdSM1dHyA1my0AfppPV2O3aKK8pdeUxjc+ImojorZ3795NPEkZ/Md/wM9/Dg8/DGPHtu9vm5m1g3Y
ZRRYRbwD3AUMj4tWceFYBvyL1kUCqZfQruawv8Eou79tI+WrXSOoK9AQWNXGvjuXYY+GII+D//T945pmio
zEza1PlHEXWW9Kmeb87cAAwO/ep1DsCeCrvTwGG5ZFh25E686dFxAJgiaS9cv/KCcDtJdfUjxA7Crg399PcDR
wkqVdugjsol3Us9XOVbbstzJ1bdDRmZm2qnJNd9gEm5JFgNcDkiLhD0g2SBpGarFF4ETgGIiFmSJgNPAyuA0yKi
vnPiVOA6oDtwZ94ArgFukDSHVHMZlu+1SNZq9pyW22Vai9durz/uWZmFUTpD36rra2Nurq6
4gJYtSqtG3PAAQ3DmM3MOjhJMyKitrFjfpO/o1i0KL3hf+KJafZlM7MK5wTTUWyxBfziFzB9OlxySdHRmJm1
mhNMR/LFL8KRR8Lo0TBrVtHRmJm1ihNMRyLBFVfAJpvAySeD+8fMrIJ5yeSOZsst4brrYMMNU8IxM6tQTj
Ad0ec+17C/fDmst15xsZiZtZCbyDqyH/0I9tknJRkzswrjBNORffCDMG0aXHxx0ZGYma0zJ5iO7Mgj06zL558PTz
xRdDRmZuvECaaju/xy6NUrvYDppjIzqyBOMB3dFlukCTFnzUovYZqZVQiPIqsEX/gC/OMf0Lfv+59rZBuAZT
KeqTy113uanMzCqCE0wlmTYNDj44DV82M+vgnGAqyeDBcNxxcOGFMHNm0dGYmTXJCabSjBsHm2j2+eRpW
9917R0ZiZrZUTTKXZfHO46ip4/HH44Q+LjsbMbK2cYCrRYYfB6afDjjsWHYmZ2Vp5mHKlGjeu6AjMzJrkGkw
li0iJxk11lZtYBOcFUMglmzIDvfz99mpl1IE4wle5nP4P4Ottkqjypy
pYtKzoaM7N/cYKpdL16wfjx8OST8MMfFh2NmdkzQmq8y76wfjx8NRT6f0YM7MOwgm
mGnzuc6kGc/HFMH9+0dGYmQFOMNXj0kth6lRPiGlmHUbZEoykjST9SqSbXgoLz2mKTZSnfFVxOXJgVmbSm1JkcyxxRhxnnpRkwxK
2R9KykISXle0h6Mh8bJ0m5vJukSbn8EUkDSq4ZIem5vI2gwzUDx5ldDkHuqsJ52Vu8Zn5lPJlCAKhKMC0Ji+aqkYPQ6hoUYpu7/iqFI+zoHSGQT2jX1nm5HlcRwE8w7yYtYk5SXjJRTYh3dOvGEIGu1JXlU4AwAbNHRk0jXALLGOeuDA0pw0rqkXG0GOo+1zs3PCf9TEQWlrDbSNuD+mf1UXqhu1M+LpTm5dhqY1mcCLtHIwDDjuiz2TcWUMuYzAjQ1QUZy4o5kTlVG1/GOmUWvJ1Ch3h5V9Igkr
2R9KykISXle0h6Mh8bJ0m5vJukSbn8EUkDSq4ZkX/jOUkjyvWcHcamm8KnP532X3650FDMzKC8NZhlwH4RsR
swCBgqaS/gbOCeiBgI3JO/I2knYBiwMzAUuEJSl3yvK4FRwMC8Dc3lI4HXI2IH4FLgx/lemwGjgT2BwcDo0kRW
1W6+Gbbf3mvHmFnhypZgInkrf10vbwEcBkzI5ROAw/P+P+YcBNEbEsIl4A5gCDJfUBNomIhyIigOvXuKb+XrcA+
+fazRBgakQsiojXgak0JKXqduCB0Lt36pN5992iozGzTqysfTCSukicSukiaCbxG+gf/EWCriFgAkD+3zKdvA8wruXx+Lt
sm769Zvto1EbECWAxs3sS91oxvlKQ6SXULFy5sxZN2IJtuCr/6FTz9NJx/ftHRmFknVtYEExErI2IQ0JdUG9mlidP
V2C2aKG/pNaXxjY+I2oio7d27dxOhVZiDD4aTT06jyqZNKzoaM+uk2mUUWUS8AdxHaqZ6NTd7kT9fy6fNB/qVXNYXeCWX922kfLVrJHUFegKLmrhX53HJJfDBD6ZRZe/5bc0s+21oNpB9a2znfh0GXnDrDxkAcnVu7zpGb/xSk/TDqR3sqyNmKjRZTh/Ruak2VdP9dwmOThZAOOyzZqZS6PZbpgskR4pZGh/C1wqIgU4Gxq+tRAXPc6/MLUSCCGGiAnmWQQJzHh3ZVG/YJ6nQ0lVr2pKWa5/LYm7YlJBU7LpnX1MOa5MUyH4GZeCUCjbjRXNm4Dzuubff0kZc/XmXS4ZdP/ImsNMm3fLKBHSLqPpPA4/xqDqmZSR6Jq/xnzQS3UyQrzl7S40TEBULgqX6ddY5qKmoeRaP6Q9Rx2UkBSlq/WXJGn0E7hq8jwoVMxb/Ctr+7NL80mcDeFOBuVy1/0UMDxc3gdxYh1bljGPQZy8tgFXbD2aTPugNvmEGtWygPA6Dv3P3hiQ+Dm5HngbL4N0SOxZsyKqE8r/Wa5tDZ3n9jJJFPt7xY+ZGDU1dV2jwWZFVeE3NJSLMzs/8YKNsaDsVa2ylyz7XA3YrZyg7HMSf+8EB8Fz26NazSHJdKENxbBmZwTBZkaeCpxpngrw8uhGNPwu7fxqkyR2Sf0qPvr3xvTfbxp6/RNB3Bx0tAGkd3dNWqlgpzKIfXH4ZT20ZVU8MCn/GvhKJ1CSg4bE9EcLY0+QCDyXr1S9thXqmKM/vStYLzUk
GAX1JtZFdSM1dHyA1my0AfppPV2O3aKK8pdeUxjc+ImojorZ3795NPEkZ/Md/wM9/Dg8/DGPHtu9vm5m1g3Y

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/lz14l15wjaY6kZyUNKSnfQ9KT+di4PAknkrpJmpTLH5E0oOSaEfk3npM0olzPae1IakguY8emGZnd8W/
WYZWzBrMC+FZEfBjYCzhN0k7A2cA9ETEQuCd/Jx8bPq9eShwhaT6WQ+vBEYBA/M2NJePBF6PiB2AS4Ef
53ttRpqMc0/yxJylicwqXERqJrvkkjTdv5OMWYdUtgQTEQsi4tG8vwR4BtgGOAyYkE+bBye9w8DboqIZRHxAj
AHGCypD7BJRDyUZ3e+fo1r6u91C7B/rt0MAaZGxKK8QNpUGpKSVToJxo2DU09NAwC++10nGbMOqMnp+tt
KbrranbQy5lb1k2dGxAJJW+bTtgEeLrlsfi5bnvfXLK+/Zl6+1wpJi4HNS8sbucaqgQSXX54Sy0UXpSn+f/CDoqMys
xJlTzCSNgJuBc6MiDe19uGljR2IJspbek1lpbKNITW9su+22a4vLOqqaGvjFL1KS2XrroqMxszWUdRSZpPVIyWVi
RNyWi1/NzV7kz9dy+XygX8nlfYFXcnnfRspXu0ZSV6AnsKiJe60mIsZHRG1E1Pbu3bulfqqmBK6+Er341fZ87
t9h4zOxfyjmKTMA1wDNrzLw8Bagf1TUCuL2kfFgeGbYdqTN/Wm5OWyJpr3zPE9a4pv5eRwH35n6au4GDJPX
KnfsH5TKrRvW14qefhp12giOPhAEDUvIZMAAmTiwyOrNOq5xNZHsDxwNPSpqZy74PXAhMljQSmAscBRARsyRNBm
kmaDDxNGoF2WkSszNedClwHdAfuzBukBHpDmDmkmsuwfK9Fki4EpufzLvDMBJ3AjjvCHnvAbbc1lM2dC6NG
pf3hw4uJy6yTUnj0DQC1tbVRV1dXdBjWWv37w0svNV7+4ovtHo5ZtZM0IyJqGzvmN/mtusyb13h5Y0nHzMrKC
caqy9pGA26+OfzsZ/DOO+0ajlln5gRj1WXMGOjRY/WyHj1S/8w3vgEf+EB6SfPdd4uJz6wTcYKx6xJ8OIwfn/pcp
PQ5fjw88ADcdx988INwxhkp0dx8c9HRmlU1JxirPsOHpw79Vav5SZ/3osU9/OiWZe+9NCaZbt1T+9tuwbFlBwZpV
LycY63z23Rf+8hc45JD0/aKLYOBA+OUv4b33io3NrIo4wVjnJDW8oLnfftC3b3b5o8c+DA1KTmRGPWak4wZvvuC
w8+CHfdBX36wCmnpGRjZq3SLrMpm3V4EgwZAgcdBHffDvkybfnzEnNaSecAOutV2yMZhXGNRizUhIMHQ
of+Uj6/pvfwJe/nIY5//rXsGJFsfGZVRAnGLOmjB4N//3f0KsXnHxySjS//W3RUZlVBCcYs6ZI8PnPw/TpcPvtsMk
mqb+m3qpVxcVm1sE5wZg1hwSHHgozZsBPPfpLKHngAPvzhtBzAypVNX2/WCTnBmK0LafWpaDbYAL70JdhlF
7jxRicasxJOMGYt9clPwmOPPpSlnunaF445LQ569BIYZ4ARj1jo1NXDUUfD44zBpUhoIKWazJQp7qOxTs0Jxqwt
1NTAF78IJ56Yvk+ZAocDrvtBrfc4kRjnZITjFk5HHpoGs68Gs68fDkcfTTSsvntaytmJxjoRJxizcujSBY
HOOE4x1Kk4wZuXUpUtaLmDWrDQFTdeu8NZbaUqa22/3gACrak4wZu2ha1cYMCDtywMuUfD44zBpUhoIIKWazJQp7qOxTs0Jxqwt
caKwKOcGYtbdddoHZs9PcZm+8kfprBg+GxYuLjsysBWha9c04mz2bLjmGth9955bRWha9c04mz2bLjmGth5Z+jZMx2bPds1GqsKTjBm
RVpvvfTuzHXXpe8vvwyDBsEnPpH6CZOTE1rNTXpc+LE4mI1W0deD8asI+ndG8aNgzFj0rIBNTUNI8/mzoVR
o9L+8OHFxWjWTGWrwUi6VtJfJrkp4qKTtP0suSZXubyH1E3SpFz+iKQBJd
eMkPRc3kaU6xnNN2tz666ck8ve/w2ab/fuw5qVL4dxzi4nBdB2Vs4nsOmBoI+WXRsSgvP0BQNJOwDBg53zNFZK
65POvBEYBA/NWf8+RwOsRsQNwKfDjfK/NgNHAnsHNHAnsBgYLSkXm3/eGZl1K0bvP5648fmzoUlS9o3HrMWKFuCi
Yj7gUXNPP0w4KaIWBYRLwBzgMGS+gCbRMRMRDERHA9cDhJddMyPu3APvn2s0QYGpELIqI14GzDq2b
bdd+7F+/eCss1KfjVkHVUQn/9ckPZGb0NRnrAOzOprFtsA80rOmZ/Ltsn7a5avErAAWA5s3s9ca/I2mUpdU
5m1tTFjVl8aANL3885LL2pecknq+P/97wsxbzf0P+EXFwiKanc5euRLsADAIWAD8NJerkXk
EbZZAYYPh/HjoX//NEz//7p++jRMHkyzJkDp58OEyJ+7MndmjkDwD7EfFqRKyMiFXhl
1KTu0LvJLL+zZSvto1kroCPUlNcmu7l1nlGT48vfm/alX6LB09tt12MHYsbLFF+j52LBx8MOy6a3qJc9myIi2I2+5d2
TTC5T6XeEUD9CLMpwLA8Mmw7Umf+tIhYACyRtFfuXkuBXzkbBuL3kmvoRYkcB9+Z+mruBgyT1yk1wB+Uys+p28
80wYUIa2nzyyan57Oqri47KOrGyvQcj6UbgaBuFnSvPXsqaU
Rad+DOvAFcA9wgaQ6p5jIs32uRpAvVCSfA8cfDPfekPpr6JZyXLoVX0oRedig2Rut
UFG6vBaC2tjbq6uqKDsOsbUWk/psrroCvfS1NsPmtb6WZAtRYd6XZupE0IyJqGzvmqWLMqll9EjniiLQ5ezX33pU
EBH/94alLzH5hWRk4wZp1Bnz5p2PO8eXD55bWpqK55ZZUtnBh
GgJ9zjmpn8asjTjBmHVGXbrA1lun/bff9Tk1mF1+cRp6deCI88USR0VmVcIIx6+wGDEi1meeeg698Je0PGgTz57/fl
WZNcoIxs2T77dNWm9dNSAfPmwaRJ0De/43zWWWm9Gr+4aevICcbMVterFxx9dNpfPgoGg8IWRa/t
GCMbO169YNHn00JZldd01r0fTrB//zP0VHZhXACCbMmibZBgQemJZwffyZ84/ZQ83t1Dz8MDz1UbHzWYTnB
mFnz7borXHstbLvhWlWgE98Am69tWFqGjOcYMysNSZgza7mNNk
pznNW/uLnVVvB//5eOvfsuLFhQbHxWKCcYM2u9Ll3gyCPhgpJPgySdh4sQ0Eq2mJn1On
FhkxNYOyjZdv5l1UjX579b994dRo9LiZ9ddl8pXrUrH5s5Nx2D1RdSsqrgGY2blsf32aWIbkku9pUvhm9
9MU9VYVXKCMbPy2mwzWLy48WOvQXe9ePht69IArr4QHHkjl//u
/adnnX/86zZHmdWsqkvtgzKz8xoxJfS8gegbY8c29MG8917aAP7xD7j99pDgg9kGWXdg3dWs

41GDMrv+HDYfz4NKpMSp/jx6/ewb/++mnYM8Cee6Z1ambNgquuSjMJzJiRRqsB3HgjHHAAnH8+3HMPvPVW
+z+TvS+Fq54A1NbWRl1dXdFhmFlz/Pa38JOfpKlrIlLi+ehH4S9/ge7dYflyWG+9oqPsFCTNiIjaxo65iczMKs9xx6
Vt8eI0MOCvf4Xnn0/JBdJ8abNmpWa1ffaBT34SPvCBhiWirV04wZhZ5erZE4YOTVupIUPSvGi//32aOw3SOXfe
mfaffTYlnK7+J7Cc/L+umVWfU05J26pVMHt2quHU9++8915asbNLl7RUdH0tZ889G86xNuEEY2bVq6YGdtopbf
Ui0swCDzyQEs/556eyCy+E730P3nwT/vSnlHS23LKw0KtB2UaRSbpW0muSniop20zSVEnP5c9eJcfOkTRH0rOS
hpSU7yHpyXxsnJQaUSV1kzQplz8iaUDJNSPybzwnaUS5ntHMKlC3bqmP5vLLYeZMeP311HR27LHp+F//muZV
22or+NCHYOTIlJDqV/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
ARMySNBl4GlgBnBYRK/OtTgWuA7oDd+YN4BrgBklzSDWXYfleiyRdCEzP510QEWsONjAza5nhw1s+Yqym
BvbYI231li5tmEX6lVdgzhy4446GJrT+/dOEn0OGpMXZli9PC7hVAE92mXmYspl1GG++CY89BtOnQ11dGr32k
Y+kWaSPOy4NGqitbRhEMHhwGv1WgKaGKTvBZE4wZtbhzZ4Nv/tdQ+KZlwfMvvQS9OsHf/wj/P3vKensthtssE
HZQyrkPRgzM2tjO+4I55wDt92Wkso//wl/+AP0zWOhbr0VTj8d9toLNt44LWFw2mkNzW1rVijKPCuBazCZazBm
VvEiUq2mri5t06enPpv77kvHDzkkLUddW5v6fa6/Ht59t+H6FsxK4CayZnCCMbOqd/75cO+9aSqcta0C2r8/vPhis2/
pBNMMTjBm1mmsXJlW/Gzs338pLXPQTO6DMTOzBl26NG9WglZygjEz64zWNivBmDFt9hNOMGZmndG6zE
rQQl7R0syss2rNrATN4BqMmZmVhROMmZmVhROMmZmVhROMmZmVhd/kzyQtBOa24hZbA
P/bRuEUqVqeA/wsHVW1PEu1PAe07ln6R0Sja847wbQRSXVrmy6hklTLc4CfpaOqlmeplueA8j2Lm8jMzKwsnG
DMzKwsnGDazviiA2gj1fIc4GfpqKrlWarlOaBMz+I+GDMzKwvXYMzMrCycYMzMrCycYFpBUj9Jf5b0jKRZks
4oOqaWkrSBpGmSHs/Pcn7RMbWGpC6SHpN0R9GxtIakFyU9KWmmpIpeclXSppJukTQ7/zfz8aJjaglJH8r/96jf3p
R0ZtFxtYSkb+T/3p+SdKOkDdr0/u6DaTlJfYA+EfGopI2BGcDhEfF0waGtM0kCNoyItyStB/wVOCMiHi44tBaR9E
2gFtgkIj5fdDwtJelFoDYiKv6FPkkTgAci4mpJ6wM9IuKNgsNqFUldgJeBPSOiNS9qtztJ25D+O98pIt6RNBn4Q0Rc
11a/4RpMK0TEgoh4NO8vAZ4Btik2qpaJ5K38db28VeRfH5L6Ap8Dri46FkskbQJ8CrgGICLCLeq/Tkku0P/KPSkkuJr
kB3SV2BHsArbXlzJ5g2ImkAsDvwSMGhtFhuVpoJvAZMjYhKfZafAd8BVhUcR1sI4I+SZkgaVXQwrbA9sBD4d
W66vFrShkUH1QaGATcWHURLRMTLwCXAS8ACYHFE/LEtf8MJpg1I2gi4FTgzIt4sOp6WioiVETEI6AsMlrR
LwSGtM0mfB16LiBlFx9JG9o6IjwIHA6dJ+lTRAbVQV+CjwJURsTvwNnB2sSG1Tm7mOxS4uehYWkJSL+AwY
Dtga2BDSV9qy99wgmml3F9xKzAxIm4rOp62kJsu7gOGFhtJi+wNHJr7Lm4C9pP0m2JDarmIeCV/vgb8DhhcbEQt
Nh+YX1IrvoWUcCrZwcCjEfFq0YG00AHACxGxMCKWA7cnCn2jLH3CCaYXcMX4N8ExEjC06ntaQ1FvSpnm/
O+n/+WYXGlQLRMQ5EdE3IgaQmi/ujYg2/ausvUjaMA8eITcnHQQ8VWxULRMR/wTmSfpMBjGMs4lgp
tHsteAvaS1CP/W7Y/qR+5zXRty5t1QnsDxwNP5r4LgO9GxB+KC6nF+gAT8qiYGmByRFT0EN8qsBXwu/TfPl2B
30bEXcWG1CqnAxNz09LzwEkFx9NiknoABwKnFB1LS0XEI5JuAR4VFvTig4jyPKg6VeJE97L+mjpLU9
kLQ1sDQifkNaVKp+avwlQH0/zsPA3pJ2yNf0kPTB94nJrNlcgzFrGyOAX+YhwGubKfjrwLWSvk1a3bHJ2YQj4j1
JxwA/z53/75D6YUrdCpyQm7imA3/P5R8BfiJpFbBfulLQg98OcCNwoqVs+/r2Se5i1imdTNjOzsnATm
ZmZlYUTjJmZlYUTjJmZlYUTjJmZlcX/B5VDraYFskFEAAAAAElFTkSuQmCC
",
    "text/plain": [
     "<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": [
    "array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,\n",
    "       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,\n",
    "       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0,\n",
    "       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
    "       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
    "       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, 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,\n",
    "       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,\n",
    "       2, 3])"
   ]
```

```
  },
  "execution_count": 32,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "model.labels_"
]
},
{
"cell_type": "code",
"execution_count": 33,
"id": "053dd73a",
"metadata": {},
"outputs": [],
"source": [
 "mb=pd.Series(model.labels_)"
]
},
{
"cell_type": "code",
"execution_count": 34,
"id": "ee476632",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/html": [
    "<div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
    "      <th></th>\n",
    "      <th>CustomerID</th>\n",
    "      <th>Gender</th>\n",
    "      <th>Age</th>\n",
    "      <th>Annual Income (k$)</th>\n",
    "      <th>Spending Score (1-100)</th>\n",
    "    </tr>\n",
    "  </thead>\n",
    "  <tbody>\n",
```

```
    "    <tr>\n",
    "      <th>0</th>\n",
    "      <td>1</td>\n",
    "      <td>1</td>\n",
    "      <td>19</td>\n",
    "      <td>15</td>\n",
    "      <td>39</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>1</th>\n",
    "      <td>2</td>\n",
    "      <td>1</td>\n",
    "      <td>21</td>\n",
    "      <td>15</td>\n",
    "      <td>81</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>2</th>\n",
    "      <td>3</td>\n",
    "      <td>0</td>\n",
    "      <td>20</td>\n",
    "      <td>16</td>\n",
    "      <td>6</td>\n",
    "    </tr>\n",
    "  </tbody>\n",
    "</table>\n",
    "</div>"
   ],
   "text/plain": [
    "   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)\n",
    "0           1       1   19                  15                      39\n",
    "1           2       1   21                  15                      81\n",
    "2           3       0   20                  16                       6"
   ]
  },
  "execution_count": 34,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "data.head(3)"
]
},
{
"cell_type": "markdown",
"id": "49ccef30",
"metadata": {},
"source": [
 "# 10. Add the cluster data with the primary dataset"
]
},
{
"cell_type": "code",
"execution_count": 35,
```

```
 "id": "80715661",
 "metadata": {},
 "outputs": [],
 "source": [
  "data['clust']=mb"
 ]
},
{
 "cell_type": "code",
 "execution_count": 36,
 "id": "0bc5b4b9",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>CustomerID</th>\n",
     "      <th>Gender</th>\n",
     "      <th>Age</th>\n",
     "      <th>Annual Income (k$)</th>\n",
     "      <th>Spending Score (1-100)</th>\n",
     "      <th>clust</th>\n",
     "    </tr>\n",
     "  </thead>\n",
     "  <tbody>\n",
     "    <tr>\n",
     "      <th>0</th>\n",
     "      <td>1</td>\n",
     "      <td>1</td>\n",
     "      <td>19</td>\n",
     "      <td>15</td>\n",
     "      <td>39</td>\n",
     "      <td>1</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>1</th>\n",
     "      <td>2</td>\n",
```

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

```
{
 "cell_type": "code",
 "execution_count": 37,
 "id": "c919ea1d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>CustomerID</th>\n",
     "      <th>Gender</th>\n",
     "      <th>Age</th>\n",
     "      <th>Annual Income (k$)</th>\n",
     "      <th>Spending Score (1-100)</th>\n",
     "      <th>clust</th>\n",
     "    </tr>\n",
     "  </thead>\n",
     "  <tbody>\n",
     "    <tr>\n",
     "      <th>195</th>\n",
     "      <td>196</td>\n",
     "      <td>0</td>\n",
     "      <td>35</td>\n",
     "      <td>120</td>\n",
     "      <td>79</td>\n",
     "      <td>3</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>196</th>\n",
     "      <td>197</td>\n",
     "      <td>0</td>\n",
     "      <td>45</td>\n",
     "      <td>126</td>\n",
     "      <td>28</td>\n",
     "      <td>2</td>\n",
     "    </tr>\n",
     "    <tr>\n",
```

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

```
{
 "cell_type": "markdown",
 "id": "bcf97bab",
 "metadata": {},
 "source": [
  "# 11. Split the data into dependent and independent variables"
 ]
},
{
 "cell_type": "code",
 "execution_count": 38,
 "id": "5ffaa6fa",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "0      1\n",
     "1      1\n",
     "2      1\n",
     "3      1\n",
     "4      1\n",
     "      ..\n",
     "195    3\n",
     "196    2\n",
     "197    3\n",
     "198    2\n",
     "199    3\n",
     "Name: clust, Length: 200, dtype: int32"
    ]
   },
   "execution_count": 38,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "#dependent\n",
  "y= data['clust']\n",
  "y"
 ]
},
{
 "cell_type": "code",
 "execution_count": 39,
 "id": "3b4a49f2",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
```

```
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Annual Income (k$)</th>\n",
"      <th>Spending Score (1-100)</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1</td>\n",
"      <td>19</td>\n",
"      <td>15</td>\n",
"      <td>39</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>1</td>\n",
"      <td>21</td>\n",
"      <td>15</td>\n",
"      <td>81</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>0</td>\n",
"      <td>20</td>\n",
"      <td>16</td>\n",
"      <td>6</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>0</td>\n",
"      <td>23</td>\n",
"      <td>16</td>\n",
"      <td>77</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>0</td>\n",
"      <td>31</td>\n",
"      <td>17</td>\n",
"      <td>40</td>\n",
```

```
    "    </tr>\n",
    "  </tbody>\n",
    "</table>\n",
    "</div>"
   ],
   "text/plain": [
    "   Gender  Age  Annual Income (k$)  Spending Score (1-100)\n",
    "0       1   19                  15                      39\n",
    "1       1   21                  15                      81\n",
    "2       0   20                  16                       6\n",
    "3       0   23                  16                      77\n",
    "4       0   31                  17                      40"
   ]
  },
  "execution_count": 39,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "#independent\n",
 "x= data.drop(columns=['CustomerID','clust'],axis=1)\n",
 "x.head()"
]
},
{
"cell_type": "code",
"execution_count": 52,
"id": "8925a5cb",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/html": [
    "<div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
    "      <th></th>\n",
    "      <th>Gender</th>\n",
    "      <th>Age</th>\n",
    "      <th>Annual Income (k$)</th>\n",
```

```
 "      <th>Spending Score (1-100)</th>\n",
 "    </tr>\n",
 "  </thead>\n",
 "  <tbody>\n",
 "    <tr>\n",
 "      <th>195</th>\n",
 "      <td>0</td>\n",
 "      <td>35</td>\n",
 "      <td>120</td>\n",
 "      <td>79</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>196</th>\n",
 "      <td>0</td>\n",
 "      <td>45</td>\n",
 "      <td>126</td>\n",
 "      <td>28</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>197</th>\n",
 "      <td>1</td>\n",
 "      <td>32</td>\n",
 "      <td>126</td>\n",
 "      <td>74</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>198</th>\n",
 "      <td>1</td>\n",
 "      <td>32</td>\n",
 "      <td>60</td>\n",
 "      <td>18</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>199</th>\n",
 "      <td>1</td>\n",
 "      <td>30</td>\n",
 "      <td>60</td>\n",
 "      <td>83</td>\n",
 "    </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "     Gender  Age  Annual Income (k$)  Spending Score (1-100)\n",
 "195       0   35                 120                      79\n",
 "196       0   45                 126                      28\n",
 "197       1   32                 126                      74\n",
 "198       1   32                  60                      18\n",
 "199       1   30                  60                      83"
]
},
"execution_count": 52,
"metadata": {},
"output_type": "execute_result"
```

```
 }
],
"source": [
 "x.tail()"
]
},
{
"cell_type": "markdown",
"id": "55027329",
"metadata": {},
"source": [
 "# 12. Split the data into training and testing"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "8c768efa",
"metadata": {},
"outputs": [],
"source": [
 "from sklearn.model_selection import train_test_split"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "65a64d15",
"metadata": {},
"outputs": [],
"source": [
 "x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)"
]
},
{
"cell_type": "markdown",
"id": "1f083fee",
"metadata": {},
"source": [
 "# 13. Build the Model"
]
},
{
"cell_type": "code",
"execution_count": 41,
"id": "2e2f691a",
"metadata": {},
"outputs": [],
"source": [
 "from sklearn.ensemble import RandomForestClassifier"
]
},
{
"cell_type": "code",
"execution_count": 42,
```

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

```
{
 "cell_type": "markdown",
 "id": "0faa2234",
 "metadata": {},
 "source": [
  "# 16. Measure the performance using Evaluvation 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
}
```