{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "markdown",

"source": [

"\*\*Assignment 4\*\*\n",

"\n",

"\*\*Name : PRINCY J\*\* \n",

"\*\*Roll No : 19CS22\*\*\n",

"\n"

],

"metadata": {

"id": "hlLw1w9K3NI6"

}

},

{

"cell\_type": "markdown",

"source": [

"\*\*1.Loading Dataset into tool\*\*"

],

"metadata": {

"id": "BF1Z3OTs3i6k"

}

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 74

},

"id": "2WgoXmvj3LpD",

"outputId": "496710eb-e12f-4a4b-b528-069ab71e8b8e"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<IPython.core.display.HTML object>"

],

"text/html": [

"\n",

" <input type=\"file\" id=\"files-94808c94-f490-43e3-a583-6454170482d8\" name=\"files[]\" multiple disabled\n",

" style=\"border:none\" />\n",

" <output id=\"result-94808c94-f490-43e3-a583-6454170482d8\">\n",

" Upload widget is only available when the cell has been executed in the\n",

" current browser session. Please rerun this cell to enable.\n",

" </output>\n",

" <script>// Copyright 2017 Google LLC\n",

"//\n",

"// Licensed under the Apache License, Version 2.0 (the \"License\");\n",

"// you may not use this file except in compliance with the License.\n",

"// You may obtain a copy of the License at\n",

"//\n",

"// http://www.apache.org/licenses/LICENSE-2.0\n",

"//\n",

"// Unless required by applicable law or agreed to in writing, software\n",

"// distributed under the License is distributed on an \"AS IS\" BASIS,\n",

"// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n",

"// See the License for the specific language governing permissions and\n",

"// limitations under the License.\n",

"\n",

"/\*\*\n",

" \* @fileoverview Helpers for google.colab Python module.\n",

" \*/\n",

"(function(scope) {\n",

"function span(text, styleAttributes = {}) {\n",

" const element = document.createElement('span');\n",

" element.textContent = text;\n",

" for (const key of Object.keys(styleAttributes)) {\n",

" element.style[key] = styleAttributes[key];\n",

" }\n",

" return element;\n",

"}\n",

"\n",

"// Max number of bytes which will be uploaded at a time.\n",

"const MAX\_PAYLOAD\_SIZE = 100 \* 1024;\n",

"\n",

"function \_uploadFiles(inputId, outputId) {\n",

" const steps = uploadFilesStep(inputId, outputId);\n",

" const outputElement = document.getElementById(outputId);\n",

" // Cache steps on the outputElement to make it available for the next call\n",

" // to uploadFilesContinue from Python.\n",

" outputElement.steps = steps;\n",

"\n",

" return \_uploadFilesContinue(outputId);\n",

"}\n",

"\n",

"// This is roughly an async generator (not supported in the browser yet),\n",

"// where there are multiple asynchronous steps and the Python side is going\n",

"// to poll for completion of each step.\n",

"// This uses a Promise to block the python side on completion of each step,\n",

"// then passes the result of the previous step as the input to the next step.\n",

"function \_uploadFilesContinue(outputId) {\n",

" const outputElement = document.getElementById(outputId);\n",

" const steps = outputElement.steps;\n",

"\n",

" const next = steps.next(outputElement.lastPromiseValue);\n",

" return Promise.resolve(next.value.promise).then((value) => {\n",

" // Cache the last promise value to make it available to the next\n",

" // step of the generator.\n",

" outputElement.lastPromiseValue = value;\n",

" return next.value.response;\n",

" });\n",

"}\n",

"\n",

"/\*\*\n",

" \* Generator function which is called between each async step of the upload\n",

" \* process.\n",

" \* @param {string} inputId Element ID of the input file picker element.\n",

" \* @param {string} outputId Element ID of the output display.\n",

" \* @return {!Iterable<!Object>} Iterable of next steps.\n",

" \*/\n",

"function\* uploadFilesStep(inputId, outputId) {\n",

" const inputElement = document.getElementById(inputId);\n",

" inputElement.disabled = false;\n",

"\n",

" const outputElement = document.getElementById(outputId);\n",

" outputElement.innerHTML = '';\n",

"\n",

" const pickedPromise = new Promise((resolve) => {\n",

" inputElement.addEventListener('change', (e) => {\n",

" resolve(e.target.files);\n",

" });\n",

" });\n",

"\n",

" const cancel = document.createElement('button');\n",

" inputElement.parentElement.appendChild(cancel);\n",

" cancel.textContent = 'Cancel upload';\n",

" const cancelPromise = new Promise((resolve) => {\n",

" cancel.onclick = () => {\n",

" resolve(null);\n",

" };\n",

" });\n",

"\n",

" // Wait for the user to pick the files.\n",

" const files = yield {\n",

" promise: Promise.race([pickedPromise, cancelPromise]),\n",

" response: {\n",

" action: 'starting',\n",

" }\n",

" };\n",

"\n",

" cancel.remove();\n",

"\n",

" // Disable the input element since further picks are not allowed.\n",

" inputElement.disabled = true;\n",

"\n",

" if (!files) {\n",

" return {\n",

" response: {\n",

" action: 'complete',\n",

" }\n",

" };\n",

" }\n",

"\n",

" for (const file of files) {\n",

" const li = document.createElement('li');\n",

" li.append(span(file.name, {fontWeight: 'bold'}));\n",

" li.append(span(\n",

" `(${file.type || 'n/a'}) - ${file.size} bytes, ` +\n",

" `last modified: ${\n",

" file.lastModifiedDate ? file.lastModifiedDate.toLocaleDateString() :\n",

" 'n/a'} - `));\n",

" const percent = span('0% done');\n",

" li.appendChild(percent);\n",

"\n",

" outputElement.appendChild(li);\n",

"\n",

" const fileDataPromise = new Promise((resolve) => {\n",

" const reader = new FileReader();\n",

" reader.onload = (e) => {\n",

" resolve(e.target.result);\n",

" };\n",

" reader.readAsArrayBuffer(file);\n",

" });\n",

" // Wait for the data to be ready.\n",

" let fileData = yield {\n",

" promise: fileDataPromise,\n",

" response: {\n",

" action: 'continue',\n",

" }\n",

" };\n",

"\n",

" // Use a chunked sending to avoid message size limits. See b/62115660.\n",

" let position = 0;\n",

" do {\n",

" const length = Math.min(fileData.byteLength - position, MAX\_PAYLOAD\_SIZE);\n",

" const chunk = new Uint8Array(fileData, position, length);\n",

" position += length;\n",

"\n",

" const base64 = btoa(String.fromCharCode.apply(null, chunk));\n",

" yield {\n",

" response: {\n",

" action: 'append',\n",

" file: file.name,\n",

" data: base64,\n",

" },\n",

" };\n",

"\n",

" let percentDone = fileData.byteLength === 0 ?\n",

" 100 :\n",

" Math.round((position / fileData.byteLength) \* 100);\n",

" percent.textContent = `${percentDone}% done`;\n",

"\n",

" } while (position < fileData.byteLength);\n",

" }\n",

"\n",

" // All done.\n",

" yield {\n",

" response: {\n",

" action: 'complete',\n",

" }\n",

" };\n",

"}\n",

"\n",

"scope.google = scope.google || {};\n",

"scope.google.colab = scope.google.colab || {};\n",

"scope.google.colab.\_files = {\n",

" \_uploadFiles,\n",

" \_uploadFilesContinue,\n",

"};\n",

"})(self);\n",

"</script> "

]

},

"metadata": {}

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Saving abalone.csv to abalone.csv\n"

]

}

],

"source": [

"from google.colab import files\n",

"uploaded = files.upload()"

]

},

{

"cell\_type": "code",

"source": [

"import pandas as pd \n",

"import numpy as np\n",

"import matplotlib.pyplot as plt\n",

"import seaborn as sns\n",

"import warnings \n",

"warnings.filterwarnings('ignore')"

],

"metadata": {

"id": "skj01X9Y3t3G"

},

"execution\_count": 2,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"data = pd.read\_csv(\"abalone.csv\")"

],

"metadata": {

"id": "4\_MJqY4z34pi"

},

"execution\_count": 3,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"\*\*2.Performing Visualization\*\*\n",

"\n",

"\*\*Univariate Analysis\*\*"

],

"metadata": {

"id": "E4WLiqDd38Nl"

}

},

{

"cell\_type": "code",

"source": [

"data.head()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 206

},

"id": "XICwfuyZ4F5Q",

"outputId": "4c89d615-04da-4621-9adb-f2107ab50d9c"

},

"execution\_count": 4,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Sex Length Diameter Height Whole weight Shucked weight Viscera weight \\\n",

"0 M 0.455 0.365 0.095 0.5140 0.2245 0.1010 \n",

"1 M 0.350 0.265 0.090 0.2255 0.0995 0.0485 \n",

"2 F 0.530 0.420 0.135 0.6770 0.2565 0.1415 \n",

"3 M 0.440 0.365 0.125 0.5160 0.2155 0.1140 \n",

"4 I 0.330 0.255 0.080 0.2050 0.0895 0.0395 \n",

"\n",

" Shell weight Rings \n",

"0 0.150 15 \n",

"1 0.070 7 \n",

"2 0.210 9 \n",

"3 0.155 10 \n",

"4 0.055 7 "

],

"text/html": [

"\n",

" <div id=\"df-52138ef8-91f4-4512-9bca-900304bf61b5\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Sex</th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" <th>Rings</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>0.455</td>\n",

" <td>0.365</td>\n",

" <td>0.095</td>\n",

" <td>0.5140</td>\n",

" <td>0.2245</td>\n",

" <td>0.1010</td>\n",

" <td>0.150</td>\n",

" <td>15</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>0.350</td>\n",

" <td>0.265</td>\n",

" <td>0.090</td>\n",

" <td>0.2255</td>\n",

" <td>0.0995</td>\n",

" <td>0.0485</td>\n",

" <td>0.070</td>\n",

" <td>7</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>F</td>\n",

" <td>0.530</td>\n",

" <td>0.420</td>\n",

" <td>0.135</td>\n",

" <td>0.6770</td>\n",

" <td>0.2565</td>\n",

" <td>0.1415</td>\n",

" <td>0.210</td>\n",

" <td>9</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>0.440</td>\n",

" <td>0.365</td>\n",

" <td>0.125</td>\n",

" <td>0.5160</td>\n",

" <td>0.2155</td>\n",

" <td>0.1140</td>\n",

" <td>0.155</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>I</td>\n",

" <td>0.330</td>\n",

" <td>0.255</td>\n",

" <td>0.080</td>\n",

" <td>0.2050</td>\n",

" <td>0.0895</td>\n",

" <td>0.0395</td>\n",

" <td>0.055</td>\n",

" <td>7</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-52138ef8-91f4-4512-9bca-900304bf61b5')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-52138ef8-91f4-4512-9bca-900304bf61b5 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-52138ef8-91f4-4512-9bca-900304bf61b5');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 4

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Diameter'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "VivZNyyL4JIT",

"outputId": "87c24c92-05a1-4ec8-d3d1-6d6749e3ff74"

},

"execution\_count": 5,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd75630590>"

]

},

"metadata": {},

"execution\_count": 5

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.hist(data['Diameter'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 352

},

"id": "WbGNdu514Ngg",

"outputId": "ae09e6b5-1b3d-46dc-fd9b-201df5a44ec4"

},

"execution\_count": 6,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(array([ 13., 66., 180., 344., 513., 812., 1017., 934., 275.,\n",

" 23.]),\n",

" array([0.055 , 0.1145, 0.174 , 0.2335, 0.293 , 0.3525, 0.412 , 0.4715,\n",

" 0.531 , 0.5905, 0.65 ]),\n",

" <a list of 10 Patch objects>)"

]

},

"metadata": {},

"execution\_count": 6

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.plot(data['Diameter'].head(10))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 282

},

"id": "P6IIpChF4RZd",

"outputId": "8b2c2743-dc0c-4223-c7ed-7524b3120955"

},

"execution\_count": 7,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"[<matplotlib.lines.Line2D at 0x7fcd750dcdd0>]"

]

},

"metadata": {},

"execution\_count": 7

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.pie(data['Diameter'].head(),autopct='%.3f')"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 508

},

"id": "2wTgIQw14UY3",

"outputId": "fe1674cf-4d2a-4706-ab09-a65ac1e52038"

},

"execution\_count": 8,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"([<matplotlib.patches.Wedge at 0x7fcd74fb7a10>,\n",

" <matplotlib.patches.Wedge at 0x7fcd74f44210>,\n",

" <matplotlib.patches.Wedge at 0x7fcd74f44190>,\n",

" <matplotlib.patches.Wedge at 0x7fcd74f4f350>,\n",

" <matplotlib.patches.Wedge at 0x7fcd74f4fe90>],\n",

" [Text(0.8507215626110557, 0.6973326486753676, ''),\n",

" Text(-0.32611344931648134, 1.0505474849691026, ''),\n",

" Text(-1.0998053664078908, -0.02069193128747144, ''),\n",

" Text(-0.08269436219656089, -1.096887251480709, ''),\n",

" Text(0.9758446362287218, -0.5076684409569241, '')],\n",

" [Text(0.46402994324239394, 0.3803632629138369, '21.856'),\n",

" Text(-0.17788006326353525, 0.5730259008922377, '15.868'),\n",

" Text(-0.5998938362224858, -0.011286507974984419, '25.150'),\n",

" Text(-0.045106015743578656, -0.5983021371712958, '21.856'),\n",

" Text(0.5322788924883937, -0.2769100587037768, '15.269')])"

]

},

"metadata": {},

"execution\_count": 8

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.distplot(data['Diameter'].head(300))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "4Pke2ujw4YXc",

"outputId": "b311b9cb-4516-4374-94a6-6bf79eb57d2b"

},

"execution\_count": 9,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd74f71810>"

]

},

"metadata": {},

"execution\_count": 9

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.scatter(data['Diameter'].head(400),data['Length'].head(400))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 282

},

"id": "TxP4XCMy4dEm",

"outputId": "250a8ef7-8b38-409e-8195-874e92abba1d"

},

"execution\_count": 11,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7fcd74f7c5d0>"

]

},

"metadata": {},

"execution\_count": 11

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.bar(data['Sex'].head(20),data['Rings'].head(20))\n",

"plt.title('Bar plot')\n",

"plt.xlabel('Diameter')\n",

"plt.ylabel('Rings')"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 312

},

"id": "4osPmUOh4hku",

"outputId": "c672ed08-e2c9-45bc-85fd-41b87a2c9e6e"

},

"execution\_count": 12,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Text(0, 0.5, 'Rings')"

]

},

"metadata": {},

"execution\_count": 12

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.barplot(data['Sex'], data['Rings'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "6L-muGNy4o0U",

"outputId": "8292959e-d736-4b30-c4ad-26ed4c30a14b"

},

"execution\_count": 13,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd742b0710>"

]

},

"metadata": {},

"execution\_count": 13

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.jointplot(data['Diameter'].head(50),data['Rings'].head(100))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 458

},

"id": "V5UEj3Sy4se3",

"outputId": "bdcd5925-71ac-4304-f7d3-6635ec874dd4"

},

"execution\_count": 14,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<seaborn.axisgrid.JointGrid at 0x7fcd7420d210>"

]

},

"metadata": {},

"execution\_count": 14

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x432 with 3 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.barplot('Diameter','Rings',hue='Sex',data=data.head())"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "1Mj9AZgP4yaj",

"outputId": "64237458-111f-4aef-eca5-6c0e3a5944cd"

},

"execution\_count": 15,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd71889f90>"

]

},

"metadata": {},

"execution\_count": 15

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.lineplot(data['Diameter'].head(),data['Rings'].head())"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "OYDADfYv42C0",

"outputId": "f106d3fc-34ec-4ab4-92cb-8aec08ef74d1"

},

"execution\_count": 16,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd717c4ad0>"

]

},

"metadata": {},

"execution\_count": 16

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Sex'].head(10),data['Diameter'].head(10),data['Rings'].head(10))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "xSzZ9q6947B1",

"outputId": "4536fa3d-4561-4f3c-ea57-3ae1eb94ba24"

},

"execution\_count": 17,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd71752ad0>"

]

},

"metadata": {},

"execution\_count": 17

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"fig=plt.figure(figsize=(8,5))\n",

"sns.heatmap(data.head().corr(),annot=True)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 410

},

"id": "b5qIh84m4-nU",

"outputId": "a7196ab8-7285-4a9c-b776-8a244ca52fa8"

},

"execution\_count": 18,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd7160a6d0>"

]

},

"metadata": {},

"execution\_count": 18

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 576x360 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.pairplot(data.head(),hue='Height')"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"id": "S6z-CUYn5DLA",

"outputId": "77731c99-9f86-474a-b06c-858c6aa80b65"

},

"execution\_count": 19,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<seaborn.axisgrid.PairGrid at 0x7fcd7149b090>"

]

},

"metadata": {},

"execution\_count": 19

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1324.62x1260 with 56 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.pairplot(data.head())"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"id": "pNvy89y15LVt",

"outputId": "5f806535-af8f-4fa5-9ee5-276afbf072a0"

},

"execution\_count": 20,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<seaborn.axisgrid.PairGrid at 0x7fcd6fe9b6d0>"

]

},

"metadata": {},

"execution\_count": 20

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1440x1440 with 72 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*3.Perform Descriptive Statistics on the dataset\*\*"

],

"metadata": {

"id": "ZR8E10Pa5SMu"

}

},

{

"cell\_type": "code",

"source": [

"data.head()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 206

},

"id": "MLveO2KT5VrU",

"outputId": "6392b9f8-399e-4abb-f027-35b3766c0569"

},

"execution\_count": 21,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Sex Length Diameter Height Whole weight Shucked weight Viscera weight \\\n",

"0 M 0.455 0.365 0.095 0.5140 0.2245 0.1010 \n",

"1 M 0.350 0.265 0.090 0.2255 0.0995 0.0485 \n",

"2 F 0.530 0.420 0.135 0.6770 0.2565 0.1415 \n",

"3 M 0.440 0.365 0.125 0.5160 0.2155 0.1140 \n",

"4 I 0.330 0.255 0.080 0.2050 0.0895 0.0395 \n",

"\n",

" Shell weight Rings \n",

"0 0.150 15 \n",

"1 0.070 7 \n",

"2 0.210 9 \n",

"3 0.155 10 \n",

"4 0.055 7 "

],

"text/html": [

"\n",

" <div id=\"df-6d87e62e-6137-4569-9c87-7edcefa0409d\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Sex</th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" <th>Rings</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>0.455</td>\n",

" <td>0.365</td>\n",

" <td>0.095</td>\n",

" <td>0.5140</td>\n",

" <td>0.2245</td>\n",

" <td>0.1010</td>\n",

" <td>0.150</td>\n",

" <td>15</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>0.350</td>\n",

" <td>0.265</td>\n",

" <td>0.090</td>\n",

" <td>0.2255</td>\n",

" <td>0.0995</td>\n",

" <td>0.0485</td>\n",

" <td>0.070</td>\n",

" <td>7</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>F</td>\n",

" <td>0.530</td>\n",

" <td>0.420</td>\n",

" <td>0.135</td>\n",

" <td>0.6770</td>\n",

" <td>0.2565</td>\n",

" <td>0.1415</td>\n",

" <td>0.210</td>\n",

" <td>9</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>0.440</td>\n",

" <td>0.365</td>\n",

" <td>0.125</td>\n",

" <td>0.5160</td>\n",

" <td>0.2155</td>\n",

" <td>0.1140</td>\n",

" <td>0.155</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>I</td>\n",

" <td>0.330</td>\n",

" <td>0.255</td>\n",

" <td>0.080</td>\n",

" <td>0.2050</td>\n",

" <td>0.0895</td>\n",

" <td>0.0395</td>\n",

" <td>0.055</td>\n",

" <td>7</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-6d87e62e-6137-4569-9c87-7edcefa0409d')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-6d87e62e-6137-4569-9c87-7edcefa0409d button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-6d87e62e-6137-4569-9c87-7edcefa0409d');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 21

}

]

},

{

"cell\_type": "code",

"source": [

"data.tail()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 206

},

"id": "GE9ARsNt5akp",

"outputId": "c36c04e9-8510-4282-ffc9-a397e2340015"

},

"execution\_count": 22,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Sex Length Diameter Height Whole weight Shucked weight \\\n",

"4172 F 0.565 0.450 0.165 0.8870 0.3700 \n",

"4173 M 0.590 0.440 0.135 0.9660 0.4390 \n",

"4174 M 0.600 0.475 0.205 1.1760 0.5255 \n",

"4175 F 0.625 0.485 0.150 1.0945 0.5310 \n",

"4176 M 0.710 0.555 0.195 1.9485 0.9455 \n",

"\n",

" Viscera weight Shell weight Rings \n",

"4172 0.2390 0.2490 11 \n",

"4173 0.2145 0.2605 10 \n",

"4174 0.2875 0.3080 9 \n",

"4175 0.2610 0.2960 10 \n",

"4176 0.3765 0.4950 12 "

],

"text/html": [

"\n",

" <div id=\"df-b65165d7-bc85-40f7-8ebe-c7079009d332\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Sex</th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" <th>Rings</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>4172</th>\n",

" <td>F</td>\n",

" <td>0.565</td>\n",

" <td>0.450</td>\n",

" <td>0.165</td>\n",

" <td>0.8870</td>\n",

" <td>0.3700</td>\n",

" <td>0.2390</td>\n",

" <td>0.2490</td>\n",

" <td>11</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4173</th>\n",

" <td>M</td>\n",

" <td>0.590</td>\n",

" <td>0.440</td>\n",

" <td>0.135</td>\n",

" <td>0.9660</td>\n",

" <td>0.4390</td>\n",

" <td>0.2145</td>\n",

" <td>0.2605</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4174</th>\n",

" <td>M</td>\n",

" <td>0.600</td>\n",

" <td>0.475</td>\n",

" <td>0.205</td>\n",

" <td>1.1760</td>\n",

" <td>0.5255</td>\n",

" <td>0.2875</td>\n",

" <td>0.3080</td>\n",

" <td>9</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4175</th>\n",

" <td>F</td>\n",

" <td>0.625</td>\n",

" <td>0.485</td>\n",

" <td>0.150</td>\n",

" <td>1.0945</td>\n",

" <td>0.5310</td>\n",

" <td>0.2610</td>\n",

" <td>0.2960</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4176</th>\n",

" <td>M</td>\n",

" <td>0.710</td>\n",

" <td>0.555</td>\n",

" <td>0.195</td>\n",

" <td>1.9485</td>\n",

" <td>0.9455</td>\n",

" <td>0.3765</td>\n",

" <td>0.4950</td>\n",

" <td>12</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-b65165d7-bc85-40f7-8ebe-c7079009d332')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-b65165d7-bc85-40f7-8ebe-c7079009d332 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-b65165d7-bc85-40f7-8ebe-c7079009d332');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 22

}

]

},

{

"cell\_type": "code",

"source": [

"data.info()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "0gm70RVH5drp",

"outputId": "c6e48ed1-6688-4a34-eb2b-1f7e572b8e0e"

},

"execution\_count": 23,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 4177 entries, 0 to 4176\n",

"Data columns (total 9 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 Sex 4177 non-null object \n",

" 1 Length 4177 non-null float64\n",

" 2 Diameter 4177 non-null float64\n",

" 3 Height 4177 non-null float64\n",

" 4 Whole weight 4177 non-null float64\n",

" 5 Shucked weight 4177 non-null float64\n",

" 6 Viscera weight 4177 non-null float64\n",

" 7 Shell weight 4177 non-null float64\n",

" 8 Rings 4177 non-null int64 \n",

"dtypes: float64(7), int64(1), object(1)\n",

"memory usage: 293.8+ KB\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"data.describe()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 300

},

"id": "HAqso6NV5g7H",

"outputId": "b7fa6f07-d9bc-4864-939a-0d30589c09ff"

},

"execution\_count": 24,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Length Diameter Height Whole weight Shucked weight \\\n",

"count 4177.000000 4177.000000 4177.000000 4177.000000 4177.000000 \n",

"mean 0.523992 0.407881 0.139516 0.828742 0.359367 \n",

"std 0.120093 0.099240 0.041827 0.490389 0.221963 \n",

"min 0.075000 0.055000 0.000000 0.002000 0.001000 \n",

"25% 0.450000 0.350000 0.115000 0.441500 0.186000 \n",

"50% 0.545000 0.425000 0.140000 0.799500 0.336000 \n",

"75% 0.615000 0.480000 0.165000 1.153000 0.502000 \n",

"max 0.815000 0.650000 1.130000 2.825500 1.488000 \n",

"\n",

" Viscera weight Shell weight Rings \n",

"count 4177.000000 4177.000000 4177.000000 \n",

"mean 0.180594 0.238831 9.933684 \n",

"std 0.109614 0.139203 3.224169 \n",

"min 0.000500 0.001500 1.000000 \n",

"25% 0.093500 0.130000 8.000000 \n",

"50% 0.171000 0.234000 9.000000 \n",

"75% 0.253000 0.329000 11.000000 \n",

"max 0.760000 1.005000 29.000000 "

],

"text/html": [

"\n",

" <div id=\"df-87a0f027-b2a3-4261-8c7d-0927fe5ed4a8\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" <th>Rings</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>0.523992</td>\n",

" <td>0.407881</td>\n",

" <td>0.139516</td>\n",

" <td>0.828742</td>\n",

" <td>0.359367</td>\n",

" <td>0.180594</td>\n",

" <td>0.238831</td>\n",

" <td>9.933684</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>0.120093</td>\n",

" <td>0.099240</td>\n",

" <td>0.041827</td>\n",

" <td>0.490389</td>\n",

" <td>0.221963</td>\n",

" <td>0.109614</td>\n",

" <td>0.139203</td>\n",

" <td>3.224169</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>0.075000</td>\n",

" <td>0.055000</td>\n",

" <td>0.000000</td>\n",

" <td>0.002000</td>\n",

" <td>0.001000</td>\n",

" <td>0.000500</td>\n",

" <td>0.001500</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>0.450000</td>\n",

" <td>0.350000</td>\n",

" <td>0.115000</td>\n",

" <td>0.441500</td>\n",

" <td>0.186000</td>\n",

" <td>0.093500</td>\n",

" <td>0.130000</td>\n",

" <td>8.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>0.545000</td>\n",

" <td>0.425000</td>\n",

" <td>0.140000</td>\n",

" <td>0.799500</td>\n",

" <td>0.336000</td>\n",

" <td>0.171000</td>\n",

" <td>0.234000</td>\n",

" <td>9.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>0.615000</td>\n",

" <td>0.480000</td>\n",

" <td>0.165000</td>\n",

" <td>1.153000</td>\n",

" <td>0.502000</td>\n",

" <td>0.253000</td>\n",

" <td>0.329000</td>\n",

" <td>11.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>0.815000</td>\n",

" <td>0.650000</td>\n",

" <td>1.130000</td>\n",

" <td>2.825500</td>\n",

" <td>1.488000</td>\n",

" <td>0.760000</td>\n",

" <td>1.005000</td>\n",

" <td>29.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-87a0f027-b2a3-4261-8c7d-0927fe5ed4a8')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-87a0f027-b2a3-4261-8c7d-0927fe5ed4a8 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-87a0f027-b2a3-4261-8c7d-0927fe5ed4a8');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 24

}

]

},

{

"cell\_type": "code",

"source": [

"data.mode().T"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 332

},

"id": "z36j8Lwt5kJV",

"outputId": "d89ac863-ca35-458c-ed90-363d3f1bd4ff"

},

"execution\_count": 25,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" 0 1\n",

"Sex M NaN\n",

"Length 0.55 0.625\n",

"Diameter 0.45 NaN\n",

"Height 0.15 NaN\n",

"Whole weight 0.2225 NaN\n",

"Shucked weight 0.175 NaN\n",

"Viscera weight 0.1715 NaN\n",

"Shell weight 0.275 NaN\n",

"Rings 9.0 NaN"

],

"text/html": [

"\n",

" <div id=\"df-0e4361eb-4f36-4748-903f-4193282c21f8\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>0</th>\n",

" <th>1</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>Sex</th>\n",

" <td>M</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Length</th>\n",

" <td>0.55</td>\n",

" <td>0.625</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Diameter</th>\n",

" <td>0.45</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Height</th>\n",

" <td>0.15</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Whole weight</th>\n",

" <td>0.2225</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Shucked weight</th>\n",

" <td>0.175</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Viscera weight</th>\n",

" <td>0.1715</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Shell weight</th>\n",

" <td>0.275</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Rings</th>\n",

" <td>9.0</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-0e4361eb-4f36-4748-903f-4193282c21f8')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-0e4361eb-4f36-4748-903f-4193282c21f8 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-0e4361eb-4f36-4748-903f-4193282c21f8');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 25

}

]

},

{

"cell\_type": "code",

"source": [

"data.shape"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "VSOZxV-l5nc6",

"outputId": "b106c72e-2642-4628-cbb0-b5b128a151ec"

},

"execution\_count": 26,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(4177, 9)"

]

},

"metadata": {},

"execution\_count": 26

}

]

},

{

"cell\_type": "code",

"source": [

"data.kurt()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "lZGJjmp-5q5D",

"outputId": "78b5281b-7f9b-456c-81cf-ea0e9048b732"

},

"execution\_count": 27,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Length 0.064621\n",

"Diameter -0.045476\n",

"Height 76.025509\n",

"Whole weight -0.023644\n",

"Shucked weight 0.595124\n",

"Viscera weight 0.084012\n",

"Shell weight 0.531926\n",

"Rings 2.330687\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 27

}

]

},

{

"cell\_type": "code",

"source": [

"data.skew()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "yjbfHgA05usL",

"outputId": "777619d5-152e-4d51-cd74-38e880c34f33"

},

"execution\_count": 28,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Length -0.639873\n",

"Diameter -0.609198\n",

"Height 3.128817\n",

"Whole weight 0.530959\n",

"Shucked weight 0.719098\n",

"Viscera weight 0.591852\n",

"Shell weight 0.620927\n",

"Rings 1.114102\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 28

}

]

},

{

"cell\_type": "code",

"source": [

"data.var()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "WLlDtoF55x0r",

"outputId": "feeafda2-1d8a-4430-e550-1780fc9c1119"

},

"execution\_count": 29,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Length 0.014422\n",

"Diameter 0.009849\n",

"Height 0.001750\n",

"Whole weight 0.240481\n",

"Shucked weight 0.049268\n",

"Viscera weight 0.012015\n",

"Shell weight 0.019377\n",

"Rings 10.395266\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 29

}

]

},

{

"cell\_type": "code",

"source": [

"data.nunique()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "p48x6DK451mf",

"outputId": "3d0b0eac-4056-4477-9a9d-740cb33b37b9"

},

"execution\_count": 30,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Sex 3\n",

"Length 134\n",

"Diameter 111\n",

"Height 51\n",

"Whole weight 2429\n",

"Shucked weight 1515\n",

"Viscera weight 880\n",

"Shell weight 926\n",

"Rings 28\n",

"dtype: int64"

]

},

"metadata": {},

"execution\_count": 30

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*4.Check for missing values and deal with them\*\*"

],

"metadata": {

"id": "MgW8dlUO55q\_"

}

},

{

"cell\_type": "code",

"source": [

"data.isna()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 424

},

"id": "rAj6qFg55\_T7",

"outputId": "9c4766e2-49ca-43f3-9a45-140482b4da74"

},

"execution\_count": 31,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Sex Length Diameter Height Whole weight Shucked weight \\\n",

"0 False False False False False False \n",

"1 False False False False False False \n",

"2 False False False False False False \n",

"3 False False False False False False \n",

"4 False False False False False False \n",

"... ... ... ... ... ... ... \n",

"4172 False False False False False False \n",

"4173 False False False False False False \n",

"4174 False False False False False False \n",

"4175 False False False False False False \n",

"4176 False False False False False False \n",

"\n",

" Viscera weight Shell weight Rings \n",

"0 False False False \n",

"1 False False False \n",

"2 False False False \n",

"3 False False False \n",

"4 False False False \n",

"... ... ... ... \n",

"4172 False False False \n",

"4173 False False False \n",

"4174 False False False \n",

"4175 False False False \n",

"4176 False False False \n",

"\n",

"[4177 rows x 9 columns]"

],

"text/html": [

"\n",

" <div id=\"df-9c446a51-7fec-4979-8a72-7a20a67a4a8d\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Sex</th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" <th>Rings</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4172</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4173</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4174</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4175</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4176</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>4177 rows Ã— 9 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-9c446a51-7fec-4979-8a72-7a20a67a4a8d')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-9c446a51-7fec-4979-8a72-7a20a67a4a8d button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-9c446a51-7fec-4979-8a72-7a20a67a4a8d');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 31

}

]

},

{

"cell\_type": "code",

"source": [

"data.isna().any()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "3KfXUqlF6Dkz",

"outputId": "2d59cdea-3b5f-4bbe-a311-51a49b175419"

},

"execution\_count": 32,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Sex False\n",

"Length False\n",

"Diameter False\n",

"Height False\n",

"Whole weight False\n",

"Shucked weight False\n",

"Viscera weight False\n",

"Shell weight False\n",

"Rings False\n",

"dtype: bool"

]

},

"metadata": {},

"execution\_count": 32

}

]

},

{

"cell\_type": "code",

"source": [

"data.isna().sum()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Qj\_uim2D6GtE",

"outputId": "dfce9f68-5db0-4950-d94c-a338e3d4ecf3"

},

"execution\_count": 33,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Sex 0\n",

"Length 0\n",

"Diameter 0\n",

"Height 0\n",

"Whole weight 0\n",

"Shucked weight 0\n",

"Viscera weight 0\n",

"Shell weight 0\n",

"Rings 0\n",

"dtype: int64"

]

},

"metadata": {},

"execution\_count": 33

}

]

},

{

"cell\_type": "code",

"source": [

"data.isna().any().sum()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "UPd\_4rxH6JcV",

"outputId": "f2646908-942d-4478-d072-156c61a63fa2"

},

"execution\_count": 34,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0"

]

},

"metadata": {},

"execution\_count": 34

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*5.Find the outliers and replace them outliers\*\*"

],

"metadata": {

"id": "Z2cUuQRz6Mx0"

}

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Diameter'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "eKPB6Jxz6PRt",

"outputId": "bdfb1206-1938-4d58-9785-abb1a171df34"

},

"execution\_count": 35,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6cc0b690>"

]

},

"metadata": {},

"execution\_count": 35

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"quant=data.quantile(q=[0.25,0.75])\n",

"quant"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 112

},

"id": "PyEeoK2Z6Tl-",

"outputId": "279a4ea5-9767-4e56-e73b-a9ebf5b6953f"

},

"execution\_count": 36,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Length Diameter Height Whole weight Shucked weight Viscera weight \\\n",

"0.25 0.450 0.35 0.115 0.4415 0.186 0.0935 \n",

"0.75 0.615 0.48 0.165 1.1530 0.502 0.2530 \n",

"\n",

" Shell weight Rings \n",

"0.25 0.130 8.0 \n",

"0.75 0.329 11.0 "

],

"text/html": [

"\n",

" <div id=\"df-e8bf301b-882c-409a-aa46-1a5a81342fe4\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" <th>Rings</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0.25</th>\n",

" <td>0.450</td>\n",

" <td>0.35</td>\n",

" <td>0.115</td>\n",

" <td>0.4415</td>\n",

" <td>0.186</td>\n",

" <td>0.0935</td>\n",

" <td>0.130</td>\n",

" <td>8.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>0.75</th>\n",

" <td>0.615</td>\n",

" <td>0.48</td>\n",

" <td>0.165</td>\n",

" <td>1.1530</td>\n",

" <td>0.502</td>\n",

" <td>0.2530</td>\n",

" <td>0.329</td>\n",

" <td>11.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-e8bf301b-882c-409a-aa46-1a5a81342fe4')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-e8bf301b-882c-409a-aa46-1a5a81342fe4 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-e8bf301b-882c-409a-aa46-1a5a81342fe4');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 36

}

]

},

{

"cell\_type": "code",

"source": [

"iqr=quant.loc[0.75]-quant.loc[0.25]\n",

"iqr"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "XkihPM8u6XcN",

"outputId": "44e9c72d-5360-4a3c-bbad-b51235377bf9"

},

"execution\_count": 37,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Length 0.1650\n",

"Diameter 0.1300\n",

"Height 0.0500\n",

"Whole weight 0.7115\n",

"Shucked weight 0.3160\n",

"Viscera weight 0.1595\n",

"Shell weight 0.1990\n",

"Rings 3.0000\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 37

}

]

},

{

"cell\_type": "code",

"source": [

"low=quant.loc[0.25]-(1.5\*iqr)\n",

"low"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "9ap1VMk46afK",

"outputId": "f967dc10-7b6c-47a6-c82a-8ed5a7bee959"

},

"execution\_count": 38,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Length 0.20250\n",

"Diameter 0.15500\n",

"Height 0.04000\n",

"Whole weight -0.62575\n",

"Shucked weight -0.28800\n",

"Viscera weight -0.14575\n",

"Shell weight -0.16850\n",

"Rings 3.50000\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 38

}

]

},

{

"cell\_type": "code",

"source": [

"up=quant.loc[0.75]+(1.5\*iqr)\n",

"up"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "7ntVsHak6dW9",

"outputId": "2be8bf35-c838-41f2-dfaa-da839fbf9281"

},

"execution\_count": 39,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Length 0.86250\n",

"Diameter 0.67500\n",

"Height 0.24000\n",

"Whole weight 2.22025\n",

"Shucked weight 0.97600\n",

"Viscera weight 0.49225\n",

"Shell weight 0.62750\n",

"Rings 15.50000\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 39

}

]

},

{

"cell\_type": "code",

"source": [

"data['Diameter']=np.where(data['Diameter']<0.155,0.4078,data['Diameter'])\n",

"sns.boxplot(data['Diameter'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "ZyWdY43q6g64",

"outputId": "4aaab2b0-78ad-4172-b96c-c11d1557f00b"

},

"execution\_count": 40,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6cbe1510>"

]

},

"metadata": {},

"execution\_count": 40

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Length'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "xsAC3Xf76kO-",

"outputId": "c8ac10d8-29b0-4358-c537-a7ab16df754f"

},

"execution\_count": 41,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6cb41410>"

]

},

"metadata": {},

"execution\_count": 41

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data['Length']=np.where(data['Length']<0.23,0.52, data['Length'])\n",

"sns.boxplot(data['Length'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "IST4w1sG6naw",

"outputId": "d77dc0b2-8593-4303-dcd8-9e7f514687af"

},

"execution\_count": 42,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6cb31350>"

]

},

"metadata": {},

"execution\_count": 42

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Height'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "TqzJUfDD6q8O",

"outputId": "47d888c3-3598-4e03-f4c1-de22c8b429f1"

},

"execution\_count": 43,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6ca91950>"

]

},

"metadata": {},

"execution\_count": 43

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data['Height']=np.where(data['Height']<0.04,0.139, data['Height'])\n",

"data['Height']=np.where(data['Height']>0.23,0.139, data['Height'])\n",

"sns.boxplot(data['Height'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "NOdWKQuV6zJm",

"outputId": "6bad1d92-7868-4481-97c3-40edd7362195"

},

"execution\_count": 44,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6ca82050>"

]

},

"metadata": {},

"execution\_count": 44

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Whole weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "CDMl7-uY63LK",

"outputId": "17757c2d-a350-44d7-be34-0dc412d4217c"

},

"execution\_count": 45,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c9f2090>"

]

},

"metadata": {},

"execution\_count": 45

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data['Whole weight']=np.where(data['Whole weight']>0.9,0.82, data['Whole weight'])\n",

"sns.boxplot(data['Whole weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "2z3Faxe666hk",

"outputId": "1a23d7b5-e1a7-4957-e843-48e07035b18b"

},

"execution\_count": 46,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c9e9390>"

]

},

"metadata": {},

"execution\_count": 46

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Shucked weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "QI2quaxP69tl",

"outputId": "c94e092b-5ba6-4d3e-a6f6-78ee8f295bc8"

},

"execution\_count": 47,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c8c3f10>"

]

},

"metadata": {},

"execution\_count": 47

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data['Shucked weight']=np.where(data['Shucked weight']>0.93,0.35, data['Shucked weight'])\n",

"sns.boxplot(data['Shucked weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "x8Uy8vbB7CK4",

"outputId": "43900cbe-8d45-41f0-b594-2f305be96f25"

},

"execution\_count": 48,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c836f50>"

]

},

"metadata": {},

"execution\_count": 48

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Viscera weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "6Gwq\_hgZ7GLP",

"outputId": "eb9ee86c-49c2-4782-a026-41730c526561"

},

"execution\_count": 49,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c801ad0>"

]

},

"metadata": {},

"execution\_count": 49

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data['Viscera weight']=np.where(data['Viscera weight']>0.46,0.18, data['Viscera weight'])\n",

"sns.boxplot(data['Viscera weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "A1Wues0q7IxO",

"outputId": "1a820336-d5c3-4e60-a076-b216b9239876"

},

"execution\_count": 50,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c785b90>"

]

},

"metadata": {},

"execution\_count": 50

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.boxplot(data['Shell weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "XH0WpA7h7MF3",

"outputId": "d1d37c83-b3d7-4bfb-cfc6-903e85a93569"

},

"execution\_count": 51,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c768b50>"

]

},

"metadata": {},

"execution\_count": 51

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data['Shell weight']=np.where(data['Shell weight']>0.61,0.2388, data['Shell weight'])\n",

"sns.boxplot(data['Shell weight'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "8\_mxsV017PDc",

"outputId": "1c9e64df-bf25-4c6d-8d74-48ae07b0462f"

},

"execution\_count": 52,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fcd6c6df850>"

]

},

"metadata": {},

"execution\_count": 52

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*6.Check for Categorical columns and perform encoding.\*\*"

],

"metadata": {

"id": "mBu2n5y67Snh"

}

},

{

"cell\_type": "code",

"source": [

"data['Sex'].replace({'M':1,'F':0,'I':2},inplace=True)\n",

"data"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 424

},

"id": "kDqHKVw17U3g",

"outputId": "e1f33f93-dbfa-4b9b-9773-532824240b6c"

},

"execution\_count": 53,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Sex Length Diameter Height Whole weight Shucked weight \\\n",

"0 1 0.455 0.365 0.095 0.5140 0.2245 \n",

"1 1 0.350 0.265 0.090 0.2255 0.0995 \n",

"2 0 0.530 0.420 0.135 0.6770 0.2565 \n",

"3 1 0.440 0.365 0.125 0.5160 0.2155 \n",

"4 2 0.330 0.255 0.080 0.2050 0.0895 \n",

"... ... ... ... ... ... ... \n",

"4172 0 0.565 0.450 0.165 0.8870 0.3700 \n",

"4173 1 0.590 0.440 0.135 0.8200 0.4390 \n",

"4174 1 0.600 0.475 0.205 0.8200 0.5255 \n",

"4175 0 0.625 0.485 0.150 0.8200 0.5310 \n",

"4176 1 0.710 0.555 0.195 0.8200 0.3500 \n",

"\n",

" Viscera weight Shell weight Rings \n",

"0 0.1010 0.1500 15 \n",

"1 0.0485 0.0700 7 \n",

"2 0.1415 0.2100 9 \n",

"3 0.1140 0.1550 10 \n",

"4 0.0395 0.0550 7 \n",

"... ... ... ... \n",

"4172 0.2390 0.2490 11 \n",

"4173 0.2145 0.2605 10 \n",

"4174 0.2875 0.3080 9 \n",

"4175 0.2610 0.2960 10 \n",

"4176 0.3765 0.4950 12 \n",

"\n",

"[4177 rows x 9 columns]"

],

"text/html": [

"\n",

" <div id=\"df-1dc962ad-5a00-42d9-8969-a6753f6810df\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Sex</th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" <th>Rings</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>0.455</td>\n",

" <td>0.365</td>\n",

" <td>0.095</td>\n",

" <td>0.5140</td>\n",

" <td>0.2245</td>\n",

" <td>0.1010</td>\n",

" <td>0.1500</td>\n",

" <td>15</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>1</td>\n",

" <td>0.350</td>\n",

" <td>0.265</td>\n",

" <td>0.090</td>\n",

" <td>0.2255</td>\n",

" <td>0.0995</td>\n",

" <td>0.0485</td>\n",

" <td>0.0700</td>\n",

" <td>7</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>0</td>\n",

" <td>0.530</td>\n",

" <td>0.420</td>\n",

" <td>0.135</td>\n",

" <td>0.6770</td>\n",

" <td>0.2565</td>\n",

" <td>0.1415</td>\n",

" <td>0.2100</td>\n",

" <td>9</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>1</td>\n",

" <td>0.440</td>\n",

" <td>0.365</td>\n",

" <td>0.125</td>\n",

" <td>0.5160</td>\n",

" <td>0.2155</td>\n",

" <td>0.1140</td>\n",

" <td>0.1550</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2</td>\n",

" <td>0.330</td>\n",

" <td>0.255</td>\n",

" <td>0.080</td>\n",

" <td>0.2050</td>\n",

" <td>0.0895</td>\n",

" <td>0.0395</td>\n",

" <td>0.0550</td>\n",

" <td>7</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4172</th>\n",

" <td>0</td>\n",

" <td>0.565</td>\n",

" <td>0.450</td>\n",

" <td>0.165</td>\n",

" <td>0.8870</td>\n",

" <td>0.3700</td>\n",

" <td>0.2390</td>\n",

" <td>0.2490</td>\n",

" <td>11</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4173</th>\n",

" <td>1</td>\n",

" <td>0.590</td>\n",

" <td>0.440</td>\n",

" <td>0.135</td>\n",

" <td>0.8200</td>\n",

" <td>0.4390</td>\n",

" <td>0.2145</td>\n",

" <td>0.2605</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4174</th>\n",

" <td>1</td>\n",

" <td>0.600</td>\n",

" <td>0.475</td>\n",

" <td>0.205</td>\n",

" <td>0.8200</td>\n",

" <td>0.5255</td>\n",

" <td>0.2875</td>\n",

" <td>0.3080</td>\n",

" <td>9</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4175</th>\n",

" <td>0</td>\n",

" <td>0.625</td>\n",

" <td>0.485</td>\n",

" <td>0.150</td>\n",

" <td>0.8200</td>\n",

" <td>0.5310</td>\n",

" <td>0.2610</td>\n",

" <td>0.2960</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4176</th>\n",

" <td>1</td>\n",

" <td>0.710</td>\n",

" <td>0.555</td>\n",

" <td>0.195</td>\n",

" <td>0.8200</td>\n",

" <td>0.3500</td>\n",

" <td>0.3765</td>\n",

" <td>0.4950</td>\n",

" <td>12</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>4177 rows Ã— 9 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-1dc962ad-5a00-42d9-8969-a6753f6810df')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-1dc962ad-5a00-42d9-8969-a6753f6810df button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-1dc962ad-5a00-42d9-8969-a6753f6810df');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 53

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*7.Split the data into dependent and independent variables.\*\*"

],

"metadata": {

"id": "bwyRAhMO7acZ"

}

},

{

"cell\_type": "code",

"source": [

"x=data.drop(columns= ['Rings'])\n",

"y=data['Rings']\n",

"x"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 424

},

"id": "o7\_rcHJv7ccB",

"outputId": "72bb0535-1e40-4367-87c6-9c87dd642447"

},

"execution\_count": 54,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Sex Length Diameter Height Whole weight Shucked weight \\\n",

"0 1 0.455 0.365 0.095 0.5140 0.2245 \n",

"1 1 0.350 0.265 0.090 0.2255 0.0995 \n",

"2 0 0.530 0.420 0.135 0.6770 0.2565 \n",

"3 1 0.440 0.365 0.125 0.5160 0.2155 \n",

"4 2 0.330 0.255 0.080 0.2050 0.0895 \n",

"... ... ... ... ... ... ... \n",

"4172 0 0.565 0.450 0.165 0.8870 0.3700 \n",

"4173 1 0.590 0.440 0.135 0.8200 0.4390 \n",

"4174 1 0.600 0.475 0.205 0.8200 0.5255 \n",

"4175 0 0.625 0.485 0.150 0.8200 0.5310 \n",

"4176 1 0.710 0.555 0.195 0.8200 0.3500 \n",

"\n",

" Viscera weight Shell weight \n",

"0 0.1010 0.1500 \n",

"1 0.0485 0.0700 \n",

"2 0.1415 0.2100 \n",

"3 0.1140 0.1550 \n",

"4 0.0395 0.0550 \n",

"... ... ... \n",

"4172 0.2390 0.2490 \n",

"4173 0.2145 0.2605 \n",

"4174 0.2875 0.3080 \n",

"4175 0.2610 0.2960 \n",

"4176 0.3765 0.4950 \n",

"\n",

"[4177 rows x 8 columns]"

],

"text/html": [

"\n",

" <div id=\"df-7d25b2c8-f089-49c3-a9f4-9a0d1b236a9f\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Sex</th>\n",

" <th>Length</th>\n",

" <th>Diameter</th>\n",

" <th>Height</th>\n",

" <th>Whole weight</th>\n",

" <th>Shucked weight</th>\n",

" <th>Viscera weight</th>\n",

" <th>Shell weight</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>0.455</td>\n",

" <td>0.365</td>\n",

" <td>0.095</td>\n",

" <td>0.5140</td>\n",

" <td>0.2245</td>\n",

" <td>0.1010</td>\n",

" <td>0.1500</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>1</td>\n",

" <td>0.350</td>\n",

" <td>0.265</td>\n",

" <td>0.090</td>\n",

" <td>0.2255</td>\n",

" <td>0.0995</td>\n",

" <td>0.0485</td>\n",

" <td>0.0700</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>0</td>\n",

" <td>0.530</td>\n",

" <td>0.420</td>\n",

" <td>0.135</td>\n",

" <td>0.6770</td>\n",

" <td>0.2565</td>\n",

" <td>0.1415</td>\n",

" <td>0.2100</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>1</td>\n",

" <td>0.440</td>\n",

" <td>0.365</td>\n",

" <td>0.125</td>\n",

" <td>0.5160</td>\n",

" <td>0.2155</td>\n",

" <td>0.1140</td>\n",

" <td>0.1550</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2</td>\n",

" <td>0.330</td>\n",

" <td>0.255</td>\n",

" <td>0.080</td>\n",

" <td>0.2050</td>\n",

" <td>0.0895</td>\n",

" <td>0.0395</td>\n",

" <td>0.0550</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4172</th>\n",

" <td>0</td>\n",

" <td>0.565</td>\n",

" <td>0.450</td>\n",

" <td>0.165</td>\n",

" <td>0.8870</td>\n",

" <td>0.3700</td>\n",

" <td>0.2390</td>\n",

" <td>0.2490</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4173</th>\n",

" <td>1</td>\n",

" <td>0.590</td>\n",

" <td>0.440</td>\n",

" <td>0.135</td>\n",

" <td>0.8200</td>\n",

" <td>0.4390</td>\n",

" <td>0.2145</td>\n",

" <td>0.2605</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4174</th>\n",

" <td>1</td>\n",

" <td>0.600</td>\n",

" <td>0.475</td>\n",

" <td>0.205</td>\n",

" <td>0.8200</td>\n",

" <td>0.5255</td>\n",

" <td>0.2875</td>\n",

" <td>0.3080</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4175</th>\n",

" <td>0</td>\n",

" <td>0.625</td>\n",

" <td>0.485</td>\n",

" <td>0.150</td>\n",

" <td>0.8200</td>\n",

" <td>0.5310</td>\n",

" <td>0.2610</td>\n",

" <td>0.2960</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4176</th>\n",

" <td>1</td>\n",

" <td>0.710</td>\n",

" <td>0.555</td>\n",

" <td>0.195</td>\n",

" <td>0.8200</td>\n",

" <td>0.3500</td>\n",

" <td>0.3765</td>\n",

" <td>0.4950</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>4177 rows Ã— 8 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-7d25b2c8-f089-49c3-a9f4-9a0d1b236a9f')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-7d25b2c8-f089-49c3-a9f4-9a0d1b236a9f button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-7d25b2c8-f089-49c3-a9f4-9a0d1b236a9f');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 54

}

]

},

{

"cell\_type": "code",

"source": [

"y"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "2AmRQHKl7gpC",

"outputId": "5efa5486-1cc8-40ad-89ee-1575f7d2f3fa"

},

"execution\_count": 55,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0 15\n",

"1 7\n",

"2 9\n",

"3 10\n",

"4 7\n",

" ..\n",

"4172 11\n",

"4173 10\n",

"4174 9\n",

"4175 10\n",

"4176 12\n",

"Name: Rings, Length: 4177, dtype: int64"

]

},

"metadata": {},

"execution\_count": 55

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*8.Scale the independent variables\*\*"

],

"metadata": {

"id": "035RalJI7j4g"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.preprocessing import scale\n",

"x = scale(x)\n",

"x"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "PWJPLBW77l5r",

"outputId": "40596a5f-9b1f-463e-950f-3766396d153f"

},

"execution\_count": 56,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([[-0.0105225 , -0.67088921, -0.50179694, ..., -0.61037964,\n",

" -0.7328165 , -0.64358742],\n",

" [-0.0105225 , -1.61376082, -1.57304487, ..., -1.22513334,\n",

" -1.24343929, -1.25742181],\n",

" [-1.26630752, 0.00259051, 0.08738942, ..., -0.45300269,\n",

" -0.33890749, -0.18321163],\n",

" ...,\n",

" [-0.0105225 , 0.63117159, 0.67657577, ..., 0.86994729,\n",

" 1.08111018, 0.56873549],\n",

" [-1.26630752, 0.85566483, 0.78370057, ..., 0.89699645,\n",

" 0.82336724, 0.47666033],\n",

" [-0.0105225 , 1.61894185, 1.53357412, ..., 0.00683308,\n",

" 1.94673739, 2.00357336]])"

]

},

"metadata": {},

"execution\_count": 56

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*9.Split the data into training and testing\*\*"

],

"metadata": {

"id": "hNXX8mri7vmL"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.model\_selection import train\_test\_split\n",

"x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size = 0.2)\n",

"print(x\_train.shape, x\_test.shape)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Sy6AlLu77xro",

"outputId": "08e9f05d-648f-4345-e158-5e7f4d3fb2b6"

},

"execution\_count": 57,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"(3341, 8) (836, 8)\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*10.Build the Model\*\*"

],

"metadata": {

"id": "E9ak5WZK72oX"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.linear\_model import LinearRegression\n",

"MLR=LinearRegression()"

],

"metadata": {

"id": "yKEjmAQl77u8"

},

"execution\_count": 58,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"\*\*11.Train the model\*\*"

],

"metadata": {

"id": "NrWn3jew7-Zc"

}

},

{

"cell\_type": "code",

"source": [

"MLR.fit(x\_train,y\_train)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "wAmozZim8A7h",

"outputId": "59ce28d9-33ae-4299-cc5c-0285cc831c71"

},

"execution\_count": 59,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"LinearRegression()"

]

},

"metadata": {},

"execution\_count": 59

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*12.Test the model\*\*"

],

"metadata": {

"id": "ea5qHm4H8Gkh"

}

},

{

"cell\_type": "code",

"source": [

"y\_pred=MLR.predict(x\_test)\n",

"y\_pred"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "pKBkvizt8ItQ",

"outputId": "48e6b118-71cf-46c9-e581-88d5ab49313e"

},

"execution\_count": 60,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([ 6.3204331 , 10.41671748, 13.91911179, 12.29316277, 8.7273177 ,\n",

" 11.04369928, 12.40210281, 11.6992544 , 12.01785949, 6.57983392,\n",

" 11.91353764, 10.79661591, 11.56560952, 10.14326497, 13.16762604,\n",

" 9.34621768, 10.76904478, 11.88283609, 9.34461447, 10.08802992,\n",

" 12.80140942, 9.58177975, 11.20908126, 10.3662699 , 10.0168299 ,\n",

" 15.92815446, 15.93700213, 7.36066362, 13.2889134 , 10.1579858 ,\n",

" 11.62833855, 11.08597007, 11.60253151, 11.74194458, 9.75151497,\n",

" 9.16685512, 7.93960537, 10.04563481, 10.81773394, 10.55133893,\n",

" 7.19389026, 9.30303442, 10.83957317, 10.63432914, 10.19371808,\n",

" 13.47423856, 9.06825076, 6.69843582, 13.38213142, 9.62823486,\n",

" 8.20174551, 7.79183041, 9.3338472 , 11.08195328, 11.25321895,\n",

" 6.11231204, 10.6960639 , 9.23348159, 7.76425036, 11.65342323,\n",

" 12.6024271 , 7.49694081, 9.71678931, 7.41119139, 6.94925679,\n",

" 6.34706174, 9.99734923, 6.70117631, 10.71374432, 9.59457302,\n",

" 7.07847213, 6.6940933 , 9.30356123, 13.66698224, 9.71369221,\n",

" 17.36952958, 7.81225327, 8.86909973, 9.29540502, 11.03405521,\n",

" 12.90720962, 13.03952065, 4.90843127, 9.50619996, 10.09434256,\n",

" 8.67296752, 9.03746047, 8.33310609, 10.60445018, 9.66636969,\n",

" 7.67351279, 8.74447193, 12.37470593, 7.70552082, 11.35599144,\n",

" 11.25726129, 10.02276461, 8.01953433, 11.39538114, 7.92288557,\n",

" 11.02588274, 7.02530311, 10.80014326, 13.22266766, 11.41469264,\n",

" 7.5577235 , 6.83654146, 6.97820486, 10.29150052, 9.1851768 ,\n",

" 9.72122817, 9.29569276, 11.98122676, 9.87982582, 8.55374278,\n",

" 7.67912597, 10.93152036, 11.90656204, 11.93625854, 12.59760271,\n",

" 11.87092001, 5.99671728, 9.20248712, 11.18185068, 11.13316757,\n",

" 12.85726928, 9.50993961, 9.39438115, 10.55793101, 8.61221838,\n",

" 7.12344177, 7.0075169 , 7.56528442, 14.02672909, 13.39176121,\n",

" 10.27099354, 13.04124533, 9.72264547, 11.63284409, 3.06922786,\n",

" 8.60297955, 10.80917425, 11.27118306, 6.4757245 , 10.27830248,\n",

" 10.17409116, 10.39178358, 6.11330216, 8.27295199, 12.0413644 ,\n",

" 10.43536813, 11.12820999, 10.56478101, 12.12900686, 9.0459273 ,\n",

" 6.50569617, 8.65471113, 11.17391657, 4.17641665, 6.45933408,\n",

" 10.94174559, 10.56404265, 7.32806471, 10.90718067, 8.76983179,\n",

" 9.54866214, 9.71969088, 9.19215908, 11.19107958, 9.95023994,\n",

" 10.33050587, 11.98860703, 5.76011208, 8.82560871, 8.26963359,\n",

" 6.41006108, 7.62776781, 7.77958091, 10.53587014, 8.89399096,\n",

" 11.50322847, 6.46552063, 6.62035734, 11.27313616, 8.28747988,\n",

" 12.05544015, 11.6973709 , 12.73972343, 11.36996234, 7.97256548,\n",

" 9.42073857, 11.25296103, 8.05208624, 10.99827477, 8.28671759,\n",

" 11.9443616 , 11.82872121, 9.74400382, 8.90145486, 8.57310105,\n",

" 7.40827472, 11.17489105, 10.0697987 , 9.82070981, 7.33964403,\n",

" 14.9428325 , 7.76026974, 12.77292992, 6.50073351, 11.29473941,\n",

" 11.88889387, 7.67192672, 11.10156897, 12.84247625, 6.80849608,\n",

" 12.6708819 , 9.56757524, 10.85921143, 10.87947611, 12.27788605,\n",

" 12.6343008 , 8.1422213 , 9.74592291, 6.44390326, 12.16548247,\n",

" 9.68626598, 7.22046972, 8.56653749, 12.44724289, 10.28404261,\n",

" 11.71924716, 6.57910451, 5.62586891, 8.74565212, 13.18908706,\n",

" 11.27318326, 9.38021053, 6.42749173, 10.51442882, 10.27033818,\n",

" 8.87378728, 8.03400206, 11.0092839 , 6.85674274, 12.05051373,\n",

" 6.69272915, 11.10719822, 6.44922697, 6.74873571, 11.24095121,\n",

" 8.70909665, 8.62212669, 12.29685468, 7.11389904, 13.34703661,\n",

" 11.08283081, 8.3922623 , 8.72365009, 9.29973736, 11.15975322,\n",

" 12.66049713, 12.94737341, 8.01252297, 14.6856936 , 7.96867592,\n",

" 12.20170934, 11.66756932, 10.13791171, 8.26813719, 7.49073268,\n",

" 7.45324295, 10.31038428, 7.99447275, 6.35227192, 7.59845402,\n",

" 9.53208026, 12.11826236, 8.53704575, 10.36120273, 9.19371627,\n",

" 7.45521096, 11.4440868 , 11.06664961, 8.04725105, 8.60560981,\n",

" 13.80099793, 8.15216844, 9.98062314, 12.74725885, 7.89003787,\n",

" 7.81036747, 11.23112413, 9.22750995, 7.11114746, 11.3826884 ,\n",

" 10.44398727, 10.04248851, 8.62061001, 3.85346936, 7.5615557 ,\n",

" 8.50076581, 10.54855634, 7.11113889, 9.96259791, 7.16231236,\n",

" 12.37775152, 10.86115071, 11.27994987, 10.87920364, 11.95932837,\n",

" 12.58798285, 6.81134765, 13.30643307, 8.75328509, 10.4636244 ,\n",

" 6.84946941, 6.04480792, 10.31013095, 10.05120403, 10.08224692,\n",

" 12.18589719, 11.04298064, 11.76136111, 16.89806828, 11.7634019 ,\n",

" 12.54460021, 6.97010314, 8.2340867 , 11.26531484, 6.46952952,\n",

" 13.53317546, 6.12586648, 11.98108901, 7.35287441, 7.41235981,\n",

" 11.07260837, 13.22783978, 7.45878435, 8.35296852, 11.2446654 ,\n",

" 10.95510293, 9.28430345, 11.26170271, 11.24714775, 12.82380236,\n",

" 9.80910029, 11.14851263, 8.93507131, 10.48203237, 12.04680128,\n",

" 10.45927558, 9.55970274, 7.96600883, 10.06900881, 11.36587023,\n",

" 10.41813388, 9.21419839, 11.06093353, 6.92189968, 8.15727622,\n",

" 10.012316 , 12.70541689, 9.39808987, 6.6137354 , 11.6823883 ,\n",

" 8.93368204, 9.8336508 , 5.81570851, 11.56276923, 10.87893019,\n",

" 12.67972891, 11.85979648, 9.00130033, 6.14079186, 12.26577301,\n",

" 9.6207766 , 13.62864939, 10.1323747 , 10.83544245, 10.21793752,\n",

" 12.44123618, 10.3300243 , 10.14492559, 10.62996651, 8.89575318,\n",

" 8.06752328, 13.48778231, 11.14849439, 9.65207022, 6.95214119,\n",

" 10.07613281, 10.75989529, 8.58803654, 6.92578354, 8.01017204,\n",

" 11.17089437, 8.45408936, 11.64213563, 12.60969766, 10.16532537,\n",

" 11.62084877, 15.56193132, 10.97215069, 8.4493314 , 9.82278678,\n",

" 12.39838927, 8.08544756, 6.5110862 , 12.62157502, 11.75762029,\n",

" 6.46343384, 11.62791829, 6.52577378, 8.09948656, 13.23056122,\n",

" 11.45545838, 7.88182125, 12.12655647, 6.2450235 , 6.85235997,\n",

" 12.02248022, 10.65499013, 9.90529278, 11.28140838, 10.69826234,\n",

" 9.30623121, 11.63073 , 3.98764614, 11.92613755, 9.15352777,\n",

" 7.68806474, 3.54599443, 7.62372757, 7.06669705, 9.81810625,\n",

" 13.32005718, 11.98237927, 13.07882898, 3.04187812, 7.59086968,\n",

" 9.36009959, 9.86269052, 13.76835171, 7.7100412 , 7.41688263,\n",

" 9.71775277, 11.70856406, 15.51747154, 9.99121698, 8.41116835,\n",

" 11.23369554, 10.28615638, 8.33105769, 11.25375717, 10.40778208,\n",

" 11.00070579, 7.11341792, 8.80790735, 8.48565552, 10.16565602,\n",

" 9.91261772, 7.2985758 , 9.6534037 , 9.36018584, 9.65373534,\n",

" 8.97920536, 7.84480454, 8.5142807 , 8.37849389, 10.19677213,\n",

" 10.95329693, 9.3449988 , 8.72403761, 10.29803147, 12.52009373,\n",

" 9.78551797, 9.18215269, 3.79981384, 10.15480885, 11.42070351,\n",

" 13.01165766, 8.61472752, 10.74128738, 7.77372474, 10.7495373 ,\n",

" 13.1988739 , 12.14129185, 11.13040855, 7.97013745, 8.09427469,\n",

" 8.24986448, 12.50126965, 10.70650153, 11.39971209, 9.97895784,\n",

" 8.74272637, 6.73410649, 7.88221575, 10.09612739, 9.23281125,\n",

" 10.31769674, 9.04469981, 12.02675645, 10.28143082, 11.22841252,\n",

" 10.59397785, 7.00727845, 11.20494643, 6.08455412, 8.21752142,\n",

" 8.75838271, 10.64278017, 7.68964903, 9.11330244, 7.97323295,\n",

" 10.67847311, 11.18094806, 10.37966496, 8.1376328 , 5.97082578,\n",

" 9.62348713, 8.05203882, 10.16553992, 11.79604071, 10.22484933,\n",

" 6.33506556, 10.43104428, 9.56329834, 7.36501586, 7.05098109,\n",

" 7.66685698, 9.6674169 , 12.17542299, 6.90212824, 10.68875015,\n",

" 12.46702201, 12.7907768 , 11.94254907, 10.76057313, 14.50338836,\n",

" 9.45736076, 10.58793144, 11.61470888, 9.59426968, 7.97892048,\n",

" 5.98062632, 10.46381968, 11.22282427, 11.23699908, 9.84161685,\n",

" 11.95892655, 11.13850591, 7.04483007, 14.2501626 , 7.79353126,\n",

" 6.45117294, 9.38456463, 8.0332045 , 9.93050258, 9.73190714,\n",

" 12.4102944 , 9.10114827, 12.8963848 , 8.41153033, 7.44370096,\n",

" 10.09617217, 12.22181033, 6.02879425, 6.01483262, 7.28727815,\n",

" 10.28199375, 12.21067661, 11.57946536, 7.85545622, 10.81624125,\n",

" 11.2752125 , 7.26977458, 8.63869718, 10.40949326, 10.20252101,\n",

" 10.20344455, 10.57901435, 10.01700129, 9.70078194, 11.06833077,\n",

" 11.42980061, 12.95055323, 12.04956675, 11.1941666 , 10.68557991,\n",

" 8.73554328, 13.34096933, 11.70650078, 9.7755308 , 6.66302311,\n",

" 7.21267073, 10.86084576, 9.73431779, 10.00611373, 9.51459537,\n",

" 11.90089784, 11.89430231, 10.99515042, 9.11142599, 9.54140857,\n",

" 11.63225945, 9.22065757, 6.81841855, 10.58189845, 6.84242119,\n",

" 6.41315328, 13.86031405, 6.53307695, 8.41661816, 13.26853831,\n",

" 10.44159961, 11.87182979, 6.77949327, 11.96989656, 11.25073461,\n",

" 6.89500467, 12.06087353, 10.14394287, 14.31986306, 8.66127254,\n",

" 11.00746954, 13.61282646, 12.91959445, 9.02821703, 10.38398051,\n",

" 14.4229355 , 8.78857021, 7.01907878, 6.46694262, 4.54961394,\n",

" 6.49789999, 7.87772534, 8.43138963, 10.29554282, 10.89357213,\n",

" 6.25196775, 10.1893591 , 11.21340643, 10.60619567, 13.5199916 ,\n",

" 10.56030953, 11.7950339 , 11.1765925 , 10.38141072, 7.42990336,\n",

" 13.41096123, 11.62301184, 10.73094454, 11.70334894, 11.57181567,\n",

" 10.80461361, 11.27209529, 8.56904593, 8.84843246, 11.74444264,\n",

" 8.50007959, 11.25363184, 9.77410872, 7.58320424, 10.07883544,\n",

" 10.7446824 , 10.68195259, 10.37965052, 10.87542144, 6.24270277,\n",

" 6.52867268, 12.98653921, 5.95809754, 11.10118332, 12.69797726,\n",

" 9.70047358, 10.75030017, 8.83916295, 12.6391372 , 8.71600195,\n",

" 10.44189797, 11.87689143, 9.81059669, 12.51448567, 6.06009622,\n",

" 11.66106213, 9.27212378, 13.17603547, 9.39914507, 10.39087202,\n",

" 6.02708892, 9.5017869 , 10.47304199, 6.30384712, 7.96227565,\n",

" 11.44159974, 6.92243911, 11.89243095, 12.39706362, 8.9185901 ,\n",

" 9.85066928, 9.17410809, 9.51438019, 9.26165211, 9.49644599,\n",

" 11.14148633, 7.81053465, 7.63742108, 11.11640929, 8.50561753,\n",

" 11.5155862 , 12.39745744, 11.48258178, 7.8396838 , 10.75488787,\n",

" 10.70876852, 10.33932821, 6.21872817, 10.96735542, 6.99161329,\n",

" 13.7782618 , 12.67371856, 12.37124617, 11.36370102, 9.919346 ,\n",

" 11.60811423, 11.03313201, 11.90824699, 11.24919199, 10.16011371,\n",

" 9.19387302, 11.31221344, 11.9582661 , 6.66072436, 12.82287322,\n",

" 6.82223028, 10.86908107, 12.0465645 , 10.22018145, 10.33663637,\n",

" 9.56617643, 12.81543797, 9.55624429, 6.40748877, 9.18245993,\n",

" 11.08940482, 10.30969793, 8.33725839, 7.02183935, 9.10858165,\n",

" 11.09159809, 10.13547053, 12.56156355, 11.88035156, 12.69848062,\n",

" 12.24640964, 7.7853713 , 10.2718951 , 11.05286956, 8.2143034 ,\n",

" 7.77738987, 7.94845909, 9.54088307, 6.29089679, 12.56416397,\n",

" 13.15876524, 10.18189042, 10.46284865, 7.04692937, 6.97763746,\n",

" 10.56467822, 13.85391527, 10.5765819 , 9.66605944, 10.55446891,\n",

" 11.7841348 , 9.21295862, 7.10579534, 10.81410739, 10.09791717,\n",

" 10.96195048, 10.95031944, 13.62602527, 10.59581821, 4.23190635,\n",

" 10.48523331, 10.40104327, 11.84799028, 10.64954069, 10.59187651,\n",

" 6.26661907, 10.32822889, 14.86508345, 11.72214699, 9.02572005,\n",

" 7.28096119, 13.32894198, 11.6627331 , 10.82653286, 6.96844694,\n",

" 10.71453819, 3.84412879, 7.99397393, 11.45321996, 6.65586194,\n",

" 10.49845467, 11.41461698, 12.30088487, 13.67361223, 10.23315043,\n",

" 8.33589046, 6.81098171, 12.27922881, 12.0691788 , 9.34292715,\n",

" 12.58929765, 11.03943241, 9.64362509, 10.51726338, 6.70400142,\n",

" 8.55015073, 6.43624566, 12.98430085, 6.7791815 , 7.33221017,\n",

" 11.78661551, 9.10621109, 11.67704104, 11.78859596, 8.80722284,\n",

" 6.34468821, 10.79631808, 9.36532306, 9.71346119, 11.29123914,\n",

" 8.93602679])"

]

},

"metadata": {},

"execution\_count": 60

}

]

},

{

"cell\_type": "code",

"source": [

"pred=MLR.predict(x\_train)\n",

"pred"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "\_CuD-rvz8RfR",

"outputId": "56d9b591-7cce-40e0-9b70-5488b4943cd1"

},

"execution\_count": 61,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([9.67807776, 9.90237308, 8.732808 , ..., 8.23154309, 9.17793652,\n",

" 8.04066563])"

]

},

"metadata": {},

"execution\_count": 61

}

]

},

{

"cell\_type": "code",

"source": [

"from sklearn.metrics import r2\_score\n",

"accuracy=r2\_score(y\_test,y\_pred)\n",

"accuracy"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "j2auH2if8XEA",

"outputId": "3e0a3b8d-04bf-4be9-b813-28f8734e6b92"

},

"execution\_count": 62,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0.45246173731319095"

]

},

"metadata": {},

"execution\_count": 62

}

]

},

{

"cell\_type": "code",

"source": [

"MLR.predict([[1,0.455,0.365,0.095,0.5140,0.2245,0.1010,0.150]])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Nv-qjVBt8a7Z",

"outputId": "6051a2c9-68c0-419f-bad0-0980bffde4e2"

},

"execution\_count": 63,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([9.88121105])"

]

},

"metadata": {},

"execution\_count": 63

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*13.Measure the performance using Metrics\*\*"

],

"metadata": {

"id": "jssKAz608enL"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn import metrics\n",

"from sklearn.metrics import mean\_squared\_error\n",

"np.sqrt(mean\_squared\_error(y\_test,y\_pred))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "1XSpOxy38iZR",

"outputId": "7206eb29-cbad-4d01-cba7-67521a296193"

},

"execution\_count": 64,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"2.426157459129611"

]

},

"metadata": {},

"execution\_count": 64

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*LASSO\*\*"

],

"metadata": {

"id": "VVs0Q9Tk8nsh"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.linear\_model import Lasso, Ridge\n",

"#intialising model\n",

"lso=Lasso(alpha=0.01,normalize=True)\n",

"#fit the model\n",

"lso.fit(x\_train,y\_train)\n",

"Lasso(alpha=0.01, normalize=True)\n",

"#prediction on test data\n",

"lso\_pred=lso.predict(x\_test)\n",

"#coef\n",

"coef=lso.coef\_\n",

"coef"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "xcZ6xayH8pw-",

"outputId": "961cca84-afaf-4875-8519-2edbc928c609"

},

"execution\_count": 65,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([-0. , 0. , 0. , 0.47895382, 0.1231748 ,\n",

" 0. , 0. , 0.84464209])"

]

},

"metadata": {},

"execution\_count": 65

}

]

},

{

"cell\_type": "code",

"source": [

"from sklearn import metrics\n",

"from sklearn.metrics import mean\_squared\_error\n",

"metrics.r2\_score(y\_test,lso\_pred)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "XnFm7f1J8uw3",

"outputId": "19d98b5c-a1a4-4f0d-a73d-f90bf3fda6fc"

},

"execution\_count": 66,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0.3408644820717798"

]

},

"metadata": {},

"execution\_count": 66

}

]

},

{

"cell\_type": "code",

"source": [

"np.sqrt(mean\_squared\_error(y\_test,lso\_pred))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "hT6MhpOb8yQB",

"outputId": "68d9b138-64e5-43a7-817e-8e0d29d11b91"

},

"execution\_count": 67,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"2.661945158379675"

]

},

"metadata": {},

"execution\_count": 67

}

]

},

{

"cell\_type": "markdown",

"source": [

"\*\*RIDGE\*\*"

],

"metadata": {

"id": "Eo0vabck81cd"

}

},

{

"cell\_type": "code",

"source": [

"#initialising model\n",

"rg=Ridge(alpha=0.01,normalize=True)\n",

"#fit the model\n",

"rg.fit(x\_train,y\_train)\n",

"Ridge(alpha=0.01, normalize=True)\n",

"#prediction\n",

"rg\_pred=rg.predict(x\_test)\n",

"rg\_pred"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "xGqh11PZ83Ts",

"outputId": "078d058d-6b97-42ad-9c34-b3053325efd4"

},

"execution\_count": 68,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([ 6.31931908, 10.30764994, 13.79582136, 12.31898366, 8.76153971,\n",

" 11.03161104, 12.36947473, 11.61494959, 11.9751636 , 6.61427002,\n",

" 11.96025268, 10.72794019, 11.47832347, 10.11563414, 13.06595338,\n",

" 9.39260908, 10.76339441, 11.91725195, 9.36307394, 10.08739488,\n",

" 12.81067168, 9.60509865, 11.22161077, 10.34000965, 9.99490475,\n",

" 15.73170012, 15.73827506, 7.39070197, 13.28647279, 10.27222883,\n",

" 11.60238358, 11.12815632, 11.54610466, 11.74210077, 9.74066812,\n",

" 9.18758732, 7.95443356, 9.97019442, 10.84527542, 10.5864829 ,\n",

" 7.21708698, 9.26208697, 10.7752225 , 10.59013091, 10.22155425,\n",

" 13.35380749, 9.15950505, 6.7399079 , 13.2683363 , 9.60102394,\n",

" 8.2303643 , 7.8098864 , 9.39868717, 11.12458359, 11.22465236,\n",

" 6.08517442, 10.71988191, 9.22838517, 7.83437767, 11.55747904,\n",

" 12.53949383, 7.51301724, 9.78148647, 7.37997405, 6.95728771,\n",

" 6.35737948, 9.93938109, 6.69320708, 10.65733704, 9.63910534,\n",

" 7.08460623, 6.75306126, 9.35523418, 13.54420957, 9.75927226,\n",

" 17.10845005, 7.80794412, 8.86341648, 9.31305116, 10.93831159,\n",

" 12.7937996 , 12.90820043, 5.0134048 , 9.52092556, 10.09121624,\n",

" 8.69256796, 9.05325416, 8.38837108, 10.60016343, 9.6674175 ,\n",

" 7.68947352, 8.75395963, 12.3557545 , 7.68394763, 11.31578086,\n",

" 11.33781658, 10.02388881, 8.09043934, 11.33539573, 7.92289814,\n",

" 11.03286658, 7.02120848, 10.88303082, 13.18223161, 11.41674891,\n",

" 7.5435003 , 6.86054075, 7.00011005, 10.37216015, 9.23843865,\n",

" 9.7407151 , 9.41073419, 11.98466884, 9.91238351, 8.55695304,\n",

" 7.70340225, 10.96050891, 11.89423954, 11.90887509, 12.48907721,\n",

" 11.86120525, 5.96452761, 9.17094432, 11.10705265, 11.18316911,\n",

" 12.90325314, 9.50633456, 9.46969255, 10.45975924, 8.59999852,\n",

" 7.14040966, 7.03512578, 7.5768313 , 13.93668208, 13.38311964,\n",

" 10.27726556, 13.01546334, 9.72632411, 11.65895358, 3.54059511,\n",

" 8.69150731, 10.77801729, 11.23928477, 6.46704004, 10.36834548,\n",

" 10.25986306, 10.44359406, 6.10329506, 8.27710513, 12.00502229,\n",

" 10.50911591, 11.1635973 , 10.68310036, 12.09904065, 9.0770447 ,\n",

" 6.51065452, 8.66830233, 11.11653573, 4.30977451, 6.59138577,\n",

" 10.94701874, 10.61697739, 7.33598097, 10.95584092, 8.76981394,\n",

" 9.56950846, 9.72685674, 9.29695148, 11.09334904, 9.96323736,\n",

" 10.29354294, 11.96082702, 5.75254094, 8.80783142, 8.31092081,\n",

" 6.47041872, 7.62067235, 7.79228716, 10.61544322, 8.884205 ,\n",

" 11.52089469, 6.44153939, 6.6562387 , 11.24714188, 8.27032166,\n",

" 11.97661312, 11.79159964, 12.64675032, 11.3290589 , 7.96929134,\n",

" 9.42738402, 11.14331892, 8.03753191, 10.94763703, 8.28272049,\n",

" 11.94238196, 11.81869976, 9.74774078, 8.87383728, 8.54169155,\n",

" 7.48524588, 11.16514963, 10.09267601, 9.82811196, 7.36491078,\n",

" 14.8249117 , 7.77936035, 12.76515154, 6.50468619, 11.28683401,\n",

" 11.89036262, 7.69687081, 11.17198455, 12.8645229 , 6.79644567,\n",

" 12.5831777 , 9.62311367, 10.87136695, 10.88101051, 12.21534543,\n",

" 12.56210374, 8.12521978, 9.71614736, 6.57664458, 12.12980021,\n",

" 9.71075339, 7.24914697, 8.54824009, 12.42390808, 10.27008299,\n",

" 11.70234014, 6.59560366, 5.60827459, 8.73943569, 13.08562025,\n",

" 11.35929656, 9.50125178, 6.56112747, 10.54122297, 10.33817158,\n",

" 8.87013363, 8.02438993, 11.03741513, 6.86779209, 12.02722733,\n",

" 6.75229057, 11.04470428, 6.5816006 , 6.74869576, 11.25503165,\n",

" 8.71340313, 8.62307556, 12.17896811, 7.11651179, 13.22858199,\n",

" 11.03542994, 8.68242386, 8.75388289, 9.31220621, 11.22739362,\n",

" 12.58476122, 12.82060341, 8.02801869, 14.55839048, 7.98037436,\n",

" 12.10783358, 11.62139305, 10.16727021, 8.30524457, 7.51349955,\n",

" 7.46121172, 10.32851777, 8.04684163, 6.36435226, 7.59357285,\n",

" 9.54603296, 12.10820207, 8.55668517, 10.39331414, 9.41983441,\n",

" 7.42807836, 11.40629368, 11.05520789, 8.05109737, 8.65000544,\n",

" 13.58247147, 8.14524251, 10.00438331, 12.63327892, 7.88626539,\n",

" 7.78947885, 11.2024822 , 9.27216037, 7.07444208, 11.30967829,\n",

" 10.45239844, 9.98694105, 8.65990093, 4.29981281, 7.5941543 ,\n",

" 8.55386218, 10.57426301, 7.16001569, 9.94249674, 7.17959432,\n",

" 12.35195505, 10.87662953, 11.29655082, 10.84966402, 11.94395065,\n",

" 12.56662233, 6.79834044, 13.25453573, 8.74285659, 10.57648071,\n",

" 6.8571706 , 6.02091079, 10.27946695, 10.00085498, 10.05819242,\n",

" 12.15656737, 11.14207726, 11.74517275, 16.70567848, 11.75540218,\n",

" 12.47195304, 6.98180704, 8.21551924, 11.23895147, 6.4647415 ,\n",

" 13.46791483, 6.19344091, 11.89417773, 7.34528918, 7.39844718,\n",

" 11.06232167, 13.18240636, 7.45749635, 8.36887778, 11.20039609,\n",

" 10.83314542, 9.26784592, 11.27284919, 11.24248021, 12.75983855,\n",

" 9.87484509, 11.10084045, 8.92379925, 10.51308497, 12.03212064,\n",

" 10.49475111, 9.53780694, 8.03166459, 10.01767226, 11.29033102,\n",

" 10.39520108, 9.27355462, 11.03737951, 6.95567127, 8.18868545,\n",

" 9.99570418, 12.62595835, 9.41602801, 6.57214301, 11.77656811,\n",

" 8.9158198 , 9.82002317, 5.81096223, 11.4449004 , 10.80813086,\n",

" 12.62661509, 11.77759639, 9.05634512, 6.14145144, 12.27932405,\n",

" 9.60446376, 13.48746808, 10.25395348, 10.83270912, 10.26680509,\n",

" 12.31403901, 10.32300297, 10.10752188, 10.65525768, 9.11181257,\n",

" 8.22364442, 13.41054005, 11.21487992, 9.74573568, 6.93231313,\n",

" 10.15223273, 10.70741877, 8.56768141, 6.89660991, 8.00943831,\n",

" 11.18463079, 8.44540559, 11.6368763 , 12.54188753, 10.36873679,\n",

" 11.59701616, 15.36276679, 10.93096729, 8.4837404 , 9.87468507,\n",

" 12.40906942, 8.09620535, 6.53384853, 12.59237314, 11.69289115,\n",

" 6.44482931, 11.57567051, 6.53179035, 8.11667395, 13.14296473,\n",

" 11.36305564, 7.88749094, 12.14814009, 6.22425857, 6.87449781,\n",

" 11.98024604, 10.70312894, 9.86368846, 11.32439485, 10.78701134,\n",

" 9.23087468, 11.65149477, 4.12562143, 11.88219103, 9.12840322,\n",

" 7.69486574, 3.98799754, 7.65691657, 7.0800083 , 9.92520215,\n",

" 13.2280293 , 12.03666064, 13.00591443, 3.23286291, 7.5569394 ,\n",

" 9.33962421, 9.8145342 , 13.57832375, 7.73077975, 7.44378546,\n",

" 9.77590317, 11.70808254, 15.37889062, 9.99283888, 8.44724429,\n",

" 11.18675112, 10.30383125, 8.35865754, 11.18156373, 10.4796538 ,\n",

" 10.9753174 , 7.10623113, 8.84051473, 8.51243614, 10.15236445,\n",

" 9.93214235, 7.29437459, 9.68774226, 9.33126569, 9.69498808,\n",

" 8.98237053, 7.86606124, 8.5140542 , 8.37921868, 10.15310283,\n",

" 11.00526371, 9.33242261, 8.72801542, 10.21741897, 12.4511196 ,\n",

" 9.80454475, 9.19775264, 3.94163456, 10.17739852, 11.4031804 ,\n",

" 12.92150008, 8.63728959, 10.84254401, 7.77935205, 10.78111568,\n",

" 13.15159484, 12.109736 , 11.11617832, 7.91430278, 8.09629638,\n",

" 8.24793335, 12.44927378, 10.70132135, 11.38081293, 9.97640527,\n",

" 9.07303767, 6.77104799, 7.86422764, 10.13037574, 9.1691105 ,\n",

" 10.32830243, 9.06190563, 12.00763433, 10.30155434, 11.22618212,\n",

" 10.61394836, 7.00569632, 11.29420189, 6.03567063, 8.20919389,\n",

" 8.77409129, 10.64517899, 7.72236576, 9.13349834, 7.99315367,\n",

" 10.65238653, 11.18826412, 10.36985468, 8.16618498, 5.94998513,\n",

" 9.63243405, 8.09959521, 10.2074877 , 11.82933143, 10.30153818,\n",

" 6.38385323, 10.48168257, 9.65539719, 7.38541236, 7.01090534,\n",

" 7.67613982, 9.63431759, 12.09709646, 6.89452111, 10.64845756,\n",

" 12.46022613, 12.72290926, 11.87343812, 10.84867098, 14.33564436,\n",

" 9.55985338, 10.63768457, 11.70455864, 9.73327024, 8.02645694,\n",

" 5.96241088, 10.41727495, 11.20842762, 11.17467459, 9.81517856,\n",

" 11.84831639, 11.17382828, 7.05703418, 14.16785346, 7.84023577,\n",

" 6.58380091, 9.38465197, 8.07288712, 9.95835429, 9.75719368,\n",

" 12.43144658, 9.12494045, 12.75807834, 8.39377984, 7.42605551,\n",

" 10.12653434, 12.20001587, 6.04054428, 5.97291445, 7.2752374 ,\n",

" 10.38284617, 12.11604503, 11.57993857, 7.78822109, 10.79691819,\n",

" 11.21808768, 7.3029155 , 8.63002723, 10.53191991, 10.21730051,\n",

" 10.17976086, 10.59655412, 10.16367237, 9.67932217, 11.02445557,\n",

" 11.39591646, 12.88598564, 12.04374052, 11.21720034, 10.74474711,\n",

" 8.74474474, 13.25094754, 11.69840145, 9.76840641, 6.7142728 ,\n",

" 7.21973318, 10.8463731 , 9.76991861, 10.11890483, 9.4546138 ,\n",

" 11.8602123 , 11.88059978, 11.01157051, 9.15613152, 9.48976217,\n",

" 11.58908588, 9.21357867, 6.82292916, 10.64664015, 6.85584691,\n",

" 6.45600551, 13.78952112, 6.52188174, 8.39963417, 13.237635 ,\n",

" 10.50377928, 11.91248147, 6.79029197, 11.94262374, 11.23832993,\n",

" 6.89736394, 11.99240252, 10.13931483, 14.21666322, 8.63922355,\n",

" 10.88585561, 13.49982674, 12.84971553, 9.07719077, 10.35881911,\n",

" 14.36981281, 8.79436754, 7.02230965, 6.46127587, 4.6842177 ,\n",

" 6.49386321, 7.88159336, 8.41099649, 10.36042231, 10.89907432,\n",

" 6.20690652, 10.17428308, 11.14022218, 10.63404138, 13.39636496,\n",

" 10.63485358, 11.80711008, 11.14592069, 10.35737586, 7.4740565 ,\n",

" 13.27346531, 11.64520865, 10.80045937, 11.5597028 , 11.59900224,\n",

" 10.82315725, 11.26395985, 8.63096482, 8.83958938, 11.72421617,\n",

" 8.51808684, 11.27290533, 9.78133078, 7.60339093, 10.06581155,\n",

" 10.72319935, 10.68969162, 10.31902336, 10.91630474, 6.22930528,\n",

" 6.49455808, 12.88396152, 5.93945896, 11.14153998, 12.62487168,\n",

" 9.74461294, 10.84191783, 8.86260749, 12.56421912, 8.74580872,\n",

" 10.38458491, 11.78305746, 9.84950651, 12.43960364, 6.10244407,\n",

" 11.60529996, 9.44847078, 13.15765729, 9.36743266, 10.39555643,\n",

" 6.02052695, 9.50582504, 10.48481156, 6.31058576, 7.9387112 ,\n",

" 11.45409577, 6.94448547, 11.87005773, 12.30520074, 8.94715289,\n",

" 9.82268447, 9.17909668, 9.54357167, 9.2430071 , 9.51536648,\n",

" 11.15601425, 7.79191147, 7.63865868, 11.19764263, 8.50520755,\n",

" 11.51225569, 12.31338119, 11.44438825, 7.82344804, 10.74650792,\n",

" 10.76979129, 10.36556441, 6.2352391 , 11.0280009 , 7.0027628 ,\n",

" 13.60556836, 12.51874346, 12.37482879, 11.29621514, 9.91151125,\n",

" 11.55540459, 11.05328228, 11.88393325, 11.26570594, 10.25207002,\n",

" 9.20658079, 11.30248574, 11.91971063, 6.65762381, 12.78048536,\n",

" 6.84004066, 10.89035763, 11.95199252, 10.29088085, 10.31386785,\n",

" 9.57976002, 12.80265645, 9.62013946, 6.4025002 , 9.19785553,\n",

" 11.03814628, 10.37798812, 8.36296873, 7.02266739, 9.16942337,\n",

" 11.05613201, 10.14778335, 12.49045472, 11.91455123, 12.63715378,\n",

" 12.22572741, 7.8019063 , 10.26936336, 11.03840342, 8.21498915,\n",

" 7.85241208, 7.96368392, 9.53789495, 6.3118226 , 12.5905665 ,\n",

" 13.05596432, 10.2453772 , 10.46458777, 7.06132256, 7.00983751,\n",

" 10.63606953, 13.75606687, 10.5812314 , 9.74178047, 10.5100023 ,\n",

" 11.66844777, 9.26727868, 7.1761172 , 10.83591448, 10.20478003,\n",

" 11.00546249, 11.13069277, 13.58873816, 10.59913809, 4.37007882,\n",

" 10.55245106, 10.45514144, 11.77244164, 10.6718892 , 10.57166522,\n",

" 6.28071813, 10.31302917, 14.65718562, 11.825138 , 9.05032041,\n",

" 7.2984991 , 13.29299725, 11.64221134, 10.83397062, 7.0039474 ,\n",

" 10.73128163, 3.98267314, 7.99220196, 11.43920317, 6.67247168,\n",

" 10.46434628, 11.45247706, 12.32387555, 13.5904689 , 10.32054563,\n",

" 8.31532952, 6.80506069, 12.21896684, 11.94598044, 9.3639535 ,\n",

" 12.61681405, 10.97547068, 9.74743217, 10.42793806, 6.72060975,\n",

" 8.60325179, 6.43066295, 12.82424837, 6.83481137, 7.32979059,\n",

" 11.65953247, 9.13948298, 11.67402272, 11.74593729, 8.85145392,\n",

" 6.33660685, 10.87374762, 9.43187008, 9.76869802, 11.41887009,\n",

" 8.95390465])"

]

},

"metadata": {},

"execution\_count": 68

}

]

},

{

"cell\_type": "code",

"source": [

"rg.coef\_"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "mbCo4fHb88Y1",

"outputId": "03c0f927-ad5d-403e-e619-92d6f8992890"

},

"execution\_count": 69,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([-0.3074739 , -0.73150514, 0.23303655, 0.99723138, 0.94304227,\n",

" -1.36153292, -0.05594202, 1.75904754])"

]

},

"metadata": {},

"execution\_count": 69

}

]

},

{

"cell\_type": "code",

"source": [

"metrics.r2\_score(y\_test,rg\_pred)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "i5M7l6O39KAd",

"outputId": "09d61a30-c8e6-4c1a-cf5b-1624035ec840"

},

"execution\_count": 70,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0.45111716055161055"

]

},

"metadata": {},

"execution\_count": 70

}

]

},

{

"cell\_type": "code",

"source": [

"np.sqrt(mean\_squared\_error(y\_test,rg\_pred))"

],

"metadata": {

"id": "FsLba\_Tq9QYP",

"outputId": "4c16a48f-bfbf-4638-9706-02785910d96d",

"colab": {

"base\_uri": "https://localhost:8080/"

}

},

"execution\_count": 71,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"2.4291345612955157"

]

},

"metadata": {},

"execution\_count": 71

}

]

}

]

}