# Importing necessary libraries and loading the data

Install the 'hvplot' library quietly (without displaying installation messages)

!pip install -q hvplot

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/3.2 MB ? eta -:--:--
━╺━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.1/3.2 MB 2.5 MB/s eta 0:00:02
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╺━━━━━━━ 2.6/3.2 MB 37.3 MB/s eta 0:00:01
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.2/3.2 MB 31.9 MB/s eta 0:00:00

Import necessary libraries

import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
import hvplot.pandas

"(function(root) {\n function now() {\n return new Date();\n }\n\n var force = true;\n var py\_version = '3.2.2'.replace('rc', '-rc.').replace('.dev', '-dev.');\n var is\_dev = py\_version.indexOf(\"+\") !== -1 || py\_version.indexOf(\"-\") !== -1;\n var reloading = false;\n var Bokeh = root.Bokeh;\n var bokeh\_loaded = Bokeh != null && (Bokeh.version === py\_version || (Bokeh.versions !== undefined && Bokeh.versions.has(py\_version)));\n\n if (typeof (root.\_bokeh\_timeout) === \"undefined\" || force) {\n root.\_bokeh\_timeout = Date.now() + 5000;\n root.\_bokeh\_failed\_load = false;\n }\n\n function run\_callbacks() {\n try {\n root.\_bokeh\_onload\_callbacks.forEach(function(callback) {\n if (callback != null)\n callback();\n });\n } finally {\n delete root.\_bokeh\_onload\_callbacks;\n }\n console.debug(\"Bokeh: all callbacks have finished\");\n }\n\n function load\_libs(css\_urls, js\_urls, js\_modules, js\_exports, callback) {\n if (css\_urls == null) css\_urls = [];\n if (js\_urls == null) js\_urls = [];\n if (js\_modules == null) js\_modules = [];\n if (js\_exports == null) js\_exports = {};\n\n root.\_bokeh\_onload\_callbacks.push(callback);\n\n if (root.\_bokeh\_is\_loading > 0) {\n console.debug(\"Bokeh: BokehJS is being loaded, scheduling callback at\", now());\n return null;\n }\n if (js\_urls.length === 0 && js\_modules.length === 0 && Object.keys(js\_exports).length === 0) {\n run\_callbacks();\n return null;\n }\n if (!reloading) {\n console.debug(\"Bokeh: BokehJS not loaded, scheduling load and callback at\", now());\n }\n\n function on\_load() {\n root.\_bokeh\_is\_loading--;\n if (root.\_bokeh\_is\_loading === 0) {\n console.debug(\"Bokeh: all BokehJS libraries/stylesheets loaded\");\n run\_callbacks()\n }\n }\n window.\_bokeh\_on\_load = on\_load\n\n function on\_error() {\n console.error(\"failed to load \" + url);\n }\n\n var skip = [];\n if (window.requirejs) {\n window.requirejs.config({'packages': {}, 'paths': {'jspanel': 'https://cdn.jsdelivr.net/npm/jspanel4@4.12.0/dist/jspanel', 'jspanel-modal': 'https://cdn.jsdelivr.net/npm/jspanel4@4.12.0/dist/extensions/modal/jspanel.modal', 'jspanel-tooltip': 'https://cdn.jsdelivr.net/npm/jspanel4@4.12.0/dist/extensions/tooltip/jspanel.tooltip', 'jspanel-hint': 'https://cdn.jsdelivr.net/npm/jspanel4@4.12.0/dist/extensions/hint/jspanel.hint', 'jspanel-layout': 'https://cdn.jsdelivr.net/npm/jspanel4@4.12.0/dist/extensions/layout/jspanel.layout', 'jspanel-contextmenu': 'https://cdn.jsdelivr.net/npm/jspanel4@4.12.0/dist/extensions/contextmenu/jspanel.contextmenu', 'jspanel-dock': 'https://cdn.jsdelivr.net/npm/jspanel4@4.12.0/dist/extensions/dock/jspanel.dock', 'gridstack': 'https://cdn.jsdelivr.net/npm/gridstack@7.2.3/dist/gridstack-all', 'notyf': 'https://cdn.jsdelivr.net/npm/notyf@3/notyf.min'}, 'shim': {'jspanel': {'exports': 'jsPanel'}, 'gridstack': {'exports': 'GridStack'}}});\n require([\"jspanel\"], function(jsPanel) {\n\twindow.jsPanel = jsPanel\n\ton\_load()\n })\n require([\"jspanel-modal\"], function() {\n\ton\_load()\n })\n require([\"jspanel-tooltip\"], function() {\n\ton\_load()\n })\n require([\"jspanel-hint\"], function() {\n\ton\_load()\n })\n require([\"jspanel-layout\"], function() {\n\ton\_load()\n })\n require([\"jspanel-contextmenu\"], function() {\n\ton\_load()\n })\n require([\"jspanel-dock\"], function() {\n\ton\_load()\n })\n require([\"gridstack\"], function(GridStack) {\n\twindow.GridStack = GridStack\n\ton\_load()\n })\n require([\"notyf\"], function() {\n\ton\_load()\n })\n root.\_bokeh\_is\_loading = css\_urls.length + 9;\n } else {\n root.\_bokeh\_is\_loading = css\_urls.length + js\_urls.length + js\_modules.length + Object.keys(js\_exports).length;\n }\n\n var existing\_stylesheets = []\n var links = document.getElementsByTagName('link')\n for (var i = 0; i < links.length; i++) {\n var link = links[i]\n if (link.href != null) {\n\texisting\_stylesheets.push(link.href)\n }\n }\n for (var i = 0; i < css\_urls.length; i++) {\n var url = css\_urls[i];\n if (existing\_stylesheets.indexOf(url) !== -1) {\n\ton\_load()\n\tcontinue;\n }\n const element = document.createElement(\"link\");\n element.onload = on\_load;\n element.onerror = on\_error;\n element.rel = \"stylesheet\";\n element.type = \"text/css\";\n element.href = url;\n console.debug(\"Bokeh: injecting link tag for BokehJS stylesheet: \", url);\n document.body.appendChild(element);\n } if (((window['jsPanel'] !== undefined) && (!(window['jsPanel'] instanceof HTMLElement))) || window.requirejs) {\n var urls = ['https://cdn.holoviz.org/panel/1.2.3/dist/bundled/floatpanel/jspanel4@4.12.0/dist/jspanel.js', 'https://cdn.holoviz.org/panel/1.2.3/dist/bundled/floatpanel/jspanel4@4.12.0/dist/extensions/modal/jspanel.modal.js', 'https://cdn.holoviz.org/panel/1.2.3/dist/bundled/floatpanel/jspanel4@4.12.0/dist/extensions/tooltip/jspanel.tooltip.js', 'https://cdn.holoviz.org/panel/1.2.3/dist/bundled/floatpanel/jspanel4@4.12.0/dist/extensions/hint/jspanel.hint.js', 'https://cdn.holoviz.org/panel/1.2.3/dist/bundled/floatpanel/jspanel4@4.12.0/dist/extensions/layout/jspanel.layout.js', 'https://cdn.holoviz.org/panel/1.2.3/dist/bundled/floatpanel/jspanel4@4.12.0/dist/extensions/contextmenu/jspanel.contextmenu.js', 'https://cdn.holoviz.org/panel/1.2.3/dist/bundled/floatpanel/jspanel4@4.12.0/dist/extensions/dock/jspanel.dock.js'];\n for (var i = 0; i < urls.length; i++) {\n skip.push(urls[i])\n }\n } if (((window['GridStack'] !== undefined) && (!(window['GridStack'] instanceof HTMLElement))) || window.requirejs) {\n var urls = ['https://cdn.holoviz.org/panel/1.2.3/dist/bundled/gridstack/gridstack@7.2.3/dist/gridstack-all.js'];\n for (var i = 0; i < urls.length; i++) {\n skip.push(urls[i])\n }\n } if (((window['Notyf'] !== undefined) && (!(window['Notyf'] instanceof HTMLElement))) || window.requirejs) {\n var urls = ['https://cdn.holoviz.org/panel/1.2.3/dist/bundled/notificationarea/notyf@3/notyf.min.js'];\n for (var i = 0; i < urls.length; i++) {\n skip.push(urls[i])\n }\n } var existing\_scripts = []\n var scripts = document.getElementsByTagName('script')\n for (var i = 0; i < scripts.length; i++) {\n var script = scripts[i]\n if (script.src != null) {\n\texisting\_scripts.push(script.src)\n }\n }\n for (var i = 0; i < js\_urls.length; i++) {\n var url = js\_urls[i];\n if (skip.indexOf(url) !== -1 || existing\_scripts.indexOf(url) !== -1) {\n\tif (!window.requirejs) {\n\t on\_load();\n\t}\n\tcontinue;\n }\n var element = document.createElement('script');\n element.onload = on\_load;\n element.onerror = on\_error;\n element.async = false;\n element.src = url;\n console.debug(\"Bokeh: injecting script tag for BokehJS library: \", url);\n document.head.appendChild(element);\n }\n for (var i = 0; i < js\_modules.length; i++) {\n var url = js\_modules[i];\n if (skip.indexOf(url) !== -1 || existing\_scripts.indexOf(url) !== -1) {\n\tif (!window.requirejs) {\n\t on\_load();\n\t}\n\tcontinue;\n }\n var element = document.createElement('script');\n element.onload = on\_load;\n element.onerror = on\_error;\n element.async = false;\n element.src = url;\n element.type = \"module\";\n console.debug(\"Bokeh: injecting script tag for BokehJS library: \", url);\n document.head.appendChild(element);\n }\n for (const name in js\_exports) {\n var url = js\_exports[name];\n if (skip.indexOf(url) >= 0 || root[name] != null) {\n\tif (!window.requirejs) {\n\t on\_load();\n\t}\n\tcontinue;\n }\n var element = document.createElement('script');\n element.onerror = on\_error;\n element.async = false;\n element.type = \"module\";\n console.debug(\"Bokeh: injecting script tag for BokehJS library: \", url);\n element.textContent = `\n import ${name} from \"${url}\"\n window.${name} = ${name}\n window.\_bokeh\_on\_load()\n `\n document.head.appendChild(element);\n }\n if (!js\_urls.length && !js\_modules.length) {\n on\_load()\n }\n };\n\n function inject\_raw\_css(css) {\n const element = document.createElement(\"style\");\n element.appendChild(document.createTextNode(css));\n document.body.appendChild(element);\n }\n\n var js\_urls = [\"https://cdn.bokeh.org/bokeh/release/bokeh-3.2.2.min.js\", \"https://cdn.bokeh.org/bokeh/release/bokeh-gl-3.2.2.min.js\", \"https://cdn.bokeh.org/bokeh/release/bokeh-widgets-3.2.2.min.js\", \"https://cdn.bokeh.org/bokeh/release/bokeh-tables-3.2.2.min.js\", \"https://cdn.holoviz.org/panel/1.2.3/dist/panel.min.js\"];\n var js\_modules = [];\n var js\_exports = {};\n var css\_urls = [];\n var inline\_js = [ function(Bokeh) {\n Bokeh.set\_log\_level(\"info\");\n },\nfunction(Bokeh) {} // ensure no trailing comma for IE\n ];\n\n function run\_inline\_js() {\n if ((root.Bokeh !== undefined) || (force === true)) {\n for (var i = 0; i < inline\_js.length; i++) {\n inline\_js[i].call(root, root.Bokeh);\n }\n // Cache old bokeh versions\n if (Bokeh != undefined && !reloading) {\n\tvar NewBokeh = root.Bokeh;\n\tif (Bokeh.versions === undefined) {\n\t Bokeh.versions = new Map();\n\t}\n\tif (NewBokeh.version !== Bokeh.version) {\n\t Bokeh.versions.set(NewBokeh.version, NewBokeh)\n\t}\n\troot.Bokeh = Bokeh;\n }} else if (Date.now() < root.\_bokeh\_timeout) {\n setTimeout(run\_inline\_js, 100);\n } else if (!root.\_bokeh\_failed\_load) {\n console.log(\"Bokeh: BokehJS failed to load within specified timeout.\");\n root.\_bokeh\_failed\_load = true;\n }\n root.\_bokeh\_is\_initializing = false\n }\n\n function load\_or\_wait() {\n // Implement a backoff loop that tries to ensure we do not load multiple\n // versions of Bokeh and its dependencies at the same time.\n // In recent versions we use the root.\_bokeh\_is\_initializing flag\n // to determine whether there is an ongoing attempt to initialize\n // bokeh, however for backward compatibility we also try to ensure\n // that we do not start loading a newer (Panel>=1.0 and Bokeh>3) version\n // before older versions are fully initialized.\n if (root.\_bokeh\_is\_initializing && Date.now() > root.\_bokeh\_timeout) {\n root.\_bokeh\_is\_initializing = false;\n root.\_bokeh\_onload\_callbacks = undefined;\n console.log(\"Bokeh: BokehJS was loaded multiple times but one version failed to initialize.\");\n load\_or\_wait();\n } else if (root.\_bokeh\_is\_initializing || (typeof root.\_bokeh\_is\_initializing === \"undefined\" && root.\_bokeh\_onload\_callbacks !== undefined)) {\n setTimeout(load\_or\_wait, 100);\n } else {\n Bokeh = root.Bokeh;\n bokeh\_loaded = Bokeh != null && (Bokeh.version === py\_version || (Bokeh.versions !== undefined && Bokeh.versions.has(py\_version)));\n root.\_bokeh\_is\_initializing = true\n root.\_bokeh\_onload\_callbacks = []\n if (!reloading && (!bokeh\_loaded || is\_dev)) {\n\troot.Bokeh = undefined;\n }\n load\_libs(css\_urls, js\_urls, js\_modules, js\_exports, function() {\n\tconsole.debug(\"Bokeh: BokehJS plotting callback run at\", now());\n\trun\_inline\_js();\n });\n }\n }\n // Give older versions of the autoload script a head-start to ensure\n // they initialize before we start loading newer version.\n setTimeout(load\_or\_wait, 100)\n}(window));"

"\nif ((window.PyViz === undefined) || (window.PyViz instanceof HTMLElement)) {\n window.PyViz = {comms: {}, comm\_status:{}, kernels:{}, receivers: {}, plot\_index: []}\n}\n\n\n function JupyterCommManager() {\n }\n\n JupyterCommManager.prototype.register\_target = function(plot\_id, comm\_id, msg\_handler) {\n if (window.comm\_manager || ((window.Jupyter !== undefined) && (Jupyter.notebook.kernel != null))) {\n var comm\_manager = window.comm\_manager || Jupyter.notebook.kernel.comm\_manager;\n comm\_manager.register\_target(comm\_id, function(comm) {\n comm.on\_msg(msg\_handler);\n });\n } else if ((plot\_id in window.PyViz.kernels) && (window.PyViz.kernels[plot\_id])) {\n window.PyViz.kernels[plot\_id].registerCommTarget(comm\_id, function(comm) {\n comm.onMsg = msg\_handler;\n });\n } else if (typeof google != 'undefined' && google.colab.kernel != null) {\n google.colab.kernel.comms.registerTarget(comm\_id, (comm) => {\n var messages = comm.messages[Symbol.asyncIterator]();\n function processIteratorResult(result) {\n var message = result.value;\n console.log(message)\n var content = {data: message.data, comm\_id};\n var buffers = []\n for (var buffer of message.buffers || []) {\n buffers.push(new DataView(buffer))\n }\n var metadata = message.metadata || {};\n var msg = {content, buffers, metadata}\n msg\_handler(msg);\n return messages.next().then(processIteratorResult);\n }\n return messages.next().then(processIteratorResult);\n })\n }\n }\n\n JupyterCommManager.prototype.get\_client\_comm = function(plot\_id, comm\_id, msg\_handler) {\n if (comm\_id in window.PyViz.comms) {\n return window.PyViz.comms[comm\_id];\n } else if (window.comm\_manager || ((window.Jupyter !== undefined) && (Jupyter.notebook.kernel != null))) {\n var comm\_manager = window.comm\_manager || Jupyter.notebook.kernel.comm\_manager;\n var comm = comm\_manager.new\_comm(comm\_id, {}, {}, {}, comm\_id);\n if (msg\_handler) {\n comm.on\_msg(msg\_handler);\n }\n } else if ((plot\_id in window.PyViz.kernels) && (window.PyViz.kernels[plot\_id])) {\n var comm = window.PyViz.kernels[plot\_id].connectToComm(comm\_id);\n comm.open();\n if (msg\_handler) {\n comm.onMsg = msg\_handler;\n }\n } else if (typeof google != 'undefined' && google.colab.kernel != null) {\n var comm\_promise = google.colab.kernel.comms.open(comm\_id)\n comm\_promise.then((comm) => {\n window.PyViz.comms[comm\_id] = comm;\n if (msg\_handler) {\n var messages = comm.messages[Symbol.asyncIterator]();\n function processIteratorResult(result) {\n var message = result.value;\n var content = {data: message.data};\n var metadata = message.metadata || {comm\_id};\n var msg = {content, metadata}\n msg\_handler(msg);\n return messages.next().then(processIteratorResult);\n }\n return messages.next().then(processIteratorResult);\n }\n }) \n var sendClosure = (data, metadata, buffers, disposeOnDone) => {\n return comm\_promise.then((comm) => {\n comm.send(data, metadata, buffers, disposeOnDone);\n });\n };\n var comm = {\n send: sendClosure\n };\n }\n window.PyViz.comms[comm\_id] = comm;\n return comm;\n }\n window.PyViz.comm\_manager = new JupyterCommManager();\n \n\n\nvar JS\_MIME\_TYPE = 'application/javascript';\nvar HTML\_MIME\_TYPE = 'text/html';\nvar EXEC\_MIME\_TYPE = 'application/vnd.holoviews\_exec.v0+json';\nvar CLASS\_NAME = 'output';\n\n/\*\*\n \* Render data to the DOM node\n \*/\nfunction render(props, node) {\n var div = document.createElement(\"div\");\n var script = document.createElement(\"script\");\n node.appendChild(div);\n node.appendChild(script);\n}\n\n/\*\*\n \* Handle when a new output is added\n \*/\nfunction handle\_add\_output(event, handle) {\n var output\_area = handle.output\_area;\n var output = handle.output;\n if ((output.data == undefined) || (!output.data.hasOwnProperty(EXEC\_MIME\_TYPE))) {\n return\n }\n var id = output.metadata[EXEC\_MIME\_TYPE][\"id\"];\n var toinsert = output\_area.element.find(\".\" + CLASS\_NAME.split(' ')[0]);\n if (id !== undefined) {\n var nchildren = toinsert.length;\n var html\_node = toinsert[nchildren-1].children[0];\n html\_node.innerHTML = output.data[HTML\_MIME\_TYPE];\n var scripts = [];\n var nodelist = html\_node.querySelectorAll(\"script\");\n for (var i in nodelist) {\n if (nodelist.hasOwnProperty(i)) {\n scripts.push(nodelist[i])\n }\n }\n\n scripts.forEach( function (oldScript) {\n var newScript = document.createElement(\"script\");\n var attrs = [];\n var nodemap = oldScript.attributes;\n for (var j in nodemap) {\n if (nodemap.hasOwnProperty(j)) {\n attrs.push(nodemap[j])\n }\n }\n attrs.forEach(function(attr) { newScript.setAttribute(attr.name, attr.value) });\n newScript.appendChild(document.createTextNode(oldScript.innerHTML));\n oldScript.parentNode.replaceChild(newScript, oldScript);\n });\n if (JS\_MIME\_TYPE in output.data) {\n toinsert[nchildren-1].children[1].textContent = output.data[JS\_MIME\_TYPE];\n }\n output\_area.\_hv\_plot\_id = id;\n if ((window.Bokeh !== undefined) && (id in Bokeh.index)) {\n window.PyViz.plot\_index[id] = Bokeh.index[id];\n } else {\n window.PyViz.plot\_index[id] = null;\n }\n } else if (output.metadata[EXEC\_MIME\_TYPE][\"server\_id\"] !== undefined) {\n var bk\_div = document.createElement(\"div\");\n bk\_div.innerHTML = output.data[HTML\_MIME\_TYPE];\n var script\_attrs = bk\_div.children[0].attributes;\n for (var i = 0; i < script\_attrs.length; i++) {\n toinsert[toinsert.length - 1].childNodes[1].setAttribute(script\_attrs[i].name, script\_attrs[i].value);\n }\n // store reference to server id on output\_area\n output\_area.\_bokeh\_server\_id = output.metadata[EXEC\_MIME\_TYPE][\"server\_id\"];\n }\n}\n\n/\*\*\n \* Handle when an output is cleared or removed\n \*/\nfunction handle\_clear\_output(event, handle) {\n var id = handle.cell.output\_area.\_hv\_plot\_id;\n var server\_id = handle.cell.output\_area.\_bokeh\_server\_id;\n if (((id === undefined) || !(id in PyViz.plot\_index)) && (server\_id !== undefined)) { return; }\n var comm = window.PyViz.comm\_manager.get\_client\_comm(\"hv-extension-comm\", \"hv-extension-comm\", function () {});\n if (server\_id !== null) {\n comm.send({event\_type: 'server\_delete', 'id': server\_id});\n return;\n } else if (comm !== null) {\n comm.send({event\_type: 'delete', 'id': id});\n }\n delete PyViz.plot\_index[id];\n if ((window.Bokeh !== undefined) & (id in window.Bokeh.index)) {\n var doc = window.Bokeh.index[id].model.document\n doc.clear();\n const i = window.Bokeh.documents.indexOf(doc);\n if (i > -1) {\n window.Bokeh.documents.splice(i, 1);\n }\n }\n}\n\n/\*\*\n \* Handle kernel restart event\n \*/\nfunction handle\_kernel\_cleanup(event, handle) {\n delete PyViz.comms[\"hv-extension-comm\"];\n window.PyViz.plot\_index = {}\n}\n\n/\*\*\n \* Handle update\_display\_data messages\n \*/\nfunction handle\_update\_output(event, handle) {\n handle\_clear\_output(event, {cell: {output\_area: handle.output\_area}})\n handle\_add\_output(event, handle)\n}\n\nfunction register\_renderer(events, OutputArea) {\n function append\_mime(data, metadata, element) {\n // create a DOM node to render to\n var toinsert = this.create\_output\_subarea(\n metadata,\n CLASS\_NAME,\n EXEC\_MIME\_TYPE\n );\n this.keyboard\_manager.register\_events(toinsert);\n // Render to node\n var props = {data: data, metadata: metadata[EXEC\_MIME\_TYPE]};\n render(props, toinsert[0]);\n element.append(toinsert);\n return toinsert\n }\n\n events.on('output\_added.OutputArea', handle\_add\_output);\n events.on('output\_updated.OutputArea', handle\_update\_output);\n events.on('clear\_output.CodeCell', handle\_clear\_output);\n events.on('delete.Cell', handle\_clear\_output);\n events.on('kernel\_ready.Kernel', handle\_kernel\_cleanup);\n\n OutputArea.prototype.register\_mime\_type(EXEC\_MIME\_TYPE, append\_mime, {\n safe: true,\n index: 0\n });\n}\n\nif (window.Jupyter !== undefined) {\n try {\n var events = require('base/js/events');\n var OutputArea = require('notebook/js/outputarea').OutputArea;\n if (OutputArea.prototype.mime\_types().indexOf(EXEC\_MIME\_TYPE) == -1) {\n register\_renderer(events, OutputArea);\n }\n } catch(err) {\n }\n}\n"

Load a dataset from a CSV file named 'USA\_Housing.csv' located in the Googe Drive input folder Display the first few rows of the dataset

ds=pd.read\_csv("//content/drive/MyDrive/USA\_Housing.csv")  
ds.head()

Avg. Area Income Avg. Area House Age Avg. Area Number of Rooms \  
0 79545.458574 5.682861 7.009188   
1 79248.642455 6.002900 6.730821   
2 61287.067179 5.865890 8.512727   
3 63345.240046 7.188236 5.586729   
4 59982.197226 5.040555 7.839388   
  
 Avg. Area Number of Bedrooms Area Population Price \  
0 4.09 23086.800503 1.059034e+06   
1 3.09 40173.072174 1.505891e+06   
2 5.13 36882.159400 1.058988e+06   
3 3.26 34310.242831 1.260617e+06   
4 4.23 26354.109472 6.309435e+05   
  
 Address   
0 208 Michael Ferry Apt. 674\nLaurabury, NE 3701...   
1 188 Johnson Views Suite 079\nLake Kathleen, CA...   
2 9127 Elizabeth Stravenue\nDanieltown, WI 06482...   
3 USS Barnett\nFPO AP 44820   
4 USNS Raymond\nFPO AE 09386

# Preprocessing the data

Create a new DataFrame 'df' by dropping the 'Address' column from the 'ds' DataFrame.

Display the first few rows of the modified 'df' DataFrame

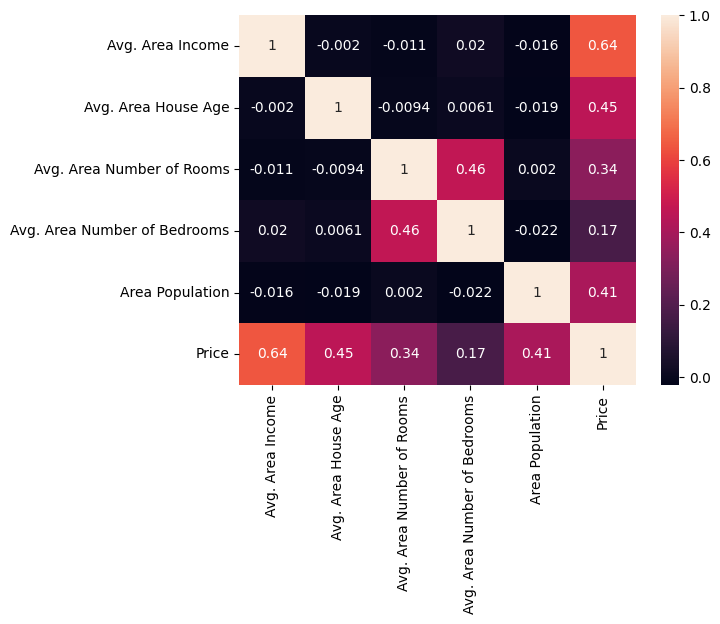
df=ds.drop(['Address'],axis=1)  
df.head()

Avg. Area Income Avg. Area House Age Avg. Area Number of Rooms \  
0 79545.458574 5.682861 7.009188   
1 79248.642455 6.002900 6.730821   
2 61287.067179 5.865890 8.512727   
3 63345.240046 7.188236 5.586729   
4 59982.197226 5.040555 7.839388   
  
 Avg. Area Number of Bedrooms Area Population Price   
0 4.09 23086.800503 1.059034e+06   
1 3.09 40173.072174 1.505891e+06   
2 5.13 36882.159400 1.058988e+06   
3 3.26 34310.242831 1.260617e+06   
4 4.23 26354.109472 6.309435e+05

Create a heatmap to visualize the correlation between columns in the 'df' DataFrame and annotate the cells with correlation values

sns.heatmap(df.corr(),annot=True)

<Axes: >



Create a new DataFrame 'x' by dropping the 'Price' column from the 'df' DataFrame

Create a Series 'y' containing the 'Price' column from the 'df' DataFrame

x=df.drop(['Price'],axis=1)  
y=df['Price']

Print the column names of the 'x' DataFrame

x.columns

Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',  
 'Avg. Area Number of Bedrooms', 'Area Population'],  
 dtype='object')

from sklearn.model\_selection import train\_test\_split  
x\_train, x\_test, y\_train, y\_test=train\_test\_split(x,y,test\_size=0.3, random\_state=42)

from sklearn import metrics  
from sklearn.model\_selection import cross\_val\_score  
from sklearn.preprocessing import StandardScaler  
from sklearn.pipeline import Pipeline  
  
def cross\_val(model):  
 pred=cross\_val\_score(model,x,y,cv=10)  
 return pred.mean()  
  
def print\_evaluate(true, predicted):  
 mae=metrics.mean\_absolute\_error(true,predicted)  
 mse=metrics.mean\_squared\_error(true,predicted)  
 rmse=np.sqrt(metrics.mean\_squared\_error(true,predicted))  
 r2\_square=metrics.r2\_score(true,predicted)  
 print('MAE: ',mae)  
 print('MSE: ',mse)  
 print('RMSE: ',rmse)  
 print('R2 Square',r2\_square)  
 print('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  
  
def evaluate(true, predicted):  
 mse =metrics.mean\_squared\_error(true, predicted)  
 mae =metrics.mean\_absolute\_error(true, predicted)  
 rmse= np.sqrt(metrics.mean\_squared\_error(true,predicted))  
 r2=square\_metrics.r2\_score (true, predicted)  
 return mae, mse, rmse, r2\_square  
  
pipeline=Pipeline([('std\_scalar',StandardScaler())])  
x\_train=pipeline.fit\_transform(x\_train)  
x\_test=pipeline.transform(x\_test)

#using LinearRegression  
from sklearn.linear\_model import LinearRegression  
lin\_reg=LinearRegression()  
lin\_reg.fit(x\_train,y\_train)  
  
test\_pred=lin\_reg.predict(x\_test)  
train\_pred=lin\_reg.predict(x\_train)  
  
print('Test set evaluation:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  
print\_evaluate(y\_test,test\_pred)  
print('Train set evaluation:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  
print\_evaluate(y\_train,train\_pred)

Test set evaluation:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
MAE: 81135.56609336878  
MSE: 10068422551.40088  
RMSE: 100341.52954485436  
R2 Square 0.9146818498754016  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Train set evaluation:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
MAE: 81480.49973174892  
MSE: 10287043161.197224  
RMSE: 101425.06180031257  
R2 Square 0.9192986579075526  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#using RandomForestRegressor  
from sklearn.ensemble import RandomForestRegressor  
rf\_reg=RandomForestRegressor(n\_estimators=1000)  
rf\_reg.fit(x\_train,y\_train)  
  
test\_pred=rf\_reg.predict(x\_test)  
train\_pred=rf\_reg.predict(x\_train)  
  
print('Test set evaluation:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  
print\_evaluate(y\_test,test\_pred)  
print('Train set evaluation:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  
print\_evaluate(y\_train,train\_pred)

Test set evaluation:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
MAE: 94148.20645204713  
MSE: 14139722963.736708  
RMSE: 118910.5670818902  
R2 Square 0.8801823224659487  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Train set evaluation:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
MAE: 35266.36426066166  
MSE: 1985024479.8320453  
RMSE: 44553.613544044274  
R2 Square 0.9844275816579577  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_