Erin Howland DSC640 wk 10 13 Aug 2022

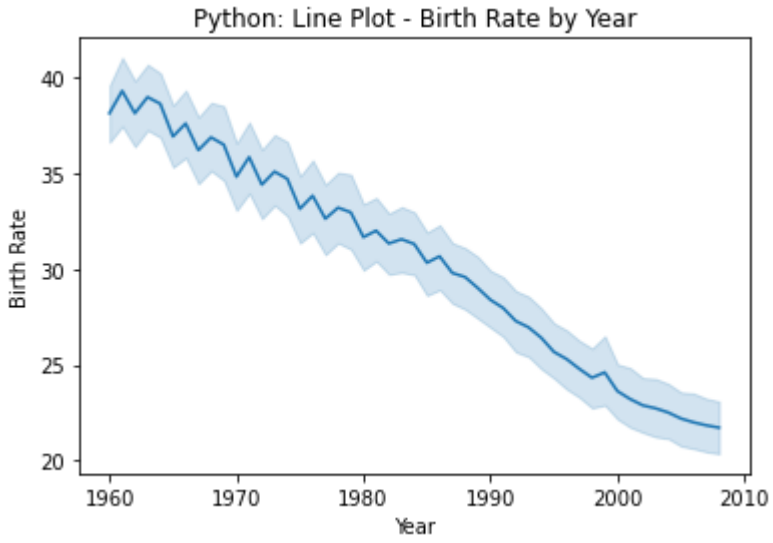
In [1]:
```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from matplotlib.ticker import FuncFormatter
```

In [2]:
```
# load data
df_br_summary = pd.read_csv(r'C:\GitHub\DSC640\DSC640\birth-rates-
yearly.csv')
df_br_summary.head()
```

Out[2]:

	year	rate
0	1960	36.400
1	1961	35.179
2	1962	33.863
3	1963	32.459
4	1964	30.994

In [3]:
```
# my choice: line chart with mean (line) and confidence interval (band)
sns.lineplot(data = df_br_summary, x = 'year', y = 'rate')
plt.title('Python: Line Plot - Birth Rate by Year')
plt.xlabel('Year')
plt.ylabel('Birth Rate')
plt.show()
```



In [4]:

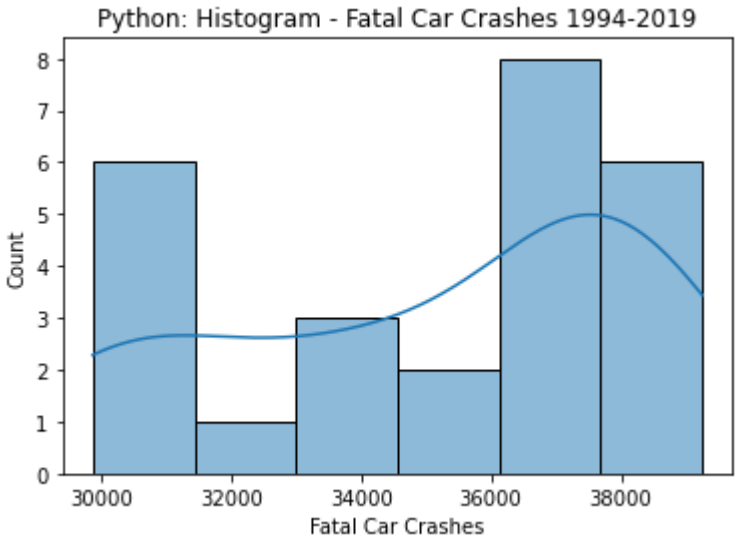
```
# load data
df_crash = pd.read_excel(r'C:\GitHub\DSC640\DSC640\FatalCarCrashes1994-2019.xlsx')
df_crash.head()
```

Out[4]:

	Year	Total
0	1994	36254
1	1995	37241
2	1996	37494
3	1997	37324
4	1998	37107

In [5]:

```
# histogram
sns.histplot(data = df_crash, x = 'Total', kde = True)
plt.title('Python: Histogram - Fatal Car Crashes 1994-2019')
plt.xlabel('Fatal Car Crashes')
plt.ylabel('Count')
plt.show()
```



In [6]:

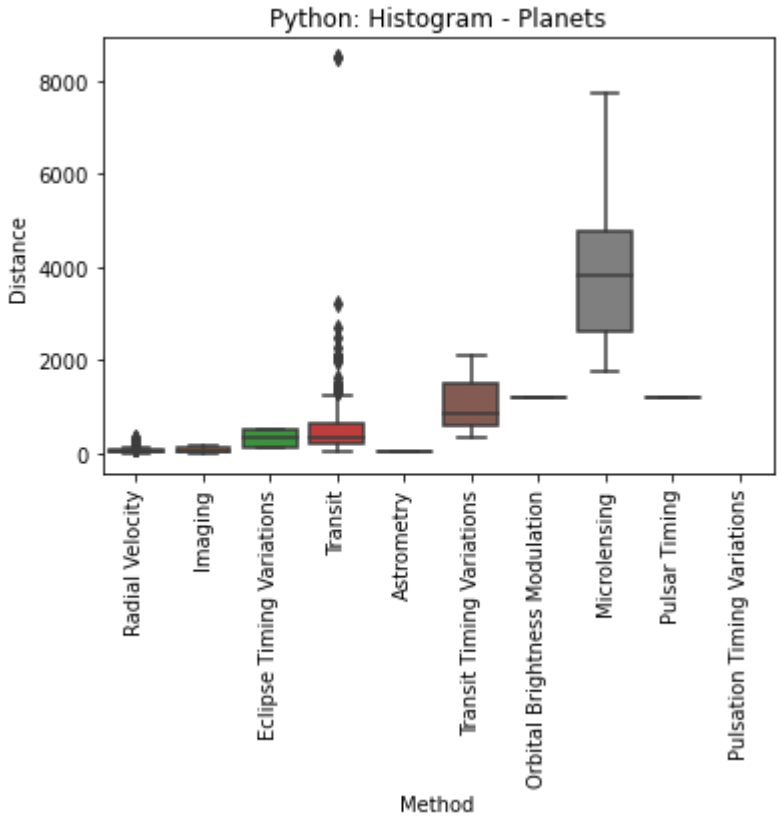
```
# load data
planets = sns.load_dataset("planets")
planets.head()
```

Out[6]:

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300	7.10	77.40	2006
1	Radial Velocity	1	874.774	2.21	56.95	2008
2	Radial Velocity	1	763.000	2.60	19.84	2011
3	Radial Velocity	1	326.030	19.40	110.62	2007
4	Radial Velocity	1	516.220	10.50	119.47	2009

In [7]:

```
# box and whiskers
sns.boxplot(x = 'method', y = 'distance', data = planets)
plt.title('Python: Histogram - Planets')
plt.xlabel('Method')
plt.ylabel('Distance')
plt.xticks(rotation=90)
plt.show()
```



In [8]:

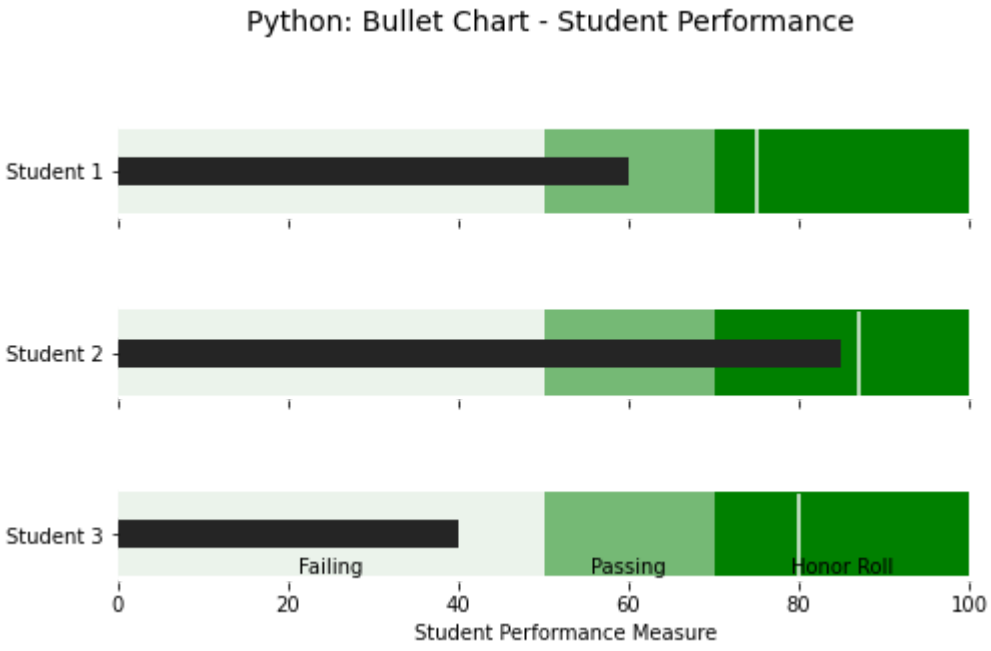
```
# bullet

# create data
data = [('Student 1', 60, 75, 100),
        ('Student 2', 85, 87, 93),
        ('Student 3', 40, 80, 75)
       ]

# fn source: https://pbpython.com/bullet-graph.html
def bulletgraph(data=None, limits=None, labels=None, axis_label=None,
title=None,
                size=(5, 3), palette=None, formatter=None,
target_color="gray",
                bar_color="black", label_color="gray"):
    """ Build out a bullet graph image
        Args:
            data = List of labels, measures and targets
            limits = list of range valules
            labels = list of descriptions of the limit ranges
            axis_label = string describing x axis
            title = string title of plot
```

```
            size = tuple for plot size
            palette = a seaborn palette
            formatter = matplotlib formatter object for x axis
            target_color = color string for the target line
            bar_color = color string for the small bar
            label_color = color string for the limit label text
        Returns:
            a matplotlib figure
    """
    # Determine the max value for adjusting the bar height
    # Dividing by 10 seems to work pretty well
    h = limits[-1] / 10

    # Use the green palette as a sensible default
    if palette is None:
        palette = sns.light_palette("green", len(limits), reverse=False)

    # Must be able to handle one or many data sets via multiple subplots
    if len(data) == 1:
        fig, ax = plt.subplots(figsize=size, sharex=True)
    else:
        fig, axarr = plt.subplots(len(data), figsize=size, sharex=True)

    # Add each bullet graph bar to a subplot
    for idx, item in enumerate(data):

        # Get the axis from the array of axes returned when the plot is
created
        if len(data) > 1:
            ax = axarr[idx]

        # Formatting to get rid of extra marking clutter
        ax.set_aspect('equal')
        ax.set_yticklabels([item[0]])
        ax.set_yticks([1])
        ax.spines['bottom'].set_visible(False)
        ax.spines['top'].set_visible(False)
        ax.spines['right'].set_visible(False)
        ax.spines['left'].set_visible(False)
```

```
        prev_limit = 0
        for idx2, lim in enumerate(limits):
            # Draw the bar
            ax.barh([1], lim - prev_limit, left=prev_limit, height=h,
                    color=palette[idx2])
            prev_limit = lim
        rects = ax.patches
        # The last item in the list is the value we're measuring
        # Draw the value we're measuring
        ax.barh([1], item[1], height=(h / 3), color=bar_color)

        # Need the ymin and max in order to make sure the target marker
        # fits
        ymin, ymax = ax.get_ylim()
        ax.vlines(
            item[2], ymin * .9, ymax * .9, linewidth=1.5,
color=target_color)

    # Now make some labels
    if labels is not None:
        for rect, label in zip(rects, labels):
            height = rect.get_height()
            ax.text(
                rect.get_x() + rect.get_width() / 2,
                -height * .4,
                label,
                ha='center',
                va='bottom',
                color=label_color)
    if formatter:
        ax.xaxis.set_major_formatter(formatter)
    if axis_label:
        ax.set_xlabel(axis_label)
    if title:
        fig.suptitle(title, fontsize=14)
    fig.subplots_adjust(hspace=0)
```

```
# create chart
bulletgraph(data,
            limits = [50, 70, 100],
            labels = ['Failing', 'Passing', 'Honor Roll'],
            size = (8,5),
            axis_label = 'Student Performance Measure',
            label_color = 'black',
            bar_color = '#252525',
            target_color = '#f7f7f7',
            title = 'Python: Bullet Chart - Student Performance'
           )
```

```
<ipython-input-8-2a8242e98500>:52: UserWarning: FixedFormatter should only be used toget
her with FixedLocator
  ax.set_yticklabels([item[0]])
```



Howland_DSC640_wk10

Howland_E

2022-08-13

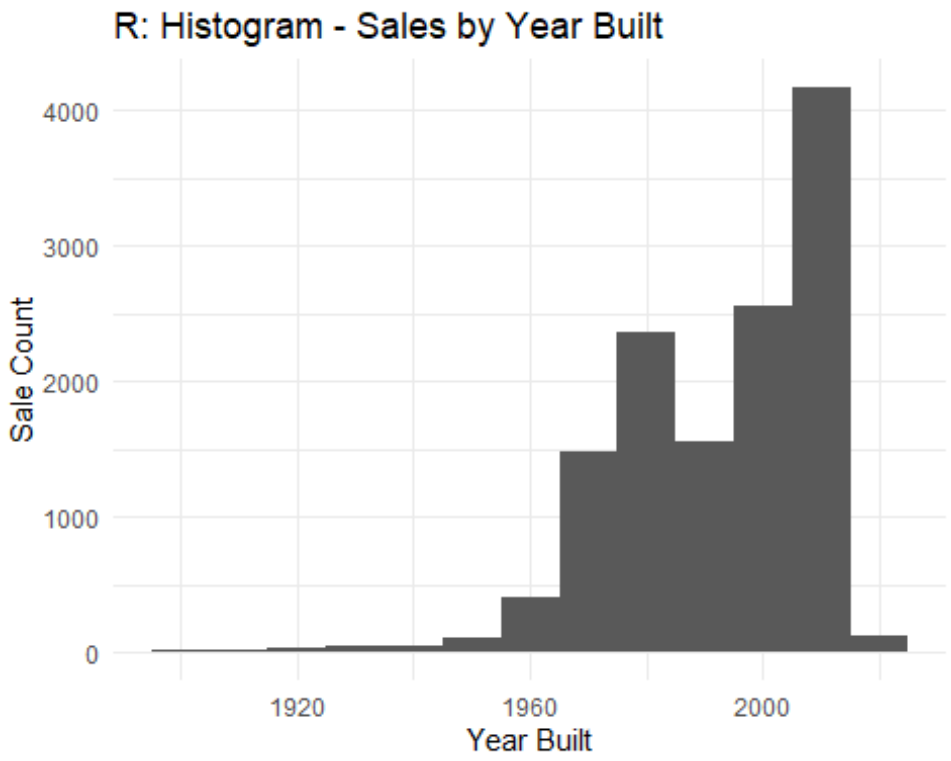
R Markdown

```
# Load the ggplot2 package
library(ggplot2)
theme_set(theme_minimal())

# Set the working directory to the root of your DSC 520 directory
# using data set from prior course
setwd("C:/GitHub/DSC520")

# Load the from American Community Survey Exercise
housing <- read.csv("housing.csv")
head(housing)
```

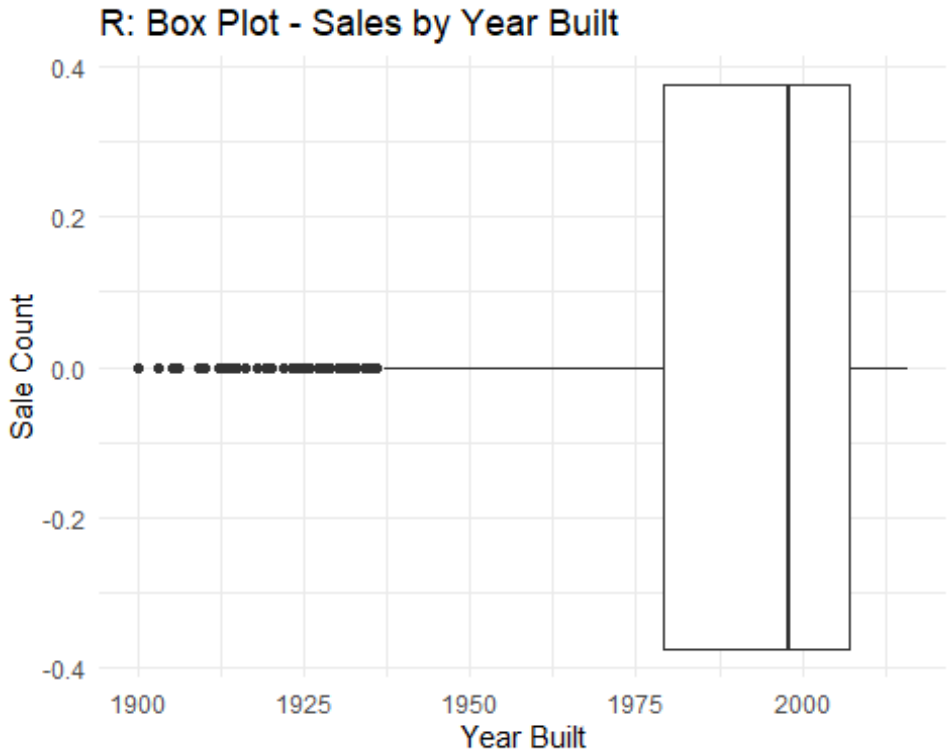
```
##   Sale.Date Sale.Price sale_reason sale_instrument sale_warning sitetype
## 1  1/3/2006     698000           1               3                    R1
## 2  1/3/2006     649990           1               3                    R1
## 3  1/3/2006     572500           1               3                    R1
## 4  1/3/2006     420000           1               3                    R1
## 5  1/3/2006     369900           1               3           15       R1
## 6  1/3/2006     184667           1              15        18 51       R1
##            addr_full  zip5 ctyname postalctyn       lon      lat
building_grade
## 1  17021 NE 113TH CT 98052 REDMOND    REDMOND -122.1124 47.70139
9
## 2  11927 178TH PL NE 98052 REDMOND    REDMOND -122.1022 47.70731
9
## 3 13315 174TH AVE NE 98052            REDMOND -122.1085 47.71986
8
## 4  3303 178TH AVE NE 98052 REDMOND    REDMOND -122.1037 47.63914
8
## 5  16126 NE 108TH CT 98052 REDMOND    REDMOND -122.1242 47.69748
7
## 6   8101 229TH DR NE 98053            REDMOND -122.0341 47.67545
7
##   square_feet_total_living bedrooms bath_full_count bath_half_count
## 1                     2810        4               2               1
## 2                     2880        4               2               0
## 3                     2770        4               1               1
## 4                     1620        3               1               0
## 5                     1440        3               1               0
## 6                     4160        4               2               1
```

```
##   bath_3qtr_count year_built year_renovated current_zoning sq_ft_lot
prop_type
## 1               0       2003              0             R4      6635
R
## 2               1       2006              0             R4      5570
R
## 3               1       1987              0             R6      8444
R
## 4               1       1968              0             R4      9600
R
## 5               1       1980              0             R6      7526
R
## 6               1       2005              0          URPSO      7280
R
##   present_use
## 1           2
## 2           2
## 3           2
## 4           2
## 5           2
## 6           2
```

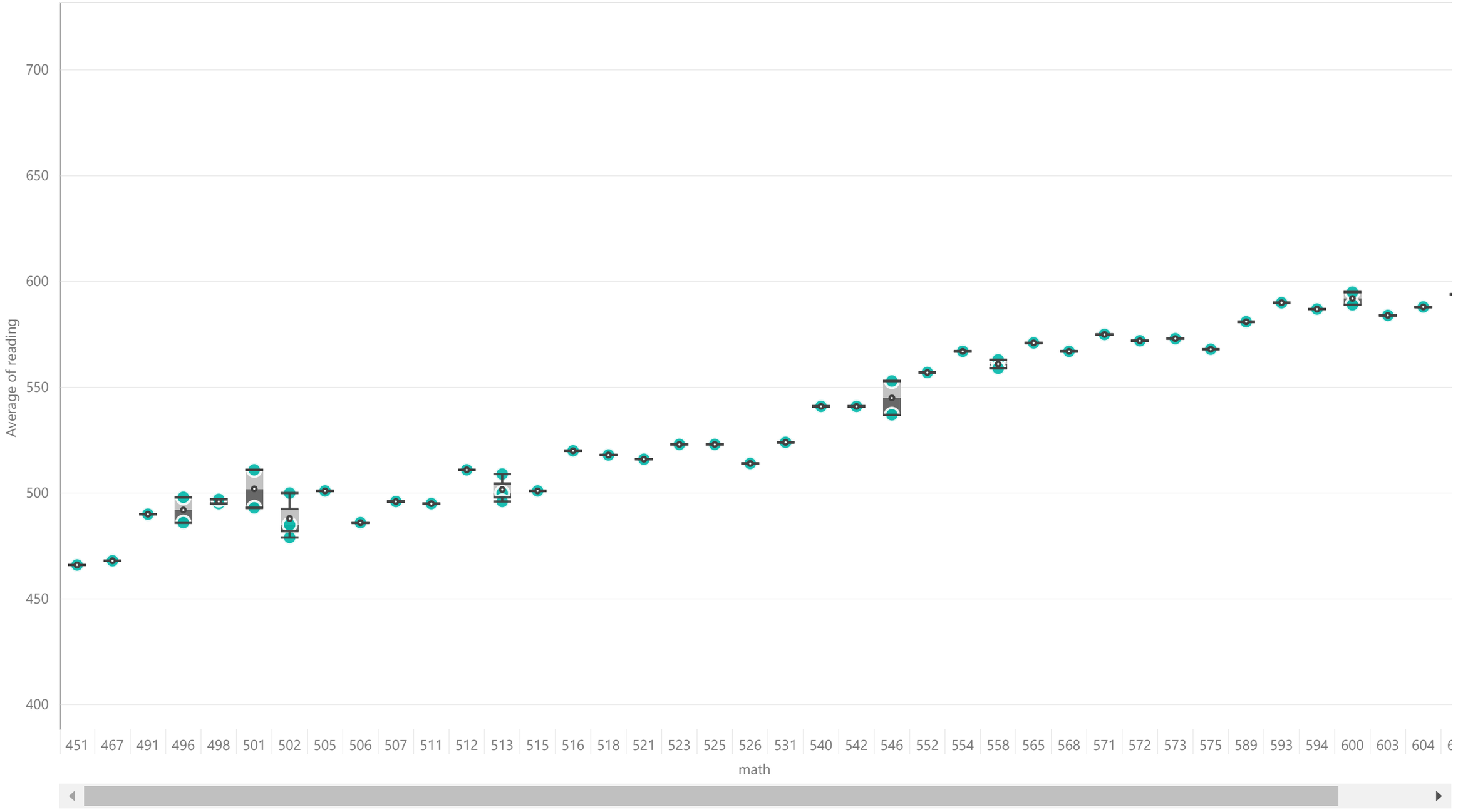
Histogram



Box Plot


R: Box Plot - Sales by Year Built
0.4
0.2
0.0
-0.2
-0.4
Sale Count
1900
1925
1950
1975
2000
Year Built

PBI: Box Plot - Average of reading by state and math



Sale Price by year_built