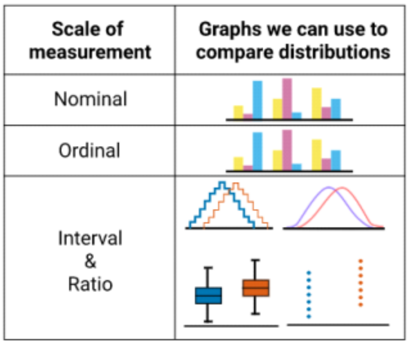
# 5.0 Comparing Frequency Distributions

June 18, 2019 9:45 PM

**Basic**

For visualizing Nominal and Ordinal Variables, use Grouped Bar Plots
For visualizing Interval & Ratio variables, use:

- Step type histograms (histogram with lines only and no colors)
- Kernel density plot (KDE)
- Box and whisker plot or box plot
- Step plot



| Scale of measurement | Graphs we can use to compare distributions |
|---|---|
| Nominal | |
| Ordinal | |
| Interval & Ratio | |

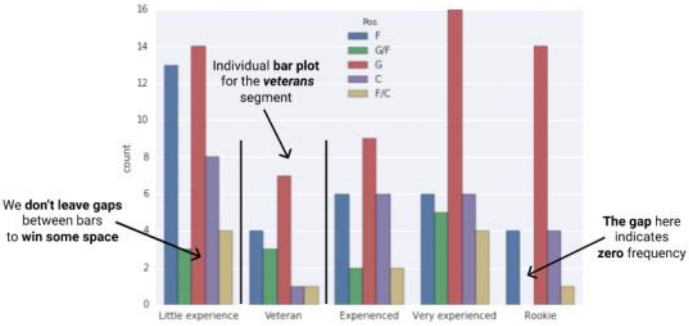
**Grouped Bar Plot**

Bar plot:

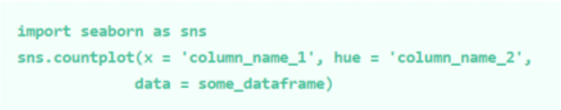
- X axis: categorical variable
- Y axis: numeric variable

In grouped bar plot, we have the same thing but we have different colors of bar for each categorical variable to represent another categorical variable

Example:

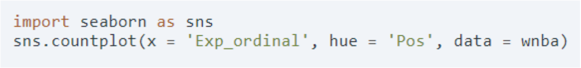


We have grouped the plots together so it is called grouped bar plot

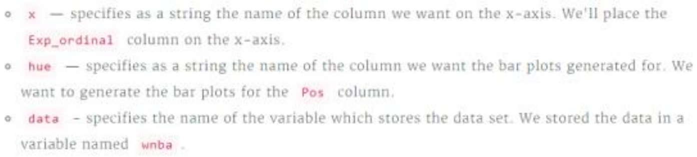


```
import seaborn as sns
sns.countplot(x = 'column_name_1', hue = 'column_name_2',
              data = some_dataframe)
```

Example:



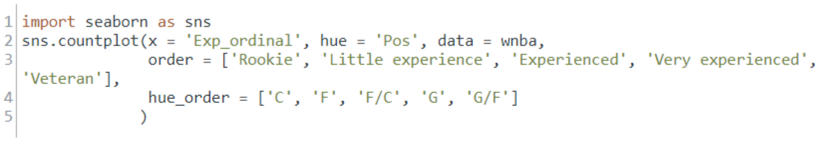
```
import seaborn as sns
sns.countplot(x = 'Exp_ordinal', hue = 'Pos', data = wnba)
```



- x — specifies as a string the name of the column we want on the x-axis. We'll place the Exp_ordinal column on the x-axis.
- hue — specifies as a string the name of the column we want the bar plots generated for. We want to generate the bar plots for the Pos column.
- data - specifies the name of the variable which stores the data set. We stored the data in a variable named wnba .

To order the categorical variable on the bottom (little experience, veteran, etc.), use the order argument and pass in the order as a list

Example



```
import seaborn as sns
sns.countplot(x = 'Exp_ordinal', hue = 'Pos', data = wnba,
              order = ['Rookie', 'Little experience', 'Experienced', 'Very experienced',
'Veteran'],
              hue_order = ['C', 'F', 'F/C', 'G', 'G/F']
             )
```

**Step-type Histogram**

When Visualizing Frequency Distribution for Variables of different types (Nominal, Ordinal, Interval, Ratio), we have two goals:

First, what type of chart do we use to visualize the frequency distribution
Second, how do we compare the frequency distribution

Key Concepts: Visualizing Frequency Distribution (Lesson 4)

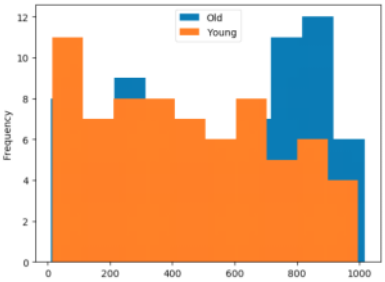
- Nominal, Ordinal: Bar Plot, Pie Chart
  - Pie chart allows to see proportions/percentages better
  - Using histogram does not make sense as the numbers don't tell me the size of difference
- Interval, Ratio: Histogram, KDE, Box Plot
- Histogram Shape
  - Histogram vs. Bar Plot: Histogram has no gap between bars ==> continuous data
  - Skewed: values bunch up to the left/right
    - Left: Sliding down toward left (skewed to the left/tailed to the left)
    - Right: Sliding down toward right (skewed to the right/tailed to the right)
  - Symmetric: Can cut the distribution in half and each half is a mirror of the other half
    - Normal: Lots in the middle, tail off to the ends
    - Evenly distributed
      - ~mirror image of each other about the middle but not quite uniform/normal distribution
    - Uniform: uniform distribution of values (~bunch of people of same height standing beside each other; we can draw a horizontal line over them)

Key Concepts: Comparing Frequency Distribution (Lesson 5)

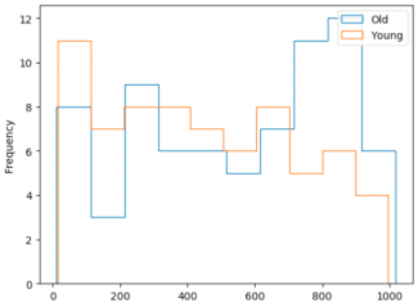
- Nominal, Ordinal: Grouped Bar Plots
  - Generate bar plot for each table and then group them together
  - Example: Experience by Position Types Frequency
    - Experience Categories in X-axis
    - Position Types Identified by Color
    - Counts identified by length of bars
    - Without grouped bar plot, Each position would generate a separate bar plot
- Interval, Ratio (See images to the left)
  - Step Type Histogram
    - Histograms with outline only superimposed on each other
    - Example: Age Frequency by Position (Each Unique Position Generates a Histogram)
  - Kernel Density Plot (KDE)
    - ~ smoothed histogram
    - Example: Age Frequency by Position (Each Unique Position Generates a KDE)
  - Strip Plot
    - Example: Position vs. height. Each position generates a strip
  - Box and Whisker Plot or Box Plot
    - The line at the far ends are telling me the max and minimum of the distribution
    - Example: Position vs. height. Each position generates a box and whisker plot
    - Shows 25% (Q1), 50% (Q2), 75% (Q3) quartile
    - IQR = Q3 - Q1
    - Outlier > 1.5xIQR + Q3 or Outlier < Q1 - 1.5xIQR

This histogram is like any other histogram but it only shows the shape of the histogram

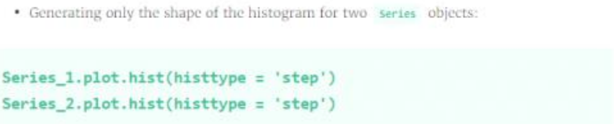
Normal histogram (two histogram superimposed)



Step-type histogram



Screen clipping taken: 2019-06-18 9:31 PM

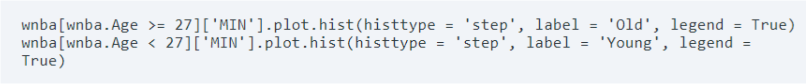


- Generating only the shape of the histogram for two `Series` objects:

```
Series_1.plot.hist(histtype = 'step')
Series_2.plot.hist(histtype = 'step')
```

Screen clipping taken: 2019-06-18 8:55 PM

Example syntax of superimposed histogram

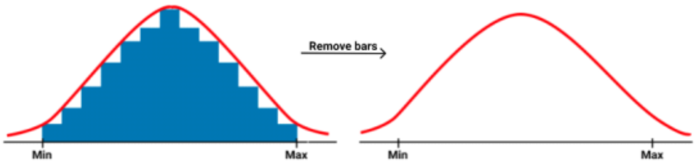


```
wnba[wnba.Age >= 27]['MIN'].plot.hist(histtype = 'step', label = 'Old', legend = True)
wnba[wnba.Age < 27]['MIN'].plot.hist(histtype = 'step', label = 'Young', legend =
True)
```

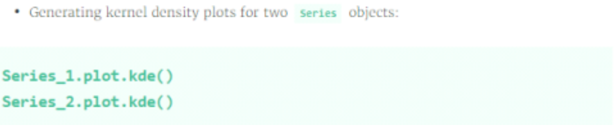
Note that, here we are not creating axis object because we have one plot. Thus plotting the two different series is putting the graphs on the same space

**Kernel Density Plot (KDE)**

Smoothed histogram



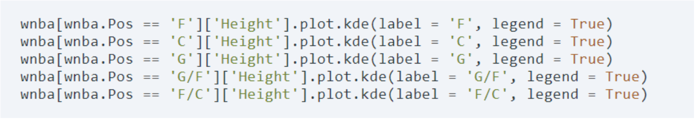
In KDE< we have probability values on the y-axis not frequencies. This will be covered later



- Generating kernel density plots for two `Series` objects:

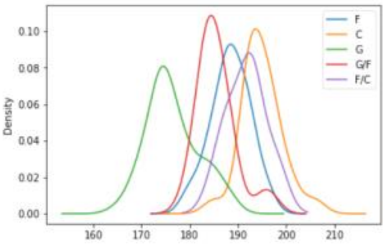
```
Series_1.plot.kde()
Series_2.plot.kde()
```

Examples



```
wnba[wnba.Pos == 'F']['Height'].plot.kde(label = 'F', legend = True)
wnba[wnba.Pos == 'C']['Height'].plot.kde(label = 'C', legend = True)
wnba[wnba.Pos == 'G']['Height'].plot.kde(label = 'G', legend = True)
wnba[wnba.Pos == 'G/F']['Height'].plot.kde(label = 'G/F', legend = True)
wnba[wnba.Pos == 'F/C']['Height'].plot.kde(label = 'F/C', legend = True)
```

Screen clipping taken: 2019-08-08 1:28 AM

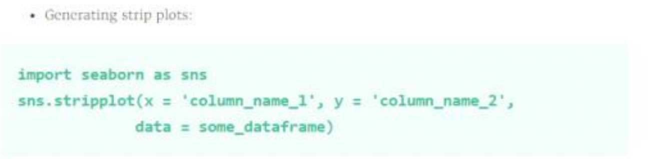


Screen clipping taken: 2019-08-08 1:28 AM

**Strip Plot**

- X axis: categorical variable
- Y axis: numerical variable

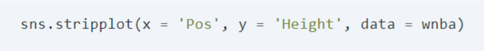
Example:



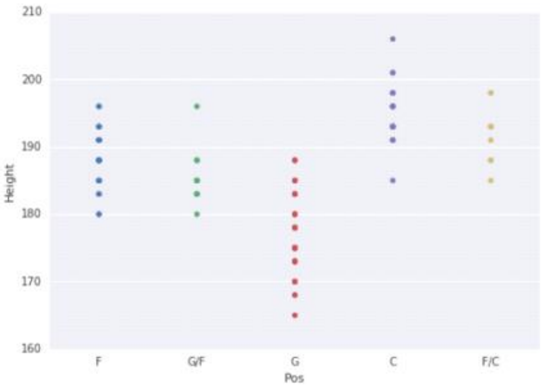
- Generating strip plots:

```
import seaborn as sns
sns.stripplot(x = 'column_name_1', y = 'column_name_2',
              data = some_dataframe)
```

Example:

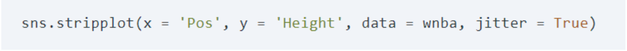


```
sns.stripplot(x = 'Pos', y = 'Height', data = wnba)
```



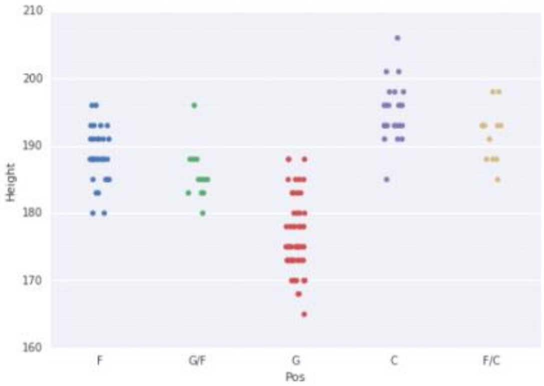
We have five "strips". This is why this is called strip plots

One point could be hiding multiple instances. To show them, use the argument "jitter = True"

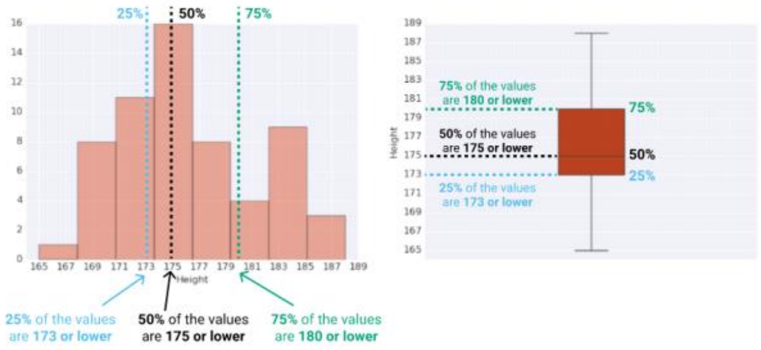


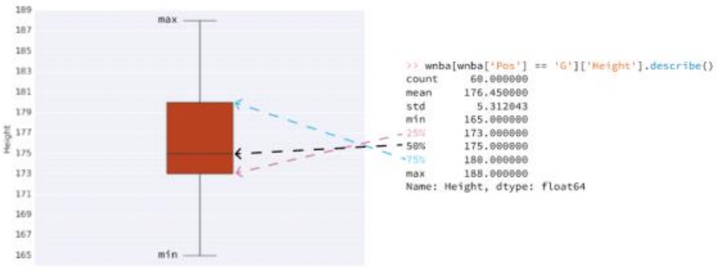
```
sns.stripplot(x = 'Pos', y = 'Height', data = wnba, jitter = True)
```

Screen clipping taken: 2019-08-08 1:29 AM



**Box and Whisker Plot / Box Plot**

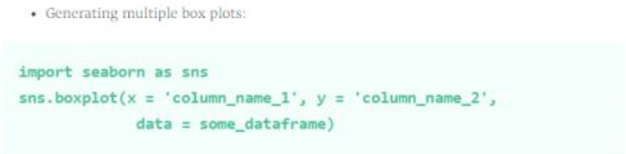




```
>> wnba[wnba['Pos'] == 'G']['Height'].describe()
count     60.000000
mean     176.450000
std        5.312043
min      165.000000
25%      173.000000
50%      175.000000
75%      180.000000
max      188.000000
Name: Height, dtype: float64
```

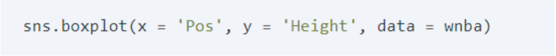
Screen clipping taken: 2019-06-18 9:18 PM

The two lines extending upwards and downwards out of the *box* in the middle look a bit like two *whiskers*, reason for which we call this plot a **box-and-whisker plot**, or, more convenient, just **box plot**.

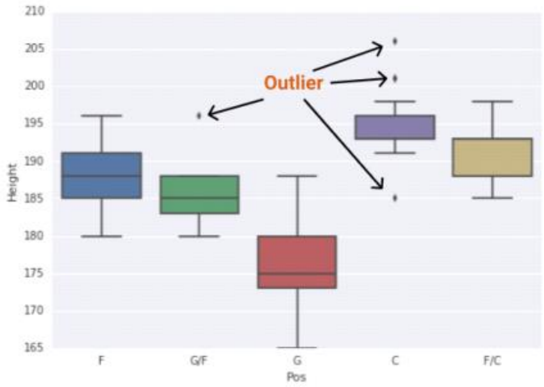


- Generating multiple box plots:

```
import seaborn as sns
sns.boxplot(x = 'column_name_1', y = 'column_name_2',
            data = some_dataframe)
```



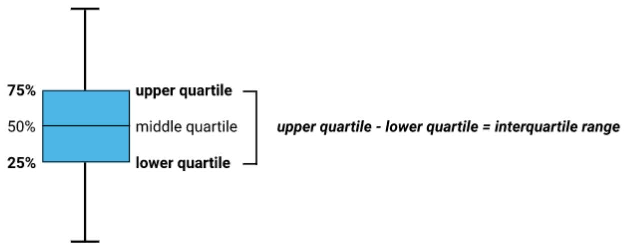
```
sns.boxplot(x = 'Pos', y = 'Height', data = wnba)
```



Outliers (the dots outside the whisker)

A value is an outlier if:

- It's larger than the upper quartile by 1.5 times the difference between the upper quartile and the lower quartile (the differ ence is also called *the interquartile range*).
- It's lower than the lower quartile by 1.5 times the difference between the upper quartile and the lower quartile (the differe nce is also called *the interquartile range*).



Screen clipping taken: 2019-06-18 9:19 PM

We could use factors other than 1.5. to change that in sns.boxplot (), use "whis = real number"

## Resources

- A seaborn tutorial on grouped bar plots, strip plots, box plots, and more.
- A seaborn tutorial on kernel density plots, histograms, and more.

From <https://app.dataquest.io/m/287/comparing-frequency-distributions/11/takeaways>