1.2.B - Descriptive Statistics

Reference

- http://stemgraphic.org/doc/intro.html#installation
- https://stackoverflow.com/questions/49703938/how-to-create-a-dot-plot-in-matplotlib-not-a-scatter-plot
- https://matplotlib.org/stable/gallery/statistics/histogram_features.html
- https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

Data

In [1]:
```
# Data from 1.2.B
data = [34, 17, 7, 10, 27, 30, 46, 23, 17, 27, \
        20, 24, 30, 32, 21, 29, 12, 37, 16, 23, \
        10, 36, 13, 27, 17, 23, 19, 13]
```

Stem and Leaf

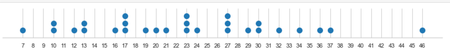
In [55]:
```
import stemgraphic

# Reference: http://stemgraphic.org/doc/modules.html#module-stemgraphic.graphic
# data - list, numpy array, time series, pandas or dask dataframe
# asc — stem sorted in ascending order, defaults to True
# scale — force a specific scale for building the plot. Defaults to None (automatic).
stemgraphic.stem_graphic(data, asc = False, scale = 10)
```

Out[55]:
```
<Figure size 540x144 with 1 Axes>,
<matplotlib.axes._axes.Axes at 0x7fcf1236f340>)
```

```
 1 | 07
11 | 10023367779
21 | 20133347779
27 | 3002467
28 | 46

≠ |4 |6 =  4.6x10 = 46.0
Key: aggr|stem|leaf
```

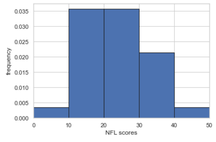
Dot plot

In [56]:
```
# Better option

import numpy as np
import matplotlib.pyplot as plt

# Preparation
values, counts = np.unique(data, return_counts=True)

# Set formatting parameters based on data
data_range = max(values)-min(values)
width = data_range/2 if data_range < 30 else 15
height = max(counts)/3 if data_range < 50 else max(counts)/4
marker_size = 10 if data_range < 50 else np.ceil(300/(data_range//10))

# Create dot plot with appropriate format
fig, ax = plt.subplots(figsize=(width, height))
for value, count in zip(values, counts):
    ax.plot([value]*count, list(range(count)), marker='o', color='tab:blue',
            ms=marker_size, linestyle='')
for spine in ['top', 'right', 'left']:
    ax.spines[spine].set_visible(False)
ax.yaxis.set_visible(False)
ax.set_ylim(-1, max(counts))
ax.set_xticks(range(min(values), max(values)+1))
ax.tick_params(axis='x', length=0, pad=10)

plt.show()
```

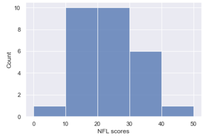


Histogram

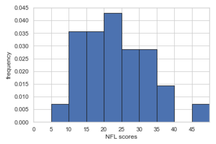
In [57]:
```
# Option 1: matplotlib
import numpy as np
import matplotlib.pyplot as plt

num_bins = np.arange(0,51,10)
fig, ax = plt.subplots()

# the histogram of the data
n, bins, patches = ax.hist(data, num_bins, density=True, ec='k')

# Set label
plt.xlabel('NFL scores')
plt.ylabel('frequency')

#Set x axis range
plt.xlim(0,50)

plt.show()
```



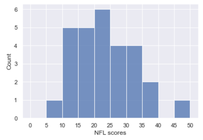
In [64]:
```
# Option 2: seaborn
import seaborn as sns
import matplotlib.pyplot as plt

# set a grey background (use sns.set_theme() if seaborn version 0.11.0 or above)
sns.set(style="darkgrid")

plt.xlabel('NFL scores')
plt.ylabel('Count')

num_bins = np.arange(0,51,10)

sns.histplot(data=data, bins=num_bins)
plt.show()
```



In [58]:
```
# Option 1: matplotlib
import numpy as np
import matplotlib.pyplot as plt

num_bins = np.arange(0,51,5)
fig, ax = plt.subplots()

# the histogram of the data
n, bins, patches = ax.hist(data, num_bins, density=True, ec="k")

# Set label
plt.xlabel('NFL scores')
plt.ylabel('freequency')

# Set x axis range
plt.xlim(0,50)

# Specify the ticket locations
# tick_locs = [5, 10, 15, 20, 50], or if you want to customize it in this way ...
tick_locs = range(0, 50, 5)
plt.xticks(tick_locs, tick_locs)

plt.show()
```



In [15]:
```
# Option 2: seaborn
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Set a grey background (use sns.set_theme() if seaborn version 0.11.0 or above)
sns.set(style="darkgrid")

# Define bins
num_bins = np.arange(0,51,5)

fig, ax = plt.subplots()
#ax.set_xlim(0,50)
ax.set_xticks(range(0,51,5))
sns.histplot(data, bins=num_bins)

plt.xlabel('NFL scores')
plt.ylabel('Count')

plt.show()
```



Describe

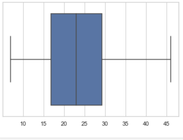
In [59]:
```
import pandas as pd

# Turn list to DataFrame
df = pd.DataFrame(data)

df.describe()
```

Out[59]:

	0
count	28.000000
mean	22.857143
std	9.387597
min	7.000000
25%	16.750000
50%	23.000000
75%	29.250000
max	46.000000

Boxplot

In [60]:
```
import seaborn as sns
sns.set_theme(style = "whitegrid")
flierprops = dict(markerfacecolor = '0.75', markersize = 5,linestyle = 'none')

# No outlier. Add an outlier by purpose to test the color.
# data = [34, 17, 7, 10, 27, 30, 46, 23, 17, 27, \
#         20, 24, 30, 32, 21, 29, 12, 37, 16, 23, \
#         10, 36, 13, 27, 17, 23, 19, 13, 100]

ax = sns.boxplot(x = data, flierprops=flierprops)
```



In [ ]:
```
```