Detecting Outliers: 1. Visualization

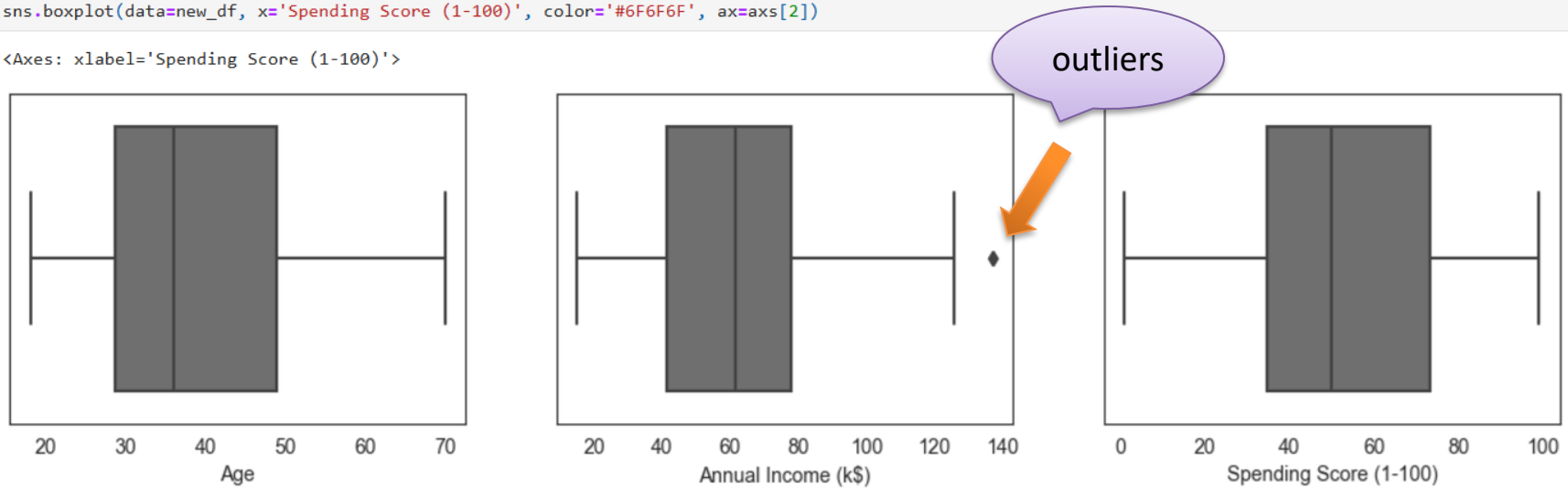
➢ Boxplot

Detecting Outliers: Visualization

In [12]:
```
sns.set_style('white')

fig, axs = plt.subplots(1, 3, figsize=(14, 3))
sns.boxplot(data=new_df, x='Age', color='#6F6F6F', ax=axs[0])
sns.boxplot(data=new_df, x='Annual Income (k$)', color='#6F6F6F', ax=axs[1])
sns.boxplot(data=new_df, x='Spending Score (1-100)', color='#6F6F6F', ax=axs[2])
```

Out[12]: <Axes: xlabel='Spending Score (1-100)'>



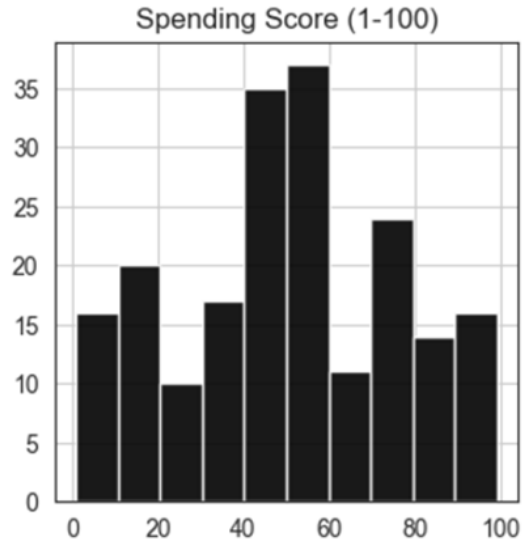
We see here that there are outliers in 'Annual Income (k$)'.

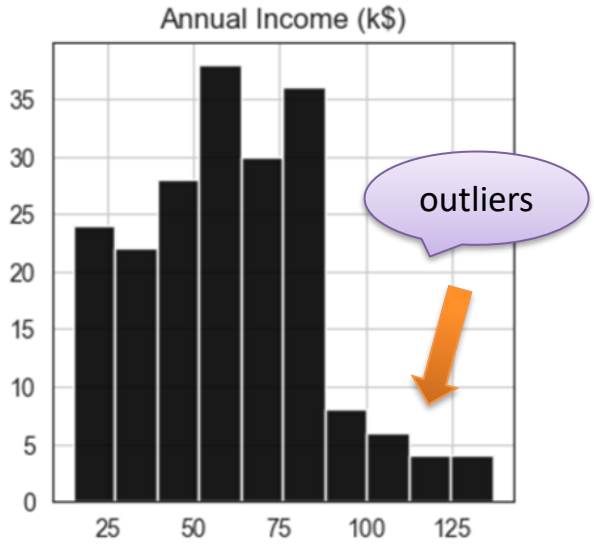
Detecting Outliers: 1. Visualization

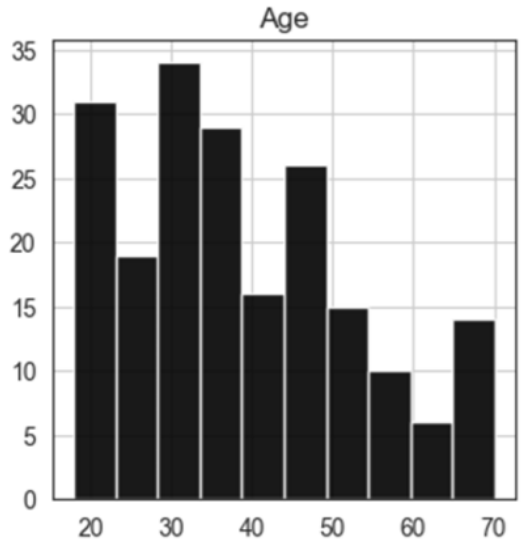
- Histogram

```
In [13]: # Distribution of numerical values
         new_df[numeric_variables].hist(figsize=(8, 8), color='k', alpha=0.9)

Out[13]: array([[<Axes: title={'center': 'Spending Score (1-100)'}>,
                 <Axes: title={'center': 'Annual Income (k$)'}>],
                [<Axes: title={'center': 'Age'}>, <Axes: >]], dtype=object)
```







Detecting Outliers: 2. Statistics

Detecting Outliers: Statistics

In [15]:
```
# Calculate the interquartile range
q25, q50, q75 = np.percentile(new_df['Annual Income (k$)'], [25,50,75])
iqr = q75 - q25

# Calculate the min/max limits to be considered an outlier
min = q25 - 1.5*(iqr)
max = q75 + 1.5*(iqr)
print(min, q25, q50, q75, max)
```

```
-13.25 41.5 61.5 78.0 132.75
```

outliers beyond max

In [16]:
```
# Identify the points
[x for x in new_df['Annual Income (k$)'] if x > max]
```

Out[16]:
```
[137, 137]
```

We see here that there are 2 outliers in 'Annual Income (k$)' with the same value of 137.

Created by: Felice Benita