

# Adding style

INTERACTIVE DATA VISUALIZATION WITH BOKEH



**George Boorman**

Core Curriculum Manager, DataCamp

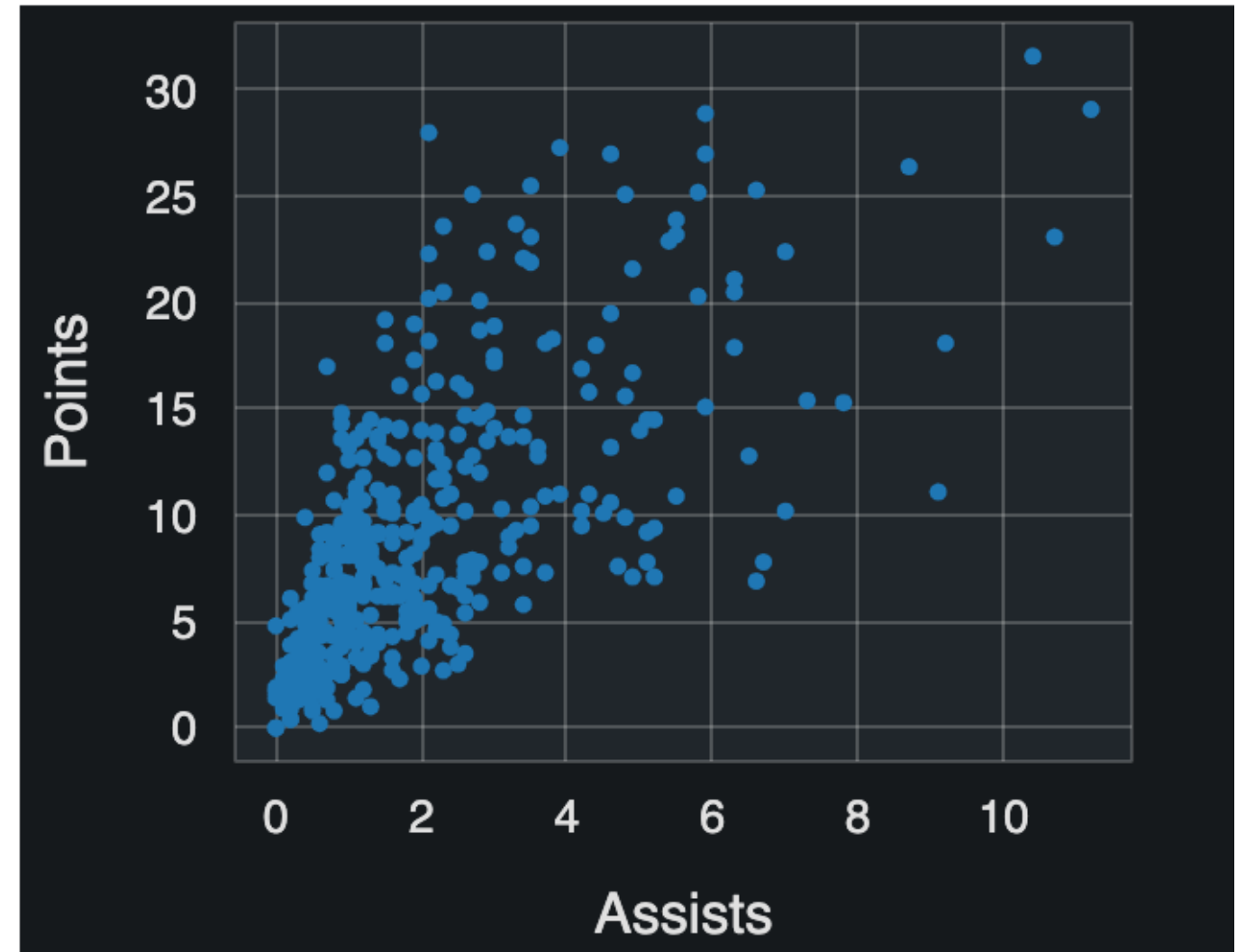
# Bokeh themes

- `caliber`
- `dark_minimal`
- `light_minimal`
- `night_sky`
- `contrast`

<sup>1</sup> [http://docs.bokeh.org/en/latest/docs/user\\_guide/styling.html#using-themes](http://docs.bokeh.org/en/latest/docs/user_guide/styling.html#using-themes)

# Using themes

```
from bokeh.io import curdoc
curdoc().theme = "dark_minimal"
fig = figure(x_axis_label="Assists",
             y_axis_label="Points")
fig.circle(x=nba["assists"],
           y=nba["points"])
output_file("dark_minimal.html")
show(fig)
```



# Subsetting data

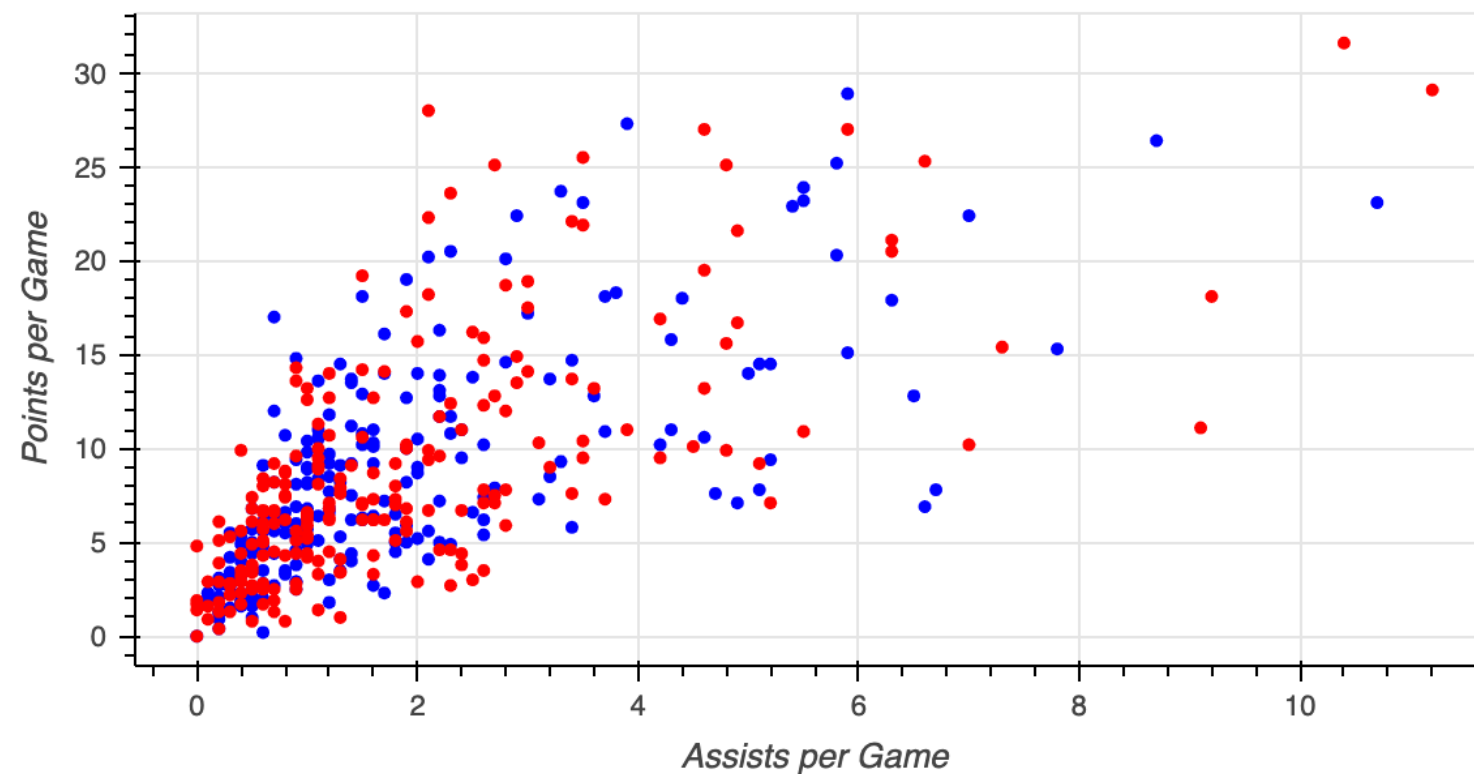
```
east = nba.loc[nba["conference"] == "East"]  
west = nba.loc[nba["conference"] == "West"]
```

```
print(nba.columns, '\n', east.shape, '\n', west.shape)
```

```
Index(['player', 'position', 'minutes', 'field_goal_perc', 'three_point_perc', 'free_throw_perc',  
      'rebounds', 'assists', 'steals', 'blocks', 'points', 'team', 'conference'],  
      dtype='object')  
(211, 13)  
(215, 13)
```

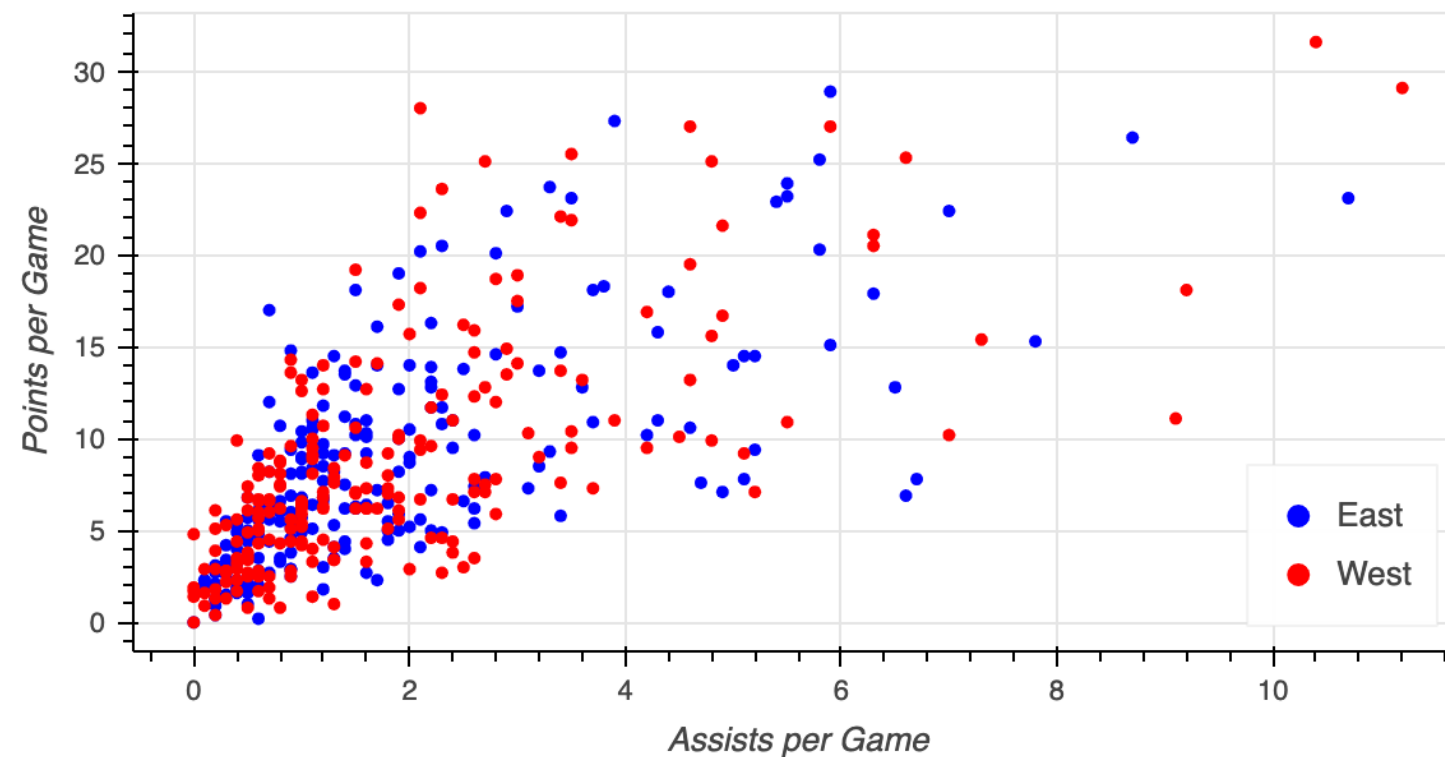
# Customizing color

```
fig = figure(x_axis_label="Assists per Game", y_axis_label="Points per Game")
fig.circle(x=east["assists"], y=east["points"], color="blue")
fig.circle(x=west["assists"], y=west["points"], color="red")
output_file(filename="east_vs_west.html")
show(fig)
```



# Adding a legend

```
fig = figure(x_axis_label="Assists per Game", y_axis_label="Points per Game")
fig.circle(x=east["assists"], y=east["points"], color="blue", legend_label="East")
fig.circle(x=west["assists"], y=west["points"], color="red", legend_label="West")
output_file(filename="east_vs_west_with_legend.html")
show(fig)
```



# Glyph types

- square



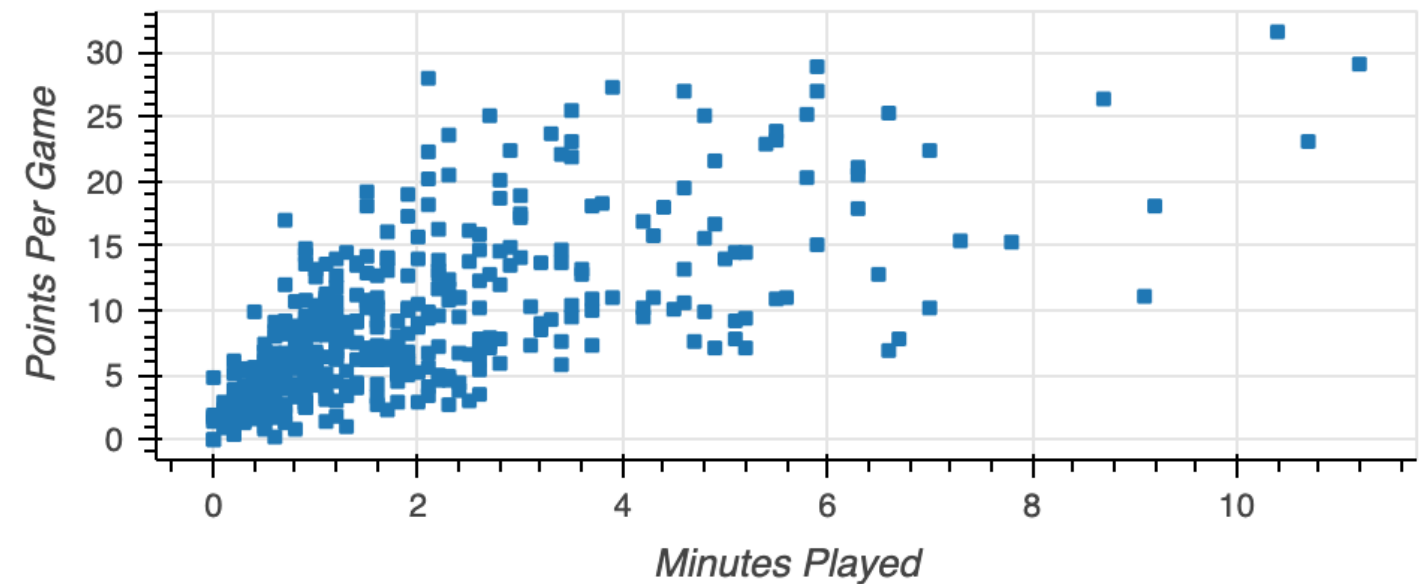
- triangle



- diamond

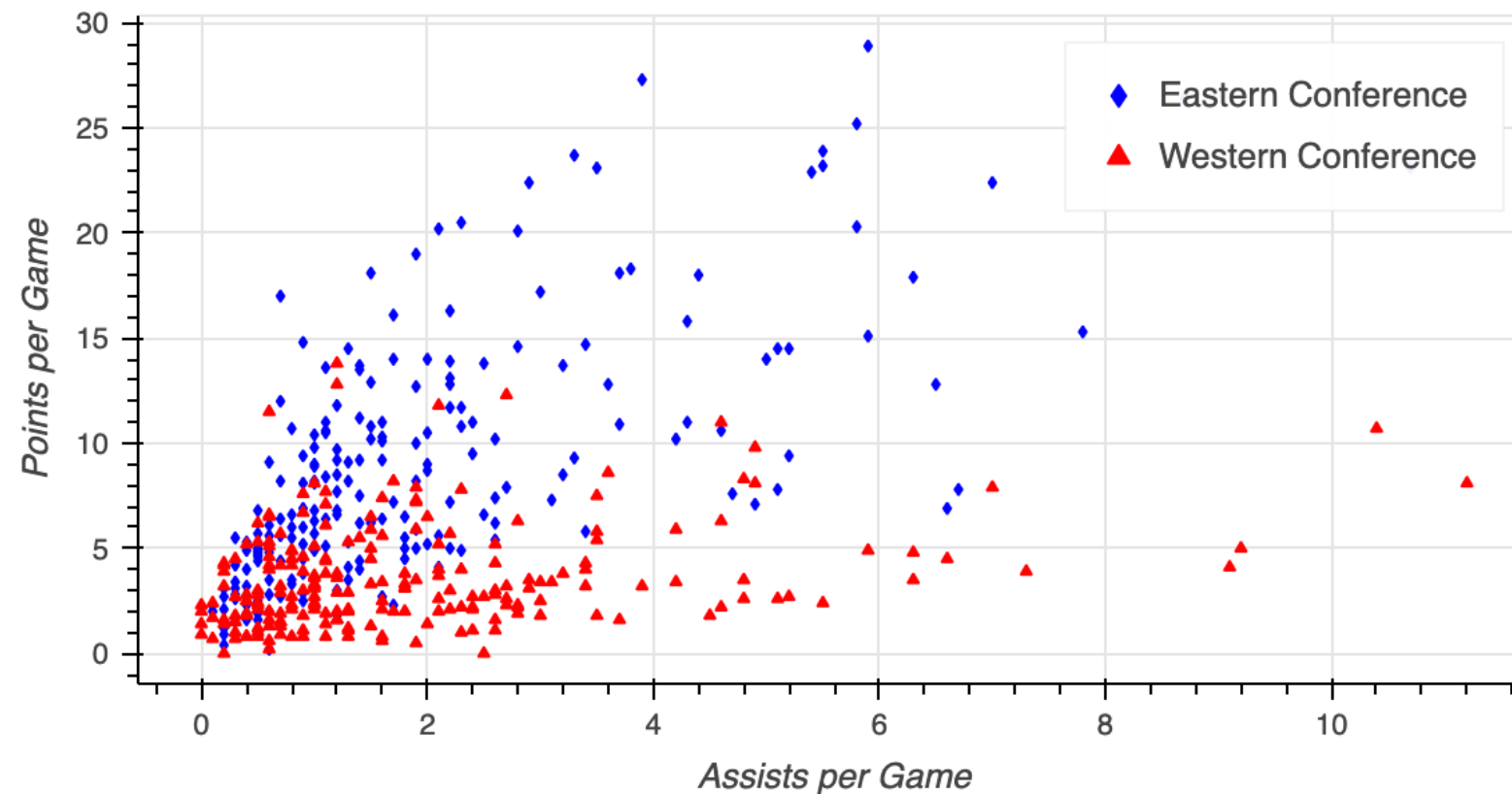


```
fig = figure(x_axis_label='Minutes Played',  
             y_axis_label='Points Per Game')  
fig.square(x=nba["minutes_played"],  
           y=nba["points_per_game"])  
output_file(filename="nba_points.html")  
show(fig)
```



# Multiple glyphs

```
fig = figure(x_axis_label="Assists per Game", y_axis_label="Points per Game")
fig.diamond(x=east["assists"], y=east["points"], color="blue", legend_label="Eastern Conference")
fig.triangle(x=west["assists"], y=west["points"], color="red", legend_label="Western Conference")
output_file(filename="multiple_glyphs.html")
show(fig)
```





# The dataset

```
print(melb.columns)
```

```
Index(['rooms', 'type', 'price', 'date', 'distance', 'bedrooms', 'bathrooms',  
      'car', 'land_area', 'building_area', 'year_built', 'council_area',  
      'region'],  
      dtype='object')
```

```
print(melb.shape)
```

```
(13580, 13)
```

# Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

# Customizing axes

INTERACTIVE DATA VISUALIZATION WITH BOKEH

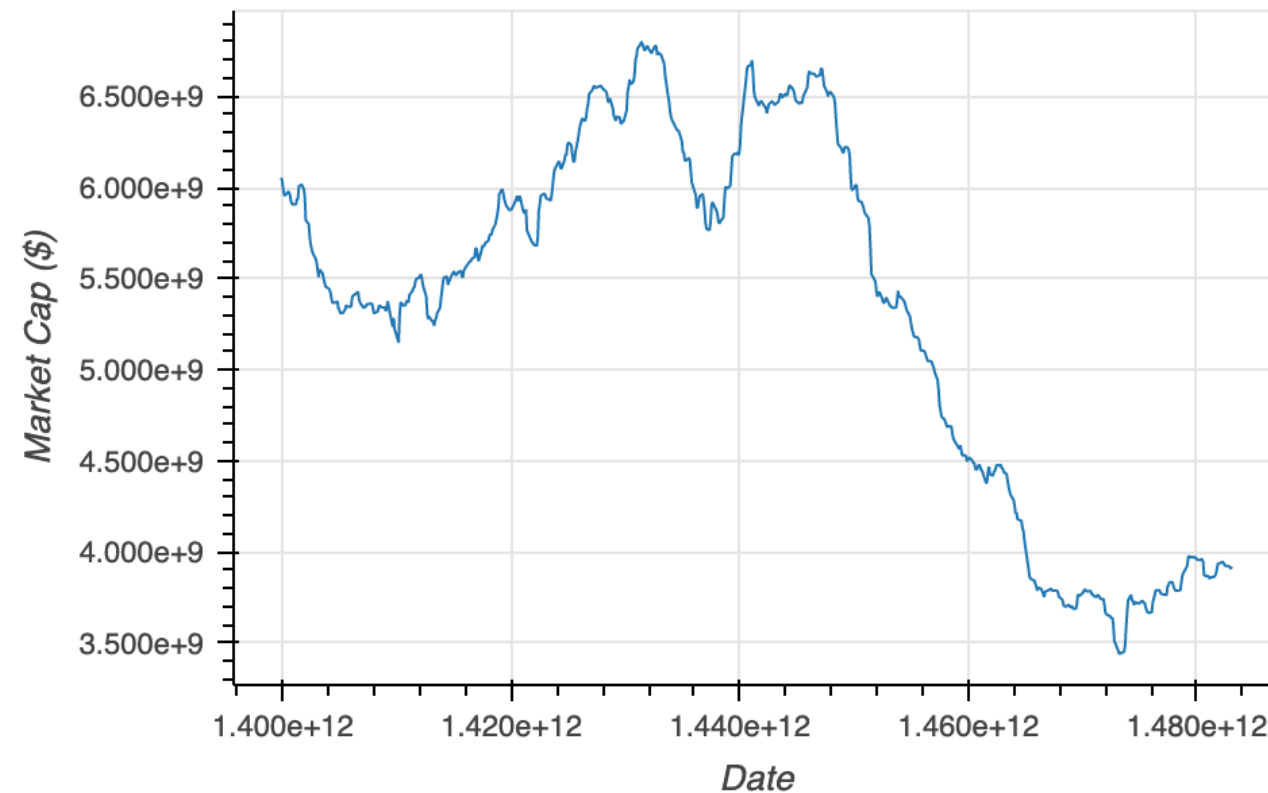


**George Boorman**

Core Curriculum Manager, DataCamp

# A line plot

```
source = ColumnDataSource(data=apple)
fig = figure(x_axis_label="Date", y_axis_label="Market Cap ($)")
fig.line(x="date", y="market_cap", source=source)
output_file(filename="unformatted_plot.html")
show(fig)
```



# X-axis type

```
source = ColumnDataSource(data=apple)
fig = figure(x_axis_label="Date", y_axis_label="Market Cap ($)",
            x_axis_type="datetime")
fig.line(x="date", y="market_cap", source=source)
output_file(filename="unformatted_plot.html")
show(fig)
```



# Formatting options

- NumeralTickFormatter
- Argument: `format`

Value	format	Output
<code>1500.00</code>	<code>"\$0.00"</code>	<code>\$1500.00</code>
<code>2000.00</code>	<code>"\$0"</code>	<code>\$2000</code>
<code>5000.00</code>	<code>"\$0,0"</code>	<code>\$5,000</code>
<code>1100000</code>	<code>"\$0.0a"</code>	<code>\$1.1m</code>
<code>500000000000</code>	<code>"\$0a"</code>	<code>\$5b</code>

- DatetimeTickFormatter
- Argument: `months`

Value	months	Output
<code>"2018-03-01"</code>	<code>"%B %Y"</code>	<code>"March 2018"</code>
<code>"2019-10-15"</code>	<code>"%b %Y"</code>	<code>"Oct 2019"</code>
<code>"2020-02-09"</code>	<code>"%b %y"</code>	<code>"Feb 20"</code>

- Other arguments:
  - `microseconds` , `milliseconds` , `seconds` ,  
`minisec` , `minutes` , `hourmin` , `hours` ,  
`days` , `years`

# NumeralTickFormatter

```
from bokeh.models import NumeralTickFormatter
fig = figure(x_axis_label="Date", y_axis_label="Price ($)")
fig.line(x="date", y="market_cap", source=source)
fig.yaxis[0].formatter = NumeralTickFormatter(format="$0.0a")
output_file(filename="formatted_y_axis.html")
show(fig)
```

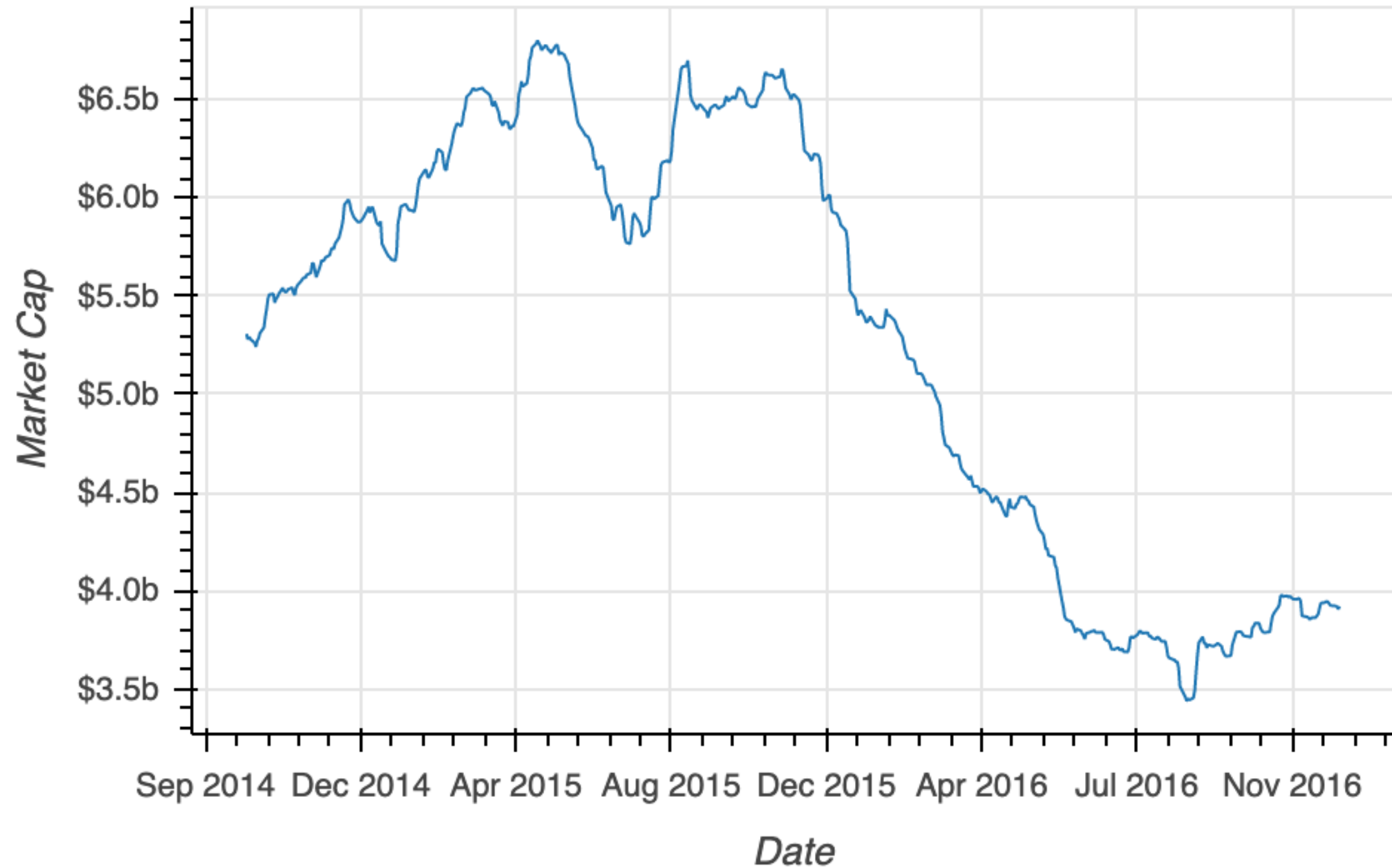


# DatetimeTickFormatter

```
from bokeh.models import NumeralTickFormatter, DatetimeTickFormatter
fig = figure(x_axis_label="Date", y_axis_label="Price ($)")
fig.line(x="date", y="price", source=source)
fig.yaxis[0].formatter = NumeralTickFormatter(format="$0.0a")
fig.xaxis[0].formatter = DatetimeTickFormatter(months="%b %Y")
output_file(filename="formatted_market_cap.html")
show(fig)
```



# The final plot

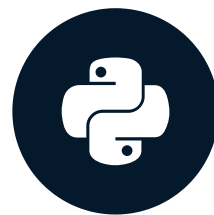


# Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

# Subplots

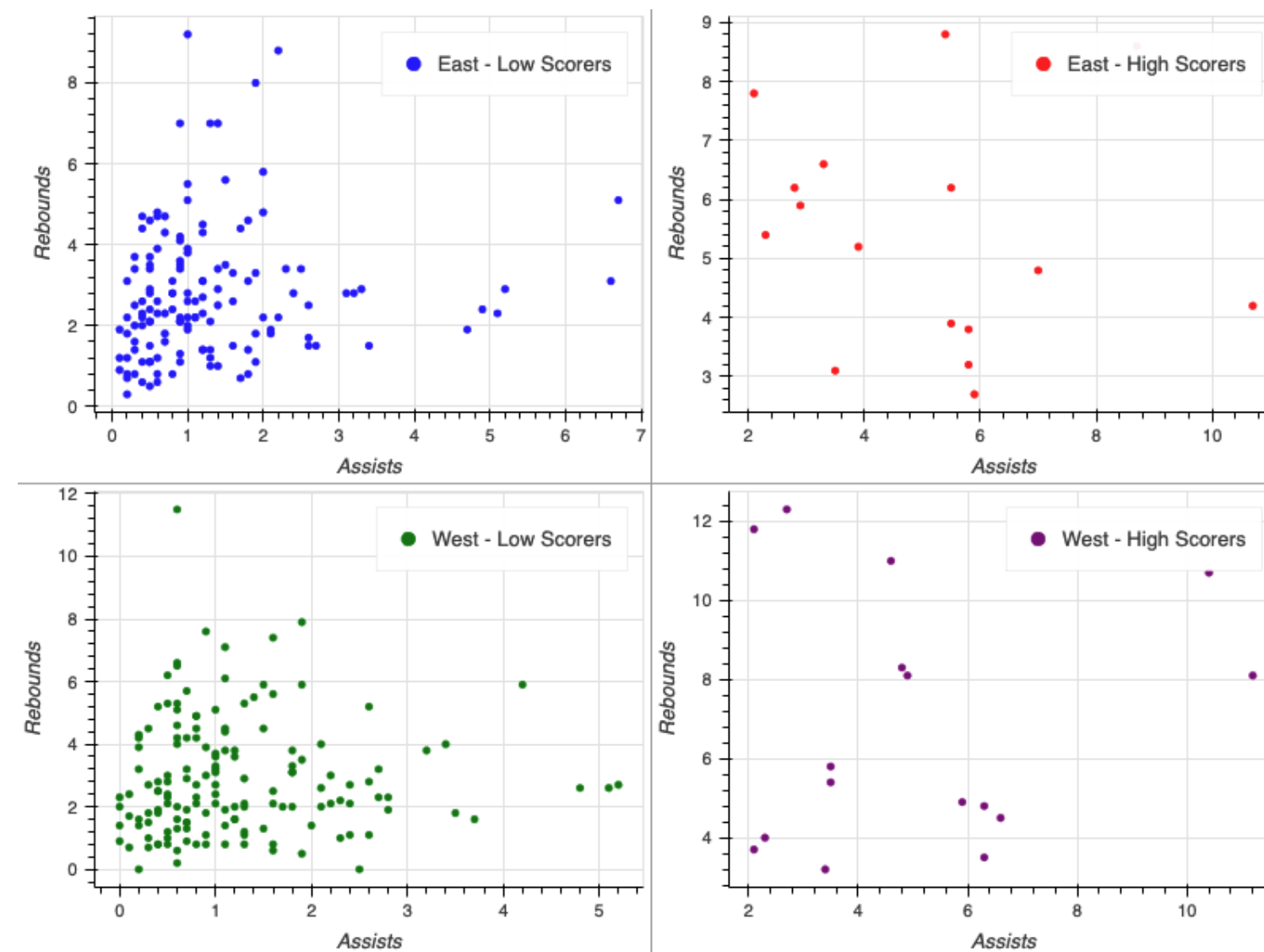
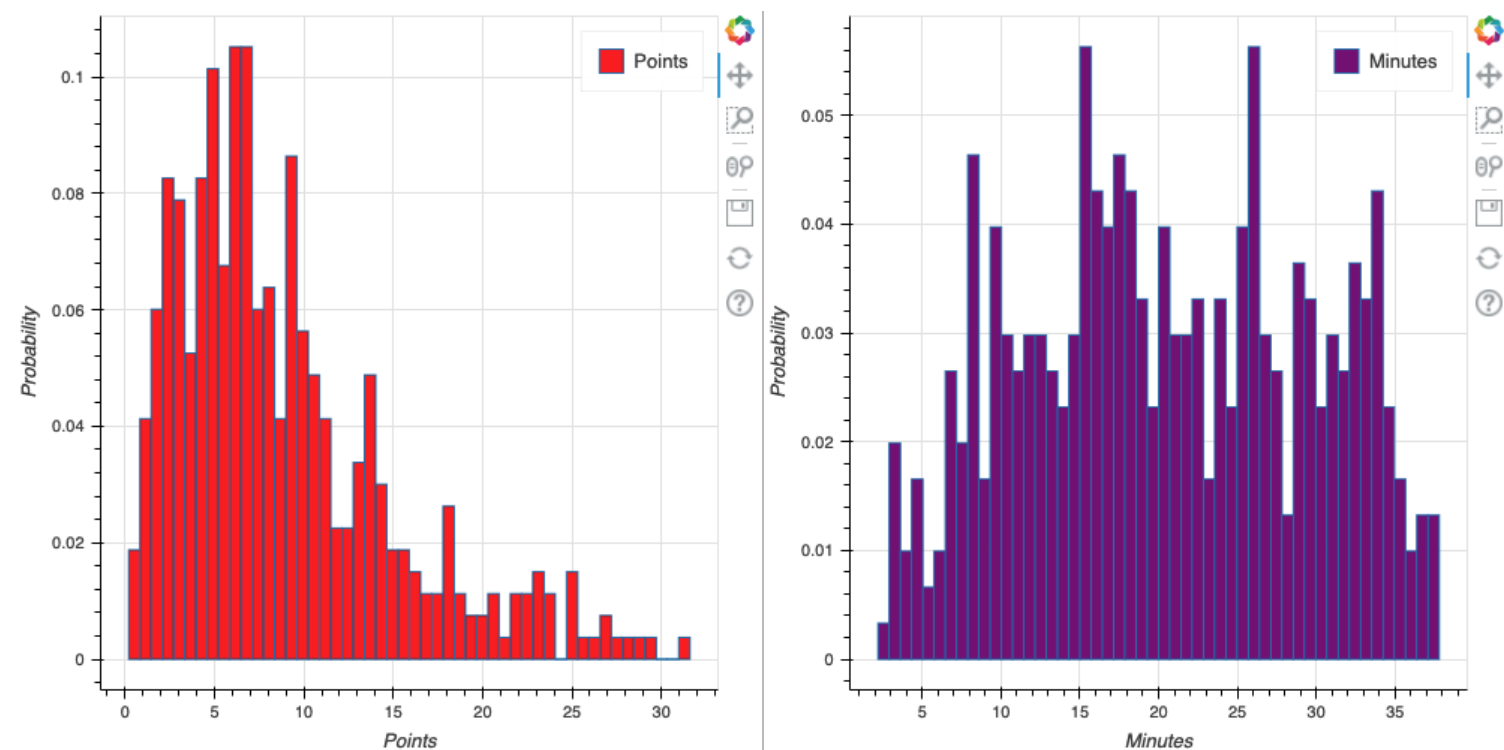
INTERACTIVE DATA VISUALIZATION WITH BOKEH



**George Boorman**

Core Curriculum Manager, DataCamp

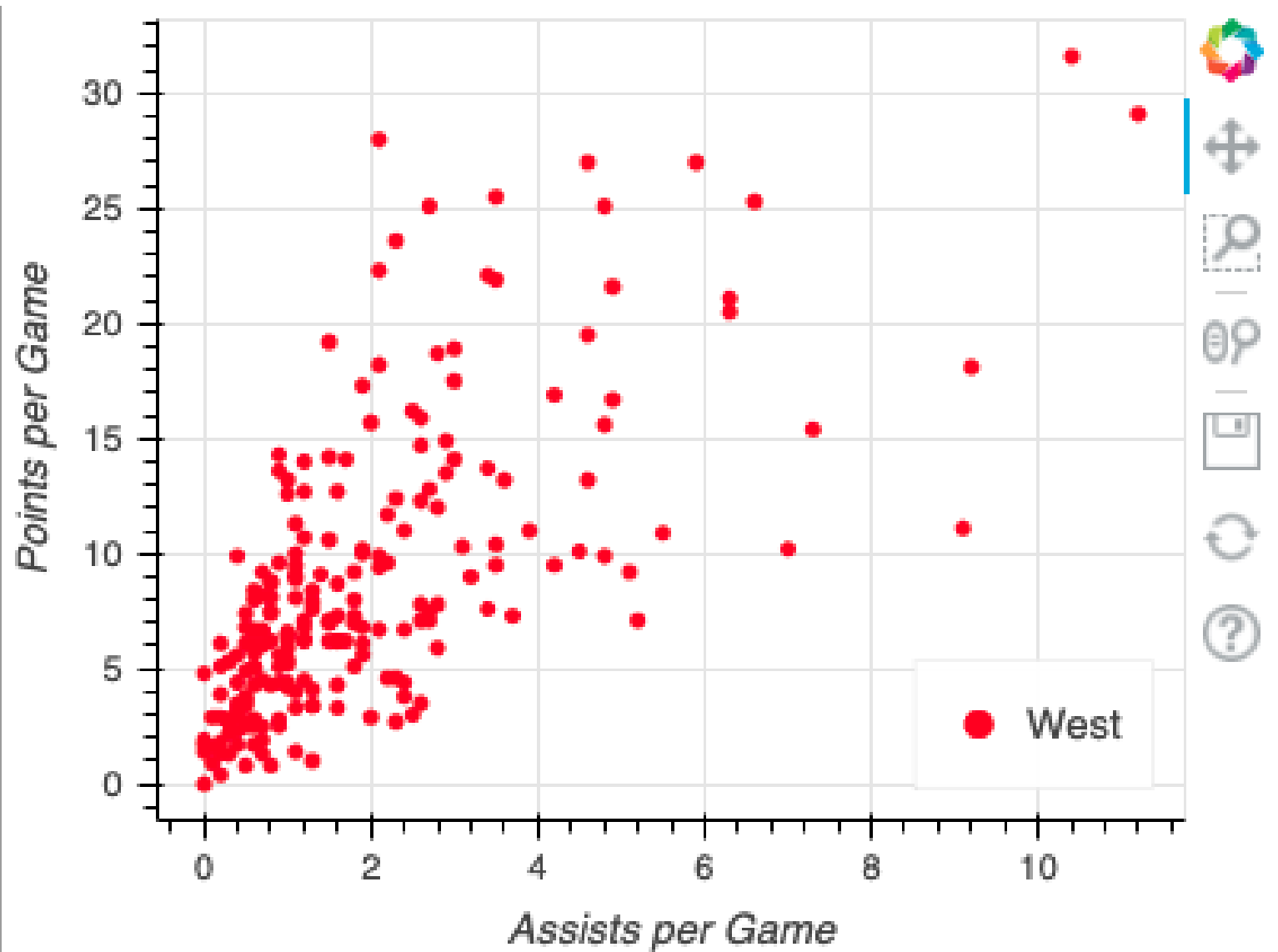
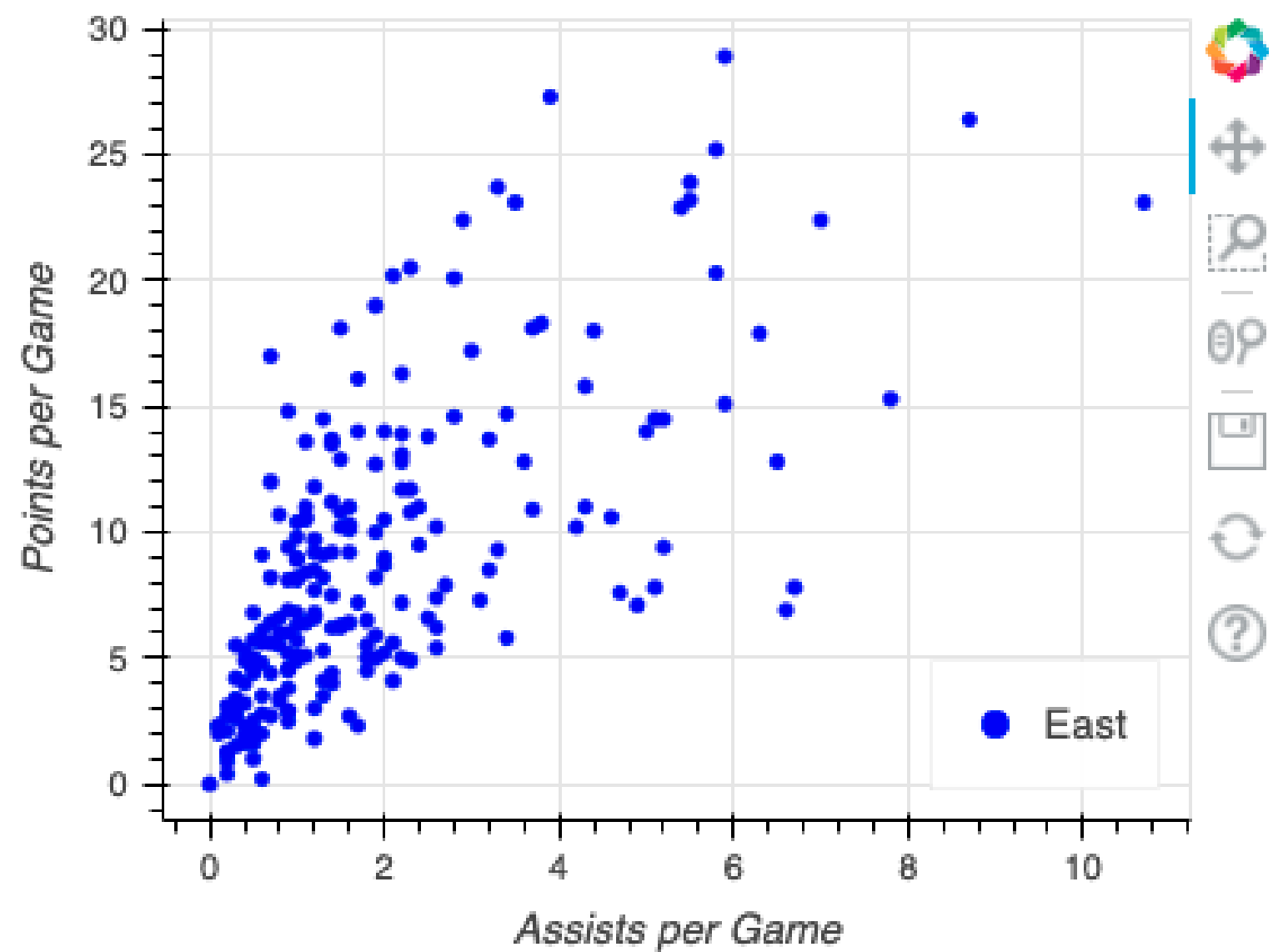
# Why use subplots?



# Building a row

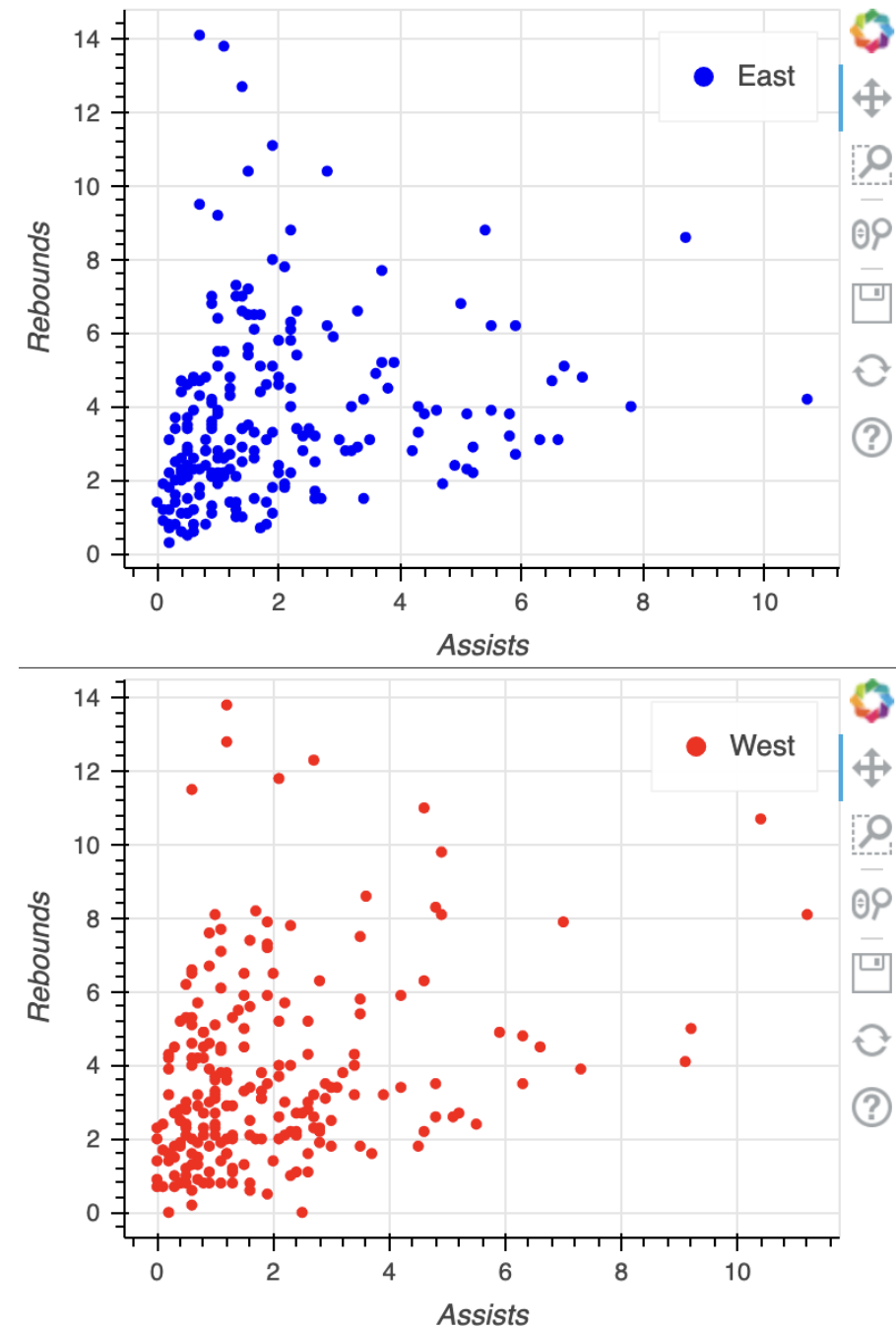
```
from bokeh.layouts import row
east_source = ColumnDataSource(data=east)
west_source = ColumnDataSource(data=west)
fig_one = figure(x_axis_label="Assists per Game", y_axis_label="Points per Game")
fig_two = figure(x_axis_label="Assists per Game", y_axis_label="Points per Game")
fig_one.circle(x="assists", y="points", source=east_source, color="blue", legend_label="East")
fig_two.circle(x="assists", y="points", source=west_source, color="blue", legend_label="West")
output_file(filename="row_plots.html")
show(row(fig_one, fig_two))
```

# Row subplot



# Column subplot

```
from bokeh.layouts import column
fig_one = figure(x_axis_label="Assists",
                 y_axis_label="Rebounds")
fig_two = figure(x_axis_label="Assists",
                 y_axis_label="Rebounds")
fig_one.circle(x="assists", y="rebounds",
               source=east_source,
               color="blue", legend_label="East")
fig_two.circle(x="assists", y="rebounds",
               source=east_source,
               color="blue", legend_label="East")
output_file(filename="column_plots.html")
show(column(fig_one, fig_two))
```

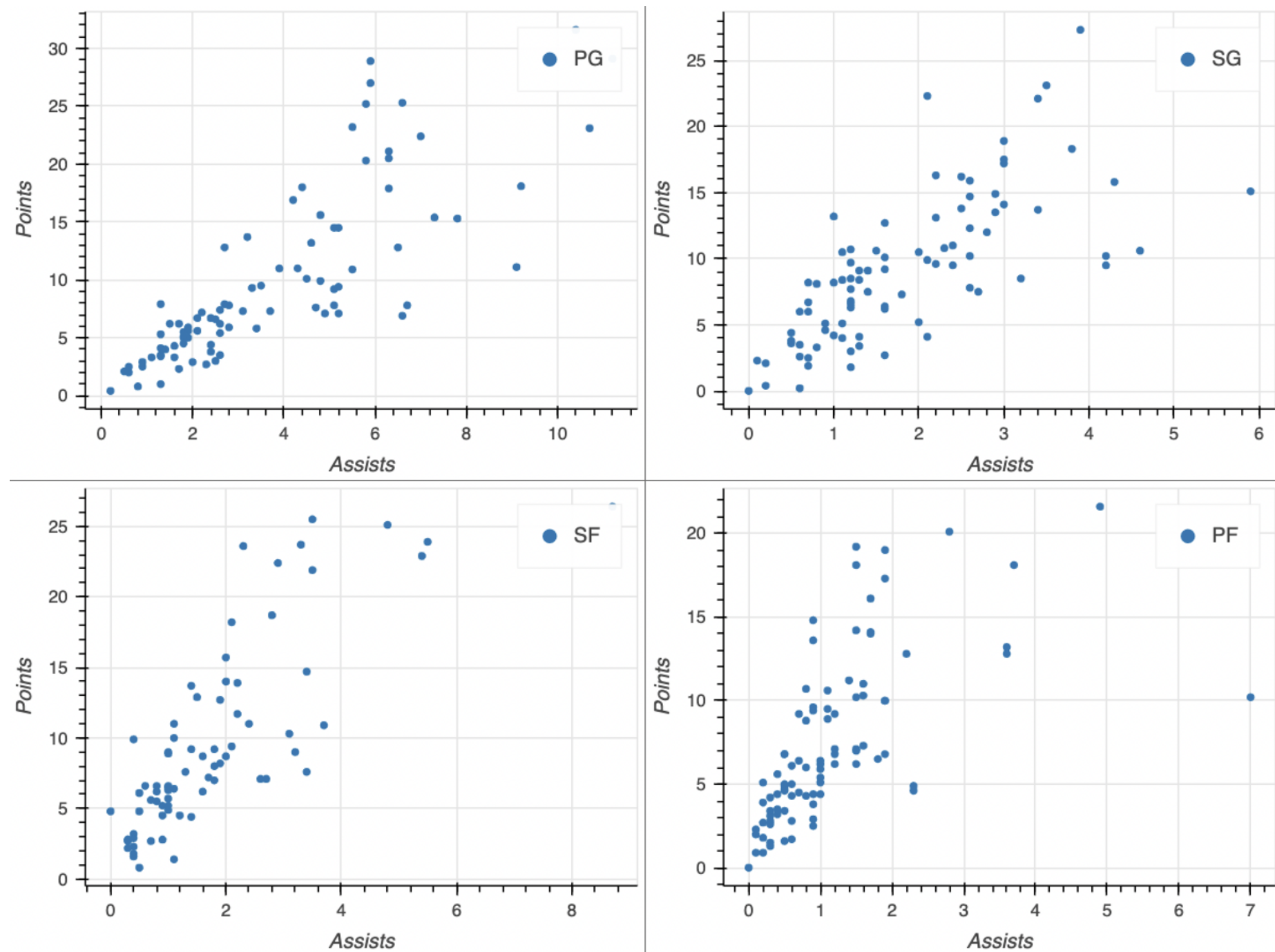


# Building a gridplot

```
from bokeh.layouts import gridplot
positions = ["PG", "SG", "SF", "PF"]
plots = []
for position in positions:
    nba_positions = nba.loc[nba["position"] == position]
    source = ColumnDataSource(data=nba_positions)
    fig = figure(x_axis_label="Assists", y_axis_label="Points")
    fig.circle(x="assists", y="points", source=source, legend_label=position)
    plots.append(fig)
output_file(filename="nba_gridplot.html")
show(gridplot(plots, ncols=2))
```

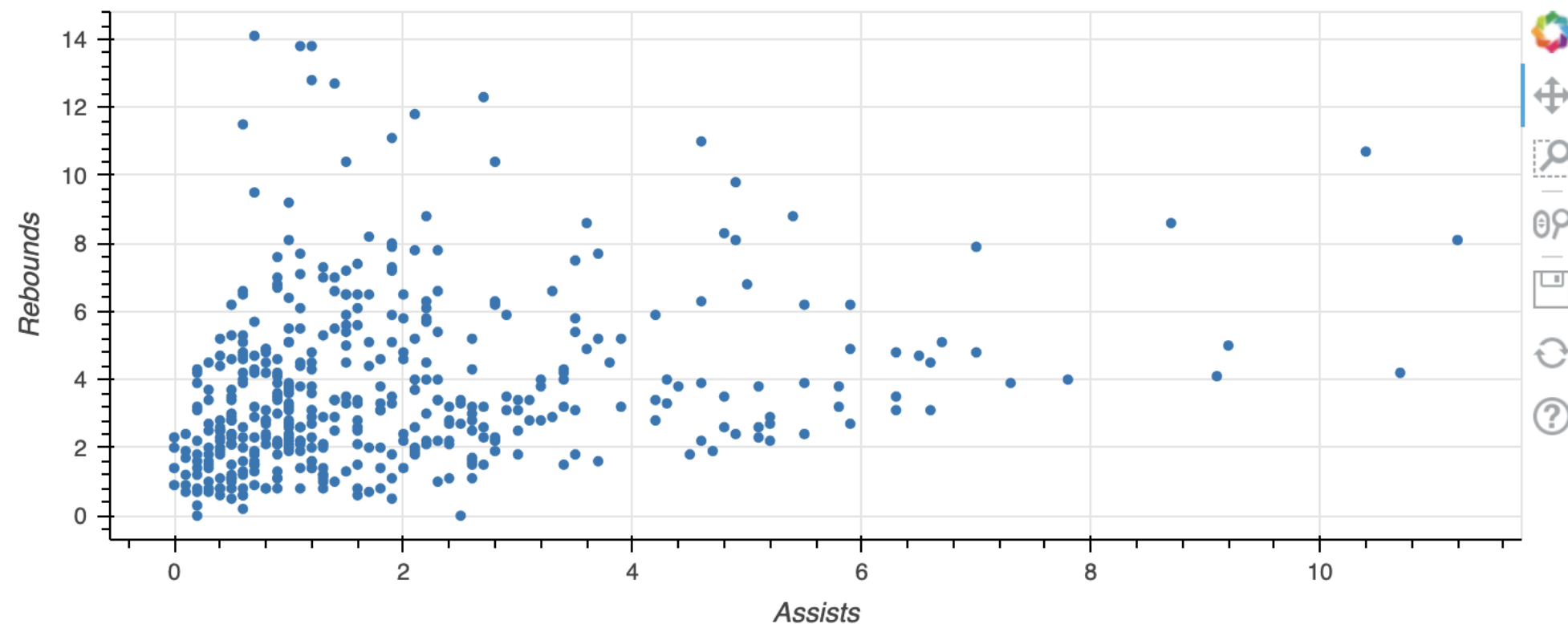


# Gridplot



# Customizing figure size

```
fig = figure(x_axis_label="Assists", y_axis_label="Rebounds",  
            width=750, height=300)  
fig.circle(x="assists", y="rebounds", source=source)  
output_file(filename="custom_size_plot.html")  
show(fig)
```



# Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

# Visualizing categorical data

INTERACTIVE DATA VISUALIZATION WITH BOKEH



**George Boorman**

Core Curriculum Manager, DataCamp

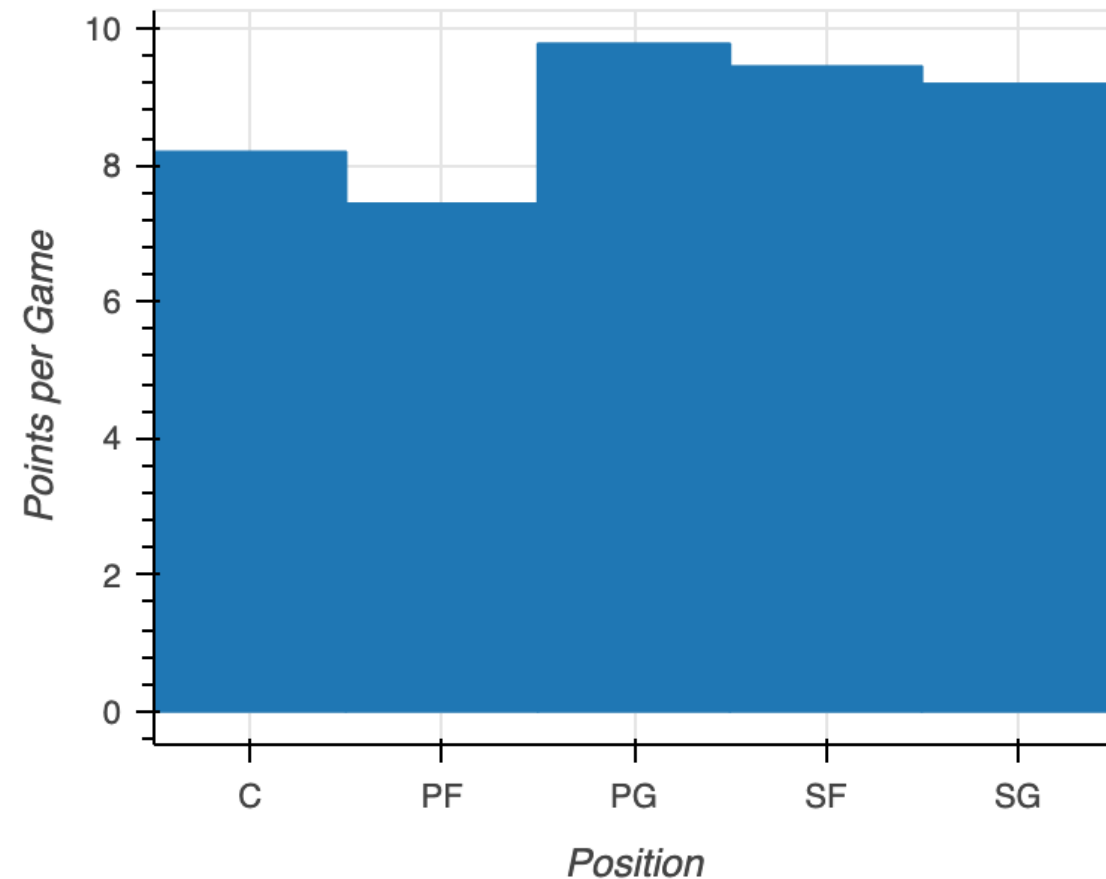
# Categorical data

- Categorical data is any data with a fixed number of options or labels.
  - Examples include gender or country of birth.
- Factors are another term for categorical variables

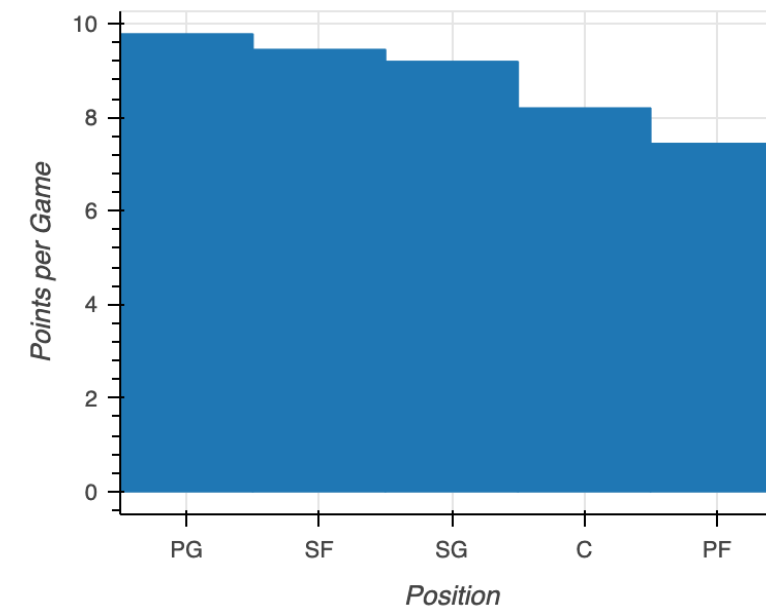
```
print(nba[["position", "team", "conference"]].head())
```

	position	team	conference
0	PG	OKC	West
1	PG	HOU	West
2	PG	BOS	East
3	C	NO	West
4	SG	TOR	East

# Sorting

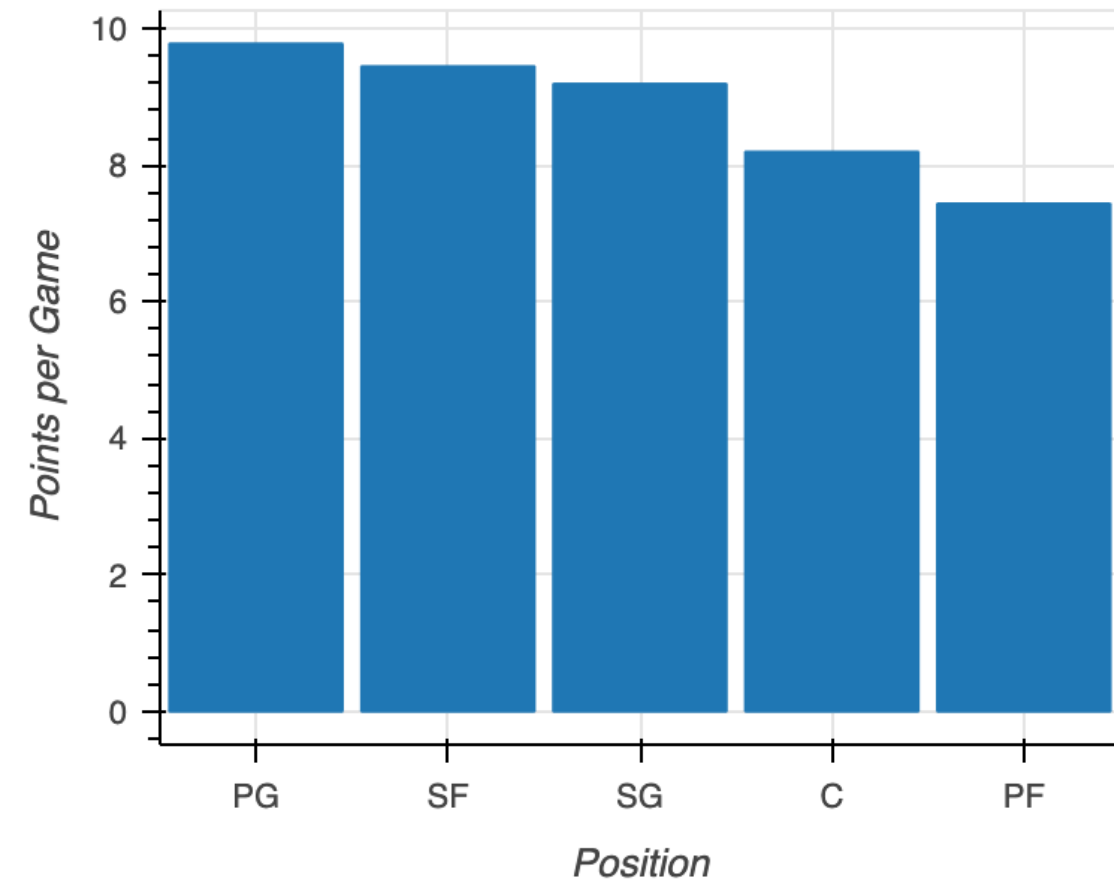


```
pos = nba.groupby("position")["points"].mean()
pos = pos.sort_values("points", ascending=False)
fig = figure(x_range=pos["position"],
             x_axis_label="Position",
             y_axis_label="Points per Game")
fig.vbar(x=pos["position"], top=pos["points"])
output_file(filename="sorted_plot.html")
show(fig)
```



# Padding

```
fig = figure(x_range=nba["position"],
             x_axis_label="Position",
             y_axis_label="Points per Game")
fig.vbar(x=nba["position"], top=nba["points"],
         width=0.9)
output_file(filename="padded_plot.html")
show(fig)
```

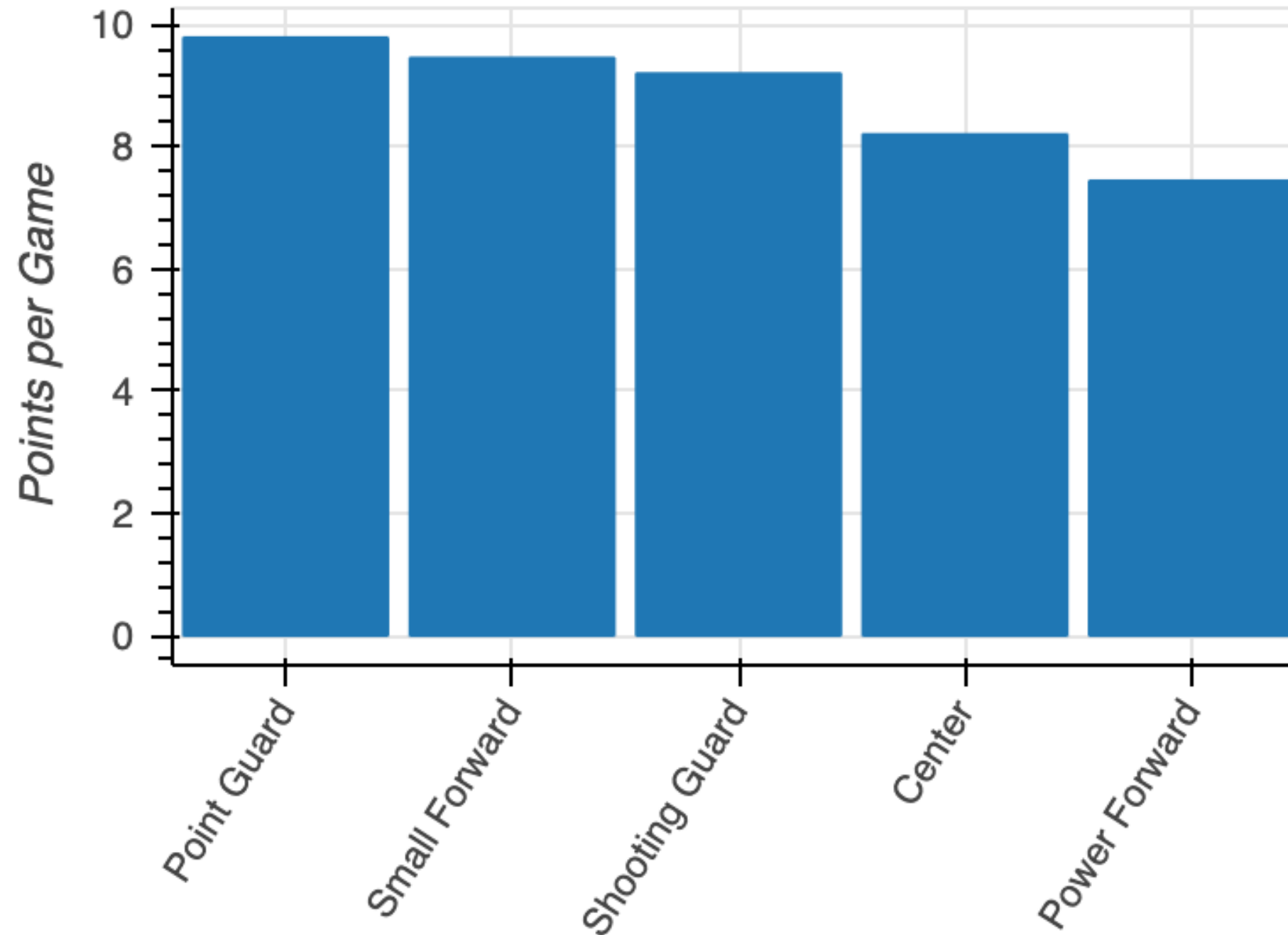


# Orientation

```
fig = figure(x_range=nba['position'],
             x_axis_label="Position",
             y_axis_label="Points per Game")
fig.vbar(x=nba["position"], top=nba["points"], width=0.9)
fig.xaxis.major_label_orientation = 45
output_file(filename="rotated_x_label_plot.html")
show(fig)
```



# Rotated x-axis labels



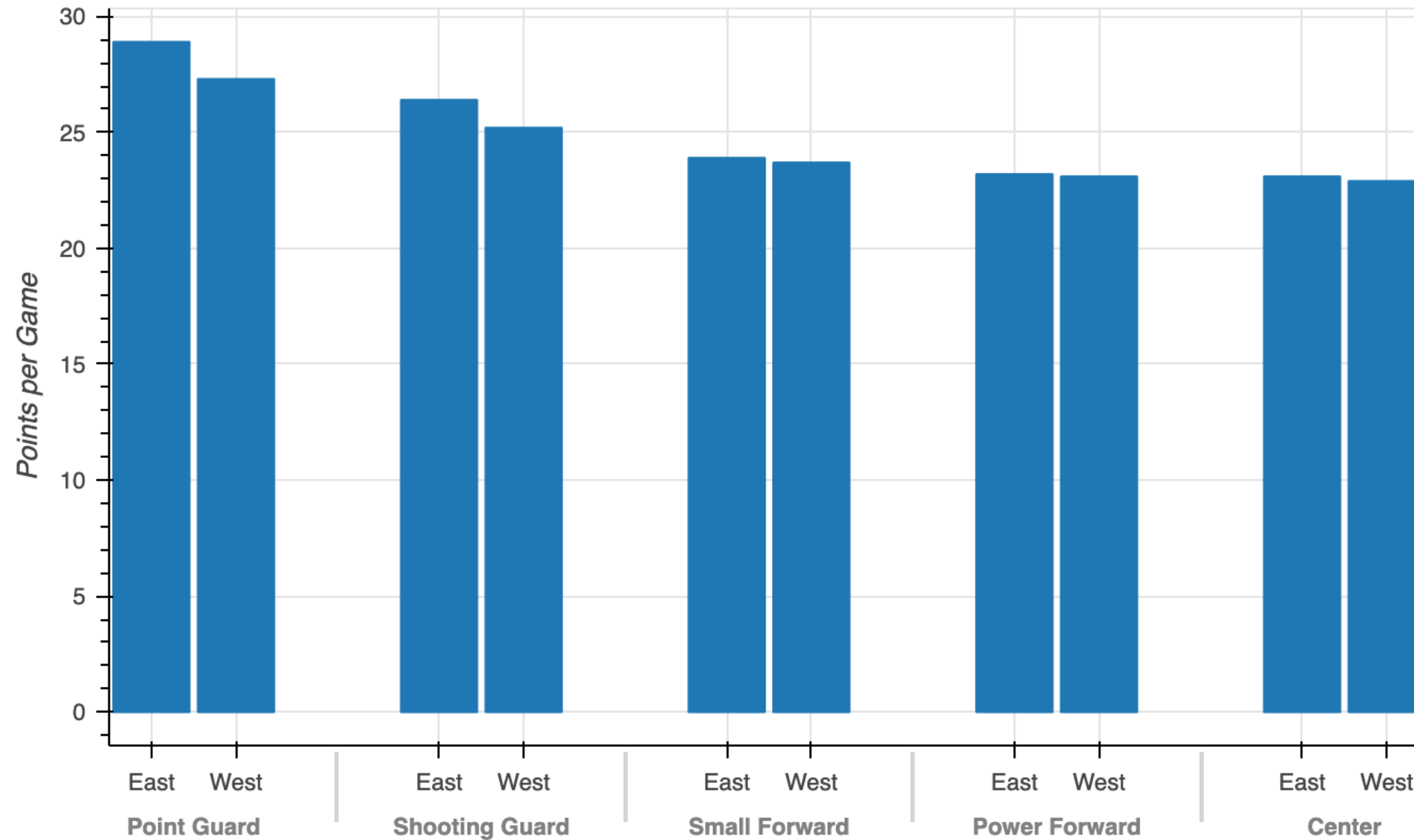
# Nested categories

```
positions = ["Point Guard", "Shooting Guard",  
            "Small Forward", "Power Forward", "Center"]  
conferences = ["East", "West"]  
factors = [("Point Guard", "East"), ("Point Guard", "West"),  
          ("Shooting Guard", "East"), ("Shooting Guard", "West"),  
          ("Small Forward", "East"), ("Small Forward", "West"),  
          ("Power Forward", "East"), ("Power Forward", "West"),  
          ("Center", "East"), ("Center", "West")]
```

# Building a grouped bar plot

```
from bokeh.models import FactorRange
fig = figure(x_range=FactorRange(*factors), y_axis_label="Points per Game")
fig.vbar(x=factors, top=nba["points"], width=0.9)
output_file(filename="grouped_bar_plot.html")
show(fig)
```

# Grouped bar plot



# Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH