Customizing glyph settings

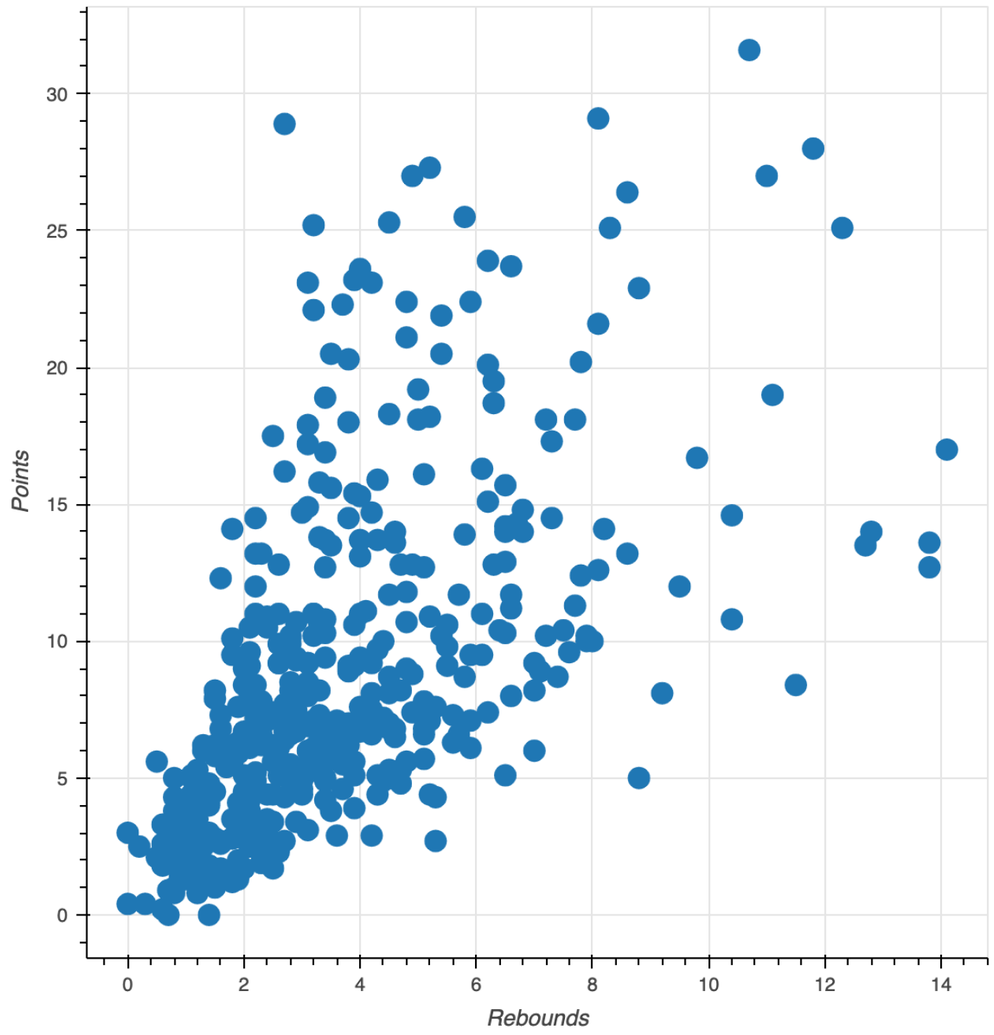
INTERACTIVE DATA VISUALIZATION WITH BOKEH

George Boorman
Core Curriculum Manager, DataCamp

datacamp

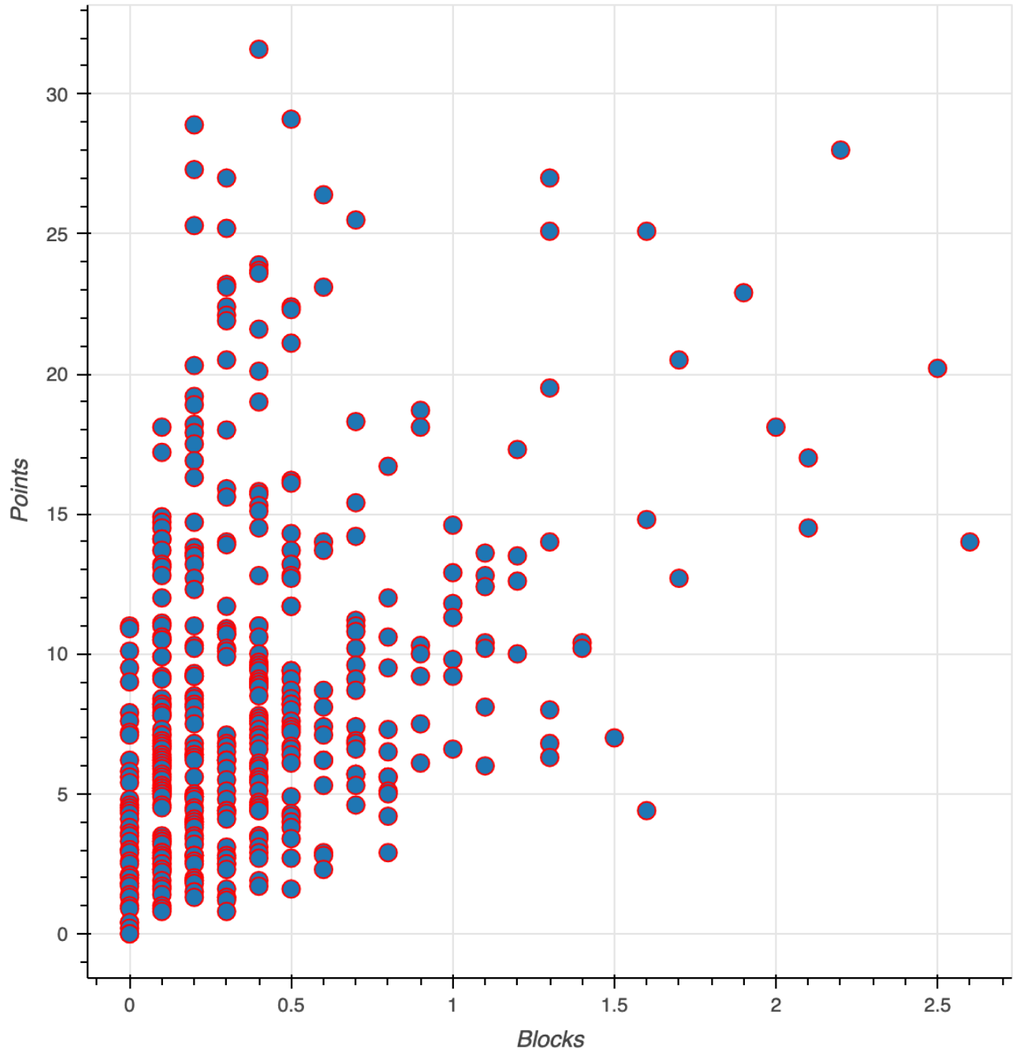
Glyph size

```
fig = figure(x_axis_label="Rebounds",
             y_axis_label="Points")
fig.circle(x="rebounds", y="points",
           source=source, size=12)
output_file(filename="size_12_scatter.html")
show(fig)
```



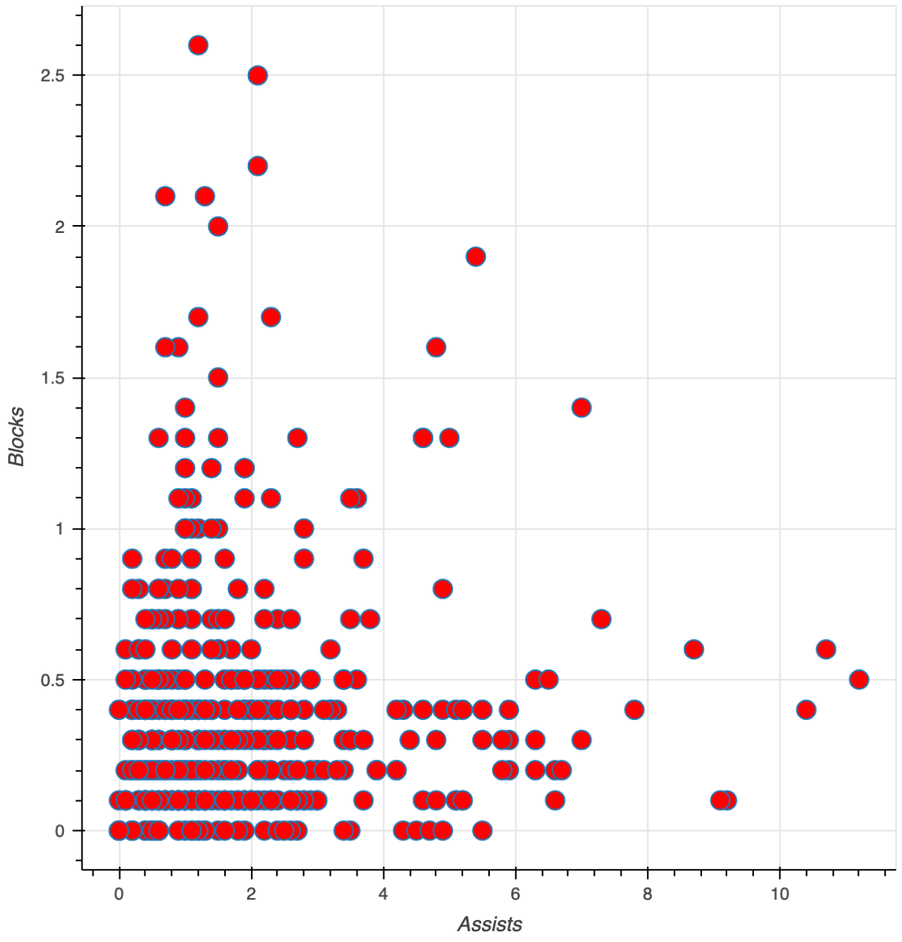
Glyph outline color

```
fig = figure(x_axis_label="Blocks",
             y_axis_label="Points")
fig.circle(x="blocks", y="points",
           source=source, size=10,
           line_color="red")
output_file(filename="red_lines.html")
show(fig)
```



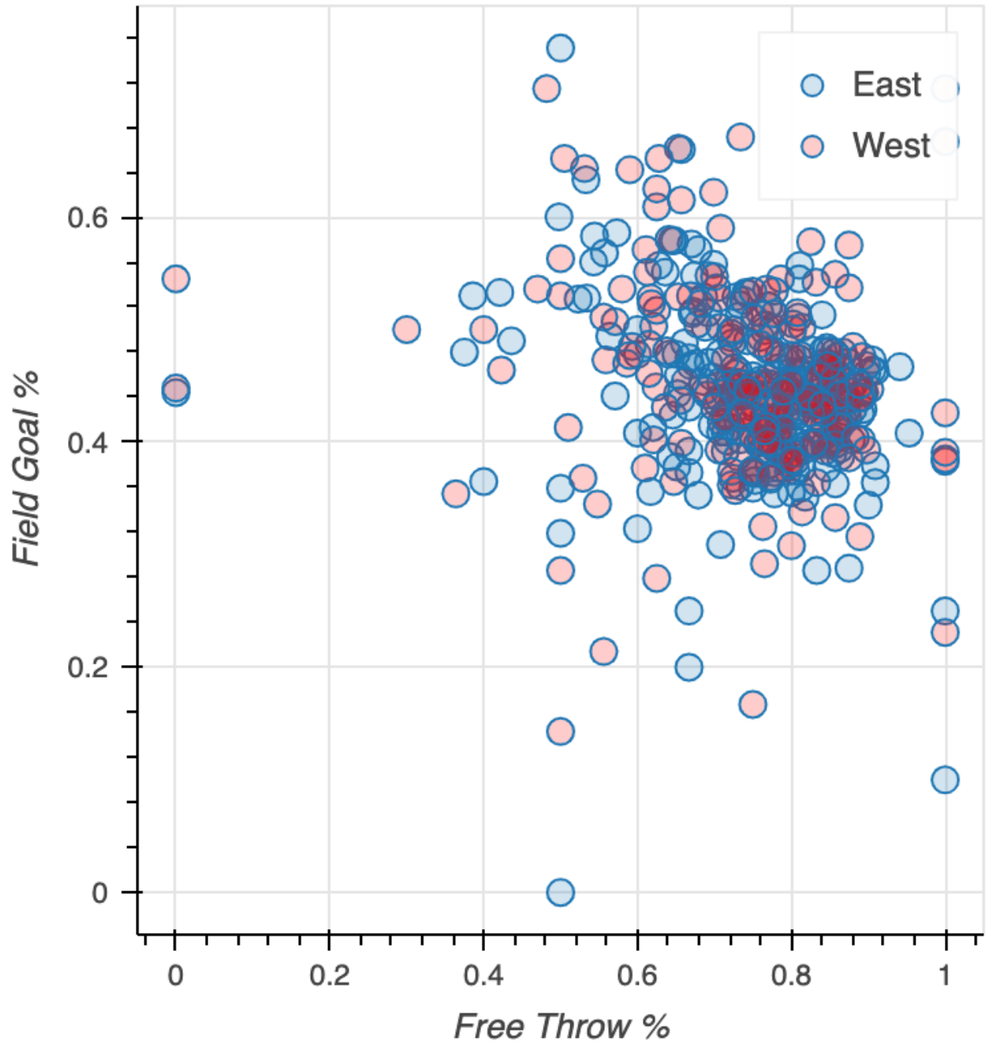
Glyph fill color

```
fig = figure(x_axis_label="Assists",
             y_axis_label="Blocks")
fig.circle(x="blocks", y="points",
           source=source, size=12,
           fill_color="red")
output_file(filename="red_circles.html")
show(fig)
```



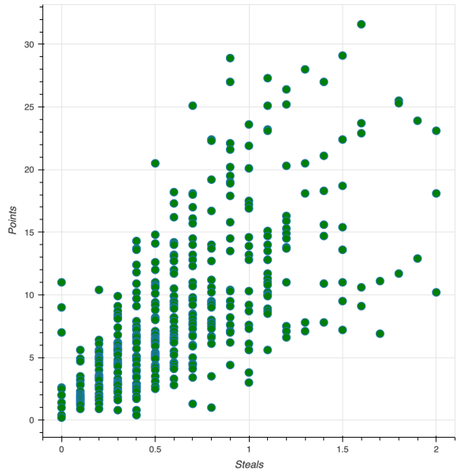
Glyph transparency

```
fig = figure(x_axis_label="Free Throw %",
             y_axis_label="Field Goal %")
fig.circle(x="free_throw_perc",
           y="field_goal_perc",
           source=east, size=10,
           fill_alpha=0.2, legend_label="East")
fig.circle(x="free_throw_perc",
           y="field_goal_perc",
           source=west, size=10,
           fill_alpha=0.2, fill_color="red",
           legend_label="West")
output_file(filename="transparent.html")
show(fig)
```

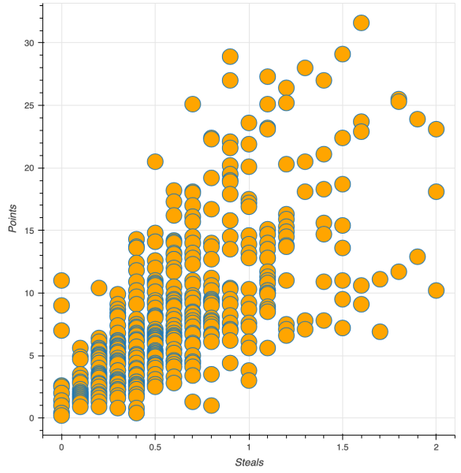


Updating glyphs

```
fig = figure(x_axis_label="Steals",
             y_axis_label="Points")
circle = fig.circle(x="steals", y="assists",
                    source=source, size=10,
                    fill_color="green")
output_file(filename="original.html")
show(fig)
```




```
circle.glyph.size = 20
circle.glyph.fill_color = "orange"
output_file(filename="updated.html")
show(fig)
```



Line glyphs

Scatter argument	Line plot equivalent
`size`	`line_width`
`fill_alpha`	`alpha`
`color`	`line_color`
`fill_color`	`Not applicable`
`line_color`	`Not applicable`

The dataset

```
print(nba.iloc[:3, :6])
```

```
     player            position    minutes    field_goal_perc    three_point_perc    free_throw_perc
0    Russell Westbrook PG          34.6       0.425              0.343               0.845
1    James Harden      PG          36.4       0.440              0.347               0.847
2    Isaiah Thomas     PG          33.8       0.463              0.379               0.909
```

```
print(nba.iloc[:3, 6:])
```

```
     rebounds    assists    steals    blocks    points    team    conference    scorer_category
0    10.7        10.4       1.6       0.4       31.6      OKC     West          High Scorer
1    8.1         11.2       1.5       0.5       29.1      HOU     West          High Scorer
2    2.7         5.9        0.9       0.2       28.9      BOS     East          High Scorer
```

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Highlighting and contrasting

INTERACTIVE DATA VISUALIZATION WITH BOKEH

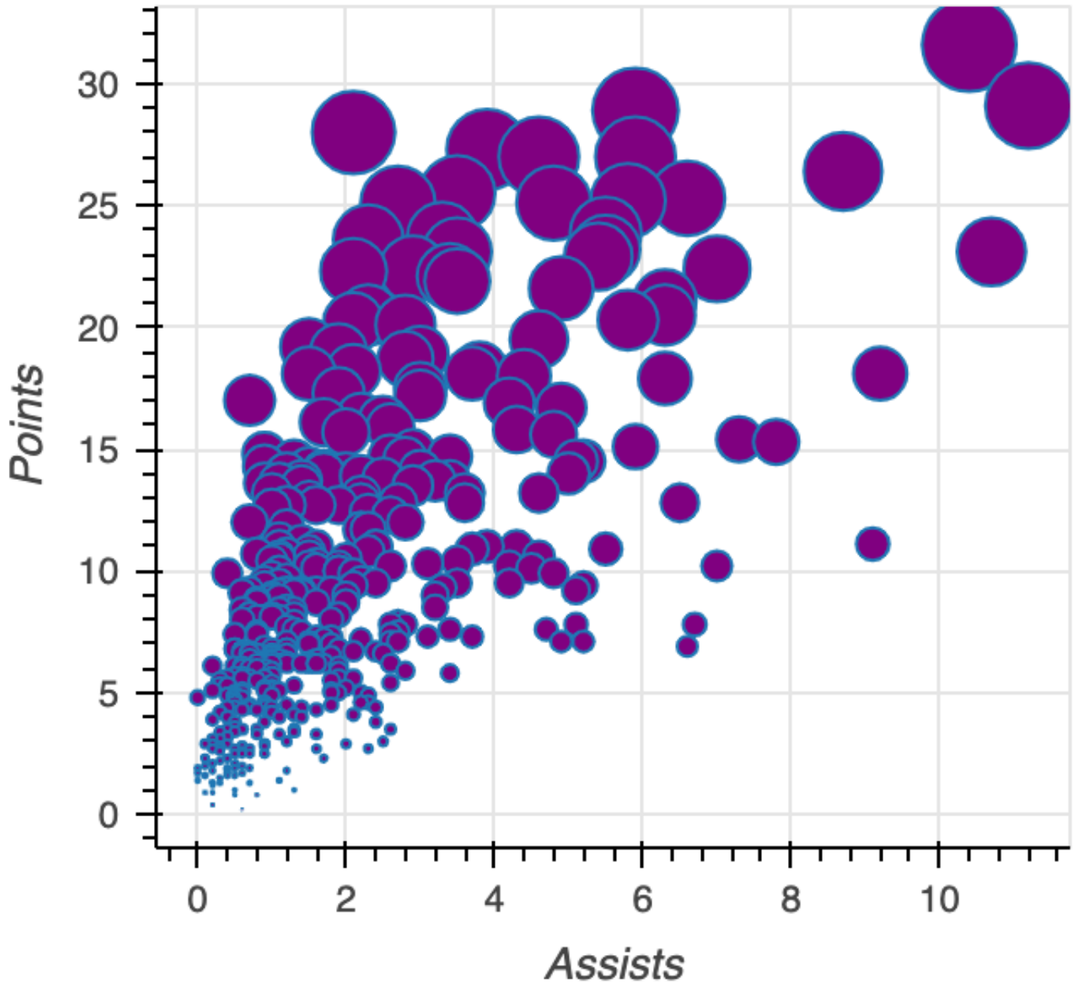
George Boorman
Core Curriculum Manager, DataCamp

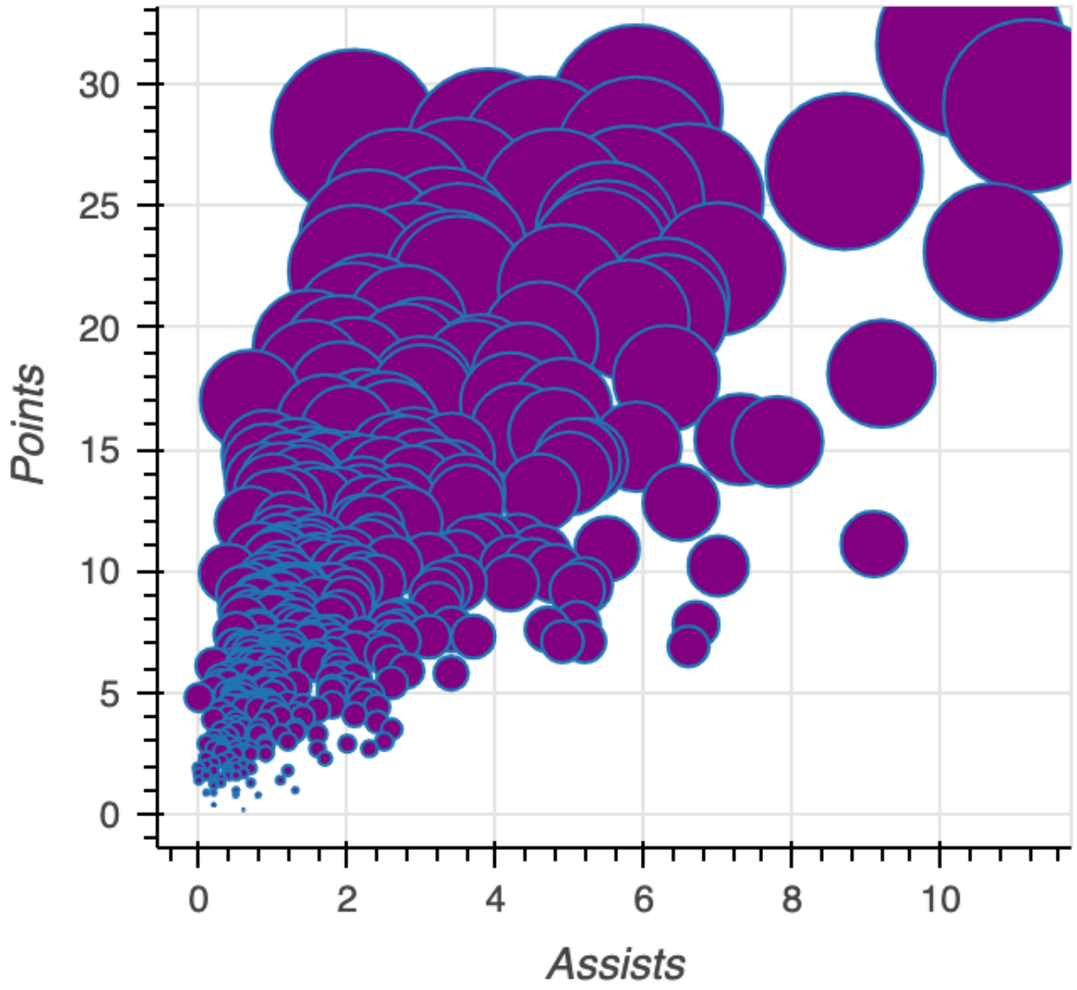
datacamp

Vectorizing glyph size

```
sizes = nba["points"] / 50
fig = figure(x_axis_label="Assists", y_axis_label="Points")
fig.circle(x=nba["assists"], y=nba["points"], fill_color="purple", radius=sizes)
output_file(filename="glyph_vectorization.html")
show(fig)
```

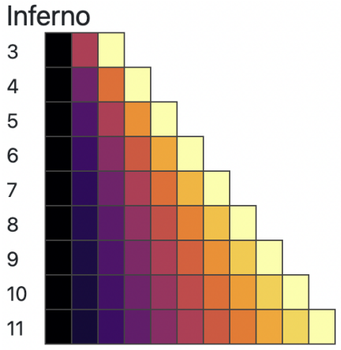
Different sizes



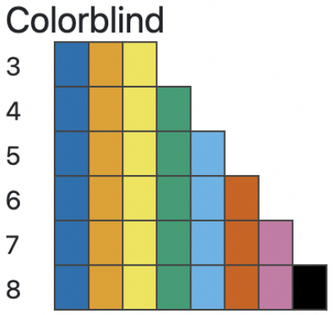


Palettes

```
from bokeh.palettes import Inferno3
```



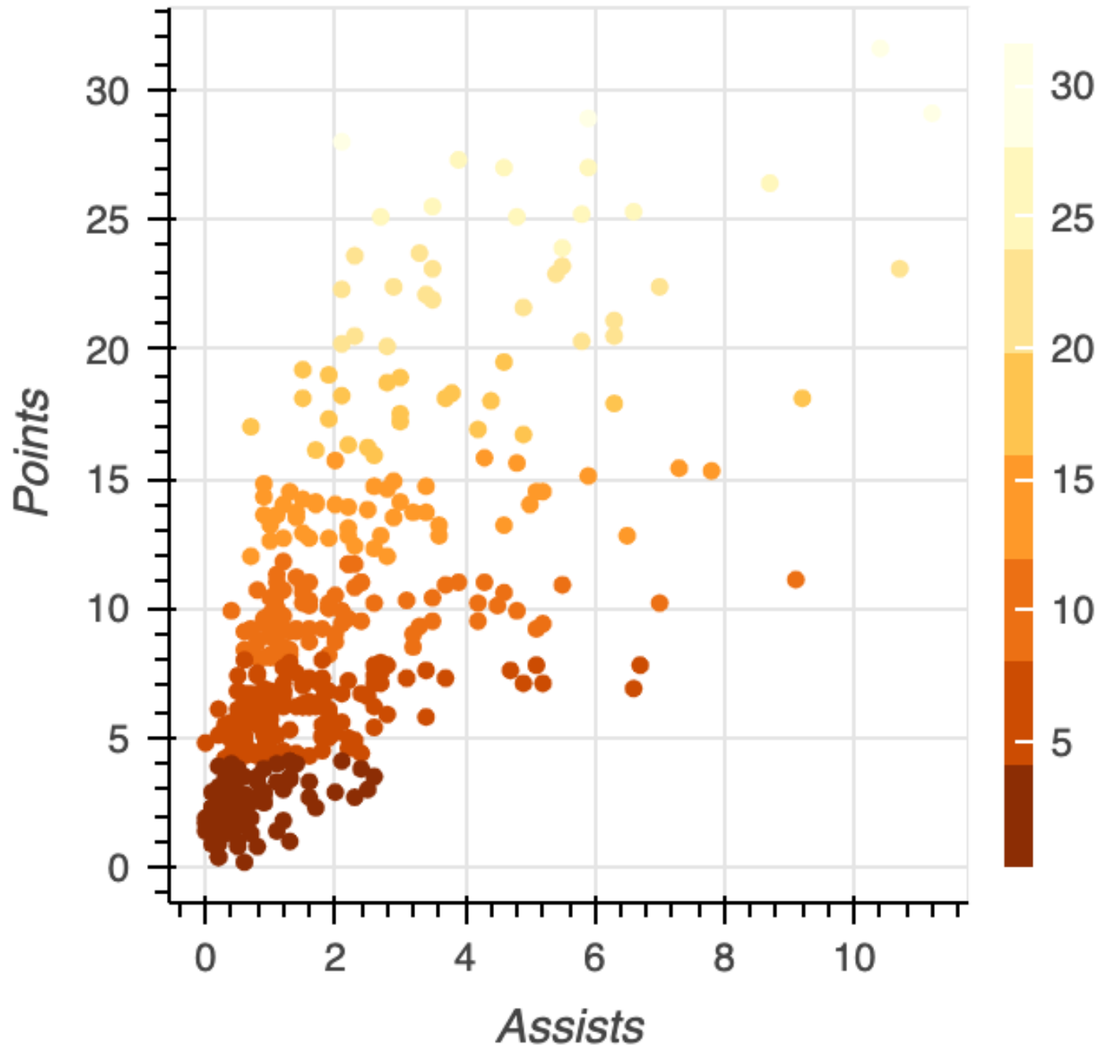

```
from bokeh.palettes import Colorblind4
```




```
from bokeh.palettes import __palettes__
__palettes__[:8]
```

```
['Accent3',
 'Accent4',
 'Accent5',
 'Accent6',
 'Accent7',
 'Accent8',
 'Blues3',
 'Blues4']
```

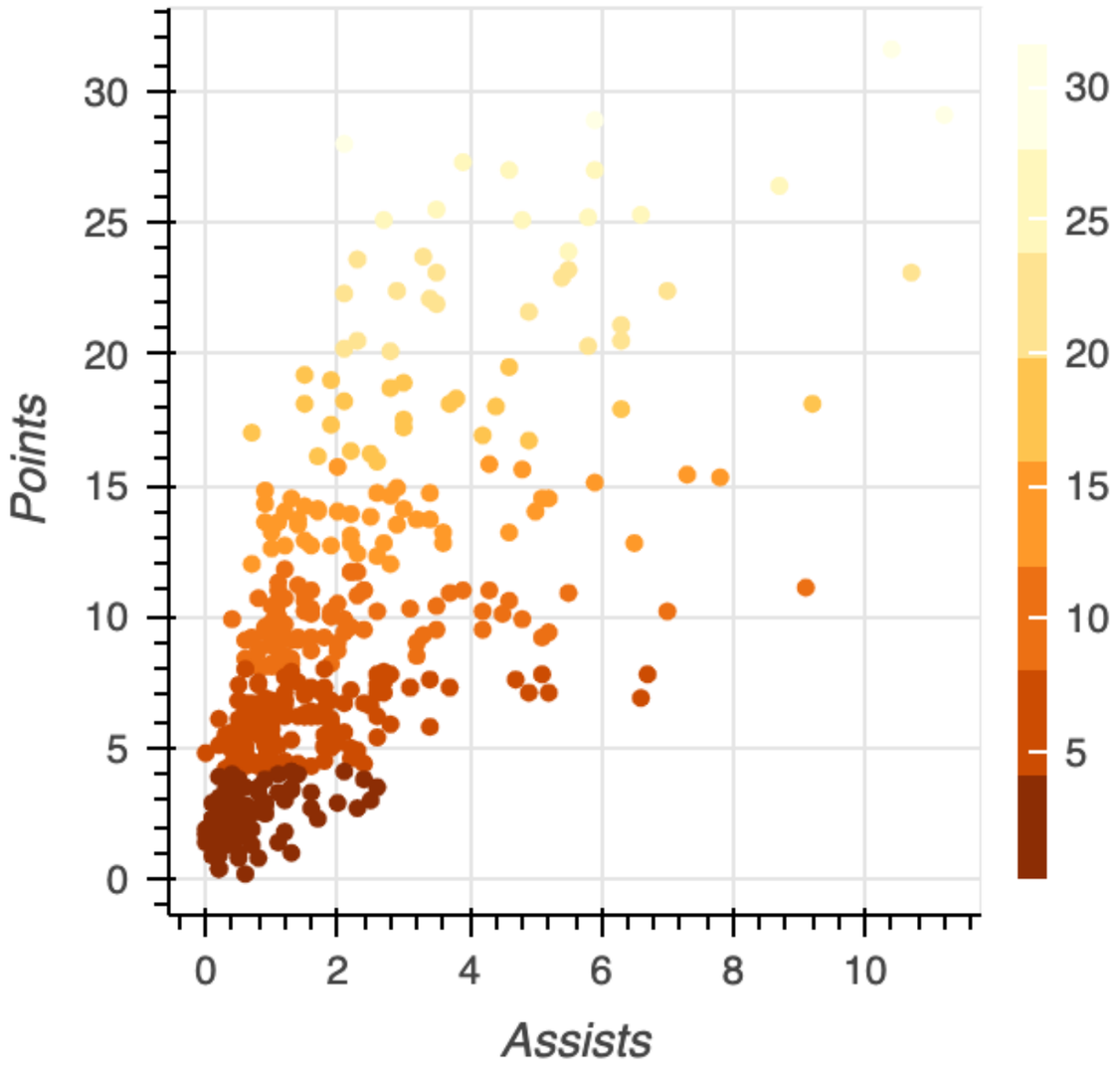
Color mapping and color bars



Linear color mapping

```
from bokeh.transform import linear_cmap
from bokeh.palettes import YlOrBr8
from bokeh.models import ColorBar
mapper = linear_cmap(field_name="points", palette=YlOrBr8,
                     low=min(nba["points"])), high=max(nba["points"]))
fig = figure(x_axis_label="Assists", y_axis_label="Points")
fig.circle(x="assists", y="points", source=source, fill_color=mapper, line_color=mapper)
color_bar = ColorBar(color_mapper=mapper["transform"], width=8)
fig.add_layout(color_bar, "right")
output_file(filename="linear_cmap.html")
show(fig)
```

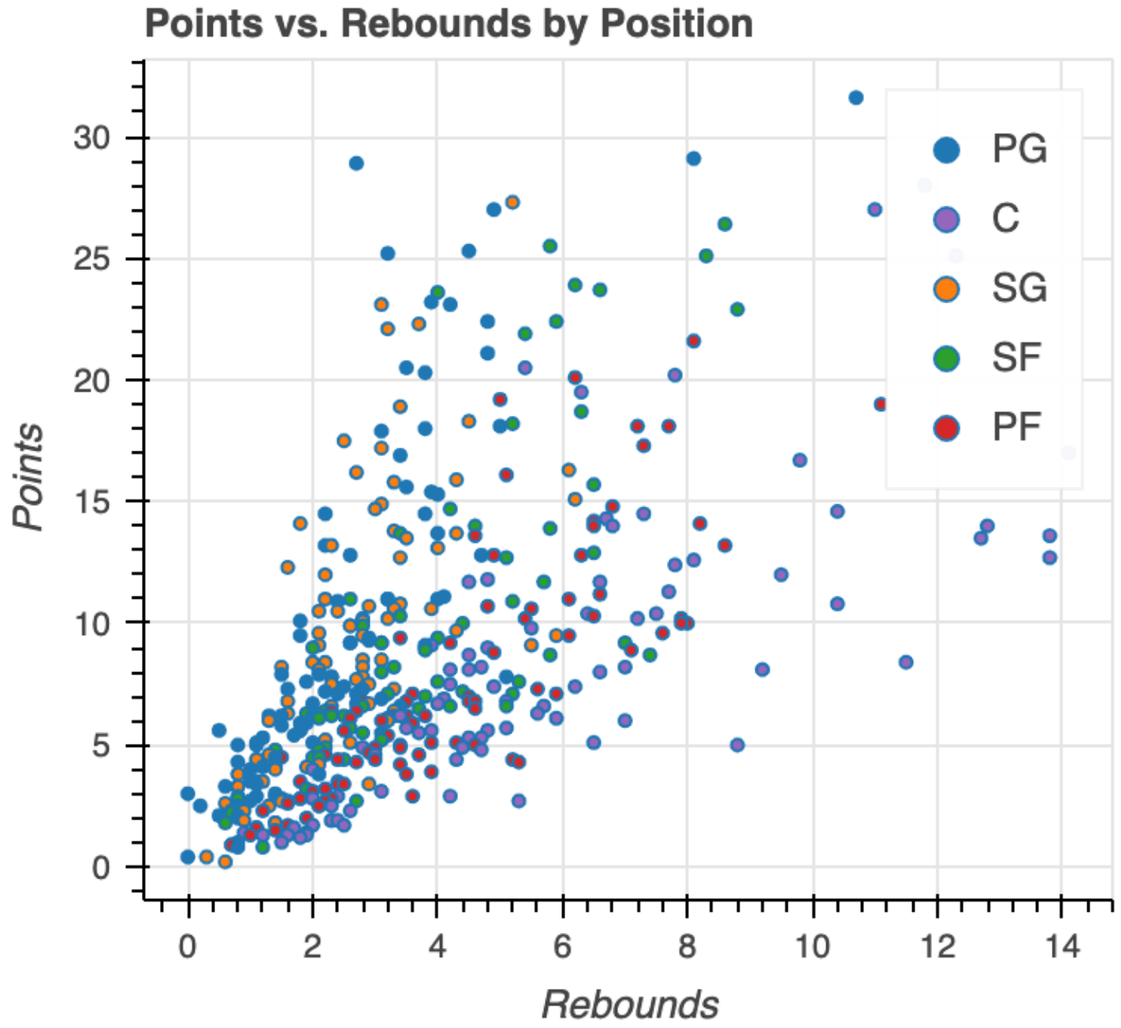
Linear color map scatter plot



Factor color mapping

```
from bokeh.transform import factor_cmap
from bokeh.palettes import Category10_5
positions = ["PG", "SG", "SF", "PF", "C"]
fig = figure(x_axis_label="Rebounds", y_axis_label="Points")
fig.circle(x="rebounds", y="points", source=source,
         fill_color=factor_cmap("position", palette=Category10_5, factors=positions),
         legend_field="position")
output_file(filename="factor_cmap.html")
show(fig)
```

Color categorized bar plot



Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

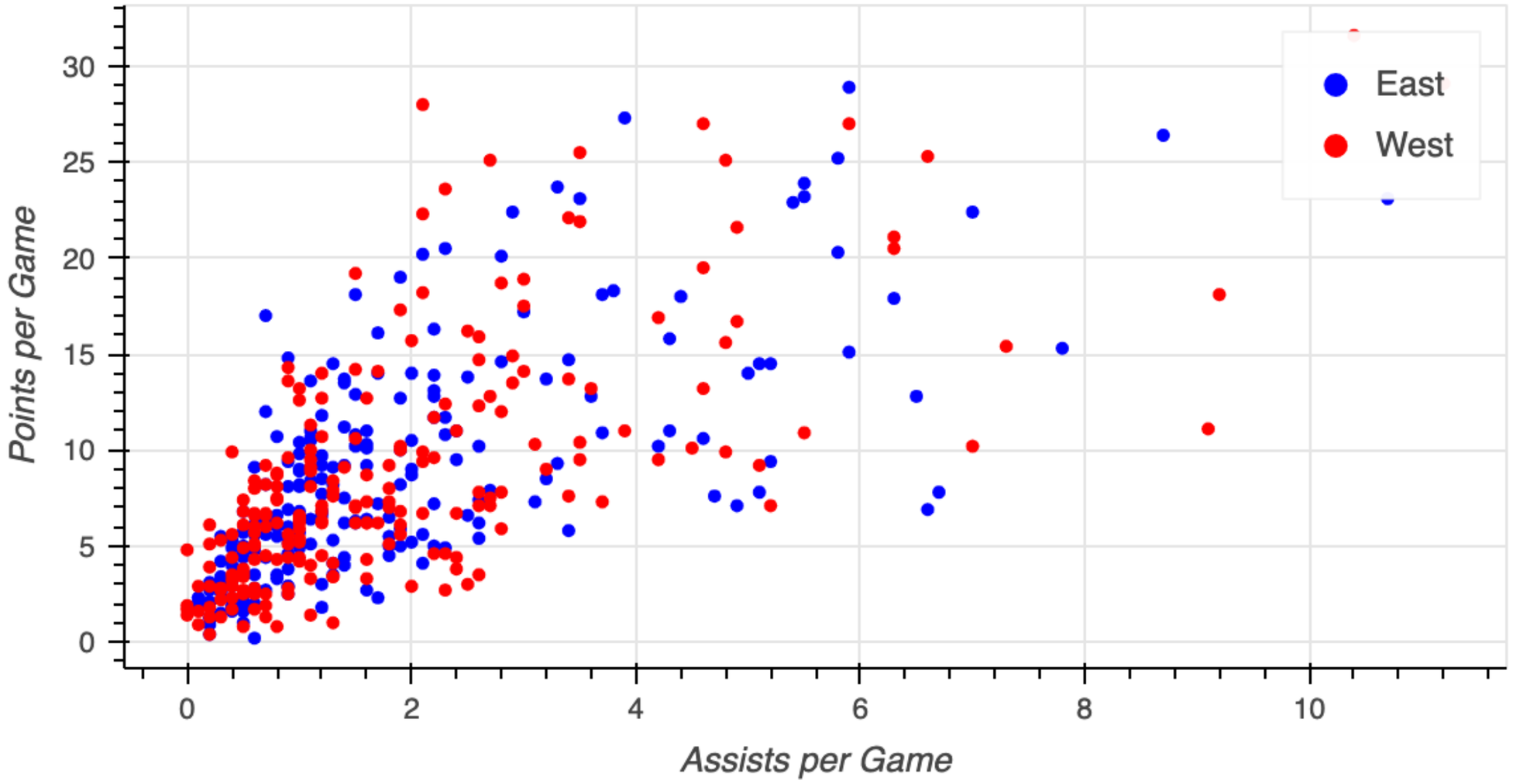
Communicating with text

INTERACTIVE DATA VISUALIZATION WITH BOKEH

George Boorman
Core Curriculum Manager, DataCamp

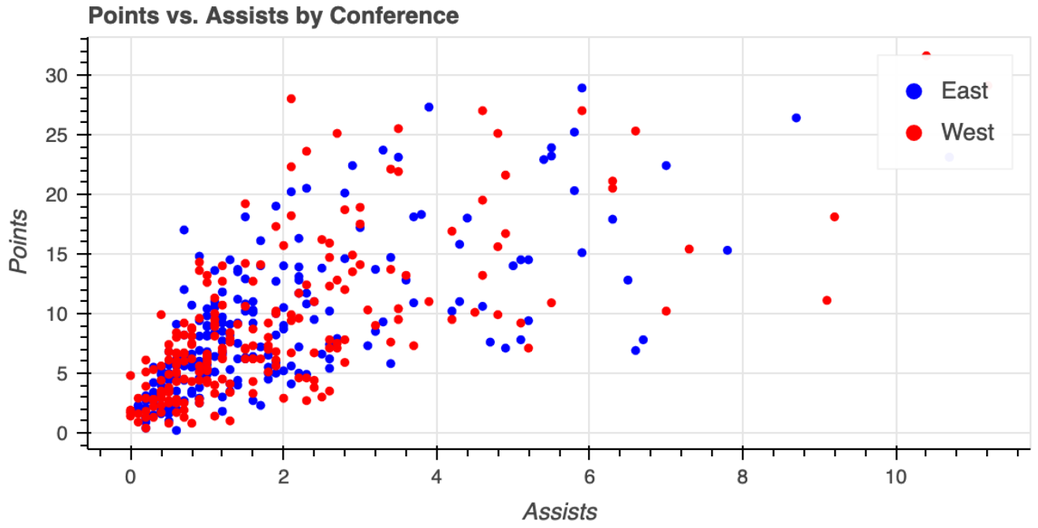
datacamp

Our plot


East
West
Points per Game
Assists per Game
0
5
10
15
20
25
30
0
2
4
6
8
10

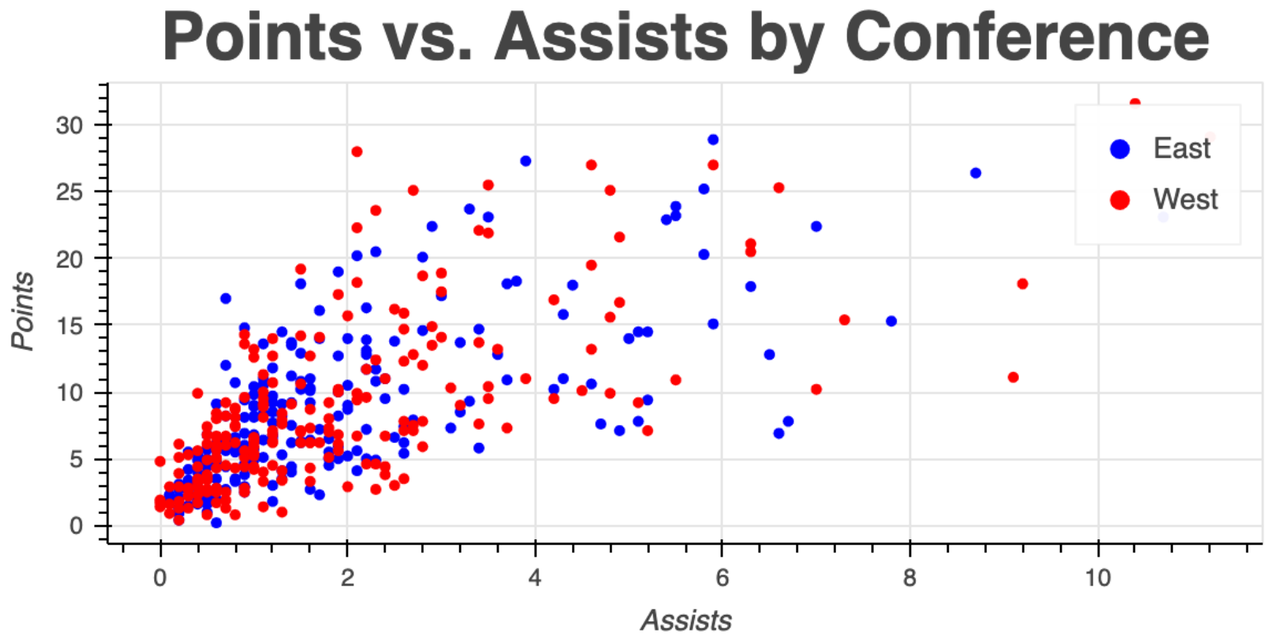
Adding a title

```
fig = figure(x_axis_label='Assists per Game', y_axis_label='Points per Game',
             title="Points vs. Assists by Conference")
fig.circle(x='assists', y='points', source=east, color='blue', legend_label='East')
fig.circle(x='assists', y='points', source=west, color='red', legend_label='West')
output_file(filename="points_vs_assists_by_conference.html")
show(fig)
```



Customizing the title

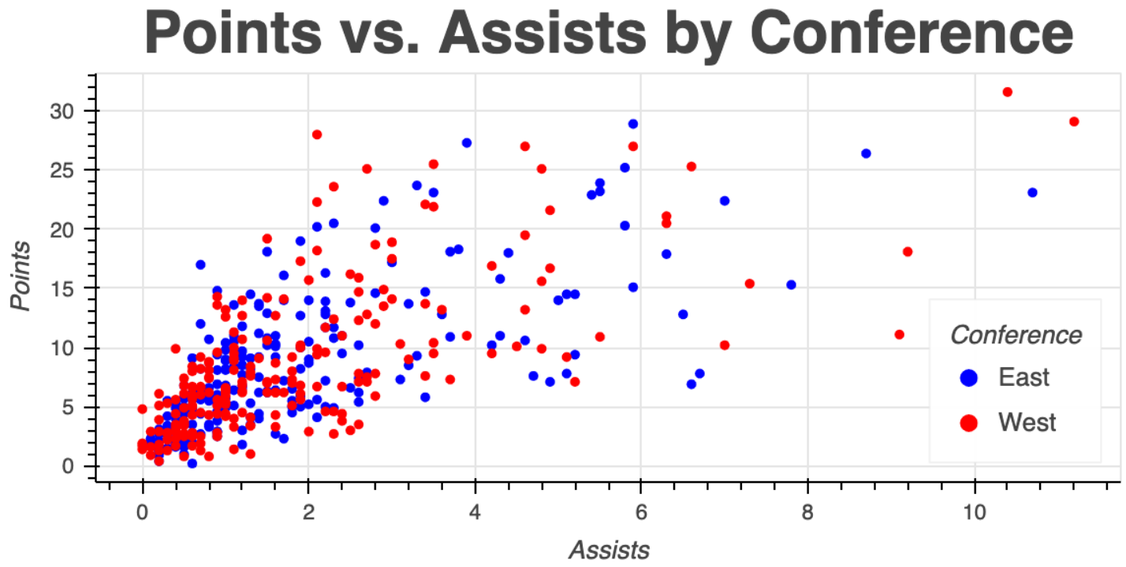
```
fig.title.text_font_size = "30px"
fig.title.align = "center"
output_file(filename="modified_title.html")
show(fig)
```



Modifying the legend

```
fig.legend.title = "Conference"
fig.legend.location = "bottom_right"
show(fig)
```

legend.location
"top_left"
"top_right"
"bottom_left"
"bottom_right"

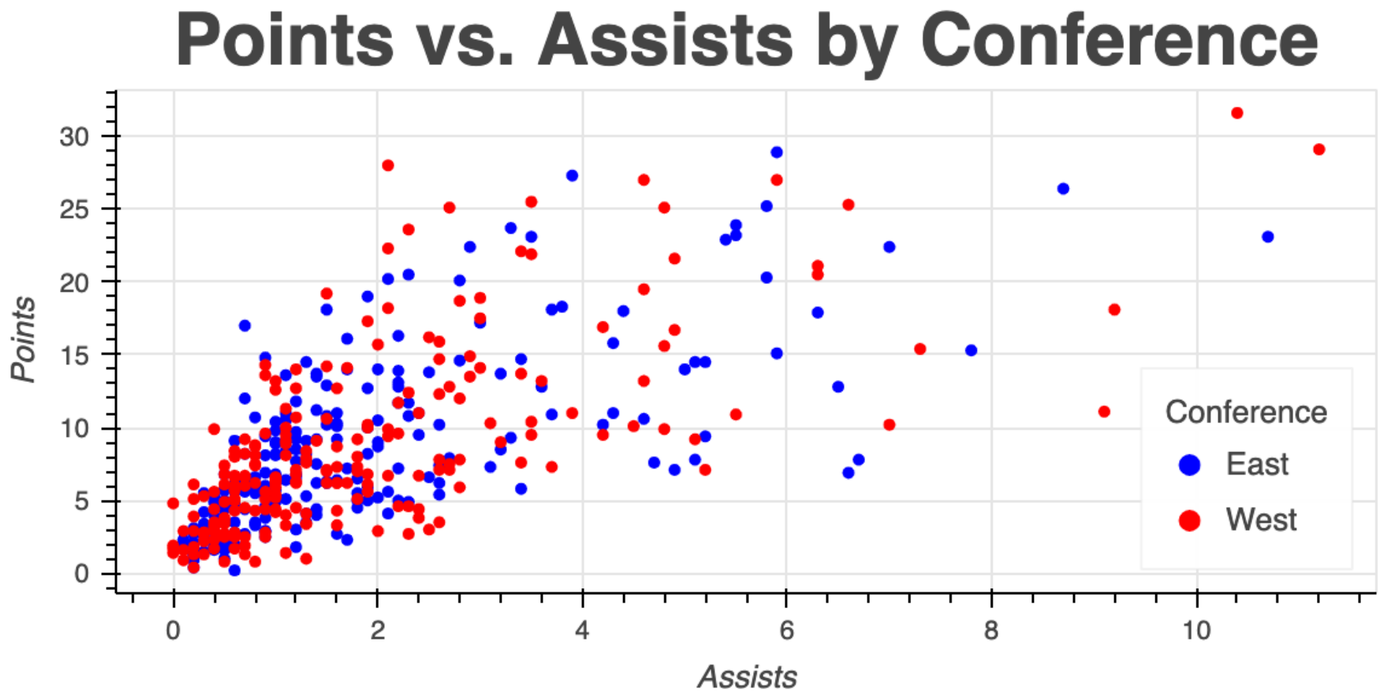


Legend title's font style

legend.title_text_font_style
"bold"
"normal"
"italic"

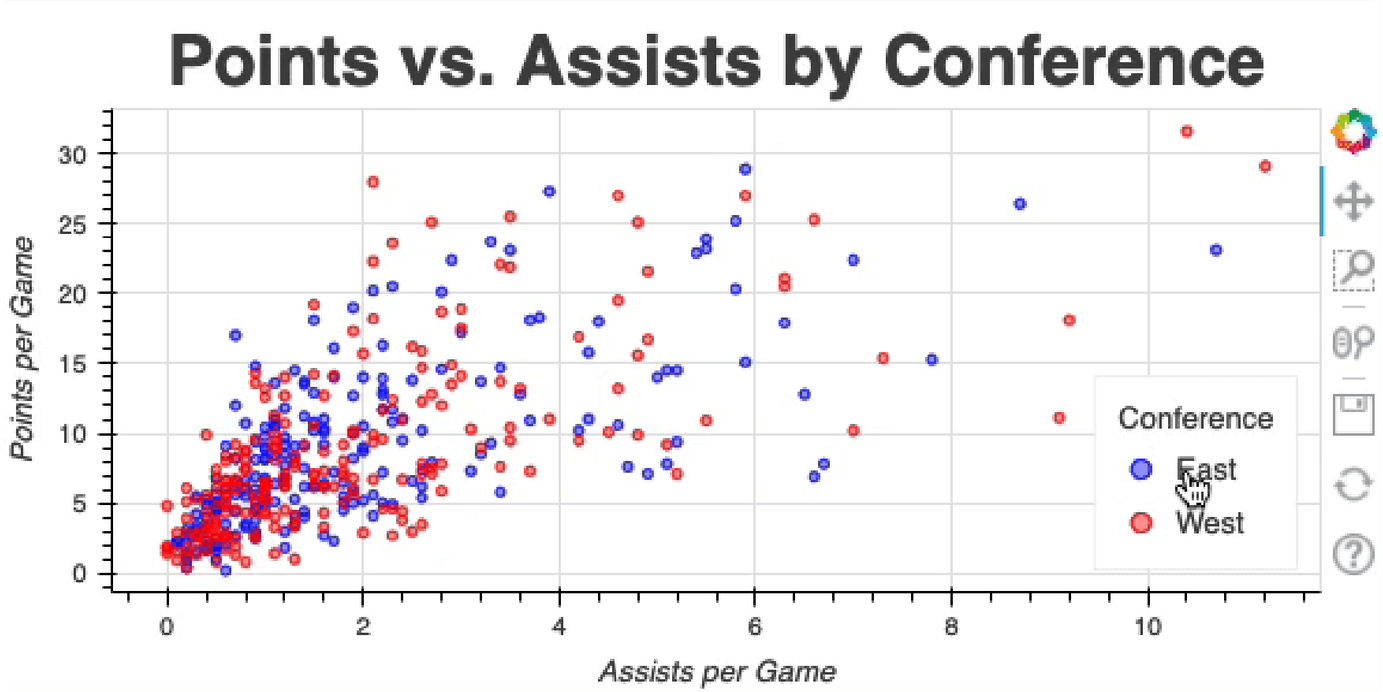
Legend title font style

```
fig.legend.title_text_font_style = "normal"
output_file(filename="normal_legend_title.html")
show(fig)
```



Displaying an interactive legend

```
fig.legend.click_policy = "hide"
output_file(filename="interactive_legend.html")
show(fig)
```



The dataset

```
print(bakery.shape)
```

```
(17486, 6)
```

```
print(bakery.columns)
```

```
Index(['transaction', 'items', 'day_time', 'day_type', 'date', 'sales'],
dtype='object')
```

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

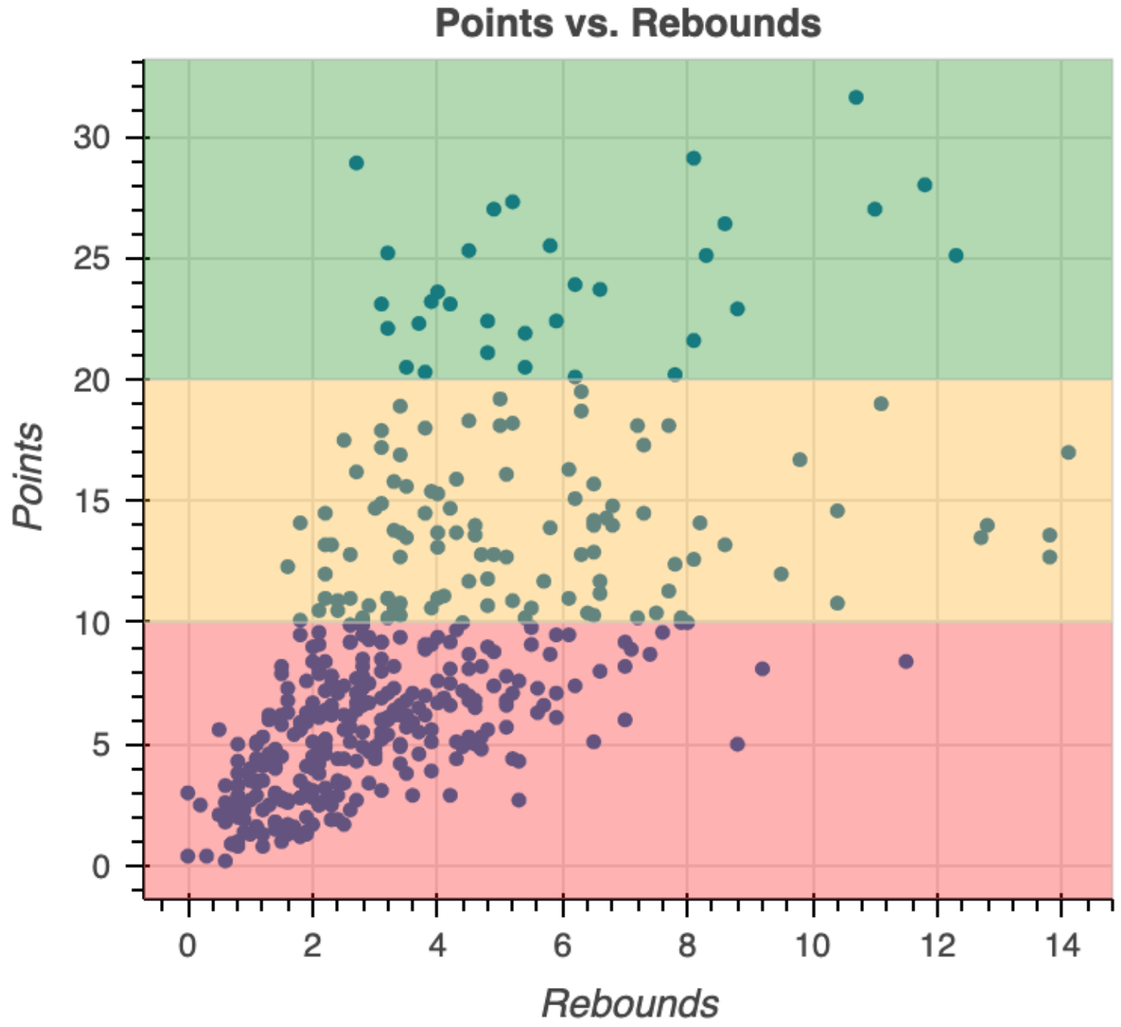
Adding annotations

INTERACTIVE DATA VISUALIZATION WITH BOKEH

George Boorman
Core Curriculum Manager, DataCamp

datacamp

Box annotation



Adding box annotations

```
from bokeh.models import BoxAnnotation
fig = figure(x_axis_label="Rebounds", y_axis_label="Points", title="Points vs. Rebounds")
fig.circle(x="rebounds", y="points", source=source)
low_box = BoxAnnotation(top=10, fill_color="red", fill_alpha=0.3)
mid_box = BoxAnnotation(bottom=10, top=20, fill_color="orange", fill_alpha=0.3)
high_box = BoxAnnotation(bottom=20, fill_color="green", fill_alpha=0.3)
fig.add_layout(low_box)
fig.add_layout(mid_box)
fig.add_layout(high_box)
fig.title.align = "center"
output_file("color_annotated_plot.html")
show(fig)
```

Polygon annotation

Adding a polygon annotation

```
import datetime as dt
from bokeh.models import PolyAnnotation
fig = figure(title="Nvidia Stock Price", x_axis_label="Date", y_axis_label="Price")
fig.line(x="date", y="close", source=source)
fig.xaxis[0].formatter = DatetimeTickFormatter(months="%b %Y")
fig.yaxis[0].formatter = NumeralTickFormatter(format="$0")
start_date = dt.datetime(2017, 5, 9)
end_date = dt.datetime(2017, 6, 12)
start_float = start_date.timestamp() * 1000
end_float = end_date.timestamp() * 1000
start_data = nvidia.loc[nvidia['date'] == start_date]['close'].values[0]
end_data = nvidia.loc[nvidia['date'] == end_date]['close'].values[0]
polygon = PolyAnnotation(fill_color="green", fill_alpha=0.4,
                         xs = [start_float, start_float, end_float, end_float],
                         ys = [start_data-8, start_data+28, end_data+20, end_data-10])
fig.add_layout(polygon)
output_file(filename="nvidia_polygon_annotation.html")
show(fig)
```

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH