## Case Study 4 Homework: Exercises 1-3

Exercise 2

1/1 point (graded)

In Exercise 2, we will create the names and colors we will use to plot the correlation matrix of whisky flavors. Later, we will also use these colors to plot each distillery geographically.

### Instructions

- Create a dictionary `region_colors` with `regions` as keys and `cluster_colors` as values.
- Print `region_colors`.

Use this code to get started:

```
cluster_colors = ['#0173b2', '#de8f05', '#029e73', '#d55e00', '#cc78bc', '#ca9161']
regions = ["Speyside", "Highlands", "Lowlands", "Islands", "Campbelltown", "Islay"]

region_colors = ## ENTER CODE HERE! ##
```

What color is associated with the key `"Campbelltown"` in the `region_colors` dictionary?

Answer = [#cc78bc]

Code = [

```
egion_colors = dict(zip(regions, cluster_colors))

print(region_colors)
{'Speyside': '#0173b2', 'Highlands': '#de8f05', 'Lowlands': '#029e73', 'Islands': '#d55e00', 'Campbelltown': '#cc78bc', 'Islay': '#ca9161'}.   ]
```

## Case Study 4 Homework: Exercises 4-7

Exercise 4

0/1 point (graded)

In Exercise 4, we will edit the given code to make an interactive grid of the correlations among distillery pairs based on the quantities found in previous exercises. Most plotting specifications are made by editing `ColumnDataSource`, a `bokeh` structure used for defining interactive plotting inputs. The rest of the plotting code is already complete.

### Instructions

- `correlation_colors` is a list of string colors for each pair of distilleries. Set this as `color` in `ColumnDataSource`.
- Define `correlations` in `source` using `correlations` fr om the previous exercise. To convert `correlations` from a `np.array` to a `list`, use the `flatten()` method. This correlation coefficient will be used to define both the color transparency as well as the hover text for each square.

Here is the code to get started:

```
source = ColumnDataSource(
    data = {
        "x": np.repeat(distilleries,len(distilleries)),
        "y": list(distilleries)*len(distilleries),
        "colors": ## ENTER CODE HERE! ##,
        "correlations": ## ENTER CODE HERE! ##,
```

```
    }
)

output_file("Whisky Correlations.html", title="Whisky
Correlations")
fig = figure(title="Whisky Correlations",
    x_axis_location="above",
x_range=list(reversed(distilleries)),
y_range=distilleries,
    tools="hover,box_zoom,reset")
fig.grid.grid_line_color = None
fig.axis.axis_line_color = None
fig.axis.major_tick_line_color = None
fig.axis.major_label_text_font_size = "5pt"
fig.xaxis.major_label_orientation = np.pi / 3
fig.rect('x', 'y', .9, .9, source=source,
     color='colors', alpha='correlations')
hover = fig.select(dict(type=HoverTool))
hover.tooltips = {
    "Whiskies": "@x, @y",
    "Correlation": "@correlations",
}
show(fig)
```

What color is the bottom right corner of the whisky correlation matrix?

pink

orange

blue
correct

green

lightgrey

white

Code = [

```
 "colors": correlation_colors,
"correlations": correlations.flatten(),
]
```

Exercise 6

1/1 point (graded)

In Exercise 6, we will define a function `location_plot(title, colors)` that takes a string `title` and a list of colors corresponding to each distillery and outputs a Bokeh plot of each distillery by latitude and longitude. It will also display the distillery name, latitude, and longitude as hover text.

**Instructions**

- Adapt the given code beginning with the first comment and ending with `show(fig)` to create the function `location_plot()`, as described above.
- `Region` is a column of in the `pandas` dataframe `whisky`, containing the regional group membership for each distillery. Make a list consisting of the value of `region_colors` for each distillery, and store this list as `region_cols`.

- Use `location_plot` to plot each distillery, colored by its regional grouping.

Here is the code you will edit to do this exercise:

```
# edit this to make the function `location_plot`.

output_file(title+".html")
location_source = ColumnDataSource(
    data = {
        "x": whisky[" Latitude"],
        "y": whisky[" Longitude"],
        "colors": colors,
        "regions": whisky.Region,
        "distilleries": whisky.Distillery
    }
)

fig = figure(title = title,
    x_axis_location = "above", tools="hover, save")
fig.plot_width  = 400
fig.plot_height = 500
fig.circle("x", "y", size=9, source=location_source,
color='colors', line_color = None)
fig.xaxis.major_label_orientation = np.pi / 3
hover = fig.select(dict(type = HoverTool))
hover.tooltips = {
    "Distillery": "@distilleries",
    "Location": "(@x, @y)"
}
show(fig)

region_cols = ## ENTER CODE HERE! ##
```

```
location_plot("Whisky Locations and Regions",
region_cols)
```

Consider the bottom green point of the given plot. What is the distillery associated with this point?

Answer = [Bladnoch]

Code = [

```
def location_plot(title, colors):
    output_file(title+".html")
    location_source = ColumnDataSource(
        data = {
            "x": whisky[" Latitude"],
            "y": whisky[" Longitude"],
            "colors": colors,
            "regions": whisky.Region,
            "distilleries": whisky.Distillery
        }
    )

    fig = figure(title = title,
        x_axis_location = "above", tools="hover, save")
    fig.plot_width  = 400
    fig.plot_height = 500
    fig.circle("x", "y", size=9, source=location_source,
         color='colors', line_color = None)
    fig.xaxis.major_label_orientation = np.pi / 3
    hover = fig.select(dict(type = HoverTool))
    hover.tooltips = {
        "Distillery": "@distilleries",
        "Location": "(@x, @y)"
    }
    show(fig)

region_cols = [region_colors[i] for i in list(whisky["Region"])]
location_plot("Whisky Locations and Regions", region_cols)
```

]

## Exercise 7

0/1 point (graded)

In Exercise 7, we will use this function to plot each distillery, colored by region and taste coclustering classification, respectively.

**Instructions**

- Create the list `region_cols` consisting of the color in `region_colors` that corresponds to each whisky in `whisky.Region`.
- Similarly, create a list `classification_cols` consisting of the color in `cluster_colors` that corresponds to each cluster membership in `whisky.Group`.
- Create two interactive plots of distilleries, one using `region_cols` and the other with colors defined by `classification_cols`. Consider how well the coclustering groupings match the regional groupings.

Here is the code to edit to make the plot:

```
region_cols = ## ENTER CODE HERE! ##
classification_cols = ## ENTER CODE HERE! ##

location_plot("Whisky Locations and Regions",
region_cols)
location_plot("Whisky Locations and Groups",
classification_cols)
```

Consider the `classification_cols` plot. What color is the point associated with the distillery Bladnoch?

pink

orange

blue
Correct

green

lightgrey

white