Jhirsc21

February 11, 2024

[1]:
```python
import warnings
warnings.filterwarnings('ignore')
```

1 Lab 3 - Visualization

[2]:
```python
%matplotlib inline
```

The special command above will make all the `matplotlib` images appear in the notebook.

1.1 General Instructions

In this course, Labs are the chance to applying concepts and methods discussed in the module. They are a low stakes (pass/fail) opportunity for you to try your hand at *doing*. Please make sure you follow the general Lab instructions, described in the Syllabus. The summary is:

- Discussions should start as students work through the material, first Wednesday at the start of the new Module week.
- Labs are due by Sunday.
- Lab solutions are released Monday.

- Post Self Evaluation and Lab to Lab Group on Blackboard and Lab to Module on Blackboard on Monday.

The last part is important because the Problem Sets will require you to perform the same or similar tasks without guidance. Problem Sets are your opportunity to demonstrate that you understand how to apply the concepts and methods discussed in the relevant Modules and Labs.

1.2 Specific Instructions

1. For Blackboard submissions, if there are no accompanying files, you should submit *only* your notebook and it should be named using *only* your JHED id: fsmith79.ipynb for example if your JHED id were "fsmith79". If the assignment requires additional files, you should name the *folder/directory* your JHED id and put all items in that folder/directory, ZIP it up (only ZIP...no other compression), and submit it to Blackboard.

   - do **not** use absolute paths in your notebooks. All resources should located in the same directory as the rest of your assignments.
   - the directory **must** be named your JHED id and **only** your JHED id.
   - do **not** return files provided by us (data files, .py files)

2. Data Science is as much about what you write (communicating) as the code you execute (researching). In many places, you will be required to execute code and discuss both the purpose and the result. Additionally, Data Science is about reproducibility and transparency. This includes good communication with your team and possibly with yourself. Therefore, you must show **all** work.

3. Avail yourself of the Markdown/Codecell nature of the notebook. If you don't know about Markdown, look it up. Your notebooks should not look like ransom notes. Don't make everything bold. Clearly indicate what question you are answering.

4. Submit a cleanly executed notebook. The first code cell should say `In [1]` and each successive code cell should increase by 1 throughout the notebook.

There are only 4 charts because we expect depth rather than breadth

Do not use any tables. Only use charts

[3]:
```
import numpy as np
import random as py_random
import numpy.random as np_random
import time
import seaborn as sns
import matplotlib.pyplot as plt

sns.set(style="whitegrid")

from IPython.display import YouTubeVideo, Image
```

[4]:
```
THEME = "darkslategray"
```

```
<strong>Note</strong>
<p>Defining your main color in a variable makes it easier when your boss says, "can you make t
```

1.3 Data Visualization

Here are four (4) Charts we want you to analyze. There are only four charts because we want you to be thorough and, based on the materials from the Module, sophisticated in your analysis. You should not answer now nor on any future assignment, "because pie charts are bad" or "because we've been told not to use pie charts". Please make sure you check the Solution on Monday and compare it to your own.

Some of these charts work as "infographics" but we want you to evaluate them as if they were slides to be presented to the CEO, that is, a data science presentation. (Those kinds of slides don't find their way onto Google).

1.3.1 Chart 1

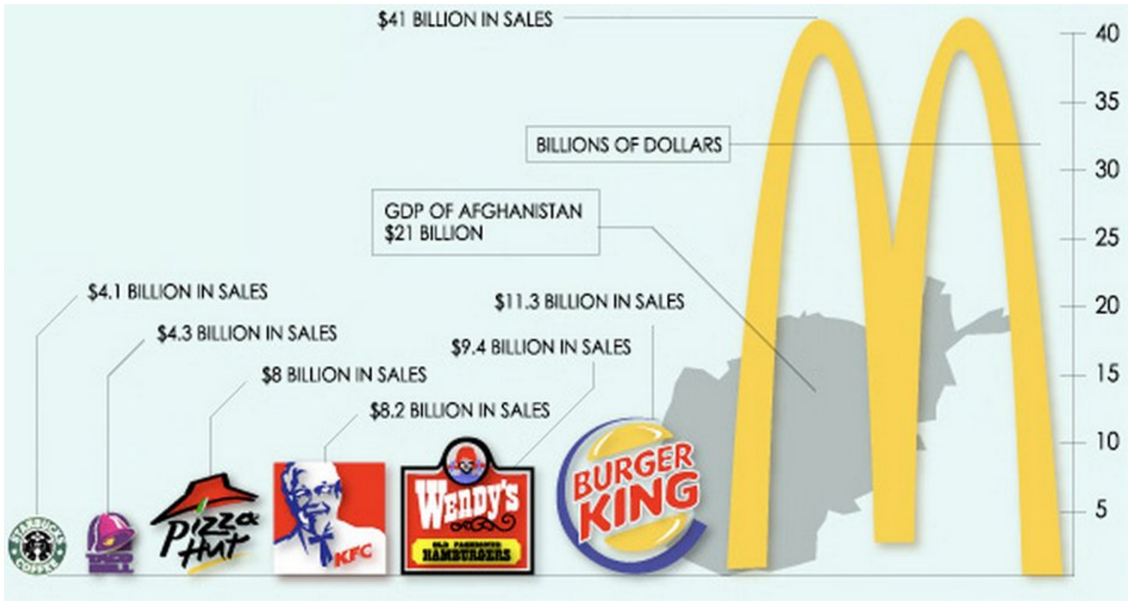
1. What do you think the main story of the chart is? Does the chart really tell it? Identify any secondary stories.
2. Discuss how the chart either adheres to or violates the principles discussed in the module (Cleveland, Ware, and Gestalt).

3. Discuss at least two alternative charts that follow the guidelines presented in the module.
4. Create one of those alternatives by decoding the values as best you as you can and using `matplotlib` (you may use `seaborn` only to change the formatting).

You may insert as many Markdown and Code cells below each chart as you need.

```
[5]: Image( "resources/chart_01.jpg", width=500)
```

[5]:



1. I think the main story of this chart is to compare fast food franchise sales, and show which companies lead the market by doing so. The chart does tell this story decent enough, showing that Mcdonald's has the most sales, while Burger King and Wendy's are not too far behind. The secondary story may be comparing fast food sales to the GDP of Afghanistan, to show how big these corporations really are.

2. I think this chart follows Cleveland's notion of `Position on a common scale`. We can clearly see how each logo starts at the same height from the bottom. Speaking of logos, we can talk about Ware's `Attributes of form`. I guess `shape` would be the closest attribute, but using familiar logos that most people would recognize is a good idea just if one views the chart at a glance. There is also `size`, as the companies with higher sales have larger logos in the chart. I don't see many of Gestalt's Principles noted here, but you could argue the sales numbers are `connected` to each logo, showing `Connection`.

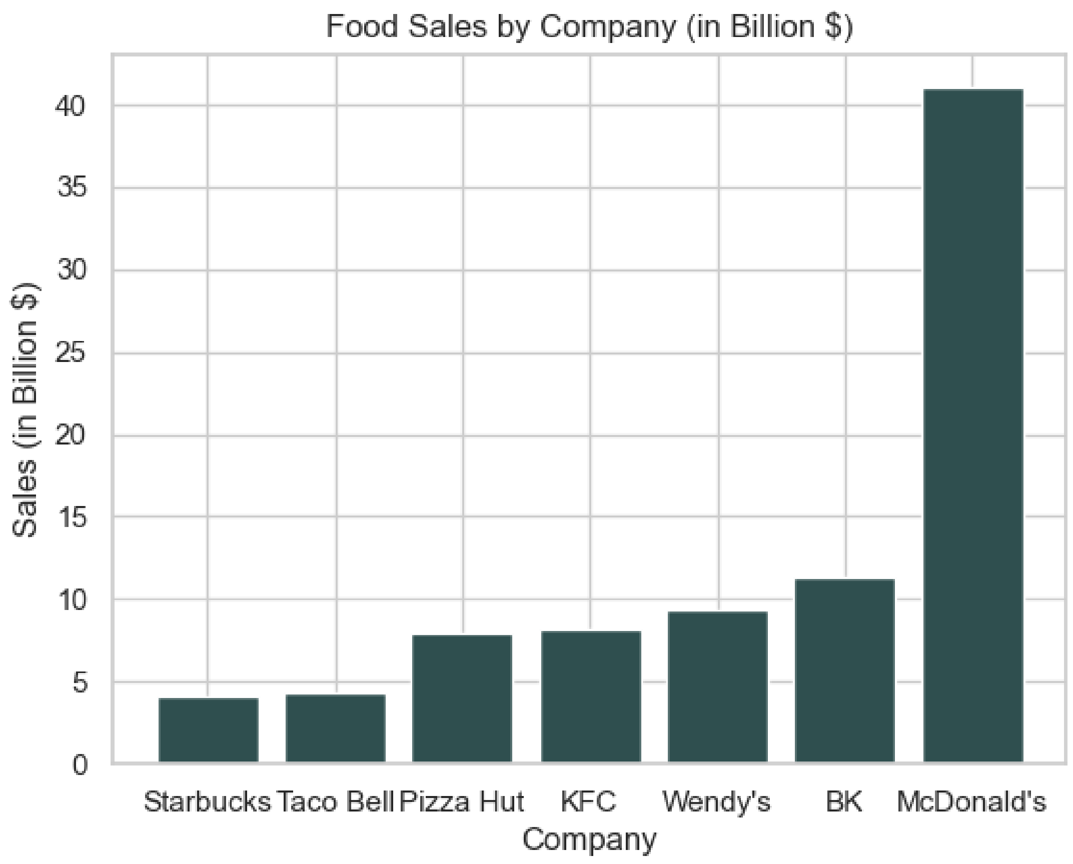
Here's what the chart does wrong: I think the y-axis title should be on the y-axis, not connected with a line and shown in the middle of the chart. I don't think the GDP of Afghanistan is appropriate to show in this chart, as it creates a separate message than just comparing sales numbers from the different companies. There's no title. Is this sales for one month, six months, one year? Who knows, we certainly don't. No labels for each company on the x-axis. If I'm not familiar with these logos, how do I know what company each sales figure is for? Like I said earlier, I don't think the logos are necessarily a bad idea, but they should definitely be accompanied by a label.

3. One alternative could be a vertical bar chart. The x-axis would be labeled so each bar would represent one company, and each bar would be uniform width. The sales can still be ranked in ascending order, as in the original. We could also think about using a point or dot chart, for the same reasons as a vertical bar chart.

4. I will do my best to recreate a vertical bar chart here

[5]:
```
brands = np.array(['Starbucks', 'Taco Bell', 'Pizza Hut', 'KFC', 'Wendy\'s',
                    'BK', 'McDonald\'s' ])
sales = np.array([4.1, 4.3, 8.0, 8.2, 9.4, 11.3, 41.0])

plt.bar(brands,sales, color=THEME)
plt.xlabel('Company')
plt.ylabel('Sales (in Billion $)')
plt.title('Food Sales by Company (in Billion $)')
```

[5]:
```
Text(0.5, 1.0, 'Food Sales by Company (in Billion $)')
```



Here it is much easier to see that Starbucks and Taco Bell are very close in sales, while McDonald's leads the charge by a significant margin. Note I did not put the GDP of Afghanistan on this chart, as I believe that to be a separate message.

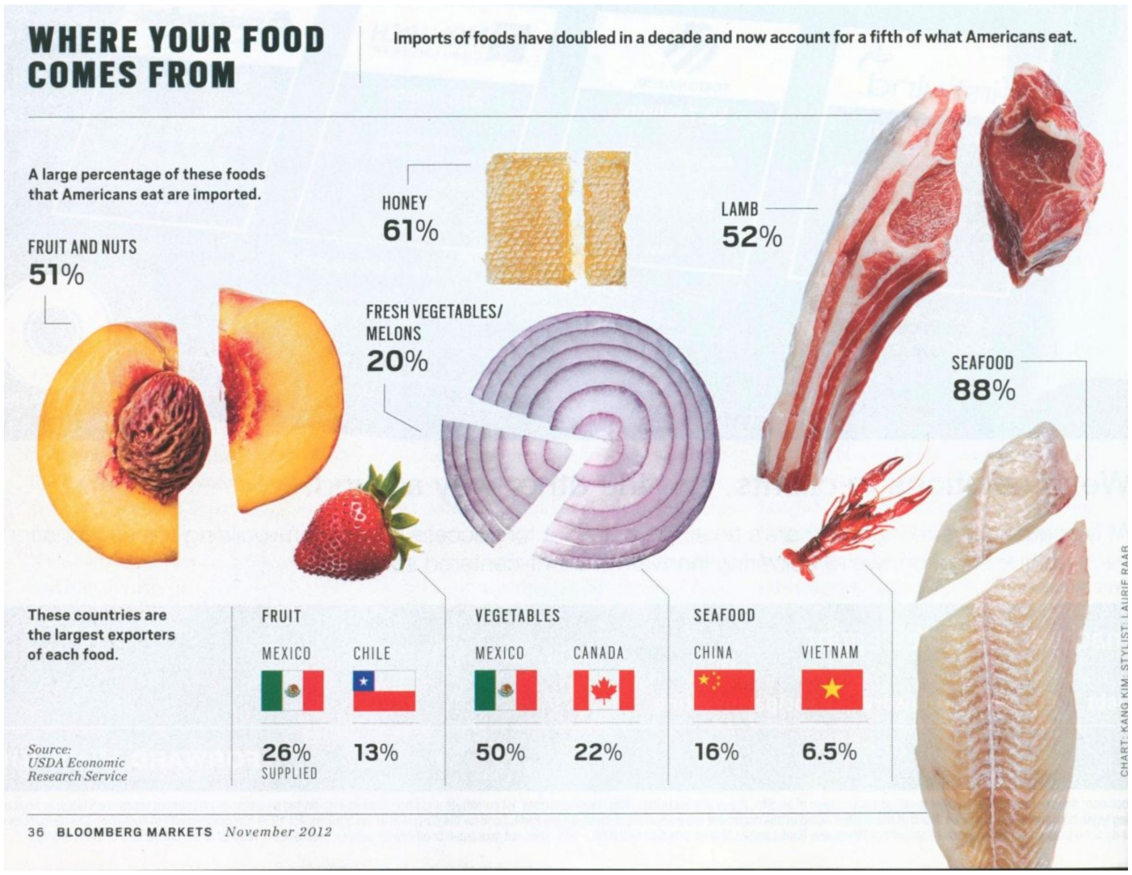
1.3.2 Chart 2

1. What do you think the main story of the chart is? Does the chart really tell it? Identify any secondary stories.
2. Discuss how the chart either adheres to or violates the principles discussed in the module (Cleveland, Ware, and Gestalt).
3. Discuss at least two alternative charts that follow the guidelines presented in the module.
4. Create one of those alternatives by decoding the values as best you as you can and using `matplotlib` (you may use `seaborn` only to change the formatting).

You may insert as many Markdown and Code cells below each chart as you need. You may insert as many Markdown and Code cells below each chart as you need.

```
[6]: Image( "resources/chart_02.jpg", width=500)
```

[6]:



1. I think the main story of this chart is to show the percentage of food that Americans eat are imported, broken down by different categories. I think the chart does tell it, but there are definitely too many messages in this chart which can be confusing to decode. It definitely takes longer to decode than should be necessary. The secondary story is showing which countries are the largest exporter of each food category.

2. Where to begin. Yikes. The secondary story has a common scale at least, but the images of

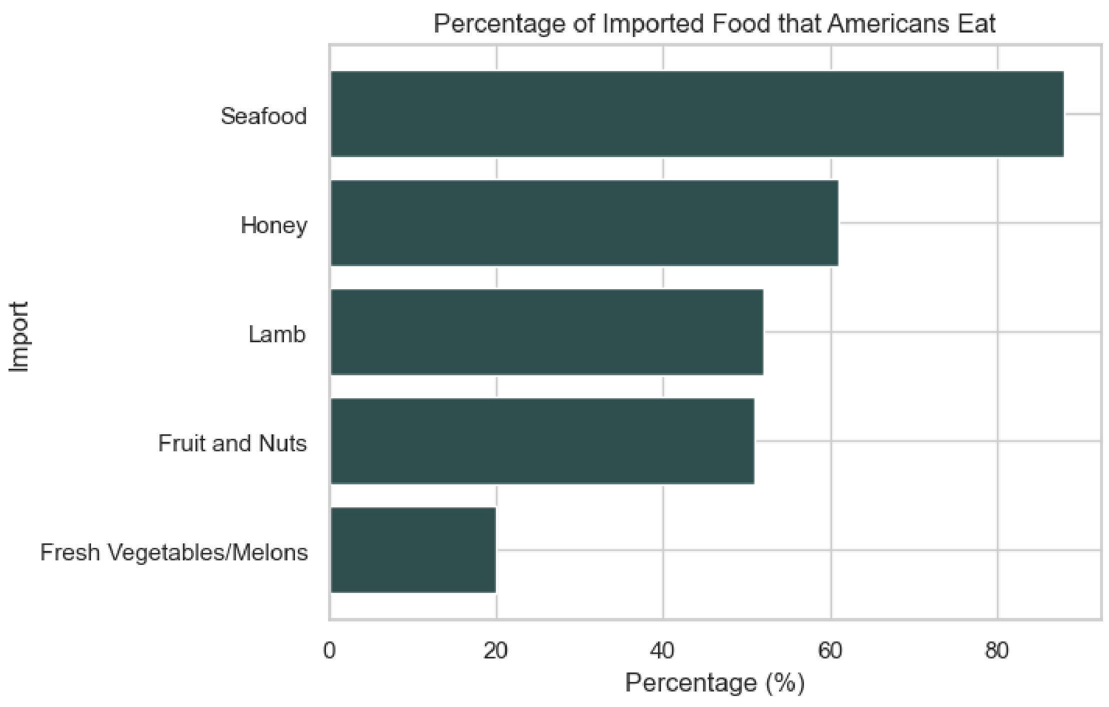
the food are all over the place; the eyes are led to wander, and it's difficult to know where you should start reading the chart from. Additionally, the percentages are not listed in any ascending/descending order, they seem to be placed in the chart at random. Why does the `Fresh Vegetables/Melons` category show and Onion, and not, ya know, a Melon? I guess we can say that `shape` of the food helps to differentiate each category, and `connection` is used from Gestalt's principles to connect each image with its percentage.

3. I think a vertical or horizontal bar chart would be good here, since we are trying to rank and compare each category again. We want to create a uniform bar so that it is easier to compare each category - not, whatever this is. Someone got paid to make this chart. Think about it. We could also consider a point/dot chart if we are so inclined.

4. Let's try a horizontal bar chart here, since it will be easier to see each food category written out (it gets too squished in a vertical bar chart).

[6]:
```python
imported_foods = np.array(['Fresh Vegetables/Melons', 'Fruit and Nuts',
                           'Lamb', 'Honey', 'Seafood',])
percentages = np.array([20, 51, 52, 61, 88])

plt.barh(imported_foods, percentages, color=THEME)
plt.xlabel('Percentage (%)')
plt.ylabel('Import')
plt.title('Percentage of Imported Food that Americans Eat')
```

[6]:
```
Text(0.5, 1.0, 'Percentage of Imported Food that Americans Eat')
```

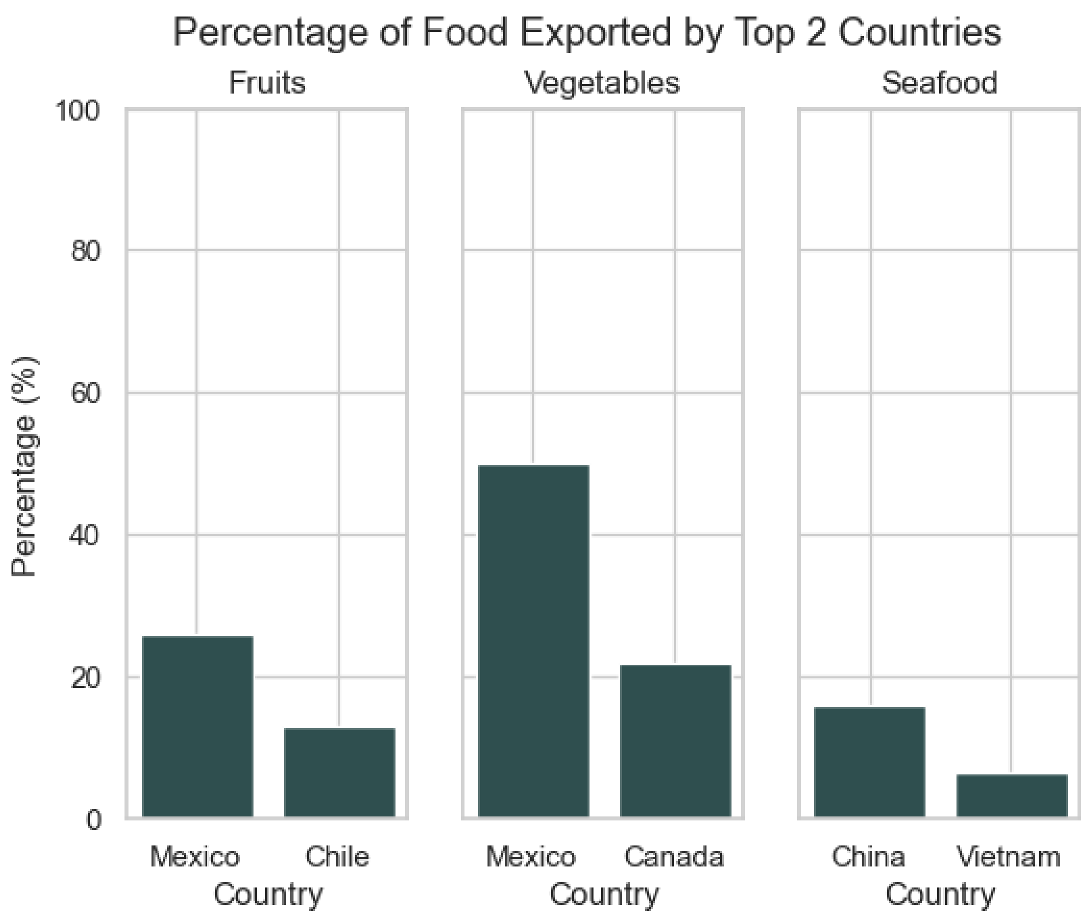


Much easier to decode in my opinion, we can clearly see how each category ranks when we have a common axis. We can see that `Fruit and Nuts` and `Lamb` are imported and eaten about the same percentage, which is harder to see in the original since they are on opposite sides of the chart.

We can also try small multiples to look at which countries export each category, as shown in the original chart.

[7]:
```
foods = np.array(['Fruit', 'Vegetables', 'Seafood'])

fruit_exports = np.array(['Mexico', 'Chile'])
fruit_export_percents = np.array([26, 13])

vegetable_exports = np.array(['Mexico', 'Canada'])
vegetable_export_percents = np.array([50, 22])

seafood_exports = np.array(['China', 'Vietnam'])
seafood_export_percents = np.array([16, 6.5])

fig, ax = plt.subplots(1,3, sharey=True)
fig.suptitle('Percentage of Food Exported by Top 2 Countries')
ax[0].bar(fruit_exports, fruit_export_percents, color=THEME)
ax[0].set_title('Fruits')
ax[0].set_ylim([0,100])
ax[0].set_ylabel('Percentage (%)')
ax[0].set_xlabel('Country')

ax[1].bar(vegetable_exports, vegetable_export_percents, color=THEME)
ax[1].set_title('Vegetables')
ax[1].set_xlabel('Country')

ax[2].bar(seafood_exports, seafood_export_percents, color=THEME)
ax[2].set_title('Seafood')
ax[2].set_xlabel('Country')
```

[7]:
```
Text(0.5, 0, 'Country')
```



Maybe not the best chart, but I think it does a better job at converying the information presented in the original. We can see that Vegetable exports from Mexico are roughly twice that of Canada. This is much easier to see on a bar chart like this, rather than looking at simply the percentage numbers, as in the original chart.

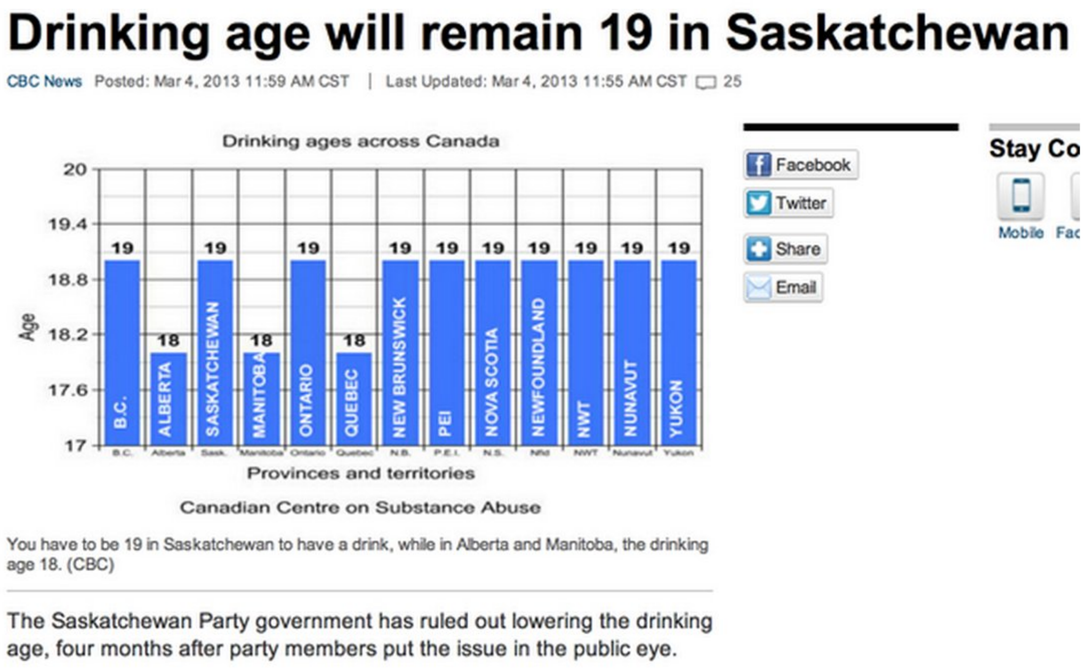
1.3.3 Chart 3

1. What do you think the main story of the chart is? Does the chart really tell it? Identify any secondary stories.
2. Discuss how the chart either adheres to or violates the principles discussed in the module (Cleveland, Ware, and Gestalt).
3. Discuss at least two alternative charts that follow the guidelines presented in the module.
4. Create one of those alternatives by decoding the values as best you as you can and using `matplotlib` (you may use `seaborn` only to change the formatting).

You may insert as many Markdown and Code cells below each chart as you need.

```
[7]: Image( "resources/chart_03.jpg", width=500)
```

```
[7]:
```



Drinking age will remain 19 in Saskatchewan

CBC News Posted: Mar 4, 2013 11:59 AM CST | Last Updated: Mar 4, 2013 11:55 AM CST 25

You have to be 19 in Saskatchewan to have a drink, while in Alberta and Manitoba, the drinking age 18. (CBC)

The Saskatchewan Party government has ruled out lowering the drinking age, four months after party members put the issue in the public eye.

1. I think the main story of the chart is to show the legal drinking age in each Canadian province. I mean the article is talking about how the drinking age will remain at 19 in Saskatchewan, so I'm not even sure a chart was needed here. I think it does tell this story, but poorly as we've seen from the previous examples.

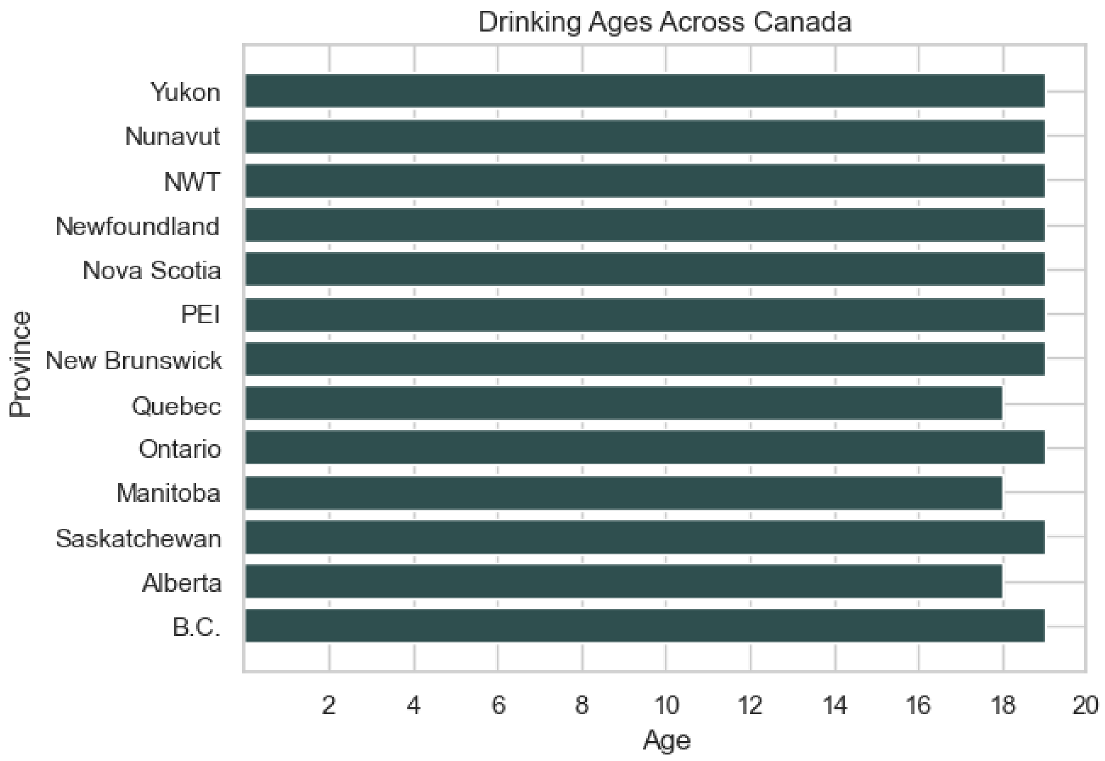
2. First off, the y-axis `Age` scale doesn't make sense. `Age` should be listed as whole numbers, each tick should increase by 1, not 0.6. Normally we should start bar charts at zero as well, not randomly at 17, although we understand why this choice was made - there will be no bar heights under 17, so the chart is able to fit better into this article. I do like vertical bar charts though. This chart was positioned on a common scale, and the length of each bar makes sense to show the age in each province. This chart does follow Ware's attributes of Length and Shape (consistent bars for each province). It also follows Gestalt's idea of Similarity and consistency for each bar. I don't think the numbers should be shown atop each bar however, and we can make a case for each province being listed vertically on each bar, and how that makes it a little harder to read - especially when the provinces are already listed on the x-axis. However, I'm going to choose to believe that these choices make it easier to read, when we consider it is a quick thing thrown together for the more important article, most likely. Finally, we can mention that there are only two values for the drinking age, either 18 or 19, and that this chart may be showing too much information that is not necessary. We could also argue the blue color doesn't do the chart any favors either.

3. I don't think a vertical bar chart is bad, the information just needs to be presented in a better way, which I will try to recreate below. Again, I think a dot/point chart would be fine here as well. We are still making comparisons as in the first two examples.

4. I will attempt a horizontal bar chart because of some of the long province names.

```
[8]: provinces = np.array(['B.C.', 'Alberta', 'Saskatchewan', 'Manitoba',
                          'Ontario', 'Quebec', 'New Brunswick', 'PEI',
                          'Nova Scotia', 'Newfoundland', 'NWT', 'Nunavut',
                          'Yukon'])
     ages = np.array([19, 18, 19, 18, 19, 18, 19, 19, 19, 19, 19, 19, 19])

     plt.barh(provinces, ages, color=THEME)
     plt.xlabel('Age')
     plt.ylabel('Province')
     plt.xticks([2,4,6,8,10,12,14,16,18,20])
     plt.title('Drinking Ages Across Canada')
```

```
[8]: Text(0.5, 1.0, 'Drinking Ages Across Canada')
```

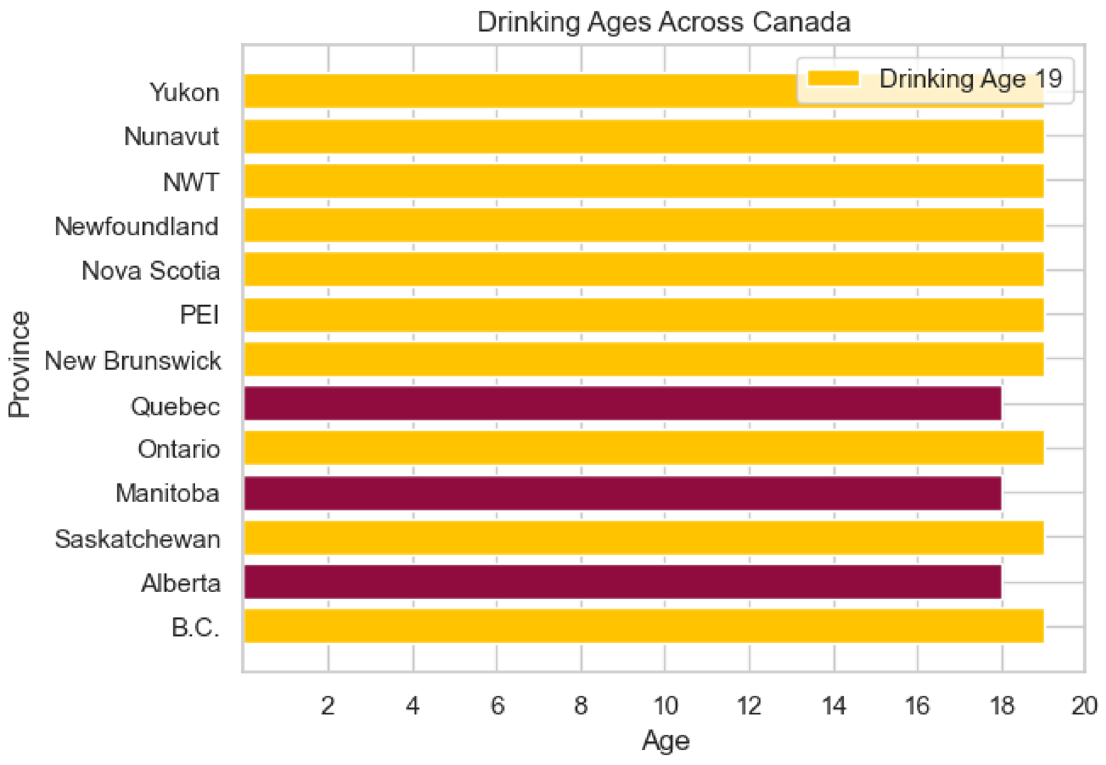


While I think this chart is ‘better’, it still doesn’t do a good job conveying the information appropriately. As I mentioned, we only have 2 values, so a bar chart like this doesn’t really make sense. One thing we can try is to add a bit of color to each province where the dirnking age is the same.

```
[9]: provinces = np.array(['B.C.', 'Alberta', 'Saskatchewan', 'Manitoba',
                          'Ontario', 'Quebec', 'New Brunswick', 'PEI',
                          'Nova Scotia', 'Newfoundland', 'NWT', 'Nunavut',
                          'Yukon'])
     ages = np.array([19, 18, 19, 18, 19, 18, 19, 19, 19, 19, 19, 19, 19])
```

```
colors = ['#FFC300', '#900C3F', '#FFC300', '#900C3F', '#FFC300', '#900C3F',
            '#FFC300', '#FFC300', '#FFC300', '#FFC300', '#FFC300', '#FFC300',␣
 ↪'#FFC300']

plt.barh(provinces, ages, color=colors)
plt.xlabel('Age')
plt.ylabel('Province')
plt.xticks([2,4,6,8,10,12,14,16,18,20])
plt.title('Drinking Ages Across Canada')

#Help
plt.legend(labels=['Drinking Age 19', 'Drinking Age 18'])
```

```
[9]: <matplotlib.legend.Legend at 0x178522f9dd0>
```



I think some color helps here. I couldn't figure out how to create two separate legend labels easily on this one plot, so I'd appreciate some tips on how to do that! Thanks!

1.3.4 Chart 4

1. What do you think the main story of the chart is? Does the chart really tell it? Identify any secondary stories.
2. Discuss how the chart either adheres to or violates the principles discussed in the module

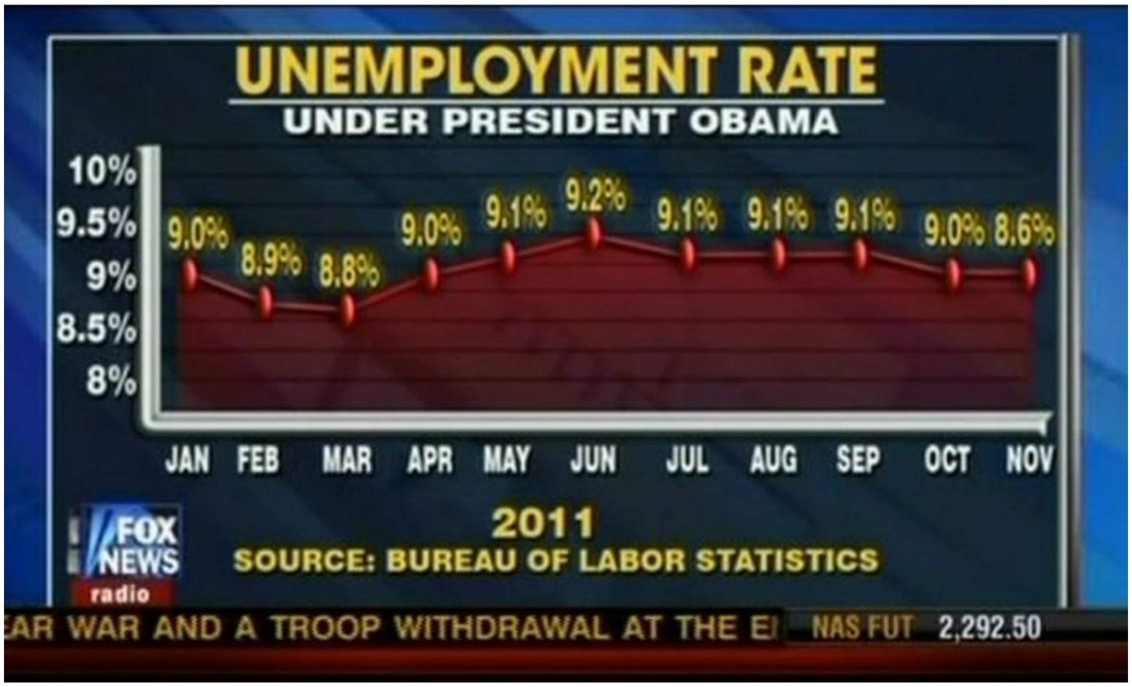
(Cleveland, Ware, and Gestalt).

3. Discuss at least two alternative charts that follow the guidelines presented in the module.
4. Create one of those alternatives by decoding the values as best you as you can and using `matplotlib` (you may use `seaborn` only to change the formatting).

You may insert as many Markdown and Code cells below each chart as you need.

[8]:
```
Image( "resources/chart_06.jpg", width=500)
```

[8]:



1. I think the main story of this chart is to show the unemployment rate in the U.S. in the year 2011, broken down by month. I think this chart show this story well. We can easily see an increase or decrease in the unemployment rate thanks to the connected line chart. The secondary story is about the `Ear War and a Troop Withdrwawal at the En`. Just kidding, I don't really see any secondary story here.

2. Where is December? That's my main complaint, although we don't know when this broadcast was. I assume this was shown in November, in which case, that's okay I suppose. What I think this chart does well is that it shows even spacing/length between each month, and is positioned on a common scale. Each point is the same shape, size, and again the length of the line connecting them is about the same, which according to Ware, is good. As for Gestalt's principles, we see similarity for each point and line, as well as connectively, which for a time series chart makes sense. In this case, I do like that each point is labeled with a percentage, because it would be difficult to eyeball the percentage from the y-axis alone. Not too fond of the color choices but overall I think this the best chart out of all the examples we've seen so far. The x- and y-axis should have labels.

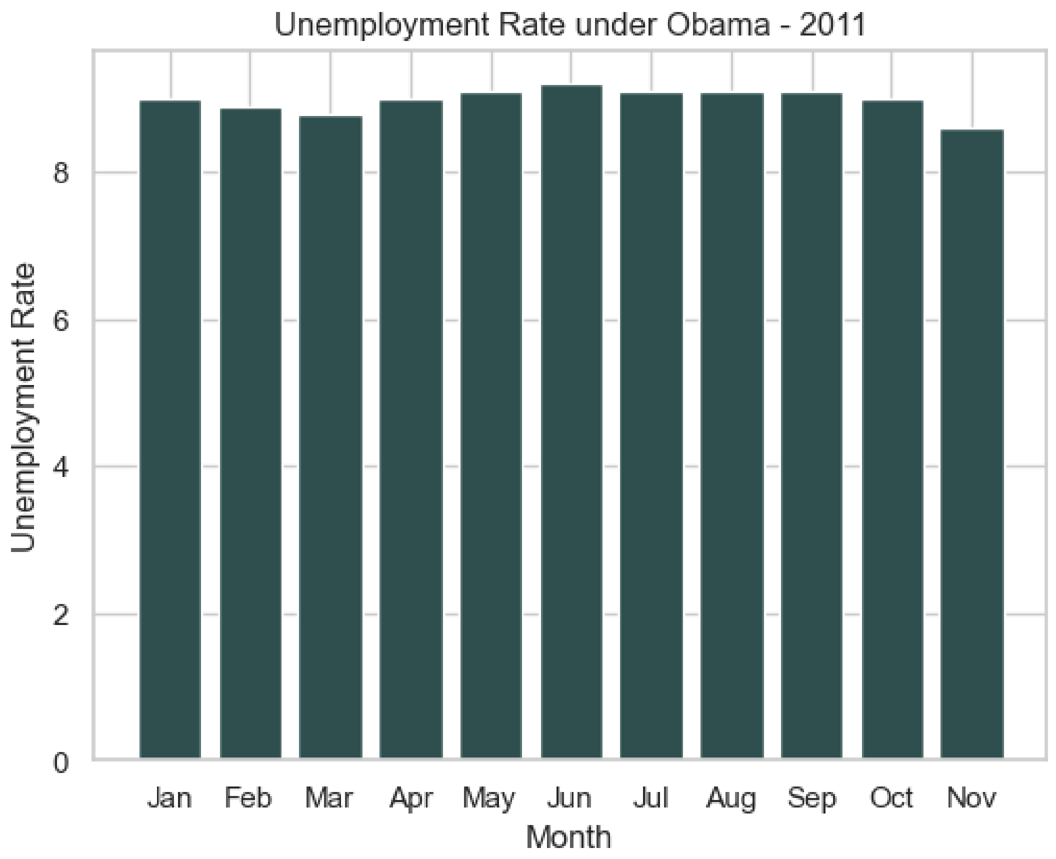
3. I think a point/line chart is good here, but maybe the scaling of the y-axis can be larger so the chart is not as 'squished'. We could also try a bar chart as well for time series. Again, I

don't have too many complaints about this one overall.

4. I will use a vertical bar chart here, since I don't think I can really improve much of a point/line chart from the original.

[10]:
```
months = np.array(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                    'Jul', 'Aug', 'Sep', 'Oct', 'Nov'])
rates = np.array([9.0, 8.9, 8.8, 9.0, 9.1, 9.2, 9.1, 9.1, 9.1, 9.0, 8.6])


plt.bar(months, rates, color=THEME)
plt.xlabel('Month')
plt.ylabel('Unemployment Rate')

plt.title('Unemployment Rate under Obama - 2011')
```

[10]:
```
Text(0.5, 1.0, 'Unemployment Rate under Obama - 2011')
```



Yeah I definitely think point/line chart is the way to go here. I think the bar chart doesn't make much sense when we have values that are so close to each other. I think the original is easier to decode than this vertical bar chart.