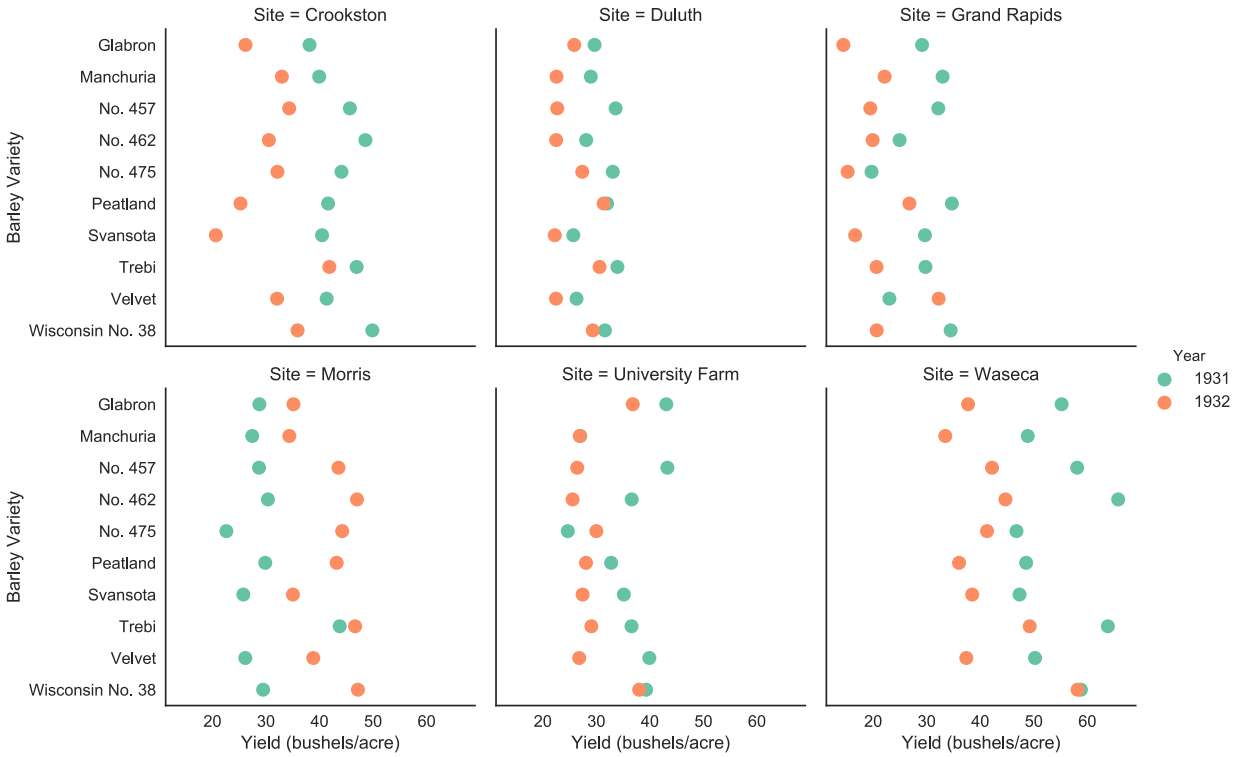

Figure 1: In 1931 and 1932 Minnesota collected data on the yield in bushels per acre of 10 varieties of barley grown in 1/40 acre plots at University Farm, St. Paul, and at the five branch experiment stations located at Waseca, Morris, Crookston, Grand Rapids, and Duluth (all in Minnesota). The varieties were grown in three randomized blocks at each of the six stations during 1931 and 1932, different land being used each year of the test. [1]

Writeup

Because the provided data was four dimensional, I began my design process by doing exploratory analysis to find which dimensions of the data were most interesting. I started by creating scatter plots for various permutations of variables for the horizontal and vertical dimensions. This exploration demonstrated that location had a more drastic effect on yield than variety and resulted in the following ordering based on interest: (1) Yield, (2) Site, (3) Year, and (4) Variety. When plotting yield against location, I noticed an interesting pattern in one specific location across the two years. While every location went down in yield from 1931 to 1932, Morris went up. This motivated me to find a visualization that would highlight this interesting point.

I decided to use a ”small multiple” [Tufte VDQI pg. 42] to enable the viewer to focus on the changes in data across location. By displaying the yield for each year and variety, it clearly shows the trend in each location being similar except for Morris. I found that the use of discrete colors for years highlights the ”flipped” nature of the Morris location effectively compared to other schemes that I tried. One of my final designs had yield on the vertical axis as I thought the ”above”/”below” nature of the vertical axis matched the semantics of comparing yield amounts. However, displaying variety on the horizontal axis required rotating the labels because of the long names. This made the figure more difficult to interpret because the viewer had to match the rotated labels with the floating data points above. Swapping the yield and variety axes resulted in a more visually appealing design with straight labels that was faster to visually parse. I wrapped the ”small-multiple” pattern into two rows so that the visualization would fit neatly on a standard sized laptop screen. If I had made the visualization a single column, it would be easier to compare yield

across all the locations, but wrapping made the visualization compact. Furthermore, wrapping helped the visualization emphasize the different trends between 1931 and 1932 instead of focusing on comparing raw yield numbers. To further emphasize the trends over the raw data points, I chose not to use a grid in the visualization. This reflects Tufte's opinion on maximizing data-ink. However, I chose to keep the lines corresponding to each axis because they provide a nice separation between each plot multiple, even though this leads to more ink in the figure not devoted to data.

To create the visualization, I used the Python package `pandas` to handle converting the `csv` data into a Python a datastructure. `pandas` provides a `read_csv` function that was able to properly parse and load the `barley2.csv` file into a `pandas DataFrame` object. The `seaborn` visualization library provides a nice abstraction for plotting small multiples called `factorplot`. It allowed me to easily experiment with different plot styles as well as changing which variables in the data were associated with plot elements. I found it perfect for rapidly exploring visualization designs without having to worry about the underlying mechanics of rendering the visualizations.

```
import pandas as pd
import seaborn as sns

# Style configuration for seaborn
sns.set(style="whitegrid", color_codes=True, font_scale=1.2)

# Load the csv data into a pandas dataframe
df = pd.read_csv("barley2.csv")

# Seaborn's factorplot provides a nice abstraction for creating small multiples
g = sns.factorplot(x="Yield", y="Variety", hue="Year", col="Site",
                   kind="point", data=df, palette="Set2", col_wrap=3, join=False)

# Improve axis labels
g.set_axis_labels("Yield (bushels/acre)", "Barley Variety")

g.savefig('variety_yield_point.svg')
```

Figure 2: Visualization source code

References

[1] https://magrawala.github.io/cs448b-fa17/a1.html