Data 550: Data Visualization I

Lecture 7: Trendlines and visualizing uncertainty

Dr. Irene Vrbik
University of British Columbia Okanagan

https://github.com/ubco-mds-2022/Data-550

1

Overview

By the end of the lecture you will be able to:

- Visualize pair-wise differences using a slope plot.
- Visualize trends using regression and loess lines.
- Create and understand how to interpret confidence intervals and confidence bands.

Suggested readings from Fundamentals of Data Visualization.

- Section 14 - 14.2 on visualizing trends.
- Section 16 on visualizing uncertainty

https://github.com/ubco-mds-2022/Data-550

Trendlines

https://github.com/ubco-mds-2022/Data-550

Introduction

- It is often the case that we are interested in the overarching trend of the data (rather than the specific values).
- Trends are usually visualized by a straight or curved line.
- These can be layered on top of or instead of the actual data points to help the reader identify key features in the data.
- Once established, we can look at deviations from the trend, or explore separating the data into multiple components

Trendlines

- Trendlines[1] highlight general trends in the data that can be hard to elucidate by looking at the raw data points.
- This can happen if there are many data points or many groups inside the data.
- Two fundamental approaches to determining a trend are:
  1. smoothing (e.g. moving average)
  2. fitting a curve with a functional form (e.g. regression)

https://github.com/ubco-mds-2022/Data-550

1. also sometimes called “lines of best fit”, or “fitted lines”

Cars data

```python
from vega_datasets import data

cars = data.cars()
cars
```

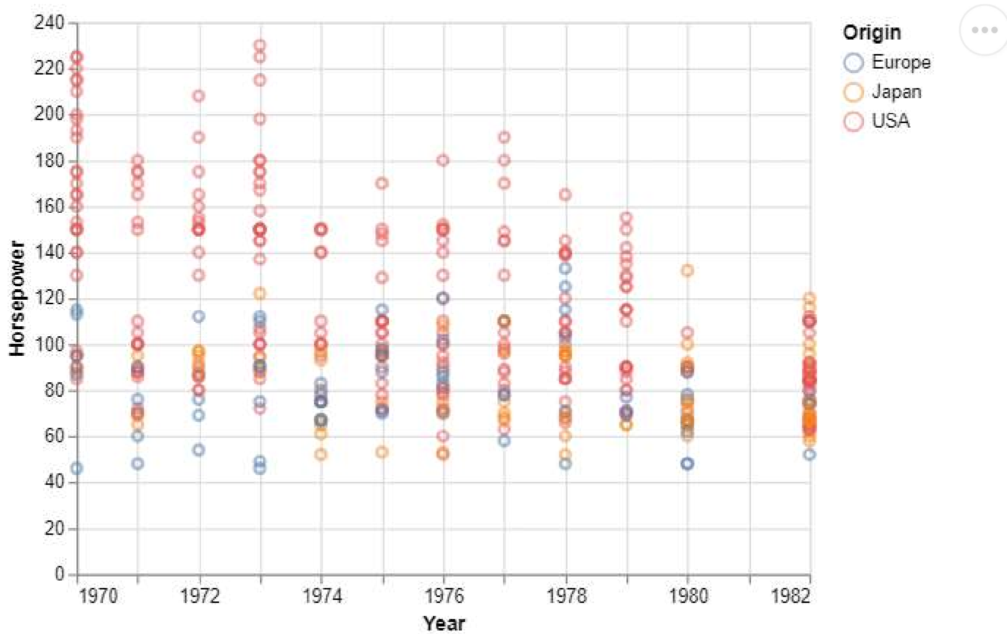
	Name	Miles_per_Gallon	Cylinders	Displacement	Horsepower	Weight_in_lbs	Acceleration	Year	Origin
0	chevrolet chevelle malibu	18.0	8	307.0	130.0	3504	12.0	1970-01-01	USA
1	buick skylark 320	15.0	8	350.0	165.0	3693	11.5	1970-01-01	USA
2	plymouth satellite	18.0	8	318.0	150.0	3436	11.0	1970-01-01	USA
3	amc rebel sst	16.0	8	304.0	150.0	3433	12.0	1970-01-01	USA
4	ford torino	17.0	8	302.0	140.0	3449	10.5	1970-01-01	USA
...	...	...	...	...	...	...	...	...	...
401	ford mustang gl	27.0	4	140.0	86.0	2790	15.6	1982-01-01	USA
402	vw pickup	44.0	4	97.0	52.0	2130	24.6	1982-01-01	Europe
403	dodge rampage	32.0	4	135.0	84.0	2295	11.6	1982-01-01	USA
404	ford ranger	28.0	4	120.0	79.0	2625	18.6	1982-01-01	USA
405	chevy s-10	31.0	4	119.0	82.0	2720	19.4	1982-01-01	USA

406 rows × 9 columns

https://github.com/ubco-mds-2022/Data-550

Scatter plot

▶ Code

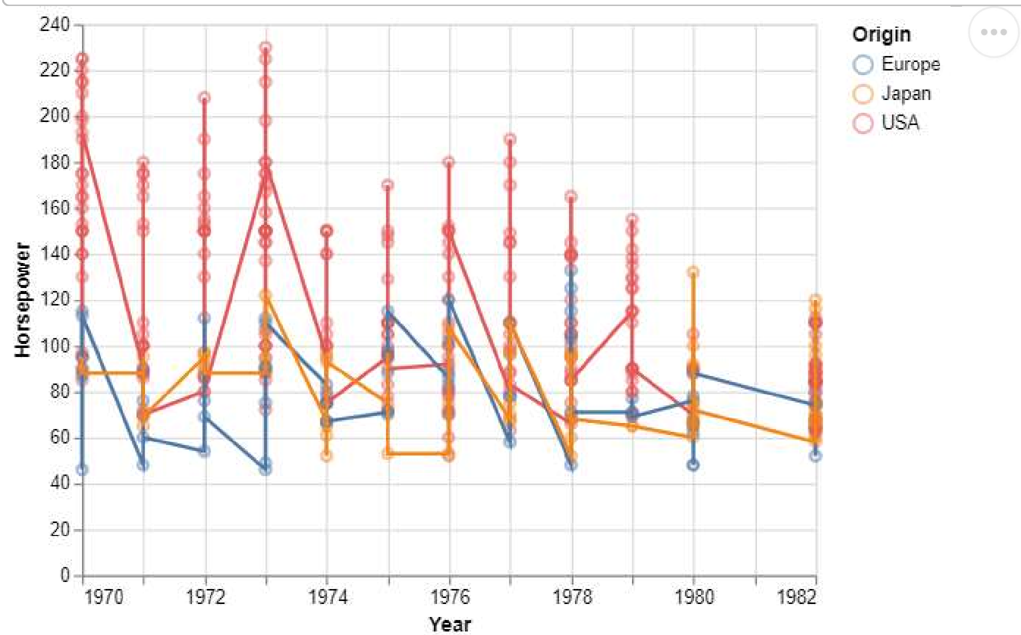


We might be interested in studying the general trend of horsepower over time for European, Japanese and US cars.

https://github.com/ubco-mds-2022/Data-550

Line chart

```
points + points.mark_line()
```

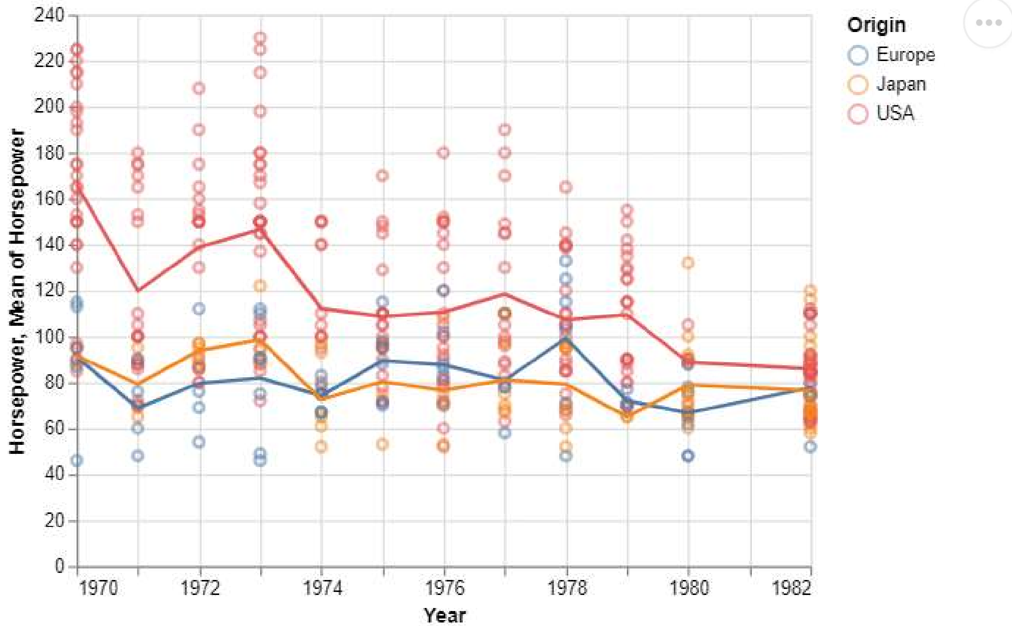


A not so effective way to visualize the trend in this data is to connect all data points with a line.

https://github.com/ubco-mds-2022/Data-550

Mean y-value

```
points + points.encode(y='mean(Horsepower)').mark_line()
```



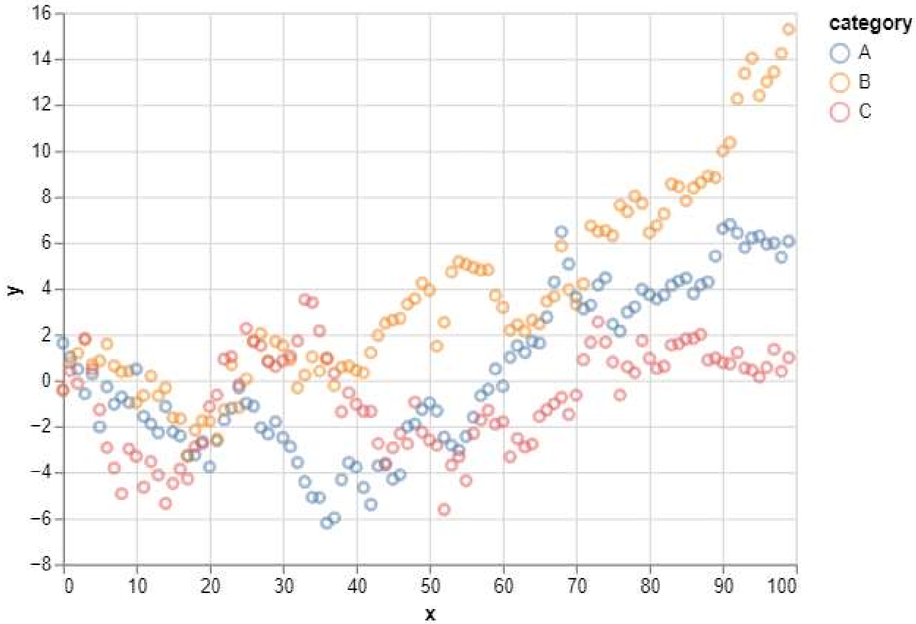
A simple trend line can be found by averaging the mean y-value at each x …

https://github.com/ubco-mds-2022/Data-550

9

Continuous x

▶ Code

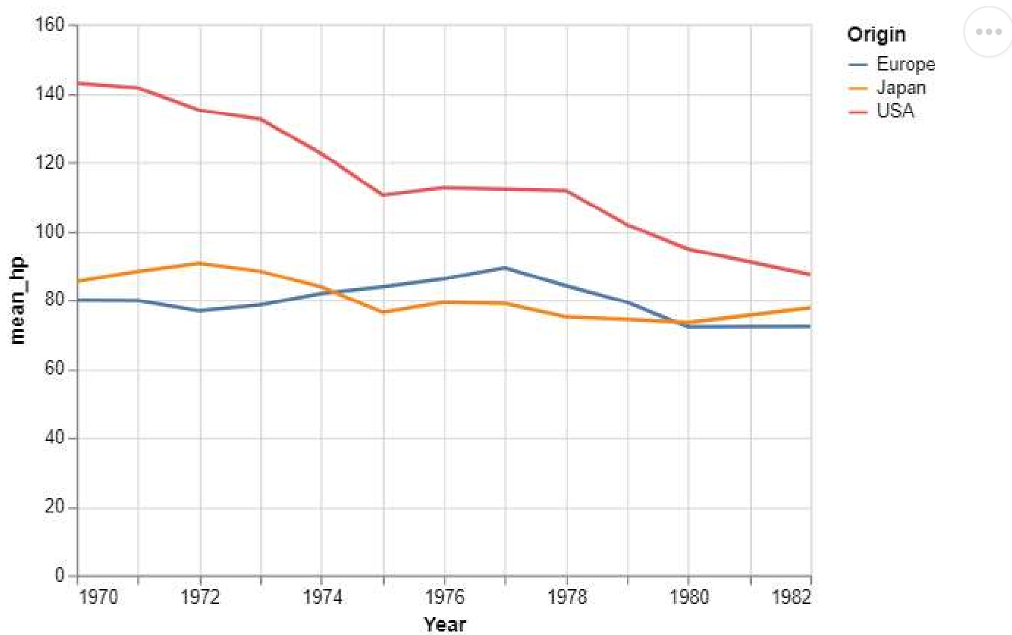


The strategy on the previous slide works for the cars data, but with a continuous x-values, we would need to bin the x-axis before taking the mean y-value.

https://github.com/ubco-mds-2022/Data-550

Moving Average

▶ Show the code (*this will not be on the quiz*)



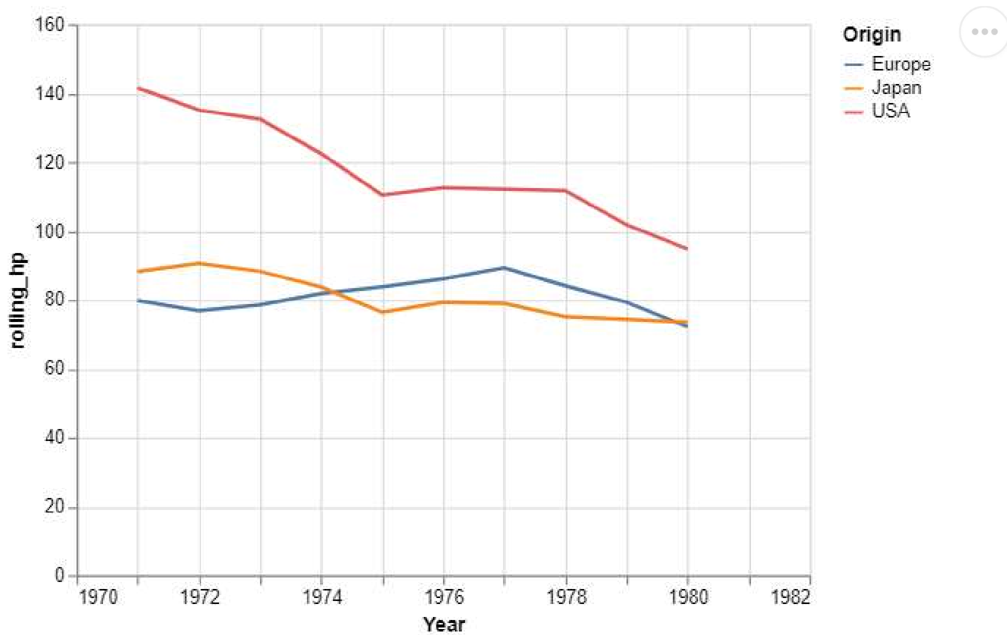
An alternative to binning continuous data is to use a moving/rolling average, that takes the mean of the last n observations.

Altair documentation for transform window.

https://github.com/ubco-mds-2022/Data-550

Rolling method

▶ Show the code (*this will not be on the quiz*)

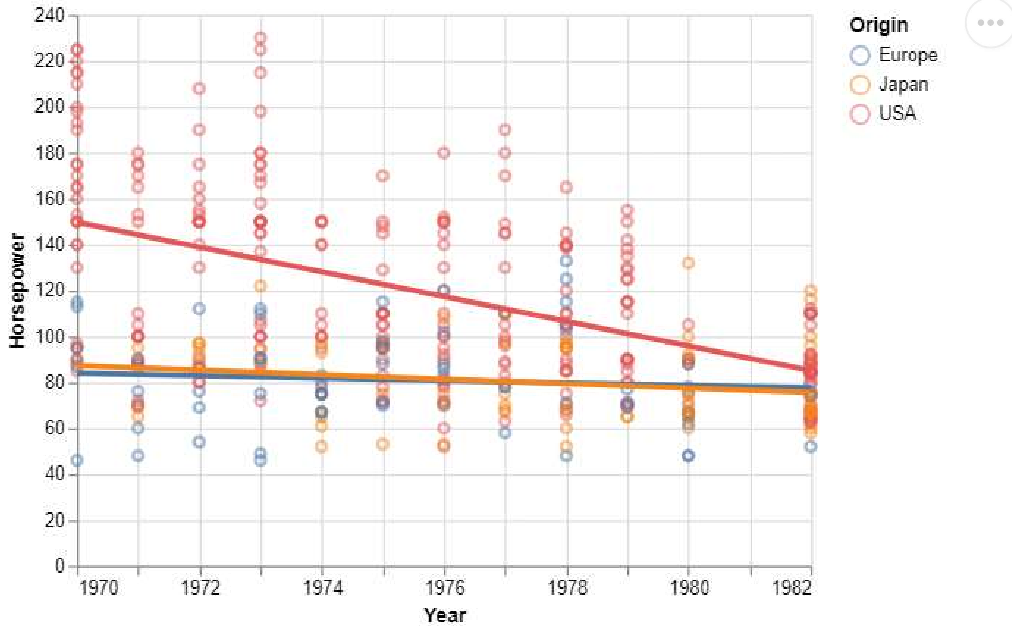


We can also use the rolling method in pandas for this calculation, but it handles the edges a bit differently.

https://github.com/ubco-mds-2022/Data-550

Regression

```
points +  points.transform_regression(
    'Year', 'Horsepower', # The field names of the x and y values, resp
    groupby=['Origin']).mark_line(size=3)
```



Another way of showing a trend in the data is via regression transform[1]

This uses ordinary least squares to fit a linear (`linear`) model with the functional form:

$$y = a + b * x$$

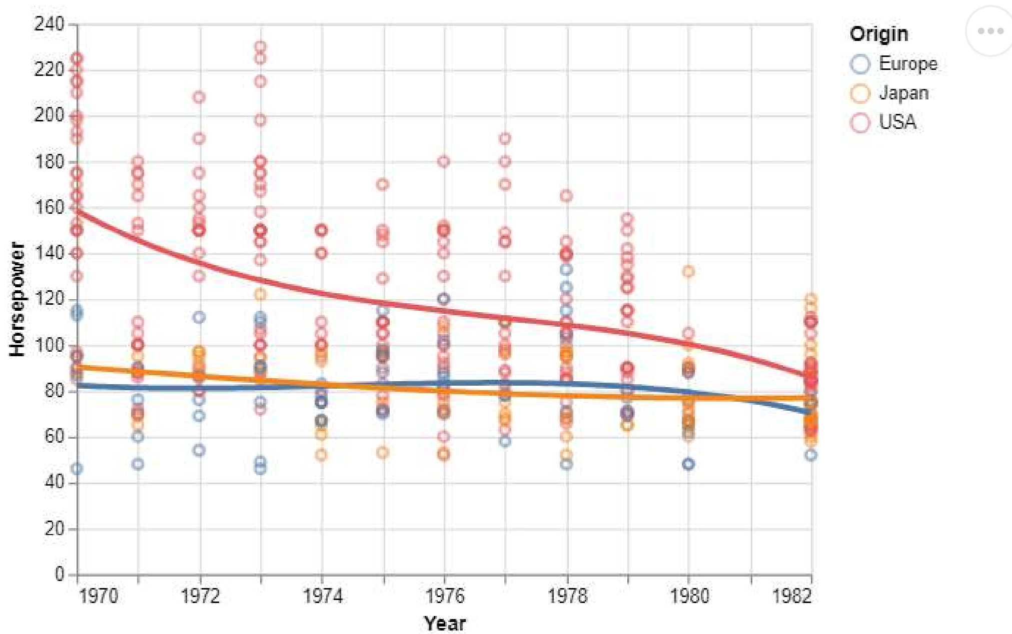
1. this can be used for smoothing and prediction

https://github.com/ubco-mds-2022/Data-550

13

Nonlinear

You are not limited to fitting linear lines; other fits that are:



logarithmic (`log`): $y = a + b * log(x)$

exponential (`exp`): $y = a + eb * x$

power (`pow`): $y = a * xb$

quadratic (`quad`): $y = a + b * x + c * x2$

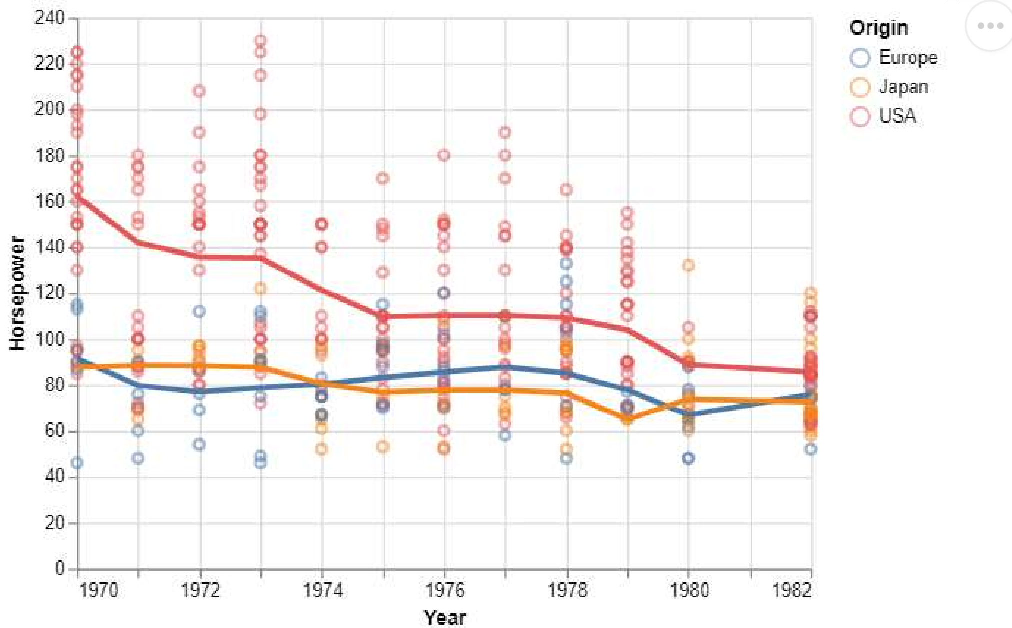
polynomial (`poly`): $y = a + b * x + \ldots + k * xorder$

```
points +  points.transform_regression(
    'Year', 'Horsepower', # The field names of the input x and y values.
    groupby=['Origin'], method='poly'
).mark_line(size=3)
```

https://github.com/ubco-mds-2022/Data-550

Loess

```
points +  points.transform_loess(
    'Year', 'Horsepower', groupby=['Origin']).mark_line(size=3)
```



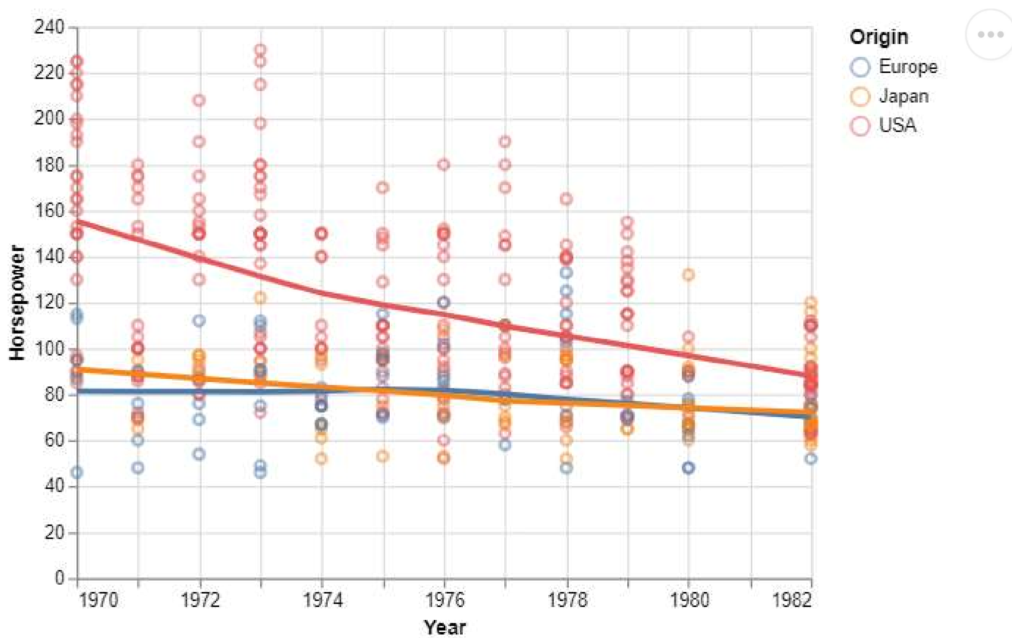
The LOESS transform[1] fits low-degree polynomials to subsets of the data such that points in the center are weighted more heavily than points at the boundaries.

see complete list of tranform loess options here

https://github.com/ubco-mds-2022/Data-550

Bandwidth

▶ Code



The `bandwidth` parameter controls how much the loess fit should be influenced by local variation in the data

https://github.com/ubco-mds-2022/Data-550

When to choose which trendline?

- The most straightforward trendlines when communicating data to a general audience rolling mean. Choose this if it is important that the line has values that are easy to interpret.
- loess works with very little assumptions and tends to produce “natural” results that look right to the human eye
- N.B. loess requires the fitting of many separate regression models, making it slow for large datasets, even on modern computing equipment.

When to choose which trendline?

- Smoothing models can produce widely different results (particularly near the boundaries of the data).
- Furthermore smoothers do not provide parameter estimates that have a meaningful interpretation.
- Therefore, whenever possible, it is preferable to fit a curve with a specific functional form that is appropriate for the data and that uses parameters with clear meaning.

Visualizing Uncertainty

https://github.com/ubco-mds-2022/Data-550

Visualizing Uncertainty

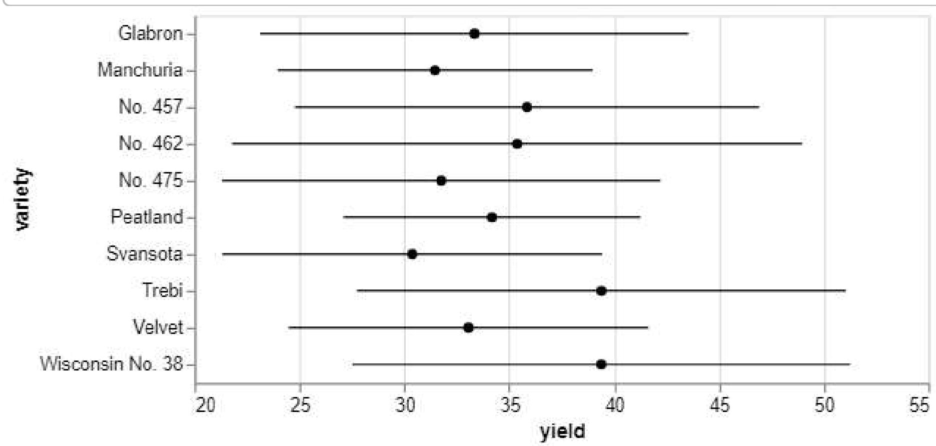
- When we see a data point drawn in a specific location, we tend to interpret it as a precise representation of the true data value.
- Whether and how we choose to represent this uncertainty can make a major difference in how accurately our audience perceives the meaning of the data.
- Two commonly used approaches to indicate uncertainty are error bars (`mark_errorbar`) and confidence bands (`mark_errorbar`).

https://github.com/ubco-mds-2022/Data-550

Error bars

This example shows error bars surrounding the average crop yield of different types of barley in the 1930s.

```
from vega_datasets import data
source = data.barley()
```



What do the bars represent?

https://github.com/ubco-mds-2022/Data-550

Comments

1. It is not obvious what the error bars represent. *Do they represent the standard deviation of the data, the standard error of the mean, a 95% confidence interval, or something else altogether? There is no commonly accepted standard.*
2. By representing each group by a single point (mean) and two error bars, we are losing a lot of information about the data.
3. symmetric error bars are misleading if there is any skew in the data

https://github.com/ubco-mds-2022/Data-550

Comments

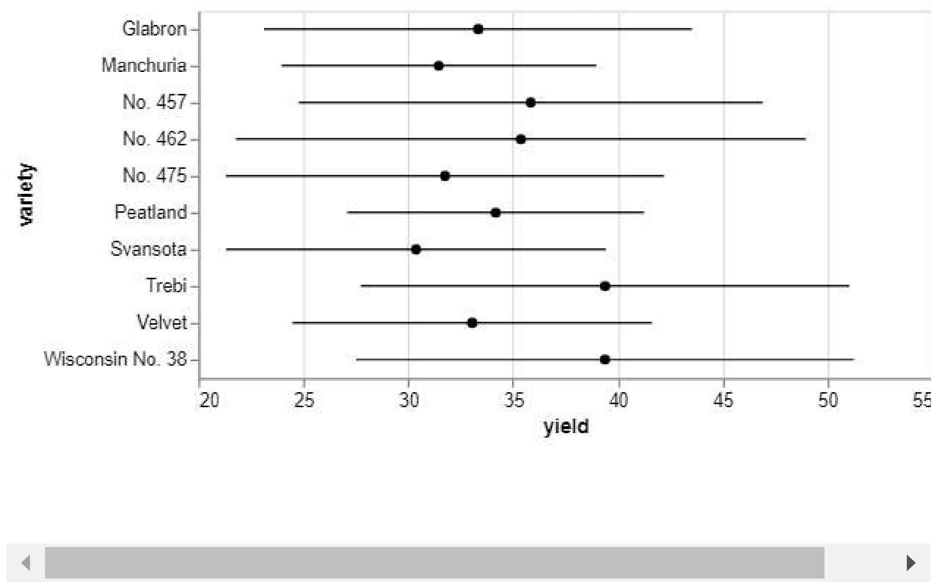
1. It is not obvious what the error bars represent. *Do they represent the standard deviation of the data, the standard error of the mean, a 95% confidence interval, or something else altogether? There is no commonly accepted standard.*
2. It is not obvious what the circles represent (*mean/median*?).
3. By representing each group by a single point (mean) and two error bars, we are losing a lot of information about the data.
4. symmetric error bars are misleading for skewed data

https://github.com/ubco-mds-2022/Data-550

Error bars with Standard Deviation

The error bars in this chart represent standard deviation.

```
error_bars = (alt.Chart(source)
.mark_errorbar(extent='stdev')
.encode(
  alt.X('yield', scale=alt.Scale(z
  alt.Y('variety')))

mean_pts = (alt.Chart(source)
.mark_point(filled=True, color='bl
.encode(
  alt.X('yield', aggregate='mean')
  alt.Y('variety')))

error_bars + mean_pts
```



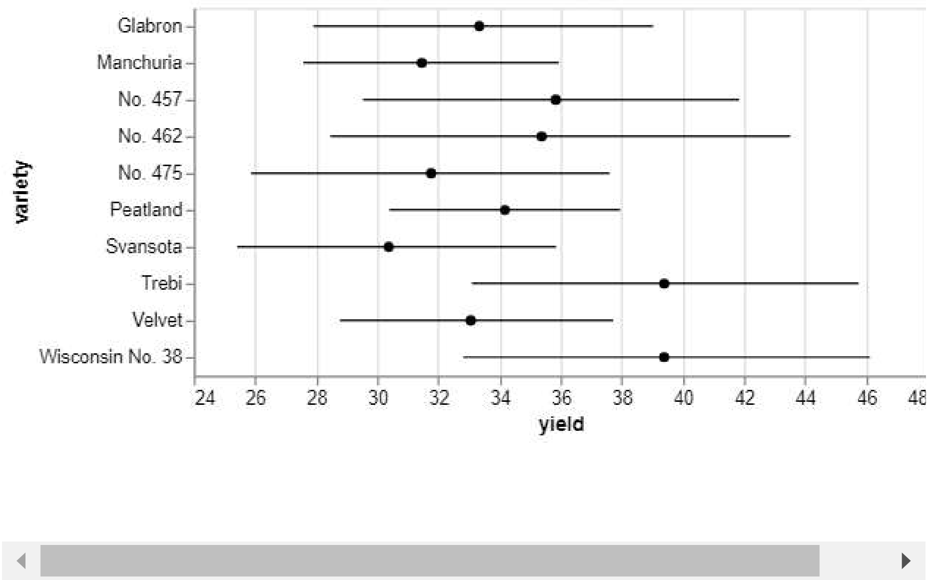
The `extend` argument of `mark_errorbar` tells Altair if you want to show the standard

https://github.com/ubco-mds-2022/Data-550

Error bars with Confidence Interval

These error bars represent a 95% confidence interval.

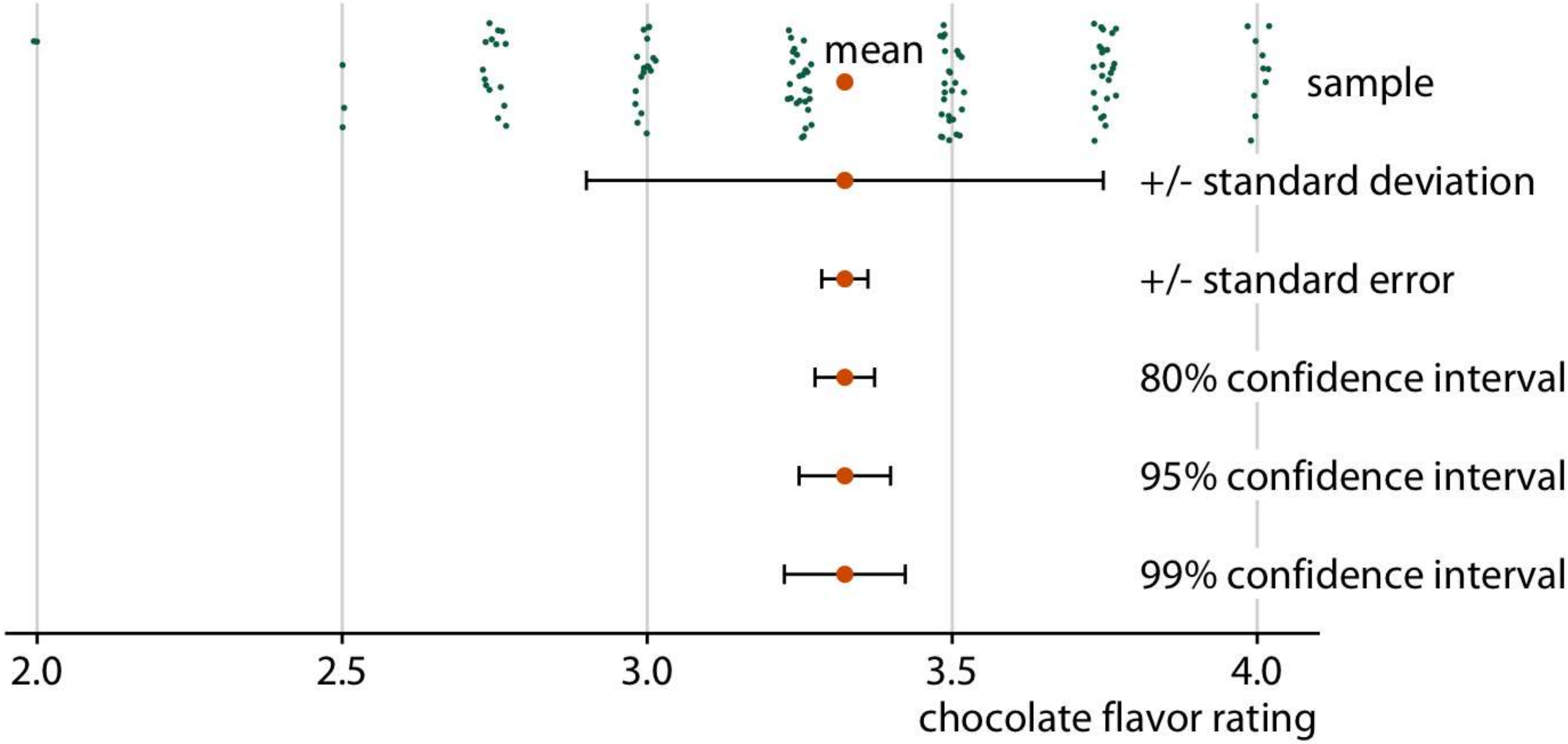
```
error_bars = (alt.Chart(source)
.mark_errorbar(extent='ci')
.encode(
  alt.X('yield', scale=alt.Scale(z
  alt.Y('variety')))

mean_pts = (alt.Chart(source)
.mark_point(filled=True, color='bl
.encode(
  alt.X('yield', aggregate='mean')
  alt.Y('variety')))

error_bars + mean_pts
```



The confidence intervals are computed internally in vega by a non-parametric bootstrap

https://github.com/ubco-mds-2022/Data-550

Error bar choices

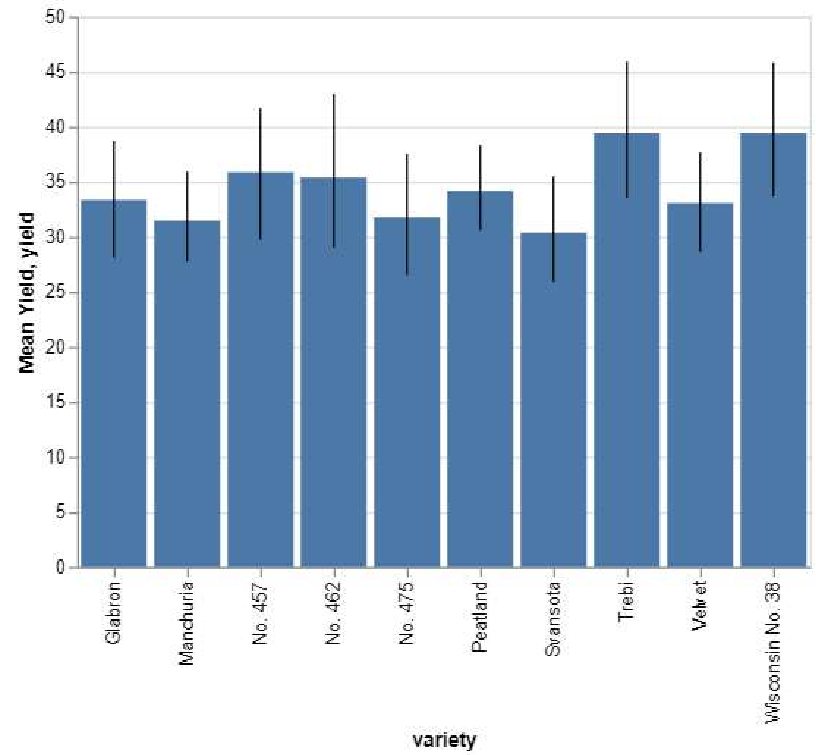


Relationship between sample, sample mean, standard deviation, standard error, and confidence intervals, in an example of chocolate bar ratings. Wilkes Ch 16 Visualizing Uncertainty, Data source: Brady Brelinski, Manhattan Chocolate Society

https://github.com/ubco-mds-2022/Data-550

Bars Charts with Error bars

```
bars = alt.Chart(
).mark_bar().encode(
    alt.X('variety'),
    alt.Y('mean(yield):Q', title='
)

error_bars = alt.Chart(
).mark_errorbar(extent='ci'
).encode(
    x='variety',
    y='yield:Q'
)

alt.layer(bars, error_bars, data=s
```

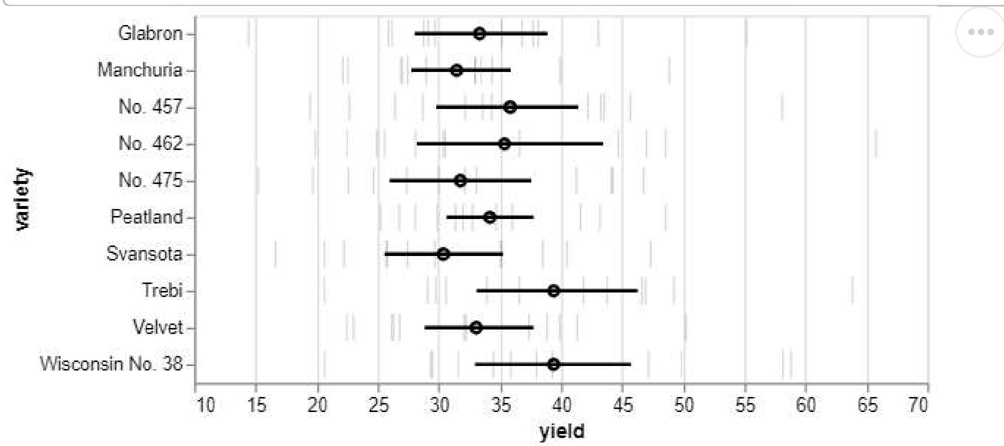


https://github.com/ubco-mds-2022/Data-550

A common alternative called a *dynamite plot* plot only the error bar on top.

Better Alternative

```
err_bars = alt.Chart(source).mark_errorbar(extent='ci', rule=alt.LineConfig
    x=alt.X('yield', scale=alt.Scale(zero=False)),
    y='variety')

(err_bars.mark_tick(color='grey', opacity=0.3)
 + err_bars
 + err_bars.mark_point(color='black').encode(x='mean(yield)'))
```



Another good alternative would be violin plots

Uncertainty of Trendlines

- Trend estimates also have uncertainty associated with them.
- A commonly used approach to show the uncertainty in a trend line with a confidence band
- The confidence band provides us with a range of different fit lines that would be compatible with the data
- To draw a confidence band, we need to specify a confidence level (95% is typical)

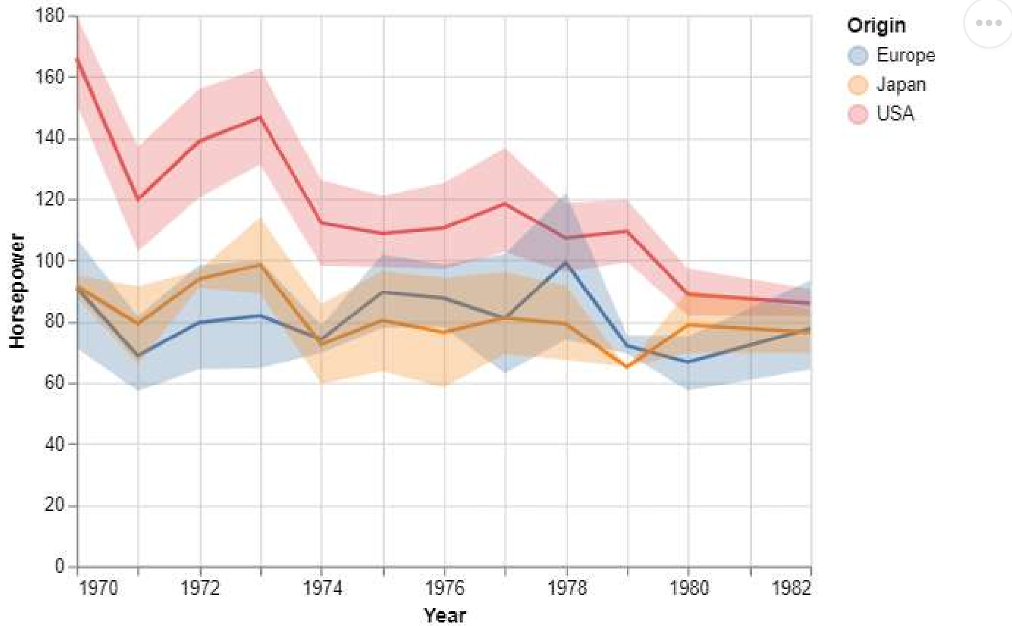
https://github.com/ubco-mds-2022/Data-550

Confidence Bands

- To show the confidence interval of the points as a band, we can use `mark_errorband`.
- As documented here we can set `extent` to:
  - `ci` for confidence interval
  - `stderr` standard error
  - `stdev` for standard deviation
  - `iqr` Extend the band to the q1 and q3.

Average with Confidence bands

```
yearly_avg = points.encode(y='mean(Horsepower)').mark_line()
yearly_avg_ci = points.mark_errorband(extent='ci')
yearly_avg + yearly_avg_ci
```



We can add in the mean line.

https://github.com/ubco-mds-2022/Data-550

https://github.com/ubco-mds-2022/Data-550