

Data 550: Data Visualization I

Lecture 6: Colour Theory and Application

Dr. Irene Vrbik
University of British Columbia Okanagan

Introduction

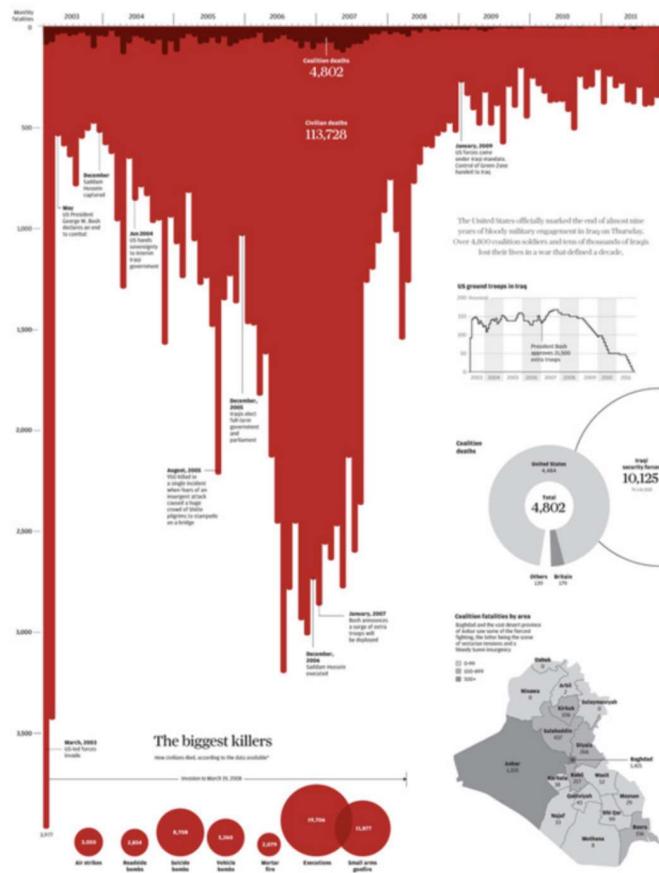
- Today we will focus on some best practices for effective figure design for communication
- We will implement these principles using Altair.
- Suggested supplementary readings are both from Fundamentals of Data Visualization.
 - [Section 3 on color choices](#)
 - [Section 19 on common color pitfalls](#)
 - [18 Handling overlapping points](#)

Lecture Objectives

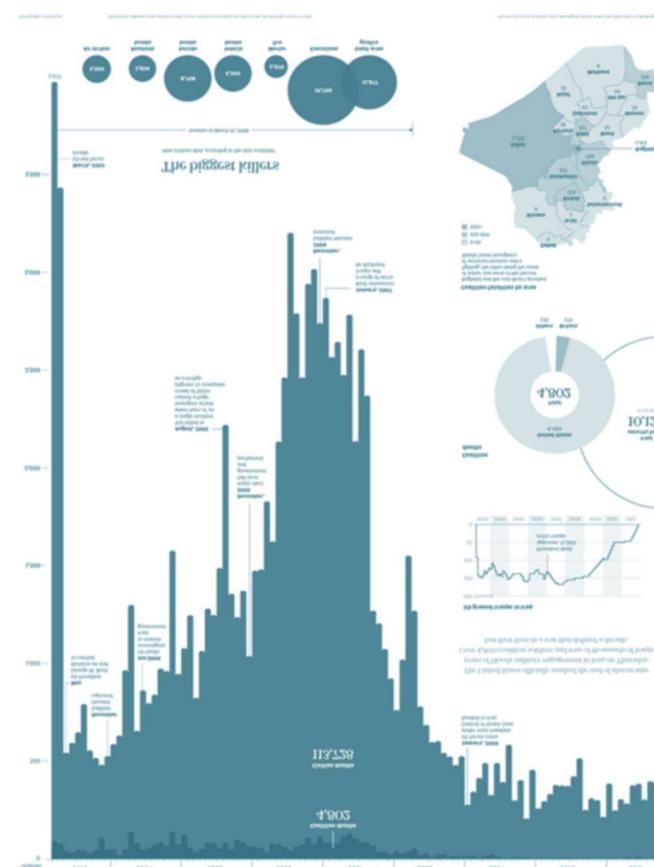
- Discuss guidelines and best practices in visualization design.
- Choose appropriate **color schemes** for your data.
- Use pre-made and custom **color schemes**
- Selectively highlight and annotate data with **color** and **text**.

Data can tell multiple stories

Iraq's bloody toll



Iraq: Deaths on the decline



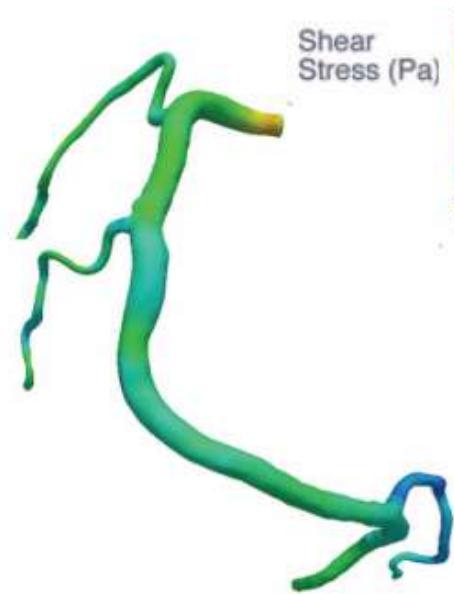
[Source: South China Morning Post](#)
in late 2011, designed by Simon Scarr (click to see more detail)

[Comparisons from Infoworld](#)

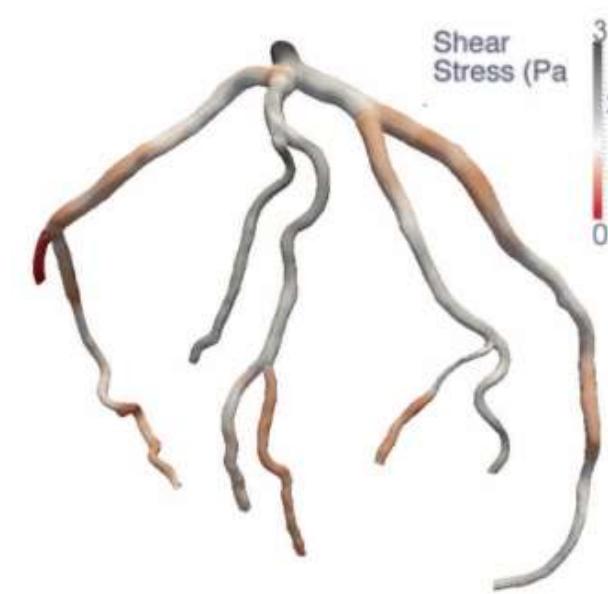
Effective use of colour for categorical data

The affect of colour choice

Recall how changing the colour scheme nearly doubled physician's detection rate of blood vessels regions indicative of potential future heart disease (from 40% to 70%).



40% detection rate



70% detection rate

Properties of colours



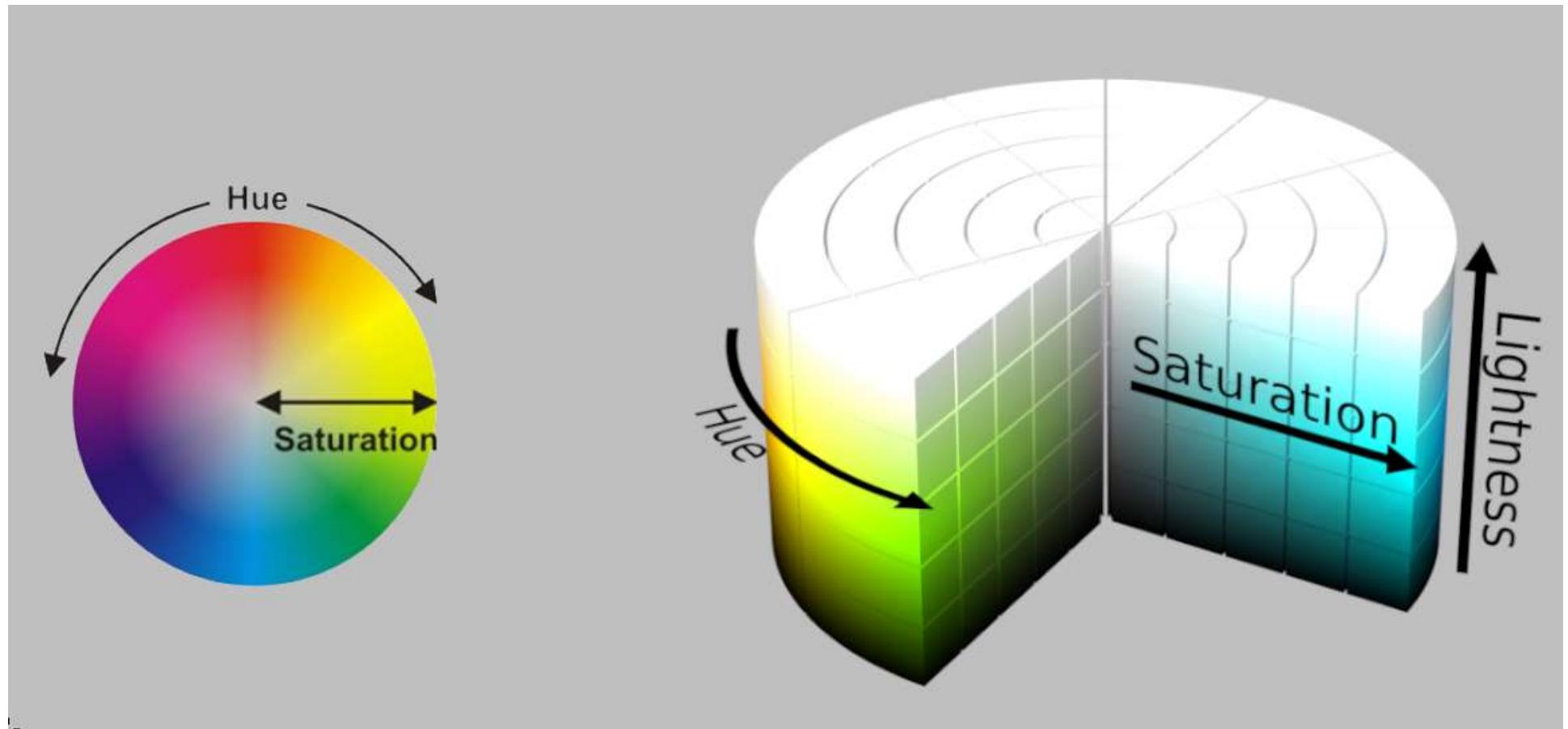
Hue is what we traditionally think of as the “colour”

Each hue can have a varying *saturation*, which ranges from a dull, greyish appearance to a vibrant fully saturated hue.

Lightness is how bright the colour is; each hue starts at black (no lightness) and ends at white (full lightness)



Properties of colours schematic



Properties of colours (cont'd)

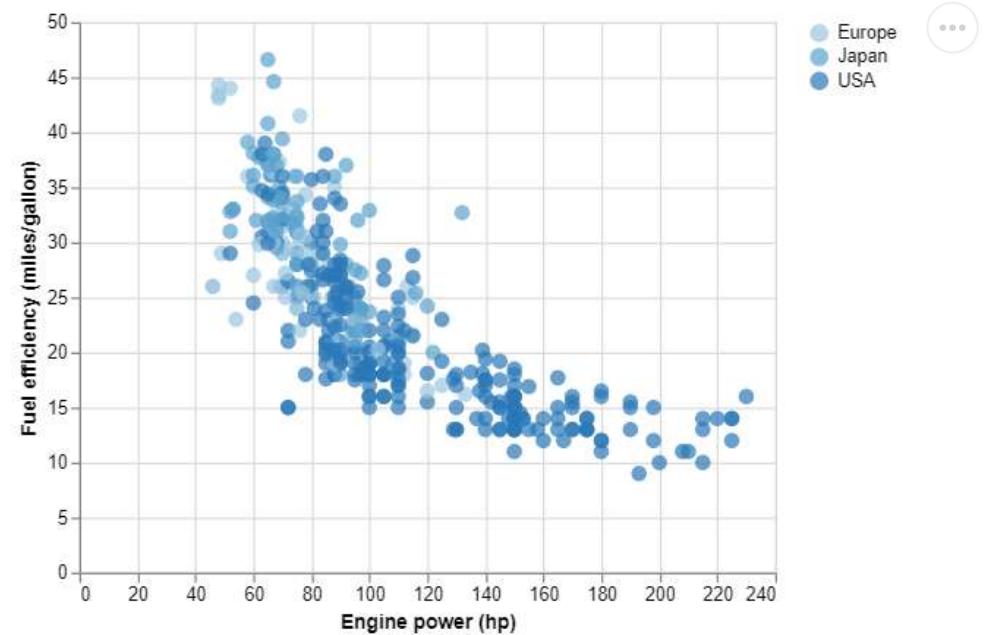
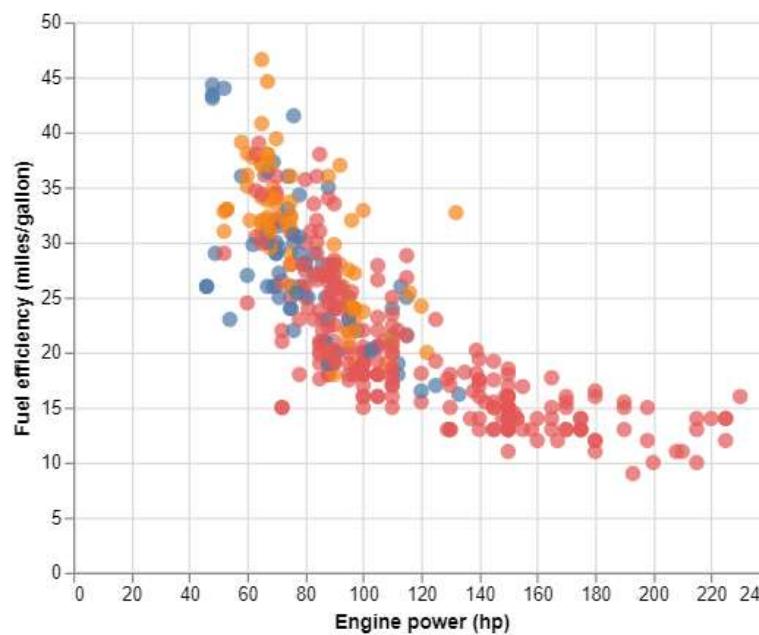


Hue and lightness are the most important from a data viz perspective, and we use them for *categorical* and *quantitative* data, respectively.

Saturation is more often used as a stylistic choice to emphasize/deemphasize.

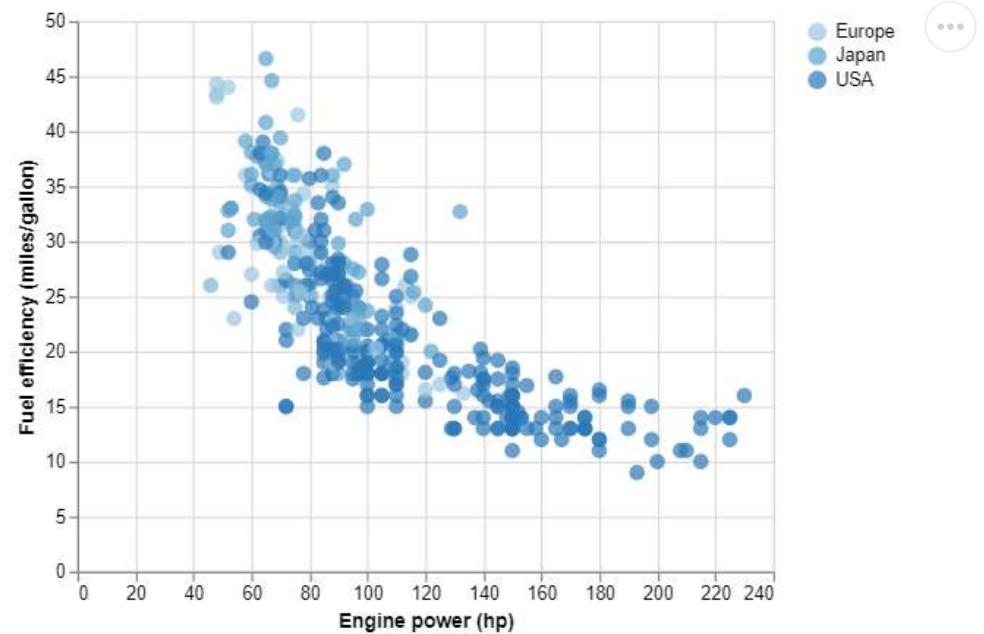
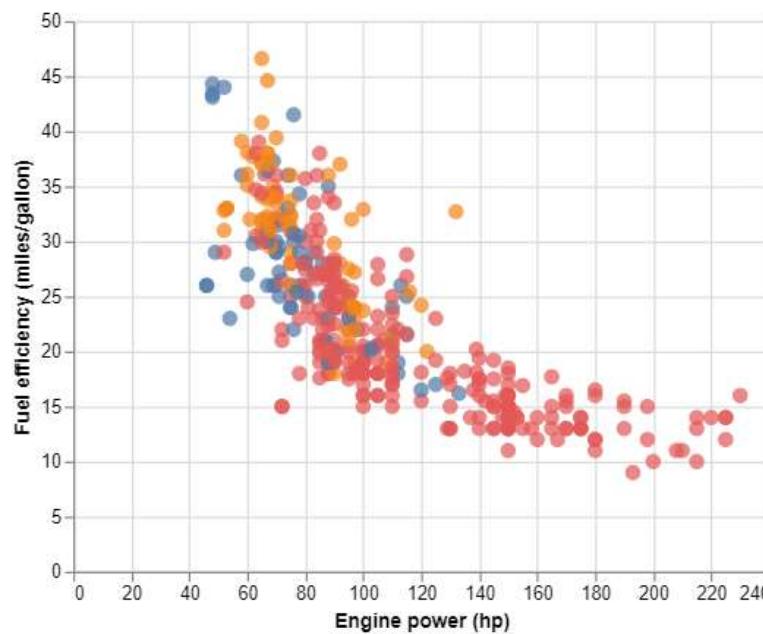
Categorical with hue

Left plot: uses different hues to separate the points belonging to different categories. We can quickly identify that there three distinct colours



Categorical with lightness

Right plot: we used lightness within a single hue (blue) to label the different categories which makes it much harder to say how many different categories there are



Specifying custom colours

- ▶ Show the code

Javascript Error: Tn.compile is
not a function

This usually means there's a typo
in your chart specification. See
the javascript console for the full
traceback.

You can define your own
*colour schemes*¹

However, it is often a
better idea to use the
defaults or select from
predefined colour schemes
designed by experts² ...

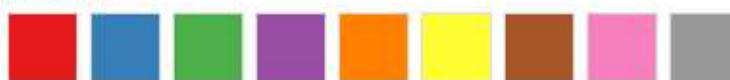
1. a.k.a colourmaps or colour palettes

2. see how the [default colormap for Matplotlib](#) (another plotting package) was created

Colour schemes for categorical values

You see the full list of colour schemes [here](#)

set1



set2



set3



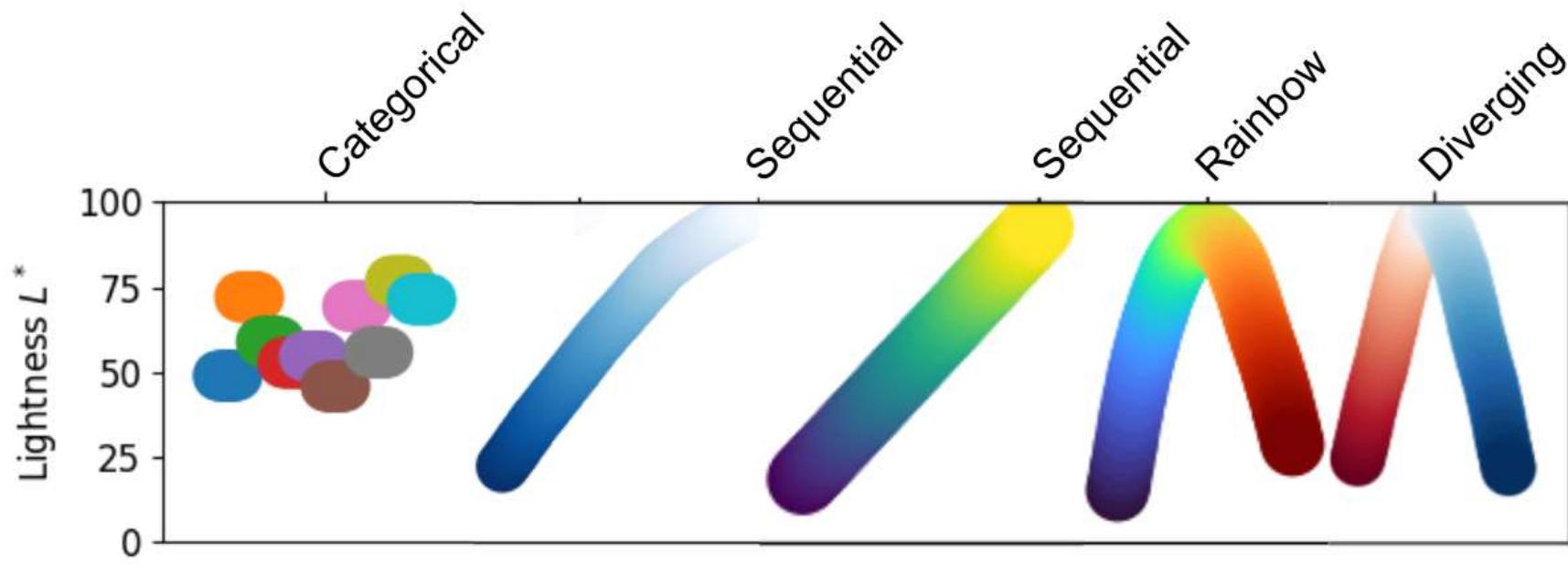
tableau10



tableau20



Colour Schemes for Quantitative Data



Diverging colour schemes are suitable for plots with a natural midpoint. Diverging colour schemes could be an effective choice to highlight values at the extreme lows and extreme highs (e.g., temperatures)

rainbow not very good because can give impression of clusters when there aren't any

There are also cyclical color schemes that start and end at the same hue; read more [here](#)

14

Specifying Colour Schemes in Altair

```
1 alt.Chart(cars).mark_point(  
2     size=70,  
3     filled=True  
4 ).encode(  
5     alt.X('Horsepower', title='Engine  
6     alt.Y('Miles_per_Gallon', title=  
7     color = alt.Color(  
8         'Origin', title=None,  
9         scale=alt.Scale(scheme='set1')  
10    )  
11 )
```

a lot of people color blind to red and green

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

Caption: A scatterplot using *set1* colour scheme

N.B. colour schemes (esp. ones that mix red and green) can be difficult to interpret for

More categories than hues

- ▶ Show the code

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

Colours are recycled when
there are too many categories.

Consider consolidating
categories or breaking up the one chart into two or three different charts.

*NOTE: The solution is **not** to use more hues*

General Guidelines

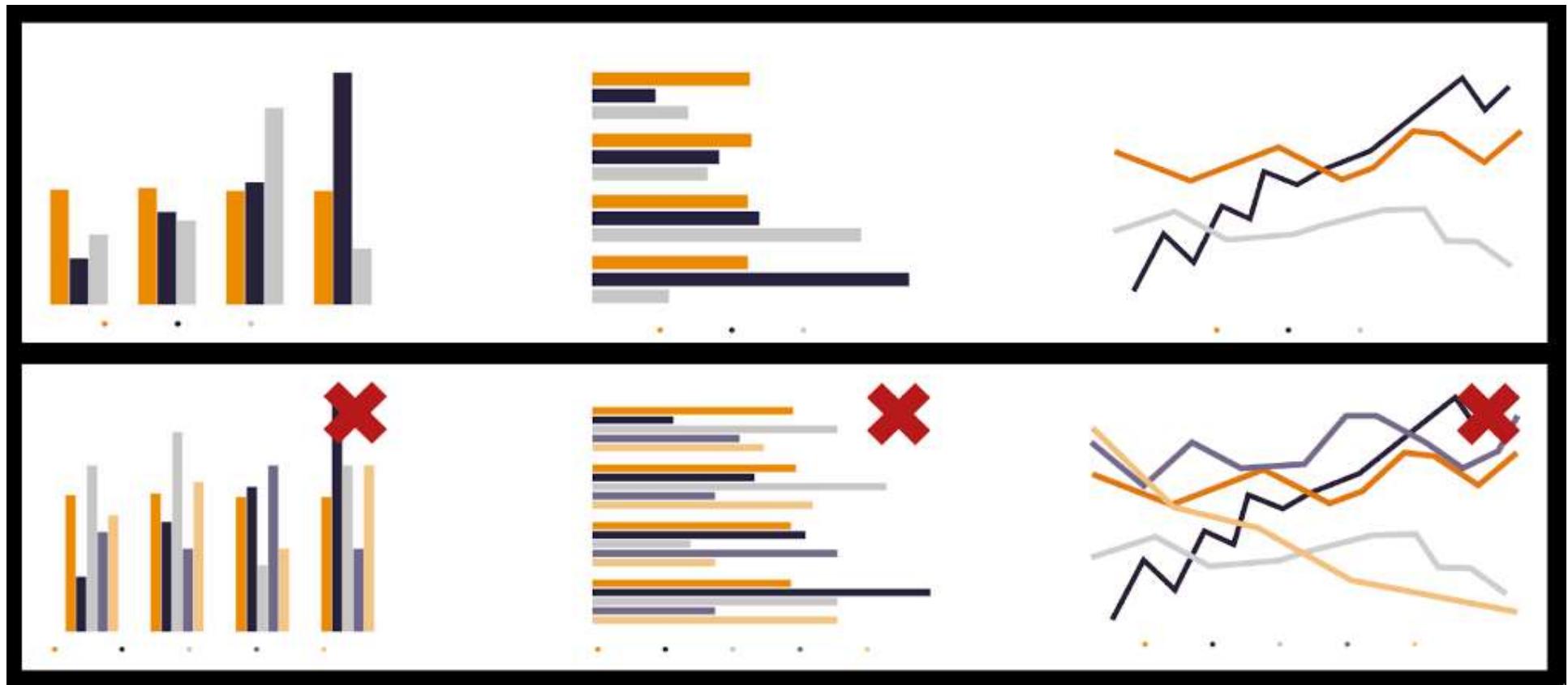


Image source: [PocketStyleBook](#)

Keep color mappings the same from graph to graph

General Guideline Summary

- The guidelines on what is too many hues differ between different sources and also depends on your use case.
- A good rule of thumb is that when you get to around five different hues, you should really consider if this is the best way to represent your data or if you could split it up into multiple visualizations instead.
- Do not use different colours for the same categories between subplots (Altair always starts with blue as the default)

Effective use of colour for quantitative data

Lightness for Quantitative Data

Lightness and Hue variations

Direction of colour scheme

- ▶ Show the code

see [matplotlib doc](#) for viridis

Javascript Error: Tn.compile is
not a function

This usually means there's a typo
in your chart specification. See
the javascript console for the full
traceback.

You do not necessarily
need to map larger values
to darker values

Guide: set the most
important values to have
the most contrast to the
background

If there is no difference in importance then the convention is to set the highest values to

Dont chase the rainbow

- ▶ Show the code

Javascript Error: Tn.compile is
not a function

This usually means there's a typo
in your chart specification. See
the javascript console for the full
traceback.

- it is less accessible
- perceive groups where
they are none
- eyes are drawn to yellow
and cyan (brighter)
- does not translate to
grayscale

More details: [How bad is your colourmap?](#), this series of posts, Google's Turbo

23

Colour Schemes for colour vision deficiencies

- ▶ Show the code

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

'cividis' is a colour scheme designed to look almost exactly the same for people with the most common colour vision deficiencies

This website lets you upload an image and simulate different colour vision deficiencies.

Examples

Diverging Colour Schemes

- ▶ Show the code

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.
See the javascript console for the full traceback.

Last 1000 days of weather data downloaded from [here](#)

Saturated Charts

- ▶ Show the code

Javascript Error: Tn.compile is
not a function

This usually means there's a typo
in your chart specification. See
the javascript console for the full
traceback.

When a plot becomes
oversaturated, it could
help to move to a 2D
histogram ...

2D Histogram

A heatmap can visualize the relationship between two distributions without saturation

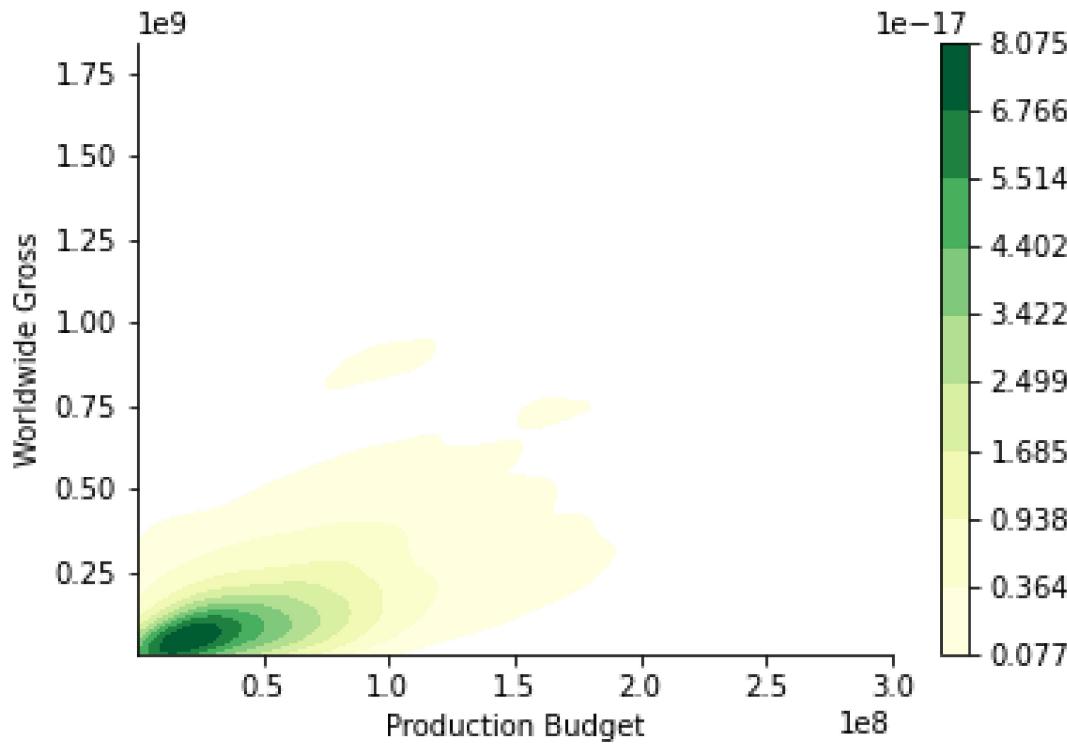
```
1 alt.Chart(movies_extended).mark_rect().encode(
2     alt.X('Production Budget', bin=alt.Bin(maxbins=60), axis=alt.Axis(format='$')),
3     alt.Y('Worldwide Gross', bin=alt.Bin(maxbins=60), axis = alt.Axis(format='$')),
4     alt.Color('count()'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.
See the javascript console for the full traceback.

2D Density plot

A 2D density plot can also visualize the relationship between two distributions without saturation



Reading in the data

```
1 import altair as alt
2 import pandas as pd
3
4 movies_extended = pd.read_csv('data/movies-extended.csv')
5 movies_extended.head()
```

	Title	US Gross	Worldwide Gross	US Gross	DVD Sales	Production Budget	Release Date	MPAA Rating	Running Time min	Distributor	Source	R C A E C A
0	Boynton Beach Club	3127472.0	3127472.0	NaN	2900000.0	Mar 24 2006	R	104.0	Wingate Distribution	Original Screenplay	C	
1	Broken Arrow	70645997.0	148345997.0	NaN	65000000.0	Feb 09 1996	R	108.0	20th Century Fox	Original Screenplay	A	
2	Brazil	9929135.0	9929135.0	NaN	15000000.0	Dec 18 1985	R	136.0	Universal	Original Screenplay	E	C
3	The Cable Guy	60240295.0	102825796.0	NaN	47000000.0	Jun 14 1996	PG-13	95.0	Sony Pictures	Original Screenplay	C	
4	Chain Reaction	21226204.0	60209334.0	NaN	55000000.0	Aug 02 1996	PG-13	106.0	20th Century Fox	Original Screenplay	A	

Bar charts

Bar charts are effective for visualizing categorical “distributions” of a single column

```
1 alt.Chart(movies_extended).mark_bar().encode(  
2     alt.X('count()'),  
3     alt.Y('Major Genre', sort='x'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.

See the javascript console for the full traceback.

Stacked bar charts

Stacked bar charts can visualize counts for combinations of two categorical columns

```
1 alt.Chart(movies_extended).mark_bar().encode(  
2     alt.X('count()'),  
3     alt.Y('Major Genre', sort='x'),  
4     alt.Color('MPAA Rating'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.
See the javascript console for the full traceback.

Rescaling

Rescaling the bar lengths facilitates comparing proportions between bars

```
1 alt.Chart(movies_extended).mark_bar().encode(  
2     alt.X('count()', stack='normalize', title='Proportion of movies'),  
3     alt.Y('Major Genre', sort='x'),  
4     alt.Color('MPAA Rating'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.
See the javascript console for the full traceback.

Sorting

Sorting by the length of one of the coloured segments make the chart easier to read

- ▶ Show code for sort_order

```
1 alt.Chart(movies_extended).mark_bar().encode(  
2     alt.X('count()', stack='normalize', title='Proportion of movies'),  
3     alt.Y('Major Genre', sort=sort_order),  
4     alt.Color('MPAA Rating'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.
See the javascript console for the full traceback.

Normalize stacked bar charts

Normalize stacked bar charts are effective at visualizing just a few categories

```
1 sort_order = ['Concert/Performance', 'Musical', 'Documentary', 'Adventure',
2                 'Comedy', 'Romantic Comedy', 'Drama', 'Action']
3 alt.Chart(movies_extended[movies_extended['MPAA Rating'].isin(['G', 'PG'])])
4     alt.X('count()', stack='normalize', title='Proportion of movies'),
5     alt.Y('Major Genre', sort=sort_order),
6     alt.Color('MPAA Rating'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.

See the javascript console for the full traceback.

Faceting

Showing bars side by side makes it easier to compare their exact heights within a category

```
1 alt.Chart(movies_extended).mark_bar().encode(
2     alt.X('count()', title=''), # rm axis titles for legibility
3     alt.Y('MPAA Rating', title=''),
4     alt.Color('MPAA Rating', legend=None))
5 .properties(width=100, height=45)
6 .facet('Major Genre', columns=4)
7 .resolve_scale(x='independent'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.

See the javascript console for the full traceback.

Faceting 2

Switching the facetting and y column targets the plot towards a slightly different question

```
1 alt.Chart(movies_extended).mark_bar().encode(
2     alt.X('count()', title=''),
3     alt.Y('Major Genre', title='', sort='x'),
4     alt.Color('MPAA Rating', legend=None))
5 .properties(width=100, height=150)
6 .facet('MPAA Rating')
7 .resolve_scale(x='independent'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.

See the javascript console for the full traceback.

Heatmaps 2

An alternative to heatmaps which is not as commonly seen would be to use a mark that maps size to color.

```
1 alt.Chart(movies_extended).mark_rect  
2     alt.Color('count()'),  
3     alt.X('MPAA Rating'),  
4     alt.Y('Major Genre', sort='col')
```

Javascript Error: Tn.compile is not a function
This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

Comparing squares vertically in this heatmap is similar to the **first faceted plot** we made and comparing them horizontally is similar to the faceted visualization in the last slide.

Sizing

Using both the colour and marker size to indicate the count creates a more effective visualization¹

```
1 alt.Chart(movies_extended
2 ).mark_circle().encode(
3     alt.X('MPAA Rating'),
4     alt.Y('Major Genre', sort='col'),
5     alt.Color('count()'),
6     alt.Size('count()))
```

Javascript Error: Tn.compile is not a function
This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

1. recall from our visual rankings chart that we cannot perceive small variations in colour as accurately as we can for other visual channels, such as the position or size.

Annotating with text and colour

Wikipedia donations by day

- ▶ Show the code

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

Here we are using the built-in calendar module to get a sorted list of the weekday abbreviations, but we could also have written it out by hand.

Wednesday stands out as the day where the highest amount of money is donated.

To draw additional attention to this, we could colour this bar to highlight it.

Colours for highlighting

► Show the code

Javascript Error: Tn.compile is
not a function

This usually means there's a
typo in your chart
specification. See the
javascript console for the full
traceback.

We'll use `this` helper function which uses
the same `.datum` prefix for accessing data
columns as data transformation functions.

We check if the `week_day`
column has the value `Wed`

- If yes, we'll use the `coral`
colour
- If no, we use the default
colour (`steelblue`)

Text labels

```
1 text_label = chart.mark_text(  
2   align='left', dx=-125, dy=-15  
3 ).encode(text=alt.condition(alt.datum.week_day == 'Wed',  
4                           alt.value('Salaries are paid on Wed'),  
5                           alt.value('')))  
6 chart + text_label
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

Similar to our direct labeling example, we can add a custom text annotation to as explanatory labels

Thanks to the grammar of graphics we can very quickly and easily add text labels indicating the total values associated with each bar ...

```
1 chart + chart.mark_text(align='left').encode(text='sum(sum)')
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.

See the javascript console for the full traceback.

The default formatting is not very easy to read, so let's use what we learned previously to reformat the label and make this chart more effective.

```
1 (chart + chart  
2 .mark_text(align='left', dx=2)  
3 .encode(text  
4 =alt.Text('sum(sum)', format='$,d')
```

Javascript Error: Tn.compile is not a function
This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

```
1 (chart + chart  
2 .mark_text(align='left', dx=2)  
3 .encode(text=  
4 alt.Text('sum(sum)', format='$.3~$'))
```

Javascript Error: Tn.compile is not a function
This usually means there's a typo in your chart specification. See the javascript console for the full traceback.

Remove x-axis

► Code for titles

```
1 chart = alt.Chart(donations, title=title).mark_bar().encode(
2     alt.Y('week_day', sort=list(day_abbr), title=None),
3     alt.X('sum(sum)', axis=None),
4     color=alt.condition(alt.datum.week_day == top_day,
5                         alt.value('coral'),
6                         alt.value('steelblue')))
7 (chart + chart.mark_text(align='left', dx=2)
8 .encode(text=alt.Text('sum(sum)', format='$.3s'))).configure_view(strokeWid
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.

See the javascript console for the full traceback.

Altair Themes

Themes can be used to change the overall styling of the chart

```
1 alt.themes.enable('dark');
2 ((chart
3   + chart
4     .mark_text(align='left', dx=2)
5     .encode(text=alt.Text('sum(sum)', format='$.3s')))
6     .configure_view(strokeWidth=0)
7     .configure_title(subtitleColor='white'))
```

Javascript Error: Tn.compile is not a function

This usually means there's a typo in your chart specification.

See the javascript console for the full traceback.

Choice of themes

All the available themes can be shown by typing:

```
1 alt.themes.names()
```

```
['dark',
'default',
'fivethirtyeight',
'ggplot2',
'latimes',
'none',
'opaque',
'quartz',
'urbaninstitute',
'vox']
```

You can see samples of what these themes look like here.

```
ThemeRegistry.enable('default')
```

