

Introduction to Data Science

Data Visualization and Effective Storytelling

Joanna Bieri DATA101

Better Visualizations

ASK

What makes a graph hard to read?

What makes a graph easy to read?

Better Visualizations

ANSWER

- Keep it simple. No 3D, no extra decoration.
- Make it easy to tell the categories apart.
- Use color to draw attention to what matters.
- Tell a story. Annotate if you need to.

Order Matters

PREDICT

df

	pet	count
0	cat	5
1	dog	9
2	fish	2

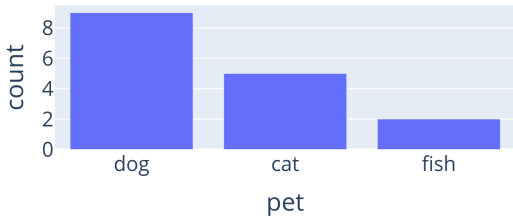
```
fig = px.bar(df, x='pet', y='count')  
fig.update_layout(  
    xaxis={'categoryorder': 'total descending'})
```

What order are the bars in, left to right?

Order Matters

ANSWER

```
fig = px.bar(df, x='pet', y='count')  
fig.update_layout(  
    xaxis={'categoryorder': 'total descending'})
```



total descending puts the biggest bar first. Without that line the bars come in the order of the rows: cat, dog, fish.

Order Matters

REVERSE

df

	size	count
0	small	4
1	large	7
2	medium	6

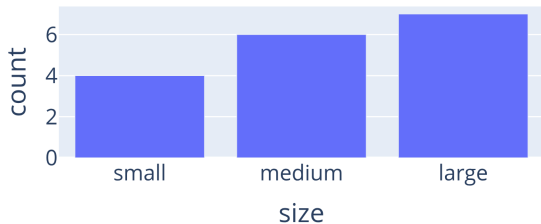
```
fig = px.bar(df, x='size', y='count')
```

We want the bars in the order small, medium, large. What is the command?

Order Matters

ANSWER

```
fig.update_layout(xaxis={'categoryorder': 'array',  
                        'categoryarray': ['small', 'medium', 'large']})
```



Alphabetical order and total order both give large, medium, small. When the categories have their own order you have to type it in.

Totals per Group

PREDICT

df

	vote	location	count
0	Right	london	3
1	Wrong	london	5
2	Right	north	4
3	Wrong	north	2

```
df.groupby('location', as_index=False).sum()
```

What comes out? How many rows?

Totals per Group

ANSWER

```
df.groupby('location', as_index=False).sum()
```

	location	vote	count
0	london	RightWrong	8
1	north	RightWrong	6

One row per location. `sum()` adds the numbers and pastes the words together. For just the counts:

```
df.groupby('location', as_index=False)['count'].sum()
```

Long Labels

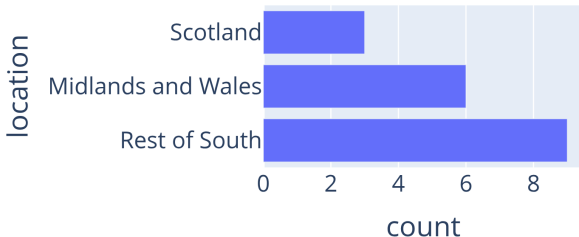
ASK

Your labels are long and they overlap,
or you have 20 bars. What do you change?

Long Labels

ANSWER

```
fig = px.bar(df, y='location', x='count')
```



Put the categories on the y axis and the numbers on the x axis. Long labels read left to right, and lots of bars fit going down the page.

Pick a Purpose

PREDICT

df

	vote	location	count
0	Right	london	3
1	Wrong	london	5
2	Right	north	4
3	Wrong	north	2

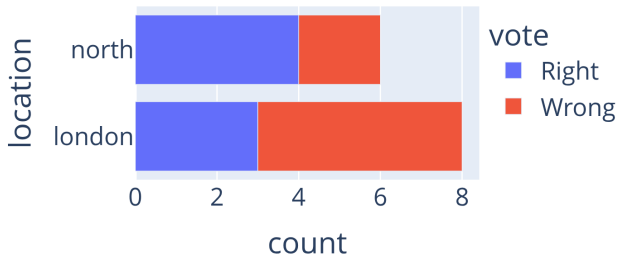
```
fig = px.bar(df, y='location', x='count', color='vote')
```

How many bars? What does the color show?

Pick a Purpose

ANSWER

```
fig = px.bar(df, y='location', x='count', color='vote')
```



One bar per location, split into one color per vote. Now we can see how the votes compare inside each location.

Facets

PREDICT

df

	vote	location	count
0	Right	london	3
1	Wrong	london	5
2	Right	north	4
3	Wrong	north	2

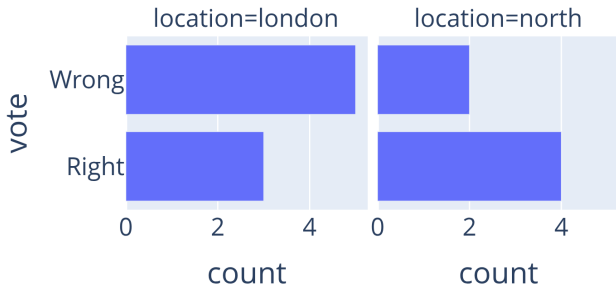
```
fig = px.bar(df, y='vote', x='count', facet_col='location')
```

How many panels? How many bars in each panel?

Facets

ANSWER

```
fig = px.bar(df, y='vote', x='count', facet_col='location')
```



One panel per location, one bar per vote. The panel titles say `location=london` until we clean them up.

Avoid Redundancy

PREDICT

df

	vote	location	count
0	Right	london	3
1	Wrong	london	5
2	Right	north	4
3	Wrong	north	2

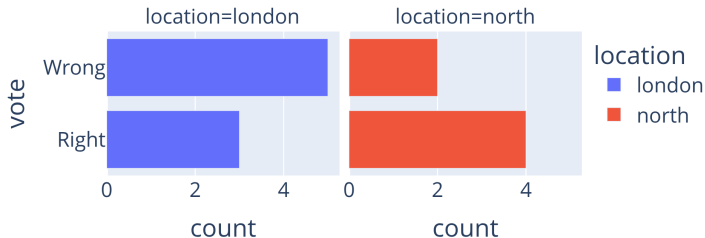
```
fig = px.bar(df, y='vote', x='count',  
             color='location', facet_col='location')
```

What does this graph tell us more than once?

Avoid Redundancy

ANSWER

```
fig = px.bar(df, y='vote', x='count',  
             color='location', facet_col='location')
```



The location is there three times: panel title, color, and legend. To fix it take out color, or add `showlegend=False`.

Meaningful Colors

REVERSE

df

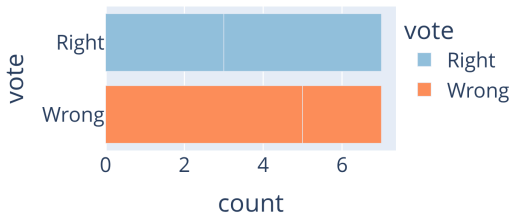
	vote	location	count
0	Right	london	3
1	Wrong	london	5
2	Right	north	4
3	Wrong	north	2

We want Right in blue #91bfdb and Wrong in orange #fc8d59.
What is the command?

Meaningful Colors

ANSWER

```
fig = px.bar(df, y='vote', x='count', color='vote',  
             color_discrete_map={'Right': '#91bfdb',  
                                 'Wrong': '#fc8d59'})
```



The names in the map have to match the values in the data exactly, spelling and capitals included.

Questions?

Questions?