Python plotting

Andreas Bjerre-Nielsen

Recap

What have we learned about basic Python and Pandas?

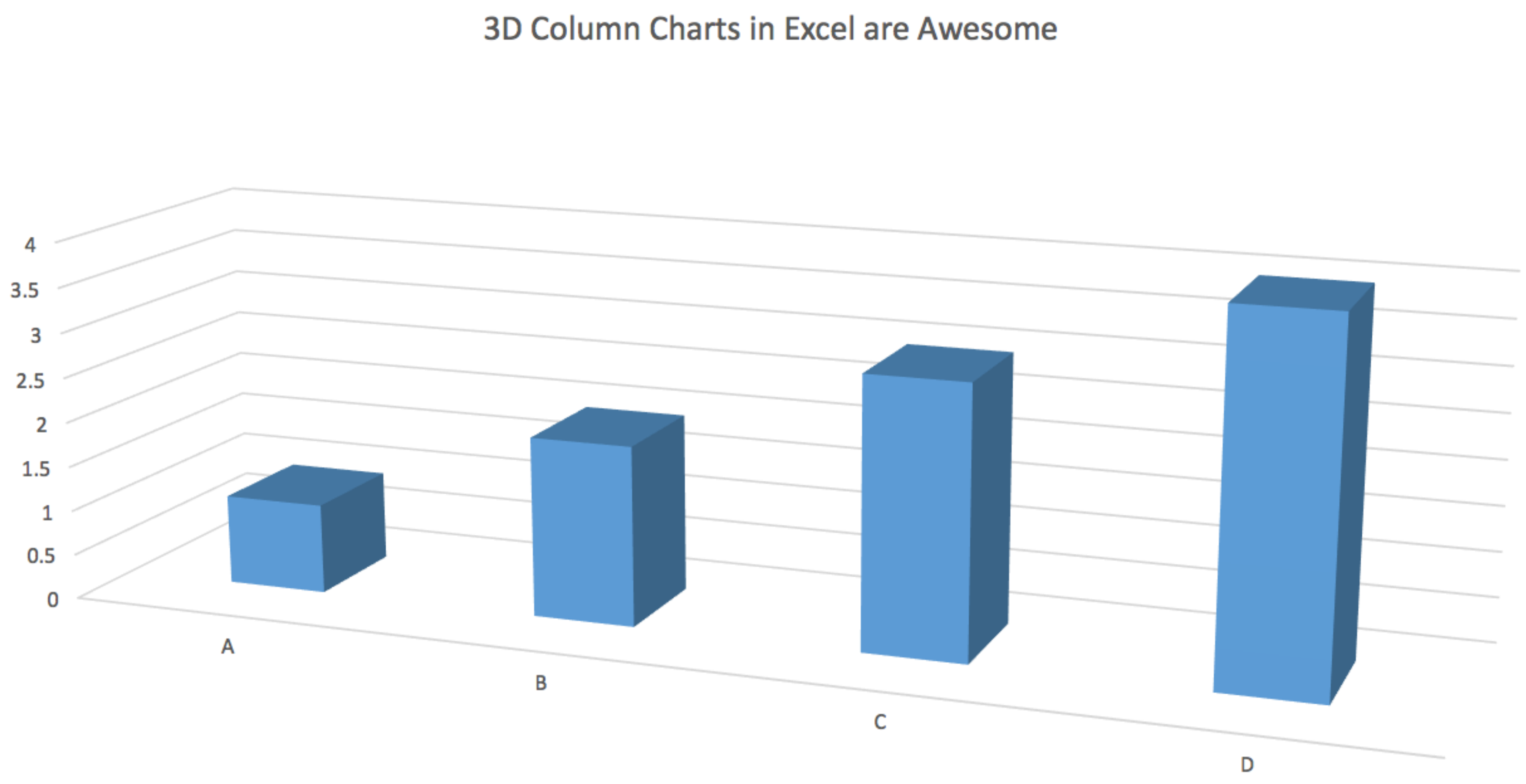
-

-

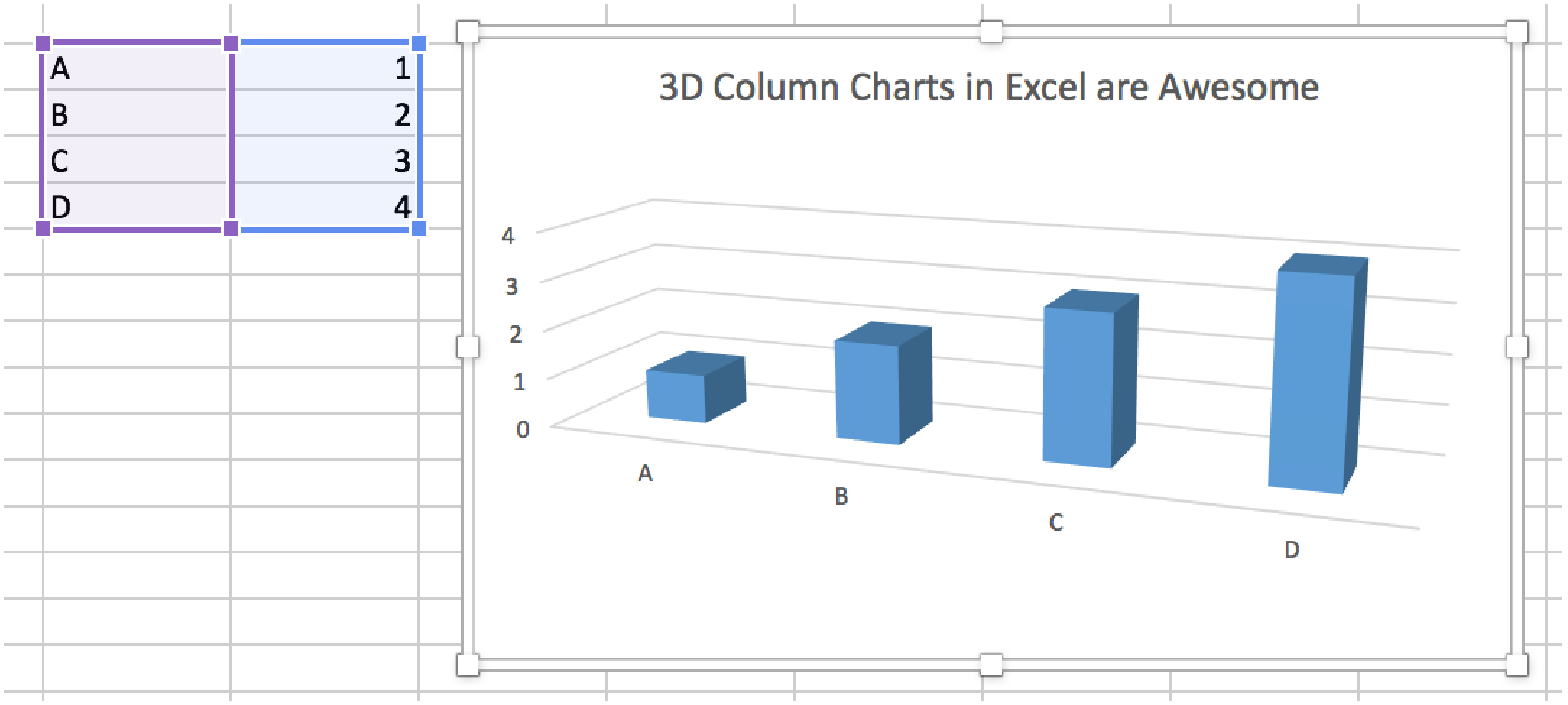
Agenda

Understanding plotting

What values do A,B,C,D have?



The shocking answer



A	1
B	2
C	3
D	4

Why are you plotting?

Who's the audience?

Others

- **Explanatory** plots: polished figures to convey your message

Yourself:

- **Exploratory** plots: fast for understanding data - minimal polishing.

How should you plot (1)

*What are some tips for making **explanatory** plots in a report?*

1. Self explanatory
   - Contain axis label, title, footnotes in text containing relevant information.
2. Eye candy
   - Choose the right plot type.
   - Make sure font size, colors, line width.
3. Narratives - should convey key point(s)
   - If you to show difference between groups in data make sure it is easy to distinguish them.
4. Keep simplicity.
   - Anything unnecessary should be removed, see this post (https://www.darkhorseanalytics.com/blog/data-looks-better-naked/).

How should you plot (2)

*What is some practical advice on making **explanatory** plots?*

1. Try out a few plot types, using exploratory analysis.
2. Apply the *"layered grammer of graphics"*.

   - Start with an empty canvas
   - Fill the necessary things (axis, ticks, bars/lines, labels)

How should you plot (3)

*What are some guidelines on making plots in **general**?*

Be aware of *what* you plot

- numerical vs. non-numeric (categorical)
- raw data vs. model results

Python plotting

Packages for Python plotting (1)

What is the fundamental tool for making plots in Python?

Matplotlib is the fundamental plotting module

- Can make almost any 2d plot.
- Can build publication ready figures.
- Caveat:
    - requires time consuming customization;
    - requires practice.

In [ ]:
```
import matplotlib.pyplot as plt

# allow printing in notebook
%matplotlib inline
```

Packages for Python plotting (2)

What are good tools for fast, exploratory plots?

`seaborn` has built-in capabilities to make plots

- Analyzing data, e.g. splitting by subsets
- Make interpolation of data to smooth noise.

`pandas` can easily convert Series and DataFrames to plots

In [ ]:
```
import pandas as pd
import seaborn as sns # high level plotting library
```

Packages for Python plotting (3)

Seaborn comes with some illustrative datasets. We load `iris` and `tips`.

In [ ]:
```
iris = sns.load_dataset('iris')
tips = sns.load_dataset('tips')
```

Plotting one numerical variable

The data

What does the `tips` data contain?

```
In [ ]: print(tips.head(3))
```

Univariate distribution (1)

How did we count categorical data?

- Using `value_counts`.

Can we do something similar with numeric data?

```
In [ ]: # cut into categorical data
        x = tips.total_bill
        cuts = np.arange(0, 70, 10)
        pd.cut(x, cuts).value_counts()
```

Univariate distribution (2)

How do we plot the distribution of numerical variables?

We often use the histogram.

- Bins data and counts observations
- Example of tips:

In [ ]:
```
histplot
```

Matplotlib and the grammar of graphics (1)

Where do I start with making a plot?

We will begin with the fundamental and flexible way. We start with our plotting canvas.

```
In [ ]: fig, ax = plt.subplots(figsize = (6, 2.5)) # create placeholder for plot
```

- ax contains most of the chart elements: the grid axes, labels, shapes we draw etc.
- `fig` the actual plot which is displayed (export to pdf etc.)

Matplotlib and the grammar of graphics (2)

We can modify our canvas, e.g the axis scaling:

In [ ]:
```
fig, ax = plt.subplots(figsize = (10, 4.5))
ax.set_xlim([0, 60]) # x-axis cutoffs
ax.set_ylim([0, 80]) # y-axis cutoffs
```

Matplotlib and the grammar of graphics (3)

We can draw plots on the canvas

```
In [ ]: fig, ax = plt.subplots(figsize = (10, 4.5))
        ax.set_xlim([0, 60])
        ax.set_ylim([0, 80])
        ax.hist(x) # make plot
```

Matplotlib and the grammar of graphics (4)

What might we change about our plot?

- We will try customization in the exercises today.

Matplotlib and the grammar of graphics (5)

Can we change matplotlib defaults?

Yes, this may be very useful. For instance plot size.

```
In [ ]:
```

```python
plt.style.use('default') # set style (colors, background, size, gridlines etc.)
plt.rcParams['figure.figsize'] = 10, 4 # set default size of plots
plt.rcParams.update({'font.size': 18})
```

Plotting with pandas

Pandas has a quick and dirty implemention. Let's try the code below.

```
In [ ]: x.head()
        #x.plot.hist()
```

Plotting with Seaborn (1)

The module Seaborn is great for fast plots that look good

```
In [ ]: sns.distplot(x)  # histogram for seaborn
```

Quiz: What is the line?

Plotting with Seaborn (2)

Can we use Seaborn for cumulative plots?

Yes, we specify `cumulative` in the keywords.

```
In [ ]: sns.distplot(x, hist_kws={'cumulative': True}, kde_kws={'cumulative': True})
```

Summing up

Group discussion (2 minutes):

- How did our tools perform?
- Which one seems most adequate for exploratory analysis? Which one for explanatory?
- Which steps could be taken towards improving our histograms?

Plotting one categorical variable

Univariate categorical

What is categorical data? How can we plot categorical data?

Pies are possible but of little use. Let's plot this with bars:

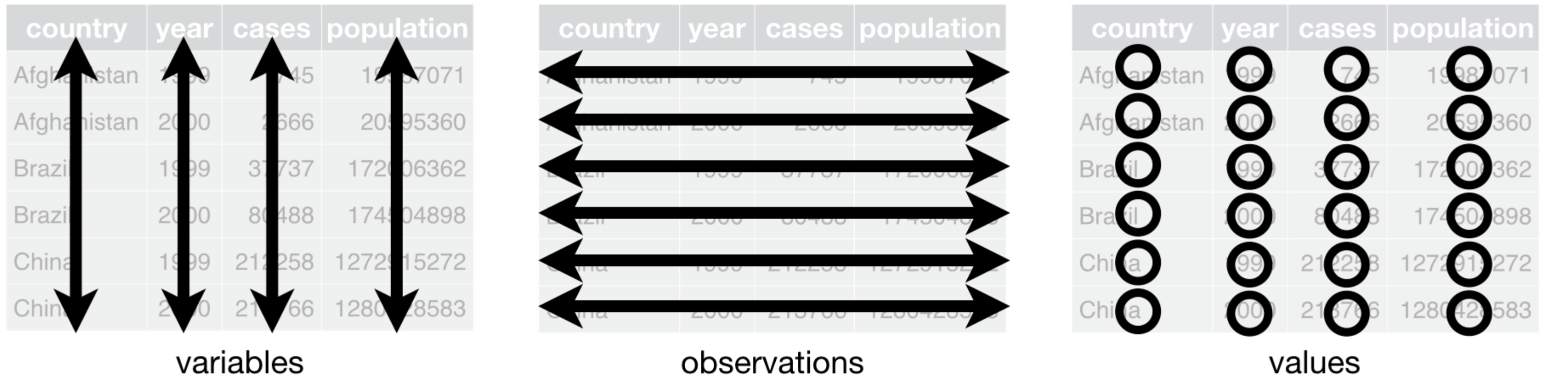
```
In [ ]: sns.countplot(x='sex', data=tips)
```

Plotting DataFrames

Table format

How did we define a tidy/long table?

One row for each observation



Plots of two numeric variables

Two numeric variables (1)

How do we plot two numeric variables?

If we have little data we can make a point cloud, i.e. a scatter plot.

```
In [ ]: plt.scatter(x=tips['total_bill'], y=tips['tip'])
```

Two numeric variables (2)

Quiz: How might we alter the scatter plot?

We can interpolate the data:

```
In [ ]: sns.jointplot(x='total_bill', y='tip', data=tips, kind='hex', size=5) # hex
```

Two numeric variables (3)

What if we want to see the linear relationship?

We use the linear model plot:

```
In [ ]: sns.lmplot(x='total_bill', y='tip', data=tips, size=5, aspect=2)
```

Plots with categorical variables

Mixed types - numeric, categorical (1)

Quiz: What is tidy format?

- One row per observation

How might we use categorical variables?

- We can split data!

In which plots might this be useful?

- We can compute mean for each categorical variables, the `barplot`.
- We can compute quartiles for each categorical variables, the `boxplot`.

Mixed types - numeric, categorical (2)

Let's make a plot the mean tips - distinguish by weekday:

```
In [ ]: f = sns.barplot(x='day', y='tip', data=tips)
```

Mixed types - numeric, categorical (2)

Let's make a plot the tip quartiles - distinguish by sex:

```
In [ ]: f = sns.boxplot(x='sex', y='tip', data=tips)
```

Advanced exploratory plotting

Plot grids (1)

How can we we plot the relationship for more than two variables?

In [ ]:
```
# A powerful method:
sns.pairplot(tips, size=1.5, aspect=1.6)
```

Plot grids (2)

Can we split the data to investigate heterogeneous relationships?

Yes, let's starting building a FacetGrid:

```
In [ ]:
g = sns.FacetGrid(tips)
g = g.map(sns.regplot, 'total_bill', 'tip')
```

Plot grids (3)

Let's try to add distinctive slopes for smoker

```
In [ ]:  g = sns.FacetGrid(tips, col='smoker') # time
         g = g.map(sns.regplot, 'total_bill', 'tip')
```

Can we say anything about smokers tipping behavior?

The end

Return to Agenda