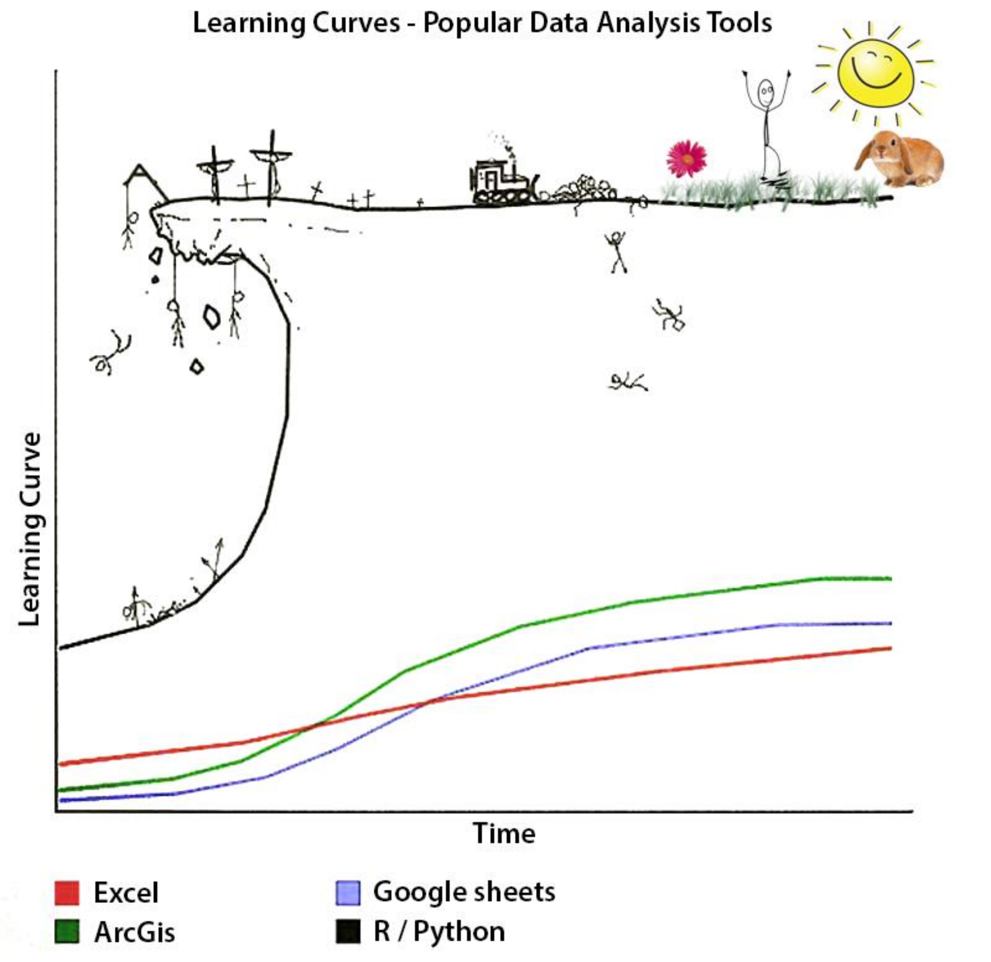
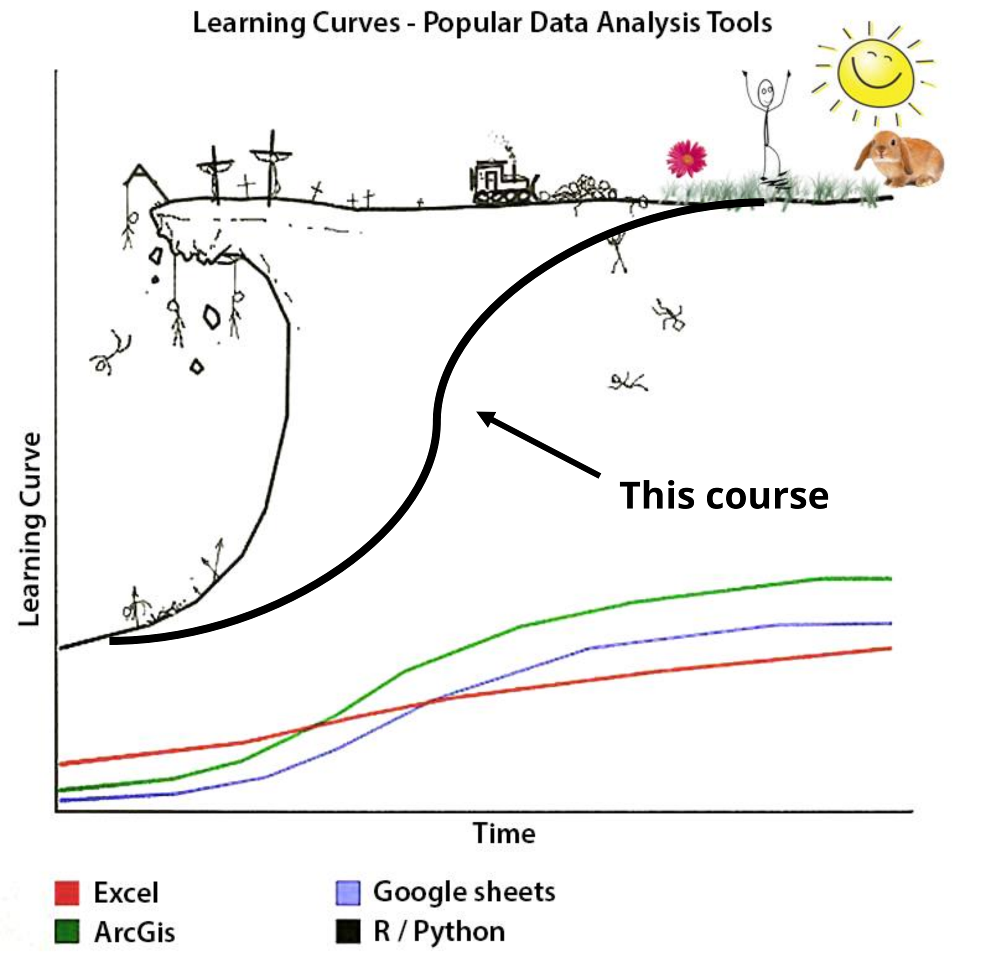
Introduction to Data Science with Python

Chapter 0

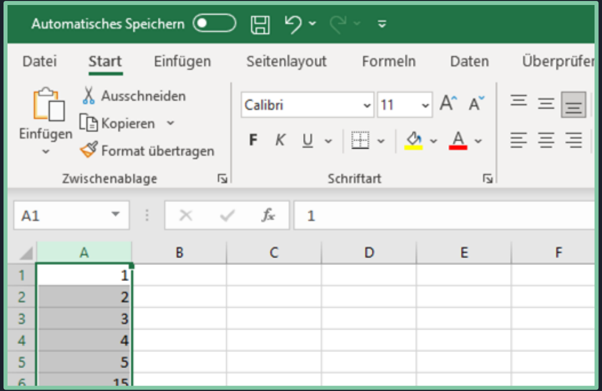
July 28, 2023


Learning Curves - Popular Data Analysis Tools
Learning Curve
Time
🟥 Excel
🟩 ArcGis
🟦 Google sheets
⬛ R / Python


Learning Curves - Popular Data Analysis Tools
Learning Curve
Time
This course
🟥 Excel
🟩 ArcGis
🟦 Google sheets
⬛ R / Python

Excel vs Python

1. Select data
2. Click on buttons :)

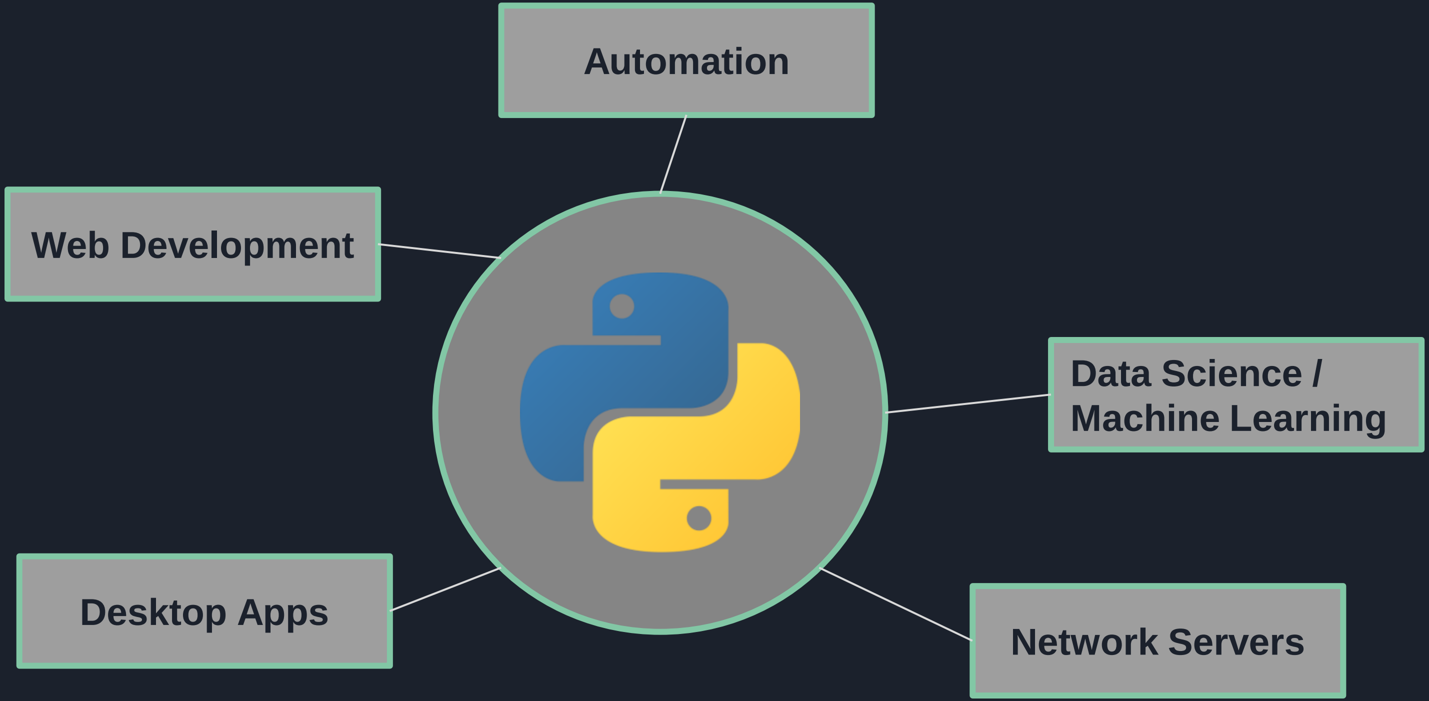


	A	B	C	D	E	F
1	1					
2	2					
3	3					
4	4					
5	5					
6	15					

1. Write code in editor
2. Execute code with Python
3. Result will be returned

```
data = pd.read_csv(file)
mean = data.mean()
print(mean)
```

A General Purpose Coding Language



Automation

Web Development

Data Science / Machine Learning

Desktop Apps

Network Servers

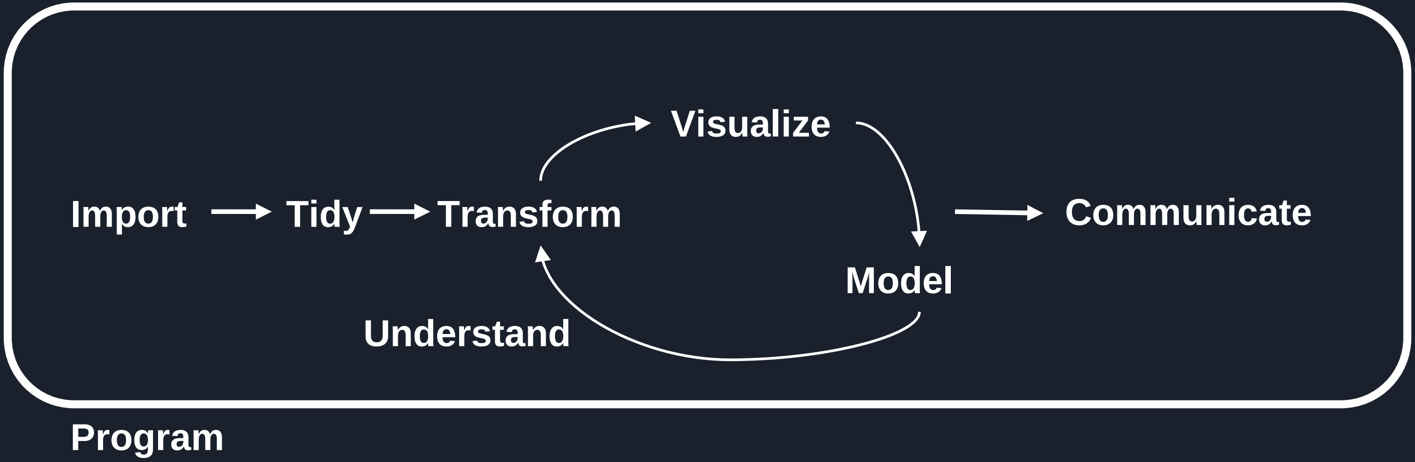
Software

Google Colab

- Write and execute code
- Accessed via Browser (runs on Google Servers)
- No pre-configurations necessary
- Independent of your local machine
- Jupyter Notebook format heavily used in data science community

Data science life cycle

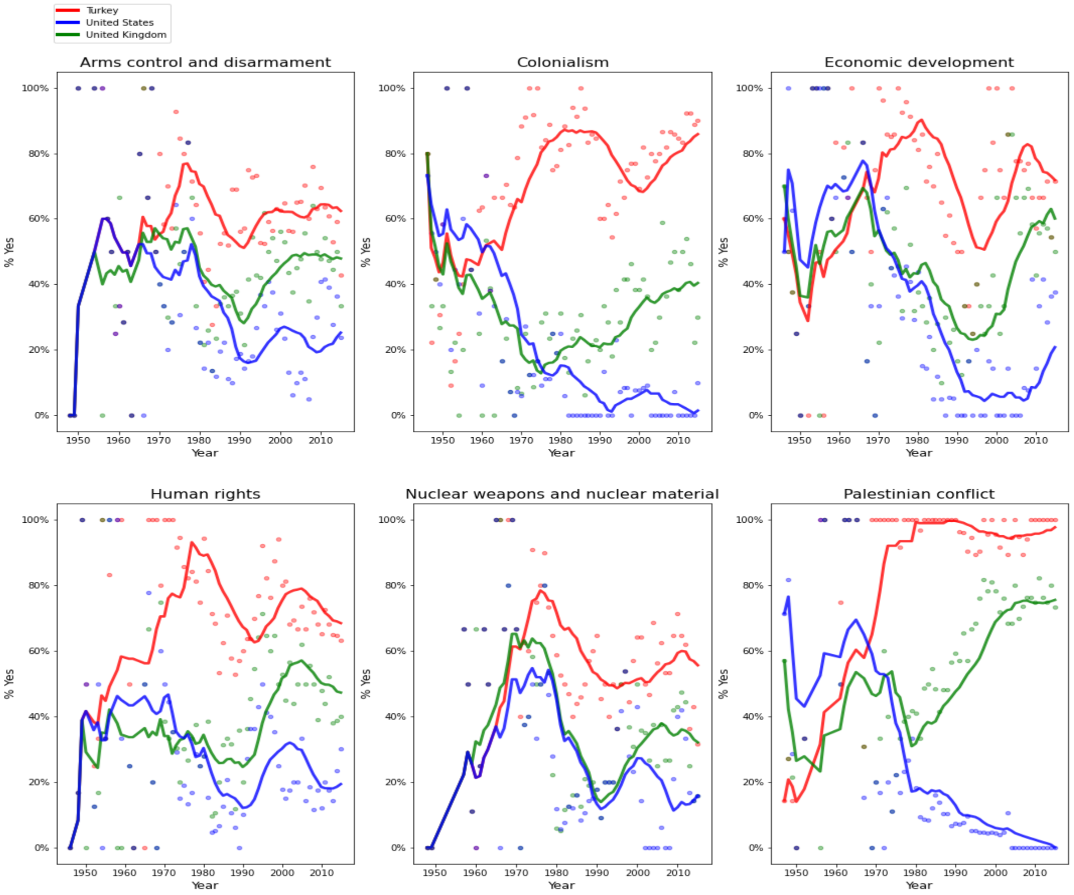
We‘ll walk you through the data science lifecycle and introduce the tools for each step

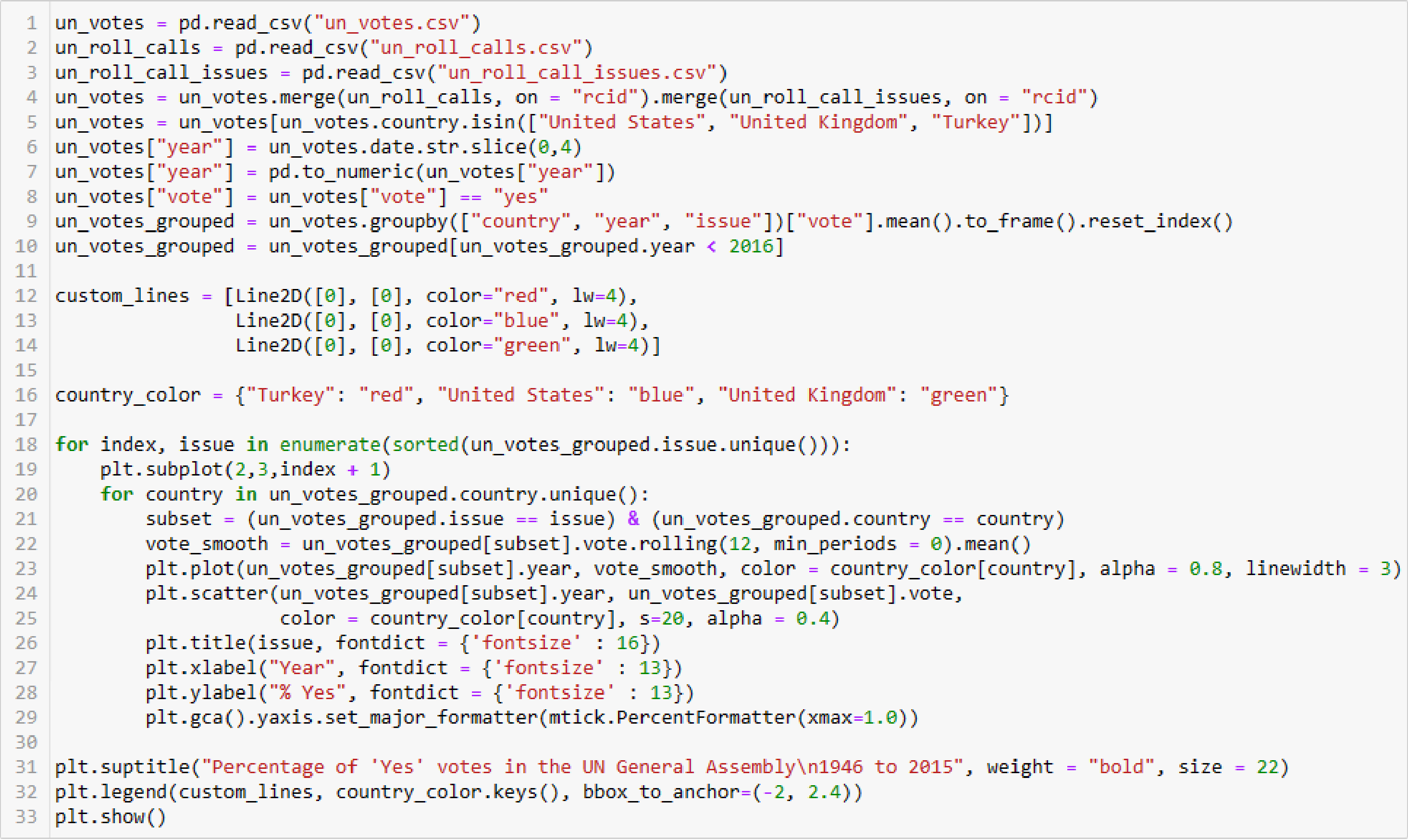


Let‘s dive in!

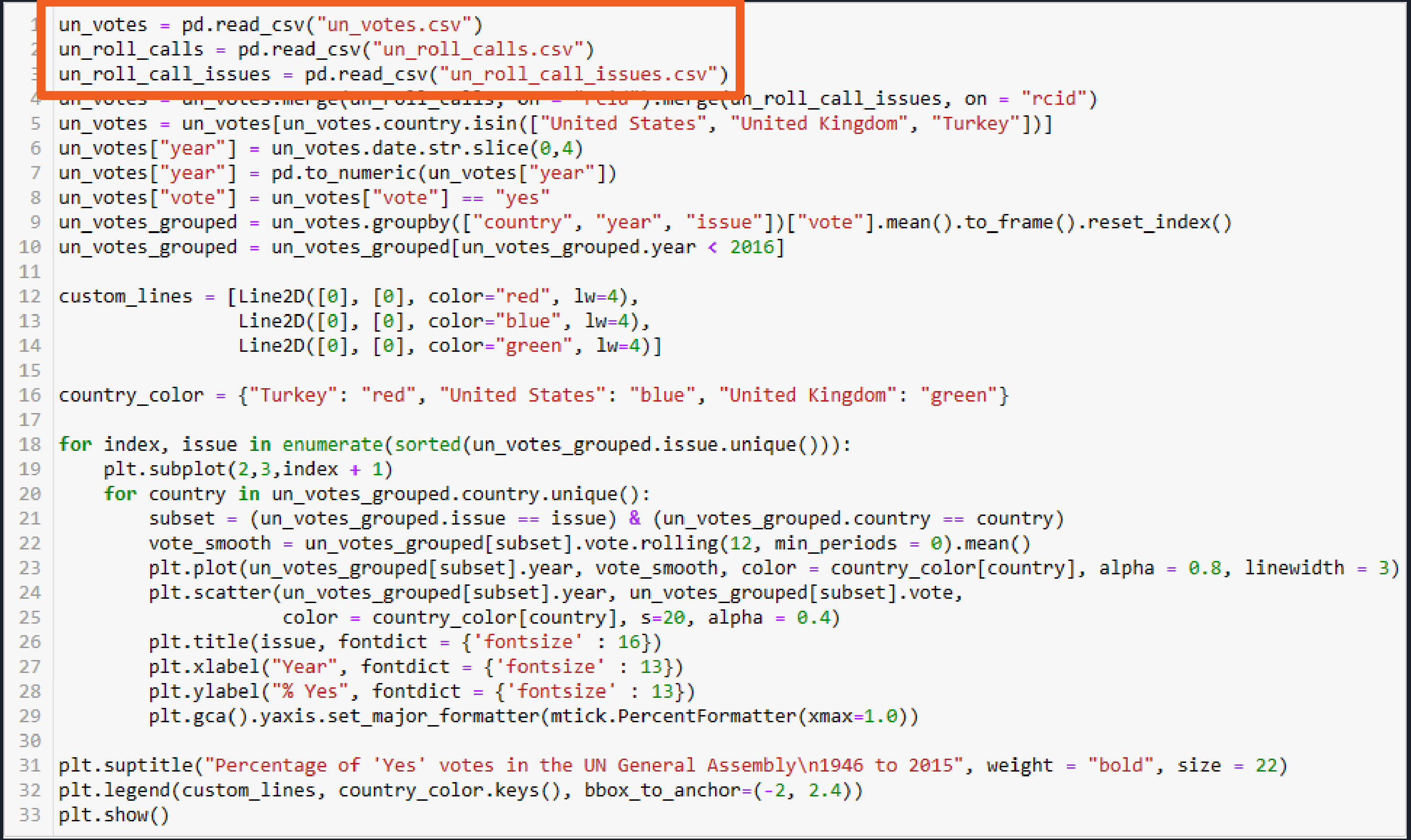
Based on datacience.org

Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015

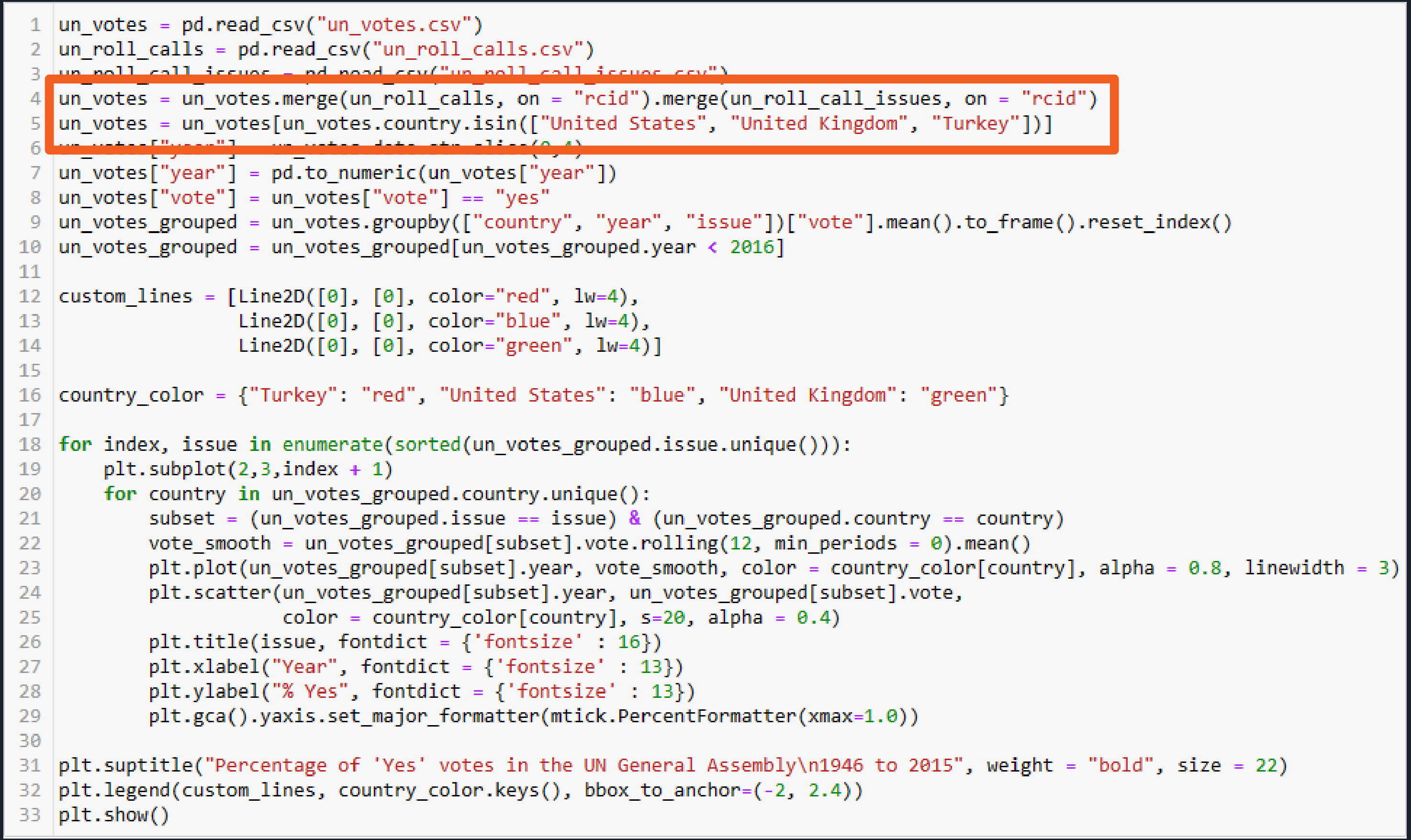




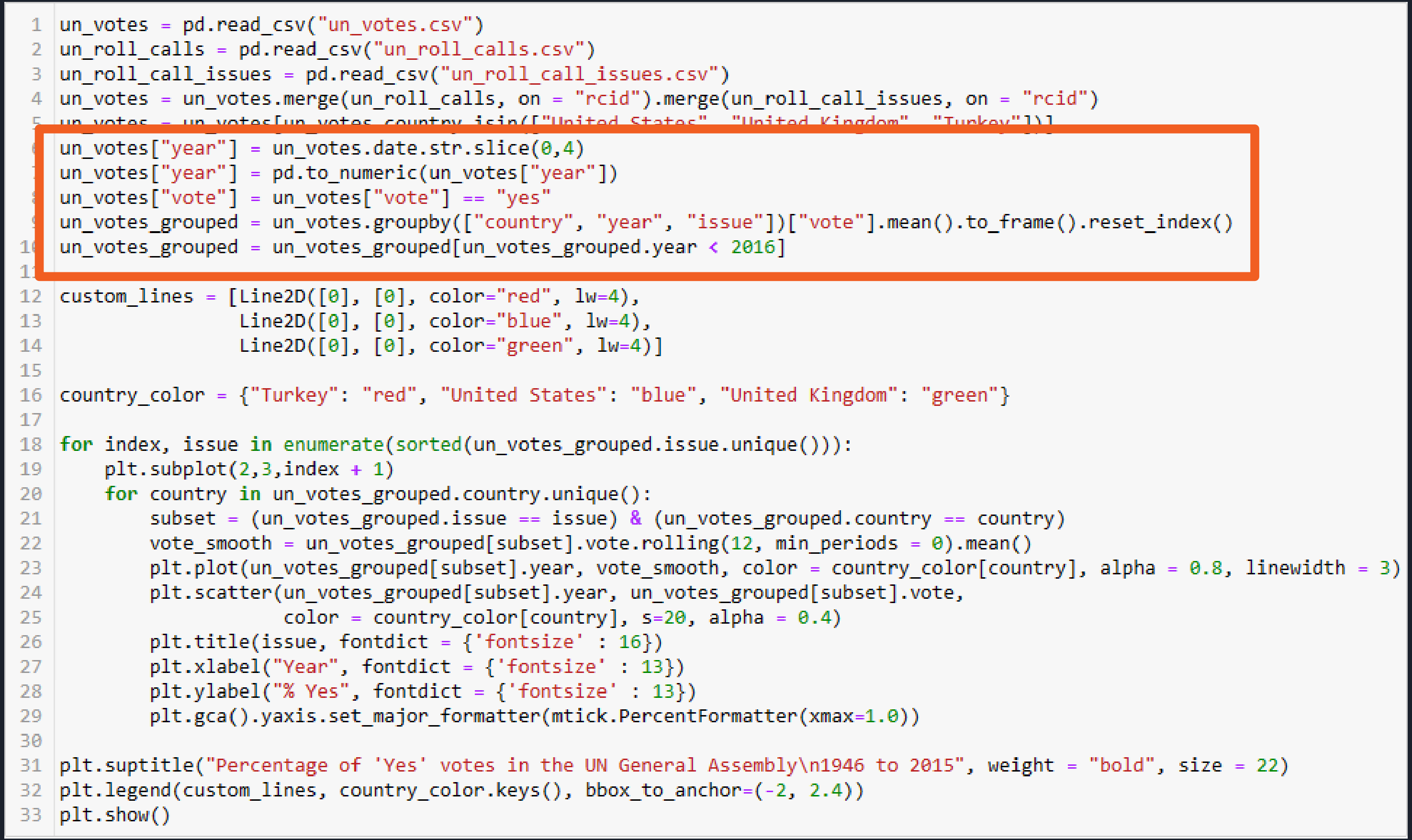
```python
un_votes = pd.read_csv("un_votes.csv")
un_roll_calls = pd.read_csv("un_roll_calls.csv")
un_roll_call_issues = pd.read_csv("un_roll_call_issues.csv")
un_votes = un_votes.merge(un_roll_calls, on = "rcid").merge(un_roll_call_issues, on = "rcid")
un_votes = un_votes[un_votes.country.isin(["United States", "United Kingdom", "Turkey"])]
un_votes["year"] = un_votes.date.str.slice(0,4)
un_votes["year"] = pd.to_numeric(un_votes["year"])
un_votes["vote"] = un_votes["vote"] == "yes"
un_votes_grouped = un_votes.groupby(["country", "year", "issue"])["vote"].mean().to_frame().reset_index()
un_votes_grouped = un_votes_grouped[un_votes_grouped.year < 2016]

custom_lines = [Line2D([0], [0], color="red", lw=4),
                Line2D([0], [0], color="blue", lw=4),
                Line2D([0], [0], color="green", lw=4)]

country_color = {"Turkey": "red", "United States": "blue", "United Kingdom": "green"}

for index, issue in enumerate(sorted(un_votes_grouped.issue.unique())):
    plt.subplot(2,3,index + 1)
    for country in un_votes_grouped.country.unique():
        subset = (un_votes_grouped.issue == issue) & (un_votes_grouped.country == country)
        vote_smooth = un_votes_grouped[subset].vote.rolling(12, min_periods = 0).mean()
        plt.plot(un_votes_grouped[subset].year, vote_smooth, color = country_color[country], alpha = 0.8, linewidth = 3)
        plt.scatter(un_votes_grouped[subset].year, un_votes_grouped[subset].vote,
                    color = country_color[country], s=20, alpha = 0.4)
        plt.title(issue, fontdict = {'fontsize' : 16})
        plt.xlabel("Year", fontdict = {'fontsize' : 13})
        plt.ylabel("% Yes", fontdict = {'fontsize' : 13})
        plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1.0))

plt.suptitle("Percentage of 'Yes' votes in the UN General Assembly\n1946 to 2015", weight = "bold", size = 22)
plt.legend(custom_lines, country_color.keys(), bbox_to_anchor=(-2, 2.4))
plt.show()
```



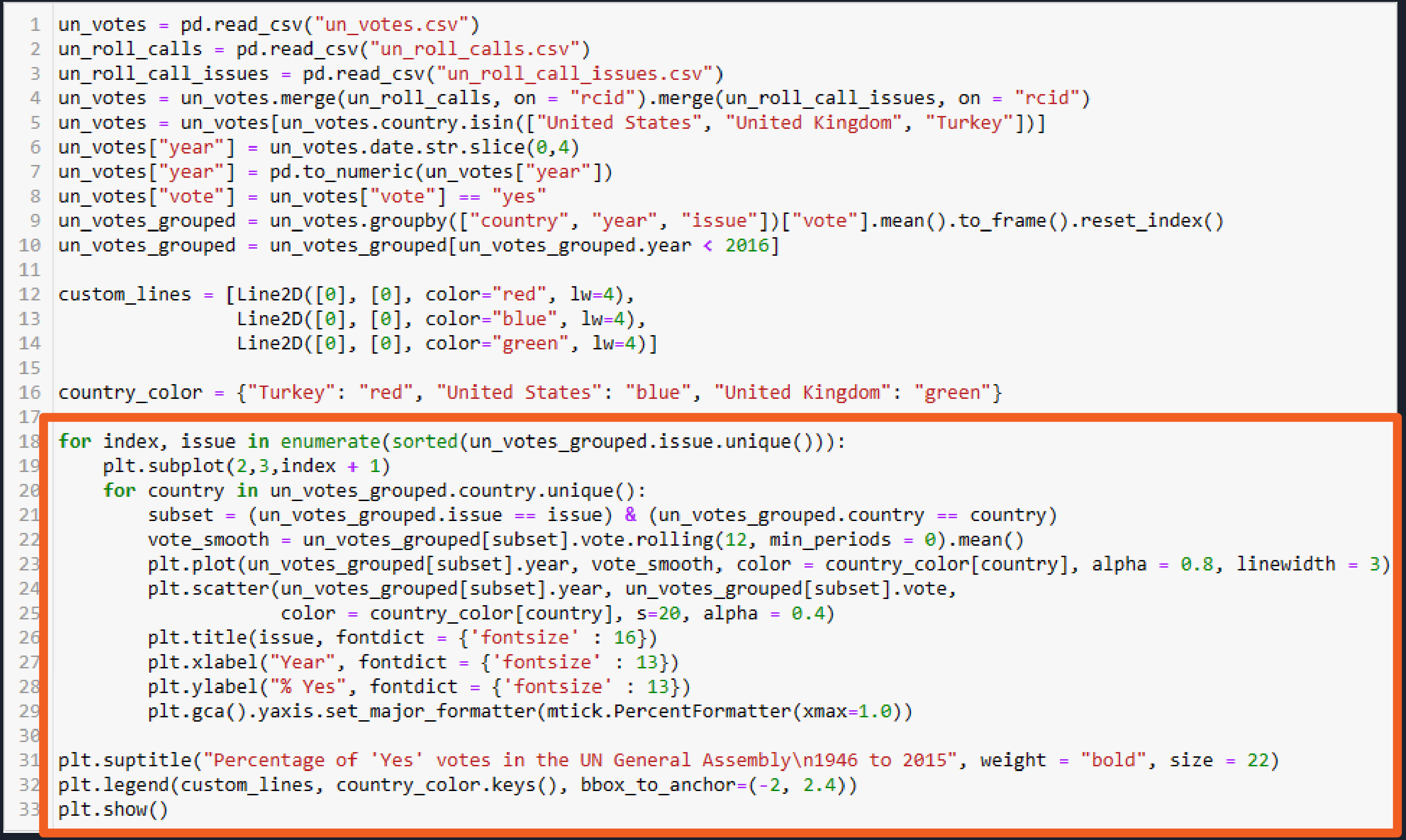
```python
un_votes = pd.read_csv("un_votes.csv")
un_roll_calls = pd.read_csv("un_roll_calls.csv")
un_roll_call_issues = pd.read_csv("un_roll_call_issues.csv")
un_votes = un_votes.merge(un_roll_calls, on = "rcid").merge(un_roll_call_issues, on = "rcid")
un_votes = un_votes[un_votes.country.isin(["United States", "United Kingdom", "Turkey"])]
un_votes["year"] = un_votes.date.str.slice(0,4)
un_votes["year"] = pd.to_numeric(un_votes["year"])
un_votes["vote"] = un_votes["vote"] == "yes"
un_votes_grouped = un_votes.groupby(["country", "year", "issue"])["vote"].mean().to_frame().reset_index()
un_votes_grouped = un_votes_grouped[un_votes_grouped.year < 2016]

custom_lines = [Line2D([0], [0], color="red", lw=4),
                Line2D([0], [0], color="blue", lw=4),
                Line2D([0], [0], color="green", lw=4)]

country_color = {"Turkey": "red", "United States": "blue", "United Kingdom": "green"}

for index, issue in enumerate(sorted(un_votes_grouped.issue.unique())):
    plt.subplot(2,3,index + 1)
    for country in un_votes_grouped.country.unique():
        subset = (un_votes_grouped.issue == issue) & (un_votes_grouped.country == country)
        vote_smooth = un_votes_grouped[subset].vote.rolling(12, min_periods = 0).mean()
        plt.plot(un_votes_grouped[subset].year, vote_smooth, color = country_color[country], alpha = 0.8, linewidth = 3)
        plt.scatter(un_votes_grouped[subset].year, un_votes_grouped[subset].vote,
                    color = country_color[country], s=20, alpha = 0.4)
        plt.title(issue, fontdict = {'fontsize' : 16})
        plt.xlabel("Year", fontdict = {'fontsize' : 13})
        plt.ylabel("% Yes", fontdict = {'fontsize' : 13})
        plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1.0))

plt.suptitle("Percentage of 'Yes' votes in the UN General Assembly\n1946 to 2015", weight = "bold", size = 22)
plt.legend(custom_lines, country_color.keys(), bbox_to_anchor=(-2, 2.4))
plt.show()
```



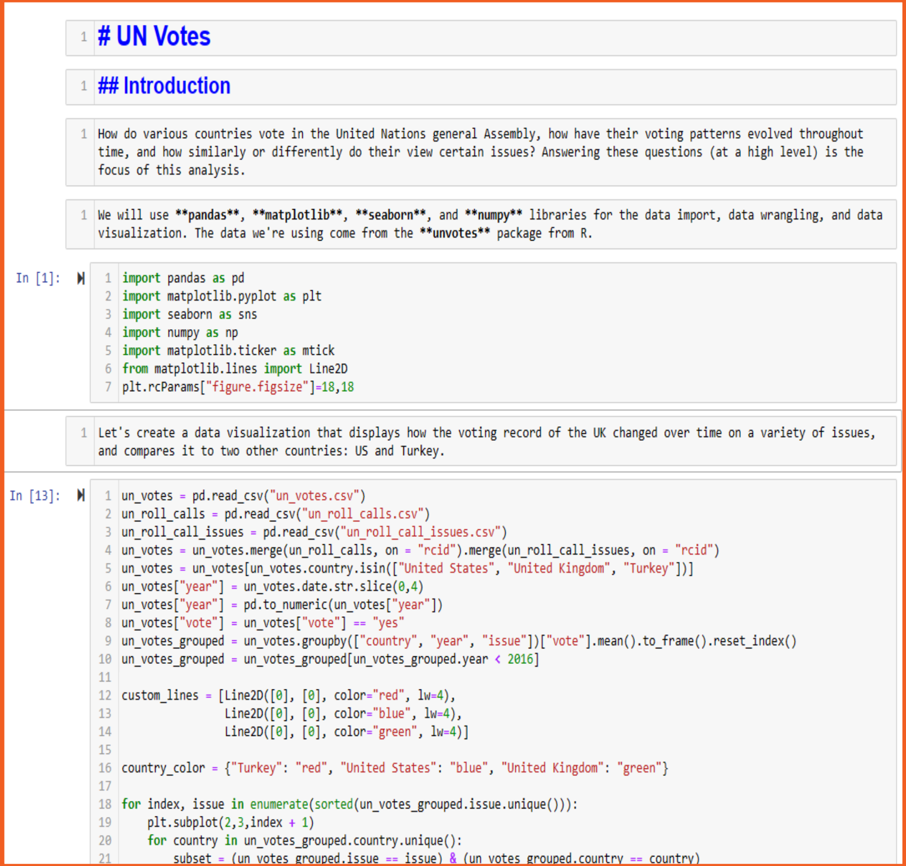
```python
un_votes = pd.read_csv("un_votes.csv")
un_roll_calls = pd.read_csv("un_roll_calls.csv")
un_roll_call_issues = pd.read_csv("un_roll_call_issues.csv")
un_votes = un_votes.merge(un_roll_calls, on = "rcid").merge(un_roll_call_issues, on = "rcid")
un_votes = un_votes[un_votes.country.isin(["United States", "United Kingdom", "Turkey"])]
un_votes["year"] = un_votes.date.str.slice(0,4)
un_votes["year"] = pd.to_numeric(un_votes["year"])
un_votes["vote"] = un_votes["vote"] == "yes"
un_votes_grouped = un_votes.groupby(["country", "year", "issue"])["vote"].mean().to_frame().reset_index()
un_votes_grouped = un_votes_grouped[un_votes_grouped.year < 2016]

custom_lines = [Line2D([0], [0], color="red", lw=4),
                Line2D([0], [0], color="blue", lw=4),
                Line2D([0], [0], color="green", lw=4)]

country_color = {"Turkey": "red", "United States": "blue", "United Kingdom": "green"}

for index, issue in enumerate(sorted(un_votes_grouped.issue.unique())):
    plt.subplot(2,3,index + 1)
    for country in un_votes_grouped.country.unique():
        subset = (un_votes_grouped.issue == issue) & (un_votes_grouped.country == country)
        vote_smooth = un_votes_grouped[subset].vote.rolling(12, min_periods = 0).mean()
        plt.plot(un_votes_grouped[subset].year, vote_smooth, color = country_color[country], alpha = 0.8, linewidth = 3)
        plt.scatter(un_votes_grouped[subset].year, un_votes_grouped[subset].vote,
                    color = country_color[country], s=20, alpha = 0.4)
        plt.title(issue, fontdict = {'fontsize' : 16})
        plt.xlabel("Year", fontdict = {'fontsize' : 13})
        plt.ylabel("% Yes", fontdict = {'fontsize' : 13})
        plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1.0))

plt.suptitle("Percentage of 'Yes' votes in the UN General Assembly\n1946 to 2015", weight = "bold", size = 22)
plt.legend(custom_lines, country_color.keys(), bbox_to_anchor=(-2, 2.4))
plt.show()
```



```python
un_votes = pd.read_csv("un_votes.csv")
un_roll_calls = pd.read_csv("un_roll_calls.csv")
un_roll_call_issues = pd.read_csv("un_roll_call_issues.csv")
un_votes = un_votes.merge(un_roll_calls, on = "rcid").merge(un_roll_call_issues, on = "rcid")
un_votes = un_votes[un_votes.country.isin(["United States", "United Kingdom", "Turkey"])]
un_votes["year"] = un_votes.date.str.slice(0,4)
un_votes["year"] = pd.to_numeric(un_votes["year"])
un_votes["vote"] = un_votes["vote"] == "yes"
un_votes_grouped = un_votes.groupby(["country", "year", "issue"])["vote"].mean().to_frame().reset_index()
un_votes_grouped = un_votes_grouped[un_votes_grouped.year < 2016]

custom_lines = [Line2D([0], [0], color="red", lw=4),
                Line2D([0], [0], color="blue", lw=4),
                Line2D([0], [0], color="green", lw=4)]

country_color = {"Turkey": "red", "United States": "blue", "United Kingdom": "green"}

for index, issue in enumerate(sorted(un_votes_grouped.issue.unique())):
    plt.subplot(2,3,index + 1)
    for country in un_votes_grouped.country.unique():
        subset = (un_votes_grouped.issue == issue) & (un_votes_grouped.country == country)
        vote_smooth = un_votes_grouped[subset].vote.rolling(12, min_periods = 0).mean()
        plt.plot(un_votes_grouped[subset].year, vote_smooth, color = country_color[country], alpha = 0.8, linewidth = 3)
        plt.scatter(un_votes_grouped[subset].year, un_votes_grouped[subset].vote,
                    color = country_color[country], s=20, alpha = 0.4)
        plt.title(issue, fontdict = {'fontsize' : 16})
        plt.xlabel("Year", fontdict = {'fontsize' : 13})
        plt.ylabel("% Yes", fontdict = {'fontsize' : 13})
        plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1.0))

plt.suptitle("Percentage of 'Yes' votes in the UN General Assembly\n1946 to 2015", weight = "bold", size = 22)
plt.legend(custom_lines, country_color.keys(), bbox_to_anchor=(-2, 2.4))
plt.show()
```



```python
un_votes = pd.read_csv("un_votes.csv")
un_roll_calls = pd.read_csv("un_roll_calls.csv")
un_roll_call_issues = pd.read_csv("un_roll_call_issues.csv")
un_votes = un_votes.merge(un_roll_calls, on = "rcid").merge(un_roll_call_issues, on = "rcid")
un_votes = un_votes[un_votes.country.isin(["United States", "United Kingdom", "Turkey"])]
un_votes["year"] = un_votes.date.str.slice(0,4)
un_votes["year"] = pd.to_numeric(un_votes["year"])
un_votes["vote"] = un_votes["vote"] == "yes"
un_votes_grouped = un_votes.groupby(["country", "year", "issue"])["vote"].mean().to_frame().reset_index()
un_votes_grouped = un_votes_grouped[un_votes_grouped.year < 2016]

custom_lines = [Line2D([0], [0], color="red", lw=4),
                Line2D([0], [0], color="blue", lw=4),
                Line2D([0], [0], color="green", lw=4)]

country_color = {"Turkey": "red", "United States": "blue", "United Kingdom": "green"}

for index, issue in enumerate(sorted(un_votes_grouped.issue.unique())):
    plt.subplot(2,3,index + 1)
    for country in un_votes_grouped.country.unique():
        subset = (un_votes_grouped.issue == issue) & (un_votes_grouped.country == country)
        vote_smooth = un_votes_grouped[subset].vote.rolling(12, min_periods = 0).mean()
        plt.plot(un_votes_grouped[subset].year, vote_smooth, color = country_color[country], alpha = 0.8, linewidth = 3)
        plt.scatter(un_votes_grouped[subset].year, un_votes_grouped[subset].vote,
                    color = country_color[country], s=20, alpha = 0.4)
        plt.title(issue, fontdict = {'fontsize' : 16})
        plt.xlabel("Year", fontdict = {'fontsize' : 13})
        plt.ylabel("% Yes", fontdict = {'fontsize' : 13})
        plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1.0))

plt.suptitle("Percentage of 'Yes' votes in the UN General Assembly\n1946 to 2015", weight = "bold", size = 22)
plt.legend(custom_lines, country_color.keys(), bbox_to_anchor=(-2, 2.4))
plt.show()
```



UN Votes

Introduction

How do various countries vote in the United Nations general Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do their view certain issues? Answering these questions (at a high level) is the focus of this analysis.

We will use **pandas**, **matplotlib**, **seaborn**, and **numpy** libraries for the data import, data wrangling, and data visualization. The data we're using come from the **unvotes** package from R.

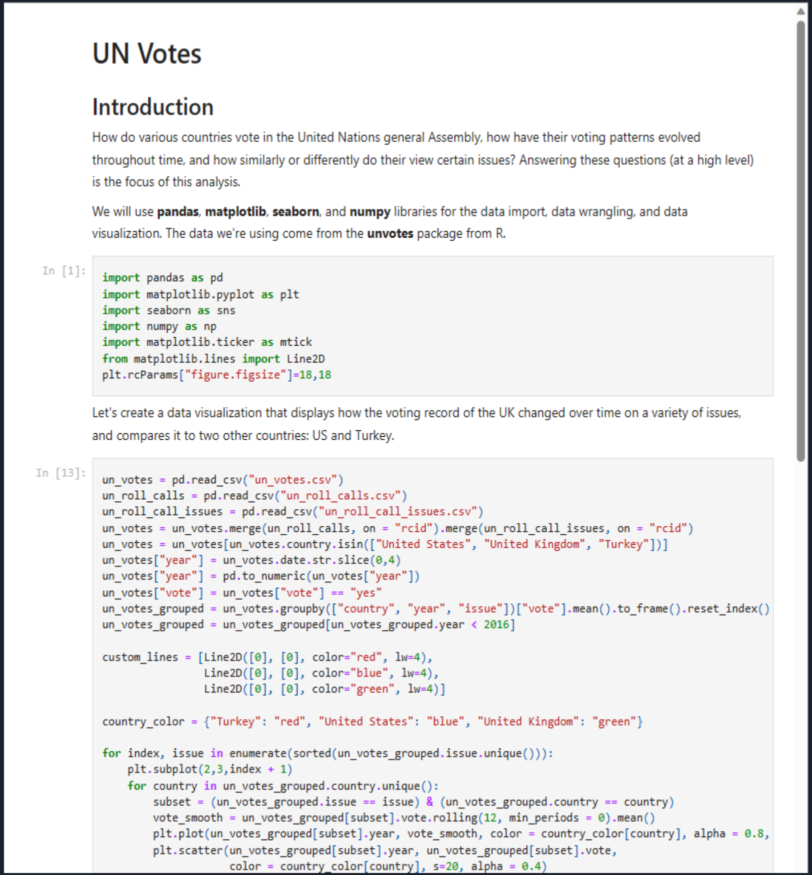
In [1]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import matplotlib.ticker as mtick
from matplotlib.lines import Line2D
plt.rcParams["figure.figsize"]=18,18
```

Let's create a data visualization that displays how the voting record of the UK changed over time on a variety of issues, and compares it to two other countries: US and Turkey.

In [13]:

```python
un_votes = pd.read_csv("un_votes.csv")
un_roll_calls = pd.read_csv("un_roll_calls.csv")
un_roll_call_issues = pd.read_csv("un_roll_call_issues.csv")
un_votes = un_votes.merge(un_roll_calls, on = "rcid").merge(un_roll_call_issues, on = "rcid")
un_votes = un_votes[un_votes.country.isin(["United States", "United Kingdom", "Turkey"])]
un_votes["year"] = un_votes.date.str.slice(0,4)
un_votes["year"] = pd.to_numeric(un_votes["year"])
un_votes["vote"] = un_votes["vote"] == "yes"
un_votes_grouped = un_votes.groupby(["country", "year", "issue"])["vote"].mean().to_frame().reset_index()
un_votes_grouped = un_votes_grouped[un_votes_grouped.year < 2016]

custom_lines = [Line2D([0], [0], color="red", lw=4),
                Line2D([0], [0], color="blue", lw=4),
                Line2D([0], [0], color="green", lw=4)]

country_color = {"Turkey": "red", "United States": "blue", "United Kingdom": "green"}

for index, issue in enumerate(sorted(un_votes_grouped.issue.unique())):
    plt.subplot(2,3,index + 1)
    for country in un_votes_grouped.country.unique():
        subset = (un_votes_grouped.issue == issue) & (un_votes_grouped.country == country)
```



UN Votes

Introduction

How do various countries vote in the United Nations general Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do their view certain issues? Answering these questions (at a high level) is the focus of this analysis.

We will use **pandas**, **matplotlib**, **seaborn**, and **numpy** libraries for the data import, data wrangling, and data visualization. The data we're using come from the **unvotes** package from R.

In [1]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import matplotlib.ticker as mtick
from matplotlib.lines import Line2D
plt.rcParams["figure.figsize"]=18,18
```

Let's create a data visualization that displays how the voting record of the UK changed over time on a variety of issues, and compares it to two other countries: US and Turkey.

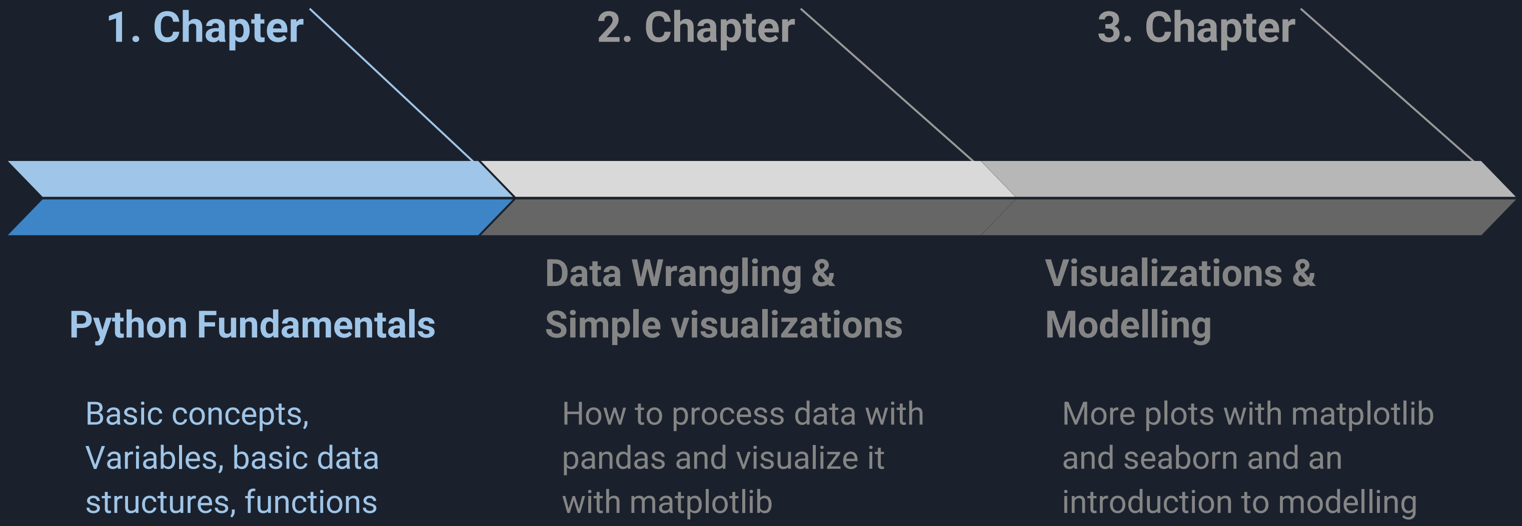
In [13]:

```python
un_votes = pd.read_csv("un_votes.csv")
un_roll_calls = pd.read_csv("un_roll_calls.csv")
un_roll_call_issues = pd.read_csv("un_roll_call_issues.csv")
un_votes = un_votes.merge(un_roll_calls, on = "rcid").merge(un_roll_call_issues, on = "rcid")
un_votes = un_votes[un_votes.country.isin(["United States", "United Kingdom", "Turkey"])]
un_votes["year"] = un_votes.date.str.slice(0,4)
un_votes["year"] = pd.to_numeric(un_votes["year"])
un_votes["vote"] = un_votes["vote"] == "yes"
un_votes_grouped = un_votes.groupby(["country", "year", "issue"])["vote"].mean().to_frame().reset_index()
un_votes_grouped = un_votes_grouped[un_votes_grouped.year < 2016]

custom_lines = [Line2D([0], [0], color="red", lw=4),
                Line2D([0], [0], color="blue", lw=4),
                Line2D([0], [0], color="green", lw=4)]

country_color = {"Turkey": "red", "United States": "blue", "United Kingdom": "green"}

for index, issue in enumerate(sorted(un_votes_grouped.issue.unique())):
    plt.subplot(2,3,index + 1)
    for country in un_votes_grouped.country.unique():
        subset = (un_votes_grouped.issue == issue) & (un_votes_grouped.country == country)
        vote_smooth = un_votes_grouped[subset].vote.rolling(12, min_periods = 0).mean()
        plt.plot(un_votes_grouped[subset].year, vote_smooth, color = country_color[country], alpha = 0.8,
        plt.scatter(un_votes_grouped[subset].year, un_votes_grouped[subset].vote,
                    color = country_color[country], s=20, alpha = 0.4)
```

Is it possible to learn all that in this course?

Yes, if you‘re actively coding along and invest some time.

We‘ll go through every step in this course:



1. Chapter

Python Fundamentals

Basic concepts, Variables, basic data structures, functions

2. Chapter

Data Wrangling & Simple visualizations

How to process data with pandas and visualize it with matplotlib

3. Chapter

Visualizations & Modelling

More plots with matplotlib and seaborn and an introduction to modelling

Structure of the course

For each of 3 Chapters:

- Introduction of new concepts
- Your turn! - Small exercises (~5 minutes)
- Live coding

Contact Us!
kraft.thilo.g@gmail.com &
jan.bischoff@tech-academy.io