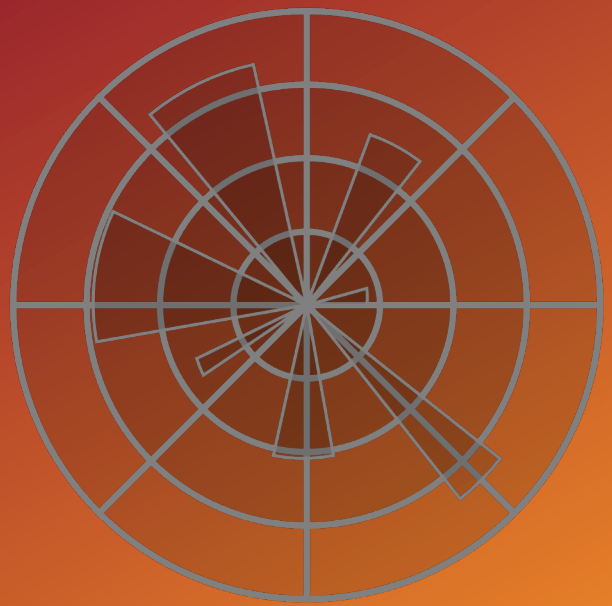
CSE 4510/5310
Big Data

Instructor: Fitzroy Nembhard, Ph.D.

Using Pandas, Matplotlib and Numpy to Preprocess Data



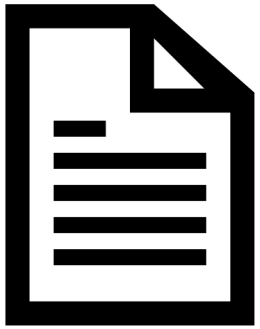
FLORIDA TECH
FLORIDA'S STEM UNIVERSITY®

Download the following datasets from Canvas

ted_main.csv

transcripts.csv

ted_main.csv

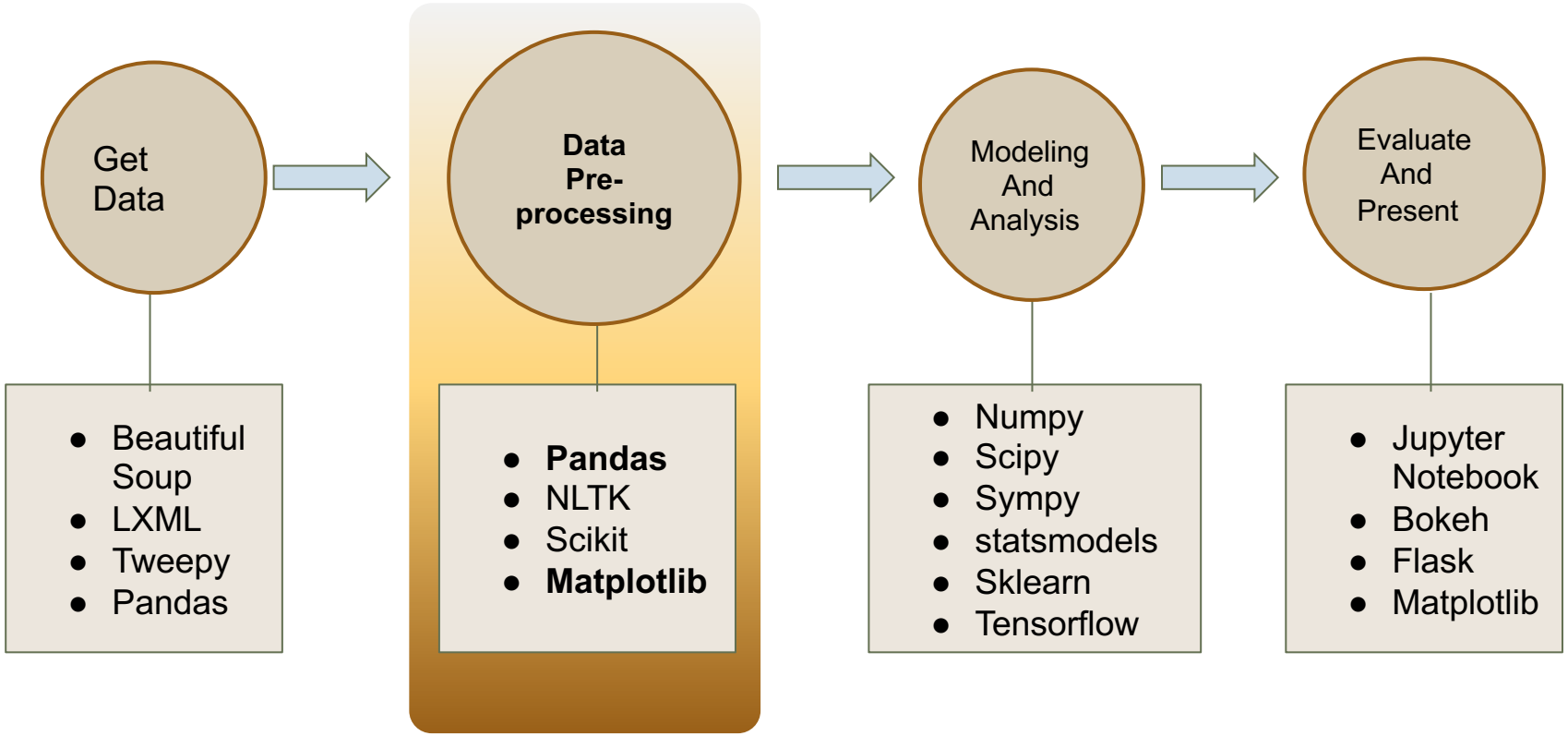


transcripts.csv

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

EDA/Preprocessing

- The goal of this activity is to practice EDA with a given dataset(s)



A view of the Datasets from Excel

On the right is a screenshot of ted_main.csv

Note that the size of this dataset starts to lag Excel on a MacBook Pro Intel Core i7 with 16GB RAM

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	comments	description	duration	event	film_date	languages	main_speake	name	num_speake	published_da	ratings	related_talks	speaker_occ	tags	title	url	views
2	4553	Sir Ken Robir	1164	TED2006	1140825600	60	Ken Robinsor	Ken Robinsor	1	1151367060	[{'id': 7, 'nam	[{'id': 865, 'h	Author/educ	['children', 'c	Do schools ki	https://ww	47227110
3	265	With the san	977	TED2006	1140825600	43	Al Gore	Al Gore: Ave	1	1151367060	[{'id': 7, 'nam	[{'id': 243, 'h	Climate advc	['alternative	Averting the	https://ww	3200520
4	124	New York Tir	1286	TED2006	1140739200	26	David Pogue	David Pogue:	1	1151367060	[{'id': 7, 'nam	[{'id': 1725, '	Technology c	['computers'	Simplicity se	https://ww	1636292
5	200	In an emotio	1116	TED2006	1140912000	35	Majora Carte	Majora Carte	1	1151367060	[{'id': 3, 'nam	[{'id': 1041, '	Activist for e	['MacArthur	Greening the	https://ww	1697550
6	593	You've never	1190	TED2006	1140566400	48	Hans Rosling	Hans Rosling	1	1151440680	[{'id': 9, 'nam	[{'id': 2056, '	Global healtl	['Africa', 'Asi	The best stat	https://ww	12005869
7	672	Tony Robbins	1305	TED2006	1138838400	36	Tony Robbins	Tony Robbins	1	1151440680	[{'id': 7, 'nam	[{'id': 229, 'h	Life coach; e	['business', '	Why we do v	https://ww	20685401
8	919	When two yc	992	TED2006	1140739200	31	Julia Sweene	Julia Sweene	1	1152490260	[{'id': 3, 'nam	[{'id': 22, 'he	Actor, comec	['Christianity	Letting go of	https://ww	3769987
9	46	Architect Jos	1198	TED2006	1140652800	19	Joshua Princ	Joshua Princ	1	1152490260	[{'id': 9, 'nam	[{'id': 750, 'h	Architect	['architectur	Behind the d	https://ww	967741
10	852	Philosopher I	1485	TED2006	1138838400	32	Dan Dennett	Dan Dennett	1	1153181460	[{'id': 3, 'nam	[{'id': 71, 'he	Philosopher,	['God', 'TED	Let's teach r	https://ww	2567958
11	900	Pastor Rick V	1262	TED2006	1140825600	31	Rick Warren	Rick Warren:	1	1153181460	[{'id': 21, 'na	[{'id': 94, 'he	Pastor, authc	['Christianity	A life of purp	https://ww	3095993
12	79	Accepting his	1414	TED2006	1140912000	27	Cameron Sin	Cameron Sin	1	1153786260	[{'id': 3, 'nam	[{'id': 1749, '	Co-founder, /	['activism', '	My wish: A c	https://ww	1211416
13	55	Jehane Nouja	1538	TED2006	1140912000	20	Jehane Nouja	Jehane Nouja	1	1153786260	[{'id': 1, 'nam	[{'id': 2228, '	Filmmaker	['TED Prize',	My wish: A g	https://ww	387877
14	71	Accepting th	1550	TED2006	1140652800	24	Larry Brillian	Larry Brillian	1	1153786260	[{'id': 8, 'nam	[{'id': 1153, '	Epidemiolog	['TED Prize',	My wish: Hel	https://ww	693341
15	242	Jeff Han shov	527	TED2006	1139184000	27	Jeff Han	Jeff Han: The	1	1154391060	[{'id': 9, 'nam	[{'id': 685, 'h	Human-comp	['demo', 'des	The radical p	https://ww	4531020
16	99	Nicholas Neg	1057	TED2006	1140652800	25	Nicholas Neg	Nicholas Neg	1	1154391060	[{'id': 3, 'nam	[{'id': 2043, '	Tech visionar	['children', 'c	One Laptop p	https://ww	358304
17	325	Violinist Sire	1481	TED2006	1140652800	31	Sirena Huang	Sirena Huang	1	1154995860	[{'id': 1, 'nam	[{'id': 2273, '	Violinist	['entertainm	An 11-year-o	https://ww	2702470
18	305	Pianist and c	1445	TED2004	1077753600	32	Jennifer Lin	Jennifer Lin:	1	1154995860	[{'id': 1, 'nam	[{'id': 2273, '	Pianist, com	['creativity',	Improvising	https://ww	1628912
19	88	Fumes from	906	TED2006	1140739200	27	Amy Smith	Amy Smith: S	1	1155600660	[{'id': 9, 'nam	[{'id': 1561, '	inventor, eng	['MacArthur	Simple desig	https://ww	1415724
20	163	Designer Ros	1170	TED2005	1109289600	22	Ross Lovegrc	Ross Lovegrc	1	1155600660	[{'id': 1, 'nam	[{'id': 2251, '	Industrial de	['DNA', 'biolo	Organic desi	https://ww	1074081
21	84	Jimmy Wale	1201	TEDGlobal 2(	1121299200	32	Jimmy Wale	Jimmy Wale	1	1156119060	[{'id': 1, 'nam	[{'id': 640, 'h	Founder of V	['business', '	The birth of \	https://ww	1106561
22	108	In 2006, oper	1114	TED2006	1140652800	27	Richard Bara	Richard Bara	1	1156119060	[{'id': 9, 'nam	[{'id': 1913, '	Education vis	['business', '	The birth of t	https://ww	966439
23	185	Performer ar	1136	TED2004	1077580800	26	Ze Frank	Ze Frank: Ne	1	1156464660	[{'id': 7, 'nam	[{'id': 148, 'h	Humorist, wi	['collaboratic	Nerdcore cor	https://ww	6141440
24	50	The founding	1006	TED2006	1140652800	20	Mena Trott	Mena Trott:	1	1156464660	[{'id': 8, 'nam	[{'id': 144, 'h	Blogger; cofc	['business', '	Meet the fou	https://ww	518624
25	556	Anthropologi	1407	TED2006	1140739200	33	Helen Fisher	Helen Fisher	1	1157501460	[{'id': 7, 'nam	[{'id': 307, 'h	Anthropologi	['cognitive sc	Why we love	https://ww	9260764
26	117	Eve Ensler, c	1225	TED2004	1075852800	23	Eve Ensler	Eve Ensler: H	1	1157501460	[{'id': 3, 'nam	[{'id': 217, 'h	Playwright, a	['culture', 'er	Happiness in	https://ww	1131864
27	184	Legendary sc	1140	TEDGlobal 2(	1121299200	29	David Deutsc	David Deutsc	1	1158019860	[{'id': 9, 'nam	[{'id': 2237, '	Quantum ph	['climate cha	Chemical scu	https://ww	1096862
28	507	Biologist Ric	1316	TEDGlobal 2(	1120694400	36	Richard Daw	Richard Daw	1	1158019860	[{'id': 1, 'nam	[{'id': 1276, '	Evolutionary	['astronomy'	Why the univ	https://ww	2885999
29	95	"Freakonomi	1275	TED2004	1077840000	25	Steven Levitt	Steven Levitt	1	1158624660	[{'id': 7, 'nam	[{'id': 20, 'he	Economist	['business', '	The freakonc	https://ww	2863214

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

A view of the Datasets from Excel

On the right is a screenshot of transcripts.csv

Note that the size of this dataset (~29MB) starts to lag Excel on a MacBook Pro Intel Core i7 with 16GB RAM

	A	B
1	transcript	url
2	Good morning. How are you?(Laughter)It's been great, hasn't it? I'	https://www.ted.com/talks/ken_robinson_says_sc
3	Thank you so much, Chris. And it's truly a great honor to have the o	https://www.ted.com/talks/al_gore_on_averting_c
4	(Music: "The Sound of Silence," Simon & Garfunkel)Hello voice ma	https://www.ted.com/talks/david_pogue_says_si
5	If you're here today ,Äî and I'm very happy that you are ,Äî you've a	https://www.ted.com/talks/majora_carter_s_tale_
6	About 10 years ago, I took on the task to teach global development	https://www.ted.com/talks/hans_rosling_shows_t
7	Thank you. I have to tell you I'm both challenged and excited. My ex	https://www.ted.com/talks/tony_robbins_asks_wh
8	On September 10, the morning of my seventh birthday, I came dow	https://www.ted.com/talks/julia_sweeney_on_lett
9	I'm going to present three projects in rapid fire. I don't have much	https://www.ted.com/talks/joshua_prince_ramus_
10	It's wonderful to be back. I love this wonderful gathering. And you	https://www.ted.com/talks/dan_dennett_s_respon
11	I'm often asked, "What surprised you about the book?" And I say, "	https://www.ted.com/talks/rick_warren_on_a_life
12	I'm going to take you on a journey very quickly. To explain the wish	https://www.ted.com/talks/cameron_sinclair_on_
13	I can't help but this wish: to think about when you're a little kid, an	https://www.ted.com/talks/jehane_noujaim_inspir
14	I'm the luckiest guy in the world. I got to see the last case of killer	https://www.ted.com/talks/larry_brilliant_wants_t
15	I'm really excited to be here today. I'll show you some stuff that's j	https://www.ted.com/talks/jeff_han_demos_his_b
16	I've been at MIT for 44 years. I went to TED I. There's only one othe	https://www.ted.com/talks/nicholas_negroponte_
17	(Music)(Music ends)(Applause)(Applause ends)Hi, everyone. I'm Si	https://www.ted.com/talks/sirena_huang_dazzles
18	(Music)(Music ends)(Applause)Thank you!(Applause continues)Tha	https://www.ted.com/talks/jennifer_lin_improvs_
19	In terms of invention, I'd like to tell you the tale of one of my favor	https://www.ted.com/talks/amy_smith_shares_si
20	My name is Lovegrove. I only know nine Lovegroves, two of which a	https://www.ted.com/talks/ross_lovegrove_shares
21	Charles Van Doren, who was later a senior editor of Britannica, sai	https://www.ted.com/talks/jimmy_wales_on_the_
22	I'm Rich Baraniuk and what I'd like to talk a little bit about today a	https://www.ted.com/talks/richard_baraniuk_on_o
23	You know, when Chris first approached me to speak at TED, I said r	https://www.ted.com/talks/ze_frank_s_nerdcore_
24	Over the past couple of days, as I've been preparing for my speech,	https://www.ted.com/talks/mena_trott_tours_her
25	I'd like to talk today about the two biggest social trends in the com	https://www.ted.com/talks/helen_fisher_tells_us_
26	I bet you're worried.(Laughter)I was worried. That's why I began th	https://www.ted.com/talks/eve_ensler_on_happin
27	We've been told to go out on a limb and say something surprising.	https://www.ted.com/talks/david_deutsch_on_our
28	My title: "Queerer than we can suppose: the strangeness of scienc	https://www.ted.com/talks/richard_dawkins_on_o
29	You'll be happy to know that I'll be talking not about my own trage	https://www.ted.com/talks/steven_levitt_analyzes

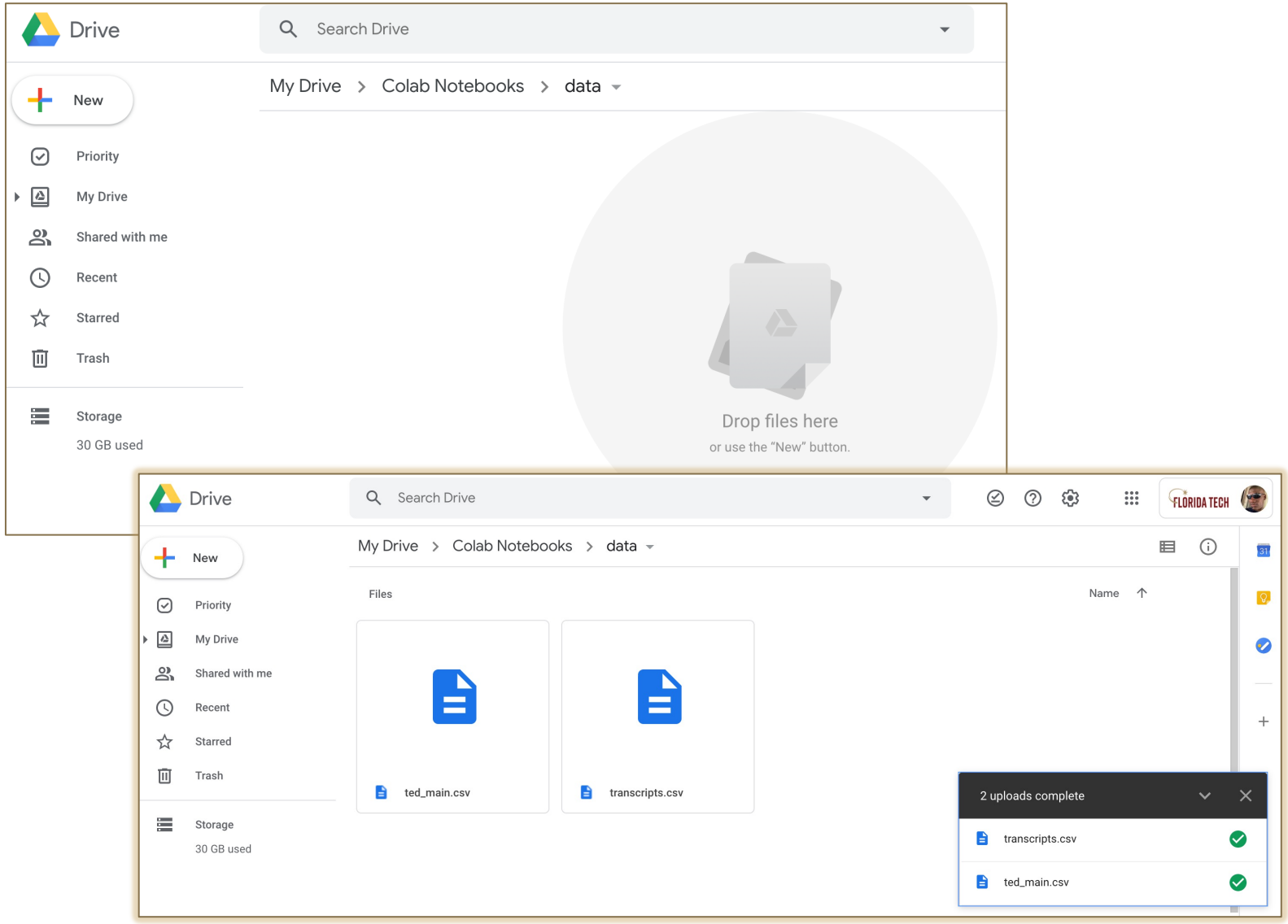
FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Make datasets accessible to Jupyter Notebook

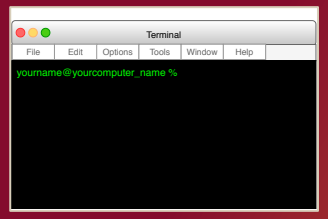
If you choose to use Colaboratory (colab.research.google.com),

Create a folder called data inside your Colab Notebooks folder. The Colab Notebooks folder is created by Colaboratory upon first time use.

Upload your two datasets here



Make datasets accessible to Jupyter Notebook



If you choose to use your local Jupyter environment, create a folder in your class folder named data. You may use the GUI or the CLI to create your folder.

mkdir data

```
(base) fitzroi@Fitzroys-MacBook-Pro-15 Documents % cd CSE_5150
(base) fitzroi@Fitzroys-MacBook-Pro-15 CSE_5150 % ls
classwork       homework
(base) fitzroi@Fitzroys-MacBook-Pro-15 CSE_5150 % mkdir data
(base) fitzroi@Fitzroys-MacBook-Pro-15 CSE_5150 %
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Make datasets accessible to Jupyter Notebook from Google Drive

You may choose to store your datasets on Google Drive and use them in your local Jupyter environment.

Make sure anyone with the link can access the file

```
# This code downloads/reads CSV data from Google Drive for use in
Pandas
# This code also works in Colaboratory

from io import StringIO
import requests

google_csv_url = "https://drive.google.com/file/d/.../view?usp=sharing"

file_name_parts = google_csv_url.split('/')
file_id = file_name_parts[len(file_name_parts) - 2]
download_url="https://drive.google.com/uc?export=download&id=" + file_id

csv_content = requests.get(download_url).text

csv_data = StringIO(csv_content)
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Make datasets accessible to Colaboratory from Google Drive

You may choose to store your datasets on Google Drive and use them in Colab

You may use private mode or public mode.

You may use private mode or public mode.

For private mode, follow step 3 in the following tutorial: https://towardsdatascience.com/3-ways-to-load-csv-files-into-colab-7c14fcbdcb92

For public mode, you may use the same steps on the previous slide.

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Import Pandas

First, Launch Jupyter notebook.

Name your notebook “preprocessing_ted_talks”

Run the following code to import pandas.

Note that you may have to install pandas preferably from your terminal.

conda install pandas

```
# The code uses Pandas to preprocess csv files

import pandas as pd
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Load the Datasets using Pandas

Run the following lines of code to load the datasets.

Pandas loads the CSV file as as DataFrame.

A DataFrame can be thought of either as a generalization of a NumPy array, or as a specialization of a Python dictionary. We will cover more on these concepts later.

```
ted_main = pd.read_csv("data/ted_main.csv")
transcripts = pd.read_csv("data/transcripts.csv")

# If you are reading data from Google drive, change the
file path to the csv_data variable previously initialized.
You may also try to skip the step that reads the text by
doing pd.read_csv(download_url)
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

What's the size of the data in terms of rows and columns?

Run the following lines of code to get the size of the data

```
In [7]: 1 ted_main.shape
Out[7]: (2550, 17)

In [8]: 1 transcripts.shape
Out[8]: (2467, 2)
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

What does the data look like?

Run ted_main.head(2) to see the first 2 rows

As a note of caution, you may not be able to look at an entire dataset.

The recommended approach is to use windowing to see snapshots of the data

Check the URL of the first row:

ted_main.url[0]

```
In [11]: 1 ted_main.head(2)
```

Out[11]:

	comments	description	duration	event	film_date	languages	main_speaker	name	num_speaker	published_date	ratings	related_talks	speaker_occupation
0	4553	Sir Ken Robinson makes an entertaining and pro...	1164	TED2006	1140825600	60	Ken Robinson	Ken Robinson: Do schools kill creativity?	1	1151367060	[{'id': 7, 'name': 'Funny', 'count': 19645}, {...	[{'id': 865, 'hero': 'https://pe.tedcdn.com/im...	Author/educator
1	265	With the same humor and humanity he exuded in ...	977	TED2006	1140825600	43	Al Gore	Al Gore: Averting the climate crisis	1	1151367060	[{'id': 7, 'name': 'Funny', 'count': 544}, {'i...	[{'id': 243, 'hero': 'https://pe.tedcdn.com/im...	Climate advocate

```
In [12]: 1 ted_main.url[0]
Out[12]: 'https://www.ted.com/talks/ken_robinson_says_schools_kill_creativity\n'
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

What does the data look like?

Let's look at a specific row

```
#Get row no. 23

ted_main.iloc[23]
```

```
In [13]:  1 ted_main.iloc[23]
          2

Out[13]: comments                                                      556
         description          Anthropologist Helen Fisher takes on a tricky ...
         duration                                                     1407
         event                                                     TED2006
         film_date                                              1140739200
         languages                                                      33
         main_speaker                                          Helen Fisher
         name                                Helen Fisher: Why we love, why we cheat
         num_speaker                                                     1
         published_date                                         1157501460
         ratings              [{'id': 7, 'name': 'Funny', 'count': 780}, {'i...
         related_talks        [{'id': 307, 'hero': 'https://pe.tedcdn.com/im...
         speaker_occupation                    Anthropologist, expert on love
         tags                 ['cognitive science', 'culture', 'evolution', ...
         title                                          Why we love, why we cheat
         url                  https://www.ted.com/talks/helen_fisher_tells_u...
         views                                                       9260764
         Name: 23, dtype: object
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

What does the data look like?

Let's look at the URL field for the first 4 rows

```
for i, row in ted_main.iterrows():
    if i > 4:
        break
    print(row["url"])
```

```
In [22]:
for i, row in ted_main.iterrows():
    if i > 4:
        break
    print(row["url"])

https://www.ted.com/talks/ken_robinson_says_schools_kill_creativity

https://www.ted.com/talks/al_gore_on_averting_climate_crisis

https://www.ted.com/talks/david_pogue_says_simplicity_sells

https://www.ted.com/talks/majora_carter_s_tale_of_urban_renewal

https://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

What does the data look like?

Let's look at the row with index number 666

ted_main.loc[666]

What's the difference between loc and iloc?

loc gets rows (or columns) with particular labels from the index.

iloc gets rows (or columns) at particular positions in the index (so it only takes integers)

```
In [25]: 1 ted_main.loc[666]

Out[25]: comments                                                        77
         description          Some 80 to 90 percent of undersea creatures ma...
         duration                                                      1039
         event                                            Mission Blue Voyage
         film_date                                              1270771200
         languages                                                      22
         main_speaker                                          Edith Widder
         name                 Edith Widder: Glowing life in an underwater world
         num_speaker                                                     1
         published_date                                         1271665800
         ratings              [{'id': 23, 'name': 'Jaw-dropping', 'count': 1...
         related_talks        [{'id': 467, 'hero': 'https://pe.tedcdn.com/im...
         speaker_occupation                                   Marine biologist
         tags                 ['animals', 'design', 'exploration', 'fish', '...
         title                              Glowing life in an underwater world
         url                  https://www.ted.com/talks/edith_widder_glowing...
         views                                                         579754
         Name: 666, dtype: object
```

What are the data types of the various fields?

Run ted_main.dtypes to get the data types

```
In [26]: 1 ted_main.dtypes

Out[26]: comments             int64
         description         object
         duration             int64
         event               object
         film_date            int64
         languages            int64
         main_speaker        object
         name                object
         num_speaker          int64
         published_date       int64
         ratings             object
         related_talks       object
         speaker_occupation  object
         tags                object
         title               object
         url                 object
         views                int64
         dtype: object
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Compute Descriptive statistics

Run ted_main.describe() to compute descriptive statistics of the numerical fields

```
In [27]: 1 ted_main.describe()
```

Out[27]:

	comments	duration	film_date	languages	num_speaker	published_date	views
count	2550.000000	2550.000000	2.550000e+03	2550.000000	2550.000000	2.550000e+03	2.550000e+03
mean	191.562353	826.510196	1.321928e+09	27.326275	1.028235	1.343525e+09	1.698297e+06
std	282.315223	374.009138	1.197391e+08	9.563452	0.207705	9.464009e+07	2.498479e+06
min	2.000000	135.000000	7.464960e+07	0.000000	1.000000	1.151367e+09	5.044300e+04
25%	63.000000	577.000000	1.257466e+09	23.000000	1.000000	1.268463e+09	7.557928e+05
50%	118.000000	848.000000	1.333238e+09	28.000000	1.000000	1.340935e+09	1.124524e+06
75%	221.750000	1046.750000	1.412964e+09	33.000000	1.000000	1.423432e+09	1.700760e+06
max	6404.000000	5256.000000	1.503792e+09	72.000000	5.000000	1.506092e+09	4.722711e+07

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Extracting columns

Run the following code to select the "main_speaker" and "speaker_occupation" for the first five rows.

This command is similar to running a projection in a database query: SELECT <..> FROM ..

head() by default returns the first 5 rows. However, we can pass an argument to the function to specify the number of rows we want.

```
snapshot = ["main_speaker", "speaker_occupation"]
ted_main[snapshot].head()
```

In [28]:

```
1 snapshot = ['main_speaker', 'speaker_occupation']
2 ted_main[snapshot].head()
```

Out[28]:

	main_speaker	speaker_occupation
0	Ken Robinson	Author/educator
1	Al Gore	Climate advocate
2	David Pogue	Technology columnist
3	Majora Carter	Activist for environmental justice
4	Hans Rosling	Global health expert; data visionary

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Selecting data that satisfy certain constraints

Run the following code to
get a subset of rows that satisfy a constraint

This is similar to a selection in a database query:

SELECT * WHERE..

```
constraint = ted_main["views"]>31415926
ted_main[constraint]
```

In [29]:

```
mapp = ted_main['views']>31415926
ted_main[mapp]
```

Out[29]:

name	num_speaker	published_date	ratings	related_talks	speaker_occupation	tags	title	url	views
Ken binson: Do schools kill ativity?	1	1151367060	[{'id': 7, 'name': 'Funny', 'count': 19645}, {...	[{'id': 865, 'hero': 'https://pe.tedcdn.com/im...	Author/educator	['children', 'creativity', 'culture', 'dance',...	Do schools kill creativity?	https://www.ted.com/talks/ken_robinson_says_sc...	47227110
Simon Sinek: How great leaders inspire action	1	1272965460	[{'id': 21, 'name': 'Unconvincing', 'count': 9...	[{'id': 814, 'hero': 'https://pe.tedcdn.com/im...	Leadership expert	['TEDx', 'business', 'entrepreneur', 'leadersh...	How great leaders inspire action	https://www.ted.com/talks/simon_sinek_how_grea...	34309432
Amy Cuddy: ur body nguage may shape ho yo...	1	1349103608	[{'id': 23, 'name': 'Jaw-dropping', 'count': 3...	[{'id': 605, 'hero': 'https://pe.tedcdn.com/im...	Social psychologist	['body language', 'brain', 'business', 'psycho...	Your body language may shape who you are	https://www.ted.com/talks/amy_cuddy_your_body_...	43155405

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Counting values

Run the following code to count the number of rows for every value of the field named "num_speaker"

```
ted_main["num_speaker"].value_counts()
```

```
In [30]: 1 ted_main['num_speaker'].value_counts()

Out[30]: 1    2492
         2      49
         3       5
         4       3
         5       1
         Name: num_speaker, dtype: int64
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Counting values

After obtaining the value_counts, we may access the values and the indices as follows:

counts.values

counts.index

```
In [16]:  1  counts = ted_main["num_speaker"].value_counts()

In [17]:  1  counts.values

Out[17]: array([2492,   49,    5,    3,    1])

In [21]:  1  counts.index

Out[21]: Int64Index([1, 2, 3, 4, 5], dtype='int64')
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Counting values

After obtaining the value_counts, we may access the values and the indices as follows:

counts.values

counts.index

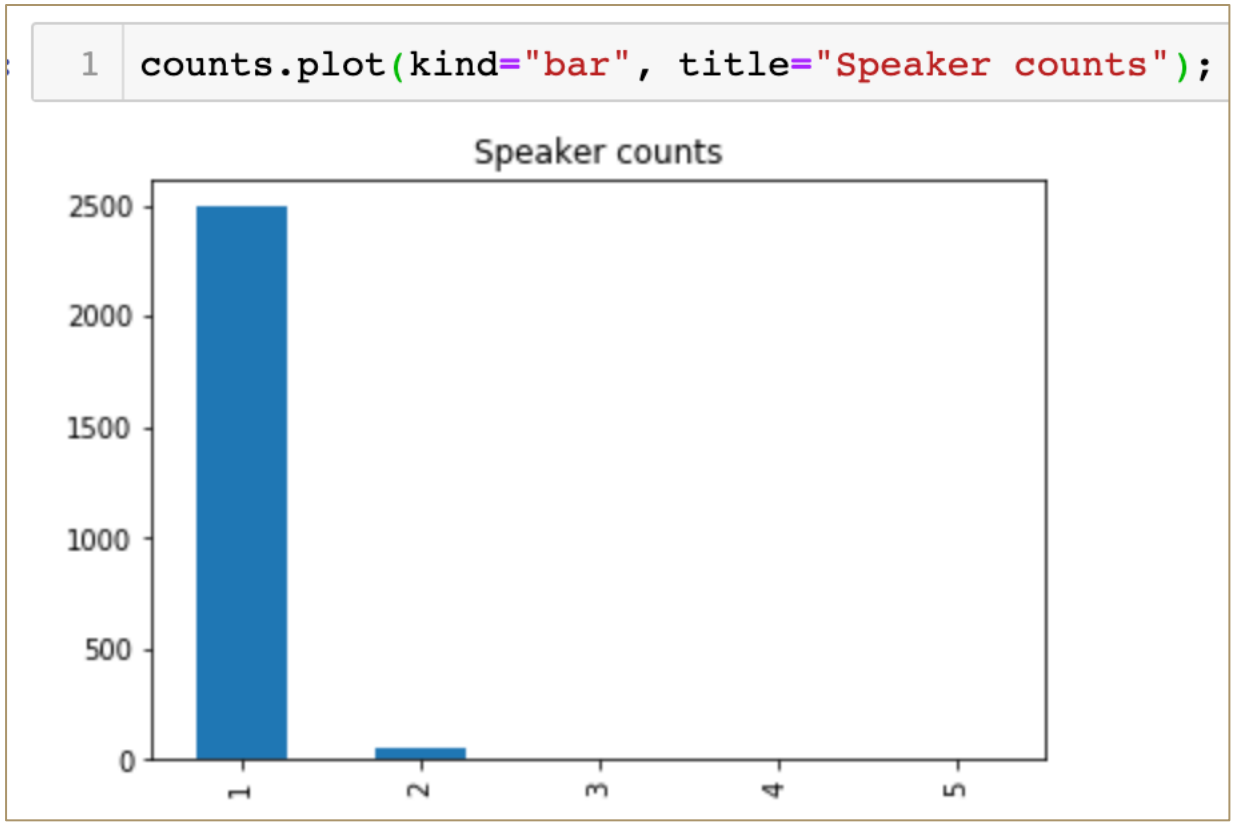
```
In [16]:  1  counts = ted_main["num_speaker"].value_counts()

In [17]:  1  counts.values

Out[17]: array([2492,   49,    5,    3,    1])

In [21]:  1  counts.index

Out[21]: Int64Index([1, 2, 3, 4, 5], dtype='int64')
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Plotting value_counts

We may also create a plot of the value_counts.

```
1  counts.plot(kind="bar", title="Speaker counts");
```



Theory on Data Cleaning

The following are guidelines for cleaning data:

Number / Character Representations

- Measurements should generally be decimal numbers
- Counts should be integers.
- Fractional quantities should be decimal, not (q,r) like (pounds,oz) or (feet,inches).

Character representations

- UTF-8 is a multibyte encoding for all Unicode characters

Name Unification

- Use simple transformations to unify names, like lower case, removing middle names, etc.
- Consider phonetic hashing methods like Soundex and Metaphone

Time / Date Unification

- Use Coordinated Universal Time (UTC), a modern standard subsuming GMT

Slide courtesy of Steven Skiena, Stony Brook University

Theory on Handling Missing Values

The following are techniques used to handle missing values:

Dealing with Missing Data

- **Three methods are considered:**
- Replace (impute)
- Examples:
  - Impute missing credit bureau scores with the average or median of the known values.
  - For marital status, the mode can then be used. Regression-based imputation to model a target variable (e.g., credit bureau score)
- **Delete:** assumes that information is missing at random and has no meaningful interpretation and/or relationship to the target
- **Keep:** Missing values can be meaningful (e.g., a customer did not disclose his or her income because he or she is currently unemployed)

Analytics in a Big Data World by Bart Baesens, 2013

Theory on Handling Missing Values

The following are techniques used to handle missing values:

Dealing with Missing Data

- Often it is better to estimate or impute missing values instead of leaving them blank.
- A good guess for your death year is birth+80

Imputation methods

- *Mean value imputation* - leaves mean same.
- *Random value imputation* - repeatedly selecting random values permits statistical evaluation of the impact of imputation.
- *Imputation by interpolation* - using linear regression to predict missing values works well if few fields are missing per record

Slide courtesy of Steven Skiena, Stony Brook University

Theory on Outlier Detection

The following are guidelines for handling outliers:

Outlier Detection

- Look critically at the maximum and minimum values for all variables.
- Normally distributed data should not have large outliers, *k sigma* from the mean.
- Visually, it is easy to detect outliers, but only in low dimensional spaces.
- It can be thought of as an unsupervised learning problem, like clustering.
- Points which are far from their cluster center are good candidates for outliers
- Deleting outliers prior to fitting can yield better models, e.g. if these points correspond to measurement error.
- Deleting outliers prior to fitting can yield worse models, e.g. if you are simply deleting points which are not explained by your simple model.

Slide courtesy of Steven Skiena, Stony Brook University

Finding null values

Run the following code to find all fields with null values:

```
ted_main.isnull().sum()
```

In Pandas, for a field to be null, it must contain the Python noneType (None) or NaN (Not a Number)

```
In [31]: 1 ted_main.isnull().sum()

Out[31]: comments            0
         description         0
         duration            0
         event               0
         film_date           0
         languages           0
         main_speaker        0
         name                0
         num_speaker         0
         published_date      0
         ratings             0
         related_talks       0
         speaker_occupation  6
         tags                0
         title               0
         url                 0
         views               0
         dtype: int64
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Recall the shape of the data

Run the following code to get the shape of the data

ted_main.shape

```
In [33]:  1  ted_main.shape

Out[33]: (2550, 17)
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Replacing missing values

Run the following code to replace missing values and to find the number of values replaced.

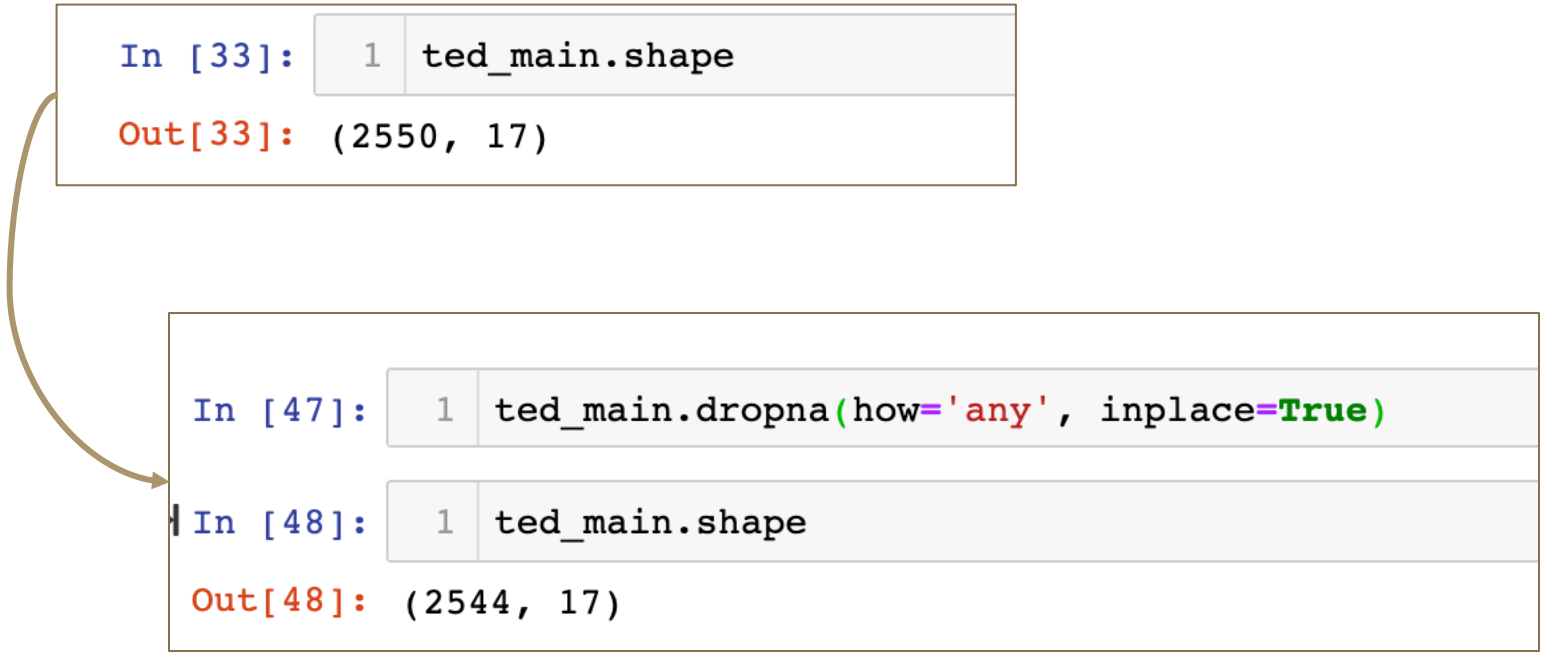
```
temp = ted_main.fillna("Not specified")
temp2 = temp[temp["speaker_occupation"] ==
"Not specified"]

len(temp2)
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Dropping rows with missing values

Run the following code to drop rows with missing values; then check the shape of the data to see how the drop affected the data.

```
ted_main.dropna(how='any', inplace=True)
ted_main.shape
```



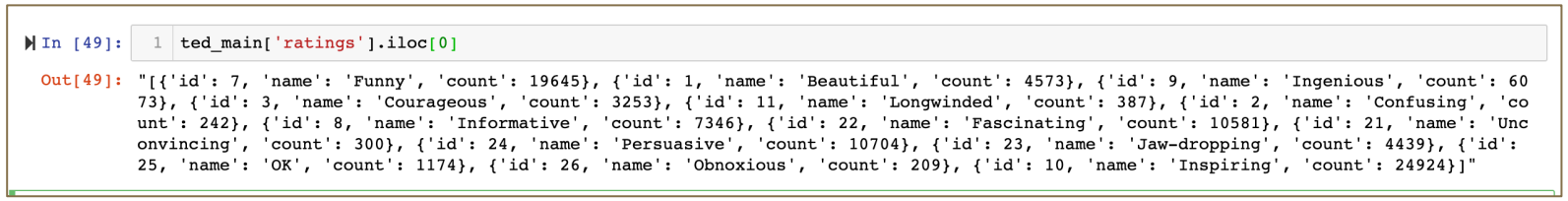
FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Changing Data Types

Let's take a look at the ratings field, the film_date and the published_date.

Notice that ratings is a string and the dates are integers (number of days since a time period such as 1-Jan-1900)

```
ted_main["ratings"].iloc[0]
ted_main["film_date"].iloc[0]
ted_main["published_date"].iloc[0]
```



```
In [49]: 1 ted_main['ratings'].iloc[0]

Out[49]: "[{'id': 7, 'name': 'Funny', 'count': 19645}, {'id': 1, 'name': 'Beautiful', 'count': 4573}, {'id': 9, 'name': 'Ingenious', 'count': 6073}, {'id': 3, 'name': 'Courageous', 'count': 3253}, {'id': 11, 'name': 'Longwinded', 'count': 387}, {'id': 2, 'name': 'Confusing', 'count': 242}, {'id': 8, 'name': 'Informative', 'count': 7346}, {'id': 22, 'name': 'Fascinating', 'count': 10581}, {'id': 21, 'name': 'Unconvincing', 'count': 300}, {'id': 24, 'name': 'Persuasive', 'count': 10704}, {'id': 23, 'name': 'Jaw-dropping', 'count': 4439}, {'id': 25, 'name': 'OK', 'count': 1174}, {'id': 26, 'name': 'Obnoxious', 'count': 209}, {'id': 10, 'name': 'Inspiring', 'count': 24924}]"
```

```
In [50]: 1 ted_main['film_date'].iloc[0]

Out[50]: 1140825600

In [51]: 1 ted_main['published_date'].iloc[0]

Out[51]: 1151367060
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Changing Data Types

Let's convert ratings, related talks, tags, and dates to proper datatypes.

The literal_eval function can be used to safely evaluate an expression node or a Unicode or Latin-1 encoded string containing a Python expression. For example, a dictionary saved as a string can be converted to a dictionary. We cover dictionaries in detail in next week's lecture.

```
from ast import literal_eval

ted_main['ratings'] = ted_main['ratings'].apply(literal_eval)
ted_main['related_talks'] = ted_main['related_talks'].apply(literal_eval)
ted_main['tags'] = ted_main['tags'].apply(literal_eval)

ted_main['film_date'] = pd.to_datetime(ted_main['film_date'], unit='s')
ted_main['published_date'] = pd.to_datetime(ted_main['published_date'],
unit='s')


# Take a look at the ratings and compare with the previous display
ted_main['ratings'][0]

# Run the following command to see what to_datetime does
pd.to_datetime?
```

The apply() method lets you apply an arbitrary function to the group results. The function should take a DataFrame, and return either a Pandas object (e.g., DataFrame, Series) or a scalar. We will cover more on DataFrames and Series later.

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Observe the changes

Notice the change with the dates and ratings, for example.

```
ted_main.head(2)
```

```
In [56]: 1 ted_main.head(2)
```

Out[56]:

	comments	description	duration	event	film_date	languages	main_speaker	name	num_speaker	published_date	ratings	related_talks	speaker_occupation	tags
0	4553	Sir Ken Robinson makes an entertaining and pro...	1164	TED2006	2006-02-25	60	Ken Robinson	Ken Robinson: Do schools kill creativity?	1	2006-06-27 00:11:00	[{'count': 19645, 'id': 7, 'name': 'Funny'}, {...	[{'duration': 1008, 'title': 'Bring on the lea...	Author/educator	[children, creativity, culture, dance, educati...
1	265	With the same humor and humanity he exuded in ...	977	TED2006	2006-02-25	43	Al Gore	Al Gore: Averting the climate crisis	1	2006-06-27 00:11:00	[{'count': 544, 'id': 7, 'name': 'Funny'}, {'c...	[{'duration': 1674, 'title': 'New thinking on ...	Climate advocate	[alternative energy, cars, climate change, cul...

Keep working

Update languages to 1, for rows that have languages set to 0

Print the url of the first row and note the newlines at the end.

Update the url column values by removing the newline characters for each row.

```
ted_main.loc[ted_main['languages']==0, 'languages'] = 1
ted_main['url'].iloc[0]
```

```
In [58]: 1 ted_main['url'].iloc[0]
Out[58]: 'https://www.ted.com/talks/ken_robinson_says_schools_kill_creativity\n'
```

```
ted_main['url'] = ted_main['url'].apply(lambda x: x.replace('\n', ''))
ted_main['url'].iloc[0]
```

```
In [60]: 1 ted_main['url'].iloc[0]
Out[60]: 'https://www.ted.com/talks/ken_robinson_says_schools_kill_creativity'
```

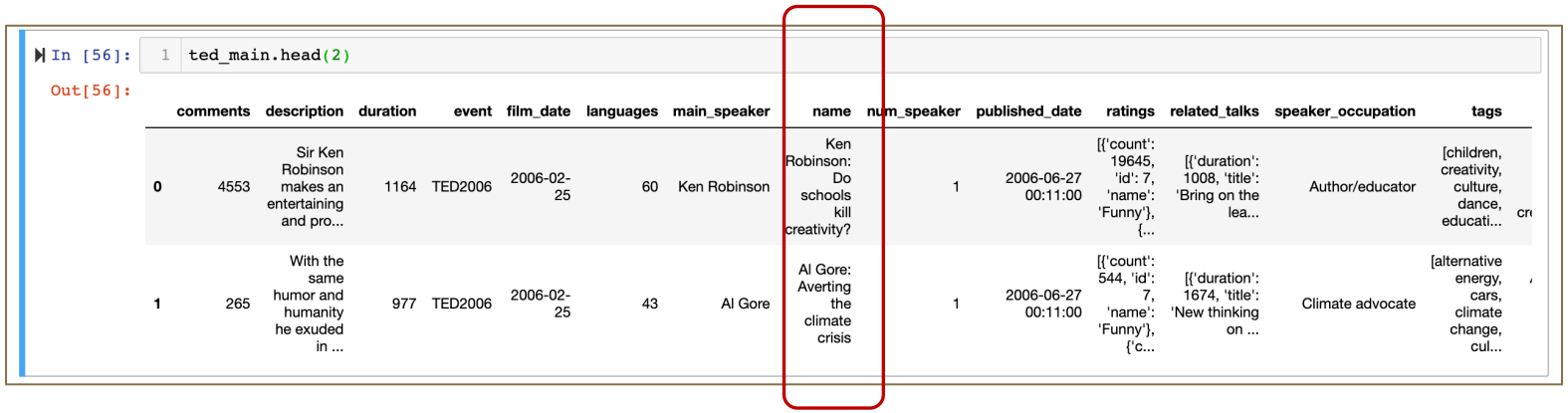
FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Using your own functions to clean data

Run ted_main.head(2) and take a look at the name field. Notice that we have the speaker's name as part of the name of the talk. Let's write a function to remove it.

Let's now apply the function to do its job.

Run ted_main.head(2) again and observe the change



```
In [56]: 1 ted_main.head(2)
```

Out[56]:

	comments	description	duration	event	film_date	languages	main_speaker	name	num_speaker	published_date	ratings	related_talks	speaker_occupation	tags
0	4553	Sir Ken Robinson makes an entertaining and pro...	1164	TED2006	2006-02-25	60	Ken Robinson	Ken Robinson: Do schools kill creativity?	1	2006-06-27 00:11:00	[{'count': 19645, 'id': 7, 'name': 'Funny'}, {...	[{'duration': 1008, 'title': 'Bring on the lea...	Author/educator	[children, creativity, culture, dance, educati...
1	265	With the same humor and humanity he exuded in ...	977	TED2006	2006-02-25	43	Al Gore	Al Gore: Averting the climate crisis	1	2006-06-27 00:11:00	[{'count': 544, 'id': 7, 'name': 'Funny'}, {'c...	[{'duration': 1674, 'title': 'New thinking on ...	Climate advocate	[alternative energy, cars, climate change, cul...

```
def remove_speaker_name(row):
  return row['name'].replace("%s:" % row['main_speaker'], '').strip()
```

```
ted_main['name'] = ted_main.apply(remove_speaker_name, axis=1)
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Remove unnecessary strings

Notice the words (Laughter) and (Music..) in the first and third rows of the transcripts dataset. These are clearly not part of the transcript.

We will use regular expressions to remove these.

lambda simply means anonymous. These functions are one-liners that return a value.

```
In [61]: 1 transcripts.head()
```

Out[61]:

	transcript	url
0	Good morning. How are you?(Laughter)It's been ...	https://www.ted.com/talks/ken_robinson_says_sc...
1	Thank you so much, Chris. And it's truly a gre...	https://www.ted.com/talks/al_gore_on_averting_...
2	(Music: "The Sound of Silence," Simon & Garfun...	https://www.ted.com/talks/david_pogue_says_sim...
3	If you're here today — and I'm very happy that...	https://www.ted.com/talks/majora_carter_s_tale...
4	About 10 years ago, I took on the task to teac...	https://www.ted.com/talks/hans_rosling_shows_t...

```
import re

transcripts['transcript'] = transcripts['transcript'].apply(lambda x:
re.sub(r'\((.*?)\)', ' ', x))
```

```
In [73]: 1 transcripts.head()
```

Out[73]:

	transcript	url
0	Good morning. How are you? It's been great, ha...	https://www.ted.com/talks/ken_robinson_says_sc...
1	Thank you so much, Chris. And it's truly a gre...	https://www.ted.com/talks/al_gore_on_averting_...
2	Hello voice mail, my old friend. I've called ...	https://www.ted.com/talks/david_pogue_says_sim...
3	If you're here today — and I'm very happy that...	https://www.ted.com/talks/majora_carter_s_tale...
4	About 10 years ago, I took on the task to teac...	https://www.ted.com/talks/hans_rosling_shows_t...

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

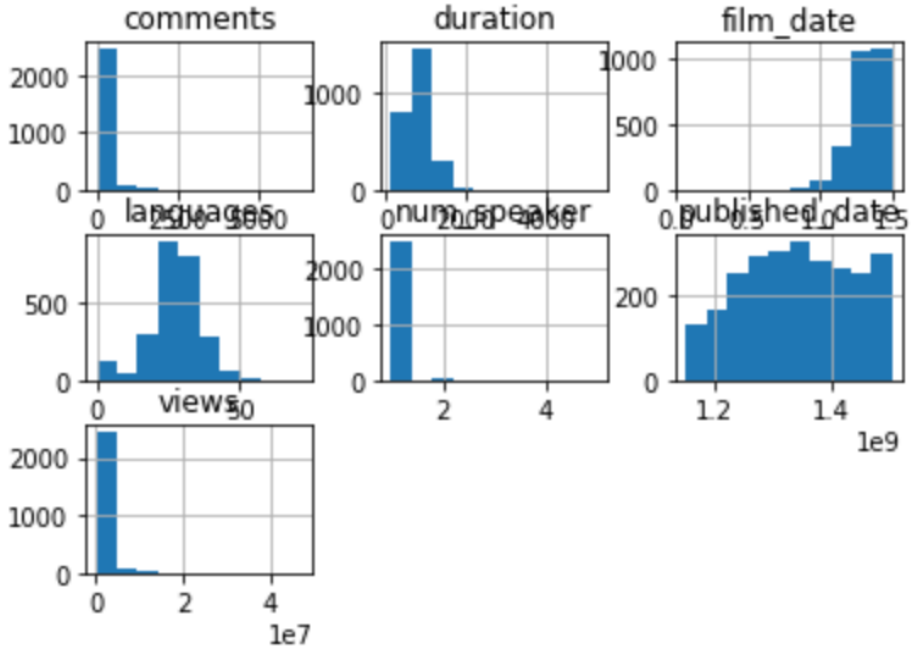
Visualizing the data

To use plots, install matplotlib. You may have to restart your kernel.

Include the following import before your pandas import

```
from matplotlib import pyplot as plt

%matplotlib inline
```

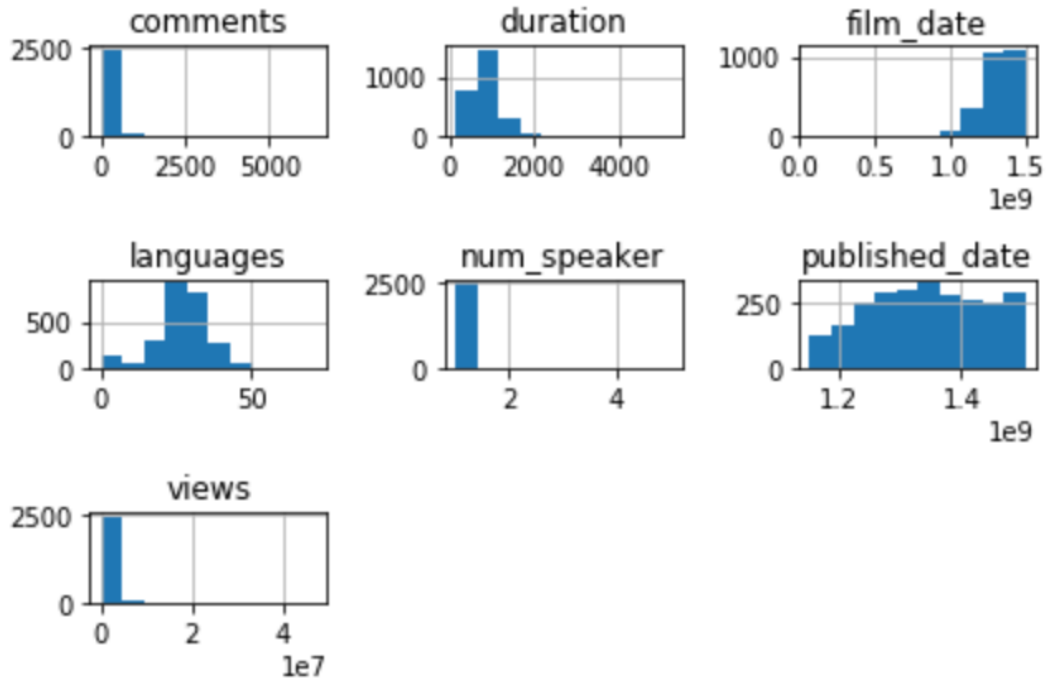
```
In [2]:  1  hists = ted_main.hist()
```



Visualizing the data

tight_layout makes things a little more organized.

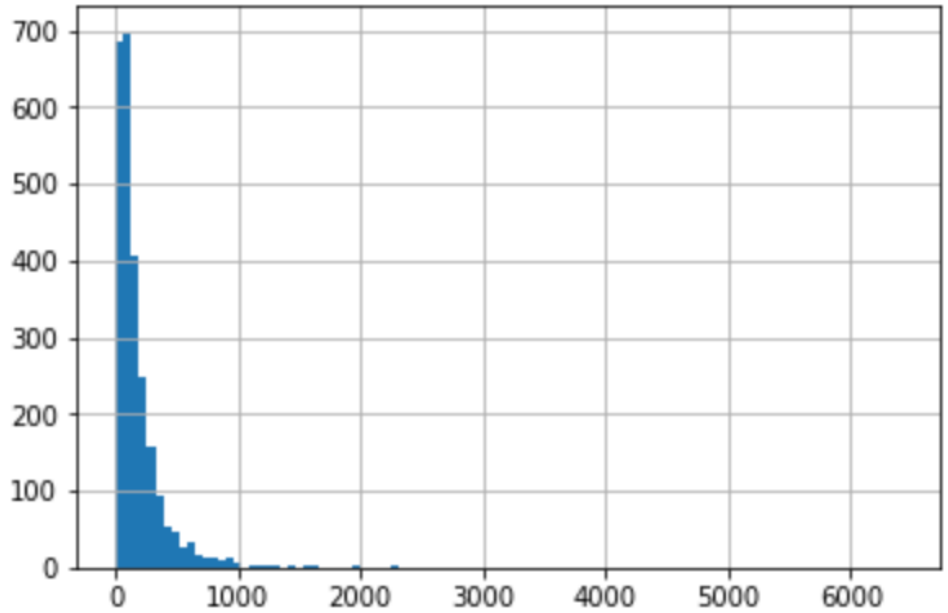
```
In [3]:
1  hists = ted_main.hist()
2  plt.tight_layout()
```



Visualizing the data

Plot comments by specifying the number of bins

```
In [4]:  1  ted_main['comments'].hist(bins=100);
```

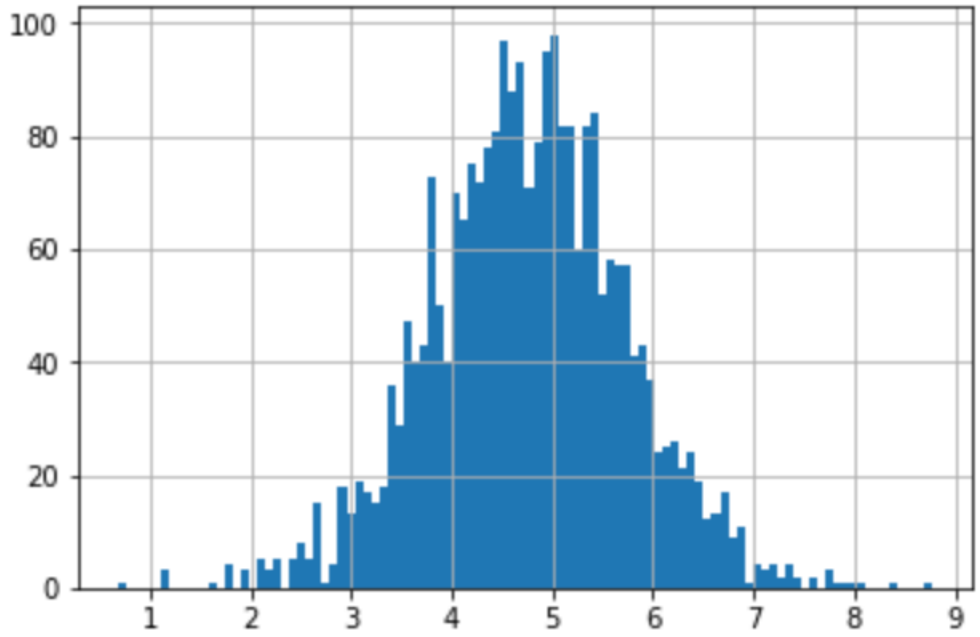


Visualizing the data

Use numpy to get a log plot of the comments. Looks like a normal distribution.

In this plot, the y-axis represents density (not a count) as in the previous plot.

```
In [5]: 1 import numpy as np

In [6]: 1 np.log(ted_main['comments']).hist(bins=100);
```

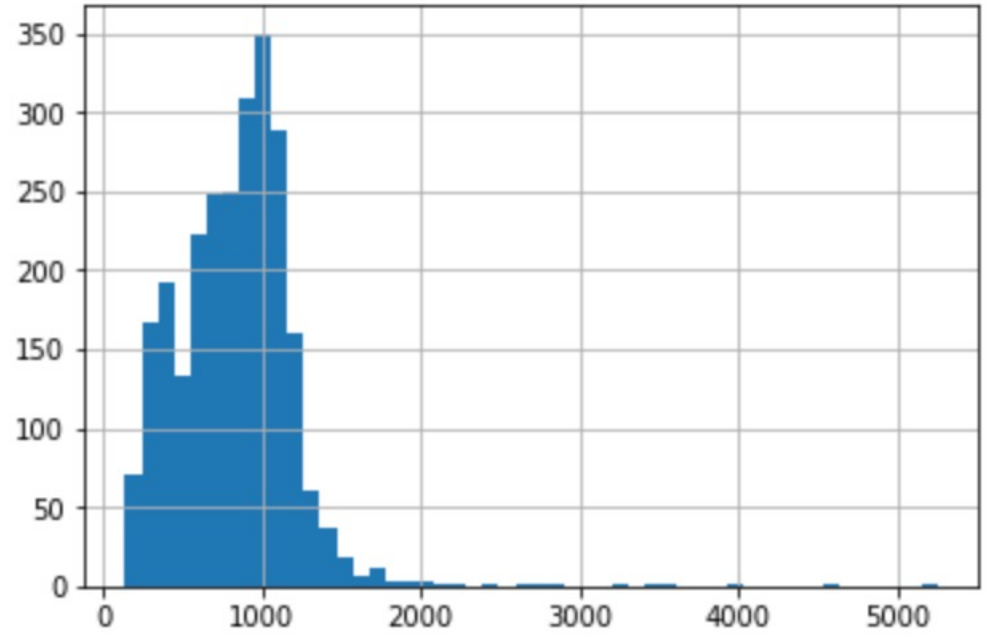


FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Visualizing the data

Look at the distribution of the duration

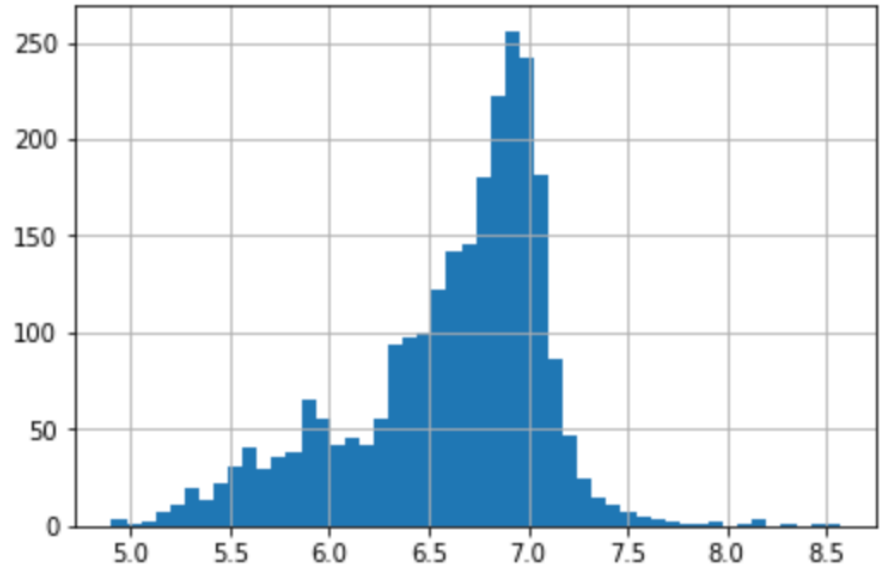
```
In [7]:  1  ted_main['duration'].hist(bins=50);
```



Visualizing the data

Look at the distribution of the duration as a log plot

```
In [8]:  1  np.log(ted_main['duration']).hist(bins=50);
```

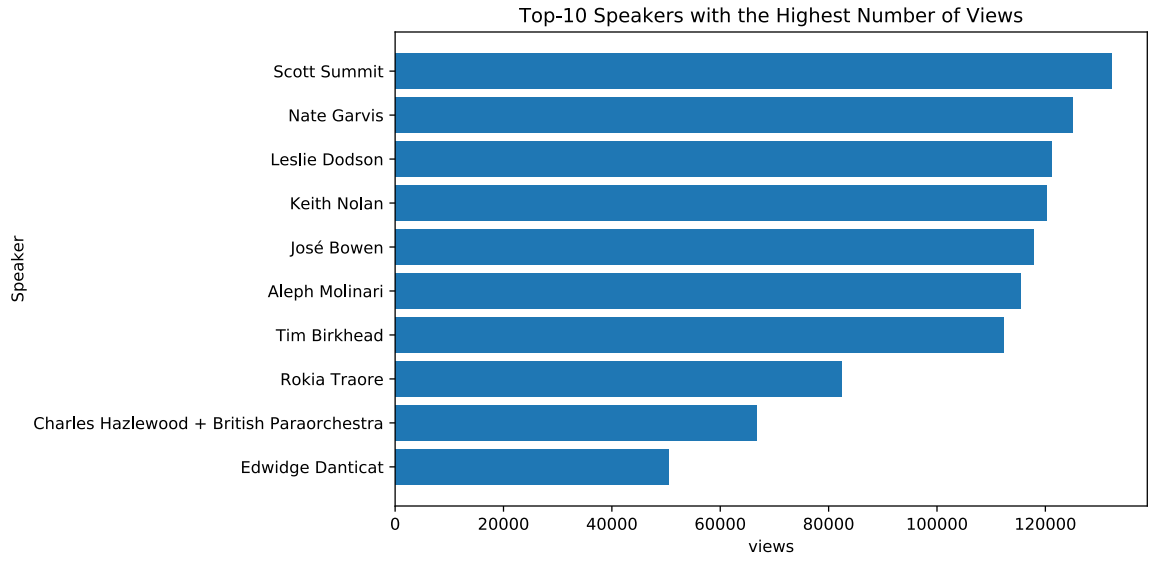


FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Visualizing the data

Let's plot a chart of the top 10 speakers based on number of views

```
sorted_data = ted_main.sort_values(by=['views'])
top_10 = sorted_data.head(10)

plt.barh(top_10["main_speaker"], top_10["views"])
plt.xlabel("views")
plt.ylabel("Speaker")
plt.title("Top-10 Speakers with the Highest Number of Views")
```



References

The following references were used to create this tutorial.

- https://www.pythonprogramming.in/what-is-difference-between-iloc-and-loc-in-pandas.html
- http://data-manual.com/
- https://www.geeksforgeeks.org/python-pandas-series-fillna/
- https://kite.com/python/docs/ast.literal_eval

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY