DataCamp Python Data Science Toolbox 1

================

- [Writing Functions](#writing-functions)

- [Twitter DataFrame Analysis Function](#twitter-dataframe-analysis-function)

- [Scope and User-Defined Functions](#scope-and-user-defined-functions)

- [Default and Flexible Arguments](#default-and-flexible-arguments)

- [Generalize the Twitter DataFrame Analysis Function](#generalize-the-twitter-dataframe-analysis-function)

- [Lambda Functions](#lambda-functions)

- [Error Handling](#error-handling)

- [Add Error Messages to Twitter DataFrame Analysis Function](#add-error-messages-to-twitter-dataframe-analysis-function)

Writing Functions

-----------------

``` python

# Strings in Python

company = 'DataCamp'

object1 = "data" + "analysis" + "visualization"

object2 = 1 \* 3

object3 = "1" \* 3

print(object1, object2, object3)

# Variable types

```

## dataanalysisvisualization 3 111

``` python

x = 4.89

y1 = str(x)

y2 = print(x)

```

## 4.89

``` python

print(type(x), type(y1), type(y2))

# Define the function shout

```

## <class 'float'> <class 'str'> <class 'NoneType'>

``` python

def shout():

"""Print a string with three exclamation marks"""

# Concatenate the strings: shout\_word

shout\_word = 'congratulations' + '!!!'

# Print shout\_word

print(shout\_word)

# Call shout

shout()

# Define shout with the parameter, word

```

## congratulations!!!

``` python

def shout(word):

"""Print a string with three exclamation marks"""

# Concatenate the strings: shout\_word

shout\_word = word + '!!!'

# Print shout\_word

print(shout\_word)

# Call shout with the string 'congratulations'

shout('congratulations')

# Define shout with the parameter, word

```

## congratulations!!!

``` python

def shout(word):

"""Return a string with three exclamation marks"""

# Concatenate the strings: shout\_word

shout\_word = word + '!!!'

# Replace print with return

return(shout\_word)

# Pass 'congratulations' to shout: yell

yell = shout('congratulations')

# Print yell

print(yell)

# Define shout with parameters word1 and word2

```

## congratulations!!!

``` python

def shout(word1, word2):

"""Concatenate strings with three exclamation marks"""

# Concatenate word1 with '!!!': shout1

shout1 = word1 +'!!!'

# Concatenate word2 with '!!!': shout2

shout2 = word2 +'!!!'

# Concatenate shout1 with shout2: new\_shout

new\_shout = shout1 + shout2

# Return new\_shout

return new\_shout

# Pass 'congratulations' and 'you' to shout(): yell

yell = shout('congratulations', 'you')

# Print yell

print(yell)

# Tuples

```

## congratulations!!!you!!!

``` python

nums = (3, 4, 5)

# Unpack nums into num1, num2, and num3

num1,num2,num3 = nums

# Construct even\_nums

even\_nums = (2, 4, 6)

# Define shout\_all with parameters word1 and word2

def shout\_all(word1, word2):

# Concatenate word1 with '!!!': shout1

shout1 = word1 + '!!!'

# Concatenate word2 with '!!!': shout2

shout2 = word2 + '!!!'

# Construct a tuple with shout1 and shout2: shout\_words

shout\_words = (shout1, shout2)

# Return shout\_words

return shout\_words

# Pass 'congratulations' and 'you' to shout\_all(): yell1, yell2

yell1, yell2 = shout\_all('congratulations', 'you')

# Print yell1 and yell2

print(yell1)

```

## congratulations!!!

``` python

print(yell2)

```

## you!!!

Twitter DataFrame Analysis Function

-----------------------------------

``` python

# Import pandas

import pandas as pd

# Import Twitter das as DataFrame

tweets = 'https://assets.datacamp.com/production/course\_1532/datasets/tweets.csv'

df = pd.read\_csv(tweets)

# Define count\_entries()

def count\_entries(df, col\_name):

"""Return a dictionary with counts of

occurrences as value for each key."""

# Initialize an empty dictionary: langs\_count

langs\_count = {}

# Extract column from DataFrame: col

col = df[col\_name]

# Iterate over lang column in DataFrame

for entry in col:

# If the language is in langs\_count, add 1

if entry in langs\_count.keys():

langs\_count[entry] +=1

# Else add the language to langs\_count, set the value to 1

else:

langs\_count[entry] = 1

# Return the langs\_count dictionary

return(langs\_count)

# Call count\_entries(): result

result = count\_entries(df,'lang')

# Print the result

print(result)

```

## {'en': 97, 'et': 1, 'und': 2}

Scope and User-Defined Functions

--------------------------------

``` python

# Create a string: team

team = "teen titans"

# Define change\_team()

def change\_team():

"""Change the value of the global variable team."""

# Use team in global scope

global team

# Change the value of team in global: team

team = 'justice league'

# Print team

print(team)

# Call change\_team()

```

## teen titans

``` python

change\_team()

# Print team

print(team)

# Define three\_shouts

```

## justice league

``` python

def three\_shouts(word1, word2, word3):

"""Returns a tuple of strings

concatenated with '!!!'."""

# Define inner

def inner(word):

"""Returns a string concatenated with '!!!'."""

return word + '!!!'

# Return a tuple of strings

return (inner(word1), inner(word2), inner(word3))

# Call three\_shouts() and print

print(three\_shouts('a', 'b', 'c'))

# Define echo

```

## ('a!!!', 'b!!!', 'c!!!')

``` python

def echo(n):

"""Return the inner\_echo function."""

# Define inner\_echo

def inner\_echo(word1):

"""Concatenate n copies of word1."""

echo\_word = word1 \* n

return echo\_word

# Return inner\_echo

return inner\_echo

# Call echo: twice

twice = echo(2)

# Call echo: thrice

thrice = echo(3)

# Call twice() and thrice() then print

print(twice('hello'), thrice('hello'))

# Define echo\_shout()

```

## hellohello hellohellohello

``` python

def echo\_shout(word):

"""Change the value of a nonlocal variable"""

# Concatenate word with itself: echo\_word

echo\_word = word + word

#Print echo\_word

print(echo\_word)

# Define inner function shout()

def shout():

"""Alter a variable in the enclosing scope"""

#Use echo\_word in nonlocal scope

nonlocal echo\_word

#Change echo\_word to echo\_word concatenated with '!!!'

echo\_word = echo\_word + '!!!'

# Call function shout()

shout()

#Print echo\_word

print(echo\_word)

#Call function echo\_shout() with argument 'hello'

echo\_shout('hello')

```

## hellohello

## hellohello!!!

Default and Flexible Arguments

------------------------------

``` python

# Define shout\_echo

def shout\_echo(word1, echo = 1):

"""Concatenate echo copies of word1 and three

exclamation marks at the end of the string."""

# Concatenate echo copies of word1 using \*: echo\_word

echo\_word = word1 \* echo

# Concatenate '!!!' to echo\_word: shout\_word

shout\_word = echo\_word + '!!!'

# Return shout\_word

return shout\_word

# Call shout\_echo() with "Hey": no\_echo

no\_echo = shout\_echo('Hey')

# Call shout\_echo() with "Hey" and echo=5: with\_echo

with\_echo = shout\_echo('Hey', echo = 5)

# Print no\_echo and with\_echo

print(no\_echo)

```

## Hey!!!

``` python

print(with\_echo)

# Define shout\_echo

```

## HeyHeyHeyHeyHey!!!

``` python

def shout\_echo(word1, echo = 1, intense = False):

"""Concatenate echo copies of word1 and three

exclamation marks at the end of the string."""

# Concatenate echo copies of word1 using \*: echo\_word

echo\_word = word1 \* echo

# Capitalize echo\_word if intense is True

if intense is True:

# Capitalize and concatenate '!!!': echo\_word\_new

echo\_word\_new = echo\_word.upper() + '!!!'

else:

# Concatenate '!!!' to echo\_word: echo\_word\_new

echo\_word\_new = echo\_word + '!!!'

# Return echo\_word\_new

return echo\_word\_new

# Call shout\_echo() with "Hey", echo=5 and intense=True: with\_big\_echo

with\_big\_echo = shout\_echo('Hey', echo = 5, intense = True)

# Call shout\_echo() with "Hey" and intense=True: big\_no\_echo

big\_no\_echo = shout\_echo('Hey', intense = True)

# Print values

print(with\_big\_echo)

```

## HEYHEYHEYHEYHEY!!!

``` python

print(big\_no\_echo)

# Define gibberish

```

## HEY!!!

``` python

def gibberish(\*args):

"""Concatenate strings in \*args together."""

# Initialize an empty string: hodgepodge

hodgepodge = ''

# Concatenate the strings in args

for word in args:

hodgepodge += word

# Return hodgepodge

return hodgepodge

# Call gibberish() with one string: one\_word

one\_word = gibberish('luke')

# Call gibberish() with five strings: many\_words

many\_words = gibberish("luke", "leia", "han", "obi", "darth")

# Print one\_word and many\_words

print(one\_word)

```

## luke

``` python

print(many\_words)

# Define report\_status

```

## lukeleiahanobidarth

``` python

def report\_status(\*\*kwargs):

"""Print out the status of a movie character."""

print("\nBEGIN: REPORT\n")

# Iterate over the key-value pairs of kwargs

for key, value in kwargs.items():

# Print out the keys and values, separated by a colon ':'

print(key + ": " + value)

print("\nEND REPORT")

# First call to report\_status()

report\_status(name="luke", affiliation="jedi", status="missing")

# Second call to report\_status()

```

##

## BEGIN: REPORT

##

## name: luke

## affiliation: jedi

## status: missing

##

## END REPORT

``` python

report\_status(name='anakin', affiliation='sith lord', status='deceased')

```

##

## BEGIN: REPORT

##

## name: anakin

## affiliation: sith lord

## status: deceased

##

## END REPORT

Generalize the Twitter DataFrame Analysis Function

--------------------------------------------------

``` python

# Import pandas

import pandas as pd

# Import Twitter das as DataFrame

tweets = 'https://assets.datacamp.com/production/course\_1532/datasets/tweets.csv'

tweets\_df = pd.read\_csv(tweets)

# Define count\_entries()

def count\_entries(df, col\_name = 'lang'):

"""Return a dictionary with counts of

occurrences as value for each key."""

# Initialize an empty dictionary: cols\_count

cols\_count = {}

# Extract column from DataFrame: col

col = df[col\_name]

# Iterate over the column in DataFrame

for entry in col:

# If entry is in cols\_count, add 1

if entry in cols\_count.keys():

cols\_count[entry] += 1

# Else add the entry to cols\_count, set the value to 1

else:

cols\_count[entry] = 1

# Return the cols\_count dictionary

return cols\_count

# Call count\_entries(): result1

result1 = count\_entries(tweets\_df)

# Call count\_entries(): result2

result2 = count\_entries(tweets\_df, 'source')

# Print result1 and result2

print(result1)

```

## {'en': 97, 'et': 1, 'und': 2}

``` python

print(result2)

# Define count\_entries()

```

## {'<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>': 24, '<a href="http://www.facebook.com/twitter" rel="nofollow">Facebook</a>': 1, '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>': 26, '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>': 33, '<a href="http://www.twitter.com" rel="nofollow">Twitter for BlackBerry</a>': 2, '<a href="http://www.google.com/" rel="nofollow">Google</a>': 2, '<a href="http://twitter.com/#!/download/ipad" rel="nofollow">Twitter for iPad</a>': 6, '<a href="http://linkis.com" rel="nofollow">Linkis.com</a>': 2, '<a href="http://rutracker.org/forum/viewforum.php?f=93" rel="nofollow">newzlasz</a>': 2, '<a href="http://ifttt.com" rel="nofollow">IFTTT</a>': 1, '<a href="http://www.myplume.com/" rel="nofollow">Plume\xa0for\xa0Android</a>': 1}

``` python

def count\_entries(df, \*args):

"""Return a dictionary with counts of

occurrences as value for each key."""

#Initialize an empty dictionary: cols\_count

cols\_count = {}

# Iterate over column names in args

for col\_name in args:

# Extract column from DataFrame: col

col = df[col\_name]

# Iterate over the column in DataFrame

for entry in col:

# If entry is in cols\_count, add 1

if entry in cols\_count.keys():

cols\_count[entry] += 1

# Else add the entry to cols\_count, set the value to 1

else:

cols\_count[entry] = 1

# Return the cols\_count dictionary

return cols\_count

# Call count\_entries(): result1

result1 = count\_entries(tweets\_df, 'lang')

# Call count\_entries(): result2

result2 = count\_entries(tweets\_df, 'lang', 'source')

# Print result1 and result2

print(result1)

```

## {'en': 97, 'et': 1, 'und': 2}

``` python

print(result2)

```

## {'en': 97, 'et': 1, 'und': 2, '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>': 24, '<a href="http://www.facebook.com/twitter" rel="nofollow">Facebook</a>': 1, '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>': 26, '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>': 33, '<a href="http://www.twitter.com" rel="nofollow">Twitter for BlackBerry</a>': 2, '<a href="http://www.google.com/" rel="nofollow">Google</a>': 2, '<a href="http://twitter.com/#!/download/ipad" rel="nofollow">Twitter for iPad</a>': 6, '<a href="http://linkis.com" rel="nofollow">Linkis.com</a>': 2, '<a href="http://rutracker.org/forum/viewforum.php?f=93" rel="nofollow">newzlasz</a>': 2, '<a href="http://ifttt.com" rel="nofollow">IFTTT</a>': 1, '<a href="http://www.myplume.com/" rel="nofollow">Plume\xa0for\xa0Android</a>': 1}

Lambda Functions

----------------

``` python

# Define echo\_word as a lambda function: echo\_word

echo\_word = (lambda word1, echo: word1 \* echo)

# Call echo\_word: result

result = echo\_word('hey',5)

# Print result

print(result)

# Create a list of strings: spells

```

## heyheyheyheyhey

``` python

spells = ["protego", "accio", "expecto patronum", "legilimens"]

# Use map() to apply a lambda function over spells: shout\_spells

shout\_spells = map(lambda item: item + '!!!', spells)

# Convert shout\_spells to a list: shout\_spells\_list

shout\_spells\_list = list(shout\_spells)

# Convert shout\_spells into a list and print it

print(shout\_spells\_list)

# Create a list of strings: fellowship

```

## ['protego!!!', 'accio!!!', 'expecto patronum!!!', 'legilimens!!!']

``` python

fellowship = ['frodo', 'samwise', 'merry', 'aragorn', 'legolas', 'boromir', 'gimli']

# Use filter() to apply a lambda function over fellowship: result

result = filter(lambda member: len(member) > 6, fellowship)

# Convert result to a list: result\_list

result\_list = list(result)

# Convert result into a list and print it

print(result\_list)

# Import reduce from functools

```

## ['samwise', 'aragorn', 'legolas', 'boromir']

``` python

from functools import reduce

# Create a list of strings: stark

stark = ['robb', 'sansa', 'arya', 'eddard', 'jon']

# Use reduce() to apply a lambda function over stark: result

result = reduce(lambda item1, item2: item1 + item2, stark)

# Print the result

print(result)

```

## robbsansaaryaeddardjon

Error Handling

--------------

``` python

# Define shout\_echo

def shout\_echo(word1, echo=1):

"""Concatenate echo copies of word1 and three

exclamation marks at the end of the string."""

# Initialize empty strings: echo\_word, shout\_words

echo\_word = ''

shout\_words = ''

# Add exception handling with try-except

try:

# Concatenate echo copies of word1 using \*: echo\_word

echo\_word = word1 \* echo

# Concatenate '!!!' to echo\_word: shout\_words

shout\_words = echo\_word + '!!!'

except:

# Print error message

print("word1 must be a string and echo must be an integer.")

# Return shout\_words

return shout\_words

# Call shout\_echo

shout\_echo("particle", echo="accelerator")

# Define shout\_echo

```

## word1 must be a string and echo must be an integer.

``` python

def shout\_echo(word1, echo=1):

"""Concatenate echo copies of word1 and three

exclamation marks at the end of the string."""

# Raise an error with raise

if echo < 0:

raise ValueError('echo must be greater than 0')

# Concatenate echo copies of word1 using \*: echo\_word

echo\_word = word1 \* echo

# Concatenate '!!!' to echo\_word: shout\_word

shout\_word = echo\_word + '!!!'

# Return shout\_word

return shout\_word

# Call shout\_echo

shout\_echo("particle", echo=5)

```

Add Error Messages to Twitter DataFrame Analysis Function

---------------------------------------------------------

``` python

# Import pandas

import pandas as pd

# Import Twitter das as DataFrame

tweets = 'https://assets.datacamp.com/production/course\_1532/datasets/tweets.csv'

tweets\_df = pd.read\_csv(tweets)

# Select retweets from the Twitter DataFrame: result

result = filter(lambda x : x[0:2] == 'RT', tweets\_df['text'])

# Create list from filter object result: res\_list

res\_list = list(result)

# Print all retweets in res\_list

for tweet in res\_list:

print(tweet)

# Define count\_entries()

```

## RT @bpolitics: .@krollbondrating's Christopher Whalen says Clinton is the weakest Dem candidate in 50 years https://t.co/pLk7rvoRSn https:/…

RT @HeidiAlpine: @dmartosko Cruz video found.....racing from the scene.... #cruzsexscandal https://t.co/zuAPZfQDk3

RT @AlanLohner: The anti-American D.C. elites despise Trump for his America-first foreign policy. Trump threatens their gravy train. https:…

RT @BIackPplTweets: Young Donald trump meets his neighbor https://t.co/RFlu17Z1eE

RT @trumpresearch: @WaitingInBagdad @thehill Trump supporters have selective amnisia.

RT @HouseCracka: 29,000+ PEOPLE WATCHING TRUMP LIVE ON ONE STREAM!!!

https://t.co/7QCFz9ehNe

RT @urfavandtrump: RT for Brendon Urie

Fav for Donald Trump https://t.co/PZ5vS94lOg

RT @trapgrampa: This is how I see #Trump every time he speaks. https://t.co/fYSiHNS0nT

RT @trumpresearch: @WaitingInBagdad @thehill Trump supporters have selective amnisia.

RT @Pjw20161951: NO KIDDING: #SleazyDonald just attacked Scott Walker for NOT RAISING TAXES in WI! #LyinTrump

#NeverTrump #CruzCrew https…

RT @urfavandtrump: RT for Brendon Urie

Fav for Donald Trump https://t.co/PZ5vS94lOg

RT @ggreenwald: The media spent all day claiming @SusanSarandon said she might vote for Trump. A total fabrication, but whatever... https:/…

RT @Pjw20161951: NO KIDDING: #SleazyDonald just attacked Scott Walker for NOT RAISING TAXES in WI! #LyinTrump

#NeverTrump #CruzCrew https…

RT @trapgrampa: This is how I see #Trump every time he speaks. https://t.co/fYSiHNS0nT

RT @mitchellvii: So let me get this straight. Any reporter can assault Mr Trump at any time and Corey can do nothing? Michelle is clearly…

RT @paulbenedict7: How #Trump Sacks RINO Strongholds by Hitting Positions Held by Dems and GOP https://t.co/D7ulnAJhis #tcot #PJNET https…

RT @DRUDGE\_REPORT: VIDEO: Trump emotional moment with Former Miss Wisconsin who has terminal illness... https://t.co/qt06aG9inT

RT @ggreenwald: The media spent all day claiming @SusanSarandon said she might vote for Trump. A total fabrication, but whatever... https:/…

RT @DennisApgar: Thank God I seen Trump at first stop in Wisconsin media doesn't know how great he is, advice watch live streaming https://…

RT @paulbenedict7: How #Trump Sacks RINO Strongholds by Hitting Positions Held by Dems and GOP https://t.co/D7ulnAJhis #tcot #PJNET https…

RT @DRUDGE\_REPORT: VIDEO: Trump emotional moment with Former Miss Wisconsin who has terminal illness... https://t.co/qt06aG9inT

RT @DennisApgar: Thank God I seen Trump at first stop in Wisconsin media doesn't know how great he is, advice watch live streaming https://…

RT @mitchellvii: So let me get this straight. Any reporter can assault Mr Trump at any time and Corey can do nothing? Michelle is clearly…

RT @sciam: Trump's idiosyncratic patterns of speech are why people tend either to love or hate him https://t.co/QXwquVgs3c https://t.co/P9N…

RT @Norsu2: Nightmare WI poll for Ted Cruz has Kasich surging: Trump 29, Kasich 27, Cruz 25. https://t.co/lJsgbLYY1P #NeverTrump

RT @thehill: WATCH: Protester pepper-sprayed point blank at Trump rally https://t.co/B5f65Al9ld https://t.co/skAfByXuQc

RT @sciam: Trump's idiosyncratic patterns of speech are why people tend either to love or hate him https://t.co/QXwquVgs3c https://t.co/P9N…

RT @ggreenwald: The media spent all day claiming @SusanSarandon said she might vote for Trump. A total fabrication, but whatever... https:/…

RT @DebbieStout5: Wow! Last I checked it was just 12 points &amp; that wasn't more than a day ago. Oh boy Trump ppl might want to rethink攼㹤愼㸰戼㹥攼㹤戼㸴㤼㸴 http…

RT @tyleroakley: i'm a messy bitch, but at least i'm not voting for trump

RT @vandives: Trump supporters r tired of justice NOT being served. There's no justice anymore. Hardworking Americans get screwed. That's n…

RT @AP: BREAKING: Trump vows to stand by campaign manager charged with battery, says he does not discard people.

RT @AP: BREAKING: Trump vows to stand by campaign manager charged with battery, says he does not discard people.

RT @urfavandtrump: RT for Jerrie (Little Mix)

Fav for Donald Trump https://t.co/nEVxElW6iG

RT @urfavandtrump: RT for Jerrie (Little Mix)

Fav for Donald Trump https://t.co/nEVxElW6iG

RT @NoahCRothman: When Walker was fighting for reforms, Trump was defending unions and collective bargaining privileges https://t.co/e1UWNN…

RT @RedheadAndRight: Report: Secret Service Says Michelle Fields Touched Trump https://t.co/c5c2sD8VO2

This is the only article you will n…

RT @AIIAmericanGirI: VIDEO=&gt; Anti-Trump Protester SLUGS Elderly Trump Supporter in the Face

https://t.co/GeEryMDuDY

RT @NoahCRothman: When Walker was fighting for reforms, Trump was defending unions and collective bargaining privileges https://t.co/e1UWNN…

RT @JusticeRanger1: @realDonaldTrump @Pudingtane @DanScavino @GOP @infowars @EricTrump

URGENT PUBLIC TRUMP ALERT:

COVERT KILL MEANS https:…

RT @AIIAmericanGirI: VIDEO=&gt; Anti-Trump Protester SLUGS Elderly Trump Supporter in the Face

https://t.co/GeEryMDuDY

RT @RedheadAndRight: Report: Secret Service Says Michelle Fields Touched Trump https://t.co/c5c2sD8VO2

This is the only article you will n…

RT @JusticeRanger1: @realDonaldTrump @Pudingtane @DanScavino @GOP @infowars @EricTrump

URGENT PUBLIC TRUMP ALERT:

COVERT KILL MEANS https:…

RT @Schneider\_CM: Trump says nobody had ever heard of executive orders before Obama started signing them. Never heard of the Emancipation P…

RT @RonBasler1: @DavidWhitDennis @realDonaldTrump @tedcruz

CRUZ SCREWS HOOKERS

CRUZ / CLINTON

RT @DonaldsAngel: Former Ms. WI just said that she is terminally ill but because of Trump pageant, her 7 yr. old son has his college educat…

RT @Schneider\_CM: Trump says nobody had ever heard of executive orders before Obama started signing them. Never heard of the Emancipation P…

RT @DonaldsAngel: Former Ms. WI just said that she is terminally ill but because of Trump pageant, her 7 yr. old son has his college educat…

RT @Dodarey: @DR8801 @SykesCharlie Charlie, let's see you get a straight "yes" or "no" answer from Cruz a/b being unfaithful to his wife @T…

RT @RonBasler1: @DavidWhitDennis @realDonaldTrump @tedcruz

CRUZ SCREWS HOOKERS

CRUZ / CLINTON

RT @RockCliffOne: Remember when the idea of a diabolical moron holding the world hostage was an idea for a funny movie? #Trump #GOP https:/…

RT @HillaryClinton: "Every day, another Republican bemoans the rise of Donald Trump... but [he] didn’t come out of nowhere." —Hillary

https…

RT @Dodarey: @DR8801 @SykesCharlie Charlie, let's see you get a straight "yes" or "no" answer from Cruz a/b being unfaithful to his wife @T…

RT @HillaryClinton: "Every day, another Republican bemoans the rise of Donald Trump... but [he] didn’t come out of nowhere." —Hillary

https…

RT @RockCliffOne: Remember when the idea of a diabolical moron holding the world hostage was an idea for a funny movie? #Trump #GOP https:/…

RT @immigrant4trump: @immigrant4trump msm, cable news attacking trump all day, from 8am to 10pm today, then the reruns come on, repeating t…

RT @immigrant4trump: @immigrant4trump msm, cable news attacking trump all day, from 8am to 10pm today, then the reruns come on, repeating t…

RT @GlendaJazzey: Donald Trump’s Campaign Financing Dodge, @rrotunda https://t.co/L8flI4lswG via @VerdictJustia

RT @TUSK81: LOUDER FOR THE PEOPLE IN THE BACK https://t.co/hlPVyNLXzx

RT @loopzoop: Well...put it back https://t.co/8Yb7BDT5VM

RT @claytoncubitt: Stop asking Bernie supporters if they’ll vote for Hillary against Trump. We got a plan to beat Trump already. Called Ber…

RT @akaMaude13: Seriously can't make this up. What a joke. #NeverTrump https://t.co/JkTx6mdRgC

``` python

def count\_entries(df, col\_name='lang'):

"""Return a dictionary with counts of

occurrences as value for each key."""

# Initialize an empty dictionary: cols\_count

cols\_count = {}

# Add try block

try:

# Extract column from DataFrame: col

col = df[col\_name]

# Iterate over the column in dataframe

for entry in col:

# If entry is in cols\_count, add 1

if entry in cols\_count.keys():

cols\_count[entry] += 1

# Else add the entry to cols\_count, set the value to 1

else:

cols\_count[entry] = 1

# Return the cols\_count dictionary

return cols\_count

# Add except block

except:

print('The DataFrame does not have a ' + col\_name + ' column.')

# Call count\_entries(): result1

result1 = count\_entries(tweets\_df, 'lang')

# Print result1

print(result1)

# Call count\_entries(): result2

```

## {'en': 97, 'et': 1, 'und': 2}

``` python

result2 = count\_entries(tweets\_df, 'lang1')

# Define count\_entries()

```

## The DataFrame does not have a lang1 column.

``` python

def count\_entries(df, col\_name='lang'):

"""Return a dictionary with counts of

occurrences as value for each key."""

# Raise a ValueError if col\_name is NOT in DataFrame

if col\_name not in df.columns:

raise ValueError('The DataFrame does not have a ' + col\_name + ' column.')

# Initialize an empty dictionary: cols\_count

cols\_count = {}

# Extract column from DataFrame: col

col = df[col\_name]

# Iterate over the column in DataFrame

for entry in col:

# If entry is in cols\_count, add 1

if entry in cols\_count.keys():

cols\_count[entry] += 1

# Else add the entry to cols\_count, set the value to 1

else:

cols\_count[entry] = 1

# Return the cols\_count dictionary

return cols\_count

# Call count\_entries(): result1

result1 = count\_entries(tweets\_df, 'lang')

# Print result1

print(result1)

```

## {'en': 97, 'et': 1, 'und': 2}