“Design” Process Pictures only

- (11/28)

```
#Beautiful Soup Approach
#import requests

#URL = 'https://www.monster.com/jobs/search/?q=Software-Developer&where=Australia'
#page = requests.get(URL)
#print(page.content)

#Scraping Bee Approach
#from selenium import webdriver


#DRIVER_PATH = '/path/to/chromedriver'
#driver = webdriver.Chrome(executable_path=DRIVER_PATH)
#driver.get('https://google.com')

#Webdriver Test Code
import time
from selenium import webdriver

driver = webdriver.Chrome('C:\\Users\\zsori\\AppData\\Roaming\\Webdriver\\chromedriver')  # Optional argument, if not specified will search path.
driver.get('https://www.youtube.com/user/VanossGaming');
time.sleep(5) # Let the user actually see something!
#search_box = driver.find_element_by_name('q')
#search_box.send_keys('ChromeDriver')
#search_box.submit()
#time.sleep(5) # Let the user actually see something!
driver.quit()
```

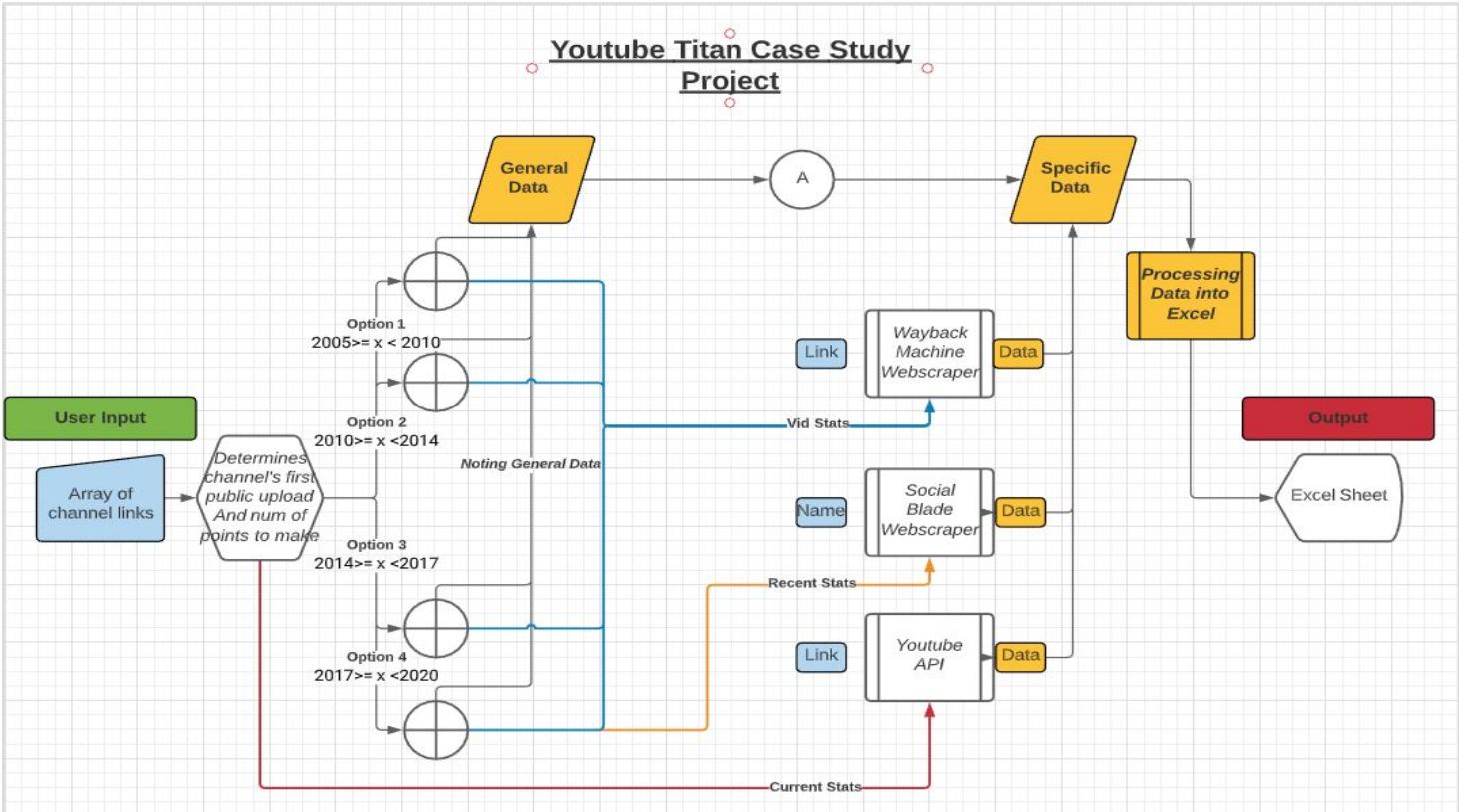
```
#Selenium Webdriver Login test code
import time
from selenium import webdriver

driver = webdriver.Chrome('C:\\Users\\zsori\\AppData\\Roaming\\Webdriver\\chromedriver')
driver.get("https://news.ycombinator.com/login")

time.sleep(5) # Let the user actually see something!

login = driver.find_element_by_xpath("//input").send_keys('USERNAME')
password = driver.find_element_by_xpath("//input[@type='password']").send_keys('PASSWORD')
submit = driver.find_element_by_xpath("//input[@value='login']").click()
time.sleep(5) # Let the user actually see something!
```

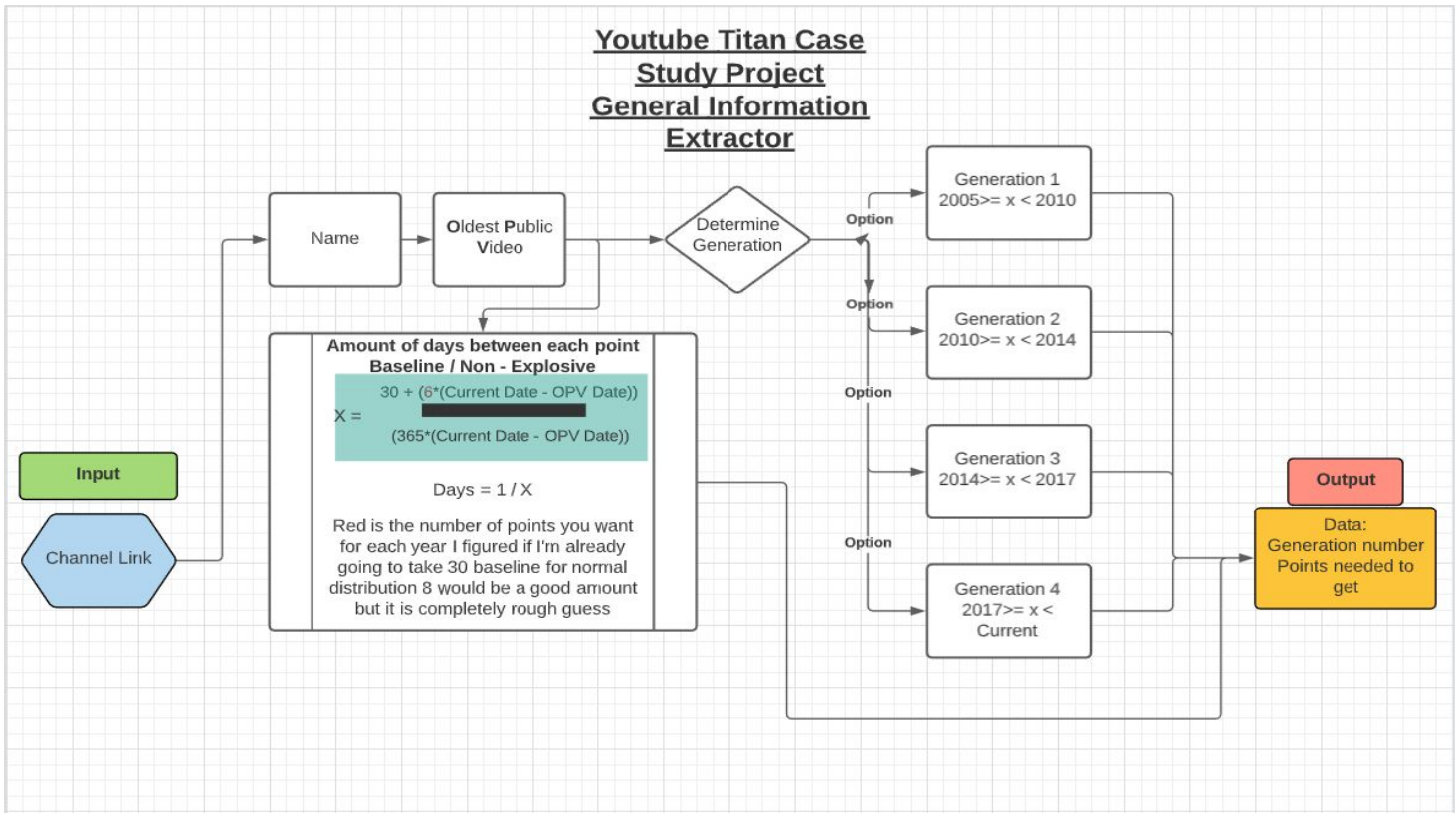
```
#Actual webdriver
import time
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

options = Options()
#options.headless = True
#options.add_argument("--window-size=1920,1200")

driver = webdriver.Chrome(options = options, executable_path = 'C:\\Users\\zsori\\AppData\\Roaming\\Webdriver\\chromedriver')
driver.get('https://www.youtube.com/channel/UCTkXRDQl0luXxVQrRQvWS6w');
print(driver.title)
literallywtf = driver.find_element_by_id('subscriber-count')
print(literallywtf)

driver.quit()
```

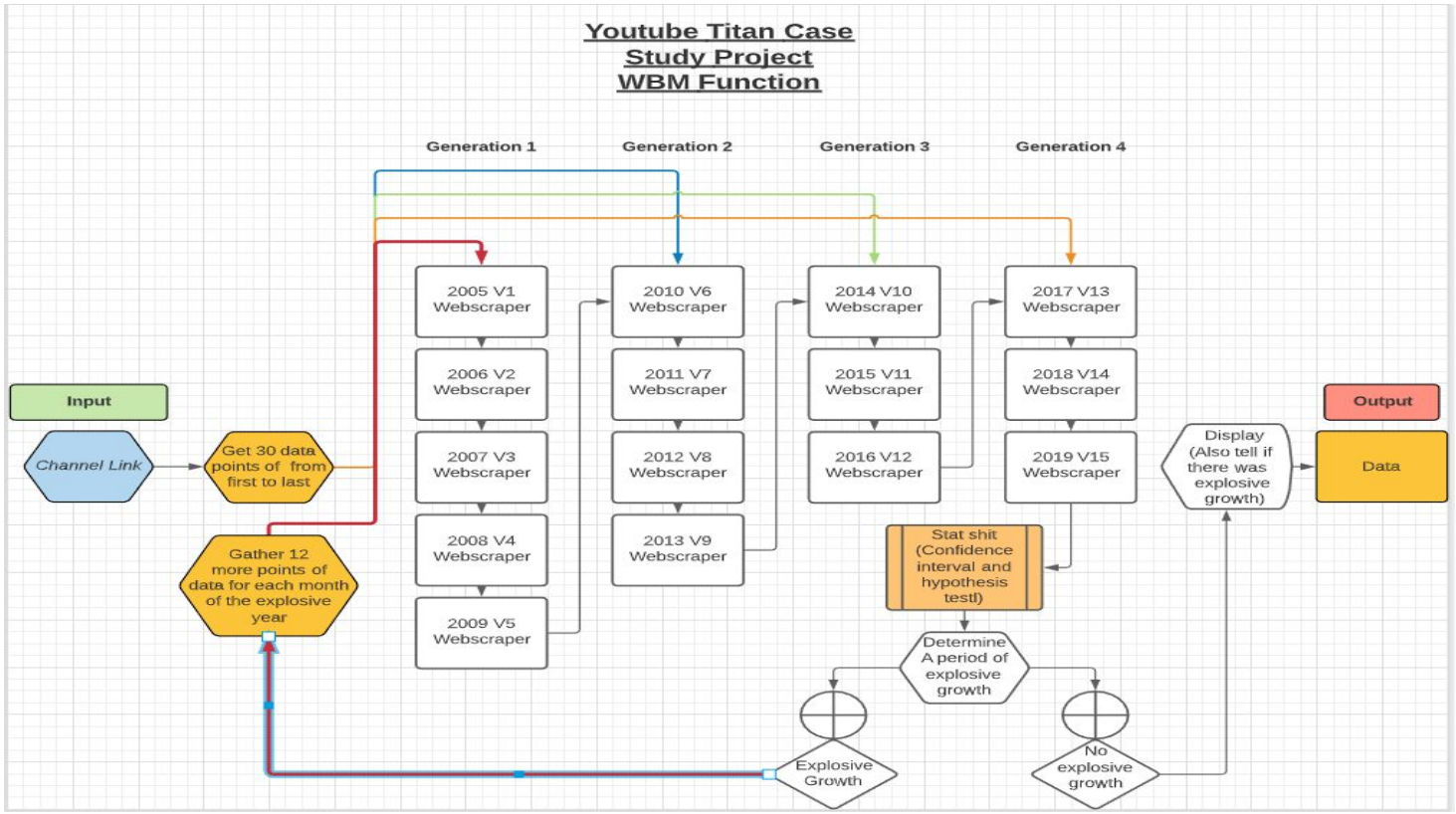
- (11/30)
- (12/1)

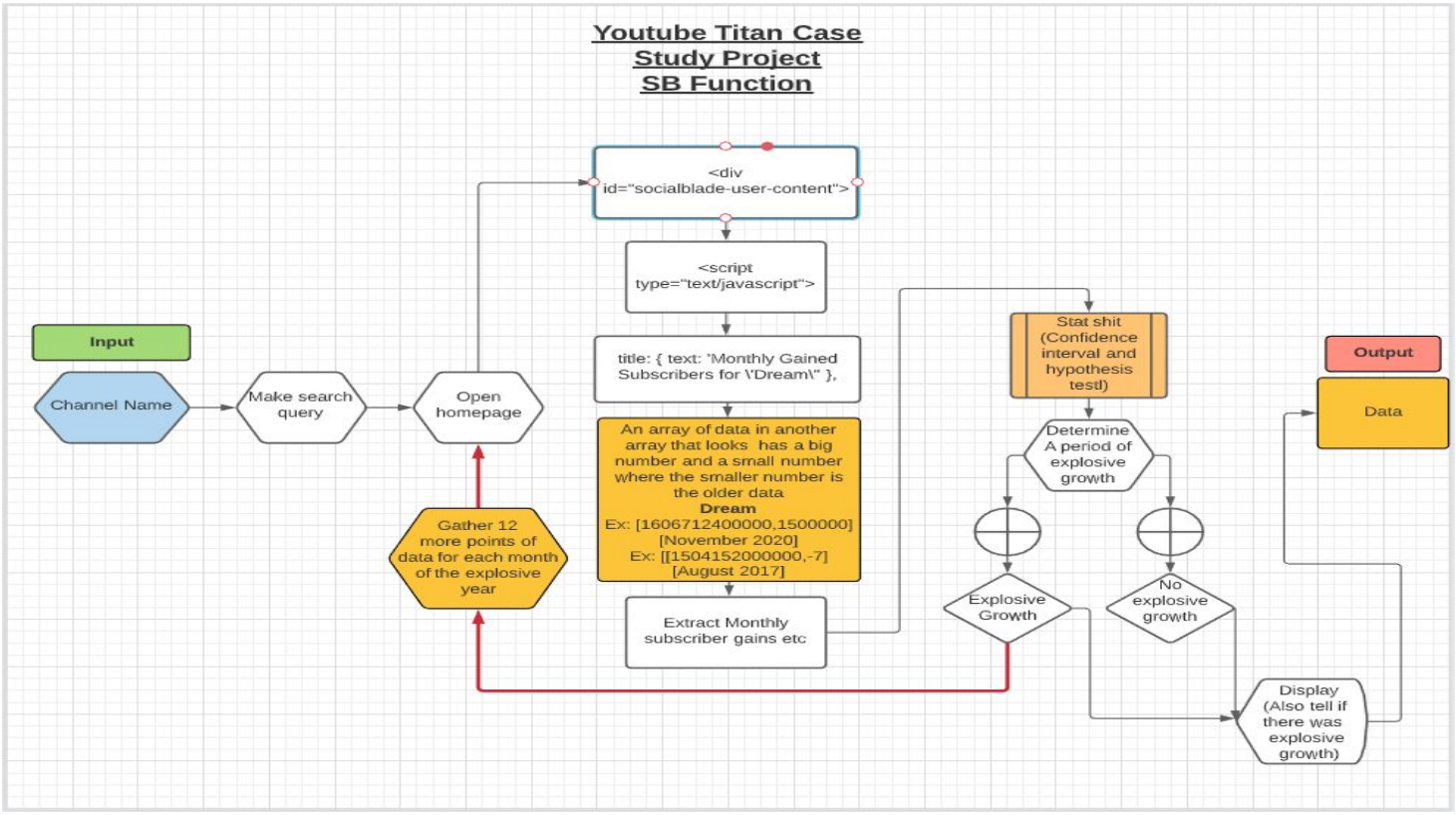
```
#Actual webdriver
import time
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

options = Options()
#options.headless = True
#options.add_argument("--window-size=1920,1200")

driver = webdriver.Chrome(options = options, executable_path = 'C:\\Users\\zsori\\AppData\\Roaming\\Webdriver\\chromedriver')
driver.get('https://www.youtube.com/user/VanossGaming');
print(driver.title)
literallywtf = driver.find_element_by_id('subscriber-count')
print(literallywtf)

driver.get('https://www.youtube.com/user/DreamTraps');
print(driver.title)

#This depends on the way the page is formatted for each channel so you need to navigate to the video section and select the most popular and the first 5 videos
#Subscriber
xpathsubscriber = '/html/body/ytd-app/div/ytd-page-manager/ytd-browse/div[3]/ytd-c4-tabbed-header-renderer/app-header-layout/div/app-header/div[2]/div[2]/div/div[1]/div/div[1]/yt-formatted-string'
subscriber = driver.find_element_by_xpath(xpathsubscriber)
print(subscriber.get_attribute('innerHTML'))

#Most popular video title
xpathnum1vid = '/html/body/ytd-app/div/ytd-page-manager/ytd-browse/ytd-two-column-browse-results-renderer/div[1]/ytd-section-list-renderer/div[2]/ytd-item-section-renderer[1]/div[3]/ytd-shelf-rend
num1vid = driver.find_element_by_xpath(xpathnum1vid)
print(num1vid.get_attribute('innerHTML'))

#Most popular video views
xpathnum1vidviews = '/html/body/ytd-app/div/ytd-page-manager/ytd-browse/ytd-two-column-browse-results-renderer/div[1]/ytd-section-list-renderer/div[2]/ytd-item-section-renderer[1]/div[3]/ytd-shelf
num1vidviews = driver.find_element_by_xpath(xpathnum1vidviews)
print(num1vidviews.get_attribute('innerHTML'))

driver.quit()
```

- (12/29)

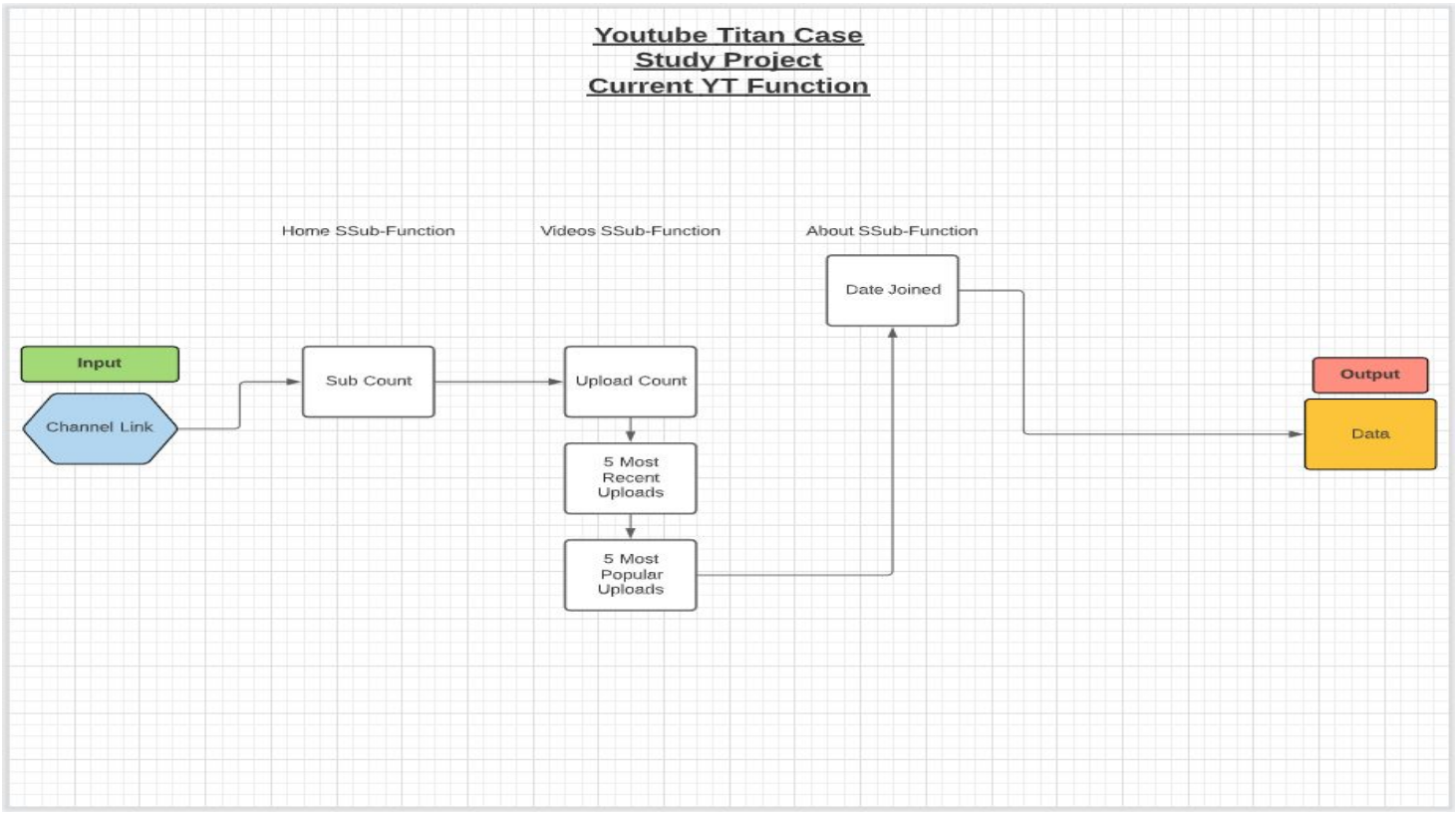



Youtube Titan Case Study Project
General Information Extractor
Input
Channel Link
Name
Oldest Public Video
Determine Generation
Option
Option
Option
Option
Generation 1
2005>= x < 2010
Generation 2
2010>= x < 2014
Generation 3
2014>= x < 2017
Generation 4
2017>= x < Current
Amount of days between each point
Baseline / Non - Explosive
X = 30 + (6*(Current Date - OPV Date)) / (365*(Current Date - OPV Date))
Days = 1 / X
Red is the number of points you want for each year I figured if I'm already going to take 30 baseline for normal distribution 8 would be a good amount but it is completely rough guess
Output
Data:
Generation number
Points needed to get


Youtube Titan Case Study Project
WBM Function
Generation 1
Generation 2
Generation 3
Generation 4
Input
Channel Link
Get 30 data points of from first to last
Gather 12 more points of data for each month of the explosive year
2005 V1 Webscraper
2006 V2 Webscraper
2007 V3 Webscraper
2008 V4 Webscraper
2009 V5 Webscraper
2010 V6 Webscraper
2011 V7 Webscraper
2012 V8 Webscraper
2013 V9 Webscraper
2014 V10 Webscraper
2015 V11 Webscraper
2016 V12 Webscraper
2017 V13 Webscraper
2018 V14 Webscraper
2019 V15 Webscraper
Stat shit (Confidence interval and hypothesis testl)
Determine A period of explosive growth
Explosive Growth
No explosive growth
Display (Also tell if there was explosive growth)
Output
Data


Youtube Titan Case Study Project SB Function

<script type="text/javascript">
title: { text: 'Monthly Gained Subscribers for \'Dream\'' },
An array of data in another array that looks has a big number and a small number where the smaller number is the older data
Dream
Ex: [1606712400000,1500000] [November 2020]
Ex: [[1504152000000,-7] [August 2017]
Extract Monthly subscriber gains etc
Stat shit (Confidence interval and hypothesis test!)
Determine A period of explosive growth
Explosive Growth
No explosive growth
Input
Channel Name
Make search query
Open homepage
Gather 12 more points of data for each month of the explosive year
Output
Data
Display (Also tell if there was explosive growth)


Youtube Titan Case Study Project Current YT Function
Home SSub-Function
Videos SSub-Function
About SSub-Function
Input
Channel Link
Sub Count
Upload Count
5 Most Recent Uploads
5 Most Popular Uploads
Date Joined
Output
Data

- (12/30)

```
def Youtube_Complete_History_Webscraper():

# User Input
# How many youtubers are you gathering data on and which ones are they

    def mainUserInput():
        mainUserLinks = []
        numYoutubers = int(input('How many youtubers are you researching today?: '))
        for i in range(0,numYoutubers):
            mainUserLinks.append(i)
            mainUserLinks[i] = str(input('What is this youtubers main link? (youtube.com/user...): '))
        return mainUserLinks
    print(mainUserLinks)

    #def generalInfoExtractor_SF(mainUserLinks):
     #   print(mainUserLinks)

 #Running everything
    mainUserInput()
   # generalInfoExtractor_SF(mainUserLinks)


Youtube_Complete_History_Webscraper()
```

- (12/31)

```python
#Initializing Processes
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

options = Options()
options.headless = True
options.add_argument("--window-size=1920,1200")

Driver_path = 'C:\\Users\\zsori\\AppData\\Roaming\\Webdriver\\chromedriver'

driver = webdriver.Chrome(options =options, executable_path = Driver_path)

#Main Function Definition
def Youtube_Complete_History_Webscraper():

# User Input
# How many youtubers are you gathering data on and which ones are they

    def mainUserInput():
        mainUserLinks = []
        numYoutubers = int(input('How many youtubers are you researching today?: '))
        for i in range(0,numYoutubers):
            mainUserLinks.append(i)
            mainUserLinks[i] = str(input('What is this youtubers main link? (youtube.com/user...): '))

        output = [numYoutubers,mainUserLinks]
        return output

# General information extractor

    def generalInfoExtractor_SF(mainUserInput):
        numYoutubers = mainUserInput[0]
        mainUserLinks = mainUserInput[1]

        for i in range (0,numYoutubers):
            print(mainUserLinks[i])
            driver.get(mainUserLinks[i])
            print(driver.title)

 #Main Execution Order
    mainUserInput = mainUserInput()
    generalInfoExtractor_SF(mainUserInput)

Youtube_Complete_History_Webscraper()
```

- (1/1)

```
        driver.page_source
        id = 'metadata-line'
        classname = 'style-scope ytd-grid-video-renderer'

        oldest_video = driver.find_element_by_id(id)
        classname = oldest_video.get_attribute('innerHTML')
        print(classname)
```

```
for i in range (0,numYoutubers):
    driver.get(mainUserLinks[i])
    print(driver.title)
    print(mainUserLinks[i])

    #There will also need to be code here to navigate from the user to the organinzed videos (user--> videos --> oldest videos)

    driver.page_source
    id = 'metadata-line'
    classname = 'style-scope ytd-grid-video-renderer'

    oldest_video = driver.find_element_by_id(id)
    idapproach = (oldest_video.text)

    oldest_video2 = driver.find_element_by_class_name(classname)
    classapproach = (oldest_video2.text)

    #This only works with channels less than 9 years old but is there anyone on my list really older than 10 years when youtube itself was made in 2005
    print(idapproach)
    numYears = len(idapproach)-11
    print(idapproach[numYears])
        #string

    if idapproach == str('8 years'):
        print('pog')
    elif idapproach == str('4.8M views8 years ago'):
        print('fuck')
    else:
        print('goddamnit')
    #print(int(idapproach[0,3],base = 10))

    #class--> innerHTML showed the exact views, and video name but a lot of other stuff too
        #Prints out length of video, title, views, date
    #id --> innerHTML showed abbreviated views, date and not a lot of other stuff
        #Prints out views and date
    #problem remains the same that I can't don't know how to print the exact things im looking for it always comes with extra bs
```

```
#Outputs Class
    class Outputs:
        def __init__ (output,UI):#,GIE,WBM,SB,CYT,Stat):
            output.UI = UIoutput
            #output.GIE = GIEoutput
            #output.WBM = WBMoutput
            #output.SB = SBoutput
            #output.CYT = CYToutput
            #output.Stat = Statoutput

#Main Execution Order
    UIoutput = mainUserInput()

    outputs = Outputs(UIoutput)

    generalInfoExtractor_SF(outputs.UI)

Youtube_Complete_History_Webscraper()
```

- (1/2 - 1/4)(Forgot to take screenshots)

```
How many youtubers are you researching today?: 3
What is this youtubers main link? (youtube.com/user...): https://m.youtube.com/c/MichaelReeves
What is this youtubers main link? (youtube.com/user...): https://www.youtube.com/user/VanossGaming
What is this youtubers main link? (youtube.com/user...): https://www.youtube.com/user/PewDiePie
Michael Reeves - YouTube
https://m.youtube.com/c/MichaelReeves
This person is Generation 4!
VanossGaming - YouTube
https://www.youtube.com/user/VanossGaming
This person is Generation 2!
PewDiePie - YouTube
https://www.youtube.com/user/PewDiePie
This person is Generation 2!
[2017, 2012, 2010]
[4, 2, 2]
[62, 102, 118]
[[4, 2, 2], [62, 102, 118]]
[3, ['https://m.youtube.com/c/MichaelReeves', 'https://www.youtube.com/user/VanossGaming', 'https://www.youtube.com/
ser/PewDiePie']]
[[4, 2, 2], [62, 102, 118]]
Press any key to continue . . .
```

- Input → goes through each link categorizing them → outputs
    - Oldest public video release date, generation #, #of data points to be taken, UI output and GIE output

```
# General information extractor

    def generalInfoExtractor_SF(mainUserInput):

        #Preallocating and unpacking
        numYoutubers = mainUserInput[0]
        mainUserLinks = mainUserInput[1]
        organizedLink = []
        genNum = []
        numDataPoints = []
        uploadYear = []
        GIEoutput = [0,0]

        for i in range (0,numYoutubers):
            #Sorts videos by adding query to base url
            organizedLink.append(i)
            organizedLink[i] = mainUserLinks[i] +'/videos?view=0&sort=da&flow=grid'
            genNum.append(i)
            genNum[i] = 0
            numDataPoints.append(i)
            numDataPoints[i] = 0
            uploadYear.append(i)
            uploadYear[i] = 0

           #selenium gets information
            driver.get(organizedLink[i])
            print(driver.title)
            print(mainUserLinks[i])

            driver.page_source
            id = 'metadata-line'
            classname = 'style-scope ytd-grid-video-renderer'

            oldest_video = driver.find_element_by_id(id)
            idapproach = (oldest_video.text)
            oldest_video2 = driver.find_element_by_class_name(classname)
            classapproach = (oldest_video2.text)
```

```
        #RE helps to find numbers within string of information returning indecies
        numbers = re.compile(r'\d+')
        allnumbers = numbers.findall(classapproach)
        uploadYear[i] = 2020 - int(allnumbers[-1])

        #Getting outputs
            #Generation number
        if (uploadYear[i] >= 2005) & (uploadYear[i] < 2010):
            genNum[i] = 1
            print('This person is Generation ', genNum[i],'!',sep='')
        elif (uploadYear[i] >= 2010) & (uploadYear[i] < 2014):
            genNum[i] = 2
            print('This person is Generation ', genNum[i],'!',sep='')
        elif (uploadYear[i] >= 2014) & (uploadYear[i] < 2017):
            genNum[i] = 3
            print('This person is Generation ', genNum[i],'!',sep='')
        elif (uploadYear[i] >= 2017) & (uploadYear[i] < 2021):
            genNum[i] = 4
            print('This person is Generation ', genNum[i],'!',sep='')
        else:
            print('wowee poggers')

        #Number of points needed
        numDataPoints[i] = int(round(30 + (((8 * (2021-uploadYear[i])) ))))

    GIEoutput[0] = genNum
    GIEoutput[1] = numDataPoints
    print(uploadYear)
    print(genNum)
    print(numDataPoints)
    print(GIEoutput)
    return GIEoutput
```

```
#Outputs Class
    class Outputs:
        def __init__ (output,UI,GIE):#,WBM,SB,CYT,Stat):
            output.UI = UIoutput
            output.GIE = GIEoutput
            #output.WBM = WBMoutput
            #output.SB = SBoutput
            #output.CYT = CYToutput
            #output.Stat = Statoutput

#Main Execution Order
    UIoutput = mainUserInput()

    GIEoutput = generalInfoExtractor_SF(UIoutput)

    outputs = Outputs(UIoutput,GIEoutput)

    print(outputs.UI)
    print(outputs.GIE)
```

- (1/6- 1/13) (Forgot to take Screenshots)

```
DevTools listening on ws://127.0.0.1:50736/devtools/browser/ec42a53a-d9e0-4d65-8303-2590ee503da8
How many youtubers are you researching today?: [19824:24744:0113/010846.298:ERROR:device_event_log_impl.cc(211)] [01:08:4
6.298] USB: usb_device_handle_win.cc:1020 Failed to read descriptor from node connection: A device attached to the system
 is not functioning. (0x1F)
3
What is this youtubers main link? (youtube.com/user...): https://m.youtube.com/c/MichaelReeves
What is this youtubers main link? (youtube.com/user...): https://youtube.com/user/VanossGaming
What is this youtubers main link? (youtube.com/user...): https://youtube.com/user/PewDiePie
Channel: Michael Reeves
Generation Number: 4
OPV Upload Year: 2017
Data Points: 62
Unique ID: UCtHaxi4GTYDpJgMSGy7AeSw

Channel: VanossGaming
Generation Number: 2
OPV Upload Year: 2012
Data Points: 102
Unique ID: UCKqH_9mk1waLgBiL2vT5b9g

Channel: PewDiePie
Generation Number: 2
OPV Upload Year: 2010
Data Points: 118
Unique ID: UC-lHJZR3Gqxm24_Vd_AJ5Yw

[3, ['https://m.youtube.com/c/MichaelReeves', 'https://youtube.com/user/VanossGaming', 'https://youtube.com/user/PewDiePi
e']]
[['Michael Reeves', 'VanossGaming', 'PewDiePie'], [4, 2, 2], [2017, 2012, 2010], [62, 102, 118], ['UCtHaxi4GTYDpJgMSGy7Ae
Sw', 'UCKqH_9mk1waLgBiL2vT5b9g', 'UC-lHJZR3Gqxm24_Vd_AJ5Yw']]
None
Press any key to continue . . .
```

```python
def generalInfoExtractor_SF(UIoutput):
    #Preallocating and unpacking
    numYoutubers = UIoutput[0]
    mainUserLinks = UIoutput[1]

    channelName = []
    genNum = []
    uploadYear = []
    numDataPoints = []
    organizedLink = []
    uniqueID = []
    passingArray = []
    passingArray2 = []

    GIEoutput = [0,0,0,0,0]

    for i in range (0,numYoutubers):
        #Concatenating oldest videos first and base URL
        organizedLink.append(i)
        organizedLink[i] = mainUserLinks[i] +'/videos?view=0&sort=da&flow=grid'

        #Adding more elements
        genNum.append(i)
        genNum[i] = 0
        numDataPoints.append(i)
        numDataPoints[i] = 0
        uploadYear.append(i)
        uploadYear[i] = 0
        channelName.append(i)
        channelName[i] = 0
        uniqueID.append(i)
        uniqueID[i] = 0
        passingArray.append(i)
        passingArray[i] = 0
```

```
        #Getting outputs
            #Generation number
        if (uploadYear[i] >= 2005) & (uploadYear[i] < 2010):
            genNum[i] = 1
        elif (uploadYear[i] >= 2010) & (uploadYear[i] < 2014):
            genNum[i] = 2
        elif (uploadYear[i] >= 2014) & (uploadYear[i] < 2017):
            genNum[i] = 3
        elif (uploadYear[i] >= 2017) & (uploadYear[i] < 2021):
            genNum[i] = 4
        else:
            print('wowee pog1gers')

        #Number of points needed
        numDataPoints[i] = int(round(30 + (((8 * (2021-uploadYear[i])) ))))

        #Unique ID information
            #Gets all link tags
        link = driver.find_elements_by_tag_name('link')
        allLinks = (len(link))

            #Putting all link tags into an array
        for j in range (0,allLinks):
            passingArray2.append(0)
            passingArray2[j] = link[j]

        relAttribute = re.compile('canonical')

            #Getting the actual code of the tags instead of reading selenium and returning which has the correct unique ID
        for k in range (0,allLinks):
            passingArray2[k] = passingArray2[k].get_attribute('outerHTML')
            canonicalString = relAttribute.search(passingArray2[k])
            if type(canonicalString) == re.Match:
                uniqueIDline = passingArray2[k]

            #Getting the unique ID
        compileuniqueID = re.compile('https://www.youtube.com/channel/')
        searchuniqueID = compileuniqueID.search(uniqueIDline)
        startuniqueID = searchuniqueID.end()
        lastuniqueID = int(len(uniqueIDline) -2)
        uniqueID[i] = uniqueIDline[startuniqueID:lastuniqueID:1]
```

```
    #Summary Statemnt:
    print('Channel: ' + channelName[i] + '\nGeneration Number: ' + str(genNum[i]) + '\nOPV Upload Year: ' + str(uploadYear[i]) + '\nData Points: ' + str(numDataPoints[i]) + '\nUnique ID: ' + str(uniqueID[i]) + '\n')

GIEoutput[0] = channelName
GIEoutput[1] = genNum
GIEoutput[2] = uploadYear
GIEoutput[3] = numDataPoints
GIEoutput[4] = uniqueID
return GIEoutput
```