### [Google spreadsheet interaction and futures contract](https://colab.research.google.com/drive/1LQ4gtxYMu0PAbRK_E48XVp5zgSYLDgjE)

구글 스프레드싯(google spreadsheet)을 사용할 수 있는 라이브러리들을 설치한다.

!pip install --upgrade -q gspread

!pip install gspread\_dataframe

Google spreadsheet을 사용할 수 있는 권한을 획득한다.

from gspread\_dataframe import get\_as\_dataframe, set\_with\_dataframe

from google.colab import auth

auth.authenticate\_user()

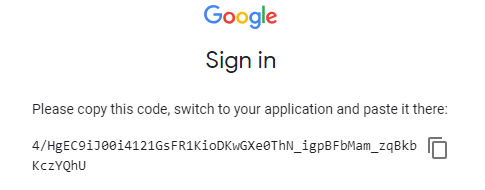
위를 실행하면 다음과 같은 메시지가 나오며 verification code를 요구한다.

Go to the following link in your browser:

<https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&prompt=select_account&response_type=code&client_id=32555940559.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive&access_type=offline>

Enter verification code:

위의 링크를 클릭하면 다음 페이지로 가서 verification code를 얻을 수 있다.



위에 있는 코드를 복사해서 아까 나온 “Enter verification code” 부분에 붙여 넣는다. 그리고 엔터키를 치면 된다. 그 다음 아래 코드를 실행해서 google spreadsheet을 사용할 수 있도록 하자.

import gspread

from oauth2client.client import GoogleCredentials

gc = gspread.authorize(GoogleCredentials.get\_application\_default())

다음 pandas와 numpy 라이브러리를 불러온다.

import pandas as pd

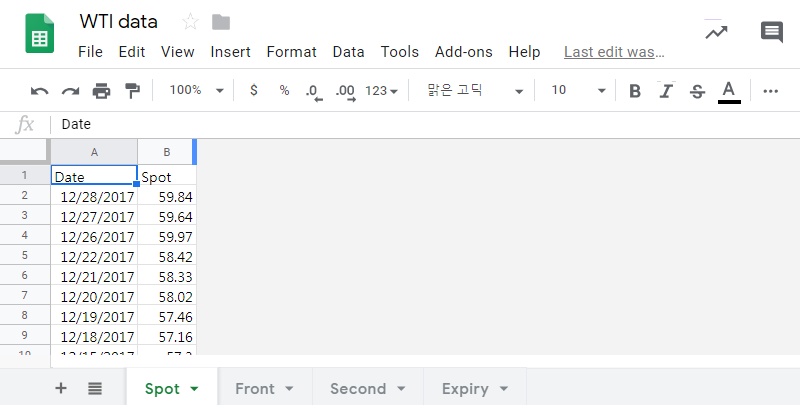
import numpy as np

다음은 WTI관련 데이터를 분석해보자. 본 분석에서는 WTI관련해서 Spot, Front, Second, Expiry 정보를 분석한다. Spot, Front, Second, Expiry는 각각 현물가격, 최근월물, 차근월물, 선물 만기일을 나타낸다. 최근월물은 선물거래 당시 만기가 가장 가까운 선물 거래 대상이다. 차근월물은 두번째로 가까운 선물 거래 대상이다.

이제 아래 링크에 필자가 올려놓은 WTI 관련 데이터를 가져온다.

<https://docs.google.com/spreadsheets/d/1OATdfY3mFXz9Y0urHF0mqTVcA0C_I2dlrkd-uKe2X1g>

위 google spreadsheet은 아래와 같이 생겼다.



위 그림에서 보듯 google spreadsheet에는 4개의 sheets가 있다: Spot, Front, Second, Expiry. 각 sheets의 데이터를 불러오되 각 sheets의 첫번째 열은 데이터가 아닌 행의 이름, 즉 index 설정해서 불러와보자. 다음을 실행해보자.

링크의 데이터를 불러와서 sheet1이란 이름으로 저장해보자.

sheet1 = gc.open\_by\_url('<https://docs.google.com/spreadsheets/d/1OATdfY3mFXz9Y0urHF0mqTVcA0C_I2dlrkd-uKe2X1g>')

이제는 sheet1이라는 이름으로 되어 있는 google spreadsheet의 데이터 중 Spot이라 되어있는 시트(sheet)의 데이터를 tmp1이라는 이름으로 불러와서 df1이라는 이름의 데이터프레임 형식으로 설정해보자.

tmp1 = sheet1.worksheet('Spot')

df1 = get\_as\_dataframe(tmp1)

df1데이터에서 첫번째 열(column)의 데이터는 각 행의 이름 즉 index로 설정한다.

df1 = df1.set\_index('Date')

두번째 sheet인 Front와 세번째 sheet인 Second에 있는 데이터도 비슷한 방식으로 불러와서 각 데이터의 이름을 df2, df3라고 정해주자.

tmp2 = sheet1.worksheet('Front')

df2 = get\_as\_dataframe(tmp2)

df2 = df2.set\_index('Date')

tmp3 = sheet1.worksheet('Second')

df3 = get\_as\_dataframe(tmp3)

df3 = df3.set\_index('Date')

데이터가 제대로 불러져 왔는지 df1, df2, df3의 처음 다섯행의 데이터를 살펴보기 위하여 다음을 실행하자.

print(df1.head())

print(df2.head())

print(df3.head())

아래 결과를 보면 데이터가 의도한데로 정리되어 출력되었다.

Spot

Date

12/28/2017 59.84

12/27/2017 59.64

12/26/2017 59.97

12/22/2017 58.42

12/21/2017 58.33

Front

Date

12/28/2017 59.84

12/27/2017 59.64

12/26/2017 59.97

12/22/2017 58.47

12/21/2017 58.36

Second

Date

12/28/2017 59.87

12/27/2017 59.69

12/26/2017 60.00

12/22/2017 58.54

12/21/2017 58.40

df = pd.concat([df1, df2, df3], axis=1)

print(df.tail())

Spot Front Second

Date

1/8/2010 82.75 82.75 83.30

1/7/2010 82.66 82.66 83.19

1/6/2010 83.18 83.18 83.75

1/5/2010 81.77 81.77 82.41

1/4/2010 81.51 81.51 82.12

df = df.fillna(method='ffill') # overwrite NaN with forward data

df = df.fillna(method='bfill') # overwrite NaN with backward data

print(df[10:15])

Spot Front Second

Date

12/13/2017 56.60 56.60 56.59

12/12/2017 57.14 57.14 57.16

12/11/2017 57.99 57.99 58.05

12/8/2017 57.36 57.36 57.44

12/7/2017 56.69 56.69 56.75

sheet1 = gc.open\_by\_url('https://docs.google.com/spreadsheets/d/1OATdfY3mFXz9Y0urHF0mqTVcA0C\_I2dlrkd-uKe2X1g')

tmp\_e = sheet1.worksheet('Expiry')

df\_e = get\_as\_dataframe(tmp\_e)

print(df\_e[120:125])

Ticker Month Last Trade Last Price

120 CLG10 10-Feb 01/20/10 77.62

121 CLH10 10-Mar 02/22/10 80.16

122 CLJ10 10-Apr 03/22/10 81.25

123 CLK10 10-May 04/20/10 83.45

124 CLM10 10-Jun 05/20/10 68.01

# 만기일 추출

df\_e['Last Trade'].head()

0 01/20/00

1 02/22/00

2 03/21/00

3 04/19/00

4 05/22/00

Name: Last Trade, dtype: object

# 만기일을 날짜 형식으로 변환

pd.to\_datetime(df\_e['Last Trade']).head()

0 2000-01-20

1 2000-02-22

2 2000-03-21

3 2000-04-19

4 2000-05-22

Name: Last Trade, dtype: datetime64[ns]

# 추출한 만기일 정보를 Series에 저장

expiry = pd.Series(pd.to\_datetime(df\_e['Last Trade']))

expiry.head()

0 2000-01-20

1 2000-02-22

2 2000-03-21

3 2000-04-19

4 2000-05-22

Name: Last Trade, dtype: datetime64[ns]

pd.to\_datetime(expiry.values)

DatetimeIndex(['2000-01-20', '2000-02-22', '2000-03-21', '2000-04-19',

'2000-05-22', '2000-06-20', '2000-07-20', '2000-08-22',

'2000-09-20', '2000-10-20',

...

'2023-08-22', '2023-09-20', '2023-10-20', '2023-11-20',

'2024-05-21', '2024-11-20', '2025-05-20', '2025-11-20',

'2026-05-19', '2026-11-20'],

dtype='datetime64[ns]', length=293, freq=None)

print(df.index)

Index(['12/28/2017', '12/27/2017', '12/26/2017', '12/22/2017', '12/21/2017',

'12/20/2017', '12/19/2017', '12/18/2017', '12/15/2017', '12/14/2017',

...

'1/15/2010', '1/14/2010', '1/13/2010', '1/12/2010', '1/11/2010',

'1/8/2010', '1/7/2010', '1/6/2010', '1/5/2010', '1/4/2010'],

dtype='object', name='Date', length=2014)

df['Expiry'] = np.nan # add Expiry column and fill blank with NaN

df.head()

Spot Front Second Expiry

Date

12/28/2017 59.84 59.84 59.87 NaN

12/27/2017 59.64 59.64 59.69 NaN

12/26/2017 59.97 59.97 60.00 NaN

12/22/2017 58.42 58.47 58.54 NaN

12/21/2017 58.33 58.36 58.40 NaN

# Trading day 중 만기일 찾아서 표시하기

for d in df.index: # read date values in df

if d in pd.to\_datetime(expiry.values): # if expiry has same value

df.loc[d].Expiry = 1 # put 1 at expiry row of df

print(df[10:15])

Spot Front Second Expiry

Date

12/13/2017 56.60 56.60 56.59 NaN

12/12/2017 57.14 57.14 57.16 NaN

12/11/2017 57.99 57.99 58.05 NaN

12/8/2017 57.36 57.36 57.44 NaN

12/7/2017 56.69 56.69 56.75 NaN

# 현물에 투자했다면

# 손익 = 현재가격 - 초기가격

price\_now = float(df.loc['12/28/2017','Spot'])

price\_before = float(df.loc['1/4/2010','Spot'])

profit = (price\_now - price\_before) \* 1000

print('손익 (현재가격 - 초기가격) : ', '{:,.2f}'.format(profit))

손익 (현재가격 - 초기가격) : -21,670.00

# 선물에 투자했다면

# 현재가격 - 초기가격 - 롤오버 비용

roll\_over\_cost\_acc = 0

for d in df.index:

if (df.loc[d].Expiry == 1):

# 롤오버 비용 = 차근월물 가격 - 최근월물 가격

roll\_over\_cost = df.loc[d].Second - df.loc[d].Front

# print(df.loc[d].Second, df.loc[d].Front, roll\_over\_cost)

roll\_over\_cost\_acc = roll\_over\_cost\_acc + roll\_over\_cost

print('누적 롤오버 비용 :', '{:,.2f}'.format(roll\_over\_cost\_acc))

누적 롤오버 비용 : 40.68

price\_now = float(df.loc['12/28/2017','Front'])

price\_before = float(df.loc['1/4/2010','Front'])

profit = (price\_now - price\_before - roll\_over\_cost\_acc) \* 1000

print('누적 손익', '{:,.2f}'.format(profit))

누적 손익 -62,350.00

END