KOREA 2021

DISCOVERY SUMMIT

EXPLORING DATA
INSPIRING INNOVATION

API를 활용한 JMP 분석 사례

2021. NOV

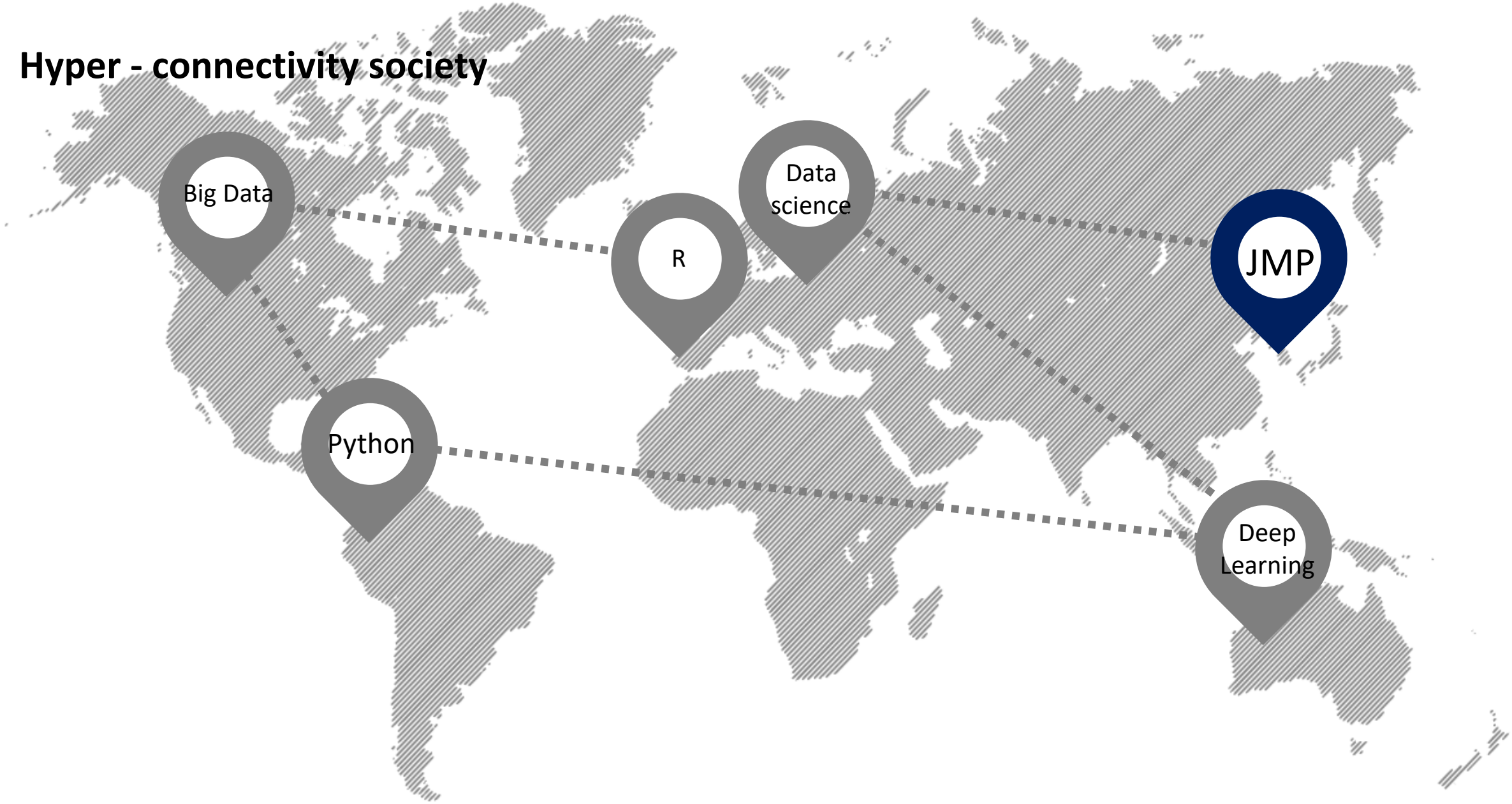
정순우
옥창수
김광희

Contents

1. Data Trend
2. Python Script
3. JMP Scripting using python code

Chapter 1 Data Trend

Hyper - connectivity society



Integration & Connection

Manufacturing site

- Product in/output data
- Volume by time

Web site(Yahoo Finance, etc.)

- Exchange rate
- S&P 500
- Crude oil

Data loading
(Long & Hard)

Free & Open Source program

- Wide variety of usage
- Enormous data source(web/API)

Portable language

- Flexible scalability(Install library)
- Easy to code

Data loading
(Fast & Simple)

Powerful Visualization

- Intutive graph
- Easy & Remarkble plotting

Data edit and analysis

- Sharing script w/ Python
- Easy loading other data

Compatible

Strengthen point?

Data Access

Input to Excel
- Take too many time
- Typo error risk
- Data format arrange

Extremely hard job

Input to JMP

Display by JMP

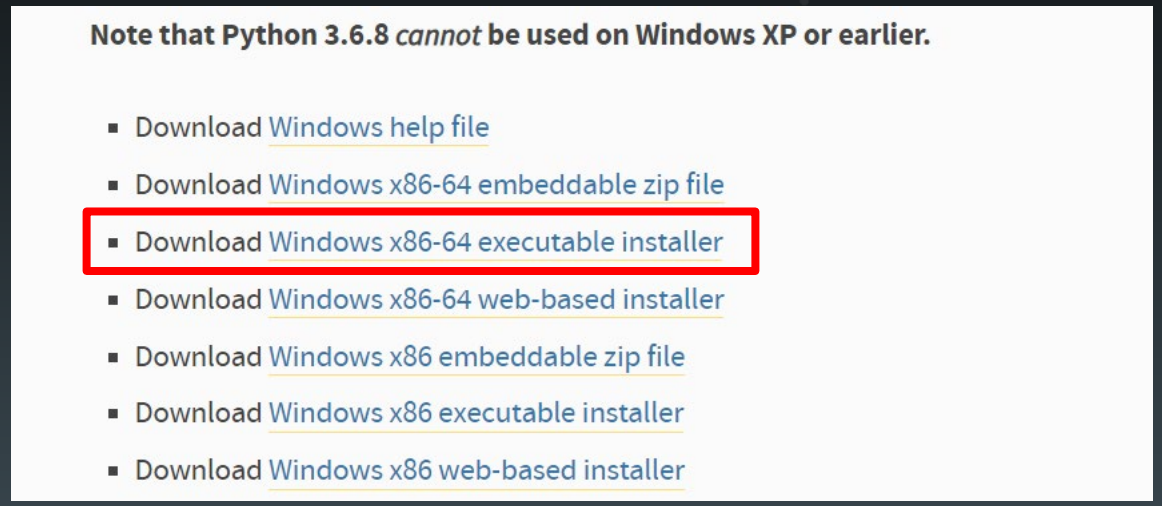
DAYS SEC.

Data Access

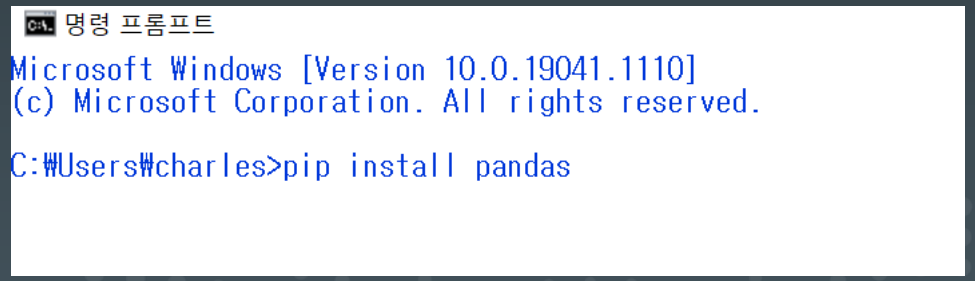
JMP
(Coding & Display)

Chapter 2 Python Script

Python Download & Library Install



Python 3.6



Python Library

Script with Python Init command (Pandas Data Reader)

```
Python Init();

Python Submit("\[



]\");

Python Term();
```

Python Init

```
Python Init();

Python Submit("\[
import pandas as pd
import pandas_datareader as pdr
from pandas_datareader import data
import pandas_datareader.data as web
from datetime import datetime


#df = data.DataReader("^KS11", "yahoo").reset_index()

start_date = datetime(2000,1,1)
end_date = pd.Timestamp.today()
df = data.get_data_yahoo("^KS11", start_date, end_date).reset_index()

]\");

df = Python Get( df );

Python Term();

df <<New Data View;
```

Kospi Data Loading

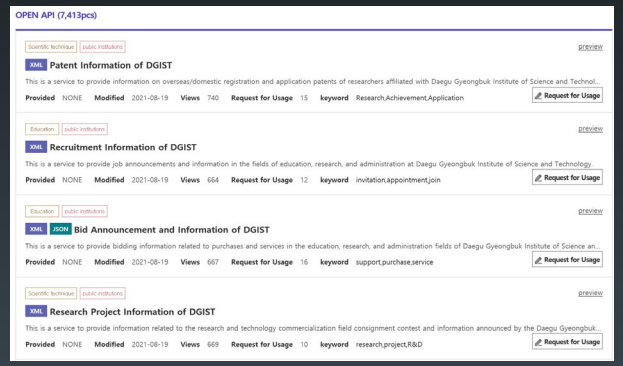
	High	Low	Open	Close	Volume	Adj Close
1	1066.1800537	1016.5900269	1028.3299561	1059.0400391	195900	1059.0400391
2	1026.5200195	984.04998779	1006.8699951	986.30999756	257700	986.30999756
3	1014.9000244	953.5	1013.9500122	960.78997803	203500	960.78997803
4	970.15997314	930.84002686	949.16998291	948.65002441	215700	948.65002441
5	994.94000244	974.82000732	979.66998291	987.23999023	240200	987.23999023
6	1005.8699951	981.22998047	992.16998291	981.33001709	257100	981.33001709
7	968.67999268	949.20001221	957.97998047	955.01000977	227100	955.01000977
8	960.72998047	939.25	955.05999756	951.04998779	223000	951.04998779
9	970.59002686	937.75	958.82000732	948.0300293	225000	948.0300293
10	986.09002686	959.35998535	962.66998291	983.27001953	212100	983.27001953
11	992.84997559	967.96002197	992.30999756	981.5300293	212100	981.5300293
12	969.30999756	938.73999023	969.30999756	938.7800293	235700	938.7800293
13	945.90002441	906.26000977	927.11999512	945.90002441	226700	945.90002441
14	940.70001221	911.83001709	928.54998779	925.15997314	255100	925.15997314
15	944.30999756	915.70001221	917.92999268	926.77001953	210400	926.77001953
16	913.19000244	891.2199707	909.11999512	891.2199707	242600	891.2199707
17	902.42999268	875.82000732	896.34002686	885.53997803	231100	885.53997803

Kospi Data

	Date	High	Low	Open	Close	Volume	Adj Close
1	2000-01-04	1066.1800537	1016.5900269	1028.3299561	1059.0400391	195900	1059.0400391
2	2000-01-05	1026.5200195	984.04998779	1006.8699951	986.30999756	257700	986.30999756
3	2000-01-06	1014.9000244	953.5	1013.9500122	960.78997803	203500	960.78997803
4	2000-01-07	970.15997314	930.84002686	949.16998291	948.65002441	215700	948.65002441
5	2000-01-10	994.94000244	974.82000732	979.66998291	987.23999023	240200	987.23999023
6	2000-01-11	1005.8699951	981.22998047	992.16998291	981.33001709	257100	981.33001709
7	2000-01-12	968.67999268	949.20001221	957.97998047	955.01000977	227100	955.01000977
8	2000-01-13	960.72998047	939.25	955.05999756	951.04998779	223000	951.04998779
9	2000-01-14	970.59002686	937.75	958.82000732	948.0300293	225000	948.0300293
10	2000-01-17	986.09002686	959.35998535	962.66998291	983.27001953	212100	983.27001953
11	2000-01-18	992.84997559	967.96002197	992.30999756	981.5300293	212100	981.5300293
12	2000-01-19	969.30999756	938.73999023	969.30999756	938.7800293	235700	938.7800293
13	2000-01-20	945.90002441	906.26000977	927.11999512	945.90002441	226700	945.90002441
14	2000-01-21	940.70001221	911.83001709	928.54998779	925.15997314	255100	925.15997314
15	2000-01-24	944.30999756	915.70001221	917.92999268	926.77001953	210400	926.77001953
16	2000-01-25	913.19000244	891.2199707	909.11999512	891.2199707	242600	891.2199707
17	2000-01-26	902.42999268	875.82000732	896.34002686	885.53997803	231100	885.53997803

Script with Python Init command (Republic Data Potal API)

Find API → **Request (Service Key)** → **Coding**



Republic Data Potal


```
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL SERVICE.</resultMsg>
  </header>
  <body>
    <items>
      <item>
        <accDefRate>1.832909316</accDefRate>
        <accExamCnt>12057825</accExamCnt>
        <accExamCompCnt>11590317</accExamCompCnt>
        <careCnt>24123</careCnt>
        <clearCnt>186192</clearCnt>
        <createDt>2021-08-09 00:00:00.000</createDt>
        <deathCnt>2125</deathCnt>
        <decideCnt>212440</decideCnt>
        <examCnt>467508</examCnt>
        <resutlNegCnt>11377877</resutlNegCnt>
        <seq>568</seq>
        <stateDt>20210809</stateDt>
        <stateTime>00:00</stateTime>
        <updateDt>2021-10-07 10:30:51.51</updateDt>
      </item>
      <item>
        <accDefRate>1.824635402</accDefRate>
        <accExamCnt>12027438</accExamCnt>
        <accExamCompCnt>11561159</accExamCompCnt>
        <careCnt>23918</careCnt>
        <clearCnt>184910</clearCnt>
        <createDt>2021-08-08 00:00:00.000</createDt>
        <deathCnt>2121</deathCnt>
        <decideCnt>210949</decideCnt>
        <examCnt>466279</examCnt>
        <resutlNegCnt>11350210</resutlNegCnt>
        <seq>567</seq>
        <stateDt>20210808</stateDt>
        <stateTime>00:00</stateTime>
        <updateDt>2021-10-07 10:30:51.51</updateDt>
      </item>
```


XML

```
Python Init();

Python Submit("\[

import requests
from urllib.request import Request, urlopen
from xml.etree import ElementTree
import pandas as pd

url="http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19InfStateJson?serviceKey=nch%2BcLEbjhkM

Params = 'pageNo=' + '1' + '&numOfRows='+'10' + '&startCreateDt='+'20200304' + '&endCreateDt='+'20210809'

url = url + Params

request = Request(url)

request.get_method = lambda:'GET'

response_body = urlopen(request).read()

root = ElementTree.fromstring(response_body)


df = pd.DataFrame()

for item in root.iter('item'):

    item_dict = {}

    item_dict['seq'] = item.find('seq').text # 게시글 번호

    item_dict['stateDt'] = item.find('stateDt').text # 기준일

    item_dict['stateTime'] = item.find('stateTime').text # 기준 시간

    item_dict['decideCnt'] = item.find('decideCnt').text # 확진자 수

    item_dict['clearCnt'] = item.find('clearCnt').text # 격리해제 수

    item_dict['examCnt'] = item.find('examCnt').text # 검사 진행 수

    item_dict['deathCnt'] = item.find('deathCnt').text # 사망자 수
```

Python Coding

Chapter 3

JMP scripting using python code

Example : Covid 19

Script

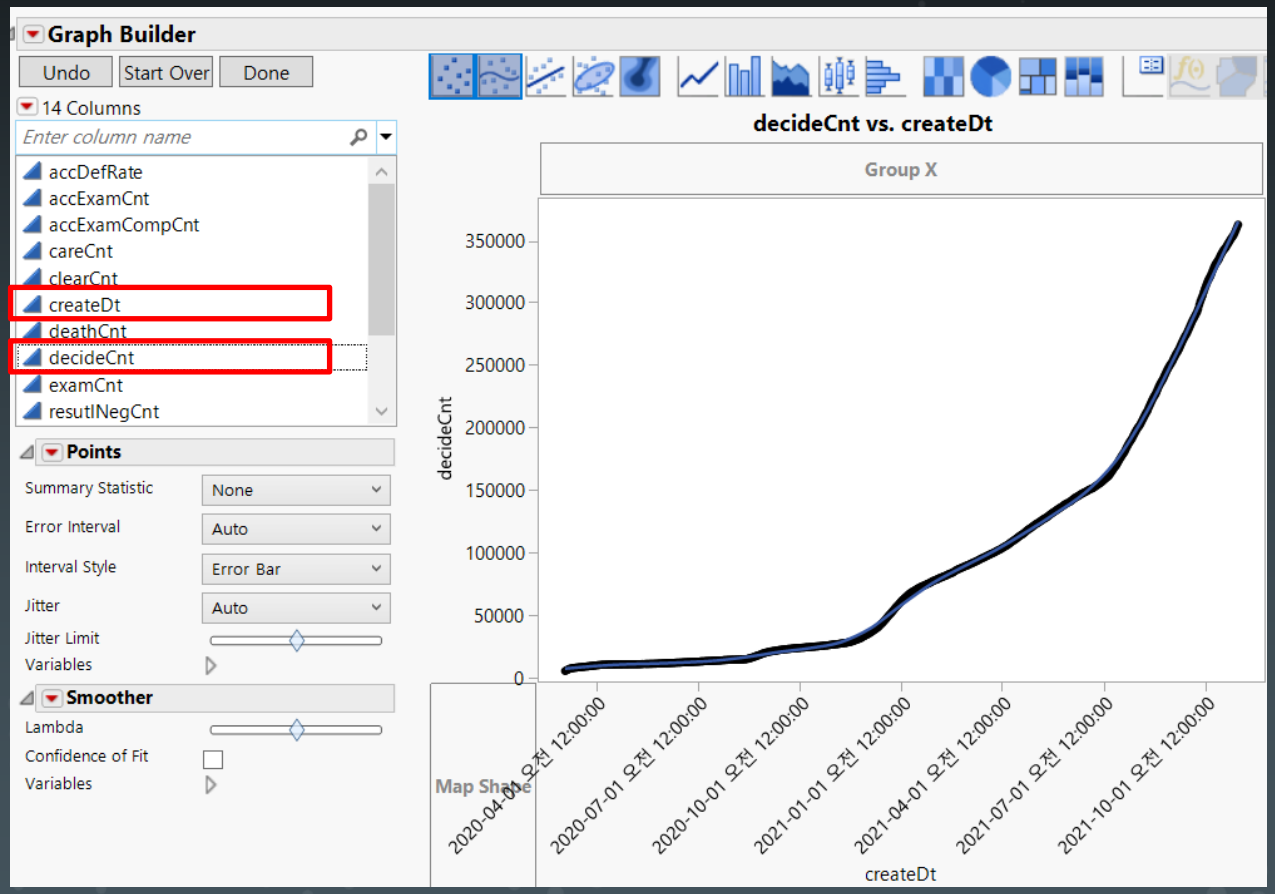
```
Python Init();

Python Submit("\[

import requests
from urllib.request import Request, urlopen
from xml.etree import ElementTree
import pandas as pd

url="http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19InfStateJson?serviceKey=nch%2BcLI

Params = 'pageNo=' + '1' + '&numOfRows='+'10' + '&startCreateDt='+'20200304' + '&endCreateDt='+'202108(

url = url + Params

request = Request(url)
request.get_method = lambda:'GET'
response_body = urlopen(request).read()
root = ElementTree.fromstring(response_body)

covid = pd.DataFrame()
for item in root.iter('item'):
    item_dict = {}
    item_dict['seq'] = item.find('seq').text # 게시글 번호
    item_dict['stateDt'] = item.find('stateDt').text # 기준일
    item_dict['stateTime'] = item.find('stateTime').text # 기준 시간
    item_dict['decideCnt'] = item.find('decideCnt').text # 확진자 수
    item_dict['clearCnt'] = item.find('clearCnt').text # 격리해제 수
    item_dict['examCnt'] = item.find('examCnt').text # 검사 진행 수
    item_dict['deathCnt'] = item.find('deathCnt').text # 사망자 수
    item_dict['careCnt'] = item.find('careCnt').text # 치료중 환자 수
    item_dict['resutlNegCnt'] = item.find('resutlNegCnt').text # 결과 음성 수
    item_dict['accExamCnt'] = item.find('accExamCnt').text # 누적 검사 수
    item_dict['accExamCompCnt'] = item.find('accExamCompCnt').text # 누적 검사 완료 수
    item_dict['accDefRate'] = item.find('accDefRate').text # 누적 환진률
    item_dict['createDt'] = item.find('createDt').text # 등록 일시분초
    item_dict['updateDt'] = item.find('updateDt').text # 수정 일시분초
    covid = covid.append(item_dict, ignore_index = True)

covid
]\");

covid = Python Get( covid );

Python Term();
```

Data Table

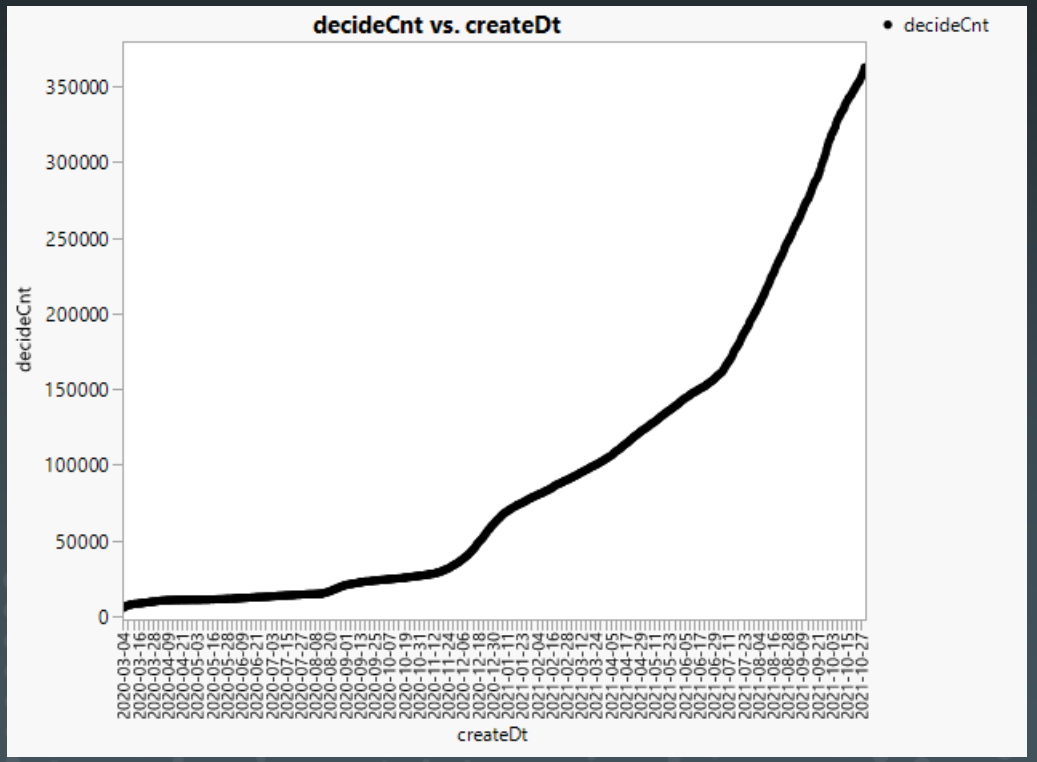
	accDefRate	accExamCnt	accExamCompCnt	careCnt	clearCnt	createDt	deathCnt	decideCnt	examCnt	resutlNegCnt	seq	stateDt	stateTime	updateDt
1	1.8329008458	12057824	11590316	24088	186226	2021-08-09 오전 ...	2125	212439	467508	11377877	568	20210809	00:00	2021-10-28 오후 ..
2	1.8246269102	12027437	11561158	23840	184987	2021-08-08 오전 ...	2121	210948	466279	11350210	567	20210808	00:00	2021-10-28 오후 ..
3	1.811282604	11995923	11550986	23435	183670	2021-08-07 오전 ...	2116	209221	444937	11341765	566	20210807	00:00	2021-10-07 오전 ..
4	1.799611891	11951646	11524596	22828	182457	2021-08-06 오전 ...	2113	207398	427050	11317198	565	20210806	00:00	2021-10-07 오전 ..
5	1.789742069	11908430	11492941	22863	180722	2021-08-05 오전 ...	2109	205694	415489	11287247	564	20210805	00:00	2021-10-07 오전 ..
6	1.77841233	11864240	11466351	22425	179388	2021-08-04 오전 ...	2106	203919	397889	11262432	563	20210804	00:00	2021-10-07 오전 ..
7	1.765622622	11820011	11451711	22300	177790	2021-08-03 오전 ...	2104	202194	368300	11249517	562	20210803	00:00	2021-10-07 오전 ..
8	1.765361674	11772599	11385429	22323	176572	2021-08-02 오전 ...	2099	200994	387170	11184435	561	20210802	00:00	2021-10-07 오전 ..
9	1.756553576	11751780	11373180	22411	175267	2021-08-01 오전 ...	2098	199776	378600	11173404	560	20210801	00:00	2021-10-07 오전 ..
10	1.745957213	11728815	11359614	21898	174341	2021-07-31 오전 ...	2095	198334	369201	11161280	559	20210731	00:00	2021-10-07 오전 ..
11	1.737598412	11682962	11325747	21860	172847	2021-07-30 오전 ...	2089	196796	357215	11128951	558	20210730	00:00	2021-10-07 오전 ..
12	1.727162415	11637500	11295174	21573	171428	2021-07-29 오전 ...	2085	195086	342326	11100088	557	20210729	00:00	2021-10-07 오전 ..
13	1.717447696	11585607	11261653	21100	170230	2021-07-28 오전 ...	2083	193413	323954	11068240	556	20210728	00:00	2021-10-07 오전 ..
14	1.707057587	11528602	11219188	20273	169166	2021-07-27 오전 ...	2079	191518	309414	11027670	555	20210727	00:00	2021-10-07 오전 ..
15	1.699126896	11472339	11191336	20776	167302	2021-07-26 오전 ...	2077	190155	281003	11001181	554	20210726	00:00	2021-10-07 오전 ..
16	1.689275533	11453340	11178579	20725	166039	2021-07-25 오전 ...	2073	188837	274761	10989742	553	20210725	00:00	2021-10-07 오전 ..
17	1.67939231	11430518	11155821	19335	165947	2021-07-24 오전 ...	2068	187350	274697	10968471	552	20210724	00:00	2021-10-07 오전 ..
18	1.671630834	11388299	11110288	18838	164819	2021-07-23 오전 ...	2066	185723	278011	10924565	551	20210723	00:00	2021-10-07 오전 ..
19	1.662335814	11343913	11074417	18251	163780	2021-07-22 오전 ...	2063	184094	269496	10890323	550	20210722	00:00	2021-10-07 오전 ..
20	1.649873688	11298669	11046482	17546	162647	2021-07-21 오전 ...	2060	182253	252187	10864229	549	20210721	00:00	2021-10-07 오전 ..
21	1.639564831	11251982	11007311	17206	161207	2021-07-20 오전 ...	2059	180472	244671	10826839	548	20210720	00:00	2021-10-07 오전 ..
22	1.634722225	11202429	10961740	17216	159920	2021-07-19 오전 ...	2058	179194	240689	10782546	547	20210719	00:00	2021-10-07 오전 ..
23	1.630927675	11175977	10910539	16683	159203	2021-07-18 오전 ...	2057	177943	265438	10732596	546	20210718	00:00	2021-10-07 오전 ..
24	1.615973942	11149223	10921587	15907	158528	2021-07-17 오전 ...	2055	176490	227636	10745097	545	20210717	00:00	2021-10-07 오전 ..
25	1.60729781	11108582	10890390	15457	157533	2021-07-16 오전 ...	2051	175041	218192	10715349	544	20210716	00:00	2021-10-07 오전 ..
26	1.598253496	11060454	10855975	14770	156686	2021-07-15 오전 ...	2050	173506	204479	10682469	543	20210715	00:00	2021-10-07 오전 ..
27	1.587769115	11008857	10826952	13775	156084	2021-07-14 오전 ...	2048	171907	181905	10655045	542	20210714	00:00	2021-10-07 오전 ..
28	1.578964888	10964297	10785167	13183	155065	2021-07-13 오전 ...	2046	170294	179130	10614873	541	20210713	00:00	2021-10-07 오전 ..
29	1.574131766	10919896	10745225	12775	154325	2021-07-12 오전 ...	2044	169144	174671	10576081	540	20210712	00:00	2021-10-07 오전 ..
30	1.566666486	10902276	10726214	12241	153760	2021-07-11 오전 ...	2043	168044	176062	10558170	539	20210711	00:00	2021-10-07 오전 ..

Example : Covid 19

Graph builder (X axis : creatDt , Y axis : decideCnt)

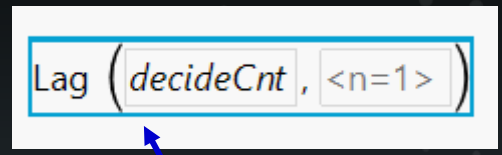


Data type change of X axis : creatDt
(continuous → Nominal)

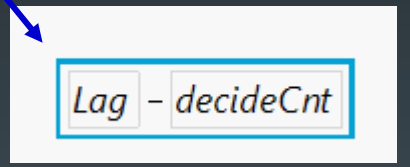


Example : Covid 19

Daily decide using formula

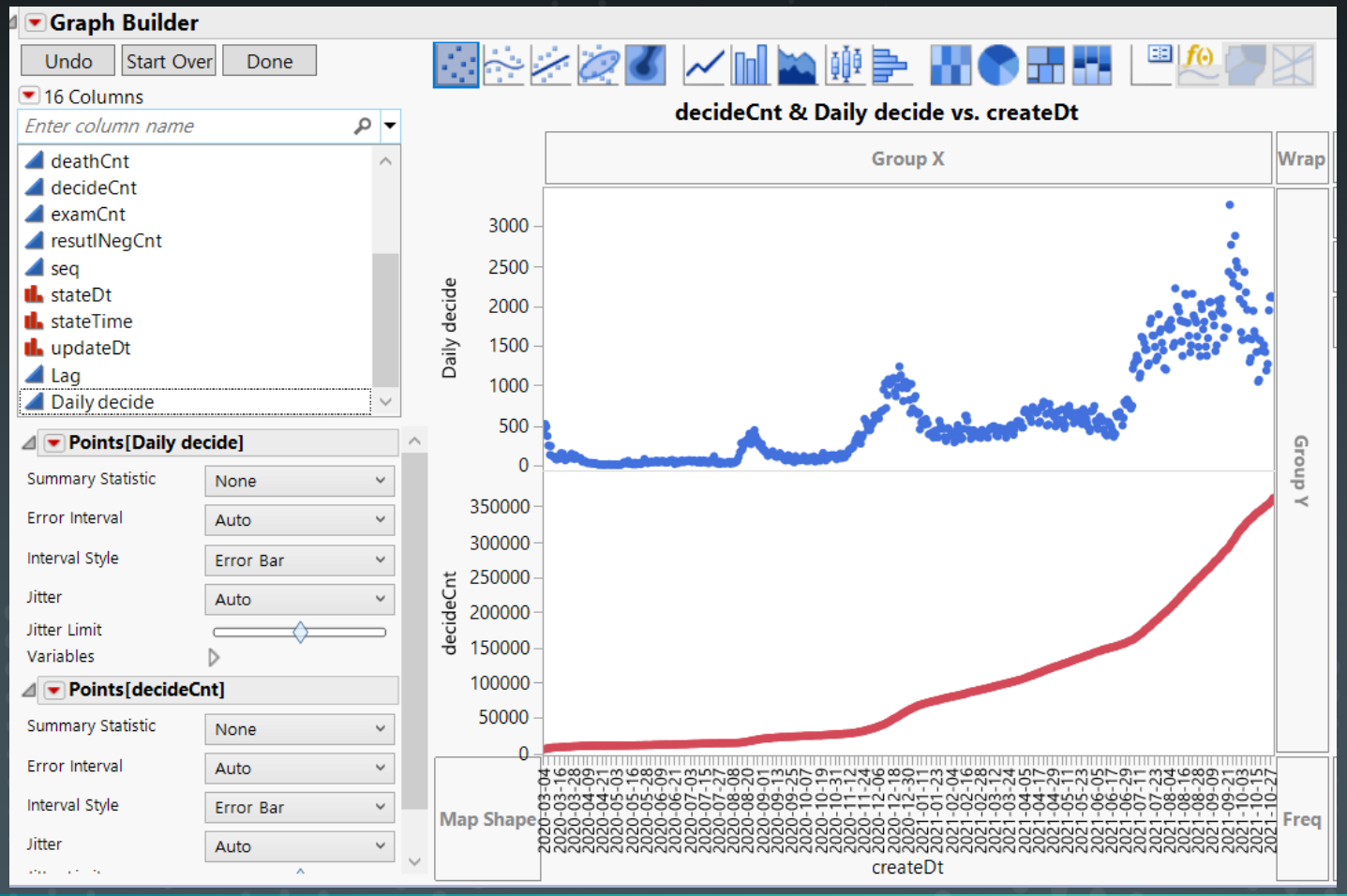


	accDefRate	accExamCnt	accExamCompCnt	careCnt	clearCnt	createDt	deathCnt	decideCnt	examCnt	resutlNegCnt	seq	stateDt	stateTime	updateDt	Lag	Daily decide
1	2.5167069578	15775700	14409266	25228	334581	2021-10-30	2830	362639	1366434	14046627	682	20211030	00:00	null		•
2	2.5085768649	15730785	14372093	24723	332995	2021-10-29	2817	360535	1358692	14011558	681	20211029	00:00	2021-10-30 09:43...	362639	2104
3	2.5082902607	15678186	14289056	24750	330853	2021-10-28	2808	358411	1389130	13930645	680	20211028	00:00	2021-10-30 09:43...	360535	2124
4	2.5019099621	15628306	14241120	23845	329658	2021-10-27	2797	356300	1387186	13884820	679	20211027	00:00	2021-10-30 09:44...	358411	2111
5	2.4932471843	15579430	14212590	23975	327592	2021-10-26	2788	354355	1366840	13858235	677	20211026	00:00	2021-10-26 19:11...	356300	1945
6	2.4931384199	15530472	14162190	25862	324448	2021-10-25	2773	353083	1368282	13809107	676	20211025	00:00	2021-10-30 09:44...	354355	1272
7	2.4867673642	15504979	14150620	25734	323393	2021-10-24	2766	351893	1354359	13798727	675	20211024	00:00	2021-10-30 09:45...	353083	1190
8	2.4782031513	15481469	14142182	25191	322536	2021-10-23	2745	350472	1339287	13791710	674	20211023	00:00	2021-10-28 09:30...	351893	1421
9	2.4731308076	15437382	14110212	25922	320317	2021-10-22	2725	348964	1327170	13761248	673	20211022	00:00	2021-10-28 09:30...	350472	1508
10	2.4721894162	15392004	14057418	27062	317755	2021-10-21	2709	347526	1334586	13709892	672	20211021	00:00	2021-10-28 09:30...	348964	1438
11	2.4685802309	15344729	14019597	26780	316607	2021-10-20	2698	346085	1325132	13673512	671	20211020	00:00	2021-10-28 09:30...	347526	1441
12	2.4655589015	15296830	13973059	28393	313432	2021-10-19	2689	344514	1323771	13628545	670	20211019	00:00	2021-10-28 09:30...	346085	1571
13	2.466224513	15246346	13925780	28992	311781	2021-10-18	2668	343441	1320566	13582339	669	20211018	00:00	2021-10-28 09:29...	344514	1073
14	2.4617514369	15220864	13908431	29387	310344	2021-10-17	2660	342391	1312433	13566040	668	20211017	00:00	2021-10-28 09:29...	343441	1050
15	2.4521548219	15198187	13904954	30140	308187	2021-10-16	2644	340971	1293233	13563983	667	20211016	00:00	2021-10-28 09:29...	342391	1420
16	2.4513858745	15152521	13843353	30877	305851	2021-10-15	2626	339354	1309168	13503999	666	20211015	00:00	2021-10-28 19:13...	340971	1617
17	2.4451502864	15106360	13809826	31334	303719	2021-10-14	2618	337671	1296534	13472155	665	20211014	00:00	2021-10-28 09:28...	339354	1683
18	2.436982744	15062412	13776544	31061	302066	2021-10-13	2605	335732	1285868	13440812	664	20211013	00:00	2021-10-28 09:28...	337671	1939
19	2.4335792555	15007572	13730763	32295	299260	2021-10-12	2594	334149	1276809	13396614	663	20211012	00:00	2021-10-28 09:28...	335732	1583
20	2.4253840387	14977407	13721662	32198	298022	2021-10-11	2583	332803	1255745	13388859	662	20211011	00:00	2021-10-28 09:28...	334149	1346
21	2.4215310314	14951646	13689934	32223	296708	2021-10-10	2575	331506	1261712	13358428	661	20211010	00:00	2021-10-28 09:27...	332803	1297
22	2.4118597481	14925863	13678739	32423	294929	2021-10-09	2560	329912	1247124	13348827	660	20211009	00:00	2021-10-28 09:27...	331506	1594
23	2.4027481497	14882186	13649329	33314	292091	2021-10-08	2554	327959	1232857	13321370	659	20211008	00:00	2021-10-28 09:27...	329912	1953
24	2.3965178152	14833965	13594224	34422	288822	2021-10-07	2544	325788	1239741	13268436	658	20211007	00:00	2021-10-28 09:27...	327959	2171
25	2.384482953	14780938	13561137	33787	287040	2021-10-06	2536	323363	1219801	13237774	657	20211006	00:00	2021-10-28 09:26...	325788	2425
26	2.3797170401	14731627	13503118	34615	284197	2021-10-05	2524	321336	1228509	13181782	656	20211005	00:00	2021-10-28 09:26...	323363	2027
27	2.3712389557	14701211	13485018	34580	282669	2021-10-04	2513	319762	1216193	13165256	655	20211004	00:00	2021-10-28 09:26...	321336	1574



Example : Covid 19

Graph builder (X axis : creatDt , Y axis : Daily decide, Y axis : decideCnt)



Example : Covid 19

Script

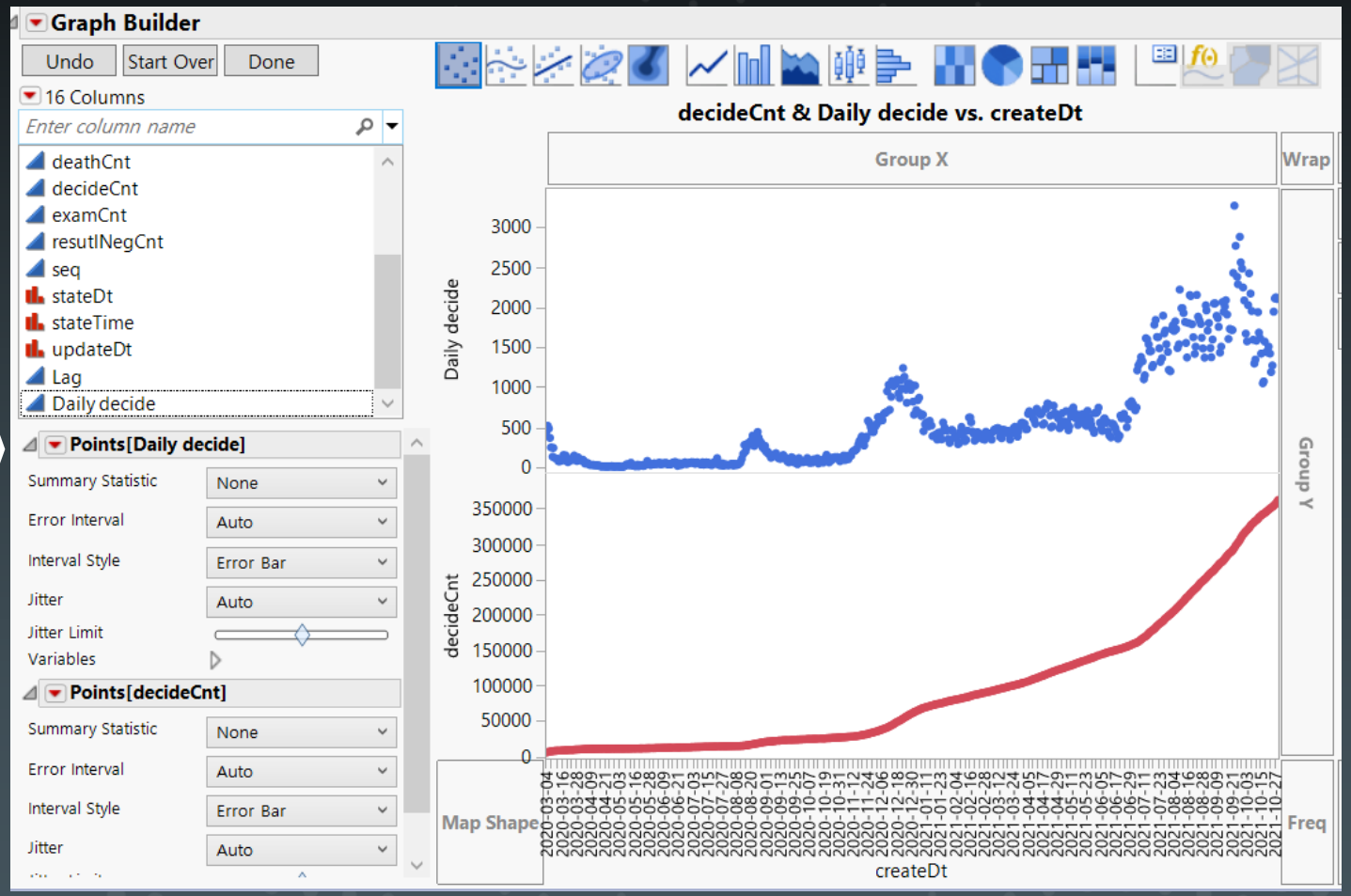
Script_코로나 Data Daily - JMP

File Edit Tables DOE Analyze Graph Tools View Window Help

(no tables)

```
Python Init();
Python Submit("\[
import requests
from urllib.request import Request, urlopen
from xml.etree import ElementTree
import pandas as pd
url="http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19InfStateJson?serviceKey=n
Params = 'pageNo=' + '1' + '&numOfRows='+'10' + '&startCreateDt='+'20200304' + '&endCreateDt='+
url = url + Params
request = Request(url)
request.get_method = lambda:'GET'
response_body = urlopen(request).read()
root = ElementTree.fromstring(response_body)
covid = pd.DataFrame()
for item in root.iter('item'):
    item_dict = {}
    item_dict['seq'] = item.find('seq').text # 게시글 번호
    item_dict['stateDt'] = item.find('stateDt').text # 기준일
    item_dict['stateTime'] = item.find('stateTime').text # 기준 시간
    item_dict['decideCnt'] = item.find('decideCnt').text # 확진자 수
    item_dict['clearCnt'] = item.find('clearCnt').text # 격리해제 수
    item_dict['examCnt'] = item.find('examCnt').text # 검사 진행 수
    item_dict['deathCnt'] = item.find('deathCnt').text # 사망자 수
    item_dict['careCnt'] = item.find('careCnt').text # 치료중 환자 수
    item_dict['resutlNegCnt'] = item.find('resutlNegCnt').text # 결과 음성 수
    item_dict['accExamCnt'] = item.find('accExamCnt').text # 누적 검사 수
    item_dict['accExamCompCnt'] = item.find('accExamCompCnt').text # 누적 검사 완료 수
    item_dict['accDefRate'] = item.find('accDefRate').text # 누적 확진률
    item_dict['createDt'] = item.find('createDt').text # 등록 일시분초
    item_dict['updateDt'] = item.find('updateDt').text # 수정 일시분초
    covid = covid.append(item_dict, ignore_index = True)
covid ]\");
covid = Python Get( covid );
Python Term();
covid <<New Data View;
Data table("covid"): createDt << Set Modeling Type( "Nominal" );
Data Table( "covid" ):createDt << Format( "yyyy-mm-dd", 25 );
col = New Column("Lag","Numeric","Continuos",Width(5));
col << Set Formula(:Lag(decideCnt));
col2 = New Column("Daily decide","Numeric","Continuos", widht(5));
col2 << Set Formula(:Lag - :decideCnt );
Graph Builder(
    Size( 1137, 703 ),
    Variables( X( :createDt ), Y( :Daily decide ), Y( :decideCnt ) ),
    Elements( Position( 1, 1 ), Points( X, Y, Legend( 4 ) ) ),
    Elements( Position( 1, 2 ), Points( X, Y, Legend( 5 ) ) )
);
```

Graph



Example : kospi vs S&P500

Script of kospi vs S&P500 using python code

1. Kospi - JMP

File Edit Tables DOE Analyze Graph Tools View Window Help

(no tables)

```
1  Python Init();
2
3  Python Submit("\[
4  import pandas as pd
5  import pandas_datareader as pdr
6  from pandas_datareader import data
7  import pandas_datareader.data as web
8  from datetime import datetime
9
10
11 #df = data.DataReader("^KS11", "yahoo").reset_inc
12
13 start_date = datetime(2000,1,1)
14 end_date = pd.Timestamp.today()
15 kospi = data.get_data_yahoo("^KS11", start_date,
16
17 ]\");
18
19 kospi = Python Get( kospi );
20
21 Python Term();
22
23 kospi <<New Data View;
24
```

2. SnP500_Script - JMP

File Edit Tables DOE Analyze Graph Tools View Window Help

(no tables)

```
1  Python Init();
2
3  Python Submit("\[
4  import pandas as pd
5  import pandas_datareader as pdr
6  from pandas_datareader import data
7  import pandas_datareader.data as web
8  from datetime import datetime
9
10 start_date = datetime(2000,1,1)
11 end_date = pd.Timestamp.today()
12 df = data.get_data_yahoo("^GSPC", start_date,
13
14 ]\");
15
16 df = Python Get( df );
17
18 Python Term();
19
20 df <<New Data View;
21
```

kospi

Date	High	Low	Open	Close	Volume	Adj Close
2000-01-04	1066.1800537	1016.5900269	1028.3299561	1059.0400391	195900	1059.0400391
2000-01-05	1026.5200195	984.04998779	1006.8699951	986.30999756	257700	986.30999756
2000-01-06	1014.9000244	953.5	1013.9500122	960.78997803	203500	960.78997803
2000-01-07	970.15997314	930.84002686	949.16998291	948.65002441	215700	948.65002441
2000-01-10	994.94000244	974.82000732	979.66998291	987.23999023	240200	987.23999023
2000-01-11	1005.8699951	981.22998047	992.16998291	981.33001709	257100	981.33001709
2000-01-12	968.67999268	949.20001221	957.97998047	955.01000977	227100	955.01000977
2000-01-13	960.72998047	939.25	955.05999756	951.04998779	223000	951.04998779
2000-01-14	970.59002686	937.75	958.82000732	948.0300293	225000	948.0300293
2000-01-17	986.09002686	959.35998535	962.66998291	983.27001953	212100	983.27001953
2000-01-18	992.84997559	967.96002197	992.30999756	981.5300293	212100	981.5300293
2000-01-19	969.30999756	938.73999023	969.30999756	938.7800293	235700	938.7800293
2000-01-20	945.90002441	906.26000977	927.11999512	945.90002441	226700	945.90002441
2000-01-21	940.70001221	911.83001709	928.54998779	925.15997314	255100	925.15997314
2000-01-24	944.30999756	915.70001221	917.92999268	926.77001953	210400	926.77001953
2000-01-25	913.19000244	891.2199707	909.11999512	891.2199707	242600	891.2199707
2000-01-26	902.42999268	875.82000732	896.34002686	885.53997803	231100	885.53997803
2000-01-27	912.48999023	879	888.27001953	909.22998047	344900	909.22998047
2000-01-28	943.95001221	916.85998535	922.73999023	941.66998291	293100	941.66998291

S&P500

Date	High	Low	Open	Close	Volume	Adj Close
1999-12-31	1472.4200439	1458.1899414	1464.4699707	1469.25	374050000	1469.25
2000-01-03	1478	1438.3599854	1469.25	1455.2199707	931800000	1455.2199707
2000-01-04	1455.2199707	1397.4300537	1455.2199707	1399.4200439	1009000000	1399.4200439
2000-01-05	1413.2700195	1377.6800537	1399.4200439	1402.1099854	1085500000	1402.1099854
2000-01-06	1411.9000244	1392.0999756	1402.1099854	1403.4499512	1092300000	1403.4499512
2000-01-07	1441.4699707	1400.7299805	1403.4499512	1441.4699707	1225200000	1441.4699707
2000-01-10	1464.3599854	1441.4699707	1441.4699707	1457.5999756	1064800000	1457.5999756
2000-01-11	1458.6600342	1434.4200439	1457.5999756	1438.5600586	1014000000	1438.5600586
2000-01-12	1442.5999756	1427.0799561	1438.5600586	1432.25	974600000	1432.25
2000-01-13	1454.1999512	1432.25	1432.25	1449.6800537	1030400000	1449.6800537
2000-01-14	1473	1449.6800537	1449.6800537	1465.1500244	1085900000	1465.1500244
2000-01-18	1465.1500244	1451.3000488	1465.1500244	1455.1400146	1056700000	1455.1400146
2000-01-19	1461.3900146	1448.6800537	1455.1400146	1455.9000244	1087800000	1455.9000244
2000-01-20	1465.7099609	1438.5400391	1455.9000244	1445.5699463	1100700000	1445.5699463
2000-01-21	1453.1800537	1439.5999756	1445.5699463	1441.3599854	1209800000	1441.3599854
2000-01-24	1454.0899658	1395.4200439	1441.3599854	1401.5300293	1115800000	1401.5300293
2000-01-25	1414.2600098	1388.4899902	1401.5300293	1410.0300293	1073700000	1410.0300293

Example : kospi vs S&P500

Rename columns

kospi

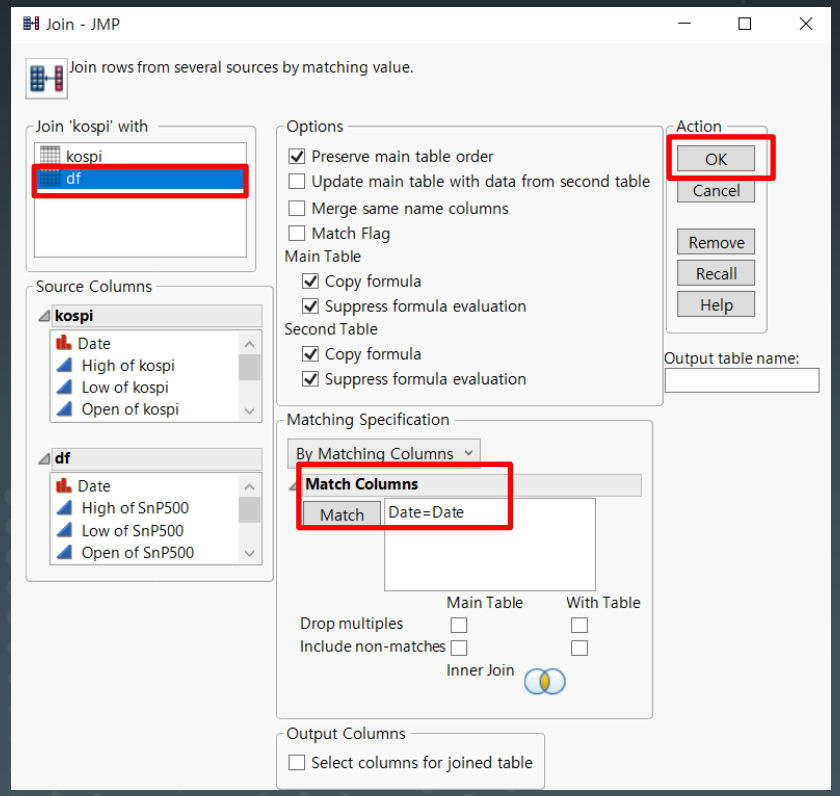
Date	High of kospi	Low of kospi	Open of kospi	Close of kospi
2000-01-04	1066.1800537	1016.5900269	1028.3299561	1059.0400391
2000-01-05	1026.5200195	984.04998779	1006.8699951	986.30999756
2000-01-06	1014.9000244	953.5	1013.9500122	960.78997803
2000-01-07	970.15997314	930.84002686	949.16998291	948.65002441
2000-01-10	994.94000244	974.82000732	979.66998291	987.23999023
2000-01-11	1005.8699951	981.22998047	992.16998291	981.33001709
2000-01-12	968.67999268	949.20001221	957.97998047	955.01000977
2000-01-13	960.72998047	939.25	955.05999756	951.04998779
2000-01-14	970.59002686	937.75	958.82000732	948.0300293
2000-01-17	986.09002686	959.35998535	962.66998291	983.27001953
2000-01-18	992.84997559	967.96002197	992.30999756	981.5300293
2000-01-19	969.30999756	938.73999023	969.30999756	938.7800293
2000-01-20	945.90002441	906.26000977	927.11999512	945.90002441
2000-01-21	940.70001221	911.83001709	928.54998779	925.15997314
2000-01-24	944.30999756	915.70001221	917.92999268	926.77001953
2000-01-25	913.19000244	891.2199707	909.11999512	891.2199707

S&P500

Date	High of SnP500	Low of SnP500	Open of SnP500	Close of SnP500
1999-12-31	1472.4200439	1458.1899414	1464.4699707	1469.25
2000-01-03	1478	1438.3599854	1469.25	1455.2199707
2000-01-04	1455.2199707	1397.4300537	1455.2199707	1399.4200439
2000-01-05	1413.2700195	1377.6800537	1399.4200439	1402.1099854
2000-01-06	1411.9000244	1392.0999756	1402.1099854	1403.4499512
2000-01-07	1441.4699707	1400.7299805	1403.4499512	1441.4699707
2000-01-10	1464.3599854	1441.4699707	1441.4699707	1457.5999756
2000-01-11	1458.6600342	1434.4200439	1457.5999756	1438.5600586
2000-01-12	1442.5999756	1427.0799561	1438.5600586	1432.25
2000-01-13	1454.1999512	1432.25	1432.25	1449.6800537
2000-01-14	1473	1449.6800537	1449.6800537	1465.1500244
2000-01-18	1465.1500244	1451.3000488	1465.1500244	1455.1400146
2000-01-19	1461.3900146	1448.6800537	1455.1400146	1455.9000244
2000-01-20	1465.7099609	1438.5400391	1455.9000244	1445.5699463
2000-01-21	1453.1800537	1439.5999756	1445.5699463	1441.3599854
2000-01-24	1454.0899658	1395.4200439	1441.3599854	1401.5300293
2000-01-25	1414.2600098	1388.4899902	1401.5300293	1410.0300293
2000-01-26	1412.7299805	1400.1600342	1410.0300293	1404.0899658

Example : kospi vs S&P500

Join the kospi vs S&P500
Menu : Tables > Join > Join kospi with df >
match column : Date = Date > OK

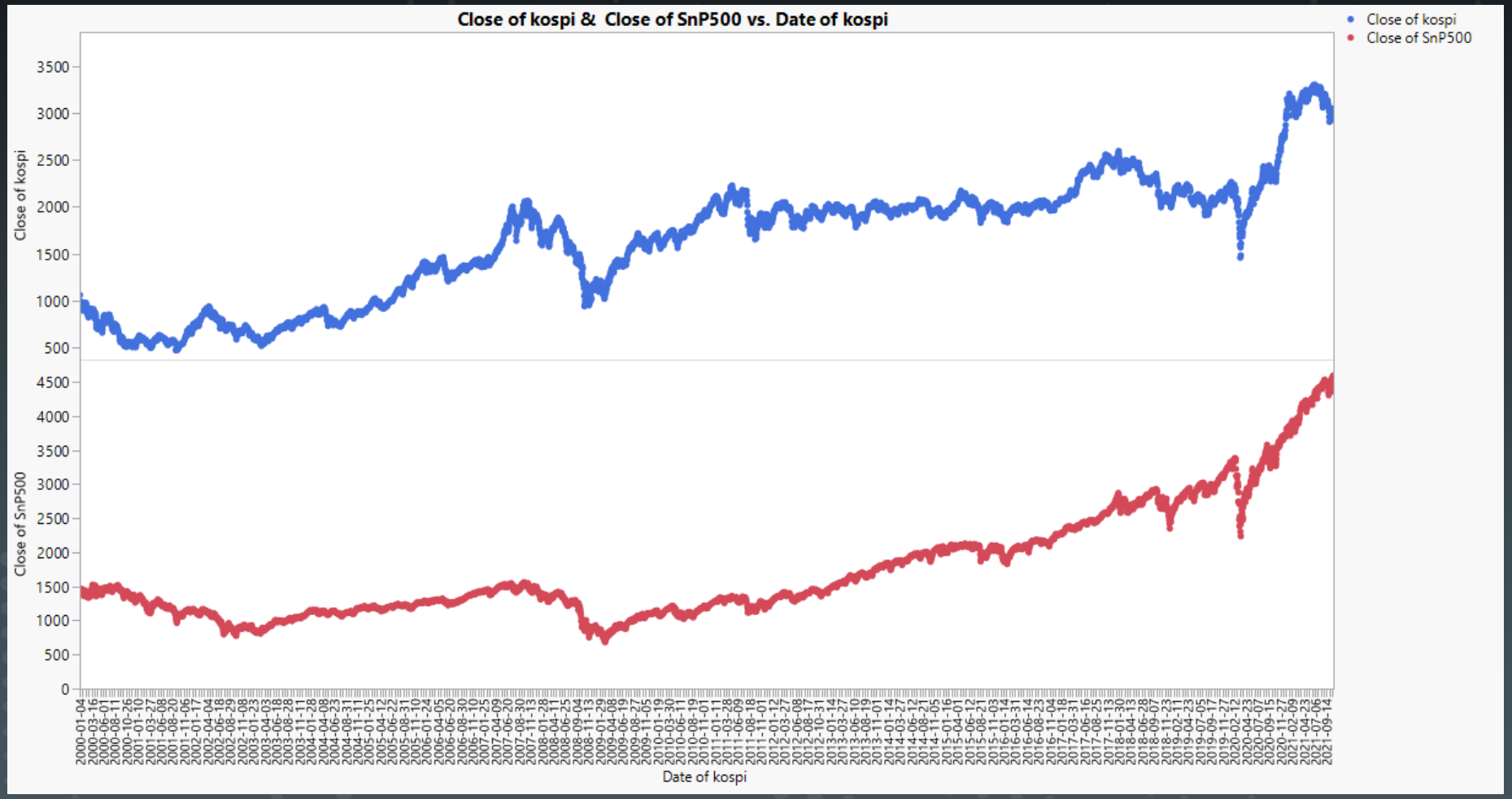


Join data Table (kospi and S&P 500)

Date of kospi	High of kospi	Low of kospi	Open of kospi	Close of kospi	Volume of kospi	Adj Close of kospi	Date of df	High of SnP500	Low of SnP500	Open of SnP500	Close of SnP500	Volume of SnP500	A
2000-01-04	1066.1800537	1016.5900269	1028.3299561	1059.0400391	195900	1059.0400391	2000-01-04	1455.2199707	1397.4300537	1455.2199707	1399.4200439	1009000000	
2000-01-05	1026.5200195	984.04998779	1006.8699951	986.30999756	257700	986.30999756	2000-01-05	1413.2700195	1377.6800537	1399.4200439	1402.1099854	1085500000	
2000-01-06	1014.9000244	953.5	1013.9500122	960.78997803	203500	960.78997803	2000-01-06	1411.9000244	1392.0999756	1402.1099854	1403.4499512	1092300000	
2000-01-07	970.15997314	930.84002686	949.16998291	948.65002441	215700	948.65002441	2000-01-07	1441.4699707	1400.7299805	1403.4499512	1441.4699707	1225200000	
2000-01-10	994.94000244	974.82000732	979.66998291	987.23999023	240200	987.23999023	2000-01-10	1464.3599854	1441.4699707	1441.4699707	1457.5999756	1064800000	
2000-01-11	1005.8699951	981.22998047	992.16998291	981.33001709	257100	981.33001709	2000-01-11	1458.6600342	1434.4200439	1457.5999756	1438.5600586	1014000000	
2000-01-12	968.67999268	949.20001221	957.97998047	955.01000977	227100	955.01000977	2000-01-12	1442.5999756	1427.0799561	1438.5600586	1432.25	974600000	
2000-01-13	960.72998047	939.25	955.05999756	951.04998779	223000	951.04998779	2000-01-13	1454.1999512	1432.25	1432.25	1449.6800537	1030400000	
2000-01-14	970.59002686	937.75	958.82000732	948.0300293	225000	948.0300293	2000-01-14	1473	1449.6800537	1449.6800537	1465.1500244	1085900000	
2000-01-18	992.84997559	967.96002197	992.30999756	981.5300293	212100	981.5300293	2000-01-18	1465.1500244	1451.3000488	1465.1500244	1455.1400146	1056700000	
2000-01-19	969.30999756	938.73999023	969.30999756	938.7800293	235700	938.7800293	2000-01-19	1461.3900146	1448.6800537	1455.1400146	1455.9000244	1087800000	
2000-01-20	945.90002441	906.26000977	927.11999512	945.90002441	226700	945.90002441	2000-01-20	1465.7099609	1438.5400391	1455.9000244	1445.5699463	1100700000	
2000-01-21	940.70001221	911.83001709	928.54998779	925.15997314	255100	925.15997314	2000-01-21	1453.1800537	1439.5999756	1445.5699463	1441.3599854	1209800000	
2000-01-24	944.30999756	915.70001221	917.92999268	926.77001953	210400	926.77001953	2000-01-24	1454.0899658	1395.4200439	1441.3599854	1401.5300293	1115800000	
2000-01-25	913.19000244	891.2199707	909.11999512	891.2199707	242600	891.2199707	2000-01-25	1414.2600098	1388.4899902	1401.5300293	1410.0300293	1073700000	
2000-01-26	902.42999268	875.82000732	896.34002686	885.53997803	231100	885.53997803	2000-01-26	1412.7299805	1400.1600342	1410.0300293	1404.0899658	1117300000	
2000-01-27	912.48999023	879	888.27001953	909.22998047	344900	909.22998047	2000-01-27	1418.8599854	1370.9899902	1404.0899658	1398.5600586	1129500000	
2000-01-28	943.95001221	916.85998535	922.73999023	941.66998291	293100	941.66998291	2000-01-28	1398.5600586	1356.1999512	1398.5600586	1360.1600342	1095800000	
2000-01-31	948.84002686	922.91998291	924.83001709	943.88000488	289400	943.88000488	2000-01-31	1394.4799805	1350.1400146	1360.1600342	1394.4599609	993800000	
2000-02-01	959.30999756	923.52001953	955.44000244	928.75	306100	928.75	2000-02-01	1412.4899902	1384.7900391	1394.4599609	1409.2800293	981000000	
2000-02-02	950.13000488	923.40002441	929.69000244	943.59002686	310400	943.59002686	2000-02-02	1420.6099854	1403.4899902	1409.2800293	1409.1199951	1038600000	
2000-02-03	959	934.10998535	950.26000977	950.2199707	277100	950.2199707	2000-02-03	1425.7800293	1398.5200195	1409.1199951	1424.9699707	1146500000	
2000-02-07	982.0300293	951.30999756	953.22998047	973.13000488	271800	973.13000488	2000-02-07	1427.1500244	1413.3299561	1424.3699951	1424.2399902	918100000	
2000-02-08	981.41998291	960.4699707	977.07000732	961.2199707	229500	961.2199707	2000-02-08	1441.8299561	1424.2399902	1424.2399902	1441.7199707	1047700000	
2000-02-09	979.91998291	966.15002441	967.5300293	976.04998779	240400	976.04998779	2000-02-09	1444.5500488	1411.6500244	1441.7199707	1411.7099609	1050500000	
2000-02-10	991.71002197	957.72998047	966.35998535	966.17999268	251600	966.17999268	2000-02-10	1422.0999756	1406.4300537	1411.6999512	1416.8299561	1058800000	
2000-02-11	990.75	941.23999023	990.10998535	953.2199707	274800	953.2199707	2000-02-11	1416.8299561	1378.8900146	1416.8299561	1387.1199951	1025700000	
2000-02-14	941.22998047	907.90997314	932.94000244	910.86999512	234200	910.86999512	2000-02-14	1394.9300537	1380.5300293	1387.1199951	1389.9399414	927300000	
2000-02-15	919.42999268	875.04998779	916.89001465	879.70001221	218600	879.70001221	2000-02-15	1407.7199707	1376.25	1389.9399414	1402.0500488	1092100000	
2000-02-16	886.63000488	843.58001709	881.0300293	879.5	214000	879.5	2000-02-16	1404.5500488	1385.5799561	1402.0500488	1387.6700439	1018800000	

Example : kospi vs S&P500

Graph builder (X axis : Date of kospi , Y axis : Close of kospi, Y axis : Close of SnP500)



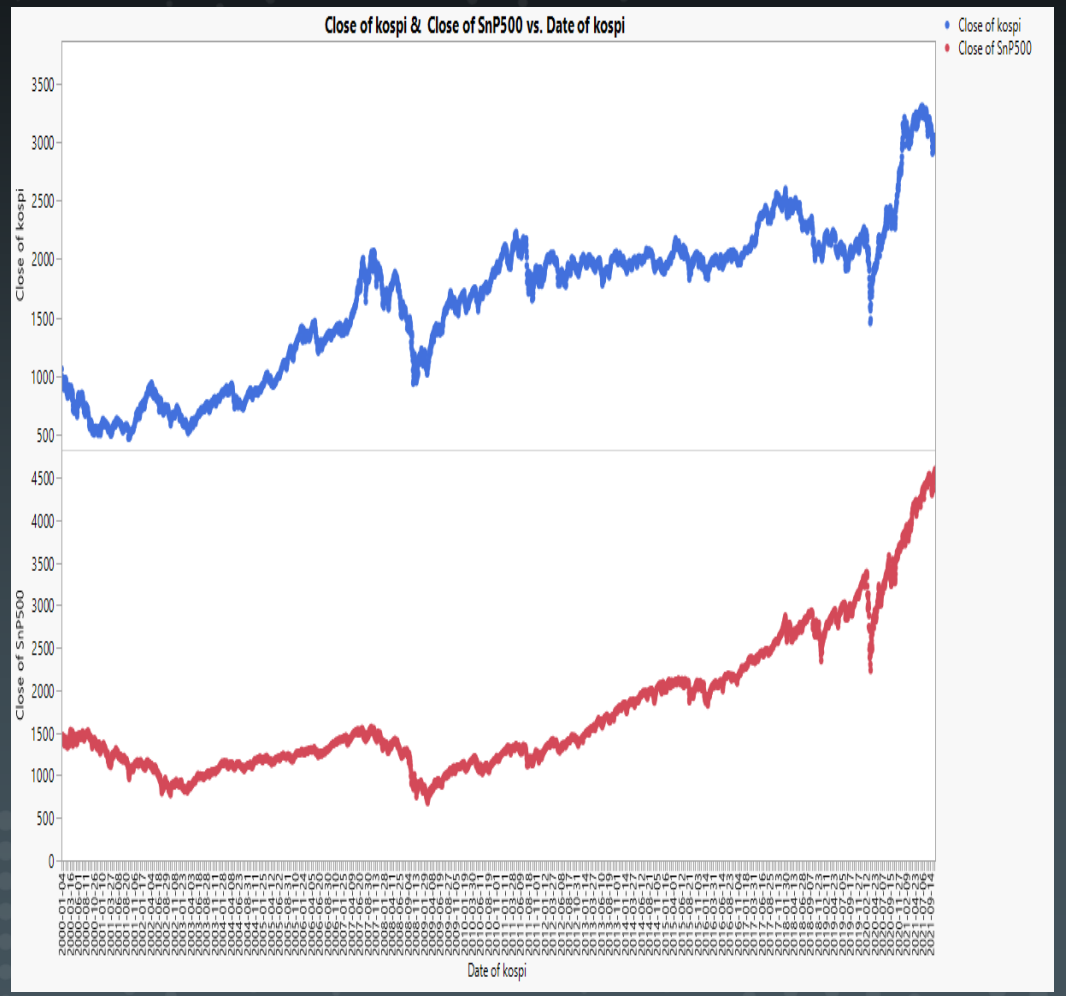
Example : kospi vs S&P500

JMP Script

```
Python Init();

Python Submit("\[
import pandas as pd
import pandas_datareader as pdr
from pandas_datareader import data
import pandas_datareader.data as web
from datetime import datetime

#df = data.DataReader("^KS11", "yahoo").reset_index()

start_date = datetime(2000,1,1)
end_date = pd.Timestamp.today()
kospi = data.get_data_yahoo("^KS11", start_date, end_d

]\");

kospi = Python Get( kospi );

Python Term();

kospi <<New Data View;
Data Table("kospi"): Date << Set Modeling Type( "Nomir
Data Table( "kospi" ):Date << Format( "yyyy-mm-dd", 25

Data Table( "kospi" ):High << Set Name(" High of kospi
Data Table( "kospi" ):Low << Set Name(" Low of kospi")
Data Table( "kospi" ):Open << Set Name(" Open of kospi
Data Table( "kospi" ):Close << Set Name(" Close of kos
Data Table( "kospi" ):Volume << Set Name("Volume of ko
Data Table( "kospi" ):Adj Close << Set Name("Adj Close

Python Init();
Python Submit("\[
import pandas as pd
import pandas_datareader as pdr
from pandas_datareader import data
import pandas_datareader.data as web
from datetime import datetime
```

```
start_date = datetime(2000,1,1)
end_date = pd.Timestamp.today()
df = data.get_data_yahoo("^GSPC", start_date, end_date
]\");

df = Python Get( df );
Python Term();

df <<New Data View;
Data Table("df"): Date << Set Modeling Type( "Nominal"
Data Table( "df" ):Date << Format( "yyyy-mm-dd", 25 );
Data Table( "df" ):High << Set Name(" High of SnP500")
Data Table( "df" ):Low << Set Name(" Low of SnP500");
Data Table( "df" ):Open << Set Name(" Open of SnP500")
Data Table( "df" ):Close << Set Name(" Close of SnP500
Data Table( "df" ):Volume << Set Name("Volume of SnP50
Data Table( "df" ):Adj Close << Set Name("Adj Close of

Data Table( "kospi" ) << Join(
    With( Data Table( "df" ) ),
    By Matching Columns( :Date = :Date ),
    Drop multiples( 0, 0 ),
    Include Nonmatches( 0, 0 ),
    Preserve main table order( 1 )
);

Graph Builder(
    Size( 1125, 703 ),
    Variables(
        X( :Date of kospi ),
        Y( :Name( " Close of kospi" ) ),
        Y( :Name( " Close of SnP500" ) )
    ),
    Elements( Position( 1, 1 ), Points( X, Y, Legend(
    Elements( Position( 1, 2 ), Points( X, Y, Legend(
);
```

Graph builder

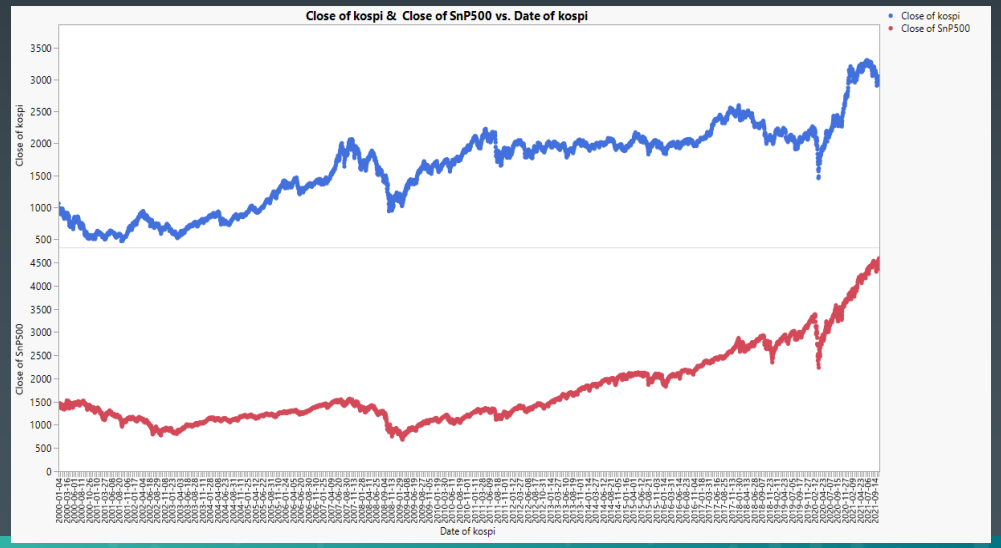


. Example : kospi vs S&P500

. Task scheduler in window system

JMP 실무 활용 가이드

Coffee with JMP Trend



KOREA 2021

DISCOVERY SUMMIT

EXPLORING DATA
INSPIRING INNOVATION

jmp