

Contents

프로젝트 진행 결과	01
- 기설수립및검증	
- 필요서비스선정및데이터분석	
개요	02
- WBS	
- R&R	
- 사용 언어, 라이브러리	
수행내용	03
- 프로젝트 수립 배경	
- 선용품구매 플랫폼구축	
- 체류시간별 관광정보 제공 서비스	
프로젝트 중점 시항 및 향후 계획	04
부록	05

1. 프로젝트 진행결과

- 1 기설수립 및 검증
 - 1 초기수립 가설

"부산항·수출·입 물동량이 증가한다면, 입항을 통한 외국 선원들의 체류시간 또한 증가할 것이다."

- 2 기설검증(데이터 상관분석)
 - 년간 부산항 물동량과 부산항 회물처리 물동량 상관분석
 - : 상관계수 약 0.999 → 양의 상관관계
 - 년간 부산항 화물처리 물동량과 체류시간 상관분석
 - : 상관계수 약 0.493 → 양의 상관관계

부산항수출·물동량증가->화물처리 물동량증가 화물처리 물동량증가->외국선박 및 외국인 선원체류시간증가

- 2 필요 서비스 선정 및 데이터 분석 진행 여부
 - 1 서비스 대상 : 부산항에 입항하는 컨테이너 선박의 외국인 선원
 - 2 필요서비스별분석

선용품 구매 플랫폼 구축

- 선용품 판매 품목 선정 -> 외국인 선원 니즈 파악
- 선용품 품목별 입점 기업 리스트 확보 -> 선용품 유통 및 판매
- 선용품 품목별 워드클라우드 -> 플랫폼 구축 시 입점 기업의 자사 제품 효율적인 판매 등록
- 공실데이터위치 -> 플랫폼 구축시, 온라인 주문을 통한상품 보관 및 수령 장소

체류시간별 관광정보 제공

- 체류시간 현황분석〉체류시간 분포에 따른 코스 선정
- 관광지, 맛집, 카페 웹크롤링 ->관광코스 데이터 시각화
- 체류시간별 추천코스시각화 -> 관광정보 코스시각화제공

구분	수집	걔공	시각화	진행률
선용품구매플랫폼구축을 위한데이터분석	√	V	√	100%
체류시간별 관광정보 제공을 위한데이터 분석	√	√	V	100 70

1 WBS(Work Breakdown structure)

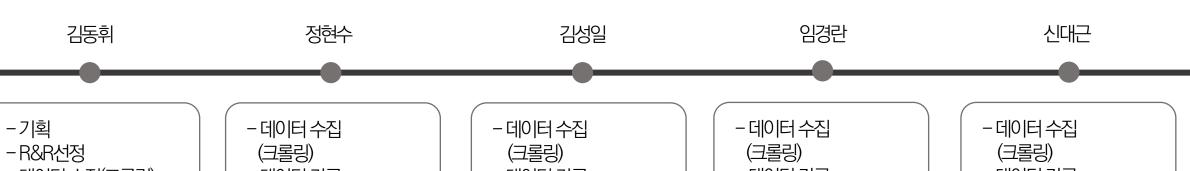
-프로젝트 기간: 24.02.27 ~ 24.03.05

-진행과업:총21건/진행률:100%

-투입인원: 총 5명(김동휘, 정현수, 김성일, 임경란, 신대근)

				소분류	세분류					2月		3月					
구분	과업명	대분류	중분류			담당자(정)	담당자(부)	진행률	27	28	29	1	2	3	4	5	
									화	수	목	금	토	일	월	화	
1	기획 및 R&R 지정					김동휘	팀원전원	100%									
						김동휘	팀원전원	100%									
2	도메인 프로셔스 및 메타 데이터 파악	-				임송위	10000	100%		 							
3	연도별 물동량 추이 파악	선용품 국내 플랫폼	데이터 수집	데이터 크롤링	연도별 물동량	김동휘	김성일	100%									
4	선사별 체류 시간 추이 파악	선용품 국내 플랫폼	데이터 수집	데이터 크롤링	부산항 선사별 체류시간	김성일	김동휘	100%									
5	Top3 품목 선정	선용품 국내 플랫폼	데이터 수집	공공 데이터 다운	관세청_전국 및 부산권항 선용품 무역통계	정현수	신대근	100%									
6	워드 귤라우드 선정	선용품 국내 플랫폼	데이터 수집	데이터 크롤링	품목별 키워드 분석(외국 → 한국)	신대근	임경란	100%									
7	부동산 공실정보 크롤링	선용품 국내 플랫폼	데이터 전처리	데이터 크롤링	네이버 부동산 : 상가/사무실/공장/창고/토지	임경란	김동휘	100%									
				414 220	4-1-11-100-10-11-20-00-0-12-1						 						
8	수집 데이터 전처리	선용품 국내 플랫폼	데이터 전처리		-	정현수	김성일	100%		-							
9	전처리 테이터 시각회	선용품 국내 플랫폼	데이터 시각화			김성일	정현수	100%									
10	Selenium + BS 데이터 수집 + 가공 : 체류시간/선사별 소재국가 데이터	체류시간별 관광코스 추천	데이터 수집	데이터 크롤링	부산항 선사별 체류시간 + 선사별 국가	김동휘	정현수	100%									
10	Selenum + BS 데이터 우집 + 가장 : 세뉴시간/건사별 조세국가 데이터	세뉴시간별 관광고의 우인	데이터 우십	네이터 크ె질링	구인당 인사별 세뉴시간 + 인사별 국가			100%									
11	Selenium + BS 데이터 수집 : 체류시간/재방문을 데이터	체류시간별 관광코스 추천	데이터 수집	데이터 크롤링	체류시간과 재방문율의 연관분석	김동휘	김성일	100%									
12	Selemium + BS 데이터 수집 : 관광지, 주소 크롤링	체류시간별 관광코스 추천	데이터 수집	데이터 크롤링	관광명소	임경란	정현수	100%									
						ALCILO	71 14 01	4000									
13	Selenium + BS 데이터 수집 : 구글 평점 4.5 이상 맛집	체류시간별 관광코스 추천	데이터 수집	데이터 크롤링	맛집명소	신대근	김성일	100%		 							
14	Selenium + BS 데이터 수집 : 구글 평점 4.5 이상 관광	체류시간별 관광코스 추천	데이터 수집	데이터 크롤링	숙박명소	정현수	임경란	100%									
15	데이터 가공 : 체류시간별 부산항 -관광지-부산항 형식의 데이터셋	체류시간별 관광코스 추천	데이터 시각화	본류	부산항 - 관광	팀원전원	팀원전원	100%									
16	데이터 가공 : 체류시간별 부산항 -관광지-맛집-부산항 형식의 데이터셋	체류시간별 관광코스 추천	데이터 시각화	분류	0	팀원전원	팀원전원	100%									
17	데이터 가공 : 체류시간별 부산항-관광지-맛집-숙박-부산항 형식의 데이터셋	체류시간별 관광코스 추천	데이터 시각화	분류	부산항 - 관광 + 맛집 + 숙박	팀원전원	팀원전원	100%									
18	미비파업 보완					팀원전원	팀원전원	100%									
					-												
19	프로젝트 중간발표	-	-			팀원전원	팀원전원	100%									
20	프로젝트 중간리뷰					팀원전원	팀원전원	100%									
24	TABLE MARKET					팀원전원	팀원전원	10006									
21	프로젝트 최종발표					15년인원	16년선원	100%									

2 R&R(Roles And Responsibilities)



- -데이터 수집(크롤링)
- -데이터가공
- -데이터시각화 (워드크라우드, Osmnx)
- -상관분석
- -Git 관리

- -데이터가공
- -시각화
- : 워드클라우드, 공실데이터
- 상관분석

- -데이터가공
- -시각화
- : 워드크라우드, 공실데이터
- 상관분석
- 배포(Stream Lit)

- -데이터가공
- -시각화
- : 워드클라우드, 공실데이터, Osmnx
- 상관분석

- -데이터가공
- -시각화
- : 워드클라우드, Osmnx
- 상관분석

3 사용언어 및 라이브러리

언어





기타시각화툴

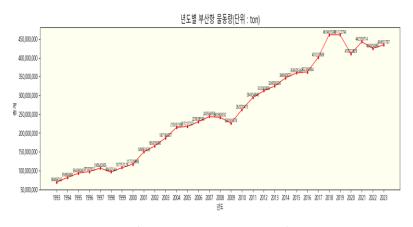


3. 프로젝트 수립 배경

1

부산항만공사의 현상황

부산항을 글로벌 해운 / 물류 중심기지로 육성하려는 목적 달성을 위해 해외 선박과 외국인 선원들을 대상으로 다양한 서비스 제공 주력 中



그 중, 부산항국제선용품유통센터(BISC)의 경우

- ✓ 외국인 선호 품목 대상이 아닌, 로프 및 튜브 등 선박 관리 용품 위주의 판매
- ✓ 온라인이 아닌 오프라인 방식으로 거래 진행
- ✓ 외국인 선원들을 대상으로 시행되는 서비스 미흡한 상황



부산항국제선용품유통센터의 구체화 및 개선 필요



- ✓ 온라인 선용품 구매 플랫폼 구축
- ✓ 체류시간별부산관광코스정보제공서비스기획

이에 맞춰 각서비스별 필요한 과업에 따라 데이터 분석 진행



1. 데이터 수집 및 전처리

: 물동량의 연관 분석을 위해 시계열 및 크롤링을 통한 데이터 수집 및 정제

데이터수집출처

• 체류시간: 공공데이터 -부산신항국제터미널 정보서비스 (https://www.pnitl.com/infoservice/vessel/vslScheduleList.jsp)

• 물동량: 부산항만공사사이트 → 부산항통계 → Port-MIS(회물) → 월별 회물 처리실적(확정)

• 기간: 2013. 01~2023. 12

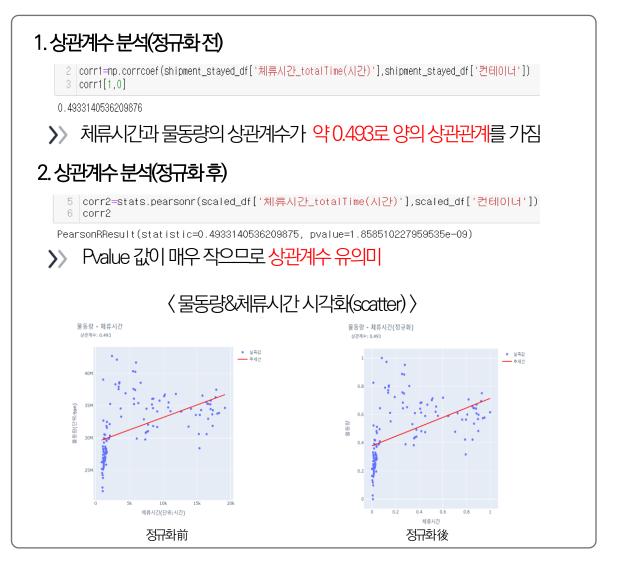
• 수집데이터타입:정형데이터

데이터정보

	년도	월	체류시간_totalTime(시간)		Unnamed: 0	년/월	년도	월 컨테이너		년도	월	컨테이너	체류시간_totalTime(시간)
0	2013	1	1088	0	0	2013-01-01	2013	1 23818165	0	2013	1	23818165	1088
1	2013	2	1052	1	1	2013-02-01	2013	2 21777930	1	2013	2	21777930	1052
2	2013	3	1175	2	2	2013-03-01	2013	3 26068371	2	2013	3	26068371	1175
3	2013	4	1203	3	3	2013-04-01	2013	4 25903253	3	2013	4	25903253	1203
4	2013	5	1180	4	4	2013-05-01	2013	5 25397562	4	2013	5	25397562	1180
127	2023	8	17367	127	127	2023-08-01	2023	8 34292855	127	2023	8	34292855	17367
128	2023	9	17003	128	128	2023-09-01	2023	9 35206478	128	2023	9	35206478	17003
129	2023	10	18293	129	129	2023-10-01	2023	10 33865762	129	2023	10	33865762	18293
130	2023	11	16577	130	130	2023-11-01	2023	11 35283219	130	2023	11	35283219	16577
131	2023	12	12671	131	131	2023-12-01	2023	12 34096319	131	2023	12	34096319	12671
A .	시계열	별.	외국선원체류시간		▲컨테이너	처리실적	넉합기	(단위 : 톤)		외국	<u> </u>	[체류시]	간+컨테이너처리실

2. 데이터 연관 분석

: 물동량과 체류시간의 연관 분석을 통한 선용품 구매 플랫폼 서비스의 니즈 파악





1,2,3번 분석 결과

- ✓ 물동량과 체류시간의 양의 상관계수 지님
- ✓ 외국인 선원들의 체류시간 증가 추세 예상 가능
- » 외국인 선원 대상 전략 인사이트 필요 · 서우프 그래 프래프 그후 및 광강(이 나이트 웨어) 후
 - : 선용품 구매 플랫폼 구축 및 관광 인사이트 제안 효과성 높을 것으로 예상

3. 선용품 품목 분석

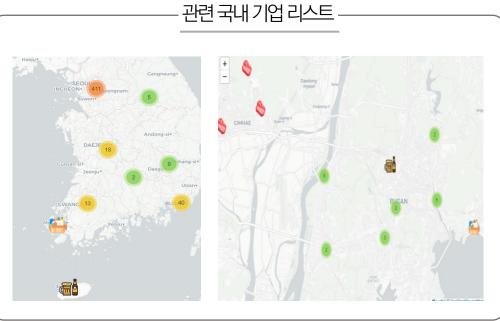
: 외국인 선호 선용품 분석을 통한 TOP4 항목에 대해 국내 기업 소싱 리스트의 지도 시각화

데이터 수집 출처

- 외국인 선호 선용품: 해당 품목 영어로 검색 후, 관련 도메인 사이트 크롤링
- 국내 기업 리스트: 전자공시시스템(DART) 사이트 -> 기업 개황 -> 업종별 크롤링
- 수집 데이터 타입:정형 데이터

데이터정보





4. 선용품 보관 창고 위치 분석

: 선용품 보관 및 수령 장소 관련한 창고 리스트의 시각화

데이터 수집출처

- 네이버 부동산 웹크롤링
- 검색조건: 부산신항만 기준으로 10~20분 내 거리에 위치한 상가·업무·공장·토지
- 수집 데이터 타입:정형 데이터

데이터정보

6개 구역에 대한 구매 형태, 면적, 금액 분석

	구분	타입	타입별 개수	계약면적(m²)	전용면적(m²)	대지면적(m²)	매매가(단위:만원)	보증금 (단위: 만원)	월세(단위:만원)
0	명지동	매매	216	467	349	595	279739	0	0
1	명지동	월세	517	1820	182	4	0	3425	207
2	명지동	전세	1	204	122	0	0	0	0
3	성북동	매매	6	0	0	4985	81677	0	0
4	송정동	매매	15	0	0	6592	981420	0	0
5	송정동	월세	5	600	71	599	0	2600	252
6	신호동	매매	23	41	31	362	80172	0	0
7	신호동	월세	19	268	174	205	0	3000	215
8	용원동	매매	9	171	136	1748	1488222	0	0
9	용원동	월세	10	493	210	0	0	6700	414
10	천성동	매매	10	0	0	1883	138200	0	0

지도 Tooltip 을통한 시각화



<u>3. 수행 내용(2) : 외국인 추천 관광 코스</u>

1. 체류 시간 및 관광 코스 관련 데이터 수집 및 정제

: 부산 지역 내의 관광 코스 리스트 크롤링 및 체류 시간별 코스 임의 지정

데이터수집출처

- 구글 맵웹크롤링
- 검색 조건 : 부산 신항만 기준으로 10km 이내 맛집, 숙박업소, 관광지 (평점 4점 이상)
- 수집 데이터 타입:정형 데이터

데이터정보

맛집, 숙박업소, 관광지 크롤링

가게명	평점	구분	lat		Ing 주소		소1					
진미식당	4	간이음식점	35.10786	12	29.0373	중잉	동4기					
종필이집	4.5	음식점	35.11187	12	29.0351	영주	제1등					
대하가야밀	4.5	음식점	35.10907	12	29.0381	중잉	동4기					
시민카츠 🖥	4.2	일식	35.10467	1.7	20 0257		2 6				. 1	_
생강나무	5	간이음식점	35.10714	13	업소	명	평	섬	구분	=	lat	Ing
유원(留園)	4.3	중식	35.10799	1	디노호	텔		4.1	3성급	호털	35.11592	129.0388
대영정	4	한식	35.10798	1.	지엔비	능턴		1	7선근	ㅎ턴	35.10127	129.0248
말자씨부의	4.3	음식점	35.10444	13	. —							
조은참치 🗧	4.4	음식점	35.10597	1	크라운	하ㅂ		4.1	4성급	호털	35.10758	129.0368
상짱	4.2	일식	35.10429	1	비센트	호털		4	2성급	호털	35.11077	129.0393
					코모도	호털		4.1	4성급	호털	35.10905	129.0351
				-	아모드	ㅎ털		4	3성급	후털	35 11377	129 0407

▲ 구글맵 API 를 이용해 크롤링한 주소 위도, 경도 추출



▲평균체류시간

체류시간별 데이터 수집 및 루트제공

◎◎ 체류7시간

1)부산신항-관광지1-관광지2-맛집-부산신항 2)부산신항-맛집1-관광지1-관광지2-부산신항



1)부산신항-맛집1-관광지1-맛집2-관광지2-숙박-부산신항 2)부산신항-관광지1-맛집1-관광지2-맛집2-숙박-부산신항



○ 체류 3일차

DAY1: 부산신항 - 관광지1 - 맛집1 - 관광지2 - 맛집2 - 숙박 DAY2:숙박-관광지3-맛집3-관광지4-맛집4-숙박 DAY3:숙박-관광지5-맛집5-관광지6-맛집6-부산신항

3. 수행 내용(2): 외국인 추천 관광 코스

2. 체류 시간에 따른 부산 지역 관광 코스 제공

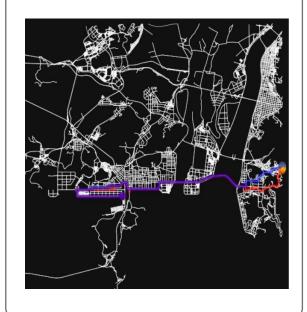
: OSmnx 라이브러리로 각 체류 시간별 맛집, 관광지, 숙박에 맞춰 루트 시각화

1 5시간코스

부산신항→〉감천문화마을 →〉 진미식당→〉 부산신항



부산신항 → 감천문화마을 → 오륙도 스카이워크 → 생강나무 → 부산신항



3 3일코스

1 DAY

- : 부산신항 -> 감천문화마을 -> 진미식당 -> 해동용궁사 -> 종필이집 -> 디노호텔 2 DAY
- : 디노호텔 -〉부산시민공원 -〉대하가아밀면 -〉이기대 농바위 -〉시민카츠 -〉디노호텔 3DAY
- : 디노호텔 -> 용두산공원 -> 생강나무 -> 오시리아 해안산책로 -> 유원 -> 부산신항



3. 수행 내용 결과 링크

데이터 분석 산출물 관련 자료 링크

1 Streamlit :시각화위주 https://busanportservice.streamlit.app/

2 GitHub :코드위주 https://github.com/Kshi0219/Adios?tab=readme-ov-file)

4. 프로젝트 중점 사항 및 향후 계획

프로젝트 수행간 중점사항

과업별 기한 엄수 주어진 기간 내 팀 목표 실현 및 성과 출현 도메인 및 메타데이터 파악 무역 프로세스와 프로세스간 발생되는 데이터 의미파악을 통한 신속한 데이터 수집/가공/시각화 가능 AS-IS & TO-BE 구체화 팀 목표와 팀원 간의 과업연계를 위해 Daily 미팅진행으로 맡은 과업내용과 목표 구체화 직접코딩 지향 - 'Excel 다운로드' 버튼을 통한 데이터 수집 지양. - Selenium, BeautifulSoup 등을 활용한 직접코딩 및 전처리를 통한 코딩스킬 극대화 Code Convention 기반의 코딩진행 변수/함수 명명 및 주석규칙을 준수하여 팀원 부재 시 지체 없는 과업진행 가능 과업별후코드리뷰

단계별 코드 공유를 통한 효율적인 코딩 스킬 학습 및 적용

향후계획

- 프로젝트간 개인담당 괴업 외 다른 팀원이 수행한 괴업관련 수집/가공/시각화 진행
- 프로젝트간 진행한 과업 별 수행내용 개인 블로그 등에 기록
- 다른 도메인 선정 및 수집/분석/시각화 진행

<u>5.</u> 부록 : 코드

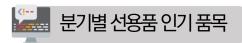


```
> Users > Lenovo > Downloads > ♥ 3 물동량 - 화물 처리실적 상관분석.py > ...
   import streamlit as st
   import numpy as np
   import pandas as pd
    import plotly
   import plotly.graph_objects as go
   from sklearn.preprocessing import MinMaxScaler
   import scipy.stats as stats
   st.title('물동량 - 화물처리실적 상관분석')
   ton_shipgoods_df=pd.read_csv('data/물동량-화물처리실적.csv').drop('Unnamed: 0',axis=1)
   ton_shipgoods_df.set_index('년도',inplace=True)
   ton shipgoods scaled=pd.DataFrame(
       scaler.fit_transform(ton_shipgoods_df),
       columns=ton_shipgoods_df.columns
   ton_shipgoods_scaled.index=ton_shipgoods_df.index
   ton_shipgoods_scaled.reset_index(inplace=True)
   corr=stats.pearsonr(ton_shipgoods_scaled['물동량'],
                      ton shipgoods scaled['화물처리실적'])
   st.dataframe(ton_shipgoods_df,use_container_width=True)
   st.write(f"상관계수 : {round(corr[0],3)}")
   st.write(f"P-values : {round(corr[1],3)}")
   def plotly_gen_corr():
       fig=go.Figure()
       fig.add_trace(go.Scatter(
          x=ton_shipgoods_scaled['물동량'],
           y=ton_shipgoods_scaled['화물처리실적'],
          mode='markers',name='실측값'
       fig.add_trace(go.Scatter(
          x=ton_shipgoods_scaled['물동량'],
           y=np.poly1d(np.polyfit(ton_shipgoods_scaled['물동량'],ton_shipgoods_scaled['화물처리실적'],1)
              (ton_shipgoods_scaled['물동량']),
           mode='lines',name='추세선',
           line=dict(color='red')
       fig.update_layout(title='연간 물동량 - 화물처리실적 상관분석',
                      xaxis=dict(title='물동량'),
                      yaxis=dict(title='화물처리실적'),
                      annotations=[
                          dict(x=-0.03,
                             y=1.15,
                             xref='paper',
                             text=f'상관계수 : {round(corr[0],3)}\t(정규화된 수치로 계산)',
                             showarrow=False)])
       return fig
   st.plotly_chart(plotly_gen_corr())
```

```
화물 처리실적 - 체류시간 상관분석.py 1 🍨
> Users > Lenovo > Downloads > 🍁 4_화물 처리실적 - 체류시간 상관분석.py > ...
   import streamlit as st
    import numpy as np
   import pandas as pd
   import plotly.graph_objects as go
   from sklearn.preprocessing import MinMaxScaler
   import scipy.stats as stats
   st.title('년/월별 화물처리량 - 체류시간 상관분석')
   ship_stayed_df=pd.read_csv('data/처리설적-체류시간_상관분석.csv').drop('Unnamed: 0',axis=1)
   ship_stayed_df.rename(columns={'컨테이너':'화물처리량',
                              체류시간_totalTime(시간) : 체류시간 },
                       inplace=True)
   ship_stayed_df=ship_stayed_df.sort_values(['년도','월']).reset_index()\
      .drop('index',axis=1)
   for i in range(len(ship_stayed_df)):
      y_m.append(str(ship_stayed_df['년도'][i])+'-'+str(ship_stayed_df['월'][i]))
   ship_stayed_df['년/월']=y_m
   ship_stayed_df['년/월']=pd.to_datetime(ship_stayed_df['년/월'])
   ship_stayed_df=ship_stayed_df.iloc[:,:-1]
   st.dataframe(ship_stayed_df,use_container_width=True)
   scaler=MinMaxScaler()
   ship_stayed_scaled=pd.DataFrame(
      scaler.fit_transform(ship_stayed_df.copy()),
       columns=ship_stayed_df.columns
   ship_stayed_scaled['년/월']=y_m
   corr=stats.pearsonr(ship_stayed_scaled['화물처리량'],
                    ship_stayed_scaled['체류시간'])
   st.write(f'상관계수 : {round(corr[0],3)}')
   st.write(f'P-value : {round(corr[1],3)}')
   st.write(f'P-value가 {round(corr[1],3)}로, 상관계수가 유의미하다고 보임')
   def plotly_gen_corr():
      fig=go.Figure()
       fig.add_trace(go.Scatter(x=ship_stayed_scaled['체류시간'],
                             y=ship_stayed_scaled['화물처리량'],
                             mode='markers',name='실촉값'))
       fig.add_trace(go.Scatter(
          x=ship_stayed_scaled['체류시간'],
           y=np.poly1d(np.polyfit(ship_stayed_scaled['체류시간'], ship_stayed_scaled['화물처리량'],1))\
              (ship_stayed_scaled['체류시간']),
          mode='lines',line=dict(color='red', width=2),
           name='추세션'
       fig.update_layout(title='년/월별 화물처리실적 - 체류시간(정규화)',
                     xaxis=dict(title='체류시간')
                     yaxis=dict(title='년/월별 화물처리실적'),
                     width=600,height=600,
                      hovermode='x',
```

▲ 물동량 & 회물처리량 상관분석 코드

▲ 화물처리량 & 체류시간 상관분석 코드

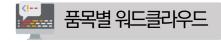


```
018년도 품목별 데이터 TOP3 (1).ipynb ●
  > Users > Lenovo > Downloads > 🛢 2018년도 품목별 데이터 TOP3 (1).ipynb > 🌵 라이브러리 호출
+ 코드 → Markdown │ D> 모두 실행 🚍 출력 모두 지우기 │ 🚞 개요 ↔
       라이브러리 호출
        import warnings
       warnings.filterwarnings('ignore')
       import pandas as pd
       df_1 = pd.read_excel('../mini_project/관세청_전국 및 부산권항 선용품 무역통계_20240131 (1).xlsx' , sheet_name = '2018년',
       df1.fillna(0, inplace = True) # 결축치들 0으로 바꿔주기
       df2.fillna(0, inplace = True) # 결촉치를 0으로 채워주기
       new_column_names = [*거래구변*, '종목*, '2018.1.건수', '2018.1.금액', '2018.2.금액', '2018.2.금액', '2018.3.건수', '2018.3.금액', 

'2018.4.건수', '2018.4.금액', '2018.5.건수', '2018.5.금액', '2018.6.건수', '2018.6.건수', '2018.6.건수', '2018.6.건수', '2018.0.금액', '2018.0.금액', '2018.0.건수', '2018.0.금액', '2018.0.건수', '2018.0.금액',
       dfl.columns = new_column_names # 칼럼 이름들 알아보기 쉽게 변경
       df2.columns = new_column_names # df1과 칼럼이름 통일
       df2.reset_index(drop=True, inplace = True) # 계산하기 위해 인덱스 정리
       df3=(df1.iloc[ : ,2:26] - df2.iloc[ : , 2:26])
       df3.insert(0, '품목', df1['품목']) # 계산하기 위해 제외시켜놨던 품목 칼럼 다시 가져옴
       df3.insert(0,'거래구분', df1['거래구분']) # 거래구분 칼럼도 다시 가져옴
       df3.to_excel('../mini_project/2018_data.xlsx', index=False)
       df4 = df3.iloc[1:13 , 1:8] # 1분기 외국선용품 품목 뽑기
       df5 = df4.iloc[:, [0,1,3,5]] # df4 에서 품목 건수만 뽑아내기
       df5['합계'] = df5['2018.1.건수'] + df5['2018.2.건수'] + df5['2018.3.건수']
       # 합계 칼럼을 만들어서 월별 각 품목마다의 합계 구하기
                품목 2018.1.건수 2018.2.건수 2018.3.건수 합계
                                      2622
                                                  3035 9029
                                                  2427 7184
```

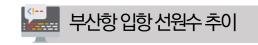
▲ 2018~2023년 분기별 인기 선용품 품목 분석

<u>5. 부록</u> : 코드



```
코드 + Markdown | ▷ 모두 실행 등 출력 모두 지우기 |  개요 ↔
 <u>주류관</u>련 크롤링
 1) 라이브러리 호출
     from selenium import webdriver #Selenium의 웹 드라이버를 사용하기 위한 모듈을 임포트
     from selenium.webdriver.common.by import By #Selenium에서 사용하는 By 클래스를 임포트합. 웹 요소를 검색하는데 사용.
     from selenium.webdriver.common.keys import Keys #기보드 입력 제어를 위해 Keys 클래스 임포트
    from selenium.webdriver.chrome.service import Service Schrome 드립이버 서비스를 사용하기 위한 모듈 임포트
from selenium.webdriver.chrome.options import Options #Chrome 드립이버 옵션을 설정하기 위한 클래스 임포트
from webdriver_manager.chrome import ChromeOriverHanager #Chrome 드립이버를 자동으로 설치 및 관리하는데 사용되는 드립이버 메니저 S
     from wordcloud import WordCloud, STOPWORDS
     from PIL import Image
     import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
 2) 셀레니움 옵션지정
     myOption = Options() #옵션객체 변수지정
     myOption.add_argument("--start-maximized") #크롬 드라이버 창 최대화
     myOption.add_argument("--incognito") #크롬 드라이버 시크릿모드로
     myOption.add_experimental_option("excludeSwitches", ["enable-automation"]) #드라이버 시작시 불필요문구 미표시되도록 설정
     myOption.add_experimental_option("excludeSwitches", ["enable-logging"]) #테미널상의 불필요문구 미표시되도록 설정
     myOption.add_experimental_option("detach", True) #드라이버 자동꺼짐 방
 3) 셀레니움 드라이버 실행
     txtList.append('''Korean Alcoholic Drinks: A Beginner's Guide
     Like other east Asian cultures, the consumption of alcohol is a practice dating back thousands of years in Korea. However, it w
     Today, let's take a tour of the best that Korean liquor, wine, and beer have to offer! Whether you're looking for a sweet Korea
     Makgeolli
     Makgeolli can be thought of as a "raw" or "original" rice wine. If you've heard of takju before, it was likely in reference to
     Soju is likely the most famous alcoholic beverage produced in Korea, strongly associated with drinking culture on the peninsula
     Rice wines hold a special place in South Korean drinking culture. Gukhwaju - also known as flower wine - is a traditional rice
     [2], CC BY-SA 2.0, via Wikimedia Commons
```

```
(1).ipynb • engliquonipynb ×
- 코드 + Markdown | ⊳ 모두 실행 拱 출력 모두 지우기 | 🚞 개요 ↔
                     ma_img = np.array(Image.open("C:/Users/tnseh/Downloads/liqourBottle.png"))
                     notUseText = ['S','re','korean','diffrent','alcohol'] #불용에 단이 추가
                      stopDict = set(STOPWORDS)
                      for n in range(len(notUseText)):
                               stopDict.add(notUseText[n])
                       myWordCloud = WordCloud(background_color="white", #워드크라우드 배경
                                                                                                relative_scaling=0.3, #단메 상대적 크기 조정
                                                                                                  max_words = 2000, #워드크라우드 표시 단어 최대 수
                                                                                                mask = ma img. #위
                                                                                                stopwords = stopDict, #불용어 단어 지정
                                                                                                  colormap='viridis', #칼라맵 사용
                                                                                                  font_path='c:/Users/tnseh/Downloads/BMHANNAPro.ttf',
                     myWordCloud = myWordCloud.generate(ma text)
                     myWordCloud
                    plt.figure(figsize=(6,6))
                     plt.imshow(myWordCloud, interpolation="bilinear")
                    plt.axis("off")
                    plt.show()
                                                      South Manual Property Name of the Party Name of 
                                                                                                                                     Spirit Province
                                                                                                                                           South Korea
                                                                     Cent TO
                                                                 drink
```

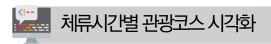


```
■ inPeople.ipynb ●
    8년도 품목별 데이터 TOP3 (1).ipynb ● 📗 engLiquor.ipynb
sython > toy Project > 🛢 inPeople.ipynb > 🌵 #라이브러리 호출
#라이브러리 호출
       import streamlit as st
       import numpy as np
       import pandas as pd
       import plotly
       import plotly.graph_objects as go
      #데이터 호출 및 페이지 세부설정
      st.title('연간 입항 외국인 선원 추이')
      st.markdown('''
      busan_sailors_df=pd.read_excel('data/항구별 출입항 선원수.xlsx',
                                  header=[0,1,2],index_col=0)
      year_list=list(range(2009,2022))
      col_list=[]
      for i in year_list:
          col=busan_sailors_df[str(i)][('입항','선원수 (명)')][1]
          col_list.append(col)
      busan_sailor_num_df=pd.DataFrame(col_list,columns=['입항 선원수(명)'],index=year_list)
      busan_sailor_num_df=busan_sailor_num_df.reset_index().rename(columns={'index':'년도'})
      def ts_plotly_bar_gen():
          fig=go.Figure()
          fig.add_trace(go.Bar(x=busan_sailor_num_df['년도'],y=busan_sailor_num_df['입항 선원수(명)'],
                            text=busan_sailor_num_df['입항 선원수(명)'],textposition='outside'))
          fig.update_layout(title='연간 외국인 선원 수 ',
                          hovermode='x',
                        xaxis=dict(title='년도'),
                        yaxis=dict(title='연간 외국인 선원 수(단위:명)'))
          return fig
      st.markdown('''
                 *#### 20년도 급감 -> 코로나 이슈로 유추할 수 있음
##### 21년도 이후 자료가 없는 점이 아쉬움
      st.plotly_chart(ts_plotly_bar_gen())
```

5. 부록: 코드

부동산 공실데이터 파악

```
■ inPeople.ipynb ●
 /thon > toy_Project > 🛢 inPeople.ipynb > 🏚 #라이브러리 호출
- 코드 + Markdown | D> 모두 실행 🕳 출력 모두 지우기 | 🏣 개요 🚥
      import streamlit as st
      import numpy as np
      import pandas as pd
      import plotly.graph_objects as go
      #데이터 호출 및 페이지 세부설정
      st.title('연간 입항 외국인 선원 추미')
      st.markdown('''
      busan_sailors_df=pd.read_excel('data/항구별 출입항 선원수.xlsx',
                                 header=[0,1,2],index_col=0)
      year_list=list(range(2009,2022))
      col_list=[]
      for i in year_list:
         col=busan_sailors_df[str(i)][('입항','선원수 (명)')][1]
         col_list.append(col)
      busan_sailor_num_df=pd.DataFrame(col_list,columns=['입항 선원수(명)'],index=year_list)
      busan sailor num df=busan sailor num df.reset index().rename(columns={'index':'년도'})
      def ts_plotly_bar_gen():
         fig=go.Figure()
         fig.add_trace(go.Bar(x=busan_sailor_num_df['년도'],y=busan_sailor_num_df['입항 선원수(명)'],
                            text=busan_sailor_num_df['입항 선원수(명)'],textposition='outside'))
         fig.update_layout(title='연간 외국인 선원 수 ',
                         hovermode='x',
                        xaxis=dict(title='년도'),
                        yaxis=dict(title='연간 외국인 선원 수(단위:명)'))
         return fig
      st.markdown('''
                 ##### 20년도 급감 -> 코로나 이슈로 유추할 수 있음
                ##### 21년도 이후 자료가 없는 점이 아쉬움
      st.plotly_chart(ts_plotly_bar_gen())
```



```
■ osmax 추천코스 시각화jpynb ● ■ inPeople.ipynb ● ■ 2018년도 품욕별 데이터 TOP3 (1).ipynb ● ■ 부산항 기준 창고 시각화 (1).ipynb ●
C. > Users > Lenovo > Downloads > ■ osmnx 추천코스 사각화(pynb > ■ 제류 5시간 추천코스(5개 경로) > ■ 코스: 부산신항 - 관광지 - 맛집 - 부산신항 > ■ 부산신항 - 해몽용공사 - 대하가아밀면 - 부산신항(c.
+ 코드 + Markdown │ ▶ 모두 실행 ➡ 출력 모두 지우기 │ ☱ 개요
    5. 부산신항 - 오륙도스카이워크 - 호렌소 - 부산신항(2시간 16분)
  1. 부산신항 - 감천문화마을 - 진미식당 - 부산신항(2시간 26분)
      busan = "부산광역시, 대한민국"
      targetPoint = ox.graph_from_point(point, network_type="drive",dist=15000)
      port_Point=ox.distance.nearest_nodes(targetPoint,point[1],point[0]) #∓살랑 좌표
       tour_Point = ox.distance.nearest_nodes(targetPoint, tour[1], tour[0]) #含含河 多里
      route2 = nx.shortest_path(targetPoint,tour_Point,food_Point) #작공자 - 맛집 최단 정로
      route3= nx.shortest_path(targetPoint,food_Point,port_Point) #한집- 부산광 최단정로
      ox.plot_graph_routes(targetPoint,[route1,route2,route3],node_size=0.5, edge_linewidth=0.5, edge_color='white',route_colors=['red','blue','yellow'])
    (<Figure size 800x800 with 1 Axes>, <Axes: >)
```

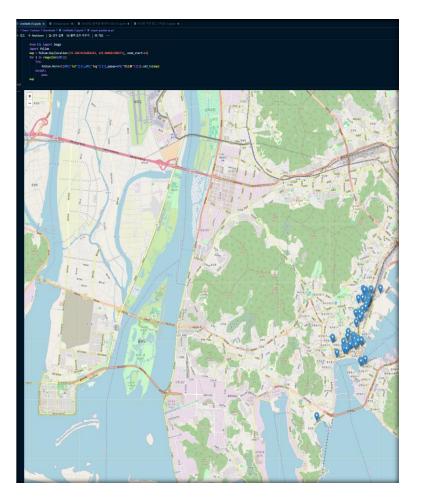


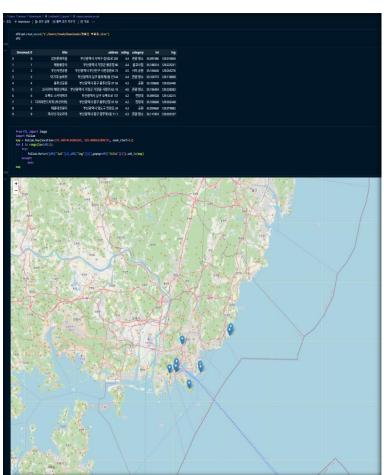
▲ 부동산 공실데이터 크롤링 및 시각화 코드

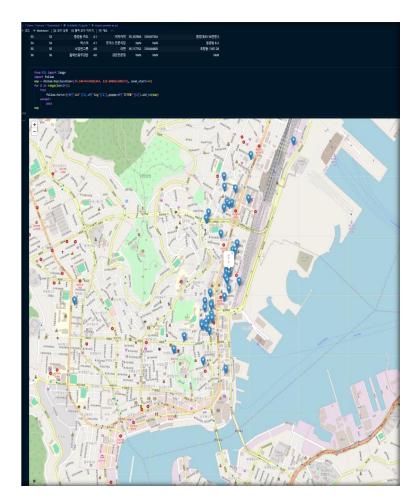
▲ 체류시간별 시각화 코드

5. 부록 : 코드

💹 부산 맛집, 관광지, 숙박 업소 위치 파악







▲ 부산 맛집, 관광코스, 숙박업소 위치 표시

