

Contents.

- 1 전처리
- 2 문제해결

Part 1

전처리

1

엑셀에서 필드추가

2

태블로에서 필드추가

3

파이썬에서 필드추가

Part 1, 전처리 - 1.엑셀에서 필드추가

Q	R
이전Lon	이전Lat
126.9212	37.55611
126.92101	37.55607
126.92084	37.55611
126.92082	37.55724
126.9195	37.55757
126.919	37.55759
126.91895	37.55824
126.91845	37.55974
126.91713	37.56123
126.91571	37.56272
126.91436	37.56353
126.91356	37.56404
126.91307	37.5649
126.91228	37.56524
126.91215	37.56577
126.9136	37.56687
126.91618	37.56773
126.91718	37.5681




이전Lon
이전Lat

MAKEPOINT에 사용하기위함.
MAKEPOINT에는 공간데이터를 사용하기때문에,태블로에서테이블
함수인 LOOKUP 참조가불가.
따라서엑셀에서 [Seq]순서로 정렬한뒤,바로 위 튜플의 [Lon], [Lat] 참조

NULL

당연히,각 [영업일]에대한첫 튜플에서는이전 [Lon],[Lat]이 존재할수 없음.
따라서이자리는 Null값으로 비움.

Part 1, 전처리 - 2. 태블로에서 필드추가

 계산 이전위치Point	 계산 현재위치Point	 계산 이동거리
<i>null</i>	Point	<i>null</i>
Point	Point	125.46
Point	Point	9.71
Point	Point	119.27
Point	Point	70.68
Point	Point	43.29
Point	Point	69.79
Point	Point	11.49
Point	Point	128.10

이전Point
현재Point

MAKEPOINT([이전Lat],[이전Lon])

앞에서 만든 [이전Lat]과 [이전Lon] 필드로 이전 위치의 공간데이터 생성.

MAKEPOINT([Latitude],[Longitude])

[Latitude], [Longitude] 필드로 해당 위치의 공간데이터 생성.

문제 1,2,3 해결을 위함

이동거리

DISTANCE([이전위치Point],[현재위치Point],'m')

앞서 만든 두 포인트 사이의 거리를 계산. (m단위)

문제 1,2,3 해결을 위함.

 계산

계산

20분 분할

7:40:00

7:40:00

8:20:00

20분
단위분할

STR(DATEPART('hour',[Date Time]))+'-'+

STR(DIV(DATEPART('minute',[Date Time]),20)*20)+'-'+00'

[Date Time]의 시간부분을 DATEPART로 가져오고,

분은 가져온 뒤, DIV를 통해 20으로 나눈 결과의 정수부분을 가져온 뒤, 20을

곱해, 20분 단위로 나눌 수 있도록함. 문제 3 해결을 위함.

Part 1, 전처리 - 2. 태블로에서 필드추가

=# 계산	=# 계산
Lon 100m	Lat 100m
126.921000	37.5560000
126.916000	37.5610000
127.025000	37.6030000

Lon 100m
Lat 100m

`ROUND([Longitude],3)` `ROUND([Latitude],3)`
3째자리까지 반올림표시를하여, 격자단위 필드 생성

=Abc 계산
그리드 100m
37.555999999999997_126.9...
37.561_126.916
37.6030000000000002_127....

그리드

`STR([Lat 100m]+'_' +STR([Lon 100m]))`
앞서 만든 두 포인트사이의거리를 계산. (m단위)

=# 계산
그리드 100m 고유개수
1
1
1

그리드
고유개수

`COUNTD([그리드 100m])`
앞서 만든 그리드100m의 고유갯수를 구하기 위함.
문제 4 해결을 위함.

Part 1, 전처리 - 3.파이썬에서 필드추가(행정동)

함수정의 엑셀파일 불러오기

#위도와 경도를 받아 행정동을 리턴해주는 함수

```
def lat_lon_to_addr(lon,lat):  
    url = 'https://dapi.kakao.com/v2/local/geo/coord2regioncode.json?\  
        x={longitude}&y={latitude}'.format(longitude=lon,latitude=lat)  
    headers = {"Authorization": "KakaoAK " + my_API_key}  
    result = json.loads(str(requests.get(url, headers=headers).text))  
    geo = result['documents'][0]['address_name']  
    return str(geo)
```

#엑셀파일 불러오기

```
dataset = pandas.read_excel('Taxi_운행기록.xls')
```

Abc
taxi
행정동
서울특별시 마포구 동교동
서울특별시 마포구 성산동

반복문을 통해 지번구하기 덮어쓰기로 저장

#8145개의 데이터에 대해 행정동을 구함.

```
for i in range(0,8145):  
    print(i+1,'processing')  
    latitude = dataset["Latitude"]  
    longitude = dataset["Longitude"]  
    result = lat_lon_to_addr(longitude, latitude)  
    dataset["지번"] = result
```

#api 차단방지를 위한 time 지연
sleep(2)

#새롭게 저장

```
path = 'Taxi_운행기록.xlsx'
```

```
with pandas.ExcelWriter(path) as writer:
```

```
    dataset.to_excel(writer, sheet_name = 'taxi') #taxi 시트에 저장
```


Part 2

문제해결

Part 2, 문제 해결

STEP 1

4회의 영업중, 운행 거리 합계가 가장 큰 영업일과 가장 작은 영업일 구하기

>>

STEP 2

거리와 시속의 시간대별 시각화를 이중축을 사용하여 만들어보기

>>

STEP 3

문제 2의 시간시각화를 20분대로 분할하여 시간시각화하기

>>

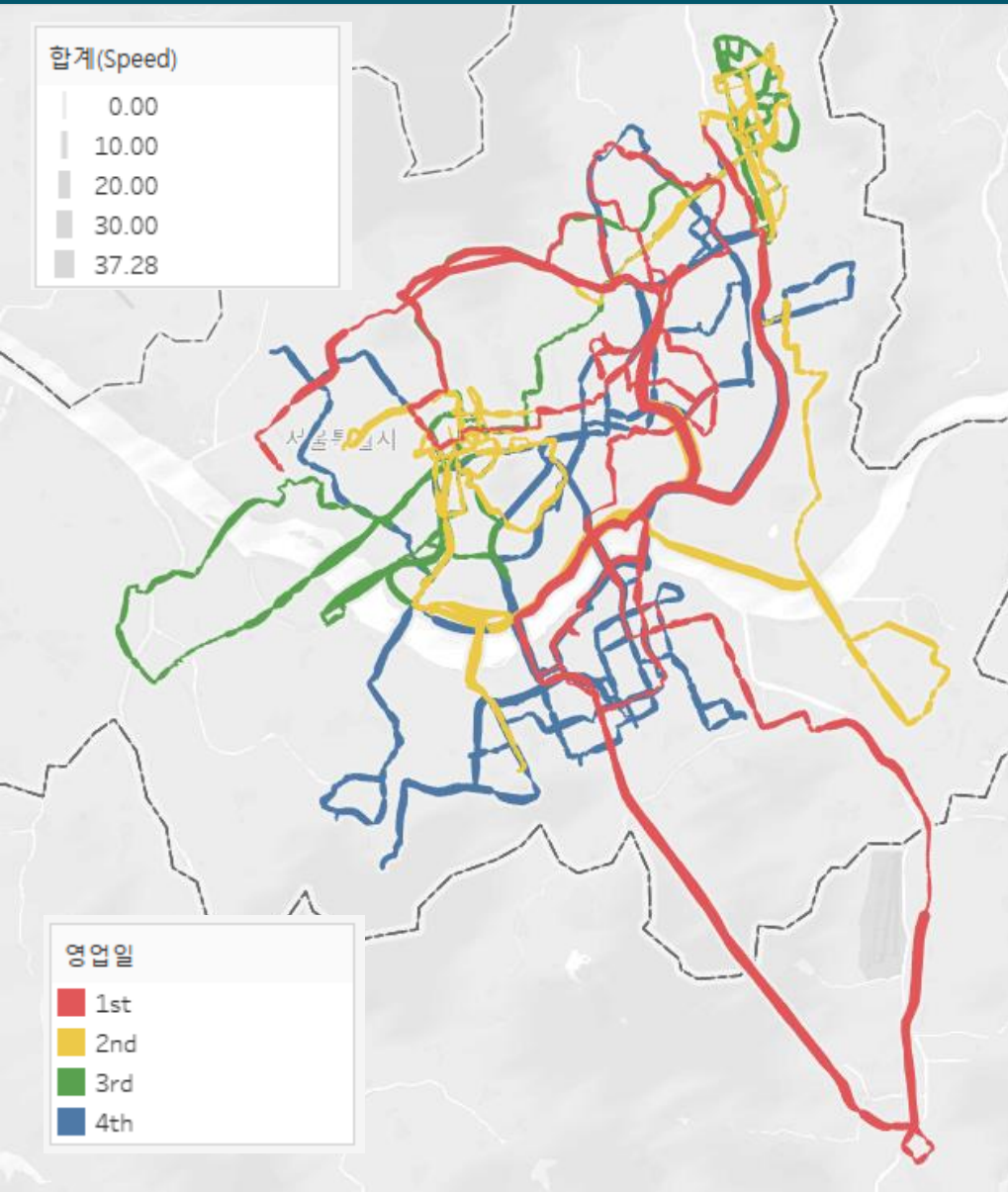
STEP 4

첫 영업일(수요일)의 셀 개수와 고유 개수, 그리고 100m 셀 개수와 고유개수를 각각 계산하기

STEP 5

택시가 방문한 행정동의 개수 계산하기

Part 2, 문제 해결 - Preview

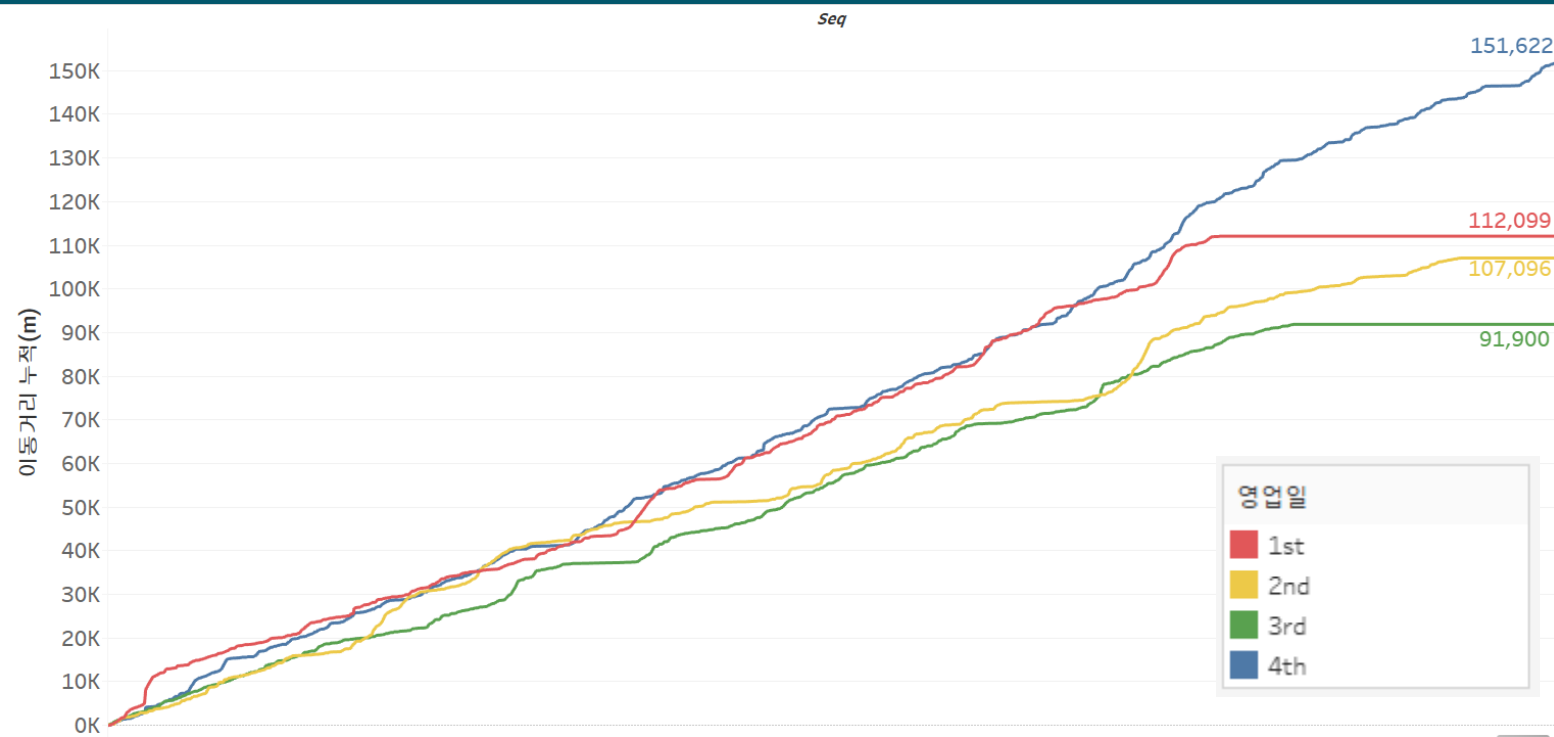


분당까지 갔던 1일차가 가장 운행거리가 크지않을지?

선이 두꺼운 곳, 즉 [Speed]가 높게 나오는 곳은
고속도로인 곳이 많음.

전체적으로 선이 고르게 굵게 나오는 편인 4일차는 도로에 차
가 많지 않은 야간 시간대라고 유추할 수도 있음.

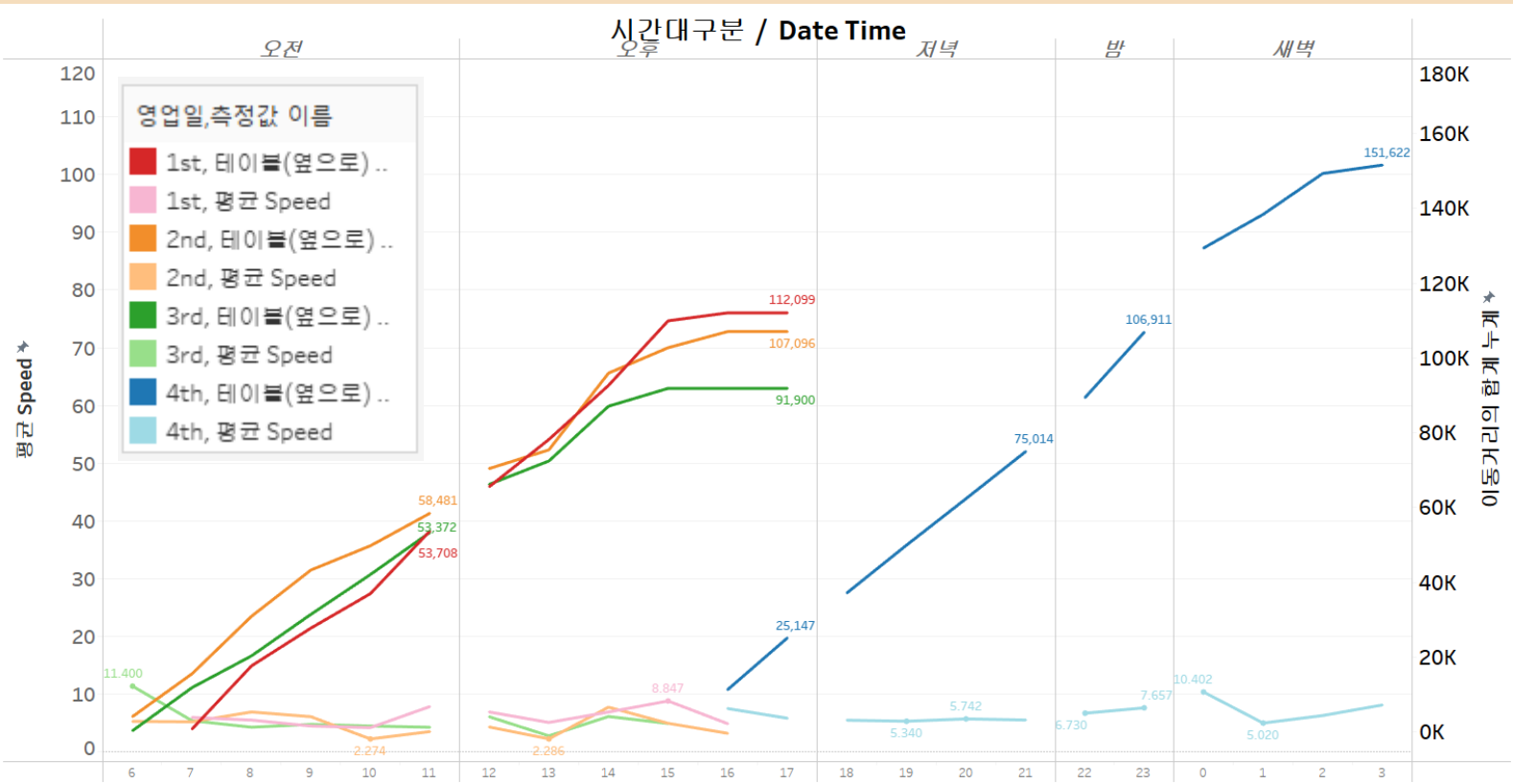
Part 2, 문제 해결 - STEP 1



[이동거리]값을 [Seq]에 따라 누적해간 값으로 그래프로 그림
1,2,3일차의 더 이상 증가하지 않는 부분은
영업이 끝난 시점임.

4일차가 운행거리합이 가장 크고 3일차가 가장 작다

Part 2, 문제 해결 - STEP2



이번에는 [Seq]가 아닌 [시간대구분] / [Date Time.hour] 기준으로 [이동거리] 누적 그래프와 별 평균 [Speed]를 그래프로 나타냄.
각 [시간대구분]별 최대값과 최소값을 레이블링

iii

시간대구분

田

시간(Date Time)

≡

평균(Speed)

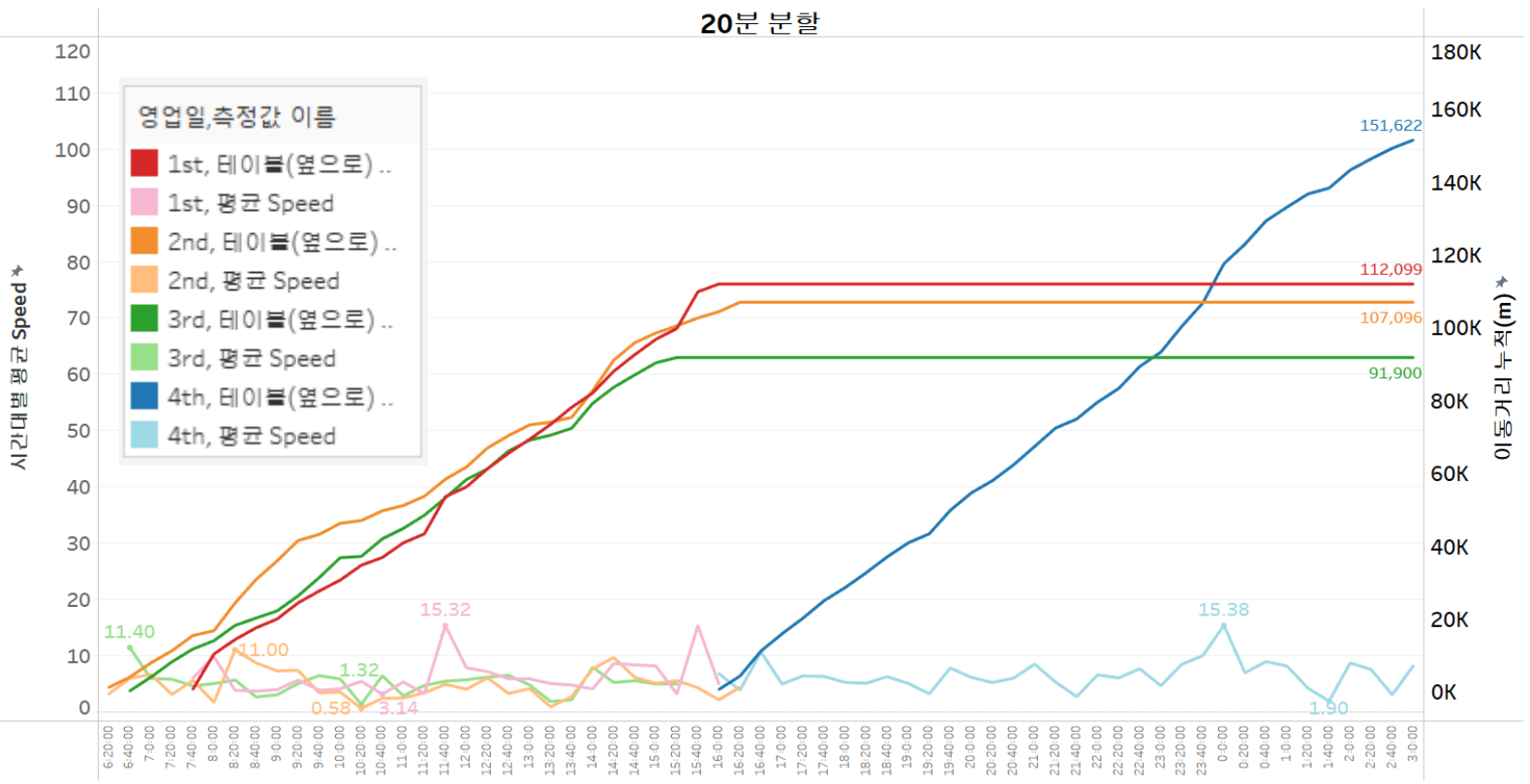
합계(이동거리)

△



4일차는 늦은 오후부터 새벽까지 운행.

Part 2, 문제 해결 - STEP 3



이번에는 Part1에서 미리 구해둔 [20분 분할] 필드기준으로 [이동거리] 누적 그래프와 평균 [Speed]를 그래프로 나타냄.

각 측정값별 최대값과 최소값을 레이블링

iii

20분 분할

≡

iii

평균(Speed)

합계(이동거리)

△



4일차는 늦은 오후부터 새벽까지 운행.

Part 2, 문제 해결 - STEP 4

그리드 셀 고유개수

영업일	그리드 1km의 카운트	그리드 1km 고유개수	그리드 100m의 카운트	그리드 100m 고유개수
1st	1,778	148	1,778	973

측정값

카운트(그리드 1km)

집계(그리드 1km 고..

카운트(그리드 100m)

집계(그리드 100m 고..

필터

측정값 이름

영업일: 1st

첫째날 기준 그리드의 셀개수와 고유개수 계산

그리드의 카운트(개수)는

`COUNT([그리드 100m])`,

그리드의 고유개수는

`COUNTD([그리드 100m])`임.

그리드의 개수는 **[Seq]**의 **최대값**과 같으므로
1km, 100m 그리드의 카운트가 같은건 당연한 사실

Part 2, 문제 해결 - STEP 5

행정동 고유개수

영업일	행정동의 카운트	행정동의 고유 카운트
1st	1,778	128
2nd	2,168	146
3rd	1,893	131
4th	2,306	134

필터

측정값 이름



영업일

측정값

카운트(행정동)

카운트(고유)(행정동)

지나간 행정동의 개수와 고유개수 계산

행정동의 카운트(개수)는

COUNT([행정동]),

행정동의 고유개수는

COUNTD([행정동])임.

Part1 에서 카카오API 를 이용해 구한 [행정동] 필드 이용