TAXI DATA ANALYSIS

Constructing model to simulate a taxi system & Data processing

Index

- Introduction
- Calculating efficiency of taxi system
- Data processing
- Summery
- Future work

Introduction

TAXI DATA

- 서울에 등록되어 있는 택시 수

  ~ 100,000

- 실제 하루동안 운행하는 택시 수

  ~ 66,000

- 데이터 상 시간 간격

  10s

TAXI DATA

- 실제 데이터의 구성:
- 택시 ID, 경도, 위도, z축, 시간, 방향, 속도, 승객 탑승 여부
- 정확도가 높은 데이터 영역:
- 택시 ID, 경도, 위도, 시간, 승객탑승여부

=> 데이터의 질적, 양적 조치 요구됨

WHAT TO PROCESS

우선순위

- 빈번하게 무거운 연산이 필요한 것
- 믿을 만한 데이터일 것
- 데이터의 압축
- 부가적인 요소

트레이드 오프

- 리소스의 제약
- 데이터의 손실
- 데이터 크기의 증가

FREQUENTLY OCCURRED HEAVY OPERATION

DISTANCE

- 거리, 속력
- 데이터상의 포인트 : 경도, 위도

TIME OPERATION

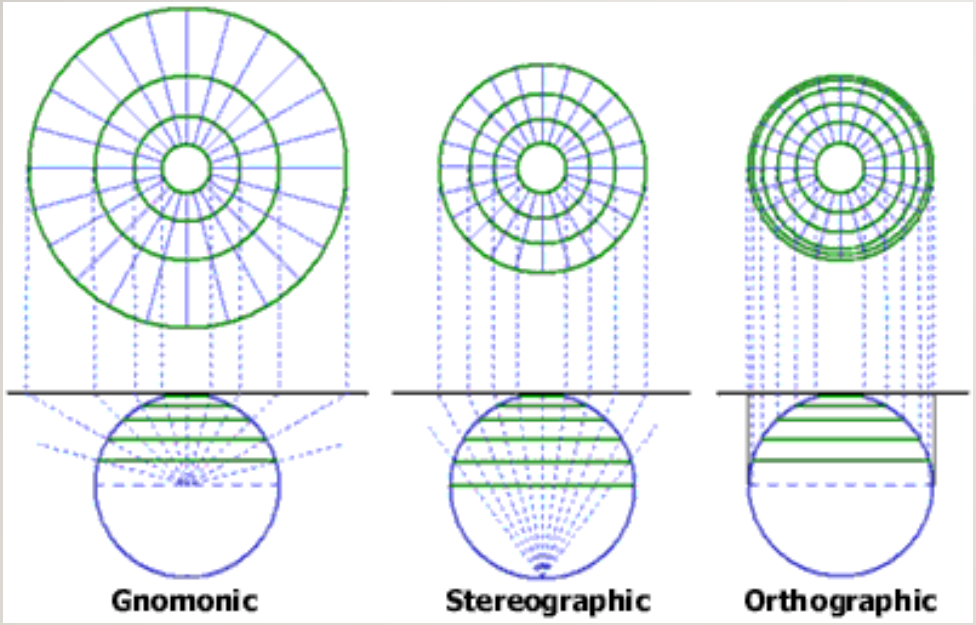
- 데이터상의 시간형식

  (YYYYmmddHHMMSS)

- 시, 분, 초 차이 계산을 계속 보정해 주어야함.

Powered by python

ORTHOGRAPHIC PROJECTION



- 정사영법 : 가장 단순하면서도 중심 근처에서 정확한 2D사영법
- 기준점 : 서울시립대학교 좌표 이용

Powered by python

TIME OPERATION

1. Datetime 모듈을 이용
2. 각 날짜를 정수로 변환
3. 데이터가 10초 간격임을 고려, 10을 나누어 저장
4. 역변환도 같은 모듈을 이용하여 변환함

Powered by python

ADDITIONAL INFORMATION : DISTRICT

- 위치 정보는 이미 모든 정보를 포함하고 있지만 실제로 특정 지역을 조사하기 위해서는 분류가 되어있는 것이 계산 속도를 올릴 수 있다.

- 따라서 모든 데이터를 서울의 구를 기준으로 분류하는 작업을 진행하였다.

Powered by python

DATA DESCRIPTION

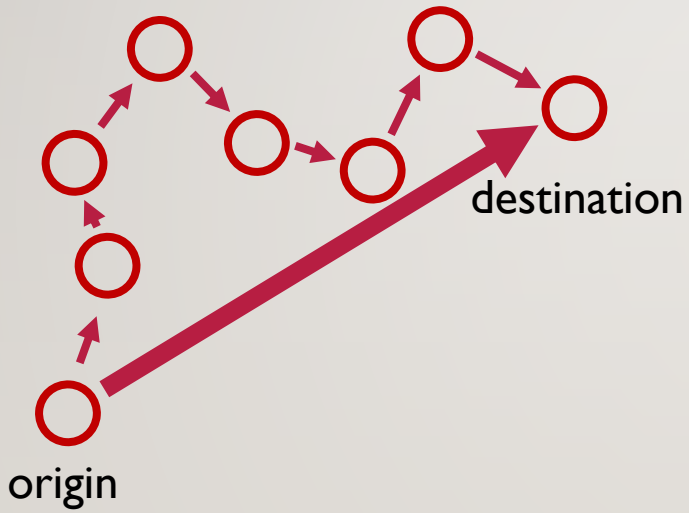
- 위의 과정들을 통하여 데이터를 총 7가지 필드로 구분하였다.

ID	X	Y	Time	Velocity	Passenger	District
택시 ID	동서방향	남북방향	정수 시간	양의 정수	참/거짓	시군구코드

처리 전	처리 후
데이터 크기 : 약 8GB 로드 속도 : 약 8 ~ 10분 필요 연산량 : 높음	데이터 크기 : 약 5GB 로드 속도 : 약 8 ~ 10초 필요 연산량 : 낮음

Powered by python

TRIP DATA



- 승하차 데이터란?

  중간의 데이터들의 정보보다 승객의 탑승과 하차에 초점을 두고 정리한 데이터

- 승하차 데이터의 목적과 의의
  - 데이터의 간소화
  - 보다 직관적인 분석 가능

- 승하차 데이터 필드
  - 택시 id
  - 시작지점(위치, 시간)
  - 목적지점(위치, 시간)

Powered by python

Calculating Efficiency of Taxi System

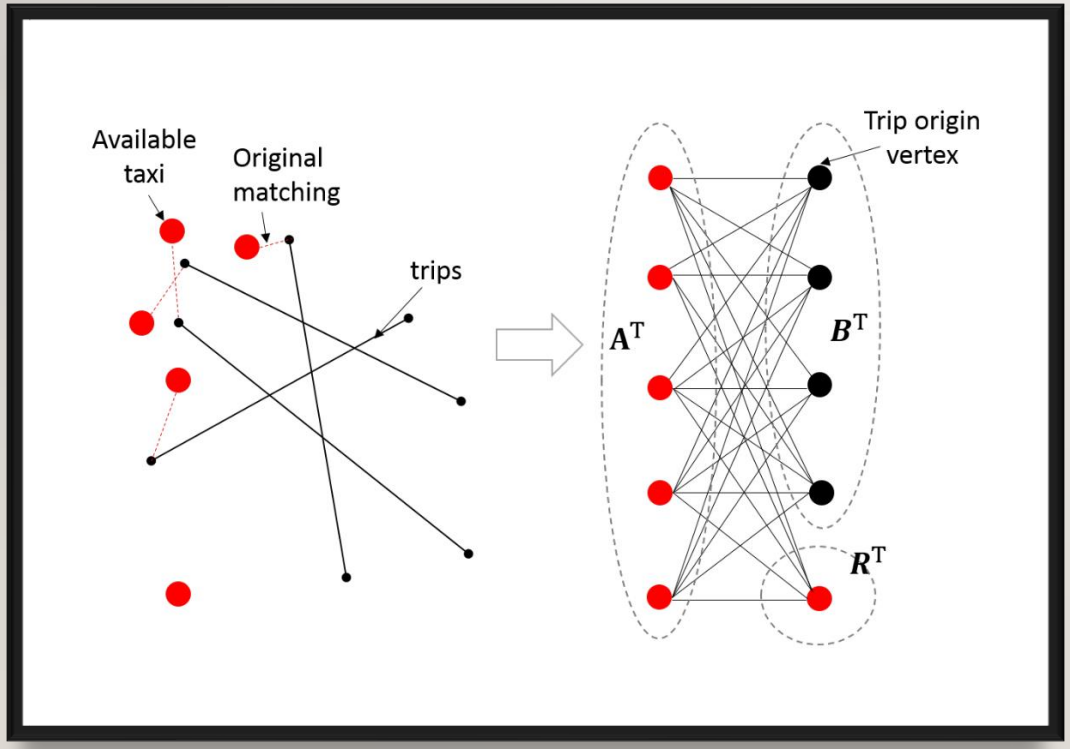
HOW TO MEASURE EFFICIENCY OF TAXI SYSTEM

X.Zhan의 국제 교통공학 워크샵 발표자료에 따르면,

승하차 데이터 ✚ 네트워크 이론

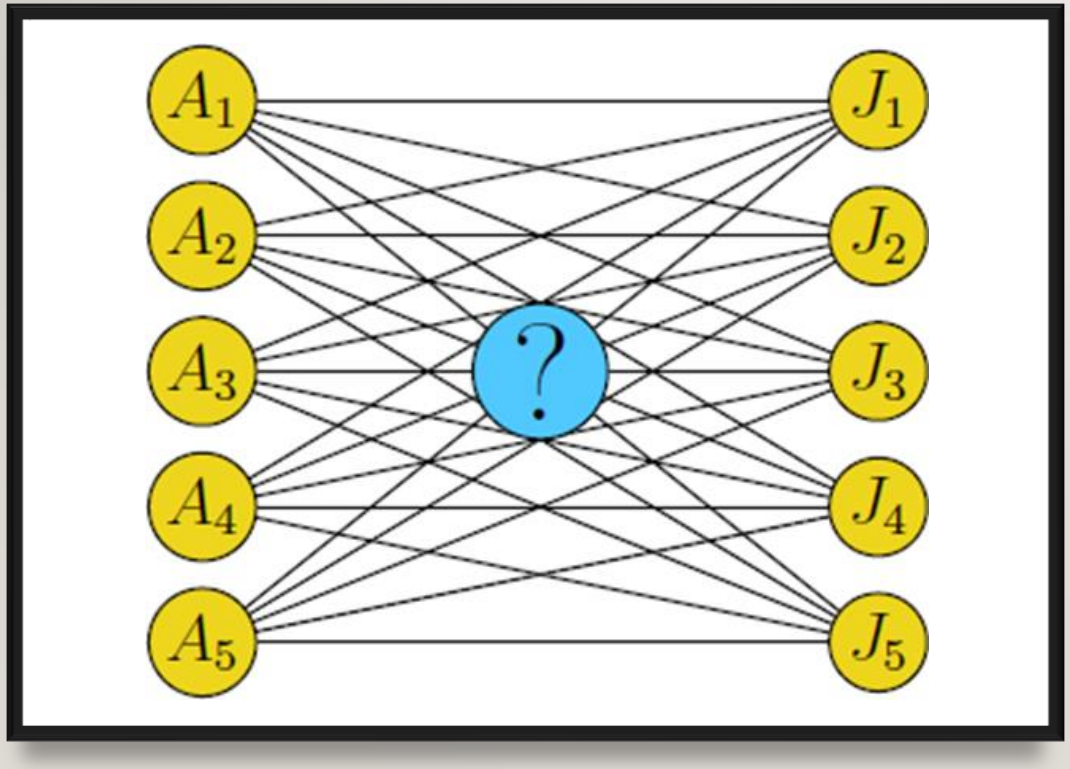
→ 택시 시스템의 효율성 분석 가능!

HOW TO MEASURE EFFICIENCY OF TAXI SYSTEM



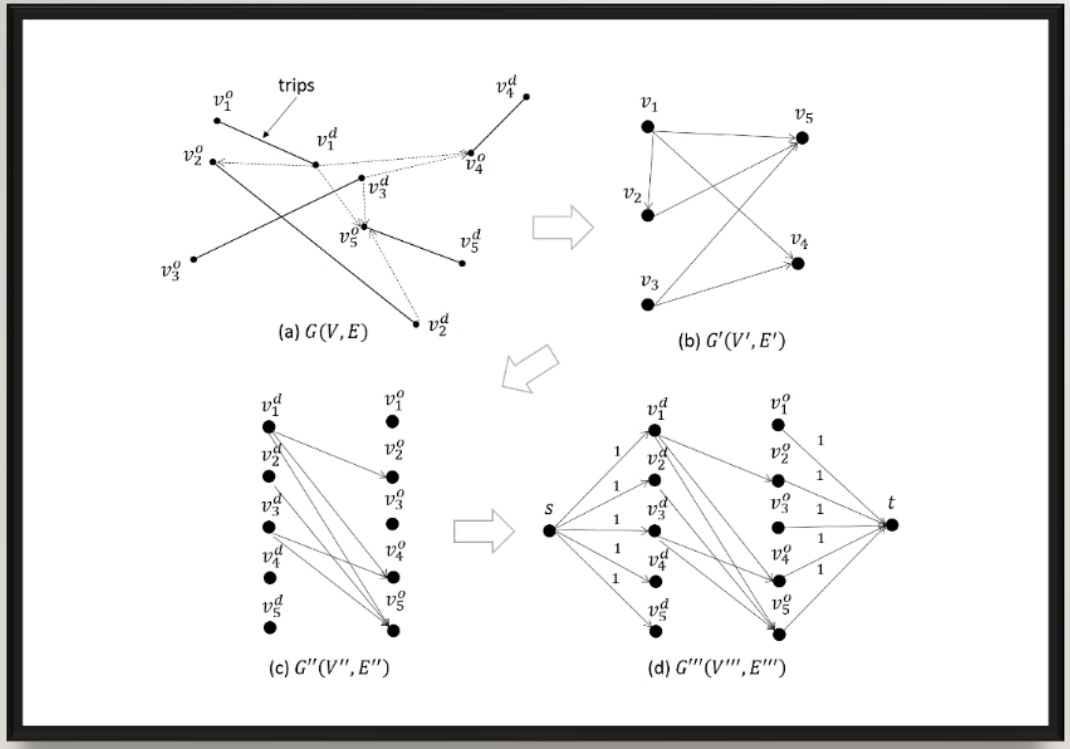
- 특정 시간 ΔT동안,
- 이용가능한 택시들과 기다리는 승객, 그리고 나머지 택시로 구분하면
- Fully connected network을 구성할 수 있음

HOW TO MEASURE EFFICIENCY OF TAXI SYSTEM



- 이러한 문제는 마치 N명의 작업자에게 N개의 일을 수행하는 알고리즘과 일치한다.
- 이를 해결하는 적당한 알고리즘으로 헝가리안 알고리즘이 있다.

TRIP INTEGRATION : UNWEIGHTED TRIP INTEGRATION

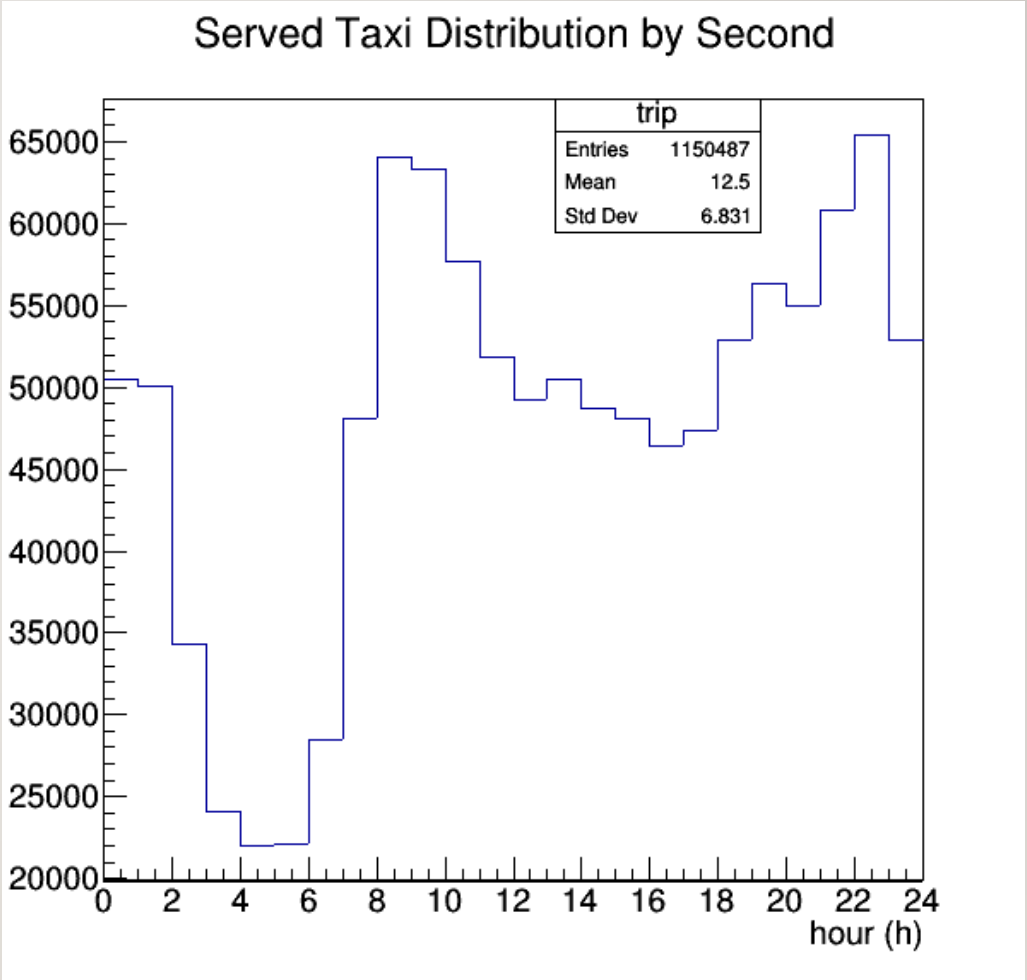


- 이를 해결하는 가장 간단한 방법으로 최대한으로 승하차를 연결시키는 방법이 있다.
- 모든 승객의 탑승은 똑같은 비중을 가지며, 승객과 승객을 가장 많이 연결할 수록 같은 일을 하는데 필요한 택시 수는 줄어든다.

TARGET DATA

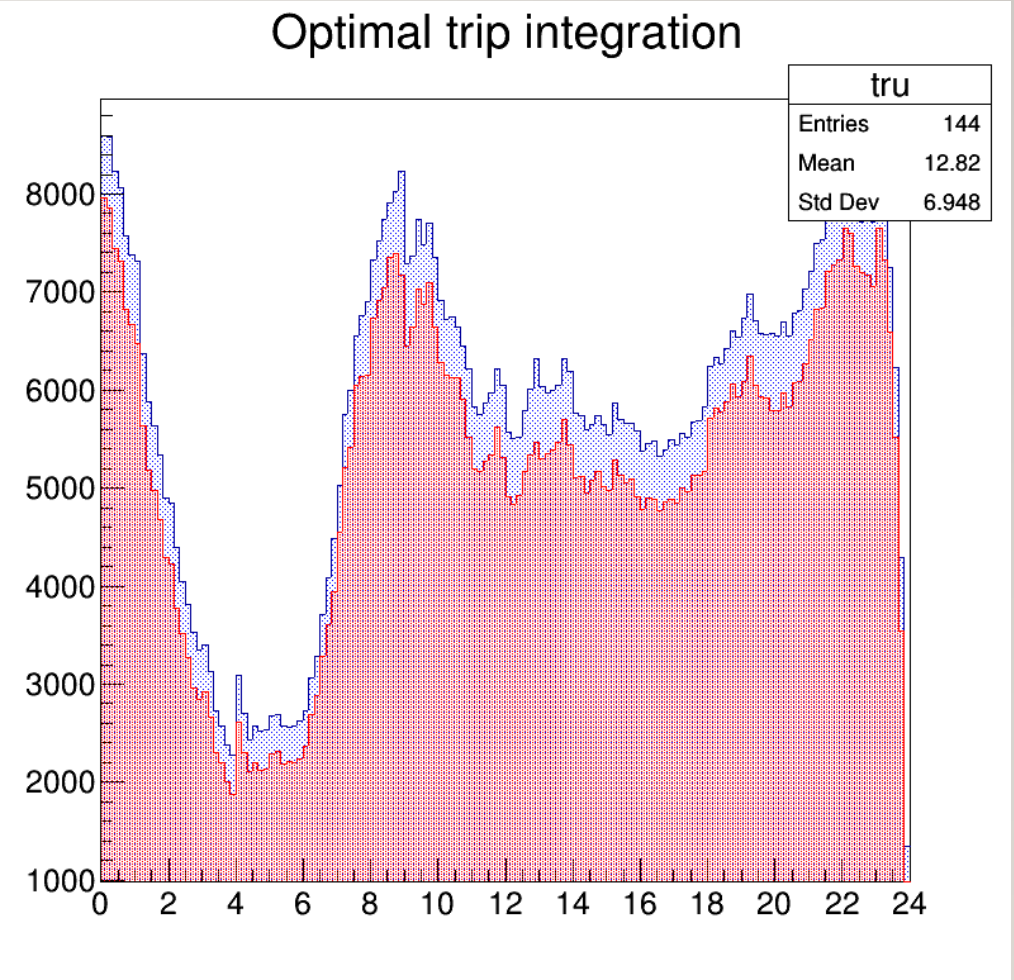
- 2016년 3월 16일 수요일
- 총 택시 이용자수 115만명
- 누적합 단위 시간 : 10분

  -> 144 단위시간 / 일

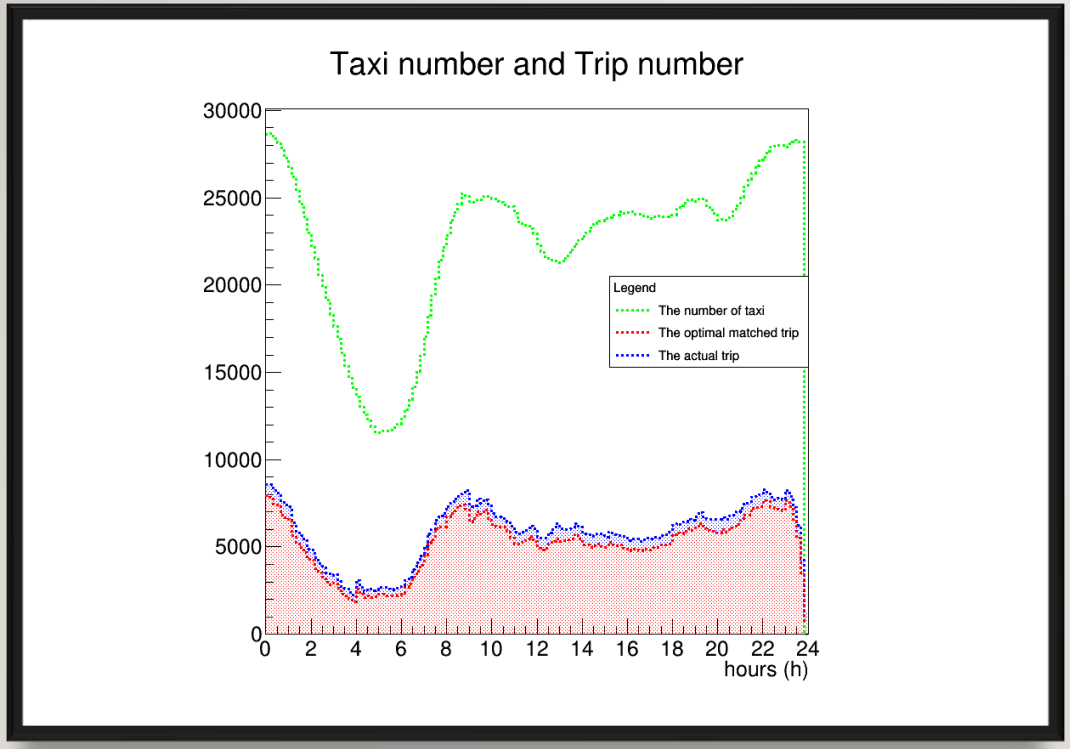


RESULT

- 2016년 3월 16일 수요일
- 약 10% 승객들이 서로 연결될 수 있음을 발견.



SUMMARY



- 정량적 분석을 좀 더 필요로 하지만, 정성적으로는 더 효율적이 체계가 존재한다면, 이론적으로 2016년의 택시 시스템은 보다 적은 택시 수로도 서비스가 가능함을 알 수 있었다.

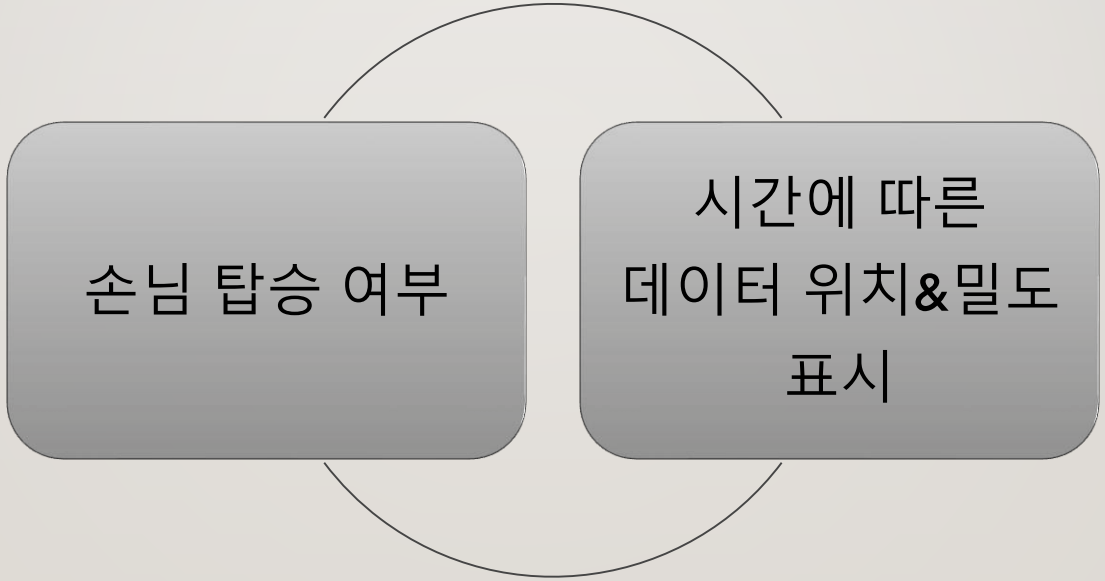
Data Processing for Controlling Big Data

Visualized by H.H. Park

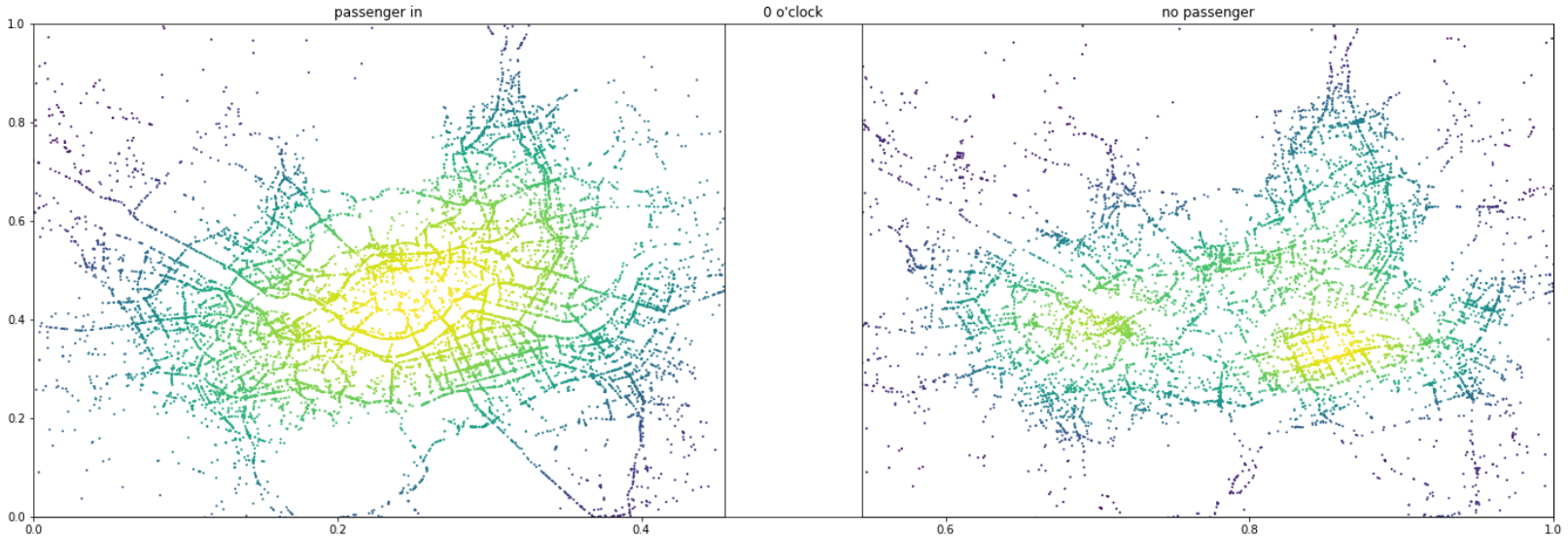
Data Visualization

Visualized by H.H. Park

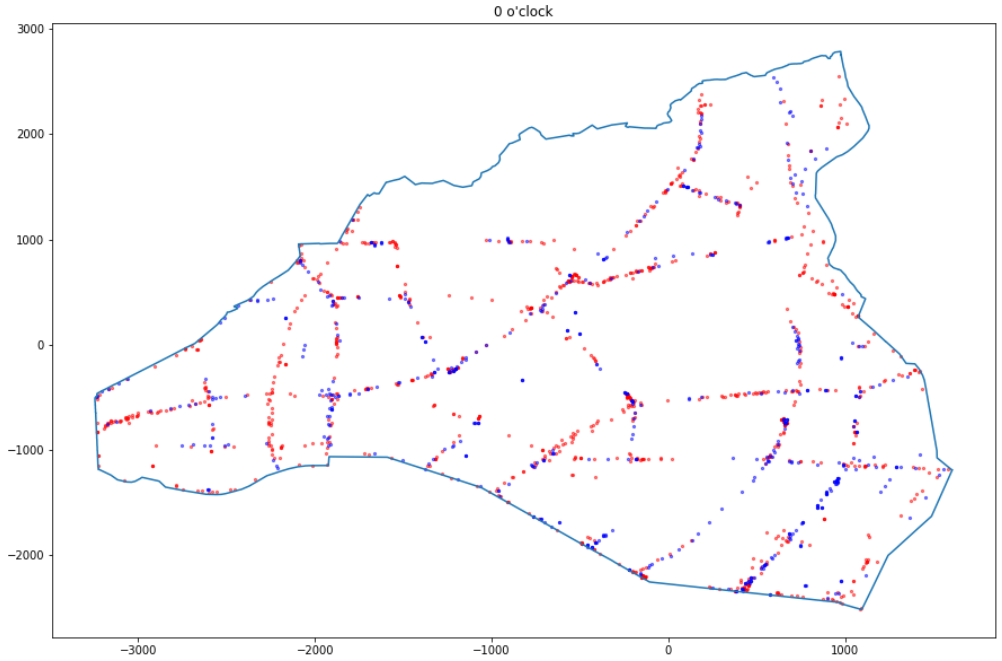
DENSITY MAP



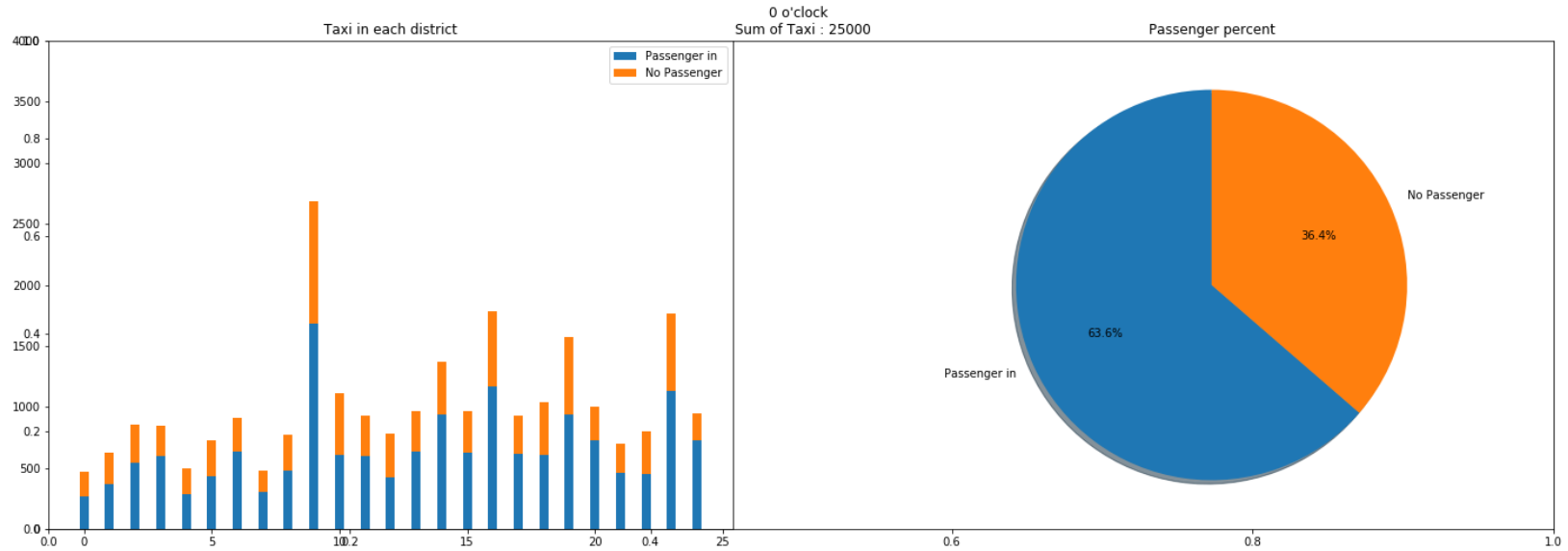
DENSITY MAP


passenger in
0 o'clock
no passenger
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

DENSITY MAP

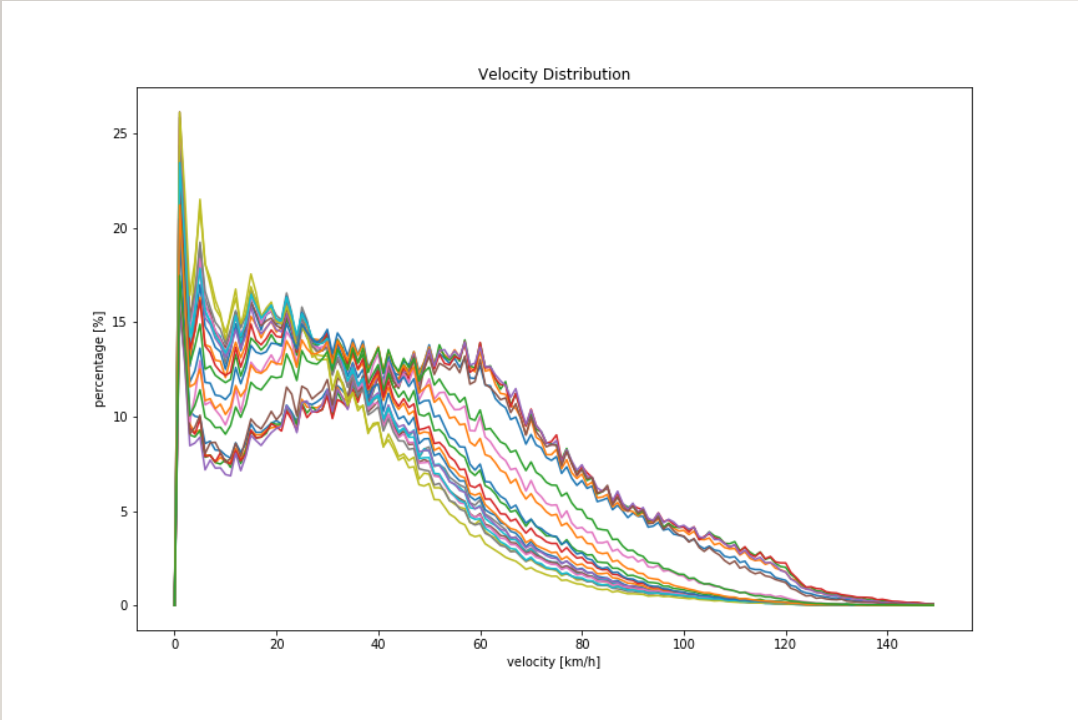


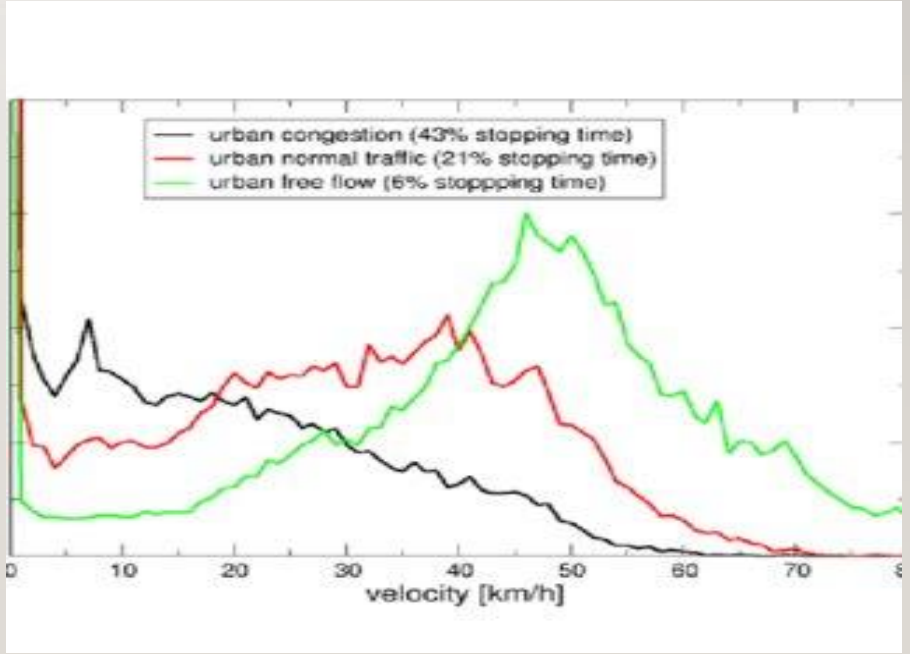
PERCENTIGE GRAPH



FEE CALCULATION

VELOCITY DISTRIBUTION





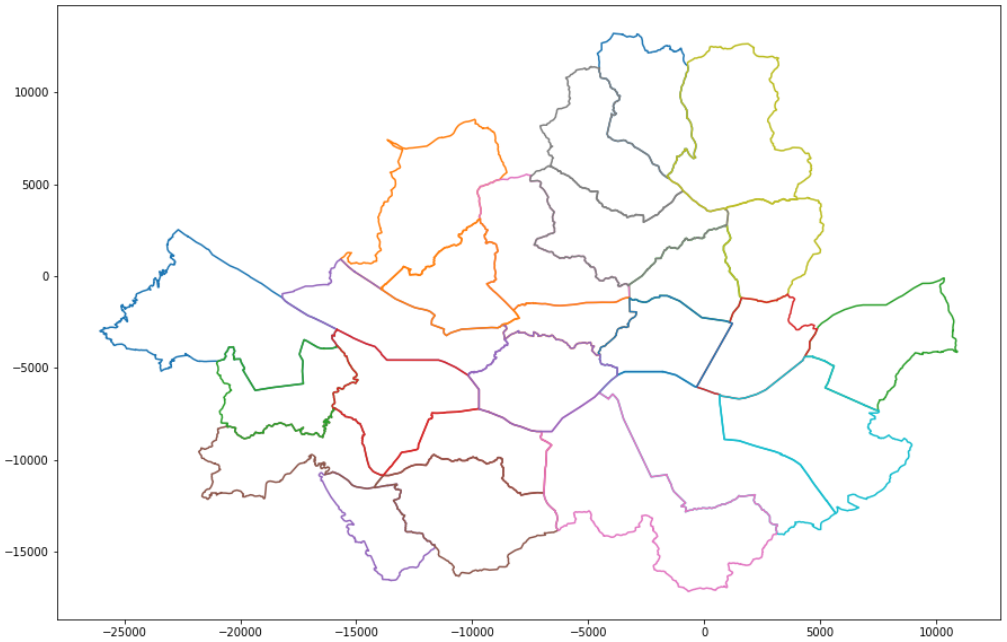
Composition and payload distribution of the on-road heavy-duty fleet in the Netherlands

SUMMARY

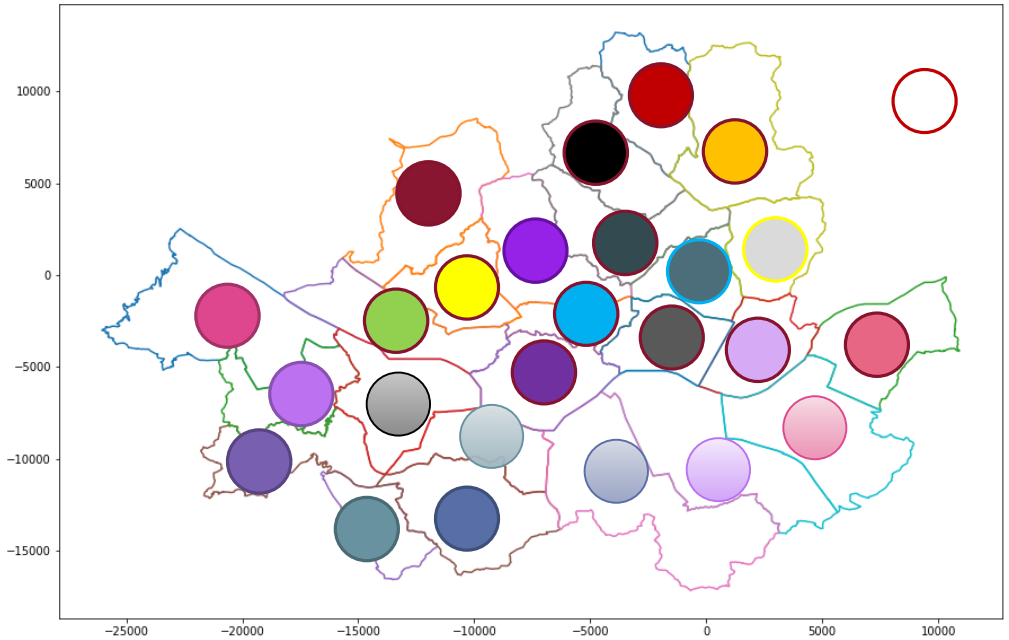
1. 간소화된 데이터를 통한 택시 데이터의 효율성 측정 시도
2. 전체 데이터를 활용하기 위한 준비
3. 간단한 시각화를 통한 데이터의 직관적인 이해

IDEAS FOR TOY TAXI SYSTEM

SEOUL

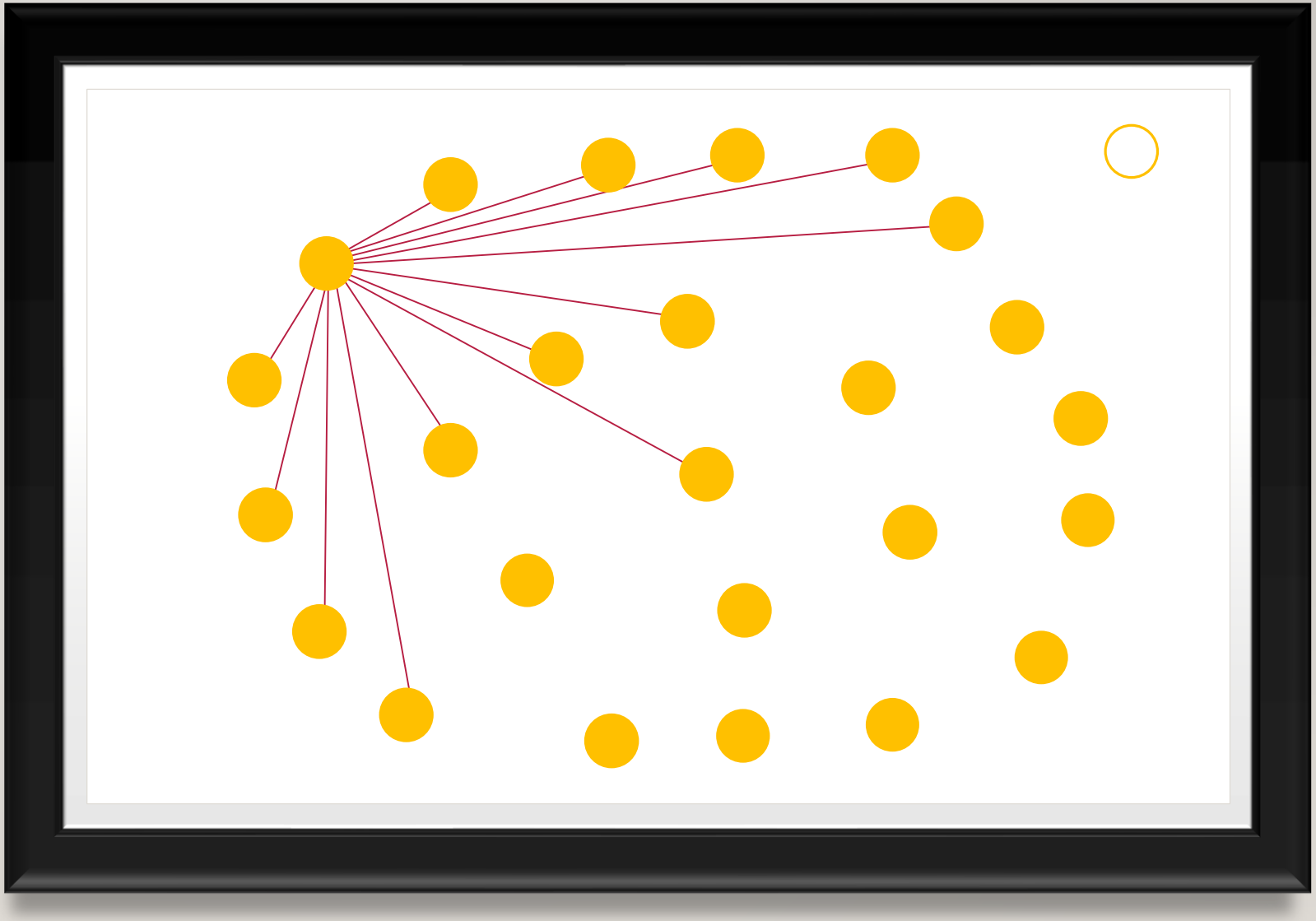

10000
5000
0
−5000
−10000
−15000
−25000
−20000
−15000
−10000
−5000
0
5000
10000

SEOUL NETWORK

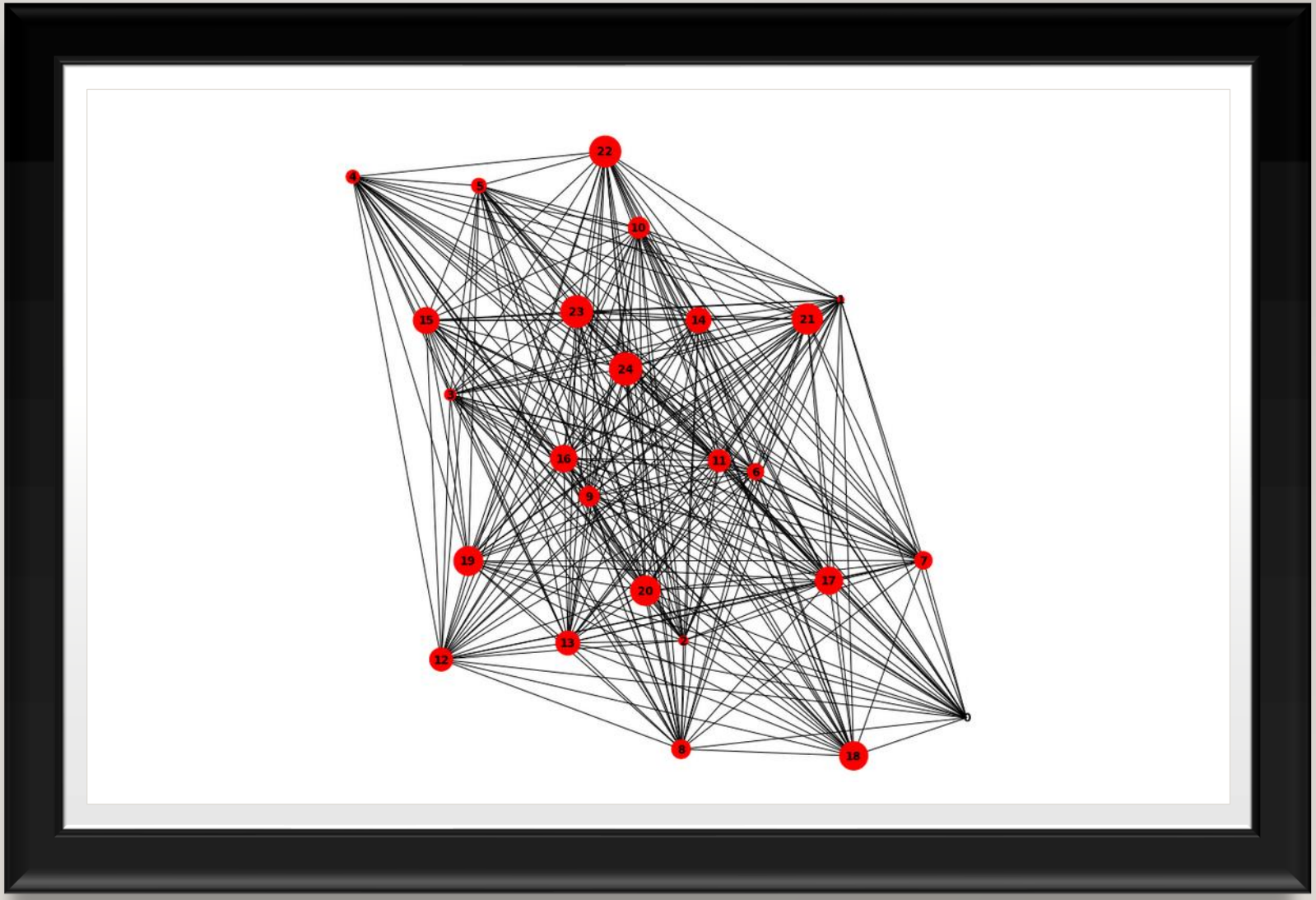


SEOUL NETWORK

SEOUL NETWORK



SEOUL NETWORK



EXTRACTING MODEL'S BASIC PROPERTIES

- With time interval $[T, T + \Delta T]$, let trips from i-th node to j-th node as W_{ij},

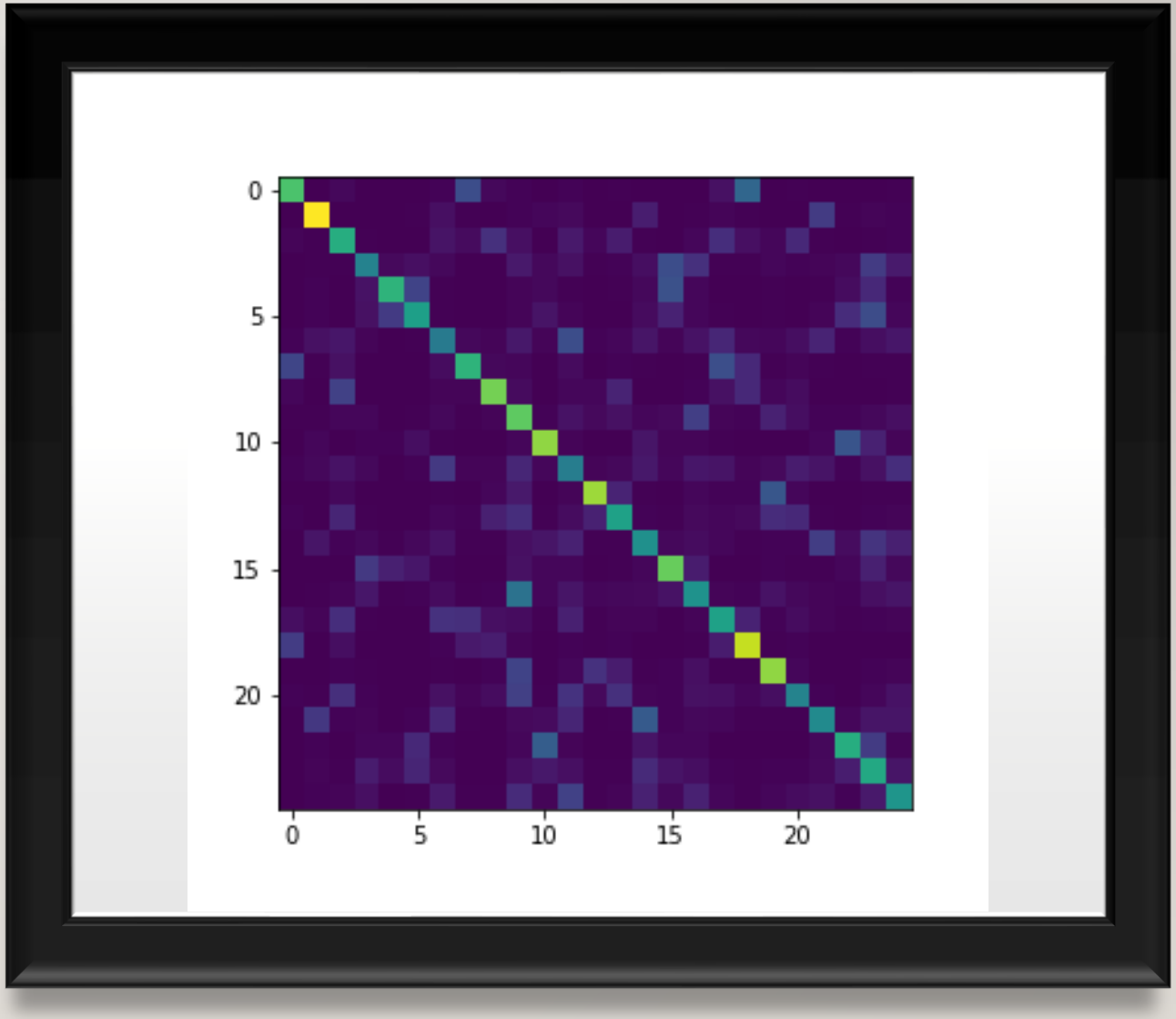
$$W_{ij}(T) = \frac{\Delta N_{i \to j}(T)}{\Delta T}$$

- Then we can get W in the form of matrix.
- And Trip occur probability $P_{ij}^{T}(T)$

$$P_{ij}^{T}(T) \sim \frac{W_{ij}(T)}{\mathcal{N}(T)}, (where\ \mathcal{N}(T) = \sum_{j} W_{ij}\,(T))$$

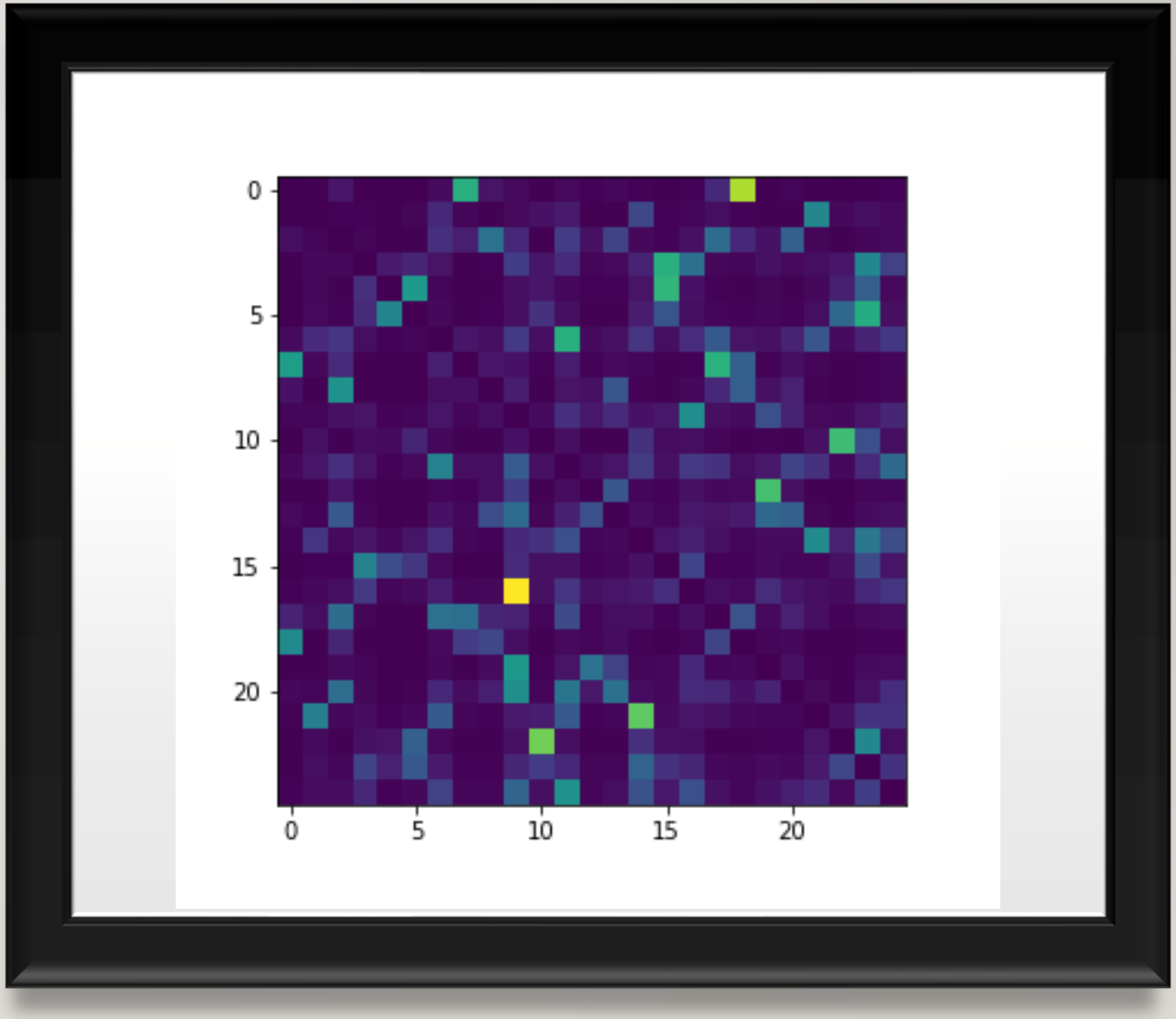
EXTRACTING MODEL'S BASIC PROPERTIES

- Trip occur Probability Matrix



EXTRACTING MODEL'S BASIC PROPERTIES

- Trip occur Probability Matrix (Off-diagonal component)



FUTURE WORK

감사합니다.