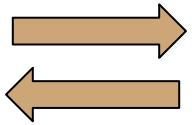
Incorporating Public Transit into Measures of Accessibility

UBCO 2021 Capstone Project



Graham Kerford

Rain Shen

Yuxuan Cui

Luka Vukovic

Joseph Kuchar

Bjenk Ellefsen

Graham

BSc Earth Science, minor Ocean Science,
Dalhousie University

Rain

BSc in Food Science and Technology, UBC

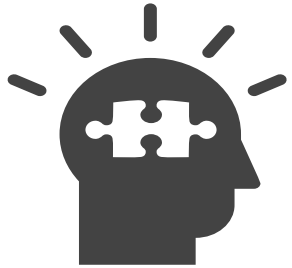
Luka

BSc Biomedical Sciences, uOttawa
Grew up in Ottawa

Yuxuan

BSc in Economics & statistics(Econometrics), SFU

1. Motivation



- 22.3% of commuters use public transport in large Canadian cities (ie. it's important in the urban system)

- There is **no standardized method** for measuring public transit accessibility across urban centres in Canada.

2. Aims and Objectives

1. To compute and visualize **Vancouver's public transit accessibility** to cultural amenities (museums, libraries, art galleries).

2. **To establish standardized methodologies** for data collection, wrangling, and accurate statistical representation of transit accessibility that can be scaled to other municipalities and points of interest.

3. Research Question

How accessible are Vancouver’s cultural amenities via the current **transit system**?

Chosen City:
Vancouver

Chosen Point of Interest:
Cultural and Art facilities

A Brief Note on Computing Accessibility

(All Origins → All Amenities)

	Mode	Access Measure based on:
Last year:		
This year:		

A Brief Note on Computing Accessibility

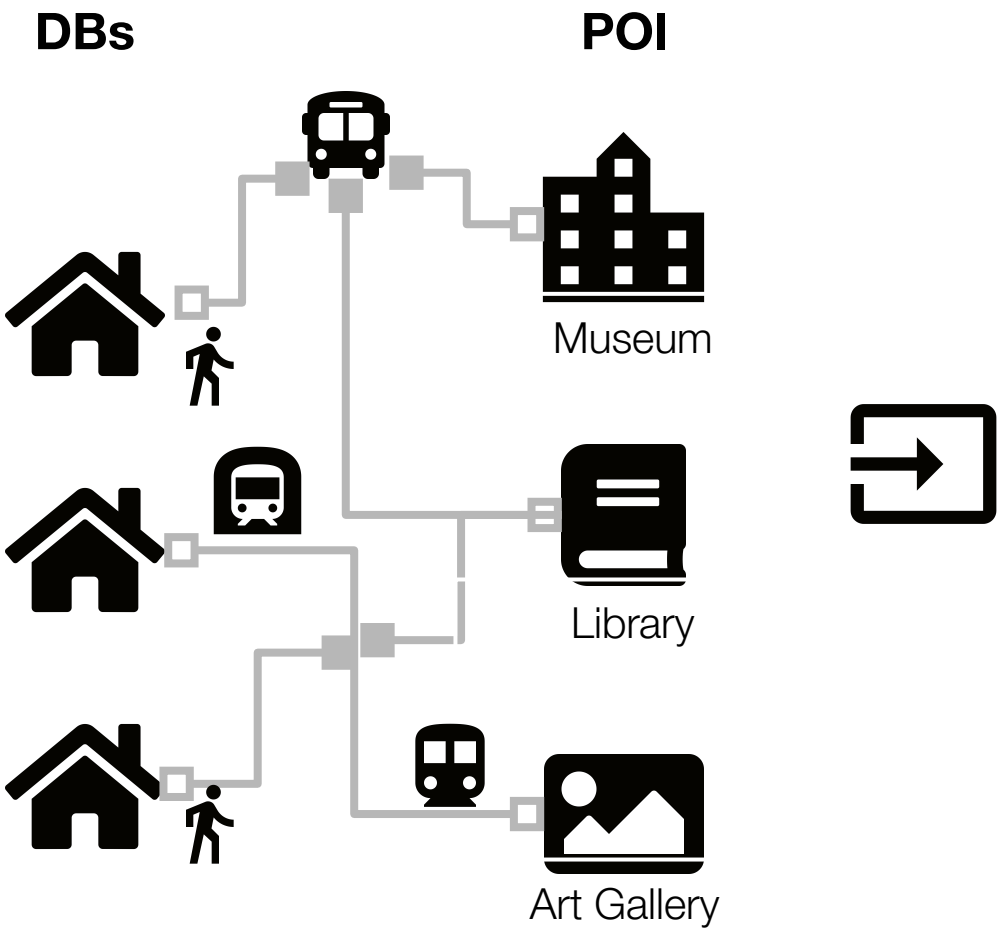
(All Origins → All Amenities)

	Mode	Access Measure based on:
Last year:	**walking/driving**	**distance**
This year:		

A Brief Note on Computing Accessibility

(All Origins → All Amenities)

	Mode	Access Measure based on:
Last year:	**walking/driving**	**distance**
This year:	**public transit**	**time**



Travel Time To Each Amenity	Accessibility Score
40 mins (Museum) 60 mins (Library) 80 mins (Art Gallery)	0.4
50 mins (Museum) 25 mins (Library) 15 mins (Art Gallery)	0.75
98 mins (Museum) 78 mins (Library) 25 mins (Art Gallery)	0.2

4. How do we do it?

4. How do we do it?

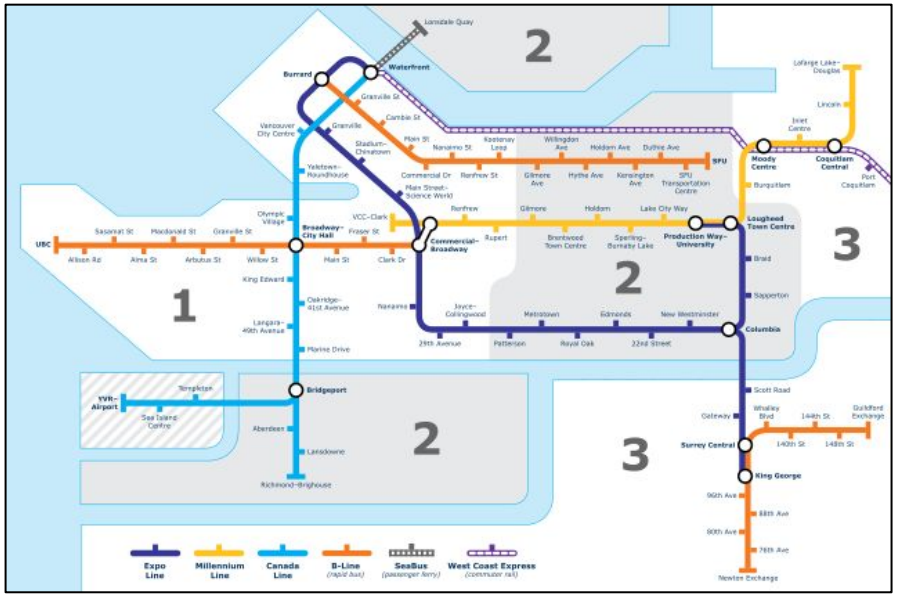
1. **What data do we require?**

4. How do we do it?

1. **What data do we require?**
    - Transit Network → General Transit Feed System (GTFS)

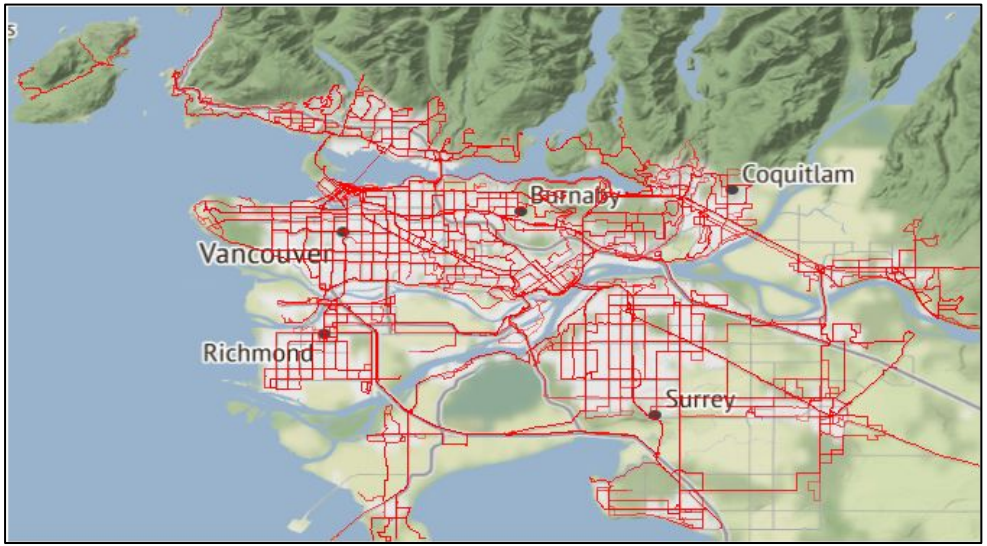
4. How do we do it?

1. **What data do we require?**
    - Transit Network → General Transit Feed System (GTFS)



4. How do we do it?

1. **What data do we require?**
    - Transit Network → General Transit Feed System (GTFS)



4. How do we do it?

1. **What data do we require?**
    - Transit Network → General Transit Feed System (**GTFS**)
    - Origins → UID, Latitude, Longitude (Census Dissemination Blocks (**CDB**))

4. How do we do it?

1. **What data do we require?**
    - Transit Network → General Transit Feed System (**GTFS**)
    - Origins → UID, Latitude, Longitude (Census Dissemination Blocks (**CDB**))

Dissemination Block UID = City Block

DBUID	lat	lon	pop
59150004004	49.3739	123.2738	241
...	...	...	...
(15,000+)			

4. How do we do it?

1. What data do we require?

- Transit Network → General Transit Feed System (**GTFS**)
- Origins → UID, Latitude, Longitude (Census Dissemination Blocks (**CDB**))
- Destinations → UID, Latitude, Longitude, Type
  (Open Database of Cultural and Art Facilities (**ODCAF**))

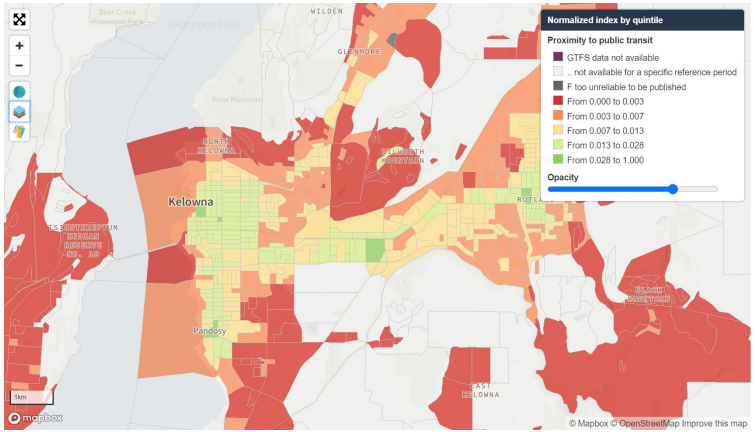
DBUID	lat	lon	pop
59150004004	49.3739	123.2738	241
...	...	...	...
(15,000+)			

ID	lat	lon	type
22	49.27879	-123.0988	gallery
...	...	...	...
(400+)			

4. How do we do it?

1. What data do we require?

- Transit Network → General Transit Feed System (**GTFS**)
- Origins → UID, Latitude, Longitude (Census Dissemination Blocks (**CDB**))
- Destinations → UID, Latitude, Longitude, Type (**ODCAF**)
- Vancouver Roads → **Open Street Map**
- Origin Polygons for Visualization → **CDB GeoJSON / Shape Files**



Dissemination block proximity to public transit in Kelowna. Obtained from the Proximity Measures Database (June 2020).

4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

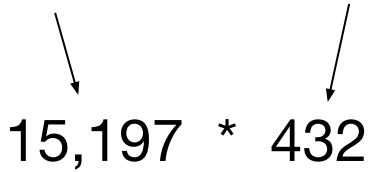
4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**
   (*all dissemination blocks → all cultural amenities*)

15,197

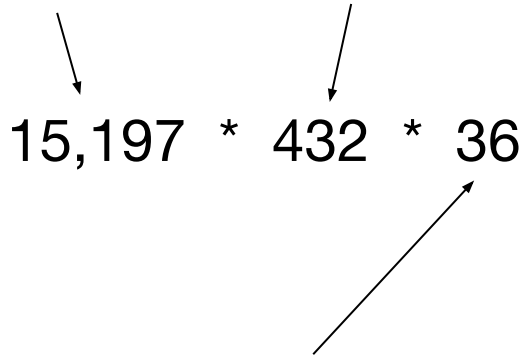
4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**
   (*all dissemination blocks → all cultural amenities*)



4. How do we do it?

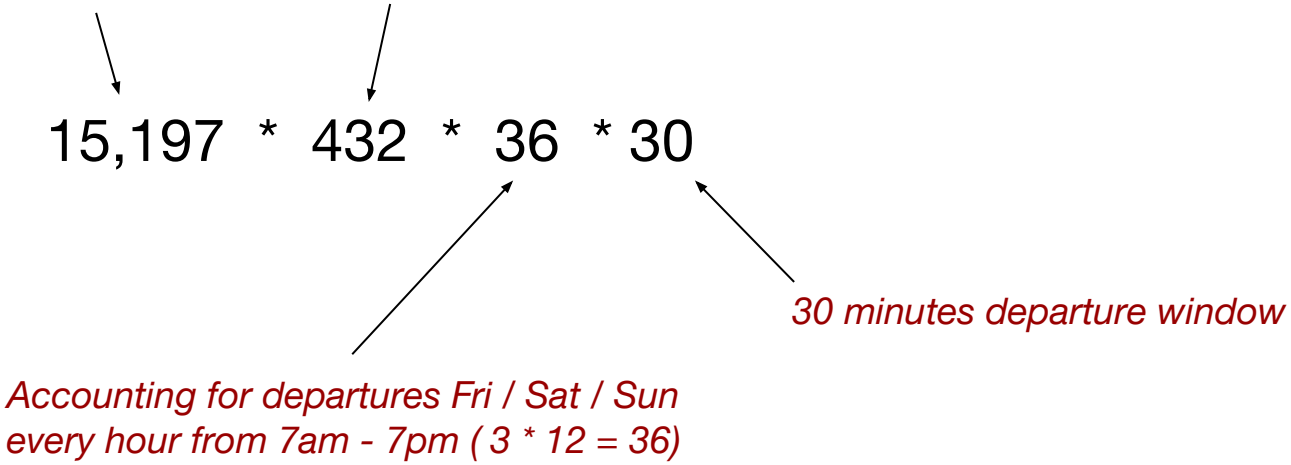
1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**
   (*all dissemination blocks → all cultural amenities)*



*Accounting for departures Fri / Sat / Sun every hour from 7am - 7pm (3 * 12 = 36)*

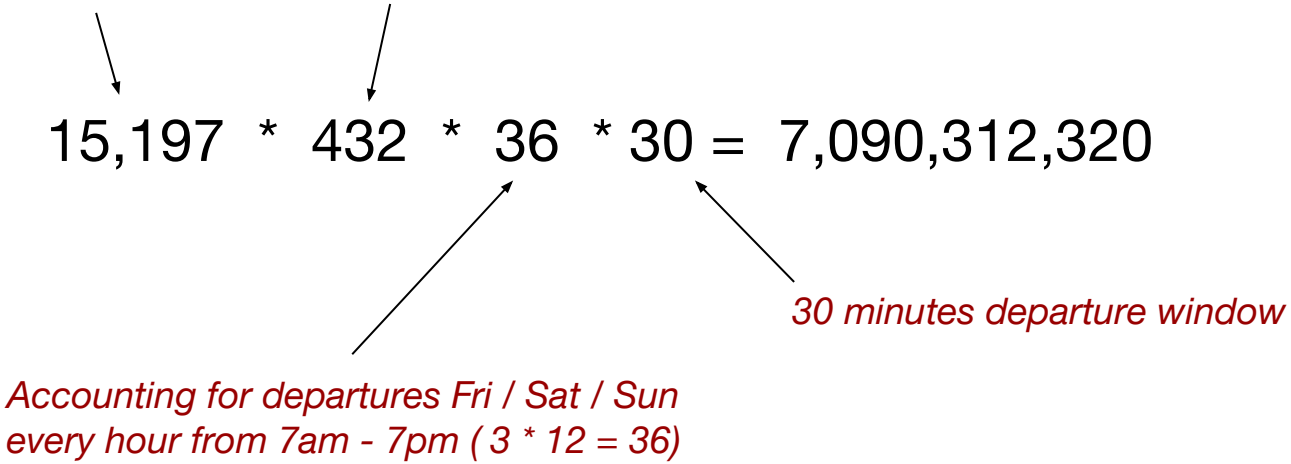
4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**
   (*all dissemination blocks → all cultural amenities)*



4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**
   (*all dissemination blocks → all cultural amenities)*



4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

7,090,312,320 * 1 us

7,090,312,320 * 1 ms

4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

7,090,312,320 * 1 us $<$ 2 hours

7,090,312,320 * 1 ms

4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

$$7{,}090{,}312{,}320 * 1\ \text{us} < 2\ \text{hours}$$

$$7{,}090{,}312{,}320 * 1\ \text{ms} > 80\ \text{days}$$

4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

$$7{,}090{,}312{,}320 * 1\ \text{us} \quad < 2\ \text{hours}$$

$$7{,}090{,}312{,}320 * 1\ \text{ms} \quad > 80\ \text{days}$$

Optimal routing engine selection is essential!

4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

    - **Conveyal R5: Rapid Realistic Routing on Real-world and Reimagined networks**
    - Extremely efficient for many-to-many routing computations

4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

   - **Conveyal R5: Rapid Realistic Routing on Real-world and Reimagined networks**
   - Extremely efficient for many-to-many routing computations
   - Available in R as the r5r library

4. How do we do it?

1. What data do we require?
2. **How can we efficiently compute a many-to-many travel time matrix?**

- **Conveyal R5: Rapid Realistic Routing on Real-world and Reimagined networks**
- Extremely efficient for many-to-many routing computations
- Available in R as the r5r library
- Travel Time Matrix (TTM) computation for a city of Vancouver's size: **~ 1 hour** when considering ~ 400 destinations

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. **How do we convert a transit time to a score?**

O	D	$\mu_{O,D}$	$\sigma_{O,D}$
fromId <chr>	toId <chr>	avg_unique_time <dbl>	sd_unique_time <dbl>
59150004004	10	99.76316	5.364721
59150004004	15	72.48718	3.401794
59150004004	157	96.69231	3.001349
59150004004	1759	106.82051	4.388213
59150004004	1760	46.58974	2.642944
59150004004	1822	76.64103	3.990035
59150004004	1839	76.15385	2.680715
59150004004	1840	75.15385	2.680715
59150004004	1916	99.07692	3.571706
59150004004	1930	86.97436	4.923024

o_i → 59150004004 ← d_j → μ_{o_i,d_j} → 72.48718 → σ_{o_i,d_j} → 3.401794

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. **How do we convert a transit time to a score?**

Unique trip score:
(from origin ***i***, o_i, to destination ***j***,d_j)

$$s_{o_i,d_j} = \frac{1}{\mu_{o_i,d_j} + 2\sigma_{o_i,d_j}}$$

Mean trip time

Std. Dev of trip time

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. **How do we convert a transit time to a score?**

Unique trip score:
(from origin ***i***, o_i, to destination ***j***,d_j)

$$s_{o_i,d_j} = \frac{1}{\mu_{o_i,d_j} + 2\sigma_{o_i,d_j}}$$

Mean trip time

Std. Dev of trip time

This is essentially just the inverse of the "worst case scenario" trip time.

Inverse yields a score between 0 and 1

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. **How do we convert a transit time to a score?**

Unique trip score:
(from origin ***i***, o_i, to destination ***j***,d_j)

$$s_{o_i,d_j} = \frac{1}{\mu_{o_i,d_j} + 2\sigma_{o_i,d_j}}$$

Unique origin score:
(from origin ***i*** to nearest ***{1,2,3,...,n}*** destinations)

$$s_{o_i} = \sum_{j=1}^{\{1,2,3,n\}} s_{o_i,d_j}$$

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. **How do we convert a transit time to a score?**

```
n1_ttm <- ttm %>%
     group_by(fromId, type) %>%
     summarise(avg_time = min(avg_time),
               sd_time = sd_time[which.min(avg_time)])


n1_ttm$unique_score <- 1/(n1_ttm $avg_time*n1_ttm $sd_time)

n1_ttm %>%
     group_by(fromId, type) %>%
     summarize(score = sum(unique_score)) %>%
     group_by(type) %>%
     mutate(score = normalize(score))
```

Get nearest time

Unique trip score

Summed Score

$$s_{o_i,d_j} = \frac{1}{\mu_{o_i,d_j} + 2\sigma_{o_i,d_j}}$$

$$s_{o_i} = \sum_{j=1}^{\{1,2,3,n\}} s_{o_i,d_j}$$

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. **How do we convert a transit time to a score?**

```
if (weight == FALSE) {
     df$unique_score <- 1/(df$avg_time*df$sd_time)
} else {
     df$unique_score <- df$weight/(df$avg_time*df$sd_time)
}
```

$$s_{o_i} = \sum_{j=1}^{\{1,2,3,n\}} \frac{1}{\mu_{o_i,d_j} + 2\sigma_{o_i,d_j}}$$

$$s_{o_i} = \sum_{j=1}^{\{1,2,3,n\}} \frac{w_{d_j}}{\mu_{o_i,d_j} + 2\sigma_{o_i,d_j}}$$

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**

4. How do we do it?

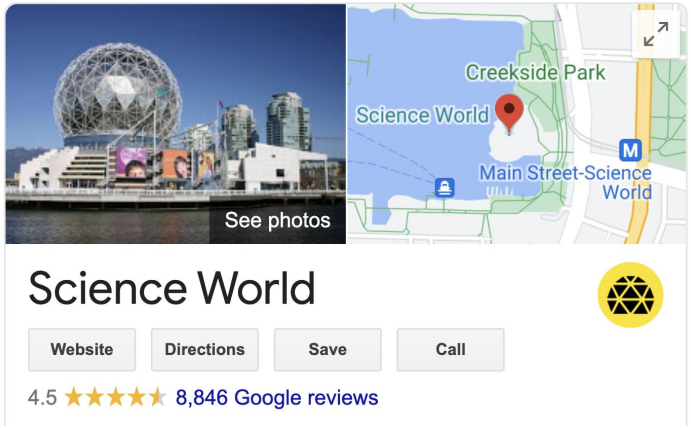
1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**
    - **Why would we even want to consider amenity popularity in the scoring?**

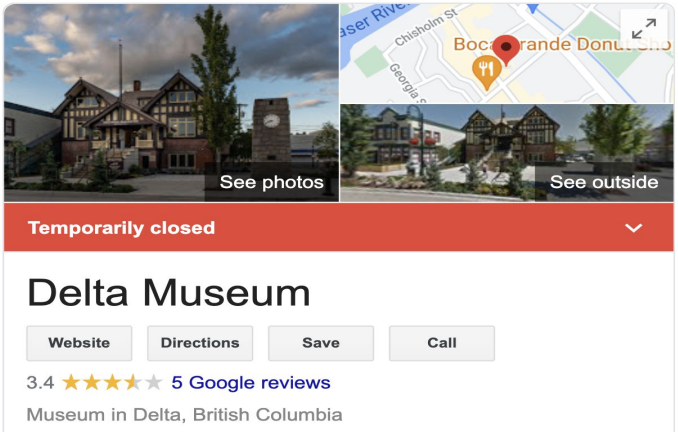
4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**
    - Why would we even want to consider amenity popularity in the scoring?
        - **People view places differently!**

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**
    - **Considering two places below**

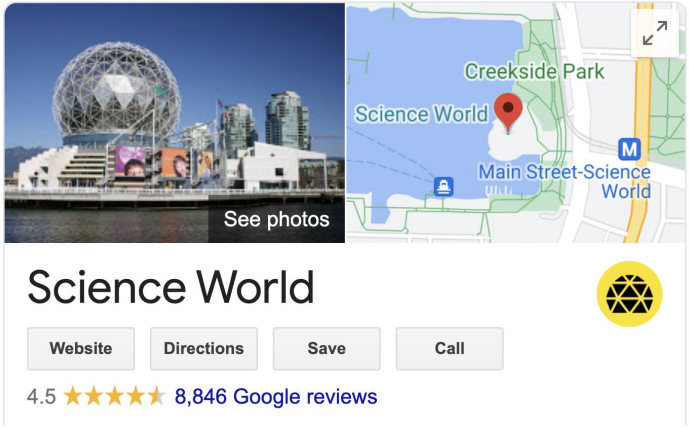


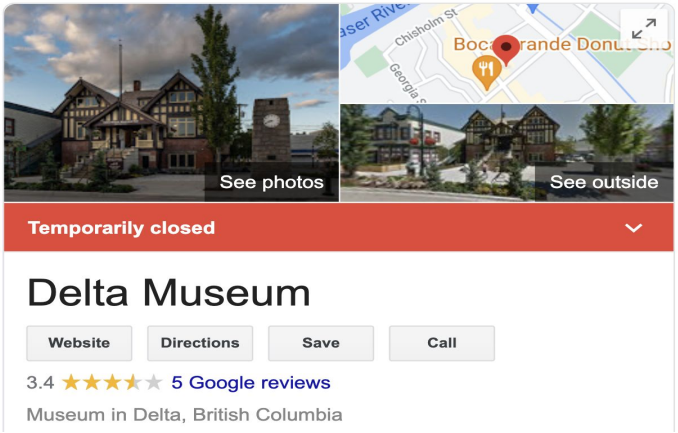


Equal weight =

Higher weight >

lower weight <





Equal weight =

Higher weight >

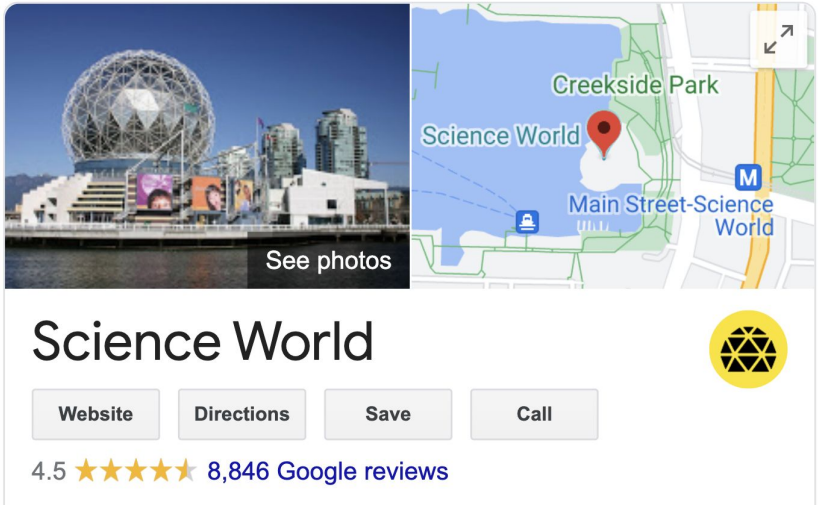
lower weight <

KPI ?

Key Performance Indicator

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**

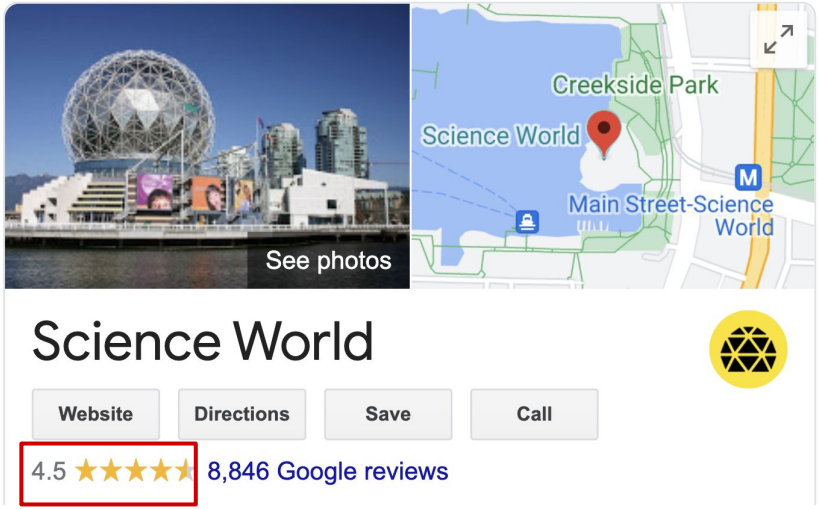


Hours:	**Sunday**	**10a.m.–5p.m.**
	Monday	10a.m.–5p.m.
	Tuesday	10a.m.–5p.m.
	Wednesday	10a.m.–5p.m.
	Thursday	10a.m.–5p.m.
	Friday	10a.m.–5p.m.
	Saturday	10a.m.–5p.m.

Suggest an edit

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**

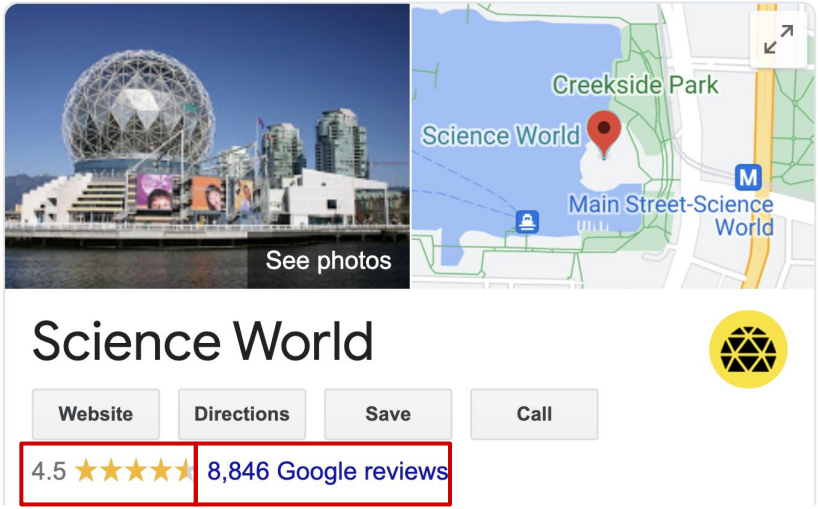


Hours:	**Sunday**	**10a.m.–5p.m.**
	Monday	10a.m.–5p.m.
	Tuesday	10a.m.–5p.m.
	Wednesday	10a.m.–5p.m.
	Thursday	10a.m.–5p.m.
	Friday	10a.m.–5p.m.
	Saturday	10a.m.–5p.m.

Suggest an edit

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**

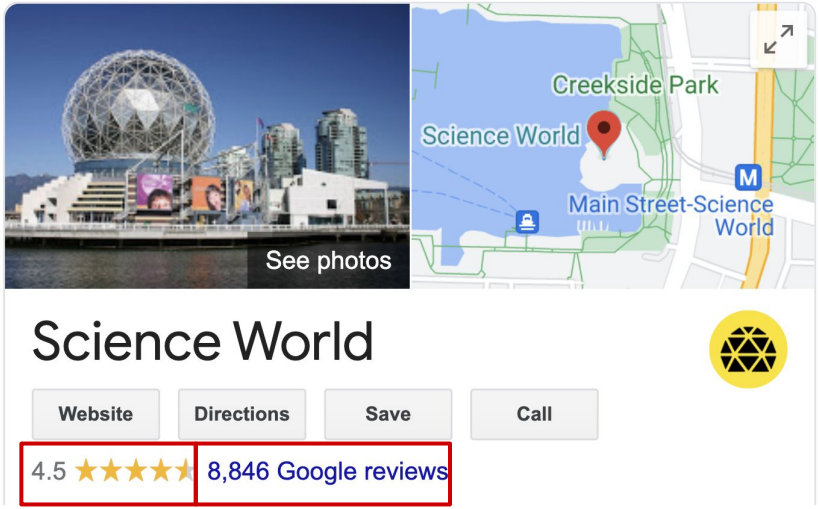


Hours:	**Sunday**	**10a.m.–5p.m.**
	Monday	10a.m.–5p.m.
	Tuesday	10a.m.–5p.m.
	Wednesday	10a.m.–5p.m.
	Thursday	10a.m.–5p.m.
	Friday	10a.m.–5p.m.
	Saturday	10a.m.–5p.m.

Suggest an edit

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. **How can we consider amenity popularity/usefulness in the scoring?**



Hours:	Sunday	10a.m.–5p.m.
	Monday	10a.m.–5p.m.
	Tuesday	10a.m.–5p.m.
	Wednesday	10a.m.–5p.m.
	Thursday	10a.m.–5p.m.
	Friday	10a.m.–5p.m.
	Saturday	10a.m.–5p.m.

Suggest an edit

4. How do we do it?

Using Google Place API to get the data!

Using name,geo data (lat,lon) to get pid

`ChIJhwQHwzh1hlQRJdE3ZdQPI9I`

Using unique pid to get place info

Create a Google places class for webscraping

```python
class GooglePlaces(object):
    def __init__(self, apiKey):
        super(GooglePlaces, self).__init__()
        self.apiKey = apiKey

    def search_places(self,name,inputtype,point):
        endpoint_url = "https://maps.googleapis.com/maps/api/place/findplacefromtext/json"
        params = {
            'input': name,
            'inputtype':inputtype,
            'locationbias':point,
            'key': self.apiKey
        }
        res = requests.get(endpoint_url, params = params)
        results =  json.loads(res.content)
        return results

    def get_place_details(self, place_id,fields):
        endpoint_url = "https://maps.googleapis.com/maps/api/place/details/json"
        params = {
            'placeid': place_id,
            'fields': ",".join(fields),
            'key': self.apiKey
        }
        res = requests.get(endpoint_url, params = params)
        place_details =  json.loads(res.content)
        return place_details
```

4. How do we do it?

Raw data before data cleaning

	Name	Rating	Total_Review	pid	opening_hours
0	39 Service Battalion (12 Company) Richmond	5	8	ChIJhwQHwzh1hlQRJdE3ZdQPI9I	[Monday: 9:30 AM – 4:00 PM, Tuesday: 9:30 AM –...
1	15th Field Artillery Regiment (RCA)	5	4	ChIJpdEUnLBzhlQR0ome6cv5el8	[Monday: Closed, Tuesday: Closed, Wednesday: 7...
2	ABC Preschool Academy	no data	0	ChIJF3ZMao7RhVQRYMRn09PBo34	0
3	Agassiz Library	5	8	ChIJNaGyI2sThFQRQ33H2ji42y4	[Monday: Closed, Tuesday: 10:00 AM – 6:00 PM, ...
4	Ponderosa Commons East (Audain Art Centre and ...	5	0	ChIJaUbtlrVyhlQREWwxkqr-Y9A	0
...	...	...	...	...	...

4. How do we do it?

After wrangling we have the clean dataset for point of interest

poi_name <chr>	open_days <dbl>	Total_hours <dbl>	Rating <dbl>	Total_Review <dbl>
Science World At Telus World of Science	7	49.00000	4.500000	8833.0000
Van Dusen Botanical Garden	7	49.00000	4.600000	6937.0000
Bloedel Conservatory	7	47.25000	4.600000	4466.0000
Lynn Canyon Ecology Centre	4	35.88077	4.700000	4053.0000
Dr. Sun Yat-Sen Classical Chinese Garden	2	10.00000	4.200000	3885.0000
Museum of Anthropology	6	42.00000	4.700000	3406.0000
UBC Museum of Anthropology	6	42.00000	4.700000	3406.0000
Burnaby Village Museum	4	35.88077	4.600000	2167.0000
UBC Botanical Garden	5	35.00000	4.500000	936.0000
Beaty Biodiversity Museum	6	42.00000	4.500000	662.0000

4. How do we do it?

- There are some issues!

poi_name <chr>	open_days <dbl>	Total_hours <dbl>	Rating <dbl>	Total_Review <dbl>
Science World At Telus World of Science	7	49.00000	4.500000	8833.0000
Van Dusen Botanical Garden	7	49.00000	4.600000	6937.0000
Bloedel Conservatory	7	47.25000	4.600000	4466.0000
Lynn Canyon Ecology Centre	4	35.88077	4.700000	4053.0000
Dr. Sun Yat-Sen Classical Chinese Garden	2	10.00000	4.200000	3885.0000
Museum of Anthropology	6	42.00000	4.700000	3406.0000
UBC Museum of Anthropology	6	42.00000	4.700000	3406.0000
Burnaby Village Museum	4	35.88077	4.600000	2167.0000
UBC Botanical Garden	5	35.00000	4.500000	936.0000
Beaty Biodiversity Museum	6	42.00000	4.500000	662.0000

4. How do we do it?

- There are some issues!
  - **They are not comparable!**

poi_name <chr>	open_days <dbl>	Total_hours <dbl>	Rating <dbl>	Total_Review <dbl>
Science World At Telus World of Science	7	49.00000	4.500000	8833.0000
Van Dusen Botanical Garden	7	49.00000	4.600000	6937.0000
Bloedel Conservatory	7	47.25000	4.600000	4466.0000
Lynn Canyon Ecology Centre	4	35.88077	4.700000	4053.0000
Dr. Sun Yat-Sen Classical Chinese Garden	2	10.00000	4.200000	3885.0000
Museum of Anthropology	6	42.00000	4.700000	3406.0000
UBC Museum of Anthropology	6	42.00000	4.700000	3406.0000
Burnaby Village Museum	4	35.88077	4.600000	2167.0000
UBC Botanical Garden	5	35.00000	4.500000	936.0000
Beaty Biodiversity Museum	6	42.00000	4.500000	662.0000

4. How do we do it?

- There are some issues!

poi_name <chr>	open_days <dbl>	Total_hours <dbl>	Rating <dbl>	Total_Review <dbl>
Science World At Telus World of Science	7	49.00000	4.500000	8833.0000
Van Dusen Botanical Garden	7	49.00000	4.600000	6937.0000
Bloedel Conservatory	7	47.25000	4.600000	4466.0000
Lynn Canyon Ecology Centre	4	35.88077	4.700000	4053.0000
Dr. Sun Yat-Sen Classical Chinese Garden	2	10.00000	4.200000	3885.0000
Museum of Anthropology	6	42.00000	4.700000	3406.0000
UBC Museum of Anthropology	6	42.00000	4.700000	3406.0000
Burnaby Village Museum	4	35.88077	4.600000	2167.0000
UBC Botanical Garden	5	35.00000	4.500000	936.0000
Beaty Biodiversity Museum	6	42.00000	4.500000	662.0000

4. How do we do it?

- **Need to normalize the data!**

4. How do we do it?

- **Need to normalize the data!**

$$Normalize_i = \frac{Value_i - min(Value)}{max(Value) - min(Value)}$$

4. How do we do it?

Normalized the amenity features:

poi_name <chr>	open_days <dbl>	Total_hours <dbl>	Rating <dbl>	Total_Review <dbl>
Science World At Telus World of Science	1.0000000	0.39423077	0.6875000	1.0000000000
Van Dusen Botanical Garden	1.0000000	0.39423077	0.7500000	0.7853260870
Bloedel Conservatory	1.0000000	0.37740385	0.7500000	0.5055480072
Lynn Canyon Ecology Centre	0.5000000	0.26808432	0.8125000	0.4587862319
Dr. Sun Yat-Sen Classical Chinese Garden	0.1666667	0.01923077	0.5000000	0.4397644928
Museum of Anthropology	0.8333333	0.32692308	0.8125000	0.3855298913
UBC Museum of Anthropology	0.8333333	0.32692308	0.8125000	0.3855298913
Burnaby Village Museum	0.5000000	0.26808432	0.7500000	0.2452445652
UBC Botanical Garden	0.6666667	0.25961538	0.6875000	0.1058650362
Beaty Biodiversity Museum	0.8333333	0.32692308	0.6875000	0.0748414855

4. How do we do it?

$Nor(Hours)$ $Nor(Rating)$ $Nor(Reviews)$

$Nor(Days)$

open_days <dbl>	Total_hours <dbl>	Rating <dbl>	Total_Review <dbl>	Index <dbl>
1.0000000	0.39423077	0.6875000	1.0000000000	0.77043269
1.0000000	0.39423077	0.7500000	0.7853260870	0.73238921
1.0000000	0.37740385	0.7500000	0.5055480072	0.65823796
0.5000000	0.26808432	0.8125000	0.4587862319	0.50984264
0.1666667	0.01923077	0.5000000	0.4397644928	0.28141548
0.8333333	0.32692308	0.8125000	0.3855298913	0.58957158
0.8333333	0.32692308	0.8125000	0.3855298913	0.58957158
0.5000000	0.26808432	0.7500000	0.2452445652	0.44083222
0.6666667	0.25961538	0.6875000	0.1058650362	0.42991177
0.8333333	0.32692308	0.6875000	0.0748414855	0.48064947

$$Naive\_Score = \frac{Nor(Hour) + Nor(Days) + Nor(Rating) + Nor(Reviews)}{N_i}$$

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 1:

- Leaflet - Interactive maps

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 1:

- Leaflet - Interactive maps
- Display data as a street map

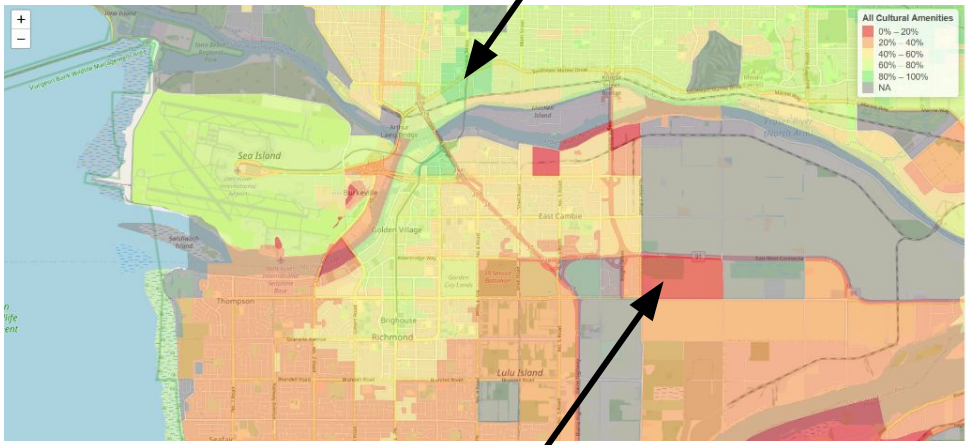
4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 1:

- Leaflet - Interactive maps
- Display data as a street map
- Display scores using color scheme
  - Red = Low accessibility
  - Green = High accessibility

Higher accessibility



Access to all Amenities

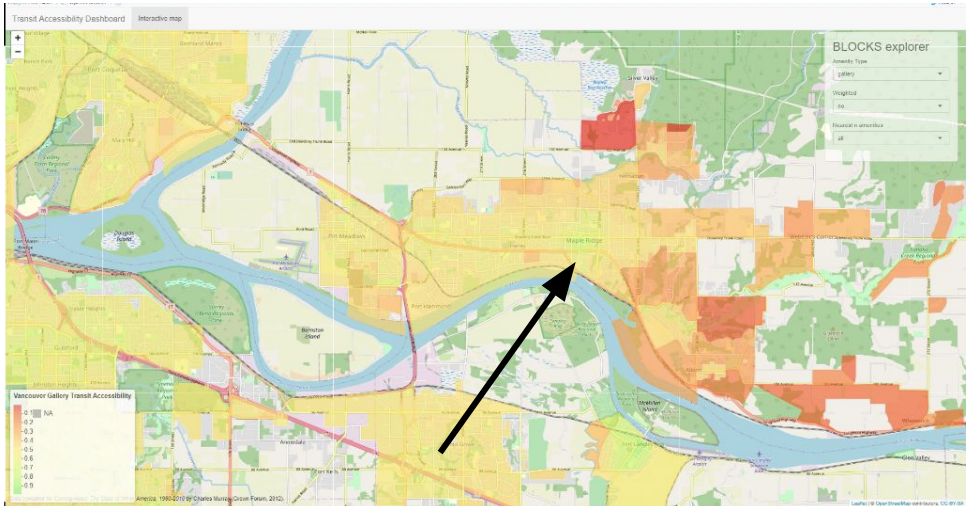
Lower accessibility

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 1:

- Color scheme
  - Numeric (Continuous)
    - 0 = Red
    - 1 = Green
  - Quantile (Discrete)
    - Red = Lowest 20 %
    - Green = Highest 20 %



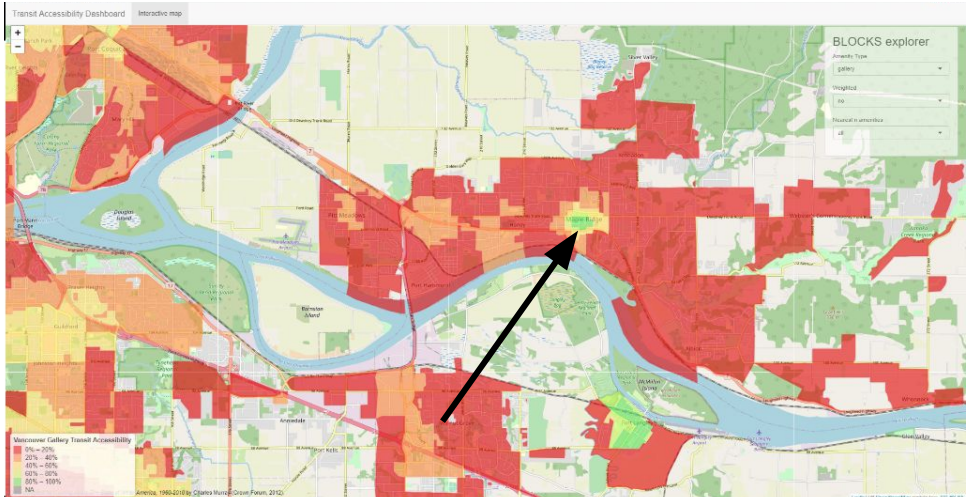
Access to all Galleries - Continuous color scheme

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 1:

- Color scheme
  - Numeric (Continuous)
    - 0 = Red
    - 1 = Green
  - Quantile (Discrete)
    - Red = Lowest 20 %
    - Green = Highest 20 %



Access to all Galleries - Discrete color scheme

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 2:

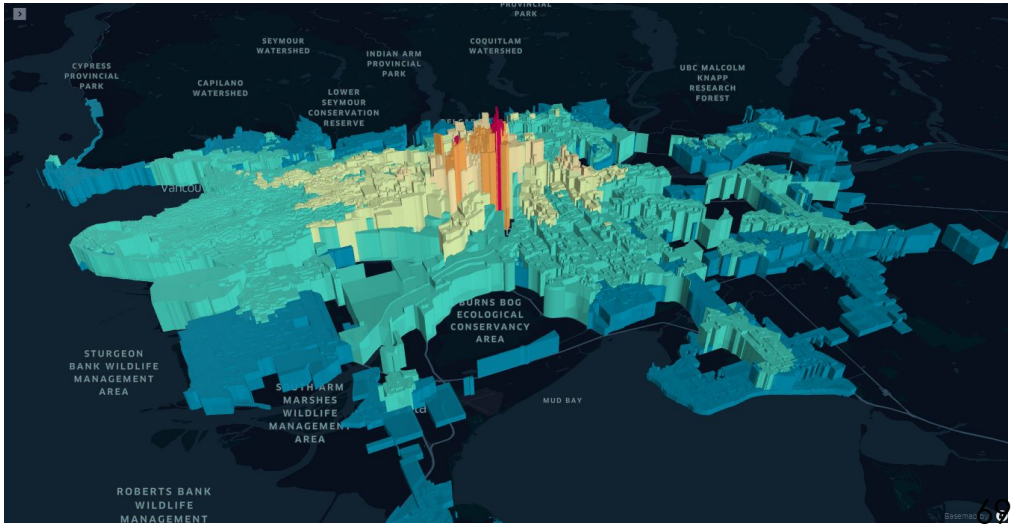
- Kelper.gl - 3D Interactive map

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 2:

- Kelper.gl - 3D Interactive map
- Color scheme and height scales



Access to all Artist

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 2:

- Kelper.gl - 3D Interactive map
- Color scheme and height scales
- Select interested variables

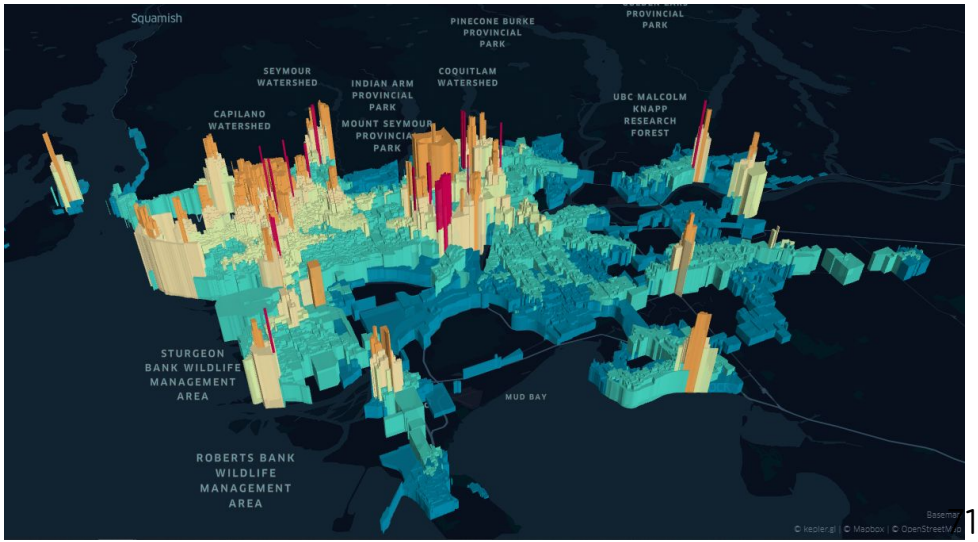
Nearest 1 - Libraries

4. How do we do it?

1. What data do we require?
2. How can we efficiently compute a many-to-many travel time matrix?
3. How do we convert a transit time to a score?
4. How can we consider amenity popularity/usefulness in the scoring?
5. **How do we visualize the data?**

Method 2:

- Kelper.gl - 3D Interactive map
- Color scheme and height scales
- Select interested variables
- Change color scales



And now,

the awesome moment you've all eagerly been waiting for…

5. Demo

6. Roadblocks

- **Score formula updates**
  - Give average travel time and standard deviation different weights
    - Original approach - same weight for both
    - Scores with higher std change significantly
  - Consider each combination of DB and destination separately
    - Original approach - n/(mean*std)
  - New approach - 1/(mean + 2*std)

- **Dashboard performance is slow**
  - Leaflet takes too long to render new maps
  - R shiny dashboard memory limit
  - Solution: pre-rendered HTML files

7. What's to come…

1. Isochrone contours for **interpretable** transit times

7. What's to come…

1. Isochrone contours for **interpretable** transit times
2. **Network efficiency analysis** using city block populations
    - *Measure difference between population and accessibility score*

      ↑ difference = disproportionate transit resource allocation

7. What's to come…

1. Isochrone contours for **interpretable** transit times
2. **Network efficiency analysis** using city block populations
    - *Measure difference between population and accessibility score*

      ↑ difference = disproportionate transit resource allocation
3. **Equity Analysis** using Vancouver demographic data

7. What's to come…

1. Isochrone contours for **interpretable** transit times
2. **Network efficiency analysis** using city block populations
    - *Measure difference between population and accessibility score*

      ↑ difference = disproportionate transit resource allocation
3. **Equity Analysis** using Vancouver demographic data

Let us know what you would be interested in!

Questions?

Capstone team

Thank you