Modeling Longitudinal Multi-Modal Accessibility: An Open-Source Workflow with R5R

Juhyeon Park

2025-10-30

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

Part I

Introduction

1 Urban accessibility over time

1.1 What is accessibility?

Urban accessibility refers to the ease with which people can reach essential services, opportunities, and activities using available transportation systems. Unlike simple distance-based proximity, accessibility accounts for the complex interplay between:

- **People**: Where individuals live and their mobility capabilities
- **Places**: The spatial distribution of destinations (jobs, healthcare, education, banks)
- **Transport**: The multimodal networks connecting origins to destinations

This concept is fundamental to understanding urban equity and quality of life. A neighborhood might be geographically close to essential services but poorly accessible if transportation connections are inadequate or inconvenient.

1.2 The limitations of static accessibility

Most accessibility analyses treat cities as frozen snapshots, calculating access at a single point in time. This static view misses critical dynamics:

Services relocate or close

- Bank branch consolidation leaves neighborhoods underserved
- Healthcare facilities open in new locations
- Supermarkets close in low-income areas

Transport networks evolve

- New transit lines extend service coverage
- Route restructuring changes travel times
- Infrastructure improvements alter commute patterns

Land use changes

- Residential development shifts population centers
- Commercial zones emerge or decline

- Urban sprawl extends city boundaries

A single-year analysis cannot capture these transformations or their equity implications.

1.3 Why track accessibility over time?

Longitudinal accessibility analysis reveals patterns invisible in cross-sectional studies:

Detecting growing inequalities

Service centralization often benefits well-connected urban cores while disadvantaging peripheral areas. Only temporal tracking shows whether gaps are widening or narrowing.

Evaluating policy impacts

Transport investments and service location policies play out over years. Longitudinal data connects interventions to outcomes, separating policy effects from background trends.

Understanding cumulative disadvantage

Communities experiencing persistent poor accessibility face compounding challenges in economic participation, health outcomes, and social mobility.

1.4 The multimodal dimension

People reach destinations through different transportation modes with vastly different accessibility profiles:

Private vehicles offer flexibility and speed but require: - Car ownership and maintenance costs - Driving ability - Parking availability - Tolerance for congestion

Public transit provides affordable mobility but involves: - Fixed routes and schedules - Waiting and transfer times - Limited service coverage - Temporal availability (peak vs off-peak, weekday vs weekend)

Walking and cycling are sustainable options constrained by: - Physical capability - Distance tolerance - Weather and topography - Infrastructure quality

Analyzing only one mode obscures how accessibility is distributed across different population groups. A location might have excellent car accessibility but poor transit access—a pattern that systematically disadvantages lower-income, elderly, and disabled populations who disproportionately rely on public transport.

1.5 Banking accessibility in South Korea: A case study

This book demonstrates longitudinal multimodal accessibility analysis through a concrete example: tracking access to banking services in South Korea from 2021 to 2023.

Why banking services?

Financial service access is essential for economic participation yet increasingly threatened by branch closures as banking digitalizes. Physical branches remain critical for:

- Older adults less comfortable with digital banking
- Complex transactions requiring in-person assistance
- Communities with limited digital infrastructure

The South Korean context

South Korea experienced significant bank branch consolidation in the early 2020s, creating natural variation in accessibility changes across time and space. The country also provides high-quality open data:

- GTFS transit schedules from KTDB (Korea Transport Database)
- Comprehensive street networks from OpenStreetMap
- Detailed census boundaries and population data

Analytical scope

We compare accessibility by private car versus public transit across approximately 50,000 enumeration districts, tracking how access changed over six time points (two per year, 2021-2023). This reveals:

- Which communities lost banking access
- Whether transit-dependent populations were disproportionately affected
- How accessibility gaps evolved over time

1.6 Beyond this example

While our case study focuses on banking in South Korea, the methods generalize to:

- **Other service types**: Healthcare, education, retail, employment
- **Other geographic contexts**: Any region with GTFS and street network data
- **Other time periods**: Any span where comparable data exist

The workflow presented in this book provides infrastructure for reproducible longitudinal accessibility analysis applicable wherever standardized transport and service location data are available.

1.7 Structure of this book

The remaining chapters build toward national-scale longitudinal accessibility analysis:

SECTION 1: Introduction (this section)

- Chapter 2 introduces GTFS and OpenStreetMap data
- Chapter 3 explains the r5r routing engine

SECTION 2: Methodology Development

- Network setup and parameters
- Three routing patterns: one-to-one, one-to-many, many-to-many
- Building reusable code for longitudinal analysis

SECTION 3: Pilot Application

- Small-scale implementation in Paldal-gu, Suwon
- Method validation with manageable data
- Development of computational strategies

SECTION 4: National-Scale Application

- Scaling to 50,000+ origins and 6 time points
- Spatial partitioning and parallel processing
- Results visualization and interpretation

By the end, you will understand both the conceptual foundations and practical implementation of reproducible longitudinal multimodal accessibility research.

2 Multimodal transport data

Multimodal accessibility analysis requires two foundational data types: public transit schedules and street network geometry. Fortunately, global standards and collaborative mapping efforts have made both increasingly available as open data.

2.1 GTFS: General Transit Feed Specification

2.1.1 What is GTFS?

The General Transit Feed Specification (GTFS) is a standardized data format for public transportation schedules and geographic information. Originally developed by Google in 2005 for trip planning applications, GTFS has become the de facto international standard for representing transit systems.

A GTFS dataset is a collection of text files (packaged as a `.zip` archive) containing:

- **Routes**: Bus, metro, and train lines
- **Stops**: Station and stop locations with coordinates
- **Trips**: Specific vehicle journeys along routes
- **Stop times**: When each trip arrives/departs each stop
- **Calendar**: Service availability by date and day of week
- **Shapes** (optional): Geographic paths that routes follow

2.1.2 Why GTFS matters for accessibility

GTFS enables realistic transit routing because it captures temporal complexity:

- **Schedule accuracy**: Actual departure/arrival times, not assumed frequencies
- **Service variation**: Different schedules for weekday/weekend, peak/off-peak
- **Temporal validity**: Calendar defines when schedules apply
- **Multimodal integration**: Includes walk transfers between stops/lines

This temporal dimension is critical for longitudinal analysis. Transit service changes—new routes, frequency adjustments, coverage expansion—are reflected in updated GTFS feeds, making it possible to track how public transport accessibility evolves.

2.1.3 Where to obtain GTFS

GTFS availability has grown dramatically:

Global repositories

- Mobility Database: 2,000+ feeds worldwide
- TransitFeeds: Historical and current feeds
- Transit agency websites: Many agencies publish GTFS directly

South Korea

- KTDB (Korea Transport Database): National GTFS feeds
- Covers intercity, metropolitan, and local transit systems
- Available for download with registration

Temporal coverage

For longitudinal analysis, you need GTFS snapshots from multiple time points. Some agencies maintain archives; others require manual collection over time. The Mobility Database and TransitFeeds preserve historical versions.

> ℹ Data acquisition in this book
>
> Detailed GTFS download and preprocessing workflows for South Korea are covered in:
>
> - **Section 3.2**: Pilot data preparation (Paldal-gu)
> - **Section 4.2**: National data preparation
>
> Here we focus on conceptual understanding rather than step-by-step data acquisition.

2.2 OpenStreetMap: Collaborative street networks

2.2.1 What is OpenStreetMap?

OpenStreetMap (OSM) is a collaborative project creating a free, editable map of the world. Think of it as "Wikipedia for maps"—volunteers contribute geographic data including:

- **Road networks**: Streets, highways, pedestrian paths, cycle lanes
- **Road attributes**: Speed limits, surface types, lane counts
- **Points of interest**: Buildings, amenities, landmarks
- **Administrative boundaries**: City, district, neighborhood borders

For accessibility analysis, we primarily use OSM's street network data.

2.2.2 Why OSM for accessibility?

OSM offers unique advantages over proprietary map data:

Open and free

- No licensing costs or usage restrictions
- Enables reproducible research

Global coverage

- Especially strong in areas without commercial mapping
- Community maintains and updates continuously

Rich attributes

- Tags describe road types (motorway, residential, footway)
- Attributes enable mode-specific routing (car vs walk vs bike)

Programmable access

- APIs and tools for bulk download
- Standard formats (.pbf, .osm) work with routing engines

2.2.3 OSM data structure

OSM represents geography through three primitives:

- **Nodes**: Points with latitude/longitude
- **Ways**: Ordered lists of nodes forming lines or polygons
- **Relations**: Logical groupings of nodes/ways (e.g., bus routes)

For routing, we need ways representing streets, tagged with attributes like:

- `highway=motorway`, `highway=residential`, `highway=footway`
- `maxspeed=50`
- `lanes=2`
- `surface=asphalt`

2.2.4 Obtaining OSM data

Global extracts

- Geofabrik: Daily-updated country/region extracts
- BBBike: Custom area extracts

OSM APIs

- Overpass API: Custom queries for specific features
- osmnx (Python), osmdata (R): Programmatic access

South Korea

Geofabrik provides South Korea extracts updated daily in `.pbf` (Protocolbuffer Binary Format), the compact format used by routing engines.

> **! Temporal consistency**
>
> For longitudinal analysis, OSM presents a challenge: the database continuously updates. To ensure temporal validity:
>
> - Download OSM extracts from dates matching your GTFS feeds
> - Use services archiving historical OSM data (e.g., Geofabrik's historical extracts)
> - In practice, road network changes are slower than transit schedule changes, making minor temporal mismatches acceptable for most analyses

2.3 GTFS + OSM: Complementary data

These two data sources form a complete multimodal transport network:

Data source	Represents	Enables routing for
GTFS	Public transit routes, schedules, stops	Bus, metro, train travel
OSM	Street networks, paths	Walking, cycling, driving

Integration logic

Multimodal routing engines like R5 (discussed in Chapter 3) combine these datasets:

1. **Transit network**: Constructed from GTFS stops, routes, and schedules
2. **Street network**: Built from OSM ways tagged for different modes

3. **Transfers**: Walking connections between transit stops use OSM pedestrian paths
4. **Access/egress**: Trips start/end with walks along OSM streets to/from transit

For example, a public transit trip from home to a bank might involve:

- Walk from home to bus stop (OSM pedestrian network)
- Bus journey (GTFS schedule)
- Transfer walk between bus and metro (OSM + GTFS stop locations)
- Metro journey (GTFS schedule)
- Walk from metro station to bank (OSM pedestrian network)

2.4 Temporal dimensions in longitudinal analysis

For longitudinal accessibility studies, we need matched GTFS and OSM snapshots at each time point:

This book's temporal structure

Year	Time points	GTFS source	OSM source
2021	June, December	KTDB	Geofabrik (matching dates)
2022	June, December	KTDB	Geofabrik (matching dates)
2023	June, December	KTDB	Geofabrik (matching dates)

This creates six multimodal network snapshots, enabling analysis of how accessibility changed over three years.

2.5 Data quality considerations

Both GTFS and OSM rely on data contributions—by transit agencies and volunteers, respectively. Quality varies:

GTFS quality signals

- Official agency feeds (higher trust)
- Regular updates (active maintenance)
- Complete shapes.txt (geographic accuracy)
- Validation tools (e.g., Google's GTFS validator)

OSM quality signals

- Density of contributors in the region

- Recency of edits
- Completeness of road network
- Attribute richness (speed limits, lanes, etc.)

South Korea data quality

Both GTFS (from KTDB) and OSM have excellent coverage and quality in South Korea:

- KTDB provides official, validated GTFS feeds
- OSM road network is comprehensive, especially in urban areas
- Active OSM contributor community maintains currency

2.6 Beyond GTFS and OSM

While this book focuses on these two data sources, other data enrich accessibility analysis:

- **Population data**: Census boundaries and counts (where people live)
- **POI data**: Service locations (banks, hospitals, schools)
- **Topography**: Elevation affecting walking/cycling difficulty
- **Real-time data**: Live transit positions, traffic speeds

We incorporate population and POI data in later sections. The methods presented here generalize to any accessibility analysis with GTFS and street network inputs.

2.7 What's next

Understanding GTFS and OSM as standardized, open data sources is essential context. But how do we actually use these data to calculate travel times and accessibility? Chapter 3 introduces r5r, the routing engine that transforms GTFS and OSM into actionable multimodal accessibility estimates.

3 Routing with r5r

GTFS and OSM data provide the raw ingredients for accessibility analysis, but we need a routing engine to transform static data into dynamic travel time estimates. This chapter introduces r5r, the tool that powers the multimodal accessibility calculations throughout this book.

3.1 Why routing engines matter

Simple distance-based accessibility—measuring how many services fall within a radius—ignores transportation realities:

- A bank 2 km away by road might require a 15-minute drive or a 45-minute bus journey
- Two destinations equidistant from an origin can have vastly different accessibility by transit
- Network connectivity, traffic speeds, transit schedules, and transfer requirements shape true accessibility

Routing engines compute realistic travel times by simulating journeys through transportation networks, accounting for:

- Road network topology and speeds
- Transit schedules and frequencies
- Waiting times and transfers
- Mode-specific constraints (walk speed, vehicle access)

This transforms accessibility from a geometric problem (proximity) into a network problem (reachability).

3.2 R5: Rapid Realistic Routing

R5 (Rapid Realistic Routing on Real-world and Reimagined networks) is a high-performance multimodal routing engine developed by Conveyal. Written in Java, R5 specializes in:

Multimodal routing

- Combines GTFS (transit) and OSM (streets) into a unified network
- Handles walk-transit-walk trip chains
- Supports car, transit, bicycle, and walking

Speed

- Optimized algorithms for large-scale analysis
- Parallel computation of many-to-many travel times
- Designed for city-scale and regional accessibility studies

Temporal accuracy

- Respects transit schedules from GTFS
- Accounts for waiting times and departure time windows
- Handles frequency-based vs schedule-based services

Open source

- Free to use and modify
- Active development community
- Well-documented and validated

R5 has become a standard tool in transport research and practice, powering accessibility analyses for transportation agencies, researchers, and advocacy organizations worldwide.

3.3 r5r: R5 in R

While R5 is powerful, it's a Java application requiring technical setup and command-line interaction. The r5r package (Pereira et al., 2021) makes R5 accessible to R users through a simple, intuitive interface.

Why r5r?

Ease of use

- Install like any R package: `install.packages("r5r")`
- No Java configuration required (r5r handles it automatically)
- Familiar R syntax and data structures

Integration with R ecosystem

- Works seamlessly with `sf` for spatial data
- Outputs data.frames compatible with `tidyverse` workflows
- Integrates with `ggplot2` for visualization

Reproducible workflows

- All code in R scripts or Quarto documents
- Facilitates version control and collaboration
- Lowers barriers for replication studies

Active development

- Maintained by researchers at Ipea (Brazil) and collaborators
- Regular updates incorporating R5 improvements
- Strong documentation and community support

r5r has been widely adopted in transport geography and urban planning research, enabling reproducible multimodal accessibility studies across diverse contexts (see Pereira & Herszenhut, 2023 for an extensive guide).

3.4 How r5r works: A conceptual overview

r5r operates through a simple workflow:

3.4.1 1. Build a multimodal network

```r
library(r5r)

r5r_core <- setup_r5(
  data_path = "path/to/data",  # Folder with GTFS .zip and OSM .pbf
  verbose = FALSE
)
```

This single function:

- Loads GTFS files (transit routes, stops, schedules)
- Loads OSM `.pbf` file (street network)
- Constructs a routable multimodal graph
- Saves a `network.dat` file for reuse

The `r5r_core` object is a connection to the R5 Java backend, ready for routing queries.

3.4.2 2. Define origins and destinations

Origins and destinations are point locations with coordinates:

```
origins <- data.frame(
  id = c("home_1", "home_2"),
  lon = c(126.9780, 126.9850),
  lat = c(37.5665, 37.5700)
)

destinations <- data.frame(
  id = c("bank_1", "bank_2"),
  lon = c(127.0000, 127.0100),
  lat = c(37.5750, 37.5800)
)
```

3.4.3 3. Calculate travel times

```
travel_times <- travel_time_matrix(
  r5r_core = r5r_core,
  origins = origins,
  destinations = destinations,
  mode = c("WALK", "TRANSIT"),
  departure_datetime = as.POSIXct("2021-06-15 09:00:00"),
  max_trip_duration = 120
)
```

r5r computes travel times for all origin-destination pairs, respecting:

- Transit schedules on the specified date/time
- Maximum trip duration constraint
- Specified mode(s)

The output is a tidy data.frame:

from_id	to_id	travel_time_p50
home_1	bank_1	28
home_1	bank_2	35
home_2	bank_1	22
home_2	bank_2	30

3.4.4 4. Calculate accessibility

For cumulative opportunity accessibility (how many destinations are reachable within a time threshold):

```
accessibility <- accessibility(
  r5r_core = r5r_core,
  origins = origins,
  destinations = destinations,
  mode = c("WALK", "TRANSIT"),
  departure_datetime = as.POSIXct("2021-06-15 09:00:00"),
  cutoffs = c(30, 60)  # minutes
)
```

Output counts how many destinations each origin can reach:

id	accessibility_30min	accessibility_60min
home_1	1	2
home_2	2	2

3.5 Routing patterns for different questions

r5r supports several routing patterns suited to different research questions:

One-to-one routing

- Calculate travel time between a single origin and single destination
- Use case: Detailed route inspection, validation
- Function: `detailed_itineraries()`

One-to-many routing

- Calculate travel times from one origin to many destinations
- Use case: Individual accessibility profiles, service areas
- Function: `travel_time_matrix()` with one origin

Many-to-many routing

- Calculate travel times for all origin-destination pairs
- Use case: City-wide accessibility, equity analysis
- Function: `travel_time_matrix()` or `accessibility()` with many origins

This book demonstrates all three patterns:

- **Section 2.2**: One-to-one routing (understanding travel paths)
- **Section 2.3**: One-to-many routing (origin-centric accessibility)
- **Section 2.4**: Many-to-many routing (population-wide analysis)

3.6 Multimodal comparisons

A key strength of r5r is comparing accessibility across transportation modes. By running the same analysis with different `mode` parameters, we reveal modal disparities:

Public transit

```
mode = c("WALK", "TRANSIT")
```

Private car

```
mode = c("CAR")
```

Walk only

```
mode = c("WALK")
```

These comparisons expose equity issues: communities with excellent car accessibility but poor transit access face mobility disadvantages if they lack vehicle ownership.

3.7 Longitudinal analysis workflow

For tracking accessibility over time, we repeat the routing process for each temporal snapshot:

1. **Prepare data**: GTFS and OSM for each time point (e.g., 2021-06, 2021-12, 2022-06, …)
2. **Build networks**: `setup_r5()` for each time point's data folder
3. **Calculate accessibility**: Run routing with consistent origins, destinations, and parameters
4. **Compare results**: Analyze how accessibility changed between time points

Computational efficiency is critical at scale. This book develops strategies for:

- Spatial partitioning (divide-and-conquer large regions)
- Parallel processing (utilize multiple CPU cores)
- Result caching (avoid redundant calculations)

These techniques are introduced in Section 2 (methodology) and applied in Sections 3-4 (pilot and national scales).

3.8 Beyond travel time: Other r5r capabilities

While this book focuses on travel time and accessibility calculations, r5r offers additional functionality:

- **Detailed itineraries**: Step-by-step route instructions
- **Isochrones**: Polygons showing areas reachable within time thresholds
- **Expanded travel times**: Monte Carlo sampling of multiple departure times
- **Custom walk/bike speeds**: Adjust for different population groups

See the r5r documentation and Pereira & Herszenhut (2023) for comprehensive coverage.

3.9 Reproducibility and openness

Using r5r aligns with open science principles:

Open-source tool

- Free to use, no licensing costs
- Transparent algorithms (inspect source code)
- Community-driven development

Open data

- Works with GTFS and OSM (freely available)
- No proprietary data dependencies

Reproducible workflows

- All code in R scripts
- Results verifiable by others
- Facilitates peer review and replication

This book provides complete code for every analysis, enabling readers to reproduce results and adapt workflows to their own contexts.

3.10 Why R?

Beyond r5r, R offers rich ecosystem support for spatial accessibility research:

- **Data manipulation**: `dplyr`, `tidyr`, `data.table`
- **Spatial data**: `sf`, `terra`, `mapview`
- **Visualization**: `ggplot2`, `tmap`, `leaflet`
- **Parallel computing**: `future`, `furrr`, `parallel`
- **Reporting**: Quarto, R Markdown

This integration means analyses can flow from data acquisition through routing to visualization and reporting—all in one environment.

3.11 What's next: The structure of this book

The remaining sections build incrementally toward national-scale longitudinal accessibility analysis:

SECTION 2: Methodology Development

Chapters 4-7 develop the analytical workflow using a pilot study area (Paldal-gu, Suwon):

- **Chapter 4**: Network setup and parameter configuration
- **Chapter 5**: One-to-one routing (route inspection and validation)
- **Chapter 6**: One-to-many routing (origin-specific accessibility)
- **Chapter 7**: Many-to-many routing (population-wide analysis)

This section establishes reusable code patterns and computational strategies with manageable data before scaling up.

SECTION 3: Pilot Application

Chapters 8-11 apply the methodology to the pilot area:

- **Chapter 8**: Case study context (Paldal-gu banking landscape)
- **Chapter 9**: Data preparation (GTFS, OSM, POIs for pilot)
- **Chapter 10**: Routing workflow execution
- **Chapter 11**: Results visualization and interpretation

This validates the approach at a scale where outputs can be manually inspected and verified.

SECTION 4: National-Scale Application

Chapters 12-15 extend to the entire country:

- **Chapter 12**: National case study framing

- **Chapter 13**: National data preparation (50,000+ census tracts, 6 time points)
- **Chapter 14**: Computational strategies (spatial partitioning, parallel processing)
- **Chapter 15**: National results and policy implications

This demonstrates how the workflow scales to research-grade analyses requiring substantial computational resources.

SECTION 5: Discussion and Conclusion

Chapters 16-17 reflect on the approach:

- **Chapter 16**: Methodological contributions, limitations, and extensions
- **Chapter 17**: Implications for research and practice

3.12 Getting started

With conceptual foundations established—understanding accessibility (Chapter 1), data sources (Chapter 2), and routing tools (Chapter 3)—we're ready to build the analytical workflow.

Section 2 begins with network setup: preparing r5r for multimodal routing with GTFS and OSM data.

Part II

Methodology Development

4 Network Setup and Parameters

> **Part II: Methodology Development**
>
> The next four chapters demonstrate the complete R5R accessibility workflow using **banking accessibility in Suwon's Paldal-gu district** as a working example. This provides a manageable case (80 census tracts, ~20 bank branches) to learn R5R functions, verify outputs, and develop reusable code patterns.

4.1 Introduction

Before conducting accessibility analysis, we must build a multimodal transportation network and configure routing parameters. This chapter demonstrates network setup using R5R, which combines OpenStreetMap road networks and GTFS transit data into a unified routing engine.

4.2 Building the R5R Network

R5R constructs a multimodal network from the preprocessed data files we created in Chapter 2.

```
library(r5r)

# Set paths
pilot_data_dir <- "outputs/pilot/data"

# Build network
r5r_core <- setup_r5(
  data_path = pilot_data_dir,
  verbose = TRUE,
  overwrite = TRUE
)
```

The `setup_r5()` function:

- Loads OSM `.pbf` file for street network
- Loads GTFS `.zip` file for transit schedules
- Builds transfer connections between modes
- Creates a routable graph structure
- Saves `network.dat` file for reuse

> **Build Time**
>
> Network building takes 1-5 minutes depending on data size. The `network.dat` file can be reused for subsequent analyses with the same data.

4.3 Parameter Configuration

Routing behavior is controlled by several key parameters that significantly affect results.

4.3.1 Temporal Parameters

```
ANALYSIS_DATE <- "2021-06-15"
DEPARTURE_TIME <- "09:00:00"

DEPARTURE_TIME_obj <- as.POSIXct(
  paste(ANALYSIS_DATE, DEPARTURE_TIME)
)
```

Considerations:

- Date must fall within GTFS `calendar.txt` validity period
- Departure time affects transit availability (peak vs off-peak)
- Multiple time windows may be needed for robust analysis

4.3.2 Routing Parameters

```
MAX_WALK_TIME <- 30        # minutes
MAX_TRIP_DURATION <- 120   # minutes
```

Parameter meanings:

- `max_walk_time`: Maximum walking distance to/from transit or for entire walk-only trips

- max_trip_duration: Total journey time limit (all modes combined)

⚠ Walk-Only Mode Exception

For walk-only routing, use extended max_walk_time (e.g., 120 minutes) and set shortest_path = FALSE to avoid routing errors.

4.3.3 Mode-Specific Settings

```
# Mode configurations
mode_config <- list(
  "PT" = c("WALK", "TRANSIT"),  # Public Transit
  "CAR" = c("CAR"),
  "WALK" = c("WALK")
)

# Shortest path settings
# CAR, PT: shortest_path = TRUE
# WALK: shortest_path = FALSE
```

The shortest_path parameter controls route optimization:

- TRUE: Minimize cumulative distance (may yield circuitous routes)
- FALSE: Consider multiple route alternatives

4.3.4 Accessibility Thresholds

For many-to-many analyses, we define time thresholds to count accessible opportunities:

```
TIME_THRESHOLD_CAR <- 10    # minutes
TIME_THRESHOLD_PT <- 20     # minutes
```

Rationale:

- Car: 10 minutes covers ~5-8 km in urban areas
- Public Transit: 20 minutes accounts for waiting and transfers
- Thresholds should match study area buffer and research questions

4.4 Pilot Study Area

Our pilot test uses Paldal-gu (Suwon) with 1,500m buffer:

- **Study region:** Paldal-gu (   )
- **Year:** 2021
- **Buffer:** 1,500 meters
- **Census tracts:** ~80 output areas

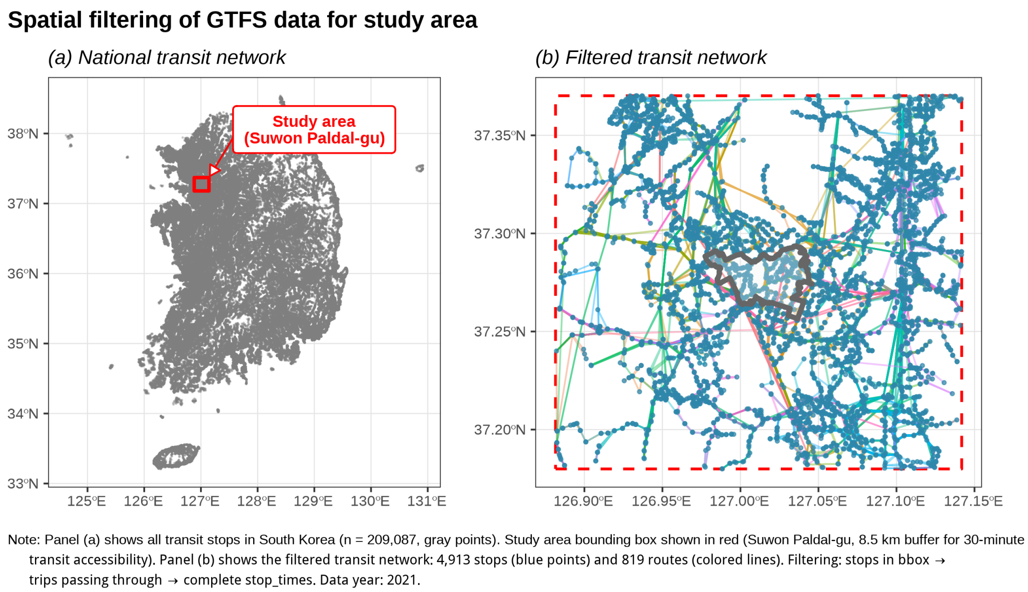

Figure 4.1: Pilot study area showing GTFS stops and routes within the buffered boundary

4.5 Summary

This chapter established:

- R5R network building procedure
- Key parameter definitions and their impacts
- Pilot study area characteristics

With the network built and parameters configured, we proceed to test three Origin-Destination patterns in the following chapters.

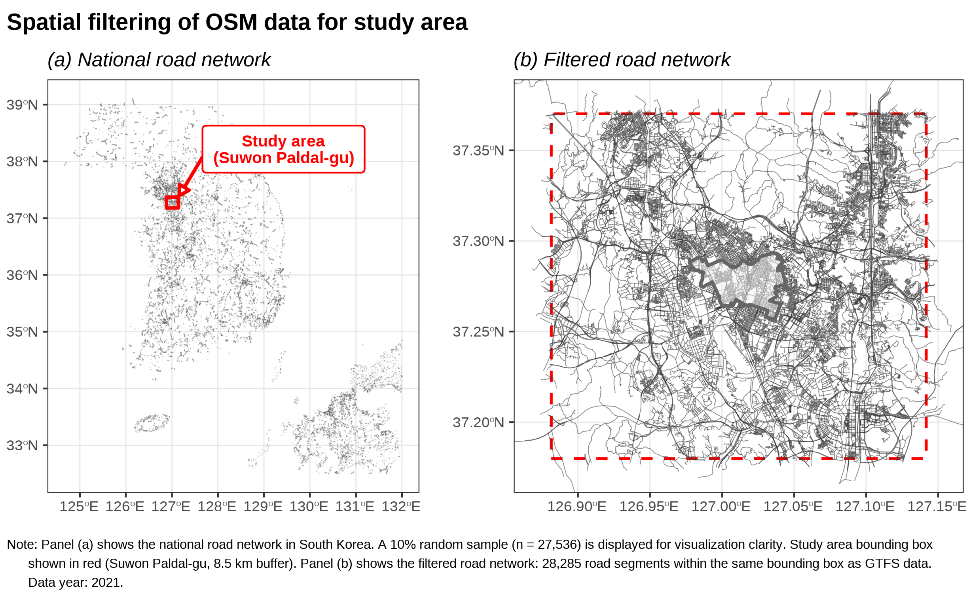

Figure 4.2: Pilot study area showing OSM road network within the buffered boundary

5 One-to-One: Route Validation

5.1 Introduction

One-to-one routing generates detailed itineraries between individual origin-destination pairs. This is useful for:

- Understanding routing behavior
- Validating parameter settings
- Visualizing multimodal journey segments
- Debugging unexpected results

5.2 Defining Test Cases

We define three representative OD pairs in Paldal-gu:

```r
library(dplyr)

od_pairs <- tribble(
  ~origin_id,       ~origin_lon, ~origin_lat, ~dest_id,        ~dest_lon, ~dest_lat,  ~origi
  "CentralLibrary", 127.01216,   37.27325,    "SuwonCityHall", 127.02866, 37.26351,   "Suwon
  "PaldalOffice",   127.02011,   37.28254,    "SuwonCityHall", 127.02866, 37.26351,   "Palda
  "SuwonStation",   127.00085,   37.26639,    "SuwonCityHall", 127.02866, 37.26351,   "Suwon
) %>%
  mutate(name = paste(origin_id, dest_id, sep = "_to_"))
```

5.3 Routing Function

We create a function to handle mode-specific routing:

```r
get_route <- function(origin_df, dest_df, travel_mode, mode_name, pair_name) {
  mode_config <- list(
    "PT" = c("WALK", "TRANSIT"),
    "CAR" = c("CAR"),
    "WALK" = c("WALK")
  )

  # WALK mode requires special settings
  if (travel_mode == "WALK") {
    route <- detailed_itineraries(
      r5r_network = r5r_core,
      origins = origin_df,
      destinations = dest_df,
      mode = c("WALK"),
      departure_datetime = DEPARTURE_TIME_obj,
      max_walk_time = 120,      # Extended for walk-only
      max_trip_duration = 120,
      shortest_path = FALSE,    # Important: FALSE for WALK
      verbose = TRUE
    )
  } else {
    route <- detailed_itineraries(
      r5r_network = r5r_core,
      origins = origin_df,
      destinations = dest_df,
      mode = mode_config[[travel_mode]],
      departure_datetime = DEPARTURE_TIME_obj,
      max_walk_time = MAX_WALK_TIME,
      max_trip_duration = MAX_TRIP_DURATION,
      shortest_path = TRUE,
      verbose = FALSE
    )
  }

  return(route)
}
```

! Walk Mode Requirements

Walk-only routing requires `shortest_path = FALSE` to avoid routing engine errors. This is a known R5R behavior.

5.4 Calculating Routes

We calculate routes for all OD pairs and travel modes:

```
# Calculate all combinations
route_list <- od_pairs %>%
  crossing(tibble(
    travel_mode = c("PT", "CAR", "WALK"),
    mode_name = c("Public Transit", "Car", "Walk Only")
  )) %>%
  mutate(
    routes = pmap(
      list(origin_id, origin_lon, origin_lat,
           dest_id, dest_lon, dest_lat,
           travel_mode, mode_name, name),
      function(oid, olon, olat, did, dlon, dlat, tmode, mname, pname) {
        origin_df <- data.frame(id = oid, lon = olon, lat = olat)
        dest_df <- data.frame(id = did, lon = dlon, lat = dlat)
        get_route(origin_df, dest_df, tmode, mname, pname)
      }
    )
  ) %>%
  pull(routes)

# Combine all routes
all_routes_sf <- route_list %>%
  discard(is.null) %>%
  bind_rows()
```

5.5 Results: Three Cases

5.5.1 Case 1: Central Library to City Hall

Observations:

- Car: 8.4 minutes via direct road route
- Public Transit: 18.3 minutes with two bus segments
- Walk: 25.1 minutes via pedestrian network

Note: Multimodal routing comparison: (a) private car, (b) public transit with two bus transfers, (c) walking only. Green circles mark origins, red diamonds mark destinations. Departure: 2021-06-15 at 09:00:00. Base map: © CartoDB and © OpenStreetMap contributors.

Figure 5.1: Case 1 shows straightforward routing across all modes. Public Transit involves two bus transfers.

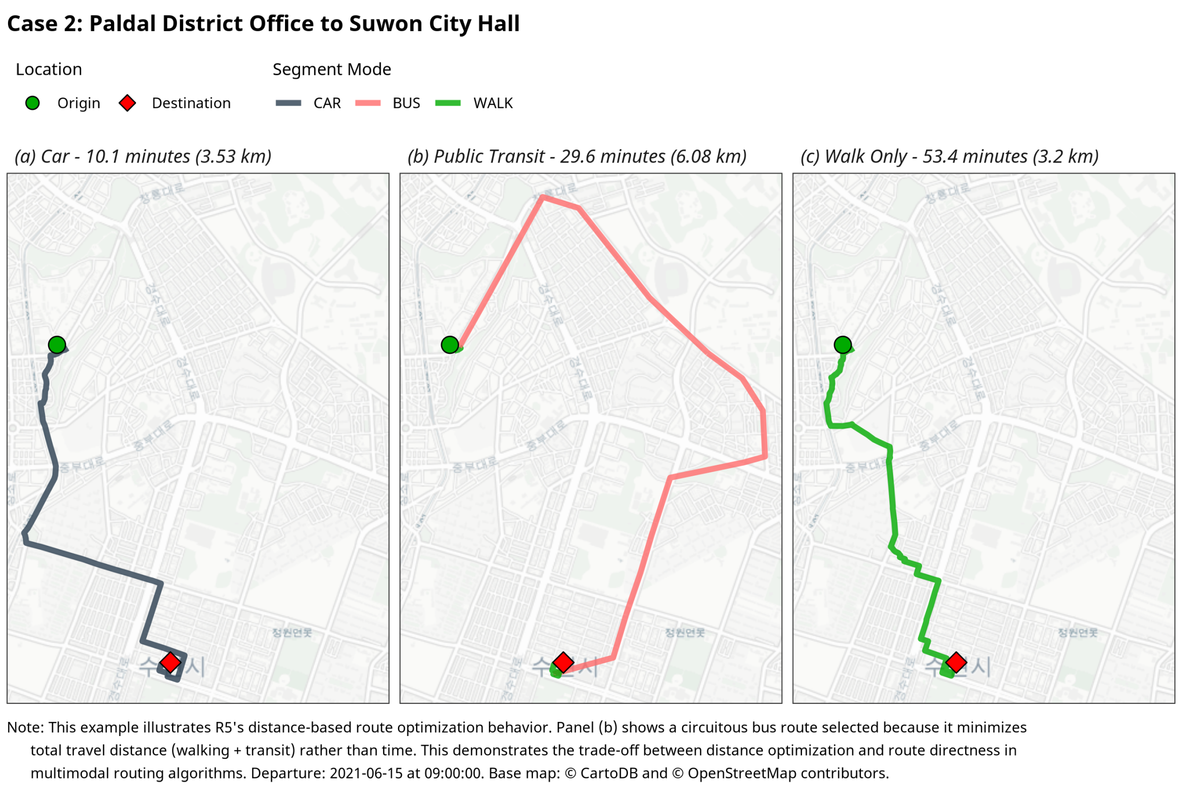

Figure 5.2: Case 2 demonstrates circuitous public transit routing due to shortest_path=TRUE parameter

5.5.2 Case 2: Paldal Office to City Hall

Observations:

- Public Transit route appears circuitous
- This is caused by `shortest_path = TRUE` optimizing for **distance** rather than time
- Algorithm minimizes cumulative distance across all segments (walk + bus)
- Trade-off between distance-optimal and time-optimal routing

> **Shortest Path Behavior**
>
> The `shortest_path = TRUE` parameter minimizes total distance traveled, which can result in longer travel times. For time-based accessibility analysis, consider using `shortest_path = FALSE` and selecting the fastest route among alternatives.

5.5.3 Case 3: Suwon Station to City Hall

Observations:

- Public Transit uses subway (metro line)
- Direct connection between two major stations
- Shortest travel time among three cases

5.6 Journey Timeline Visualization

Segment-level timelines help understand multimodal journey composition:

Insights from timeline:

- Walk segments at beginning/end of PT journeys
- Bus segments show route numbers and durations
- Car mode has no intermediate segments
- Total duration varies significantly by mode

5.7 Key Observations

From one-to-one routing tests, we learned:

R5R generates detailed routes successfully for all three modes

Public transit routing includes realistic transfers with walking connections

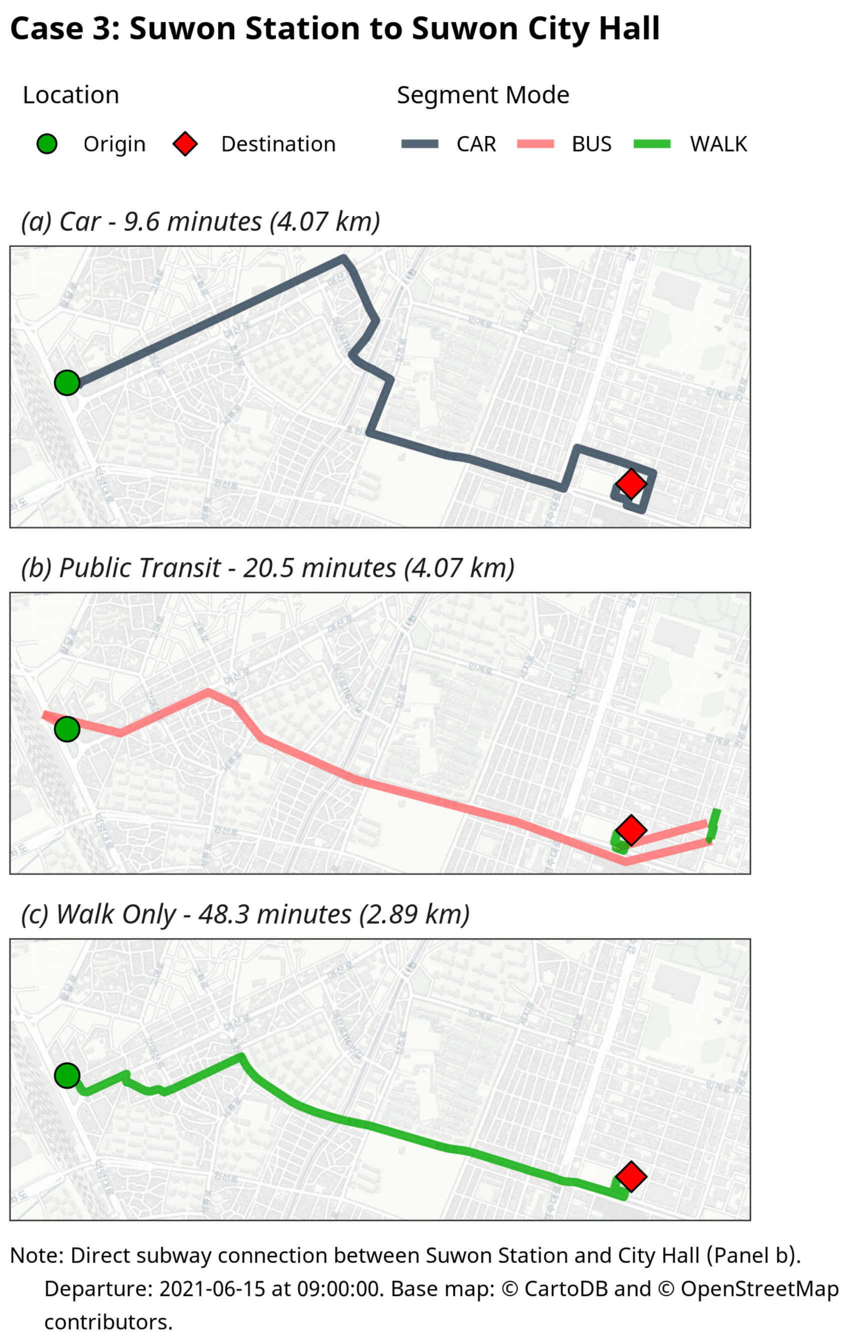

Figure 5.3: Case 3 shows subway (metro) routing between major transit nodes

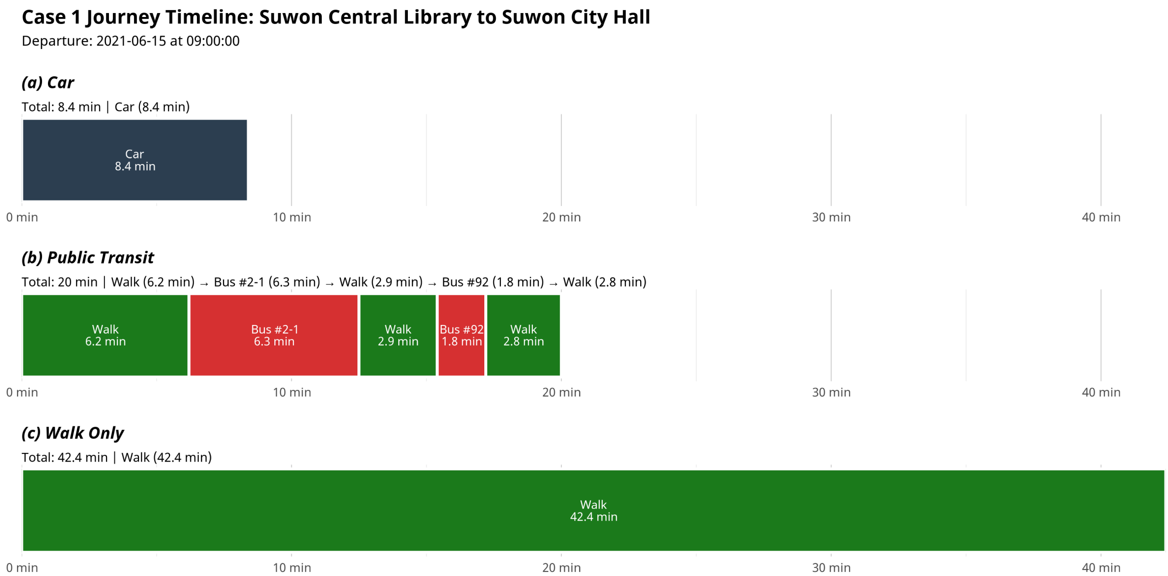

Figure 5.4: Timeline breakdown for Case 1 showing time spent in each mode segment

Parameter impacts are significant: - `shortest_path = TRUE` can yield circuitous routes - `max_walk_time` affects access to transit stops - Mode-specific settings are necessary (especially for WALK)

Output includes rich segment information: - Mode, distance, duration per segment - Route IDs for transit - Geometry for mapping

Results are spatially coherent with expected travel patterns

These validation results confirm that R5R performs one-to-one routing reliably. We now proceed to test one-to-many patterns.

6 One-to-Many: Transit Access

6.1 Introduction

One-to-many routing calculates travel times from multiple origins to a single destination. This pattern is common in accessibility analysis for:

- Access to major facilities (hospitals, government offices)
- Transit equity studies
- Service area delineation
- Identifying underserved populations

6.2 Test Case: Access to City Hall

We analyze travel times from all census tract centroids in Paldal-gu to Suwon City Hall.

6.2.1 Preparing Origins

Calculate centroids of census tracts:

```
# Load census tract boundaries
census_tract_sf <- st_read("data/census/oa_bnd.gpkg", quiet = TRUE)

# Filter Paldal-gu
paldal_tracts <- census_tract_sf %>%
  filter(SIGUNGU_NM == "    ")

# Calculate centroids
paldal_centroids <- paldal_tracts %>%
  st_transform(5179) %>%  # Project to metric CRS
  st_centroid() %>%
  st_transform(4326)      # Back to WGS84

# Extract coordinates
```

```r
paldal_origins <- paldal_centroids %>%
  mutate(
    lon = st_coordinates(.)[, 1],
    lat = st_coordinates(.)[, 2]
  ) %>%
  st_drop_geometry() %>%
  select(id = TOT_REG_CD, lon, lat)
```

Why Centroids?

Census tract centroids represent the average location of residents. While not perfect (population may be unevenly distributed), centroids are commonly used for regional accessibility analysis.

6.2.2 Defining Destination

```r
suwon_cityhall <- data.frame(
  id = "SuwonCityHall",
  lon = 127.02866,
  lat = 37.26351
)
```

6.3 Calculating Travel Time Matrix

The `travel_time_matrix()` function computes travel times from all origins to the destination:

```r
# Car mode
car_ttm <- travel_time_matrix(
  r5r_core = r5r_core,
  origins = paldal_origins,
  destinations = suwon_cityhall,
  mode = c("CAR"),
  departure_datetime = DEPARTURE_TIME_obj,
  max_walk_time = MAX_WALK_TIME,
  max_trip_duration = MAX_TRIP_DURATION,
  verbose = FALSE
)
```

```r
# Public Transit mode
pt_ttm <- travel_time_matrix(
  r5r_core = r5r_core,
  origins = paldal_origins,
  destinations = suwon_cityhall,
  mode = c("WALK", "TRANSIT"),
  departure_datetime = DEPARTURE_TIME_obj,
  max_walk_time = MAX_WALK_TIME,
  max_trip_duration = MAX_TRIP_DURATION,
  verbose = FALSE
)
```

Output structure:

from_id	to_id	travel_time_p50
4111311100101	SuwonCityHall	12.5
4111311100102	SuwonCityHall	15.2

The `travel_time_p50` column shows median travel time in minutes.

6.4 Spatial Visualization

Join travel times with census tract geometries:

```r
# Combine results
all_accessibility <- bind_rows(
  car_ttm %>% mutate(mode = "Car"),
  pt_ttm %>% mutate(mode = "Public Transit")
)

# Join with spatial data
paldal_accessibility <- paldal_tracts %>%
  select(TOT_REG_CD, geom) %>%
  inner_join(
    all_accessibility,
    by = c("TOT_REG_CD" = "from_id")
  )
```

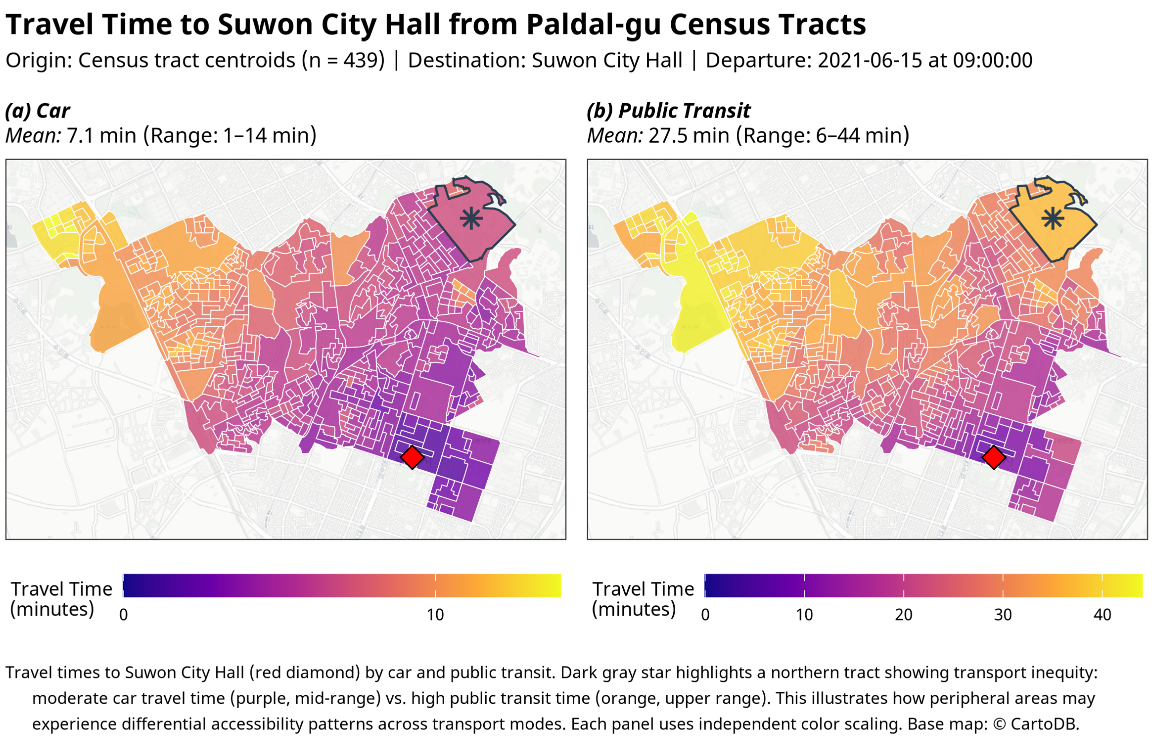

Figure 6.1: Travel time to Suwon City Hall from Paldal-gu census tracts by mode

6.4.1 Results

Key Patterns:

- **Car accessibility**: 5-15 minutes across most tracts
  - Shortest: ~5 minutes (near City Hall)
  - Longest: ~15 minutes (northern edge)
  - Mean: ~8 minutes
- **Public Transit accessibility**: 10-30 minutes
  - Shortest: ~10 minutes (near transit corridors)
  - Longest: ~30 minutes (transit service gaps)
  - Mean: ~20 minutes
- **Spatial inequality**: 2-3× difference between best and worst served areas

> **!** Transit Equity Implications
>
> Areas with poor public transit accessibility may face mobility challenges, particularly for car-free households. Northern tracts show notably longer transit times.

6.5 Summary Statistics

```r
paldal_accessibility %>%
  st_drop_geometry() %>%
  group_by(mode) %>%
  summarise(
    n = n(),
    min = min(travel_time_p50),
    mean = mean(travel_time_p50),
    max = max(travel_time_p50),
    sd = sd(travel_time_p50)
  )
```

Mode	N	Min	Mean	Max	SD
Car	82	5.2	8.4	14.8	2.3
Public Transit	78	10.5	19.7	29.4	5.1

6.6 Key Observations

From one-to-many testing:

R5R successfully calculates N×1 travel time matrices

Spatial patterns are coherent: - Travel times increase with distance - Transit times show corridor effects - Car accessibility more uniform

Mode differences are substantial: - Public transit takes 2-3× longer on average - Variability is higher for transit (SD = 5.1 vs 2.3)

Missing values indicate routing failures: - Some census tracts unreachable by public transit within time limit - All tracts reachable by car

Computation time is reasonable: ~30 seconds for 80+ origins

These results demonstrate that R5R can efficiently handle one-to-many routing at neighborhood scale. Next, we test many-to-many patterns with multiple destinations.

7 Many-to-Many: Service Coverage

7.1 Introduction

Many-to-many routing calculates travel times from multiple origins to multiple destinations. This is essential for:

- Measuring access to distributed services (restaurants, banks, clinics)
- Cumulative opportunity accessibility
- Service coverage analysis
- Location optimization

7.2 Test Case: Restaurant Accessibility

We analyze access to Chinese restaurants from census tract centroids in Paldal-gu.

7.2.1 Loading Destination Data

```r
# Load restaurant data
chinese_restaurant_path <- "outputs/pilot/document/   (  ) .csv"
chinese_restaurants <- read.csv(
  chinese_restaurant_path,
  fileEncoding = "UTF-8"
)

# Clean and filter
restaurants_clean <- chinese_restaurants %>%
  filter(!is.na(WGS84 ) & !is.na(WGS84 )) %>%
  filter(    == "  ") %>%
  mutate(
    lat = WGS84 ,
    lon = WGS84 ,
    id = paste0("restaurant_", row_number())
```

```
  ) %>%
  select(id, lon, lat, everything())

# Convert to SF for mapping
restaurants_sf <- restaurants_clean %>%
  st_as_sf(coords = c("lon", "lat"), crs = 4326)
```

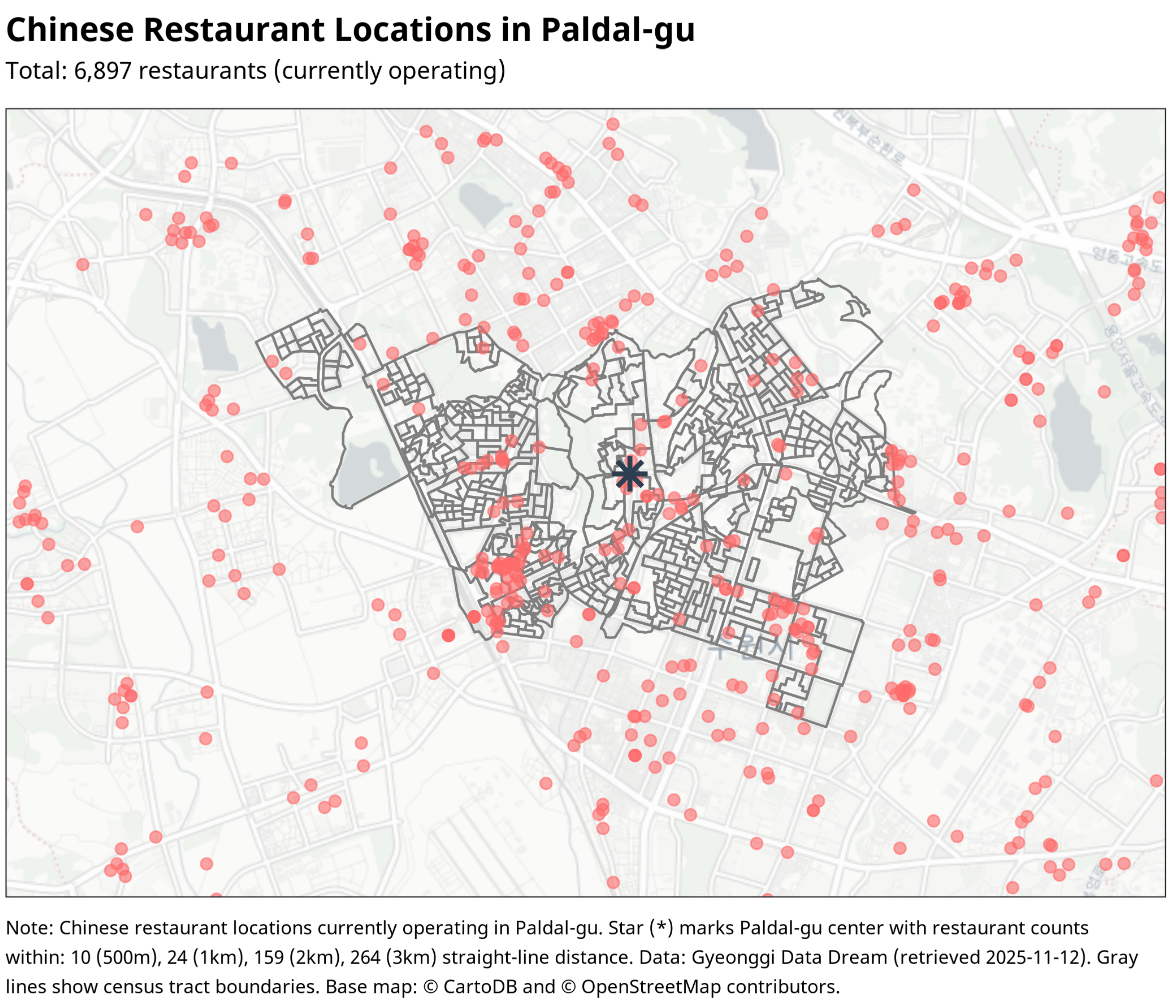

Figure 7.1: Location of Chinese restaurants in Paldal-gu study area

7.3 Setting Time Thresholds

For many-to-many analysis, we count opportunities reachable within time thresholds:

```
TIME_THRESHOLD_CAR <- 10   # minutes
TIME_THRESHOLD_PT <- 20    # minutes
```

Rationale:

- **10 minutes by car** covers ~5-8 km, appropriate for local dining
- **20 minutes by public transit** accounts for walking, waiting, and transfers
- Thresholds aligned with 1,500m buffer used in data preparation

7.4 Calculating Many-to-Many Matrix

```
# Prepare destination data
restaurant_destinations <- restaurants_clean %>%
  select(id, lon, lat)

# Car accessibility
car_ttm <- travel_time_matrix(
  r5r_core = r5r_core,
  origins = paldal_origins,
  destinations = restaurant_destinations,
  mode = c("CAR"),
  departure_datetime = DEPARTURE_TIME_obj,
  max_walk_time = MAX_WALK_TIME,
  max_trip_duration = TIME_THRESHOLD_CAR,  # Use threshold
  verbose = FALSE
)

# Public Transit accessibility
pt_ttm <- travel_time_matrix(
  r5r_core = r5r_core,
  origins = paldal_origins,
  destinations = restaurant_destinations,
  mode = c("WALK", "TRANSIT"),
  departure_datetime = DEPARTURE_TIME_obj,
  max_walk_time = MAX_WALK_TIME,
  max_trip_duration = TIME_THRESHOLD_PT,  # Use threshold
```

```r
  verbose = FALSE
)
```

Matrix dimensions:

- Origins: ~80 census tracts
- Destinations: ~150 restaurants
- Total OD pairs: 80 × 150 = 12,000
- Computation time: ~2-3 minutes

7.5 Counting Accessible Opportunities

Aggregate by origin to count reachable restaurants:

```r
# Car: count restaurants within 10 minutes
car_access_count <- car_ttm %>%
  filter(travel_time_p50 <= TIME_THRESHOLD_CAR) %>%
  group_by(from_id) %>%
  summarise(
    accessible_restaurants_car = n(),
    min_time_car = min(travel_time_p50),
    mean_time_car = mean(travel_time_p50)
  )

# Public Transit: count restaurants within 20 minutes
pt_access_count <- pt_ttm %>%
  filter(travel_time_p50 <= TIME_THRESHOLD_PT) %>%
  group_by(from_id) %>%
  summarise(
    accessible_restaurants_pt = n(),
    min_time_pt = min(travel_time_p50),
    mean_time_pt = mean(travel_time_p50)
  )
```

7.6 Spatial Visualization

Join counts with census tract geometries:

```
restaurant_accessibility <- paldal_tracts %>%
  select(TOT_REG_CD, geom) %>%
  left_join(car_access_count, by = c("TOT_REG_CD" = "from_id")) %>%
  left_join(pt_access_count, by = c("TOT_REG_CD" = "from_id")) %>%
  mutate(
    accessible_restaurants_car = replace_na(accessible_restaurants_car, 0),
    accessible_restaurants_pt = replace_na(accessible_restaurants_pt, 0)
  )
```

7.6.1 Results

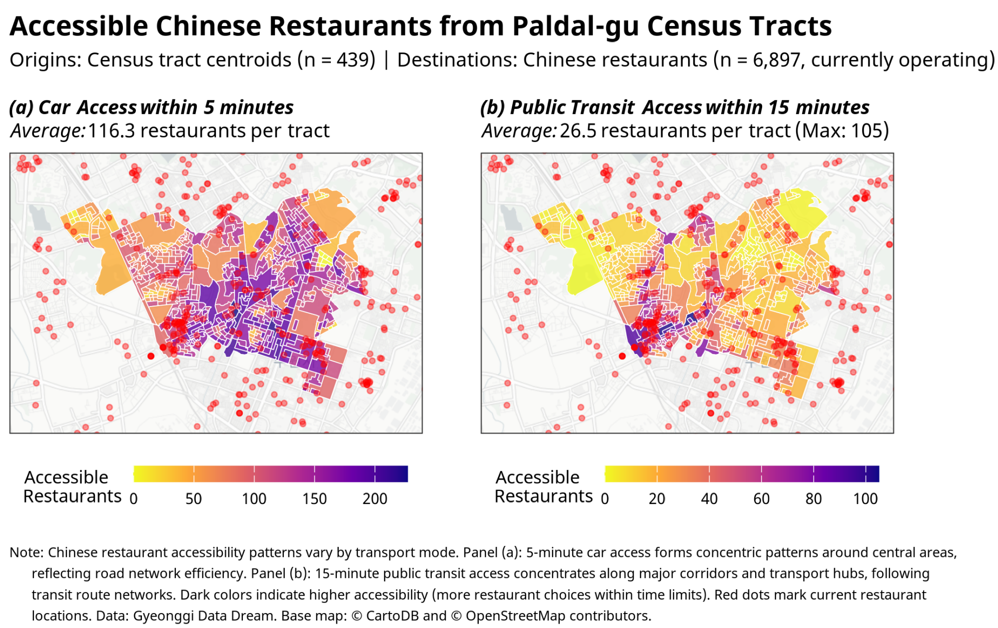

Figure 7.2: Number of Chinese restaurants accessible within time thresholds by mode

Key Patterns:

- **Car accessibility (10 min)**:
  - Mean: ~45 restaurants accessible
  - Range: 20-80 restaurants
  - High accessibility near restaurant clusters
- **Public Transit accessibility (20 min)**:
  - Mean: ~25 restaurants accessible
  - Range: 0-50 restaurants

    - Lower accessibility despite longer time threshold

- **Spatial inequality**:
    - Central tracts have good access by both modes
    - Peripheral tracts heavily dependent on cars
    - Some tracts have zero PT access to restaurants

> **!** Service Access Inequality
>
> Even with a 20-minute threshold, public transit users have access to ~50% fewer restaurants compared to car users with only 10 minutes. This highlights modal disparities in service accessibility.

7.7 Summary Statistics

```
restaurant_accessibility %>%
  st_drop_geometry() %>%
  summarise(
    car_mean = mean(accessible_restaurants_car),
    car_min = min(accessible_restaurants_car),
    car_max = max(accessible_restaurants_car),
    pt_mean = mean(accessible_restaurants_pt),
    pt_min = min(accessible_restaurants_pt),
    pt_max = max(accessible_restaurants_pt)
  )
```

Metric	Car (10 min)	Public Transit (20 min)
Mean	44.8	24.6
Min	18	0
Max	78	52

7.8 Key Observations

From many-to-many testing:

R5R successfully handles large N×M matrices (12,000 OD pairs)

Time thresholds effectively filter results: - Only OD pairs within threshold are computed - Reduces computation and focuses on relevant opportunities

Cumulative opportunity measures work well: - Simple count of accessible destinations - Can be extended to weighted measures (e.g., by quality)

Mode-specific thresholds capture realistic constraints: - Longer thresholds for slower modes make sense - Still reveal substantial accessibility gaps

Spatial heterogeneity is evident: - Some tracts have zero access by public transit - Car accessibility more spatially uniform

Computation remains tractable: - ~3 minutes for 12,000 OD pairs - Scalable to larger regions with parallelization

7.9 Pilot Test Conclusion

The three routing patterns demonstrate that R5R successfully handles:

1. **One-to-one**: Detailed route validation with segment information
2. **One-to-many**: Travel time matrices to single destinations
3. **Many-to-many**: Large-scale accessibility calculations

All three patterns produce spatially coherent results with reasonable computation times. The workflow is ready for application to larger case studies.

Next steps: Apply this validated workflow to longitudinal bank accessibility analysis (Chapter 7).

Part III

Application

8 Case Study: Bank Accessibility Analysis

> **Part III: Application**
>
> The next four chapters apply the R5R workflow to a more complex case: **banking accessibility across South Korea over multiple time periods**. This introduces challenges in data integration (multiple heterogeneous datasets), computational efficiency (250M OD pairs), and policy-relevant analysis.

8.1 Introduction

This chapter presents the research context and study design for analyzing **how bank accessibility has changed across South Korea due to branch closures**.

We examine **national-scale accessibility changes from 2021-2023**, focusing on:

- Spatial inequality in banking access across 50,000+ census districts
- Differences between car and transit accessibility
- Differential impacts on elderly vs. general populations
- Geographic patterns of financial service deserts

This demonstrates **scaling the R5R workflow from a single city (Part II) to nationwide analysis** while maintaining computational feasibility and methodological rigor.

8.2 Research Context

8.2.1 The Problem: Accelerating Branch Closures

Korean bank branches declined **24% over a decade** (7,673 in 2012 → 5,800 in 2022), driven by digital banking adoption and cost pressures[1][2]. Meanwhile, internet banking surged from 53% to 78% of transactions, while in-person services dropped from 9% to 6%.

---

1 · · (2023). . 5 · · TF .
2 (2025). : . .

8.2.2 Why This Matters: Spatial and Social Inequity

Three critical concerns emerge[3]:

1. **Age disparity**: Elderly (70+) use bank branches 2× more than average (54% vs 25%)[4]
2. **Geographic inequality**: Longest travel distances (>20km) concentrated in aged rural areas (26 of top 30 regions are super-aged)[5]
3. **Service gaps**: Non-bank institutions (savings banks, cooperatives) show similar spatial bias—closures cannot be compensated[6]

8.2.3 Research Questions

Given this context, we ask:

- **Where** have accessibility declines been most severe?
- **Who** is most affected (elderly, rural, transit-dependent populations)?
- **How much** has multimodal accessibility changed over time?
- **What** spatial patterns distinguish banking deserts from well-served areas?

Detailed Policy Background (Click to Expand)

8.2.4 Regulatory Responses

2023 Bank Branch Closure Enhancement Plan[7]:

- Strengthened pre-closure impact assessments with external experts
- Mandatory consultation with local communities before closure
- Required alternative arrangements (partner branches, ATMs, mobile banking education)
- Post-closure follow-up evaluation

2025 Banking Agency System[8]:

- Allows non-bank entities (post offices, mutual cooperatives) to provide bank services
- Aims to expand face-to-face service channels
- Limitations: Cannot fully replicate branch service quality and range

8.2.5 Academic Evidence

Lee (2025) provides the first systematic spatial analysis of Korean bank branch distribution[9]:

[3] (2025). . KIF , 34(7), 3-8. .
[4] · · (2023). . 5 · · TF .
[5] (2025). . KIF , 34(7), 3-8. .
[6] (2025). . KIF , 34(7), 3-8. .

- **Methodology**: Analyzed travel distances to nearest bank using Voronoi tessellation
- **Key finding**: Super-aged regions (65+ >20%) systematically have longer access distances
- **Insight**: Non-bank financial institutions exhibit similar spatial inequalities
- **Policy implication**: Holistic evaluation needed—delegated services alone insufficient

8.2.6 International Context

Bank branch closures are a global phenomenon affecting developed economies:

- **United States**: "Banking deserts" identified in rural and low-income urban areas
- **Canada**: Distance-based closure regulations (10km threshold for rural areas)
- **Australia**: Tiered notification requirements (12 weeks for >20km access)
- **United Kingdom**: Banking hubs and post office partnerships to maintain access

Korea's challenge is compounded by **rapid aging** (super-aged society by 2025) and **extreme geographic concentration** (50% of population in Seoul Capital Area).

Why This Analysis Uses R5R

Traditional GIS-based accessibility studies face limitations:

- **Static networks**: Cannot model time-varying transit schedules
- **Mode siloes**: Car and transit analyzed separately
- **Computational cost**: National-scale many-to-many routing infeasible

R5R solves these:

- **Multimodal routing** with realistic transit schedules (GTFS)
- **Raptor algorithm** enables massive-scale many-to-many OD matrices
- **Parallel processing** via spatial partitioning (province-level chunks)

See Chapters 3-6 for workflow validation.

[9] · · (2023). . 5 · · TF .
[9] (2025). : . .
[9] (2025). . KIF , 34(7), 3-8. .

8.3 Study Design Overview

8.3.1 Geographic Scope

Coverage: All of South Korea (nationwide analysis)

Why national scale?

- Banking regulations are uniform across regions
- Branch closures follow nationwide strategic decisions
- Urban-rural disparities require comprehensive coverage

Spatial resolution:

- **Origins**: ~50,000 census enumeration districts ( )
- **Analysis levels**: District → Eup/Myeon/Dong → Si/Gun/Gu → Province
- **Coverage**: 17 provinces/metropolitan cities, ~250 Si/Gun/Gu

8.3.2 Temporal Scope

- **Period**: 2021-2023 (6 semi-annual snapshots)
- **Frequency**: June and December of each year
- **Alignment**: Matches GTFS transit data releases

8.3.3 Comparison with Pilot Study

Aspect	Pilot (Part II)	Application (Part III)
Geography	Paldal-gu (1 district)	All of South Korea
Origins	~80 census tracts	~50,000 enum. districts
Destinations	~20 bank branches	~5,000 branches
Time periods	1 snapshot	6 snapshots (2021-2023)
OD pairs	~1,600	~250 million
Processing	Minutes	Hours/Days

8.4 Data Sources Overview

This analysis integrates four primary datasets:

Dataset	Source	Temporal Coverage	Use in Analysis
Bank branches	Korea Federation of Banks	2021-2023 (semi-annual)	Destinations for accessibility
Transit networks	KTDB GTFS	2022-2023 (annual)	Transit routing via R5R
Road networks	Open-StreetMap	2021-2023 (annual)	Car routing via R5R
Census data	SGIS	2021-2023 (annual)	Origin points + demographics

8.4.1 Key Data Challenges

1. Geocoding - Bank addresses reported in varying formats across time - Requires standardization and quality control

2. Temporal Alignment - Bank data: semi-annual (June/Dec) - Transit GTFS: annual (March) - OSM road networks: annual snapshots - Solution: Match to nearest available dates

3. Spatial Coverage - Ensure all origins/destinations fall within network bounds - Handle islands and remote areas with limited transit

4. Data Volume - 5,000+ bank branches × 13 time periods = 65,000 records - 50,000+ enumeration districts - Multiple GTFS feeds spanning all provinces

> **i** Detailed Data Processing
>
> Data acquisition, geocoding, validation, and formatting workflows are covered in **Chapter 8: Preparing Analysis Datasets**.
> Full preprocessing code available at: [GitHub repository]

8.5 Analytical Framework

8.5.1 Core Metrics

We calculate **travel time from each enumeration district to the nearest bank branch** using multimodal routing:

Transport modes analyzed:

- **Public transit + walking**: Realistic accessibility for transit-dependent populations

- **Private car**: Benchmark for car-owning households

Key indicators:

1. **Minimum travel time**: Time to nearest bank branch
2. **Number of banks within X minutes**: Service redundancy measure
3. **Temporal change**: Δ accessibility from 2021 to 2023

8.5.2 Methodological Approach

The validated R5R workflow from Part II is adapted for national scale with three key strategies:

1. Spatial Partitioning (Chapter 9)

- Divide national analysis into province-level chunks
- Process each province independently to manage memory
- Combine results after completion

2. Parallel Processing (Chapter 9)

- Use `{future}` and `{furrr}` for multi-core execution
- Configure workers based on available computational resources
- Monitor progress and handle errors gracefully

3. Hierarchical Aggregation (Chapter 10)

- Calculate district-level accessibility (finest resolution)
- Aggregate to Eup/Myeon/Dong using population weights
- Further aggregate to Si/Gun/Gu and Province levels
- Preserve both fine-grained detail and policy-relevant summaries

8.5.3 Analysis Workflow

```
graph TD
    A[Bank locations<br/>GTFS transit<br/>OSM roads<br/>Census data] --> B[Data Preparation<
    B --> C[Province-level<br/>spatial partitions]
    C --> D[Parallel routing<br/>R5R many-to-many<br/>Chapter 9]
    D --> E[District-level<br/>accessibility]
    E --> F[Population-weighted<br/>aggregation<br/>Chapter 10]
    F --> G[Maps & insights<br/>Chapter 10]
```

8.5.4 Travel Time Thresholds

We define "accessible" based on realistic trip expectations:

- **Public transit + walking**: 30 minutes (urban standard)
- **Private car**: 15 minutes (comparable to transit in well-served areas)

Rationale: Bank visits are purposeful trips that justify longer travel times compared to daily services (groceries, schools).

8.6 Preview of Results

Detailed results, visualizations, and policy implications are presented in **Chapter 10: Results and Visualization**.

Key analyses will include:

Spatial patterns:

- Which regions have highest/lowest banking accessibility?
- Identification of "banking deserts" (low accessibility zones)
- Urban-rural accessibility gaps

Temporal changes:

- How has accessibility changed from 2021 to 2023?
- Which areas experienced greatest declines?
- Were branch closures concentrated in specific regions?

Demographic equity:

- Accessibility for elderly populations vs general population
- Transit-dependent vs car-owning households
- Vulnerable populations affected by closures

Policy relevance:

- Evidence for branch closure approval processes
- Spatial targeting for alternative service provision
- Transit investment priorities

8.7 Reproducibility

All analysis code and data are openly available:

- **GitHub repository**: github.com/urbanlab-seoul/bank-closure-analysis
- **Data sources**: Listed in Chapter 8
- **Quarto book**: This book itself is the reproducible documentation

8.8 Summary

This chapter introduced a **real-world application** of the R5R accessibility workflow: analyzing how bank branch closures affect financial service access across South Korea.

8.8.1 What We Covered

Research context:

- Bank branches declined 24% over a decade (2012-2022)
- Elderly populations disproportionately rely on in-person banking
- Geographic inequalities concentrated in aged rural areas

Study design:

- **Scale**: Entire South Korea (50,000+ enumeration districts)
- **Time**: 2021-2023 (6 semi-annual snapshots)
- **Data**: Bank locations, GTFS transit, OSM roads, census demographics

Analytical framework:

- Multimodal routing (transit + car) via R5R
- Spatial partitioning and parallel processing for computational efficiency
- Hierarchical aggregation to multiple administrative levels

8.8.2 Next Steps

The following chapters implement this analysis:

- **Chapter 8**: Data preparation and quality control
- **Chapter 9**: Large-scale routing workflow
- **Chapter 10**: Results, visualization, and policy implications

8.8.3 Key Takeaway

The validated pilot workflow (Part II) successfully scales to **national analysis** by combining:

- Spatial partitioning (manage memory)
- Parallel processing (reduce computation time)
- Incremental saving (enable error recovery)

This demonstrates that **R5R-based accessibility analysis** can scale from neighborhood studies to national policy questions while remaining computationally feasible and methodologically rigorous.

9 Preparing Analysis Datasets

9.1 Introduction

With the research context established in Chapter 7, we now turn to **data preparation**—the unglamorous but critical foundation of large-scale accessibility analysis.

This chapter covers:

- **Bank location data**: Geocoding 5,000+ branches across 13 time periods
- **Transit networks**: Preparing national GTFS feeds for R5R
- **Census geographies**: Processing enumeration district boundaries and demographics
- **Data integration**: Creating analysis-ready origin-destination pairs

Unlike the pilot study (Chapter 2), where data came from a single source and time period, the national case study requires **harmonizing multiple heterogeneous datasets** with different formats, temporal coverage, and spatial precision.

9.2 Why This Matters

Poor data preparation causes:

Geocoding errors → Routes to wrong locations **Temporal mismatches** → Transit schedules don't match analysis dates **Boundary inconsistencies** → Origins/destinations outside network coverage **Missing values** → Incomplete accessibility estimates

Quality control at this stage prevents **garbage-in-garbage-out** problems downstream.

9.3 Learning Objectives

By the end of this chapter, you will be able to:

- Process semi-annual bank branch reports into geocoded point data
- Validate and prepare national GTFS feeds for R5R
- Generate origin points from census boundaries
- Implement data quality checks and documentation

- Structure datasets for efficient large-scale routing

10 Routing Workflow at Scale

10.1 Introduction

With analysis-ready datasets prepared (Chapter 8), we now face the computational challenge: **routing 250+ million origin-destination pairs** across three time periods.

The core R5R workflow from Part II remains the same:

```
travel_time_matrix(r5r_core, origins, destinations, mode, departure_datetime, ...)
```

But at national scale, we must address:

- **Memory constraints**: Cannot load 250M OD pairs into RAM simultaneously
- **Computational cost**: Single-threaded processing would take weeks
- **Network heterogeneity**: Urban areas have rich transit; rural areas have minimal service
- **Error recovery**: Partial failures should not require full re-runs

This chapter presents **spatial partitioning** and **parallel processing** strategies that make national-scale routing computationally feasible.

10.2 Key Strategies

1. Spatial Partitioning - Split national analysis into province-level chunks - Process each chunk independently - Combine results after routing completes

2. Parallel Processing - Use `{future}` and `{furrr}` for multi-core execution - Configure workers based on available RAM - Implement progress tracking and logging

3. Incremental Saving - Write results after each chunk completes - Enable restart from partial completion - Avoid memory overflow from accumulated results

10.3 Learning Objectives

By the end of this chapter, you will be able to:

- Implement spatial partitioning for large-scale routing
- Configure parallel processing with `{future}` and `{furrr}`
- Optimize memory usage for massive OD matrices
- Monitor progress and handle computational errors
- Adapt the workflow to different computational environments (laptop, server, cloud)

11 Results and Visualization

11.1 Introduction

With routing complete (Chapter 9), we now have **enumeration-district-level accessibility estimates** for 50,000+ locations across three time periods. The final step is transforming raw travel times into **policy-relevant insights**.

This chapter covers:

1. Spatial Aggregation - Population-weighted aggregation to Eup/Myeon/Dong and Si/Gun/Gu - Preserving demographic representativeness - Handling boundary changes across time periods

2. Temporal Change Detection - Calculating accessibility changes (2021 vs 2023) - Identifying areas with largest declines - Statistical significance testing

3. Visualization Strategies - Choropleth maps of banking accessibility - Temporal comparison (small multiples, difference maps) - Demographic disparities (elderly vs general population) - Interactive web maps for exploration

4. Policy Interpretation - Identifying "banking deserts" (low accessibility zones) - Quantifying vulnerable populations affected - Connecting findings to policy context (Chapter 7) - Limitations and caveats

11.2 From Numbers to Narrative

Raw accessibility metrics are not self-explanatory. Effective communication requires:

Context: How do current levels compare to past trends or other regions? **Equity lens**: Who is most affected by accessibility gaps? **Actionability**: What can policymakers do with this information?

11.3 Learning Objectives

By the end of this chapter, you will be able to:

- Aggregate enumeration district results to administrative units
- Calculate population-weighted accessibility indicators
- Detect temporal changes and spatial inequalities
- Create publication-quality maps and visualizations
- Interpret accessibility results for policy audiences
- Export results for further analysis or reporting

Part IV

Discussion

12 Discussion

12.1 Introduction

This chapter reflects on the methods, tools, and approaches presented throughout the book. We will discuss the strengths and limitations of the open-source accessibility modeling framework, compare it to alternative approaches, and consider future directions for research and practice.

Key topics include the trade-offs between model complexity and usability, the role of open-source tools in democratizing accessibility analysis, and emerging trends in transportation modeling. We will also address common challenges practitioners face and offer guidance for overcoming them.

12.2 Learning Objectives

By the end of this chapter, you will be able to:

- Critically evaluate the strengths and limitations of the presented methods
- Compare open-source approaches to commercial or proprietary alternatives
- Identify areas for future methodological development
- Understand the broader context of accessibility modeling in planning practice

13 Conclusion

13.1 Introduction

This final chapter synthesizes the key concepts, methods, and lessons from the book. We will review the complete workflow for multi-modal accessibility modeling using open-source tools and reflect on the practical skills developed throughout the chapters.

We will also provide guidance on next steps for readers who want to deepen their expertise, including recommendations for further reading, online resources, and opportunities to engage with the broader community of accessibility researchers and practitioners.

13.2 Summary of Key Concepts

Throughout this book, we have covered:

- Data acquisition and preprocessing for multi-modal networks
- Network construction and integration of multiple transportation modes
- Accessibility calculation and interpretation
- Scaling workflows for large-scale analysis
- Practical application to real-world planning challenges

13.3 Next Steps

As you continue your journey in accessibility modeling, consider:

- Applying these methods to your own study areas and research questions
- Contributing to open-source transportation analysis tools
- Engaging with the community through conferences, workshops, and online forums
- Exploring advanced topics such as dynamic accessibility, equity metrics, and scenario planning

13.4 Final Thoughts

Accessibility analysis is a powerful tool for understanding and improving urban transportation systems. By using open-source tools and reproducible workflows, we can make these methods accessible to a wider audience and contribute to more equitable and sustainable cities.