Shanghai Bus Analysis

Load Required Libraries

```r
library(ggplot2)
library(knitr)
library(dplyr)
library(reticulate)
library(formatR)
library(sf)
library(raster)
library(ggpubr)
library(jpeg)
library(maps)
library(rnaturalearth)
library(classInt)
library(ggmap)
```

Loading Bus Route Data

```r
bus_routes <- st_read("busline.json", quiet = TRUE)
head(bus_routes, 10)
```

```
## Simple feature collection with 2 features and 2 fields
## Geometry type: LINESTRING
## Dimension:     XYZ
## Bounding box:  xmin: 121.3269 ymin: 31.17032 xmax: 121.4876 ymax: 31.23493
## z_range:       zmin: 0 zmax: 0
## Geodetic CRS:  WGS 84
##   Id                            name                       geometry
## 1  0 71 (   -   ) LINESTRING Z (121.4868 31.2...
## 2  1 71 (   -   ) LINESTRING Z (121.3298 31.1...
```

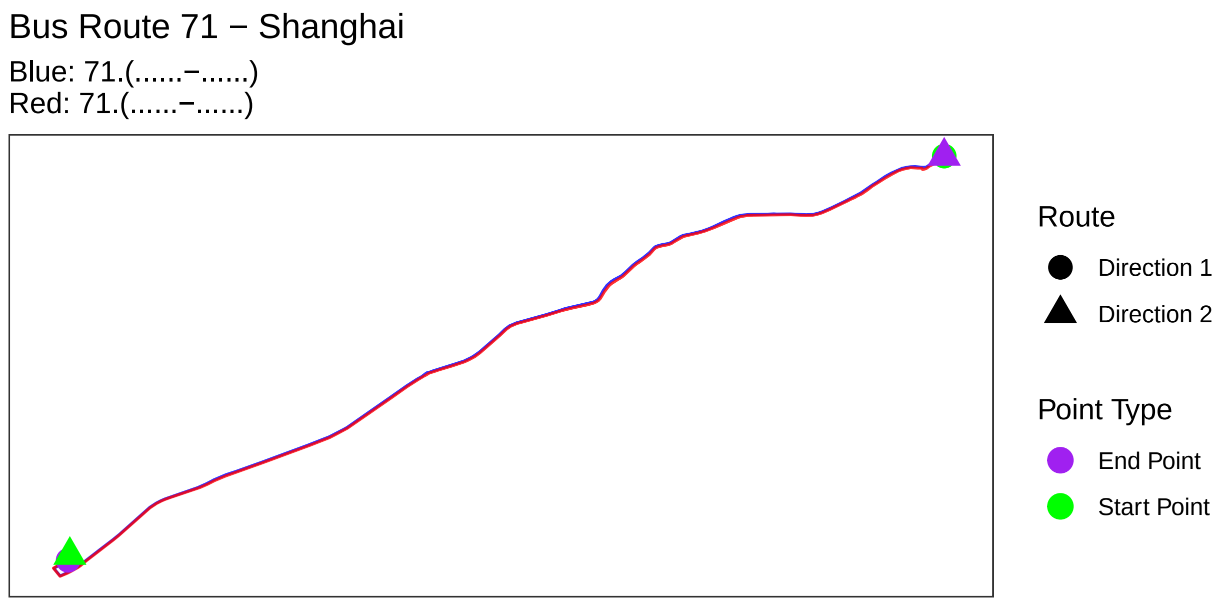
Extracting Bus Route Information

```r
route1 <- bus_routes %>% filter(Id == 0)
route2 <- bus_routes %>% filter(Id == 1)

route1_coords <- st_coordinates(route1)[, 1:2]
route2_coords <- st_coordinates(route2)[, 1:2]

route1_start <- route1_coords[1, ] %>% as.numeric()
```

```r
route1_end <- route1_coords[nrow(route1_coords), ] %>% as.numeric()
route2_start <- route2_coords[1, ] %>% as.numeric()
route2_end <- route2_coords[nrow(route2_coords), ] %>% as.numeric()

start_end_points <- rbind(data.frame(name = "Direction 1 Start", lon = route1_start[1],lat = route1_sta
                          data.frame(name = "Direction 1 End",   lon = route1_end[1],  lat = route1_end
                          data.frame(name = "Direction 2 Start", lon = route2_start[1],lat = route2_sta
                          data.frame(name = "Direction 2 End",   lon = route2_end[1],  lat = route2_end

start_end_points_sf <- st_as_sf(start_end_points, coords = c("lon", "lat"), crs = st_crs(bus_routes))

direction1_name <- bus_routes$name[1]
direction2_name <- bus_routes$name[2]
```

Bus Route Map Visualization

```r
ggplot() +
  geom_sf(data = route1, color = "blue", size = 1.2, alpha = 0.8) +
  geom_sf(data = route2, color = "red", size = 1.2, alpha = 0.8) +
  geom_sf(data = start_end_points_sf, aes(color = type, shape = route), size = 4) +
  scale_color_manual(values = c("start" = "green", "end" = "purple"),
                    labels = c("End Point", "Start Point")) +
  scale_shape_manual(values = c("Direction 1" = 16, "Direction 2" = 17)) +
  theme_bw() +
  labs(title = "Bus Route 71 - Shanghai",
    subtitle = paste0("Blue: ", direction1_name, "\nRed: ", direction2_name),
    color = "Point Type",
    shape = "Route") +
    coord_sf(datum = NA)
```



This map visualization shows the geographic layout of Bus Route 71 in Shanghai, with two different directions shown in blue and red. The route follows a predominantly east-west axis across the city, with start and end points clearly marked.

Route Information Summary

```
route1_length <- st_length(route1) %>% units::set_units(km) %>% round(2)
```

```
## st_as_s2(): dropping Z and/or M coordinate
```

```
route2_length <- st_length(route2) %>% units::set_units(km) %>% round(2)
```

```
## st_as_s2(): dropping Z and/or M coordinate
```

```
route_info <- data.frame(Direction = c(1, 2),
                          Name = c(direction1_name, direction2_name),
                     Length_km = c(route1_length, route2_length),
                    Num_Points = c(nrow(route1_coords), nrow(route2_coords)))

knitr::kable(route_info, caption = "Bus Route 71 Information")
```

Table 1: Bus Route 71 Information

Direction	Name	Length_km	Num_Points
1	71 (-)	17.56 [km]	180
2	71 (-)	17.96 [km]	234

Loading GPS Bus Data

```r
busgps_df <- read.csv("busgps.csv", header = FALSE, stringsAsFactors = FALSE)

colnames(busgps_df) <- c('GPSDateTime', 'LineId', 'LineName', 'NextLevel', 'PrevLevel',
                         'Strlatlon', 'ToDir', 'VehicleId', 'VehicleNo', 'unknow')

busgps_df$GPSDateTime <- as.POSIXct(busgps_df$GPSDateTime)
head(busgps_df)
```

```
##           GPSDateTime LineId LineName NextLevel PrevLevel            Strlatlon
## 1 2019-01-16 23:59:59   7100       71         2         1 121.335413,31.173188
## 2 2019-01-17 00:00:00   7100       71         2         1 121.334616,31.172271
## 3 2019-01-17 00:00:00   7100       71        24        23 121.339955,31.173025
## 4 2019-01-17 00:00:01   7100       71        14        13  121.409491,31.20433
## 5 2019-01-17 00:00:03   7100       71        15        14 121.398615,31.200253
## 6 2019-01-17 00:00:04   7100       71         2         1 121.334853,31.171997
##   ToDir VehicleId VehicleNo unknow
## 1     1  D-R7103  Z5A-0021      1
## 2     1  D-R1273  Z5A-0002      1
## 3     0  D-R5257  Z5A-0020      1
## 4     0  D-R5192  Z5A-0013      1
## 5     0  D-T0951  Z5A-0022      1
## 6     1  D-R6089  Z5A-0001      1
```

Updating Chinese Route Names to English

```r
route_info <- data.frame(
  direction = c(direction1_name, direction2_name),
  start_station = c("   ", "   "),
  end_station = c("   ", "   "),
  route_id = c(0, 1))

knitr::kable(route_info, caption = "Original Chinese Station and Direction Names")
```

Table 2: Original Chinese Station and Direction Names

direction	start_station	end_station	route_id
71 (-)			0
71 (-)			1

```r
route_info <- route_info %>%
  mutate(start_station = case_when(
          start_station == "   " ~ "Yanan Road Station",
          start_station == "   " ~ "Shenkun Station",
                   TRUE ~ start_station),
        end_station = case_when(
          end_station == "   " ~ "Shenkun Station",
          end_station == "   " ~ "Yanan Road Station",
                 TRUE ~ end_station),
        direction_en = case_when(
          route_id == 0 ~ "Yanan Road-Shenkun",
          route_id == 1 ~ "Shenkun-Yanan Road",
                   TRUE ~ as.character(direction)))

knitr::kable(route_info, caption = "Updated English Station and Direction Names")
```

Table 3: Updated English Station and Direction Names

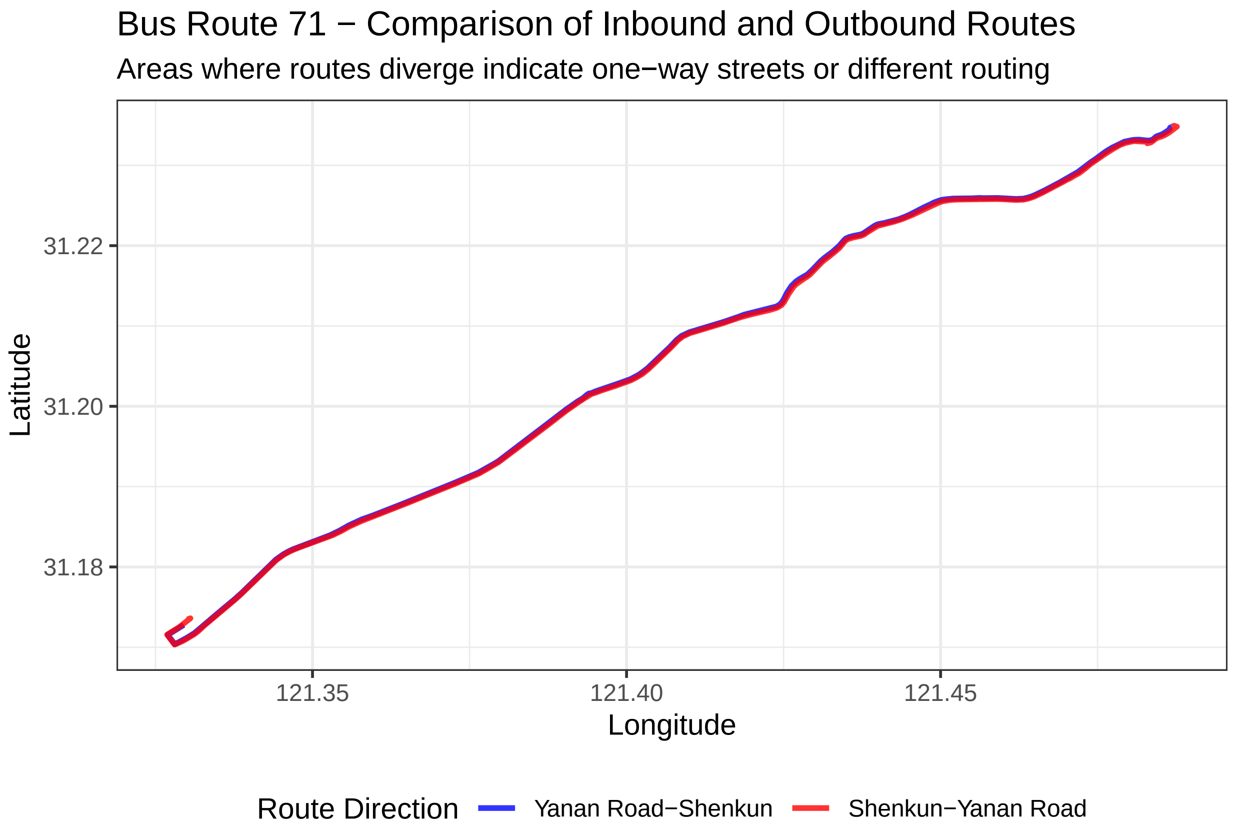
direction	start_station	end_station	route_id	direction_en
71 (-)	Yanan Road Station	Shenkun Station	0	Yanan Road-Shenkun
71 (-)	Shenkun Station	Yanan Road Station	1	Shenkun-Yanan Road

```r
direction1_name_en <- route_info$direction_en[1]
direction2_name_en <- route_info$direction_en[2]
```

Comparison of Inbound and Outbound Routes

```r
route1_df <- data.frame(st_coordinates(route1)[, 1:2]) %>%
  mutate(route = "Direction 1", point_id = 1:n())

route2_df <- data.frame(st_coordinates(route2)[, 1:2]) %>%
  mutate(route = "Direction 2", point_id = 1:n())

combined_routes <- bind_rows(route1_df, route2_df)

ggplot(combined_routes, aes(x = X, y = Y, color = route)) +
  geom_path(size = 1, alpha = 0.8) +
  theme_bw() +
  scale_color_manual(values = c("Direction 1" = "blue", "Direction 2" = "red"),
                    labels = c(direction1_name_en, direction2_name_en)) +
  labs(title = "Bus Route 71 - Comparison of Inbound and Outbound Routes",
    subtitle = "Areas where routes diverge indicate one-way streets or different routing",
           x = "Longitude",
           y = "Latitude",
       color = "Route Direction") +
  theme(legend.position = "bottom")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



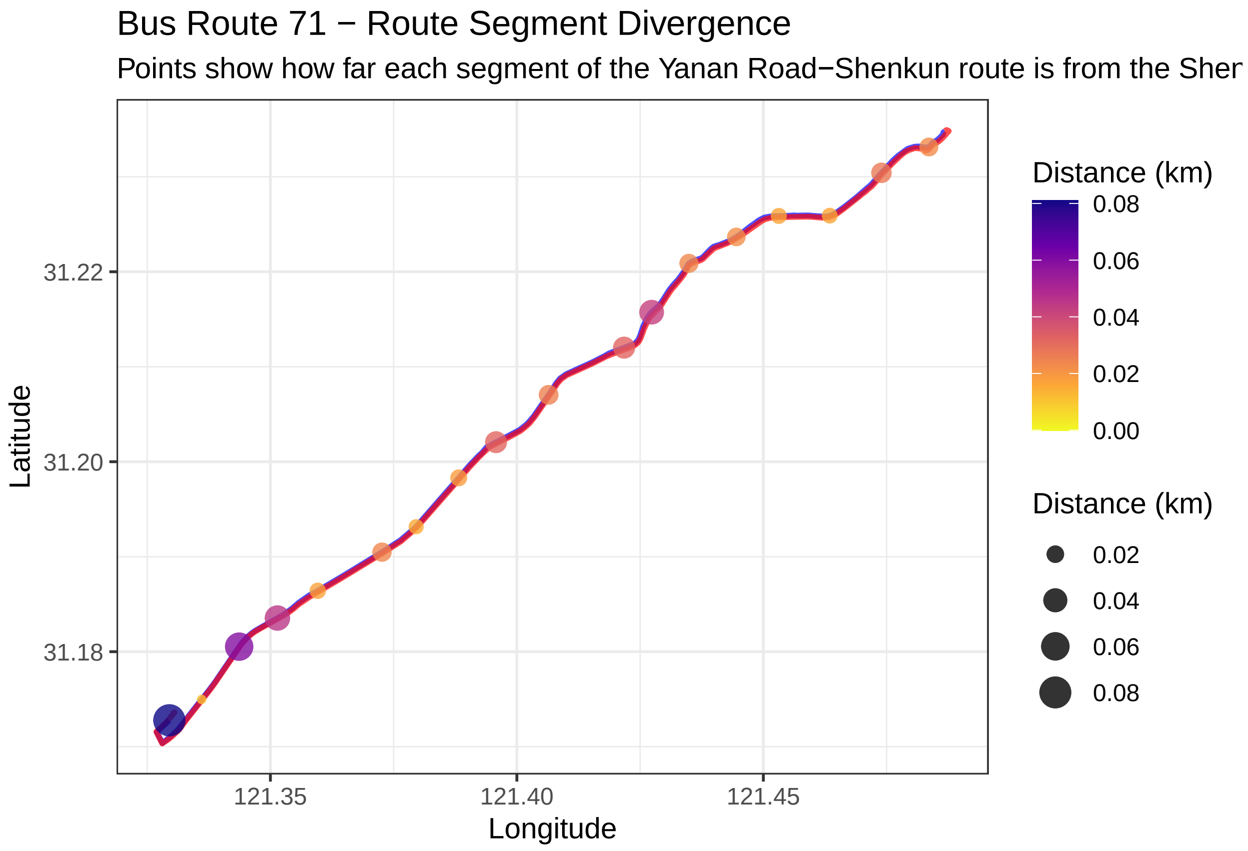
This visualization compares the inbound and outbound routes of Bus 71, highlighting areas where the two directions diverge. The report indicates this comparison helps identify one-way streets and routing differences between the Yanan Road-Shenkun and Shenkun-Yanan Road directions.

Route Divergence Analysis

```
route1_closest <- route1_df %>%
  filter(point_id %% 10 == 0) %>%
  rowwise() %>%
  mutate(
    dist_to_route2 = min(sqrt((X - route2_df$X)^2 + (Y - route2_df$Y)^2)),
    dist_km = dist_to_route2 * 111.32,
    divergence_class = cut(dist_km,
                           breaks = c(0, 0.05, 0.1, 0.2, 0.5, Inf),
                           labels = c("Very Close", "Close", "Moderate", "Far", "Very Far")))
```

Route Segment Divergence Visualization

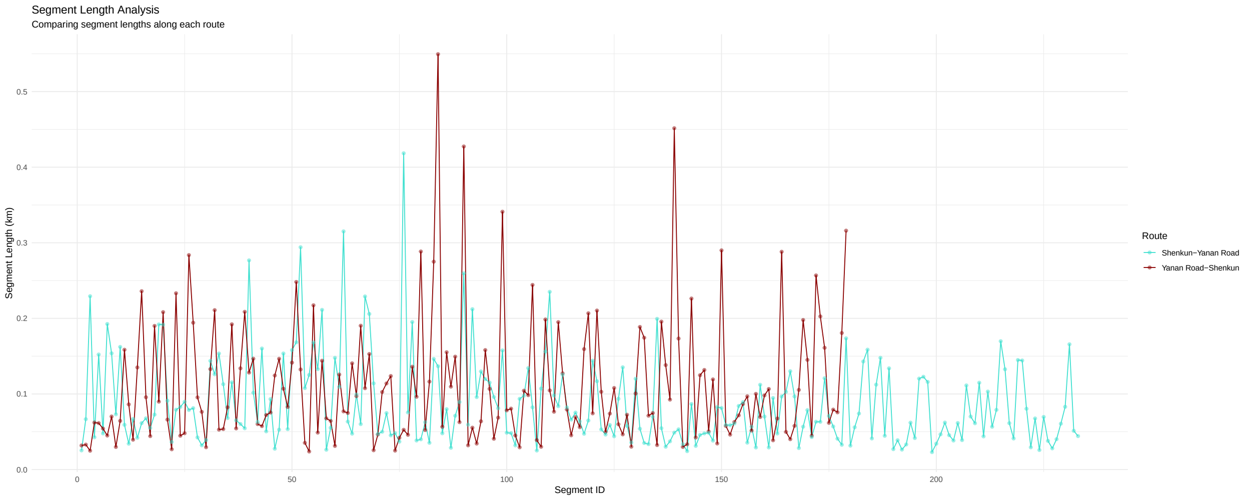
```r
ggplot() +
  geom_path(data = route1_df, aes(x = X, y = Y), color = "blue", size = 1, alpha = 0.7) +
  geom_path(data = route2_df, aes(x = X, y = Y), color = "red", size = 1, alpha = 0.7) +
  geom_point(data = route1_closest,
             aes(x = X, y = Y, color = dist_km, size = dist_km),
             alpha = 0.8) +
  scale_color_viridis_c("Distance (km)",
                        option = "plasma",
                        direction = -1,
                        limits = c(0, max(route1_closest$dist_km))) +
  scale_size_continuous("Distance (km)",
                        range = c(1, 5)) +
  theme_bw() +
  labs(title = "Bus Route 71 - Route Segment Divergence",
    subtitle = paste("Points show how far each segment of the", direction1_name_en, "route is from the"
          x = "Longitude",
          y = "Latitude") +
  guides(color = guide_colorbar(title.position = "top"),
         size = guide_legend(title.position = "top"))
```



This visualization in Figure 17 of the report analyzes the divergence of Bus Route 71 from the Yanan Road path, with larger and darker points indicating greater distances from the reference path. Most significant divergences are in the southwestern part of the route, while points become smaller and lighter as the route continues northeast.

Segment Length Analysis

```r
calculate_segment_properties <- function(route_df) {
  segments <- route_df %>%
    mutate(
      next_X = lead(X),
      next_Y = lead(Y),
      segment_length = sqrt((next_X - X)^2 + (next_Y - Y)^2) * 111.32,
      segment_direction = atan2(next_Y - Y, next_X - X) * 180/pi,
      segment_direction = ifelse(segment_direction < 0, segment_direction + 360, segment_direction)
    ) %>%
    filter(!is.na(segment_length))
  return(segments)}

route1_segments <- calculate_segment_properties(route1_df) %>%
  mutate(route = direction1_name_en,
         segment_id = row_number(),
         cumulative_distance = cumsum(segment_length))

route2_segments <- calculate_segment_properties(route2_df) %>%
  mutate(route = direction2_name_en,
         segment_id = row_number(),
         cumulative_distance = cumsum(segment_length))

all_segments <- bind_rows(route1_segments, route2_segments)

ggplot(all_segments, aes(x = segment_id, y = segment_length, color = route)) +
  geom_line() +
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("turquoise", "darkred")) +
  theme_minimal() +
  labs(
    title = "Segment Length Analysis",
    subtitle = "Comparing segment lengths along each route",
    x = "Segment ID",
    y = "Segment Length (km)",
    color = "Route"
  )
```



This graph shown in Figure 18 compares segment lengths between the Shenkun-Yanan Road and Yanan Road-Shenkun routes. The report notes that segment lengths vary significantly and asymmetrically between directions, with the Yanan Road-Shenkun route (brown line) showing greater variability and higher peaks, while the Shenkun-Yanan Road route (cyan line) has shorter, more consistent segments.

Turn Analysis by Route

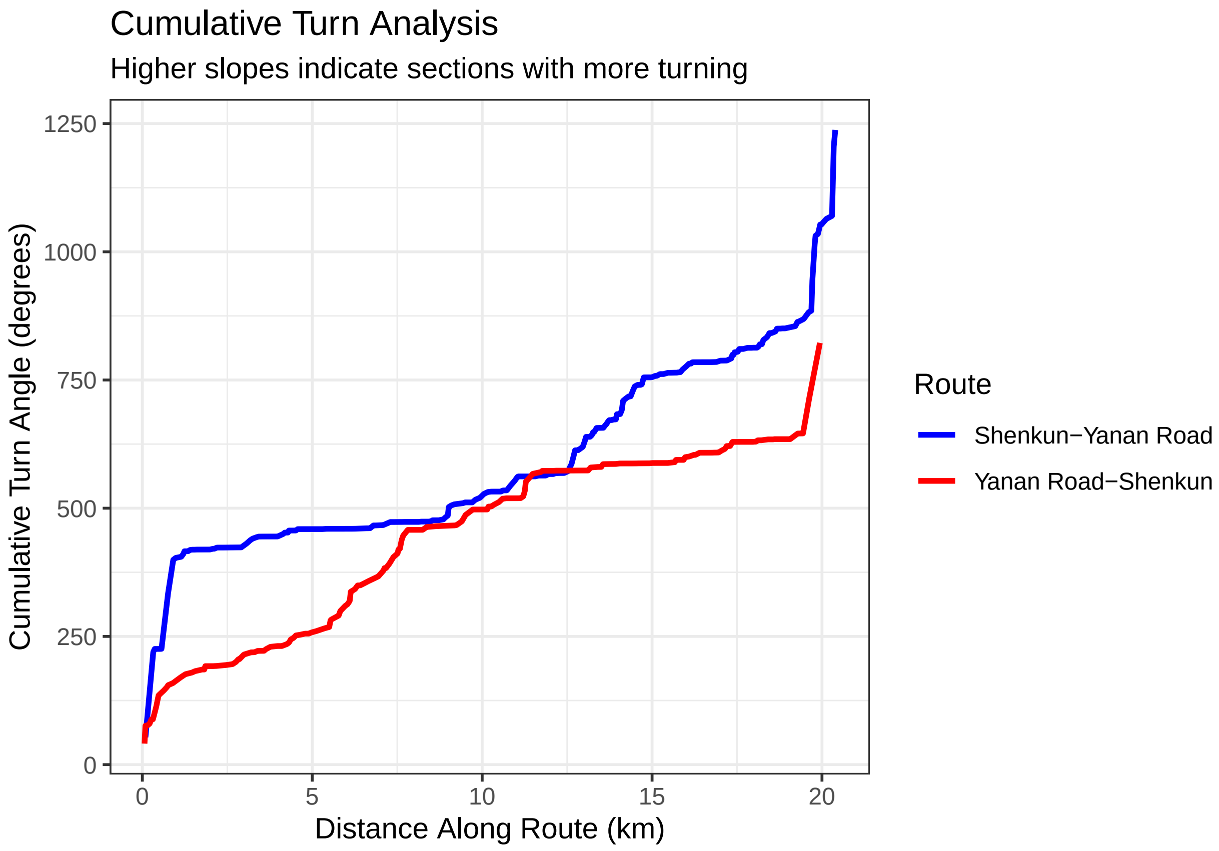
```r
direction_changes <- all_segments %>%
  group_by(route) %>%
  mutate(
    prev_direction = lag(segment_direction),
    turn_angle = abs(((segment_direction - prev_direction + 180) %% 360) - 180),
    significant_turn = turn_angle > 25
  ) %>%
  filter(!is.na(turn_angle))

turn_summary <- direction_changes %>%
  group_by(route) %>%
  summarize(
    total_turns = n(),
    significant_turns = sum(significant_turn),
    avg_turn_angle = mean(turn_angle),
    max_turn_angle = max(turn_angle)
  )

knitr::kable(turn_summary, caption = "Turn Analysis by Route")
```

Table 4: Turn Analysis by Route

route	total_turns	significant_turns	avg_turn_angle	max_turn_angle
Shenkun-Yanan Road	232	8	5.334342	166.1686
Yanan Road-Shenkun	178	4	4.620225	108.7062

Cumulative Turn Analysis

```
direction_changes %>%
  group_by(route) %>%
  mutate(cumulative_turns = cumsum(turn_angle)) %>%
  ggplot(aes(x = cumulative_distance, y = cumulative_turns, color = route)) +
  geom_line(size = 1) +
  scale_color_manual(values = c("blue", "red")) +
  theme_bw() +
  labs(title = "Cumulative Turn Analysis",
    subtitle = "Higher slopes indicate sections with more turning",
           x = "Distance Along Route (km)",
           y = "Cumulative Turn Angle (degrees)",
       color = "Route")
```



According to Figure 19(a) in the report, this analysis shows that the Shenkun-Yanan Road direction has a higher cumulative turn angle (exceeding 1200°) compared to the Yanan-Shenkun direction (under 850°). This suggests that the Shenkun-Yanan route has more frequent and sharper turns, potentially leading to slower travel times.

Travel Time Estimation Model

```r
estimate_segment_speeds <- function(segments_df) {
  segments_df %>%
    group_by(route) %>%
    mutate(
      prev_direction = lag(segment_direction),
      turn_angle = if_else(is.na(prev_direction), 0,
                         abs(((segment_direction - prev_direction + 180) %% 360) - 180)),

      segment_speed_kmh = case_when(
        turn_angle > 45 ~ 10,
        turn_angle > 30 ~ 15,
        turn_angle > 15 ~ 20,
        TRUE ~ 25
      ),

      segment_time_min = (segment_length / segment_speed_kmh) * 60,

      cumulative_time_min = cumsum(segment_time_min)
    ) %>%
    ungroup()
}

route1_timed <- estimate_segment_speeds(route1_segments)
route2_timed <- estimate_segment_speeds(route2_segments)

all_timed_segments <- bind_rows(route1_timed, route2_timed)
```

Estimated Travel Time Visualization

```r
ggplot(all_timed_segments, aes(x = cumulative_distance, y = cumulative_time_min, color = route)) +
  geom_line(size = 1.2) +
  geom_point(data = all_timed_segments %>% filter(segment_id %% 20 == 0), size = 3, alpha = 0.7) +
  scale_color_manual(values = c("blue", "red")) +
  theme_bw() +
  labs(title = "Estimated Travel Time Along Bus Route 71",
    subtitle = "Based on segment lengths and estimated speeds",
           x = "Distance Along Route (km)",
           y = "Cumulative Travel Time (minutes)",
       color = "Route")
```

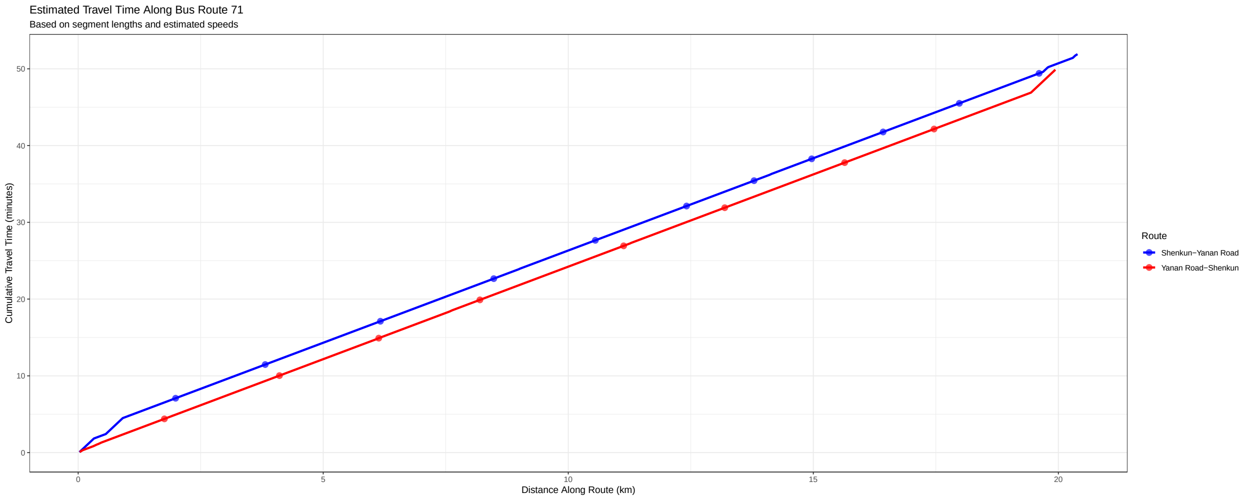
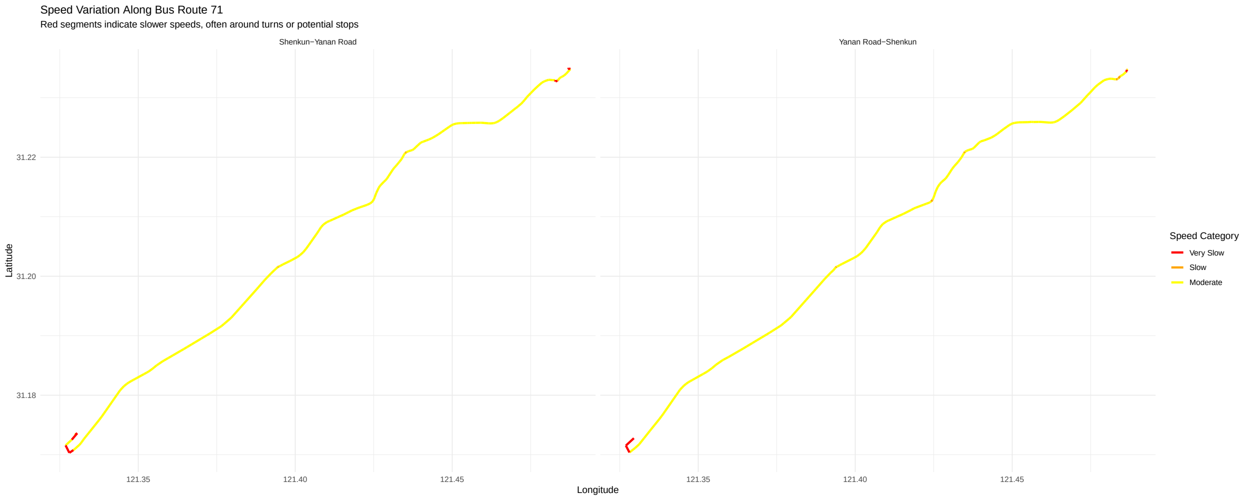


Figure 19(b) shows the estimated travel time along both directions. The report indicates that the Shenkun-Yanan Road direction has a longer estimated travel time (over 52 minutes) compared to the Yanan-Shenkun direction (around 51 minutes). While the absolute difference is small, it's consistent throughout the route, suggesting that route geometry and turns impact service time.

Speed Variation Map

```
slow_threshold <- 20

segment_map_data <- all_timed_segments %>%
  dplyr::select(X, Y, next_X, next_Y, segment_speed_kmh, route) %>%
  mutate(speed_category = cut(segment_speed_kmh,
                              breaks = c(0, 15, 20, 25, Inf),
                              labels = c("Very Slow", "Slow", "Moderate", "Normal")))

ggplot() +
  geom_segment(data = segment_map_data,
               aes(x = X, y = Y, xend = next_X, yend = next_Y,
                   color = speed_category),
               size = 1.2) +
  scale_color_manual(values = c("Very Slow" = "red", "Slow" = "orange",
                                "Moderate" = "yellow", "Normal" = "green")) +
  facet_wrap(~ route) +
  theme_minimal() +
  labs(
    title = "Speed Variation Along Bus Route 71",
    subtitle = "Red segments indicate slower speeds, often around turns or potential stops",
    x = "Longitude",
    y = "Latitude",
    color = "Speed Category"
  )
```



Key Points Analysis

```
sample_interval_km <- 1

extract_key_points <- function(route_df, interval_km = 1) {
  route_df$cum_dist <- c(0, cumsum(sqrt(diff(route_df$X)^2 + diff(route_df$Y)^2))) * 111.32

  max_dist <- max(route_df$cum_dist)
  target_dists <- seq(0, max_dist, by = interval_km)

  key_points <- data.frame(
    target_dist_km = target_dists,
    point_idx = numeric(length(target_dists)),
    stringsAsFactors = FALSE)

  for(i in 1:length(target_dists)) {
    key_points$point_idx[i] <- which.min(abs(route_df$cum_dist - target_dists[i]))
  }

  result <- route_df[key_points$point_idx, ] %>%
    mutate(key_point_id = row_number(),
           key_point_label = paste0(round(cum_dist), " km"))

  return(result)
}

route1_key_points <- extract_key_points(route1_df, sample_interval_km)
route2_key_points <- extract_key_points(route2_df, sample_interval_km)

ggplot() +
  geom_path(data = route1_df, aes(x = X, y = Y), color = "blue", alpha = 0.5) +
  geom_path(data = route2_df, aes(x = X, y = Y), color = "darkred", alpha = 0.5) +
  geom_point(data = route1_key_points, aes(x = X, y = Y), color = "blue", size = 3) +
  geom_text(data = route1_key_points, aes(x = X, y = Y, label = key_point_label),
            color = "darkblue", vjust = -1.5, size = 3) +
  geom_point(data = route2_key_points, aes(x = X, y = Y), color = "darkred", size = 3) +
  geom_text(data = route2_key_points, aes(x = X, y = Y, label = key_point_label),
            color = "darkred", vjust = 1.5, size = 3) +
  theme_bw() +
  labs(title = "Key Points Along Bus Route 71", subtitle = paste("Points sampled at approximately", samp
    x = "Longitude",
    y = "Latitude")
```

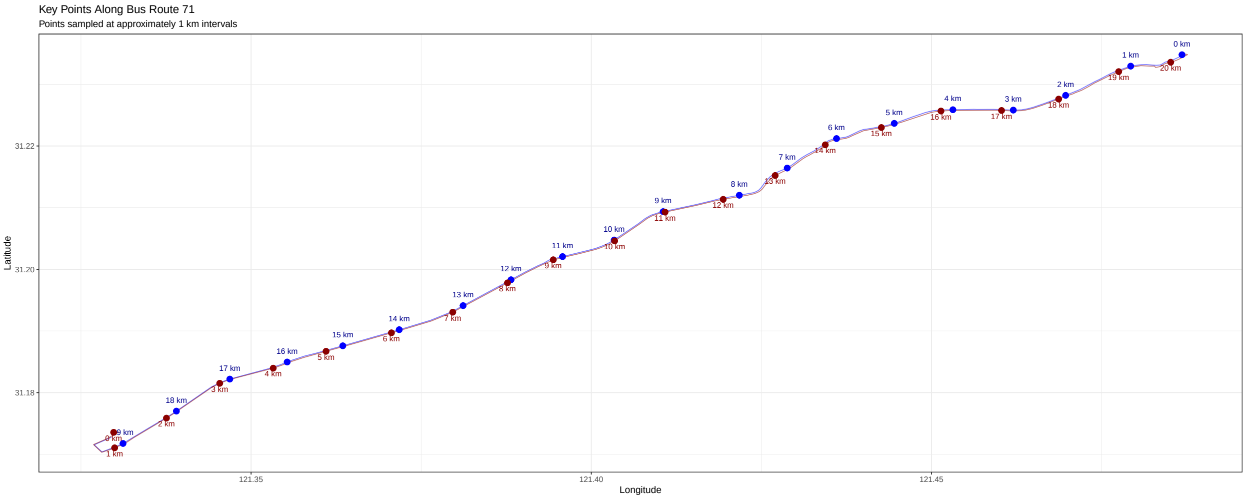


Figure 20 in the report shows key points along Bus Route 71 spaced at 1 km intervals. This visualization helps identify deviations, overlaps, and directional symmetry in the route, which is useful for transport planning and optimization.

Distance Matrix Calculation

```
calculate_distance_matrix <- function(points_df) {
  n <- nrow(points_df)
  dist_matrix <- matrix(0, nrow = n, ncol = n)

  for(i in 1:n) {
    for(j in 1:n) {
      if(i != j) {
        dist_matrix[i, j] <- sqrt(
          (points_df$X[i] - points_df$X[j])^2 +
          (points_df$Y[i] - points_df$Y[j])^2
        ) * 111.32
      }
    }
  }

  rownames(dist_matrix) <- points_df$key_point_label
  colnames(dist_matrix) <- points_df$key_point_label

  return(dist_matrix)
}
```

```
route1_dist_matrix <- calculate_distance_matrix(route1_key_points)
route2_dist_matrix <- calculate_distance_matrix(route2_key_points)

route1_dist_long <- as.data.frame(as.table(route1_dist_matrix)) %>%
  rename(from = Var1, to = Var2, distance_km = Freq)
```

Distance Matrix Visualization

```
ggplot(route1_dist_long, aes(x = from, y = to, fill = distance_km)) +
  geom_tile() +
  scale_fill_viridis_c("Distance (km)", option = "plasma") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        axis.title = element_blank()) +
  labs(title = paste("Distance Matrix Between Key Points -", direction1_name_en),
    subtitle = "Colors indicate the distance (km) between any two points along the route")
```

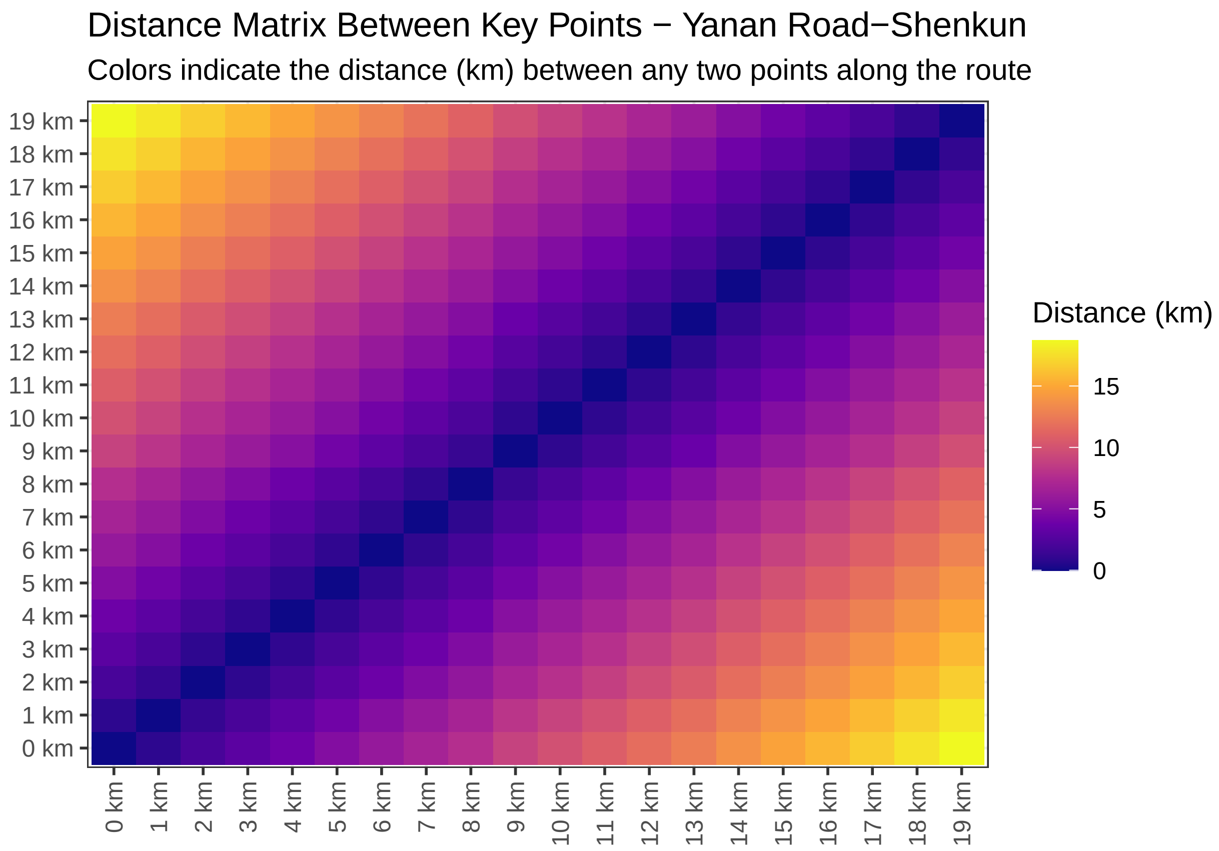


Figure 21 displays a distance matrix between key points along Bus Route 71. The matrix shows pairwise distances between points spaced every 1 km along the route. This visualization helps identify patterns of linearity and irregularity in the route, useful for understanding its spatial structure and finding opportunities for optimization.

Speed Analysis Along Route

```
ggplot(all_timed_segments, aes(x = cumulative_distance, y = segment_speed_kmh, color = route)) +
  geom_line() +
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("blue", "red")) +
  theme_bw() +
  labs(
    title = "Estimated Speeds Along Bus Route 71",
    subtitle = "Showing how speed varies based on segment characteristics",
    x = "Distance Along Route (km)",
    y = "Speed (km/h)",
    color = "Route")
```

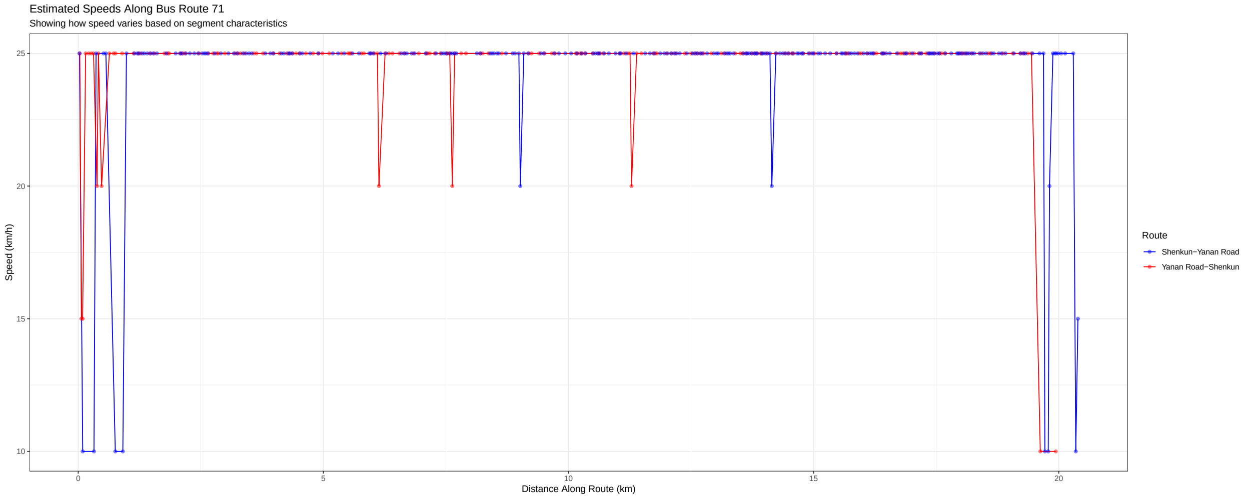


Figure 22 shows estimated speeds along both directions of Bus Route 71. Most segments operate near the maximum speed (25 km/h), but some drop to 10-15 km/h, particularly near the route ends. This suggests areas with signal delays, congestion, or complex intersections.

Travel Time Summary and Route with Time Markers

```
if(!exists("all_timed_segments")) {
  route1_segments <- calculate_segment_properties(route1_df) %>%
    mutate(route = direction1_name_en,
           segment_id = row_number(),
           cumulative_distance = cumsum(segment_length))

  route2_segments <- calculate_segment_properties(route2_df) %>%
    mutate(route = direction2_name_en,
           segment_id = row_number(),
           cumulative_distance = cumsum(segment_length))

  all_segments <- bind_rows(route1_segments, route2_segments)

  estimate_segment_speeds <- function(segments_df) {
    segments_df %>%
      group_by(route) %>%
      mutate(
        prev_direction = lag(segment_direction),
        turn_angle = if_else(is.na(prev_direction), 0,
                           abs(((segment_direction - prev_direction + 180) %% 360) - 180)),

        segment_speed_kmh = case_when(
          turn_angle > 45 ~ 10,
          turn_angle > 30 ~ 15,
          turn_angle > 15 ~ 20,
          TRUE ~ 25
        ),
        segment_time_min = (segment_length / segment_speed_kmh) * 60,

        cumulative_time_min = cumsum(segment_time_min)) %>%
      ungroup()
    }

  all_timed_segments <- estimate_segment_speeds(all_segments)
}

route_time_summary <- all_timed_segments %>%
  group_by(route) %>%
  summarize(total_distance_km = max(cumulative_distance),
               total_time_min = max(cumulative_time_min),
                avg_speed_kmh = total_distance_km / (total_time_min / 60),
            min_segment_speed = min(segment_speed_kmh),
            max_segment_speed = max(segment_speed_kmh))

knitr::kable(route_time_summary, caption = "Estimated Travel Time Summary", digits = 2)
```

Table 5: Estimated Travel Time Summary

route	total_distance_km	total_time_min	avg_speed_kmh	min_segment_speed	max_segment_speed
Shenkun-Yanan Road	20.39	51.91	23.57	10	25
Yanan Road-Shenkun	19.94	49.88	23.98	10	25

```
find_time_markers <- function(route_data) {
  target_times <- seq(5, max(route_data$cumulative_time_min), by = 5)

  result <- data.frame()

  for(target_time in target_times) {
    idx <- which.min(abs(route_data$cumulative_time_min - target_time))

    marker <- route_data[idx, ] %>%
      mutate(time_marker = paste0(round(cumulative_time_min), " min"))

    result <- bind_rows(result, marker)
  }

  return(result)
}

time_markers <- all_timed_segments %>%
  split(.$route) %>%
  lapply(find_time_markers) %>%
  bind_rows()

ggplot() +
  geom_path(data = route1_df, aes(x = X, y = Y), color = "blue", alpha = 0.5, size = 0.8) +
  geom_path(data = route2_df, aes(x = X, y = Y), color = "red", alpha = 0.5, size = 0.8) +
  geom_point(data = time_markers, aes(x = X, y = Y, color = route), size = 3) +
  geom_text(data = time_markers, aes(x = X, y = Y, label = time_marker, color = route),
            vjust = -1.5, size = 3) +
  scale_color_manual(values = c("blue", "red")) +
  theme_minimal() +
  labs(
    title = "Bus Route 71 with Time Markers",
    subtitle = "Estimated travel time markers at 5-minute intervals",
    x = "Longitude",
    y = "Latitude",
    color = "Route"
  )
```

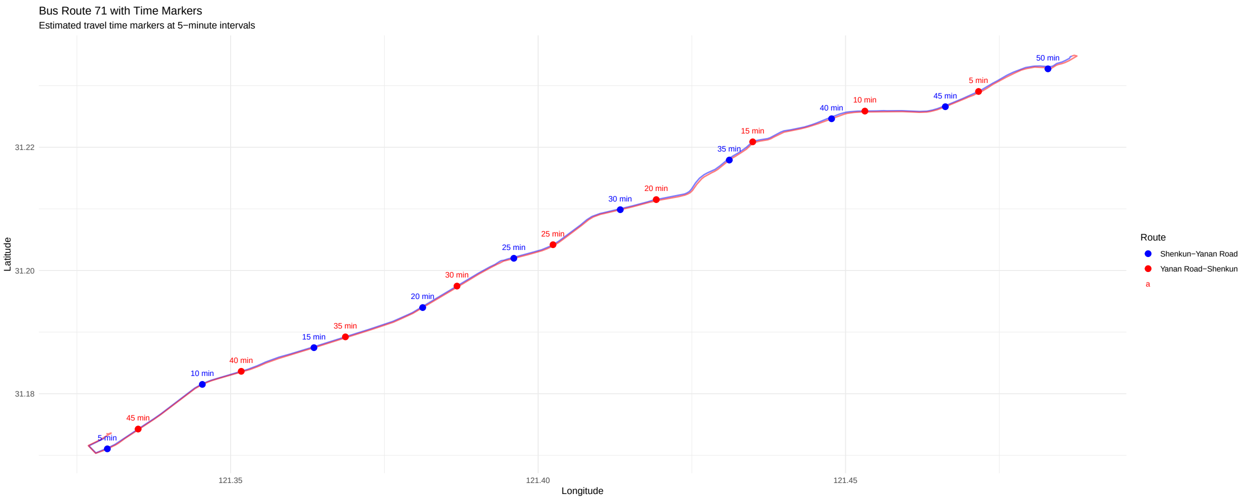


Figure 23 shows estimated travel times mapped along both directions of Route 71, with time markers at 5-minute intervals. The report notes that both directions follow similar paths, but time markers indicate non-uniform travel times across segments, highlighting directional inefficiencies that could be targets for optimization.

Segment Length Distribution by Route

```r
ggplot(all_segments, aes(x = route, y = segment_length, fill = route)) +
  geom_boxplot() +
  scale_fill_manual(values = c("turquoise", "yellow")) +
  theme_bw() +
  labs(
    title = "Distribution of Segment Lengths by Route",
    subtitle = "Comparing segment length characteristics between routes",
    x = "Route",
    y = "Segment Length (km)",
    fill = "Route"
  ) +
  theme(legend.position = "none")
```

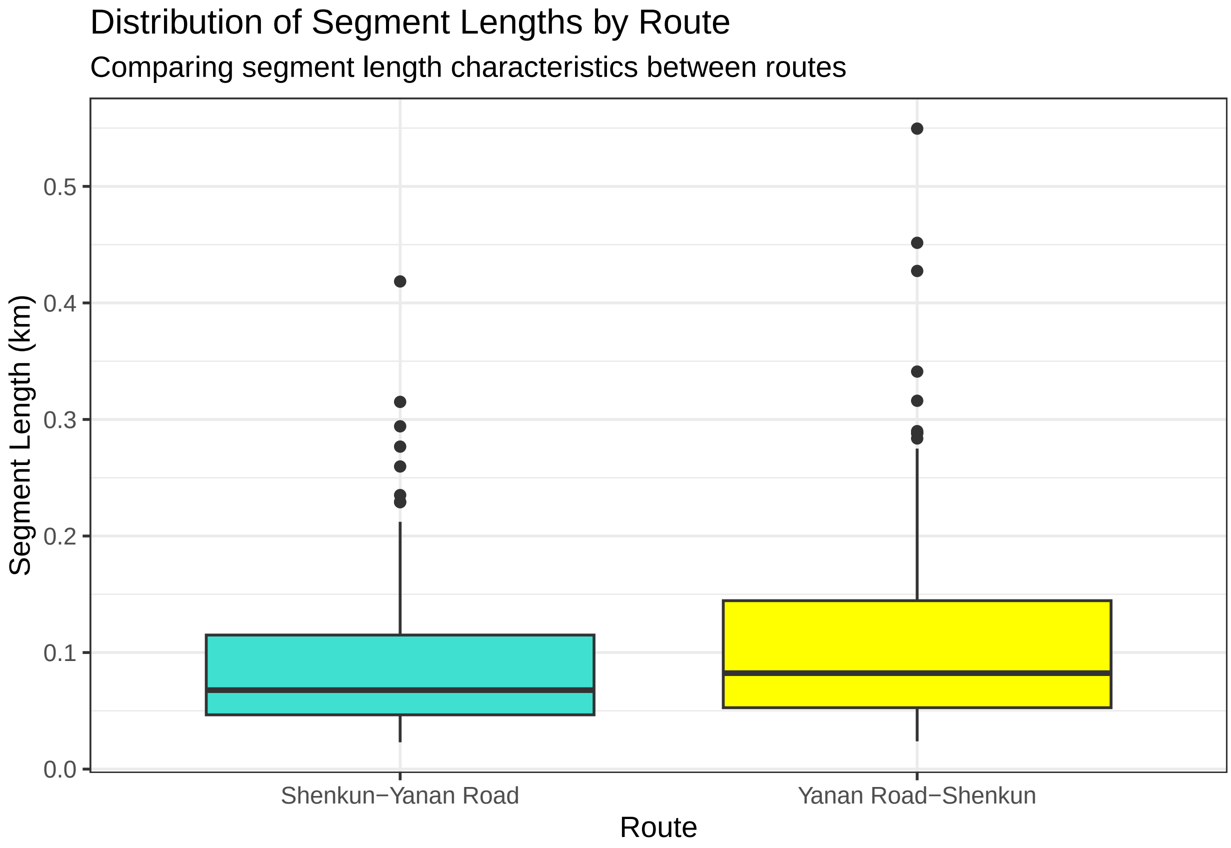


Figure 24 compares the distribution of segment lengths for both directions. The Shenkun-Yanan Road direction (turquoise) shows more uniform segment lengths with a lower median, while the Yanan-Shenkun Road direction (yellow) has greater variability and a higher median segment length. This asymmetry suggests differences in operational segmentation.

Speed Category Distribution by Route

```r
ggplot(segment_map_data, aes(x = route, fill = speed_category)) +
  geom_bar(position = "fill") +
  scale_fill_manual(values = c("Very Slow" = "red", "Slow" = "orange",
                               "Moderate" = "yellow", "Normal" = "green")) +
  theme_bw() +
  labs(title = "Proportion of Speed Categories by Route",
    subtitle = "Comparing the distribution of speed segments between routes",
           x = "Route",
           y = "Proportion",
        fill = "Speed Category") +
  coord_flip()
```

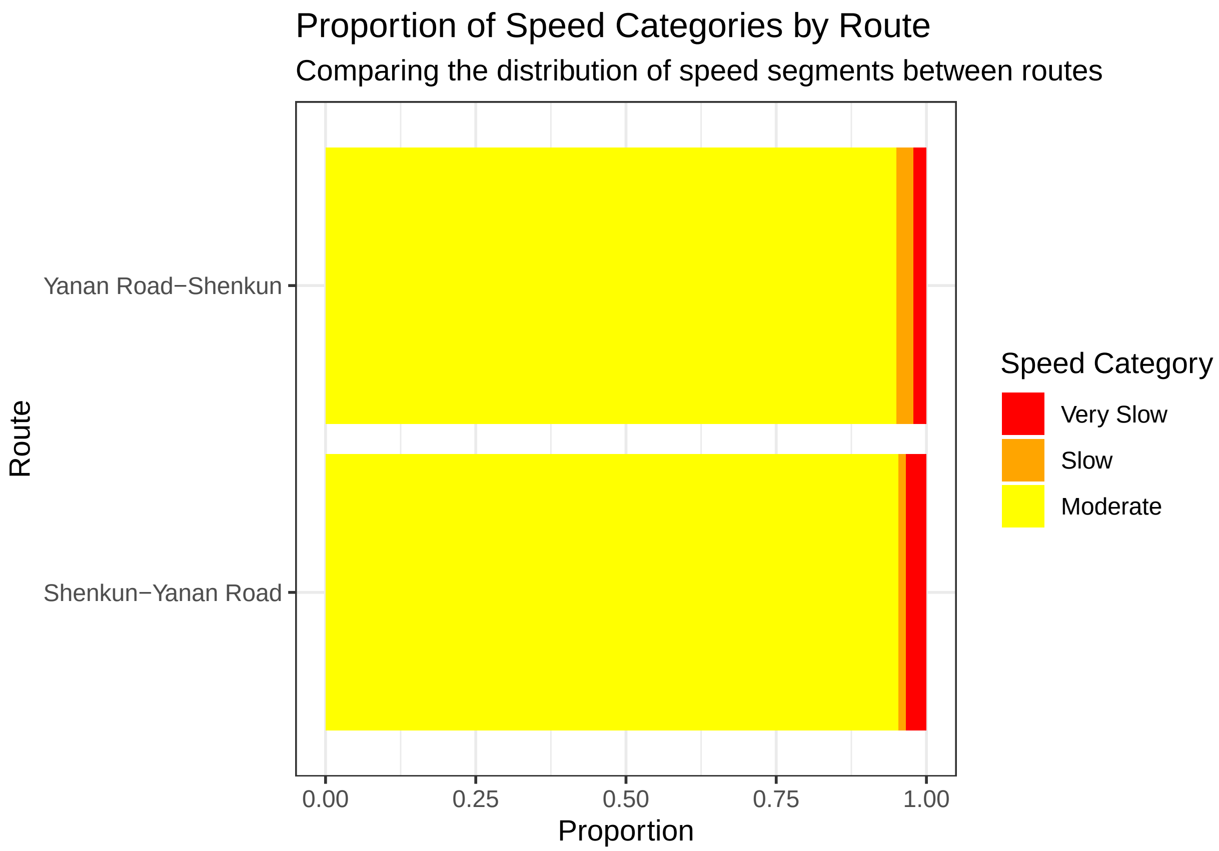


Figure 25 shows the proportion of speed categories along both directions. Moderate speeds (yellow) dominate both directions, but the Yanan-Shenkun direction has slightly more segments in the Slow and Very Slow categories, suggesting occasional congestion or operational issues like passenger boarding delays.

Vehicle ID Mapping

```
busgps_df <- busgps_df %>%
  mutate(jp_vehicleID = VehicleId) %>%
  group_by(VehicleId) %>%
  mutate(temp_id = cur_group_id()) %>%
  ungroup() %>%
  mutate(VehicleId = paste0("Vehicle_", temp_id)) %>%
  subset(select = -temp_id)


vehicle_id_cols <- c("VehicleId", "jp_vehicleID")
vehicle_id_mapping <- unique(busgps_df[, vehicle_id_cols])
vehicle_id_mapping <- vehicle_id_mapping[order(vehicle_id_mapping$VehicleId), ]

knitr::kable(vehicle_id_mapping, caption = "Vehicle ID Mapping Reference")
```

Table 6: Vehicle ID Mapping Reference

VehicleId	jp_vehicleID
Vehicle_1	D-98307
Vehicle_10	D-R2773
Vehicle_11	D-R3531
Vehicle_12	D-R3571
Vehicle_13	D-R5192
Vehicle_14	D-R5257
Vehicle_15	D-R5627
Vehicle_16	D-R6025
Vehicle_17	D-R6051
Vehicle_18	D-R6079
Vehicle_19	D-R6089
Vehicle_2	D-R0725
Vehicle_20	D-R6573
Vehicle_21	D-R6679
Vehicle_22	D-R7103
Vehicle_23	D-R7162
Vehicle_24	D-R7165
Vehicle_25	D-R7265
Vehicle_26	D-R7506
Vehicle_27	D-R7537
Vehicle_28	D-R7622
Vehicle_29	D-R7667
Vehicle_3	D-R0937
Vehicle_30	D-R7673
Vehicle_31	D-R7689
Vehicle_32	D-R7715
Vehicle_33	D-R7920
Vehicle_34	D-R8357
Vehicle_35	D-S3726
Vehicle_36	D-S8053
Vehicle_37	D-T0951
Vehicle_38	D-T0971
Vehicle_39	D-T7031
Vehicle_4	D-R1137
Vehicle_40	D-T7051
Vehicle_41	D-T9651
Vehicle_5	D-R1273
Vehicle_6	D-R1530
Vehicle_7	D-R2192
Vehicle_8	D-R2311

VehicleId	jp_vehicleID
Vehicle_9	D-R2313

Vehicle Activity Distribution

```r
busgps_df <- busgps_df %>%
  mutate(
    coordinates = strsplit(as.character(Strlatlon), ","),
    latitude = as.numeric(sapply(coordinates, function(x) x[1])),
    longitude = as.numeric(sapply(coordinates, function(x) x[2])))

vehicle_activity <- busgps_df %>%
  count(VehicleId) %>%
  arrange(desc(n))

ggplot(vehicle_activity, aes(x = reorder(VehicleId, -n), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Vehicle Activity Distribution",
    subtitle = "Number of GPS records per vehicle",
    x = "Vehicle ID",
    y = "Number of GPS Records"
  )
```

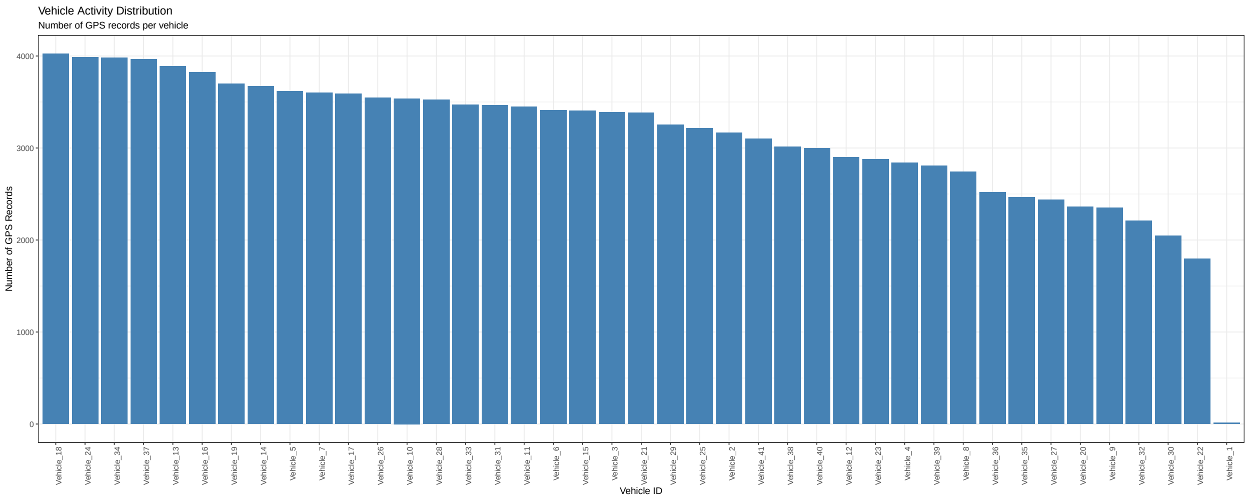


Figure 26 shows the distribution of GPS records across vehicles. Most vehicles show high activity levels (3,000-4,000 records), with a few showing significantly lower counts. This helps identify consistently active vehicles versus those that may have malfunctioning GPS devices or limited service periods.

Vehicle Activity by Hour of Day

```r
busgps_df <- busgps_df %>%
  mutate(hour_of_day = as.numeric(format(GPSDateTime, "%H")))

hourly_activity <- busgps_df %>%
  count(VehicleId, hour_of_day) %>%
  filter(VehicleId %in% head(vehicle_activity$VehicleId, 10))

ggplot(hourly_activity, aes(x = hour_of_day, y = VehicleId, fill = n)) +
  geom_tile() +
  scale_fill_viridis_c("Number of Records", option = "plasma") +
  theme_bw() +
  labs(
    title = "Vehicle Activity by Hour of Day",
    subtitle = "Shows when each vehicle is most active",
    x = "Hour of Day (24-hour format)",
    y = "Vehicle ID") +
  scale_x_continuous(breaks = 0:23)
```

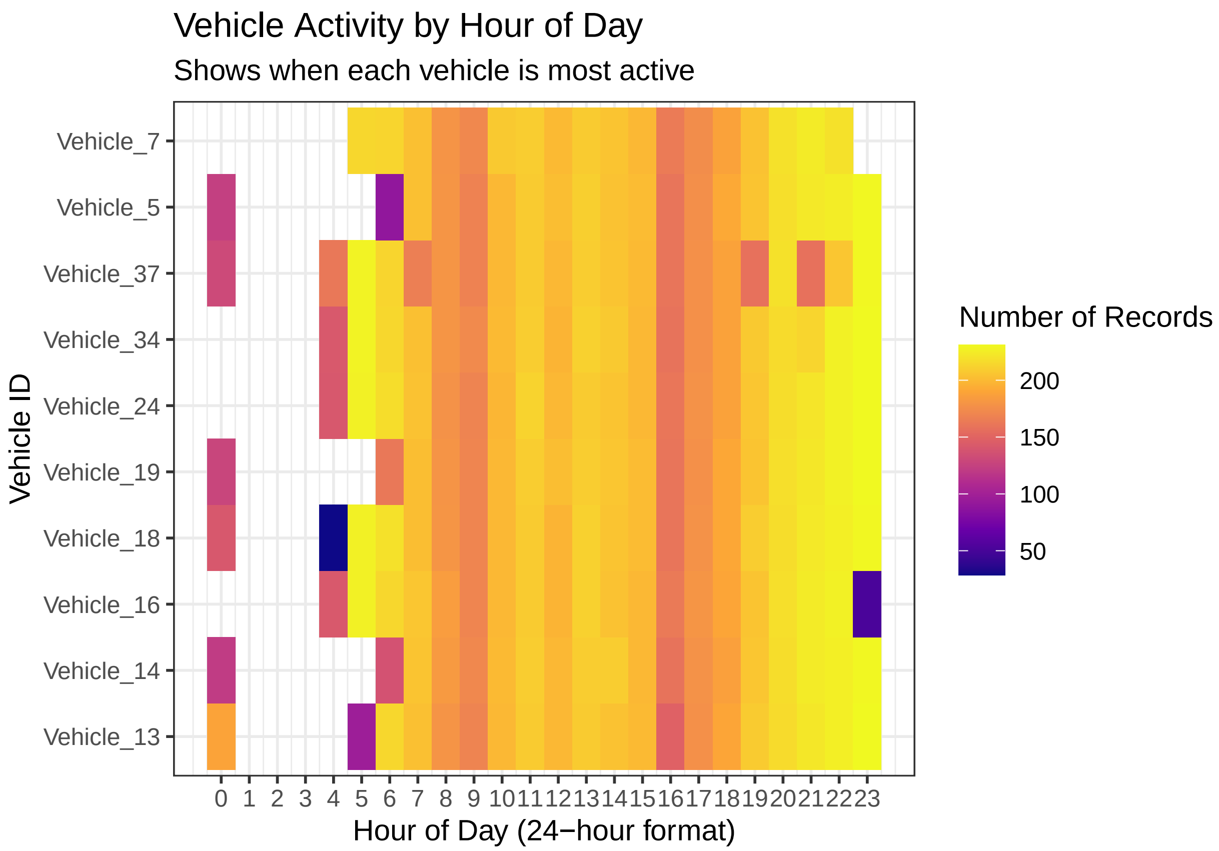


Figure 27 is a heatmap showing hourly activity for the ten most active vehicles. Peak activity occurs between 6-10 AM and during late afternoon hours. Most vehicles show minimal activity from midnight to 5 AM, matching the expected service patterns with morning and evening rush hours being the busiest periods.

Vehicle Speed Analysis

```
vehicle_speed <- busgps_df %>%
  arrange(VehicleId, GPSDateTime) %>%
  group_by(VehicleId) %>%
  mutate(prev_lat = lag(latitude),
         prev_lon = lag(longitude),
        prev_time = lag(GPSDateTime),

    dist_km = ifelse(!is.na(prev_lat),
                     sqrt((longitude - prev_lon)^2 + (latitude - prev_lat)^2) * 111.32,
                     NA),

    time_diff_hours = as.numeric(difftime(GPSDateTime, prev_time, units = "hours")),

    speed_kmh = ifelse(!is.na(time_diff_hours) & time_diff_hours > 0 & time_diff_hours < 0.1,
                       dist_km / time_diff_hours, NA)) %>%
  filter(!is.na(speed_kmh) & speed_kmh < 100)

ggplot(vehicle_speed, aes(x = reorder(VehicleId, speed_kmh, FUN = median), y = speed_kmh)) +
  geom_boxplot(fill = "turquoise") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Distribution of Vehicle Speeds",
    subtitle = "Comparing operating speeds across different vehicles",
    x = "Vehicle ID",
    y = "Speed (km/h)") +
  coord_cartesian(ylim = c(0, 60))
```

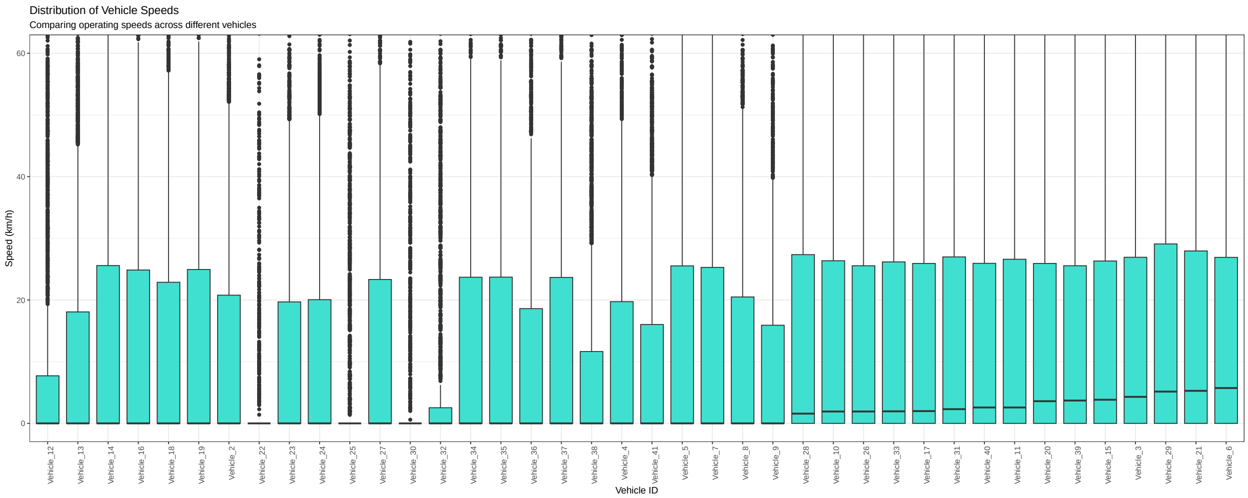


Figure 28 shows speed distributions for various vehicles. The report notes that some vehicles (like D-R2313 and D-R71745) have very low median speeds, while others (D-R1137, D-R2192, D-R6025) maintain higher, more consistent speeds. This helps identify potential mechanical issues or route-specific constraints affecting vehicle performance.

Direction Distribution by Vehicle

```
direction_by_vehicle <- busgps_df %>%
  count(VehicleId, ToDir) %>%
  group_by(VehicleId) %>%
  mutate(proportion = n / sum(n))

ggplot(direction_by_vehicle, aes(x = VehicleId, y = proportion, fill = factor(ToDir))) +
  geom_bar(stat = "identity") +
  scale_fill_brewer(palette = "Set1", name = "Direction") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Direction Distribution by Vehicle",
    subtitle = "Shows whether vehicles serve specific directions or both",
    x = "Vehicle ID",
    y = "Proportion of Records"
  )
```

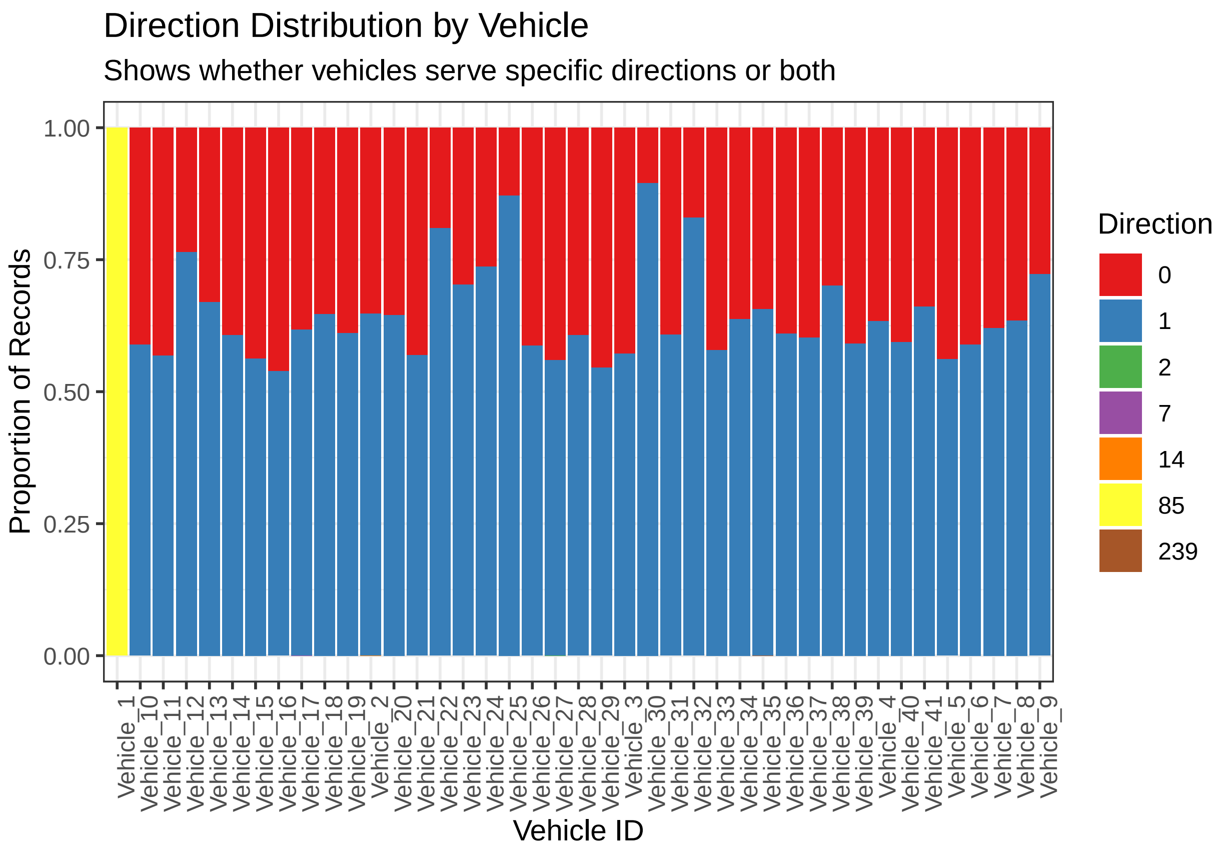


Figure 29 shows how vehicles are distributed across route directions. Most vehicles serve two main directions (0 and 1), likely representing inbound and outbound routes. Some vehicles are evenly split between directions, while others are predominantly assigned to one direction. The graph also reveals some anomalous direction codes that may indicate data issues.

Vehicle Dwell Time Analysis

```r
dwell_analysis <- vehicle_speed %>%
  mutate(
    is_stationary = speed_kmh < 3 & !is.na(speed_kmh),
    dwell_group = cumsum(c(1, diff(as.numeric(is_stationary)) != 0))
  ) %>%
  filter(is_stationary) %>%
  group_by(VehicleId, dwell_group) %>%
  summarize(
    dwell_time_min = as.numeric(difftime(max(GPSDateTime), min(GPSDateTime), units = "mins")),
    avg_latitude = mean(latitude),
    avg_longitude = mean(longitude),
    .groups = "drop"
  ) %>%
  filter(dwell_time_min > 1)

ggplot(dwell_analysis, aes(x = reorder(VehicleId, dwell_time_min, FUN = median), y = dwell_time_min)) +
  geom_boxplot(fill = "lightgreen") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Vehicle Dwell Time Analysis",
    subtitle = "Time spent stopped at locations (potential bus stops or traffic)",
    x = "Vehicle ID",
    y = "Dwell Time (minutes)"
  ) +
  coord_cartesian(ylim = c(0, 15))
```

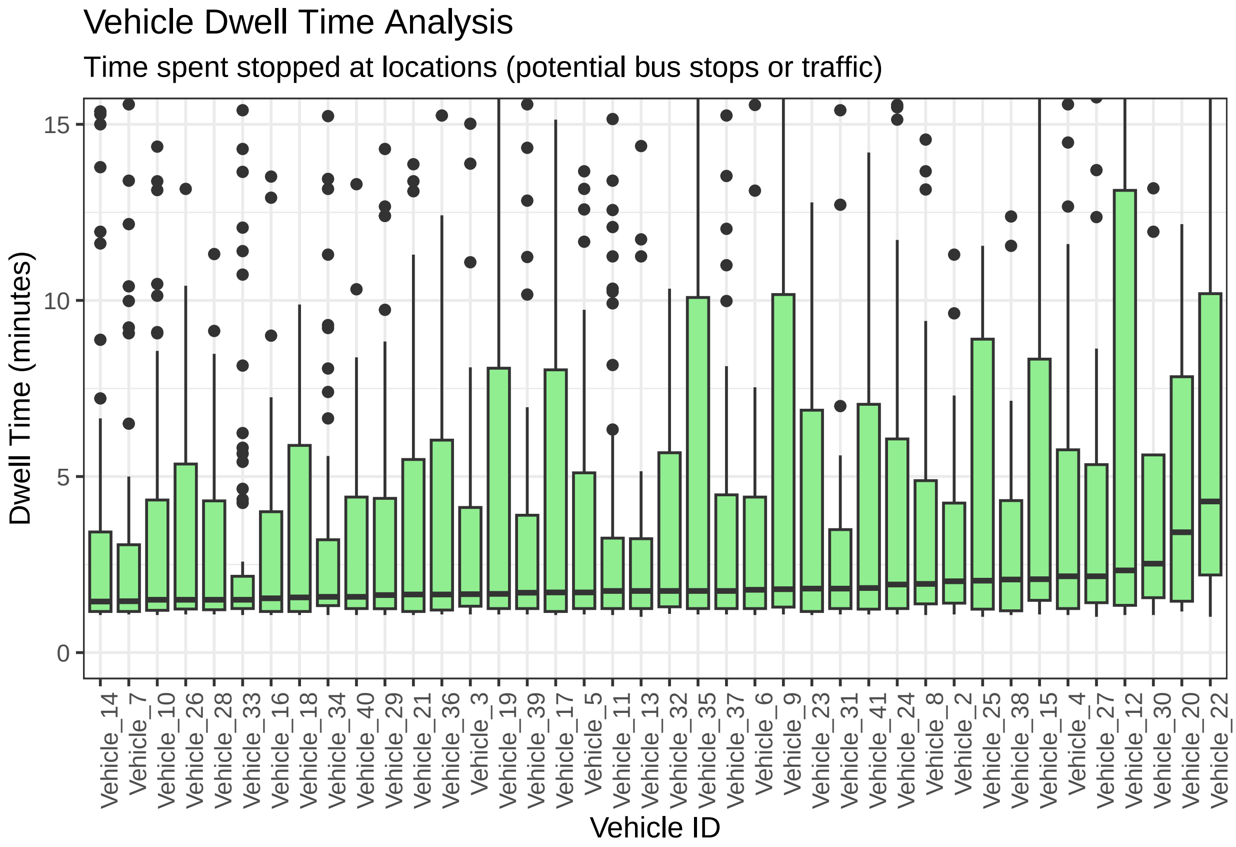


Figure 30 shows the dwell time distribution for each vehicle, revealing how long buses remain stationary at specific locations. The report indicates that longer dwell times may suggest high passenger activity or traffic congestion, while frequent extended stops could indicate high-demand areas that may need service adjustments.

Turn Analysis by Route

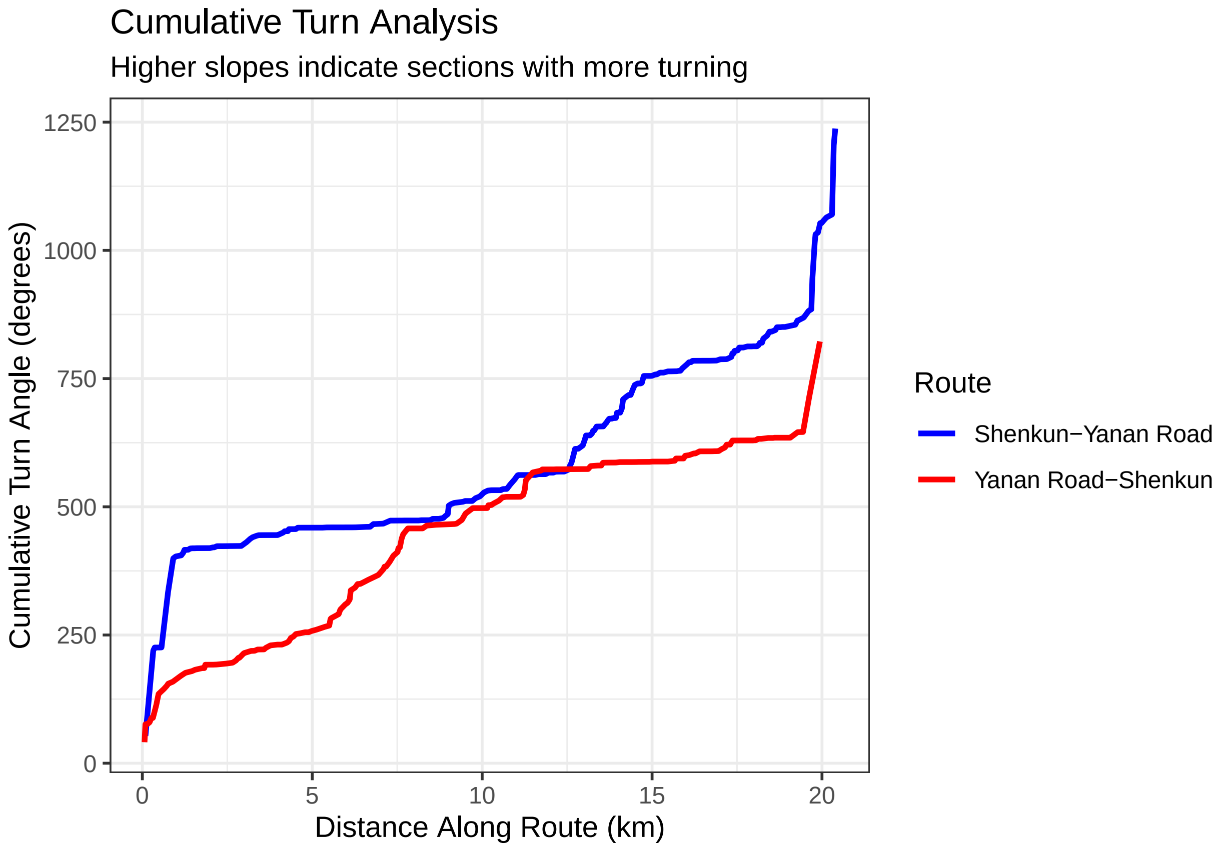
```
# Analyze the directional changes along the routes
# Calculate turn angles between segments
direction_changes <- all_segments %>%
  group_by(route) %>%
  mutate(
    prev_direction = lag(segment_direction),
    turn_angle = abs(((segment_direction - prev_direction + 180) %% 360) - 180),
    significant_turn = turn_angle > 25 # Define significant turns as >25 degrees
  ) %>%
  filter(!is.na(turn_angle))

# Count the number of significant turns on each route
turn_summary <- direction_changes %>%
  group_by(route) %>%
  summarize(
    total_turns = n(),
    significant_turns = sum(significant_turn),
    avg_turn_angle = mean(turn_angle),
    max_turn_angle = max(turn_angle)
  )

# Display the turn summary
knitr::kable(turn_summary, caption = "Turn Analysis by Route")
```

Table 7: Turn Analysis by Route

route	total_turns	significant_turns	avg_turn_angle	max_turn_angle
Shenkun-Yanan Road	232	8	5.334342	166.1686
Yanan Road-Shenkun	178	4	4.620225	108.7062

Cumulative Turn Analysis

```
# Visualize the cumulative turn angles along the routes
direction_changes %>%
  group_by(route) %>%
  mutate(cumulative_turns = cumsum(turn_angle)) %>%
  ggplot(aes(x = cumulative_distance, y = cumulative_turns, color = route)) +
  geom_line(size = 1) +
  scale_color_manual(values = c("blue", "red")) +
  theme_bw() +
  labs(title = "Cumulative Turn Analysis",
    subtitle = "Higher slopes indicate sections with more turning",
           x = "Distance Along Route (km)",
           y = "Cumulative Turn Angle (degrees)",
       color = "Route")
```



According to Figure 19(a) in the report, this analysis shows that the Shenkun-Yanan Road direction has a higher cumulative turn angle (exceeding 1200°) compared to the Yanan-Shenkun direction (under 850°). This suggests that the Shenkun-Yanan route has more frequent and sharper turns, potentially leading to slower travel times.

Travel Time Estimation Model

```
# Create a time estimation model based on segment lengths and estimated speeds
# First, let's make intelligent speed assignments based on segment characteristics

# Assign speeds based on segment properties (turns, density, etc.)
estimate_segment_speeds <- function(segments_df) {
  segments_df %>%
    group_by(route) %>%
    mutate(
      # Calculating turn sharpness from previous segment
      prev_direction = lag(segment_direction),
      turn_angle = if_else(is.na(prev_direction), 0,
                         abs(((segment_direction - prev_direction + 180) %% 360) - 180)),

      # Assigning speeds based on segment characteristics
      # 1. Slower speeds for sharp turns (>30 degrees)
      # 2. Average urban bus speed for normal segments
      # 3. Slightly higher speeds for straight segments
      segment_speed_kmh = case_when(
        turn_angle > 45 ~ 10,  # Very sharp turns: 10 km/h
        turn_angle > 30 ~ 15,  # Sharp turns: 15 km/h
        turn_angle > 15 ~ 20,  # Moderate turns: 20 km/h
        TRUE ~ 25              # Straight segments: 25 km/h
      ),

      # Calculate time for each segment (in minutes)
      segment_time_min = (segment_length / segment_speed_kmh) * 60,

      # Calculate cumulative time
      cumulative_time_min = cumsum(segment_time_min)
    ) %>%
    ungroup()
}

# Apply the speed model to both routes
route1_timed <- estimate_segment_speeds(route1_segments)
route2_timed <- estimate_segment_speeds(route2_segments)

# Combine the timed routes
all_timed_segments <- bind_rows(route1_timed, route2_timed)
```

Estimated Travel Time Visualization

```
# Create a time-distance plot for both routes
ggplot(all_timed_segments, aes(x = cumulative_distance, y = cumulative_time_min, color = route)) +
  geom_line(size = 1.2) +
  geom_point(data = all_timed_segments %>% filter(segment_id %% 20 == 0), size = 3, alpha = 0.7) +
  scale_color_manual(values = c("blue", "red")) +
  theme_bw() +
  labs(title = "Estimated Travel Time Along Bus Route 71",
    subtitle = "Based on segment lengths and estimated speeds",
           x = "Distance Along Route (km)",
           y = "Cumulative Travel Time (minutes)",
       color = "Route")
```

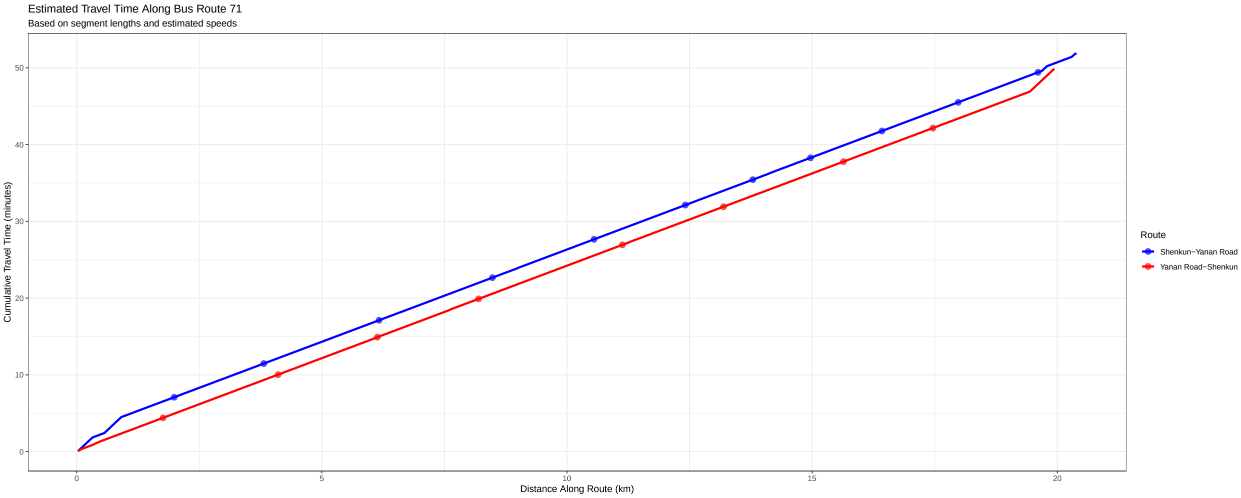
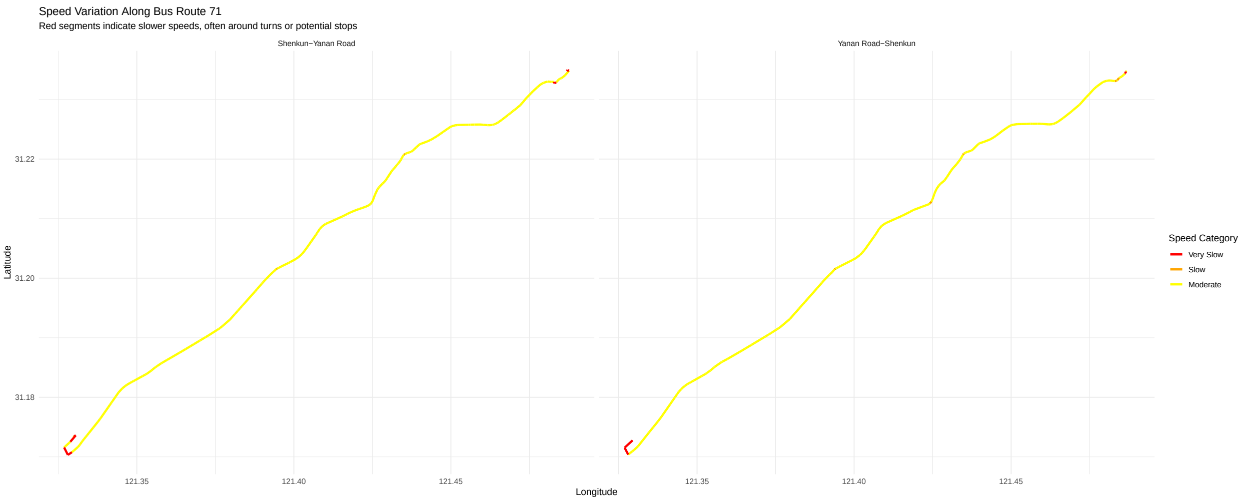


Figure 19(b) shows the estimated travel time along both directions. The report indicates that the Shenkun-Yanan Road direction has a longer estimated travel time (over 52 minutes) compared to the Yanan-Shenkun direction (around 51 minutes). While the absolute difference is small, it's consistent throughout the route, suggesting that route geometry and turns impact service time.

Speed Variation Map

```
slow_threshold <- 20  # km/h

# Create segment mapping dataset
segment_map_data <- all_timed_segments %>%
  dplyr::select(X, Y, next_X, next_Y, segment_speed_kmh, route) %>%
  mutate(speed_category = cut(segment_speed_kmh,
                              breaks = c(0, 15, 20, 25, Inf),
                              labels = c("Very Slow", "Slow", "Moderate", "Normal")))

# Create map showing speed variation along the routes
ggplot() +
  # Draw segments colored by speed
  geom_segment(data = segment_map_data,
               aes(x = X, y = Y, xend = next_X, yend = next_Y,
                   color = speed_category),
               size = 1.2) +
  # Color scale for speeds
  scale_color_manual(values = c("Very Slow" = "red", "Slow" = "orange",
                                "Moderate" = "yellow", "Normal" = "green")) +
  # Facet by route
  facet_wrap(~ route) +
  theme_minimal() +
  labs(
    title = "Speed Variation Along Bus Route 71",
    subtitle = "Red segments indicate slower speeds, often around turns or potential stops",
    x = "Longitude",
    y = "Latitude",
    color = "Speed Category"
  )
```



Key Points Analysis

```
#### sample_interval_km can be user input in shiny


# Distance Matrix Analysis
# Identify key points along the route (e.g., every 1km)
# and calculate distances between them

# Create a set of key points along each route
sample_interval_km <- 1 # Sample points approximately every 1km

# Function to extract key points at regular distance intervals
extract_key_points <- function(route_df, interval_km = 1) {
  # Calculate cumulative distance
  route_df$cum_dist <- c(0, cumsum(sqrt(diff(route_df$X)^2 + diff(route_df$Y)^2))) * 111.32

  # Define target distances
  max_dist <- max(route_df$cum_dist)
  target_dists <- seq(0, max_dist, by = interval_km)

  # For each target distance, find the closest point
  key_points <- data.frame(
    target_dist_km = target_dists,
    point_idx = numeric(length(target_dists)),
    stringsAsFactors = FALSE)

  for(i in 1:length(target_dists)) {
    key_points$point_idx[i] <- which.min(abs(route_df$cum_dist - target_dists[i]))
  }

  # Extract the key points
  result <- route_df[key_points$point_idx, ] %>%
    mutate(key_point_id = row_number(),
           key_point_label = paste0(round(cum_dist), " km"))

  return(result)
}

# Extract key points from both routes
route1_key_points <- extract_key_points(route1_df, sample_interval_km)
route2_key_points <- extract_key_points(route2_df, sample_interval_km)

# Visualize the key points on the map
ggplot() +
  # Base routes
  geom_path(data = route1_df, aes(x = X, y = Y), color = "blue", alpha = 0.5) +
  geom_path(data = route2_df, aes(x = X, y = Y), color = "darkred", alpha = 0.5) +
  # Key points
  geom_point(data = route1_key_points, aes(x = X, y = Y), color = "blue", size = 3) +
  geom_text(data = route1_key_points, aes(x = X, y = Y, label = key_point_label),
            color = "darkblue", vjust = -1.5, size = 3) +
  geom_point(data = route2_key_points, aes(x = X, y = Y), color = "darkred", size = 3) +
  geom_text(data = route2_key_points, aes(x = X, y = Y, label = key_point_label),
            color = "darkred", vjust = 1.5, size = 3) +
  theme_bw() +
  labs(title = "Key Points Along Bus Route 71", subtitle = paste("Points sampled at approximately", samp
    x = "Longitude",
    y = "Latitude")
```

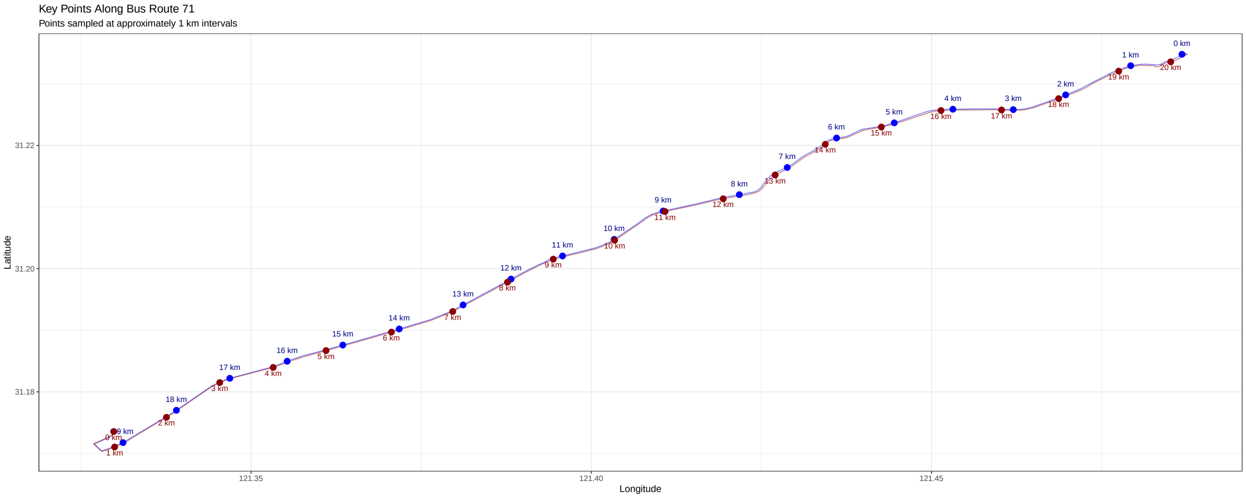


Figure 20 in the report shows key points along Bus Route 71 spaced at 1 km intervals. This visualization helps identify deviations, overlaps, and directional symmetry in the route, which is useful for transport planning and optimization.

Distance Matrix Calculation

```r
# Now, calculate the distance matrix
# Function to calculate distance matrix between points
calculate_distance_matrix <- function(points_df) {
  n <- nrow(points_df)
  dist_matrix <- matrix(0, nrow = n, ncol = n)

  for(i in 1:n) {
    for(j in 1:n) {
      if(i != j) {
        # Calculate Euclidean distance in km
        dist_matrix[i, j] <- sqrt(
          (points_df$X[i] - points_df$X[j])^2 +
          (points_df$Y[i] - points_df$Y[j])^2
        ) * 111.32
      }
    }
  }

  rownames(dist_matrix) <- points_df$key_point_label
  colnames(dist_matrix) <- points_df$key_point_label

  return(dist_matrix)
}

# Calculate distance matrices
route1_dist_matrix <- calculate_distance_matrix(route1_key_points)
route2_dist_matrix <- calculate_distance_matrix(route2_key_points)

# Convert distance matrix to long format for visualization
route1_dist_long <- as.data.frame(as.table(route1_dist_matrix)) %>%
  rename(from = Var1, to = Var2, distance_km = Freq)
```

Distance Matrix Visualization

```r
ggplot(route1_dist_long, aes(x = from, y = to, fill = distance_km)) +
  geom_tile() +
  scale_fill_viridis_c("Distance (km)", option = "plasma") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        axis.title = element_blank()) +
  labs(title = paste("Distance Matrix Between Key Points -", direction1_name_en),
    subtitle = "Colors indicate the distance (km) between any two points along the route")
```

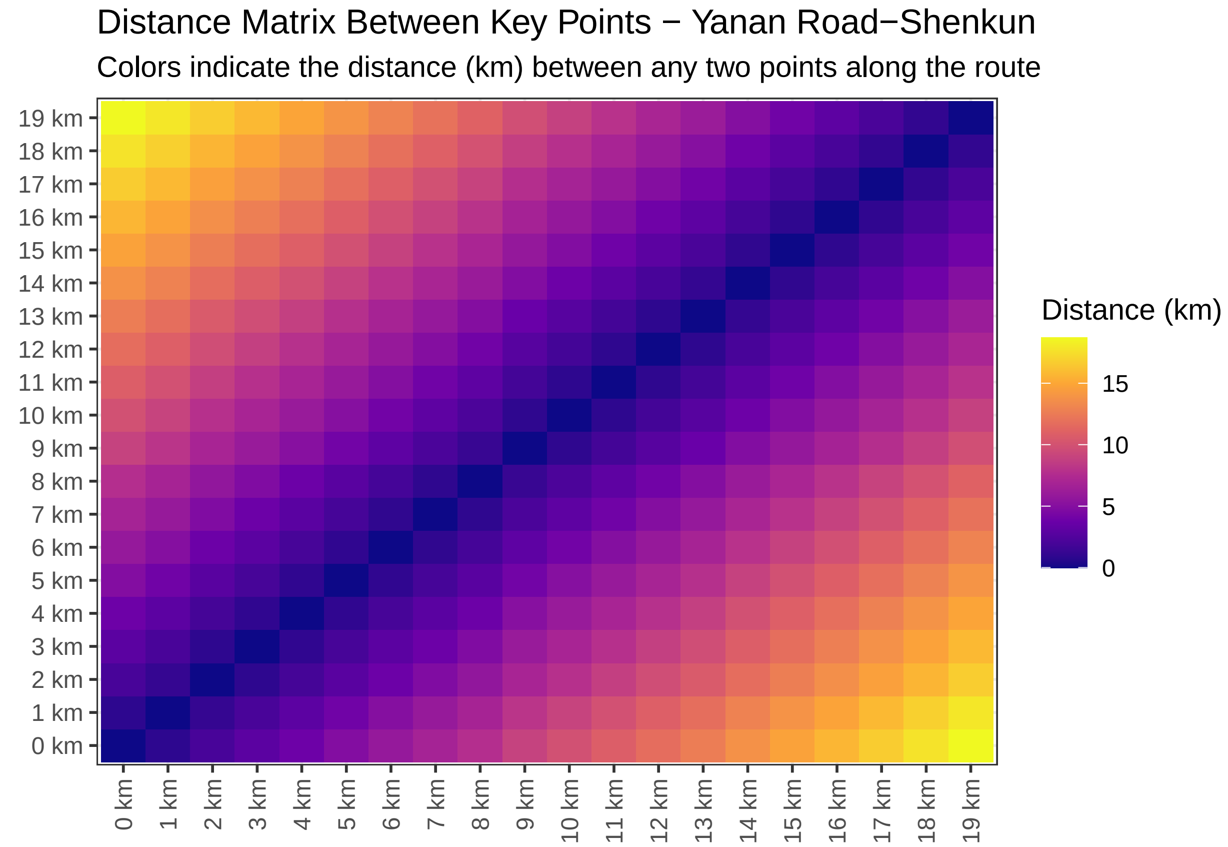


Figure 21 displays a distance matrix between key points along Bus Route 71. The matrix shows pairwise distances between points spaced every 1 km along the route. This visualization helps identify patterns of linearity and irregularity in the route, useful for understanding its spatial structure and finding opportunities for optimization.

Speed Analysis Along Route

```r
ggplot(all_timed_segments, aes(x = cumulative_distance, y = segment_speed_kmh, color = route)) +
  geom_line() +
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("blue", "red")) +
  theme_bw() +
  labs(
    title = "Estimated Speeds Along Bus Route 71",
    subtitle = "Showing how speed varies based on segment characteristics",
    x = "Distance Along Route (km)",
    y = "Speed (km/h)",
    color = "Route")
```

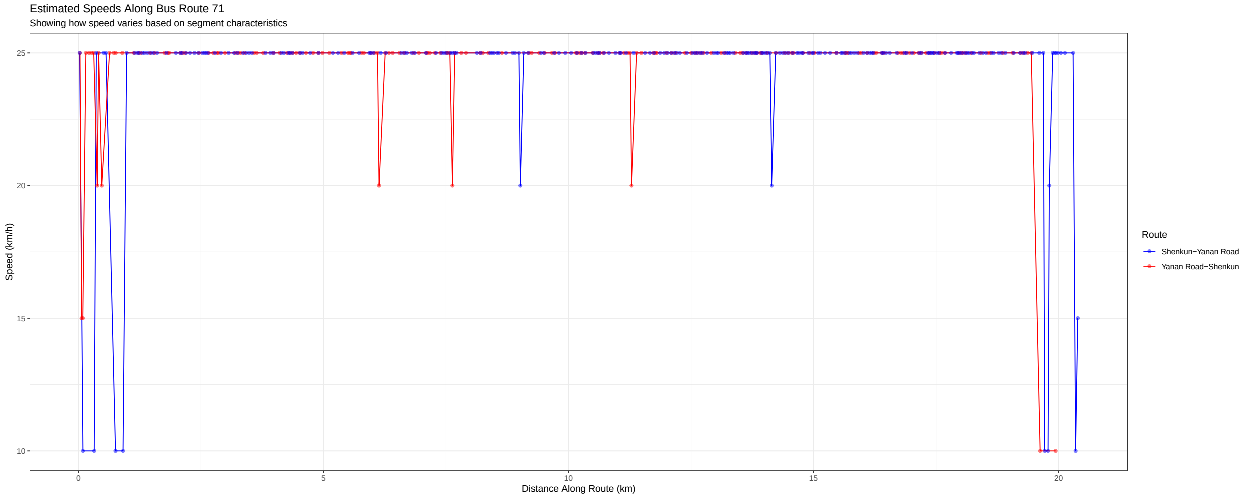


Figure 22 shows estimated speeds along both directions of Bus Route 71. Most segments operate near the maximum speed (25 km/h), but some drop to 10-15 km/h, particularly near the route ends. This suggests areas with signal delays, congestion, or complex intersections.

Travel Time Summary and Route with Time Markers

```r
# First, make sure all_timed_segments exists and has the necessary columns
if(!exists("all_timed_segments")) {
  # If not, recreate the route segments with time estimates
  route1_segments <- calculate_segment_properties(route1_df) %>%
    mutate(route = direction1_name_en,
           segment_id = row_number(),
           cumulative_distance = cumsum(segment_length))

  route2_segments <- calculate_segment_properties(route2_df) %>%
    mutate(route = direction2_name_en,
           segment_id = row_number(),
           cumulative_distance = cumsum(segment_length))

  # Combine segments
  all_segments <- bind_rows(route1_segments, route2_segments)

  # Assign speeds based on segment properties
  estimate_segment_speeds <- function(segments_df) {
    segments_df %>%
      group_by(route) %>%
      mutate(
        # Calculate turn sharpness from previous segment
        prev_direction = lag(segment_direction),
        turn_angle = if_else(is.na(prev_direction), 0,
                           abs(((segment_direction - prev_direction + 180) %% 360) - 180)),

        # Assign speeds based on segment characteristics
        segment_speed_kmh = case_when(
          turn_angle > 45 ~ 10,   # Very sharp turns: 10 km/h
          turn_angle > 30 ~ 15,  # Sharp turns: 15 km/h
          turn_angle > 15 ~ 20,  # Moderate turns: 20 km/h
          TRUE ~ 25              # Straight segments: 25 km/h
        ),
        # Calculate time for each segment (in minutes)
        segment_time_min = (segment_length / segment_speed_kmh) * 60,

        # Calculate cumulative time
        cumulative_time_min = cumsum(segment_time_min)) %>%
      ungroup()
    }

  # Apply the speed model to all segments
  all_timed_segments <- estimate_segment_speeds(all_segments)
}

# Calculate route summary statistics
route_time_summary <- all_timed_segments %>%
  group_by(route) %>%
  summarize(total_distance_km = max(cumulative_distance),
               total_time_min = max(cumulative_time_min),
                avg_speed_kmh = total_distance_km / (total_time_min / 60),
            min_segment_speed = min(segment_speed_kmh),
            max_segment_speed = max(segment_speed_kmh))

# Display the summary
knitr::kable(route_time_summary, caption = "Estimated Travel Time Summary", digits = 2)
```

Table 8: Estimated Travel Time Summary

route	total_distance_km	total_time_min	avg_speed_kmh	min_segment_speed	max_segment_speed
Shenkun-Yanan Road	20.39	51.91	23.57	10	25
Yanan Road-Shenkun	19.94	49.88	23.98	10	25

```
# Create time-distance markers every 5 minutes - rewritten without do()
# First create a function to find markers for one route
find_time_markers <- function(route_data) {
  target_times <- seq(5, max(route_data$cumulative_time_min), by = 5)

  # Initialize results dataframe
  result <- data.frame()

  for(target_time in target_times) {
    # Find the closest point to the target time
    idx <- which.min(abs(route_data$cumulative_time_min - target_time))

    # Add this point to our results
    marker <- route_data[idx, ] %>%
      mutate(time_marker = paste0(round(cumulative_time_min), " min"))
```

```r
    result <- bind_rows(result, marker)
  }

  return(result)
}

# Apply the function to each route
time_markers <- all_timed_segments %>%
  split(.$route) %>%
  lapply(find_time_markers) %>%
  bind_rows()

# Plot route map with time markers
ggplot() +
  # Base routes
  geom_path(data = route1_df, aes(x = X, y = Y), color = "blue", alpha = 0.5, size = 0.8) +
  geom_path(data = route2_df, aes(x = X, y = Y), color = "red", alpha = 0.5, size = 0.8) +
  # Time markers
  geom_point(data = time_markers, aes(x = X, y = Y, color = route), size = 3) +
  geom_text(data = time_markers, aes(x = X, y = Y, label = time_marker, color = route),
            vjust = -1.5, size = 3) +
  scale_color_manual(values = c("blue", "red")) +
  theme_minimal() +
  labs(
    title = "Bus Route 71 with Time Markers",
    subtitle = "Estimated travel time markers at 5-minute intervals",
    x = "Longitude",
    y = "Latitude",
    color = "Route"
  )
```

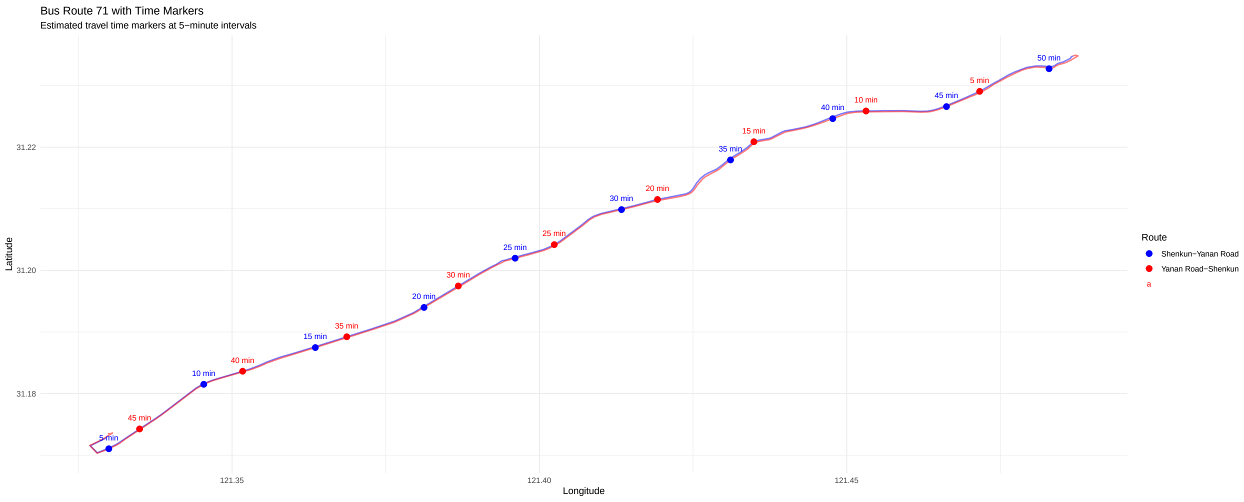


Figure 23 shows estimated travel times mapped along both directions of Route 71, with time markers at 5-minute intervals. The report notes that both directions follow similar paths, but time markers indicate non-uniform travel times across segments, highlighting directional inefficiencies that could be targets for optimization.

Segment Length Distribution by Route

```r
# 1. Boxplot comparing segment lengths between routes
ggplot(all_segments, aes(x = route, y = segment_length, fill = route)) +
  geom_boxplot() +
  scale_fill_manual(values = c("turquoise", "yellow")) +
  theme_bw() +
  labs(
    title = "Distribution of Segment Lengths by Route",
    subtitle = "Comparing segment length characteristics between routes",
    x = "Route",
    y = "Segment Length (km)",
    fill = "Route"
  ) +
  theme(legend.position = "none")
```

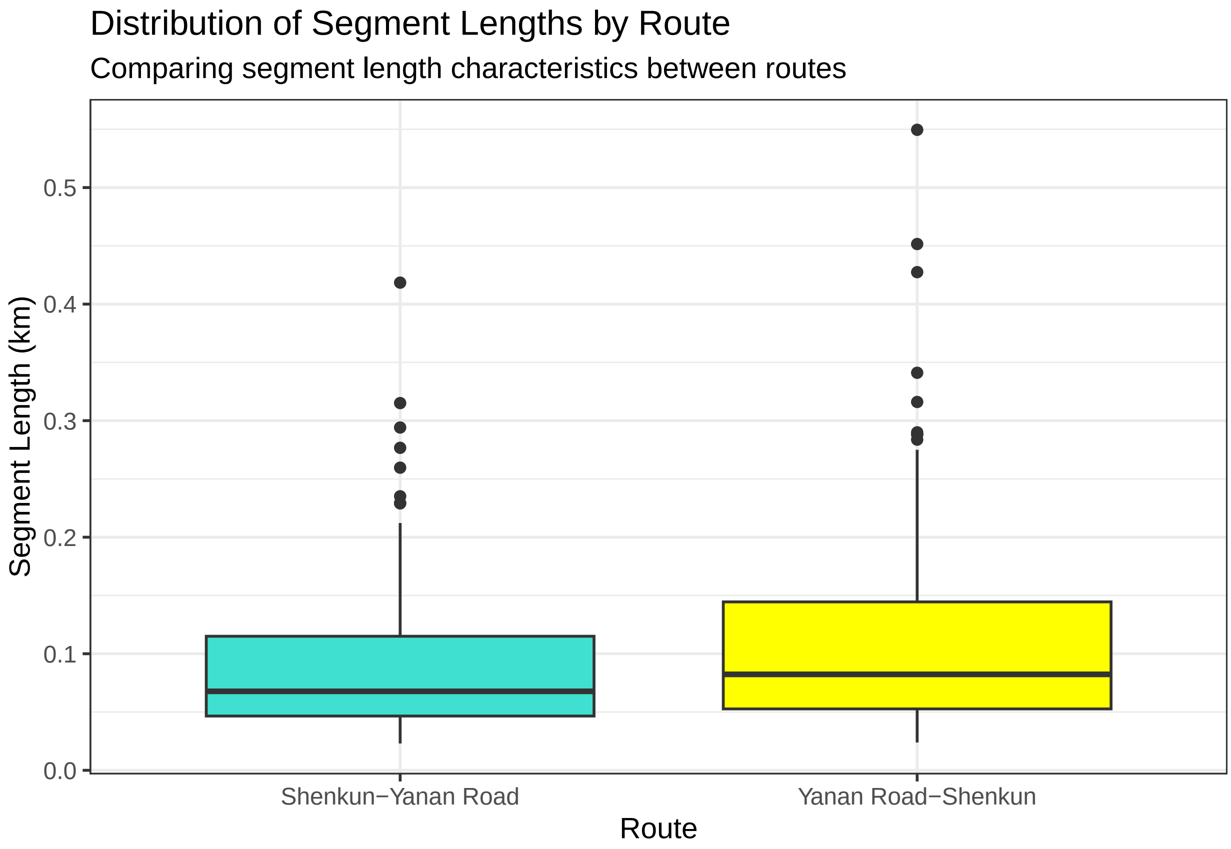


Figure 24 compares the distribution of segment lengths for both directions. The Shenkun-Yanan Road direction (turquoise) shows more uniform segment lengths with a lower median, while the Yanan-Shenkun Road direction (yellow) has greater variability and a higher median segment length. This asymmetry suggests differences in operational segmentation.

Speed Category Distribution by Route

```r
ggplot(segment_map_data, aes(x = route, fill = speed_category)) +
  geom_bar(position = "fill") +
  scale_fill_manual(values = c("Very Slow" = "red", "Slow" = "orange",
                               "Moderate" = "yellow", "Normal" = "green")) +
  theme_bw() +
  labs(title = "Proportion of Speed Categories by Route",
    subtitle = "Comparing the distribution of speed segments between routes",
           x = "Route",
           y = "Proportion",
        fill = "Speed Category") +
  coord_flip()
```

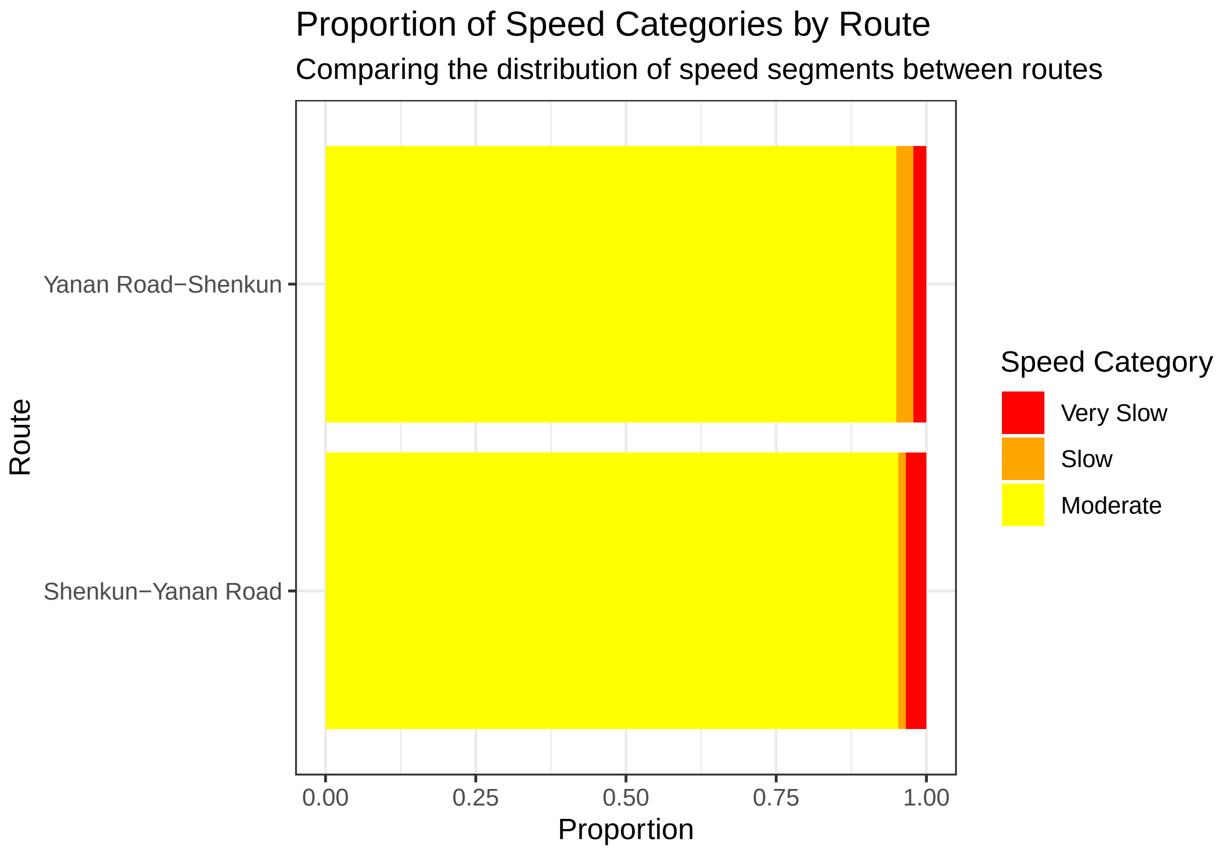


Figure 25 shows the proportion of speed categories along both directions. Moderate speeds (yellow) dominate both directions, but the Yanan-Shenkun direction has slightly more segments in the Slow and Very Slow categories, suggesting occasional congestion or operational issues like passenger boarding delays.

Vehicle ID Mapping

```r
# Creating a copy of the original VehicleId and creating simplified labels
busgps_df <- busgps_df %>%
  # First we store the original IDs
  mutate(jp_vehicleID = VehicleId) %>%
  group_by(VehicleId) %>% # A mapping of unique VehicleIDs to simplified labels
  mutate(temp_id = cur_group_id()) %>%
  ungroup() %>%
  mutate(VehicleId = paste0("Vehicle_", temp_id)) %>% # Simplified VehicleId labels
  # Removing the temporary ID column without using select()
  subset(select = -temp_id)


vehicle_id_cols <- c("VehicleId", "jp_vehicleID")
vehicle_id_mapping <- unique(busgps_df[, vehicle_id_cols])
vehicle_id_mapping <- vehicle_id_mapping[order(vehicle_id_mapping$VehicleId), ]

knitr::kable(vehicle_id_mapping, caption = "Vehicle ID Mapping Reference")
```

Table 9: Vehicle ID Mapping Reference

VehicleId	jp_vehicleID
Vehicle_1	Vehicle_1
Vehicle_10	Vehicle_18
Vehicle_11	Vehicle_19
Vehicle_12	Vehicle_2
Vehicle_13	Vehicle_20
Vehicle_14	Vehicle_21
Vehicle_15	Vehicle_22
Vehicle_16	Vehicle_23
Vehicle_17	Vehicle_24
Vehicle_18	Vehicle_25
Vehicle_19	Vehicle_26
Vehicle_2	Vehicle_10
Vehicle_20	Vehicle_27
Vehicle_21	Vehicle_28
Vehicle_22	Vehicle_29
Vehicle_23	Vehicle_3
Vehicle_24	Vehicle_30
Vehicle_25	Vehicle_31
Vehicle_26	Vehicle_32
Vehicle_27	Vehicle_33
Vehicle_28	Vehicle_34
Vehicle_29	Vehicle_35
Vehicle_3	Vehicle_11
Vehicle_30	Vehicle_36
Vehicle_31	Vehicle_37
Vehicle_32	Vehicle_38
Vehicle_33	Vehicle_39
Vehicle_34	Vehicle_4
Vehicle_35	Vehicle_40
Vehicle_36	Vehicle_41
Vehicle_37	Vehicle_5

VehicleId	jp_vehicleID
Vehicle_38	Vehicle_6
Vehicle_39	Vehicle_7
Vehicle_4	Vehicle_12
Vehicle_40	Vehicle_8
Vehicle_41	Vehicle_9
Vehicle_5	Vehicle_13
Vehicle_6	Vehicle_14
Vehicle_7	Vehicle_15
Vehicle_8	Vehicle_16
Vehicle_9	Vehicle_17

Vehicle Activity Distribution

```r
# First, let's extract coordinates from Strlatlon
busgps_df <- busgps_df %>%
  mutate(
    # Extract coordinates from Strlatlon column (assuming format "lat,lon")
    coordinates = strsplit(as.character(Strlatlon), ","),
    latitude = as.numeric(sapply(coordinates, function(x) x[1])),
    longitude = as.numeric(sapply(coordinates, function(x) x[2])))

# 1. Vehicle activity distribution - number of GPS points per vehicle
vehicle_activity <- busgps_df %>%
  count(VehicleId) %>%
  arrange(desc(n))

# Plot the distribution
ggplot(vehicle_activity, aes(x = reorder(VehicleId, -n), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Vehicle Activity Distribution",
    subtitle = "Number of GPS records per vehicle",
    x = "Vehicle ID",
    y = "Number of GPS Records"
  )
```

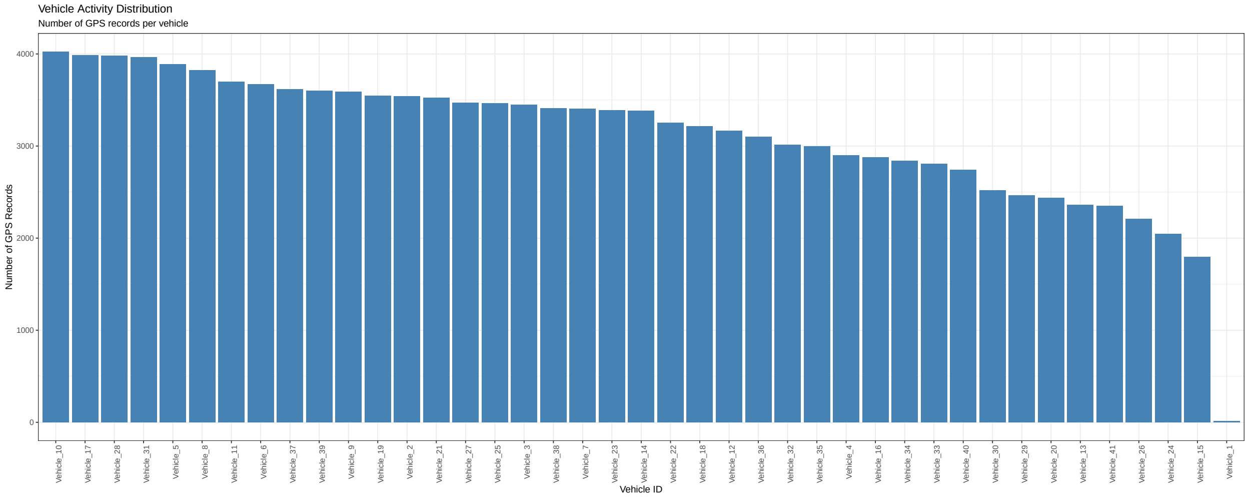


Figure 26 shows the distribution of GPS records across vehicles. Most vehicles show high activity levels (3,000-4,000 records), with a few showing significantly lower counts. This helps identify consistently active vehicles versus those that may have malfunctioning GPS devices or limited service periods.

Vehicle Activity by Hour of Day

```
# Extract hour from timestamp
busgps_df <- busgps_df %>%
  mutate(hour_of_day = as.numeric(format(GPSDateTime, "%H")))

# Analyze vehicle activity by hour of day
hourly_activity <- busgps_df %>%
  count(VehicleId, hour_of_day) %>%
  # Get the top 10 most active vehicles
  filter(VehicleId %in% head(vehicle_activity$VehicleId, 10))

# Create a heatmap of vehicle activity by hour
ggplot(hourly_activity, aes(x = hour_of_day, y = VehicleId, fill = n)) +
  geom_tile() +
  scale_fill_viridis_c("Number of Records", option = "plasma") +
  theme_bw() +
  labs(
    title = "Vehicle Activity by Hour of Day",
    subtitle = "Shows when each vehicle is most active",
    x = "Hour of Day (24-hour format)",
    y = "Vehicle ID") +
  scale_x_continuous(breaks = 0:23)
```

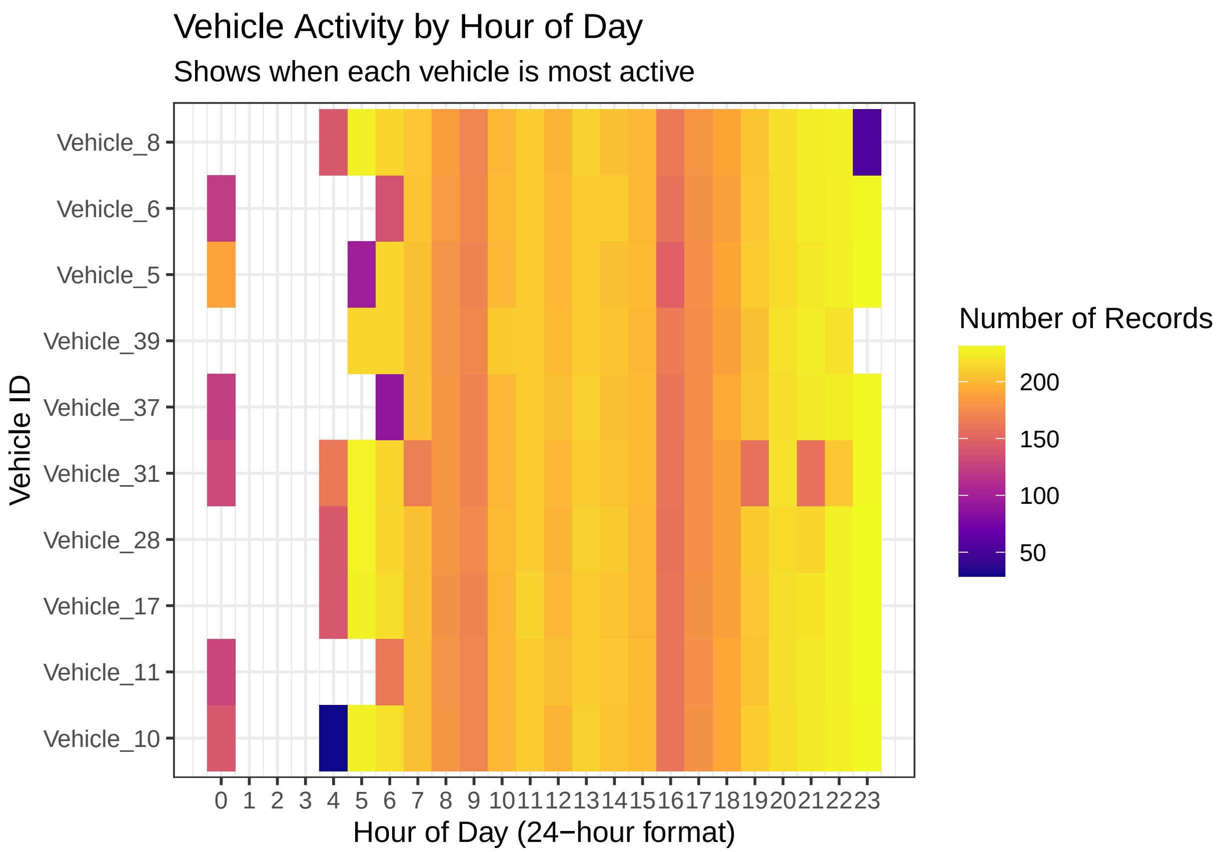


Figure 27 is a heatmap showing hourly activity for the ten most active vehicles. Peak activity occurs between 6-10 AM and during late afternoon hours. Most vehicles show minimal activity from midnight to 5 AM, matching the expected service patterns with morning and evening rush hours being the busiest periods.

Vehicle Speed Analysis

```
# Calculate speed between consecutive points for each vehicle
vehicle_speed <- busgps_df %>%
  arrange(VehicleId, GPSDateTime) %>%
  group_by(VehicleId) %>%
  mutate(prev_lat = lag(latitude),
         prev_lon = lag(longitude),
        prev_time = lag(GPSDateTime),

    # Calculate distance in km (approximate using Haversine)
    dist_km = ifelse(!is.na(prev_lat),
                    sqrt((longitude - prev_lon)^2 + (latitude - prev_lat)^2) * 111.32,
                    NA),

    # Calculate time difference in hours
    time_diff_hours = as.numeric(difftime(GPSDateTime, prev_time, units = "hours")),

    # Calculate speed in km/h
    speed_kmh = ifelse(!is.na(time_diff_hours) & time_diff_hours > 0 & time_diff_hours < 0.1,
                       dist_km / time_diff_hours, NA)) %>%
  filter(!is.na(speed_kmh) & speed_kmh < 100) # Filter out unrealistic speeds

# Boxplot of speeds by vehicle
ggplot(vehicle_speed, aes(x = reorder(VehicleId, speed_kmh, FUN = median), y = speed_kmh)) +
  geom_boxplot(fill = "turquoise") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Distribution of Vehicle Speeds",
    subtitle = "Comparing operating speeds across different vehicles",
    x = "Vehicle ID",
    y = "Speed (km/h)") +
  coord_cartesian(ylim = c(0, 60))
```

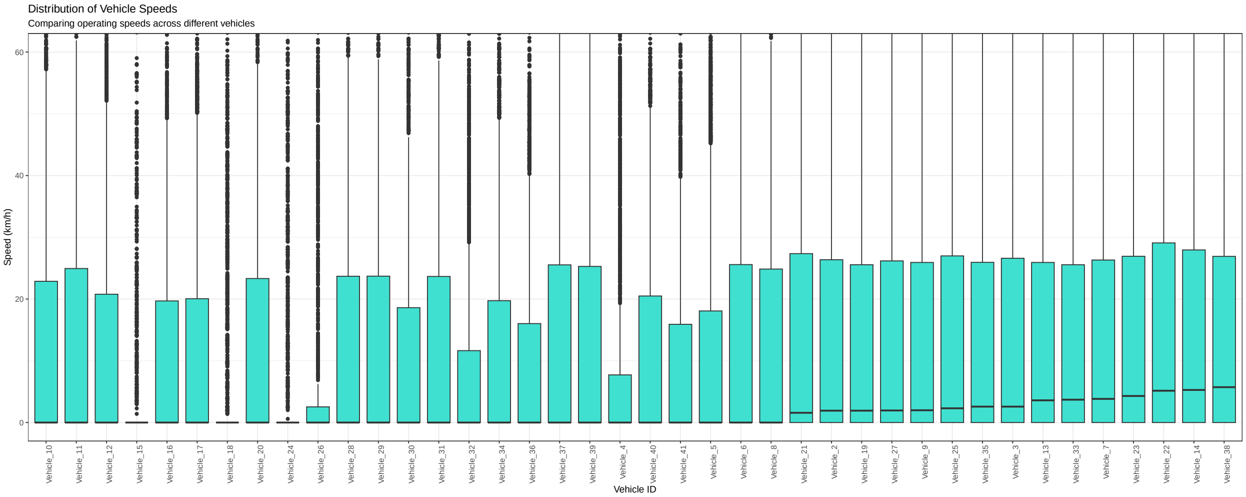


Figure 28 shows speed distributions for various vehicles. The report notes that some vehicles (like D-R2313 and D-R71745) have very low median speeds, while others (D-R1137, D-R2192, D-R6025) maintain higher, more consistent speeds. This helps identify potential mechanical issues or route-specific constraints affecting vehicle performance.

Direction Distribution by Vehicle

```
# Analyze which direction each vehicle typically serves
direction_by_vehicle <- busgps_df %>%
  count(VehicleId, ToDir) %>%
  group_by(VehicleId) %>%
  mutate(proportion = n / sum(n))

# Plot the proportion of directions by vehicle
ggplot(direction_by_vehicle, aes(x = VehicleId, y = proportion, fill = factor(ToDir))) +
  geom_bar(stat = "identity") +
  scale_fill_brewer(palette = "Set1", name = "Direction") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Direction Distribution by Vehicle",
    subtitle = "Shows whether vehicles serve specific directions or both",
    x = "Vehicle ID",
    y = "Proportion of Records"
  )
```

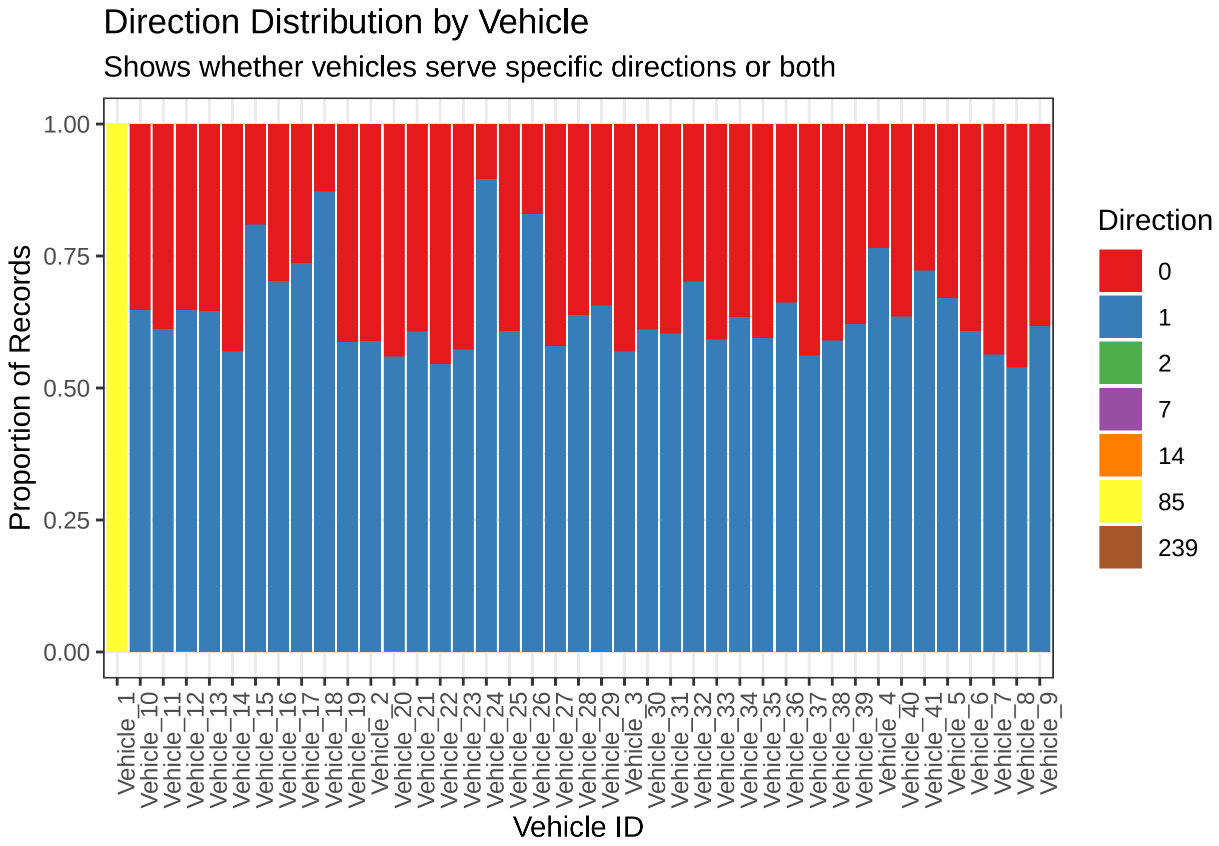


Figure 29 shows how vehicles are distributed across route directions. Most vehicles serve two main directions (0 and 1), likely representing inbound and outbound routes. Some vehicles are evenly split between directions, while others are predominantly assigned to one direction. The graph also reveals some anomalous direction codes that may indicate data issues.

Vehicle Dwell Time Analysis

```
# Calculate dwell time (time spent at one location) by looking for consecutive GPS points
# that are very close to each other
dwell_analysis <- vehicle_speed %>%
  mutate(
    is_stationary = speed_kmh < 3 & !is.na(speed_kmh),
    dwell_group = cumsum(c(1, diff(as.numeric(is_stationary)) != 0))
  ) %>%
  filter(is_stationary) %>%
  group_by(VehicleId, dwell_group) %>%
  summarize(
    dwell_time_min = as.numeric(difftime(max(GPSDateTime), min(GPSDateTime), units = "mins")),
    avg_latitude = mean(latitude),
    avg_longitude = mean(longitude),
    .groups = "drop"
  ) %>%
  filter(dwell_time_min > 1) # Focus on stops longer than 1 minute

# Boxplot of dwell times by vehicle
ggplot(dwell_analysis, aes(x = reorder(VehicleId, dwell_time_min, FUN = median), y = dwell_time_min)) +
  geom_boxplot(fill = "lightgreen") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Vehicle Dwell Time Analysis",
    subtitle = "Time spent stopped at locations (potential bus stops or traffic)",
    x = "Vehicle ID",
    y = "Dwell Time (minutes)"
  ) +
  coord_cartesian(ylim = c(0, 15))
```

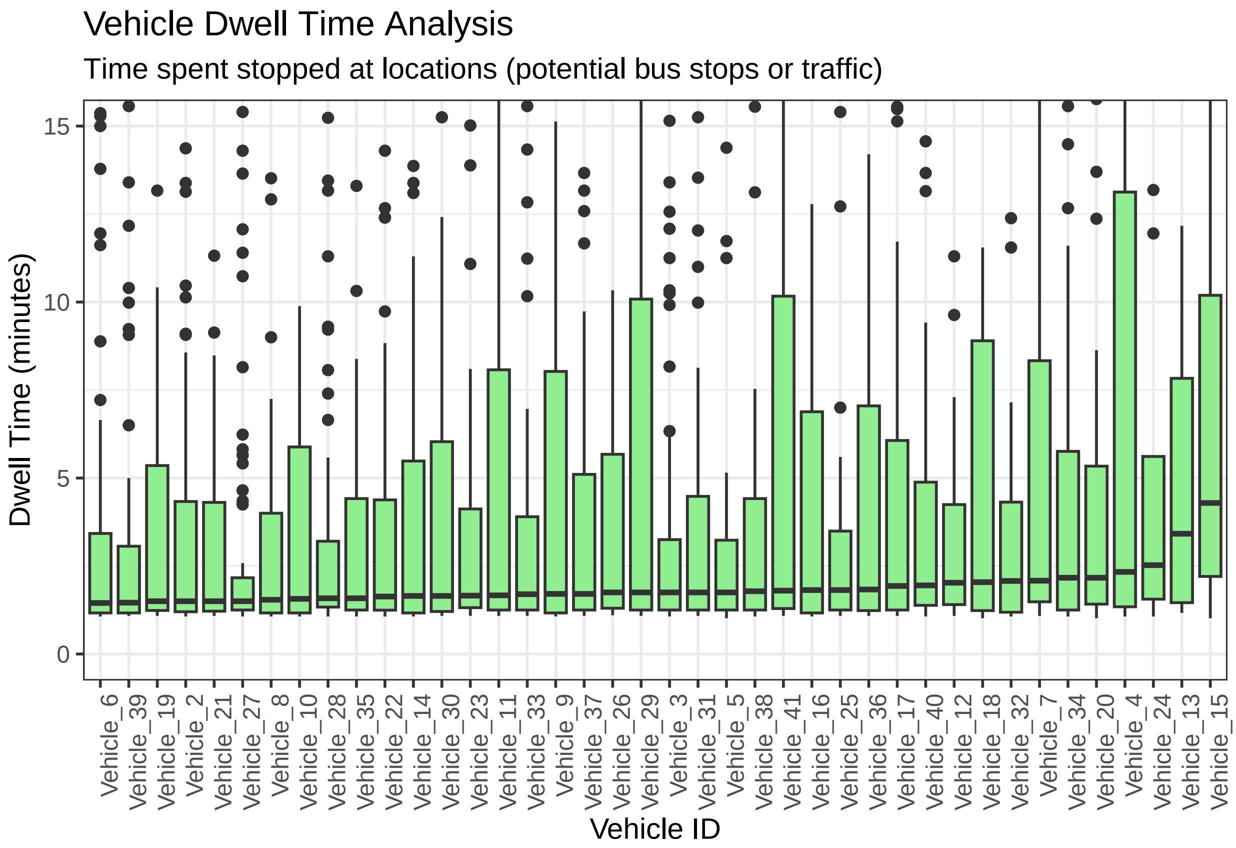


Figure 30 shows the dwell time distribution for each vehicle, revealing how long buses remain stationary at specific locations. The report indicates that longer dwell times may suggest high passenger activity or traffic congestion, while frequent extended stops could indicate high-demand areas that may need service adjustments.