![ipea Instituto de Pesquisa Econômica Aplicada]

## 9 ACCESSIBILITY ESTIMATES

Finally, the {aopdata} package also allows one to download estimates of accessibility to jobs, public health facilities, public schools and social assistance services. These estimates were calculated using 2017, 2018 and 2019 as reference years.

This data can be downloaded with the `read_access()` function, which works similarly to `read_population()` and `read_landuse()`. Besides indicating the city (`city` parameter) and the reference year (`year`), though, it is also necessary to inform the transport mode (`mode`) and the interval of the day (peak, between 6 am and 8 am, or off-peak, between 2 pm and 4 pm, controlled by `peak`) which identify the accessibility data that should be downloaded.

With the code below, we show how to download accessibility estimates that refer to the peak period in São Paulo in 2019. In this example, we downloaded accessibility estimates both by car and by public transport and merged them into a single `data.frame`. Please note that this function results in a table that also includes sociodemographic and land use data.

```
access_pt <- aopdata::read_access(
  city = "São Paulo",
  mode = "public_transport",
  year = 2019,
  peak = TRUE,
  geometry = TRUE,
  showProgress = FALSE
)

access_car <- aopdata::read_access(
  city = "São Paulo",
  mode = "car",
  year = 2019,
  peak = TRUE,
  geometry = TRUE,
  showProgress = FALSE
)

data_sp <-rbind(access_pt, access_car)
```

```
names(data_sp)
```

```
  [1] "id_hex"          "abbrev_muni"    "name_muni"      "code_muni"      "year"
  [6] "P001"            "P002"           "P003"           "P004"           "P005"
 [11] "P006"            "P007"           "P010"           "P011"           "P012"
 [16] "P013"            "P014"           "P015"           "P016"           "R001"
 [21] "R002"            "R003"           "T001"           "T002"           "T003"
 [26] "T004"            "E001"           "E002"           "E003"           "E004"
 [31] "M001"            "M002"           "M003"           "M004"           "S001"
 [36] "S002"            "S003"           "S004"           "C001"           "mode"
 [41] "peak"            "CMATT15"        "CMATB15"        "CMATM15"        "CMATA15"
 [46] "CMAST15"         "CMASB15"        "CMASM15"        "CMASA15"        "CMAET15"
 [51] "CMAEI15"         "CMAEF15"        "CMAEM15"        "CMAMT15"        "CMAMI15"
 [56] "CMAMF15"         "CMAMM15"        "CMACT15"        "CMPPT15"        "CMPPH15"
 [61] "CMPPM15"         "CMPPB15"        "CMPPA15"        "CMPPI15"        "CMPPN15"
 [66] "CMPP0005I15"     "CMPP0614I15"    "CMPP1518I15"    "CMPP1924I15"    "CMPP2539I15"
 [71] "CMPP4069I15"     "CMPP70I15"      "CMATT30"        "CMATB30"        "CMATM30"
 [76] "CMATA30"         "CMAST30"        "CMASB30"        "CMASM30"        "CMASA30"
 [81] "CMAET30"         "CMAEI30"        "CMAEF30"        "CMAEM30"        "CMAMT30"
 [86] "CMAMI30"         "CMAMF30"        "CMAMM30"        "CMACT30"        "CMPPT30"
 [91] "CMPPH30"         "CMPPM30"        "CMPPB30"        "CMPPA30"        "CMPPI30"
 [96] "CMPPN30"         "CMPP0005I30"    "CMPP0614I30"    "CMPP1518I30"    "CMPP1924I30"
[101] "CMPP2539I30"     "CMPP4069I30"    "CMPP70I30"      "CMATT60"        "CMATB60"
[106] "CMATM60"         "CMATA60"        "CMAST60"        "CMASB60"        "CMASM60"
[111] "CMASA60"         "CMAET60"        "CMAEI60"        "CMAEF60"        "CMAEM60"
[116] "CMAMT60"         "CMAMI60"        "CMAMF60"        "CMAMM60"        "CMACT60"
[121] "CMPPT60"         "CMPPH60"        "CMPPM60"        "CMPPB60"        "CMPPA60"
[126] "CMPPI60"         "CMPPN60"        "CMPP0005I60"    "CMPP0614I60"    "CMPP1518I60"
[131] "CMPP1924I60"     "CMPP2539I60"    "CMPP4069I60"    "CMPP70I60"      "CMATT90"
[136] "CMATB90"         "CMATM90"        "CMATA90"        "CMAST90"        "CMASB90"
[141] "CMASM90"         "CMASA90"        "CMAET90"        "CMAEI90"        "CMAEF90"
[146] "CMAEM90"         "CMAMT90"        "CMAMI90"        "CMAMF90"        "CMAMM90"
[151] "CMACT90"         "CMPPT90"        "CMPPH90"        "CMPPM90"        "CMPPB90"
[156] "CMPPA90"         "CMPPI90"        "CMPPN90"        "CMPP0005I90"    "CMPP0614I90"
[161] "CMPP1518I90"     "CMPP1924I90"    "CMPP2539I90"    "CMPP4069I90"    "CMPP70I90"
[166] "CMATT120"        "CMATB120"       "CMATM120"       "CMATA120"       "CMAST120"
[171] "CMASB120"        "CMASM120"       "CMASA120"       "CMAET120"       "CMAEI120"
[176] "CMAEF120"        "CMAEM120"       "CMAMT120"       "CMAMI120"       "CMAMF120"
[181] "CMAMM120"        "CMACT120"       "CMPPT120"       "CMPPH120"       "CMPPM120"
[186] "CMPPB120"        "CMPPA120"       "CMPPI120"       "CMPPN120"       "CMPP0005I120"
[191] "CMPP0614I120"    "CMPP1518I120"   "CMPP1924I120"   "CMPP2539I120"   "CMPP4069I120"
[196] "CMPP70I120"      "TMIST"          "TMISB"          "TMISM"          "TMISA"
[201] "TMIET"           "TMIEI"          "TMIEF"          "TMIEM"          "TMICT"
[206] "geometry"
```

The names of the accessibility estimates columns, such as `CMAEF30`, `TMISB` and `CMPPM60`, result from a combination of three components, as follows.

1) The type of accessibility measure, which is indicated by the first 3 letters of the code. The data includes three types of measures:

a) `CMA` – active cumulative accessibility;

b) `CMP` – passive cumulative accessibility; and

c) `TMI` – minimum travel time to the nearest opportunity.

2) The type of activity to which the accessibility levels were calculated, indicated by the following two letters, in the middle of the column name. The data includes accessibility estimates to various types of activities:

a) `TT` – all jobs;

b) `TB` – low education jobs;

c) `TM` – middle education jobs;

d) `TA` – high education jobs;

e) `ST` – all public health facilities;

f) `SB` – low complexity public health facilities;

g) `SM` – medium complexity public health facilities;

h) `SA` – high complexity public health facilities;

i) `ET` – all public schools;

j) `EI` – early childhood public schools;

k) `EF` – primary public schools;

l) `MS` – secondary public schools;

m) `MT` – total number of enrollments in public schools;

n) `MI` – number of enrollments in early childhood public schools;

o) `MF` – number of enrollments in primary public schools;

p) `MM` – number of enrollments in secondary public schools; and

q) `CT` – all CRAS.

In the case of the passive cumulative measure, the letters in the middle of the column name indicate the population group which the accessibility estimates refer to:

a) PT – the entire population;

b) PH – male population;

c) PM – female population;

d) PB – white population;

e) PN – black population;

f) PA – yellow population;

g) PI – indigenous population;

h) P0005I – population from 0 to 5 years old;

i) P0614I – population from 6 to 14 years old;

j) P1518I – population from 15 to 18 years old;

k) P1924I – population from 19 to 24 years old;

l) P2539I – population from 25 to 39 years old;

m) P4069I – population from 40 to 69 years old; and

n) P70I – population aged 70 years old and over.

3) The travel time threshold used to estimate the accessibility levels, which is indicated by the two numbers at the end of the column name. This component only applies to the active and passive cumulative measure. The data includes accessibility estimates calculated with cutoffs of 15, 30, 45, 60, 90 and 120 minutes, depending on the transport mode.

Examples:

• CMAEF30: number of accessible primary public schools within 30 minutes of travel;

• TMISB: minimum travel time to the closest low complexity public health facility; and

• CMPPM60: number of women that can access a certain grid cell within 60 minutes of travel.

The full description of the columns can also be found in the function documentation, running the ?read_access command in R. The following sections show examples illustrating how to create spatial visualizations and charts out of the accessibility dataset.

### 9.1 Map of travel time to access the nearest hospital

In this example, we compare the access time from each grid cell to the nearest public hospital by car and by public transport. To analyze the minimum travel time (TMI) to high complexity public hospitals (SA), we use the TMISA column. With the code below, we load the data visualization libraries and configure the maps showing the spatial distribution of access time by both transport modes. Because public transport trips are usually much longer than car trips, we truncate the travel time distribution to 60 minutes.

```r
library(ggplot2)
library(patchwork)

# truncates travel times to 60 minutes
data_sp$TMISA <- ifelse(data_sp$TMISA > 60, 60, data_sp$TMISA)

ggplot(subset(data_sp, !is.na(mode))) +
  geom_sf(aes(fill = TMISA), color = NA, alpha = 0.9) +
  scale_fill_viridis_c(
    option = "cividis",
    direction = -1,
    breaks = seq(0, 60, 10),
    labels = c(seq(0, 50, 10), "60+")
  ) +
  labs(fill = "Time\n(minutes)") +
  facet_wrap(
    ~ mode,
    labeller = as_labeller(
      c(car = "Car", public_transport = "Public transport")
    )
  ) +
  theme_void()
```
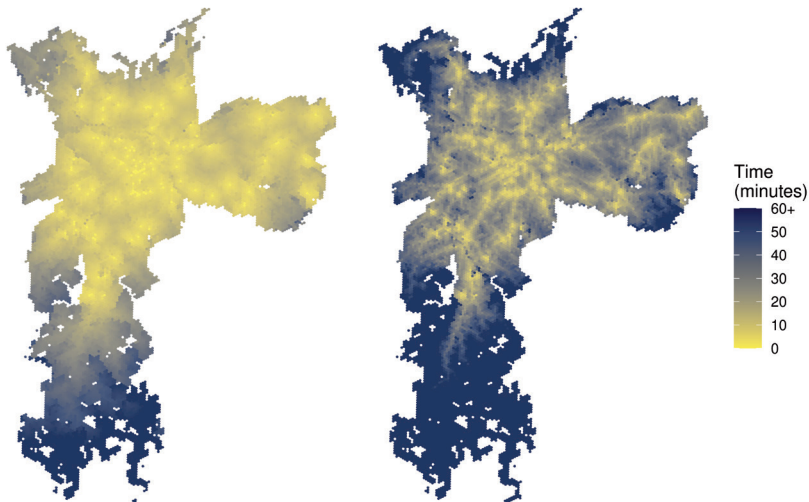
FIGURE 30
**Travel time to the closest high complexity public hospital in São Paulo**
Car                                             Public transport



Source: Figure generated by the code snippet above.

## 9.2 Map of employment accessibility

The accessibility dataset also makes it very easy to compare the number of accessible opportunities when considering different travel time thresholds. Using the code below, for example, we illustrate how to visualize, side-by-side, the spatial distribution of employment accessibility by public transport trips of up to 60 and 90 minutes.

```r
# determine min and max values for the legend
limit_values  <-c(0, max(access_pt $CMATT90, na.rm = TRUE) / 1000000)

fig60 <- ggplot(subset(access_pt, ! is.na(mode))) +
  geom_sf(aes(fill = CMATT60 / 1000000), color = NA, alpha = 0.9) +
  scale_fill_viridis_c(option = "inferno", limits = limit_values) +
  labs(subtitle = "Up to 60 minutes" , fill = "Jobs\n(millions)") +
  theme_void()

fig90 <- ggplot(subset(access_pt, ! is.na(mode))) +
  geom_sf(aes(fill = CMATT90 / 1000000), color = NA, alpha = 0.9) +
  scale_fill_viridis_c(option = "inferno", limits = limit_values) +
  labs(subtitle = "Up to 90 minutes", fill = "Jobs\n(millions)") +
  theme_void()

fig60 + fig90 + plot_layout(guides = "collect")
```
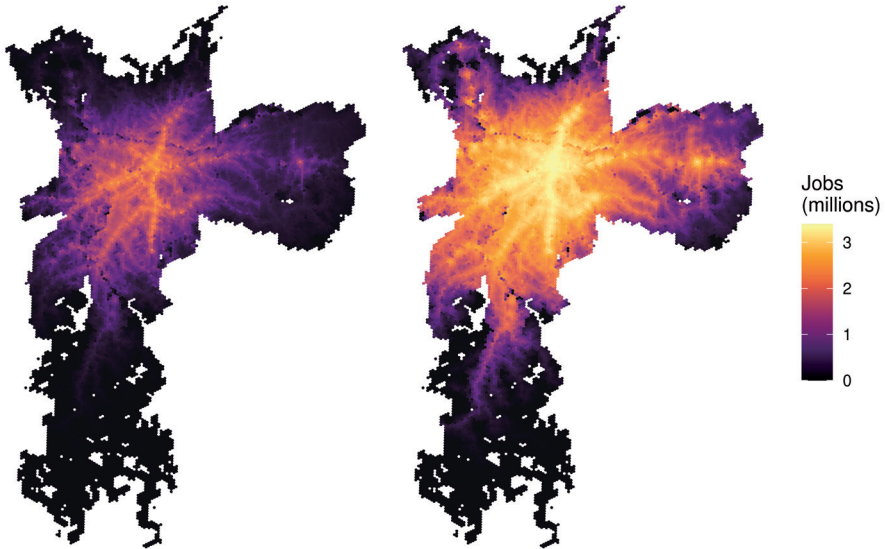
FIGURE 31
**Job accessibility by public transport in São Paulo**
Up to 60 minutes                          Up to 90 minutes



Source: Figure generated by the code snippet above.

## 9.3 Accessibility inequalities

Finally, {aopdata} accessibility dataset can be used to analyze accessibility inequalities across different Brazilian cities in several different ways. In this subsection, we present three examples of this type of analysis.

### 9.3.1 Inequality in travel time to access opportunities

In this first example, we compare the average travel time to the nearest high complexity public hospital for people of different income levels. To do this, we calculate, for each income group, the average travel time to reach the nearest high complexity health facility, weighted by the population of each grid cell. Weighting the travel time by population is necessary because each cell has a different population size, thus contributing differently to the average accessibility of the population as a whole.

Before performing the calculation, we should note that some grid cells cannot reach any high complexity hospital within two hours of travel. In these cases, the minimum travel time columns assume an infinite value (Inf). To deal with this situation in our example, we replace all Inf values by a travel time of 120 minutes.

```r
# copies access data into a new data.frame
ineq_pt <-data.table::as.data.table(access_pt)

# replaces Inf values with 120
ineq_pt [, TMISA := ifelse(is.infinite(TMISA), 120, TMISA)]

# calculates the average travel time by income decile
ineq_pt <- ineq_pt[
  ,
  .(avrg = weighted.mean(x = TMISA, w = P001, na.rm = TRUE)),
  by = R003
]
ineq_pt <- subset(ineq_pt, ! is.na(avrg))

ggplot(ineq_pt) +
  geom_col(aes(y = avrg, x = factor(R003)), fill =
  "#2c9e9e", color = NA) +
  scale_x_discrete(
      labels = c("D1\npoorest", paste0("D", 2:9), "D10\nwealthiest")
  ) +
  labs(x = "Income decile", y = "Travel time (minutes)") +
  theme_minimal()
```
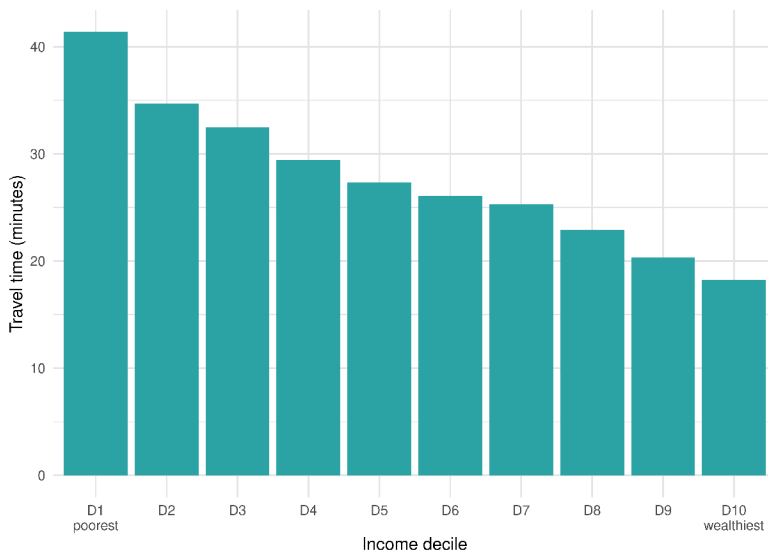
FIGURE 32

**Average travel time by public transport to the nearest high complexity hospital in São Paulo**



Source: Figure generated by the code snippet above.

9.3.2 Inequality in the number of accessible opportunities

Another way of examining accessibility inequalities is by comparing the number of opportunities that can be reached by different population groups considering the same transport modes and travel time limits. In this case, we analyze the active cumulative accessibility measure, represented by columns whose names start with CMA in the {aopdata} dataset. Using the code below, we compare the number of jobs accessible by people of different income deciles by public transport in up to 60 minutes.

```r
ggplot(subset(access_pt, !is.na(R003))) +
  geom_boxplot(
    aes(x = factor(R003), y = CMATT60 / 1000000, color = factor(R003))
  ) +
  scale_color_brewer(palette = "RdBu") +
  labs(
    color = "Income\ndecile",
    x = "Income decile",
    y = "Accessible jobs (millions)"
  ) +
  scale_x_discrete(
    labels = c("D1\npoorest", paste0("D", 2:9), "D10\nwealthiest")
  ) +
  theme_minimal()
```

FIGURE 33
**Distribution of job accessibility by public transport in up to 60 minutes of travel in São Paulo**
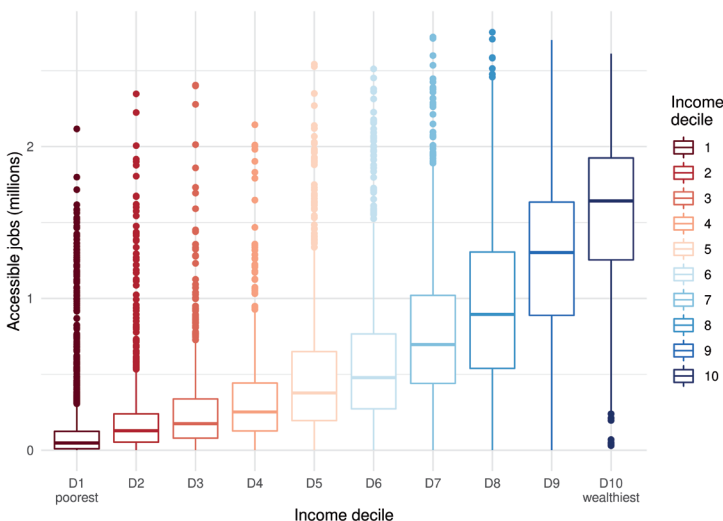


Source: Figure generated by the code snippet above.

Finally, we can also compare how the usage of different transport modes can lead to different accessibility levels and how the discrepancy between modes varies across cities. In the example below, we compare the number of jobs that one can access in up to 30 minutes of walking and driving. To do this, we first download accessibility estimates by both transport modes for all cities covered by AOP.

```
data_car <- aopdata::read_access(
  city = "all",
  mode = "car",
  year = 2019,
  showProgress = FALSE
)

data_walk <- aopdata::read_access(
  city = "all",
  mode = "walk",
  year = 2019,
  showProgress = FALSE
)
```

Next, we calculate, for each city and transport mode, the weighted average number of jobs accessible by trips of up to 30 minutes (CMATT30). We then join these estimates together into a single table and calculate the ratio between car and walk accessibility levels.

```
avg_car <- data_car[
  ,
  .(access_car = weighted.mean(CMATT30, w = P001, na.rm = TRUE)),
  by = name_muni
]

avg_walk <- data_walk[
  ,
  .(access_walk = weighted.mean(CMATT30, w = P001, na.rm = TRUE)),
  by = name_muni
]

# merges the data and calculates the ratio between access by
# car and on foot
avg_access <- merge(avg_car, avg_walk)
avg_access[, ratio := access_car / access_walk]

head(avg_access)
```

```
        name_muni access_car access_walk    ratio
1:          Belem   155270.4    9392.235 16.53179
2: Belo Horizonte   529890.0   12464.233 42.51284
3:       Brasilia   220575.9    4110.703 53.65892
4:       Campinas   256333.1    6748.923 37.98133
5:   Campo Grande   172680.5    4181.209 41.29919
6:       Curitiba   494376.9   10471.135 47.21331
```

Finally, we can analyze the results using a chart:

```
ggplot(avg_access, aes(x = ratio, y = reorder(name_muni, ratio))) +
  geom_bar(stat = "identity") +
  geom_text(aes(x = ratio + 3 , label = paste0(round(ratio), "x"))) +
  labs(y = NULL, x = "Ratio between car and walk accessibility") +
  theme_classic()
```

FIGURE 34
**Ratio between job accessibility levels by car and by foot considering trips of up to 30 minutes in the 20 biggest Brazilian cities**
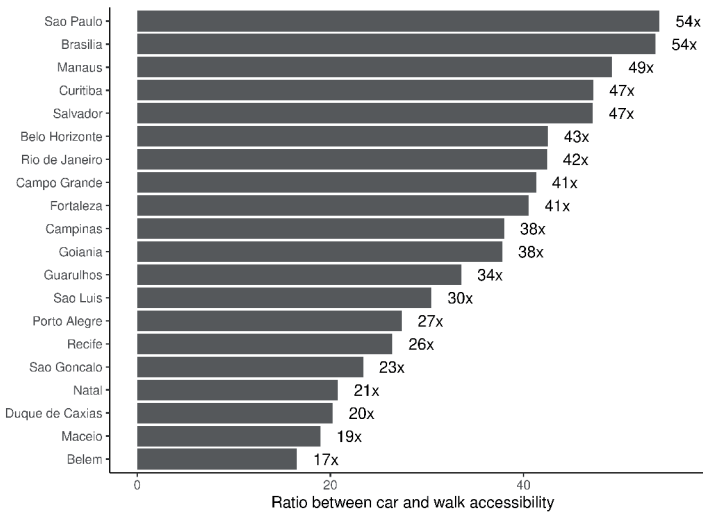


Source: Figure generated by the code snippet above.

As expected, figure 34 shows that car trips lead to much higher accessibility levels than equally long walking trips. This difference, however, greatly varies across cities. In São Paulo and Brasília, a 30-minute car trip allows one to access, on average, 54 times more jobs than what it would be possible to access with walking trips. In Belém, the city from our sample with the smallest difference, one can access 17 times more jobs by car than by foot – still a substantial difference, but much smaller than in other cities.