UNIVERSITÉ DE NEUCHÂTEL

Institut de recherches économiques

IRENE Working Paper 21-04

Travel distance and travel time using Stata: New features and major improvements in georoute

Sylvain Weber, Martin Péclat, August Warren

Travel distance and travel time using Stata: New features and major improvements in georoute*

Sylvain Weber† Martin Péclat‡ August Warren§

April 29, 2021

Abstract

The user-written command `georoute` is designed to calculate travel distance and travel time between two addresses or two geographical points identified by their coordinates. Since its conception and description by Weber and Péclat (2017), the command has been gradually maintained and enriched. The new version of `georoute` presented in this article encompasses major improvements, such as the possibility to specify transport mode and departure time. The new features open the way to a multitude of more sophisticated research applications.

JEL Classification: C87, R41.

Keywords: Stata, geocoding, travel distance, travel time.

*This research is part of the activities of SCCER CREST, which is financially supported by Innosuisse (grant n° KTI 1155000154.

†University of Neuchâtel, Institute of Economic Research, Rue Abram-Louis Breguet 2, 2000 Neuchâtel, Switzerland. sylvain.weber@unine.ch.

‡Romande Energie SA, Switzerland.

§Office of the State Superintendent of Education, Washington, DC 20002, USA

1 Introduction

Weber and Péclat (2017) introduced the user-written command `georoute`, which calculates travel distance and travel time between two points defined by their addresses or their geographical coordinates. Three years after publication, the Stata Journal article and Stata module received over 30 citations from applications in various research fields such as transportation (Theisen, 2020), health (e.g., Fischer et al., 2018; Myers et al., 2019; Myers and Ladd, 2020), medicine (Dayal et al., 2019), environment (Makov et al., 2020), or real estate (Theisen and Emblem, 2018), thereby demonstrating the usefulness of the command for a number of researchers.

The capabilities of the first version of `georoute` were relatively modest. Travel distance was calculated as the number of miles (or kilometers) one should drive by car to join the departure point to the destination point. No other transport means were possible, and it was not possible to introduce any kind of restriction on the road to be followed. Travel time was calculated as the time needed to cover the travel distance "under normal traffic conditions", admittedly a rather vague expression. Specifying departure time was not possible.

The new release of the command presented in this article overcomes several limitations. Important new options have been implemented to let the user specify transport mode, departure time, and various features of the route to be followed. The new version of `georoute` therefore constitutes a major improvement and will allow much more sophisticated applications than its first version.

2 The updated command

2.1 georoute

The syntax of `georoute` (version 3.0) is as follows:

```
georoute [if] [in], herekey(API KEY) {startaddress(varlist)|startxy(xvar yvar)}
    {endaddress(varlist)|endxy(xvar yvar)} [ tmode(string|varname) rtype(string|varname)
    traffic(string|varname) dtime(string,mask|varname|"now") avoid(string)
    softexclude(string) strictexclude(string) distance(newvar) time(newvar)
    diagnostic(newvar) coordinates(str1 str2) replace km timer pause observations
    nosettings ]
```

Main (compulsory)

`herekey`(*API KEY*) provides the credentials of the HERE application to be used.[1]

`startaddress(varlist)` and `endaddress(varlist)` specify the addresses of departure and destination. Addresses can be inserted as a single variable or as a variable list. Alternatively, `startxy()` and `endxy()` can be used. Either `startaddress()` or `startxy()` is required. Either `endaddress()` or `endxy()` is required.

`startxy(xvar yvar)` and `endxy(xvar yvar)` specify the geographical coordinates (in decimal degrees) of the departure and destination points. They can be used as an alternative to `startaddress()` and `endaddress()`. Two numeric variables containing latitude (x) and longitude (y) coordinates of the starting and ending points must be provided in `startxy()` and `endxy()`.

Routing options

`tmode`(*string*|*varname*) specifies the transport mode. The default is `tmode`("*car*"). The following transport modes are available:

- *car*
- *carHOV*
- *pedestrian*
- *publicTransport*
- *publicTransportTimeTable*
- *truck*
- *bicycle*

Transport modes can be specified either via a string (for instance `tmode`("*car*")) or via a variable (for instance `tmode`(*vehicle*)). When a string is used, the same transport mode will be used for all observations. Using a variable makes it possible to specify transport mode at the observation level, in which case the variable must be a string variable composed of the transport modes typed exactly as above (mind the capitalized letters). Any missing values will be assigned the default transport mode ("*car*").

`rtype`(*string*|*varname*) specifies the routing type. The default is `rtype`("*balanced*"). The following routing types available are:

- *fastest*
- *shortest*
- *balanced*

[1] In HERE applications created before December 2019, credentials consisted of an APP ID and an APP CODE. In earlier versions of `georoute`, the application credentials were therefore provided using `hereid`(*APP ID*) and `herecode`(*APP CODE*). These options are still available but are not documented anymore as they might become obsolete in the future. The same applies to `herepaid`. See Weber and Péclat (2017) for more information.

Routing types can be abbreviated to a minimum of 1 letter: 'f' for *fastest*, 's' for *shortest*, 'b' for *balanced*.

Routing types can be specified either via a string (for instance `rtype`("*fastest*")) or via a variable (for instance `rtype`(*routing*)). When a string is used, the same routing type will be used for all observations. Using a variable makes it possible to specify routing type at the observation level, in which case the variable must be a string variable composed of the routing types indicated above (possibly abbreviated). Any missing values will be assigned the default routing type ("*balanced*").

`traffic`(*string*|*varname*) specifies whether to optimize a route for traffic. The default is `traffic`("*default*"). The following traffic modes are available:

- *enabled*
- *disabled*
- *default*

Traffic modes can be abbreviated to a minimum of 2 letters: 'en' for *enabled*, 'di' for *disabled*, 'de' for *default*.

Traffic modes can be specified either via a string (for instance `traffic`("*enabled*")) or via a variable (for instance `traffic`(*traf*)). When a string is used, the same traffic mode will be used for all observations. Using a variable makes it possible to specify traffic mode at the observation level, in which case the variable must be a string variable composed of the traffic modes indicated above (possibly abbreviated). Any missing values will be assigned the default traffic mode ("*default*").

Note that traffic mode has no impact for transport modes "*pedestrian*" and "*bicycle*".

`dtime`(*string*,*mask*|*varname*|"*now*") specifies the date and time when travel is expected to start. The default is `dtime`("*now*"), i.e., the current time as of running the calculation.

Departure time can be specified either via a string and a mask (for instance `dtime`("01Jul2020 08:00:00", "DMYhms"); see `clock`) or via a variable (for instance `dtime`(*t*)). When a string is used, the same departure time will be used for all observations. Using a variable makes it possible to specify departure time at the observation level, in which case the format of the variable must be %tc or %tC. Any missing values will be assigned the default departure time ("*now*").

Note that departure time has no impact for transport modes "*pedestrian*" and "*bicycle*". The extent to which it is possible to calculate travel times in the past depends on other parameters, in particular the transport mode specified in `tmode`(). It moreover seems that historical traffic data is not available from HERE API before January 2020, resulting in a travel time that is independent of departure time before then.

`avoid`(*string*), `softexclude`(*string*) and `strictexclude`(*string*) specify routing features to be avoided / softly excluded / strictly excluded. The following routing features are available:

- *tollroad*
- *motorway*
- *boatFerry*
- *railFerry*
- *tunnel*
- *dirtRoad*
- *park*

Routing features can be abbreviated to a minimum of 2 letters: 'to' for *tollroad*, 'mo' for *motorway*, 'bo' for *boatFerry*, 'ra' for *railFerry*, 'tu' for *tunnel*, 'di' for *dirtRoad*, 'pa' for *park*.

Note that putting restriction of feature "*park*" has no impact for transport modes other than "*pedestrian*" and "*bicycle*".

New variables

`distance(`*newvar*`)` creates a new variable containing the travel distance between departure and destination points. If `distance()` is not specified, travel distance will be stored in a variable named *travel_distance*.

`time(`*newvar*`)` creates a new variable containing the travel time between departure and destination points. If `time()` is not specified, travel time will be stored in a variable named *travel_time*.

`diagnostic` creates a new variable containing a diagnostic code for the geocoding and georouting outcome of each observation in the database: 0 = OK, 1 = No route found, 2 = Start and/or end not geocoded, 3 = Start and/or end coordinates missing, 4 = No route searched. If `diagnostic()` is not specified, diagnostic codes will be stored in a variable named *georoute_diagnostic*.

`coordinates(`*str1 str2*`)` creates new variables containing the coordinates and the match code of the starting (*str1_x*, *str1_y*, *str1_match*) and ending (*str2_x*, *str2_y*, *str2_match*) addresses. This option is irrelevant if geographical coordinates (rather than addresses) are provided for departure and destination points. If `coordinates()` is not specified, coordinates and match code will not be saved. The match code indicates how well the result matches the request in a 4-point scale: 1 = exact, 2 = ambiguous, 3 = upHierarchy, 4 = ambiguousUpHierarchy.

`replace` specifies that the variables in `distance`, `time`, `coordinates`, and `diagnostic` may be replaced if they already exist in the database. It should be used cautiously because it might definitively drop some data.

Reporting

`km` specifies that distances should be returned in kilometers. The default is to return distances in miles.

`timer` requests that a timer is printed while geocoding. If specified, a dot is printed for every centile of the dataset that has been geocoded and a number is printed every 10%. When geocoding large numbers of observations, this option will let the user when to expect the end.

`pause` can be used to slow the geocoding process by asking Stata to sleep for 30 seconds every 100$^{\text{th}}$ observation. This could be useful for large databases, which might overload the HERE API and result in missing values for batches of observations.

`observations` can be used to print a detailed observation account, showing how many observations were discarded and why.

`nosettings` suppresses display of the settings report.

2.2 georoutei

In order to facilitate quick requests for single pairs of addresses or geographical coordinates, a command with immediate arguments proves handy. The syntax of `georoutei` is similar to that of `georoute`, except that all arguments must be parsed as strings:[2]

```
georoutei , herekey(API KEY) {startaddress(string)|startxy(#x,#y)}
    {endaddress(string)|endxy(#x,#y)} [ tmode(string) rtype(string) traffic(string)
    dtime(string,mask|"now") avoid(string) softexclude(string) strictexclude(string) km
    nosettings ]
```

2.2.1 Saved results

`georoutei` saves the following results in `r()`:

Scalars

`r(dist)`	Travel distance	`r(time)`	Travel time
`r(startx)`	x-coordinate (latitude) of departure point		
`r(starty)`	y-coordinate (longitude) of departure point		
`r(endx)`	x-coordinate (latitude) of destination point		
`r(endy)`	y-coordinate (longitude) of destination point		

[2]Note also that the options relative to the creation of new variables and some reporting options are not available with the immediate command because they are irrelevant in this context.

3 Examples

3.1 Using georoutei

We start our demonstration with the immediate command `georoutei`, which is simpler to use than the full one. For new users, it is probably a good idea to get familiar with the immediate command before trying to use the full one.

Let's say we are interested in traveling from the New York Stock Exchange to the Madison Square Garden. We would then type the following:[3]

```
. global nyse "11 Wall St, New York, NY 10005, USA"
. global msg "4 Pennsylvania Plaza, New York, NY 10001, USA"
. georoutei, herekey("$apikey") startad("$nyse") endad("$msg")
SETTINGS
------------------------------------------------------------------------------
Start:     11 Wall St, New York, NY 10005, USA (40.70714,-74.01086)
End:       4 Pennsylvania Plaza, New York, NY 10001, USA (40.75067,-73.99235)
Mode:      car (assigned by default)
Route:     balanced (assigned by default)
Traffic:   default (assigned by default)
Departure: 13 Mar 2021 17:04:13 (now; assigned by default)
------------------------------------------------------------------------------

ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:       3.61 miles
Travel time:          18.45 minutes
------------------------------------------------------------------------------
```

Note that we have started by storing the addresses in global macros so as to simplify the code, and also to be able to reuse the same addresses later without having to retype them.

The output indicates that travel distance between these two places is around 3.6 miles, and it could be covered in a bit more than 18 minutes. Said otherwise, the average speed (that could be computed by typing `display r(dist)/r(time)*60`) would be lower than 12 MPH. Moreover, note that since we have not specified anything else than the departure and destination points, all the options have been automatically set to their default. The first block of output shows the settings and indeed indicates that transport mode was assigned to "car", routing type was assigned to "balanced", traffic mode was assigned to "default", and departure time was assigned to "now" (which is the time when the user runs the command).

With the previous release of the command, none of the default settings could be altered. With the new version, however, we have the possibility to change, among others, the transport mode:

[3]Before running these lines, store the API KEY for your HERE application in the global macro $apikey.

```
. georoutei, herekey("$apikey") startad("$nyse") endad("$msg") tmode("bicycle")
SETTINGS
------------------------------------------------------------------------------
Start:     11 Wall St, New York, NY 10005, USA (40.70714,-74.01086)
End:       4 Pennsylvania Plaza, New York, NY 10001, USA (40.75067,-73.99235)
Mode:      bicycle
Route:     balanced (assigned by default)
Traffic:   (no impact with selected transport mode)
Departure: (no impact with selected transport mode)
------------------------------------------------------------------------------

ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:       3.52 miles
Travel time:          30.40 minutes
------------------------------------------------------------------------------

. georoutei, herekey("$apikey") startad("$nyse") endad("$msg") tmode("publicTran
> sport")
SETTINGS
------------------------------------------------------------------------------
Start:     11 Wall St, New York, NY 10005, USA (40.70714,-74.01086)
End:       4 Pennsylvania Plaza, New York, NY 10001, USA (40.75067,-73.99235)
Mode:      publicTransport
Route:     balanced (assigned by default)
Traffic:   default (assigned by default)
Departure: 13 Mar 2021 17:04:14 (now; assigned by default)
------------------------------------------------------------------------------

ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:       3.61 miles
Travel time:          22.37 minutes
------------------------------------------------------------------------------
```

We observe that travel distance (in this case) is a bit shorter with a bicycle than with a car, which is caused by the HERE API considering some roads (such as those located in parks) as open to cyclists but not to cars. Travel time is nevertheless naturally longer with a bicycle. Note also that the output signals that accounting for traffic conditions and departure time would have no impact on the outcome for bicycle. Travel time calculated with public transport is between the durations calculated for car and bicycle.

Rather than letting the time of departure be assigned by default (and therefore possibly obtain a different outcome after every run), we could decide to specify the date and time of departure, and simultaneously take into account historical traffic information:

```
. georoutei, herekey("$apikey") startad("$nyse") endad("$msg") tm("car") dt("12m
> ar2021 12:00", "DMYhm") traffic("enabled") noset
ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:       3.62 miles
Travel time:          28.17 minutes
------------------------------------------------------------------------------

. georoutei, herekey("$apikey") startad("$nyse") endad("$msg") tm("car") dt("12m
> ar2021 23:59", "DMYhm") traffic("enabled") noset
ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:       3.96 miles
Travel time:          17.05 minutes
------------------------------------------------------------------------------
```

Here we additionally used option `nosettings` to save space by suppressing the display of the settings. The results show that both travel distance and travel time would be

different if driving at noon or at midnight (on Friday, March 12, 2021). Travel distance changes with hours of the day in particular because of possible time-zone restrictions, while travel time is of course affected by congestion during peak hours.

The last series of options available with `georoutei` make it possible to introduce various restrictions on the route to be followed (or not). To illustrate these options, we consider a trip from Geneva (Switzerland) to Saint-Tropez (France):

```
. global GVA "Geneva, Switzerland"
. global ST "Saint-Tropez, France"
. georoutei, herekey("$apikey") startad("$GVA") endad("$ST") km noset
ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:     562.00 kilometers
Travel time:         332.02 minutes
------------------------------------------------------------------------------

. georoutei, herekey("$apikey") startad("$GVA") endad("$ST") rtype("short") km n
> oset
ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:     444.00 kilometers
Travel time:         473.60 minutes
------------------------------------------------------------------------------

. georoutei, herekey("$apikey") startad("$GVA") endad("$ST") avoid("toll") km no
> set
ROUTE CALCULATED
------------------------------------------------------------------------------
Travel distance:     480.20 kilometers
Travel time:         514.68 minutes
------------------------------------------------------------------------------
```

For this trip, we observe that the default settings yield a travel distance of more than 560 kilometers (we used option `km` to respect the standard European units for this example). If we ask the API to look for the shortest distance, using option `rtype`("*short*"), we obtain a travel distance that is more than 100 km shorter. Notice that travel duration is however much longer. These differences are explained by the fact the shortest route goes through mountainous regions using small roads. Contrarily, the highway forces drivers to make detours, but permits of course a much faster speed. Finally, note that avoiding toll roads, using option `avoid`("*toll*"), results in the longest trip duration. This clearly illustrates how the French road legislation imposes tradeoffs between travel costs and travel time, which certainly contributed to the discontent that dramatically led to the "yellow vests movement".[4]

3.2 Using georoute

The full command `georoute` makes it possible to calculate travel distances and travel times for batches of observations. Concretely, it does exactly the same requests as `georoutei`, but for each observation of the dataset and therefore returns the results

[4]See for instance The Economist (27.11.2018): "What, and who, are France's 'gilets jaunes'?".

in variables rather than printing them in the results window.

To show the equivalence between the two commands, we create a small dataset and replicate the calculations of travel distance and travel time between the New York Stock Exchange and the Madison Square Garden as we already did above:

```
. clear
. set obs 4
number of observations (_N) was 0, now 4
. gen place1 = "11 Wall St, New York, NY 10005, USA"
. gen place2 = "4 Pennsylvania Plaza, New York, NY 10001, USA"
. qui: gen vehicle = "car" in 1
. qui: replace vehicle = "bicycle" in 2
. qui: replace vehicle = "publicTransport" in 3
. qui: replace vehicle = "publicTransportTimeTable" in 4
. georoute, herekey("$apikey") startad(place1) endad(place2) tm(vehicle)
georoute has successfully calculated travel distance and travel time for 4 obser
> vations based on the following settings:
------------------------------------------------------------------------------
Start:     adresses in place1 (variable)
End:       adresses in place2 (variable)
Mode:      transport modes in vehicle (variable)
Route:     balanced (assigned by default) for all observations (hard coded)
Traffic:   default (assigned by default) for all observations (hard coded)
Departure: 13 Mar 2021 17:04:21 (now; assigned by default) for all observations
> (hard coded)
------------------------------------------------------------------------------
. list vehicle travel_distance travel_time

     +-------------------------------------------------+
     |                  vehicle   trave~ce   trave~me |
     |-------------------------------------------------|
  1. |                      car   3.607681      18.45 |
  2. |                  bicycle   3.523175       30.4 |
  3. |            publicTransport   3.610167   22.36667 |
  4. | publicTransportTimeTable   3.612652   18.68333 |
     +-------------------------------------------------+
```

The first three observations show that results are strictly equivalent as those obtained with `georoutei` above in section 3.1.[5] The last observation additionally shows the results obtained with transport mode "publicTransportTimeTable", which allows to highlight the difference with "publicTransport". Travel time is shorter when officially published timetable information is considered: with that option, the assumption is that the passenger will not depart before there is a good connection. Note however that "publicTransportTimeTable" only works for a relatively recent period of time.

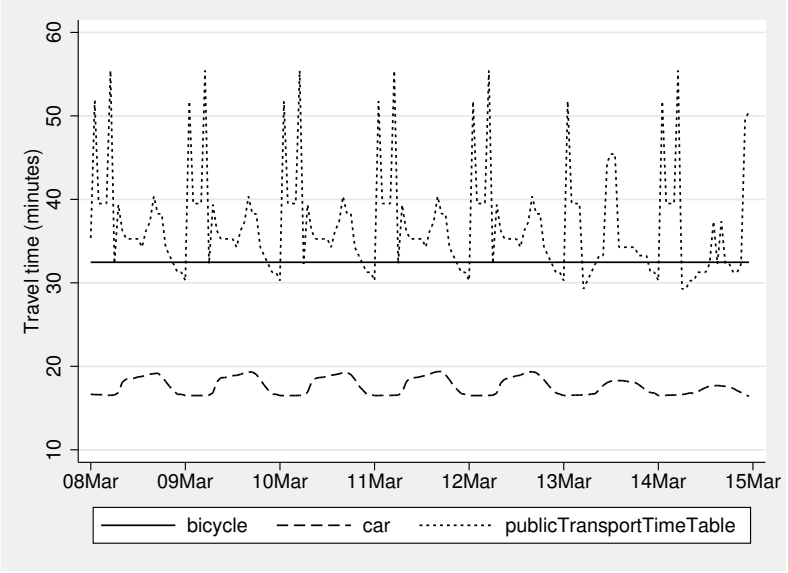
Finally, let us consider the trip from the United Nations' building in Geneva (Switzerland) to Geneva's International Airport. In distance, this is a short trip (less than 5 kilometers). However, because of important congestion, travel time may sometimes be quite long (which may have serious consequences when trying to catch a plane). We now construct a database to compare travel time using three different transport modes and for each hour of a week:

[5]The equivalence holds as long as the two commands are issued approximately at the same time, since departure time is set by default at the current time.

```
. clear
. cap: georoutei, herekey($apikey) startad("Palais des Nations, 1211 Genève, Swi
> tzerland") endad("Route de l´Aéroport 21, 1215 Le Grand-Saconnex, Switzerland"
> )
. set obs 3
number of observations (_N) was 0, now 3
. gen startx = r(startx)
. gen starty = r(starty)
. gen endx = r(endx)
. gen endy = r(endy)
. qui: gen vehicle = "car" in 1
. qui: replace vehicle = "bicycle" in 2
. qui: replace vehicle = "publicTransportTimeTable" in 3
. expand 24*7
(501 observations created)
. local today = date(c(current_date),"DMY")
. local Monday1: di %td `today´ - 6 - dow(`today´)
. local Monday2: di %td `today´ + 1 - dow(`today´)
. bys vehicle: gen double t = tc(`Monday1´ 00:00) + msofminutes(60)*(_n-1)
. format t %tc
. georoute, herekey("$apikey") startxy(startx starty) endxy(endx endy) tm(vehicl
> e) dt(t) traffic("enabled") timer observations
---------
Geocoding
---------
......... 10% ......... 20% ......... 30% ......... 40% ......... 50% .........
> 60% ......... 70% ......... 80% ......... 90% ......... 100%
georoute has successfully calculated travel distance and travel time for 504 obs
> ervations based on the following settings:
------------------------------------------------------------------------------
Start:     coordinates in startx starty (variables)
End:       coordinates in endx endy (variables)
Mode:      transport modes in vehicle (variable)
Route:     balanced (assigned by default) for all observations (hard coded)
Traffic:   enabled for all observations (hard coded)
Departure: times in t (variable)
------------------------------------------------------------------------------
Detailed observation account:
------------------------------------------------------------------------------
Total in database:                          504
Total after if/in condition(s):             504
Considered for geocoding:                   504
Successfully coded:                         504
------------------------------------------------------------------------------
. encode vehicle, gen(v)
. xtset v t, delta(60 min)
       panel variable:  v (strongly balanced)
        time variable:  t, 08mar2021 00:00:00 to 14mar2021 23:00:00
                delta:  1 hour
. xtline travel_time, over ylab(10(10)60) ysc(r(10 60)) xlab(`=tc(`Monday1´ 00:0
> 0)´(`=msofminutes(60)*24´)`=tc(`Monday2´ 00:00)´, format(%tcDDMon)) xti("") le
> gend(row(1))
```

In the above example, note that we have used `georoutei` in a preliminary step to geocode the addresses. The obtained geographical coordinates were then saved in variables and fed into `georoute`. The rationale for doing this is to save multiple requests for the same addresses. If the addresses were introduced in `georoute`, the same pair of addresses would in fact be geocoded as many times as the number of observations, which is clearly a waste of resources (the number of requests is capped at 250,000 per month with a free HERE account) and time. Feeding the geographical coordinates in this case permits to

Figure 1: Travel time over the week and for different transport mode



save requests to the HERE API and speeds up the process. The database is then expanded and a variable `t` is created with each hour of a one-week period starting Monday, March 8, 2021 at 00:00.

Travel times (and distances) for all 504 observations of the database were then calculated using a single call to `georoute`. Because we expected the geocoding could take a few moments, we made use of the `timer` option to keep track of the evolution of the process.

Figure 1 displays the travel times calculated for each transport mode. We observe that travel time is constant with a bicycle, as expected because this transport mode is not affected by traffic and departure time. There are however large variations for car and public transport. Travel time by car is longer during the day, especially on weekdays. The pattern is different for public transportation, for which travel time is longer at night when connections are less frequent. In average, we also notice that travel time is shorter by bicycle than by public transportation, a specific characteristic in the city of Geneva.

4 Caveats

The user-written Stata command `georoute` automates requests from the HERE API, which is convenient for extracting travel distances and times. However, the HERE API might suffer from some weaknesses. For instance, the authors of this article have en-

countered situations which created strange outcomes.[6] `georoute` should not be considered as responsible for the quality of the results, which only depend on the HERE API. Therefore, `georoute` users are invited to use the HERE documentation (`https://developer.here.com/`) to understand exactly each parameter of the requests. The authors of this article are not affiliated in any respect with HERE and should not be considered as experts of this platform.

5 Conclusion

This article presents the new functionalities available in the updated version of `georoute`, a command to determine travel distances and times between addresses or geographical coordinates. As illustrated in various examples, the improved version of the command opens the way to a multitude of potential applications. While this was not possible with the initial version, the new command allows comparisons of the performance of competing transport modes in terms of time and distance for a specific trip. It also becomes possible, among others, to determine the most appropriate departure time.

The `georoute` command, like all API-based commands, is dependent on the stability of these interfaces. This risk, which we already mentioned in Weber and Péclat (2017), has so far proved to be moderate. Although we had to implement updates in recent years, these have been marginal and mostly related to the HERE credentials. The new functionalities offered by HERE, which enabled us to improve `georoute`, have been added without modifying the basic operation of the API. On this basis, we believe that `georoute` will continue to benefit the Stata user community in the future.

References

Dayal, P., Chang, C. H., Benko, W. S., Ulmer, A. M., Crossen, S. S., Pollock, B. H., Hoch, J. S., Kissee, J. L., Warner, L., and Marcin, J. P. (2019). Appointment completion in pediatric neurology telemedicine clinics serving underserved patients. *Neurology: Clinical Practice*, 9(4):314–321.

Fischer, S., Royer, H., and White, C. (2018). The impacts of reduced access to abortion and family planning services on abortions, births, and contraceptive purchases. *Journal of Public Economics*, 167:43–68.

[6]For instance, trying to compute travel distance from Roma (Italy) to Dubrovnik (Croatia) by car and excluding boat ferries yields an error. Setting the transport mode on truck however yields a sensible outcome.

Makov, T., Shepon, A., Krones, J., Gupta, C., and Chertow, M. (2020). Social and environmental analysis of food waste abatement via the peer-to-peer sharing economy. *Nature communications*, 11(1):1–8.

Myers, C., Jones, R., and Upadhyay, U. (2019). Predicted changes in abortion access and incidence in a post-roe world. *Contraception*, 100(5):367–373.

Myers, C. and Ladd, D. (2020). Did parental involvement laws grow teeth? the effects of state restrictions on minors' access to abortion. *Journal of Health Economics*, 71:102302.

Theisen, T. (2020). The impact of an urban toll ring on housing prices. *Research in Transportation Economics*, 82:100882. Special Issue in honour of Finn Jørgensen.

Theisen, T. and Emblem, A. W. (2018). House prices and proximity to kindergarten – costs of distance and external effects? *Journal of Property Research*, 35(4):321–343.

Weber, S. and Péclat, M. (2017). A simple command to calculate travel distance and travel time. *Stata Journal*, 17(4):962–971.