GPS2space

An Open-source Python Library for Spatial Measure Extraction from GPS Data

Shuai Zhou and Yanling Li*
Pennsylvania State University

QuantDev Brownbag
February 10, 2021, University Park, PA

* We appreciate contributions from other co-authors: **Guangqing Chi, Zita Oravecz, Yosef Bodovski, Naomi Friedman, Scott Vrieze, & Sy-Miin Chow**

Introduction

Spatial analyses have gained popularity in social, behavioral, and environmental sciences for the following reasons:

- The development of spatial methods and spatial computational power;
- The availability of spatial data from multiple resources.

The voice calling for “making a place for space” (Logan, 2012) is particularly strong in social science.

However, conventional non-programmable spatial analysis tools face (1) license restrictions, (2) reproducibility issues and, (3) are often incapable and impractical in dealing with big data.

Research objectives

1. To introduce GPS2space;
2. To demonstrate the utility of GPS2space with code examples;
3. To apply GPS2space to the Colorado Online Twin Study (CoTwins) and explore the seasonal, age, gender, and zygosity effects in shaping the twins' activity space and shared space.

Commonly used spatial analysis tools

Conventional spatial analysis software:

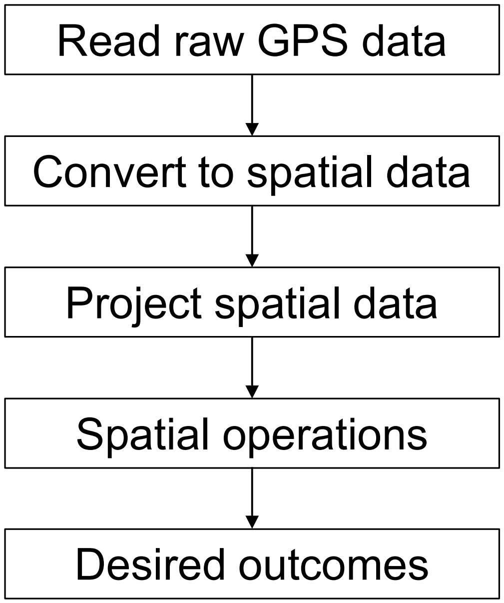
	Free	Programmable	On HPC
ArcGIS	x	√	x
TransCAD	x	x	x
MapInfo	x	x	x
QGIS	√	√	x

Python spatial analysis packages:

GDAL	Supports 168 raster data formats and 99 vector data formats
Fiona	Work with vector data
Rasterio	Work with raster data
Pyproj	Spatial projection and coordinate transformation
Shapely	Spatial operation
PySAL	Exploratory Spatial Data Analysis (ESDA) and spatial modeling
GeoPandas	A combination of GIS and Pandas

Limitations of commonly used Python spatial analysis packages

- They assume users have sufficient knowledge in GIS and programming
- Users should go through several steps and correctly specify the parameters at each step
- Their units of measure may not be intuitive for users unfamiliar with GIS
- Those packages do not provide readily available functions for spatial measure

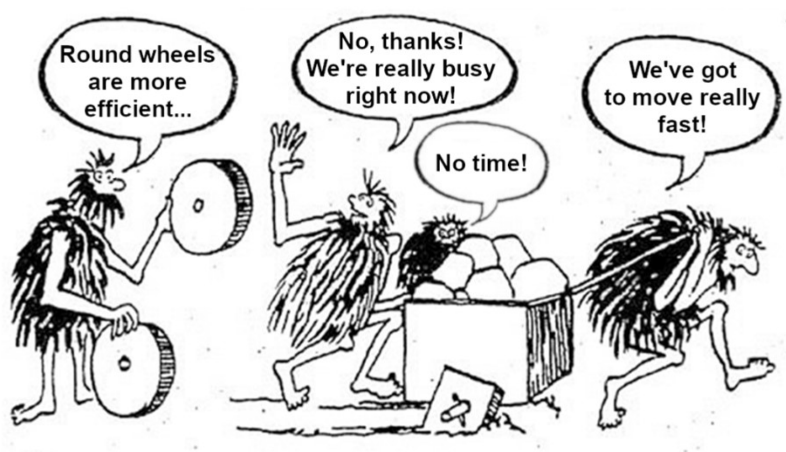


Contributions of GPS2space

- GPS2space provides readily replicable and open-source solution to working with GPS data

- GPS2space provides default parameterizations while also allows custom specifications of the parameters

- GPS2space extends the spatialities of GPS data




```
│       gitignore
│       LICENSE
│       README.md
│       setup.py
│
├───data
│   │   example.csv
│   │   .
│   │   .
│   │   .
│   │
│   └───rawdata
│           airprtx020.dbf
│           .
│           .
│           .
│
├───gps2space
│   │   dist.py
│   │   geodf.py
│   │   space.py
│   │   __init__.py
│   │
│   └───__pycache__
│           dist.cpython-37.pyc
│           geodf.cpython-37.pyc
│           __init__.cpython-37.pyc
│
└───notebooks
    │   createdata.ipynb
    │
    └───.ipynb_checkpoints
            createdata-checkpoint.ipynb
```

Illustrative examples

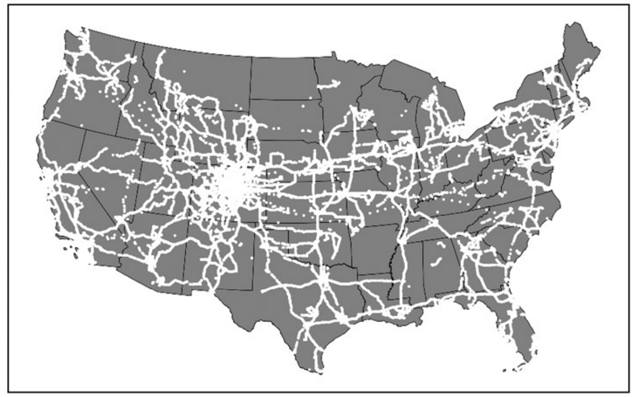
Colorado **O**nline **Twin S**tudy (CoTwins)

- Participants
  - 350 adolescent twins (670 individuals) aged from 14 to 17 at enrollment
- Time period
  - June 2016 to November 2018
- Assessment
  - Substance use (i.e., alcohol, marijuana, tobacco)
- Real-time location
  - iOS device: every time participants moved a significant distance (i.e., 500 meters or more)
  - Android device: every 5 minutes

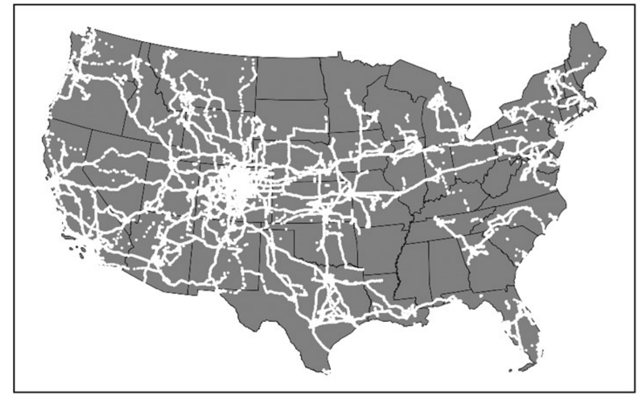
Illustrative examples (Cont.)

(a) Distribution of the twins' geolocations in the US in 2016

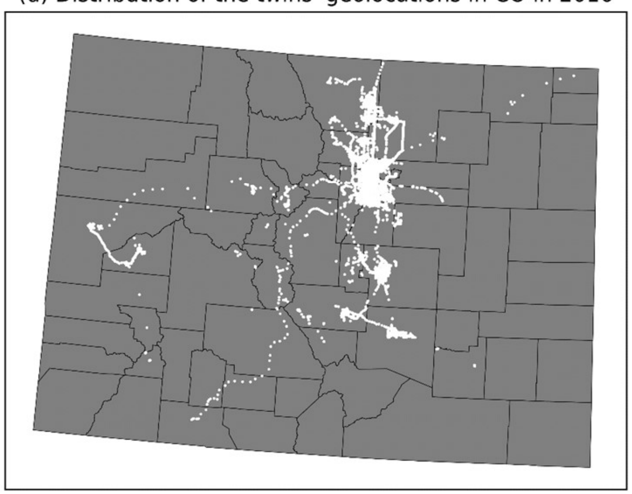
(b) Distribution of the twins' geolocations in the US in 2017



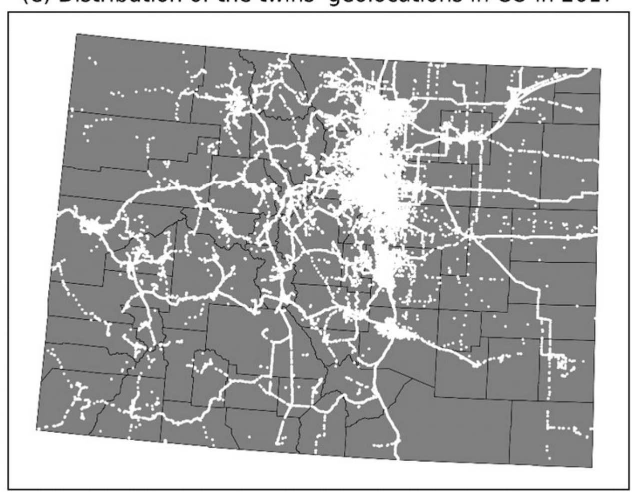
(c) Distribution of the twins' geolocations in the US in 2018



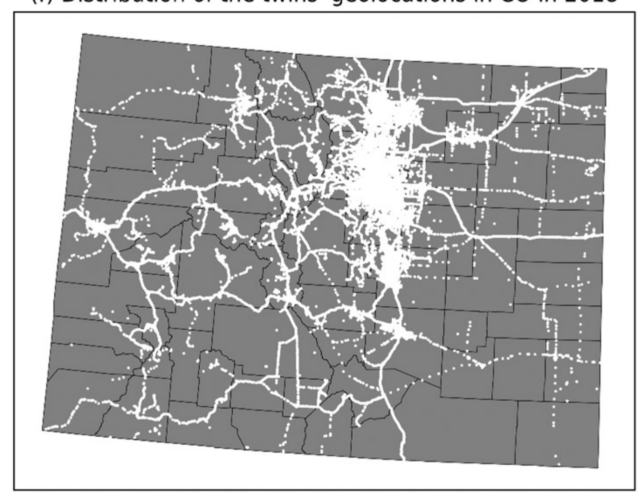
(d) Distribution of the twins' geolocations in CO in 2016



(e) Distribution of the twins' geolocations in CO in 2017



(f) Distribution of the twins' geolocations in CO in 2018



geodf: Building spatial data from raw GPS data

```
gdf = geodf.df_to_gdf(df, x='your_long_column', y='your_lat_column’)
```

- *df* is the raw GPS dataframe with raw Lat/Long coordinate pairs
- *x* is the column name that indicates the longitude
- *y* is the column name that indicates the latitude

geodf (Cont.)

Code example: Building spatial data

```
# Import required libraries for the analyses.
import pandas as pd
import geopandas as gpd
from gps2space import geodf, space, dist

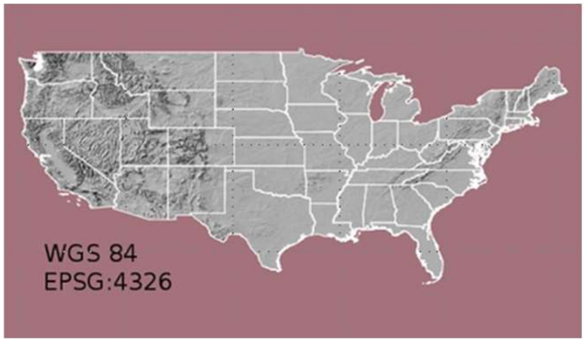
# Use the read_csv function from the Pandas library to read in raw latitude and longitude coordinate pairs as Pandas dataframe and assign df_twinXa_512 and df_twinXb_512 to the dataframe, respectively. TwinXa and TwinXb represents each of the twin pairs, respectively. The same designating approach is applied to the rest of the twin pairs in the CoTwins study. We use relative file path throughout the examples, users should use their own file path, either absolute or relative.
df_twinXa_512 = pd.read_csv('./data/TwinXa_512.csv')
df_twinXb_512 = pd.read_csv('./data/TwinXb_512.csv')

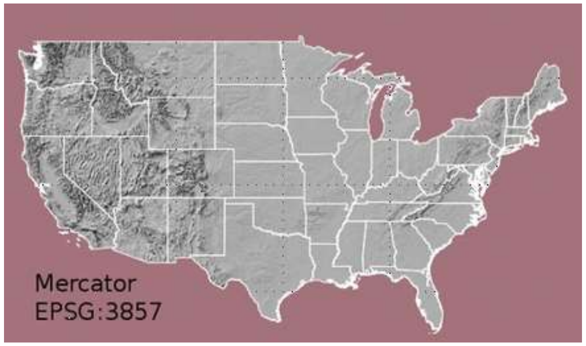
# Convert dataframe to GeoPandas dataframe using df_to_gdf function from GPS2space and assign gdf_twinXa_512 and gdf_twinXb_512 to the GeoPandas dataframe. x and y refer to the column names of the longitude and latitude, respectively, and must not be specified the other way around.
gdf_twinXa_512 = geodf.df_to_gdf(df_twinXa_512, x='longitude', y='latitude')
gdf_twinXb_512 = geodf.df_to_gdf(df_twinXb_512, x='longitude', y='latitude')
```

space: Spatial measure extraction

```
buffer_space = space.buffer_space(gdf, dist=100, dissolve='time_variable', proj=2163)
convex_space = space.convex_space(gdf, group='time_variable', proj=2163)
```

- *gdf* is the unprojected spatial dataframe
- *dist* is the buffer distance whose unit of measure is related to *proj*
- *dissolve/group* is the level of aggregating to form polygons/multi-polygons
- *proj* is the EPSG codes for your projection





space (Cont.)

Code example: Constructing activity space and shared space

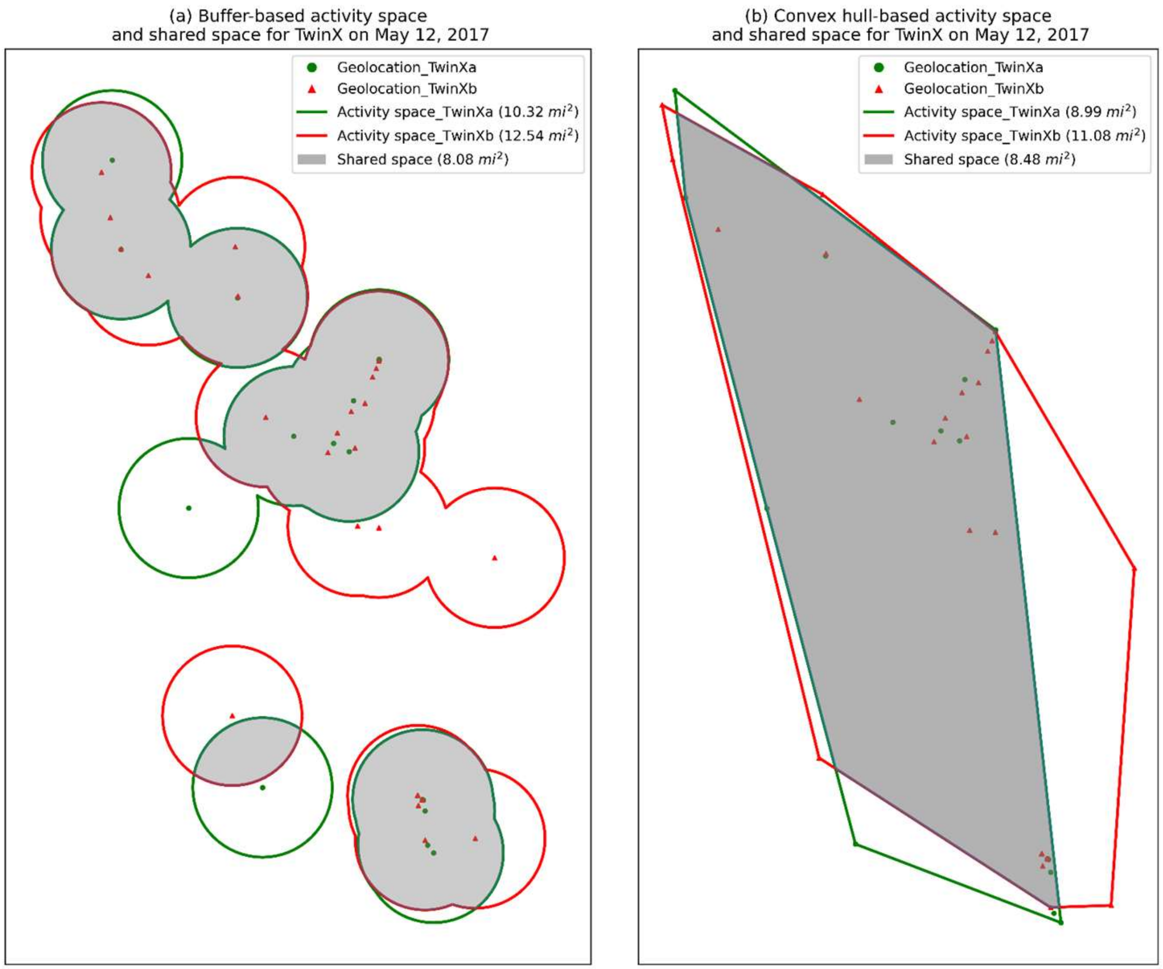
```
# Calculate buffer- and convex hull-based activity space using the
space.buffer_space and space_convex_space functions from GPS2space. The
dissolve and group accept the time variable (in this case, day). proj is the
EPSG identifier which should be specified accordingly, depending on the research
area.
buff_twinXa_512 = space.buffer_space(gdf_twinXa_512, dist=1000,
                                     dissolve='day', proj=2163)
buff_twinXb_512 = space.buffer_space(gdf_twinXb_512, dist=1000,
                                     dissolve='day', proj=2163)

convex_twinXa_512 = space.convex_space(gdf_twinXa_512,
                                       group='day', proj=2163)
convex_twinXb_512 = space.convex_space(gdf_twinXb_512,
                                       group='day', proj=2163)

# Calculate shared space from activity space using the overlay function from
GeoPandas and name the column "share_space".
buff_share = gpd.overlay(buff_twinXa_512, buff_twinXb_512,
                         how='intersection')
buff_share['share_space'] = buff_share['geometry'].area
convex_share = gpd.overlay(convex_twinXa_512, convex_twinXb_512,
                           how='intersection')
convex_share['share_space'] = convex_share['geometry'].area
```

space (Cont.)

Pros and cons of buffer method and convex hull method



dist: Measuring the nearest distance

```
distance = dist.dist_to_point(gdf_origin, gdf_destination, proj=2163)
```

- *gdf_origin* is the place of origin
- *gdf_destination* is the place of destination
- *proj* is the EPSG codes for projection

dist (Cont.)

Code example: Measuring the nearest distance

```
# Use the read_csv function from the Pandas library to read in raw latitude and
longitude coordinate pairs as Pandas dataframe and assign df_market to the
dataframe, then convert the dataframe to spatial data using the df_to_gdf
function from GPS2space. Notice in this file, POINT_X represents the longitude
and should be passed to x and POINT_Y represents the latitude and should be
passed to y.
df_market = pd.read_csv('./data/Colorado_Supermarkets_OSM.csv')
gdf_market = geodf.df_to_gdf(df_market, x='POINT_X', y='POINT_Y')

# Calculate the nearest distance from twinXa_512 to supermarket using the
dist_to_point function from GPS2space. distance. The first parameter (in this
case, gdf_twinXa_512) is the origin while the second parameter (in this case,
gdf_market) is the destination. proj is the EPSG identifier which should be
specified accordingly, depending on the research area.
dist = dist.dist_to_point(gdf_twinXa_512, gdf_market, proj=2163)
```

Call GPS2space from R

```
```{r}
library(reticulate)
conda_install("gps2space")
```
```

```
```{python}
# import modules
from gps2space import geodf
from gps2space import space
import pandas as pd

# read data
df = pd.read_csv("C:/Users/yanli/Box/GPS project/Python tutorial paper/example.csv")

# convert data to GeoDataFrame
gdf = geodf.df_to_gdf(df, x='longitude', y='latitude')

# calculate activity space
buff_space = space.buffer_space(gdf, dist=1000, dissolve="week", proj=2163)
```
```

Research questions

- Whether there were ***seasonal*** effects in twins' activity space/shared space;
- Whether there were ***weekend*** effects in twins' activity space/shared space;
- How activity space/shared space changed with ***ages***;
- Whether there were ***between-individual differences*** in the levels and growth rates of activity space/shared space; if so, how ***gender*** might influence such differences;
- Whether there were ***between-family differences*** in the levels and growth rates of shared space; if so, how ***zygosity*** might influence such differences.

Growth curve modeling

Level-1 model:

$$AS_{itk} = \beta_{0ik} + \beta_{1ik}Age_{itk} + \beta_2 Weekend_t + \beta_3 Summer_t + \beta_4 Fall_t + \beta_5 Winter_t + e_{itk}$$

seasonal and weekend effects

Level-2 model:

$$\beta_{0ik} = \gamma_{00k} + \gamma_{01k}Female_{ik} + u_{0ik}$$

$$\beta_{1ik} = \gamma_{10k} + \gamma_{11k}Female_{ik} + u_{1ik}$$

Between-Individual differences

Level-3 model:

$$\gamma_{00k} = \delta_{000} + v_{0k}$$

$$\gamma_{01k} = \delta_{010} + v_{1k}$$

$$\gamma_{10k} = \delta_{100} + v_{2k}$$

$$\gamma_{11k} = \delta_{110} + v_{3k}$$

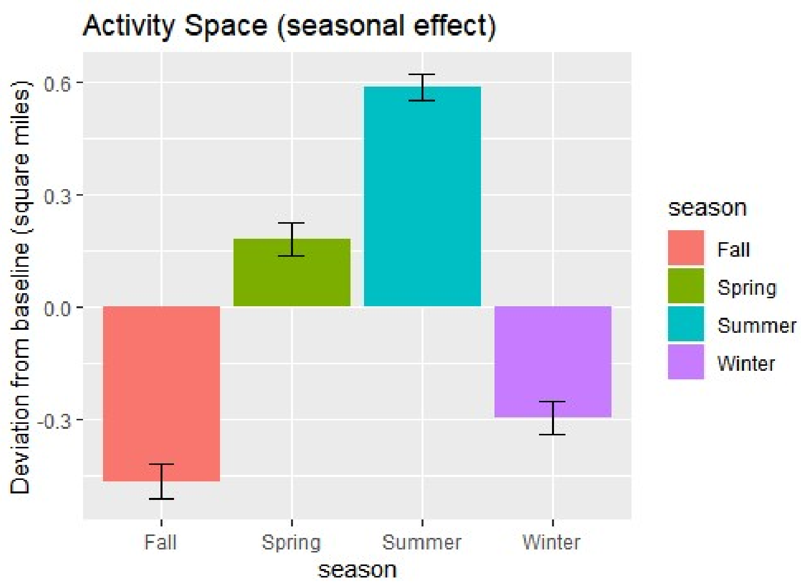
Between-family differences

$$e_{itk} \sim N(0, \sigma^2),$$

$$[u_{0ik},\ u_{1ik}]^T \sim MN\left(\mathbf{0}, \mathbf{T} = \begin{bmatrix} \tau_0^2 & \tau_{01} \\ \tau_{01} & \tau_1^2 \end{bmatrix}\right),$$

$$[v_{0k},\ v_{1k},\ v_{2k},\ v_{3k}]^T \sim MN\left(\mathbf{0}, \mathbf{\Phi} = \begin{bmatrix} \varphi_0^2 & \varphi_{01} & \varphi_{02} & \varphi_{03} \\ \varphi_{01} & \varphi_1^2 & \varphi_{12} & \varphi_{13} \\ \varphi_{02} & \varphi_{12} & \varphi_2^2 & \varphi_{23} \\ \varphi_{03} & \varphi_{13} & \varphi_{23} & \varphi_3^2 \end{bmatrix}\right)$$

Results – Activity space



We also found between-individual and between-family differences in the initial level of activity space.

Generalized growth curve modeling

- The shared space was defined as the ***proportion*** of the participant's activity space which overlapped with his/her twin sibling's activity space.
- We chose to use ***beta regression*** in this scenario.
- Beta distribution:

$$f(y|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1}(1-y)^{\beta-1}, 0 < y < 1$$

Let $\mu = \alpha/(\alpha+\beta)$ and $\phi = \alpha + \beta$, then the mean and variance are equal to

$$E(y) = \mu$$

$$Var(y) = \frac{\mu(1-\mu)}{1+\phi}$$

- Model the mean of shared space (i.e., μ)

Generalized growth curve modeling (Cont.)

Level-1 model:

Logit transformation

$$\eta_{itk} = \log\left(\frac{\mu_{itk}}{1-\mu_{itk}}\right) = \beta_{0ik}+\beta_{1ik}Age_{itk} + \beta_2 Weekend_t + \beta_3 Summer_t + \beta_4 Fall_t + \beta_5 Winter_t$$

Level-2 model:

$$\beta_{0ik} = \gamma_{00k} + \gamma_{01k}Female_{ik} + u_{0ik}$$

$$\beta_{1ik} = \gamma_{10k} + \gamma_{11k}Female_{ik} + u_{1ik}$$

Level-3 model:

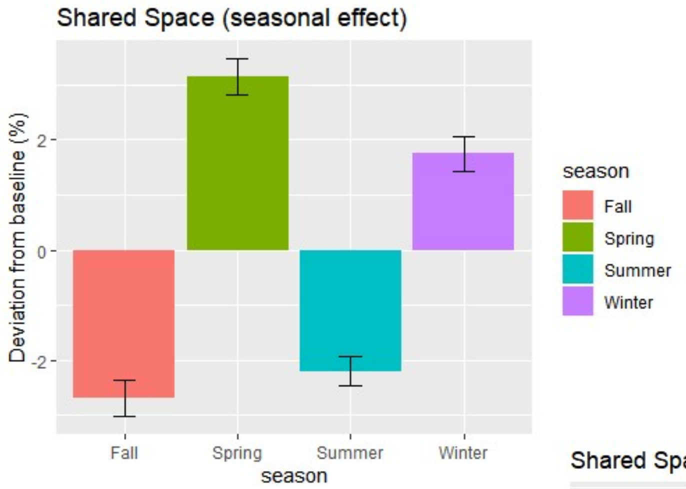
$$\gamma_{00k} = \delta_{000} + \delta_{001}MZ_k + v_{0k}$$

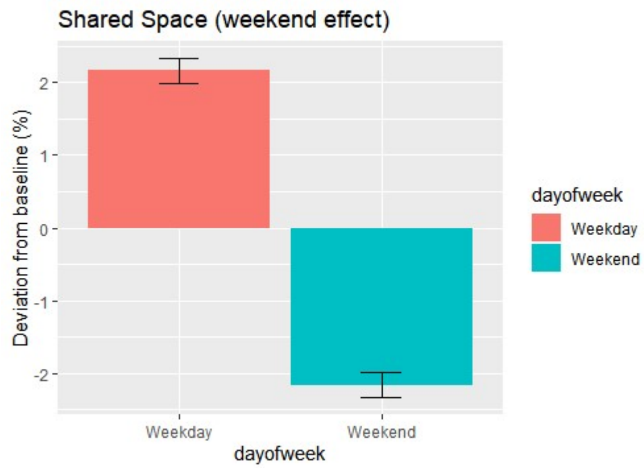
$$\gamma_{01k} = \delta_{010} + \delta_{011}MZ_k + v_{1k}$$

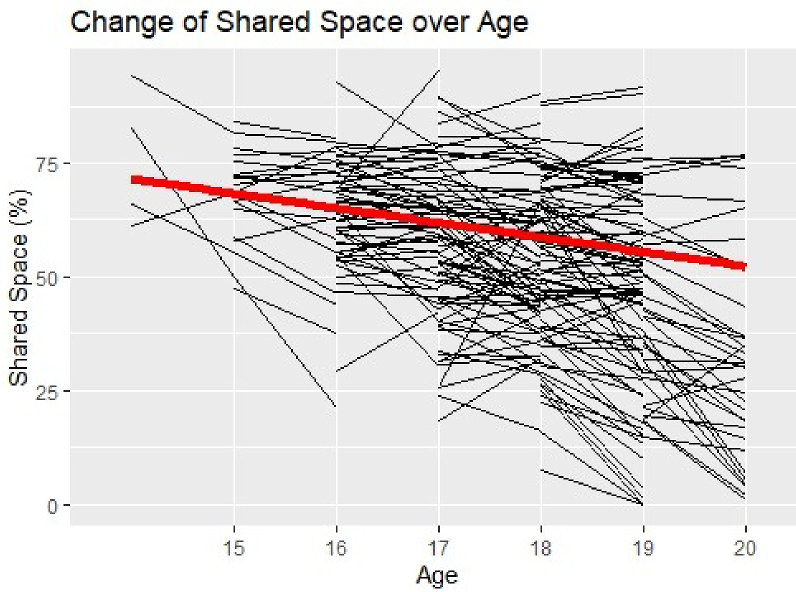
$$\gamma_{10k} = \delta_{100} + \delta_{101}MZ_k + v_{2k}$$

$$\gamma_{11k} = \delta_{110} + \delta_{111}MZ_k + v_{3k}$$

Results – Shared space







We also found between-individual and between-family differences in the initial level of shared space.

Take-home Messages

- GPS2space provides open-source solutions to building spatial data, extracting spatial measures, and conducting spatial query
- GPS2space incorporates cKDTree technology to dramatically increase the speed of nearest distance query
- An application of GPS2space was conducted, finding different patterns of seasonal effects in shaping activity space and shared space, and age effect in determining shared space in the CoTwins study

Potential application

GPS2space can also be used in social mobility, health studies, and many other areas that rely on GPS data or geo-tagged location data.

Future development

- Include concave hull, hexagon, and rectangle methods in extracting spatial measures
- Provide parameterization for users to specify the column names of their desired spatial measures
- Incorporate functions for the nearest distance query among point/multi-point, line/multi-line, polygon/multi-polygon spatial features
- Other spatial measures such as travel distance and proximity measure between twins/individuals

Thanks.

Documentation of GPS2space:

https://gps2space.readthedocs.io/en/latest/ or Google “GPS2space”

Contact:
Shuai Zhou
sxz217@psu.edu