Space-Time

Michal Michalski

Table of contents

Prep

Data First

```
library(sf) # for spatial classes
```

```
Linking to GEOS 3.9.1, GDAL 3.4.0, PROJ 8.1.1; sf_use_s2() is TRUE
```

```
library(sfdep) # spatial dependence and ESDA
library(dplyr) # tidy data
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(stringr) # text
library(ggplot2) # plot data
library(h3jsr) # uber hexagons
```

Next, load data from TBS survey - Survey Extent and Archaeological Sites

```
# find out layer names
st_layers("../data/vect/data.gpkg")
```

```
Driver: GPKG
Available layers:
         layer_name     geometry_type features fields
1   tbs_sites_point             Point       88     11
2 tbs_sites_polygon     Multi Polygon      323      6
3 tbs_survey_extent     Multi Polygon        1      1
4   tbs_hollow_ways Multi Line String      568      4
5           tbs_aoi           Polygon        1      5
6          njs_bbox           Polygon        1      1
```

```
# survey boundary
survey_sf = st_read("../data/vect/data.gpkg", layer = "tbs_survey_extent") %>%
  st_transform(crs = 4326)
```

```
Reading layer `tbs_survey_extent' from data source
  `/Users/michalmichalski/Documents/survey/data/vect/data.gpkg'
  using driver `GPKG'
Simple feature collection with 1 feature and 1 field
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 628692.6 ymin: 4053955 xmax: 654692.6 ymax: 4079955
Projected CRS: WGS 84 / UTM zone 37N
```

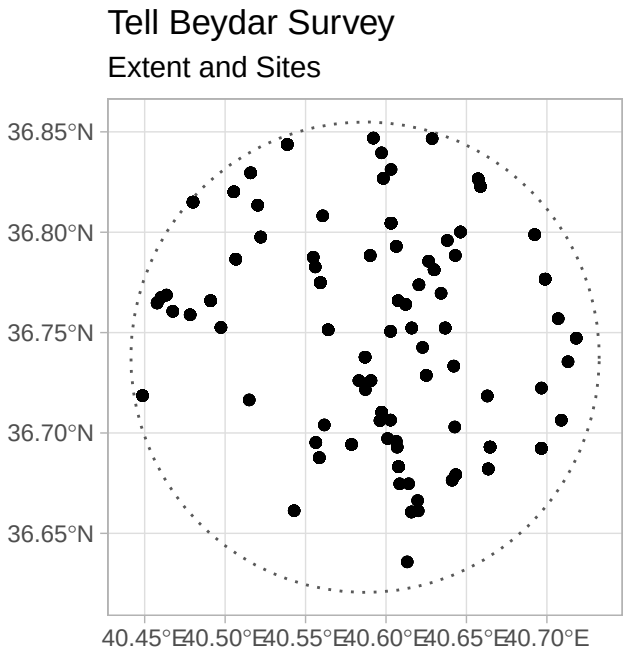
```
# sites
sites = readRDS("../data/vect/n_mes_sites_LONG.rds") %>%
```

```
  filter(str_detect(id,"TBS")) %>%
  select(id, cent_start, longitude, latitude) %>%
  mutate(century = as.numeric(cent_start)) %>%
  select(-cent_start)

# create sf class
sites_sf = st_as_sf(sites, coords = c("longitude", "latitude"), crs = st_crs(4326))
```

Plot the data

```
ggplot() +
  geom_sf(data = survey_sf, fill = NA, linetype = 21) +
  geom_sf(data = sites_sf) +
  ggtitle('Tell Beydar Survey', subtitle = 'Extent and Sites') +
  coord_sf() +
  theme_light()
```



Hexagons

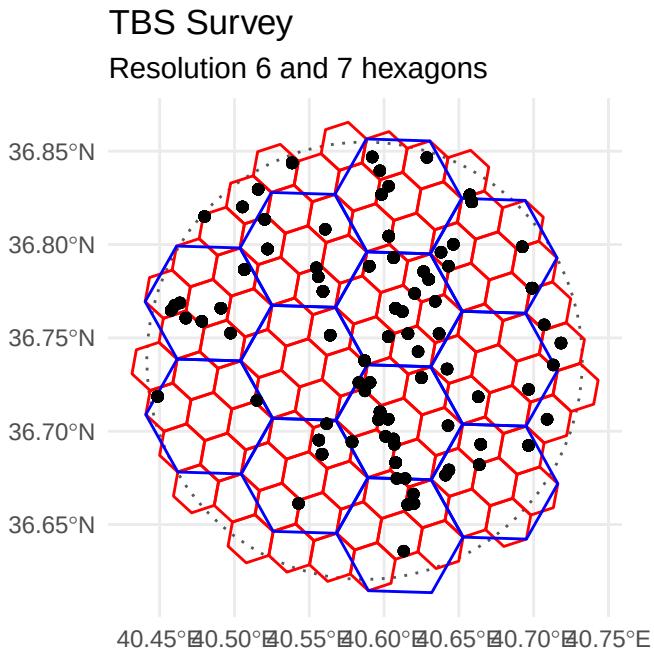
```
tbs_7 <- polyfill(survey_sf, res = 7, simple = FALSE)

tbs_7 <- h3_to_polygon(unlist(tbs_7$h3_polyfillers), simple = FALSE)

tbs_6 <- polyfill(survey_sf, res = 6, simple = FALSE)

tbs_6 <- h3_to_polygon(unlist(tbs_6$h3_polyfillers), simple = FALSE)

ggplot() +
  geom_sf(data = survey_sf, fill = NA, linetype = 21) +
  geom_sf(data = tbs_7, fill = NA, colour = 'red') +
  geom_sf(data = tbs_6, fill = NA, colour = 'blue') +
  geom_sf(data = sites_sf) +
  ggtitle('TBS Survey', subtitle = 'Resolution 6 and 7 hexagons') +
  theme_minimal() +
  coord_sf()
```



Point - in - Polygon: join unique hexagons identifier to points

```
head(tbs_6)
```

```
Simple feature collection with 6 features and 2 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 40.44076 ymin: 36.61354 xmax: 40.71615 ymax: 36.85655
Geodetic CRS:  WGS 84
       h3_address h3_resolution                       geometry
1 862d8da17ffffff             6 POLYGON ((40.5882 36.85655,...
2 862d8da97ffffff             6 POLYGON ((40.65256 36.70376...
3 862d8db97ffffff             6 POLYGON ((40.46174 36.73866...
4 862d8daafffffff             6 POLYGON ((40.65196 36.82471...
5 862d8d14fffffff             6 POLYGON ((40.58923 36.67509...
6 862d8da87ffffff             6 POLYGON ((40.65226 36.76424...
```

```
head(sites_sf)
```

```
Simple feature collection with 6 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 40.58699 ymin: 36.73775 xmax: 40.58699 ymax: 36.73775
Geodetic CRS:  WGS 84
# A tibble: 6 x 3
  id        century            geometry
  <chr>       <dbl>         <POINT [°]>
1 TBS_1_0_0    -600 (40.58699 36.73775)
2 TBS_1_0_0    -500 (40.58699 36.73775)
3 TBS_1_0_0    -400 (40.58699 36.73775)
4 TBS_1_0_0    -900 (40.58699 36.73775)
5 TBS_1_0_0    -800 (40.58699 36.73775)
6 TBS_1_0_0    -700 (40.58699 36.73775)
```

```
sites_sf = st_join(sites_sf, tbs_7)

sites_sf = sites_sf %>% select(-h3_resolution)

tbs_df = sites_sf %>%
  st_drop_geometry()
```

```r
tbs_df = tbs_df %>%
  dplyr::as_tibble() %>%
  filter(!is.na(h3_address)) %>%
  count(century, h3_address, .drop = FALSE) %>%
  tidyr::complete(century, h3_address, fill = list(n = 0))
```

Prepare Hexagon geometry

```r
head(tbs_7)
```

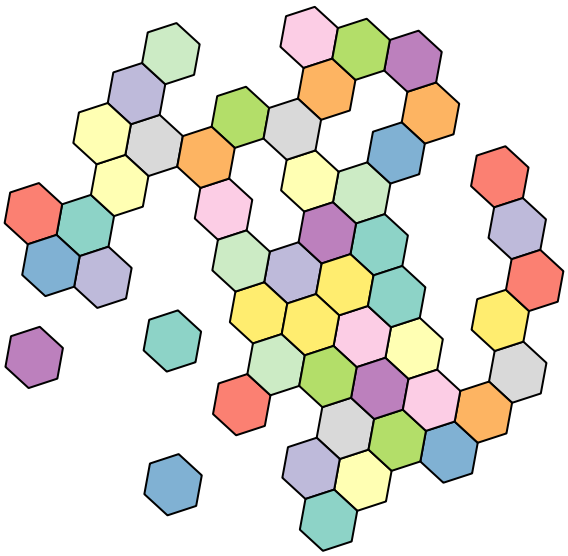
```
Simple feature collection with 6 features and 2 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 40.54025 ymin: 36.61457 xmax: 40.71564 ymax: 36.83365
Geodetic CRS:  WGS 84
       h3_address h3_resolution                       geometry
1 872d8d169ffffff             7 POLYGON ((40.55278 36.70191...
2 872d8daabffffff             7 POLYGON ((40.6884 36.79787,...
3 872d8d14effffff             7 POLYGON ((40.58038 36.63641...
4 872d8db8affffff             7 POLYGON ((40.5432 36.78424,...
5 872d8da90ffffff             7 POLYGON ((40.66173 36.68193...
6 872d8da13ffffff             7 POLYGON ((40.62469 36.82972...
```

```r
tbs_geo = tbs_7 %>% select(-h3_resolution)

tbs_geo_subset = semi_join(tbs_geo,tbs_df)
```

```
Joining, by = "h3_address"
```

```r
plot(tbs_geo_subset)
```




```
tbs_spt <- spacetime(tbs_df,
                     tbs_geo_subset,
                     "h3_address",
                     "century")

is_spacetime(tbs_spt)
```

```
[1] TRUE
```

```
tbs_spt = complete_spacetime_cube(tbs_spt)

tbs_spt
```

```
spacetime

Context:`data`

51 locations `h3_address`
```

```
87 time periods `century`

-- data context --------------------------------------------------------------

# A tibble: 4,437 x 3
   century h3_address          n
 *   <dbl> <chr>           <int>
 1   -6800 872d8d148ffffff     0
 2   -6800 872d8d149ffffff     0
 3   -6800 872d8d14dffffff     0
 4   -6800 872d8d169ffffff     0
 5   -6800 872d8d16effffff     0
 6   -6800 872d8da10ffffff     0
 7   -6800 872d8da11ffffff     0
 8   -6800 872d8da15ffffff     0
 9   -6800 872d8da16ffffff     0
10   -6800 872d8da1effffff     0
# ... with 4,427 more rows
# i Use `print(n = ...)` to see more rows
```

Emerging Hot Spots Analysis

```
# conduct EHSA
ehsa = emerging_hotspot_analysis(tbs_spt, "n")
```

have a look at results

```
 head(ehsa)
```

```
# A tibble: 6 x 4
  location            tau p_value classification
  <chr>             <dbl>   <dbl> <chr>
1 872d8d169ffffff  0.0475 0.523   no pattern detected
2 872d8daabffffff  0.239  0.00124 sporadic hotspot
3 872d8db8affffff  0.207  0.00504 no pattern detected
4 872d8da90ffffff  0.185  0.0121  sporadic hotspot
5 872d8daa6ffffff -0.196  0.00822 no pattern detected
6 872d8d149ffffff -0.165  0.0250  sporadic hotspot
```

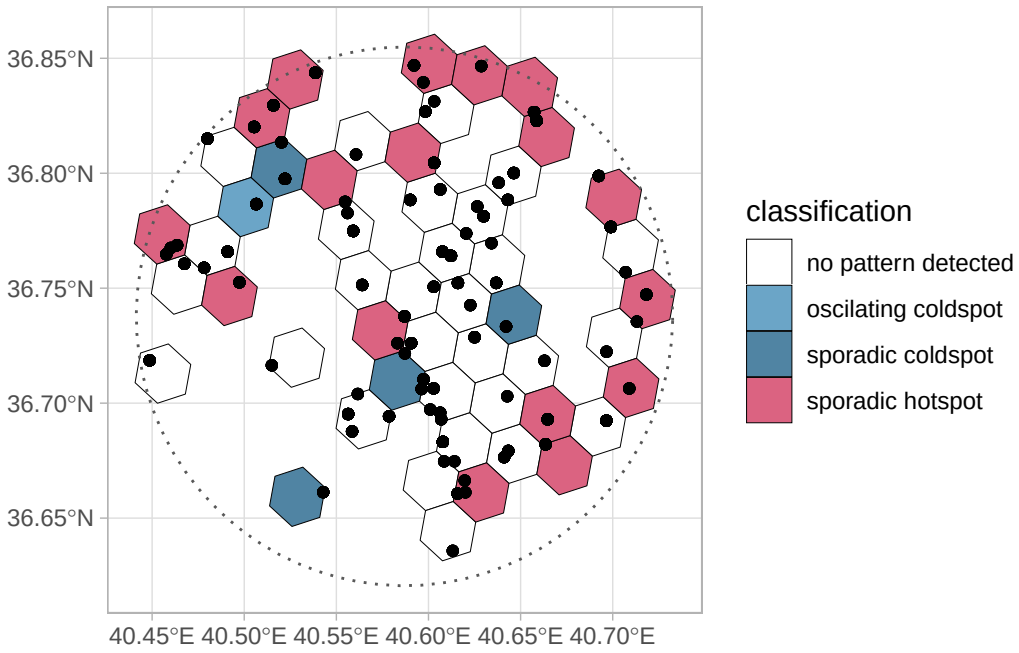
classification

```
unique(ehsa$classification)
```

```
[1] "no pattern detected" "sporadic hotspot"    "sporadic coldspot"
[4] "oscilating coldspot"
```

plot results

```
dplyr::left_join(ehsa, tbs_geo_subset, by = c("location" = "h3_address")) |>
  sf::st_as_sf() |>
  ggplot() +
  geom_sf(aes(fill = classification), lwd = 0.1, color = "black") +
  geom_sf(data = survey_sf, fill = NA, linetype = 21) +
  geom_sf(data = sites_sf) +
  # specify colors
  scale_fill_manual(
    values = c(
      "no pattern detected" = "white",
      #"new coldspot" = "#84d6e3",
      "oscilating coldspot" = "#6ba5c7",
      "sporadic coldspot" = "#5084a3",
      #"consecutive coldspot" = "#386682",
      "sporadic hotspot" = "#db6381"
      #"consecutive hotspot" = "#a3485f"
      )
  ) +
  theme_light()
```


36.85°N
36.80°N
36.75°N
36.70°N
36.65°N
40.45°E 40.50°E 40.55°E 40.60°E 40.65°E 40.70°E
classification
no pattern detected
oscilating coldspot
sporadic coldspot
sporadic hotspot