Transport networks and road safety

Andrea Gilardi [1] Robin Lovelace [2]

[1]University of Milan - Bicocca

[2]University of Leeds - ITS

Who am I

Overview of the seminar

1. A

A few definitions

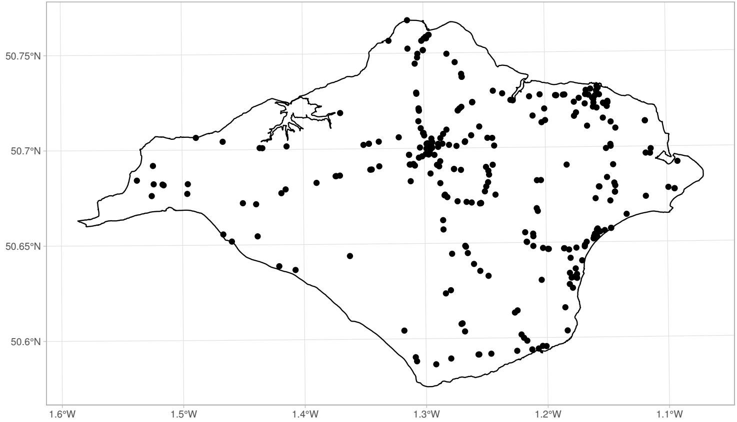
- In a super informal way we can say that a **Point Process** is a random mechanism whose outcomes are **Point Patterns**, i.e. a (finite) sequence of points in the space.
- Classical examples of point processes are: tree locations in a forest (the classic swedish pines data), animal nesting sites, ambulance interventions or, as in this seminar, car crashes.
- We will use these data to formalize the first steps we took towards the definition of a precise model that can be used to locate the most dangerous locations for car crashes (i.e. the black spots).

Car crashes data

- In the following part of this seminar, we will analyze data for car crashes that occurred in the Isle of Wight (UK) during 2018.
- We downloaded the data using the `stats19` package, which is a tool to help download, process and analyse the UK road collision data collected using the 'STATS19' form.
- These data are really rich and they include several additional information (like the severity of the crash, the weather, the light condition and several other markers) but, for the moment, we will focus only on the location of the events.

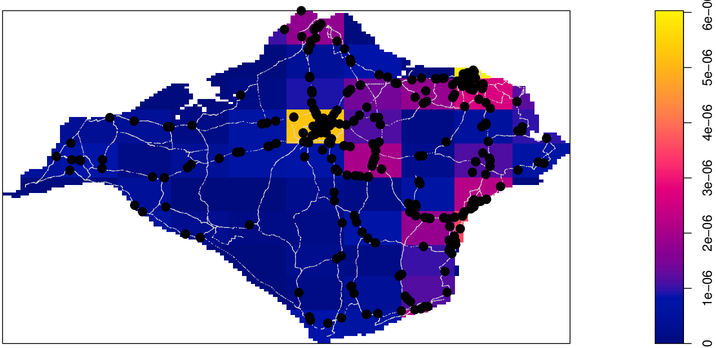
Car crashes data (cont)

This is a graphical representation of the car crashes occurred in the Isle of Wight (UK) during 2018. There are some clear patterns in the data that we need to take into account.



Point Processes on a Street Network

Car crashes represent a classical example of a point process occurring on a linear network and the usual statistical techniques (as the following quadratcount) are not valid.

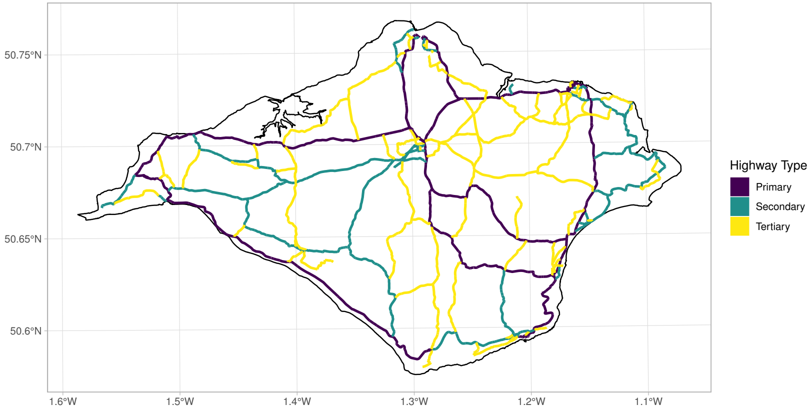


Street Networks

- The road network we use is built using OpenStreetMap data.
- OpenStreetMap is a project that aims at building a free and editable map of the World with an open-content license.
- The basic components of OpenStreetMap data are called *elements* and they consist of:
  - *nodes*: representing points on the earth surface;
  - *ways*: which is an ordered list of nodes;
  - *relations*: which is a list of nodes, ways and other relations. Each member has additional information that describe its relationship with the other elements. Roads, turn restrictions and administrative boundaries are usually described as relations.

Street Network in the Isle of Wight

This is a graphical representation of the main roads in the Isle of Wight.



`stplanr` - **networks**

- Broadly speaking, let's say that a street network is a network whose nodes and edges are associated with geographical elements in the space.
- In the `stplanr` representation of a street network, the edges are the ways that were download from OSM while the vertexes are the starting and ending node of each way.
- This representation implies that two or more edges are *connected* if and only if they share one or more boundary point.

Problems...

Theories and definitions are fine but obviously the data we face in the wild world is quite different. We will discuss three probems: **roundabouts** (i.e. circular ways), **overpasses** (i.e. intersecting ways that are not really connected due to a vertical grade of separation) and (some) **street intersections**.

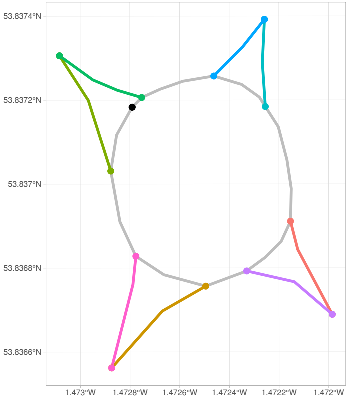
Theory

Real Data

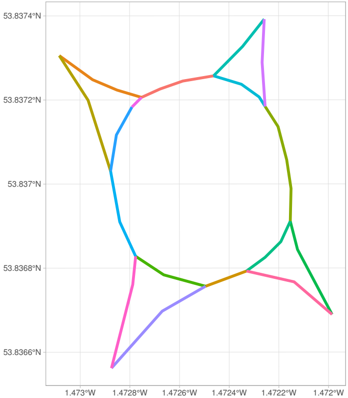
Roundabouts, i.e. circular ways

The roundabout on the left is unroutable by `stplanr`-definition of spatial network since the roundabout is not connected to the other edges.

Before



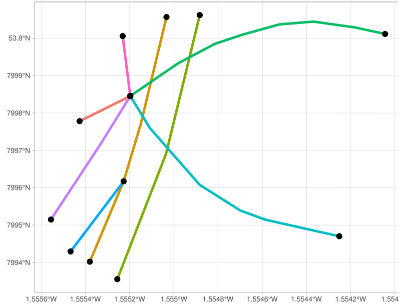
After



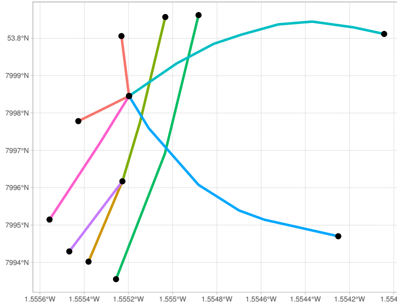
Bridges, overpasses and underpasses

Even if we break up a street network (unroutable on the left) we must be sure not to ruin overpasses and underpasses relations.

Before



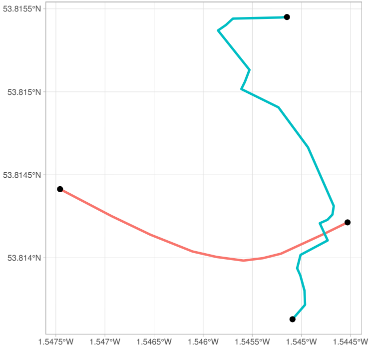
After



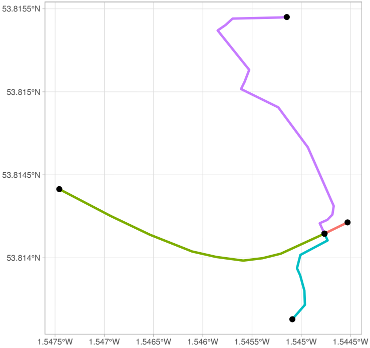
Streets intersections

There are also some cases where two streets intersects and they don't share any vertex.

Before

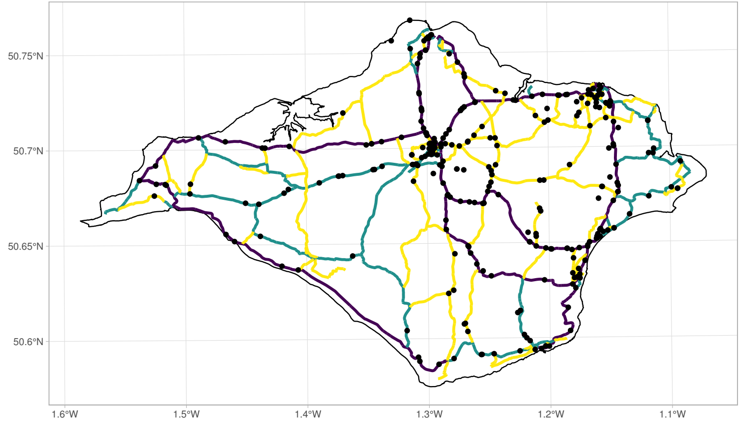


After



Fixing the street networks

We developed a new function (called `rnet_breakup_vertices` and stored in `stplanr` package) to fix all these problems and this is the result.



Modeling of crashes on street networks

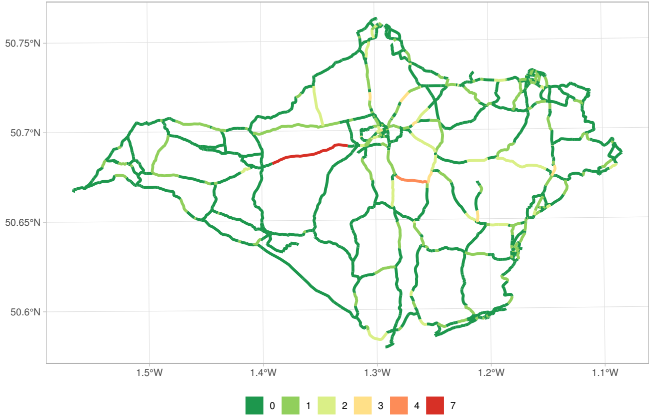
Summary of the models

Counting crashes on street networks

- Following the same ideas behind the quadratcounts on the plane (and some statistical theories) we wrote a few function to count the number of car crashes occurring on each edge of the street network.
- R can only represent exactly integers numbers and fractions whose denominator is a power of two (source), so none of the car crashes (whose coordinates are represented as double numbers with a 53 binary digits accuracy) lie exactly on the street network.
- For that reason we wrote a few R function to match each car crash with the nearest edge on the network and count the occurrences.

The first road risk measure

This is the result. You should note that we excluded all car crashes that were farther than 100m from the nearest network edge (33 crashes).

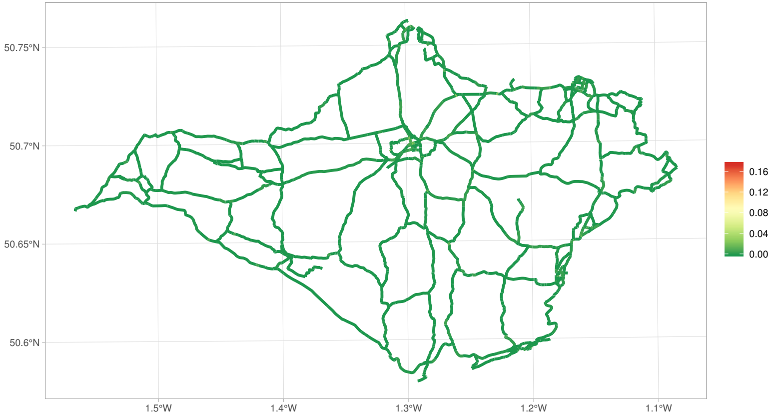


Problems with the raw counts measure

- There are some clear problems with the previous risk measure and, the most important one, is that we are comparing ways with very different length (which is what we are going to call *exposure*).
- There are two possible solutions: 1) rebuild the network cutting and pasting ways in such a way that they all have approximately the same length; 2) estimate the number of car crashes per meter.
- For the moment we are working with the second solution since the other one is much more difficult to implement.

Car crashes per meter

This is the result and, then again, there are some obvious problems: there are a few car crashes occurring in very small road segments that artificially inflate this risk index.



Local smoothing on a linear network

- Let n be the number of edges in the street network, $x_i,\ i = 1, \ldots, n$ the number of car crashes occurring in each edge and $l_i,\ i = 1 \ldots n$ the length of each way. The ratio

$$y_i = \frac{x_i}{l_i},\ i = 1, \ldots, n$$

represent the number of car crashes per meter.

- Let $\delta_{i,p}$ represent the set of all neighbours of each street i up to order p. If $p = 0$ then $\delta_{i,p}$ includes only the i-th street segment; if $p = 1$ then $\delta_{i,1}$ includes the street segment i and its neighbours[1]; if $p = 2$ then $\delta_{i,2}$ includes the street segment, its neighbours and the neighbours of its neighbours and so on...

[1] In the `stplanr` representation of street networks, two ways are neighbours if they share one boundary point

Local smoothing on a linear network

- Now we can perform a local smoothing of y_i, i.e. the number of car crashes per meter, as

$$\tilde{y}_i = \frac{1}{m_i} \sum_{j \in \delta_{i,p}} y_j$$

  where m_i represent the number of elements included in $\delta_{i,p}$.

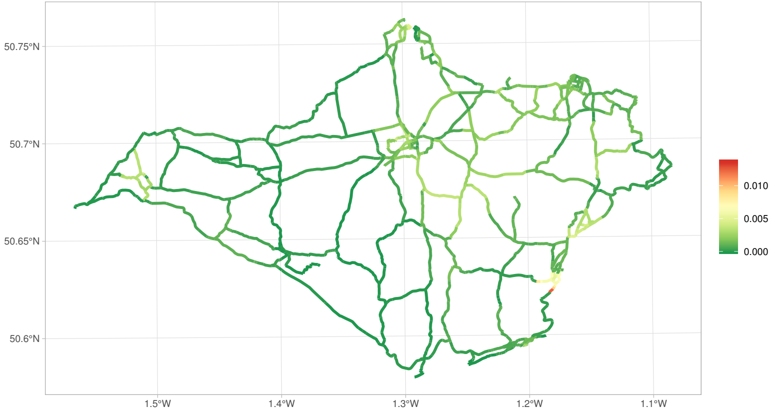
- The value of p is an input for the procedure and represent the degree of the smoothing: small values represent a local smoothing while higher values create a "global" smoothing. For example if we take $p = n$ then every segment is linked with the same value, i.e. $\tilde{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \ \forall i = 1, \ldots, n$.

- We'll see later a few methods that can be used to decide the value of p.

- The procedure was coded using the `igraph` package.

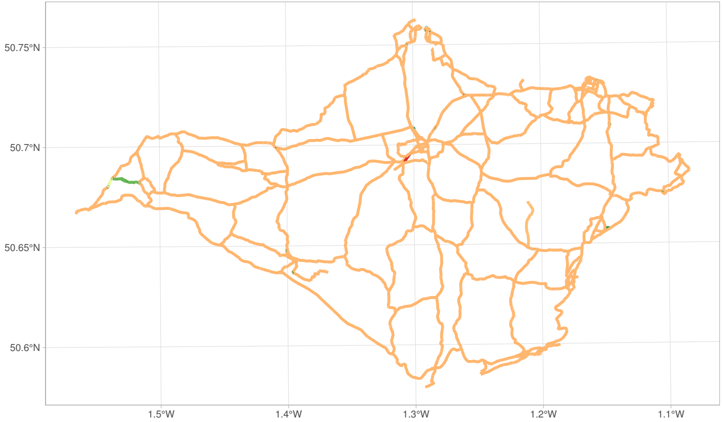
Smoothing on a linear network

This is the result with $p = 6$. Obviously this procedure has several drawbacks but at least now it's possible to extract some information from that plot.



Why network cleaning is important?

This is the same procedure applied to the street network without the cleaning and with $p = n$. It's clear that there is something wrong.



TODO

- Add citations and fix bibliography
- Add credits to memes, OSM and beamer theme
- highlight important word (i.e. ABC)