Wild Boar observations in Ardèche

Facundo Muñoz
facundo.munoz@cirad.fr
famuvie
cirad

ASTRE
AnimalS health
Territories Risks Ecosystems

Biostatistics for epidemiology
December 2018

Packages

```r
pacman::p_load(
  DCluster,   # cluster detection
  fasterize,  # high-performance rasterisation of sf polygons
  gganimate,  # animated plots
  gridExtra,  # arragement of multiple plots
  leaflet,    # interactive mapping
  tidyverse,  # data manipulation, plotting
  raster,     # get geographic data, raster handling
  rasterVis,  # raster visualisation tools
  sf,         # Simple Features
  spatstat,   # Statistical analysis of spatial point patterns
  tmap        # map plotting
)
```

Load dataset

```
wb <- readRDS("data/wb_ardeche.rds")
ardeche <- readRDS("data/municip_ardeche.rds")
wb_env <- readRDS("data/wb_env.rds")
```

Inspect structure

Skim summary statistics
n obs: 195
n variables: 5

Variable type: character

variable	missing	complete	n	min	max	empty	n_unique
geometry	0	195	195	35	37	0	152
municipality	0	195	195	4	29	0	71

Variable type: integer

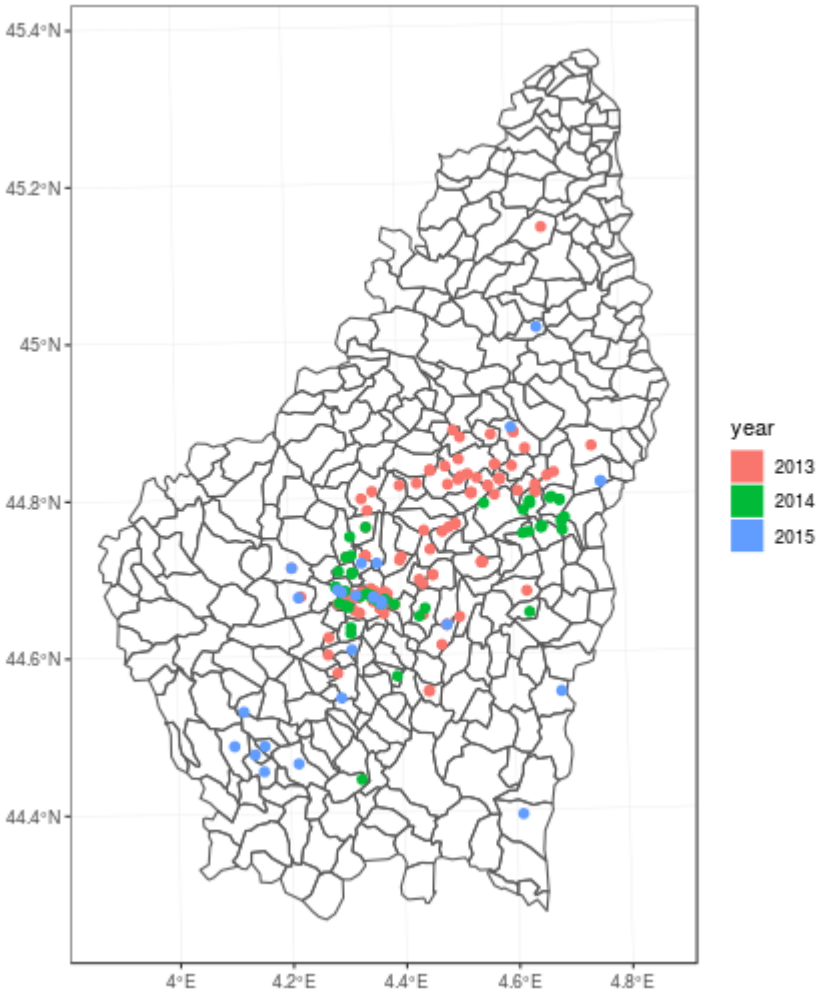
variable	missing	complete	n	mean	sd	p0	p25	p50	p75	p100	hist
week	8	187	195	33.68	7.02	0	30	32	38	53	
year	0	195	195	2013.66	0.89	2013	2013	2013	2014	2017	

Variable type: POSIXct

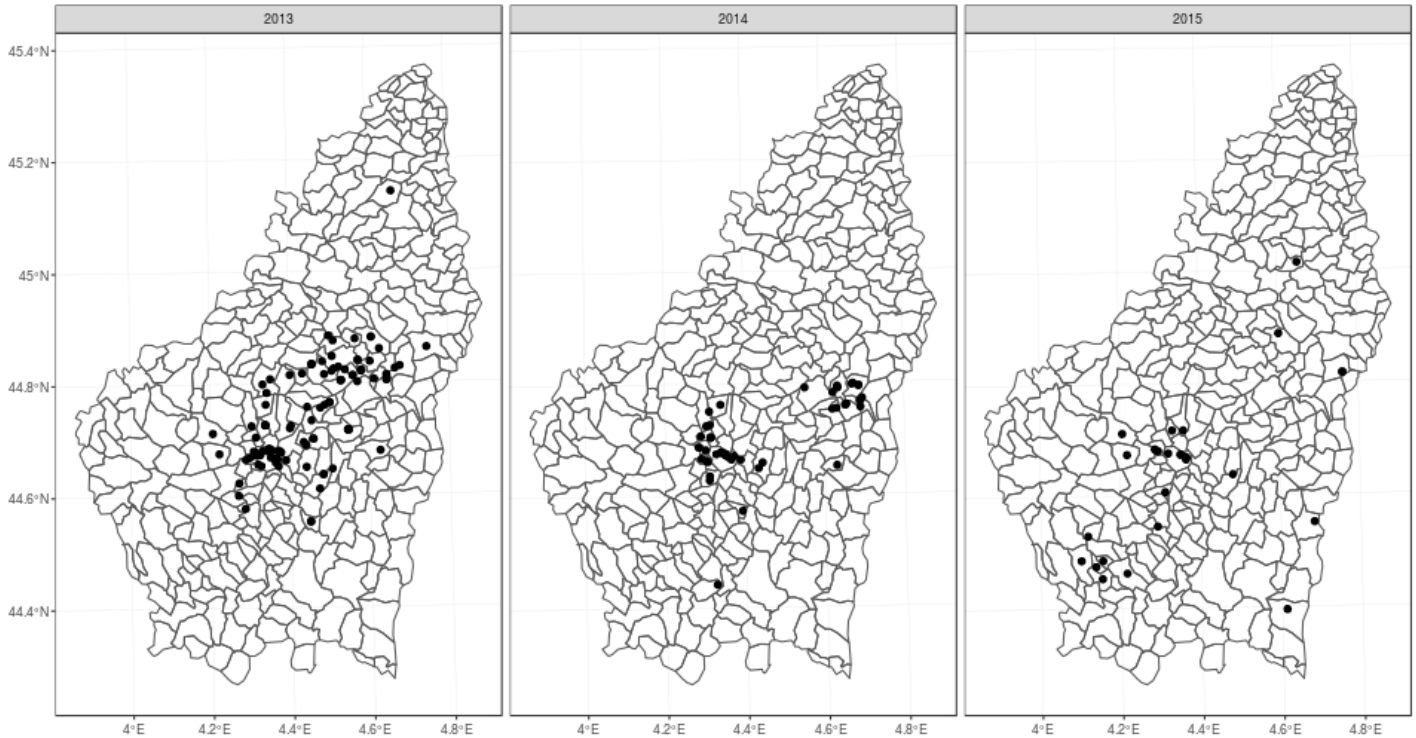
variable	missing	complete	n	min	max	median	n_unique
date	3	192	195	2012-11-15	2017-07-27	2013-10-13	130

Spatial representation

Visualise


45.4°N
45.2°N
45°N
44.8°N
44.6°N
44.4°N
4°E
4.2°E
4.4°E
4.6°E
4.8°E
year
2013
2014
2015

Visualise by year



Do you think the observations are

clustered?

Clustering assessment

Excercise

Produce a density estimate of the observations locations

Yeah... but

can't this be just **chance**?

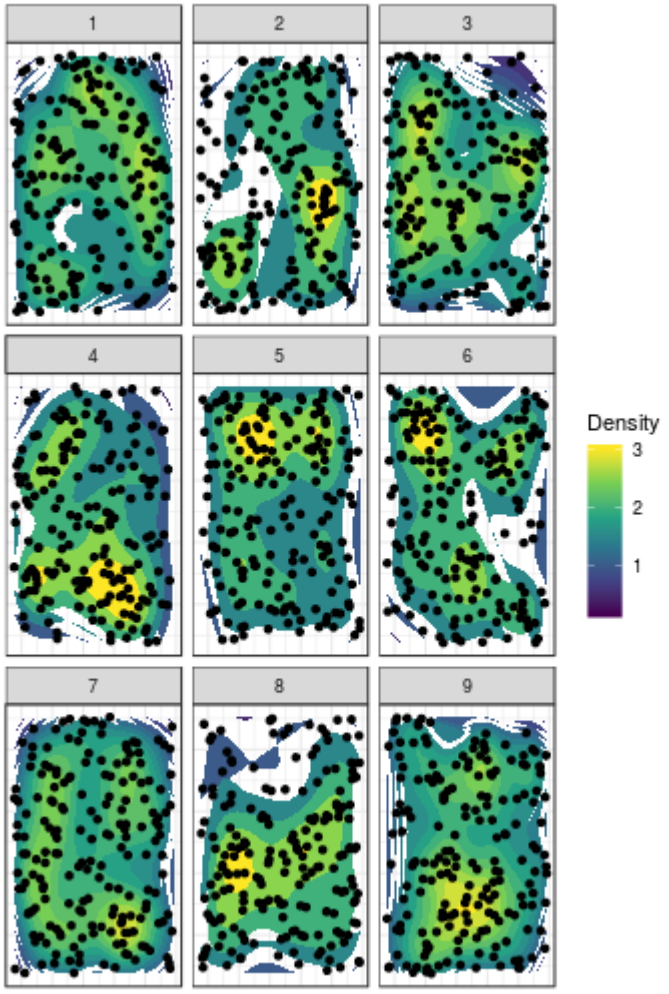
Yeah... but

can't this be just **chance**?

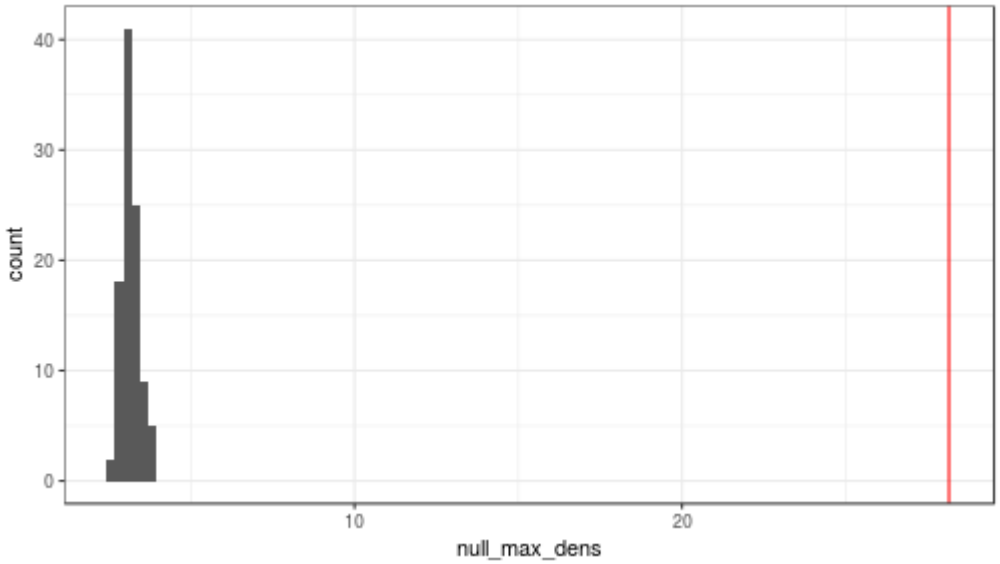
Yeah... but

can't this be just **chance**?

We can simulate many times assuming *complete spatial randomness* (CSR)...



- summarise each realisation with a number (maximum density, here)
- to compare realisations vs observed value:



Summary of hypothesis testing

- **Assume** a (null) hypothesis to be tested
- **Summarise** a realisation of the data with some number (test statistic)
- **Sampling distribution** of the statistic under the null (e.g. MC)
- **Compare** with observed statistic (p-value)

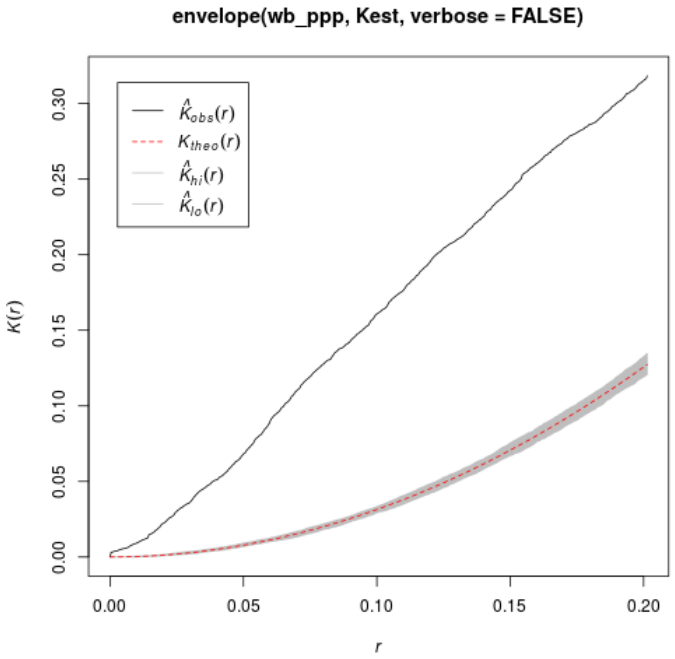
Ripley's K function

- A classical measure (and test) of spatial clustering for point patterns
- **Null**: independent, uniform distribution (as before)
- **Statistic**: Expected number of events within distance r

```r
## spatstat's class `ppp`
ardeche_border <- st_union(
  ardeche
)

ardeche_owin <- owin(
  poly = scale(
    st_coordinates(
      ardeche_border
    )[rev(seq_len(nrow(st_coordi
    center = FALSE, scale = attr
  )
)
wb_ppp <- ppp(
  wb_coord_sc[,1],
  wb_coord_sc[,2],
  window = ardeche_owin
)

# plot(wb_ppp)
plot(
  envelope(
    wb_ppp,
    Kest,
    verbose = FALSE
  )
)
```



Conclusions

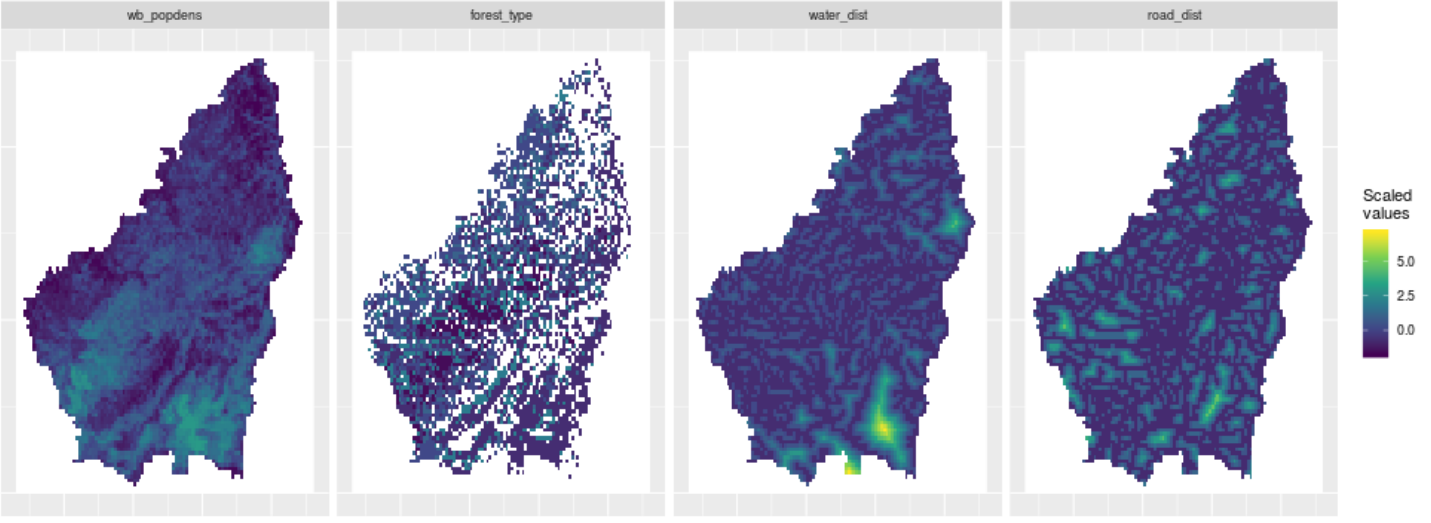
- Seems clear that the observations are **clustered** (as expected!)
- Now we need to find our own *pump*.

What can explain the patterns of diseased WB findings?

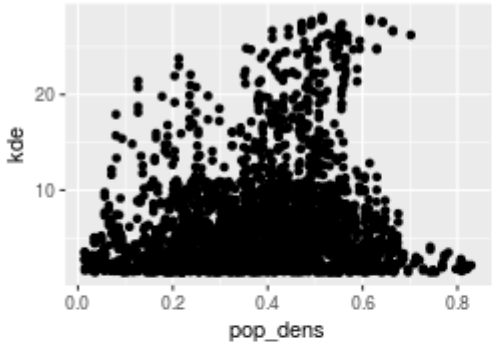
+

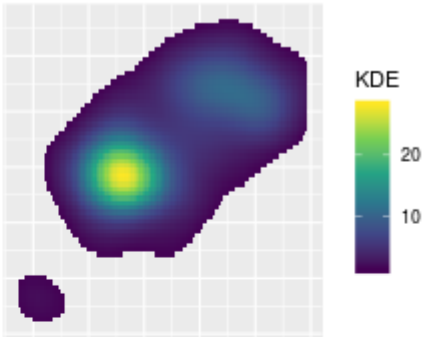
−

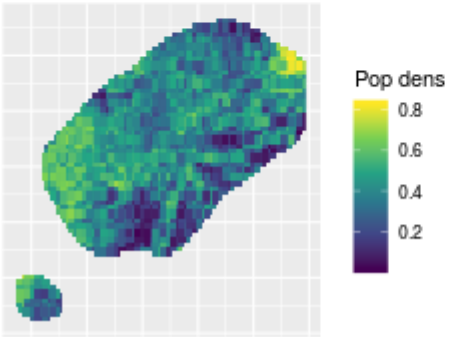
Potential risk factors



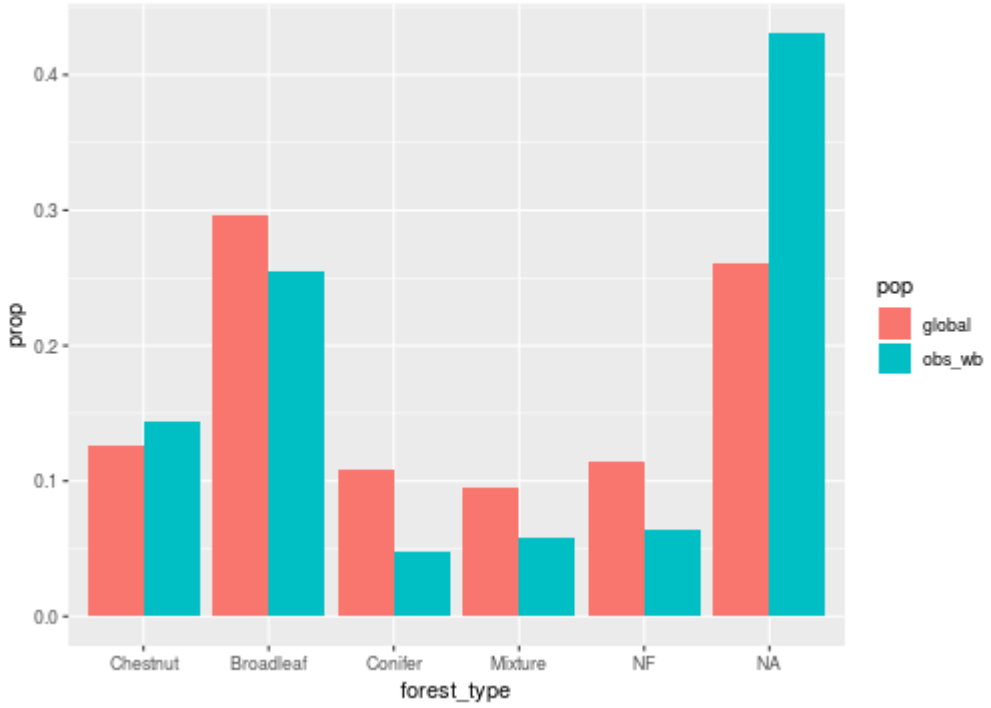
WB density





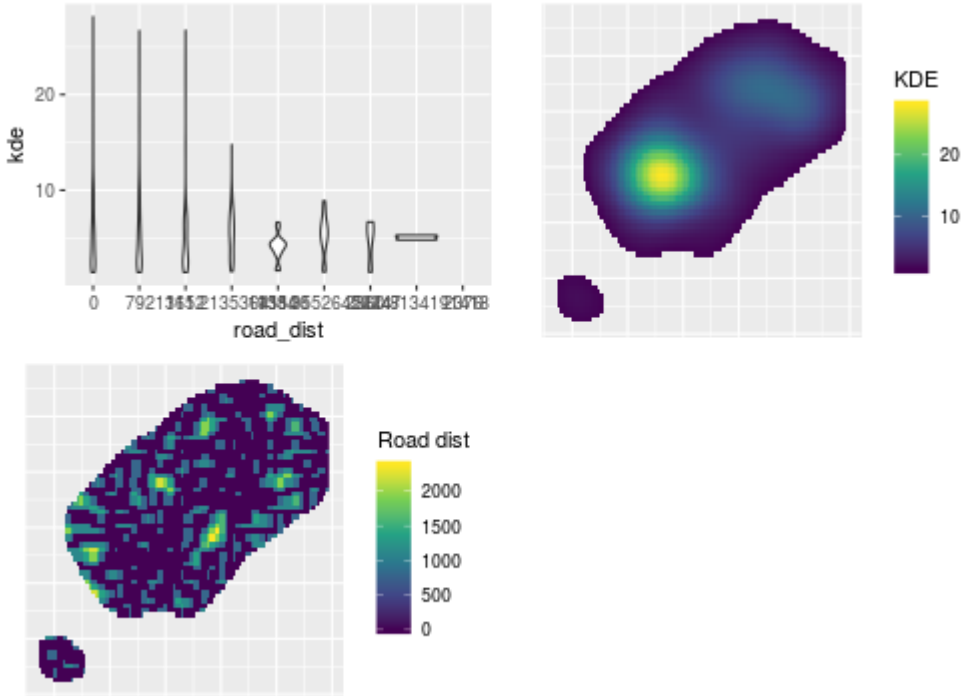


Forest type



Note that there is a lot of autocorrelation in *global*. Perhaps better to sample.

Road distance



Note that high kde is observed *only* near some road.

Further options as spatial point pattern

- Account for population (*control*) density: analyse the *estimated risk* surface $\rho(x) = \lambda_1(x)/\lambda_0(x)$
  - but this does not account for other covariates (e.g. road-dist)
- Model the intesity surface using covariates (`ppm`)
  - beyond the scope of the course
- Discretise and use `glm`s (e.g. Poisson regression)
  - requires large samples

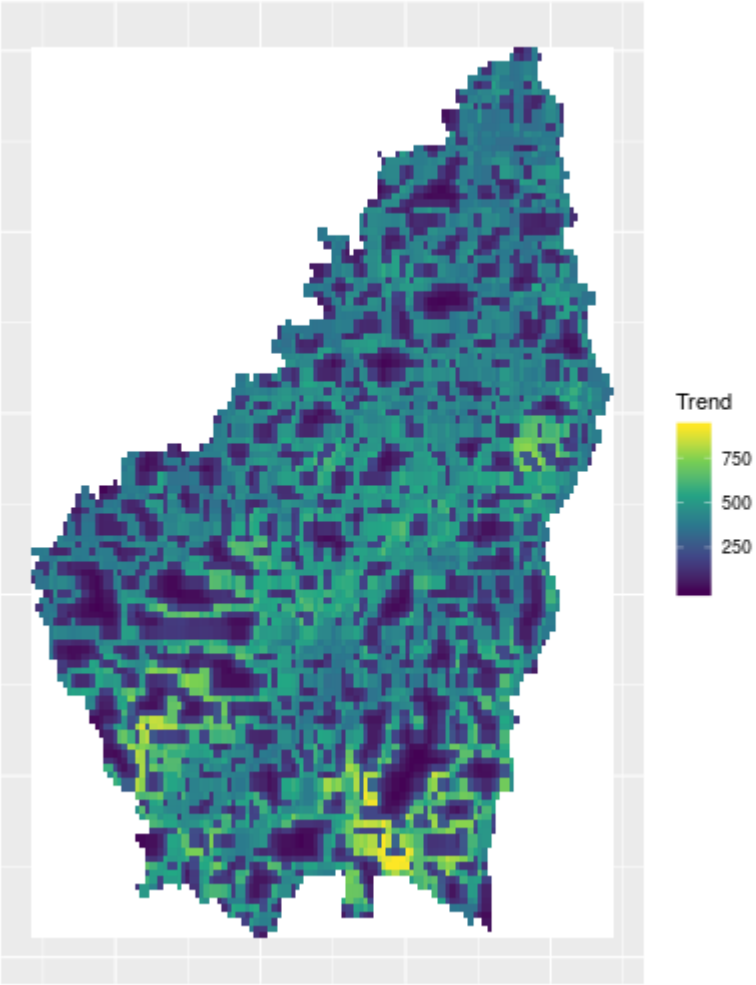
Glimpse on modelling

```
coord_env_sc <-
  scale(
    coordinates(wb_env),
    center = FALSE,
    scale = attr(wb_coord_sc, "scaled:scale")
  )

pop_dens_im <-
  im(
    mat = as.matrix(wb_env$wb_popdens)[115:1, ],
    xcol = sort(unique(coord_env_sc[, "x"])),
    yrow = sort(unique(coord_env_sc[, "y"]))
  )
# plot(pop_dens_im)

road_dist_im <-
  im(
    mat = as.matrix(wb_env$road_dist)[115:1, ],
    xcol = sort(unique(coord_env_sc[, "x"])),
    yrow = sort(unique(coord_env_sc[, "y"]))
  )
# plot(road_dist_im)
```

```
gplot(raster(predict(fm1, type = "trend"))) +
  geom_raster(aes(fill = value), na.rm = TRUE) +
  wb_plot("Trend")
```



```r
plot(envelope(fm1, Kest, nsim = 39))
```

```
## Generating 39 simulated realisations of fitted cluster model  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
##  39.
##
## Done.
```

Computation of *Risk*

$$Risk = Cases/(Cases + Controls)$$

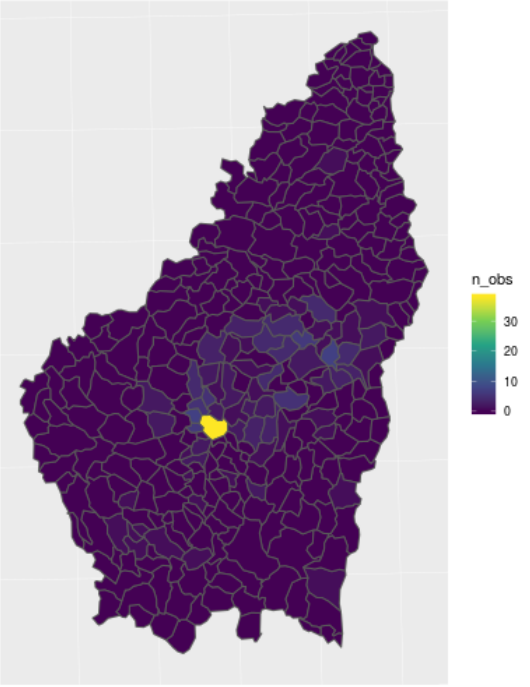
We don't really have **control** observation, but let's take the estimated *trend*.

Aggregated (areal) data

Counts by municipality

```
ardeche$n_obs <- lengths(
  st_covers(ardeche, wb)
)

ggplot() +
  geom_sf(
    data = ardeche,
    aes(fill = n_obs)
  ) +
  scale_fill_viridis_c() +
  labs(x = NULL, y = NULL) +
  theme(
    axis.text = element_blank(),
    axis.ticks = element_blank()
  )
```

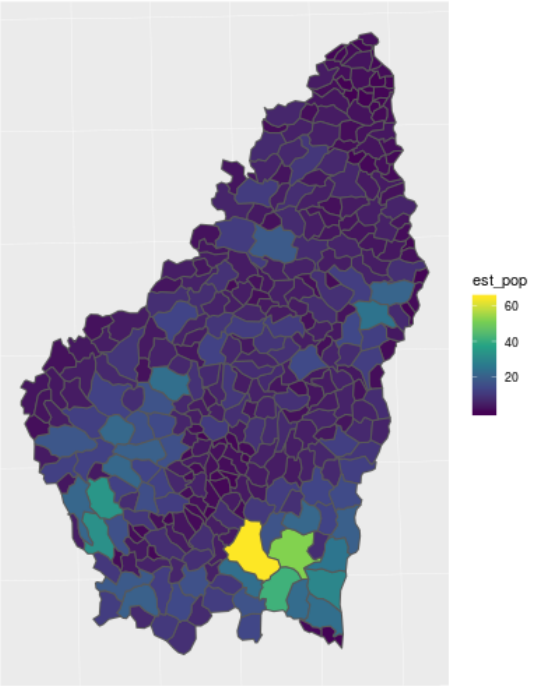


Estimated Population size by municipality

Integrate the population density estimate within each municipality

```
ardeche$est_pop <-
  raster::extract(
    x = wb_env$wb_popdens,
    y = as(ardeche, "Spatial"),
    df = TRUE
  ) %>%
  group_by(ID) %>%
  summarise(
    est_pop = sum(wb_popdens)
  ) %>%
  with(., as.vector(est_pop))

ggplot() +
  geom_sf(
    data = ardeche,
    aes(fill = est_pop)
  ) +
  scale_fill_viridis_c() +
  labs(x = NULL, y = NULL) +
  theme(
```



DCluster

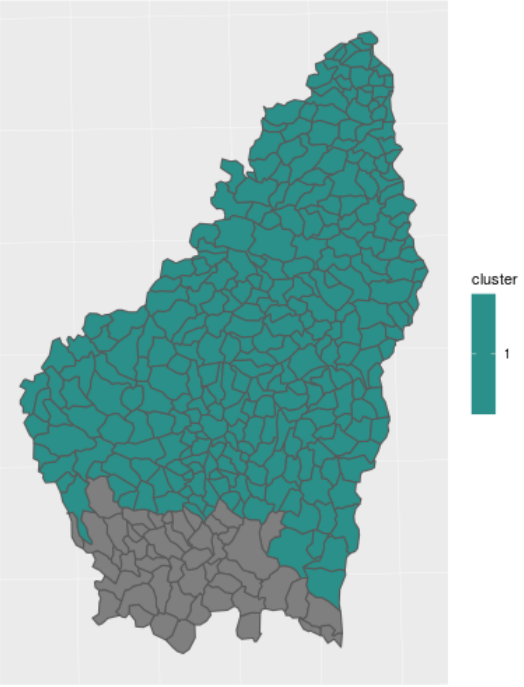
`DCluster` expects data in a `data.frame` with a specific format

```
ardeche_wb <-
  bind_cols(
    ardeche %>%
      st_geometry %>%
      st_centroid %>%
      st_coordinates %>%
      as.data.frame %>%
      rename(x = X, y = Y),
    ardeche %>%
      as.data.frame() %>%
      dplyr::select(n_obs, est_pop)
  ) %>%
  mutate(
    Observed = n_obs,
    Population = est_pop,
    Expected = Population * sum(Observed)/sum(Population)
  )
```

Kulldorff and Nagarwalla's method (SatScan)

```
#K&N's method over the centroids
mle<-calculate.mle(
  ardeche_wb,
  model="poisson"
)

knresults<-opgam(
  data=ardeche_wb,
  thegrid=ardeche_wb[,c("x","y")
  alpha=.01,
  iscluster=kn.iscluster,
  fractpop=.5,
  R=100,
  model="poisson",
  mle=mle
)
```



???

Moran's I statistic

- Analogous to Pearson's correlation coefficient for spatial data
- First, need to compute **neighbourhood** structure

```
## Nearest-neighbouring observation
(wb_nb <- ardeche %>%
   st_centroid %>%
   st_coordinates %>%
   knearneigh %>%
   knn2nb
)
```

```
## Neighbour list object:
## Number of regions: 339
## Number of nonzero links: 339
## Percentage nonzero weights: 0.2949853
## Average number of links: 1
## Non-symmetric neighbours list
```

```r
moran.test(
  ardeche$n_obs,
  listw = nb2listw(wb_nb)
)
```

```
##
##     Moran I test under randomisation
##
## data:  ardeche$n_obs
## weights: nb2listw(wb_nb)
##
## Moran I statistic standard deviate = 2.8591, p-value = 0.002124
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.121592056      -0.002958580       0.001897775
```

But this is a test for raw counts, which is not very interesting.

Spatio-temporal clustering

(next time)