Regional Segregation in Finland: Bayesian Hierarchical Model Study

Introduction

Statistics Finland periodically publishes a statistics dataset, Paavo, containing groups of variables over postal code areas describing demographics, household wealth, dwelling type and size, among other characteristics. Using Paavo, we chose to investigate regional wealth distribution using the proportion data of households in the wealthiest quantile. “Wealth” is here understood narrowly, ignoring relative factors such as cost of living.

For starters, we wanted to understand the spatial nature of our data and the information embedded in the dataset primary key e.g. the postal code. Next, we evaluated several model alternatives (see ch. “Focusing the Research Question”) and chose to go with a univariate hierarchical model. We parsed, cleaned and explored the data, e.g. from a text file to a tidied-up Pandas dataframe, and identified the columns (variables) we wanted to use. Then, we formulated the first model, which evolved in three steps - we present three alternative models and discuss the pros and cons of each. Each model is evaluated using the PSIS-LOO validation algorithm presented in the course.

Finally, we illustrate inferred differences between a handful of representative postal code regions, using the posterior samples obtained from our models, and assess our overall impression of the fit of the problem to methods used, and make conclusions.

Team, contributors

Markku Luotamo: model 1, hierarchical two-level Gaussian, using a pre-calculated wealth fraction
Janne Holopainen: model 2, introduction of binomial distribution to better account for uncertainty caused by disparate amounts of data per region in a more Bayesian way => hierarchical two-level binomial.
Markku Luotamo: model 3, reparametrization of binomial to eliminate constraints => hierarchical Gaussian-Logit-Binomial

Thanks to:
Janne Sinkkonen for general and specific model doctoring and valuable Bayesian insights.

Data

Postal Codes

Postal service in Finland has a nationwide coverage, and a hierarchical system of postal codes in place to coordinate mail delivery. The hierarchy has several levels embedded in the digits, most importantly the first three digits, biggest to smallest from left to right. So, given a postal code, you can look up its (two-digit) region. One design in the assignment of the postal codes, is that areas that can be seen to belong to distinguishable region, have postal code starting with same combination of numbers (e.g. 00 start for Helsinki) - also if postal codes (number) are close to each others, then one can assume that also the physical areas are close to each others (land area). [Yle new's article about the postal codes (In finnish)]

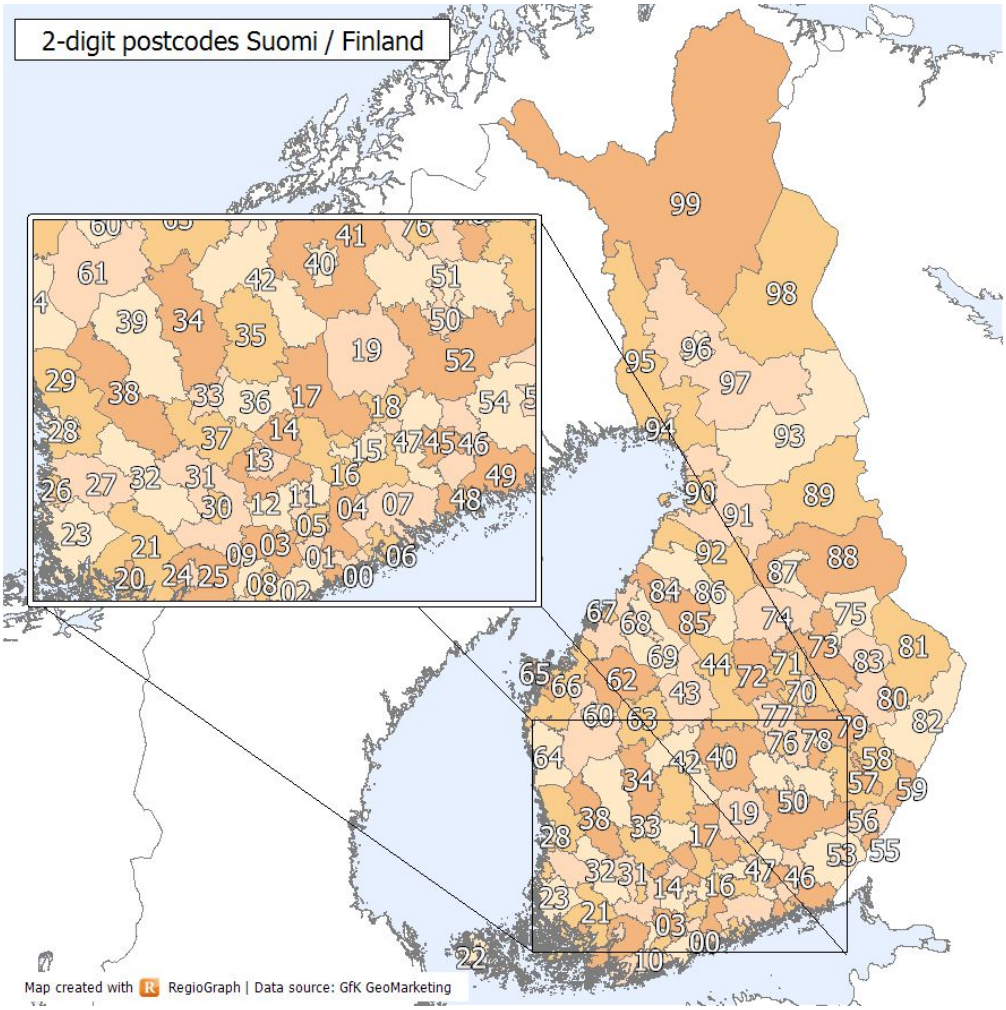


Figure: National & Regional levels of the Finnish postal code. Each two-digit region consists of multiple five-digit postal code areas.

Scoping the Paavo Dataset

Postal code area
x
y
surface_area
Inhabitants, total, 2016 (HE)
n_females_2016
n_males_2016
avg_age_2016
0-2 years, 2016 (HE)
3-6 years, 2016 (HE)
7-12 years, 2016 (HE)
13-15 years, 2016 (HE)
16-17 years, 2016 (HE)
18-19 years, 2016 (HE)
20-24 years, 2016 (HE)
25-29 years, 2016 (HE)
30-34 years, 2016 (HE)
35-39 years, 2016 (HE)
40-44 years, 2016 (HE)
45-49 years, 2016 (HE)
50-54 years, 2016 (HE)
55-59 years, 2016 (HE)
60-64 years, 2016 (HE)
65-69 years, 2016 (HE)
70-74 years, 2016 (HE)
75-79 years, 2016 (HE)
80-84 years, 2016 (HE)
85 years or over, 2016 (HE)
Aged 18 or over, total, 2016 (KO)
Basic level studies, 2016 (KO)
With education, total, 2016 (KO)
Matriculation examination, 2016 (KO)
Vocational diploma, 2016 (KO)
academic_degree_lower_2016
academic_degree_higher_2016
Aged 18 or over, total, 2015 (HR)
avg_inhabitant_income_2015
median_inhabitant_income_2015
n_inhabitants_lowest_income_2015
n_inhabitants_middle_income_2015
n_inhabitants_highest_income_2015
acc_inhabitant_purch_power_2015
n_households_2016
Average size of households, 2016 (TE)
Occupancy rate, 2016 (TE)
Young single persons, 2016 (TE)
Young couples w/o children, 2016 (TE)
Households with children, 2016 (TE)
Households with small children, 2016 (TE)
Households with children under school age, 2016 (TE)
Households with school-age children, 2016 (TE)
Households with teenagers, 2016 (TE)
Adult households, 2016 (TE)
Pensioner households, 2016 (TE)
Households living in owner-occupied dwellings, 2016 (TE)
Households living in rented dwellings, 2016 (TE)
Households living in other dwellings, 2016 (TE)
n_households_2015
avg_household_income_2015
median_household_income_2015
n_households_lowest_income_2015
n_households_middle_income_2015
n_households_highest_income_2015
acc_household_purchasing_power_2015
Free-time residences, 2016 (RA)
Buildings, total, 2016 (RA)
Other buildings, 2016 (RA)
Residential buildings, 2016 (RA)
n_dwellings_2016
avg_floor_area_2016
Dwellings in small houses, 2016 (RA)
Dwellings in blocks of flats, 2016 (RA)
Workplaces, 2015 (TP)
Primary production, 2015 (TP)
Processing, 2015 (TP)
Services, 2015 (TP)
A Agriculture, forestry and fishing, 2015 (TP)
B Mining and quarrying, 2015 (TP)
C Manufacturing, 2015 (TP)
D Electricity, gas, steam and air conditioning supply, 2015 (TP)
E Water supply; sewerage, waste management and remediation activities, 2015 (TP)
F Construction, 2015 (TP)
G Wholesale and retail trade; repair of motor vehicles and motorcycles, 2015 (TP)
H Transportation and storage, 2015 (TP)
I Accommodation and food service activities, 2015 (TP)
J Information and communication, 2015 (TP)
K Financial and insurance activities, 2015 (TP)
L Real estate activities, 2015 (TP)
M Professional, scientific and technical activities, 2015 (TP)
N Administrative and support service activities, 2015 (TP)
O Public administration and defence; compulsory social security, 2015 (TP)
P Education, 2015 (TP)
Q Human health and social work activities, 2015 (TP)
R Arts, entertainment and recreation, 2015 (TP)
S Other service activities, 2015 (TP)
T Activities of households as employers; undifferentiated goods- and services-producing activities of households for own use, 2015 (TP)
U Activities of extraterritorial organisations and bodies, 2015 (TP)
X Industry unknown, 2015 (TP)
n_inhabitants_2015
n_labour_force_2015
n_employed_2015
n_unemployed_2015
n_nonlabour_2015
Children aged 0 to 14, 2015 (PT)
Students, 2015 (PT)
Pensioners, 2015 (PT)
Others, 2015 (PT)
n_households_highest_income_2015_pc
n_inhabitants_highest_income_2015_pc
postal_region
postal_region_ix

Figure: Paavo dataset column layout. Note that the “primary key” is the ‘Postal Code Area’, e.g. ‘00100’ for Helsinki 10.

Column layout of the Paavo dataset is illustrated in the table above. Additional derivative columns were added for the two-digit postal region (e..g '00' for Helsinki), 1-based postal region index (1 for Helsinki 00). Out of a total of 3030 code areas, 2842 had non-NaN values for both our data columns of interest, i.e.. n_households_highest_income_2015 and n_households_2015 . For simplicity, we only use the 2015 data. Columns suffixed *_pc are our pre-calculated ratios for proportion of affluent households in the postal code area.

Focusing the Research Question : Alternative Bayesian Approaches

Initially, it was suggested to us to perform a factor analysis of "wellbeing" on Paavo data as a 2D dimension-reduction of several variables over postal codes and visualize it over a map. We did consider a simplified version of this idea, e.g. probabilistic PCA. However, we wanted to illustrate the full Bayesian workflow, whose steps would not be very clear in such a problem statement, not least because the interpretation of a single reduced component is highly subjective.

Next, we considered a regression model of some Paavo variables to explain Trafi data variables e.g. luxure car registrations, also available by postal code, but dismissed this as too time-consuming in our scope.

So, we decided to see how far we can evolve and harness a univariate hierarchical model to examine regional differences, using the Paavo data and the intrinsic hierarchical structure of the postal codes.

Models

Model 1: Gaussian common variance

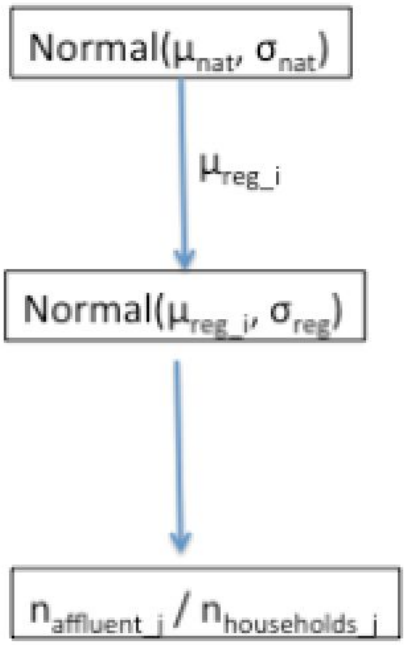


Figure: Model 1 - Gaussian hierarchical.
Indexes: i spans regions, j postal code areas.

Model description and rationale

This model was the starting point, an experiment to prove whether the data, containing an inherent hierarchical structure, would at all be suitable for a Bayesian hierarchical treatment. The model uses simple distributions with the minimal number of hierarchical levels, e.g. a national top-level Gaussian generating mu:s for a regional gaussian, generating affluence ratios [0,1].
The data was a pre-computed fraction n_affluent_households / n_households (which is bit of a frequentist simplification, addressed in Model 2 and its evolution Model 3)

Benefits

- Simplicity
- Small number of parameters, e.g. 204
- The first results in region comparison seemed quite plausible with respect to general knowledge and naively computed reference proportions for each region
- Reliable according to the k-hat values (< .5)
- Largest PSIS-LOO value, but can it really be positive ? (~4000 , see drawbacks)

Drawbacks

- Both hierarchy levels are modelled with real-support distributions well beyond [0,1] that need to be constrained, much like in the eight schools example. This disqualifies a large number of samples and can distort the results.
- Possibility: can small regions cause a numeric problem and bloat the loo values?
- Ignores the uncertainty emerging from regional differences in number of data points (postal code areas)
- The PSIS-LOO value averages to slightly over 1.0 (2000 samples for all 2841 postal code areas) , this can happen as our Gaussian is restricted to tight interval [0,1] and subsequently on some parts, the pdf can have relative probabilities higher than 1.0. We were not able to decide, if this would have negative effect on the PSIS-LOO calculation.

Prior

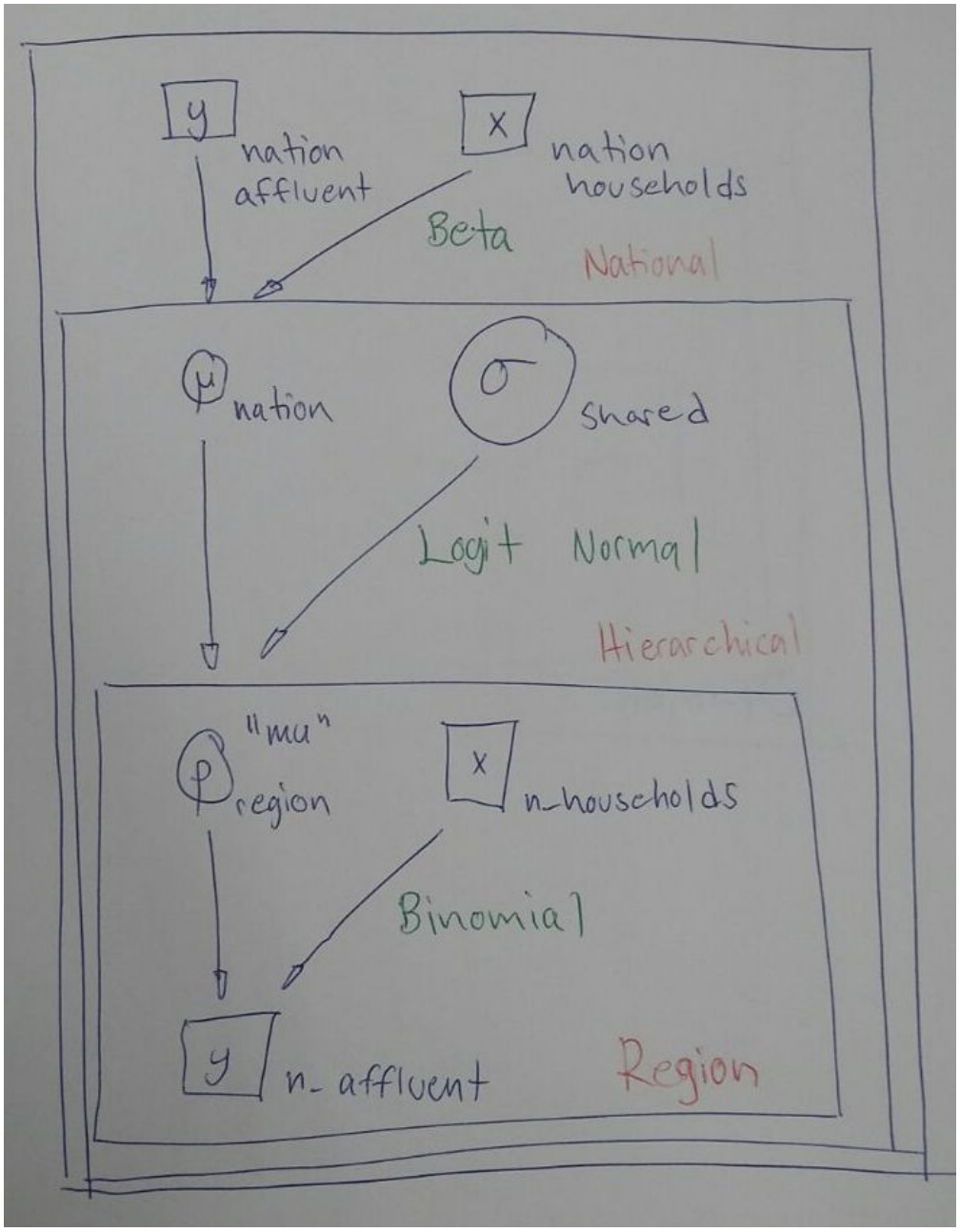
As with the eight schools model, the non-informative Stan uniform prior is assumed.

Model 2: Binomial-Gaussian common variance

Model description and rationale

Wanted to add some uncertainty to the data (i.e. amount of affluent households). Finding a good way to model and restrict the distribution of regional fractions, caused some headache.

- The true fraction of the affluent households might be uncertain.
- Regional fractions, would follow Gaussian distribution



Binomial-gaussian model using tight prior (close to national average).
Contains a mistake, on the beta-part: **nation_households** was later fixed to **nation_non_affluent_households**

Benefits

- Quite simple
- Takes into account uncertainty in the amount of affluent households and regional differences in data point amounts
- Most of the k-hat values are below 0.7 (167 of 2840)

Drawbacks

- Probably tried making the model too simple for the case - the binomial does not allow enough variation for the different postal codes inside the region.
- Low PSIS-LOO value - ‘bad’ (compared to logit-binomial) at predicting
- Regional values tend towards national average.
- Kontula has so small likelihood to be in the model, that a constant term needed to be added to its probability. (Pushes towards conclusion, that the binomial-gaussian might not be reliable.)

Prior

For the average of the regional averages, following averages were tested:

- **Tight prior**: Close to national fraction of affluent households. Idea was that overall variation of the regional fractions would converge towards the national average. This did not however take into account that, the average is centered around 0.18, meaning that tail cases (say >0.5) can have higher impact on the outcome.
    - Beta(national_affluent, national_non_affluent)
- **Loose prior**: As the amount of data is reasonable (multiple data-points per region), a loose prior allows “the data to speak”. Here also was taken into account the effect of using log-normal distribution.
    - Beta(6, 15)

Model 3: Logit-Binomial-Gaussian regional variance

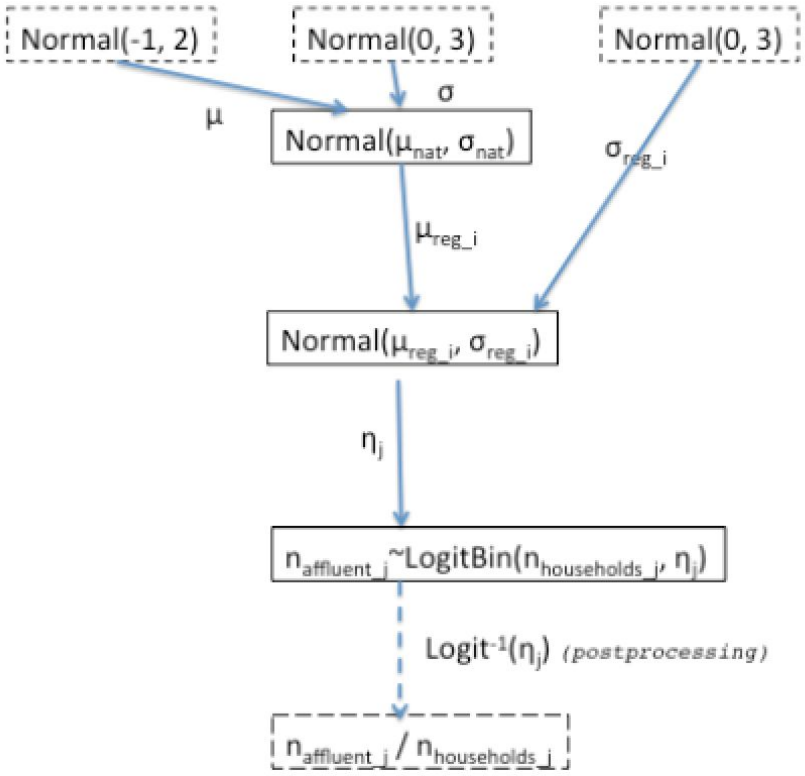


Figure: Model 3 – Logit-binomial hierarchical.
Indexes: i spans regions, j postal code areas.

Model description and rationale

The aim was to

- eliminate support constraints from the model altogether (std > 0 constraints remaining) to get as “natural” and robust a representation as possible
- Use stan’s built-in reparametrized distributions to avoid intermediate transformed terms
- Allow for differing regional variances

The challenge of the proportion variable having a support of [0,1] was addressed by logit reparametrization, which was supported directly by Stan as the binomial_logit distribution, which, instead of a probability p, accepts a transformed probability logit(p) ε R.

So, instead of n~Bin(N,p), we have n~Bin(N,η=logit(p)) . The expected values of the Gaussians (generating p) are expressed in the real-valued logit space, thus removing constraints.

Benefits

- Constraint-free supports
- Clarity due to simple formulation
- Intra-regional variance
- Of the three PSISLOO values of the respective models, this is the middle ranking model. So, should the Gaussian psisloo be disqualified (> +4000), this looks like the next best in terms of PSISLOO

Drawbacks

- Slightly more parameters than data, which we heard is not unusual for Bayesian models
- Reliability: roughly half of K_hat values are >0.7, mostly between 0 and 1.2, so this casts some doubt on model reliability.

Priors

Loose priors were set for the national level:

- μ_nat: normally distributed , centered a bit below the southern Finland affluence fraction of 0.26 (= expit(-1)), variance of 2
- σ_*: The positive half of a Gaussian prior distribution for all σ , with a zero-centered variance of 3 is an educated guess - let the data speak for the rest.

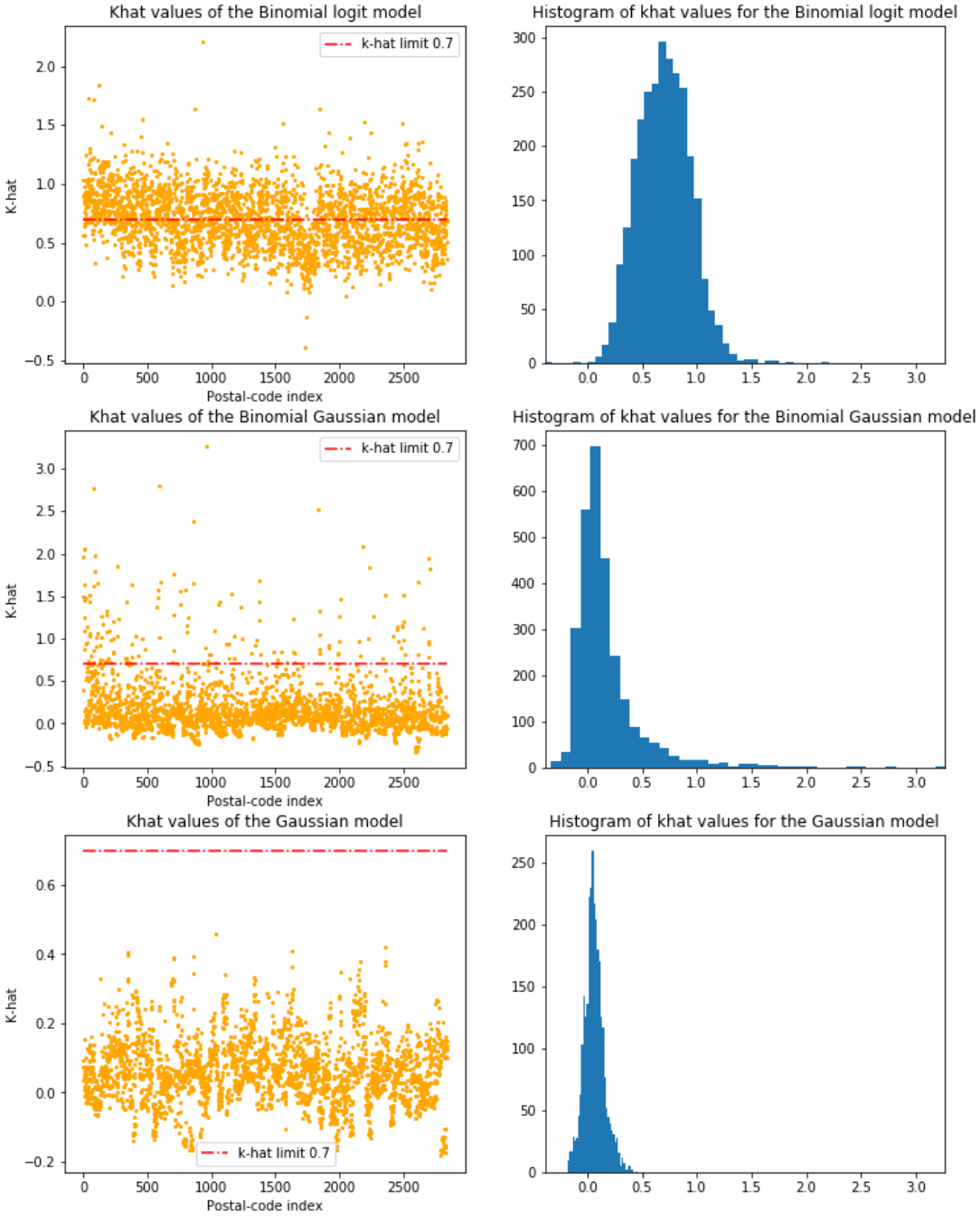
Model comparison and selection

PSIS-LOO and p_eff

Model name	PSIS-LOO	p_eff (effective number of parameters based on PSIS-LOO)	Actual parameters
Gaussian	4024	82.98	103
Binomial-Gaussian	-39812	2063.23	202
Logit-Binomial-Gaussian	-10570	1715.47	3043

The above table would suggest that Binomial-Gaussian model **is too simple** for modeling spatial regions as hierarchical structure of the postal code areas. As mentioned in the discussion of the Gaussian model, we don't quite know if the PSIS-LOO value of Gaussian model is comparable to our other models (Our binomial models use pmf, so probability is always <= 1).

K-Hat values



Visualisation of the k-hat distributions.

The only model with really well behaving k-hat values is the Gaussian model (presumably because of the simplicity). The Binomial model has 176 data-points over the 0.7 threshold, which could be explained by the outlying data-points in the data (This would include e.g.

Otaniemi and Kontula, which differ a lot from the regional average). We can also see that the k-hat values for the Logit-Binomial-Gaussian model are distributed around the 0.7 threshold - this does cast uncertainty about our predictive performance estimation. The results however, do follow reasonably well the results of the Gaussian model (See Regional comparison using posterior estimates) which has well behaving k-hat values.

Divergence of the models

During the Stan sampling, the size of the chains were adjusted so, that the R_hat values of all of the models were between 1.0 and 1.02.

For the models, the following options were finally used to run and compare the models:

Model	Iterations	Number of chains
Gaussian	1000	4
Binomial-Gaussian	1000	2
Logit-Binomial-Gaussian	5000	2

Table for the used sampling parameters.

Regional comparison using posterior estimates

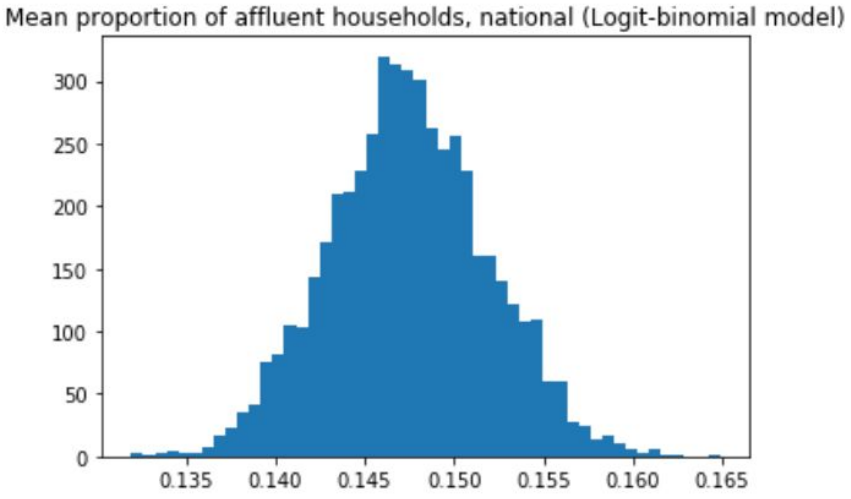


Figure: Posterior marginal distribution for the nationwide mean fraction of affluent households, e.g. obtained as the the logit^-1(μ_nat) parameter of Model 3

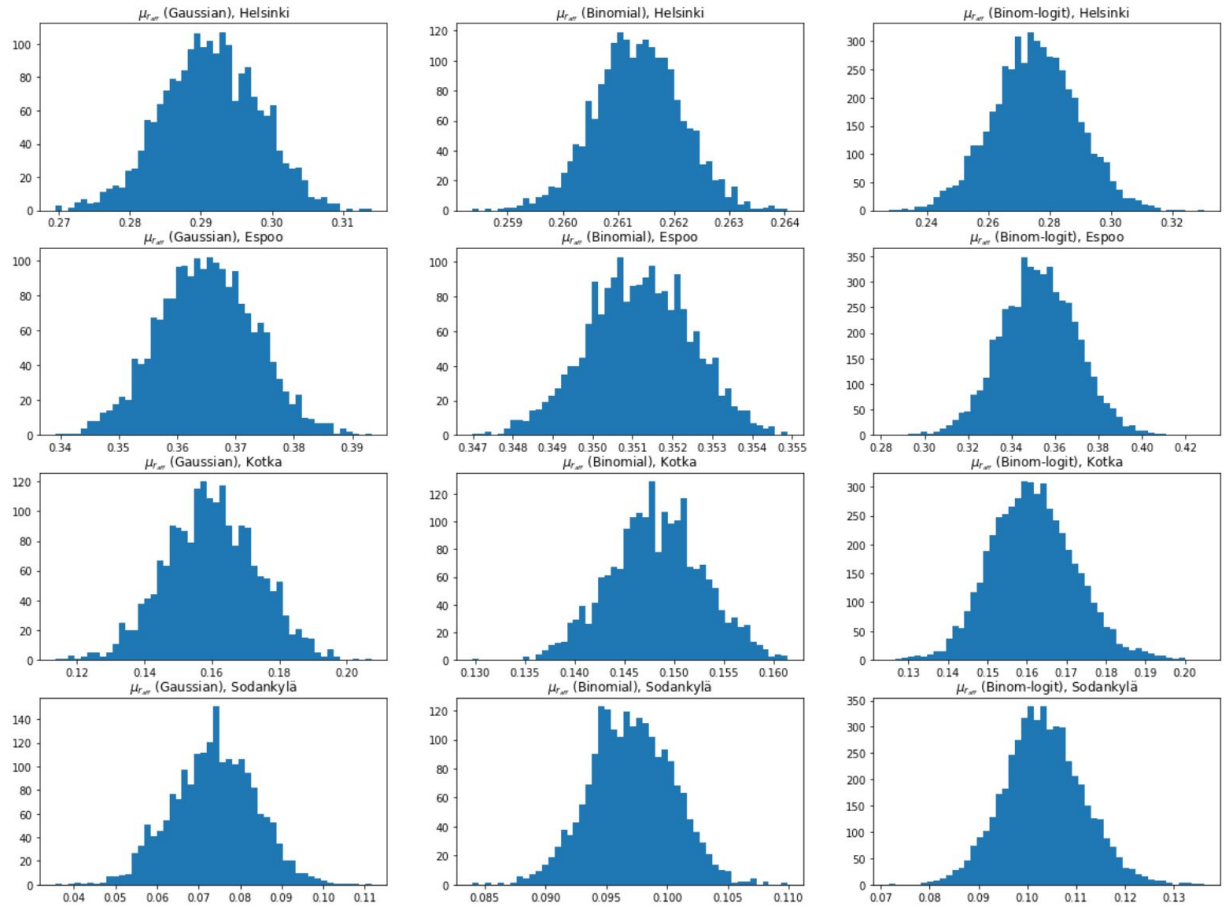


Figure: Comparing posterior mean affluence fractions between postal code regions and models

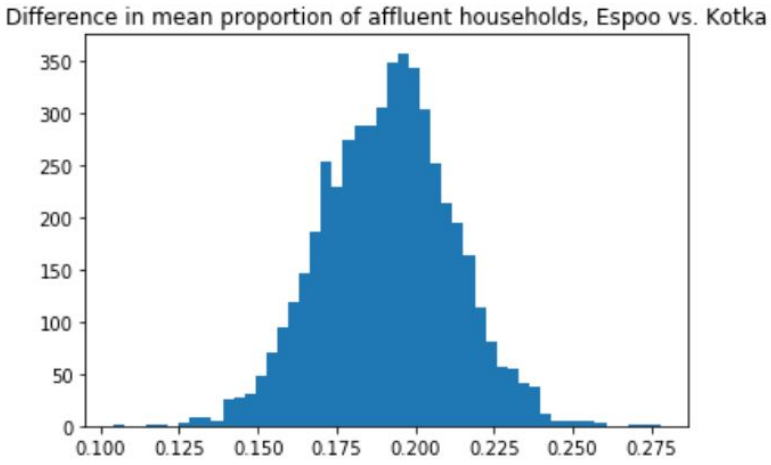


*Figure: Posterior difference of Espoo and Kotka in the fraction of wealthy household*s

Visualizing the estimates, the expected values or modes for the mean affluence fraction (i.e. proportion of households in the highest Paavo quantile) suggested by the different models for a handful of selected regions seem to agree roughly: Espoo is in a class of its own (around 35% of

households are in the most affluent quantile) , Helsinki around 28..29%, Kotka 16%, and Sodankylä 10%. Notably though, the variance of Model 2 (referred as Binomial above) seems to be quite narrow, see Model 2 assessment above. The Bayesian ease of computing proportion differences is illustrated with the proportion difference between Espoo and Kotka, centering around a 19% difference. Also, computation of posterior Central Confidence Intervals would be relatively straightforward, but omitted given the time constraints of the project.

Conclusions

Although the Gaussian model, has the highest PSIS-LOO value, the uncertainty around the high relative probability caused us to conclude that out of the three alternatives the **Logit-Binomial-Gaussian** is most likely the best performing one. Most pressing reasons for this is, that we are able to include the uncertainty of the data to the model. Also for this model the PSIS-LOO value is reasonable, and the p_eff value is lower than the number of actual parameters.
Additionally it seems reasonable to conclude, that the structure of the postal-code system does contain information about the neighboring postal code areas, in terms of economical status of the inhabitants.

As far as the socio-economical scope of our research question goes, there do seem to be readily visible indications of segregation between cities, and on the other hand, cities and the countryside.

Potential improvements

Currently the model is only modeling a single feature (fraction of affluent households) for each region and postal code area. Our initial plan was to test if the estimated feature would have correlation with other features, e.g. building of certain type of housing, or registration of electric cars.
Using the estimated features, one could perform better comparisons between different regions of the nation.
If one would wish to improve the estimates, one could try to find and include an explanatory covariate variable for the fraction (Paavo dataset does contain information about, for example, people working in different industries).

Source Codes

```
data {
  int<lower=0> n_postal_codes; // number of postal code data points
  int <lower=0> n_postal_regions; // number of two-digit areas (groups)
  int<lower=1,upper=n_postal_regions> postal_region_ix[n_postal_codes]; // group indicator
  vector[n_postal_codes] pct_affluent_households; // observations
}
parameters {
 real<lower=0> mu_national;      // hyperprior mean
 real<lower=0> sigma_national;      // hyperprior mean
 vector<lower=0>[n_postal_regions] mu_regional;      // group means
 real<lower=0> sigma_regional;    // group stds
}

model {
 mu_regional ~ normal(mu_national, sigma_national);
 pct_affluent_households ~ normal(mu_regional[postal_region_ix], sigma_regional);
}
generated quantities {
  vector[n_postal_codes] log_lik;
  for (i in 1:n_postal_codes)
     log_lik[i] = normal_lpdf(pct_affluent_households[i] | mu_regional[postal_region_ix[i]],
sigma_regional);
}
```

Stan-code for the **Gaussian model**.

```
data {
  int<lower=0> n_postal_codes; // number of postal code data points
  int<lower=0> n_postal_regions; // number of two-digit areas (groups)
  int<lower=1,upper=n_postal_regions> postal_region_ix[n_postal_codes]; // group indicator
  int n_affluent_households[n_postal_codes]; // observations
  int n_households[n_postal_codes];
}
```

```
transformed data {
  int<lower=0> n_national_households;
  int<lower=0> n_national_affluent;
  int<lower=0> n_national_nonaffluent;
  n_national_households = sum(n_households);
  n_national_affluent = sum(n_affluent_households);
  n_national_nonaffluent = n_national_households - n_national_affluent;
}
parameters {
  vector[n_postal_regions] logit_p_regional;
  real<lower=0> national_sigma;
  real<lower=0, upper=1> national_mu;
}
transformed parameters {
  vector<lower=0, upper=1>[n_postal_regions] p_regional;
  // To use logit-normal, Stan likes to have untransformed variables on the left hand-side of
  // ~ -sign. Thus we will have 'logit_p_regional' follow normal, and only output the transformed
  // parameter 'p_regional' that we are interested in.
  p_regional = inv_logit(logit_p_regional);
}
model {
  national_mu ~ beta(6, 15);
  // Using logit-normal distribution, but Stan requires us to do the logit part in the transformed
  // parameters part
  logit_p_regional ~ normal(national_mu, national_sigma);
  for (i in 1:n_postal_codes) {
      n_affluent_households[i] ~ binomial(n_households[i], p_regional[postal_region_ix[i]]);
  }
}
generated quantities {
  vector[n_postal_codes] log_lik;
  for (i in 1:n_postal_codes) {
    log_lik[i] = binomial_lpmf(n_affluent_households[i] | n_households[i],
p_regional[postal_region_ix[i]]);
  }
}
```

Stan code for the **Binomial-Gaussian model.**

```
data {
  int<lower=0> n_postal_codes; // number of postal code data points
  int <lower=0> n_postal_regions; // number of two-digit areas (groups)
  int<lower=1,upper=n_postal_regions> postal_region_ix[n_postal_codes]; // group indicator
  int n_affluent[n_postal_codes]; // observations
  int n_households[n_postal_codes]; // total number of households per postal code
}
parameters {
 real mu_national;      // hyperprior mean
 real<lower=0> sigma_national;      // hyperprior std
 vector[n_postal_regions] mu_regional;      // group means
 vector<lower=0>[n_postal_regions] sigma_regional;      // group std
 vector[n_postal_codes] eta; // logit proportion for reparametrized binomial
}
model {
 mu_national ~ normal(-1, 2);
 sigma_national ~ normal(0, 3);
 sigma_regional ~ normal(0, 3);
 mu_regional ~ normal(mu_national, sigma_national);
 eta ~ normal(mu_regional[postal_region_ix], sigma_regional[postal_region_ix]);
 n_affluent ~ binomial_logit(n_households, eta);
}
generated quantities {
  vector[n_postal_codes] log_lik;
  for (i in 1:n_postal_codes)
    log_lik[i] = binomial_logit_lpmf(n_affluent[i] | n_households[i], eta[i]);
}
```

Stan code for the **Logit-Binomial-Gaussian model**

```
import pystan
import numpy as np
import pickle
import pandas as pd
```

```
with open('paavodata_cleaned_df.pkl', 'rb') as f:
  paavo_df = pickle.load(f)

n_postal_regions = paavo_df['postal_region'].nunique()
n_postal_codes = paavo_df.shape[0]
postal_region_ix = paavo_df['postal_region_ix']
pct_affluent_households = paavo_df['n_households_highest_income_2015_pc']
print(f'n_postal_codes={n_postal_codes}, n_postal_regions={n_postal_regions},
postal_region_ix={postal_region_ix}')
print(f'pct_affluent_households={pct_affluent_households}')

n_households_total = paavo_df['n_households_2015']
n_affluent_households = paavo_df['n_households_highest_income_2015']

not_isnan_ix = np.logical_not(np.logical_or(np.isnan(n_households_total),
np.isnan(n_affluent_households)))
postal_region_ix = paavo_df['postal_region_ix'][not_isnan_ix]
n_affluent_households = n_affluent_households[not_isnan_ix]
n_households_total = n_households_total[not_isnan_ix]
pct_affluent_households = n_affluent_households/n_households_total
n_postal_codes = np.sum(not_isnan_ix)
n_postal_regions = paavo_df['postal_region'][not_isnan_ix].nunique()

data = dict(n_postal_codes=n_postal_codes,
        n_postal_regions=n_postal_regions,
        pct_affluent_households = pct_affluent_households,
        postal_region_ix=postal_region_ix)

model = pystan.StanModel(model_code=stan_code)
fit = model.sampling(data=data, iter=1000, chains=4)
print(fit)
extracts = fit.extract(permuted=True)
posterior_samples = [extracts[param] for param in ['mu_national', 'mu_regional', 'sigma_national',
'sigma_regional', 'log_lik']]

with open('wellbeing_hierarchical_gaussian_fit.txt', "w+") as f:
  f.write(str(fit))
```

```
with open('wellbeing_hierarchical_gaussian.pkl', 'wb') as f:
  pickle.dump(posterior_samples, f)
```

Python code used to run the **Gaussian model.**

```
import pystan
import numpy as np
import pickle
import pandas as pd
from os import path as op

with open('paavodata_cleaned_df.pkl', 'rb') as f:
  paavo_df = pickle.load(f)

# Areas with no households?
paavo_df = paavo_df.loc[~paavo_df["n_households_2015"].isna()]

n_postal_regions = paavo_df['postal_region'].nunique()
n_postal_codes = paavo_df.shape[0]
postal_region_ix = paavo_df['postal_region_ix']
n_affluent_households = paavo_df['n_households_highest_income_2015']
n_households_total = paavo_df['n_households_2015']

# Nan cleaning
not_isnan_ix = np.logical_not(np.logical_or(np.isnan(n_households_total),
np.isnan(n_affluent_households)))
postal_region_ix = paavo_df['postal_region_ix'][not_isnan_ix]
n_affluent_households = n_affluent_households[not_isnan_ix].astype(int)
n_households_total = n_households_total[not_isnan_ix].astype(int)
n_postal_codes = np.sum(not_isnan_ix)
n_postal_regions = paavo_df['postal_region'][not_isnan_ix].nunique()


print(f'n_postal_codes={n_postal_codes}, n_postal_regions={n_postal_regions},
postal_region_ix={len(postal_region_ix)}')
print(f'n_affluent_households={len(n_affluent_households)},
n_households={len(n_households_total)}')
```

```
data = dict(
  n_postal_codes=n_postal_codes,
  n_postal_regions=n_postal_regions,
  n_affluent_households=n_affluent_households,
  n_households=n_households_total,
  postal_region_ix=postal_region_ix)

model = pystan.StanModel(file=op.join(op.dirname(__file__),"single_param_bino.stan"))
fit = model.sampling(data=data, iter=1000, chains=2)
extracts = fit.extract(permuted=True)

posterior_samples = [extracts[param] for param in ['p_regional', 'log_lik', 'national_sigma',
'national_mu']]

with open('wellbeing_hierarchical_binomial_fit.txt', "w+") as f:
  f.write(str(fit))

with open('wellbeing_hierarchical_binomial.pkl', 'wb') as f:
  pickle.dump(posterior_samples, f)
```

Python code used for running the **Binomial-Gaussian model.**

```
import pystan
import numpy as np
import pickle
import pandas as pd

with open('paavodata_cleaned_df.pkl', 'rb') as f:
  paavo_df = pickle.load(f)

n_affluent_households = paavo_df['n_households_highest_income_2015']
n_households_total = paavo_df['n_households_2015']
# Nan cleaning
not_isnan_ix = np.logical_not(np.logical_or(np.isnan(n_households_total),
np.isnan(n_affluent_households)))
postal_region_ix = paavo_df['postal_region_ix'][not_isnan_ix]
n_affluent_households = n_affluent_households[not_isnan_ix]
```

```
n_households_total = n_households_total[not_isnan_ix]
n_postal_codes = np.sum(not_isnan_ix)
n_postal_regions = paavo_df['postal_region'][not_isnan_ix].nunique()

data = dict(
    n_postal_codes=n_postal_codes,
    n_postal_regions=n_postal_regions,
    postal_region_ix=postal_region_ix.values,
    n_affluent=n_affluent_households.values.astype(int),
    n_households=n_households_total.values.astype(int))
model = pystan.StanModel(model_code=stan_code)
fit = model.sampling(data=data, iter=5000, chains=2)

print(fit)
extracts = fit.extract(permuted=True)
posterior_samples = [extracts[param] for param in ['mu_national', 'mu_regional', 'sigma_national',
'sigma_regional', 'eta', 'log_lik']]

with open('affluence_hierarchical_logit_bin_fit.txt', "w+") as f:
  f.write(str(fit))

with open('affluence_hierarchical_logit_bin.pkl', 'wb') as f:
  pickle.dump((postal_region_ix, posterior_samples), f)
```

Python code for running the **Logit-Binomial-Gaussian model.**

As the data preprocessing and plotting spans across multiple jupyter-notebooks, we chose not to include them here. The used data is described in “Scoping the Paavo Dataset”-part.