Project work

Anonymous

Contents

1 Introduction to our project **2**

1.1 Our problem . . . 2

1.2 Our data . . . 2

1.3 Data preprocessing . . . 3

2 The pooled model **4**

2.1 The description of model . . . 4

2.1.1 The model . . . 4

2.1.2 The prior choices . . . 4

2.2 Stan-model . . . 5

2.3 Running the Stan-model . . . 5

2.4 Convergence diagnostics . . . 6

2.4.1 HMC/NUTS specific diagnostics . . . 6

2.4.2 Rhat and ESS diagnostics . . . 6

2.5 Posterior predictive checking & Predictive performance . . . 8

2.6 Sensitivity analysis . . . 9

3 The hierarchical model **10**

3.1 The description of model . . . 10

3.1.1 The model . . . 10

3.1.2 The prior choices . . . 10

3.2 Stan-model . . . 11

3.3 Running the Stan-model . . . 12

3.4 Convergence diagnostics . . . 12

3.4.1 HMC/NUTS specific diagnostics . . . 12

3.4.2 Rhat and ESS diagnostics . . . 13

3.5 Posterior predictive checking & Predictive performance . . . 16

3.6 Sensitivity analysis . . . 17

4 Comparing the models **18**

4.1 Convergence . . . 18

4.2 Posterior predictiveness . . . 18

4.3 Sensitivity to chosen priors . . . 18

4.4 Applicability to our problem . . . 18

5 Conclusion **19**

5.1 Potential improvements . . . 19

5.2 Main findings . . . 19

6 References **20**

1 Introduction to our project

1.1 Our problem

The World Health Organization (WHO) reports annually the life expectancies of each its regions.

“Life expectancy at birth reflects the overall mortality level of a population. It summarizes the mortality pattern that prevails across all age groups in a given year – children and adolescents, adults and the elderly.” (WHO)

The Organisation for Economic Co-operation and Development (OECD) reminds us that gains in life expectancy at birth can be connected to a several factors, including rising living standards, improved lifestyle and better education, as well as greater access to quality health services. Therefore, modeling the global life expectancy mathematically could reveal the general trend on our globe. Especially the hierarchical approach could help us to estimate the *superpopulation* of which each WHO-region has obtained their values. The underlying population could let us truly monitor the general trend globally, and conclude how fairly the positive changes are spread and distributed.

In this project, we are trying out two different models: **The pooled model** which treats all WHO-regions the same and **The hierarchical model** which considers the parameters of each region as a sample from a superpopulation. The latter is an intermediate solution where the level of pooling is determined by the data and a prior on the amount of pooling. The main task is to **model the varying slopes and intercepts of each region** and **estimate the underlying superpopulation** that let us make inferences on other regions and globally.

We perform the same tricks, plots and diagnostics for both models in their own sections. The final model comparision occurs in the section 4. The mathematical formulations for both models are also described later in the notebook. In the last section we list the main findings and potential improvements.

1.2 Our data

Our main goal during the project was to focus on the methodology and bayesian framework. We decided to keep the number of variables low in order study the behaviour of our parameters more visually and invest our time in careful and effective implementation. The models below are easily generalized to multidimensional case.

The data set is provided by WHO and it can be found here: http://apps.who.int/gho/data/node.main.

Data set is 30x5 matrix and the columns were the year, WHO-region and the life expectancies at birth for male, female and both sexes. Our data points were from the years of: - 2000, 2005, 2010, 2015 and 2016.

The WHO-regions that are included in the data set are the following: - Africa, Europe, Americas, Eastern Mediterranean, South-East Asia, Western Pacific.

The descriptive statistics of our data are introduced in the next section. Other relevant information on the data (directly copied from data source):

- Rationale: Life expectancy at birth reflects the overall mortality level of a population. It summarizes the mortality pattern that prevails across all age groups - children and adolescents, adults and the elderly.
- Definition: The average number of years that a newborn could expect to live, if he or she were to pass through life exposed to the sex- and age-specific death rates prevailing at the time of his or her birth, for a specific year, in a given country, territory, or geographic area.
- Method of measurement Life expectancy at birth is derived from life tables and is based on sex- and age-specific death rates. Life expectancy at birth values from the United Nations correspond to mid-year estimates, consistent with the corresponding United Nations fertility medium-variant quinquennial population projections.
- Preferred data sources: Civil registration with complete coverage

1.3 Data preprocessing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import arviz as az
import pystan
import seaborn as sns
from matplotlib.patches import Patch
from matplotlib.lines import Line2D
import warnings
warnings.filterwarnings('ignore')
```

```
# Load the data
df_original = pd.read_csv("~/BDA_course_Aalto/prokkis/BDA-project2019/data/data.csv")
df_original = df_original.rename(columns=df_original.iloc[0])
df = df_original.drop(df_original.index[0])

df["Year"] = df["Year"].astype(int)
df["Male"] = df["Male"].astype(float)
df["Female"] = df["Male"].astype(float)
df["Both sexes"] = df["Both sexes"].astype(float)

# Quick glance at our data
df.describe().T.round(1)
```

```
##              count    mean  std     min     25%     50%     75%     max
## Year          30.0  2009.2  6.1  2000.0  2005.0  2010.0  2015.0  2016.0
## Both sexes    30.0    69.6  7.3    50.8    65.7    71.0    75.6    77.5
## Male          30.0    67.4  6.7    49.6    64.5    68.2    72.6    75.0
## Female        30.0    67.4  6.7    49.6    64.5    68.2    72.6    75.0
```

2 The pooled model

2.1 The description of model

2.1.1 The model

Pooled regression model is one type of model that has constant coefficients, referring to both intercepts and slopes. Formally,

$$y_n = \beta x_n + \alpha + \epsilon_n \quad \text{where} \quad \epsilon_n \sim \mathrm{N}(0, \sigma^2)$$

The same model can be also written as

$$y_n \sim \mathrm{N}(\beta x_n + \alpha, \sigma^2)$$

Note that the stan model uses the deviation as the second parameter - not the variance.

2.1.2 The prior choices

Both the slope and intercept parameters will be modelled with a normal distribution. The prior for the intercept will be $\alpha \sim N(-600, 100)$ because we want to give a very vague weakly-informative effect on the intercepts. The mean of the intercept for a linear fit to the full data is close to -600 so the distribution is centered at -600. The standard deviation is kept as a large value because we know that there is a large variation in the intercepts between the regional groups. The prior for the slope parameter is also modelled with a normal distribution with the approaches given below. The last two approaches are used only for **the sensitivity analysis** and the first prior is used in the model that is used for inference.

- Scenario I. (*This one was selected for inference, the other priors are used only for sensitivity analysis*)

$$\beta \sim \mathrm{N}(0, \tau^2) \quad \text{where} \quad \tau = 16$$

Reasoning: It is very unlikely that the mean life expectancy would change more than 50 % in one year. The prior above bounds the rate to have P$(-\frac{69.6}{2} \leq \tau \leq \frac{69.6}{2} = 0.99)$.

- Scenario II.

$$\beta \sim \mathrm{N}(0, \tau^2) \quad \text{where} \quad \tau = 26$$

Reasoning: For the sensitivity analysis, we let the increase or decrease per year be even larger than 50%.

- Scenario III.

$$\beta \sim \text{uniform}$$

Reasoning: The noninformative or flat prior, which is equivalent to uniform prior on the parameter. This prior was also set for sensitivity analysis purposes.

2.2 Stan-model

```
the_pooled_model = """
    data {
        int<lower=0> N;
        vector[N] x;
        vector[N] y;
        real xpred;
        real tau;                  // prior std for beta
    }
    parameters {
        real alpha;                // intercept
        real beta;                 // slopes of vars
        real<lower=0> sigma;       // total error
    }
    model{
        beta ~ normal(0, tau);     // Beta parameter prior-distribution
        alpha ~ normal(-600, 100); // Alpha parameter prior-distribution
        sigma ~ normal(0, 10);     // error
        y ~ normal(alpha + beta*x, sigma);
    }
    generated quantities {
        vector[N] yrep;
        real ypred;

        for (n in 1:N) {
            yrep[n] = normal_rng(alpha + beta*x[n], sigma);
        }

        ypred = normal_rng(alpha + beta*xpred, sigma);
    }
"""
```

2.3 Running the Stan-model

```
pooled_data_dict = {}
pooled_data_dict["N"] = len(df["Year"])
pooled_data_dict["x"] = np.asarray(df["Year"].values, dtype=int)
pooled_data_dict["y"] = np.asarray(df["Both sexes"].values, dtype=float)
pooled_data_dict["xpred"] = 2020
pooled_data_dict["tau"] = 16

pooled_fit = pystan.stan(
    model_code = the_pooled_model,
    data = pooled_data_dict,
    iter = 2000,
    chains = 4,
    control=dict(
        max_treedepth=15
    )
)
```

2.4 Convergence diagnostics

2.4.1 HMC/NUTS specific diagnostics

```
p = pooled_fit.to_dataframe(diagnostics=True)
print('divergent transitions: ', any(p['divergent__']))
```

```
## divergent transitions:  False
```

"If there are too many divergent transitions, then the sampler is not drawing samples from the entire posterior and inferences will be biased." [4] The `adapt_delta` -parameter could be used to tune the step size but there is no need since the run did not produce *divergent transitions* and therefore we did not have to perform any step size tuning.

"If the sampler is often hitting the maximum number of steps, it means that the optimal number of steps to take in each iteration is higher than the maximum. While divergent transitions bias inference, a too-small maximum tree-depth only affects efficiency." [4]. We adjusted the tree depth to size 15, to let the sampler explore the posterior more. This was sufficient increase.

```
print('treedepth quantiles:\n', p['treedepth__'].quantile([0, 0.25, 0.5, 0.75, 1]))
```

```
## treedepth quantiles:
##  0.00    1.0
## 0.25    6.0
## 0.50    8.0
## 0.75    8.0
## 1.00    9.0
## Name: treedepth__, dtype: float64
```

2.4.2 Rhat and ESS diagnostics

```
print(pooled_fit.stansummary(pars=['sigma', 'alpha', 'beta', 'ypred']))
```

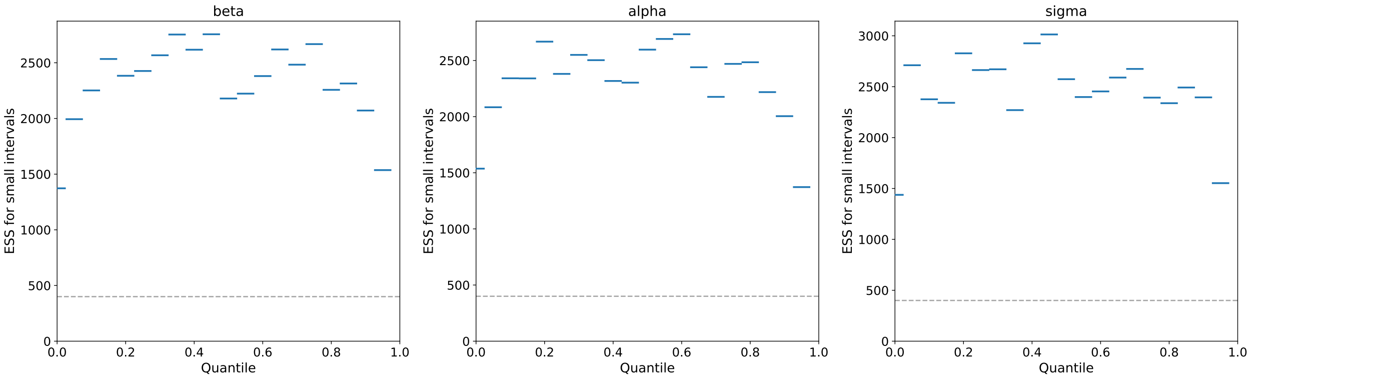
```
## Inference for Stan model: anon_model_5eae7a025e53e65aacd44b6c2f3b03c7.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##         mean se_mean     sd   2.5%    25%    50%    75%  97.5%  n_eff   Rhat
## sigma   7.27    0.03   1.04   5.55   6.52   7.18   7.89   9.59   1541    1.0
## alpha -601.6    2.84  95.92 -783.9 -667.1 -603.2 -535.6 -418.1   1141    1.0
## beta    0.33  1.4e-3   0.05   0.24    0.3   0.33   0.37   0.42   1141    1.0
## ypred  73.35    0.12   7.54  58.53  68.37  73.33  78.38  88.64   4122    1.0
##
## Samples were drawn using NUTS at Sun 08 Dec 2019 19:27:06 .
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Rhat: "The $\hat{R}$ statistic measures the ratio of the average variance of draws within each chain to the variance of the pooled draws across chains; if all chains are at equilibrium, these will be the same and $\hat{R}$ will be one." The Rhat values are required to be less than 1.05 to justify convergence of the draws. [1]

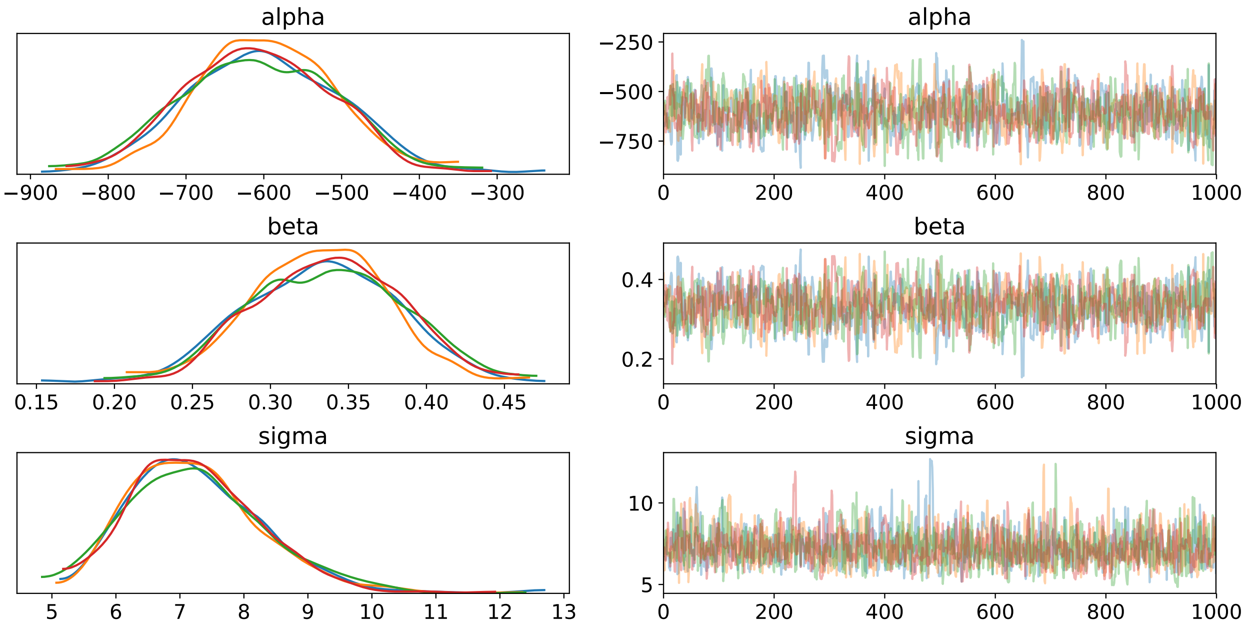
ESS: "The n_{eff} is an estimate of the sample size required to achieve the same level of precision if that sample was a simple random sample. Since the inference of bayesian models is based on the posterior draws and the MCMC process causes the draws to be correlated, the ESS is usually lower than the actual number of draws. Now the higher the effective sample size the more likely the chains for the draws have converged." It is stated in stan documentation that the bulk ESS should be atleast 100 times the number of chains. [2]

The Rhat values for each of the parameters is equal to 1 and the ESS values clearly exceed the 100*n_chains criteria as can be seen from the prints above and plots below. In fact, what can bee seen from the ESS-local-plot below is that during posterior draws the ESS never falls below the ertical line. This line is considered to be the minimum for convergence at any time. Also the trace plots below (which are time series plots of the Markov chains) shows obivous convergence. Based on these, it is clear that the model has converged properly.

```
_ = az.plot_ess(
  pooled_fit, var_names=["beta", "alpha", "sigma"],
  kind="local", marker="_", ms=20, mew=2, figsize=(20, 6)
)
plt.show()
```

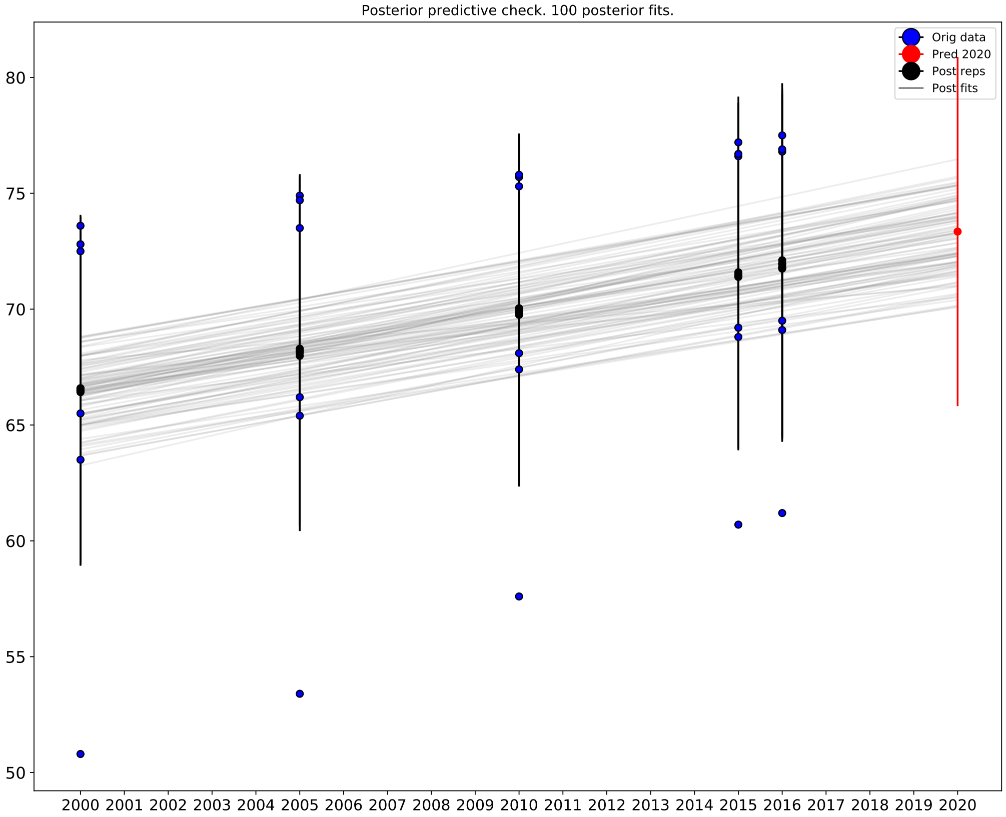


```
_ = az.plot_trace(
  pooled_fit,
  var_names = ["alpha", "beta", "sigma"]
)
plt.show()
```



2.5 Posterior predictive checking & Predictive performance

From the plot below, we demonstrate this by plotting the posterior prediction distribution for 2020. It can be seen from the plot that these predictions vary extensively. We also plot the y_{rep} data distributions simulated from the posterior predictive distribution that are used for the R^2 calculation. We clearly see that the deviance of the posterior repetitive distributions is also very large, meaning that the predictive performance is not robust. The R^2-value, that should be close to 1 for a good fit, is also presented and this indicates a poor predictive performance for the model. However, the pooled model is good at giving predictions for the global life expectancy where no differences among regions is taken into account. In fact, the mean of the predictive distribution for the global life expectancy in 2020 is less than one year away from the global life expectancy prediction provided by UN DESA (United Nations Department of Economic and Social Affairs). [3]



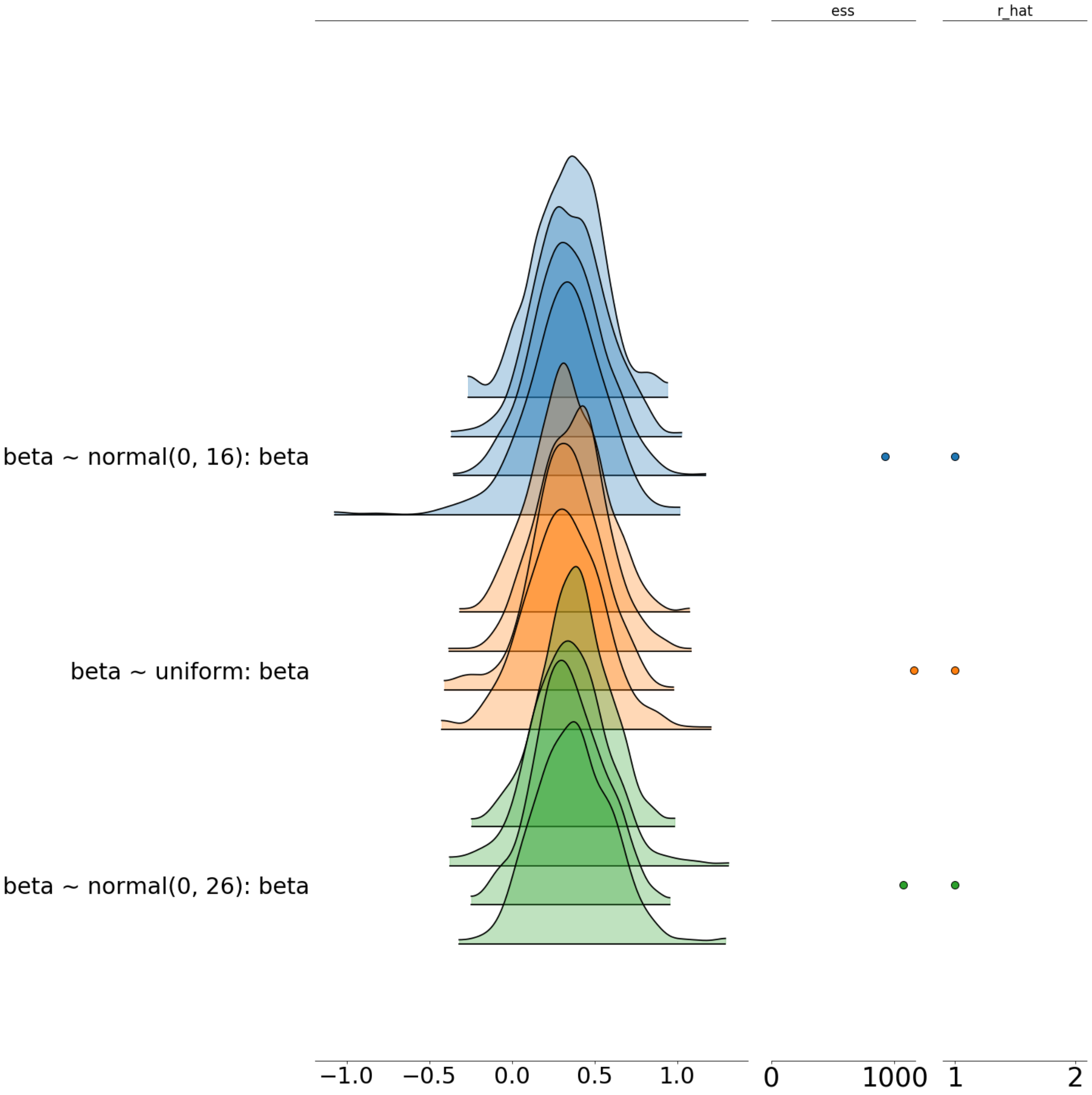
```
az.r2_score(pooled_data_dict["y"], np.mean(pooled_fit["yrep"], axis=0))
```

```
## r2        0.079947
## r2_std    0.000000
## dtype: float64
```

2.6 Sensitivity analysis

We will run the pooled model with different priors for the slope parameter to estimate to robustness of the model. The priors used here were specified and justified in the description of the model. The same Stan model was used, but the beta-prior was commented out for the uniform prior and different τ-parameter was used for the last scenario.

As can be seen from the ridge plots below. The model seems to be robust with different prior choices for beta since there is no clear deviation in the posterior distributions with different models. Also based on the R_hat values (all are equal to 1), all the models seem to converge just fine. Interestingly though, the ESS is the highest for the model with uniform prior. So based on this the convergence is best when the beta prior is set to non-informative uniform prior. However the other prior choices had ESS values of sufficient size also and there is no reason to believe that there were problems with the convergence of the models.



3 The hierarchical model

3.1 The description of model

3.1.1 The model

By using the hierarchical model, we are able to consider the natural hierarchical structure of observed data. We believe that both rate and the baseline between the continents vary. This model lets slopes and the intersects vary on group-level. On the other hand, modeling each continent separately might be too radical. A hierarchical model is an intermediate solution where the level of pooling is determined by the data and a prior on the amount of pooling.

$$y_{nj} = \alpha_j + \beta_j x_{nj} + \epsilon_{nj}$$

y_{nj} refers to the life expectancy on the dependent variable (year) for an individual observation at level 1.

x_{nj} refers to the level 1 predicator.

α_j refers to the intercept of the dependent variable in group j.

β_j refers to the slope for the relationship in group j between the level 1 predictor.

ϵ_{ij} represents the random errors of prediction for the level 1 equation.

3.1.2 The prior choices

We will be modelling both the intercept and slope parameters with normal distributions. However, since this model is hierarchically grouped to regions, the groups will be sharing priors for both of the parameters.

The slope parameter's normal distribution has a standard deviation distributed according to a $\beta \sim Scale - inv - \chi^2(\nu, \tau^2)$ (scaled inverse chi squared distribution) with parameters $\nu = 4$ and $\tau^2 = 1$. This prior restricts the standard deviation to be positive. While most of the mass of this distribution is around 1, the tail of the distribution is still fat enough to allow enough deviance from this value. The distribution still restricts the group level variance in slopes to be quite small which is desired because the slopes should not deviate drastically between groups. The mean for the slope distribution is identical to the pooled model and the same justification apply here as well (refer to section 2.I).

The intercept parameter's normal distribution has a standard deviation distributed according to $\alpha \sim Cauchy(x_0, \gamma)$ that is centered at zero and the parameter $\gamma = 10$. This gives out a wide range of possible values for the deviation of the intercept which is desired since the group level variance in intercepts is quite large between groups. The mean comes from a normal distribution centered at zero and $\sigma = 10$. This is a vague prior and used only to make sure there is enough deviance between posterior samples.

All the priors work as weakly-informative ones. This can be verified when we run the model. It can be seen that the priors do not restrict the parameter values too much but they still provide enough structure to the model.

The **sensitivity analysis** was conducted with the same approach as with the pooled model. so the same prior β for each group was changed in an identical manner (refer to section 2.I).

3.2 Stan-model

```
the_hier_model = """
    data {
        int<lower=0> N;
        vector[N] x;
        vector[N] y;
        int<lower=0> n_region;
        int<lower=1, upper=n_region> region_enc[N];
        real xpred;
        real tau;
    }
    parameters {
        real<lower=0> sigma;
        real global_mu_alpha;
        real<lower=0> global_sd_alpha;
        real global_mu_beta;
        real<lower=0> global_sd_beta;

        vector[n_region] alpha_region;
        vector[n_region] beta_region;
    }

    model {

        global_mu_alpha ~ normal(0, 10);
        global_sd_alpha ~ cauchy(0, 10);

        global_mu_beta ~ normal(0, tau);
        global_sd_beta ~ scaled_inv_chi_square(4, 1);

        for (c in 1:n_region) {
              alpha_region[c]~ normal(global_mu_alpha, global_sd_alpha);
              beta_region[c] ~ normal(global_mu_beta, global_sd_beta);
        }

        sigma ~ normal(0, 10);  // prior for error
        y ~ normal(alpha_region[region_enc] + beta_region[region_enc].*x, sigma);
    }
    generated quantities {
        matrix[N, n_region] yrep;
        vector[n_region] ypred;
        for (k in 1:n_region) {
            for(n in 1:N) {
                yrep[n, k] = normal_rng(alpha_region[k] + beta_region[k]*x[n], sigma);
            }
            ypred[k] = normal_rng(alpha_region[k] + beta_region[k]*xpred, sigma);
        }

    }

"""
```

3.3 Running the Stan-model

```
hier_data_dict = {}
hier_data_dict["N"] = 30
hier_data_dict["x"] = np.asarray(df["Year"].values, dtype=int)
hier_data_dict["y"] = np.asarray(df["Both sexes"].values, dtype=float)
hier_data_dict["n_region"] = 6
hier_data_dict["xpred"] = 2020
hier_data_dict["tau"] = 16
region = df["WHO region"] # Region encoding
region = region.replace("Africa", 1)
region = region.replace("Americas", 2)
region = region.replace("South-East Asia", 3)
region = region.replace("Europe", 4)
region = region.replace("Eastern Mediterranean", 5)
region = region.replace("Western Pacific", 6)
hier_data_dict["region_enc"] = region

hier_fit = pystan.stan(model_code = the_hier_model, data = hier_data_dict,
                       iter = 4000, chains = 4, control=dict(max_treedepth=15))
```

3.4 Convergence diagnostics

3.4.1 HMC/NUTS specific diagnostics

The run did not produce *divergent transitions*, therefore we did not have to perform any step size tuning. This means that the step sizes are of sufficient sizes, the discrete approximation of the HMC/NUTS numerical integral works and that the sampler is drawing samples from the entire posterior resulting in non-biased inference.

```
h = hier_fit.to_dataframe(diagnostics=True)
print('divergent transitions: ', any(h['divergent__']))
```

```
## divergent transitions:  True
```

Similarly to the pooled model, we increased the maximum treedepth parameter to 15. This was done to let the sampler explore the posterior distribution in more depth. The increase was sufficient as can be seen from the print. This means that the optimal number of steps to take in each iteration is less than the maximum which is desired for efficiency.

```
print('treedepth quantiles for draws:\n', h['treedepth__'].quantile([0, 0.25, 0.5, 0.75, 1]))
```

```
## treedepth quantiles for draws:
##  0.00     2.0
## 0.25    11.0
## 0.50    11.0
## 0.75    12.0
## 1.00    15.0
## Name: treedepth__, dtype: float64
```

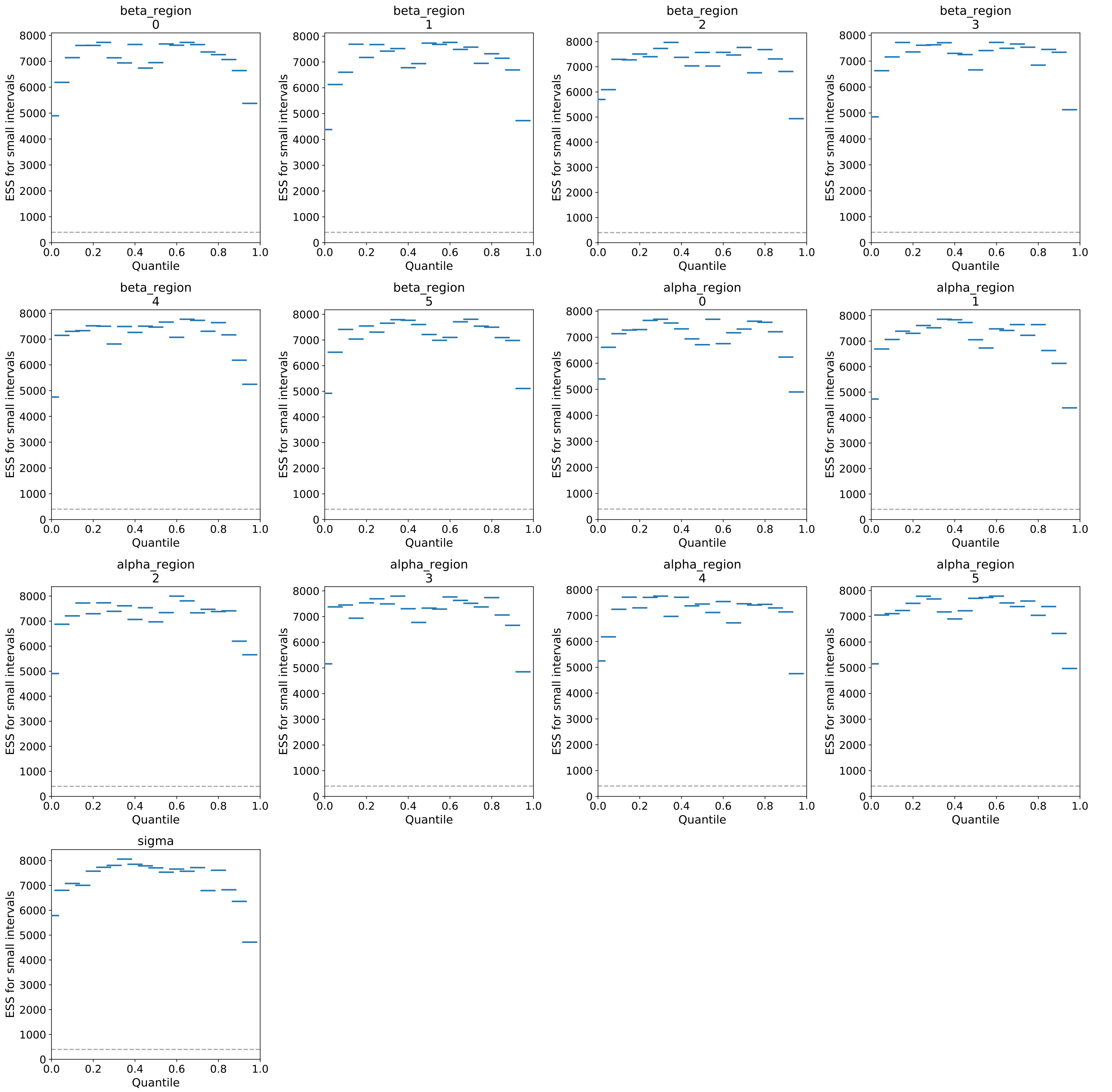
3.4.2 Rhat and ESS diagnostics

Identically to the pooled model, the **Rhat** values for each of the parameters is equal to 1 and the **ESS** values clearly exceed the 100*n_chains criteria. Similarlrly to the pooled model, the ESS never falls below the thin vertical line in the ESS-local-plot which is the minimum for convergence. Also the trace plots seem reasonable and just by eyeballing them the convergence seems obivous. Again, based on these, it is clear that the model has converged properly. (The plots start from the next page).

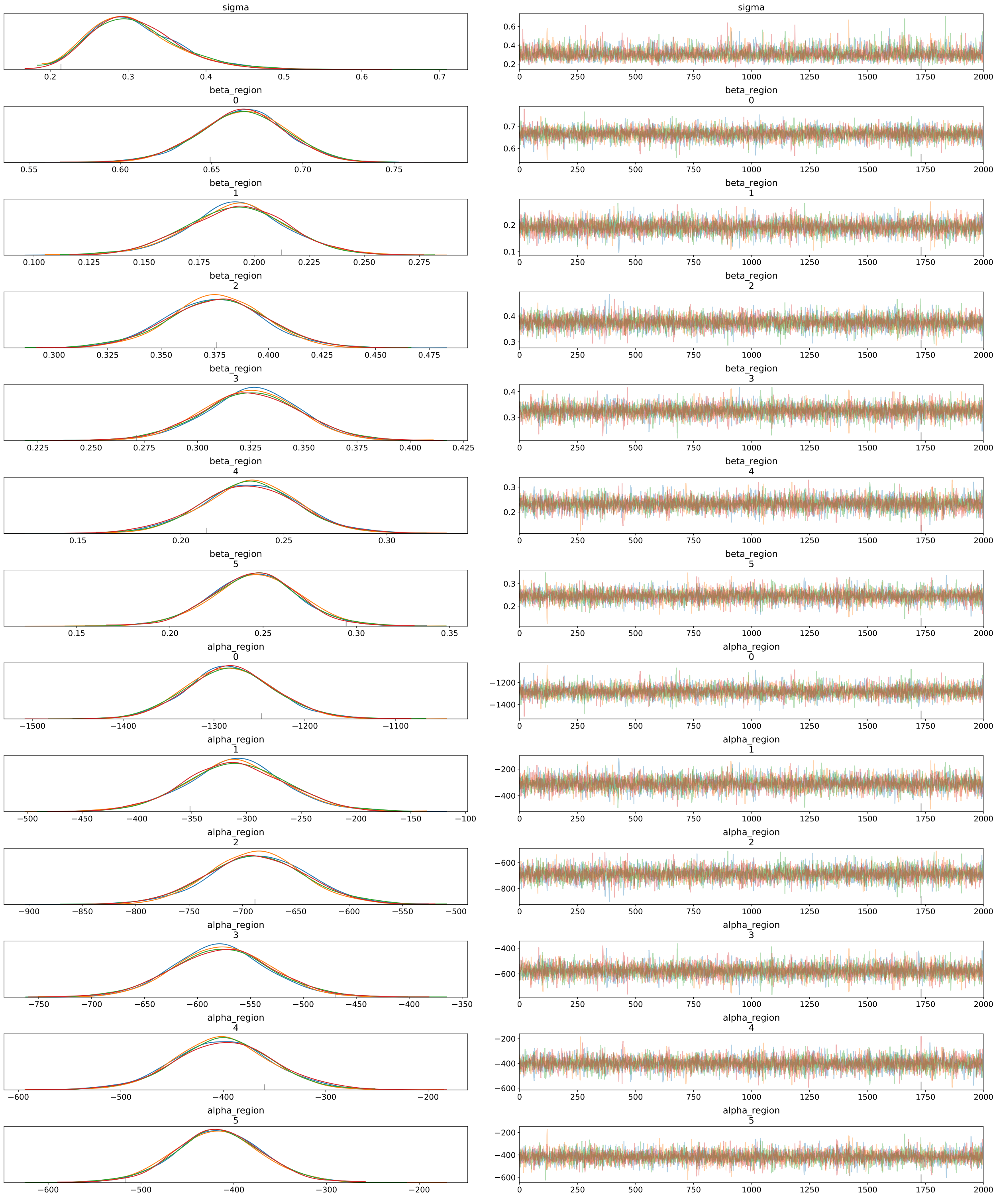
```
print(hier_fit.stansummary(pars=['sigma', 'alpha_region', 'beta_region']))
```

```
## Inference for Stan model: anon_model_0579b78437fdbb935fc466821f0aa85d.
## 4 chains, each with iter=4000; warmup=2000; thin=1;
## post-warmup draws per chain=2000, total post-warmup draws=8000.
##
##                    mean se_mean     sd   2.5%    25%    50%    75%  97.5%  n_eff   Rhat
## sigma              0.31  9.3e-4   0.06   0.22   0.27    0.3   0.34   0.44   3689    1.0
## alpha_region[1]  -1281    0.61  47.39  -1371  -1313  -1282  -1252  -1185   6119    1.0
## alpha_region[2] -310.3    0.62  45.67 -401.0 -339.8 -310.9 -280.9 -218.3   5357    1.0
## alpha_region[3] -687.5    0.56  45.28 -777.2 -716.8 -688.2 -657.8 -596.8   6454    1.0
## alpha_region[4] -577.5    0.54  45.74 -666.7 -607.5 -577.7 -548.3 -485.8   7111    1.0
## alpha_region[5] -400.8    0.59  47.03 -495.0 -431.3 -400.9 -370.5 -306.5   6443    1.0
## alpha_region[6] -417.2    0.56  46.93 -510.3 -447.0 -417.9 -387.2 -323.3   6998    1.0
## beta_region[1]     0.67  3.0e-4   0.02   0.62   0.65   0.67   0.68   0.71   6121    1.0
## beta_region[2]     0.19  3.1e-4   0.02   0.15   0.18   0.19   0.21   0.24   5357    1.0
## beta_region[3]     0.38  2.8e-4   0.02   0.33   0.36   0.38   0.39   0.42   6457    1.0
## beta_region[4]     0.32  2.7e-4   0.02   0.28   0.31   0.33   0.34   0.37   7110    1.0
## beta_region[5]     0.23  2.9e-4   0.02   0.19   0.22   0.23   0.25   0.28   6444    1.0
## beta_region[6]     0.25  2.8e-4   0.02    0.2   0.23   0.25   0.26   0.29   6998    1.0
##
## Samples were drawn using NUTS at Sun 08 Dec 2019 19:30:30 .
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
_ = az.plot_ess(
      hier_fit,
      var_names=["beta_region", "alpha_region", "sigma"],
      kind="local",
      marker="_",
      ms=20, mew=2,
      figsize=(20, 20)
)
plt.show()
```

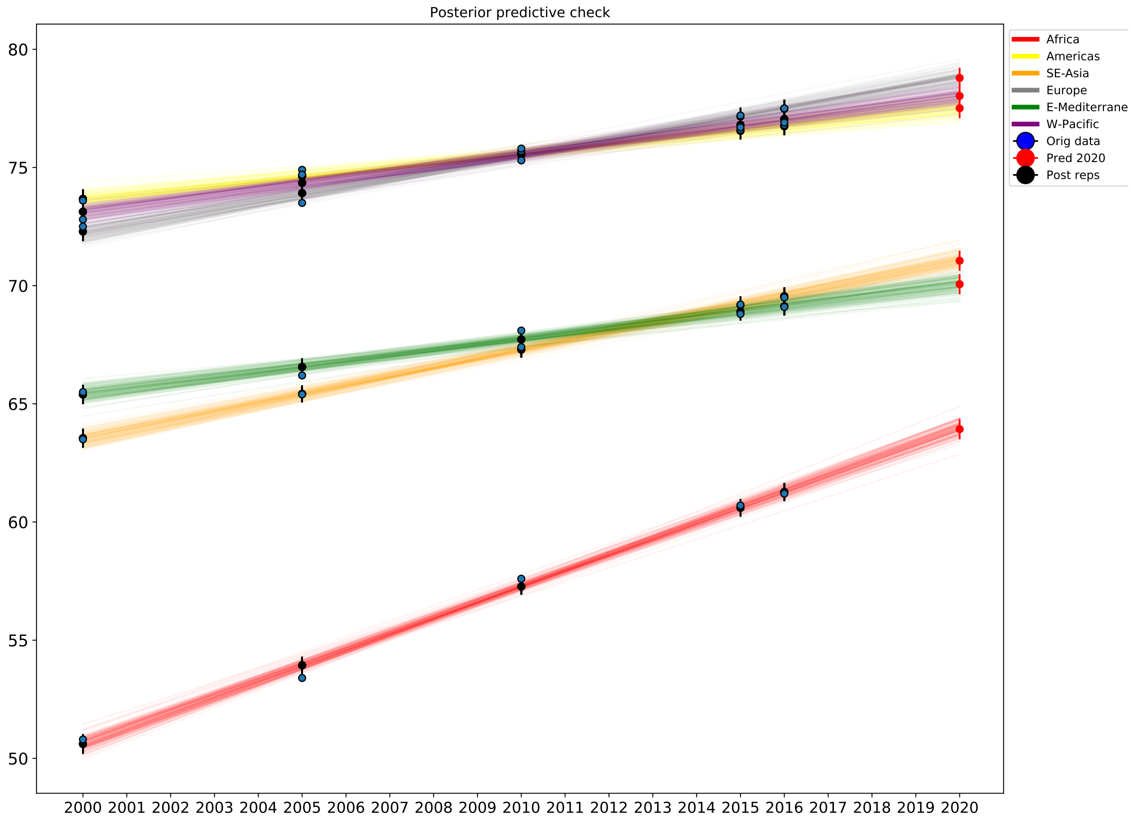


```
_ = az.plot_trace(
      hier_fit,
      var_names = ["sigma", "beta_region", "alpha_region"],
      figsize=(25, 30)
)
plt.show()
```


sigma
0.2
0.3
0.4
0.5
0.6
0.7
0.2
0.4
0.6
0
250
500
750
1000
1250
1500
1750
2000
beta_region
0
0.55
0.60
0.65
0.70
0.75
0.6
0.7
beta_region
1
0.100
0.125
0.150
0.175
0.200
0.225
0.250
0.275
0.1
0.2
beta_region
2
0.300
0.325
0.350
0.375
0.400
0.425
0.450
0.475
0.3
0.4
beta_region
3
0.225
0.250
0.275
0.300
0.325
0.350
0.375
0.400
0.425
0.3
0.4
beta_region
4
0.15
0.20
0.25
0.30
0.2
0.3
beta_region
5
0.15
0.20
0.25
0.30
0.35
0.2
0.3
alpha_region
0
−1500
−1400
−1300
−1200
−1100
−1400
−1200
alpha_region
1
−500
−450
−400
−350
−300
−250
−200
−150
−100
−400
−200
alpha_region
2
−900
−850
−800
−750
−700
−650
−600
−550
−500
−800
−600
alpha_region
3
−750
−700
−650
−600
−550
−500
−450
−400
−350
−600
−400
alpha_region
4
−600
−500
−400
−300
−200
−600
−400
−200
alpha_region
5
−600
−500
−400
−300
−200
−600
−400
−200

3.5 Posterior predictive checking & Predictive performance

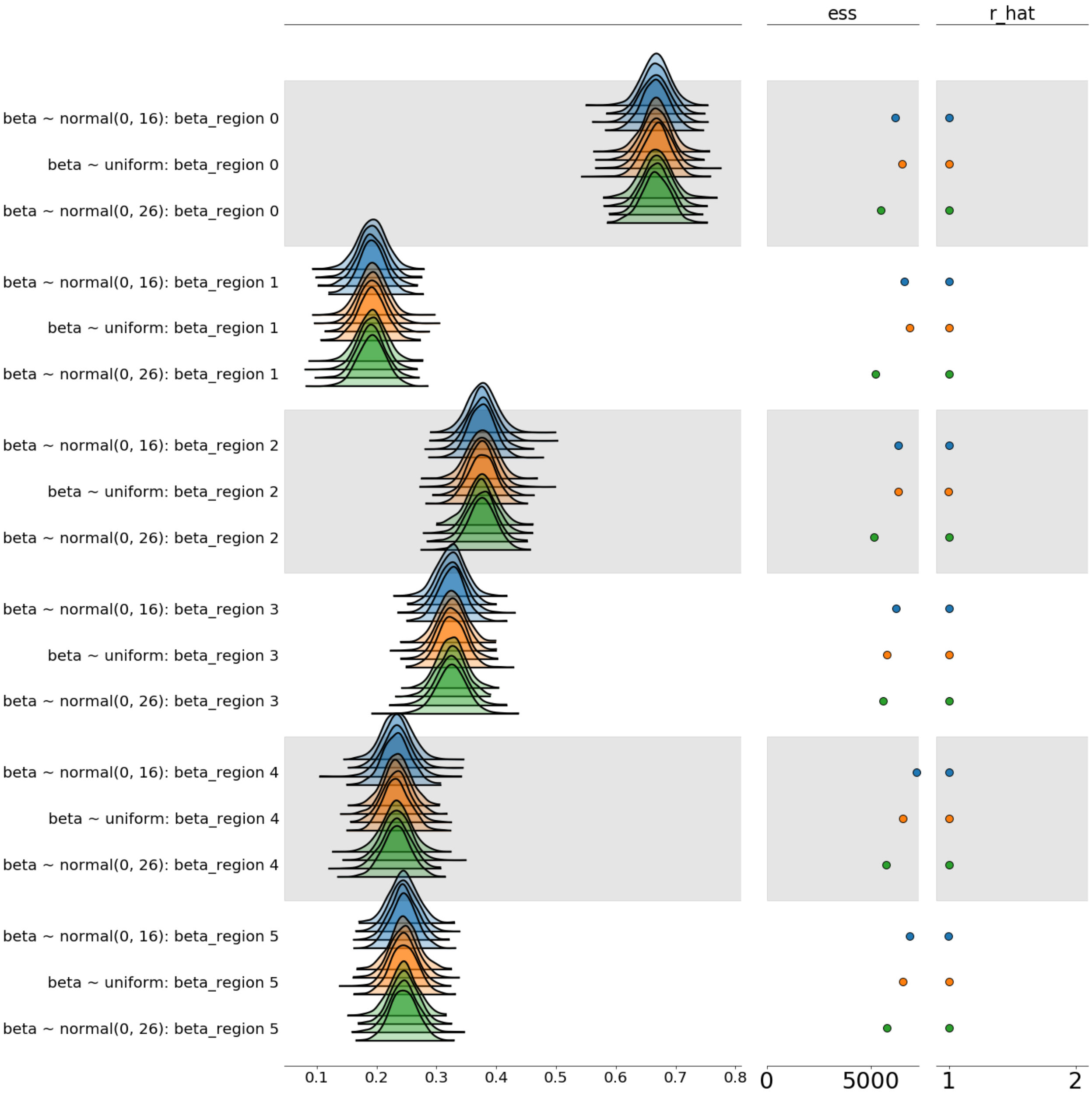
From the plot below, we clearly see that the predictions for each group seem reasonable. Again we demonstrate this by plotting the prediction of life expectancy for each region in 2020. It can be seen from the plot that these are very reasonable predictions. Especially in comparison to the pooled model, where the deviance between different predictions is huge. We also plot the y_{rep} data distributions simulated from the posterior predictive distribution that are used for the R^2 calculation. We clearly see that the deviance the posterior repetitive distributions is almost neglible, meaning that the predictive performance is a lot better with the hierarchical model. Below, the R^2-values are also presented and these too indicate a substantial increase in the predictive performance of the hierarchical model in comparison to the pooled model.



```
##            Africa  Americas  E Mediter    Europe   SE Asia  W Pacific
## r2       0.994469   0.97568    0.86701  0.987599  0.722199   0.972311
## r2_std   0.000000   0.00000    0.00000  0.000000  0.000000   0.000000
```

3.6 Sensitivity analysis

As can be seen from the forest plots for each group below The model seems to be robust with different prior choices for different beta_regions, since there is no clear deviation in the posterior distributions with different models. Also based on the Rhat (all are equal to 1) and ESS values, all the models seem to converge just fine with different prior settings.



4 Comparing the models

First we compare the models based on the diagnostics and convergence results. Secondly, we evaluate the model performances overall and comment on their applicability in our problem. We refer to hierarchical model as H-model and pooled model as P-model.

4.1 Convergence

Both models perform neatly in terms of convergence. All the parameters obtained, in both cases, $\hat{R} = 1$, which implies convergence. In addition to the quantitive assessment, one can see visually how the chain simulations are stable and the sequences are overlapping (Look at the traceplots 2.IV and 3.IV). We conclude that the mixing and stationarity conditions for robust convergence hold for both models.

4.2 Posterior predictiveness

The R^2 score that is used to evaluate how close the data are to the fitted regression line, was poor for the pooled one. The P-model was not able explain to what extent the variance of year explains the variance of the life expectancy. The H-model outperformed the P-model clearly in this category as we compare the R^2 scores listed previously. This is also seen in the posterior predictive checking plots above, where the error plot around the posterior estimates is much wider for P-model than H-model (Look at plots 2.V and 3.V). Therefore, we conclude that the predictiveness performance is extensively better for H-model.

4.3 Sensitivity to chosen priors

As stated previously in the sensitivity analysis parts. Both models are concluded to be robust so there are no clear winner in this part.

4.4 Applicability to our problem

We saw that the P-model was very applicable when predicting the global life expectancy. Obivously this happens because the averaging effect of the parameter pooling. The H-model is suitable when the multilevel structure of life-expectancy data is taken into account since clearly there are differences in life expectancy between regions in the world. The hierarchical approach should be the better one here since the regional differences are real.

Furthermore, the H-model includes the superpopulation for the parameters (slope, intercept). This lets us sample the parameters for an unknown region if needed. We believe that sampling from the superpopulation distributions, one could captivate the behaviour of the global life expectancy trends more deeply. This cannot be usually done with the most of the other approaches.

5 Conclusion

5.1 Potential improvements

- We could add another level to the model i.e. instead of modelling the overall life expectancy between regions, we could also model the life expectancy between regions and for each region, between genders. It would be interesting to investigate wether or not there are any differences between the genders and the reasons behind the results.
- Studying the dependence between the slope and intercept could be crucial as it might turn out that they are better modeled together than separately. This makes sense, especially in our case: The lower the intercept, the higher the rate of the life expectancy. The regions that have lower baseline due to lower living standards, education or lack of quality health services might be reaching higher life expectancies much faster than the more developed regions. This type of phenomenon is in fact visible in the plot 3.V.
- The sensitivity analysis should be conducted more robustly and more priors for each of the parameters should be tested.

5.2 Main findings

The benefits of the each model depend strongly on the context. If the researcher is happy with a crude estimation of life expectancy, on average, around the globe, then the pooled model is the most straightforward option. Using the bayesian framework is still highly recommended as it allows the researcher to take the uncertainty regarding the parameters into account in a very intuitive way. However, investing time in building a hierarchical model let the researcher to unravel trends more deeply on the region level - yet obtaining some overall picture from the global level trend. We believe this approach could be very powerful tool for organization like WHO as they tend to work with data that has a natural hierarchical sturcture to some extent. In addition to that, WHO should be interested in the underlying populations as they can reveal surprising things and provide great channels for overall monitoring.

This project was a great pedagogical experience and a very compelling way to let us apply our bayesian statistical skills.

6 References

[1] https://cran.r-project.org/web/packages/bayesplot/vignettes/visual-mcmc-diagnostics.html

[2] https://www.displayr.com/what-is-effective-sample-size/

[3] https://www.statista.com/statistics/673420/projected-global-life-expectancy/

[4] https://jrnold.github.io/bayesian_notes/mcmc-diagnostics.html