BDA - Project

Anonymous

Contents

1. Introduction

Education plays a role in developing the vocabulary of people, but exactly how much? If someone studied longer than someone else, does this mean that most likely they have a more developed vocabulary? A research done by University of Reading [1] found that there is indeed a positive linear relationship between the number of years someone spends in education (primary school, high school, university) and the development of their vocabulary. A standardize test was used to assess the vocabulary of the participants of both genders and their number of years spent on education (from 0 to 20) was recorded as well. The vocabulary test consists of 10 questions.

Even thought this relationship has been shown to exist, we are wondering whether or not persons with a specific educational level have a highly varying level of vocabulary, or if the level of education determines the level of vocabulary very strongly. For example, there could exist two different groups of people with a lower educational level. One group chose to learn a trade on their own and developed a strong vocabulary despite of their education, where another group stopped their education and did not develop their vocabulary further. The same could hold for people with a higher education. Some may end up in high positions in government or industry and need to possess a very high level of vocabulary, while other may have chosen a specific trade where vocabulary is much less important. The mean value of vocabulary does not reveal this variance and we were not able to find a study that analysed this.

We aim to find out how strongly the level of education determines your vocabulary. Furthermore, we will compare the results for each education level and give several possible explanations for the results.

2. Problem Description & Approach

A previous study has shown that there exists a positive relationship between the number of years that someone spends in education and the strength of a person's vocabulary. However, this study did not find if there are large varianced between people with a specific number of education years. These variances are interesting, since a large variance may indicate that education does not strictly determine the level of vocabulary someone has. On the other hand, a small variance of the vocabulary of a certain education level may indicate that it is much harder to reach a stronger vocabulary without spending more years in education.

To find out how large these variances are, we will be using the same dataset as in the beforementioned study. Firstly, we need to decide how to group the participants with a certain education level together, since there are very few participants that have a very low or a very high educational level. After we identified the variances of the vocabulary strength for all educational levels, we will choose a suitable prior and construct two different posterior models.

We will compare these models and perform predictive checking and performance assessment. Lastly, we will identify how our prior choise affects our posterior and we will end with a discussion about our results and approach.

3. Data Preprocessing

Our data consists of 6 columns where every row corresponds to a unique person. In total there are 30351 persons. For each person there is an ID (first 2 columns), a year in which the vocabulary test was conducted, the sex of the person, the number of years the person spent in education (primary school, high school and university) and the vocabulary score (a number between 1 and 10). The vocabulary score is based on a 10-word test that the participant did in the given year and each person only did the test once.

At first glance, there are a number of possible problems with this dataset. First of all, the 10-word test was different each year, which means there could be variance in scores between the years. Furthermore, the number of males and females could be different between the years the test was conducted. Another problem could arise if there are very few samples from people with very low or high education levels.

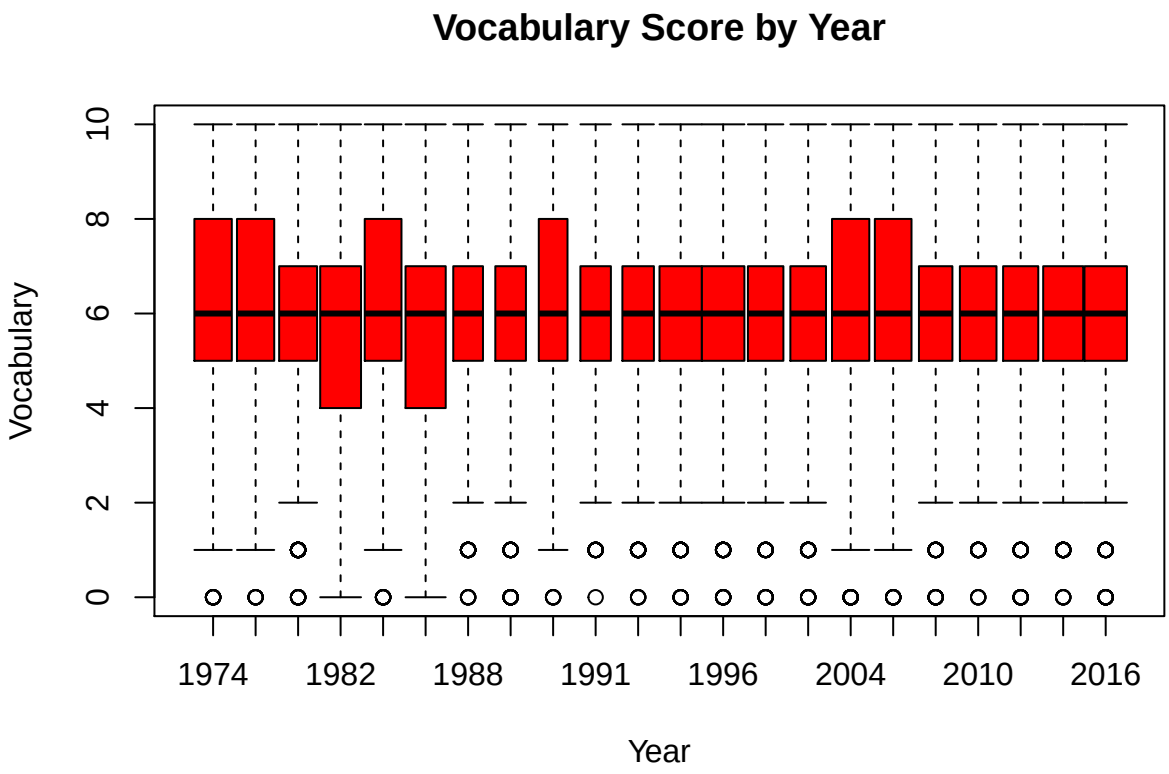
Firstly we will identify if the 10-word tests differed substantially throughout the years. To do this, we visualize the results of the tests with a boxplot. Data preparation

According to source [2]

```
#create summary variables
vocab <- read.csv("vocab.csv")
#frequency by year
FreqByYear<-count(vocab,'year')
names(FreqByYear)[2]<-"ResponsesPerYear"

#percentage variable (composition) by sex and year
FreqBySexYear<-count(vocab,c("year","sex"))
names(FreqBySexYear)[3]<-"ResponsesPerSexYear"
Temp<-join(FreqBySexYear,FreqByYear,by='year',type="left")
Temp$SexPerYear.percent<-round((Temp$ResponsesPerSexYear/Temp$ResponsesPerYear)*100,3)

#create variables in dataset
vocab<-join(vocab,Temp,by=c('year','sex'),type="left")

#create total count and percentage variables
FreqSampleSex<-count(vocab,"sex")
```

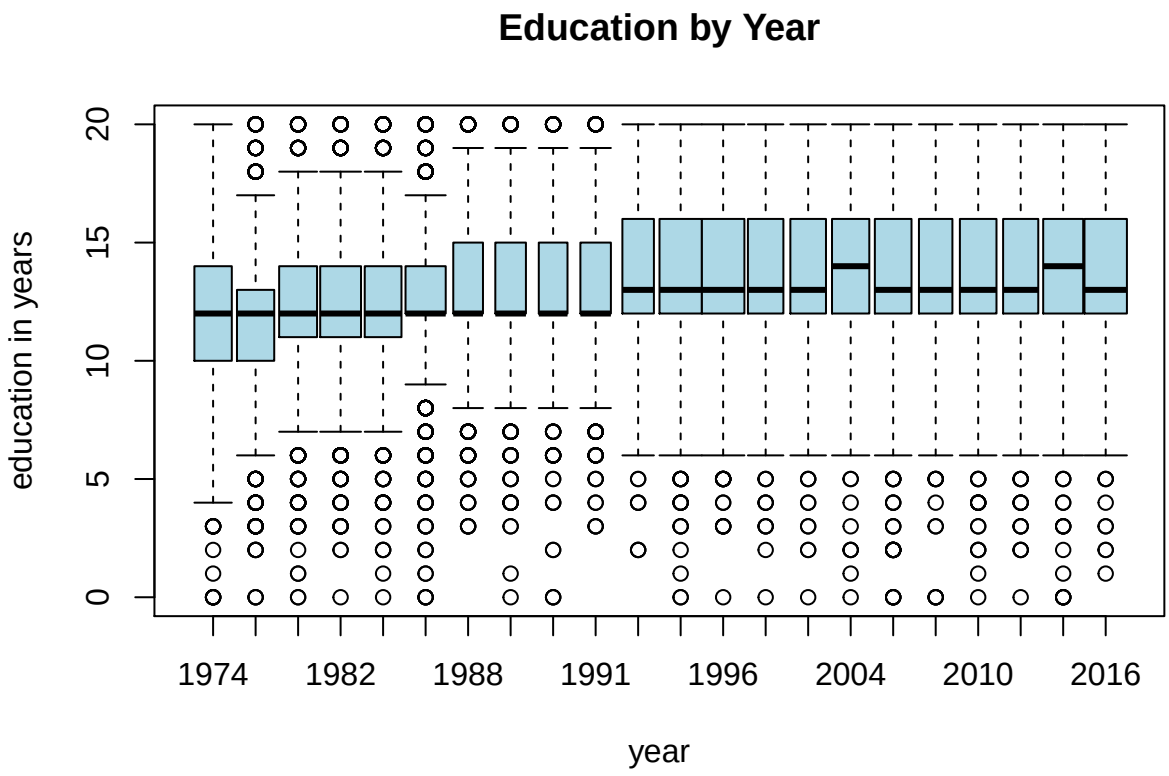
```
FreqSampleSex$FreqSampleSex.percent<-round((FreqSampleSex$freq/sum(FreqSampleSex$freq))*100,3)
names(FreqSampleSex)[2]<-"FreqSampleSex.freq"
#merge with vocab
vocab<-join(vocab,FreqSampleSex,by="sex",type="left")

#add mean education and score by year for each sex.
MeanEducBySexYr<-aggregate(vocab$education,by=list(vocab$sex,vocab$year),FUN=mean,na.rm=TRUE)
names(MeanEducBySexYr)[3]<-"MeanEducBySexYr"
names(MeanEducBySexYr)[2]<-"year"
names(MeanEducBySexYr)[1]<-"sex"
vocab<-join(vocab,MeanEducBySexYr,by=c('year','sex'),type="left")

#create variable for mean education by year
MeanEducByYr<-aggregate(vocab$education,by=list(vocab$year),FUN=mean,na.rm=TRUE)
names(MeanEducByYr)[2]<-"MeanEducByYr"
names(MeanEducByYr)[1]<-"year"
vocab<-join(vocab,MeanEducByYr,by='year',type="left")

#create variable for add mean education and score by year for each sex.
MeanVocabBySexYr<-aggregate(vocab$vocabulary,by=list(vocab$sex,vocab$year),FUN=mean,na.rm=TRUE)
names(MeanVocabBySexYr)[3]<-"MeanVocabBySexYr"
names(MeanVocabBySexYr)[2]<-"year"
names(MeanVocabBySexYr)[1]<-"sex"
vocab<-join(vocab,MeanVocabBySexYr,by=c('year','sex'),type="left")

#create variable for mean vocabulary score by year
MeanVocabByYr<-aggregate(vocab$vocabulary,by=list(vocab$year),FUN=mean,na.rm=TRUE)
names(MeanVocabByYr)[2]<-"MeanVocabByYr"
names(MeanVocabByYr)[1]<-"year"
vocab<-join(vocab,MeanVocabByYr,by='year',type="left")

#round all values to 3 significant digits.
is.num<-sapply(vocab,is.numeric)
vocab[is.num]<-lapply(vocab[is.num],round,3)
```

```
boxplot(vocabulary~year,
        data=vocab,
        main="Vocabulary Score by Year",
        xlab="Year",
        ylab="Vocabulary",
        varwidth=T,
        col="red",
        pars = list(boxwex = 1, staplewex = 1, outwex = 1))
```



As can be seen, the test scores stayed mostly the same throughout the years. This does not mean that the vocabulary strength stayed the same over this time period, but since we are only focused on the relative test-score variance, this dataset seems appropriate.
However mean education in years slightly increased which can be seen from below.

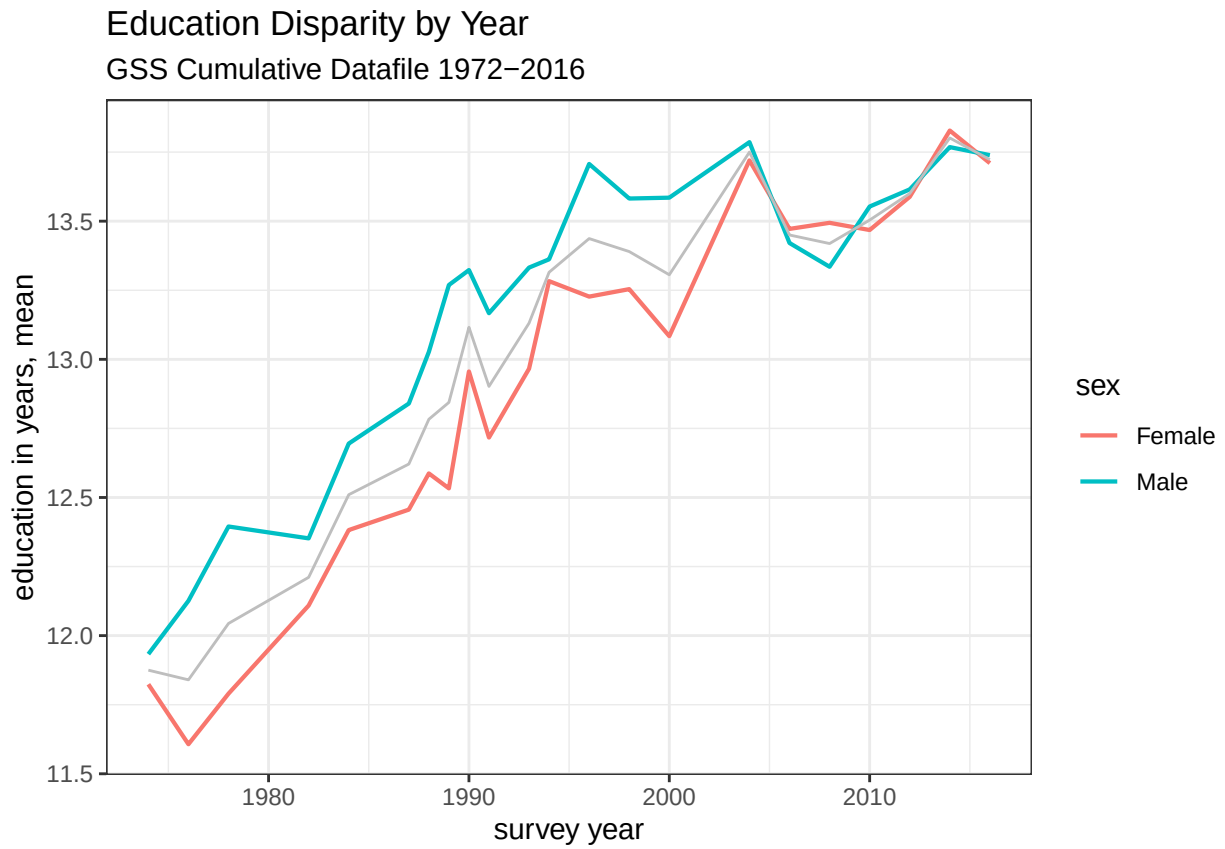
```
boxplot(education~year,
        data=vocab,
        main="Education by Year",
        xlab="year",
        ylab="education in years",
        varwidth=T,
        col="lightblue",
        pars = list(boxwex = 1, staplewex = 1, outwex = 1))
```



But we see that even if length of education increased, the mean vocabulary scores didnot vary!

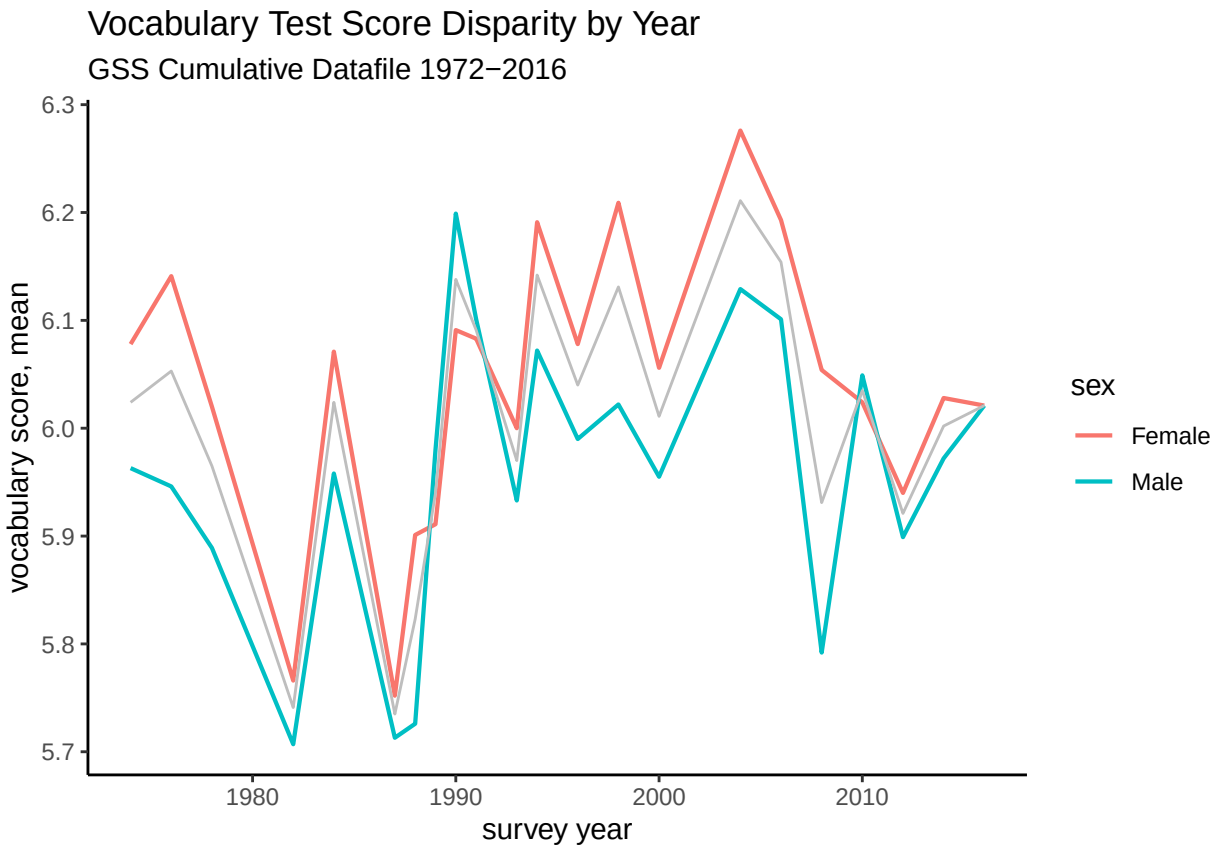
Also, we would like to know if the education length and vocabulary scores of male and female participants varied throughout time.

```
plot2<-ggplot(vocab, aes(x=year, y=MeanEducBySexYr, color=sex))+
  geom_line(size=.75)+
  stat_summary(aes(y=MeanEducByYr,group=1),fun.y=mean,color="gray",geom="line",group=1) +
  #scale_color_wsj("colors6")+
  theme_bw()+
  ggtitle("Education Disparity by Year",subtitle="GSS Cumulative Datafile 1972-2016")+
  xlab("survey year")+
  ylab("education in years, mean")
suppressWarnings(print(plot2))
```



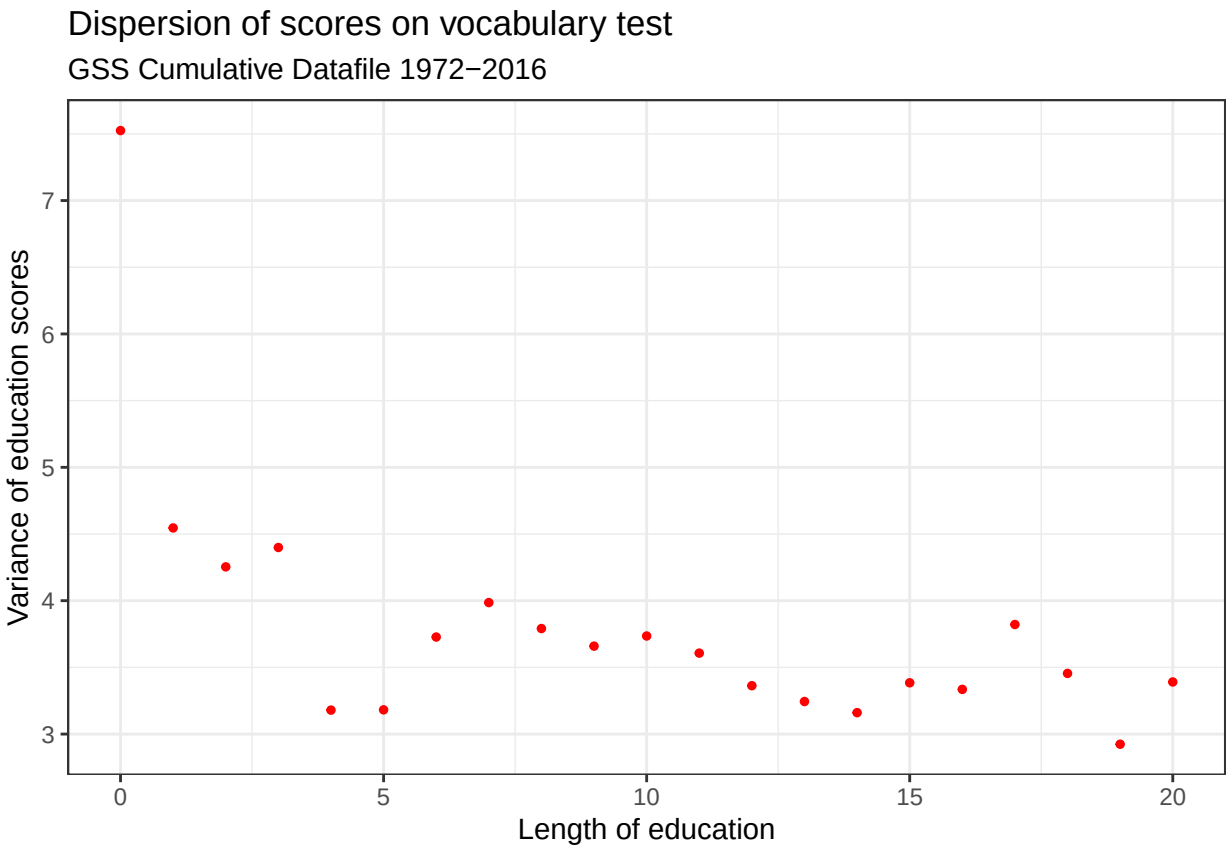
We see the difference, where man had more years of education in 1972-2004 and then the reverse is observed, however that difference is not drastical.
And finally it is interesting to see the scores obtained by male and female. Below is result:

```
plot3<-ggplot(vocab, aes(x=year, y=MeanVocabBySexYr, color=sex))+
  geom_line(size=.75)+
  stat_summary(aes(y=MeanVocabByYr,group=1),fun.y=mean,color="gray",geom="line",group=1)+
  #scale_color_wsj("colors6")+
  theme_bw()+
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))+
  ggtitle("Vocabulary Test Score Disparity by Year",subtitle="GSS Cumulative Datafile 1972-2016")+
  xlab("survey year")+
  ylab("vocabulary score, mean")

suppressWarnings(print(plot3))
```



The graph shows that women scored more than men. Even having less education in general. We want to investigate if the varince of scores obtained depends on the length of the education. For example if people with education of 12 years can have both score 3 and 10, or people with education of 5 years have small variance of scores (for example only in the range 2-5)? We plot the relationship between variance and years spent on education:

```
test <- aggregate(vocab$vocabulary, by = list(vocab$education), FUN=var, na.rm = FALSE)

plot4<-ggplot() + geom_point(aes(test$Group.1, test$x), data = test, size =1, color = 'red') +
  geom_line(size=.75)+

  #scale_color_wsj("colors6")+
  theme_bw()+
  ggtitle("Dispersion of scores on vocabulary test",subtitle="GSS Cumulative Datafile 1972-2016")+
  xlab("Length of education")+
  ylab("Variance of education scores")

suppressWarnings(print(plot4))
```



We see that variance tends to decrease with more education. There is a big outlier for people with 0 education which probably means they had their education at home.

4. Prior Choice

Followin the article on prior selection[3]:

Prior selection: we thought that the pr You think a parameter could be anywhere from 0 to 1, so you set the prior to uniform(0,1). Try normal(.5,.5) instead. So we chose near the mean. And for beta we assume that it doesnt change for 0.3 per year. as seen from the graph.

5. Construction of our Models

We tried to fit diffrent linear models based on the Kilpisjarvi summer temperature examples. We also want to check if the 21 year long education gives
bigger variance in 0.5 variance. Linear Gaussian model with adjustable priors:

```
d_lin <-list(N = nrow(test),
             x = test$Group.1,
             xpred = 21,
             y = test$x)
cat(readLines('lin_pr.stan'), sep='\n')
```

```
## Warning in readLines("lin_pr.stan"): incomplete final line found on
```

```
## 'lin_pr.stan'

## // Gaussian linear model with adjustable priors
## data {
##   int<lower=0> N; // number of data points
##   vector[N] x; //
##   vector[N] y; //
##   real xpred; // input location for prediction
##   real pmualpha; // prior mean for alpha
##   real psalpha;  // prior std for alpha
##   real pmubeta;  // prior mean for beta
##   real psbeta;   // prior std for beta
## }
## parameters {
##   real alpha;
##   real beta;
##   real<lower=0> sigma;
## }
## transformed parameters {
##   vector[N] mu;
##   mu = alpha + beta*x;
## }
## model {
##   alpha ~ normal(pmualpha, psalpha);
##   beta ~ normal(pmubeta, psbeta);
##   y ~ normal(mu, sigma);
## }
## generated quantities {
##   real ypred;
##   vector[N] log_lik;
##   ypred = normal_rng(alpha + beta*xpred, sigma);
##   for (i in 1:N)
##     log_lik[i] = normal_lpdf(y[i] | mu[i], sigma);
## }
```

```
d_lin_priors <- c(list(
    pmualpha = mean(unlist(test$x)), #  centered on 3.793728
    psalpha = 10, # non informative
    pmubeta = 0, # a priori incr. and decr. as likely
    psbeta = (0.5--0.5)/21), # does not increase more than 0.3 for 1 additional year
  d_lin)
fit_lin <- stan(file = 'lin_pr.stan', data = d_lin_priors, seed = 48927)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'C:
## \Users\Dana\Documents\lin_pr.stan'
```

Calculate if the probability that vocabulary score will increase.

```
#monitor(fit_lin, probs = c(0.1, 0.5, 0.9))
samples_lin <- rstan::extract(fit_lin, permuted = T)
mean(samples_lin$beta>0.2) # probability that beta > 0
```

```
## [1] 0
```

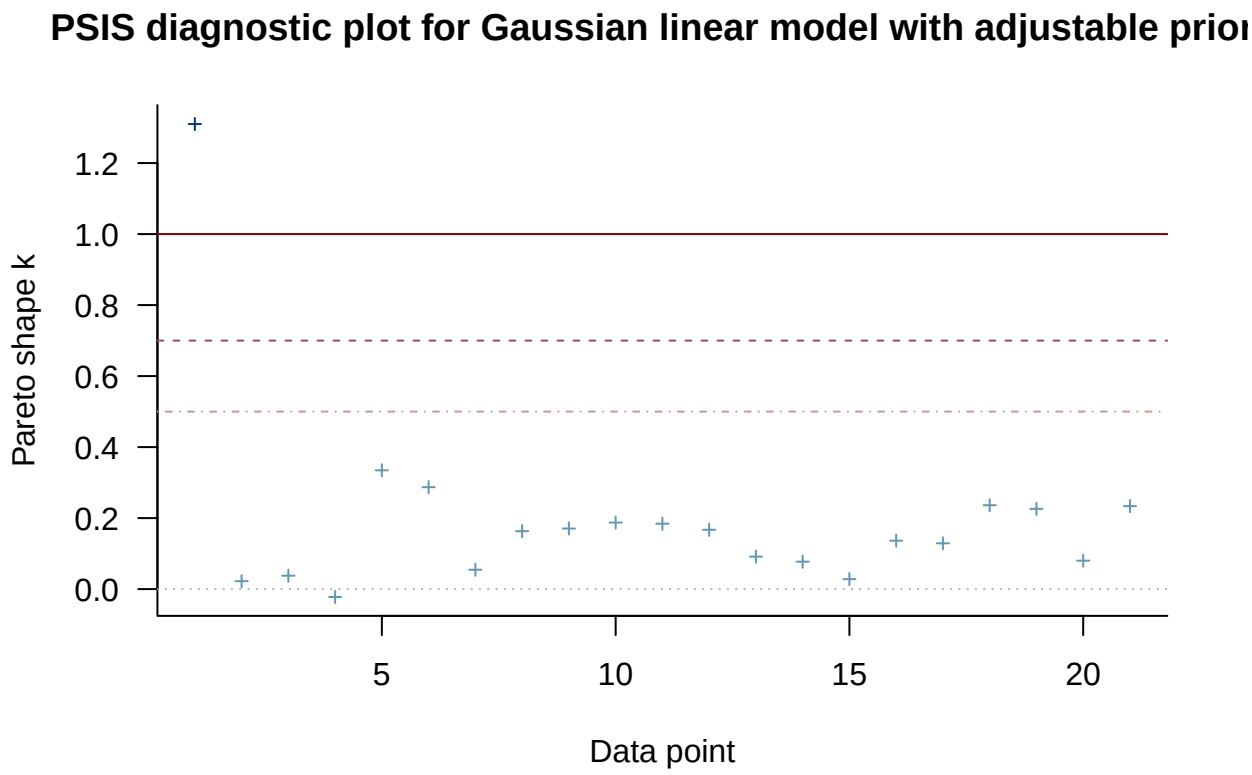
```
g_log_lik <- extract_log_lik(fit_lin, merge_chains = FALSE)
r_eff <- relative_eff(exp(g_log_lik))
g_loo_lin <- loo(g_log_lik, r_eff = r_eff)
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```
g_loo_lin$elpd_loo
```

```
## Warning: Accessing elpd_loo using '$' is deprecated and will be removed in
## a future release. Please extract the elpd_loo estimate from the 'estimates'
## component instead.

## [1] -30.84845
```

```
plot(g_loo_lin, diagnostic = c("k", "n_eff"), label_points = FALSE,
main = "PSIS diagnostic plot for Gaussian linear model with adjustable priors")
```




```
pareto_k_table(g_loo_lin)
```

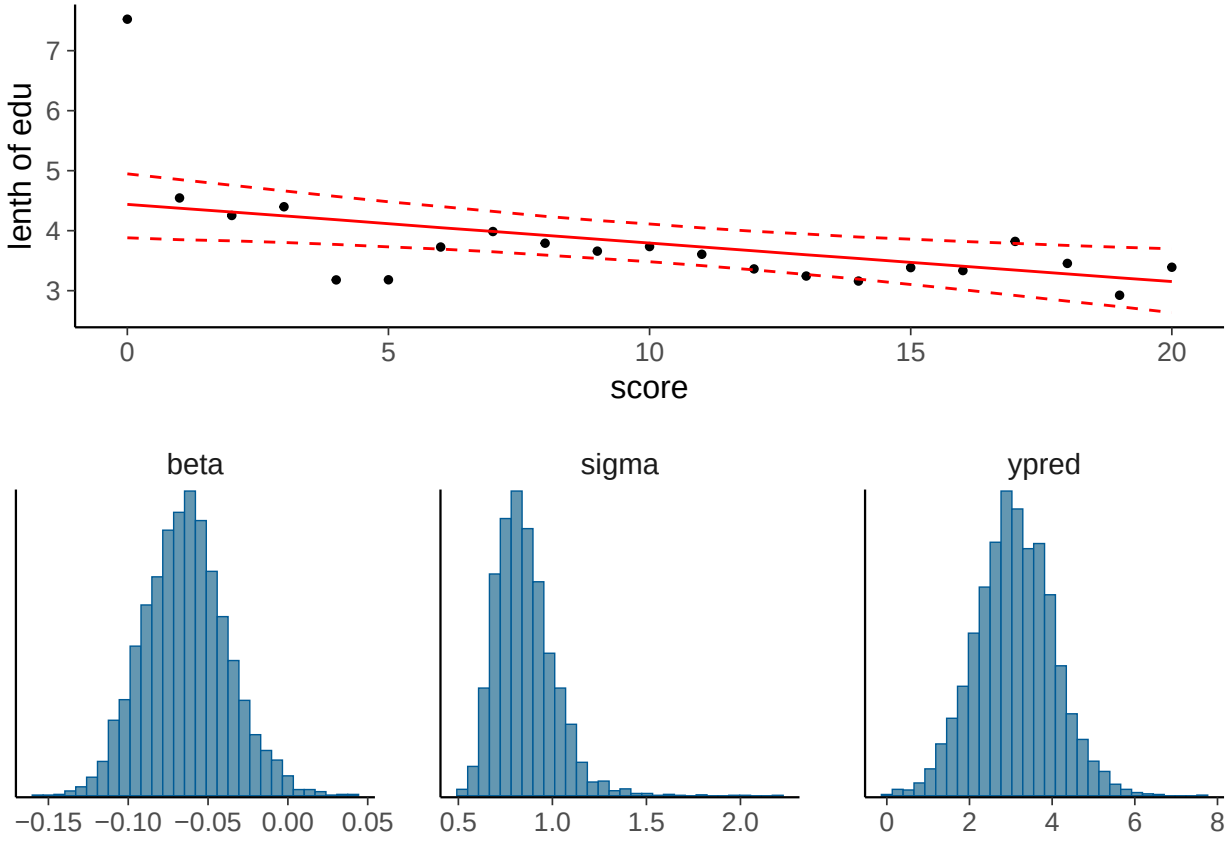
```
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     20    95.2%   1436
##  (0.5, 0.7]   (ok)        0     0.0%   <NA>
##    (0.7, 1]   (bad)       0     0.0%   <NA>
##    (1, Inf)   (very bad)  1     4.8%   5
```

Here we calculate the PSIS-LOO elpd values and the k-values using "loo" function with the calculated log-likelihood value. The interpretation of the k-values is as follows: if all the k-values are below 0.7, the model is reliable. For linear Gaussian it is okay. The k-values can be visualized by plotting the output of the "loo" function.

Now we can plot the model fit and prediction for year 21:

```
mu <- apply(samples_lin$mu, 2, quantile, c(0.05, 0.5, 0.95)) %>%
  t() %>% data.frame(x = d_lin$x, .)  %>% gather(pct, y, -x)
pfit <- ggplot() +
  geom_point(aes(x, y), data = data.frame(d_lin), size = 1) +
  geom_line(aes(x, y, linetype = pct), data = mu, color = 'red') +
  scale_linetype_manual(values = c(2,1,2)) +
  labs(y = 'lenth of edu', x= "score") +
  guides(linetype = F)
pars <- intersect(names(samples_lin), c('beta','sigma','ypred'))
draws <- as.data.frame(fit_lin)
phist <- mcmc_hist(draws, pars = pars)
grid.arrange(pfit, phist, nrow = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Gaussian linear model with standardized data

Before unnormalized data was used. But it can be the case that there is a very strong posterior dependency between alpha and beta, which can be removed by normalizing the data to have zero mean.

```
writeLines(readLines("lin_std_pr.stan"))
```

```
## Warning in readLines("lin_std_pr.stan"): incomplete final line found on
## 'lin_std_pr.stan'
```

```
## // Gaussian linear model with standardized data
## data {
##   int<lower=0> N; // number of data points
##   vector[N] x; //
##   vector[N] y; //
##   real xpred; // input location for prediction
## }
## transformed data {
##   vector[N] x_std;
##   vector[N] y_std;
##   real xpred_std;
##   x_std = (x - mean(x)) / sd(x);
##   y_std = (y - mean(y)) / sd(y);
##   xpred_std = (xpred - mean(x)) / sd(x);
## }
## parameters {
##   real alpha;
##   real beta;
##   real<lower=0> sigma_std;
## }
## transformed parameters {
##   vector[N] mu_std;
##   mu_std = alpha + beta*x_std;
## }
## model {
##   alpha ~ normal(0, 1);
##   beta ~ normal(0, 1);
##   y_std ~ normal(mu_std, sigma_std);
## }
## generated quantities {
##   vector[N] mu;
##   real<lower=0> sigma;
##   real ypred;
##   vector[N] log_lik;
##   mu = mu_std*sd(y) + mean(y);
##   sigma = sigma_std*sd(y);
##   ypred = normal_rng((alpha + beta*xpred_std)*sd(y)+mean(y), sigma*sd(y));
##   for (i in 1:N)
##     log_lik[i] = normal_lpdf(y[i] | mu[i], sigma);
## }
```

```
fit_lin_std <- stan(file = 'lin_std_pr.stan', data = d_lin, seed = 48927)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'C:
## \Users\Dana\Documents\lin_std_pr.stan'
```

```
samples_lin_std <- rstan::extract(fit_lin_std, permuted = T)
mean(samples_lin_std$beta>0.2)
```

```
## [1] 0
```

Checking

```
fit_lin_std_log_lik <- extract_log_lik(fit_lin_std, merge_chains = FALSE)
r_eff <- relative_eff(exp(fit_lin_std_log_lik))

fit_lin_std_loo_lin <- loo(fit_lin_std_log_lik, r_eff = r_eff)
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

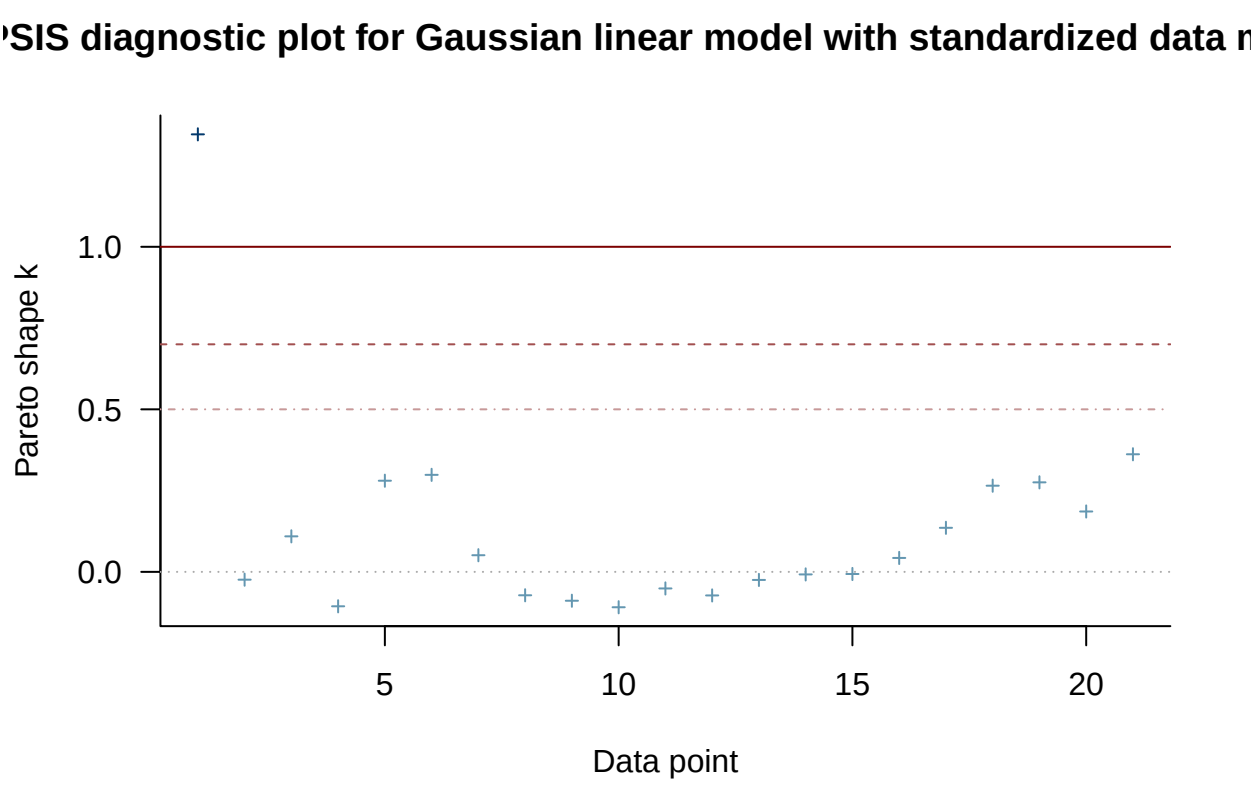
```
## Warning in log(z): NaNs produced
```

```
fit_lin_std_loo_lin$elpd_loo
```

```
## Warning: Accessing elpd_loo using '$' is deprecated and will be removed in
## a future release. Please extract the elpd_loo estimate from the 'estimates'
## component instead.
```

```
## [1] -30.77337
```

```
plot(fit_lin_std_loo_lin, diagnostic = c("k", "n_eff"), label_points = FALSE,
main = "PSIS diagnostic plot for Gaussian linear model with standardized data model")
```




```
pareto_k_table(fit_lin_std_loo_lin)
```

```
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     20    95.2%   2177
##  (0.5, 0.7]   (ok)        0     0.0%   <NA>
##    (0.7, 1]   (bad)       0     0.0%   <NA>
##    (1, Inf)   (very bad)  1     4.8%   2
```

Linear Student's t model.

Using more robust more robust Student's t observation model.

```
writeLines(readLines("lin_t_pr.stan"))
```

```
## Warning in readLines("lin_t_pr.stan"): incomplete final line found on
## 'lin_t_pr.stan'
```

```
## // Linear student-t model
## data {
##   int<lower=0> N; // number of data points
##   vector[N] x; //
##   vector[N] y; //
```

```
##   real xpred; // input location for prediction
## }
## parameters {
##   real alpha;
##   real beta;
##   real<lower=0> sigma;
##   real<lower=1, upper=80> nu;
## }
## transformed parameters {
##   vector[N] mu;
##   mu = alpha + beta*x;
## }
## model {
##   nu ~ gamma(2, 0.1); // Juárez and Steel(2010)
##   y ~ student_t(nu, mu, sigma);
## }
## generated quantities {
##   real ypred;
##   vector[N] log_lik;
##   ypred = normal_rng(alpha + beta*xpred, sigma);
##   for (i in 1:N)
##     log_lik[i] = student_t_lpdf(y[i] | nu, mu[i], sigma);
## }
```

```
fit_lin_t <- stan(file = 'lin_t_pr.stan', data = d_lin, seed = 48927)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'C:
## \Users\Dana\Documents\lin_t_pr.stan'
```

```
samples_lin_t <- rstan::extract(fit_lin_t, permuted = T)
mean(samples_lin_t$beta>0.2)
```

```
## [1] 0
```

```
fit_lin_t_log_lik <- extract_log_lik(fit_lin_t, merge_chains = FALSE)
r_eff <- relative_eff(exp(fit_lin_t_log_lik))

fit_lin_t_loo_lin <- loo(fit_lin_t_log_lik, r_eff = r_eff)
```

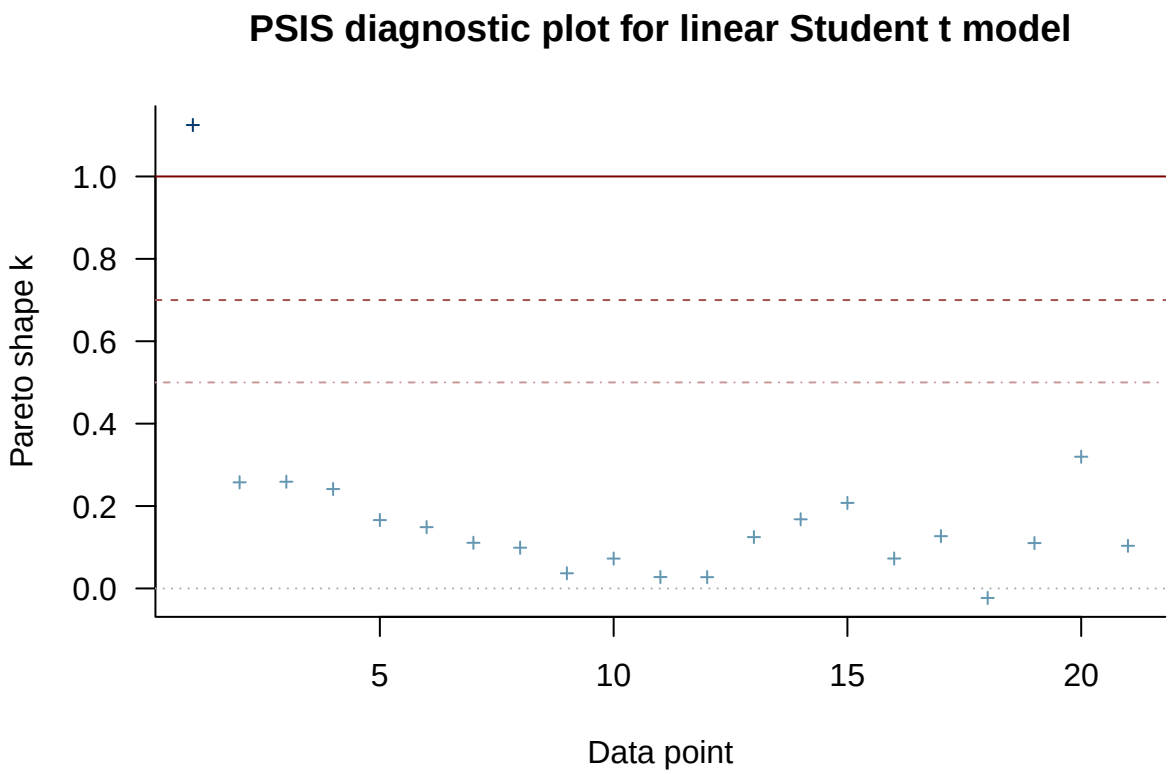
```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```
## Warning in log(z): NaNs produced
```

```
fit_lin_t_loo_lin$elpd_loo
```

```
## Warning: Accessing elpd_loo using '$' is deprecated and will be removed in
## a future release. Please extract the elpd_loo estimate from the 'estimates'
## component instead.
```

```
plot(fit_lin_t_loo_lin, diagnostic = c("k", "n_eff"), label_points = FALSE,
main = "PSIS diagnostic plot for linear Student t model")
```




```
pareto_k_table(fit_lin_t_loo_lin)
```

Hierarchical model for gender

```
test1 <-aggregate(vocab$vocabulary,by=list(vocab$sex, vocab$education),FUN=var,na.rm=TRUE)
```

```
d_hier <-list(N = nrow(test1),
              K = 2,
              x = rep(1:2, nrow(test)),
              y = test1$x)
cat(readLines('lin_pr.stan'), sep='\n')
```

```
## Warning in readLines("lin_pr.stan"): incomplete final line found on
## 'lin_pr.stan'

## // Gaussian linear model with adjustable priors
## data {
##   int<lower=0> N; // number of data points
##   vector[N] x; //
```

```
##   vector[N] y; //
##   real xpred; // input location for prediction
##   real pmualpha; // prior mean for alpha
##   real psalpha;  // prior std for alpha
##   real pmubeta;  // prior mean for beta
##   real psbeta;   // prior std for beta
## }
## parameters {
##   real alpha;
##   real beta;
##   real<lower=0> sigma;
## }
## transformed parameters {
##   vector[N] mu;
##   mu = alpha + beta*x;
## }
## model {
##   alpha ~ normal(pmualpha, psalpha);
##   beta ~ normal(pmubeta, psbeta);
##   y ~ normal(mu, sigma);
## }
## generated quantities {
##   real ypred;
##   vector[N] log_lik;
##   ypred = normal_rng(alpha + beta*xpred, sigma);
##   for (i in 1:N)
##     log_lik[i] = normal_lpdf(y[i] | mu[i], sigma);
## }
```

```
writeLines(readLines("pr_hier_prior_mean.stan"))
```

```
## Warning in readLines("pr_hier_prior_mean.stan"): incomplete final line
## found on 'pr_hier_prior_mean.stan'
```

```
## // Comparison of k groups with common variance and
## // hierarchical prior for the mean
## data {
##     int<lower=0> N; // number of data points
##     int<lower=0> K; // number of groups
##     int<lower=1,upper=K> x[N]; // group indicator
##     vector[N] y; //
## }
## parameters {
##     real mu0;             // prior mean
##     real<lower=0> sigma0; // prior std
##     vector[K] mu;         // group means
##     real<lower=0> sigma;  // common std
## }
## model {
##   //mu ~ normal(mu0, sigma0); // population prior with unknown parameters
##   //y ~ normal(mu[x], sigma);
##   mu0 ~ normal(0,5);      // weakly informative prior
##   sigma0 ~ normal(0,4);     // weakly informative prior
```

```
##   mu ~ normal(mu0, sigma0); // population prior with unknown parameters
##   sigma ~ normal(0,4);      // weakly informative prior
##   y ~ normal(mu[x], sigma);
## }
## generated quantities {
##   vector[N] log_lik;
##   for (i in 1:N)
##   log_lik[i] = normal_lpdf(y[i] | mu[x[i]], sigma);
## }
```

```
fit_hier <- stan(file = 'pr_hier_prior_mean.stan', data = d_hier, seed = 48927)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'C:
## \Users\Dana\Documents\pr_hier_prior_mean.stan'
```

```
## Warning: There were 353 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
#monitor(fit_hier, probs = c(0.1, 0.5, 0.9))
```

#test

```
hier_log_lik <- extract_log_lik(fit_hier, merge_chains = FALSE)
r_eff <- relative_eff(exp(hier_log_lik))
hier_loo_lin <- loo(hier_log_lik, r_eff = r_eff)
```

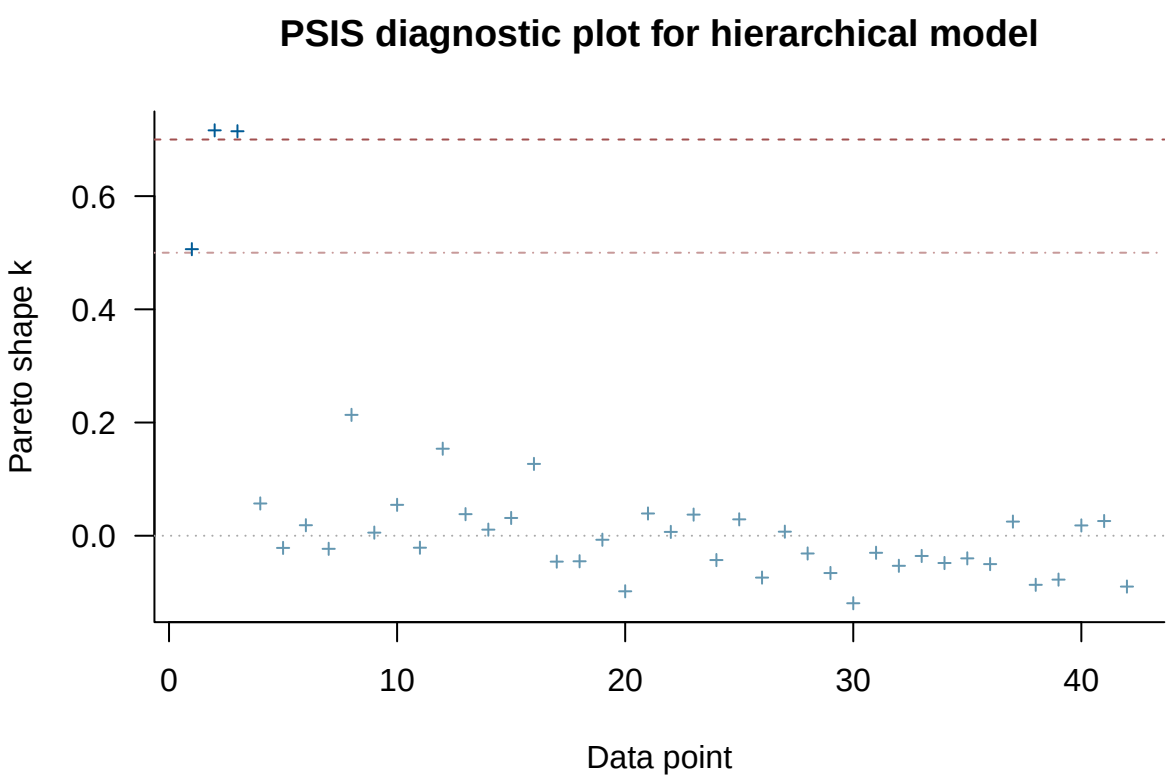
```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```
hier_loo_lin$elpd_loo
```

```
## Warning: Accessing elpd_loo using '$' is deprecated and will be removed in
## a future release. Please extract the elpd_loo estimate from the 'estimates'
## component instead.
```

```
## [1] -74.65278
```

```
plot(hier_loo_lin, diagnostic = c("k", "n_eff"), label_points = FALSE,
main = "PSIS diagnostic plot for hierarchical model")
```




```
pareto_k_table(hier_loo_lin)
```

```
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     39    92.9%   1232
##  (0.5, 0.7]   (ok)        1     2.4%   169
##    (0.7, 1]   (bad)       2     4.8%   59
##    (1, Inf)   (very bad)  0     0.0%   <NA>
```

6. Model comparison

PSIS-LOO differences in models

We can use loo_compare function to compare the different models, which unsuprisingly returns hierarchical model as the best model as it has the lowest elpdloo_cv value. The models differ in elpdloo_cv and standard error according to loo_compare.

```
compare(g_loo_lin , fit_lin_std_loo_lin,fit_lin_t_loo_lin)
```

```
##                     elpd_diff se_diff elpd_loo p_loo looic
## fit_lin_t_loo_lin     0.0       0.0   -20.8      7.1  41.6
## fit_lin_std_loo_lin -10.0       2.8   -30.8      7.6  61.5
## g_loo_lin           -10.1       3.5   -30.8      7.4  61.7
```

7. Prior Sensitivity Analysis

8. Discussion & Conclusion

9. References

[1] http://centaur.reading.ac.uk/29879/2/vocabulary%20size%20revisited_JM_JTD.pdf
[2] https://rstudio-pubs-static.s3.amazonaws.com/458076_f4e193525a424489ba0175e29038d32d.html
[3]https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations