University of Edinburgh

School of Mathematics

Bayesian Data Analysis

Assignment 2 - s2512060 (Aryan Verma)

```
rm(list = ls(all = TRUE))
#Do not delete this!
#It clears all variables to ensure reproducibility
```

Figure 1: **The dataset is about the houses found in a given California district and some summary stats about them based on the 1990 census data.**

```
#
library(INLA)
```

```
## Warning: package 'INLA' was built under R version 4.3.2

## Loading required package: Matrix

## Loading required package: sp

## This is INLA_24.02.09 built 2024-02-09 03:43:24 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
##  - List available models/likelihoods/etc with inla.list.models()
##  - Use inla.doc(<NAME>) to access documentation
```

```r
housing<-read.csv("housing.csv")
#removing rows with NA's, there are only a few of these
housing=housing[complete.cases(housing), ]
#creating a new covariate
housing$average_bed_rooms=housing$total_bedrooms/housing$households

##############################################
### Transforming the data for every model ####
##############################################
housing$log_median_income <- log(housing$median_income) # Taking log of median income
housing$y <- log(housing$median_house_value) # Log of median house value (Response)

# Scaling the non-categorical coordinates
housing$longitude = scale(housing$longitude)
housing$latitude = scale(housing$latitude)
housing$housing_median_age = scale(housing$housing_median_age)
housing$population = scale(housing$population)
housing$log_median_income = scale(housing$log_median_income)
housing$average_bed_rooms = scale(housing$average_bed_rooms)

# Additional characteristics required in question 2
housing$log_median_income2<-scale(housing$log_median_income^2) # log_median_income^2
housing$log_median_income3<-scale(housing$log_median_income^3) # log_median_income^3
housing$log_median_income4<-scale(housing$log_median_income^4) # log_median_income^4
housing$housing_median_age2<-scale(housing$housing_median_age^2) # housing_median_age^2
housing$housing_median_age3<-scale(housing$housing_median_age^3) # housing_median_age^2
housing$housing_median_age4<-scale(housing$housing_median_age^4) # housing_median_age^2
```

The covariates in the dataset are as follows:

longitude, latitude, housing_median_age (median age of houses in district), total_rooms (total rooms in all houses in district), total_bedrooms (total bedrooms in all houses in district), population (population of district), households (number of households in district), median_income (median income in district), median_house_value (median house value in district), ocean_proximity (categorical covariate about proximity of district to ocean), average_bed_rooms (average number of bedrooms of houses in district).

```r
# We split the original dataset into two parts, training and test
housing.training<-housing[seq(from=1,to=nrow(housing),by=2), ]
housing.test<-housing[seq(from=2,to=nrow(housing),by=2), ]
```

Q1)[**10 marks**]

Fit a Bayesian Linear regression model in INLA (with Gaussian likelihood) using the housing.training dataset such that the response variable is the log(median_house_value), and the covariates in the model are as follows:

longitude, latitude, housing_median_age, log(median_income), ocean_proximity, average_bed_rooms.

Use scaled versions of the non-categorical covariates in your model.

Print out the model summary and interpret the posterior means of the regression coefficients.

```r
# Define the model formula
formula1 <- y ~ 1 + longitude + latitude + housing_median_age + log_median_income + as.factor(ocean_pro

# Defining prior for sigma
sigma.unif.prior = "expression:
  b = 5;
  log_dens = (theta>=(-2*log(b)))*(-log(b)-theta/2-log(2)) + (theta<(-2*log(b)))*(-Inf);
  return(log_dens);
"
b1=5;
prec.prior1 <- list(prec=list(prior = sigma.unif.prior,initial = (-2*log(b1)+1), fixed = FALSE))
# Beta prior
prior.beta1=list(mean.intercept = 0, prec.intercept = 0.1,
                    mean = 0, prec = 0.1)

# Trainin the model using INLA
model1 <- inla(formula1, family="gaussian",
              data=housing.training,
              control.family=list(hyper=prec.prior1),
              control.fixed=prior.beta1,
              control.compute=list(cpo=T,dic=T,waic=T, config=TRUE),
              control.predictor = list(compute=TRUE))

# Print out the model summary
summary(model1)
```

```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts,
##    ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##    scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##    ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##    verbose, ", " lincomb = lincomb, selection = selection, control.compute
##    = control.compute, ", " control.predictor = control.predictor,
##    control.family = control.family, ", " control.inla = control.inla,
##    control.fixed = control.fixed, ", " control.mode = control.mode,
##    control.expert = control.expert, ", " control.hazard = control.hazard,
##    control.lincomb = control.lincomb, ", " control.update =
##    control.update, control.lp.scale = control.lp.scale, ", "
##    control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##    ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##    num.threads, ", " keep = keep, working.directory = working.directory,
##    silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug =
##    debug, .parent.frame = .parent.frame)" )
## Time used:
##     Pre = 0.655, Running = 0.736, Post = 0.151, Total = 1.54
## Fixed effects:
##                                     mean    sd 0.025quant 0.5quant
## (Intercept)                       12.185 0.006     12.173   12.185
## longitude                         -0.313 0.014     -0.341   -0.313
## latitude                          -0.324 0.015     -0.354   -0.324
## housing_median_age                 0.028 0.004      0.020    0.028
## log_median_income                  0.322 0.004      0.315    0.322
```

```
## as.factor(ocean_proximity)INLAND     -0.310 0.012     -0.334   -0.310
## as.factor(ocean_proximity)ISLAND      0.706 0.241      0.234    0.706
## as.factor(ocean_proximity)NEAR BAY   -0.005 0.013     -0.031   -0.005
## as.factor(ocean_proximity)NEAR OCEAN -0.014 0.011     -0.036   -0.014
## average_bed_rooms                     0.032 0.003      0.025    0.032
##                                      0.975quant   mode kld
## (Intercept)                              12.196 12.185   0
## longitude                                -0.285 -0.313   0
## latitude                                 -0.295 -0.324   0
## housing_median_age                        0.035  0.028   0
## log_median_income                         0.329  0.322   0
## as.factor(ocean_proximity)INLAND         -0.286 -0.310   0
## as.factor(ocean_proximity)ISLAND          1.177  0.706   0
## as.factor(ocean_proximity)NEAR BAY        0.021 -0.005   0
## as.factor(ocean_proximity)NEAR OCEAN      0.007 -0.014   0
## average_bed_rooms                         0.039  0.032   0
##
## Model hyperparameters:
##                                         mean    sd 0.025quant 0.5quant
## Precision for the Gaussian observations 8.61 0.121       8.37     8.61
##                                         0.975quant mode
## Precision for the Gaussian observations       8.85 8.61
##
## Deviance Information Criterion (DIC) ...............: 7001.28
## Deviance Information Criterion (DIC, saturated) ....: 10217.30
## Effective number of parameters .....................: 6.06
##
## Watanabe-Akaike information criterion (WAIC) ...: 7031.82
## Effective number of parameters .................: 26.92
##
## Marginal log-Likelihood:  -3568.09
## CPO, PIT is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Interpretation of the posterior means of coefficients:

Though this model is simply trained with bayesian linear regression and is not performing well, still let us try to interpret the posterior means of the regression coefficients.

- As we are able to observe that if the home is situated INLAND, it can negatively impact the median house value, with probably a 31% decline on the log scale. Also, a very less negative effect (1.4%) is observed for the houses near the ocean, and further less in magnitude (0.5%) for NEAR THE BAY. Also, the houses on ISLAND seems to have associated with more house value on a log scale by 70.6%, keeping other things constant. (Although, this effect is un-justified, as this much difference in the value can be really surprising, there is a need to model other factors with random effects, that we will see in coming answers)
- As seen from the coefficient of housing_median_age, the median value of the house seems to go up 2.8% on the log scale, when moving ahead from the average house age.
- Log_median_income seems to impact the median value of the house positively and more in magnitude. It looks that it can influence upto 32.2% for a unit-change in the log_median_income from the average.

- Average bedrooms also are positively related with the median house value on a log scale, where a change of 1 unit from the expected average bedrooms the house value on log scale can increase by 3.2%.
- The location (longitudes and longitudes) seem to have a major impact, but that can't be quantified directly as it won't make sense. it will be better judged when we will be plotting the SPDE effect in terms of location.

Compute the DIC, NLSCPO and WAIC scores.

Check the sensitivity of your results to changing the priors.

```
cat("Marginal log-likelihood of model:",model1$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -3567.702
```

```
cat("DIC of model:",model1$dic$dic,"\n")
```

```
## DIC of model: 7001.28
```

```
cat("WAIC of model:",model1$waic$waic,"\n")
```

```
## WAIC of model: 7031.821
```

```
cat("NSLCPO of model:",-sum(log(model1$cpo$cpo)),"\n")
```

```
## NSLCPO of model: 3512.783
```

Checking the sensitivity of the results using various priors for the beta, and precision parameter

```
# Let us change the priors and check for the sensitivty of the results
b=5;
alt_prec_prior <- list(prec=list(prior = sigma.unif.prior, initial = (-2*log(b)+1), fixed = FALSE))
alt_prior_beta <- list(mean.intercept = 0, prec.intercept = 1e-5,
                       mean = 0, prec = 1e-4)

model_alt <- inla(formula1,family="gaussian",
              data=housing.training,
              control.family=list(hyper=alt_prec_prior),
              control.fixed=alt_prior_beta,
              control.compute=list(cpo=T,dic=T,waic=T))

cat("Marginal log-likelihood of model:",model_alt$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -3595.921
```

```
cat("DIC of model:",model_alt$dic$dic,"\n")
```

```
## DIC of model: 7001.292
```

```r
cat("WAIC of model:",model_alt$waic$waic,"\n")
```

```
## WAIC of model: 7031.829
```

```r
cat("NSLCPO of model:",-sum(log(model_alt$cpo$cpo)),"\n")
```

```
## NSLCPO of model: 3512.789
```

As we can see that by increasing the variance of the priors, the change in the marginal log-likelihood is noticed (It decreases with the decrease in he variance of the priors) but not much in the DIC, WAIC, NSLCPO scores. These scores remain almost same, as the priors are changed.

Q2)[10 marks]

Update your model in Q1 to also include an rw1 random effect model for the housing_median_age, and an ar1 random effect model for log(median_income).

Print out the model summary and interpret the posterior means of the regression coefficients.

```r
# Define the model formula
formula2 <- y ~ 1 + longitude + latitude + average_bed_rooms + as.factor(ocean_proximity) +f(housing_med

# Fit the model using INLA
sigma.unif.prior = "expression:
  b = 5;
  log_dens = (theta>=(-2*log(b)))*(-log(b)-theta/2-log(2)) + (theta<(-2*log(b)))*(-Inf);
  return(log_dens);
"
b2=10;
prec.prior2 <- list(prec=list(prior = sigma.unif.prior,initial = (-2*log(b2)+1), fixed = FALSE))
prior.beta2=list(mean.intercept = 0, prec.intercept = 0.1,
                    mean = 0, prec = 1)
model2 <- inla(formula2,family="gaussian",
               data=housing.training,
               control.family=list(hyper=prec.prior2),
               control.fixed=prior.beta2,
               control.predictor = list(compute=TRUE),
               control.compute=list(cpo=T,dic=T,waic=T, config=TRUE))

# Print out the model summary
summary(model2)
```

```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts,
##    ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##    scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##    ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##    verbose, ", " lincomb = lincomb, selection = selection, control.compute
##    = control.compute, ", " control.predictor = control.predictor,
##    control.family = control.family, ", " control.inla = control.inla,
##    control.fixed = control.fixed, ", " control.mode = control.mode,
##    control.expert = control.expert, ", " control.hazard = control.hazard,
```

```
##    control.lincomb = control.lincomb, ", " control.update =
##    control.update, control.lp.scale = control.lp.scale, ", "
##    control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##    ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##    num.threads, ", " keep = keep, working.directory = working.directory,
##    silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug =
##    debug, .parent.frame = .parent.frame)" )
## Time used:
##     Pre = 0.624, Running = 4.63, Post = 0.588, Total = 5.84
## Fixed effects:
##                                        mean    sd 0.025quant 0.5quant
## (Intercept)                          12.198 0.360     11.328   12.232
## longitude                            -0.295 0.014     -0.322   -0.295
## latitude                             -0.307 0.015     -0.336   -0.307
## average_bed_rooms                     0.029 0.003      0.022    0.029
## as.factor(ocean_proximity)INLAND     -0.318 0.012     -0.341   -0.318
## as.factor(ocean_proximity)ISLAND      0.594 0.227      0.148    0.594
## as.factor(ocean_proximity)NEAR BAY   -0.039 0.013     -0.065   -0.039
## as.factor(ocean_proximity)NEAR OCEAN -0.013 0.011     -0.033   -0.013
##                                      0.975quant   mode   kld
## (Intercept)                              12.825 12.274 0.005
## longitude                                -0.268 -0.295 0.000
## latitude                                 -0.279 -0.307 0.000
## average_bed_rooms                         0.035  0.029 0.000
## as.factor(ocean_proximity)INLAND         -0.295 -0.318 0.000
## as.factor(ocean_proximity)ISLAND          1.040  0.594 0.000
## as.factor(ocean_proximity)NEAR BAY       -0.013 -0.039 0.000
## as.factor(ocean_proximity)NEAR OCEAN      0.008 -0.013 0.000
##
## Random effects:
##   Name     Model
##     housing_median_age RW1 model
##    log_median_income AR1 model
##
## Model hyperparameters:
##                                           mean    sd 0.025quant 0.5quant
## Precision for the Gaussian observations   9.27  0.13      9.009     9.27
## Precision for housing_median_age        132.23 57.17     58.370   120.18
## Precision for log_median_income          10.14 12.27      0.653     6.32
## Rho for log_median_income                 1.00  0.00      1.000     1.00
##                                         0.975quant  mode
## Precision for the Gaussian observations       9.52  9.27
## Precision for housing_median_age            278.06 99.40
## Precision for log_median_income              42.39  1.76
## Rho for log_median_income                     1.00  1.00
##
## Deviance Information Criterion (DIC) ...............: 6340.09
## Deviance Information Criterion (DIC, saturated) ....: 10311.82
## Effective number of parameters .....................: 92.77
##
## Watanabe-Akaike information criterion (WAIC) ...: 6358.88
## Effective number of parameters .................: 105.80
##
## Marginal log-Likelihood:  -3340.99
```

```
## CPO, PIT is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

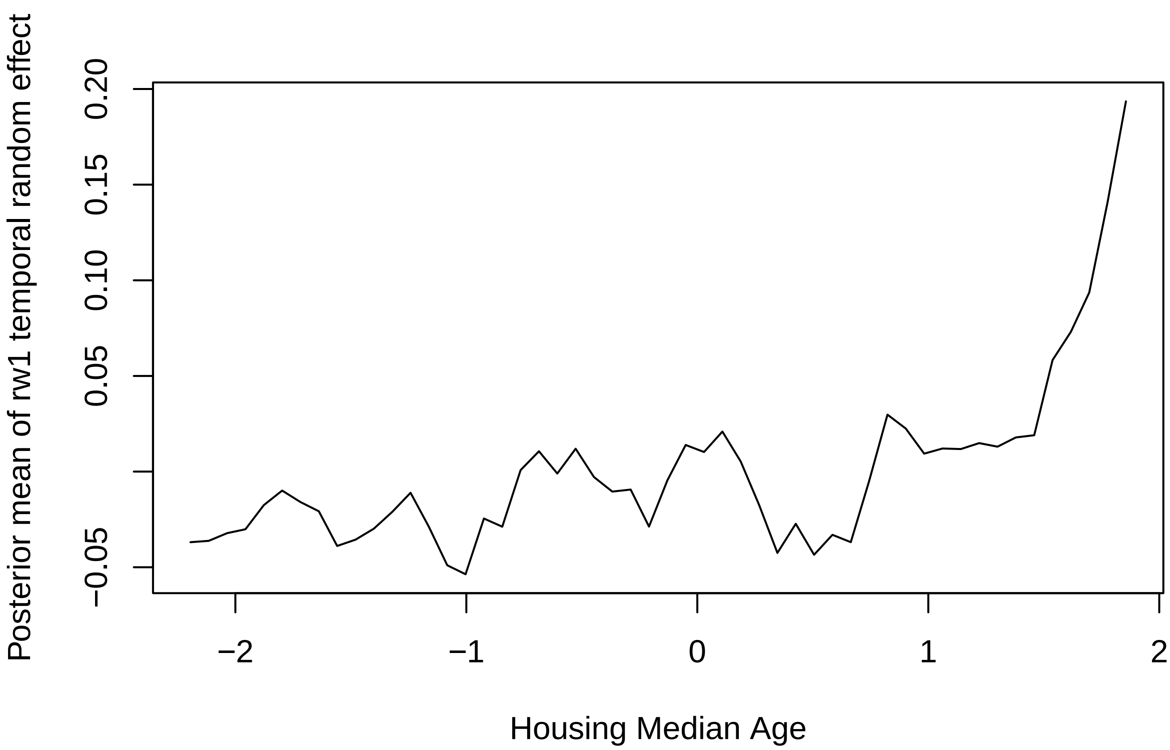
Interpretation of the posterior means of coefficients:

This model is trained with bayesian linear regression and includes the Random Walk 1 model for housing_median_age, and Auto Regressive 1 model for log_median_income covariates. Let us try to interpret the posterior means of the regression coefficients.

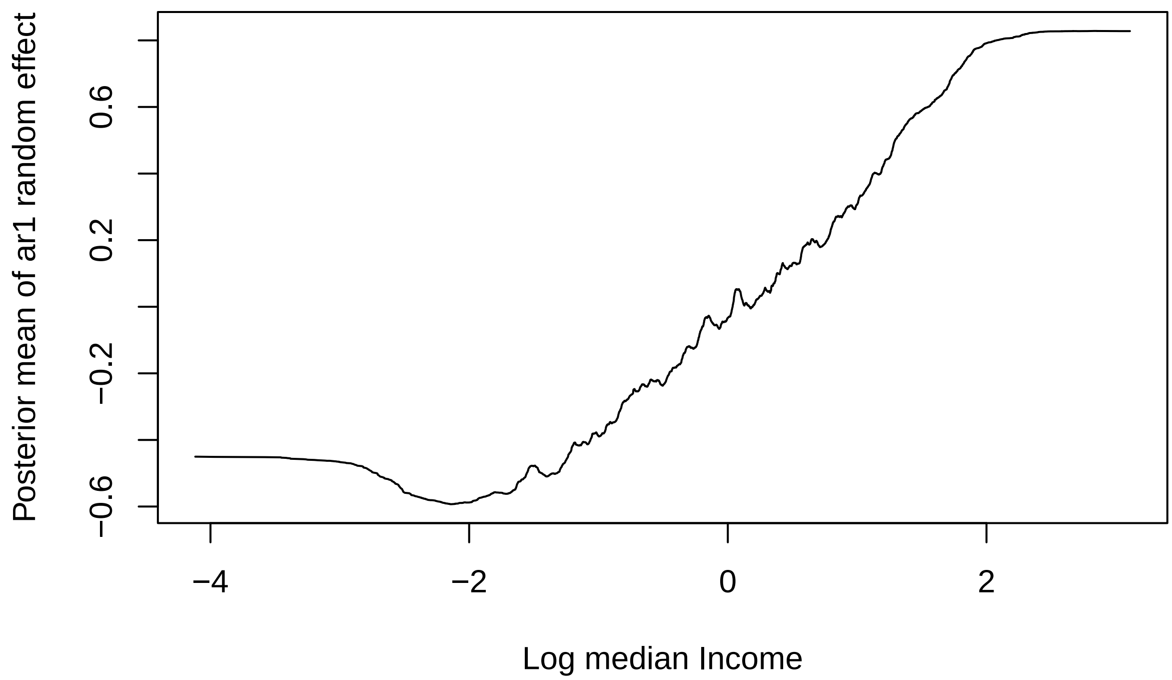
- As we are able to observe that if the home is situated INLAND, it can negatively impact the median house value, with probably a 31% decline on the log scale. Also, a negligible negative effect (1.3%) is observed for the houses near the ocean (Decreased from previous model)
- As we see, compared to previous model, the negative impact magnitude for NEAR THE BAY houses increases from 0.5% in first model to 3.9% here. Also, the houses on ISLAND seems to have associated with more house value on a log scale by 59.3% (As compared to the previous model, with 70.6% change in log scale value of houses.
- Average bedrooms also are positively related with the median house value on a log scale, where a change of 1 unit from the expected average bedrooms the house value on log scale can increase by 2.9%.

Plot the posterior means of the random effects for housing_median_age and log(median_income). The x-axis should be the covariate value (such as housing_median_age), and the y-axis should be the posterior mean of the random effect.

```r
plot(sort(unique(housing.training$housing_median_age)),model2$summary.random$housing_median_age$mean,typ
```



```r
plot(sort(unique(housing.training$log_median_income)),model2$summary.random$log_median_income$mean  , t
```



- As we can notice here that till a specific level the housing median age doesn't play a major role in decision making for the house value, but after some time (More than average), the house value tends to increase with the Median Housing Age.
- A strong increase in median house value is reflected by the graph as the log median income increases. It is even noticed from the very start, and increases with increasing median income.

Compute the DIC, NLSCPO and WAIC scores.

Check the sensitivity of your results to changing the priors.

```r
cat("Marginal log-likelihood of model:",model2$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -3343.739
```

```r
cat("DIC of model:",model2$dic$dic,"\n")
```

```
## DIC of model: 6340.087
```

```r
cat("WAIC of model:",model2$waic$waic,"\n")
```

```
## WAIC of model: 6358.877
```

```r
cat("NSLCPO of model:",-sum(log(model2$cpo$cpo)),"\n")
```

```
## NSLCPO of model: 3178.869
```

Checking the sensitivity of the results using various priors for the beta, and precision parameter

```r
# Let us change the priors and check for the sensitivty of the results
alt_beta_prior <- list(mean.intercept = 0, prec.intercept = 0.00001,
                       mean = 0, prec = 0.001)

model_alt <- inla(formula2,family="gaussian",
              data=housing.training,
              control.family=list(hyper=prec.prior2),
              control.fixed=alt_beta_prior,
              control.compute=list(cpo=T,dic=T,waic=T))

cat("Marginal log-likelihood of model:",model_alt$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -3364.605
```

```r
cat("DIC of model:",model_alt$dic$dic,"\n")
```

```
## DIC of model: 6340.055
```

```r
cat("WAIC of model:",model_alt$waic$waic,"\n")
```

```
## WAIC of model: 6358.805
```

```r
cat("NSLCPO of model:",-sum(log(model_alt$cpo$cpo)),"\n")
```

```
## NSLCPO of model: 3178.836
```

As we can see that by increasing the variance of the intercept and beta parameter, the change in the marginal log-likelihood is noticed (It increases up to a level with the increase in the variance of the priors) but not much in the DIC, WAIC, NSLCPO scores. These scores remain almost same, as the priors are changed.

Q3)[10 marks]

In this question, we will use a spatial random effects model for the location.

Create a Bayesian regression model in INLA or inlabru with Gaussian likelihood using the housing.training dataset with log(median_house_value) as the response variable, and the fixed effects in the model are as follows:

longitude, latitude,

housing_median_age, $(housing\_median\_age)^2$,$(housing\_median\_age)^3$,$(housing\_median\_age)^4$

log(median_income), $(\log(median\_income))^2$, $(\log(median\_income))^3$, $(\log(median\_income))^4$ **,**

housing_median_age*log(median_income),

ocean_proximity, average_bed_rooms.

Use scaled versions of the non-categorical covariates in your model.

Include a spatial (spde2) random effect for the location (longitude, latitude), with Matern covariance. [Hint: You must create a mesh first; see the code for Lecture 7 and the solutions of Workshop 5.]

Print out the model summary and interpret the posterior means of the regression coefficients.

Dear Instructor, here I am one-hot encoding the ocean_proximity because it gives me some errors while using INLA or INLABRU package. Also, I will put the interaction term in the dataset itself for ease in the formula. Also, for self-learning and validation I have used both the approaches that you have taught us, using the INLA and Inlabru.

```
#############################################
########### Preparing the data  ############
#############################################



# one hot encoding the ocean proximity in train and test sets
onehot_ocean_proximity.training <- model.matrix(~0+housing.training$ocean_proximity)
onehot_ocean_proximity.test <- model.matrix(~0+housing.test$ocean_proximity)

onehot_housing.training <- cbind(housing.training,onehot_ocean_proximity.training)
onehot_housing.test <- cbind(housing.test,onehot_ocean_proximity.test)
colnames(onehot_housing.training)[which(names(onehot_housing.training) == "housing.training$ocean_proxi
colnames(onehot_housing.training)[which(names(onehot_housing.training) == "housing.training$ocean_proxi
colnames(onehot_housing.training)[which(names(onehot_housing.training) == "housing.training$ocean_proxi
colnames(onehot_housing.training)[which(names(onehot_housing.training) == "housing.training$ocean_proxi
colnames(onehot_housing.training)[which(names(onehot_housing.training) == "housing.training$ocean_proxi
colnames(onehot_housing.test)[which(names(onehot_housing.test) == "housing.test$ocean_proximity<1H OCEA
colnames(onehot_housing.test)[which(names(onehot_housing.test) == "housing.test$ocean_proximityINLAND")]
colnames(onehot_housing.test)[which(names(onehot_housing.test) == "housing.test$ocean_proximityISLAND")]
colnames(onehot_housing.test)[which(names(onehot_housing.test) == "housing.test$ocean_proximityNEAR BAY
colnames(onehot_housing.test)[which(names(onehot_housing.test) == "housing.test$ocean_proximityNEAR OCE

# Introducing the interaction variable
onehot_housing.training['housing_median_age*log_median_income'] = onehot_housing.training$housing_media
colnames(onehot_housing.training)[which(names(onehot_housing.training) == "housing_median_age*log_media
onehot_housing.test['housing_median_age*log_median_income'] = onehot_housing.test$housing_median_age*one
colnames(onehot_housing.test)[which(names(onehot_housing.test) == "housing_median_age*log_median_income
```

Modelling using the INLA

```
Locations = cbind(onehot_housing.training$longitude, onehot_housing.training$latitude)
prdomain <- inla.nonconvex.hull(as.matrix(onehot_housing.training[, 1:2]),
  convex = -0.03, concave = -0.05,
  resolution = c(100, 100))

prmesh <- inla.mesh.2d(boundary = prdomain, ## Creating the mesh
  max.edge = c(0.5, 1), cutoff = 0.06)
#plot(prmesh)

loc.spde = inla.spde2.pcmatern(mesh = prmesh, ## SPDE
           prior.range = c(1, 0.1),
           prior.sigma = c(10, 0.001))
```

```r
loc.A <- inla.spde.make.A(prmesh, loc = Locations)
loc.w <- inla.spde.make.index('w', n.spde = loc.spde$n.spde)

# Making the data
X0 <- model.matrix(as.formula(" ~ 0+ longitude + latitude + housing_median_age + housing_median_age2 + h

X <- as.data.frame(X0) # convert to a data frame.

N <- nrow(onehot_housing.training) # Number of rows data


# Making the stack
StackPR<- inla.stack(
  data = list(y = onehot_housing.training$y), # y is the response variable

  A = list(1, 1, loc.A), # Vector of Multiplication factors for  fixed effects

  effects = list(
    Intercept = rep(1, N), # Manual intercept
    X = X, # attaching the model matrix
    w = loc.w) ) # attaching the w

# Finally fitting the model
model3 <- inla(y ~ 0 + Intercept + longitude + latitude + housing_median_age + housing_median_age2 + hou
            family = "Gaussian",
            data = inla.stack.data(StackPR),
            control.compute = list(cpo=T,dic = T, waic=T, config=TRUE),
            control.predictor = list(A = inla.stack.A(StackPR), compute=TRUE))

summary(model3)
```

```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts,
##    ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##    scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##    ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##    verbose, ", " lincomb = lincomb, selection = selection, control.compute
##    = control.compute, ", " control.predictor = control.predictor,
##    control.family = control.family, ", " control.inla = control.inla,
##    control.fixed = control.fixed, ", " control.mode = control.mode,
##    control.expert = control.expert, ", " control.hazard = control.hazard,
##    control.lincomb = control.lincomb, ", " control.update =
##    control.update, control.lp.scale = control.lp.scale, ", "
##    control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##    ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##    num.threads, ", " keep = keep, working.directory = working.directory,
##    silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug =
##    debug, .parent.frame = .parent.frame)" )
## Time used:
##     Pre = 0.817, Running = 1.93, Post = 0.0855, Total = 2.84
## Fixed effects:
##                         mean     sd 0.025quant 0.5quant 0.975quant   mode kld
```

```
## Intercept          10.026 12.910    -15.289   10.026     35.342 10.026   0
## longitude          -0.397  0.089     -0.570   -0.399     -0.216 -0.398   0
## latitude           -0.384  0.091     -0.561   -0.385     -0.197 -0.385   0
## housing_median_age  -0.060  0.007     -0.073   -0.060     -0.047 -0.060   0
## housing_median_age2 -0.034  0.010     -0.053   -0.034     -0.015 -0.034   0
## housing_median_age3  0.057  0.006      0.045    0.057      0.070  0.057   0
## housing_median_age4  0.065  0.010      0.045    0.065      0.084  0.065   0
## log_median_income    0.333  0.005      0.324    0.333      0.343  0.333   0
## log_median_income2   0.072  0.006      0.061    0.072      0.083  0.072   0
## log_median_income3  -0.064  0.005     -0.073   -0.064     -0.054 -0.064   0
## log_median_income4  -0.025  0.006     -0.037   -0.025     -0.013 -0.025   0
## Interaction_term     0.025  0.003      0.019    0.025      0.031  0.025   0
## Ocean_lt_1hour       1.843 12.910    -23.472    1.843     27.158  1.843   0
## Inland               1.921 12.910    -23.394    1.921     27.236  1.921   0
## Island               2.519 12.913    -22.802    2.519     27.840  2.519   0
## Near_Bay             1.857 12.910    -23.459    1.857     27.172  1.857   0
## Near_Ocean           1.887 12.910    -23.428    1.887     27.202  1.887   0
## average_bed_rooms    0.019  0.003      0.012    0.019      0.025  0.019   0
##
## Random effects:
##   Name     Model
##     w SPDE2 model
##
## Model hyperparameters:
##                                           mean    sd 0.025quant 0.5quant
## Precision for the Gaussian observations 12.464 0.176     12.128   12.461
## Range for w                              0.852 0.300      0.517    0.777
## Stdev for w                              0.395 0.035      0.336    0.393
##                                         0.975quant   mode
## Precision for the Gaussian observations     12.821 12.448
## Range for w                                  1.646  0.613
## Stdev for w                                  0.473  0.383
##
## Deviance Information Criterion (DIC) ...............: 3340.17
## Deviance Information Criterion (DIC, saturated) ....: 10330.71
## Effective number of parameters .....................: 116.21
##
## Watanabe-Akaike information criterion (WAIC) ...: 3336.70
## Effective number of parameters .................: 116.52
##
## Marginal log-Likelihood:  -1915.23
## CPO, PIT is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Modelling with inlabru package (Gives same results)

```r
library(inlabru)
```

```
## Loading required package: fmesher
```

```r
# Initiating the mesh
Locations = data.frame(easting=onehot_housing.training$longitude, northing=onehot_housing.training$lati
loc.mesh <- inla.mesh.2d(Locations, max.edge = c(0.5, 1), cutoff = 0.06)
loc.spde = inla.spde2.pcmatern(mesh = loc.mesh,
           prior.range = c(1, 0.1),
           prior.sigma = c(10, 0.001))

# Transformed locations
onehot_housing.training$sfLocations <- sf::st_as_sf(Locations,coords = c("easting", "northing"))$geomet

# Formula
cmp <- y ~ floc(sfLocations, model = loc.spde) + longitude + latitude + housing_median_age + housing_me

# Modelling
model3_bru <- bru(cmp, onehot_housing.training,
             family = "gaussian",
             samplers = prdomain,
             domain = list(coordinates = prmesh),
             options=list(control.compute=list(cpo=T,dic=T,waic=T),
                    control.inla=list(tolerance=1e-10)))

summary(model3_bru)
```

```
## inlabru version: 2.10.1
## INLA version: 24.02.09
## Components:
## floc: main = spde(sfLocations), group = exchangeable(1L), replicate = iid(1L)
## longitude: main = linear(longitude), group = exchangeable(1L), replicate = iid(1L)
## latitude: main = linear(latitude), group = exchangeable(1L), replicate = iid(1L)
## housing_median_age: main = linear(housing_median_age), group = exchangeable(1L), replicate = iid(1L)
## housing_median_age2: main = linear(housing_median_age2), group = exchangeable(1L), replicate = iid(1L
## housing_median_age3: main = linear(housing_median_age3), group = exchangeable(1L), replicate = iid(1L
## housing_median_age4: main = linear(housing_median_age4), group = exchangeable(1L), replicate = iid(1L
## log_median_income: main = linear(log_median_income), group = exchangeable(1L), replicate = iid(1L)
## log_median_income2: main = linear(log_median_income2), group = exchangeable(1L), replicate = iid(1L)
## log_median_income3: main = linear(log_median_income3), group = exchangeable(1L), replicate = iid(1L)
## log_median_income4: main = linear(log_median_income4), group = exchangeable(1L), replicate = iid(1L)
## Interaction_term: main = linear(Interaction_term), group = exchangeable(1L), replicate = iid(1L)
## Ocean_lt_1hour: main = linear(Ocean_lt_1hour), group = exchangeable(1L), replicate = iid(1L)
## Inland: main = linear(Inland), group = exchangeable(1L), replicate = iid(1L)
## Island: main = linear(Island), group = exchangeable(1L), replicate = iid(1L)
## Near_Bay: main = linear(Near_Bay), group = exchangeable(1L), replicate = iid(1L)
## Near_Ocean: main = linear(Near_Ocean), group = exchangeable(1L), replicate = iid(1L)
## average_bed_rooms: main = linear(average_bed_rooms), group = exchangeable(1L), replicate = iid(1L)
## Intercept: main = linear(1), group = exchangeable(1L), replicate = iid(1L)
## Likelihoods:
##   Family: 'gaussian'
##     Data class: 'data.frame'
##     Predictor: y ~ .
## Time used:
##     Pre = 0.829, Running = 3.02, Post = 0.247, Total = 4.1
## Fixed effects:
##                          mean     sd 0.025quant 0.5quant 0.975quant   mode kld
```

```
## longitude           -0.424  0.065     -0.554   -0.424     -0.298 -0.424   0
## latitude            -0.447  0.066     -0.580   -0.446     -0.319 -0.446   0
## housing_median_age  -0.042  0.006     -0.054   -0.042     -0.030 -0.042   0
## housing_median_age2 -0.033  0.009     -0.050   -0.033     -0.016 -0.033   0
## housing_median_age3  0.027  0.006      0.016    0.027      0.038  0.027   0
## housing_median_age4  0.049  0.009      0.032    0.049      0.066  0.049   0
## log_median_income    0.303  0.005      0.294    0.303      0.312  0.303   0
## log_median_income2   0.063  0.005      0.053    0.063      0.074  0.063   0
## log_median_income3  -0.073  0.004     -0.082   -0.073     -0.065 -0.073   0
## log_median_income4  -0.028  0.005     -0.039   -0.028     -0.018 -0.028   0
## Interaction_term     0.013  0.003      0.008    0.013      0.018  0.013   0
## Ocean_lt_1hour       1.908 12.910    -23.407    1.908     27.224  1.908   0
## Inland               1.917 12.910    -23.398    1.917     27.233  1.917   0
## Island               2.351 12.914    -22.971    2.351     27.674  2.351   0
## Near_Bay             1.943 12.910    -23.372    1.943     27.259  1.943   0
## Near_Ocean           1.908 12.910    -23.407    1.908     27.223  1.908   0
## average_bed_rooms    0.014  0.003      0.008    0.014      0.020  0.014   0
## Intercept           10.029 12.910    -15.287   10.029     35.344 10.029   0
##
## Random effects:
##   Name     Model
##     floc SPDE2 model
##
## Model hyperparameters:
##                                             mean    sd 0.025quant 0.5quant
## Precision for the Gaussian observations 17.250 0.249     16.767   17.248
## Range for floc                            0.371 0.036      0.307    0.369
## Stdev for floc                            0.387 0.025      0.342    0.386
##                                         0.975quant   mode
## Precision for the Gaussian observations     17.748 17.242
## Range for floc                               0.447  0.364
## Stdev for floc                               0.438  0.383
##
## Deviance Information Criterion (DIC) ...............: 256.41
## Deviance Information Criterion (DIC, saturated) ....: 10571.85
## Effective number of parameters .....................: 401.38
##
## Watanabe-Akaike information criterion (WAIC) ...: 348.31
## Effective number of parameters .................: 393.42
##
## Marginal log-Likelihood:  -637.91
## CPO, PIT is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Interpretation of the posterior means of coefficients:

This model includes the spde random effect on the locations. Let us analyse the coefficients

- Unlike other two models, when the location spde random effect is taken into consideration, the ocean proximity becomes less relevant to be watched upon, instead the spde posterior means are more valuable here to look for. Also, this can be seen by the relevancy of the longitude and latitude, and irrelevancy of the Ocean proximity.
- There is negligible interaction between the housing_median_age and log_median_income.

- Log of the median income is still a major decision maker in determining the value of the house. This is observed to be positively correlated, while its higher orders not being so significant.
- Here, housing median age is seen to be negatively impacting the value of the house. It can be seen that a -3.1% change is seen on the log scale of the house value, when 1-unit median age is increased.

Plot the posterior mean of the spatial random effect in terms of the location.

```
library(devtools)
```

```
## Loading required package: usethis
```

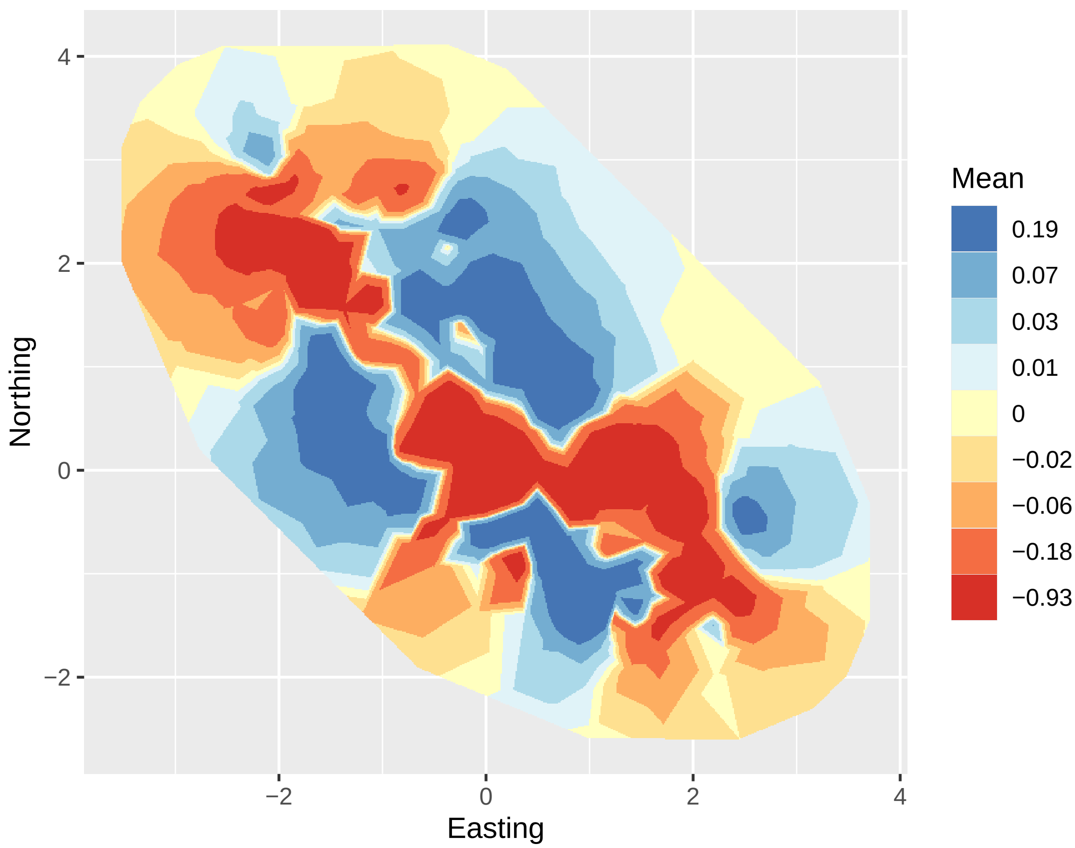
```
if(!require(ggregplot)){
    devtools::install_github("gfalbery/ggregplot")
    library(ggregplot)
}
```

```
## Loading required package: ggregplot

##
## Attaching package: 'ggregplot'

## The following objects are masked _by_ '.GlobalEnv':
##
##     Locations, X
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(RColorBrewer)

ggField(model3, prmesh, Groups = 1,Res=600) + scale_fill_brewer(palette = "RdYlBu")
```



As visible from the figure above, there are certain areas where the value of the houses tend to be more as compared to the other areas.

Compute the DIC, NLSCPO and WAIC scores.

Compare the models in Q1) - Q3) in terms of DIC, NLSCPO and WAIC scores.

```
cat("Marginal log-likelihood of model:",model1$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -3567.702
```

```
cat("DIC of model:",model1$dic$dic,"\n")
```

```
## DIC of model: 7001.28
```

```
cat("WAIC of model:",model1$waic$waic,"\n")
```

```
## WAIC of model: 7031.821
```

```
cat("NSLCPO of model:",-sum(log(model1$cpo$cpo)),"\n")
```

```
## NSLCPO of model: 3512.783
```

```r
cat("\n","\n")
```

```r
cat("Marginal log-likelihood of model:",model2$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -3343.739
```

```r
cat("DIC of model:",model2$dic$dic,"\n")
```

```
## DIC of model: 6340.087
```

```r
cat("WAIC of model:",model2$waic$waic,"\n")
```

```
## WAIC of model: 6358.877
```

```r
cat("NSLCPO of model:",-sum(log(model2$cpo$cpo)),"\n")
```

```
## NSLCPO of model: 3178.869
```

```r
cat("\n","\n")
```

```r
cat("Marginal log-likelihood of model:",model3$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -1916.545
```

```r
cat("DIC of model:",model3$dic$dic,"\n")
```

```
## DIC of model: 3340.174
```

```r
cat("WAIC of model:",model3$waic$waic,"\n")
```

```
## WAIC of model: 3336.697
```

```r
cat("NSLCPO of model:",-sum(log(model3$cpo$cpo)),"\n")
```

```
## NSLCPO of model: 1680.625
```

Clearly, Model 3 is the best model, as it is having the maximum marginal log-likelihood, with minimum DIC, WAIC, and NSLCPO in all the three models. Model 3 performs far better on all the three criteria, hence that is the best model

Check the sensitivity of your results to changing the priors and using a finer mesh.

Let us change the priors of the spde, and also use the finer mesh.

- We will increase the variance of the SPDE priors, and intialise them with a greater value
- Also, we will reduce the cutt-off for the mesh, with small decrease in the max.edge, making the mesh more finer.

```r
# Initiating the mesh
loc.mesh <- inla.mesh.2d(Locations, max.edge = c(0.4, 0.8), cutoff = 0.03)
loc.spde = inla.spde2.pcmatern(mesh = loc.mesh,
           prior.range = c(1, 0.001),
           prior.sigma = c(100, 0.0001))

# Formula
cmp <- y ~ floc(sfLocations, model = loc.spde) + longitude + latitude + housing_median_age + housing_med

# Modelling
model3_bru <- bru(cmp, onehot_housing.training,
             family = "gaussian",
             samplers = prdomain,
             domain = list(coordinates = prmesh),
             options=list(control.compute=list(cpo=T,dic=T,waic=T),
                          control.inla=list(tolerance=1e-10)))

cat("Marginal log-likelihood of model:",model3_bru$mlik[1],"\n")
```

```
## Marginal log-likelihood of model: -120.3566
```

```r
cat("DIC of model:",model3_bru$dic$dic,"\n")
```

```
## DIC of model: -1167.897
```

```r
cat("WAIC of model:",model3_bru$waic$waic,"\n")
```

```
## WAIC of model: -1180.359
```

```r
cat("NSLCPO of model:",-sum(log(model3_bru$cpo$cpo)),"\n")
```

```
## NSLCPO of model: -505.0812
```

As we see here that using a finer mesh, and giving the parameters of the spde more variance helps us to model the complexities in the locations using the random effects. Now, this has to be controlled using various predictive checks on the test dats also. As, there has to be a trade-off between the training accuracy and testing accuracy of the model.

Hence, for these performance increment, th performance of the test set should also be continuously checked for .

Q4)[10 marks]

In this question, we will evaluate the predictive performance of these models.

Do the following two tests for all 3 models.

First, compute the posterior mean of the log(median_house_value) for the districts in the training dataset housing.training. Compute the median absolute difference between the posterior means of the log(median_house_value) and its true values on the training dataset. This can be done by including the posterior means in an array v , the true values in an array t, and computing $\text{median}(|v - t|)$.

Second, evaluate the log(median__house__value) 's posterior predictive means on the test dataset housing.test. Compute the median absolute difference between the log(median__house__value) 's posterior predictive mean and its true value on the test dataset.

Discuss the results.

```r
## Now for evaluating the posterior means of log(median_house_value) on test data, we will bind the data

# Evaluate the posterior predictive means of log(median_house_value) on the test dataset
y_test <- housing.test$y # Note th response variable separately
housing.test$y <- NA # set NA as response in data

data_binded <- rbind(housing.training, housing.test) # merge train and test (with NA)

# Model 1
model1_test <- inla(formula1,family="gaussian",
               data=data_binded,
               control.family=list(hyper=prec.prior1),
               control.fixed=prior.beta1,
               control.compute=list(cpo=T,dic=T,waic=T),
               control.predictor = list(compute=TRUE))
# Model 2
model2_test <- inla(formula2,family="gaussian",
               data=data_binded,
               control.family=list(hyper=prec.prior2),
               control.fixed=prior.beta2,
               control.predictor = list(compute=TRUE),
               control.compute=list(cpo=T,dic=T,waic=T))

# Model 3
onehot_housing.test$y <- NA # Using the one-hot encoded data with NA response
onehot_housing.training <- subset(onehot_housing.training, select = -c(sfLocations)) # Removing due to 
data_binded <- rbind(onehot_housing.training, onehot_housing.test) # Merging train and test
Locations = cbind(data_binded$longitude, data_binded$latitude)
prdomain <- inla.nonconvex.hull(as.matrix(data_binded[, 1:2]),
  convex = -0.03, concave = -0.05,
  resolution = c(100, 100))
prmesh <- inla.mesh.2d(boundary = prdomain,
  max.edge = c(0.45, 1), cutoff = 0.1)
loc.spde = inla.spde2.pcmatern(mesh = prmesh,
           prior.range = c(1, 0.1),
           prior.sigma = c(100, 0.1))
loc.A <- inla.spde.make.A(prmesh, loc = Locations)
loc.w <- inla.spde.make.index('w', n.spde = loc.spde$n.spde)
X0 <- model.matrix(as.formula(" ~ 0+ longitude + latitude + housing_median_age + housing_median_age2 + 

X <- as.data.frame(X0) # convert to a data frame.
N <- nrow(data_binded)

StackPR<- inla.stack(
  data = list(y = data_binded$y), # specify the response variable
  A = list(1, 1, loc.A), # Vector of Multiplication factors for  fixed effects
  effects = list(
    Intercept = rep(1, N), # specify the manual intercept!
```

```r
    X = X, # attach the model matrix
   # insert vectors of any random effects
    w = loc.w) ) # attach the w

model3_test <- inla(y ~ 0 + Intercept + longitude + latitude + housing_median_age + housing_median_age2
            family = "Gaussian",
            data = inla.stack.data(StackPR),
            control.compute = list(cpo=T,dic = T, waic=T),
            control.predictor = list(A = inla.stack.A(StackPR), compute=TRUE))


# Compute the posterior mean of log(median_house_value) for the districts in the training dataset
posterior_means_model1 <- model1$summary.fitted.values$mean
posterior_means_model2 <- model2$summary.fitted.values$mean
posterior_means_model3 <- model3$summary.fitted.values$mean

# Compute the posterior mean of log(median_house_value) for the districts in the test dataset
posterior_means_model1_test <- model1_test$summary.fitted.values$mean
posterior_means_model2_test <- model2_test$summary.fitted.values$mean
posterior_means_model3_test <- model3_test$summary.fitted.values$mean

# Compute the median absolute difference between the posterior means and the true values on the trainin
median_absolute_diff_model1 <- median(abs(posterior_means_model1[1:10217] - onehot_housing.training$y))
median_absolute_diff_model2 <- median(abs(posterior_means_model2[1:10217] - onehot_housing.training$y))
median_absolute_diff_model3 <- median(abs(posterior_means_model3[1:10217] - onehot_housing.training$y))

# Compute the median absolute difference between the posterior means and the true values on the test da
median_absolute_diff_model1_test <- median(abs(posterior_means_model1_test[10218:20433] - y_test))
median_absolute_diff_model2_test <- median(abs(posterior_means_model2_test[10218:20433] - y_test))
median_absolute_diff_model3_test <- median(abs(posterior_means_model3_test[10218:20433] - y_test))


# Print the results
cat("Median Absolute Difference (Training Dataset) - Model 1:", median_absolute_diff_model1, "\n")
```

```
## Median Absolute Difference (Training Dataset) - Model 1: 0.2094092
```

```r
cat("Median Absolute Difference (Training Dataset) - Model 2:", median_absolute_diff_model2, "\n")
```

```
## Median Absolute Difference (Training Dataset) - Model 2: 0.1993266
```

```r
cat("Median Absolute Difference (Training Dataset) - Model 3:", median_absolute_diff_model3, "\n")
```

```
## Median Absolute Difference (Training Dataset) - Model 3: 0.1657802
```

```r
cat("Median Absolute Difference (Test Dataset) - Model 1:", median_absolute_diff_model1_test, "\n")
```

```
## Median Absolute Difference (Test Dataset) - Model 1: 0.2083049
```

```r
cat("Median Absolute Difference (Test Dataset) - Model 2:", median_absolute_diff_model2_test, "\n")
```

```
## Median Absolute Difference (Test Dataset) - Model 2: 0.2031361
```

```r
cat("Median Absolute Difference (Test Dataset) - Model 3:", median_absolute_diff_model3_test, "\n")
```

```
## Median Absolute Difference (Test Dataset) - Model 3: 0.162631
```

Le us discuss the predictive performance of the models trained here:

- Model 1: This model performs the worst out of all three models. As can be seen the median absolute error is highest on both training and testing data.
- Model 2: This model performs better than Model 1 on training data, but is again unable to capture the nature of the data, as can be seen from almost similar performance to model 1 on test data. Hence, this is some what better that model 1, but not more than model 3.
- Model 3: This model is able to capture the complexities of the data through the SPDE modelling for location. This performs equally well on the train and test sets, hence, is a very good model for this data. There is no evidence of overfiting as the performance is really well on both the sets of data.

Q5)[10 marks] Perform posterior predictive checks (using replicates) on all 3 models Q1-Q3 fitted on the housing.training dataset. Choose your test functions to provide insight into the model. Discuss the results.

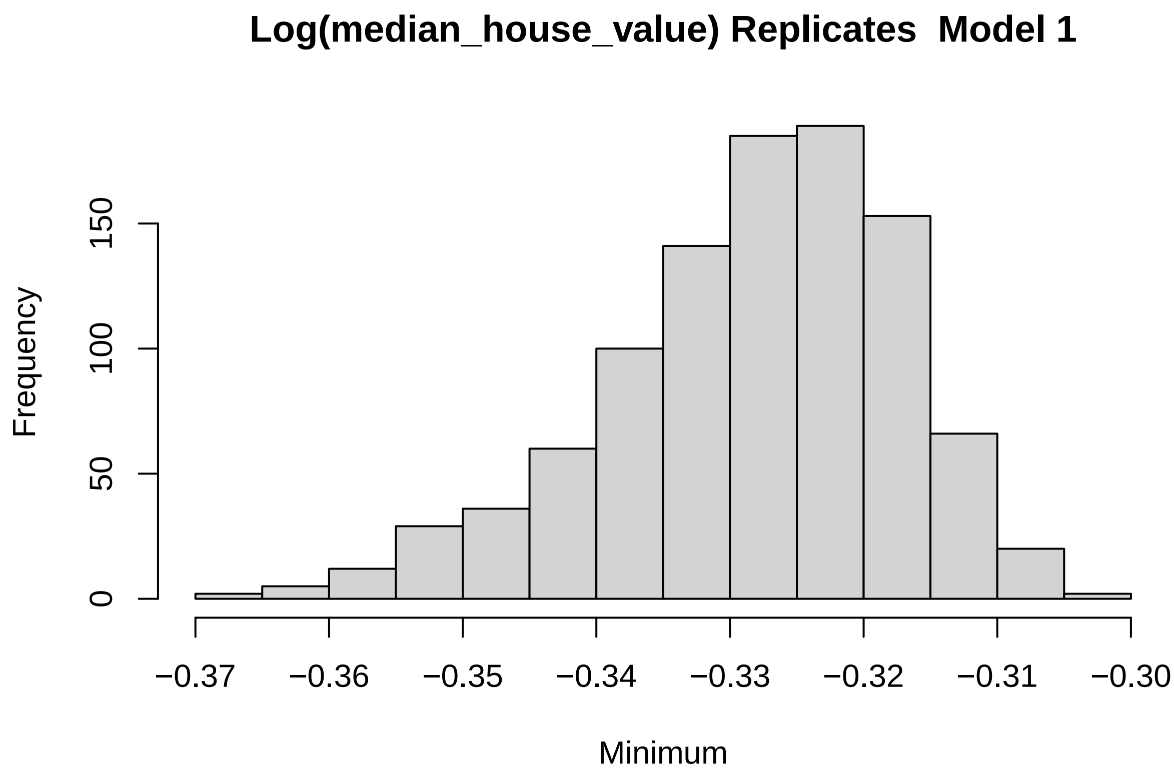
Dear Instructor, I first used the mean and standard deviation as test functions, but soon came across various skewed distributions, hence, resorted to be choosing Min, Max, Median, Skewness, and Kurtosis functions. Now, let us evaluate these functions for all the three models on the replicated data.

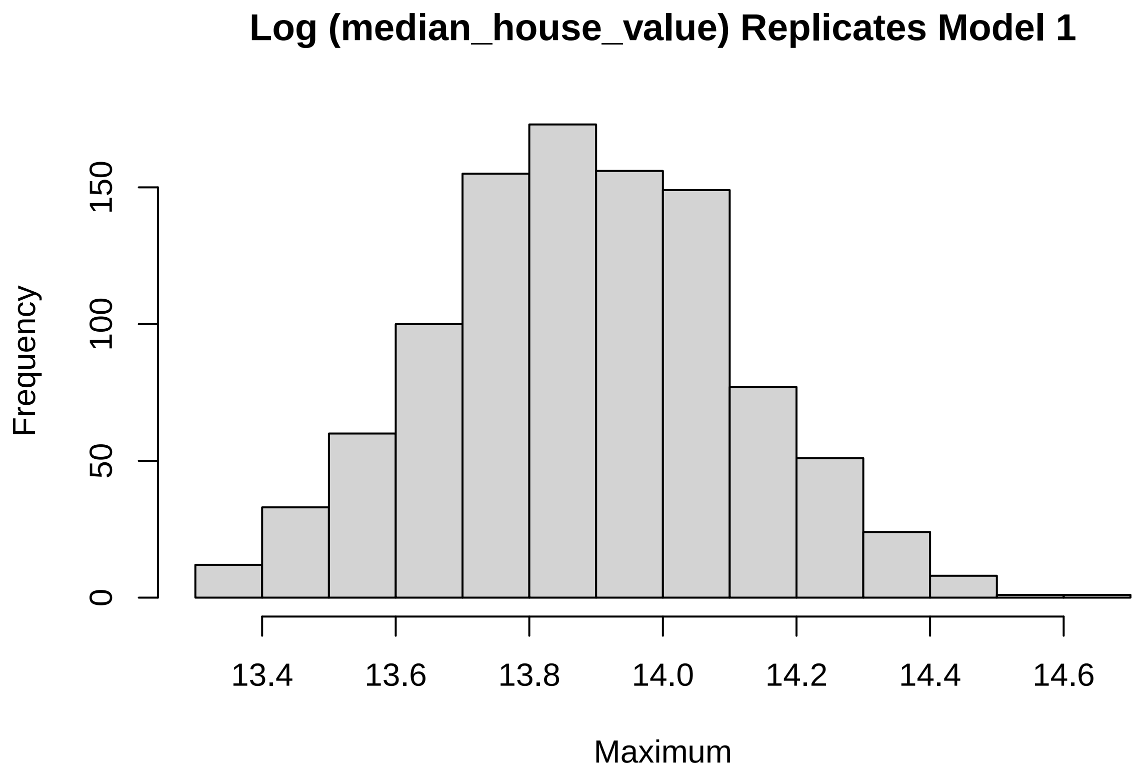
```r
require(fBasics)
```

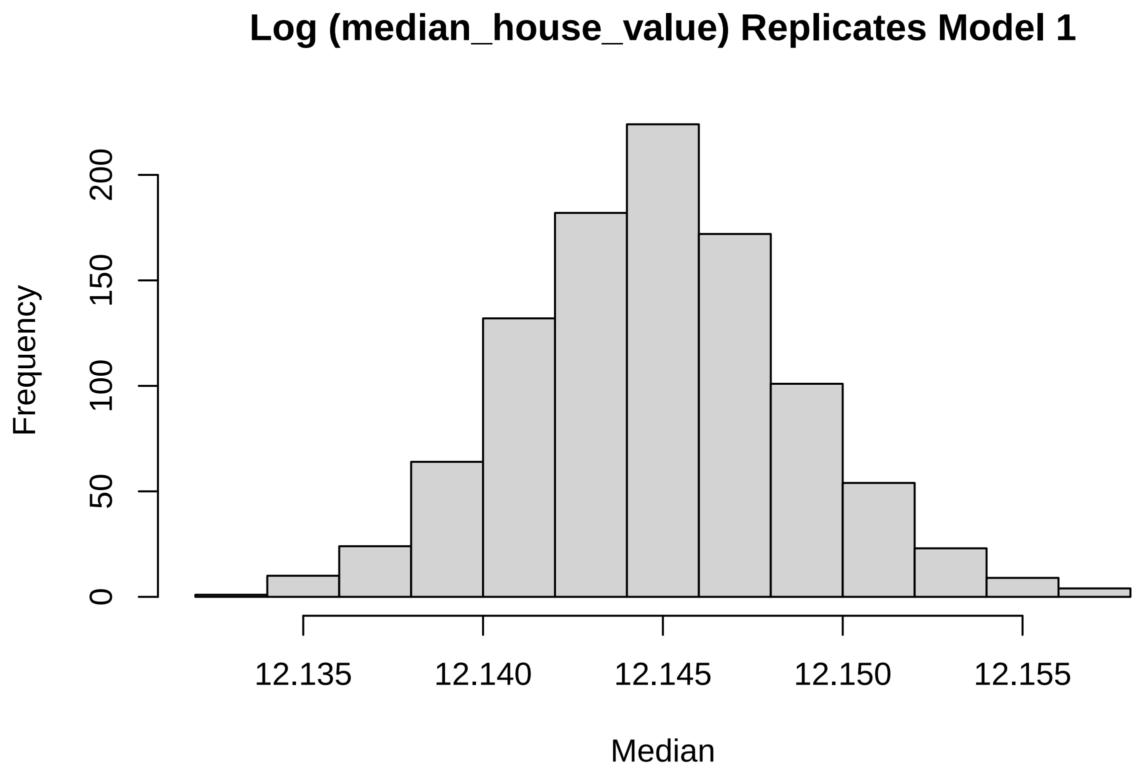
```
## Loading required package: fBasics
```

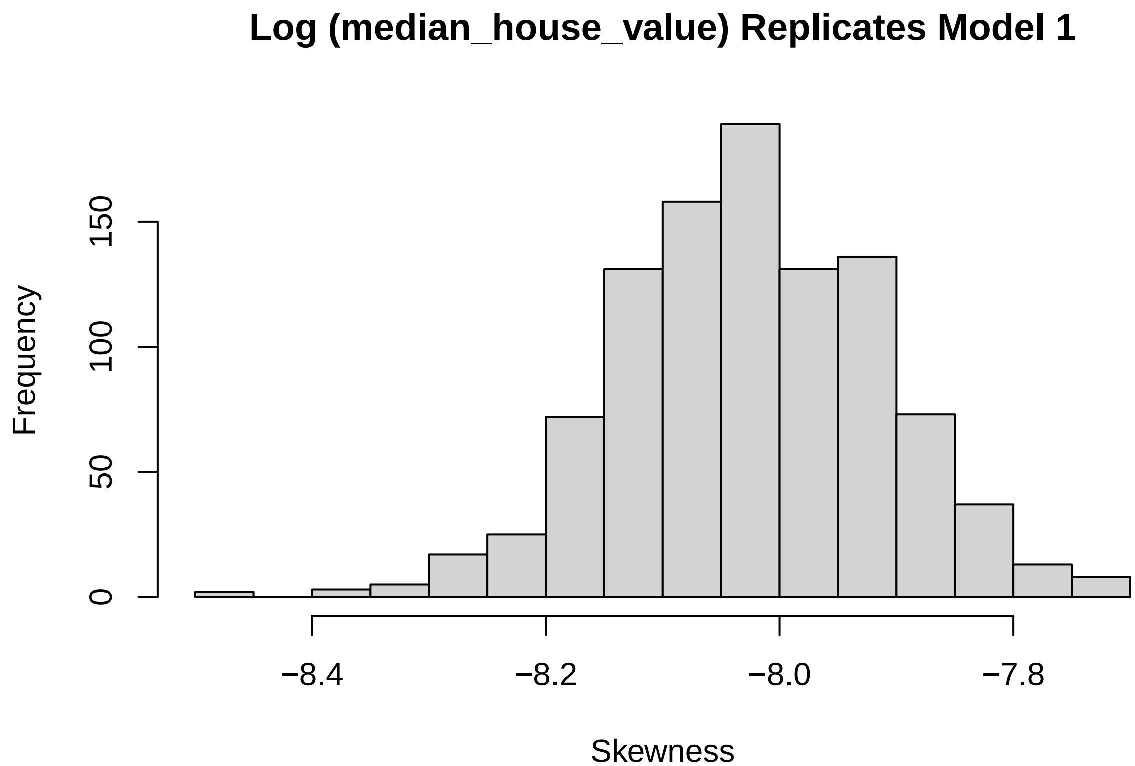
```r
# Function to perform posterior predictive checks on every model
posterior_predictive_checks <- function(model, model_name ,num_replicates = 1000) {
    # Generate replicated datasets
    replicated_datasets <- inla.posterior.sample(model, n = num_replicates)

    # Compute test functions
    min_list <- numeric(num_replicates)
    max_list <- numeric(num_replicates)
    median_list <- numeric(num_replicates)
    skewness_list <- numeric(num_replicates)
    kurtosis_list <- numeric(num_replicates)
    for (i in 1:num_replicates) {
        replicated_data <- replicated_datasets[[i]]$latent
        min_list[i] <- min(replicated_data)
        max_list[i] <- max(replicated_data)
        median_list[i] <- median(replicated_data)
        skewness_list[i] <- skewness(replicated_data)
        kurtosis_list[i] <- kurtosis(replicated_data)
    }
```

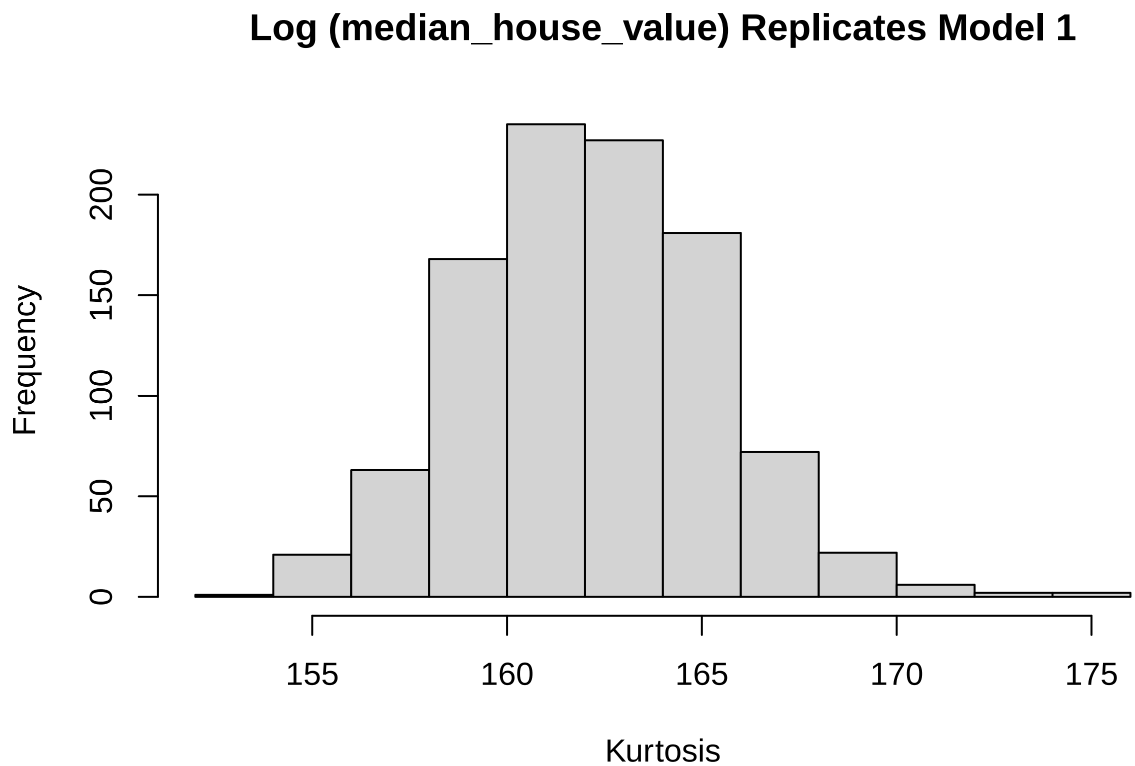
```r
  # Plot the distributions of test functions
  hist(min_list, main = paste("Log(median_house_value) Replicates ", model_name), xlab = "Minimum")
  abline(v=min(housing.training$y),col="red",lwd=5)

  hist(max_list, main = paste("Log (median_house_value) Replicates", model_name), xlab = "Maximum")
  abline(v=max(housing.training$y),col="red",lwd=5)

  # Plot the distributions of test functions
  hist(median_list, main = paste("Log (median_house_value) Replicates", model_name), xlab = "Median")
  abline(v=median(housing.training$y),col="red",lwd=5)

  hist(skewness_list, main = paste("Log (median_house_value) Replicates", model_name), xlab = "Skewne
  abline(v=skewness(housing.training$y),col="red",lwd=5)

  # Plot the distributions of test functions
  hist(kurtosis_list, main = paste("Log (median_house_value) Replicates", model_name), xlab = "Kurtos
  abline(v=kurtosis(housing.training$y),col="red",lwd=5)
}


posterior_predictive_checks(model1, "Model 1",1000) # Model 1 Checks
```



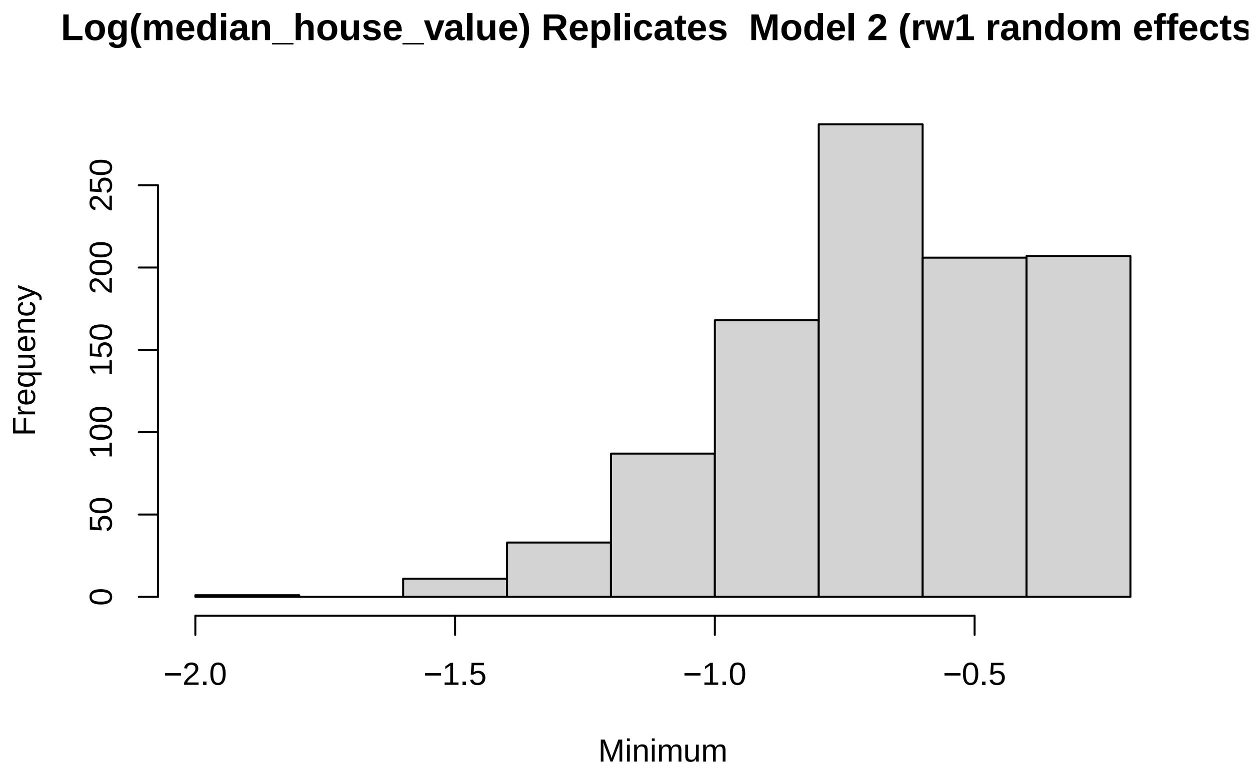

Log (median_house_value) Replicates Model 1
Frequency
0
50
100
150
13.4
13.6
13.8
14.0
14.2
14.4
14.6
Maximum

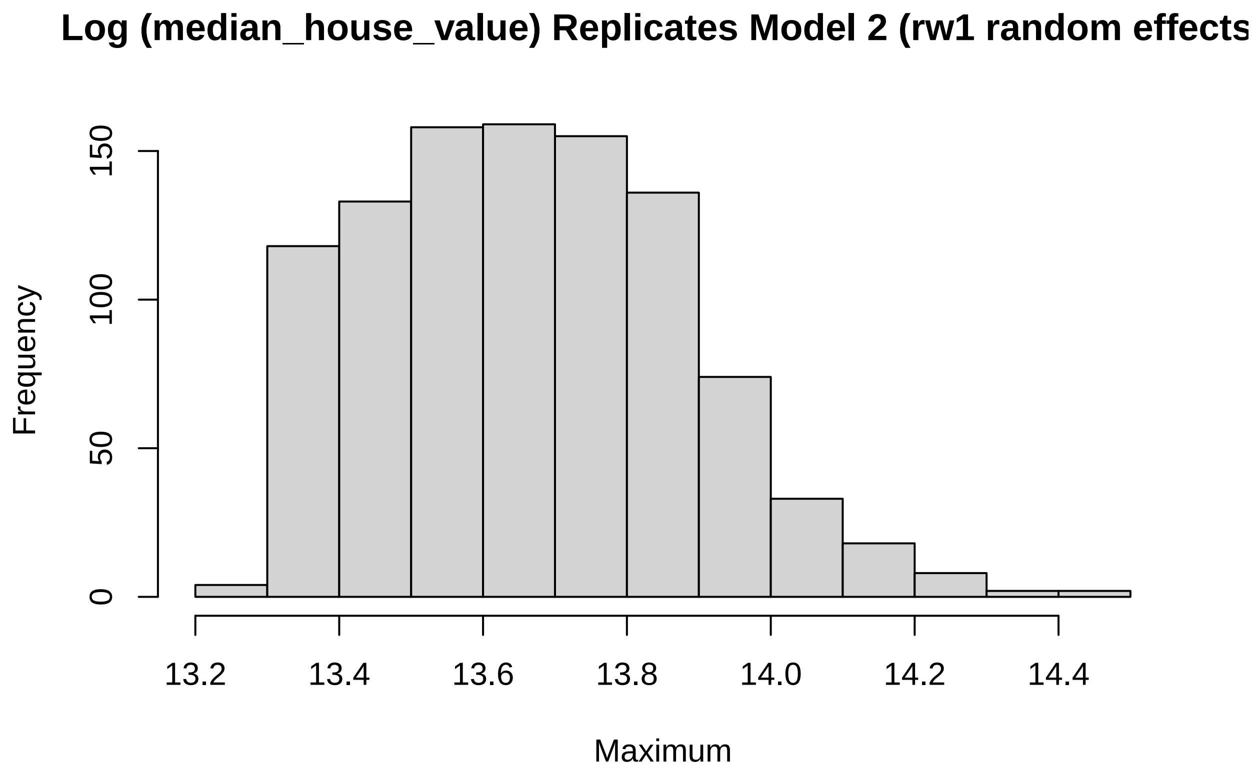

Log (median_house_value) Replicates Model 1
Frequency
0
50
100
150
200
12.135
12.140
12.145
12.150
12.155
Median

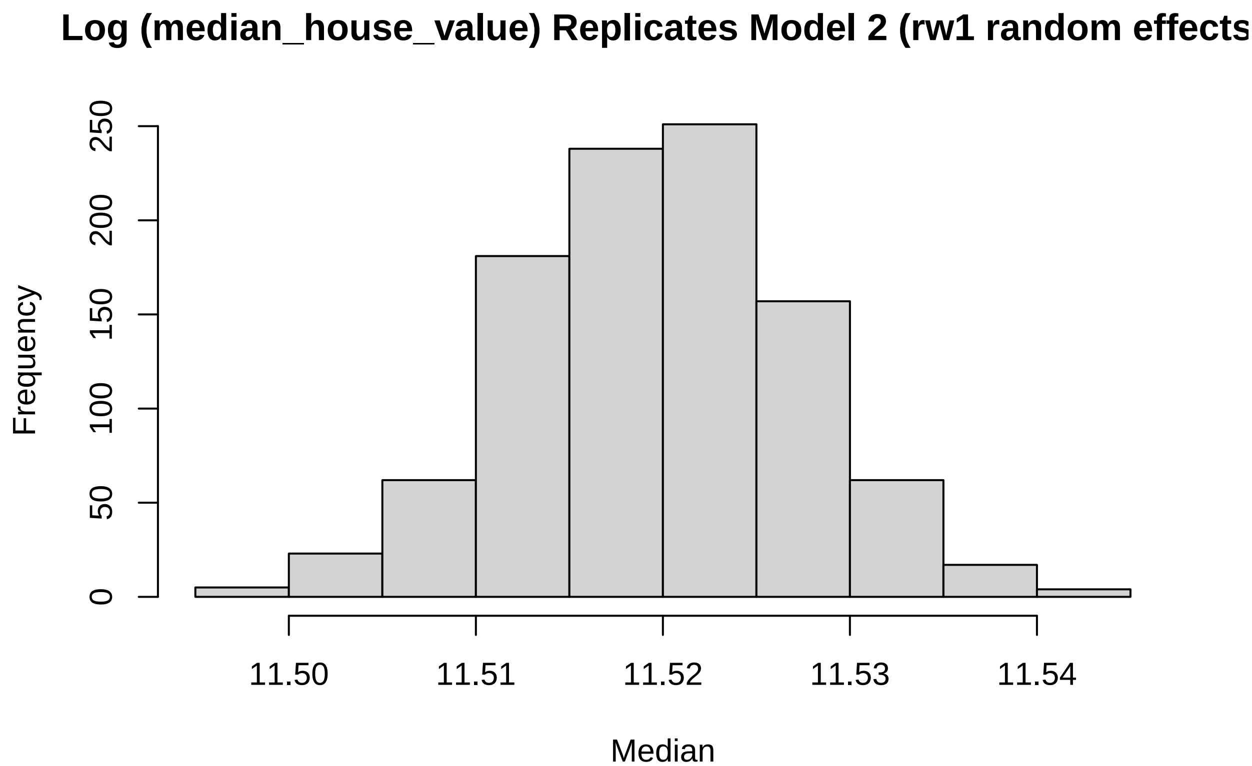

Log (median_house_value) Replicates Model 1
Frequency
0
50
100
150
−8.4
−8.2
−8.0
−7.8
Skewness

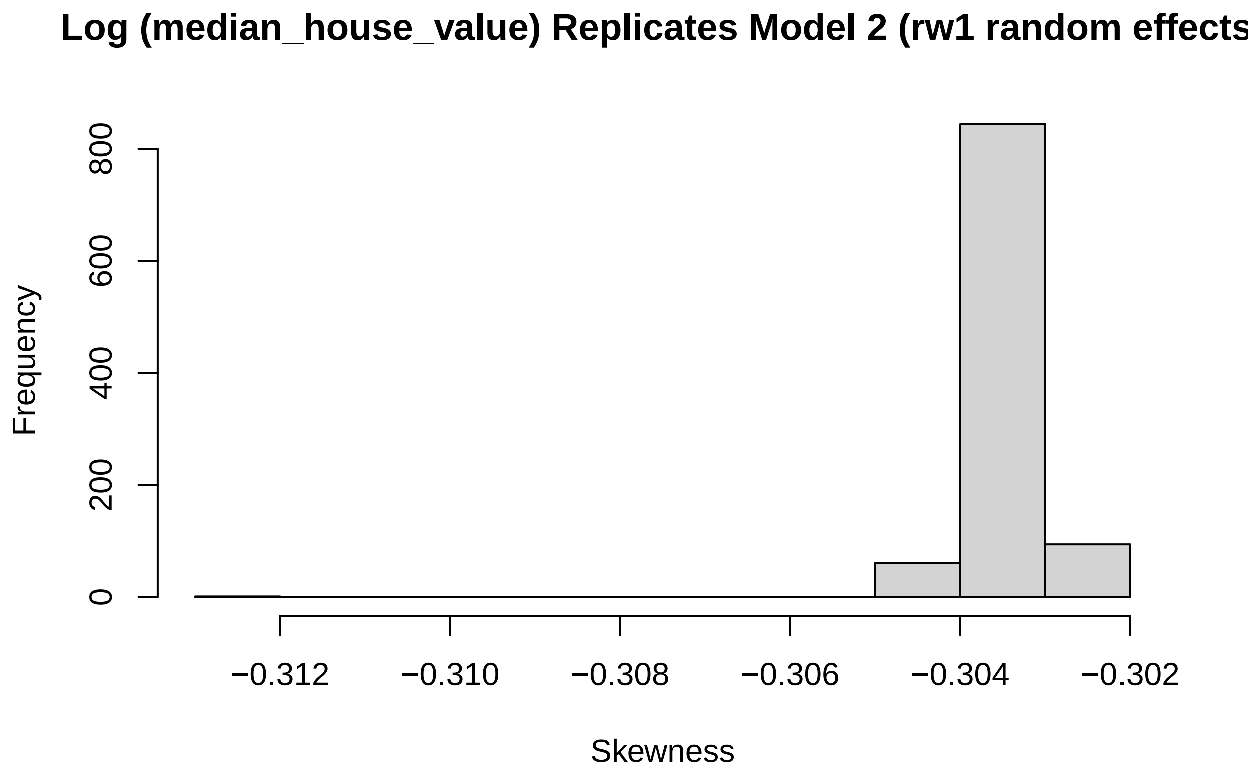


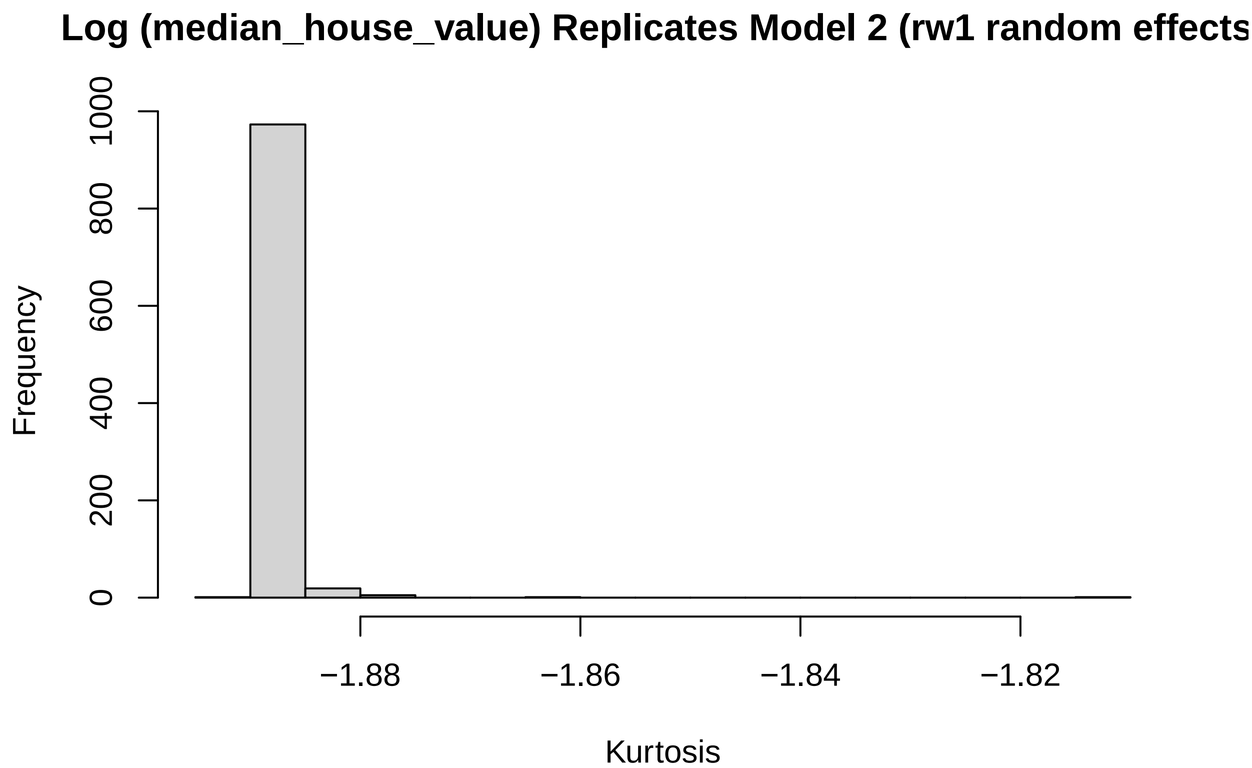

```
posterior_predictive_checks(model2, "Model 2 (rw1 random effects)", 1000) #Model 2
```


Log(median_house_value) Replicates Model 2 (rw1 random effects
Frequency
0
50
100
150
200
250
−2.0
−1.5
−1.0
−0.5
Minimum

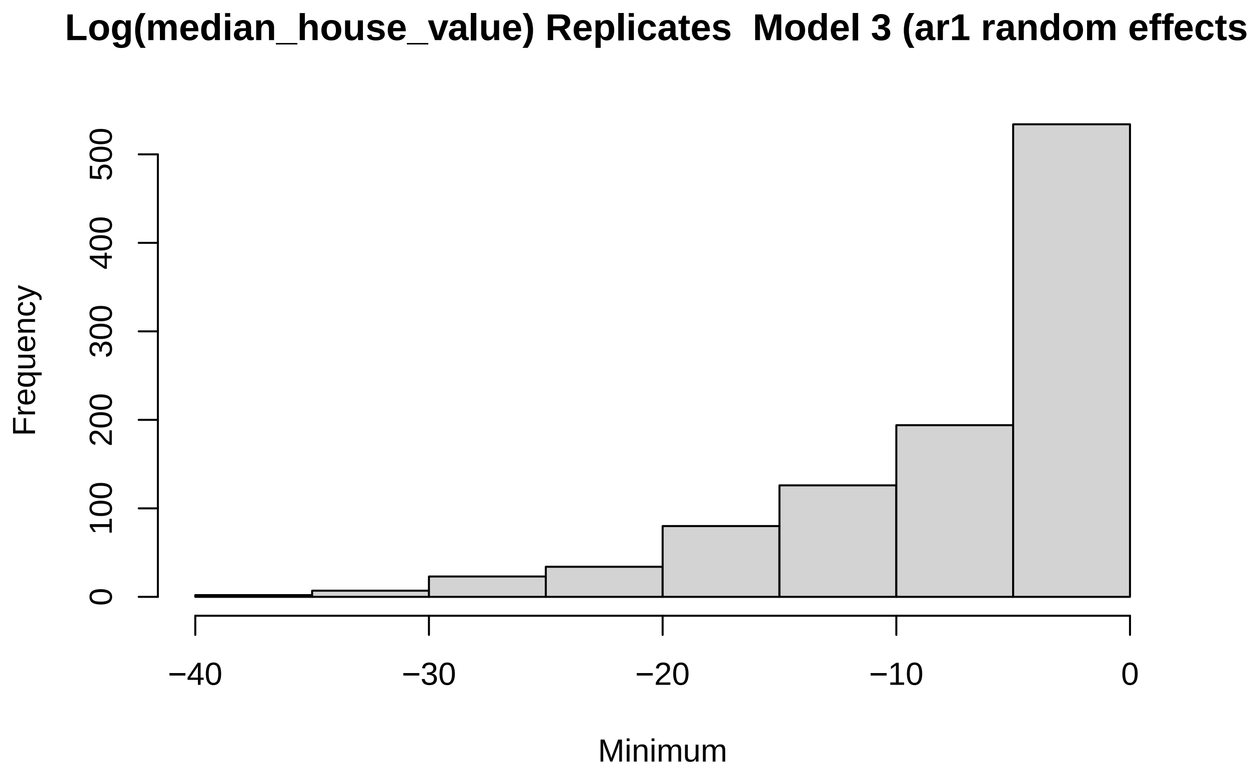

Log (median_house_value) Replicates Model 2 (rw1 random effects
Frequency
0
50
100
150
13.2
13.4
13.6
13.8
14.0
14.2
14.4
Maximum

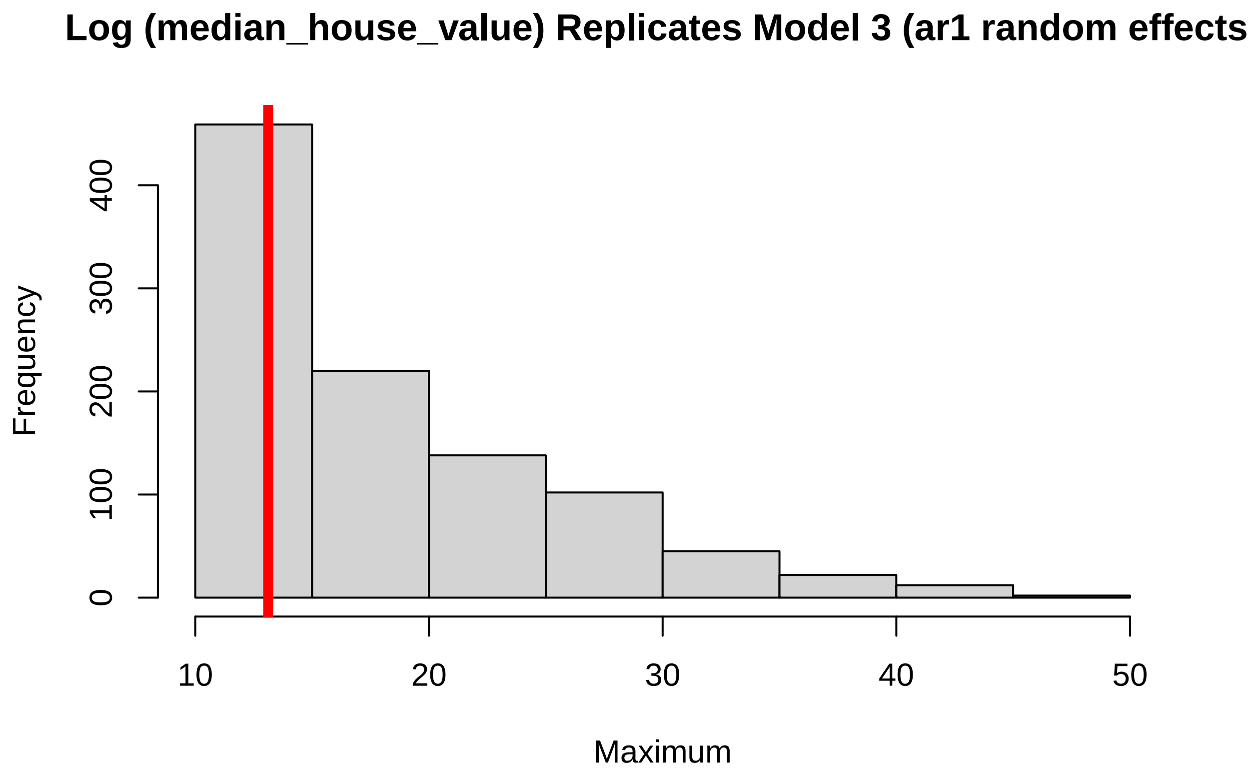

Log (median_house_value) Replicates Model 2 (rw1 random effects
Frequency
0
50
100
150
200
250
11.50
11.51
11.52
11.53
11.54
Median


Log (median_house_value) Replicates Model 2 (rw1 random effects
Frequency
0
200
400
600
800
−0.312
−0.310
−0.308
−0.306
−0.304
−0.302
Skewness

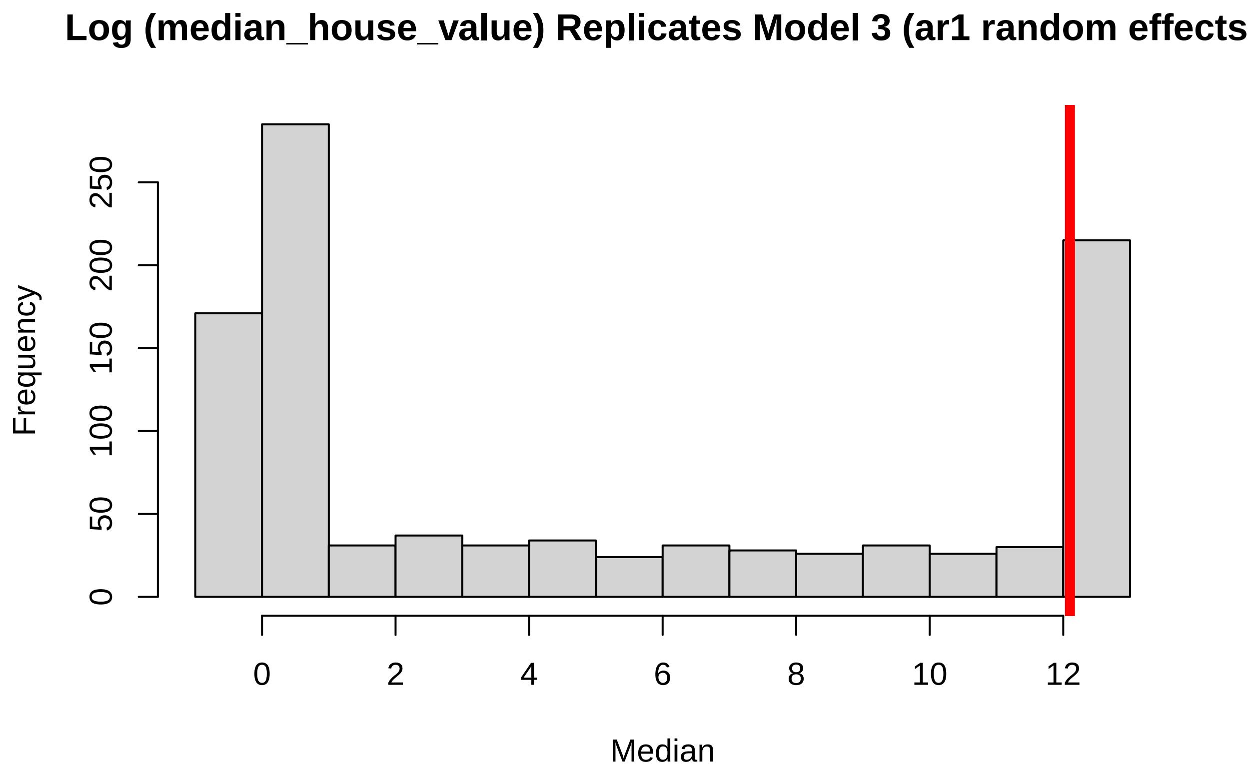


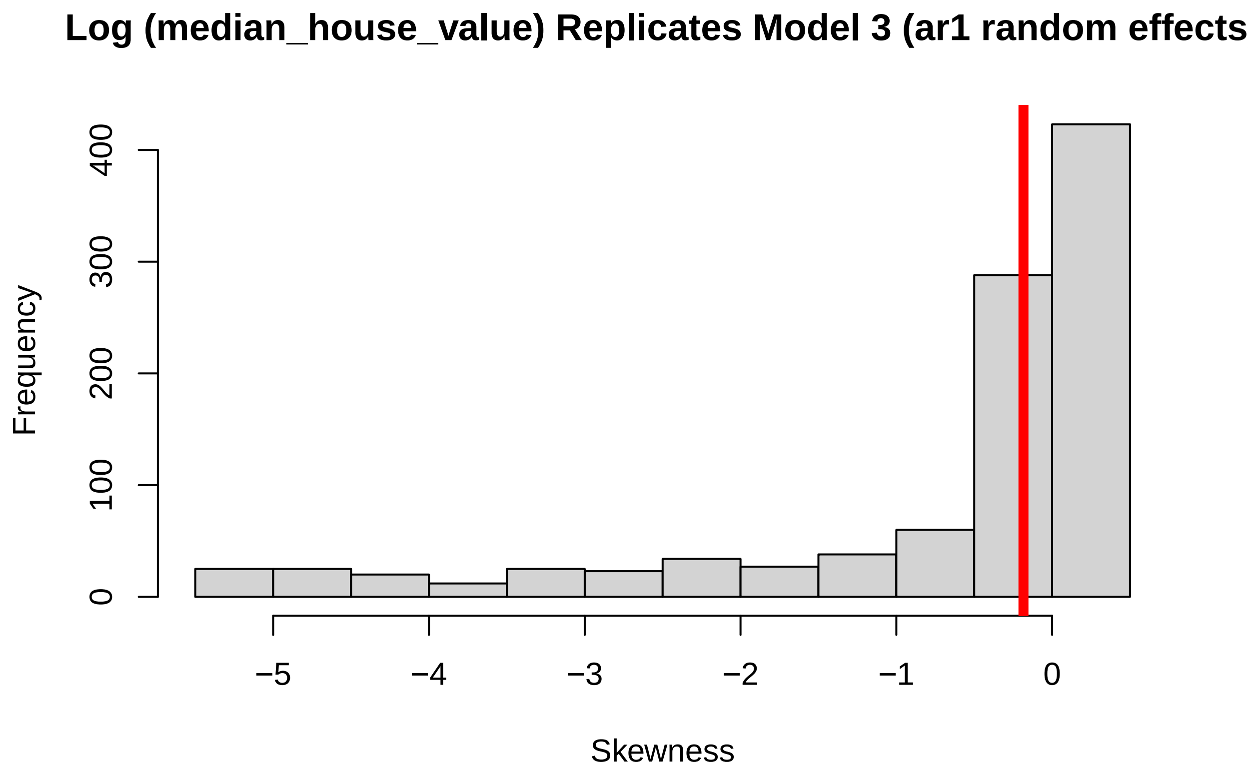

```
posterior_predictive_checks(model3, "Model 3 (ar1 random effects)", 1000) # Model 3
```

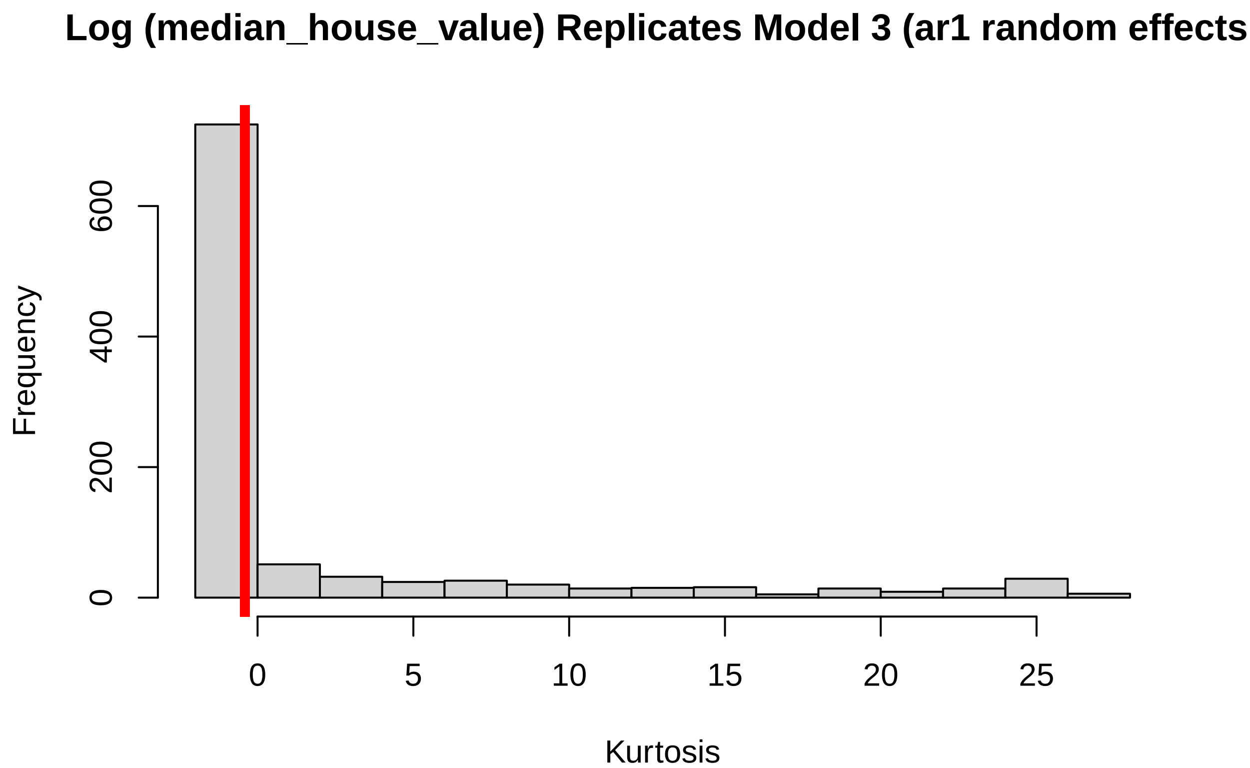

Log(median_house_value) Replicates Model 3 (ar1 random effects
Frequency
0
100
200
300
400
500
−40
−30
−20
−10
0
Minimum



Log (median_house_value) Replicates Model 3 (ar1 random effects
Frequency
0
100
200
300
400
10
20
30
40
50
Maximum


Log (median_house_value) Replicates Model 3 (ar1 random effects
Frequency
0
50
100
150
200
250
0
2
4
6
8
10
12
Median


Log (median_house_value) Replicates Model 3 (ar1 random effects
Frequency
0
100
200
300
400
−5
−4
−3
−2
−1
0
Skewness



Analysing the histograms, it is quite clear that the model 3 is able to capture the complexities better than that of the model 1 and model 2. The test functions show that the model 3 is somehow close to the real value of the test, when visualised from the histograms. Also, the model 3 is not able to capture the Minimum, which is evident from the test function. this may be due to presence of some outliers in the dataset.

Hence, from the analysis of all the three models 1,2,3, it is clear that the model 3 with SPDE random effects over location, are best capturing the intrecacies of the data and modelling it with good performance. This can be made finer by tuning and adjusting the parameters more, or performing grid search over various priors and meshes.