`geoR` uses the function `krige.bayes` to fit and provide prediction of a Gaussian process using Bayesian methods. The strategy of `krige.bayes` is that of discretizing the distribution of the range parameter as well as the *relative nugget* defined as $\gamma^2 = \tau^2/\sigma^2$.

`geoR` uses the function `krige.bayes` to fit and provide prediction of a Gaussian process using Bayesian methods. The strategy of `krige.bayes` is that of discretizing the distribution of the range parameter as well as the *relative nugget* defined as $\gamma^2 = \tau^2/\sigma^2$.

Sampling of the posterior distribution is achieved by direct sampling of the parameters using the block conditioning

$$\pi(\beta, \psi, \tau^2, \gamma^2|X) = \pi(\beta|\tau^2, \psi, \gamma^2, X)\pi(\tau^2|\psi, \gamma^2, X)\pi(\psi, \gamma^2|X)$$

The model to be fitted in `bayes.krige` can be specified with the function `model.control`. The model defaults to an exponential correlation with no nugget and no Box and Cox transformation.

The model to be fitted in `bayes.krige` can be specified with the function `model.control`. The model defaults to an exponential correlation with no nugget and no Box and Cox transformation.

Prior distributions for the parameters can be specified using the function `prior.control`.

Controlling `geoR` input/output

The model to be fitted in `bayes.krige` can be specified with the function `model.control`. The model defaults to an exponential correlation with no nugget and no Box and Cox transformation.

Prior distributions for the parameters can be specified using the function `prior.control`.

The output from `bayes.krige` can be specified using the function `output.control`. This includes the number of samples in the posterior and the predictive posterior distributions.

Set the priors:

```
prior.swiss=prior.control(phi.discrete=seq(7.5,150,len=20),
phi.prior='reciprocal',
tausq.rel.prior ='uniform',tausq.rel.discrete=
seq(0,0.5,len=11))
```

Set The model:

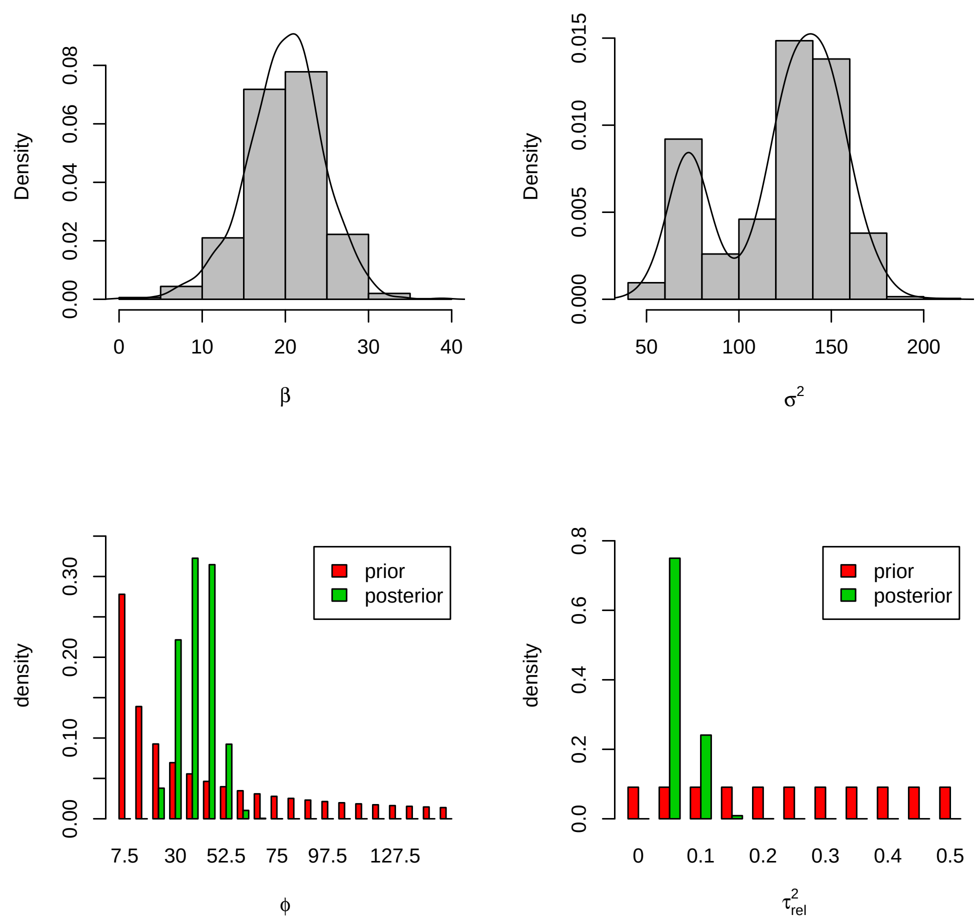
```
model.swiss=model.control(kappa=1,lambda=0.5)
```

Fit the model:

```
bayes.fit=krige.bayes(sic.all,prior=prior.swiss,
model=model.swiss)
```

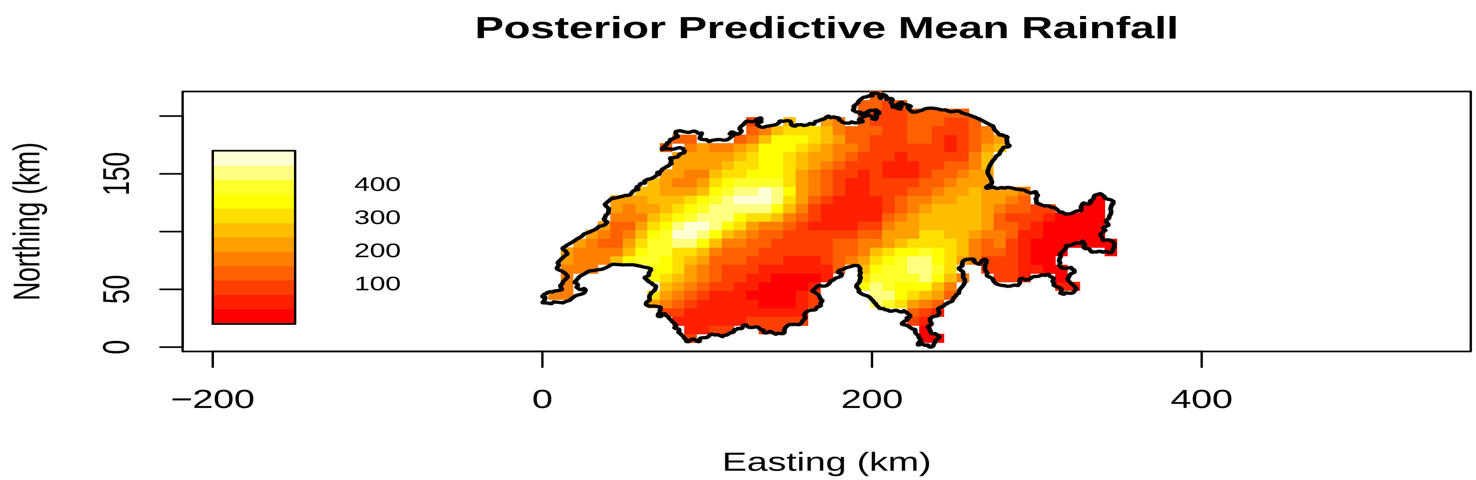
Posterior Results

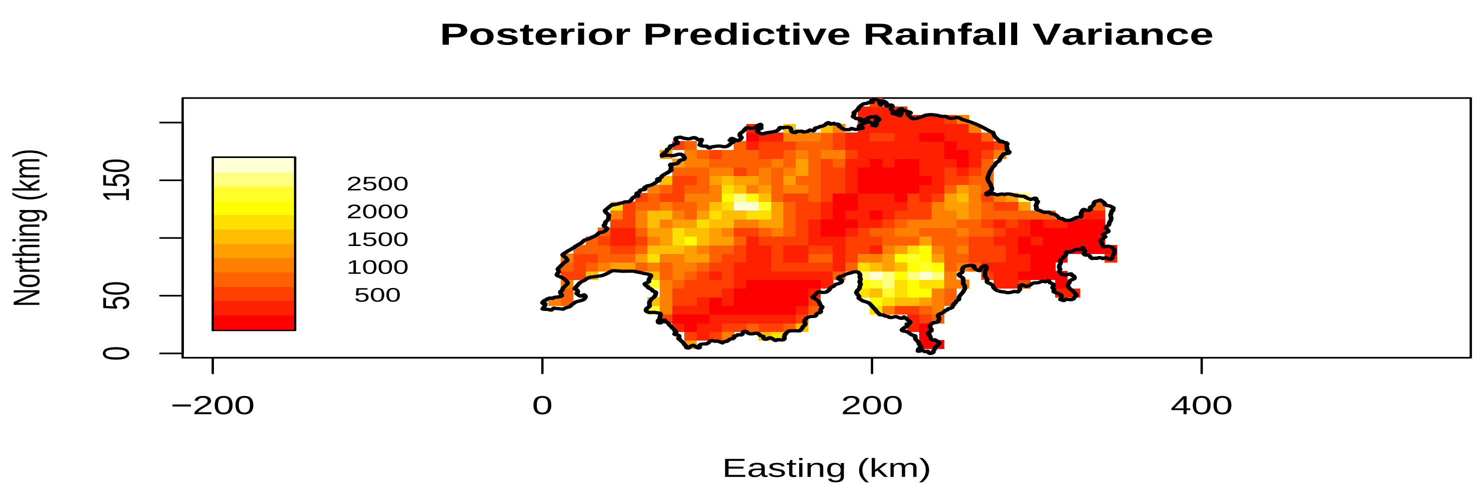
Summary of 1,000 samples from the posterior distribution for the Swiss data example obtained with `hist(bayes.fit, pars=c('beta', 'sigmasq'))` `plot(bayes.fit, col=2:3)`



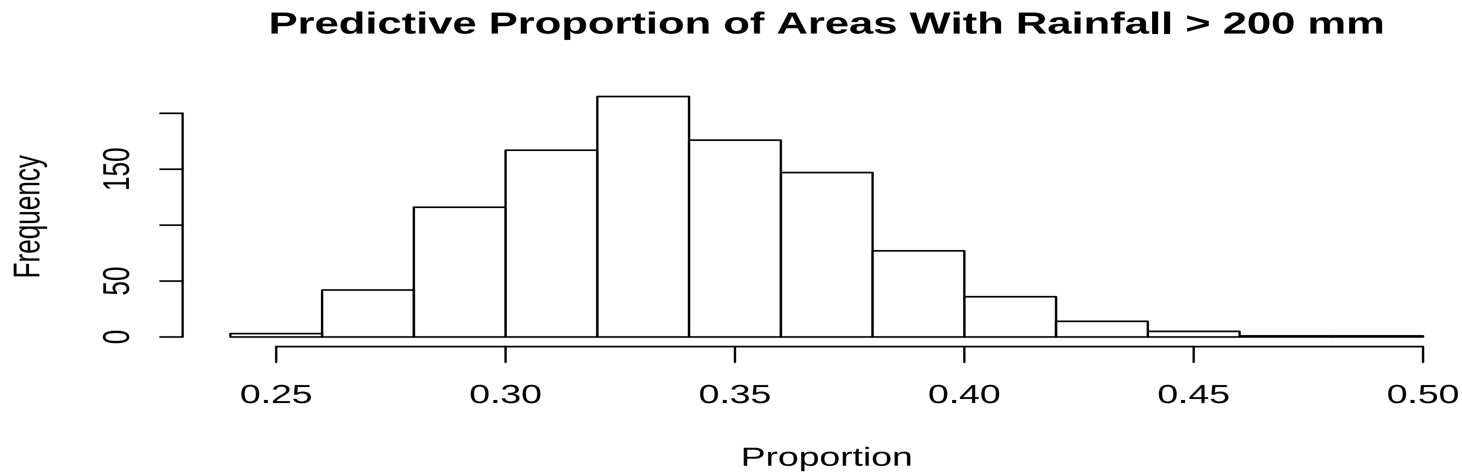
```
> summary(bayes.fit$posterior$sample$beta)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4443 17.0900 20.2000 19.9100 22.7900 38.7900
> summary(bayes.fit$posterior$sample$sigmasq)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  43.53   90.85  130.50  122.10  146.20  204.10
```

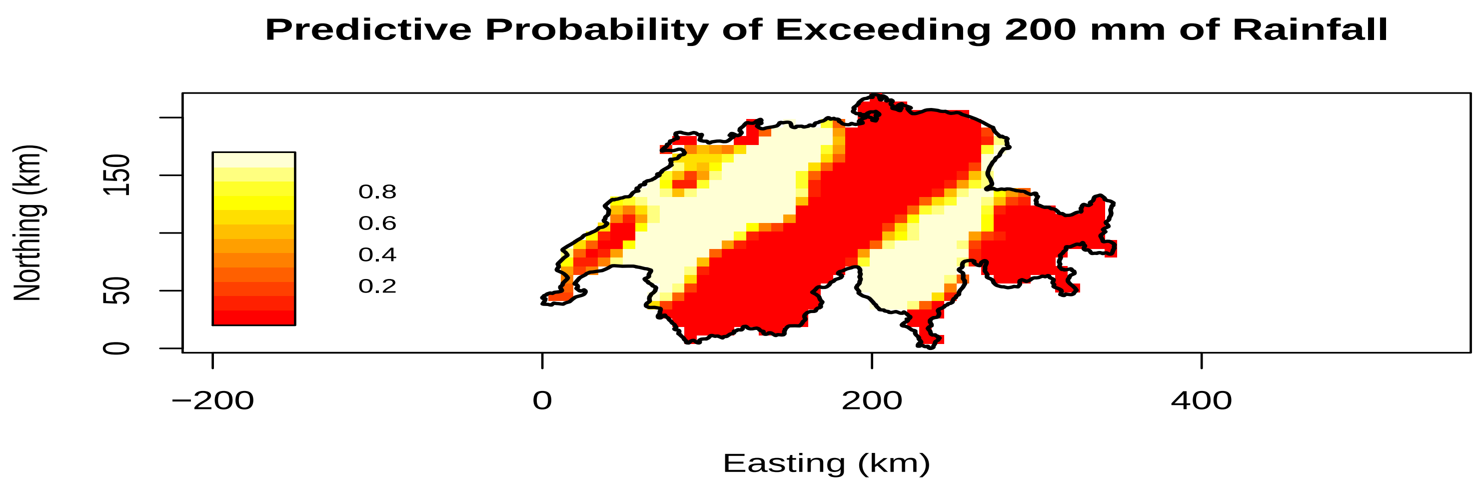
```
image(bayes.fit,borders=sic.borders,
x.leg=c(-200,-150),y.leg=c(20,170),ylab='Northing (km)',
xlab='Easting (km)',vertical=T)
title('Posterior Predictive Mean Rainfall')
#Variance
image(bayes.fit,borders=sic.borders,
val='variance',x.leg=c(-200,-150),
y.leg=c(20,170),ylab='Northing (km)',
xlab='Easting (km)',vertical=T)
title('Posterior Predictive Rainfall Variance')
```


Posterior Predictive Mean Rainfall
Northing (km)
0
50
150
400
300
200
100
−200
0
200
400
Easting (km)


Posterior Predictive Rainfall Variance
Northing (km)
0
50
150
2500
2000
1500
1000
500
−200
0
200
400
Easting (km)

```
#Proportion of area with more than 200 mm of rainfall
A200=apply(bayes.fit$predictive$simulations,2,function(y)
sum(y>200)/length(y))
hist(A200,xlab='Proportion',
main='Predictive Proportion of Areas With Rainfall > 200 mm')
#Probabilities of exceeding 200 mm
prob=apply(bayes.fit$predictive$simulations,1,function(y)
sum(y>200)/length(y))
image(bayes.fit,,borders=sic.borders,
x.leg=c(-200,-150),y.leg=c(20,170),ylab='Northing (km)',
xlab='Easting (km)',vertical=T,val=prob)
title('Predictive Probability of Exceeding 200 mm of Rainfall')
```


Predictive Proportion of Areas With Rainfall > 200 mm
Frequency
0
50
150
0.25
0.30
0.35
0.40
0.45
0.50
Proportion


Predictive Probability of Exceeding 200 mm of Rainfall
Northing (km)
0
50
150
0.8
0.6
0.4
0.2
−200
0
200
400
Easting (km)

The core function in `spBayes` is `spGGT` and NOT `ggt.sp` as described in the tech. report!

Set the priors using the function `prior`:

```
phi.prior=prior(dist="UNIF",a=7.5,b=150) #range
Psi.prior=prior(dist="IG",shape=50,scale=10) #nugget
K.prior=prior(dist="IG",shape=1,scale=1) #sill
nu.prior=prior(dist="FIXED") #smoothness
```

Set The model:

```
var.update.control=
  list("K"=list(starting=12000,tuning=0.1,prior=K.prior),
       "Psi"=list(starting=.3,tuning=0.5,prior=Psi.prior),
       "phi"=list(starting=57,tuning=0.5,prior=phi.prior),
       "nu"=list(starting=1,tuning=0.5,prior=nu.prior))
beta.control=list(update="GIBBS",prior=prior(dist="FLAT"))
```

Run the model for the Swiss rainfall data

```
run.control=list("n.samples"=1000)
bayes.fit=spGGT(formula=sic.all$data~1,coords=sic.all$coords,
  run.control=run.control,var.update.control=var.update.control,
  beta.update.control=beta.control,cov.model="matern")
```

```
summary(bayes.fit$p.samples[-seq(200),])
       K                  Psi                   Phi              (Intercept)
 Min.   :11023   Min.   :0.1337    Min.   :53.55   Min.   :169.4
 1st Qu.:11984   1st Qu.:0.1842    1st Qu.:55.93   1st Qu.:180.4
 Median :12510   Median :0.1981    Median :57.12   Median :183.9
 Mean   :12542   Mean   :0.2019    Mean   :57.72   Mean   :184.2
 3rd Qu.:13077   3rd Qu.:0.2192    3rd Qu.:58.94   3rd Qu.:187.8
 Max.   :14892   Max.   :0.3224    Max.   :63.47   Max.   :200.9
```

```
gr=pred_grid(sic.borders,by=7.5) #using geoR
bayes.predict=spPredict(bayes.fit,gr,start=200,
  pred.covars=matrix(rep(1,dim(gr)[1])))
#... forever and a day later:
# The names of the components of the resulting object
# do not coincide with those in the help file!
```

Model Assessment

The methods for assessing the goodness of fit a Bayesian spatial model follow along the lines of Bayesian predictive checking that is common for the Bayesian approach. The fundamental idea is to compare the posterior predictive distribution to observed data.

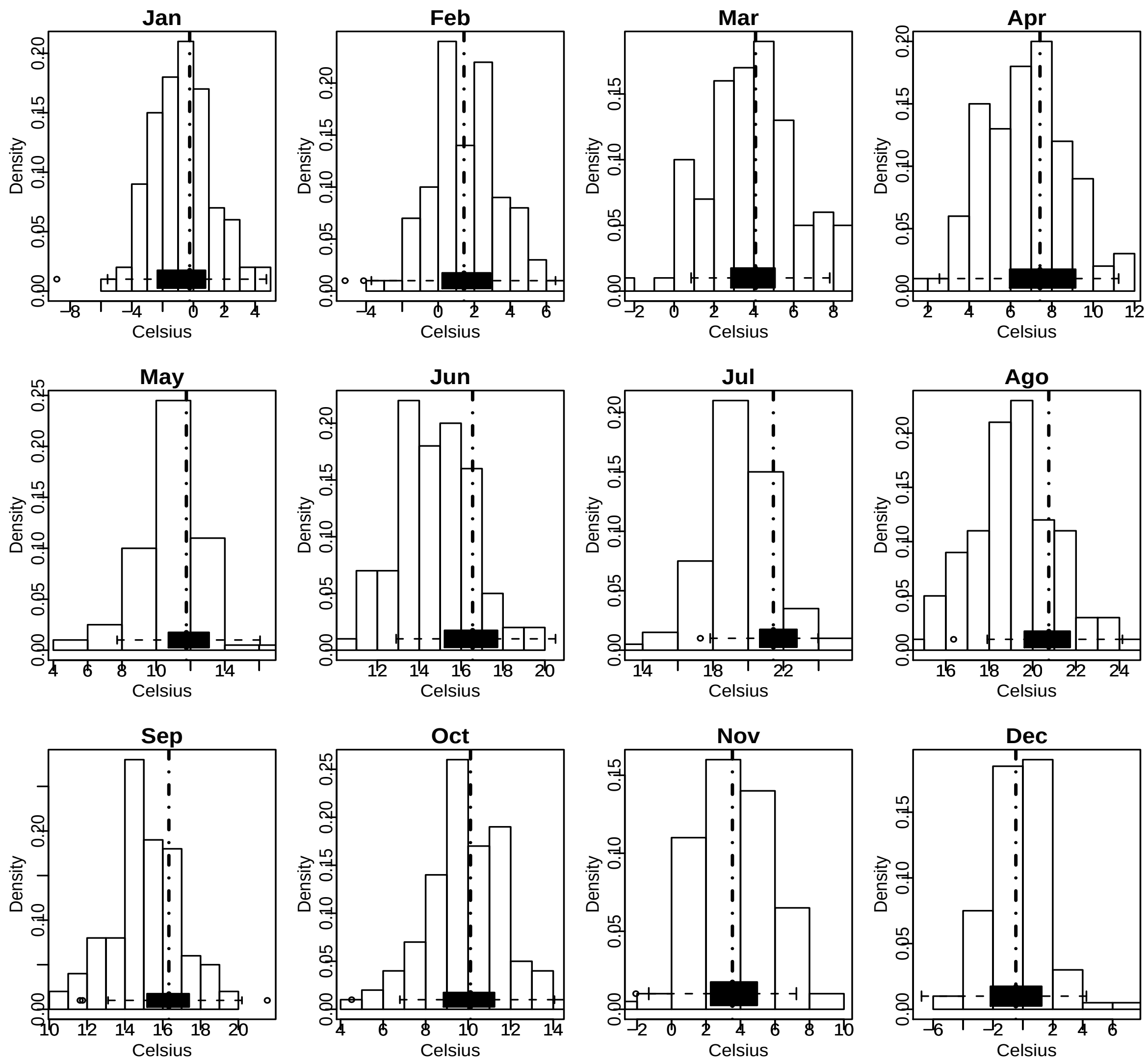
Model Assessment

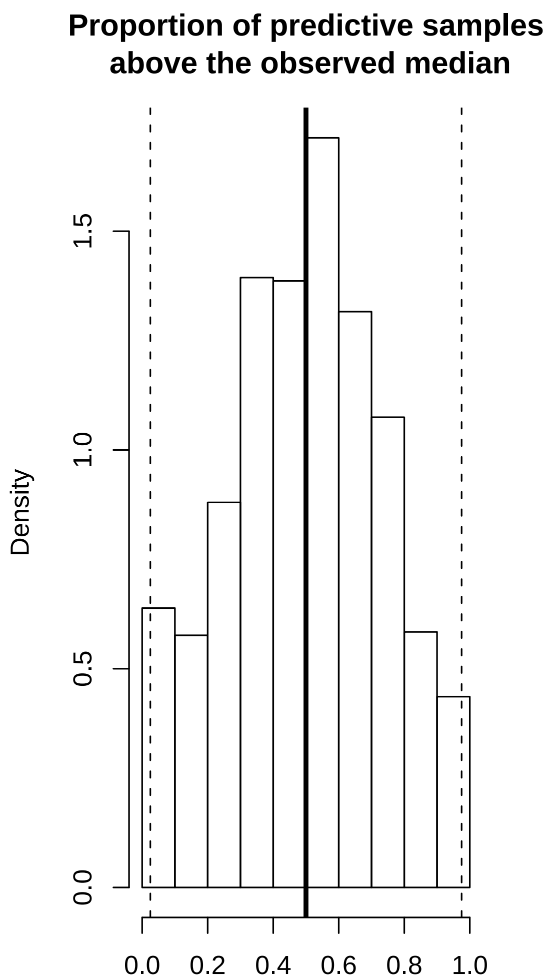
The methods for assessing the goodness of fit a Bayesian spatial model follow along the lines of Bayesian predictive checking that is common for the Bayesian approach. The fundamental idea is to compare the posterior predictive distribution to observed data.

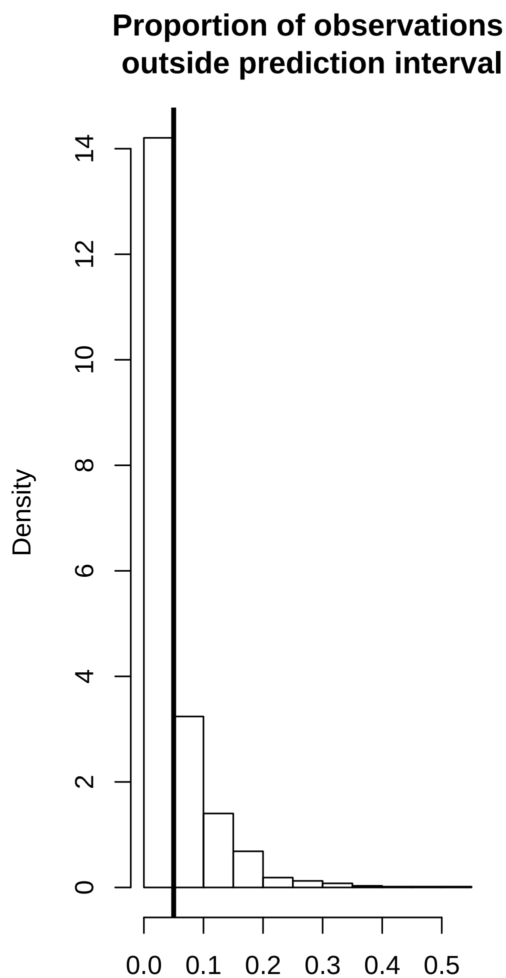
- Select some locations for validation
- Fit the model to the remaining data set
- Obtain the posterior predictive for the validation locations
- Compare to observed data

California Temperature Data

Posterior predictive distribution for temperature at Bowman Dam. Horizontal boxplots correspond to 50 years of observations







Other summaries of the predictive distribution can also be explored. Warning: In these plots we pooled together samples from all other locations. So histograms are produced with dependent data

The posterior predictive means can be use to obtain residuals for each of the locations. Posterior predictive samples can be used to produce distribution for such residuals. A leave one out analysis can be used for a systematic exploration of possible spatial patterns in the residuals.

The posterior predictive means can be use to obtain residuals for each of the locations. Posterior predictive samples can be used to produce distribution for such residuals. A leave one out analysis can be used for a systematic exploration of possible spatial patterns in the residuals.

For a single goodness of fit statistics:

- For validation locations $s_1, \ldots, s_k$ define $Z = (z(s_1) \ldots z(s_k))'$ and z the corresponding observations
- Obtain samples $Z^{(1)} \ldots Z^{(l)}$ of $p(Z|X)$.
- Let $\overline{Z}$ and $\hat{\Sigma}$ be the sample mean and variance respectively
- $D^2 = (z - \overline{Z})'\hat{\Sigma}^{-1}(z - \overline{Z}) \sim \chi^2_k$