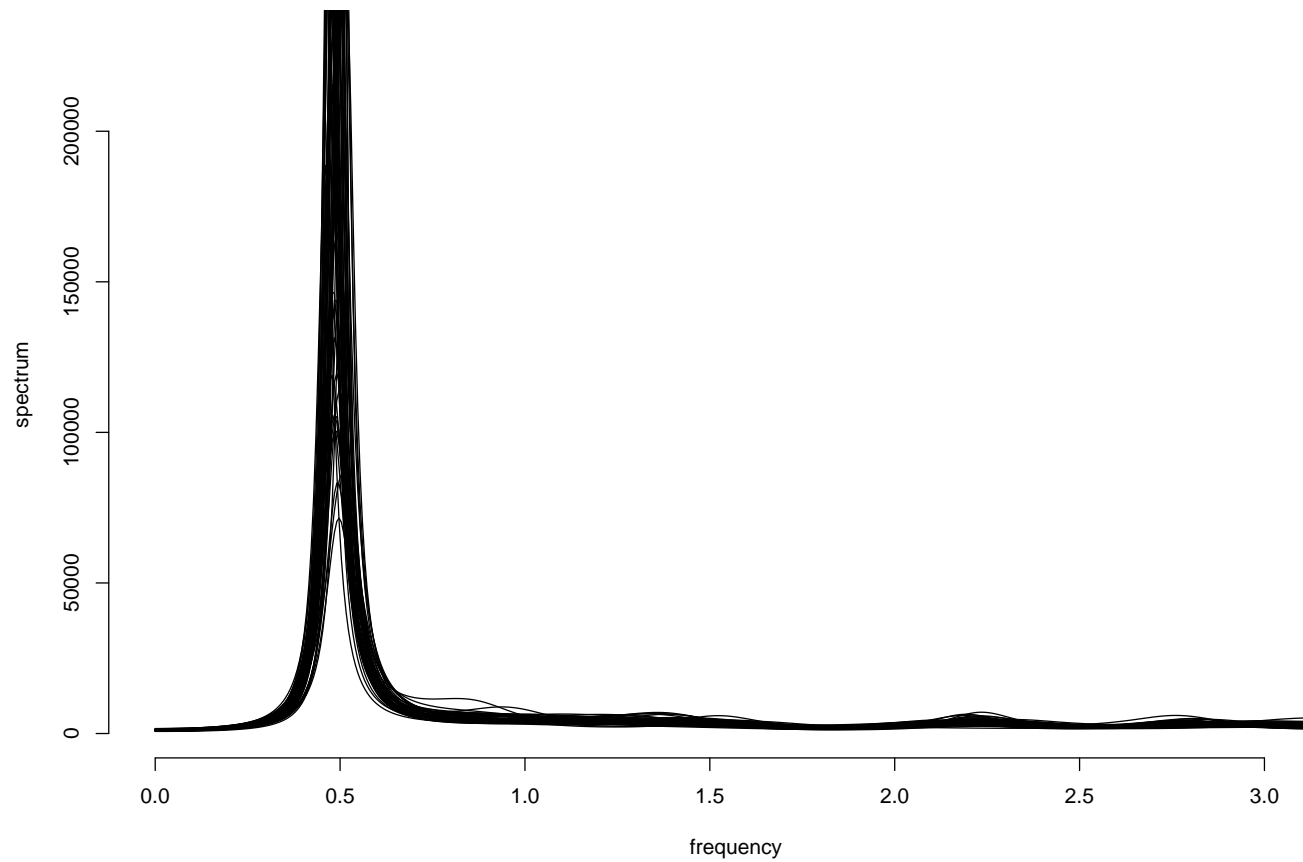


## Comment about AR spectral estimation

- Usually an estimate is produced by computing the AR theoretical spectrum at  $(\hat{\phi}, \hat{\sigma}^2)$ .
- With our Monte Carlo simulation approach, for every draw  $(\phi, \sigma^2)$ , we can compute the spectrum and obtain a draw for  $f(\omega)$ .
- Typically the mean of these draws will be similar to the spectrum at  $(\hat{\phi}, \hat{\sigma}^2)$ .
- With this posterior simulation, we have the possibility of computing *quantiles*, *probability intervals* or simply a “band” for the spectral density.
- The purpose of the “band” is to get an idea of the uncertainty of the estimation.

- EEG example. The next figure shows several spectrum curves for 50 draws of  $(\phi, \sigma^2)$ .
- Recall that the object *phsim* has the draws of  $\phi$  coefficients and *sigma2*, the draws for the variance of the error term  $\sigma^2$ .

# 50 posterior samples of AR(10) spectrum



```
a=ar(eeg,order=10,aic=F)
a$ar=as.vector(apply(phsim,2,mean))
a$var=mean(sigma2)
x=spec.ar(a,n.freq=500,plot=F)
plot(2*pi*x$freq,x$spec,type="l",axes=F)
axis(1)
axis(2)
for(i in 1:50){
    a$ar=as.vector(phsim[i,])
    a$var=sigma2[i]
    x=spec.ar(a,n.freq=500,plot=F)
    lines(2*pi*x$freq,x$spec)
    print(i)
}
```

## Portmanteau lack of fit test

- For this test we need to consider the estimated residuals for the AR model  $\hat{\epsilon}_t = x_t - \sum_{j=1}^p \hat{\phi}_j x_{t-j}$  where  $\hat{\phi}_j$  is some estimator of the model parameters.
- The purpose of this test is to determine if the residuals are correlated or not..
- The null hypothesis is  $H_o : \rho_1 = \rho_2 = \dots = \rho_K = 0$
- The proposed test statistic is:

$$Q = n(n + 2) \sum_{k=1}^K (n - k)^{-1} \hat{\rho}_k^2$$

where  $\hat{\rho}_k$  is the sample ACF of the estimated residuals and  $K$  is a fixed integer.

- The paper by Ljung and Box (1978), “On a measure of lack of fit in time series models”, *Biometrika*, **65**, 297-303 shows that under the null hypothesis,  $Q$  approximately follows a chi-square distribution with  $K - (p + 1)$  degrees of freedom or  $Q \sim \chi^2_{K-(p+1)}$

- The testing procedure is: reject the null hypothesis at the  $\alpha$  level if

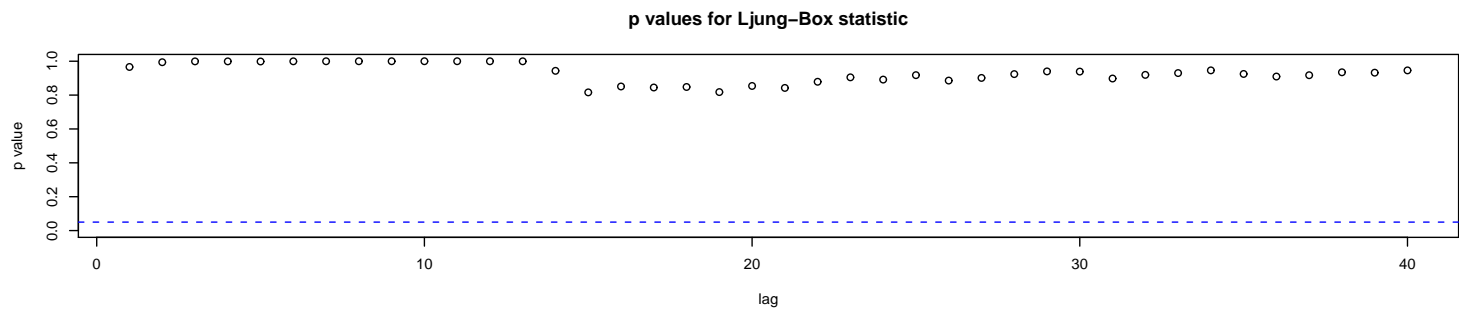
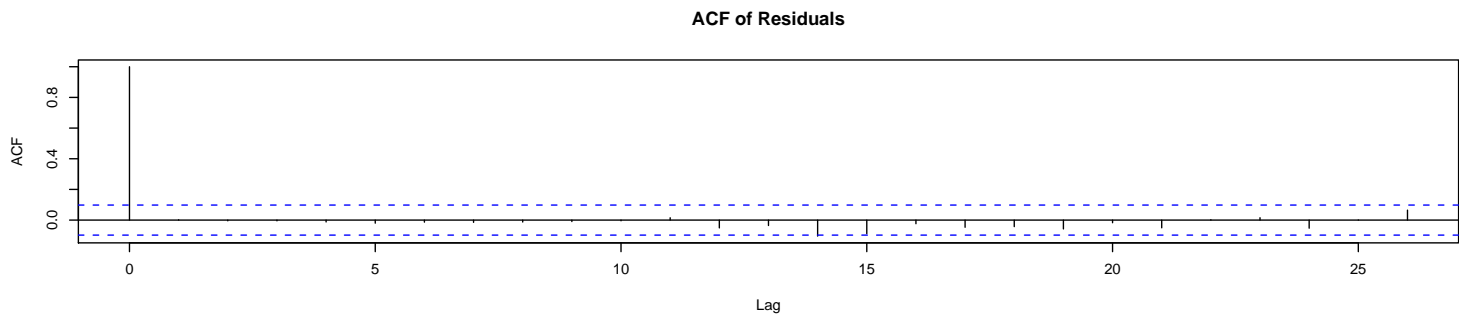
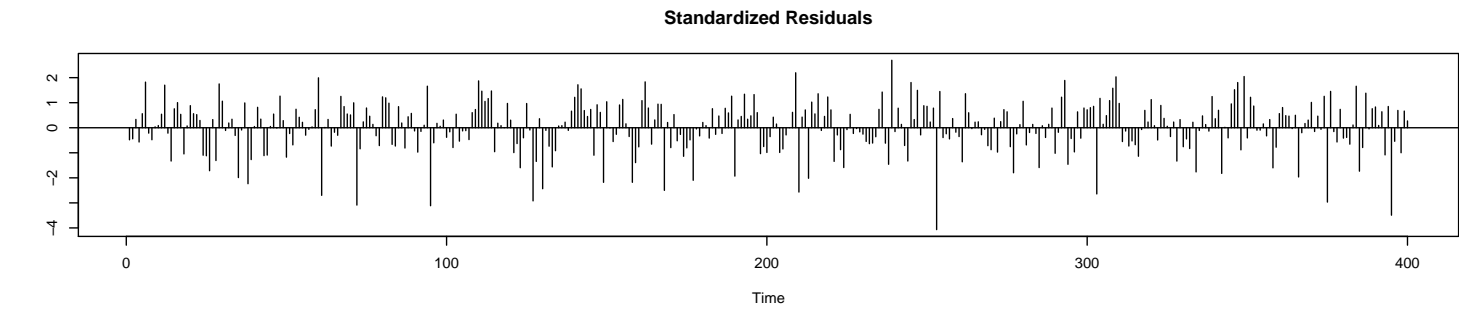
$$Q > \chi^2_{K-(p+1)}(1 - \alpha).$$

where  $\chi^2_{K-(p+1)}(1 - \alpha)$  is the  $(1 - \alpha)$  quantile of the chi-square distribution with  $K - (p + 1)$  degrees of freedom.

- A problem with this test is that there is no formal rule to select the value  $K$ .

- A common approach is to compute the p-value of test for different values of  $K$ .

```
fit=arima(eeg,order=c(10,0,0))  
tsdiag(fit)
```





## Model order via likelihood approaches: AIC, BIC

- We want to define a criteria that allows to select the order  $p$  of an AR process.
- We are thinking of the AR model as a linear regression model with  $p$  covariates.
- As  $p$  increases the likelihood (or log-likelihood) of the model evaluated at the MLE  $(\hat{\phi}, s^2)$  also increases.
- However, as  $p$  increases we may have high autocorrelations of regressors.
- A penalty function could be added to the likelihood function to compensate for more parameters in the model.
- A general selection criteria is to find the value of  $p$  such

that minimizes

$$-2\log[L(\hat{\phi}, s^2)] + f(p)$$

where  $L(\cdot)$  is the likelihood function of the regression model and  $f(\cdot)$  is a penalty function.

- This penalty function  $f(p)$  is assumed to be an increasing function of  $p$ .
- Since we are working with a Normal linear model, we can show that

$$-2\ln[L(\hat{\phi}, s^2)] = m(\log(2\pi + 1)) + m\log(s_p^2)$$

where  $m = n - p$  is the length of the response vector

- In fact, for the AR model  $x = F\phi + \epsilon$ , the likelihood

function

$$L(\phi, s^2) = \left( \frac{1}{2\pi\sigma^2} \right)^{m/2} \exp \left( -\frac{1}{2\sigma^2} (x - F\phi)' (x - F\phi) \right)$$

- Recall that the MLE,  $\hat{\phi} = (F'F)^{-1}F'x$  and  $s^2 = (x - F\hat{\phi})'(x - F\hat{\phi})/m$  and so

$$L(\hat{\phi}, s^2) = \left( \frac{1}{2\pi s^2} \right)^{m/2} \exp \left( -\frac{m}{2} \right)$$

- The first term of  $-2\ln[L(\hat{\phi}, s^2)]$  does not depend on  $p$ .
- The criteria reduces to find the value of  $p$  for which

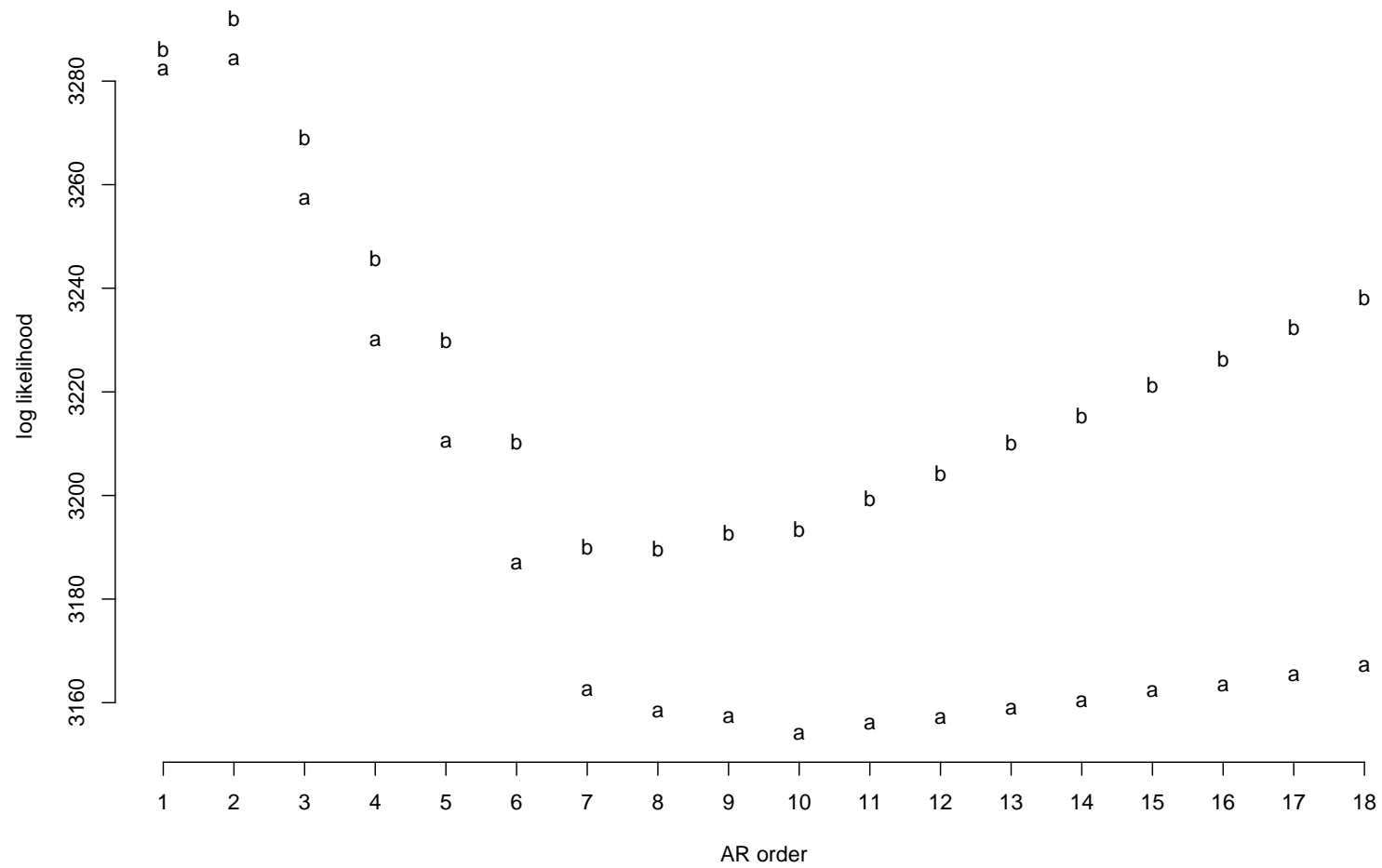
$$n\log(s_p^2) + f(p)$$

is minimum.

- The evaluation must be based on a common sample size. We

fix a maximum order  $p^*$  and fit AR models for values of  $p \leq p^*$  based only on  $n^* = n - p^*$  observations.

- Then we compute  $n^* \log(s_p^2) + f(p); p = 0, 1, \dots, p^*$  and find the max over the range  $0, 1, \dots, p^*$
- If we set  $f(p) = 2p$ , we have the Akaike information criteria (AIC).
- This AIC tends to give overestimated values of  $p$ .
- If we fix  $f(p) = \log(n^*)p$  we have the Bayesian information criteria (BIC).
- The BIC tends to give smaller values of  $p$  in comparison to AIC.



## Forecasting with AR models

- We will consider forecasting from both Bayesian and non-Bayesian perspectives.
- We wish to produce inference about the “future”.
- From time  $n$ , we wish to produce a statement about  $X_{n+1}, X_{n+2}, \dots, X_{n+h}$  where  $h$  is the forecasting horizon (how far we wish to predict in time).
- In a Bayesian setup, this translates into considering the *Predictive distribution* for the future values,

$$p(x_{n+h}, x_{n+h-1}, \dots, x_{n+1} | x_n, \dots, x_1) = \int \int p(x_{n+h}, x_{n+h-1}, \dots, x_{n+1} | x_n, \dots, x_1, \phi, \sigma^2) p(\phi, \sigma^2) d\phi d\sigma^2$$

- For AR models even with the non-informative prior

$p(\phi, \sigma^2) \propto 1/\sigma^2$ , this distribution does not have a recognizable form.

- However, using posterior simulation it is relatively simple to obtain samples of values for  $X_{n+1}, X_{n+2}, \dots, X_{n+h}$
- We can proceed in the following way:
  - Draw a pair  $(\phi, \sigma^2)$  from the Normal-Inverse Gamma distribution as we discussed before.
  - Using this pair, draw a value  $x_{n+1}$  from a Normal distribution with mean  $\sum_{j=1}^p \phi_j x_{n+1-j}$  and variance  $\sigma^2$ .
  - Draw  $x_{n+2}$  from a Normal distribution with mean  $\sum_{j=1}^p \phi_j x_{n+2-j}$  and variance  $\sigma^2$ . (In one of the terms of the autoregression we are using the draw for  $x_{n+1}$ ).

- Continue in this way until we generate a value for  $x_{n+h}$  from a Normal with mean  $\sum_{j=1}^p \phi_j x_{n+h-j}$  and variance  $\sigma^2$
- Repeat all the steps until we obtain  $M$  samples of values  $x_{n+1}, x_{n+2}, \dots, x_{n+h}$
- An approximation to this scheme is to make draws from a predictive distribution which is conditional to an estimate of the model parameters  $(\hat{\phi}, \hat{\sigma}^2)$

$$p(x_{n+h}, x_{n+h-1}, \dots, x_{n+1} | \hat{\phi}, \hat{\sigma}^2, x_n, x_{n-1}, \dots, x_2, x_1).$$

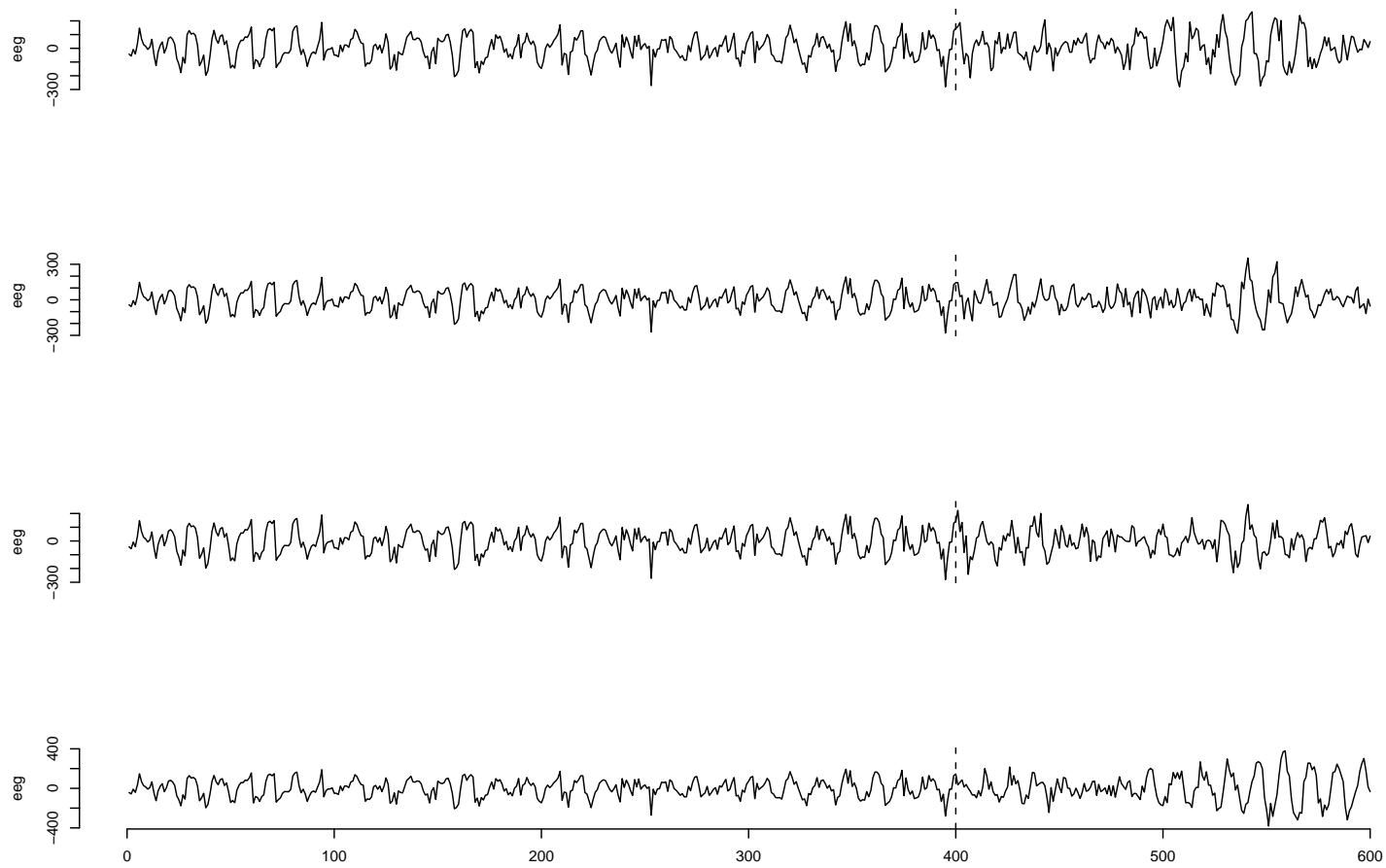
- We are treating  $(\hat{\phi}, \hat{\sigma}^2)$  as the “true” parameter.
- If the sample size  $n$  is large this should produce similar results with respect to full Bayesian approach that uses



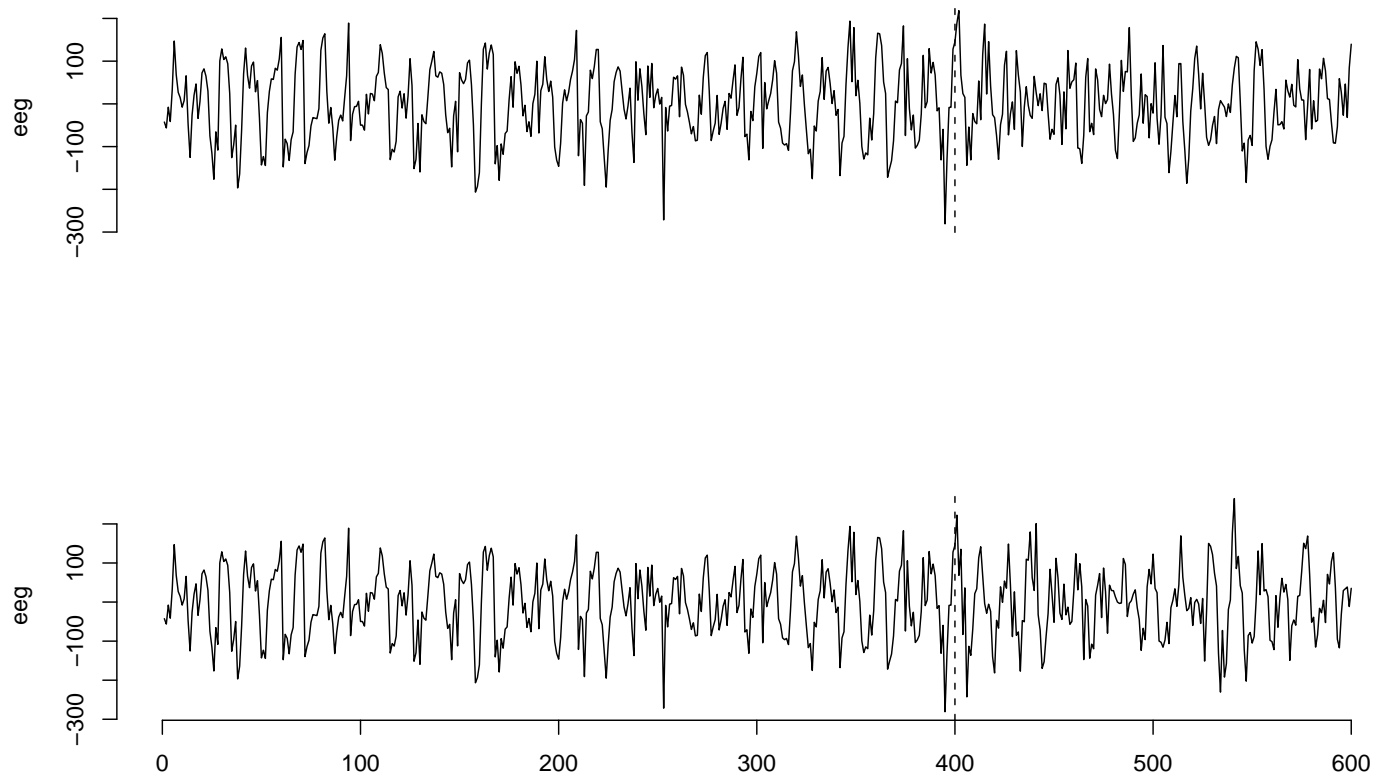
draws of  $(\phi, \sigma^2)$ .

- However, if the sample size is small we could find differences between the distributions.
- Once again, consider the EEG data with an AR(10) model.
- The figures show:
  - Samples of predictive values and data.
  - Comparison of 'full predictive' with 'MLE predictive'
  - Posterior mean of forecasts.
  - Posterior mean and 95% predictive forecasts.
  - Parts of code included in file code6.s

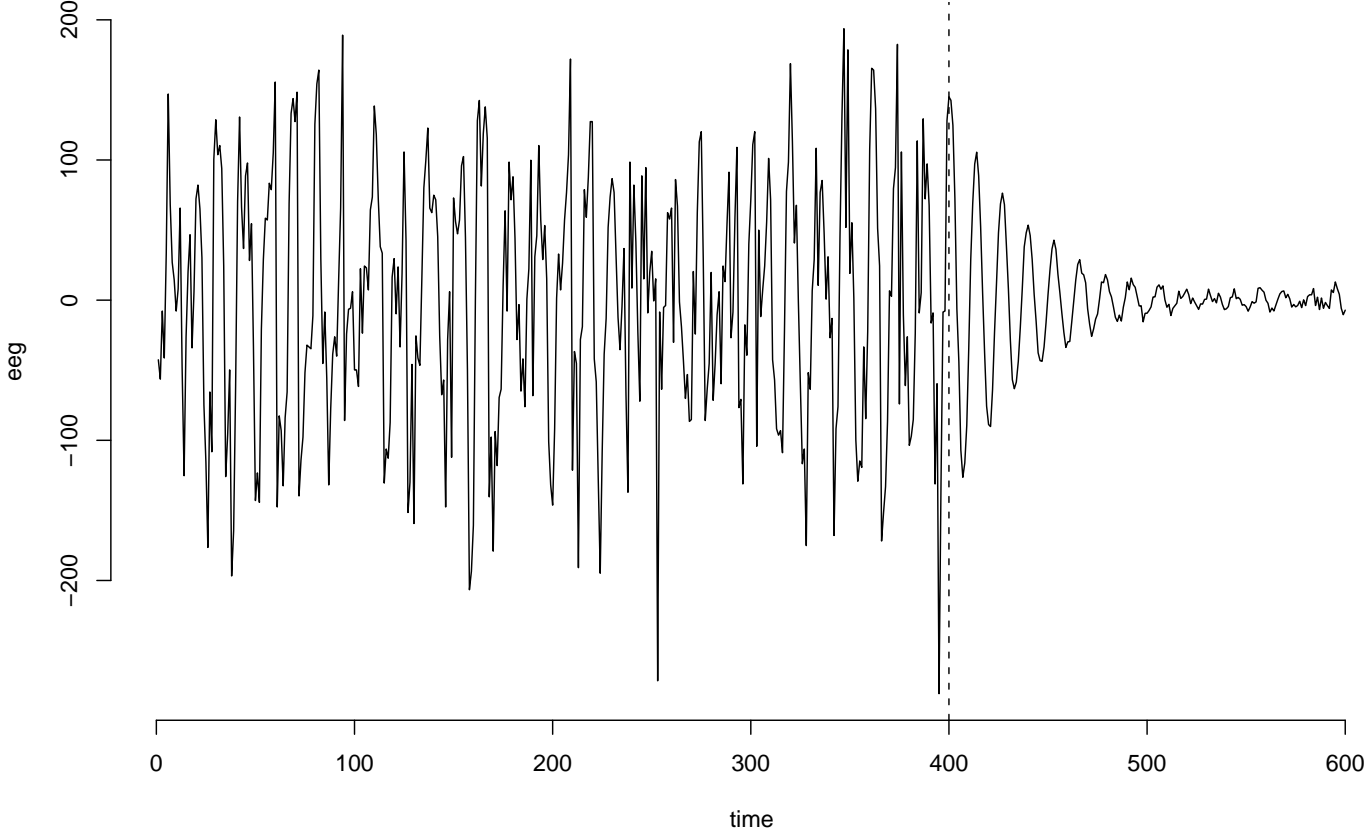
EEG data and 4 samples from the predictive distribution

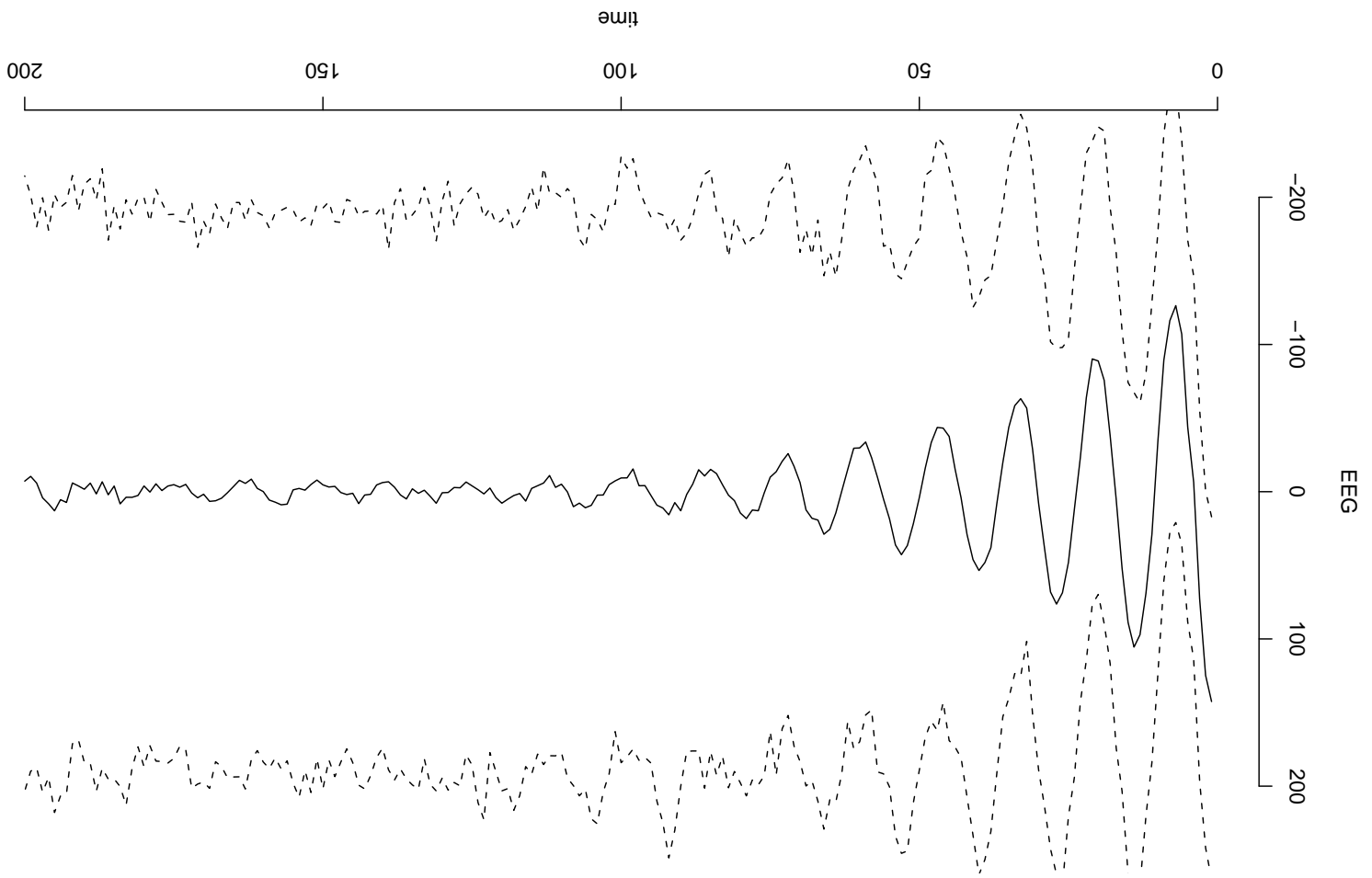


Sample at MLE compared to sample from predictive



EEG data and posterior mean for forecasts





```

# function to produce forecasts
# ph are the model coefficients,
#h is the forecasting horizon
# zt last p values of time series
forcar=function(ph,v,h,zt)
{
x=rep(NA,h);p=length(zt)
for(i in 1:h)
{
  x[i]=sum(ph*zt)+sqrt(v)*rnorm(1)
  zt[2:p]=zt[1:(p-1)]
  zt[1]=x[i]
}
return(x)

```

```
}  
p=10  
zt=rev(eeg[(n-p+1):n])  
forcar(phsim[10,],sigma2[10],200,zt)  
forcar(phhat,s,200,zt)  
# 500 samples and mean  
fr=matrix(NA,200,500)  
for(i in 1:500){  
fr[,i]=forcar(phsim[i,],sigma2[i],200,zt)  
}  
meanfor=apply(fr,1,mean)
```