Estimation and forecasting for time series models

Eric Ward

FISH 507 – Applied Time Series Analysis

12 January 2017

Topics Week 2

- Summarizing ARIMA models
- Maximum Likelihood and Bayesian Estimation
- Prediction & forecasting
- Evaluating forecasts

Review: ARMA models

- A time series is *autoregressive moving average*, or ARMA(*p*,*q*), if it is stationary and

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}$$

- <u>AR processes</u> in biology generally arise from lagged impacts, e.g. the effect of population size on population growth rates

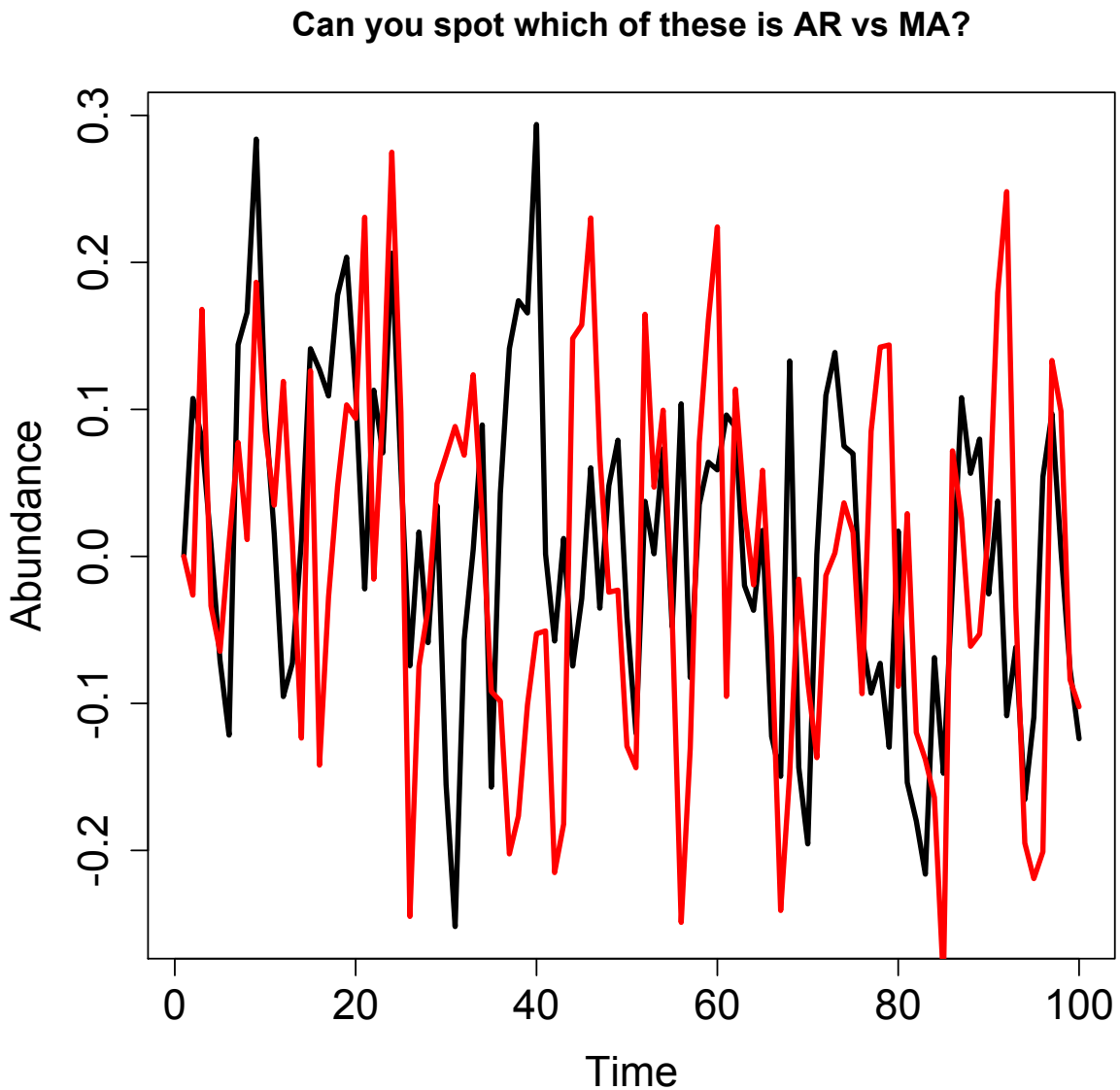
- <u>MA processes</u> describe how random 'shocks' or differences between predictions and observations propagate through time – including unknown external drivers, species interactions, etc

Biological time series are relatively short

- We should only be using lower-order ARMA models with < 40 data points (Ives et al. 2010)

- <u>Examples</u>:
- ARMA(1,1): tree rings, Woollon & Norton 2003
- ARMA (1,1): Vucetich et al. 1997 Cons Bio (moose in ISRO)
- ARMA(2,2): fire dynamics, Beckage & Platt 2003
- ARMA(2,1): Norwegian cod, Stinseth et al. 1999

Visualizing AR & MA processes

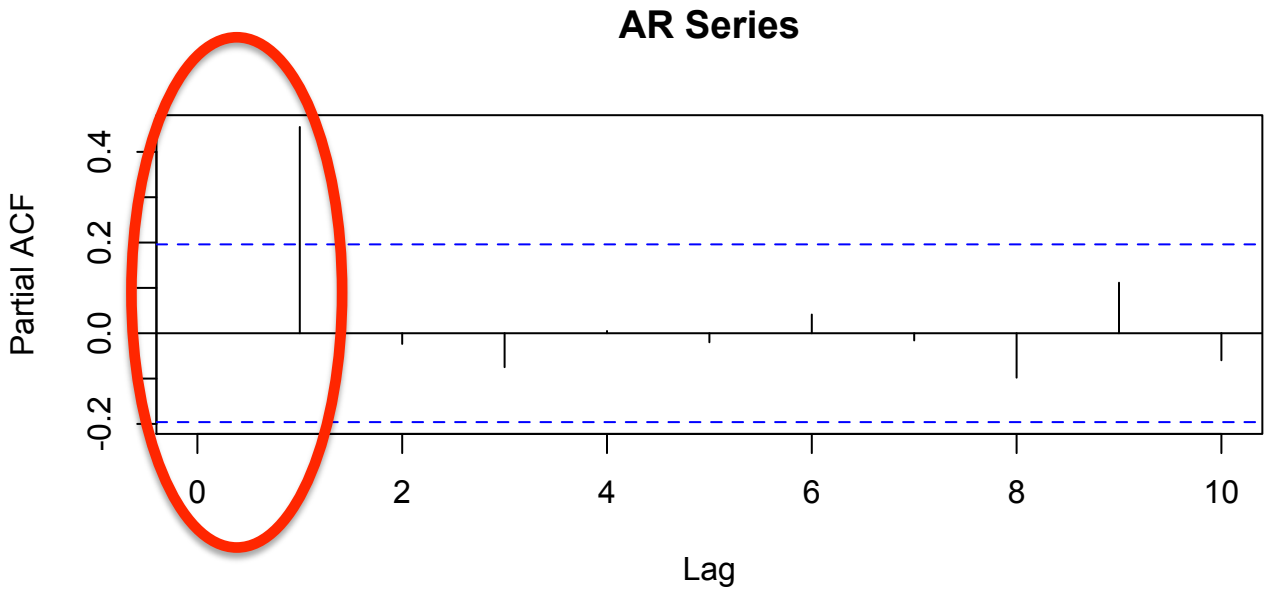


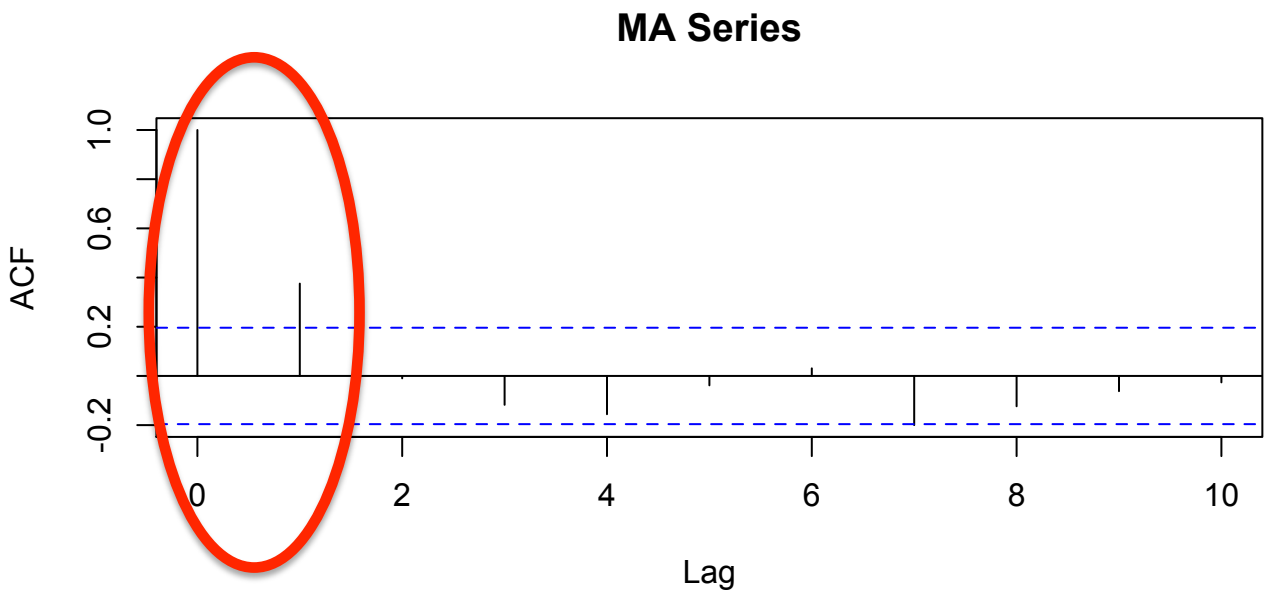
What's the order of the AR() and MA() processes?

Remember Mark's lecture: use ACF & PACF for model ID

	ACF	PACF
AR(p)	Tails off	Cuts off after lag-p
MA(q)	Cuts off after lag-q	Tails off
ARMA(p,q)	Tails off (after lag [q-p])	Tails off (after lag [p-q])

This is an AR(1) and MA(1) model





Topics Week 2

- Summarizing ARIMA models
- Maximum Likelihood and Bayesian Estimation
- Prediction & forecasting
- Evaluating forecasts

Maximum Likelihood

- A priori, we specify that the data (or some process) has been generated from a distribution (e.g. Normal)
- The likelihood of data x_1, x_2, x_3, can then be calculated

$$L(\theta \mid x_1, x_2, x_3) = L(\theta \mid x_1) \times L(\theta \mid x_2) \times L(\theta \mid x_3)$$

$$\theta = (u, \sigma)$$

- R provides built in functions for doing this: `dnorm()`, `dpois()`, `dbinom()`, `dgamma()`, `etc.`

Interpretation of maximum likelihood

- When we write the likelihood, the parameters are conditioned on the data.
- "What's the likelihood of the parameters given the data?"
- **Parameters are fixed quantities, data are random**

$$L(\theta \mid x_1, x_2, x_3) = L(\theta \mid x_1) \times L(\theta \mid x_2) \times L(\theta \mid x_3)$$

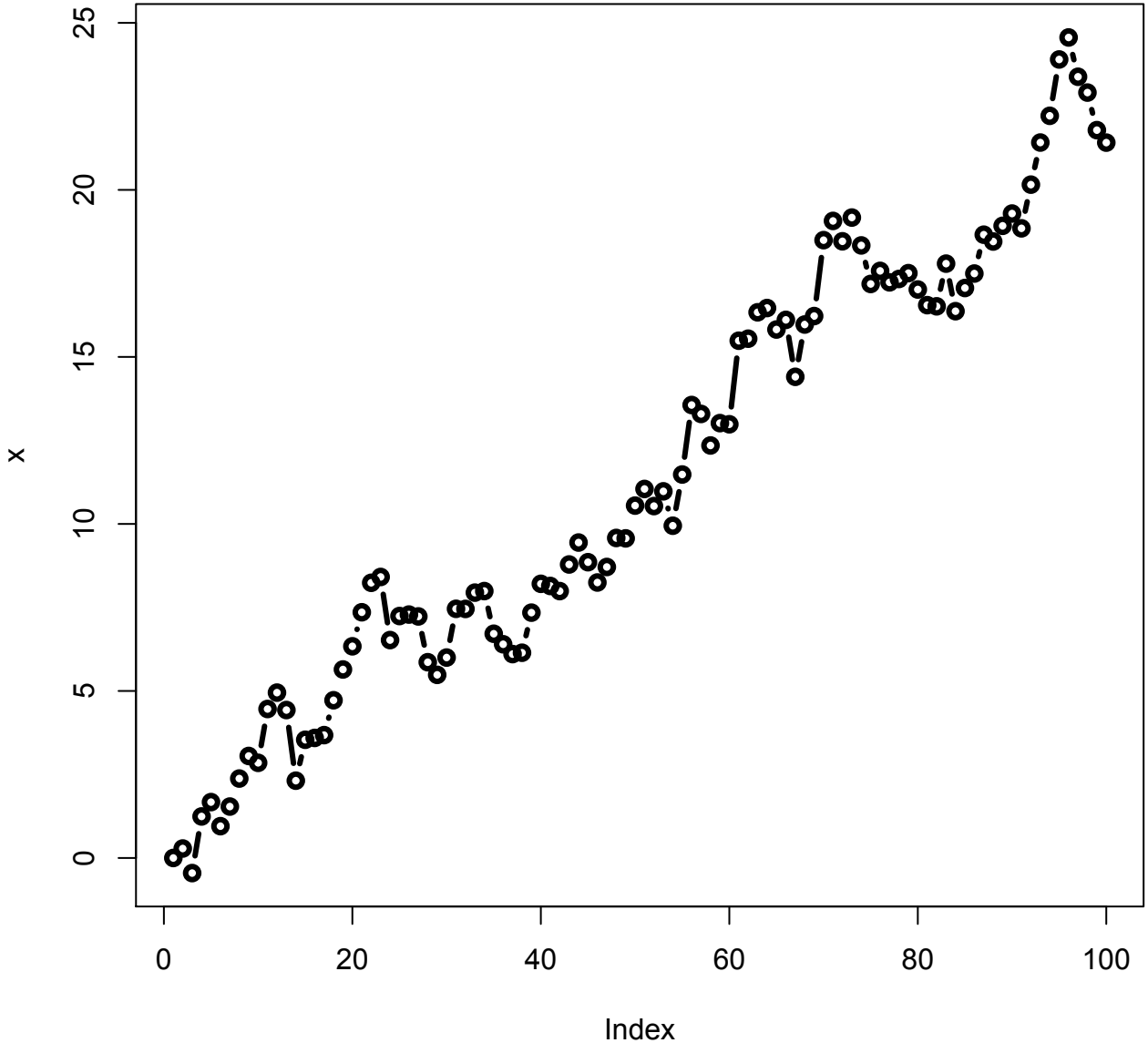
$$\theta = (u, \sigma)$$

Goal: find parameters that maximize likelihood

- Parameters that maximize the likelihood will also maximize the log-likelihood
- Equivalently, we can minimize the negative log-likelihood
- Example: generate random walk with drift
  - Stochastic model aka "process error model"

```
set.seed(1)
e = rnorm(100,0,1) # white noise ~ Normal(0,1)
x = 0 # initial value
for(i in 2:100) {x[i] = x[i-1] + 0.1 + e[i]}
```

The Data



Steps to find MLE

- Write a function that takes in parameters, and returns NLL (= Negative log likelihood)

```
rwdrift = function(pars) {
        # pars[1] = drift, pars[2] = error
        drift = pars[1]
        sigma = exp(pars[2]) # trick to keep positive
        predx= 0
        for(i in 2:100) {
                predx[i] = x[i-1] + drift
        }
        logLike = sum(dnorm(x[2:100], predx[2:100], sd = sigma,
log=TRUE))
        return(-logLike) # return NLL because optim minimizes
}
```

Use your favorite minimizer/ maximizer

```
> optim(runif(2), rwdrift)
$par
[1] 0.2163119 -0.1108552

$value
[1] 129.5063

$counts
function gradient
      59       NA

$convergence
[1] 0

$message
NULL
```

- Even for 100 data points, estimates of drift and error variance aren't perfect

optim() tells us that the algorithm has converged at the MLE

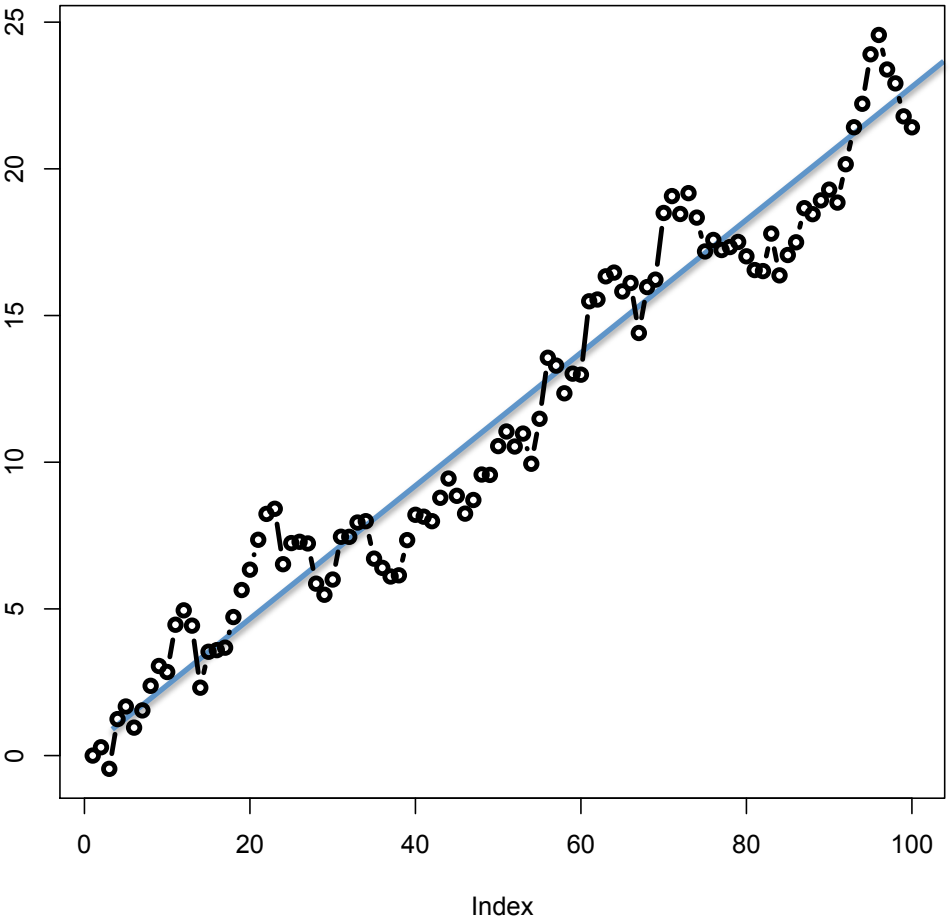
Other functions in R

- Many existing functions we're using – lm(), arima(), Arima(), MARSS(), are also using maximum likelihood
- rwf() in 'forecast' does the exact same thing as our function 'rwdrift'

```
> summary(rwf(x,drift=TRUE))

Forecast method: Random walk with drift

Model Information:
$drift
[1] 0.2163151

$drift.se
[1] 0.09042107

$sd
[1] 0.8996783

$call
rwf(x = x, drift = TRUE)
```

- A second type of model we could fit would be fitting a regression line through the data?

$$x_t = \alpha_0 + \alpha_1 u_t + z_t;\ z_t \sim Normal(0,\sigma)$$




```
> summary(lm(x~seq(1,100)))

Call:
lm(formula = x ~ seq(1, 100))

Residuals:
    Min      1Q  Median      3Q     Max
-2.5603 -1.0185 -0.2884  1.0472  3.1156

Coefficients:
            Estimate Std. Error t value Pr(>|t
(Intercept) 0.355070   0.291620   1.218    0.2
seq(1, 100) 0.219724   0.005013  43.827   <2e-
---
```

Tiny bit different from the trend estimate we got from rwf() –

1. Is this difference meaningful?
2. Any insight as to why they're different?
3. What does this imply for predictions?

- Regression is fitting a deterministic process through the data (all residual error = “observation error”)
- Random walks are fitting a stochastic process (no observation error, all process variability)
- In the lab, we’ll also introduce univariate ‘state-space’ models, which estimate both process and observation error variances

Bayesian Estimation

- Subtle but important differences between maximum likelihood / Bayesian approaches

- Bayesians also use likelihood, but view the data as fixed and the parameters as random

$$L(x \mid \theta) \quad vs \quad L(\theta \mid x)$$

Bayesian

Maximum likelihood

Bayes Theorem

- Based on laws of conditional probability

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- Using our previous notation & likelihood,

$$P(\theta \mid x) = \frac{P(x \mid \theta)P(\theta)}{P(x)}$$

- P(x) is a constant, and usually not written, so

$$P(\theta \mid x) = P(x \mid \theta)P(\theta)$$

Bayes Theorem

- What are the components of this equation?

$$P(\theta \mid x) = P(x \mid \theta) P(\theta)$$

$P(x \mid \theta)$ is the likelihood

$P(\theta)$ is the prior probability distribution

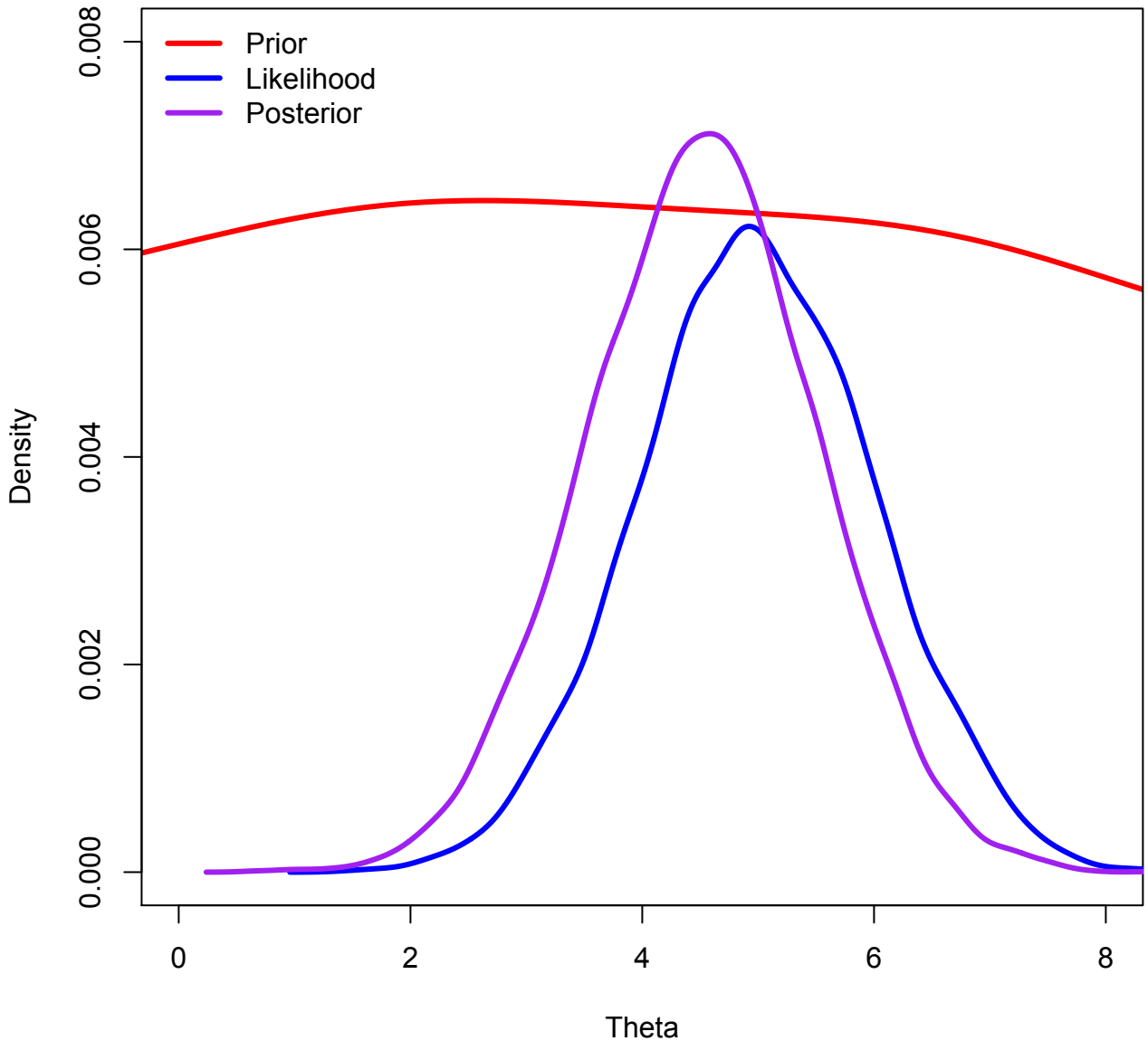
$P(\theta \mid x)$ is the posterior probability distribution

Why use a prior

- Prior necessary to express the posterior as a probability distribution
- It's also expression of a priori belief

- Posterior is thus a probability distribution
  - Difference between posterior and prior is a measure of how much you 'learn' by seeing data
  - Can also be thought of as weighted average of data + beliefs

Example


Prior
Likelihood
Posterior
Density
0.000
0.002
0.004
0.006
0.008
0
2
4
6
8
Theta

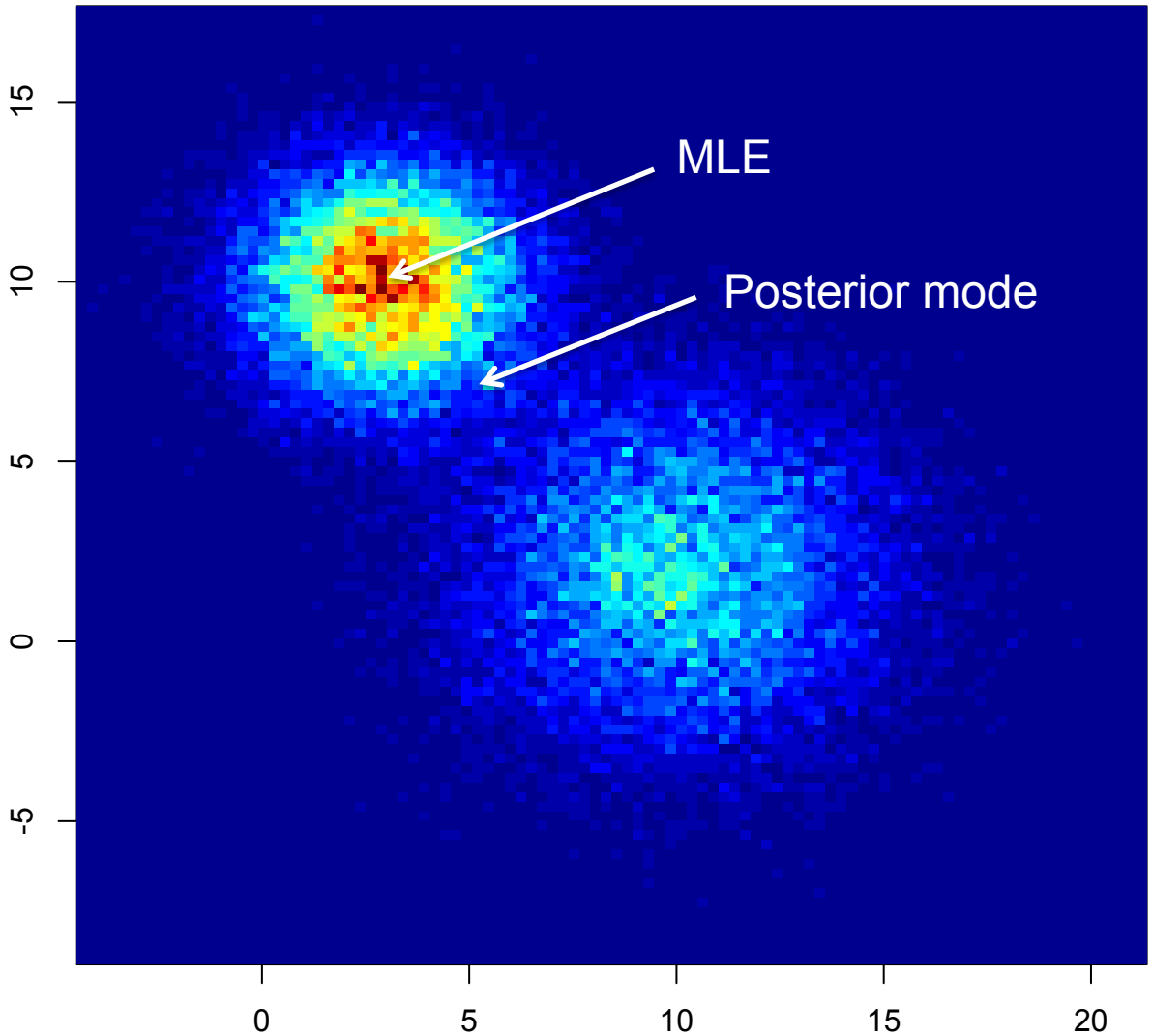
Differences in estimation

- Bayesian methods require evaluating the likelihood by integrating over the parameter space (instead of maximizing)

- Fish 507: there are a handful of ways to do numerical integration. For this class, we'll only use Markov Chain Monte Carlo (MCMC)
  - Sample sequentially 1000s of samples of parameter space

2D posterior surface: think about animal foraging on landscape

- MLE is finding the absolute best point.
- Bayesian methods attempt to find the best point on average.



Overview of MCMC

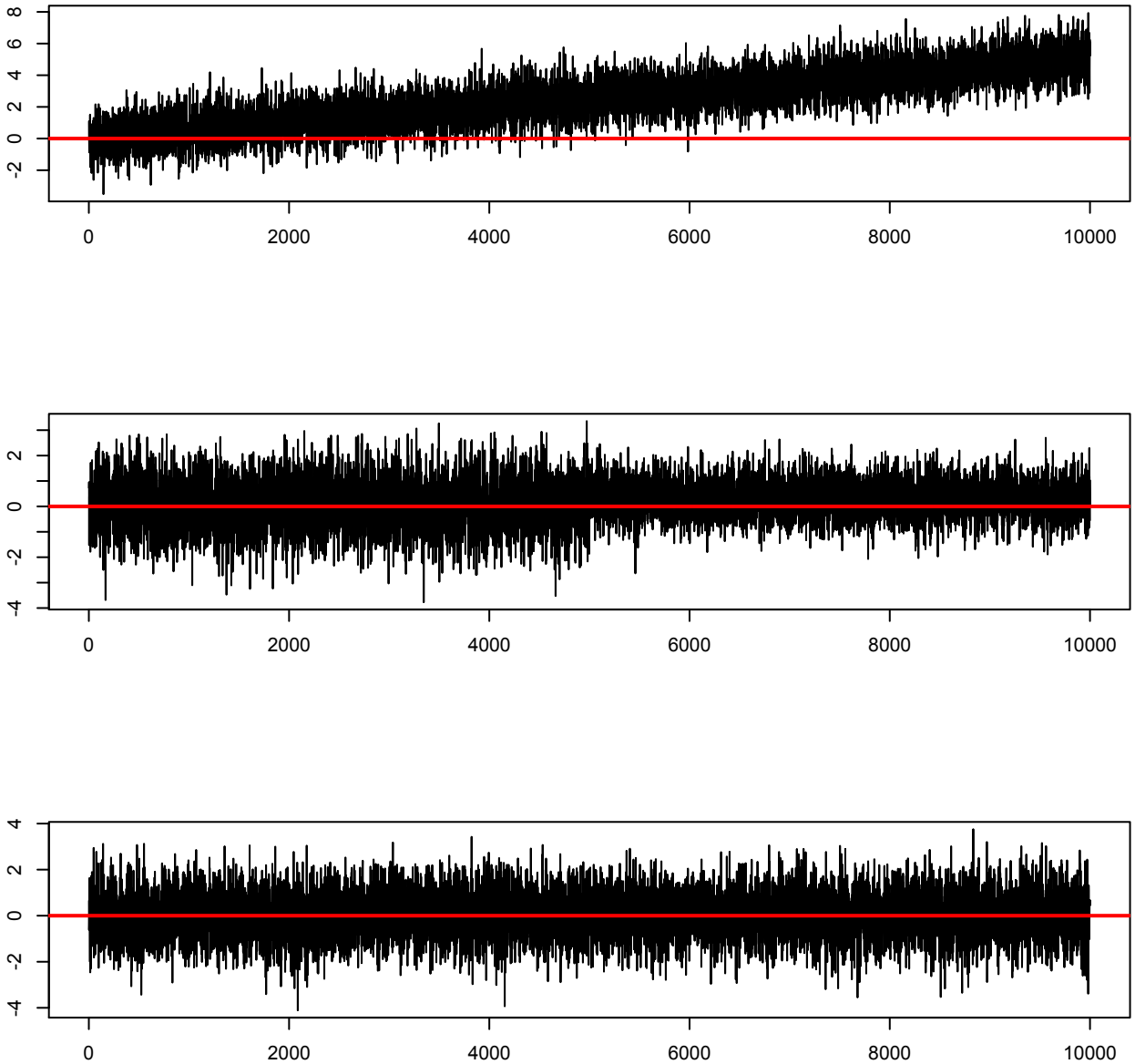
- Simulate random walks over the parameter space, with a tendency to spend more time in areas of high likelihood
- Each random walk = MCMC chain.
  - Each initialized from unique starting point
  - Each samples independently for 1000s of iterations
    - We'll discard the first XX samples, as a "burn-in period"

MCMC convergence

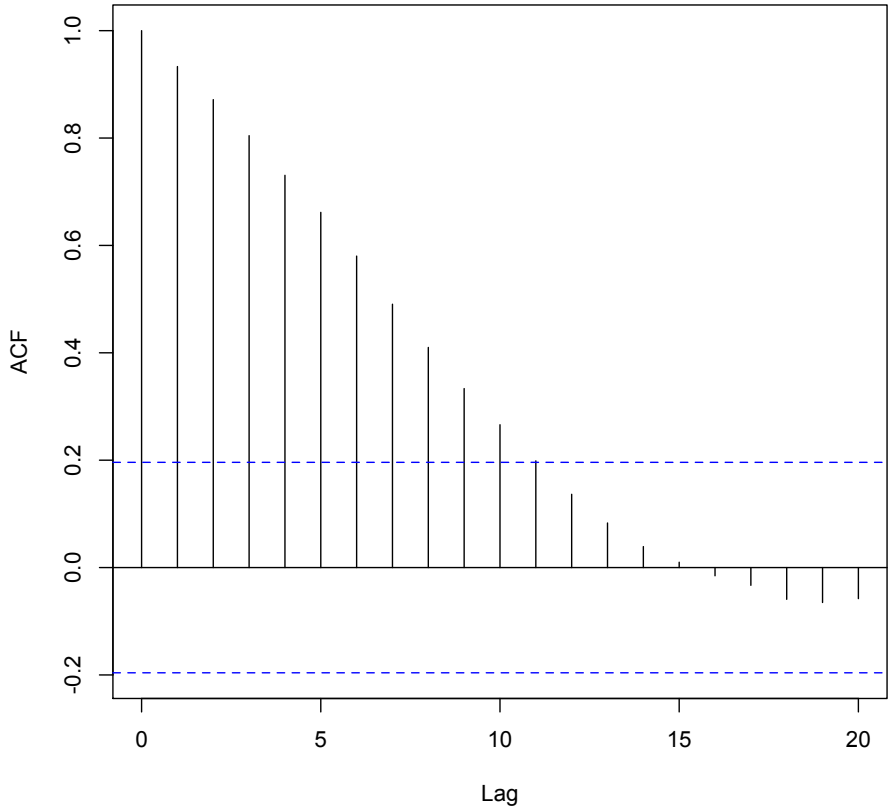
- Assessing convergence is more difficult than maximum likelihood

- We'll run through a couple diagnostics, but this is not comprehensize (Andre's 507 dives into this more)

We need chains to be stationary



We want each sample to be approximately independent



- If lag(1) acf is too high, increase the ‘thinning rate’ of the MCMC chain

Quantitative tests of convergence

- Gelman-Rubin diagnostic
  - Used to check convergence of multiple chains in parallel. Goal: R_hat in (1.0, 1.05)
- Geweke diagnostic
  - Is mean of first 10% of MCMC chain the same as the last 50%?
- Heidelberger-Welch diagnostic
  - Is the entire chain stationary? If not, is the last 90%? 80%? 70%? Etc.

Topics Week 2

- Summarizing ARIMA models
- Maximum Likelihood and Bayesian Estimation
- Prediction & forecasting
- Evaluating forecasts

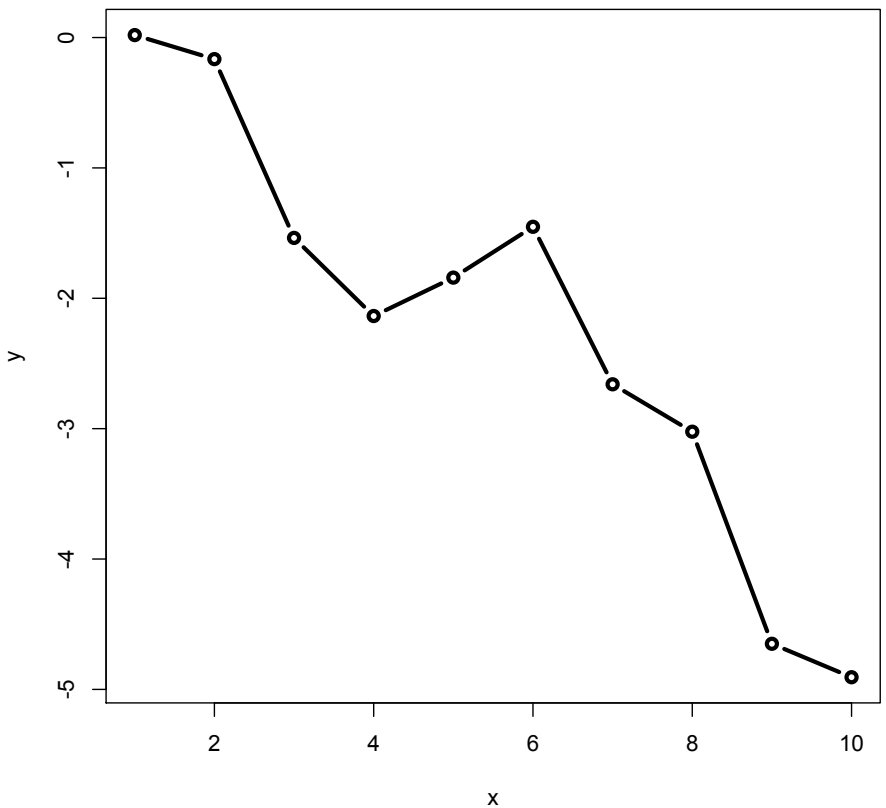
Forecasting

- We'll use the word forecasting to refer to out of sample prediction
- Consider the following 2 basic models
  - Linear regression
  - Random walk
- What can you say about their predictions – or where their errors are coming from?
- Linear regression errors = observation, random walk errors = process

predict() function

- After fitting a regression model, we can use predict() or predict.lm()

- Example



```
> # create a hypothetical dataset
> set.seed(10)
> x = seq(1,10)
> y = cumsum(rnorm(10, 0, 1))
>
> # do basic linear regression
> mod = lm(y ~ x)
> summary(mod)

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66743 -0.52735  0.04611  0.32685  1.03915

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.57589    0.40756   1.413    0.195
x           -0.51112    0.06568  -7.781 5.33e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5966 on 8 degrees of freedom
Multiple R-squared: 0.8833, Adjusted R-squared: 0.8687
F-statistic: 60.55 on 1 and 8 DF,  p-value: 5.328e-05
```

- If we apply predict() with default arguments, our prediction se is smaller than the residual error!

```
> predict(mod, newdata=list(x=11), se.fit=T)
$fit
        1
-5.046396

$se.fit
[1] 0.4075607

$df
[1] 8

$residual.scale
[1] 0.5966078
```

Confidence v Prediction Intervals

- Confidence intervals on the mean (in-sample)

```
predict(mod,
newdata=list(x=11),
se.fit=T,
interval="confidence")
```

- Confidence intervals on the mean (in-sample)

- Interval = (-5.99, -4.10)

- Prediction intervals should be used for new observations (in or out of sample)

```
predict(mod,
newdata=list(x=11),
se.fit=T,
interval=”prediction”)
```

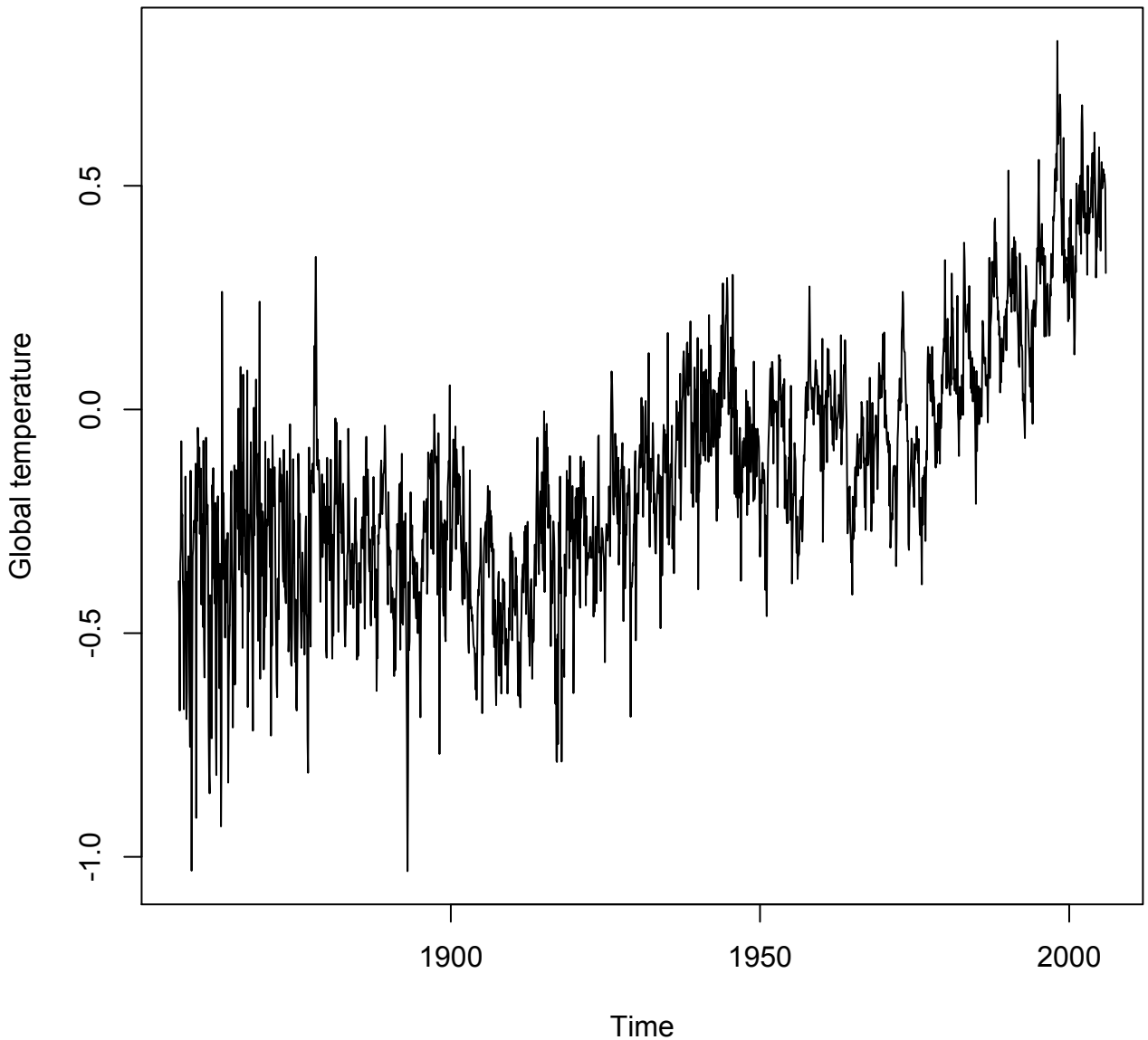
- Interval = (-6.71, -3.38)

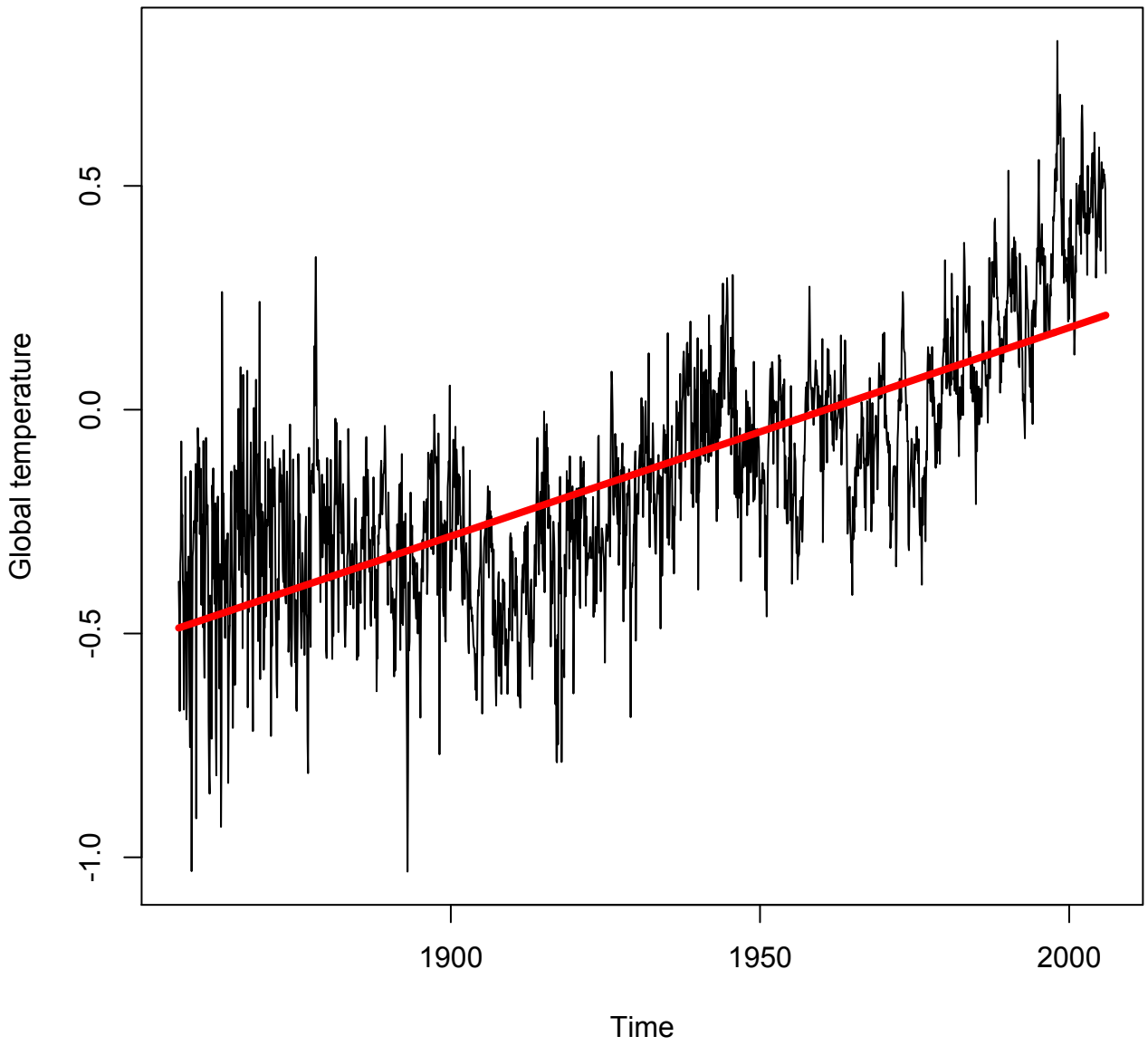
Using gam() as forecasting model

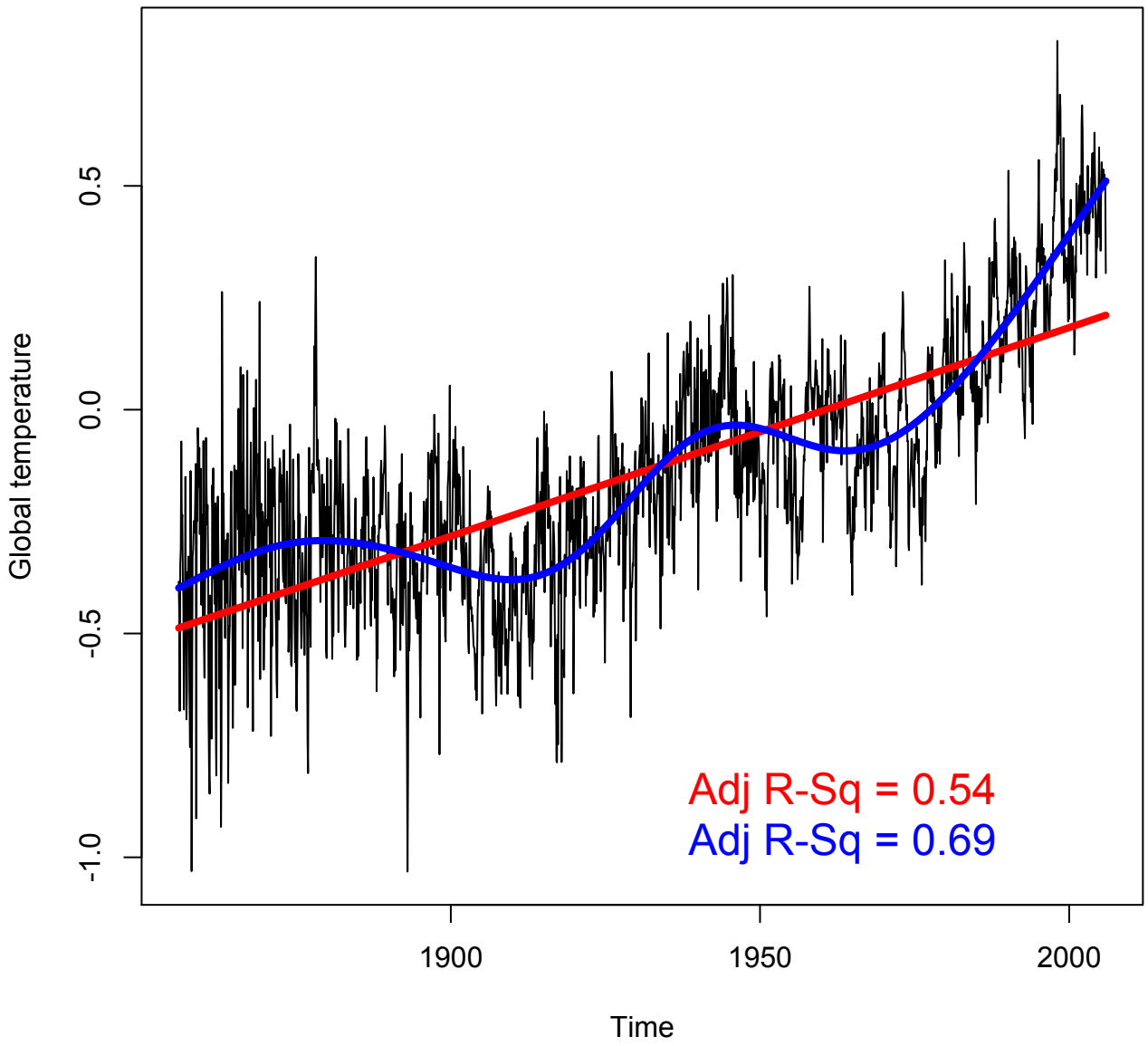
- gam() is in “mgcv”

- More flexible than OLS regression
  - Non-linear
  - Non normal errors

- Complexity can be captured via fitting a series of polynomial splines


Global temperature
0.5
0.0
-0.5
-1.0
1900
1950
2000
Time


Global temperature
0.5
0.0
-0.5
-1.0
1900
1950
2000
Time


Global temperature
0.5
0.0
-0.5
-1.0
1900
1950
2000
Time
Adj R-Sq = 0.54
Adj R-Sq = 0.69

Predictions from GAMs

```
predict(gam.mod,newdata=list(time=2006),type
="terms",se=TRUE)
```

- type = “terms” only includes uncertainty in spline terms
- type = “iterms” or “response” will make se.fit larger because it includes uncertainty in intercept

Forecasting with arima()

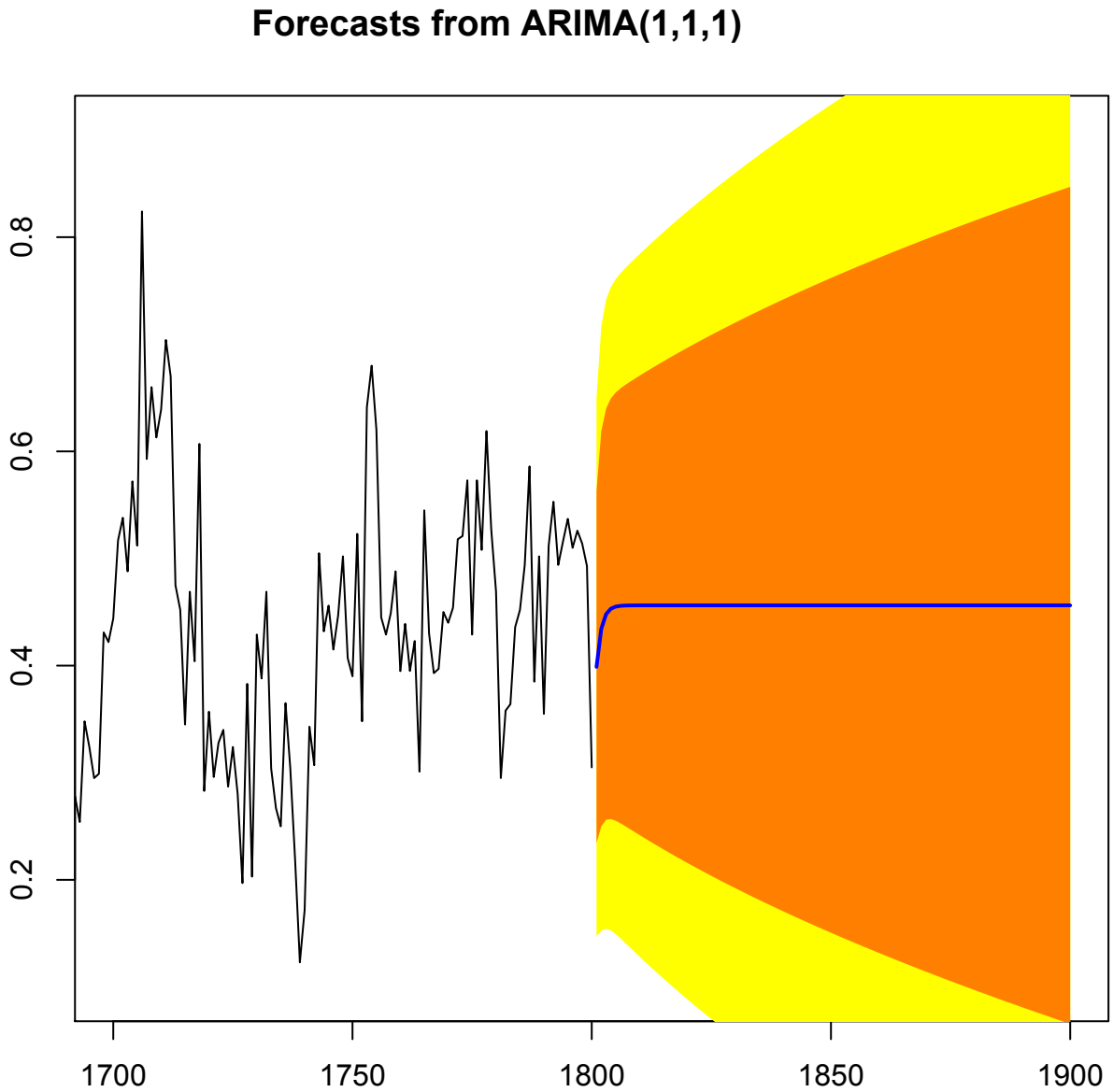
- Let's fit an ARMA(1,1) model to the global temperature data, after 1st differencing to remove trend
- You can use the arima() function or Arima() function – Arima() is a wrapper for arima()

```
# for simplicity, we won't include a separate ARMA model for seasonality
ar.global.1 = Arima(Global, order =
c(1,1,1),seasonal=list(order=c(0,0,0),period=12))

f1 = forecast(ar.global.1, h = 10)
```

What does f1 contain?

```
Model Information:
Series: Global
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.3797  -0.8700
s.e.  0.0433   0.0293

sigma^2 estimated as 0.01644:  log likelihood=1142.13
AIC=-2278.26   AICc=-2278.25   BIC=-2261.77

In-sample error measures:
           ME         RMSE          MAE          MPE         MAPE
2.270029e-03 1.281826e-01 9.390127e-02 1.313987e+01 1.076644e+02

Forecasts:
     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
1801      0.3988276 0.2345093 0.5631459 0.1475244 0.6501308
1802      0.4344510 0.2500229 0.6188790 0.1523926 0.7165093
1803      0.4479760 0.2560411 0.6399109 0.1544369 0.7415151
1804      0.4531111 0.2567326 0.6494896 0.1527761 0.7534460
1805      0.4550607 0.2552030 0.6549184 0.1494047 0.7607167
1806      0.4558009 0.2528226 0.6587791 0.1453725 0.7662293
1807      0.4560819 0.2501388 0.6620251 0.1411191 0.7710448
1808      0.4561886 0.2473628 0.6650145 0.1368171 0.7755602
1809      0.4562291 0.2445748 0.6678835 0.1325318 0.7799265
1810      0.4562445 0.2418046 0.6706844 0.1282870 0.7842021
> |
```

plot fitted arima() object



Summary

- As expected, uncertainty increases as a function of forecast length
- We could also perform forecasts by bootstrapping new values of arima() parameters

- Bayesian forecasts are very similar
  - Forecasts can be made based on the mode, HPD region, or for all MCMC samples

With lab

- Practice using arima.sim () to simulate time series of different AR and MA orders