	MA0	MA1	MA2	MA3	MA4	MA5
AR0	166.78	46.98	7.71	-13.70	-17.62	-24.89
AR1	-37.25	-36.62	-34.74	-33.13	-33.14	-31.18
AR2	-36.52	-37.41	-35.89	-33.89	-33.24	-31.91
AR3	-34.79	-34.43	-32.44	-31.89	-32.05	-32.14
AR4	-33.19	-33.91	-33.48	-33.54	-30.15	-29.52

Table Corrected

inconsistent

inconsistent

Question 5.3. What do we learn by interpreting the results in the above table of AIC values?

Question 5.4. In what ways might we have to be careful not to over-interpret the results of this table?

- Let's fit the ARMA(2,1) model recommended by consideration of AIC.

```
huron_arma21 <- arima(huron_level,order=c(2,0,1))
huron_arma21


Call:
arima(x = huron_level, order = c(2, 0, 1))

Coefficients:
          ar1     ar2     ma1  intercept
      -0.0561  0.7935  1.0000   176.4591
s.e.   0.0521  0.0525  0.0257     0.1209

sigma^2 estimated as 0.04217:  log likelihood = 23.71,  aic = -37.41
```

MA root is essentially -1 [it turns out that this makes std errors untrustworthy].

note: R gives a very small standard error here: the MA(1) parameter is supposedly estimated with high accuracy.

- We can examine the roots of the AR polynomial,

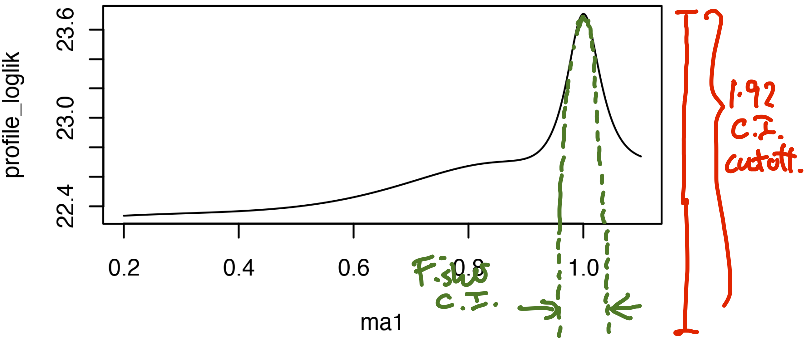
```
AR_roots <- polyroot(c(1,-coef(huron_arma21)[c("ar1","ar2")]))
AR_roots

[1]  1.158532-0i -1.087774+0i
```

- The roots are just outside the unit circle, suggesting we have a stationary causal fitted ARMA.
- However, the MA root is -1, showing that the fitted model is at the threshold of non-invertibility.
- Do we have a non-invertibility problem? We investigate this using profile and bootstrap methods. The claimed standard error on the MA1 coefficient, from the Fisher information approach used by `arima`, is small.

- First, we can see if the approximate confidence interval constructed using profile likelihood is in agreement with the approximate confidence interval constructed using the observed Fisher information.
- To do this, we need to maximize the ARMA likelihood while fixing the MA1 coefficient at a range of values. This is done using `arima` in the code below.
- Note that the `fixed` argument expects a vector of length $p+q+1$ corresponding to a concatenated vector $(\phi_{1:p}, \psi_{1:q}, \mu)$. Somehow, the Gaussian white noise variance, σ^2, is not included in this representation. Parameters with `NA` entries in `fixed` are estimated.

```
K <- 500
ma1 <- seq(from=0.2,to=1.1,length=K)
profile_loglik <- rep(NA,K)
for(k in 1:K){
   profile_loglik[k] <- logLik(arima(huron_level,order=c(2,0,1),
      fixed=c(NA,NA,ma1[k],NA)))
}
plot(profile_loglik~ma1,ty="l")
```



Question 5.5. Interpret the profile likelihood plot for ψ_1.

The profile C.I. is very wide, essentially any MA(1) is consistent with the data.

The observed Fisher information matches the profile cutoff but for a quadratic approximation to the profile at its maximum [this is not obvious, but true]

Question 5.6. What do you conclude about the Fisher information confidence interval proposed by `arima`?

The quadratic approximation used by the Fisher method is not reliable over the range of the C.I., so the method is probably unreliable.

Question 5.7. In what situations is the Fisher information confidence interval reliable? Perhaps when roots are not close to 1? Or not close to canceling?

when the profile likelihood is nicely quadratic over many units of log likelihood - when the model is well behaved and there is plenty of data in a central limit theorem situation, the log likelihood should become quadratic.

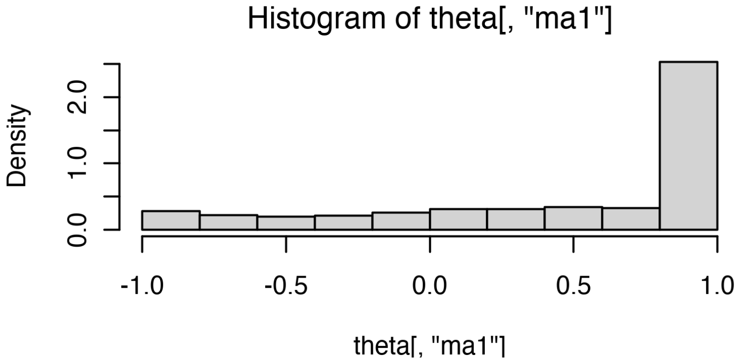
Question 5.8. Is this profile likelihood plot, and its statistical interpretation, reliable? How could you support your opinion on this?

At this point, we are motivated to do a simulation study to find out.

A simulation study

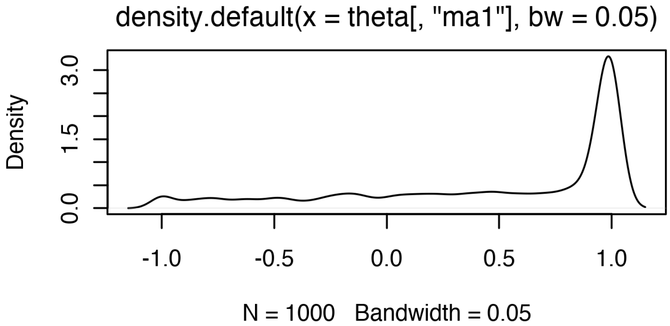
```
set.seed(578922)
J <- 1000
params <- coef(huron_arma21)
ar <- params[grep("^ar",names(params))]
ma <- params[grep("^ma",names(params))]
intercept <- params["intercept"]
sigma <- sqrt(huron_arma21$sigma2)
theta <- matrix(NA,nrow=J,ncol=length(params),
   dimnames=list(NULL,names(params)))
for(j in 1:J){
 try({
   Y_j <- arima.sim(
      list(ar=ar,ma=ma),
      n=length(huron_level),
      sd=sigma
   )+intercept
   theta[j,] <- coef(arima(Y_j,order=c(2,0,1)))
 })
}
```

"try" stops code from crashing when there is an error.



- This seems consistent with the profile likelihood plot.
- A density plot shows this similarity even more clearly.

```
plot(density(theta[,"ma1"],bw=0.05))
```



- Here, we look at the raw plot for instructional purposes. For a report, one should improve the default axis labels and title.
- Note that `arima` transforms the model to invertibility. Thus, the estimated value of θ_1 can only fall in the interval $[-1, 1]$.

```
range(theta[,"ma1"])

[1] -1  1
```

- A minor technical issue: estimated densities outside $[-1,1]$ are artifacts of the density estimation procedure.

Question 5.9. How would you refine this density estimation procedure to respect the range of the parameter estimation procedure?

We could transform the parameter before applying the kernel density estimate & then transform back.

- We do a simulation study for which we fit ARMA(2,1) when the true model is AR(1).

Using multiple cores for simulation studies

- When doing simulation studies, **multicore computing** is helpful. All modern computers have multiple cores.
- A basic approach to multicore statistical computing is to tell R you want it to look for available processors, using the `doParallel` package.
- We can use `foreach` in the `doParallel` package to carry out a parallel `for` loop where jobs are sent to different processors.

```
library(doParallel)
registerDoParallel()
```

slightly more than ½ the class have no prior multi-core coding.

```
J <- 1000
huron_ar1 <- arima(huron_level,order=c(1,0,0))
params <- coef(huron_ar1)
ar <- params[grep("^ar",names(params))]
intercept <- params["intercept"]
sigma <- sqrt(huron_ar1$sigma2)
t1 <- system.time(
  huron_sim <- foreach(j=1:J) %dopar% {
    Y_j <- arima.sim(list(ar=ar),n=length(huron_level),
      sd=sigma)+intercept
    try(coef(arima(Y_j,order=c(2,0,1))))
  }
)
```

note: %do% instead of %dopar% it uses a standard single-core non-parallel for loop.
This is good for debugging: %dopar% should be equal to %do% but quicker!

- Some of these `arima` calls did not successfully produce parameter estimates. The `try` function lets the simulation proceed despite these errors. Let's see how many of them fail:

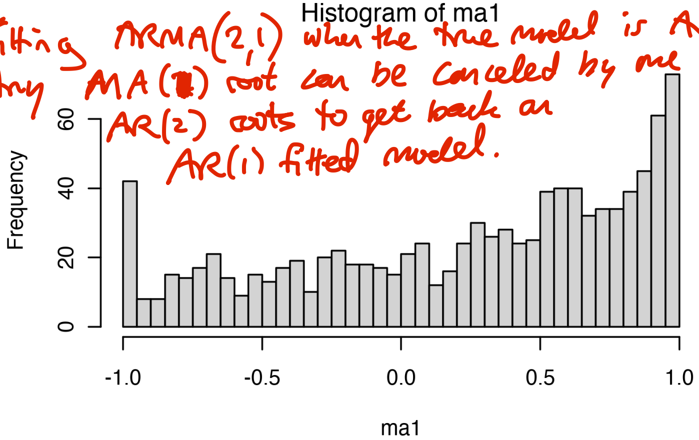
```
sum(sapply(huron_sim, function(x) inherits(x,"try-error")))

[1] 1
```

- Now, for the remaining ones, we can look at the resulting estimates of the MA1 component:

```
ma1 <- unlist(lapply(huron_sim,function(x)
   if(!inherits(x,"try-error"))x["ma1"] else NULL ))
hist(ma1,breaks=50)
```

Fitting ARMA(2,1) when the true model is AR(1). Any MA(1) root can be canceled by one of the AR(2) roots to get back an AR(1) fitted model.



- When the true model is AR1 and we fit ARMA(2,1), it seems that we often obtain a model with estimated MA1 coefficient on the boundary of invertibility.
- Thus, we cannot reject an AR1 hypothesis for the Huron data, even though the Fisher information based analysis appears to give strong evidence that the data should be modeled with a nonzero MA1 coefficient.
- It may be sensible to avoid fitted models too close to the boundary of invertibility. This is a reason not to blindly accept whatever model AIC might suggest.

Question 5.10. What else could we look for to help diagnose, and understand, this kind of model fitting problem? Hint: pay some more attention to the roots of the fitted ARMA(2,1) model.

AR roots: 1.16 and -1.09

MA root: -1.000

The AR & MA roots almost cancel

Assessing the numerical correctness of evaluation and maximization of the likelihood function

- We can probably suppose that `arima()` has negligible numerical error in evaluating the likelihood.
- Likelihood evaluation is a linear algebra computation which should be numerically stable away from singularities.
- Possibly, numerical problems could arise for models very close to reducibility (canceling AR and MA roots).
- Numerical optimization is more problematic.
- `arima` calls the general purpose optimization routine `optim`.
- The likelihood surface can be multimodal and have nonlinear ridges, when AR and MA roots almost cancel. unlike regression, generalized linear models,
- No optimization procedure is reliable for maximizing awkward, non-convex functions.
- Evidence for imperfect maximization (assuming negligible likelihood evaluation error) can be found in the AIC table, copied below.

	MA0	MA1	MA2	MA3	MA4	MA5
AR0	166.8	47.0	7.7	-13.7	-17.6	-24.9
AR1	-37.2	-36.6	-34.7	-33.1	-33.1	-31.2
AR2	-36.5	-37.4	-35.9	-33.9	-33.2	-31.9
AR3	-34.8	-34.4	-32.4	-31.9	-32.0	-32.1
AR4	-33.2	-33.9	-33.5	-33.5	-30.1	-29.5

Question 5.11. How is this table inconsistent with perfect maximization?

covered earlier.

- Hint: recall that, for nested hypotheses $H^{\langle 0\rangle} \subset H^{\langle 1\rangle}$, the likelihood maximized over $H^{\langle 1\rangle}$ cannot be less than the likelihood maximized over $H^{\langle 0\rangle}$.
- Recall also the definition of AIC,
  AIC = -2$\times$ maximized log likelihood + 2$\times$ number of parameters

Further reading

- Section 3.5 of Shumway and Stoffer (2017) gives a complementary discussion of parameter estimation for ARMA models.
- Section 3.7 of Shumway and Stoffer (2017) takes a different perspective on selecting ARMA models, putting less emphasis on likelihood. Both perspectives can be valuable.

References and Acknowledgements

Shumway RH, Stoffer DS (2017). *Time Series Analysis and its Applications: With R Examples*. Springer.

- Compiled on January 27, 2022 using R version 4.1.2.

- We acknowledge previous versions of this course.