AR(1)

1. The AR(1) model is $y_t = \phi y_{t-1} + \varepsilon_t$, where ϕ is a constant coefficient and ε is white noise with standard deviation σ. You can generate an AR(1) series with this code:

```
N <- 500
y <- rep(0,N)
sigma <- 1
phi <- 0.5
ep <- rnorm(N,0,sigma)
for (t in 2:N) { y[t] <- phi*y[t-1] + ep[t] }
```

   Generate series with various choices of ϕ. For each, plot the series and examine its autocorrelogram. Summarize what you've learned about how AR(1) series depend on ϕ.

2. Pick a reasonable value of ϕ and investigate how σ affects the AR(1) series.

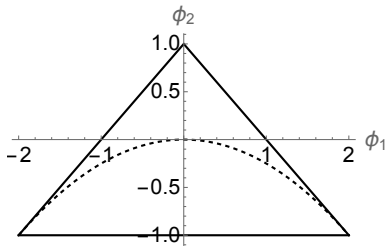
3. The built-in function `ar` fits an AR model to data. How well does `ar(y, order.max=1)` recover ϕ and σ for your simulated series?

4. You can also fit AR models with the built-in linear modeling command `lm`, where the explanatory variable is the lagged series. That is, `lm(y ~ dplyr::lag(y))`. Check that this also recovers ϕ and σ for your simulated series.

AR(2)

5. The AR(2) model is $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t$, where ϕ_1, ϕ_2 are constant coefficients and ε is white noise with standard deviation σ. Adapt your code from AR(1) to generate AR(2) models.

   Generate series with various choices of ϕ_1, ϕ_2. For each, plot the series and examine its autocorrelogram. This picture shows interesting regions to explore:



6. The base R command `ARMAacf` produces the theoretical ACF values for an AR model. Give it the vector of ϕ_i values as the `ar` argument, and you want to set `lag.max` to see more lags. Plot the results with `plot(type = 'h')` for the classic vertical bar appearance.

   Check some of your simulated series against the theoretical ACF.

7. Summarize what you've learned about how AR(2) series depend on ϕ_1 and ϕ_2.

Nile

8. Use `ar` to fit an AR(p) model to the built-in Nile time series. What value of p did it choose? What were the values of the coefficients ϕ and the standard deviation σ?

9. Generate a few random $N = 100$ term AR series with the same ϕ and σ values as the Nile series. How do they compare with the original data?