DSC 152:
Applied Statistical Data Analysis and Inference

Lecture #19
Time Series Regression
Power Estimation

Tuesday, June 2
Spring Quarter 2026
Peter Chi

Recall yet again. . .

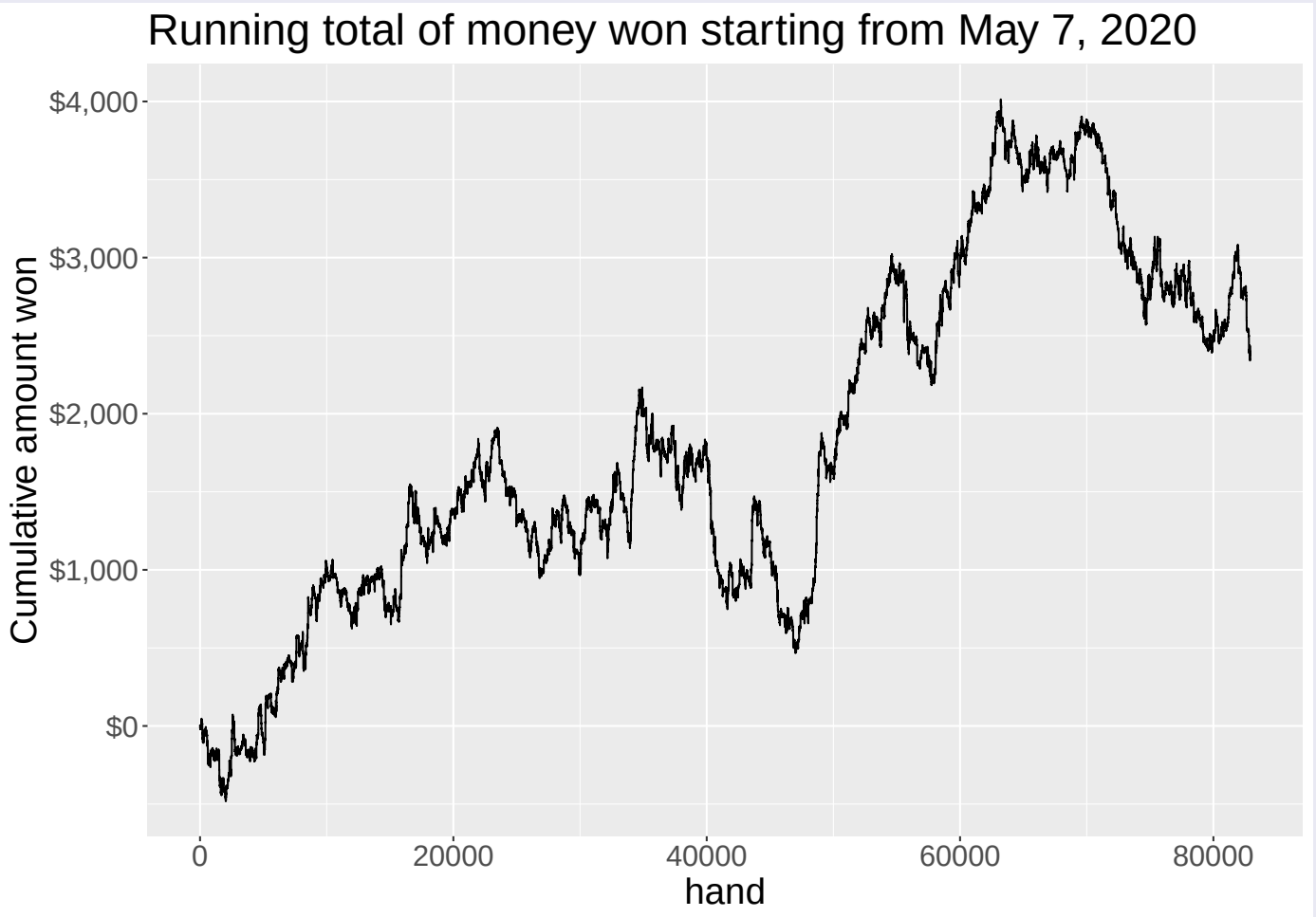
Conditions for validity of inference in linear regression

- The relationship between X and Y, if there is one, is actually **L**inear
  - e.g. not quadratic, exponential, etc.
- **I**ndependence of observations
- **N**ormality of ϵ_i
  - Note that this can also be achieved, due to the central limit theorem, with a large sample size even if ϵ_i does not follow a normal distribution
- **E**qual variance across all values of X
  - Also known as homoskedasticity

How much does it actually matter?

What happens if the data actually are time-dependent, but we ignore that in our analysis?

Recall the poker example. . .



Simulation under an AR(p) model

Recall: an AR(2) model

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t$$

Now suppose the data generating process is this AR(2) model with:

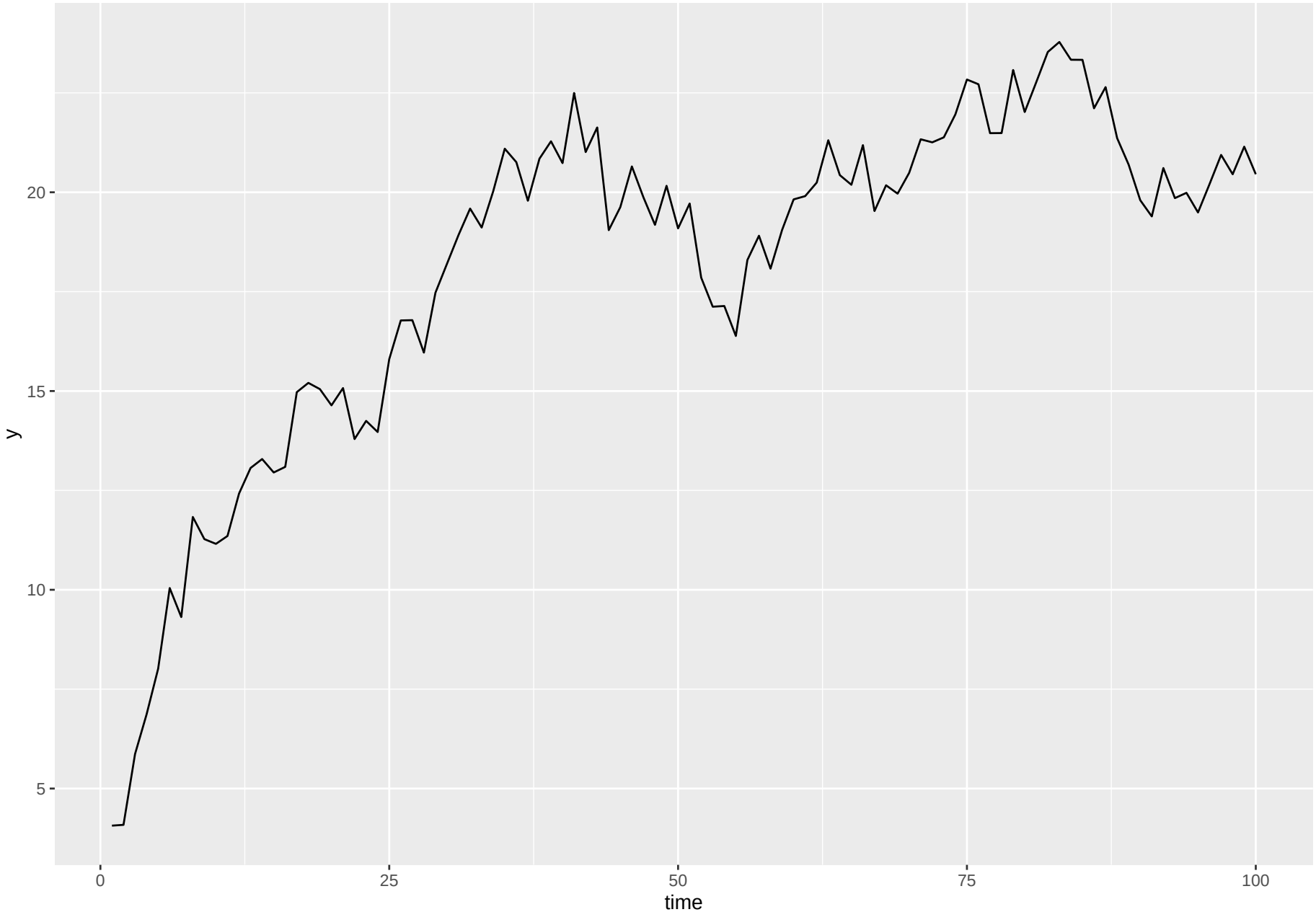
- $c = 2$
- $\phi_1 = 0.65$
- $\phi_2 = 0.25$
- $\epsilon_t \sim N(0, 1)$

which would give:

$$y_t = 2 + 0.65 y_{t-1} + 0.25 y_{t-2} + \epsilon_t$$

Recall that c is the drift parameter...

Simulation under an AR(p) model



Simulation under an AR(p) model

```
library(ggplot2)
set.seed(3)
eps <- rnorm(102, 0, 1)

# We need two burn-in values
y <- 2 + eps[1]
y[2] <- 2 + 0.65*y + eps[2]
for(i in 3:102){
  y[i] <- 2 + 0.65*y[i-1] + 0.25*y[i-2] + eps[i]
}

time <- 1:100
y <- y[3:102] # discard the burn-in
df <- data.frame(time=time, y = y)

ggplot(df, aes(x=time, y=y)) + geom_line()
```

Simulation under an AR(p) model

What would a t-test give on the differences for this particular simulation?

```
diffs <- NULL
for(i in 1:99){
  diffs[i] <- y[i+1] - y[i]
}

t.test(diffs)
```

Wait why on the differences?

Simulation under an AR(p) model

For this particular simulation:

```
t.test(diffs)
```

```
##
##  One Sample t-test
##
## data:  diffs
## t = 1.7389, df = 98, p-value = 0.08519
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0233729  0.3544216
## sample estimates:
## mean of x
## 0.1655244
```

Simulation under an AR(p) model

```
library(forecast)
library(lmtest)

model2 <- Arima(y, order=c(2,0,0), xreg=time)
coeftest(model2)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ar1         0.789643   0.097974  8.0597 7.645e-16 ***
## ar2         0.184329   0.099031  1.8613  0.062697 .
## intercept 7.345378   3.990363  1.8408  0.065654 .
## xreg        0.159421   0.050579  3.1519  0.001622 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Simulation under an AR(p) model

What did we observe?

In this particular case, we fail to reject H_0 with the t-Test, but correctly reject H_0 when we run the AR(2) model.

What would happen on average?

Simulation under an AR(p) model

Your Turn #1

First let's write two functions:

- One to take in inputs of:
  - n
  - ϕ_1
  - ϕ_2
  - and then output a dataframe of `y` and `time`. You can hardcode `c=2` and `sd=1`.
- One to calculate `diffs` as in Slide 7

Then, use the function...

to run a simulation study of 1000 reps with
$n = 100, \phi_1 = 0.65, \phi_2 = 0.25$ and compare power with:

- the `t.test` (ignoring the AR(2) structure of the data)
- `Arima` (properly accounting for the AR(2) structure of the data)

Finally, repeat with $\phi_1 = -0.65, \phi_2 = -0.25$.

Now, how about in Time Series *Regression*?

Recall the Jalen Brunson example from last time...

- The covariate of interest was the opponent's defensive rating
- In the simulated example, we generated data under an MA(1) model
- We observed that in the analysis with `lm`, we failed to reject H_0, whereas in the analysis with `Arima`, we correctly rejected H_0.

But that was just one iteration. What happens on average?

Let's stick with an AR(2) model as in the previous example from today, and simulate without context...

Now, how about in Time Series *Regression*?

Suppose the true model is:

$$y_t = \beta_0 + \beta_1 x_t + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t$$

where:

- $\beta_0 = 0$
- $\beta_1 = 2$
- $\phi_1 = 0.65$
- $\phi_2 = 0.25$
- $\epsilon_t \sim N(0, 3)$

Let's do a simulated power comparison!

Now, how about in Time Series *Regression*?

Your Turn #2

First, together we will write a new function that will be a modification of the `gen_AR2` function from Your Turn #1.

- β_0, β_1 and the distribution of ϵ_t can be hardcoded.
- Let the x values come from a Unif(0, 10) distribution (which can also be hardcoded)
- You'll need two burn-in values in a similar manner to that of Your Turn #1

Now, how about in Time Series *Regression*?

Your Turn #2

Next, use that function to write a simulation comparing:

- the statistical power with the `lm` function (ignoring the time dependency structure)

- the statistical power with `Arima` (accounting for the time dependency structure)

Recap and Looking Ahead

Recap

Failing to account for time dependency in your data can lead to incorrect inference. In every example here, we showed a loss of power, but other things could happen.

Looking Ahead

- Thursday's class: Final Exam Review
- Saturday: Final Exam from 3-6pm. EVERYONE will be in SOLIS 107

Today's Daily Check

The two Your Turns

Quiz 3 Review