Intro to Time Series Regression

William Chiu

About Me

- MSIA Class of 2013
- Works for a bank
- Team develops models to explain and forecast:
  - Credit Losses
  - Loan Prepayments
  - Mortgage Rates
  - Deposit Attrition
- Primarily uses SQL and R

Time Series Regression

- Time series is data collected in equally-spaced time intervals.
- Linear regression estimates the linear relationship between a continuous response (`y`) and one or more predictors (`x`).
- Ordinary least squares (OLS) is the most common implementation of linear regression, and estimates the coefficients that minimize the error sum of squares:

$$ESS = \sum_{i=1}^{n} \left(y_i - (\widehat{\beta_0} + \widehat{\beta_1} X_{1,i} + \ldots + \widehat{\beta_p} X_{p,i}) \right)^2$$

- Using linear regression on time series is called time series regression.

OLS Assumptions for Time Series

- Population data generating process (dgp) most suitable for OLS:

$$y_t = \beta_0 + \beta_1 X_{1,t} + \ldots + \beta_p X_{p,t} + \varepsilon_t$$
$$\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

- Linear parameters
- Independent errors (no serial correlation)
- Normally distributed errors
- Equal variance errors (homoscedasticity)
- Stationary errors

Violation: Serial Correlation

- Unfortunately, the data generating process (dgp) for time series often violates the independent errors assumption. For example:

$$y_t = \beta_0 + \beta_1 X_{1,t} + \ldots + \beta_p X_{p,t} + \varepsilon_t$$
$$\varepsilon_t = \rho \varepsilon_{t-1} + v_t$$
$$v_t \sim \mathcal{N}(0, \sigma^2)$$

- If $\rho = 0$, then there is no serial correlation.
- If $-1 < \rho < 1$, then there is serial correlation.

Simulate predictors with serial correlation

```
set.seed(2013)

x1 <- arima.sim(list(order = c(1,0,0), ar = 0.6), n =
200, sd=2, mean=2)
x2 <- arima.sim(list(order = c(1,0,0), ar = -0.6), n =
200, sd=3, mean=2)
x3 <- arima.sim(list(order = c(1,0,0), ar = 0.6), n =
200, sd=4, mean=2)
```

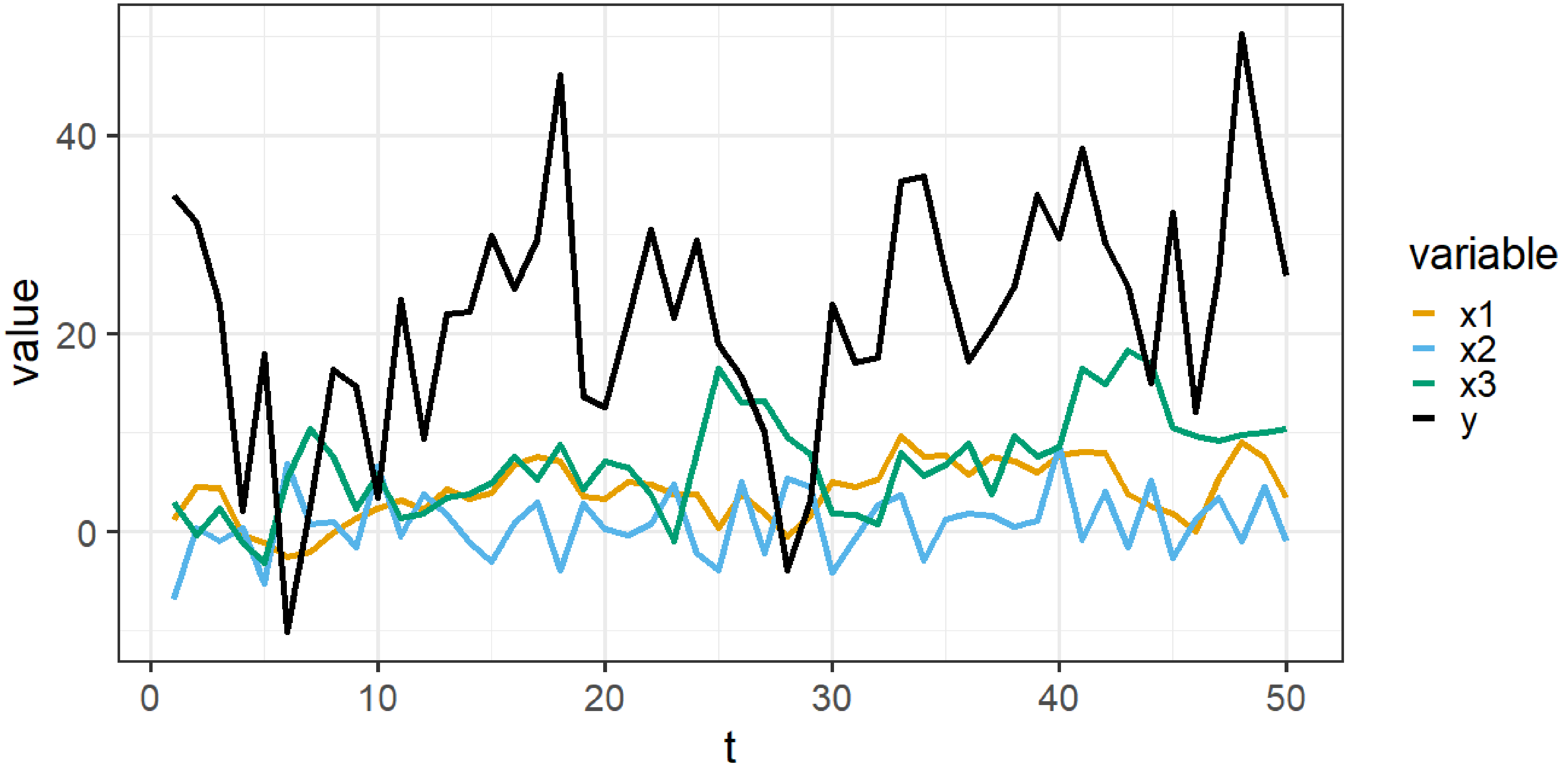
Simulate response with serial correlation

```
epsilon <- arima.sim(list(order = c(1,0,0), ar = 0.6), n
= 200, sd=5, mean=0)

y <- 10 + 3.5*x1 - 1.5*x2 + epsilon
## x3 is not part of the dgp

df <- data.frame(y=y, x1=x1, x2=x2, x3=x3, t=1:length(y))
```

Plot simulated data (first 50 obs)



Split between train and test

```
train <- head(df, 150)
test <- tail(df, 50)
```

OLS ignores serial correlation

- Under serial correlation, OLS parameter estimates are unbiased and consistent
- However, standard errors are incorrect
- And p-values are incorrect
- `x3` should *not* be statistically significant

term	estimate	std.error	statistic	p.value
(Intercept)	8.98	1.147	7.84	0.000
x1	3.61	0.204	17.74	0.000
x2	-1.50	0.121	-12.40	0.000
x3	0.17	0.101	1.68	0.094

Detecting serial correlation

- Durbin-Watson test is a popular test for serial correlation, but suffers from some major drawbacks.
- Breusch-Godfrey Test is a better test.

statistic	p.value	parameter	method
37.4	9.84e-10	1	Breusch-Godfrey test for serial correlation of order up to 1

Null Hypothesis: No serial correlation

Remediating serial correlation

- Replace OLS with GLS
- Keep OLS but estimate robust (HAC) standard errors
- Keep OLS but bootstrap the standard errors

Generalized Least Squares (Cochrane–Orcutt)

1. Estimate ρ
2. Transform each response and predictor into "partial differences" (e.g., $y_t - \hat{\rho}y_{t-1}$)
3. Estimate the transformed model using OLS
4. Re-calculate the coefficient estimates in terms of the original training data

term	estimate	std.error	statistic	p.value
(Intercept)	9.40	1.431	6.566	0.000
x1	3.60	0.224	16.064	0.000
x2	-1.47	0.078	-18.933	0.000
x3	0.10	0.116	0.863	0.389

Newey-West HAC Correction

- Keep OLS but estimate robust (HAC) standard errors

term	estimate	std.error	statistic	p.value
(Intercept)	8.98	1.067	8.42	0.000
x1	3.61	0.237	15.26	0.000
x2	-1.50	0.087	-17.33	0.000
x3	0.17	0.123	1.38	0.169

Andrews HAC Correction

- Keep OLS but estimate robust (HAC) standard errors

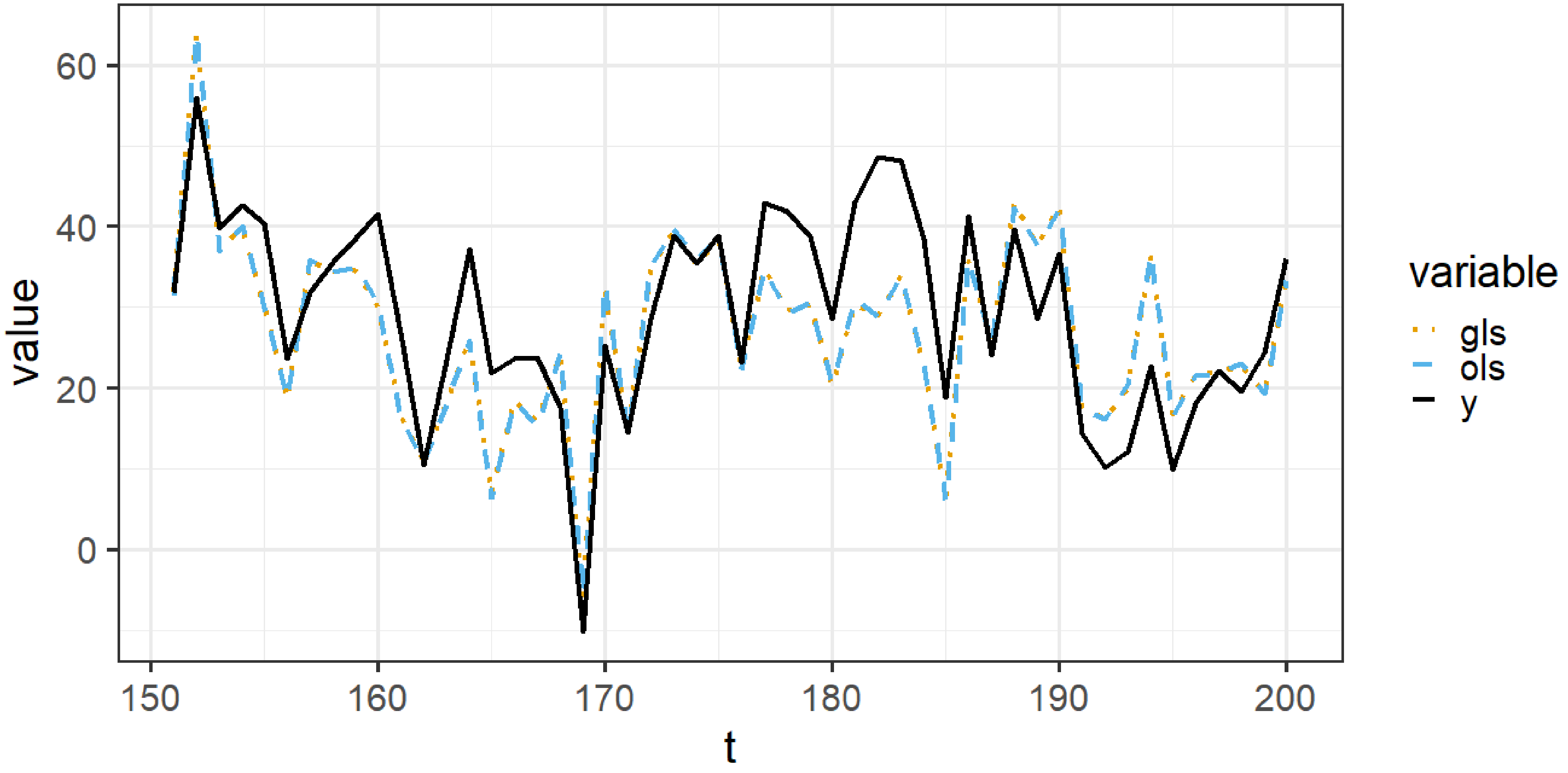
term	estimate	std.error	statistic	p.value
(Intercept)	8.98	1.119	8.03	0.000
x1	3.61	0.247	14.62	0.000
x2	-1.50	0.086	-17.58	0.000
x3	0.17	0.114	1.49	0.139

Block Bootstrap

- Keep the OLS coefficient estimates but *bootstrap* the standard errors
- Break the time series into sequential blocks (non-random)
- Randomly sample blocks with replacement
- Unfortunately, results are sensitive to block size

term	estimate	std.error	statistic	p.value
(Intercept)	8.98	1.076	8.35	0.000
x1	3.61	0.221	16.35	0.000
x2	-1.50	0.088	-17.11	0.000
x3	0.17	0.126	1.36	0.177

Test Set Performance (Ex-post, OLS vs. GLS)



Inference is harder than prediction

- Test set performance appears similar between OLS and GLS
- However, the p-values for `x3` differ substantially by method: `lm`, `gls`, `newey-west`, `andrews`, `bootstrap`.
- The p-values from simulated data are *sensitive* to the random seeds
- In order to measure the quality of the methods:
  1. Re-simulate the training data ($K = 1000$) with varying seeds and ρ
  2. Measure how often Type 1 errors occur ($\alpha = 0.10$)
  3. Measure test set error (i.e., average MSE over K simulations, then take the square root)
- Since $\beta_3 = 0$ in the simulation, any p-values <= α is a Type 1 Error

Type 1 Error Frequency

rho	method	Prob_Type_1_Error	Avg_Test_RMSE
1-Low (0.1)	gls	0.109	5.12
1-Low (0.1)	lm	0.114	5.12
1-Low (0.1)	bootstrap	0.120	NA
1-Low (0.1)	andrews	0.128	NA
1-Low (0.1)	nw	0.138	NA

rho	method	Prob_Type_1_Error	Avg_Test_RMSE
2-Moderate (0.7)	gls	0.108	7.10
2-Moderate (0.7)	andrews	0.132	NA
2-Moderate (0.7)	nw	0.148	NA
2-Moderate (0.7)	bootstrap	0.174	NA
2-Moderate (0.7)	lm	0.319	7.21

Test Set RMSE

$$RMSE_{test} = \sqrt{\sigma_\varepsilon^2} = \sqrt{\frac{\sigma_v^2}{1-\rho^2}}$$

The Pennsylvania State University. (2018). 10.3 - Regression with Autoregressive Errors. Applied Regression Analysis. Retrieved February 27, 2022, from https://online.stat.psu.edu/stat462/node/189/

Very High Serial Correlation

- Under very high serial correlation, most of the previous methods struggle to return the correct p-values.
- If $\hat{\rho}$ is close to -1 or 1, then check for non-stationary errors (e.g., Phillips-Ouliaris Cointegration Test; Pesaran-Shin-Smith Cointegration Test).

rho	method	Prob_Type_1_Error	Avg_Test_RMSE
3-Very High (0.9)	gls	0.096	12.0
3-Very High (0.9)	andrews	0.186	NA
3-Very High (0.9)	nw	0.195	NA
3-Very High (0.9)	bootstrap	0.233	NA
3-Very High (0.9)	lm	0.550	12.7

Violation: Non-stationary errors

- When $\rho = 1$ or $\rho = -1$, the errors are non-stationary (i.e. “random walk”).

$$\varepsilon_t = \rho\varepsilon_{t-1} + v_t$$

- Under non-stationary errors, OLS and GLS are both biased and inconsistent (i.e., spurious)
- Non-stationary errors adversely affect *both* prediction and inference (while serially correlated errors affect only inference)

Simulate non-stationary predictors

```
set.seed(2013)

w1 <- arima.sim(list(order = c(0,1,0)), n = 200, sd=2,
mean=2)
w2 <- arima.sim(list(order = c(0,1,0)), n = 200, sd=3,
mean=2)
w3 <- arima.sim(list(order = c(0,1,0)), n = 200, sd=4,
mean=2)
```

Simulate non-stationary response

- Suppose z_t is a random walk and does **not** depend on any predictors

$$z_t = z_{t-1} + v_t$$
$$v_t \sim \mathcal{N}(0, \sigma^2)$$

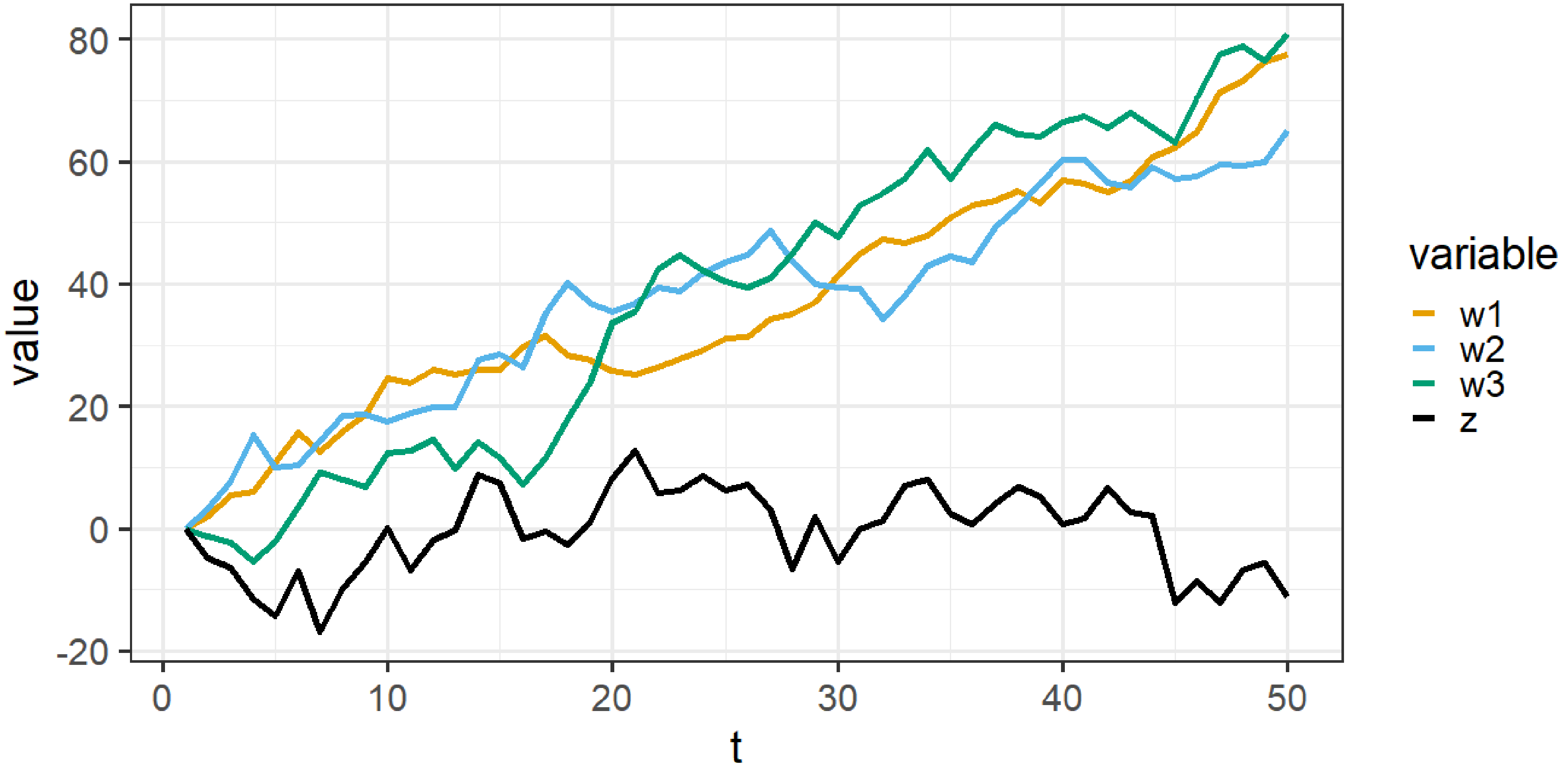
- The predictors w_1, w_2, w_3 should not affect z_t

```
z <- arima.sim(list(order = c(0,1,0)), n = 200, sd=5,
mean=0)

## w1, w2, w3 are not part of the dgp (and z is a random
walk)

df_nonstationary <- data.frame(z=z, w1=w1, w2=w2, w3=w3,
t=1:length(z))
```

Plot simulated data (first 50 obs)



Split between train and test

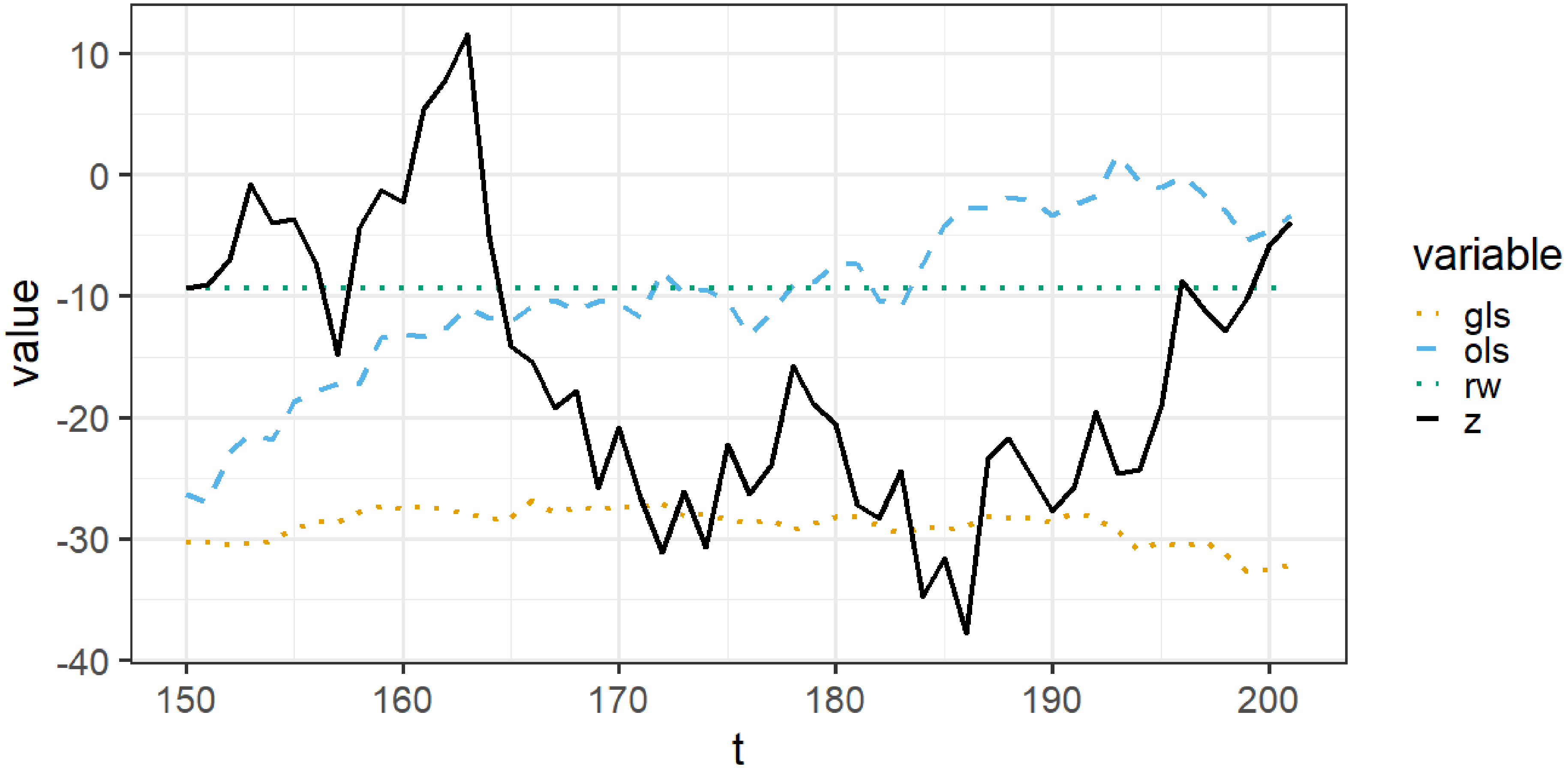
```
train_ns <- head(df_nonstationary, 150)
test_ns <- tail(df_nonstationary, 52) # helps with
plotting
```

OLS ignores non-stationary errors

term	estimate	std.error	statistic	p.value
(Intercept)	-0.291	1.338	-0.217	0.828
w1	0.234	0.086	2.718	0.007
w2	0.173	0.065	2.675	0.008
w3	-0.441	0.042	-10.598	0.000

r.squared	adj.r.squared
0.736	0.731

Test Set Error Explodes



Remediating non-stationary errors

1. Transform each response and predictor into “first differences” (e.g., $y_t - y_{t-1}$)
2. Estimate the transformed model using OLS or GLS
3. Forecast with the last historical value and the cumulative sum of the predicted first differences:

$$\widehat{y_{t+h}} = y_t + \sum_{i=t+1}^{t+h} \widehat{\Delta y_i}$$

Conclusions

- OLS is most suitable for independent errors
- The independent error assumption is often violated with time series regression
- Under serial correlation, standard errors and p-values from OLS are unreliable
- Fortunately, OLS coefficient estimates remain unbiased and consistent
- Remediating serial correlation:
  - GLS
  - OLS with HAC standard errors
  - OLS with block bootstrapped standard errors
- Under non-stationary errors, beware of spurious regressions