Time Series

WUDAC Analytics 101

27 November 2018

Contents

- What is “time series” and what are its applications?
- The data-generating process
- Trend and seasonality
- Serial correlation

What is "time series" and what are its applications?

- Like any other regression, except your dependent variable (y) changes **over time**
- Used for forecasting and prediction
- Applications: Mostly social science!
  - Economics
  - Finance
  - Business
  - Political science

The data-generating process

- Say that we want to forecast a certain variable. Let's call this variable y. We first set it up in a regression on many other predictive variables:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_K x_{K,t} + \varepsilon_t,$$

  where we call β_0 the **intercept**, the other β_i's the **(partial) slope coefficients**, and the corresponding x_i's the **features**. The ε at the end is called the **error**.
- **Note:** We assume that errors of the regression have a normal distribution with mean zero and a constant variance σ^2:

$$\varepsilon_t \sim \mathcal{N}\left(0, \sigma^2\right).$$

Our final goal: Reducing the residuals to white noise

- Define a **residual** as the difference between the actual value of what we are interested in forecasting and our forecasted value produced by our model:
$$e_t \stackrel{\text{def}}{=} y_t - \hat{y}_t.$$
- Our residuals $\{e_t\}$ are a great estimate of the errors of our underlying model, $\{\varepsilon_t\}$. Since we assume our errors to be randomly normally distributed around zero, we also want our residuals to have such a distribution (we will say we want our residuals to look like **white noise**). **This will be our primary criterion for model evaluation**.

Trend

- Sometimes, the variable that we are trying to explain (y) has long-term trends that we try to explain. We can capture this effect by simply regressing on time:

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t.$$

- If we want to capture nonlinear effects of changes in time on our variable of interest, then we can add a time-squared term to the regression (This is called a **second-order Taylor approximation**):

$$y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \varepsilon_t.$$

Seasonality

- Sometimes our variable of interest y may systematically behave certain ways during some seasons and other times during other seasons. We can capture this effect in our model by regressing on **seasonal dummies**: a set of binary variables (0 or 1) that indicate which season y_t falls in.
- We can define "seasons" any way we want. If we have K seasons, we should include a maximum of $K-1$ seasonal dummies in the regression to avoid collinearity among the features and the intercept:

$$y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 \mathrm{summer}_t + \beta_4 \mathrm{fall}_t + \beta_5 \mathrm{winter}_t + \varepsilon_t.$$

For example, in this regression, summer_t is 1 when t is in the summer and 0 otherwise.

Serial Correlation

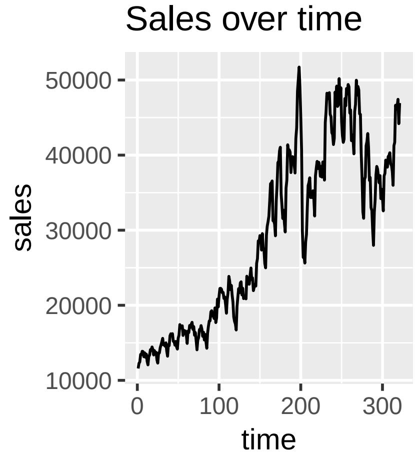
- After we regress on trend and seasonality, we may still notice that the residuals aren't quite white noise. This suggests that our model is not quite fully specified—It means that we're leaving predictive features on the table.
- A prime example of one of such characteristics we might observe is persistence of residuals. We can capture these effects by regressing on lagged values of the dependent variable:

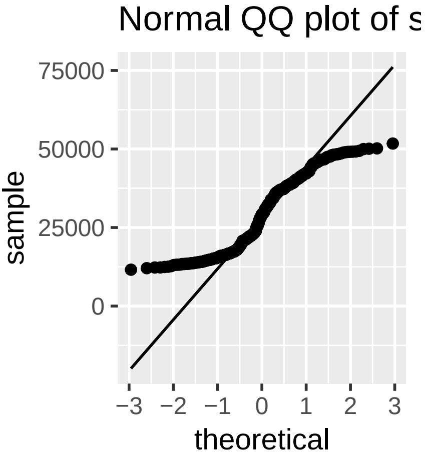
$$\begin{aligned} y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 \mathrm{summer}_t \\ + \beta_4 \mathrm{fall}_t + \beta_5 \mathrm{winter}_t + \beta_6 y_{t-1} + \varepsilon_t. \end{aligned}$$

  But we could regress on many more than just one lagged value. But how do we decide how many lags to regress on? We can use AIC or BIC to decide.

Our Example: Monthly U.S. Gasoline Sales from 1992 to 2018

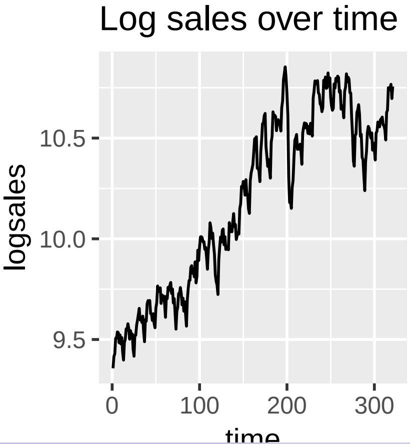
- Let's look at the distribution of sales.

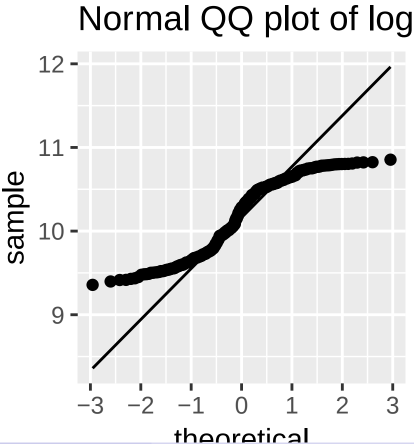




Logging the data

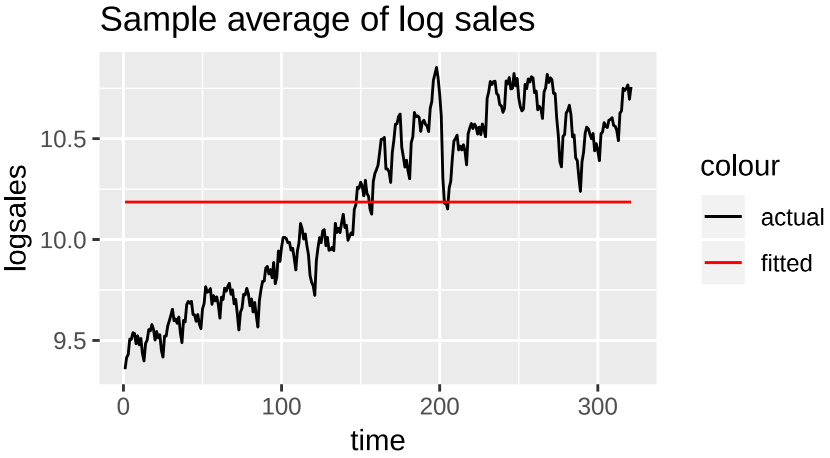
- Let's try logging our data to tighten the variance. We're also more interested in **changes** in gasoline sales, and considering the logarithm as our dependent variable allows us to do that when we interpret our model later on.





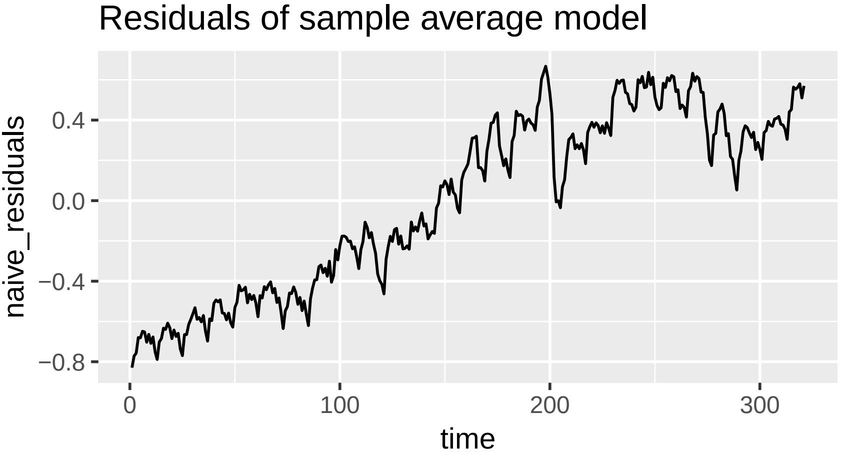
A naive model: sample average

- Let's build a naive regression of log sales on time. What if we just took the sample average?



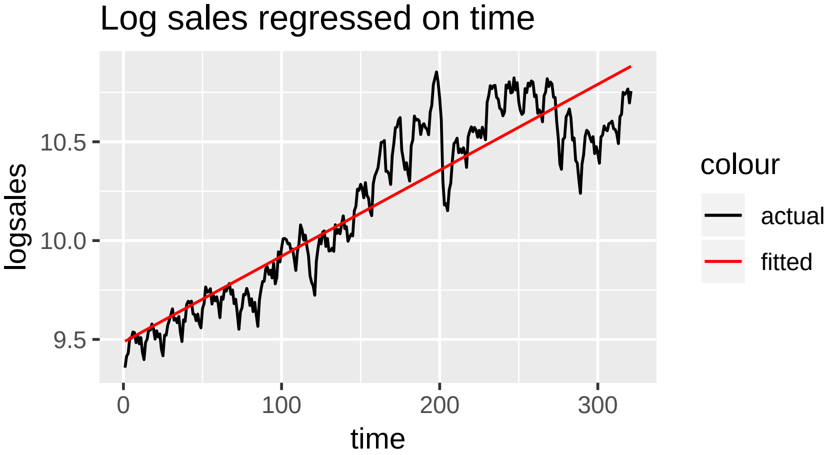
A naive model: sample average

- Note that our residuals are certainly anything **but** white noise:



A better model: regressing on time (trend)

- What if we also regressed on a time vector?



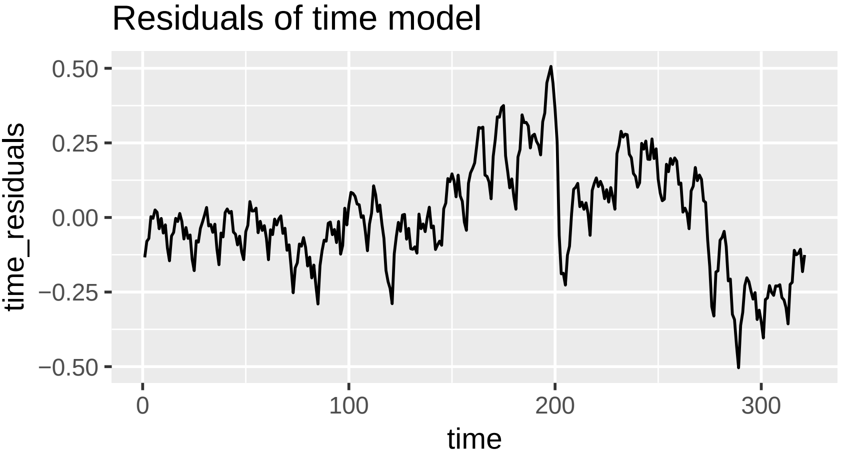
A better model: regressing on time (trend)

- This kind of upward slope is referred to as **trend**. Let's do a significance test to see if this is a statistically significant phenomenon in our model:

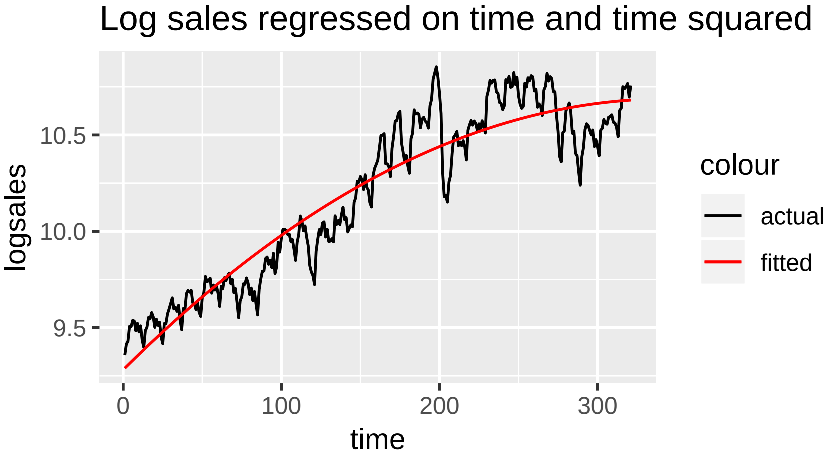
```
##                Estimate      Pr(>|t|)
## (Intercept) 9.486378522  0.000000e+00
## time        0.004348041 4.003506e-131
```

A better model: regressing on time (trend)

- Our residuals now closer to white noise, but it defintely doesn't look random. There's larger some slope to it that we could perhaps take advantage of.



Picking up second-order time effects



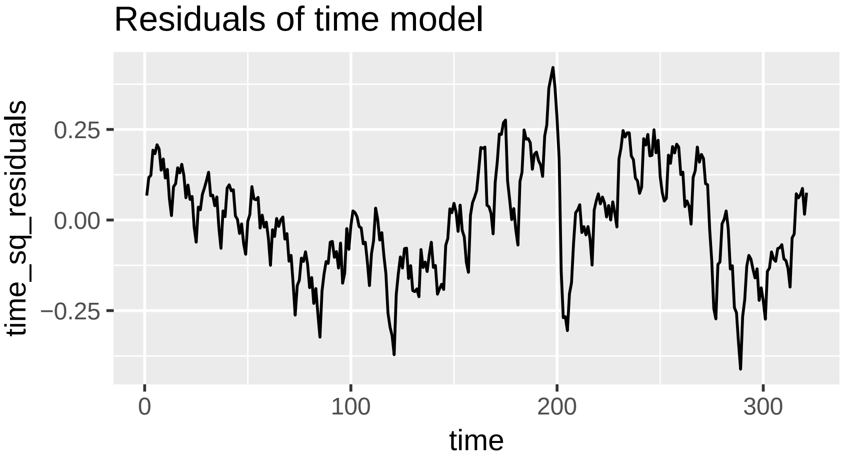
Picking up second-order time effects

- All our variables are statistically significant:

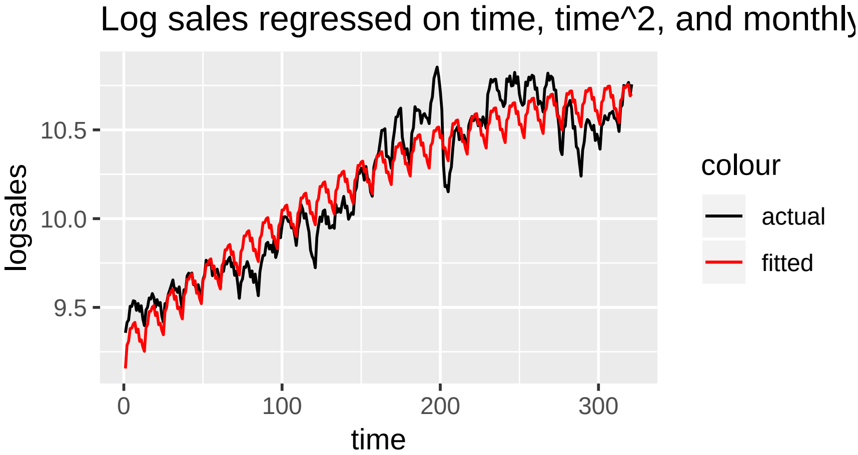
```
##                   Estimate      Pr(>|t|)
## (Intercept)  9.281430e+00 0.000000e+00
## time         8.155142e-03 3.721060e-69
## time2       -1.182329e-05 3.675302e-24
```

Picking up second-order time effects

- Our residuals now look more random than before, but there's a recurring cyclical pattern we can take advantage of.



Seasonality

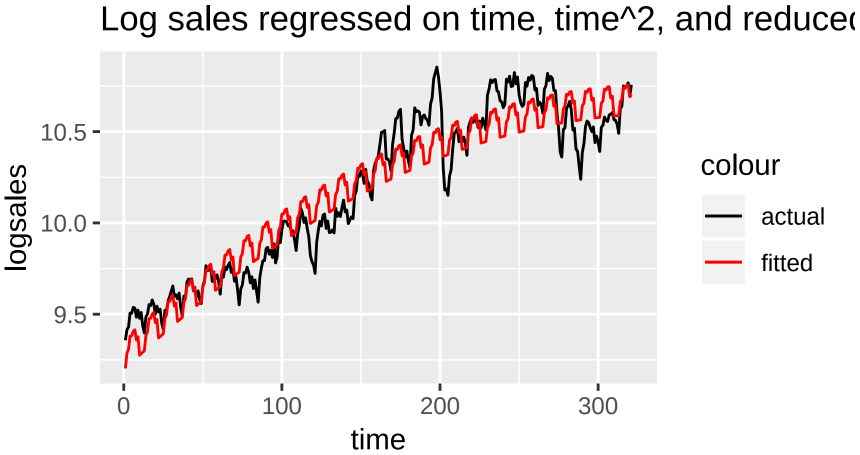


Seasonality

Taking a look at a full statistical summary of the model, notice that the February, November, and December dummies are insignificant even at the 0.1 level. Let's pull these out of the model.

```
##
## Call:
## lm(formula = logsales ~ time + time2 + month, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27771 -0.11358  0.00264  0.11138  0.34147
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.181e+00  3.397e-02 270.293  < 2e-16 ***
## time         8.203e-03  3.227e-04  25.423  < 2e-16 ***
## time2       -1.200e-05  9.705e-07 -12.363  < 2e-16 ***
```

Seasonality



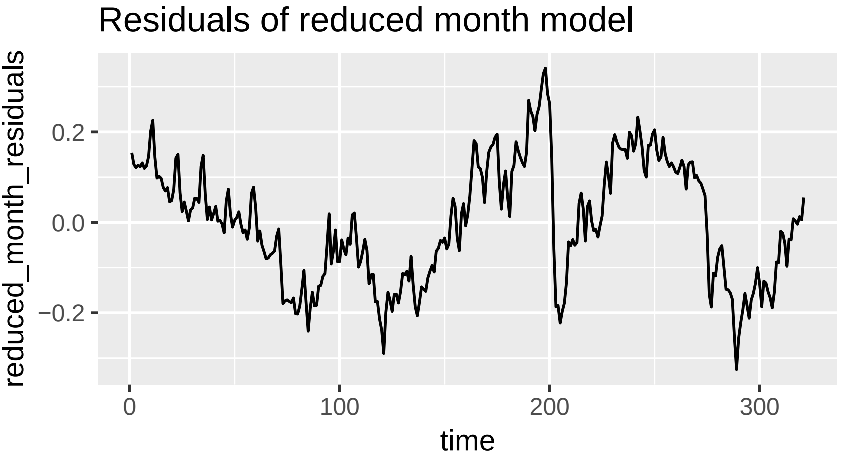
Seasonality

- Now, we see that all the features are significant:

```
##
## Call:
## lm(formula = reduced_month_formula, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32533 -0.10980  0.00263  0.11489  0.34109
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        9.195e+00  2.522e-02 364.580  < 2e-16 *
## time               8.219e-03  3.246e-04  25.323  < 2e-16 *
## time2             -1.204e-05  9.762e-07 -12.335  < 2e-16 *
## I(month == 3)TRUE  7.570e-02  2.898e-02   2.612 0.009429 *
## I(month == 4)TRUE  8.830e-02  2.898e-02   3.047 0.002507 *
```

Seasonality

- What do our residuals look like now?



Seasonality

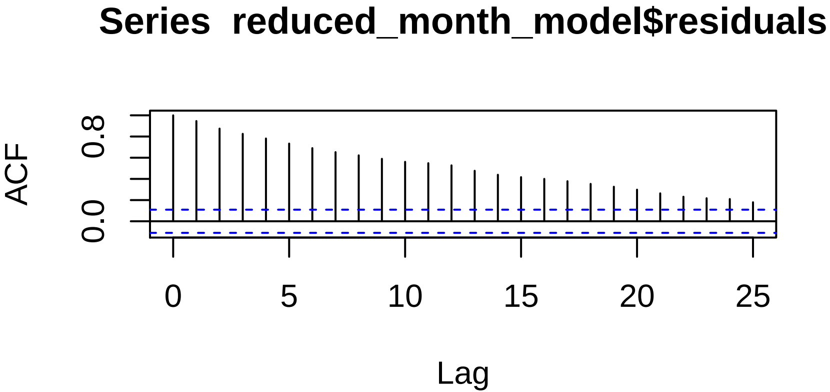
- Notice how our residuals show **persistence**: positive for a while, then negative for a while... This is caused by business cycles! But how do we capture this?

Regressing on lagged log sales

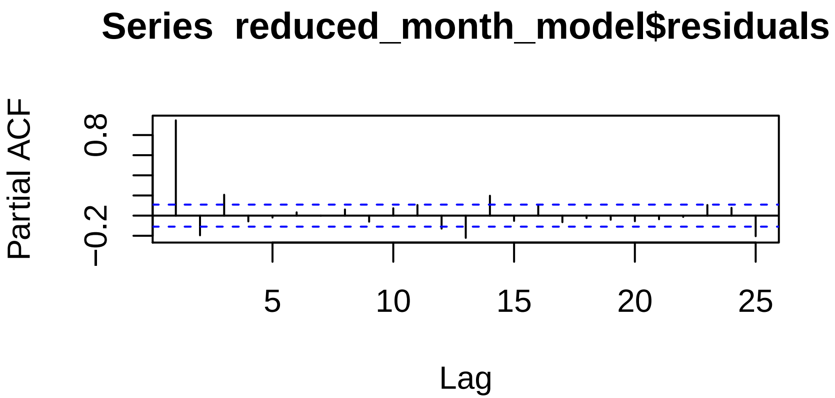
- Looking at the correlations of log sales with past values of log sales

```
##
##  Durbin-Watson test
##
## data:  reduced_month_model
## DW = 0.10295, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater tha
```

Regressing on lagged log sales



Regressing on lagged log sales

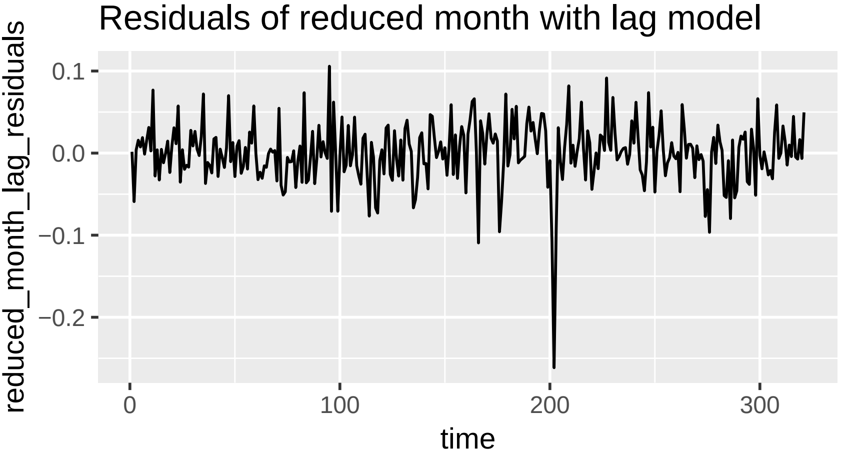


Regressing on lagged log sales

- Let's build the model.

Regressing on lagged log sales

- Looking at the residuals:



Testing forecasting accuracy