

# Solving Forecasting Problems in R and Python

John Mount, Ph.D.  
Win Vector LLC  
[jmount@win-vector.com](mailto:jmount@win-vector.com)  
<https://www.win-vector.com>

PyData Global 2024 <https://global2024.pydata.org/cfp/talk/3NYJVA>  
12-05, 17:00–17:30 (UTC), General Track

All slides, code, and data here: <https://github.com/WinVector/Examples/tree/main/TimeSeries>



John Mount

Win Vector Principal Consultant and Trainer. John Mount has a Ph.D. in computer science from Carnegie Mellon and over 15 years of applied experience in biotech research, online advertising, price optimization and finance. He is one of the authors of the popular book "Practical Data Science with R", Manning, 2020 (now in its second edition).

1



Hold on this slide until ready to start.

## Motivation

"One of the biggest mistakes in my data science career was not paying enough attention to time series models.

Time series is everywhere—finance, sales, marketing—and it can drastically improve your decision-making process."

— Mark Eltsefon, Staff Scientist at Meta

[https://www.linkedin.com/posts/mark-eltsefon\\_datasciencetimeseries-activity-7253363004694491137-94D8?utm\\_source=share&utm\\_medium=member\\_desktop](https://www.linkedin.com/posts/mark-eltsefon_datasciencetimeseries-activity-7253363004694491137-94D8?utm_source=share&utm_medium=member_desktop)

2



Forecasting problems tend to be high value.

## Outline

- Motivation
- What is time series forecasting?
- Families of methodologies
- Our example problem
- Solving the problem using Stan
- What is Stan?
- Results/Observations



3



We will fill in some details as we go.

## What is Time Series Forecasting?

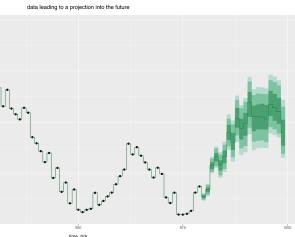
(a slightly heterodox view)

4

4

## Time Series Forecasting

- Produce an estimate of the future using trends and relations observed in the past.

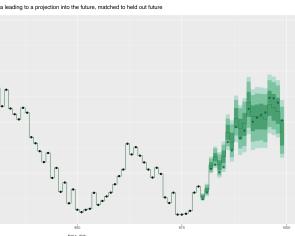


5

The dots are actual counts as a function of time, and the green line is the central prediction with uncertainty bands.

## How to Evaluate Forecasts

- Hold out future, and compare prediction to eventual outcomes.



6

A prediction is good if it matches future data. Don't get bullied out of this.

## Business Applications

- Predicting future demand
  - Online sales
  - Parking
  - Restaurant
- Predicting future price
  - Expenses
  - Stocks
- Predicting future revenue
  - Financial reporting/planning
- Physical systems affecting business
  - Tides
  - Weather



In 1876 A. Lége & Co., 20 Cross Street, Hatton Gardens, London completed the first "tide calculating machine" for William Thomson (later Lord Kelvin)

## de-Motivation

"We have found that choosing the wrong model or parameters can often yield poor results, and it is unlikely that even experienced analysts can choose the correct model and parameters efficiently given this array of choices."

– Facebook research announcement of Prophet, 2017  
<https://research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/>

7

It is easy to get your hand caught in the gears.

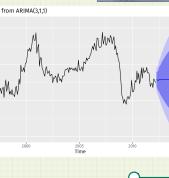
9

## Example

From a manual on how to fit an ARIMA time series forecast model.

Two pages of deep technical analysis justifying the modeling choices and confirming technical modeling assumptions.

Followed by failing to call out that this is obviously a degenerate fit!



At no point in the training period was the curve flat. But that is precisely the predicted expected value. Yes, models of expected tend to have less variance than actuals <https://win-vector.com/2024/06/15/good-models-do-not-match-variance/>. However, this is a degenerate fit. Most references avoid making the obvious by the simple expedient of never showing the final graph. Can't over emphasize the importance of insisting on method-agnostic hold out tests.

## External Regressors

- For many business applications we need to model future values as a function of past values *and* additional facts about the future called “external regressors.”
- This is where it all goes *wrong*.
  - ARMAX, ARIMAX, SARIMAX, transfer functions, regression with ARIMA residuals all *claim* to solve this.
  - No two of them seem to be solving the same problem (if you can even find a description of what problem they claim to solve).
    - <https://win-vector.com/2024/07/15/armax-offerings-remain-a-muddle/>
    - The ARIMAX model muddle <https://robjhyndman.com/hyndtsight/arimax/>
- To the extent time series modeling is magic, it obscures out the effects of external regressors.

10

There is a fiction that ARMAX is how time series researches work. That appears to be wishful thinking. Time series research moved on to transfer functions and other terms of art.

## Families of Methodologies

11

One can really get lost in the terminology.

## Methodology Families

- ARIMA derived methods
- Brute force methods
- Decomposition or attribution methods
- Hidden state inference

12

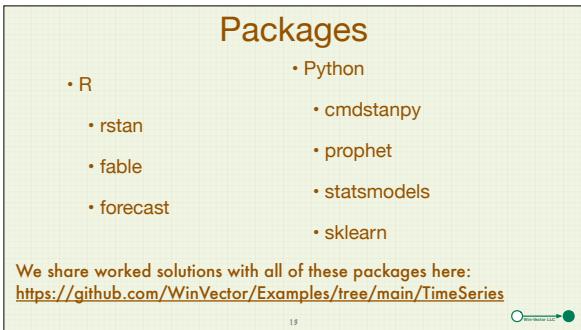
We can crudely put things like exponential smoothing under the “ARIMA” hat.

**Packages**

- R
- rstan
- fable
- forecast
- Python
- cmdstanpy
- prophet
- statsmodels
- sklearn

We share worked solutions with all of these packages here:  
<https://github.com/WinVector/Examples/tree/main/TimeSeries>

13



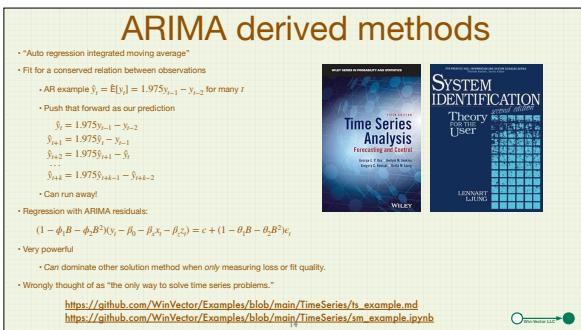
One advantage of R packages: Hyndman's writing is quite good.

**ARIMA derived methods**

- "Auto regression integrated moving average"
- Fit for a conserved relation between observations
- AR example  $\hat{y}_t = E[y_t] = 1.975y_{t-1} - y_{t-2}$  for many  $t$
- Push that forward as our prediction
- $\hat{y}_1 = 1.975y_0 - y_{-2}$
- $\hat{y}_2 = 1.975y_1 - y_{-1}$
- $\hat{y}_3 = 1.975y_2 - \hat{y}_1$
- ...
- $\hat{y}_{k+1} = 1.975\hat{y}_k - \hat{y}_{k-2}$
- Can run away!
- Regression with ARIMA residuals:
$$(1 - \theta_1B - \theta_2B^2)(y_t - \beta_0 - \beta_1y_t - \beta_2z_t) = c + (1 - \theta_1B - \theta_2B^2)\epsilon_t$$
- Very powerful
- Can dominate other solution method when only measuring loss or fit quality.
- Wrongly thought of as "the only way to solve time series problems."

[https://github.com/WinVector/Examples/blob/main/TimeSeries/ts\\_example.md](https://github.com/WinVector/Examples/blob/main/TimeSeries/ts_example.md)  
[https://github.com/WinVector/Examples/blob/main/TimeSeries/im\\_example.ipynb](https://github.com/WinVector/Examples/blob/main/TimeSeries/im_example.ipynb)

14



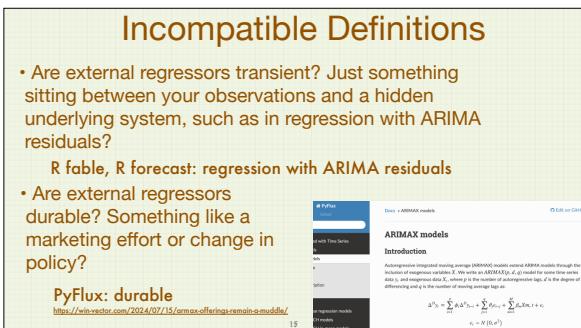
Among the most powerful methods. The "get k steps out by running a 1 step out process k times" is very powerful, but also very sensitive. The epsilon terms are called shock or noise. It is a very bad idea to think of them as noise as they are often important external drivers (such as deposits to a bank account on in-flow to a reservoir).

**Incompatible Definitions**

- Are external regressors transient? Just something sitting between your observations and a hidden underlying system, such as in regression with ARIMA residuals?
- R fable, R forecast: regression with ARIMA residuals
- Are external regressors durable? Something like a marketing effort or change in policy?

**PyFlux: durable**  
<https://win-vector.com/2024/07/15/arimax-offerings-remain-a-muddle/>

15



We don't have to fully read the equations to work out the implications, instead look at subscripts and difference operators. In our case the external regressors have only the current time-index and no integration or shifted time indices (such as the shock terms).

## Brute force

- Treat as a standard machine learning problem
- Model  $\hat{y}_{t+k} = \hat{E}[y_{t+k}] = f_k(y_t, \dots, y_{t-n})$  for  $k = 1 \dots u$
- Fit for all the  $f_k()$ 
  - Linear methods
  - <https://win-vector.com/2023/05/07/a-time-series-apology/>
- Neural net methods such as LSTM

[https://github.com/WinVector/Examples/blob/main/TimeSeries/nested\\_model\\_example.ipynb](https://github.com/WinVector/Examples/blob/main/TimeSeries/nested_model_example.ipynb)

16

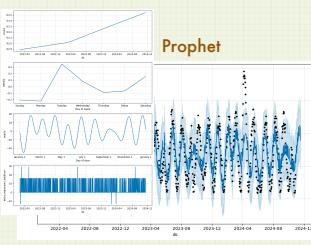
Not actually a bad idea. Misses a lot of the traps in forecasting by ignoring them. Kind of treated as “not allowed” until deep learning had great success with it. Low explainability even with linear methods, as we are a bundle of different models.

## Decomposition or Attribution

- Fit past to things we know future values of:
  - Month of year ...
- Use the combination of the future values of these things as our future prediction.



[https://github.com/WinVector/Examples/blob/main/TimeSeries/Prophet\\_example.ipynb](https://github.com/WinVector/Examples/blob/main/TimeSeries/Prophet_example.ipynb)



17

Fit the past to things we know the future of: such as seasonal averages. Use this fit to project into the future. Very useful in attributing or explaining past effects. Tends to under-perform ARIMA on model quality evaluations—but returns more explainable results and useful data decompositions. The idea is: match past to things we know the future of.

## Hidden State Inference

- Includes methods such as EM and particle filters
- Guess hidden state and parameters  $\beta, \zeta$

$$\text{Estimate } \hat{\beta} = \hat{E}[\beta] = \frac{\int_{\beta, \zeta} d\beta, \zeta P[\text{training data} | \beta, \zeta] \beta}{\int_{\beta, \zeta} d\beta, \zeta P[\text{training data} | \beta, \zeta]}$$

[https://github.com/WinVector/Examples/blob/main/TimeSeries/Stan\\_soln.ipynb](https://github.com/WinVector/Examples/blob/main/TimeSeries/Stan_soln.ipynb)

18

Computationally expensive, but very versatile. This is the method we will pursue in this session.

## Our Example Problem

19

19



## Our Notional Example Problem

20

- We are modeling the number of diners visiting a restaurant each day.
- Improve prediction *and* utility of prediction by bringing in information
  - We have transient external regressors such as events being near our restaurant.
  - We have durable external regressors such as price changes and restaurant reviews.
  - Usually also add a few more easy external regressors to represent seasonality (day of week, month of year, and so on).
- Can also model this as two sub-populations of guests: loyal and transient
  - The issue is: these subpopulations are not labeled in our training data!



20



## As Equations

21

$$\begin{aligned} \text{loyal\_guests}_{\text{date}} &= \beta_{\text{loyal}} + \beta_1 \text{loyal\_guests}_{\text{date}-1} + \beta_2 \text{loyal\_guests}_{\text{date}-2} + \beta_{\text{review}} x_{\text{review}, \text{date}-1} \\ \text{transient\_guests}_{\text{date}} &= \beta_{\text{tmp}} + \beta_{\text{events}} x_{\text{events}, \text{date}} \\ \text{total\_guests}_{\text{date}} &= \text{loyal\_guests}_{\text{date}} + \text{transient\_guests}_{\text{date}} \end{aligned}$$



- These equations are our specification of the problem. Under different assumptions we would write different equations.
  - Subtle point: we are modeling realized values, not the traditional expected values.
  - We only observe **total\_guests** (not **loyal\_guests** or **transient\_guests**).
  - In practice we benefit from a *lot* more external regressors- such as reservations by date.
- Without controllable external regressors forecasting devolves into Cassandra's curse.

21

This immediately requires bi-temporal notation: what was known about date A on date B. We will use “date” and “time” as synonyms in this talk.

## Can Standard Packages Solve This?

22

$$\begin{aligned} \text{loyal\_guests}_{\text{date}} &= \beta_{\text{loyal}} + \beta_1 \text{loyal\_guests}_{\text{date}-1} + \beta_2 \text{loyal\_guests}_{\text{date}-2} + \beta_{\text{review}} x_{\text{review}, \text{date}-1} \\ \text{total\_guests}_{\text{date}} &= \text{loyal\_guests}_{\text{date}} + \beta_{\text{imp}} + \beta_{\text{events}} x_{\text{events}, \text{date}} \end{aligned}$$

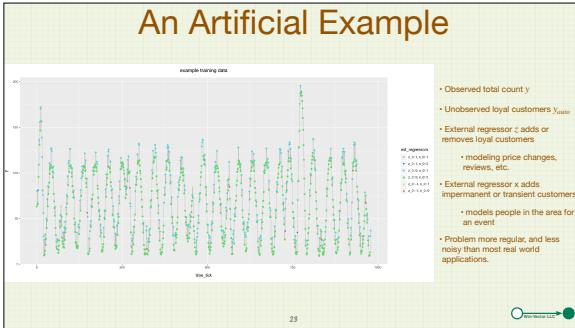
- If we force  $\beta_{\text{review}} = 0$ , then this is “regression with ARIMA residuals” variation of ARIMAX.
- If we force  $\beta_{\text{events}} = 0$ , then this is the sort of system social scientists want to study policy changes (such as tariffs or seatbelt laws). Also called an ARMAX variation (may or may not in fact be).
- None** of the ARIMA style packages we surveyed offered structural *choice* or *control*. The equations the package solves either match yours, or do not.

22



## An Artificial Example

23



23

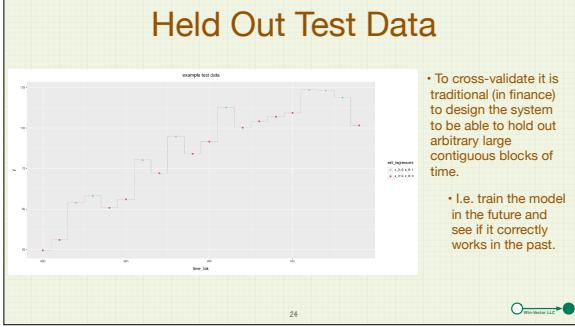


Time series methods tend to be really good at modeling  $\sin(x)$ . So if they are not good at noised up copies of  $\sin(x)$ , what good are they going to be on brutal real world data?

No real world data is ever this easy, so using the methodology does take some engineering (adding more lags, feature engineering and so on).

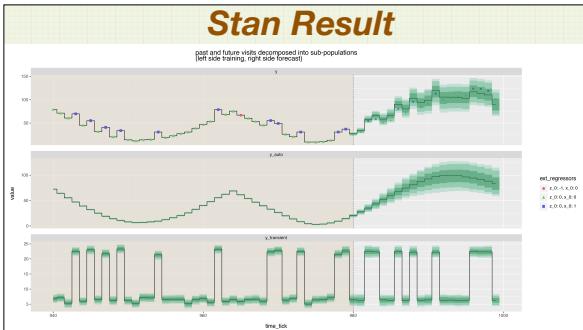
## Held Out Test Data

24



24

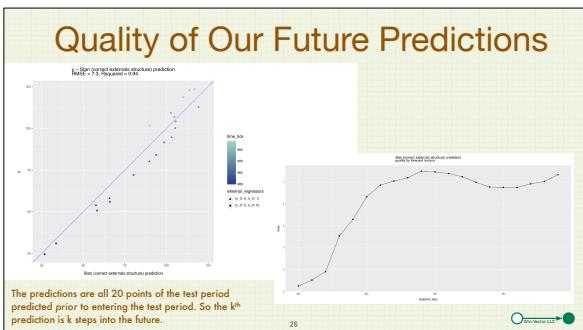




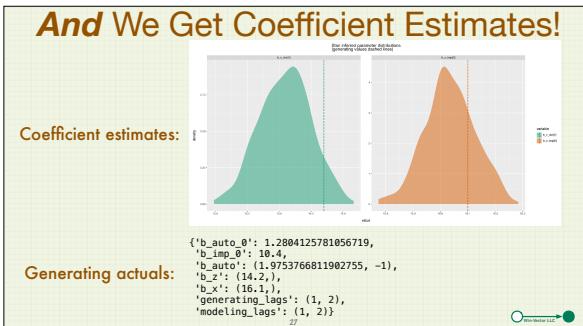
25

Even the historic factorization can be valuable to the business.  $y_{auto}$  is the loyal or durable customers,  $y_{transient}$  are the transient or impulse customers, and  $y$  is the sum of the  $y_{loyal}$  and  $y_{transient}$ . Only  $y$  is observed, and only in the training region.

In time series forecasting, really have to come to accept that the usual sklearn separation of `.fit()` and `.predict()` is not appropriate. Inferring parameters and the future are very related, and insisting on separating the two (while desirable) leads to problems.



26



27

Again even estimating  $b_z$  (the durable external regressor effects) and  $b_x$  (the transient external regressor effects) can be very valuable. The models that don't separate these concerns bias the estimate of at least one of these towards zero. These graphs are not evidence of bias as the distributions are the posterior estimates of the hidden parameters from a single data draw, and we reserve bias to describe a systematic defect seen across multiple data draws.

# How We Solve

- Copy our equations into “Stan”

```

• And “turn the crank.”


    y_auto[3:(N_y_observed + N_y_future)] ~ normal(
      b_auto[0]
      + b_auto[1] * y_auto[2:(N_y_observed + N_y_future - 1)]
      + b_auto[2] * y_auto[1:(N_y_observed + N_y_future - 2)]
      + b_x_imp[1] * x_dur_1[3:(N_y_observed + N_y_future)],
      b_var_y_auto);

    y ~ normal(
      y_auto
      + b_imp_0 + b_x_imp[1] * x_imp_1,
      b_var_y);

```



Complete workbook: [https://github.com/WinVector/Examples/blob/main/TimeSeries/Stan\\_soln.ipynb](https://github.com/WinVector/Examples/blob/main/TimeSeries/Stan_soln.ipynb)

28

28

## What we get from Stan

lp	accept_stat	stepsize	treedepth	n_leapfrog	divergent_	energy_	b_auto_0	b_mp_0	b_auto[0]	y[990]	y[991]	y[1]
0	-3016.91	0.947288	0.07176	9.0	5110	0	35588.0	0.877827	63.9335	-158711	47.5861	117.973
1	-2995.13	0.947288	0.07176	8.0	2550	0	35224.8	0.828451	33.999	-180253	47.5861	117.973
2	-2998.29	0.947288	0.07176	8.0	4690	0	35258.4	0.819555	19.6464	-178463	92.3212	85.8090
3	-3042.34	0.950042	0.07176	9.0	5110	0	35402.0	0.978913	23.7999	-118162	92.3212	85.6033
4	-3044.77	0.950042	0.07176	9.0	5110	0	35408.1	0.783727	91.7220	-149999	41.7989	119.5969
3995	-2378.51	0.980328	0.01044	10.0	10230	0	2925.2	0.30392	58.940	0.04456	49.0182	165.884
3996	-2338.98	0.976501	0.01044	10.0	10230	0	2948.5	0.320605	58.940	-0.000322	-28.0628	60.1195
3997	-2326.53	0.952665	0.01044	10.0	10230	0	2984.1	0.332494	58.2261	-0.000322	23.1797	30.3515
3998	-2326.55	0.948885	0.01044	10.0	10230	0	2973.0	0.337639	57.9308	-0.000329	23.0562	76.7616
3999	-2344.09	0.950051	0.01044	10.0	10230	0	2838.8	0.397008	56.0786	-0.023105	40.2663	88.4096

28

29

Each row is a sample. The columns we are interested in for a given row are the parameters ("b\_\*") and hidden state ("y[\*]", "y\_auto[\*]", "y\_future[\*]"). One thing to keep in mind: the estimated state and parameters in a given row are mutually consistent as they were picked with the plausibility  $(\exp(lp_{\text{--}}) / Z)$ , for some  $Z$  high. It is a domain question if and when we can average these across rows.

# Our Thesis

**In business forecasting: ability to specify actionable structure beats technique in forecasting.**

30

By actionable structure I mean properly structured external regressors that are under our control.

Thank you

All materials: <https://github.com/WinVector/Examples/tree/main/TimeSeries#readme>

§1

