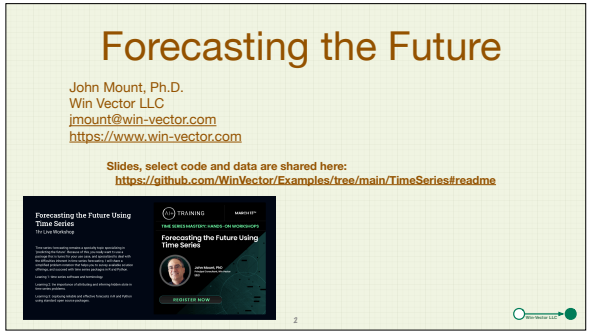
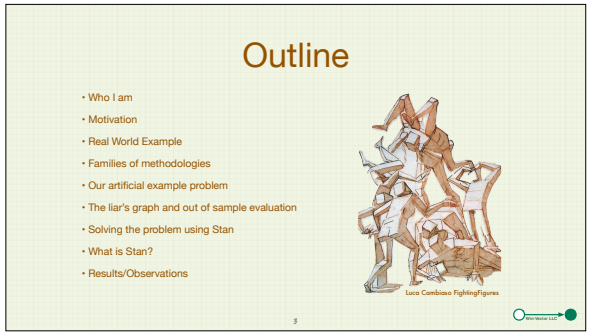

1




2

Hold on this slide until ready to start.



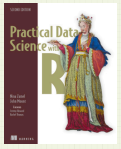

3

We will fill in some details as we go.

# Who I am

4

- John Mount, General Partner at Win Vector LLC.
  - Co-author of *Practical Data Science with R*, 2nd edition, Manning, 2020.
  - Co-author of the `vtreat` R package for re-encoding high cardinality explanatory variables.
- Win Vector LLC is a statistics, machine learning, and data science consultancy and training organization.
  - Specialize in solution design, technology evaluation, and prototyping.
  - We help greatly speed up solution and production deployment.
  - Looking for some more engagements!
    - Please contact: jmount@win-vector.com .
- Please follow us on the Win Vector blog: https://win-vector.com/blog-2/

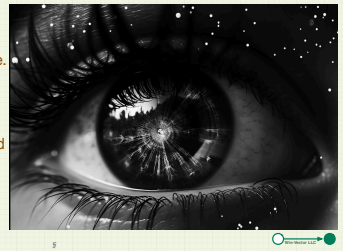



4

# Motivation

5

- For some businesses there is great value in being able to estimate and plan into the future.
- How you attempt this depends on your business.
- I am sharing the *ideas* here, and the Python and R code in the supporting materials.

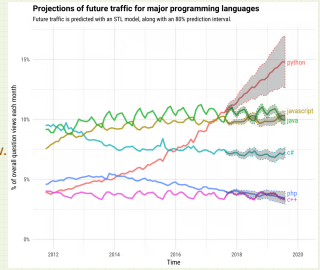


5

# Real World Data Example

6

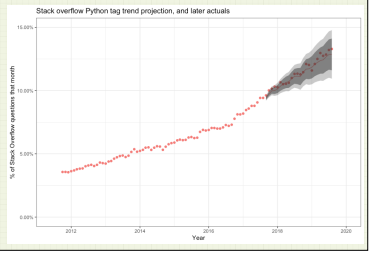
- The Incredible Growth of Python on Stack Overflow
  - https://stackoverflow.blog/2017/09/06/incredible-growth-python/
  - September 2017 article calling out Python's growth and projecting future dominance on Stack Overflow.
  - Allows planning of tools, training, and course material.



## Was it a good prediction?

7

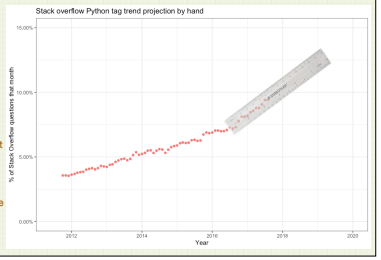
- Yes.
  - The advice was simple and actionable over a 2 year interval.
  - The future data ended up matching the prediction.




## Is this better than projecting by hand?

8

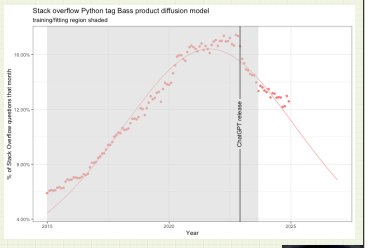
- Yes.
  - Can be automated to great scale.
  - Allows statement and control of modeling assumptions.
    - This case: Seasonal-Trend decomposition using LOESS: "STL".
  - Can measure effectiveness of different modeling assumptions that may have different implied consequences.
  - Can help move towards objectivity (it is no longer obviously your hand on the ruler or scale).




## Is that it?

9

- No!
  - Identifying *which* evolutionary processes are most compatible with observations can be high value.
  - In this case the Bass product diffusion model now seems to fit.
    - This is a bit scary as this model is about modeling total sales of products that are *assumed* to be going obsolete.
  - Same scare graph for Stack Overflow questions on data science topics: https://win-vector.com/2025/03/02/best-before-dates-by-bass/




This is the situation as more data came in. Was not the job of the forecasters to anticipate the ChatGPT introduction.

Laying down a ruler still sees the downward trend, but the model hints at the mechanism driving the decrease.

## de-Motivation

10

"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/

Time series material is overly technical, under taught, and often mis-explained from a tools (not tasks) point of view.


10

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

---

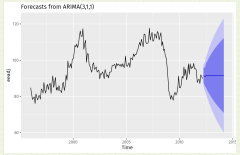
## Example

11

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 notice out that this is **obviously** a low value degenerate fit!


11

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.

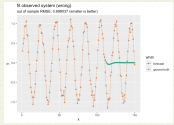
---

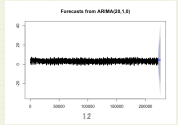
## More Examples of Degenerate Fits

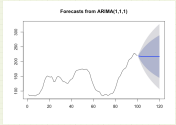
12

- *No* amount of worrying over specialist statistics can replace *looking* at anticipated out of sample performance.
- Under the "you don't learn from success" rubric: there are, *unfortunately*, a *lot* of learning opportunities with ARIMA.

Quickly learn to recognize degenerate fits with more examples:
https://win-vector.com/2023/05/07/a-time-series-apologia/
https://github.com/WinVector/Examples/blob/main/Tides/TideR_ARIMA.md





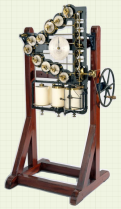




12

## Business Applications

13

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



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

13

## Broad Families of Time Series Methodologies

14

14

## For the Analysis to Add Value

15

- A number of different problem specifications or methods *must* be tried
  - Not just to get better fits
  - But to help identify the dynamics generating the data
  - Some pain in not having a "only try this one best specification"
- Must be able to collect domain expert advice into the model
  - Eliminates both black box and narrow box models
- Must be able to drive actions or useful projections

15

"narrow box models" models that force their structure instead of accepting a user specification.

## Dirty Secrets of Forecasting

16

- Forecasting is *extrapolation*, not interpolation.
  - Much dicier than usual machine learning.
- Things change over time.
  - Built in concept drift. Covid impact on business for example.
- You usually have a very small amount of training data
  - Events are naturally summarized into counts
  - Sub-dividing your data into more time buckets or more categories doesn't really give you more data.
- Many of the models can imitate each other.
  - Somewhat spoils using mere quality of fit on training data to pick among methodologies.
- Usually you want to forecast un-observed quantities such as demand.
  - If you only model observables such as sales you have end up losing a lot of your modeling power to modeling censorship effects (counts not going below zero, counts not going above site capacity, and so on).



## Methodology Families

17

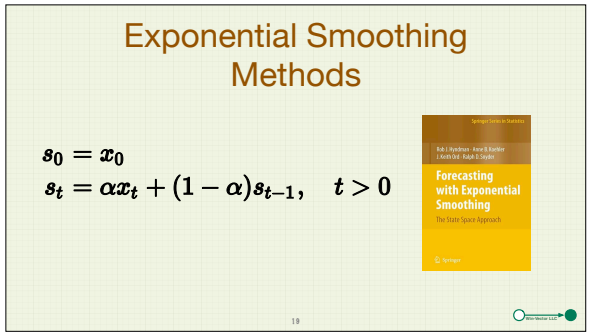
- Exponential Smoothing Methods
- ARIMA derived methods
- Brute force methods
- Decomposition or attribution methods
- Hidden state inference



## Our Opinion of Black Box Methods

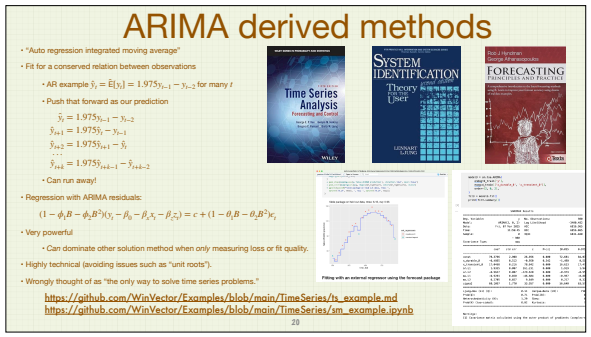
18

- An ideal analyst understands the business and understands the modeling methods.
  - A domain expert may be able to provide good inference without knowing a lot of math.
  - A modeling expert be able to provide good advice without knowing a lot about the domain.
- A "auto ML" style methodology gives possibly close predictions without any understanding of the business or modeling biases.



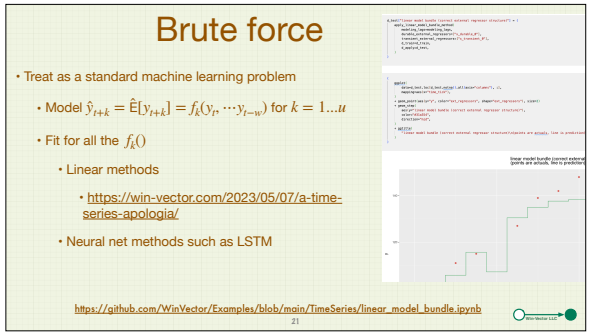

19

Includes Holt-Winters, Kalman filter and many other methodologies.




20

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.



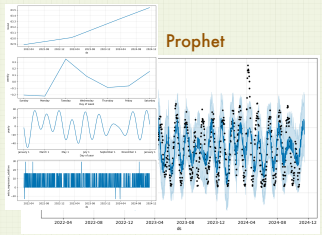

21

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

22

- 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

22

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 evaluation- but returns more explainable results and useful data decompositions.

---

## Prophet Has the Right Ideas

23

- Allow the user to specify the nature of variables
  - Multiplicative versus additive seasonality
  - External regressors as first class citizens
- Prophet is specialized to dates, years, months, and quarters
  - Not quite appropriate for our artificial task
  - Very good for daily business tasks!
- Assume observations are noisy copies of unobserved ideal data!
- Use Bayesian through Stan to do the heavy lifting.

23

---

## Opinion

24

- Good time series and system identification solutions combine a few ideas from a few of these methodological themes.
  - Many of the methodologies simplify *if* you delegate the solution to a systematic solver.
- Most of the solving methods are *excessively* technical and restrictive.
  - There *are* in fact issues with time series forecasting require expert packages.
  - If the package wasn't built for your application, you *may not* be able to adapt it to your application. You are instead forced to adapt your goals to the package.
- A case where there are *more* packages in Python, but more *good* packages in R.


24

Notice that none of the methods are “just a linear regression”

25

## Our Example Problem

25

26

## External Regressors

- For many business applications we need to model future values as a function of past values *and* additional facts 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/hyndsight/arimax/
- To the extent time series modeling is magic, it obscures out the effects of external regressors.

26

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.

27

## Our Notional Example Problem

- We are modeling a restaurant
  - We imagine number of diners on a given day is related to number of guests in the last two days.
    - These are called "lags" or the auto-regressive portion of the model.
    - In practice we would also add most recent same day of week as a lag.
- 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!

27

## As Equations

28

$$\text{loyal\_guests}_{date} = \beta_{loyal} + \beta_1 \text{ loyal\_guests}_{date-1} + \beta_2 \text{ loyal\_guests}_{date-2} + \beta_{review} \cdot x_{review,date-1}$$
$$\text{transient\_guests}_{date} = \beta_{events} \cdot x_{events,date}$$
$$\text{total\_guests}_{date} = \text{loyal\_guests}_{date} + \text{transient\_guests}_{date} + \text{transient\_noise}_{date}$$

- These equations are our specification of the problem. Under different assumptions we would write different equations.
  - Subtle point: we observe realized guests, not underlying ideal demand.
  - 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.

28

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. Also we rapidly run into range censorship (0 and the turn over ability of the restaurant). Stan has methods to deal with this.

---

## The Equations in Detail

29

$$\text{loyal\_guests}_{date} = \beta_{loyal} + \beta_1 \text{ loyal\_guests}_{date-1} + \beta_2 \text{ loyal\_guests}_{date-2} + \beta_{review} \cdot x_{review,date-1}$$
$$\text{transient\_guests}_{date} = \beta_{events} \cdot x_{events,date}$$
$$\text{total\_guests}_{date} = \text{loyal\_guests}_{date} + \text{transient\_guests}_{date} + \text{transient\_noise}_{date}$$

- $x_{review,*}, x_{events,*}$ features engineered by the data scientist.
  - We only know about reviews in the past, but claim to know about events in the future.
- $\text{total\_guests}_{date}$ observed
- $\text{loyal\_guests}_{date}$ unobserved (to be inferred)
  - Classic Bayesian set up. Our population is a mixture of unlabeled sub-populations with different behaviors. Could divide into even more sub-populations if we cared.
- $\beta_{loyal}, \beta_1, \beta_2, \beta_{review}, \beta_{events}$ the linear model coefficients to be inferred
  - We call $\beta_{loyal}, \beta_1, \beta_2$ the “durable” auto-regressive portion of the model.
  - We call $\beta_{review}$ a “durable” effect, associated with a durable external regressor.
  - We call $\beta_{events}$ a “transient” or “impermanent” effect, associated with a transient external regressor.

29

---

## How can we know events in the future?

30

- Buy the information from somebody that curates records about future events.

PredictHQ

(Not an endorsement, I just used them for a couple of clients.
Some notes of mine on problems in using *past* predictions of the future:
https://win-vector.com/2024/09/09/please-version-data/ )

30

## Is Forecasting Really Everything?

31

- No!
  - Even an estimate of loyal_guests / (loyal_guests + transient_guests) can have large business value.
  - Knowing $\beta_{loyal}, \beta_1, \beta_2$ tells one the stability of the business.
  - Knowing $\beta_{events}$ even for only past events allows one to lesson omitted variable bias in fitting the model.

$$\text{loyal\_guests}_{date} = \beta_{loyal} + \beta_1 \text{ loyal\_guests}_{date-1} + \beta_2 \text{ loyal\_guests}_{date-2} + \beta_{review}\, x_{review,date-1}$$
$$\text{transient\_guests}_{date} = \beta_{events}\, x_{events,date}$$
$$\text{total\_guests}_{date} = \text{loyal\_guests}_{date} + \text{transient\_guests}_{date} + \text{transient\_noise}_{date}$$

## Can Standard Packages Solve This?

32

$$\text{loyal\_guests}_{date} = \beta_{loyal} + \beta_1 \text{ loyal\_guests}_{date-1} + \beta_2 \text{ loyal\_guests}_{date-2} + \beta_{review}\, x_{review,date-1}$$
$$\text{transient\_guests}_{date} = \beta_{events}\, x_{events,date}$$
$$\text{total\_guests}_{date} = \text{loyal\_guests}_{date} + \text{transient\_guests}_{date} + \text{transient\_noise}_{date}$$

- If we force $\beta_{review} = 0$, then this is "regression with ARIMA residuals" variation of ARIMAX.
- If we force $\beta_{events} = 0$, 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.

## An Artificial Example

33


- Observed total count $y$
- Unobserved loyal customers $y_{auto}$
- Durable external regressors/ impacts add or remove loyal customers
  - modeling price changes, marketing, reviews, etc.
- Transient regressors add or remove transient customers
  - models people in the area for an event
- Problem more regular, less noisy, and more periods measured than most real world applications.
  - both a hard and easy example in different senses.

Time series methods tend to be really good and 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).

## The Nature of this Artificial Example

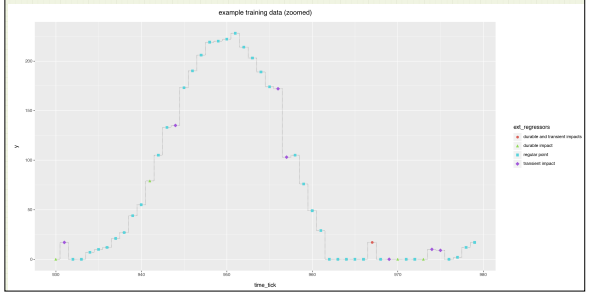
34

- This artificial example is “trouble” in that it leaves in the types of issues that practitioners encounter when they move from textbook data to real world applications.
  - Not fully characterized noise model.
  - Non-negativity of counts.
  - Sections where the near future is not a function of the *observed* near past (coming out of zero regions).
  - No strong seasonality (curve changes phase due to external influences).



## Closer In

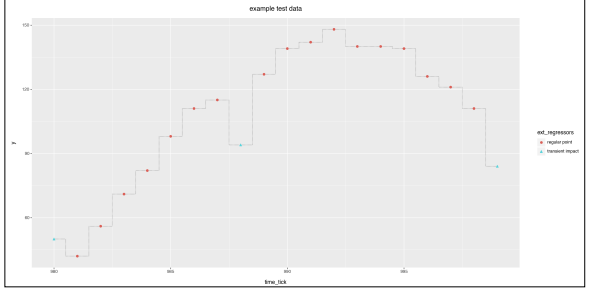
35




b_z is our data generating durable effect coefficient, and b_x is our transient effect coefficient. We generate the data and then see if the system can recover these coefficients from unlabeled data.

## Held Out Test Data

36



# The Liar's Graph and Out of Sample Evaluation

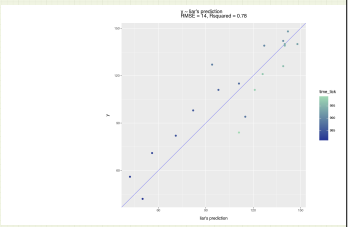
37



## The Liar's Graph

38

- If you take away one thing today, please take away this.
  - Most common scatter plots of "time series performance" plot how well $\hat{y}_t \sim y_t$.
  - This is often "one tick out performance" even if your application requires predicting many time periods out! That is using $\hat{y}_t = y_{t-1}$. Now $|y_t - y_{t-1}|$ tends to be small. So this artificially *looks* like a good prediction, it just isn't implementable k-ticks out!
- Do NOT get conned by the complicated diagnostics and the scatter plot. Insist on seeing the model applied.

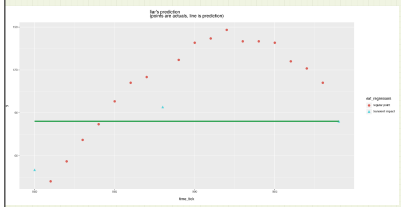


https://github.com/WinVector/Examples/blob/main/TimeSeries/nested_model_example.ipynb



## Look at Out of Sample Predictions!

39



- This and the scatter plot on the previous slide are plotting the performance of the model $\hat{y}_t = y_{t-1}$!
- Once you get out of the training region this is just the horizontal line.
- It is a "good model", just not implantable; as eventually you do not know $y_{t-1}$

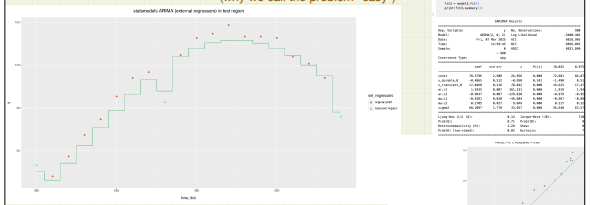
# Solving the Problem



40

# ARIMAX Result

(why we call the problem "easy")





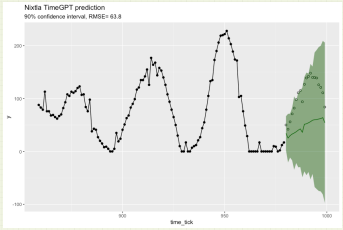
https://github.com/WinVector/Examples/blob/main/TimeSeries/sm_example.ipynb



41

# TimeGPT Result

(why we call the problem hard)

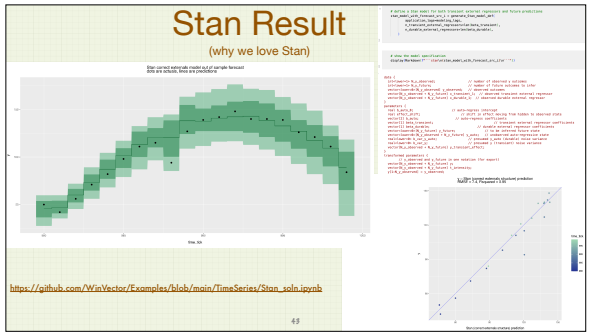



- To be useful the solver has to embody a simple, discussable, set of modeling assumptions.
- The role of time series modeling is not just predicting. It also usually includes system identification.
  - One of the reasons the data scientist has to try many models.
- Doesn't matter "what the LLM/AI knows" if it doesn't succinctly share that information with you or your stakeholders.

https://github.com/WinVector/Examples/blob/main/TimeSeries/other/Nixtla/Nixtla.md



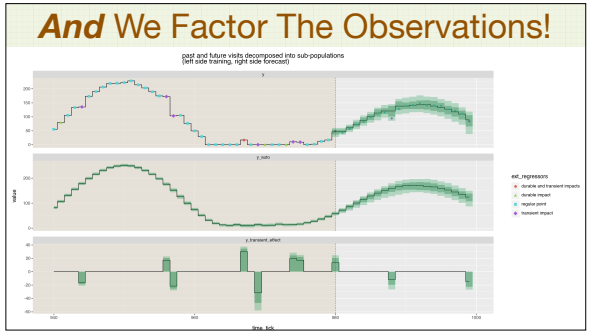
42




43

We get predictions, and they are indeed close the future realized actuals (not seen during training or forecasting). In this, all predictions are made with only information available at time = 979.

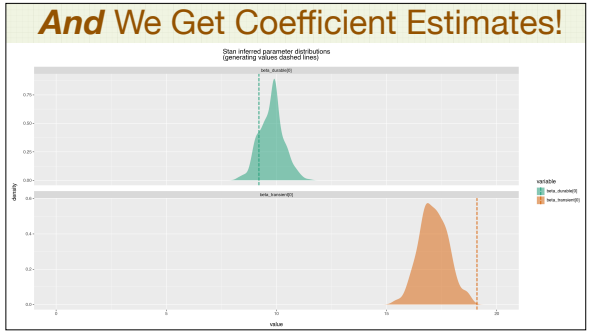
---




44

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.

---



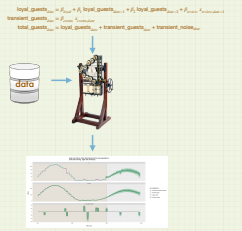

45

The ARIMAX estimates are biased much lower.

# How We Solve

46

- Copy our assumed equations into “Stan”
- And “turn the crank.”




Complete workbook: https://github.com/WinVector/Examples/blob/main/TimeSeries/Stan_soln.ipynb



# What is Stan?

47



# Stan

48

- A Markov Chain Monte Carlo sampler
- Can be used for complicated Bayesian inference
  - Guesses likely values of parameters, and nuisance variables, *and unobserved intermediate state* conditioned on observed data.
  - Returns distributional answers.
  - Subsumes the implementation portion of many other systems/ideas (such as “particle filters” and “importance sampling”).
- Fairly high dependency (requires C++ compiler and linker)
  - May eventually see competition from Torch based alternatives

## What We Did

- Asked Stan to use hidden state methods to solve an ARIMAX *style* formulation of the problem.
- We directly specified how different regressors last over time.
- We directly specified two sub-populations of customers (with different behavior) that we only observe the sum of.
  - We asked Stan to "un-stir" the mixture.



49

## Why it all Works

- We want to infer parameters $\beta$ and hidden state $\zeta$ given data. That is pick $\beta, \zeta$ such that $P[\beta, \zeta \mid data]$ is maximal.
- By Bayes' Law $P[\beta, \zeta \mid data] = P[\beta, \zeta]P[data \mid \beta, \zeta]/P[data]$.
  - Can ignore $P[data]$ as it is free of our parameter estimates.
  - Therefore: picking $\beta, \zeta$ such that $P[\beta, \zeta]P[data \mid \beta, \zeta]$ is large is the same as picking such that $P[\beta, \zeta \mid data]$ is large.
- Structural mis-specification *much* more risky than "wrong priors" when we have a lot of training data.



50

(splash slide, move from quickly)

## The Stan Solution in Detail

```
      // autoregressive system evolution
  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)]
     + beta_durable[1] * x_durable_1[3:(N_y_observed + N_y_future)],
    b_var_y_auto);
      // criticize observations
  for (i in 1:N_y_observed) {
      if (y_observed[i] > 0) {
        target += normal_lpdf(
            y_observed[i] |
            effect_shift + y_transient_effect[i] + y_auto[i],
            b_var_y);
      } else {
        target += normal_lcdf(  // Tobit style scoring, matching above loss
            0 |
            effect_shift + y_transient_effect[i] + y_auto[i],
            b_var_y);
      }
  }
```

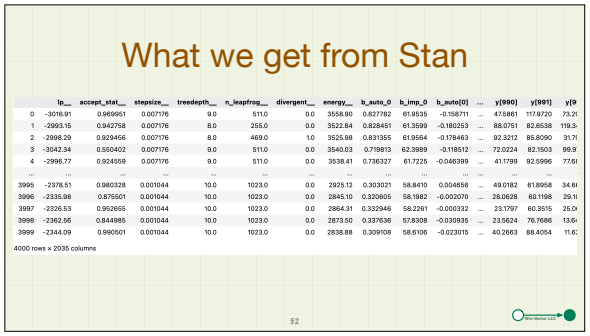
https://github.com/WinVector/Examples/blob/main/TimeSeries/Stan_soln.ipynb



51

(may or may not go into the workbook)

## What we get from Stan



| | lp__ | accept_stat__ | stepsize__ | treedepth__ | n_leapfrog__ | divergent__ | energy__ | b_auto_0 | b_imp_0 | b_auto[0] | ... | y[990] | y[991] | y[9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -3016.91 | 0.969961 | 0.007176 | 9.0 | 511.0 | 0.0 | 3558.90 | 0.827762 | 61.9535 | -0.158711 | ... | 47.5861 | 117.9720 | 73.2 |
| 1 | -2993.15 | 0.942758 | 0.007176 | 8.0 | 255.0 | 0.0 | 3522.84 | 0.828451 | 61.3599 | -0.180253 | ... | 88.0751 | 82.6538 | 119.3 |
| 2 | -2998.29 | 0.929456 | 0.007176 | 8.0 | 469.0 | 1.0 | 3525.98 | 0.831355 | 61.9564 | -0.178463 | ... | 92.3212 | 85.8090 | 31.7 |
| 3 | -3042.34 | 0.550402 | 0.007176 | 9.0 | 511.0 | 0.0 | 3540.03 | 0.719813 | 62.3989 | -0.118512 | ... | 72.0224 | 82.1503 | 99.9 |
| 4 | -2996.77 | 0.924559 | 0.007176 | 9.0 | 511.0 | 0.0 | 3538.41 | 0.736327 | 61.7225 | -0.046399 | ... | 41.1799 | 92.5996 | 77.6 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 3995 | -2378.51 | 0.980328 | 0.001044 | 10.0 | 1023.0 | 0.0 | 2925.12 | 0.303021 | 58.8410 | 0.004656 | ... | 49.0182 | 61.8958 | 34.6 |
| 3996 | -2335.98 | 0.875501 | 0.001044 | 10.0 | 1023.0 | 0.0 | 2845.10 | 0.320605 | 58.1982 | -0.002070 | ... | 28.0628 | 60.1198 | 29.1 |
| 3997 | -2326.53 | 0.952655 | 0.001044 | 10.0 | 1023.0 | 0.0 | 2864.31 | 0.332946 | 58.2261 | -0.000332 | ... | 23.1797 | 60.3515 | 25.0 |
| 3998 | -2362.56 | 0.844985 | 0.001044 | 10.0 | 1023.0 | 0.0 | 2873.50 | 0.337636 | 57.8308 | -0.030935 | ... | 23.5624 | 76.7686 | 13.6 |
| 3999 | -2344.09 | 0.990501 | 0.001044 | 10.0 | 1023.0 | 0.0 | 2838.88 | 0.309108 | 58.6106 | -0.023015 | ... | 40.2663 | 88.4054 | 11.6 |

4000 rows × 2035 columns

52

52

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__) / 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.**

53

53

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

---

## Follow up resources

- This talk
  - https://github.com/WinVector/Examples/tree/main/TimeSeries#readme
- Some of the Perils of Time Series Forecasting
  - https://win-vector.com/2023/05/25/some-of-the-perils-of-time-series-forecasting/
- A Time Series Apologia
  - https://win-vector.com/2023/05/07/a-time-series-apologia/
- Please Version Data
  - https://win-vector.com/2024/09/09/please-version-data/
- The `vtreat` data preparation system
  - https://github.com/WinVector/pyvtreat
  - https://github.com/WinVector/vtreat

54

54

55

# Thank you

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



56

# Part of Our Using Stan to Solve Problems Training Offering

- The series currently includes:
  - Dealing with range censored data, or tobit style regression.
  - Learning rank preferences from observed actions.
  - Time series with external explanatory variables.
- Please contact jmount@win-vector.com for custom data science research, training and consulting.

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



57

# Appendices

(code and additional teaching)

## ARIMAX Solutions

- https://github.com/WinVector/Examples/blob/main/TimeSeries/ts_example.md
- https://github.com/WinVector/Examples/blob/main/TimeSeries/sm_example.ipynb
- https://github.com/WinVector/Examples/blob/main/TimeSeries/ts_example.Rmd



58

## Ad-Hoc Nested Regression Solution

- https://github.com/WinVector/Examples/blob/main/TimeSeries/linear_bundle_fns.py
- https://win-vector.com/2023/05/07/a-time-series-apologia/



59

## A Few Things From Experience

- At first it feels like forecasting software binding fitting and prediction together is a mistake.
  - Violates the *very* useful **`sklearn`** separate **`.fit()`** and **`.predict()`** API by forcing a **`.fit_predict()`** pattern.
  - However, part of forecast inference is to also estimate un-observed details of past state. So fitting is in fact not separate from forecasting.
  - And you may want to re-run with different values of future external regressors under your control.
    - Such as offering a discount on a day that is predicted to have low attendance.
  - Our own **`vtreat`** high cardinality variable re-encoding package uses the same restriction to prevent over fitting.
- Stan great for prototyping
  - Not forced to use full Stan methodology in production.
- Can, in many cases, export inferences from Stan for use by simpler implementations.
  - https://win-vector.com/2019/07/03/replicating-a-linear-model/



60