If you are having the following symptoms at your company when it comes to business KPI forecasting, then maybe you need to look at **automated forecasting:**

* Ugly Excel spreadsheets with multiple tabs and 2000s style pastel formatting
* Business unit managers, store managers, operations managers, sales teams, and finance teams who give convoluted and indirect answers to basic questions about their forecasting methodology
* Too much manual and human intervention giving “guard rails” to the forecasts with no documentation on why they were put in place
* Lack of data science or data analyst personnel to create statistical forecasts
* Executives reaming you and your team on why the forecasts are always inaccurate and why there’s always a long turnaround time to update them

Automated forecasting is the process of automating data wrangling and data preparation of your time series data, splitting the data into training and holdout data, training several different time series models, testing each of those models onto a holdout data set to measure its accuracy, then choosing the most accurate model and re-fitting on the entire data set to create a forecast over a specified time horizon. **This could typically take several steps and hundreds of lines of code, but AutoTS does this type of automated forecasting in a single line of code.**

**What are some examples of forecasting at businesses and enterprises?**

Typically, when companies are creating forecasts, they’re creating forecasts on a time series basis. That is, they are generating daily, weekly, monthly, quarterly or yearly forecasts.

Some examples of forecasting that we’ve seen at Fortune 500 companies and tech startups by industry are:

* **Brick-and-Mortar Retail or B2B Distributors**
  + Weekly Revenue Forecasts at a Branch/Store Level
  + Daily Units Sold by SKU. This is for inventory management and helps to effectively plan for demand for a certain item to reduce stock-out costs and prevent purchasing more supply than there’s demand for.
  + Daily Number of Reward Customer Sign-Ups. If your company has a rewards program, knowing how many are going to sign-up will help you effectively plan for how much “welcome to rewards” print material to procure.
* **ECommerce**
  + Daily Direct Traffic.
  + Daily Organic Traffic
  + Daily Number of Total Sessions
  + Monthly Revenue for Affiliates. This helps plan for how much commission you need to pay to Affiliates per month.
* **Hospitality and Tourism**
  + Daily Revenue Forecasts
  + Daily Occupancy Forecasts. This is the number of rooms the hotel is forecasting will be booked in the future as a percent of total available rooms.
  + Daily Patrons at Restaurant Outlets. Knowing this helps the F&B Managers of these outlets allocate labor efficiently as they’ll staff less during non-busy days and staff more during busy days. This reduces labor costs during non-busy days and ensures no lost revenue due to insufficient staffing during busy days.

**What are some of the current enterprise challenges of forecasting?**

Some of the challenges of enterprise forecasting is doing so in an **automated**, **scalable**, and **unbiased** way. Too many times when creating forecasts, business unit stakeholders create complicated Excel spreadsheets, with lots of tabs and formulas and ugly formatting, using their own individual methodology, and leaving no process for how to update or reverse engineer. Often, when the employee(s) who manages those Excel spreadsheets leave(s) the company, the enterprise use of the forecast stops, and the process has to be re-built from scratch.

So this current process is neither automated (it requires specific personnel to manually update it), scalable (because Excel doesn’t scale, and the forecasts stop as soon as the employee leaves), nor unbiased (as the employee had their own individual methodology to forecast without giving insight into it). Additionally, forecasts at enterprises are generated by non-qualified, non-quantitative personnel with poor Excel skills and likely no coding or statistical background, resulting in forecast errors.

**What is AutoTS?**

**AutoTS stands for automated time series, and it automatically finds and creates the most accurate forecast** from a list of 7 econometric time series models including ARIMA, Holt-Winters, and Autoregressive Neural Networks.

It’s a function inside the [RemixAutoML](https://github.com/AdrianAntico/RemixAutoML) package in the open-source programming language R. R is a popular programming language for data scientists and analysts that is used to build statistical and machine learning models along with data visualizations.

**The beauty of AutoTS and** [**RemixAutoML**](https://github.com/AdrianAntico/RemixAutoML) **is their simplicity and ease of use. Even if you’ve never programmed in R, you can still use AutoTS easily. If you’ve ever used a function inside Excel like sum() or if() formulas, then** [**you can code using AutoTS**](https://www.remyxcourses.com/course?courseid=intro-to-remixautoml-in-r)**.**

The logo of AutoTS is a robot sniper, which symbolizes automation and accuracy.

**How does AutoTS solve these challenges and produce accurate forecasts?**

AutoTS solves the automation problem because it eliminates manual updates of Excel forecast templates and eliminates relying on an employee’s methodology with no oversight. This methodology was likely created by someone with a non-quantitative background, but AutoTS uses best-in-class statistical and machine learning models. So you won’t have to worry about inaccurate forecasts.

AutoTS solves the scalability problem since it’s open source and code-based, and therefore, by its nature, reproducible. It can also be integrated into several popular BI platforms that have R integration, such as Tableau and PowerBI, as well as drag-and-drop analytics platforms like Alteryx.

AutoTS solves the bias problem since it doesn’t rely on human judgement, intuition, or manual intervention. That’s typically what creates error and bad decision-making in the first place. AutoTS is machine learning and statistically based.

AutoTS produces accurate forecasts by running your data through 7 different econometric time series models and choosing the most accurate one that predicts best out-of-sample. Out-of-sample is defined as the holdout data set. Accuracy is defined as lowest mean absolute percentage error (MAPE).

**R Code for Creating Automated Forecasts with AutoTS on the Walmart Weekly Sales Data Set.**

The data set we’re using is [weekly sales by Walmart store from Kaggle](https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data). The R code will do some basic data wrangling to get total sales by week for the highest grossing store, as the raw data set is by week, store, and department. If you have a internal company data set with a metric you want to forecast grouped by day, you can substitute it at Line 34, where “top\_store\_weekly\_sales” is defined. Then change the TimeUnit in AutoTS to “day”.

You can see how few lines of code are needed to create accurate, automated, scalable, and unbiased forecasts using machine learning. No more messy spreadsheets. Technically, AutoTS only uses 1 line of R code, but we dedicated each function argument as its own line just for tutorial presentation purposes.

We drew some inspiration for branding the forecast plot output with RemixAutoML based on Michael Toth’s blog [here](https://michaeltoth.me/you-need-to-start-branding-your-graphs-heres-how-with-ggplot.html).

library(RemixAutoML)

library(data.table)

library(dplyr)

library(magrittr)

library(ggplot2)

library(scales)

library(magick)

library(grid)

# IMPORT DATA FROM REMIX INSTITUTE BOX ACCOUNT ----------

# link to manually download file: <https://remixinstitute.app.box.com/v/walmart-store-sales-data/>

walmart\_store\_sales\_data = data.table::fread("<https://remixinstitute.box.com/shared/static/9kzyttje3kd7l41y1e14to0akwl9vuje.csv>", header = T, stringsAsFactors = FALSE)

# FIND TOP GROSSING STORE (USING dplyr) ---------------------

# group by Store, sum Weekly Sales

top\_grossing\_store = walmart\_store\_sales\_data %>% dplyr::group\_by(., Store) %>%

dplyr::summarize(., Weekly\_Sales = sum(Weekly\_Sales, na.rm = TRUE))

# max Sales of 45 stores

max\_sales = max(top\_grossing\_store$Weekly\_Sales)

# find top grossing store

top\_grossing\_store = top\_grossing\_store %>% dplyr::filter(., Weekly\_Sales == max\_sales)

top\_grossing\_store = top\_grossing\_store$Store %>% as.numeric(.)

# what is the top grossing store?

print(paste("Store Number: ", top\_grossing\_store, sep = ""))

# FIND WEEKLY SALES DATA FOR TOP GROSSING STORE (USING data.table) ----------

top\_store\_weekly\_sales <- walmart\_store\_sales\_data[Store == eval(top\_grossing\_store),

.(Weekly\_Sales = sum(Weekly\_Sales, na.rm = TRUE)),

by = "Date"]

# FORECAST WEEKLY SALES FOR WALMART STORE USING AutoTS ------

# forecast for the next 16 weeks - technically 1 line of code, but

# each argument was dedicated its own line for presentation purposes

weekly\_forecast = RemixAutoML::AutoTS(

data = top\_store\_weekly\_sales,

TargetName = "Weekly\_Sales",

DateName = "Date",

FCPeriods = 16,

HoldOutPeriods = 12,

TimeUnit = "week"

)

# VISUALIZE AutoTS FORECASTS ----------------

# view 16 week forecast

View(weekly\_forecast$Forecast)

# View model evaluation metrics

View(weekly\_forecast$EvaluationMetrics)

# which model won?

print(weekly\_forecast$ChampionModel)

# see ggplot of forecasts

plot = weekly\_forecast$TimeSeriesPlot

#change y-axis to currency

plot = plot + ggplot2::scale\_y\_continuous(labels = scales::dollar)

#RemixAutoML branding. Inspiration here: <https://michaeltoth.me/you-need-to-start-branding-your-graphs-heres-how-with-ggplot.html>

logo = magick::image\_read("<https://www.remixinstitute.com/wp-content/uploads/7b-Cheetah_Charcoal_Inline_No_Sub_No_BG.png>")

plot

grid::grid.raster(logo, x = .73, y = 0.01, just = c('left', 'bottom'), width = 0.25)