**Libraries**

Load the following libraries. For the purposes of this tutorial, I’m setting all plots to static ggplot2 using interactive <- FALSE, but I encourage you to switch this to TRUE to see how easy it is to make interactive plotly plots.

library(tidyverse)

library(lubridate)

library(timetk)

# Setup for the plotly charts (# FALSE returns ggplots)

interactive <- FALSE

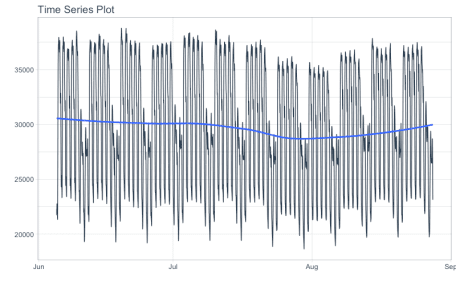
**Part 1: Autocorrelation**

***Autocorrelation*** is the presence of correlation that is connected to lagged versions of a time series. In laymen’s terms, this means that past history is related to future history. We can visualize this relationship with an **ACF plot.**

First, plot the time series we’ll be looking at taylor\_30\_min using plot\_time\_series

taylor\_30\_min %>%

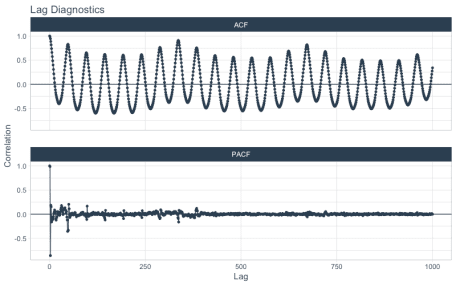
plot\_time\_series(date, value, .interactive = interactive)



This series represents hourly electricity demand taken at 30-min intervals for about 3-months. We can visualize the autocorrelation in the series using a new function, plot\_acf\_diagnostics().

taylor\_30\_min %>%

plot\_acf\_diagnostics(date, value, .interactive = interactive)



**Why are ACF and PACF important?**

From the plot\_acf\_diagnostics() we get:

* **ACF Plot:** The autocorrleation (y-axis), which is the relationship between the series and each progressive lag (x-axis) with the series.
* **PACF Plot:** The partial-autocorrelation vs lags. The Partial Autocorrelation shows how much each progressive ACF adds to the predictability. In other words, lags that are correlated with each other are de-weighted so the most important lags are present.

**These 2 visualizations help us model relationships and develop predictive forecasts**:

* ***Seasonality:*** Possible Fourier Series we can use to model a relationship
* ***Lags as Predictors***: We can find important lags to include in our models

**Grouped ACF and PACFs**

Often in time series we are dealing with more than one series – these are called groups. Let’s switch to a different hourly dataset, m4\_hourly, that contains 4-groups.

m4\_hourly %>%

group\_by(id) %>%

plot\_time\_series(date, value,

.facet\_ncol = 2,

.facet\_scale = "free",

.interactive = interactive)



We can get the ACF and PACF plots easily using plot\_acf\_diagnostics(). We can isolate 14-days of lags using the .lags = "14 days".

m4\_hourly %>%

group\_by(id) %>%

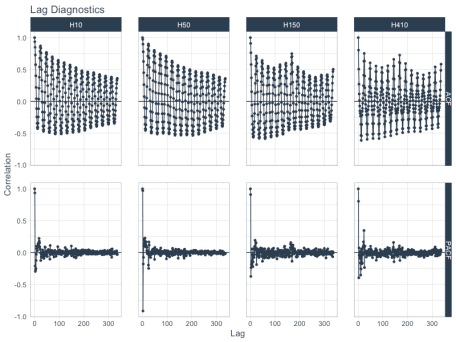
plot\_acf\_diagnostics(

date, value, # ACF & PACF

.lags = "14 days", # 14-Days of hourly lags

.interactive = interactive

)



**Why use time series groups?**

* Using groups helps us to evaluate time series **much faster** than analyzing every time series individually. We’re able to quickly evaluate 4 time series.
* Grouped analysis can highlight **similarities and differences between time series**. We can see H150 and H410 have spikes at 1-week in addition to the daily frequency.

**Part 2: Cross Correlation**

The last example here is **Cross Correlation**, an important technique for finding external predictors. We start with a new time series, walmart\_sales\_weekly, which contains weekly sales for walmart, time series groups consisting of various departments, and several (potential) predictors including temperature and fuel price.

*Note that you will need to the development version of timetk for this functionality until timetk 2.0.1 is released.* You can upgrade using devtools::install\_github("business-science/timetk").

walmart\_sales\_weekly

## # A tibble: 1,001 x 17

## id Store Dept Date Weekly\_Sales IsHoliday Type Size Temperature

##

## 1 1\_1 1 1 2010-02-05 24924. FALSE A 151315 42.3

## 2 1\_1 1 1 2010-02-12 46039. TRUE A 151315 38.5

## 3 1\_1 1 1 2010-02-19 41596. FALSE A 151315 39.9

## 4 1\_1 1 1 2010-02-26 19404. FALSE A 151315 46.6

## 5 1\_1 1 1 2010-03-05 21828. FALSE A 151315 46.5

## 6 1\_1 1 1 2010-03-12 21043. FALSE A 151315 57.8

## 7 1\_1 1 1 2010-03-19 22137. FALSE A 151315 54.6

## 8 1\_1 1 1 2010-03-26 26229. FALSE A 151315 51.4

## 9 1\_1 1 1 2010-04-02 57258. FALSE A 151315 62.3

## 10 1\_1 1 1 2010-04-09 42961. FALSE A 151315 65.9

## # … with 991 more rows, and 8 more variables: Fuel\_Price ,

## # MarkDown1 , MarkDown2 , MarkDown3 , MarkDown4 ,

## # MarkDown5 , CPI , Unemployment

We can visualize Cross Correlations using the .ccf\_vars between Weekly Sales and Temperature and Fuel Price.

walmart\_sales\_weekly %>%

select(id, Date, Weekly\_Sales, Temperature, Fuel\_Price) %>%

group\_by(id) %>%

plot\_acf\_diagnostics(

Date, Weekly\_Sales, # ACF & PACF

.ccf\_vars = c(Temperature, Fuel\_Price), # CCFs

.show\_ccf\_vars\_only = TRUE, # Toggle just CCFs?

.lags = "2 years", # Lags

.interactive = interactive

)

