

Grocery Sales Forecast

Christine Vu¹, Dave Friesen²

12/12/2022

Define Goal

- Apply advanced forecasting techniques to “grocery” sales.
 - Demonstrate data-driven and model-based approaches across product families complimentary with a “typical” grocery store chain.
 - Establish, in practical terms, the feasibility of using one or more forecasting approaches in a real-world scenario to support optimal pricing, inventory management, timely promotion, optimal staffing, and other retain grocery objectives.
-
- **[Descriptive Goal? (18)] [Predictive Goal? (18)]** - Add more specifically to these types of forecasting?
 - **[Forecast Horizon? (20)] [Forecast Use? (21)]** - Add more detail (here or below in forecasting section)?
 - **[Level of Automation? (21, 72)]** - Address here as part of overall goal, or elsewhere?

Get Data

Reference: [<https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=transactions.csv>]
(<https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=transactions.csv>)

```
# Load dataset(s); assumes folder structure with data parallel to src
sales_df <- read.csv("../data/train.csv", header = TRUE)
sales_test_df <- read.csv("../data/test.csv", header = TRUE)
stores_df <- read.csv("../data/stores.csv", header = TRUE)
oil_df <- read.csv("../data/oil.csv", header = TRUE)
events_df <- read.csv("../data/holidays_events.csv", header = TRUE)

# Data validation and understanding, including structure, content, and statistical characteristics covered below
```

Explore & Visualize Series

Univariate Analysis and Preliminary Pre-Processing

```
# e.g., statistical characteristics (including distribution, skewness, outliers)
# +[optionally] review sample observations
univariate(sales_df); head(sales_df, 3); #str(sales_df)
```

```
Summary Univariate Analysis for (sales_df) (3,000,888 observations)
```

	Type	NA%	Blank%	Unique	Freq	Min	Max	Mean	Median	Outlier<	>Outlier	Skewness	nZV	ACF1
id	integer			3000888			3000887		1500444	No	No		N	1.0
date	character			1684	1								N	1.0
store_nbr	integer			54		1	54		28	No	Yes		N	1.0
family	character			33									N	0.8
sales	numeric	31%		379610			124717.0	357.8	11.0	No	Yes	7.4	N	
onpromotion	integer	79%		362			741			No	No	11.2	N	

```
  id    date store_nbr    family sales onpromotion
1  0 2013-01-01         1 AUTOMOTIVE      0          0
2  1 2013-01-01         1  BABY CARE      0          0
3  2 2013-01-01         1   BEAUTY      0          0
```

```
univariate(sales_test_df); head(sales_test_df, 3); #str(sales_test_df)
```

```
Summary Univariate Analysis for (sales_test_df) (28,512 observations)
```

	Type	NA% Blank%	Unique Freq	Min	Max	Mean	Median	Outlier<	>Outlier	Skewness	nZV	ACF1
id	integer		28512	3000888	3029399		3015144	No	No		N	1.0
date	character		16	1							N	1.0
store_nbr	integer		54	1	54		28	No	Yes		N	1.0
family	character		33								N	0.8
onpromotion	integer	55%	212		646			No	Yes	8.5	N	

```
id      date store_nbr family onpromotion
1 3000888 2017-08-16      1 AUTOMOTIVE      0
2 3000889 2017-08-16      1 BABY CARE      0
3 3000890 2017-08-16      1 BEAUTY        2
```

```
univariate(stores_df); head(stores_df, 3); #str(stores_df)
```

```
Summary Univariate Analysis for (stores_df) (54 observations)
```

	Type	NA% Blank%	Unique Freq	Min	Max	Mean	Median	Outlier<	>Outlier	Skewness	nZV	ACF1
store_nbr	integer		54	1	54		28	No	No		N	0.9
city	character		22								N	0.3
state	character		16								N	0.4
type	character		5								N	0.6
cluster	integer		17	1	17		8	No	Yes		N	0.2

```
store_nbr city state type cluster
1      1 Quito Pichincha D      13
2      2 Quito Pichincha D      13
3      3 Quito Pichincha D       8
```

```
univariate(oil_df); head(oil_df, 3); #str(oil_df)
```

```
Summary Univariate Analysis for (oil_df) (1,218 observations)
```

	Type	NA% Blank%	Unique Freq	Min	Max	Mean	Median	Outlier<	>Outlier	Skewness	nZV	ACF1
date	character		1218	1							N	1.0
dcoilwtico	numeric	3	998	26.2	110.6	67.7	53.2	No	Yes	0.3	N	0.9

```
date dcoilwtico
1 2013-01-01      NA
2 2013-01-02      93
3 2013-01-03      93
```

```
univariate(events_df); head(events_df, 3); #str(events_df)
```

```
Summary Univariate Analysis for (events_df) (350 observations)
```

	Type	NA% Blank%	Unique Freq	Min	Max	Mean	Median	Outlier<	>Outlier	Skewness	nZV	ACF1
date	character		312	7							N	1.0
type	character		6								N	0.3
locale	character		3								N	0.3
locale_name	character		24								N	
description	character		103								N	0.2
transferred	character		2								Y	

```
date type locale locale_name description transferred
1 2012-03-02 Holiday Local Manta Fundacion de Manta False
2 2012-04-01 Holiday Regional Cotopaxi Provincializacion de Cotopaxi False
3 2012-04-12 Holiday Local Cuenca Fundacion de Cuenca False
```

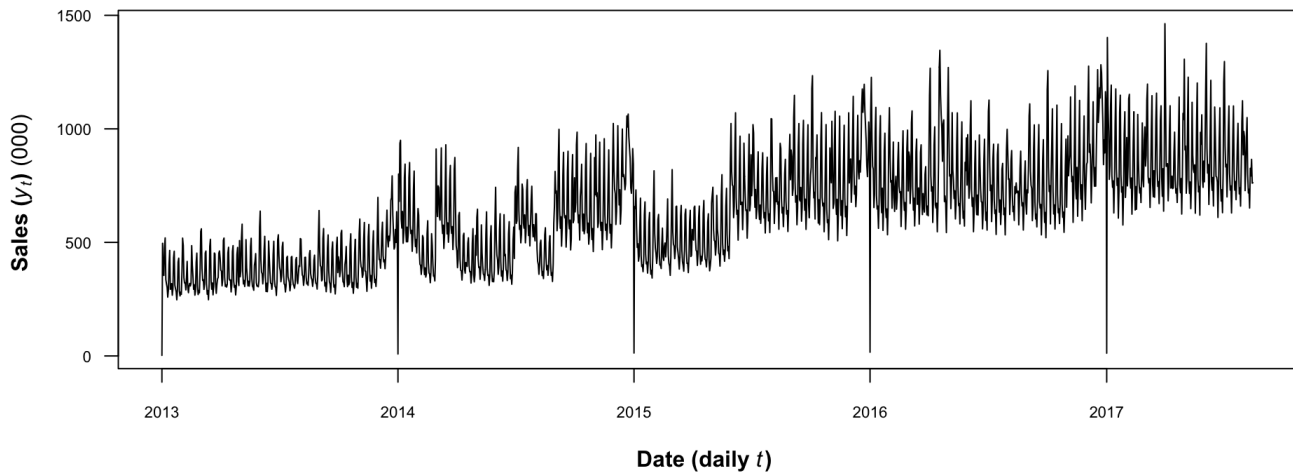
- __[commentary on data usage - e.g., test data as provided adds no value?]

```
# Convert string mm/dd/yyyy to Date values and confirm sort
sales_df <- (sales_df %>%
  mutate(date = as.Date(date, format = "%Y-%m-%d"),) %>%
  arrange(date))
```

Series Visualization - All Product Families

```
# Aggregate base dataframe by date (i.e., sum all product lines) and initially plot series at
# provided time granularity
sales_agg_df <- as.data.frame(sales_df %>%
                              group_by(date) %>%
                              summarize(sales = sum(sales / 1000.0)))
plot(x = sales_agg_df$date, y = sales_agg_df$sales, type = "l",
     main = "Store Sales for All Product Families | Daily | All Dates",
     xlab = TeX(r"\textbf{Date (daily \textit{\$t\$})} (000) )"), ylab = TeX(r"\textbf{Sales (\textit{\$y_t\$})} (000) )"),
     las = 1, cex.axis = 0.7)
```

Store Sales for All Product Families | Daily | All Dates



```
# Aggregate base dataframe from daily to weekly for all product families
sales_wk_df <- as.data.frame(sales_df %>%
                              mutate(year = year(date), week = week(date)) %>%
                              group_by(year, week) %>%
                              summarize(sales = sum(sales / 1000.0)))

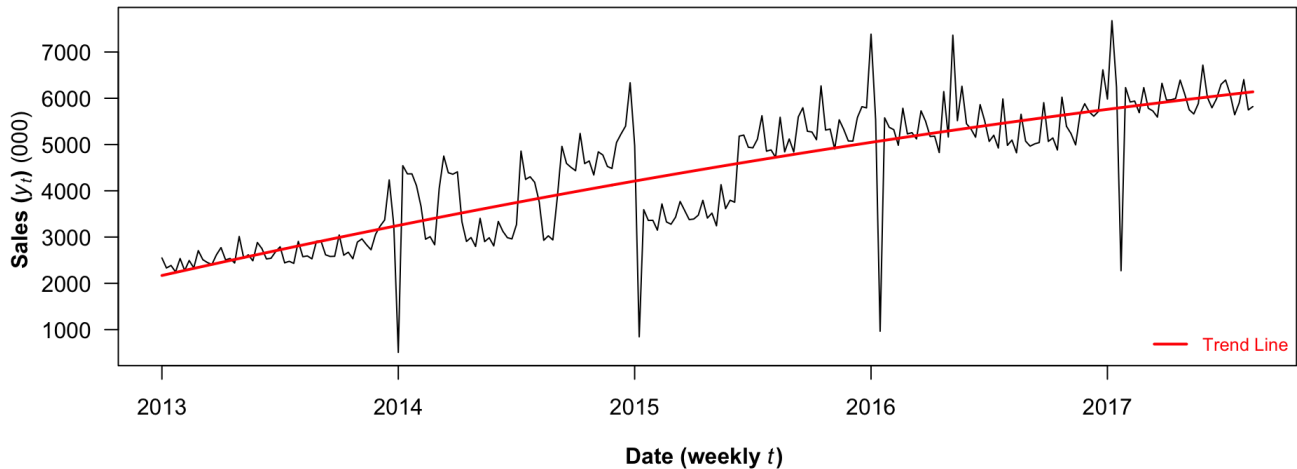
# Create overall time series
sales_begin_year <- head(sales_wk_df$year, 1)
sales_begin_week <- head(sales_wk_df$week, 1)
sales_end_year <- tail(sales_wk_df$year, 1)
sales_end_week <- tail(sales_wk_df$week, 1)
sales_ts <- ts(sales_wk_df$sales,
               start = c(sales_begin_year, sales_begin_week),
               end = c(sales_end_year, sales_end_week), freq = 52)

# Plot overall time series with trend line
plot(sales_ts, type = "l",
     main = "Store Sales for All Product Families | Weekly | All Dates",
     xlab = TeX(r"\textbf{Date (weekly \textit{\$t\$})} (000) )"), ylab = TeX(r"\textbf{Sales (\textit{\$y_t\$})} (000) )"),
     las = 1, cex.axis = 0.7)

sales_lm <- tslm(sales_ts ~ trend + I(trend^2))
lines(sales_lm$fitted, lwd = 2, lty = 1, col = "red")

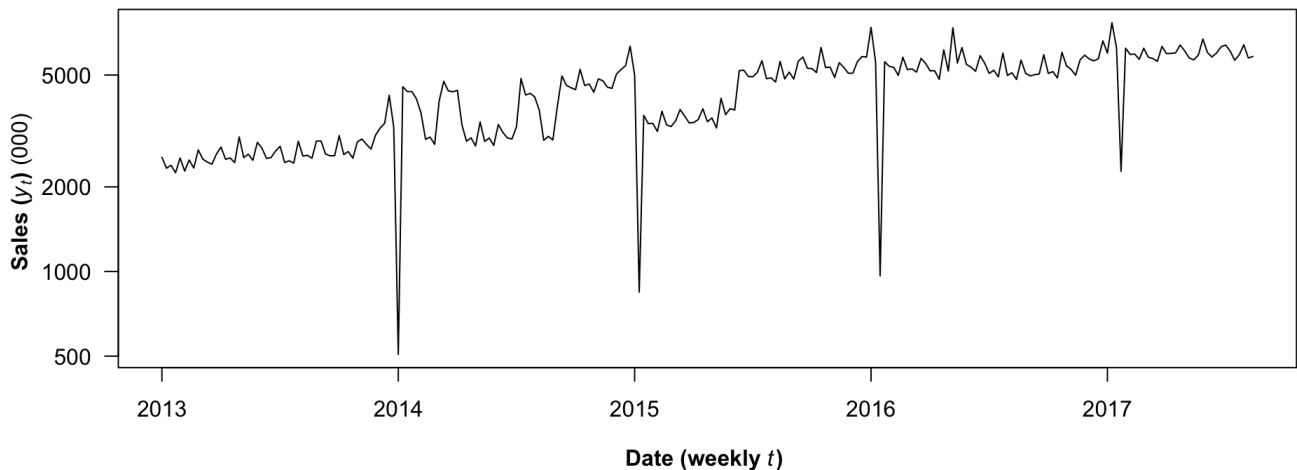
legend("bottomright",
      legend = c("Trend Line"),
      col = c("red"),
      lwd = 2, lty = 1.2, cex = 0.8,
      box.lty = 0, bg = NULL, text.col = "red")
```

Store Sales for All Product Families | Weekly | All Dates



```
# Plot overall time series w/log scale
plot(sales_ts, type = "l",
     main = "Store Sales for All Product Families | Weekly | All Dates | Log Scale",
     xlab = TeX(r"(\textbf{Date (weekly \textit{\$t\$})} )"), ylab = TeX(r"(\textbf{Sales (\textit{\$y_t\$})} (000) )"),
     las = 1, cex.axis = 0.7,
     log = "y")
```

Store Sales for All Product Families | Weekly | All Dates | Log Scale



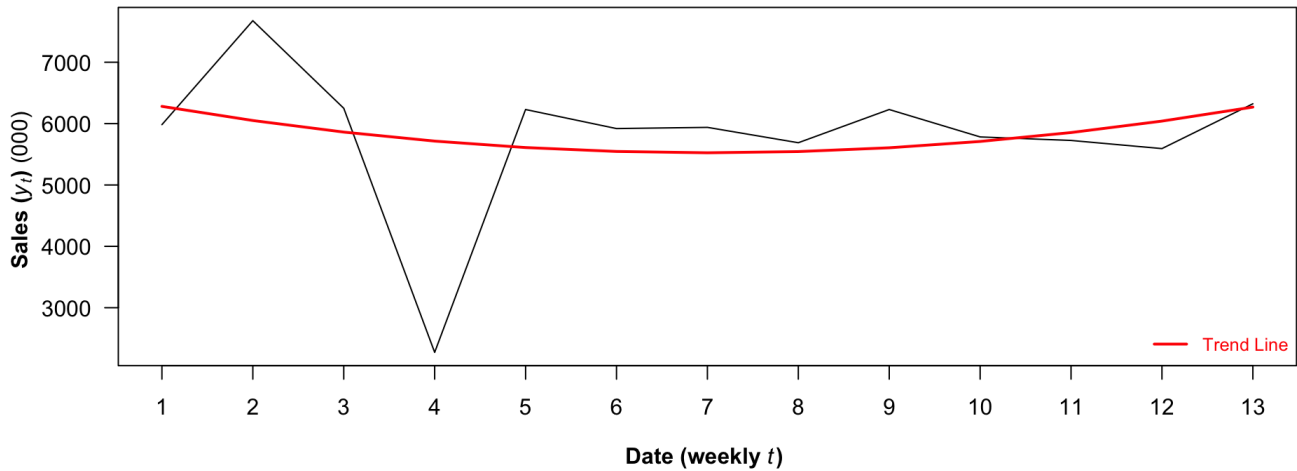
```
# Plot zoomed time series with trend line
sales_zoom_ts <- window(sales_ts, start = c(sales_end_year, 1), end = c(sales_end_year, 13))
plot(sales_zoom_ts, type = "l",
     main = "Store Sales for All Product Families | Weekly | One Quarter",
     xlab = TeX(r"(\textbf{Date (weekly \textit{\$t\$})} )"), ylab = TeX(r"(\textbf{Sales (\textit{\$y_t\$})} (000) )"),
     xaxt = "n",
     las = 1, cex = 0.7)

sales_zoom_lm <- tslm(sales_zoom_ts ~ trend + I(trend^2))
lines(sales_zoom_lm$fitted, lwd = 2, lty = 1, col = "red")

axis(1, at = as.numeric(time(sales_zoom_ts)), labels = seq(sales_zoom_ts))

legend("bottomright",
     legend = "Trend Line",
     col = "red",
     lwd = 2, lty = 1.2, cex = 0.8,
     box.lty = 0, bg = NULL, text.col = "red")
```

Store Sales for All Product Families | Weekly | One Quarter



Series Visualization - Individual Product Families

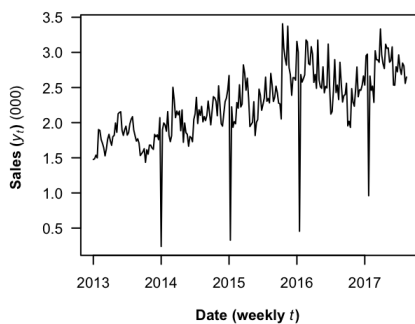
```
# Aggregate base dataframe from daily to weekly by product family
sales_wk_df <- as.data.frame(sales_df %>%
  mutate(year = year(date), week = week(date)) %>%
  group_by(family, year, week) %>%
  summarize(sales = sum(sales / 1000.0)))

opar = par()
par(mfrow = c(1, 3))

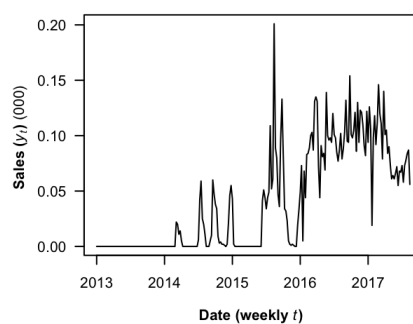
for (f in unique(sales_wk_df$family)) {
  # Subset data by product family and create time series
  df <- filter(sales_wk_df, family == f)
  df_ts <- ts(df$sales,
    start = c(sales_begin_year, sales_begin_week),
    end = c(sales_end_year, sales_end_week), freq = 52)

  # Plot time series
  plot(df_ts, type = "l",
    main = paste("Store Sales for ", f, " | All Dates", sep = ""),
    height = 0.8,
    xlab = TeX(r"(\textbf{Date (weekly \textit{\$t\$})} )"), ylab = TeX(r"(\textbf{Sales (\textit{\$y_t\$})} (000) )"),
    las = 1, cex.axis = 0.7)
}
```

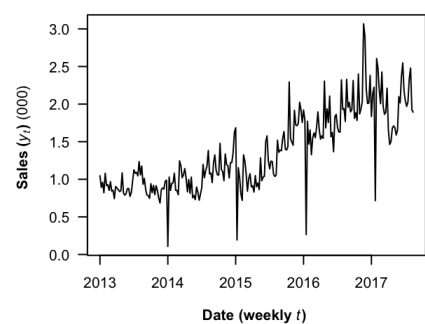
Store Sales for AUTOMOTIVE | All Dates



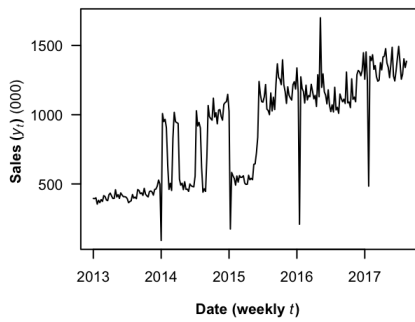
Store Sales for BABY CARE | All Dates



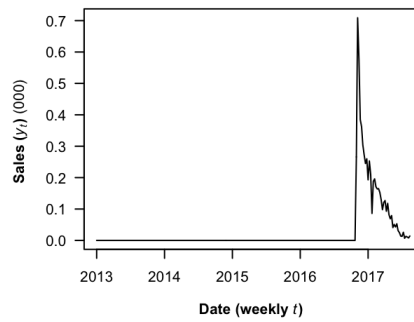
Store Sales for BEAUTY | All Dates



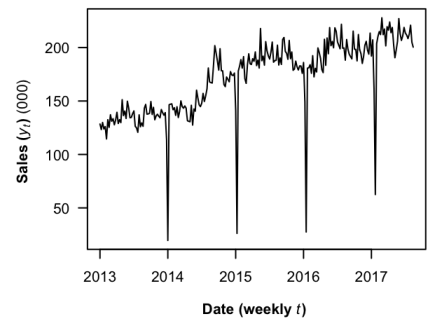
Store Sales for BEVERAGES | All Dates



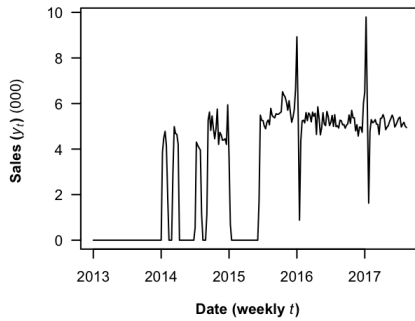
Store Sales for BOOKS | All Dates



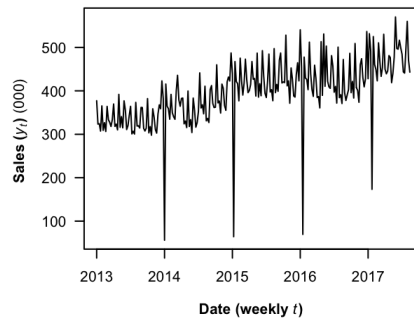
Store Sales for BREAD/BAKERY | All Dates



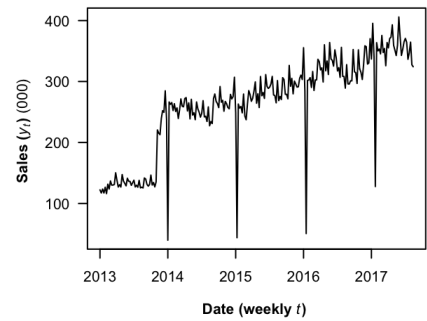
Store Sales for CELEBRATION | All Dates



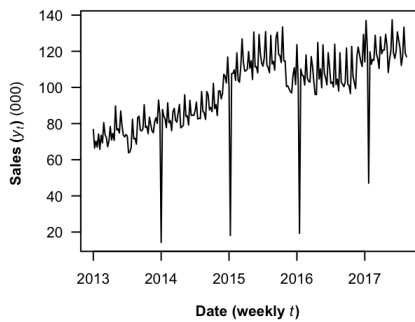
Store Sales for CLEANING | All Dates



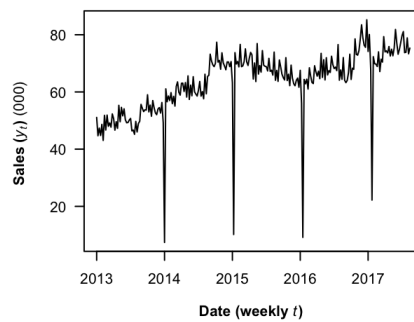
Store Sales for DAIRY | All Dates



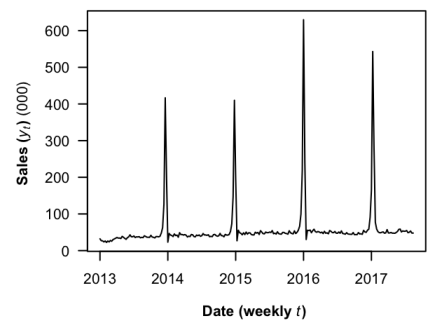
Store Sales for DELI | All Dates



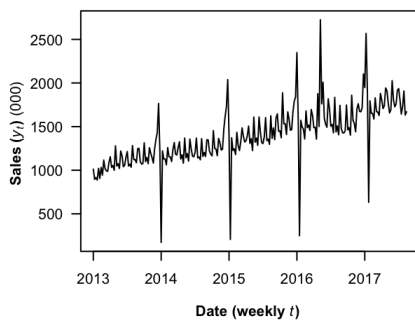
Store Sales for EGGS | All Dates



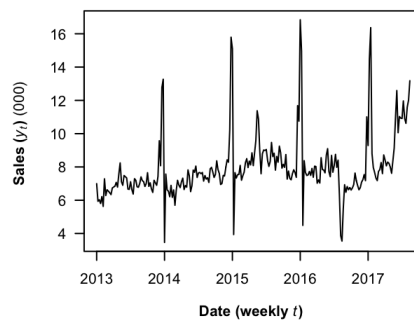
Store Sales for FROZEN FOODS | All Dates



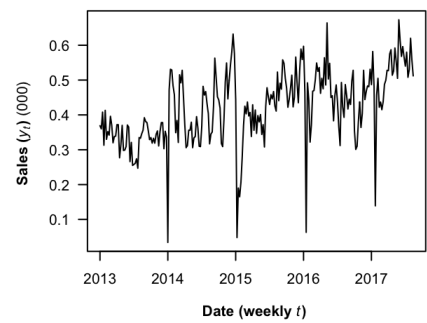
Store Sales for GROCERY I | All Dates



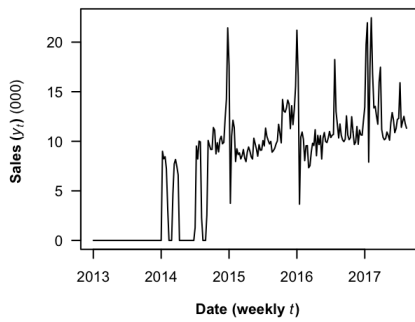
Store Sales for GROCERY II | All Dates



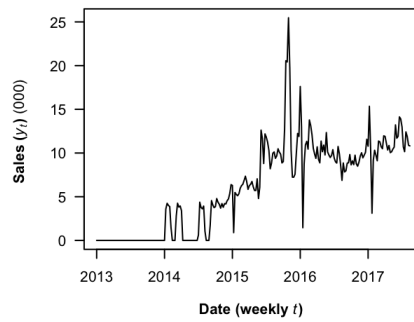
Store Sales for HARDWARE | All Dates



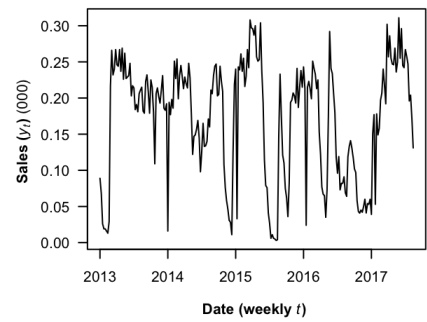
Store Sales for HOME AND KITCHEN I | All Dates



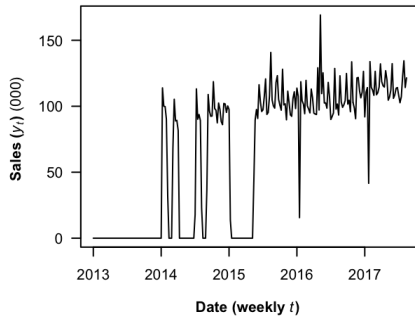
Store Sales for HOME AND KITCHEN II | All Dates



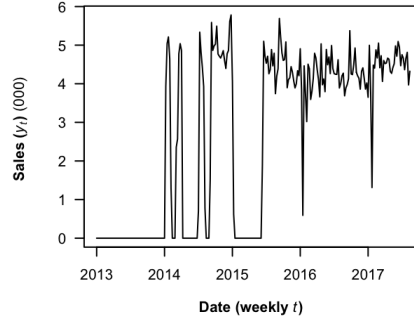
Store Sales for HOME APPLIANCES | All Dates



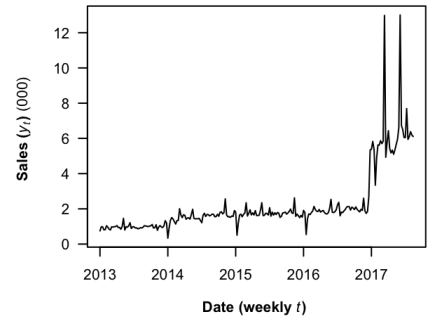
Store Sales for HOME CARE | All Dates



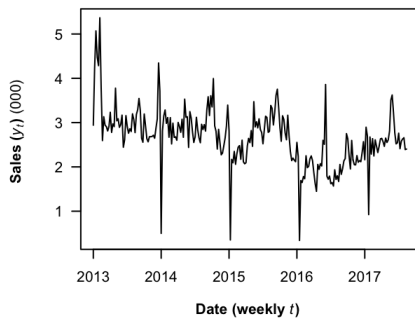
Store Sales for LADIESWEAR | All Dates



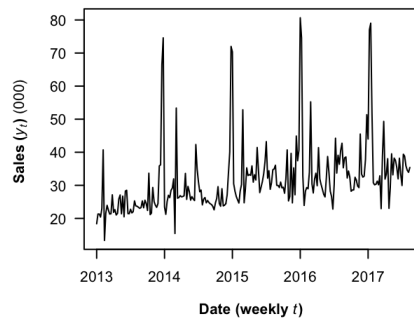
Store Sales for LAWN AND GARDEN | All Dates



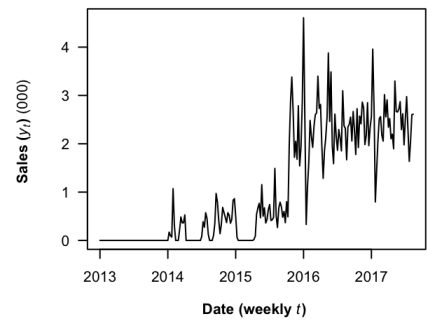
Store Sales for LINGERIE | All Dates



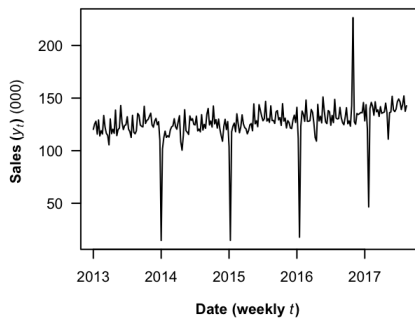
Store Sales for LIQUOR,WINE,BEER | All Dates



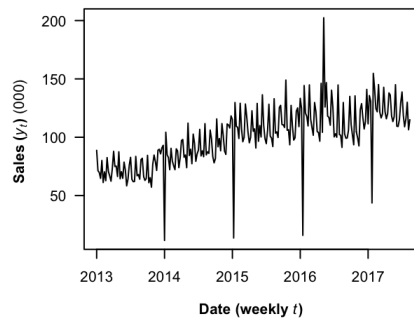
Store Sales for MAGAZINES | All Dates



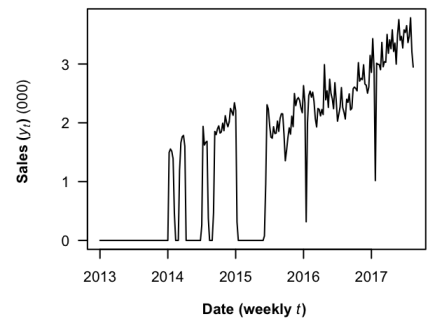
Store Sales for MEATS | All Dates



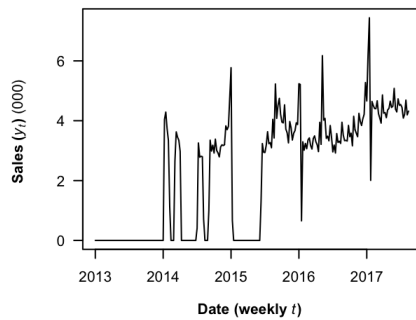
Store Sales for PERSONAL CARE | All Dates



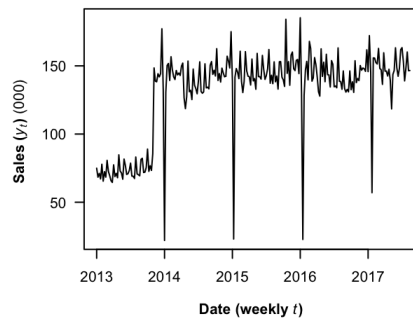
Store Sales for PET SUPPLIES | All Dates



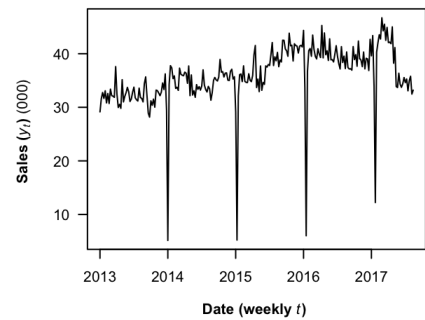
Store Sales for PLAYERS AND ELECTRONICS | All D



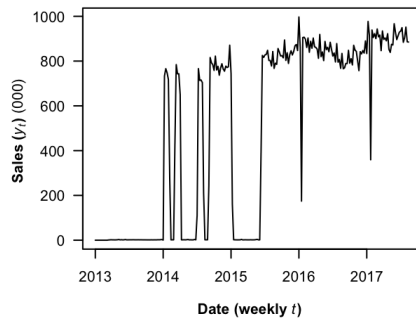
Store Sales for POULTRY | All Dates



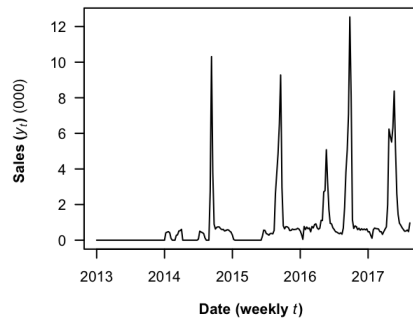
Store Sales for PREPARED FOODS | All Dates



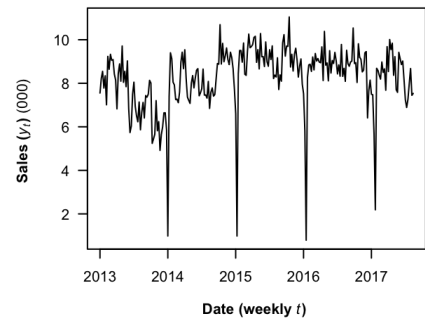
Store Sales for PRODUCE | All Dates



Store Sales for SCHOOL AND OFFICE SUPPLIES | All



Store Sales for SEAFOOD | All Dates



```
par(opar)
```

- **[Visualizing? (30)] [Zooming In? (30)]** - The above figures visualize the provided grocery store sales series across different frequencies, grains, and windows, with highlighted trend lines.
- **[Temporal Frequency? (26)]** - Series data is provided at a daily frequency. However, for purposes of forecasting to project goals, we are choosing to aggregate to a lower frequency of weekly?
- **[Series Granularity? (27)]** - Series grain includes store numbers and individual product families. Grocery chain sub-goals may have different requirements for this grain. For example, optimal pricing, inventory management, and timely promotion may benefit from product family detail, while optimal staffing may benefit from store number. For the purposes of this project, we are focusing on overall forecasting - i.e., aggregated above individual store and product family - knowing that patterns appear similar and project work can be taken to a new level of detail in the future?
- **[Series Granularity / Product Families]** - The above analysis highlights product families or categories that appear *adjacent* - i.e., not "core" - to the grocery business. Some of these families, e.g., books, celebration, or home appliances were introduced at different times and may be "opportunistic" or pilot families. Given this, the project is removing a subset of these types of product families in favor of basic grocery families. (*Does this also get to [domain expertise? (27)]?*)
- **[Time Series Components? (28)]** - Store sales at daily and weekly frequencies, across and within individual product families, exhibit systematic components of [average] level (by default), trend (changes from one period to the next), and [multiplicative?] seasonality, or cyclical behavior. Of course, all include non-systematic "noise."

Pre-Process Data

- **[Data Quality? (25)]** - *Not seeing major issues?*
- **[Missing Values? (39)]** - *Not seeing major issues?*
- **[Unequally Spaced Series? (40)]** - *Not seeing major issues?*
- **[Extreme Values? (40)]** - *Not seeing major issues?*


```

# Placeholder code to remove rows with NAs
#df %>%
# na.omit()

# Placeholder code to remove rows with NAs in specific column
#df %>%
# filter(!is.na(column_name))

# Placeholder code to remove duplicates
#df %>%
# distinct()

# Placeholder code to remove rows by index position
#df %>%
# filter(!row_number() %in% c(1, 2, 4))

# Remove rows not considered "core"
keep_families <- c("BEAUTY", "BEVERAGES", "BREAD/BAKERY", "CLEANING", "DAIRY",
                  "DELI", "EGGS", "FROZEN FOODS", "GROCERY", "GROCERY II", "HARDWARE",
                  "LIQUOR,WINE,BEER", "MEATS", "PERSONAL CARE", "PET SUPPLIES", "POULTRY",
                  "PREPARED FOODS", "PRODUCE", "SEAFOOD")
remove_families <- c("AUTOMOTIVE", "BABY CARE", "BOOKS", "CELEBRATION", "HOME APPLIANCES",
                    "HOME AND KITCHEN I", "HOME AND KITCHEN II", "HOME CARE", "LADIESWEAR",
                    "LAWN AND GARDEN", "LINGERIE", "MAGAZINES", "PET SUPPLIES",
                    "PLAYERS AND ELECTRONICS", "SCHOOL AND OFFICE SUPPLIES")
sales_core_df <- (sales_df %>%
                  filter(family %in% keep_families))

```

Series Visualization - Select ("Core") Product Families

```

# Re-aggregate base dataframe from daily to weekly for all product families
sales_wk_df <- as.data.frame(sales_core_df %>%
                             mutate(year = year(date), week = week(date)) %>%
                             group_by(year, week) %>%
                             summarize(sales = sum(sales / 1000.0)))

# Re-create overall time series
sales_begin_year <- head(sales_wk_df$year, 1)
sales_begin_week <- head(sales_wk_df$week, 1)
sales_end_year <- tail(sales_wk_df$year, 1)
sales_end_week <- tail(sales_wk_df$week, 1)
sales_ts <- ts(sales_wk_df$sales,
               start = c(sales_begin_year, sales_begin_week),
               end = c(sales_end_year, sales_end_week), freq = 52)

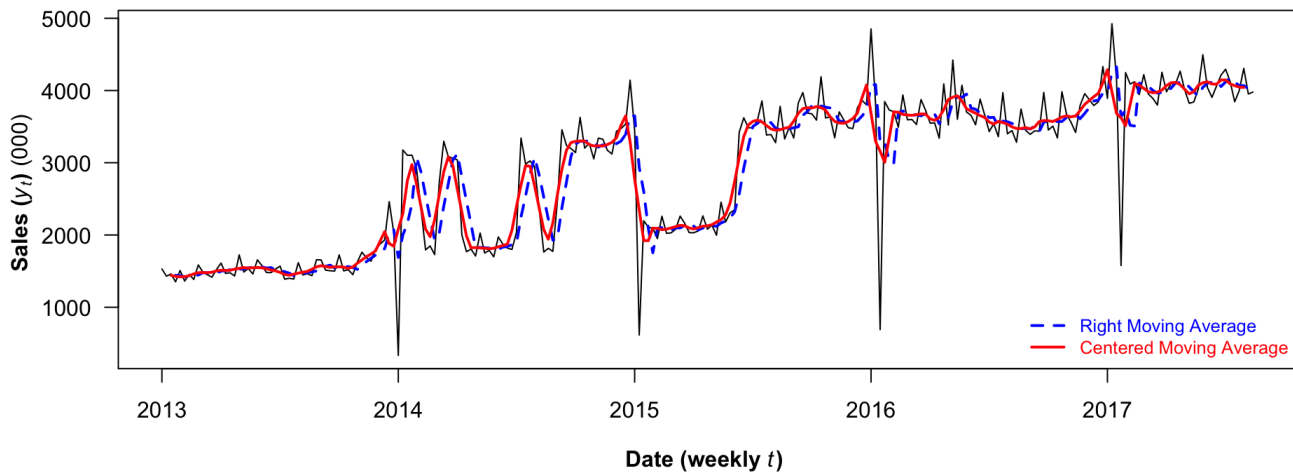
# Plot overall time series with moving averages
plot(sales_ts, type = "l",
     main = "Store Sales for Select Product Families | Weekly | All Dates",
     xlab = TeX(r"\textbf{Date (weekly \textit{\$t\$})} )"), ylab = TeX(r"\textbf{Sales (\textit{\$y_t\$})} (000) )"),
     las = 1, cex.axis = 0.7)

sales_l_ma <- rollmean(sales_ts, k = 4, align = "right")
sales_c_ma <- ma(sales_ts, order = 4)
lines(sales_l_ma, lwd = 2, lty = 2, col = "blue")
lines(sales_c_ma, lwd = 2, lty = 1, col = "red")

legend("bottomright",
      legend = c("Right Moving Average", "Centered Moving Average"),
      col = c("blue", "red"),
      lwd = 2, lty = c(2, 1), cex = 0.8,
      box.lty = 0, bg = NULL, text.col = c("blue", "red"))

```

Store Sales for Select Product Families | Weekly | All Dates



- **[Centered Moving Average Smoothing Method? (80)] [Choosing Window Width? (83)]** - The above figure provides a view of the overall time series, aggregated to weekly, for select “core grocery chain” product families. The centered and trailing moving average lines are illustrative and not intended for forecasting given trend and seasonality components. Note also that these use a window width of quarterly, again, for smoothing visualization (series understanding) only.

Partition Series

- **[Temporal Data Partitioning? (45-46)] [Choosing the Validation Period (48)] [Fixed Partition? (65)] [choice of time span? (41)]** - Given an overall series span of over five years, the series is temporally partitioned with a fixed validation period of one year. The balance of series data is segmented for training (with the full series span appearing relevant to future forecasts). The one year validation span was chosen based in part on the trend and seasonal natures of the grocery business, as previously highlighted in preliminary analysis, and the possibility need for a forecast horizon of up to one year.

```
# Use one year (52 weeks) as validation period (representative set of quarters, seasons)
sales_n_valid <- 52
sales_n_train <- length(sales_ts) - sales_n_valid

# Split data into training and validation periods
sales_train_ts <- window(sales_ts, start = c(sales_begin_year, 1), end = c(sales_begin_year, sales_n_train))
sales_valid_ts <- window(sales_ts, start = c(sales_begin_year, sales_n_train + 1), end = c(sales_begin_year, sales_n_train +
sales_n_valid))

# Set x axis values based on ts ranges
x_train <- sales_begin_year
x_valid <- sales_begin_year + ((sales_n_train + 1) / 52)
x_future <- x_valid + ((sales_n_valid + 1) / 52)
```

Apply Forecasting Method(s)

- **[Model-Based vs. Data-Driven (69)]** -

Naive Forecast Baseline

- [Trailing Moving Average Smoothing Method? (81)] -
- [Choosing Window Width (83)] -
- [Differencing? (85)] -
- [Removing Trend? (85)] -
- [Removing Seasonality? (87)] -
- [Removing Trend and Seasonality? (87)] -
- [Simple Exponential Smoothing? (87)] -
- [Advanced Exponential Smoothing? (90)] -
- [Measuring Predictive Accuracy (51++)] -

```
# Fit Holt-Winter's model with (M)ultiplicative error, (A)dditive trend, and (M)ultiplicative seasonality
# for last year of training series
sales_hw <- ets(window(sales_train_ts, start = c(sales_begin_year, sales_n_train + 1 - sales_n_valid),
                    end = c(sales_begin_year, sales_n_train + 1)),
               model = "MAM")

# Forecast (predict) for validation period
sales_hw_pred <- forecast(sales_hw, h = length(sales_valid_ts), level = 0)

# Summarize forecast (validation) performance
accuracy(sales_hw_pred, sales_valid_ts)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-39	523	258	-9	14	NaN	-0.002	NA
Test set	578	781	673	13	18	NaN	0.334	0.8

```
# Plot Holt-Winter's model
plot(sales_hw_pred, type = "l",
     main = "Store Sales for Select Product Families | Holt-Winter's Forecaster",
     xlab = TeX(r"(\textbf{Date (weekly \textit{\$t\$})} )"), ylab = TeX(r"(\textbf{Sales (\textit{\$y_t\$})} (000) )"),
     las = 1, cex.axis = 0.7,
     xlim = c(x_train, xmax), ylim = c(0, ymax))

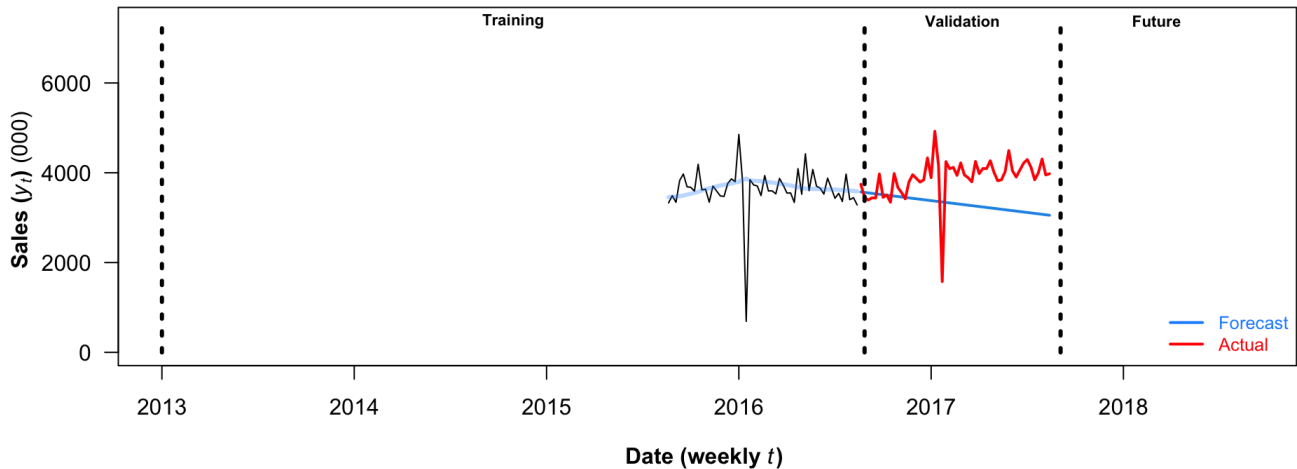
# Add additional lines
lines(sales_hw$fitted, lwd = 3, col = adjustcolor("dodgerblue", alpha.f = 0.3))
lines(sales_valid_ts, lwd = 2, col = "red")

# Add plot definitions
lines(c(x_train, x_train), c(0, ymax), lwd = 3, lty = 3)
lines(c(x_valid, x_valid), c(0, ymax), lwd = 3, lty = 3)
lines(c(x_future, x_future), c(0, ymax), lwd = 3, lty = 3)

text(x_train + ((x_valid - x_train) * 0.5), ymax, "Training", font = 2, cex = 0.7)
text(x_valid + ((x_future - x_valid) * 0.5), ymax, "Validation", font = 2, cex = 0.7)
text(x_future + 0.5, ymax, "Future", font = 2, cex = 0.7)

legend("bottomright",
      legend = c("Forecast", "Actual"),
      col = c("dodgerblue", "red"),
      lwd = 2, lty = 1, cex = 0.8,
      box.lty = 0, bg = NULL, text.col = c("dodgerblue", "red"))
```

Store Sales for Select Product Families | Holt-Winter's Forecaster



Model-Driven: Regression

- [Regression Model with Trend? (117)] -
- [Regression Model with Seasonality? (125)] -
- [Regression Model with Trend and Seasonality? (129)] -
- [Measuring Predictive Accuracy (51+)] -

```
# Fit basic regression model with trend and seasonality
sales_lm <- tslm(sales_train_ts ~ trend + I(trend ^ 2) + season)

# Forecast (predict) for validation period
sales_lm_pred <- forecast(sales_lm, h = sales_n_valid, level = 0)

# Summarize performance metrics
accuracy(sales_lm_pred, sales_valid_ts)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.0000000000000005	525	393	-8	21	0.4	0.47	NA
Test set	-529.415459820942715	735	583	-16	17	0.6	0.06	0.7

```
# Plot simple regression model
plot(sales_lm_pred, type = "l",
     main = "Store Sales for Select Product Families | Regression Forecaster",
     xlab = TeX(r"(\textbf{Date (weekly \textit{\$t\$}}) )"), ylab = TeX(r"(\textbf{Sales (\textit{\$y_t\$}}) (000) )"),
     las = 1, cex.axis = 0.7,
     xlim = c(x_train, xmax), ylim = c(0, ymax))

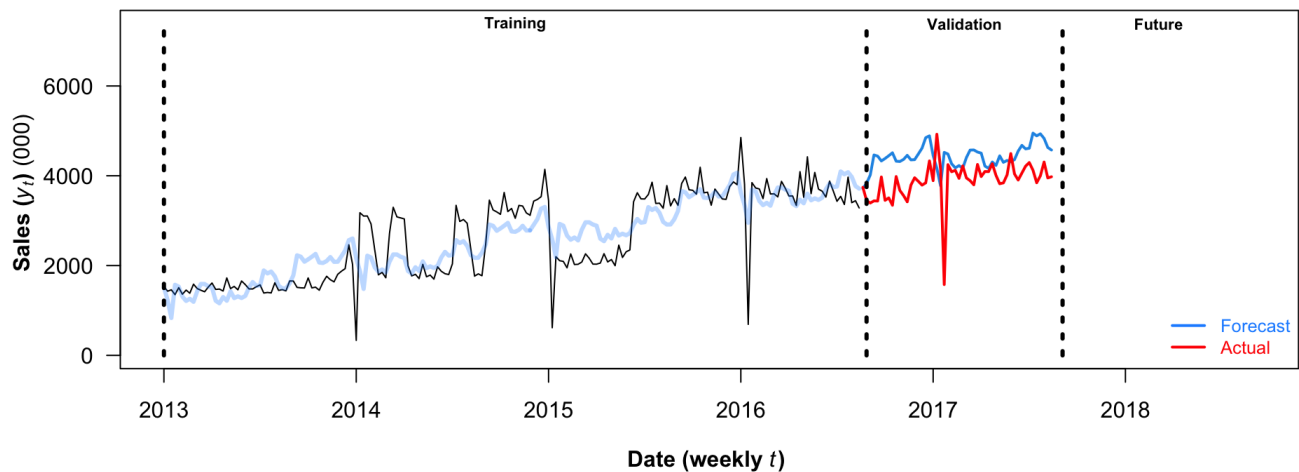
# Add additional lines
lines(sales_lm$fitted, lwd = 3, col = adjustcolor("dodgerblue", alpha.f = 0.3))
lines(sales_valid_ts, lwd = 2, col = "red")

# Add plot defintions
lines(c(x_train, x_train), c(0, ymax), lwd = 3, lty = 3)
lines(c(x_valid, x_valid), c(0, ymax), lwd = 3, lty = 3)
lines(c(x_future, x_future), c(0, ymax), lwd = 3, lty = 3)

text(x_train + ((x_valid - x_train) * 0.5), ymax, "Training", font = 2, cex = 0.7)
text(x_valid + ((x_future - x_valid) * 0.5), ymax, "Validation", font = 2, cex = 0.7)
text(x_future + 0.5, ymax, "Future", font = 2, cex = 0.7)

legend("bottomright",
     legend = c("Forecast", "Actual"),
     col = c("dodgerblue", "red"),
     lwd = 2, lty = 1, cex = 0.8,
     box.lty = 0, bg = NULL, text.col = c("dodgerblue", "red"))
```

Store Sales for Select Product Families | Regression Forecaster

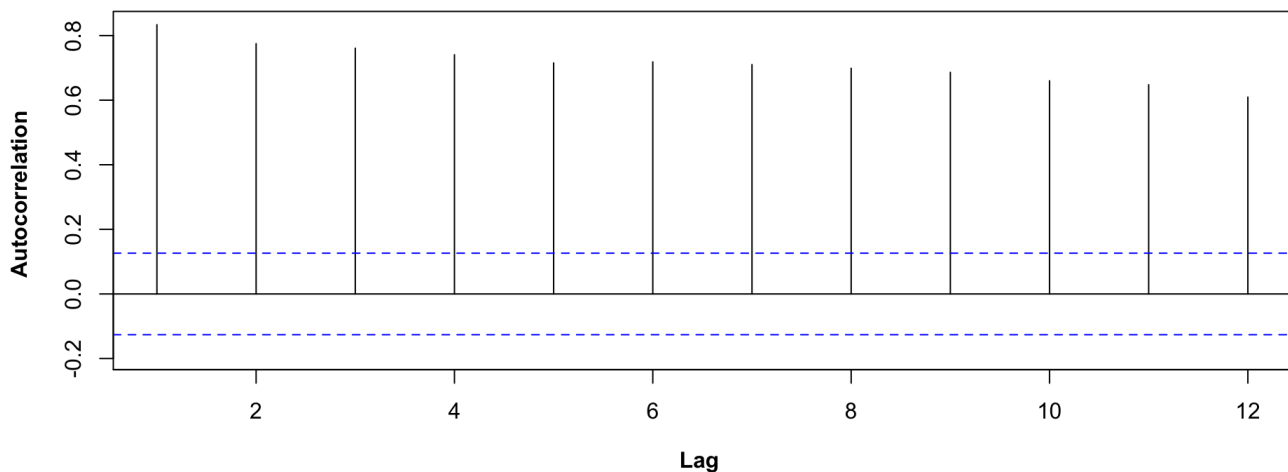


Model-Driven: Regression w/Autocorrelation and External Information

- [Autocorrelation? (143)] -
- [ARIMA Model? (147)] -
- [External Information (70)] - [consider causal/correlated?]
- [Including External Information? (154)] -

```
#  
Acf(sales_ts, lag.max = 12, type = "correlation", plot = TRUE,  
    main = "Store Sales for Select Product Families | Autocorrelation",  
    xlab = TeX(r"(\textbf{Lag} )"), ylab = TeX(r"(\textbf{Autocorrelation} )"),  
    xaxt = "n",  
    level = 95)
```

Store Sales for Select Product Families | Autocorrelation



Model-Driven: Opportunities

- [Combining Methods and Ensembles (73)] -

Evaluate & Compare Performance

- [\[Comparing Two Models? \(65\)\]](#) -
 - [\[Evaluating Predictability? \(153\)\]](#) -
-

Implement Forecasts/System

- [\[Joining Partitions for Forecasting \(47\)\]](#) -
 - [\[Creating Forecasts from the Chosen Model? \(132\)\]](#) -
-

1. University of San Diego, cvu@sandiego.edu (<mailto:cvu@sandiego.edu>)↗

2. University of San Diego, dfriesen@sandiego.edu (<mailto:dfriesen@sandiego.edu>)↗