

```
knitr::opts_chunk$set(echo = TRUE)
```

Data: Transaction Stream Time-series

The R ecosystem contains a vast number of time series standards and packages; this can make operations with time series confusing. Addressing the possibilities is beyond the scope of this book, though the auditor should be aware that there are many methods available for forecasting and analysis of the time series of events and transaction streams. The following code chunks provide examples of forecasting using the **base R** time series class **ts** and packages that operate on that class of data. The package **tsbox** is brought in once to regularize (fill in missing dates) the time series with **ts_regular()** and **forecast** package for ARIMA forecasting. The following code chunk reads credit sales and collections data in from the 2019 simulation data generated by code in the last chapter, and converts these to **ts** class for use in ARIMA forecasting of 2020 cash and collections transaction streams. These can then be differenced to forecast the change in accounts receivable balance.

```
library(lubridate)
library(tidyverse)
library(readr)
library(tsbox)

## Set the directory for the new files (modify the path as needed)
default_dir <- "/home/westland/audit_analytics_book/audit_simulated_files/"

if (file.exists(default_dir)){
  setwd(default_dir)
} else {
  dir.create(default_dir)
  setwd(default_dir)
}

credit_sales <-
  read_csv("real_world_credit_sales.csv", col_types = cols(X1 =
col_skip())) %>%
  mutate(sales_amount = sales_count * sales_unit_price) %>%
  select(
    sales_date = invoice_date,
    sales_amount,
    collection_date,
    collection_amount
  )

ggplot(credit_sales) +
  geom_line(aes(x=sales_date, y=sales_amount, color="red"))+
  geom_line(aes(x=collection_date, y=collection_amount, color="blue"))
```

```
sales <-
  credit_sales[,1:2] %>%
  group_by(sales_date) %>%
  summarize(sales_amount = sum(sales_amount, na.rm = TRUE)) %>%
  arrange(sales_date)

sales_st <- sales[1,1]
sales_end <- sales[nrow(sales),1]

sales <- tsbox::ts_regular(sales)
## set dates with no sales to zero with tsbox package
sales$sales_amount[is.na(sales$sales_amount)] <- 0
## set dates with NA to zero
sales <- as.ts(sales)

collections <-
  credit_sales[,3:4] %>%
  group_by(collection_date) %>%
  summarize(collection_amount = sum(collection_amount, na.rm = TRUE)) %>%
  arrange(collection_date)

collections_st <- collections[1,1]
collections_end <- collections[nrow(sales),1]

collections <-
  ts_regular(collections)
## set dates with no sales to zero
collections$collection_amount[
  is.na(collections$collection_amount)] <- 0
## set dates with NA to zero
collections <- as.ts(collections)
```

Forecasts

Good forecasting involves making educated guesses. Good forecasts satisfy four criteria:

1. **Non-arbitrary:** forecasting methodology defines a clear role for data, model drivers, assumptions, and hypotheses, which will be applied consistently from analysis to analysis. We can minimize these biases by using a standard formal methodology which encourages four traits of good research: (1) explains observed phenomena, and explains why; (2) is consistent with previously established knowledge; (3) is verifiable by other parties with access to the same data; and (4) stimulates further discussion, investigation, and revision as data become available.
2. **Collective:** forecasts should be easily understood by others, and their assumptions stated clearly enough to assure that others can replicate their conclusions.
3. **Reliable:** forecasts can be relied upon to make decisions. Reliability implies not only a specified degree of accuracy in financial reporting, but a clear idea of how accurate the reported numbers are

through reporting of a dispersion statistic – for example, variance or standard deviation – to measure the reliability of the reported value.

4. Consistent and robust: forecasts will not change in the absence of fundamental information. Robust methods would limit and assure that different analysts using similar data would produce similar valuations, or where they were different, could clearly explain the assumptions which account for that difference.

This sort of valuation analysis is common in financial analysis – the representation is called a binomial variance lattice representation of a real options problem – and the problem is that of finding the value of management's real option. In financial options analysis, there are multiple methodologies and approaches used to calculate an option's value including closed-form equations like the Black-Scholes model and its modifications, Monte Carlo path-dependent simulation methods, lattices, variance reduction and other numerical techniques, and partial-differential equations (of which the Black-Scholes model is but one solution). Binomial lattices are easy to implement and easy to explain. In the limit, results obtained through the use of binomial lattices tend to approach those derived from closed-form solutions.

The expanding range of possible values between the confidence limits (the two lines extending into the future) is called a 'value cone' and describes the risk profile of the particular business model that management designs to promote the new or untested venture. The value cone reflects the projection of the variability in our behavioral model into the future. It is in general expanding because our forecasting errors compound themselves moving further into the future. Any misestimation of value generation next year will incorrectly forecast the path of development of the business model, and that misforecast will in turn create even greater forecast errors in the subsequent years. Forecast errors result because: (1) we have incomplete control over the drivers of value creation, and (2) we have imperfect knowledge about exogenous events that influence the success of our business.

The behavioral model, which encapsulates what is known about comparable firms, the industry 'best practices' and prior years' results, should be used to develop the forecasting model. In the code chunk below, it is assumed that the behavioral model suggests an ARIMA time series model based on Holt's linear method with additive errors. The three components ARIMA(p, d, q) are:

1. the AR order, typically chosen to minimize the Akaike information,
2. the degree of differencing. Differencing is a transformation applied to time-series data in order to make it stationary. A stationary time series' properties do not depend on the time at which the series is observed., and
3. the moving average order.

```
library(forecast)

# fit an ARIMA (p, d, q) model
# order is the non-seasonal
#part of the ARIMA model:
#the three components (p, d, q) are
#the AR order, the degree of differencing, and the MA order.
# p is the order of the autoregressive part and
# q is the order of the moving average part.
```

```

# Holt's linear method with additive errors,
#or double exponential smoothing
fit <- arima(sales[,2], order=c(0,2,2))

# predictive accuracy, show the first few values
head(accuracy(fit))

# predict next year's observations
f_cast <- forecast(fit, 365)
plot(f_cast,
     main="ARIMA(0,2,2) Forecasts for Sales Transactions",
     xlab="Time Period (Days) starting 2019-1-1",
     ylab="Daily Transaction Total"
     )

fit <- arima(collections[,2], order=c(0,2,2))

# predictive accuracy, show the first few values
head(accuracy(fit))

# predict next year's observations
f_cast <- forecast(fit, 365)
plot(f_cast,
     main="ARIMA(0,2,2) Forecasts for Collections Transactions",
     xlab="Time Period (Days) starting 2019-1-1",
     ylab="Daily Transaction Total"
     )

forecast_sales <-
  arima(sales[,2], order=c(0,2,2)) %>%
  forecast(365)

upper_sales <-
  as.ts(forecast_sales$upper[,2] ) %>%
  window(366, 365*2) %>%
  sum()

lower_sales <-
  as.ts(forecast_sales$lower[,2] ) %>%
  window(366, 365*2) %>%
  sum()

forecast_sales <-
  as.ts(forecast_sales$mean ) %>%
  window(366, 365*2) %>%
  sum()

```

```

forecast_collections <-
  arima(collections[,2], order=c(0,2,2)) %>%
  forecast(365)

upper_collections <-
  as.ts(forecast_collections$upper[,2] ) %>%
  window(366, 365*2) %>%
  sum()

lower_collections <-
  as.ts(forecast_collections$lower[,2] ) %>%
  window(366, 365*2) %>%
  sum()

forecast_collections <-
  as.ts(forecast_collections$mean ) %>%
  window(366, 365*2) %>%
  sum()

change_ar <- forecast_sales - forecast_collections
upper_ar <- upper_sales - upper_collections
lower_ar <- lower_sales - lower_collections

cat(
  "\n\n The Expected 2020 Sales = $",
  prettyNum(forecast_sales, big.mark=","))
cat(
  "\n The Expected 2020 Collections = $",
  prettyNum(forecast_collections, big.mark=","))
cat(
  "\n The Expected 2020 Change in Accounts Receivable Balance = $",
  prettyNum(change_ar, big.mark=","))
cat(
  "\n 2020 AR Change : 95% Upper Bound = $",
  prettyNum(upper_ar, big.mark=","),
  " & 95% Lower Bound = $",
  prettyNum(lower_ar, big.mark=","))

```

The previous calculations estimate an expected increase in accounts receivable value from beginning to end of 2020 with a wide range of values within the confidence limits. Forecasted collections are understated, mistimed, or misforecasted in this rather crude analysis. The auditor would need to further investigate collection transaction flows to provide a more reliable forecast.

Generating a Current Valuation

Forecast models generate estimates and confidence bounds for value-flows that will be generated from the business surrounding the new or untested venture over a series of future time periods. In order to

obtain a present value for the business (or more correctly, a pair of confidence limits for the present value) it is necessary to discount the future values back to the current value. This is done by summing all the future values generated in future periods t by a discount factor $\frac{1}{(1+r)^t}$ where r is the discount rate for each period. Discount rate r can be viewed either as:

1. a measure of management's impatience – how quickly they want to get their investment back; or
2. as an opportunity cost – a measure of potential return from other investment opportunities which were foregone in order to fund this new or untested venture business. From either perspective, deciding on the appropriate discount rate can be difficult. There is an extensive literature in finance that addresses the selection of discount rates. A detailed review is beyond the scope of this text, but we will briefly look at some of the more important considerations here.

Once components are assembled for a valuation, the future cash flows may be forecasted and then discounted back to the 'present value' at the selected discount rate. Use the standard net present value formula:

$$NPV = cf_0 - \sum_{k=1}^n \frac{cf_k}{(1+i)^{k}}$$

```
daily_discount <- .1/365 # 10% annual rate
NPV <- 0
for(i in 1:length(forecast_sales)) NPV <-
  NPV +
  as.numeric(forecast_sales[i]/
              ((daily_discount+1)^(i)))

cat(
  "\n\n The Jan. 1, 2020 NPV of the forecasted sales transactions for 2020
= $",
  prettyNum(NPV, big.mark=","))
```