

Transferring 2013 Analysis to R

Sean Davern

June 5, 2019

Objective of this Work

The objective of the work captured in this notebook is to translate part of an analysis done in 2013 (See Davern (2013)) into R to learn and demonstrate multiple R capabilities and workflows.

Import of Data

The original data was provided by John Earling in a work book entitled 'Weekly PayPay & Tithes .xls' workbook. The layout/format of that workbook was not conducive to easily loading into R [nor JMP originally] and so was transcribed into the workbook 'Giving Data.xlsx' and then read into R.

```
source("../code/0-Extract data from Excel.R")
df # This is the data frame resulting from the import.
```

```
## # A tibble: 469 x 7
##   week.ending      month  year paypal offering total monthly.giving.~
##   <dtm>          <chr> <dbl> <dbl>    <dbl> <dbl>          <dbl>
## 1 2010-01-03 00:00:00 Janua~ 2010    75    1560    1635             NA
## 2 2010-01-10 00:00:00 Janua~ 2010   575    3129    3704             NA
## 3 2010-01-17 00:00:00 Janua~ 2010   475    2025    2500             NA
## 4 2010-01-24 00:00:00 Janua~ 2010    75    1180.   1255.             NA
## 5 2010-01-31 00:00:00 Janua~ 2010  2180    2967.   5147.             45
## 6 2010-02-07 00:00:00 Febru~ 2010    75    4722.   4798.             NA
## 7 2010-02-14 00:00:00 Febru~ 2010   585    2925    3510             NA
## 8 2010-02-21 00:00:00 Febru~ 2010   200    3299    3499             NA
## 9 2010-02-28 00:00:00 Febru~ 2010  6350    4281   10631             41
## 10 2010-03-07 00:00:00 March  2010   770    1170    1940             NA
## # ... with 459 more rows
```

Some Minor Data Validation

As a first validation I'll check that the weekly PayPal and offering amounts sum to the weekly totals ($paypal_i + offering_i \stackrel{?}{=} total_i$), reporting only those that aren't equal:

```
source("../code/1-Validate totals.R")
```

```
## ***** WARNING *****
## Some 'total' observations don't equal the sum of 'paypal' and 'offering'!
## # A tibble: 3 x 8
##   week.ending      month  year paypal offering total calcd.total diff
##   <dtm>          <chr> <dbl> <dbl>    <dbl> <dbl>    <dbl> <dbl>
## 1 2015-12-27 00:00:00 Decemb~ 2015   1902    9306   16958    11208   5750
## 2 2016-12-25 00:00:00 Decemb~ 2016   4635   10089   22849    14724   8125
## 3 2017-04-23 00:00:00 April  2017    635    4480    4615     5115  -500
```

Ok, so December 2015 and 2016 seem to have totals greater than accounted for by the PayPal and offering amounts. That's perhaps explainable by other end-of-year giving coming in another way. However, the April 2017 discrepancy seems to be missing \$500. I'll need to look into that.

Data Transformation

Aggregating the monthly totals and preparing to model month values...

```
# Data transformation: Calculate monthly giving totals.
# Make Month a categorical variable with levels in the order that
# months occur in the year otherwise months are sorted alphabetically.
df$month <- factor(df$month, month.name)
# Aggregate the monthly Totals from giving.data in sums for each month.
MonthTotals <-
  aggregate(df$total, by = list(df$month, df$year), FUN = sum)
# Exclude the months that don't have totals yet.
MonthTotals <- MonthTotals[complete.cases(MonthTotals), ]
# Extract only rows containing 'monthly.giving.families' data.
df <- df[!is.na(df$monthly.giving.families),]
# Now replace Totals (which were weekly totals) with calculated aggregates
df$total <- MonthTotals$x
# paypal & offering columns are now misleading (only week's value) so remove them.
df <- select(df, -paypal, -offering)
```

Adding the number of giving Sundays in the month and the average giving each week per month...

```
library("magrittr")
source("../code/NumOfGivenDayOfWeekInMonth.R")
# Calculate and add the columns SundaysInMonth with calculated values
# and MonthsGivingPerWeek
df <- df %>%
  mutate(SundaysInMonth =
    NumOfGivenDayOfWeekInMonth(df$week.ending, "Sunday")) %>%
  mutate(MonthsGivingPerWeek = total / SundaysInMonth)
```

Enable modeling year as factor rather than a number...

```
# Make year a categorical variable so coefficients are easier to interpret.
df$year <- as.factor(df$year)
```

Save the resulting R tibble:

```
# Code chunk eval=false so files don't get overwritten willy nilly.
# Write it as a csv:
write.csv(x = df,
  file = "../data/Cleaned and Transformed Giving Data.csv",
  row.names = FALSE)
# Save it also as an R object that can be loaded into a new R object.
saveRDS(df, file = "../data/Cleaned and Transformed Giving Data.rds")
```

Replicating Previous Modeling

The relatively simple model derived in 2013 (see Davern 2013, pg. 11) and used again in 2018 used this model:

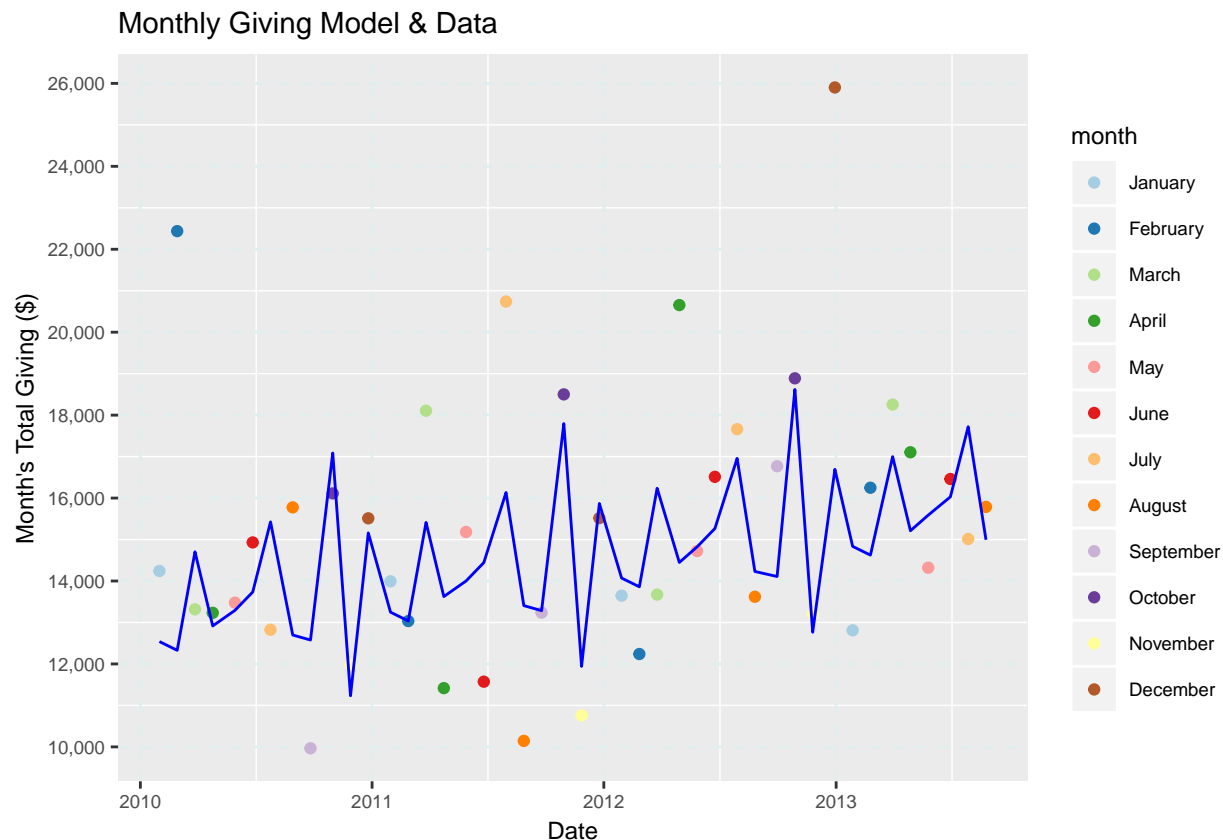
$$\text{Monthly Giving} = a + b_{\text{year}} + c_{\text{month}}$$

where a is an overall grand average of the monthly giving amount, b_{year} is an adjustment for the given year and c_{month} is an adjustment for the month. The model was originally regressed on giving data from Jan 2010 through August 2013 excluding 3 high-fliers with known exceptional donations.

The R function `lm` uses a similar model except where a is the predicted Jan 2011 giving amount, so $b_{2011} = 0$ and $c_{\text{January}} = 0$. Regressing this model:

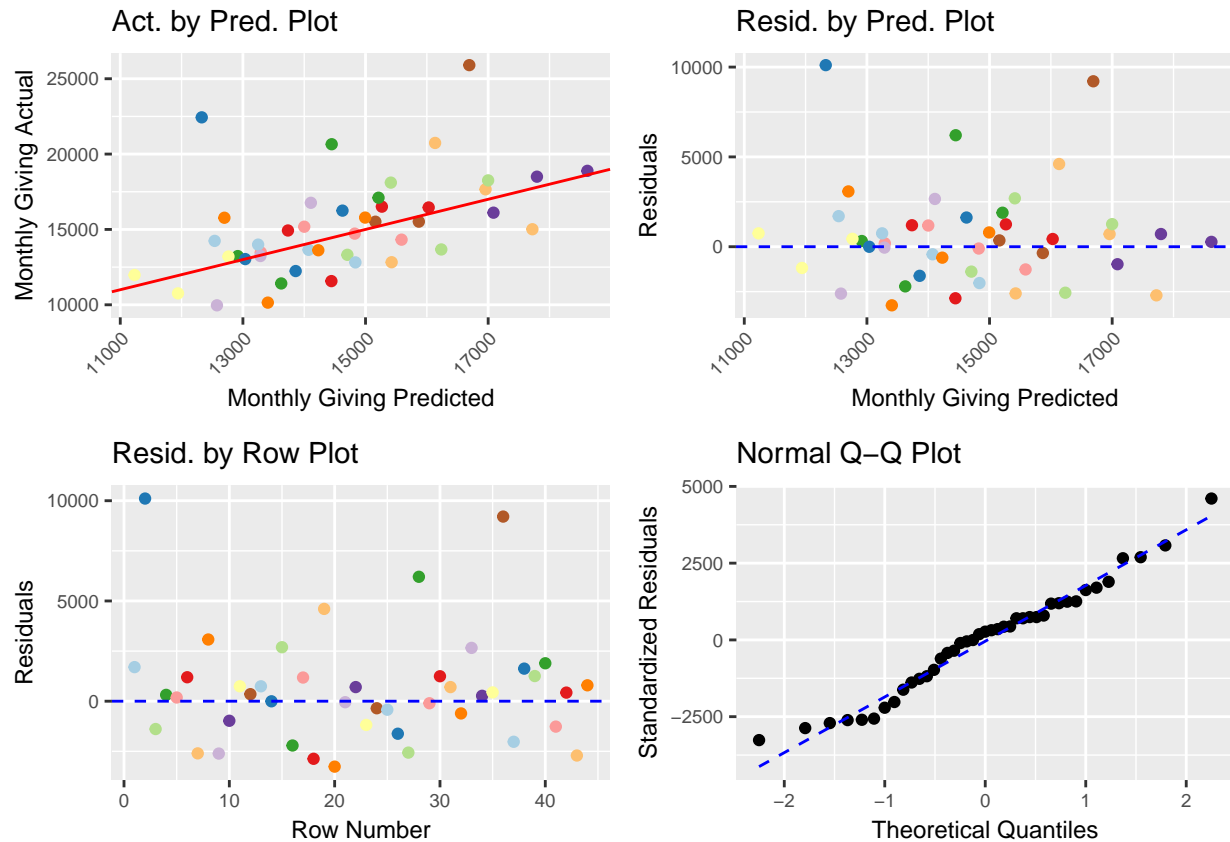
```
library("magrittr")
# Pair the data down to that used in the original analysis
df2 <- df[as.Date(df$week.ending) > "2010-01-01" &
         as.Date(df$week.ending) < "2013-08-31",] %>%
  mutate(excluded = FALSE)
df2$excluded[as.Date(df2$week.ending) == "2010-02-28"] <- TRUE
df2$excluded[as.Date(df2$week.ending) == "2012-04-29"] <- TRUE
df2$excluded[as.Date(df2$week.ending) == "2012-12-30"] <- TRUE
# Regress the model cluding the indicated values:
mod <- lm(
  formula = total ~
    year + month,
  data = df2[df2$excluded!=TRUE,]
)
```

Which gives the resulting model fit:



Note: excluded points [high fliers] are shown (above and below) though they weren't included in the regression.
Here are the fit diagnostics:

```
## Analysis of Variance Table
##
## Response: total
##      Df    Sum Sq Mean Sq F value Pr(>F)
## year    3  21431854  7143951   1.4506  0.251
## month   11  93473089  8497554   1.7254  0.123
## Residuals 26 128049780 4924992
```



```
##
## Call:
## lm(formula = total ~ year + month, data = df2[df2$excluded !=
##      TRUE, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3262.7  -1266.4    269.4   1181.2   4604.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12539.40    1264.30   9.918 2.52e-10 ***
## year2011       708.72     933.26   0.759  0.4544
## year2012      1531.73     997.47   1.536  0.1367
## year2013      2295.36    1082.20   2.121  0.0436 *
## monthFebruary  -210.59    1706.96  -0.123  0.9028
## monthMarch      2164.52    1569.23   1.379  0.1795
```

```

## monthApril      376.99    1707.70    0.221    0.8270
## monthMay        752.68    1569.23    0.480    0.6355
## monthJune       1195.27    1569.23    0.762    0.4531
## monthJuly       2886.33    1569.23    1.839    0.0773 .
## monthAugust     158.54    1569.23    0.101    0.9203
## monthSeptember   37.28    1710.48    0.022    0.9828
## monthOctober    4547.45    1710.48    2.659    0.0132 *
## monthNovember  -1306.57    1710.48   -0.764    0.4518
## monthDecember   2621.14    1956.39    1.340    0.1919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2219 on 26 degrees of freedom
## Multiple R-squared:  0.4729, Adjusted R-squared:  0.1892
## F-statistic: 1.666 on 14 and 26 DF,  p-value: 0.1261

```

The results obtained are different in a number of ways from what was obtained in 2013. The general fit (residuals) and shape of the predicted values are similar to that obtained in 2013 though the June predictions seem further away than the other months. The regressed coefficients are obviously very different, but this is largely due to the method JMP uses for regressing factor coefficients. However, the difference in the June predictions and slightly different factor p-values tells me that something in the underlying data is probably different. I'm not going to spend the time to diagnose the precise details of the difference since the objective is to translate the analysis to R rather than reproduce JMP.

References

Davern, Sean. 2013. "An Assessment of Requested Seed Core Support for Expansion Based on Analysis of Giving." *Internal Report*, August. file://../Reports/August%202013%20Analysis%20of%20Giving.pdf.