```
# Kaggle Walmart recruiting competition 2014-02-20 to 2014-05-05.

WORKING_DIRECTORY = "~/walmart"
options(stringsAsFactors = FALSE)
setwd(WORKING_DIRECTORY)

library(Hmisc)  # Hmisc is first so its summarize function does not mask plyr's
library(plyr)
library(testthat)
library(lubridate)
library(stringr)

trend_sales <- function(v_sales, v_id, v_dt, id_num, trend_fctr) {

  # Apply a trend factor to the historical sales, moving them to the
  # beginning of the test period.
  #
  # Args:
  #   v_sales: Vector of all sales in the test set.
  #   v_id: Vector of all store or department ids in the test set.
  #   v_dt: Vector of all dates in the test set.
  #   id_num: The store or department for which sales are to be trended.
  #   trend_fctr: The historical (not prospective) trend factor.
  #
  # Returns:
  #   The revised sales vector with trend applied to the
  #   components corresponding to id_num.

  ind <- which(v_id == id_num)
  wks_between <- as.integer(difftime (http://www.opengroup.org/onlinepubs/009695399/functions/difftime.html)(v_dt[ind], min(v_dt[ind]), units="weeks"))
  fctr <- trend_fctr^(1/52 * (52 - wks_between))
  v_sales[ind] <- round(v_sales[ind] * fctr, 2)
  return(v_sales)
}

blend_weeks <- function(next_yr_dt, coef1 = NULL, coef2 = NULL) {

  # Given a date from the test set, the week ending on the corresponding date
  # in the training set will usually straddle two training weeks. This function
  # calculates an appropriate weighted average of the train weeks for
  # predicting the test week.
  #
  # Args:
  #   next_year_dt: An end of week date (must be a Friday) from the test set
  #   coef1, coef2: Specify the weights rather than calculating them. Not used.
  #
  # Returns:
  #   A data frame with the test set id and predicted sales for next_yr_dt.
  #
  # Note:
  # Dataframes test and train are used globally and are referenced within the
  # blend_weeks function, although not passed as arguments.

  stopifnot(wday(next_yr_dt) == 6)  # End of week must be a Friday.
  dt <- next_yr_dt  - years(1)
  stopifnot(wday(dt) != 6)
```

```
  days_to_friday <- (13 - wday(dt)) %% 7
  next_friday <- dt + days(days_to_friday)
  prev_friday <- next_friday - days(7)
  stopifnot(wday(next_friday) == 6)
  stopifnot(wday(prev_friday) == 6)

  df1 <- subset(train, dt == next_friday)
  df2 <- subset(train, dt == prev_friday)
  df_valid <- subset(test, dt == next_yr_dt)[, c("Store", "Dept")]

  df_both <- merge(df1[, 1:4], df2[, 1:4], by = c("Store", "Dept"),
        all = TRUE)
  df_both <- merge(df_valid, df_both, by = c("Store", "Dept"), all.x = T)
  df_both[, c("sales.x", "sales.y")] <-
    Hmisc::impute(df_both[, c("sales.x", "sales.y")], 0)

  if(is.null(coef1)) coef1 <- 1 - days_to_friday/7
  if(is.null(coef2)) coef2 <- days_to_friday/7
  blended_sales <- round(with(df_both, coef1 * sales.x +
                                coef2 * sales.y), 0)
  Id <- with(df_both, paste(Store, Dept, next_yr_dt, sep = "_"))
  df_ans <- data.frame(Id = Id, sales = blended_sales)
  return(df_ans)
}

# Read and validate the data ---------------------------------------------------
train <- readRDS("train.rds")  # Training data covers 2010-02-05 to 2012-11-01
test <- readRDS("test.rds")    # Test data covers 2012-11-02 to 2013-07-26
expect_equal(nrow(train), 421570)
expect_equal(nrow(test), 115064)
expect_equal(with(train, length(unique(paste(Store, Dept, Date)))), nrow(train))
expect_equal(with(test, length(unique(paste(Store, Dept, Date)))), nrow(test))

# Create derived variables -----------------------------------------------------
train <- mutate(train, dt = ymd(Date), yr = year(dt), wk = week(dt))
train <- rename (http://www.opengroup.org/onlinepubs/009695399/functions/rename.
html)(train, replace = c("Weekly_Sales" = "sales"))
test <- mutate(test, dt = ymd(Date), yr = year(dt), wk = week(dt),
               prior_yr = yr - 1)

# Map weeks of test period to corresponding weeks in train period -------------
# Week Mapping Adjustments:
# Thanksgiving 2012 is in week 47, Thanksgiving 2011 in week 48,
# thus 47 is replaced with 48 and 48 is replaced by 49.
#
# Easter 2013 is on March 31 (week 13).
# Model week after Easter (14) by week after Easter (15).
# For Easter week wound up just doing the same blending as for other weeks.
test$wk <- plyr::mapvalues(test$wk, from = c(47, 48, 14), to = c(48, 49, 15))

# Make initial predictions -----------------------------------------------------
# Construct the initial test set predictions (just a merge with train, lagging
# the test set by one year).
ans <- merge(test, train, by.x = c("Store", "Dept", "prior_yr", "wk"),
             by.y = c("Store", "Dept", "yr", "wk"), all.x = TRUE)
ans$sales[is.na(ans$sales)] <- 0
ans <- ans[, c("Store", "Dept", "Date.x", "sales")]
ans$Id <- with(ans, paste(Store, Dept, Date.x, sep = "_"))

# Week blending adjustments ----------------------------------------------------
# Remove records in the test set that will be replaced by records derived
# from blending.
```

```
UNBLENDED_DATES <- c("2012-11-23", "2012-11-30", "2013-04-05")
BLEND_DATES <- setdiff(as.character(ymd("2012-11-02") + weeks(0:38)),
                       UNBLENDED_DATES)
ans <- subset(ans, !(Date.x %in% BLEND_DATES))
sub <- ans[, c("Id", "sales")]

# Calculate the blended weeks and add them back to sub using plyr::rbind.fill.
blended_weeks <- plyr::rbind.fill(lapply(ymd(BLEND_DATES), blend_weeks))
sub <- rbind(sub, blended_weeks)

# Reconstruct date, store, and department from the submission -----------------
# (awkward - could be cleaned up)
dt <- ymd(str_extract(sub$Id, ".{10}$" ))
store <- str_extract(sub$Id, "[0-9]+")
dept <- substr(str_extract(sub$Id, "_[0-9]+"), 2, 3)

# Make the trend adjustments (geometric mean of quarters). --------------------
store_trend_data <- list(c(1, 1.01), c(2, 1.01), c(3, 1.07), c(4, 1.02),
                         c(5, 1.05), c(6, 1.01), c(7, 1.03), c(8, 1.00),
                         c(9, 1.01), c(10, 0.97), c(11, 1.00), c(12, 0.99),
                         c(13, 1.01), c(14, 0.85), c(15, 0.95), c(16, 0.99),
                         c(17, 1.04), c(18, 1.03), c(19, 0.96), c(20, 0.99),
                         c(21, 0.90), c(22, 0.97), c(23, 1), c(24, 0.99),
                         c(25, 1.00), c(26, 1.00), c(27, 0.94), c(28, 0.95),
                         c(29, 0.98), c(30, 1.01), c(31, 0.96), c(32, 0.99),
                         c(33, 1.04), c(34, 1.01), c(35, 1.00), c(36, 0.80),
                         c(37, 0.97), c(38, 1.10), c(39, 1.07), c(40, 0.99),
                         c(41, 1.04), c(42, 1.00), c(43, 0.97), c(44, 1.08),
                         c(45, 0.97))
for(v in store_trend_data) {
  sub$sales <- trend_sales(sub$sales, store, dt, v[1], v[2])
}

dept_trend_data <- list(c(1, 0.96), c(2, 0.98), c(3, 1.01), c(4, 1),
                        c(5, 0.91), c(6, 0.79), c(7, 0.99), c(8, 0.99),
                        c(9, 1.03), c(10, 0.99), c(11, 0.98), c(12, 0.98),
                        c(13, 0.98), c(14, 1.02), c(16, 0.95), c(17, 0.97),
                        c(18, 0.87), c(19, 1.06), c(20, 0.98), c(21, 0.94),
                        c(22, 1.01), c(23, 1.02), c(24, 1), c(25, 0.96),
                        c(26, 0.96), c(27, 1.02), c(28, 0.89), c(29, 1.02),
                        c(30, 0.92), c(31, 0.9), c(32, 0.97), c(33, 0.99),
                        c(34, 1.02), c(35, 0.92), c(36, 0.79), c(37, 0.97),
                        c(38, 0.98), c(40, 1.01), c(41, 0.94), c(42, 1.01),
                        c(44, 1.02), c(45, 0.53), c(46, 0.99), c(48, 1.96),
                        c(49, 0.96), c(50, 0.97), c(52, 0.93), c(54, 0.54),
                        c(55, 0.83), c(56, 0.93), c(58, 1.13), c(59, 0.7),
                        c(60, 1.02), c(65, 1.09), c(67, 1.02), c(71, 0.98),
                        c(72, 0.96), c(74, 0.97), c(79, 0.98), c(80, 0.96),
                        c(81, 0.98), c(82, 1.02), c(83, 1.01), c(85, 0.9),
                        c(87, 1.14), c(90, 0.98), c(91, 0.98), c(92, 1.04),
                        c(93, 1.02), c(94, 0.96), c(95, 0.99), c(96, 1.04),
                        c(97, 0.97), c(98, 0.95), c(99, 1.19))
for(v in dept_trend_data) {
  sub$sales <- trend_sales(sub$sales, dept, dt, v[1], v[2])
}

# Save the submission ----------------------------------------------------------
sub <- sub[, c("Id", "sales")]
names(sub) <- c("Id", "Weekly_Sales")
sub <- arrange(sub, Id)
expect_equal(nrow(sub), 115064)
z <- gzfile("submission.csv.gz")
```

```
write.csv(sub, z, row.names = FALSE)
```

Runtime error #stdin #stdout #stderr 0.3s 22832KB

comments (0)

stdin copy

```
Standard input is empty
```

stdout

```
Standard output is empty
```

stderr

```
Error in setwd(WORKING_DIRECTORY) : cannot change working directory
Execution halted
```

14, 0.02s, 0.0 s