dplyr Example 3 - Waiting for BART

Jim Porzak

2016-02-19

This example is inspired by a dplyr challenge from Ira Sharenow. Thanks Ira!

> Every hour one of my workers shows up at 15 after the hour to take a BART train home. Every hour two BART trains show up randomly in the interval [0,60). I wish to gather data on wait times. The problem is when both trains arrive in [0,15), so I need data from the next hour.

See Ira's full email at end.

Basically we have a simulation problem. We are going to generate two random arrival times for each hour. The number of hours, N_hours, to include in the simulation will determine the accuracy of the simulation.

Once we have the arrival times set up we just need to get the lag between 15 minutes after the hour and the next BART arrival.

The only tricky bit below is handling the case when a random arrival in an hour is before the worker gets to the platform. In that case, we bump the arival time to the next hour with `+ ifelse(train1_min < worker_minute, 60, 0)`.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(lubridate)
set.seed(1234)                  ## make reproducable
N_hours <- 10000                ## number of hours for the simulation

worker_minute <- 15             ## workers always arive a fixed minutes after the hour
hours <- seq(as.POSIXct("2001-01-01 00:00"), by = "hour", length.out = N_hours)
Waits <- data.frame(hours) %>%
  mutate(worker_at = hours + dminutes(worker_minute),
         train1_min = runif(N_hours, 0, 59.9999),
         train1_at = hours + dminutes(train1_min + ifelse(train1_min < worker_minute, 60, 0)),
         train1_wait = as.numeric(difftime(train1_at, worker_at, units = "mins")),
         train2_min = runif(N_hours, 0, 59.9999),
         train2_at = hours + dminutes(train2_min + ifelse(train2_min < worker_minute, 60, 0)),
         train2_wait = as.numeric(difftime(train2_at, worker_at, units = "mins")),
         worker_wait = ifelse(train1_wait <= train2_wait, train1_wait, train2_wait)
  )
glimpse(Waits)
```

```
## Observations: 10,000
## Variables: 9
## $ hours       (time) 2001-01-01 00:00:00, 2001-01-01 01:00:00, 2001-01...
## $ worker_at   (time) 2001-01-01 00:15:00, 2001-01-01 01:15:00, 2001-01...
## $ train1_min  (dbl) 6.8221933, 37.3379021, 36.5564230, 37.4027042, 51....
## $ train1_at   (time) 2001-01-01 01:06:49, 2001-01-01 01:37:20, 2001-01...
## $ train1_wait (dbl) 51.822193, 22.337902, 21.556423, 22.402704, 36.654...
## $ train2_min  (dbl) 12.278671, 29.054714, 38.145988, 57.552336, 10.725...
## $ train2_at   (time) 2001-01-01 01:12:16, 2001-01-01 01:29:03, 2001-01...
## $ train2_wait (dbl) 57.278671, 14.054714, 23.145988, 42.552336, 55.725...
## $ worker_wait (dbl) 51.822193, 14.054714, 21.556423, 22.402704, 36.654...
```

```
summary(Waits)
```

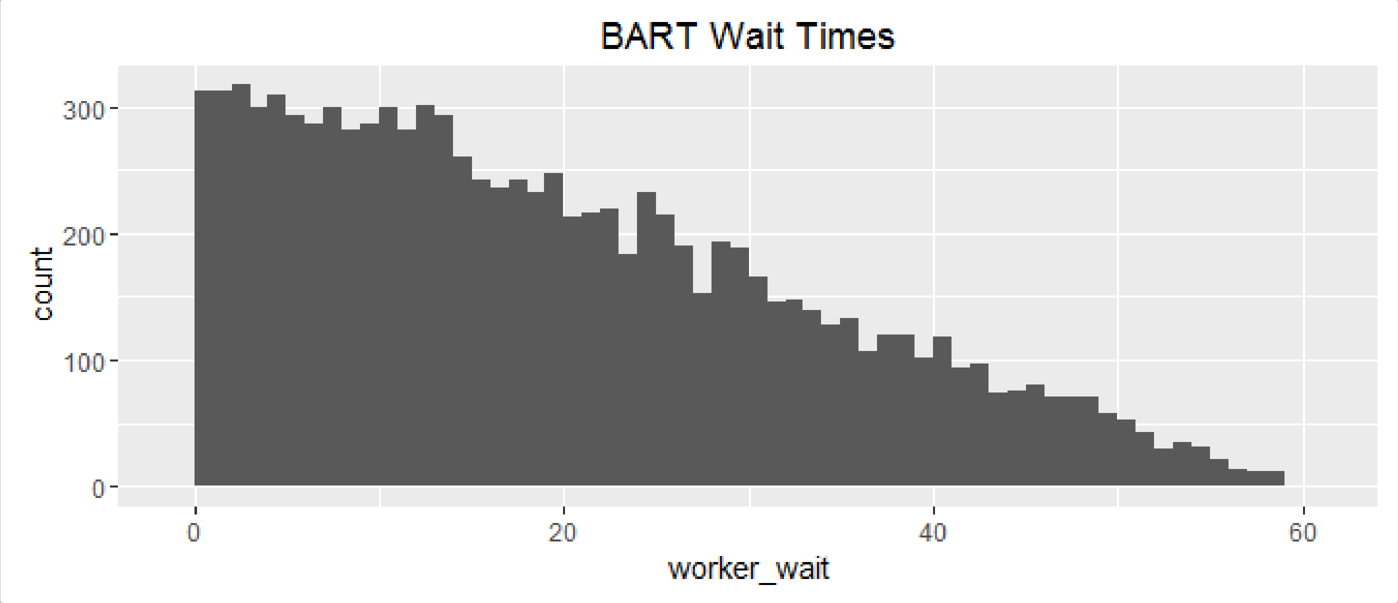
```
##      hours                       worker_at                  
##  Min.   :2001-01-01 00:00:00   Min.   :2001-01-01 00:15:00  
##  1st Qu.:2001-04-15 04:45:00   1st Qu.:2001-04-15 05:00:00  
##  Median :2001-07-28 08:30:00   Median :2001-07-28 08:45:00  
##  Mean   :2001-07-28 08:30:00   Mean   :2001-07-28 08:45:00  
##  3rd Qu.:2001-11-09 11:15:00   3rd Qu.:2001-11-09 11:30:00  
##  Max.   :2002-02-21 15:00:00   Max.   :2002-02-21 15:15:00  
##    train1_min         train1_at                    train1_wait     
##  Min.   : 0.02051   Min.   :2001-01-01 01:06:49   Min.   : 0.0041  
##  1st Qu.:15.14921   1st Qu.:2001-04-15 05:45:31   1st Qu.:14.8345  
##  Median :30.07590   Median :2001-07-28 09:25:37   Median :29.7087  
##  Mean   :30.01869   Mean   :2001-07-28 09:14:50   Mean   :29.8447  
##  3rd Qu.:44.84959   3rd Qu.:2001-11-09 12:06:19   3rd Qu.:44.7760  
##  Max.   :59.97553   Max.   :2002-02-21 15:59:23   Max.   :59.9977  
##    train2_min         train2_at                    train2_wait      
##  Min.   : 0.02558   Min.   :2001-01-01 01:12:16   Min.   : 0.03042  
##  1st Qu.:15.51914   1st Qu.:2001-04-15 05:12:58   1st Qu.:14.72610  
##  Median :30.18585   Median :2001-07-28 09:31:37   Median :29.44056  
##  Mean   :30.27424   Mean   :2001-07-28 09:14:47   Mean   :29.79424  
##  3rd Qu.:44.89734   3rd Qu.:2001-11-09 12:08:24   3rd Qu.:44.52156  
##  Max.   :59.99837   Max.   :2002-02-21 15:17:03   Max.   :59.99396  
##   worker_wait     
##  Min.   : 0.0041  
##  1st Qu.: 8.2188  
##  Median :17.3943
##  Mean   :19.7767
##  3rd Qu.:29.2651
##  Max.   :59.1778
```

Results

```
summary(Waits$worker_wait)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0041  8.2190 17.3900 19.7800 29.2700 59.1800
```

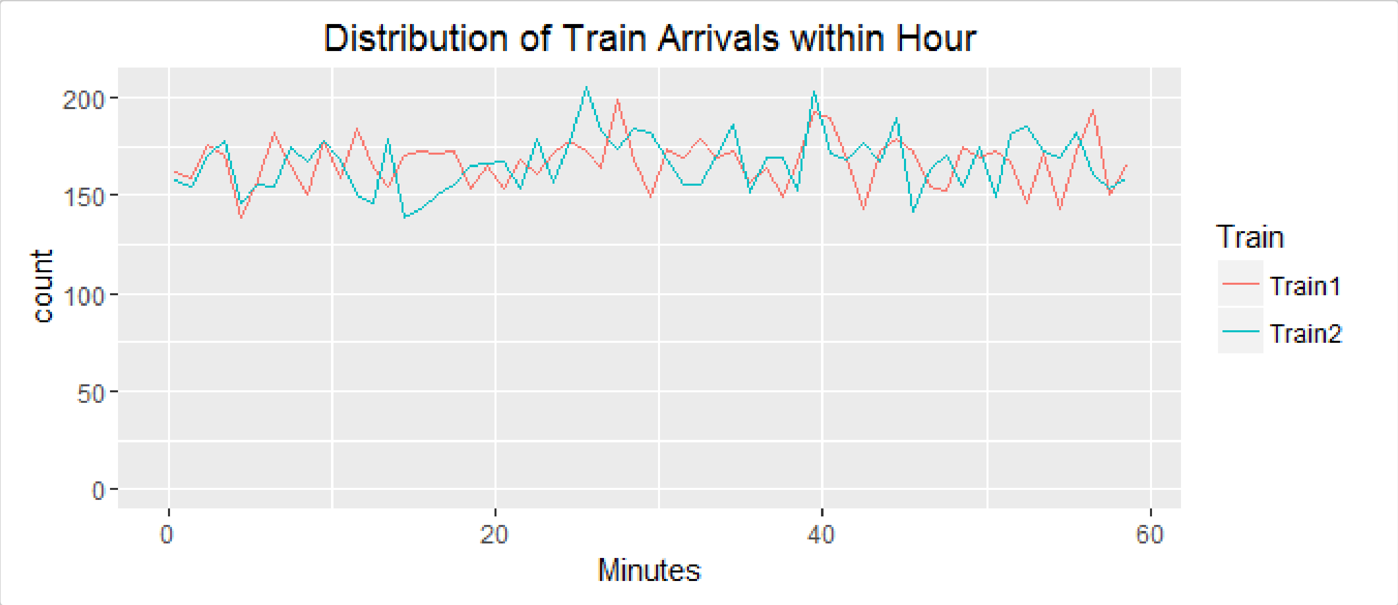
```
ggplot(Waits, aes(worker_wait)) +
  geom_histogram(binwidth = 1) +
  ggtitle("BART Wait Times")
```



Check distribution of train arrivals within hour

```
Train_Minutes <- Waits %>%
  select(hours, train1_min, train2_min) %>%
  rename(Train1 = train1_min, Train2 = train2_min) %>%
  gather(Train, Minutes, -hours)
ggplot(Train_Minutes, aes(Minutes, color = Train)) +
  geom_freqpoly(binwidth = 1) +
  xlim(0, 59) +
  ggtitle("Distribution of Train Arrivals within Hour")
```

```
## Warning: Removed 331 rows containing non-finite values (stat_bin).
## Warning: Removed 4 rows containing missing values (geom_path).
```



Ira’s email

I’ve taken a different approach than Ira to illustrate a pure dplyr solution.

```
Jim,

Thanks for giving the talk.

Example 1. Every hour one of my workers shows up at 15 after the hour to take a BART train home.
Every hour two BART trains show up randomly in the interval [0,60). I wish to gather data on
wait times. The problem is when both trains arrive in [0,15), so I need data from the next hour.

There is also a problem with &

Thanks for your help.

Ira
---
> library(dplyr)
> over15 = function(x,y) {if( x > 15 & y > x) {res = x - 15}
+   else if(x > 15 & y < 15) {res = x - 15}
+   else if (y > 15) {res = y - 15}
+   else res = 500 # needs to be changed. In there to avoid error messages
+   return(res) }
> over15(20, 35)
[1] 5
> over15(8,37)
[1] 22
> over15(54, 28)
[1] 13
> over15(9,7) # The problem case in the data frame
```

```
[1] 500
> # row 3 is a problem
> waits = data.frame(Hour = c(1,2,3,4), Arrival = rep(15,4), Train1 = c(20, 8, 2, 5),
  Train2 = c(37,20,8,53))
> waits
  Hour Arrival Train1 Train2
1    1      15     20     37
2    2      15      8     20
3    3      15      2      8
4    4      15      5     53
> waits = tbl_df(waits)
> waits
Source: local data frame [4 x 4]

   Hour Arrival Train1 Train2
  (dbl)   (dbl)  (dbl)  (dbl)
1     1      15     20     37
2     2      15      8     20
3     3      15      2      8
4     4      15      5     53
> mutate(waits,
+        delay = over15(Train1, Train2))
Source: local data frame [4 x 5]

   Hour Arrival Train1 Train2 delay
  (dbl)   (dbl)  (dbl)  (dbl) (dbl)
1     1      15     20     37     5
2     2      15      8     20    -7
3     3      15      2      8   -13
4     4      15      5     53   -10
Warning message:
In if (x > 15 & y > x) { :
  the condition has length > 1 and only the first element will be used
> mapply(over15, waits$Train1, waits$Train2)
[1]   5   5 500  38
```