PATH 2020-06

Planning & Regional Development

5/29/2020

Copy and move files. First, dates. Second, inputs.

```
### PATH FORECASTING SCRIPT
# Reduce to 125 lines tops. Save relevant stuff. Zip final file. Send to all relevant parties.



## Set working drive
#setwd("S:/Current/REA - Economic and Activity Forecasts/Line Departments/PATH") # work
#setwd("C:/Users/ceshleman/Dropbox/Work and research/Port Authority/PA data & analysis/PA PATH")
setwd("~/Dropbox/Work and research/Port Authority/pathforecast")

cat("\014") # clear the console
```

```
rm(list=ls())
options(scipen=999)
dev.off()
```

```
## null device
##           1
```

```
library(broom)
library(knitr)
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(reshape2)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.2

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries)
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
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```r
library(doBy)
library(mice)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'mice'
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
library(lmtest)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:mice':
##
##     complete
```

```
## The following object is masked from 'package:reshape2':
##
##     smiths
```

```r
#library(yardstick)
#library(Hmisc)
set.seed(101)


start = "2004-01-01"
end = "2020-02-01" #"2019-12-01"
end_and_one = "2020-03-01" #"2020-01-01"
extra = as.Date(end_and_one)-as.Date(end)
future = "2040-12-31"

elapsed_months <- function(end_date, start_date) {
  ed <- as.POSIXlt(end_date)
  sd <- as.POSIXlt(start_date)
  12 * (ed$year - sd$year) + (ed$mon - sd$mon)
}
horizon = elapsed_months(future,start)+1
forec_horizon = elapsed_months(future,end)
forec_horizon
```

```
## [1] 250
```

```
(2040-2018)*12
```

```
## [1] 264
```

```
##############################################################################################
### LOAD DATA LOAD DATA LOAD DATA LOAD DATA LOAD DATA LOAD DATA LOAD DATA LOAD DATA LOAD DATA LOAD DATA
##############################################################################################

days = read.csv("./Dates_with_holidays.csv"); names(days) = c("month","weekdays","saturdays","sundays")
days$month = as.Date(days$month,format="%m/%d/%Y")


# New ridership variables for 2019-07-23 model run are avg_wkdayholminor_tstile avg_satholmajor_tstile
#path = read.csv("S:/Current/REA - Economic and Activity Forecasts/Line Departments/PATH/PATH forecast
#path = read.csv("./PA PATH input/PATH input 2019q4.csv") #path = read.csv("./PA PATH input/PATH input
path = read.csv("./PATH input 2020q1.csv")
head(path,3)
```

```
##    month week sat sun sat_alt sun_alt weekdays saturdays sundays dum_911_base
## 1 1/1/96   NA  NA  NA      NA      NA       21         4       6            0
## 2 2/1/96   NA  NA  NA      NA      NA       20         4       5            0
## 3 3/1/96   NA  NA  NA      NA      NA       21         5       5            0
##   supersandy summer_of_hell end_close mon dummy_1 dummy_2 dummy_3 dummy_4
## 1          0              0         0   1       1       0       0       0
## 2          0              0         0   2       0       1       0       0
## 3          0              0         0   3       0       0       1       0
##   dummy_5 dummy_6 dummy_7 dummy_8 dummy_9 dummy_10 dummy_11 dummy_12 pop_hudson
## 1       0       0       0       0       0        0        0        0   582.4916
## 2       0       0       0       0       0        0        0        0   583.0440
## 3       0       0       0       0       0        0        0        0   583.5642
##   real_fare sun_x_avg sat_x_avg week_x_avg  man_hud avg_wkdayholminor_tstile
## 1  1.635771        NA        NA         NA 2207.044                       NA
## 2  1.626886        NA        NA         NA 2214.710                       NA
## 3  1.619069        NA        NA         NA 2223.642                       NA
##   avg_satholmajor_tstile avg_sun_tstile man_hud_opt man_hud_pess
## 1                     NA             NA    2207.010     2207.044
## 2                     NA             NA    2218.606     2214.710
## 3                     NA             NA    2226.262     2223.642
##   population_hud_opt population_hud_pess real_fare_q1 num_wkdayholminor
## 1           582.4916            582.4916     1.698699                NA
## 2           583.0112            583.0112     1.698699                NA
## 3           583.5420            583.5420     1.695071                NA
##   num_satholmajor num_sun total_days  X X.1     X.2
## 1              NA      NA         NA NA  NA 165.374
## 2              NA      NA         NA NA  NA      NA
## 3              NA      NA         NA NA  NA      NA
```

```
path$month = as.Date(path$month,format="%m/%d/%y")

# Econ from Q3
#econ = read.csv("S:/Current/REA - Economic and Activity Forecasts/Line Departments/PATH/PATH forecast
# Econ from Q4 - opt and pess
```

```
##################################################################################################
### INTERPOLATE QUARTERS TO MONTHS
##################################################################################################

#done elsewhere

##################################################################################################
### PREP PREP PREP PREP PREP PREP PREP PREP
##################################################################################################

before = subset(path,path$month<=end & path$month>="2002-01-01")#path$month<=as.Date(end,format="%Y-%m-
after = subset(path,path$month>end)


##################################################################################################
### SPECIAL: WEEKEND CLOSURES    SPECIAL: WEEKEND CLOSURES    SPECIAL: WEEKEND CLOSURES    SPECIAL: WEEK
##################################################################################################

# another time

##################################################################################################
### CHOOSE MODEL COVARIATES CHOOSE MODEL COVARIATES CHOOSE MODEL COVARIATES CHOOSE MODEL COVARIATES CHOO
##################################################################################################
names(before)
```

```
##  [1] "month"                    "week"
##  [3] "sat"                      "sun"
##  [5] "sat_alt"                  "sun_alt"
##  [7] "weekdays"                 "saturdays"
##  [9] "sundays"                  "dum_911_base"
## [11] "supersandy"               "summer_of_hell"
## [13] "end_close"                "mon"
## [15] "dummy_1"                  "dummy_2"
## [17] "dummy_3"                  "dummy_4"
## [19] "dummy_5"                  "dummy_6"
## [21] "dummy_7"                  "dummy_8"
## [23] "dummy_9"                  "dummy_10"
## [25] "dummy_11"                 "dummy_12"
## [27] "pop_hudson"               "real_fare"
## [29] "sun_x_avg"                "sat_x_avg"
## [31] "week_x_avg"               "man_hud"
## [33] "avg_wkdayholminor_tstile" "avg_satholmajor_tstile"
## [35] "avg_sun_tstile"           "man_hud_opt"
## [37] "man_hud_pess"             "population_hud_opt"
## [39] "population_hud_pess"      "real_fare_q1"
## [41] "num_wkdayholminor"        "num_satholmajor"
## [43] "num_sun"                  "total_days"
## [45] "X"                        "X.1"
## [47] "X.2"
```

```
### WEEKDAYS
oldreg=as.matrix(data.frame(before$man_hud,
                            before$dummy_2,before$dummy_3,before$dummy_4,before$dummy_5,before$dum_911_
```

```
                             before$dummy_6,before$dummy_7,before$dummy_8,before$dummy_9,before$dummy_10
                             before$supersandy, before$real_fare_q1)) #real_fare_q4
newreg=as.matrix(data.frame(after$man_hud,
                             after$dummy_2,after$dummy_3,after$dummy_4,after$dummy_5, after$dum_911_base
                             after$dummy_6, after$dummy_7, after$dummy_8, after$dummy_9, after$dummy_10,
                             after$supersandy, after$real_fare_q1))
### SATURDAY & SUNDAY
oldregsat=as.matrix(data.frame(before$pop_hudson,before$dummy_2, before$dummy_3, before$dummy_4,before$d
                             before$dummy_5, before$dummy_6, before$dummy_7, before$dummy_8, before$dummy
                             before$dummy_12,before$supersandy, before$end_close,
                             before$real_fare_q1))
newregsat=as.matrix(data.frame(after$pop_hudson,after$dummy_2, after$dummy_3, after$dummy_4,after$dum_91
                             after$dummy_5, after$dummy_6, after$dummy_7, after$dummy_8, after$dummy_9, a
                             after$dummy_12,after$supersandy, after$end_close,
                             after$real_fare_q1))

t.test(before$sun,before$real_fare)
```

```
##
##  Welch Two Sample t-test
##
## data:  before$sun and before$real_fare
## t = 88.264, df = 208, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  77739.79 81291.87
## sample estimates:
##    mean of x    mean of y
## 79518.091244     2.262067
```

```
###############################################################################################
### MODELS MODELS MODELS MODELS MODELS MODELS MODELS MODELS MODELS MODELS MODELS MODELS MODELS MODELS M
###############################################################################################
names(before)
```

```
##  [1] "month"                      "week"
##  [3] "sat"                        "sun"
##  [5] "sat_alt"                    "sun_alt"
##  [7] "weekdays"                   "saturdays"
##  [9] "sundays"                    "dum_911_base"
## [11] "supersandy"                 "summer_of_hell"
## [13] "end_close"                  "mon"
## [15] "dummy_1"                    "dummy_2"
## [17] "dummy_3"                    "dummy_4"
## [19] "dummy_5"                    "dummy_6"
## [21] "dummy_7"                    "dummy_8"
## [23] "dummy_9"                    "dummy_10"
## [25] "dummy_11"                   "dummy_12"
## [27] "pop_hudson"                 "real_fare"
## [29] "sun_x_avg"                  "sat_x_avg"
## [31] "week_x_avg"                 "man_hud"
## [33] "avg_wkdayholminor_tstile" "avg_satholmajor_tstile"
## [35] "avg_sun_tstile"             "man_hud_opt"
## [37] "man_hud_pess"               "population_hud_opt"
## [39] "population_hud_pess"        "real_fare_q1"
## [41] "num_wkdayholminor"        "num_satholmajor"
## [43] "num_sun"                  "total_days"
## [45] "X"                        "X.1"
## [47] "X.2"
```

```
fit = arima(ts(before$avg_wkdayholminor_tstile),xreg = oldreg, order=c(0,0,1), include.mean=T)# as of 2
#fit = arima(ts(before$avg_wkdayholminor_tstile),xreg = oldreg, order=c(0,0,1), include.mean=T)# as of 
  #fit = auto.arima(ts(before$week),xreg=oldreg, ic="aic", trace=TRUE, allowdrift=FALSE)#,lambda=0, sea
fitsat = arima(ts(before$avg_satholmajor_tstile),xreg=oldregsat,order=c(1,1,0))# as of 2018-09 (1,1,0)
#fitsat = arima(ts(before$avg_satholmajor_tstile),xreg=oldregsat,order=c(1,1,0))# as of 2018-09 (1,1,0)
  #fitsat = auto.arima(ts(before$sat_mice),xreg=oldregsat, ic="aic", trace=TRUE, allowdrift=FALSE ,lamb
fitsun = arima(ts(before$avg_sun_tstile),xreg=oldregsat,order=c(1,1,1))# as of 2018-09 (1,1,1) before t
#fitsun = arima(ts(before$avg_sun_tstile),xreg=oldregsat,order=c(1,1,1))# as of 2018-09 (1,1,1) before 
  #fitsun = auto.arima(ts(before$sun_mice),xreg=oldregsat, ic="aic", trace=TRUE, allowdrift=FALSE,lambd

pathpredict = predict(fit, n.ahead=forec_horizon, newxreg=newreg, level=95)#interval = "prediction", co
pathpredictsat = predict(fitsat, n.ahead=forec_horizon, newxreg=newregsat) # predict
pathpredictsun = predict(fitsun, n.ahead=forec_horizon, newxreg=newregsat) # predict

pathpredict_by_month = as.data.frame(cbind(pathpredict$pred,pathpredictsat$pred,pathpredictsun$pred)); 
head(pathpredict_by_month,3)
```

```
##   week_avg  sat_avg  sun_avg
## 1 285468.9 122614.5 85339.92
## 2 295123.2 124630.2 91125.56
## 3 297174.7 116473.1 95440.08
```

```
end
```

```
## [1] "2020-02-01"
```

```
future
```

```
## [1] "2040-12-31"
```

```
pathpredict_by_month$month = seq(as.Date(end)+extra,as.Date(future),by="mon")

## Add old stuff (January 2017, for example) back to the pile.
before_mini = data.frame((before$week*before$weekdays),(before$sat*before$saturdays),(before$sun*before
  names(before_mini) = c("week","sat","sun","month")

## Now multiply by number of days per month ...
pathpredict_by_month = merge(pathpredict_by_month,days)
pathpredict_by_month$week = pathpredict_by_month$week_avg*pathpredict_by_month$weekdays
pathpredict_by_month$sat = pathpredict_by_month$sat_avg*pathpredict_by_month$saturdays
pathpredict_by_month$sun = pathpredict_by_month$sun_avg*pathpredict_by_month$sundays


pathpredict_mini = data.frame(pathpredict_by_month$month,pathpredict_by_month$week,pathpredict_by_month
  names(pathpredict_mini) = c("month","week","sat","sun")



###############################################################################################
### DIAGNOSTICS DIAGNOSTICS DIAGNOSTICS DIAGNOSTICS DIAGNOSTICS DIAGNOSTICS DIAGNOSTICS DIAG
###############################################################################################
```

```
out1 = tidy(fit)
  #out2 = tidy(glance(fit)) ## why is this crashing my program?
out2 = glance(fit)
out1
```

```
## # A tibble: 17 x 3
##    term                 estimate  std.error
##    <fct>                   <dbl>      <dbl>
##  1 ma1                     0.580     0.0445
##  2 intercept          -93038.    24308.
##  3 before.man_hud        148.       14.4
##  4 before.dummy_2       3041.     3037.
##  5 before.dummy_3       5303.     4077.
##  6 before.dummy_4      10500.     4098.
##  7 before.dummy_5      12678.     4094.
##  8 before.dum_911_base -51663.     4064.
##  9 before.dummy_6      15592.     4092.
## 10 before.dummy_7      11724.     4090.
## 11 before.dummy_8       5471.     4089.
## 12 before.dummy_9      16071.     4089.
## 13 before.dummy_10     13454.     4090.
## 14 before.dummy_11      7646.     4127.
## 15 before.dummy_12     -1310.     3143.
## 16 before.supersandy  -75225.     9904.
## 17 before.real_fare_q1 -18867.     6577.
```

```
out2
```

```
## # A tibble: 1 x 4
##    sigma logLik   AIC   BIC
##    <dbl>  <dbl> <dbl> <dbl>
## 1 10742. -2333. 4702. 4763.
```

```
out3 = tidy(fitsat)
out4 = glance(fitsat)
out3
```

```
## # A tibble: 17 x 3
##    term                 estimate std.error
##    <fct>                   <dbl>     <dbl>
##  1 ar1                    -0.543    0.0570
##  2 before.pop_hudson       566.     660.
##  3 before.dummy_2         5737.    1655.
##  4 before.dummy_3        19407.    1548.
##  5 before.dummy_4        20604.    1858.
##  6 before.dum_911_base   -3137.    5446.
##  7 before.dummy_5        12550.    1865.
##  8 before.dummy_6        19430.    1967.
##  9 before.dummy_7        15694.    1950.
## 10 before.dummy_8        14345.    1975.
## 11 before.dummy_9        15788.    1871.
## 12 before.dummy_10       19241.    1869.
## 13 before.dummy_11       14433.    1563.
## 14 before.dummy_12       15372.    1705.
## 15 before.supersandy    -44726.    4168.
## 16 before.end_close     -12883.    2126.
## 17 before.real_fare_q1 -24290.    9773.
```

```
out4
```

```
## # A tibble: 1 x 4
##   sigma logLik   AIC   BIC
##   <dbl>  <dbl> <dbl> <dbl>
## 1 6047. -2198. 4431. 4492.
```

```
out5 = tidy(fitsun)
out6 = glance(fitsun)
out5
```

```
## # A tibble: 18 x 3
##    term                    estimate std.error
##    <fct>                      <dbl>     <dbl>
##  1 ar1                      0.00277    0.0848
##  2 ma1                     -0.792      0.0495
##  3 before.pop_hudson      355.       184.
##  4 before.dummy_2        4689.      1396.
##  5 before.dummy_3        7492.      1434.
##  6 before.dummy_4       13136.      1446.
##  7 before.dum_911_base  -6636.      2969.
##  8 before.dummy_5       17311.      1451.
##  9 before.dummy_6       22577.      1454.
## 10 before.dummy_7       16303.      1454.
## 11 before.dummy_8       15224.      1457.
## 12 before.dummy_9       19731.      1453.
## 13 before.dummy_10      16550.      1447.
## 14 before.dummy_11      13354.      1448.
## 15 before.dummy_12      16047.      1436.
## 16 before.supersandy   -37263.      3647.
## 17 before.end_close    -10419.      1399.
## 18 before.real_fare_q1 -12366.      5536.
```

```
out6
```

```
## # A tibble: 1 x 4
##   sigma logLik   AIC   BIC
##   <dbl>  <dbl> <dbl> <dbl>
## 1 4757. -2146. 4330. 4394.
```

```
accuracy(fit)
```

```
##                     ME     RMSE      MAE        MPE     MAPE    MASE      ACF1
## Training set -22.19489 10741.57 8079.563 -0.3107499 3.626442 1.11551 0.2833142
```

```
accuracy(fit)[,'MAPE']
```

```
## [1] 3.626442
```

```
pathpredict_month_backup = pathpredict_by_month
pathpredict_month = rbind(before_mini,pathpredict_mini) #meh. figure this out later



##########################################################################################
### ANNUAL PREDICTIONS ANNUAL PREDICTIONS ANNUAL PREDICTIONS ANNUAL PREDICTIONS ANNUAL PREDICTIONS ANNU
##########################################################################################
```

```
pathpredict_month$year = year(pathpredict_month$month)

pathpredict_year = summaryBy(week + sat + sun ~ year, data = pathpredict_month, FUN = sum); names(pathp
pathpredict_year$total = pathpredict_year$week + pathpredict_year$sat + pathpredict_year$sun
pathpredict_month$year = NULL
years = pathpredict_year
pathpredict_year[14,5] = 76565451
pathpredict_year[15,5] = 78517120


resids = as.data.frame(cbind(as.vector(resid(fit)),as.vector(resid(fitsat)),as.vector(resid(fitsun))))


#####################################################################################################
### RESULTS EXPORT RESULTS EXPORT RESULTS EXPORT RESULTS EXPORT RESULTS EXPORT RESULTS EXPORT
#####################################################################################################

tail(pathpredict_by_month)
```

```
##          month week_avg  sat_avg  sun_avg weekdays saturdays sundays    week
## 245 2040-07-01 340663.5 156222.2 117295.0       22         5       4 7494598
## 246 2040-08-01 334631.9 154898.7 116232.1       21         5       5 7027271
## 247 2040-09-01 345450.5 156412.9 120783.9       21         5       4 7254461
## 248 2040-10-01 343049.7 159997.5 117684.9       22         5       4 7547093
## 249 2040-11-01 337242.1 155189.3 114489.7       19         6       5 6407601
## 250 2040-12-01 328285.7 156128.2 117182.3       22         5       4 7222285
##          sat      sun
## 245 781111.1 469180.1
## 246 774493.3 581160.6
## 247 782064.5 483135.5
## 248 799987.7 470739.6
## 249 931135.6 572448.6
## 250 780640.8 468729.2
```

```
tail(pathpredict_year)
```

```
##    year     week     sat     sun     total
## 34 2035 81626842 9227705 5860259  96714806
## 35 2036 82542049 9260605 5876074  97678728
## 36 2037 82911982 9288215 5893378  98093576
## 37 2038 83601403 9330707 5916005  98848116
## 38 2039 84294864 9373265 5938668  99606797
## 39 2040 84976639 9411679 5959232 100347550
```

```
#tail(pathpredict_by_month, 50)
getwd()
```

```
## [1] "/Users/chriseshleman/Dropbox/Work and research/Port Authority/pathforecast"
```

```
#write.csv(pathpredict_by_month,"./PATH forecast products/PATH forecast output/PATH q2/PATH month_ 2019
 write.csv(pathpredict_month_backup, "./PATH output test 20200529.csv")
#  write.csv(fitted(fit), "./PA PATH output & viz/PATH fitted _week 2020q1.csv")
#  write.csv(fitted(fitsat), "./PA PATH output & viz/PATH fitted _sat 2020q1.csv")
#  write.csv(fitted(fitsun), "./PA PATH output & viz/PATH fitted _sun 2020q1.csv")
#  write.csv(resids, "./PATH forecast products/PATH forecast output/PATH q2/PATH residuals _nodummy 201
```

```r
#write.csv(pathpredict_year,"./PA PATH output & viz/PATH ANNUAL_Q4 FARE TRNSTL RIDERS PESS.csv")
#  write.csv(pathpredict_by_month, "./PA PATH output & viz/PATH MONTH_2020Q1.csv")
```