

PATH 2020-06

Planning & Regional Development

5/29/2020

Copy and move input files - input variables and dates. DONE. Incorporate quarter-to-month interpolation. DONE. Bind dates, ridership, monthly economics, other. Add and update diagnostics. Confirm representns 2020 Q1 (?) forecast outputs. Package outputs (inputs, forecast and diagnostics). Clean code. Add narrative. Automate real fare calculation. Package (zip?) and share.

1. Setup setup setup.

```
setwd("~/Dropbox/Work and research/Port Authority/pathforecast")
```

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

```
rm(list=ls())
options(scipen=999)
```

```
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
library(lubridate)

##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
## date
library(doBy)
library(mice)

## Loading required package: lattice
##
## Attaching package: 'mice'
## The following objects are masked from 'package:base':
##
## cbind, rbind
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
library(zoo)
#library(yardstick)

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

```

2. Interpolation: convert quarterly data to monthly.

```
jobs = read.csv("./econ_vars_quar 2020_06.csv")
jobs1 = jobs#[complete.cases(jobs),]

# IF QUARTERS ARE 'AVERAGE' OF RELEVANT THREE MONTHS (define 'average' later)
jobs1$year=NULL
jobs1$quarter=NULL
jobs1$Indicator=NULL

jobs1$Month = as.Date(jobs1$Month, "%m/%d/%y")

jobs2 = jobs1
jobs2$quarter2 = NULL
jobs2 = read.zoo(jobs2) # Converts the data frame to a time series matrix

tt = as.yearmon(seq(start(jobs2), end(jobs2), "month")) # Makes months, different format (unsure why ne
jobs2$Indicator = NULL

zm = as.data.frame(na.spline(jobs2, as.yearmon, xout = tt))
zm$month_ = seq(as.Date("1996/1/1"), as.Date("2035/10/01"), by="month") # Add date

zm2 = subset(zm,zm$month_=="2035-10-01")
zm3 = subset(zm,zm$month_=="2035-10-01")
zm2$month_="2035-11-01"
zm3$month_="2035-12-01"
zm=rbind(zm,zm2)
zm=rbind(zm,zm3)
zm = zm[order(as.Date(zm$month_, format="%Y-%m-%d")),]

write.csv(zm,"./econ_vars_months 2020_06.csv")
jobs_month = zm
names(jobs_month) = tolower(names(jobs_month))

rm(jobs,jobs1,jobs2,tt,zm,zm2,zm3)
```

3. Load data load data.

```
days = read.csv("./Dates_dummies.csv")
#; names(days) = c("month", "weekdays", "saturdays", "sundays")
days$month = as.Date(days$month, "%m/%d/%y")

path = read.csv("./PATH input 2020_06.csv")
path$month = as.Date(days$month, "%m/%d/%y")

other = read.csv("./other.csv")
other$month = as.Date(other$month, "%m/%d/%y")

path = merge(days,path)#,by="month")
path = merge(path,other)#,by="month")
#path$month = as.Date(days$month,format="%m/%d/%y")
```

4.Prepare for model.

```
#before = subset(path,path$month<=end & path$month>="2002-01-01")
before = head(path,218)
#after = subset(path,path$month>end)
after = tail(path,250)
summary(after)
```

```
##      month      num_wkdayholminor num_satholmajor  num_sun
## Min.   :2020-03-01   Min.    :18           Min.    :4.000   Min.    :4.000
## 1st Qu.:2025-05-08   1st Qu.:20           1st Qu.:4.000   1st Qu.:4.000
## Median :2030-07-16   Median :21           Median :5.000   Median :4.000
## Mean   :2030-07-17   Mean    :21           Mean    :5.088   Mean    :4.348
## 3rd Qu.:2035-09-23   3rd Qu.:22           3rd Qu.:6.000   3rd Qu.:5.000
## Max.   :2040-12-01   Max.    :23           Max.    :7.000   Max.    :5.000
##
##      weekdays      saturdays      sundays      dum_911_base  supersandy
## Min.   :17.00   Min.    :4.00   Min.    :4.000   Min.    :0      Min.    :0
## 1st Qu.:20.00   1st Qu.:4.00   1st Qu.:5.000   1st Qu.:0      1st Qu.:0
## Median :21.00   Median :4.00   Median :5.000   Median :0      Median :0
## Mean   :20.86   Mean    :4.34   Mean    :5.256   Mean    :0      Mean    :0
## 3rd Qu.:22.00   3rd Qu.:5.00   3rd Qu.:6.000   3rd Qu.:0      3rd Qu.:0
## Max.   :23.00   Max.    :5.00   Max.    :8.000   Max.    :0      Max.    :0
##
##      summer_of_hell  end_close      mon      dummy_1      dummy_2
## Min.    :0          Min.    :0   Min.    : 1.00   Min.    :0.00   Min.    :0.00
## 1st Qu.:0          1st Qu.:0   1st Qu.: 4.00   1st Qu.:0.00   1st Qu.:0.00
## Median :0          Median :0   Median : 7.00   Median :0.00   Median :0.00
## Mean    :0          Mean    :0   Mean    : 6.54   Mean    :0.08   Mean    :0.08
## 3rd Qu.:0          3rd Qu.:0   3rd Qu.: 9.75   3rd Qu.:0.00   3rd Qu.:0.00
## Max.    :0          Max.    :0   Max.    :12.00   Max.    :1.00   Max.    :1.00
##
##      dummy_3      dummy_4      dummy_5      dummy_6
## Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
## 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000
## Median :0.000   Median :0.000   Median :0.000   Median :0.000
## Mean    :0.084   Mean    :0.084   Mean    :0.084   Mean    :0.084
## 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000
## Max.    :1.000   Max.    :1.000   Max.    :1.000   Max.    :1.000
##
##      dummy_7      dummy_8      dummy_9      dummy_10
## Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
## 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000
## Median :0.000   Median :0.000   Median :0.000   Median :0.000
## Mean    :0.084   Mean    :0.084   Mean    :0.084   Mean    :0.084
## 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000
## Max.    :1.000   Max.    :1.000   Max.    :1.000   Max.    :1.000
##
##      dummy_11      dummy_12      week      sat      sun
## Min.   :0.000   Min.   :0.000   Min.    : NA   Min.    : NA   Min.    : NA
## 1st Qu.:0.000   1st Qu.:0.000   1st Qu.: NA   1st Qu.: NA   1st Qu.: NA
## Median :0.000   Median :0.000   Median : NA   Median : NA   Median : NA
## Mean    :0.084   Mean    :0.084   Mean    :NaN   Mean    :NaN   Mean    :NaN
## 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.: NA   3rd Qu.: NA   3rd Qu.: NA
```

```

## Max. :1.000 Max. :1.000 Max. : NA Max. : NA Max. : NA
## NA's :250 NA's :250 NA's :250
## sat_alt sun_alt avg_wkdayholminor_tstile avg_satholmajor_tstile
## Min. : NA Min. : NA Min. : NA Min. : NA
## 1st Qu.: NA 1st Qu.: NA 1st Qu.: NA 1st Qu.: NA
## Median : NA Median : NA Median : NA Median : NA
## Mean :NaN Mean :NaN Mean :NaN Mean :NaN
## 3rd Qu.: NA 3rd Qu.: NA 3rd Qu.: NA 3rd Qu.: NA
## Max. : NA Max. : NA Max. : NA Max. : NA
## NA's :250 NA's :250 NA's :250 NA's :250
## avg_sun_tstile real_fare man_hud_KEEP man_hud_opt man_hud_pess
## Min. : NA Min. :2.75 Min. :2904 Min. :2899 Min. :2851
## 1st Qu.: NA 1st Qu.:2.75 1st Qu.:2947 1st Qu.:2937 1st Qu.:2869
## Median : NA Median :2.75 Median :3027 Median :3020 Median :2939
## Mean :NaN Mean :2.75 Mean :3037 Mean :3041 Mean :2954
## 3rd Qu.: NA 3rd Qu.:2.75 3rd Qu.:3122 3rd Qu.:3134 3rd Qu.:3032
## Max. : NA Max. :2.75 Max. :3213 Max. :3239 Max. :3120
## NA's :250
## population_hud_opt population_hud_pess real_fare_q1_KEEP pop_hudson_KEEP
## Min. :683.7 Min. :683.7 Min. :2.75 Min. :683.9
## 1st Qu.:709.5 1st Qu.:709.5 1st Qu.:2.75 1st Qu.:709.8
## Median :733.9 Median :733.9 Median :2.75 Median :733.8
## Mean :727.3 Mean :727.3 Mean :2.75 Mean :726.2
## 3rd Qu.:743.9 3rd Qu.:743.9 3rd Qu.:2.75 3rd Qu.:743.6
## Max. :756.4 Max. :756.4 Max. :2.75 Max. :749.5
##
### WEEKDAYS
oldreg=as.matrix(data.frame(before$man_hud_KEEP, before$dummy_2,before$dummy_3,
before$dummy_6,before$dummy_7,before$dummy_8,before$dummy_9,before$dummy_10,before$dummy_11,
before$dummies, before$supersandy, before$real_fare_q1_KEEP)) #real_fare_q4
newreg=as.matrix(data.frame(after$man_hud_KEEP, after$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_KEEP))

### SATURDAY & SUNDAY
oldregsat=as.matrix(data.frame(before$pop_hudson_KEEP,before$dummy_2, before$dummy_3, before$dummy_4,before$dummy_5,
before$dummy_6, before$dummy_7, before$dummy_8, before$dummy_9, before$dummy_10, before$dummy_11,
before$dummy_12,before$supersandy, before$end_close, before$real_fare_q1_KEEP))
newregsat=as.matrix(data.frame(after$pop_hudson_KEEP,after$dummy_2, after$dummy_3, after$dummy_4,after$dum_911_base,
after$dummy_5, after$dummy_6, after$dummy_7, after$dummy_8, after$dummy_9, after$dum_911_base,
after$dummy_12,after$supersandy, after$end_close, after$real_fare_q1_KEEP))

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

```

5. Models and forecasts.

5a. Fit models (estimate equations).

Weekday

```
summary(before)
```

```
##      month      num_wkdayholminor num_satholmajor  num_sun
## Min.   :2000-01-01  Min.   :18.00      Min.   :4.000  Min.   :4.000
## 1st Qu.:2004-07-08  1st Qu.:20.00      1st Qu.:4.000  1st Qu.:4.000
## Median :2009-01-16  Median :21.00      Median :5.000  Median :4.000
## Mean   :2009-01-15  Mean   :20.98      Mean   :5.101  Mean   :4.349
## 3rd Qu.:2013-07-24  3rd Qu.:22.00      3rd Qu.:6.000  3rd Qu.:5.000
## Max.   :2018-02-01  Max.   :23.00      Max.   :7.000  Max.   :5.000
##
##      weekdays      saturdays      sundays      dum_911_base
## Min.   :18.00  Min.   :4.00  Min.   :4.000  Min.   :0.000
## 1st Qu.:20.00  1st Qu.:4.00  1st Qu.:4.000  1st Qu.:0.000
## Median :21.00  Median :5.00  Median :4.000  Median :0.000
## Mean   :21.25  Mean   :4.83  Mean   :4.349  Mean   :0.133
## 3rd Qu.:22.00  3rd Qu.:5.00  3rd Qu.:5.000  3rd Qu.:0.000
## Max.   :23.00  Max.   :7.00  Max.   :5.000  Max.   :1.000
##
##      supersandy      summer_of_hell      end_close      mon
## Min.   :0.000000  Min.   :0.000000  Min.   :0.0000  Min.   : 1.000
## 1st Qu.:0.000000  1st Qu.:0.000000  1st Qu.:0.0000  1st Qu.: 3.000
## Median :0.000000  Median :0.000000  Median :0.0000  Median : 6.000
## Mean   :0.009174  Mean   :0.009174  Mean   :0.1422  Mean   : 6.454
## 3rd Qu.:0.000000  3rd Qu.:0.000000  3rd Qu.:0.0000  3rd Qu.: 9.000
## Max.   :1.000000  Max.   :1.000000  Max.   :1.0000  Max.   :12.000
##
##      dummy_1      dummy_2      dummy_3      dummy_4
## Min.   :0.00000  Min.   :0.00000  Min.   :0.00000  Min.   :0.00000
## 1st Qu.:0.00000  1st Qu.:0.00000  1st Qu.:0.00000  1st Qu.:0.00000
## Median :0.00000  Median :0.00000  Median :0.00000  Median :0.00000
## Mean   :0.08716  Mean   :0.08716  Mean   :0.08257  Mean   :0.08257
## 3rd Qu.:0.00000  3rd Qu.:0.00000  3rd Qu.:0.00000  3rd Qu.:0.00000
## Max.   :1.00000  Max.   :1.00000  Max.   :1.00000  Max.   :1.00000
##
##      dummy_5      dummy_6      dummy_7      dummy_8
## Min.   :0.00000  Min.   :0.00000  Min.   :0.00000  Min.   :0.00000
## 1st Qu.:0.00000  1st Qu.:0.00000  1st Qu.:0.00000  1st Qu.:0.00000
## Median :0.00000  Median :0.00000  Median :0.00000  Median :0.00000
## Mean   :0.08257  Mean   :0.08257  Mean   :0.08257  Mean   :0.08257
## 3rd Qu.:0.00000  3rd Qu.:0.00000  3rd Qu.:0.00000  3rd Qu.:0.00000
## Max.   :1.00000  Max.   :1.00000  Max.   :1.00000  Max.   :1.00000
##
##      dummy_9      dummy_10      dummy_11      dummy_12
## Min.   :0.00000  Min.   :0.00000  Min.   :0.00000  Min.   :0.00000
## 1st Qu.:0.00000  1st Qu.:0.00000  1st Qu.:0.00000  1st Qu.:0.00000
## Median :0.00000  Median :0.00000  Median :0.00000  Median :0.00000
## Mean   :0.08257  Mean   :0.08257  Mean   :0.08257  Mean   :0.08257
```

```
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000
##
## week sat sun sat_alt
## Min. :121710 Min. : 59748 Min. : 37912 Min. : 70082
## 1st Qu.:213119 1st Qu.: 93399 1st Qu.: 68720 1st Qu.: 90758
## Median :245792 Median :107246 Median : 80691 Median :107732
## Mean :234065 Mean :104812 Mean : 79418 Mean :105129
## 3rd Qu.:258834 3rd Qu.:117278 3rd Qu.: 90155 3rd Qu.:118358
## Max. :293286 Max. :137725 Max. :105328 Max. :137725
## NA's :24 NA's :24 NA's :24 NA's :55
## sun_alt avg_wkdayholminor_tstile avg_satholmajor_tstile
## Min. : 49634 Min. :121710 Min. : 59748
## 1st Qu.: 67950 1st Qu.:216286 1st Qu.: 92275
## Median : 81866 Median :247566 Median :104762
## Mean : 81190 Mean :235284 Mean :103386
## 3rd Qu.: 91718 3rd Qu.:259011 3rd Qu.:115096
## Max. :123865 Max. :293286 Max. :137725
## NA's :55
## avg_sun_tstile real_fare man_hud_KEEP man_hud_opt man_hud_pess
## Min. : 37912 Min. :1.446 Min. :2316 Min. :2316 Min. :2316
## 1st Qu.: 67867 1st Qu.:1.938 1st Qu.:2377 1st Qu.:2377 1st Qu.:2377
## Median : 78785 Median :2.053 Median :2469 Median :2469 Median :2469
## Mean : 78539 Mean :2.150 Mean :2496 Mean :2496 Mean :2496
## 3rd Qu.: 89746 3rd Qu.:2.363 3rd Qu.:2563 3rd Qu.:2563 3rd Qu.:2563
## Max. :105328 Max. :2.869 Max. :2834 Max. :2833 Max. :2834
##
## population_hud_opt population_hud_pess real_fare_q1_KEEP pop_hudson_KEEP
## Min. :607.8 Min. :607.8 Min. :1.508 Min. :607.8
## 1st Qu.:615.2 1st Qu.:615.2 1st Qu.:2.006 1st Qu.:615.2
## Median :625.0 Median :625.0 Median :2.142 Median :625.0
## Mean :635.2 Mean :635.2 Mean :2.241 Mean :635.2
## 3rd Qu.:657.9 3rd Qu.:657.9 3rd Qu.:2.471 3rd Qu.:657.9
## Max. :675.2 Max. :675.2 Max. :2.986 Max. :675.2
##
```

```
fit = arima(ts(before$avg_wkdayholminor_tstile),xreg = oldreg, order=c(0,0,1), include.mean=T)#method="
#fit = arima(ts(before$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
```

Saturday

```
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
```

Sunday

```
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
```

5b. Predict (forecast).

```
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
```

6. Clean and consolidate results for export.

```
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 297566.5 123671.0 84750.51
## 2 301544.8 124449.9 90117.51
## 3 303780.6 116634.6 95142.09
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$sun*before$sundays),
  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$sat,pathpredict_by_month$sun)
names(pathpredict_mini) = c("month","week","sat","sun")
```

7. Model diagnostics.

```
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                  0.593      0.0432
## 2 intercept          -132264.    28012.
## 3 before.man_hud_KEEP    165.       14.0
```



```
## 4 before.dummy_2      3299.    3020.
## 5 before.dummy_3      6553.    4069.
## 6 before.dummy_4      10829.   4099.
## 7 before.dummy_5      13206.   4095.
## 8 before.dum_911_base -48079.   3836.
## 9 before.dummy_6      16322.   4092.
## 10 before.dummy_7     12287.   4089.
## 11 before.dummy_8      6896.   4088.
## 12 before.dummy_9     16711.   4080.
## 13 before.dummy_10    14442.   4079.
## 14 before.dummy_11     9607.   4116.
## 15 before.dummy_12     1129.   3118.
## 16 before.supersandy  -77070.   9873.
## 17 before.real_fare_q1_KEEP -20760.  4626.
```

```
out2
```

```
## # A tibble: 1 x 4
##   sigma logLik  AIC  BIC
##   <dbl> <dbl> <dbl> <dbl>
## 1 10665. -2331. 4699. 4760.
```

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

```
## # A tibble: 17 x 3
##   term                estimate std.error
##   <fct>              <dbl>    <dbl>
## 1 ar1                -0.516    0.0580
## 2 before.pop_hudson_KEEP    528.    666.
## 3 before.dummy_2          5344.   1635.
## 4 before.dummy_3         18704.   1575.
## 5 before.dummy_4         19213.   1886.
## 6 before.dum_911_base     -6226.   3964.
## 7 before.dummy_5         11224.   1913.
## 8 before.dummy_6         18160.   2002.
## 9 before.dummy_7         15350.   1993.
## 10 before.dummy_8         13968.   2005.
## 11 before.dummy_9         15178.   1898.
## 12 before.dummy_10        18966.   1871.
## 13 before.dummy_11        14681.   1600.
## 14 before.dummy_12        13532.   1693.
## 15 before.supersandy     -43005.   4239.
## 16 before.end_close      -13054.   3005.
## 17 before.real_fare_q1_KEEP -15831.   6221.
```

```
out4
```

```
## # A tibble: 1 x 4
##   sigma logLik  AIC  BIC
##   <dbl> <dbl> <dbl> <dbl>
## 1 6127. -2200. 4437. 4498.
```

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

```
## # A tibble: 18 x 3
##   term                estimate std.error
##   <fct>              <dbl>    <dbl>
## 1 ar1                0.0209    0.0873
## 2 ma1               -0.789     0.0528
## 3 before.pop_hudson_KEEP    314.     186.
## 4 before.dummy_2          4388.    1362.
## 5 before.dummy_3          7420.    1412.
## 6 before.dummy_4         12568.    1437.
## 7 before.dum_911_base     -7081.    2167.
## 8 before.dummy_5         17468.    1440.
## 9 before.dummy_6         21697.    1441.
## 10 before.dummy_7         15889.    1440.
## 11 before.dummy_8         14965.    1440.
## 12 before.dummy_9         19523.    1431.
## 13 before.dummy_10        16775.    1423.
## 14 before.dummy_11        13045.    1430.
## 15 before.dummy_12        15156.    1405.
## 16 before.supersandy     -36393.    3611.
## 17 before.end_close      -9678.    1686.
## 18 before.real_fare_q1_KEEP -8031.    3620.
```

```
out6
```

```
## # A tibble: 1 x 4
##   sigma logLik  AIC  BIC
##   <dbl> <dbl> <dbl> <dbl>
## 1 4695. -2143. 4324. 4388.
```

```
accuracy(fit)
```

```
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 16.04865 10665.16 8040.24 -0.295112 3.651406 1.161673 0.3086743
```

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

```
## [1] 3.651406
```

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

```
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))))
```

8. Save (export).

```
#####  
### RESULTS EXPORT RESULTS EXPORT RESULTS EXPORT RESULTS EXPORT RESULTS EXPORT  
#####
```

```
tail(pathpredict_by_month)
```

```
##          month week_avg  sat_avg  sun_avg num_wkdayholminor num_satholmajor  
## 245 2040-07-01 352532.5 154785.0 113793.9                22                5  
## 246 2040-08-01 347388.9 153425.9 112884.1                21                5  
## 247 2040-09-01 357448.2 154702.6 117482.0                21                5  
## 248 2040-10-01 355421.1 158613.4 114806.9                22                5  
## 249 2040-11-01 350586.0 154328.0 111076.9                19                6  
## 250 2040-12-01 342107.2 153178.5 113187.4                22                5  
##      num_sun weekdays saturdays sundays dum_911_base supersandy summer_of_hell  
## 245      4      22      4      5      0      0      0  
## 246      5      22      4      5      0      0      0  
## 247      4      20      5      5      0      0      0  
## 248      4      22      4      5      0      0      0  
## 249      5      18      4      8      0      0      0  
## 250      4      21      5      5      0      0      0  
##      end_close mon dummy_1 dummy_2 dummy_3 dummy_4 dummy_5 dummy_6 dummy_7  
## 245      0      7      0      0      0      0      0      0      1  
## 246      0      8      0      0      0      0      0      0      0  
## 247      0      9      0      0      0      0      0      0      0  
## 248      0     10      0      0      0      0      0      0      0  
## 249      0     11      0      0      0      0      0      0      0  
## 250      0     12      0      0      0      0      0      0      0  
##      dummy_8 dummy_9 dummy_10 dummy_11 dummy_12      week      sat      sun  
## 245      0      0      0      0      0  0 7755715 619140.1 568969.4  
## 246      1      0      0      0      0  0 7642556 613703.8 564420.7  
## 247      0      1      0      0      0  0 7148964 773513.1 587410.2  
## 248      0      0      1      0      0  0 7819264 634453.4 574034.3  
## 249      0      0      0      1      0  0 6310549 617312.1 888615.3  
## 250      0      0      0      0      0  1 7184251 765892.3 565937.0
```

```
tail(pathpredict_year)
```

```
##      year      week      sat      sun      total  
## 35 2035 83638001 7815338 6875322 98328661  
## 36 2036 84618046 7848909 6899548 99366503  
## 37 2037 85382252 7882537 6923814 100188603  
## 38 2038 86149678 7916220 6948120 101014018  
## 39 2039 86921601 7949955 6972464 101844020  
## 40 2040 87681301 7980303 6994216 102655819
```

```
#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.

#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")
```

Did the final year of output total 100,347,550?