#TBATS Model

output: pdf_document: default html_document: default —

```
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(ggplot2)
library(TTR)
library(imputeTS)
```

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

```
library(forecast)
```

```
#read in the interpolated data
data <- readr::read_csv(file = 'data/data_shipping_ca_az_or_interpolated.csv')
```

```
## Warning: Missing column names filled in: 'X1' [1]

##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   Mode = col_character(),
##   ORegionDAT = col_character(),
##   DRegionDAT = col_character(),
##   yw = col_character(),
##   sanitized_cost = col_double(),
##   approx_cost = col_double()
## )
```

```
# filter to just refridered trucks lane FRS to CHI
df_R_IL <- data %>%
  filter(data$Mode == "R", data$DRegionDAT == "IL_CHI", data$ORegionDAT == "CA_FRS")

tail(df_R_IL)
```

```
## # A tibble: 6 x 7
##      X1 Mode  ORegionDAT DRegionDAT yw       sanitized_cost approx_cost
##   <dbl> <chr> <chr>      <chr>      <chr>             <dbl>       <dbl>
## 1   230 R     CA_FRS     IL_CHI     2021 W21           3.13        3.13
## 2   231 R     CA_FRS     IL_CHI     2021 W22           3.11        3.11
## 3   232 R     CA_FRS     IL_CHI     2021 W23           3.20        3.20
## 4   233 R     CA_FRS     IL_CHI     2021 W24           3.10        3.10
## 5   234 R     CA_FRS     IL_CHI     2021 W25           3.10        3.10
## 6   235 R     CA_FRS     IL_CHI     2021 W26           3.15        3.15
```

```
#convert df_R_IL to time series to use autolayer when plotting later
ts <- ts(df_R_IL$approx_cost,frequency = 52,start = c(2017,01),end = c(2021,26))

ts
```

```
## Time Series:
## Start = c(2017, 1)
## End = c(2021, 26)
## Frequency = 52
##   [1] 1.565401 1.574481 1.547267 1.490165 1.462800 1.450093 1.420317 1.520521
##   [9] 1.402882 1.441905 1.465706 1.461992 1.473950 1.576948 1.656571 1.685428
##  [17] 1.657013 1.617382 1.722793 1.734288 1.950231 1.951489 2.153641 2.049965
##  [25] 1.951086 1.930969 1.944457 1.843359 1.834456 1.840848 1.781914 1.771574
##  [33] 1.747427 1.723280 1.752998 1.748400 1.761150 1.773900 1.773311 1.845116
##  [41] 1.793181 1.730538 1.712662 1.755339 1.830673 2.016542 1.896090 1.894823
##  [49] 1.969136 1.934059 2.090832 2.111540 2.173636 2.104403 1.947692 1.816810
##  [57] 1.641852 1.590409 1.619490 1.680216 1.607894 1.614384 1.620874 1.596928
##  [65] 1.756107 1.910363 1.966787 2.075901 2.096956 2.170187 2.214185 2.263016
##  [73] 2.419380 2.795106 2.666169 2.587869 2.611045 2.556201 2.343486 2.306517
##  [81] 2.288575 2.194607 2.192388 2.144689 2.098797 2.137834 2.159540 2.118428
##  [89] 2.094663 2.135035 2.115321 2.075720 2.033471 2.017613 2.024695 2.095456
##  [97] 2.211883 2.047605 2.007062 2.055710 2.013495 1.942834 1.945276 1.952673
## [105] 1.893033 1.739078 1.577270 1.642625 1.635440 1.570025 1.472177 1.465623
## [113] 1.501438 1.502210 1.462346 1.652368 1.698370 1.768270 1.891327 1.866554
## [121] 1.923215 1.888436 1.888714 1.877327 1.907018 2.030244 2.003457 2.040700
## [129] 2.048048 2.053496 1.989254 1.960468 1.955139 1.938055 1.892161 1.858698
## [137] 1.842677 1.826113 1.832473 1.809506 1.790416 1.750327 1.738252 1.738057
## [145] 1.629884 1.627459 1.665784 1.707958 1.819812 1.855762 1.904261 1.825806
## [153] 1.747352 1.807305 1.811109 1.807615 1.826484 1.737479 1.635804 1.603633
## [161] 1.587081 1.467021 1.428272 1.433854 1.506975 1.574007 1.772860 1.746883
## [169] 1.597605 1.617973 1.624631 1.634802 1.754427 1.930885 1.943768 2.138672
## [177] 2.132989 2.124347 2.072462 2.082013 2.078068 2.089675 2.062370 2.094305
## [185] 2.204175 2.235686 2.173398 2.133494 2.107996 2.092221 2.146061 2.139121
## [193] 2.115704 2.064587 2.152277 2.131431 2.222024 2.368182 2.267157 2.286262
## [201] 2.379540 2.457909 2.332682 2.197172 2.160951 2.105582 2.125096 2.123742
## [209] 2.122387 2.081140 2.080354 1.950824 2.002137 1.975779 1.977317 2.073338
## [217] 2.200157 2.338675 2.511865 2.495905 2.546230 2.625556 2.571481 2.708806
## [225] 2.753332 2.716657 3.045382 2.956507 3.012633 3.126758 3.107483 3.198408
## [233] 3.097934 3.101859
```

```
#Here we want to use imputeTS. This imputation algorithm fills in all missing values in a time series.
ts= na_interpolation(ts)
ts
```

```
## Time Series:
## Start = c(2017, 1)
## End = c(2021, 26)
## Frequency = 52
##   [1] 1.565401 1.574481 1.547267 1.490165 1.462800 1.450093 1.420317 1.520521
##   [9] 1.402882 1.441905 1.465706 1.461992 1.473950 1.576948 1.656571 1.685428
##  [17] 1.657013 1.617382 1.722793 1.734288 1.950231 1.951489 2.153641 2.049965
##  [25] 1.951086 1.930969 1.944457 1.843359 1.834456 1.840848 1.781914 1.771574
##  [33] 1.747427 1.723280 1.752998 1.748400 1.761150 1.773900 1.773311 1.845116
##  [41] 1.793181 1.730538 1.712662 1.755339 1.830673 2.016542 1.896090 1.894823
##  [49] 1.969136 1.934059 2.090832 2.111540 2.173636 2.104403 1.947692 1.816810
##  [57] 1.641852 1.590409 1.619490 1.680216 1.607894 1.614384 1.620874 1.596928
##  [65] 1.756107 1.910363 1.966787 2.075901 2.096956 2.170187 2.214185 2.263016
##  [73] 2.419380 2.795106 2.666169 2.587869 2.611045 2.556201 2.343486 2.306517
##  [81] 2.288575 2.194607 2.192388 2.144689 2.098797 2.137834 2.159540 2.118428
##  [89] 2.094663 2.135035 2.115321 2.075720 2.033471 2.017613 2.024695 2.095456
##  [97] 2.211883 2.047605 2.007062 2.055710 2.013495 1.942834 1.945276 1.952673
## [105] 1.893033 1.739078 1.577270 1.642625 1.635440 1.570025 1.472177 1.465623
## [113] 1.501438 1.502210 1.462346 1.652368 1.698370 1.768270 1.891327 1.866554
## [121] 1.923215 1.888436 1.888714 1.877327 1.907018 2.030244 2.003457 2.040700
## [129] 2.048048 2.053496 1.989254 1.960468 1.955139 1.938055 1.892161 1.858698
## [137] 1.842677 1.826113 1.832473 1.809506 1.790416 1.750327 1.738252 1.738057
## [145] 1.629884 1.627459 1.665784 1.707958 1.819812 1.855762 1.904261 1.825806
## [153] 1.747352 1.807305 1.811109 1.807615 1.826484 1.737479 1.635804 1.603633
## [161] 1.587081 1.467021 1.428272 1.433854 1.506975 1.574007 1.772860 1.746883
## [169] 1.597605 1.617973 1.624631 1.634802 1.754427 1.930885 1.943768 2.138672
## [177] 2.132989 2.124347 2.072462 2.082013 2.078068 2.089675 2.062370 2.094305
## [185] 2.204175 2.235686 2.173398 2.133494 2.107996 2.092221 2.146061 2.139121
## [193] 2.115704 2.064587 2.152277 2.131431 2.222024 2.368182 2.267157 2.286262
## [201] 2.379540 2.457909 2.332682 2.197172 2.160951 2.105582 2.125096 2.123742
## [209] 2.122387 2.081140 2.080354 1.950824 2.002137 1.975779 1.977317 2.073338
## [217] 2.200157 2.338675 2.511865 2.495905 2.546230 2.625556 2.571481 2.708806
## [225] 2.753332 2.716657 3.045382 2.956507 3.012633 3.126758 3.107483 3.198408
## [233] 3.097934 3.101859
```

```
#create tbats model and forecast
tbats_mod <- tbats(train_1)
tbats_for = forecast(tbats_mod, h=12)
```

```
#plot forecast with original data autolayer
autoplot(tbats_for) +
  autolayer(ts, color = "BLACK") +
  labs(
    y = "Approximate Cost",
    title = "Forecasts for weekly cost (CA_FRS to IL_CHI)"
  )
```

```
#here we can summarize to see MAPE
summary(tbats_for)
```

```
#create forecasting function
tbats_for <- function(x, h) {
  forecast(tbats(x), h =h)
}
```

```r
#CV of time series using our tbats forecast function
cross_validation <- tsCV(ts, forecastfunction = tbats_for, h = 12, initial = 155)

cross_validation[,12]
```

```
## Time Series:
## Start = c(2017, 1)
## End = c(2021, 26)
## Frequency = 52
##   [1]           NA           NA           NA           NA           NA
##   [6]           NA           NA           NA           NA           NA
##  [11]           NA           NA           NA           NA           NA
##  [16]           NA           NA           NA           NA           NA
##  [21]           NA           NA           NA           NA           NA
##  [26]           NA           NA           NA           NA           NA
##  [31]           NA           NA           NA           NA           NA
##  [36]           NA           NA           NA           NA           NA
##  [41]           NA           NA           NA           NA           NA
##  [46]           NA           NA           NA           NA           NA
##  [51]           NA           NA           NA           NA           NA
##  [56]           NA           NA           NA           NA           NA
##  [61]           NA           NA           NA           NA           NA
##  [66]           NA           NA           NA           NA           NA
##  [71]           NA           NA           NA           NA           NA
##  [76]           NA           NA           NA           NA           NA
##  [81]           NA           NA           NA           NA           NA
##  [86]           NA           NA           NA           NA           NA
##  [91]           NA           NA           NA           NA           NA
##  [96]           NA           NA           NA           NA           NA
## [101]           NA           NA           NA           NA           NA
## [106]           NA           NA           NA           NA           NA
## [111]           NA           NA           NA           NA           NA
## [116]           NA           NA           NA           NA           NA
## [121]           NA           NA           NA           NA           NA
## [126]           NA           NA           NA           NA           NA
## [131]           NA           NA           NA           NA           NA
## [136]           NA           NA           NA           NA           NA
## [141]           NA           NA           NA           NA           NA
## [146]           NA           NA           NA           NA           NA
## [151]           NA           NA           NA           NA           NA
## [156]  0.269125204 -0.034392467 -0.078901860 -0.110067290 -0.196438377
## [161] -0.088709648  0.139495054  0.155078968  0.248861092  0.072257526
## [166] -0.120102779 -0.328458770 -0.360297832 -0.097316588  0.075395647
## [171]  0.160716652  0.256313234  0.360945045  0.287893375  0.290484928
## [176]  0.175857548  0.213735463  0.247869630  0.290735722  0.294710862
## [181]  0.263249636  0.203044341  0.382503746  0.319425578  0.317699613
## [186]  0.484628138  0.288043204  0.309942476  0.334841809  0.352190870
## [191]  0.179274688  0.067489129  0.085920366 -0.180053990 -0.225883886
## [196] -0.131726464 -0.123841598 -0.006018628  0.075380668  0.017011514
## [201] -0.304608120  0.194072036  0.157922251  0.345266652  0.474021583
## [206]  0.532199806  0.740502380  0.662760958  0.649620559  0.668469590
## [211]  0.661237038  0.618938070  0.623616622  0.358964946  0.516112144
## [216]  0.203038512  0.306068954 -0.009751871  0.206000715 -0.052941328
```

```
## [221] -0.042087215  0.025977975           NA           NA           NA
## [226]           NA           NA           NA           NA           NA
## [231]           NA           NA           NA           NA
```

```
summary(cross_validation)
```

```
##       h=1                h=2                h=3                h=4
##  Min.   :-0.21409   Min.   :-0.26671   Min.   :-0.33004   Min.   :-0.37742
##  1st Qu.:-0.03191   1st Qu.:-0.03291   1st Qu.:-0.02261   1st Qu.:-0.05016
##  Median : 0.02213   Median : 0.06707   Median : 0.06658   Median : 0.09086
##  Mean   : 0.02365   Mean   : 0.04419   Mean   : 0.06274   Mean   : 0.07606
##  3rd Qu.: 0.07910   3rd Qu.: 0.11533   3rd Qu.: 0.18078   3rd Qu.: 0.19419
##  Max.   : 0.23011   Max.   : 0.36951   Max.   : 0.39578   Max.   : 0.48227
##  NA's   :156        NA's   :157        NA's   :158        NA's   :159
##       h=5                h=6                h=7                h=8
##  Min.   :-0.46971   Min.   :-0.48900   Min.   :-0.51151   Min.   :-0.49623
##  1st Qu.:-0.03029   1st Qu.:-0.05009   1st Qu.:-0.04264   1st Qu.:-0.03255
##  Median : 0.09497   Median : 0.11871   Median : 0.11593   Median : 0.07997
##  Mean   : 0.08619   Mean   : 0.09792   Mean   : 0.10927   Mean   : 0.12546
##  3rd Qu.: 0.22803   3rd Qu.: 0.23787   3rd Qu.: 0.24231   3rd Qu.: 0.27595
##  Max.   : 0.60486   Max.   : 0.62007   Max.   : 0.66463   Max.   : 0.78510
##  NA's   :160        NA's   :161        NA's   :162        NA's   :163
##       h=9                h=10               h=11               h=12
##  Min.   :-0.47723   Min.   :-0.38859   Min.   :-0.42612   Min.   :-0.36030
##  1st Qu.:-0.02957   1st Qu.:-0.02071   1st Qu.:-0.01405   1st Qu.:-0.00789
##  Median : 0.12928   Median : 0.16171   Median : 0.19128   Median : 0.20304
##  Mean   : 0.14020   Mean   : 0.15949   Mean   : 0.17619   Mean   : 0.18915
##  3rd Qu.: 0.27025   3rd Qu.: 0.27152   3rd Qu.: 0.31772   3rd Qu.: 0.32713
##  Max.   : 0.72305   Max.   : 0.72908   Max.   : 0.75576   Max.   : 0.74050
##  NA's   :164        NA's   :165        NA's   :166        NA's   :167
```

```
#calc error stats for when h=12 by writing forecast and findind its accuracy
accuracy(cross_validation[,12]+ts, ts)
```

```
##                    ME      RMSE     MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.1891539 0.3213088 0.26353 -9.174402 13.11715 0.8219962  2.025622
```