```
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_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(),
##   prcp = col_double(),
##   tavg = col_double(),
##   tmax = col_double(),
##   tmin = col_double(),
##   approx_cost = col_double()
## )
```

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

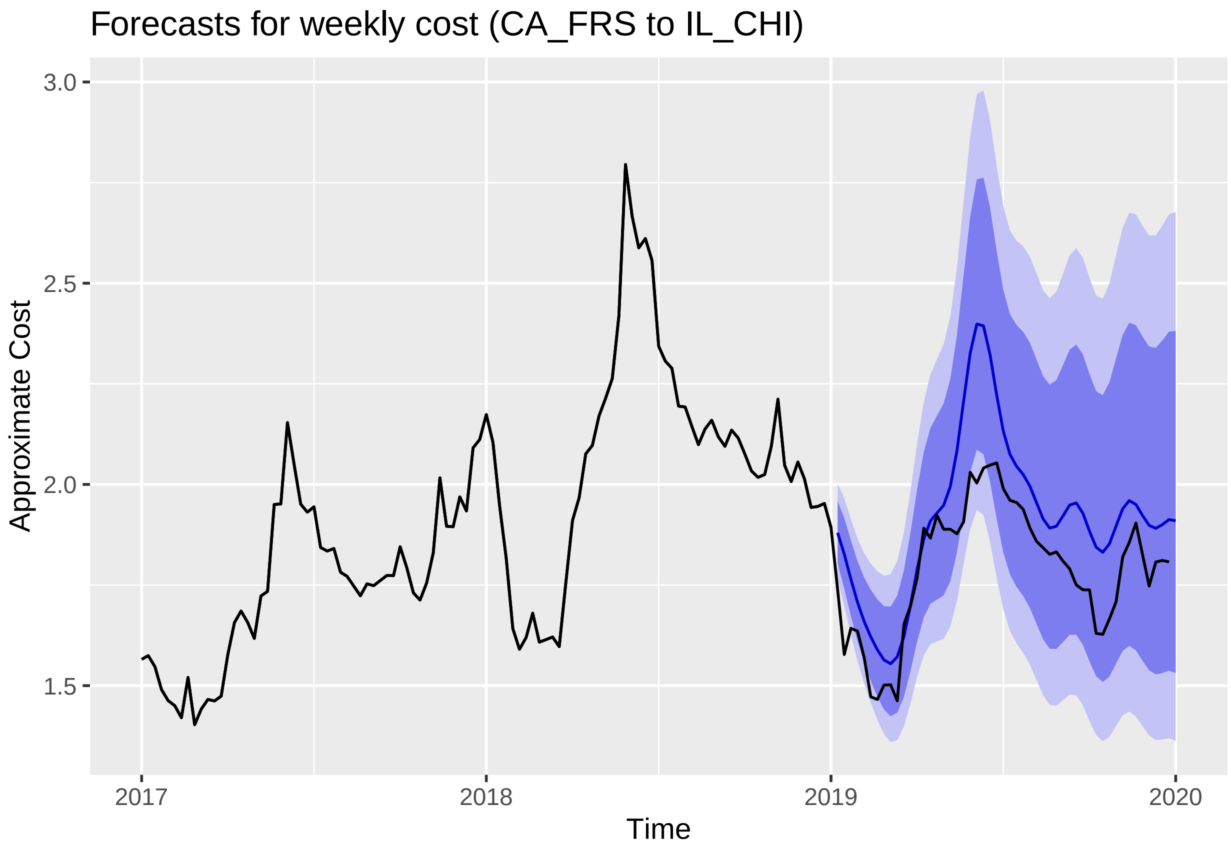
```
# convert
train <- ts(df_R_IL$approx_cost,frequency = 52,start = c(2017,01),end = c(2019,01))

#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(2019,52))
```

```
#Here we want to use imputeTS. This imputation algorithm fills in all missing values in a time series.
train_1= na_interpolation(train)
```

```
#create tbats model and forecast
tbats_mod <- tbats(train_1)
tbats_for = forecast(tbats_mod, h=52)
```

```
#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)"
  )
```


Forecasts for weekly cost (CA_FRS to IL_CHI)
Approximate Cost
Time