Modeling Demand

Gustavo Mello

2022-03-28

Data loading

```r
# Imports dataset made in EDA script
cwd <- getwd()
daily.revenue.listings <- read_csv(paste0(cwd, "/../data/output/daily_revenue_listings.csv"))
```

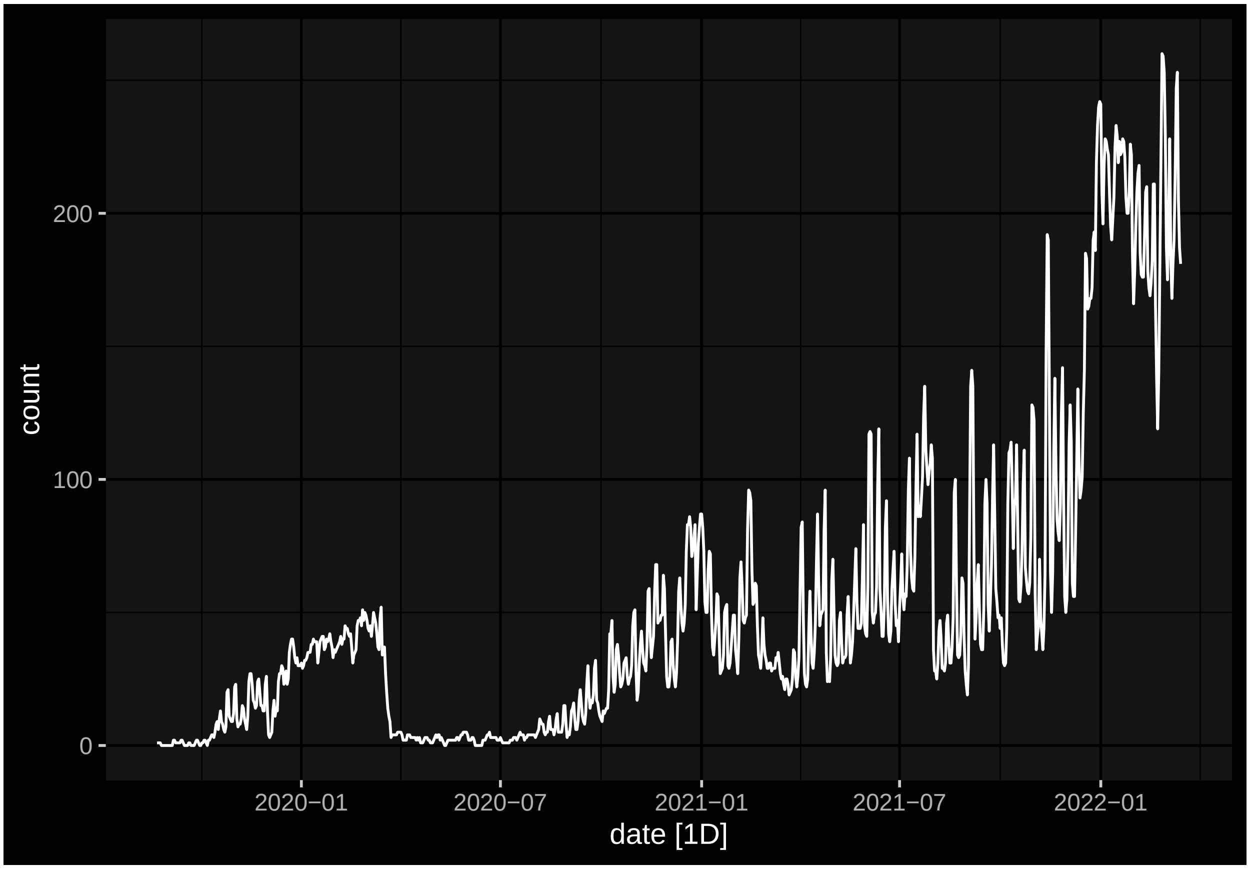
```
## Rows: 289021 Columns: 26
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): listing, Localização, Categoria, Hotel, Status, Tipo
## dbl  (17): last_offered_price, occupancy, revenue, blocked, reservation_adva...
## date  (3): date, creation_date, Data.Inicial.do.contrato
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Reservations across time

```r
reservations.daily <- daily.revenue.listings %>%
  filter(date <= "2022-03-15", occupancy==1, blocked==0) %>%
  select(date) %>%
  group_by(date) %>%
  summarise(count=n()) %>%
  tsibble(index=date) %>%
  fill_gaps(count=0, .full=TRUE) %>%
  mutate(is.workday = !(timeDate::isWeekend(date)
                        | (date %in% holidaysANBIMA)))

autoplot(reservations.daily)
```

```
## Plot variable not specified, automatically selected `.vars = count`
```



Forecasting

Dynamic Harmonic Regression

To forecast with this dataset, we need a model that captures multiple seasonalities and is robust to sudden changes (like the pandemic). Also, as the seasonality effects seems to scale with the response variable magnitude, it is reasonable to test applying a log transform prior to modeling. For these reasons, a dynamic harmonic regression model is fitted using Fourier terms to account for seasonality and an ARMA model to account for short-term dynamics.

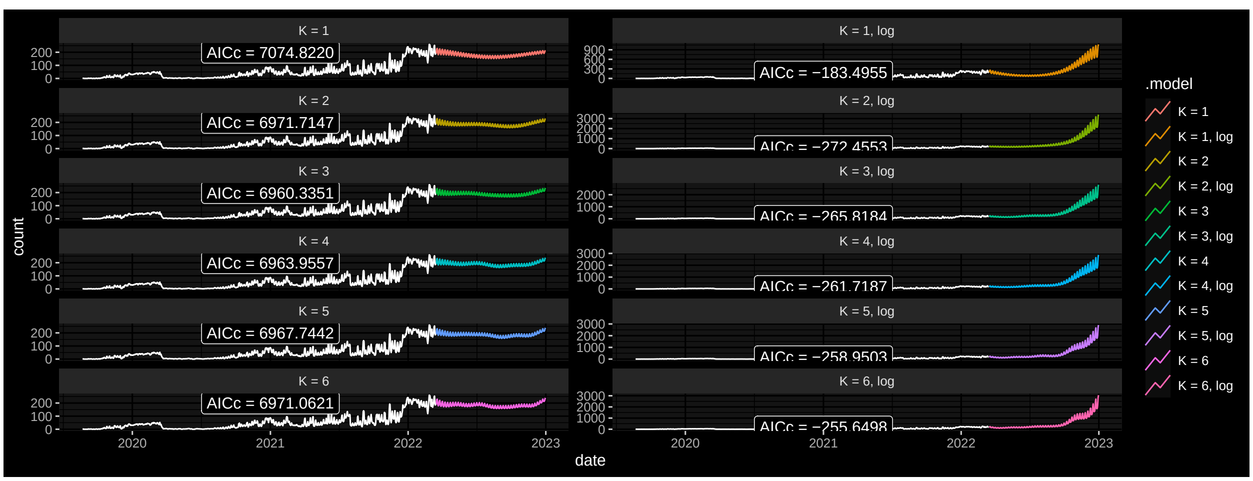
```
dynamic_harmonic_regression <- function(K, log_transform=FALSE){
  K_weekly <- if (K > 3) {3} else {K}
  if(log_transform){
    ARIMA(log(count) ~ PDQ(0, 0, 0)
                       + fourier(K=K_weekly, period=7)
                       + fourier(K=K, period=365.25)
                       + is.workday)
  } else{
    ARIMA(count ~ PDQ(0, 0, 0)
                  + fourier(K=K_weekly, period=7)
                  + fourier(K=K, period=365.25)
                  + is.workday)
  }
}

fit <- reservations.daily %>%
  mutate(count=count + 1) %>%
```

```r
  model("K = 1"       = dynamic_harmonic_regression(K=1),
        "K = 2"       = dynamic_harmonic_regression(K=2),
        "K = 3"       = dynamic_harmonic_regression(K=3),
        "K = 4"       = dynamic_harmonic_regression(K=4),
        "K = 5"       = dynamic_harmonic_regression(K=5),
        "K = 6"       = dynamic_harmonic_regression(K=6),
        "K = 1, log" = dynamic_harmonic_regression(K=1, log_transform = TRUE),
        "K = 2, log" = dynamic_harmonic_regression(K=2, log_transform = TRUE),
        "K = 3, log" = dynamic_harmonic_regression(K=3, log_transform = TRUE),
        "K = 4, log" = dynamic_harmonic_regression(K=4, log_transform = TRUE),
        "K = 5, log" = dynamic_harmonic_regression(K=5, log_transform = TRUE),
        "K = 6, log" = dynamic_harmonic_regression(K=6, log_transform = TRUE))
```

```r
reservations.daily.new.data <- reservations.daily %>%
  new_data(291L) %>%
  mutate(is.workday = !(timeDate::isWeekend(date)
                        | (date %in% holidaysANBIMA)))

fit %>%
  forecast(new_data=reservations.daily.new.data) %>%
  autoplot(reservations.daily, level=NULL) +
  facet_wrap(vars(.model), ncol=2, scale="free_y") +
  geom_label(
    aes(x = ymd("2021-01-01"), y = 200,
        label = paste0("AICc = ", format(AICc))),
    data = glance(fit))
```



DHR excluding COVID lockdowns in 2022 and prior data

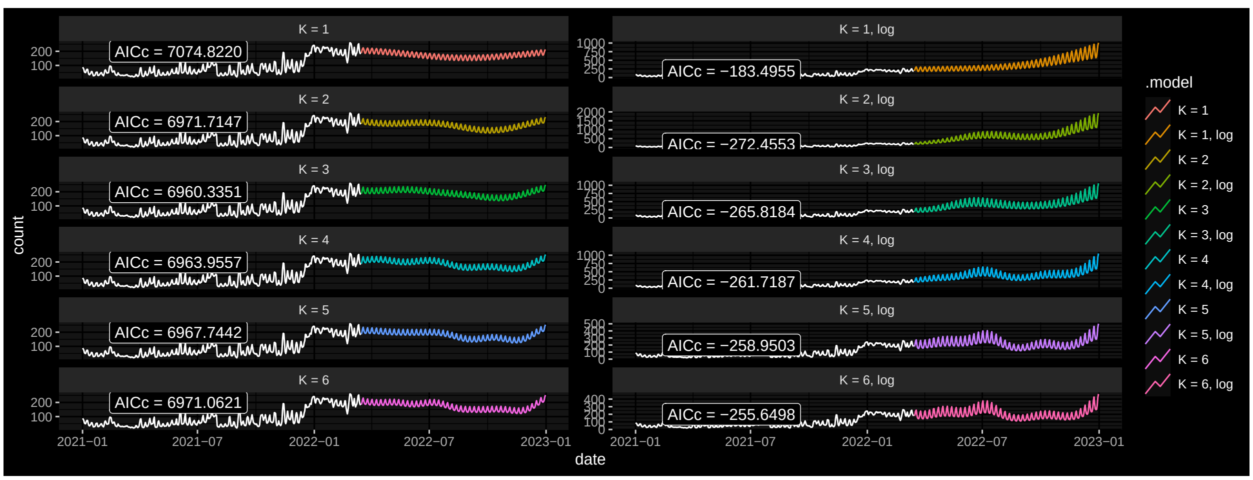
```r
reservations.daily.COVID.excluded <- reservations.daily %>%
  filter_index("2021-01-01" ~ "2022-03-15")

fit.COVID.excluded <- reservations.daily.COVID.excluded %>%
  mutate(count=count + 1) %>%
  model("K = 1"       = dynamic_harmonic_regression(K=1),
        "K = 2"       = dynamic_harmonic_regression(K=2),
        "K = 3"       = dynamic_harmonic_regression(K=3),
        "K = 4"       = dynamic_harmonic_regression(K=4),
```

```
        "K = 5"      = dynamic_harmonic_regression(K=5),
        "K = 6"      = dynamic_harmonic_regression(K=6),
        "K = 1, log" = dynamic_harmonic_regression(K=1, log_transform = TRUE),
        "K = 2, log" = dynamic_harmonic_regression(K=2, log_transform = TRUE),
        "K = 3, log" = dynamic_harmonic_regression(K=3, log_transform = TRUE),
        "K = 4, log" = dynamic_harmonic_regression(K=4, log_transform = TRUE),
        "K = 5, log" = dynamic_harmonic_regression(K=5, log_transform = TRUE),
        "K = 6, log" = dynamic_harmonic_regression(K=6, log_transform = TRUE))
```

```
reservations.daily.COVID.excluded.new.data <- reservations.daily.COVID.excluded %>%
  new_data(291L) %>%
  mutate(is.workday = !(timeDate::isWeekend(date)
                        | (date %in% holidaysANBIMA)))

fit.COVID.excluded %>%
  forecast(new_data=reservations.daily.COVID.excluded.new.data) %>%
  autoplot(reservations.daily.COVID.excluded,
           level=NULL) +
  facet_wrap(vars(.model), ncol=2, scale="free_y") +
  geom_label(
    aes(x = ymd("2021-06-01"), y = 200,
        label = paste0("AICc = ", format(AICc))),
    data = glance(fit))
```



K hyperparameter tuning

```
plot_K <- function(K=2){
  fit_test <- reservations.daily.COVID.excluded %>%
    mutate(count=count + 1) %>%
    model(dynamic_harmonic_regression(K=K, log_transform = TRUE))

    fit_test %>%
  forecast(new_data=reservations.daily.COVID.excluded.new.data) %>%
  autoplot(reservations.daily.COVID.excluded.new.data,
           level=NULL) +
  geom_label(
    aes(x = ymd("2021-06-01"), y = 200,
        label = paste0("AICc = ", format(AICc))),
```

```r
    data = glance(fit))
}
```

```r
models <- reservations.daily.COVID.excluded %>%
  filter_index(~ "2022-03-04") %>%
  stretch_tsibble(.init = 369, .step=10) %>%
  model(
    "K = 1"  = dynamic_harmonic_regression(K=1,  log_transform=TRUE),
    "K = 2"  = dynamic_harmonic_regression(K=2,  log_transform=TRUE),
    "K = 5"  = dynamic_harmonic_regression(K=5,  log_transform=TRUE)#, Cached
    # "K = 10" = dynamic_harmonic_regression(K=10, log_transform=TRUE),
    # "K = 20" = dynamic_harmonic_regression(K=20, log_transform=TRUE),
    # "K = 30" = dynamic_harmonic_regression(K=30, log_transform=TRUE),
    # "K = 50" = dynamic_harmonic_regression(K=50, log_transform=TRUE)
  )
```

```r
CV.data <- reservations.daily.COVID.excluded %>%
  filter_index(~ "2022-03-04") %>%
  stretch_tsibble(.init = 369, .step=10) %>%
  new_data(10L) %>%
  mutate(is.workday = !(timeDate::isWeekend(date)
                        | (date %in% holidaysANBIMA)))

results <- models %>%
  forecast(new_data=CV.data) %>%
  accuracy(reservations.daily.COVID.excluded) %>%
  select(.model, RMSE:MAPE)

results %>% arrange(RMSE)
```

```
## # A tibble: 3 x 5
##   .model  RMSE   MAE   MPE  MAPE
##   <chr>  <dbl> <dbl> <dbl> <dbl>
## 1 K = 1   48.5  38.6 -8.35  19.0
## 2 K = 2   53.8  40.2 -5.59  19.3
## 3 K = 5   55.7  42.6 11.0   20.5
```

```r
cross.validate.for <- function(K){
  model_for_K <- reservations.daily.COVID.excluded %>%
    filter_index(~ "2022-03-04") %>%
    stretch_tsibble(.init = 369, .step=10) %>%
    model(dynamic_harmonic_regression(K=K,  log_transform=TRUE))

  CV.data <- reservations.daily.COVID.excluded %>%
  filter_index(~ "2022-03-04") %>%
  stretch_tsibble(.init = 369, .step=10) %>%
  new_data(10L) %>%
  mutate(is.workday = !(timeDate::isWeekend(date)
                        | (date %in% holidaysANBIMA)))

  model_for_K %>%
  forecast(new_data=CV.data) %>%
  accuracy(reservations.daily.COVID.excluded) %>%
  select(.model, RMSE:MAPE)
}
```

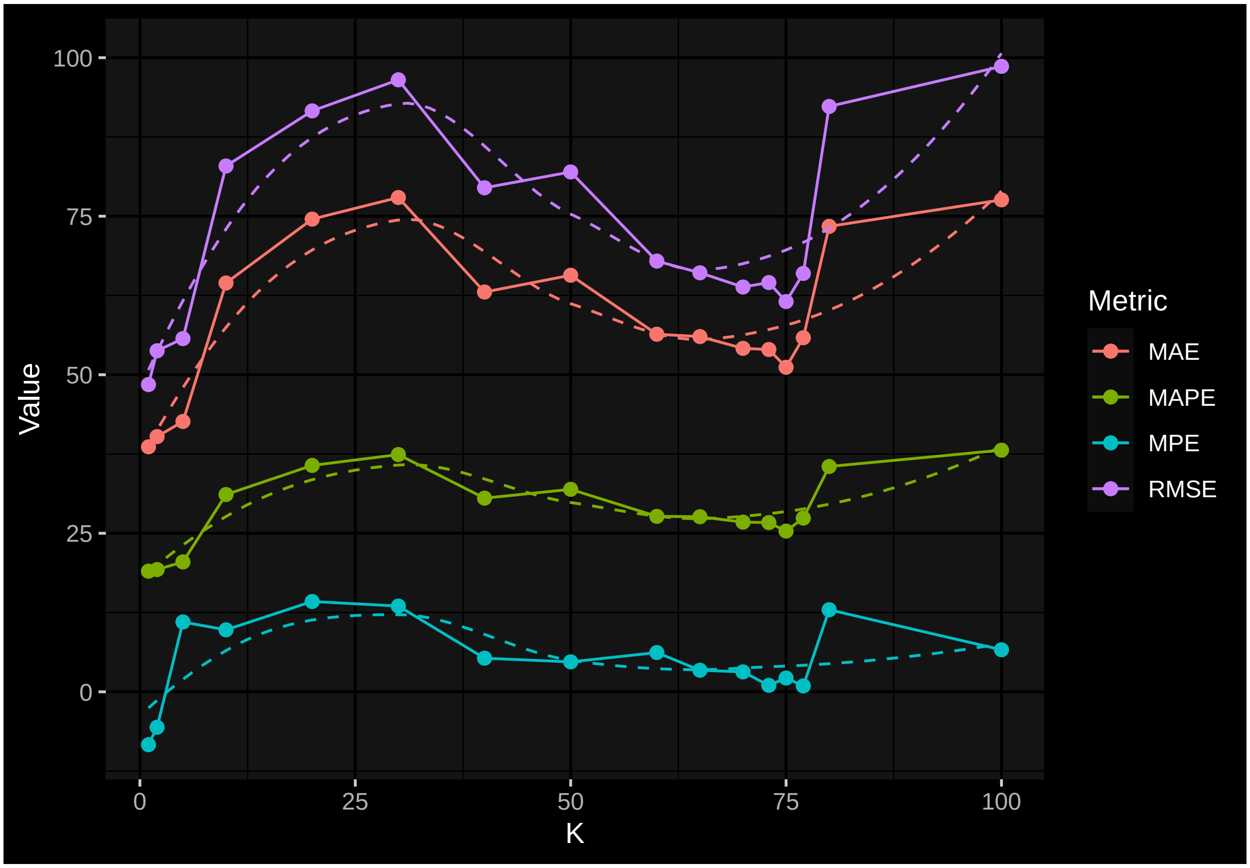
```r
parameter_metrics <- results %>%
  mutate(K=sapply(results$.model,
                  function(x){
                    parse_integer(substring(x, first=5))
                  })) %>%
  add_row(.model="K = 10", RMSE=82.92, MAE=64.47,
          MPE=9.77, MAPE=31.11, K=10L) %>%
  add_row(.model="K = 20", RMSE=91.61, MAE=74.54,
          MPE=14.24, MAPE=35.70, K=20L) %>%
  add_row(.model="K = 30", RMSE=96.51, MAE=77.95,
          MPE=13.52, MAPE=37.41, K=30L) %>%
  add_row(.model="K = 50", RMSE=81.98, MAE=65.69,
          MPE=4.72, MAPE=31.93, K=50L) %>%
  add_row(.model="K = 100", RMSE=98.63, MAE=77.58,
          MPE=6.63, MAPE=38.10, K=100L) %>%
  add_row(.model="K = 60", RMSE=67.92, MAE=56.39,
          MPE=6.19, MAPE=27.67, K=60L) %>%
  add_row(.model="K = 40", RMSE=79.48, MAE=63.04,
          MPE=5.30, MAPE=30.54, K=40L) %>%
  add_row(.model="K = 70", RMSE=63.83, MAE=54.16,
          MPE=3.14, MAPE=26.76, K=70L) %>%
  add_row(.model="K = 80", RMSE=92.31, MAE=73.40,
          MPE=12.94, MAPE=35.53, K=80L) %>%
  add_row(.model="K = 65", RMSE=66.08, MAE=56.02,
          MPE=3.40, MAPE=27.61, K=65L) %>%
  add_row(.model="K = 75", RMSE=61.54, MAE=51.18,
          MPE=2.17, MAPE=25.34, K=75L) %>%
  add_row(.model="K = 77", RMSE=65.98, MAE=55.84,
          MPE=0.93, MAPE=27.39, K=77L) %>%
  add_row(.model="K = 73", RMSE=64.54, MAE=53.98,
          MPE=1.01, MAPE=26.69, K=73L) %>%
  arrange(K)

parameter_metrics
```

```
## # A tibble: 16 x 6
##    .model   RMSE   MAE   MPE  MAPE     K
##    <chr>   <dbl> <dbl> <dbl> <dbl> <int>
##  1 K = 1    48.5  38.6 -8.35  19.0     1
##  2 K = 2    53.8  40.2 -5.59  19.3     2
##  3 K = 5    55.7  42.6 11.0   20.5     5
##  4 K = 10   82.9  64.5  9.77  31.1    10
##  5 K = 20   91.6  74.5 14.2   35.7    20
##  6 K = 30   96.5  78.0 13.5   37.4    30
##  7 K = 40   79.5  63.0  5.3   30.5    40
##  8 K = 50   82.0  65.7  4.72  31.9    50
##  9 K = 60   67.9  56.4  6.19  27.7    60
## 10 K = 65   66.1  56.0  3.4   27.6    65
## 11 K = 70   63.8  54.2  3.14  26.8    70
## 12 K = 73   64.5  54.0  1.01  26.7    73
## 13 K = 75   61.5  51.2  2.17  25.3    75
## 14 K = 77   66.0  55.8  0.93  27.4    77
## 15 K = 80   92.3  73.4 12.9   35.5    80
## 16 K = 100  98.6  77.6  6.63  38.1   100
```

```
parameter_metrics %>%
  pivot_longer(RMSE:MAPE, names_to = "Metric", values_to = "Value") %>%
  ggplot(aes(x=K, y=Value, color=Metric)) +
  geom_point(size=2) +
  geom_line() +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5)
```

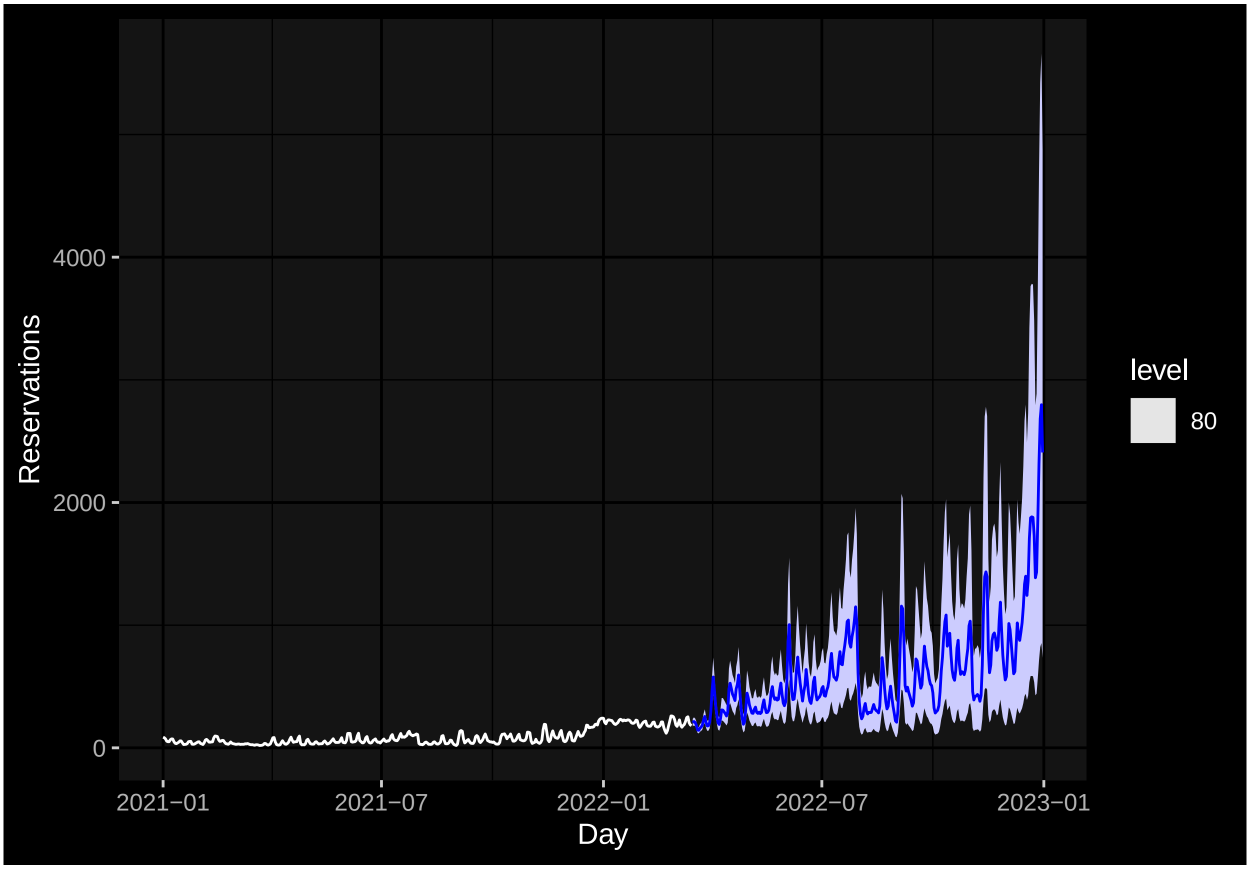
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



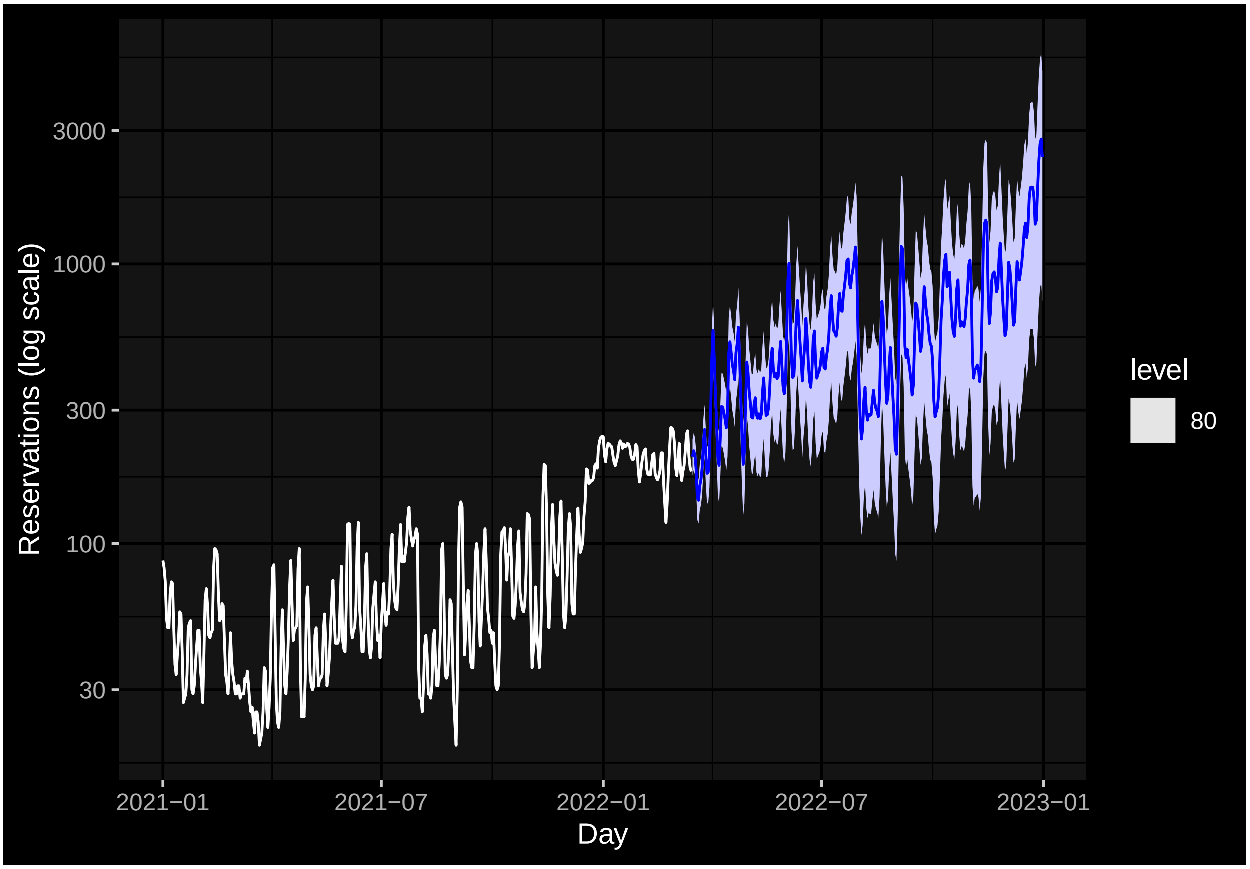
Final model selection The MPE may be seen as a bias metric while the other accuracy assessors act more like variance metrics. For this reason, since the model is expected to forecast to a distant horizon (291 days into the future), a low bias is prioritized and balanced with lesser possible variance.

```
# May take a while to train
final.model <- reservations.daily.COVID.excluded %>%
  model(dynamic_harmonic_regression(K=75, log_transform=TRUE))
```

```
final.model %>%
  forecast(reservations.daily.COVID.excluded.new.data) %>%
  autoplot(reservations.daily.COVID.excluded, level=80) +
  labs("Daily Number of Reservations Forecast",
       x="Day",
       y="Reservations")
```




```
final.model %>%
  forecast(reservations.daily.COVID.excluded.new.data) %>%
  autoplot(reservations.daily.COVID.excluded, level=80) +
  labs("Daily Number of Reservations Forecast",
       x="Day",
       y="Reservations (log scale)") +
  scale_y_log10()
```


Reservations (log scale)
3000
1000
300
100
30
2021−01
2021−07
2022−01
2022−07
2023−01
Day
level
80