Google Trends Tiktok

Nazlı Ece Baltepe

04 10 2022

```
library(forecast)
```

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

```
library(ggplot2)
library(gtrendsR)
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(caret)
```

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.2     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
library(ISLR)
library(broom)
library(lubridate)
```

```
##
## Attaching package: ’lubridate’

## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

#Pulling the data from Google Trends

Pulling the Google Trends data on the keyword “tiktok” via the gtrendsR package for the selected dates.

```
tiktoktrends<-gtrends(keyword = c("tiktok"), time="2019-01-01 2022-10-02")
```

Checking the structure

Checking the structure of the newly created object, tiktoktrends

```
str(tiktoktrends)
```

```
## List of 7
##  $ interest_over_time :’data.frame’: 195 obs. of  7 variables:
##   ..$ date    : POSIXct[1:195], format: "2019-01-06" "2019-01-13" ...
##   ..$ hits    : int [1:195] 2 2 2 2 2 2 3 3 3 3 ...
##   ..$ keyword : chr [1:195] "tiktok" "tiktok" "tiktok" "tiktok" ...
##   ..$ geo     : chr [1:195] "world" "world" "world" "world" ...
##   ..$ time    : chr [1:195] "2019-01-01 2022-10-02" "2019-01-01 2022-10-02" "2019-01-01 2022-10-02"
##   ..$ gprop   : chr [1:195] "web" "web" "web" "web" ...
##   ..$ category: int [1:195] 0 0 0 0 0 0 0 0 0 0 ...
##  $ interest_by_country:’data.frame’: 250 obs. of  5 variables:
##   ..$ location: chr [1:250] "Indonesia" "American Samoa" "Nepal" "Philippines" ...
##   ..$ hits    : int [1:250] 100 NA 50 47 NA NA 30 NA NA 26 ...
##   ..$ keyword : chr [1:250] "tiktok" "tiktok" "tiktok" "tiktok" ...
##   ..$ geo     : chr [1:250] "world" "world" "world" "world" ...
##   ..$ gprop   : chr [1:250] "web" "web" "web" "web" ...
##  $ interest_by_region : NULL
##  $ interest_by_dma    :’data.frame’: 306 obs. of  5 variables:
##   ..$ location: chr [1:306] "Fresno-Visalia CA" "Bakersfield CA" "Laredo TX" "Charlottesville VA" ..
##   ..$ hits    : int [1:306] 100 97 91 89 88 86 82 82 80 80 ...
##   ..$ keyword : chr [1:306] "tiktok" "tiktok" "tiktok" "tiktok" ...
##   ..$ geo     : chr [1:306] "world" "world" "world" "world" ...
##   ..$ gprop   : chr [1:306] "web" "web" "web" "web" ...
##  $ interest_by_city   :’data.frame’: 200 obs. of  5 variables:
##   ..$ location: chr [1:200] "Cipeundeuy" "Banjarsari" "Lohbener" "Pandeglang" ...
##   ..$ hits    : int [1:200] NA NA NA NA NA 100 NA NA NA 94 ...
##   ..$ keyword : chr [1:200] "tiktok" "tiktok" "tiktok" "tiktok" ...
##   ..$ geo     : chr [1:200] "world" "world" "world" "world" ...
##   ..$ gprop   : chr [1:200] "web" "web" "web" "web" ...
##  $ related_topics     :'data.frame': 35 obs. of  5 variables:
##   ..$ subject       : chr [1:35] "100" "20" "10" "4" ...
##   ..$ related_topics: chr [1:35] "top" "top" "top" "top" ...
##   ..$ value         : chr [1:35] "TikTok" "Download" "Video Downloader" "Watermark" ...
##   ..$ keyword       : chr [1:35] "tiktok" "tiktok" "tiktok" "tiktok" ...
##   ..$ category      : int [1:35] 0 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "reshapeLong")=List of 4
##   .. ..$ varying:List of 1
##   .. .. ..$ value: chr "top"
##   .. .. ..- attr(*, "v.names")= chr "value"
##   .. .. ..- attr(*, "times")= chr "top"
##   .. ..$ v.names: chr "value"
##   .. ..$ idvar  : chr "id"
##   .. ..$ timevar: chr "related_topics"
##  $ related_queries    :'data.frame': 50 obs. of  5 variables:
##   ..$ subject        : chr [1:50] "100" "93" "62" "29" ...
##   ..$ related_queries: chr [1:50] "top" "top" "top" "top" ...
##   ..$ value          : chr [1:50] "download tiktok" "video tiktok" "download tiktok video" "downloade
##   ..$ keyword        : chr [1:50] "tiktok" "tiktok" "tiktok" "tiktok" ...
##   ..$ category       : int [1:50] 0 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "reshapeLong")=List of 4
##   .. ..$ varying:List of 1
##   .. .. ..$ value: chr "top"
##   .. .. ..- attr(*, "v.names")= chr "value"
##   .. .. ..- attr(*, "times")= chr "top"
##   .. ..$ v.names: chr "value"
##   .. ..$ idvar  : chr "id"
##   .. ..$ timevar: chr "related_queries"
##  - attr(*, "class")= chr [1:2] "gtrends" "list"
```

Pulling the data on the interest over time

Creating a new data frame with only data on the interest over time, and checking its structure.

```
interestovertime<-tiktoktrends$interest_over_time
str(interestovertime)
```

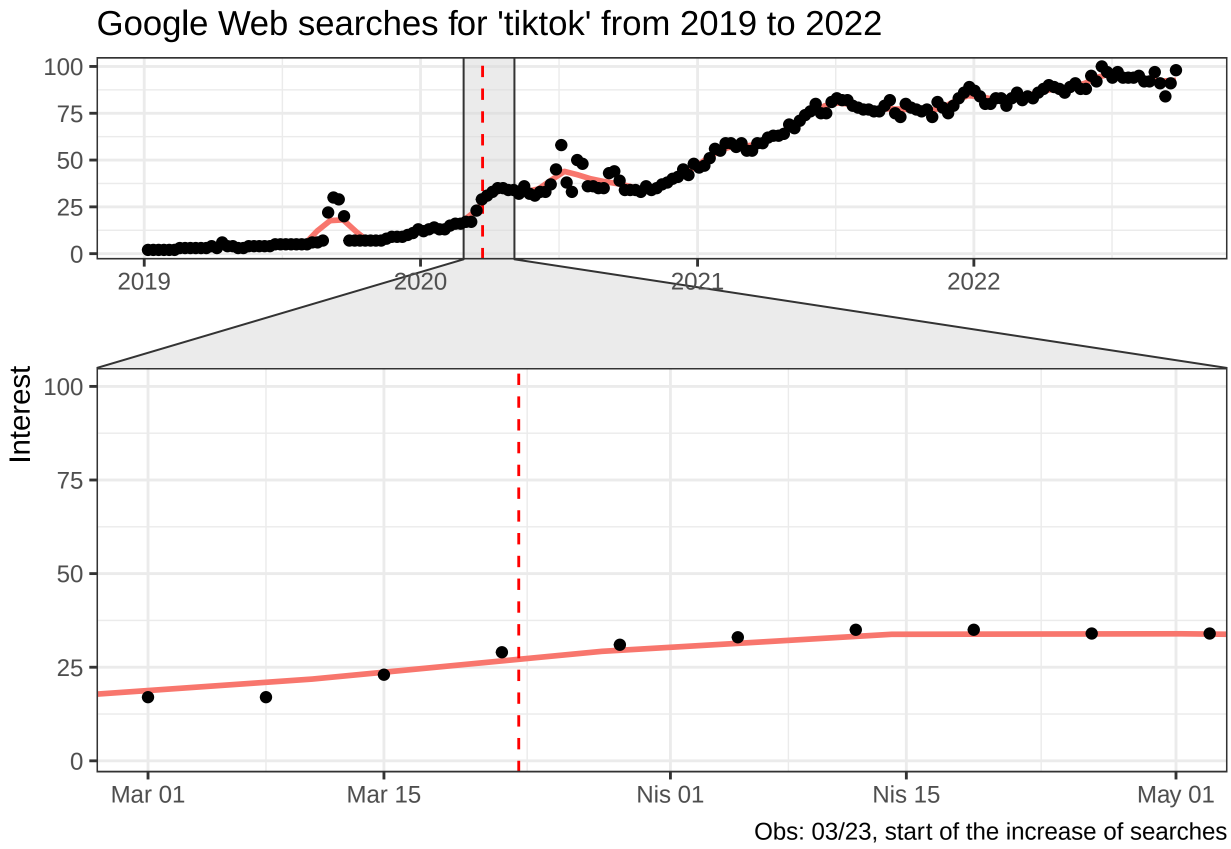
```
## 'data.frame':    195 obs. of  7 variables:
##  $ date    : POSIXct, format: "2019-01-06" "2019-01-13" ...
##  $ hits    : int  2 2 2 2 2 2 3 3 3 3 ...
##  $ keyword : chr  "tiktok" "tiktok" "tiktok" "tiktok" ...
##  $ geo     : chr  "world" "world" "world" "world" ...
##  $ time    : chr  "2019-01-01 2022-10-02" "2019-01-01 2022-10-02" "2019-01-01 2022-10-02" "2019-01-01
##  $ gprop   : chr  "web" "web" "web" "web" ...
##  $ category: int  0 0 0 0 0 0 0 0 0 0 ...
```

Creating the time series graph showing the change in interest over time

I will create the time series graph, zooming in the dates where the striking increase of the app's popularity started.

```
interestovertime %>%
  ggplot(aes(x = date,
             y = hits,group=keyword,
             color = keyword))  +
  theme_bw()+
  labs(title = "Google Web searches for 'tiktok' from 2019 to 2022",
       caption = "Obs: 03/23, start of the increase of searches",
       x= NULL, y = "Interest")+
  ggforce::facet_zoom(xlim = c(as.POSIXct(as.Date("2020-03-01")),as.POSIXct(as.Date("2020-05-01")))) +
  geom_smooth(span=0.1,se=FALSE) + geom_vline(xintercept = as.POSIXct(as.Date("2020-03-23")),color = "r
  theme(legend.position = "none") +
  geom_point(color="black")
```

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



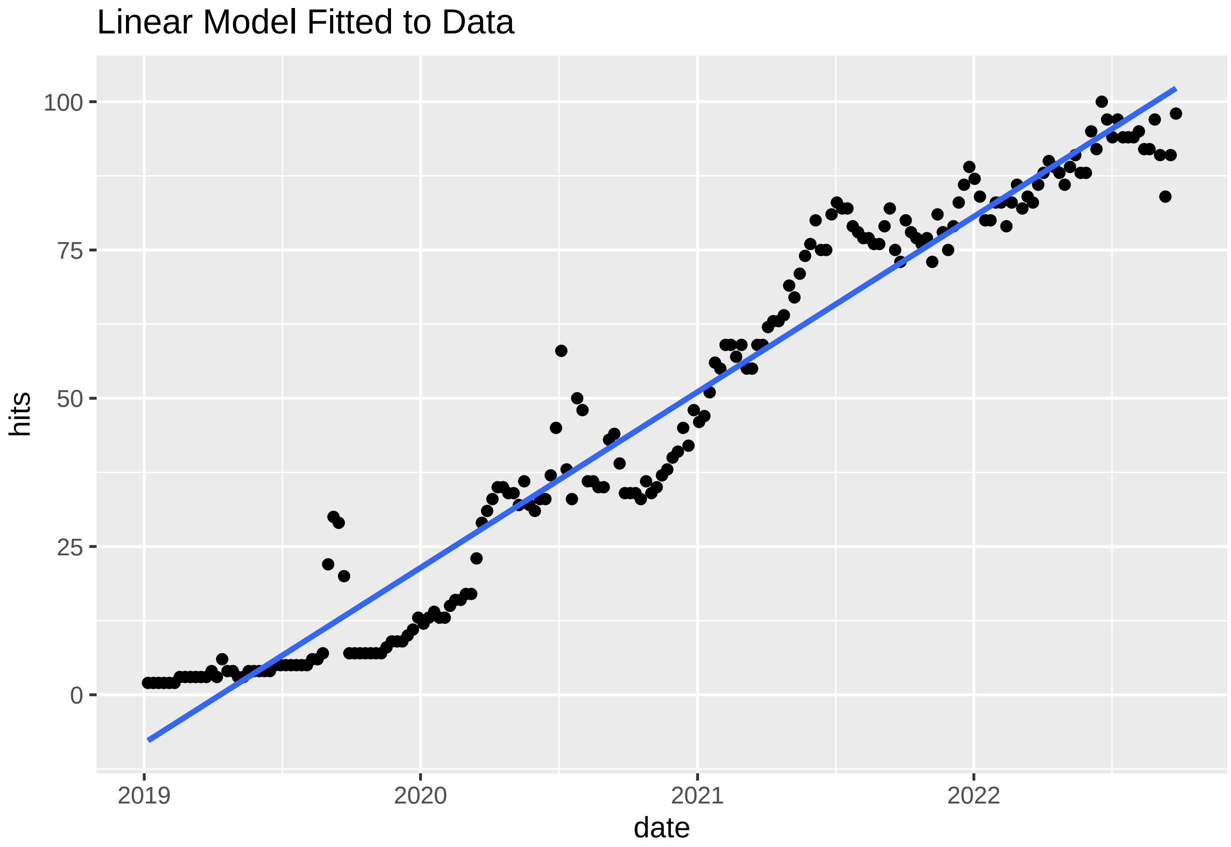
Starting with the pandemic, there is a steady increase in the popularity of the app, as demonstrated in the Google trends information.

Checking the linearity of the relationship between variables

Let’s see if the relationship between the hits and date is linear.

```
ggplot(data = interestovertime, aes(date, hits)) +
geom_point() + geom_smooth(method = "lm", se=FALSE)+
ggtitle("Linear Model Fitted to Data")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



#Looking for a better model

The relationship does not seem completely linear. The pattern seems slightly non-linear. Let’s compute the test error estimates for polynomials up to the 3rd degree, using the bootstrapping approach. I select 100 samples, and set the seed to 2.

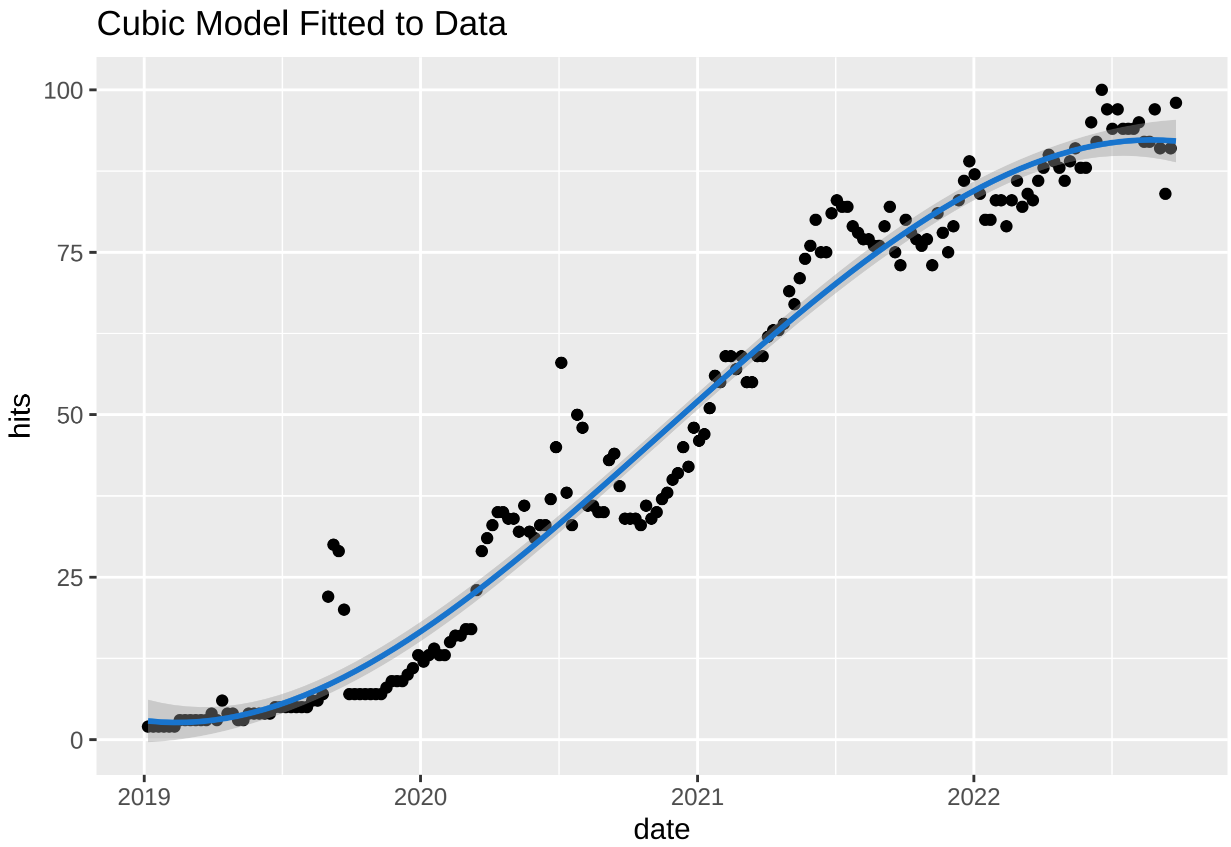
```
set.seed(2)
rmse <- numeric(3)
for(i in 1:3){
train_control <- trainControl(method = "boot",
number = 100)
f <- bquote(hits ~ poly(date, .(i)))
models <- train(as.formula(f), data = interestovertime,
trControl=train_control, method='glm')
rmse[i] <- models$results$RMSE
}

rmse
```

```
## [1] 7.214355 7.198654 5.942421
```

Plotting the graph with the polynomial regression line

The model using the cubic function as has the lowest RMSE. I would like to see how this model fits on a graph first.

```
ggplot(interestovertime, aes(date, hits)) +
geom_point() + geom_smooth(method = "lm", col="dodgerblue3",
formula=y~poly(x,3))+
ggtitle("Cubic Model Fitted to Data")
```



This model fits much better!

P value and the R-squared of the polynomial model

Let's see the P value and the R-squared value.

```
model <- lm(hits ~ poly(date,3),data=interestovertime)
summary(model)
```

```
##
## Call:
## lm(formula = hits ~ poly(date, 3), data = interestovertime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6875  -3.9850  -0.6458   2.5758  24.5384
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     47.2667     0.4226 111.842   <2e-16 ***
## poly(date, 3)1 445.6722     5.9016  75.518   <2e-16 ***
## poly(date, 3)2   1.4535     5.9016   0.246    0.806
## poly(date, 3)3 -56.4629     5.9016  -9.567   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 5.902 on 191 degrees of freedom
## Multiple R-squared:  0.9681, Adjusted R-squared:  0.9676
## F-statistic:  1931 on 3 and 191 DF,  p-value: < 2.2e-16
```

P value is below 0.05, indicating a statistical relationship between date and Google hits. The R-squared is 0.9681, which means that the model explains %96.81 of the variability in the response variable, which is hits. This indicates that the model has a high validity.

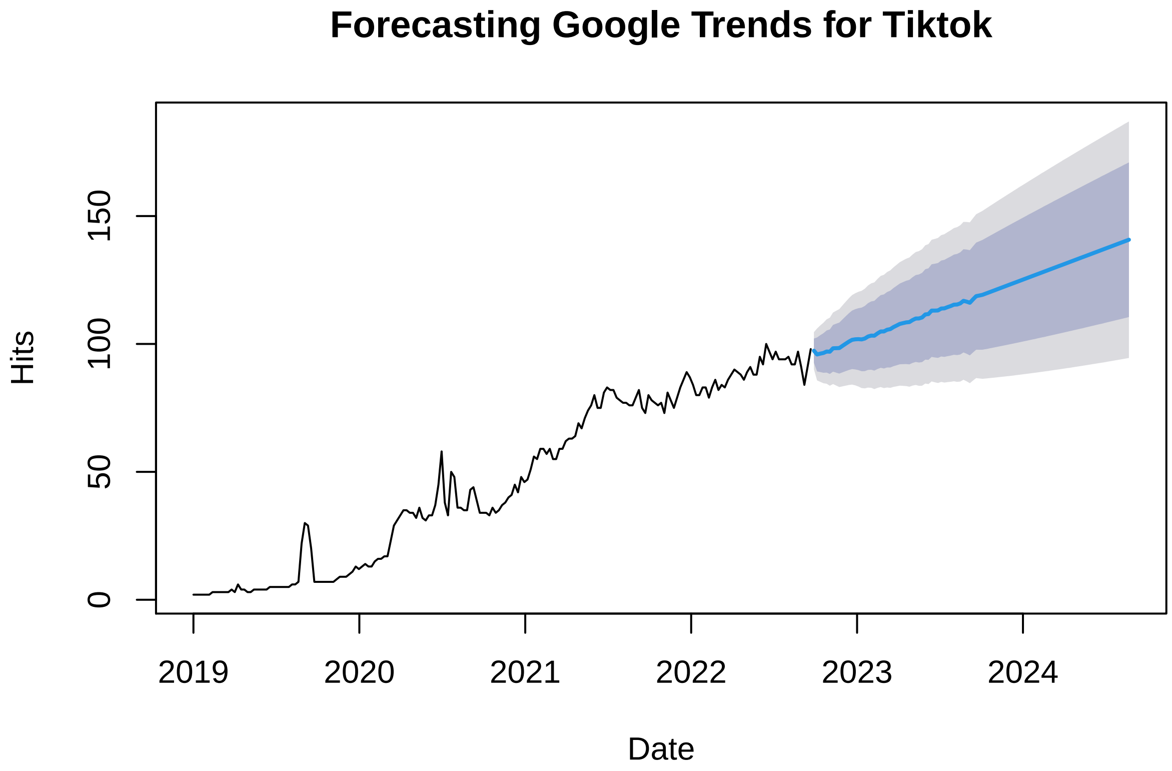
Forecasting the future trend via auto.arima

auto.arima is a function in the forecast package which fits the best ARIMA model to our data. As it works only with univariate time series, I pull the hits data and turn it into time series before creating the model. Then, I plot the forecast for the next 100 steps, providing us the forecast until 2024.

```
onlyhits<-ts(interestovertime$hits,start= c(2019,1,1), frequency= 52.14)
onlyhits
```

```
## Time Series:
## Start = 2019
## End = 2022.72075182202
## Frequency = 52.14
##   [1]   2   2   2   2   2   2   3   3   3   3   3   3   4   3   6   4   4   3
##  [19]   3   4   4   4   4   4   5   5   5   5   5   5   5   6   6   7  22  30
##  [37]  29  20   7   7   7   7   7   7   7   8   9   9   9  10  11  13  12  13
##  [55]  14  13  13  15  16  16  17  17  23  29  31  33  35  35  34  34  32  36
##  [73]  32  31  33  33  37  45  58  38  33  50  48  36  36  35  35  43  44  39
##  [91]  34  34  34  33  36  34  35  37  38  40  41  45  42  48  46  47  51  56
## [109]  55  59  59  57  59  55  55  59  59  62  63  63  64  69  67  71  74  76
## [127]  80  75  75  81  83  82  82  79  78  77  77  76  76  79  82  75  73  80
## [145]  78  77  76  77  73  81  78  75  79  83  86  89  87  84  80  80  83  83
## [163]  79  83  86  82  84  83  86  88  90  89  88  86  89  91  88  88  95  92
## [181] 100  97  94  97  94  94  94  95  92  92  97  91  84  91  98
```

```
modelm<-auto.arima(onlyhits)
forecast_data<-forecast(modelm, 100)
plot(forecast_data, main = "Forecasting Google Trends for Tiktok", ylab = "Hits", xlab = "Date")
```



According to this forecast, the popularity of Tiktok will continue to increase in next couple of years.