LabWork

Ashish Pal

September 22, 2017

Analysis Onion Data for the year 2017 and predicting for future Year

```
# Reading Libraries
library(rvest)
```

```
## Loading required package: xml2
```

```
library(tidyr)
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(stringr)
library(ggplot2)
library(prophet)
```

```
## Loading required package: Rcpp
```

Frame

1st part - Find the State with the highest quantity sales. 2nd part - Predict the daily price for the next 30 days for that state.

Acquire

Getting data from NHRDF Database

```
setwd("C:/Users/Ashish/Desktop/GL/Day2/HomeWork")
getwd()
```

```
## [1] "C:/Users/Ashish/Desktop/GL/Day2/HomeWork"
```

```
Odf = read_html("DailyWiseMarketArrivals2017.html") %>%
html_node("#dnn_ctr966_DailyWiseMarketArrivals_GridView1") %>%
html_table()
Odf_1 = Odf
str(Odf_1)
```

```
## 'data.frame':    15079 obs. of  6 variables:
##  $ Date                : chr  "02/Jan/2017" "02/Jan/2017" "03/Jan/2017" "04/Jan/2017" ...
##  $ Market              : chr  "ABOHAR(PB)" "AGRA(UP)" "AGRA(UP)" "AGRA(UP)" ...
##  $ Arrival(q)          : int  200 2850 2950 3400 3800 3700 3300 3500 3150 3100 ...
##  $ Price Minimum (Rs/q): chr  "750" "600" "800" "780" ...
##  $ Price Maximum (Rs/q): chr  "1000" "925" "900" "900" ...
##  $ Modal Price (Rs/q)  : chr  "850" "850" "840" "830" ...
```

Refine

```
col_names =  c("Date","Market","Quantity","Min_Price","Max_Price","Mod_Price")
colnames(Odf_1) = col_names
str(Odf_1)
```

```
## 'data.frame':    15079 obs. of  6 variables:
##  $ Date     : chr  "02/Jan/2017" "02/Jan/2017" "03/Jan/2017" "04/Jan/2017" ...
##  $ Market   : chr  "ABOHAR(PB)" "AGRA(UP)" "AGRA(UP)" "AGRA(UP)" ...
##  $ Quantity : int  200 2850 2950 3400 3800 3700 3300 3500 3150 3100 ...
##  $ Min_Price: chr  "750" "600" "800" "780" ...
##  $ Max_Price: chr  "1000" "925" "900" "900" ...
##  $ Mod_Price: chr  "850" "850" "840" "830" ...
```

Transfrom

```
Odf_2 = Odf_1
Odf_2$Date = as.Date(Odf_1$Date, "%d/%b/%Y")
Odf_2$Min_Price = as.numeric(Odf_2$Min_Price)
```

```
## Warning: NAs introduced by coercion
```

```
Odf_2$Max_Price = as.numeric(Odf_2$Max_Price)
```

```
## Warning: NAs introduced by coercion
```

```
Odf_2$Mod_Price = as.numeric(Odf_1$Mod_Price)
```

```
## Warning: NAs introduced by coercion
```

```
str(Odf_2)
```

```
## 'data.frame':    15079 obs. of  6 variables:
##  $ Date     : Date, format: "2017-01-02" "2017-01-02" ...
##  $ Market   : chr  "ABOHAR(PB)" "AGRA(UP)" "AGRA(UP)" "AGRA(UP)" ...
##  $ Quantity : int  200 2850 2950 3400 3800 3700 3300 3500 3150 3100 ...
##  $ Min_Price: num  750 600 800 780 750 750 800 750 800 800 ...
##  $ Max_Price: num  1000 925 900 900 850 850 900 880 900 900 ...
##  $ Mod_Price: num  850 850 840 830 800 810 840 820 840 850 ...
```

```
dim(Odf_2)
```

```
## [1] 15079     6
```

```
Odf_3 = Odf_2 %>% filter(Market != "Total") %>% mutate(market1 = Market) %>%
separate(market1,c("city","state"), sep = "\\(")
```

```
## Warning: Too many values at 62 locations: 3739, 3740, 3741, 3742, 3743,
## 3744, 3745, 3746, 3747, 3748, 3749, 3750, 3751, 3752, 3753, 3754, 3755,
## 3756, 3757, 3758, ...
```

```
## Warning: Too few values at 2121 locations: 1859, 1860, 1861, 1862, 1863,
## 1864, 1865, 1866, 1867, 1868, 1869, 1870, 1871, 1872, 1873, 1874, 1875,
## 1876, 1877, 1878, ...
```

```
dim(Odf_3)
```

```
## [1] 15078     8
```

```
Odf_3$state = Odf_3$state %>% str_replace("\\)","")
head(Odf_3)
```

```
##         Date     Market Quantity Min_Price Max_Price Mod_Price   city
## 1 2017-01-02 ABOHAR(PB)      200       750      1000       850 ABOHAR
## 2 2017-01-02   AGRA(UP)     2850       600       925       850   AGRA
## 3 2017-01-03   AGRA(UP)     2950       800       900       840   AGRA
## 4 2017-01-04   AGRA(UP)     3400       780       900       830   AGRA
## 5 2017-01-06   AGRA(UP)     3800       750       850       800   AGRA
## 6 2017-01-07   AGRA(UP)     3700       750       850       810   AGRA
##   state
## 1    PB
## 2    UP
## 3    UP
## 4    UP
## 5    UP
## 6    UP
```

```
unique(Odf_3$state)
```

```
##  [1] "PB"        "UP"        "GUJ"       "MS"        "OR"       
##  [6] "RAJ"       "WB"        NA          "KNT"       "Telangana"
## [11] "TN "       "UTT"       "TN"        "TELANGANA" "AS"       
## [16] "MP"        "HR"        "HP"        "AP"        "KER"      
## [21] "RJ"        "CHATT"     "CHGARH"    "F&V "
```

```
# Removing NA fields from state
Odf_4 <- Odf_3 %>% mutate(state = ifelse(is.na(state), Market, state))

Odf_4 = within(Odf_4,state[state == "Telangana"] <- "TELANGANA")
Odf_4 = within(Odf_4,state[state == "RJ" ]<- "RAJ")
Odf_4 = within(Odf_4,state[state == "M.P."] <- "MP")
Odf_4 = within(Odf_4,state[state == "JAIPUR"] <- "RAJ")
Odf_4 = within(Odf_4,state[state == "MS"] <- "MAHARASHTRA")
Odf_4 = within(Odf_4,state[state == "BANGALORE"] <- "KNT")
Odf_4 = within(Odf_4,state[state == "MS"] <- "MAHARASHTRA")
Odf_4 = within(Odf_4,state[state == "BHOPAL"] <- "MP")
Odf_4 = within(Odf_4,state[state == "CHENNAI"] <- "TN")
Odf_4 = within(Odf_4,state[state == "TN "] <- "TN")
Odf_4 = within(Odf_4,state[state == "HYDERABAD"] <- "AP")
Odf_4 = within(Odf_4,state[state == "LUCKNOW"] <- "UP")
Odf_4 = within(Odf_4,state[state == "SHAHJAHANPUR"] <- "UP")
unique(Odf_4$state)
```

```
##  [1] "PB"          "UP"          "GUJ"         "MAHARASHTRA" "OR"         
##  [6] "RAJ"         "WB"          "KNT"         "MP"          "TELANGANA"  
## [11] "BULANDSHAHR" "CHANDIGARH"  "TN"          "UTT"         "DELHI"      
## [16] "GUWAHATI"    "AS"          "AP"          "JAMMU"       "HR"         
## [21] "KOLKATA"     "HP"          "MUMBAI"      "NAGPUR"      "KER"        
## [26] "PATNA"       "CHATT"       "CHGARH"      "F&V "
```

Explore

State with Highest Quantity Sales in 2017

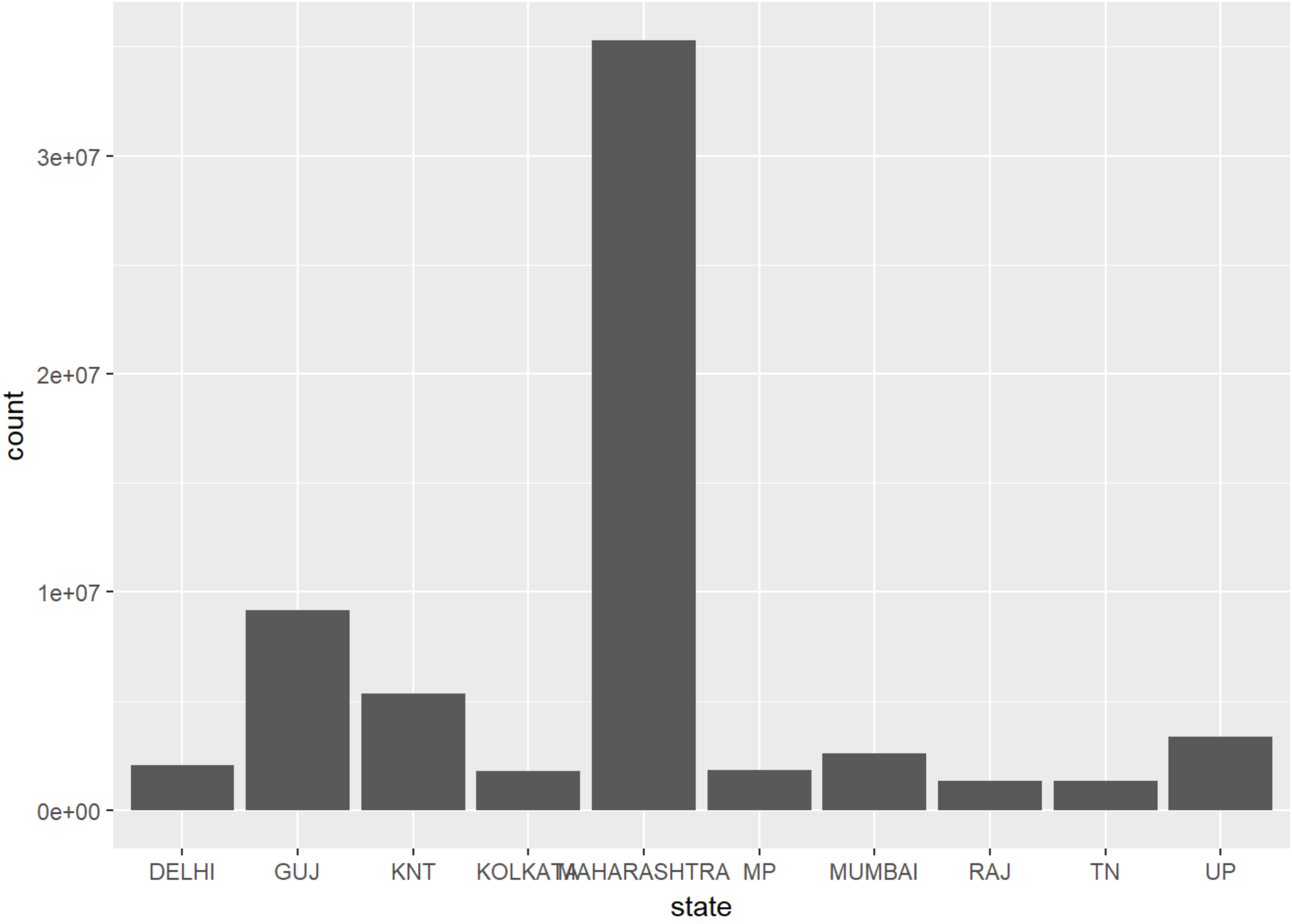
```
sum_quantity_df <- Odf_4 %>%
group_by(state) %>%
summarize(sum_quantity = sum(Quantity),avg_price = mean(Max_Price))
str(sum_quantity_df)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    29 obs. of  3 variables:
##  $ state       : chr  "AP" "AS" "BULANDSHAHR" "CHANDIGARH" ...
##  $ sum_quantity: int  1296925 1700 35 205986 50487 120250 2053518 2001 9180053 1586 ...
##  $ avg_price   : num  1059 1400 850 1271 1258 ...
```

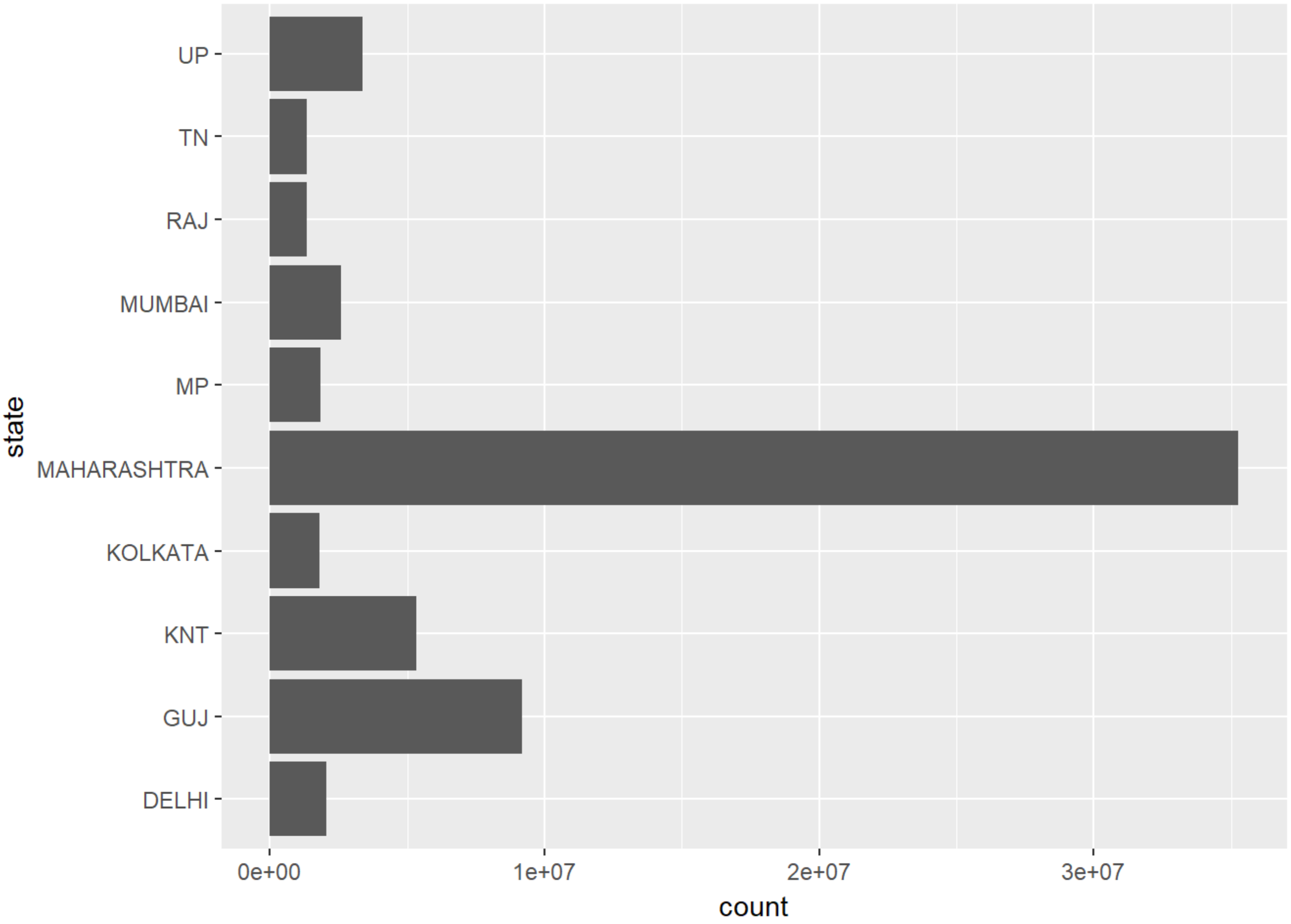
```
Top_quantity = sum_quantity_df %>%
  arrange(desc(sum_quantity))

head(Top_quantity)
```

```
## # A tibble: 6 x 3
##         state sum_quantity avg_price
##         <chr>        <int>     <dbl>
## 1 MAHARASHTRA     35270591  980.8894
## 2         GUJ      9180053  916.1689
## 3         KNT      5334293 1203.6736
## 4          UP      3379929 1120.7454
## 5      MUMBAI      2598632 1064.5767
## 6       DELHI      2053518 1274.4251
```

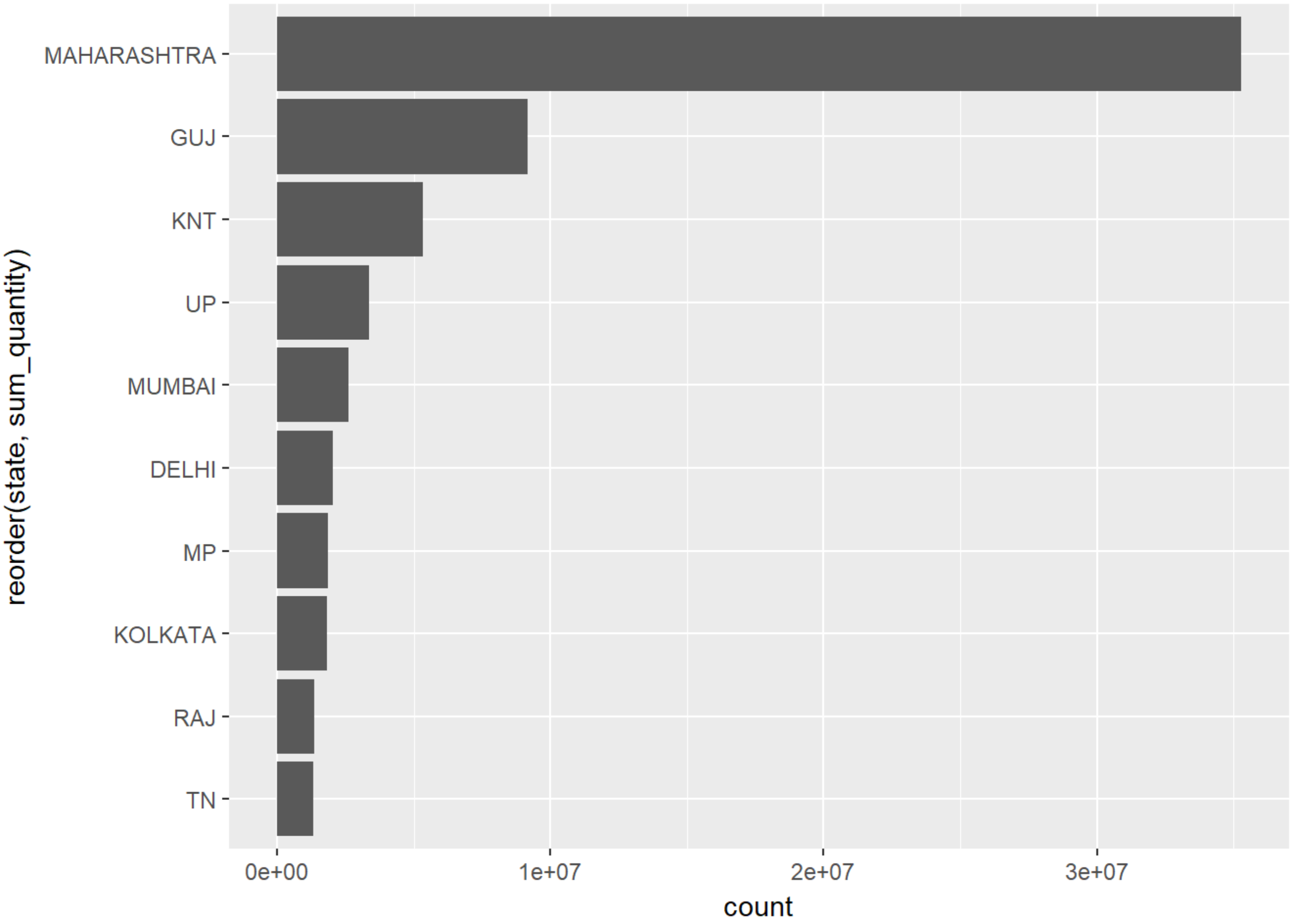
```
#Visualizing data
Top_quantity %>% head(10) %>% ggplot() + aes(state, weight = sum_quantity) + geom_bar()
```



```
Top_quantity %>% head(10) %>% ggplot() + aes(state, weight = sum_quantity) + geom_bar()+coord_flip()
```



```
Top_quantity %>% head(10) %>% ggplot() + aes(reorder(state,sum_quantity), weight = sum_quantity)
 + geom_bar()+coord_flip()
```



##

This clearly shows that

Predicting price of Onion for state = MAHARASHTRA for next 30 days.

```
Ms_Price = Odf_4 %>% filter(state == "MAHARASHTRA") %>% group_by(Date) %>% summarise(Mod_Price_max = max(Mod_Price)) %>% select(Date,Mod_Price_max) %>% arrange(Date)


dim(Ms_Price)
```

```
## [1] 266   2
```

```
head(Ms_Price)
```

```
## # A tibble: 6 x 2
##         Date Mod_Price_max
##       <date>         <dbl>
## 1 2017-01-01           800
## 2 2017-01-02           900
## 3 2017-01-03           770
## 4 2017-01-04           850
## 5 2017-01-05           762
## 6 2017-01-06           750
```

```
str(Ms_Price)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    266 obs. of  2 variables:
##  $ Date         : Date, format: "2017-01-01" "2017-01-02" ...
##  $ Mod_Price_max: num  800 900 770 850 762 750 700 600 670 770 ...
```

```
col1 = c("ds","y")
colnames(Ms_Price) = col1
d =Ms_Price
m = prophet(d)
```

```
## Disabling yearly seasonality. Run prophet with yearly.seasonality=TRUE to override this.
```

```
## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.
```

```
## Initial log joint probability = -6.03098
## Optimization terminated normally:
##   Convergence detected: relative gradient magnitude is below tolerance
```

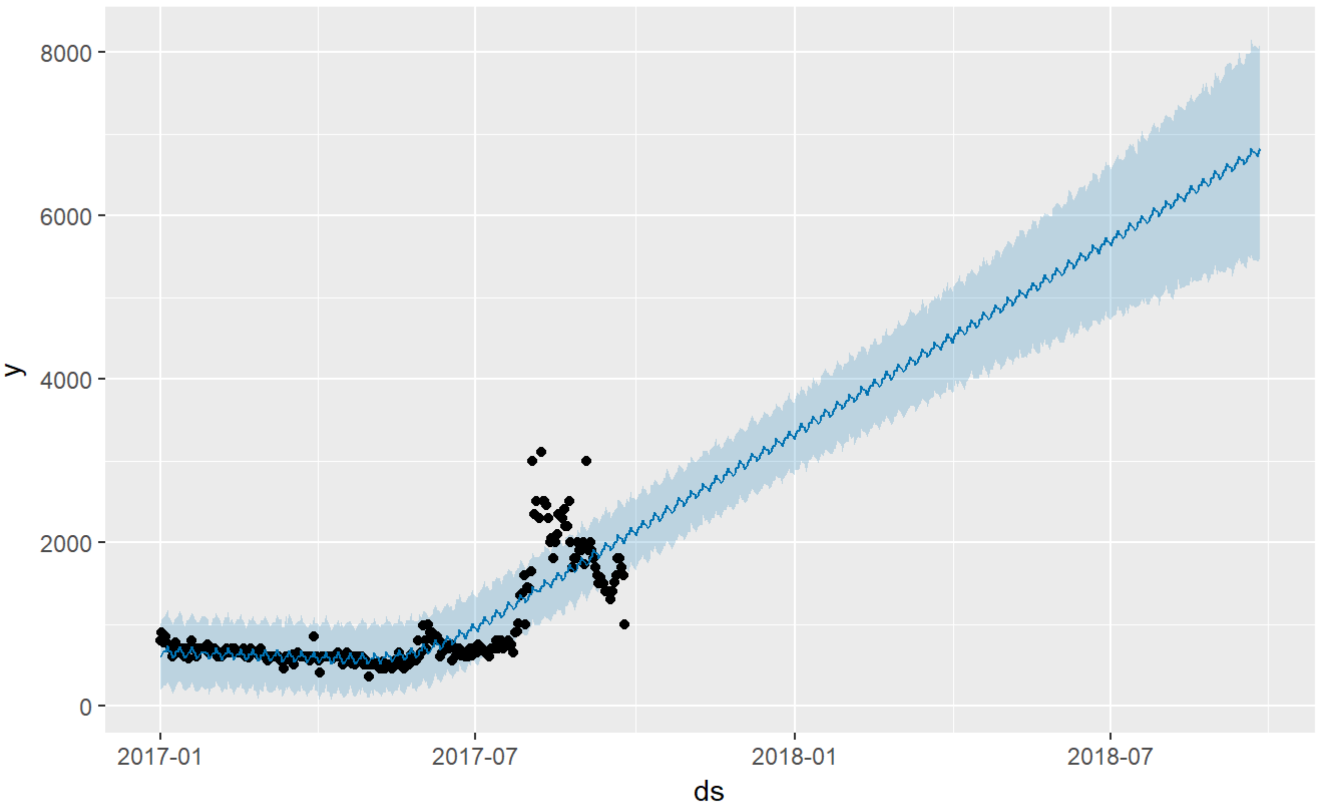
```
future <- make_future_dataframe(m, periods = 365)
head(future)
```

```
##           ds
## 1 2017-01-01
## 2 2017-01-02
## 3 2017-01-03
## 4 2017-01-04
## 5 2017-01-05
## 6 2017-01-06
```

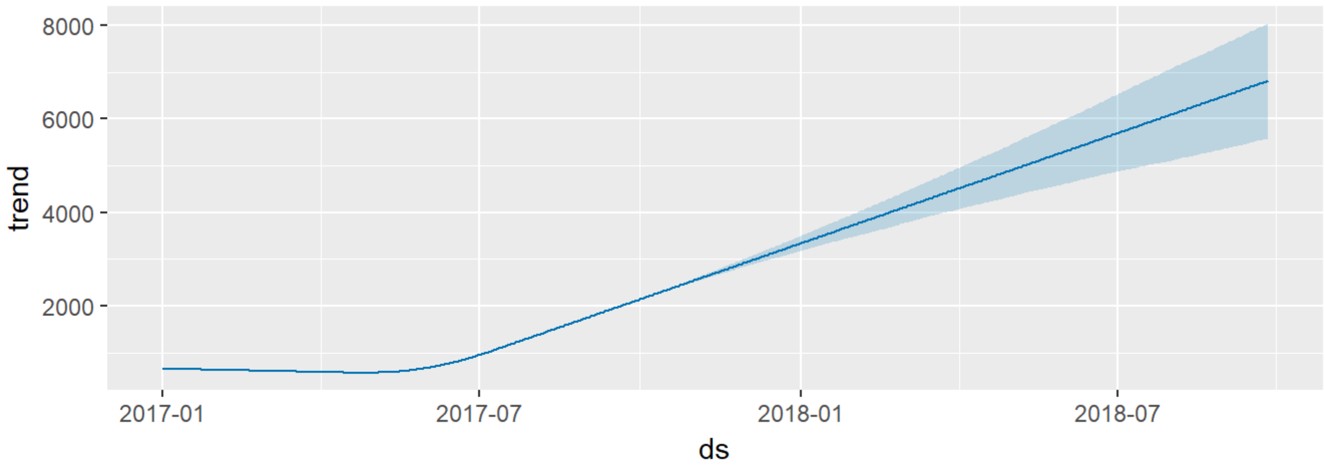
```
forecast <- predict(m, future)
head(forecast)
```

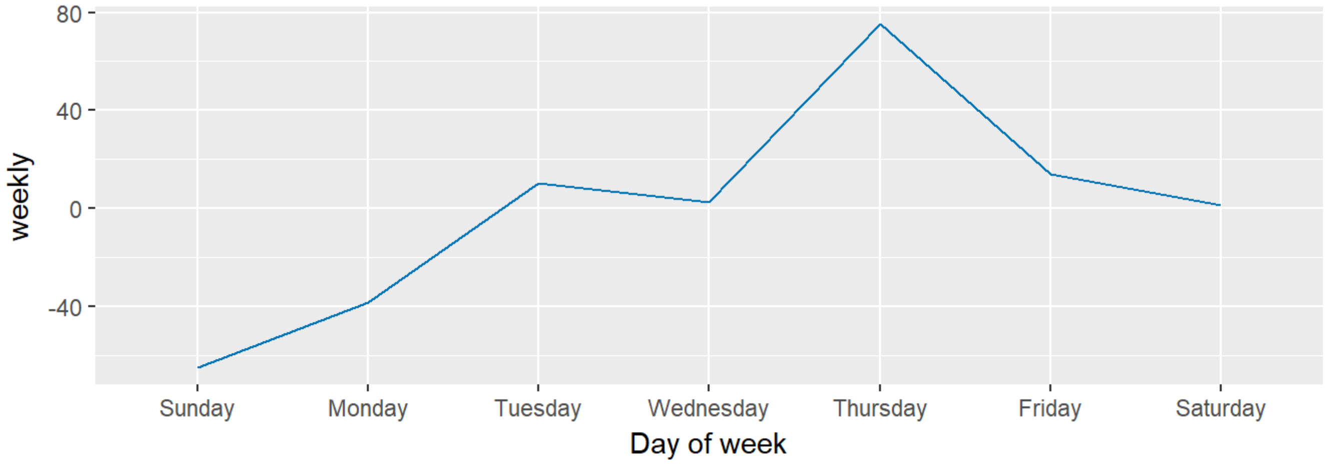
```
##           ds    trend   seasonal seasonal_lower seasonal_upper
## 1 2017-01-01 663.4488 -64.951710     -64.951710     -64.951710
## 2 2017-01-02 662.6475 -38.285602     -38.285602     -38.285602
## 3 2017-01-03 661.8461  10.334934      10.334934      10.334934
## 4 2017-01-04 661.0447   2.408376       2.408376       2.408376
## 5 2017-01-05 660.2434  75.197037      75.197037      75.197037
## 6 2017-01-06 659.4420  13.818605      13.818605      13.818605
##   seasonalities seasonalities_lower seasonalities_upper     weekly
## 1    -64.951710          -64.951710          -64.951710 -64.951710
## 2    -38.285602          -38.285602          -38.285602 -38.285602
## 3     10.334934           10.334934           10.334934  10.334934
## 4      2.408376            2.408376            2.408376   2.408376
## 5     75.197037           75.197037           75.197037  75.197037
## 6     13.818605           13.818605           13.818605  13.818605
##   weekly_lower weekly_upper yhat_lower yhat_upper trend_lower trend_upper
## 1   -64.951710   -64.951710   195.4580   1001.700    663.4488    663.4488
## 2   -38.285602   -38.285602   211.9747   1054.698    662.6475    662.6475
## 3    10.334934    10.334934   247.3341   1082.770    661.8461    661.8461
## 4     2.408376     2.408376   242.3489   1120.200    661.0447    661.0447
## 5    75.197037    75.197037   293.0296   1172.860    660.2434    660.2434
## 6    13.818605    13.818605   265.0934   1090.098    659.4420    659.4420
##       yhat
## 1 598.4971
## 2 624.3619
## 3 672.1810
## 4 663.4531
## 5 735.4404
## 6 673.2606
```

```
plot(m, forecast)
```



```
prophet_plot_components(m, forecast)
```


trend
8000
6000
4000
2000
2017-01
2017-07
2018-01
2018-07
ds


weekly
80
40
0
-40
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week