Onion Assignment - Shahid Khan

Load all packages

```
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(ggplot2)
library(httr)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(stringr)
library(prophet)
```

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

Aquire the daily onion data from nhrdf.org

```
getwd()
```

```
## [1] "C:/bigdata"
```

```
setwd("C:/bigdata")
page=read_html("DailyWiseMarketArrivals_2016.html")
table_node=html_node(page,"#dnn_ctr966_DailyWiseMarketArrivals_GridView1")
table=html_table(table_node)
str(table)
```

```
## 'data.frame':    19481 obs. of  6 variables:
##  $ Date                : chr  "10/Jun/2016" "28/Jun/2016" "29/Jun/2016" "30/Jun/2016" ...
##  $ Market              : chr  "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" ...
##  $ Arrival(q)          : int  350 300 30 30 60 400 150 200 200 40 ...
```

```
##  $ Price Minimum (Rs/q): chr  "450" "540" "550" "550" ...
##  $ Price Maximum (Rs/q): chr  "650" "670" "850" "850" ...
##  $ Modal Price (Rs/q)  : chr  "550" "600" "650" "650" ...
```

```
df=table
```

Start refining the data by putting in column names which are meaningful

```
dim(df)
```

```
## [1] 19481     6
```

```
column_name = c('date','market','quantity','priceMin','priceMax','priceMod')
colnames(df)=column_name
df$priceMax=as.numeric(df$priceMax)
```

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

```
df$priceMin=as.numeric(df$priceMin)
```

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

```
df$priceMod=as.numeric(df$priceMod)
```

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

```
df$date = as.Date(df$date,format = "%d/%B/%Y")
str(df)
```

```
## 'data.frame':    19481 obs. of  6 variables:
##  $ date    : Date, format: "2016-06-10" "2016-06-28" ...
##  $ market  : chr  "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" ...
##  $ quantity: int  350 300 30 30 60 400 150 200 200 40 ...
##  $ priceMin: num  450 540 550 550 650 650 750 650 650 750 ...
##  $ priceMax: num  650 670 850 850 1065 ...
##  $ priceMod: num  550 600 650 650 850 725 850 850 700 850 ...
```

Remove the NA from the dataset and mutate data

```
df=filter(df,!is.na(priceMod))
df=filter(df,!is.na(priceMax))
df=filter(df,!is.na(priceMin))
df1 <- df %>% mutate(market1 = market) %>% separate(market1,c("city", "state"),sep="\\(")
```

```
## Warning: Too many values at 654 locations: 4479, 4480, 4481, 4482, 4483,
## 4484, 4485, 4486, 4487, 4488, 4489, 4490, 4491, 4492, 4493, 4494, 4495,
## 4496, 4497, 4498, ...
```

```
## Warning: Too few values at 3204 locations: 1898, 1899, 1900, 1901, 1902,
## 1903, 1904, 1905, 1906, 1907, 1908, 1909, 1910, 1911, 1912, 1913, 1914,
## 1915, 1916, 1917, ...
```

```
unique(df1$state)
```

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

Data refining for cities and states

```
str(df1)
```

```
## 'data.frame':    19480 obs. of  8 variables:
##  $ date    : Date, format: "2016-06-10" "2016-06-28" ...
##  $ market  : chr  "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" ...
##  $ quantity: int  350 300 30 30 60 400 150 200 200 40 ...
##  $ priceMin: num  450 540 550 550 650 650 750 650 650 750 ...
##  $ priceMax: num  650 670 850 850 1065 ...
##  $ priceMod: num  550 600 650 650 850 725 850 850 700 850 ...
##  $ city    : chr  "ABOHAR" "ABOHAR" "ABOHAR" "ABOHAR" ...
##  $ state   : chr  "PB)" "PB)" "PB)" "PB)" ...
```

```
#Find out cities which are not a duplicate
citiesWithNoState = df1 %>% filter(is.na(state))
citiesWithNoState = citiesWithNoState[!duplicated(citiesWithNoState[,c("city")]),]
#put all cities with no state with either
#View(citiesWithNoState$city)
#find all states
statesWithNoDupes = df1[!duplicated(df1[,c("state")]),]
df1$state = str_replace(df1$state,"\\)","")
df1$state = str_trim(df1$state)

#find cities which have the following state codes
#F&V,KER,Others,RAJ,RJ,CHATT,CHGARH
df2=df1 %>% filter(state == c('KER','Others','RAJ','RJ','CHATT','CHGARH'))
```

```
## Warning in state == c("KER", "Others", "RAJ", "RJ", "CHATT", "CHGARH"):
## longer object length is not a multiple of shorter object length
```

```
df2=df2[!duplicated(df2[,c("state","city")]),]

sizeofdf2=nrow(df2)
#print the states identified with each of these cities
for(i in 1:sizeofdf2){
  cat(df2[i,]$city,df2[i,]$state,"\n")
}
```

```
## AJMER RAJ
## ALWAR RAJ
## BIKANER RAJ
## CHOMU RAJ
## JODHPUR RAJ
## KOTA RAJ
```

```
## PALAYAM KER
## PRATAPGARH  RJ
## RAIGARH  CHATT
## RAIPUR CHGARH
## SRIGANGANAGAR RAJ
## TIPHRA  CHATT
## UDAIPUR RAJ
```

```
# from the output below, we can make the city pratapgarh to be same as others in RAJ
# In addition, CHATT and CHGARG are same state i.e. chhatisgarh
df1$state=str_replace(df1$state,"CHATT","CHGARH")
df1$state=str_replace(df1$state,"RJ","RAJ")
df1$state=str_replace(df1$state,"TELANGANA","Telangana")
df1$state=str_replace(df1$state,"Others","DELHI")
df1$state=str_replace(df1$state,"F\\&V","MP")
#remove special characters like "." from states
df1$state=str_replace_all(df1$state,"\\.","")

#first find all cities which dont have a state
allCities = df1 %>% filter(is.na(state))
unique(allCities[,c('city')])
```

```
##  [1] "BANGALORE"    "BHOPAL"       "BULANDSHAHR"  "CHANDIGARH"
##  [5] "CHENNAI"      "DELHI"        "GUWAHATI"     "HYDERABAD"
##  [9] "IMPHAL"       "JAIPUR"       "JAMMU"        "KOLKATA"
## [13] "LUCKNOW"      "MUMBAI"       "NAGPUR"       "PATNA"
## [17] "SHAHJAHANPUR"
```

```
#Now start replacing all of them with appropriate state
df1$state[df1$city=='DELHI'] <- 'DELHI'
df1$state[df1$city=='BANGALORE'] <- 'KNT'
df1$state[df1$city=='BULANDSHAHR'] <- 'UP'
df1$state[df1$city=='SHAHJAHANPUR'] <- 'UP'
df1$state[df1$city=='CHENNAI'] <- 'TN'
df1$state[df1$city=='MUMBAI'] <- 'MS'
df1$state[df1$city=='NAGPUR'] <- 'MS'
df1$state[df1$city=='JAIPUR'] <- 'RAJ'
df1$state[df1$city=='HYDERABAD'] <- 'Telangana'
df1$state[df1$city=='GUWAHATI'] <- 'AS'
df1$state[df1$city=='PATNA'] <- 'BHR'
df1$state[df1$city=='IMPHAL'] <- 'Meghalaya'
df1$state[df1$city=='KOLKATA'] <- 'WB'
df1$state[df1$city=='LUCKNOW'] <- 'UP'
df1$state[df1$city=='BHOPAL'] <- 'MP'
df1$state[df1$city=='CHANDIGARH'] <- 'PB'
df1$state[df1$city=='JAMMU'] <- 'JK'
df1$state[df1$city=='Others'] <- 'Delhi'
df1$state[df1$city=="F\\&V "] <- 'MP'

#Below lines of code are just to check whether there could be other rows
#with Others as state
allFAndVState = df1 %>% filter(state == "Others")
str(df1)
```

```
## 'data.frame':    19480 obs. of  8 variables:
##  $ date    : Date, format: "2016-06-10" "2016-06-28" ...
```

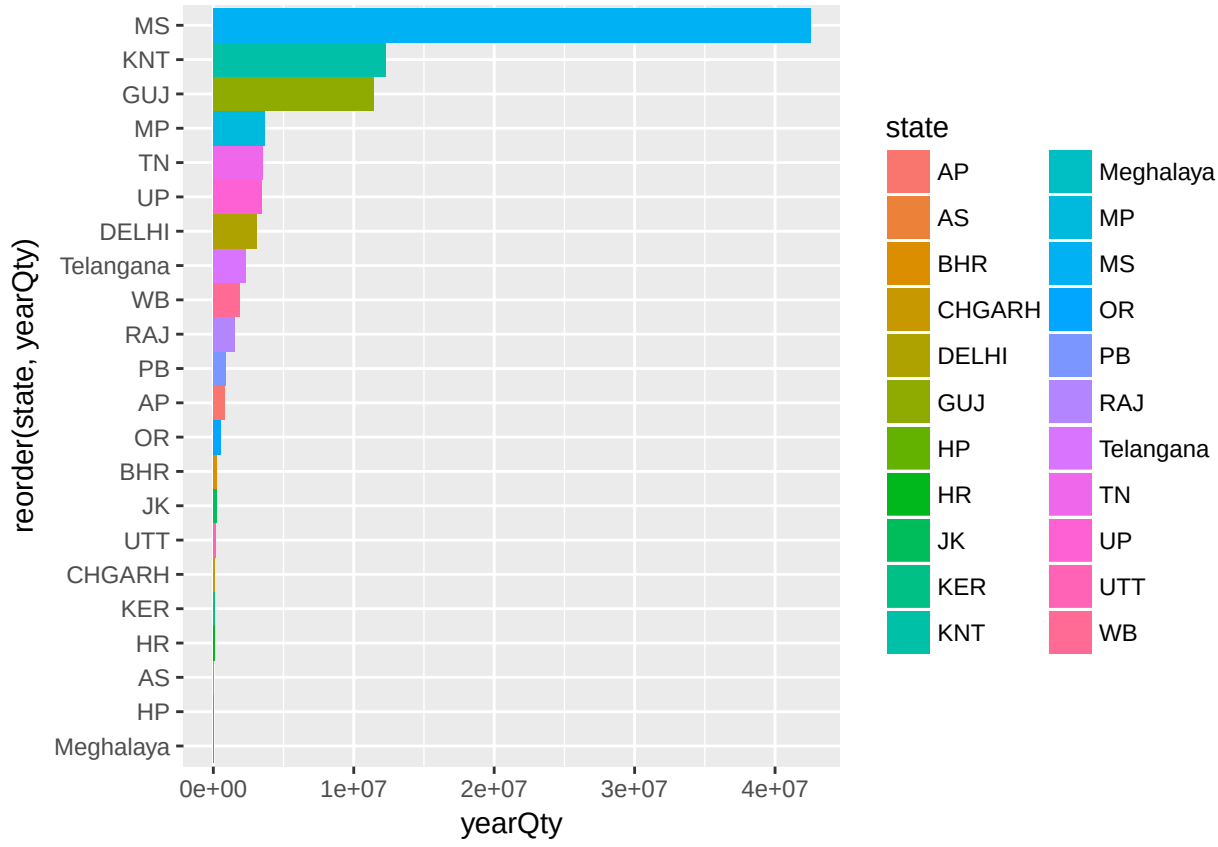
```
##  $ market  : chr  "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" "ABOHAR(PB)" ...
##  $ quantity: int  350 300 30 30 60 400 150 200 200 40 ...
##  $ priceMin: num  450 540 550 550 650 650 750 650 650 750 ...
##  $ priceMax: num  650 670 850 850 1065 ...
##  $ priceMod: num  550 600 650 650 850 725 850 850 700 850 ...
##  $ city    : chr  "ABOHAR" "ABOHAR" "ABOHAR" "ABOHAR" ...
##  $ state   : chr  "PB" "PB" "PB" "PB" ...
```

sum up the quantity and arrange them in desc order to find the highest consuming state

```
dfQty = df1 %>% group_by(state) %>% summarize(yearQty=sum(quantity)) %>% arrange(desc(yearQty))
summary(dfQty)
```

```
##     state              yearQty
##  Length:22          Min.   :    1836
##  Class :character   1st Qu.:  123452
##  Mode  :character   Median :  848730
##                     Mean   : 4041956
##                     3rd Qu.: 3355468
##                     Max.   :42520605
```

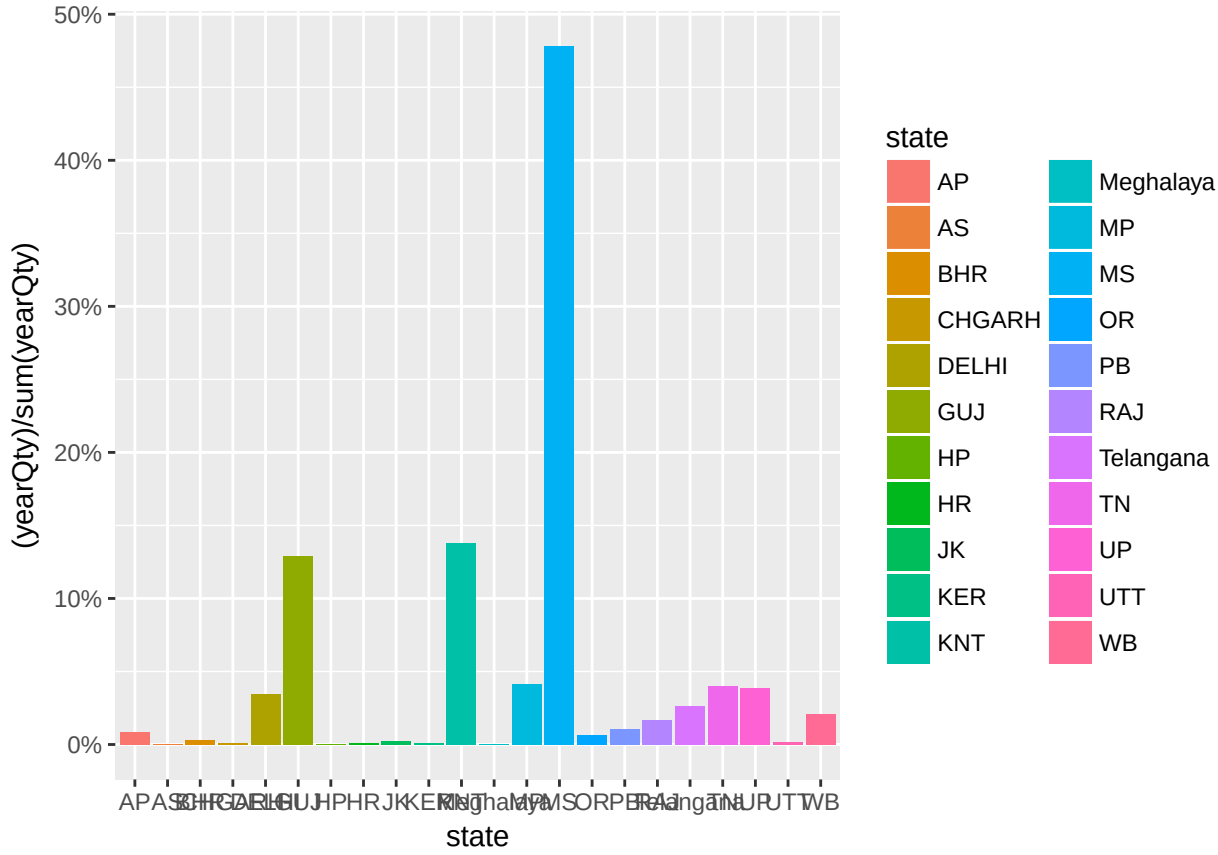
```
#plot the usage
ggplot(data=dfQty) + aes(reorder(state,yearQty),yearQty,fill=state) + geom_col(width=1) +coord_flip()
```



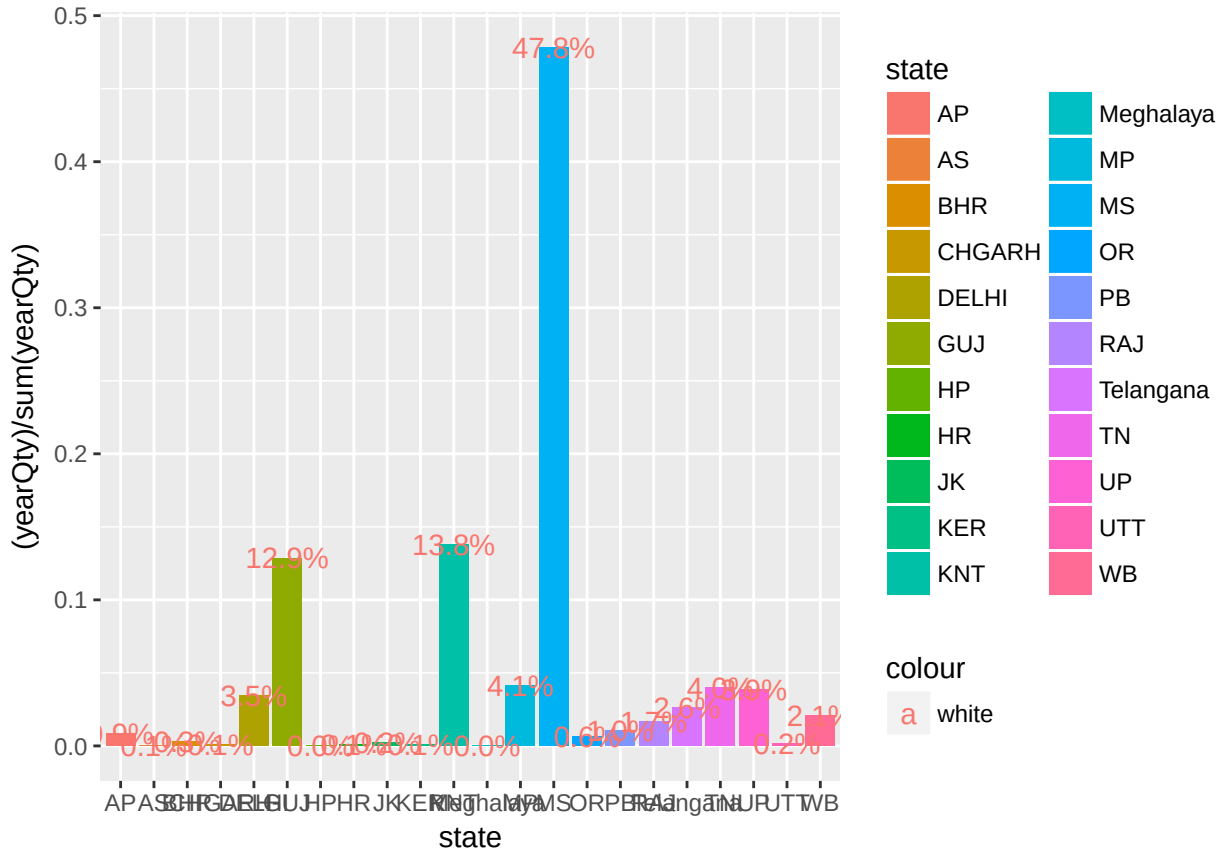
What is the percent usage from the above

```
#In terms of percentage
ggplot(dfQty,aes(fill=state)) +
        geom_bar(aes(x=state,y = (yearQty)/sum(yearQty)),geom ="text",stat = "identity") + scale_y_con
```

```
## Warning: Ignoring unknown parameters: geom
```

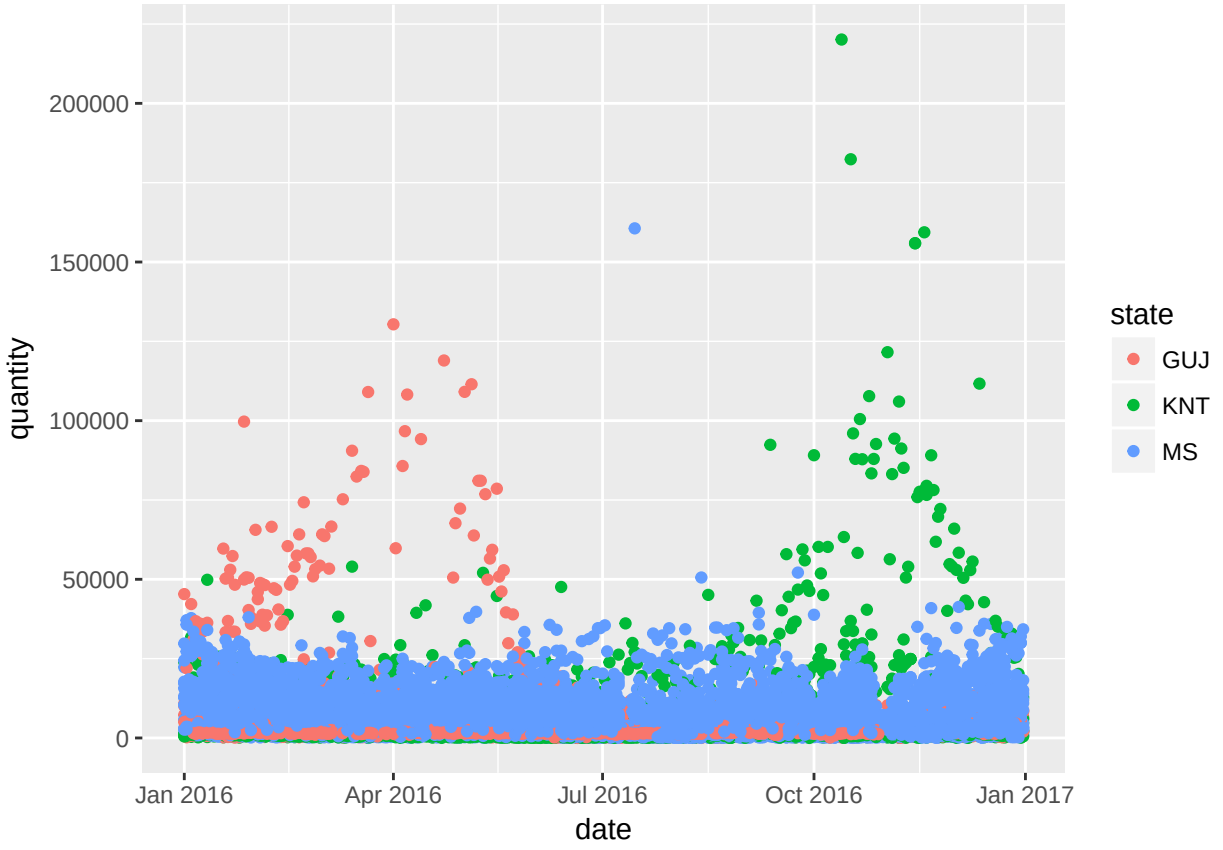



```
#% as a label on each state
ggplot(dfQty,aes(x=state, y=(yearQty)/sum(yearQty), fill=state)) +
  geom_bar(stat="identity") + geom_text(aes(label=paste0(sprintf("%1.1f", (yearQty)/sum(yearQty)*100),"
```



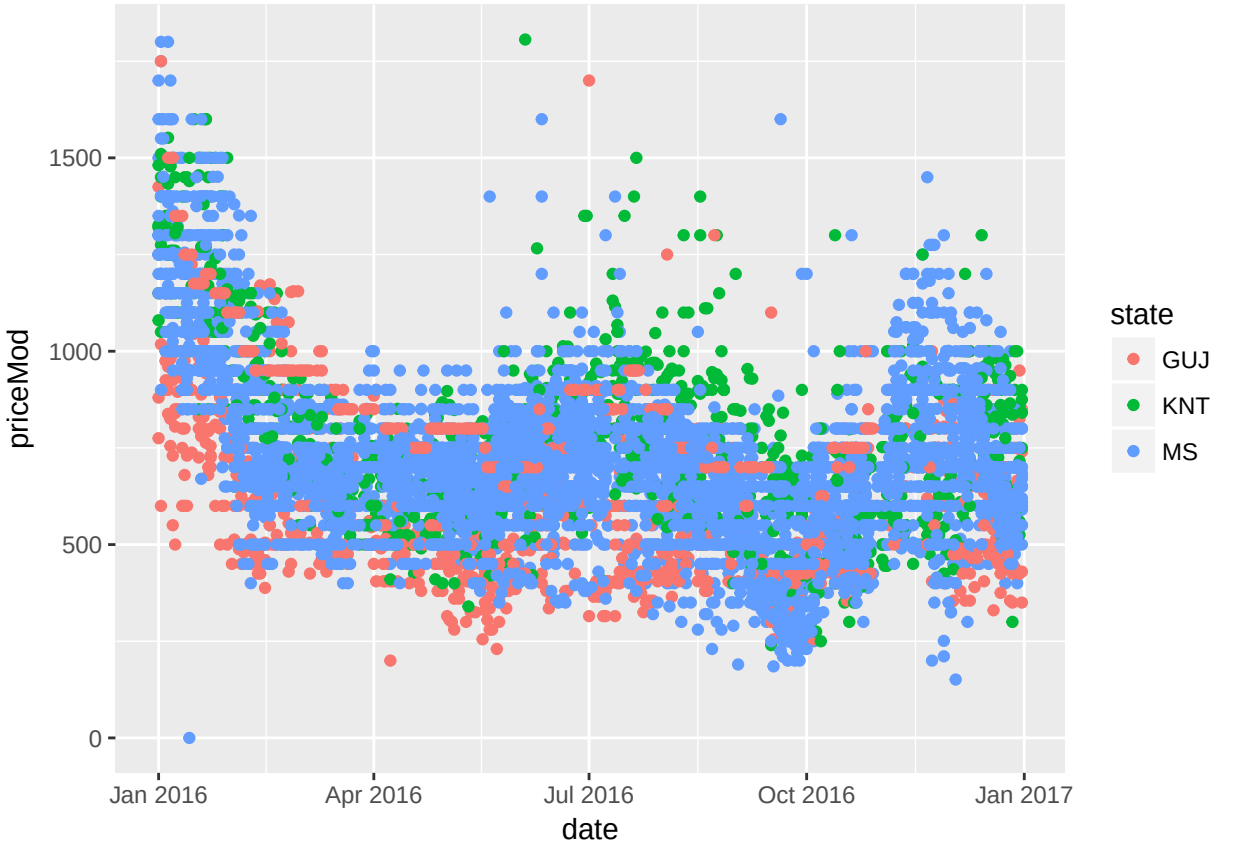
scatter plot for consumption in top 3 states

```
ggplot(subset(df1,state %in% c("MS","KNT","GUJ")),aes(x=date,y=quantity,col=state)) + geom_point()
```



PriceTrend in the top 3 states

```
ggplot(subset(df1,state %in% c("MS","KNT","GUJ")),aes(x=date,y=priceMod,col=state)) + geom_point()
```



Predict prices for the next 30 days in the state which consumes the most

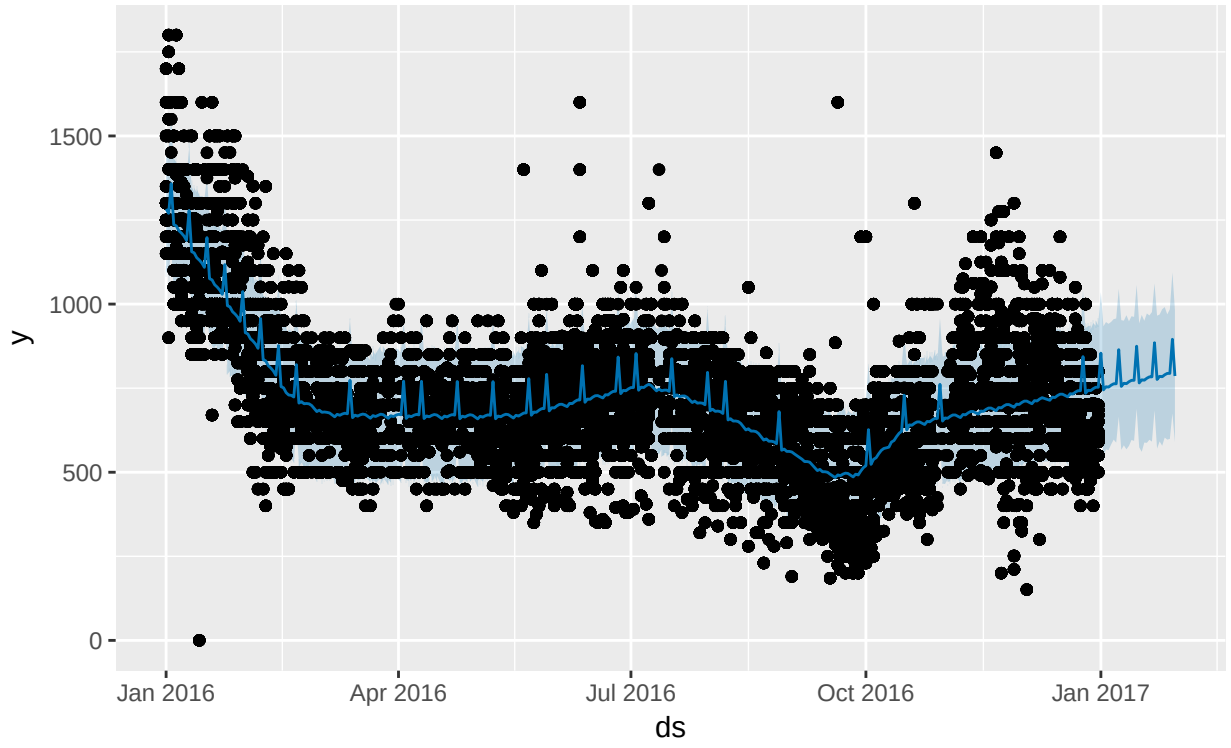
```
dfBang = df1 %>% filter(state=="MS") %>% select(date,priceMod)
colnames(dfBang) = c('ds','y')
m=prophet(dfBang)
```

```
## 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 = -89.0131
## Optimization terminated normally:
##   Convergence detected: relative gradient magnitude is below tolerance
```

```
future = make_future_dataframe(m,period=30,freq = 'd')
forecast = predict(m,future)
plot(m,forecast)
```



Identify some forecast trend

```
str(forecast)
```

```
## 'data.frame':    5582 obs. of  16 variables:
##  $ ds                : POSIXct, format: "2016-01-01" "2016-01-01" ...
##  $ trend             : num  1293 1293 1293 1293 1293 ...
##  $ seasonal          : num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ seasonal_lower    : num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ seasonal_upper    : num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ seasonalities     : num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ seasonalities_lower: num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ seasonalities_upper: num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ weekly            : num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ weekly_lower      : num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ weekly_upper      : num  -10.1 -10.1 -10.1 -10.1 -10.1 ...
##  $ yhat_lower        : num  1082 1088 1106 1093 1097 ...
##  $ yhat_upper        : num  1477 1467 1481 1478 1465 ...
##  $ trend_lower       : num  1293 1293 1293 1293 1293 ...
##  $ trend_upper       : num  1293 1293 1293 1293 1293 ...
##  $ yhat              : num  1283 1283 1283 1283 1283 ...
```

```
max(forecast$trend)
```

```
## [1] 1292.795
```

```r
min(forecast$trend)
```

```
## [1] 505.6451
```

Conclusions

1. Maharashtra is top consuming state
2. Maharashtra, Karnataka and Gujarat are the top 3 consuming states. Together they consume ~75% of the overall produce.
3. In Gujarat, there is a spike in consumption in the first half of the year, in karnataka the consumption spikes in the 4th quarter. In Maharashtra the consumption is constant throughout the year.
4. Price is similar across the top 3 states throughout