ShahidKhan_Aug_SVAP_Asmt_R2

Shahid Khan

7 October 2017

Framing of question

1. Find out the state with the best GDP growth over the last couple of decades
2. Identify if there is any relationship between GDP and state revenues or fiscal deficit
3. A fiscal deficit occurs when a government's total expenditures exceed the revenue that it generates, excluding money from borrowings.

load all libraries

```
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(reshape2)
library(stringr)
library(prophet)
```

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

```
library(caTools)
```

Acquire the data

```
#The data for this is available on niti aayog website http://niti.gov.in/state-statistics
# Download the excel files available and then remove metadata from the excel files
aggregateExpense <- readxl::read_xlsx("C:/bigdata/download/Aggregate Expenditure.xlsx")
capitalExpense <- readxl::read_xlsx("C:/bigdata/download/Capital Expenditure.xlsx")
revenueExpense <- readxl::read_xlsx("C:/bigdata/download/Revenue Expenditure.xlsx")
socialExpense <- readxl::read_xlsx("C:/bigdata/download/Social Sector Expenditure.xlsx")
fiscalDeficit <- readxl::read_xlsx("C:/bigdata/download/Fiscal Deficits.xlsx")
nominalGDP <- readxl::read_xlsx("C:/bigdata/download/Nominal GSDP Series.xlsx")
taxRevenue <- readxl::read_xlsx("C:/bigdata/download/Own Tax Revenues.xlsx")
```

Start refining the data

```
#create an additional column storing category variable for each of the vectors
#For data which is missing in some columns store NA

aggregateExpense$category <- "AggregateExpense"
capitalExpense$category <- "CapitalExpense"
revenueExpense$catetory <- "RevenueExpense"
socialExpense$`2016-17` <- NA
socialExpense$category <- "SocialExpense"
fiscalDeficit$`2016-17` <- NA
fiscalDeficit$category <- "FiscalDeficit"
nominalGDP$`2016-17` <- NA
nominalGDP$category <- "NominalGDP"
taxRevenue$category <- "TaxRevenue"

#Identiy column names
colnames(fiscalDeficit)
```

```
##  [1] "Year"         "1980-81"      "1981-82"      "1982-83"
##  [5] "1983-84"      "1984-85"      "1985-86"      "1986-87"
##  [9] "1987-88"      "1988-89"      "1989-90"      "1990-91"
## [13] "1991-92"      "1992-93"      "1993-94"      "1994-95"
## [17] "1995-96"      "1996-97"      "1997-98"      "1998-99"
## [21] "1999-00"      "2000-01"      "2001-02"      "2002-03"
## [25] "2003-04"      "2004-05"      "2005-06"      "2006-07"
## [29] "2007-08"      "2008-09"      "2009-10"      "2010-11"
## [33] "2011-12"      "2012-13"      "2013-14"      "2014-15 (RE)"
## [37] "2015-16 (BE)" "2016-17"      "category"
```

```
#Rename column names
column_name = c("state","1980-81","1981-82","1982-83","1983-84","1984-85","1985-86","1986-87","1987-88"
                "2009-10","2010-11","2011-12","2012-13","2013-14","2014-15","2015-16","2016-17","catego
colnames(aggregateExpense) = column_name
colnames(capitalExpense) = column_name
colnames(revenueExpense) = column_name
colnames(socialExpense) = column_name
colnames(fiscalDeficit) = column_name
colnames(nominalGDP) = column_name
colnames(taxRevenue) = column_name

# Do a row binding of data
combined <- rbind(aggregateExpense,capitalExpense,revenueExpense,socialExpense,fiscalDeficit,nominalGDP

#Convert all numbers into numeric
convertColNames <- colnames(combined[2:38])
combined[,convertColNames] <-lapply(combined[,convertColNames,drop=FALSE],as.numeric)
```

```
## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
```

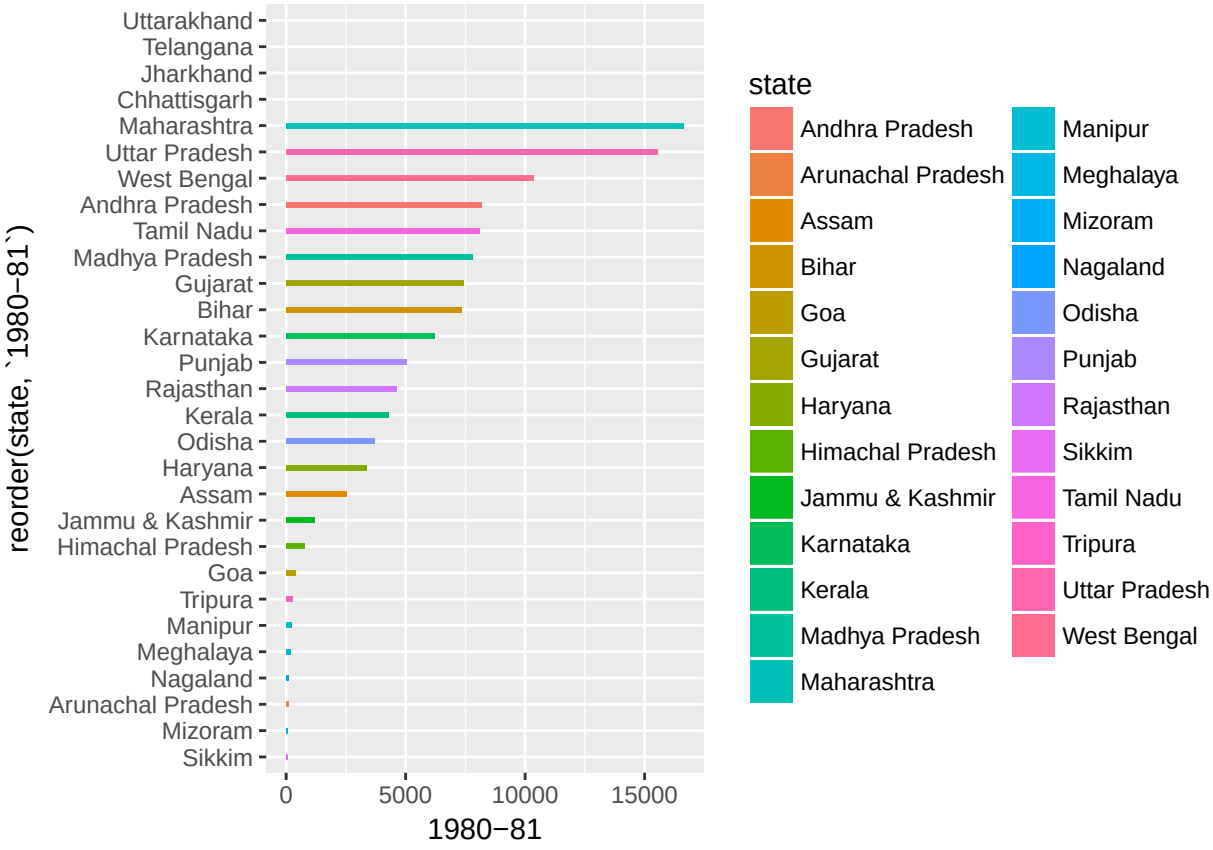
```
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
```

```
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion

## Warning in lapply(combined[, convertColNames, drop = FALSE], as.numeric):
## NAs introduced by coercion
```

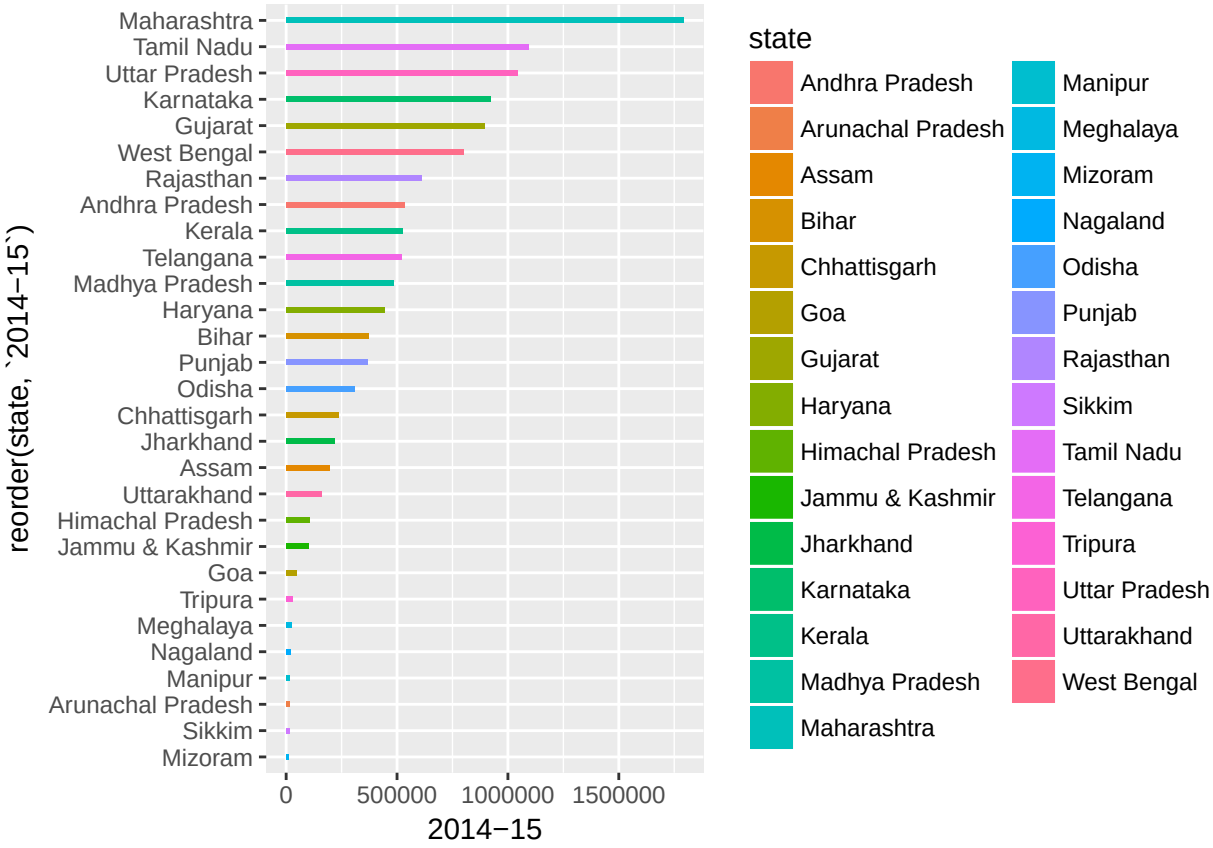
Visualization of data across various parameters

```r
#View the GDP figures of the year 1980-81 across all states
nominalGDP <- combined %>% filter(category =="NominalGDP") %>% arrange(`1980-81`)
ggplot(nominalGDP) + aes(reorder(state,`1980-81`),`1980-81`,fill=state) + geom_col(width = 0.2) + coord_
```

```
## Warning: Removed 4 rows containing missing values (position_stack).
```




```
#Plot for the year 2014-15 as that is the last year which has full data
nominalGDP <- combined %>% filter(category =="NominalGDP") %>% arrange(`2014-15`)
ggplot(nominalGDP) + aes(reorder(state,`2014-15`),`2014-15`,fill=state) + geom_col(width = 0.2) + coord_
```




```
# Clear observation from the above graphs is that the difference in growth
# between Maharashtra and UP is very big
```

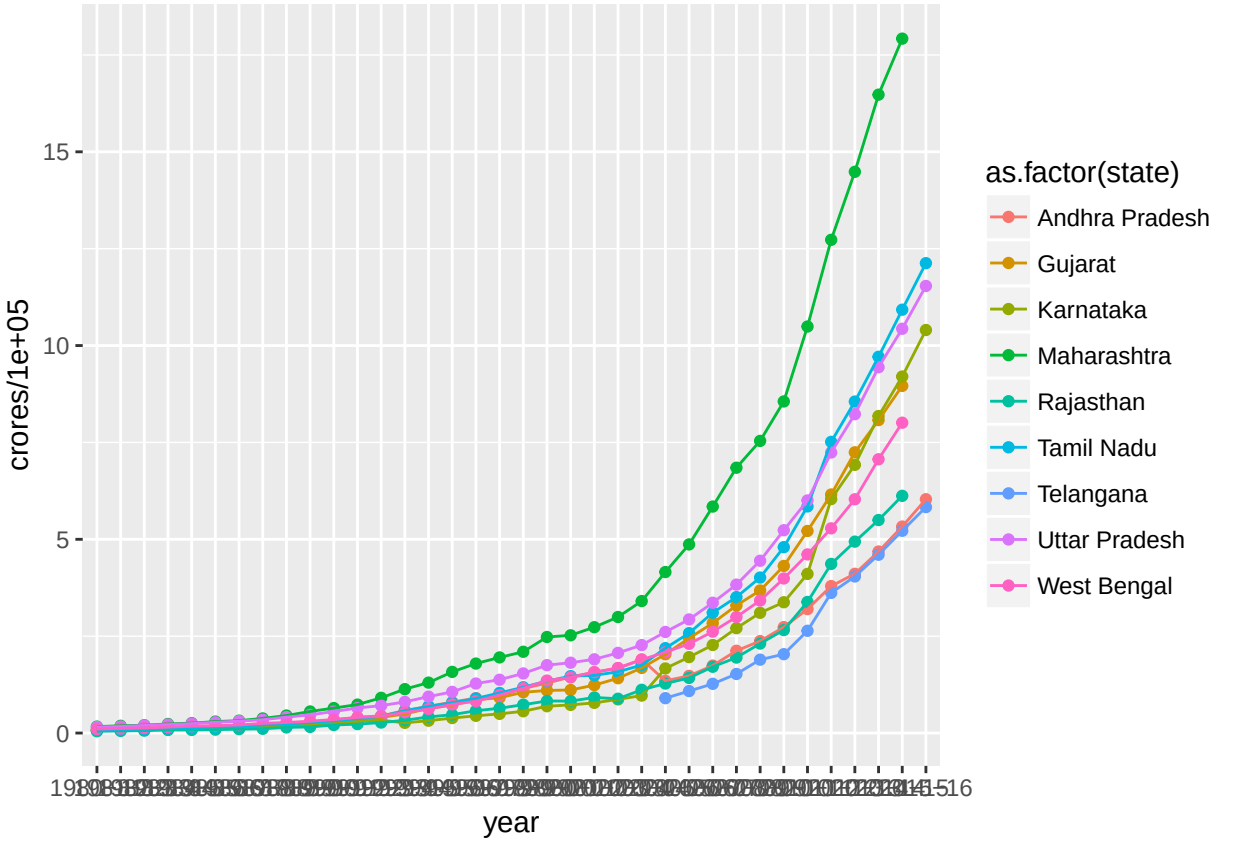
Rearrange the data and do visualization across all states and years

```
#use melt to move column based data to row based on state and category
combined_m <- melt(combined,id.vars= c("state","category"),measure.vars=c("1980-81","1981-82","1982-83"
                                                                           "2009-10","2010-11","2011-12","2012-13","201
colnames(combined_m) <- c("state","category","year","crores")
combined_m$crores <- as.numeric(combined_m$crores)
```

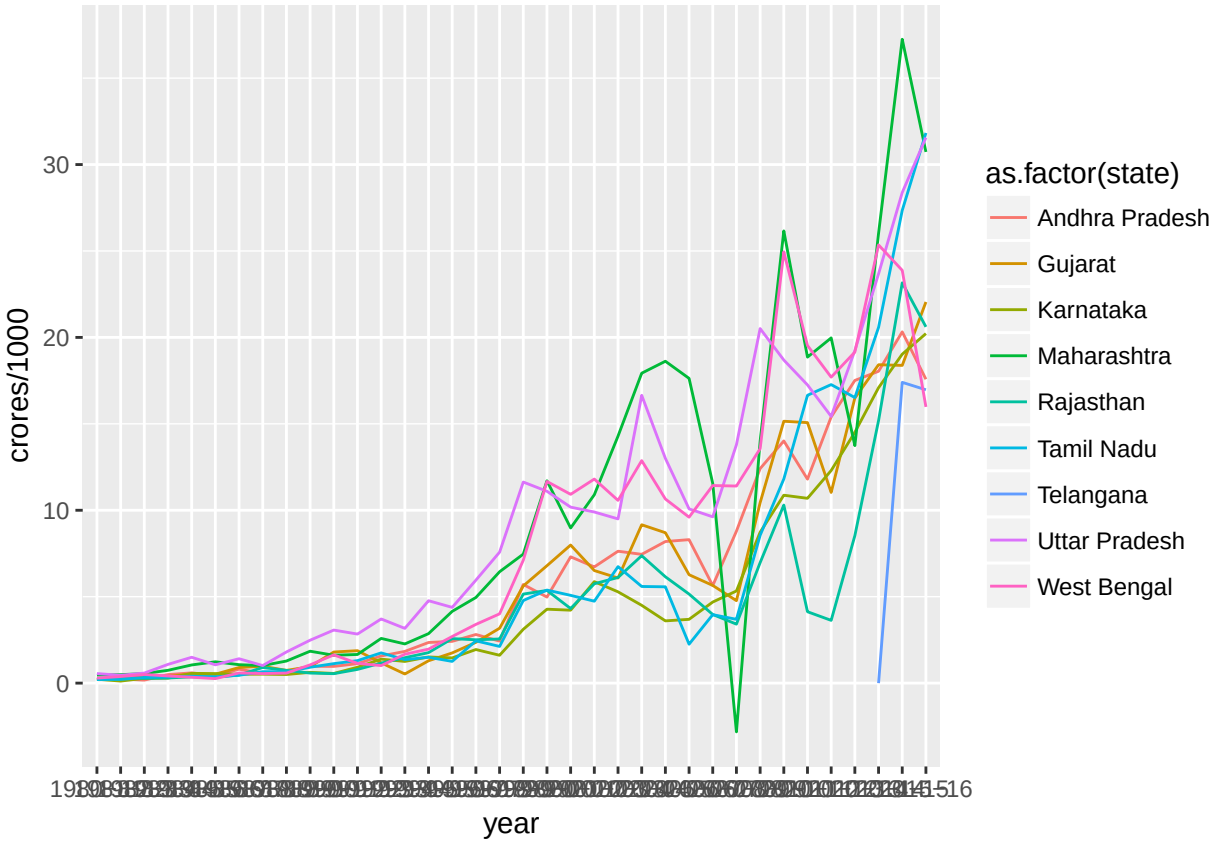
```
## Warning: NAs introduced by coercion
```

```
#delete data with na rows
delete.na <- function(DF, n=0) {
  DF[rowSums(is.na(DF)) <= n,]
}
combined_m <- delete.na(combined_m)

#Find the growth for the top 10 states
nominalGDP_m <- combined_m %>% filter(state %in% c("Maharashtra","Karnataka","Uttar Pradesh","Gujarat",
nominalGDP_m <- filter(nominalGDP_m,!is.na(crores))
nominalGDP_m <- delete.na(nominalGDP_m)
ggplot(nominalGDP_m, aes(y = crores/100000, x = year,
                         colour = as.factor(state))) +
```

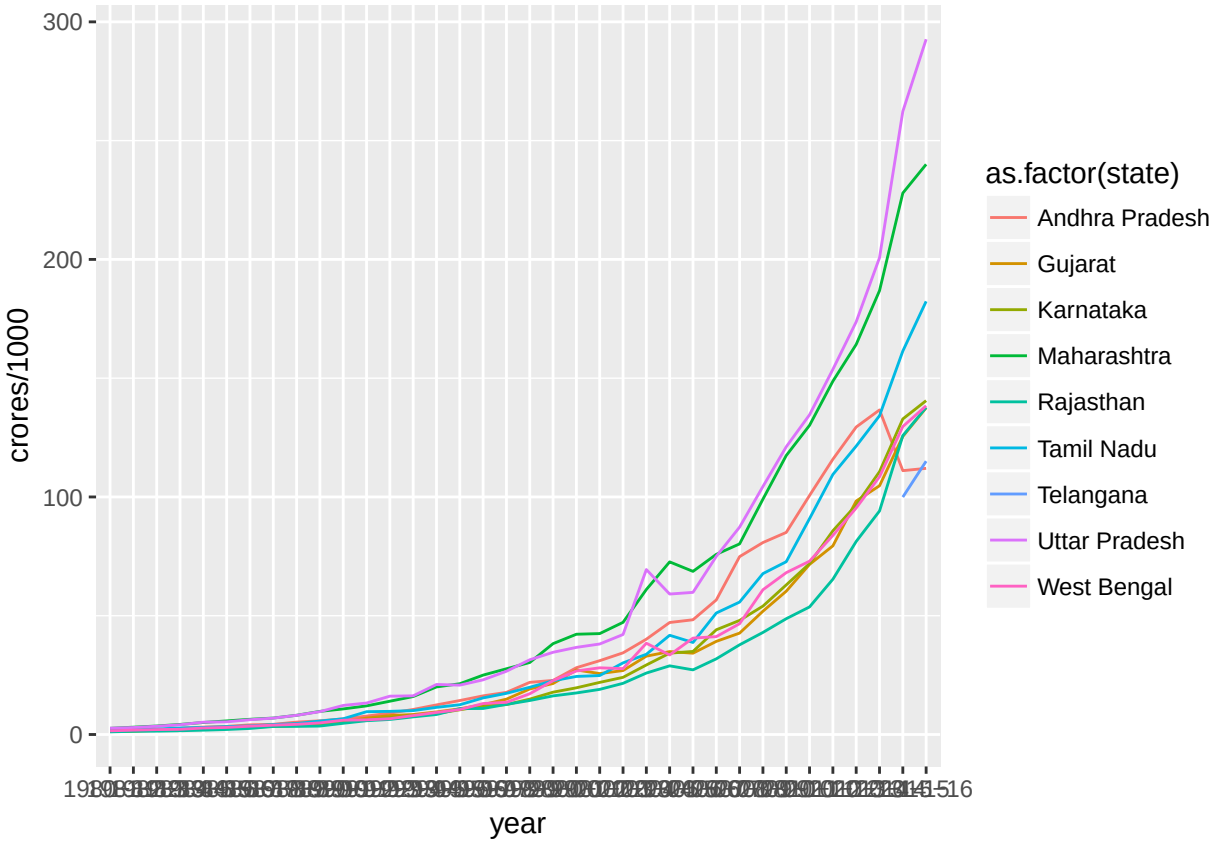
```
  geom_point()+
    geom_line(aes(group = state))
```



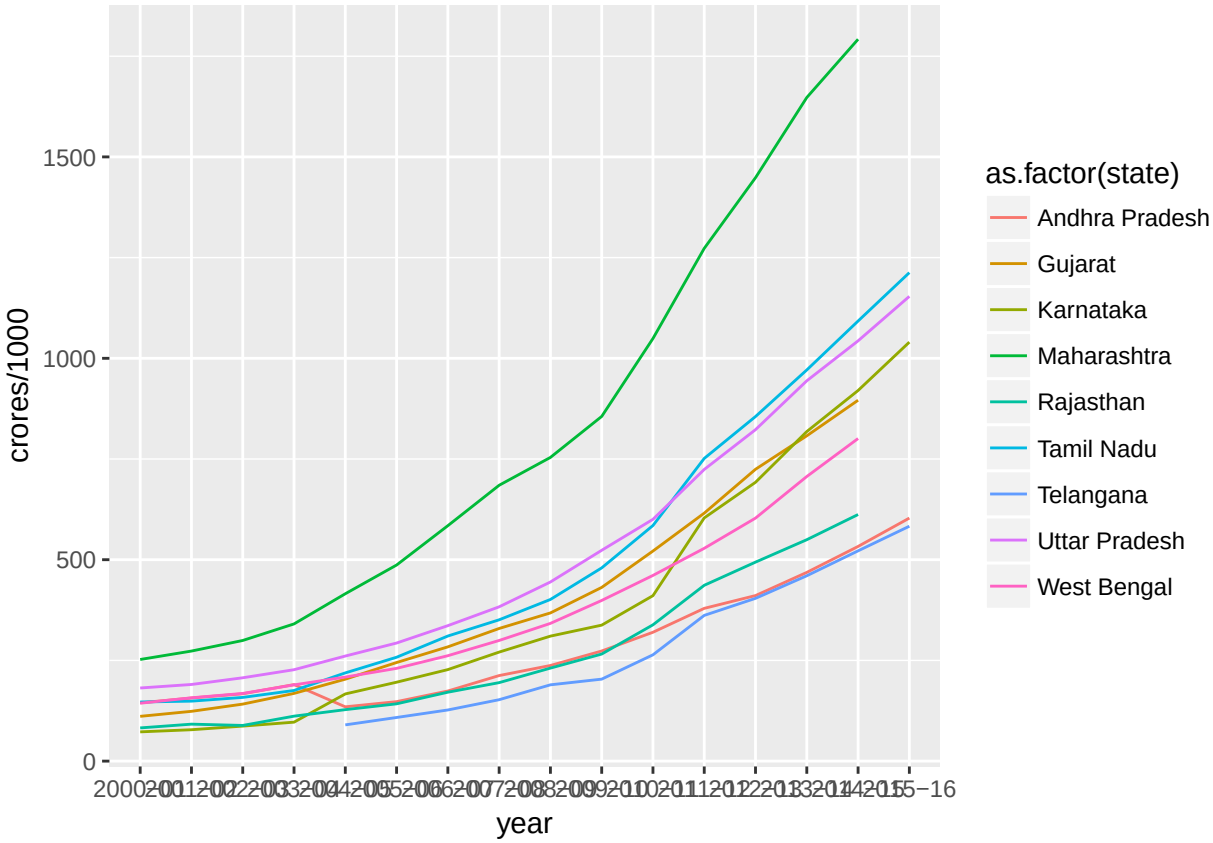

```
#Show line plot for all states for fiscal deficit
fiscalDeficit_m <- combined_m %>% filter(state %in% c("Maharashtra","Karnataka","Uttar Pradesh","Gujarat
fiscalDeficit_m <- filter(fiscalDeficit_m,!is.na(crores))
fiscalDeficit_m <- delete.na(fiscalDeficit_m)
ggplot(fiscalDeficit_m, aes(y = crores/1000, x = year,
                      colour = as.factor(state))) +
  geom_line(aes(group = state))
```



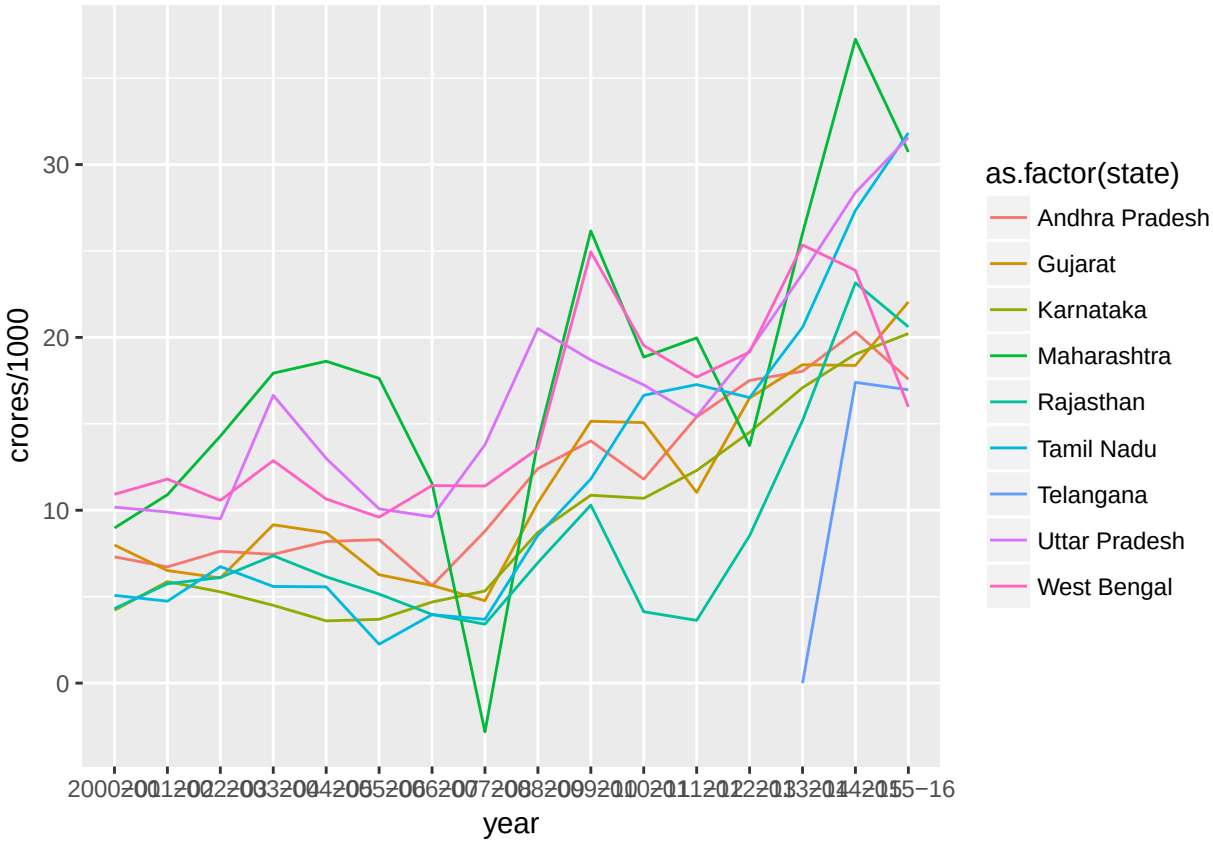

```
# Same for aggregate revenue
aggregate_m <- combined_m %>% filter(state %in% c("Maharashtra","Karnataka","Uttar Pradesh","Gujarat","
aggregate_m <- filter(aggregate_m,!is.na(crores))
aggregate_m <- delete.na(aggregate_m)
ggplot(aggregate_m, aes(y = crores/1000, x = year,
                        colour = as.factor(state))) +
  geom_line(aes(group = state))
```



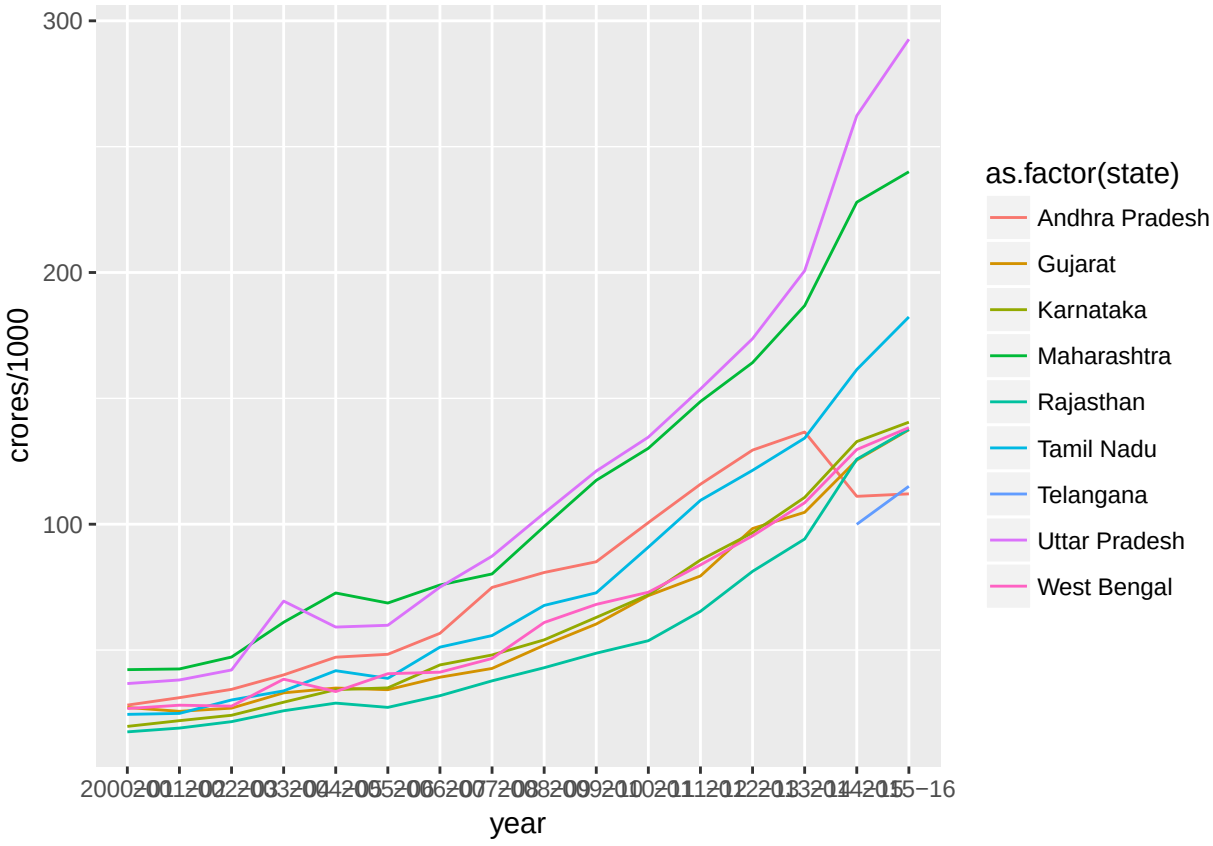

```
#Given that the plot looks clumsy plot all parameters only from the 2000s
nominalGDP_m2000s <- nominalGDP_m %>% filter(str_detect(year,"20"))
fiscalDeficit_m2000s <- fiscalDeficit_m %>% filter(str_detect(year,'20'))
aggregate_m2000s <- aggregate_m %>% filter(str_detect(year,'20'))

#Now plot for 2000 onwards
ggplot(nominalGDP_m2000s, aes(y = crores/1000, x = year,
                              colour = as.factor(state))) +
  geom_line(aes(group = state))
```



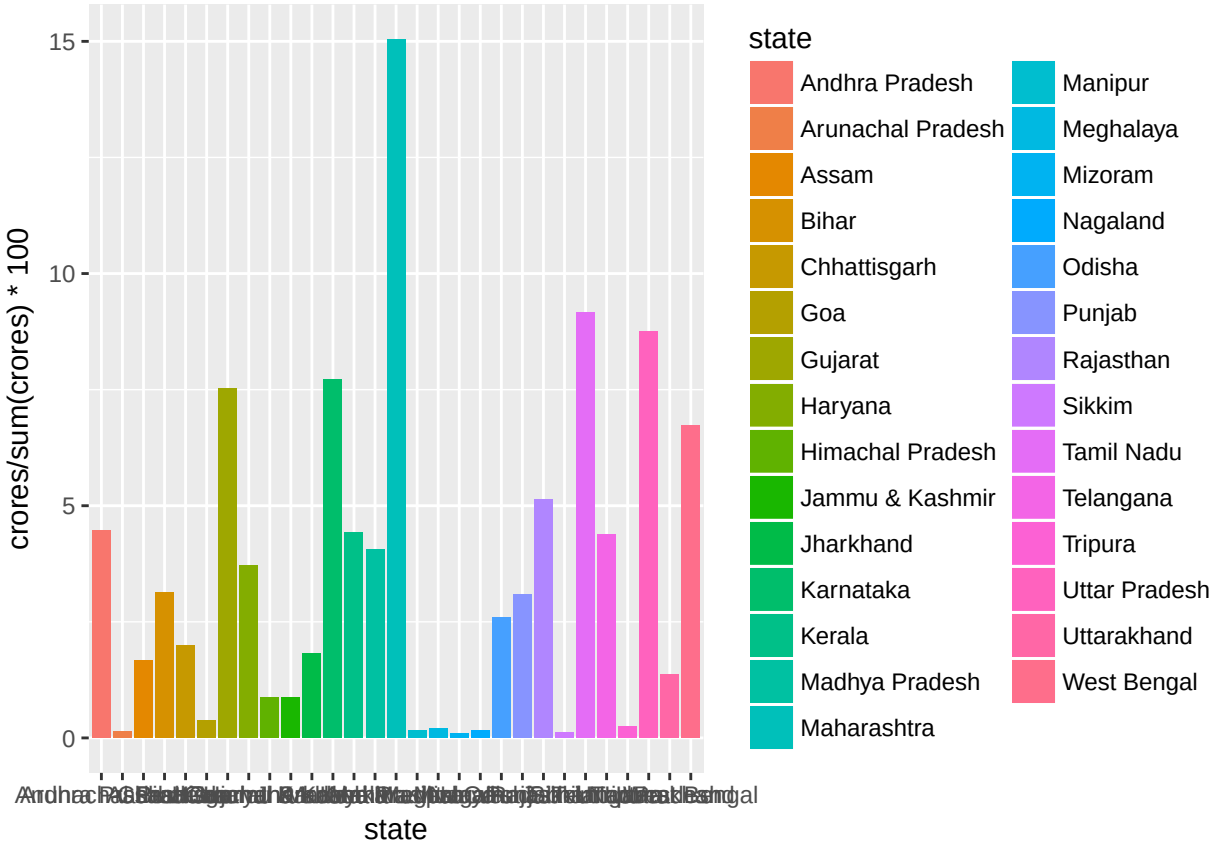

```
ggplot(fiscalDeficit_m2000s, aes(y = crores/1000, x = year,
                                 colour = as.factor(state))) +
  geom_line(aes(group = state))
```




```
ggplot(aggregate_m2000s, aes(y = crores/1000, x = year,
                             colour = as.factor(state))) +
  geom_line(aes(group = state))
```




```
# From the above UP has the highest aggregate expense, Maharashtra has the highest GDP

#Plot the % of overall GDP from states
statesNominalGDP <- combined_m %>% filter(year == "2014-15") %>% filter (state != "All States") %>% fil
ggplot() + geom_col(data=statesNominalGDP,aes(x=state,y=crores/sum(crores)*100,fill=state))
```



Find relationships between different variables for a given state

```
gdpMH <- nominalGDP_m %>% filter(state == "Maharashtra")
#remove data for 2015-16 from fiscal and aggregate as this is not available
#for gdp
fiscalMH <- fiscalDeficit_m %>% filter(state == "Maharashtra") %>% filter(year != "2015-16")
aggregateMH <- aggregate_m %>% filter(state == "Maharashtra") %>% filter(year != "2015-16")


cor(fiscalMH$crores,gdpMH$crores)
```

```
## [1] 0.8242331
```

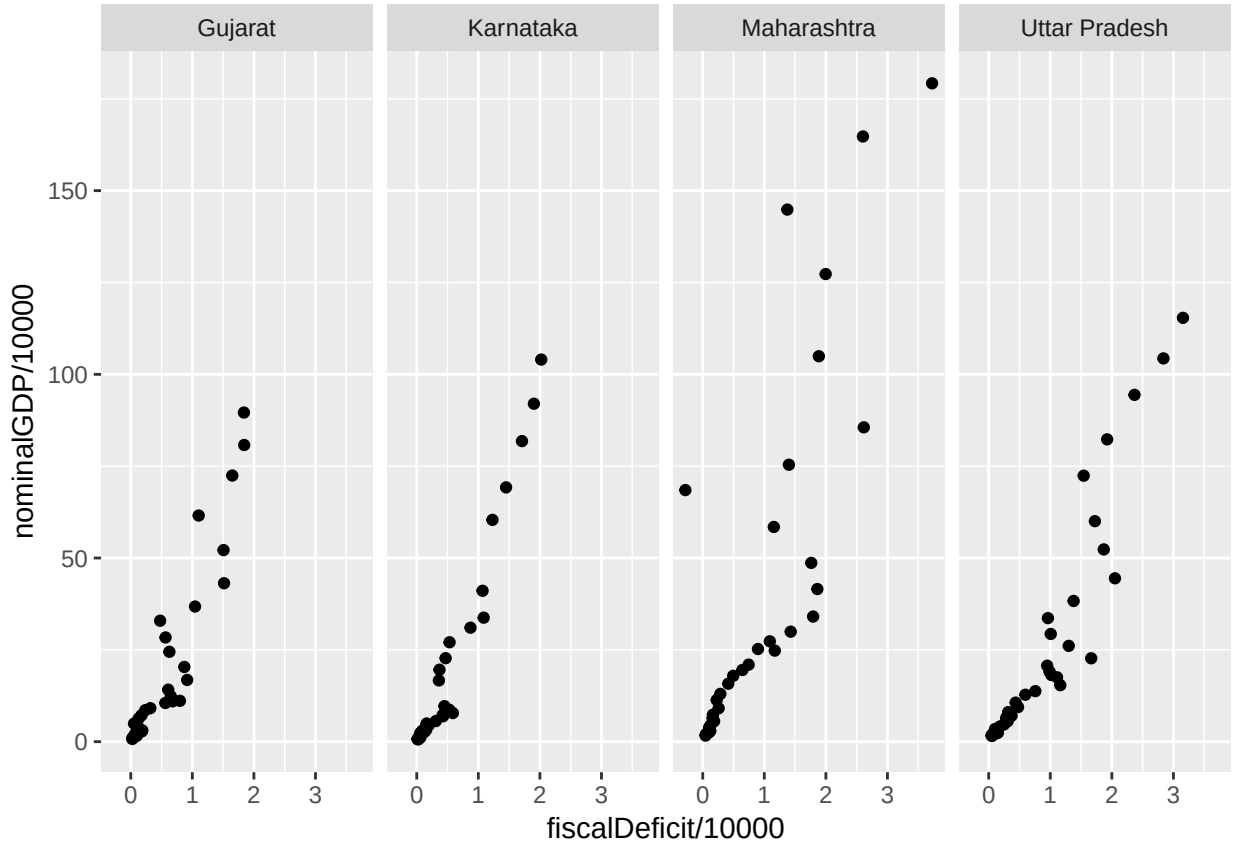
```
cor(aggregateMH$crores,gdpMH$crores)
```

```
## [1] 0.9933557
```

```
# For Maharashtra, India's highest GDP state both fiscal deficit and aggregate are correlated to GDP
# The correlation is higher betweeen aggregated expense and GDP

#Combine the data across top 5 states and plot the relationship

MyMerge <- function(x, y){
  df <- merge(x, y, by=c("state","year"), all = TRUE)
  return(df)
}
```

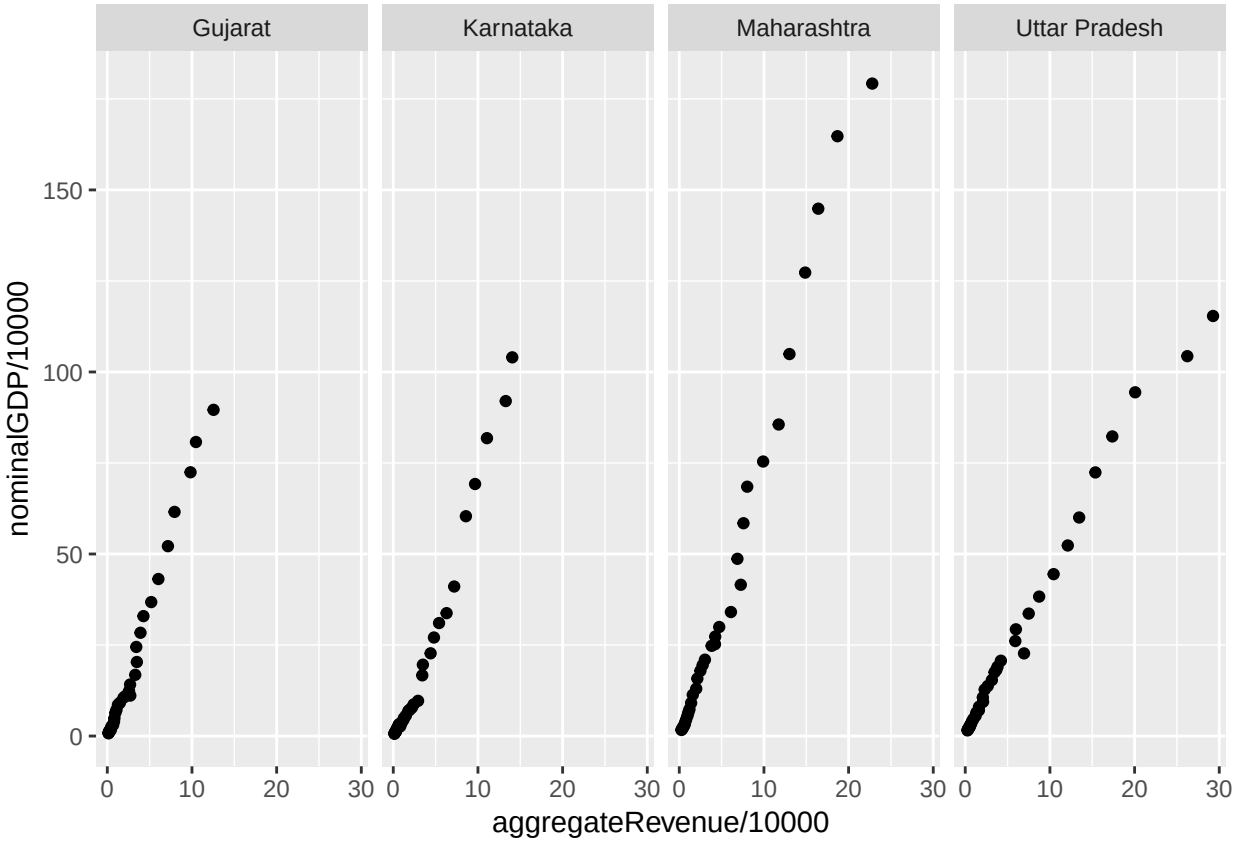
```
mergedData <- Reduce(MyMerge,list(nominalGDP_m,fiscalDeficit_m,aggregate_m))
mergedData <- subset(mergedData,select = -c(3,5,7))
# Drop the categorical columns as they dont have any significance
colnames(mergedData) = c("state","year","nominalGDP","fiscalDeficit","aggregateRevenue")
mergedDate <- delete.na(mergedData)

#Filter this for the top 5 states
mergedData <- mergedData %>% filter(state %in% c("Maharashtra","Karnataka","Uttar Pradesh","Gujarat"))

#plot gdp against fiscal deficit and aggregate revenue
ggplot(mergedData,aes(x=fiscalDeficit/10000,y=nominalGDP/10000)) + geom_point() + facet_grid(~mergedDat
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

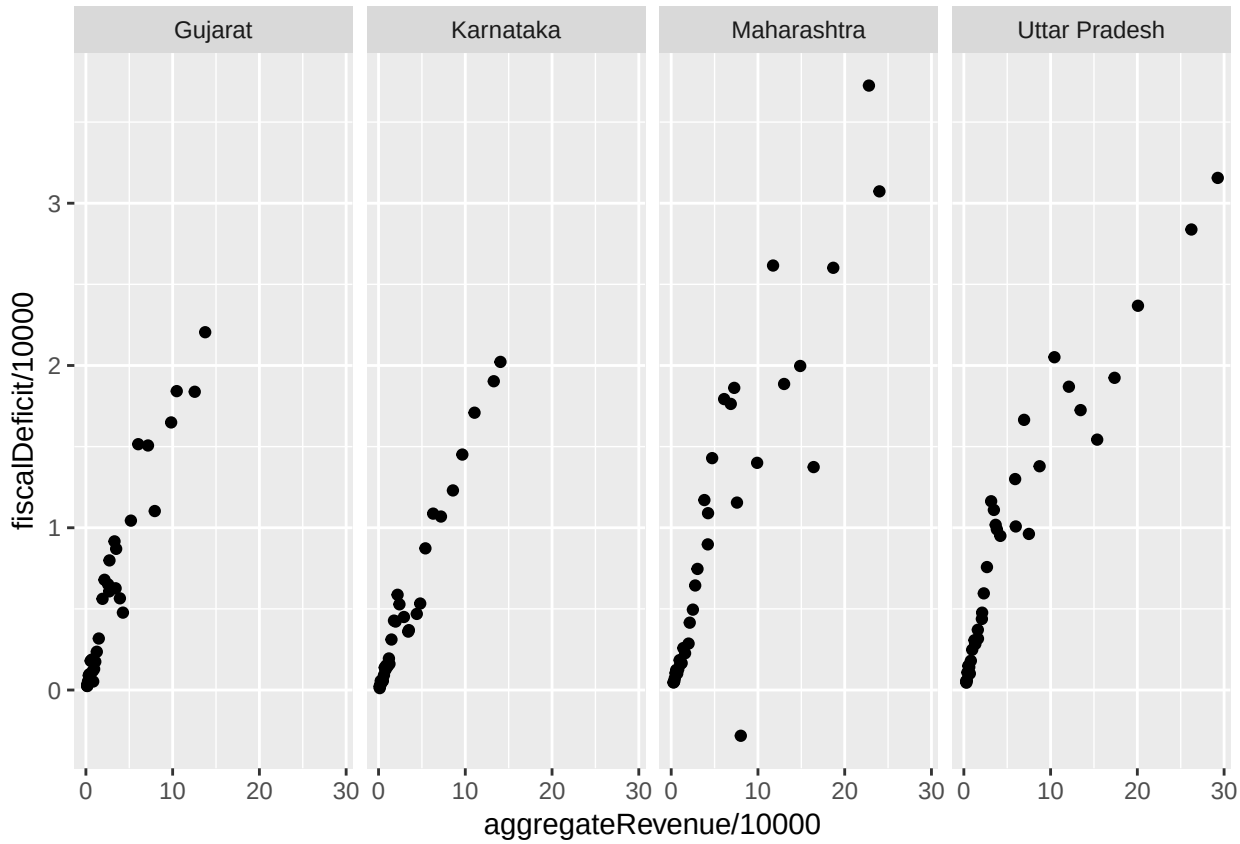



```
ggplot(mergedData,aes(x=aggregateRevenue/10000,y=nominalGDP/10000)) + geom_point() + facet_grid(~mergedI
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

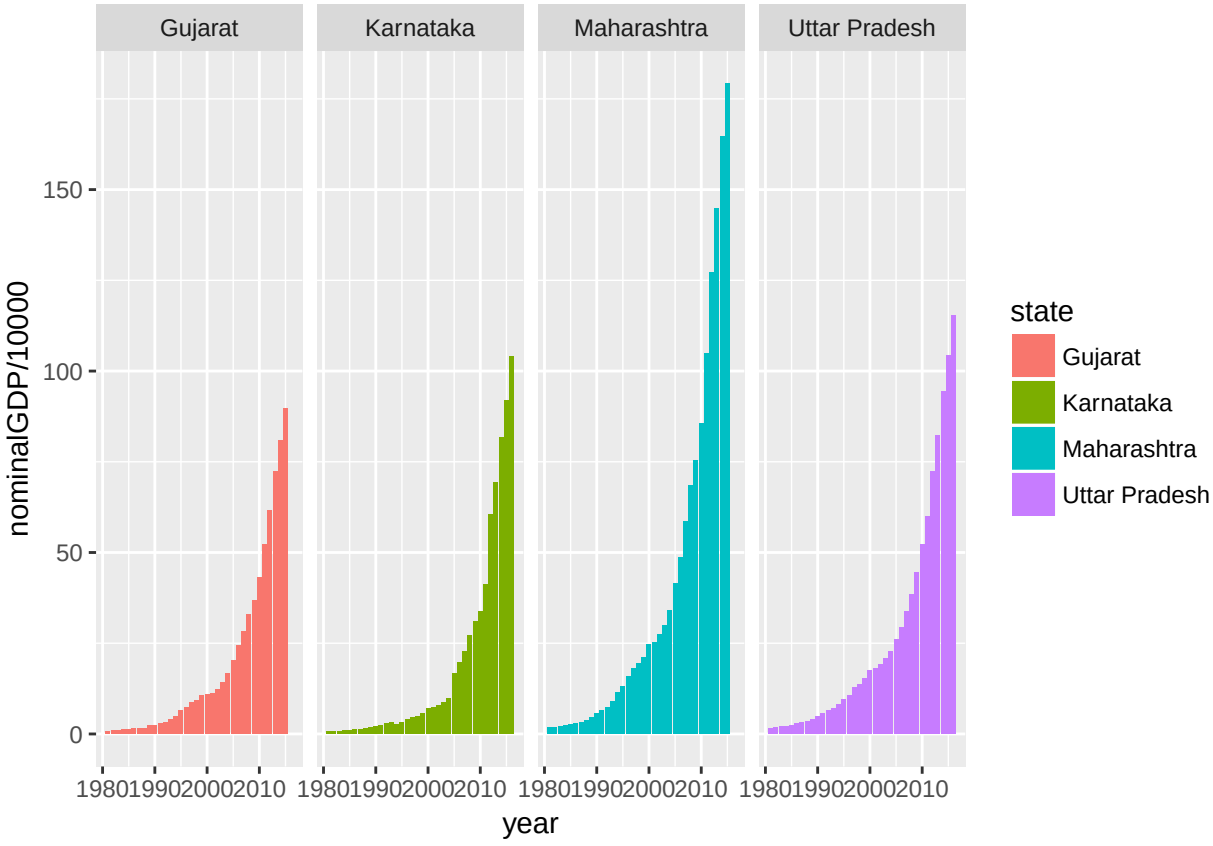



```
ggplot(mergedData,aes(x=aggregateRevenue/10000,y=fiscalDeficit/10000)) + geom_point() + facet_grid(~merg
```




```
# From the above plot  fiscaldeficit grows as the GDP grows. At the same time when aggregate expenditur
# Fiscal deficit also grows along with average expenditure

# Make the year portion as date
mergedData$yearAsDate <- as.Date(paste("31","12",substr(mergedData$year,1,nchar(as.character(mergedData$
mergedData <- subset(mergedData,select = -c(2))
mergedData <- subset(mergedData,select = c("state","yearAsDate","nominalGDP","fiscalDeficit","aggregateR
names(mergedData)[names(mergedData) == 'yearAsDate'] <- 'year'
#Plot the GDP growth in a bar graph
ggplot() + geom_col(data=mergedData,aes(x=year,y=nominalGDP/10000,fill=state)) + facet_grid(~mergedData$
```

```
## Warning: Removed 2 rows containing missing values (position_stack).
```



prophet based prediction for top most state

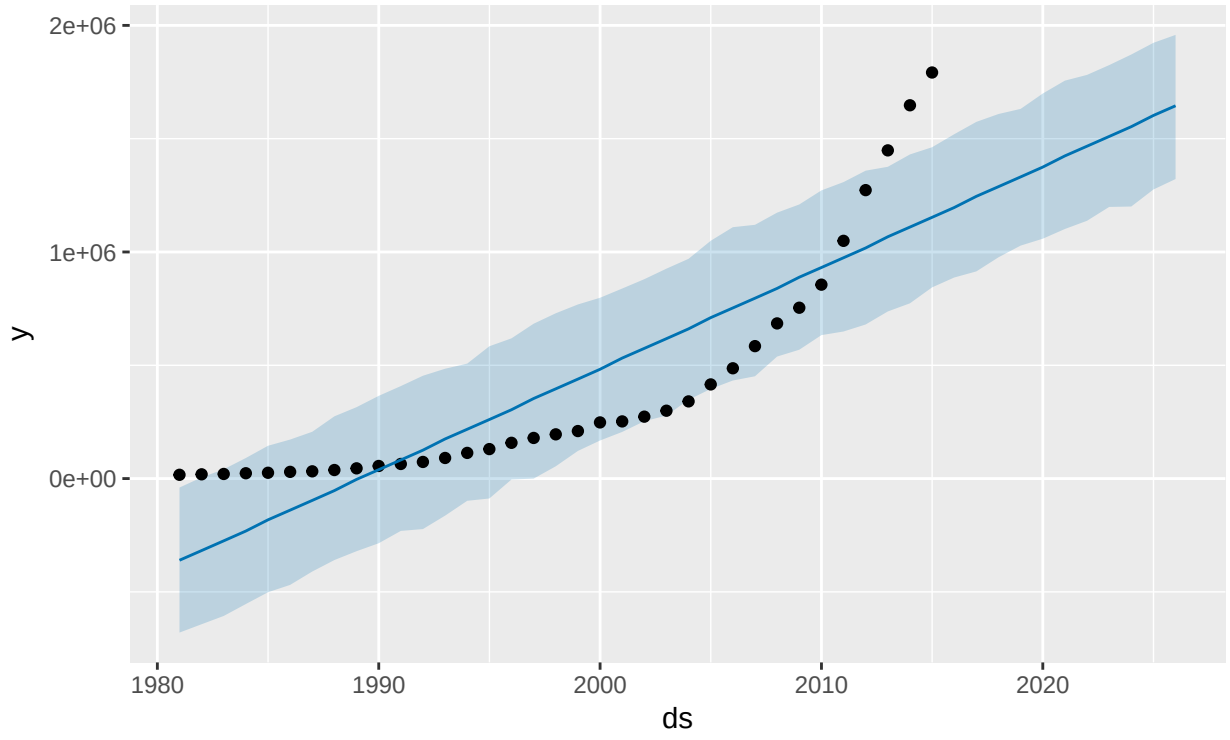
```
gdpMH <- mergedData %>% filter(state == "Maharashtra") %>% select("year","nominalGDP")
colnames(gdpMH) <- c('ds','y')
#prophet needs date in a specific format
gdpMH$ds <- as.Date(gdpMH$ds,"%Y-%m-%d")
m=prophet(gdpMH)
```

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

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

```
## Initial log joint probability = -3.83318
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
```

```
future = make_future_dataframe(m,period=10,freq = "year")
forecast = predict(m,future)
plot(m,forecast)
```



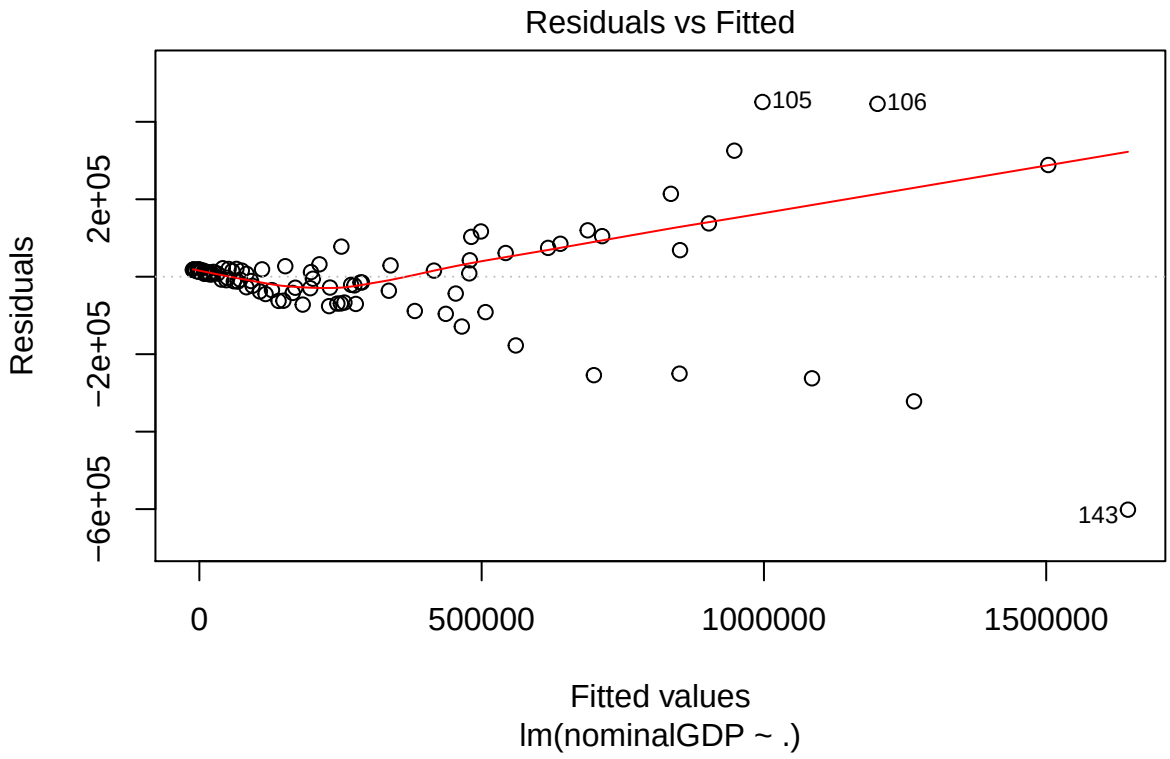
Apply linear regression on the data set

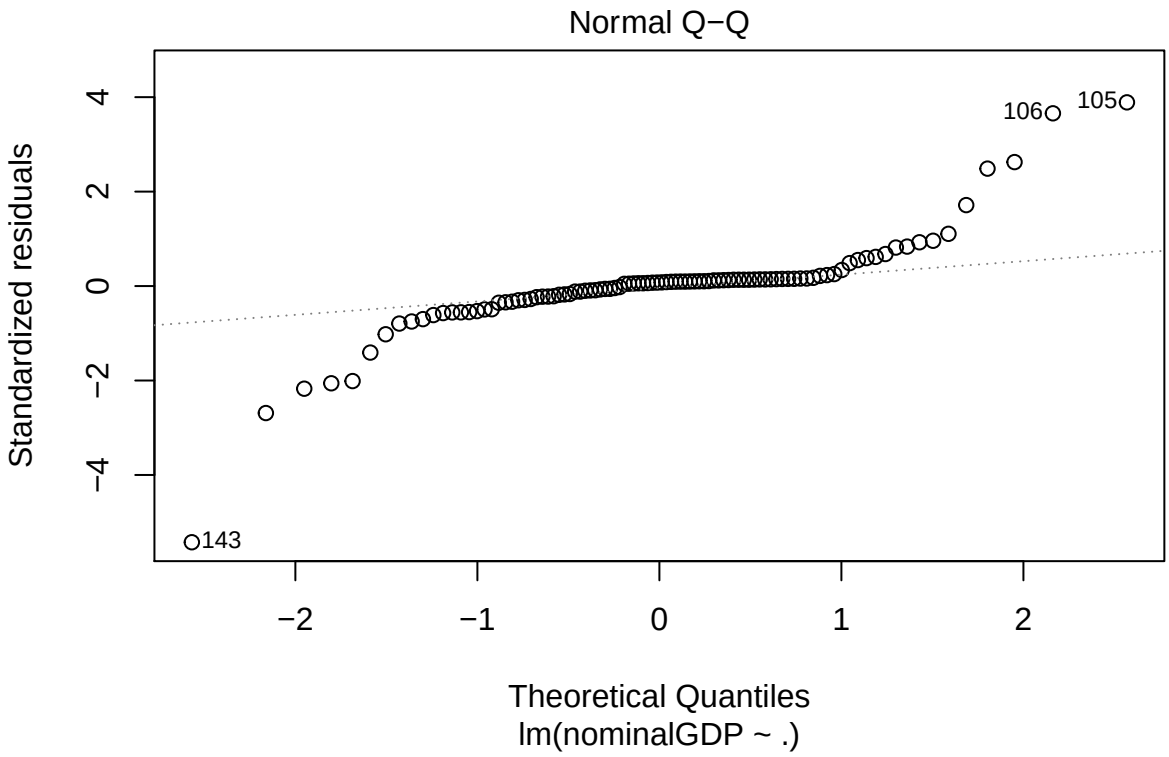
```
set.seed(200)
#Make a copy of the mergedData
mergedDataLM <- mergedData

#drop the state and year column
mergedDataLM <- subset(mergedDataLM,select = -c(1,2))
#split the data for train and test
split <- sample.split(mergedDataLM$nominalGDP, SplitRatio = 0.7)
train <- subset(mergedDataLM, split == TRUE)
test <- subset(mergedDataLM, split == FALSE)

# 1st Model
model <- lm(nominalGDP ~ ., data = train)
summary(model)
```

```
##
## Call:
## lm(formula = nominalGDP ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -601396  -29357    9544   18901  451099
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2.033e+04  1.845e+04  -1.102    0.273
## fiscalDeficit     6.125e+00  4.497e+00   1.362    0.176
## aggregateRevenue  5.685e+00  6.292e-01   9.036  1.9e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127000 on 95 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:   0.89,  Adjusted R-squared:  0.8877
## F-statistic: 384.4 on 2 and 95 DF,  p-value: < 2.2e-16
```

```
# Aggregate revenue is the most significant variable for nominal GDP as it has the highest t value and
#

plot(model,which = c(1,2))
```





Conclusions

1. Maharashtra is the highest GDP growth state. Maharashtra's contribution was 15% of the overall GDP in 2014-15. This was almost twice that of the next best.
2. Maharashtra is followed by Gujarat, Karnataka and UP
3. Nominal GDP is correlated to both fiscal deficit and aggregate expense in a positive way i.e. GDP grows as and when fiscal deficit and aggregate expense grows. This implies that fiscal deficit and state expense are good for the economy.
4. GDP growth has shown a significant upward trend from early 2000 onwards.