FDA_Project

Dharaneesh A, Vishwanath A L, Shravani Swaroop Urala

09/09/2021

#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
```

the Datasets

```
waterData = read.csv("/Users/shravaniurala/Downloads/water_dataX.csv")
str(waterData)
```

```
## 'data.frame':    1991 obs. of  12 variables:
##  $ STATION.CODE                    : chr  "1393" "1399" "1475" "3181" ...
##  $ LOCATIONS                       : chr  "DAMANGANGA AT D/S OF MADHUBAN, DAMAN" "ZUARI AT D/S OF PT
##  $ STATE                           : chr  "DAMAN & DIU" "GOA" "GOA" "GOA" ...
##  $ Temp                            : num  30.6 29.8 29.5 29.7 29.5 30 29.2 29.6 30 30.1 ...
##  $ D.O...mg.l.                     : num  6.7 5.7 6.3 5.8 5.8 5.5 6.1 6.4 6.4 6.3 ...
##  $ PH                              : num  7.5 7.2 6.9 6.9 7.3 7.4 6.7 6.7 7.6 7.6 ...
##  $ CONDUCTIVITY..µmhos.cm.         : chr  "203" "189" "179" "64" ...
##  $ B.O.D...mg.l.                   : chr  "NAN" "2" "1.7" "3.8" ...
##  $ NITRATENAN.N..NITRITENANN..mg.l.: num  0.1 0.2 0.1 0.5 0.4 0.1 0.3 0.2 0.1 0.1 ...
##  $ FECAL.COLIFORM..MPN.100ml.      : chr  "11" "4953" "3243" "5382" ...
##  $ TOTAL.COLIFORM..MPN.100ml.Mean  : chr  "27" "8391" "5330" "8443" ...
##  $ year                            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014 ...
```

Cleaning Water data

```
waterData = waterData %>%
  rename(DO=D.O...mg.l.,pH=PH,Conductivity=CONDUCTIVITY..µmhos.cm., BOD=B.O.D...mg.l., Nitrates=NITRATEN
waterData$Conductivity = as.numeric(waterData$Conductivity)
waterData$BOD = as.numeric(waterData$BOD)
waterData$FecalColiform = as.numeric(waterData$FecalColiform)
waterData$TotalColiform = as.numeric(waterData$TotalColiform)
sum(is.na(waterData$DO))
```

```
## [1] 31
```

```
sum(is.na(waterData$Temp))
```

```
## [1] 92
```

```
sum(is.na(waterData$pH))
```

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

```
sum(is.na(waterData$BOD))
```

```
## [1] 43
```

```
sum(is.na(waterData$Conductivity))
```

```
## [1] 25
```

```
sum(is.na(waterData$FecalColiform))
```

```
## [1] 316
```

```
sum(is.na(waterData$TotalColiform))
```

```
## [1] 132
```

```
waterData$DO = replace(waterData$DO, is.na(waterData$DO), mean(waterData$DO))
waterData$pH = replace(waterData$pH, is.na(waterData$pH), mean(waterData$pH))
waterData$Temp = replace(waterData$Temp, is.na(waterData$Temp), mean(waterData$Temp))
waterData$BOD = replace(waterData$BOD, is.na(waterData$BOD), mean(waterData$BOD))
waterData$Conductivity = replace(waterData$Conductivity, is.na(waterData$Conductivity), mean(waterData$C
waterData$FecalColiform = replace(waterData$FecalColiform, is.na(waterData$FecalColiform), mean(waterDa
waterData$TotalColiform = replace(waterData$TotalColiform, is.na(waterData$TotalColiform), mean(waterDa
max(waterData$pH)
```

```
## [1] NaN
```

```
waterData$pH = replace(waterData$pH, waterData$pH>88, 88.13)
```

Loading the crop dataset

```
cropData = read.csv("/Users/shravaniurala/Downloads/crop_production.csv")
str(cropData)
```

```
## 'data.frame':    246091 obs. of  7 variables:
##  $ State_Name   : chr  "Andaman and Nicobar Islands" "Andaman and Nicobar Islands" "Andaman and Nicob
##  $ District_Name: chr  "NICOBARS" "NICOBARS" "NICOBARS" "NICOBARS" ...
##  $ Crop_Year    : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ Season       : chr  "Kharif     " "Kharif     " "Kharif     " "Whole Year " ...
##  $ Crop         : chr  "Arecanut" "Other Kharif pulses" "Rice" "Banana" ...
##  $ Area         : num  1254 2 102 176 720 ...
##  $ Production   : num  2000 1 321 641 165 65100000 100 2 15 169 ...
```

Basic analysis of Area and Production relation

```
summary(lm(Production~Area, cropData))
```

```
##
## Call:
```

```
## lm(formula = Production ~ Area, data = cropData)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -109718077    -462316    -421847    -417266 1249256816
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.168e+05  3.561e+04    11.7   <2e-16 ***
## Area        1.362e+01  6.811e-01    20.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17050000 on 242359 degrees of freedom
##   (3730 observations deleted due to missingness)
## Multiple R-squared:  0.001647,   Adjusted R-squared:  0.001643
## F-statistic: 399.9 on 1 and 242359 DF,  p-value: < 2.2e-16
```

It can be seen that area has a high relation with the crop prduction, like it usually has

DIfferent states information present in water dataset

```
waterStates = unique(waterData$STATE)
waterStates
```

```
##   [1] "DAMAN & DIU"
##   [2] "GOA"
##   [3] "MAHARASHTRA"
##   [4] "KERALA"
##   [5] "ANDHRA PRADESH"
##   [6] "KARNATAKA"
##   [7] "ODISHA"
##   [8] "PONDICHERRY"
##   [9] "TAMILNADU"
##  [10] "HARYANA"
##  [11] "PUNJAB"
##  [12] "RAJASTHAN"
##  [13] "HIMACHAL PRADESH"
##  [14] "MEGHALAYA"
##  [15] "MIZORAM"
##  [16] "TRIPURA"
##  [17] "GUJARAT"
##  [18] "NAN"
##  [19] "ORISSA"
##  [20] "MADHYA PRADESH"
##  [21] "MANIPUR"
##  [22] "DAMAN, DIU, DADRA NAGAR HAVELI"
##  [23] "TAMIL NADU"
##  [24] "tripura"
##  [25] "DAMANGANGA AT KACHIGAON U\\S AT GIDC WIER,GUJARAT"
##  [26] "DAMANGANGA AT KACHIGAON D\\S (DAMAN),GUJARAT"
##  [27] "DAMANGANGA AT ZARI CAUSE WAY BRIDGE,DAMAN"
##  [28] "DAMANGANGA DISCHARGE POINT OF DISTILLERY, DAMAN"
```

```
##  [29] "DAMANGANGA AT DAMAN JETTY, MOTI DAMAN"
##  [30] "DAMANGANGA VAPI WEIR,VAPI, DAMAN"
##  [31] "DAMANGANGA AT LAVACHA TEMPLE, SILVASSA"
##  [32] "DAMANGANGA D/S OF M/S SURAT BEVERAGES, VILLAGE DADRA, SILVASSA"
##  [33] "DAMANGANGA AT NAROLI BRIDGE, SILVASSA"
##  [34] "DAMANGANGA AT VILLAGE NAMDHA, VAPI"
##  [35] "DAMANGANGA AT D/S OF MADHUBAN, DAMAN"
##  [36] "BALEHWAR KHADI AT N.H. NO. 8"
##  [37] "RIVER PURNA ON BRIDGE AT SURATNANNAVSARI HIGHWAY"
##  [38] "RIVER KAVERI ON BRIDGE AT BILLIMORANANVALSAD ROAD"
##  [39] "RIVER DHADAR AT KOTHADA"
##  [40] "AMBIKA AT BILIMORA, GUJARAT"
##  [41] "AMLAKHADI AFTER CONFL. OF W. WATER FROM ANKLESH,GUJARAT"
##  [42] "BHADAR D/S JETPUR VILL. AFTER CONF. OF W.WATER FROM JETPUR CITY,GUJARAT"
##  [43] "MINDHOLA AT STATE HIGHWAY BRIDGE SACHIN, GUJARAT"
##  [44] "RIVER BHOGAVO D/S OF SURENDRANAGAR."
##  [45] "TRIVENI SANGAM, NR. SOMNATH TEMPLE, VERAVAL, DIST. JUNAGADH."
##  [46] "ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOINS, GOA"
##  [47] "MANDOVI AT NEGHBOURHOOD OF PANAJI, GOA"
##  [48] "ZUARI AT PANCHAWADI"
##  [49] "MANDOVI AT TONCA, MARCELA, GOA"
##  [50] "RIVER KALNA AT CHANDELNAN PERNEM, GOA"
##  [51] "RIVER VALVANT AT SANKLI NAN BICHOLIM, GOA"
##  [52] "RIVER MADAI AT DABOS NAN  VALPOI, GOA"
##  [53] "RIVER KHANDEPAR AT OPA NAN PONDA, GOA"
##  [54] "RIVER TALPONA AT CANACONA, GOA"
##  [55] "RIVER ASSONORA AT ASSONORA, GOA"
##  [56] "RIVER KHANDEPAR AT CODLI NEAR BRIDGE ,U/S OPA WATERWORKS,SANGUEM"
##  [57] "RIVER SAL PAZORKHONI,CUNCOLIM(NEAR CULVERT MARGAONAN CANACONA NATIONAL HIGHWAY)"
##  [58] "RIVER KUSHAWATI NEAR BUND AT KEVONA,RIVON,SANGUEM"
##  [59] "RIVER SAL NEAR HOTEL LEELA MOBOR,CAVELOSSIM"
##  [60] "RIVER MAPUSA ON CULVERT ON HIGHWAY MAPUSANANPANAJI"
##  [61] "RIVER CHAPORA NEAR ALORNA FORT ,PERNEM"
##  [62] "RIVER BICHOLIM VARAZAN NAGAR , BICHOLIM"
##  [63] "KALU AT ATALE VILLAGE, MAHARASHTRA"
##  [64] "ULHAS AT U/S OF NRC BUND AT MOHANE, MAHARASHTRA"
##  [65] "ULHAS AT U/S OF BADLAPUR, MAHARASHTRA"
##  [66] "BHATSA AT D/S OF PISE DAM NEAR PISE VILLAGE (ULHAS), MAHARASHTRA"
##  [67] "PATALGANGA NEAR INTAKE OF MIDC W/W, MAHARASHTRA"
##  [68] "ULHAS RIVER ATJAMBHUL WATER WORKS."
##  [69] "VASHISTI RIVER AT U/S OF THREE M PAPER MILLS NEAR M/S MULTIFILMS PLASTIC PVT. LTD. AT KHERDI."
##  [70] "MITHI RIVER"
##  [71] "KUNDALIKA RIVER AT ARE KHURD ( SALINE ZONE )"
##  [72] "SAVITRI RIVER AT OVALE VILLAGE."
##  [73] "PATALGANGA AT SHILPHATA, MAHARASHTRA"
##  [74] "KUNDALIKA AT ROHA CITY, MAHARASHTRA"
##  [75] "PERIYAR NEAR ALWAYENANELOOR, KERALA"
##  [76] "PERIYAR AT KALADY, KERALA"
##  [77] "CHALIYAR AT KOOLIMADU, KERALA"
##  [78] "CHALIYAR AT CHUNGAPALLY, KERALA"
##  [79] "KALLADA AT PERUMTHOTTAMKADAVU, PUNALOOR, KERALA"
##  [80] "MUVATTAPUZHA AT VETTIKKATTUMUKKU, KERALA"
##  [81] "CHALAKUDY AT PULICKALKANANDAVU, KERALA"
##  [82] "KARAMANA AT MOONNATTUMUKKU, KERALA"
```

```
##  [83] "PAMBA AT CHENGANNUR, KERALA"
##  [84] "PERIYAR AT SEWAGE DISCHARGE POINT, KERALA"
##  [85] "MEENACHIL AT KIDANGOOR, KERALA"
##  [86] "MANIMALA  AT KALLOOPARA, KERALA"
##  [87] "PAMBA AT THAKAZHY, KERALA"
##  [88] "ACHENKOIL AT THUMPAMON, KERALA"
##  [89] "MANIMALA AT THONDRA, KERALA"
##  [90] "VAMANAPURAM, KERALA"
##  [91] "ACHENKOIL AT CHENNITHULA, KERALA"
##  [92] "NEYYAR AMARAVILA, KERALA"
##  [93] "AYUR, KERALA"
##  [94] "PAMBA DOWN, KERALA"
##  [95] "THIRURANGADY, KERALA"
##  [96] "KUTTIYADY ESTATE, KERALA"
##  [97] "MAHI VALAYAM, KERALA"
##  [98] "KUPPAM THALIPARAMBA, KERALA"
##  [99] "NEELASHWER HOSDURG, KERALA"
## [100] "KORINGODA KAKKADAVU,  KERALA"
## [101] "CHANDRIGIRI PADIYATHADKA, KERALA"
## [102] "CHAITRAPUZHA IRUMPANAM, KERALA"
## [103] "R NEYYAR AT ARUVIPURAM"
## [104] "R MAMOM AT MAMOM BRIDGE"
## [105] "R AYROOR AT AYROOR BRIDGE"
## [106] "R ITHIKKARA AT ITHIKKARA"
## [107] "R PALLICKAL AT NELLIMUKAL"
## [108] "R KARUVANNURR AT KARUVANNUR BRIDGE"
## [109] "R PUZHACKAL AT PUZHACKAL BRIDGE"
## [110] "R KEECHERI AT VADAKKANCHERY BRIDGE"
## [111] "R THIRUR AT THALAKKADATHUR BRIDGE"
## [112] "R KADALUNDI AT HAJIRAPPALLY"
## [113] "R KALLAI AT KALLAI BRIDGE"
## [114] "R CORAPUZHA AT KANAYANKODE"
## [115] "R THALLASSERY AT PATHIPPALAM"
## [116] "R ANCHARAKANDY AT MERUVAMBA"
## [117] "R KUPPAM AT RAYAROM"
## [118] "R RAMAPURAM AT RAMAPURAM BRIDGE"
## [119] "R PERUVAMBA AT CHANDAPPURA"
## [120] "R KAVVAI AT KUTTIYOL PALAM"
## [121] "R NEELASWARAM AT NAMBIARKAL DAM"
## [122] "R PULLUR AT PULLUR BR."
## [123] "R MOGRAL AT MOGRAL BR."
## [124] "R SHRIYA AT ANGADIMOGARU"
## [125] "R UPPALA AT  UPPALA BR."
## [126] "R MANJESWAR AT BAJRAKKARA BR."
## [127] "R KORAYAR AT KANJIKODE"
## [128] "R BHARATHAPUZHA AT KUTTIPPURAM"
## [129] "R BHARATHAPUZHA AT PATTAMBI"
## [130] "RIVER PERIYAR AT MUPPATHADAM"
## [131] "RIVER PERIYAR AT PATHALAM"
## [132] "R PERIYAR AT KALAMASSERY"
## [133] "R PERIYAR AT PURAPPALLIKAVU"
## [134] "R KADAMBAYAR AT BRAHMAPURAM"
## [135] "R KADAMBAYAR AT MANCKAKADAVU"
## [136] "NAGAVALLI AT THOTAPALLI REGULATOR, A.P."
```

```
## [137] "RUSHIKULYA AT GANJAM U/S, ORISSA"
## [138] "RUSHIKULYA AT GANJAM D/S, ORISSA"
## [139] "NAGAVALLI AT JAYKAYPUR D/S, ORISSA"
## [140] "NAGAVALLI AT RAYAGADA D/S, ORISSA"
## [141] "NAGAVALI PENTA U/S, JAYKAYPUR TOWN"
## [142] "BUDHABALANGA, D/S OF BARIPADA TOWN"
## [143] "BUDHABALANGA AT BALASORE D/S"
## [144] "RIVER KERANDI (INTAKE WELL OF NALCO REFINARY, HAL, SUNABEDA)"
## [145] "Vansdhara MUNIGUDA (D/S OF M/S VEDANTAALUMINA PROJECT)"
## [146] "Vansdhara GUNUPUR (INTERSTATE BOUNDRY)"
## [147] "GAUTAMINANGODAVARI RIVER"
## [148] "CORINGA RIVER"
## [149] "MAHE RIVER"
## [150] "ARASALAR RIVER KARAIKAL REGION, PONDICHERRY"
## [151] "TAMBIRAPARANI AT BDG.NR. MADURA COATS LTD.PAPAVINASAM,TAMILNADU"
## [152] "TAMBIRAPARANI AT CHERANMADEVI,CAUSE WAY,TAMILNADU"
## [153] "TAMBIRAPARANI AT TIRUNELVELI,COLLECTORATE,TAMILNADU."
## [154] "TAMBIRAPARANI AT MURAPPANADU, TAMILNADU"
## [155] "TAMBIRAPARANI AT PAPPANKULAM,TAMILNADU"
## [156] "TAMBIRAPARANI AT RAIL BDG. NR. AMBASAMUDAM, TAMILNADU"
## [157] "TAMBIRAPARANI AT ARUMUGANERI, TAMILNADU"
## [158] "PALAR AT VANIYAMBADI WATER SUPPLY HEAD WORK, TAMILNADU"
## [159] "KALI AT D/S WEST COAST PAPER MILL, KARNATAKA"
## [160] "NETRAVATHI U/S OF DHARMASTALA AT WATER SUPPLY INTAKE POINT"
## [161] "KUMARADHARA NAN U/S OF UPPINAGADY TOWN BEFORE CONFLUENCE WITH RIVER NETHRAVATHI"
## [162] "RIVER VAMSHADHARA, KALINGAPATNAM, VIZIANAGARAM"
## [163] "GHAGGAR AT MUBARAKPUR REST HOUSE (PATIALA), PUNJAB"
## [164] "GHAGGAR AT 100M D/S CONF. WITH R. SARASWATI (PATIALA),PUNJAB"
## [165] "GHAGGAR GHNAN1 AT ROAD BRDG. SIRSA,DEBWALI ROAD,HARYANA"
## [166] "GHAGGAR GHNAN2 AT CHANDARPUR SYPHON,  HARYANA"
## [167] "GHAGGAR NEAR BANKARPUR,DERA BASSI,PUNJAB"
## [168] "GHAGGAR AT RATANHERI, D/S OF PATIALA NADI (AFTER CONFL.), PUNJAB"
## [169] "GHAGGAR AT D/S CHHATBIR, PUNJAB"
## [170] "GHAGGAR AT U/S DHAKANSU NALLAH, PUNJAB"
## [171] "D/S DHAKANSU NALLAH, PUNJAB"
## [172] "GHAGGAR AT D/S JHARMAL NADI, PUNJAB"
## [173] "GHAGGAR AT U/S JHARMAL NADI, PUNJAB"
## [174] "GHAGGER AT MOONAK, PUNJAB"
## [175] "GHAGGAR AT D/S SARDULGARH, PUNJAB"
## [176] "GHAGGAR AT U/S SARDULGARH, PUNJAB"
## [177] "GHAGGAR AT KALA AMB D/S MARKANDA RIVER"
## [178] " RIVER GHAGGAR AT D/S OF SURAJPUR"
## [179] "GHAGGAR BEFORE OTTU WEIR (BEFORE MIXING OF SATLUJ CANAL WATER) (HARYANA)"
## [180] "RIVER SUKHANA AT PARWANOO, DISTT. SOLAN, H.P."
## [181] "KODRA DAM, MOUNT ABU, RAJASTHAN"
## [182] "IMPHAL AT MAHABALI, MANIPUR"
## [183] "IMPHAL AT KOIRENGEI, MANIPUR"
## [184] "IRIL AT PORAMPET, MANIPUR"
## [185] "IRIL LILONG, MANIPUR"
## [186] "NAMBUL HUMP BRIDGE, MANIPUR"
## [187] "NAMBUL HEIRANGOITHONG, MANIPUR"
## [188] "KIYAMGI, Imphal, MANIPUR"
## [189] "MINUTHONG IMPHAL RIVER, MANIPUR"
## [190] "KHUGA RIVER (CHURACHANDPUR DIST.)"
```

```
## [191] "KHUJAIROK RIVER, MOREH (CHANDEL DIST.)"
## [192] "UMTREW AT BYRNIHAT EAST, MEGHALAYA"
## [193] "KHARKHLA NEAR SUTNGA KHLIERIAT,JAINTIA HILLS DT.,MEGHALAYA"
## [194] "TLAWNG UPSTREAM AIZAWL"
## [195] "TLAWNG DOWNSTREAM AIZAWL"
## [196] "TUIRIAL UPPER CATCHMENT"
## [197] "TUIRIAL LOWER CATCHMENT"
## [198] "GUMTI AT U/S SOUTH TRIPURA,TRIPURA"
## [199] "GUMTI AT D/S SOUTH TRIPURA, TRIPURA"
## [200] "CHANDRAPUR, AGARTALA D/S OF HAORA RIVER, TRIPURA"
## [201] "MYNTDU JAINTIA HILLS MEGHALAYA"
## [202] "SIMSANG, WILLIAMNAGAR, MEGHALAYA"
## [203] "GANOL, TURA, MEGHALAYA"
```

Data cleaning part

```
cropData$State_Name = sub("Dadra and Nagar Haveli","Daman and Diu",cropData$State_Name)
cropStates = unique(cropData$State_Name)
print(cropStates)
```

```
##  [1] "Andaman and Nicobar Islands" "Andhra Pradesh"
##  [3] "Arunachal Pradesh"           "Assam"
##  [5] "Bihar"                       "Chandigarh"
##  [7] "Chhattisgarh"                "Daman and Diu"
##  [9] "Goa"                         "Gujarat"
## [11] "Haryana"                     "Himachal Pradesh"
## [13] "Jammu and Kashmir "          "Jharkhand"
## [15] "Karnataka"                   "Kerala"
## [17] "Madhya Pradesh"              "Maharashtra"
## [19] "Manipur"                     "Meghalaya"
## [21] "Mizoram"                     "Nagaland"
## [23] "Odisha"                      "Puducherry"
## [25] "Punjab"                      "Rajasthan"
## [27] "Sikkim"                      "Tamil Nadu"
## [29] "Telangana "                  "Tripura"
## [31] "Uttar Pradesh"               "Uttarakhand"
## [33] "West Bengal"
```

Analysing the common years information, which can be used for combining

```
waterYears = unique(waterData$year)
cropYears = unique(cropData$Crop_Year)
print(waterYears)
```

```
##  [1] 2014 2013 2012 2011 2010 2009 2008 2007 2006 2005 2004 2003
```

```
print(cropYears)
```

```
##  [1] 2000 2001 2002 2003 2004 2005 2006 2010 1997 1998 1999 2007 2008 2009 2011
## [16] 2012 2013 2014 2015
```

```
cropData = cropData %>%
  filter(Crop_Year %in% 2003:2014)
cropYears = unique(cropData$Crop_Year)
print(cropYears)
```

```
##  [1] 2003 2004 2005 2006 2010 2007 2008 2009 2011 2012 2013 2014
```

Checking for the missing data

```
sum(is.na(waterData))
```

```
## [1] 872
```

```
nrow(waterData)
```

```
## [1] 1991
```

```
water_data = na.omit(waterData)
nrow(water_data)
```

```
## [1] 1577
```

```
sum(is.nan(water_data$FECAL.COLIFORM..MPN.100ml.))
```

```
## [1] 0
```

Data cleaning

```
#unique(water_data$STATE)
k=c('Telangana','Andhra Pradesh',"AP",'A.P.','Arunachal Pradesh','Assam','Bihar','Chhattisgarh','Goa',
'Chandigarh','Dadra and Nagar Haveli','Daman and Diu','Daman','Lakshadweep','Puducherry','PONDICHERRY',
length(k)
```

```
## [1] 43
```

```
k <- toupper(k)
pattern = paste(paste(".*\\b(", paste(k, collapse="|")), ")\\b.*")
pattern
```

```
## [1] ".*\\b( TELANGANA|ANDHRA PRADESH|AP|A.P.|ARUNACHAL PRADESH|ASSAM|BIHAR|CHHATTISGARH|GOA|GUJARAT|
```

```
water_data$states =  sub(pattern, "\\1", water_data$STATE)
#water_data$states =  sub(pattern, "\\1", water_data$STATE)
water_data$states =  ifelse(water_data$states == "NAN", sub(pattern, "\\1", water_data$LOCATIONS), wate
water_data$states = sub('^(A.P.|AP)$','ANDHRA PRADESH',water_data$states)
water_data$states = sub("^H.P.$","HIMACHAL PRADESH",water_data$states)
water_data$states = sub("^TAMILNADU$","TAMIL NADU",water_data$states)
water_data$states = sub("^DAMAN$","DAMAN AND DIU",water_data$states)
water_data$states = sub("^PONDICHERRY$","PUDUCHERRY",water_data$states)
water_data$states = sub("^tripura$","TRIPURA",water_data$states)
water_data$states = sub("^ORISSA$","ODISHA",water_data$states)
#View(water_data)
unique(water_data$states)
```

```
##   [1] "GOA"
##   [2] "MAHARASHTRA"
##   [3] "KERALA"
##   [4] "ANDHRA PRADESH"
##   [5] "KARNATAKA"
##   [6] "ODISHA"
##   [7] "TAMIL NADU"
##   [8] "PUNJAB"
##   [9] "RAJASTHAN"
```

```
##  [10] "HIMACHAL PRADESH"
##  [11] "MEGHALAYA"
##  [12] "MIZORAM"
##  [13] "TRIPURA"
##  [14] "GUJARAT"
##  [15] "MADHYA PRADESH"
##  [16] "DAMAN AND DIU"
##  [17] "ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOINS"
##  [18] "ZUARI AT PANCHAWADI"
##  [19] "MANDOVI AT NEGHBOURHOOD OF PANAJI"
##  [20] "MANDOVI AT TONCA, MARCELA"
##  [21] "KALNA AT CHANDELNAN PERNEM"
##  [22] "ASSONORA AT ASSONORA"
##  [23] "RIVER BICHOLIM VARAZAN NAGAR , BICHOLIM"
##  [24] "RIVER CHAPORA NEAR ALORNA FORT ,PERNEM"
##  [25] "KHANDEPAR AT OPA NAN PONDA"
##  [26] "KHANDEPAR AT CODLI NR BDG,U/S OPA WATERWORKS, SANGUEM"
##  [27] "KUSHAWATI NEAR BUND AT KEVONA,RIVON,SANGUEM"
##  [28] "MADAI AT DABOS NAN  VALPOI"
##  [29] "MAPUSA ON CULVERT ON HIGHWAY MAPUSANANPANAJI"
##  [30] "SAL PAZORKHONI, CUNCOLIM (NR CULVERT MARGAONAN CANACONA N.H.)"
##  [31] "SAL NEAR HOTEL LEELA MOBOR, CAVELOSSIM"
##  [32] "TALPONA AT CANACONA"
##  [33] "VALVANT AT SANKLINANBICHOLIM"
##  [34] "AMBA AT D/S OF WAKEN BDG U/S OF JS PETROCHEMICALS, WAKENPHATA, ROHA"
##  [35] "BHATSA AT D/S OF PISE DAM NEAR PISE VILLAGE (ULHAS)"
##  [36] "KALU AT ATALE VILLAGE"
##  [37] "KAN NEAR SAKRI WATER WORKS, SAKRI, DHULE"
##  [38] "KUNDALIKA AT ROHA CITY"
##  [39] "KUNDALIKA RIVER AT ARE KHURD ( SALINE ZONE )"
##  [40] "KUNDALIKA NEAR SALAV BDG (SALINA ZONE), SALAV, ROHA, RAIGAD"
##  [41] "KUNDALIKA  AT DHATAV AT JACKWELL, DHATAV, ROHA, RAIGAD"
##  [42] "MITHI RIVER"
##  [43] "MUCHKUNDI AT WAKED NEAR M/S ASAHI INDIA GLASS LTD, WAKED, LANJA, RATNAGIRI"
##  [44] "PATALGANGA AT SHILPHATA"
##  [45] "PATALGANGA NEAR INTAKE OF MIDC W/W"
##  [46] "PATALGANGA AT D/S OF KHARPADA BDG, KHARPADA, KHALAPUR, RAIGAD"
##  [47] "PATALGANGA AT VYAL PUMP HOUSE, VYAL, KHALAPUR, RAIGAD"
##  [48] "PATALGANGA AT KHALAPUR PUMPING STATION, RAIGAD"
##  [49] "PATALGANGA AT SAVROLI BDG, KHALAPUR. RAIGAD"
##  [50] "PATALGANGA RIVER AT GAGANGIRI MAHARAJ TEMPLE, KHOPOLI, KHALAPUR, RAIGAD"
##  [51] "PEHLAR DAM ON PEHLAR RIVERNANINLET OF WATER WORKS, VASAI, THANE"
##  [52] "SAVITRI AT OVALE VILLAGE."
##  [53] "SAVITRI JACKWELL AT UPSA KENDRE, NANGALWADI, MAHAD, RAIGAD"
##  [54] "SAVITRI RIVER AT SHEDAV DOH, MAHAD,RAIGAD"
##  [55] "SAVITRI AT DADLI BRIDGE, DADLI, MAHAD, RAIGADH"
##  [56] "SAVITRI AT MUTHAVALI VILLAGE, MAHAD, RAIGADH"
##  [57] "SURYA U/S OF SURYA DAM, DHAMMI, VIKRAMGAD, THANE"
##  [58] "SURYA AT MIDC PUMPING STATION, GARVASHET, PALGHAR, THANE"
##  [59] "SURYA RIVER AT INTAKE OF VASAINANVIRAR W/S SCHEME, MASVAN, PALGHAR, THANE"
##  [60] "ULHAS AT U/S OF NRC BUND AT MOHANE"
##  [61] "ULHAS AT U/S OF BADLAPUR"
##  [62] "ULHAS RIVER ATJAMBHUL WATER WORKS."
##  [63] "VASHISTI AT U/S OF THREE M PAPER MILLS NEAR M/S MULTIFILMS PLASTIC PVT. LTD. AT KHERDI."
```

```
##  [64] "VASHISHTI AT D/S OF THREE M PAPER MILLS NR CHIPLUN WATER INTAKE JACKWELL, KHERDI, RATNAGIRI"
##  [65] "VASHISHTI AT U/S OF POPHALI NEAR KONPHANSAWANE BRIDGE, CHIPLUN, RATNAGIRI"
##  [66] "PERIYAR NEAR ALWAYENANELOOR"
##  [67] "PERIYAR AT KALADY"
##  [68] "CHALIYAR AT KOOLIMADU"
##  [69] "CHALIYAR AT CHUNGAPALLY"
##  [70] "KALLADA AT PERUMTHOTTA MKADAVU, PUNALOOR"
##  [71] "MUVATTAPUZHA AT VETTIKKATTUMUKKU"
##  [72] "CHALAKUDY AT PULICKALKANANDAVU"
##  [73] "KARAMANA AT MOONNATTUMUKKU"
##  [74] "PAMBA AT CHENGANNUR"
##  [75] "KABBANI AT MUTHANKARA"
##  [76] "BHAVANI AT ELACHIVAZHY"
##  [77] "PERIYAR AT SEWAGE DISCHARGE POINT"
##  [78] "MEENACHIL AT KIDANGOOR"
##  [79] "MANIMALA  AT KALLOOPARA"
##  [80] "PAMBA AT THAKAZHY"
##  [81] "ACHENKOIL AT THUMPAMON"
##  [82] "MANIMALA AT THONDRA"
##  [83] "VAMANAPURAM"
##  [84] "ACHENKOIL AT CHENNITHULA"
##  [85] "AMARAVILA"
##  [86] "AYUR"
##  [87] "PAMBA DOWN"
##  [88] "THIRURANGADY"
##  [89] "KUTTIYADY ESTATE"
##  [90] "VALAYAM"
##  [91] "THALIPARAMBA"
##  [92] "HOSDURG"
##  [93] "PADIYATHADKA"
##  [94] "IRUMPANAM"
##  [95] "R NEYYAR AT ARUVIPURAM"
##  [96] "R MAMOM AT MAMOM BRIDGE"
##  [97] "R AYROOR AT AYROOR BRIDGE"
##  [98] "R ITHIKKARA AT ITHIKKARA"
##  [99] "PALLICKAL AT NELLIMUKAL"
## [100] "R KARUVANNURR AT KARUVANNUR BRIDGE"
## [101] "PUZHACKAL AT PUZHACKAL BRIDGE"
## [102] "R KEECHERI AT VADAKKANCHERY BRIDGE"
## [103] "R THIRUR AT THALAKKADATHUR BRIDGE"
## [104] "R KADALUNDI AT HAJIRAPPALLY"
## [105] "R KALLAI AT KALLAI BRIDGE"
## [106] "CORAPUZHA AT KANAYANKODE"
## [107] "R THALLASSERY AT PATHIPPALAM"
## [108] "ANCHARAKANDY AT MERUVAMBA"
## [109] "R KUPPAM AT RAYAROM"
## [110] "RAMAPURAM AT RAMAPURAM BRIDGE"
## [111] "PERUVAMBA AT CHANDAPPURA"
## [112] "R KAVVAI AT KUTTIYOL PALAM"
## [113] "NEELASWARAM AT NAMBIARKAL DAM"
## [114] "NAGAVALLI AT THOTAPALLI REGULATOR"
## [115] "VAMSHADHARA, KALINGAPATNAM, VIZIANAGARAM"
## [116] "KALI AT D/S WEST COAST PAPER MILL"
## [117] "RUSHIKULYA AT GANJAM D/S, "
```

```
## [118] "NAGAVALLI AT JAYKAYPUR D/S, "
## [119] "NAGAVALLI AT RAYAGADA D/S, "
## [120] "PENTA U/S, JAYKAYPUR TOWN"
## [121] "BUDHABALANGA, D/S OF BARIPADA TOWN"
## [122] "BUDHABALANGA AT BALASORE D/S"
## [123] "RIVER KERANDI(INTAKE WELL OF NALCO REFINARY, HAL, SUNABEDA)"
## [124] "MUNIGUDA (D/S OF M/S VEDANTAALUMINA PROJECT)"
## [125] "GUNUPUR (INTERSTATE BOUNDRY)"
## [126] "TAMBIRAPARANI AT BDG.NR. MADURA COATS LTD. PAPAVINASAM"
## [127] "TAMBIRAPARANI AT CHERANMADEVI,CAUSE WAY"
## [128] "TAMBIRAPARANI AT TIRUNELVELI, COLLECTORATE."
## [129] "TAMBIRAPARANI AT MURAPPANADU"
## [130] "TAMBIRAPARANI AT PAPPANKULAM"
## [131] "TAMBIRAPARANI AT RAIL BDG. NR. AMBASAMUDAM"
## [132] "TAMBIRAPARANI AT ARUMUGANERI"
## [133] "PALAR AT VANIYAMBADI WATER SUPPLY HEAD WORK"
## [134] "TAMBIRAPARANI AT KALLIDAI KURICHI, TIRUNELVELI"
## [135] "TAMBIRAPARANI AT SRIVAIKUNTAM, D/S OF SK ANAICUT, TIRUNELVELI"
## [136] "TAMBIRAPARANI AT VELLAKOIL, TIRUNELVELI"
## [137] "TAMBIRAPARANI AT SIVALAPERI, CONFLUENCE POINT OF KUTTRALAM FALLS, D/S OF PALAYAMKOTTAI, TIRUN
## [138] "GHAGGAR AT RATANHERI, D/S OF PATIALA NADI (AFTER CONFL.)"
## [139] "HARYANA"
## [140] "GAGGAR D/S HANUMANGARH NEAR 2 KNJ NAI ABADI, MAKKASR,"
## [141] "KALA AMB D/S MARKANDA RIVER"
## [142] "KODRA DAM, MOUNT ABU, "
## [143] "MARKANDA U/S KALA AMB"
## [144] "UMTREW AT BYRNIHAT EAST"
## [145] "KYRHUKHLA NEAR SUTNGA KHLIERIAT,JAINTIA HILLS DT."
## [146] "DAMANGANGA AT LAVACHA TEMPLE, SILVASSA"
## [147] "DAMANGANGA D/S OF M/S SURAT BEVERAGES, VILLAGE DADRA, SILVASSA"
## [148] "DAMANGANGA AT NAROLI BRIDGE, SILVASSA"
## [149] "DAMANGANGA AT VILLAGE NAMDHA, VAPI"
## [150] "BALEHWAR KHADI AT N.H. NO. 8"
## [151] "RIVER PURNA ON BRIDGE AT SURATNANNAVSARI HIGHWAY"
## [152] "RIVER KAVERI ON BRIDGE AT BILLIMORANANVALSAD ROAD"
## [153] "RIVER DHADAR AT KOTHADA"
## [154] "RIVER BHOGAVO D/S OF SURENDRANAGAR."
## [155] "TRIVENI SANGAM, NR. SOMNATH TEMPLE, VERAVAL, DIST. JUNAGADH."
## [156] "RIVER KHANDEPAR AT CODLI NEAR BRIDGE ,U/S OPA WATERWORKS,SANGUEM"
## [157] "RIVER SAL PAZORKHONI,CUNCOLIM(NEAR CULVERT MARGAONAN CANACONA NATIONAL HIGHWAY)"
## [158] "RIVER KUSHAWATI NEAR BUND AT KEVONA,RIVON,SANGUEM"
## [159] "RIVER MAPUSA ON CULVERT ON HIGHWAY MAPUSANANPANAJI"
## [160] "VASHISTI RIVER AT U/S OF THREE M PAPER MILLS NEAR M/S MULTIFILMS PLASTIC PVT. LTD. AT KHERDI.
## [161] "SAVITRI RIVER AT OVALE VILLAGE."
## [162] "R PALLICKAL AT NELLIMUKAL"
## [163] "R PUZHACKAL AT PUZHACKAL BRIDGE"
## [164] "R CORAPUZHA AT KANAYANKODE"
## [165] "R ANCHARAKANDY AT MERUVAMBA"
## [166] "R RAMAPURAM AT RAMAPURAM BRIDGE"
## [167] "R PERUVAMBA AT CHANDAPPURA"
## [168] "R NEELASWARAM AT NAMBIARKAL DAM"
## [169] "R PULLUR AT PULLUR BR."
## [170] "R MOGRAL AT MOGRAL BR."
## [171] "R SHRIYA AT ANGADIMOGARU"
```

```
## [172] "R UPPALA AT  UPPALA BR."
## [173] "R MANJESWAR AT BAJRAKKARA BR."
## [174] "R KORAYAR AT KANJIKODE"
## [175] "R BHARATHAPUZHA AT KUTTIPPURAM"
## [176] "R BHARATHAPUZHA AT PATTAMBI"
## [177] "RIVER PERIYAR AT MUPPATHADAM"
## [178] "RIVER PERIYAR AT PATHALAM"
## [179] "R PERIYAR AT KALAMASSERY"
## [180] "R PERIYAR AT PURAPPALLIKAVU"
## [181] "R KADAMBAYAR AT BRAHMAPURAM"
## [182] "R KADAMBAYAR AT MANCKAKADAVU"
## [183] "NAGAVALI PENTA U/S, JAYKAYPUR TOWN"
## [184] "RIVER KERANDI (INTAKE WELL OF NALCO REFINARY, HAL, SUNABEDA)"
## [185] "Vansdhara MUNIGUDA (D/S OF M/S VEDANTAALUMINA PROJECT)"
## [186] "Vansdhara GUNUPUR (INTERSTATE BOUNDRY)"
## [187] "RIVER VAMSHADHARA, KALINGAPATNAM, VIZIANAGARAM"
## [188] "R KARUVANNURR AT KARUVANNUR BDG."
## [189] "RIVER VAMSHADHARA, KALINGAPATNAM,VIZIANAGARAM"
## [190] "RIVER DHADHAR AT KOTHWADA"
```

```
#install.packages("stringr")
library(stringr)
print(paste("Number of unique sates generated : ",length(unique(water_data[str_detect(water_data$states
```

```
## [1] "Number of unique sates generated :  17"
```

```
unique(water_data[str_detect(water_data$states, pattern),]$states)
```

```
##  [1] "GOA"              "MAHARASHTRA"      "KERALA"           "ANDHRA PRADESH"
##  [5] "KARNATAKA"        "ODISHA"           "TAMIL NADU"       "PUNJAB"
##  [9] "RAJASTHAN"        "HIMACHAL PRADESH" "MEGHALAYA"        "MIZORAM"
## [13] "TRIPURA"          "GUJARAT"          "MADHYA PRADESH"   "DAMAN AND DIU"
## [17] "HARYANA"
```

```
# number of states matching with dataset
nrow(water_data[str_detect(water_data$states, pattern),])
```

```
## [1] 1322
```

```
nrow(water_data)
```

```
## [1] 1577
```

```
# arranging the data frame
#View(water_data)
water_data_clean = water_data[str_detect(water_data$states, pattern),][,-c(1,2,3)]
water_data_clean = water_data_clean[,c(9,10,1,2,3,4,5,6,7,8)]
water_data_clean <- rename(water_data_clean,STATE=states)
#View(water_data_clean)
```

Removing the missing data here, as it is comparatively very low, and can't be replaced with other data, as it would lead it wrong conclusions

```
nrow(cropData)
```

```
## [1] 168892
```

```
sum(is.na(cropData))
```

```
## [1] 3076
```

```
cropData = na.omit(cropData)
```

#Extracted data

```
sWater_data <- water_data_clean
nrow(sWater_data)
```

```
## [1] 1322
```

```
#View(sWater_data)
years = unique(waterData$year)
#sCrop_data <- cropData %>%
#  filter(Crop_Year %in% years)
states <- tolower(unique(water_data[str_detect(water_data$states, pattern),]$states))
req_Crop_data <- cropData %>%
  filter(tolower(State_Name) %in% states)
nrow(req_Crop_data)
```

```
## [1] 88571
```

```
summary(lm(Production~Area, cropData))
```

```
##
## Call:
## lm(formula = Production ~ Area, data = cropData)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
##  -26350571    -423363    -349877    -341017 1248322491
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.398e+05  4.625e+04   7.349 2.01e-13 ***
## Area        2.585e+01  1.178e+00  21.933  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18010000 on 165814 degrees of freedom
## Multiple R-squared:  0.002893,   Adjusted R-squared:  0.002887
## F-statistic: 481.1 on 1 and 165814 DF,  p-value: < 2.2e-16
```

```
sCrop_data = req_Crop_data %>% group_by(Crop_Year, State_Name) %>%
  summarize(totalArea=mean(Area), totalProd=mean(Production))
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
sWater_data = sWater_data %>% group_by(year, STATE) %>%
  summarize(Temp=mean(Temp), DO=mean(DO), pH = mean(pH),Conductivity=mean(Conductivity), BOD = mean(BOD
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

Merging data

```
sCrop_data <- rename(sCrop_data,STATE=State_Name)
sCrop_data$STATE <- tolower(sCrop_data$STATE)
sWater_data$STATE <- tolower(sWater_data$STATE)

sCrop_data = rename(sCrop_data, year=Crop_Year)
# View(sCrop_data)
# View(sWater_data)
com_data <- merge(sWater_data,sCrop_data,by=c("year","STATE"))
# View(com_data)
```

```
#nrow(com_data)
#View(waterData)
sum(is.na(com_data))
```

```
## [1] 0
```

Basic relation ship analysis

```
AreaRel = lm(totalProd~totalArea, com_data)
summary(AreaRel)
```

```
##
## Call:
## lm(formula = totalProd ~ totalArea, data = com_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3977500 -3604168 -2567155  -442737 28288536
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4011008.07 1017678.47   3.941 0.000138 ***
## totalArea       -94.19      57.96  -1.625 0.106816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7523000 on 117 degrees of freedom
## Multiple R-squared:  0.02208,    Adjusted R-squared:  0.01372
## F-statistic: 2.641 on 1 and 117 DF,  p-value: 0.1068
```

Checking for basic relationship between totalProduction and all the other attributes

```
AllRel = lm(totalProd~., com_data[,c(-2)])
summary(AllRel)
```

```
##
## Call:
## lm(formula = totalProd ~ ., data = com_data[, c(-2)])
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -5650865 -4036524 -1871696     5952 27934326
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.519e+08  5.337e+08  -0.285    0.776
## year           7.608e+04  2.656e+05   0.286    0.775
## Temp           2.896e+05  2.407e+05   1.203    0.232
## DO            -5.861e+05  1.028e+06  -0.570    0.570
## pH            -9.925e+03  4.377e+04  -0.227    0.821
## Conductivity   8.682e+01  2.943e+02   0.295    0.769
## BOD           -5.420e+04  5.569e+04  -0.973    0.333
## Nitrates      -2.386e+05  3.352e+05  -0.712    0.478
## FecalColiform  7.192e-01  2.319e+00   0.310    0.757
## TotalColiform -5.775e-01  1.587e+00  -0.364    0.717
## totalArea     -8.377e+01  6.830e+01  -1.226    0.223
##
## Residual standard error: 7645000 on 108 degrees of freedom
## Multiple R-squared:  0.0677, Adjusted R-squared:  -0.01862
## F-statistic: 0.7843 on 10 and 108 DF,  p-value: 0.6437
```

```
str(com_data)
```

```
## 'data.frame':    119 obs. of  12 variables:
##  $ year         : int  2003 2003 2003 2003 2003 2003 2003 2004 2005 2005 ...
##  $ STATE        : chr  "gujarat" "karnataka" "kerala" "maharashtra" ...
##  $ Temp         : num  28 27 26.8 26.1 29 ...
##  $ DO           : num  7.7 7.7 6.93 7.54 7.6 ...
##  $ pH           : num  88.1 88.1 68.5 88.1 88.1 ...
##  $ Conductivity : num  4.8 6.5 6.53 6.29 7.47 ...
##  $ BOD          : num  6.9 2.4 0.879 5.7 2.1 ...
##  $ Nitrates     : num  2 0.9 0.366 0.58 0.996 ...
##  $ FecalColiform: num  2 0.9 0.366 0.58 0.996 ...
##  $ TotalColiform: num  10050 688 1966 224 6375 ...
##  $ totalArea    : num  17772 8557 5678 19786 6899 ...
##  $ totalProd    : num  64414 22205 15851074 52329 10639 ...
```

```
crops = unique(req_Crop_data$Crop)
print(crops)
```

```
##   [1] "Arhar/Tur"                "Bajra"
##   [3] "Castor seed"              "Cotton(lint)"
##   [5] "Dry chillies"             "Groundnut"
##   [7] "Horse-gram"               "Jowar"
##   [9] "Korra"                    "Maize"
##  [11] "Moong(Green Gram)"        "Onion"
##  [13] "Other Kharif pulses"      "Ragi"
##  [15] "Rice"                     "Samai"
##  [17] "Small millets"            "Soyabean"
##  [19] "Sunflower"                "Urad"
##  [21] "Gram"                     "Linseed"
##  [23] "Other  Rabi pulses"       "Safflower"
##  [25] "Wheat"                    "Arecanut"
##  [27] "Banana"                   "Bhindi"
##  [29] "Brinjal"                  "Citrus Fruit"
```

```
##  [31] "Coconut "                  "Coriander"
##  [33] "Cucumber"                  "Grapes"
##  [35] "Mango"                     "Orange"
##  [37] "Other Fresh Fruits"        "Other Vegetables"
##  [39] "Papaya"                    "Pome Fruit"
##  [41] "Sugarcane"                 "Sweet potato"
##  [43] "Tobacco"                   "Tomato"
##  [45] "Potato"                    "Turmeric"
##  [47] "Mesta"                     "Cowpea(Lobia)"
##  [49] "other oilseeds"            "Lemon"
##  [51] "Pome Granet"               "Sapota"
##  [53] "Cabbage"                   "Cashewnut"
##  [55] "Sesamum"                   "Beans & Mutter(Vegetable)"
##  [57] "Tapioca"                   "Rapeseed &Mustard"
##  [59] "Niger seed"                "Bottle Gourd"
##  [61] "Dry ginger"                "Varagu"
##  [63] "Garlic"                    "Ginger"
##  [65] "Sannhamp"                  "Black pepper"
##  [67] "Pineapple"                 "Moth"
##  [69] "Guar seed"                 "Other Cereals & Millets"
##  [71] "Barley"                    "Masoor"
##  [73] "Peas & beans (Pulses)"     "Cardamom"
##  [75] "Bitter Gourd"              "Drum Stick"
##  [77] "Jack Fruit"                "Snak Guard"
##  [79] "other misc. pulses"        "Cauliflower"
##  [81] "Jute"                      "Khesari"
##  [83] "Other Citrus Fruit"        "Water Melon"
##  [85] "Oilseeds total"            "other fibres"
##  [87] "Ash Gourd"                 "Beet Root"
##  [89] "Lab-Lab"                   "Pump Kin"
##  [91] "Redish"                    "Ribed Guard"
##  [93] "Yam"                       "Apple"
##  [95] "Carrot"                    "Peach"
##  [97] "Pear"                      "Plums"
##  [99] "Litchi"                    "Ber"
## [101] "Jute & mesta"
```

```
which.max(table(req_Crop_data$Crop))
```

```
## Rice
##   81
```

Function for analysing the linear relationships

```
cropWiseRel = function(crop){
  DrumStickData = req_Crop_data %>%
  filter(Crop == crop)
  DrumStickData = DrumStickData %>% group_by(Crop_Year, State_Name) %>%
    summarize(totalArea=mean(Area), totalProd=mean(Production))
  DrumStickData <- rename(DrumStickData,STATE=State_Name, year=Crop_Year)
  DrumStickData$STATE <- tolower(DrumStickData$STATE)
  #View(DrumStickData)
  com_data <- merge(sWater_data,DrumStickData,by=c("year","STATE"))
  nrow(com_data)
```

```
  DS_AreaRel = lm(totalProd~totalArea, com_data)
  #summary(DS_AreaRel)
  print(paste(crop, " : "))
  print(DS_AreaRel$coefficients)
  print(cor(com_data$totalArea, com_data$totalProd))
  #cropWiseRel = DS_AreaRel
}
```

This is done to analyse the relation between crops area and their production

```
for(i in crops){
  try(cropWiseRel(i))
    #print(paste("*********Error with : ", i))
}
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Arhar/Tur  : "
## (Intercept)   totalArea
## 150.7644076   0.6706558
## [1] 0.9426845
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Bajra  : "
##  (Intercept)    totalArea
## 2504.2125406    0.8056589
## [1] 0.9240012
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Castor seed  : "
##  (Intercept)    totalArea
## -1766.767715     1.916114
## [1] 0.9612852
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Cotton(lint)  : "
##  (Intercept)    totalArea
## 11400.919225     2.131518
## [1] 0.8618364
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Dry chillies  : "
##  (Intercept)    totalArea
## -2337.459907     3.938498
## [1] 0.9082757
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Groundnut  : "
## (Intercept)   totalArea
## 1480.777900    1.024797
## [1] 0.8313038
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Horse-gram  : "
## (Intercept)   totalArea
##  33.7423852   0.4130988
## [1] 0.901265
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Jowar  : "
## (Intercept)   totalArea
##  2309.00599     0.73566
## [1] 0.9614209
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Korra  : "
## (Intercept)   totalArea
##    60.61111          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Maize  : "
## (Intercept)   totalArea
## 9683.659522    1.716007
## [1] 0.7069523
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Moong(Green Gram)  : "
## (Intercept)   totalArea
## 150.5406044   0.3671067
## [1] 0.8305556
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Onion  : "
## (Intercept)   totalArea
##  1583.12566    13.73524
## [1] 0.6279869
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Other Kharif pulses  : "
## (Intercept)   totalArea
## 114.6954898   0.4598883
## [1] 0.7954034
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Ragi  : "
## (Intercept)   totalArea
## -514.466745    1.401326
## [1] 0.9535556
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Rice  : "
##   (Intercept)     totalArea
## -25650.177384      3.405099
## [1] 0.9303315
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##   0 (non-NA) cases
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Small millets  : "
## (Intercept)   totalArea
##  11.3382316   0.6371966
## [1] 0.9290279
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Soyabean  : "
## (Intercept)   totalArea
## 941.6545794   0.9585864
## [1] 0.9180091
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Sunflower  : "
## (Intercept)   totalArea
## -50.3582832   0.6222855
## [1] 0.9345412
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Urad  : "
##  (Intercept)    totalArea
## -373.9888141    0.6044315
## [1] 0.9273647
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Gram  : "
##  (Intercept)    totalArea
## -759.2127367    0.9319047
## [1] 0.9187983
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Linseed  : "
## (Intercept)   totalArea
##  53.4055948   0.2516351
## [1] 0.8702332
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Other  Rabi pulses  : "
## (Intercept)   totalArea
##  73.6400895   0.6904206
## [1] 0.849254
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Safflower  : "
## (Intercept)   totalArea
## 196.1299677   0.5479065
## [1] 0.9519884
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Wheat  : "
##   (Intercept)     totalArea
```

```
## -32303.669287      4.438819
## [1] 0.978526
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Arecanut  : "
##  (Intercept)    totalArea
## -7265.772779     5.767299
## [1] 0.7444207
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Banana  : "
## (Intercept)   totalArea
## 26511.43231    21.35769
## [1] 0.4018947
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Bhindi  : "
## (Intercept)   totalArea
##  -705.17476    10.61553
## [1] 1
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Brinjal  : "
## (Intercept)   totalArea
## -1140.23713    13.88694
## [1] 0.9289652
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Citrus Fruit  : "
## (Intercept)   totalArea
##  6418.36157   -12.94744
## [1] -0.4643587
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Coconut   : "
##   (Intercept)     totalArea
## -41370793.356      7621.138
## [1] 0.9181681
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Coriander  : "
## (Intercept)   totalArea
## -492.705760    1.010721
## [1] 0.9393075
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Cucumber  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Grapes  : "
## (Intercept)   totalArea
```

```
##   -54.89388    27.26780
## [1] 0.998946
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Mango  : "
##   (Intercept)     totalArea
## -16925.303584      8.592345
## [1] 0.985755
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Orange  : "
## (Intercept)   totalArea
##  -15.532906    2.071054
## [1] 1
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Other Fresh Fruits  : "
## (Intercept)   totalArea
## -3941.57510    12.13879
## [1] 0.9714815
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Other Vegetables  : "
##  (Intercept)    totalArea
## -5174.142185     8.487193
## [1] 0.6001782
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Papaya  : "
##  (Intercept)    totalArea
## 18884.163705    -4.397162
## [1] -0.11971
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Pome Fruit  : "
## (Intercept)   totalArea
## -389.218033    8.299496
## [1] 0.9990149
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Sugarcane  : "
##  (Intercept)    totalArea
## -12157.39459     83.52186
## [1] 0.9740623
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Sweet potato  : "
## (Intercept)   totalArea
##  251.643563    5.480511
## [1] 0.8258782
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Tobacco  : "
## (Intercept)   totalArea
```

```
## -369.249638    1.473743
## [1] 0.9050117
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Tomato  : "
##  (Intercept)    totalArea
## -25684.17641     35.53826
## [1] 0.7786953
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Potato  : "
## (Intercept)   totalArea
## -8530.76719    21.23692
## [1] 0.9160086
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Turmeric  : "
## (Intercept)   totalArea
## -460.084360    6.680868
## [1] 0.945078
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Mesta  : "
##  (Intercept)    totalArea
## -2443.446627     8.859708
## [1] 0.9984143
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Cowpea(Lobia)  : "
## (Intercept)   totalArea
## 211.3469625   0.3095507
## [1] 0.6894876
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "other oilseeds  : "
##  (Intercept)    totalArea
## -3238.861932     7.649785
## [1] 0.4369862
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Lemon  : "
##  (Intercept)    totalArea
## 18102.951030    -2.191311
## [1] -0.4075544
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Pome Granet  : "
## (Intercept)   totalArea
##  -532.27350    16.05293
## [1] 0.9599324
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Sapota  : "
##  (Intercept)    totalArea
```

```
## -48333.59861     79.77297
## [1] 0.7321973
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Cabbage  : "
## (Intercept)   totalArea
## -10915.8966    163.4281
## [1] 0.9679173
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Cashewnut  : "
##  (Intercept)    totalArea
## 1154.2267478    0.3530884
## [1] 0.8791483
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Sesamum  : "
## (Intercept)   totalArea
##  59.6380951   0.3779253
## [1] 0.8692234
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Beans & Mutter(Vegetable)  : "
## (Intercept)   totalArea
## -2404.66215    11.93112
## [1] 1
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Tapioca  : "
##  (Intercept)    totalArea
## -13358.25290     33.30759
## [1] 0.9209563
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Rapeseed &Mustard  : "
## (Intercept)   totalArea
##  103.060648    1.230146
## [1] 0.9854713
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Niger seed  : "
## (Intercept)   totalArea
## -96.5908518   0.3804662
## [1] 0.8670088
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Bottle Gourd  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Dry ginger  : "
## (Intercept)   totalArea
```

```
##  -3106.4530     17.3873
## [1] 0.7249941
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Varagu  : "
## (Intercept)   totalArea
##       991.4          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Garlic  : "
## (Intercept)   totalArea
## -406.940033    6.618921
## [1] 0.9303253
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Ginger  : "
## (Intercept)   totalArea
##        1735          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Sannhamp  : "
## (Intercept)   totalArea
##   3.4346976   0.4457737
## [1] 0.432563
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Black pepper  : "
##  (Intercept)    totalArea
## 1274.7924926    0.2186862
## [1] 0.4100028
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Pineapple  : "
## (Intercept)   totalArea
## 1518.735244    6.368928
## [1] 0.9307125
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Moth  : "
## (Intercept)   totalArea
##  650.279272    0.229098
## [1] 0.7264013
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Guar seed  : "
##  (Intercept)    totalArea
## 2470.2464967    0.3037797
## [1] 0.8492728
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Other Cereals & Millets  : "
## (Intercept)   totalArea
```

```
##  -93.226593    1.075586
## [1] 0.9700044
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Barley  : "
## (Intercept)   totalArea
##  437.899820    2.561347
## [1] 0.9289402
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Masoor  : "
## (Intercept)   totalArea
##  26.2450301   0.6602479
## [1] 0.9878415
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Peas & beans (Pulses)  : "
## (Intercept)   totalArea
##   194.05870     1.06864
## [1] 0.9451832
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Cardamom  : "
##  (Intercept)    totalArea
## -260.7230190    0.2600878
## [1] 0.8496591
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Bitter Gourd  : "
## (Intercept)   totalArea
##           0           0
```

```
## Warning in cor(com_data$totalArea, com_data$totalProd): the standard deviation
## is zero
```

```
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Drum Stick  : "
## (Intercept)   totalArea
## -279.682178    1.237833
## [1] 0.9986334
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Jack Fruit  : "
## (Intercept)   totalArea
## 975.8311937  -0.1434471
## [1] -1
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Snak Guard  : "
## (Intercept)   totalArea
##           0           0
```

```
## Warning in cor(com_data$totalArea, com_data$totalProd): the standard deviation
## is zero
```

```
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##   0 (non-NA) cases
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Cauliflower  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Jute  : "
## (Intercept)   totalArea
## -547.695091    9.690617
## [1] 0.9922843
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Khesari  : "
## (Intercept)   totalArea
##    932.9286          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Other Citrus Fruit  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Water Melon  : "
## (Intercept)   totalArea
##           0           0
```

```
## Warning in cor(com_data$totalArea, com_data$totalProd): the standard deviation
## is zero
```

```
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Oilseeds total  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "other fibres  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Ash Gourd  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Beet Root  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Lab-Lab  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Pump Kin  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Redish  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Ribed Guard  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Yam  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Apple  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Carrot  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Peach  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Pear  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Plums  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Litchi  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## [1] "Ber  : "
## (Intercept)   totalArea
##           0          NA
## [1] NA
```

```
## `summarise()` has grouped output by 'Crop_Year'. You can override using the `.groups` argument.
```

```
## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##   0 (non-NA) cases
```

the results here are too discrete

Usage of neural Network over Combined data

```
# install.packages("neuralnet")
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'
```

```
## The following object is masked from 'package:dplyr':
##
##     compute
```

for scaling the data and accuracy calculation

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
#install.packages("MLmetrics")
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following objects are masked from 'package:caret':
##
##     MAE, RMSE
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
# install.packages("Metrics")
```

Functions for neural network analysis

```
scale = function(s){
  mod = preProcess(data.frame(s), method = 'range')
  scale = predict(mod, data.frame(s))
  scale = scale$s
}

meanSqErr = function(a,b){
  suM = 0
  for(i in seq(length(a))){
    t = (a[i]-b[i,])^2
    suM = suM + t
  }
  return (suM/length(a))
}

stateWiseNNAcc = function(s){
  stateData = com_data %>%
  filter(STATE == s)
  stateData$totalArea = scale(stateData$totalArea)
  stateData$Temp = scale(stateData$Temp)
  stateData$DO = scale(stateData$DO)
  stateData$pH = scale(stateData$pH)
  stateData$Conductivity = scale(stateData$Conductivity)
  stateData$BOD = scale(stateData$BOD)
  stateData$Nitrates = scale(stateData$Nitrates)
  stateData$FecalColiform = scale(stateData$FecalColiform)
  stateData$TotalColiform = scale(stateData$TotalColiform)
  stateData$totalProd = scale(stateData$totalProd)
  # op_Sca = preProcess(data.frame(stateData$totalProd), method = "range")
  # stateData$totalProd = as.numeric(predict(op_Sca, data.frame(stateData$totalProd))$stateData.totalPr
  # View(stateData)
  par_ProRel = neuralnet(totalProd~totalArea+Temp+DO+pH+Conductivity+BOD+Nitrates+FecalColiform+TotalCo

  res = predict(par_ProRel, stateData[c(-12)])
  print(paste(paste(paste("RMSE in state ", s), " is "), meanSqErr(res, stateData[c(12)])))
  return(par_ProRel)
```

```
}
```

```
# mses = c()
unique(com_data$STATE)
```

```
##  [1] "gujarat"          "karnataka"        "kerala"           "maharashtra"
##  [5] "odisha"           "rajasthan"        "tamil nadu"       "andhra pradesh"
##  [9] "goa"              "himachal pradesh" "meghalaya"        "punjab"
## [13] "tripura"          "daman and diu"    "haryana"          "madhya pradesh"
```

```
# summary(stateWiseNNAcc("tamil nadu"))
# plot(stateWiseNNAcc("tamil nadu"))
```

Analysis using the weights generated by neural network, for each state

```
plot(stateWiseNNAcc("gujarat"))
```

```
## [1] "RMSE in state  gujarat  is  0.00291015053028963"
```

Inference : In gujarat it can be seen that, increase in the temperature, Nitrates is leading to decrease in total production, while increase in the DO, BOD is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("karnataka"))
```

```
## [1] "RMSE in state  karnataka  is  0.00754727457770838"
```

Inference : In karnataka it can be seen that, increase in the DO,pH,Conductivity,FecalColiform is leading to decrease in total production, while increase in the BOD,NItrates is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("kerala"))
```

```
## [1] "RMSE in state  kerala  is  0.0110426953286201"
```

Inference : In kerala it can be seen that, increase in the DO,pH,BOD, Notrates and TotalColiform is leading to decrease in total production, while increase in the temperature, FecalColiform is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("maharashtra"))
```

```
## [1] "RMSE in state  maharashtra  is  0.0081551380941276"
```

Inference : In maharastra it can be seen that, increase in the Temperature,DO,pH,Noterates and TotalColiform is leading to decrease in total production, while increase in the BOD, FecalColiform is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("odisha"))
```

```
## [1] "RMSE in state  odisha  is  0.00633146782575212"
```

Inference : In odisha it can be seen that, increase in the temperatue,DO,pH,BOD,FecalColiform is leading to decrease in total production, while increase in the TotalColiform is leading to the increase in the total crop production

```r
plot(stateWiseNNAcc("rajasthan"))
```

```
## [1] "RMSE in state  rajasthan  is  0.131700482727121"
```

Inference : In rajasthan it can be seen that, increase in most of the variables is leading to the increase in the total crop production

```r
plot(stateWiseNNAcc("madhya pradesh"))
```

```
## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'
```

```
## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## [1] "RMSE in state  madhya pradesh  is  9.70528559126645e-06"
```

Inference : In Madhya Pradesh it can be seen that, increase in the Temperature,DO,pH,Nitrates is leading to decrease in total production, while increase in the Conductivity,BOD, TotalColiform is leading to the increase in the total crop production

```r
plot(stateWiseNNAcc("tamil nadu"))
```

```
## [1] "RMSE in state  tamil nadu  is  0.000351410463544572"
```

Inference : In Tamil Nadu it can be seen that, increase in the DO,pH,BOD,TotalColiform is leading to decrease in total production, while increase in the Temperature,Conductivity,Niterates, is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("andhra pradesh"))
```

```
## [1] "RMSE in state  andhra pradesh  is  0.0148821726289864"
```

Inference : In ANdhra Pradesh it can be seen that increase in the Temperature,BOD,TotalColiform is leading to decrease in total production, while increase in the DO,pH,Conductivity,Niterates,FecalColiform is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("goa"))
```

```
## [1] "RMSE in state  goa  is  0.00146785662634067"
```

Inference : In Goa it can be seen that increase in the FecalColiform,BOD is leading to decrease in total production, while increase in the DO,TotalColiform is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("himachal pradesh"))
```

```
## [1] "RMSE in state  himachal pradesh  is  0.12670691173834"
```

Inference : In Himachal Pradesh it can be seen that increase in the pH,BOD,Niterates is leading to decrease in total production, while increase in the FecalColiform,TotalColiform is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("meghalaya"))
```

```
## [1] "RMSE in state  meghalaya  is  0.0787242071383724"
```

Inference : In Meghalaya it can be seen that, increase in the Temperature,DO,BOD,Niterates,FecalColiform,TotalCoiform is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("punjab"))
```

```
## [1] "RMSE in state  punjab  is  0.00230584904165228"
```

Inference : In Punjab it can be seen that increase in the DO,Conductivity,TotalColiform is leading to decrease in total production, while increase in the Temperature,Nitrates,FecalColiform is leading to the increase in the total crop production

```
plot(stateWiseNNAcc("tripura"))
```

```
## [1] "RMSE in state  tripura  is  0.00297788056735351"
```

Inference : In Tripura it can be seen that increase in the Nitrates,FecalColiform,DO is leading to decrease in total production, while increase in the pH,Conductivity,BOD is leading to the increase in the total crop production

```r
plot(stateWiseNNAcc("daman and diu"))
```

```
## [1] "RMSE in state  daman and diu  is  0.000145072107985249"
```

Inference : In Daman and Diu it can be seen that increase in the DO,Nitrates,FecalColiform,TotalColiform is leading to decrease in total production, while increase in the BOD is leading to the increase in the total crop production

```r
plot(stateWiseNNAcc("haryana"))
```

```
## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'
```

```
## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## Warning in preProcess.default(data.frame(s), method = "range"): No variation for
## for: s

## Warning in sweep(x[, method$range, drop = FALSE], 2, rangeBounds[1], "+"): STATS
## is longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(newdata[, object$method$range, drop = FALSE], 2,
## rangeBounds$lower, : STATS is longer than the extent of 'dim(x)[MARGIN]'

## [1] "RMSE in state  haryana  is  5.41531176465518e-05"
```

Inference : In Haryana it can be seen that, increase in the DO,pH,BOD is leading to decrease in total production

It can be be seen that in some of the states, the totalArea had a negative impact on the total production, which is due to the diversity of the crops information present in each state, this can be resolved by analysing the information of each crop in each state, which is the future scope for this project.We had implemented Random forest algorithm as well for better understanding of the behavior of chemical on the crop, which is given below

Random Forest

```
# install.packages("caret", dependencies = TRUE)
# install.packages("randomForest")
library(caret)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(MLmetrics)
```

```
set.seed(51)
com_data$yield = com_data$totalProd/com_data$totalArea
MAE=c()
MSE=c()
RMSE=c()
R2=c()
states = unique(com_data$STATE)
statewiseRF = function(){
  for(i in states){
    state = com_data %>%
      filter(STATE == i)
    if(nrow(state)>1){
      model <- train(yield ~ Temp + DO + pH + Conductivity + BOD + Nitrates + FecalColiform + TotalColi
                     ,trControl = trainControl(method = 'cv',number = 4))
      pred_yeild = predict(model, com_data[-c(1,2,11,12,13)])
      fmae = MAE(pred_yeild, com_data$yield)
      fmse = MSE(pred_yeild, com_data$yield)
      frmse = RMSE(pred_yeild, com_data$yield)
      fr2 = R2(pred_yeild, com_data$yield)
      MAE <<- append(MAE,fmae)
      MSE <<- append(MSE,fmse)
      RMSE <<- append(RMSE,frmse)
      R2 <<- append(R2,fr2)
      cat(i,":\n")
      cat(" MAE:", fmae, "\n", "MSE:", fmse, "\n", "RMSE:", frmse, "\n", "R-squared:", fr2, "\n")
    }
  }
}

options(warn=-1)
statewiseRF()
```

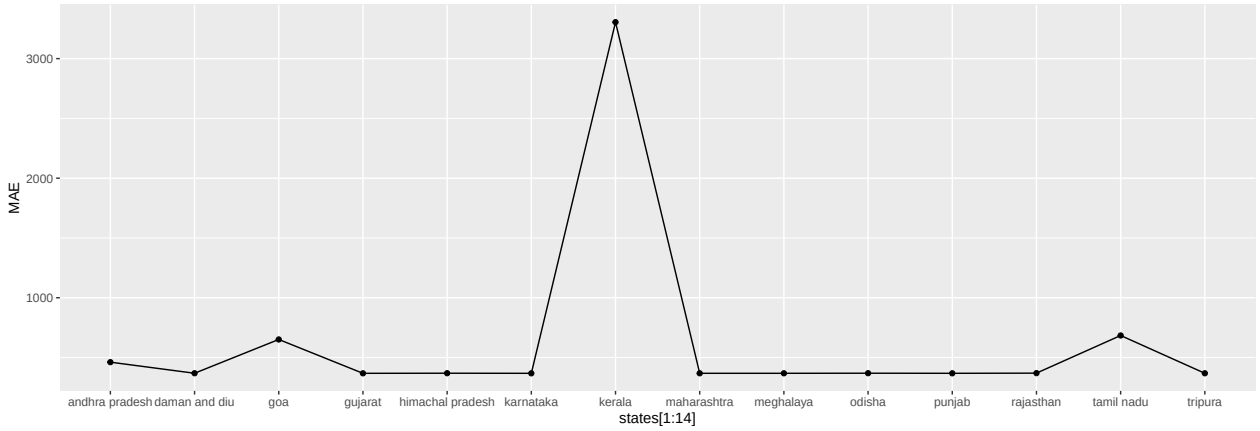
```
## gujarat :
```

```
##  MAE: 368.0038
##  MSE: 1210118
##  RMSE: 1100.054
##  R-squared: 0.05026292
## karnataka :
##  MAE: 367.9838
##  MSE: 1209348
##  RMSE: 1099.704
##  R-squared: 0.02397724
## kerala :
##  MAE: 3304.921
##  MSE: 11900506
##  RMSE: 3449.711
##  R-squared: 0.031253
## maharashtra :
##  MAE: 368.1314
##  MSE: 1209041
##  RMSE: 1099.564
##  R-squared: 0.04035512
## odisha :
##  MAE: 368.9376
##  MSE: 1211617
##  RMSE: 1100.735
##  R-squared: 0.01223768
## rajasthan :
##  MAE: 369.4318
##  MSE: 1212058
##  RMSE: 1100.935
##  R-squared: 0.02429402
## tamil nadu :
##  MAE: 684.3467
##  MSE: 1110525
##  RMSE: 1053.815
##  R-squared: 0.009708527
## andhra pradesh :
##  MAE: 461.178
##  MSE: 1121380
##  RMSE: 1058.952
##  R-squared: 0.002621338
## goa :
##  MAE: 651.0588
##  MSE: 1202671
##  RMSE: 1096.663
##  R-squared: 0.01962454
## himachal pradesh :
##  MAE: 368.8303
##  MSE: 1211543
##  RMSE: 1100.701
##  R-squared: 0.02754817
## meghalaya :
##  MAE: 368.1418
##  MSE: 1210691
##  RMSE: 1100.314
##  R-squared: 9.614935e-05
```

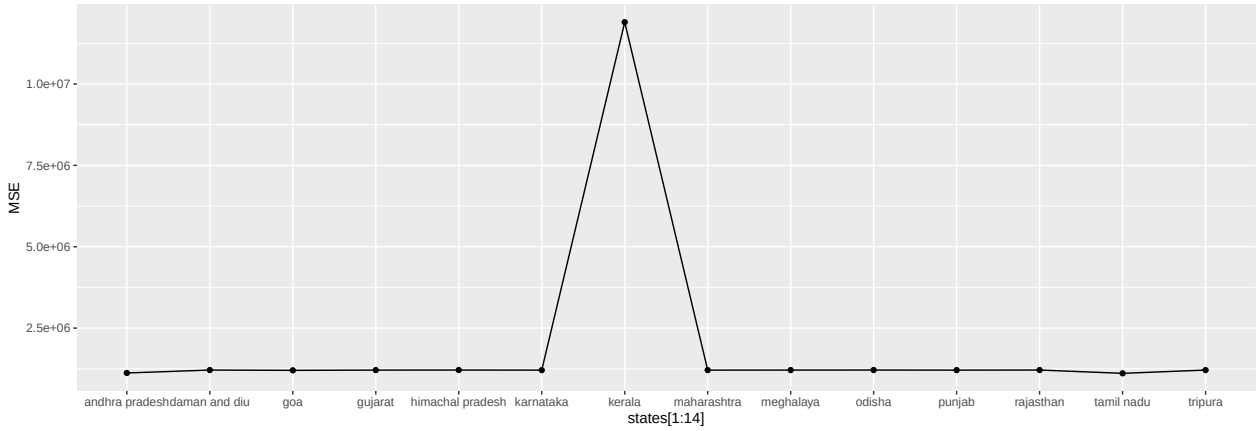
```
## punjab :
##  MAE: 368.0641
##  MSE: 1209279
##  RMSE: 1099.672
##  R-squared: 0.03215256
## tripura :
##  MAE: 368.2483
##  MSE: 1210864
##  RMSE: 1100.393
##  R-squared: 0.08391543
## daman and diu :
##  MAE: 368.5188
##  MSE: 1211316
##  RMSE: 1100.598
##  R-squared: 0.08400347
```

```
options(warn=0)
```

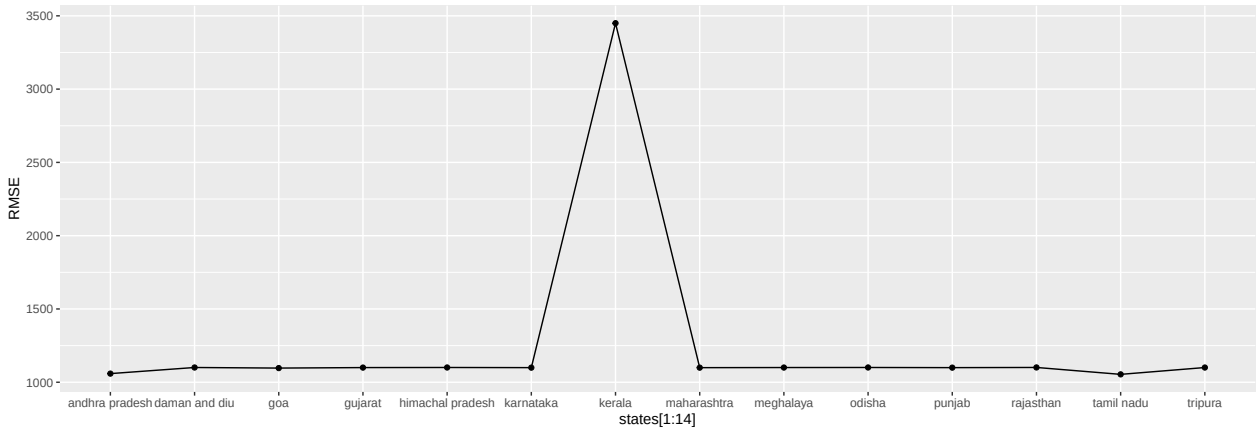
```
ggplot(data.frame(states[1:14],MAE), aes(states[1:14],MAE)) + geom_point() + geom_line(aes(group = 1))
```



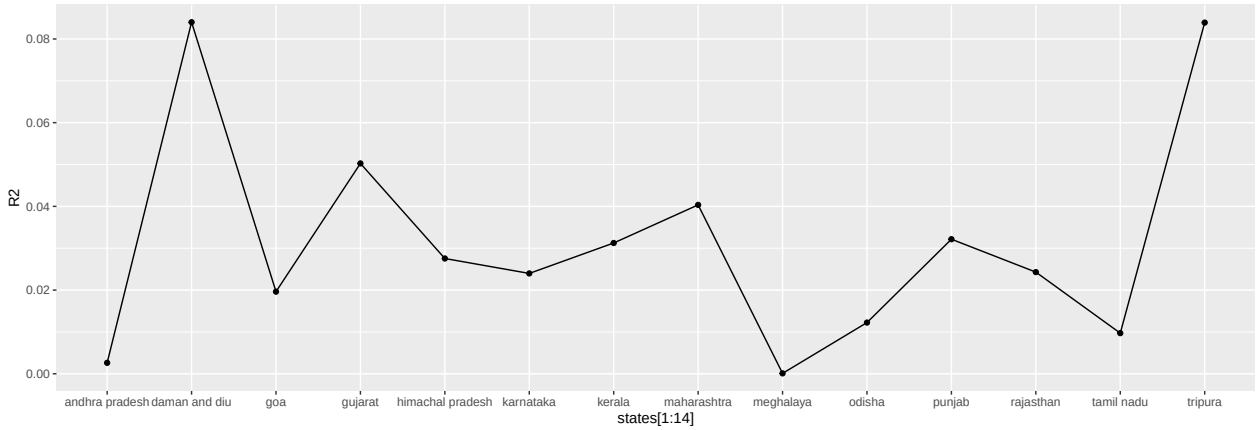

```
ggplot(data.frame(states[1:14],MSE), aes(states[1:14],MSE)) + geom_point() + geom_line(aes(group = 1))
```




```
ggplot(data.frame(states[1:14],RMSE), aes(states[1:14],RMSE)) + geom_point() + geom_line(aes(group = 1))
```




```
ggplot(data.frame(states[1:14],R2), aes(states[1:14],R2)) + geom_point() + geom_line(aes(group = 1))
```



This model can be used to predict the total production for reessuring the surity of the predictions, and can be used for ensembling as a future scope.