India States and United Territories: Percent Female

Brandon Tao

March 2019

Notes

The 2001 data used here is from previous "Demographics_Of_India - wikepedia" content. The current content has 2011 data. There are additional tables of data of possible interest. For example there is national table of statistics for seven religious groups categories. Variables include ten year growth percent, urban, rural and composite sex ratios, literacy percent and work participation percent.

Sections

0. Setup

This setup defines a function used later in the script.

```
library(tidyverse)
library(lattice)
source('hw.r')
```

1. Read the data into R, look at it, filter rows and select columns

To explore a data set with statistics on India's States and United Territories. Our choice is focus on percent female and for other data set variables are might reasonable use in an model that that helps explain the variation in percent female. As we look at the data we can also think about potential relevant variables that are not at hand.

The exploratory process can include literature searches and getting guidance from those with expertise on the topic.

Here we just jump start looking for patterns.

Using View(), or other way2 to look a the data, we can quickly see that one row provides the total for India. The filtering is to omit this row.

```
indiaAll <- read_csv(file = "India 2001 Wiki.csv")
indiaAll[11:15,]
```

```
## # A tibble: 5 x 13
##   StateUT_Index State_UT  Type   Population PopPctTotal AreaKm2 Density
##   <chr>         <chr>     <chr>       <int>       <dbl>   <int>   <int>
## 1 15            Assam     State    26655528        2.58   78438     397
## 2 11            Odisha    State    36804660        3.47  155707     269
## 3 TOTAL         India     29 + 7 1210726932      100    3287240     382
## 4 24            Meghalaya State     2318822        0.25   22429     132
## 5 8             Karnataka State    52850562        5.05  191791     319
## # ... with 6 more variables: Males <int>, Females <int>, SexRatio <int>,
## #   Literacy <dbl>, RuralPop <int>, UrbanPop <int>
```

```
# In the StateUT_Index column we see that the middle row,
# row 13 is the total line for all of India.
#
# The script below shows one way to remove
# row 13 that has State_UT = "India".
```

Below we use %>% to pipe the indiaAll tibble into the filter function as its first argument. We specify keeping all rows whose State_UT value is not equal (!=) to “India”.

We pipe the resulting tibble into the first argument of the select function. The function second argument specifies the columns that we want to keep or remove. We put a minus sign (-) in front of the State_UT column to remove this unneeded column.

Finally the script assigns the name india1 to the resulting tibble and puts it in the workspace.

```
india1 <- indiaAll %>%
   filter(State_UT != "India") %>%
   select(-StateUT_Index)
india1$State_UT
```

```
##  [1] "Bihar"                      "Arunachal Pradesh"
##  [3] "Rajasthan"                  "Jharkhand"
##  [5] "Telangana"                  "Jammu and Kashmir"
##  [7] "Andhra Pradesh*"            "Uttar Pradesh"
##  [9] "Madhya Pradesh"             "Chhattisgarh"
## [11] "Assam"                      "Odisha"
## [13] "Meghalaya"                  "Karnataka"
## [15] "Haryana"                    "Punjab"
## [17] "Dadra and Nagar Haveli"     "West Bengal"
## [19] "Gujarat"                    "Uttarakhand"
## [21] "Manipur"                    "Nagaland"
```

```
## [23] "Tamil Nadu"                    "Sikkim"
## [25] "Maharashtra"                   "Himachal Pradesh"
## [27] "Pondicherry"                   "Chandigarh"
## [29] "Delhi"                         "Andaman and Nicobar Islands"
## [31] "Daman and Diu"                 "Goa"
## [33] "Mizoram"                       "Lakshadweep"
## [35] "Tripura"                       "Kerala"
```

2. Data transformations using the mutate function

Our interest focuses on the percent of females in the States and United Territories of India. We compute the percent. Some may prefer to use the sex ratio that came with the data set. This is the number of females per 1000 males.

While parental choices are directly associated to the variation in percents, the choices in turn may be influenced by society and environmental variables. We can make some plots as think about modeling the percent female as a function of variables in the data set or obtaining more variables from additional sources.

Variables with the data set, such as percent literacy, might be related to percent female. There might be urban and rural patterns so we compute percent urban below. There may be a relationship to population density so we compute that below. We also transform some positive variables with a thick right tail using a log transformation.

```
# A reminder of the names
colnames(india1)
```

```
##  [1] "State_UT"    "Type"        "Population"  "PopPctTotal" "AreaKm2"
##  [6] "Density"     "Males"       "Females"     "SexRatio"    "Literacy"
## [11] "RuralPop"    "UrbanPop"
```

```
# Here we put the original and mutated variables
# in the new tibble, india2.

india2 <- india1 %>%
  mutate(
    FemalePct = 100 * Females/(Females + Males),
    UrbanPct =  100 * UrbanPop/(UrbanPop + RuralPop),
    lAreaKm2 = log(AreaKm2),
    lUrbanPct = log(100 * UrbanPop/(UrbanPop + RuralPop)),
    lPopPctTotal = log(PopPctTotal)
  )

colnames(india2)
```

```
##  [1] "State_UT"     "Type"         "Population"   "PopPctTotal"
##  [5] "AreaKm2"      "Density"      "Males"        "Females"
##  [9] "SexRatio"     "Literacy"     "RuralPop"     "UrbanPop"
## [13] "FemalePct"    "UrbanPct"     "lAreaKm2"     "lUrbanPct"
## [17] "lPopPctTotal"
```

3. Looking at candidate variables with a scatterplot matrix and smoothes

First, we pick a subset of variables

```
indiaExp <- select(india2,FemalePct, Literacy,
  lAreaKm2:lPopPctTotal)

# We provide better labels for these variable
# that include the units of measure.  These will
# be used in the splom function as the argument varnames.

varnames = c("Percent\nFemale","Percent\nLiterate",
  "Pop. Density Per\nSquare Kilometers","Percent\nUrban",
   "Percent\nIndia Pop.")
```

Next we modify grapics functions used in splom(). For example we skip the hexagon binning function since the number points plotted in each scatterplot is small.

```
# Modify functions for use in splom()

offDiag <- function(x,y,...){
  panel.grid(h = -1,v = -1,...)
#  panel.hexbinplot(x,y,xbins = 15,...,border=gray(.7),
#    trans=function(x)x^.5)
  panel.points(x,y,...,cex = .8, pch = 16, col = "black")
  panel.loess(x , y, ..., lwd = 3, col = 'red')
}

# We might change the density line color and
# width in the diagonal panel.

onDiag <- function(x, ...){
    yrng <- current.panel.limits()$ylim
    d <- density(x, na.rm = TRUE)
    d$y <- with(d, yrng[1] + 0.95 * diff(yrng) * y / max(y) )
    panel.lines(d,col = rgb(.83,.66,1),lwd = 3)
    diag.panel.splom(x, ...)
 }
```

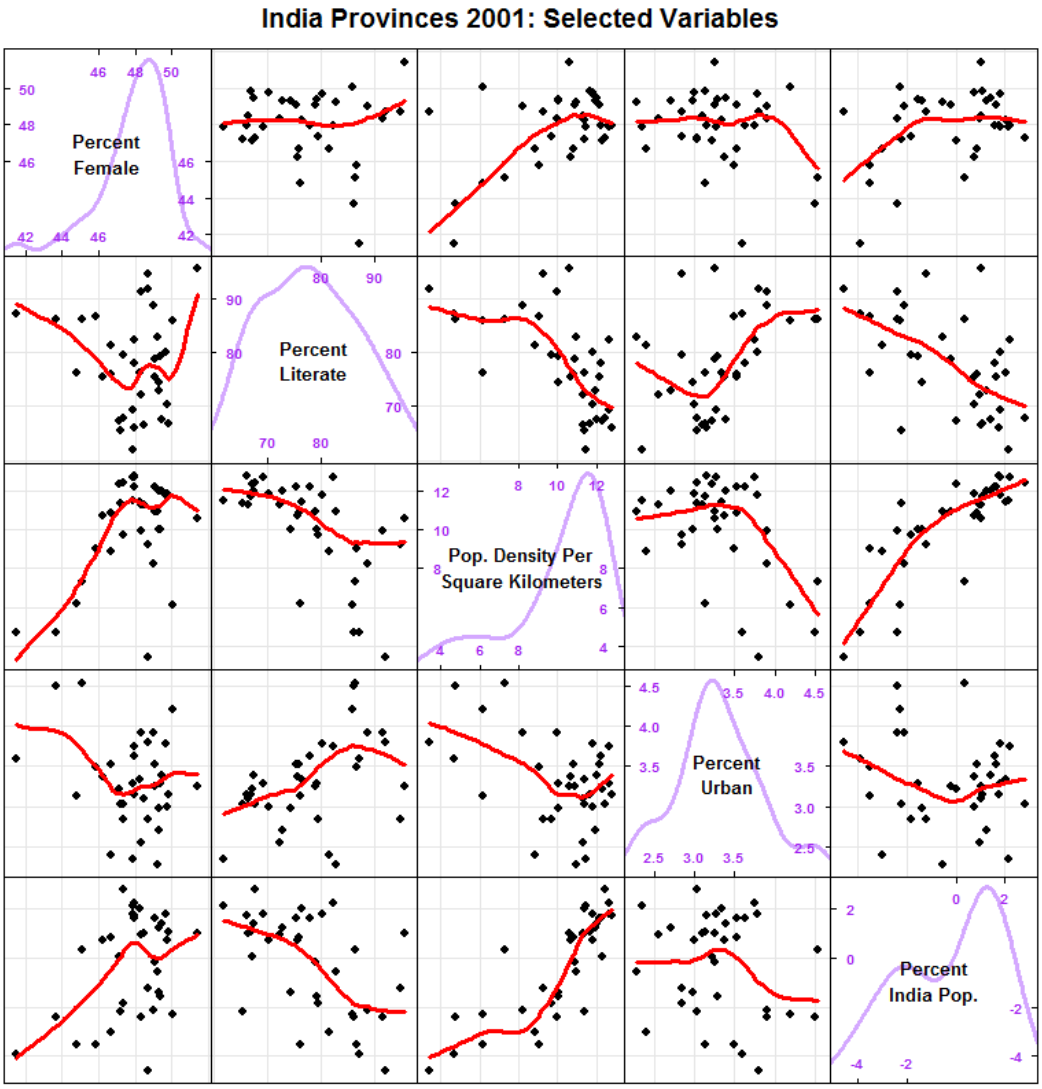
Finally we call the splom function and modify some arguments here as well.

Below we set the pscale argument, pscale = 4, to roughly specify the number of tick marks produced.

The other argument keywords are suggestive of what can be changed.

```
splom(indiaExp, as.matrix = TRUE,
  xlab = '',main = "India Provinces 2001: Selected Variables",
  varnames = varnames,
  pscale = 4, varname.cex = 0.8,  varname.font = 2,
  axis.text.cex = 0.6,
```

```
  axis.text.col = "purple",axis.text.font = 2,
  axis.line.tck = .5,
  panel = offDiag,
  diag.panel = onDiag
)
```



I was hoping to see a stronger pattern in the top row of panels.The smoothes suggest that the percent female increases a little with the percent literacy but there are points far below the curve. There is more of an increase with population density per square kilometers and with the percent of the India population. There is some decrease for the very high values of percent urban.

Perhaps more would be revealed by looking other variables such as religion (as suggest above), and poverty. We could fit some regression models but with variables at had but this isn't promising.

We have seen that summary statistics for US states do not reveal the variation of the summary statistics for state counties. A state mortality rate for a particular type of cancer may look typical compared to other states but one or more its counties may have really high rates that are of concern. Thee will also be counties very low rates that of of interest. Is the variation random or partially explained but other variables.

Similarly, looking at smaller adminstrative regions of India provides a better start toward understanding its incredible diversity. Below we shift attention to showing the data use perceptual grouped row labels.

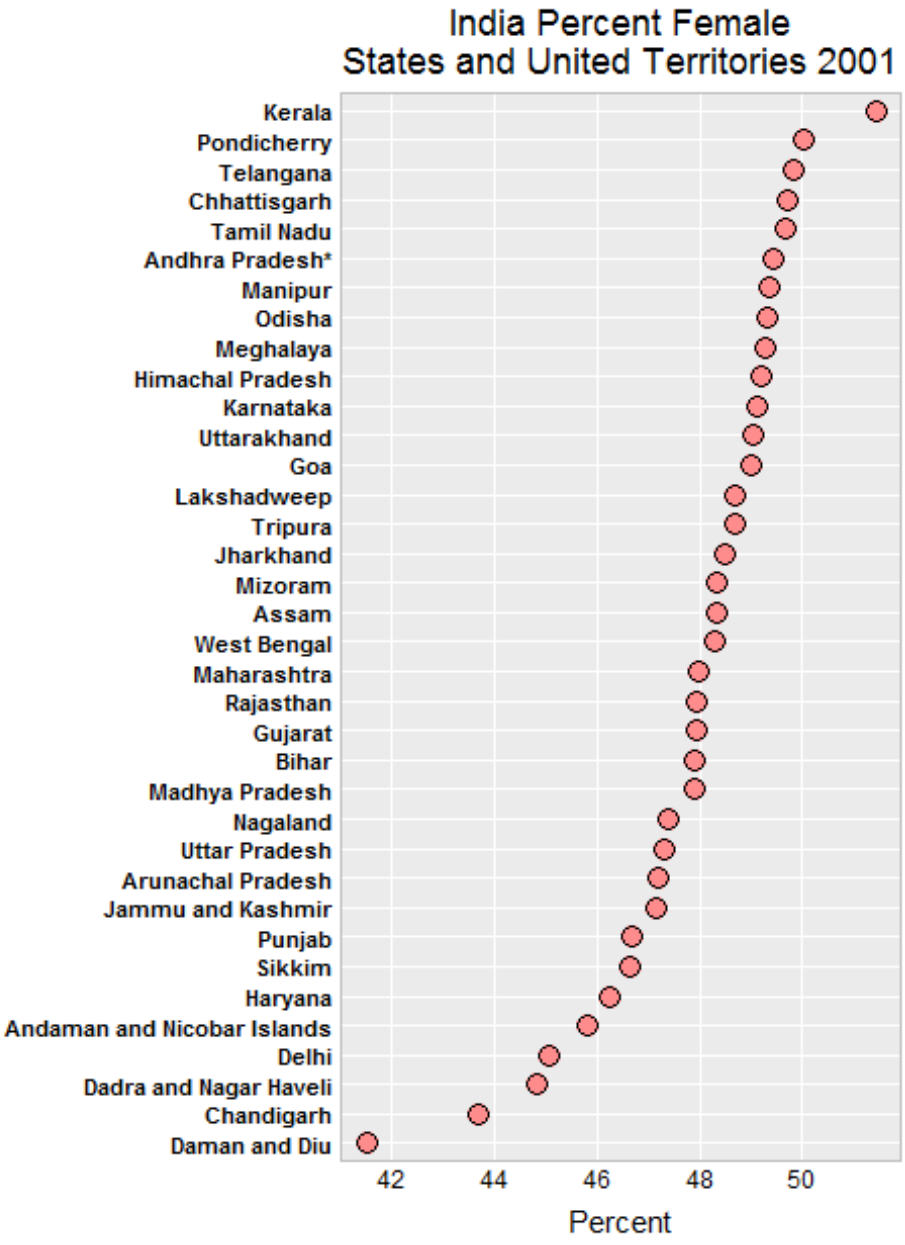
4. Produce a row labeled dot plot

First we select variables for use in the section.

```
india4 <- india2 %>% select(
   State_UT, FemalePct, Literacy, UrbanPct, Density)
```

Below we put the States and United Territories in percent female descending order from top to the bottom. Here we control y-axis order when we specify what controls the y axis in the aes function.

```
ggplot(india4,
  aes(x = FemalePct, y = reorder(State_UT,FemalePct),
    descend = TRUE)) +
geom_point(shape = 21, fill = rgb(1,.55,.55),
  col = "black",size = 3.5) +
labs(x = "Percent",y = '',
  title = paste0("India Percent Female\n",
  "States and United Territories 2001")) + hw +
  theme(axis.text.y = element_text(size = rel(.95),
    face = 'bold'))
```



Kerala, with a percent higher the 50% appears exceptional.

5. Produce a perceptually grouped row labeled dot plot with four variables

One preparation procedure is:

1) Sort the tibble rows so the region names are in the desired order.

2) Rebuild the region names factor Use the reverse of the sorted region names as factor levels.

3) Add a row grouping factor Add a row numbe within group factor

4) Use gather to stack the variable values and column in two long tibble column and replicate other other column values for each stack.

5.1 Preparation

```
#1 Sort rows
ord <- order(india4$FemalePct,decreasing = TRUE)
india4Sort <- india4[ord,]

#2 Rebuild factor
nam <- as.character(india4Sort$State_UT)  #
india4Sort$State_UT = factor(nam,levels = rev(nam))

#3 Add grouping and row factor columns
#  Here 9 groups of size 4 each
grpLevels <- paste0('G',1:9)
grpNams <- rep(grpLevels, each = 4)
grpFactor <- factor(grpNams,levels = grpLevels)

rowLevels = paste0(1:4)
rowNams <- rep(rowLevels,9)
rowFactor <- factor(rowNams,levels = rowLevels)

india4Sort$Grp = grpFactor
india4Sort$Row = rowFactor

# Look at the group factor
grpFactor
```

```
##  [1] G1 G1 G1 G1 G2 G2 G2 G2 G3 G3 G3 G3 G4 G4 G4 G4 G5 G5 G5 G5 G6 G6 G6
## [24] G6 G7 G7 G7 G7 G8 G8 G8 G8 G9 G9 G9 G9
## Levels: G1 G2 G3 G4 G5 G6 G7 G8 G9
```

```
# Use gather to gather the 4 variables with percents
# into one column and add a column with the
# variable names. Below label the columns
# Percents and varNam
# This include a column called varName for the
# column labels

india4Gath <- gather(india4Sort,
  value = Percents, key = varNames,
  FemalePct:Density,
  factor_key = TRUE)
head(india4Gath)
```

```
## # A tibble: 6 x 5
##   State_UT        Grp   Row   varNames  Percents
##   <fct>           <fct> <fct> <fct>        <dbl>
## 1 Kerala          G1    1     FemalePct     51.4
```

```
## 2 Pondicherry     G1    2     FemalePct     50.0
## 3 Telangana       G1    3     FemalePct     49.8
## 4 Chhattisgarh    G1    4     FemalePct     49.7
## 5 Tamil Nadu      G2    1     FemalePct     49.7
## 6 Andhra Pradesh* G2    2     FemalePct     49.4
```

5.2 Plot production

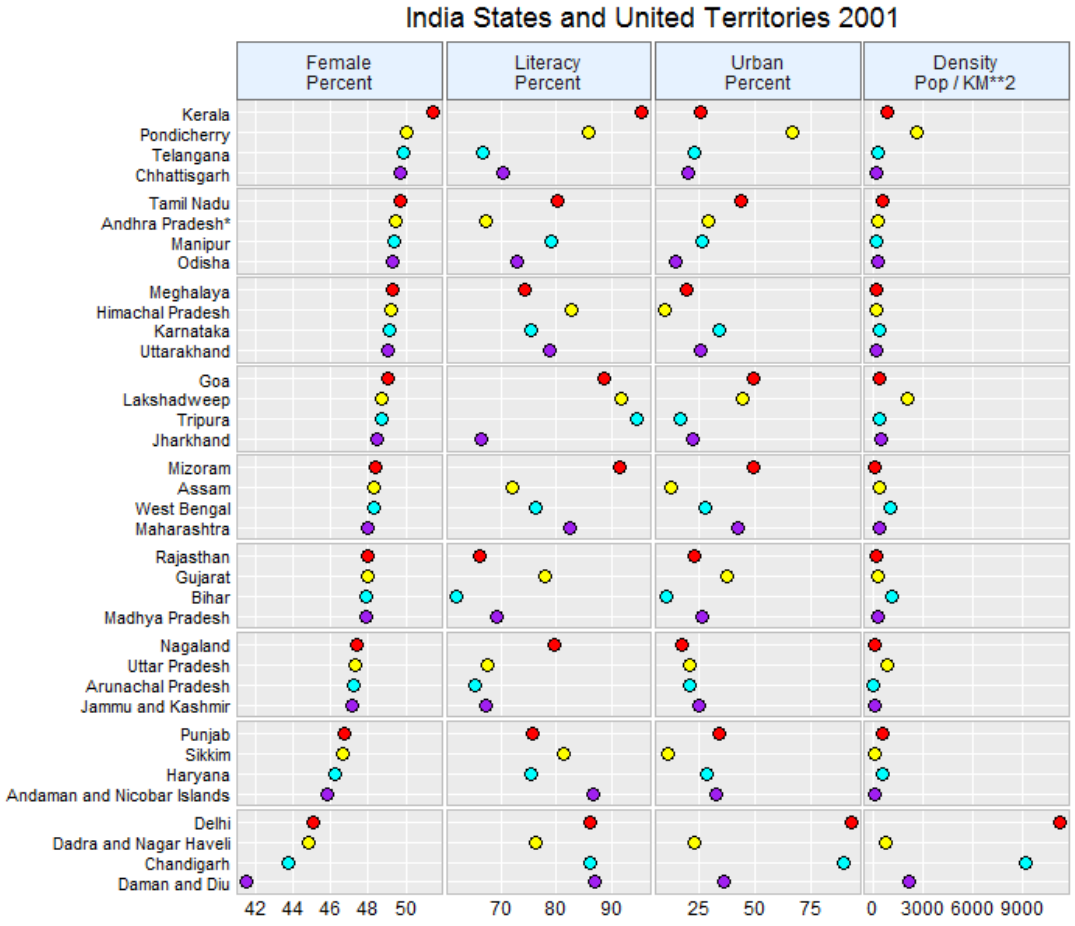
The facet_grid function below provides the crucial specification scale="free". This applies to both the x and y axis. All of the x-axis scales are different. All of the y-axis scale are different sinc each has 4 different provinces or territories.

This example also shows creating and using strip labels for the column facets. The labeller function can do more.

```
stripLabs <- c(
  FemalePct = "Female\nPercent",
  Literacy = "Literacy\nPercent",
  UrbanPct = "Urban\nPercent",
  Density = "Density\nPop / KM**2")

dotfill = c('red','yellow','cyan','purple')

ggplot(india4Gath,
  aes(x = Percents, y = State_UT, fill = Row)) +
geom_point(shape = 21, col = "black",size = 2.8) +
scale_fill_manual(values = dotfill, guide = FALSE) +
facet_grid(Grp~varNames, scale = "free",
  labeller = labeller(varNames = stripLabs)) +
labs(x = "",y = "", title =
  "India States and United Territories 2001") + hw
```



5.3 Comments on color fill, grid lines and legend variations

There are reasonable variations of the plot. The class color guidelines are that fill colors should be distinct and have familiar color names. This still leaves many good choices for fill colors.

With the color links we might try dropping the horizontal grid lines. This could simplify appearance a little.

With groups of four rows or less, finding the points in a the same row of the group is pretty easy without the color links. Using one fill color may suffice. This may provide a simpler appearance and avoid raising a question about what the fill colors represent. The currently suppressed legend could be included for audiences that are unfamilar with color linking across columns of panels.