Zonal Statistics for DMSP OLS Datasets

Zonal statistics refers to the calculation of statistics on values of a raster within the zones of another dataset.

This project is a collaborative work of UXO India and IDFC.

In the following example the zonal statistics i.e., sum, mean, median, mode, minimum value, maximum value, standard deviation for DMSP OLS Original, Deblurred and Radiance datasets are calculated.

Including Libraries

```
library(rgdal) # To import raster data

library(maptools) # To plot the data

library(proj4)  # To reproject raster

library(xtable) # To export data to html tables

library (raster) # Required for rgdal

library (rgeos) # Required for maptools

library (spatstat) # Analysing spatial point patterns

library (tiff) # Read TIFF images and required for rgdal

library (sp) #Required for maptools

library (data.table) # Modifying columns

library (modeest) #To calculate mode value for the zone

library (foreign) # Required for maptools
```

Define memory size

```
memory.size(100000)

## [1] 1e+05
```

Set working directory i.e giving the path of input files

```
setwd("D:/IDFC work/Bulk Zonal Stat Calculation/INPUT/R_Script_Directory") # To set directory
```

Non Radiance Zonal Stat Calculation Function

```
Zonal_Stat_NR <- function(x,y) # Define function
{
  A<-extract(x,y) # Extract raster data zone-wise
  R<-array(0,dim=c(length(A),48)) # Create an empty array
  for (i in 1:length(A))  # Create for loop
  {
    temp=A[[i]]  #Get temperary memory
    NRLumin1<-names(NRLumin)
    NRLumin2<- gsub(".tif","",NRLumin1)#gsub() function replaces all matches of a string,
    B<-mlv(temp, method = "mfv") # Find mode of that zone
    R[i,1]="https://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html#AVSLCFC"
    R[i,2]=NRLumin2     # To mention Luminosity Data_Source & year
    SR1<-length(temp[temp>0])
    SR2<-length(temp)  # Get the length of temp
    SR3<-SR1/SR2
    SR<-paste(round(SR3*100,digits=2),"%",sep="")#Rounds the values in its first argument to the specified number of decimal places
    R[i,3]= SR2 # Find count for that zone
    R[i,4]= SR1 # Find Lit up_Pixel count for that zone
    R[i,5]= SR # Percentage area cover by light
    R[i,6]= mean(temp) # Find mean of that zone
    R[i,7]= min(temp) # Find minimum of that zone
    R[i,8]= max(temp) # Find maximum of that zone
    R[i,9]= median(temp) # Find median of that zone
    R[i,10]= sd(temp) # Find Std_Dev of that zone
    R[i,11]= sum(temp) # Find sum of that zone
    R[i,12]=B$M # Add data from column M of data B to 12th column of R
    R[i,13:48]=""
    rm(temp) # Delete the data from temp
  }

  BQ=paste(substr(names(x),(stri_length(names(x))-4),(stri_length(names(x)))), y[[6]], sep="_")
  CQ=paste(substr(names(x),(stri_length(names(x))-3),(stri_length(names(x)))))
  colnames(R) <- c("NR_Link","NR_Data_Source","NR_Count","NR_Lit_up_Pixel_count","NR_Percentage_light_cover_Area","NR_Mean","NR_Min","NR_Max","NR_Median","NR_Std_Dev","NR_Sum","NR_Mode","NRD_Link","NRD_Data_Source","NRD_Count","NRD_Lit_up_Pixel_count","NRD_Percentage_light_cover_Area","NRD_Mean","NRD_Min","NRD_Max","NRD_Median","NRD_Std_Dev","NRD_Sum","NRD_Mode","Rad_Link","Rad_Data_Source","Rad_Count","Rad_Lit_up_Pixel_count","Rad_Percentage_light_cover_Area","Rad_Mean","Rad_Min","Rad_Max","Rad_Median","Rad_Std_Dev","Rad_Sum","Rad_Mode","VIIRS_Link","VIIRS_Data_Source","VIIRS_Count","VIIRS_Lit_up_Pixel_count","VIIRS_Percentage_light_cover_Area","VIIRS_Mean","VIIRS_Min","VIIRS_Max","VIIRS_Median","VIIRS_Std_Dev","VIIRS_Sum","VIIRS_Mode") # Change column header
  z<-cbind(y[[1]],y[[2]],y[[3]],y[[4]],y[[5]],y[[6]],CQ,R,BQ) # Bind data with shape file
  return(z)
}
```

Non Radiance Deblurr Zonal Statistics Calculation Function

```
Zonal_Stat_NRD <- function(xA,yA) # Define function
{
  AA<-extract(xA,yA) # Extract raster data zone-wise

  RA<-array(0,dim=c(length(AA),48)) # Create an empty array
  for (j in 1:length(AA))  # Create for loop
  {
    temp=AA[[j]]  #Get temperary memory
    NRDLumin1<-names(NRDLumin)
    NRDLumin2<- gsub(".tif","",NRDLumin1)
    BA<-mlv(temp, method = "mfv") # Find mode of that zone
    RA[j,1:12]=""
    RA[j,13]="https://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html#AXP"
    RA[j,14]=NRDLumin2     # To mention Luminosity Data_Source & year
    SA1<-length(temp[temp>0])
    SA2<-length(temp)
    SA3<-SA1/SA2
    SA<-paste(round(SA3*100,digits=2),"%",sep="")
    RA[j,15]= SA2 # Find count for that zone
    RA[j,16]= SA1 # Find Lit up_Pixel count for that zone
    RA[j,17]= SA # Percentage area cover by light
    RA[j,18]=mean(temp) # Find mean of that zone
    RA[j,19]=min(temp) # Find minimum of that zone
    RA[j,20]=max(temp) # Find maximum of that zone
    RA[j,21]=median(temp) # Find median of that zone
    RA[j,22]=sd(temp) # Find Std_Dev of that zone
    RA[j,23]=sum(temp) # Find sum of that zone
    RA[j,24]=BA$M # Find mode of that zone
    RA[j,25:48]=""
    rm(temp)
  }
  BQ=paste(substr(names(xA),(stri_length(names(xA))-4),(stri_length(names(xA)))), yA[[6]], sep="_")
  CQ=paste(substr(names(xA),(stri_length(names(xA))-3),(stri_length(names(xA)))))
  colnames(RA) <- c("NR_Link","NR_Data_Source","NR_Count","NR_Lit_up_Pixel_count","NR_Percentage_light_cover_Area","NR_Mean","NR_Min","NR_Max","NR_Median","NR_Std_Dev","NR_Sum","NR_Mode","NRD_Link","NRD_Data_Source","NRD_Count","NRD_Lit_up_Pixel_count","NRD_Percentage_light_cover_Area","NRD_Mean","NRD_Min","NRD_Max","NRD_Median","NRD_Std_Dev","NRD_Sum","NRD_Mode","Rad_Link","Rad_Data_Source","Rad_Count","Rad_Lit_up_Pixel_count","Rad_Percentage_light_cover_Area","Rad_Mean","Rad_Min","Rad_Max","Rad_Median","Rad_Std_Dev","Rad_Sum","Rad_Mode","VIIRS_Link","VIIRS_Data_Source","VIIRS_Count","VIIRS_Lit_up_Pixel_count","VIIRS_Percentage_light_cover_Area","VIIRS_Mean","VIIRS_Min","VIIRS_Max","VIIRS_Median","VIIRS_Std_Dev","V
IIRS_Sum","VIIRS_Mode") # Change column header
  zA<-cbind(yA[[1]],yA[[2]],yA[[3]],yA[[4]],yA[[5]],yA[[6]],CQ,RA,BQ) # Bind data with shape file
  return(zA)
}
```

Radiance Zonal Statistics Calculation Function

```
Zonal_Stat_Rad <- function(xR,yR) # Define function
{
  AAA<-extract(xR,yR) # Extract raster data zone-wise

  RR<-array(0,dim=c(length(AAA),48)) # Create an empty array
  for (k in 1:length(AAA))  # Create for loop
  {
    temp=AAA[[k]]  #Get temperary memory
    RLumin1<-names(RLumin)
    RLumin2<- gsub(".tif","",RLumin1)
    BR<-mlv(temp, method = "mfv") # Find mode of that zone
    RR[k,1:24]=""
    RR[k,25]="https://ngdc.noaa.gov/eog/dmsp/download_radcal.html"
    RR[k,26]=RLumin2      # To mention Luminosity Data_Source & year
    SRR1<-length(temp[temp>0])
    SRR2<-length(temp)  # Get the size of temp
    SRR3<-SRR1/SRR2
    SRR<-paste(round(SRR3*100,digits=2),"%",sep="") # Round off the value
    RR[k,27]= SRR2 # Find count for that zone
    RR[k,28]= SRR1 # Find Lit up Pixel count for that zone
    RR[k,29]= SRR # Percentage area cover by light
    RR[k,30]=mean(temp) # Find mean of that zone
    RR[k,31]=min(temp) # Find minimum of that zone
    RR[k,32]=max(temp) # Find maximum of that zone
    RR[k,33]=median(temp) # Find median of that zone
    RR[k,34]=sd(temp) # Find Std_Dev of that zone
    RR[k,35]=sum(temp) # Find sum of that zone
    RR[k,36]=BR$M# Find mode of that zone
    RR[k,37:48]=""
    rm(temp)
  }

  BQ=paste(substr(names(xR),(stri_length(names(xR))-4),(stri_length(names(xR)))), yR[[6]], sep="_")
  CQ=paste(substr(names(xR),(stri_length(names(xR))-3),(stri_length(names(xR)))))
  colnames(RR) <- c("NR_Link","NR_Data_Source","NR_Count","NR_Lit_up_Pixel_count","NR_Percentage_light_cover_Area","NR_Mean","NR_Min","NR_Max","NR_Median","NR_Std_Dev","NR_Sum","NR_Mode","NRD_Link","NRD_Data_Source","NRD_Count","NRD_Lit_u
p_Pixel_count","NRD_Percentage_light_cover_Area","NRD_Mean","NRD_Min","NRD_Max",
"NRD_Median","NRD_Std_Dev","NRD_Sum","NRD_Mode","Rad_Link","Rad_Data_Source",
"Rad_Count","Rad_Lit_up_Pixel_count","Rad_Percentage_light_cover_Area","Rad_Mean","Ra
d_Min","Rad_Max","Rad_Median","Rad_Std_Dev","Rad_Sum","Rad_Mode","VIIRS_Link","V
IIRS_Data_Source","VIIRS_Count","VIIRS_Lit_up_Pixel_count","VIIRS_Percentage_light_co
ver_Area","VIIRS_Mean","VIIRS_Min","VIIRS_Max","VIIRS_Median","VIIRS_Std_Dev","V
IIRS_Sum","VIIRS_Mode") # Change column header
  zR<-cbind(yR[[1]],yR[[2]],yR[[3]],yR[[4]],yR[[5]],yR[[6]],CQ,RR,BQ) # Bind data with shap
e file
  return(zR)
}
```

VIIRS Zonal Statistics Calculation Function

```
Zonal_Stat_VIIRS <- function(xB,yB) # Define function
{
  AB<-extract(xB,yB) # Extract raster data zone-wise

  RB<-array(0,dim=c(length(AB),48)) # Create an empty array
  for (l in 1:length(AB))  # Create for loop
  {
    temp=AB[[l]]  #Get temperary memory
    VLumin1<-names(VLumin)
    VLumin2<- gsub(".tif","",VLumin1)
    BB<-mlv(temp, method = "mfv") # Find mode of that zone
    RB[l,1:36]=""
    RB[l,37]= "https://ngdc.noaa.gov/eog/viirs/download_monthly.html"
    RB[l,38]=VLumin2 # To mention Luminosity Data_Source & year
    SRRR1<-length(temp[temp>0])
    SRRR2<-length(temp)
    SRRR3<-SRRR1/SRRR2
    SRRR<-paste(round(SRRR3*100,digits=2),"%",sep="")
    RB[l,39]= SRRR2 # Find count for that zone
    RB[l,40]= SRRR1 # Find Lit up_Pixel count for that zone
    RB[l,41]= SRRR # Percentage area cover by light
    RB[l,42]=mean(temp) # Find mean of that zone
    RB[l,43]=min(temp) # Find minimum of that zone
    RB[l,44]=max(temp) # Find maximum of that zone
    RB[l,45]=median(temp) # Find median of that zone
    RB[l,46]=sd(temp) # Find Std_Dev of that zone
    RB[l,47]=sum(temp) # Find sum of that zone
    RB[l,48]=BB$M      # Find mode of that zone
    rm(temp)
  }
  BQ=paste(substr(names(xB),(stri_length(names(xB))-4),(stri_length(names(xB)))), yB[[6]],
sep="_")
  CQ=paste(substr(names(xB),(stri_length(names(xB))-3),(stri_length(names(xB)))))
  colnames(RB) <- c("NR_Link","NR_Data_Source","NR_Count","NR_Lit_up_Pixel_count","
NR_Percentage_light_cover_Area","NR_Mean","NR_Min","NR_Max","NR_Median","NR_Std_Dev","NR_Sum","NR_Mode","NRD_Link","NRD_Data_Source","NRD_Count","NRD_Lit_up_Pixel_count","NRD_Percentage_light_cover_Area","NRD_Mean","NRD_Min","NRD_Max","NRD_Median","NRD_Std_Dev","NRD_Sum","NRD_Mode","Rad_Link","Rad_Data_Source","Rad_Count","Rad_Lit_up_Pixel_count","Rad_Percentage_light_cover_Area","Rad_Mean","Rad_Min","Rad_Max","Rad_Median","Rad_Std_Dev","Rad_Sum","Rad_Mode","VIIRS_Link","VIIRS_Data_Source","VIIRS_Count","VIIRS_Lit_up_Pixel_count","VIIRS_Percentage_light_cover_Area","VIIRS_Mean","VIIRS_Min","VIIRS_Max","VIIRS_Median","VIIRS_Std_Dev","VIIRS_Sum","VIIRS_Mode") # Change column header
  zB<-cbind(yB[[1]],yB[[2]],yB[[3]],yB[[4]],yB[[5]],yB[[6]],CQ,RB,BQ) # Bind data with shape file
  return(zB)
}
```

To Calculate Zonal Statistics for DMSP OLS Original Data

```
ptm <- proc.time()
NRfileR <- list.files(getwd(), pattern="NR.*.tif$", full.names=FALSE) # Read list of Raster
for(m in 1:length(NRfileR)) # Flow control for all the non radiance data
{
  NRLumin <-raster(NRfileR[m]) # Read the raster data
  x = NRLumin@crs # Read the projection of the raster
  Sfile<- list.files(getwd(),pattern=".*.shp$", full.name=FALSE) # Get the shapefiles list in the working directory
  for(q in 1:length(Sfile))  # Flow control for all the non radiance data
  {
    Zone <-shapefile(Sfile[q]) # Assign the shapefile to variable
    Zone <- spTransform(Zone,x) # Reproject the vector to the raster projection
    M<-Zonal_Stat_NR (NRLumin,Zone) #Call the function
    if (q<2) { temp_shape_NR<-M }   else    { temp_shape_NR<-rbind(temp_shape_NR,M) } # Bind the zonal statistics to temp_shape_NR
  }
  if (m<2) { temp_raster_NR<-temp_shape_NR }  else  { temp_raster_NR<-rbind(temp_raster_NR,temp_shape_NR) } # Bind the zonal statistics to temp_raster_NR
  rm(temp_shape_NR) # Removes the files in temp_shape_NR
}
proc.time() - ptm
```

```
##    user  system elapsed 
##  387.72    2.64  391.08
```

To Calculate Zonal Statistics for DMSP OLS Deblurred Data

```
ptm <- proc.time()
NRDfileR <- list.files(getwd(), pattern="DBR.*.tif$", full.names=FALSE) # Read list of Raster
for(n in 1:length(NRDfileR))                                    # for loop to read raster
{
  NRDLumin <-raster(NRDfileR[n])                              #Move raster to variable Lumin
  y = NRDLumin@crs
  Sfile<- list.files(getwd(),pattern=".*.shp$", full.name=FALSE)    #Read list of Zone
  for(r in 1:length(Sfile))
  {
    Zone <-shapefile(Sfile[r])
    Zone <- spTransform(Zone,y)
    N<-Zonal_Stat_NRD (NRDLumin,Zone)
    if (r<2) { temp_shape_NRD<-N }   else    { temp_shape_NRD<-rbind(temp_shape_NRD,N)
}
  }
  if (n<2) { temp_raster_NRD<-temp_shape_NRD }  else  { temp_raster_NRD<-rbind(temp_ras
ter_NRD,temp_shape_NRD) }
  rm(temp_shape_NRD)
}
proc.time() - ptm
```

```
##    user  system elapsed 
##  703.93  103.55  808.82
```

To Calculate Zonal Statistics for DMSP OLS Radiance Data

```
ptm <- proc.time()
RfileR <- list.files(getwd(), pattern="RAD.*.tif$", full.names=FALSE) # Read list of Raster
for(o in 1:length(RfileR))                                       # for loop to read raster
{
  RLumin <-raster(RfileR[o]) #Move raster to variable Lumin
  f = RLumin@crs  #Get the projection of raster
  Sfile<- list.files(getwd(),pattern=".*.shp$", full.name=FALSE)    #Read list of Zone
  for(s in 1:length(Sfile))  # Loop to read the files
  {
    Zone <-shapefile(Sfile[s]) #Assign the shapefile to a variable[zone]
    Zone <- spTransform(Zone,f) # Reproject the shapefile to raster projection system
    O<-Zonal_Stat_Rad (RLumin,Zone)  # Call the function
    if (s<2) { temp_shape_R<-O }   else    { temp_shape_R<-rbind(temp_shape_R,O) } #Write th
e zonal statistics output to temp_shape_V

  }
  if (o<2) { temp_raster_R<-temp_shape_R }  else  { temp_raster_R<-rbind(temp_raster_R,temp
_shape_R) } #Write the zonal statistics output to temp_raster_V
  rm(temp_shape_R)
}
proc.time() - ptm
```

```
##    user  system elapsed 
##   92.77    1.27   94.34
```

To Calculate Zonal Statistics for VIIRS Data

```
ptm <- proc.time()
VfileR <- list.files(getwd(), pattern="NPP.*.tif$", full.names=FALSE) # Read list of Raster
for(p in 1:length(VfileR))                                      # for loop to read raster
{
  VLumin <-raster(VfileR[p])                                      #Move raster to variable Lumin
  g = VLumin@crs
  Sfile<- list.files(getwd(),pattern=".*.shp$", full.name=FALSE)    #Read list of Zone
  for(t in 1:length(Sfile))  #Loop for reading the files
  {
    Zone <-shapefile(Sfile[t])  #Assign the shapefile to the variable
    Zone <- spTransform(Zone,g) #Reproject the vector to raster projection system
    P<-Zonal_Stat_VIIRS (VLumin,Zone) #Calling the function
    if (t<2) { temp_shape_V<-P }   else    { temp_shape_V<-rbind(temp_shape_V,P) } #Write the zonal statistics output to temp_shape_V
  }
  if (p<2) { temp_raster_V<-temp_shape_V }  else  { temp_raster_V<-rbind(temp_raster_V,temp_shape_V) } Write the zonal statistics output to temp_raster_V
  rm(temp_shape_V)
}
proc.time() - ptm
```

```
##    user  system elapsed
##   97.09    1.40   98.68
```

```
ptm <- proc.time()
```

```
NR<-temp_raster_NR # Assign the output to the variable
NRD<-temp_raster_NRD # Assign the output to the variable
Rad<-temp_raster_R # Assign the output to the variable
VRS<-temp_raster_V # Assign the output to the variable
```

```
AS<-merge(NR, NRD,by="BQ" , all = TRUE ) # Merge Non Rad & Non Rad Deblurr Data & save to variable "AS"
```

```
AS["NRD_Link.x"]<-AS["NRD_Link.y"] # Assign the NRD_Link.y as NRD_Link.x
AS["NRD_Data_Source.x"]<-AS["NRD_Data_Source.y"]
AS["NRD_Count.x"]<-AS["NRD_Count.y"]
AS["NRD_Lit_up_Pixel_count.x"]<-AS["NRD_Lit_up_Pixel_count.y"]
AS["NRD_Percentage_light_cover_Area.x"]<-AS["NRD_Percentage_light_cover_Area.y"]
AS["NRD_Mean.x"]<-AS["NRD_Mean.y"]
AS["NRD_Min.x"]<-AS["NRD_Min.y"]
AS["NRD_Max.x"]<-AS["NRD_Max.y"]
AS["NRD_Median.x"]<-AS["NRD_Median.y"]
```

```
AS["NRD_Std_Dev.x"]<-AS["NRD_Std_Dev.y"]
AS["NRD_Sum.x"]<-AS["NRD_Sum.y"]
AS["NRD_Mode.x"]<-AS["NRD_Mode.y"]
```

```
ASK<-AS[1:56] #Assign 1 to 56 columns of AS to ASK
```

```
CS<-merge(ASK, Rad,by="BQ" , all = TRUE )  # Merge "AS" variable with Radiance data
colnames(CS) <- c("BQ","Census_Code","State_Name","State_Census_Cd","District_Name","District_Census_ID","Area","Year","NR_Link","NR_Data_Source","NR_Count","NR_Lit_up_pixel_count","NR_Light_pixcel_percentage","NR_Mean","NR_Min","NR_Max","NR_Median","NR_Std_Dev","NR_Sum","NR_Mode","NRD_Link","NRD_Data_Source","NRD_Count","NRD_Lit_up_pixel_count","NRD_Light_pixcel_percentage","NRD_Mean","NRD_Min","NRD_Max","NRD_Median","NRD_Std_Dev","NRD_Sum","NRD_Mode","Rad_Link","Rad_Data_Source","Rad_Count","Rad_Lit_up_pixel_count","Rad_Light_pixcel_percentage","Rad_Mean","Rad_Min","Rad_Max","Rad_Median","Rad_Std_Dev","Rad_Sum","Rad_Mode","VIIRS_Link","VIIRS_Data_Source","VIIRS_Count","VIIRS_Lit_up_pixel_count","VIIRS_Light_pixcel_percentage","VIIRS_Mean","VIIRS_Min","VIIRS_Max","VIIRS_Median","VIIRS_Std_Dev","VIIRS_Sum","VIIRS_Mode","Census_Code1","State_Name1","State_Census_Cd1","District_Name1",
"District_Census_ID1","Area1","Year1","NR_Link1","NR_Data_Source1","NR_Count1","NR_Lit_up_pixel_count1","NR_Light_pixcel_percentage1","NR_Mean1","NR_Min1","NR_Max1","NR_Median1","NR_Std_Dev1","NR_Sum1","NR_Mode1","NRD_Link1","NRD_Data_Source1","NRD_Count1","NRD_Lit_up_pixel_count1","NRD_Light_pixcel_percentage1","NRD_Mean1","NRD_Min1","NRD_Max1","NRD_Median1","NRD_Std_Dev1","NRD_Sum1","NRD_Mode1","Rad_Link1","Rad_Data_Source1","Rad_Count1","Rad_Lit_up_pixel_count1","Rad_Light_pixcel_percentage1","Rad_Mean1","Rad_Min1","Rad_Max1","Rad_Median1","Rad_Std_Dev1","Rad_Sum1","Rad_Mode1","VIIRS_Link1","VIIRS_Data_Source1","VIIRS_Count1","VIIRS_Lit_up_pixel_count1","VIIRS_Light_pixcel_percentage1","VIIRS_Mean1","VIIRS_Min1","VIIRS_Max1","VIIRS_Median1","VIIRS_Std_Dev1","VIIRS_Sum1","VIIRS_Mode1") # Change column header
```

```
levels(CS$Year)<-unique(c(levels(CS$Year),levels(CS$Year1)))
levels(CS$Year)

##  [1] "1992" "1993" "1994" "1995" "1996" "1997" "1998" "1999" "2000" "2001"
## [11] "2002" "2003" "2004" "2005" "2006" "2007" "2008" "2009" "2010" "2011"
## [21] "2012" "2013"
```

```
for(d in 1:lengths(CS[1], use.names = FALSE)) # Run the loop from 1 to length(CS)
{
  if(is.na(CS[d,"Census_Code"])) # Enter the loop if the CS contained d or Census_Code
  {
    CS[d,"Census_Code"]<-CS[d,"Census_Code1"]  # Assign the Census_Code1 to Census_Code
    CS[d,"State_Name"]<-CS[d,"State_Name1"]
    CS[d,"State_Census_Cd"]<-CS[d,"State_Census_Cd1"]
    CS[d,"District_Name"]<-CS[d,"District_Name1"]
    CS[d,"District_Census_ID"]<-CS[d,"District_Census_ID1"]
    CS[d,"Area"]<-CS[d,"Area1"] ## Rename the codes
    CS[d,"Year"]<-CS[d,"Year1"] ## Rename the codes
  }
}
```

```
CS["Rad_Link"]<-CS["Rad_Link1"] # Assign the Rad_Link 1to Rad_Link
CS["Rad_Data_Source"]<-CS["Rad_Data_Source1"]
CS["Rad_Count"]<-CS["Rad_Count1"]
CS["Rad_Lit_up_pixel_count"]<-CS["Rad_Lit_up_pixel_count1"]
CS["Rad_Light_pixcel_percentage"]<-CS["Rad_Light_pixcel_percentage1"]
CS["Rad_Mean"]<-CS["Rad_Mean1"] ## Rename the codes
CS["Rad_Min"]<-CS["Rad_Min1"]
CS["Rad_Max"]<-CS["Rad_Max1"]
CS["Rad_Median"]<-CS["Rad_Median1"]
CS["Rad_Std_Dev"]<-CS["Rad_Std_Dev1"]
CS["Rad_Sum"]<-CS["Rad_Sum1"]
CS["Rad_Mode"]<-CS["Rad_Mode1"]
```

```
CSK<-CS[1:56] #Assign 1 to 56 columns of CS to CSK
```

```
ES<-merge(CSK,VRS,by="BQ" , all = TRUE )  # Merge "AS" variable with Radiance data
colnames(ES) <- c("BQ","Census_Code","State_Name","State_Census_Cd","District_Name","
District_Census_ID","Area","Year","NR_Link","NR_Data_Source","NR_Count","NR_Lit_up_p
ixel_count","NR_Light_pixcel_percentage","NR_Mean","NR_Min","NR_Max","NR_Median","
NR_Std_Dev","NR_Sum","NR_Mode","NRD_Link","NRD_Data_Source","NRD_Count","NR
D_Lit_up_pixel_count","NRD_Light_pixcel_percentage","NRD_Mean","NRD_Min","NRD_M
ax","NRD_Median","NRD_Std_Dev","NRD_Sum","NRD_Mode","Rad_Link","Rad_Data_Sour
ce","Rad_Count","Rad_Lit_up_pixel_count","Rad_Light_pixcel_percentage","Rad_Mean","Rad
_Min","Rad_Max","Rad_Median","Rad_Std_Dev","Rad_Sum","Rad_Mode","VIIRS_Link","VI
IRS_Data_Source","VIIRS_Count","VIIRS_Lit_up_pixel_count","VIIRS_Light_pixcel_percent
age","VIIRS_Mean","VIIRS_Min","VIIRS_Max","VIIRS_Median","VIIRS_Std_Dev","VIIRS_
Sum","VIIRS_Mode","Census_Code1","State_Name1","State_Census_Cd1","District_Name1",
"District_Census_ID1","Area1","Year1","NR_Link1","NR_Data_Source1","NR_Count1","NR_
Lit_up_pixel_count1","NR_Light_pixcel_percentage1","NR_Mean1","NR_Min1","NR_Max1","
NR_Median1","NR_Std_Dev1","NR_Sum1","NR_Mode1","NRD_Link1","NRD_Data_Source1
","NRD_Count1","NRD_Lit_up_pixel_count1","NRD_Light_pixcel_percentage1","NRD_Mean
1","NRD_Min1","NRD_Max1","NRD_Median1","NRD_Std_Dev1","NRD_Sum1","NRD_Mod
e1","Rad_Link1","Rad_Data_Source1","Rad_Count1","Rad_Lit_up_pixel_count1","Rad_Light_
pixcel_percentage1","Rad_Mean1","Rad_Min1","Rad_Max1","Rad_Median1","Rad_Std_Dev1"
,"Rad_Sum1","Rad_Mode1","VIIRS_Link1","VIIRS_Data_Source1","VIIRS_Count1","VIIRS_
Lit_up_pixel_count1","VIIRS_Light_pixcel_percentage1","VIIRS_Mean1","VIIRS_Min1","VII
RS_Max1","VIIRS_Median1","VIIRS_Std_Dev1","VIIRS_Sum1","VIIRS_Mode1") # Change
column header
```

```
levels(ES$Year)<-unique(c(levels(ES$Year),levels(ES$Year1)))
levels(ES$Year)
```

```
##  [1] "1992" "1993" "1994" "1995" "1996" "1997" "1998" "1999" "2000" "2001"
## [11] "2002" "2003" "2004" "2005" "2006" "2007" "2008" "2009" "2010" "2011"
## [21] "2012" "2013" "2014" "2015" "2016"
```

```
for(z in 1:lengths(ES[1], use.names = FALSE))  # Run the loop from 1 to length(ES)
{
  if(is.na(ES[z,"Census_Code"]))  # Enter the loop if the ES contained z or Census_Code
  {
    ES[z,"Census_Code"]<-ES[z,"Census_Code1"] # Assign the Census_Code1 to Census_Code
    ES[z,"State_Name"]<-ES[z,"State_Name1"]
    ES[z,"State_Census_Cd"]<-ES[z,"State_Census_Cd1"]
    ES[z,"District_Name"]<-ES[z,"District_Name1"]
    ES[z,"District_Census_ID"]<-ES[z,"District_Census_ID1"]
    ES[z,"Area"]<-ES[z,"Area1"]
    ES[z,"Year"]<-ES[z,"Year1"]
  }
}
```

```
ES["VIIRS_Link"]<-ES["VIIRS_Link1"] # Assign the VIIRS_Link1 to VIIRS_Link
ES["VIIRS_Data_Source"]<-ES["VIIRS_Data_Source1"]
ES["VIIRS_Count"]<-ES["VIIRS_Count1"]
ES["VIIRS_Lit_up_pixel_count"]<-ES["VIIRS_Lit_up_pixel_count1"]
ES["VIIRS_Light_pixcel_percentage"]<-ES["VIIRS_Light_pixcel_percentage1"]
ES["VIIRS_Mean"]<-ES["VIIRS_Mean1"]
ES["VIIRS_Min"]<-ES["VIIRS_Min1"]
ES["VIIRS_Max"]<-ES["VIIRS_Max1"]
ES["VIIRS_Median"]<-ES["VIIRS_Median1"]
ES["VIIRS_Std_Dev"]<-ES["VIIRS_Std_Dev1"]
ES["VIIRS_Sum"]<-ES["VIIRS_Sum1"]
ES["VIIRS_Mode"]<-ES["VIIRS_Mode1"]
```

```
ESK<-ES[1:56] # Assign 1 to 56 columns of ES to ESK

colnames(ESK) <- c("System_Gen_UID","Census_Code","State_Name","State_Census_Cd","D
istrict_Name","District_Census_ID","Area","Year","NR_Link","NR_Data_Source","NR_Count"
,"NR_Lit_up_pixel_count","NR_Light_pixcel_percentage","NR_Mean","NR_Min","NR_Max",
"NR_Median","NR_Std_Dev","NR_Sum","NR_Mode","NRD_Link","NRD_Data_Source","NR
D_Count","NRD_Lit_up_pixel_count","NRD_Light_pixcel_percentage","NRD_Mean","NRD_
Min","NRD_Max","NRD_Median","NRD_Std_Dev","NRD_Sum","NRD_Mode","Rad_Link","Rad_Data_Source","Rad_Count","Rad_Lit_up_pixel_count","Rad_Light_pixcel_percentage","Rad_Mean","Rad_Min","Rad_Max","Rad_Median","Rad_Std_Dev","Rad_Sum","Rad_Mode","VIIRS_Link","VIIRS_Data_Source","VIIRS_Count","VIIRS_Lit_up_pixel_count","VIIRS_Light_pixcel_percentage","VIIRS_Mean","VIIRS_Min","VIIRS_Max","VIIRS_Median","VIIRS_Std_Dev","VIIRS_Sum","VIIRS_Mode") # Change column names
```

Write output to the csv file

```
write.csv(ESK,"D:/IDFC work/Bulk Zonal Stat Calculation/INPUT/R_Script_Directory/29_admin_zs.csv", na="NA") # Enter Output csv file name and path
```