

## Chapter 10

# Miscellaneous Code

### Contents

---

10.1 Remove or search for duplicated GPS locations in a data frame . . . . .	172
10.2 Need to convert back to a matrix to be able to export the data or manipulate the data . . . . .	173
10.3 Remove quotations marks around values in results table or printout . .	173
10.4 Bin numeric variables into categories . . . . .	173
10.5 Recode variables in Rcmdr . . . . .	173
10.6 Jitter UTM coordinates before making SpatialPointsDataFrame . . . . .	173
10.7 Remove extraneous locations or remove all data for a single animal by animal ID . . . . .	174
10.8 Generate sequential numbers as ID's for each location then add back to original dataset . . . . .	174
10.9 Rename data by deleting a portion of the string . . . . .	174
10.10 Grab text included in strings of varying lengths . . . . .	174
10.11 Rename levels of factor . . . . .	175
10.12 Recode numeric values as factors into categories . . . . .	175
10.13 Add leading zeros to single digit numbers to match datasets . . . . .	175
10.14 Force a DBRB class output to a data frame . . . . .	176
10.15 Subset GPS locations by a date range . . . . .	176
10.16 Drivers for rdgal input/ouput but run command for complete list if needed . . . . .	176

---

### 10.1 Remove or search for duplicated GPS locations in a data frame

```
#Removes duplicate entries
newgps <- newgps[!duplicated(newgps$DT),]

#Use code to look for NAs which are very bad and can cause code failure
#merge$DT

#Check for duplicates on a variety of data types

duplicated(HexPols2@polygons)#No duplicates here!
duplicated(deer.spdf@data)#No duplicates here!
duplicated(o2)
```

## 10.2 Need to convert back to a matrix to be able to export the data or manipulate the data

```
#Convert matrix from data.frame to export into csv file
mean <- as.matrix(summary$table)
#Write.table gives csv output of Summary. Be sure to specify the directory and
#the output files will be stored there

write.table(mean, file = "Distance.csv", sep = ",", row.names = TRUE, col.names = TRUE,
            qmethod = "double")
```

## 10.3 Remove quotations marks around values in results table or printout

```
m1 <- noquote(m1)
```

## 10.4 Bin numeric variables into categories

```
library(Rcmdr)
BinAlt <- bin.var(pelican$Altitude, bins=10, method='intervals',
                labels=c('1','2','3','4','5','6','7','8','9','10'))
```

## 10.5 Recode variables in Rcmdr

```
NA = "NA"
0 = "0"
1:200 = "1-200"
201:400 = "201-400"
401:600 = "401-600"
601:800 = "601-800"
801:1000 = "801-1000"
1001:1200 = "1001-1200"
1201:1400 = "1201-1400"
1401:1600 = "1401-1600"
1601:1800 = "1601-1800"
1801:2040 = "1801-2040"
```

## 10.6 Jitter UTM coordinates before making SpatialPointsDataFrame

```
#Jitter x coordinate before making dataframe
muleys$Xj <- jitter(muleys$X, factor=50, amount=NULL)
muleys$Yj <- jitter(muleys$Y, factor=50, amount=NULL)
coords2<-data.frame(x = muleys$Xj, y = muleys$Yj)
crs<-"+proj=utm +zone=12 +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"
jitter.spdf <- SpatialPointsDataFrame(coords= coords2, data = muleys,
```

```

proj4string = CRS(crs))
proj4string(jitter.spdf)
points(jitter.spdf, col="red")

```

## 10.7 Remove extraneous locations or remove all data for a single animal by animal ID

```

#Import original dataset
muleys <-read.csv("DCmuleysedited.csv", header=T)
str(muleys)
#Remove outlier locations and Mule deer D12 with too few locations
muleys <-subset(muleys, muleys$X > 599000 & muleys$X < 705000 & muleys$Y > 4167000
  & muleys$id != "D12")
muleys$id <- factor(muleys$id)#This step must be done to completely remove D12
summary(muleys$id)

```

## 10.8 Generate sequential numbers as ID's for each location then add back to original dataset

```

seqIDs <-c(1:nrow(muleys))
muleys <- cbind(muleys,seqIDs)

```

## 10.9 Rename data by deleting a portion of the string

```

#Remove text using substring function
# (i.e., "2004_Resident_1315_Adult" to "Resident_1315_Adult")
awp$code <- substr(awp$res_mig1, 6, 24)

#Concatenate above with "Season" column to make unique categories
awp$season_code <- paste(awp$Season, awp$code, sep="_")

#Results in "Summer_Resident_1315_Adult"

```

## 10.10 Grab text included in strings of varying lengths

```

#This grabs the last character of a string that has different lengths.
#For example, a vector of animal ID along with sex
#Code courtesy of D. Diefenbach, PA Coop Unit

#First read in a table with string as a factor
locweather <-read.table(file="locweather.txt",header=T, stringsAsFactors=FALSE)

```

```

#Create new variable by pasting last digit of variable regardless of variable length
#(i.e., number of characters)
sex <- as.factor(substr(locweather$indlocalident,nchar(locweather$indlocalident),
nchar(locweather$indlocalident)))

#This will change a vector from this:
#3455F
#21M
#44890F

#Returning a vector 'sex'
#F
#M
#F

```

## 10.11 Rename levels of factor

```

#Original age classes
# 2-5      3      6+      Adult      Calf Yearling
# 188      1      79      5      36      30

#Recode variables within a column that are factors such as combining age classes
#Combine ages classes
df$NewAge <- df$Age
levels(df$NewAge)<-list(Yearling=c("Calf","Yearling"),Adult=c("2-5","3","6+","Adult"))

summary(df$NewAge)
#Yearling Adult
# 66      273

```

## 10.12 Recode numeric values as factors into categories

```

#Originally we have multiple numbers of GPS locations that we
#want to recode into categories 1-4 or some other descriptor
#We will give NBLOCS a new name (LocsCat) in our dataframe, make it a
#factor (as.factor) then include breaks that we want to represent each
#category. For example, 0:100 represents category 1 with locations from
#0 to 100 m

merge$LocsCat <- as.factor(recode(merge$NBLOCS, "0:100='1';101:500='2';
501:1000='3'; 1001:10000='4'"))

```

## 10.13 Add leading zeros to single digit numbers to match datasets

```

#Sometimes we may want to "join" data and one may have single digits
#and the other may have leading zeros. We can use the "stringr" package
#to add leading zeros to only the numbers that occur in single digits

```

```
Label WMU COUNTY
14_D320 4D      7      #note COUNTY is an integer here
```

```
library(stingr)
none$COUNTY <- str_pad(none$COUNTY,2, pad = "0")
```

```
#Now the resulting data
Label WMU COUNTY
14_D320 4D      07
```

## 10.14 Force a DBRB class output to a data frame

```
#Allows us to write diffusion coefficients from movement-based
#home range output to a more useable form
dafr <- do.call(rbind.data.frame, vv)
write.table(dafr, "DiffCoeff2.txt", sep="\t", append=TRUE, col.names=F)
```

## 10.15 Subset GPS locations by a date range

```
#First make original date field (GPS.Fix.Time) a Date
muleys$Date <- as.Date(muleys$GPS.Fix.Time, "%Y.%m.%d")
#NOTE: The date in GPS.Fix.Time is formatted 2011.12.31 so formats
#must match the date format in line of code above.
```

```
locs2012 <- subset(muleys, Date > "2011-12-31" & Date < "2012-12-31")
```

## 10.16 Drivers for rdgal input/ouput but run command for complete list if needed

```
>getGDALDriverNames()
      name                               long_name create copy
AAIGrid      Arc/Info ASCII Grid  FALSE  TRUE
AIG          Arc/Info Binary Grid  FALSE FALSE
DOQ1        USGS DOQ (Old Style)  FALSE FALSE
DOQ2        USGS DOQ (New Style)  FALSE FALSE
E00GRID      Arc/Info Export E00 GRID  FALSE FALSE
HFA         Erdas Imagine Images (.img)  TRUE  TRUE
NITF        National Imagery Transmission Format  TRUE  TRUE
USGSDEM     USGS Optional ASCII DEM (and CDED)  FALSE  TRUE
```