Optimizing Tower Plot number in AK with NIWO HBP data

Courtney Meier

19 April 2016

Load parsed L0 data from Viewer

Data were parsed using the DPS Shiny App (from Claire L.) https://cklunch.shinyapps.io/parser_app/

Need to fix some residual errors in parsed data, as following variables did not get populated by parser:

- domainID
- siteID
- plotID
- subplotID <- delete this field, not needed for analysis

In addition:

- Date field requires simplifying to take data left of 'T' character
- Filter out records with `qaDryMass=Y`

```
# Read in output file from DPS parser
parsed.df <- read.csv("NIWO_hbpL0ParsedData_2015_v1.csv", header=T, stringsAsFactors = F)

# Add domainID, siteID, and plotID; remove subplotID
parsed.df$domainID <- "D13"
parsed.df$siteID <- "NIWO"
parsed.df$plotID <- str_sub(parsed.df$clipID, 1, 8)
parsed.df <- select(parsed.df, -subplotID, -remarks)

# Simplify `date` field
##  Creates a list with one list item per 'date', each list item has two elements one for data left of
date <- str_split(parsed.df$date, "T")
# Here, the x[1] specifies we keep the first column of the dataframe created by ldply; to keep all colu
date <- plyr::ldply(date, function(x) x[1])
# The df created by ldply has column names V1, V2, etc., and don't want that name, so need to rename
parsed.df$date <- date$V1

# Filter out values with `qaDryMass=Y`
parsed.df <- filter(parsed.df, qaDryMass=="N")

# Write out filtered, cleaned parsed.df to a .csv file
write.csv(parsed.df, file="NIWO_hbpL0ParsedData_2015_v2.csv", row.names=F)
```

Check for NAs

Identify records with plotID, clipID, or date that have `NA` values for any of these three fields

```
# The dplyr::tbl_df function makes display of the data frame in the console much more reasonable.
niwoHbp <- tbl_df(read.csv("NIWO_hbpL0ParsedData_2015_v2.csv", header=T, stringsAsFactors = F))

# Identify incomplete records in the dataframe
which(is.na(niwoHbp$plotID))
```

```
## [1]  78 116 124
```

```
# -> rows 78, 116, 124

which(is.na(niwoHbp$clipID))
```

```
## [1]  78 116 124
```

```
# -> rows 78, 116, 124

which(is.na(niwoHbp$date))
```

```
## integer(0)
```

```
# -> All are complete

# Next, need to fix the 'NA' values identified
niwoHbp$plotID[78] <- "NIWO_054"
niwoHbp$clipID[78] <- "NIWO_054_0186"

niwoHbp$plotID[116] <- "NIWO_046"
niwoHbp$clipID[116] <- "NIWO_046_0091"

niwoHbp$plotID[124] <- "NIWO_046"
niwoHbp$clipID[124] <- "NIWO_046_0091"
```

Calculate `totalDryMass` for Tower Plots only

Simulated sample sizes will require total `dryMass` values for a given `clipID`. For NIWO, there are no exclosures, and there should only be one clipID per plot. However, need to verify this assumption.

```
# Group the data, calculate the sum of dryMass values and then store as a new output
niwoHbp %>%
  group_by(plotID, date, clipID) %>%
  summarise(totalDryMass = sum(dryMass, na.rm=TRUE)) %>%
  arrange(plotID) -> niwoTot

# Check for duplicates in plotID and clipID
niwoTot %>%
  group_by(plotID, clipID) %>%
  summarize(n=n()) -> dupNiwo

# Join data in niwoTot with applicableModules.csv based on plotID and plotType - want to determine whic
setwd("~/Documents/neonScienceDocs/gitRepositories/devTOS/spatialData/supportingDocs")
```

```r
am <- read.csv("applicableModules.csv", header=T, stringsAsFactors = F)
am %>% filter(siteID=="NIWO", subtype=="basePlot") %>%
  select(plotID, plotType, subtype) -> amNiwo

# Use a left join to keep all rows of niwoTot; plotIDs in niwoTot with no match in amNiwo will be retur
niwoTot <- left_join(niwoTot, amNiwo, by="plotID")
niwoTot %>% filter(plotType=="tower") %>%
  select(-subtype) -> niwoTot

# Verify number of records is not > 30 for Tower Plots
nrow(niwoTot)   # -> 29; good
```

```
## [1] 29
```

```r
# Write out as .csv
write.csv(niwoTot, file="hbp_niwoTotDryMass_2015.csv", row.names = F)
```

Based on results above, `totalDryMass` data from Tower Plots are ready to use for the sampling simulation as there are no duplicates based on `plotID` and `clipID`.

Sampling simulation

- Goal is to sample with replacement at a variety of reduced sample sizes, calculate a mean for each sampling event, and create a distribution of means at each sample size. The 95% CI for the original n=29 sample size will also be calculated, and for each sample size, the % of means falling within the 95% CI of the original data will be calculated.