DataCleaning

Data Cleaning

BCEENET Workshop: Basics for cleaning specimen data.
2020-06-26
ML Gaynor

Load Packages

```
library(dplyr)
library(lubridate)
library(CoordinateCleaner)
```

Load raw data

```
rawdf <- read.csv("data/raw/Shortia_galacifolia_062620.csv")
```

Inspect dataframe

What columns are included?

```
names(rawdf)
```

```
##  [1] "coreid"
##  [2] "aec.associatedTaxa"
##  [3] "dc.rights"
##  [4] "dcterms.accessRights"
##  [5] "dcterms.bibliographicCitation"
##  [6] "dcterms.language"
##  [7] "dcterms.license"
##  [8] "dcterms.modified"
##  [9] "dcterms.references"
## [10] "dcterms.rights"
## [11] "dcterms.rightsHolder"
## [12] "dcterms.source"
## [13] "dcterms.type"
## [14] "dwc.Identification"
## [15] "dwc.MeasurementOrFact"
## [16] "dwc.ResourceRelationship"
## [17] "dwc.VerbatimEventDate"
## [18] "dwc.acceptedNameUsage"
```

```
##  [19] "dwc.acceptedNameUsageID"
##  [20] "dwc.accessRights"
##  [21] "dwc.associatedMedia"
##  [22] "dwc.associatedOccurrences"
##  [23] "dwc.associatedOrganisms"
##  [24] "dwc.associatedReferences"
##  [25] "dwc.associatedSequences"
##  [26] "dwc.associatedTaxa"
##  [27] "dwc.basisOfRecord"
##  [28] "dwc.bed"
##  [29] "dwc.behavior"
##  [30] "dwc.catalogNumber"
##  [31] "dwc.class"
##  [32] "dwc.classs"
##  [33] "dwc.collectionCode"
##  [34] "dwc.collectionID"
##  [35] "dwc.continent"
##  [36] "dwc.coordinatePrecision"
##  [37] "dwc.coordinateUncertaintyInMeters"
##  [38] "dwc.country"
##  [39] "dwc.countryCode"
##  [40] "dwc.county"
##  [41] "dwc.dataGeneralizations"
##  [42] "dwc.datasetID"
##  [43] "dwc.datasetName"
##  [44] "dwc.dateIdentified"
##  [45] "dwc.day"
##  [46] "dwc.decimalLatitude"
##  [47] "dwc.decimalLongitude"
##  [48] "dwc.disposition"
##  [49] "dwc.dynamicProperties"
##  [50] "dwc.earliestAgeOrLowestStage"
##  [51] "dwc.earliestEonOrLowestEonothem"
##  [52] "dwc.earliestEpochOrLowestSeries"
##  [53] "dwc.earliestEraOrLowestErathem"
##  [54] "dwc.earliestPeriodOrLowestSystem"
##  [55] "dwc.endDayOfYear"
##  [56] "dwc.establishmentMeans"
##  [57] "dwc.eventDate"
##  [58] "dwc.eventID"
##  [59] "dwc.eventRemarks"
##  [60] "dwc.eventTime"
##  [61] "dwc.family"
##  [62] "dwc.fieldNotes"
##  [63] "dwc.fieldNumber"
##  [64] "dwc.footprintSRS"
##  [65] "dwc.footprintSpatialFit"
##  [66] "dwc.footprintWKT"
##  [67] "dwc.formation"
##  [68] "dwc.genus"
##  [69] "dwc.geodeticDatum"
##  [70] "dwc.geologicalContextID"
##  [71] "dwc.georeferenceProtocol"
##  [72] "dwc.georeferenceRemarks"
```

```
##  [73] "dwc.georeferenceSources"
##  [74] "dwc.georeferenceVerificationStatus"
##  [75] "dwc.georeferencedBy"
##  [76] "dwc.georeferencedDate"
##  [77] "dwc.group"
##  [78] "dwc.habitat"
##  [79] "dwc.higherClassification"
##  [80] "dwc.higherGeography"
##  [81] "dwc.higherGeographyID"
##  [82] "dwc.highestBiostratigraphicZone"
##  [83] "dwc.identificationID"
##  [84] "dwc.identificationQualifier"
##  [85] "dwc.identificationReferences"
##  [86] "dwc.identificationRemarks"
##  [87] "dwc.identificationVerificationStatus"
##  [88] "dwc.identifiedBy"
##  [89] "dwc.individualCount"
##  [90] "dwc.informationWithheld"
##  [91] "dwc.infraspecificEpithet"
##  [92] "dwc.institutionCode"
##  [93] "dwc.institutionID"
##  [94] "dwc.island"
##  [95] "dwc.islandGroup"
##  [96] "dwc.kingdom"
##  [97] "dwc.language"
##  [98] "dwc.latestAgeOrHighestStage"
##  [99] "dwc.latestEonOrHighestEonothem"
## [100] "dwc.latestEpochOrHighestSeries"
## [101] "dwc.latestEraOrHighestErathem"
## [102] "dwc.latestPeriodOrHighestSystem"
## [103] "dwc.lifeStage"
## [104] "dwc.lithostratigraphicTerms"
## [105] "dwc.locality"
## [106] "dwc.locationAccordingTo"
## [107] "dwc.locationID"
## [108] "dwc.locationRemarks"
## [109] "dwc.lowestBiostratigraphicZone"
## [110] "dwc.materialSampleID"
## [111] "dwc.maximumDepthInMeters"
## [112] "dwc.maximumElevationInMeters"
## [113] "dwc.member"
## [114] "dwc.minimumDepthInMeters"
## [115] "dwc.minimumElevationInMeters"
## [116] "dwc.modified"
## [117] "dwc.month"
## [118] "dwc.municipality"
## [119] "dwc.nameAccordingTo"
## [120] "dwc.namePublishedIn"
## [121] "dwc.namePublishedInID"
## [122] "dwc.namePublishedInYear"
## [123] "dwc.nomenclaturalCode"
## [124] "dwc.nomenclaturalStatus"
## [125] "dwc.occurrenceDetails"
## [126] "dwc.occurrenceID"
```

```
## [127] "dwc.occurrenceRemarks"
## [128] "dwc.occurrenceStatus"
## [129] "dwc.order"
## [130] "dwc.organismID"
## [131] "dwc.organismName"
## [132] "dwc.organismQuantity"
## [133] "dwc.organismQuantityType"
## [134] "dwc.organismRemarks"
## [135] "dwc.originalNameUsage"
## [136] "dwc.originalNameUsageID"
## [137] "dwc.otherCatalogNumbers"
## [138] "dwc.ownerInstitutionCode"
## [139] "dwc.parentNameUsage"
## [140] "dwc.phylum"
## [141] "dwc.pointRadiusSpatialFit"
## [142] "dwc.preparations"
## [143] "dwc.previousIdentifications"
## [144] "dwc.recordNumber"
## [145] "dwc.recordedBy"
## [146] "dwc.reproductiveCondition"
## [147] "dwc.rights"
## [148] "dwc.rightsHolder"
## [149] "dwc.sampleSizeValue"
## [150] "dwc.samplingEffort"
## [151] "dwc.samplingProtocol"
## [152] "dwc.scientificName"
## [153] "dwc.scientificNameAuthorship"
## [154] "dwc.scientificNameID"
## [155] "dwc.sex"
## [156] "dwc.specificEpithet"
## [157] "dwc.startDayOfYear"
## [158] "dwc.stateProvince"
## [159] "dwc.subgenus"
## [160] "dwc.taxonID"
## [161] "dwc.taxonRank"
## [162] "dwc.taxonRemarks"
## [163] "dwc.taxonomicStatus"
## [164] "dwc.typeStatus"
## [165] "dwc.verbatimCoordinateSystem"
## [166] "dwc.verbatimCoordinates"
## [167] "dwc.verbatimDepth"
## [168] "dwc.verbatimElevation"
## [169] "dwc.verbatimEventDate"
## [170] "dwc.verbatimLatitude"
## [171] "dwc.verbatimLocality"
## [172] "dwc.verbatimLongitude"
## [173] "dwc.verbatimSRS"
## [174] "dwc.verbatimTaxonRank"
## [175] "dwc.vernacularName"
## [176] "dwc.waterBody"
## [177] "dwc.year"
## [178] "gbif.Identifier"
## [179] "gbif.Reference"
## [180] "idigbio.recordId"
```

```
## [181] "symbiota.recordEnteredBy"
## [182] "symbiota.verbatimScientificName"
## [183] "zan.ChronometricDate"
```

How many observations do we start with?

```
nrow(rawdf)
```

```
## [1] 387
```

1. Resolve taxon names

Inspect scientific names included in the raw df

```
unique(rawdf$dwc.scientificName)
```

```
##  [1] Shortia
##  [2] Shortia galacifolia Torr. & Gray
##  [3] Shortia galacifolia
##  [4] Berneuxia thibetica Decaisne
##  [5] Sclerotium
##  [6] Discohainesia oenotherae
##  [7] Shortia galacifolia Torr. & A.Gray
##  [8] Shortia galacifolia var. brevistyla
##  [9] Shortia galacifolia Torrey & A. Gray
## [10] Berneuxia yunnanensis H. L. Li
## [11] Shortia galacifolia var. galacifolia
## [12] Sherwoodia galacifolia
## [13] Pezizella lythri
## [14] Shortia galacifolia var. brevistyla P. A. Davies
## [15] Shortia galacifolia Torr. & A. Gray
## 15 Levels: Berneuxia thibetica Decaisne ... Shortia galacifolia var. galacifolia
```

Create a list of accepted names based on the dwc.scientificName in your dataframe

```
acceptednames <- c("Shortia galacifolia Torr. & Gray",
                   "Shortia galacifolia",
                   "Shortia galacifolia Torr. & A.Gray",
                   "Shortia galacifolia var. brevistyla",
                   "Shortia galacifolia Torrey & A. Gray",
                   "Shortia galacifolia var. galacifolia",
                   "Sherwoodia galacifolia",
                   "Shortia galacifolia var. brevistyla P. A. Davies",
                   "Shortia galacifolia Torr. & A. Gray")
```

Filter to only include accepted names

Using the R package dplyr, we:
1. filter the dataframe to only include rows with the accepted names.
2. filter out any rows with NA for dwc.scientificName.
3. create a column called name and set it equal to *"Shortia galacifolia"*

```
df <- rawdf %>%
      filter(dwc.scientificName %in% acceptednames) %>%
      filter(!is.na(dwc.scientificName))  %>%
      mutate(name = "Shortia galacifolia")
```

How many observations do we have now?

```
nrow(df)
```

```
## [1] 309
```

2. Remove Duplicates

Subset columns

Using the R package dplyr, we:
1. select and rename columns.

```
df <- df %>%
       dplyr::select(ID = coreid,
                     name = name,
                     basis = dwc.basisOfRecord,
                     catalogNumber = dwc.catalogNumber,
                     collectionCode = dwc.collectionCode,
                     collectionID = dwc.collectionID,
                     coordinateUncertaintyInMeters = dwc.coordinateUncertaintyInMeters,
                     lat = dwc.decimalLatitude,
                     long = dwc.decimalLongitude,
                     date = dwc.eventDate)
```

Fix dates

Using the R package lubridate, we first parse the date into the same format.

```
df$date <- lubridate::ymd(df$date)
```

Next you are going to seperate date into year, month, and day - where every column only contains one set of information.

```
df <- df %>%
       dplyr::mutate(year = lubridate::year(date),
                     month = lubridate::month(date),
                     day = lubridate::day(date))
```

Remove rows with identical lat, long, year, month, and day

If a specimen shares lat, long, and event date we are assuming that it is identical. Many specimen lack date and lat/long, so this may be getting rid of information you would want to keep.

```
df <- distinct(df, lat, long, year, month, day, .keep_all = TRUE)
```

How many observations do we have now?

```
nrow(df)
```

```
## [1] 57
```

3. Location cleaning

Filter missing lat and long

Using the R package dplyr, we:
1. filter out(!) any rows where long 'is.na'
2. filter out(!) any rows where lat 'is.na'

```
df <- df %>%
      filter(!is.na(long)) %>%
      filter(!is.na(lat))
```

How many observations do we have now?

```
nrow(df)
```

```
## [1] 13
```

Precision

Using the R base function 'round', we round lat and long to two decimal places.

```
df$lat <- round(df$lat, digits = 2)
df$long <- round(df$long, digits = 2)
```

Remove unlikely points

Remove points at 0.00, 0.00

Using the R package dplyr, we:
1. filter to retain rows where long is NOT(!) equal to 0.00
2. filter to retain rows where long is NOT(!) equal to 0.00

```
df <- df %>%
      filter(long != 0.00) %>%
      filter(lat != 0.00)
```

Remove coordinates in cultivated zones, botanical gardens, and outside our desired range

Using the R package CoordinateCleaner, we first if points are at biodiversity institutions and remove any points that are.

```
df <- cc_inst(df,
                    lon = "long",
                    lat = "lat",
                    species = "name")
```

```
## Testing biodiversity institutions
```

```
## Removed 0 records.
```

Next, we look for geographic outliers and remove outliers.

```
df <- cc_outl(df,
                    lon = "long",
                    lat = "lat",
                    species = "name")
```

```
## Testing geographic outliers
```

```
## Removed 1 records.
```

How many observations do we have now?

```
nrow(df)
```

```
## [1] 12
```

4. Save Cleaned .csv

```
write.csv(df,
          "data/cleaned/Shortia_galacifolia_062620-cleaned.csv",
          row.names = FALSE)
```