Solution NEON Part 2

K Duffy

8/24/2020

Sorry all, I haven't had time to finish this solution due to recording some future lectures for our class. I've push up an incomplete version, I'll revise and add more tomorrow.

```
library(tidyverse)
library(dplyr, quietly=T)
library(lubridate)
library(neonUtilities)
library(httr)
library(jsonlite)
library(downloader)
source('/Users/kdw223/Research/katharynduffy.github.io/neon_token_source.R')
```

1. Use the answers that you've provided above to select a *single* NEON site.

e.g. `ONAQ`

2. Use the answers that you've provided above to select **3 NEON data products** from either the TOS, TIS or ARS (AOP) collection methods. Sumarize each product with its NEON identifier, along with a sumarry.

In this example solution I'm going to look at 3 phenological-realted data products. **This will vary based on your research and interests**

Product 1, TOS Data: **DP1.10055.001**: Plant phenology observations: phenophase status and intensity of tagged plants. This data product contains the quality-controlled, native sampling resolution data from in-situ observations of plant leaf development and reproductive phenophases, at **D15.ONAQ**. Here I will focus on the phenophase intensity data, which is a measure of how prevalent that particular phenophase is in the sampled plants.

Product 2, TIS Data: **DP1.00033.001** Phenology images: RGB and IR images of the plant canopy taken from an automated camera on the tower top (note, there is another camera to capture understory imagery which is provided is a separate data product). Images are collected every 15 minutes and closely follow protocols of the Phenocam Network.

Product 3, RS Data: **DP3.30015.001** Ecosystem structure: If you're lazy like me, maybe you'd want to pull the abstract first and write something based on that:

```
# Request data using the GET function & the API call
req <- GET("http://data.neonscience.org/api/v0/products/DP3.30015.001")
req.content <- content(req, as="parsed")
req.content$data$productAbstract
```

```
## [1] "Forests store and sequester a considerable proportion of the terrestrial global carbon budget. 
```

3. Using the NEON Ulitites package or the API pull in those data along with metadata.

```
#TOS Phenology Data
sitesOfInterest <- c("ONAQ")
```

```r
dpid <- as.character('DP1.10055.001') #phe data

pheDat <- loadByProduct(dpID="DP1.10055.001",
                        site = sitesOfInterest,
                        package = "expanded",
                        check.size = FALSE,
                        token=NEON_TOKEN)
```

Just to 'mix things up' I'll use the API for my imagery:

```r
#TIS Phenology Imagery
# Request data using the GET function & the API call
req <- GET("http://data.neonscience.org/api/v0/products/DP1.00033.001")
req
```

```
## Response [https://data.neonscience.org/api/v0/products/DP1.00033.001]
##   Date: 2020-08-28 23:18
##   Status: 200
##   Content-Type: application/json;charset=UTF-8
##   Size: 161 kB
```

```r
# View requested data
req.content <- content(req, as="parsed")
names(req.content$data)
```

```
##  [1] "productCodeLong"              "productCode"
##  [3] "productCodePresentation"      "productName"
##  [5] "productDescription"           "productStatus"
##  [7] "productCategory"              "productHasExpanded"
##  [9] "productScienceTeamAbbr"       "productScienceTeam"
## [11] "productPublicationFormatType" "productAbstract"
## [13] "productDesignDescription"     "productStudyDescription"
## [15] "productBasicDescription"      "productExpandedDescription"
## [17] "productSensor"                "productRemarks"
## [19] "themes"                       "changeLogs"
## [21] "specs"                        "keywords"
## [23] "siteCodes"
```

```r
req.content$data$productAbstract
```

```
## [1] "Phenology is the study of reoccurring life cycle events that are driven by environmental factor
```

Interesting! I actually can't pull the NEON phenology imagery data from the NEON API. Good thing I checked the abstract. If I were a student (and not a former member of the PhenoCam lab) I'd probably go to the website, search for data etc etc, and then months later realize there's a slick `phenocamAPI` *(yay APIs!!)* that I can use.

I'll jump a few chapters ahead and use that data:

```r
library(phenocamapi)
# obtaining midday_images for NEON ONAQ
ONAQ_middays <- get_midday_list('NEON.D15.ONAQ.DP1.00033')
# see the first few rows
head(ONAQ_middays)
```

```
## [1] "http://phenocam.sr.unh.edu/data/archive/NEON.D15.ONAQ.DP1.00033/2016/12/NEON.D15.ONAQ.DP1.00033
## [2] "http://phenocam.sr.unh.edu/data/archive/NEON.D15.ONAQ.DP1.00033/2016/12/NEON.D15.ONAQ.DP1.00033
```

```
## [3] "http://phenocam.sr.unh.edu/data/archive/NEON.D15.ONAQ.DP1.00033/2016/12/NEON.D15.ONAQ.DP1.00033_
## [4] "http://phenocam.sr.unh.edu/data/archive/NEON.D15.ONAQ.DP1.00033/2016/12/NEON.D15.ONAQ.DP1.00033_
## [5] "http://phenocam.sr.unh.edu/data/archive/NEON.D15.ONAQ.DP1.00033/2016/12/NEON.D15.ONAQ.DP1.00033_
## [6] "http://phenocam.sr.unh.edu/data/archive/NEON.D15.ONAQ.DP1.00033/2016/12/NEON.D15.ONAQ.DP1.00033_
```

Okay, now I have links to go grab those 'mid-day' images.

BUT…

The way I was calling data from the `neonUtilities` package doesn't work anymore, so I could ask the `phenocamapi` package what it wants:

```
phenocams=get_phenos()
head(phenocams)
```

```
##                   site      lat        lon elev active utc_offset date_first
## 1: aafcottawacfiaf14e 45.29210  -75.76640   90   TRUE         -5 2020-04-27
## 2: aafcottawacfiaf14w 45.29210  -75.76640   90   TRUE         -5 2020-05-01
## 3:             acadia 44.37694  -68.26083  158   TRUE         -5 2007-03-15
## 4:      aguatibiaeast 33.62200 -116.86700 1086  FALSE         -8 2007-08-16
## 5:     aguatibianorth 33.60222 -117.34368 1090  FALSE         -8 2003-10-01
## 6:           ahwahnee 37.74670 -119.58160 1199   TRUE         -8 2008-08-28
##     date_last infrared                                      contact1
## 1: 2020-08-24        N Elizabeth Pattey <elizabeth.pattey@canada.ca>
## 2: 2020-08-24        N Elizabeth Pattey <elizabeth.pattey@canada.ca>
## 3: 2020-08-27        N                  Dee Morse <dee_morse@nps.gov>
## 4: 2019-01-25        N               Ann E Mebane <amebane@fs.fed.us>
## 5: 2006-10-25        N
## 6: 2020-08-27        N                  Dee Morse <dee_morse@nps.gov>
##                                         contact2
## 1: Luc Pelletier <luc.pelletier3@canada.ca>
## 2: Luc Pelletier <luc.pelletier3@canada.ca>
## 3:          John Gross <John_Gross@nps.gov>
## 4:   Kristi Savig <KSavig@air-resource.com>
## 5:
## 6:          John Gross <John_Gross@nps.gov>
##                                                 site_description site_type
## 1:  AAFC Site - Ottawa (On) - CFIA - Field F14 -East Flux Tower        II
## 2:  AAFC Site - Ottawa (On) - CFIA - Field F14 -West Flux Tower        II
## 3: Acadia National Park, McFarland Hill, near Bar Harbor, Maine       III
## 4:                            Agua Tibia Wilderness, California       III
## 5:                            Agua Tibia Wilderness, California       III
## 6:          Ahwahnee Meadow, Yosemite National Park, California       III
##                   group       camera_description camera_orientation flux_data
## 1:                 <NA> Campbell Scientific CCFC                 NE      TRUE
## 2:                 <NA> Campbell Scientific CCFC                WNW      TRUE
## 3: National Park Service                 unknown                 NE     FALSE
## 4:                 USFS                 unknown                 SW     FALSE
## 5:                 USFS                 unknown                 NE     FALSE
## 6: National Park Service                 unknown                  E     FALSE
##    flux_networks flux_sitenames
## 1:     <list[0]>           <NA>
## 2:     <list[1]>           <NA>
## 3:     <list[0]>
## 4:     <list[0]>
## 5:     <list[0]>
```

```
## 6:     <list[0]>
##                                             dominant_species primary_veg_type
## 1: Zea mays, Triticum aestivum, Brassica napus, Glycine max               AG
## 2: Zea mays, Triticum aestivum, Brassica napus, Glycine max               AG
## 3:                                                                        DB
## 4:                                                                        SH
## 5:                                                                        SH
## 6:                                                                        EN
##    secondary_veg_type site_meteorology MAT_site MAP_site MAT_daymet MAP_daymet
## 1:                 AG             TRUE      6.4      943       6.30        952
## 2:                 AG             TRUE      6.4      943       6.30        952
## 3:                 EN            FALSE       NA       NA       7.05       1439
## 4:                               FALSE       NA       NA      15.75        483
## 5:                               FALSE       NA       NA      16.00        489
## 6:                 GR            FALSE       NA       NA      12.25        871
##    MAT_worldclim MAP_worldclim koeppen_geiger ecoregion landcover_igbp
## 1:           6.0           863            Dfb         8             12
## 2:           6.0           863            Dfb         8             12
## 3:           6.5          1303            Dfb         8              5
## 4:          14.9           504            Csa        11              7
## 5:          13.8           729            Csa        11              7
## 6:          11.8           886            Csb         6              8
##    dataset_version1
## 1:               NA
## 2:               NA
## 3:               NA
## 4:               NA
## 5:               NA
## 6:               NA
##
## 1:  Camera funded by Agriculture and Agri-Food Canada (AAFC) Project J-001735 - Commercial inhibitors
## 2: Cameras funded by Agriculture and Agri-Food Canada (AAFC) Project J-001735 - Commercial inhibitors
## 3:
## 4:
## 5:
## 6:
##                          modified flux_networks_name flux_networks_url
## 1: 2020-05-04T10:46:30.065790-04:00               <NA>              <NA>
## 2: 2020-05-04T10:46:32.523976-04:00              OTHER
## 3: 2016-11-01T15:42:15.016778-04:00               <NA>              <NA>
## 4: 2016-11-01T15:42:15.086984-04:00               <NA>              <NA>
## 5: 2016-11-01T15:42:15.095277-04:00               <NA>              <NA>
## 6: 2016-11-01T15:42:15.111916-04:00               <NA>              <NA>
##    flux_networks_description
## 1:                      <NA>
## 2:        Other/Unaffiliated
## 3:                      <NA>
## 4:                      <NA>
## 5:                      <NA>
## 6:                      <NA>
```

Looking at the dataframe and searching for `ONAQ` I see that I have to give this full name to pull the imagery: *NEON.D15.ONAQ.DP1.00033*

```r
# open a temporary directory
tmp_dir <- tempdir()

# download a subset.
download_midday_images(site = 'NEON.D15.ONAQ.DP1.00033', # which site
                       y = 2019, # which year(s)
                       months = 1:12, # which month(s)
                       days = 15, # which days on month(s)
                       download_dir = tmp_dir) # where on your computer
```

```
##   |                                                                      |
```

```
## [1] "/var/folders/5n/vxrdb72140b43_jhsmhfy514pqgq1n/T//RtmpBIcEwV"
```

```r
# list of downloaded files
ONAQ_middays_path <- dir(tmp_dir, pattern = 'NEON.D15.ONAQ.*', full.names = TRUE)

head(ONAQ_middays_path)
```

```
## [1] "/var/folders/5n/vxrdb72140b43_jhsmhfy514pqgq1n/T//RtmpBIcEwV/NEON.D15.ONAQ.DP1.00033_2019_01_15
## [2] "/var/folders/5n/vxrdb72140b43_jhsmhfy514pqgq1n/T//RtmpBIcEwV/NEON.D15.ONAQ.DP1.00033_2019_02_15
## [3] "/var/folders/5n/vxrdb72140b43_jhsmhfy514pqgq1n/T//RtmpBIcEwV/NEON.D15.ONAQ.DP1.00033_2019_03_15
## [4] "/var/folders/5n/vxrdb72140b43_jhsmhfy514pqgq1n/T//RtmpBIcEwV/NEON.D15.ONAQ.DP1.00033_2019_04_15
## [5] "/var/folders/5n/vxrdb72140b43_jhsmhfy514pqgq1n/T//RtmpBIcEwV/NEON.D15.ONAQ.DP1.00033_2019_05_15
## [6] "/var/folders/5n/vxrdb72140b43_jhsmhfy514pqgq1n/T//RtmpBIcEwV/NEON.D15.ONAQ.DP1.00033_2019_06_15
```

Great, the data is there.

4. Organize your data into `data.frames` and produce summaries for each of your data:

```r
#NEON sends the data as a nested list, so I need to undo that
# unlist all data frames
list2env(pheDat ,.GlobalEnv)
summary(phe_perindividualperyear)
summary(phe_statusintensity)
```

5. Filter and format your data based on metadata and quality flags:

```r
#remove duplicate records
phe_statusintensity <- select(phe_statusintensity, -uid)
phe_statusintensity <- distinct(phe_statusintensity)
#Format dates (native format is 'factor' silly R)
phe_statusintensity$date <- as.Date(phe_statusintensity$date, "%Y-%m-%d")
phe_statusintensity$editedDate <- as.Date(phe_statusintensity$editedDate, "%Y-%m-%d")
phe_statusintensity$year <- substr(phe_statusintensity$date, 1, 4)
phe_statusintensity$monthDay <- format(phe_statusintensity$date, format="%m-%d")
```

Now I want to remove NA values so I can see what's really going on:

```r
phe_statusintensity=phe_statusintensity%>%
  filter(!is.na(phe_statusintensity$phenophaseIntensity))
```

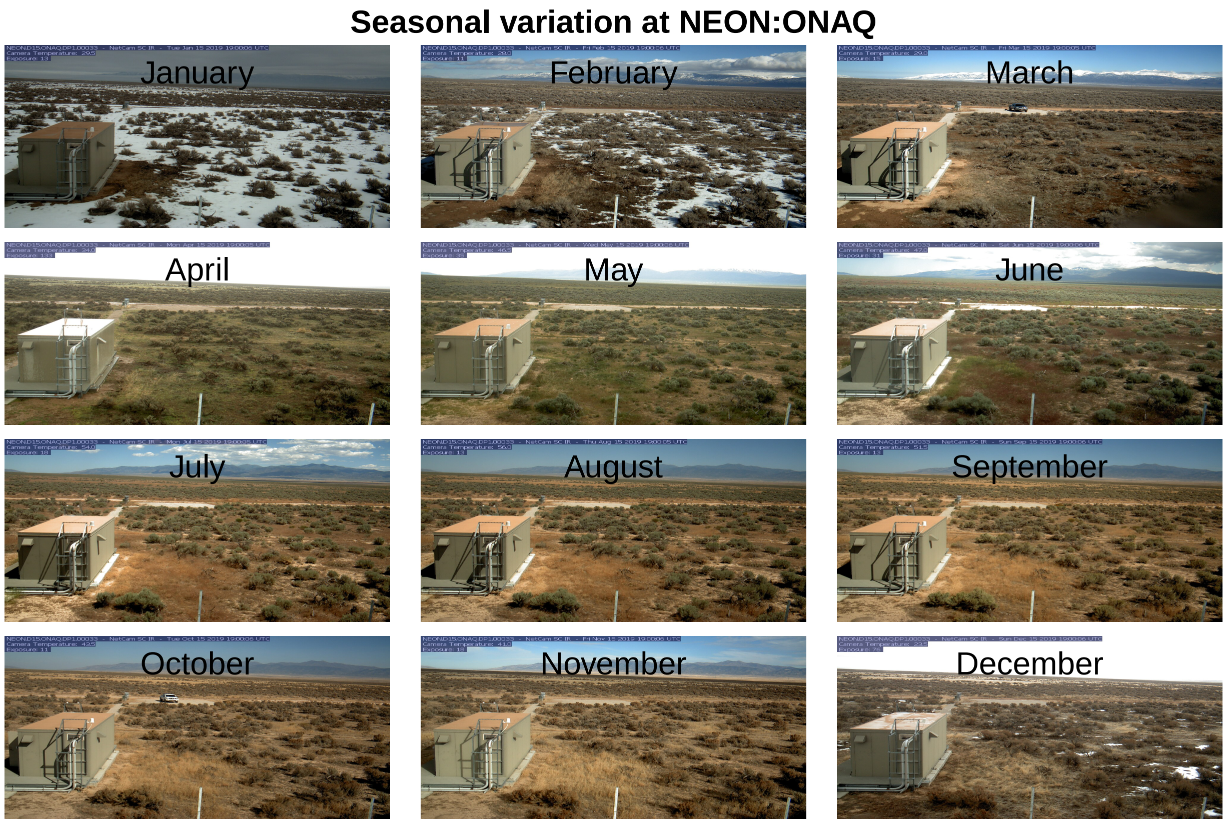
Based on looking at the status intensity data and metadata I'm going to just stick to the `status intensity:leaves` metrics here.

6. Create minimum of 1 plot per data type (minimum of 3 plots total). These will vary based on that data that you've chosen.

Second, let's look at the PhenoCam imagery over a year:

```r
library(jpeg)
n <- length(ONAQ_middays_path)
par(mar= c(0,0,0,0), mfrow=c(4,3), oma=c(0,0,3,0))

for(i in 1:n){
  img <- readJPEG(ONAQ_middays_path[i])
  plot(0:1,0:1, type='n', axes= FALSE, xlab= '', ylab = '')
  rasterImage(img, 0, 0, 1, 1)
  mtext(month.name[i], line = -2)
}
mtext('Seasonal variation at NEON:ONAQ', font = 2, outer = TRUE)
```



7. What is the temporal frequency of observations in the data you decided was of interest? How do the data align to answer a central question? What challenges did you run into when investigating these data? How will you address these challenges and document your code? *One to two paragraphs*

Pulling the abstracts really helped me with this. The TOS phenology data is measured every

The PhenoCam imagery is taken every 15 minutes. Think about just how much that that is!

The AOP flights are ideally every year, but that can be hard to coordinate (we'll talk about that in Chapter 7).