ESPM 174A - Homework assignment 2

Daniela Rodriguez-Chavez

September 29, 2023

Instructions

Work on the exercise below, due September 29 (before midnight) on bCourses. Please write **short answers (<100 words)** in "your text here", as well as the R code you used to get there (in "your code here").

```
# Load packages
library(tidyverse)
library(neon4cast) # remotes::install_github("eco4cast/neon4cast")
library(neonUtilities)
library(BiocManager)
```

1) Preliminary statistics [3 points].

1) Please upload a sample of the data (.csv file) you plan to analyze for your final project. You can (but do not need to) upload the whole dataset. However, the closer the resemblance of this data set to the one you will end up analyzing, the better. E.g., if your question is at the community level, then include several species; if you would like to compare a particular physical variable across different sites, then include several sites. The goal is for you to start getting familiar with your data and its level of complexity. In the code below, import your data set in R, and examine the following properties: (1) length and frequency of the time series (whether it is one, or multiple time series); (2) completeness of each time series; (3) basic descriptive statistics for each time series (at least mean, CV, ACF for each variable; plus anything else you would like to add). [3 points total]

```
# 1. the length and frequency of the time series
tick_data <- read_csv("https://data.ecoforecast.org/neon4cast-targets/ticks/ticks-targets.csv.gz")
```

```
## Rows: 622 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): site_id, variable, iso_week
## dbl  (1): observation
## date (1): datetime
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# we can use the max and min to find the span of the time series (including NAs)
min_date <- min(tick_data$datetime)
max_date <- max(tick_data$datetime)
# to get the total length in days we have
paste('There is a time difference of', max_date - min_date, 'days, with the starting date of',
      min_date); paste('and the last date recorded being', max_date)
```

```
## [1] "There is a time difference of 3318 days, with the starting date of 2014-04-14"
## [1] "and the last date recorded being 2023-05-15"
```

```
# 2. completeness of each time series

# note that there are 9 total sites
sitenames <- unique(tick_data$site_id)
# so we could do completeness for each site based on the whole maximum min/max across all sites:
max_range <- length(seq(min_date, max_date, "week"))

for (i in sitenames) {
  SITE_DATA <- tick_data |> filter(site_id == i) |> select(c('datetime', 'observation'))
  numrows <- dim(SITE_DATA)[1]
  percent_missing <- round((1 - numrows/max_range)*100, digits = 2)
  print(paste('Site', i, 'has', percent_missing, 'percent missing data'))
}
```

```
## [1] "Site BLAN has 85.68 percent missing data"
## [1] "Site KONZ has 88 percent missing data"
## [1] "Site LENO has 94.74 percent missing data"
## [1] "Site ORNL has 83.16 percent missing data"
## [1] "Site OSBS has 84.63 percent missing data"
## [1] "Site SCBI has 81.05 percent missing data"
## [1] "Site SERC has 84.63 percent missing data"
## [1] "Site TALL has 82.74 percent missing data"
## [1] "Site UKFS has 84.42 percent missing data"
```

```
# or we could split up each site into different datasets and do it that way
expand_data <- function(SITE_NAME) {

  SITE_DATA <- tick_data |> filter(site_id == SITE_NAME) |> select(c('datetime', 'observation'))
  # getting date ranges of the specific site
  min_date <- min(SITE_DATA$datetime)
  max_date <- max(SITE_DATA$datetime)

  # create a sequence of weeks
  datetime <- seq(min_date, max_date, "week")
  # create a df where this is a column
  EXPANDED_DATA <- data.frame(datetime)
  # create a baseline NaN observation vals so we can fill them in if we have them
  EXPANDED_DATA['observation'] <- NaN

  for (i in 1:length(datetime)) {
    # checking to see if the week has an observation in the site-specific dataset
    if (EXPANDED_DATA[i,]$datetime %in% SITE_DATA$datetime) {
      # getting the week to check with the site-specific dataset
      present_week <- EXPANDED_DATA[i,]$datetime
      # getting the specific row in the site-specific dataset
      count <- SITE_DATA |> filter(datetime == EXPANDED_DATA[i,]$datetime)
      # getting the observation number
      count <- count$observation
      # replacing the value in the larger dataset
      EXPANDED_DATA[i,]$observation <- count
      }
  }
EXPANDED_DATA
```

```r
}

# all the different sites
BLAN <- expand_data('BLAN')
KONZ <- expand_data('KONZ')
LENO <- expand_data('LENO')
ORNL <- expand_data('ORNL')
OSBS <- expand_data('OSBS')
SCBI <- expand_data('SCBI')
SERC <- expand_data('SERC')
TALL <- expand_data('TALL')
UKFS <- expand_data('UKFS')

k <- 1
for (i in list(BLAN, KONZ, LENO, ORNL, OSBS, SCBI, SERC, TALL, UKFS)){
  site_names <- c("BLAN", "KONZ", "LENO", "ORNL", "OSBS", "SCBI", "SERC", "TALL", "UKFS")
  percent_missing <- round((1 - sum(!is.na(i$observation))/dim(i)[1])*100, digits = 2)
  print(paste('Site', site_names[k], 'has', percent_missing, 'percent missing data'))
  k <- k + 1
}
```

```
## [1] "Site BLAN has 82.25 percent missing data"
## [1] "Site KONZ has 84.88 percent missing data"
## [1] "Site LENO has 90.46 percent missing data"
## [1] "Site ORNL has 82.8 percent missing data"
## [1] "Site OSBS has 83.41 percent missing data"
## [1] "Site SCBI has 79.17 percent missing data"
## [1] "Site SERC has 81.09 percent missing data"
## [1] "Site TALL has 82.52 percent missing data"
## [1] "Site UKFS has 80.37 percent missing data"
```

```r
# 3. basic descriptive statistics for each time series (at least mean, CV, ACF for each variable
# plus anything else you would like to add)

# checking if our data missing percentage goes down if we group by month
# also calculating CV for each month
CV <- function(x, ...){(sd(x, ...)/mean(x, ...))*100} # specify a function for CV
# recall that the coefficient of variation (CV) is the ratio of the standard deviation to the mean.
# The higher the coefficient of variation, the greater the level of dispersion around the mean.
# It is generally expressed as a percentage.
get_month_mean_CV <- function(DATA){
  FINAL_DF <- DATA %>% group_by(month = month(datetime), year = year(datetime)) %>%
  summarise(observation_monthly_mean = mean(observation, na.rm=TRUE),
            monthly_CV = CV(observation, na.rm = T))
  FINAL_DF
}

BLAN_month <- get_month_mean_CV(BLAN)
KONZ_month <- get_month_mean_CV(KONZ)
LENO_month <- get_month_mean_CV(LENO)
ORNL_month <- get_month_mean_CV(ORNL)
OSBS_month <- get_month_mean_CV(OSBS)
SCBI_month <- get_month_mean_CV(SCBI)
SERC_month <- get_month_mean_CV(SERC)
```

```
TALL_month <- get_month_mean_CV(TALL)
UKFS_month <- get_month_mean_CV(UKFS)
```

```
# we see that when grouping by month, the % missing data significantly decreases
# but also note that if you looked at CV, we get NaNs when there is only
# one data point, so CV coverage is still really sparse until you do by year
k <- 1
for (i in list(BLAN_month, KONZ_month, LENO_month, ORNL_month, OSBS_month,
               SCBI_month, SERC_month, TALL_month, UKFS_month)){
  site_names <- c("BLAN", "KONZ", "LENO", "ORNL", "OSBS", "SCBI", "SERC", "TALL", "UKFS")
  percent_missing <- round((1 - sum(!is.na(i$observation_monthly_mean))/dim(i)[1])*100, digits=2)
  print(paste('When grouped by month, site', site_names[k], 'has',
              percent_missing, 'percent missing data'))
  k <- k + 1
}
```

```
## [1] "When grouped by month, site BLAN has 41.57 percent missing data"
## [1] "When grouped by month, site KONZ has 52.27 percent missing data"
## [1] "When grouped by month, site LENO has 65.57 percent missing data"
## [1] "When grouped by month, site ORNL has 46.3 percent missing data"
## [1] "When grouped by month, site OSBS has 47.06 percent missing data"
## [1] "When grouped by month, site SCBI has 36 percent missing data"
## [1] "When grouped by month, site SERC has 44.94 percent missing data"
## [1] "When grouped by month, site TALL has 45.87 percent missing data"
## [1] "When grouped by month, site UKFS has 42.53 percent missing data"
```

```
# getting mean for each year
k <- 1
for (i in list(BLAN_month, KONZ_month, LENO_month, ORNL_month, OSBS_month,
               SCBI_month, SERC_month, TALL_month, UKFS_month)){
  site_names <- c("BLAN", "KONZ", "LENO", "ORNL", "OSBS", "SCBI", "SERC", "TALL", "UKFS")
  DATA <- i %>% group_by(year = year) %>%
    summarise(year_mean = round(mean(observation_monthly_mean, na.rm=TRUE), digits=2)) %>%
    pivot_wider(names_from = year, values_from = year_mean) %>% as.matrix()
  print(paste('Site', site_names[k], 'has yearly mean values of:'))
  print(DATA)
  cat(' ', sep="\n\n")
  k <- k + 1
}
```

```
## [1] "Site BLAN has yearly mean values of:"
##      2015 2016 2017  2018 2019 2020 2021  2022
## [1,]  2.7 6.28 0.83 11.04 3.09 0.81 3.11 11.17
##
## [1] "Site KONZ has yearly mean values of:"
##      2015   2016  2017 2018  2019   2020   2021  2022
## [1,] 22.5 130.83 48.33  108 99.17 186.25 114.72 57.36
##
## [1] "Site LENO has yearly mean values of:"
##       2016 2017  2018  2019 2020 2021
## [1,] 28.89 17.8 13.84 13.78  NaN   10
##
## [1] "Site ORNL has yearly mean values of:"
##        2014  2015  2016  2017   2018  2019  2020 2021 2022  2023
## [1,] 240.66 45.12 46.84 81.84 195.55 36.77 66.18 28.7 36.7 46.41
```

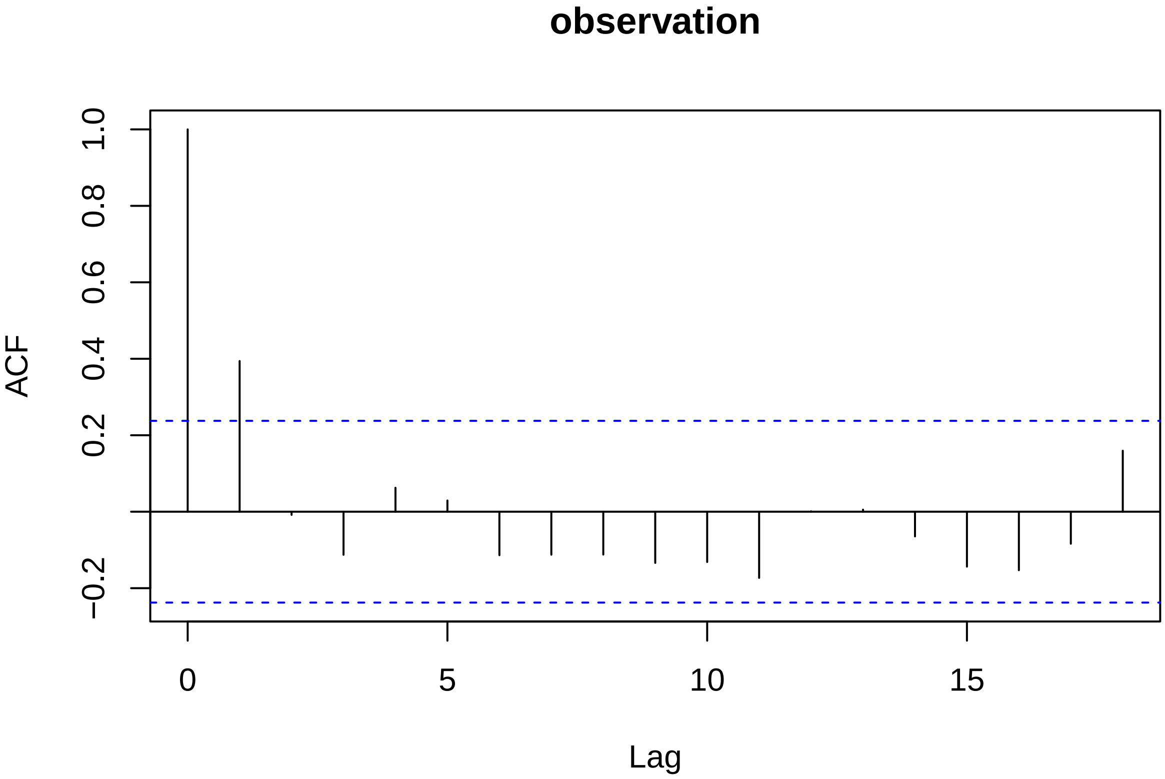
```
##
## [1] "Site OSBS has yearly mean values of:"
##       2014  2015  2016  2017 2018  2019  2020  2021  2022
## [1,] 39.38 34.14 69.76 128.8 21.6 42.93 21.67 25.24 22.02
##
## [1] "Site SCBI has yearly mean values of:"
##       2014  2015 2016 2017  2018 2019 2020  2021  2022
## [1,] 12.42 61.76 4.56 26.9 30.93 6.91 7.23 12.96 17.72
##
## [1] "Site SERC has yearly mean values of:"
##       2015  2016  2017  2018  2019 2020  2021  2022
## [1,] 48.86 56.96 65.98 93.84 46.63 3.75 70.98 43.45
##
## [1] "Site TALL has yearly mean values of:"
##      2014  2015  2016   2017  2018 2019 2020   2021 2022  2023
## [1,]  111 45.72 62.01 102.09 51.08 51.8 57.4 141.98   12 47.66
##
## [1] "Site UKFS has yearly mean values of:"
##        2015  2016   2017  2018   2019 2020   2021   2022
## [1,] 227.77 69.31 139.38 88.34 104.32  185 310.02 399.55
##
```

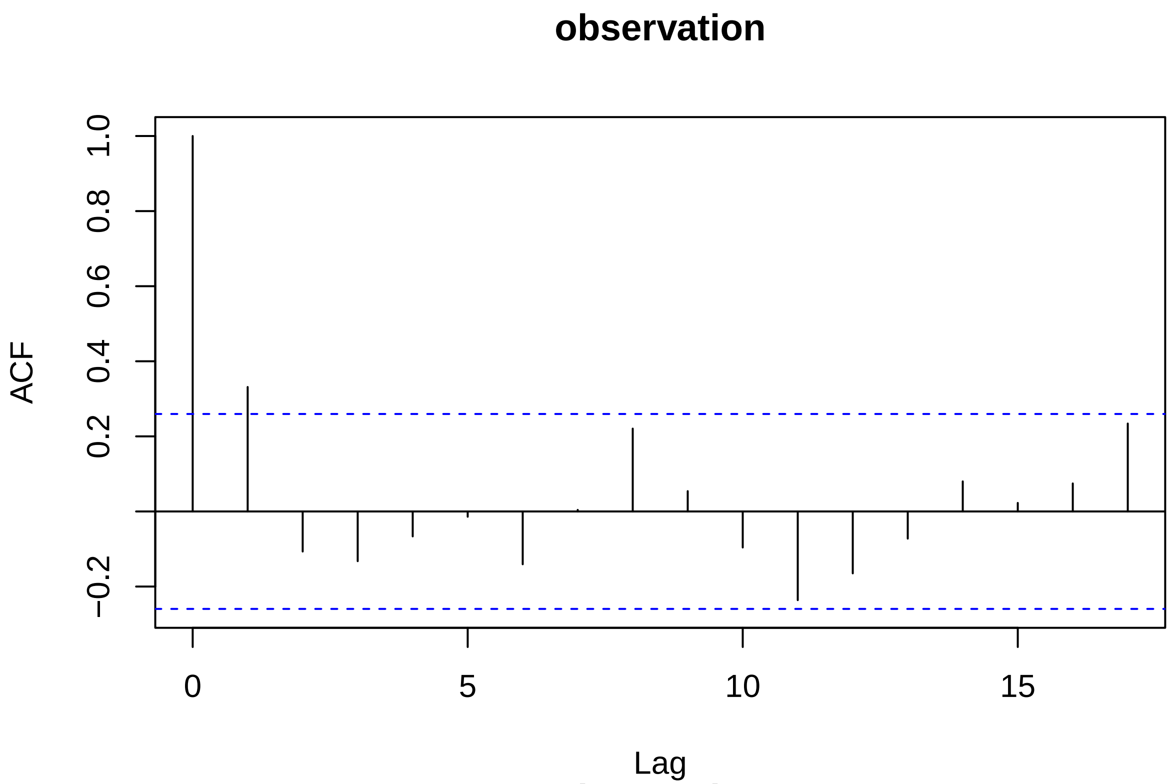
```
# CV for each year
k <- 1
for (i in list(BLAN_month, KONZ_month, LENO_month, ORNL_month, OSBS_month,
               SCBI_month, SERC_month, TALL_month, UKFS_month)){
  site_names <- c("BLAN", "KONZ", "LENO", "ORNL", "OSBS", "SCBI", "SERC", "TALL", "UKFS")
  DATA <- i %>% group_by(year = year) %>%
    summarise(year_CV = round(CV(monthly_CV, na.rm=TRUE), digits=2)) %>%
    pivot_wider(names_from = year, values_from = year_CV) %>% as.matrix()
  print(paste('Site', site_names[k], 'has yearly CV values of:'))
  print(DATA)
  cat(' ', sep="\n\n")
  k <- k + 1
}
```

```
## [1] "Site BLAN has yearly CV values of:"
##      2015 2016 2017  2018   2019 2020 2021 2022
## [1,]   NA 58.8    0 67.34 138.22   NA   NA   NA
##
## [1] "Site KONZ has yearly CV values of:"
##      2015  2016  2017  2018   2019  2020  2021 2022
## [1,]   NA 42.49 58.02 35.36 141.42 48.17 53.93   NA
##
## [1] "Site LENO has yearly CV values of:"
##      2016 2017 2018 2019 2020 2021
## [1,]   NA 8.57   NA   NA   NA   NA
##
## [1] "Site ORNL has yearly CV values of:"
##      2014  2015  2016  2017  2018  2019  2020  2021   2022 2023
## [1,]   NA 11.29 92.73 81.87 33.44 72.72 33.93 53.75 121.62   NA
##
## [1] "Site OSBS has yearly CV values of:"
##        2014  2015  2016   2017  2018  2019 2020   2021  2022
## [1,] 117.85 49.96 67.44 104.81 19.97 49.92   NA 104.21 45.53
```

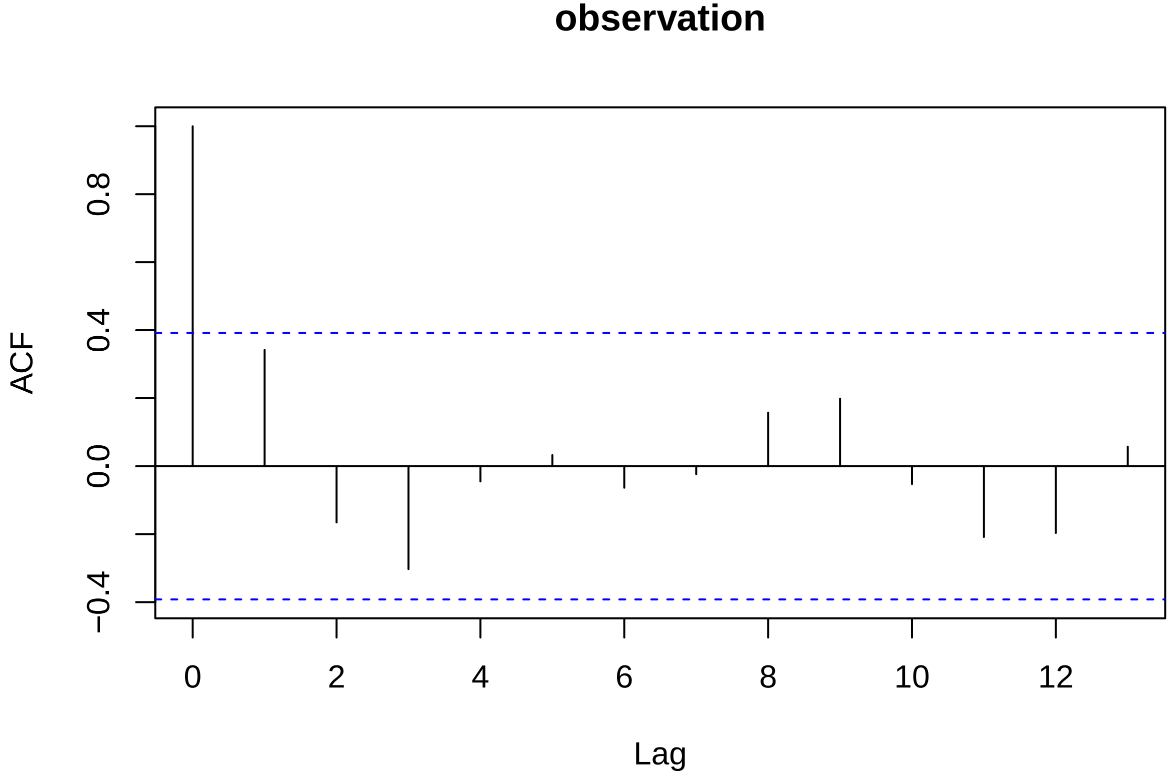
```
##
## [1] "Site SCBI has yearly CV values of:"
##       2014   2015  2016  2017  2018  2019 2020   2021  2022
## [1,] 52.67 134.44 14.29 56.81 36.41 72.03   NA 106.15 51.06
##
## [1] "Site SERC has yearly CV values of:"
##       2015  2016  2017   2018  2019 2020  2021  2022
## [1,] 70.33 67.72 41.18 113.07 61.32   NA 75.35 71.69
##
## [1] "Site TALL has yearly CV values of:"
##      2014 2015   2016  2017   2018  2019  2020  2021  2022 2023
## [1,]   NA 38.8 120.02 86.39 106.01 83.96 77.33 93.05 42.03   NA
##
## [1] "Site UKFS has yearly CV values of:"
##      2015  2016  2017 2018   2019   2020  2021 2022
## [1,]   NA 78.87 68.13 67.4 105.45 140.39 87.68 9.07
##
```

```
# getting ACF
for (i in list(BLAN, KONZ, LENO, ORNL, OSBS, SCBI, SERC, TALL, UKFS)){
  ACF_DATA <- i %>% dplyr::select(observation) %>% drop_na() %>% as.matrix()
  acf(ACF_DATA)
}
```

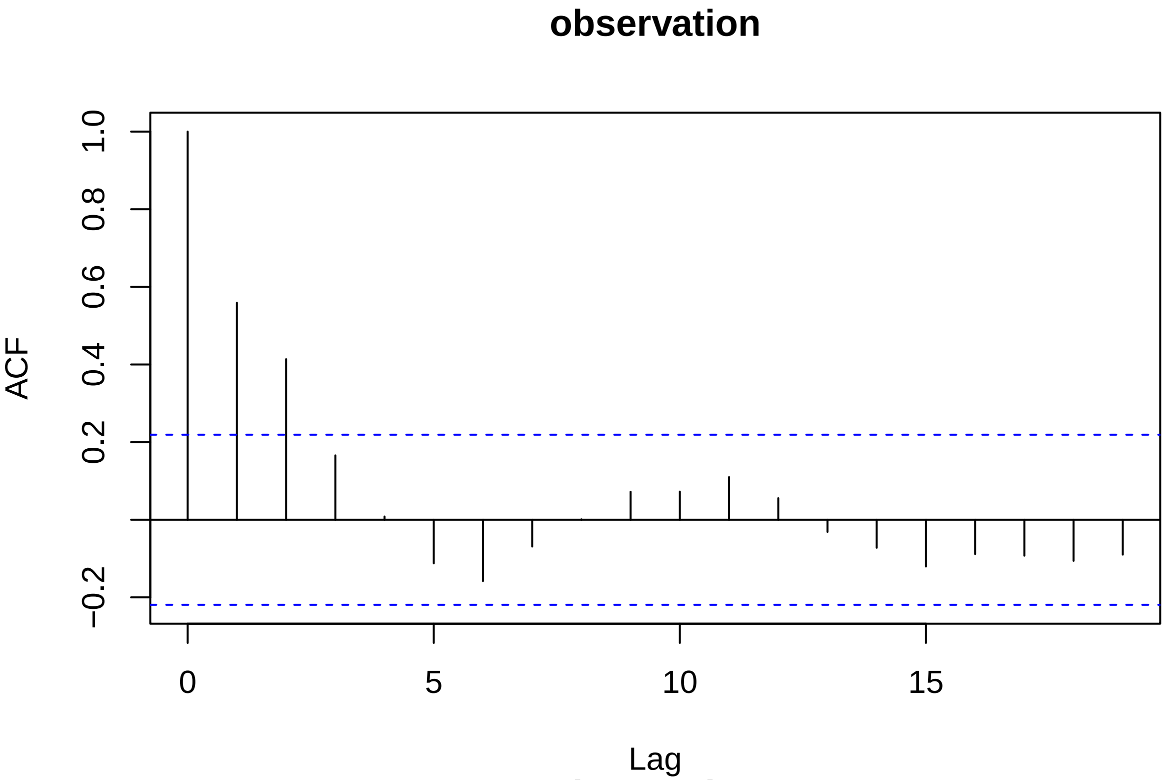




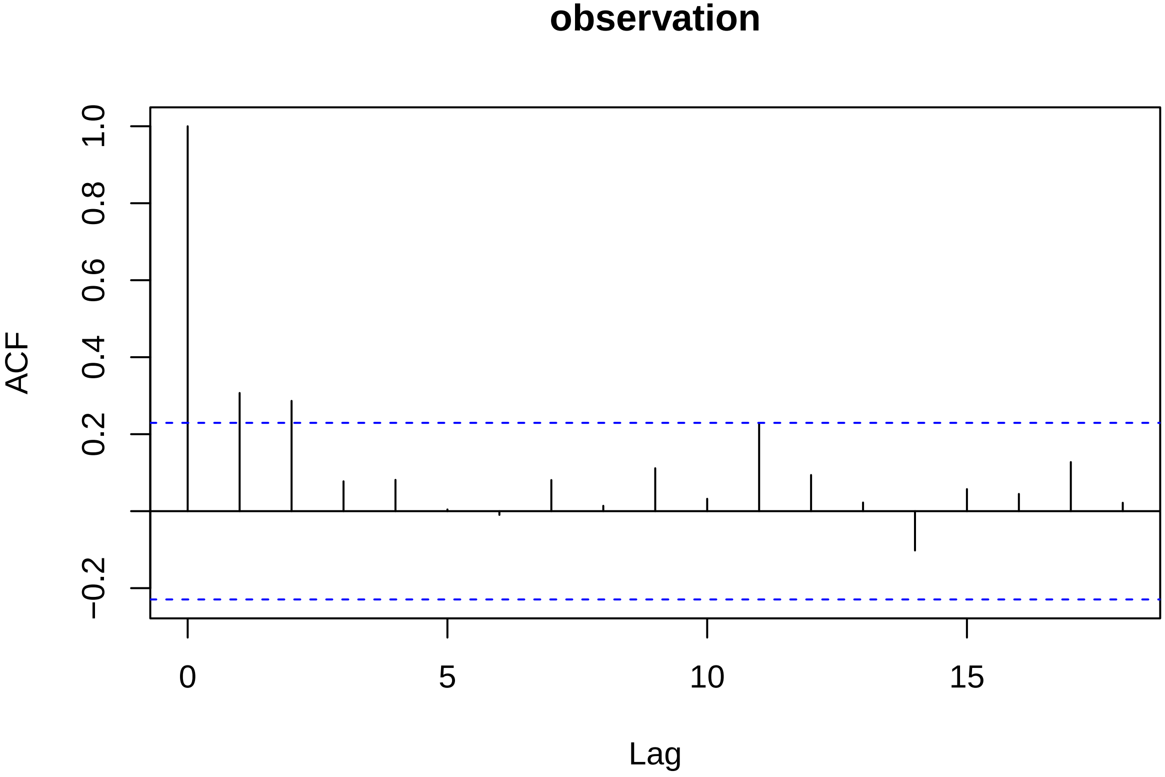
observation



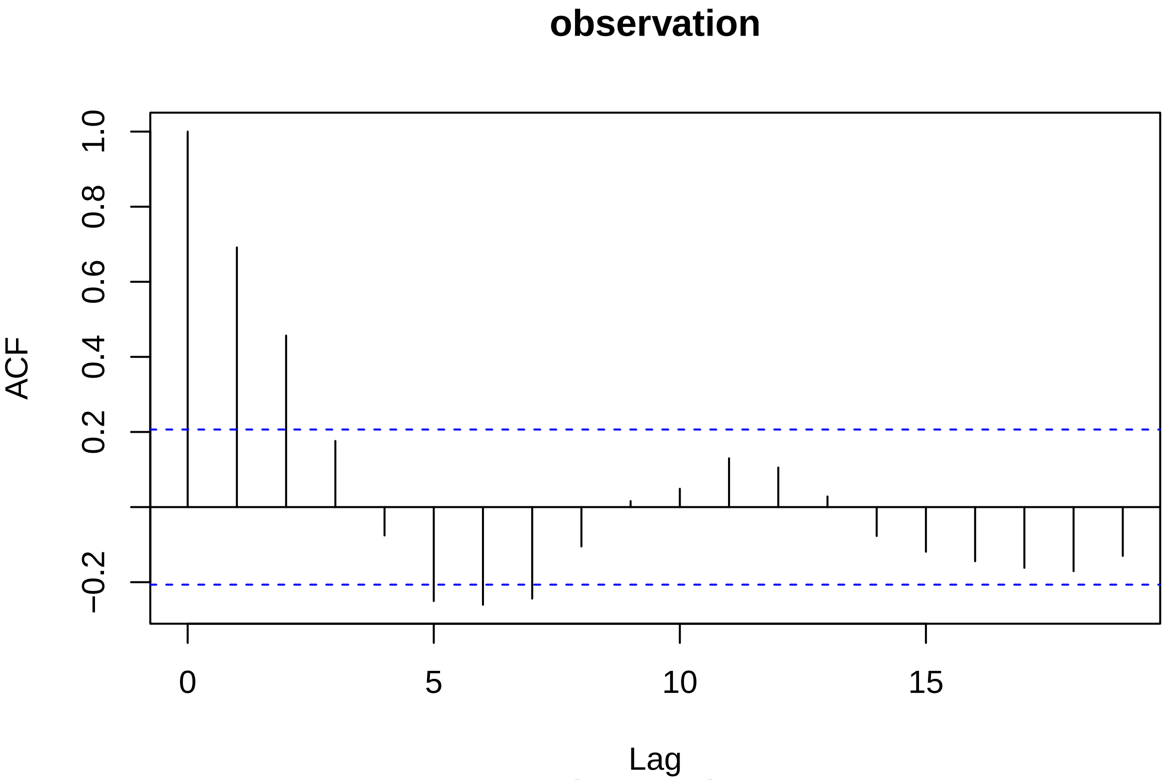
observation

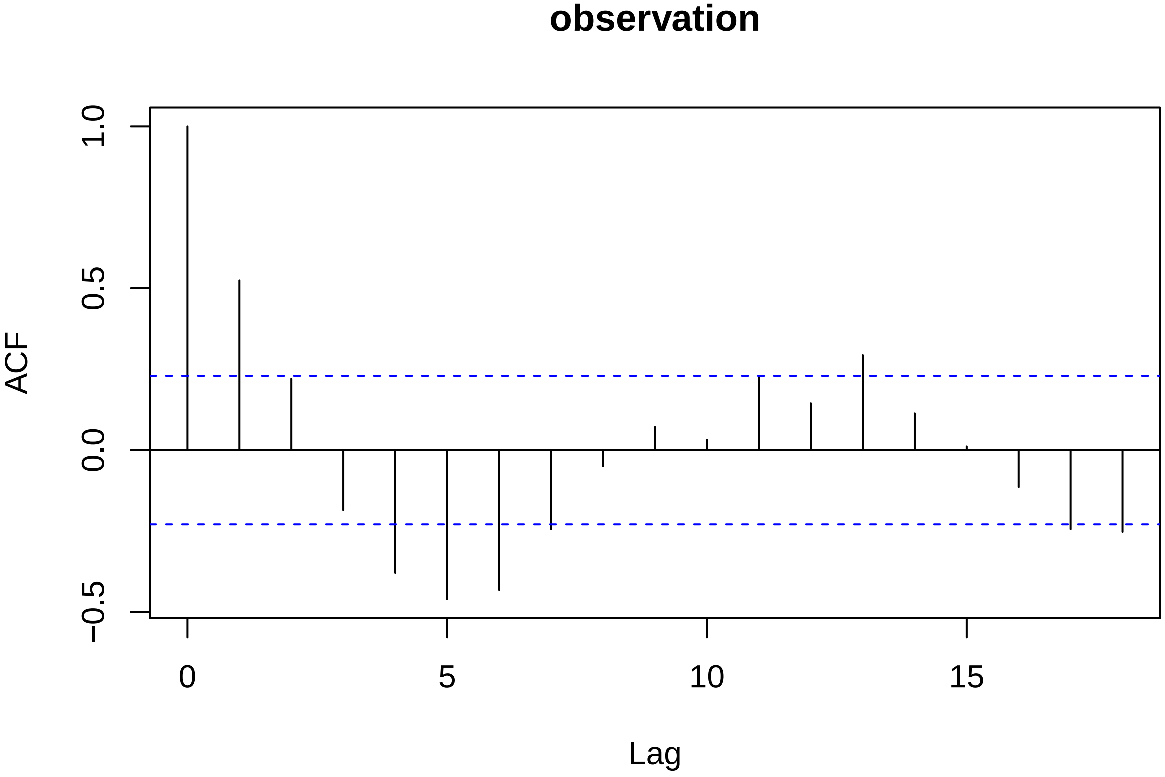


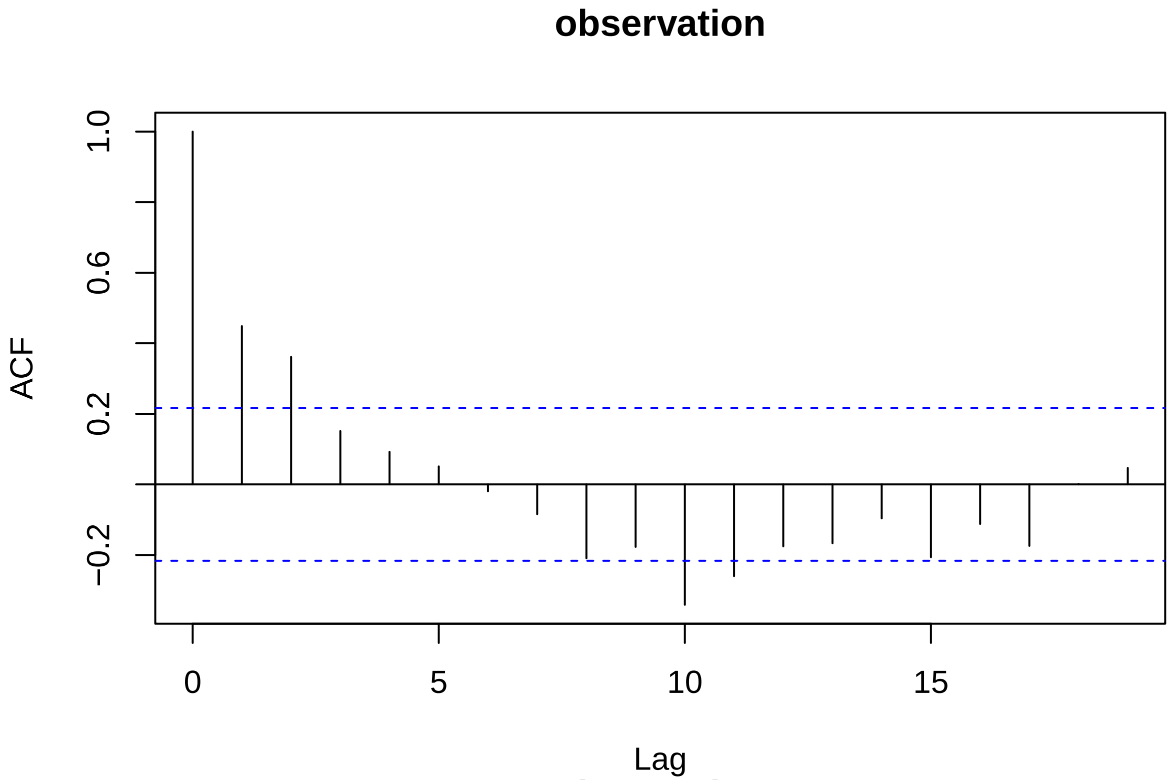
observation

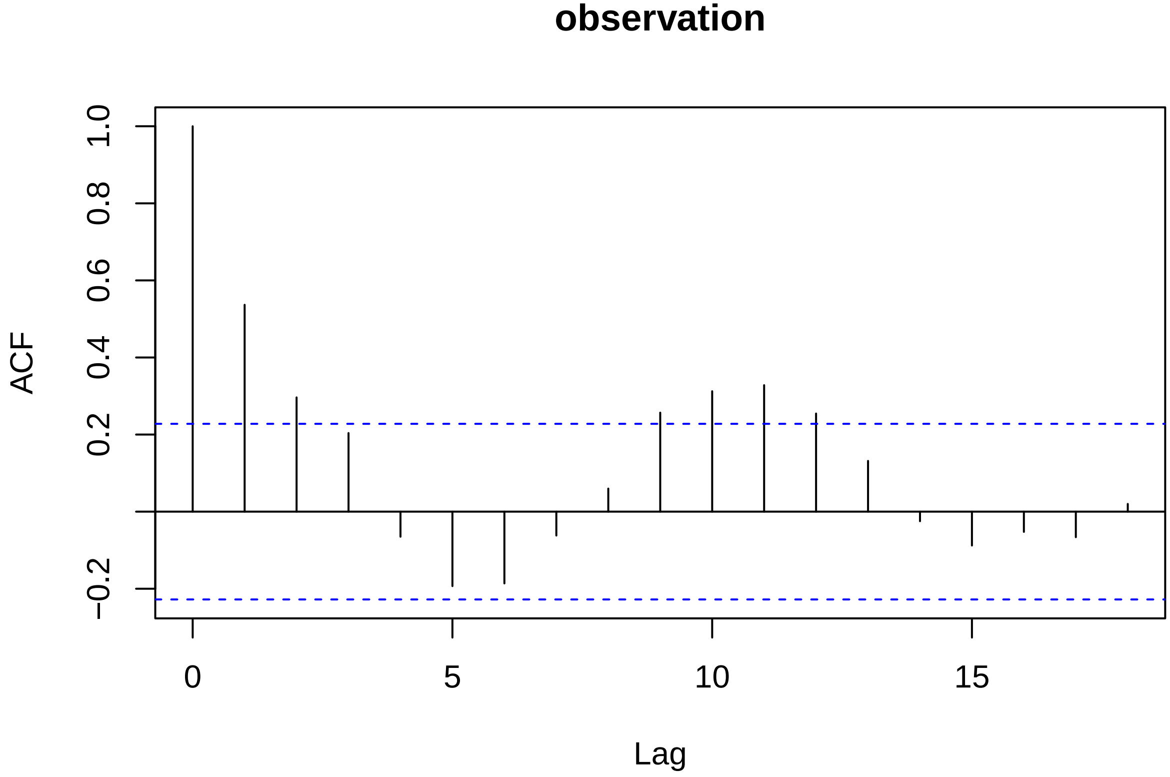


observation


observation
ACF
1.0
0.8
0.6
0.4
0.2
−0.2
0
5
10
15
Lag


observation
ACF
1.0
0.5
0.0
−0.5
0
5
10
15
Lag

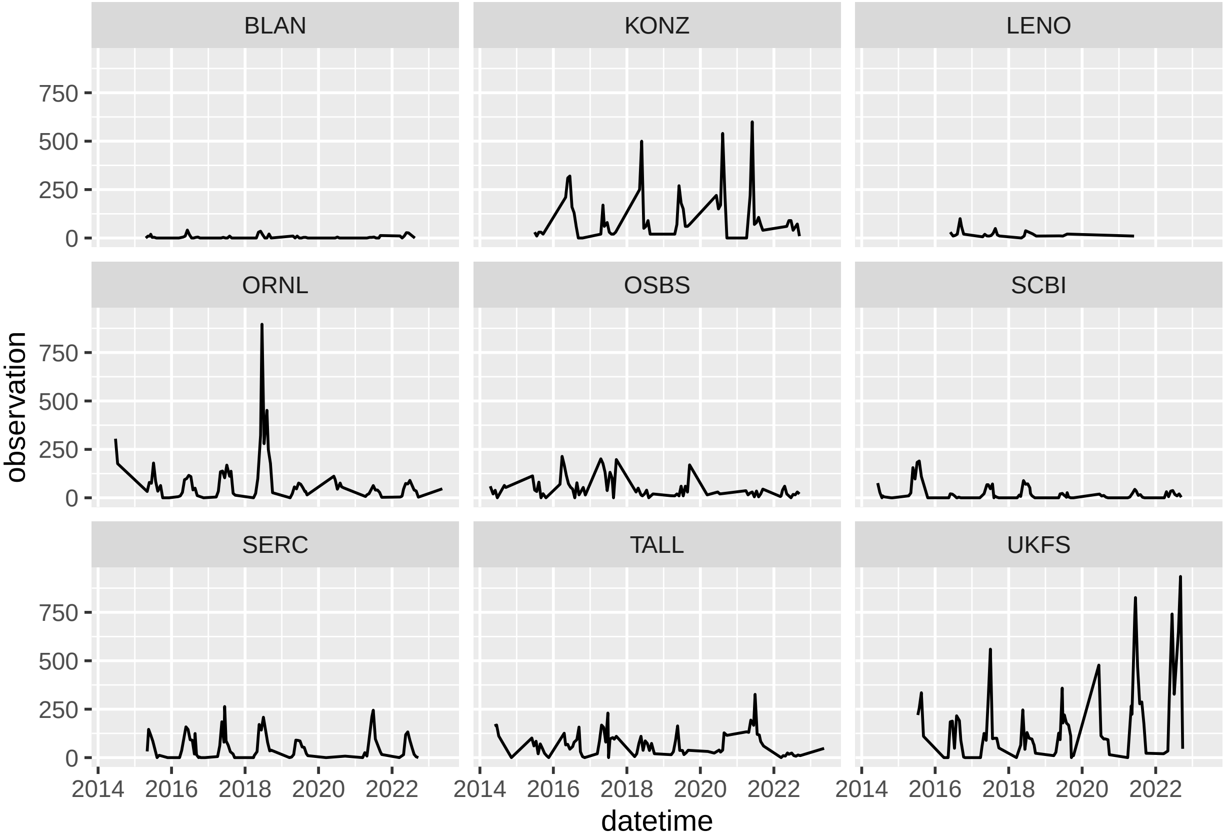




The main takeaways are even though that the data are taken over nine years (not necessarily on a weekly basis but the data has been grouped by the weekly mean observations), it is not equally distributed between all the nine sites present. Additionally, the data are pretty sparse and it is not until grouped by month that missing data decreases significantly (this sparseness thus affects monthly CV). Finally, looking at the ACFs for each of the nine sites, we can see that periodicity is present in some of them and there is mild drift also present in some of the sites.

2) Plot the data [1 point].

```
tick_data |> ggplot(aes(datetime, observation)) + geom_line() + facet_wrap(~site_id)
```



Above we have the data plotted for each of the nine sites. We can start to see a little bit of a seasonal pattern and we see that there are varying degrees of abundance across the different sites. It will be interesting to dive deeper into abundance prediction for each of the sites and how they differ in terms of strength of covariates.

3) What is your main research question? Do you have any working hypothesis? [1 point].

My main research question is to look at how the tick population of Amblyomma americanum changes over time in different sites. How do different sites differ in their tick populations and how do weather covariates play different roles in different sites? My working hypothesis is that precipitation and temperature will have a strong influence on the abundance of ticks in following years.

Any notes (optional)

The data was collected from the NEON Forecast Challenge: https://projects.ecoforecast.org/neon4cast-docs/Ticks.html. That website shows how the data was collected and then preprocessed. In additon, I did not add the weather covariates in the data, but I will be using the dataset also from the NEON Forecast Challenge: https://projects.ecoforecast.org/neon4cast-docs/Shared-Forecast-Drivers.html. They have the following weather covariates I could use in my analysis: air temperature, air pressure, wind speed, precipitation, downwelling longwave radiation, downwelling shortwave radiation, and relative humidity.