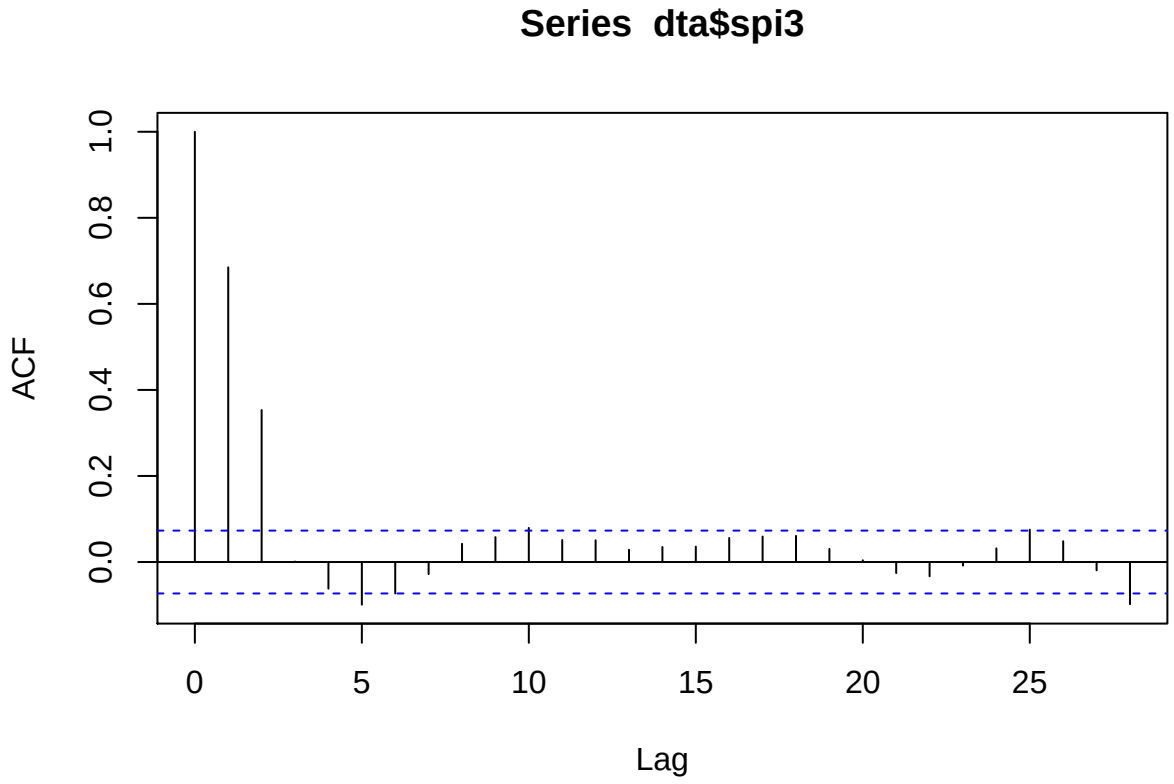
Discussion Regarding ACF Problem

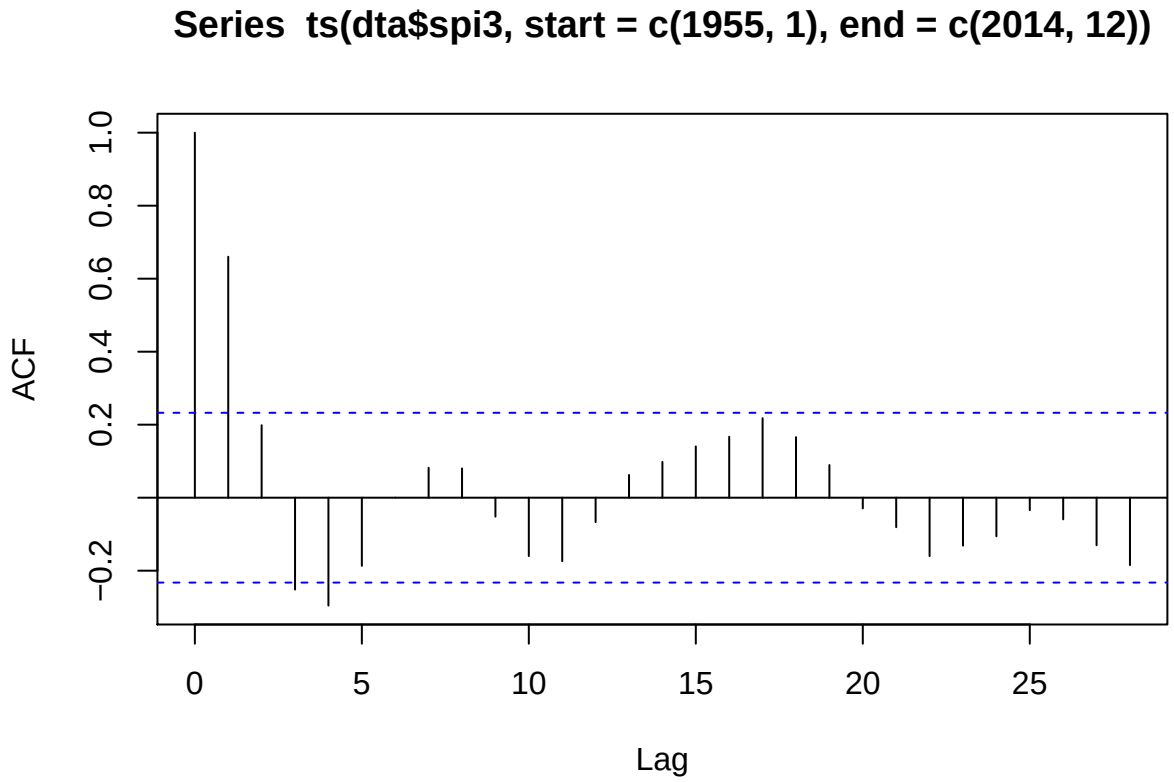
```
library(tidyverse)
data(dta)

# Greg's Approach
acf(
  dta$spi3
) %>% str
```



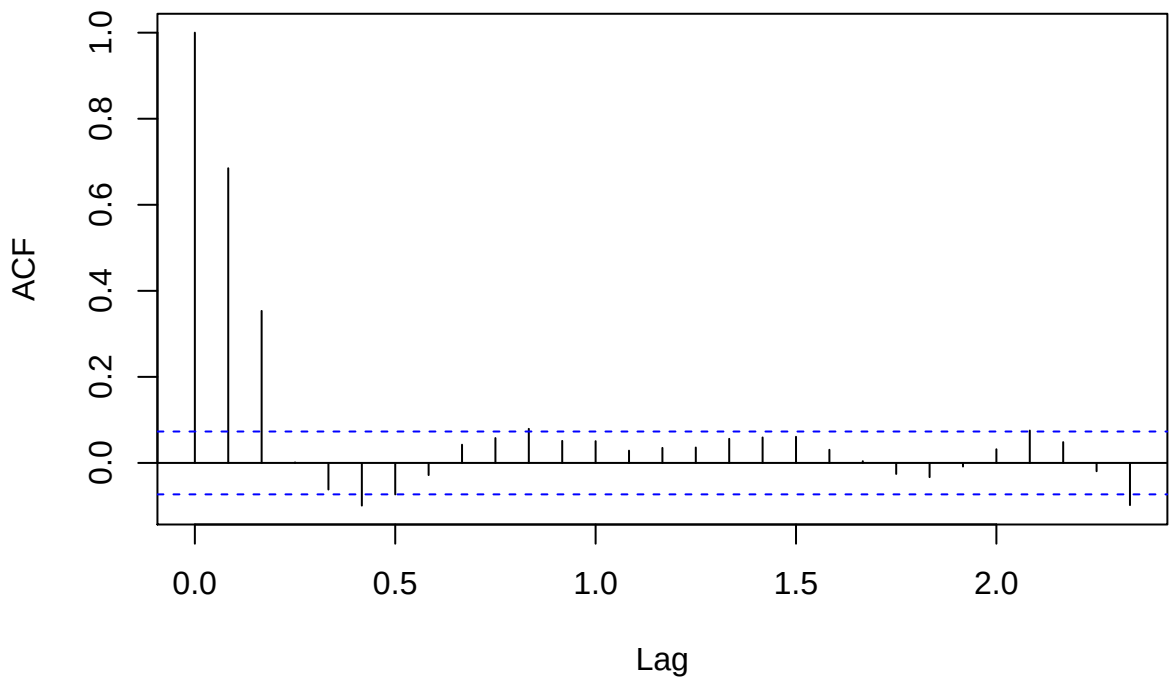

```
## List of 6
##  $ acf    : num [1:29, 1, 1] 1 0.6849 0.3533 0.0012 -0.0618 ...
##  $ type   : chr "correlation"
##  $ n.used: int 720
##  $ lag    : num [1:29, 1, 1] 0 1 2 3 4 5 6 7 8 9 ...
##  $ series: chr "dta$spi3"
##  $ snames: NULL
##  - attr(*, "class")= chr "acf"
```

```
# Brian's Approach - Slightly Modified
acf(
  ts(dta$spi3,
     start = c(1955,1),
     end = c(2014,12)),
  lag.max = 28
) %>% str
```




```
## List of 6
##  $ acf   : num [1:29, 1, 1] 1 0.66 0.199 -0.252 -0.296 ...
##  $ type  : chr "correlation"
##  $ n.used: int 71
##  $ lag   : num [1:29, 1, 1] 0 1 2 3 4 5 6 7 8 9 ...
##  $ series: chr "ts(dta$spi3, start = c(1955, 1), end = c(2014, 12))"
##  $ snames: NULL
##  - attr(*, "class")= chr "acf"
```

```r
# Brian's Approach Modified
acf(
  ts(dta$spi3,
     start = c(1955,1),
     end = c(2014,12),
     freq = 12)
) %>% str
```




```
## List of 6
##  $ acf   : num [1:29, 1, 1] 1 0.6849 0.3533 0.0012 -0.0618 ...
##  $ type  : chr "correlation"
##  $ n.used: int 720
##  $ lag   : num [1:29, 1, 1] 0 0.0833 0.1667 0.25 0.3333 ...
##  $ series: chr "ts(dta$spi3, start = c(1955, 1), end = c(2014, 12), freq = 12)"
##  $ snames: NULL
##  - attr(*, "class")= chr "acf"
```

We see that when we modified the second approach to define the frequency, or number of observations per unit of time (i.e., the first element of the start and end vectors), the approaches are identical. At issue is how the **ts** function works in conjunction with the **acf** function in R. Most specifically, when you pass an object to the **ts** function with start and end times but no frequency (and really in any case for that matter), the function truncates the number of observations in the series by the following code from the **ts** function:

```
if (length(start) > 1L) {
    start <- start[1L] + (start[2L] - 1)/frequency
}
if (length(end) > 1L) {
    end <- end[1L] + (end[2L] - 1)/frequency
}
if (missing(end))
    end <- start + (ndata - 1)/frequency
else if (missing(start))
    start <- end - (ndata - 1)/frequency
if (start > end)
    stop("'start' cannot be after 'end'")
nobs <- floor((end - start) * frequency + 1.01)
if (nobs != ndata)
    data <- if (NCOL(data) == 1) {
```

```
        if (ndata < nobs)
            rep_len(data, nobs)
        else if (ndata > nobs)
            data[1L:nobs]
    }
    else {
        if (ndata < nobs)
            data[rep_len(1L:ndata, nobs), ]
        else if (ndata > nobs)
            data[1L:nobs, ]
    }
```

As a result, we end up with a series of length 71 from 1955 to 2025 after R has coerced a redefintion of the start and end timeframes and number of observations.

All that being said, and because we most certainly do not want to truncate the series as such, the first or modified **ts** option appear to be correct.