Time series and interactive visualization

23 marks

You will need the interactive visualization system called `loon`. If you have not already installed it, do so from CRAN.

a. In this part, you will be using three different `R` functions, `paste()`, `rev()`, and `rep()`, to construct different character vectors (vectors whose elements are `"strings"`. Look at the `help()` for each of these to better understand how to use them. You will also use some existing vectors, namely `LETTERS`, `letters`, and `month.name` (just print these to see their values).

   i. *2 marks* Using the above functions and data structures, produce a vector with the following contents:

```
## [1] "A.z" "B.y" "C.x" "D.w" "E.v" "F.u" "G.t" "H.s" "I.r" "J.q" "K.p" "L.o"
## [13] "M.n" "N.m" "O.l" "P.k" "Q.j" "R.i" "S.h" "T.g" "U.f" "V.e" "W.d" "X.c"
## [25] "Y.b" "Z.a"
```

   Show your code.

```r
content = paste(LETTERS, rev(letters), sep = '.')
content
```

```
##  [1] "A.z" "B.y" "C.x" "D.w" "E.v" "F.u" "G.t" "H.s" "I.r" "J.q" "K.p" "L.o"
## [13] "M.n" "N.m" "O.l" "P.k" "Q.j" "R.i" "S.h" "T.g" "U.f" "V.e" "W.d" "X.c"
## [25] "Y.b" "Z.a"
```

```
ii. *2 marks* Using the above functions and data structures, produce a vector with the following contents:


    ```
    ## [1] "1(a)" "1(b)" "1(c)" "2(a)" "2(b)" "2(c)" "3(a)" "3(b)" "3(c)"
    ```
    Show your code.
```

```r
paste0(rep(1:3, each=3), '(', letters[1:3], ')')
```

```
## [1] "1(a)" "1(b)" "1(c)" "2(a)" "2(b)" "2(c)" "3(a)" "3(b)" "3(c)"
```

~~~
iii. *4 marks* Recall the `getYears()` function from the lecture slides which was used to extract the years

    ```r
    getTSlabels <- function(ts, freqNames) {
                    ... }
    ~~~
    so that

    ```r
    head(getTSlabels(co2, month.abb))
    ```
    returns

    ```
    ## [1] "Jan 1959" "Feb 1959" "Mar 1959" "Apr 1959" "May 1959" "Jun 1959"
    ```
    and
```
```

~~~
```r
tail(getTSlabels(co2, month.abb))
```

returns


```
## [1] "Jul 1997" "Aug 1997" "Sep 1997" "Oct 1997" "Nov 1997" "Dec 1997"
```
The function should be using `paste()`, `rep()`, etc.  You will also want to use `frequency()`.  Show y
~~~

```r
getYears = function(ts){unique(floor(time(ts)))}

getTSlabels <- function(ts, freqNames) {
    years = getYears(ts)
    if (length(freqNames) != frequency(ts)){
        stop()
    }
    paste(freqNames, rep(years, each=frequency(ts)))
}

labs = getTSlabels(co2, month.abb)
head(labs)
```

```
## [1] "Jan 1959" "Feb 1959" "Mar 1959" "Apr 1959" "May 1959" "Jun 1959"
```

```r
tail(labs)
```

```
## [1] "Jul 1997" "Aug 1997" "Sep 1997" "Oct 1997" "Nov 1997" "Dec 1997"
```

b. In this question you are going to interactively explore the co2 data.

   Construct the interactive plot

   ```r
   library(loon)
   pointlabels <- getTSlabels(co2, month.abb)
   decomp <- stl(co2, s.window = 6, s.degree = 1)
   p_stl <- l_plot(decomp, linkingGroup = "co2",
                   itemLabel = pointlabels,
                   showItemLabels = TRUE)
   ```

   i. *2 marks* Click on the plot of residuals. Using only the mouse, identify the month and year of the most negative residual. Identify the month and year of the most positive residual. Report your answers. (It might be helpful to zoom in on the residuals using CMD/ALT and scrolling; select scale to plot in the inspector to get back to original.)

   minimum residue: at April 1971, largest residue: at March 1985

   ii. *1 mark* You can select a large number of points by clicking on the plot background and moving the mouse (while the left mouse button is held down). This is called *sweep selection*. Try it on any plot. What happens to the points in the other plots?

   they got highlighted in pink

   iii. *2 marks* Use sweep selection on the plot of residuals to select only the most positive 10-20 residuals. These are places where the fit of the model is worst. Ideally, there should be no obvious pattern in where these poorer fits occur in our model. For example, the poor fit might be only at the beginning of each season, or at the seasonal extremes. Comment on whether there are any such obvious patterns here.

before year 1972, most poor fits appear at seaonal extremes. around year 1973, there are a lots of poor fits than other times. other than that, there is no obvious pattern.

iv. Click on the plot of the seasonal component; its picture should now appear in the inspector window. In the inspector window, click on the "Layers" tab and once there select the label "line". With "line" selected, go to the bottom ans click on a button that looks like an eye with a stroke through it; this will turn off the lines in the plot. Select the "Analysis" tab to get back to where you were.

   Now click again on the plot of the seasonal component. While holding the CTRL key down, scroll so that the series shrinks and expands only **horizontally**.

   Shrink the plot horizontally until separate curves are distinguishable.

   It might be helful to construct the following plot *as well*:

```
seasonal <- decomp$time.series[, "seasonal"]
p <- l_plot(seasonal, linkingGroup = "co2", ylabel = "seasonal estimate",
            itemLabel = pointlabels, showItemLabels = TRUE)
```

- *2 marks* What does the highest curve represent? The lowest? What does each curve represent?

highest curve: (weighted estimate of) co2 level of the same month(the month with largest average co2 level) through the years

lowest curve:(weighted estimate of) co2 level of the same month(the month with lowest average co2 level) through the years

each curve: (weighted estimate of) co2 level of the a month through the years

- *2 marks* Recall how the `stl` decomposition is constructed. Explain how each of these curves was constructed and how the seasonal pattern is constructed. Horizontal zooming will help you understand these ideas.

how a curve i is constructed:

it is constructed by 38 points, each representing the leoss regression estimate of co2 level of month i at year j (j ranges from first year to last year) the less regression estimate of co2 level at month i year j depend on weighted average of all co2 level at month i (from first year to last year).

The seaonal pattern is constructed by connecting the the dots by another way, specifically, connect the dots that are cloest in time. for example, connect co2 level at "year 1 month 11" to "year 1 month 12" to "year 2 month 1". then we will be seeing a sin wave.

c. The sunspot data. Consider the following interactive plot:

```
p <- l_plot(sunspot.month,
            ylabel="Mean sqrt monthly sunspots",
            xlabel="Time",
            title="Sunspot activity",
            linkingGroup="sunspots",
            size=1,
            showGuides=TRUE)
l_layer_line(p, sunspot.month)
```

There are 24 *complete* cycles in this data if we measure a cycle as being peak to peak. The objective of this question is to capture the lengths of these cycles.

Using *sweep selection*, an entire cycle can be selected at once (from peak to peak). To record the values, we construct a numeric vector in `R`

```
cycle_lengths <- numeric()
```

and note that

```
sum(p["selected"])
```

will at anytime return the count of the number of selected points.

You could do these for each cycle and record the count, OR, you could accumulate all 24 by executing the following

```
cycle_lengths <- c(cycle_lengths, sum(p["selected"]))
save = cycle_lengths
length(save)
temp = 0
save[1]
for (i in 1:24){
save[i]=save[i]-temp
temp=temp+save[i]
i=i+1
}
```

immediately after each new cycle has been selected on the plot `p`.

Accumulate the 24 cycle lengths (each being a “V” in the plot `p`) in this way by selecting the “V”s left to right one after the other in the plot `p`.

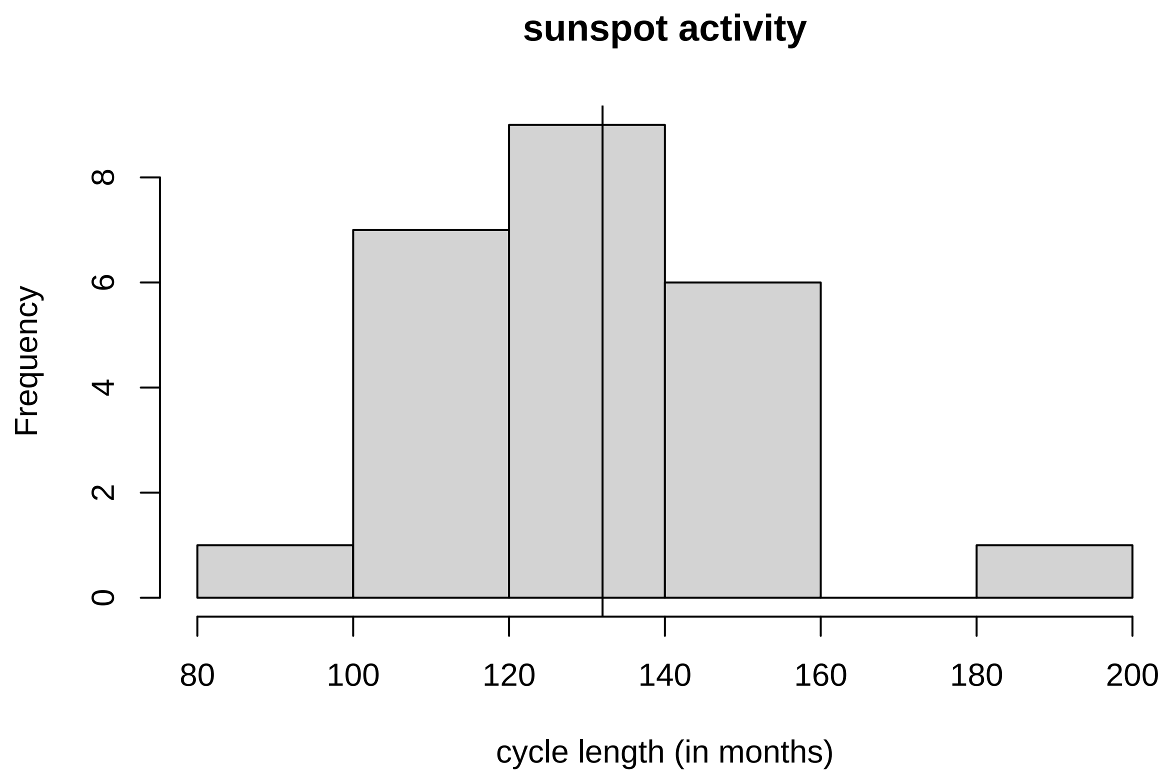
When you have all 24 cycle lengths,

- *1 mark* print them;

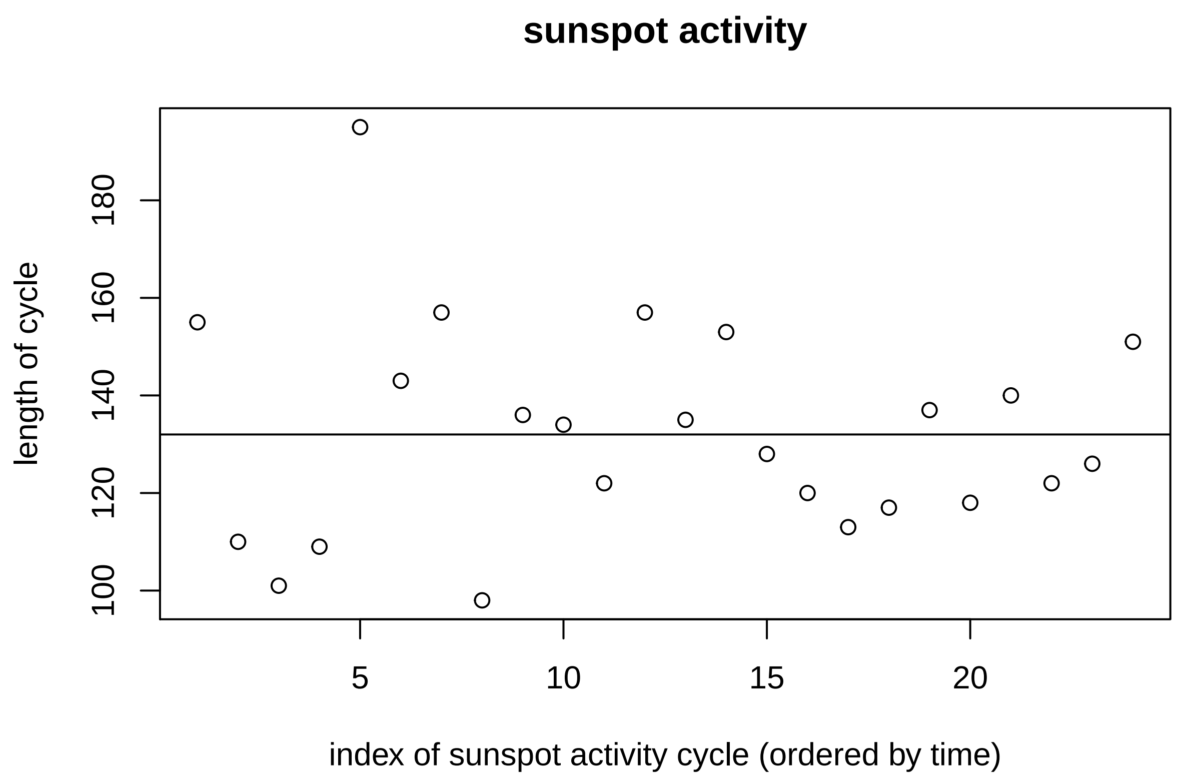
155 110 101 109 195 143 157 98 136 134 122 157 135 153 128 120 113 117 137 118 140 122 126 151

- *1 mark* plot (and hand in) a histogram of the `cycle_lengths` and mark on the histogram with a vertical line (see `abline()`) the 11 year cycle length;

```
cyclen=c(155, 110, 101, 109, 195, 143, 157,  98, 136, 134, 122, 157, 135, 153, 128, 120, 113, 117, 137, 118
hist(cyclen, xlab="cycle length (in months)", main="sunspot activity")
abline(v=12*11)
```



```r
plot(1:length(cyclen), cyclen, xlab = "index of sunspot activity cycle (ordered by time)", ylab="length of
abline(h=11*12)
```



- *1 mark* plot (and hand in) the cycle lengths over time (again with the 11 year cycle marked, now as a horizontal line)

```
- *3 marks* Comment on your findings about the cycle length of sunspots.

cycle length's 1d distribution is clustered around 11 years, roughly symmetric with some but not
a lot extreme values. from its 2d distribution, we see little correlation of cycle length with time
```