Statistical computing MATH10093
Computer lab 7
Solutions

Finn Lindgren

13/3/2019

Summary

In this lab session you will develop code for bootstrap estimation of confidence intervals for probabilities and quantiles.. You will not hand in anything, but you should keep your code script file for later use.

1. Complete lab 6!

2. Initialise lab 7: Open `RStudio` and

   (a) Make sure you have the files from lab 6 in your project folder.

   (b) Open a new R script file for your lab 7 code.

```
# From lab 6
source("lab06code.R")
TMINallobs <- read.csv(file = "data/TMINallobs.csv",
                       header = TRUE,
                       stringsAsFactors = FALSE)
TMINalltest <- read.csv(file = "data/TMINalltest.csv",
                        header = TRUE,
                        stringsAsFactors = FALSE)
```

3. Write a function `boot_resample(data)` that takes input

`data`: A `data.frame` with one observation per row

The output should be a `data.frame` with a bootstrap sample from the rows of `data`.

```
# Solution:
boot_resample <- function(data) {
  data[sample(nrow(data), size = nrow(data), replace = TRUE), , drop = FALSE]
}
# drop=FALSE is a safety feature to make sure the result is still a
# data.frame object. It's not usually needed when doing row indexing
# and the input has multiple columns, but it makes the code more robust.
```

Test the function:

```
data_boot <- boot_resample(TMINallobs)
```

Compare the full data with the subsample; they should be different:

```
# In R, to view the first rows of the data frames:
head(TMINallobs)
head(data_boot)
```

```
# In Rmd or Rnw, with library(xtable) and code chunk option results="asis":
print(xtable(head(TMINallobs)), size = "\\scriptsize")
```

	ID	Year	Month	Element	Day	Value	DecYear	Latitude	Longitude	Elevation	Name
1	UKE00105875	1960	1	TMIN	2	-6.70	1960.00	57.04	-3.22	283	BALMORAL
2	UKE00105875	1960	1	TMIN	3	-3.90	1960.01	57.04	-3.22	283	BALMORAL
3	UKE00105875	1960	1	TMIN	7	-3.30	1960.02	57.04	-3.22	283	BALMORAL
4	UKE00105875	1960	1	TMIN	9	-1.10	1960.02	57.04	-3.22	283	BALMORAL
5	UKE00105875	1960	1	TMIN	10	-7.20	1960.02	57.04	-3.22	283	BALMORAL
6	UKE00105875	1960	1	TMIN	11	0.60	1960.03	57.04	-3.22	283	BALMORAL

```
print(xtable(head(data_boot)), size = "\\scriptsize")
```

	ID	Year	Month	Element	Day	Value	DecYear	Latitude	Longitude	Elevation	Name
69207	UKE00105888	1978	4	TMIN	21	5.60	1978.30	55.97	-3.21	26	EDINBURGH
84075	UKE00105930	1974	5	TMIN	16	9.60	1974.37	56.03	-4.99	12	BENMORE:
73055	UKE00105888	1992	2	TMIN	5	2.50	1992.10	55.97	-3.21	26	EDINBURGH
85068	UKE00105930	1977	10	TMIN	14	9.00	1977.78	56.03	-4.99	12	BENMORE:
43823	UKE00105886	2001	11	TMIN	25	9.90	2001.90	56.38	-2.86	10	LEUCHARS
15972	UKE00105875	2016	11	TMIN	1	2.40	2016.83	57.04	-3.22	283	BALMORAL

4. We are interested in the probability of freezing weather at Balmoral, i.e. $\theta = \mathrm{P}(Y < 0)$. A simple estimator is $\widehat{\theta} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{Y_i < 0\}$.

   (a) Compute an estimate $\widehat{\theta}$ of θ when the population of interest is when we pick a random day of the year (i.e. similar to CWA Q1 where we did not take seasons into account; here we also ignore the climate change issue).

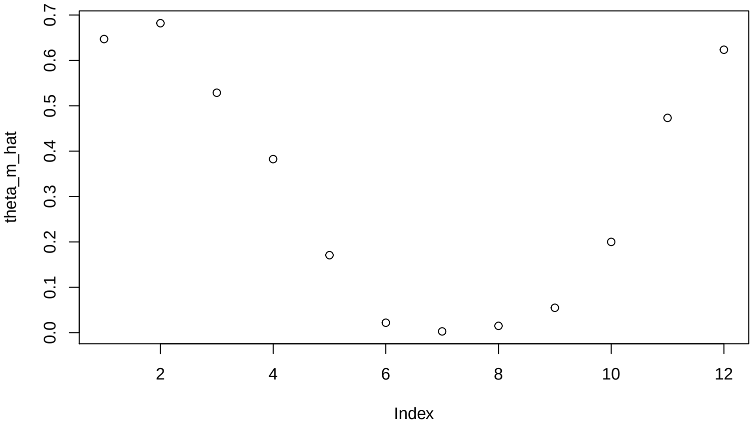
   ```
   # Solution:
   balmoral <- TMINallobs %>% filter(ID == "UKE00105875")
   theta_hat <- mean(balmoral$Value < 0)
   theta_hat

   ## [1] 0.3073125
   ```

   (b) Compute a vector of 12 monthly estimates $\widehat{\theta}_m$ of the monthly probabilities θ_m, $m = 1, \ldots, 12$, to see the seasonal variation.

```
# Solution:
theta_m_hat <- numeric(12)
for (m in 1:12) {
  theta_m_hat[m] <- mean((balmoral %>% filter(Month == m))$Value < 0)
}
theta_m_hat
```

```
##  [1] 0.64705882 0.68200161 0.52870992 0.38252570 0.17081851 0.02190332
##  [7] 0.00278940 0.01502146 0.05482456 0.20000000 0.47331240 0.62352013
```

```
plot(theta_m_hat)
```



The tidyverse filter() function can be used to construct the needed data subsets (based on station, and the also on month). Create a new data.frame called balmoral that contains only the data from that station, so that you don't need to filter on that everywhere.

```
# Example of filter() use; extract the data from the first of all months, for Balmoral.
# The "pipe operator" "%>%" is helpful for structuring this kind of data wrangling,
# where the result of one operation is used (silently) as the first parameter of
# the next operation.
library(tidyverse) # You only need this line once in your script file
TMINallobs %>%
  filter(ID == "UKE00105875", Day == 1) %>%
  as.data.frame() %>%
  head()
```

```
##            ID Year Month Element Day Value  DecYear Latitude Longitude
## 1 UKE00105875 1960     2    TMIN   1   2.2 1960.085  57.0367     -3.22
## 2 UKE00105875 1960     4    TMIN   1  -0.6 1960.249  57.0367     -3.22
## 3 UKE00105875 1960     5    TMIN   1  -5.6 1960.331  57.0367     -3.22
## 4 UKE00105875 1960     6    TMIN   1   8.3 1960.415  57.0367     -3.22
## 5 UKE00105875 1960     7    TMIN   1   4.4 1960.497  57.0367     -3.22
## 6 UKE00105875 1960     8    TMIN   1   3.3 1960.582  57.0367     -3.22
##   Elevation     Name
## 1       283 BALMORAL
## 2       283 BALMORAL
## 3       283 BALMORAL
## 4       283 BALMORAL
## 5       283 BALMORAL
## 6       283 BALMORAL
```

5. Construct a bootstrap distribution of θ estimates, $\{\widehat{\theta}^{(1)}, \ldots, \theta^{(J)}\}$, $J = 1000$. Recall your `boot_resample()` function:

```
boot_resample(TMINallobs %>%
                filter(...))
```

```
# Solution:
J <- 1000
theta_hat_boot <- numeric(J)
for (j in 1:J) {
  theta_hat_boot[j] <- mean(boot_resample(balmoral)$Value < 0)
}
```

The result should look something like this:

```
summary(theta_hat_boot)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2953  0.3047  0.3071  0.3072  0.3096  0.3205
```

6. Construct a 95% bootstrap confidence interval for θ. See Lecture 6. The `quantile()` function is helpful.

```
theta_CI <- theta_hat - quantile(theta_hat_boot - theta_hat, probs = c(0.975, 0.025))
theta_CI

##     97.5%      2.5%
## 0.3002469 0.3144375
```

7. Construct bootstrap distributions for the monthly estimates of θ_m, with one set $\{\widehat{\theta}_m^{(1)}, \ldots, \theta_m^{(J)}\}$ for each month. Store the results in a `data.frame` with one column for each month. You can reduce J to 250 to save computing time in the lab.

```r
# Solution:
J <- 250
theta_m_hat_boot <- data.frame()
months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
for (m in 1:12) {
  for (j in 1:J) {
    theta_m_hat_boot[j,months[m]] <-
      mean(boot_resample(balmoral %>%
                           filter(Month == m))$Value < 0)
  }
}
```

8. Construct 95% bootstrap confidence intervals for each θ_m. Store the CI:s in a matrix with 12 rows and two columns (one column each for the left and right interval endpoints).

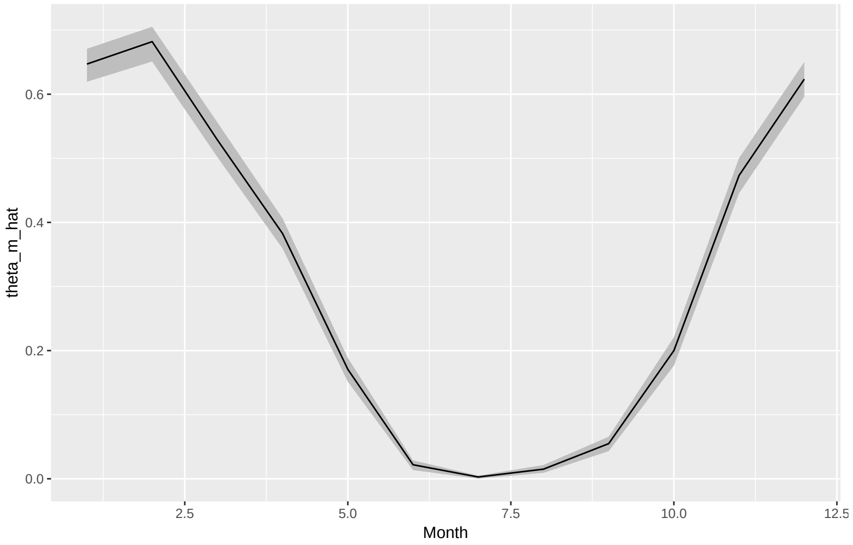
```r
# Solution:
theta_m_CI <- matrix(0, 12,2)
for (m in 1:12) {
  theta_m_CI[m,] <- theta_m_hat[m] - quantile(theta_m_hat_boot[,m] - theta_m_hat[m],
                                              probs = c(0.975, 0.025))
}
theta_m_CI
```

```
##               [,1]       [,2]
##  [1,] 0.6194157393 0.67090620
##  [2,] 0.6510694108 0.70522599
##  [3,] 0.5018642804 0.55596570
##  [4,] 0.3591960352 0.40625918
##  [5,] 0.1515302491 0.18861210
##  [6,] 0.0135951662 0.02870091
##  [7,] 0.0001569038 0.00488145
##  [8,] 0.0092989986 0.02145923
##  [9,] 0.0431286550 0.06562500
## [10,] 0.1764473684 0.22088346
## [11,] 0.4455847724 0.50043171
## [12,] 0.5960734017 0.65035517
```

Plot the results with code similar to this:

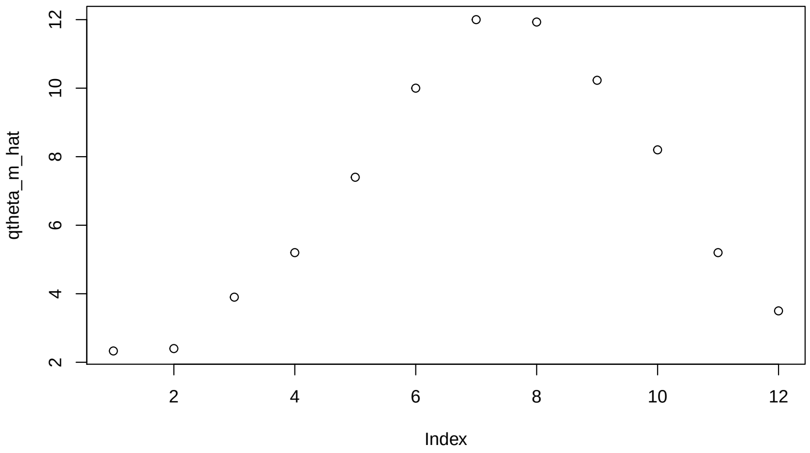
```r
plot(1:12, theta_m_hat, type = "l")
lines(1:12, theta_m_CI[,1], lty = 2)
lines(1:12, theta_m_CI[,2], lty = 2)
```

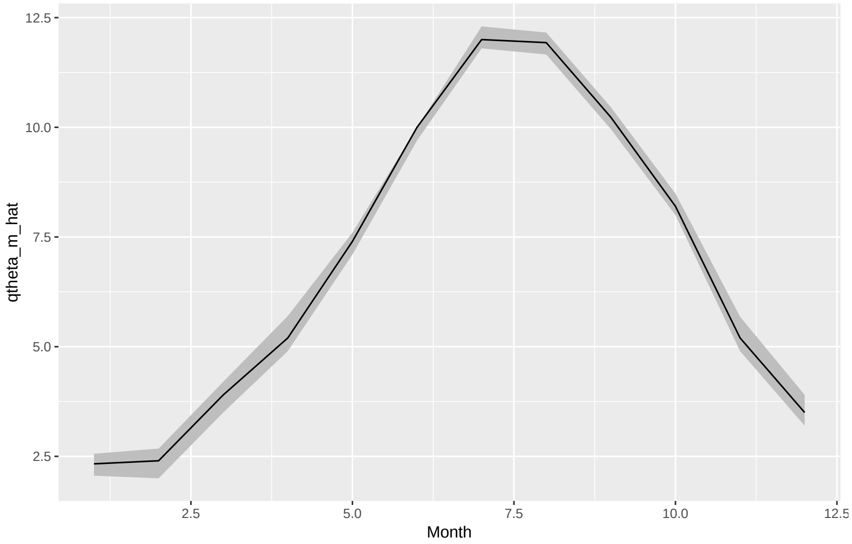
```
library(ggplot2)
ggplot(data = data.frame(Month = 1:12,
                         theta_m_hat = theta_m_hat,
                         CI_lower = theta_m_CI[,1],
                         CI_upper = theta_m_CI[,2])) +
  geom_ribbon(aes(x = Month, ymin = CI_lower, ymax = CI_upper), fill = "Grey") +
  geom_line(aes(Month, theta_m_hat))
```



9. We are now interested in the upper 90% quantiles, i.e. the value θ such that $\mathrm{P}(Y \geq \theta) = 0.9$. Produce bootstrap confidence intervals for the monthly quantiles θ_m.

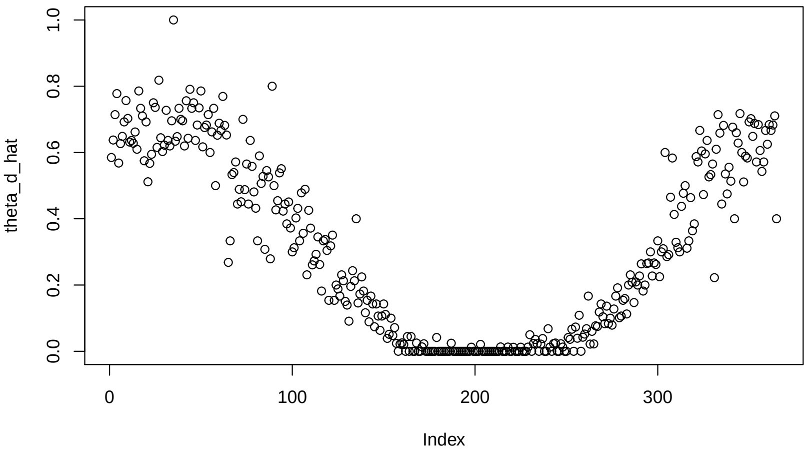
```
# Solution:
qtheta_m_hat <- numeric(12)
for (m in 1:12) {
  qtheta_m_hat[m] <-
    quantile((balmoral %>%
                filter(Month == m))$Value, 0.9)
}
plot(qtheta_m_hat)

# Bootstrap samples:
J <- 250
qtheta_m_hat_boot <- data.frame()
for (m in 1:12) {
  for (j in 1:J) {
    qtheta_m_hat_boot[j,m] <-
      quantile(boot_resample((balmoral %>%
                                filter(Month == m)))$Value, 0.9)
  }
}

# Confidence intervals
qtheta_m_CI <- matrix(0, 12,2)
for (m in 1:12) {
  qtheta_m_CI[m,] <- qtheta_m_hat[m] - quantile(qtheta_m_hat_boot[,m] - qtheta_m_hat[m],
                                                probs = c(0.975, 0.025))
}

# Plot the results
ggplot(data = data.frame(Month = 1:12,
                         qtheta_m_hat = qtheta_m_hat,
                         CI_lower = qtheta_m_CI[,1],
                         CI_upper = qtheta_m_CI[,2])) +
  geom_ribbon(aes(x = Month, ymin = CI_lower, ymax = CI_upper), fill = "Grey") +
  geom_line(aes(Month, qtheta_m_hat))
```

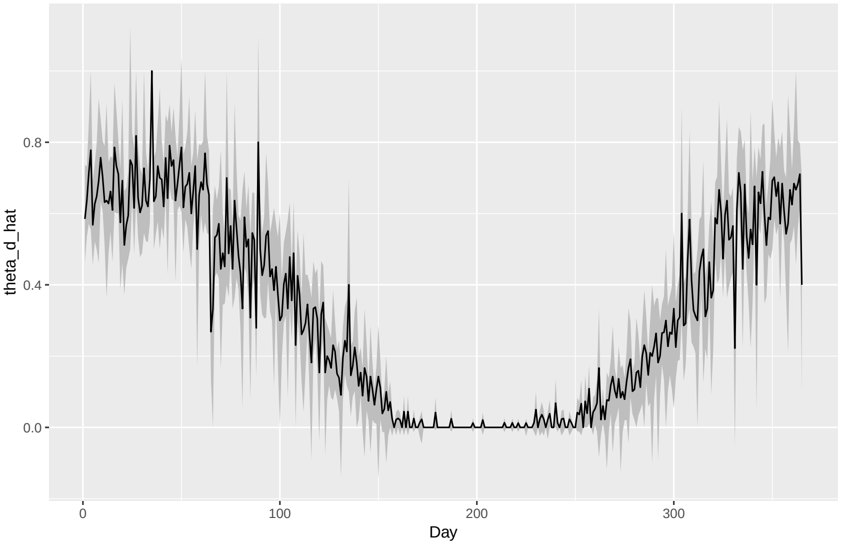




10. Redo task 8 (and the needed previous tasks) for each day of the year instead of each month. To avoid leap year problems, we let $t = 1, \ldots, 365$ and define each data point to belong to day t if `floor(DecYear * 365) %% 365 == t-1`. Reduce J to, for example, $J = 100$ to save computing time in the lab (this may introduce a lot of Monte Carlo variance in the bootstrap estimates, so larger J should be used when possible!).

```
# Solution:
theta_d_hat <- numeric(365)
for (d in 1:365) {
  theta_d_hat[d] <-
    mean((balmoral %>%
            filter(floor(DecYear * 365) %% 365 == d-1))$Value < 0)
}
plot(theta_d_hat)

# Bootstrap samples:
J <- 100
theta_d_hat_boot <- data.frame()
for (d in 1:365) {
  for (j in 1:J) {
    theta_d_hat_boot[j,d] <-
      mean(boot_resample(balmoral %>%
                           filter(floor(DecYear * 365) %% 365 == d-1))$Value < 0)
  }
}

# Confidence intervals
theta_d_CI <- matrix(0, 365,2)
for (d in 1:365) {
  theta_d_CI[d,] <- theta_d_hat[d] - quantile(theta_d_hat_boot[,d] - theta_d_hat[d],
                                              probs = c(0.975, 0.025))
}

# Plot the results
ggplot(data = data.frame(Day = 1:365,
                         theta_d_hat = theta_d_hat,
                         CI_lower = theta_d_CI[,1],
                         CI_upper = theta_d_CI[,2])) +
  geom_ribbon(aes(x = Day, ymin = CI_lower, ymax = CI_upper), fill = "Grey") +
  geom_line(aes(Day, theta_d_hat))
```




theta_d_hat
0.8
0.4
0.0
0
100
200
300
Day