2020년도 1학기 통계프로그래밍언어 연습 문제 (part II)

## Prerequisites

1. rmarkdown, knitr, tidyverse, readxl, RColorBrewer 패키지를 읽어오시오.

## 기말고사 대비 연습문제 part II

1. 다음의 연립 4차원 일차 연립 방정식의 해를 구한 후(a), 도출한 해가 맞는지 확인(b) 하시오. 단, 일차원적인 소거법을 통해 해를 구할 경우 감점 처리

x <- c(3, -4, 3, -2,  
 1, -1, 1, -1,  
 2, -4, 2, -3,  
 2, -1, 1, 3)  
X <- matrix(x, ncol = 4, byrow = TRUE)  
y <- c(8, 4, 1, 5)  
sol <- solve(X) %\*% y; sol # 1-a

[,1]  
[1,] 2.3333333  
[2,] 3.6666667  
[3,] 5.0000000  
[4,] -0.3333333

X %\*% sol # 1-b

[,1]  
[1,] 8  
[2,] 4  
[3,] 1  
[4,] 5

1. 다음과 같은 모수 , 을 갖는 확률밀도함수(probability density function, pdf)가 주어졌을 때,
2. 크기가 인 랜덤표본을 번 반복하여 생성 하시오(단, 재현성을 위해 seed 번호는 20200511로 고정, for loop 문은 사용하지 않음).

n <- 25; N <- 100  
theta1 <- 5; theta2 <- sqrt(3)  
set.seed(2020511)  
xl <- mapply(rnorm,   
 rep(25, N),   
 rep(theta1, N),   
 rep(theta2, N))  
# dim(xl)

1. a)에서 생성한 100번 반복한 25개 표본의 평균 과 표준편차 를 계산 하시오.

mx <- apply(xl, 2, mean)  
sx <- apply(xl, 2, sd)

1. 각 반복에 대한 개별 표본 평균 의 95 % 신뢰구간을 구하시오(단, 모분산이 알려저 있지 않을 때 % 신뢰구간에 대한 공식은 아래와 같음).

여기서 는 자유도가 일 때 분포의 % 분위에 해당하는 값이고, 는 표준편차임. 분포의 분위수 값을 구하는 방법은 help(qnorm) 참고.

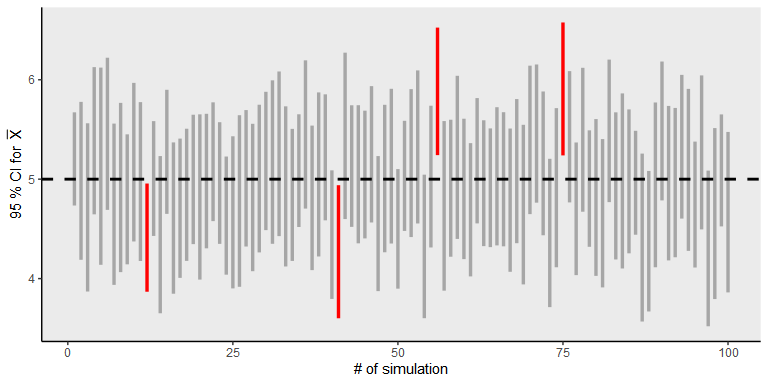
alpha <- 0.05  
upper\_ci <- mx + qt(1 - alpha/2, n-1)\*sx/sqrt(n)  
lower\_ci <- mx - qt(1 - alpha/2, n-1)\*sx/sqrt(n)

1. c)에서 구한 의 95 % 신뢰구간을 이용해 아래와 같은 그래프를 생성 하시오.

**고려 사항**

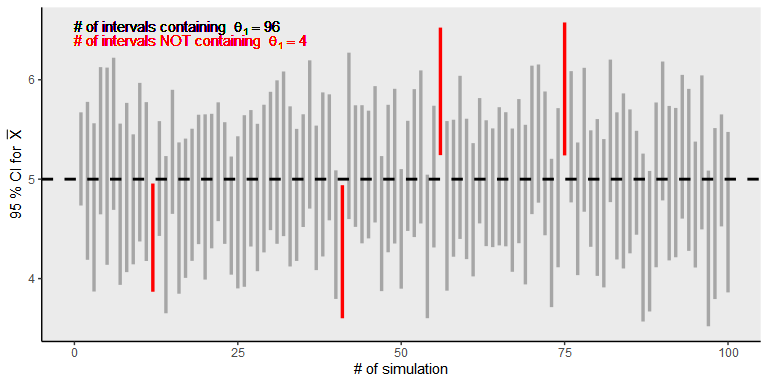
* ggplot 활용
* 을 포함하는 경우 “gray65”, 포함하지 않은 경우 “red” 색상 사용
* c)에서 구한 신뢰구간으로 데이터 프레임(또는 티블 객체 생성)
* 선의 굵기는 1.2로 설정
* y = 0에 해당하는 선은 검정색 점선으로 표시
* x-y 축 선의 굵기는 0.5, 색은 검정색

# 답안  
tibble(upper\_ci, lower\_ci) %>%   
 mutate(nsim = seq\_len(100),   
 contain\_mu = lower\_ci <= theta1 &   
 theta1 <= upper\_ci) -> dfci  
  
gpci <- dfci %>%   
 ggplot(aes(ymin = lower\_ci,   
 ymax = upper\_ci,   
 x = nsim,   
 color = contain\_mu)) +   
 geom\_linerange(size = 1.2) +   
 geom\_hline(yintercept = theta1,   
 color = "black",   
 linetype = "dashed",   
 size = 1) +  
 scale\_color\_manual(values = c("red", "gray65")) +  
 labs(  
 x = "# of simulation",  
 y = expression(paste("95 % CI for" ~~ bar(X)))  
 ) +   
 theme(legend.position = "none",   
 panel.grid = element\_blank(),   
 axis.line = element\_line(size = 0.5, color = "black"))  
gpci



1. 을 포함한 개수와 포함하지 않은 개수를 아래 그림과 같이 표시하시오.

text1 <- bquote(paste(  
 "# of intervals containing",   
 ~~ theta[1]   
 == .(sum(dfci$contain\_mu))))  
text2 <- bquote(paste(  
 "# of intervals NOT containing",   
 ~~ theta[1]   
 == .(sum(!dfci$contain\_mu))))  
lab\_txt <- c(text1, text2)  
col <- c("black", "red")  
  
for (i in 1:2) {  
 gpci <- gpci +   
 geom\_text(aes(x = 0, y = Inf),   
 label = deparse(lab\_txt[[i]]),   
 parse = TRUE,   
 color = col[i],   
 vjust = i + 1,   
 hjust = 0)  
}  
gpci



1. 다음은 [Our World in Data](https://ourworldindata.org/)에서 다운로드한 각 국가별 코로나 19(COVID-19) 관련 데이터에 대한 내용이다.
2. 현재 작업폴더(작업 디렉토리)에 사이버 캠퍼스에 업로드 되어 있는 다음 파일을 다운로드 후 R 작업공간으로 읽어오시오. 각 데이터셋에 대한 코드북은 아래 출력 결과 참조.
   * country\_info.xlsx: 각 국가에 대한 인구학적 정보(총인구수, 인구밀도, 중위 연령 등)
   * country\_pubhealth.csv: 각 국가에 대한 공중보건 지수 정보
   * covid19-cases-20200601.txt: 각 국가별 COVID-19 확진자 및 검사 수 정보

require(kableExtra)  
owid\_codebook <- readxl::read\_excel("d:/Current-Workspace/Lecture/cnu-r-programming-lecture-note/dataset/covid-19-dataset/owid\_covid\_codebook.xlsx")  
covid19 <- read\_delim("d:/Current-Workspace/Lecture/cnu-r-programming-lecture-note/dataset/covid-19-dataset/covid19-cases-20200601.txt",   
 delim = "\t")  
country <- readxl::read\_excel("d:/Current-Workspace/Lecture/cnu-r-programming-lecture-note/dataset/covid-19-dataset/country\_info.xlsx")  
country\_pubhealth <- read\_csv("d:/Current-Workspace/Lecture/cnu-r-programming-lecture-note/dataset/covid-19-dataset/country\_pubhealth.csv")

## # A tibble: 10 x 3  
## Dataset Varialbe Description   
## <chr> <chr> <chr>   
## 1 country iso\_code 3자리 국가코드   
## 2 country location 국가명   
## 3 country region 지역   
## 4 country population 총인구수   
## 5 country population\_density 인구밀도   
## 6 country median\_age 중위연령   
## 7 country aged\_65\_older 65세 이상 인구수  
## 8 country aged\_70\_older 70세 이상 인구수  
## 9 country gdp\_per\_capita 일인당 국민소득   
## 10 country extreme\_poverty 극빈층 비율

## # A tibble: 7 x 3  
## Dataset Variable Description   
## <chr> <chr> <chr>   
## 1 pubhealth location 국가명   
## 2 pubhealth cvd\_death\_rate 심혈관계 질환 사망률   
## 3 pubhealth diabetes\_prevalence 당뇨병 유병률   
## 4 pubhealth female\_smokers 여성 흡연율   
## 5 pubhealth male\_smokers 남성흡연율   
## 6 pubhealth handwashing\_facilities 구내 세수시설 설치 비율  
## 7 pubhealth hospital\_beds\_per\_100k 1000명 당 병상 수

## # A tibble: 9 x 3  
## Dataset Variable Description   
## <chr> <chr> <chr>   
## 1 covid19 iso\_code 3자리 국가코드   
## 2 covid19 date 조사일자   
## 3 covid19 total\_cases 누적(전체) 확진자   
## 4 covid19 new\_cases 신규 확진자 수   
## 5 covid19 total\_deaths 누적(전체) 사망자 수   
## 6 covid19 new\_deaths 신규사망자 수   
## 7 covid19 total\_tests 누적(전체) 검사 수   
## 8 covid19 new\_tests 신규 검사수   
## 9 covid19 new\_tests\_smoothed 신규 검사 수(7일 이동평균 값)

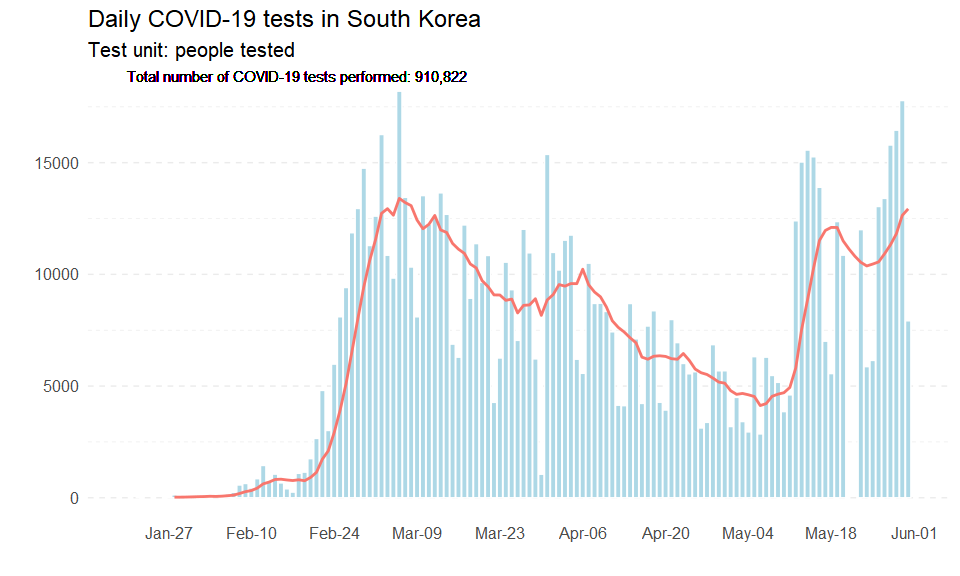
1. 아래 명령을 수행하는 스크립트를 dplyr 및 tidyr 패키지의 동사(함수) 및 파이프 연산자를 이용해 covid19-full 객체에 저장하시오.
   * covid19-cases-20200601.txt 가 저장되어 있는 객체를 기준으로 나머지 두 데이터 셋을 결합(join).
   * as.Date() 함수를 이용해 date가 2020-01-20 부터 2020-05-31 까지 케이스만 추출.
   * 정규표현식을 이용해 region 변수로부터 “Asia”, “Europe”, “America”, “Oceania” 값을 갖는 continent 변수를 생성.
   * 변수명에 “cases”, “deaths”가 포함된 변수들로부터 인구 백만명 당 케이스 및 사망자 수를 저장한 \*\_per\_million 변수 생성. 백만명 당 케이스 및 사망자 수는 를 통해 계산. 여기서 는 “case”, “deaths”가 포함된 변수들임.
   * 각 국가별 최종 총 확진자 수(2020-05-31 기준)이 1000 이상인 케이스만 추출
   * 각 국가별로 total\_tests 에 대해 첫 번째 값이 결측인 경우 0으로, 결측이 아닌 경우 원래 값으로 변환(hint: ifelse() 함수와 행의 색인을 반환해주는 row\_number()라는 함수 이용)
   * total\_cases() 결측값을 이전 시점 값으로 대입(hint: 이러한 결측 대체법을 last observarion carried forward: LOCF라고 하며, tidyr::fill()이라는 함수를 통해 수행 가능)
   * 변수명이 "\_tests" 문자열을 포함한 변수들에 대해 인구 천명 당 검사 수를 저장한 \*\_per\_thousand를 셍성
   * iso\_code, date, location, continent, population, total\_cases, new\_cases, total\_deaths, new\_deaths, total\_tests, new\_tests, new\_tests\_smoothed, total\_cases\_per\_million, new\_cases\_per\_million, total\_deaths\_per\_million, new\_deaths\_per\_million, total\_tests\_per\_thousand, new\_tests\_per\_thousand, new\_tests\_smoothed\_per\_thousand 변수 선택
2. 다음 그래프는 2020년 1월 20일 부터 2020년 5월 31일 까지 대한민국의 일일 COVID-19 진단검사 수(막대도표)와 ggplot()과 적절한 함수들을 이용해 아래와 동일한 도표를 생성하시오.

**Hint**

* 빨강색 선의 굵기(size)는 1이고, 신규 검사수(7일 이동평균) 값임
* 막대의 색은 lightblue이고, 바탕 선의 색은 흰색임.
* x 축의 date 표시는 scale\_x\_date() 함수를 통해 생성할 수 있으며, help() 함수를 통해 사용 방법 익힐 것
* strptime 에 대한 R 제공 도움말을 참고하여 날짜 표기 표준(POSIX) 문자 표기 규칙을 확인. 단 영문으로 월 약어 표기는 Sys.setlocale("LCTIME", "english")을 실행 후 가능
* “Total number ~” 의 텍스트 크기는 4로 조정
* theme\_minimal() 사용하고 기본 크기를 15로 지정하고, 그래프 내 가로 격자선의 굵기는 0.5, 점선(dashed) 사용.

#3-b 답:   
covid19\_full <- covid19 %>%   
 filter(iso\_code != "OWID\_WRL",   
 date > as.Date("2020-01-20") &  
 date < as.Date("2020-06-01")) %>%   
 left\_join(country, by = "iso\_code") %>%   
 left\_join(country\_pubhealth, by = c("location")) %>%   
 mutate(continent = gsub("(.+\\s)", "", region) %>%   
 factor) %>%   
 mutate\_at(vars(matches("cases|deaths")),   
 list(per\_million =~ ./population \* 10^6)) %>%   
 group\_by(location) %>%   
 filter(max(total\_cases) > 1000) %>%   
 mutate(total\_tests = ifelse(row\_number() == 1 &   
 is.na(first(total\_tests)),   
 0, total\_tests)) %>%  
 fill(total\_tests, .direction = "down") %>%   
 ungroup %>%   
 mutate\_at(vars(contains("\_tests")),  
 list(per\_thousand = ~ ./population \* 10^3)) %>%   
 select(iso\_code:date, location, continent, population,   
 matches("cases|deaths|tests"))

# 3-c 답:   
Sys.setlocale("LC\_TIME", "english")  
covid19\_full %>%   
 filter(iso\_code == "KOR") %>%   
 group\_by(iso\_code) %>%   
 mutate(total\_tests\_n = max(total\_tests)) %>%   
 ungroup %>%   
 ggplot(aes(x = date)) +   
 geom\_bar(aes(y = new\_tests),   
 stat = "identity",   
 fill = "lightblue",   
 color = "white") +   
 geom\_line(aes(y = new\_tests\_smoothed,   
 group = location,   
 color = location),   
 size = 1) +   
 scale\_x\_date(date\_breaks = "2 weeks",   
 date\_labels = "%b-%d") +   
 geom\_text(aes(x = as.Date('2020-01-20'), y = Inf,   
 label = paste("Total number of COVID-19 tests performed:",   
 format(unique(total\_tests\_n),   
 big.mark = ","))),   
 vjust = 1,   
 hjust = 0,   
 color = "black",   
 size = 4) +   
 theme\_minimal(base\_size = 15) +   
 labs(x = "",   
 y = "",   
 title = "Daily COVID-19 tests in South Korea",   
 subtitle = "Test unit: people tested") +   
 theme(  
 legend.position = "none",   
 panel.grid = element\_line(size = 0.5, linetype = "dashed"),   
 panel.grid.minor.x = element\_blank(),   
 panel.grid.major.x = element\_blank(),   
 # axis.text.y = element\_blank(),  
 axis.ticks = element\_blank()  
 )



1. 다음은 2020년 5월 31일 기준 상위 24개 국가의 국가 별 확진자 대비 COVID-19 검사 수(전체, 전체 검사수/전체 확진자 수)를 도식화한 그래프이다. ggplot을 이용해 동일한 그래프를 생성하는 스크립트를 작성 하시오.

**Hint**

* 색상 palatte는 “Set1”을 사용 했으며, 총 색을 24개로 확장
* y 축의 범위는 0 ~ 300
* 막대 옆 표시된 y의 값에 대한 텍스트 크기는 4, 텍스트 위치 조정 시 hjust = -0.5 사용
* c와 동일한 배경 테마 및 격자선 굵기 적용

# 3-d 답:   
require(RColorBrewer)  
getPalette = colorRampPalette(brewer.pal(9, "Set1"))  
covid19\_full %>%  
 filter(date == as.Date("2020-05-31")) %>%  
 mutate(total\_tests\_per\_cases = total\_tests/total\_cases) %>%  
 filter(total\_tests\_per\_cases != 0) %>%  
 select(iso\_code:continent,  
 total\_tests\_per\_cases) %>%  
 arrange(desc(total\_tests\_per\_cases)) %>%   
 slice(1:24) %>%   
 mutate(location = factor(location,  
 levels = unique(location))) %>%  
 ggplot(aes(x = reorder(location, desc(location)),  
 y = total\_tests\_per\_cases)) +  
 geom\_bar(aes(fill = location),  
 stat = "identity") +  
 geom\_text(aes(label = sprintf("%.1f", total\_tests\_per\_cases)),  
 hjust = -0.5,  
 size = 4) +  
 scale\_y\_continuous(expand = c(0, 0),  
 limits = c(0, 300),  
 breaks = c(0, 100, 200)) +  
 scale\_fill\_manual(values = getPalette(24)) +  
 labs(x = "", y = "",  
 title = "Total COVID-19 tests per each confirmed case at May 31, 2020") +  
 coord\_flip() +  
 theme\_minimal(base\_size = 15) +  
 theme(  
 legend.position = "none",  
 panel.grid = element\_line(size = 0.5,  
 linetype = "dashed"),  
 panel.grid.major.y = element\_blank(),  
 panel.grid.minor.y = element\_blank()  
 )

