Exercise sheet 3
Prepare the below such that you are able to discuss it on Wednesday 17th March

Exercise 1 Basic reproduction number

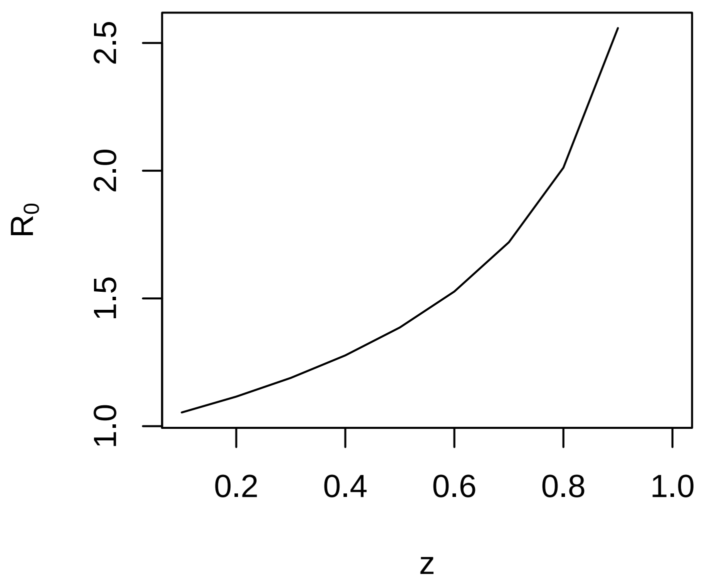
Recall the final outbreak size $Z = Y_1 + Y_2 + \ldots$ from last week's exercises. This is related to the basic reproduction number via the average final fraction infected ($z = Z/N \in [0, 1]$):

$$1 - z = \exp(-R_0 z)$$

a) Solve this expression for R_0 and describe how it relates to z
(*Hint*: A graphical representation may help visualise the results)
Solution: Taking the logarithm on both sides, multiplying by -1, and division by z yields

$$R_0 = -\frac{\log(1 - z)}{z}$$

```r
z <- seq(0.1, 1, 0.1)
rnaught <- function(z){- log(1 - z) / z}
plot(z, rnaught(z), type = "l", ylab = expression("R"["0"]))
```



We see a larger final fraction corresponds to a larger basic reproduction number; a more transmissible outbreak of the disease. Note that now z cannot be 0 (the immediate die out from last week's exercises) and if we were implementing this function for a package we were developing, we may want to restrict R_0 to be 1 for such an input value.

b) Supposing a proportion p of the initial population were immune (through vaccination or natural immunity), we have instead

$$1 - z(1 - p) = \exp(-R_0 z(1 - p))$$

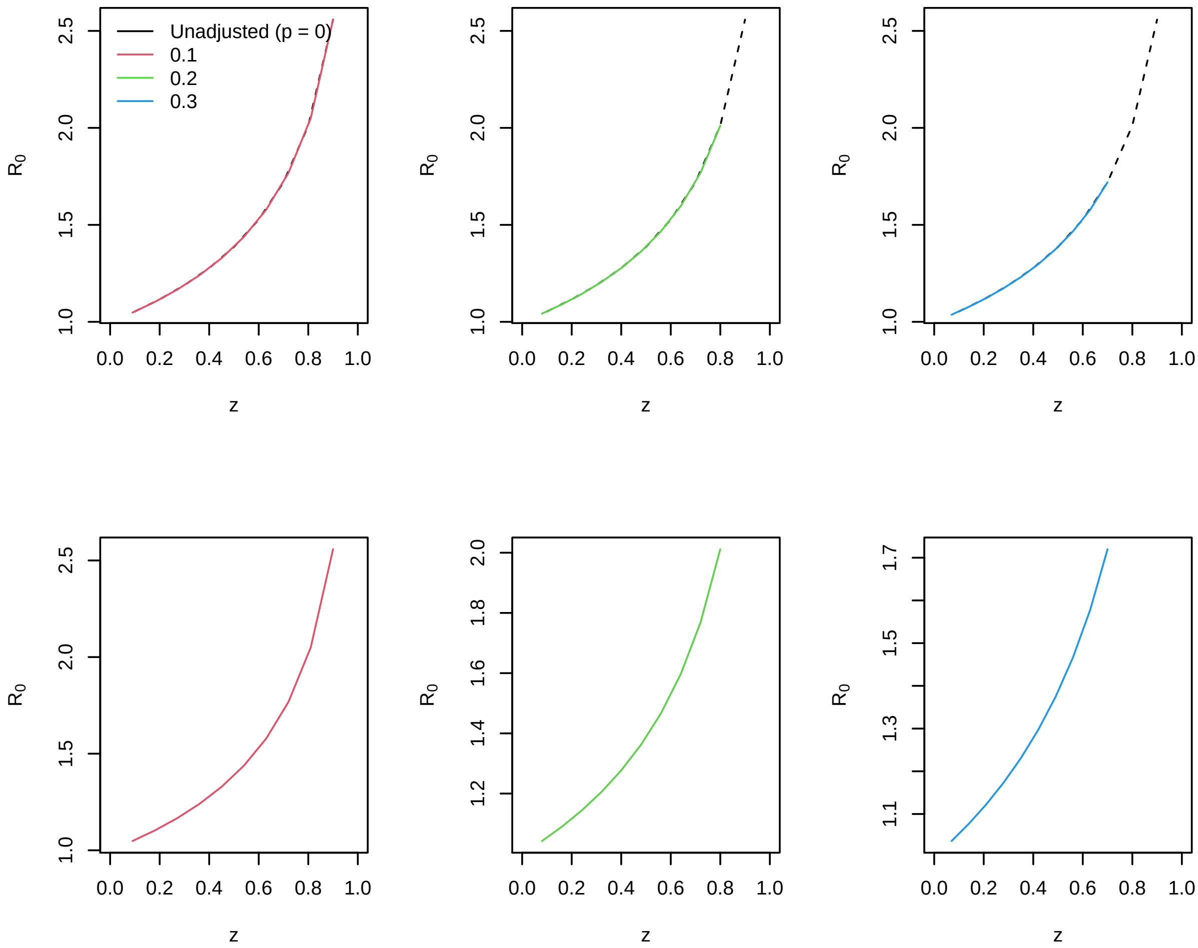
Compute $z(1-p)$ for $p = 10\%, 20\%, 30\%$ and $N = 1000$, $Z = 100, 200, \ldots, 900$. What are the corresponding R_0 estimates?
(*Hint*: A graphical representation may help visualise the results)
Solution: We insert these new values $z^* = z(1-p)$ in the expression from the previous question.

```r
# The z from the previous question can be reused
all.equal(z, seq(100, 1000, 100) / 1000)

## [1] TRUE

p <- c(0.1, 0.2, 0.3)

# The combinations we are considering
mat <- matrix(NA, ncol = 3)
for(i in z){
  mat <- rbind(mat, i * (1 - p))
}
mat <- mat[- is.na(mat), ]
mat <- cbind(z, mat)
```

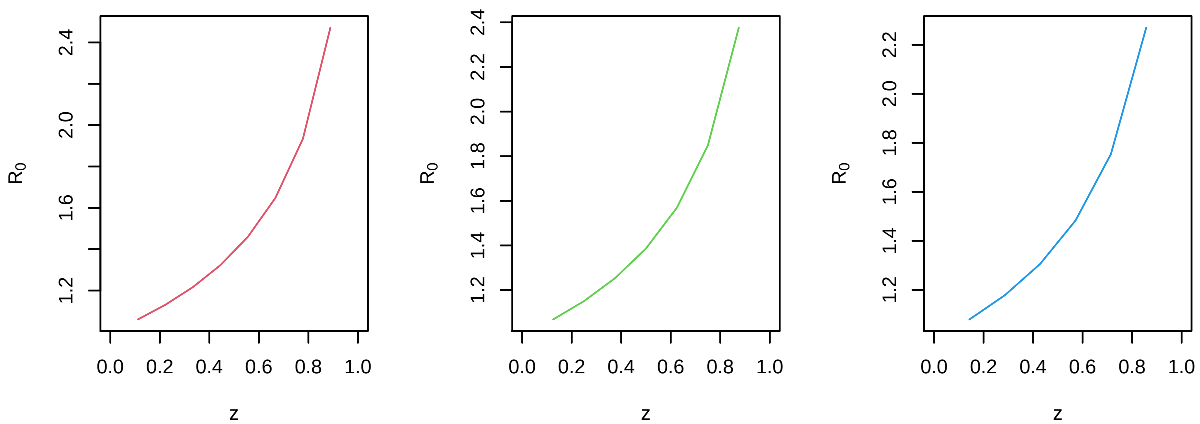


It was suggested to instead scale N by $(1-p)$ and consider the fraction Z/N^*, here are the plots for such an approach:

```r
# The z from the previous question can be reused
Z <- seq(100, 1000, 100)
N <- 1000
N <- (1 - p) * N

# The combinations we are considering
mat <- matrix(NA, ncol = 3, nrow = length(Z))
for(i in seq_along(Z)){
  mat[i, ] <- Z[i] / N
}
# Doesn't make sense to have fraction be greater than 1
mat[mat > 1] <- NA

par(mfrow = c(1, 3))
plot(y = rnaught(mat[, 1]), x = mat[, 1], type = "l", col = 2,
     ylab = expression("R"["0"]), xlab = "z",
     xlim = c(0, 1))
plot(y = rnaught(mat[, 2]), x = mat[, 2], type = "l", col = 3,
     ylab = expression("R"["0"]), xlab = "z",
     xlim = c(0, 1))
plot(y = rnaught(mat[, 3]), x = mat[, 3], type = "l", col = 4,
     ylab = expression("R"["0"]), xlab = "z",
     xlim = c(0, 1))
```



We find that the size of the outbreak (represented by the final fraction) and R_0 decrease with increased immunity (comparison of values using z and z^*).
In summary:

$$\downarrow R_0 \Rightarrow z \downarrow$$
$$\uparrow p \Rightarrow z \downarrow$$

c) Why is this not a useful approach for obtaining an estimate of R_0 for COVID-19?
Solution: The outbreak has not been declared over yet, so we cannot determine the final outbreak size (or final fraction). (Other options for estimating R_0 exist which do not require waiting for the oubreak to finish but they are beyond the scope of this question.)

Good comments on the fact that R_0 is defined in a fully susceptible population were raised!

Exercise 2 Effective reproduction number

Recall the discretised and smoothed instantaneous reproduction numbers

$$R(t_i) = \frac{I_i}{\sum_{j=0}^{n} w_j I_{i-j}} \qquad \text{(discretised)}$$

$$R_\tau(t_i) = \frac{\sum_{k=1-\tau+1}^{i} I_k}{\sum_{k=i-\tau+1}^{i} \sum_{j=0}^{n} w_j I_{k-j}} \qquad \text{(smoothed)}$$

a) Assume a generation time of exactly 4 days. Derive a direct expression for the 7-day smoothed instantaneous reproduction number estimator for this generation time.
Solution: We want a time window of seven days and all probability mass to be on day four, meaning our building blocks are:

$$\tau = 7 \quad \text{and} \quad w_i = \begin{cases} 1 & i = 4 \\ 0 & \text{else} \end{cases}$$

This gives us the expression

$$R_7(t_i) = \frac{\sum_{k=i-7+1}^{i} I_k}{\sum_{k=i-7+1}^{i} I_{k-4}}$$

b) Download the Swiss time series of reported cases from FOPH's COVID-19 website and write a function

```
Rt4 <- function(y, t)
```

which calculates $R(t)$ for the time series using the expression from the previous question
Solution:

```r
# Download the case data
if (!file.exists("data/COVID19Cases_geoRegion.csv")) {
  file_url <- "https://www.covid19.admin.ch/api/data/20210308-nsrvnhng/downloads/sou
  download.file(url = file_url, destfile = "sources-csv.zip", mode = "wb")
  utils::unzip(zipfile = "sources-csv.zip")
}
dat <- read.csv(file = file.path("data", "COVID19Cases_geoRegion.csv"))
dat <- dat[dat$geoRegion == "CH", ]

# datum is t and entries is y
dat <- dat[, c("datum", "entries")]
names(dat)[names(dat) == "entries"] <- "I"
```

We see that at each step, the function should add an additional term to each sum (as i is increased by 1) but the smallest initial term in the sums should also be removed, by virtue of how k is given. There is always seven terms (the difference between k and i) in both parts of the fractions. For the initial steps, $i \leq 7$, some of the terms will not be included as they do not make sense, e.g. counts for negative time I_m where $m < 0$. To illustrate what happens at each step, we show the calcuations for $i = 17$ and $i = 18$. For $i = 17$ we calculate the estimate as

$$\frac{I_{11} + \ldots I_{17}}{I_7 + \ldots + I_{13}}$$

and for $i = 18$ we calculate

$$\frac{I_{12} + \ldots I_{18}}{I_8 + \ldots + I_{14}}$$

```
library(zoo)

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

# Create the function
Rt4 <- function(y, t){
  rollsumr(dat$I, k = 7, fill = NA) /
    rollsumr(lag(zoo(dat$I), -4, na.pad = TRUE), # shift by 4
             k = 7, fill = NA)
}
ests <- Rt4(y = dat$I, t = dat$datum)
```

c) Modify the call to **EpiEstim** used in the slides so it uses this point generation time distribution instead. How does this compare with the results from the previous two questions?

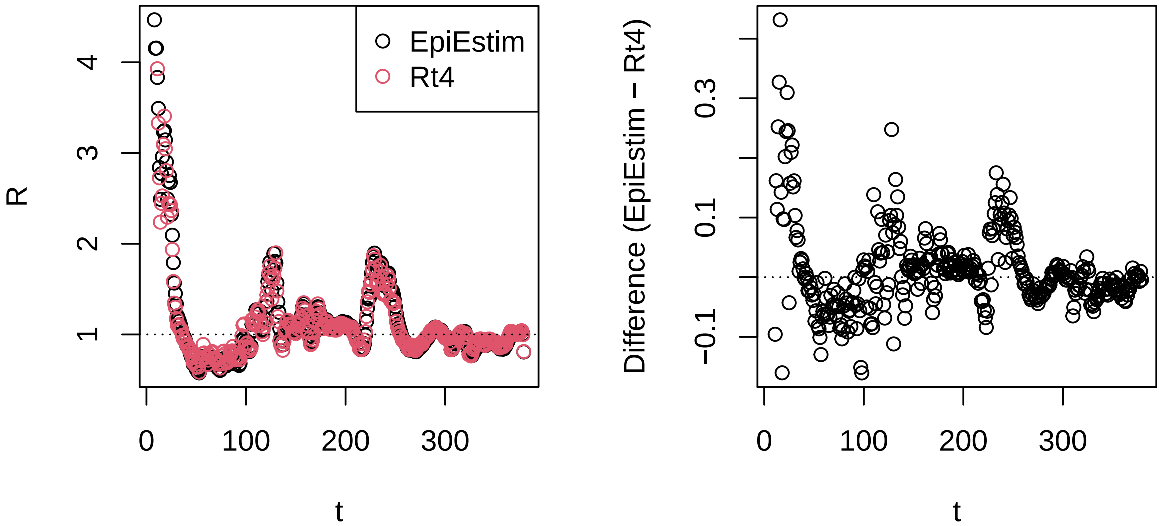
```
# Define generation time to use
GT_pmf <- c(0, 0, 0, 0, 1)
GT_obj <- R0::generation.time("empirical", val = GT_pmf)

# Estimate the instantaneous reproduction number
# Use smoothed version
res <- EpiEstim::estimate_R(dat, method = "parametric_si",
                            config = EpiEstim::make_config(mean_si = 4.8,
                                                           std_si = 2.3,
                                                           t_start = 2 : (nrow(dat)
                                                           t_end = 8 : nrow(dat)))
```

To compare the two approaches we plot them together

```
lngh <- length(8 : nrow(dat))
par(mfrow = c(1, 2))
plot(x = 8 : (length(ests)),
     y = tail(c(NA, res$R$`Mean(R)`), lngh),
     ylab = "R", xlab = "t")
points(x = 8 : (length(ests)),
       y = tail(ests, lngh), col = 2)
```

```
legend("topright", c("EpiEstim", "Rt4"),
       col = 1 : 2, pch = 1)
abline(h = 1, lty = 3)
plot(x = 8 : length(ests),
     res$R$`Mean(R)` - tail(ests, lngh),
     ylab = "Difference (EpiEstim - Rt4)", xlab = "t")
abline(h = 0, lty = 3)
```



d) In the lecture, the effect of underreproting was shown through considering what the effect on $R(t_i)$ is when a fraction $\rho \in [0, 1]$ of cases are reported. What happens to the effective reproduction number if ρ decreases by a factor $r < 1$ each day?

Solution: Recall that if we observe $C_i = \rho I_i$ (ρ is time-constant), we can replace I_i with C_i/ρ and the ρ's cancel out in the discretised expression, and we see the robustness of the estimate. Now, we have something like $\rho(t) = r^t\rho$ and hence

$$R(t_i) = \frac{C_i/r^i\rho}{\sum_{j=0}^{n} w_j C_{i-j}/r^{i-j}\rho} = \frac{C_i}{w_0 C_i + \sum_{j=1}^{n} w_j C_{i-j}/r^{i-j}} \neq \frac{C_i}{\sum_{j=0}^{n} w_j C_{i-j}}$$

giving us a biased estimate. Our estimate of $R(t_i)$ is too large since the denominator is too small

$$w_0 C_i + \sum_{j=1}^{n} w_j C_{i-j}/r^{i-j} < w_0 C_i + \sum_{j=1}^{n} w_j C_{i-j} \tag{1}$$

so when the fraction reported changes we no longer have the $rho(t)$'s cancelling out