Denis Ostroushko - HW6

Question 1

I used code provided by the instructor in their slides as a base for this assignment. All code in the chunk below is utilized to produce estimates and bootstrap standard errors.

```r
covparams <-
  list(covmodels = c(
    A ~ lag1_A + R + lag1_R + baseline_CD + t0,
    R ~ lag1_A + lag1_R + baseline_CD + t0)
    )

ymodel <- Y ~ R +  lag1_A + lag1_R + baseline_CD

intvars <- list('A', 'A', "A", "A", "A")

interventions <- list(list(c(static, c(0, 0, 0, 0, 0))),
                      list(c(static, c(0, 0, 0, 1, 1))),
                      list(c(static, c(0, 0, 1, 1, 1))),
                      list(c(static, c(0, 1, 1, 1, 1))),
                      list(c(static, c(1, 1, 1, 1, 1))))

int_descript <- c('0,0,0,0,0,',
                  '0, 0, 0, 1, 1',
                  "0, 0, 1, 1, 1",
                  "0, 1, 1, 1, 1",
                  "1, 1, 1, 1, 1")

res <-
  gformula(
    obs_data = hiv_data_long,
    id = 'id',
    time_name = 't0',
    covnames = c('A', 'R'),
    outcome_name = 'Y',
    outcome_type = 'continuous_eof',
    covtypes = c('binary', 'binary'),
    histories = c(lagged),
    histvars = list(c('A', 'R')),
    covparams = covparams,
    ymodel = ymodel,
    intvars = intvars,
    interventions = interventions,
    int_descript = int_descript,
    basecovs = c("baseline_CD"),
    nsimul = 10000,
    nsamples = 10,
    parallel = FALSE,
    seed = 1234)
```

I removed all irrelevant output from the g-formula computation output. Only average estimate for non-parametric sample average and all other treatment regimes are given in the table below:

```
        Treatment g-form mean  Mean SE
1:             NP    474.5487 11.71196
2: 0, 0, 0, 0, 0     515.4943 20.28554
3:  0, 0, 0, 1, 1    511.4025 14.60875
4:  0, 0, 1, 1, 1    474.9117 13.83271
5:  0, 1, 1, 1, 1    455.4457 14.51811
6:  1, 1, 1, 1, 1    443.8471 14.99956
```

Question 2

Using R-package software we obtain estimate using IPW.

```
w2 <- ipwtm(exposure = A,
            family = "binomial",
            link = "logit",
            numerator = ~ 1,
            denominator = ~ CD + R,
            id = id,
            type = "all",
            timevar = t0,
            data = hiv_data_long %>% filter(t0 < 4) )
```

Then, I subset the data to have only patients with the desired treatment sequence, and using obtained IPW weights, I create expected average response for a given sequence.

Using bootstrap procedure, I resample the data, re-estimate the weights, and get standard errors for each average. Code for bootstrap is in the appendix. I made sure to resample entire sequences for the same patient to account for correlation within a subject and get approximately correct variance estimator. Results, the final answer, is given below:

```
        trt_year estimate        se
0,0,0,0  0,0,0,0 447.4065  36.78166
0,0,0,1  0,0,0,1 417.7496  34.33983
0,0,1,1  0,0,1,1 463.6620  23.64141
0,1,1,1  0,1,1,1 503.4729  24.56417
1,1,1,1  1,1,1,1 520.8249 148.44654
```

Question 3

3 - A

Uisng fucntion below, we obtain a model estimated by the software:

```
w2 <- ipwtm(
  exposure = transplanted,
  family = "binomial",
  link = "logit",
  numerator = ~ 1,

  denominator = ~ year_round +
    age + sex + edema + serBilir +
    albumin + prothrombin,

  id = id,
  type = "first",
  timevar = year_round,
  data = hw6data
)
```

```
# A tibble: 9 x 5
  term                    estimate std.error statistic  p.value
  <chr>                      <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)               1.55     0.729        2.13 3.29e- 2
2 year_round                0.0836   0.0264       3.16 1.56e- 3
3 age                      -0.0381   0.00544     -7.00 2.53e-12
4 sexmale                   0.254    0.169        1.50 1.34e- 1
5 edemaedema no diuretics   0.862    0.254        3.39 6.99e- 4
6 edemaNo edema             0.544    0.250        2.17 2.99e- 2
7 serBilir                  0.0194   0.0112       1.74 8.27e- 2
8 albumin                  -1.09     0.127       -8.60 8.02e-18
9 prothrombin               0.0828   0.0341       2.43 1.52e- 2
```

3 - B

Average expected number of patients who are alive at the end of the timeline is given below

```
names(results) <- paste0("Transplant at year ", 0:6)
results
```

```
Transplant at year 0 Transplant at year 1 Transplant at year 2
           1.0000000            0.7846457            0.5670584
Transplant at year 3 Transplant at year 4 Transplant at year 5
           0.4024991            0.1973918            0.1440736
Transplant at year 6
           0.2192943
```

3 - C

Code for the bootstrap procedure is given in the appendix. Again, I made sure to resample the data such that entire sequence for the same patient is resampled.

```
                     trt_year  estimate         se
Transplant at year 0        0 1.0000000 0.00000000
Transplant at year 1        1 0.7846457 0.03045258
Transplant at year 2        2 0.5670584 0.03829953
Transplant at year 3        3 0.4024991 0.03540550
Transplant at year 4        4 0.1973918 0.02145902
Transplant at year 5        5 0.1440736 0.01773615
Transplant at year 6        6 0.2192943 0.02231659
```

Appendix

2 - bootstrap

```r
hiv_data_long %>%
  group_by(id) %>%
  reframe(n = n()) %>%
  select(n) %>% summary()

## each subject has 5 ids.
## for bootstrap we need to resample independent data- i.e. independent clusters

B <- 250
res <- matrix(rep(NA, B * 5),
              nrow = B,
              ncol = 5
              )

set.seed(1981)
for(i in 1:B){

  print(i)

  hiv_data_long %>% select(id) %>% unique() -> ids

  ids2 <- ids[sample(1:nrow(ids), replace = T), ] %>% arrange(id) %>%
    group_by(id) %>%
    mutate(sec_id = 1:n()) %>%
    arrange(id, sec_id)

  frame_ <- rbind(ids2, ids2, ids2, ids2, ids2) %>%
    group_by(id, sec_id) %>%
    mutate(t0 = 1:n(),
           t0 = t0 - 1) %>%
    arrange(id, sec_id, t0) %>% ungroup() %>% select(-sec_id)

  boot_df <-
    frame_ %>%
    left_join(hiv_data_long, by = c('id', 't0'))

  boot_df <- boot_df %>% select(A, CD, R, id, t0) %>% filter(t0 < 4)
  boot_df <- as.matrix(boot_df) %>% as.data.frame() # this needs to be done to make sure t

  w2_boot <- ipwtm(exposure = A,
            family = "binomial",
            link = "logit",
            numerator = ~ 1,
            denominator = ~ CD + R,
            id = id,
            type = "all",
            timevar = t0,
            data = boot_df )

  boot_df <-
    boot_df %>%
    filter(t0 < 4) %>%
    mutate(ipw.weights = w2_boot$ipw.weights)
  ### reuse list of ids for each sequence that we had previously, and everything else
  ### used in estimation for problem 2

  res[i,] <- sapply(trt_sequences, function(x){q2_est(data = boot_df, trt_sequence = x)})

}

write_rds(res, "q2_boot.rds")
```

3 - C: bootstrap for point estiamtes

```
B <- 100

res <- matrix(rep(NA, B * length(trt_year_options)
                  ),
              nrow = B,
              ncol = length(trt_year_options)
              )

set.seed(15578)
for(i in 1:B){

  print(i)
```

```
    ## again make sure we resample ids and get all correlated outcome within the same subjec
    hw6data %>% select(id) %>% unique() %>% data.frame()-> ids

    ids2 <- ids[sample(1:nrow(ids), replace = T), ] %>% data.frame() %>%
      {colnames(.) = 'id'; .} %>% arrange(id) %>%
      group_by(id) %>%
      mutate(sec_id = 1:n()) %>%
      arrange(id, sec_id)

    frame_ <- rbind(ids2, ids2, ids2, ids2, ids2, ids2, ids2) %>%
      group_by(id, sec_id) %>%
      mutate(year_round = 1:n(),
             year_round = year_round - 1) %>%
      arrange(id, sec_id, year_round) %>% ungroup() %>% select(-sec_id)

    boot_df <-
      frame_ %>%
      inner_join(hw6data, by = c('id', 'year_round'))

    results_boot <- sapply(trt_year_options, function(x) trt_regime_est(df_raw = boot_df, x_

    res[i,] <- results_boot

  }

  write_rds(res, "q3c_boot.rds")
```