R Assignment 2

Alvaro J. Castro Rivadeneira

November 1, 2021

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'ggdag'

## The following object is masked from 'package:stats':
##
##     filter

## here() starts at /Users/ajcr./OneDrive - University of Massachusetts/micokoch/R/biostats683_causal
```

2 Roadmap Questions

1. Step 3: Observed data & **link to causal model:** Suppose the observed date consist of n independent, identically distributed (i.i.d.) draws of the random variable $O = (W1, W2, A, Y)$.

(a) Specify the link between the SCM and the observed data.

We are asked to assume that the observed data $O = (W1, W2, A, Y)$. were generated by sampling n i.i.d. times from a data generating system compatible with $\mathcal{M}^*$. This provides a link between the causal model $\mathcal{M}^*$ and the observed data O. The distribution of the background variables U and the structural equations F identify the distribution of the endogenous variables X and thus the distribution of the observed data O.

(b) What restrictions, if any, does the SCM place on the set of allowed distributions for the observed data?

We have not placed any restrictions on the statistical model $\mathcal{M}$, which is thereby non-parametric.

(c) What notation do we use to denote the true (but unknown) distribution of the observed data and the statistical model?

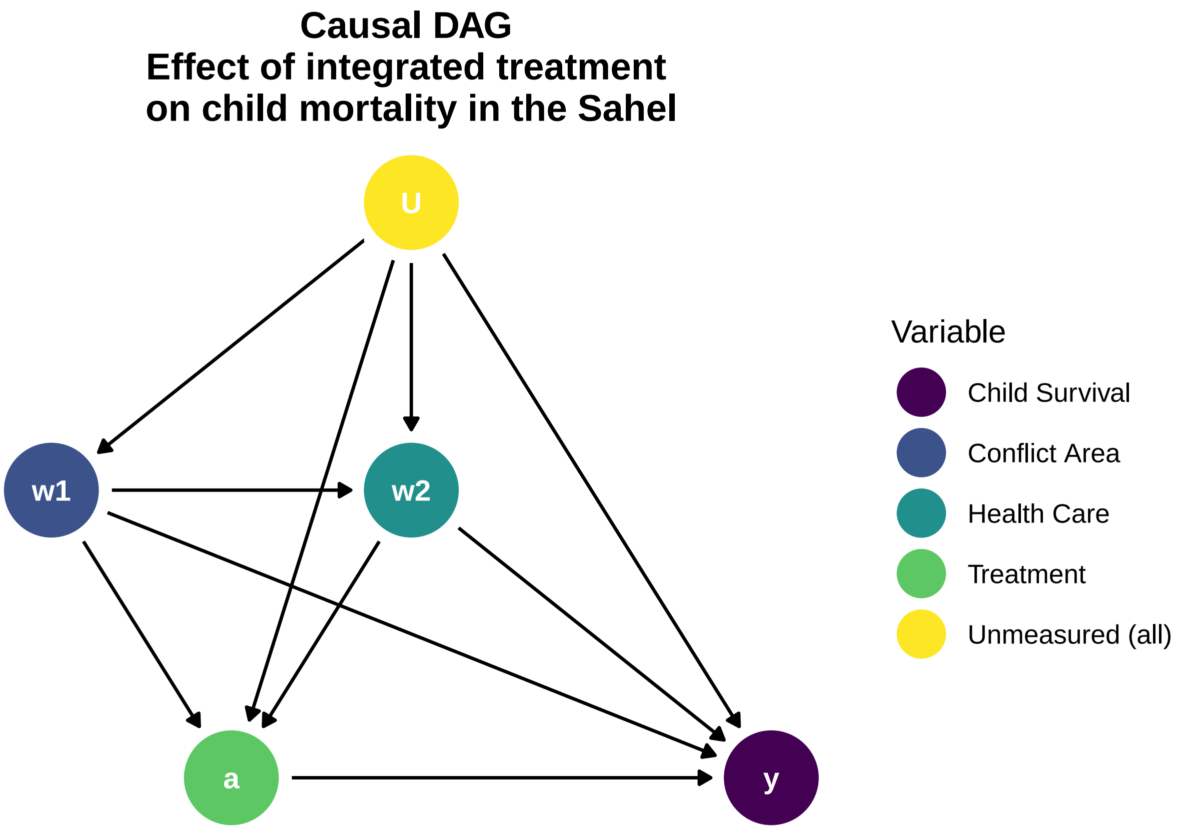
$O = (W1, W2, A, Y) \sim \mathbb{P}_0$

2. Step 4-5: Identification & statistical estimand:

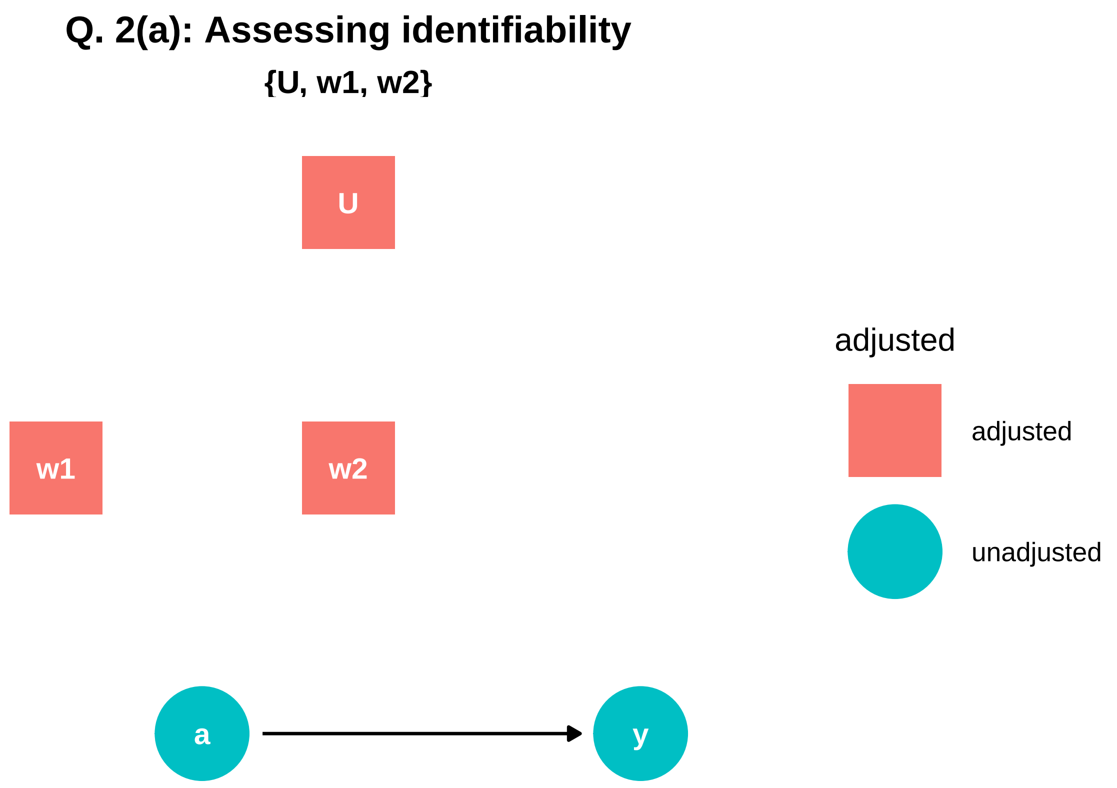
(a) Using the backdoor criterion, assess identifiability.

Since we have not made any independence assumptions on the background factors, then there can be no identifiability.

```
child_mortality <- dagify(y ~ w1 + w2 + a + U,
                          a ~ w1 + w2 + U,
                          w1 ~ U,
                          w2 ~ w1 + U,
                     labels = c("y" = "Child Survival",
                                "a" = "Treatment",
                                "w1" = "Conflict Area",
                                "w2" = "Health Care",
                                "U" = "Unmeasured all"),
                     exposure = "a",
                     outcome = "y",
                     coords = list(x = c(y = 5, a = 2, w1 = 1, w2 = 3, U = 3),
                                   y = c(y = 1, a = 1, w1 = 2, w2 = 2, U = 3))) %>%
  tidy_dagitty() %>%
  dplyr::mutate(Variable = case_when(
    name == "y" ~ "Child Survival",
    name == "a" ~ "Treatment",
    name == "w1" ~ "Conflict Area",
    name == "w2" ~ "Health Care",
    name == "U" ~ "Unmeasured (all)"))

child_mortality_dag <- child_mortality %>%
  ggplot(aes(
    x = x,
    y = y,
    xend = xend,
    yend = yend
  )) +
  geom_dag_point(aes(color = Variable)) +
  geom_dag_edges() +
  geom_dag_text() +
  theme_dag() +
  scale_color_viridis_d()+
  ggtitle("Causal DAG \nEffect of integrated treatment \non child mortality in the Sahel") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold")) +
  guides(color = guide_legend(override.aes = list(size = 8)))

child_mortality_dag
```



```r
ggsave(here("r_labs/r_hw2/child_mortality_dag.jpg"), width = 8, height = 6, units = "in")

w_adj_dag_1 <- ggdag_adjustment_set(child_mortality, exposure = "a", outcome = "y", type = "all") +
  theme_dag() +
  labs(title = "Q. 2(a): Assessing identifiability") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

w_adj_dag_1
```

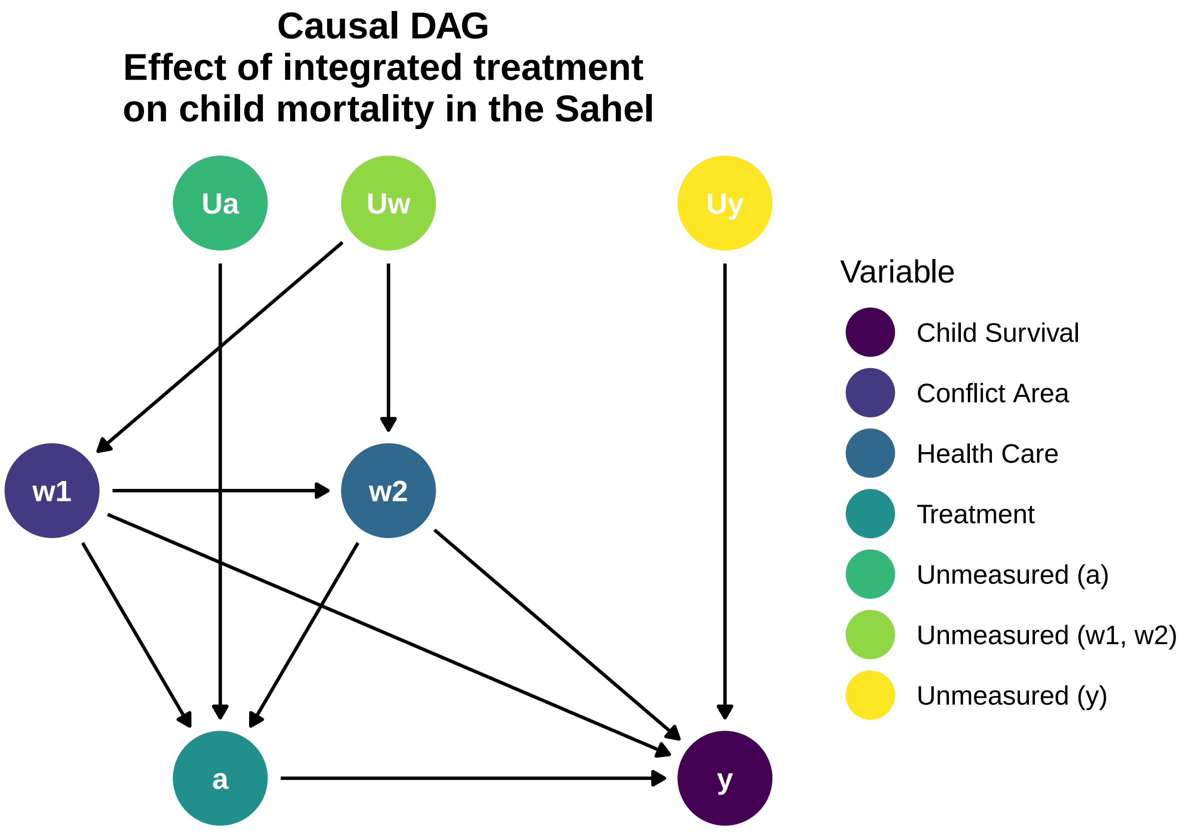


```
ggsave(here("r_labs/r_hw2/w_adj_dag_1.jpg"), width = 8, height = 6, units = "in")
```

(b) If the target causal parameter is not identified, under what assumptions would it be?

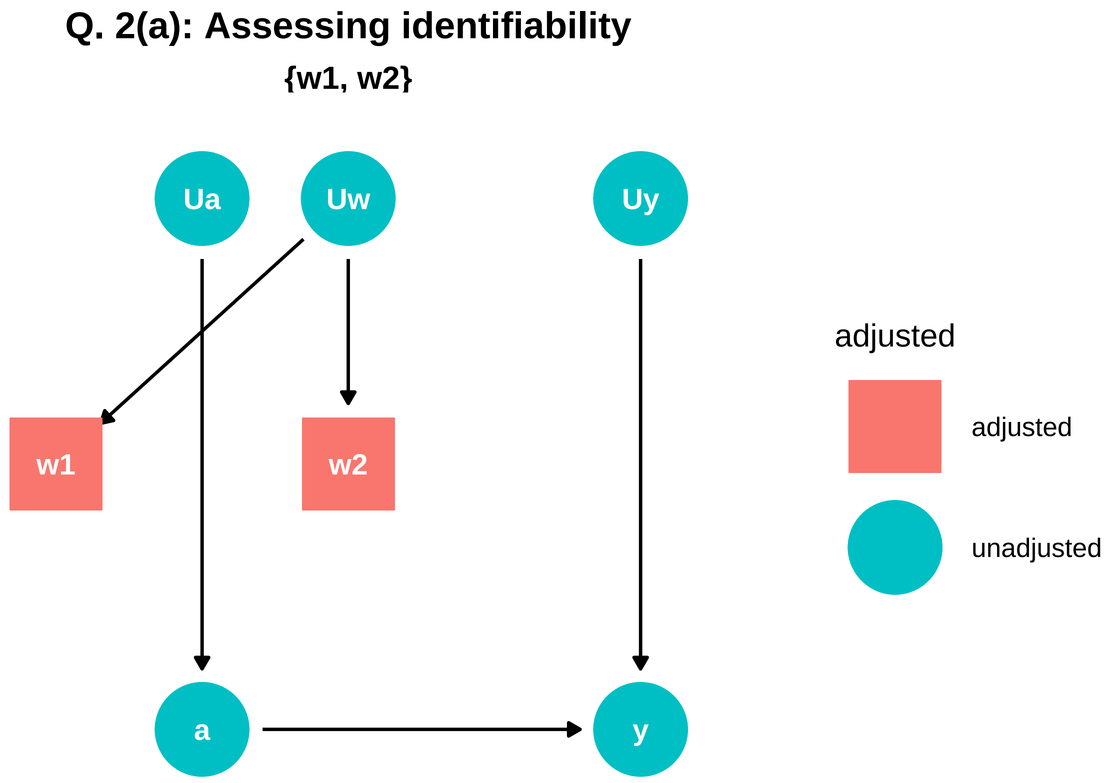
We would require some independence assumption between U_A and U_Y, for instance if treatment were randomly assigned. One example is given below, where the target parameter could be identified by adjusting for $W1$ and $W2$:

```
child_mortality_2 <- dagify(y ~ w1 + w2 + a + Uy,
                            a ~ w1 + w2 + Ua,
                            w1 ~ Uw,
                            w2 ~ w1 + Uw,
                       labels = c("y" = "Child Survival",
                                  "a" = "Treatment",
                                  "w1" = "Conflict Area",
                                  "w2" = "Health Care",
                                  "Uw" = "Unmeasured for w1, w2",
                                  "Ua" = "Unmeasured a",
                                  "Uy" = "Unmeasured y"
                                  ),
                       exposure = "a",
                       outcome = "y",
                       coords = list(x = c(y = 5, a = 2, w1 = 1, w2 = 3, Uw = 3, Ua = 2, Uy = 5),
                                     y = c(y = 1, a = 1, w1 = 2, w2 = 2, Uw = 3, Ua = 3, Uy = 3))) %>%
  tidy_dagitty() %>%
  dplyr::mutate(Variable = case_when(
    name == "y" ~ "Child Survival",
    name == "a" ~ "Treatment",
    name == "w1" ~ "Conflict Area",
    name == "w2" ~ "Health Care",
    name == "Uw" ~ "Unmeasured (w1, w2)",
    name == "Ua" ~ "Unmeasured (a)",
    name == "Uy" ~ "Unmeasured (y)"))

child_mortality_2_dag <- child_mortality_2 %>%
  ggplot(aes(
    x = x,
    y = y,
    xend = xend,
    yend = yend
  )) +
  geom_dag_point(aes(color = Variable)) +
  geom_dag_edges() +
  geom_dag_text() +
  theme_dag() +
  scale_color_viridis_d()+
  ggtitle("Causal DAG \nEffect of integrated treatment \non child mortality in the Sahel") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold")) +
  guides(color = guide_legend(override.aes = list(size = 8)))

child_mortality_2_dag
```



```
ggsave(here("r_labs/r_hw2/child_mortality_2_dag.jpg"), width = 8, height = 6, units = "in")

w_adj_dag_2 <- ggdag_adjustment_set(child_mortality_2, exposure = "a", outcome = "y", type = "minimal")
```

```
  theme_dag() +
  labs(title = "Q. 2(a): Assessing identifiability") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

w_adj_dag_2
```



```
ggsave(here("r_labs/r_hw2/w_adj_dag_2.jpg"), width = 8, height = 6, units = "in")
```

(c) Specify the target parameter of the observed data distribution (i.e., the statistical estimand). Interpret it.

We can "commit" to the following interesting statistical estimand, inspired by our scientific/causal question:

$$\Psi(\mathbb{P}_0) = \mathbb{E}_0[\mathbb{E}_0(Y|A=1, W1, W2) - \mathbb{E}_0(Y|A=0, W1, W2)]$$

This means that our target parameter is the expected value of the difference in survival between children given the treatment and those not given the treatment, conitional on the covariates.

(d) What is the relevant positivity assumption? Is it reasonable here?

The positivity assumption in this instance is that there are children who will receive treatment and not receive treatment under each of the four covariate permutations - that is, when $W1$ and $W2$ are equal to 0 and 1. In our simplified model, this is a reasonable assumption, if we implement a cluster randomized trial to ensure we have groups under both covariate conditions. Otherwise, it would be very challenging, as it might be impossible to avoid contamination.

3 A specific data generating process

1. Evaluate the positivity assumption in closed form for this data generating process.

$$\Psi(\mathbb{P}_0) = \mathbb{E}_0[\mathbb{E}_0(Y|A=1,W1,W2) - \mathbb{E}_0(Y|A=0,W1,W2)]$$

$$= \sum_{w1,w2} [\mathbb{E}_0(Y|A=1,W1=w1,W2=w2) - \mathbb{E}_0(Y|A=0,W1=w1,W2=w2)]\mathbb{P}_0(W2=w2|W1=w1)\mathbb{P}_0(W1=w1)$$

In this particular data generating system (one of many compatible with the SCM), the conditional probability of receiving the intervention given the adjustment variables is

$$\mathbb{P}_0(A=1|W1,W2) = logit^{-1}(-0.5 + W1 - 1.5 * W2)$$

2. ***Bonus (Optional):*** **Evaluate the statistical estimand $\Psi(\mathbb{P}_{\nvdash})$ in closed form for this data generating process.**

```r
Psi.P0 <- (plogis(-0.5+0.5-(1.5*0.5)))
Psi.P0
```

```
## [1] 0.3208213
```

4 Translate this data generating process into simulations.

1. First set the seed to 252.

```r
set.seed(252)
```

2. Write a function to generate the observed data $O = (W1, W2, A, Y)$ and the counterfactual outcomes (Y_1, Y_0). Recall we generate the counterfactual outcome Y_1 by intervening to set the exposure to the combination package $(A = 1)$, and we generate the counterfactual outcomes Y_0 by intervening to set the exposure to the standard of care $(A = 0)$. Also recall $logit^{-1}$ function is given by the `plogis` function in `R`.

```r
generate.data <- function(n){
  U.W1<- runif(n, min=0, max=1)
  U.W2<- runif(n, min=0, max=1)
  U.A<- runif(n, min=0, max=1)
  U.Y<- runif(n, min=0, max=1)
  #
  W1 <- as.numeric(U.W1 < 0.5)
  W2 <- as.numeric(U.W2 < 0.5)
  A <- as.numeric(U.A < plogis(-0.5+W1-(1.5*W2)))
  Y <- as.numeric(U.Y < plogis(-0.75+W1-(2*W2)+(2.5*A)+(A*W1)))
  #
  Y.1 <- as.numeric(U.Y < plogis(-0.75+W1-(2*W2)+(2.5)+(W1)))
  Y.0 <- as.numeric(U.Y < plogis(-0.75+W1-(2*W2)))
  #
  data.frame(cbind(W1, W2, A, Y, Y.1, Y.0))
}
```

3. Suppose our target population consists of 100,000 people. Set the number of draws $n = 100,000$. Use your function to generate n i.i.d. observations.

```r
n <- 100000
Obs <- generate.data(n)
head(Obs)
```

```
##   W1 W2 A Y Y.1 Y.0
## 1  0  0 1 1   1   1
## 2  0  0 1 1   1   0
## 3  1  0 1 1   1   1
## 4  0  0 1 1   1   0
## 5  0  0 1 1   1   1
## 6  1  0 1 1   1   1
```

```r
summary(Obs)
```

```
##        W1               W2                A                 Y
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :1.0000   Median :0.0000   Median :0.0000
##  Mean   :0.499   Mean   :0.5007   Mean   :0.3466   Mean   :0.4451
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##       Y.1              Y.0
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000
##  Mean   :0.7787   Mean   :0.2717
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

4. Does the counterfactual outcome Y_a equal the observed outcome Y when the observed exposure is $A = a$?

```r
Y.a <- mean(Obs$Y.1) - mean(Obs$Y.0)
Y.a
```

```
## [1] 0.50707
```

```r
Y.a1 <- filter(Obs, A==1)
Y.a0 <- filter(Obs, A==0)
Y.obs <- mean(Y.a1$Y.1) - mean(Y.a0$Y.0)
Y.obs
```

```
## [1] 0.6502963
```

```r
diff <- Y.a - Y.obs
diff
```

```
## [1] -0.1432263
```

No, the observed survival is about 15% higher than the counterfactual.

5. *Bonus:* Evaluate and interpret the causal parameter $\Psi^*(\mathbb{P}^*)$.

5 The simple substitution estimator based on the G-Computation formula

1. Set the number of iterations R to 500 and the number of observations n to 200. Do *not* reset the seed.

```
R <- 500
n <- 200
```

2. Create a $R = 500$ by 4 matrix estimates to hold the resulting point estimates obtained at each iteration. The rows will correspond to iterations and the columns to different estimators.

```
# Hint: the following code creates a matrix filled with NA of size 10 by 10
# estimates <- matrix(NA, nrow=10, ncol=10)
estimates <- matrix(NA, nrow=R, ncol=4)
```

3. Inside a for loop from r equals 1 to R (500), do the following. *Note: see RAssign2.pdf for further detailed instructions and hints.*

```
# (a) Use your function from Part 4 to generate n i.i.d. observations. Subset the resulting data frame

# (b) Copy the data set Obs into two new data frames txt and control.  Then set $A=1 for all units in t

# (c) Implement the simple substitution estimator (i.e., parametric G-computation) using each one of th

# (d) Assign the resulting point estimates as a row in the matrix estimates.
for(i in 1:R){
  df <- generate.data(n)
  Obs <- subset(df, select=c(W1,W2,A,Y))

  reg.model_1 <- glm(Y ~ A, family='binomial', data=Obs)
  reg.model_2 <- glm(Y ~ A + W1, family='binomial', data=Obs)
  reg.model_3 <- glm(Y ~ A + W2, family='binomial', data=Obs)
  reg.model_4 <- glm(Y ~ A*(W1 +W2), family='binomial', data=Obs)

  txt<- control <- Obs
  txt$A <- 1
  control$A <- 0

  predictY.txt.1 <- predict(reg.model_1, newdata = txt, type='response')
  predictY.txt.2 <- predict(reg.model_2, newdata = txt, type='response')
  predictY.txt.3 <- predict(reg.model_3, newdata = txt, type='response')
  predictY.txt.4 <- predict(reg.model_4, newdata = txt, type='response')

  predictY.control.1 <- predict(reg.model_1, newdata = control, type='response')
  predictY.control.2 <- predict(reg.model_2, newdata = control, type='response')
  predictY.control.3 <- predict(reg.model_3, newdata = control, type='response')
  predictY.control.4 <- predict(reg.model_4, newdata = control, type='response')

  estimates[i,] <- mean(predictY.txt.1 - predictY.control.1) %>%
    append(mean(predictY.txt.2 - predictY.control.2)) %>%
    append(mean(predictY.txt.3 - predictY.control.3)) %>%
    append(mean(predictY.txt.4 - predictY.control.4))
}
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
head(estimates)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.7184953 0.6997235 0.6575439 0.6495538
## [2,] 0.7448098 0.7250050 0.7056346 0.6698799
## [3,] 0.6274725 0.5811075 0.5536597 0.4529381
## [4,] 0.6345486 0.6214173 0.5810069 0.5417187
## [5,] 0.6374454 0.6162322 0.5685941 0.5341143
## [6,] 0.6001838 0.5833091 0.5542173 0.5245538
```

6 Performance of the estimators.

1. What is the average point estimate from each?

```
meanEst.1 <- mean(estimates[,1])
meanEst.2 <- mean(estimates[,2])
meanEst.3 <- mean(estimates[,3])
meanEst.4 <- mean(estimates[,4])
meanEst <- cbind(meanEst.1, meanEst.2, meanEst.3, meanEst.4)
meanEst
```

```
##      meanEst.1 meanEst.2 meanEst.3 meanEst.4
## [1,] 0.6505123 0.6228431 0.5653621 0.5060037
```

```
estimates %>%
  as_tibble() %>%
  summarise(across(V1:V4, mean))
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
## # A tibble: 1 x 4
##      V1    V2    V3    V4
##   <dbl> <dbl> <dbl> <dbl>
## 1 0.651 0.623 0.565 0.506
```

2. Estimate the bias of each estimator. For each estimator, average the difference between point estimate psi_n and the truth psi_0.

```
head(estimates)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.7184953 0.6997235 0.6575439 0.6495538
## [2,] 0.7448098 0.7250050 0.7056346 0.6698799
## [3,] 0.6274725 0.5811075 0.5536597 0.4529381
## [4,] 0.6345486 0.6214173 0.5810069 0.5417187
## [5,] 0.6374454 0.6162322 0.5685941 0.5341143
## [6,] 0.6001838 0.5833091 0.5542173 0.5245538
```

```
bias.mtx <- estimates - Psi.P0
bias <- bias.mtx %>%
  as_tibble() %>%
  summarise(across(V1:V4, mean))
bias
```

```
## # A tibble: 1 x 4
##      V1    V2    V3    V4
##   <dbl> <dbl> <dbl> <dbl>
## 1 0.330 0.302 0.245 0.185
```

3. Estimate the variance of each estimator.

```
var <- estimates %>%
  as_tibble() %>%
  summarise(across(V1:V4, var))
var
```

```
## # A tibble: 1 x 4
##        V1      V2      V3      V4
##     <dbl>   <dbl>   <dbl>   <dbl>
## 1 0.00318 0.00373 0.00471 0.00616
```

4. Estimate the mean squared error of each estimator.

```
mse.mtx <- ((estimates-Psi.P0)^2)
mse <- mse.mtx %>%
  as_tibble() %>%
  summarise(across(V1:V4, mean))
mse
```

```
## # A tibble: 1 x 4
##      V1     V2     V3     V4
##   <dbl>  <dbl>  <dbl>  <dbl>
## 1 0.112 0.0949 0.0645 0.0404
```

5. Briefly comment on the performance of the estimators in this simulation setting. Which estimator has the lowest MSE over the $R = 500$ iterations? Are you surprised?

The performance improved as the model became more complex, which makes sense. In fact, the fourth model was the only one to include both covariates W1 and W2. The bias and MSE both decreased with model complexity, although the variance increased, which also probably more accurately represents the variability in the data.

7 Identifying the mean counterfactual outcome under a dynamic intervention

1. Explain why (1) holds using properties of conditional expectations.

Given $Y_d \perp\!\!\!\perp A|W1, W2$ and that W2 is now part of the binary dynamic treatment, the expectation of Y_d is the sum of expectations under each covariate multiplied by the probability of each covariate.

2. Explain why (2) holds using properties of conditional expectations and the fact that $Y_d \perp A|W1, W2$. Note: No need to explain $Y_d \perp A|W1, W2$ in the context of the study since you have already discussed the assumptions need for the backdoor criterion to hold, and the backdoor criterion implies $Y_d \perp A|W1, W2$.

The previous expectation can be expanded as is done in (*) by the property of conditional expectations (lecture 6). Given the nature of the dynamic treatment, it can be expanded to be conditional on the assignment $A = d(w2)$. Since it is a dynamic treatment, the distribution should be the same across the assigned treatment.

3. Explain why (3) holds.

The counterfactual distribution of covariates should be equivalent to the observed distribution of covariates in an RCT.

4. Explain why (4) holds.

Given the consistency assumption, and the assignment of treatment, the observed distribution of outcomes should be equivalent to the counterfactual distribution.

Note - I submitted the first version of this assignment on time, but kept updating this file, because there were parts which I hadn't adequately completed. This final version was submitted before the answer key had been made available in the shared drive. Thanks for your understanding.