Inverse Probability Weighting

From backdoor criterion part, we learned that PA(X) always satisfies backdoor criterion. Hence, we can represent postintervention causal effect as such:

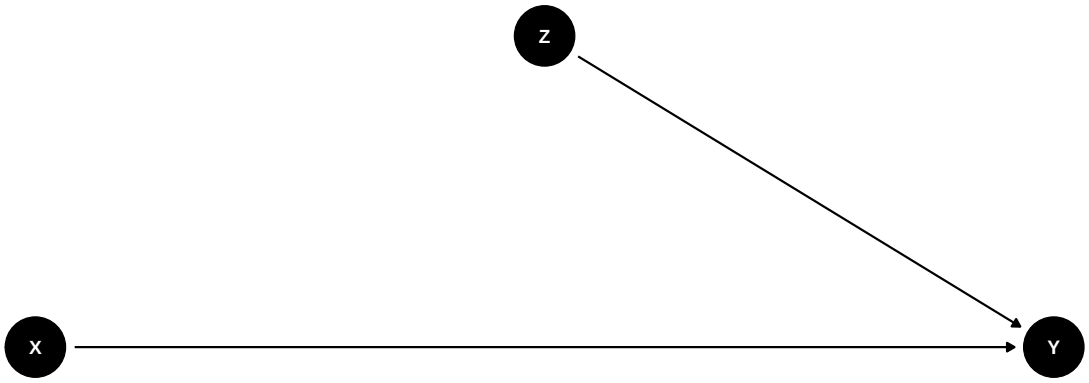
$P(Y = y \mid do(X = x)) = \sum_z P(Y = y \mid X = x, PA = z)P(PA = z)$

Modifying this equation, we can obtain the following:

$P(Y = y \mid do(X = x)) = \sum_z \frac{P(Y=y,X=x,PA=z)}{P(X=x|PA=z)}$

The factor $P(X = x \mid PA = z)$ in the denominator is known as the "propensity score."

Now, assume we have a graphical model with an intervention, do(X=x), on the model.



$P(Y = y \mid do(X = x))$

$= P_m(Y = y \mid X = x)$ (by definition)

$= \sum_z P_m(Y = y \mid X = x, Z = z)P(Z = z \mid X = x)$ (Bayes' rule)

$= \sum_z P_m(Y = y \mid X = x, Z = z)P(Z = z)$ $(X \perp\!\!\!\perp Z)$

$= \sum_z P(Y = y \mid X = x, Z = z)P(Z = z)$ (invariance relations)

$= \sum_z \frac{P(Y=y|X=x,Z=z)P(X=x|Z=z)P(Z=z)}{P(X=x|Z=z)}$

$= \sum_z \frac{P(Y=y,X=x,Z=z)}{P(X=x|Z=z)}$

From this result, we can see that postintervention causal effect can be computed by multiplying the pre-treatment distribution of (X,Y,Z) by a factor $1/P(X = x \mid Z = z)$, propensity score. Namely, each case $(Y = y, X = x, Z = z)$in the population receives a weight of the inverse of the conditional probability of receiving the treatment level given a set of observed covariates. This is the reason why this method is called "inverse probability weighting."

Creating data (1000 individuals) and computing the true log odds ratio (B_1)

```
pre_data <- defData(varname = "L", formula = 0.37,
                dist = "binary")
pre_data <- defData(pre_data, varname = "Y0", formula = "-1.7 + 1.77*L",
                dist = "binary", link = "logit")
pre_data <- defData(pre_data, varname = "Y1", formula = "-0.7 + 1.77*L",
                dist = "binary", link = "logit")
pre_data <- defData(pre_data, varname = "A", formula = "0.27 + 0.37 * L",
                dist = "binary")
pre_data <- defData(pre_data, varname = "Y", formula = "Y0 + A * (Y1 - Y0)",
                dist = "nonrandom")

set.seed(77777)
df <- genData(1000, pre_data)

odds <- function(p) {
    return((p/(1 - p)))
}

log(odds(mean(df$Y1)) / odds(mean(df$Y0)))
```

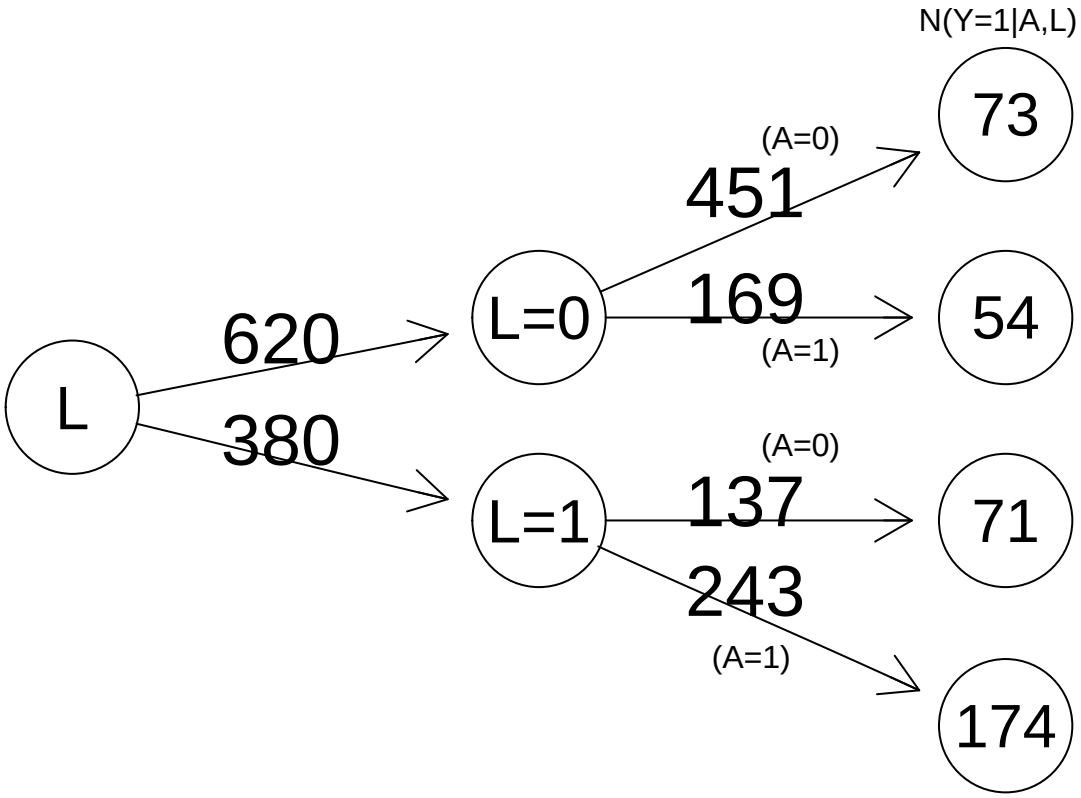
```
## [1] 0.7417183
```

Generating Joint Probability Distribution P(Y,A,L)

```
table <-
  df %>%
  group_by(L,A,Y) %>%
  summarise(percent_population = n()/1000)  %>%
  kbl(
      caption     = "Joint Probability Distribution P(Y,A,L)"
    , col.names   = c("L", "A", "Y", "% of Population")
    , format.args = list(big.mark = ',')
  ) %>%
  # further map to a more professional-looking table
  kable_paper("striped", full_width = F)
```

Table 1: Joint Probability Distribution P(Y,A,L)

L	A	Y	% of Population
0	0	0	0.378
0	0	1	0.073
0	1	0	0.115
0	1	1	0.054
1	0	0	0.066
1	0	1	0.071
1	1	0	0.069
1	1	1	0.174



Computing the marginal causal effect ignoring L before applying IP weights

$P(Y = 1 \mid A = 0) = \frac{(451 \times \frac{73}{451} + 137 \times \frac{71}{137})}{(451+137)} = 0.245$

$P(Y = 1 \mid A = 1) = \frac{(169 \times \frac{54}{169} + 243 \times \frac{174}{243})}{(169+243)} = 0.186 = 0.553$

Computing crude log odds ratio

$LOR_{A=1 \ vs \ A=0} = log\left(\frac{(0.553/0.447)}{(0.245/0.755)}\right) = 1.34$

Comparing the result with coefficient of estimate ($Beta_1$) from a logistic model

```
glm(Y ~ A, data = df, family ="binomial") %>% broom::tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    -1.13    0.0959    -11.7  7.80e-32
## 2 A               1.34    0.138       9.72 2.48e-22
```

It is to be observed that the crude log odds ratio is the same as the coefficient of estimate. As expected, however, the crude log odds ratio is biased because we did not take into account the confounder L. The true odds ratio is 0.74 whereas the crude estimate is 1.34.

Computing Propensity Score

1. $P(A = 0 \mid L = 0) = \frac{451}{620}$
2. $P(A = 1 \mid L = 0) = \frac{169}{620}$
3. $P(A = 0 \mid L = 1) = \frac{137}{380}$
4. $P(A = 1 \mid L = 1) = \frac{243}{380}$

Let weight $W^A = \frac{1}{P(A|L)}$

$W_1^A = 1.37$

$W_2^A = 3.67$

$W_3^A = 2.77$

$W_4^A = 1.56$

Applying IP Weights to each individual

If we apply IP Weights to each individual, the hypothetical population becomes double the size of the original population and it's called the pseudo-population. This is because every individual appears both treated and untreated in the pseudo-population.

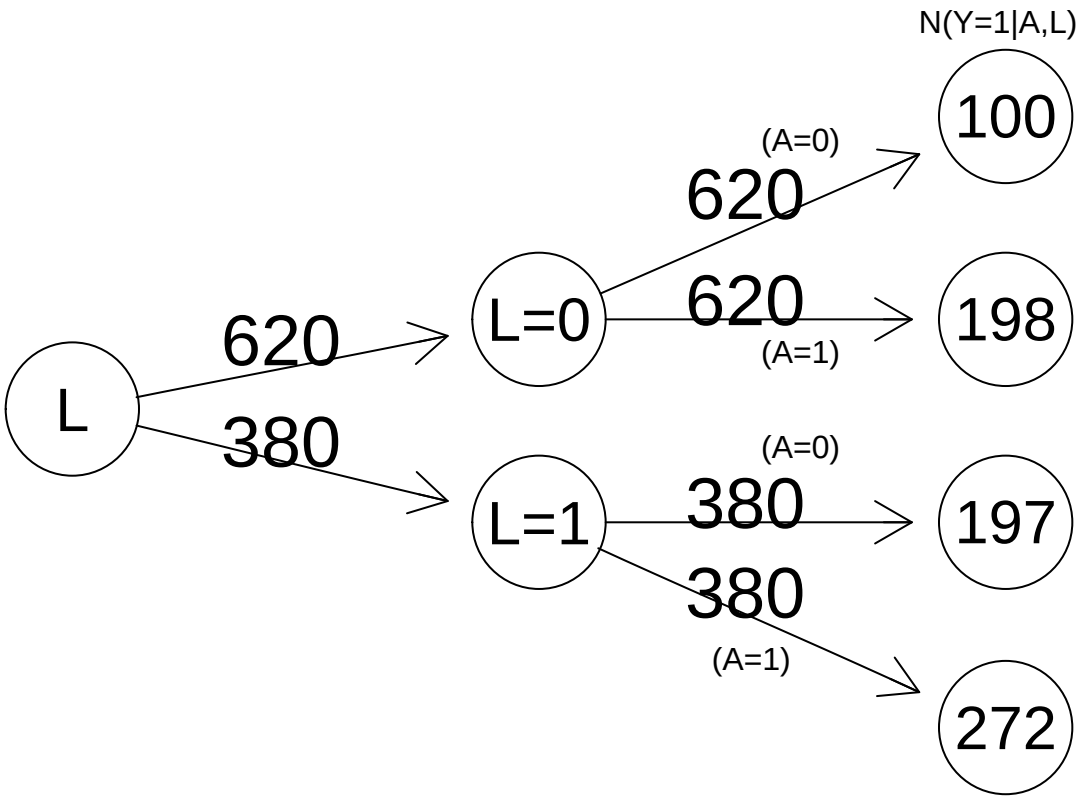
$N(Y = 1 \mid A = 0, L = 0) \times W_1^A = 73 \times 1.37 = 100$

$N(Y = 1 \mid A = 1, L = 0) \times W_1^A = 54 \times 3.67 = 198$

$N(Y = 1 \mid A = 0, L = 1) \times W_1^A = 71 \times 2.77 = 197$

$N(Y = 1 \mid A = 1, L = 1) \times W_1^A = 174 \times 1.56 = 272$

Applying IP weights to each individual, resulting in the pseudo-population



Computing the marginal causal effect ignoring L after applying IP weights

$P(Y = 1 \mid A = 0) = \frac{(620 \times \frac{100}{620} + 357 \times \frac{197}{380})}{(620+380)} = 0.297$

$P(Y = 1 \mid A = 1) = \frac{(620 \times \frac{198}{620} + 380 \times \frac{272}{380})}{(620+380)} = 0.47$

$LOR_{A=1 \ vs \ A=0} = log\left(\frac{(0.47/0.53)}{(0.297/0.703)}\right) = 0.74$

Comparing the result with coefficient of estimate ($Beta_1$) from a logistic model

```
tidy(glm(Y ~ A , data = df3, family = "binomial", weights = IPW))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -0.860    0.0692    -12.4  1.71e-35
## 2 A              0.741    0.0938      7.90 2.84e-15
```

Through IP weighting method, we have been able to compute the log odds ratio(Y=1) very close to true odds ratio without taking into account L. Important thing to note here is that the IP weighting is only valid when L satisfies the backdoor criterion.

Importing NHEFS data

```
nhefs <- read_csv("nhefs.csv")

nhefs_uncensored <-
  nhefs %>%
  mutate(cens = ifelse(is.na(wt82), 1, 0)) %>%
  relocate(cens, wt82) %>%
  filter(!is.na(wt82))
```

Estimation of IP weights via a logistic model with multiple covariates

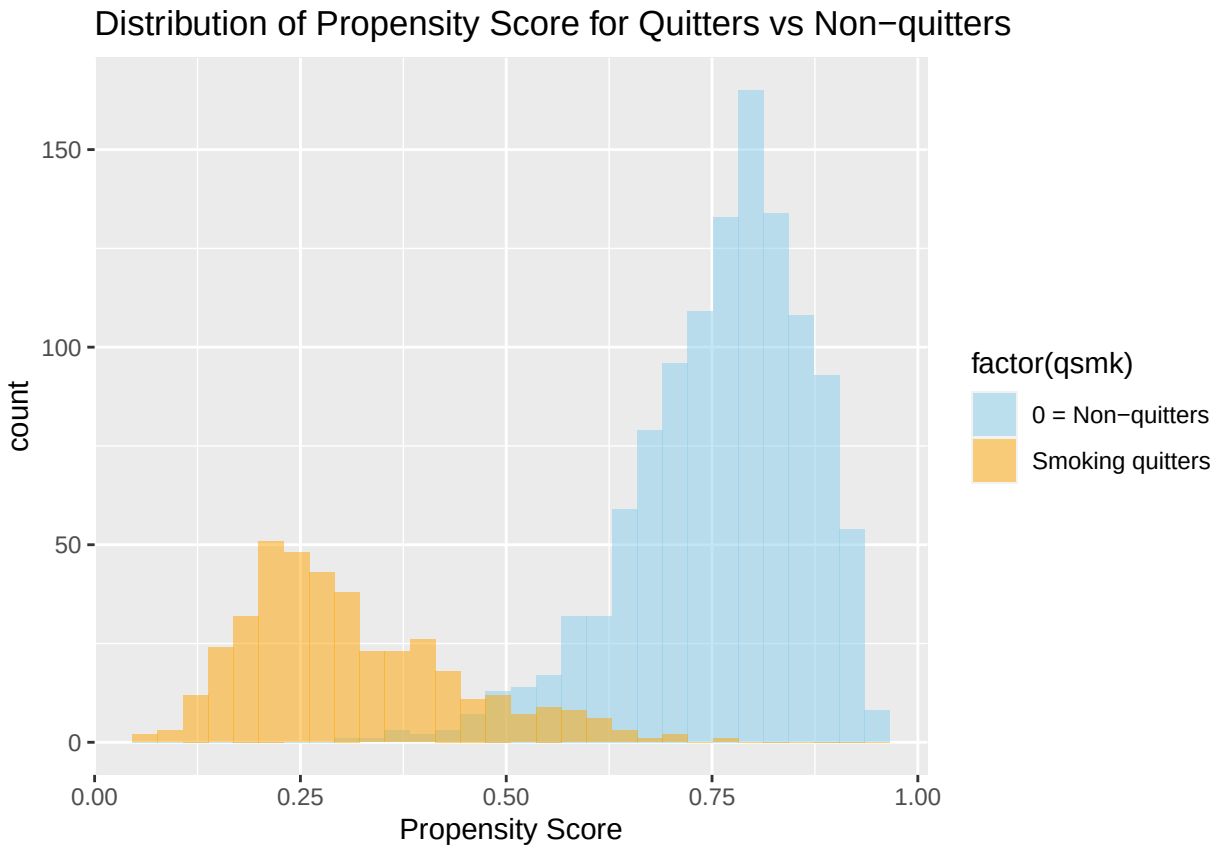
```
propensity_model <- glm(
  qsmk ~ sex + race + age + I(age ^ 2) +
    as.factor(education) + smokeintensity +
    I(smokeintensity ^ 2) + smokeyrs + I(smokeyrs ^ 2) +
    as.factor(exercise) + as.factor(active) + wt71 + I(wt71 ^ 2),
  family = binomial(),
  data = nhefs_uncensored
)
```

Computing propensity score. Note that Pr[A=0|L] = 1-Pr[A=1|L]

```
p.qsmk.obs <-
  ifelse(nhefs_uncensored$qsmk == 0,
         1 - predict(propensity_model, type = "response"),
         predict(propensity_model, type = "response"))

nhefs_uncensored <-
  nhefs_uncensored %>%
  mutate(w = 1/p.qsmk.obs, #inverse propensity score (weight)
         ps = p.qsmk.obs) #propensity score

nhefs_uncensored %>%
  ggplot(aes(ps, group = qsmk, fill = factor(qsmk))) +
  geom_histogram(alpha = 0.5, position = "identity") +
  scale_fill_manual(values = c("skyblue", "orange"),
                    labels = c("0 = Non-quitters", "Smoking quitters")) +
  labs(x = "Propensity Score",
       title = "Distribution of Propensity Score for Quitters vs Non-quitters")
```



Computing a coefficient $Beta_1$ of the marginal structural model

```
msm.w <- geeglm(
  wt82_71 ~ qsmk,
  data = nhefs_uncensored,
  weights = w,
  id = seqn,
  corstr = "independence"
)

beta <- coef(msm.w)
SE <- coef(summary(msm.w))[, 2]
Lower_CI <- beta - qnorm(0.975) * SE
Upper_CI <- beta + qnorm(0.975) * SE
cbind(beta, Lower_CI, Upper_CI)
```

```
##                beta Lower_CI Upper_CI
## (Intercept) 1.779978 1.339514 2.220442
## qsmk        3.440535 2.410587 4.470484
```

Reference

Pearl, J., Glymour, M., & Jewell, N. P. (2019). Causal inference in statistics a primer. Wiley.

Hernán MA, Robins JM (2020). Causal Inference: What If. Boca Raton: Chapman & Hall/CRC.