Evaluation of Test-Negative Designs with Randomized Placebo-Controlled Trials

L. Andrews

2024-09-20

This R Vignette supplements and mimics the analyses from the manuscript, "COVID-19 Vaccine Effectiveness: An Evaluation of the Test-Negative Design with Randomized Placebo-Controlled Clinical Trials." The manuscript resamples data from five Phase 3 COVID-19 Prevention Network (CoVPN) Randomized Placebo-Controlled Trials (RCTs) using several test-negative design (TND) sampling approaches and applies a novel semiparametric logistic regression approach involving targeted maximum likelihood estimation to estimate vaccine effectiveness. The CoVPN RCT data cannot be shared so instead, we generated a toy RCT dataset. We applied four common TND sampling methods to the toy RCT dataset to obtain the following TND samples: participant-based sample without censoring for COVID-19, the participant-based sample with censoring for COVID-19, the specimen-based sample, and the random specimen-based sample. We then applied the semiparametric logistic regression involving targeted maximum likelihood estimation and ordinary logistic regression approaches described in the manuscript to each TND sample to estimate TND vaccine effectiveness. We also estimated RCT vaccine efficacy using an unadjusted Cox Proportional Hazards Regression model for comparison.

Generating Data

Since we cannot share the CoVPN RCT data, we generate a toy phase 3 COVID-19 RCT cohort and obtain four TND samples from this cohort. We also define a non-SARS-CoV-2 endpoint, defined as meeting the COVID-19 symptom definition and testing SARS-CoV-2 negative, in the RCT cohort to mimic our analyses in the manuscript that assess Noncase Exchangeability violations.

RCT Cohort

Our toy RCT dataset consists of 30,000 participants who are randomly assigned to COVID-19 vaccine or placebo at a 1:1 ratio. The blinded phase of the RCT takes place from August 1, 2020 to March 31, 2021. For simplicity, all participants are vaccinated in the first two weeks of August and no participants were unblinded early or lost to follow up. At RCT enrollment, the following demographic and clinical characteristics were measured: age, sex at birth, race/ethnicity, geographic region, and presence of any comorbidities.

```
## Demographic and Clinical Characteristics
# One Row per Participant
set.seed(7)
nsubj <- 30000

demo.df0 <- data.frame(USUBJID = paste0("ID", 1:nsubj),
                AGE = round(runif(nsubj, min=18, max=80)),
                SEX = sample(c("F","M"),nsubj, replace=TRUE),
```

```
    # Dichotomized Race/Ethnicity (Hispanic/Latino or Non-White vs.
    # Non-Hispanic/Non-Latino White)
    POC = sample(c("POC","White"),nsubj,
                 replace=TRUE, prob=c(1/3,2/3)),
  # US Census Regions
    REGION = sample(c("Midwest","NE","South","West"),nsubj,
                    replace=TRUE, prob = c(2/10,1/10,5/10,2/10)),
  # Comorbidities
    COMORB = sample(c("At Risk","Not At Risk"),nsubj,
                    replace=TRUE, prob=c(1/4,3/4)),
  # Final Intervention Date (Placebo or Vaccine)
  DOSE2DT = sample(seq(as.Date('2020/08/01'),as.Date('2020/08/14'), by="day"),
                   nsubj, replace=TRUE),
  # Randomized Vaccination Status
  VACCINE = sample(0:1,nsubj, replace=TRUE),
  # End of Blinded Phase Date
  ENDDT = as.Date('2021/03/31'))
```

During the RCT, participants may experience an illness episode or symptomatic period and are then prompted to seek SARS-CoV-2 testing. In this dataset, participants can experience 0-5 illness episodes during the study and the date of symptom onset (`ONSETDT`) and first date of meeting the COVID-19 like symptom definition (`SYMDT`) are documented. Illness episodes may be truly caused by SARS-CoV-2 or by a non-SARS-CoV-2 illness (e.g., other respiratory viruses or allergies). By construction, the COVID-19 vaccine protects against COVID-19 (vaccine efficacy = 80%) and does not affect non-SARS-CoV-2 illness.

Participants will obtain 1-10 SARS-CoV-2 PCR tests before and/or during an illness episode. While an illness episode may be truly caused by SARS-CoV-2 (`eps_sars_inf=1`), SARS-CoV-2 test results (`POSTEST`) are subject to misclassification (100% specificity, 85% sensitivity).

```
# Adding Illness Episode Information
demo.df <- demo.df0 %>% mutate(
     # True SARS-CoV-2 Infection Status at Any Point during Blinded Phase
     # (Unknown in Observed Clinical Data)
      any_sars_inf = rbinom(nrow(.),1,exp( log(0.04) + log(0.2)*VACCINE) ),
      # Number of SARS-CoV-2 Illness Episodes
      n_sars_eps = any_sars_inf*sample(1:2, nsubj, replace=TRUE, prob = c(0.95,0.05)),
      # Number of Non-SARS-CoV-2 Illness Episodes
      n_nonsars_eps = sample(0:3, nsubj, replace=TRUE, prob = c(0.9,0.07,0.02,0.01)))


## Illness Episode Characteristics
# Creating One Row Per Illness Episode

# SARS-CoV-2 Illness Episodes
sars.eps.df0 <- uncount(demo.df , n_sars_eps) %>%
  # Illness Episode SARS-CoV-2 Infection Status
  mutate(eps_sars_inf = 1) %>% select(-n_nonsars_eps)

# Non-SARS-CoV-2 Illness Episodes
non.sars.eps.df0 <- uncount(demo.df , n_nonsars_eps) %>%
  # Illness Episode SARS-CoV-2 Infection Status
  mutate(eps_sars_inf = 0) %>% select(-n_sars_eps)

# Adding Testing and Symptom Information for Both Types of Illness Episodes
```

```
eps.df00 <- rbind(sars.eps.df0, non.sars.eps.df0) %>% arrange(USUBJID)
eps.df0 <- eps.df00   %>% mutate(
              # Number of SARS-CoV-2 Tests within an Illness Episode
              ntests = sample(1:10, nrow(.), replace=TRUE),
              # Date of Symptom Onset and Meeting Symptom Definition within an Illness Episode
              ONSETDT = sample(seq(as.Date('2020/09/01'), as.Date('2021/03/31'),
                            by="day"), nrow(.), replace=TRUE),
              # Date of Meeting the COVID-19 Symptom Definition
              SYMDT = ONSETDT + sample(0:5, n(), replace=TRUE))%>%
                arrange(USUBJID, ONSETDT)
eps.df <- eps.df0 %>% group_by(USUBJID) %>% arrange(ONSETDT) %>%
  # Label Illness Episodes in Chronological Order
  mutate(EPISODE = 1:n()) %>% arrange(USUBJID)


## Testing Characteristics
# Creating One Row per SARS-CoV-2 Test Result
# All SARS-CoV-2 tests Arise from an Illness Episode that Meets the Symptom Definition
test.df0 <- uncount(eps.df, ntests)
test.df <- test.df0 %>% group_by(USUBJID)%>% mutate(
  # Test Date
  TESTDT = ONSETDT + sample(-3:15, n(), replace=TRUE),
  # Test Result Roughly Based on SARS-CoV-2 PCR Sensitivity and Specificity
  # 1 = positive, 0 = negative
  POSTEST = eps_sars_inf*rbinom(n(),1,0.85)) %>%
  # Only Include Tests while Blinded
  filter(TESTDT<=as.Date('2021/03/31'))
```

We used the demographic and testing data to identify RCT COVID-19 cases (`RCTCASE=1`), or participants who meet the COVID-19 symptom definition and obtain a SARS-CoV-2 positive test. For this RCT, these criteria can be met in either order as long as both dates are within 15 days apart. The remaining RCT participants were right-censored at the end of the blinded phase.

```
## Creating RCT Analysis Dataset


# Identify First SARS-CoV-2 Positive Test from RCT COVID Cases
# (By Construction, All Illness Episodes Met the Symptom Definition)
rct.case.df <- test.df %>%
  # Only take those who test SARS-CoV-2 Positive
  filter(POSTEST==1) %>% group_by(USUBJID)%>%
  slice_min(TESTDT, with_ties=FALSE)

# Adding COVID-19 Case's Testing Information to Demographic Information
rct.df0<- left_join(demo.df, rct.case.df[c("USUBJID","SYMDT","TESTDT","POSTEST")])
rct.df <- rct.df0 %>% rowwise() %>% mutate(
  # Endpoint Date: Unblinding Date or
  # Earliest Date of Meeting Symptom Definition and Positive Test
  RCTADT = if_else(is.na(POSTEST), ENDDT,min(SYMDT,TESTDT))) %>%
  ungroup()%>% mutate(
  # RCT COVID-19 Case Status
  RCTCASE= ifelse(is.na(POSTEST),0,1),
  # Time to COVID or Censoring
```

```
  RCTTT = as.numeric(RCTADT-(DOSE2DT+14)+1))


# RCT Cases (By Construction, All Illness Episodes Met the Symptom Definition)
rct_case_ids<- subset(rct.df, RCTCASE==1)$USUBJID

with(rct.df, table(VACCINE, RCTCASE))
```

```
##         RCTCASE
## VACCINE     0     1
##       0 14415   596
##       1 14872   117
```

From the RCT cohort of 3×10^4 participants, 713 participants were COVID-19 cases.

TND Samples

To create TND samples from RCT testing data, we must manually group SARS-CoV-2 tests into illness episodes since the `EPISODE` variable may not be available or defined identically to our TND analysis plan. In our TND analysis, we define illness episodes as symptomatic periods that last from symptom onset to at least 15 days after symptom onset and are separated by eligible SARS-CoV-2 tests that occur at least 30 days apart.

To create TND samples from RCT testing data, we only retain eligible SARS-CoV-2 tests, which must occur at least 14 days after final vaccination, within ten days after symptom onset, after meeting the symptom definition, and while blinded. In the TND analysis, we define illness episodes as symptomatic periods that last from symptom onset to at least 15 days after symptom onset and are separated by eligible SARS-CoV-2 tests that occur at least 30 days apart. We propose separating illness episodes by eligible SARS-CoV-2 test dates rather than dates of symptom onset and resolution, which would be more difficult to recall and collect. Since the illness episode variable `EPISODE` is defined differently in the RCT, we apply a stricter criteria to the existing RCT illness episode `EPISODE` variable. We define a positive episode as an illness episode that triggers at least one eligible SARS-CoV-2 positive test and a negative episode does not trigger any SARS-CoV-2 positive tests and only triggers eligible SARS-CoV-2 negative tests.

```
# Only Retain TND Eligible Tests
tnd.test.df <- test.df %>% filter(  # Test At Least 14 Days After Vaccination
                                    TESTDT >= (DOSE2DT+14) &
                                    # Test within 10 Days After Symptom Onset
                                    TESTDT >= ONSETDT &
                                    TESTDT <= (ONSETDT +10) &
                                    # Test After Meeting the Symptom Definition
                                    TESTDT >= SYMDT &
                                    # Test While Blinded
                                    TESTDT <= ENDDT) %>%
            # Create Two Week Calendar Time Variable
            mutate(TWOWEEK0 = cut(TESTDT, "2 weeks"))

# Function that identifies tests that are at least 30 days from each other
# Must be applied to data grouped by participant ID and testing date
# inputs:
#      col_vals: the values from a date or numeric column
# outputs: boolean vector, TRUE meaning rows are at least 30 days apart
```

```
#      first value will always be true
#
over30_func <- function(col_vals){
  # MUST BE USED AFTER GROUPED BY SUBJID AND SORTED BY DATE

  # starts with TRUE because have to keep first test
  empty_vec <- c(TRUE)
  nobs <- length(col_vals)
  if(nobs>1){
    for(i in 1:(nobs-1)){
      # are two tests at least 30 days apart?
      val <- col_vals[i+1]-col_vals[i]>=30
      empty_vec <- c(empty_vec, val)
    }
  }
  empty_vec
}
```

Specimen-Based Sample

The first TND sample we create is the specimen-based sample, since all the other sampling methods require illness episodes using our TND definition. In the specimen-based sample, positive and negative episodes are defined as cases and noncases, respectively. Given low SARS-CoV-2 reinfection during the course of the RCT, participants only contribute their first positive episode and all negative episodes. The first positive or negative episode is contributed from each positive or negative episode, respectively.

```
# Take One Test per RCT-Defined Illness Episode
# (Earliest Negative Test if All Negative Tests, or Earliest Positive Test)
spec.df0  <- tnd.test.df %>% group_by(USUBJID, EPISODE)%>%
  arrange(USUBJID, TESTDT) %>% slice(which.max(POSTEST))

# Only Retain Illness Episodes That Are at Least 30 Days Apart
spec.df01 <- spec.df0 %>% group_by(USUBJID) %>%
  arrange(TESTDT)%>% mutate(SEPILL = over30_func(TESTDT))
spec.df02 <- spec.df01 %>% filter(SEPILL==TRUE)

# Only Retain Earliest Positive Episode
spec.df <- spec.df02[!(duplicated(spec.df02[c("USUBJID","POSTEST")]) &
                                        spec.df02$POSTEST==1),] %>%
  mutate(TWOWEEK = droplevels(TWOWEEK0))

with(spec.df, table(VACCINE, POSTEST))
```

```
##         POSTEST
## VACCINE    0    1
##       0 1709  488
##       1 1644   94
```

The specimen-based sample has 3935 participants and 582 cases.

Random Specimen-Based Sample

In the random specimen-based sample, we randomly select one eligible SARS-CoV-2 test from each illness episode included in the specimen-based sample instead of choosing the earliest SARS-CoV-2 positive test from a positive episode and earliest SARS-CoV-2 negative test from a negative episode.

```
# Randomly Choosing One Test Per Illness Episode
# Use Same Participants and Illness Visits from Specimen-Basede Sample
# So Choice of Test is Only Source of Randomness
rspec.df0 <- left_join(spec.df[c("USUBJID","EPISODE")], tnd.test.df)
rspec.df <- rspec.df0 %>% group_by(USUBJID, EPISODE) %>%
  slice_sample(n = 1, replace = FALSE) %>%
  mutate(TWOWEEK = droplevels(TWOWEEK0))

with(rspec.df, table(VACCINE, POSTEST))
```

```
##         POSTEST
## VACCINE    0    1
##       0 1767  430
##       1 1657   81
```

The random specimen-based sample has 3935 participants and 511 cases.

Participant-Based Sample without Censoring for COVID-19

Next, we create the participant-based sample without censoring, in which cases are participants with at least one positive episode and contribute their earliest eligible SARS-CoV-2 positive test from a positive episode and noncases are participants with at least one negative episode (even if they also had a positive episode during the study) and contribute their earliest eligible SARS-CoV-2 negative test from a negative episode.

```
part.neg.df <- spec.df %>% filter(POSTEST==0) %>%
  group_by(USUBJID) %>% arrange(USUBJID, TESTDT) %>% slice(which.min(TESTDT))

# Put negative and positive episodes together
# Particpiant can have many negative episodes but only one positive episode
part.woc.df0 <- rbind((spec.df %>% filter(POSTEST==1)), part.neg.df)

part.woc.df <- part.woc.df0 %>%
  mutate(TWOWEEK = droplevels(TWOWEEK0))

with(part.woc.df , table(VACCINE, POSTEST))
```

```
##         POSTEST
## VACCINE    0    1
##       0 1372  488
##       1 1335   94
```

The participant-based sample without censoring has 3289 participants and 582 cases.

Participant-Based Sample with Censoring for COVID-19

Lastly, we create the participant-based sample with censoring, in which cases are participants with at least one positive episode and contribute their earliest eligible SARS-CoV-2 positive test from a positive episode and noncases are participants with no positive episodes and only negative episodes and contribute their earliest eligible SARS-CoV-2 negative test from a negative episode.

```
part.wcc.df0  <- spec.df %>% group_by(USUBJID) %>% arrange(USUBJID, TESTDT) %>%
  slice(which.max(POSTEST))

part.wcc.df <- part.wcc.df0 %>%
  mutate(TWOWEEK = droplevels(TWOWEEK0))

with(part.wcc.df , table(VACCINE, POSTEST))
```

```
##         POSTEST
## VACCINE    0    1
##       0 1339  488
##       1 1326   94
```

The participant-based sample with censoring has 3247 participants and 582 cases.

Non-SARS-CoV-2 Illness

One of the core TND assumptions for valid inference is that vaccinated and unvaccinated individuals in the healthcare-seeking population with the same demographic and clinical characteristics have the same probability of meeting the COVID-19 symptom definition and testing SARS-CoV-2 negative. In the manuscript, we evaluate if any COVID-19 vaccines affect the probability of experiencing non-SARS-CoV-2 illness (meeting the symptom definition and testing negative), which would violate this assumption. Participants in the RCT cohort have non-SARS-CoV-2 illness if they have negative episode (meet the symptom definition and have only negative eligible SARS-CoV-2 negative tests). There is no censoring for positive episodes.

```
## Adding Non-SARS-CoV-2  Illness Variables

# Take Earliest Negative Episodes per Participant
part.neg.df <- part.neg.df %>% mutate(
  #Non-SARS-CoV-2 Illness
  NEG = 1-POSTEST,
  # Date of Non-SARS-CoV-2 Illness
  NEGADT0= TESTDT
)

rct.df<-left_join(rct.df, part.neg.df[c("USUBJID","NEG","NEGADT0")])
rct.df <- rct.df %>% rowwise() %>% mutate(
    # Non-SARS-CoV-2 Date is Date of Negative Episode or End of Study
    NEGADT = min(NEGADT0, ENDDT, na.rm=TRUE),
    # Time to Non-SARS-CoV-2 Illness or Censoring
    NEGTT = as.numeric(NEGADT-(DOSE2DT+14)+1),
    # Non-SARS-CoV-2 Illness Status
    NEG = ifelse(!is.na(NEGADT0)&NEGADT0==NEGADT,1,0))

with(rct.df , table(VACCINE, NEG))
```

```
##         NEG
## VACCINE     0     1
##       0 13639  1372
##       1 13654  1335
```

The RCT cohort has 2707 participants with non-SARS-CoV-2 illness.

Statistical Analyses

In this next section, we estimate vaccine efficacy against COVID-19 using the RCT cohort, vaccine effectiveness against COVID-19 using the four TND samples and two statistical methods, and vaccine efficacy against non-SARS-CoV-2 illness using the RCT cohort.

RCT COVID-19 Vaccine Efficacy

```
rct.mod <- coxph(Surv(RCTTT, RCTCASE)~VACCINE, data=rct.df)
rct.ve<- round(1-summary(rct.mod)$conf.int["VACCINE","exp(coef)"], 3)*100
rct.veCIlow <- round(1-summary(rct.mod)$conf.int["VACCINE","upper .95"], 3)*100
rct.veCIhigh <- round(1-summary(rct.mod)$conf.int["VACCINE","lower .95"],3)*100
```

From the RCT cohort of 3×10^4 participants, we estimate vaccine efficacy, defined as 1 minus the hazard ratio (vaccine vs. placebo), using an unadjusted Cox proportional hazards regression model. RCT vaccine efficacy against COVID-19 is 80.7% (95% CI: 76.4, 84.1).

TND COVID-19 Vaccine Effectiveness Using Semiparametric Logistic Regression

We use the `causalglm` R package developed by Lars van der Laan (https://tlverse.org/causalglm/), which implements semiparametric and nonparametric generalized linear models using targeted maximum likelihood estimation to conduct causal inference on heterogeneous treatment effects. The package relies on `tlverse/tmle3` for targeted maximum likelihood estimation and `tlverse/sl3` and `tlverse/hal9001` for machine-learning. We must load the most up to date version of these packages from Github.

```
#  Install devtools CRAN package
if(!require(devtools)) {
  install.packages(devtools)
}

# Install Github Packages
devtools::install_github("tlverse/causalglm")
devtools::install_github("tlverse/hal9001@master")
devtools::install_github("tlverse/tmle3@general_submodels_devel")
devtools::install_github("tlverse/sl3@Larsvanderlaan-formula_fix")
```

For our semiparametric logistic regression, we apply the partially linear first-order smooth highly adaptive lasso to partially linear logistic regression model of vaccination status (vaccine vs. placebo) on case status, adjusted for age, sex assigned at birth, race/ethnicity, region, comorbidities, and two-week intervals and allowing for two-way interactions.

To run a semiparametric logistic regression, we use the `spglm` function in `causalglm`, which currently requires all variables in the input dataset to be numeric. We use `model.matrix` to convert all character and factor variables into numeric or indicator variables for each TND sample.

```
## Convert Data to Only Contain Numeric/Indicator Variables (spglm requirement)
# Use model.matrix to convert character/factor variables for each TND sample
part.woc.semi.model.df <- as.data.frame(model.matrix(~VACCINE + POSTEST + AGE + SEX +
                      POC + REGION + COMORB + TWOWEEK, data=part.woc.df))
part.wcc.semi.model.df <- as.data.frame(model.matrix(~VACCINE + POSTEST + AGE + SEX +
                      POC + REGION + COMORB + TWOWEEK, data=part.wcc.df))
spec.semi.model.df <- as.data.frame(model.matrix(~VACCINE + POSTEST + AGE + SEX +
                      POC + REGION + COMORB + TWOWEEK, data=spec.df))
rspec.semi.model.df <- as.data.frame(model.matrix(~VACCINE + POSTEST + AGE + SEX +
                      POC + REGION + COMORB + TWOWEEK, data=rspec.df))

## Run Semiparametric Model For Each TND Sample (May Take A Few Minutes)
part.woc.semi.mod <- spglm( ~1, part.woc.semi.model.df,
                 # Vector of Adjustment Covariate Names from Converted Dataset
                 W = setdiff(names(part.woc.semi.model.df),
                             c("(Intercept)","VACCINE","POSTEST")),
                 # Regression Predictor of Interest
                 A = "POSTEST",
                 # Regression Outcome
                 Y = "VACCINE",
                 estimand = "OR",
                 HAL_args_YOW = list(smoothness_orders = 1,
                                     max_degree = 2,
                                     num_knots = c(10, 10)))

part.wcc.semi.mod <- spglm( ~1, part.wcc.semi.model.df,
                 # Vector of Adjustment Covariate Names from Converted Dataset
                 W = setdiff(names(part.wcc.semi.model.df),
                             c("(Intercept)","VACCINE","POSTEST")),
                 # Regression Predictor of Interest
                 A = "POSTEST",
                 # Regression Outcome
                 Y = "VACCINE",
                 estimand = "OR",
                 HAL_args_YOW = list(smoothness_orders = 1,
                                     max_degree = 2,
                                     num_knots = c(10, 10)))

spec.semi.mod <- spglm(  # ~ 1 for No Effect Modification,
                         # ~1 + Covariate Names for Effect Modification
                         formula= ~1,
                         # Converted Dataset
                         spec.semi.model.df,
                 # Vector of Adjustment Covariate Names from Converted Dataset
                 W = setdiff(names(spec.semi.model.df),
                             c("(Intercept)","VACCINE","POSTEST")),
                 # Regression Predictor of Interest
                 A = "POSTEST",
                 # Regression Outcome
                 Y = "VACCINE",
```

```
                estimand = "OR",
                HAL_args_YOW = list(smoothness_orders = 1,
                                    max_degree = 2,
                                    num_knots = c(10, 10)))

rspec.semi.mod <- spglm( # ~ 1 for No Effect Modification,
                         # ~1 + Covariate Names for Effect Modification
                         formula= ~1,
                         # Converted Dataset
                         data= rspec.semi.model.df,
                # Vector of Adjustment Covariate Names from Converted Dataset
                W = setdiff(names(rspec.semi.model.df),
                            c("(Intercept)","VACCINE","POSTEST")),
                # Regression Predictor of Interest
                A = "POSTEST",
                # Regression Outcome
                Y = "VACCINE",
                estimand = "OR",
                HAL_args_YOW = list(smoothness_orders = 1,
                                    max_degree = 2,
                                    num_knots = c(10, 10)))
```

```
# Example Output
summary(part.woc.semi.mod)
```

```
## A causalglm fit object obtained from spglm for the estimand OR with formula:
## log OR(W) = -1.62 * (Intercept)
##
## Coefficient estimates and inference:
##      type       param tmle_est        se     lower     upper   psi_exp
##    <char>      <char>    <num>     <num>     <num>     <num>     <num>
## 1:     OR (Intercept) -1.62489 0.1194677 -1.859042 -1.390737 0.1969334
##    lower_exp upper_exp  Z_score p_value
##        <num>     <num>    <num>   <num>
## 1: 0.1558218 0.2488917 13.60108       0
```

`summary(spglm)` function reports summary statistics for the sample log conditional odds ratio of vaccination by case status, adjusting for covariates. `tmle_est` is the log conditional odds ratio and `psi_exp` is the conditional odds ratio. `lower_exp` and `upper_exp` are the exponentiated lower and upper bounds of the log conditional odds ratio 95% confidence interval.

TND vaccine effectiveness against virologically-confirmed symptomatic COVID-19 using the semiparametric logistic regression is as follows:

- Participant-Based Sample without Censoring for COVID-19: 80.3% (95% CI: 75.1, 84.4)
- Participant-Based Sample with Censoring for COVID-19: 80.6% (95% CI: 75.5, 84.7)
- Specimen-Based Sample without Censoring for COVID-19: 80% (95% CI: 74.8, 84.2)
- Random Specimen-Based Sample without Censoring for COVID-19: 79.9% (95% CI: 74.3, 84.3)

TND COVID-19 Vaccine Effectiveness Using Ordinary Logistic Regression

We also estimate the TND sample odds ratio of vaccination by case status, adjusted for age, sex assigned at birth, race/ethnicity, region, comorbidities, and two-week intervals main effects using an ordinary logistic regression.

```
part.woc.ord.mod <- glm(VACCINE ~ POSTEST + AGE + SEX + POC + REGION +
                        COMORB + TWOWEEK, data=part.woc.df, family = "binomial")
part.wcc.ord.mod <- glm(VACCINE ~ POSTEST + AGE + SEX + POC + REGION +
                        COMORB + TWOWEEK, data=part.wcc.df, family = "binomial")
spec.ord.mod <- glm(VACCINE ~ POSTEST + AGE + SEX + POC + REGION +
                        COMORB + TWOWEEK, data=spec.df, family = "binomial")
rspec.ord.mod <- glm(VACCINE ~ POSTEST + AGE + SEX + POC + REGION +
                        COMORB + TWOWEEK, data=rspec.df, family = "binomial")
```

TND vaccine effectiveness against virologically-confirmed symptomatic COVID-19 using the ordinary logistic regression is as follows:

- Participant-Based Sample without Censoring for COVID-19: 80.6% (95% CI: 75.5, 84.7)
- Participant-Based Sample with Censoring for COVID-19: 81% (95% CI: 75.9, 85)
- Specimen-Based Sample without Censoring for COVID-19: 80.2% (95% CI: 75, 84.3)
- Random Specimen-Based Sample without Censoring for COVID-19: 80.1% (95% CI: 74.5, 84.5)

RCT Non-SARS-CoV-2 Illness Vaccine Efficacy

From the RCT cohort of 3×10^4 participants, we estimate vaccine efficacy against non-SARS-CoV-2 illness , defined as 1 minus the hazard ratio (vaccine vs. placebo), using an unadjusted Cox proportional hazards regression model. While we should evaluate vaccine efficacy in every subgroup adjusted for in a typical TND analysis, for simplicity we only estimated vaccine efficacy in the overall cohort and two age subgroups, less than 60 years and at least 60 years.

```
rctneg_mod <- coxph(Surv(NEGTT, NEG)~VACCINE, data=rct.df)
rctnegy_mod <- coxph(Surv(NEGTT, NEG)~VACCINE, data=subset(rct.df, AGE<60))
rctnego_mod <- coxph(Surv(NEGTT, NEG)~VACCINE, data=subset(rct.df, AGE>=60))
```

RCT vaccine efficacy against non-SARS-CoV-2 illness is as follows:

- Overall Cohort: 2.7% (95% CI: -5, 9.7, p = 0.484)
- Less than 60 Subgroup: 1.9% (95% CI: -7.6, 10.5, p = 0.686)
- 60 and Over Subgroup: 4.2% (95% CI: -9.2, 15.9, p = 0.524)

There is no evidence that the vaccine affects non-SARS-CoV-2 illness, which is consistent with our data-generating process.

Summary of Results

We use forest plots to visualize the COVID-19 RCT vaccine efficacy and TND vaccine effectiveness estimates together and the Non-SARS-CoV-2 Illness RCT vaccine efficacy by subgroup.