Aim 2.2 & 2.3 — Urbanicity, Uninsurance, and Maternal Vaccination (Updated)

Ruiqi Li

2025-10-22

Contents

0.1 0. Import & normalize names

```
vacc_path <- "vaccination.csv"
ins_path  <- "insurance_summary_by_state_2012_2022.csv"
nchs_path <- "state_urban_index_2013_2023.csv"

vacc_raw <- readr::read_csv(vacc_path, show_col_types = FALSE)
ins_raw  <- readr::read_csv(ins_path,  show_col_types = FALSE)
nchs_raw <- readr::read_csv(nchs_path, show_col_types = FALSE)

norm_names <- function(df){
  nm <- names(df)
  nm <- gsub("[^A-Za-z0-9]+","_", nm)
  nm <- gsub("_+","_", nm)
  nm <- tolower(nm)
  names(df) <- nm
  df
}
vacc_raw <- norm_names(vacc_raw)
ins_raw  <- norm_names(ins_raw)
nchs_raw <- norm_names(nchs_raw)

list(
  vacc_cols = names(vacc_raw),
  ins_cols  = names(ins_raw),
  nchs_cols = names(nchs_raw)
)
```

```
## $vacc_cols
##  [1] "vaccine"                       "geography_type"
##  [3] "geography"                     "survey_year_influenza_season"
##  [5] "dimension_type"                "dimension"
##  [7] "estimate_"                     "x95_ci_"
##  [9] "sample_size"                   "state"
##
## $ins_cols
## [1] "year"      "state"     "insured"   "uninsured"
##
## $nchs_cols
## [1] "st_abbrev" "index2013" "index2023" "change"
```

0.2 1) Build state–year outcomes (de-dup → pivot)

```
# Adaptive column name selector
pick_col <- function(nms, ...) {
  pats <- unlist(list(...))
  hits <- NULL
  for (p in pats) {
    idx <- grep(p, nms, ignore.case = TRUE, perl = TRUE)
    if (length(idx)) { hits <- idx[1]; break }
  }
  if (is.null(hits)) return(NA_character_)
  nms[hits]
}

nms <- names(vacc_raw)

year_col     <- pick_col(nms, "^survey.*year", "influenza.*season", "^year$")
estimate_col <- pick_col(nms, "^estimate", "estimate.*percent", "value$")
sample_col   <- pick_col(nms, "^sample.*size", "n$")
state_col    <- pick_col(nms, "^state$", "state.*abbr", "location")
vax_col      <- pick_col(nms, "^vaccine$", "vax", "vaccine.*type")
dimtype_col  <- pick_col(nms, "^dimension.*type$", "dim.*type")
dim_col      <- pick_col(nms, "^dimension$", "^dim$")

# Print the matching results to confirm the mapping.
```

```r
cat("Mapped columns:\n",
    "year      ->", year_col, "\n",
    "estimate  ->", estimate_col, "\n",
    "sample    ->", sample_col, "\n",
    "state     ->", state_col, "\n",
    "vaccine   ->", vax_col, "\n",
    "dim.type  ->", dimtype_col, "\n",
    "dimension ->", dim_col, "\n")
```

```
## Mapped columns:
##  year      -> survey_year_influenza_season
##  estimate  -> estimate_
##  sample    -> sample_size
##  state     -> state
##  vaccine   -> vaccine
##  dim.type  -> dimension_type
##  dimension -> dimension
```

```r
# Basic check: at least year / estimate / sample / state / vaccine
need <- c(year_col, estimate_col, sample_col, state_col, vax_col)
if (any(is.na(need))) {
  stop("vaccination.csv is missing key columns, please run `names(vacc_raw)`
       first to see the actual column names and tell me the first 5 rows")
}

# Placed before the same code block: universal value/year conversion
num_any  <- function(x) {
  if (is.numeric(x)) as.numeric(x) else
    readr::parse_number(as.character(x))
}
year_any <- function(x) {
  if (is.numeric(x)) as.integer(round(x)) else as.integer(readr::parse_number(as.character(x)))
}

# Construct vacc_overall (compatible character/numeric value
vacc_overall0 <- vacc_raw %>%
  transmute(
    state   = toupper(.data[[state_col]]),
    year    = year_any(.data[[year_col]]),
    vaccine = as.character(.data[[vax_col]]),
    est     = num_any(.data[[estimate_col]]),
    n       = num_any(.data[[sample_col]]),
    dimtype = if (!is.na(dimtype_col)) .data[[dimtype_col]] else NA_character_,
    dim     = if (!is.na(dim_col))     .data[[dim_col]]     else NA_character_
  )

# If Dimension column exists, filter Age: >=18 Years; otherwise skip filtering
if (!all(is.na(vacc_overall0$dimtype)) && !all(is.na(vacc_overall0$dim))) {
  vacc_overall <- vacc_overall0 %>%
    filter(dimtype %in% c("Age","age"), dim %in% c(">=18 Years", ">= 18 Years"))
  if (nrow(vacc_overall) == 0L) {
    message("Age: >=18 Years not found in Dimension column,
            fallback to use unfiltered version")
    vacc_overall <- vacc_overall0
  }
} else {
  message("Dimension column not detected, use the overall sample directly")
  vacc_overall <- vacc_overall0
}

# De-aggregation (preventing multiple rows of vaccines in the same state-year)
vacc_overall_dedup <- vacc_overall %>%
  group_by(state, year, vaccine) %>%
  summarise(
    est = if (all(is.na(n))) mean(est, na.rm = TRUE)
    else stats::weighted.mean(est, n, na.rm = TRUE),
    n   = sum(n, na.rm = TRUE),
    .groups = "drop"
  )

# Map vaccine ID to flu/tdap
vacc_overall_dedup <- vacc_overall_dedup %>%
  mutate(vax = case_when(
    grepl("influenza|flu", vaccine, ignore.case = TRUE) ~ "flu",
    grepl("tdap",          vaccine, ignore.case = TRUE) ~ "tdap",
    TRUE ~ NA_character_
  )) %>%
  filter(!is.na(vax))

vacc_wide <- vacc_overall_dedup %>%
  select(state, year, vax, est, n) %>%
  tidyr::pivot_wider(names_from = vax, values_from = c(est, n), names_sep = "_") %>%
  dplyr::rename(
    vacc_flu_pct  = est_flu,
    vacc_tdap_pct = est_tdap,
    n_flu         = n_flu,
    n_tdap        = n_tdap
  )

# quick look
print(head(vacc_wide, 5))
```

```
## # A tibble: 5 x 6
##   state  year vacc_flu_pct vacc_tdap_pct n_flu n_tdap
##   <chr> <int>        <dbl>         <dbl> <dbl>  <dbl>
## 1 AK     2012         49.2            NA   852     NA
## 2 AK     2013         57.6            NA  1240     NA
## 3 AK     2014         61.8            NA  1145     NA
## 4 AK     2015         56.7            NA  1154     NA
## 5 AK     2016         57.4            NA  1133     NA
```

0.3 2) Predictors: Uninsurance & NCHS

```r
stopifnot(all(c("state","year","insured","uninsured") %in% names(ins_raw)))
ins <- ins_raw %>%
  transmute(
    state = toupper(state),
    year  = as.integer(year),
    insured = num(insured),
    uninsured = num(uninsured),
    uninsured_rate = uninsured / (insured + uninsured)
  )

stopifnot(all(c("st_abbrev","index2013","index2023","change") %in% names(nchs_raw)))
nchs <- nchs_raw %>%
  transmute(
    state = toupper(st_abbrev),
    nchs_2013   = num(index2013),
    nchs_2023   = num(index2023),
    nchs_change = num(change)
  )
```

0.4 3) Merge → state×year panel + harmonize years

```r
dat <- vacc_wide %>%
  inner_join(ins,  by = c("state","year")) %>%
  left_join(nchs, by = "state") %>%
  mutate(
    vacc_flu_pct  = num(vacc_flu_pct),
    vacc_tdap_pct = num(vacc_tdap_pct),
    n_flu         = num(n_flu),
    n_tdap        = num(n_tdap)
  ) %>%
  arrange(state, year)

# keep years observed for BOTH vaccines to balance samples in Aim 2.3
years_keep <- intersect(unique(dat$year[!is.na(dat$vacc_flu_pct)]),
                        unique(dat$year[!is.na(dat$vacc_tdap_pct)]))
dat <- dat %>% filter(year %in% years_keep)

summary(dat[,c("vacc_flu_pct","vacc_tdap_pct","uninsured_rate","nchs_2013")])
```

```
##   vacc_flu_pct  vacc_tdap_pct   uninsured_rate      nchs_2013
##  Min.   :23.4   Min.   : 8.20   Min.   :0.02530   Min.   :1.000
##  1st Qu.:53.0   1st Qu.:62.50   1st Qu.:0.06785   1st Qu.:2.616
##  Median :59.5   Median :74.90   Median :0.09817   Median :3.022
##  Mean   :59.4   Mean   :68.47   Mean   :0.10547   Mean   :3.193
##  3rd Qu.:66.1   3rd Qu.:81.20   3rd Qu.:0.13198   3rd Qu.:3.937
##  Max.   :82.2   Max.   :90.20   Max.   :0.25953   Max.   :4.989
##                 NA's   :178                       NA's   :8
```

	Flu mean ~ NCHS (between)	Tdap mean ~ NCHS (between)
(Intercept)	58.184	69.103
	(1.346)[<0.001]	(2.630)[<0.001]
nchs13_z	0.681	4.856
	(1.361)[0.619]	(2.889)[0.106]
Num.Obs.	46	26
R2	0.006	0.105

0.5 4) Aim 2.2 — Between-state association (state means)

```
between_df <- dat %>%
  group_by(state) %>%
  summarise(
    vacc_flu_mean  = wmean_safe(vacc_flu_pct,  n_flu),
    vacc_tdap_mean = wmean_safe(vacc_tdap_pct, n_tdap),
    nchs_2013      = dplyr::first(nchs_2013),
    .groups = "drop"
  ) %>%
  mutate(nchs13_z = as.numeric(scale(nchs_2013)))

m22_flu_between  <- lm(vacc_flu_mean  ~ nchs13_z, data = between_df)
m22_tdap_between <- lm(vacc_tdap_mean ~ nchs13_z, data = between_df)

modelsummary::msummary(
  list("Flu mean ~ NCHS (between)"  = m22_flu_between,
       "Tdap mean ~ NCHS (between)" = m22_tdap_between),
  statistic = "({std.error})[{p.value}]",
  gof_omit = "AIC|BIC|Log.Lik|F|Adj|RMSE"
)
```

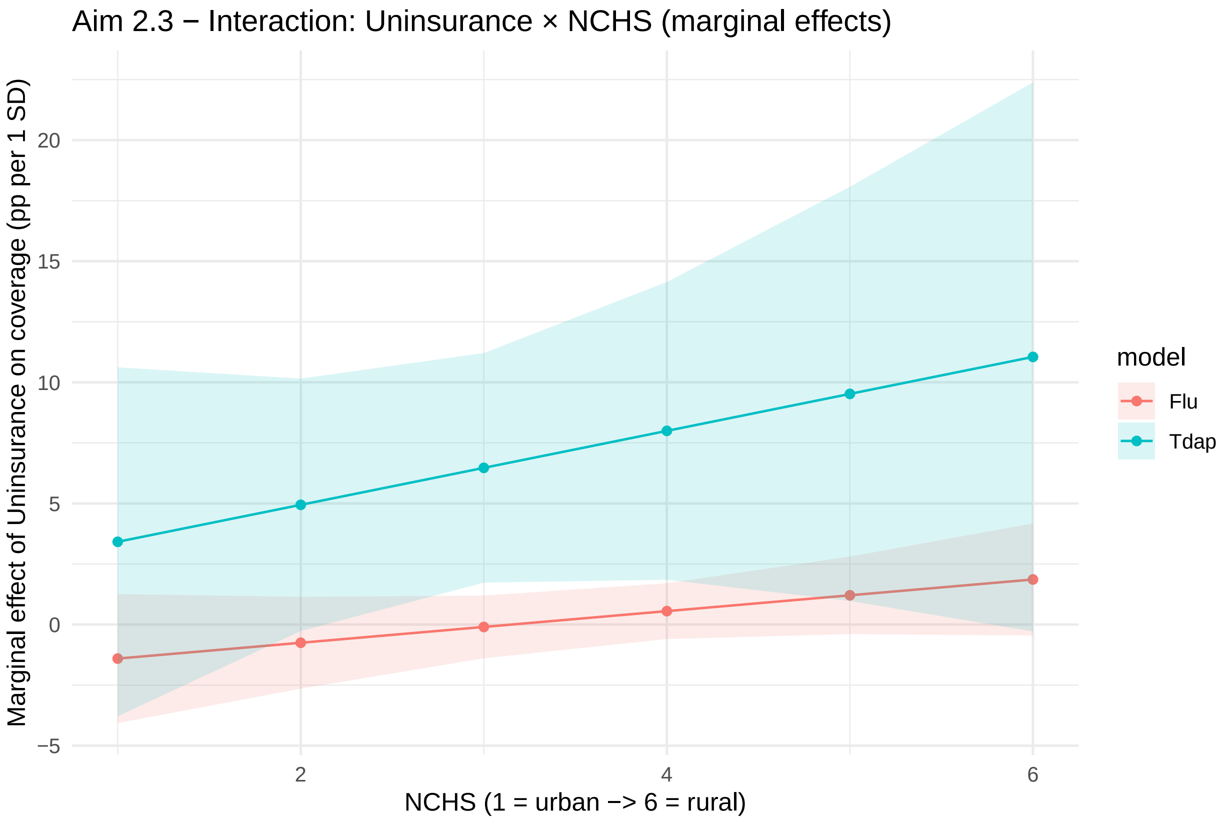
0.6 5) Aim 2.3 — TWFE with interaction (state & year FE)

```
dat <- dat %>%
  mutate(
    unins_z  = as.numeric(scale(uninsured_rate)),
    nchs13_z = as.numeric(scale(nchs_2013))
  )

dat_flu <- dat %>%
  filter(!is.na(vacc_flu_pct), !is.na(n_flu),
         !is.na(unins_z), !is.na(nchs13_z))

dat_tdap <- dat %>%
  filter(!is.na(vacc_tdap_pct), !is.na(n_tdap),
         !is.na(unins_z), !is.na(nchs13_z))

m23_flu <- feols(
  vacc_flu_pct ~ unins_z * nchs13_z | state + year,
  data    = dat_flu,
  weights = ~ n_flu,
  cluster = ~ state
)

m23_tdap <- feols(
  vacc_tdap_pct ~ unins_z * nchs13_z | state + year,
  data    = dat_tdap,
  weights = ~ n_tdap,
  cluster = ~ state
)

fixest::etable(
  m23_flu, m23_tdap,
  se   = "cluster",
  dict = c(unins_z = "Uninsurance (z)",
           nchs13_z = "NCHS 2013 (z)",
           "unins_z:nchs13_z" = "Unins × NCHS13"),
  title = "Aim 2.3 - TWFE with interaction (state & year FE; clustered SE by state)"
)
```

```
##                          m23_flu       m23_tdap
## Dependent Var.:     vacc_flu_pct  vacc_tdap_pct
##
## Uninsurance (z) 0.0271 (0.6218) 6.530* (2.431)
## Unins × NCHS13 0.6254 (0.4335)  1.242 (1.314)
## Fixed-Effects:  ---------------- --------------
## state                        Yes            Yes
## year                         Yes            Yes
## ________________ ________________ ______________
## S.E.: Clustered        by: state      by: state
## Observations                 321            151
## R2                       0.93303        0.97527
## Within R2                0.02077        0.08689
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

0.7 6) Marginal effects of Uninsurance across rurality

```
me_table <- function(mod, dat0, title){
  b   <- coef(mod)
  V   <- vcov(mod)
  b1 <- b["unins_z"]
  b3 <- b["unins_z:nchs13_z"]
  V11 <- V["unins_z","unins_z"]
  V33 <- V["unins_z:nchs13_z","unins_z:nchs13_z"]
  V13 <- V["unins_z","unins_z:nchs13_z"]

  nchs_seq <- 1:6
  mu   <- mean(dat0$nchs_2013, na.rm=TRUE)
  sdv <- sd(dat0$nchs_2013,   na.rm=TRUE)
  z   <- (nchs_seq - mu)/sdv

  tibble(
    NCHS = nchs_seq,
    nchs13_z = z,
    marginal_effect = b1 + b3 * z,
    se = sqrt(V11 + (z^2)*V33 + 2*z*V13),
    lo95 = marginal_effect - 1.96*se,
    hi95 = marginal_effect + 1.96*se,
    model = title
  )
}

me_flu  <- me_table(m23_flu,  dat_flu,  "Flu")
me_tdap <- me_table(m23_tdap, dat_tdap, "Tdap")
me_all  <- dplyr::bind_rows(me_flu, me_tdap)

ggplot(me_all, aes(x = NCHS, y = marginal_effect, color = model)) +
  geom_line() + geom_point() +
  geom_ribbon(aes(ymin = lo95, ymax = hi95, fill = model), alpha = .15, color = NA) +
  labs(x = "NCHS (1 = urban → 6 = rural)",
       y = "Marginal effect of Uninsurance on coverage (pp per 1 SD)",
       title = "Aim 2.3 - Interaction: Uninsurance × NCHS (marginal effects)") +
  theme_minimal()
```

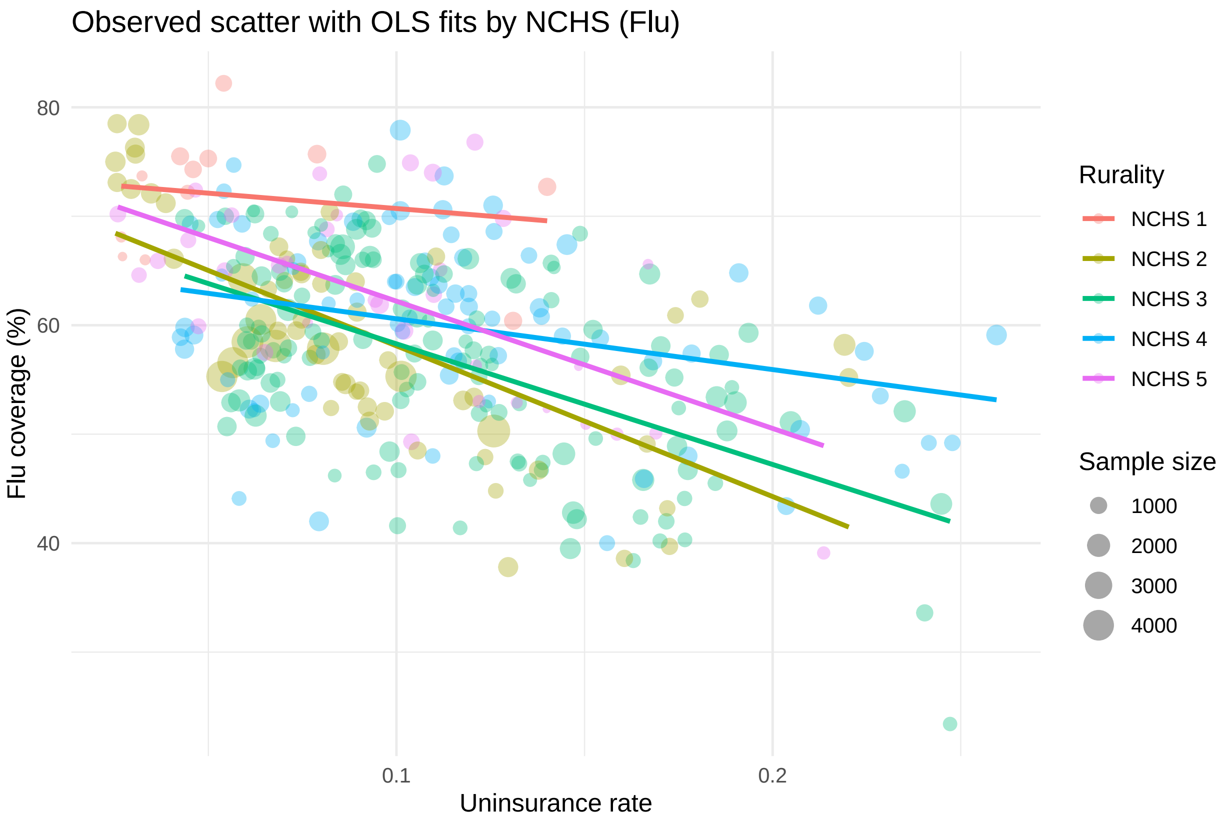


0.8 6A) Observation scatter + linear fit grouped by NCHS (one sheet each for Flu/Tdap)

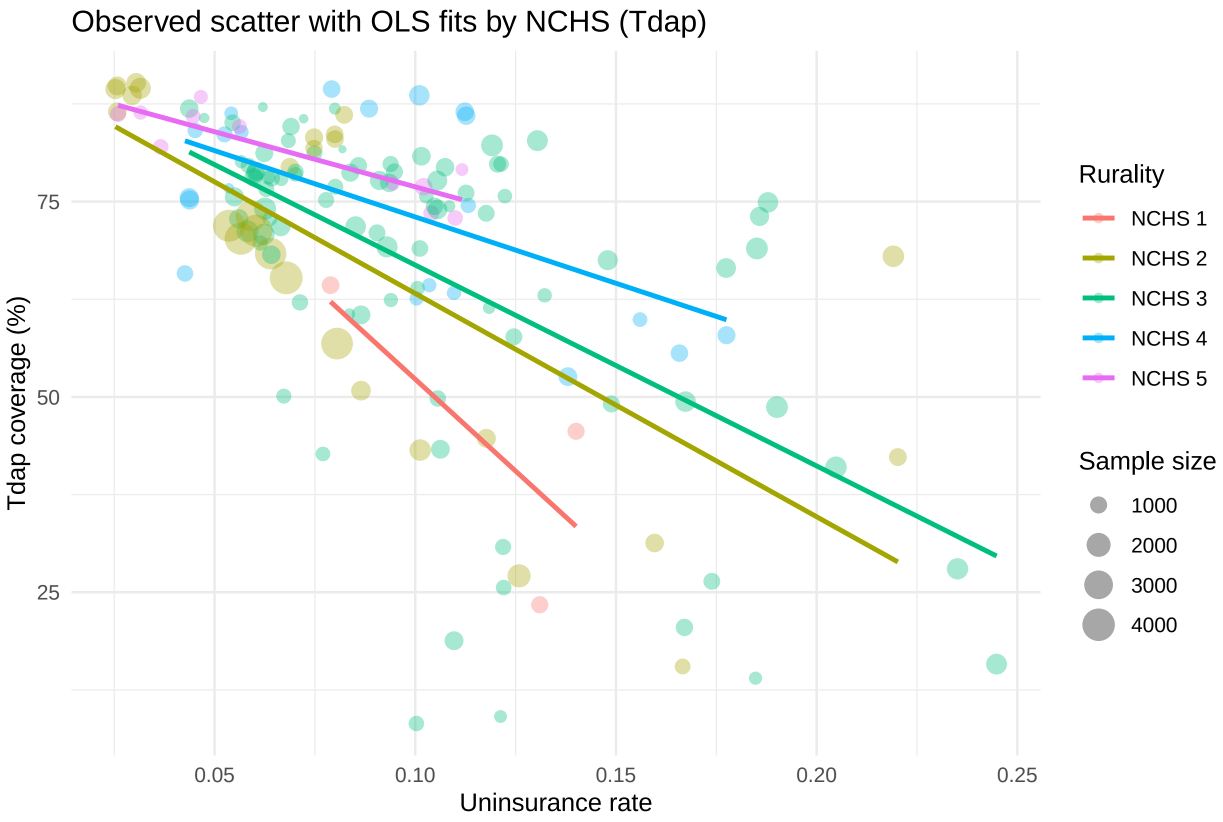
Demonstrate the relationship between uninsurance and coverage under different rural levels in real observations. Point size = sample size

```r
# helper
num_any  <- function(x) if (is.numeric(x)) as.numeric(x) else readr::parse_number(as.character(x))
norm_names <- function(df){
  nm <- names(df)
  nm <- gsub("[^A-Za-z0-9]+","_", nm); nm <- gsub("_+","_", nm); nm <- tolower(nm)
  names(df) <- nm; df
}
pick <- function(nm, cands){
  cand <- intersect(tolower(cands), nm);if (length(cand)) cand[1]
  else NA_character_ }

# Rebuild nchs_lookup from original NCHS CSV
nchs_raw <- readr::read_csv("state_urban_index_2013_2023.csv",
                            show_col_types = FALSE) |> norm_names()

cand_state <- c("st_abbrev","state","state_abbrev","state_abbreviation","st")
cand_2013  <- c("index2013","index_2013","nchs2013","nchs_2013","score2013")
cand_2023  <- c("index2023","index_2023","nchs2023","nchs_2023","score2023")

scol <- pick(names(nchs_raw), cand_state)
c13  <- pick(names(nchs_raw), cand_2013)
c23  <- pick(names(nchs_raw), cand_2023)

# If it stops here, it means that the column name in the file is not among the candidates, and the real
stopifnot(!is.na(scol), !is.na(c13), !is.na(c23))

nchs_lookup <- nchs_raw |>
  dplyr::transmute(
    state      = toupper(.data[[scol]]),
    nchs_2013  = num_any(.data[[c13]]),
    nchs_2023  = num_any(.data[[c23]])
  ) |>
  dplyr::mutate(nchs_base = dplyr::coalesce(nchs_2013, nchs_2023))

# Use it to make a special panel for 6A drawing (no year intersection)
panel_plot <- vacc_wide %>%
  dplyr::inner_join(ins,  by = c("state","year")) %>%
  dplyr::left_join(nchs_lookup %>%
                     dplyr::select(state, nchs_2013, nchs_2023, nchs_base), by = "state") %>%
  dplyr::mutate(
    vacc_flu_pct  = as.numeric(vacc_flu_pct),
    vacc_tdap_pct = as.numeric(vacc_tdap_pct),
    n_flu         = as.numeric(n_flu),
    n_tdap        = as.numeric(n_tdap)
  )

# plot
lab_nchs <- function(x) factor(x, levels = 1:6, labels = paste0("NCHS ", 1:6))
```

```r
panel_plot_grp <- panel_plot %>%
  dplyr::mutate(
    # Map continuous indices to 1..6, and truncate out-of-bounds indices to [1,6]
    nchs_grp = dplyr::case_when(
      is.na(nchs_base) ~ NA_integer_,
      TRUE ~ pmin(6L, pmax(1L, as.integer(round(nchs_base))))
    ),
    nchs_lab = lab_nchs(nchs_grp)
  )


flu_plot_df <- panel_plot_grp %>%
  dplyr::filter(!is.na(vacc_flu_pct),
                !is.na(uninsured_rate),
                !is.na(nchs_grp))

ggplot(flu_plot_df, aes(uninsured_rate, vacc_flu_pct, color = nchs_lab)) +
  geom_point(aes(size = n_flu), alpha = .35) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_size_continuous(name = "Sample size") +
  scale_color_discrete(name = "Rurality") +
  labs(x = "Uninsurance rate", y = "Flu coverage (%)",
       title = "Observed scatter with OLS fits by NCHS (Flu)") +
  theme_minimal()
```



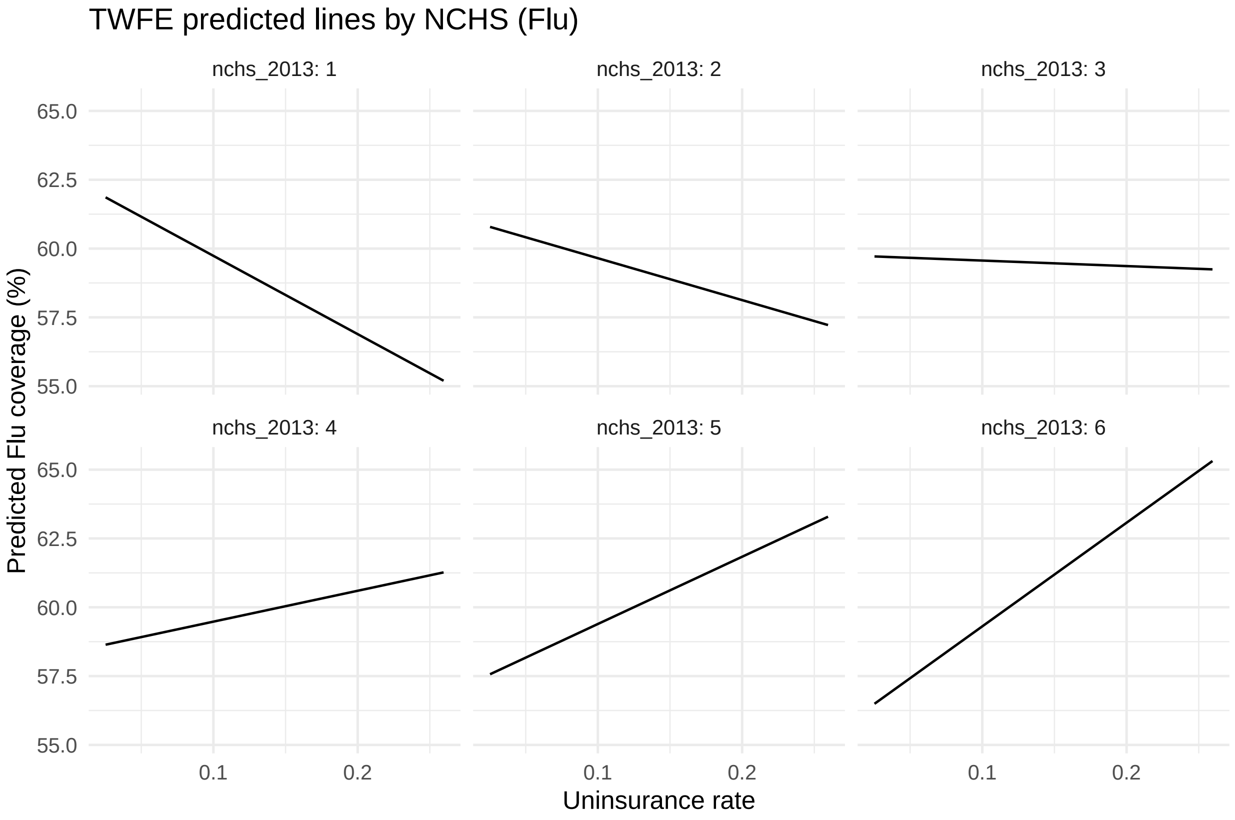
```
tdap_plot_df <- panel_plot_grp %>%
  dplyr::filter(!is.na(vacc_tdap_pct),
                !is.na(uninsured_rate),
                !is.na(nchs_grp))

ggplot(tdap_plot_df, aes(uninsured_rate, vacc_tdap_pct, color = nchs_lab)) +
  # If n_tdap is missing, the default size will be given..
  geom_point(aes(size = tidyr::replace_na(n_tdap, 1000)), alpha = .35) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_size_continuous(name = "Sample size") +
  scale_color_discrete(name = "Rurality") +
  labs(x = "Uninsurance rate", y = "Tdap coverage (%)",
       title = "Observed scatter with OLS fits by NCHS (Tdap)") +
  theme_minimal()
```



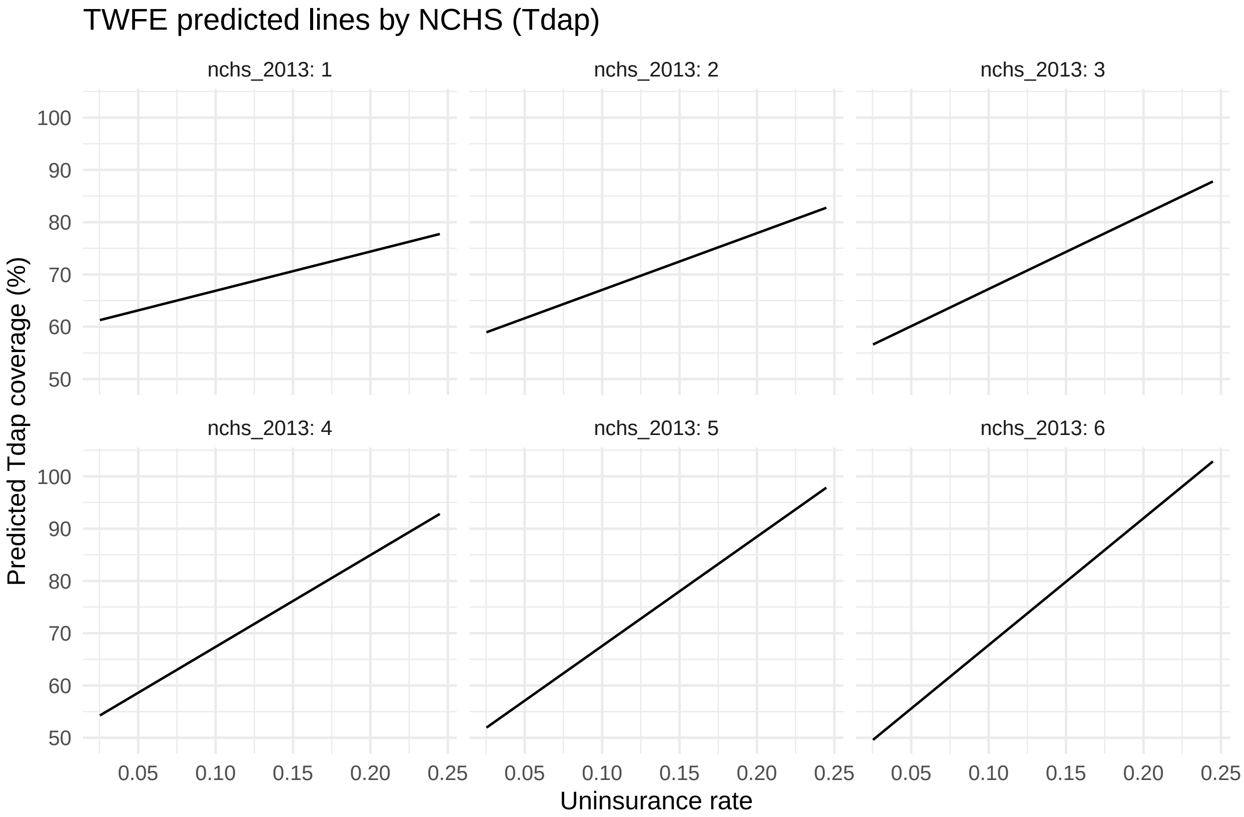
0.9 6B) Prediction line (TWFE model) × NCHS facet

Use the estimated model to draw "predicted coverage curves changing with the uninsured rate under different NCHS (1..6)"; and mutual confirmation with 6A's "observation + OLS"

```
# Generate prediction grid
mk_grid <- function(dat){
  g <- expand.grid(
    nchs_2013 = 1:6,
    uninsured_rate = seq(min(dat$uninsured_rate, na.rm=TRUE),
                         max(dat$uninsured_rate, na.rm=TRUE), length.out = 50),
    year  = median(dat$year, na.rm=TRUE),
    state = dat$state[1]
  )
  g <- dplyr::mutate(
    g,
    nchs13_z = (nchs_2013 - mean(dat$nchs_2013, na.rm=TRUE))/sd(dat$nchs_2013, na.rm=TRUE),
    unins_z  = (uninsured_rate - mean(dat$uninsured_rate,
                                      na.rm=TRUE))/sd(dat$uninsured_rate, na.rm=TRUE)
  )
  g
}

grid_flu  <- mk_grid(dat_flu)
grid_flu$pred  <- predict(m23_flu,  newdata = grid_flu)

grid_tdap <- mk_grid(dat_tdap)
grid_tdap$pred <- predict(m23_tdap, newdata = grid_tdap)

# Faceted prediction line
ggplot(grid_flu, aes(uninsured_rate, pred)) +
  geom_line() +
  facet_wrap(~ nchs_2013, nrow = 2, labeller = label_both) +
  labs(x="Uninsurance rate", y="Predicted Flu coverage (%)",
       title="TWFE predicted lines by NCHS (Flu)") +
  theme_minimal()
```



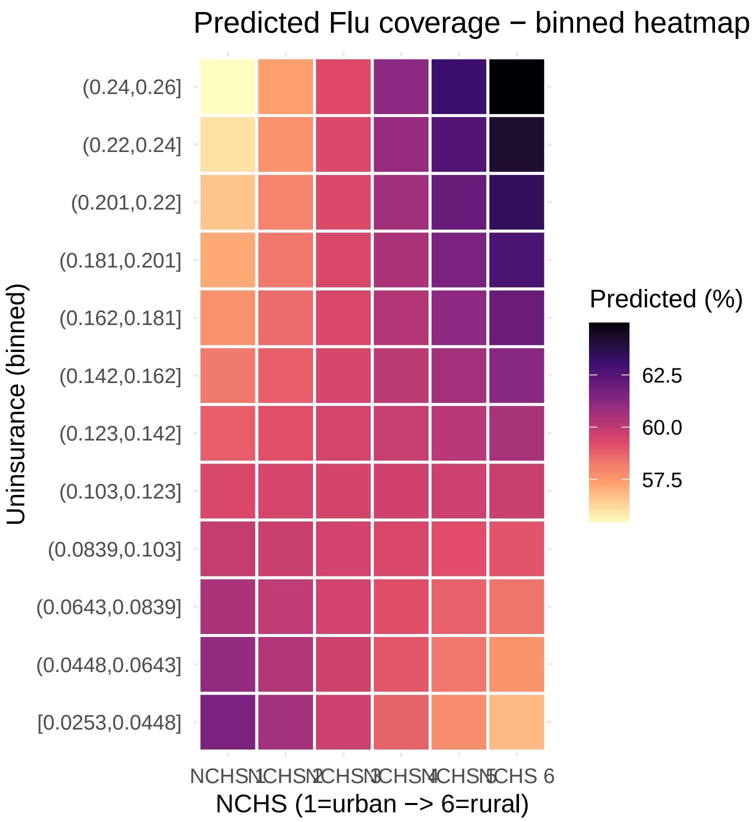
```r
ggplot(grid_tdap, aes(uninsured_rate, pred)) +
  geom_line() +
  facet_wrap(~ nchs_2013, nrow = 2, labeller = label_both) +
  labs(x="Uninsurance rate", y="Predicted Tdap coverage (%)",
       title="TWFE predicted lines by NCHS (Tdap)") +
  theme_minimal()
```



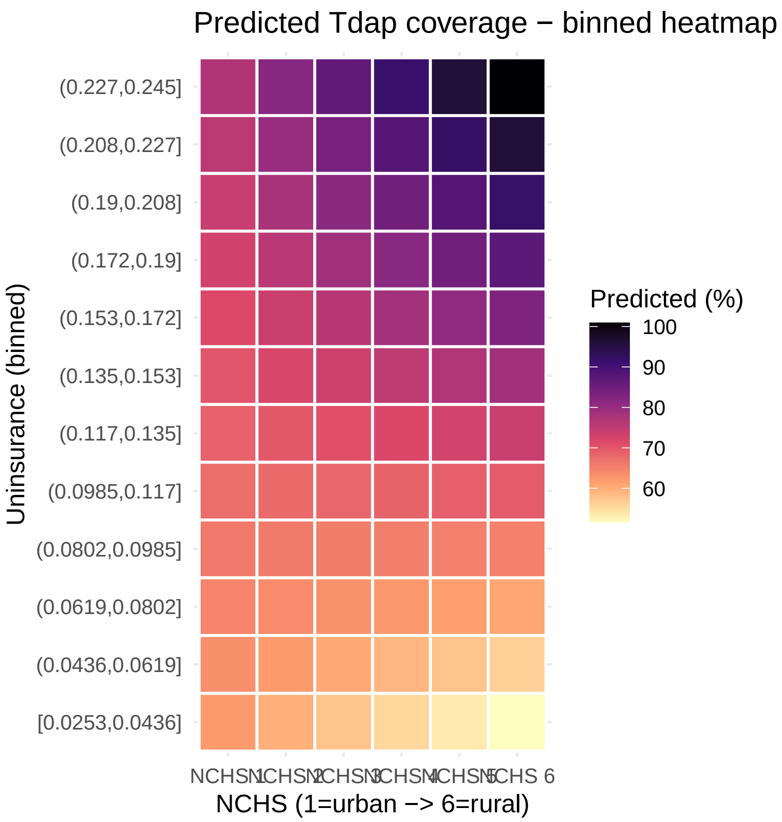
0.10 6C) Forecast heat map (Uninsurance × NCHS → Forecast coverage)

Suitable for "global sense" display interaction: horizontal axis uninsured rate, vertical axis NCHS, color = predicted coverage.

```r
# Predict grid function

mk_heat <- function(dat, mod){
  g <- expand.grid(
    nchs_2013 = 1:6,
    uninsured_rate = seq(min(dat$uninsured_rate, na.rm=TRUE),
                         max(dat$uninsured_rate, na.rm=TRUE), length.out = 80),
    year  = median(dat$year, na.rm=TRUE),
    state = dat$state[1]
  )
  g <- dplyr::mutate(
    g,
    nchs13_z = (nchs_2013 - mean(dat$nchs_2013, na.rm=TRUE))/sd(dat$nchs_2013, na.rm=TRUE),
    unins_z  = (uninsured_rate - mean(dat$uninsured_rate,
                                      na.rm=TRUE))/sd(dat$uninsured_rate, na.rm=TRUE)
  )
  g$pred <- predict(mod, newdata = g)
  g
}

heat_flu  <- mk_heat(dat_flu,  m23_flu)
heat_tdap <- mk_heat(dat_tdap, m23_tdap)

# Discretize the continuous uninsurance into "grid strips" and take
# the average predicted value for each grid
# we can increase or decrease nbins to control the number of grids
nbins <- 12

bin_and_avg <- function(g){
  rng <- range(g$uninsured_rate, na.rm = TRUE)
  breaks <- seq(rng[1], rng[2], length.out = nbins + 1)
  g %>%
    dplyr::mutate(
      nchs_bin  = factor(nchs_2013, levels = 1:6, labels = paste0("NCHS ", 1:6)),
      unins_bin = cut(uninsured_rate, breaks = breaks, include.lowest = TRUE)
    ) %>%
    dplyr::group_by(nchs_bin, unins_bin) %>%
    dplyr::summarise(pred = mean(pred, na.rm = TRUE), .groups = "drop")
}

heat_flu_b  <- bin_and_avg(heat_flu)
heat_tdap_b <- bin_and_avg(heat_tdap)

# heatmap

ggplot(heat_flu_b, aes(x = nchs_bin, y = unins_bin, fill = pred)) +
  geom_tile(color = "white", size = 0.6) +  # Add white grid lines
  scale_fill_gradient(low = "#f7fbff", high = "#08306b",
                      name = "Predicted coverage (%)") +
  labs(x="NCHS (1=urban → 6=rural)", y="Uninsurance (binned)",
       title="Predicted Flu coverage - binned heatmap") +
  coord_equal() +
  scale_fill_viridis_c(option = "magma", direction = -1, name = "Predicted (%)") +
  theme_minimal()
```



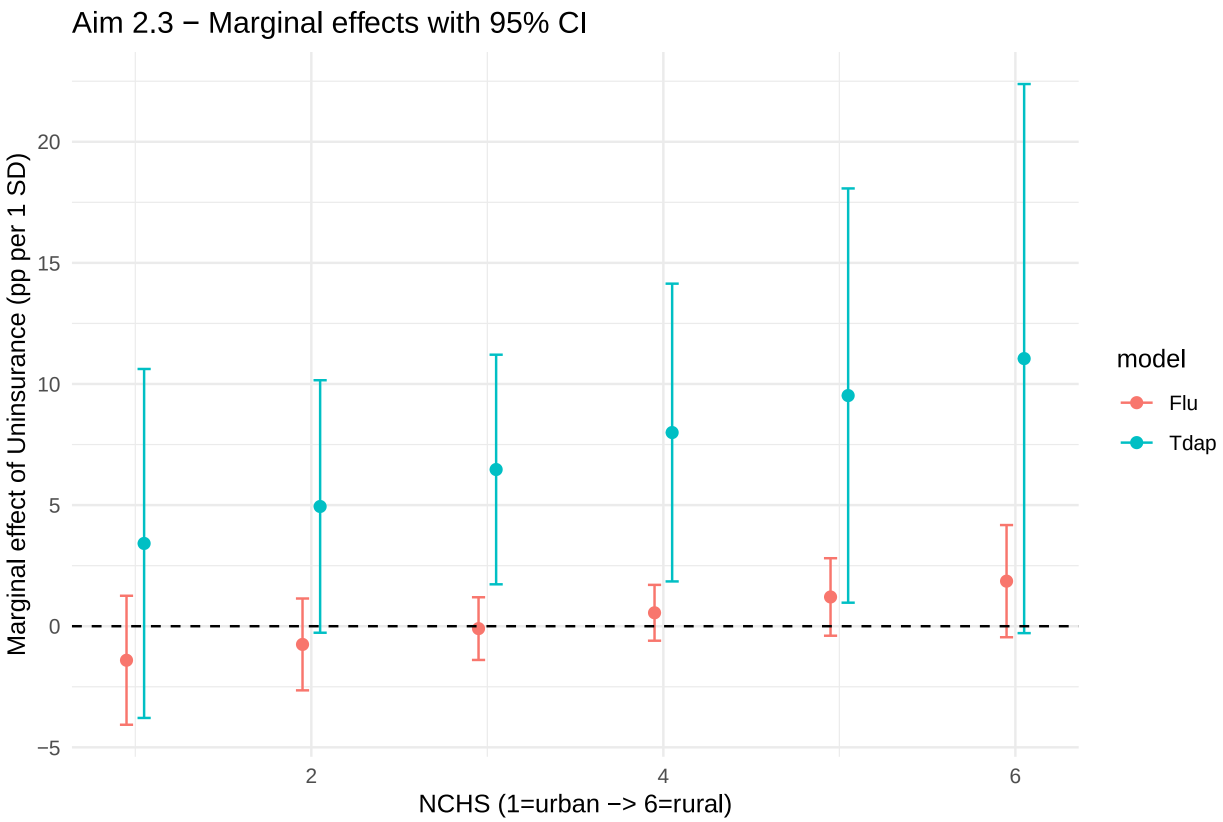
```r
ggplot(heat_tdap_b, aes(x = nchs_bin, y = unins_bin, fill = pred)) +
  geom_tile(color = "white", size = 0.6) +
  scale_fill_gradient(low = "#f7fbff", high = "#08306b",
                      name = "Predicted coverage (%)") +
  labs(x="NCHS (1=urban → 6=rural)", y="Uninsurance (binned)",
       title="Predicted Tdap coverage - binned heatmap") +
  coord_equal() +
  scale_fill_viridis_c(option = "magma", direction = -1, name = "Predicted (%)") +
  theme_minimal()
```



0.11 6D) Error bar graph of interactive "marginal effects"

Also based on 6-step me_all, only changed to point + error bar style

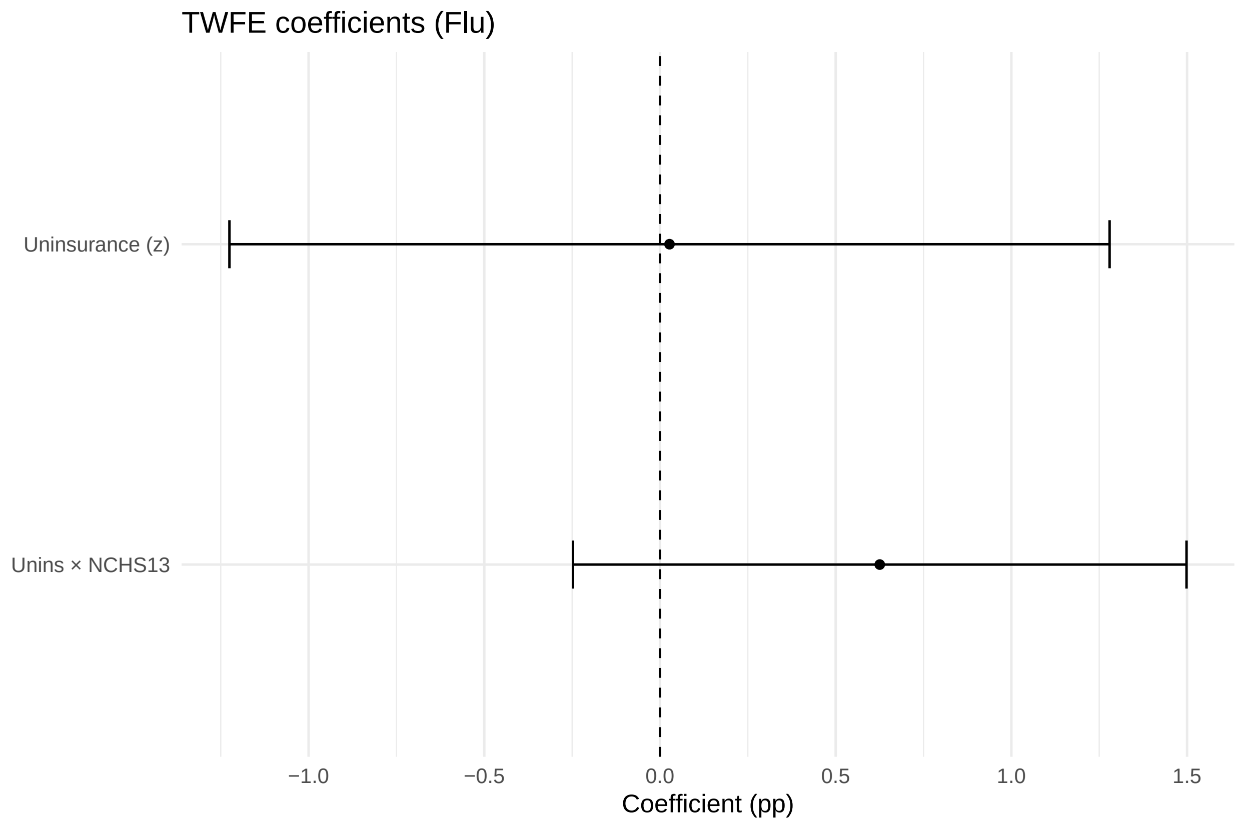
```r
ggplot(me_all, aes(x = NCHS, y = marginal_effect, color = model)) +
  geom_point(position = position_dodge(width=.2), size = 2) +
  geom_errorbar(aes(ymin = lo95, ymax = hi95), width = .15,
                position = position_dodge(width=.2)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "NCHS (1=urban → 6=rural)",
       y = "Marginal effect of Uninsurance (pp per 1 SD)",
       title = "Aim 2.3 - Marginal effects with 95% CI") +
  theme_minimal()
```



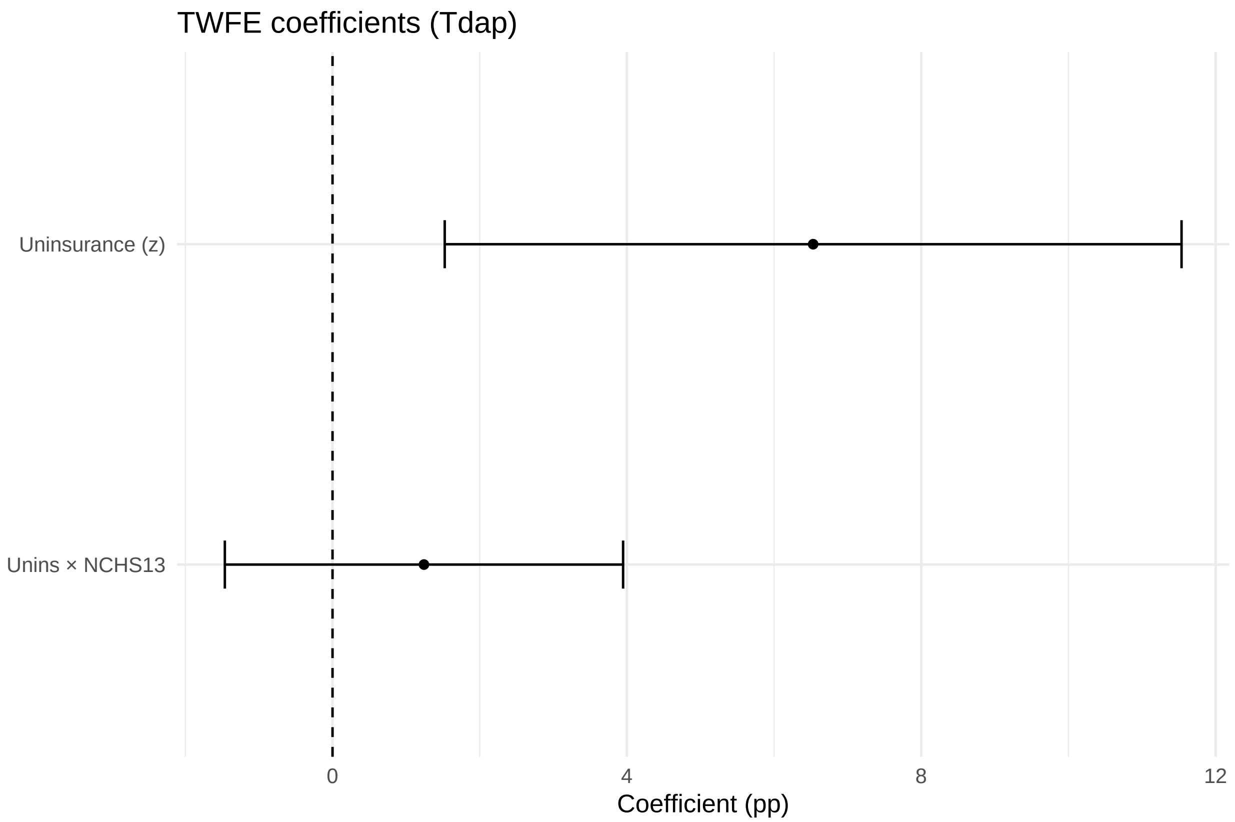
0.12 6E) Coefficient plot

Draw the key coefficients (uninsz and uninsz:nchs13_z) as error bars to visually see the significance and direction.

```
coef_plot <- function(mod, title){
  broom::tidy(mod, conf.int = TRUE) |>
    filter(term %in% c("unins_z","unins_z:nchs13_z")) |>
    mutate(term = recode(term,
                         "unins_z" = "Uninsurance (z)",
                         "unins_z:nchs13_z" = "Unins × NCHS13")) |>
    ggplot(aes(estimate, term)) +
    geom_point() +
    geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = .15) +
    geom_vline(xintercept = 0, linetype = "dashed") +
    labs(x = "Coefficient (pp)", y = NULL, title = title) +
    theme_minimal()
}
coef_plot(m23_flu,  "TWFE coefficients (Flu)")
```



```
coef_plot(m23_tdap, "TWFE coefficients (Tdap)")
```



0.13 7) Robustness

```r
# Replace NCHS with 2023 index in interaction
m23_flu_2023 <- feols(vacc_flu_pct  ~ unins_z * scale(nchs_2023)[,1] | state + year,
                      data = dat_flu, weights = ~ n_flu, cluster = ~ state)
m23_tdap_2023 <- feols(vacc_tdap_pct ~ unins_z * scale(nchs_2023)[,1] | state + year,
                       data = dat_tdap, weights = ~ n_tdap, cluster = ~ state)
fixest::etable(m23_flu_2023, m23_tdap_2023, se="cluster", title="Replace NCHS with 2023 index")
```

```
##                                   m23_flu_2023  m23_tdap_2023
## Dependent Var.:                   vacc_flu_pct  vacc_tdap_pct
##
## unins_z                        0.0775 (0.6222) 5.868* (2.187)
## unins_z x scale(nchs_2023)[,1] 0.4636 (0.4114) 0.1392 (1.262)
## Fixed-Effects:                 --------------- --------------
## state                                      Yes            Yes
## year                                       Yes            Yes
## ______________________________ _______________ ______________
## S.E.: Clustered                      by: state      by: state
## Observations                               321            151
## R2                                     0.93249        0.97501
## Within R2                              0.01283        0.07757
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Unweighted versions
m23_flu_unw  <- feols(vacc_flu_pct  ~ unins_z * nchs13_z |
                        state + year, data = dat_flu,  cluster = ~ state)
m23_tdap_unw <- feols(vacc_tdap_pct ~ unins_z * nchs13_z |
                        state + year, data = dat_tdap, cluster = ~ state)
fixest::etable(m23_flu_unw, m23_tdap_unw, se="cluster", title="Unweighted models")
```

```
##                         m23_flu_unw   m23_tdap_unw
## Dependent Var.:        vacc_flu_pct  vacc_tdap_pct
##
## unins_z            -0.0496 (0.6224) 5.719* (2.538)
## unins_z x nchs13_z  0.3875 (0.3731) 0.9742 (1.472)
## Fixed-Effects:     ---------------- --------------
## state                           Yes            Yes
## year                            Yes            Yes
## __________________ ________________ ______________
## S.E.: Clustered            by: state      by: state
## Observations                    321            151
## R2                          0.93006        0.97632
## Within R2                   0.00715        0.07441
## ---
## Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```