Aim 1.2 Code

Ruining Zhou

2025-10-23

1. Data management for merged version dataset

```
vac <- read.csv("~/Desktop/QBS 181/Wrangle Avengers/MergedVersion.csv")
colnames(vac) <- c("Vaccine", "Geography_Type", "State_full","Survey_Year", "Adjustment", "Value", "Esti

vac <- vac %>%
  mutate(State = toupper(State)) %>%
  mutate(
    across(c(Vaccine, Geography_Type, State_full, Survey_Year, Year, Adjustment, Value, State), as.facto
    across(CI, as.character),
    across(c(Estimate, Sample_Size, Insured, Uninsured, UrbanicityIndex),
           ~ as.numeric(gsub("[^0-9.]", "", .)))
  )

# check missing pattern
dim(vac)
```

```
## [1] 4568   14
```

```
colSums(is.na(vac))
```

```
##         Vaccine  Geography_Type      State_full     Survey_Year      Adjustment
##               0               0               0               0               0
##           Value        Estimate              CI     Sample_Size           State
##               0             457               0             213               0
##            Year         Insured      Uninsured UrbanicityIndex
##               0             634             492              72
```

```
summary(vac)
```

```
##       Vaccine     Geography_Type        State_full    Survey_Year
##  Influenza:3078   States:4568    New York    : 440   2018   : 524
##  Tdap     :1490                  Delaware    : 176   2019   : 524
##                                  Missouri    : 176   2020   : 492
##                                  Pennsylvania: 176   2021   : 468
##                                  Wisconsin   : 176   2017   : 436
##                                  Colorado    : 160   2016   : 428
##                                  (Other)     :3264   (Other):1696
##               Adjustment                                   Value
```

```
##  Age                :2304   >=18 Years                            : 576
##  Race and Ethnicity:2264   >=35 Years                            : 576
##                            18-24 Years                           : 576
##                            25-34 Years                           : 576
##                            Other or Multiple Races, Non-Hispanic: 575
##                            White, Non-Hispanic                   : 575
##                            (Other)                               :1114
##     Estimate          CI                Sample_Size          State
##  Min.   : 5.2   Length:4568        Min.   :  30.0   NY     : 440
##  1st Qu.:54.3   Class :character   1st Qu.: 162.0   DE     : 176
##  Median :64.1   Mode  :character   Median : 303.0   MO     : 176
##  Mean   :62.9                      Mean   : 423.0   PA     : 176
##  3rd Qu.:73.2                      3rd Qu.: 600.5   WI     : 176
##  Max.   :99.6                      Max.   :2370.0   CO     : 160
##  NA's   :457                       NA's   :213      (Other):3264
##       Year          Insured           Uninsured       UrbanicityIndex
##  2018   : 524   Min.   : 131800   Min.   :   4800   Min.   :1.000
##  2019   : 524   1st Qu.: 505500   1st Qu.:  48800   1st Qu.:2.435
##  2020   : 492   Median :1503700   Median : 141900   Median :2.876
##  2021   : 468   Mean   :1849184   Mean   : 192760   Mean   :3.071
##  2017   : 436   3rd Qu.:2401175   3rd Qu.: 257300   3rd Qu.:3.593
##  2016   : 428   Max.   :6491000   Max.   :1821300   Max.   :4.911
##  (Other):1696   NA's   :634       NA's   :492       NA's   :72
```

```
# check age group, missing estimates, and sample size
age_missing_keys <- vac %>%
  filter(Adjustment == "Age" & (is.na(Estimate) | is.na(Sample_Size))) %>%
  distinct(State_full, Vaccine, Year)
# check those same combinations for Race and Ethnicity
race_check <- vac %>%
  filter(Adjustment == "Race and Ethnicity")%>%
  semi_join(age_missing_keys, by = c("State_full", "Vaccine", "Year")) %>%
  group_by(State_full, Vaccine, Year) %>%
  summarise(
    total = n(),
    missing_estimate = sum(is.na(Estimate)),
    missing_sample_size = sum(is.na(Sample_Size)),
    all_complete = all(!is.na(Estimate) & !is.na(Sample_Size))
  ) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'State_full', 'Vaccine'. You can override
## using the `.groups` argument.
```

```
race_check[race_check$all_complete==TRUE,]
```

```
## # A tibble: 0 x 7
## # i 7 variables: State_full <fct>, Vaccine <fct>, Year <fct>, total <int>,
## #   missing_estimate <int>, missing_sample_size <int>, all_complete <lgl>
```

```
# use age group to recalcute the estimates for year and
vac_filtered <- vac %>%
```

```
  filter(Adjustment == "Age") %>%
  anti_join(age_missing_keys, by = c("State_full", "Vaccine", "Year"))

# check missingness
vac_filtered %>%
  filter(Adjustment == "Age" & (is.na(Estimate) | is.na(Sample_Size))) %>%
  distinct(State_full, Vaccine, Year)
```

```
## [1] State_full Vaccine    Year
## <0 rows> (or 0-length row.names)
```

```
# check consistency for insured, uninsured, and urbanicity
vac_filtered %>%
  group_by(State_full, Vaccine, Year) %>%
  summarise(
    n_rows = n(),
    insured_var = n_distinct(Insured, na.rm = TRUE),
    uninsured_var = n_distinct(Uninsured, na.rm = TRUE),
    urbanicity_var = n_distinct(UrbanicityIndex, na.rm = TRUE)
  ) %>%
  ungroup() %>%
  summarise(
    groups_total = n(),
    groups_insured_vary = sum(insured_var > 1),
    groups_uninsured_vary = sum(uninsured_var > 1),
    groups_urbanicity_vary = sum(urbanicity_var > 1)
  )
```

```
## `summarise()` has grouped output by 'State_full', 'Vaccine'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 1 x 4
##   groups_total groups_insured_vary groups_uninsured_vary groups_urbanicity_vary
##          <int>               <int>                 <int>                   <int>
## 1          523                   0                     0                       0
```

```
df <- vac_filtered %>%
  group_by(State_full, Vaccine, Year) %>%
  summarise(
    Weighted_Estimate = sum(Estimate * Sample_Size) /
                        (100*sum(Sample_Size)),
    Total_Sample = sum(Sample_Size),
    State = first(State),
    Insured = first(Insured),
    Uninsured = first(Uninsured),
    UrbanicityIndex = first(UrbanicityIndex)
  ) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'State_full', 'Vaccine'. You can override
## using the `.groups` argument.
```

```r
# exclude all missingness
colSums(is.na(df))
```

```
##        State_full           Vaccine              Year Weighted_Estimate
##                 0                 0                 0                 0
##      Total_Sample             State           Insured         Uninsured
##                 0                 0                81                60
##   UrbanicityIndex
##                 9
```

```r
df <- df %>% filter(complete.cases(.))
dim(df)
```

```
## [1] 434   9
```

```r
df <- df %>%
  mutate(State = toupper(State)) %>%
  mutate(insurance_coverage = Insured/(Insured+Uninsured))

head(df)
```

```
## # A tibble: 6 x 10
##   State_full Vaccine   Year  Weighted_Estimate Total_Sample State Insured
##   <fct>      <fct>     <fct>             <dbl>        <dbl> <chr>   <dbl>
## 1 Alabama    Influenza 2014              0.424         1646 AL    1235300
## 2 Alabama    Influenza 2015              0.476         1608 AL    1265800
## 3 Alabama    Influenza 2017              0.473         1684 AL    1269000
## 4 Alabama    Influenza 2018              0.468         1560 AL    1257900
## 5 Alabama    Influenza 2019              0.530         1536 AL    1270200
## 6 Alabama    Influenza 2021              0.459         1334 AL    1287800
## # i 3 more variables: Uninsured <dbl>, UrbanicityIndex <dbl>,
## #   insurance_coverage <dbl>
```

2. Prepare mapping data

```r
# get U.S. state boundaries (excluding territories)
states_sf <- states(cb = TRUE, year = 2023) %>%
  st_as_sf() %>%
  filter(!STUSPS %in% c("AS", "GU", "MP", "PR", "VI")) %>%
  select(STUSPS, NAME, geometry)

# join vaccination + insurance data
merged_df <- states_sf %>%
  left_join(df, by = c("STUSPS" = "State"))

# check unmatched states
unmatched <- merged_df %>% filter(is.na(Vaccine))
if (nrow(unmatched) > 0) {
  cat("Warning: Some states did not match:\n")
  print(unmatched$NAME)
}
```

```
## Warning: Some states did not match:
## [1] "California"     "Idaho"          "Nevada"         "South Carolina"
## [5] "Connecticut"
```

```
# save as GeoJSON for future use
st_write(merged_df, "pregnant_vaccination.geojson", driver = "GeoJSON", delete_dsn = TRUE)
```

```
## Deleting source 'pregnant_vaccination.geojson' using driver 'GeoJSON'
## Writing layer 'pregnant_vaccination' to data source
##   'pregnant_vaccination.geojson' using driver 'GeoJSON'
## Writing 439 features with 11 fields and geometry type Multi Polygon.
```

Pevious steps are Aim 1 data cleaning.

Aim 1.2: Compare temporal and sociodemographic patterns of maternal influenza and Tdap vaccination coverage from 2013 to 2022 to assess changes in coverage inequities across states and population subgroups, and evaluate whether disparities in vaccine uptake have narrowed over time.

```
# trends of maternal vaccination coverage (2013-2022)
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:viridis':
##
##     viridis_pal

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

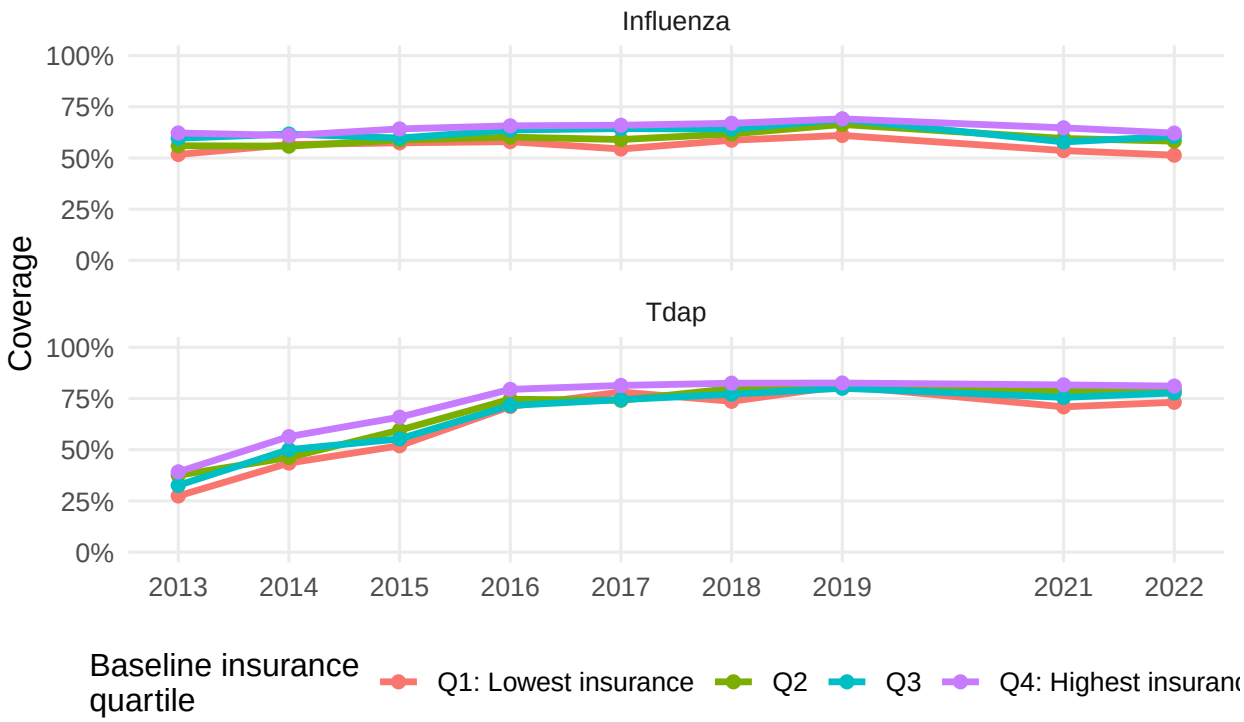
```
library(glue)
library(tibble)

# make sure year is numeric
vac <- vac %>%
  mutate(
    Year = as.integer(as.character(Year)),
    Vaccine = as.factor(Vaccine)
  )

# weighted estimates per state vaccine year
build_level <- function(v, adj_name) {
  v %>%
```

```
    filter(Adjustment == adj_name) %>%
    group_by(State, Vaccine, Year) %>%
    summarise(Weighted_Estimate = sum(Estimate * Sample_Size, na.rm = TRUE) /(100 * sum(Sample_Size, na
      Total_Sample = sum(Sample_Size, na.rm = TRUE),
      .groups = "drop"
    ) %>%
    mutate(source = adj_name)
}

age_tbl <- build_level(vac, "Age")
race_tbl <- build_level(vac, "Race and Ethnicity")

core_tbl <- bind_rows(age_tbl, race_tbl) %>%
  mutate(priority = match(source, c("Age", "Race and Ethnicity"))) %>%
  arrange(State, Vaccine, Year, priority) %>%
  group_by(State, Vaccine, Year) %>%
  slice(1) %>%
  ungroup()

# insurance info for each state
insurance_tbl <- vac %>%
  group_by(State, Year) %>%
  summarise(Insured = first(na.omit(Insured)),
    Uninsured = first(na.omit(Uninsured)),
    .groups = "drop"
  )

# merge insurance coverage
trend_dat <- core_tbl %>%
  left_join(insurance_tbl, by = c("State", "Year")) %>%
  filter(
    !is.na(Insured),
    !is.na(Uninsured),
    is.finite(Weighted_Estimate)
  ) %>%
  mutate(
    insurance_coverage = Insured / (Insured + Uninsured),
    Vaccine = as.factor(Vaccine),
    Year = as.integer(Year)
  )

stopifnot(nrow(trend_dat) > 0)

# define baseline 2013 insurance
trend_13_22<- trend_dat %>%
  filter(Year >= 2013, Year <= 2022)

baseline_2013<- trend_13_22 %>%
  filter(Year == 2013) %>%
  transmute(State, base_ins_cover = insurance_coverage) %>%
  filter(is.finite(base_ins_cover))

# quartiles by baseline insurance
```

```r
quartile_tbl <- baseline_2013 %>%
  mutate(
    q = ntile(base_ins_cover, 4),
    qlab = factor(q,
      levels = 1:4,
      labels = c("Q1: Lowest insurance", "Q2", "Q3", "Q4: Highest insurance")
    )
  )

trend_q <- trend_13_22 %>%
  inner_join(quartile_tbl, by = "State")
```

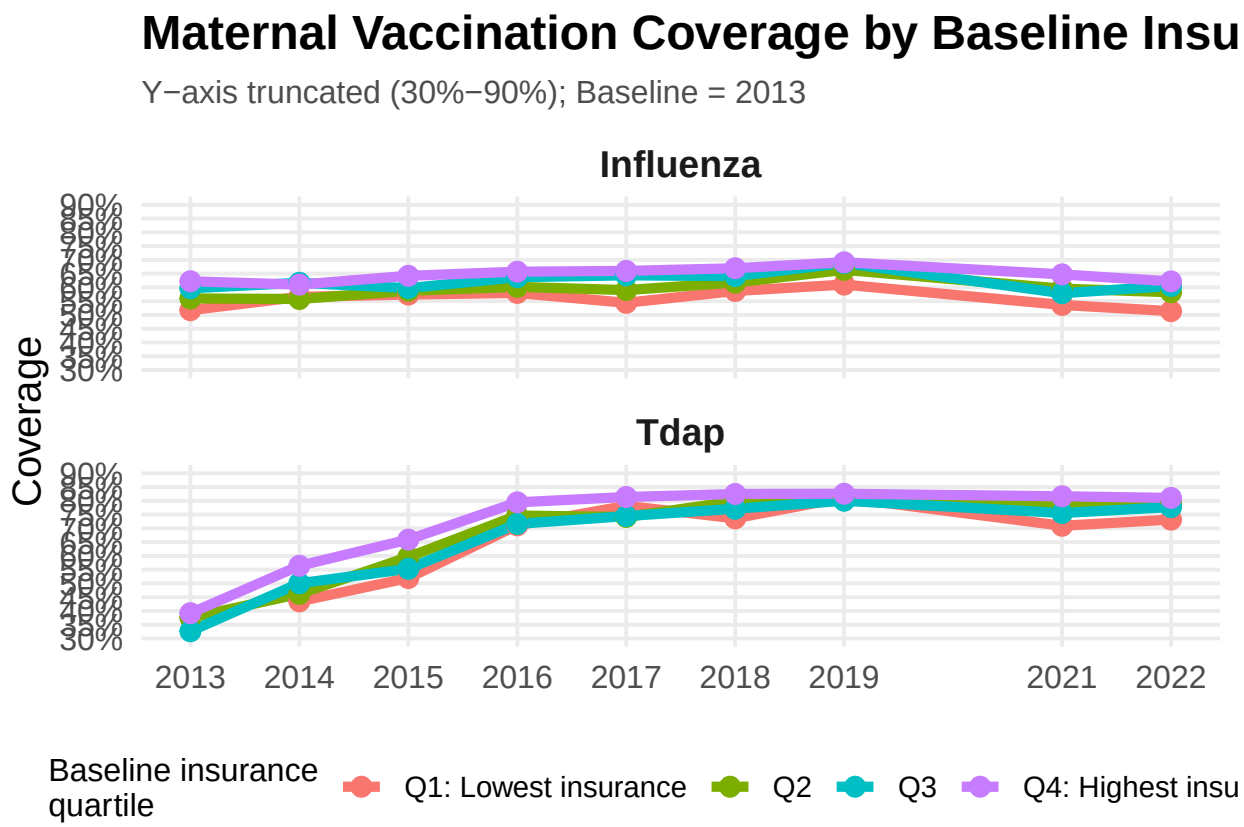
```
## Warning in inner_join(., quartile_tbl, by = "State"): Detected an unexpected many-to-many relationshi
## i Row 22 of `x` matches multiple rows in `y`.
## i Row 1 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
# summaries across state
trend_summary <- trend_q %>%
  group_by(Vaccine, Year, qlab) %>%
  summarise(
    mean_cov = mean(Weighted_Estimate, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(Year = as.integer(as.character(Year)),
    Vaccine = as.factor(Vaccine),
    qlab = droplevels(qlab)
  ) %>%
  filter(is.finite(Year))

year_breaks <- sort(unique(trend_summary$Year))

# plot
p_trend <- ggplot(trend_summary, aes(x = Year, y = mean_cov, color = qlab, group = qlab)) +
  geom_line(linewidth = 1.2, na.rm = TRUE) +
  geom_point(size = 2, na.rm = TRUE) +
  scale_y_continuous(labels = label_percent(accuracy = 1), limits = c(0, 1)) +
  scale_x_continuous(breaks = year_breaks) +
  labs( title = "Maternal Vaccination Coverage by Baseline Insurance Quartiles",
    subtitle = "Baseline = 2013; lines show state-mean coverage",
    x = NULL,
    y = "Coverage",
    color ="Baseline insurance\nquartile"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.minor = element_blank(),
    legend.position = "bottom"
  ) +
  facet_wrap(vars(Vaccine), ncol = 1, drop = TRUE)

print(p_trend)
```




```r
# pretty version more clear
y_min <- 0.30
y_max <- 0.90

p_trend_trunc <- ggplot(trend_summary,aes(x = Year, y = mean_cov, color = qlab, group = qlab)) +
  geom_line(linewidth = 1.8, na.rm = TRUE) +
  geom_point(size = 3.2, na.rm = TRUE) +
  scale_y_continuous(
    labels = label_percent(accuracy = 1),
    limits = c(y_min, y_max),
    breaks = seq(y_min, y_max, by = 0.05)
  ) +
  scale_x_continuous(breaks = year_breaks) +
  labs(title = "Maternal Vaccination Coverage by Baseline Insurance Quartiles",
    subtitle = glue("Y-axis truncated ({percent(y_min)}-{percent(y_max)}); Baseline = 2013"),
    x = NULL,
    y = "Coverage",
    color = "Baseline insurance\nquartile"
  ) +
  theme_minimal(base_size = 15) +
  theme(
    panel.grid.minor  = element_blank(),
    legend.position   = "bottom",
    legend.title      = element_text(size = 12),
    legend.text       = element_text(size = 11),
    plot.title        = element_text(face = "bold", size = 18),
```

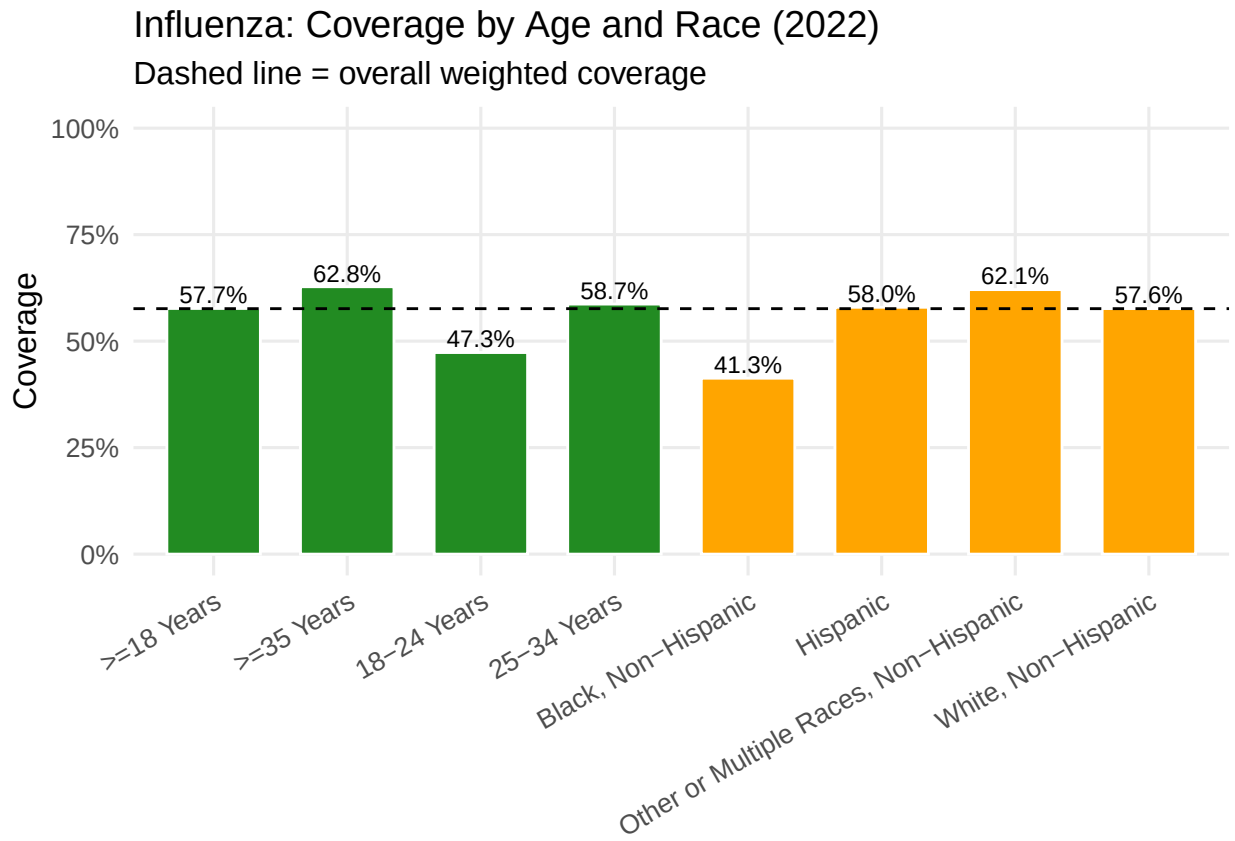
```
    plot.subtitle     = element_text(size = 12, colour = "grey30"),
    strip.text        = element_text(face = "bold", size = 14)
  ) +
  facet_wrap(vars(Vaccine), ncol = 1, drop = TRUE)

print(p_trend_trunc)
```



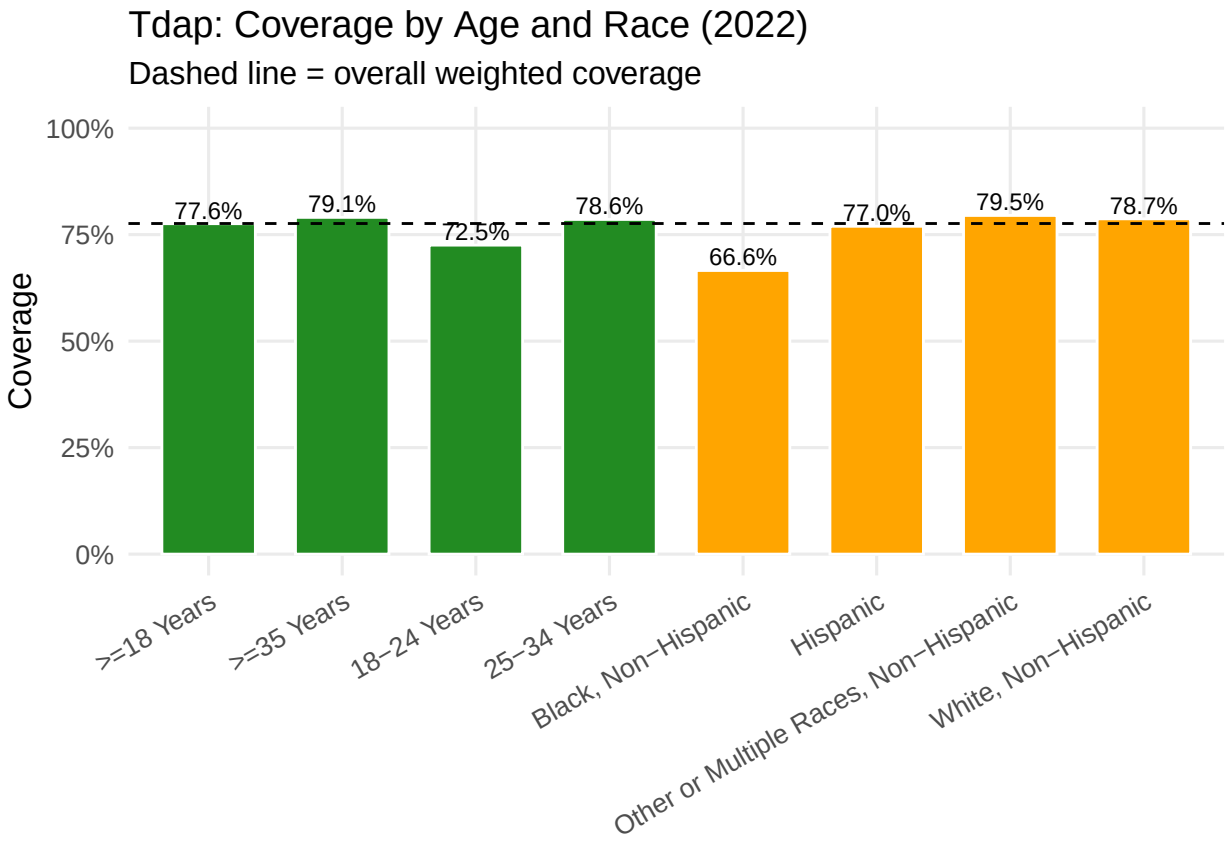

```
# bar charts for age and race (2022)
library(dplyr)
library(ggplot2)
library(scales)
library(glue)
library(tibble)

# build age and race dataset for each vaccine 2022
age_race_group <- function(vac, year_pick = 2022, vaccine_pick = "Influenza") {
  dict_adj <- tibble::tibble(
    Adjustment = c("Age","Race and Ethnicity"),
    group_set  = c("By Age","By Race/Ethnicity")
  )

  vac_clean <- vac %>% filter(Year == year_pick, Vaccine == vaccine_pick)

  overall_try <- function(adj) {
    vac_clean %>%
```

```
      filter(Adjustment == adj) %>%
      summarise( ov = sum(Estimate * Sample_Size, na.rm = TRUE) /
             (100 * sum(Sample_Size, na.rm = TRUE))
      ) %>% pull(ov)
  }

  overall <- if (!all(is.na(vac_clean$Estimate))) {
    if (any(vac_clean$Adjustment == "Age")) {
      overall_try("Age")
    } else {
      overall_try("Race and Ethnicity")
    }
  } else NA_real_

  # compute a national weighted mean across states (weights = state total n)
  group_list <- list()
  for (adj in dict_adj$Adjustment) {
    if (!adj %in% unique(vac_clean$Adjustment)) next

    tmp <- vac_clean %>%
      filter(Adjustment == adj) %>%
      group_by(Value, State) %>%
      summarise(w_est = sum(Estimate * Sample_Size, na.rm = TRUE) /
                  (100 * sum(Sample_Size, na.rm = TRUE)),
        n_sum = sum(Sample_Size, na.rm = TRUE),
        .groups = "drop"
      ) %>%
      group_by(Value) %>%
      summarise(
        cov = weighted.mean(w_est, w = n_sum, na.rm = TRUE),
        N   = sum(n_sum, na.rm = TRUE),
        .groups = "drop"
      ) %>%
      mutate(Adjustment = adj)
    group_list[[adj]] <- tmp
  }

  # check data
  grp_all <- bind_rows(group_list)
  if (nrow(grp_all) == 0) stop("No subgroup data for this year/vaccine.")
  grp_all <- grp_all %>%
    left_join(dict_adj, by = "Adjustment") %>%
    mutate(
      group_set = factor(group_set, levels = c("By Age","By Race/Ethnicity")),
      label = Value
    ) %>%
    arrange(group_set, label)
  list(overall = overall, tbl = grp_all)
}

# plot
plot_group_bar <- function(obj, year_pick, vaccine_pick) {
  overall_val <- obj$overall
```

```r
  dat <- obj$tbl
  ggplot(dat, aes(x = reorder(label, as.numeric(group_set)), y = cov)) +
    geom_col(aes(fill = group_set), width = 0.7, color = "white") +
    {if (is.finite(overall_val))
        geom_hline(yintercept = overall_val, linetype = 2, color = "black") } +
    geom_text(aes(label = percent(cov, accuracy = 0.1)),
              vjust = -0.3, size = 3.2, color = "black") +
    scale_y_continuous(labels = percent_format(accuracy = 1), limits = c(0, 1)) +
    scale_fill_manual(values = c("By Age" = "forestgreen",
                                 "By Race/Ethnicity" = "orange"),
                      guide = "none") +
    labs(title = glue("{vaccine_pick}: Coverage by Age and Race ({year_pick})"),
      subtitle = "Dashed line = overall weighted coverage",
      x = NULL, y = "Coverage"
    ) +
    theme_minimal(base_size = 12) +
    theme(
      panel.grid.minor = element_blank(),
      axis.text.x = element_text(angle = 30, hjust = 1),
      strip.text = element_text(face = "bold")
    )
}

#Influenza (2022)
gA <- age_race_group(vac, year_pick = 2022, vaccine_pick = "Influenza")
p_gA <- plot_group_bar(gA, 2022, "Influenza")
print(p_gA)
```




```r
# Tdap (2022)
gB <- age_race_group(vac, year_pick = 2022, vaccine_pick = "Tdap")
p_gB <- plot_group_bar(gB, 2022, "Tdap")
print(p_gB)
```


Tdap: Coverage by Age and Race (2022)
Dashed line = overall weighted coverage
Coverage
100%
75%
50%
25%
0%
77.6%
79.1%
72.5%
78.6%
66.6%
77.0%
79.5%
78.7%
>=18 Years
>=35 Years
18–24 Years
25–34 Years
Black, Non–Hispanic
Hispanic
Other or Multiple Races, Non–Hispanic
White, Non–Hispanic