Figure 2

Brittany A. Petros

2024-02-26

Set up the workspace.

```r
rm(list=ls(all.names=TRUE))
setwd("/Users/bpetros/Desktop/PHIS")
libs <- c("cowplot", "ggsignif", "tidyverse")
invisible(lapply(libs, function(x) suppressPackageStartupMessages(library(x, character.only = TRUE))))
options(stringsAsFactors=FALSE, scipen = 999)
theme_set(theme_classic())
```

Read in the cleaned input files.

```r
# read in cleaned pt data
pt <- read.csv("cleaned/rsv_patient.csv")
pt$Date = ymd(pt$Discharge_Date)

# read in cleaned tested pt data
ptest <- read.csv("cleaned/tested_patient.csv")
ptest$Date = ymd(ptest$Discharge_Date)
```

This function takes a df with column "Admit_Age_In_Days" (integer) as input. It creates a column called Age_Group that labels rows as belonging to one of the following age groups: 0-3 months, 3-6 months, 6-12 months, 1-2 years, 2-4 years, and >= 5 years.

```r
add_age_group <- function(df) {
  df$Age_Group <- cut(
    df$Admit_Age_In_Days,
    breaks = c(0, 90, 180, 365, 730, 1460, Inf),
    labels = c("0-3 months", "3-6 months", "6-12 months", "1-2 years", "2-4 years", "5-17 years"),
    right = FALSE)
  return(df)}
```

This function takes a df with column "column_name" as input. The column must represent a binary variable using either 0s and 1s (integer) or "N" and "Y" (character). The function prints the percent of rows for which the column equals either 1 or "Y".

```r
summarize_and_print_column <- function(df, column_name) {
  df_summary <- df %>%
    summarize(
      Percent = round(sum(.data[[column_name]] == "Y" | .data[[column_name]] == 1) / n() * 100, 2))
  cat(column_name, "- Percent:", df_summary$Percent, "%\n")}
```

This function takes a df generated from a PHIS database download as input (i.e., multiple specific column names are required). It calculates and prints demographic (e.g., age, race, ethnicity) and clinical (e.g., presence of comorbid conditions, outcomes) summary statistics.

```
analyze_dataframe <- function(df) {

  # counts and percent of patients in each category (ED, ED -> IP, IP)
  ed_discharged_home <- sum(df$Patient_Type_Title == "ED Visit")
  percent_ed_discharged_home <- (ed_discharged_home / nrow(df)) * 100
  ed_admitted <- sum(df$Patient_Type_Title != "ED Visit" & df$ED_entry == 1)
  percent_ed_admitted <- (ed_admitted / nrow(df)) * 100
  admitted_without_ed <- sum(df$Patient_Type_Title != "ED Visit" & df$ED_entry == 0)
  percent_admitted_without_ed <- (admitted_without_ed / nrow(df)) * 100

  # patient type df
  ptype_counts <- list(
    ED_Discharged_Home_Count = ed_discharged_home,
    ED_Admitted_Count = ed_admitted,
    Admitted_Without_ED_Count = admitted_without_ed)

  # print results
  cat("ED Discharged Home:\n")
  cat("Count:", ed_discharged_home, "\n")
  cat("Percentage:", percent_ed_discharged_home, "%\n\n")
  cat("ED Admitted:\n")
  cat("Count:", ed_admitted, "\n")
  cat("Percentage:", percent_ed_admitted, "%\n\n")
  cat("Admitted Without ED:\n")
  cat("Count:", admitted_without_ed, "\n")
  cat("Percentage:", percent_admitted_without_ed, "%\n\n")

  # median (IQR) age, in months
  df_summary <- df %>%
    summarize(
      Median_Age_Months = round(median(Admit_Age_In_Days) / 30.44, 2),
      Q1 = round(quantile(Admit_Age_In_Days, 0.25) / 30.44, 2),
      Q3 = round(quantile(Admit_Age_In_Days, 0.75) / 30.44, 2))

  # print age statistics
  cat("Admit_Age_In_Mo - Median:", df_summary$Median_Age_Months, "months\n")
  cat("Admit_Age_In_Mo - IQR (Q1 - Q3):", df_summary$Q1, "months -", df_summary$Q3, "months\n")

  # report percent in each age category
  age_groups = table(df$Age_Group)
  age_percents <- round(table(df$Age_Group) / nrow(df) * 100, 2)
  cat("Admit_Age_In_Days - Percent:\n")
  print(age_percents)

  # report percent male
  df_gender_summary <- df %>%
    summarize(Percent_Male = round(sum(Gender_Title == "Male") / n() * 100, 2))
  cat("Gender_Title - Percent Male:", df_gender_summary$Percent_Male, "%\n")

  # report percent in each ethnicity category
```

```r
Ethnicity_Summary <- df %>%
  summarize(
    Percent_Hispanic = round(sum(Ethnicity_Title == "Hispanic or Latino") / n() * 100, 2),
    Percent_Not_Hispanic = round(sum(Ethnicity_Title == "Not Hispanic or Latino") / n() * 100, 2),
    Percent_Unknown = round(sum(Ethnicity_Title == "Unknown") / n() * 100, 2))
cat("Ethnicity_Title - Percent Hispanic or Latino:", Ethnicity_Summary$Percent_Hispanic, "%\n")
cat("Ethnicity_Title - Percent Not Hispanic or Latino:", Ethnicity_Summary$Percent_Not_Hispanic, "%\n")
cat("Ethnicity_Title - Percent Unknown:", Ethnicity_Summary$Percent_Unknown, "%\n")

# report percent in each racial group
race_columns <- c("Race_White", "Race_Black", "Race_Asian", "Race_Pacific_Islander", "Race_American_In
# column for multiple races
df <- df %>%
  mutate(Multiple = ifelse(rowSums(select(., all_of(race_columns)) == "Y") > 1, "Y", "N"))
# percent in each individual category
Race_Summary <- df %>%
  summarize(across(all_of(race_columns), ~ round(sum(. == "Y" & Multiple == "N") / n() * 100, 2))) %>%
  rename_with(~ paste0(.x, "_Percentage"), all_of(race_columns))
# percent in multiple categories
Multiple_Percentage <- df %>%
  summarize(Multiple_Percentage = round(sum(Multiple == "Y") / n() * 100, 2))
# percent unknown
Unknown_Percentage <- 100 - sum(Race_Summary) - Multiple_Percentage
# print race category percents
Race_Summary <- Race_Summary %>%
  bind_rows(data.frame(Multiple = Multiple_Percentage, Unknown = Unknown_Percentage)) %>%
  set_names(nm = c("Race_White_Percentage", "Race_Black_Percentage", "Race_Asian_Percentage",
                   "Race_Pacific_Islander_Percentage", "Race_American_Indian_Percentage",
                   "Race_Other_Percentage", "Multiple_Percentage", "Unknown_Percentage"))
print(Race_Summary)
# save counts for stat testing
Race_Summary <- df %>%
  summarize(across(all_of(c(race_columns, "Multiple")), ~ sum(. == "Y")))

# report test categories
test_columns <- c("culture", "PCR", "SC2_multi", "antigen", "DNA_probe", "unspecified")
# column for multiple tests
df <- df %>%
  mutate(Multiple = ifelse(rowSums(select(., all_of(test_columns)) > 0) > 1, 1, 0))
# percent in each individual category
Test_Summary <- df %>%
  summarize(across(all_of(test_columns), ~ round(sum(. > 0 & Multiple == 0) / n() * 100, 3))) %>%
  rename_with(~ paste0(.x, "_Percentage"), all_of(test_columns))
# percent in multiple categories
Multiple_Percentage <- df %>%
  summarize(Multiple_Percentage = round(sum(Multiple == 1) / n() * 100, 3))
# percent without test
No_Test_Percentage <- 100 - sum(Test_Summary) - Multiple_Percentage
# print test category percents
Test_Summary <- Test_Summary %>%
  bind_rows(data.frame(Multiple = Multiple_Percentage, No_Test = No_Test_Percentage)) %>%
  set_names(nm = c("culture", "PCR", "SC2_multi", "antigen", "DNA_probe",
                   "unspecified", "Multiple", "No_Test"))
```

```r
print(Test_Summary)
# save counts for stat testing
test_columns <- c("culture", "PCR", "SC2_multi", "antigen", "DNA_probe")
Test_Summary <- df %>%
  summarize(across(all_of(c(test_columns, "Multiple")), ~ sum(. > 0)),
            No_Test = sum(rowSums(select(., all_of(test_columns))) == 0))

# print percent with CCC
summarize_and_print_column(df, "Complex_Chronic_Condition_Flag")

# print percent with neonatal CCC
summarize_and_print_column(df, "Premature_And_Neonatal_Flag")

# print percent with CCC per age group
complex_condition_counts <- table(df$Age_Group[df$Complex_Chronic_Condition_Flag == "Y"])
complex_condition_percentages <- round(complex_condition_counts / table(df$Age_Group) * 100, 2)
cat("Age Categories - Complex Chronic Condition Percentage:\n")
print(complex_condition_percentages)

# print percent with positive test per age group
if ("pos" %in% colnames(df)) {
  summarize_and_print_column(df, "pos")
  pos_counts_by_age <- df %>%
    filter(!is.na(Age_Group)) %>%
    group_by(Age_Group) %>%
    summarize(
      Pos_1_Count = sum(pos == 1, na.rm = TRUE),
      Pos_0_Count = sum(pos == 0, na.rm = TRUE),
      Pos_1_Percent = round(mean(pos == 1, na.rm = TRUE) * 100, 5))
  cat("Age Groups - Percentage of rows where pos == 1:\n")
  print(data.frame(pos_counts_by_age))

  # report percent of positive tests by test type
  test_columns <- c("culture", "antigen", "PCR", "SC2_multi", "DNA_probe")
  if(all(test_columns %in% colnames(df))) {
    test_counts <- lapply(test_columns, function(test_col) {
      df %>%
        filter(.data[[test_col]] > 0) %>%
        summarise(
          Pos_1_Count = sum(.data[[test_col]] > 0 & pos == 1, na.rm = TRUE),
          Pos_0_Count = sum(.data[[test_col]] > 0 & pos == 0, na.rm = TRUE),
          Pos_1_Percent = round(sum(.data[[test_col]] > 0 & pos == 1) / sum(.data[[test_col]] > 0) *
        mutate(Test = test_col)})
    test_counts <- bind_rows(test_counts)
    cat("Test Counts - Positive cases where test > 0 and pos == 1 or pos == 0:\n")
    print(test_counts)}}
else {
  pos_counts_by_age = c()
  test_counts = c()}

# statistics of interest only for patients diagnosed with RSV
if ("flu" %in% colnames(df)) {
  # print percent with concurrent influenza
```

```r
  summarize_and_print_column(df, "flu")

  # print percent with concurrent SC2
  summarize_and_print_column(df, "sc2")

  # print percent admitted from ED
  ED_Dispo_Admitted_Percent <- df %>%
  filter(ED_entry == 1) %>%
  filter(!is.na(ED_Dispo)) %>%
  summarize(ED_Dispo_Admitted_Percent = round(sum(ED_Dispo == "ED_Admission") / n() * 100, 2))
  cat("Percent ED Patients Admitted:", ED_Dispo_Admitted_Percent$ED_Dispo_Admitted_Percent, "%\n")

  # only consider inpatients for the following statistics
  # print median (IQR) length of stay
  df = df %>% filter %>%
    filter(Patient_Type_Title != "ED Visit")
  LOS_Summary <- df %>%
    summarize(Median_Length_Of_Stay = round(median(Length_Of_Stay), 2),
    Q1_Length_Of_Stay = round(quantile(Length_Of_Stay, 0.25), 2),
    Q3_Length_Of_Stay = round(quantile(Length_Of_Stay, 0.75), 2))
  cat("Length_Of_Stay - Median:", LOS_Summary$Median_Length_Of_Stay, "\n")
  cat("Length_Of_Stay - Q1:", LOS_Summary$Q1_Length_Of_Stay, "\n")
  cat("Length_Of_Stay - Q3:", LOS_Summary$Q3_Length_Of_Stay, "\n")

  # print percent in each LOS category
  los_groups <- c(0, 1, 2, 4, 7, Inf)
  los_labels <- c("0-1 days", "1-2 days", "2-4 days", "5-7 days", ">7 days")
  df$LOS_Group <- cut(df$Length_Of_Stay, breaks = los_groups, labels = los_labels, right = FALSE)
  los_groups = table(df$LOS_Group)
  los_percents <- round(table(df$LOS_Group) / nrow(df) * 100, 2)
  cat("Length_Of_Stay - Percent:\n")
  print(los_percents)

  # print percent of IPs admitted to intensive care
  df$nicu_icu = ifelse(df$ICU_Flag == "Y" | df$NICU_Flag == "Y", "Y", "N")
  summarize_and_print_column(df, "nicu_icu")

  # print percent of IPs requiring mechanical ventilation
  summarize_and_print_column(df, "Mechanical_Vent_Flag")

  # print percent of IPs requiring ECMO
  summarize_and_print_column(df, "ECMO_Flag")

  # print percent of IPs expiring in hospital
  summarize_and_print_column(df, "Discharge_Mortality_Flag")}

  else {
    los_groups = c()}

 return(list(ptype = ptype_counts, age = age_groups, los = los_groups, race = Race_Summary, test = Test
```

This function takes two dfs with column "Age_Group" (character) as input. It determines the distribution of the categorical variable Age_Group in the reference_df and randomly samples rows in target_df, with

replacement, to generate a new target_df with the same distribution of Age_Group as reference_df.

```
bootstrap_df_by_age <- function(target_df, reference_df, sample_size = nrow(target_df)) {
  # calc the distribution of Age_Group in reference_df
  reference_distribution <- table(reference_df$Age_Group)
  subset_df <- data.frame()
  # loop through each age group
  for (age_group in unique(reference_df$Age_Group)) {
    # calc the target percent for this age group
    target_percentage <- reference_distribution[age_group] / sum(reference_distribution)
    # calc the sample size for this age group
    n_to_select <- round(target_percentage * sample_size)
    # select rows from target_df matching the age group
    available_rows <- target_df %>%
      filter(Age_Group == age_group)
    # make sure there are rows in this age group present
    if (nrow(available_rows) > 0) {
      # randomly sample rows with replacement
      sampled_rows <- available_rows %>% sample_n(size = n_to_select, replace = TRUE)
      subset_df <- bind_rows(subset_df, sampled_rows)}}
  return(subset_df)}
```

Add the columns "Age_Group", "nicu_icu", and "Race" to the dfs with diagnosed and tested patient information.

```
# create the "Age_Group" column from the Admit_Age_In_Days column
pt = add_age_group(pt)
ptest = add_age_group(ptest)

# create the "nicu_icu" column from the binary ICU flags
pt$nicu_icu = ifelse(pt$ICU_Flag == "Y" | pt$NICU_Flag == "Y", "Y", "N")

# create the "Race" column from the binary race columns
race_columns <- c("Race_White", "Race_Black", "Race_Asian", "Race_Pacific_Islander", "Race_American_Indi
pt$Race <- apply(pt[race_columns], 1, function(row) {
  y_columns <- names(row[row == "Y"])
  if (length(y_columns) == 1) {
    return(y_columns)
  } else if (length(y_columns) > 1) {
    return("Multiple")
  } else {
    return("Unknown")}})
ptest$Race <- apply(ptest[race_columns], 1, function(row) {
  y_columns <- names(row[row == "Y"])
  if (length(y_columns) == 1) {
    return(y_columns)
  } else if (length(y_columns) > 1) {
    return("Multiple")
  } else {
    return("Unknown")}})
```

Report the median (IQR) of monthly RSV cases across all hospitals during the intermediate period.

```r
# print median (IQR) of cases between phases
pandemic <- pt %>%
  filter(Date >= as.Date('2020-04-01') & Date < as.Date('2021-04-01')) %>%
  mutate(YearMonth = format(Date, "%Y-%m")) %>%
  group_by(YearMonth) %>%
  summarise(Count = n())

print(paste("Median:", median(pandemic$Count)))
```

```
## [1] "Median: 46.5"
```

```r
print(paste("Q1:", quantile(pandemic$Count, probs = 0.25)))
```

```
## [1] "Q1: 32"
```

```r
print(paste("Q3:", quantile(pandemic$Count, probs = 0.75)))
```

```
## [1] "Q3: 78.5"
```

```r
rm(pandemic)
```

Produce summary statistics for patients tested for RSV in the pre-pandemic and post-emergence phases, and identify statistically significant differences in demographic or clinical variables across the two phases (Table 1).

```r
# pre-pandemic
pre_ptest = subset(ptest, Date < as.Date("2020-04-01"))
print("Pre-pandemic tested patients:")
```

```
## [1] "Pre-pandemic tested patients:"
```

```r
pre_ptest_cat = analyze_dataframe(pre_ptest)
```

```
## ED Discharged Home:
## Count: 119092
## Percentage: 56.59621 %
##
## ED Admitted:
## Count: 73090
## Percentage: 34.73463 %
##
## Admitted Without ED:
## Count: 18242
## Percentage: 8.669163 %
##
## Admit_Age_In_Mo - Median: 8.57 months
## Admit_Age_In_Mo - IQR (Q1 - Q3): 3.12 months - 20.01 months
## Admit_Age_In_Days - Percent:
##
```

```
##  0-3 months  3-6 months 6-12 months   1-2 years   2-4 years  5-17 years
##      23.91       15.63       20.29       19.38        9.99       10.81
## Gender_Title - Percent Male: 55.84 %
## Ethnicity_Title - Percent Hispanic or Latino: 31.59 %
## Ethnicity_Title - Percent Not Hispanic or Latino: 62.8 %
## Ethnicity_Title - Percent Unknown: 5.61 %
##   Race_White_Percentage Race_Black_Percentage Race_Asian_Percentage
## 1                 43.94                    24                  2.31
## 2                    NA                    NA                    NA
##   Race_Pacific_Islander_Percentage Race_American_Indian_Percentage
## 1                             0.35                            0.23
## 2                               NA                              NA
##   Race_Other_Percentage Multiple_Percentage Unknown_Percentage
## 1                  9.91                  NA                 NA
## 2                    NA                1.15              18.11
##   culture   PCR SC2_multi antigen DNA_probe unspecified Multiple
## 1   3.054 5.256        0  83.387     4.774       0.011       NA
## 2      NA    NA       NA      NA        NA          NA    3.518
##                  No_Test
## 1                     NA
## 2 0.000000000000008881784
## Complex_Chronic_Condition_Flag - Percent: 18.45 %
## Premature_And_Neonatal_Flag - Percent: 2.71 %
## Age Categories - Complex Chronic Condition Percentage:
##
##  0-3 months  3-6 months 6-12 months   1-2 years   2-4 years  5-17 years
##      13.77       10.55       11.39       14.64       27.68       51.75
## pos - Percent: 21.5 %
## Age Groups - Percentage of rows where pos == 1:
##     Age_Group Pos_1_Count Pos_0_Count Pos_1_Percent
## 1  0-3 months       13230       37074      26.30010
## 2  3-6 months        9035       23859      27.46702
## 3 6-12 months        9658       33040      22.61933
## 4   1-2 years        8297       32475      20.34975
## 5   2-4 years        3789       17229      18.02741
## 6  5-17 years        1242       21496       5.46222
## Test Counts - Positive cases where test > 0 and pos == 1 or pos == 0:
##   Pos_1_Count Pos_0_Count Pos_1_Percent      Test
## 1         942        8833        9.6368   culture
## 2       41883      139957       23.0329   antigen
## 3        2305       10384       18.1653       PCR
## 4           0           0           NaN SC2_multi
## 5        1054       12674        7.6777 DNA_probe
```

```r
# post-emergence
post_ptest <- subset(ptest, Date >= as.Date("2021-04-01"))
print("Post-emergence tested patients:")
```

```
## [1] "Post-emergence tested patients:"
```

```r
post_ptest_cat = analyze_dataframe(post_ptest)
```

```
## ED Discharged Home:
```

```
## Count: 479941
## Percentage: 66.33967 %
##
## ED Admitted:
## Count: 195318
## Percentage: 26.99776 %
##
## Admitted Without ED:
## Count: 48201
## Percentage: 6.662566 %
##
## Admit_Age_In_Mo - Median: 35.05 months
## Admit_Age_In_Mo - IQR (Q1 - Q3): 11.99 months - 88.21 months
## Admit_Age_In_Days - Percent:
##
##  0-3 months  3-6 months 6-12 months   1-2 years   2-4 years  5-17 years
##        8.50        5.73       10.77       15.63       17.57       41.80
## Gender_Title - Percent Male: 54.19 %
## Ethnicity_Title - Percent Hispanic or Latino: 30.67 %
## Ethnicity_Title - Percent Not Hispanic or Latino: 63.81 %
## Ethnicity_Title - Percent Unknown: 5.51 %
##   Race_White_Percentage Race_Black_Percentage Race_Asian_Percentage
## 1                 48.89                 23.07                  4.33
## 2                    NA                    NA                    NA
##   Race_Pacific_Islander_Percentage Race_American_Indian_Percentage
## 1                             0.45                            0.35
## 2                               NA                              NA
##   Race_Other_Percentage Multiple_Percentage Unknown_Percentage
## 1                 13.73                  NA                 NA
## 2                    NA                3.22               5.96
##   culture     PCR SC2_multi antigen DNA_probe unspecified Multiple
## 1    0.26 10.777    80.782   7.518     0.142           0       NA
## 2      NA     NA        NA      NA        NA          NA    0.521
##                    No_Test
## 1                       NA
## 2 0.000000000000000007771561
## Complex_Chronic_Condition_Flag - Percent: 15.84 %
## Premature_And_Neonatal_Flag - Percent: 1.64 %
## Age Categories - Complex Chronic Condition Percentage:
##
##  0-3 months  3-6 months 6-12 months   1-2 years   2-4 years  5-17 years
##       18.73       11.84       10.32       10.35       12.63       20.63
## pos - Percent: 7.48 %
## Age Groups - Percentage of rows where pos == 1:
##      Age_Group Pos_1_Count Pos_0_Count Pos_1_Percent
## 1  0-3 months        8967       52518      14.58404
## 2  3-6 months        7209       34261      17.38365
## 3 6-12 months        9049       68843      11.61737
## 4   1-2 years       11333      101761      10.02087
## 5   2-4 years       10800      116280       8.49858
## 6  5-17 years        6740      295699       2.22855
## Test Counts - Positive cases where test > 0 and pos == 1 or pos == 0:
##   Pos_1_Count Pos_0_Count Pos_1_Percent      Test
## 1         183        1797        9.2424   culture
```

```
## 2          6583        48200        12.0165    antigen
## 3          8923        72421        10.9695        PCR
## 4         38740       549335         6.5876 SC2_multi
## 5            65          984         6.1964 DNA_probe
```

```r
# compare pre-pandemic and post-emergence phases
agepv = wilcox.test(pre_ptest$Admit_Age_In_Days/30.44, post_ptest$Admit_Age_In_Days/30.44, paired = FALS
print("Age p-value:")
```

```
## [1] "Age p-value:"
```

```r
agepv
```

```
## [1] 0
```

```r
agecatpv = chisq.test(rbind(pre_ptest_cat$age, post_ptest_cat$age))$p.value
print("Age group p-value:")
```

```
## [1] "Age group p-value:"
```

```r
agecatpv
```

```
## [1] 0
```

```r
sexpv = fisher.test(rbind(table(pre_ptest$Gender_Title)[1:2], table(post_ptest$Gender_Title)[1:2]))$p.va
print("Sex p-value:")
```

```
## [1] "Sex p-value:"
```

```r
sexpv
```

```
## [1] 0.0000000000000000000000000000000000000007940738
```

```r
ethnpv = fisher.test(rbind(table(pre_ptest$Ethnicity_Title)[1:2], table(post_ptest$Ethnicity_Title)[1:2]
print("Ethnicity p-value:")
```

```
## [1] "Ethnicity p-value:"
```

```r
ethnpv
```

```
## [1] 0.00000000000000005942188
```

```r
racepv = chisq.test(rbind(pre_ptest_cat$race, post_ptest_cat$race))$p.value
print("Race p-value:")
```

```
## [1] "Race p-value:"
```

```
racepv
```

```
## [1] 0
```

```
cccpv = fisher.test(rbind(table(pre_ptest$Complex_Chronic_Condition_Flag[pre_ptest$Complex_Chronic_Condi
                          table(post_ptest$Complex_Chronic_Condition_Flag[post_ptest$Complex_Chronic_Co
print("CCC p-value:")
```

```
## [1] "CCC p-value:"
```

```
cccpv
```

```
## [1] 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
# Fisher's for each age group's CCC distribution
cc_pv <- list()
age_groups <- unique(as.character(ptest$Age_Group))
for (age_group in age_groups) {
  table_pre <- table(pre_ptest$Complex_Chronic_Condition_Flag[pre_ptest$Age_Group == age_group & pre_pt
  table_post <- table(post_ptest$Complex_Chronic_Condition_Flag[post_ptest$Age_Group == age_group & pos
  pvalue <- fisher.test(rbind(table_pre, table_post))$p.value
  cc_pv[[age_group]] <- pvalue}
# print results
for (i in 1:length(age_groups)) {
  cat("Age Group:", age_groups[i], "- p-value:", cc_pv[[i]], "\n")}
```

```
## Age Group: 1-2 years - p-value: 0.00000000000000000000000000000000000000000000000000000000000000000
## Age Group: 2-4 years - p-value: 0
## Age Group: 0-3 months - p-value: 0.0000000000000000000000000000000000000000000000000000000000000000
## Age Group: 5-17 years - p-value: 0
## Age Group: 3-6 months - p-value: 0.00000002722992
## Age Group: 6-12 months - p-value: 0.00000001276932
```

```
prempv = fisher.test(rbind(table(pre_ptest$Premature_And_Neonatal_Flag[pre_ptest$Premature_And_Neonatal_
print("Premature & neonatal CCC p-value:")
```

```
## [1] "Premature & neonatal CCC p-value:"
```

```
prempv
```

```
## [1] 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
ptypepv = chisq.test(rbind(unlist(pre_ptest_cat$ptype), unlist(post_ptest_cat$ptype)))$p.value
print("Patient type p-value:")
```

```
## [1] "Patient type p-value:"
```

```
ptypepv
```

```
## [1] 0
```

```
testpv = chisq.test(rbind(unlist(pre_ptest_cat$test), unlist(post_ptest_cat$test)))$p.value
print("Test type p-value:")
```

```
## [1] "Test type p-value:"
```

```
testpv
```

```
## [1] 0
```

```
# Fisher's for each age group's test positivity
pos_pv <- list()
age_groups <- unique(as.character(ptest$Age_Group))
for (age_group in age_groups) {
  table_pre <- table(pre_ptest$pos[pre_ptest$Age_Group == age_group])
  table_post <- table(post_ptest$pos[post_ptest$Age_Group == age_group])
  pvalue <- fisher.test(rbind(table_pre, table_post))$p.value
  pos_pv[[age_group]] <- pvalue}
for (i in 1:length(age_groups)) {
  cat("Age stratified test positivity:", age_groups[i], " p-value:", pos_pv[[i]], "\n")}
```

```
## Age stratified test positivity: 1-2 years  p-value: 0
## Age stratified test positivity: 2-4 years  p-value: 0
## Age stratified test positivity: 0-3 months  p-value: 0
## Age stratified test positivity: 5-17 years  p-value: 0.00000000000000000000000000000000000000000000000000000
## Age stratified test positivity: 3-6 months  p-value: 0.00000000000000000000000000000000000000000000000000000
## Age stratified test positivity: 6-12 months  p-value: 0
```

```
pos_pv = fisher.test(rbind(table(pre_ptest$pos), table(post_ptest$pos)))$p.value
print("Positive test p-value:")
```

```
## [1] "Positive test p-value:"
```

```
pos_pv
```

```
## [1] 0
```

```
pos_cult_pv = fisher.test(rbind(table(pre_ptest$pos[pre_ptest$culture > 0]), table(post_ptest$pos[post_
print("Positive culture p-value:")
```

```
## [1] "Positive culture p-value:"
```

```
pos_cult_pv
```

```
## [1] 0.6152608
```

```
pos_ant_pv = fisher.test(rbind(table(pre_ptest$pos[pre_ptest$antigen > 0]), table(post_ptest$pos[post_p
print("Positive antigen p-value:")
```

[1] "Positive antigen p-value:"

```
pos_ant_pv
```

[1] 0

```
pos_pcr_pv = fisher.test(rbind(table(pre_ptest$pos[pre_ptest$PCR > 0]), table(post_ptest$pos[post_ptest$
print("Positive PCR p-value:")
```

[1] "Positive PCR p-value:"

```
pos_pcr_pv
```

[1] 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

```
pos_dna_pv = fisher.test(rbind(table(pre_ptest$pos[pre_ptest$DNA_probe > 0]), table(post_ptest$pos[post_
print("Positive DNA probe p-value:")
```

[1] "Positive DNA probe p-value:"

```
pos_dna_pv
```

[1] 0.08967613

```
rm(list = ls(pattern = "pv|tab"))
rm(i)
```

Produce summary statistics for patients diagnosed with RSV in the pre-pandemic and post-emergence phases, and identify statistically significant differences in demographic or clinical variables across the two phases (Table 2).

```
pt = pt %>% filter(Date < as.Date("2020-04-01") | Date >= as.Date("2021-04-01"))
pt$phase = ifelse(pt$Date < as.Date("2020-04-01"), "Pre-Pandemic", "Post-Emergence")

# pre-pandemic
pre_pt = subset(pt, Date < as.Date("2020-04-01"))
print("Pre-pandemic diagnosed patients:")
```

[1] "Pre-pandemic diagnosed patients:"

```
pre_pt_cat = analyze_dataframe(pre_pt)
```

```
## ED Discharged Home:
## Count: 62933
## Percentage: 31.13476 %
##
## ED Admitted:
## Count: 105867
## Percentage: 52.37544 %
##
## Admitted Without ED:
## Count: 33331
## Percentage: 16.4898 %
##
## Admit_Age_In_Mo - Median: 6.87 months
## Admit_Age_In_Mo - IQR (Q1 - Q3): 2.6 months - 17.48 months
## Admit_Age_In_Days - Percent:
##
##  0-3 months  3-6 months 6-12 months   1-2 years   2-4 years  5-17 years
##       27.99       17.85       18.85       17.72       11.16        6.43
## Gender_Title - Percent Male: 55.31 %
## Ethnicity_Title - Percent Hispanic or Latino: 25.45 %
## Ethnicity_Title - Percent Not Hispanic or Latino: 69.83 %
## Ethnicity_Title - Percent Unknown: 4.72 %
##   Race_White_Percentage Race_Black_Percentage Race_Asian_Percentage
## 1                 58.61                 18.43                   2.8
## 2                    NA                    NA                    NA
##   Race_Pacific_Islander_Percentage Race_American_Indian_Percentage
## 1                             0.53                            0.29
## 2                               NA                              NA
##   Race_Other_Percentage Multiple_Percentage Unknown_Percentage
## 1                 10.31                  NA                 NA
## 2                    NA                1.49               7.54
##   culture  PCR SC2_multi antigen DNA_probe unspecified Multiple No_Test
## 1   0.282 1.09        0 20.293     0.274       0.002       NA      NA
## 2      NA   NA       NA     NA        NA          NA    0.441  77.618
## Complex_Chronic_Condition_Flag - Percent: 16.84 %
## Premature_And_Neonatal_Flag - Percent: 3.1 %
## Age Categories - Complex Chronic Condition Percentage:
##
##  0-3 months  3-6 months 6-12 months   1-2 years   2-4 years  5-17 years
##       12.14        8.91       11.89       16.67       29.22       52.76
## flu - Percent: 2.19 %
## sc2 - Percent: 0 %
## Percent ED Patients Admitted: 50.55 %
## Length_Of_Stay - Median: 3
## Length_Of_Stay - Q1: 2
## Length_Of_Stay - Q3: 5
## Length_Of_Stay - Percent:
##
## 0-1 days 1-2 days 2-4 days 5-7 days  >7 days
##     3.10    19.91    36.62    22.90    17.47
## nicu_icu - Percent: 32.25 %
## Mechanical_Vent_Flag - Percent: 11.9 %
## ECMO_Flag - Percent: 0.22 %
## Discharge_Mortality_Flag - Percent: 0.24 %
```

```r
# post-emergence
post_pt <- subset(pt, Date >= as.Date("2021-04-01"))
print("Post-emergence diagnosed patients:")
```

```
## [1] "Post-emergence diagnosed patients:"
```

```r
post_pt_cat = analyze_dataframe(post_pt)
```

```
## ED Discharged Home:
## Count: 62192
## Percentage: 44.58016 %
##
## ED Admitted:
## Count: 64719
## Percentage: 46.39155 %
##
## Admitted Without ED:
## Count: 12595
## Percentage: 9.028286 %
##
## Admit_Age_In_Mo - Median: 11.89 months
## Admit_Age_In_Mo - IQR (Q1 - Q3): 4.2 months - 28.32 months
## Admit_Age_In_Days - Percent:
##
##  0-3 months  3-6 months 6-12 months    1-2 years    2-4 years  5-17 years
##       18.18       14.56       17.40        20.00        18.46       11.40
## Gender_Title - Percent Male: 54.75 %
## Ethnicity_Title - Percent Hispanic or Latino: 27.09 %
## Ethnicity_Title - Percent Not Hispanic or Latino: 67.74 %
## Ethnicity_Title - Percent Unknown: 5.17 %
##   Race_White_Percentage Race_Black_Percentage Race_Asian_Percentage
## 1                 57.03                 19.15                   3.3
## 2                    NA                    NA                    NA
##   Race_Pacific_Islander_Percentage Race_American_Indian_Percentage
## 1                              0.8                             0.3
## 2                               NA                              NA
##   Race_Other_Percentage Multiple_Percentage Unknown_Percentage
## 1                 11.21                  NA                 NA
## 2                    NA                2.72               5.49
##   culture   PCR SC2_multi antigen DNA_probe unspecified Multiple No_Test
## 1   0.126 2.632    31.042   4.645     0.042           0       NA      NA
## 2      NA    NA        NA      NA        NA          NA     0.29  61.223
## Complex_Chronic_Condition_Flag - Percent: 11.69 %
## Premature_And_Neonatal_Flag - Percent: 2.57 %
## Age Categories - Complex Chronic Condition Percentage:
##
##  0-3 months  3-6 months 6-12 months    1-2 years    2-4 years  5-17 years
##       11.98        6.46        7.74         9.45        13.21       25.46
## flu - Percent: 1.38 %
## sc2 - Percent: 2.73 %
## Percent ED Patients Admitted: 38.7 %
## Length_Of_Stay - Median: 2
```

```
## Length_Of_Stay - Q1: 1
## Length_Of_Stay - Q3: 4
## Length_Of_Stay - Percent:
##
## 0-1 days 1-2 days 2-4 days 5-7 days  >7 days
##     3.76    23.35    39.83    20.99    12.08
## nicu_icu - Percent: 27.16 %
## Mechanical_Vent_Flag - Percent: 7.36 %
## ECMO_Flag - Percent: 0.15 %
## Discharge_Mortality_Flag - Percent: 0.16 %
```

```r
# compare pre-pandemic and post-emergence phases
agepv = wilcox.test(pre_pt$Admit_Age_In_Days/30.44, post_pt$Admit_Age_In_Days/30.44, paired = FALSE)$p.v
print("Age p-value:")
```

```
## [1] "Age p-value:"
```

```r
agepv
```

```
## [1] 0
```

```r
agecatpv = chisq.test(rbind(pre_pt_cat$age, post_pt_cat$age))$p.value
print("Age group p-value:")
```

```
## [1] "Age group p-value:"
```

```r
agecatpv
```

```
## [1] 0
```

```r
sexpv = fisher.test(rbind(table(pre_pt$Gender_Title)[1:2], table(post_pt$Gender_Title)[1:2]))$p.value
print("Sex p-value:")
```

```
## [1] "Sex p-value:"
```

```r
sexpv
```

```
## [1] 0.001180125
```

```r
ethnpv = fisher.test(rbind(table(pre_pt$Ethnicity_Title)[1:2], table(post_pt$Ethnicity_Title)[1:2]))$p.v
print("Ethnicity p-value:")
```

```
## [1] "Ethnicity p-value:"
```

```r
ethnpv
```

```
## [1] 0.00000000000000000000000000000003077643
```

```r
racepv = chisq.test(rbind(pre_pt_cat$race, post_pt_cat$race))$p.value
print("Race p-value:")
```

```
## [1] "Race p-value:"
```

```r
racepv
```

```
## [1] 0.00000000000000000000000000000000000000000000000000000000000000000000000000
```

```r
flupv = fisher.test(rbind(table(pre_pt$flu), table(post_pt$flu)))$p.value
print("Flu p-value:")
```

```
## [1] "Flu p-value:"
```

```r
flupv
```

```
## [1] 0.000000000000000000000000000000000000000000000000000000000000005255498
```

```r
print("SC2 p-value:")
```

```
## [1] "SC2 p-value:"
```

```r
sc2pv = fisher.test(rbind(table(pre_pt$sc2), table(post_pt$sc2)))$p.value
sc2pv
```

```
## [1] 0
```

```r
cccpv = fisher.test(rbind(table(pre_pt$Complex_Chronic_Condition_Flag), table(post_pt$Complex_C
print("CCC p-value:")
```

```
## [1] "CCC p-value:"
```

```r
cccpv
```

```
## [1] 0
```

```r
# Fisher's for each age group's CCC distribution
cc_pv <- list()
age_groups <- unique(as.character(pt$Age_Group))
for (age_group in age_groups) {
  table_pre <- table(pre_pt$Complex_Chronic_Condition_Flag[pre_pt$Age_Group == age_group])
  table_post <- table(post_pt$Complex_Chronic_Condition_Flag[post_pt$Age_Group == age_group])
  pvalue <- fisher.test(rbind(table_pre, table_post))$p.value
  cc_pv[[age_group]] <- pvalue}
for (i in 1:length(age_groups)) {
  cat("Age stratified CCC:", age_groups[i], "p-value:", cc_pv[[i]], "\n")}
```

```
## Age stratified CCC: 0-3 months p-value: 0.5015325
## Age stratified CCC: 3-6 months p-value: 0.000000000000000000000001500192
## Age stratified CCC: 2-4 years p-value: 0
## Age stratified CCC: 6-12 months p-value: 0.00000000000000000000000000000000000000000000000000000000000
## Age stratified CCC: 1-2 years p-value: 0.00000000000000000000000000000000000000000000000000000000000
## Age stratified CCC: 5-17 years p-value: 0
```

```r
prempv = fisher.test(rbind(table(pre_pt$Premature_And_Neonatal_Flag), table(post_pt$Premature_And_Neona
print("Premature & neonatal CCC p-value:")
```

```
## [1] "Premature & neonatal CCC p-value:"
```

```r
prempv
```

```
## [1] 0.000000000000000000001076395
```

```r
ptypepv = chisq.test(rbind(unlist(pre_pt_cat$ptype), unlist(post_pt_cat$ptype)))$p.value
print("Patient type p-value:")
```

```
## [1] "Patient type p-value:"
```

```r
ptypepv
```

```
## [1] 0
```

```r
testpv = chisq.test(rbind(unlist(pre_pt_cat$test), unlist(post_pt_cat$test)))$p.value
print("Test type p-value:")
```

```
## [1] "Test type p-value:"
```

```r
testpv
```

```
## [1] 0
```

```r
admpv = fisher.test(rbind(table(pre_pt$ED_Dispo[pre_pt$ED_entry == 1]),
                          table(post_pt$ED_Dispo[post_pt$ED_entry == 1])))$p.value
print("ED admission p-value:")
```

```
## [1] "ED admission p-value:"
```

```r
admpv
```

```
## [1] 0
```

```r
lospv = wilcox.test(pre_pt$Length_Of_Stay[pre_pt$Patient_Type_Title != "ED Visit"],
                    post_pt$Length_Of_Stay[post_pt$Patient_Type_Title != "ED Visit"], paired = FALSE)$p
print("Length of stay p-value:")
```

```
## [1] "Length of stay p-value:"
```

```
lospv
```

```
## [1] 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
loscatpv = chisq.test(rbind(pre_pt_cat$los, post_pt_cat$los))$p.value
print("Length of stay (categorical) p-value:")
```

```
## [1] "Length of stay (categorical) p-value:"
```

```
loscatpv
```

```
## [1] 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
icupv = fisher.test(rbind(table(pre_pt$nicu_icu[pre_pt$Patient_Type_Title != "ED Visit"]),
                          table(post_pt$nicu_icu[post_pt$Patient_Type_Title != "ED Visit"])))$p.value
print("ICU p-value:")
```

```
## [1] "ICU p-value:"
```

```
icupv
```

```
## [1] 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
mechpv = fisher.test(rbind(table(pre_pt$Mechanical_Vent_Flag[pre_pt$Patient_Type_Title != "ED Visit"]),
                           table(post_pt$Mechanical_Vent_Flag[post_pt$Patient_Type_Title != "ED Visit"]
print("Mechanical ventilation p-value:")
```

```
## [1] "Mechanical ventilation p-value:"
```

```
mechpv
```

```
## [1] 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
ecmopv = fisher.test(rbind(table(pre_pt$ECMO_Flag[pre_pt$Patient_Type_Title != "ED Visit"]),
                           table(post_pt$ECMO_Flag[post_pt$Patient_Type_Title != "ED Visit"])))$p.value
print("ECMO p-value:")
```

```
## [1] "ECMO p-value:"
```

```
ecmopv
```

```
## [1] 0.0004307284
```

```
deathpv = fisher.test(rbind(table(pre_pt$Discharge_Mortality_Flag[pre_pt$Patient_Type_Title != "ED Visi
                            table(post_pt$Discharge_Mortality_Flag[post_pt$Patient_Type_Title != "ED Vis
print("Death p-value:")
```

```
## [1] "Death p-value:"
```

```
deathpv
```

```
## [1] 0.0003390302
```

```
rm(list = ls(pattern = "pv|tab"))
rm(i)
```

Plot the proportion of tested patients in each age group (Figure 2A).

```
test_age = rbind(pre_ptest_cat$age/nrow(pre_ptest), post_ptest_cat$age/nrow(post_ptest))
rownames(test_age) <- c("Pre-Pandemic", "Post-Emergence")

# pivot the data to long format for plotting
test_age_long = gather(as.data.frame(test_age), key = "Age Group", value = "Proportion")
test_age_long$Group <- rownames(test_age)
test_age_long$Group <- factor(test_age_long$Group, levels = c("Pre-Pandemic", "Post-Emergence"))
test_age_long$`Age Group` <- factor(test_age_long$`Age Group`, levels = c("0-3 months",
                              "3-6 months", "6-12 months", "1-2 years", "2-4 years", "5-17 years"))

# plot the proportions
fig2a <- ggplot(test_age_long, aes(x = `Age Group`, y = Proportion, fill = Group)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "",
       x = "Age Group",
       y = "Proportion of Tests") +
  scale_fill_manual(values = c("grey", "darkmagenta")) +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5),
        legend.position = "top",
        text = element_text(size = 12)) +
  guides(fill = guide_legend(title = NULL))
fig2a
```
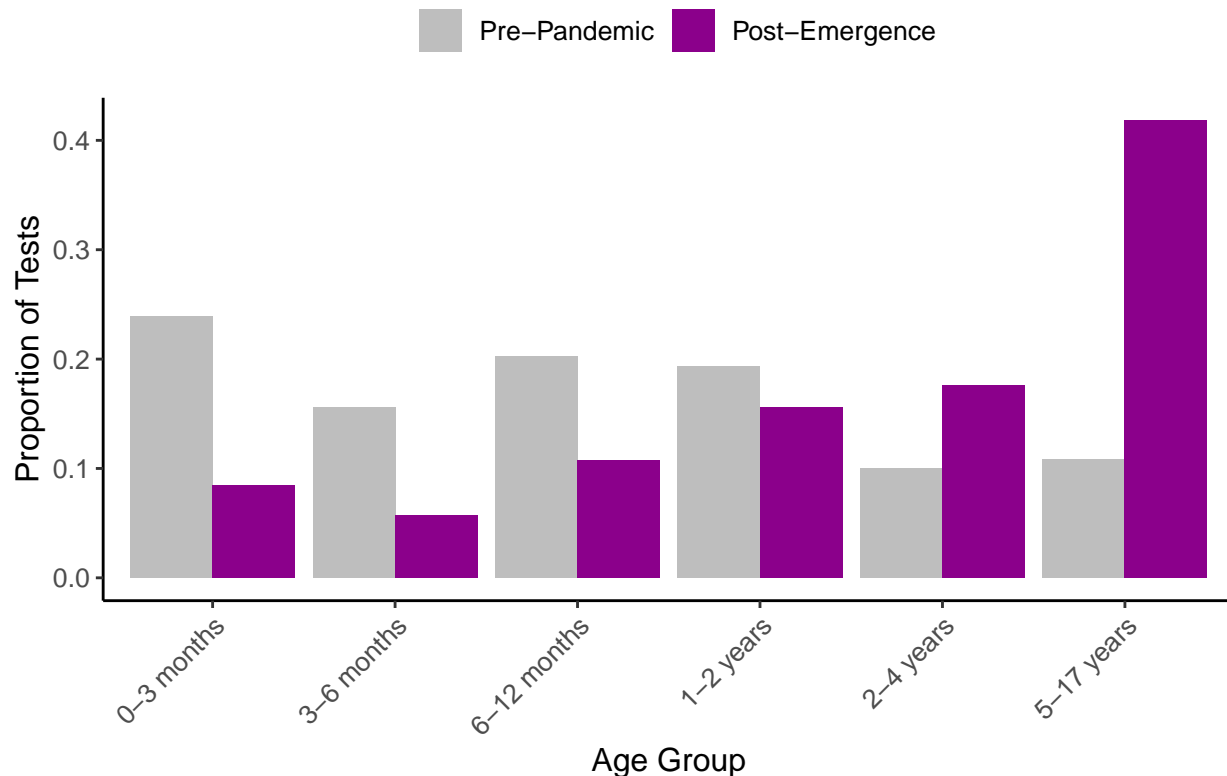
```r
rm(test_age, test_age_long)
```

Use bootstrapping to generate a post-emergence testing df with the age distribution (in 90-day bins) derived from the pre-pandemic testing df. Plot the pre-pandemic, post-emergence, and test-adjusted age distributions of tested patients (Supplementary Figure 3).

```r
# match testing by 3-month periods of age
post_ptest$Age_Group <- as.character(floor(post_ptest$Admit_Age_In_Days / 90))
pre_ptest$Age_Group <- as.character(floor(pre_ptest$Admit_Age_In_Days / 90))

# bootstrapping post-emergence phase tests
boot_post = bootstrap_df_by_age(post_ptest, pre_ptest)

# plot the pre-pandemic, post-emergence, and test-adjusted age distributions
age_tests <- rbind(data.frame(Age = pre_ptest$Admit_Age_In_Days/365, Group = "Pre-Pandemic"),
                   data.frame(Age = post_ptest$Admit_Age_In_Days/365, Group = "Post-Emergence"),
                   data.frame(Age = boot_post$Admit_Age_In_Days/365, Group = "Post-Emergence\n(Test Adj
age_tests$Group = factor(age_tests$Group, levels = c("Pre-Pandemic", "Post-Emergence", "Post-Emergence\n

sf3 <- ggplot(age_tests, aes(x = Group, y = Age)) +
  geom_boxplot(outlier.shape = NA, fill = "grey") +
  labs(x = "", y = "Age of Tested Individuals (Years)") +
  theme_classic() +
  coord_cartesian(ylim = c(0, 19)) +
  geom_signif(comparisons = list(c("Pre-Pandemic","Post-Emergence"),
```

```
                              c("Pre-Pandemic", "Post-Emergence\n(Test Adjusted)"),
                              c("Post-Emergence", "Post-Emergence\n(Test Adjusted)")),
              test = "wilcox.test",
              map_signif_level = TRUE,
              textsize = 3,
              y_position = (c(17, 18.5, 17.5)),
              tip_length = 0.005) +
  scale_y_continuous(breaks = c(0, 2, 4, 6, 8, 10, 12, 14, 16, 18)) +
  theme(legend.position = "none",
        text = element_text(size = 12))
sf3
```
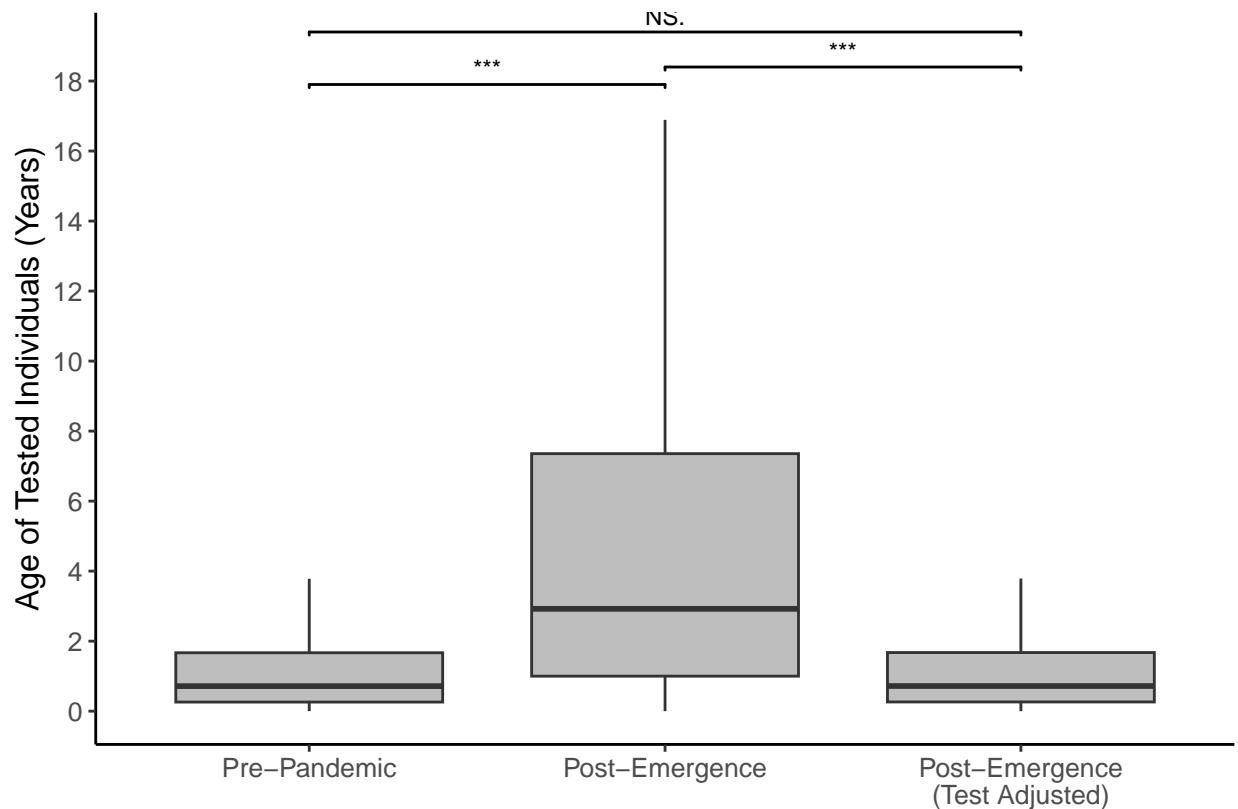


```
ggsave("figs/suppfig3.pdf", plot = sf3, width = 6, height = 6)
rm(age_tests, sf3)
```

Assess whether the distribution of patients diagnosed with RSV would have changed if the age of patients tested for RSV remained constant from the pre-pandemic to the post-emergence phase. Conduct 50 replicates of bootstrapping to ensure that the results are robust to sampling error.

```
# create empty df to store results
age_sim <- data.frame(min = numeric(0), q1 = numeric(0), median = numeric(0),
                      q3 = numeric(0), max = numeric(0))
for (i in 1:50){
  # bootstrap tests according to pre-pandemic age distribution
  boot_post = bootstrap_df_by_age(post_ptest, pre_ptest)
```

```
  # isolate positive bootstrapped tests
  pt_post = boot_post %>% filter(Discharge_ID %in% pt$Discharge_ID)
  # append age range to df
  age_sim[nrow(age_sim)+1, ] <- c(quantile(pt_post$Admit_Age_In_Days))}

# check for consistency across replicates
summary(age_sim/30.44) # convert to months
```

```
##      min              q1              median            q3              max
##  Min.   :0    Min.   :2.431    Min.   :5.388    Min.   :12.48    Min.   :214.6
##  1st Qu.:0    1st Qu.:2.431    1st Qu.:5.420    1st Qu.:12.65    1st Qu.:215.4
##  Median :0    Median :2.464    Median :5.420    Median :12.71    Median :215.7
##  Mean   :0    Mean   :2.453    Mean   :5.431    Mean   :12.70    Mean   :215.6
##  3rd Qu.:0    3rd Qu.:2.464    3rd Qu.:5.453    3rd Qu.:12.75    3rd Qu.:215.9
##  Max.   :0    Max.   :2.497    Max.   :5.486    Max.   :12.84    Max.   :215.9
```

```
rm(age_sim)
```

Plot the pre-pandemic, post-emergence, and test-adjusted age distributions of patients with a laboratory-confirmed diagnosis of RSV (Figure 2B).

```
# create plotting df
age_adj_df <- rbind(data.frame(Age = (pre_pt %>% filter(Num_Tests > 0))$Admit_Age_In_Days/365, Group =
                    data.frame(Age = (post_pt %>% filter(Num_Tests > 0))$Admit_Age_In_Day/365, Group =
                    data.frame(Age = pt_post$Admit_Age_In_Day/365, Group = "Post-Emergence\n(Test Adjus
age_adj_df$Group = factor(age_adj_df$Group, levels = c("Pre-Pandemic", "Post-Emergence", "Post-Emergence

# print summary statistics for each group
tapply(12*age_adj_df$Age, age_adj_df$Group, quantile) # convert to months
```

```
## $`Pre-Pandemic`
##        0%        25%        50%        75%        100%
##    0.000000    2.498630    6.049315   13.873973  215.736986
##
## $`Post-Emergence`
##        0%        25%        50%        75%        100%
##    0.00000    4.70137   13.47945   30.46849  216.09863
##
## $`Post-Emergence\n(Test Adjusted)`
##        0%        25%        50%        75%        100%
##    0.000000    2.432877    5.424658   12.723288  215.901370
```

```
# plot age distributions
fig2b <- ggplot(age_adj_df, aes(x = Group, y = Age)) +
  geom_boxplot(outlier.shape = NA, fill = "grey") +
  labs(x = "", y = "Age of Diagnosed Individuals (Years)") +
  theme_classic() +
  coord_cartesian(ylim = c(0, 7)) +
  geom_signif(comparisons = list(c("Pre-Pandemic","Post-Emergence"),
                                 c("Pre-Pandemic", "Post-Emergence\n(Test Adjusted)"),
                                 c("Post-Emergence", "Post-Emergence\n(Test Adjusted)")),
              test = "wilcox.test",
```

```
                map_signif_level = TRUE,
                textsize = 3,
                y_position = c(5.1,5.5,6.0),
                tip_length = 0.003) +
    theme(legend.position = "none",
          text = element_text(size = 12))
fig2b
```
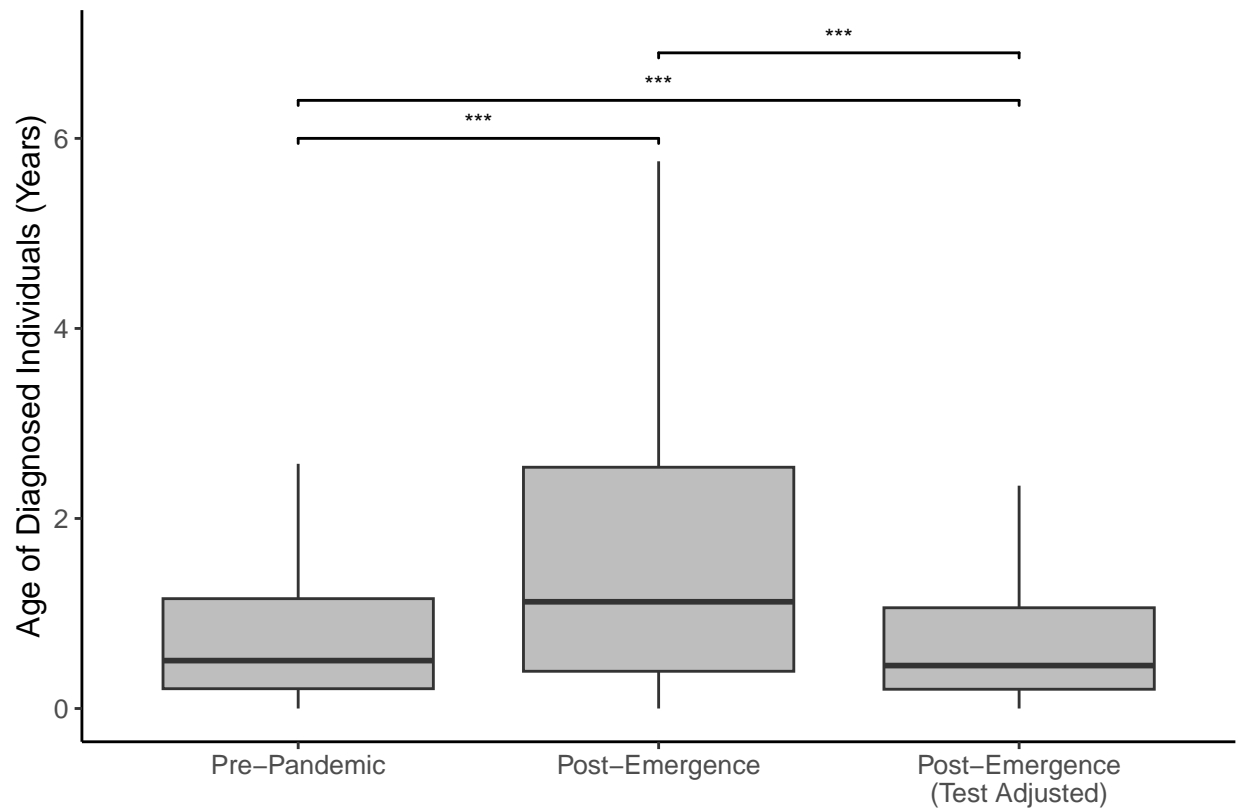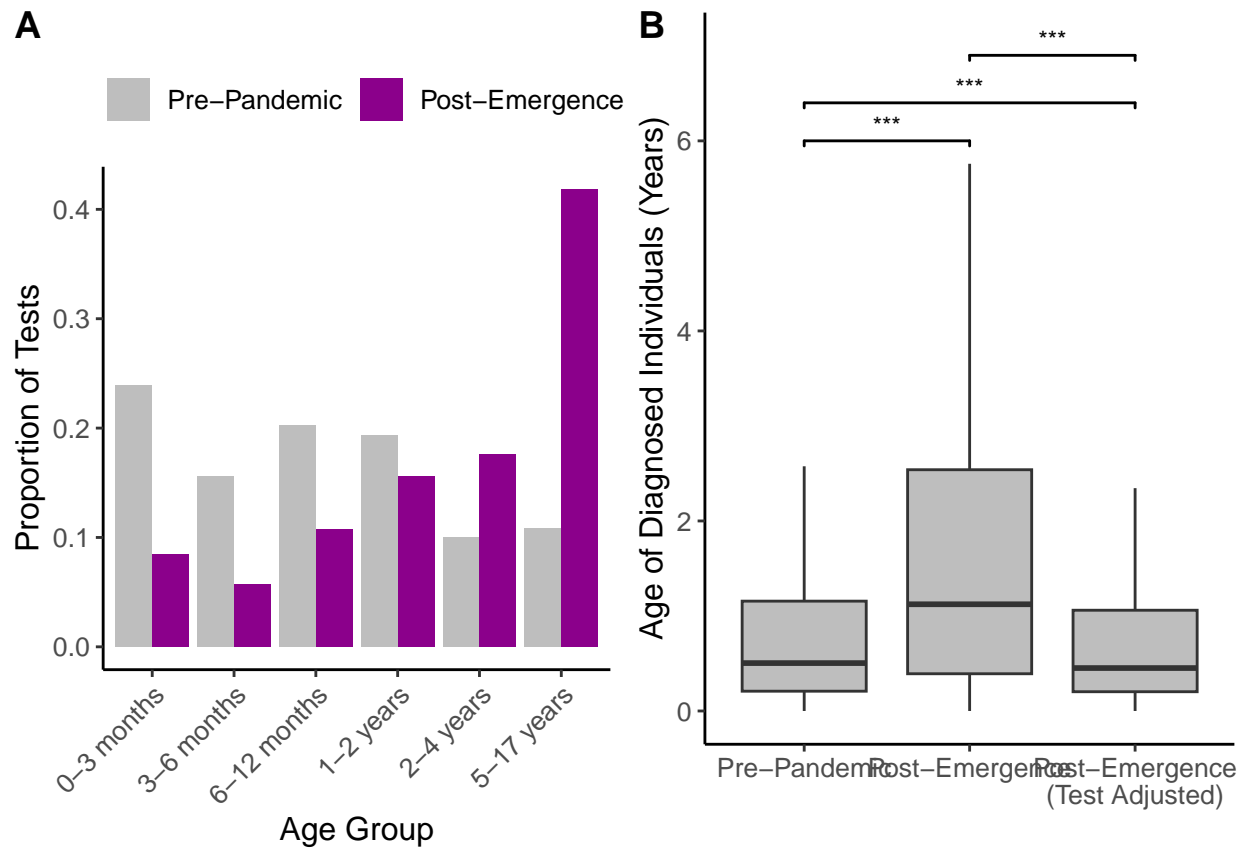


```
rm(age_adj_df)
```

Generate Figure 2.

```
fig2 = plot_grid(fig2a, fig2b, nrow = 1, labels = c("A", "B"))
fig2
```

```
ggsave("figs/fig2.pdf", plot = fig2, width = 11, height = 5)
```