STATS 485 Unit 1 Paper 1 Appendix Version 2

Caroline Moy

2025-02-06

Required packages and data download.

```
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(knitr)

covid_df = readr::read_tsv("https://dept.stat.lsa.umich.edu/~bbh/s485/data/severe_covid_IL_2021_08_15.ts
```

```
## Rows: 7 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): ages
## dbl (4): cases_unvax, cases_vax, pop_unvax, pop_vax
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Overview

This appendix accompanies the Unit 1 paper "COVID-19 Severe Disease Proportion and Vaccine Efficacy Estimation." First, required packages and datasets will be downloaded, followed by light data exploration, and then the computations required for the paper. The point estimates and modified-Wilson confidence intervals will be computed for the proportion of severe COVID-19 disease incident for each distinct age group and vaccination status (Question 1). Then the age-specific vaccine efficacy confidence intervals will be computed using the proportion confidence intervals, procedure explained below (Question 2). Finally, the point estimate of the overall unweighted vaccine efficacy will be calculated (Question 3). These statistics will be used to compare the age-stratified vaccine efficacy confidence intervals and the point estimate of the overall vaccine efficacy, so see if the two statistics arrive at different conclusions.

Light Exploratory Data Analysis

First, the total proportion of vaccinated individuals is calculated. Then the percentage of the severe cases in which the patient is vaccinated.

```
sum(covid_df['pop_vax']) / (sum(covid_df['pop_vax']) + sum(covid_df['pop_unvax']))
```

```
## [1] 0.8121943
```

```
sum(covid_df['cases_vax']) / (sum(covid_df['cases_vax']) + sum(covid_df['cases_unvax']))
```

```
## [1] 0.584466
```

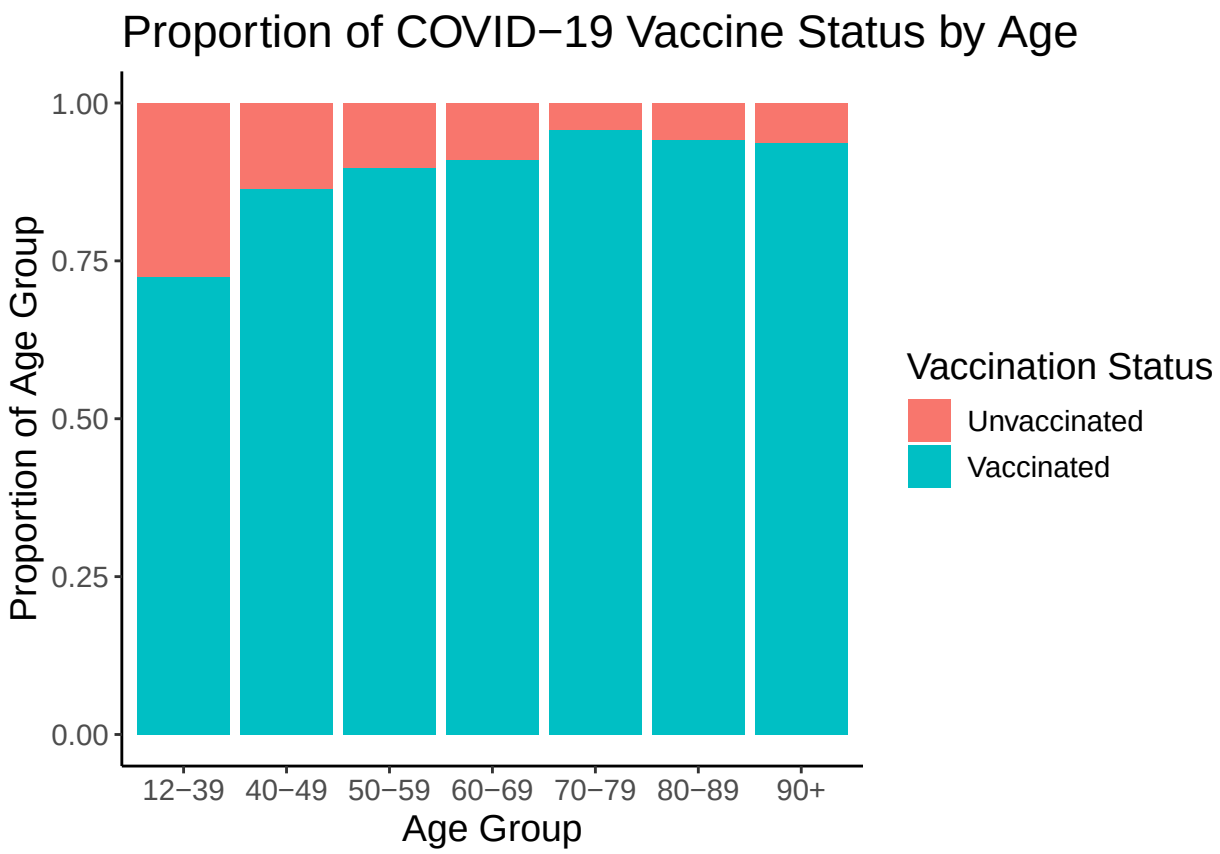
Additional variable creation to explore the data set further.

```
covid_df['ages'] = factor(covid_df[['ages']], levels = c('12-39', '40-49', '50-59', '60-69', '70-79', '8
covid_df['p_vax'] = covid_df['cases_vax'] / covid_df['pop_vax']
covid_df['p_unvax'] = covid_df['cases_unvax'] / covid_df['pop_unvax']
covid_df['overall_p_vax'] = (covid_df['pop_vax']) / (covid_df['pop_unvax'] + covid_df['pop_vax'])
covid_df['overall_p_unvax'] = (covid_df['pop_unvax']) / (covid_df['pop_unvax'] + covid_df['pop_vax'])
```

Stacked bar chart of the proportions of unvaccinated and vaccinated individuals in each age group.

```
changed_df = covid_df %>%
  dplyr::select(ages,overall_p_vax, overall_p_unvax) %>%
  pivot_longer(cols = c(overall_p_vax, overall_p_unvax), names_to = "vax_stat", values_to = "percentage

ggplot(data = changed_df, aes(x = ages, y = percentage, fill = vax_stat)) +
   geom_bar(stat = 'identity') +
  ggtitle("Proportion of COVID-19 Vaccine Status by Age") +
  xlab('Age Group') +
  ylab('Proportion of Age Group') +
  guides(fill=guide_legend(title="Vaccination Status")) +
  scale_fill_discrete(name = "vax_stat", labels = c("Unvaccinated", "Vaccinated")) +
  theme_classic() +
  theme(text=element_text(size=14))
```



Question 1

Question 1 of the analysis asked for the 95% confidence interval for the underlying rates of severe disease for vaccinated and unvaccinated populations specific to age.

General Kappa to be used throughout calculations.

```
k = qnorm(0.975)
```

Function to create center of Wilson confidence interval and testing.

```
wilson_center = function(n, phat){
  X = n * phat
  center = (X + k^2 / 2) / (n + k^2)
  return(center)
}

# expected 0.5
wilson_center(10, 0.5)
```

```
## [1] 0.5
```

```
# expected 0.2678...
wilson_center(50, .25)
```

```
## [1] 0.2678369
```

```
# expected 0.8836
wilson_center(90, .9)
```

```
## [1] 0.8836257
```

```r
# expected 0.1019...
wilson_center(15, 0)
```

```
## [1] 0.1019417
```

Function for standard error of Wilson confidence interval and testing.

```r
wilson_se = function(n, phat){
  se = (k * sqrt(n)) / (n + k^2) * sqrt(phat * (1 - phat) + k^2 / (4*n))
  return(se)
}

# expected 0.2634...
wilson_se(10, 0.5)
```

```
## [1] 0.2634069
```

```r
# expected 0.1170...
wilson_se(50, .25)
```

```
## [1] 0.1170292
```

```r
# expected 0.0629...
wilson_se(90, .9)
```

```
## [1] 0.0628675
```

```r
# expected 0.1019...
wilson_se(15, 0)
```

```
## [1] 0.1019417
```

Modified Lower Bound (for when X is less than or equal to 3) and testing.

```r
lb_modified = function(alpha, n, phat){
  lambda = 0.5 * qchisq(p = alpha, df = 2 * n * phat)
  modified_lowbound = lambda / n
  return(modified_lowbound)
}

# expected 0.1461...
lb_modified(0.05, 50, .25)
```

```
## [1] 0.1461141
```

```r
# expected practically 0
lb_modified(0.1, 15, 0.001)
```

```
## [1] 8.163561e-69
```

Modified Upper Bound (for when X is greater than or equal to the sample size minus three) and testing.

```r
ub_modified = function(alpha, n, phat){
  lambda = 0.5 * qchisq(p = 1 - alpha, df = 2 * n * phat)
  modified_upbound = lambda / n
  return(modified_upbound)
}

# expected 0.376...
ub_modified(0.05, 50, .25)
```

```
## [1] 0.3765248
```

```
# expected 3.37 x 10^5
ub_modified(0.1, 15, 0.001)
```

```
## [1] 3.37492e-05
```

Functions for the creation of modified-Wilson confidence intervals' lower and upper bounds.

```
wilson_lower_ci = function(n, phat){
  modified_lowbound = lb_modified(alpha = 0.05, n = n, phat = phat)
  classic_lowbound = wilson_center(n, phat) - wilson_se(n, phat)
  low_bound = ifelse(n * phat <= 3, modified_lowbound, classic_lowbound)
  return(low_bound)
}

wilson_upper_ci = function(n, phat){
  modified_upbound = ub_modified(alpha = 0.05, n = n, phat = phat)
  classic_upbound = wilson_center(n, phat) + wilson_se(n, phat)
  up_bound = ifelse(n * phat >= n - 3, modified_upbound, classic_upbound)
  return(up_bound)
}
```

Using the modified-Wilson confidence interval functions to calculate the lower and upper bounds of the 95% confidence intervals for each age group and vaccination status. The age group, 95% confidence interval bounds, point estimates, and vaccination status are coerced into a data frame for use later.

```
lower_bound_vax = wilson_lower_ci(n = covid_df[['pop_vax']], phat = covid_df[['p_vax']])
upper_bound_vax = wilson_upper_ci(n = covid_df[['pop_vax']], phat = covid_df[['p_vax']])

lower_bound_unvax = wilson_lower_ci(n = covid_df[['pop_unvax']], phat = covid_df[['p_unvax']])
upper_bound_unvax = wilson_upper_ci(n = covid_df[['pop_unvax']], phat = covid_df[['p_unvax']])

midpoint_vax = covid_df[['p_vax']]
midpoint_unvax = covid_df[['p_unvax']]

num_age_groups = nrow(unique(covid_df['ages']))

conf_int_df = data.frame(age = factor(covid_df[['ages']], levels = c('12-39', '40-49', '50-59', '60-69'
           lower_bound = c(lower_bound_vax,lower_bound_unvax),
           midpoint = c(midpoint_vax, midpoint_unvax),
           upper_bound = c(upper_bound_vax,upper_bound_unvax),
           vax_status = factor(c(rep('Vaccinated', 7), rep('Unvaccinated', 7)), levels = c('Vaccinated'
```

Average expected length of the 95% modified-Wilson confidence intervals.

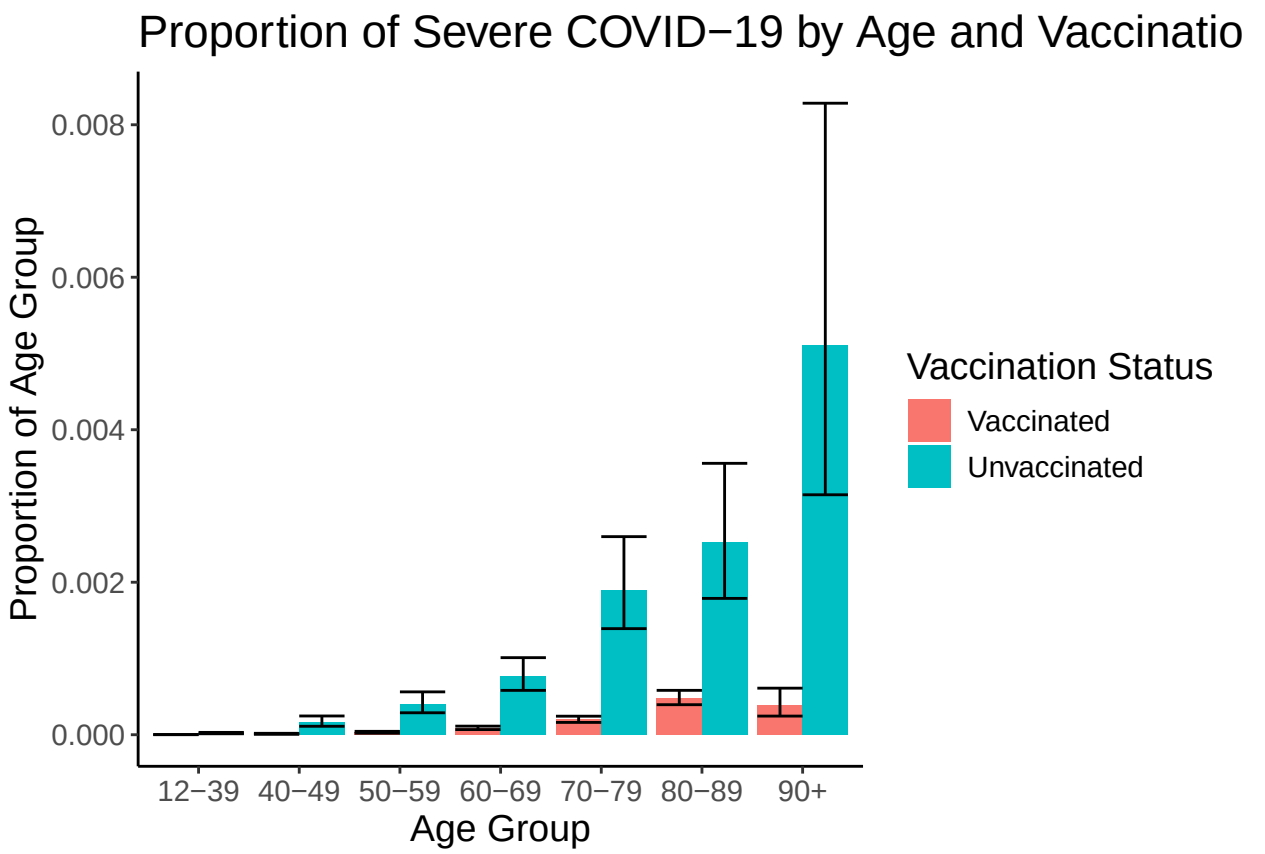
```
conf_int_df %>%
  mutate(length = upper_bound - lower_bound) %>%
  summarize(mean_length = mean(length))
```

```
##    mean_length
## 1 0.0006923002
```

Bar chart with point estimates and 95% modified-Wilson confidence intervals of severe COVID-19 cases by age and vaccination status.

```
conf_int_df %>%
  ggplot(aes(x = age, y = midpoint, fill = vax_status)) +
```

```
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = lower_bound, ymax = upper_bound), position = "dodge") +
  ggtitle("Proportion of Severe COVID-19 by Age and Vaccination Status") +
  xlab('Age Group') +
  ylab('Proportion of Age Group') +
  guides(fill=guide_legend(title="Vaccination Status")) +
  theme_classic() +
  theme(text = element_text(size = 14))
```



Question 2

Question 2 of the analysis asked for age specific vaccine efficacy and their confidence intervals.

Computation of age stratified vaccine efficacy confidence intervals. First separate data frames were created based on vaccination status. Then the 95% confidence intervals were calculated by using the bounds of the confidence intervals created in Question 1. The vaccine efficacy lower bound was created by subtracting the ratio of the upper bound of the vaccinated population to the lower bound of the unvaccinated population from 1. This specific ratio maximized the ratio of proportion of severe disease between vaccinated and unvaccinated populations, and as it was subtracted from 1 it minimized the difference, creating the lower bound. A similar procedure was used for the upper bound but instead the ratio was taken of the lower bound of the vaccinated population to the upper bound of the unvaccinated population, minimizing the ratio and maximizing the difference. This was computed for each individual age division. Then a data frame containing all the information was created. The formulas would look as so...

Confidence Intervals, Vaccinated then Unvaccinated

$$CI_V = (L_V, U_V)$$

$$CI_U = (L_U, U_U)$$

Defining Lower and Upper Bound Calculations for Vaccine Efficacy by Age Group

$$\text{VE}_{LowerBound} = 1 - \frac{U_v}{L_U}$$

$$\text{VE}_{UpperBound} = 1 - \frac{L_v}{U_U}$$

```
vaccine_conf_int = conf_int_df %>%
  filter(vax_status == "Vaccinated")

unvaccine_conf_int = conf_int_df %>%
  filter(vax_status == "Unvaccinated")

up_bound_eff = 1 - vaccine_conf_int[['lower_bound']] / unvaccine_conf_int[['upper_bound']]
low_bound_eff = 1 - vaccine_conf_int[['upper_bound']] / unvaccine_conf_int[['lower_bound']]
mid_point_eff = (low_bound_eff + up_bound_eff) / 2

conf_int_eff_df = data.frame(age = factor(covid_df[['ages']], levels = c('12-39', '40-49', '50-59', '60
                             lower_bound_eff = low_bound_eff,
                             midpoint = mid_point_eff,
                             upper_bound_eff = up_bound_eff)
```

Question 3

Question 3 of the analysis asked for a vaccine efficacy that described all groups at once; just a point estimate. This report calculated the unweighted overall vaccine efficacy, hence not taking into account the age-groups.

```
prop_covid_unvax = sum(covid_df['cases_unvax']) / sum(covid_df['pop_unvax'])
prop_covid_vax = sum(covid_df['cases_vax']) / sum(covid_df['pop_vax'])
vaccine_efficacy = 1 - (prop_covid_vax / prop_covid_unvax)
vaccine_efficacy
```

```
## [1] 0.6747618
```

Graph for age-specific vaccine efficacies versus the overall unweighted vaccine efficacy.

```
conf_int_eff_df %>%
  ggplot(aes(x = age, y = midpoint)) +
  geom_point() +
  geom_errorbar(aes(ymin = lower_bound_eff, ymax = upper_bound_eff)) +
  geom_hline(yintercept = vaccine_efficacy, linetype = 'dashed', color = 'red') +
  ggtitle("Overall and Age Group Vaccine Efficacy (Israel 2021)") +
  xlab('Age Group') +
  ylab('Vaccine Efficacy (%)') +
  theme_classic() +
  theme(text = element_text(size = 14))
```

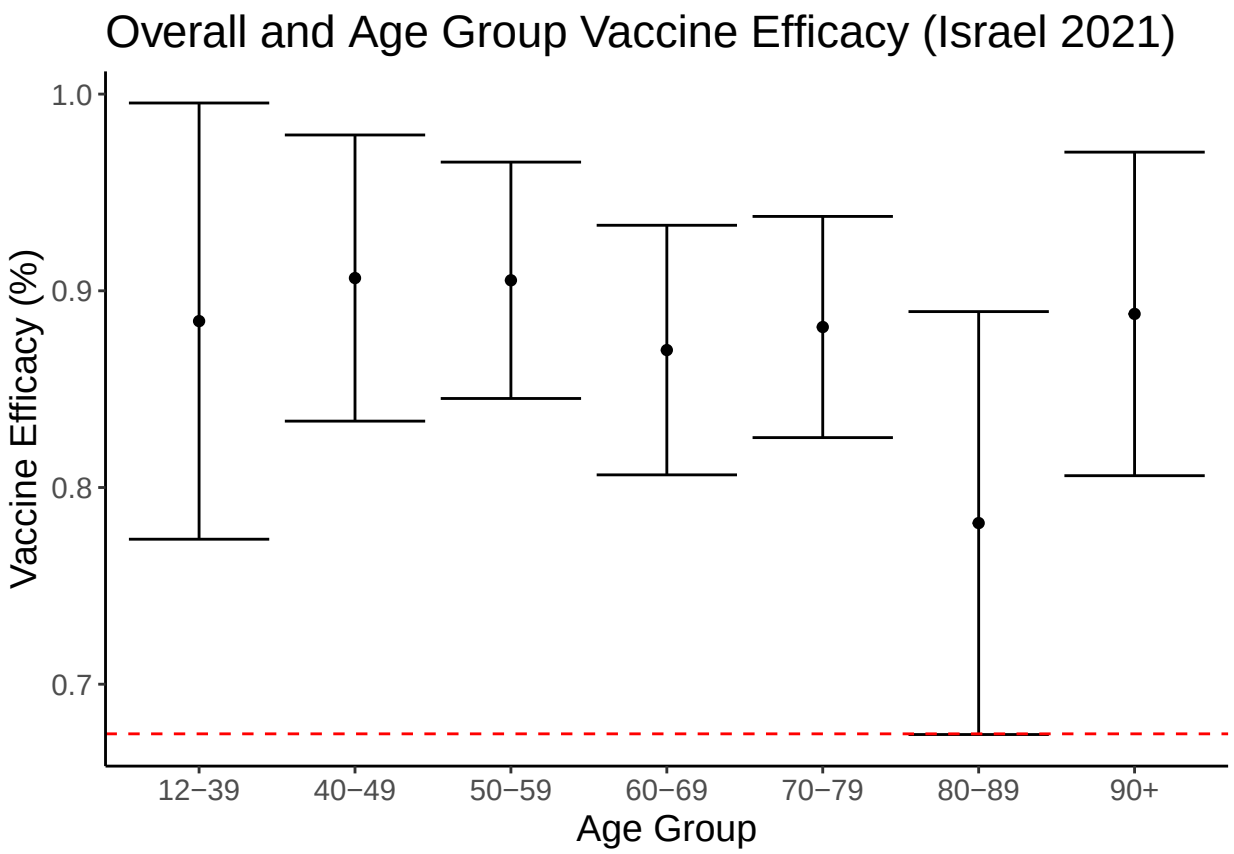


Table for age-specific vaccine efficacies versus the overall unweighted vaccine efficacy.

```
table_df = conf_int_eff_df %>%
  mutate(cross_overall = ifelse(lower_bound_eff < vaccine_efficacy, "Yes", "No"),
         at_least_90 = ifelse(upper_bound_eff > .9, "Yes", "No"),
         "Age Group" = age,
         "Lower Bound" = lower_bound_eff,
         "Midpoint" = midpoint,
         "Upper Bound" = upper_bound_eff,
         "CI Contains Population VE" = cross_overall,
         "CI Contains 90% VE" = at_least_90) %>%
  select(7:12)

knitr::kable(table_df,
             caption = "Vaccine Efficacy by Age Group",
             "simple",
             digits = 3)
```

Table 1: Vaccine Efficacy by Age Group

Age Group	Lower Bound	Midpoint	Upper Bound	CI Contains Population VE	CI Contains 90% VE
12-39	0.774	0.885	0.995	No	Yes
40-49	0.834	0.906	0.979	No	Yes
50-59	0.845	0.905	0.965	No	Yes
60-69	0.806	0.870	0.933	No	Yes
70-79	0.825	0.882	0.938	No	Yes
80-89	0.674	0.782	0.889	Yes	No

Age Group	Lower Bound	Midpoint	Upper Bound	CI Contains Population VE	CI Contains 90% VE
90+	0.806	0.888	0.970	No	Yes