



Using nest-map-unnest workflow in R for sub-group analysis in clinical trial reporting

Sheng-Wei Wang & Mia Qi

Presented at PHUSE US Connect 2022

May 3rd, 2022

Jennifer Jacobs, *Stowaway*

Jennifer is a New York-based artist
living with Type 1 diabetes.

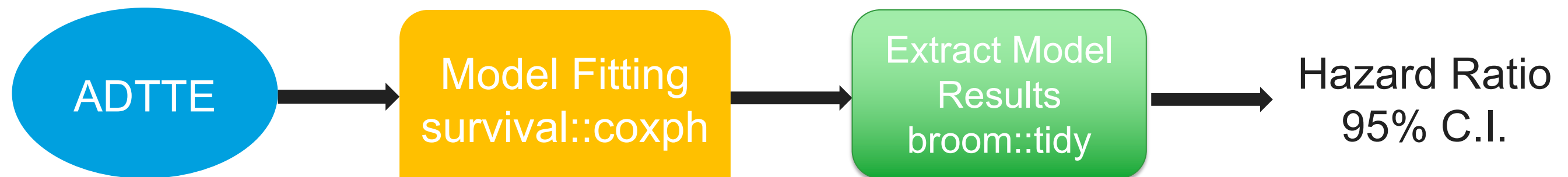
Outline

- Cox proportional hazard model fitting
- Sub-group analysis workflow
- Case study: CDISC pilot study
- Step-by-step demonstration of the nest-map-unnest workflow
- Sub-group analysis for multiple categories
- Display sub-group analysis results in a forest plot

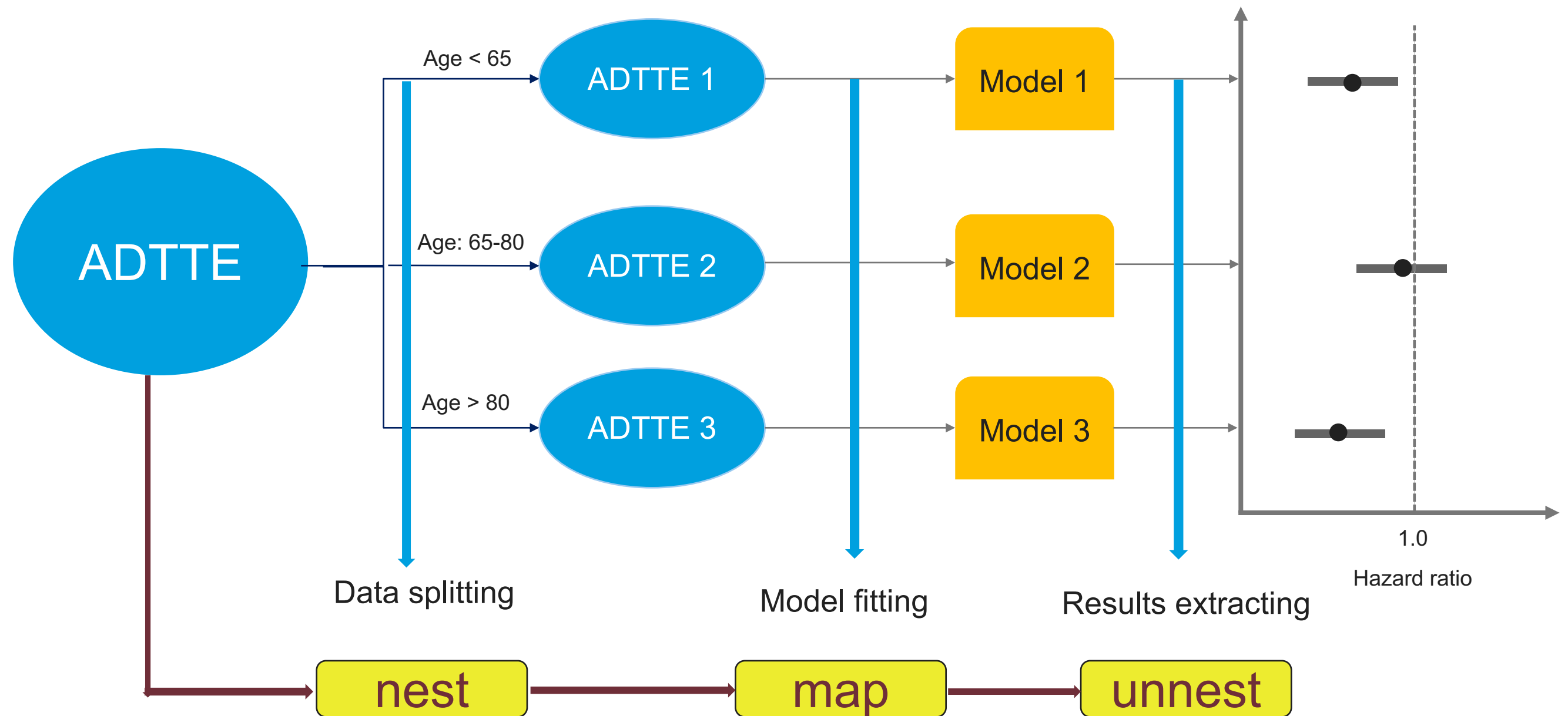
Cox Proportional Hazard Model Fitting

for the whole analysis population

- Cox proportional hazards model - semi-parametric model, commonly used to assess the relationship between the survival time and explanatory variables
- The `survival::coxph` function is used to fit the Cox proportional hazards model for time-to-event analysis dataset
- The `broom::tidy` function is used to extract key model results and convert to tidy data frame



Sub-group analysis workflow



Case Study: CDISC Pilot Study

- Analysis endpoint: time to dermatological events
- Analysis summary: hazard ratio (95% C.I.) of Xanomeline High Dose vs. Placebo for the safety population

```
# load libraries
library(haven)
library(tidyverse)
library(survival)
library(broom)

# read in analysis data for safety population, and exclude the low dose arm
adtte <- read_xpt(url("https://github.com/phuse-org/TestDataFactory/raw/main/Updated/TDF_ADaM/adtte.xpt")) %>%
  filter(SAFFL == "Y" & TRTA != "Xanomeline Low Dose") %>%
  # convert AGEGR1, SEX to factor
  mutate(AGEGR1 = fct_reorder(AGEGR1, AGEGR1N),
         SEX = factor(SEX, levels = c("M", "F")))

# fit Cox Proportional Hazard model
coxph(Surv(AVAL, 1 - CNSR) ~ TRTAN, data = adtte) %>%
  # apply tidy function to get hazard ratio and 95% C.I.
  tidy(exponentiate = TRUE, conf.int = TRUE, conf.level = 0.95) %>%
  select(estimate, conf.low, conf.high)

#> # A tibble: 1 × 3
#>   estimate conf.low conf.high
#>   <dbl>    <dbl>    <dbl>
#> 1     1.02     1.01     1.03
```

Sub-group analysis workflow Step 1: nest

- `tidyr::nest` - create a column that contains a list of data frames (list-column), where each row corresponds to one group of data defined by `dplyr::group_by`
- Split `adtte` by age group through nesting; each row of the data column contains the sub-group data for each age group; the grouping variable (`AGEGR1`) is kept outside of the data list-column

```
# split adtte by age group through nesting
```

```
adtte_nest <- adtte %>%
```

```
  group_by(AGEGR1) %>%
```

```
  nest()
```

```
adtte_nest
```

```
#> # A tibble: 3 × 2
```

```
#> # Groups:   AGEGR1 [3]
```

```
#>   AGEGR1 data
```

```
#>   <fct>   <list>
```

```
#> 1 <65     <tibble [25 × 25]>
```

```
#> 2 65-80   <tibble [97 × 25]>
```

```
#> 3 >80     <tibble [48 × 25]>
```

```
adtte_nest$data[[1]]
```

```
#> # A tibble: 25 × 25
```

```
#>   STUDYID SITEID USUBJID   AGE AGEGR1N RACE  RACEN SEX  TRTSDT   TRTEDT
```

```
#>   <chr>   <chr>   <chr>   <dbl>   <dbl> <chr> <dbl> <chr> <date>   <date>
```

```
#> 1 CDISCPI~ 701    01-701~   63         1 WHITE    1 F    2014-01-02 2014-07-02
```

```
#> 2 CDISCPI~ 701    01-701~   64         1 WHITE    1 M    2012-08-05 2012-09-01
```

```
#> 3 CDISCPI~ 701    01-701~   52         1 WHITE    1 M    2014-03-12 2014-09-09
```

```
#> 4 CDISCPI~ 701    01-701~   57         1 WHITE    1 M    2013-08-23 2014-02-20
```

```
#> 5 CDISCPI~ 701    01-701~   56         1 WHITE    1 M    2013-02-12 2013-03-18
```

```
#> 6 CDISCPI~ 701    01-701~   56         1 WHITE    1 M    2014-01-11 2014-07-10
```

```
#> 7 CDISCPI~ 701    01-701~   61         1 AMER~    6 M    2014-02-07 2014-05-31
```

```
#> 8 CDISCPI~ 701    01-701~   56         1 WHITE    1 F    2014-01-25 2014-07-26
```

```
#> 9 CDISCPI~ 701    01-701~   61         1 WHITE    1 M    2013-08-29 2013-11-05
```

```
#> 10 CDISCPI~ 701    01-701~   63         1 WHITE    1 F    2013-10-08 2014-03-18
```

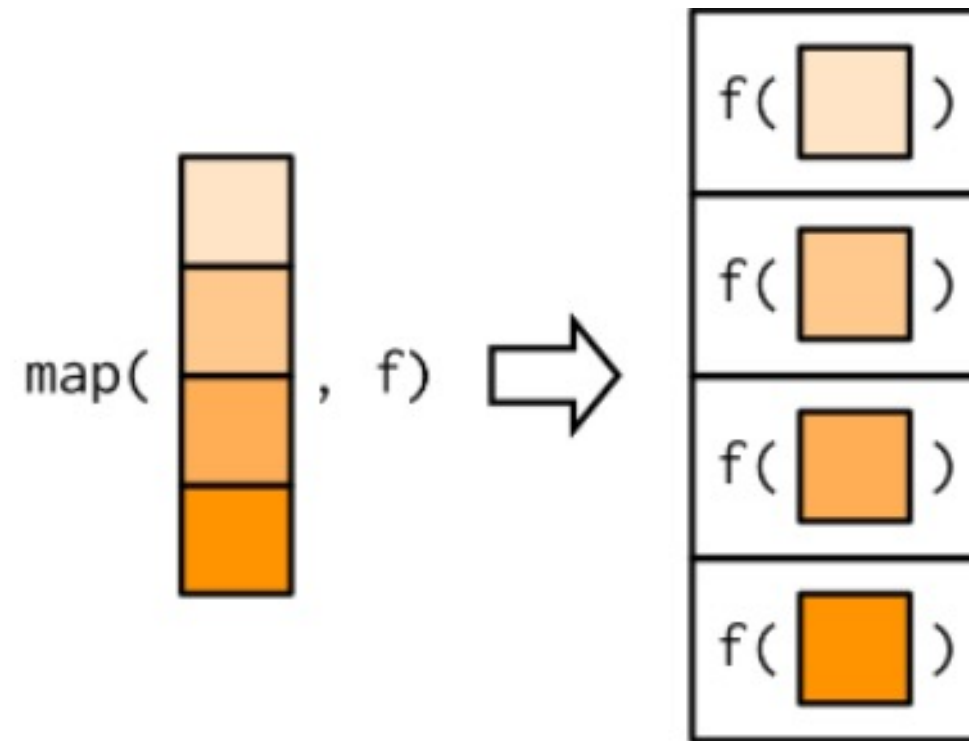
```
#> # ... with 15 more rows, and 15 more variables: TRTDUR <dbl>, TRTP <chr>,
```

```
#> #   TRTA <chr>, TRTAN <dbl>, PARAM <chr>, PARAMCD <chr>, AVAL <dbl>,
```

```
#> #   STARTDT <date>, ADT <date>, CNSR <dbl>, EVNTDESC <chr>, SRCDOM <chr>,
```

Sub-group analysis workflow Step 2: map

- Repeat Cox proportional hazard model fitting for each sub-group data
 - Nesting create a list of data frames
 - A list of Cox models
- `purrr::map` – provides vectorized operation for model fitting to create a list of models



Sub-group analysis workflow Step 2: map

- Define the model fitting function of the Cox model to be used in the map function
- Use the map function inside the mutate function to create a list of models

```
# define model fitting function
fit_coxph <- function(df){
  coxph(Surv(AVAL, 1 - CNSR) ~ TRTAN, data = df) %>%
    # apply tidy function to get hazard ratio and 95% C.I.
    tidy(exponentiate = TRUE, conf.int = TRUE, conf.level = 0.95) %>%
    select(estimate, conf.low, conf.high)
}

# create a list of models through mapping
adtte_nest <- adtte %>%
  group_by(AGEGR1) %>%
  nest() %>%
  mutate(model = map(data, ~fit_coxph(.x)))
```

```
adtte_nest
#> # A tibble: 3 × 3
#> # Groups:   AGEGR1 [3]
#>   AGEGR1 data          model
#>   <fct> <list>         <list>
#> 1 <65   <tibble [25 × 25]> <tibble [1 × 3]>
#> 2 65-80 <tibble [97 × 25]> <tibble [1 × 3]>
#> 3 >80   <tibble [48 × 25]> <tibble [1 × 3]>
```

Sub-group analysis result for the <65 age group

```
adtte_nest$model[[1]]
#> # A tibble: 1 × 3
#>   estimate conf.low conf.high
#>   <dbl>    <dbl>    <dbl>
#> 1    1.02    1.01    1.04
```

Sub-group analysis workflow Step 3: unnest

- `tidyr::unnest` – flatten a list-column back out into regular columns
- Cox model results (estimate, conf.low, conf.high) can be obtained through unnesting the model list-column

```
# unnest model list-column
adtte_nest <- adtte %>%
  group_by(AGEGR1) %>%
  nest() %>%
  mutate(model = map(data, ~fit_coxph(.x))) %>%
  unnest(model)

adtte_nest
#> # A tibble: 3 × 5
#> # Groups:   AGEGR1 [3]
#>   AGEGR1 data                estimate conf.low conf.high
#>   <fct>  <list>              <dbl>    <dbl>    <dbl>
#> 1 <65    <tibble [25 × 25]>         1.02     1.01     1.04
#> 2 65-80  <tibble [97 × 25]>         1.02     1.01     1.03
#> 3 >80    <tibble [48 × 25]>         1.01     1.00     1.02
```

Sub-group Analysis for Multiple Categories

- Create a sub-group function with the nest-map-unnest workflow for one sub-group category
- Use `purrr::map2_df` to iterate the sub-group function over the lists of sub-group categories (age group, gender, etc.)

```
# define sub-group function
call_subgrp <- function(df, sgrp, label){
  df %>%
    group_by(.data[[sgrp]]) %>%
    nest() %>%
    mutate(model = map(data, ~fit_coxph(.x))) %>%
    unnest(model) %>%
    ungroup() %>%
    # rename sub-group column to Grp
    rename(Grp = .data[[sgrp]]) %>%
    select(-data) %>%
    # add label for sub-group category
    add_row(Grp = label, .before = 1)
}

# define list of sub-group variables
subgrp_list <- c("AGEGR1", "SEX")
# define sub-group labels
subgrp_label <- c("Age Group", "Gender")
```

```
# map over 2 sets of inputs simultaneously
tbl <- map2_df(subgrp_list, subgrp_label,
              ~call_subgrp(adtte, sgrp = .x, label = .y))

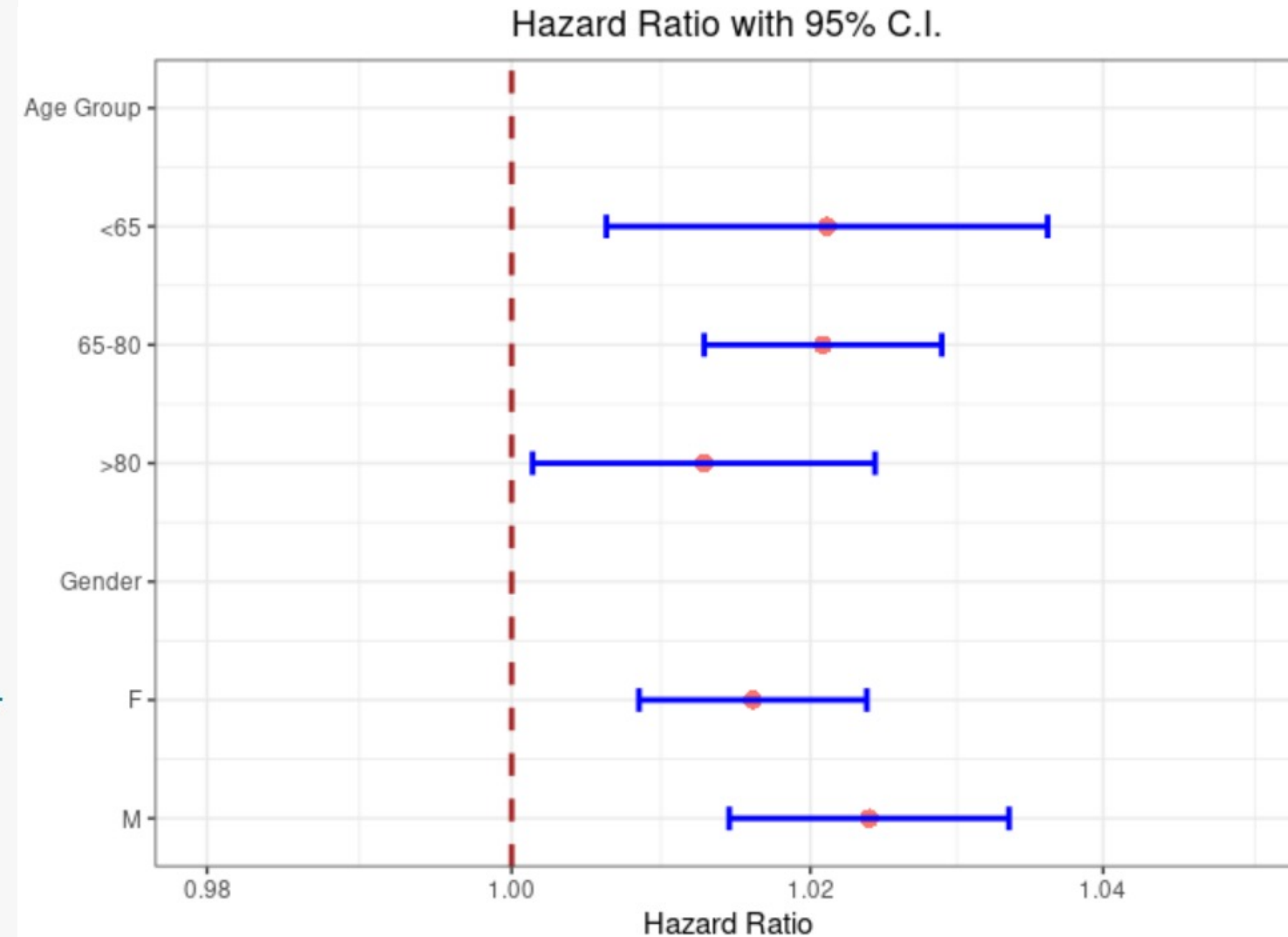
tbl
#> # A tibble: 7 × 4
#>   Grp      estimate conf.low conf.high
#>   <chr>      <dbl>    <dbl>    <dbl>
#> 1 Age Group      NA        NA        NA
#> 2 <65           1.02      1.01      1.04
#> 3 65-80         1.02      1.01      1.03
#> 4 >80           1.01      1.00      1.02
#> 5 Gender        NA        NA        NA
#> 6 F             1.02      1.01      1.02
#> 7 M             1.02      1.01      1.03
```

Display Sub-group Analysis Results in Forest Plot

- The multiple sub-group analysis outputs can be directly used to create forest plot

```
# create sub-group label vector for y-axis
ytxt <- tbl %>% pull(Grp)

# create forest plot of hazard ratio with 95% C.I.
tbl %>%
  # create ord number for proper ordering
  rownames_to_column(var = "ord") %>%
  mutate(ord = as.numeric(ord)) %>%
  # start plotting
  ggplot(aes(x = estimate, y = ord)) +
  # plot hazard ratio points
  geom_point(aes(shape = "circle filled", color="red"), size=3) +
  # plot 95% C.I.
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high),
                 color="blue", size=1, height = .2) +
  # plot the vertical line of hr = 1.0
  geom_vline(xintercept = 1, color="brown", size=1, linetype='dashed') +
  scale_x_continuous(limits = c(0.98,1.05), trans = "log10") +
  scale_y_continuous(breaks = c(1:nrow(tbl)),
                     labels = ytxt, trans = "reverse") +
  # define title and axis labels
  labs(title = "Hazard Ratio with 95% C.I.",
       x = "Hazard Ratio", y = "") +
  theme_bw() + # define plot theme
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5))
```



Q&A

Viral exacerbation at 40x magnification