R Notebook

Import mock database

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## Warning in loadNamespace(j <- i[[1L]], c(lib.loc, .libPaths()), versionCheck =
## vI[[j]]): package 'utf8' has no 'package.rds' in Meta/
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'purrr' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## Warning: package 'stringr' was built under R version 4.1.3
```

```
## Warning: package 'forcats' was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dbplyr)
```

```
## Warning: package 'dbplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dbplyr'
##
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
con <- DBI::dbConnect(RSQLite::SQLite(), dbname = 'simulation/experiments/mock_results.db')
agents_db <- tbl(con, 'agents')
networks_db <- tbl(con, 'networks')
parameters_db <- tbl(con, 'parameters')
```

Tag different stages of the simulation

```
networks_db <- networks_db %>%
  left_join(
    y = parameters_db %>%
      select(sim_id, intv_class_name),
    by = 'sim_id') %>%
  mutate(
    phase = case_when(
      tick < 300 ~ 'burn_in',
      tick %in% 300:349 ~ 'baseline',
      tick %in% 350:359 ~ 'active',
      tick %in% 360:409 ~ 'post',
      tick %in% 410:470 ~ 'followup'),
    group_size = case_when(
      n_agents < 15 ~ '04-14',
      n_agents %in% 15:24 ~ '15-24',
      n_agents %in% 25:35 ~ '25-35'))
```

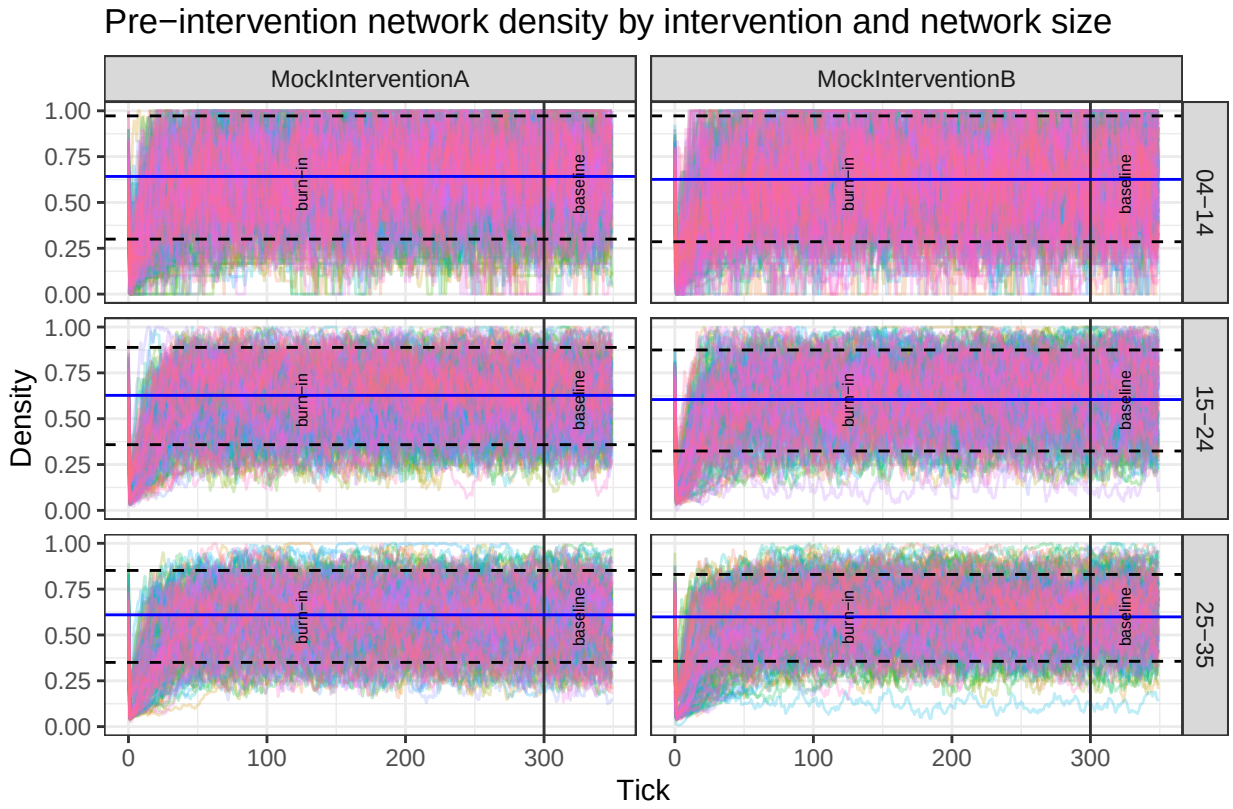
```
agents_db <- agents_db %>%
  left_join(
    y = parameters_db %>%
      select(sim_id, intv_class_name),
    by = 'sim_id') %>%
  left_join(
    y = networks_db %>%
      select(sim_id, tick, n_agents),
    by = c('sim_id', 'tick')) %>%
  mutate(
    phase = case_when(
      tick < 250 ~ 'burn_in',
      tick %in% 250:349 ~ 'baseline',
      tick %in% 350:359 ~ 'active',
      tick %in% 360:409 ~ 'post',
      tick %in% 410:470 ~ 'followup'),
    group_size = case_when(
      n_agents < 15 ~ '04-14',
      n_agents %in% 15:24 ~ '15-24',
      n_agents %in% 25:35 ~ '25-35'))
```

Check pre-intervention density

```
quantile_df <- networks_db %>%
  filter(tick > 250) %>%
  as_tibble() %>%
  group_by(intv_class_name, group_size) %>%
  summarise(
    q10 = quantile(density, probs = .10),
    mean = mean(density, na.rm = T),
    q90 = quantile(density, probs = .90))
```

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

```
plot_df <- networks_db %>%
  filter(tick < 350)

ggplot(plot_df, aes(x = tick, y = density, color = factor(sim_id))) +
  facet_grid(group_size ~ intv_class_name) +
  geom_line(alpha = .25) +
  geom_vline(xintercept = 300, linetype = 1, color = 'grey20') +
  # geom_vline(xintercept = 350, linetype = 1, color = 'grey20') +
  # geom_vline(xintercept = 360, linetype = 1, color = 'grey20') +
  # geom_vline(xintercept = 410, linetype = 1, color = 'grey20') +
  geom_hline(data = quantile_df, mapping = aes(yintercept = q10), linetype = 2) +
  geom_hline(data = quantile_df, mapping = aes(yintercept = q90), linetype = 2) +
  geom_hline(data = quantile_df, mapping = aes(yintercept = mean), color = 'blue') +
  annotate('text', x = 125, y = .60, label = 'burn-in', angle = 90, size = 2) +
  annotate('text', x = 325, y = .60, label = 'baseline', angle = 90, size = 2) +
  # annotate('text', x = 355, y = .60, label = 'active tx', angle = 90, size = 2) +
  # annotate('text', x = 385, y = .60, label = 'post tx', angle = 90, size = 2) +
  # annotate('text', x = 460, y = .60, label = 'follow-up', angle = 90, size = 2) +
  labs(
    title = 'Pre-intervention network density by intervention and network size',
    caption = 'Dashed lines represent inner 80% of scores',
    y = 'Density',
    x = 'Tick'
    ) +
  theme_bw() +
  theme(legend.position = 'none')
```



Plot change in risk levels

```
risk_plot_df <- agents_db %>%
  filter(phase %in% c('baseline', 'active', 'post', 'followup')) %>%
  group_by(sim_id, phase, intv_class_name) %>%
  summarise(cur_risk = mean(50*cur_risk, na.rm = T)) %>%
  as_tibble() %>%
  mutate(
    phase = factor(
      x = phase,
      levels = c('baseline', 'active', 'post', 'followup'),
      ordered = T))
```

```
## `summarise()` has grouped output by "sim_id" and "phase". You can override
## using the `.groups` argument.
```

```
risk_plot_df
```

```
## # A tibble: 4,000 x 4
##    sim_id phase    intv_class_name    cur_risk
##     <int> <ord>    <chr>                 <dbl>
##  1      0 active   MockInterventionB    0.0222
```

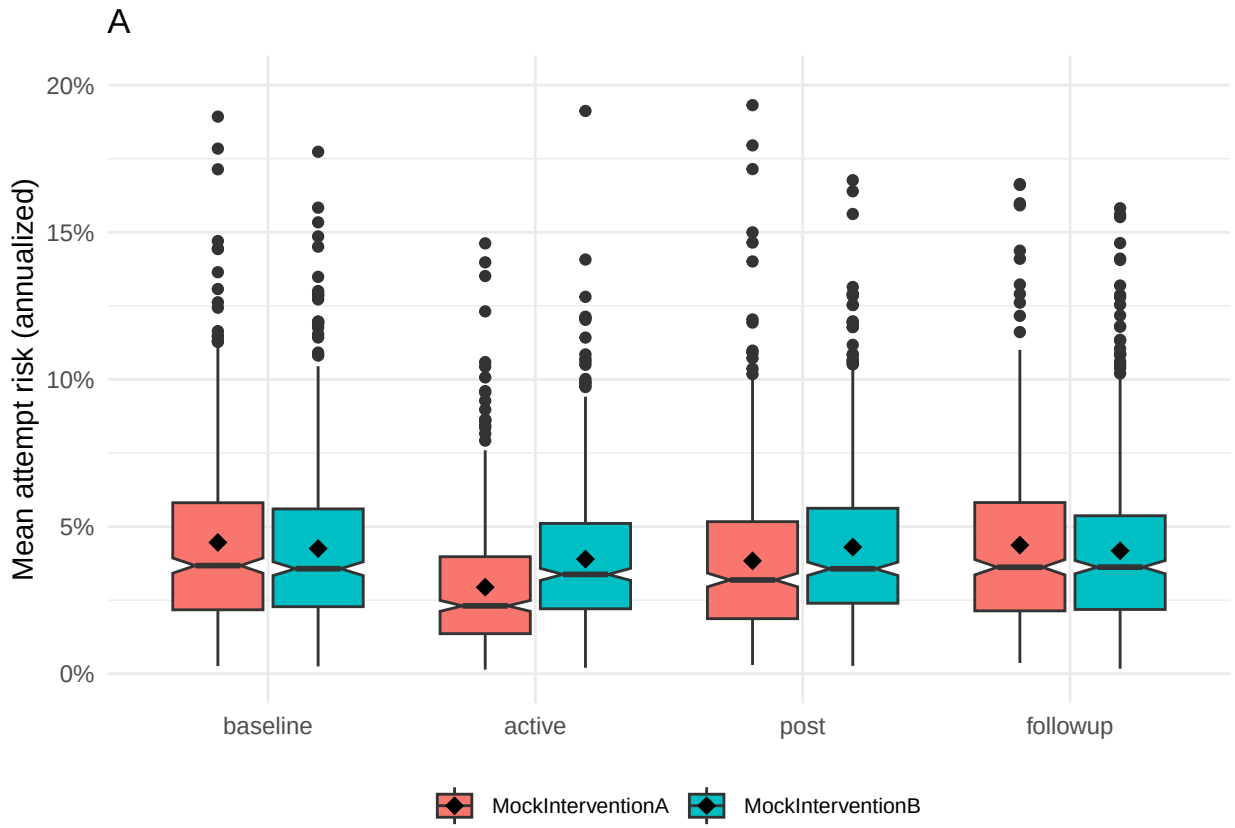
```
##  2      0 baseline MockInterventionB   0.0242
##  3      0 followup MockInterventionB   0.0216
##  4      0 post     MockInterventionB   0.0270
##  5      1 active   MockInterventionA   0.0141
##  6      1 baseline MockInterventionA   0.0436
##  7      1 followup MockInterventionA   0.0505
##  8      1 post     MockInterventionA   0.0403
##  9      2 active   MockInterventionA   0.0148
## 10      2 baseline MockInterventionA   0.0196
## # ... with 3,990 more rows
```

```
risk_plot_means <- risk_plot_df %>%
  group_by(phase, intv_class_name) %>%
  summarise(cur_risk = mean(cur_risk))
```

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

```
risk_boxplot <- ggplot(
    data = risk_plot_df,
    mapping = aes(x = phase, y = cur_risk, fill = intv_class_name)) +
  geom_boxplot(notch = T) +
  geom_point(
    data = risk_plot_means,
    position = position_dodge2(width = .750),
    color = 'black',
    shape = 18,
    size = 3) +
  scale_y_continuous(
    limits = c(0, .20),
    labels = function(y) paste0(100*y, '%')) +
  labs(
    title = 'A',
    x = 'Simulation Phase',
    y = 'Mean attempt risk (annualized)'
  ) +
  theme_minimal() +
  theme(
    legend.position = 'bottom',
    axis.title.x = element_blank(),
    legend.title = element_blank(),
    legend.text = element_text(size=8)
  )

risk_boxplot
```

```
## Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).
```




```
risk_improvement_df <- risk_plot_df %>%
  group_by(sim_id) %>%
  mutate(
    baseline_risk = cur_risk[which(phase == 'baseline')],
    rel_change = cur_risk/baseline_risk - 1)

risk_improvement_means <- risk_improvement_df %>%
  group_by(phase, intv_class_name) %>%
  summarise(rel_change = mean(rel_change, na.rm = T)) %>%
  ungroup() %>%
  mutate(
    change_label = ifelse(phase != 'baseline', paste0(round(100*rel_change), '%'), NA),
    change_label = ifelse(rel_change > 0, paste0('+', change_label), change_label))
```

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

```
risk_imp_plot <- ggplot(
  data = risk_improvement_df,
  mapping = aes(
    x = as.numeric(phase),
    y = rel_change,
    color = intv_class_name)) +
  geom_line(alpha = .05, mapping = aes(group = sim_id)) +
  # geom_jitter(alpha = .05, mapping = aes(group = sim_id), position = position_dodge2(width = .30)) +
```

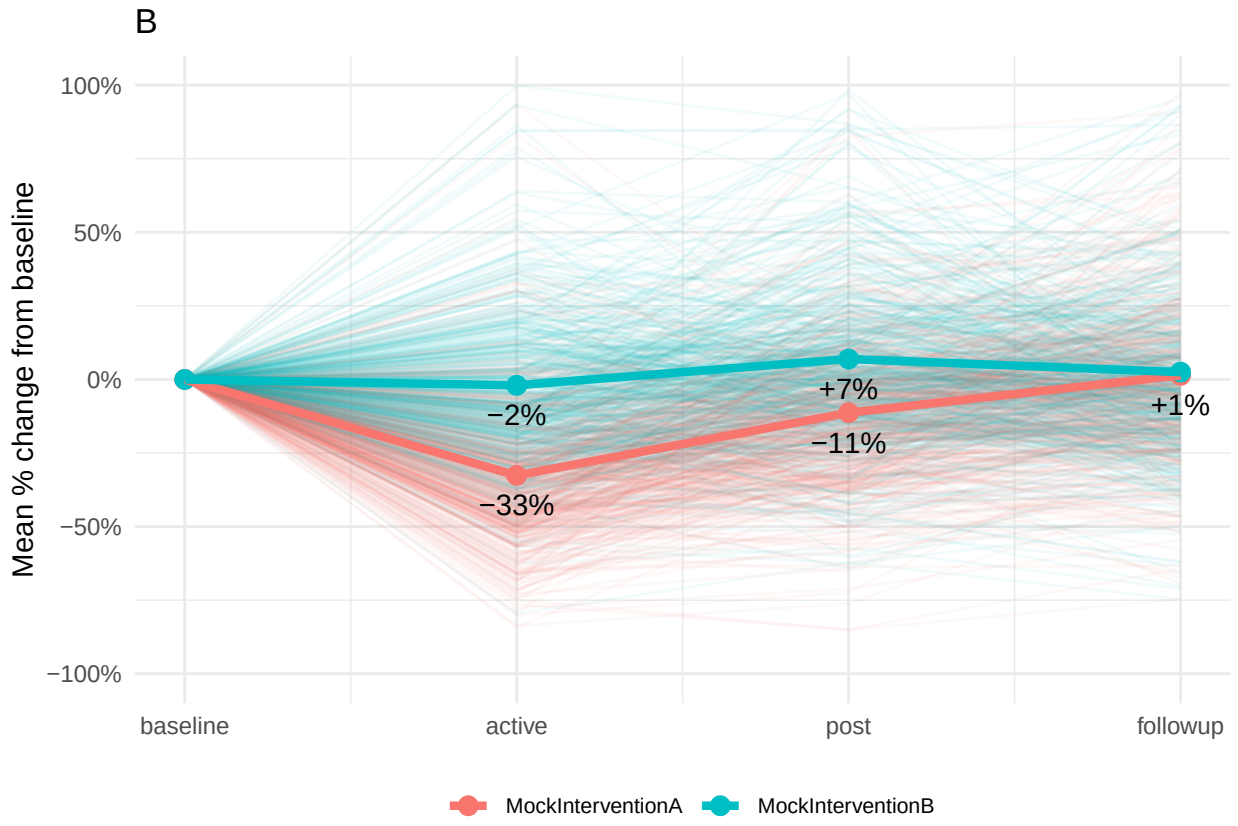
```
  geom_point(data = risk_improvement_means, size = 3) +
  geom_line(data = risk_improvement_means, size = 1.5) +
  geom_text(
    data = risk_improvement_means,
    mapping = aes(label = change_label),
    nudge_y = -.1,
    check_overlap = T,
    color = 'black') +
  scale_x_continuous(breaks = 1:4, labels = c('baseline', 'active', 'post', 'followup')) +
  scale_y_continuous(
    limits = c(-1, 1),
    labels = function(y) paste0(100*y, '%')) +
  labs(
    title = 'B',
    x = 'Simulation Phase',
    y = 'Mean % change from baseline') +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    legend.position = 'bottom',
    legend.title = element_blank(),
    legend.text = element_text(size=8)
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```

```
risk_imp_plot
```

```
## Warning: Removed 13 rows containing missing values (‘geom_line()‘).
```

```
## Warning: Removed 2 rows containing missing values (‘geom_text()‘).
```

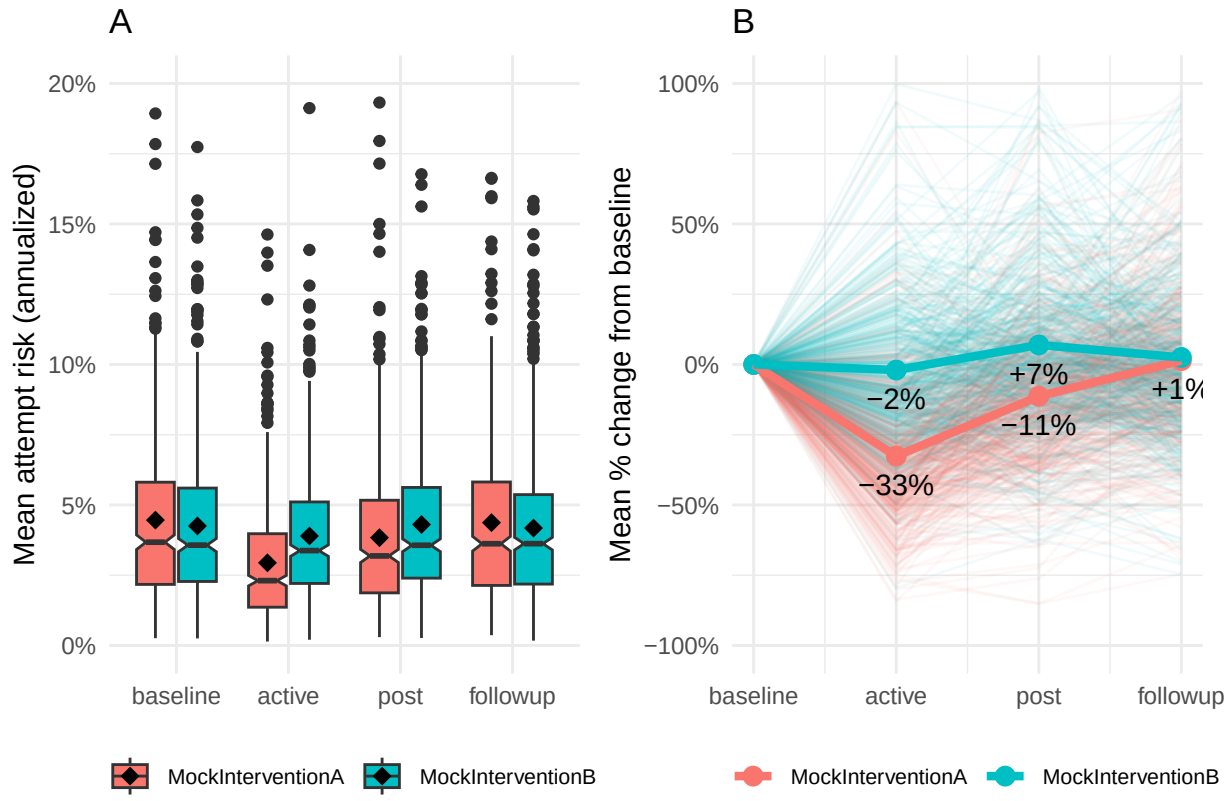



```
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.1.3
```

```
risk_plot <- (risk_boxplot + risk_imp_plot)

risk_plot
```

```
## Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 13 rows containing missing values (`geom_line()`).

## Warning: Removed 2 rows containing missing values (`geom_text()`).
```



Plot attempts observed in each condition

```
networks_db %>%
  group_by(sim_id) %>%
  filter(tick == 0) %>%
  ungroup() %>%
  summarise(n_agents = sum(n_agents))
```

```
## Warning: Missing values are always removed in SQL aggregation functions.
## Use 'na.rm = TRUE' to silence this warning
## This warning is displayed once every 8 hours.
```

```
## # Source:   SQL [1 x 1]
## # Database: sqlite 3.39.4 [C:\Users\icero\Box\code_repo_test\project_repo\nhiabm_v2\simulation\experi
##   n_agents
##      <int>
## 1    19502
```

```
attempt_count1 <- agents_db %>%
  filter(phase %in% c('baseline', 'active', 'post', 'followup')) %>%
  group_by(phase, sim_id, intv_class_name) %>%
  summarise(observed_attempts = sum(cur_attempt)) %>%
```

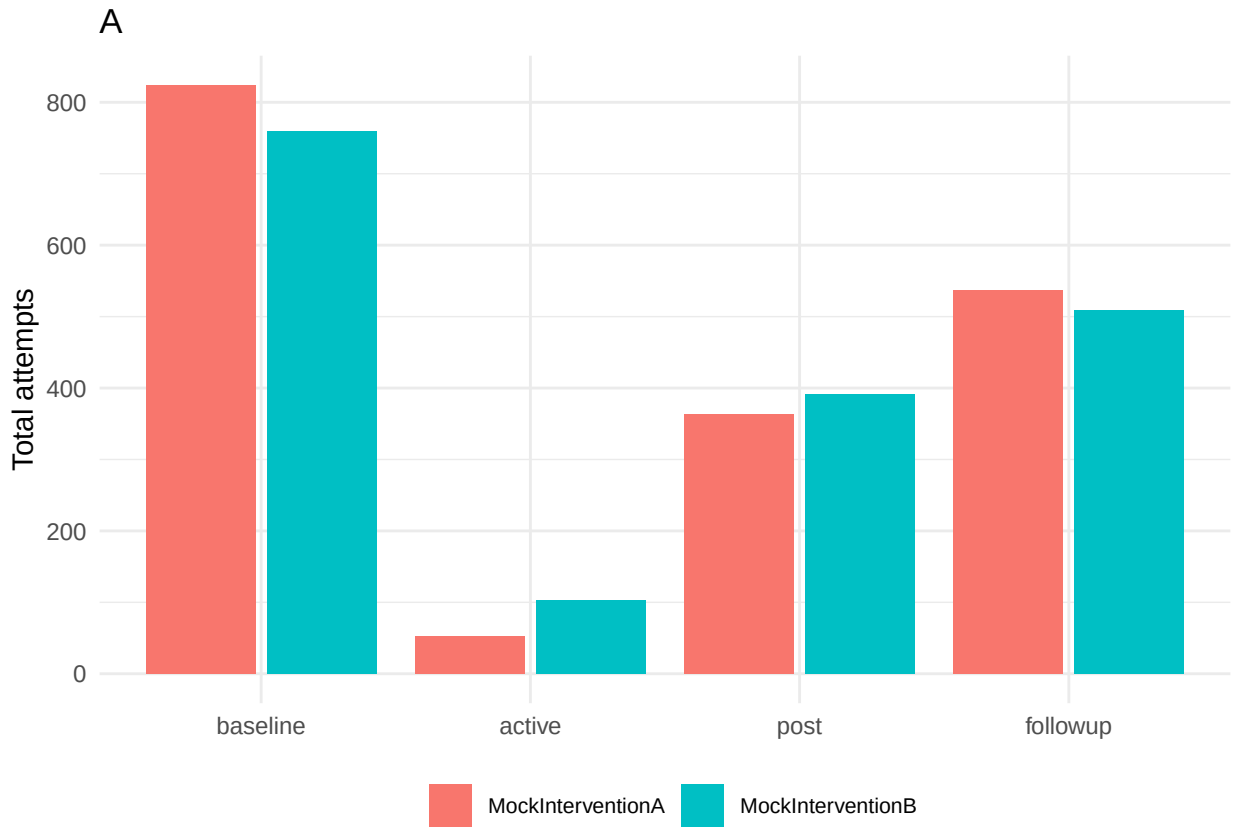
```
  as_tibble() %>%
  mutate(
    phase = factor(
      x = phase,
      levels = c('baseline', 'active', 'post', 'followup'),
      ordered = T))
```

```
## `summarise()` has grouped output by "phase" and "sim_id". You can override
## using the `.groups` argument.
```

```
attempt_count1
```

```
## # A tibble: 4,000 x 4
##    phase  sim_id intv_class_name   observed_attempts
##    <ord>   <int> <chr>                         <int>
##  1 active      0 MockInterventionB                 0
##  2 active      1 MockInterventionA                 0
##  3 active      2 MockInterventionA                 0
##  4 active      3 MockInterventionB                 0
##  5 active      4 MockInterventionB                 0
##  6 active      5 MockInterventionA                 0
##  7 active      6 MockInterventionB                 0
##  8 active      7 MockInterventionB                 0
##  9 active      8 MockInterventionB                 0
## 10 active      9 MockInterventionB                 0
## # ... with 3,990 more rows
```

```
attempt_count_plot <- ggplot(
    data = attempt_count1,
    mapping = aes(x = phase, y = observed_attempts, fill = intv_class_name)) +
  geom_bar(stat = 'summary', fun = 'sum', position = position_dodge2()) +
  labs(
    title = 'A',
    y = 'Total attempts') +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    legend.position = 'bottom',
    legend.title = element_blank(),
    legend.text = element_text(size=8))

attempt_count_plot
```



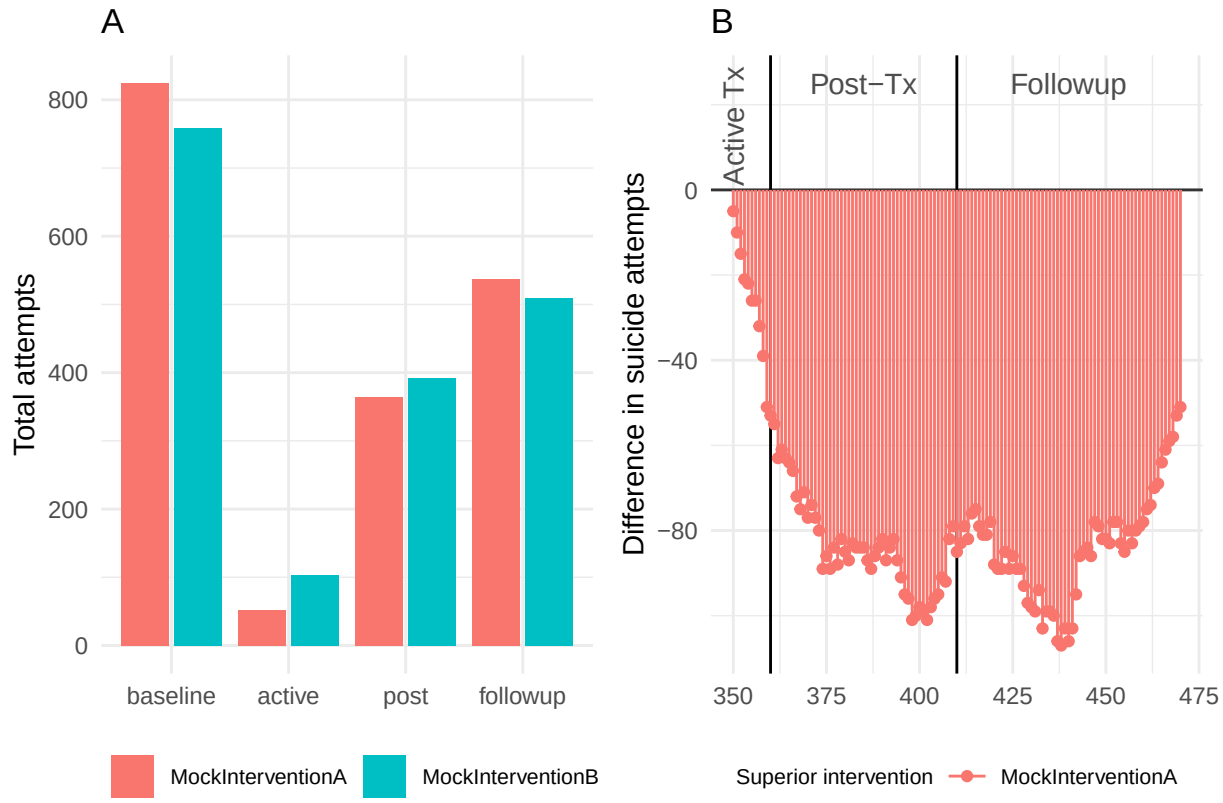

```
attempt_count2 <- agents_db %>%
  filter(phase %in% c('active', 'post', 'followup')) %>%
  group_by(intv_class_name, tick) %>%
  summarise(attempts = sum(cur_attempt)) %>%
  ungroup() %>%
  as_tibble() %>%
  group_by(intv_class_name) %>%
  arrange(tick) %>%
  mutate(cum_attempts = cumsum(attempts)) %>%
  ungroup() %>%
  pivot_wider(
    id_cols = tick,
    names_from = intv_class_name,
    values_from = cum_attempts) %>%
  mutate(a_over_b = MockInterventionA - MockInterventionB)
```

```
## `summarise()` has grouped output by "intv_class_name". You can override using
## the `.groups` argument.
```

```
attempt_count3 <- attempt_count2 %>%
  mutate(
    winning = case_when(
      tick < 350 ~ NA_character_,
      a_over_b < 0 ~ 'MockInterventionA',
      TRUE ~ 'MockInterventionB'))
```

```
cum_advantage_plot <- ggplot(attempt_count3, aes(x = tick, y = a_over_b, color = winning)) +
  geom_hline(yintercept = 0, color = 'grey20') +
  annotate(geom = 'text', x = 350, y = 15, label = 'Active Tx', color = 'grey30', angle = 90) +
  geom_vline(xintercept = 360) +
  annotate(geom = 'text', x = 385, y = 25, label = 'Post-Tx', color = 'grey30') +

  geom_vline(xintercept = 410) +
  annotate(geom = 'text', x = 440, y = 25, label = 'Followup', color = 'grey30') +
  geom_point() +
  geom_segment(
    mapping = aes(xend=tick, y = 0, yend=a_over_b, color=winning)) +

  labs(
    title = 'B',
    color = 'Superior intervention',
    y = 'Difference in suicide attempts') +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    legend.position = 'bottom',
    legend.title = element_text(size = 8),
    legend.text = element_text(size=8))


cum_advantage_plot
```

```
attempt_plot <- attempt_count_plot + cum_advantage_plot

attempt_plot
```




```
DBI::dbDisconnect(con)
```