

Main analysis v1

A computational model of network health interventions for suicide

Ian Cero & Peter Wyman

2023-01-04

This study compares simulated versions of network health interventions (NHIs) for suicide to traditional individual-focused interventions (INDs). The simulations are designed so that each simulated social network runs for roughly one-half of its total run time to establish a baseline for that network, after which an NHI or IND is introduced briefly (i.e., 1 to 10 simulation-weeks/ticks) and results are then tracked for the second half of the simulation time. We make several predictions about the performance of NHIs in our simulations, which are organized into sections below. Each section includes a draft analysis that was performed on mock simulation data (i.e., data produced from simulations of two “mock” interventions).

Import main database

We begin by importing the simulated data.

```
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 tidyrr   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 'tidyrr' 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

library(patchwork)

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

con <- DBI::dbConnect(
  drv = RSQLite::SQLite(),
  dbname = 'simulation/experiments/main_results_2023-01-04.db')
agents_db <- tbl(con, 'agents')
networks_db <- tbl(con, 'networks')
parameters_db <- tbl(con, 'parameters')

```

We then tag the different phases of each simulation for subsequent analysis.

```

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'))

```

Planned simulation checks

Ensure that network densities are stable by the end of burn-in period.

```

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)

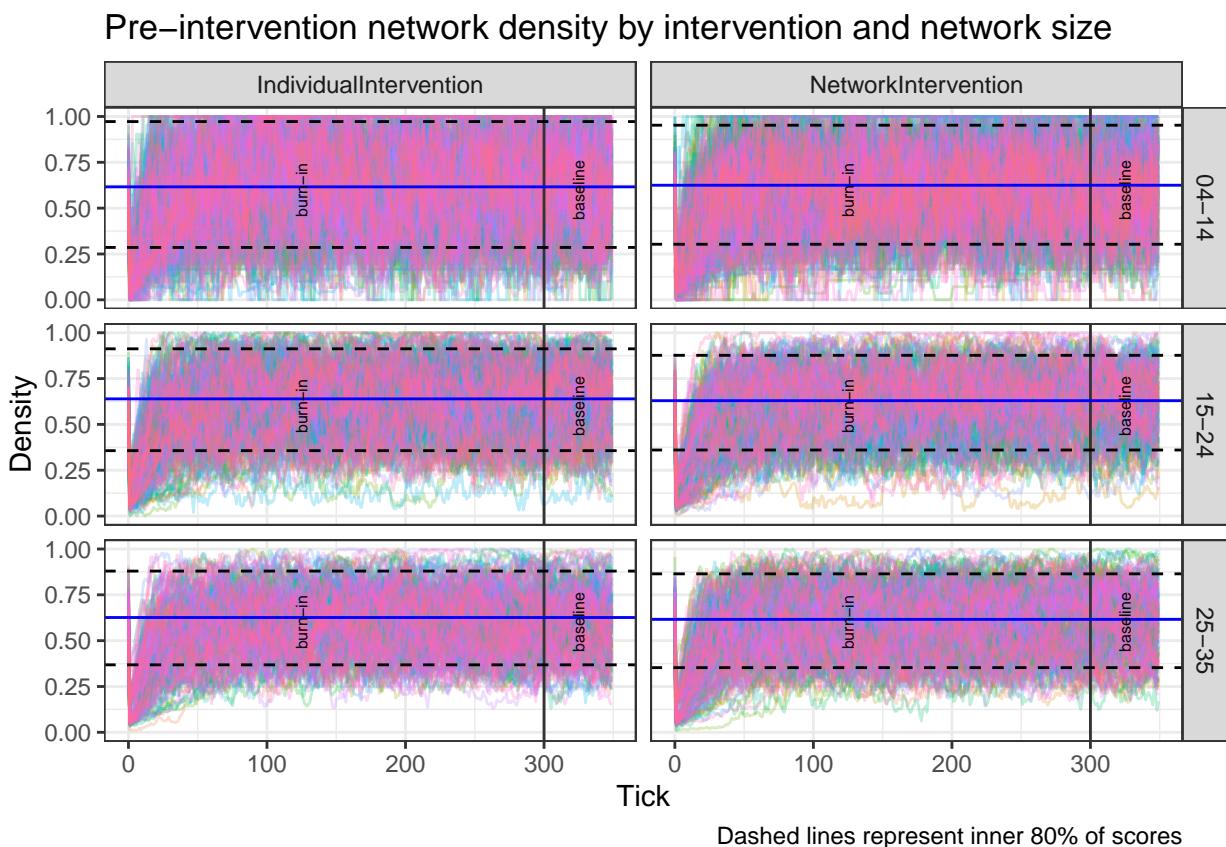
ggplot(plot_df, aes(x = tick, y = density, color = factor(sim_id))) +
  facet_grid(group_size ~ intv_class_name) +
  geom_line(alpha = .25) +
  geom_vline(xintercept = 300, linetype = 1, color = 'grey20') +
  # geom_vline(xintercept = 350, linetype = 1, color = 'grey20') +
  # geom_vline(xintercept = 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')

```



Planned analyses

What follows are drafts of the analyses which are planned for the primary interventions under consideration, which are conducted on mock interventions. This was done to remove as many bugs and ad hoc decision points before the real interventions are ever simulated.

1. Group-level suicide prevention effectiveness:

The following are the primary hypotheses of the study. They will be analyzed from multiple perspectives, which are sketched below on mock simulation data.

- a. Both IND- and NHI- treated groups will produce reductions in (i) average suicide risk level and (ii) reductions in the rate of observed suicide attempts, compared to baseline.
- b. During active intervention time, IND-treated groups will have (i) lower average suicide risk levels and (ii) greater reductions in the rate of observed suicide attempts than NHI-treated groups.
- c. However, in the simulation-year after intervention has concluded, NHI-treated groups will have (i) lower average suicide risk levels and (ii) greater reductions in the rate of observed suicide attempts than IND-treated groups.
- d. Both IND-treated groups and NHI-treated groups will revert to within 10% of baseline (i) average suicide risk levels and (ii) suicide attempt rates by two simulation-years after their conclusion.

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_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',

```

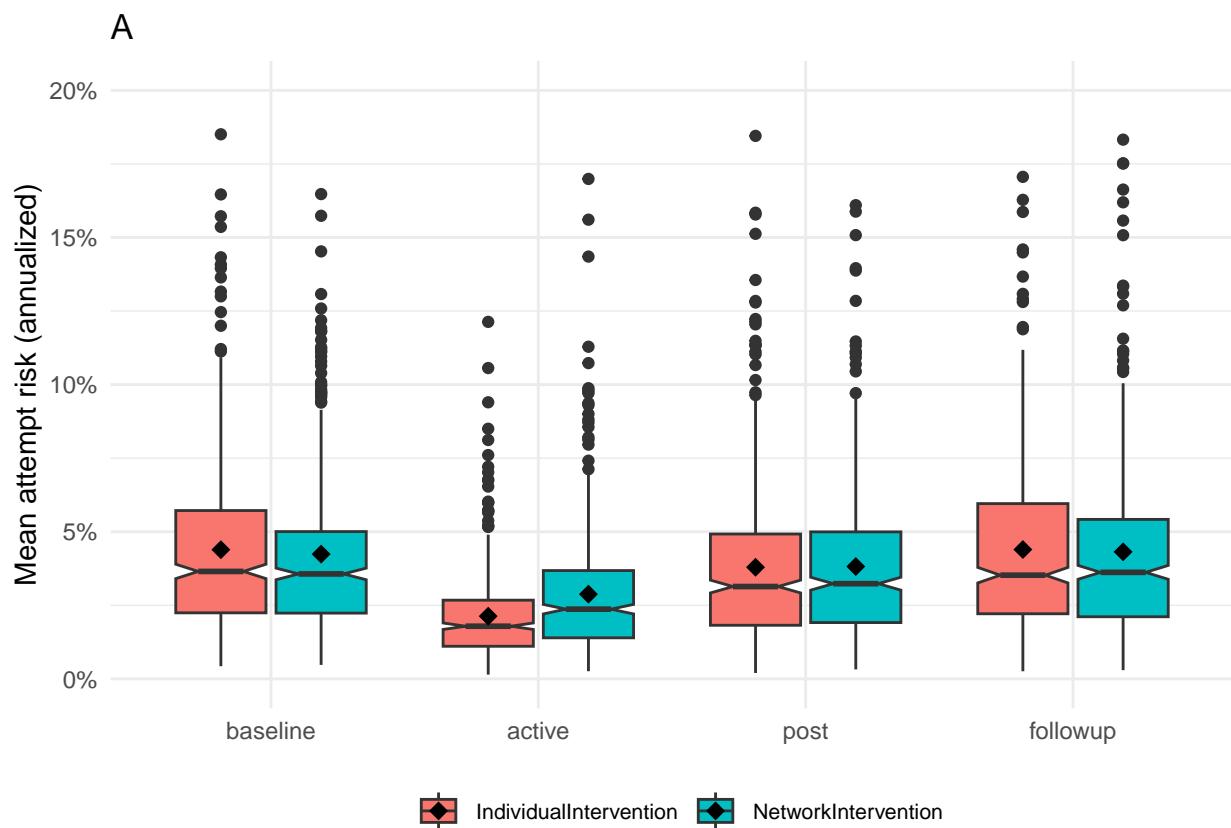
```

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 13 rows containing non-finite values ('stat_boxplot()').



Plot relative change in risk levels after baseline.

```

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)
  )

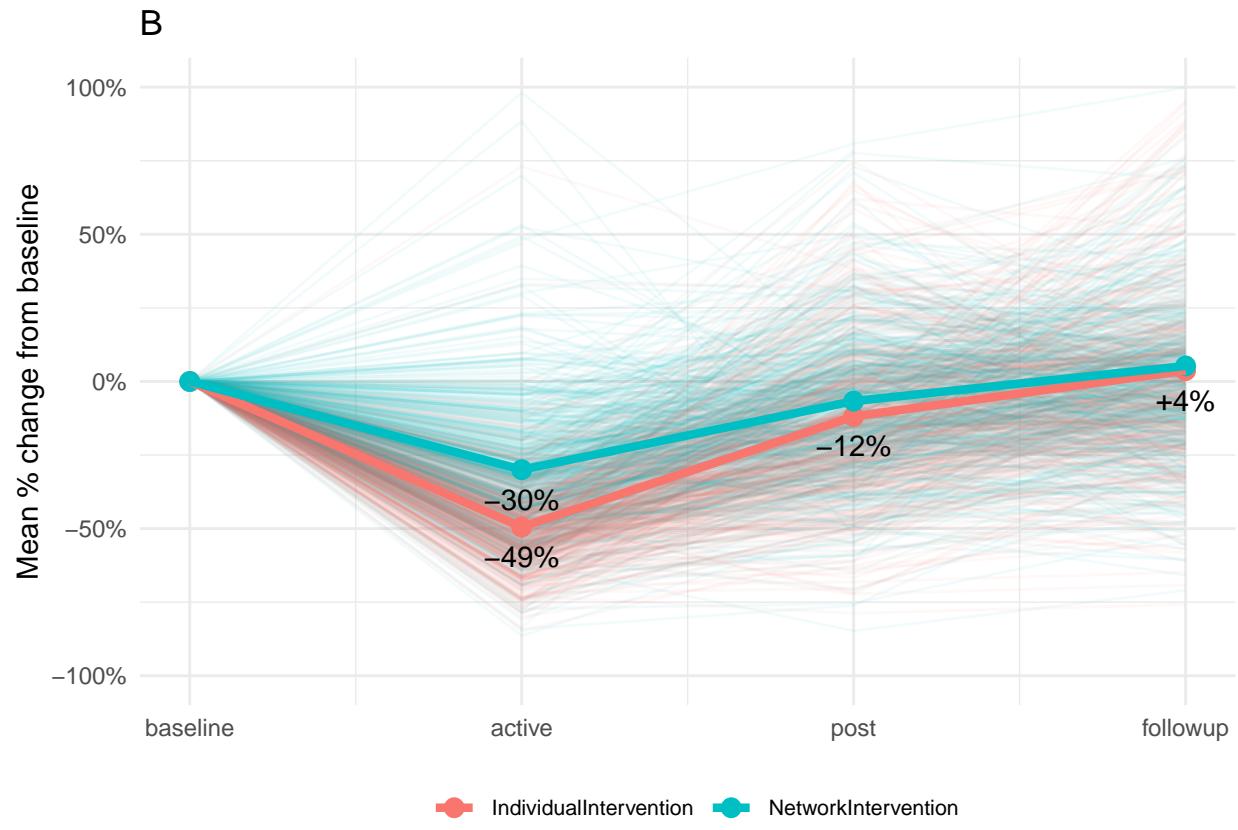
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.

risk_imp_plot

## Warning: Removed 25 rows containing missing values ('geom_line()').

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

```

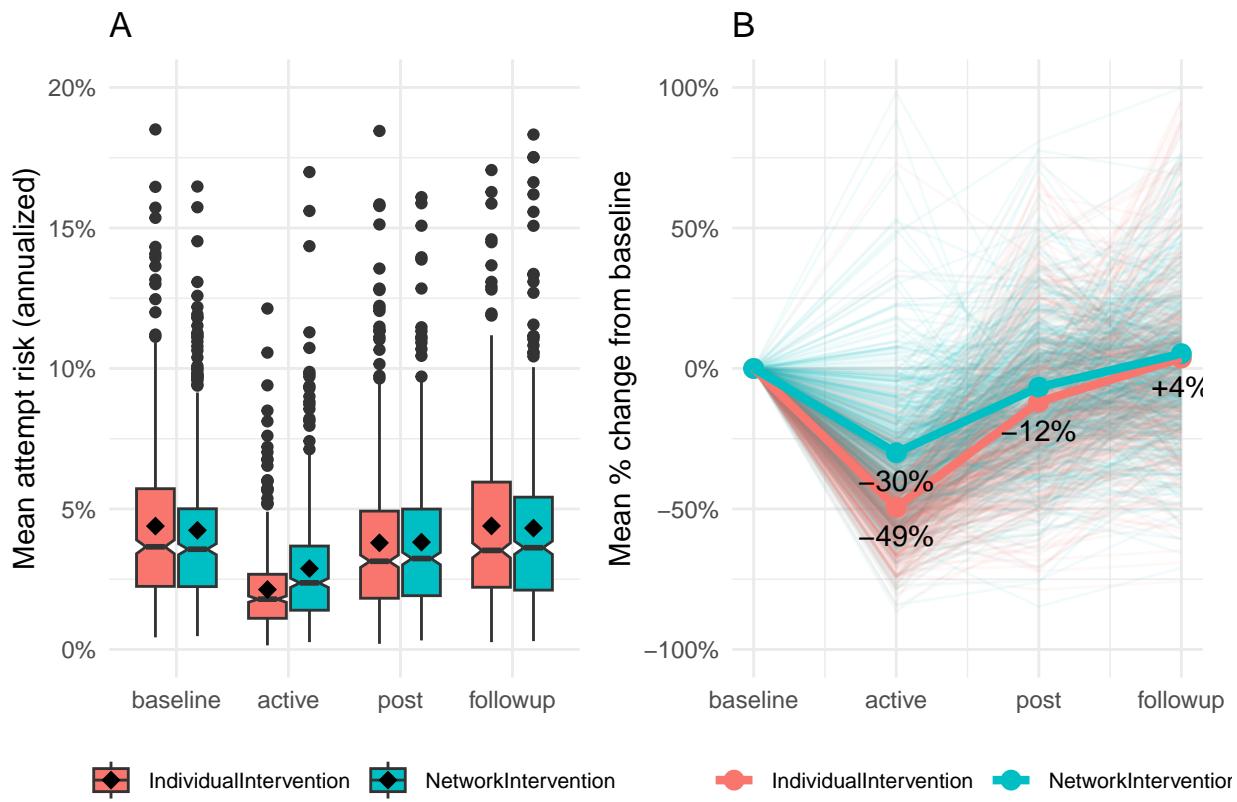


Create combined risk change plot.

```
risk_plot <- (risk_boxplot + risk_imp_plot)

risk_plot

## Warning: Removed 13 rows containing non-finite values ('stat_boxplot()').
## Warning: Removed 25 rows containing missing values ('geom_line()').
## Warning: Removed 2 rows containing missing values ('geom_text()').
```



Plot attempts observed in each condition

```

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.

## 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.

attempt_count_plot <- ggplot(
  data = attempt_count1,

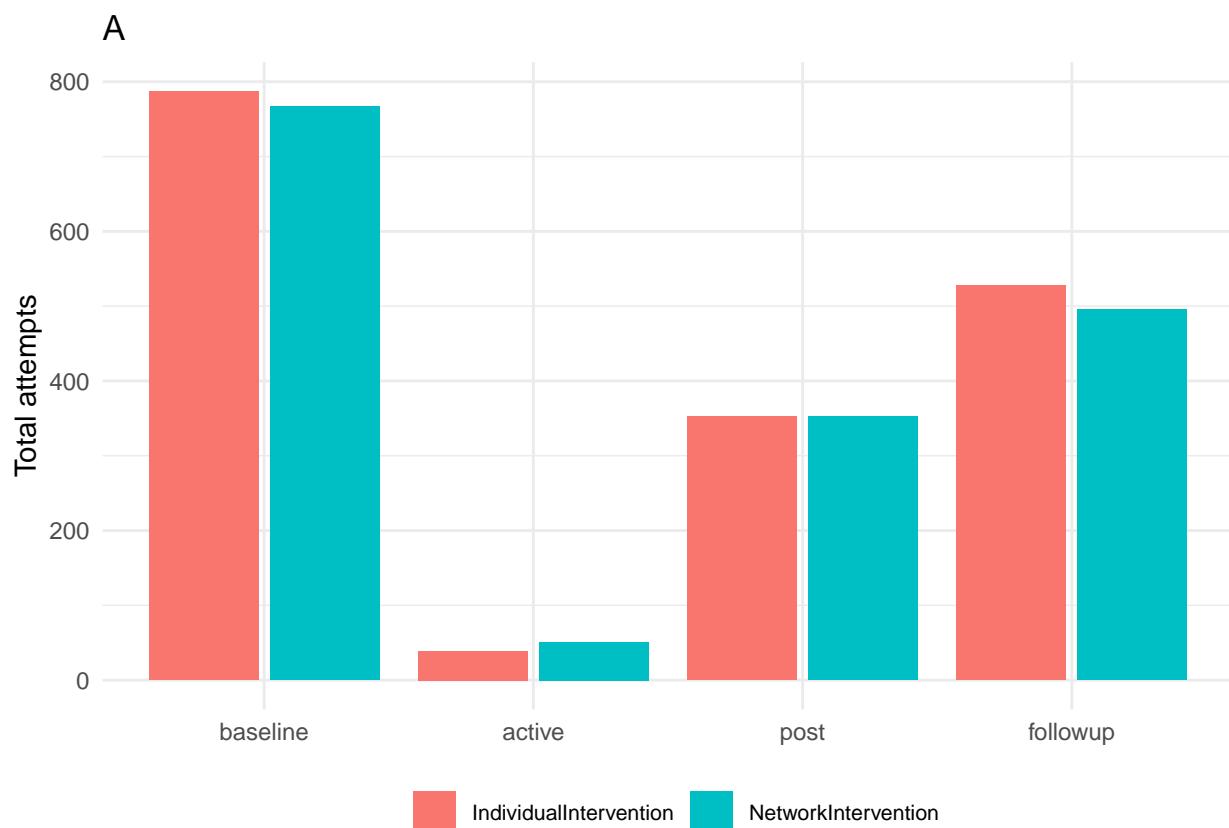
```

```

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_count1 %>%
  filter(phase != 'baseline') %>%
  group_by(intv_class_name) %>%
  summarise(total_attempts = sum(observed_attempts))

```

```

## # A tibble: 2 x 2
##   intv_class_name     total_attempts
##   <chr>                  <int>
## 1 IndividualIntervention      920
## 2 NetworkIntervention        899

```

```

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 = NetworkIntervention - IndividualIntervention)

```

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

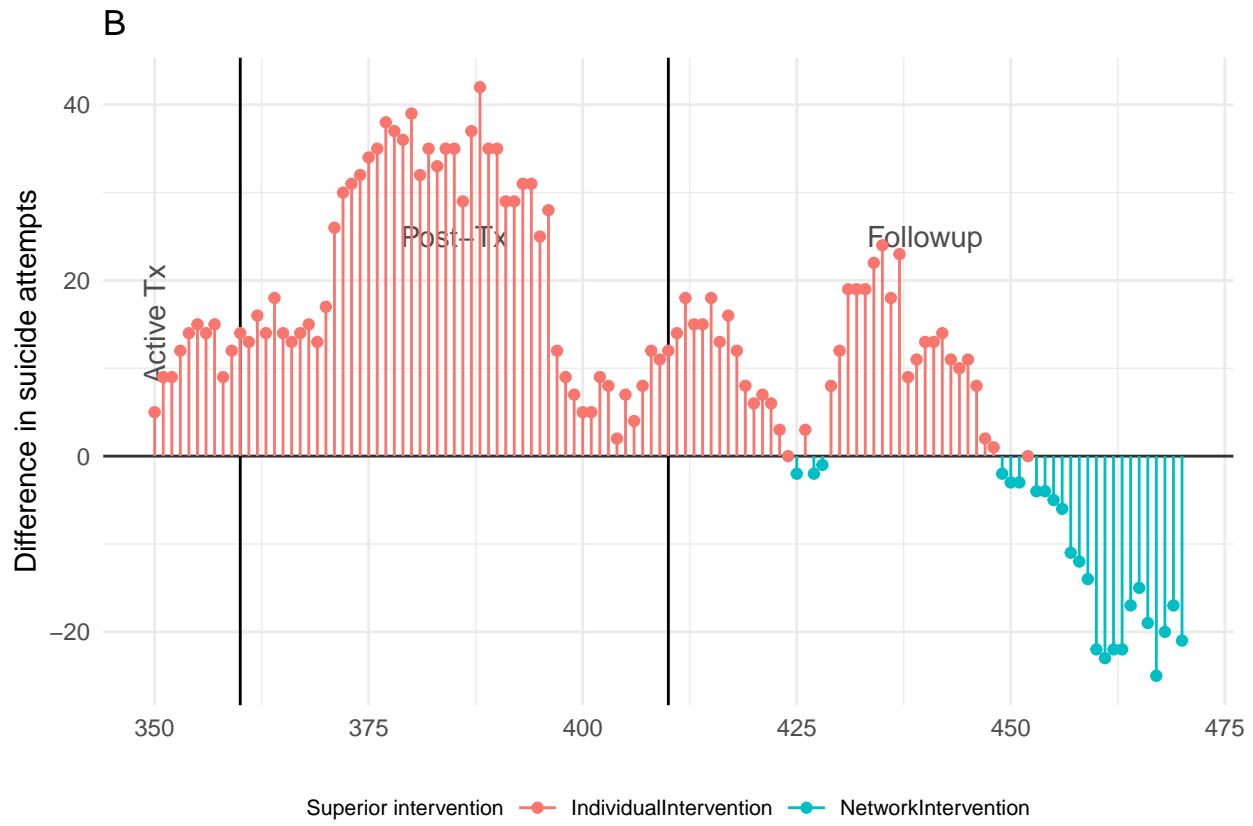
```

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

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

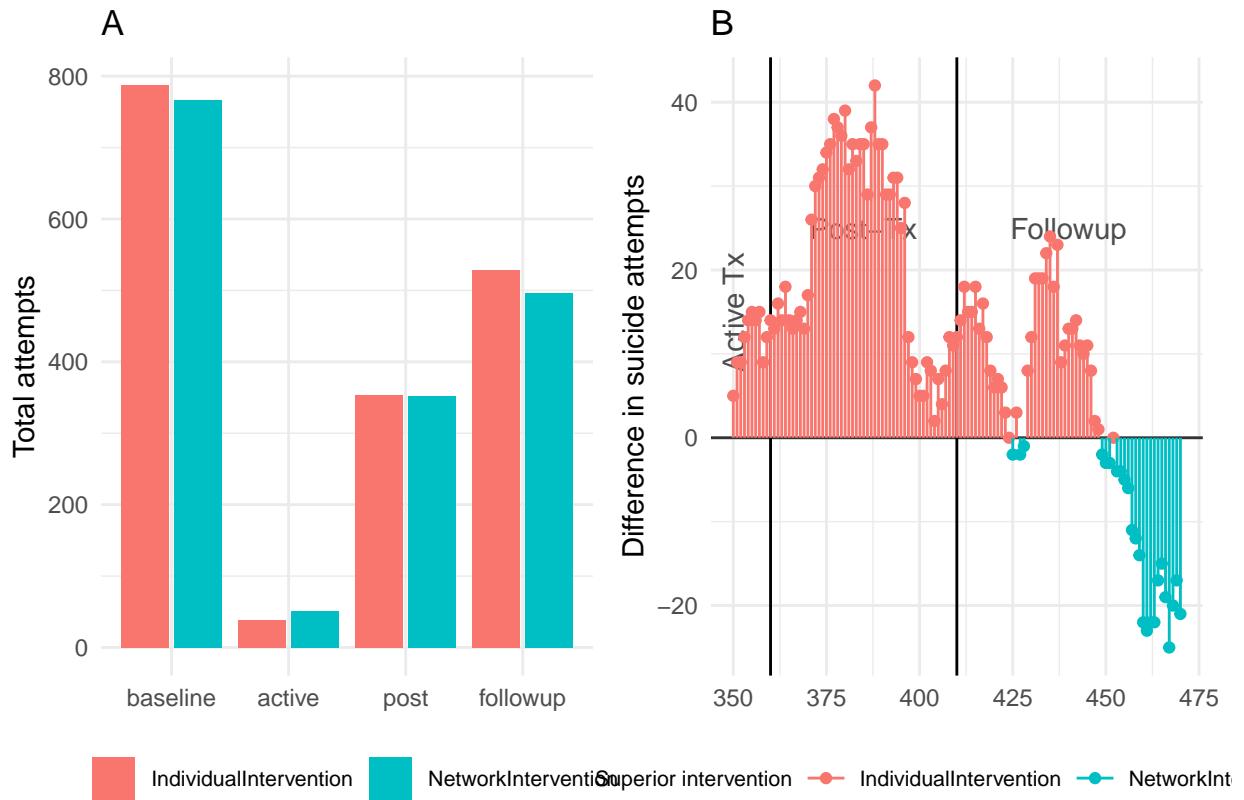
```



Combine the attempt plots into a single figure

```
attempt_plot <- attempt_count_plot + cum_advantage_plot

attempt_plot
```



2. NHI-effectiveness moderators

2a. Agents with fewer ties at baseline (including isolates) will demonstrate greater improvement after NHI treatment, as measured by the increase in their total degree and decrease in their suicide risk (compared to their baseline scores).

```
risk_change_df <- agents_db %>%
  filter(phase %in% c('baseline', 'active', 'post', 'followup')) %>%
  group_by(sim_id, id, intv_class_name, phase) %>%
  summarise(
    max_tick = max(tick),
    emulatable alters = mean(emulatable alters),
    cur_risk = mean(cur_risk)) %>%
  as_tibble() %>%
  group_by(sim_id, id) %>%
  arrange(sim_id, id, max_tick) %>%
  mutate(
    baseline alters = emulatable alters[1],
    alter_change = emulatable alters - emulatable alters[1],
    risk_change = cur_risk - cur_risk[1]) %>%
  filter(phase != 'baseline')
```

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

```

ggplot(risk_change_df, aes(baseline_alters, y = risk_change, color = intv_class_name)) +
  facet_grid(phase ~ intv_class_name) +
  geom_point(alpha = .01) +
  geom_smooth(method = 'lm', se = F) +
  scale_y_continuous(limits = c(-.005, .005))

```

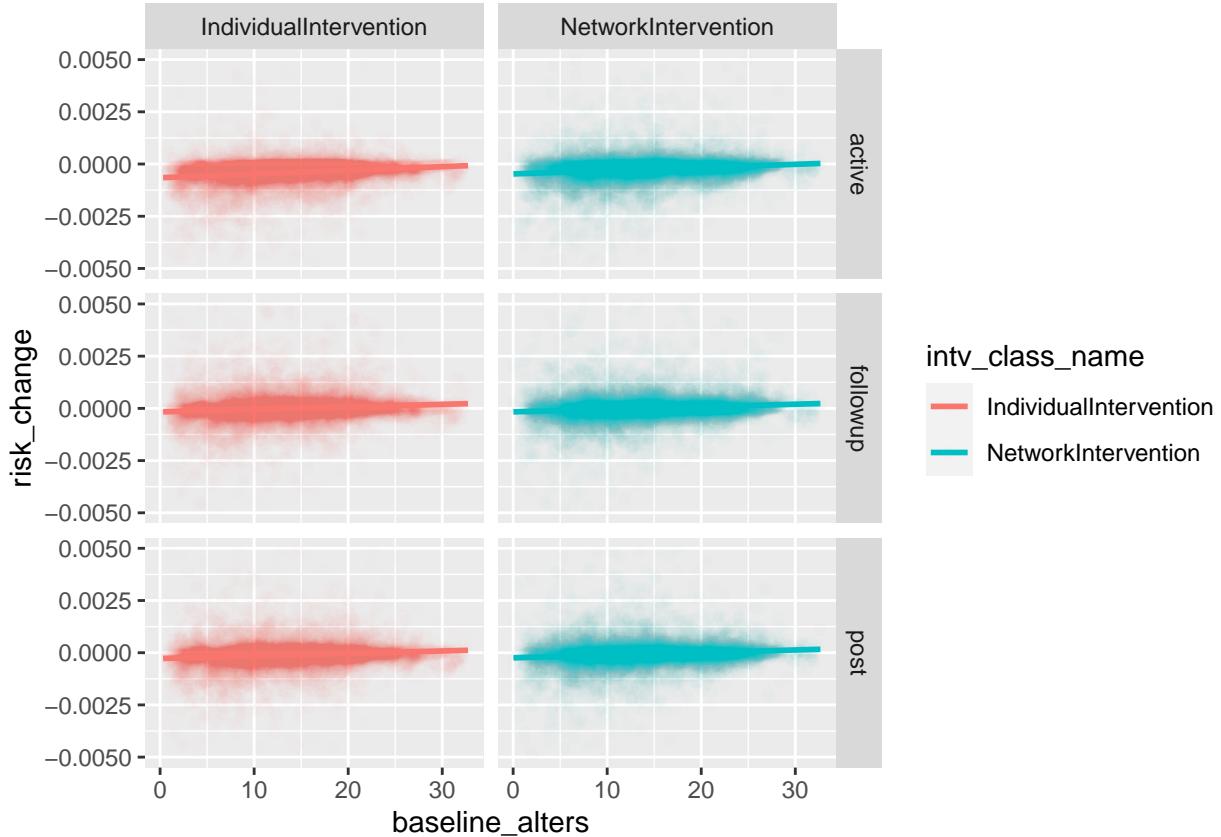
```

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 228 rows containing non-finite values ('stat_smooth()').

## Warning: Removed 228 rows containing missing values ('geom_point()').

```



```

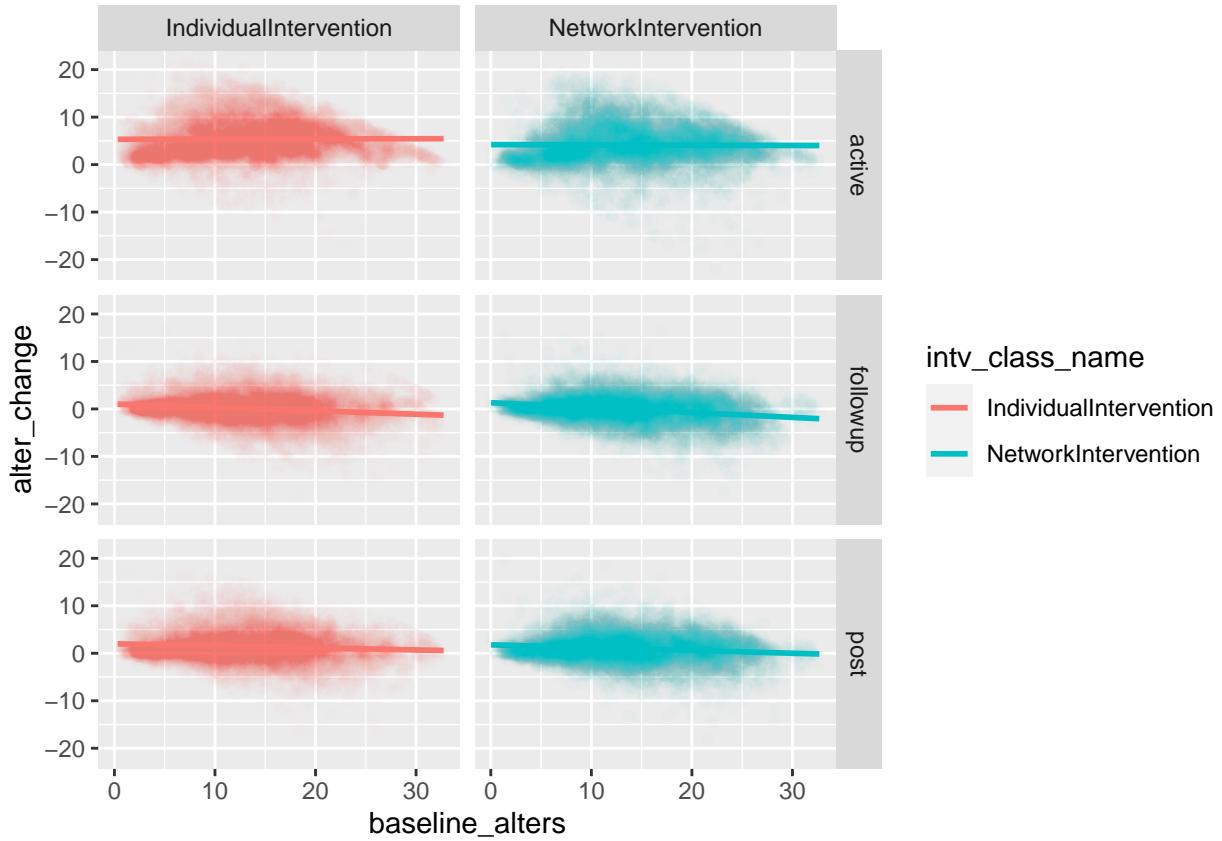
ggplot(risk_change_df, aes(baseline_alters, y = alter_change, color = intv_class_name)) +
  facet_grid(phase ~ intv_class_name) +
  geom_point(alpha = .01) +
  geom_smooth(method = 'lm', se = F)

```

```

## `geom_smooth()` using formula = 'y ~ x'

```



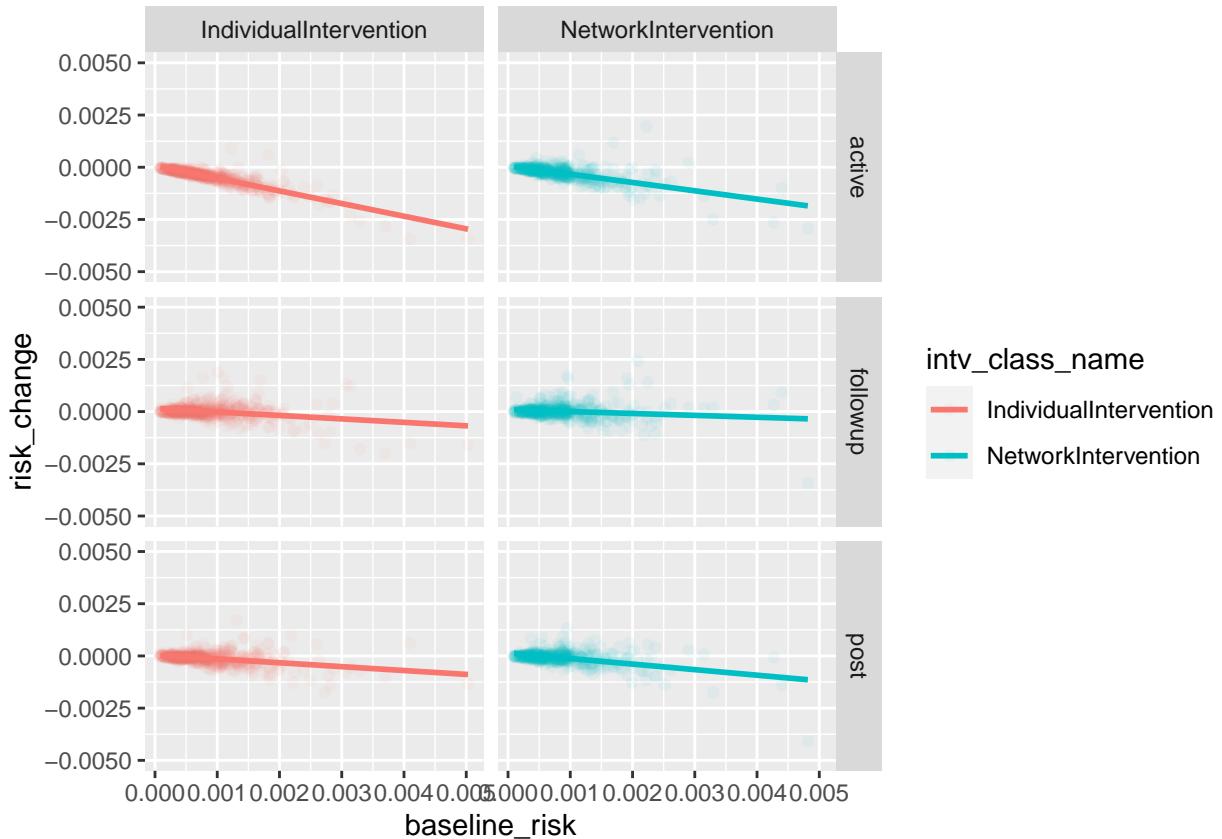
- b. Groups with higher average suicide risk scores at baseline, will demonstrate greater improvements after NHI treatment, as measured by increased edge density and decreased average suicide risk.

```
group_imp_df <- agents_db %>%
  filter(phase %in% c('baseline', 'active', 'post', 'followup')) %>%
  group_by(sim_id, intv_class_name, phase) %>%
  summarise(
    max_tick = max(tick),
    cur_risk = mean(cur_risk),
    n_edges = mean(emulatable alters)/2) %>%
  as_tibble() %>%
  arrange(sim_id, max_tick) %>%
  group_by(sim_id) %>%
  mutate(
    baseline_risk = cur_risk[1],
    risk_change = cur_risk - cur_risk[1],
    edge_change = n_edges - n_edges[1]) %>%
  ungroup() %>%
  filter(phase != 'baseline')
```

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

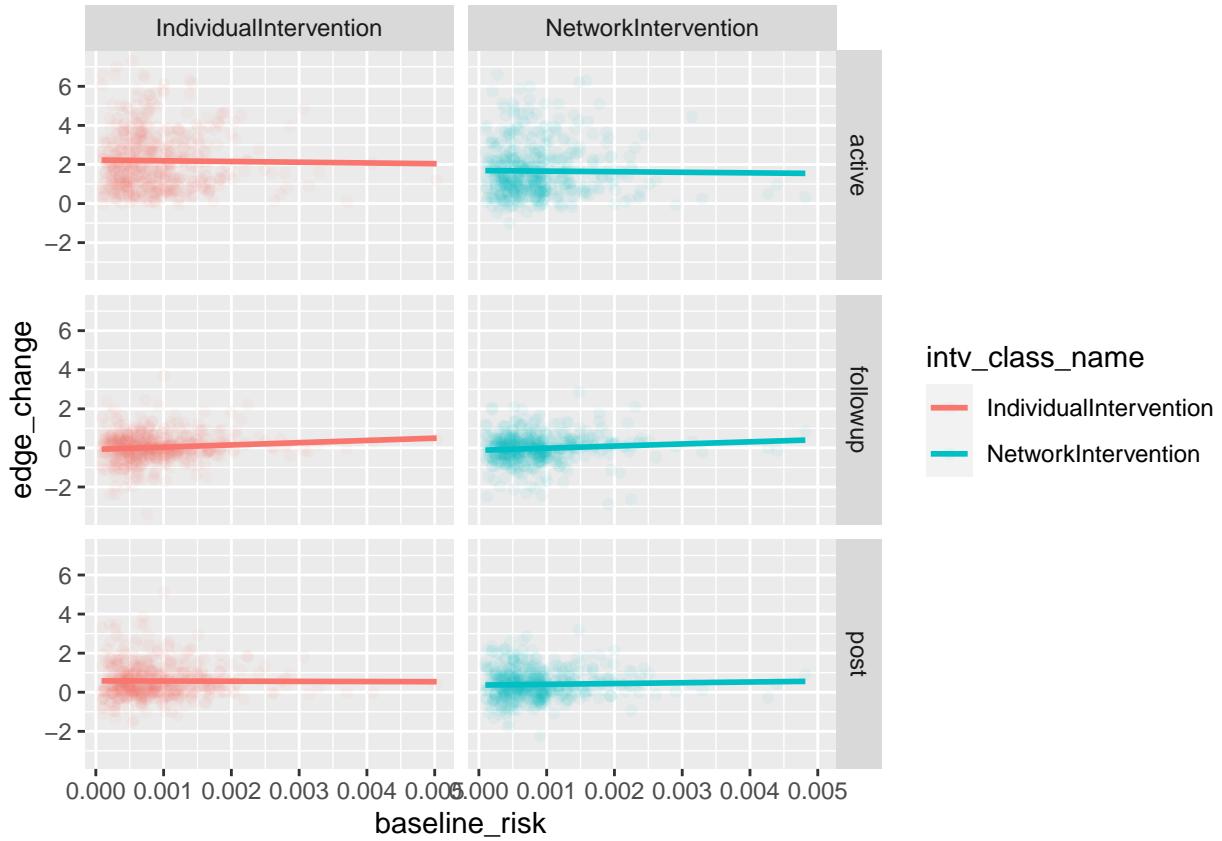
```
ggplot(group_imp_df, aes(baseline_risk, y = risk_change, color = intv_class_name)) +
  facet_grid(phase ~ intv_class_name) +
  geom_point(alpha = .05) +
  geom_smooth(method = 'lm', se = F) +
  scale_y_continuous(limits = c(-.005, .005))
```

`geom_smooth()` using formula = 'y ~ x'



```
ggplot(group_imp_df, aes(baseline_risk, y = edge_change, color = intv_class_name)) +
  facet_grid(phase ~ intv_class_name) +
  geom_point(alpha = .05) +
  geom_smooth(method = 'lm', se = F)
```

`geom_smooth()` using formula = 'y ~ x'



3. Exploratory

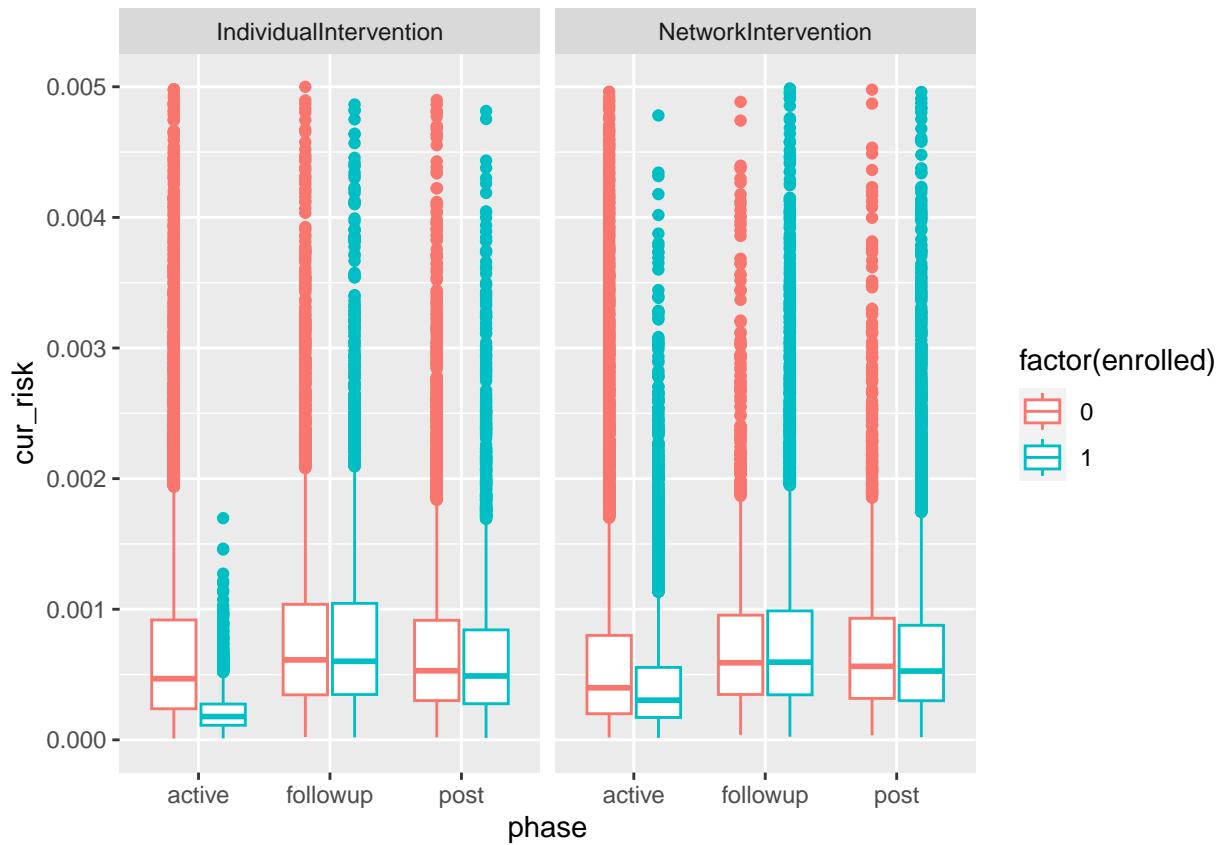
3a. Among agents not enrolled in an intervention, NHIs will produce greater benefits than INDs, as measured by suicide attempt risk and post-intervention suicide attempts.

```
risk_change_df <- agents_db %>%
  filter(phase %in% c('baseline', 'active', 'post', 'followup')) %>%
  group_by(sim_id, id, intv_class_name, phase, enrolled) %>%
  summarise(
    max_tick = max(tick),
    cur_risk = mean(cur_risk),
    cur_attempts = sum(cur_attempt)) %>%
  as_tibble() %>%
  ungroup() %>%
  filter(phase != 'baseline')
```

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

```
ggplot(risk_change_df, aes(x = phase, y = cur_risk, color = factor(enrolled))) +
  facet_grid(. ~ intv_class_name) +
  geom_boxplot() +
  scale_y_continuous(limits = c(0, .005))
```

```
## Warning: Removed 603 rows containing non-finite values ('stat_boxplot()').
```



```
Disconnect from database
```

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