HW5

Ruwen Zhou

11/18/2020

Problem 1

Read in the data.

```
homicide_df =
  read_csv("homicide_data/homicide-data.csv") %>%
  mutate(
    city_state = str_c(city, state, sep = "_"),
    resolved = case_when(
      disposition == "Closed without arrest" ~ "unsolved",
      disposition == "Open/No arrest"        ~ "unsolved",
      disposition == "Closed by arrest"      ~ "solved",
    )
  ) %>%
  select(city_state, resolved) %>%
  filter(city_state != "Tulsa_AL")
```

```
## Parsed with column specification:
## cols(
##   uid = col_character(),
##   reported_date = col_double(),
##   victim_last = col_character(),
##   victim_first = col_character(),
##   victim_race = col_character(),
##   victim_age = col_character(),
##   victim_sex = col_character(),
##   city = col_character(),
##   state = col_character(),
##   lat = col_double(),
##   lon = col_double(),
##   disposition = col_character()
## )
```

Let's look at this a bit

```
aggregate_df =
  homicide_df %>%
  group_by(city_state) %>%
  summarize(
    hom_total = n(),
    hom_unsolved = sum(resolved == "unsolved")
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

Can I do a prop test for a single city?

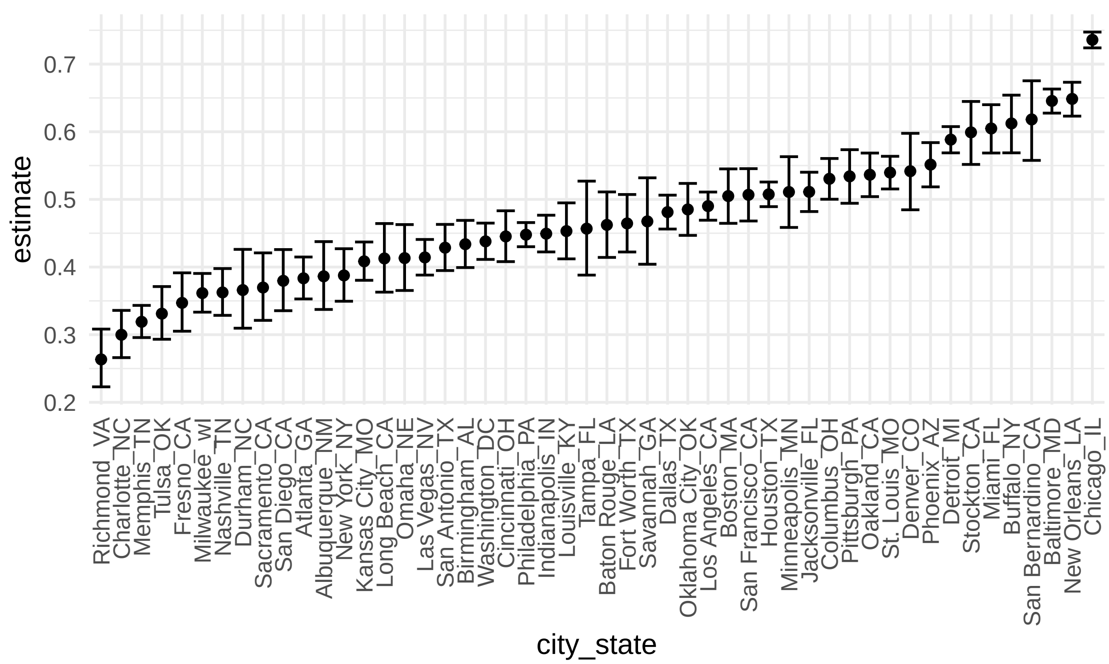
```r
prop.test(
  aggregate_df %>% filter(city_state == "Baltimore_MD") %>% pull(hom_unsolved),
  aggregate_df %>% filter(city_state == "Baltimore_MD") %>% pull(hom_total)) %>%
  broom::tidy()
```

```
## # A tibble: 1 x 8
##   estimate statistic  p.value parameter conf.low conf.high method    alternative
##      <dbl>     <dbl>    <dbl>     <int>    <dbl>     <dbl> <chr>     <chr>
## 1    0.646      239. 6.46e-54         1    0.628     0.663 1-sample~ two.sided
```

Try to iterate ....

```r
results_df =
  aggregate_df %>%
  mutate(
    prop_tests = map2(.x = hom_unsolved, .y = hom_total, ~prop.test(x = .x, n = .y)),
    tidy_tests = map(.x = prop_tests, ~broom::tidy(.x))
  ) %>%
  select(-prop_tests) %>%
  unnest(tidy_tests) %>%
  select(city_state, estimate, conf.low, conf.high)
```

```r
results_df %>%
  mutate(city_state = fct_reorder(city_state, estimate)) %>%
  ggplot(aes(x = city_state, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



```r
homicide_df =
  read_csv("homicide_data/homicide-data.csv") %>%
  mutate(
    city_state = str_c(city, state, sep = "_"),
    resolved = case_when(
      disposition == "Closed without arrest" ~ "unsolved",
      disposition == "Open/No arrest"        ~ "unsolved",
      disposition == "Closed by arrest"      ~ "solved",
    )
  ) %>%
  select(city_state, resolved) %>%
  filter(city_state != "Tulsa_AL") %>%
  nest(data = resolved)
```

```
## Parsed with column specification:
## cols(
##   uid = col_character(),
##   reported_date = col_double(),
##   victim_last = col_character(),
##   victim_first = col_character(),
##   victim_race = col_character(),
##   victim_age = col_character(),
##   victim_sex = col_character(),
##   city = col_character(),
##   state = col_character(),
##   lat = col_double(),
##   lon = col_double(),
##   disposition = col_character()
## )
```

Problem 2

Create a tidy dataframe containing data from all participants, including the subject ID, arm, and observations over time

```r
tidy_df = tibble(
    path = list.files("lda_data"),
  ) %>%
  mutate(
    path = str_c("lda_data/", path),
    data = map(.x = path, ~read_csv(.x)),
    arm_id = str_remove(path, "lda_data/"),
    arm_id = str_remove(arm_id, ".csv")) %>%
  unnest(data) %>%
  select(-path) %>%
  pivot_longer(
    week_1:week_8,
    values_to = "observation_data",
    names_to = "week",
    names_prefix = "week_",
  ) %>%
  separate(arm_id, into = c("arm", "subject_id"), sep = "_")
```

```
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
```

```
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
```

```
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
```

```
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
## Parsed with column specification:
## cols(
##   week_1 = col_double(),
##   week_2 = col_double(),
##   week_3 = col_double(),
##   week_4 = col_double(),
##   week_5 = col_double(),
```

```
##   week_6 = col_double(),
##   week_7 = col_double(),
##   week_8 = col_double()
## )
```

```
tidy_df %>% knitr::kable()
```

arm	subject_id	week	observation_data
con	01	1	0.20
con	01	2	-1.31
con	01	3	0.66
con	01	4	1.96
con	01	5	0.23
con	01	6	1.09
con	01	7	0.05
con	01	8	1.94
con	02	1	1.13
con	02	2	-0.88
con	02	3	1.07
con	02	4	0.17
con	02	5	-0.83
con	02	6	-0.31
con	02	7	1.58
con	02	8	0.44
con	03	1	1.77
con	03	2	3.11
con	03	3	2.22
con	03	4	3.26
con	03	5	3.31
con	03	6	0.89
con	03	7	1.88
con	03	8	1.01
con	04	1	1.04
con	04	2	3.66
con	04	3	1.22
con	04	4	2.33
con	04	5	1.47
con	04	6	2.70
con	04	7	1.87
con	04	8	1.66
con	05	1	0.47
con	05	2	-0.58
con	05	3	-0.09
con	05	4	-1.37
con	05	5	-0.32
con	05	6	-2.17
con	05	7	0.45
con	05	8	0.48
con	06	1	2.37
con	06	2	2.50
con	06	3	1.59
con	06	4	-0.16
con	06	5	2.08

arm	subject_id	week	observation_data
con	06	6	3.07
con	06	7	0.78
con	06	8	2.35
con	07	1	0.03
con	07	2	1.21
con	07	3	1.13
con	07	4	0.64
con	07	5	0.49
con	07	6	-0.12
con	07	7	-0.07
con	07	8	0.46
con	08	1	-0.08
con	08	2	1.42
con	08	3	0.09
con	08	4	0.36
con	08	5	1.18
con	08	6	-1.16
con	08	7	0.33
con	08	8	-0.44
con	09	1	0.08
con	09	2	1.24
con	09	3	1.44
con	09	4	0.41
con	09	5	0.95
con	09	6	2.75
con	09	7	0.30
con	09	8	0.03
con	10	1	2.14
con	10	2	1.15
con	10	3	2.52
con	10	4	3.44
con	10	5	4.26
con	10	6	0.97
con	10	7	2.73
con	10	8	-0.53
exp	01	1	3.05
exp	01	2	3.67
exp	01	3	4.84
exp	01	4	5.80
exp	01	5	6.33
exp	01	6	5.46
exp	01	7	6.38
exp	01	8	5.91
exp	02	1	-0.84
exp	02	2	2.63
exp	02	3	1.64
exp	02	4	2.58
exp	02	5	1.24
exp	02	6	2.32
exp	02	7	3.11
exp	02	8	3.78
exp	03	1	2.15

arm	subject_id	week	observation_data
exp	03	2	2.08
exp	03	3	1.82
exp	03	4	2.84
exp	03	5	3.36
exp	03	6	3.61
exp	03	7	3.37
exp	03	8	3.74
exp	04	1	-0.62
exp	04	2	2.54
exp	04	3	3.78
exp	04	4	2.73
exp	04	5	4.49
exp	04	6	5.82
exp	04	7	6.00
exp	04	8	6.49
exp	05	1	0.70
exp	05	2	3.33
exp	05	3	5.34
exp	05	4	5.57
exp	05	5	6.90
exp	05	6	6.66
exp	05	7	6.24
exp	05	8	6.95
exp	06	1	3.73
exp	06	2	4.08
exp	06	3	5.40
exp	06	4	6.41
exp	06	5	4.87
exp	06	6	6.09
exp	06	7	7.66
exp	06	8	5.83
exp	07	1	1.18
exp	07	2	2.35
exp	07	3	1.23
exp	07	4	1.17
exp	07	5	2.02
exp	07	6	1.61
exp	07	7	3.13
exp	07	8	4.88
exp	08	1	1.37
exp	08	2	1.43
exp	08	3	1.84
exp	08	4	3.60
exp	08	5	3.80
exp	08	6	4.72
exp	08	7	4.68
exp	08	8	5.70
exp	09	1	-0.40
exp	09	2	1.08
exp	09	3	2.66
exp	09	4	2.70
exp	09	5	2.80

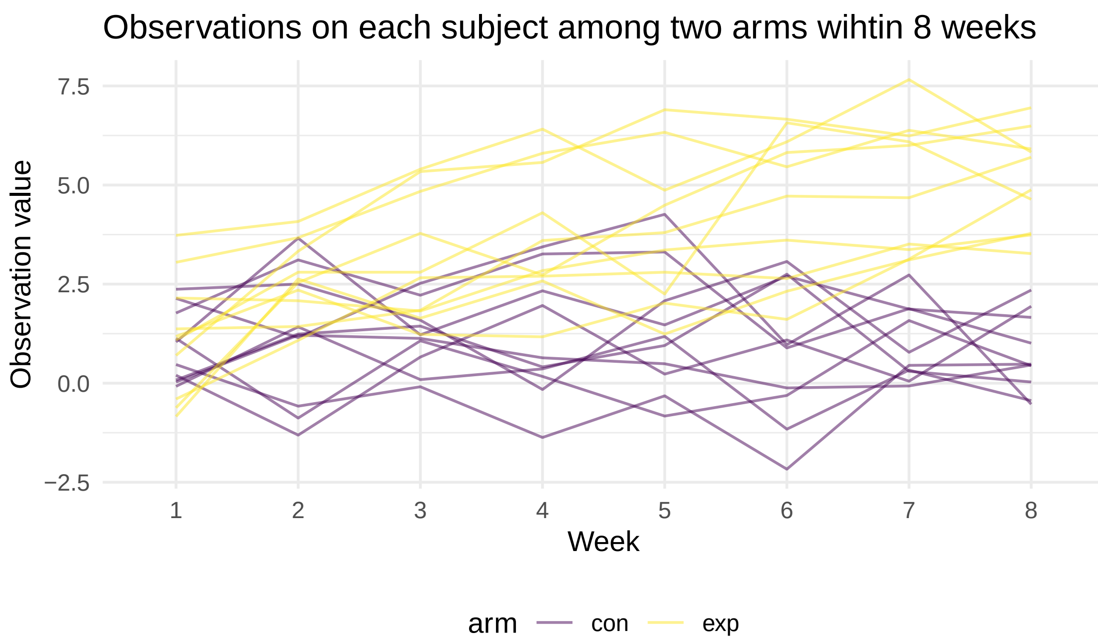
arm	subject_id	week	observation_data
exp	09	6	2.64
exp	09	7	3.51
exp	09	8	3.27
exp	10	1	1.09
exp	10	2	2.80
exp	10	3	2.80
exp	10	4	4.30
exp	10	5	2.25
exp	10	6	6.57
exp	10	7	6.09
exp	10	8	4.64

```
tidy_df
```

```
## # A tibble: 160 x 4
##    arm   subject_id week  observation_data
##    <chr> <chr>      <chr>            <dbl>
##  1 con   01         1                 0.2 
##  2 con   01         2                -1.31
##  3 con   01         3                 0.66
##  4 con   01         4                 1.96
##  5 con   01         5                 0.23
##  6 con   01         6                 1.09
##  7 con   01         7                 0.05
##  8 con   01         8                 1.94
##  9 con   02         1                 1.13
## 10 con   02         2                -0.88
## # ... with 150 more rows
```

Make a spaghetti plot showing observations on each subject over time, and comment on differences between groups

```
tidy_df %>%
  unite("arm_id", c(arm, subject_id), sep = "_", remove = F) %>%
  ggplot(aes(x = week, y = observation_data)) +
  geom_path(aes(color = arm, group = as.factor(arm_id)),alpha = 0.5) +
  labs(
    x = "Week",
    y = "Observation value",
    title = "Observations on each subject among two arms wihtin 8 weeks"
    )
```



The observation data of experimental arm increases faster than the control arm over time. The measure in control arm is more stable and decreases a little bit after week 6.

Problem 3

T test

```
n = 30
mu = 0
sigma = 5
x = rnorm(n, mean = mu, sd = sigma)
t.test(x, mu = mu, conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  x
## t = 0.45714, df = 29, p-value = 0.651
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.254171  1.976210
## sample estimates:
## mean of x
## 0.3610195
```

Generate 5000 datasets from the model

```
sim_test = function(n = 30, mu = 0, sigma = 5) {
     x = rnorm(n, mean = mu, sd = sigma)
```

```r
    t_test = t.test(x, conf.int = 0.95) %>% broom::tidy()
    t_test
}
output = vector("list", 5000)
for (i in 1:5000) {
  output[[i]] = sim_test()
}
output %>% head()
```

```
## [[1]]
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method     alternative
##      <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>      <chr>
## 1   -0.351    -0.381   0.706        29    -2.24      1.54 One Sampl~ two.sided
##
## [[2]]
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method     alternative
##      <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>      <chr>
## 1   -0.316    -0.384   0.703        29    -2.00      1.37 One Sampl~ two.sided
##
## [[3]]
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method     alternative
##      <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>      <chr>
## 1  -0.0366   -0.0343   0.973        29    -2.22      2.14 One Sampl~ two.sided
##
## [[4]]
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method     alternative
##      <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>      <chr>
## 1   -0.160    -0.154   0.878        29    -2.27      1.95 One Sampl~ two.sided
##
## [[5]]
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method     alternative
##      <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>      <chr>
## 1  -0.0274   -0.0235   0.981        29    -2.41      2.36 One Sampl~ two.sided
##
## [[6]]
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method     alternative
##      <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>      <chr>
## 1     1.52      1.85  0.0748        29   -0.163      3.21 One Sampl~ two.sided
```

for mu = {0,1,2,3,4,5,6}

```r
set.seed(1000)
combine =
  tibble(mu = c(0, 1, 2, 3, 4, 5, 6)) %>%
  mutate(
```

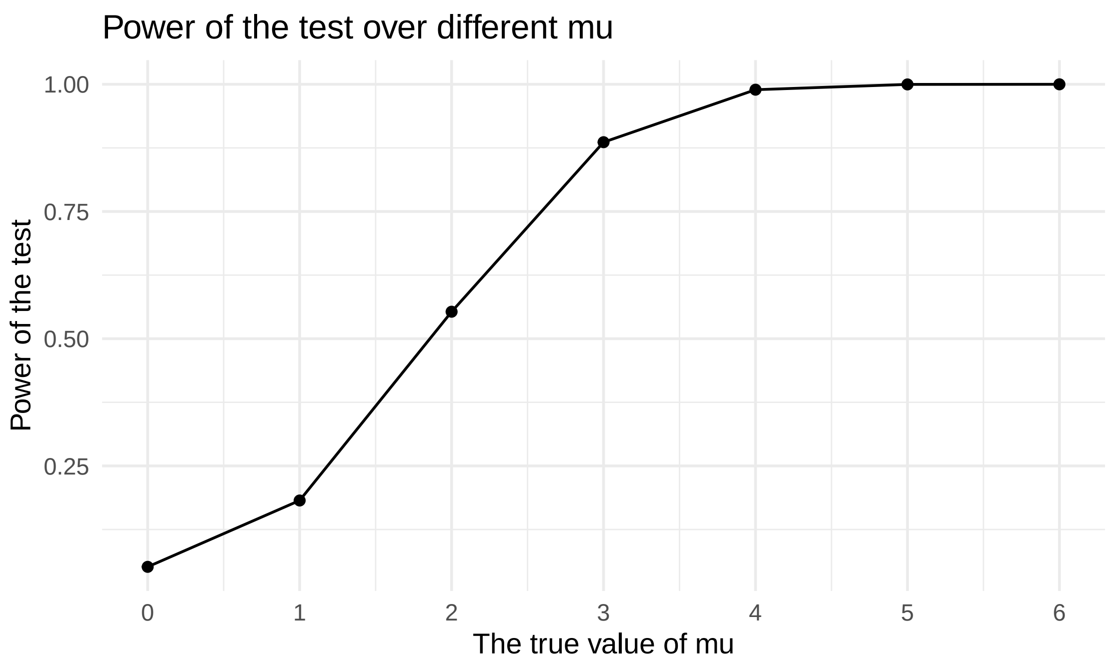
```
    output = map(.x = mu, ~rerun(5000, sim_test(mu = .x))),
    new = map(output, bind_rows)) %>%
  select(-output) %>%
  unnest(new)
combine %>% head()
```

```
## # A tibble: 6 x 9
##      mu estimate statistic p.value parameter conf.low conf.high method
##   <dbl>    <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>
## 1     0   -0.758    -0.850   0.402        29   -2.58       1.06 One S~
## 2     0   -0.593    -0.651   0.520        29   -2.46       1.27 One S~
## 3     0    1.02      1.13    0.267        29   -0.824      2.86 One S~
## 4     0    0.991     1.04    0.306        29   -0.956      2.94 One S~
## 5     0    0.183     0.235   0.816        29   -1.41       1.78 One S~
## 6     0   -0.101    -0.120   0.905        29   -1.82       1.62 One S~
## # ... with 1 more variable: alternative <chr>
```

Make a plot showing the proportion of times the null was rejected (the power of the test) on the y axis and the true value of mu on the x axis.

```
combine %>%
  filter(p.value < 0.05) %>%
  group_by(mu) %>%
  summarize(prop_rej = n()) %>%
  mutate(prop_rej = prop_rej/5000) %>%
  ggplot(aes(x = mu, y = prop_rej), color = mu) +
  geom_point() +
  geom_line() +
  scale_x_continuous(limits = c(0,6), breaks = seq(0,6,1)) +
  labs(
    title = "Power of the test over different mu",
    x = "The true value of mu",
    y = "Power of the test"
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



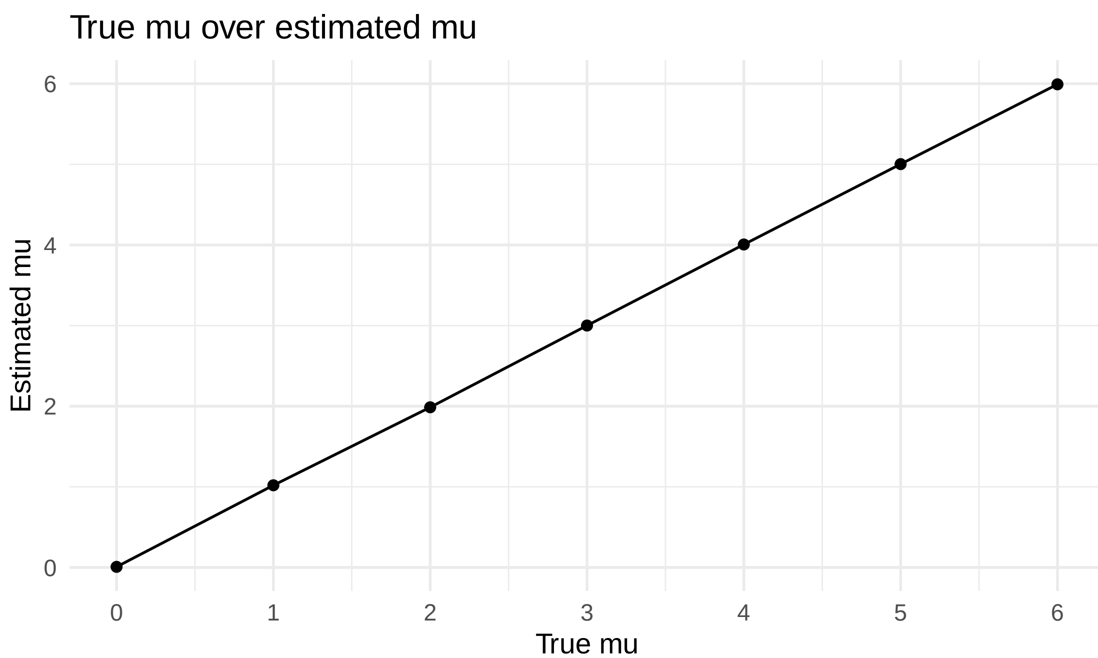
As the number of mu increases, the power of the test also increases. The power converges to 1 when mu = 4.

Make a plot showing the average estimate of mu on the y axis and the true value of mu on the x axis.

```
first_plot = combine %>%
  group_by(mu) %>%
  summarise(estimate_mu = mean(estimate)) %>%
  ggplot(aes(x = mu, y = estimate_mu), color = mu) +
  geom_point() +
  geom_line() +
  scale_x_continuous(limits = c(0,6), breaks = seq(0,6,1)) +
  labs(title = "True mu over estimated mu",
       x = "True mu",
       y = "Estimated mu")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
first_plot
```

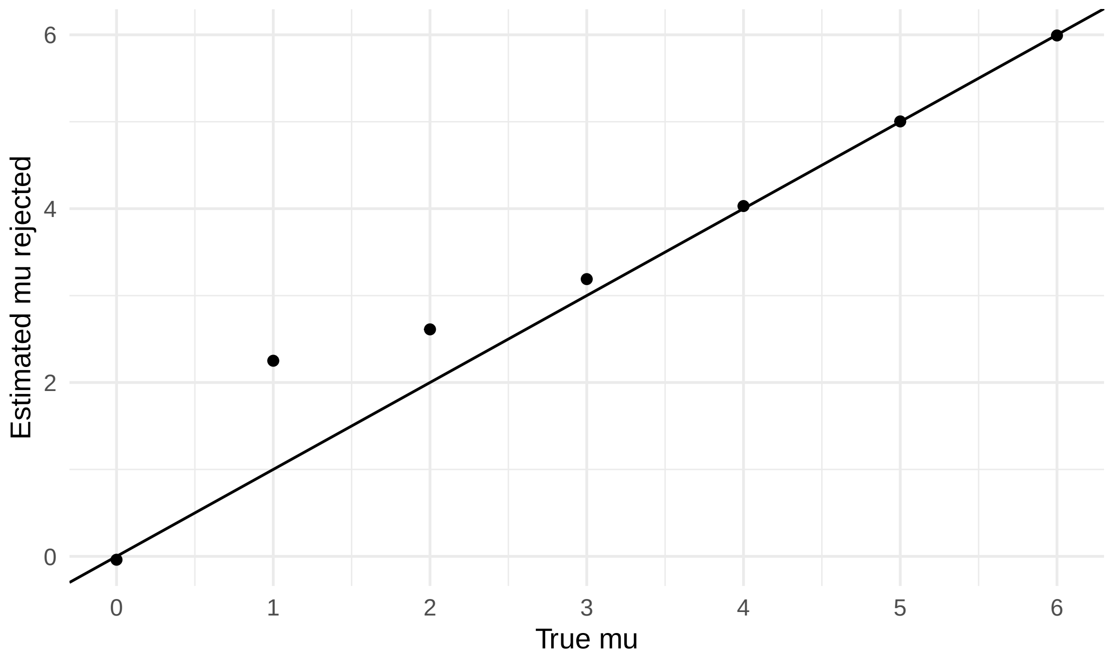


Make a second plot (or overlay on the first) the average estimate of mu only in samples for which the null was rejected on the y axis and the true value of mu on the x axis.

```
second_plot = combine %>%
  filter(p.value < 0.05) %>%
  group_by(mu) %>%
  summarize(rej_estimate_mu = mean(estimate)) %>%
  ggplot(aes(x = mu, y = rej_estimate_mu ), color = mu) +
  geom_point() +
  geom_abline() +
  scale_x_continuous(limits = c(0,6), breaks = seq(0,6,1)) +
  labs(x = "True mu",
    y = "Estimated mu rejected")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
second_plot
```



- From the two plots, when mu = 1, 2, average estimate of mu is not exactly equal to the true value of mu. When mu = 3,4,5,6, they are equal.
- Because when mu is close to 0, the number of samples for which the null was rejected decreases and the mu hat of these samples would be far away from 0.