

# Supplement 2

Descriptive statistics of baseline variables: SN:yes vs SN:no

*Peter Kamerman and Prinisha Pillay*

*17 March 2019*

## Contents

<b>Define bootstrap functions</b>	<b>2</b>
<b>Import data</b>	<b>3</b>
<b>Process data</b>	<b>3</b>
<b>Inspect data</b>	<b>4</b>
<b>Analyses</b>	<b>5</b>
Age . . . . .	5
Body mass . . . . .	9
Height . . . . .	13
Sex . . . . .	19
CD4 T-cell count . . . . .	24
Viral load . . . . .	28
Alcohol . . . . .	32
TB . . . . .	34
Currently infected with TB . . . . .	34
Currently receiving TB treatment? . . . . .	39
<b>Session information</b>	<b>44</b>

---

This script describes variables collected at baseline (visit day: 0, visit\_number: 1) conditioned on sensory neuropathy (SN) status (i.e., whether SN developed at any point during the follow-up period).

For descriptive statistics of baseline variables for the whole cohort, see: `suppl-01-descriptive-whole-cohort.Rmd/html/m`

The following data columns were not analysed:

- `pain` and `pain score`: related to SN only, therefore not relevant at baseline when everyone was free from SN.
  - `*_record`: inconsistent patient records.
  - `hba1c_percent/diabetic_hba1c` and `vitaminB12_pmol.l/vitaminB12_deficiency`: Zero and one participant had diabetes mellitus or vitamin B12 deficiency, respectively, so these data were not analysed.
  - `ID`, `visit_day`, `hivsn_present`, `visit_months`: provide sorting and grouping information only.
-

## Define bootstrap functions

Functions calculate the 95% confidence interval of the mean/median difference between SN:yes and SN:no, or the 95% confidence interval for the odds ratio between the two groups.

```
# Difference between means
## d = dataframe object
## i = boot index
## data_column = data column (character vector of length 1)
## grouping_column = grouping variable column (character vector of length 1)
boot_deltaMean <- function(d, i, data_column = NULL, grouping_column = NULL){
  # Sample
  df <- d[i, c(data_column, grouping_column)]
  # Rename columns
  colnames(df) <- c('x', 'y')
  # Calculate means
  df <- df %>%
    filter(!is.na(x)) %>%
    group_by(y) %>%
    summarise(mean = mean(x)) %>%
    ungroup()
  # Calculate difference in means
  df$mean[1] - df$mean[2]
}

# Difference between medians
## d = dataframe object
## i = boot index
## data_column = data column (character vector of length 1)
## grouping_column = grouping variable column (character vector of length 1)
boot_deltaMedian <- function(d, i, data_column = NULL, grouping_column = NULL){
  # Sample
  df <- d[i, c(data_column, grouping_column)]
  # Rename columns
  colnames(df) <- c('x', 'y')
  # Calculate means
  df <- df %>%
    filter(!is.na(x)) %>%
    group_by(y) %>%
    summarise(median = median(x)) %>%
    ungroup()
  # Calculate difference in means
  df$median[1] - df$median[2]
}

# Odds ratio
## d = dataframe object
## i = boot index
## data_column = data column (character vector of length 1)
## grouping_column = grouping variable column (character vector of length 1)
boot_OR <- function(d, i, data_column = NULL, grouping_column = NULL){
  # Sample
  df <- d[i, c(data_column, grouping_column)]
  # Rename columns
```

```

colnames(df) <- c('x', 'y')
# xtabulate
x_tab <- xtabs(~ x + y,
               data = df)
# Calculate odds ratio
fisher.test(x_tab)$estimate
}

```

---

## Import data

```

data <- read_rds('data-cleaned/clean_data.rds') %>%
  # Remove columns that won't be analysed
  select(-starts_with('pain'), -ends_with('_record'),
         -starts_with('vitaminB12'), -visit_months,
         -hba1c_percent, -vitaminB12_pmol.l, -diabetic_hba1c)

```

## Process data

Add a column indicating whether a participant developed SN at any time during the follow-up period, and then filter the dataframe to only contain rows of data from visit 1.

```

# Identify and extract information on SN development (at anytime)
# by looking at the presence of SN at the final visit
data_sn <- data %>%
  select(ID, visit_number, hivsn_present) %>%
  group_by(ID) %>%
  mutate(max_visit = max(visit_number)) %>%
  filter(visit_number == max_visit) %>%
  select(ID, hivsn_present) %>%
  rename(sn_present = hivsn_present)

# Join data_sn to data
data %<>%
  left_join(data_sn)

# Restrict data to the baseline visit (visit 1)
data %<>%
  filter(visit_number == 1)

# Order sn_present factor to improved plotting order
data %<>%
  mutate(sn_present = factor(sn_present,
                             levels = c('yes', 'no'),
                             ordered = TRUE))

```

## Inspect data

```
# Dimensions
dim(data)

## [1] 120 17

# Column names
names(data)

## [1] "ID" "visit_number"
## [3] "visit_day" "age_years"
## [5] "mass_kg" "height_m"
## [7] "sex" "hivsn_present"
## [9] "CD4_cell.ul" "viral_load_copies.ml"
## [11] "consumes_alcohol" "alcohol_units.week"
## [13] "TB_current" "pyridoxine_prophylaxis"
## [15] "rifafour_treatment" "ARV_regimen"
## [17] "sn_present"

# Head and tail
head(data)

## # A tibble: 6 x 17
## ID visit_number visit_day age_years mass_kg height_m sex
## <chr> <int> <int> <dbl> <dbl> <dbl> <fct>
## 1 001 1 0 59 41.4 1.56 F
## 2 002 1 0 23 70.2 1.56 F
## 3 003 1 0 27 75 1.64 M
## 4 004 1 0 26 68.8 1.74 M
## 5 005 1 0 37 107 1.6 F
## 6 006 1 0 34 85.5 1.53 F
## # ... with 10 more variables: hivsn_present <fct>, CD4_cell.ul <dbl>,
## # viral_load_copies.ml <dbl>, consumes_alcohol <fct>,
## # alcohol_units.week <int>, TB_current <fct>,
## # pyridoxine_prophylaxis <fct>, rifafour_treatment <fct>,
## # ARV_regimen <fct>, sn_present <ord>

tail(data)

## # A tibble: 6 x 17
## ID visit_number visit_day age_years mass_kg height_m sex
## <chr> <int> <int> <dbl> <dbl> <dbl> <fct>
## 1 115 1 0 29 55.1 1.66 M
## 2 116 1 0 30 93.7 1.55 F
## 3 117 1 0 30 58.2 1.6 F
## 4 118 1 0 30 61.2 1.64 F
## 5 119 1 0 22 62.7 1.63 F
## 6 120 1 0 58 71.2 1.74 M
## # ... with 10 more variables: hivsn_present <fct>, CD4_cell.ul <dbl>,
## # viral_load_copies.ml <dbl>, consumes_alcohol <fct>,
## # alcohol_units.week <int>, TB_current <fct>,
## # pyridoxine_prophylaxis <fct>, rifafour_treatment <fct>,
## # ARV_regimen <fct>, sn_present <ord>

# Data structure
glimpse(data)
```

```
## Observations: 120
## Variables: 17
## $ ID <chr> "001", "002", "003", "004", "005", "006...
## $ visit_number <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ visit_day <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ age_years <dbl> 59, 23, 27, 26, 37, 34, 44, 34, 32, 29,...
## $ mass_kg <dbl> 41.4, 70.2, 75.0, 68.8, 107.0, 85.5, 12...
## $ height_m <dbl> 1.56, 1.56, 1.64, 1.74, 1.60, 1.53, 1.6...
## $ sex <fct> F, F, M, M, F, F, F, F, F, M, M, M, M, ...
## $ hivsn_present <fct> no, no, no, no, no, no, no, no, no, no, no,...
## $ CD4_cell.ul <dbl> 35, 285, 28, 270, 310, 247, 439, 311, 1...
## $ viral_load_copies.ml <dbl> 6.103804, 5.041393, 5.181844, 2.484300,...
## $ consumes_alcohol <fct> no, no, no, no, yes, no, no, no, no, no,...
## $ alcohol_units.week <int> 0, 0, 0, 0, 15, 0, 0, 0, 0, 0, 0, 6, 9,...
## $ TB_current <fct> no, no, yes, no, no, no, no, no, no, no, no...
## $ pyridoxine_prophylaxis <fct> no, no, yes, no, no, no, no, no, no, no...
## $ rifafour_treatment <fct> no, no, yes, no, no, no, no, no, no, no...
## $ ARV_regimen <fct> TDF_FTC_EFV, TDF_FTC_EFV, TDF_FTC_EFV, ...
## $ sn_present <ord> no, no, no, no, no, yes, yes, no, no, n...
```

---

## Analyses

### Age

*# Tabular summary*

```
data %>%
  select(age_years, sn_present) %>%
  group_by(sn_present) %>%
  skim()
```

```
## Skim summary statistics
## n obs: 120
## n variables: 2
## group variables: sn_present
##
```

```
## -- Variable type:numeric -----
## sn_present variable missing complete n mean sd p0 p25 p50 p75
## yes age_years 0 20 20 41.8 8.47 26 36.5 40.5 46.75
## no age_years 0 100 100 36.96 9.36 21 29.75 36 43
## p100 hist
## 59
## 59
```

*# 95% bootstrap confidence interval of the mean age by SN status*

```
## Method = BCa, Resamples = 1999
set.seed(1234)
groupwiseMean(age_years ~ sn_present,
  data = data,
  R = 1999,
  traditional = FALSE,
  boot = TRUE,
  bca = TRUE)[c(1:3, 5, 6, 7)]
```

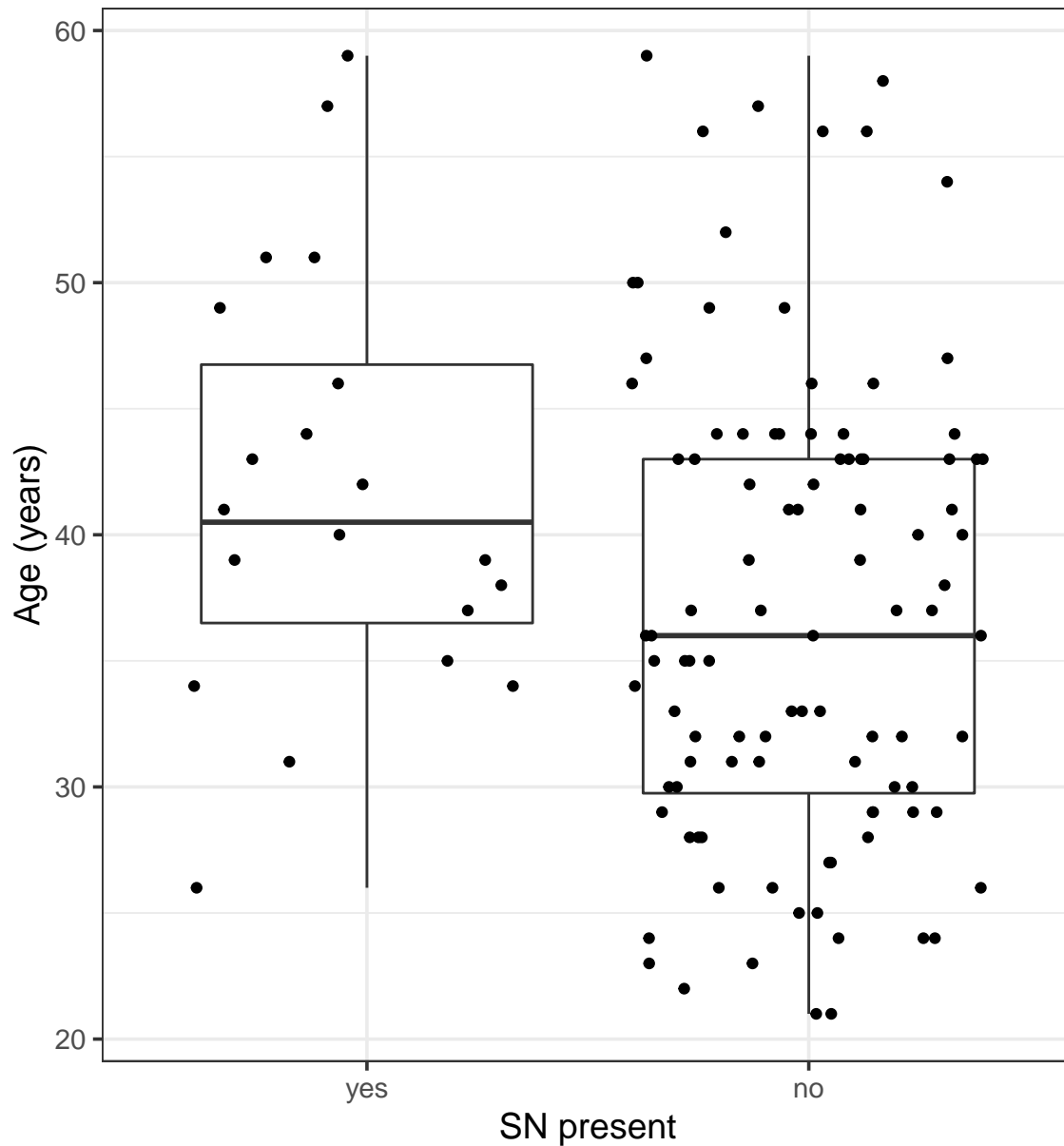
##	sn_present	n	Mean	Conf.level	Bca.lower	Bca.upper
## 1	yes	20	41.8	0.95	38.5	45.4
## 2	no	100	37.0	0.95	35.2	38.8

*# Plot*

```
data %>%
  ggplot(data = .) +
  aes(y = age_years,
       x = sn_present) +
  geom_boxplot() +
  geom_jitter(height = 0) +
  labs(title = 'Age at recruitment',
       subtitle = 'SN:yes vs SN:no',
       y = 'Age (years)',
       x = 'SN present')
```

## Age at recruitment

SN:yes vs SN:no



```
# Bootstrap 95% CI for the difference in mean age
## Resample: 1999
set.seed(1234)
boot_age <- boot.ci(boot(data = data,
                        statistic = boot_deltaMean,
                        data_column = 'age_years',
                        grouping_column = 'sn_present',
                        R = 1999,
                        stype = 'i'),
                  type = 'bca') %>%
  tibble(n = nrow(filter(data, !is.na(age_years))),
        Mean.difference = round(.$t0, 3),
```

```

    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 0 & Bca.upper >= 0,
                           yes = 'yes',
                           no = 'no')) %>%

.[1, -1] %>%
as.data.frame(); boot_age

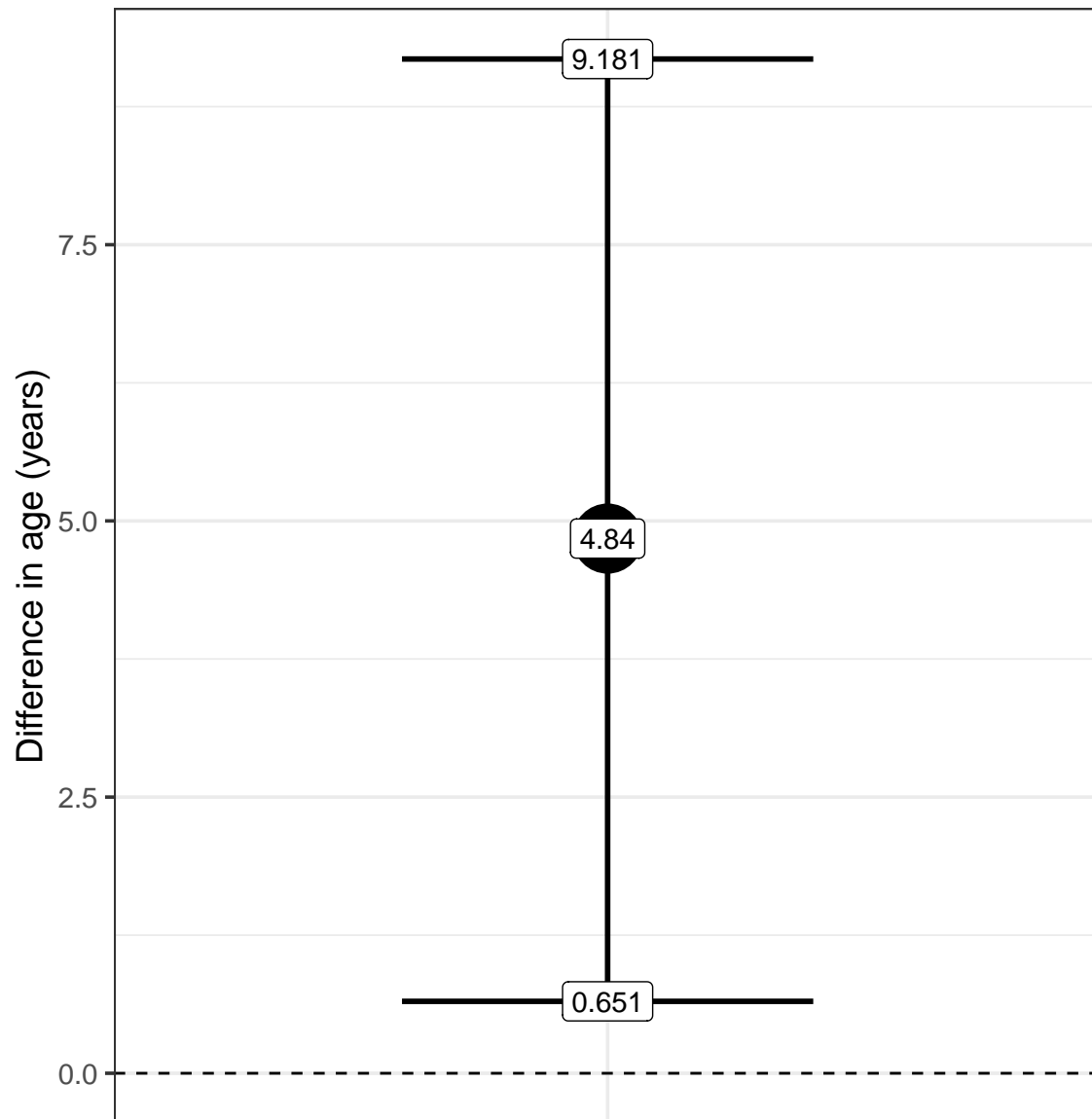
##      n Mean.difference Conf.level Bca.lower Bca.upper Includes.zero
## 1 120          4.84      0.95      0.651      9.181             no

ggplot(data = boot_age) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Mean.difference),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Mean.difference,
                 label = Mean.difference)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Bootstrap 95% CI of the difference between mean age',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = 'Difference in age (years)') +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```



Bootstrap 95% CI of the difference between mean a  
SN:yes vs SN:no  
Interval type: BCa, Resamples: 1999



Body mass

```
# Tabular summary
data %>%
  select(mass_kg, sn_present) %>%
  group_by(sn_present) %>%
  skim()

## Skim summary statistics
##   n obs: 120
##   n variables: 2
##   group variables: sn_present
```

```
##
## -- Variable type:numeric -----
## sn_present variable missing complete  n mean  sd  p0  p25  p50
##      yes mass_kg      0      20 20 72.97 18.05 47.6 59.48 69.45
##      no  mass_kg      0     100 100 63.43 13.5  41.4 53.98 62.1
##      p75 p100      hist
## 84.15 121.4
## 71.58 107

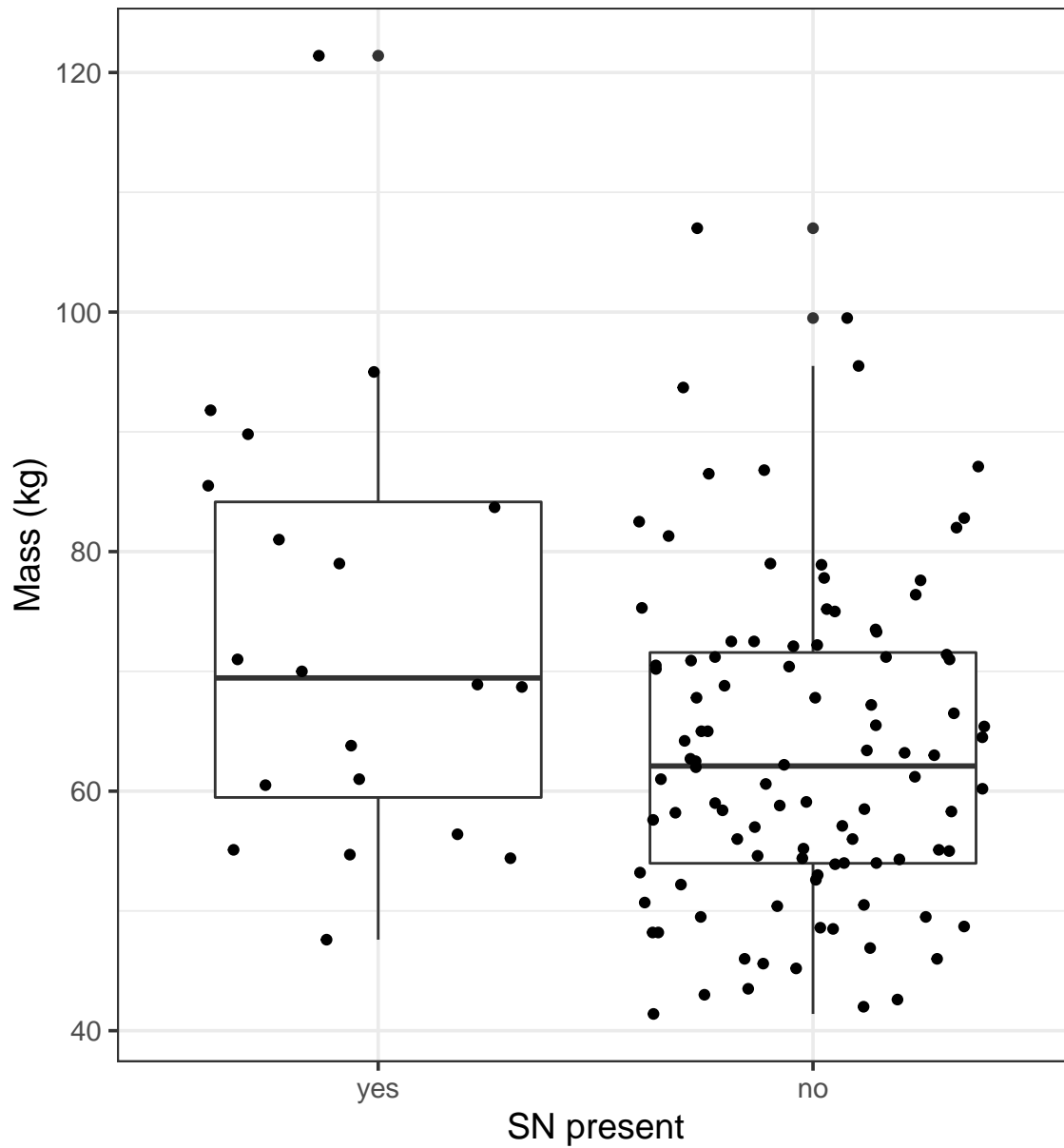
# 95% bootstrap confidence interval of the mean body mass by SN status
## Method = BCa, Resamples = 1999
set.seed(1234)
groupwiseMean(mass_kg ~ sn_present,
              data = data,
              R = 1999,
              traditional = FALSE,
              boot = TRUE,
              bca = TRUE)[c(1:3, 5, 6, 7)]

## sn_present  n Mean Conf.level Bca.lower Bca.upper
## 1      yes  20 73.0      0.95      65.9      82.1
## 2      no 100 63.4      0.95      61.0      66.2

# Plot
data %>%
  ggplot(data = .) +
  aes(y = mass_kg,
       x = sn_present) +
  geom_boxplot() +
  geom_jitter(height = 0) +
  labs(title = 'Mass at recruitment',
       subtitle = "SN:yes vs SN:no",
       y = 'Mass (kg)',
       x = 'SN present')
```

## Mass at recruitment

SN:yes vs SN:no



```
# Bootstrap 95% CI for the difference in mean mass
## Resample: 1999
set.seed(1234)
boot_mass <- boot.ci(boot(data = data,
  statistic = boot_deltaMean,
  data_column = 'mass_kg',
  grouping_column = 'sn_present',
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(n = nrow(filter(data, !is.na(mass_kg))),
    Mean.difference = round(.$t0, 3),
```

```

    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 0 & Bca.upper >= 0,
                           yes = 'yes',
                           no = 'no')) %>%

.[1, -1] %>%
as.data.frame(); boot_mass

##      n Mean.difference Conf.level Bca.lower Bca.upper Includes.zero
## 1 120          9.535         0.95      1.946      18.972           no

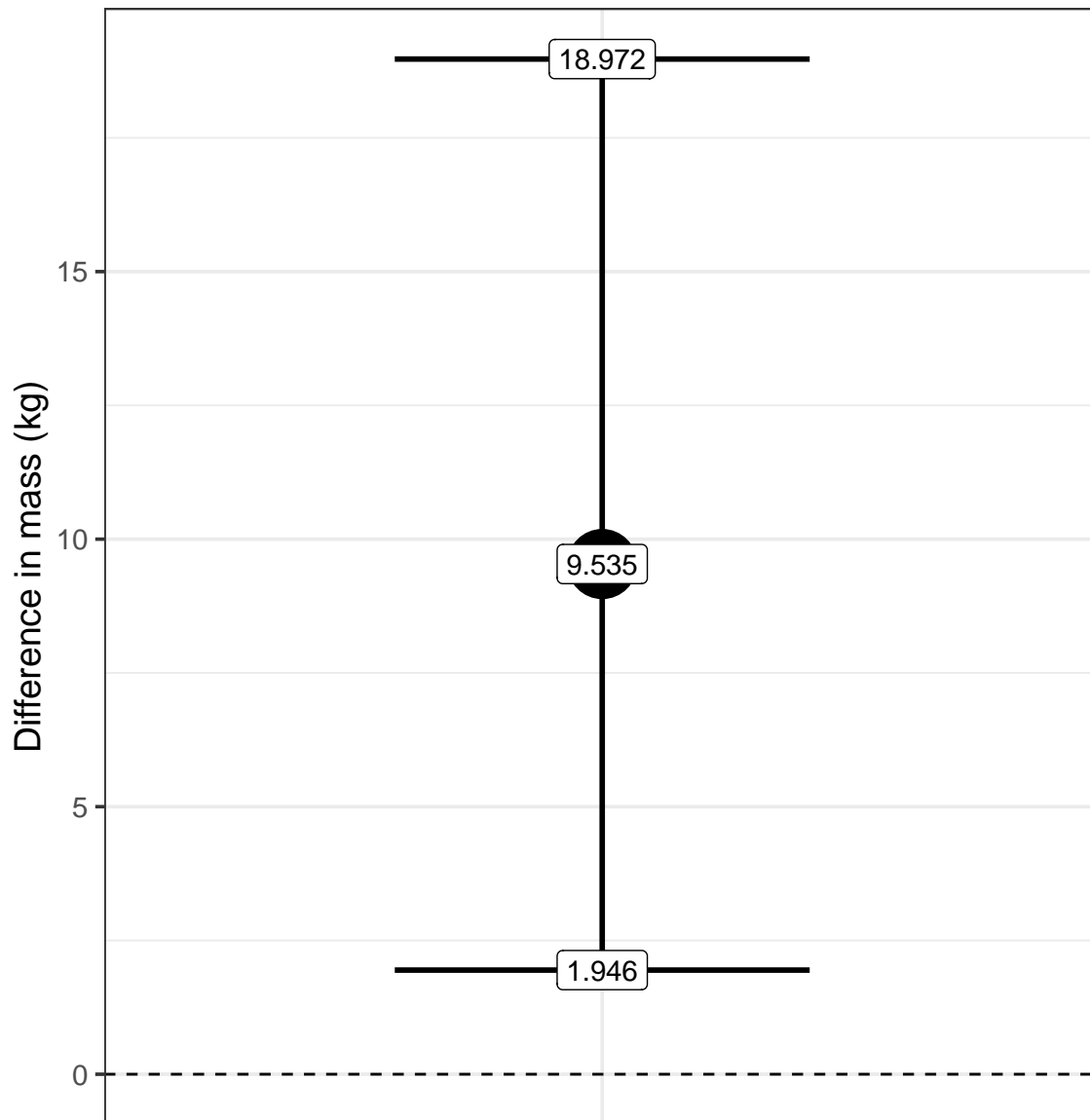
ggplot(data = boot_mass) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Mean.difference),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Mean.difference,
                 label = Mean.difference)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Bootstrap 95% CI of the difference in mean mass',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = 'Difference in mass (kg)') +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```

## Bootstrap 95% CI of the difference in mean mass

SN:yes vs SN:no

Interval type: BCa, Resamples: 1999



## Height

Expect height to show sex difference, so analyse separately for males and females.

*# Tabular summary*

```
data %>%  
  select(height_m, sex, sn_present) %>%  
  group_by(sn_present, sex) %>%  
  skim()
```

```
## Skim summary statistics
```

```
##   n obs: 120
```

```
## n variables: 3
## group variables: sn_present, sex
##
## -- Variable type:numeric -----
## sn_present sex variable missing complete n mean sd p0 p25 p50
## yes F height_m 0 8 8 1.64 0.087 1.53 1.58 1.66
## yes M height_m 0 12 12 1.72 0.045 1.65 1.69 1.73
## no F height_m 0 58 58 1.57 0.056 1.46 1.53 1.57
## no M height_m 0 42 42 1.7 0.051 1.58 1.66 1.7
## p75 p100 hist
## 1.7 1.76
## 1.75 1.79
## 1.62 1.68
## 1.72 1.82
```

*# 95% bootstrap confidence interval of the mean height by SN status*

```
## Method = BCa, Resamples = 1999
```

```
set.seed(1234)
```

```
groupwiseMean(height_m ~ sn_present + sex,
  data = data,
  R = 1999,
  traditional = FALSE,
  boot = TRUE,
  bca = TRUE)[c(1:3, 5, 6:8)]
```

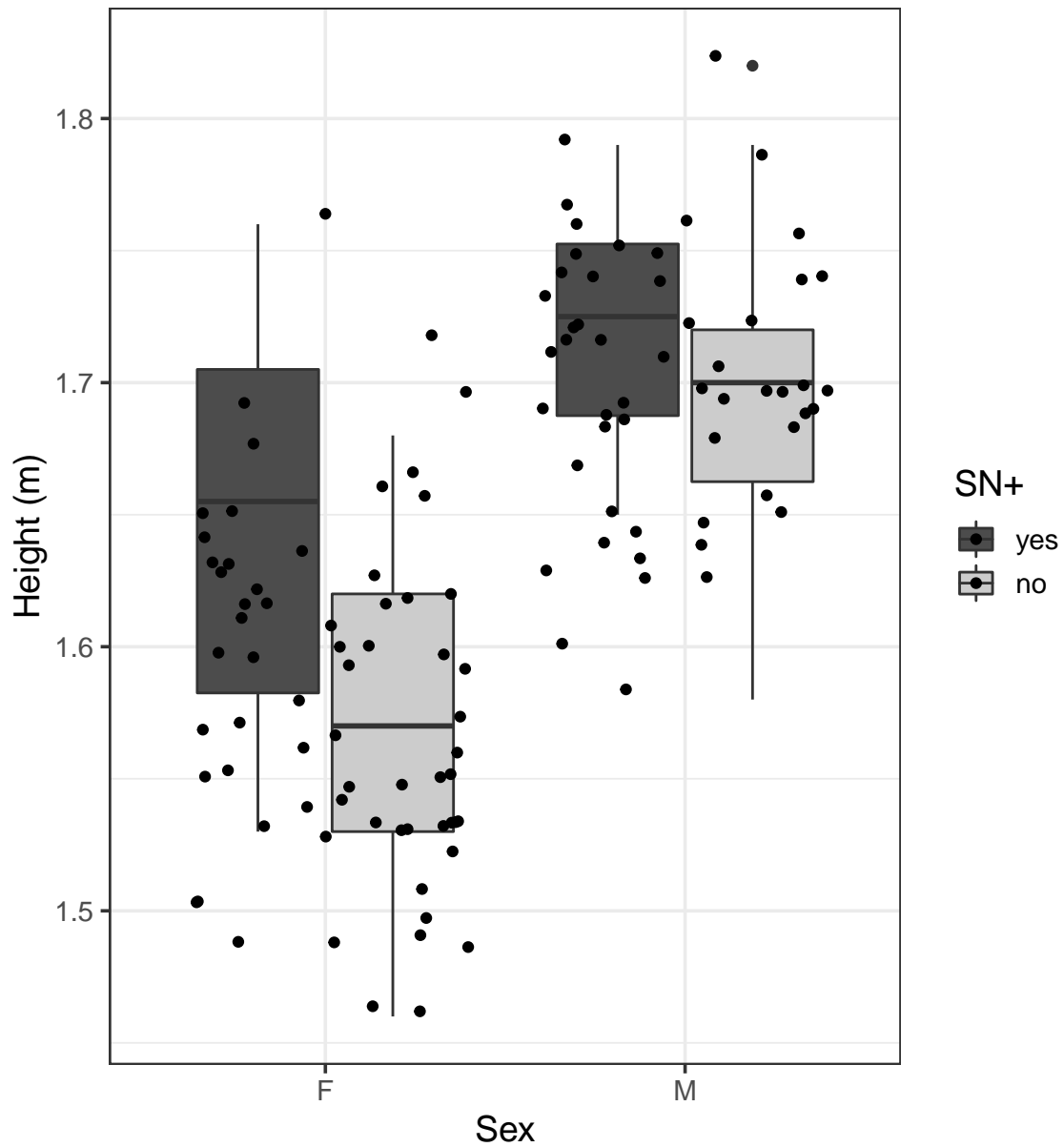
```
## sn_present sex n Boot.mean Conf.level Bca.lower Bca.upper
## 1 yes F 8 1.64 0.95 1.58 1.69
## 2 yes M 12 1.72 0.95 1.69 1.74
## 3 no F 58 1.57 0.95 1.56 1.58
## 4 no M 42 1.70 0.95 1.68 1.71
```

*# Plots*

```
data %>%
  ggplot(data = .) +
  aes(y = height_m,
    x = sex,
    fill = sn_present) +
  geom_boxplot() +
  geom_jitter() +
  scale_fill_manual(name = 'SN+',
    values = c('#4C4C4C', '#CCCCCC')) +
  labs(title = 'Height at recruitment',
    subtitle = "SN:yes vs SN:no",
    y = 'Height (m)',
    x = 'Sex')
```

## Height at recruitment

SN:yes vs SN:no



```
# MALES ONLY
```

```
## Bootstrap 95% CI for the difference in mean height
```

```
### Resample: 1999
```

```
set.seed(1234)
```

```
boot_hm <- boot.ci(boot(data = data[data$sex == 'M', ],
  statistic = boot_deltaMean,
  data_column = 'height_m',
  grouping_column = 'sn_present',
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(n = nrow(filter(data, !is.na(height_m) & sex == 'M'))),
```

```

    Mean.difference = round(.$t0, 3),
    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 0 & Bca.upper >= 0,
                           yes = 'yes',
                           no = 'no')) %>%

.[1, -1] %>%
as.data.frame(); boot_hm

##      n Mean.difference Conf.level Bca.lower Bca.upper Includes.zero
## 1 54          0.025      0.95    -0.005     0.054             yes

ggplot(data = boot_hm) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Mean.difference),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Mean.difference,
                 label = Mean.difference)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Males: Bootstrap 95% CI of the difference in mean height',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = 'Difference in height (m)') +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

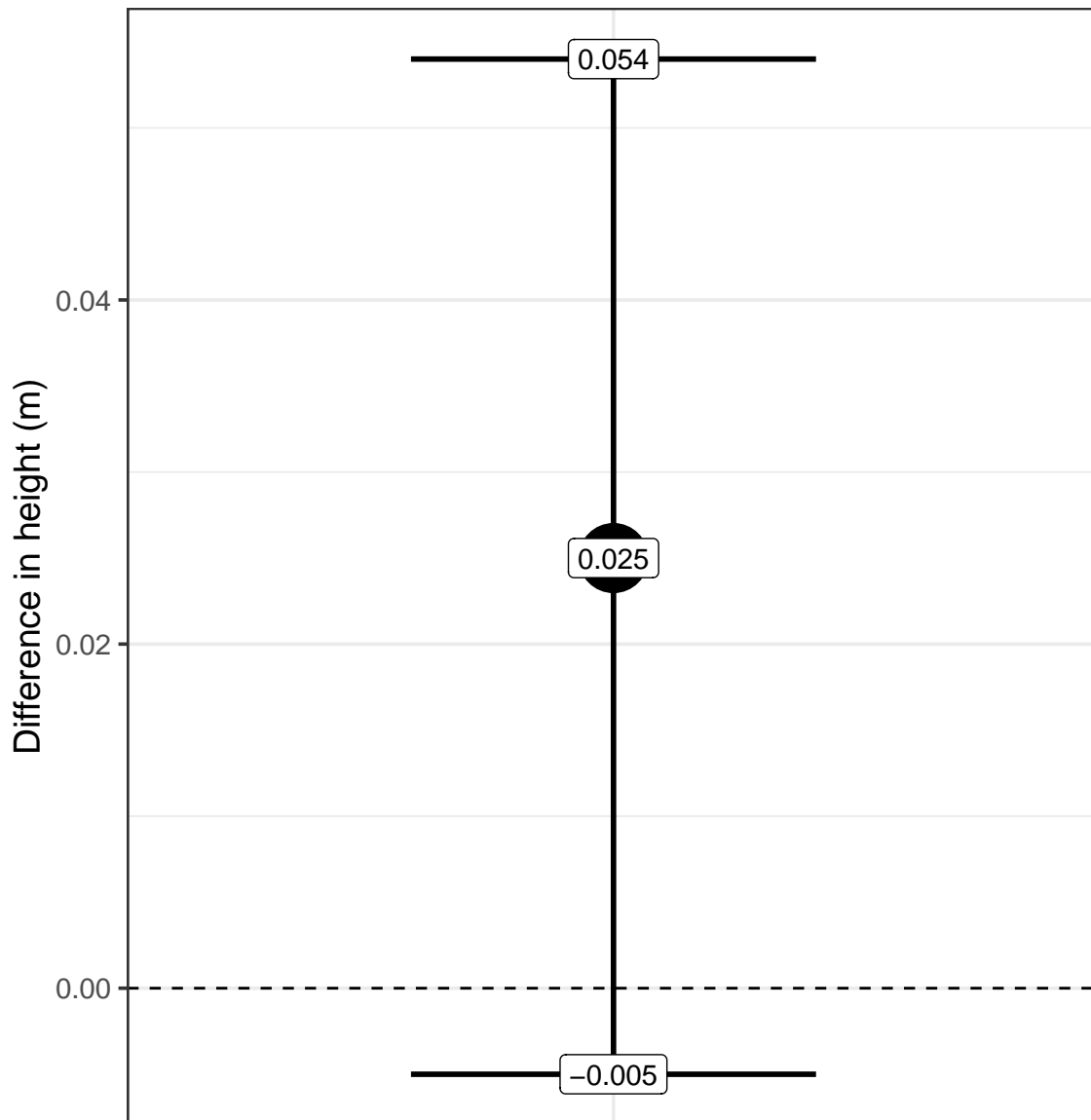
```



## Males: Bootstrap 95% CI of the difference in mean

SN:yes vs SN:no

Interval type: BCa, Resamples: 1999



```
# FEMALES ONLY
```

```
## Bootstrap 95% CI for the difference in mean height
```

```
### Resample: 9000 (1999 gives a wierd error)
```

```
set.seed(1234)
```

```
boot_hf <- boot.ci(boot(data = data[data$sex == 'F', ],  
  statistic = boot_deltaMean,  
  data_column = 'height_m',  
  grouping_column = 'sn_present',  
  R = 1999,  
  stype = 'i'),  
  type = 'bca') %>%  
  tibble(n = nrow(filter(data, !is.na(height_m) & sex == 'F'))),
```

```

    Mean.difference = round(.$t0, 3),
    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 0 & Bca.upper >= 0,
                           yes = 'yes',
                           no = 'no')) %>%

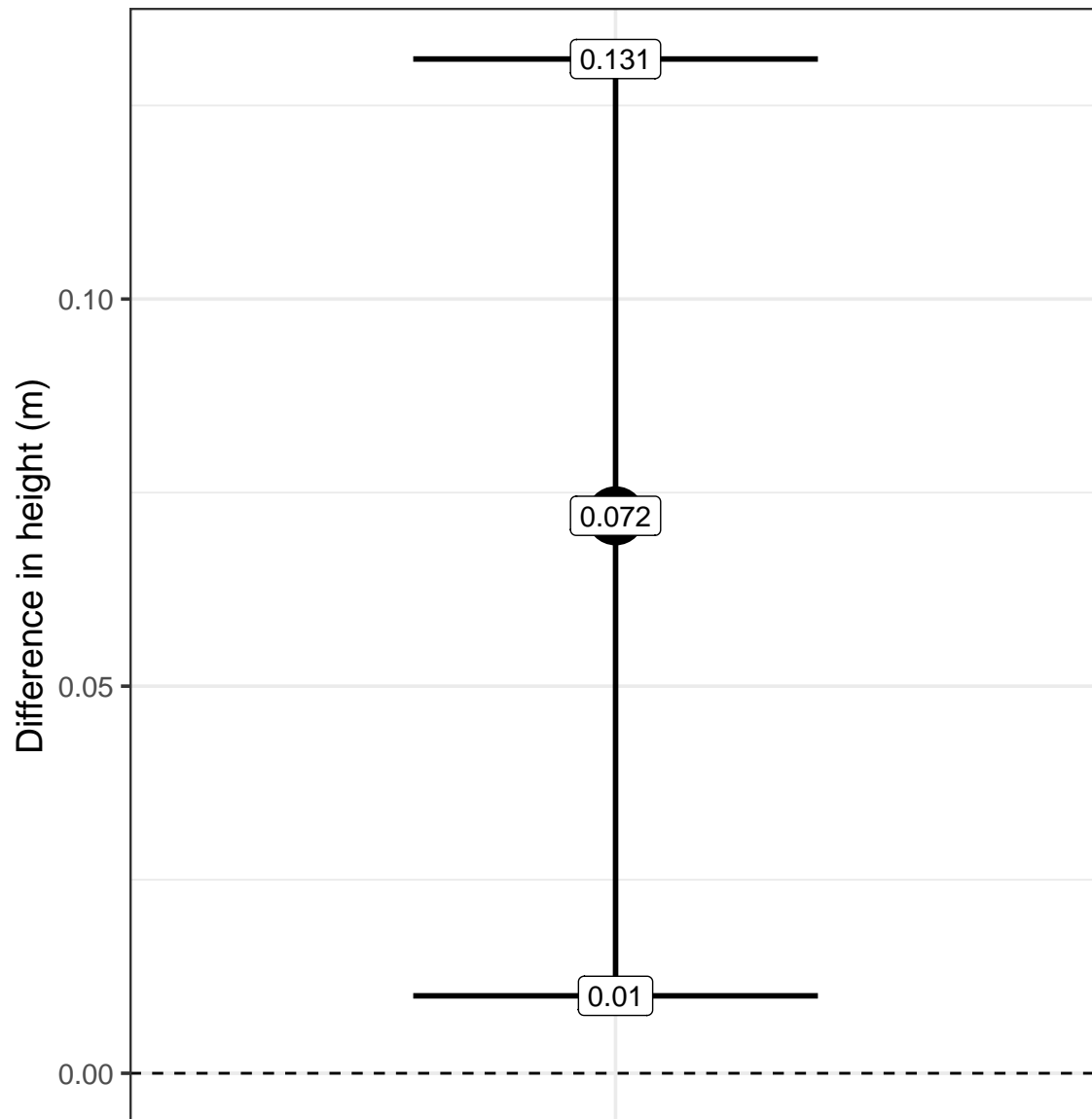
.[1, -1] %>%
as.data.frame(); boot_hf

##      n Mean.difference Conf.level Bca.lower Bca.upper Includes.zero
## 1 66          0.072      0.95      0.01      0.131             no

ggplot(data = boot_hf) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Mean.difference),
             size = 10) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Mean.difference,
                 label = Mean.difference)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Females: Bootstrap 95% CI of the difference in mean height',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = 'Difference in height (m)') +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```

Females: Bootstrap 95% CI of the difference in me  
SN:yes vs SN:no  
Interval type: BCa, Resamples: 1999



Sex

```
# Tabular summary
data %>%
  select(sex, sn_present) %>%
  group_by(sn_present) %>%
  skim()

## Skim summary statistics
##   n obs: 120
##   n variables: 2
##   group variables: sn_present
```

```
##
## -- Variable type:factor -----
## sn_present variable missing complete  n n_unique      top_counts
##      yes      sex      0      20 20      2 M: 12, F: 8, NA: 0
##      no      sex      0     100 100      2 F: 58, M: 42, NA: 0
## ordered
##      FALSE
##      FALSE

# 95% bootstrap confidence interval of the proportion of females by SN status
## Method = BCa, Resamples = 1999
### SN:yes
set.seed(1234)
sn_yes <- boot.ci(boot(data = data[data$sn_present == 'yes', ],
  statistic = function(d, i){
    mean(d[i, 'sex'] == 'F')},
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(sn_present = 'yes',
    n = nrow(filter(data, !is.na(sex) &
      sn_present == 'yes')),
    Proportion = round(.$t0, 3),
    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3)) %>%
  .[1, -1] %>%
  as.data.frame()

### SN:no
set.seed(1234)
sn_no <- boot.ci(boot(data = data[data$sn_present == 'no', ],
  statistic = function(d, i){
    mean(d[i, 'sex'] == 'F')},
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(sn_present = 'no',
    n = nrow(filter(data, !is.na(sex) &
      sn_present == 'yes')),
    Proportion = round(.$t0, 3),
    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3)) %>%
  .[1, -1] %>%
  as.data.frame()

### Put sn and sn_no together and print
sn_yes %>%
  bind_rows(sn_no)

## sn_present  n Proportion Conf.level Bca.lower Bca.upper
## 1      yes 20      0.40      0.95      0.15      0.60
## 2      no 20      0.58      0.95      0.47      0.66
```

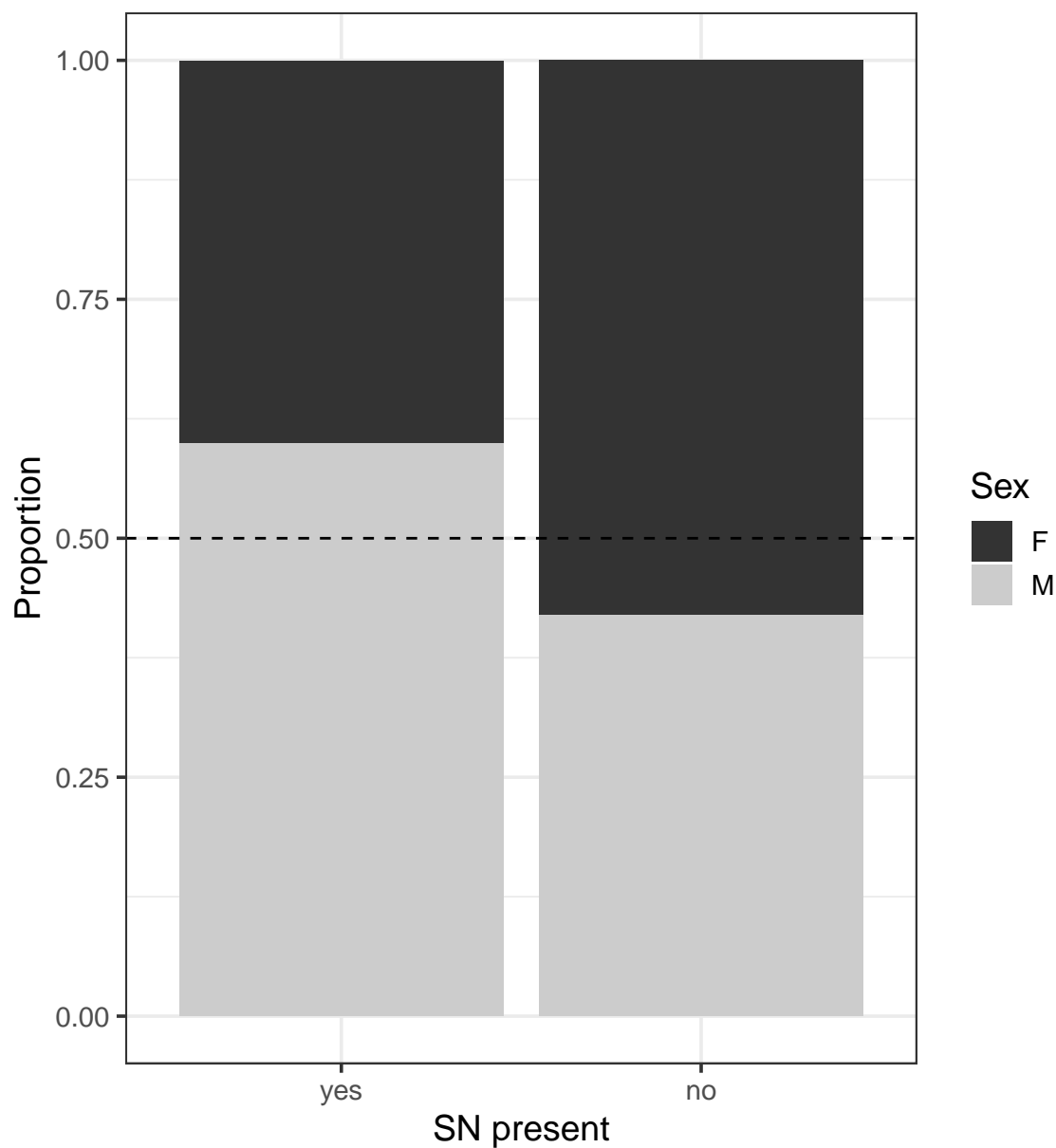
```

# Plot
data %>%
  ggplot(data = .) +
  aes(x = sn_present,
      fill = sex) +
  geom_bar(position = 'fill') +
  geom_hline(yintercept = 0.5,
             linetype = 2) +
  labs(title = 'Sex ratio at recruitment',
       subtitle = "SN:yes vs SN:no",
       y = 'Proportion',
       x = 'SN present') +
  scale_fill_grey(name = 'Sex')

```

## Sex ratio at recruitment

SN:yes vs SN:no



```
# Bootstrap 95% CI for the odds ratio of SN:yes vs SN:no
## Resample: 1999
set.seed(1234)
boot_sex <- boot.ci(boot(data = data,
  statistic = boot_OR,
  data_column = 'sex',
  grouping_column = 'sn_present',
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(n = nrow(filter(data, !is.na(sex))),
    Odds.ratio = round(.$t0, 3),
```

```

    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.one = ifelse(Bca.lower <= 1 & Bca.upper >= 1,
                          yes = 'yes',
                          no = 'no')) %>%

.[1, -1] %>%
as.data.frame(); boot_sex

##      n Odds.ratio Conf.level Bca.lower Bca.upper Includes.one
## 1 120      0.486      0.95      0.15      1.374          yes

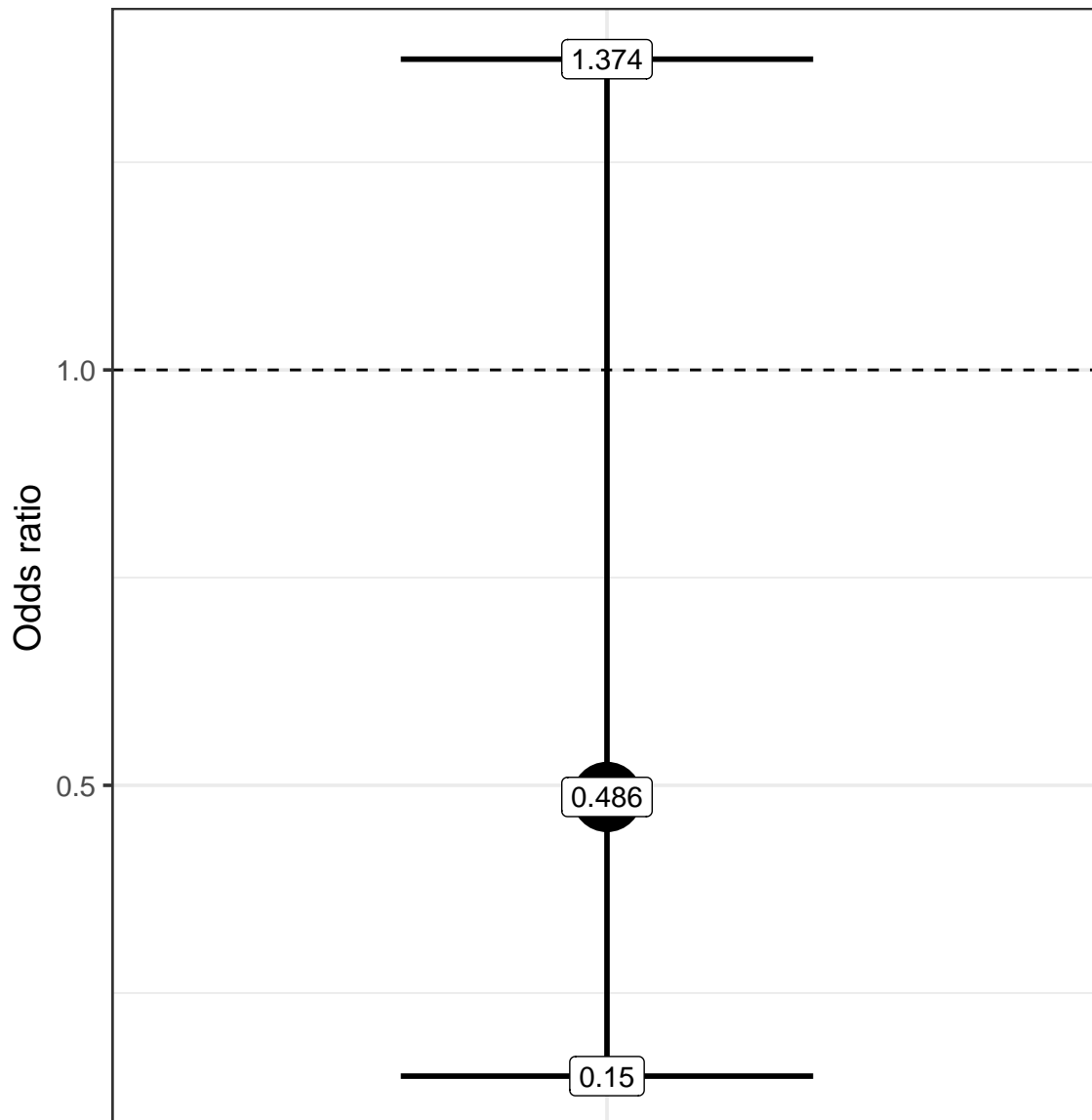
ggplot(data = boot_sex) +
  geom_hline(yintercept = 1,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Odds.ratio),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Odds.ratio,
                 label = Odds.ratio)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Bootstrap 95% CI of the odds ratio for being female',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = 'Odds ratio') +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```

## Bootstrap 95% CI of the odds ratio for being female

SN:yes vs SN:no

Interval type: BCa, Resamples: 1999



## CD4 T-cell count

```
# Tabular summary
data %>%
  select(CD4_cell.ul, sn_present) %>%
  group_by(sn_present) %>%
  skim()

## Skim summary statistics
##   n obs: 120
##   n variables: 2
##   group variables: sn_present
```



```
##
## -- Variable type:numeric -----
## sn_present      variable missing complete   n   mean    sd p0   p25   p50
##      yes CD4_cell.ul      0      20  20 223.55 176.37  4   71.5 200.5
##      no  CD4_cell.ul      1     99 100 274.41 233.13  1  123   234
##      p75 p100      hist
## 308.25 673
## 345    1347

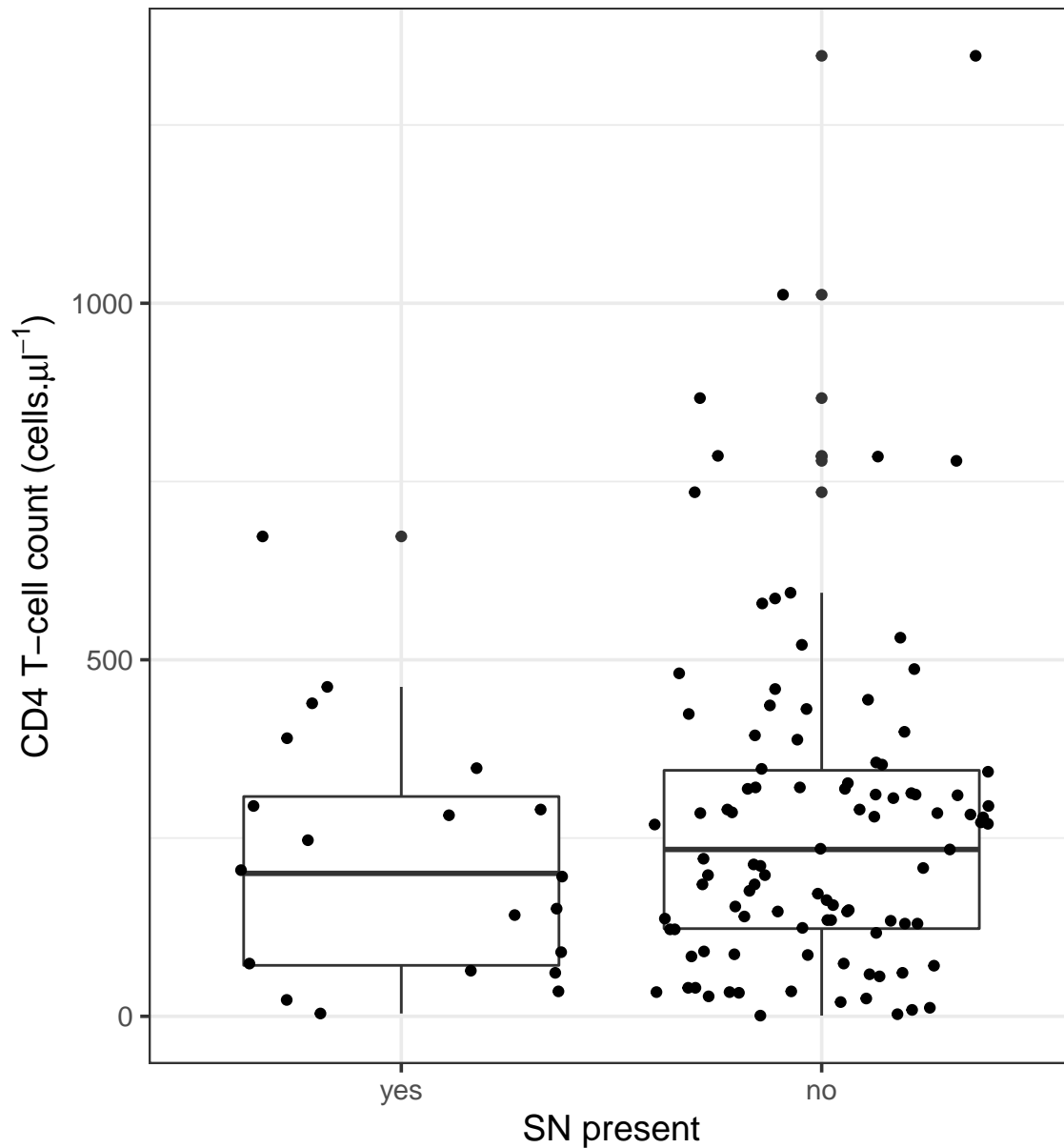
# 95% bootstrap confidence interval of the median CD4 T-cell count by SN status
## Method = BCa, Resamples = 1999
set.seed(1234)
groupwiseMedian(CD4_cell.ul ~ sn_present,
  data = data %>% filter(!is.na(CD4_cell.ul)),
  R = 1999,
  boot = TRUE,
  bca = TRUE)[c(1:3, 5, 6, 7)]

## sn_present   n Median Conf.level Bca.lower Bca.upper
## 1      yes 20   200      0.95      74      290
## 2      no 99   234      0.95     163     285

# Plot
data %>%
  ggplot(data = .) +
  aes(y = CD4_cell.ul,
      x = sn_present) +
  geom_boxplot() +
  geom_jitter(height = 0) +
  labs(title = 'CD4 T-cell count at recruitment',
       subtitle = "SN:yes vs SN:no",
       y = expression(paste('CD4 T-cell count (cells.', mu, 10^-1, ')')),
       x = 'SN present')
```

## CD4 T-cell count at recruitment

SN:yes vs SN:no



```
# Bootstrap 95% CI for the difference in median CD4 T-cell count
## Resample: 1999
set.seed(1234)
boot_cd4 <- boot.ci(boot(data = data,
  statistic = boot_deltaMedian,
  data_column = 'CD4_cell.ul',
  grouping_column = 'sn_present',
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(n = nrow(filter(data, !is.na(CD4_cell.ul))),
    Median.difference = round(.$t0, 3),
```

```

    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 0 & Bca.upper >= 0,
                           yes = 'yes',
                           no = 'no')) %>%

.[1, -1] %>%
as.data.frame(); boot_cd4

##      n Median.difference Conf.level Bca.lower Bca.upper Includes.zero
## 1 119          -33.5      0.95      -180        87             yes

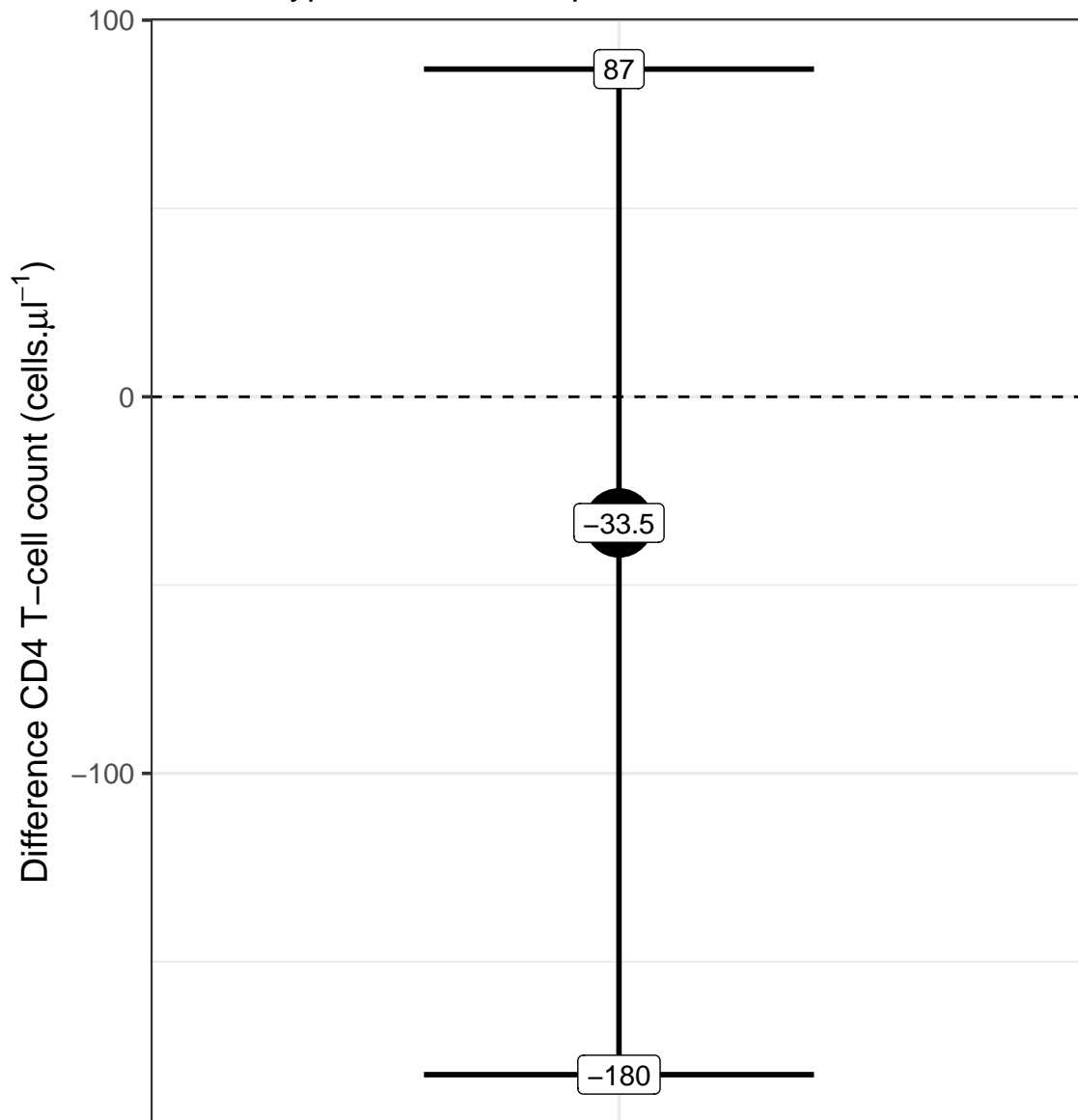
ggplot(data = boot_cd4) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Median.difference),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Median.difference,
                 label = Median.difference)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
labs(title = 'Bootstrap 95% CI of the difference in median CD4 T-cell count',
     subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
     y = expression(paste('Difference CD4 T-cell count (cells.', mu, l^{-1}, ')'))) +
theme(axis.ticks.x = element_blank(),
      axis.text.x = element_blank(),
      axis.title.x = element_blank())

```

## Bootstrap 95% CI of the difference in median CD4

SN:yes vs SN:no

Interval type: BCa, Resamples: 1999



## Viral load

```
# Tabular summary
data %>%
  select(viral_load_copies.ml, sn_present) %>%
  group_by(sn_present) %>%
  skim()

## Skim summary statistics
##   n obs: 120
##   n variables: 2
##   group variables: sn_present
```

```
##
## -- Variable type:numeric -----
## sn_present          variable missing complete   n mean   sd   p0 p25
##      yes viral_load_copies.ml           1      19  20 3.59 1.59 1.83 2.22
##      no viral_load_copies.ml           11     89 100 3.45 1.26 1.7  2.63
##    p50 p75 p100    hist
##    3   4.73 6.45
##    3.16 4.26 6.51

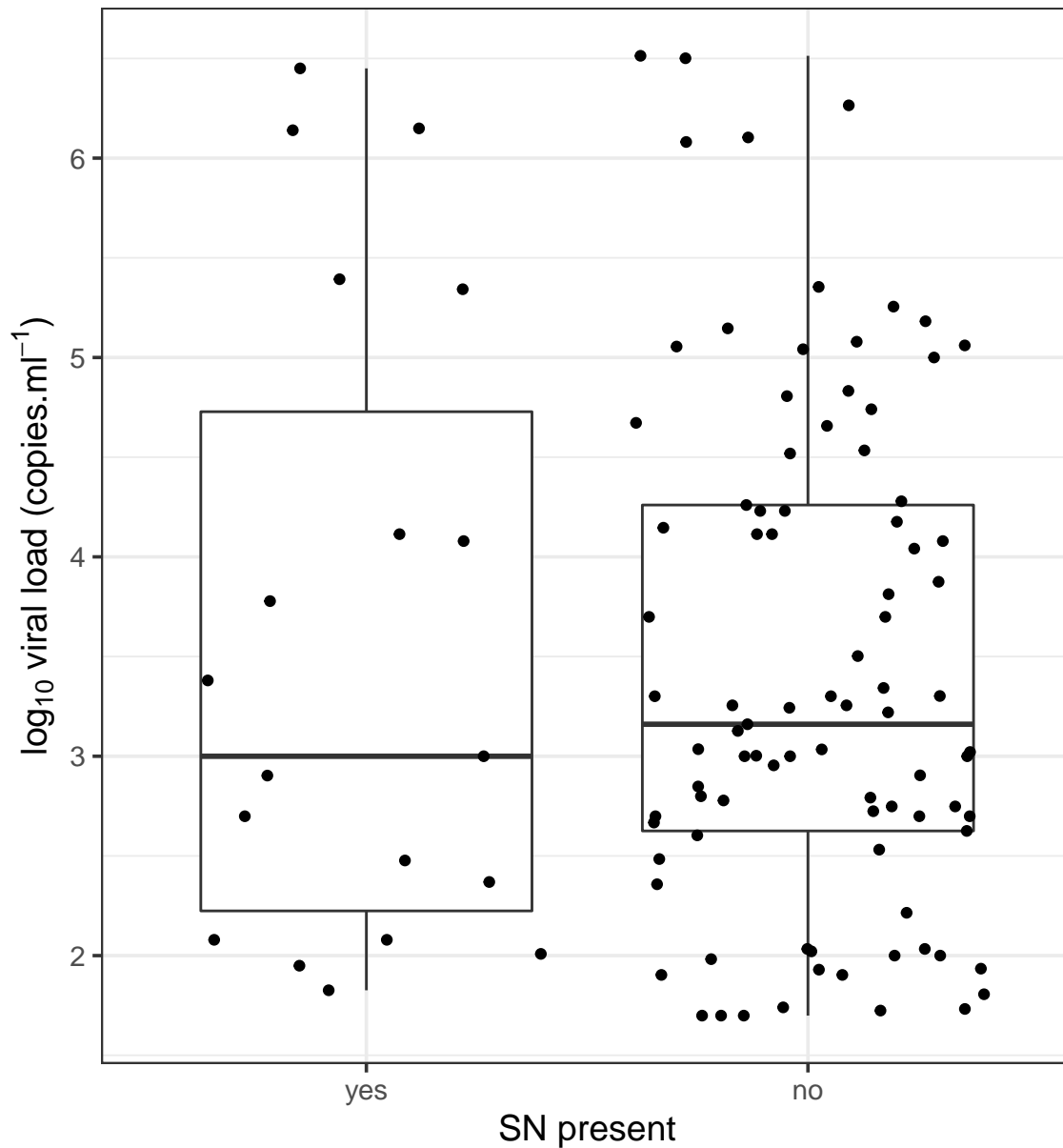
# 95% bootstrap confidence interval of the median viral load by SN status
## Method = BCa, Resamples = 1999
set.seed(1234)
groupwiseMedian(viral_load_copies.ml ~ sn_present,
                 data = data[!is.na(data$viral_load_copies.ml), ], # Remove <NA>
                 R = 1999,
                 boot = TRUE,
                 bca = TRUE)[c(1:3, 5, 6, 7)]

##    sn_present   n Median Conf.level Bca.lower Bca.upper
## 1      yes 19    3.00      0.95      2.08      4.08
## 2      no 89    3.16      0.95      2.90      3.34

# Plot
data %>%
  filter(!is.na(viral_load_copies.ml)) %>%
  ggplot(data = .) +
  aes(y = viral_load_copies.ml,
       x = sn_present) +
  geom_boxplot() +
  geom_jitter(height = 0) +
  labs(title = 'Viral load at recruitment',
       subtitle = 'SN:yes vs SN:no',
       x = 'SN present',
       y = expression(paste('log' [10], ' viral load (copies.ml' ^-1, ')')))
```

## Viral load at recruitment

SN:yes vs SN:no



```
# Bootstrap 95% CI for the difference in median viral load
## Resample: 1999
set.seed(1234)
boot_vl <- boot.ci(boot(data = data,
  statistic = boot_deltaMedian,
  data_column = 'viral_load_copies.ml',
  grouping_column = 'sn_present',
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(n = nrow(filter(data, !is.na(viral_load_copies.ml))),
    Median.difference = round(.$t0, 3),
```

```

    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 0 & Bca.upper >= 0,
                           yes = 'yes',
                           no = 'no')) %>%

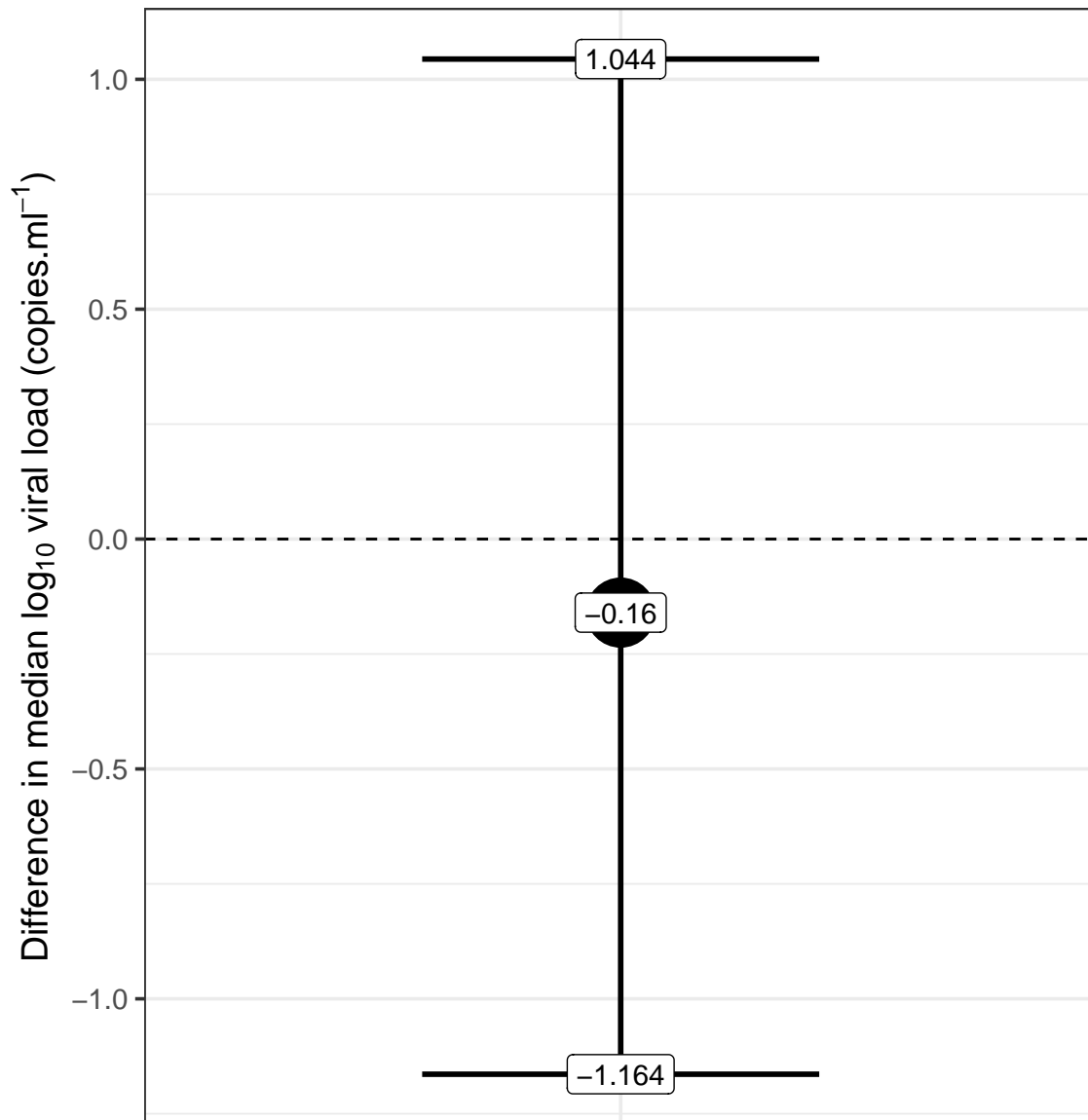
.[1, -1] %>%
as.data.frame(); boot_v1

##      n Median.difference Conf.level Bca.lower Bca.upper Includes.zero
## 1 108          -0.16         0.95    -1.164     1.044             yes

ggplot(data = boot_v1) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Median.difference),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Median.difference,
                 label = Median.difference)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Bootstrap 95% CI of the difference in median viral load',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = expression(paste('Difference in median log' [10], ' viral load (copies.ml' ^-1, ')')) +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```

Bootstrap 95% CI of the difference in median viral  
SN:yes vs SN:no  
Interval type: BCa, Resamples: 1999



## Alcohol

```
# Tabular summary
data %>%
  select(beer_units.week, sn_present) %>%
  mutate(drinks_alcohol = case_when(
    beer_units.week >= 1 ~ 'Yes',
    beer_units.week == 0 ~ 'No'
  )) %>%
  mutate(drinks_alcohol = factor(drinks_alcohol)) %>%
  group_by(sn_present, drinks_alcohol) %>%
```



```

skim()

## Skim summary statistics
## n obs: 120
## n variables: 3
## group variables: sn_present, drinks_alcohol
##
## -- Variable type:integer -----
## sn_present drinks_alcohol variable missing complete n mean
##      yes      No alcohol_units.week      0      18 18 0
##      yes      Yes alcohol_units.week      0       2 2 7.5
##      no      No alcohol_units.week      0      75 75 0
##      no      Yes alcohol_units.week      0      25 25 24.96
##      sd p0  p25  p50  p75 p100      hist
##      0   0  0   0   0   0
##      4.95 4 5.75 7.5 9.25 11
##      0   0  0   0   0   0
##      26   3  9   15  29   95

# ALCOHOL DRINKERS ONLY
## 95% bootstrap confidence interval of the median alcohol consumption by SN status
### Method = BCa, Resamples = 1999
set.seed(1234)
groupwiseMedian(alcohol_units.week ~ sn_present,
                 data = data[data$alcohol_units.week > 0, ], # Remove none drinkers
                 R = 100000, # Had to increase this one to avoid extreme zero-order stats
                 boot = TRUE,
                 bca = TRUE)[c(1:3, 5, 6, 7)]

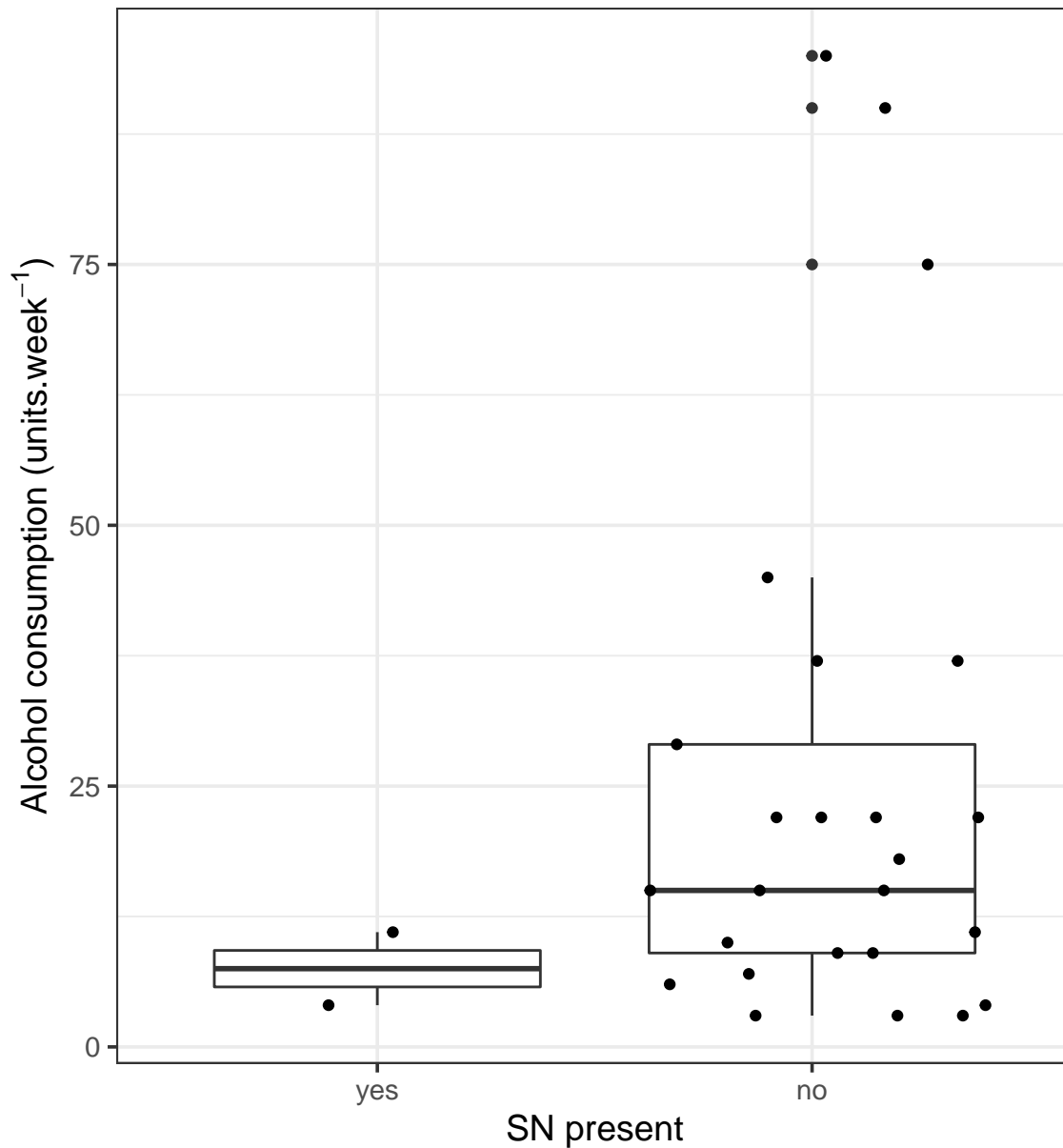
## sn_present n Median Conf.level Bca.lower Bca.upper
## 1      yes 2 7.5 0.95 4 7.5
## 2      no 25 15.0 0.95 4 15.0

# Plot
data %>%
  filter(alcohol_units.week > 0) %>%
  ggplot(data = .) +
  aes(x = sn_present,
      y = alcohol_units.week) +
  geom_boxplot() +
  geom_jitter(height = 0) +
  labs(title = 'Alcohol consumption at recruitment',
       subtitle = 'SN:yes vs SN:no',
       x = 'SN present',
       y = expression(paste('Alcohol consumption (units.week' ^-1, ')'))))

```

## Alcohol consumption at recruitment

SN:yes vs SN:no



*# With only two drinkers in the SN:yes group, we did not pursue additional exploration.*

## TB

*Note: Treatment policy was to start some patients, irrespective of TB diagnosis, on TB treatment. Therefore current TB infection and treatment for TB analysed separately.*

## Currently infected with TB

*# Tabular summary*

*data %>%*

```

select(TB_current, sn_present) %>%
group_by(sn_present) %>%
skim()

## Skim summary statistics
##   n obs: 120
##   n variables: 2
##   group variables: sn_present
##
## -- Variable type:factor -----
##   sn_present   variable missing complete   n n_unique
##         yes TB_current         0        20  20         2
##         no  TB_current         0       100 100         2
##           top_counts ordered
##   no: 11, yes: 9, NA: 0   FALSE
##   no: 89, yes: 11, NA: 0  FALSE

# 95% bootstrap confidence interval of the proportion with current TB by SN status
## Method = BCa, Resamples = 1999
### SN:yes
set.seed(1234)
sn_yes <- boot.ci(boot(data = data[data$sn_present == 'yes', ],
                      statistic = function(d, i){
                        mean(d[i, 'TB_current'] == 'yes')},
                      R = 1999,
                      stype = 'i'),
                 type = 'bca') %>%
tibble(sn_present = 'yes',
       n = nrow(filter(data, !is.na(TB_current) &
                        sn_present == 'yes')),
       Proportion = round(.$t0, 3),
       Conf.level = 0.95,
       Bca.lower = round(.$bca[[4]], 3),
       Bca.upper = round(.$bca[[5]], 3)) %>%
.[1, -1] %>%
as.data.frame()

### SN:no
set.seed(1234)
sn_no <- boot.ci(boot(data = data[data$sn_present == 'no', ],
                      statistic = function(d, i){
                        mean(d[i, 'TB_current'] == 'yes')},
                      R = 1999,
                      stype = 'i'),
                 type = 'bca') %>%
tibble(sn_present = 'no',
       n = nrow(filter(data, !is.na(TB_current) &
                        sn_present == 'no')),
       Proportion = round(.$t0, 3),
       Conf.level = 0.95,
       Bca.lower = round(.$bca[[4]], 3),
       Bca.upper = round(.$bca[[5]], 3)) %>%
.[1, -1] %>%
as.data.frame()

```

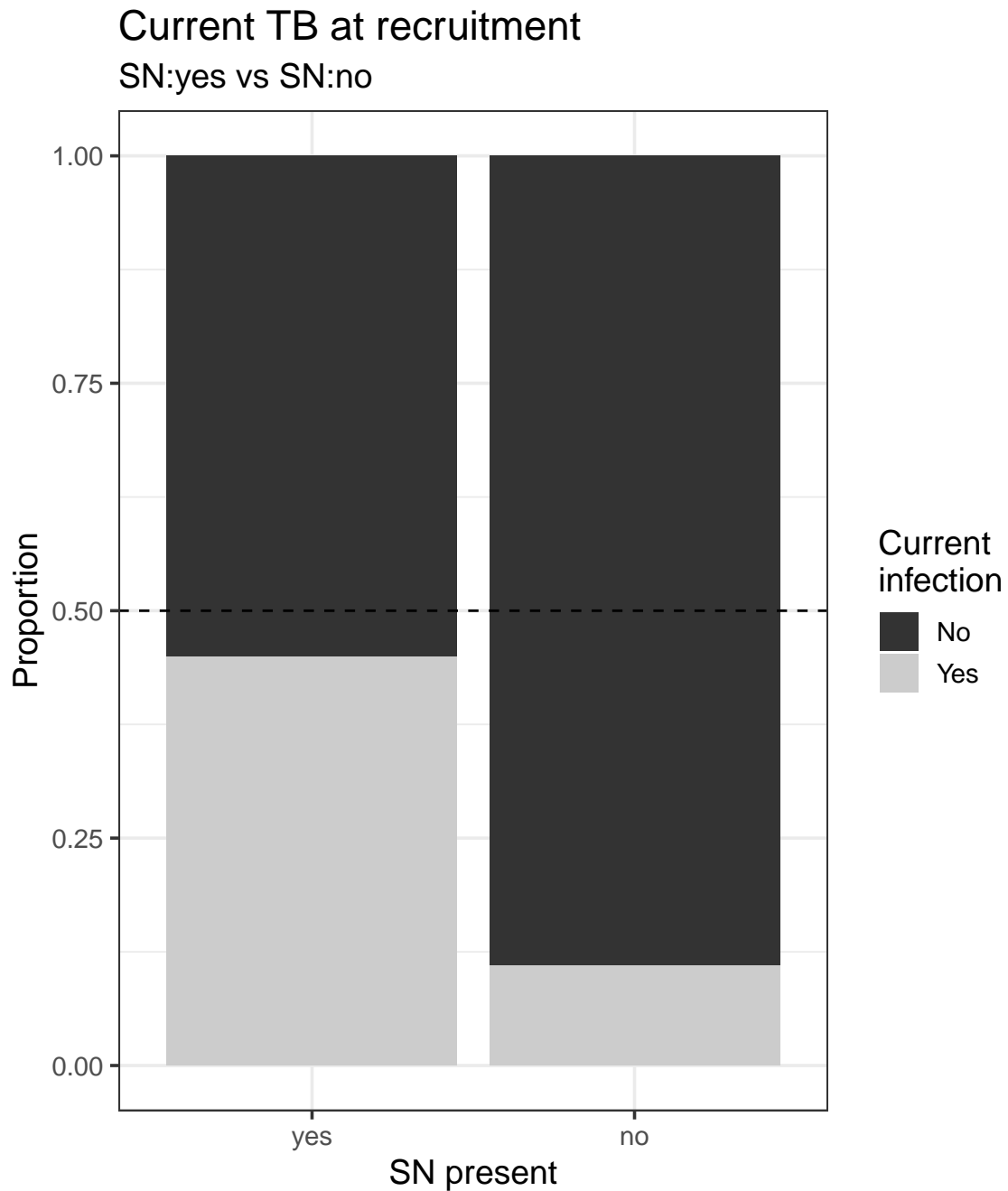
```

### Put sn and sn_no together and print
sn_yes %>%
  bind_rows(sn_no)

##   sn_present    n Proportion Conf.level Bca.lower Bca.upper
## 1         yes   20         0.45         0.95         0.20         0.60
## 2         no  100         0.11         0.95         0.05         0.17

# Plot
data %>%
  mutate(TB_current = str_to_title(TB_current)) %>%
  ggplot(data = .) +
  aes(x = sn_present,
      fill = TB_current) +
  geom_bar(position = 'fill') +
  geom_hline(yintercept = 0.5,
            linetype = 2) +
  labs(title = 'Current TB at recruitment',
       subtitle = 'SN:yes vs SN:no',
       x = 'SN present',
       y = 'Proportion') +
  scale_fill_grey(name = 'Current\\ninfection')

```



```
# Bootstrap 95% CI for the odds ratio of current TB in SN:yes vs SN:no
## Resample: 1999
set.seed(1234)
boot_tb <- boot.ci(boot(data = data,
                        statistic = boot_OR,
                        data_column = 'TB_current',
                        grouping_column = 'sn_present',
                        R = 1999,
                        stype = 'i'),
                  type = 'bca') %>%
  tibble(n = nrow(filter(data, !is.na(TB_current))),
         Odds.ratio = round(.$t0, 3),
```

```

    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 1 & Bca.upper >= 1,
                           yes = 'yes',
                           no = 'no')) %>%

.[1, -1] %>%
as.data.frame(); boot_tb

##      n Odds.ratio Conf.level Bca.lower Bca.upper Includes.zero
## 1 120      0.155      0.95      0.051      0.47              no

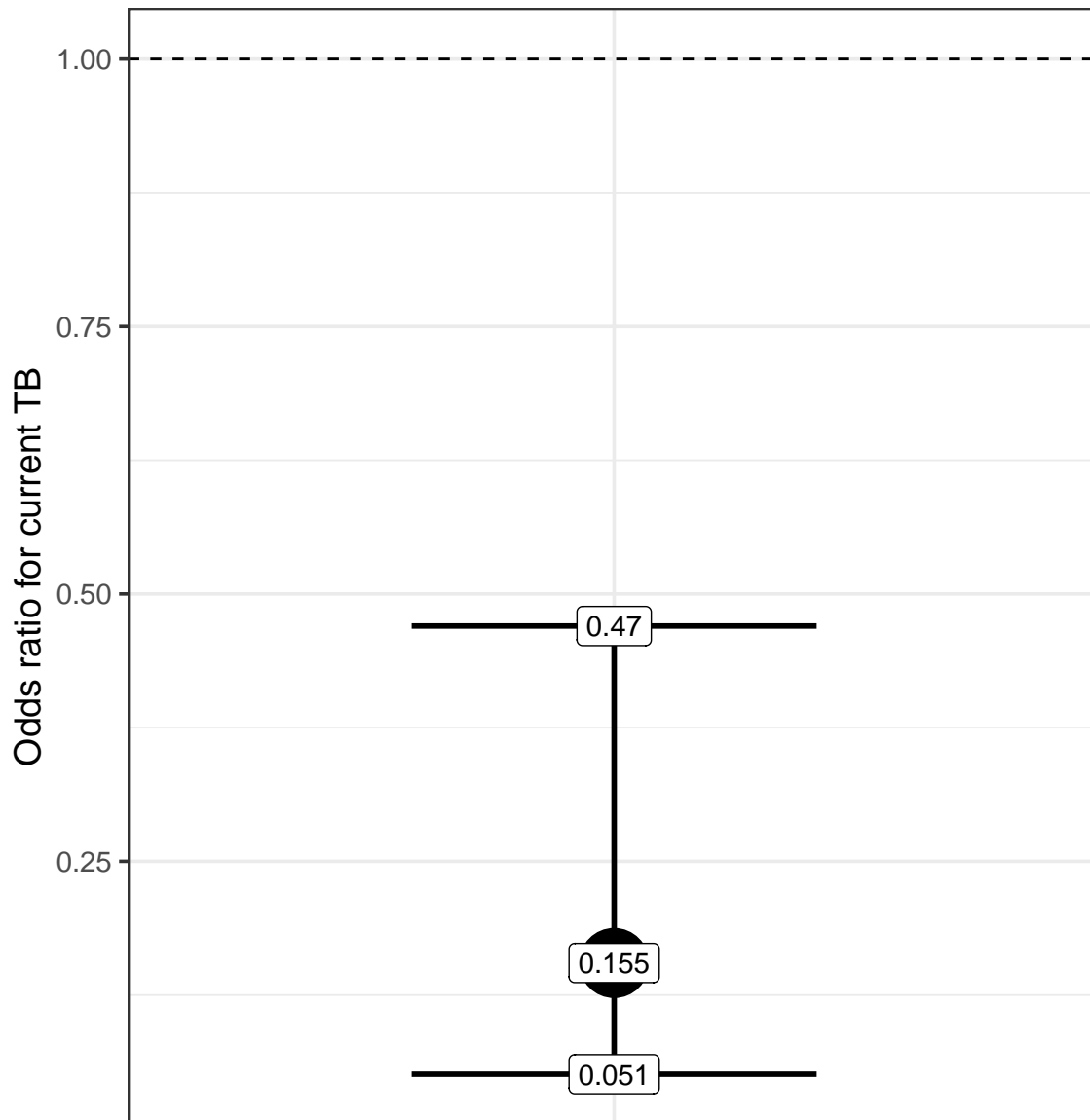
ggplot(data = boot_tb) +
  geom_hline(yintercept = 1,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Odds.ratio),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Odds.ratio,
                 label = Odds.ratio)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Bootstrap 95% CI of the odds ratio for current TB',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = 'Odds ratio for current TB') +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```

## Bootstrap 95% CI of the odds ratio for current TB

SN:yes vs SN:no

Interval type: BCa, Resamples: 1999



### Currently receiving TB treatment?

Treatment consisted of rifafour and pyridoxine (prophylaxis). Therefore only need to analyse rifafour data. Data coded as 'No' (not being treated), 'Yes' (being treated for active TB), and 'Prophylaxis' (being treated prophylactically for TB).

```
# Double-check matching between rifafour and pyridoxine columns
unique(data$rifafour_treatment == data$pyridoxine_prophylaxis)

## [1] TRUE

# Tabular summary
data %>%
```

```

select(rifafour_treatment, pyridoxine_prophylaxis, sn_present) %>%
group_by(pyridoxine_prophylaxis, sn_present) %>%
skim()

## Skim summary statistics
##   n obs: 120
##   n variables: 3
##   group variables: pyridoxine_prophylaxis, sn_present
##
## -- Variable type:factor -----
##   pyridoxine_prophylaxis sn_present      variable missing complete   n
##               no          yes rifafour_treatment      0         12 12
##               no          no  rifafour_treatment      0         75 75
##               yes          yes rifafour_treatment      0           8  8
##               yes          no  rifafour_treatment      0         11 11
##               prophylaxis    no  rifafour_treatment      0         14 14
##   n_unique      top_counts ordered
##       1 no: 12, yes: 0, pro: 0, NA: 0  FALSE
##       1 no: 75, yes: 0, pro: 0, NA: 0  FALSE
##       1 yes: 8, no: 0, pro: 0, NA: 0  FALSE
##       1 yes: 11, no: 0, pro: 0, NA: 0  FALSE
##       1 pro: 14, no: 0, yes: 0, NA: 0  FALSE

# Proportion on prophylaxis treatment
## Too low to analyse separately
round(mean(data$rifafour_treatment == 'prophylaxis'), 3)

## [1] 0.117

## ...so collapse 'yes' and 'prophylaxis'
data_tb <- data %>%
  mutate(rifafour_treatment = fct_collapse(rifafour_treatment,
                                           yes = c('yes', 'prophylaxis')))

# 95% bootstrap confidence interval of the proportion with current TB treatment by SN status
## Method = BCa, Resamples = 1999
### SN:yes
set.seed(1234)
sn_yes <- boot.ci(boot(data = data_tb[data_tb$sn_present == 'yes', ],
                      statistic = function(d, i){
                        mean(d[i, 'rifafour_treatment'] == 'yes')},
                      R = 1999,
                      stype = 'i'),
                  type = 'bca') %>%
  tibble(sn_present = 'yes',
         n = nrow(filter(data_tb, !is.na(rifafour_treatment) &
                           sn_present == 'yes')),
         Proportion = round(.$t0, 3),
         Conf.level = 0.95,
         Bca.lower = round(.$bca[[4]], 3),
         Bca.upper = round(.$bca[[5]], 3)) %>%
  .[1, -1] %>%
  as.data.frame()

### SN:no
set.seed(1234)

```



```

sn_no <- boot.ci(boot(data = data_tb[data_tb$sn_present == 'no', ],
  statistic = function(d, i){
    mean(d[i, 'rifafour_treatment'] == 'yes')),
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
tibble(sn_present = 'no',
  n = nrow(filter(data_tb, !is.na(rifafour_treatment) &
    sn_present == 'no')),
  Proportion = round(.$t0, 3),
  Conf.level = 0.95,
  Bca.lower = round(.$bca[[4]], 3),
  Bca.upper = round(.$bca[[5]], 3)) %>%
.[1, -1] %>%
as.data.frame()

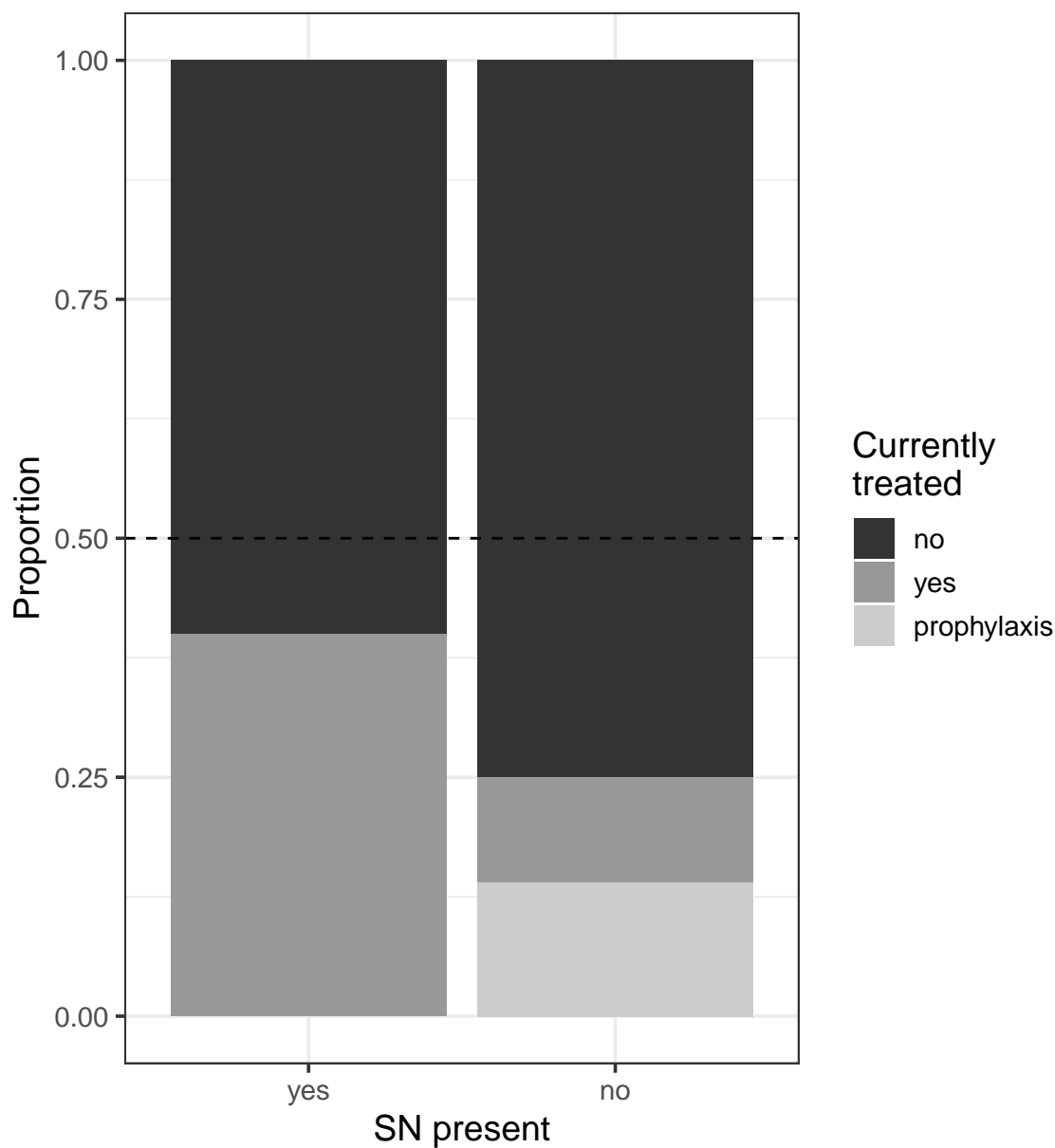
### Put sn and sn_no together and print
sn_yes %>%
  bind_rows(sn_no)

##   sn_present   n Proportion Conf.level Bca.lower Bca.upper
## 1         yes  20         0.40         0.95         0.15         0.55
## 2         no 100         0.25         0.95         0.16         0.33

# Plot
data %>%
  mutate(TB_current = str_to_title(rifafour_treatment)) %>%
  ggplot(data = .) +
  aes(x = sn_present,
    fill = rifafour_treatment) +
  geom_bar(position = 'fill') +
  geom_hline(yintercept = 0.5,
    linetype = 2) +
  labs(title = 'Being treated for TB at recruitment',
    subtitle = 'SN:yes vs SN:no',
    x = 'SN present',
    y = 'Proportion') +
  scale_fill_grey(name = 'Currently\ntreated')

```

## Being treated for TB at recruitment SN:yes vs SN:no



```
# Bootstrap 95% CI for the odds ratio of current TB treatment by SN status
## Resample: 1999
set.seed(1234)
boot_tb2 <- boot.ci(boot(data = data_tb,
  statistic = boot_OR,
  data_column = 'rifafour_treatment',
  grouping_column = 'sn_present',
  R = 1999,
  stype = 'i'),
  type = 'bca') %>%
  tibble(n = nrow(filter(data_tb, !is.na(rifafour_treatment))),
    Odds.ratio = round(.$t0, 3),
```

```

    Conf.level = 0.95,
    Bca.lower = round(.$bca[[4]], 3),
    Bca.upper = round(.$bca[[5]], 3),
    Includes.zero = ifelse(Bca.lower <= 1 & Bca.upper >= 1,
                           yes = 'yes',
                           no = 'no')) %>%

.[1, -1] %>%
as.data.frame(); boot_tb2

##      n Odds.ratio Conf.level Bca.lower Bca.upper Includes.zero
## 1 120      0.503      0.95      0.168      1.456             yes

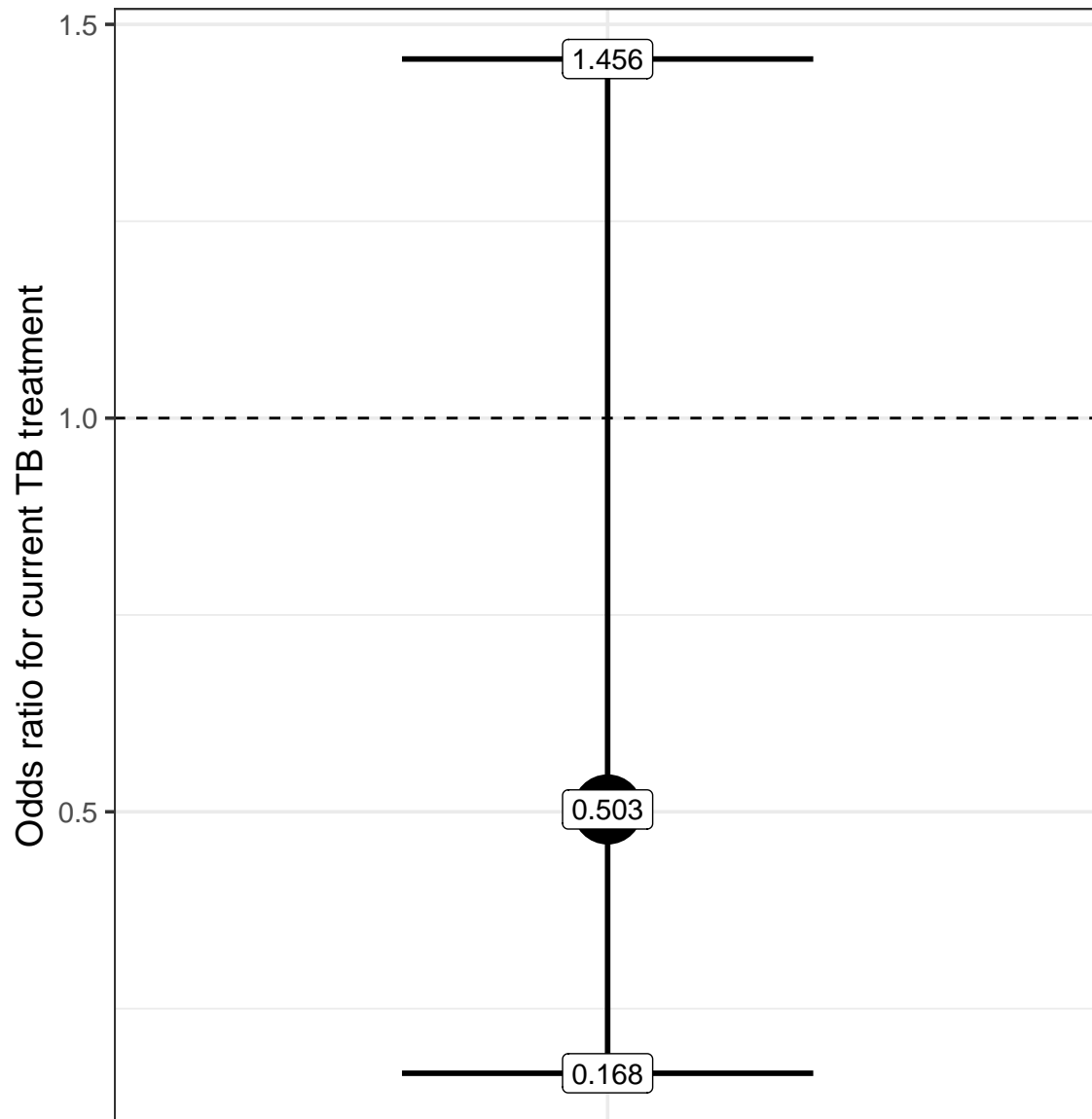
ggplot(data = boot_tb2) +
  geom_hline(yintercept = 1,
             linetype = 2) +
  geom_point(aes(x = 'x',
                 y = Odds.ratio),
             size = 12) +
  geom_errorbar(aes(x = 'x',
                   ymin = Bca.lower,
                   ymax = Bca.upper),
               width = 0.5,
               size = 1) +
  geom_label(aes(x = 'x',
                 y = Bca.lower,
                 label = Bca.lower)) +
  geom_label(aes(x = 'x',
                 y = Odds.ratio,
                 label = Odds.ratio)) +
  geom_label(aes(x = 'x',
                 y = Bca.upper,
                 label = Bca.upper)) +
  labs(title = 'Bootstrap 95% CI of the odds ratio for current TB treatment',
       subtitle = "SN:yes vs SN:no\nInterval type: BCa, Resamples: 1999",
       y = 'Odds ratio for current TB treatment') +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank())

```

## Bootstrap 95% CI of the odds ratio for current TB tre

SN:yes vs SN:no

Interval type: BCa, Resamples: 1999



## Session information

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.3
##
## Matrix products: default
```

```

## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] skimr_1.0.5      rcompanion_2.1.1 boot_1.3-20      forcats_0.4.0
## [5] stringr_1.4.0    dplyr_0.8.0.1    purrr_0.3.1      readr_1.3.1
## [9] tidyr_0.8.3      tibble_2.0.1     ggplot2_3.1.0    tidyverse_1.2.1
## [13] magrittr_1.5
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.0        lubridate_1.7.4    mvtnorm_1.0-9
## [4] lattice_0.20-38   multcompView_0.1-7 zoo_1.8-4
## [7] utf8_1.1.4        lmtest_0.9-36      assertthat_0.2.0
## [10] digest_0.6.18     R6_2.4.0           cellranger_1.1.0
## [13] plyr_1.8.4        backports_1.1.3    EMT_1.1
## [16] stats4_3.5.2      evaluate_0.13      http_1.4.0
## [19] pillar_1.3.1      rlang_0.3.1        lazyeval_0.2.1
## [22] multcomp_1.4-8    readxl_1.3.0       rstudioapi_0.9.0
## [25] Matrix_1.2-15     rmarkdown_1.11     labeling_0.3
## [28] splines_3.5.2     foreign_0.8-71     munsell_0.5.0
## [31] broom_0.5.1       compiler_3.5.2     modelr_0.1.4
## [34] xfun_0.5          pkgconfig_2.0.2    manipulate_1.0.1
## [37] DescTools_0.99.27 htmltools_0.3.6    tidyselect_0.2.5
## [40] expm_0.999-3      coin_1.2-2         codetools_0.2-16
## [43] fansi_0.4.0       crayon_1.3.4       withr_2.1.2.9000
## [46] MASS_7.3-51.1     grid_3.5.2         nlme_3.1-137
## [49] jsonlite_1.6      gtable_0.2.0       scales_1.0.0
## [52] cli_1.0.1         stringi_1.3.1      xml2_1.2.0
## [55] generics_0.0.2    nortest_1.0-4      sandwich_2.5-0
## [58] TH.data_1.0-10    tools_3.5.2        glue_1.3.0
## [61] hms_0.4.2         survival_2.43-3    yaml_2.2.0
## [64] colorspace_1.4-0  rvest_0.3.2        knitr_1.21
## [67] haven_2.1.0       modeltools_0.2-22

```