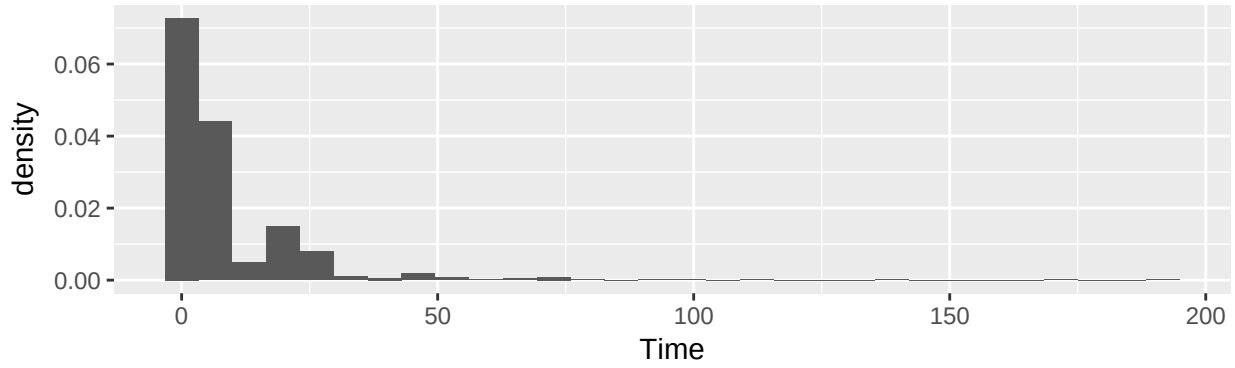
Bootstrap t Confidence Intervals

Part 2 - Bootstrap t Confidence Intervals

Verizon Repair Times

This example is taken from Hesterberg (2014). We have data on the amount of time it took Verizon to repair problems in their telephone lines in the state of New York. For legal reasons, there was interest in estimating the mean repair time.

```
library(readr)
library(dplyr)
library(ggplot2)

verizon <- read_csv("http://www.evanlray.com/data/chihara_hesterberg/Verizon.csv")
verizon_ilec <- verizon %>%
  filter(Group == "ILEC")

ggplot(data = verizon_ilec, mapping = aes(x = Time)) +
  geom_histogram(mapping = aes(y = ..density..))
```



Let's compare three interval estimates for the mean repair time:

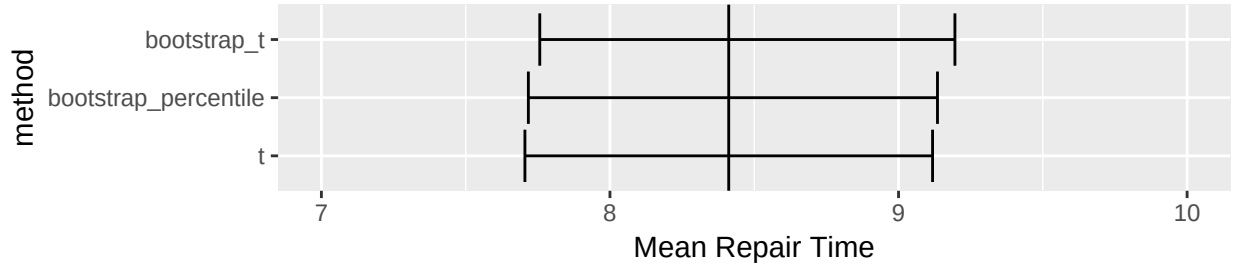
- t, from normal theory (intro stats)
- bootstrap percentile
- bootstrap t

```r
# sample size
n <- nrow(verizon_ilec)

# how many bootstrap samples to take, and storage space for the results
num_samples <- 10^4
bootstrap_results <- data.frame(
  estimate = rep(NA, num_samples), # for bootstrap percentile interval
  t_star = rep(NA, num_samples) # for bootstrap t interval
)

# draw many samples from the observed data and calculate mean of each simulated sample
for(i in seq_len(num_samples)) {
  ## Draw a bootstrap sample of size n with replacement from the observed data
  sampled_obs <- verizon_ilec %>%
    sample_n(size = n, replace = TRUE)

  ## Calculate mean of bootstrap sample
  bootstrap_results$estimate[i] <- mean(sampled_obs$Time)

  ## Calculate t statistic based on bootstrap sample
  bootstrap_results$t_star[i] <- (mean(sampled_obs$Time) - mean(verizon_ilec$Time)) /
    (sd(sampled_obs$Time) / sqrt(n))
}

# 95% CI from standard t results
t_interval <- t.test(verizon_ilec$Time)$conf.int

# 95% Bootstrap Percentile Interval
bootstrap_percentile_interval <- quantile(bootstrap_results$estimate, prob = c(0.025, 0.975))

# 95% Bootstrap t Interval
bootstrap_t_interval <- c(
  mean(verizon_ilec$Time) -
    quantile(bootstrap_results$t_star, prob = 0.975) * sd(verizon_ilec$Time) / sqrt(n),
  mean(verizon_ilec$Time) -
    quantile(bootstrap_results$t_star, prob = 0.025) * sd(verizon_ilec$Time) / sqrt(n)
)
```

```r
cis_df <- data.frame(
  pt_est = mean(verizon_ilec$Time),
  lower = c(t_interval[1], bootstrap_percentile_interval[1], bootstrap_t_interval[1]),
  upper = c(t_interval[2], bootstrap_percentile_interval[2], bootstrap_t_interval[2]),
  method = factor(c("t", "bootstrap_percentile", "bootstrap_t"),
    levels = c("t", "bootstrap_percentile", "bootstrap_t"),
    ordered = TRUE)
)

cis_df
```

```
##         pt_est    lower    upper               method
##       8.411611 7.705276 9.117945                    t
## 2.5%  8.411611 7.717223 9.135002 bootstrap_percentile
## 97.5% 8.411611 7.756769 9.195426          bootstrap_t
```

```
ggplot(data = cis_df, mapping = aes(xmin = lower, xmax = upper, x = pt_est, y = method)) +
  geom_errorbarh() +
  geom_vline(mapping = aes(xintercept = mean(verizon_ilec$Time))) +
  xlim(c(7, 10)) +
  xlab("Mean Repair Time")
```



Note: the Bootstrap t interval is asymmetric about the point estimate. This reflects the fact that the sampling distribution of the estimator is skewed, and results in better performance.