Exponential Distribution

Abhijith Asok, Chris Hilger, Liam Loscazo, Katherine Wang

Point Estimation: Methods

We compared the following point estimators: - Maximum Likelihood Estimator - Unbiased correction for the MLE - Second Method of Moment Estimator

add criteria for comparing estimators!

Maximum Likelihood Estimator

```
##code here
```

Unbiased correction for the MLE

```
##code here
```

Second Method of Moment Estimator

```
##code here
```

Confidence Intervals: Methods

We compared the following confidence intervals: - Wald-based Confidence Interval - Gamma-based Confidence Interval - Score-based Confidence Interval - Bootstrap Confidence Interval

add criteria for comparing CIs!!

Wald Confidence Interval

```r
wald_ci <- function(N, rate, alpha = 0.05){
  x <- rexp(N, rate = rate)
  x_bar <- mean(x)
  se <- sd(x)/sqrt(N)
  ci <- x_bar + c(-1, 1)*qnorm(1 - (alpha / 2))
  return(ci)
}
```

Gamma Confidence Interval

```r
gamma_ci <- function(N, rate, alpha = 0.05){
  # Inspiration: https://math.stackexchange.com/questions/1288
  x <- rexp(N, rate = rate)
  x_bar <- mean(x)
  ci_rate <- qgamma(c(alpha / 2, 1 - (alpha / 2)), N, N) / x_b
  ci_mean <- 1 / ci_rate
  return(c(ci_mean[2], ci_mean[1]))
}
```

Score Confidence Interval

```r
score_ci <- function(n, lambda, alpha) {
  x <- rexp(n, rate = lambda)
  xbar <- mean(x)
  return((1/xbar)*(1 + c(-1, 1)*qnorm(1 - alpha/2)/sqrt(n)))
}
```

Bootstrap Confidence Interval

```r
bootstrap_ci <- function(N, rate, alpha = 0.05){
  # Function to calculate bootstrap CI
  x <- rexp(N, rate = rate)
  x_bar <- mean(x)
  # Number of bootstrap samples
  nb <- 1000
  # Take boostrap samples
  bootstrap_samples <- sample(x, N * nb, replace = TRUE) %>%
    matrix(nrow = N, ncol = nb)
  # Get means of columns
  means <- colMeans(bootstrap_samples)
  # Get deltas (x* - x)
  deltas <- means - x_bar
  deltas <- sort(deltas)
  # Calculate CIs
  ci <- x_bar - quantile(deltas, probs = c(alpha / 2, 1 - (al
  return(c(ci[2], ci[1]))
```

Other

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Other functions
coverage_probability <- function(N, rate, ci_fun, alpha = 0.05
  # Match input function to actual function
  fun <- tryCatch(match.fun(ci_fun),
                  error = function(e) print(paste0("ci_fun: '"
```

Summary of Findings and Recommendations

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```