

Computational Statistics

Lab 5

Emil K Svensson and Rasmus Holm

2017-03-07

Question 1

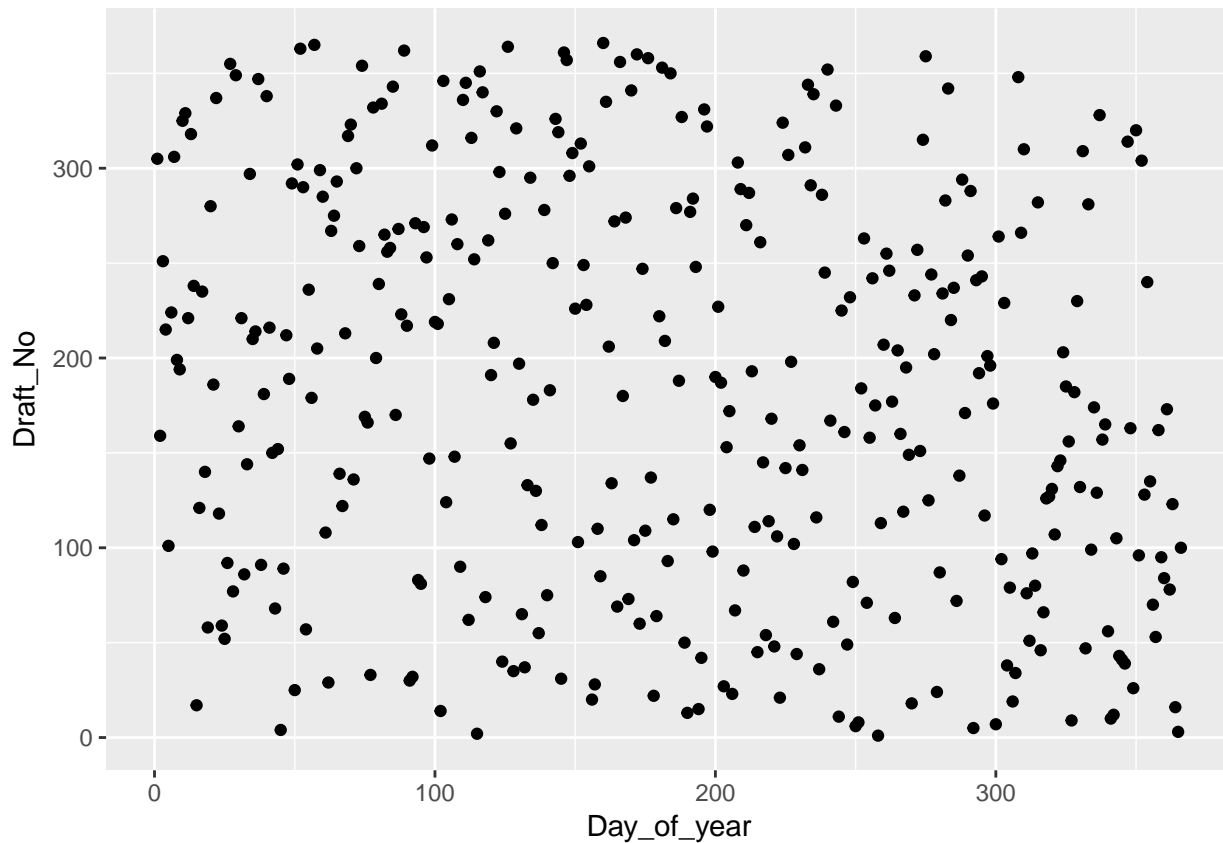
1.1

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
lottery <- read.csv2("../data/lottery.csv")
```

```
q11 <- ggplot(lottery, aes(x = Day_of_year, y = Draft_No)) + geom_point()  
plot(q11)
```



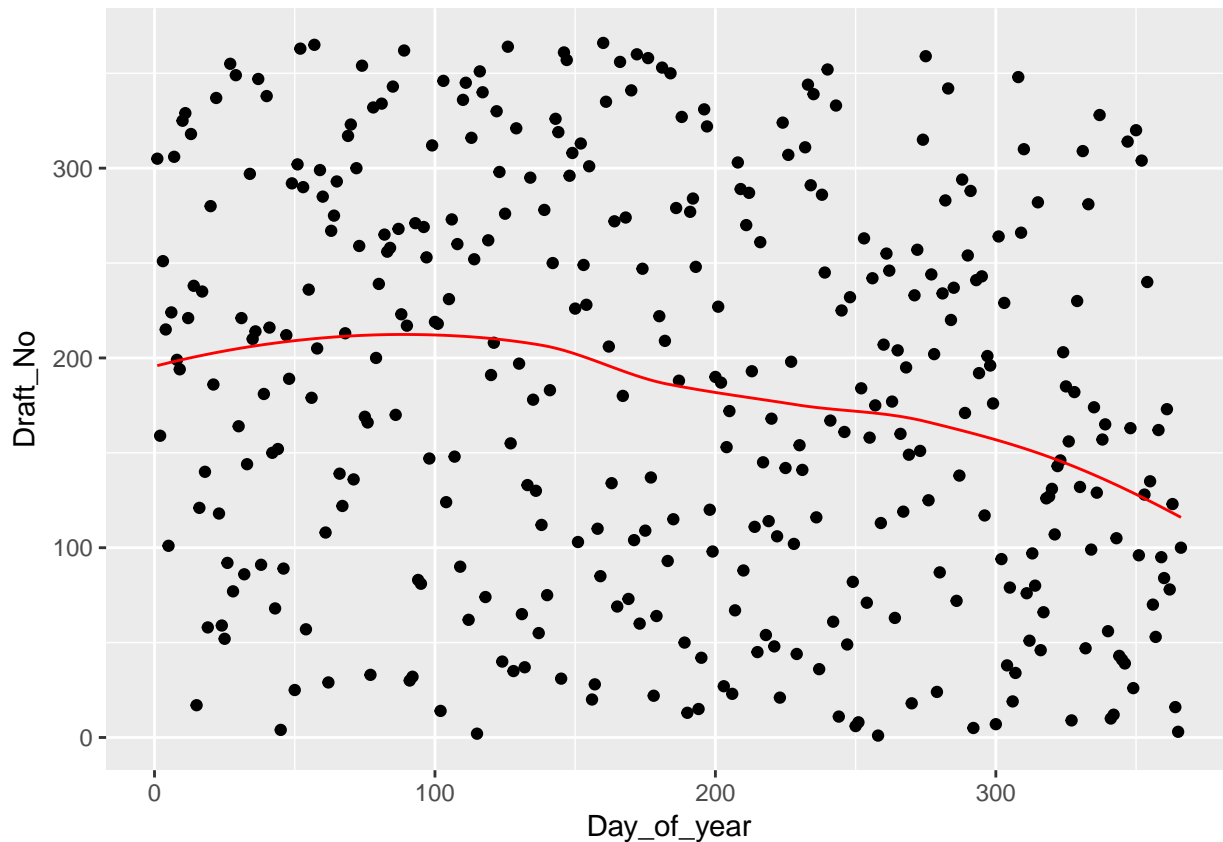
```
data <- data.frame(x=lottery$Day_of_year, y=lottery$Draft_No)
```

The data looks random although there might be some sort of skewness in the right side of the graph where there are a lacking some observations and therefore having a lower probability of getting selected.

1.2

```
loessfit <- loess(y ~ x, data=data)
data$pred <- predict(loessfit, data$x)

q12 <- q11 + geom_path(data = data, aes(x=x, y=pred), col = "red")
plot(q12)
```



The fit (line) doesn't seem straight and seems to have a decreasing trend which would support previous statements of people born on a days later on in a year has a lower probability of being selected.

1.3

```
teststat <- function(model) {
  function(data) {
    xa <- data$x[which.min(data$y)]
    xb <- data$x[which.max(data$y)]

    fit <- model(y ~ x, data)

    ya <- predict(fit, xa)
    yb <- predict(fit, xb)

    (yb - ya) / (xb - xa)
  }
}
```

```

}

teststat_boot <- function(data, idx, stat) {
  data <- data[idx,]
  stat(data)
}

library(boot)

B <- 2000

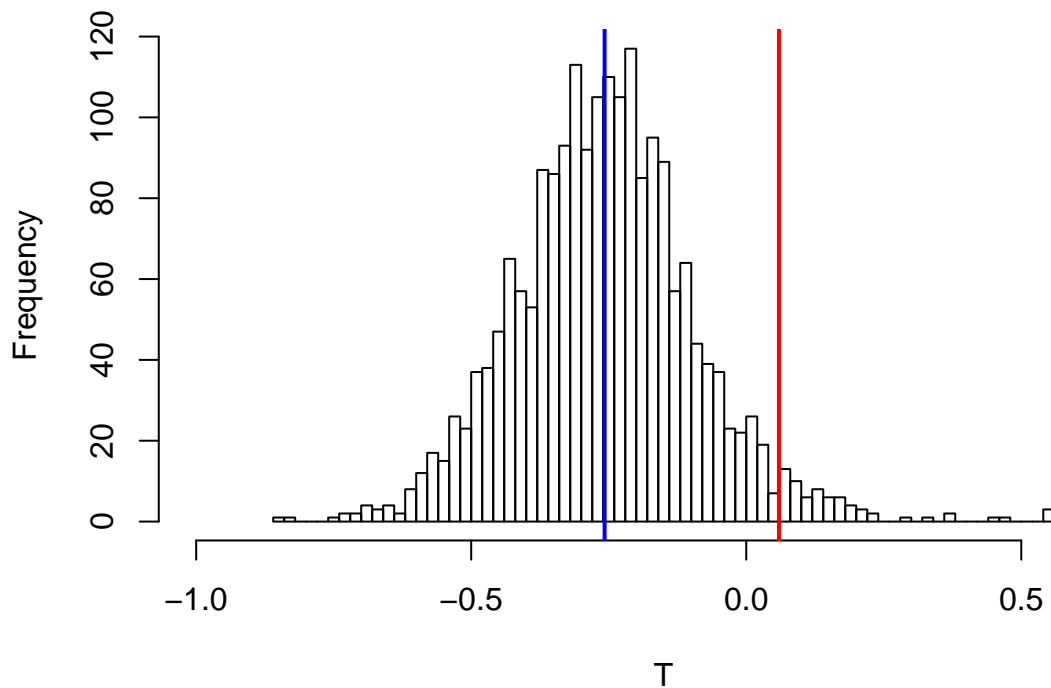
set.seed(123456)
npboot <- boot(data=data, statistic=teststat_boot, R=B, stat=teststat(model=loess))
pvalue <- sum(npboot$t > 0) / B
pvalue

## [1] 0.0595

hist(npboot$t, xlim = c(-1,0.7), breaks = 50,
     main = "Histogram for bootstrap t-values", xlab = "T")
abline(v=pvalue, col = "red", lwd = 2)
abline(v=mean(npboot$t), col="blue", lwd=2)

```

Histogram for bootstrap t-values



The p-value was calculated to 0.0595 and we can't reject the null hypothesis at a 0.05 alpha-level. We conclude H_0 and that the lottery is random.

1.4

```

teststat_permutation<- function(data, B, stat) {
  n <- nrow(data)
  statistics <- rep(0, B)
  newdata <- data.frame(x=data$x, y=sample(data$y, n))

  for (b in 1:B) {
    statistics[b] <- stat(newdata)
    newdata$y <- sample(data$y, n)
  }

  sum(abs(statistics) >= abs(stat(data))) / B
}

set.seed(123456)
pvalue2 <- teststat_permutation(data, B, teststat(loess))
pvalue2

```

```
## [1] 0.0925
```

We conclude H_0 in this test as well, the lottery is random.

1.5

```

genranddata <- function(x, alpha) {
  data.frame(x=x, y=pmax(0, pmin(alpha * x + rnorm(length(x), mean=183, sd=10), 366)))
}

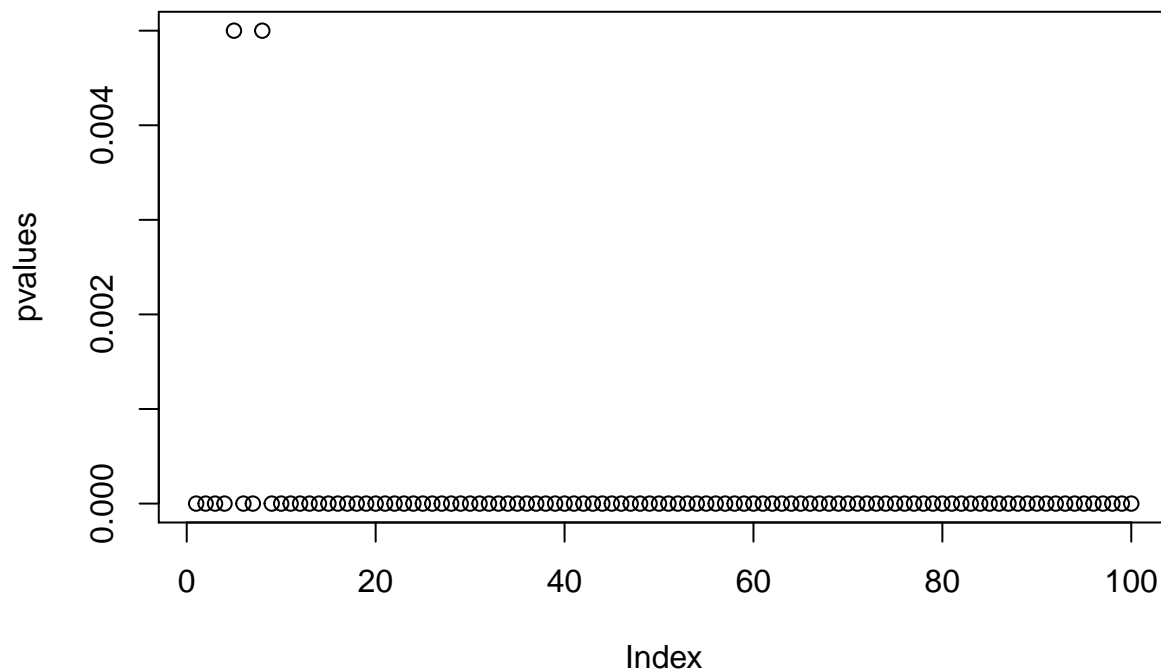
alphas <- seq(0.1, 10, by=0.1)
pvalues <- rep(0, length(alphas))

set.seed(123456)

for (i in 1:length(alphas)) {
  newdata <- genranddata(data$x, alphas[i])
  pvalues[i] <- teststat_permutation(newdata, 200, teststat(loess))
}

plot(pvalues)

```



```
print(sum(pvalues <= 0.05))
```

```
## [1] 100
```

Since we reject H_0 in every test the power will be 1 because we can't have any Type-2 errors when we reject everything. This isn't strange since the test-data is obviously non-random and gives us an indication that the test-statistic is a good one.

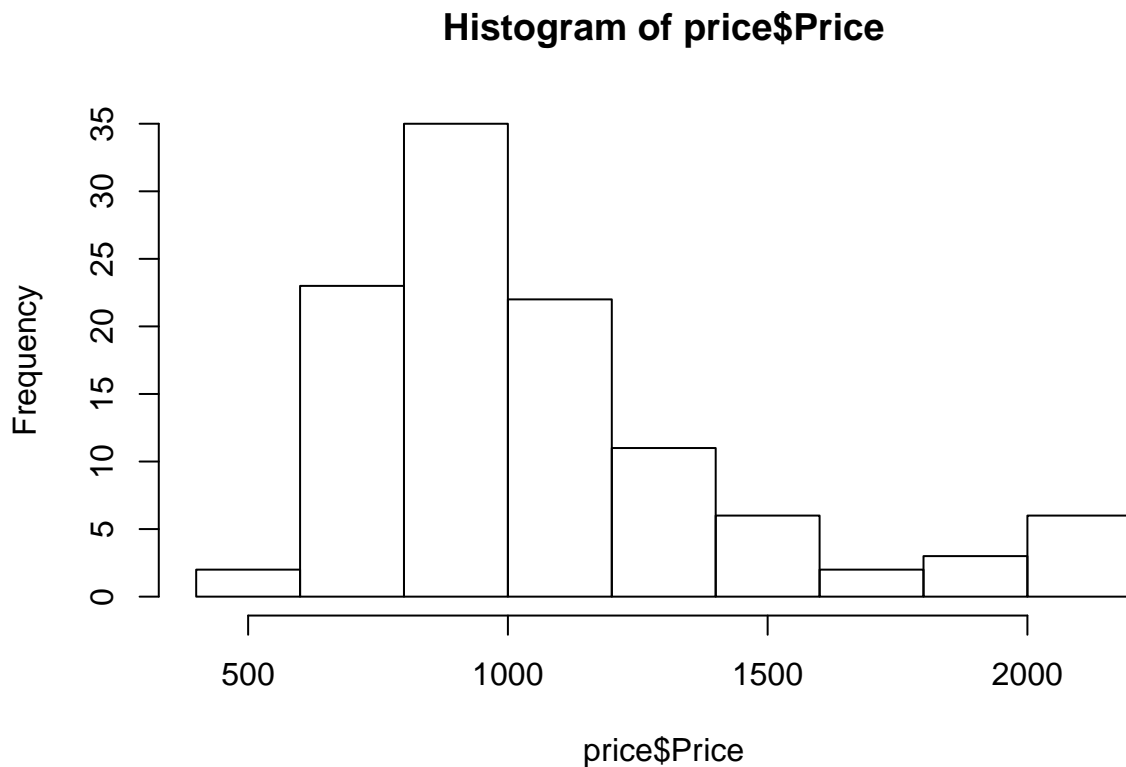
Question 2

2.1

```
price <- read.csv("../data/prices1.csv", sep=";")  
mean(price$Price)
```

```
## [1] 1080.473
```

```
hist(price$Price)
```



Looks like a Gamma distribution.

2.2

```
bootmean <- function(data,ind){  
  data <- data[ind]  
  mean(data)  
}
```

```
B <- 2000
```

```
estmean <- boot(data = price$Price,bootmean, R = B)
```

```
2*estmean$t0 - mean(estmean$t)
```

```
## [1] 1080.828
```

```

mean(estmean$t- estmean$t0)

## [1] -0.355

sum((estmean$t-mean(estmean$t))^2)/(B-1)

## [1] 1264.299

cibo <- boot.ci(estmean)

## Warning in boot.ci(estmean): bootstrap variances needed for studentized
## intervals

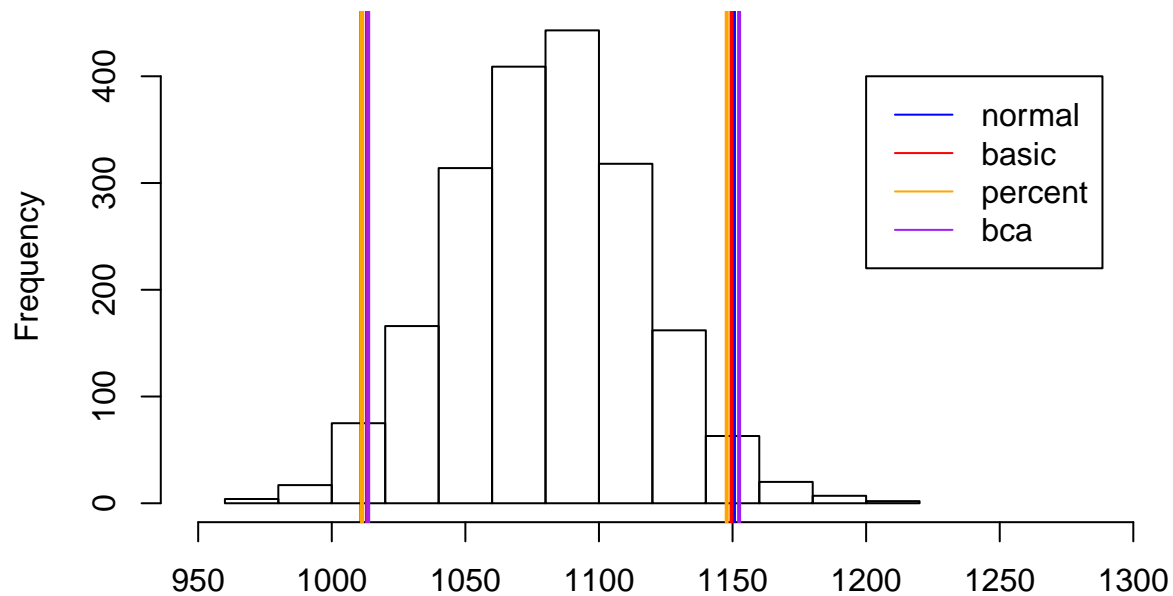
print(cibo)

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = estmean)
##
## Intervals :
## Level      Normal          Basic
## 95%   (1011, 1151 )   (1013, 1150 )
##
## Level      Percentile      BCa
## 95%   (1011, 1148 )   (1014, 1152 )
## Calculations and Intervals on Original Scale

hist(estmean$t, xlim = c(950,1300), main = "Bootstrap estimation of mean", xlab = "")
abline(v= cibo$normal[2:3], col = "blue", lwd = 2)
abline(v= cibo$basic[4:5], col = "red", lwd = 2)
abline(v= cibo$percent[4:5], col = "orange", lwd = 2)
abline(v= cibo$bca[4:5], col = "purple", lwd = 2)
legend(y=400,x =1200,c("normal","basic","percent","bca"),lty = c(1,1),
      col = c("blue","red","orange","purple"))

```

Bootstrap estimation of mean



2.3

```
jackknife <- function(data,B,tstat){
  stopifnot( B <= length(data))
  est <- rep(1,times = B)
  for(i in 1:B){
    est[i] <- tstat(data[-i])
  }
  return(est)
}

jackest<-jackknife(price$Price, B = length(price$Price), tstat = mean)

mean(jackest)

## [1] 1080.473

n <- length(price$Price)

tstar<- n*mean(price$Price)-(n-1)*jackest
JT<-mean(tstar)
jackvar <- (1/(n*(n-1)))*sum((tstar- JT)^2)

lowerci <- JT - 1.96*sqrt(jackvar)
upperci <- JT + 1.96*sqrt(jackvar)

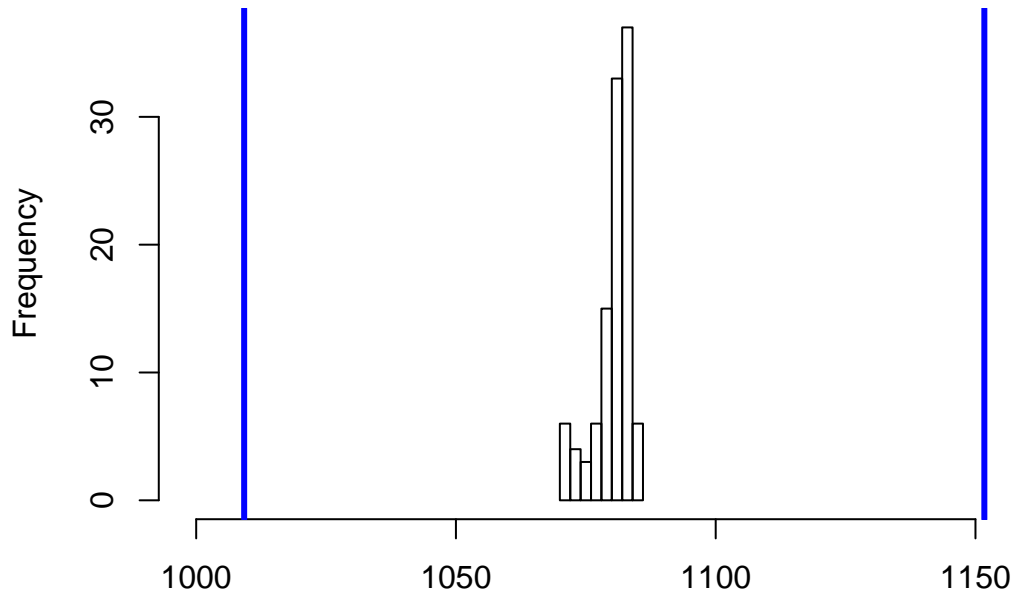
# mean(tstar) + qt(0.975,length(price$Price)-1)*sqrt(var(tstar)/length(price$Price))
```



```
# mean(tstar) - qt(0.975,length(price$Price)-1)*sqrt(var(tstar)/length(price$Price))

hist(jackest,breaks = 10, xlim = c(1000,1180), xlab = "" ,main = "Jackknife estimate")
abline(v=lowerci, col ="blue", lwd = 3)
abline(v=upperci, col ="blue", lwd = 3)
```

Jackknife estimate



2.4

```
library(knitr)

## Warning: package 'knitr' was built under R version 3.3.2

cimatrix<-rbind(
  c("Normal",cibo$normal[2],mean(estmean$t),cibo$normal[2]),
  c("Percent",cibo$percent[4],mean(estmean$t),cibo$percent[5]),
  c("Basic",cibo$basic[4],mean(estmean$t),cibo$basic[5]),
  c("Bca",cibo$bca[4],mean(estmean$t),cibo$bca[5]),
  c("Jackknife",lowerci,mean(jackest),upperci)
)
colnames(cimatrix) <- c("Name","Lower","Mean","Upper")

kable(cimatrix, digits = 3)
```

Name	Lower	Mean	Upper
Normal	1011.13731332721	1080.11772727273	1011.13731332721
Percent	1011.30801927524	1080.11772727273	1147.81474498594
Basic	1013.13070955952	1080.11772727273	1149.63743527021
Bca	1013.62920802339	1080.11772727273	1152.45553911654
Jackknife	1009.23782103407	1080.47272727273	1151.70763351138