Final

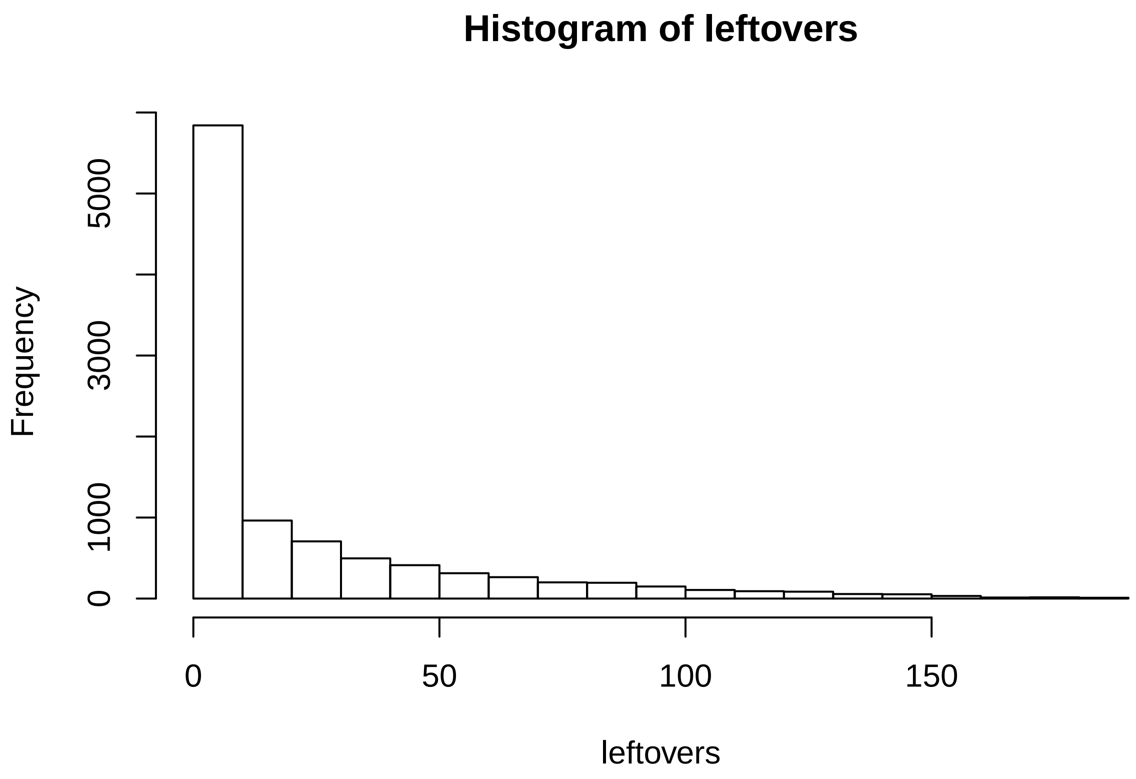
Max Wagner

May 19, 2016

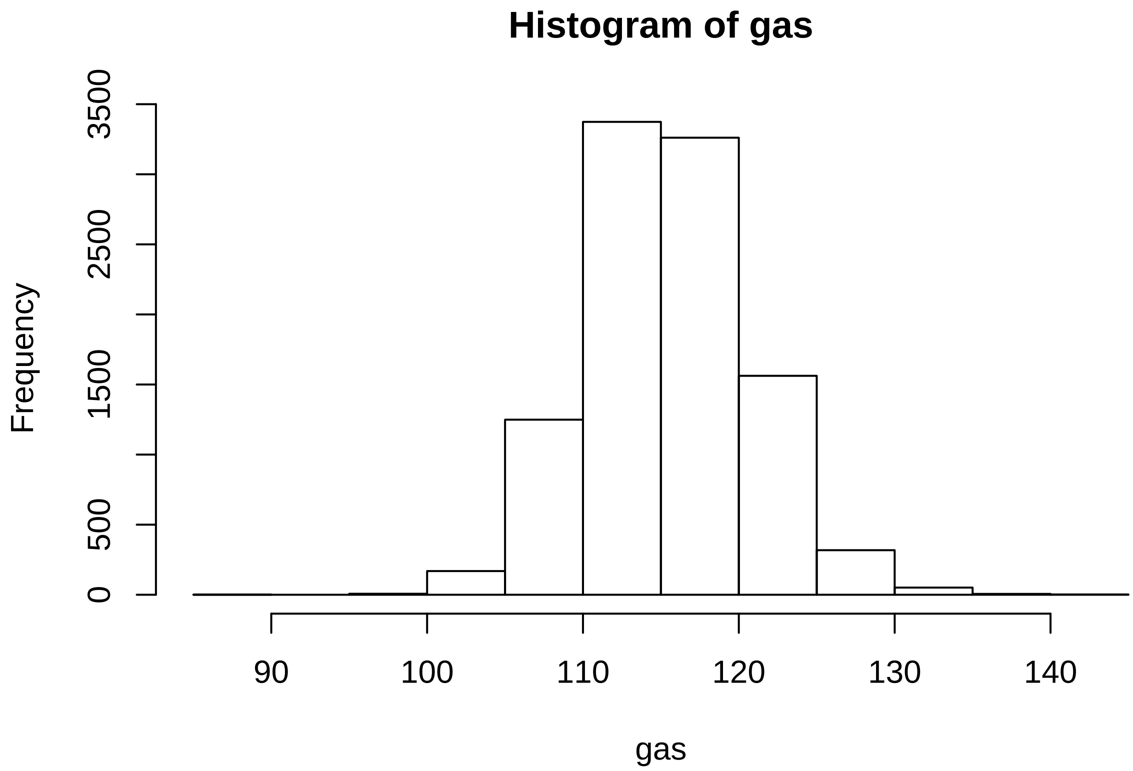
Setting up a queue where t = timer, d = drivers, p = passengers, and drivers sit still after their fare.

```
leftovers <- c()
gas <- c()
for (z in 1:10000) {
  #starting amount of drivers and passengers
  d.start <- 250
  p.start <- 250
  p.done <- rep(0,p.start)

  #start time for the sim
  t.start <- rep(0,d.start)
  t.current <- t.start

  #400 locations a passenger or driver could be
  d.quadrant <- c(1:400)
  p.quadrant <- c(1:400)

  #how far should a driver travel to find a passenger?
  d.dist <- 25

  #starting locations of drivers and passengers, sort them to get them as close to each as we can
  d.start.loc <- sample(d.quadrant, d.start, replace = T)
  d.start.loc <- sort(d.start.loc)
  p.start.loc <- sample(p.quadrant, p.start, replace = T)
  p.start.loc <- sort(p.start.loc)
  p.d.diff <- abs(p.start.loc - d.start.loc)
  d.takefare <- c()
  for (i in 1:length(p.d.diff)) {
    if (p.d.diff[i]==0 | p.d.diff[i]<=d.dist) {
      d.takefare[i] <- 1
      p.done[i] <- 1
    } else
      d.takefare[i] <- 0
  }

  #time for driver to get to fare, if they take it
  d.timetofare <- c()
  for (i in 1:length(d.takefare)) {
    if (d.takefare[i] == 1)
      d.timetofare[i] <- p.d.diff[i]
    else
      d.timetofare[i] <- NA
  }

  #end destination of p, how long is it
  p.dest <- sample(p.quadrant, p.start, replace = T)
```

```r
  p.length <- abs(p.dest - p.start)

  #when and where do the fare end
  for (i in 1:length(d.takefare)) {
    if (d.takefare[i] == 1) {
      t.current[i] <- d.timetofare[i] + p.length[i]
      d.start.loc[i] <- p.dest[i]
    } else
      t.current[i] <- t.start[i]
  }

  # create a data frame from it, sort it again by driver start location
  dat <- data.frame(cbind(t.start, p.done, t.current, d.start.loc, p.start.loc, p.d.diff, d.takefare, d
  leftovers[z] <- p.start - sum(dat[p.done==1,]$p.done)
  gas[z] <- mean(dat[p.done==1,]$t.current)
}
leftover1 <- mean(leftovers)
gas1 <- mean(gas)
hist(leftovers)
```

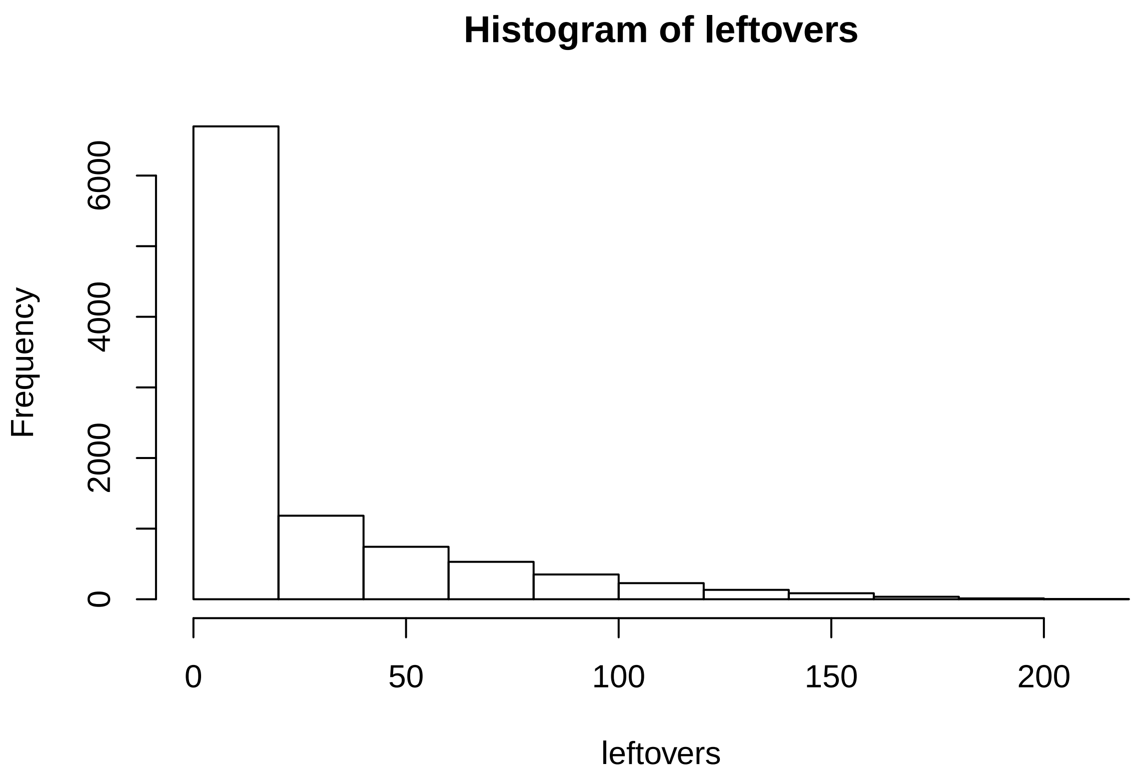


```r
hist(gas)
```

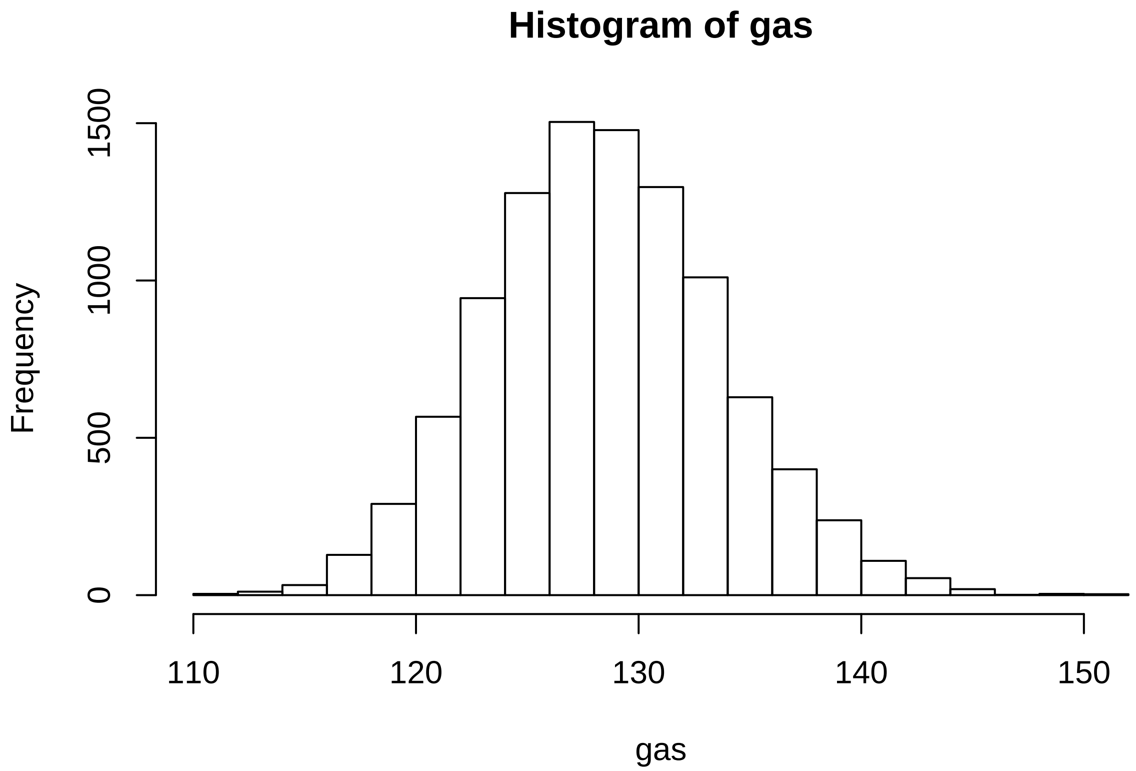


Now we move randomly:

```r
leftovers <- c()
gas <- c()
for (z in 1:10000) {
  #starting amount of drivers and passengers
  d.start <- 250
  p.start <- 250
  p.done <- rep(0,p.start)

  #start time for the sim
  t.start <- rep(0,d.start)
  t.current <- t.start

  #400 locations a passenger or driver could be
  d.quadrant <- c(1:400)
  p.quadrant <- c(1:400)

  #how far should a driver travel to find a passenger?
  d.dist <- 25

  #starting locations of drivers and passengers, sort them to get them as close to each as we can
  d.start.loc <- sample(d.quadrant, d.start, replace = T)
  d.change <- sample(-d.dist:d.dist,d.start, replace = TRUE)
  d.start.loc <- d.start.loc + d.change
  for (i in 1:length(d.start.loc)) {
    if (d.start.loc[i] > 400)
      d.start.loc[i] <- 400
    else if (d.start.loc[i] < 1)
      d.start.loc[i] <- 1
  }
  d.start.loc <- sort(d.start.loc)
  p.start.loc <- sample(p.quadrant, p.start, replace = T)
  p.start.loc <- sort(p.start.loc)
  p.d.diff <- abs(p.start.loc - d.start.loc)
  d.takefare <- c()
  for (i in 1:length(p.d.diff)) {
    if (p.d.diff[i]==0 | p.d.diff[i]<=d.dist) {
      d.takefare[i] <- 1
      p.done[i] <- 1
    } else
      d.takefare[i] <- 0
  }

  #time for driver to get to fare, if they take it
  d.timetofare <- c()
  for (i in 1:length(d.takefare)) {
    if (d.takefare[i] == 1)
      d.timetofare[i] <- p.d.diff[i]
    else
      d.timetofare[i] <- NA
  }

  #end destination of p, how long is it
  p.dest <- sample(p.quadrant, p.start, replace = T)
  p.length <- abs(p.dest - p.start)

  #when and where do the fare end
  for (i in 1:length(d.takefare)) {
    if (d.takefare[i] == 1) {
      t.current[i] <- d.timetofare[i] + p.length[i] + abs(d.change[i])
      d.start.loc[i] <- p.dest[i]
    } else
      t.current[i] <- t.start[i] + abs(d.change[i])
  }

  # create a data frame from it, sort it again by driver start location
  dat <- data.frame(cbind(t.start, p.done, t.current, d.start.loc, p.start.loc, p.d.diff, d.takefare, d
  leftovers[z] <- p.start - sum(dat[p.done==1,]$p.done)
  gas[z] <- mean(dat[p.done==1,]$t.current)
}
leftover2 <- mean(leftovers)
gas2 <- mean(gas)
hist(leftovers)
```

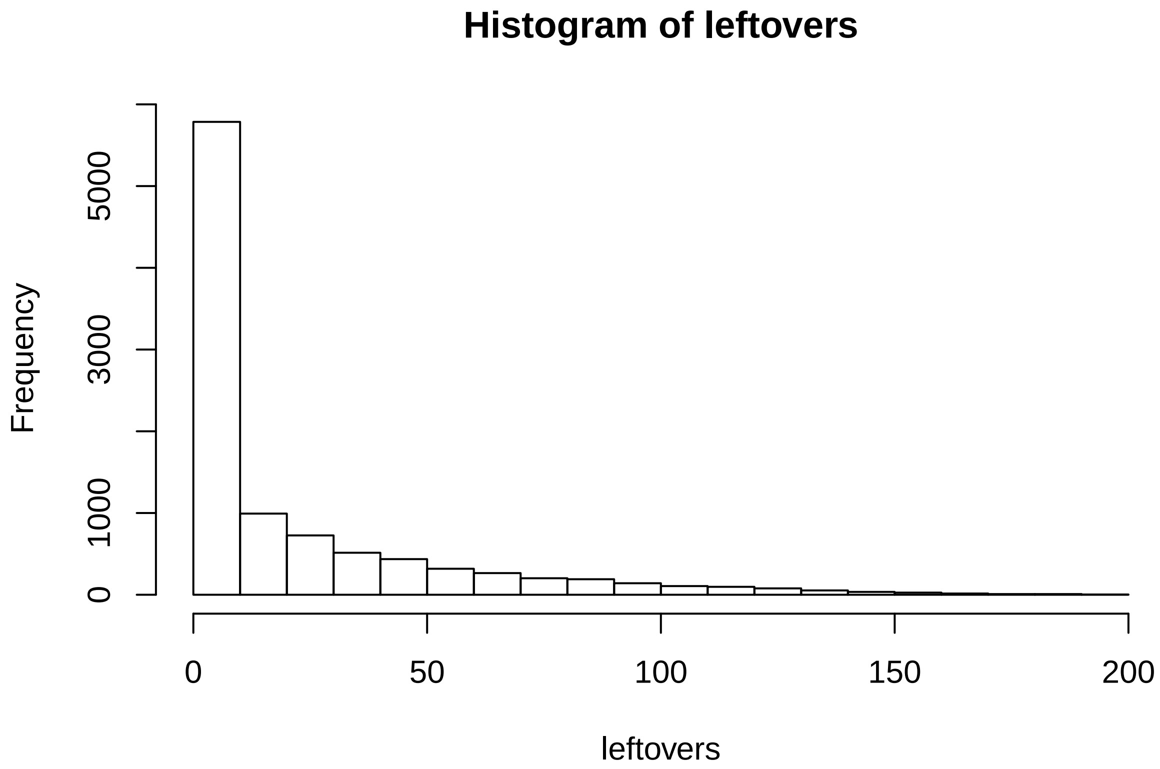

Histogram of leftovers
Frequency
0
2000
4000
6000
0
50
100
150
200
leftovers

```r
hist(gas)
```

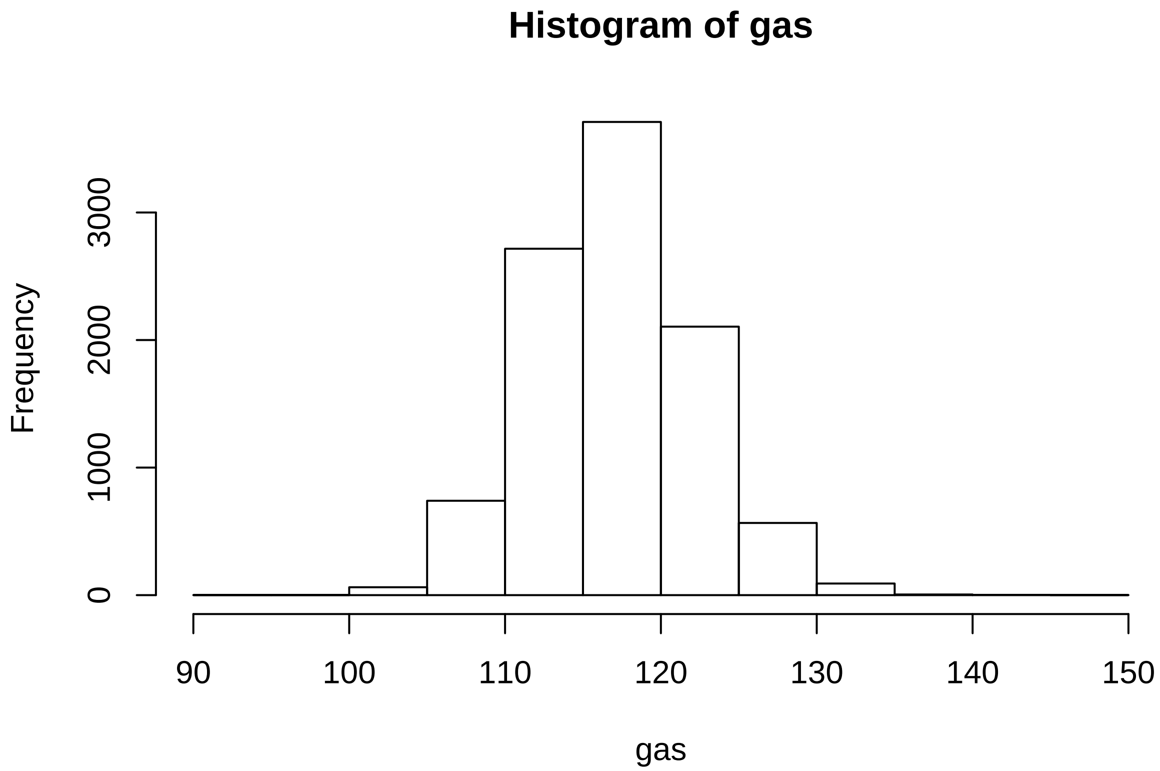


And now we move toward a target:

```
leftovers <- c()
gas <- c()
for (z in 1:10000) {
  #starting amount of drivers and passengers
  d.start <- 250
  p.start <- 250
  p.done <- rep(0,p.start)

  #start time for the sim
  t.start <- rep(0,d.start)
  t.current <- t.start

  #400 locations a passenger or driver could be
  d.quadrant <- c(1:400)
  p.quadrant <- c(1:400)

  #how far should a driver travel to find a passenger?
  d.dist <- 25

  #starting locations of drivers and passengers, sort them to get them as close to each as we can
  p.start.loc <- sample(p.quadrant, p.start, replace = T)
  p.start.loc <- sort(p.start.loc)
  d.start.loc <- sample(d.quadrant, d.start, replace = T)
  mean.d.start.loc <- median(p.start.loc)
  move.d.start.loc <- sample(1:2,d.start, replace = TRUE)
  for (i in 1:length(d.start.loc)) {
    if (d.start.loc[i] > mean.d.start.loc)
      d.start.loc[i] <- d.start.loc[i] - move.d.start.loc[i]
    else if(d.start.loc[i] < mean.d.start.loc)
      d.start.loc[i] <- d.start.loc[i] + move.d.start.loc[i]
  }
  d.start.loc <- sort(d.start.loc)
  p.d.diff <- abs(p.start.loc - d.start.loc)
  d.takefare <- c()
  for (i in 1:length(p.d.diff)) {
    if (p.d.diff[i]==0 | p.d.diff[i]<=d.dist) {
      d.takefare[i] <- 1
      p.done[i] <- 1
    } else
      d.takefare[i] <- 0
  }

  #time for driver to get to fare, if they take it
  d.timetofare <- c()
  for (i in 1:length(d.takefare)) {
    if (d.takefare[i] == 1)
      d.timetofare[i] <- p.d.diff[i]
    else
      d.timetofare[i] <- NA
  }

  #end destination of p, how long is it
  p.dest <- sample(p.quadrant, p.start, replace = T)
  p.length <- abs(p.dest - p.start)

  #when and where do the fare end
  for (i in 1:length(d.takefare)) {
    if (d.takefare[i] == 1) {
      t.current[i] <- d.timetofare[i] + p.length[i] + move.d.start.loc[i]
      d.start.loc[i] <- p.dest[i]
    } else
      t.current[i] <- t.start[i] + move.d.start.loc[i]
  }

  # create a data frame from it, sort it again by driver start location
  dat <- data.frame(cbind(t.start, p.done, t.current, d.start.loc, p.start.loc, p.d.diff, d.takefare, d
  leftovers[z] <- p.start - sum(dat[p.done==1,]$p.done)
  gas[z] <- mean(dat[p.done==1,]$t.current)
}
leftover3 <- mean(leftovers)
gas3 <- mean(gas)
hist(leftovers)
```


Histogram of leftovers
Frequency
0
1000
3000
5000
0
50
100
150
200
leftovers

```r
hist(gas)
```



And now we can compare all 3 together:

```
titles <- c("stationary", "random", "hotspot")
leftovers <- c(leftover1,leftover2,leftover3)
gas <- c(gas1,gas2,gas3)
cbind(titles,leftovers,gas)
```

```
##      titles       leftovers gas
## [1,] "stationary" "21.7394" "115.477643747725"
## [2,] "random"     "22.6617" "128.51025366149"
## [3,] "hotspot"    "21.4243" "117.109562309126"
```