Problem Set #3 - Florence

Azka, Jonathan, Jordan

Monday, February 15, 2016

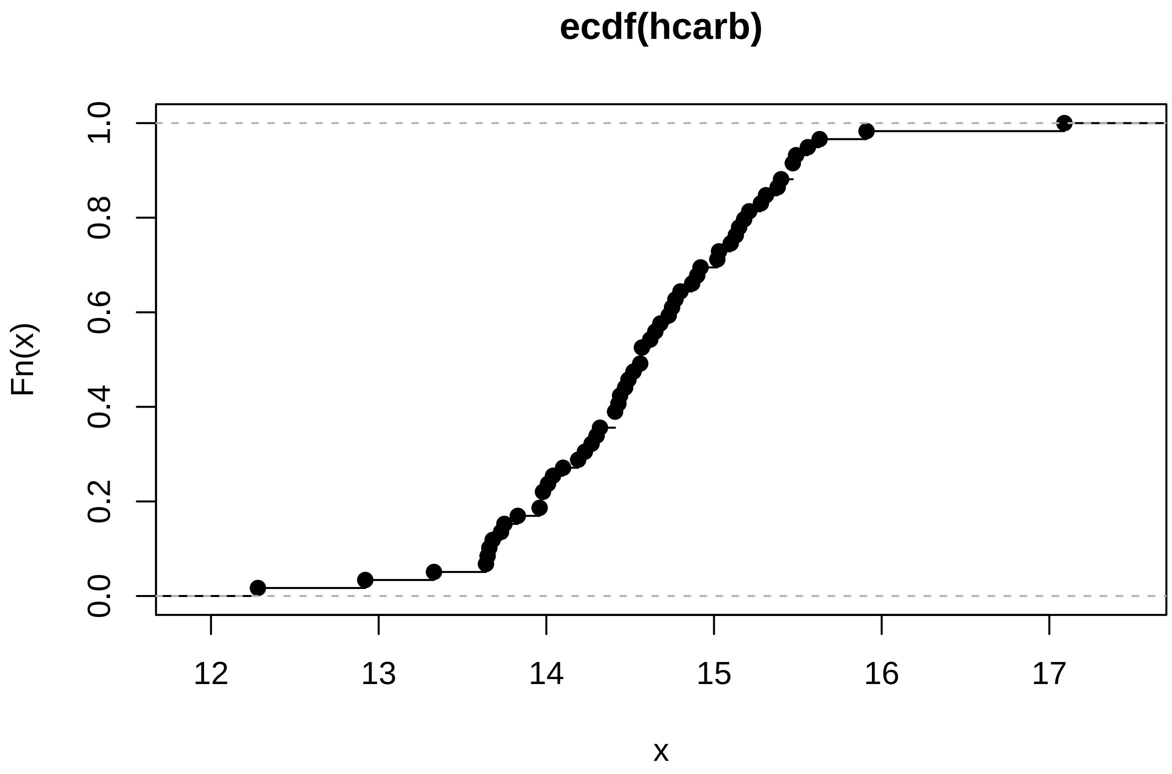
Problem 10.6

Various chemical tests were conducted on beeswax. In particular, the percentage of hydrocarbons in each sample of wax was determined.

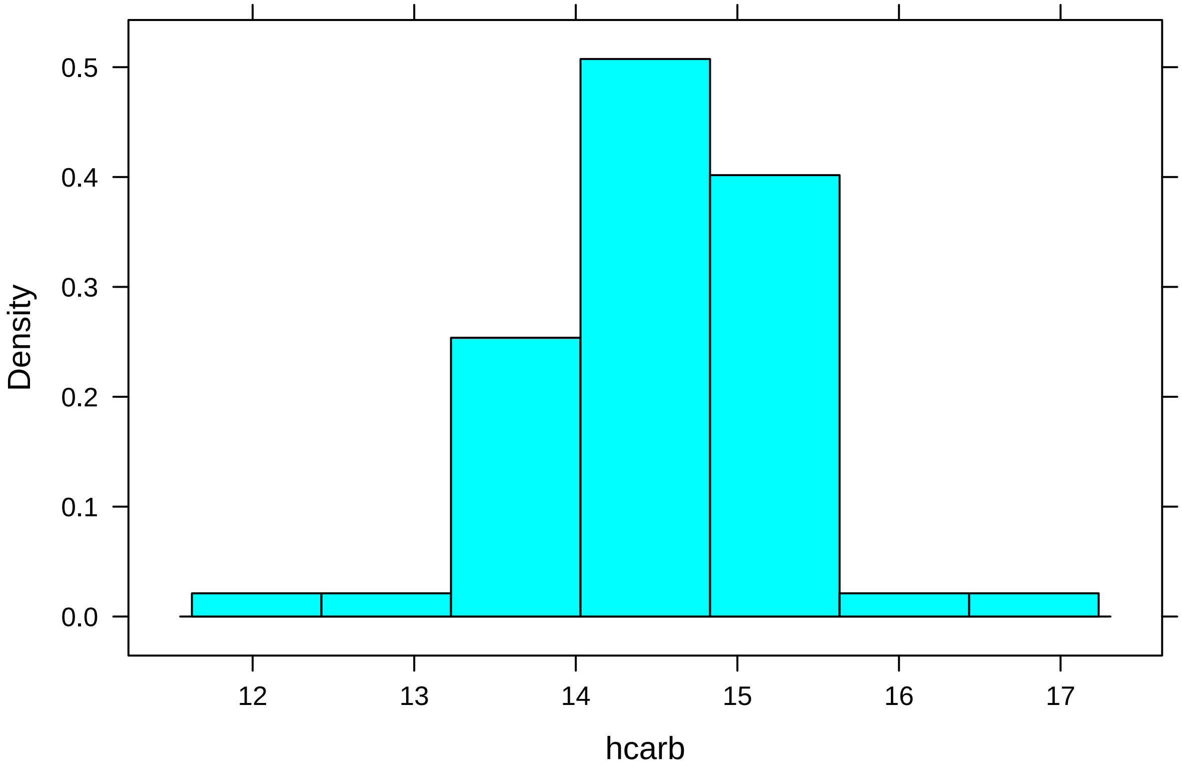
Part (a)

Plot the ecdf, the histogram, and a normal probability plot of the percentages of hydrocarbons. Find the .90, .75, .50, .25, and .10 quantiles. Does the distribution appear Gaussian?

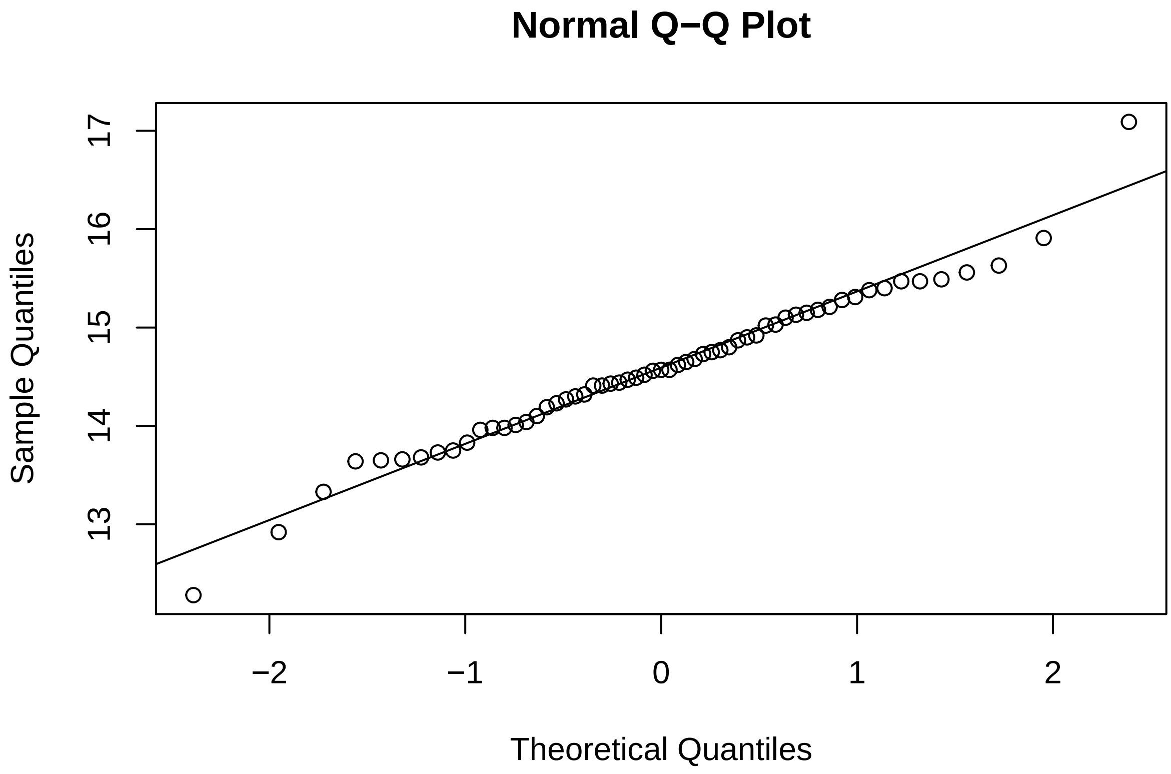
```
beeswax <- read.csv("http://www3.amherst.edu/~nhorton/rice/chapter10/beeswax.csv")
hcarb <- beeswax$Hydrocarbon

hcarb.ecdf <- ecdf(hcarb)
plot(hcarb.ecdf)
```



```
histogram(hcarb)
```



```
qqnorm(hcarb)
qqline(hcarb)
```



```
quantile(hcarb, c(.90, .75, .50, .25, .10))
```

```
##    90%    75%    50%    25%    10% 
## 15.470 15.115 14.570 14.070 13.676
```

The distribution does appear Gaussian.

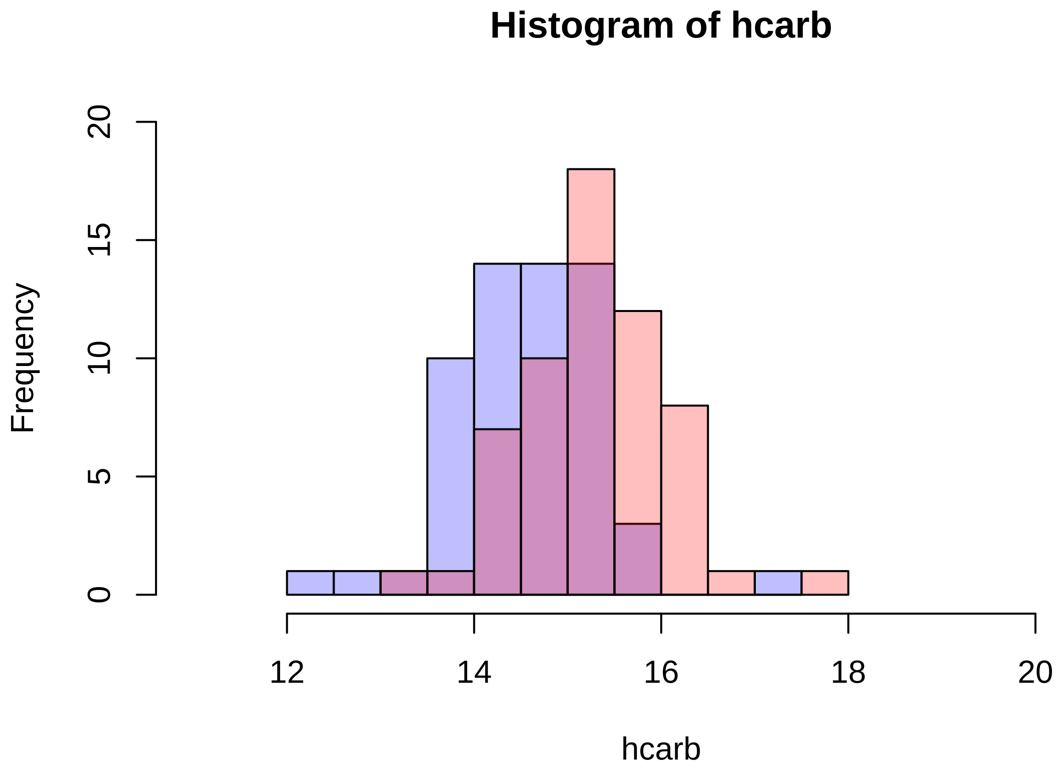
Part (b)

The average percentage of hydrocarbons in microcrystalline wax (a synthetic commercial wax) is 85%. Suppose that beeswax was diluted with 1% microcrystalline wax. Could this be detected? What about a 3% or a 5% dilution?

To help illustrate these dilutions, we plot histograms of the original beeswax compared to histograms of the diluted beeswaxes.

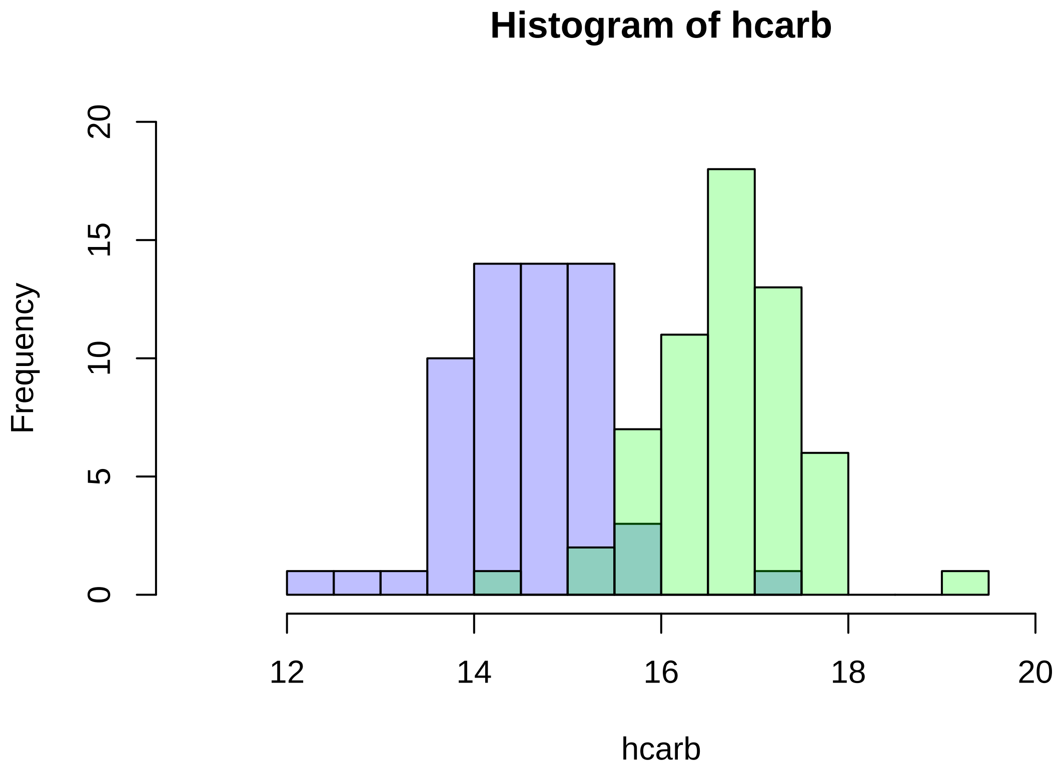
```
undiluted <- hist(hcarb)
synth.01 <- hist(hcarb*.99 + 85*.01)
synth.03 <- hist(hcarb*.97 + 85*.03)
synth.05 <- hist(hcarb*.95 + 85*.05)
```

```
plot(undiluted, col=rgb(0,0,1,1/4), xlim=c(11, 21), ylim = c(0, 20))
plot(synth.01, col=rgb(1,0,0,1/4), xlim=c(11, 21), add = TRUE)
```



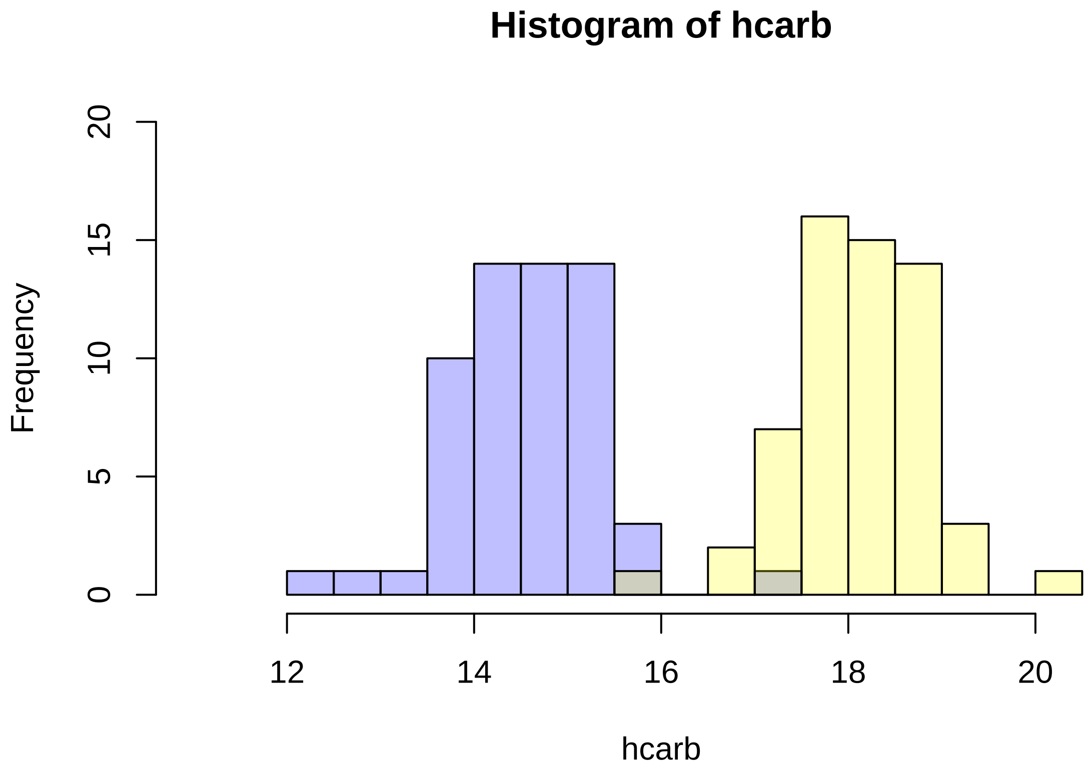
As the histograms above show, it would be difficult to detect a 1% dilution.

```
plot(undiluted, col=rgb(0,0,1,1/4), xlim=c(11, 21), ylim = c(0, 20))
plot(synth.03, col=rgb(0,1,0,1/4), xlim=c(11, 21), add = TRUE)
```



As the histograms above show, it would not be too difficult to detect a 3% dilution.

```
plot(undiluted, col=rgb(0,0,1,1/4), xlim=c(11, 21), ylim = c(0, 20))
plot(synth.05, col=rgb(1,1,0,1/4), xlim=c(11, 21), add = TRUE)
```



As the histograms above show, it would be relatively easy to detect a 5% dilution.

Problem 10.10

Part (a)

Find the mean and variance of $X_(k)$ from a uniform distribution on [0,1].

```
numsim <- 10000

n <- 99 # Number of elements in each sample
k <- 50 # Arbitrary order statistic to examine; here we use median

getOrderStat <- function(){
  x <- runif(n, 0, 1)
  return(sort(x)[k])
}

vals <- do(numsim)*getOrderStat()
confint(t.test(~vals[,1])); k/(n+1)
```

```
##   mean of x     lower     upper level
## 1 0.5007647 0.4997959 0.5017334  0.95

## [1] 0.5
```

```r
var(vals[,1]); (1/(n+2))*(k/(n+1))*(1-k/(n+1))
```

```
## [1] 0.0024426

## [1] 0.002475248
```

We note that the calculated mean lies within the confidence interval for the value of the empirical mean. Also, the calculated and empirical variances are essentially equal.

Part (b)

Find the approximate mean and variance of $Y_(k)$ from an arbitrary distribution with cdf F.

We use an exponential distribution with rate parameter 1:

```r
n <- 99  # Number of elements in each sample
k <- 50  # Arbitrary order statistic to examine; here we use median
lambda <- 1

getOrderStatExp <- function() {
    x <- rexp(n, rate = lambda)
    return(sort(x)[k])
}

vals <- do(numsim) * getOrderStatExp()
confint(t.test(vals[, 1]))
```

```
##   mean of x     lower     upper level
## 1 0.6978266 0.6958471 0.6998061  0.95
```

```r
(-1/lambda) * log(1 - k/(n + 1))
```

```
## [1] 0.6931472
```

```r
var(vals[, 1])
```

```
## [1] 0.0101982
```

```r
(k/(n + 1)) * (1 - k/(n + 1)) * (1/(n + 2)) * (1/(dexp((-1/lambda) * log(1 -
    k/(n + 1)), rate = lambda))^2)
```

```
## [1] 0.00990099
```

We note that the calculated mean lies within the confidence interval for the value of the empirical mean. Also, the calculated and empirical variances are essentially equal.

Part (c)

The median is the .5th sample quantile, so the variance of the median (as calculated in part (b)) should be approximately:

```
p <- .5 # median is .5th sample quantile

# Note: median of exponential with rate parameter 1 = ln(2)
p*(1-p)/(n*(lambda*exp(-lambda*log(2)))^2); var(vals[,1])
```

```
## [1] 0.01010101
```

```
## [1] 0.0101982
```

```
# Or: p*(1-p)/(n*(dexp(log(2), rate=lambda))^2); var(vals[,1])
```

This approximation is very good as well.

Part (d)

Find the approximate variance of the median of a sample of size n from a $N(\mu, \sigma^2)$ distribution. Compare this variance to the variance of the sample mean.

```
numsim <- 10000

n <- 99 # Number of elements in each sample
k <- 50 # Arbitrary order statistic to examine; here we use median
mean <- 2
sd <- 1
p <- .5 # median is .5th sample quantile

getOrderStatNorm <- function(){
  x <- rnorm(n, mean = mean, sd = sd)
  return(sort(x)[k])
}
medVals <- do(numsim)*getOrderStatNorm()
var(medVals[,1]); p*(1-p)/(n*(dnorm(mean, mean=mean, sd=sd))^2) # For Normal, median = mean
```

```
## [1] 0.01579252
```

```
## [1] 0.01586663
```

```
getMeanNorm <- function(){
  x <- rnorm(n, mean = mean, sd = sd)
  return(mean(x))
}
meanVals <- do(numsim)*getMeanNorm()
var(meanVals[,1]); sd^2/n
```

```
## [1] 0.009897745
```

```
## [1] 0.01010101
```

We notice that the variance of the sample median is greater than the variance of the sample mean.

Problem 10.28

We use a standard Normal as our continuous probability distribution. So, the median = 0.

```
# Insert Jordan's simulation
```

Problem 10.50

Part (a)

For each station, plot flow and occupancy versus time. Explain the patterns you see. Can you deduce from the plots what the days of the week were?

```
p1 = xyplot(Lane.1.Flow ~ Timestamp, data=flow)
p2 = xyplot(Lane.1.Occ ~ Timestamp, data=flow)
print(p1, position = c(0, 0.5, 1, 1), more = TRUE)
print(p2, position = c(0, 0, 1, 0.5))
```

```
p3 = xyplot(Lane.2.Flow ~ Timestamp, data=flow)
p4 = xyplot(Lane.2.Occ ~ Timestamp, data=flow)
print(p3, position = c(0, 0.5, 1, 1), more = TRUE)
print(p4, position = c(0, 0, 1, 0.5))
```

```
p5 = xyplot(Lane.3.Flow ~ Timestamp, data=flow)
p6 = xyplot(Lane.3.Occ ~ Timestamp, data=flow)
print(p5, position = c(0, 0.5, 1, 1), more = TRUE)
print(p6, position = c(0, 0, 1, 0.5))
```

There is a sin-curve-like pattern to the plots. The second and third "bumps" look to be slightly less variable than the other bumps, so the days of the week could be Friday, Saturday, Sunday, Monday, Tuesday, Wednesday.

Part (b)

Compare the flows in the three lanes by making parallel boxplots. Which lane typically serves the most traffic?

```
boxplot(flow$Lane.1.Flow, flow$Lane.2.Flow, flow$Lane.3.Flow, names = c("Lane 1", "Lane 2", "Lane 3"))
```

Lane 2 typically serves the most traffic.

Part (c)

```
xyplot(Lane.1.Flow + Lane.2.Flow + Lane.3.Flow ~ Timestamp, data=flow, auto.key = TRUE)
```

The flow in lane 2 looks to typically be greater than the flow in lane 1, which looks to typically by greater than the flow in lane 3.

Part (d)

```
mean(flow$Lane.1.Occ); median(flow$Lane.1.Occ)
```

```
## [1] 0.06120776

## [1] 0.0478
```

```
mean(flow$Lane.2.Occ); median(flow$Lane.2.Occ)
```

```
## [1] 0.06118247

## [1] 0.05505
```

```
mean(flow$Lane.3.Occ); median(flow$Lane.3.Occ)
```

```
## [1] 0.05123862

## [1] 0.0413
```

In all three lanes, the medians are less than the means. This suggests that the data are skewed right, with some high-valued outliers that increase the mean.

Part (e)

```
histogram(flow$Lane.1.Occ, breaks = 10)
histogram(flow$Lane.1.Occ, breaks = 20)

histogram(flow$Lane.2.Occ, breaks = 10)
histogram(flow$Lane.2.Occ, breaks = 20)

histogram(flow$Lane.3.Occ, breaks = 10)
histogram(flow$Lane.3.Occ, breaks = 20)
```

20 bins looks to give a good representation of the shape of the distribution (skewed right, slightly bimodal). The bimodal nature can be explained by rush hours. Most of the time, when traffic is light, cars do not occupy the same area for very long. Sometimes, though, when traffic is heavy, cars tend to occupy the same area for longer times.

Part (f)

```
xyplot(Lane.1.Occ + Lane.2.Occ + Lane.3.Occ ~ Timestamp, data=flow, auto.key = TRUE)
```

From this plot, we can tell that all 3 lanes tend to have high occupancies at around the same times.

Part (g)

```r
xyplot(Lane.1.Flow ~ Lane.1.Occ, data = flow)
xyplot(Lane.2.Flow ~ Lane.2.Occ, data = flow)
xyplot(Lane.3.Flow ~ Lane.3.Occ, data = flow)
```

The first part of the conjecture seems plausible. When occupancy is low, it tends to increase with flow. Beyond a certain point (around Occ=0.2), however, flow begins to decrease with greater occupancy.

Part (h)

Part (i)

Part (j)

Part (k)

Section i

Section ii

Section iii

Part (l)