Early Government of Manitoba COVID Data Analysis

Quinn Wegner

11/12/2021

Part 1: Data Exploration and Representation

Load in the *Cases_by_Region* data. This dataset contains a summary of all reported Covid-19 cases, organized by RHA (Regional Health Authority: A division of Manitoba into five distinct regions for the purpose of organization of the health care system).

This data has been given to you as-is from the Government of Manitoba's Covid-19 Data Repository. However, it is not in a friendly state for statistical analysis.

Question 1 [5 marks]

Reorganize this data so that each date has one row, and each RHA (as well as the provincial total) has one column for Daily Cases. That is, the columns should have the form: Date | Interlake-Eastern Cases | … | Winnipeg Cases | All |

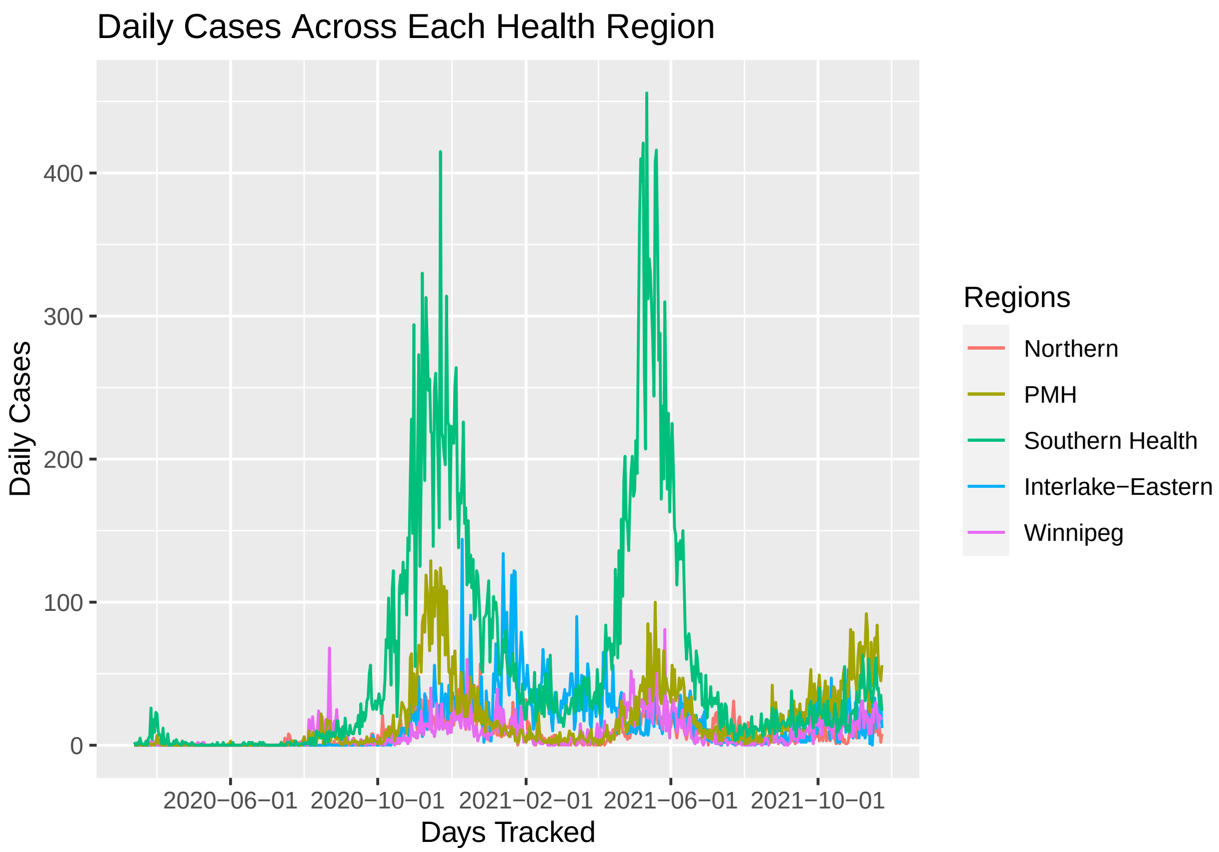
```r
Cases <- read.csv("C:/Users/Quinn/Desktop/COVIDcases/Cases_By_Region.csv",
                  header = TRUE)

titles = c("Date", "Interlake-Eastern", "Northern", "Prairie Mountain Health",
           "Southern Health", "Winnipeg", "All")

new.days <- lapply(Cases$Date, FUN = function(x) {substring(x, 0, 10)})
new.days = unique(new.days)

comp = data.frame(matrix(ncol = 7, nrow = 621))
colnames(comp) = titles

interlake = Cases[which(Cases$RHA == "Interlake-Eastern"), ]
northern = Cases[which(Cases$RHA == "Northern"), ]
pmh = Cases[which(Cases$RHA == "Prairie Mountain Health"), ]
southern = Cases[which(Cases$RHA == "Southern Health-SantÃ© Sud"), ]
winnipeg = Cases[which(Cases$RHA == "Winnipeg"), ]
allofem = Cases[which(Cases$RHA == "All"), ]

for(i in 1:621){
  comp$Date[i] = new.days[i]
  comp$`Interlake-Eastern`[i] = interlake$Daily_Cases[i]
  comp$Northern[i] = northern$Daily_Cases[i]
  comp$`Prairie Mountain Health`[i] = pmh$Daily_Cases[i]
  comp$`Southern Health`[i] = southern$Daily_Cases[i]
  comp$Winnipeg[i] = winnipeg$Daily_Cases[i]
  comp$All[i] = allofem$Daily_Cases[i]
}
head(comp)
```

```
##         Date Interlake-Eastern Northern Prairie Mountain Health Southern Health
## 1 2020/03/13                 0        0                       0               0
## 2 2020/03/14                 0        0                       0               0
## 3 2020/03/15                 0        0                       0               0
## 4 2020/03/16                 1        0                       0               0
## 5 2020/03/17                 0        0                       0               0
## 6 2020/03/18                 0        0                       0               2
##   Winnipeg All
## 1        2   2
## 2        1   1
## 3        0   0
## 4        2   3
## 5        1   1
## 6        5   7
```

Question 2 [15 marks] (Part A)

Create a line graph showing the daily cases (that is, one single graph, with five lines) for each RHA.

```
library(ggplot2)
daily.region.plot <- ggplot(comp, aes(x = as.Date.character(comp$Date))) +
  geom_line(aes(x = as.Date.character(comp$Date), y = comp$`Interlake-Eastern`, color = "blue")) +
  geom_line(aes(x = as.Date.character(comp$Date), y = Northern, color = "red")) +
  geom_line(aes(x = as.Date.character(comp$Date), y = `Prairie Mountain Health`, color = "yellow")) +
  geom_line(aes(x = as.Date.character(comp$Date), y = `Southern Health`, color = "green")) +
  geom_line(aes(x = as.Date.character(comp$Date), y = Winnipeg, color = "purple")) +
  ylab("Daily Cases") +
  xlab("Days Tracked") +
  ggtitle("Daily Cases Across Each Health Region") +
  scale_color_hue(labels = c("Northern", "PMH", "Southern Health",
                             "Interlake-Eastern", "Winnipeg")) +
  guides(col = guide_legend("Regions")) +
  scale_x_date(date_breaks = "4 months")


daily.region.plot
```



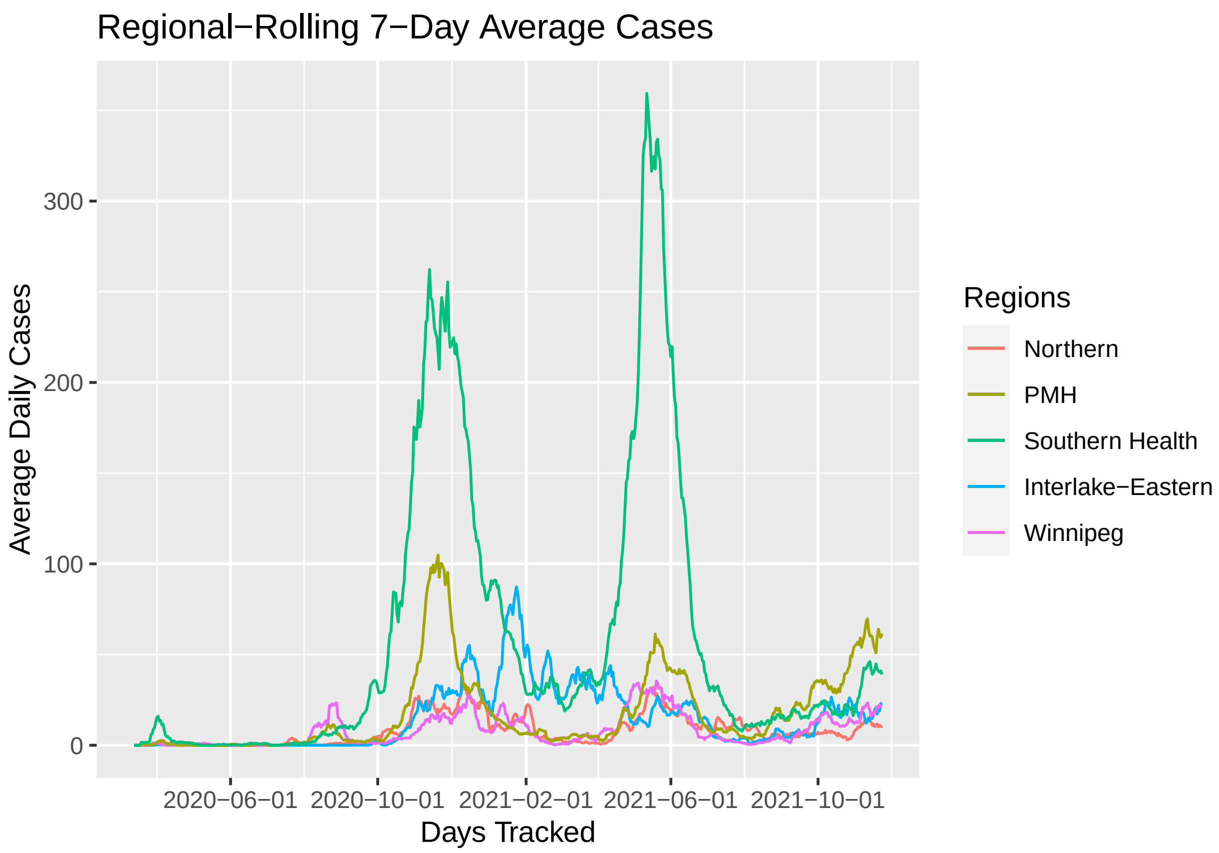
Question 2 (Part B)

Repeat Part A, but replace the regional data will regional rolling 7-day averages.

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(ggplot2)

seven.day.average <- ggplot(comp, aes(x = as.Date.character(comp$Date))) +

  geom_line(aes(x = as.Date.character(comp$Date),
             y = rollmean(comp$`Interlake-Eastern`,
                          k = 7, fill = 0, align = "right"), color = "blue")) +

  geom_line(aes(x = as.Date.character(comp$Date),
             y = rollmean(comp$Northern,
                          k = 7, fill = 0, align = "right"), color = "red")) +
  geom_line(aes(x = as.Date.character(comp$Date),
             y = rollmean(comp$`Prairie Mountain Health`,
                          k = 7, fill = 0, align = "right"), color = "yellow")) +
  geom_line(aes(x = as.Date.character(comp$Date),
                y = rollmean(comp$`Southern Health`,
                             k = 7, fill = 0, align = "right"), color = "green")) +
  geom_line(aes(x = as.Date.character(comp$Date),
                y = rollmean(comp$Winnipeg,
                             k = 7, fill = 0, align = "right"), color = "purple")) +
  ylab("Average Daily Cases") +
  xlab("Days Tracked") +
  ggtitle("Regional-Rolling 7-Day Average Cases") +
  scale_color_hue(labels = c("Northern", "PMH", "Southern Health",
                             "Interlake-Eastern", "Winnipeg")) +
  guides(col = guide_legend("Regions"))+
  scale_x_date(date_breaks = "4 months")

seven.day.average
```

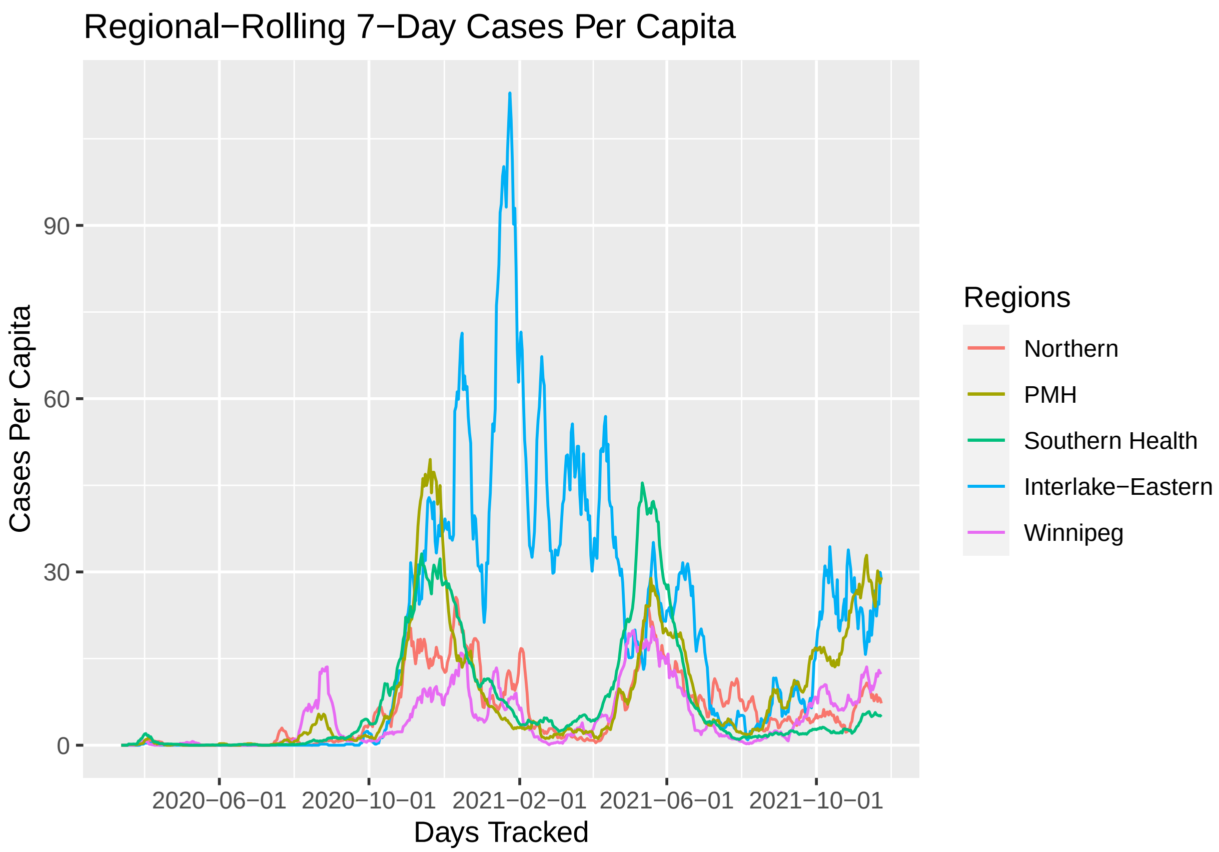


Question 2 (Part C)

Repeat Part B, but replace the regional rolling 7-day average cases with regional rolling 7-day average cases-per-100k (that is, cases-per-capita, multiplied by 100 thousand) Go to here and check Page 17 for a table containing 2020 estimates of the population of each RHA

```
library(zoo)
library(ggplot2)

mod = 100000
interpop = 133834
northpop = 77283
```

```
pmhpop = 172641
southpop = 211896
winipop = 791284



seven.day.capita <- ggplot(comp, aes(x = as.Date.character(comp$Date))) +

  geom_line(aes(x = as.Date.character(comp$Date),
                y = (rollmean(comp$`Interlake-Eastern`,
                              k = 7, fill = 0, align = "right")/interpop)*mod, color = "blue")) +

  geom_line(aes(x = as.Date.character(comp$Date),
                y = (rollmean(comp$Northern,
                              k = 7, fill = 0, align = "right")/northpop)*mod, color = "red")) +
  geom_line(aes(x = as.Date.character(comp$Date),
                y = (rollmean(comp$`Prairie Mountain Health`,
                              k = 7, fill = 0, align = "right")/pmhpop)*mod, color = "yellow")) +
  geom_line(aes(x = as.Date.character(comp$Date),
                y = (rollmean(comp$`Southern Health`,
                              k = 7, fill = 0, align = "right")/southpop)*mod, color = "green")) +
  geom_line(aes(x = as.Date.character(comp$Date),
                y = (rollmean(comp$Winnipeg,
                              k = 7, fill = 0, align = "right")/winipop)*mod, color = "purple")) +
  ylab("Cases Per Capita") +
  xlab("Days Tracked") +
  ggtitle("Regional-Rolling 7-Day Cases Per Capita") +
  scale_color_hue(labels = c("Northern", "PMH", "Southern Health",
                             "Interlake-Eastern", "Winnipeg")) +
  guides(col = guide_legend("Regions")) +
  scale_x_date(date_breaks = "4 months")

seven.day.capita
```



Question 3 [**5 marks**]

Comment on what you see in Question 2, Part C. Which RHA has the highest caseload, relative to their population? Is this changing over time?

The RHA with the highest caseload per-capita appears to be the Interlake-Eastern health region. Over time, the Interlake-Eastern region tracked relatively closely with all other regions until around October of 2020, where their cases per-capita began to exponentially increase through to mid-December of 2020. Following a significant decline leading into 2021, the Interlake-Eastern caseload once again exponentially increased leading into February of 2021. The January to February surge greatly surpassed it's earlier peak-surge that occurred the previous year from October to December. After this January-February peak, the cases per-capita for the Interlake-Eastern region began to decrease significantly, although caseload was still much greater than other regions. By around May of 2021, the Interlake-Eastern region had reached an equivalent per-capita caseload to the other four regions.

Part 2: Estimation and Inference

One important quantity that is important to understand in the study of viruses is the *incubation period*, the time between infection and development of symptoms. It is important to know what the incubation period of a virus is as it allows for a better understanding of when an individual may have been infected, among other reasons.

Attached is a dataset of 38 individuals, for whom the date of symptom onset is known, as well as the potential time of exposure.We are going to focus on the Incubation midpoint.

Question 4 [5 marks]

Determine the sample mean and median incubation time, as well as the sample 95th percentile of incubation times, based on this dataset.

```
Incubation <- read.csv("C:/Users/Quinn/Desktop/COVIDcases/Incubation.csv",
                       header = TRUE)

inc.mid.mean = mean(Incubation$Incubation.Midpoint)
inc.mid.mean
```

```
## [1] 5.065789
```

```
inc.mid.median = median(Incubation$Incubation.Midpoint)
inc.mid.median
```

```
## [1] 4.5
```

```
percentile.95 = quantile(Incubation$Incubation.Midpoint, 0.95)
percentile.95
```

```
##   95%
## 12.15
```

When modelling data, one popular choice is to use a distribution called the *Weibull* distribution, which is a continuous random variable with the density below:

$$f(x; k, \lambda) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, \quad x \geq 0, \quad \lambda > 0, k > 0.$$

This distribution is popular because it can fit multiple types of data with the appropriate choices of k and λ. Of course, what is required is to estimate k and λ from the sample data.

We will assume that the Incubation data follows a Weibull distribution, and our task is to estimate k and λ. However, for the Weibull distribution, this problem cannot be solved with pen and paper.

Question 5 [10 marks] (Part A) Use the *multiroot* and *optim* functions to determine the MOM and MLE estimators of k and λ.

```
library(rootSolve)
options(scipen=999)

#first moment
my.wei.m1 = function(my.par){
  return((my.par[2]^1)*gamma(1+1/my.par[1]))
}

#second moment
my.wei.m2 = function(my.par){
  return((my.par[2]^2)*gamma(1+2/my.par[1]))
}
```

```r
my.g = function(my.par, sample.data){
  eq.1 = mean(sample.data) - my.wei.m1(my.par)
  eq.2 = mean(sample.data2) - my.wei.m2(my.par)
  return(c(eq.1, eq.2))
}

my.par = c(2, 6)
sample.data = Incubation$Incubation.Midpoint
sample.data2 = sample.data^2

multiroot(my.g, start = c(1.5,5), sample.data = sample.data)$root
```

```
## [1] 1.745955 5.687184
```

```r
#----------------------------------------------

sample.data = Incubation$Incubation.Midpoint

my.test = function(par, sample.data){
  #return(-sum(log(par[1]/par[2])
  #            +par[1]*log(sample.data/par[2])
  #            -log(sample.data/par[2])
  #            -(sample.data/par[2])^par[1]))
  return(-sum(log({par[1]/par[2]}
                *{{sample.data/par[2]}^{par[1]-1}}
                *{exp(-{sample.data/par[2]}^par[1])})))

}
my.form = function(par, sample.data){
  return(-sum(log({par[1]/par[2]}
                *{{sample.data/par[2]}^{par[1]-1}}
                *{exp(-{sample.data/par[2]}^par[1])})))
}

optim(par = c(1.5, 5), my.form, sample.data = sample.data)
```

```
## $par
## [1] 1.803006 5.723942
##
## $value
## [1] 90.97362
##
## $counts
## function gradient
##       47       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

The mean of Weibull distribuion is can be shown to be equal to

$$\lambda\Gamma\Big(1+\frac{1}{k}\Big).$$

In general, the rth moment is given by

$$E[X^r] = \mu_r = \lambda^r \Gamma\Big(1 + \frac{r}{k}\Big).$$

The median is

$$\lambda(\ln 2)^{1/k}$$

Finally, the CDF of a Weibull distribution is given by

$$F(x) = 1 - e^{-(x/\lambda)^k}.$$

Question 5 (Part B)

For the pair of MLE estimates, and the pair of MOM estimates, estimate the mean incubation period, as well as the median incubation period, and the 95th percentile of incubation time.

```
mom.k = 1.745955
mom.lam = 5.687184

mom.mean = mom.lam*gamma(1+1/mom.k)
mom.median = mom.lam*log(2)^(1/mom.k)

my.inv.mom = function(x){
  return(mom.lam*(-log(1-x))^(5377/9388))
}
mom.95 = my.inv.mom(0.95)

"MOM Mean:"
```

```
## [1] "MOM Mean:"
```

```
mom.mean
```

```
## [1] 5.06579
```

```
"MOM Median:"
```

```
## [1] "MOM Median:"
```

```
mom.median
```

```
## [1] 4.610305
```

```
"MOM 95th Percentile:"
```

```
## [1] "MOM 95th Percentile:"
```

```
mom.95
```

```
## [1] 10.66143
```

```
mle.k = 1.803006
mle.lam = 5.723942

my.inv.mle = function(x){
  return(mle.lam*(-log(1-x))^(4990/8997))
}
mle.95 = my.inv.mle(0.95)

mle.mean = mle.lam*gamma(1+1/mle.k)
mle.median = mle.lam*log(2)^(1/mle.k)
```

```
"MLE Mean:"
```

```
## [1] "MLE Mean:"
```

```
mle.mean
```

```
## [1] 5.089817
```

```
"MLE Median:"
```

```
## [1] "MLE Median:"
```

```
mle.median
```

```
## [1] 4.671027
```

```
"MLE 95th Percentile:"
```

```
## [1] "MLE 95th Percentile:"
```

```
mle.95
```

```
## [1] 10.51908
```

When faced with multiple choices for estimating parameters, it is important to know which estimation technique is better.

Question 6 [10 marks] (Part A)

Use simulation to determine the bias, variance, and MSE of the MLE and the MOM estimators, for $n = 38$. Run at least 20 000 iterations for each simulation.

```
library(rootSolve)
mom.k = 1.745955
mom.lam = 5.687184

mom.values = c()

my.wei.m1 = function(my.par){
  return((my.par[2]^1)*gamma(1+1/my.par[1]))
}
my.wei.m2 = function(my.par){
  return((my.par[2]^2)*gamma(1+2/my.par[1]))
}

my.g = function(my.par, sample.data){
  eq.1 = mean(sample.data) - my.wei.m1(my.par)
  eq.2 = mean(sample.data^2) - my.wei.m2(my.par)
  return(c(eq.1, eq.2))
}
my.par = c(1.745955, 5.687184)

sample.size = 38
k.mom.values = c()
lam.mom.values = c()
B = 20000

for(i in 1:B){
  sample.data = rweibull(sample.size, my.par[1], my.par[2])
  mom.values = multiroot(my.g, start = c(mom.k,mom.lam), sample.data = sample.data)$root
  mom.frame = data.frame(mom.values)
  k.mom.values[i] = mom.frame[1,1]
  lam.mom.values[i] = mom.frame[2,1]
}

for(j in 1:B){
  if(k.mom.values[j] >= 100){
    k.mom.values[j] = k.mom.values[-k.mom.values[j]]
  }
  if(lam.mom.values[j] >= 100){
    lam.mom.values[j] = lam.mom.values[-lam.mom.values[j]]
  }
}
"Mean of MOM.k"
```

```
## [1] "Mean of MOM.k"
```

```
mean(k.mom.values)
```

```
## [1] 1.812103
```

```
"Var of MOM.k"
```

```
## [1] "Var of MOM.k"
```

```
var(k.mom.values)
```

```
## [1] 0.05728257
```

```
"MSE of MOM.k"
```

```
## [1] "MSE of MOM.k"
```

```
(mean(k.mom.values) - mom.k)^2 + var(k.mom.values)
```

```
## [1] 0.06165808
```

```
"Mean of MOM.lambda"
```

```
## [1] "Mean of MOM.lambda"
```

```
mean(lam.mom.values)
```

```
## [1] 5.674617
```

```
"Var of MOM.lambda"
```

```
## [1] "Var of MOM.lambda"
```

```
var(lam.mom.values)
```

```
## [1] 0.3080976
```

```
"MSE of MOM.lambda"
```

```
## [1] "MSE of MOM.lambda"
```

```
(mean(lam.mom.values) - mom.lam)^2 + var(lam.mom.values)
```

```
## [1] 0.3082556
```

```
"We can see that the mean and variance of our estimates are biased, where both k
and lambda are overestimates. Further, the MSEs are consistent with the
overestimation conclusion for the MOM estimates."
```

```
## [1] "We can see that the mean and variance of our estimates are biased, where both k \nand lambda ar
```

```
mle.k = 1.803006
mle.lam = 5.723942

mle.values = c()
k.mle.values = c()
lam.mle.values = c()

for(q in 1:B){
  sample.data = rweibull(sample.size, mle.k, mle.lam)
  mle.values[q] = optim(par = c(mle.k, mle.lam), my.test, sample.data = sample.data)
}
mle.frame = data.frame(mle.values)

k.mle.row = c()
for(g in 1:B){
  k.mle.row[g] = mle.frame[1,g]
}

lam.mle.row = c()
for(k in 1:B){
  lam.mle.row[k] = mle.frame[2,k]
}

for(h in 1:B){
  if(k.mle.row[h] >= 100){
    k.mle.row[h] = k.mle.row[-k.mle.row[h]]
  }
  if(lam.mle.row[h] >= 100){
    lam.mle.row[h] = lam.mle.row[-lam.mle.row[h]]
  }
}
"Mean of MLE.k"
```

```
## [1] "Mean of MLE.k"
```

```
mean(k.mle.row)
```

```
## [1] 1.871239
```

```
"Var of MLE.k"
```

```
## [1] "Var of MLE.k"
```

```
var(k.mle.row)
```

```
## [1] 0.06223329
```

```
"MSE of MLE.k"
```

```
## [1] "MSE of MLE.k"
```

```
(mean(k.mle.row) - mle.k)^2 + var(k.mle.row)
```

```
## [1] 0.06688903
```

```
"Mean of MLE.lambda"
```

```
## [1] "Mean of MLE.lambda"
```

```
mean(lam.mle.row)
```

```
## [1] 5.718623
```

```
"Var of MLE.lambda"
```

```
## [1] "Var of MLE.lambda"
```

```
var(lam.mle.row)
```

```
## [1] 0.2887743
```

```
"MSE of MLE.lambda"
```

```
## [1] "MSE of MLE.lambda"
```

```
(mean(lam.mle.row) - mle.lam)^2 + var(lam.mle.row)
```

```
## [1] 0.2888026
```

```
"We can see that the mean and variance of our estimates are biased, where both k
and lambda are overestimates. Further, the MSEs are consistent with the
overestimation conclusion for the MLE estimates."
```

```
## [1] "We can see that the mean and variance of our estimates are biased, where both k \nand lambda are
```

Question 6 (Part B)

By these metrics, which estimation technique is superior?

From the results listed above, the MLE estimates for k and lambda appear to be more accurate. The MSE of lambda for the MLE estimates is significantly lower than the MSE of the MOM estimates. Further, differences in variance between the MLE and MOM estimates is small, thereby reinforcing the conclusion that the MLE estimates are superior as a result of the MSE of the MLE.lambda being smaller.

It is important to understand the distributions of our estimators, so we know how much uncertainty is present within the estimation.

Question 7 [10 marks] (Part A)

Use bootstrapping to approximate the bias and the variance of your MLE estimates. Run $B = 5000$ iterations.

```
data = Incubation$Incubation.Midpoint
mle.k.estimate = 1.803006
mle.lam.estimate = 5.723942
mle.mean.est = 5.089817
mle.median.est = 4.671027

B = 5000
boot.vec = c()
for(b in 1:B){
  data.boot = sample(data, 38, replace = TRUE)
  boot.vec[b] = optim(par = c(1.803006, 5.723942), my.test, sample.data = data.boot)
}
boot.frame = data.frame(boot.vec, row.names = c("row1", "row2"))

k.row = c()
for(i in 1:B){
  k.row[i] = boot.frame[1,i]
}

lam.row = c()
```

```r
for(j in 1:B){
  lam.row[j] = boot.frame[2,j]
}

boot.bias.k = mean(k.row) - mle.k.estimate
"Bias for mean estimate:"
```

```
## [1] "Bias for mean estimate:"
```

```r
boot.bias.k
```

```
## [1] 0.06411343
```

```r
boot.k.var = var(k.row)
"Variance for MLE mean estimate:"
```

```
## [1] "Variance for MLE mean estimate:"
```

```r
boot.k.var
```

```
## [1] 0.04650241
```

```r
boot.bias.lam = mean(lam.row) - mle.lam.estimate
"Bias for mean estimate:"
```

```
## [1] "Bias for mean estimate:"
```

```r
boot.bias.lam
```

```
## [1] -0.008881918
```

```r
boot.lam.var = var(lam.row)
"Variance for MLE mean estimate:"
```

```
## [1] "Variance for MLE mean estimate:"
```

```r
boot.lam.var
```

```
## [1] 0.2905898
```

Question 7 (Part B) Thus, the bias-corrected MLE estimates of k and λ are

```r
"MLE, k:"
```

```
## [1] "MLE, k:"
```

```r
mle.k.estimate - boot.bias.k
```

```
## [1] 1.738893
```

```r
"MLE, lambda:"
```

```
## [1] "MLE, lambda:"
```

```r
mle.lam.estimate - boot.bias.lam
```

```
## [1] 5.732824
```

Question 7 (Part C) Use bootstrap to approximate a 95% confidence interval for k and λ from your MLE estimators.

```r
mle.k.quantile = quantile(k.row, c(0.025, 0.975))
"MLE, k CI:"
```

```
## [1] "MLE, k CI:"
```

```
mle.k.quantile
```

```
##     2.5%    97.5%
## 1.533596 2.380248
```

```
mle.lam.quantile = quantile(lam.row, c(0.025, 0.975))
"MLE, lambda CI:"
```

```
## [1] "MLE, lambda CI:"
```

```
mle.lam.quantile
```

```
##     2.5%    97.5%
## 4.722869 6.810328
```

Question 8 [10 marks] Run a goodness-of-fit test to determine whether the Weibull model is appropriate, at the 5% level of significance. (Both parameters are unknown and must be estimated from data.)

```
data = Incubation$Incubation.Midpoint
n = length(data)
k = 2
my.k = 1.803006
my.lam = 5.723942

c = 38
bounds = qweibull((0:c)/c, my.k, my.lam)

counts.o = c()
counts.e = c()

for(i in 1:length(bounds)-1){
  counts.o[i] = sum((data >= bounds[i])*(data < bounds[i+1]))
  counts.e[i] = n*(pweibull(bounds[i + 1], my.k, my.lam) - pweibull(bounds[i], my.k, my.lam))
}

chisq.stat = sum((counts.o - counts.e)^2/counts.e)
chisq.stat
```

```
## [1] 82
```

```
v = c - (k + 1)
v
```

```
## [1] 35
```

```
1 - pchisq(chisq.stat, v)
```

```
## [1] 0.0000121712
```

```
"Ho: X!~weibull(k, lambda) vs. Ha: X~weibull(k,lambda)

Since the p-value of the test is <0.05, we reject Ho and conclude that the
weibull distribution is appropriate for modelling our data."
```

```
## [1] "Ho: X!~weibull(k, lambda) vs. Ha: X~weibull(k,lambda)\n\nSince the p-value of the test is <0.05
```

NOTE: This incubation data is quite old, and these estimation techniques could use a good amount of improvement (using techniques you will learn as you continue to study statistics).

Do not take these estimates over those published in peer-reviewed journals or those recommended by Physicians or the Public Health Agency of Canada.