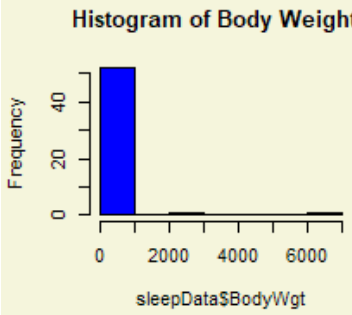
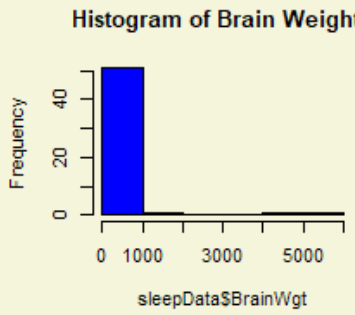
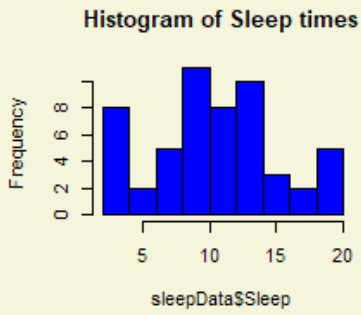
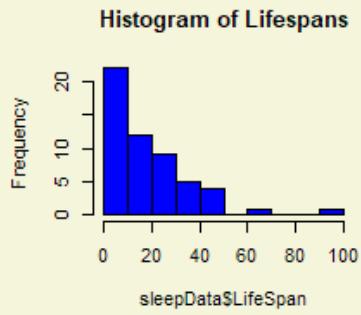
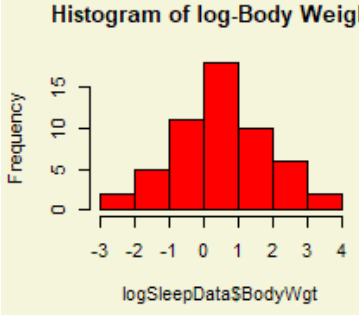
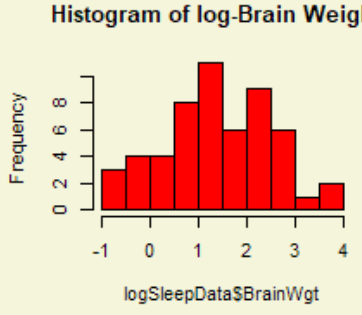

Histogram of Body Weights
Frequency
0
20
40
0
2000
4000
6000
sleepData$BodyWgt

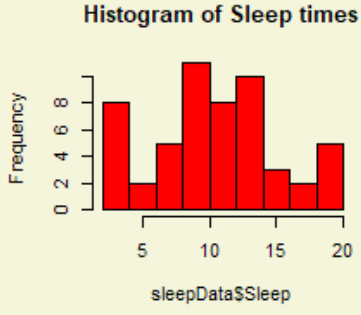

Histogram of Brain Weights
Frequency
0
20
40
0
1000
3000
5000
sleepData$BrainWgt

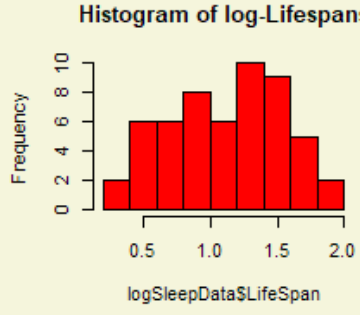

Histogram of Sleep times
Frequency
0
2
4
6
8
5
10
15
20
sleepData$Sleep

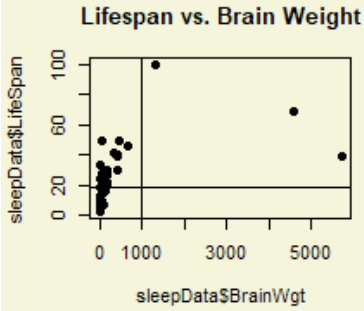

Histogram of Lifespans
Frequency
0
5
10
20
0
20
40
60
80
100
sleepData$LifeSpan

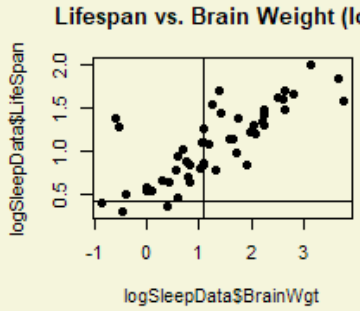

Histogram of log-Body Weights
Frequency
0
5
10
15
-3
-2
-1
0
1
2
3
4
logSleepData$BodyWgt

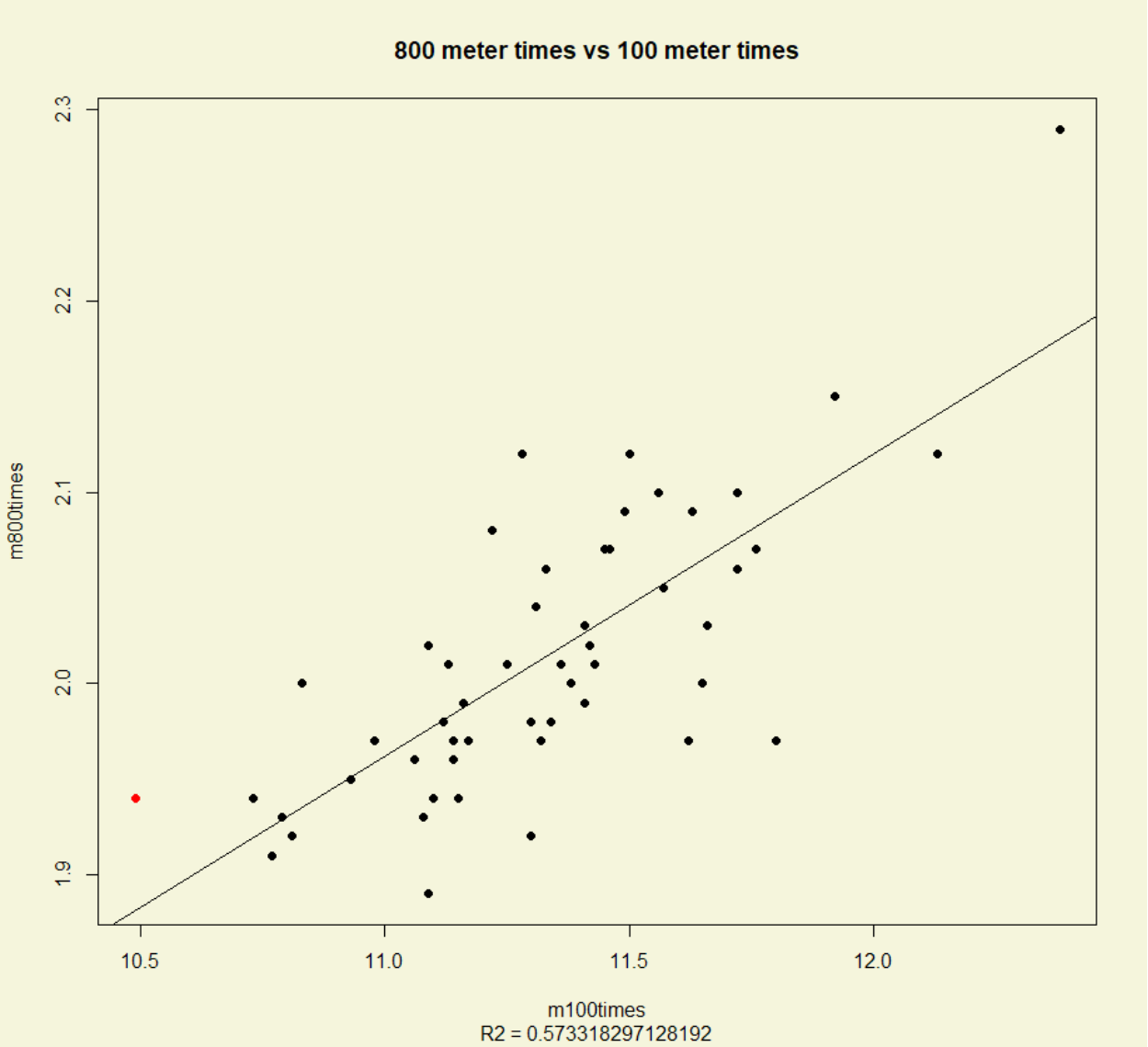

Histogram of log-Brain Weights
Frequency
0
2
4
6
8
-1
0
1
2
3
4
logSleepData$BrainWgt


Histogram of Sleep times
Frequency
0
2
4
6
8
5
10
15
20
sleepData$Sleep


Histogram of log-Lifespans
Frequency
0
2
4
6
8
10
0.5
1.0
1.5
2.0
logSleepData$LifeSpan


Lifespan vs. Brain Weight
sleepData$LifeSpan
0
20
60
100
0
1000
3000
5000
sleepData$BrainWgt


Lifespan vs. Brain Weight (log)
logSleepData$LifeSpan
0.5
1.0
1.5
2.0
-1
0
1
2
3
logSleepData$BrainWgt


800 meter times vs 100 meter times
m800times
2.3
2.2
2.1
2.0
1.9
10.5
11.0
11.5
12.0
m100times
R2 = 0.573318297128192

PART I

```
#clear current graphics
dev.off()

sleepData = read.table("Sleepdata.txt", header = TRUE, row.names = "Species")
par(mfrow = c(4,4), bg = "beige")

#histograms
logSleepData = log10(sleepData)
hist(sleepData$BodyWgt, main = "Histogram of Body Weights", col = "blue")
hist(sleepData$BrainWgt, main = "Histogram of Brain Weights", col = "blue")
hist(sleepData$Sleep, main = "Histogram of Sleep times", col = "blue")
hist(sleepData$LifeSpan, main = "Histogram of Lifespans", col = "blue")

#log histograms (except sleep b/c no transform)
hist(logSleepData$BodyWgt, main = "Histogram of log-Body Weights", col = "red")
hist(logSleepData$BrainWgt, main = "Histogram of log-Brain Weights", col = "red")
hist(sleepData$Sleep, main = "Histogram of Sleep times", col = "red")
hist(logSleepData$LifeSpan, main = "Histogram of log-Lifespans", col = "red")

#scatterplots for lifespan vs. brain weight
plot(sleepData$BrainWgt, sleepData$LifeSpan, main = "Lifespan vs. Brain Weight", pch = 16)
abline(v = sd(sleepData$BrainWgt)); abline(h = sd(sleepData$LifeSpan))
plot(logSleepData$BrainWgt, logSleepData$LifeSpan, main = "Lifespan vs. Brain Weight (log)", pch = 16)
abline(v = sd(logSleepData$BrainWgt)); abline(h = sd(logSleepData$LifeSpan))

#Brainwgt proportions
```

```
brainWgtProportion = length(sleepData$BrainWgt[sleepData$BrainWgt > sd(sleepData$BrainWgt) -
2*sd(sleepData$BrainWgt) & sleepData$BrainWgt < sd(sleepData$BrainWgt) +
2*sd(sleepData$BrainWgt)])/length(sleepData$BrainWgt)


logBrainWgtProportion = length(logSleepData$BrainWgt[logSleepData$BrainWgt >
sd(logSleepData$BrainWgt) - 2*sd(logSleepData$BrainWgt) & logSleepData$BrainWgt <
sd(logSleepData$BrainWgt) + 2*sd(logSleepData$BrainWgt)])/length(logSleepData$BrainWgt)


#LifeSpan proportions

lifeSpanProportion = length(sleepData$LifeSpan[sleepData$LifeSpan > sd(sleepData$LifeSpan) -
2*sd(sleepData$LifeSpan) & sleepData$LifeSpan < sd(sleepData$LifeSpan) +
2*sd(sleepData$LifeSpan)])/length(sleepData$LifeSpan)


logLifeSpanProportion = length(logSleepData$LifeSpan[logSleepData$LifeSpan >
sd(logSleepData$LifeSpan) - 2*sd(logSleepData$LifeSpan) & logSleepData$LifeSpan <
sd(logSleepData$LifeSpan) + 2*sd(logSleepData$LifeSpan)])/length(logSleepData$LifeSpan)


#find correlations

brainCor = cor(sleepData$BrainWgt, sleepData$LifeSpan)

logBrainCor = cor(logSleepData$BrainWgt, logSleepData$LifeSpan)


#find least squares lines

reg = lm(sleepData$LifeSpan ~ sleepData$BrainWgt)

logReg = lm(logSleepData$LifeSpan ~ logSleepData$BrainWgt)


predictedLifeSpans = reg$fitted.values

names(predictedLifeSpans) = rownames(sleepData)

predic tedHomoSapiens = predictedLifeSpans["Homo sapiens"]
```

PART II

```
#clear current graphics, set parameters

graphics.off()

par(bg = "beige")


#read trackadata

trackData = read.csv("TrackRecords.csv", row.names = 1)


#store means of races

trackMeans = c(mean(trackData$m100), mean(trackData$m200), mean(trackData$m400),
mean(trackData$m800), mean(trackData$m1500), mean(trackData$m3000),
mean(trackData$Marathon))

names(trackMeans) = c("m100", "m200", "m400", "m800", "m1500", "m3000", "Marathon")


#store medians of races

trackMedians = c(median(trackData$m100), median(trackData$m200), median(trackData$m400),
median(trackData$m800), median(trackData$m1500), median(trackData$m3000),
median(trackData$Marathon))

names(trackMedians) = c("m100", "m200", "m400", "m800", "m1500", "m3000", "Marathon")


#store Standard Deviations of races

trackStdDevs = c(sd(trackData$m100), sd(trackData$m200), sd(trackData$m400), sd(trackData$m800),
sd(trackData$m1500), sd(trackData$m3000), sd(trackData$Marathon))

names(trackStdDevs) = c("m100", "m200", "m400", "m800", "m1500", "m3000", "Marathon")


#data oustide of 2 sd's

more2SDm100 = trackData[which(trackData$m100 > trackMeans["m100"]+2*trackStdDevs["m100"]), ]

less2SDm800 = trackData[which(trackData$m800 < trackMeans["m800"]-2*trackStdDevs["m800"]), ]


#data in top 10% and bottom 10%
```

```
top10pm100cutoff = sort(trackData$m100)[floor(.1*length(trackData$m800))]

top10pm100 = trackData[which(trackData$m100 >= top10pm100cutoff), ]

bot10pm800cutoff = sort(trackData$m800)[floor(.1*length(trackData$m800))]

bot10pm800 = trackData[which(trackData$m800 <= bot10pm800cutoff), ]


#set names of data subsets

m100times = trackData$m100

names(m100times) <- rownames(trackData)

m800times = trackData$m800

names(m800times) = rownames(trackData)


#plot 800 times vs 100 times + lin. regression line

plot(m100times, m800times, col = ifelse(names(m100times) == "USA", "red", "black"), pch  = 16, sub = paste("R2 =", summary(lm(m800times ~ m100times))$r.squared), main = "800 meter times vs 100 meter times")

abline(lm(m800times ~ m100times))
```