

Data File Used in this Analysis:

```
# M3070 - 1      Geyser Data   Treibergs
# Oct. 7, 2010
# Data from Navidi, "Principles of Statistics for Engineers and Scientists"
# McGraw Hill, 2010
# The following are durations in minutes of 40 consecutive time intervals
# between eruptions of the Old Faithful geyser in Yellowstone national park
#
Dormant
91
51
79
53
82
51
76
82
84
53
86
51
85
45
88
51
80
49
82
75
73
67
68
86
72
75
75
66
84
70
79
60
86
71
67
81
76
83
76
```

R Session:

R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.31 (5538) powerpc-apple-darwin8.11.1]

[Workspace restored from /Users/andrejstreibergs/.RData]

```
> # We find Confidence Intervals for the Mean of Geyser Data
> # using Bootstrapping. We compare with the CI's obtained
> # via z-test and t-test.
```

```
>
```

```
>
```

```
> tt = read.table("M3074GeyserData.txt",header=TRUE);tt
```

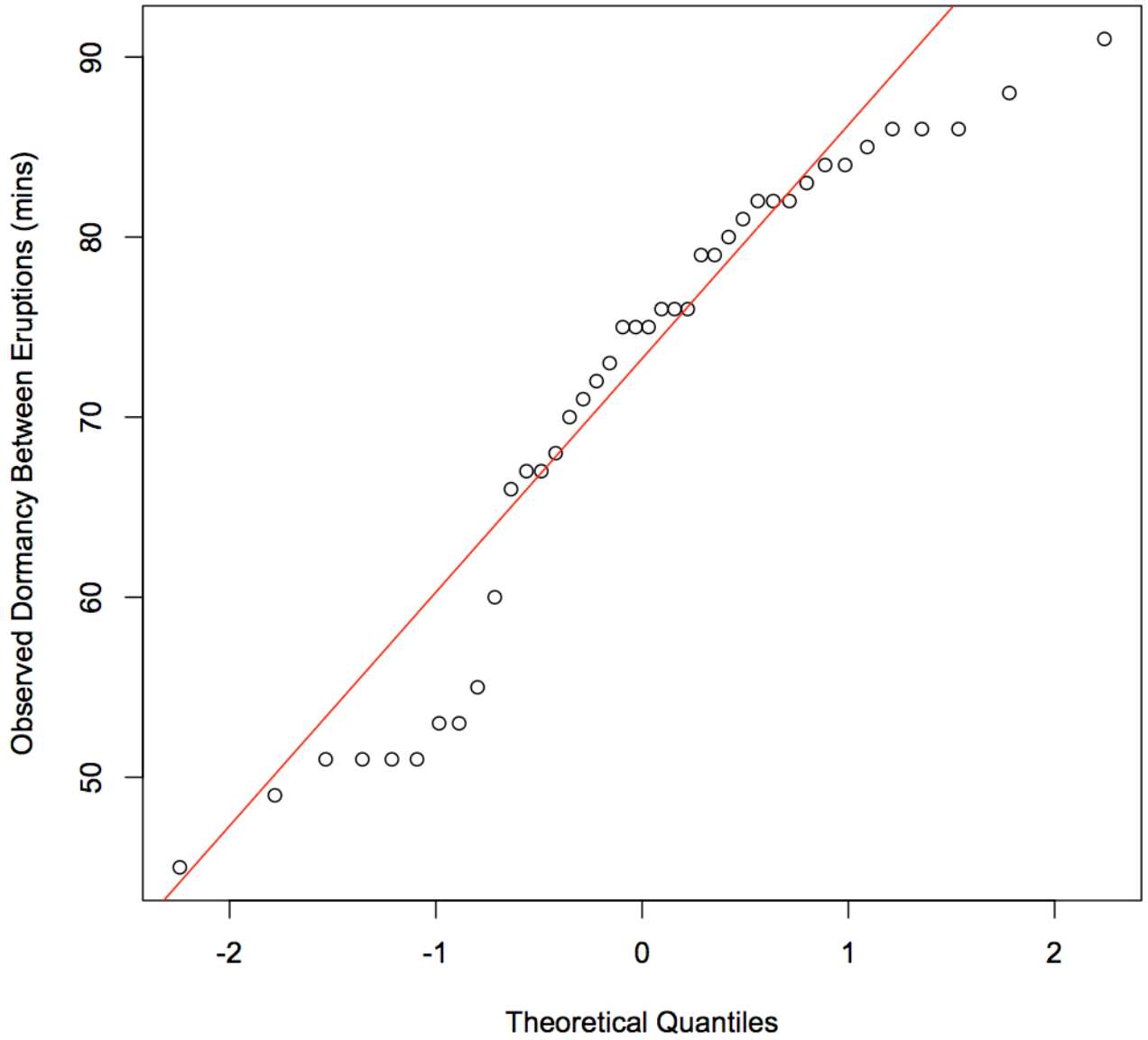
```
  Dormant
1      91
2      51
3      79
4      53
5      82
6      51
7      76
8      82
9      84
10     53
11     86
12     51
13     85
14     45
15     88
16     51
17     80
18     49
```

```

19      82
20      75
21      73
22      67
23      68
24      86
25      72
26      75
27      75
28      66
29      84
30      70
31      79
32      60
33      86
34      71
35      67
36      81
37      76
38      83
39      76
40      55
> attach(tt)
>
>
> ##### CHECK FOR NORMALITY #####
>
> # Normal QQ-Plot shows ditribution looks light-tailed.
>
> qqnorm(Dormant,main="QQ Plot to Chech Normality",
+ ylab="Observed Dormancy Between Eruptions (mins)")
> qqline(Dormant,col=2)
> #
> # M3074GeyserQQ.pdf

```

QQ Plot to Check Normality



```

> # This code was adapted from Crawley's "Statistics: An Introduction Using R,"
> # Wiley, 2005, p47.
>
> ##### CI's FOR MEAN USING USUAL Z-TEST AND T-TEST #####
>
> # First we obtain the CI's using the t-test and z-test.
> # For the z.test
>
> # Let us compute the 0.05 two sided CI for the mean of Dormancy Times.
> # By the texts rule of thumb, it is ok to assume the sample mean is
> # normally distributed if n>30. Thus with our sample, the CI is
>
> n = length(Dormant); s=sd(Dormant); xbar=mean(Dormant)
> c(n,xbar,s)
[1] 40.00000 71.60000 13.08709
> alpha = 0.05; zstar = qnorm(1-alpha/2)
> SE = s/sqrt(n)
> c(xbar-zstar*SE,xbar+zstar*SE)

[1] 67.54434 75.65566

>
>
>
> # But since we are using the sample standard deviation, the CI is in fact
> # slightly wider, and is given from the t-distribution. We can get the CI
> # the same way, or we can call the t.test routine that computes the t-test
> # based CI.
>
> tstar = qt(1-alpha/2,df=n-1)
> c(xbar-tstar*SE,xbar+tstar*SE)

[1] 67.41455 75.78545

>
> # Alternative code using t.test:
>
> t.test(Dormant)

One Sample t-test

data:  Dormant
t = 34.6019, df = 39, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 67.41455 75.78545
sample estimates:
mean of x
 71.6

```

```

> ##### CI's FOR MEAN USING BOOTSTRAPPING #####
>
> # Now let us compute the Bootstrapped CI from the data.
> # The idea is that we replace the Dormancy population
> # with the sample Dormant and select random samples from it
> # to simulate random samples from the Dormancy population. e.g.
> # a couple of randm samples of size n=40 with replacement are
> sample(Dormant,40,replace=T)
  [1] 55 51 51 83 84 76 76 51 75 51 91 85 73 51 76 73 86 53 82 51 51 82 85 75
 [25] 91 70 51 71 72 76 55 66 76 75 86 85 55 85 86 55
> sample(Dormant,40,replace=T)
  [1] 86 81 82 51 71 53 73 68 84 86 85 51 76 75 82 53 51 51 83 55 72 85 86 53
 [25] 82 85 84 88 82 71 75 75 82 80 51 51 85 68 68 76
> sample(Dormant,40,replace=T)
  [1] 71 60 70 84 51 83 86 79 82 66 84 49 68 76 53 53 86 67 79 80 75 53 82 76
 [25] 80 70 67 86 83 88 71 86 85 75 86 53 84 76 83 53
>
> # Note that each sample is different and that repeats are allowed.
>
> # We take 10000 random samples of size n=40 from the data (with replacement
> # and record the means. From the means we extract the .025 and .975 quantiles
> # that five the Bootstrapped CI
>
> a <- numeric(10000)
> for( i in 1:10000){ a[i] <- mean(sample(Dormant,n,replace=T)) }
> quantile(a,c(0.025,0.975))
   2.5%   97.5%
67.45000 75.52563
>
> # Close!!!!!!!!!!!!!! Note that if we do it again, we get a slightly different CI
>
> for( i in 1:10000){ a[i] <- mean(sample(Dormant,n,replace=T)) }
> quantile(a,c(0.025,0.975))
   2.5%   97.5%
67.500 75.525
>

```

```

> ##### BOOTSTRAPPED CI'S FOR MEAN FOR VARIOUS SAMPLE SIZES #####
>
>
> # Note that we could have just as well sampled any k from Dormant
> # instead of k=n and obtain the CI for the population mean based on means
> # with k observations. We plot the Bootstrapped CI's vs z-test vs t-test
> # CI's for any number k < n.
> #
> # Set up the plotting window.
> plot(c(0,40),c(40,90),type="n",xlab="Sample Size",ylab="Bootstrapped CI")
>
> # Look through several k's. For each take 10000 samples from Dormant of
> # size k (with replacement!) From the 10000 sample means, extract the
> # .025 and .975 quantiles and plot as circles.
>
> for (k in seq(5,40,3)){
+           a <- numeric(10000)
+           for( i in 1:10000){
+               a[i] <- mean(sample(Dormant,k,replace=T))
+           }
+           points(c(k,k),quantile(a,c(0.025,0.975)),type="b",col=4)
+       }
> # Add the curves corresponding to the z-test CI and the t-test CI
> xv <- seq(5,40,0.1)
> yv <- mean(Dormant)+1.96*sqrt(var(Dormant)/xv)
> lines(xv,yv,col=2)
> yv <- mean(Dormant)-1.96*sqrt(var(Dormant)/xv)
> lines(xv,yv,col=2)
> yv <- mean(Dormant)+qt(.975,xv)*sqrt(var(Dormant)/xv)
> lines(xv,yv,col=5)
> yv <- mean(Dormant)-qt(.975,xv)*sqrt(var(Dormant)/xv)
> lines(xv,yv,col=5)
> # Add a line corresponding to the mean and the legend.
> mu <- mean(Dormant)+0*xv
> lines(xv,mu,col=6,lty=3)
> legend(20,55,c("Normal CI","T-test CI"),fill=c("red","cyan"))
>
> # In this run, the n=40 sample CI was
> quantile(a,c(0.025,0.975))
  2.5%  97.5%
67.3421 75.5000
>

```

Bootstrapping CI for Geyser Data

