**Lab 8 – Normality testing; confidence intervals**

**To submit: answers to all numbered questions. When the question asks you to write code or create graphs, submit the code and/or graphs in the Word document as part of your answer. Also submit a single .R file that contains all of your code.**

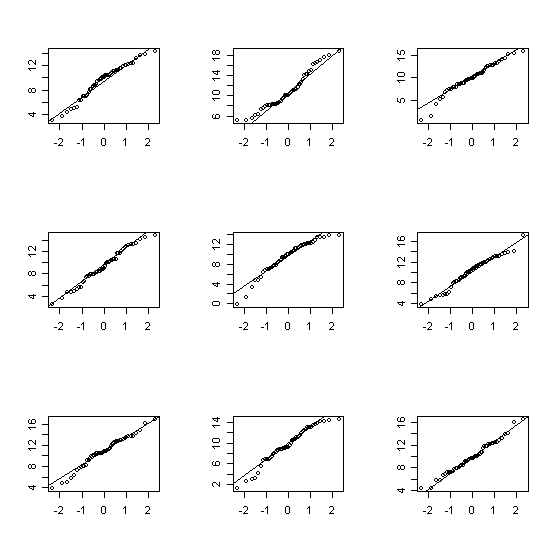
We saw in our last lab that sample means are distributed normally if the population follows a normal distribution, or if the sample is “large enough”. When this is the case, we can construct confidence intervals using the formulas we encountered in class.

In the examples we saw in class, we were told that the population was distributed normally. In real life, however, we often have to determine for ourselves whether this is the case. One way to do that is by producing a **quantile-quantile plot**, which we can do easily in R.

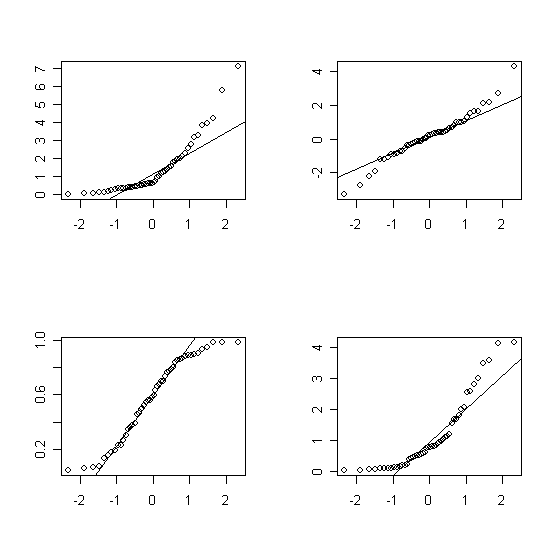
A quantile-quantile plot, or **QQ plot**, is a graph specifically designed to check for normality. The plot usually requires a minimum of 15 – 20 data points in order to accurately reveal normality.

If the data comes from a normal distribution the points should form a line with positive slope.

Examples of QQ plots of normal-ish data:



Examples of QQ plots of non-normal data:

We will do a few examples before delving into the theory.

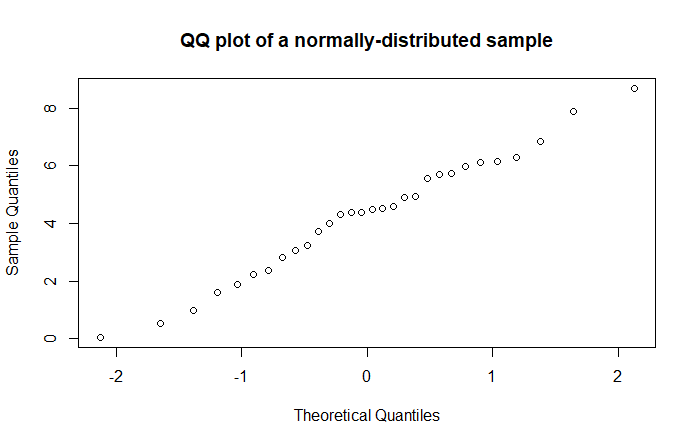
# Quantile-quantile plots of normal data

We will begin by generating a sample that comes from a normal distribution. We have generated similar samples before, using the **rnorm** function. Here are thirty data points that come from a normally-distributed dataset with mean 4 and standard deviation 2, though any mean and standard deviation will do:

> normaldata=rnorm(30,4,2)

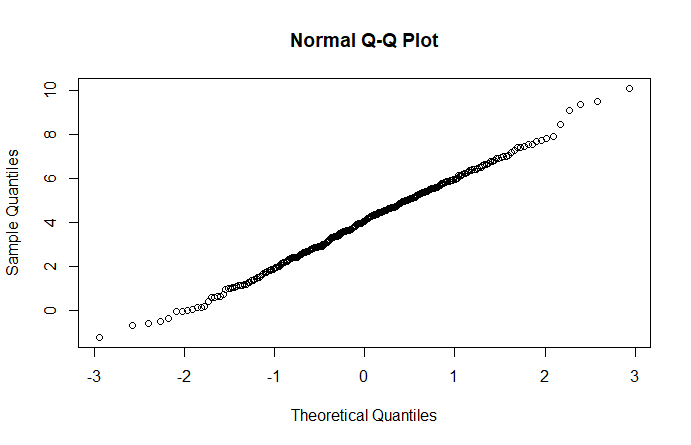
We then produce the QQ plot with the **qqnorm** function:

> qqnorm(normaldata, main="QQ plot of a normally-distributed sample")

The points roughly follow a line. If our sample were larger, we’d see a clearer pattern:

> normaldata=rnorm(300,4,2)

> qqnorm(normaldata)



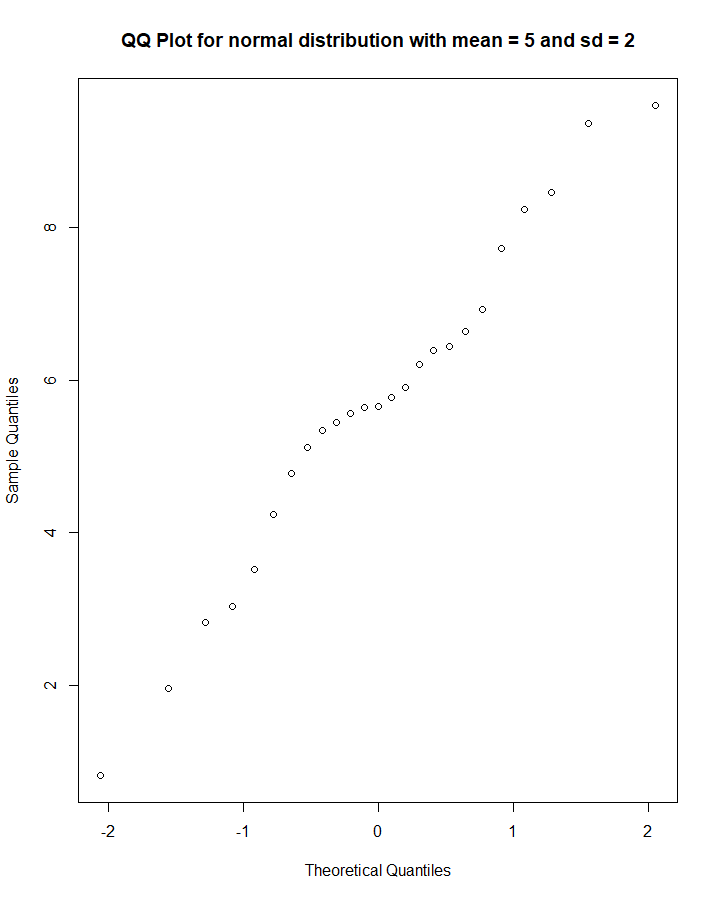
Note that the graph gets a bit weird at the edges. This is typical.

# QQ plots – a bit of theory

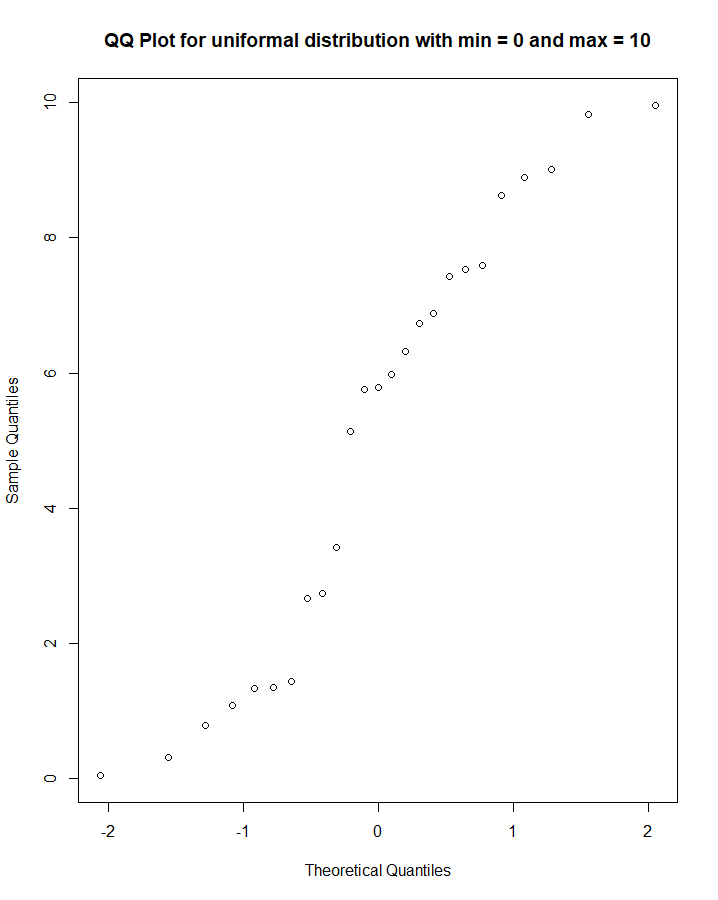
A QQ plot plots sample data against the z-scores that correspond to the percentiles of our data. If our data is normally distributed, then we would expect the same proportion of z-scores corresponding to the data as generated z-scores to be between *a* and *b*, for any numbers *a* and *b*. Graphically, this means that if our sample data came from a normally-distributed population, then the horizontal spacing between any two data points will be proportional to the vertical spacing, and our plot will be a straight line with positive slope. The less “normal” our sample data is, the more our plot will differ from being a straight line.

# Quantile-quantile plots of data from different distributions

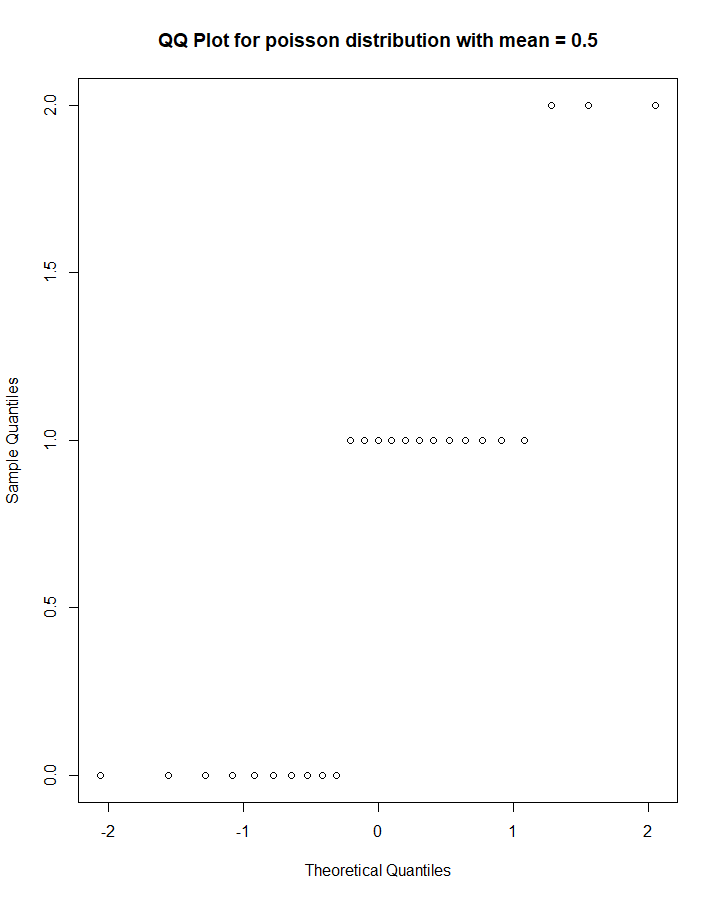
1. For each of the following distributions, generate a random sample of size 25 and create a QQ plot for that sample. Give your plots descriptive titles.
   1. Normal distribution with mean 5, standard deviation 2
   2. Uniform distribution with min=0, max=10
   3. Poisson distribution with mean 0.5
   4. Poisson distribution with mean 50
   5. Exponential distribution with mean rate 10
   6. Geometric distribution with p=0.1
   7. Geometric distribution with p=0.9
   8. Binomial distribution with n=10, p=0.1
   9. Binomial distribution with n=100, p=0.1
   10. Binomial distribution with n=1000, p=0.5
2. > qqnorm(rnorm(25,mean = 5, sd = 2), main="QQ Plot for normal distribution with mean = 5 and sd = 2")



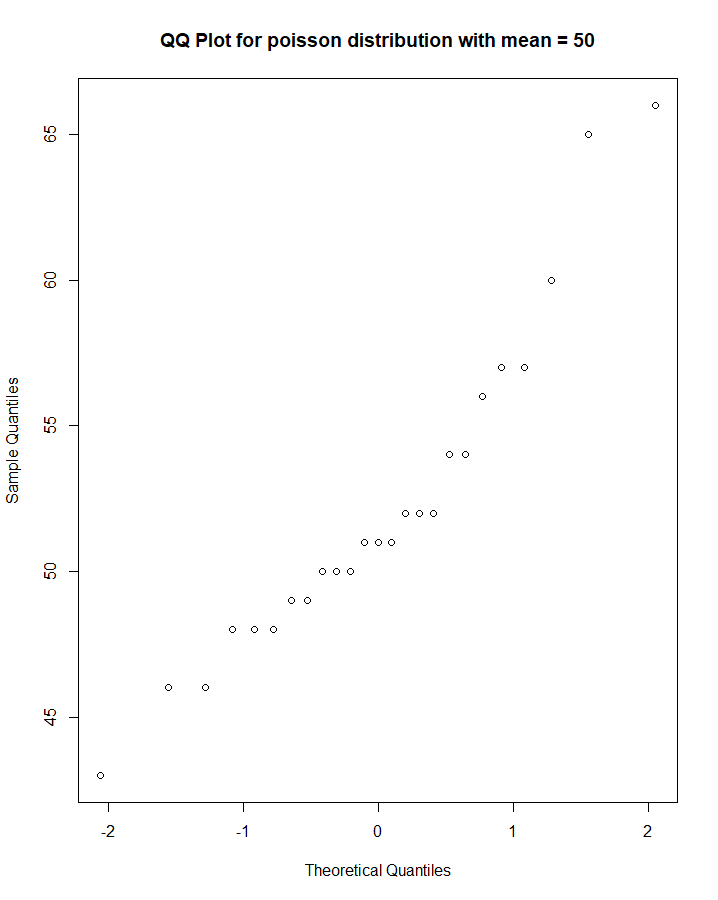
b. > qqnorm(runif(25,min=0, max=10), main="QQ Plot for uniformal distribution with min = 0 and max = 10")



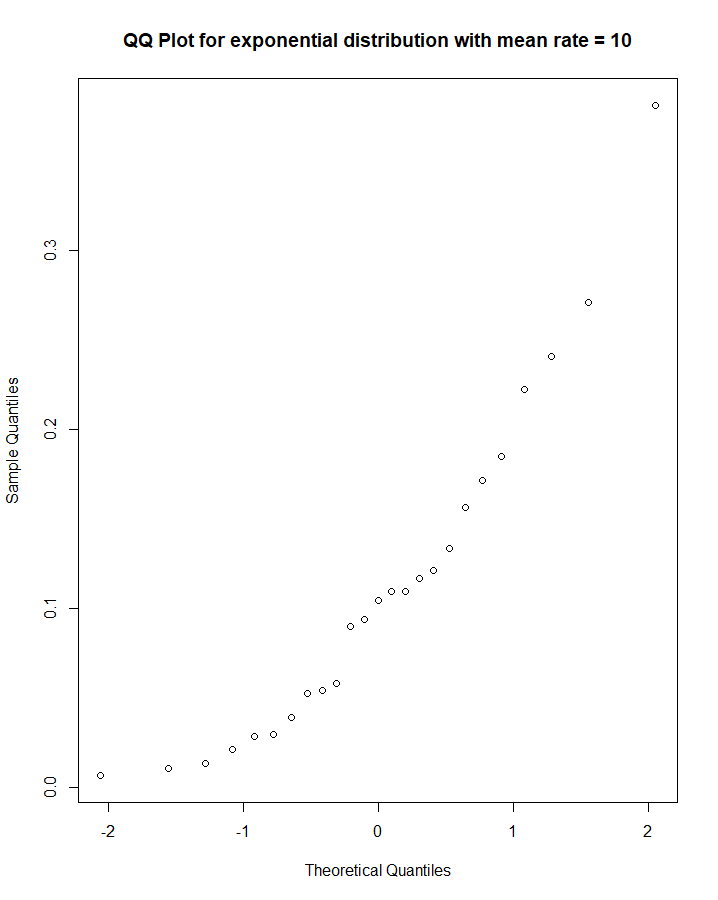
c. > qqnorm(rpois(25,0.5), main="QQ Plot for poisson distribution with mean = 0.5")



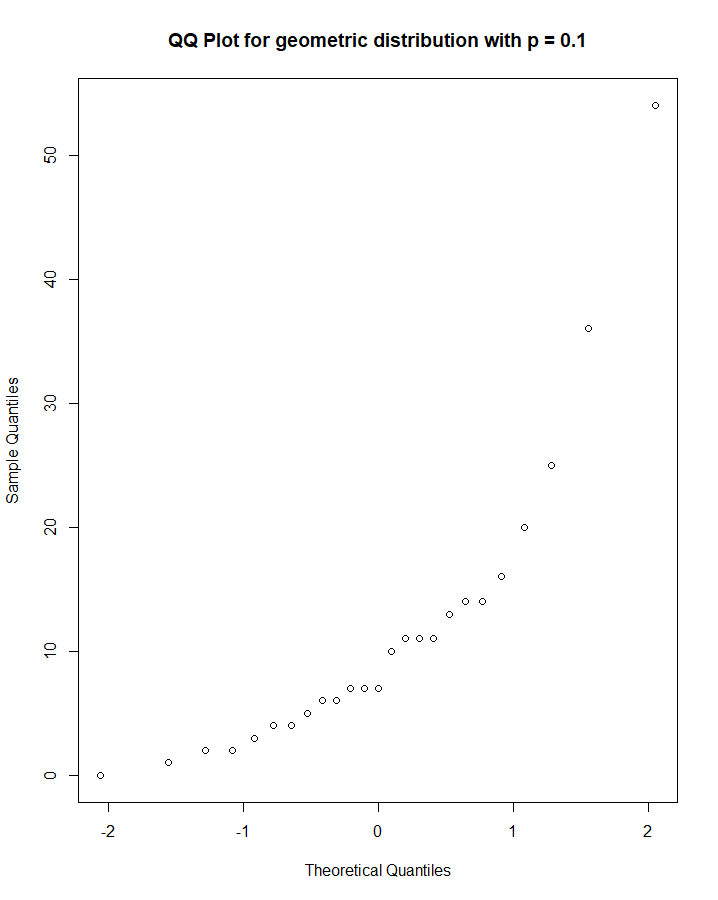
d. > qqnorm(rpois(25,50), main="QQ Plot for poisson distribution with mean = 50")



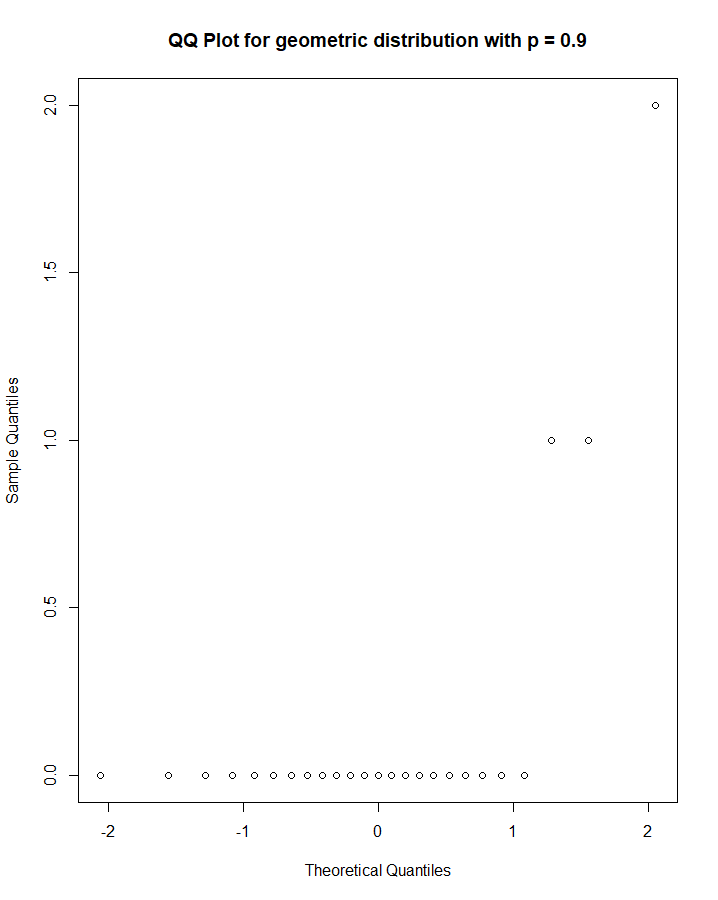
e. > qqnorm(rexp(25,10), main="QQ Plot for exponential distribution with mean rate = 10")



f. > qqnorm(rgeom(25,0.1), main="QQ Plot for geometric distribution with p = 0.1")



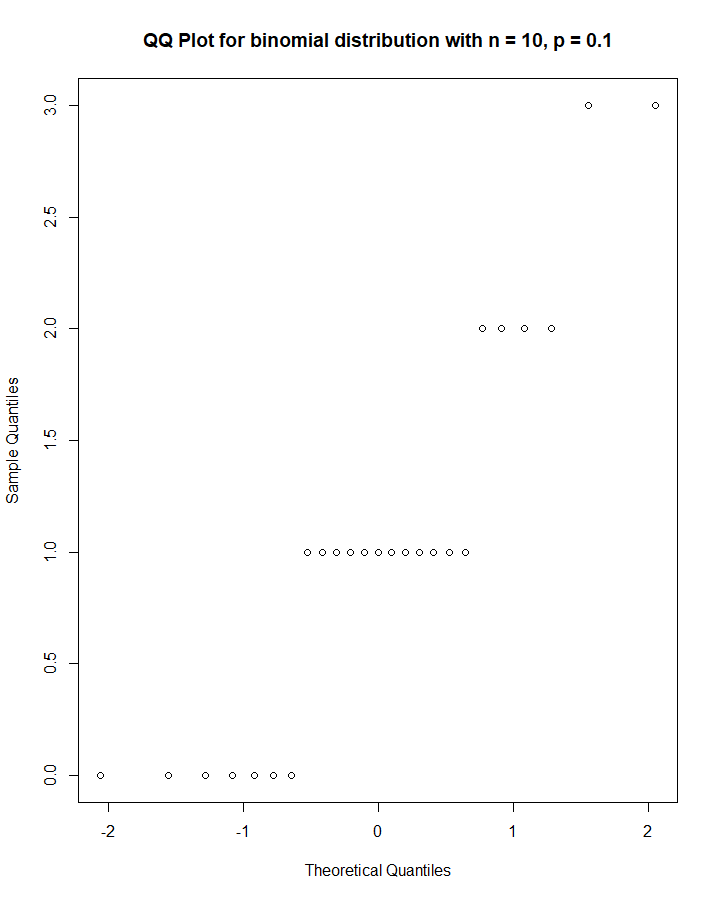
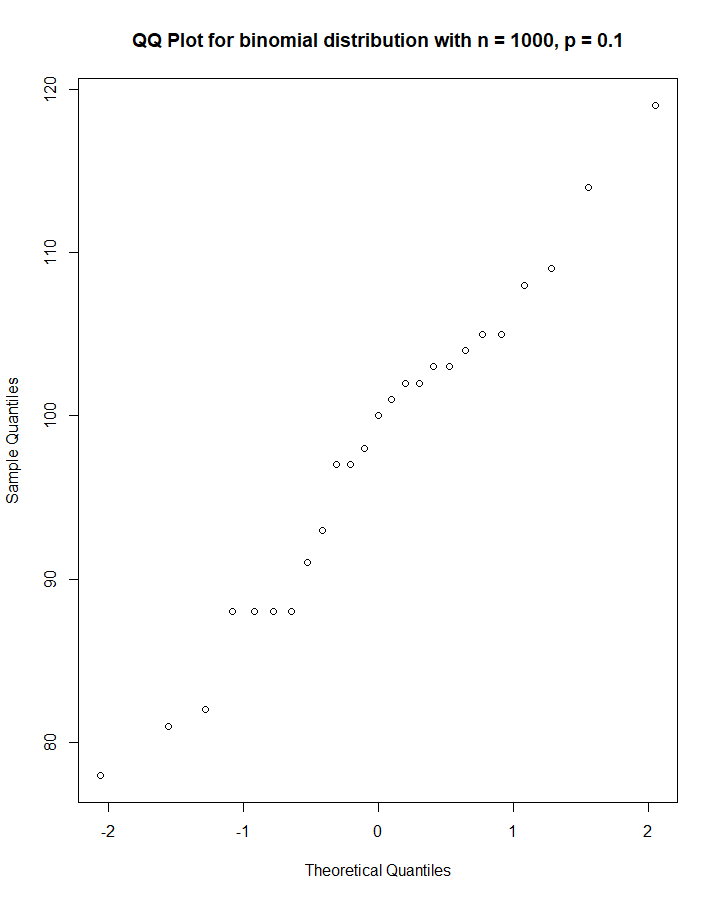
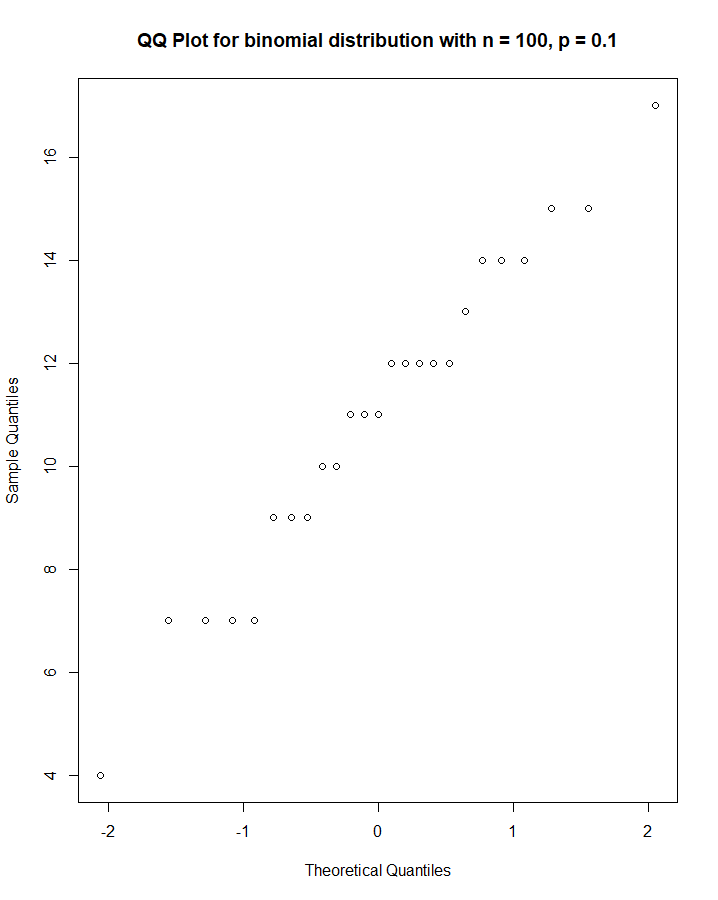
g. > qqnorm(rgeom(25,0.9), main="QQ Plot for geometric distribution with p = 0.9")



h. > qqnorm(rbinom(25, 10, 0.1), main="QQ Plot for binomial distribution with n = 10, p = 0.1")

i. > qqnorm(rbinom(25, 100, 0.1), main="QQ Plot for binomial distribution with n = 100, p = 0.1")

j. > qqnorm(rbinom(25, 1000, 0.1), main="QQ Plot for binomial distribution with n = 1000, p = 0.1")

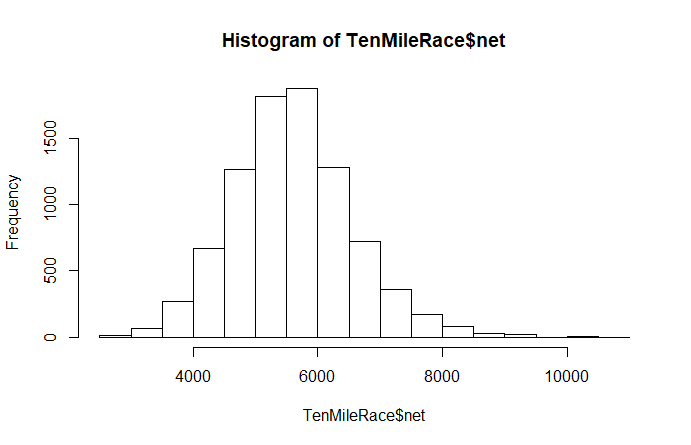
 

1. Which of the samples in Question 1 are the closest to normal? Which are the furthest? In class, we learned that some kinds of distributions (ie, binomial, Poisson) look bell shaped when certain conditions (which?) are met. Are your answers to Question 1 consistent with the theory we encountered in class? If not, were there any false positives (normal-looking samples that came from “abnormal” populations)? Any false negatives (abnormal-looking samples that came from normal populations)?

A,b,d,e,f,j are the closest ones to the normal. C,g,h and I are the furthest. In fact, they don’t even look close to it. The poisson distribution has bell shape when the sample size is large, but the probability of event is low (so, the lambda value is high). Our results are consistent with the theory.

# Confidence intervals

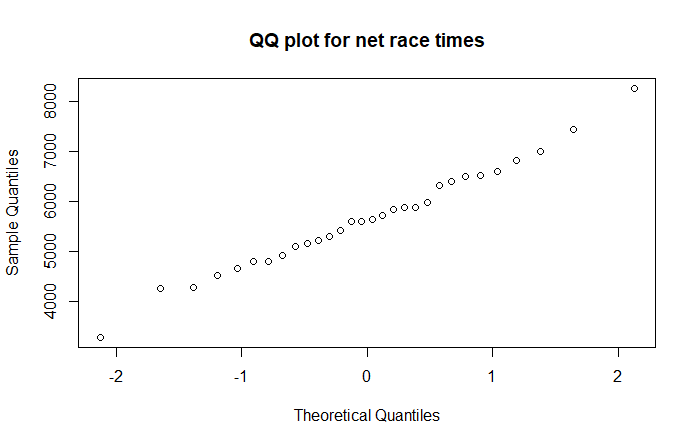
Let’s try this with some real data. Recall the dataset **TenMileRace** from Lab 1. Load this dataset (you will have to load the **mosaicData** library). Here is a histogram of net race times, which certainly appear to follow a normal distribution:



In real life, we often don’t have access to entire populations, or even large samples. When our samples are small, we need to check if they are normal before finding t-values to generate confidence intervals. For small samples, a QQ plot gives us a clearer picture of the distribution than a histogram. Since the population of net times is normally distributed, a sample of times will probably be close to normally distributed as well. Eg, here are 30 randomly-selected race times (the **sample** function gives us 30 indices, and then we retrieve the corresponding race times):

> racetimes30=TenMileRace$net[sample(1:8636, 30)]

> qqnorm(racetimes30, main="QQ plot for net race times")



It certainly appears that the sample of race times, like the population, is normally distributed.

(NOTE: your QQ plot will be different, and it might not look like it reflects a normal distribution. Unfortunately, this sort of thing happens occasionally: samples from normally distributed populations aren’t always normally distributed. Conversely, on occasion we have normally distributed samples that come from populations that are not normally distributed. You may have seen examples of both in Question 1. This sort of thing is why, in statistics, we never “prove” anything with complete certainty – we are only 90%, or 95%, or 99% confident of our conclusion. The other 10% or 5% or 1%, we just happened to get a sample that was not representative of the population. For instance, if the population is large and normally distributed, it’s possible (but not very likely) that through no fault of our own, it turned out that every number in our sample is more than two standard deviations from the mean. That said, if your QQ plot looks very different from a straight line, generate another sample of size 30 and get a QQ plot for it. It will probably look better than your last one.)

We saw in class that we can construct confidence intervals for means of small (n less than or equal to 30) data sets as long as the sample is normally distributed. The **t.test** function in R allows us to construct confidence intervals in a single line:

> t.test(racetimes30)

One Sample t-test

data: racetimes30

t = 35.126, df = 29, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

5194.265 5836.535

sample estimates:

mean of x

5515.4

The highlighted line is the one we care about: it tells us that we are 95% sure that the mean net race time is between 5194.265 seconds and 5836.535 seconds.

We can find the mean net race time for the entire population:

> mean(TenMileRace$net)

[1] 5599.065

This value is indeed in the confidence interval.

1. Our theory tells us that 95% of the time, or 19 times out of 20, the true population mean will be contained in the 95% confidence interval. (The other 5% of the time, our sample happened to be “weird” – ie, with a distribution much different from that of the population.) Generate 20 different random samples of size 25 from the population of net race times and construct confidence intervals. How many of them contained the true mean?

confidence25 = function() {

racetimes25=TenMileRace$net[sample(1:8636, 25)]

t.test(racetimes25)

}

The true mean is > mean(TenMileRace$net)

[1] 5599.065

All 20 of them contained the true mean in the 95% confidence interval, but some of them were very close to fall off.

1. > confidence25()
2. One Sample t-test
3. data: racetimes25
4. t = 23.653, df = 24, p-value < 2.2e-16
5. alternative hypothesis: true mean is not equal to 0
6. 95 percent confidence interval:
7. 4870.03 5801.17
8. sample estimates:
9. mean of x
10. 5335.6
11. > confidence25()
12. One Sample t-test
13. data: racetimes25
14. t = 29.254, df = 24, p-value < 2.2e-16
15. alternative hypothesis: true mean is not equal to 0
16. 95 percent confidence interval:
17. 4962.335 5715.665
18. sample estimates:
19. mean of x
20. 5339
21. > confidence25()
22. One Sample t-test
23. data: racetimes25
24. t = 30.58, df = 24, p-value < 2.2e-16
25. alternative hypothesis: true mean is not equal to 0
26. 95 percent confidence interval:
27. 5271.326 6034.354
28. sample estimates:
29. mean of x
30. 5652.84
31. > confidence25()
32. One Sample t-test
33. data: racetimes25
34. t = 28.981, df = 24, p-value < 2.2e-16
35. alternative hypothesis: true mean is not equal to 0
36. 95 percent confidence interval:
37. 5307.817 6121.783
38. sample estimates:
39. mean of x
40. 5714.8
41. > confidence25()
42. One Sample t-test
43. data: racetimes25
44. t = 30.142, df = 24, p-value < 2.2e-16
45. alternative hypothesis: true mean is not equal to 0
46. 95 percent confidence interval:
47. 5269.797 6044.523
48. sample estimates:
49. mean of x
50. 5657.16
51. > confidence25()
52. One Sample t-test
53. data: racetimes25
54. t = 24.744, df = 24, p-value < 2.2e-16
55. alternative hypothesis: true mean is not equal to 0
56. 95 percent confidence interval:
57. 5153.54 6091.50
58. sample estimates:
59. mean of x
60. 5622.52
61. > confidence25()
62. One Sample t-test
63. data: racetimes25
64. t = 30.498, df = 24, p-value < 2.2e-16
65. alternative hypothesis: true mean is not equal to 0
66. 95 percent confidence interval:
67. 5267.194 6031.846
68. sample estimates:
69. mean of x
70. 5649.52
71. > confidence25()
72. One Sample t-test
73. data: racetimes25
74. t = 34.723, df = 24, p-value < 2.2e-16
75. alternative hypothesis: true mean is not equal to 0
76. 95 percent confidence interval:
77. 5319.14 5991.42
78. sample estimates:
79. mean of x
80. 5655.28
81. > confidence25()
82. One Sample t-test
83. data: racetimes25
84. t = 31.803, df = 24, p-value < 2.2e-16
85. alternative hypothesis: true mean is not equal to 0
86. 95 percent confidence interval:
87. 5242.485 5970.155
88. sample estimates:
89. mean of x
90. 5606.32
91. > confidence25()
92. One Sample t-test
93. data: racetimes25
94. t = 37.671, df = 24, p-value < 2.2e-16
95. alternative hypothesis: true mean is not equal to 0
96. 95 percent confidence interval:
97. 5486.204 6122.196
98. sample estimates:
99. mean of x
100. 5804.2
101. > confidence25()
102. One Sample t-test
103. data: racetimes25
104. t = 26.389, df = 24, p-value < 2.2e-16
105. alternative hypothesis: true mean is not equal to 0
106. 95 percent confidence interval:
107. 5092.294 5956.426
108. sample estimates:
109. mean of x
110. 5524.36
111. > confidence25()
112. One Sample t-test
113. data: racetimes25
114. t = 24.773, df = 24, p-value < 2.2e-16
115. alternative hypothesis: true mean is not equal to 0
116. 95 percent confidence interval:
117. 5327.172 6295.468
118. sample estimates:
119. mean of x
120. 5811.32
121. > confidence25()
122. One Sample t-test
123. data: racetimes25
124. t = 24.604, df = 24, p-value < 2.2e-16
125. alternative hypothesis: true mean is not equal to 0
126. 95 percent confidence interval:
127. 5478.196 6481.404
128. sample estimates:
129. mean of x
130. 5979.8
131. > confidence25()
132. One Sample t-test
133. data: racetimes25
134. t = 26.271, df = 24, p-value < 2.2e-16
135. alternative hypothesis: true mean is not equal to 0
136. 95 percent confidence interval:
137. 5077.64 5943.48
138. sample estimates:
139. mean of x
140. 5510.56
141. > confidence25()
142. One Sample t-test
143. data: racetimes25
144. t = 33.434, df = 24, p-value < 2.2e-16
145. alternative hypothesis: true mean is not equal to 0
146. 95 percent confidence interval:
147. 5472.853 6192.987
148. sample estimates:
149. mean of x
150. 5832.92
151. > confidence25()
152. One Sample t-test
153. data: racetimes25
154. t = 24.338, df = 24, p-value < 2.2e-16
155. alternative hypothesis: true mean is not equal to 0
156. 95 percent confidence interval:
157. 5235.673 6205.927
158. sample estimates:
159. mean of x
160. 5720.8
161. > confidence25()
162. One Sample t-test
163. data: racetimes25
164. t = 30.435, df = 24, p-value < 2.2e-16
165. alternative hypothesis: true mean is not equal to 0
166. 95 percent confidence interval:
167. 5302.021 6073.419
168. sample estimates:
169. mean of x
170. 5687.72
171. > confidence25()
172. One Sample t-test
173. data: racetimes25
174. t = 39.09, df = 24, p-value < 2.2e-16
175. alternative hypothesis: true mean is not equal to 0
176. 95 percent confidence interval:
177. 5115.687 5685.993
178. sample estimates:
179. mean of x
180. 5400.84
181. > confidence25()
182. One Sample t-test
183. data: racetimes25
184. t = 32.795, df = 24, p-value < 2.2e-16
185. alternative hypothesis: true mean is not equal to 0
186. 95 percent confidence interval:
187. 5575.134 6323.986
188. sample estimates:
189. mean of x
190. 5949.56
191. > confidence25()
192. One Sample t-test
193. data: racetimes25
194. t = 22.634, df = 24, p-value < 2.2e-16
195. alternative hypothesis: true mean is not equal to 0
196. 95 percent confidence interval:
197. 4842.427 5814.133
198. sample estimates:
199. mean of x
200. 5328.28

Note again that this theory only works for small samples if they come from a normally-distributed population. If we compute a 95% confidence interval the same way for a small sample that is very abnormal, the range we get will not be a true 95% confidence interval. That is, it will not be the case that 95% of confidence intervals obtained with our method will contain the true population mean.

Load the **genotype** dataframe, which is part of the MASS library. Take a minute to read over its help file.

1. Filter your data to obtain four data sets, grouped by litter genotype. Note that these data sets are small, so we require them to be normally distributed if we wish to find a confidence interval using the methods we learned in class. Create QQ plots for weight for each genotype. Which of the four appears to be the most normal? Create a 95% confidence interval for the mean litter weight gain for rats of that genotype, and include a sentence interpreting your result.

Now load the **cats** dataframe.

> litterA = filter(genotype, genotype$Litter == 'A')

> litterB = filter(genotype, genotype$Litter == 'B')

> litterI = filter(genotype, genotype$Litter == 'I')

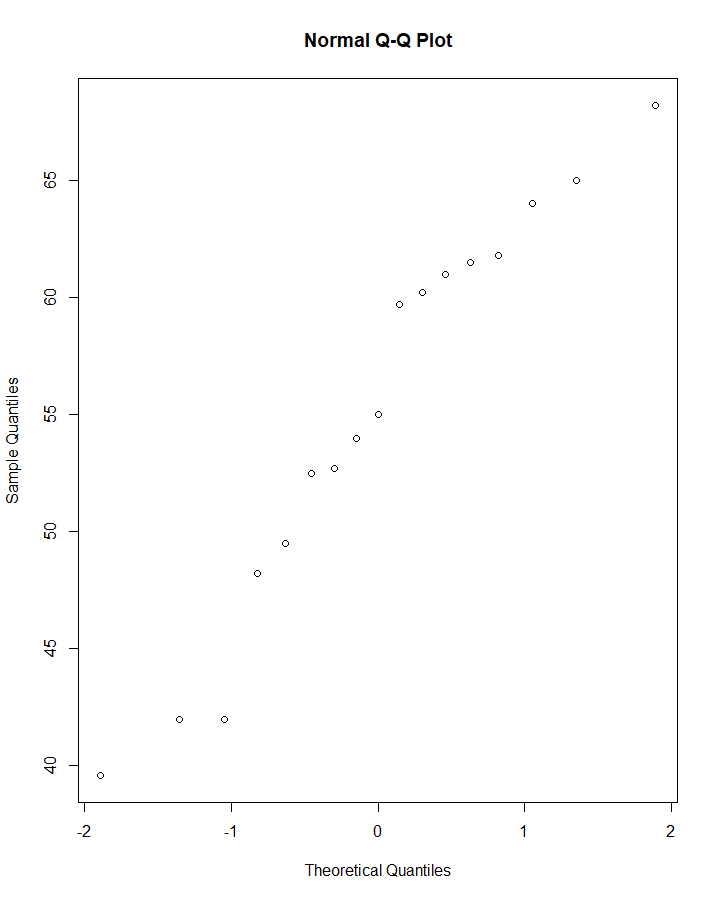
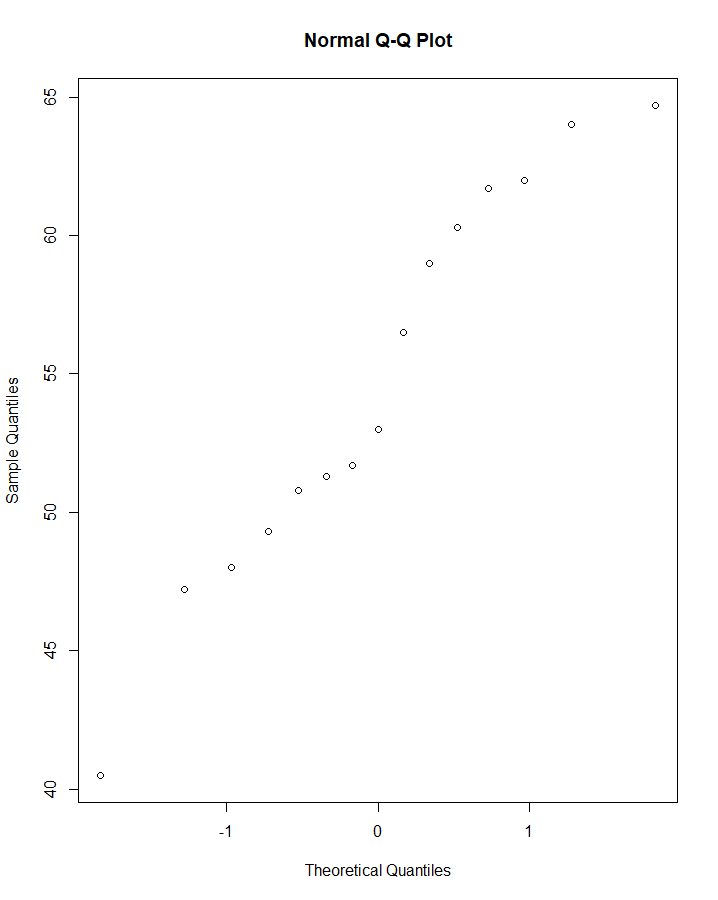
> litterJ = filter(genotype, genotype$Litter == 'J')

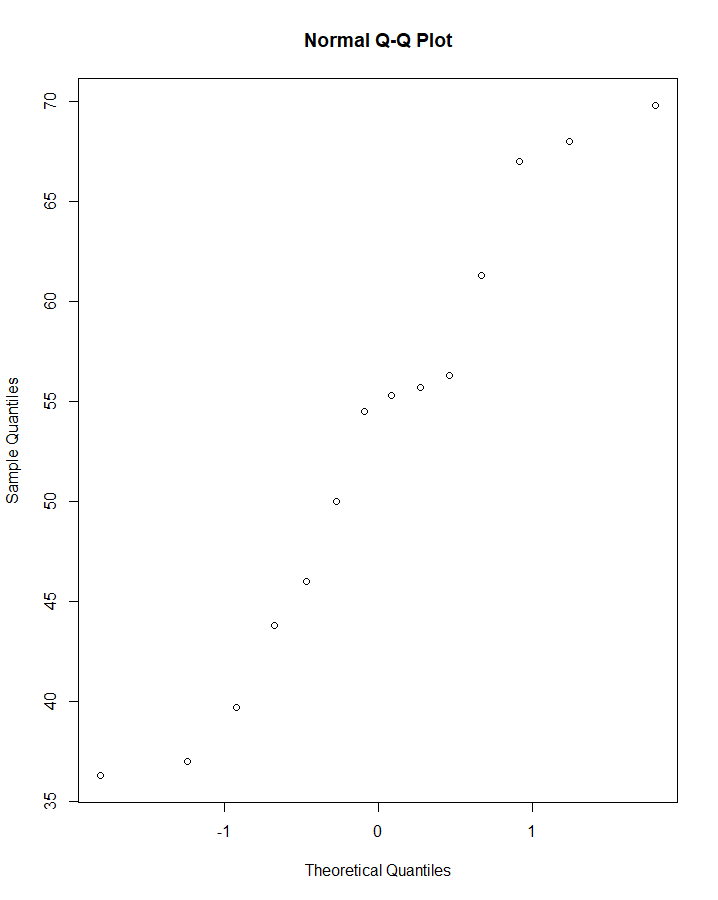
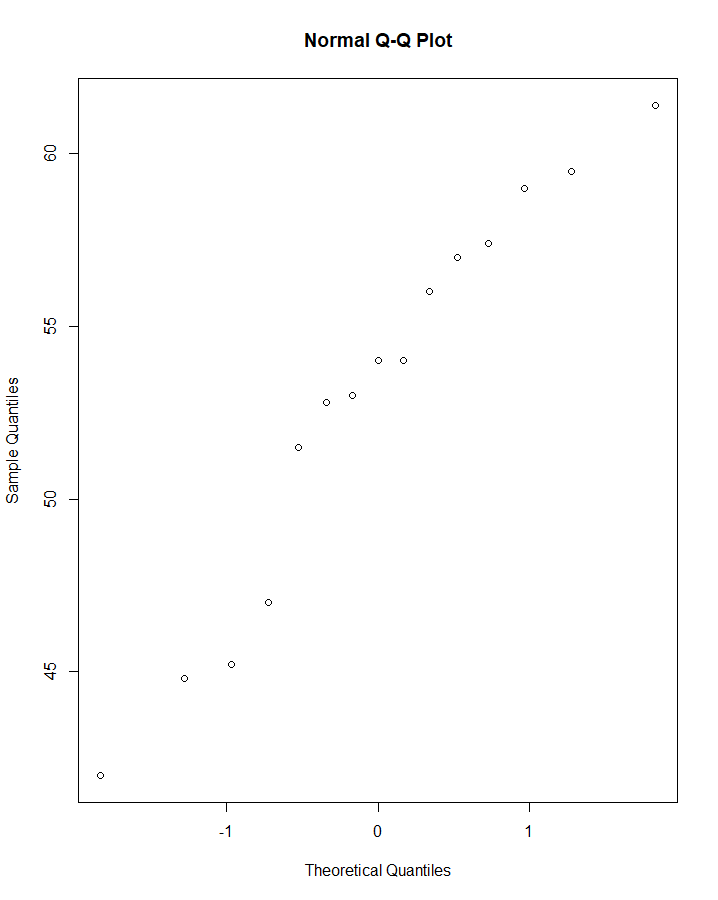
> qqnorm(litterA$Wt)

> qqnorm(litterB$Wt)

> qqnorm(litterI$Wt)

> qqnorm(litterJ$Wt)

> t.test(litterA$Wt)

One Sample t-test

data: litterA$Wt

t = 26.317, df = 16, p-value = 1.341e-14

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

50.67238 59.55115

sample estimates:

mean of x

55.11176

> t.test(litterB$Wt)

One Sample t-test

data: litterB$Wt

t = 29.679, df = 14, p-value = 4.839e-14

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

50.71616 58.61717

sample estimates:

mean of x

54.66667

> t.test(litterI$Wt)

One Sample t-test

data: litterI$Wt

t = 17.56, df = 13, p-value = 1.948e-10

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

46.39810 59.41618

sample estimates:

mean of x

52.90714

> t.test(litterJ$Wt)

One Sample t-test

data: litterJ$Wt

t = 34.947, df = 14, p-value = 5.057e-15

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

49.72219 56.22448

sample estimates:

mean of x

52.97333

> mean(genotype$Wt)

[1] 53.97049

All the confidence intervals contain the true mean of the population despite the fact of the very small sample size and population.

1. Find 95% confidence intervals for body weight and heart weight for male and female cats. (ie, you will be finding four confidence intervals.) Note that these samples are large enough that we don’t have to worry about whether they are normally distributed. Include sentences interpreting your results. Based on your intervals, does there appear to be a significant difference between the heart weights of male cats versus female cats? How about body weights?

> catM = filter(cats, cats$Sex == 'M')

> catF = filter(cats, cats$Sex == 'F')

> t.test(catM$Bwt)

One Sample t-test

data: catM$Bwt

t = 61.097, df = 96, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.805781 2.994219

sample estimates:

mean of x

2.9

> t.test(catM$Hwt)

One Sample t-test

data: catM$Hwt

t = 43.864, df = 96, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

10.81030 11.83506

sample estimates:

mean of x

11.32268

> t.test(catF$Bwt)

One Sample t-test

data: catF$Bwt

t = 59.041, df = 46, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.279129 2.440020

sample estimates:

mean of x

2.359574

> t.test(catF$Hwt)

One Sample t-test

data: catF$Hwt

t = 46.467, df = 46, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

8.803502 9.600753

sample estimates:

mean of x

9.202128

Based on my observation there is a significant weight difference in both heart weight and body weight between Male and Female cats.

The next question will have you working from the functions you created in Assignment 4. Get your script from that assignment and save it as Lab8.R.

1. Modify your simulations to obtain 95% confidence intervals for the mean time required to repair **m** computers for each of the three computer repair companies. (You should only need to change one line in each of three functions.) Run your three functions for:
   1. m=20, n=30 (ie, 30 simulations of 20 computer repairs)
   2. m=20, n=230
   3. m=20, n=430

> SimulateConstantRepairs(30,20)

One Sample t-test

data: arr

t = 17.708, df = 29, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.307081 2.909586

sample estimates:

mean of x

2.608333

> SimulateConstantRepairs(230,20)

One Sample t-test

data: arr

t = 61.842, df = 229, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.528314 2.694729

sample estimates:

mean of x

2.611522

> SimulateConstantRepairs(430,20)

One Sample t-test

data: arr

t = 69.905, df = 429, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.567919 2.716500

sample estimates:

mean of x

2.642209

> SimulateExponentialRepairs(30, 20)

One Sample t-test

data: arr

t = 13.452, df = 29, p-value = 5.381e-14

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.755146 3.743132

sample estimates:

mean of x

3.249139

> SimulateExponentialRepairs(230, 20)

One Sample t-test

data: arr

t = 36.994, df = 229, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.947308 3.278934

sample estimates:

mean of x

3.113121

> SimulateExponentialRepairs(430, 20)

One Sample t-test

data: arr

t = 54.916, df = 429, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.870582 3.083694

sample estimates:

mean of x

2.977138

> SimulateNormalRepairs(30, 20)

One Sample t-test

data: arr

t = 21.087, df = 29, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.482548 3.015846

sample estimates:

mean of x

2.749197

> SimulateNormalRepairs(230, 20)

One Sample t-test

data: arr

t = 46.885, df = 229, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.622593 2.852695

sample estimates:

mean of x

2.737644

> SimulateNormalRepairs(430, 20)

One Sample t-test

data: arr

t = 80.607, df = 429, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

2.574691 2.703392

sample estimates:

mean of x

2.639041

Our results are consistent with the theory. It seems that the obtained results flatten out if taken large sample, that’s why there is no big difference between sampling 230 or 430, but there is a huge difference between 30 and 230 (or 430).

Include the commands you used, as well as sentences interpreting your conclusions.

Notice that the number of simulations, or equivalently the size of the samples, differed by 200 from part a) to b) and again from part b) to c). However, although the confidence intervals shrunk with each sample increase, they didn’t shrink by nearly as much the second time around. In other words: although larger samples give tighter estimates, there’s generally a point at which the improvement obtained by increasing the sample size isn’t worth the extra time or cost. This is why, in practice, we seldom see samples larger than a few hundred.