

PH525.1x: Week1 & Week2

Shu Guo

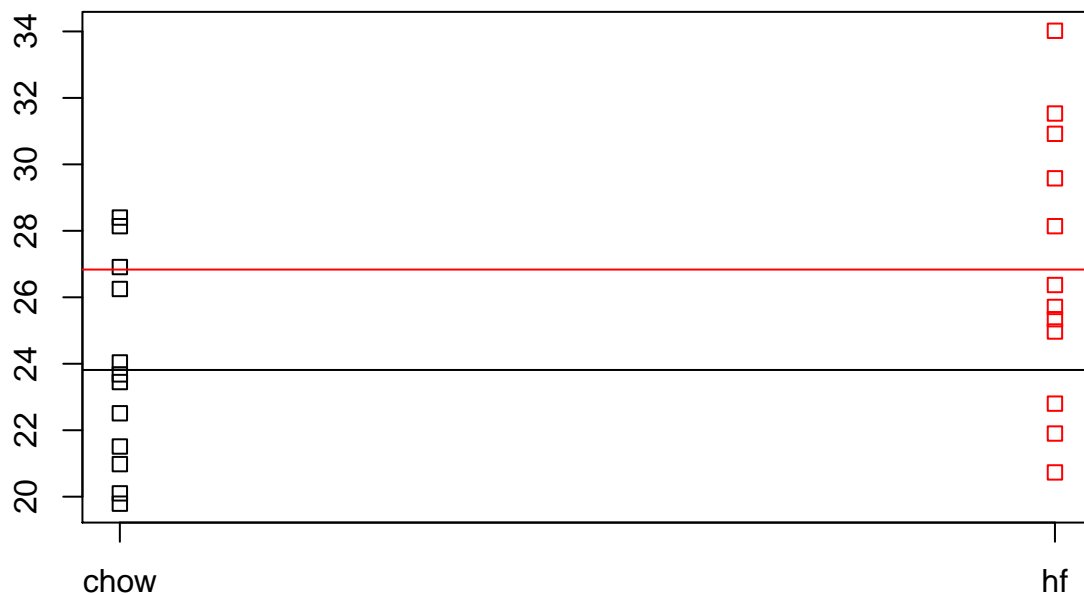
Sunday, February 01, 2015

Introduction to Random Variabbe I

```
## Load the data
dat = read.csv("femaleMiceWeights.csv")
## The observed difference between high fat diet and control was
mean(dat[13:24,2]) - mean(dat[1:12,2])
```

```
## [1] 3.020833
```

```
## A strip chart of the weights of these two groups
s = split(dat[,2], dat[,1])
stripchart(s, vertical=TRUE, col=1:2)
## Add the means to the plot
abline(h=sapply(s, mean), col=1:2)
```



```
## Question 1.1 How many of the high fat mice weigh less than the mean of the control mice (chow)?
sum(s$hf < mean(s$chow))
```

```
## [1] 3
```

```
## Question 1.2 How many of the control mice weigh more than the mean of the high fat mice?
sum(s$chow > mean(s$hf))
```

```
## [1] 3
```

```
## Question 1.3 What is the proportion of high fat diet mice over 30?
sum(s$hf > 30)/length(s$hf)
```

```
## [1] 0.25
```

Introduction to Random Variables II

```
## Course example
dat[1:12, 2]
```

```
## [1] 21.51 28.14 24.04 23.45 23.68 19.79 28.40 20.98 22.51 20.10 26.91
## [12] 26.25
```

```
mean(dat[13:24, 2] - mean(dat[1:12, 2]))
```

```
## [1] 3.020833
```

```
population <- read.csv("femaleControlsPopulation.csv")
```

```
n <- 10000
null <- vector("numeric", n)
for (i in 1:n){
  control <- sample(population[, 1], 12)
  treatment <- sample(population[, 1], 12)
  null[i] <- mean(treatment) - mean(control)
}
```

```
diff <- mean(dat[13:24, 2]) - mean(dat[1:12, 2])
#what percent are bigger than `diff`?
mean(null > diff)
```

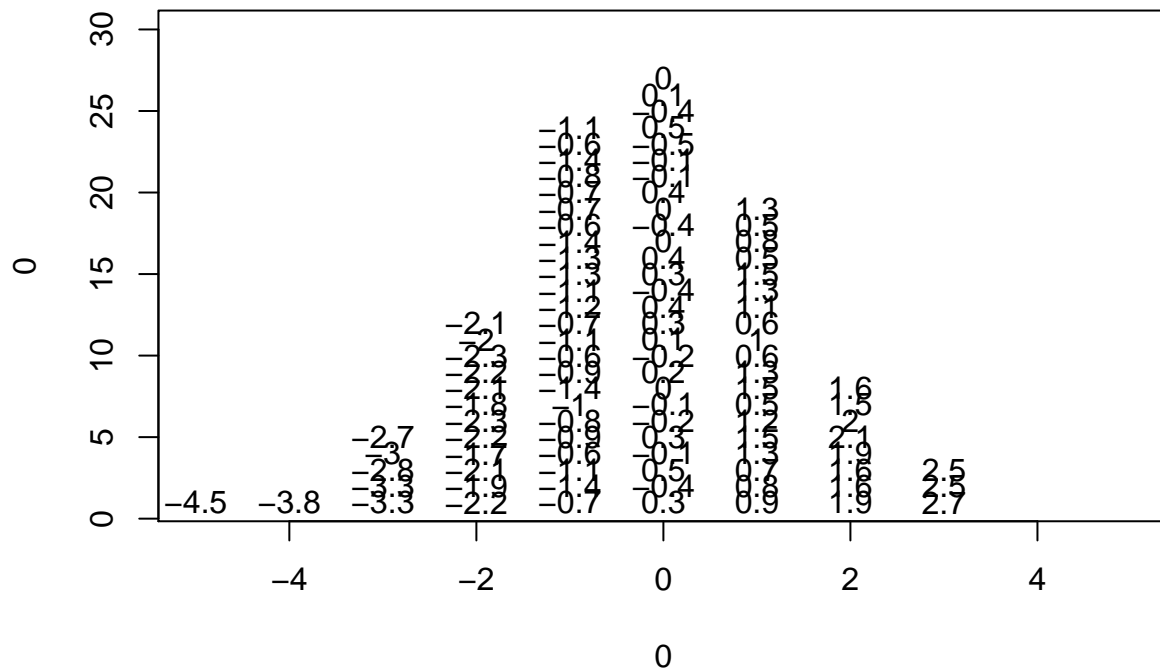
```
## [1] 0.0124
```

Illustration of the null distribution Repeat the loop above but this time add a point to the figure every time re-run the experiment

```

# Read the population
n <- 100
plot(0, 0, xlim = c(-5, 5), ylim = c(1, 30), type = "n")
totals <- vector("numeric", 11)
for (i in 1:n){
  control <- sample(population[, 1], 12)
  treatment <- sample(population[, 1], 12)
  nulldiff <- mean(treatment) - mean(control)
  j <- pmax(pmin(round(nulldiff) + 6, 11), 1)
  totals[j] <- totals[j] + 1
  text(j - 6, totals[j], pch = 15, round(nulldiff, 1))
  ##if(i < 15) scan() ## add this line to interactively see values appear
}

```



Recreate the vector of differences between means of random samples from the control population.

```

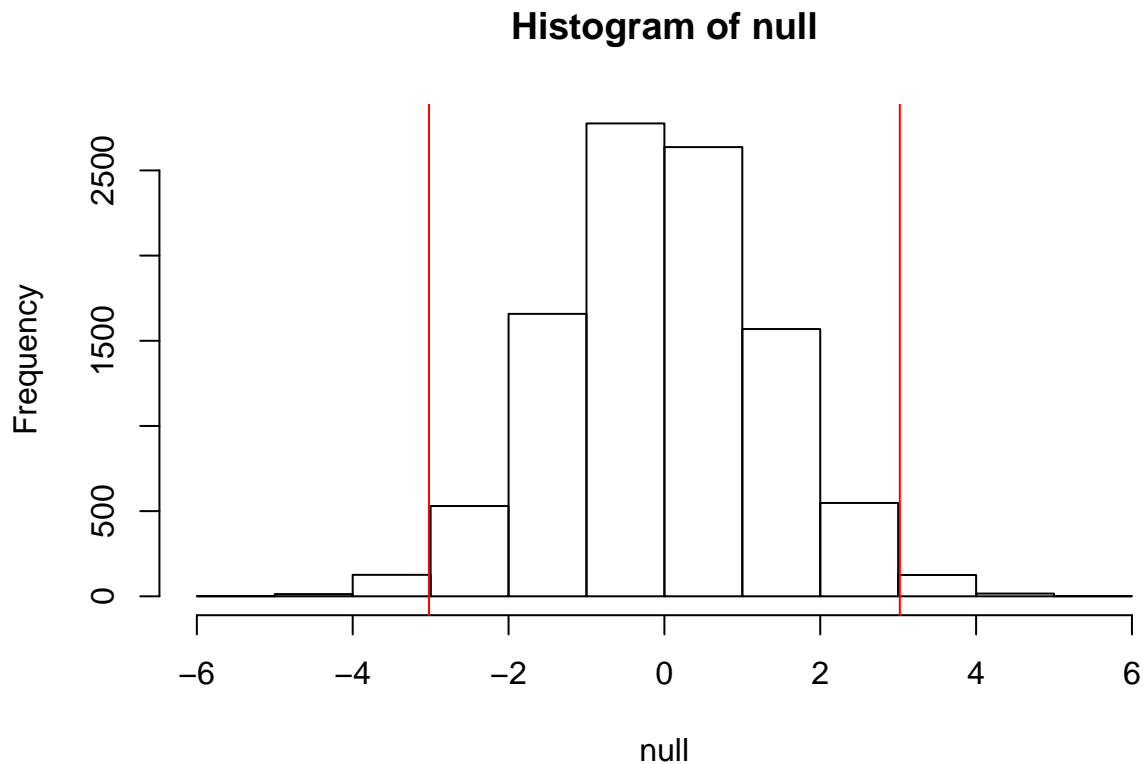
null <- replicate(10000, mean(sample(population[, 1], 12)) -
                           mean(sample(population[, 1], 12)))

# Make a histogram
hist(null)

# The original difference observed between the mice fed high fat diets and control mice:
diff = mean(dat[13:24,2]) - mean(dat[1:12,2])
# Add this difference to the histogram:
abline(v=diff, col="red")
# Also add the negative of the difference:

```

```
abline(v=-diff, col="red")
```



If we look for the number of null distribution values to the right of the (right) red line, we would say “we calculated the probability of observing a larger difference from the null distribution”. This is sometimes called a “one-tailed” probability, because we only look at one “tail” of the histogram (the left and right sides where the bars become short).

By looking at the tails on both sides of the histogram, we can say “we calculated the probability of observing as extreme a difference from the null distribution”. This is sometimes called a “two-tailed” probability. This probability is commonly referred to as a p-value.

Question 3.1: What is the one-tailed probability of seeing as big a difference as we observed, calculated from your null distribution? (0.0137)

Question 3.2: What is the two-tailed probability of seeing as big a difference as we observed, calculated from your null distribution? (0.0274)

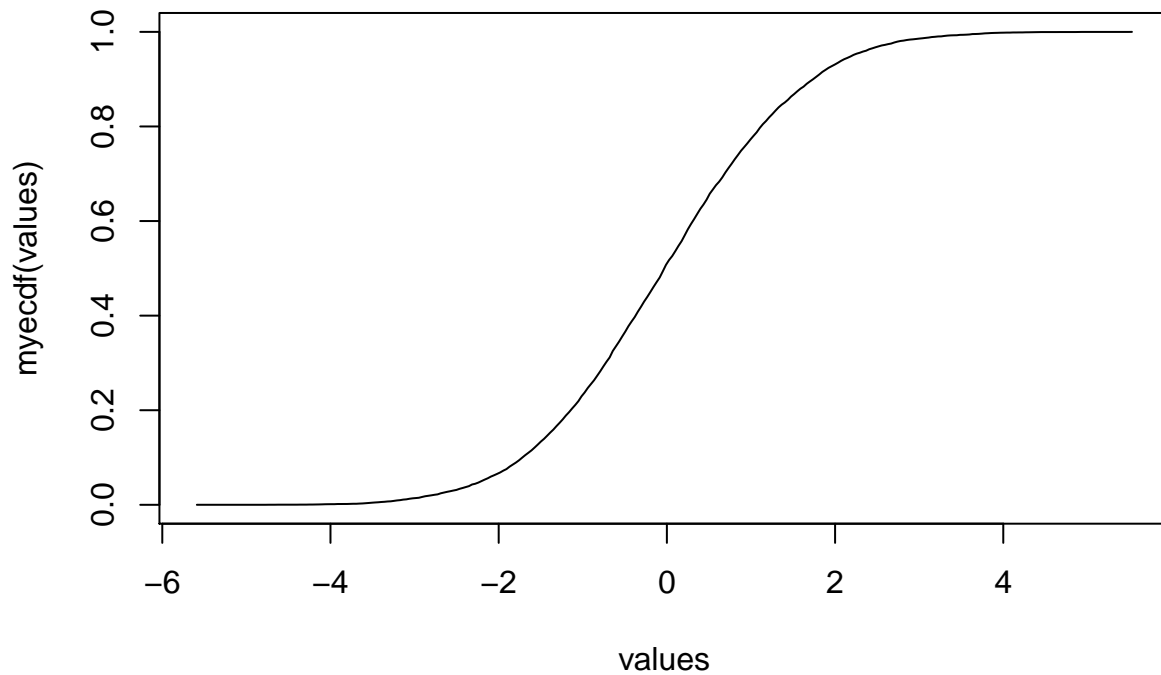
Distributions

A *distribution* is as a compact description of many numbers. For example, in the previous section we defined an object ‘null’ with 10,000 average differences created under the null. To define a distribution we compute, for all possible values of a the proportion of numbers in our list that are below a . We use the following notation

$$F(a) \equiv \Pr(x \leq a)$$

This is called the empirical cumulative distribution function. We can plot $F(a)$ versus a like this

```
values <- seq(min(null), max(null), len = 300)
myecdf <- ecdf(null)
plot(values, myecdf(values), type = "l")
```



Histograms give us the same information but show us the proportion of values in intervals:

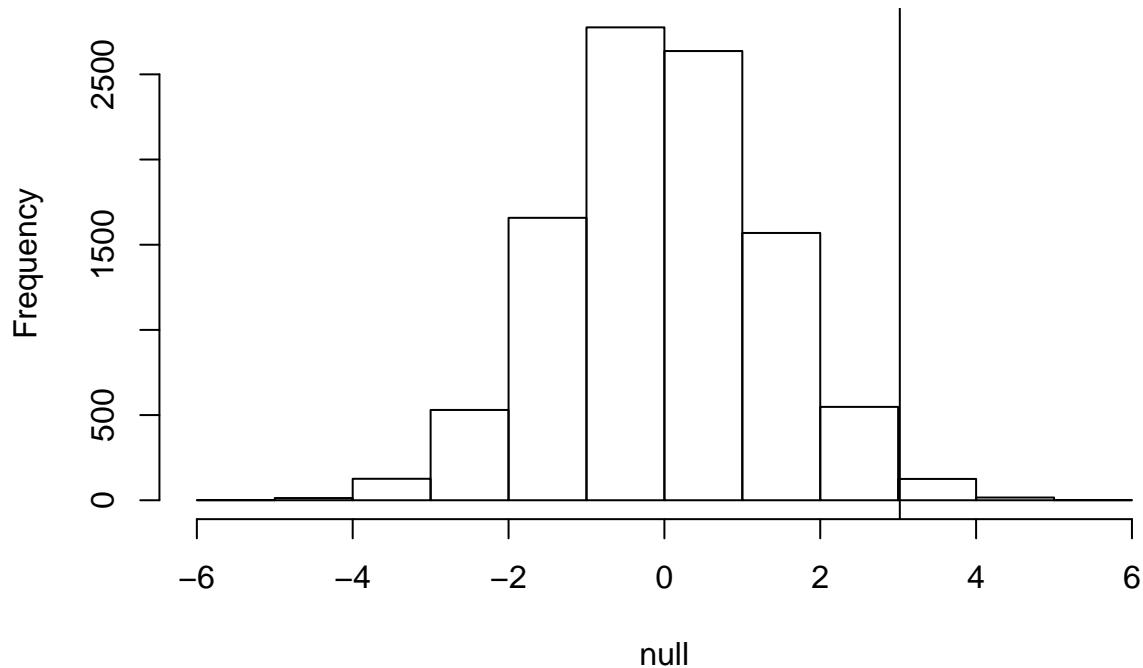
$$\Pr(a \leq x \leq b) = F(b) - F(a)$$

This is a more useful plot because we are usually more interested in intervals. It is also easier to distinguish different types (families) of distributions by looking at histograms.

Note that from the histogram we can see that values as large as **diff** are relatively rare

```
hist(null)
abline(v = diff)
```

Histogram of null



Normal distribution

When the histogram of a list of numbers approximates the normal distribution we can use a convenient mathematical formula to approximate the proportion of individuals in any given interval

$$\Pr(a < x < b) = \int_a^b \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx$$

Here μ and σ are referred to as the mean and standard deviation. If this approximation holds for our list then the population mean and variance of our list can be used in the formula above. To see this with an example remember that above we noted that only 1.5% of values on the null distribution were above `diff`. We can compute the proportion of values below a value `x` with `pnorm(x,mu,sigma)` without knowing all the values. The normal approximation works very well here:

```
1-pnorm(diff,mean(null),sd(null))
```

```
## [1] 0.01230661
```

A very useful characteristic of this approximation is that one only needs to know μ and σ to describe the entire distribution. From this we can compute the proportion of values in any interval.

Summary

Note that to make this calculation we did the equivalent of buying all the mice available from Jackson laboratories and performed our experiment over and over again to define the null distribution. This is not something we can do in practice. Statistical Inference is the mathematical theory that permits you to approximate this with only the data from your sample, i.e. the original 24 mice.

Questions

Question 1.2: We will use the data set called “Gapminder” which is available as an R-package on Github. This data set contains the life expectancy, GDP per capita, and population by country, every five years, from 1952 to 2007. It is an excerpt of a larger and more comprehensive set of data available on Gapminder.org, and the R package of this dataset was created by the statistics professor Jennifer Bryan.

```
# First, install the gapminder data set using the devtools R-package.
library(devtools)
install_github("jennybc/gapminder")
```

Next, load the gapminder data set.

```
library(gapminder)
data(gapminder)
head(gapminder)
```

```
##      country continent year lifeExp      pop gdpPercap
## 1 Afghanistan      Asia 1952  28.801  8425333  779.4453
## 2 Afghanistan      Asia 1957  30.332  9240934  820.8530
## 3 Afghanistan      Asia 1962  31.997 10267083  853.1007
## 4 Afghanistan      Asia 1967  34.020 11537966  836.1971
## 5 Afghanistan      Asia 1972  36.088 13079460  739.9811
## 6 Afghanistan      Asia 1977  38.438 14880372  786.1134
```

```
# What is the proportion of countries in 1952 that have a life expectancy less than or equal to 40?
mean(gapminder[gapminder$year == 1952, ]$lifeExp <= 40)
```

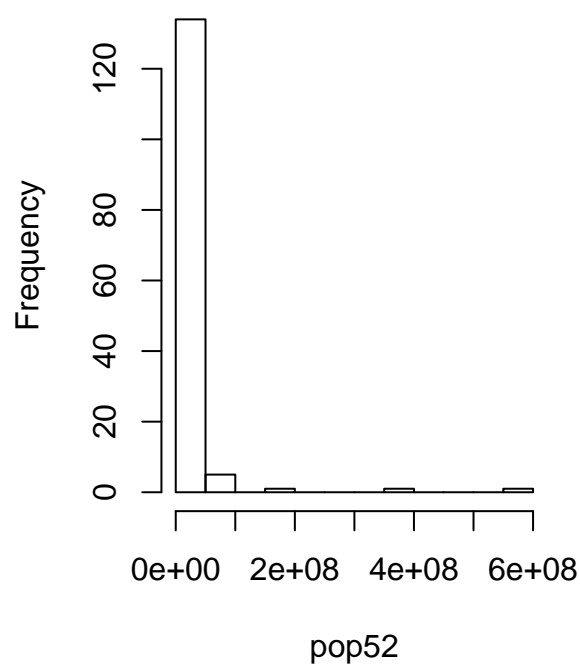
```
## [1] 0.2887324
```

Question 1.2: What is the proportion of countries in 1952 that have a life expectancy between 40 and 60 years? This is the proportion that have a life expectancy less than or equal to 60 years, minus the proportion that have a life expectancy less than or equal to 40 years. 0.4647887

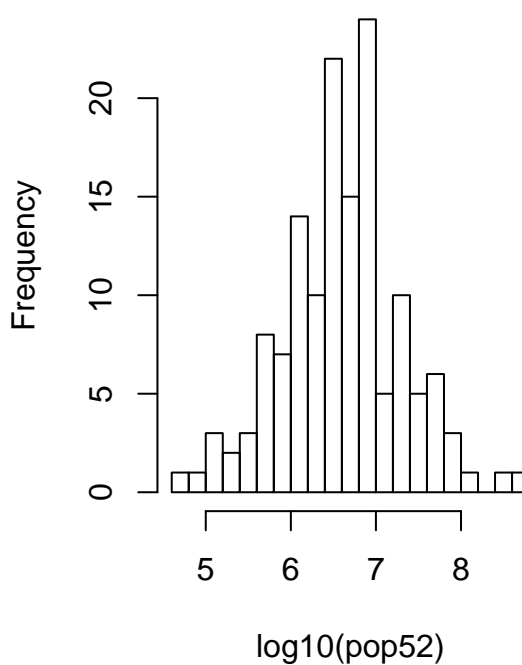
Question 2.1: Create a vector which gives the population sizes of the countries in 1952. Examine the histogram of these population sizes. Now examine the histogram of the log10 of these population sizes. What is the standard deviation of the log10 of population size of the countries in 1952?

```
pop52 <- gapminder$pop[gapminder$year == 1952]
par(mfrow = c(1, 2))
hist(pop52, breaks = 16)
hist(log10(pop52), breaks = 16)
```

Histogram of pop52



Histogram of log10(pop52)



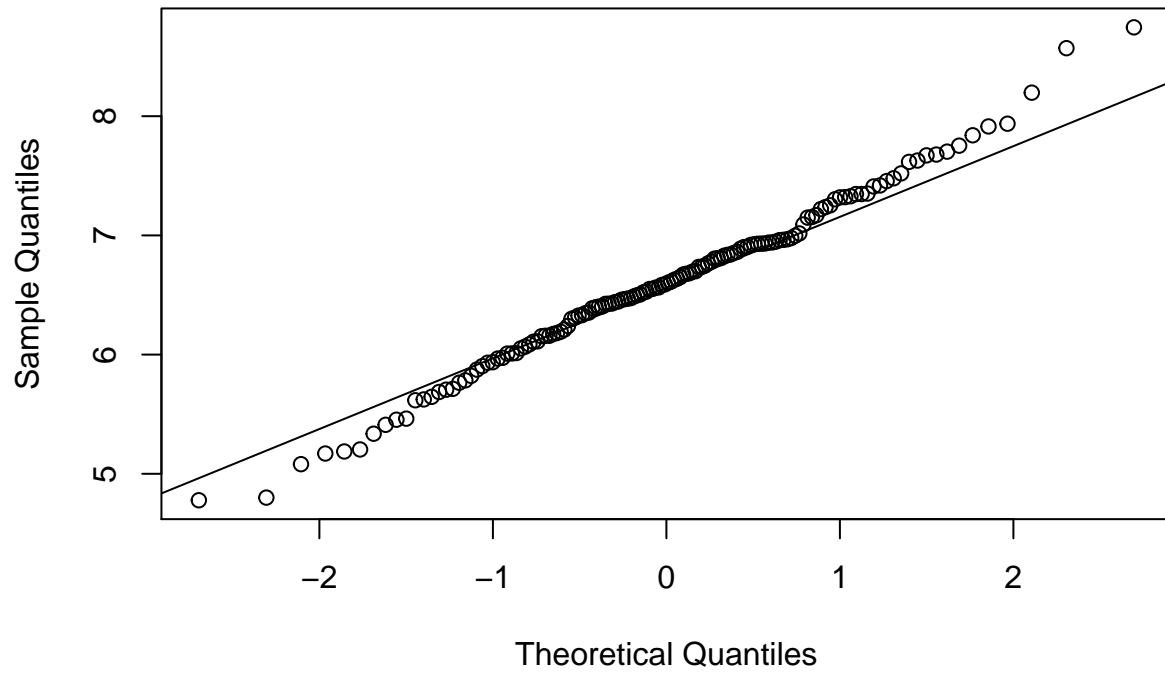
```
par(mfrow = c(1, 1))
# The standard deviation of the log10 of population size of the countries in 1952
sd(log10(pop52))
```

```
## [1] 0.7070292
```

Question 2.2:

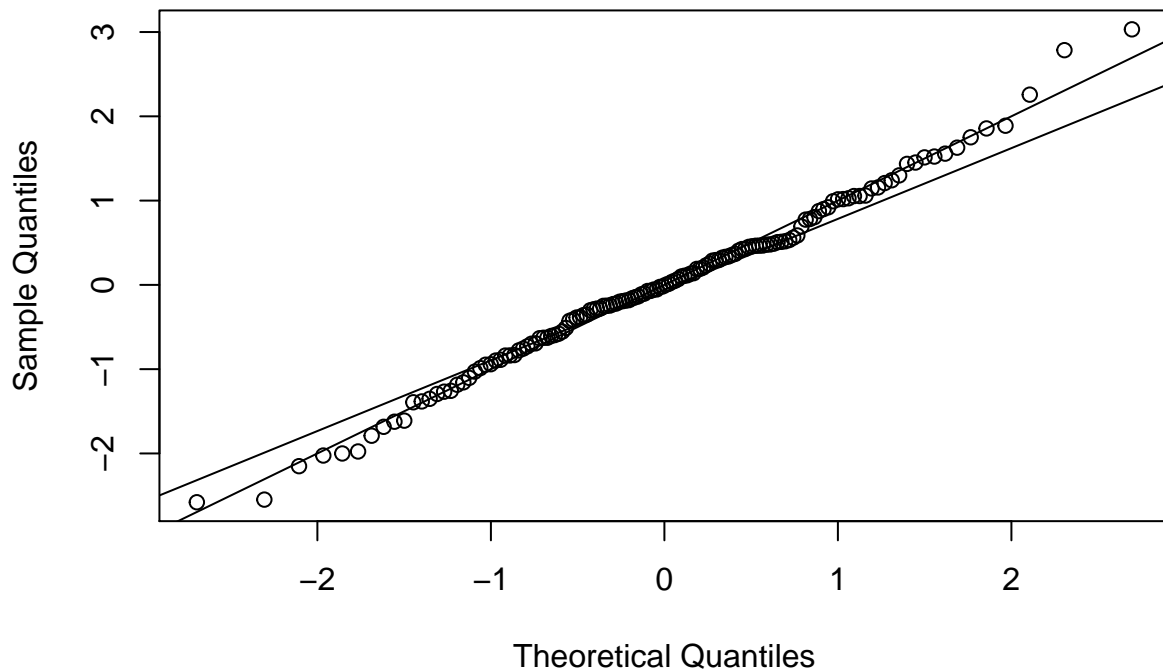
```
# Create a vector 'x' of the log10 of the 1952 population sizes
x <- log10(pop52)
qqnorm(x);qqline(x)
```


Normal Q-Q Plot



```
# Standardize the log10 population size vector  
z <- scale(x, center = TRUE, scale = TRUE)  
qqnorm(z);qqline(z);abline(0, 1)
```

Normal Q-Q Plot



```
# What is the z-score of the country with the largest population size?  
tail(sort(z), 1)
```

```
## [1] 3.03194
```

Question 2.3: Now we will use a Normal distribution approximation of the real distribution to ask questions about the expected proportions.

We will use the vector 'x' of the log10 of the 1952 population sizes for each country. We are going to create a function which gives back the Normal distribution cumulative density function for a Normal with the same mean and standard deviation as the vector x. This is accomplished by writing a one line function which uses pnorm(). pnorm() takes a value 'q', and returns the proportion of a Normal distribution which is less than or equal to 'q', for a Normal with a given mean and standard deviation.

```
F = function(q) pnorm(q, mean=mean(x), sd=sd(x))
```

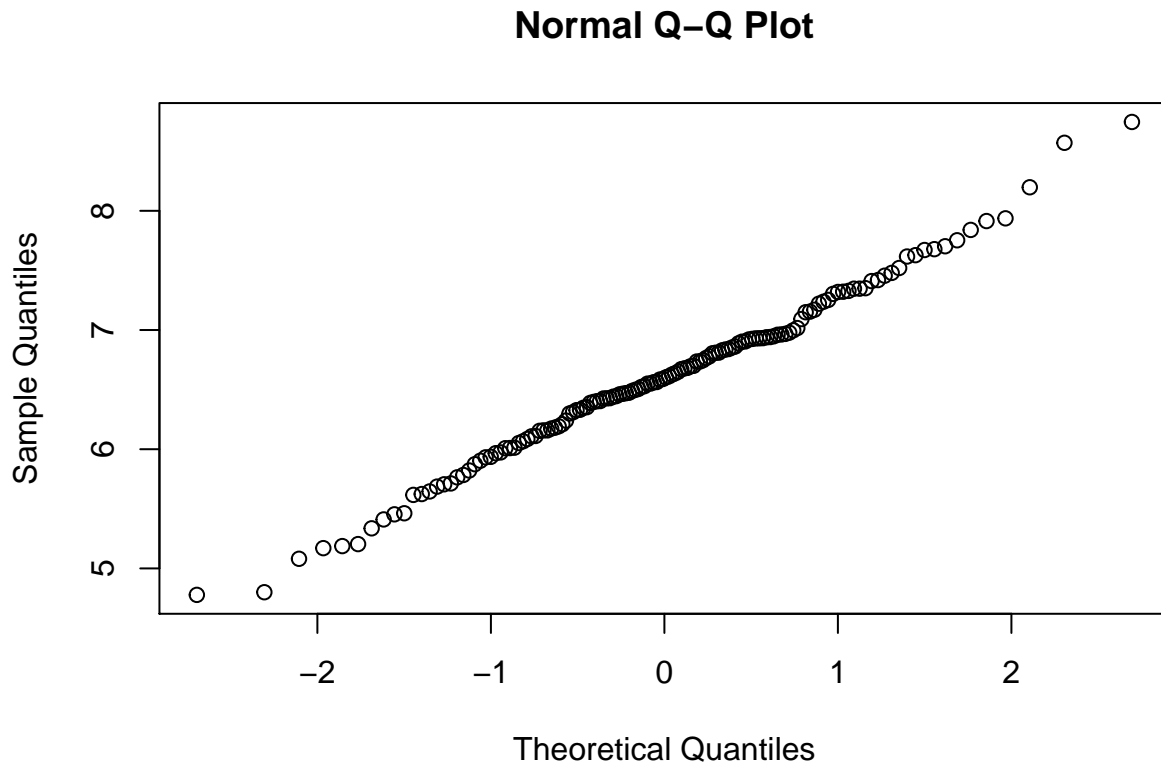
Using the Normal approximation, estimate the number of countries that should have a log10 1952 population between 6 and 7 (i.e., between 1 million and 10 million people)?

```
length(x)*(F(7) - F(6))
```

```
## [1] 73.27622
```

Question 2.4 Recreate a QQ plot from scratch. The plot we want to create is:

```
qqnorm(x)
```



... which shows the quantiles of a standard normal distribution (mean 0, standard deviation 1) on the x-axis, and the sample quantiles on the y-axis. If the points fall along a line, then the sample is approximately normal distributed. We can build this plot using the quantiles of a standard normal, and the same quantiles of our sample. First we need to find the number of observations in our sample:

```
n = length(x)
```

The sorted values of x represent n quantiles of the sample distribution spread from 0.0 to 1.0, however the quantiles do not include 0 and 1. Therefore, the sorted values of x are considered the $0.5/n, 1.5/n, 2.5/n, \dots, (n-0.5)/n$ quantiles of the sample distribution. Or in code, we want to find the quantiles of the standard normal distribution which are associated with the following probabilities:

```
ps = ((1:n)-0.5)/n
```

The matching quantiles of the standard normal distribution can be found by plugging in ps into `qnorm()`

What is the quantile of the standard normal distribution which matches to the smallest number in x (the first element of `sort(x)`)

```
qnorm(0.5/n)
```

```
## [1] -2.69484
```

We can then construct our plot, as we have the quantiles of the sample distribution, and can plug 'ps' into qnorm to get the matching sample quantiles of a normal distribution:

```
plot(qnorm(ps), sort(x))
```

