Week 2 Notes Part 3 Hypothesis Testing

Code

Author: Brendan Gongol

Last update: 05 January, 2023

1 Overview

In this note we learn how to compare two samples; "compare" here means answering the question "given the fluctuations and randomness in the data and the sampling process, is it likely that the two samples at hand came from the same underlying distribution, or the differences between the samples are so pronounced that they cannot be explained by chance and we have to conclude that the samples (or rather their underlying distributions) are truly different". We will first perform a numerical resampling experiment in a toy model derived from the ALL dataset; then we will look at t-test and its uses.

2 Comparing two samples

Let us now return to the ALL dataset. In previous week's notes we briefly looked at the distribution of ages of male and female patients. Let's reproduce those plots first:

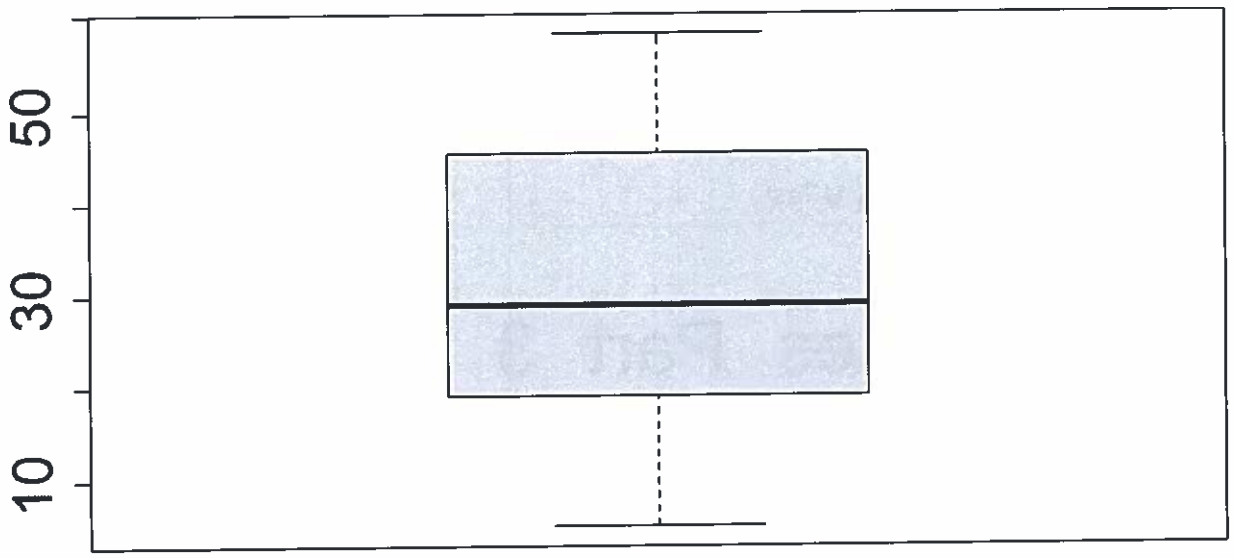
Hide

```
library(ALL); data(ALL)
pData(ALL)[1:5,1:5]                # take a quick look at patients data
```

```
##        cod diagnosis sex age BT
## 01005 1005 5/21/1997   M  53 B2
## 01010 1010 3/29/2000   M  19 B2
## 03002 3002 6/24/1998   F  52 B4
## 04006 4006 7/17/1997   M  38 B1
## 04007 4007 7/22/1997   M  57 B2
```

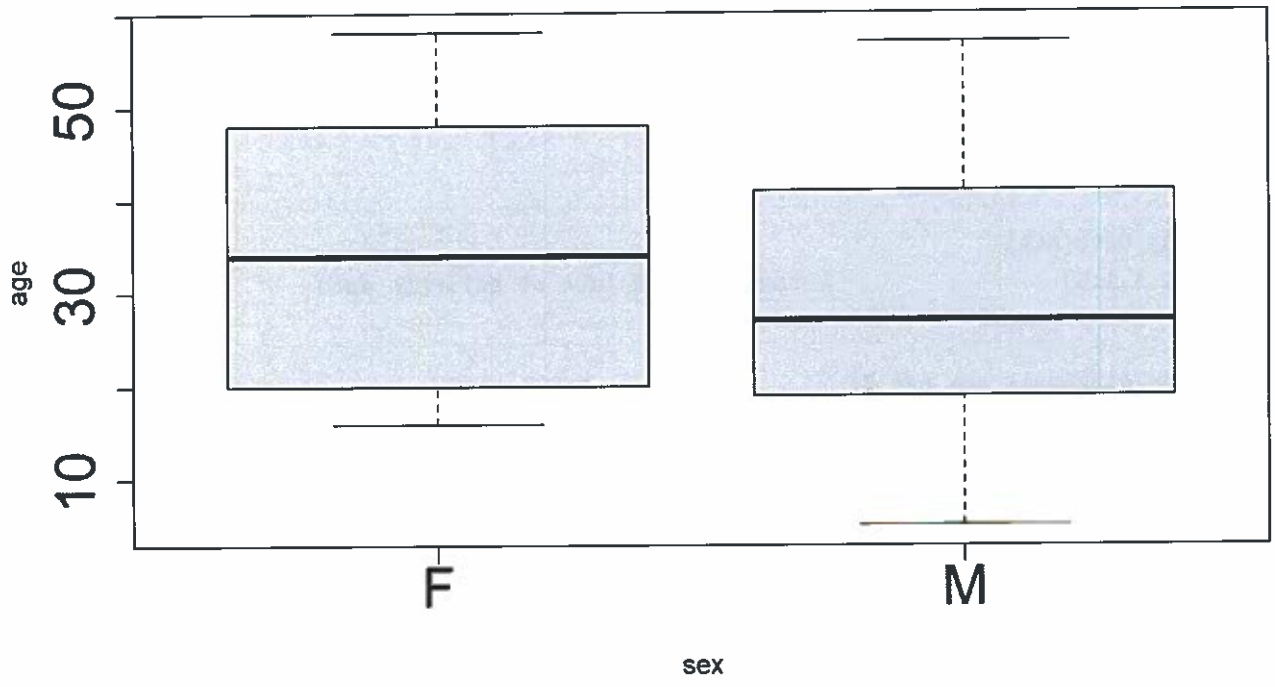
Hide

```
boxplot(pData(ALL)$age,cex.axis=2) # all patients
```



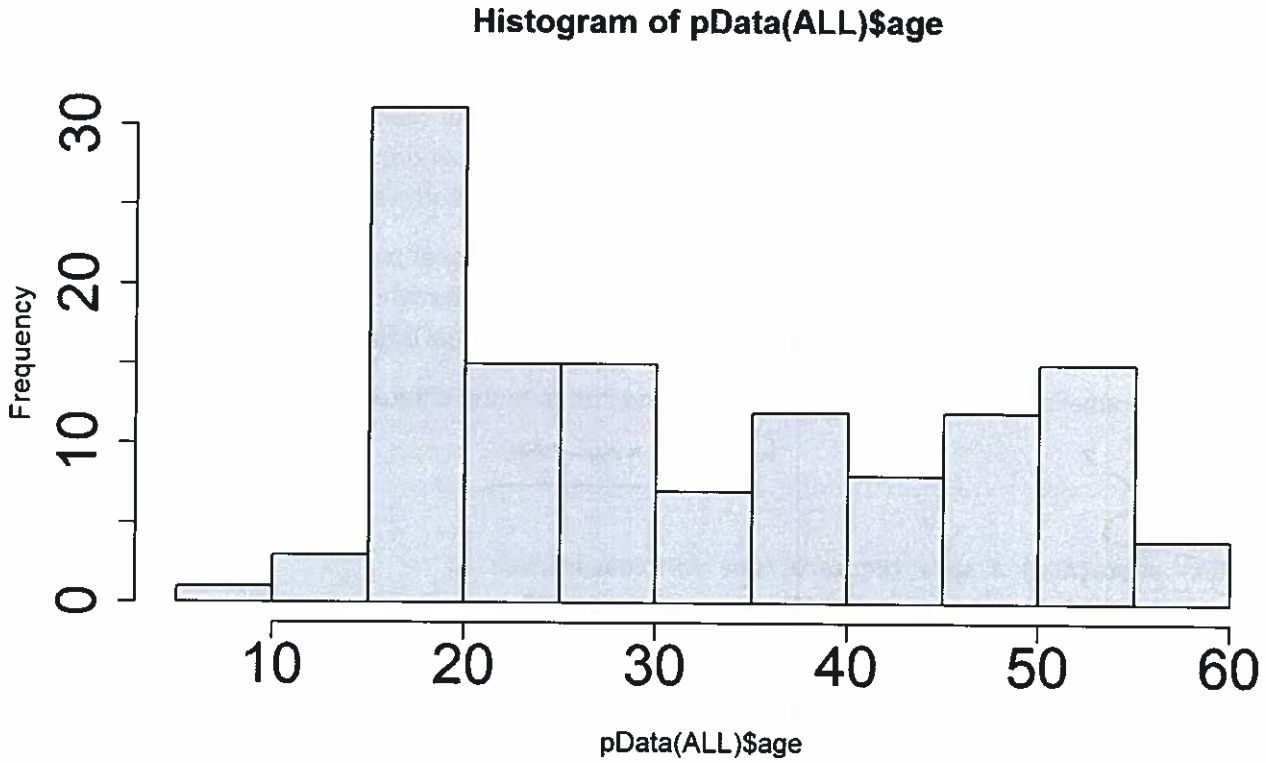
Hide

```
# stratified by gender (NOTE formula syntax!!):
boxplot(age~sex,pData(ALL),cex.axis=2)
```



Hide

```
# let's also check the empirical distribution:
hist(pData(ALL)$age,cex.axis=2)
```



As we can see from the second boxplot (the one in the middle), there is some observed difference between age distributions of male and female patients: the distribution for males seems to have lower median age and its overall mass is shifted to lower ages as well. But is this difference real? When we discussed population and sample statistics, we noted that samples are drawn randomly from the underlying population. It is possible that even if we draw two samples from the same underlying distribution, one will have lower mean and/or median than the other, simply due to the random nature of the sampling process. Hence, we can refine our question as follows: given the observed difference between the locations of the age distributions of males and females, how likely it is that these two samples (ages of males and females) were drawn from the same distribution? If it is likely, we have to assume that the observed difference is due to chance and there is no real effect; if it is unlikely, than the observed difference could be real.

How can we estimate whether the observed difference, D, could be explained by chance? In order to do so, we have to define what chance (what random process, specifically) has to be considered. Since we are looking at the difference between the means in the two samples, a reasonable approach is as follows: suppose we do have one common underlying distribution of ages of the leukemia patients (this is the hypothesis we are testing); we can then draw pairs of samples of the same size as M and F cohorts in our case over and over again, and every time compute the difference between the means of the two samples. If we observe differences as large or even larger than D very often, then it is logical to conclude that our initial observation of the difference D is very typical and easily expected as the result of random selection from the underlying distribution. On the other hand, if no matter how hard we try and how many times we re-draw samples from the underlying distribution, the difference between their means is always or almost always below our initial observation D, then we can conclude that observing D would be highly unlikely if the underlying distributions of male and female ages were the same.

The procedure described above represents brute force re-sampling. In our case, however, there is one problem: we do not have the underlying distribution we could re-sample from. We can try solving this problem in a few different ways. First, we could assume that underlying distribution has specific functional form, e.g. Gaussian. From looking at the empirical distribution of the sample that we have (a histogram in the right panel in the figure above), it does not look like a very good assumption: the histogram does not look very similar to the bell-shaped Gaussian distribution. We could try this nonetheless in the hope that while not very accurate, our estimates would still give us a reasonable approximation. Besides, as we will learn soon, if we do assume Gaussian underlying distribution, we do not really need to run brute-force re-sampling, but can instead use analytical machinery of the t-test. The second possible approach would be to try to model the underlying distribution using empirical distribution of the sample we have.

Here, instead, we will avoid these complications and for purely educational purposes perform the following experiment: we will postulate that patients present in the dataset are the whole population (let's reiterate: this is an illustration and this is probably not how you'd want to analyze real data).

First, let us draw a smaller sample from that population and compute the difference between the means:

Hide

```
set.seed(1234)
all.pdat <- pData(ALL) # save the dataframe for convenience
m.ages <- all.pdat[all.pdat$sex=="M","age"] # get all male's ages
f.ages <- all.pdat[all.pdat$sex=="F","age"] # all female's ages
x.m <- sample(m.ages[!is.na(m.ages)],9) # random sample: 9 male ages
x.f <- sample(f.ages[!is.na(f.ages)],11) # 11 random female ages
x.m
```

```
## [1] 21 30 28 33 17 27 37 57 37
```

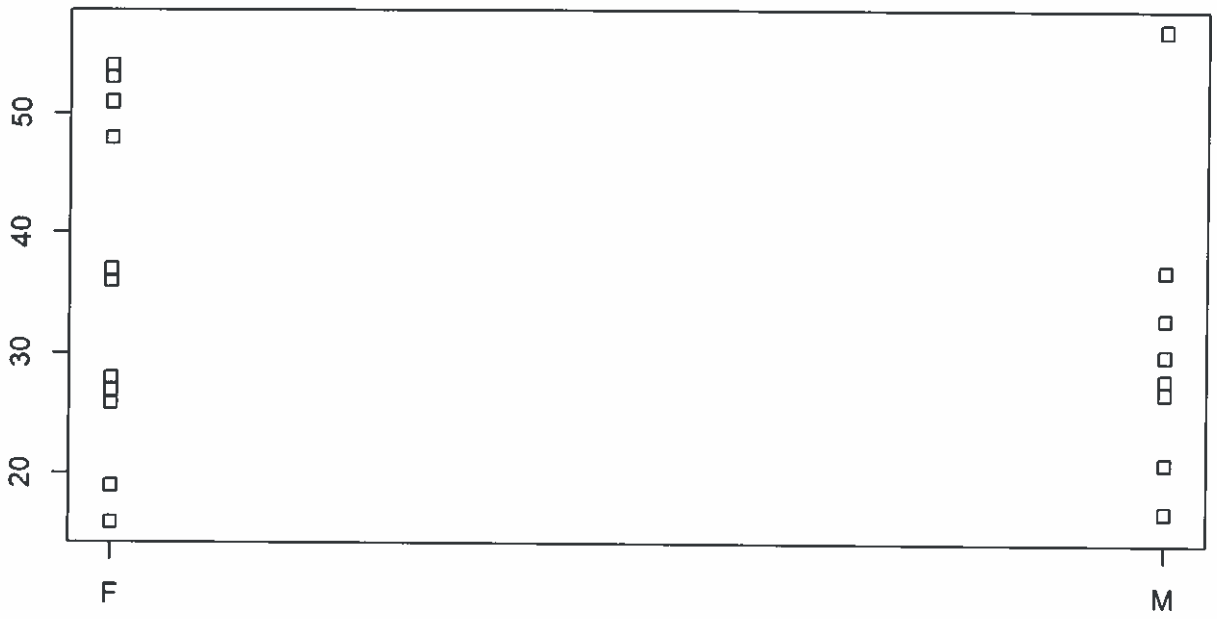
Hide

```
x.f
```

```
##  [1] 51 16 19 36 28 48 26 27 53 37 54
```

Hide

```
stripchart(list(F=x.f,M=x.m),vertical=TRUE) # compare samples
```



Hide

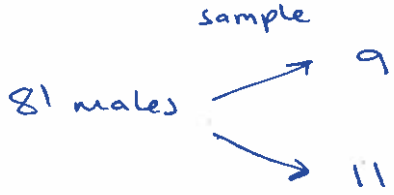
```
mean(x.f)-mean(x.m)
```

```
## [1] 4.020202
```

We started the experiment in the snippet above by selecting a sample of 9 males and 11 females (or rather their ages). The stripchart shows how the two selected samples compare, and in the last line we compute the difference between the means in our two samples.

Let's now define how big the difference could be if the samples were selected from the same distribution. Let's take the vector of ages of all 81 men present in the dataset as the "population distribution" and select two subsets, of sizes 9 and 11 from that vector. By doing that, we are modeling the assumption that distributions of male and female ages are exactly the same. If the originally observed difference is significantly larger than what we usually obtain using this assumption, then the latter is invalidated.

Hide



```
ori.diff <- mean(x.f)-mean(x.m) # diff between original samples
diff.sim <- numeric() # create an empty vector
ge.cnt <- 0 # how many times resampled diff is greater than original
n.sims <- 10000 # perform 10000 resamplings
for ( i.sim in 1:n.sims ) { # in each iteration:
 x.sim <- sample(m.ages[!is.na(m.ages)],20) # sample 20 ages…
 #... assign them to two samples and compute diff between means
 diff.sim[i.sim] <- mean(x.sim[1:9])-mean(x.sim[10:20])
 if ( diff.sim[i.sim] >= ori.diff ) {
 ge.cnt <- ge.cnt + 1
 }
}
```

Note that in the code above we simply select 20 age values from the set of all male ages, then use first 9 as the first sample and the remaining 11 as the second sample. This is legitimate, since sample() already returns its result in random order, so there is no need to additionally reshuffle it between the first and second sample. What we achieved so far is the following: we have two "original" samples, 9 males and 11 females, and the difference between their mean. We also have 10000 differences computed by repeated re-samplings of 9 and 11 values from the set of all male ages (i.e. from the same distribution).

which we calc mean(x.f) + mean(x.m from

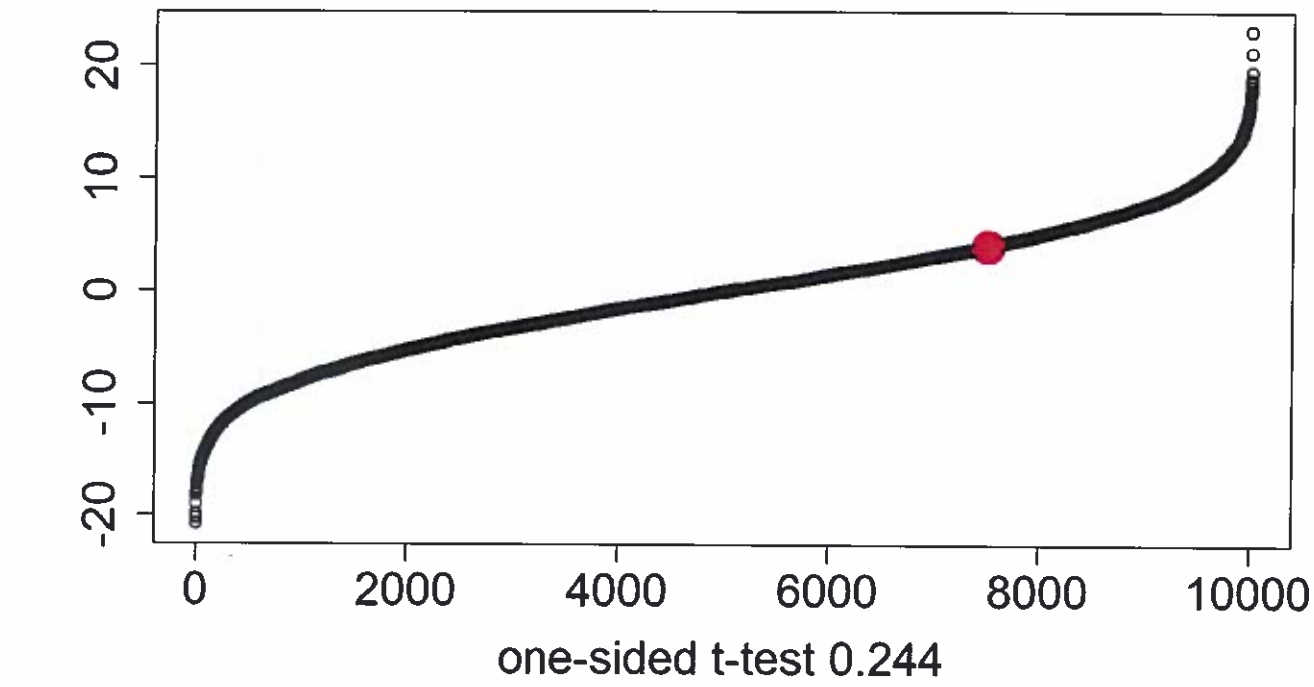
Let us now plot all those re-sampled differences and see how the original difference is located with respect to their distribution. Note that we put some additional information right into plot labels:

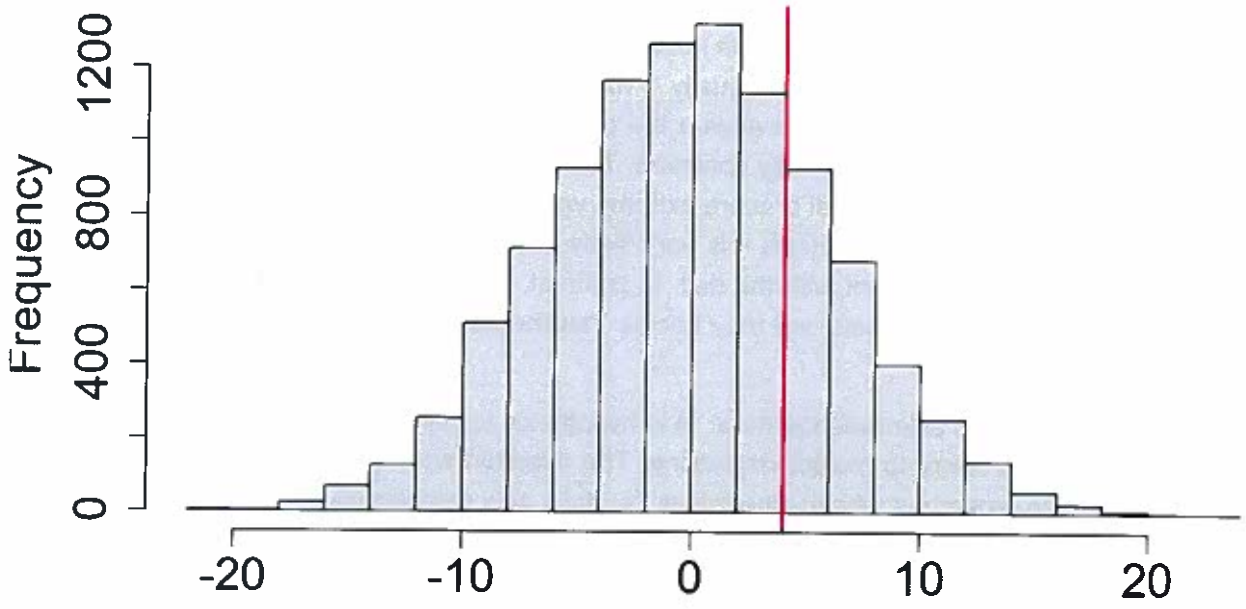
Hide

```
old.par <- par(mfrow=c(2,1),ps=20)
plot(sort(diff.sim),main=paste("Rank",
  ge.cnt/n.sims),ylab="",
  xlab=paste("one-sided t-test",
  signif(t.test(x.f,x.m,alt="greater")$p.value,3)))
points(n.sims-ge.cnt,ori.diff,
  cex=4,col="red",pch=20)
plot(hist(diff.sim,breaks=20,plot=F),main="",xlab="")
abline(v=ori.diff,lwd=2,col="red")
```

* of all males:
  1. take a sample of 20 ages
  2. (calc mean of 1st 9) - (calc mean of last 11)
     ↳ add this mean difference to diff.sim.
  3. if this mean difference is >, the original diff of mean between the original mean of female ages - mean of male ages, then add 1 to the counter (ge.cnt).

do this 1000 times!





Hide

```
par(old.par)
```

Note the first command in the snippet: `par()` sets graphical parameters (there are many, use docs!). In this case we instruct R to generate a single plot (in a single figure pane) consisting of the two vertically oriented panels. When graphics area is split into panels by `par(mfrow=...)`, each plotting command that would normally start a new plot (new device) will instead start drawing in the next available panel. The output of `par()` is previous graphics context, so it can be saved and restored later.

After splitting the graphics device into two panels, we first draw sorted set of re-sampled differences; then we indicate where the original difference is located on that curve (note that `points()` command does not start a new plot but adds to the existing one). Then we draw histogram (empirical distribution) of the re-sampled differences and indicate with a line where the original difference lies (`abline()` as well as `lines()` commands also add to the current plot).

The `hist()` command in `plot=F` mode returns an object with all the results; we wrap this command into a `plot()` command, just to show how different commands operate (normally you can just use `hist()` for plotting). The message is that `plot()` can also draw a histogram object. Red point and red line represent the location of the original difference in those distributions.

From our simulation experiment we see that when we tried drawing the second sample (11 patients), from the same set of all male ages as the first sample, we still ended up having ~12% retrials that resulted in even higher differences than originally observed. Based on that, we can hardly call originally observed difference a significant one. Interestingly, despite the fact that the distribution we are using is non-Gaussian, t-test still performed a very decent job and computed value (0.123) very close to the one we obtained by brute force re-sampling (0.119).

3 Hypothesis Testing: t-test

The approach we have taken in the previous section is very common. What we did is known as "hypothesis testing". First we formulate a null hypothesis – usually a minimalist statement assuming that there is no effect whatsoever. Under this assumption we evaluate the probability to observe the same or even more extreme value of some quantity than was actually observed. The quantity chosen for testing is known as test statistics and the probability to observe equal or more extreme value of test statistics is called p-value. If this probability is high, i.e. under the null hypothesis it is very likely to make observations as or more extreme than we did, then our data are consistent with the null. In contrast, if the probability of the observation we made is too low under the null hypothesis, we may decide that the data contradict the null and the latter should be rejected.

There are two important points to acknowledge here: 1) in hypothesis testing, the "probability to observe the value of test statistic" always refers to multiple re-testing. The question we are asking is, invariably, "what would happen if we were to repeat our whole experiment, i.e. draw new samples, over and over again – how often would we observe the test statistics as extreme, assuming the null hypothesis were true?" In our example discussed above, the test statistic was the difference between the means of the two samples and we chose to perform resampling explicitly through brute force simulation. In many cases the probability can be estimated and/or approximated analytically, using standard statistical tests – those simply employ some clever math that can tell what would happen, under certain assumptions, if we were to run those retrials, without actually running the latter; 2) the test statistics is calculated for the sample (e.g. mean of the sample) and is thus a random variable itself; hence we can speak only about the probability for the test statistic to exceed any particular value. In this sense, we can only tell that a particular observation is very unlikely under the null hypothesis (i.e. has very low probability, or p-value). It is an arbitrary and subjective decision whether the probability is low enough to reject the null. One traditionally used value of the p-value cutoff is

0.05 (i.e. 5% probability that there is in fact no effect, null is correct, and the observed difference is due to chance); in many applications however, you may want to choose much more stringent cutoffs, especially in multiple testing as we will discuss later on.

3.1 T-test

Now that we understand the general logic of hypothesis testing, we can consider a very important special case: the t-test, originally developed in 1908 by W.Gosset, a chemist at Guinness (pen name "Student" - thus Student's t-test; for a bit of trivia – Guinness required Gosset to publish under a pen name since the fact that they might use statistical analysis in their manufacturing process was deemed a trade secret, or so the story goes anyway). The test statistics of this test (known as t-statistics) is pretty much either the mean or difference between the means, properly re-scaled (there are flavors of the test for one or two samples), and the test assumes the underlying distributions(s) to be normal (Gaussian); importantly, by making these assumptions, one can obtain an analytical result and thus avoid performing 10000 re-samplings every time one needs to test for significance. In the single-sample case (the null is that the sample came from a distribution with specific mean, and the observed difference between that mean and sample mean is solely due to chance), the t-test machinery uses the following:

sample mean — $\bar{X}_n = \frac{(X_1 + \dots X_n)}{n}$ — null hypothesis = due to chance.

unbiased sample variance $S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2$ — when we reject the null hypothesis we are stating that there is a real difference.

$T = \frac{\bar{X}_n - \mu}{S_n/\sqrt{n}}$ — μ = population mean

We are already familiar with the sample mean and (unbiased) sample variance shown above. Those quantities are used to define the t-statistics T. Note that T still reflects the difference between the observed (sample) mean and the underlying population mean, μ. The denominator is a normalization factor. Intuitively, imagine two distributions with the same mean but different variances, one small and the other large. If we repeatedly sample from the distribution with small variance, our random samples will be also distributed relatively tightly around the true mean, i.e. the sample variance will be small. The samples from the second distribution will be "all over the place" simply because the underlying distribution is wider. It is thus natural to expect that since the samples in the second case are usually more spread, the fluctuations of their sample mean from sample to sample will be also larger, but this is still simply chance. By normalizing the deviation of the sample mean from the underlying (expected) population mean by the variance, we remove the aforementioned effect of differences between magnitudes of sample mean fluctuations due to different widths of underlying distributions. It is possible to derive analytically a closed form expression for the distribution of the t-statistics:

$$f(x) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\Gamma(\frac{v}{2})}(1 + \frac{t^2}{v})^{-}(\frac{v+1}{2})$$

where the "number of degrees of freedom" $\nu = n - 1$ (n is the sample size). Note that this expression is universal in a sense that it does not depend on the sample variance. While this expression does involve special functions, it is much more efficient computationally to calculate a value of gamma-function than to

perform large number of brute-force re-samplings; it had even more of an advantage in the old days of low computational power (or none at all), since the values of special functions were simply tabulated, while brute-force re-sampling was largely out of question.

According to the formulas given above, application of one sample t-test thus amounts to the following (R will do it for you, but it's good to understand what goes on under the hood): take the sample, calculate its mean, take the difference between the sample mean and the mean of the underlying distribution that we hypothesize (that's our null hypothesis), normalize by the sample variance to arrive to the value of t-statistics for our sample. Look at the known distribution of the t-statistics (shown above) and calculate from that distribution how likely it is to observe the value of t as extreme (i.e. calculate the "mass" of the part of that distribution, in other words the area under the curve, at values greater than observer t).

The "classical" formulation of the t-test given above involves just one sample: we expect (hypothesize) that the underlying distribution has mean μ, and we are asking if a sample at hand contradicts this assumption too much (i.e. the probability to draw such sample from the hypothesized distribution is too low). A two-sample formulation is also available. In this formulation, very much alike the numerical experiment we ran above, our null hypothesis is that the two independently drawn samples at hand were drawn from the same underlying distribution, or strictly speaking, from the underlying distributions with the same means, and the observed difference between the means of the two samples can be thus explained by chance. If the p-value (probability to observe such or even more extreme value of the difference when drawing from the distributions with the same mean) is too low, we have to reject the null and conclude that the underlying distributions actually have difference means. Since we are dealing with two samples, the null hypothesis needs to be refined. In addition to assuming that the underlying distributions the two samples came from have equal means, we

have to make an assumption about their variances. In the simple case, we can assume that the variances are also the same (and since normal distribution is completely characterized by its mean and variance, we are thus assuming in this case that the two underlying distributions are actually one and the same distribution) – this is the original two-sample t-test. Alternatively, we can consider a more general case and assume that the variances of the two underlying distributions are different. This variant of the two sample t-test, also known as Welch test uses the following definition of the t-statistics:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

This statistics still follows the t-distribution shown above, and after redefining t as shown above, all the rest stays the same.

3.2 Use of t-test

When the t-test can be actually used and when its results can be relied upon is a somewhat contrived issue. The test is very attractive because it is simple and computationally efficient. However, it explicitly relies on the assumption of normality of the underlying distribution(s). It turns out (it is possible even to run some brute-force simulations along the lines of what we have done above) that if the distributions are reasonable (first of all unimodal, i.e. have only one maximum), the p-value calculated by t-test is still not too far off the correct p-value. The exact value of the t-test p-value cannot be relied upon for non-normal distributions, but it often can be used for rough classification: if t-test p-value computed for samples drawn from non-normal distribution is, e.g., 0.3, the difference is insignificant (the correct p-value could be 0.2 or 0.4 but that would not change the conclusion). If the t-test p-value is 0.001, the difference is likely significant (although the

correct p-value could be equal to 0.0001 or 0.01 – we do have to calculate it if we need exact significance level). In other words, t-test is usually sufficiently robust for working with non-normal but "reasonable", unimodal distributions. In practical situations, it is worth verifying that this is indeed the case with specific data at hand (e.g. by doing limited-scope resampling experiment).

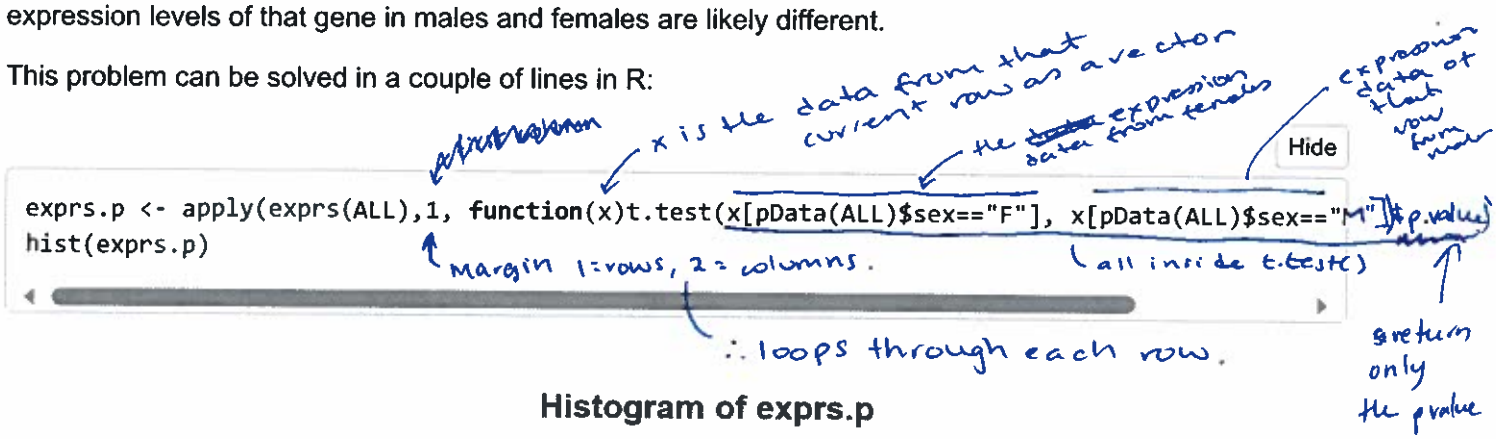
3.3 t-test example

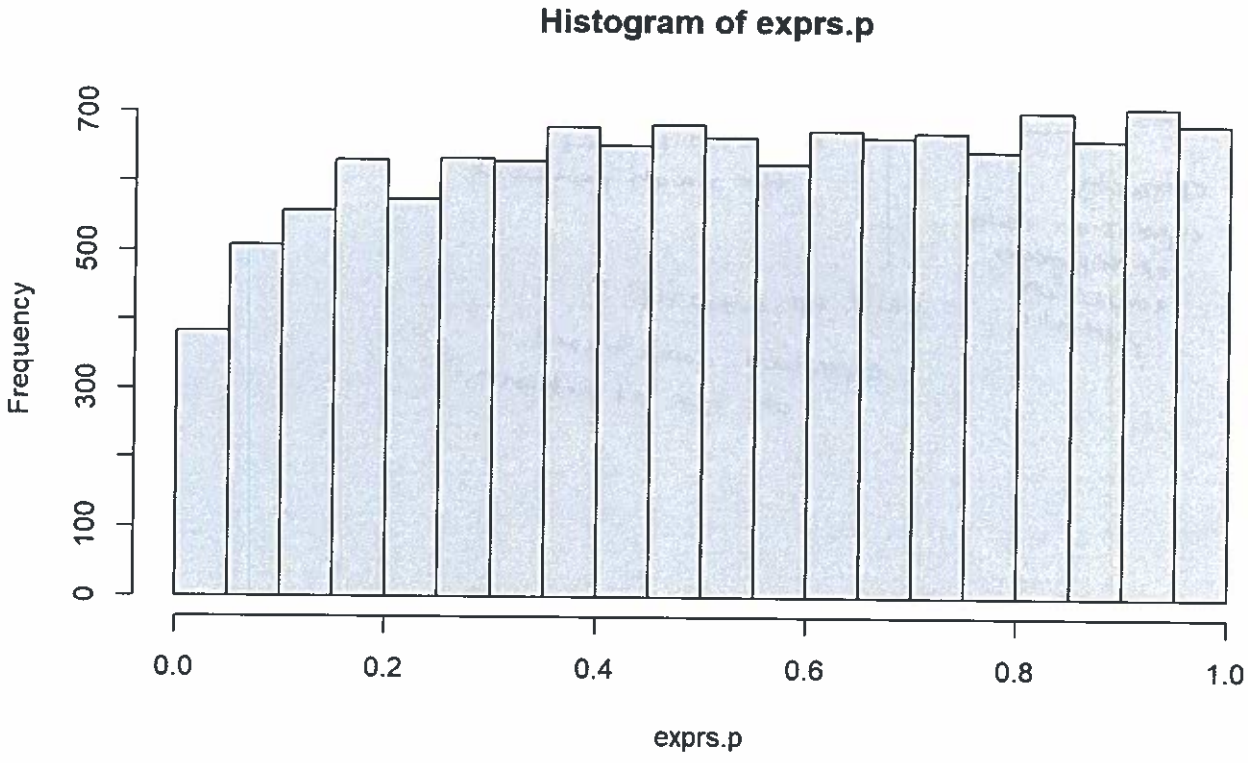
To conclude our discussion, let us consider a simple example of the use of t-test. We will ask ourselves a question: are there genes in the ALL dataset that exhibit significant differences in expression levels between males and females? This is a classical setting for hypothesis testing. Indeed, we have samples (measured values) of highly fluctuating quantities (gene expression levels, strongly varying from patient to patient in general). For each gene, individually, we consider its measured expression levels in males and females, respectively, as the two samples and we ask if, given the overall variance in expression levels, it is likely that these two samples were drawn from the same underlying distribution (i.e. there is only some overall patient-to-patient variation, but no specific bias between males and females with regard to that gene's expression), or, on the contrary, the difference between these two samples is so large that the distributions for the expression levels of that gene in males and females are likely different.

This problem can be solved in a couple of lines in R:

Hide

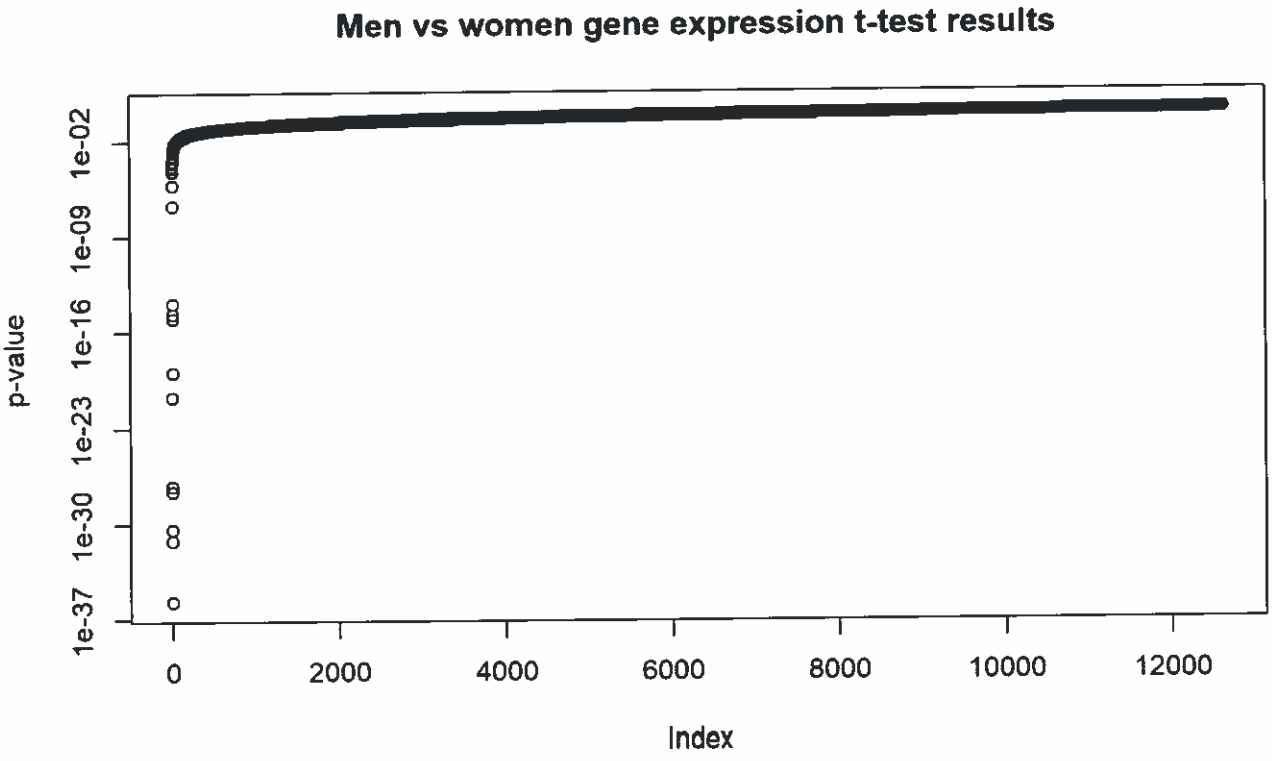
```
exprs.p <- apply(exprs(ALL),1, function(x)t.test(x[pData(ALL)$sex=="F"], x[pData(ALL)$sex=="M"])$p.value)
hist(exprs.p)
```





Hide

```
plot(sort(exprs.p),log="y",ylab="p-value", main="Men vs women gene expression t-test results")
```



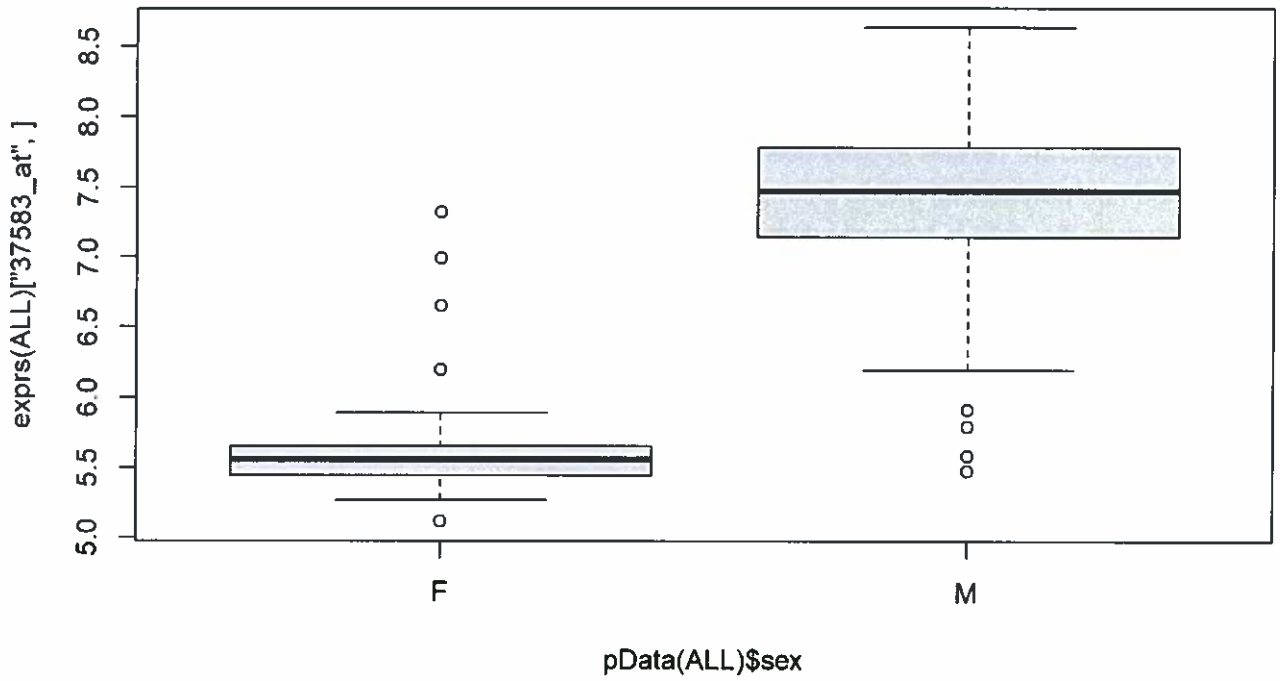
Hide

```
boxplot(exprs(ALL)["37583_at",]~pData(ALL)$sex)
```

extracts expression data of this gene across all patients

extracts sex information for each patient

tells R to group the expression data based on sex of patient.



First, we apply t-test. Note the use of the apply() function, which is a convenient (and more efficient in R) way of "applying the specified function to each row or column of a matrix", i.e. it replaces an explicit loop. We apply our function (t-test) to every row (gene), as specified by the `MARGIN=1` parameter. The function itself receives each row of the matrix as its (vector) argument – that's what `apply()` promises to do for us; inside our custom function we split the row (vector of gene expressions) passed into it into values measured in males and in females, respectively (we do that by using M/F annotations provided in the `pdata(ALL)`), and then we run t-test on those two samples. From the data structure (list) returned by t.test() function we extract only one field, the calculated p-value, and this is what our custom function returns. When `apply()` runs our function on each row of a matrix it generates a single value (returned by our custom function) per row, and the result is thus a vector with one element (p-value in our case) per input row.

Next, we draw a histogram of those p-values calculated for all the genes (rows of input matrix) and also plot the sorted p-values. Note that there is only a handful of genes with small p-values, i.e. those that exhibit significant differences in expression between males and females. To illustrate our findings, we select one gene with very small p-value and draw (observed) distributions of its expression levels in males and females side by side using boxplot and employing the formula syntax ('~') in order to make boxplot stratify the data automatically. The boxplot illustrates that the distributions indeed look very differently. If we looked up this Affymetrix microarray probe, we would discover that this gene is actually located on Y chromosome, so it's not totally surprising that it is expressed differently between males and females. Note that depending on the problem at hand, this trivial observation can be legitimate or can become a very annoying confounder. Suppose you have a treatment and control group and you are comparing gene expression levels across the two groups, just like we did it above for males vs. females. Next suppose that control group has more females – the sample could end up containing more females just by chance because of random sampling; or there could be real bias due to some additional circumstances. As the result, we could find many genes with "significant" changes in expression between the treatment and control group which are in fact Y-chromosome genes or are otherwise linked to patients sex, not the treatment!

4 testing for data normality

All hypothesis tests have specific assumptions about the data being analyzed that must be correct in order for the accuracy of the test to be optimal. Therefore, prior to performing any hypothesis testing on the data, the correct test needs to be selected that matches the criteria of the dataset being analyzed. Of the assumptions about the data, one of the most commonly observed data assumptions observed in clinical and genomic data analyses are related to data normality. Often, genomic data and clinical data is non-normally distributed (called non-parametric) and is an important feature of the data to understand before performing t-tests, which assumes the data is normally distributed. There are several methods of determining if the data is normally distributed included creating a QQ-plot. Although QQ-plots are useful visualizations of the data, kin this illustration, we will use a Shapiro-Wilk test of normality to quantify the normality of the data which is defined by the following:

$$W = \frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n}(x_1 - \bar{x})^2}$$

In essence, the Shapiro-Wilk test tests the null hypothesis that $x_1, \ldots, x_n$ came from a normally distributed population. Therefore, a non-significant value (i.e $p < 0.05$) indicates the data is not significantly different from a normal distribution and is itself normally distributed. We can see this in the following example. First, we will load a dataset from the `medicaldata` package:

Hide

```
# install.packages("medicaldata")
library(medicaldata)
head(blood_storage)
```

```
##   RBC.Age.Group Median.RBC.Age  Age AA FamHx  PVol TVol T.Stage bGS BN+
## 1             3             25 72.1  0     0  54.0    3       1   3   0
## 2             3             25 73.6  0     0  43.2    3       2   2   0
## 3             3             25 67.5  0     0 102.7    1       1   3   0
## 4             2             15 65.8  0     0  46.0    1       1   1   0
## 5             2             15 63.2  0     0  60.0    2       1   2   0
## 6             3             25 65.4  0     0  45.9    2       1   1   0
##   OrganConfined PreopPSA PreopTherapy Units sGS AnyAdjTherapy AdjRadTherapy
## 1             0    14.08            1     6   1             0             0
## 2             1    10.50            0     2   3             0             0
## 3             1     6.98            1     1   1             0             0
## 4             1     4.40            0     2   3             0             0
## 5             1    21.40            0     3   3             0             0
## 6             0     5.10            0     1   3             0             0
##   Recurrence Censor TimeToRecurrence
## 1          1      0             2.67
## 2          1      0            47.63
## 3          0      1            14.10
## 4          0      1            59.47
## 5          0      1             1.23
## 6          0      1            74.70
```

next, we will create a function that loops through columns of the data set and performs a Shapiro-wilk test of normality on each column independently. In the function arguments, `DT` is the data table containing the data and `cols` contains the column numbers that the test should be performed on.

Hide

```
normalityfindR <- function(DT, cols){
  finDT <- data.table()
  for(i in 1:length(cols)){
    if(length(table(as.numeric(DT[[cols[i]]]))) > 1){ - if # rows in the column > 1
    test <- shapiro.test(as.numeric(DT[[cols[i]]])) - perform shapiro on that column values.
    if(test$p.value < 0.05){
      distribution <- "Not normally distributed" - reject null hypothesis that it follow a normal distribution
    }else{
      distribution <- "normally distributed"
    }
    Nobs <- sum(!is.na(DT[[cols[i]]])) - checks for NA's, then reverses it. ∴ sum is counting how many non-NA values are present in that column. ~~sum values in that column~~ ← NO!
    DT2 <- data.table(method = test$method,
                      pvalue = test$p.value,
                      variable = colnames(DT[,cols[i], with = FALSE]),
                      normality = distribution,
                      NumberObservations = Nobs
    )
    finDT <- rbind(finDT,DT2) ← DT2 is being recreated for each column specified in the cols arguement. Here, we are adding this 1x5 table to finDT at each iteration.
    } else { ie = ∅
      print(paste("Aw Snap. All values in", colnames(DT)[cols[i]], "are identical"))
    }
  }
  return(finDT)
}
```

create new data table

Now that our function is loaded, we can perform the test on the sample dataset on columns containing continuous data.

Hide

```
library(data.table)
normalityfindR(DT=as.data.table(blood_storage), cols=c(2,6,12,20))
```

```
##                         method       pvalue         variable
## 1: Shapiro-Wilk normality test 5.250461e-21   Median.RBC.Age
## 2: Shapiro-Wilk normality test 1.007383e-23             PVol
## 3: Shapiro-Wilk normality test 2.976355e-23         PreopPSA
## 4: Shapiro-Wilk normality test 2.514174e-13 TimeToRecurrence
##                    normality NumberObservations
## 1: Not normally distributed                316
## 2: Not normally distributed                307
## 3: Not normally distributed                313
## 4: Not normally distributed                315
```

which column

Based on these results, we see that the data we performed the test on are not normally distributed. Therefore, performing t-test hypothesis testing on these samples will not provide accurate results and we will need to use a non-parametric test of significance when identifying relationships with data in these columns.

5 Hypothesis Testing: Mann-Whitney U Test

In the event that a dataset is not-parametrically distributed a Mann-Whitney U Test (aka Wilcoxon Rank Sum Test) can be used for hypothesis testing and is defined as follows:

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$$

$$U_2 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_2$$

n_1, n_2 = sample sizes

R_1 = Sum of ranks for first sample. (see summarized notes for meaning)

and the assumptions of this test are as follows:

- Comparison groups must be independent of one another.
- The data is continuous.
- Both samples that are compared are random.

in R, this test can be performed using the `wilcox.test()` function. Using our example from the previous section, let's say we wanted to determine if there is a significant difference between `Median.RBC.Age` and `PreopPSA` (not that comparing these two variables will be meaningful. In this example, we really only want to illustrate how to perform the analysis). This could be performed with the following code:

Hide

```
wilcox.test(blood_storage$Median.RBC.Age, blood_storage$PreopPSA, conf.int = TRUE, paired = FALSE, formula = "lhs")
```

data frame; column of interest; 2nd column of interest

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  blood_storage$Median.RBC.Age and blood_storage$PreopPSA
## W = 88005, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  7.199952 9.200053
## sample estimates:
## difference in location
##               8.300047
```

extremely low, reject null hypothesis & conclude the observed difference between two groups not by chance.

Wilcoxon rank sum statistic. It is used to calculate the P value