Permutation Test

Elvan Ceyhan

11/11/2022

```
library(tidyverse)
```

Permutation Testing

Permutation tests are a type of randomization test. The theoretical difference between permutation tests and inferential tests is that with permutation tests we build the sampling distribution from the observed data, rather than inferring or assuming that a sampling distribution exist.

In practice, what a permutation test does is to take your observed data and then shuffle (or permute) part of it. After each shuffle, some aspect of the data is recalculated. That could be for instance the correlation coefficient, or it could be a difference in means between two groups. The data then get randomly reshuffled again, and the test-statistic is recalculated again. This goes on for thousands of times - for as many shuffles are deemed acceptable. This is usually a minimum of 1,000 but typically at least 10,000 shuffles are done. After all the permutations (shuffles) are performed, a distribution of the statistic of interest is generated from the permutations. This is compared to the original observed statistics (e.g. correlation coefficient, difference in group means) to see if the observed value is unusually large compared to the permuted data.

If this seems a little confusing, hopefully seeing it in action will help…

Example 1

Let's take a look at an example from clinical trials. Here, we have various subjects' ratings their anxiety levels. They do this after either taking a new anxiolytic drug or a placebo. The subjects in each group are independent of each other. The placebo group has 19 subjects and the drug group has 21 subjects.

The data:

```
placebo <- c(15, 16, 19, 19, 17, 20, 18, 14, 18, 20, 20, 20, 13, 11, 16, 19, 19, 16, 10)
drug <- c(15, 15, 16, 13, 11, 19, 17, 17, 11, 14, 10, 18, 19, 14, 13, 16, 16, 17, 14, 10, 14)

n=length(placebo) #19
m=length(drug) #21
c(n,m)
```

```
## [1] 19 21
```

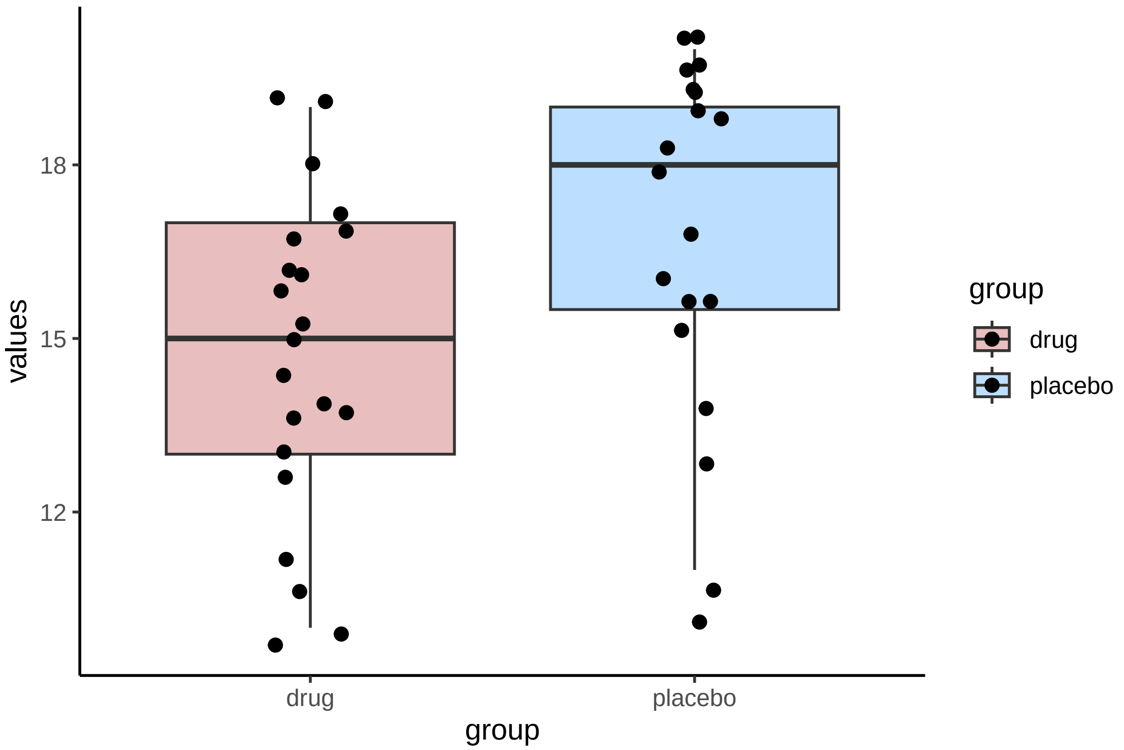
We put the data into a dataframe:

```
dd <- data.frame(values = c(placebo, drug),
                 group = c(rep("placebo",n), rep("drug", m)) )
head(dd)
```

```
##   values   group
## 1     15 placebo
## 2     16 placebo
## 3     19 placebo
## 4     19 placebo
## 5     17 placebo
## 6     20 placebo
```

We can plot these data as boxplots to get a sense of the within group variation as well as the observed differences between the groups:

```
ggplot(dd, aes(x = group, y = values, fill = group)) +
  geom_boxplot(alpha=.3, outlier.shape = NA) +
  geom_jitter(width=.1, size=2) +
  theme_classic() +
  scale_fill_manual(values = c("firebrick", "dodgerblue"))
```



Now, from our two independent samples, we can directly observe what the difference in sample means is. This is just calculated by subtracting one sample mean from the other:

```
mean.diff <- mean(placebo) - mean(drug)   # 2.13
mean.diff
```

```
## [1] 2.12782
```

So, from our samples, we observed a difference in grades of 2.13 between the groups. Typically, we would run an independent *t*-test to test whether these two samples came from the same or different populations:

```
t.test(placebo, drug) #can add var.equal=T option here
```

```
##
##  Welch Two Sample t-test
##
## data:  placebo and drug
## t = 2.3057, df = 36.187, p-value = 0.02697
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.256502 3.999137
## sample estimates:
## mean of x mean of y
##  16.84211  14.71429
```

This *Student's t-test* suggests that this is a significant difference, meaning that the groups do differ in their population means.

However, this test relies on several assumptions (see Lecture 13). Instead, we could apply a *permutation test* that is free of assumptions (except within and between sample independence).

Essentially what we are going to do is ask how surprising it was to get a difference of 2.13 given our real data. Put another way, if we shuffled the data into different groups of 19 and 21 (the respective sample sizes of placebo and drug), would we get a difference in sample means of greater or lower than 2.13. If we did this thousands of times, how many times would we get differences in sample means above 2.13?

Let's apply this theory to just one permutation.

First, we combine all the data:

```
set.seed(2) # just to keep the random number generator the same for all of us
all.levels <- c(placebo, drug)
all.levels
```

```
##  [1] 15 16 19 19 17 20 18 14 18 20 20 20 13 11 16 19 19 16 10 15 15 16 13 11 19
## [26] 17 17 11 14 10 18 19 14 13 16 16 17 14 10 14
```

Next, we shuffle them into new groups of 19 and 21:

```
x <- split(sample(all.levels), rep(1:2, c(n,m)))
x
```

```
## $`1`
##  [1] 15 16 20 14 19 14 19 14 13 20 14 18 16 20 15 19 16 19 11
##
## $`2`
##  [1] 10 16 18 17 10 13 19 17 17 16 15 13 19 11 14 17 20 16 10 18 11
```

We have two brand new samples that contain all of the scores from our original data, but they've just been shuffled around. We could look at what the difference in sample means is between these two new samples:

```
x[[1]] # this is our shuffled sample of size 19
```

```
##  [1] 15 16 20 14 19 14 19 14 13 20 14 18 16 20 15 19 16 19 11
```

```
x[[2]] # this is our shuffled sample of size 21
```

```
##  [1] 10 16 18 17 10 13 19 17 17 16 15 13 19 11 14 17 20 16 10 18 11
```

```
mean(x[[1]])  # mean of the new sample of size 19
```

```
## [1] 16.42105
```

```
mean(x[[2]])  # mean of the new sample of size 21
```

```
## [1] 15.09524
```

```
# what's the difference in their means?
mean(x[[1]]) - mean(x[[2]])
```

```
## [1] 1.325815
```

The difference in sample means is 1.32, which is much smaller than our original difference in sample means.

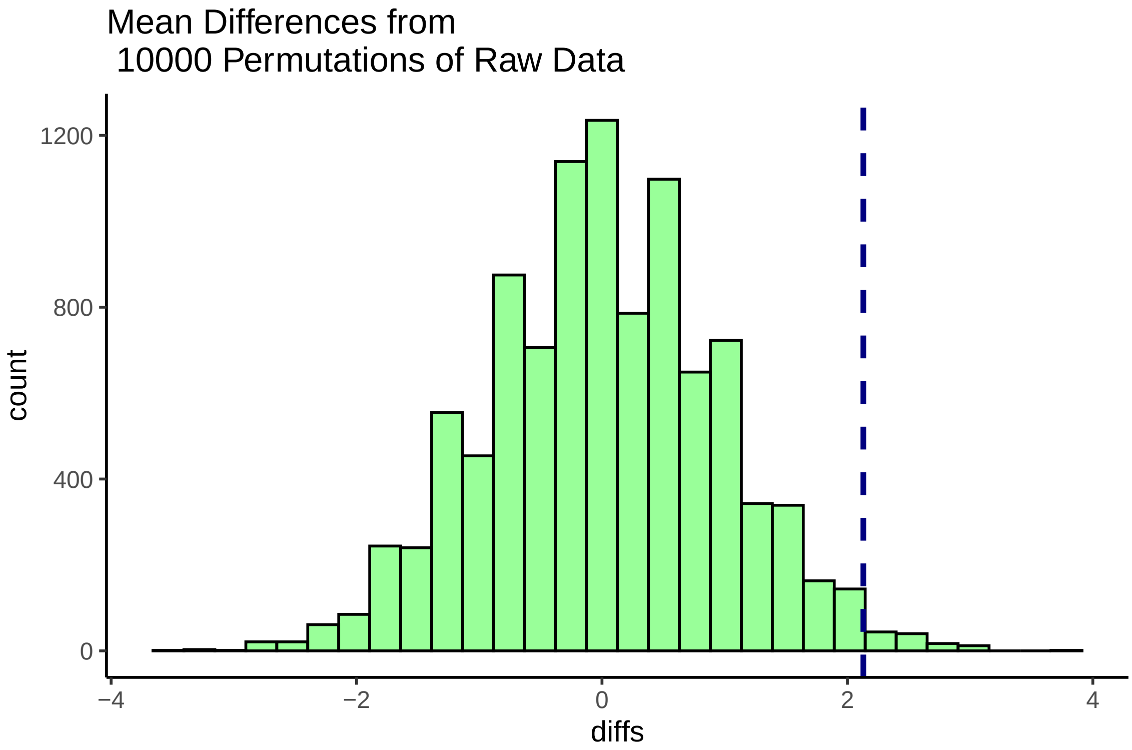
Let's do this same process 10,000 times! Don't worry too much about the details of the code. What we are doing is the above process, just putting it in a loop and asking it to do it 10,000 times. We save all the results in an object called `results`.

```
results<-vector('list',10000)
for(i in 1:10000){
  x <- split(sample(all.levels), rep(1:2, c(n,m)))
  results[[i]]<-mean(x[[1]]) - mean(x[[2]])
}

head(unlist(results)) # these are all our mean differences from 10,000 shuffles of the data. We're just
```

```
## [1] -1.9824561  1.4260652  0.8245614  0.6240602  0.4235589 -0.9799499
```

We can actually make a histogram showing the distribution of these differences in sample means.

```
df <- data.frame(diffs = unlist(results))

ggplot(df, aes(x=diffs)) +
  geom_histogram(color="black", fill="green", alpha=.4) +
  geom_vline(color="navy",lwd=1,lty=2,xintercept = 2.13) +
  theme_classic()+
  ggtitle("Mean Differences from \n 10000 Permutations of Raw Data")
```



This histogram shows that for some of our 10,000 shuffles, we actually got some differences higher than 2.13 (the dotted blue line), but the vast majority of shuffles led to samples that had mean differences lower than 2.13. In fact, only several shuffles led to samples where the sample of size 21 (drug in the original data) had a sample mean that was higher than the sample of size 19 (placebo in the original data).

We can directly calculate how many times out of 10,000 shuffles we got a difference in sample means that was greater than 2.13

```
sum(unlist(results) >= 2.13)  # 114 times out of 10000
```

```
## [1] 114
```

To convert this to a p-value, we simply divide this value by the number of shuffles we ran - which was 10,000.

```
p1s = sum(unlist(results) >= 2.13) /10000  # which is 0.0202 proportion of the time
p1s
```

```
## [1] 0.0114
```

So, our p-value is $p = 0.0114$ which is the one-sided p-value. If we wished to have a 2-tailed p-value we would simply multiply this value by 2:

```
# 2-tailed value
2 * p1s
```

```
## [1] 0.0228
```

Example 2 (Two Independent Samples from the Same Population)

Let's take a look at another example from clinical trials. Here, we have various subjects' ratings their anxiety levels. They do this after either taking a new anxiolytic drug or an existing drug (there is some prior evidence that new medicine has similar effect as the old, but cheaper to produce). The subjects in each group are independent of each other. The drug group has 19 subjects and the drug group has 21 subjects.

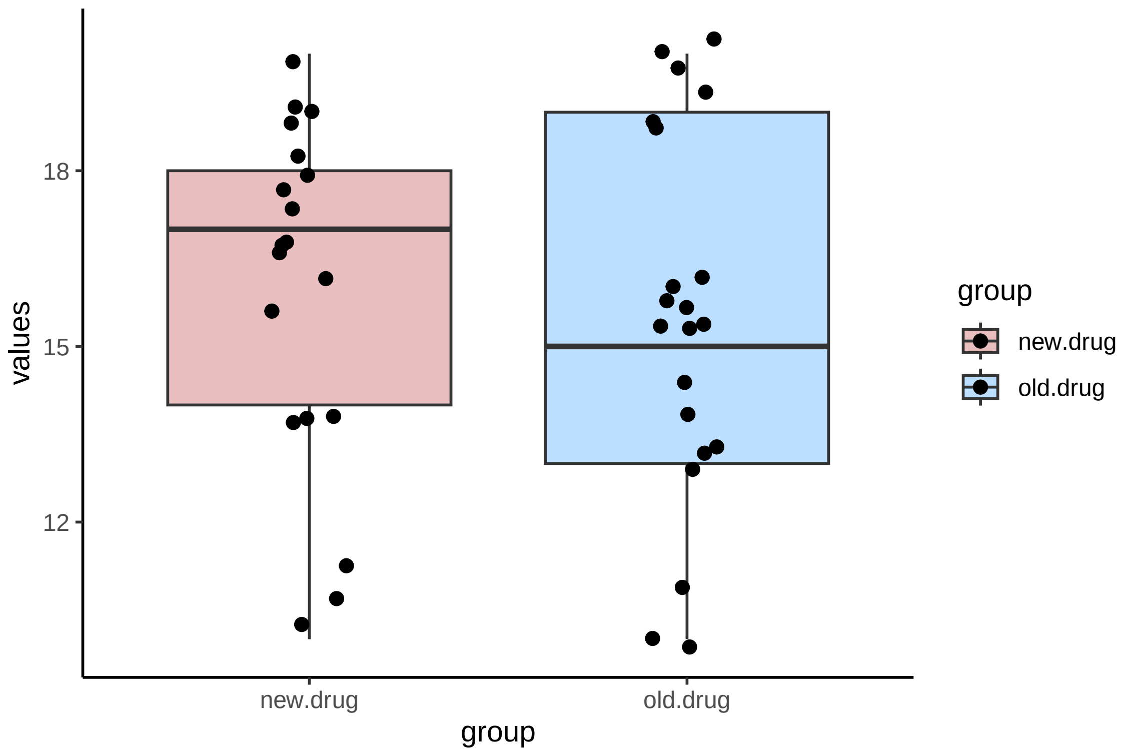
The data:

```r
new.drug <- c(18, 16, 11, 19, 14, 19, 17, 17, 19, 17, 10, 14, 16, 11, 18, 17, 14, 18, 20)
old.drug <- c(13, 10, 19, 16, 19, 20, 13, 14, 20, 16, 10, 15, 13, 16, 19, 14, 15, 15, 20, 16, 11)

n=length(new.drug) #19
m=length(old.drug)
# put into a dataframe:
dd <- data.frame(values = c(new.drug, old.drug),
                 group = c(rep("new.drug",n), rep("old.drug", m))
)

head(dd)
```

```
##   values    group
## 1     18 new.drug
## 2     16 new.drug
## 3     11 new.drug
## 4     19 new.drug
## 5     14 new.drug
## 6     19 new.drug
```

The boxplots for the two groups:

```r
ggplot(dd, aes(x = group, y = values, fill = group)) +
  geom_boxplot(alpha=.3, outlier.shape = NA) +
  geom_jitter(width=.1, size=2) +
  theme_classic() +
  scale_fill_manual(values = c("firebrick", "dodgerblue"))
```



Now, we compute the mean differences between the two samples:

```
mean.diff <- mean(new.drug)  - mean(old.drug)   # 0.624
mean.diff
```

```
## [1] 0.6240602
```

From our samples, we observed a difference in grades of 0.624 between the groups. Below, we run an independent *t*-test to test whether these two samples came from same or different populations:

```
t.test(new.drug, old.drug) #can add var.equal=T option here
```

```
## 
##  Welch Two Sample t-test
## 
## data:  new.drug and old.drug
## t = 0.64279, df = 37.974, p-value = 0.5242
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.341404  2.589524
## sample estimates:
## mean of x mean of y 
##  16.05263  15.42857
```

This Student's *t*-test suggests that there is no (significant) difference, meaning that the groups don't differ in their population means.

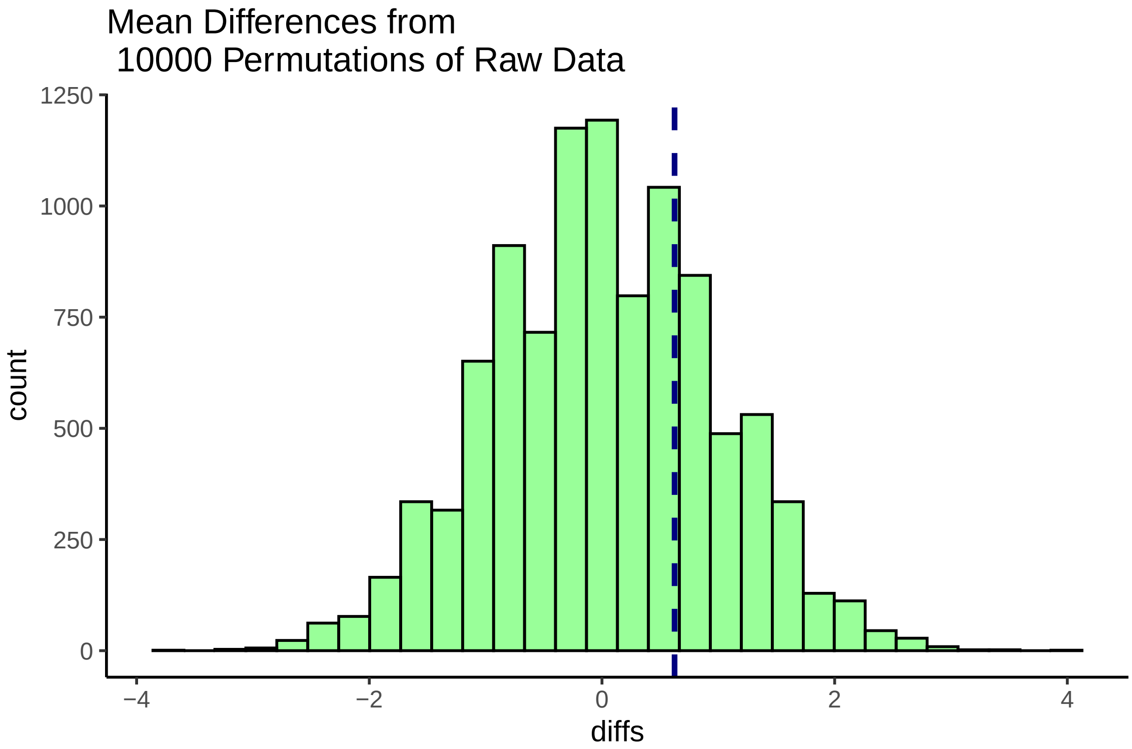
Now, let's apply the permutation test as above.

```r
results<-vector('list',10000)
for(i in 1:10000){
  x <- split(sample(all.levels), rep(1:2, c(n,m)))
  results[[i]]<-mean(x[[1]]) - mean(x[[2]])
}

head(unlist(results)) # these are all our mean differences from 10,000 shuffles of the data. We're just
```

```
## [1] -0.07769424  0.02255639 -0.17794486  0.12280702  1.72681704 -1.98245614
```

The histogram of the mean differences:

```r
df <- data.frame(diffs = unlist(results))

ggplot(df, aes(x=diffs)) +
  geom_histogram(color="black", fill="green", alpha=.4) +
  geom_vline(color="navy",lwd=1,lty=2,xintercept = 0.624) +
  theme_classic()+
  ggtitle("Mean Differences from \n 10000 Permutations of Raw Data")
```



This histogram shows that 0.624 is not that different than most of the shuffled sample differences (the observed difference in the original samples is close to center).

We can compute a one-sided p-value as

```r
p1s=sum(unlist(results) > 0.624) /10000  # which is 0.0202 proportion of the time
p1s
```

```
## [1] 0.2841
```

```
# 2-tailed value
2 * p1s
```

```
## [1] 0.5682
```

So, one-sided p-value is $p = 0.2817$ and two-sided p-value is $p = 0.5634>$

Lecture 14

Kolmogorov-Smirnov & Permutation Tests

Kolmogorov-Smirnov (KS) Test

- Used to test
  (i) whether two data sets are from the same distribution or not
  (ii) whether a data set is from a specified distribution or not.

One-Sample KS Test

$D = \{X_1, X_2, \ldots, X_n\}$ $X_i \overset{iid}{\sim} F_X$, no further assumption!

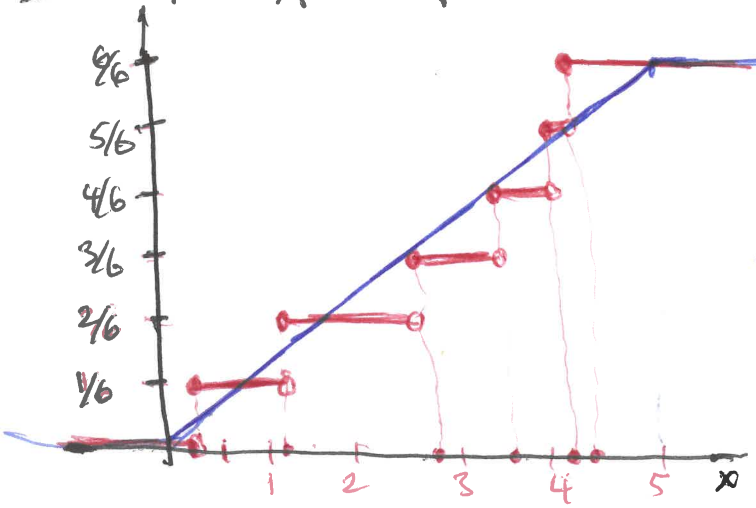
Qu: (1) Is $F_X \equiv \text{Exp}(1)$?

(2) Is $F_X \equiv N(0,1)$? etc.

Ex: $D = \{1.1, 4.3, 0.2, 3.6, 2.9, 4.2\}$, $X_i \overset{iid}{\sim} F_X$

Qu: Is $F_X \equiv \text{Uniform}(0,5)$?

Sol'n: $H_0: F_X \equiv \text{Uniform}(0,5)$ vs $H_a: F_X \not\equiv \text{Uniform}(0,5)$



$\leftarrow \hat{F}_X$: ecdf

$$\longleftarrow F_X = \begin{cases} 0 & x < 0 \\ x/5 & 0 \le x < 5 \\ 1 & x \geq 5 \end{cases} \quad \text{cdf for Unif}(0,5)$$

Test stat:

$$d_{KS} = dist(\hat{F}_X, F_X)$$

$a/5$ here

$$= \max_{a \in (0,5)} |\hat{F}_X(a) - F_X(a)|$$

$a \in \Omega$ or $a \in \mathcal{X}$ in general

Under Ho, $d_{KS} \sim$ KS distribution $\equiv F_{KS}$

not commonly known or covered in classes

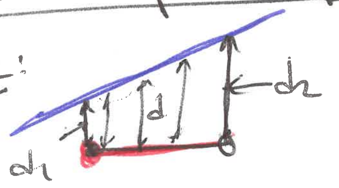
DR: Reject Ho, if

$$d_{KS} > c = KS_\alpha$$

from table or software

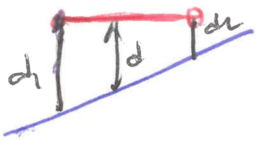
How to compute d_{KS}?

Case 1:



$d \leq \max(d_1, d_2)$

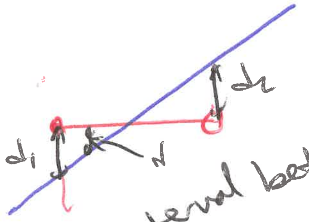
Case 2:



$d \leq \max(d_1, d_2)$

Case 3:

$d \leq \max(d_1, d_2)$



So, at each interval between consecutive (sorted) data values, $x_{(i)} \rightarrow x_{(i+1)}$, max dist occur at the datapoints.

Hence, overall max distance, which is d_{KS}, occurs at the data point values (which is the max distance of distances for the intervals)

Tabulate

a	$F_X(a)$	$\hat{F}_X^-(a)$	$\hat{F}_X^+(a)$	$\lvert \hat{F}_X^-(a) - F_X(a) \rvert$	$\lvert \hat{F}_X^+(a) - F_X(a) \rvert$
0.2	0.2/5	0	1/6	0.04	$\lvert 0.04 - 1/6 \rvert$
1.1	1.1/5	1/6	2/6	⋮	⋮
2.9	2.9/5	2/6	3/6	·	·
3.6	3.6/5	3/6	4/6	·	·
4.2	4.2/5	4/6	5/6	·	·
4.3	4.3/5	5/6	6/6	·	

maximum of these n

$d_{KS} = \max_{a} \lvert \hat{F}_X(a) - F_X(a) \rvert$

KS Test for Two Independent Samples

$D_1 = \{X_1, X_2, \ldots, X_n\}$, $X_i \overset{iid}{\sim} F_X$

$D_2 = \{Y_1, Y_2, \ldots, Y_m\}$, $Y_j \overset{iid}{\sim} F_Y$

$X_i \perp Y_j$ for all i,j

$H_0: F_X \equiv F_Y$ vs $H_a: F_X \not\equiv F_Y$

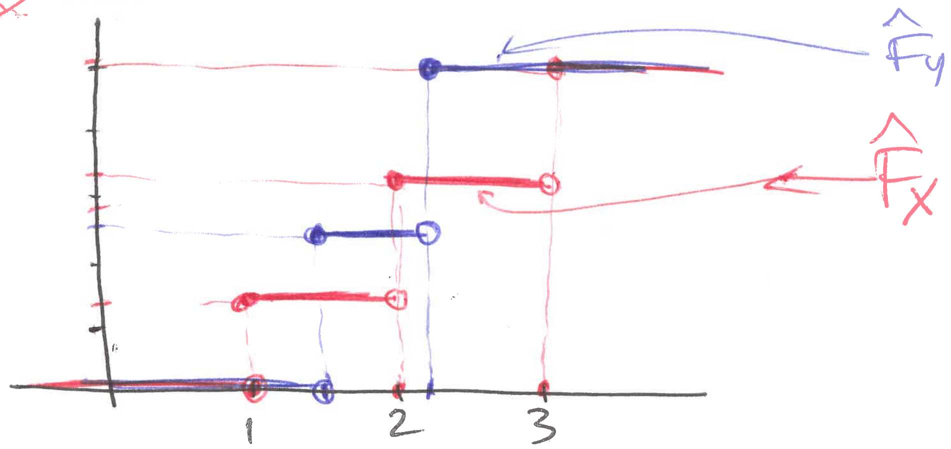
test stat: $d_{KS} = \max\limits_{\substack{a \\ \uparrow \\ a \in D_1 \cup D_2}} \lvert \hat{F}_X(a) - \hat{F}_Y(a) \rvert$

Under H_0, $d_{KS} \sim$ KS distribution

DR: Reject H_0 if $d_{KS} > KS_\alpha$
↑ from table or software.

Ex: $D_1 = \{3, 1, 2\}$ $D_2 = \{1.5, 2.1\}$ ← iid from F_Y

iid from F_X



Pick one data set as reference & tabulate for values in that data set

a	$\hat{F}_1(a)$	$\hat{F}_2^-(a)$	$\hat{F}_2^+(a)$	$\lvert\hat{F}_1(a) - \hat{F}_2^-(a)\rvert$	$\lvert\hat{F}_1(a) - \hat{F}_2^+(a)\rvert$
1	.	.	.	.	.
2	.	.	.	.	.
3	.	.	.	.	.

find max of these, say D_1

Do the same for the other data set, i.e.,

a	$\hat{F}_2(a)$	$\hat{F}_1^-(a)$	$\hat{F}_1^+(a)$	$\lvert\hat{F}_2(a) - \hat{F}_1^-(a)\rvert$	$\lvert\hat{F}_2(a) - \hat{F}_1^+(a)\rvert$
1.5	.	.	.	.	.
2.1	.	.	.	.	.

find max of these as well, say D_2

$$\Rightarrow \boxed{d_{KS} = \max(D_1, D_2)}$$

Permutation Test

Ubiquous method, works for any test stat for the 2 (or more) independent sample setting.

Setting

$D_1 = \{X_1, X_2, \ldots, X_n\}$, $X_i \overset{iid}{\sim} F_X$ $\quad X_i \perp Y_j$ for all i, j

$D_2 = \{Y_1, Y_2, \ldots, Y_m\}$, $Y_j \overset{iid}{\sim} F_Y$

- No further assumptions

$$H_0: F_X \equiv F_Y \quad \text{vs} \quad H_a: F_X \not\equiv F_Y$$

So, under Ho, each sample is a random sample from the same distribution (i.e. same population)

Thus (1) Both data sets yield similar estimates (hence similar test)

compute a test stat $T_0 = T(D_1, D_2)$

(2) One can view $D = D_1 \cup D_2$ a random sample from F $(\equiv F_X \equiv F_Y)$

(3) If we randomly a sample of size n from D, say D_1^*, that will have the same distr. as D_1, and the remaining in D, say D_2^*, will have " " " D_2

Compute $T_1^* = T(D_1^*, D_2^*)$

$T_1^* \overset{d}{=} T_0$ (so their values would be similar)

The exact distr of T_1^* can be obtained by considering all possible permutations (there are $\binom{N}{n}$ such permutations w/ $N = n+m$)

Since T_0 & $T_1^*, T_2^*, \ldots, T_{\binom{N}{n}}^*$ have the same distribution, T_0 will not be in the extremes of the distr. of T_i^* but rather closer to the center or median.

But if $F_X \not\equiv F_Y$, then, T_0's distr. would be (very) differt from T_i^*'s distr, so T_0 would be in the extremes of T_i^*'s distr.

Hence, a reasonable estimate of p-value would be

$$\text{p-value} = 2\min\left(\frac{\sum_{i=1}^{N!} I(T_i^* \le T_0)}{\binom{N}{n}}, \frac{\sum_{i=1}^{N!} I(T_i^* \ge T_0)}{\binom{N}{n}}\right)$$

DR: Reject H_0, if p-value $< \alpha$

<u>Note</u>: In practice, $\binom{N}{n}$ grows very quickly, so we repeat step (3) M (large but $<< \binom{N}{n}$) many times, and replace $\binom{N}{n}$ with M.

See the permutation test example in R.

<u>Note</u>: Permutation tests (or randomization tests) can be adapted to any HT testing setting with multiple (2 or more) samples for which you can compute the test statistic.