Non parametric tests

Are the Panini cards packaged at random?

We have a suspicion that the Panini cards are not pacaged completely at random becausewe tend to get a lot of duplicates, especially of some card types. We'll now use a simulation test to test our hypothesis.

The null hypothesis is that the cards are packaged at random with replacement. The alternative is the following: the cards are packaged at random with replacement, but k = 100 of the card types are 5 times more common than the others.

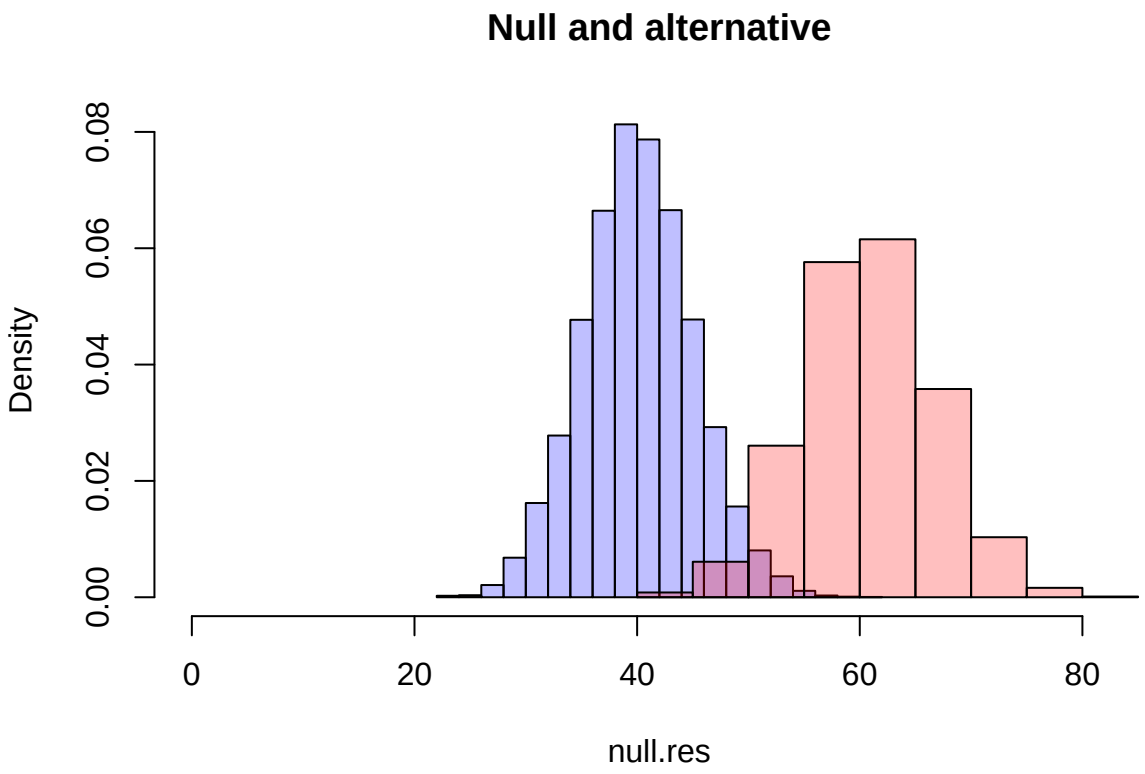
```
npacks <- 50
ncards <- 682
nsim <- 10000
```

```
sim.from.null <- function(npacks=50){
  sampled <- sample(1:ncards, size=5*npacks, replace=T)
  # count the number of duplicates
  dupl <- 5*npacks - length(unique(sampled))
  return(dupl)
}
alt.cards <- c(101:682,rep(1:100,5))
sim.from.alt <- function(npacks=50){
  sampled <- sample(alt.cards, size=5*npacks, replace=T)
  dupl <- 5*npacks - length(unique(sampled))
  return(dupl)
}
```

Let's simulate from both hypothesis and look at what we get.

```
null.res <- replicate(nsim, sim.from.null())
alt.res <- replicate(nsim, sim.from.alt())
```

```
p1 <-hist(null.res, plot = F)
p2 <- hist(alt.res, plot=F)
plot(p1, col=rgb(0,0,1,1/4), xlim=c(0,max(null.res,alt.res)), main="Null and alternative", freq = F)
plot(p2, col=rgb(1,0,0,1/4), add=T, freq=F)
```



As it is possible to see from the plot above the test has a high power, meaning that the distributions are easily distinguishable. But let's put some numbers into that claim.

```
# power of a test = probability of the rejection region of the null , given that the alternative is tru
# To get the power we need 2 things:
# the rejection region for the level alpha=0.05
# the density of the alternative - which we have computed by simulation above

# note that our alternative claims to have more duplicates than the null, hence the rejection region is
rej.reg <- quantile(null.res, probs = 0.95) + 1 #we subtract one because of the discretization of the d
rej.reg
```

```
## 95%
##  50
```

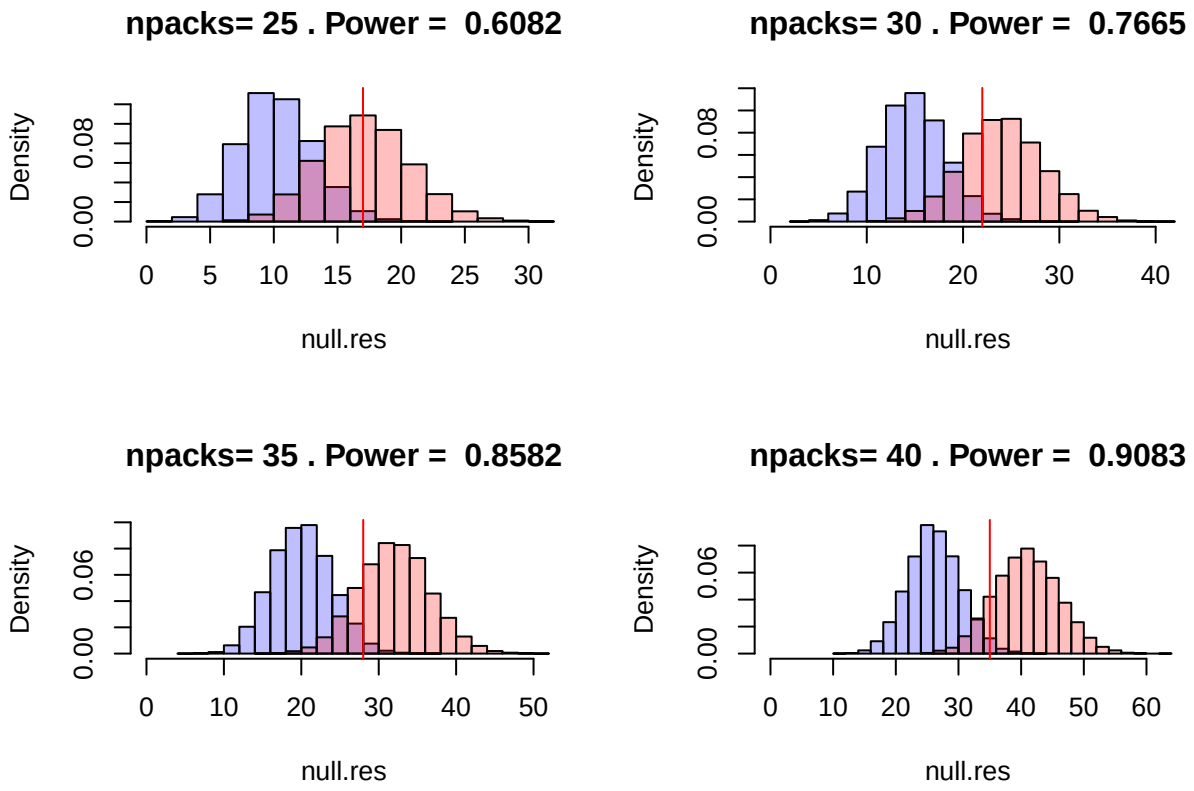
```
# let's now count how many of the results under the alternative fall under this threshold
power <- sum(alt.res >= rej.reg)/nsim
power
```

```
## [1] 0.9777
```

Our calculations confirm our intuition: the test has high power. Let's try to change the number of packs opened and see how this affects the power.

```
npacks <- c(25,30,35,40)
powers <- rep(0,4)
par(mfrow=c(2,2))
for(i in 1:4){
  # simulating
```

```
  np <- npacks[i]
  null.res <- replicate(nsim, sim.from.null(np))
  alt.res <- replicate(nsim, sim.from.alt(np))
  # power
  rej.reg <- quantile(null.res, probs = 0.95) + 1
  power <- sum(alt.res >= rej.reg)/nsim
  powers[i] <- power
  # plotting
  p1 <-hist(null.res, plot = F)
  p2 <- hist(alt.res, plot=F)
  plot(p1, col=rgb(0,0,1,1/4), xlim=c(0,max(null.res,alt.res)), main=paste("npacks=",np,". Power = ",pow
  plot(p2, col=rgb(1,0,0,1/4), add=T, freq = F)
  abline(v=rej.reg, col="red")
}
```



We can conclude that to obtain power >= 85% from our test we need to open at least 35 packs.

Permutation tests

A good example of a permutation test is the Wilcoxon signed rank test. It is an un-paired 2 sample test that checks whether the two samples come from the same distribution. We'll generate some data to analyse the performance of the method on it.

```
#install.packages("rmutil")
require("rmutil")
```

```
## Loading required package: rmutil
```

```
## Warning: package 'rmutil' was built under R version 3.6.3
```

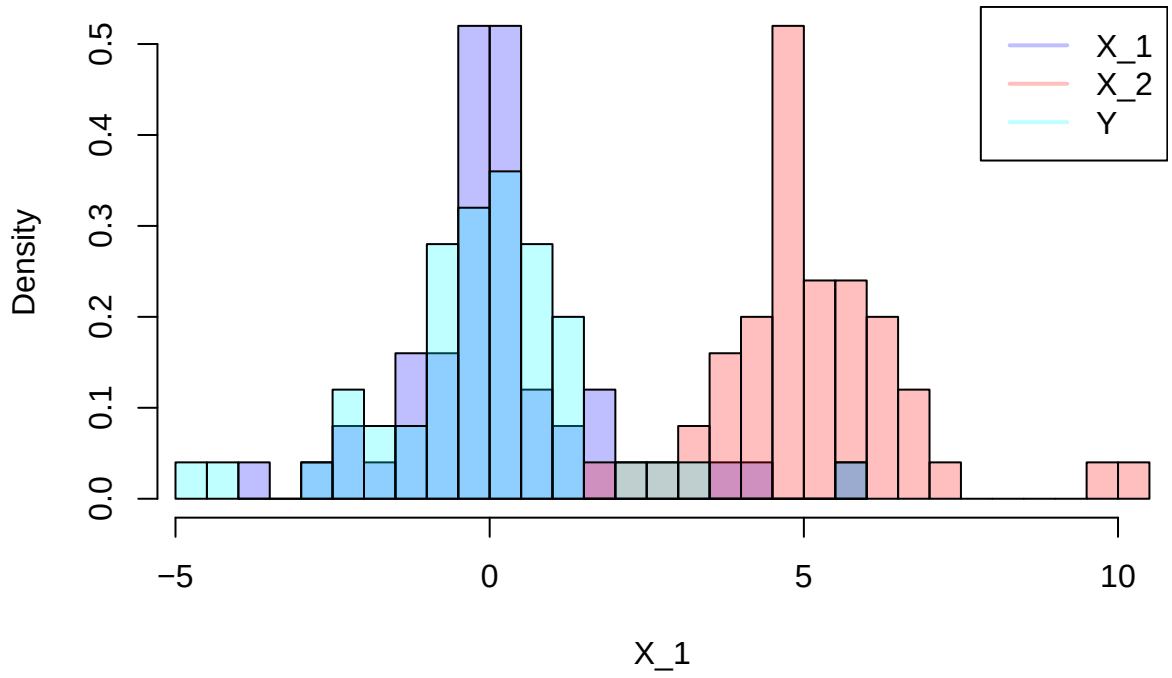
```
##
## Attaching package: 'rmutil'
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following objects are masked from 'package:base':
##
##     as.data.frame, units
```

```
m <- 50
# in this case the samples come from the same population
X_1 <- rlaplace(n = m, m = 0, s = 1)
Y <- rlaplace(n = m, m = 0, s = 1)
# in this case they come from two different populations, each with a different mean
X_2 <- rlaplace(n=m, m=5, s=1)
```

```
p1 <- hist(X_1, breaks=20,plot=F)
p2 <- hist(X_2, breaks=20, plot=F)
p3 <- hist(Y, breaks=20, plot=F)
plot(p1, col=rgb(0,0,1,1/4), xlim=c(min(X_1,X_2,Y),max(X_1,X_2,Y)), freq = F, main="")
plot(p2, col=rgb(1,0,0,1/4), add=T, freq = F)
plot(p3, col=rgb(0,1,1,1/4), add=T, freq = F)
legend("topright", col=c(rgb(0,0,1,1/4),rgb(1,0,0,1/4),rgb(0,1,1,1/4)), legend=c("X_1","X_2","Y"), lty =
```



Okay by eye it's easy to spot the difference in the two populations. Bt let's try to use the WIlcoxon test to

assess it.

```
#Wilcoxon statistic: the sum of the ranks
# The rank statistic is very robust since it does not depend on the original distribution of the data
get.sum.rank <- function(n1, samples){
  # assuming the first n1 samples come from the first population
  ranks <- rank(samples)
  return(sum(ranks[1:n1])-sum(ranks[(n1+1):length(samples)]))
}
do1perm <- function(n1, data){
  # permute
  new.data <- sample(data, length(data), replace = F)
  # compute statistic
  rank.sum <- get.sum.rank(n1, new.data)
  return(rank.sum)
}
```

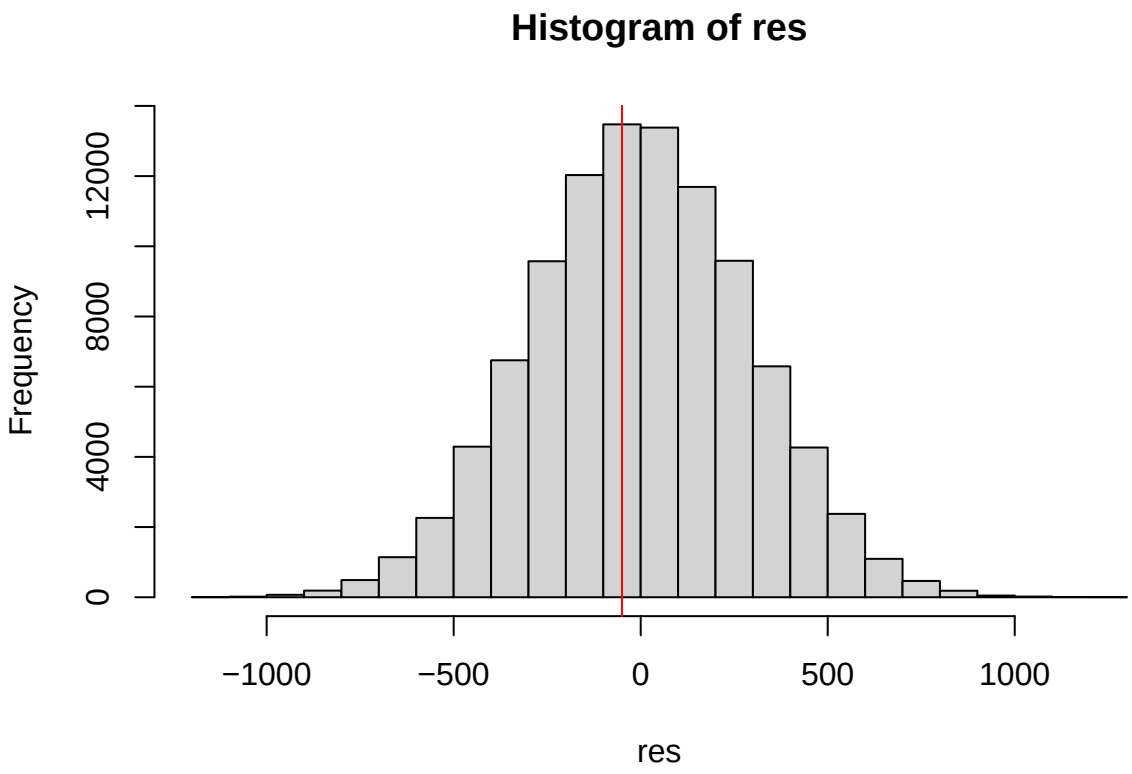
We'll first test X_1 and Y:

```
nperm <- 100000
data <- c(X_1,Y)
n1 <- length(X_1)
res <- replicate(nperm, do1perm(n1, data))
```

Note that distribution is approximately normal and centered around 0. Let's now compute the p-value associated to our original data.

```
original.stat <- get.sum.rank(n1, data)
```

```
hist(res, col="lightgray")
abline(v=original.stat, col="red")
```




```
# Note that our original data is positive, hence our alternative hypothesis is whether the population X
# Thus the test will be on the right tail of the statistic permutation distribution
p.value <- (sum(res>=original.stat)+1)/(nperm + 1)
p.value
```

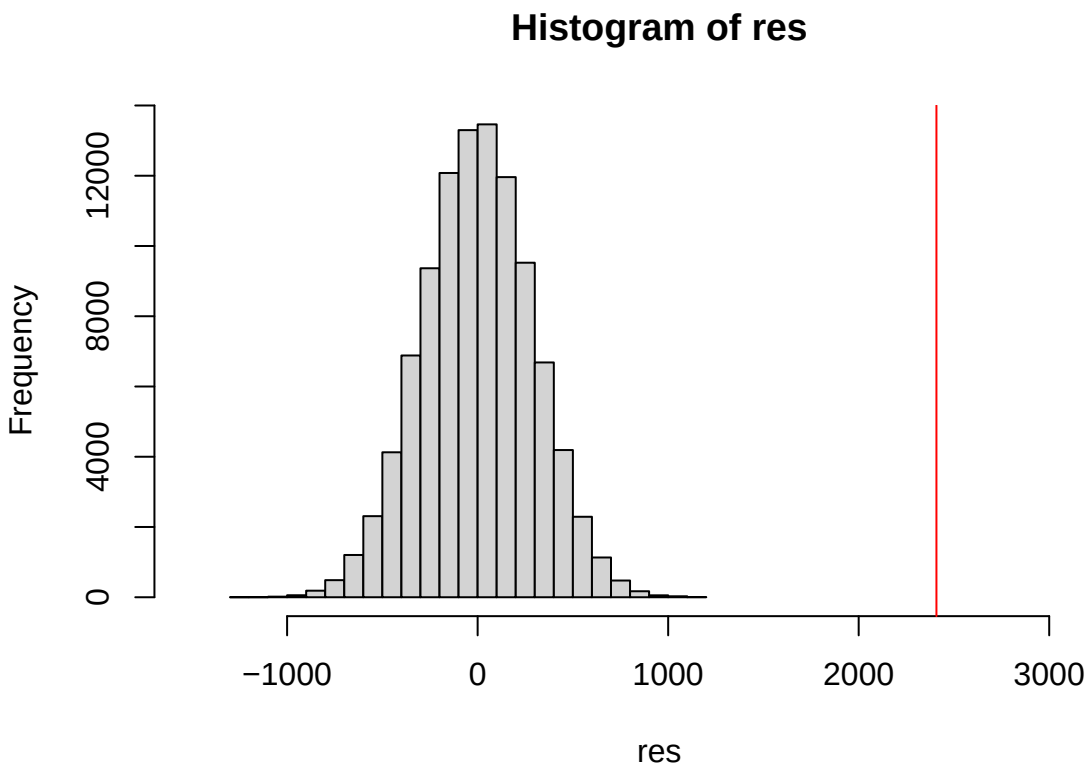
```
## [1] 0.5676643
```

We hence fail to reject the null at level 0.05. Let's now repeat the same test for the two populations Y and X_2, for which the alternative is true.

```
nperm <- 100000
data <- c(X_2,Y)
n1 <- length(X_2)
res <- replicate(nperm, do1perm(n1, data))
```

```
original.stat <- get.sum.rank(n1, data)
```

```
hist(res, col="lightgray", breaks=20, xlim=c(-1500,original.stat+1000))
abline(v=original.stat, col="red")
```



In this case the original statistic is clearly off the permutation distribution, meaning that it can hardly be due to chance in the null hypothesis context.

```
p.value <- (sum(res>=original.stat)+1)/(nperm + 1)
p.value
```

```
## [1] 9.9999e-06
```

In this case we reject the null at level 0.05.

Now that we have assessed the functioning of the permutation test we'll use it to work with some real-world data. Specifically, we're going to use the dataset *immer*, with its two columns Y1 and Y2. They measure the yield in the year 1931 and 1932, respectively. We omit the information that each field / observation was assigned to one of the six different locations and that one of the five different varieties of barley was grown. The farmer suspects that the yield was significantly less in the second year, regardless of the location and the type of barley. Let's test it!

```
require("MASS")
```

```
## Loading required package: MASS
```

```
?immer
```

```
## starting httpd help server ...
```

```
##  done
```

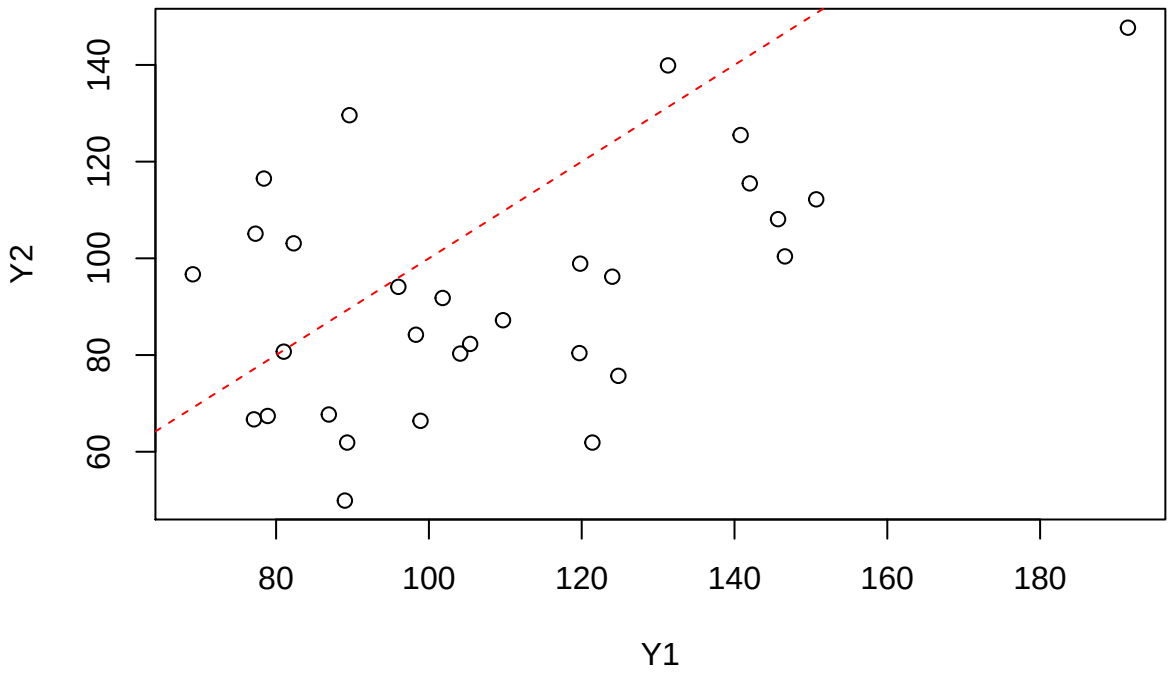
```
head(immer)
```

```
##   Loc Var    Y1    Y2
## 1  UF   M  81.0  80.7
```

```
## 2  UF   S 105.4  82.3
## 3  UF   V 119.7  80.4
## 4  UF   T 109.7  87.2
## 5  UF   P  98.3  84.2
## 6   W   M 146.6 100.4
```

```
attach(immer)
```

```
plot(Y1,Y2)
abline(a=0,b=1, col="red", lty="dashed")
```



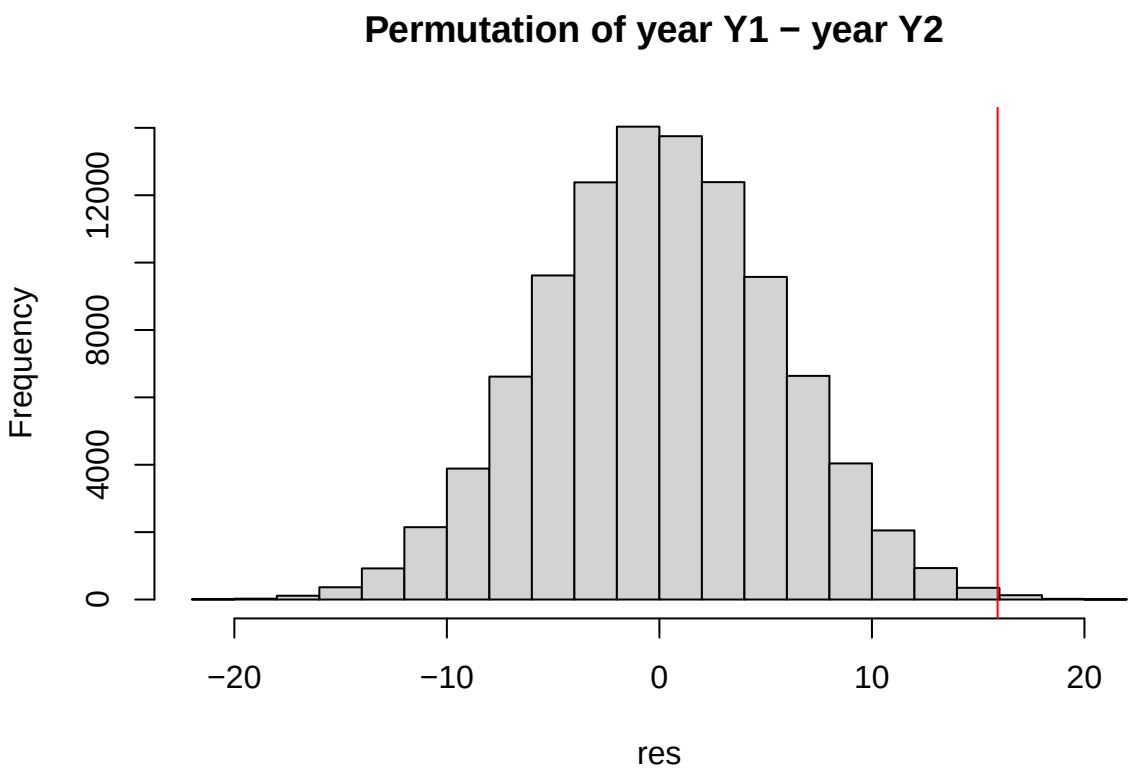
Many of the points seem to be below the line, hence for the majority of the samples the yield on year Y1 appears to be higher than the yield on the year Y2. But let's put some numbers to this impression.

```
#Note that in this case the test is a paired 2 sample test. Hence we're not permuting the values from t
diff <- Y1 - Y2
#Null hypothesis: no difference between Y1 and Y2
#Alternative hypothesis: Y1 is shifted to the right with respect to Y2, hence the difference vector is

#This time we'll use the mean of the difference vector as a statistic
do.step <- function(diff){
  n <- length(diff)
  signs <- sample(c(1,-1), size=n, replace=T)
  return(mean(diff*signs))
}
```

```
nperm <- 100000
res <- replicate(nperm, do.step(diff))
```

```r
original.stat <- mean(diff)
hist(res, col="lightgray", breaks=20, main="Permutation of year Y1 - year Y2 ")
abline(v=original.stat, col="red")
```



The farmer impression seems to have statistical significance, meaning that the difference in the two harvests seem to be highly unlikely. To know how much unlikely exactly, we need a p-value:

```r
p.value <- (sum(res>=original.stat)+1)/(nperm + 1)
p.value
```

```
## [1] 0.001679983
```

We reject the null at level 0.05 with this test. Let's compare this result with what we would have got with the Wilcoxon test.

```r
wilcox.test(Y1,Y2, alternative ="greater", paired=T)
```

```
## Warning in wilcox.test.default(Y1, Y2, alternative = "greater", paired = T):
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  Y1 and Y2
## V = 368.5, p-value = 0.002659
## alternative hypothesis: true location shift is greater than 0
```

Again with the Wilcox test we reject the null hypothesis at level 0.05.

Permutation tests for regression

We'll here use permutation tests in a different setting than paired tests, i.e. the regression setting. By permuting the Y values we break the relationship between the labels and the predictors. Hence, looking at the permutation distribution of the statistic assessing the regression model we can evaluate the amount by which our original statistic value can be due to chance. Let's do it.

```
data <- read.csv("C:/Users/Giulia/Downloads/data_ex3.csv")
head(data)
```

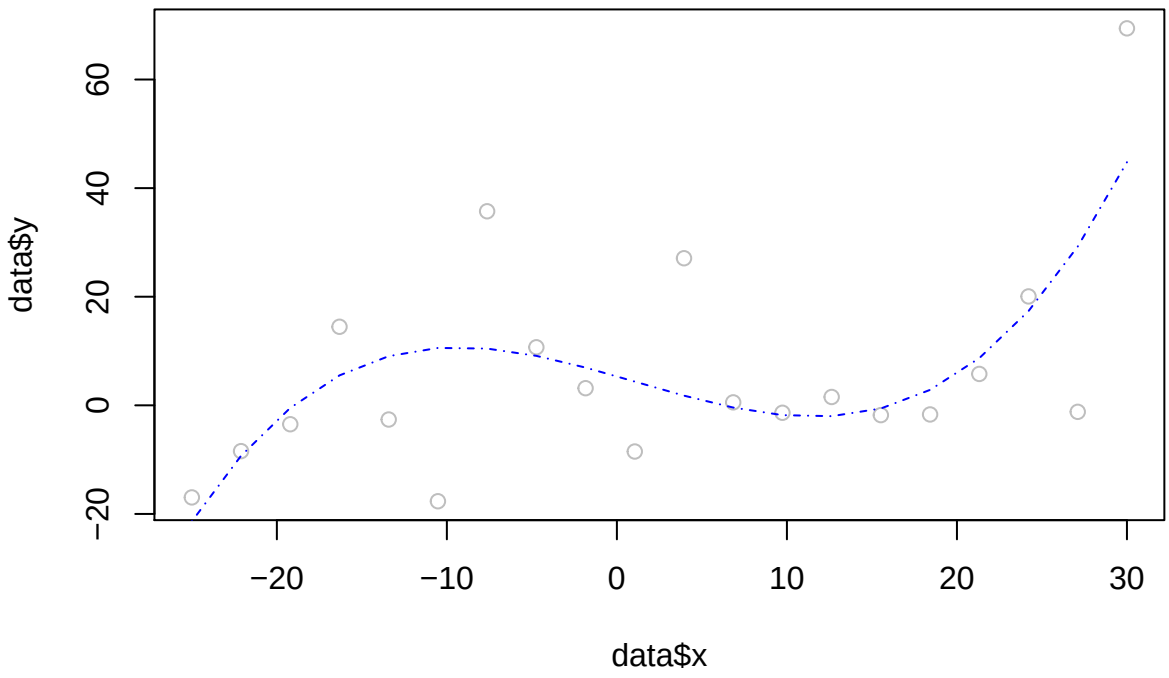
```
##            y         x
## 1 -16.955577 -25.00000
## 2  -8.436601 -22.10526
## 3  -3.482003 -19.21053
## 4  14.465474 -16.31579
## 5  -2.611938 -13.42105
## 6 -17.662856 -10.52632
```

We believe this data comes from a degree three polynomial, so we fit a polynomial regression model to it.

```
poly.fit <- lm(y~I(poly(x,3)), data=data)
summary(poly.fit)
```

```
##
## Call:
## lm(formula = y ~ I(poly(x, 3)), data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.4469  -3.9743   0.6474   3.7564  25.3044
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.239      3.576   1.745   0.1002
## I(poly(x, 3))1   36.558     15.991   2.286   0.0362 *
## I(poly(x, 3))2   12.693     15.991   0.794   0.4389
## I(poly(x, 3))3   44.794     15.991   2.801   0.0128 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.99 on 16 degrees of freedom
## Multiple R-squared:  0.4613, Adjusted R-squared:  0.3603
## F-statistic: 4.568 on 3 and 16 DF,  p-value: 0.01706
```

```
plot(data$x,data$y, col="gray")
lines(data$x, poly.fit$fitted.values, col="blue", lty=10)
```

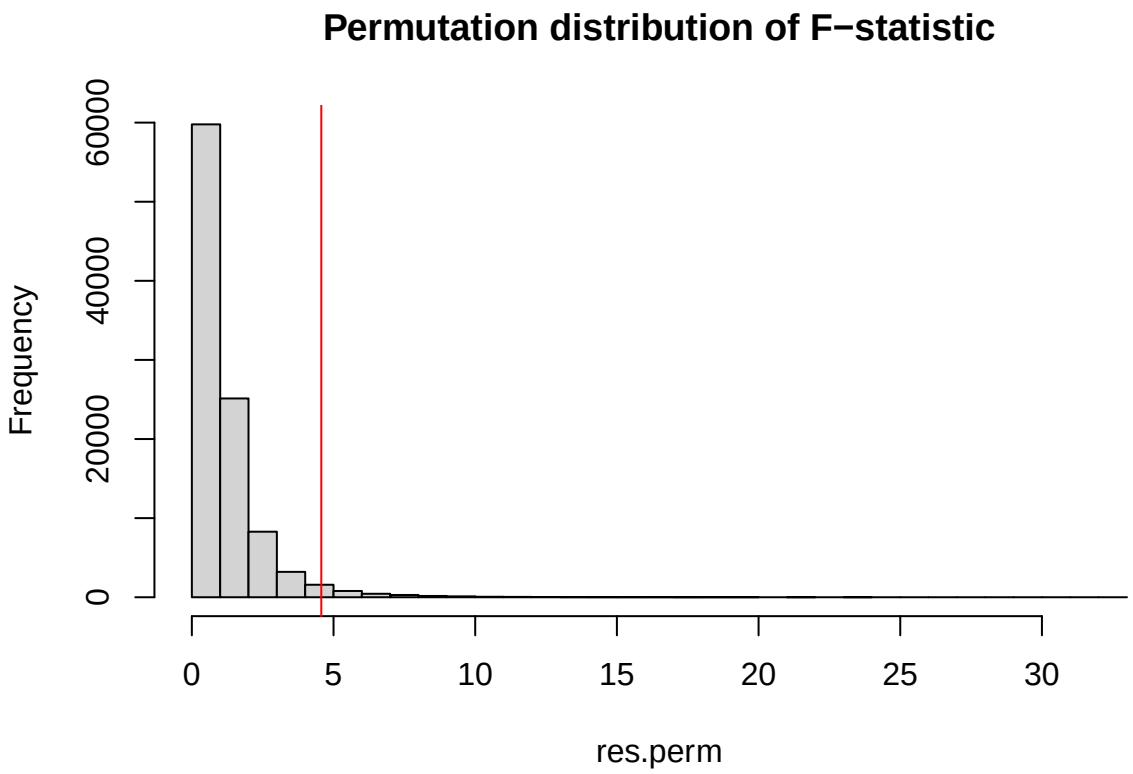


From the t-test the first and third order terms seem to have statistical significance. Also, from the p-value of the global F-test rejects the global null at level 0.05. However, let's use a permutation test first to check for the global null.

```
do.1.perm.F <- function(x,y){
  y.new <- sample(y, size=length(y),replace=F)
  poly.fit <- lm(y.new~I(poly(x,3)))
  s <- summary(poly.fit)
  return(s$fstatistic["value"])
}
```

```
s <- summary(poly.fit)
original.stat <- s$fstatistic["value"]
nperm <- 100000
res.perm <- replicate(nperm, do.1.perm.F(data$x, data$y))
```

```
hist(res.perm, col="lightgray", breaks=40, main="Permutation distribution of F-statistic")
abline(v=original.stat, col="red")
```




```
# Here being the distribution symmetric the
p.value <- (sum(res.perm>=original.stat)+sum(res.perm<=-1*original.stat)+1)/(nperm + 1)
p.value
```

```
## [1] 0.02543975
```

The permutation test result suggests to reject the null at level 0.05.