ANOVA for Randomized Block Design

Estimating which screen sizes have significantly different average prices, with RAM size as the blocking factor

Pranav Gopalkrishna, 1940223

2021-09-08

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Setting work directory

setwd("~/Documents/Study/computerScience/programming/r/data/")

# Data set

This dataset is for basic data analysis, and contains basic information about various different computer models, including RAM size, screen width and price.

I want to find out how a computer’s screen size can affect the computer’s price on average.

I pick RAM size as the blocking factor since it is generally the most significant factor in determining the price of the computer. Hence, we would expect there to be substantial mean price differences between groups of computers having different RAM sizes, implying relatively more homogeneity in prices within each group (this is visually confirmed in the box plots below).

Hence,  
**treatment (*tr*) = screen  
blocking factor (*bf*) = ram  
*y* = price**

myData = read.csv("computersBasics.csv")[c(2, 5, 6)]  
head(myData)

## price ram screen  
## 1 1499 4 14  
## 2 1795 2 14  
## 3 1595 4 15  
## 4 1849 8 14  
## 5 3295 16 14  
## 6 3695 16 14

## Assigning variables

y = myData$price  
bf = as.factor(myData$ram) # Blocking factor  
tr = as.factor(myData$screen) # Treatment  
myData = data.frame(tr, bf, y)

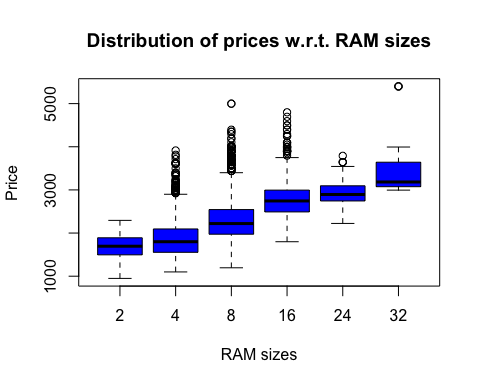
## Summary of data

summary(myData)

## tr bf y   
## 14:3661 2 : 394 Min. : 949   
## 15:1992 4 :2236 1st Qu.:1794   
## 17: 606 8 :2320 Median :2144   
## 16: 996 Mean :2220   
## 24: 297 3rd Qu.:2595   
## 32: 16 Max. :5399

## Visualizing distribution of prices w.r.t. blocks i.e. RAM sizes…

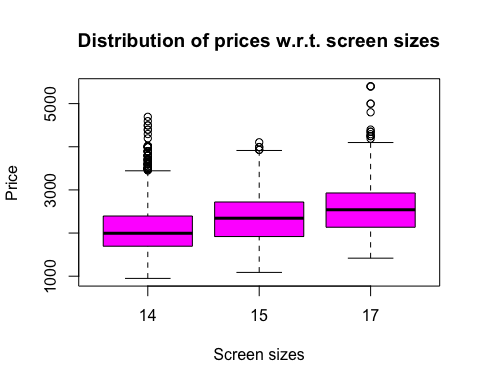
boxplot(y~bf, main = "Distribution of prices w.r.t. RAM sizes", xlab = "RAM sizes", ylab = "Price", col = "blue")



Here we see that each RAM has quite a visually distinct range of prices from other RAM sizes. This indicates homogeneity within blocks, and heterogeneity between blocks.

## Visualizing distribution of prices w.r.t. treatment i.e. screen sizes…

boxplot(y~tr, main = "Distribution of prices w.r.t. screen sizes", xlab = "Screen sizes", ylab = "Price", col = "magenta")



Here, we see that there is some distinctness of price levels between each screen size, although some differences may be statistically insignificant.

# ANOVA

First, we perform the test to see whether or not the treatment levels and the blocking factors have at least some significantly different means or not.

## Hypotheses

**For treatment levels**

H0: The means for all treatment levels are equal.

H1: The means of at least two treatment levels are unequal.

**For blocks**

H0: The means for all blocks are equal.

H1: The means of at least two blocks are unequal.

***Our level of significance will be 0.05.***

## Analysis using functions

Multiple linear regression model for two regressors (blocking factor and treatment...  
model = lm(y~tr + bf)  
# ANOVA analysis on the above model  
anova(model)

## Analysis of Variance Table  
##   
## Response: y  
## Df Sum Sq Mean Sq F value Pr(>F)   
## tr 2 195114561 97557281 525.39 < 2.2e-16 \*\*\*  
## bf 5 755202831 151040566 813.43 < 2.2e-16 \*\*\*  
## Residuals 6251 1160713992 185685   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Here we can see that both the factors are significant, since their p-values are each less than the significance level 0.05.

**Hence, we reject the null hypothesis for both treatment levels and blocks.**

This means at least two treatment levels have significantly different means from each other, and at least two blocks have significantly different means from each other.

# Post hoc analysis

We will use the lsmeans package to find out which pair(s) of means are significantly different, for both treatment levels and blocks. The comparison of means is done using the Tukey test.

# install.packages("lsmeans")  
library(lsmeans)

## Loading required package: emmeans

## Warning: package 'emmeans' was built under R version 3.6.2

## The 'lsmeans' package is now basically a front end for 'emmeans'.  
## Users are encouraged to switch the rest of the way.  
## See help('transition') for more information, including how to  
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.

The lsmeans function from this package computes least-squares means for specified factors or factor combinations in a linear model. Optionally, it makes comparisons or contrasts among them.

## Least square estimates of y averaged for blocks.

I.e. the total least square estimates of y for a particular block is averaged. Here, we have 6 blocks (for 6 RAM sizes) So, for each level of the treatment (screen size), the estimates are added and divided by 6.

## lse\_over\_blocks = lsmeans(model, "tr") Least square estimates of y averaged over levels of the treatment

I.e. the total least square estimates of y for a particular treatment level is averaged. Here, we have 3 treatment levels (for 3 screen sizes) So, for each level block (created by RAM size), the estimates are added and divided by 3.

lse\_over\_treatments = lsmeans(model, "bf")

## Finding the pairs of treatment (i.e. screen size) values with significantly different means...

pairs(lse\_over\_blocks)

## contrast estimate SE df t.ratio p.value  
## 14 - 15 -11.3 12.7 6251 -0.893 0.6448  
## 14 - 17 -345.9 19.2 6251 -18.040 <.0001  
## 15 - 17 -334.5 20.1 6251 -16.648 <.0001  
##   
## Results are averaged over the levels of: bf   
## P value adjustment: tukey method for comparing a family of 3 estimates

Here we see that only the difference in the means of screen sizes 14 and 15 are not significantly different as the associated p-value is greater than 0.05. Every other pair is significantly different, since their p-values are less than 0.0001, hence definitely below 0.05.

## Finding the pairs of blocks (created by RAM size) values with significantly different means...

pairs(lse\_over\_treatments)

## contrast estimate SE df t.ratio p.value  
## 2 - 4 -146 23.6 6251 -6.191 <.0001  
## 2 - 8 -553 23.7 6251 -23.314 <.0001  
## 2 - 16 -1039 26.1 6251 -39.801 <.0001  
## 2 - 24 -1196 33.9 6251 -35.272 <.0001  
## 2 - 32 -1798 110.3 6251 -16.293 <.0001  
## 4 - 8 -407 13.0 6251 -31.375 <.0001  
## 4 - 16 -893 16.9 6251 -52.821 <.0001  
## 4 - 24 -1050 27.4 6251 -38.291 <.0001  
## 4 - 32 -1652 108.5 6251 -15.228 <.0001  
## 8 - 16 -486 16.4 6251 -29.586 <.0001  
## 8 - 24 -643 26.9 6251 -23.863 <.0001  
## 8 - 32 -1245 108.3 6251 -11.493 <.0001  
## 16 - 24 -156 28.6 6251 -5.469 <.0001  
## 16 - 32 -758 108.7 6251 -6.976 <.0001  
## 24 - 32 -602 110.7 6251 -5.439 <.0001  
##   
## Results are averaged over the levels of: tr   
## P value adjustment: tukey method for comparing a family of 6 estimates

As expected (due to distinct ranges for each block i.e. RAM size), each block has a significantly different mean price than any other block, as we see here with every p-value being below 0.0001.