Solution: Spanish Translation A/B Test

```
#libraries needed
require(dplyr)
require(rpart)
require(ggplot2)
```

```
#read data
user = read.csv("Translation_Test/user_table.csv")
test = read.csv("Translation_Test/test_table.csv")

#let's create one data set
length(unique(test$user_id)) == length(test$user_id) # are there dupes?
```

```
## [1] TRUE
```

```
length(unique(user$user_id)) == length(user$user_id) # are there dupes?
```

```
## [1] TRUE
```

```
length(user$user_id) - length(test$user_id) # everyone in one table also in the
other one?
```

```
## [1] -454
```

Looks like the user table is busted and we have some user ids missing. When joining, we have to be careful to do not lose the user ids in the test table, but not in the user table.

```
data = merge(test,user, by = "user_id", all.x = TRUE) # this way we don't lose data
data$date = as.Date(data$date)
summary(data)
```

```
##     user_id             date                source         device
##  Min.   :      1   Min.   :2015-11-30   Ads   :181877   Mobile:201756
##  1st Qu.: 249816   1st Qu.:2015-12-01   Direct: 90834   Web   :251565
##  Median : 500019   Median :2015-12-03   SEO   :180610
##  Mean   : 499938   Mean   :2015-12-02
##  3rd Qu.: 749522   3rd Qu.:2015-12-04
##  Max.   :1000000   Max.   :2015-12-04
##
##  browser_language   ads_channel            browser         conversion
##  EN   : 63137     Bing    : 13689   Android_App:155135   Min.   :0.00000
##  ES   :377547     Facebook: 68425   Chrome     :101929   1st Qu.:0.00000
##  Other: 12637     Google  : 68180   FireFox    : 40766   Median :0.00000
##                   Other   :  4148   IE         : 61715   Mean   :0.04958
##                   Yahoo   : 27435   Iphone_App : 46621   3rd Qu.:0.00000
##                   NA's    :271444   Opera      :  6090   Max.   :1.00000
##                                     Safari     : 41065
##       test          sex              age             country
##  Min.   :0.0000   F   :188382   Min.   :18.00   Mexico   :128484
##  1st Qu.:0.0000   M   :264485   1st Qu.:22.00   Colombia : 54060
##  Median :0.0000   NA's:   454   Median :26.00   Spain    : 51782
##  Mean   :0.4764                 Mean   :27.13   Argentina: 46733
##  3rd Qu.:1.0000                 3rd Qu.:31.00   Peru     : 33666
##  Max.   :1.0000                 Max.   :70.00   (Other)  :138142
##                                 NA's   :454     NA's     :   454
```

First question is: check test results. But even before that, let’s make sure it is true Spain converts much better than the rest of LatAm countries.

```
data_conversion_country =  data %>%
                           group_by(country) %>%
                           summarize( conversion = mean(conversion[test == 0]))
%>%# we check the old version
                           arrange (desc(conversion))

head(data_conversion_country)
```

```
## Source: local data frame [6 x 2]
##
##       country conversion
##        (fctr)      (dbl)
## 1       Spain 0.07971882
## 2          NA 0.07755102
## 3 El Salvador 0.05355404
## 4   Nicaragua 0.05264697
## 5  Costa Rica 0.05225564
## 6    Colombia 0.05208949
```

Yes. Definitely true.

```
#a simple t-test here should work. We have collected ~0.5MM data and test/control
split is ~50/50.
data_test = subset(data, country != "Spain") #nothing changed in Spain, so no poin
t in keeping those users

t.test(data_test$conversion[data_test$test == 1], data_test$conversion[data_test$t
est == 0])
```

```
##
##  Welch Two Sample t-test
##
## data:  data_test$conversion[data_test$test == 1] and data_test$conversion[dat
a_test$test == 0]
## t = -7.3539, df = 385260, p-value = 1.929e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.006181421 -0.003579837
## sample estimates:
##  mean of x  mean of y
## 0.04341116 0.04829179
```

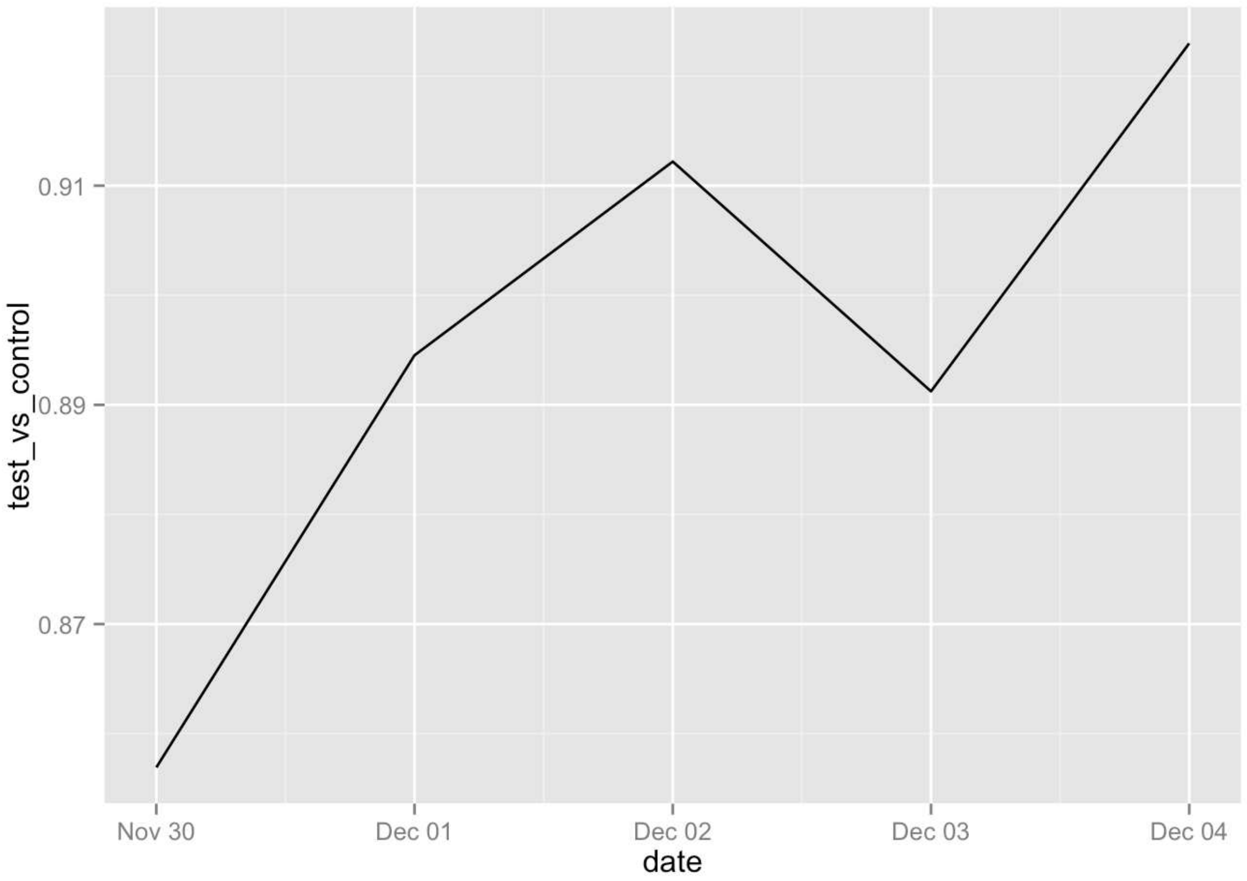
Mmh…not in the test are converting at 4.8% while users in the test just at 4.3%. That's a 10% drop, which would be dramatic if it were true. The most likely reason for weird A/B test results are:

1. We didn't collect enough data.
2. Some bias has been introduced in the experiment so that test/control people are not really random.

In data science, whenever results appear too bad or too good to be true, they are not true.

Firstly, let's plot day by day, to see if these weird results have been constantly happening or they just started happening all of a sudden.

```
data_test_by_day = data_test %>%
                   group_by(date) %>%
                   summarize(test_vs_control = mean(conversion[test==1])/
                                                mean(conversion[test==0])
                             )
qplot(date, test_vs_control, data= data_test_by_day, geom = "line")
```



From the plot, we notice a couple of things:

1. Test has constantly been worse than control and there is relatively little variance across days. That probably means that we do have enough data, but there was some bias in the experiment set up.
2. On a side note, we just ran it for 5 days. We should always run the test for at least 1 full week to capture weekly patterns, 2 weeks would be much better.

Time to find out the bias! Likely, there is for some reason some segment of users more likely to end up in test or in control, this segment had a significantly above/below conversion rate and this affected the overall results.

In an ideal world, the distribution of people in test and control for each segment should be the same. There are many ways to check this. One way is to build a decision tree where the variables are the user dimensions and the outcome variable is whether the user is in test or control. If the tree splits, it means that for given values of that variable you are more likely to end up in test or control. But this should be impossible! Therefore, if the randomization worked, the tree should not split at all (or at least not be able to separate the two classes well).

Let’s check this:

```
tree = rpart(test ~ .,data_test[,-8], #we remove conversion. Doesn't matter now.
             control = rpart.control(minbucket = nrow(data_test)/100, maxdepth =
2) # we only look for segments representing at least 1% of the populations.
             )
tree # here we are not too interested in predictive power, we are mainly using the
tree as a descriptive stat tool.
```

```
## n= 401085
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 401085 99692.820 0.5379757
##   2) country=Bolivia,Chile,Colombia,Costa Rica,Ecuador,El Salvador,Guatemala,Ho
nduras,Mexico,Nicaragua,Panama,Paraguay,Peru,Venezuela 350218 87553.970 0.4987693
*
##   3) country=Argentina,Uruguay 50867  7894.097 0.8079108 *
```

Looks very interesting. The randomization is perfect for the countries on one side of the split (country=Bolivia, Chile, Colombia, Costa Rica, Ecuador, EL Salvador, Guatemala, Honduras, Mexico, Nicaragua, Panama, Paraguay, Peru, Venezuela). Indeed, in that leaf the test/control ratio is 0.498! However, Argentina and Uruguay together have 80% test and 20% control! So let's check the test results after controlling for country. That is, we check for each country how the test is doing:

```
data_test_country =  data_test %>%
                     group_by(country) %>%
                     summarize( p_value = t.test( conversion[test==1],conversion[t
est==0])$p.value,
                                conversion_test = t.test( conversion[test==1],conv
ersion[test==0])$estimate[1],
                                conversion_control = t.test( conversion[test==1],c
onversion[test==0])$estimate[2]
                              ) %>%
                     arrange (p_value)

data_test_country
```

```
## Source: local data frame [16 x 4]
##
##        country   p_value conversion_test conversion_control
##         (fctr)     (dbl)           (dbl)              (dbl)
## 1       Mexico 0.1655437      0.05118631         0.04949462
## 2  El Salvador 0.2481267      0.04794689         0.05355404
## 3        Chile 0.3028476      0.05129502         0.04810718
## 4    Argentina 0.3351465      0.01372502         0.01507054
## 5     Colombia 0.4237191      0.05057096         0.05208949
## 6     Honduras 0.4714629      0.04753981         0.05090576
## 7    Guatemala 0.5721072      0.04864721         0.05064288
## 8    Venezuela 0.5737015      0.04897831         0.05034367
## 9   Costa Rica 0.6878764      0.05473764         0.05225564
## 10      Panama 0.7053268      0.04937028         0.04679552
## 11     Bolivia 0.7188852      0.04790097         0.04936937
## 12        Peru 0.7719530      0.05060427         0.04991404
## 13   Nicaragua 0.7804004      0.05417676         0.05264697
## 14     Uruguay 0.8797640      0.01290670         0.01204819
## 15    Paraguay 0.8836965      0.04922910         0.04849315
## 16     Ecuador 0.9615117      0.04898842         0.04915381
```

After we control for country, the test clearly appears non significant. Not a great success given that the goal was to improve conversion rate, but at least we know that a localized translation didn't make things worse!