Conversion Rate

```
#read library
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(rpart)
library(rpart.plot)
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
library(RColorBrewer)
setwd("/Users/zhongming/Desktop/DS\ Project/Conversion\ Rate/")
data=read.csv('conversion_data.csv')
head(data)
```

```
##   country age new_user source total_pages_visited converted
## 1      UK  25        1    Ads                   1         0
## 2      US  23        1    Seo                   5         0
## 3      US  28        1    Seo                   4         0
## 4   China  39        1    Seo                   5         0
## 5      US  30        1    Seo                   6         0
## 6      US  31        0    Seo                   1         0
```

```
#see structure of data
summary(data)
```

```
##     country            age            new_user         source
##  China  : 76602   Min.   : 17.00   Min.   :0.0000   Ads    : 88740
##  Germany: 13056   1st Qu.: 24.00   1st Qu.:0.0000   Direct: 72420
##  UK     : 48450   Median : 30.00   Median :1.0000   Seo    :155040
##  US     :178092   Mean   : 30.57   Mean   :0.6855
##                   3rd Qu.: 36.00   3rd Qu.:1.0000
##                   Max.   :123.00   Max.   :1.0000
##  total_pages_visited   converted
##  Min.   : 1.000      Min.   :0.00000
##  1st Qu.: 2.000      1st Qu.:0.00000
##  Median : 4.000      Median :0.00000
##  Mean   : 4.873      Mean   :0.03226
##  3rd Qu.: 7.000      3rd Qu.:0.00000
##  Max.   :29.000      Max.   :1.00000
```

```
str(data)
```

```
## 'data.frame':    316200 obs. of  6 variables:
##  $ country            : Factor w/ 4 levels "China","Germany",..: 3 4 4 1 4 4 1
4 3 4 ...
##  $ age                : int  25 23 28 39 30 31 27 23 29 25 ...
##  $ new_user           : int  1 1 1 1 1 0 1 0 0 0 ...
##  $ source             : Factor w/ 3 levels "Ads","Direct",..: 1 3 3 3 3 3 3 1 2
1 ...
##  $ total_pages_visited: int  1 5 4 5 6 1 4 4 4 2 ...
##  $ converted          : int  0 0 0 0 0 0 0 0 0 0 ...
```

data seems to be clean. However, the 123 age dosen't make sense.

```
sort(unique(data$age),decreasing=T)
```

```
##  [1] 123 111  79  77  73  72  70  69  68  67  66  65  64  63  62  61  60
## [18]  59  58  57  56  55  54  53  52  51  50  49  48  47  46  45  44  43
## [35]  42  41  40  39  38  37  36  35  34  33  32  31  30  29  28  27  26
## [52]  25  24  23  22  21  20  19  18  17
```

Why not take a look at it?

```
table(data$age)
```

```
##
##    17    18    19    20    21    22    23    24    25    26    27    28
##  7597  8466  9349 10156 10966 11701 12336 12960 13460 13931 14084 14341
##    29    30    31    32    33    34    35    36    37    38    39    40
## 14158 14346 13692 13507 12631 12108 11471 10779  9761  8970  8202  7148
##    41    42    43    44    45    46    47    48    49    50    51    52
##  6401  5588  4904  4224  3634  2994  2504  2121  1629  1356  1093   935
##    53    54    55    56    57    58    59    60    61    62    63    64
##   627   520   394   286   247   165   127    94    71    59    35    27
##    65    66    67    68    69    70    72    73    77    79   111   123
##    15     9     5     5     3     2     1     1     1     1     1     1
```

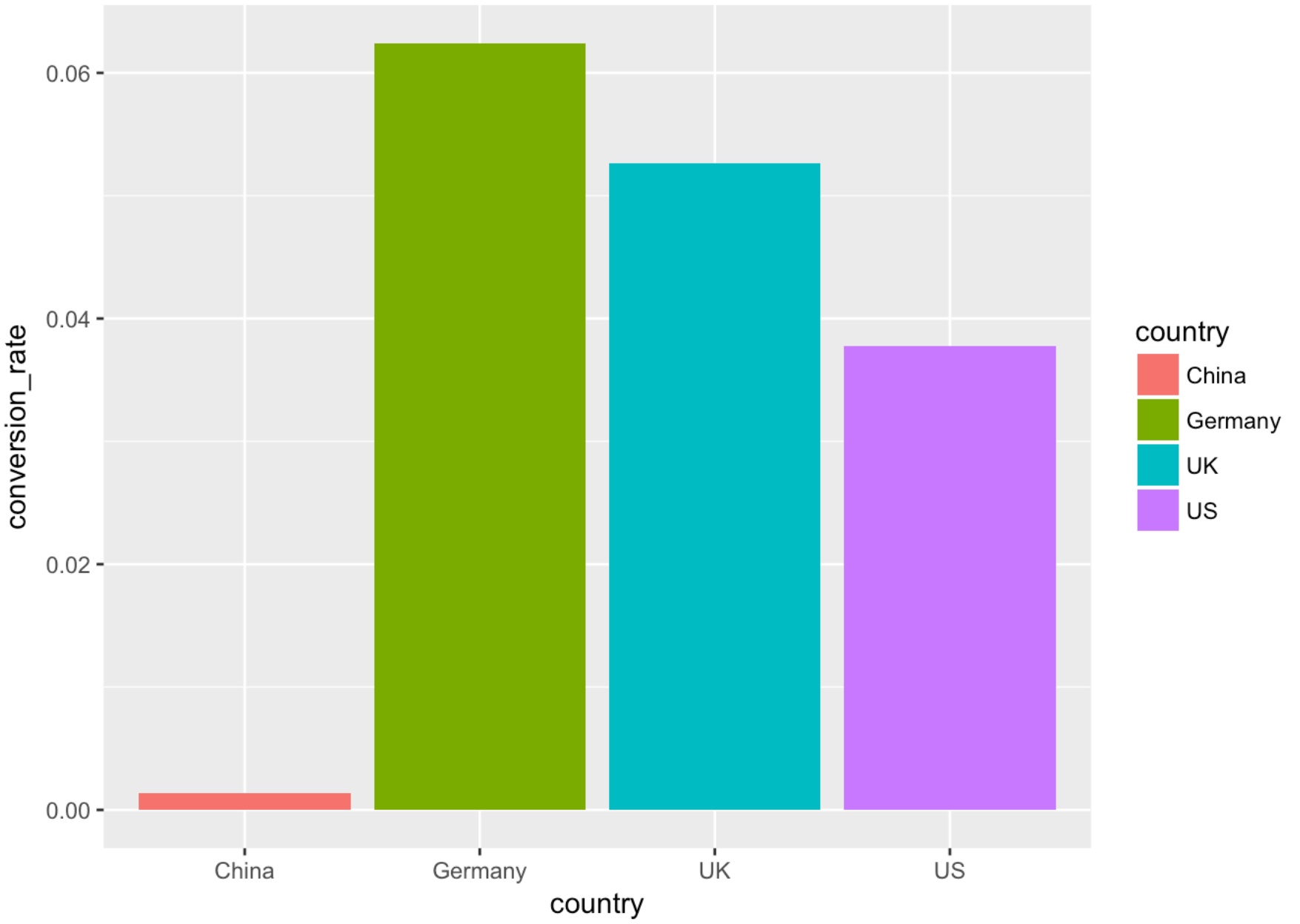
We can see that only two users have really unreasonable age. We can simly remove this 2 users.

```
data=subset(data,age<80)
```

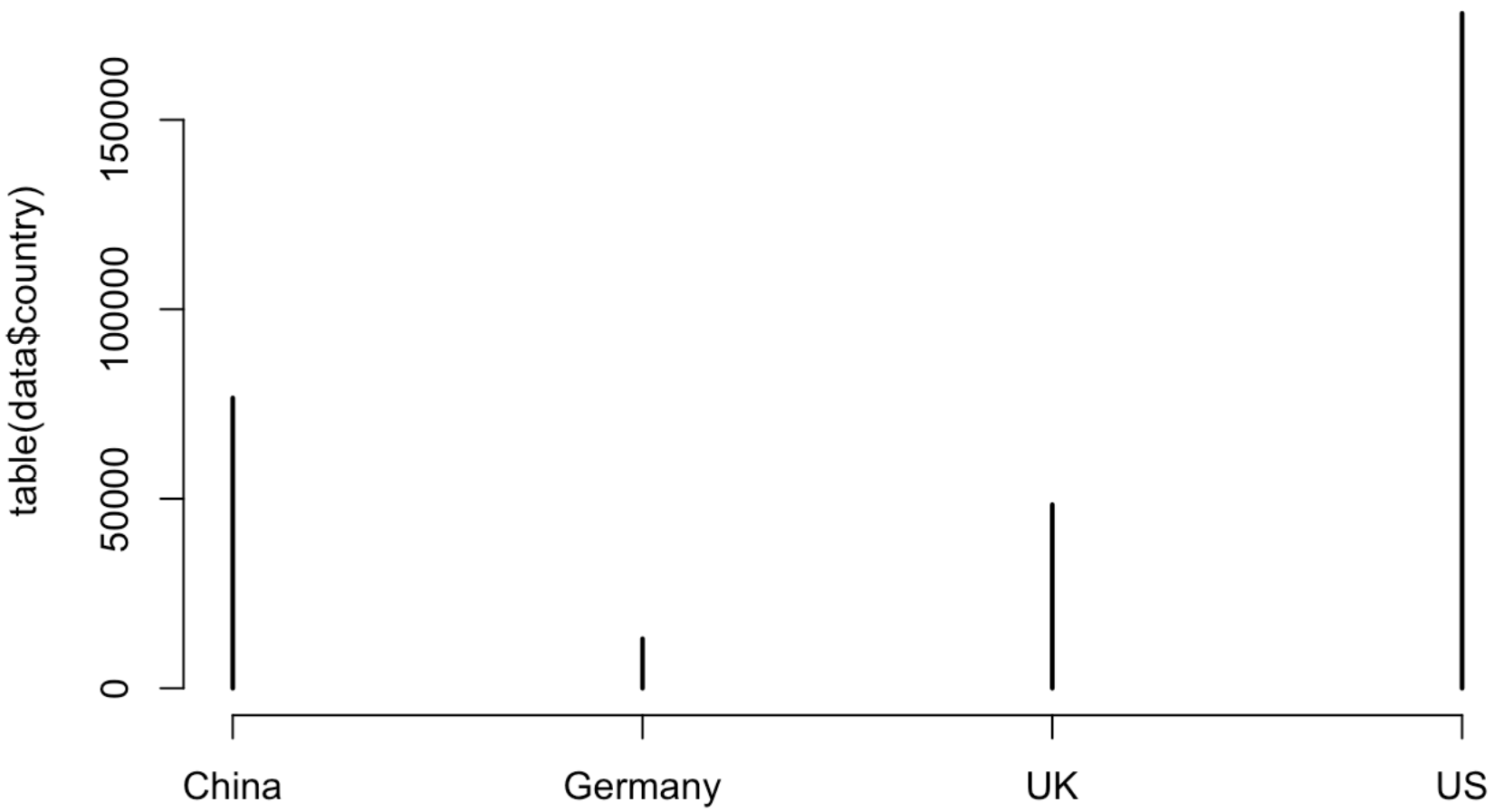
Let's do some EDA first! Now, remeber we care about the conversion rate. So, we need to see whether country/new_user/sources/pages have impact on convversion rate.

```
#see country
data_country=data%>%group_by(country)%>%summarise(conversion_rate=mean(converted))

ggplot(data=data_country,aes(x=country,y=conversion_rate))+geom_bar(stat="identity
",aes(fill=country))
```

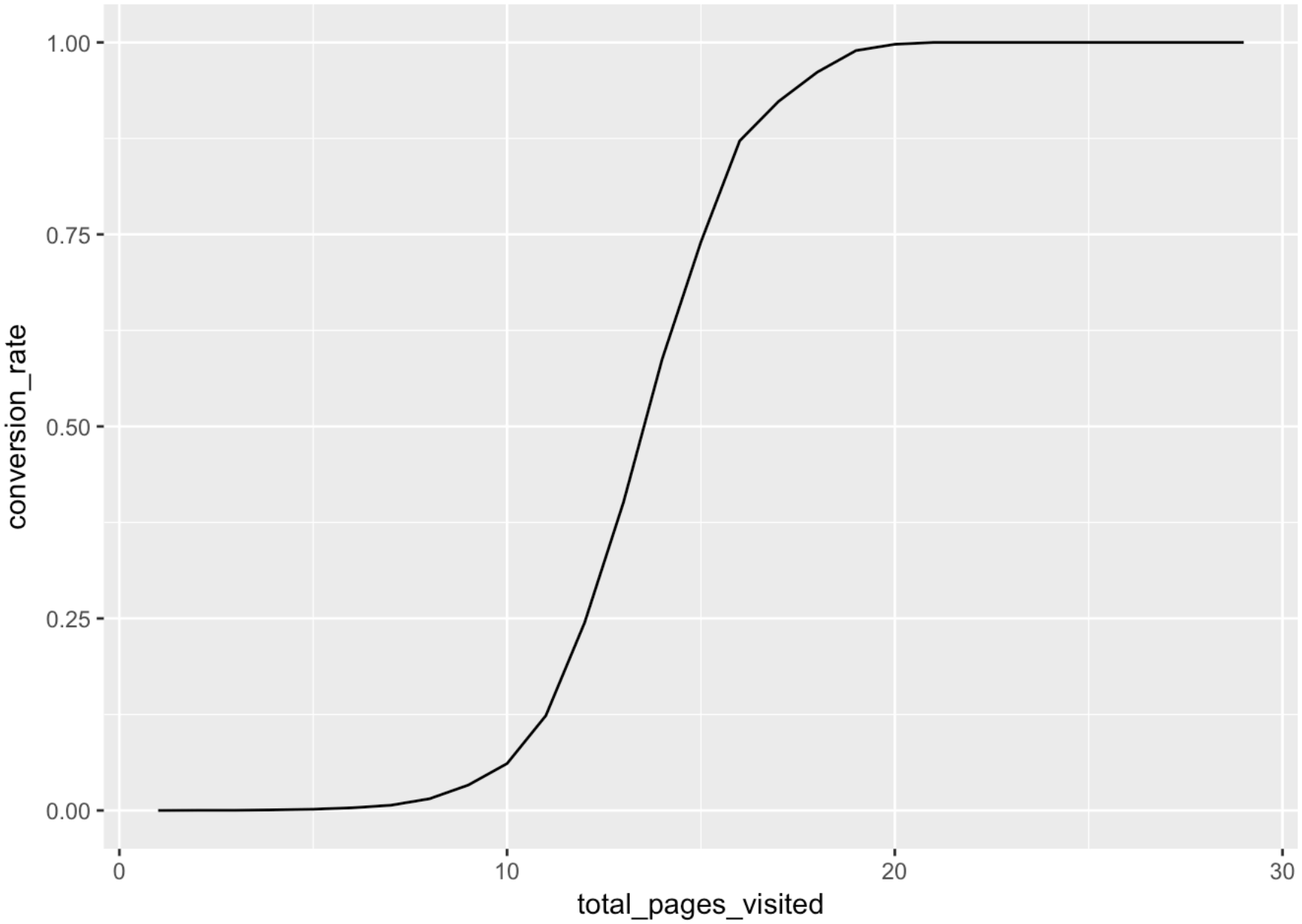



```
plot(table(data$country))
```



So, it is clear that China has small conversion rate.

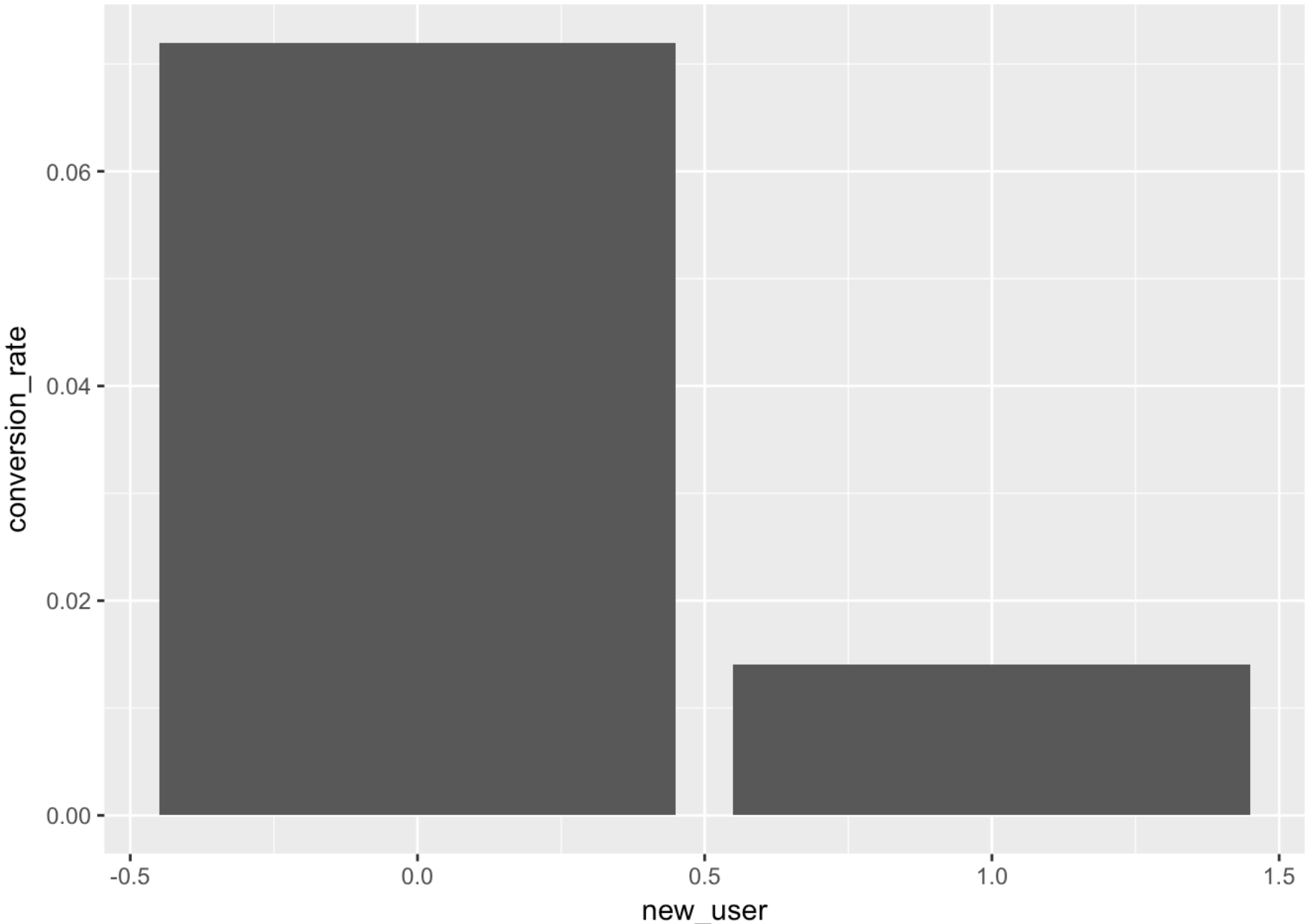
```
#Take a look at page
data_pages=data%>%group_by(total_pages_visited)%>%summarise(conversion_rate=mean(c
onverted))
qplot(total_pages_visited,conversion_rate,data=data_pages,geom="line")
```



Very interesting.Total pages and conversion rate seems to have a logistic relationship. Because people spend more time on websites, they tend to convvert more easily.

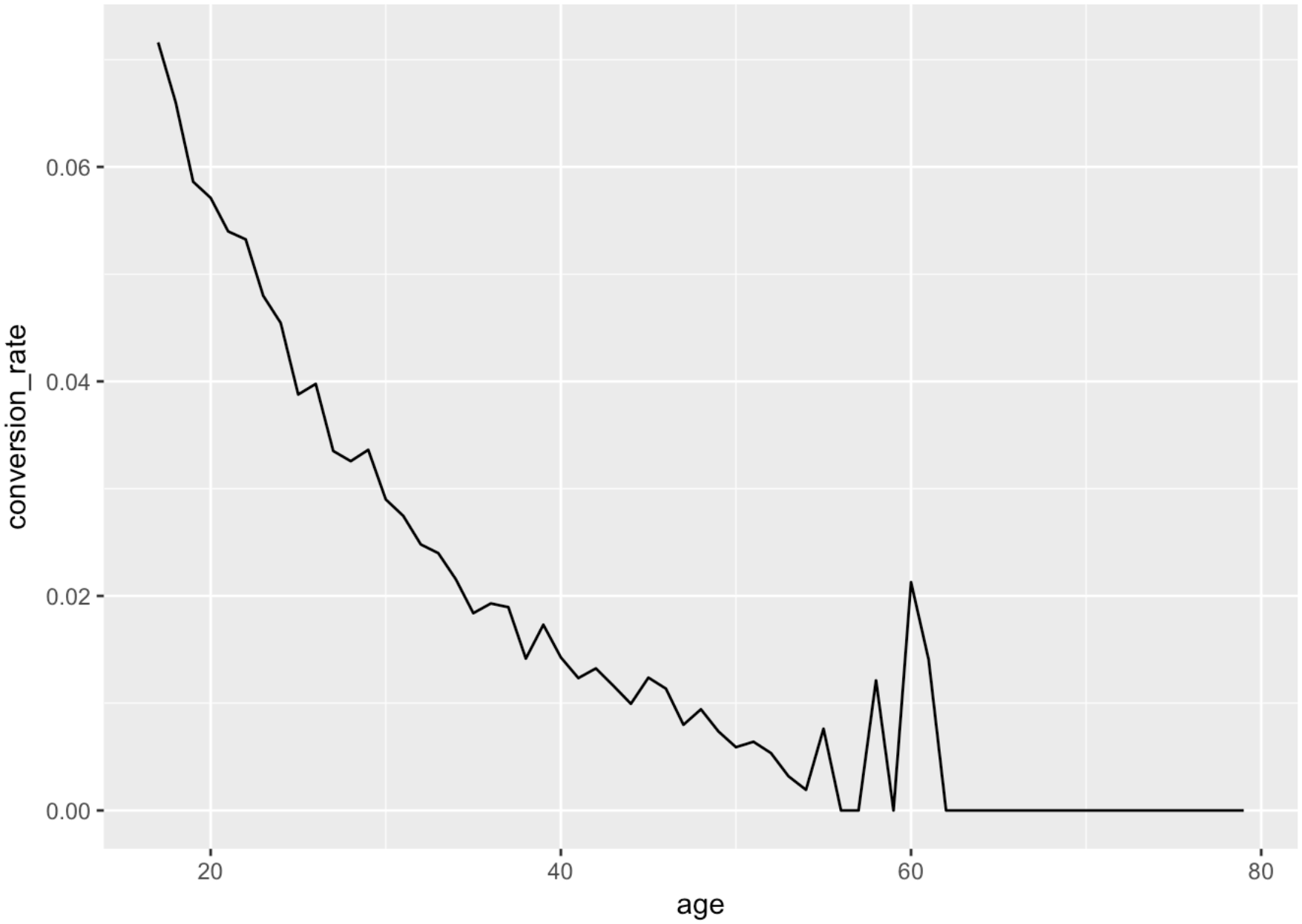
```
data_users=data%>%group_by(new_user)%>%summarise(conversion_rate=mean(converted))

ggplot(data=data_users,aes(x=new_user,y=conversion_rate))+geom_bar(stat="identity"
)
```



Old uses are more easily to converte. THis is very reasonable.

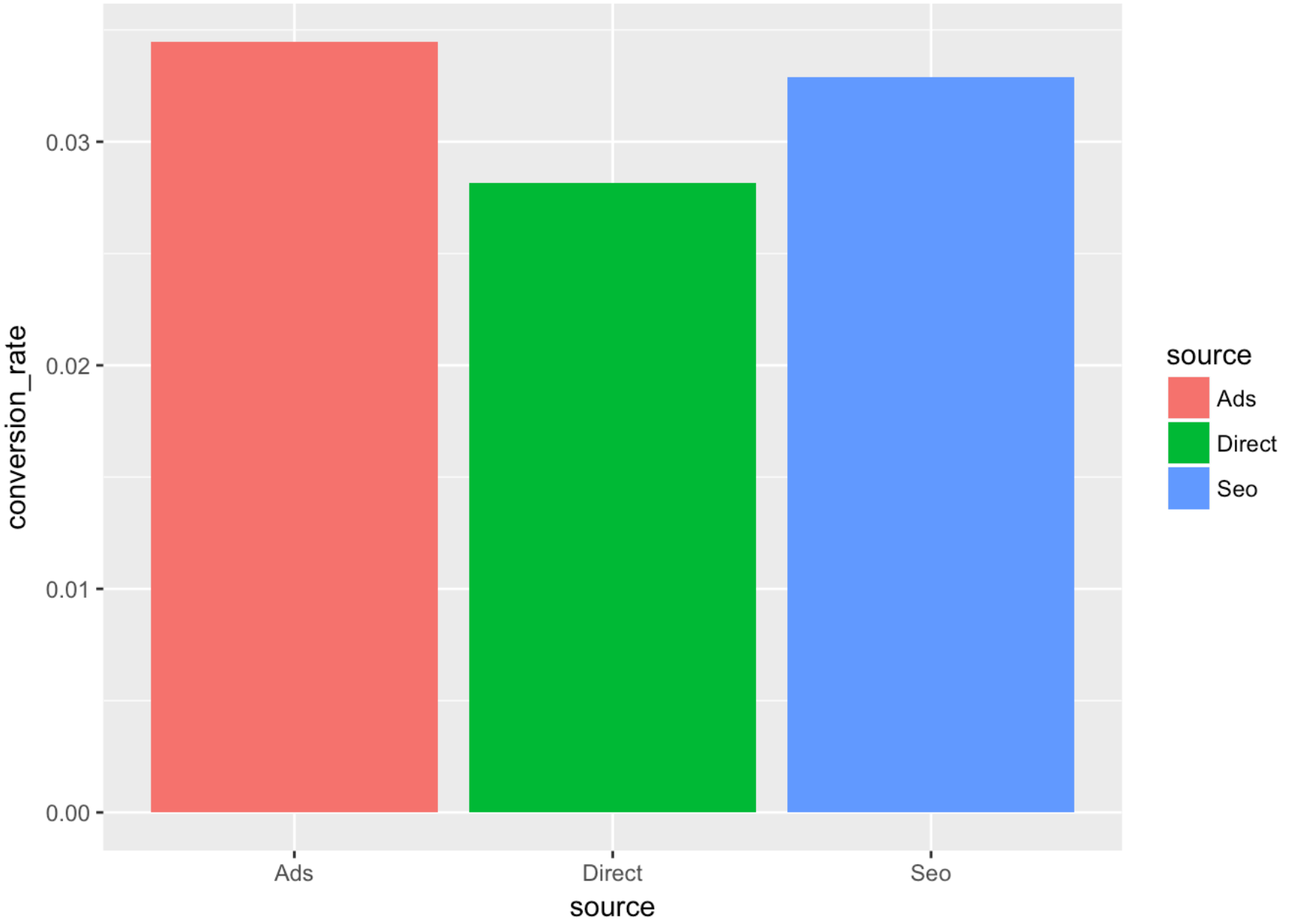
```
#Take a look at  age
data_age=data%>%group_by(age)%>%summarise(conversion_rate=mean(converted))
qplot(age,conversion_rate,data=data_age,geom="line")
```



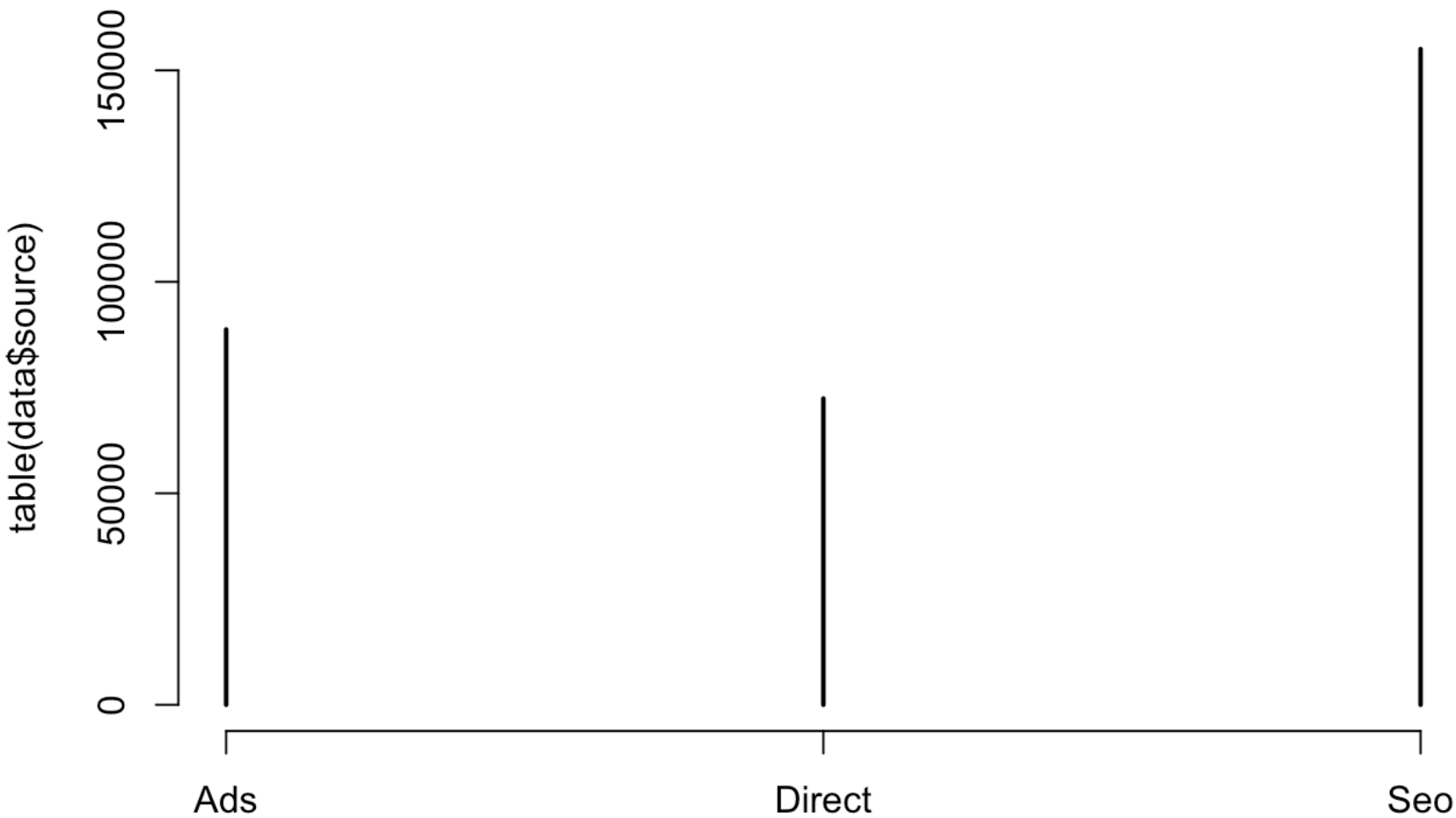
Well, I guess we can say that old people are stubborn.Or old people are not easy to change.HAHA!

```
#take a look at souces
data_source=data%>%group_by(source)%>%summarise(conversion_rate=mean(converted))

ggplot(data=data_source,aes(x=source,y=conversion_rate))+geom_bar(stat="identity",
aes(fill=source))
```




```
plot(table(data$source))
```



Well, most of the users click through SEO. Other use Ads. Not much people visit the site through direct URL.Ads has the highest conversion rate.

Machine Learning!!

Here is the fun but the hardest part of this project!It is clear that it is a binary classificaiton problem. 1.Logistic2.Decision Tree3.RuleFit4.Random Forest in combination with partial dependence plots5.KNN6.SVM7.MLP8.lightGBM9.xgboost10.NN11.Naive Bayes Classifer12.RVM or Relevance Vector Machine13.PerceptronBut we may just try only several of them because some methods are better to be perfromed in pyton rather than R.

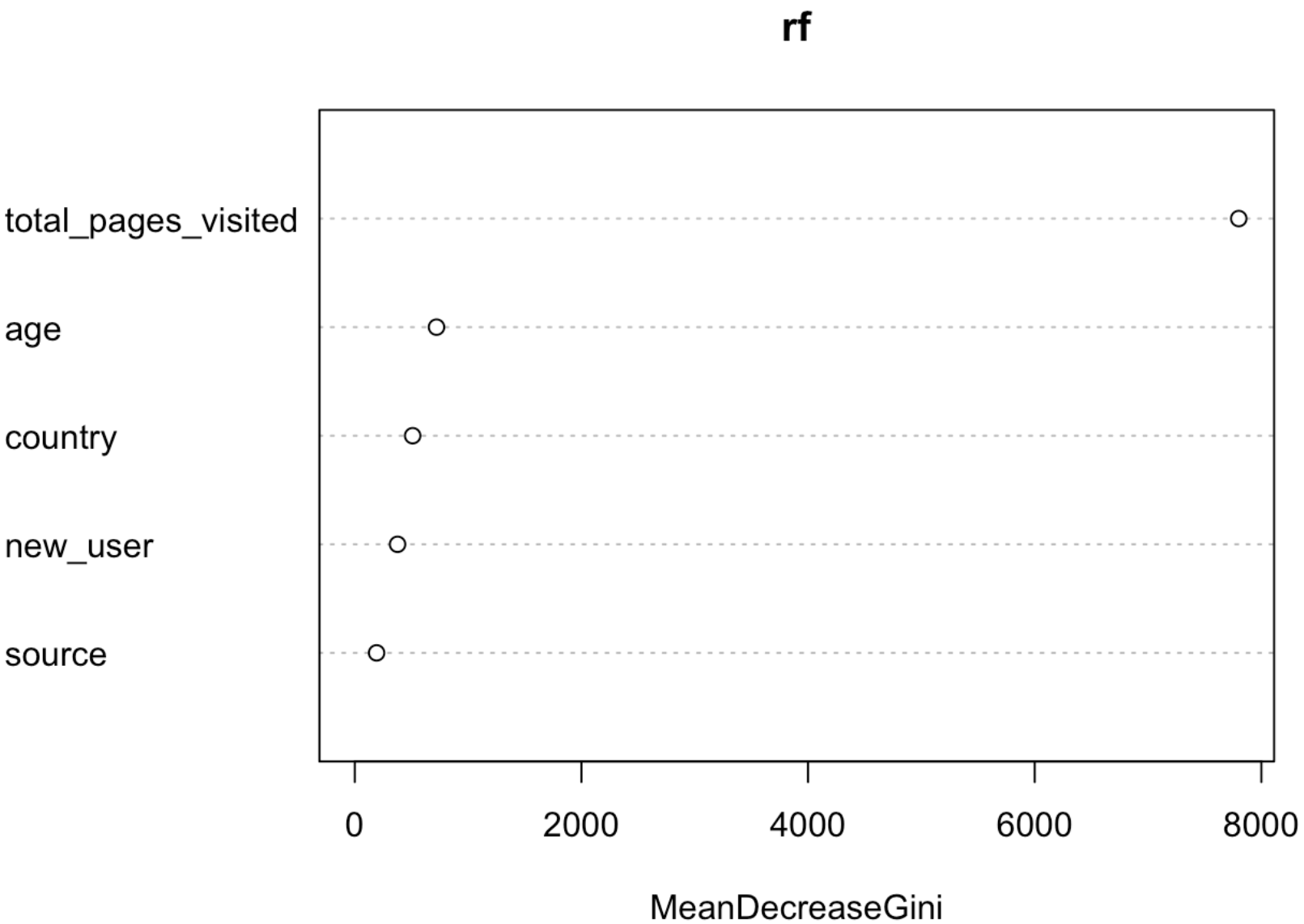
```
#convert bnary variable into factor
data$new_user=as.factor(data$new_user)
data$converted=as.factor(data$converted)
```

```
#Alright! Split data into train and test set
train_sample=sample(nrow(data),size=nrow(data)*0.66)
train_data=data[train_sample,]
test_data=data[-train_sample,]
#Random Forest.Wow
rf=randomForest(y=train_data$converted,x=train_data[,-ncol(train_data)],ytest=test
_data$converted,xtest=test_data[,-ncol(test_data)],ntree=100,mtry=3,keep.forest=T)
rf
```

```
##
## Call:
##  randomForest(x = train_data[, -ncol(train_data)], y = train_data$converted,
xtest = test_data[, -ncol(test_data)], ytest = test_data$converted,      ntree = 1
00, mtry = 3, keep.forest = T)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.42%
## Confusion matrix:
##        0    1 class.error
## 0 201088  832 0.004120444
## 1   2139 4631 0.315952733
##                 Test set error rate: 1.52%
## Confusion matrix:
##        0    1 class.error
## 0 103644  436 0.004189085
## 1   1194 2234 0.348308051
```

Well. We can see that the error rate from train set and test set is similart.So we are not overfitting. And the error is really small.However, when we see the class error, this algorithm clssify 32% of conversion as not converted. This algorithm should be further modified using ROCR and find the best cut-off point.

```
varImpPlot(rf,type=2)
```

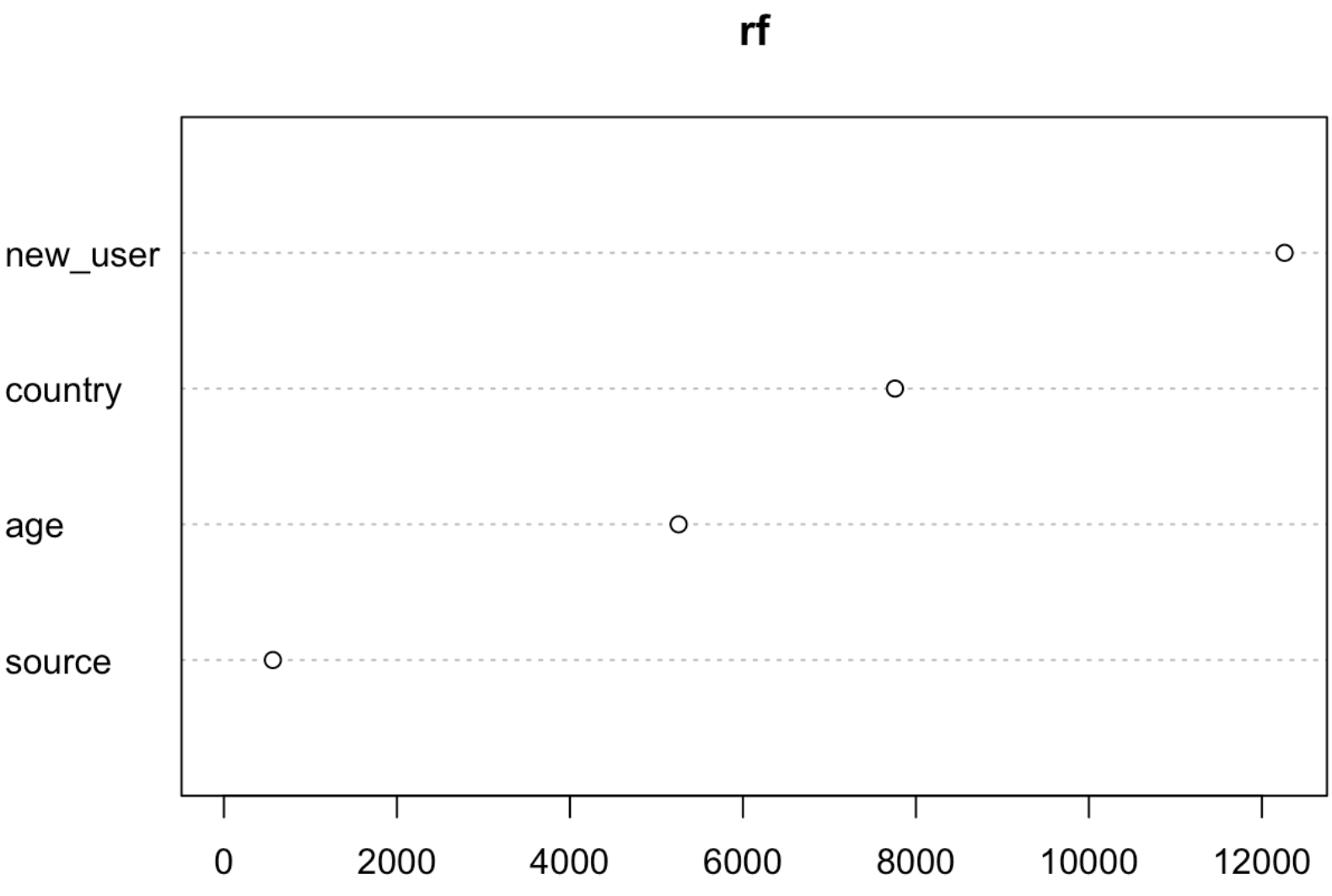


The total pages visited is the most important factor. But we cannot do much about that. SAD!

```
rf=randomForest(y=train_data$converted,x=train_data[,-c(5,ncol(train_data))],ytest
=test_data$converted,xtest=test_data[,-c(5,ncol(test_data))],ntree=100,mtry=3,keep
.forest=T,classwt=c(0.7,0.3))
rf
```

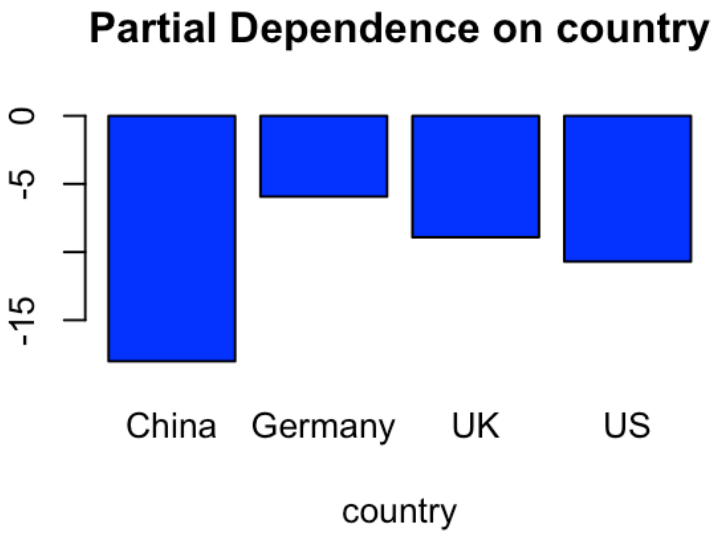
```
##
## Call:
##  randomForest(x = train_data[, -c(5, ncol(train_data))], y = train_data$convert
ed,      xtest = test_data[, -c(5, ncol(test_data))], ytest = test_data$converted,
ntree = 100, mtry = 3, classwt = c(0.7, 0.3), keep.forest = T)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 14.16%
## Confusion matrix:
##        0     1 class.error
## 0 175433 26487   0.1311757
## 1   3069  3701   0.4533235
##                 Test set error rate: 14.29%
## Confusion matrix:
##       0     1 class.error
## 0 90281 13799   0.1325807
## 1  1568  1860   0.4574096
```

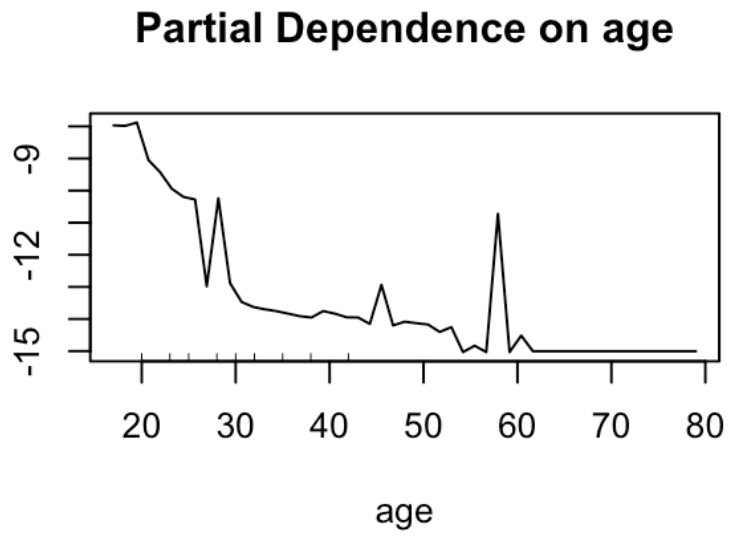
```
varImpPlot(rf,type=2)
```

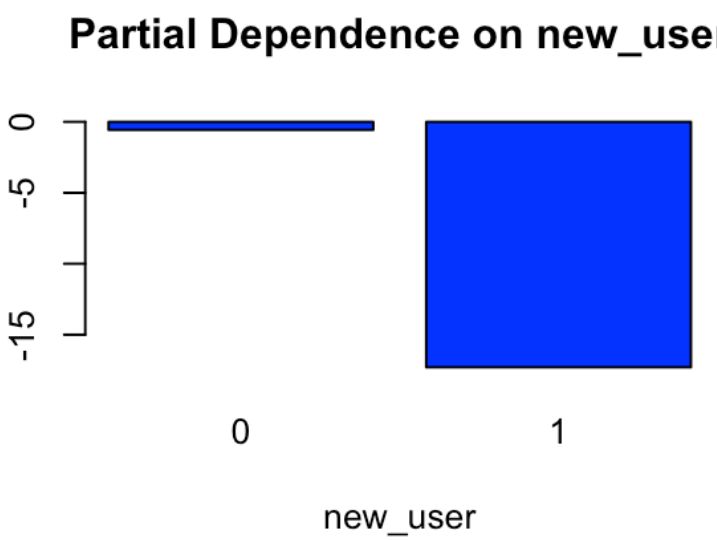


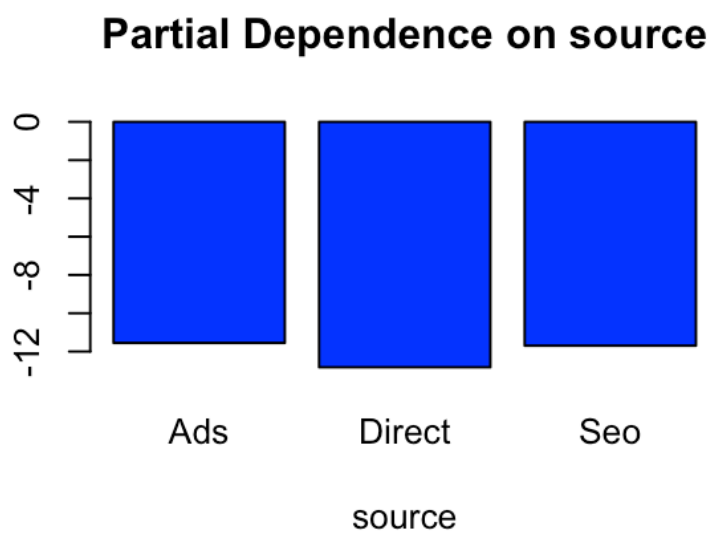
Wow. After ditching the page visited, we can see that new_user is actually the most important factor. And source seems to be not important.

```
op<-par(mfrow=c(2,2))
partialPlot(rf,train_data,country,1)
partialPlot(rf,train_data,age,1)
partialPlot(rf,train_data,new_user,1)
partialPlot(rf,train_data,source,1)
```







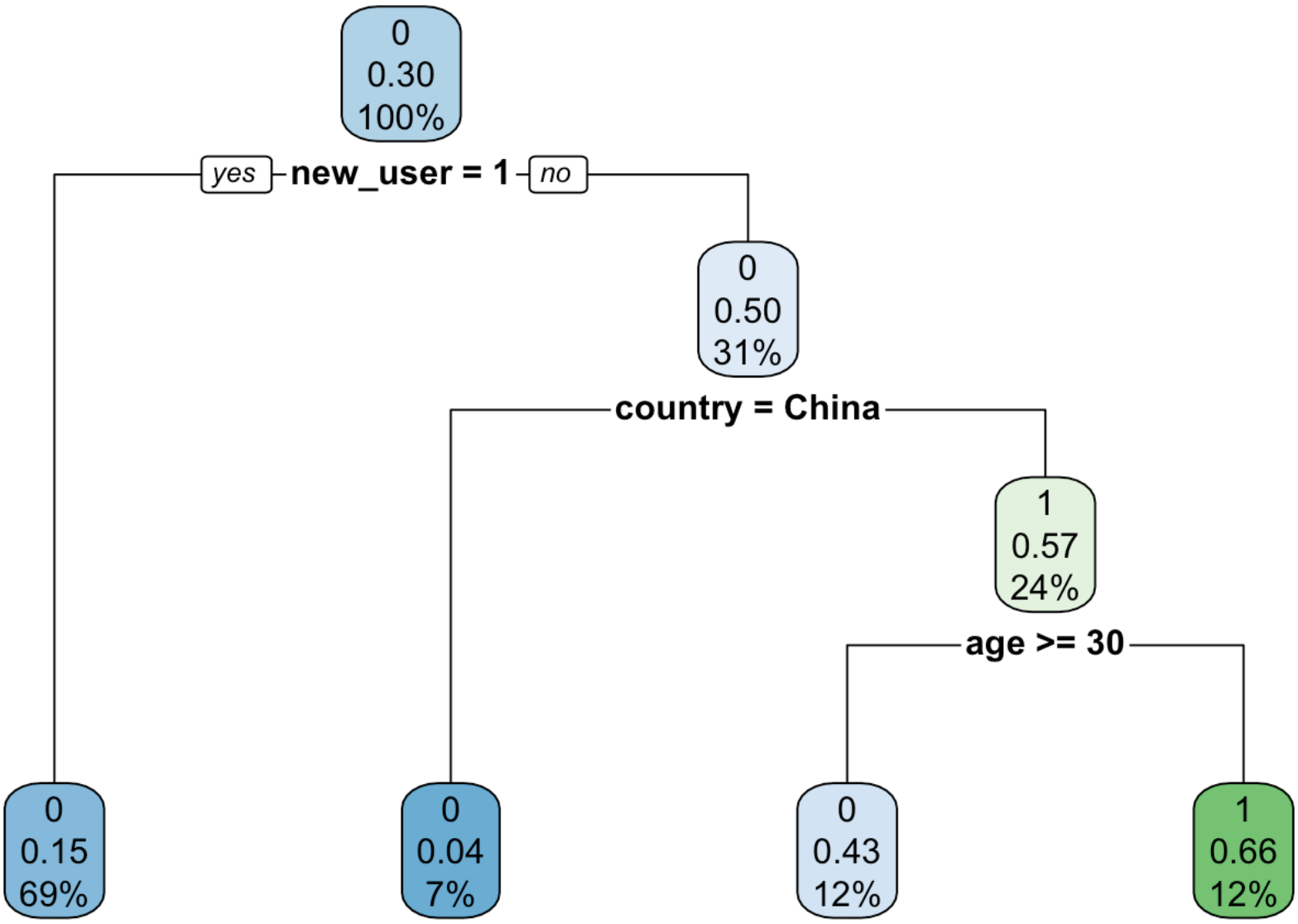


Here are the insight we can tell:Users from China are tough to convert.The site works well for young people but less attractive for old people(>30 years old). For the sudden bouce near 60, I think we can not draw any conclusion because data point near 60 is small. source is not important.

```
#build a simple decision tree to confirm our conclusion
tree=rpart(data$converted~.,data[,-c(5,ncol(data))],control=rpart.control(maxdepth
= 3),parms=list(prior=c(0.7,0.3)))
tree
```

```
## n= 316198
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 316198 94859.4000 0 (0.70000000 0.30000000)
##    2) new_user=1 216744 28268.0600 0 (0.84540048 0.15459952) *
##    3) new_user=0 99454 66591.3400 0 (0.50063101 0.49936899)
##      6) country=China 23094   613.9165 0 (0.96445336 0.03554664) *
##      7) country=Germany,UK,US 76360 50102.8100 1 (0.43162227 0.56837773)
##       14) age>=29.5 38341 19589.5200 0 (0.57227507 0.42772493) *
##       15) age< 29.5 38019 23893.0000 1 (0.33996429 0.66003571) *
```

```
rpart.plot(tree)
```



I have to say: this is way more fun than I imagine. To convert a user, we need it to be an old user, and we don’t want it from China(HAHAHA!). Finally, we want to be younger(<30years)Then we can make good suggestion to PM!1. Marketing should focus on yong people.2. The site is working well for Gemnay. But we don’t have many Germany users. We can explor the market more.3. Use emails targeting on old users. 4. It might be something wrong with the Chinese verison of this site that prevent people from converting.Try to figure out. 5. Old people may not be familiar with the sites. Try to find out why. UI design ? Many those control buttons is hard to locate for old poeple. 6. For those who have already looked through many webpages in the site, we can send email to lure them into conversion.