Simple Predicting Conversion Rate from Data

Load required libraries

```
library(ggplot2)
library(dplyr)
```

```
##
## 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(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.2.1 --
```

```
## v tibble  1.4.2     v purrr   0.2.5
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
```

Read CSV Data and Cast Column Types

```
conversion_data <- read_csv("conversion_data.csv",
                            col_types = cols(
                                                new_user = col_factor(levels = NULL),
                                                source = col_factor(levels = NULL),
                                                country = col_factor(levels = NULL)))
conversion_data
```

```
## # A tibble: 316,200 x 6
##    country   age new_user source total_pages_visited converted
##    <fct>   <int> <fct>    <fct>                <int>     <int>
##  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
##  7 China      27 1        Seo                      4         0
##  8 US         23 0        Ads                      4         0
##  9 UK         29 0        Direct                   4         0
## 10 US         25 0        Ads                      2         0
## # ... with 316,190 more rows
```

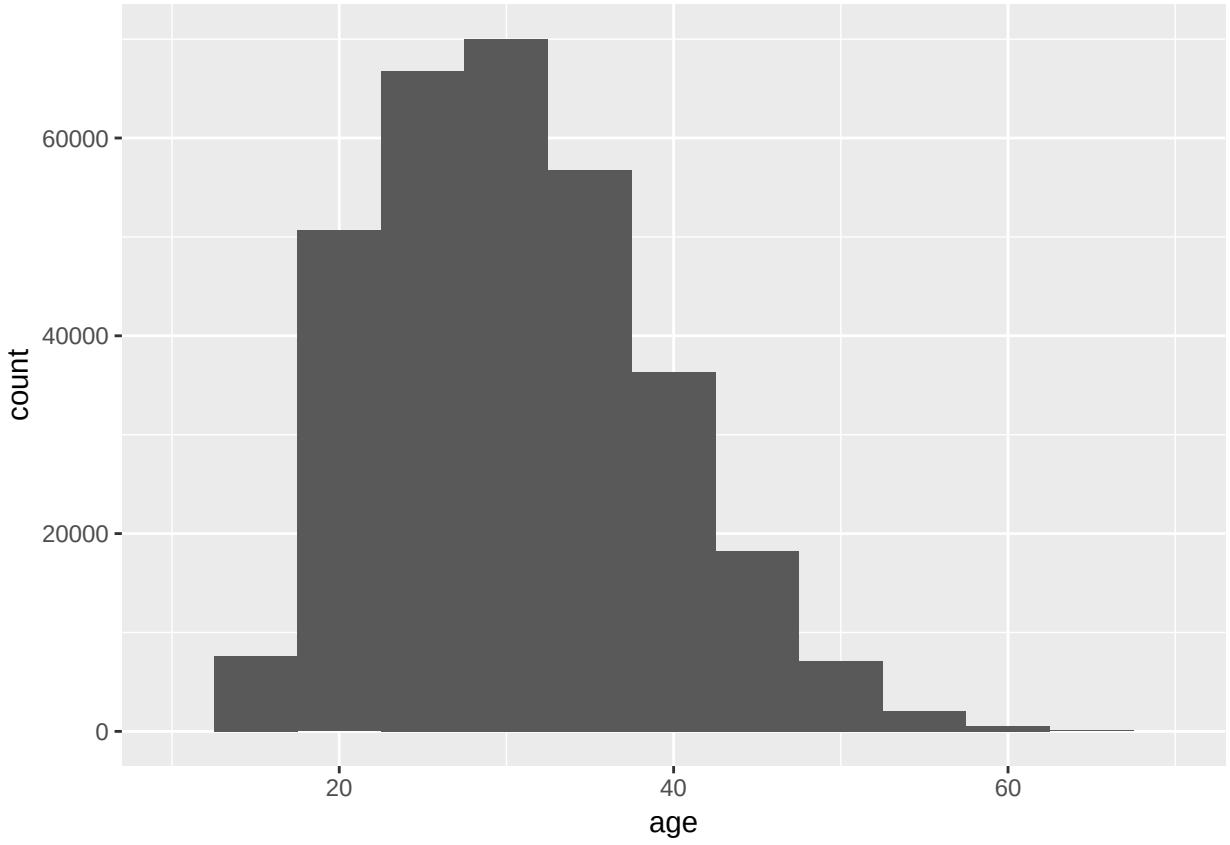
```r
summary(conversion_data)
```

```
##     country            age          new_user      source
##  UK      : 48450   Min.   : 17.00   1:216744   Ads    : 88740
##  US      :178092   1st Qu.: 24.00   0: 99456   Seo    :155040
##  China   : 76602   Median : 30.00              Direct: 72420
##  Germany: 13056   Mean   : 30.57
##                   3rd Qu.: 36.00
##                   Max.   :123.00
##  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
```

Data Exploration

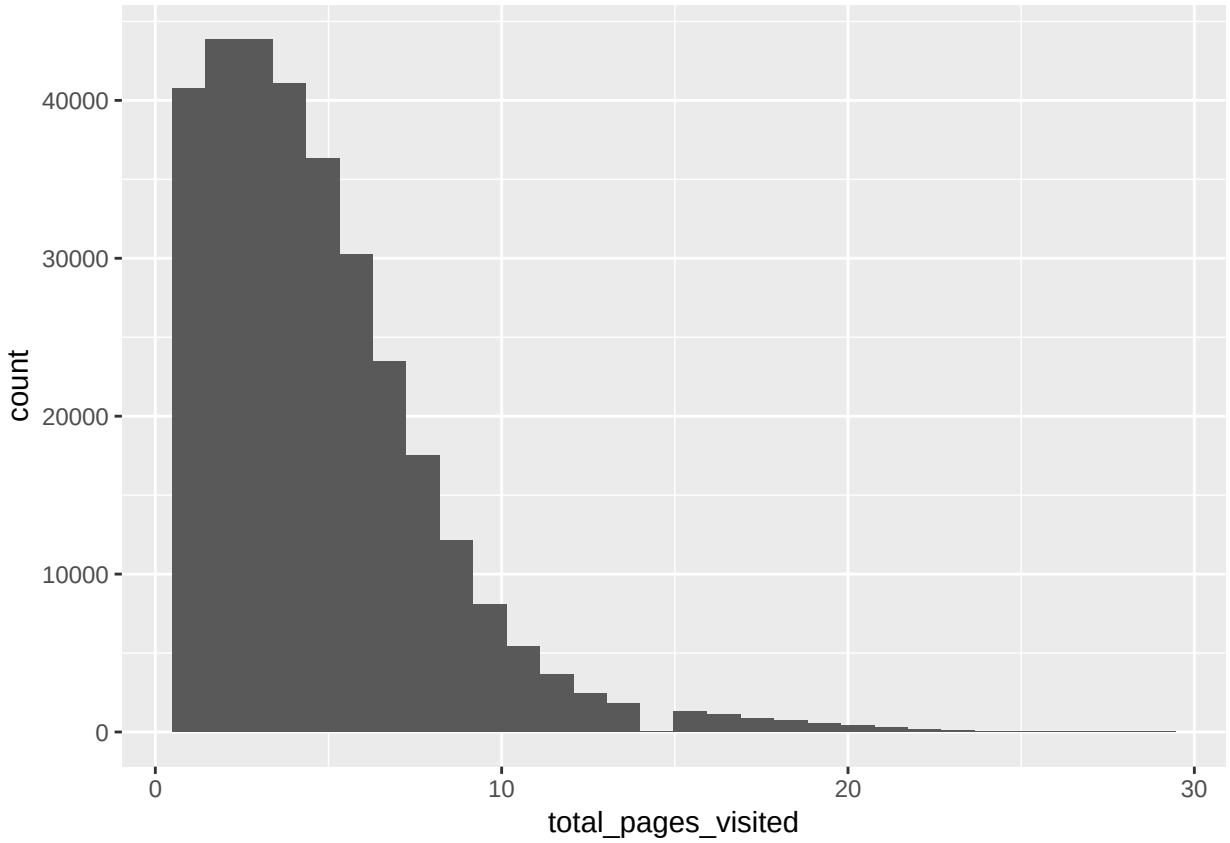
```r
#Histogram for Age
ggplot(conversion_data, aes(x=age)) + geom_histogram(binwidth = 5) + xlim(10,70)
```

```
## Warning: Removed 6 rows containing non-finite values (stat_bin).
```




```r
#Histogram for Total Pages Visited
ggplot(conversion_data, aes(x=total_pages_visited)) + geom_histogram()
```

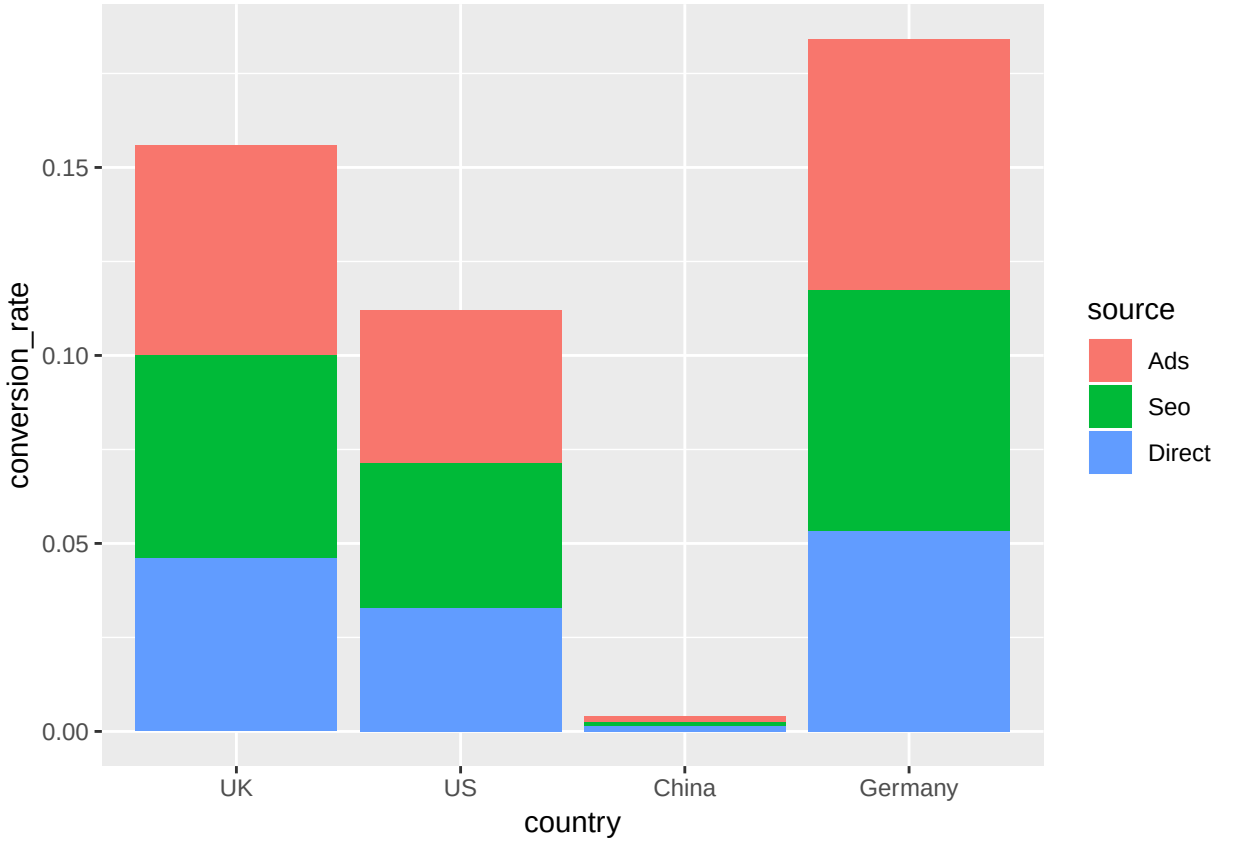
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
data_country_source <- conversion_data %>%
  group_by(country,source) %>%
  summarize(conversion_rate = mean(converted))

data_country_source
```

```
## # A tibble: 12 x 3
## # Groups:   country [?]
##    country source conversion_rate
##    <fct>   <fct>            <dbl>
##  1 UK      Ads             0.0556
##  2 UK      Seo             0.0539
##  3 UK      Direct          0.0463
##  4 US      Ads             0.0406
##  5 US      Seo             0.0385
##  6 US      Direct          0.0329
##  7 China   Ads             0.00148
##  8 China   Seo             0.00122
##  9 China   Direct          0.00137
## 10 Germany Ads             0.0668
## 11 Germany Seo             0.0641
## 12 Germany Direct          0.0534
```

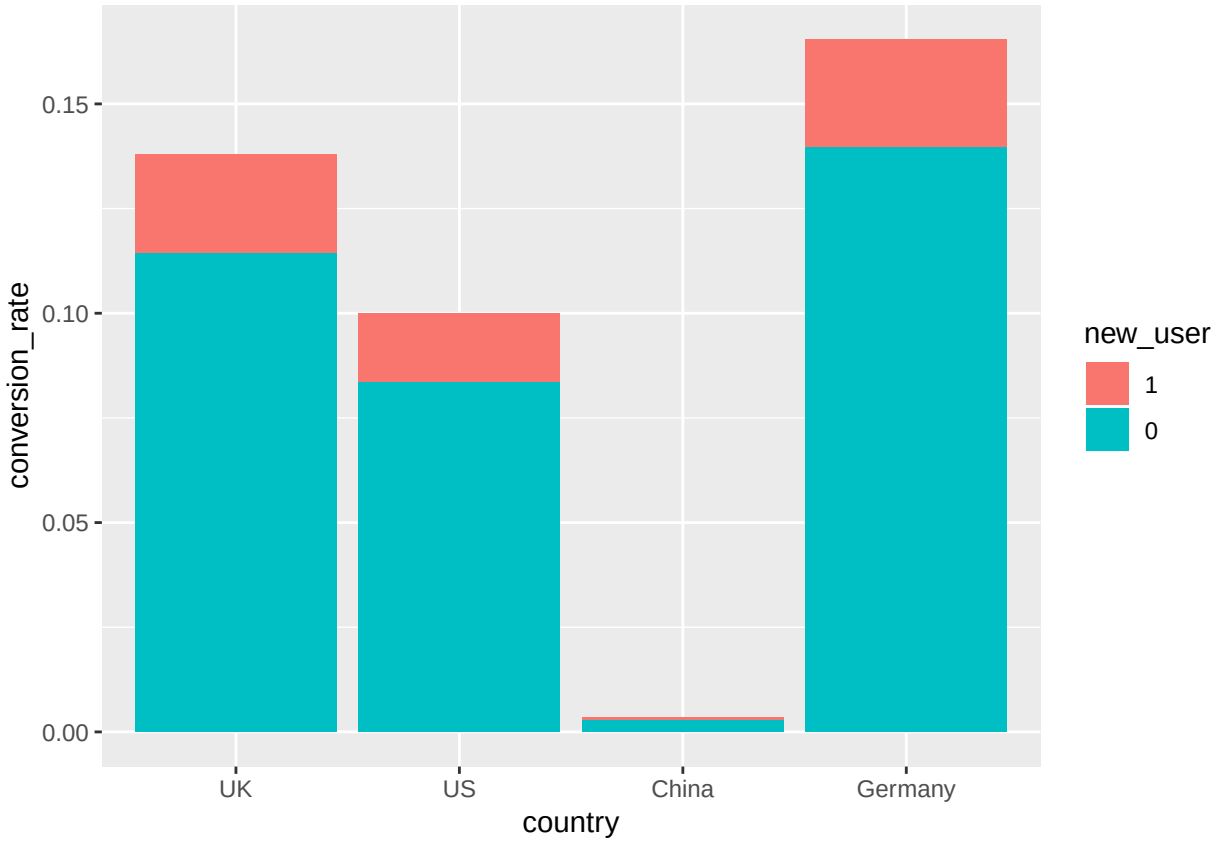
```
ggplot(data_country_source, aes(x=country, y=conversion_rate, fill = source))+  geom_col()
```




```
#country new users
data_country_users <- conversion_data %>%
  group_by(country,new_user) %>%
  summarize(conversion_rate = mean(converted))

data_country_users
```

```
## # A tibble: 8 x 3
## # Groups:   country [?]
##   country new_user conversion_rate
##   <fct>   <fct>              <dbl>
## 1 UK      1                0.0236
## 2 UK      0                0.114
## 3 US      1                0.0165
## 4 US      0                0.0836
## 5 China   1                0.000673
## 6 China   0                0.00286
## 7 Germany 1                0.0257
## 8 Germany 0                0.140
```

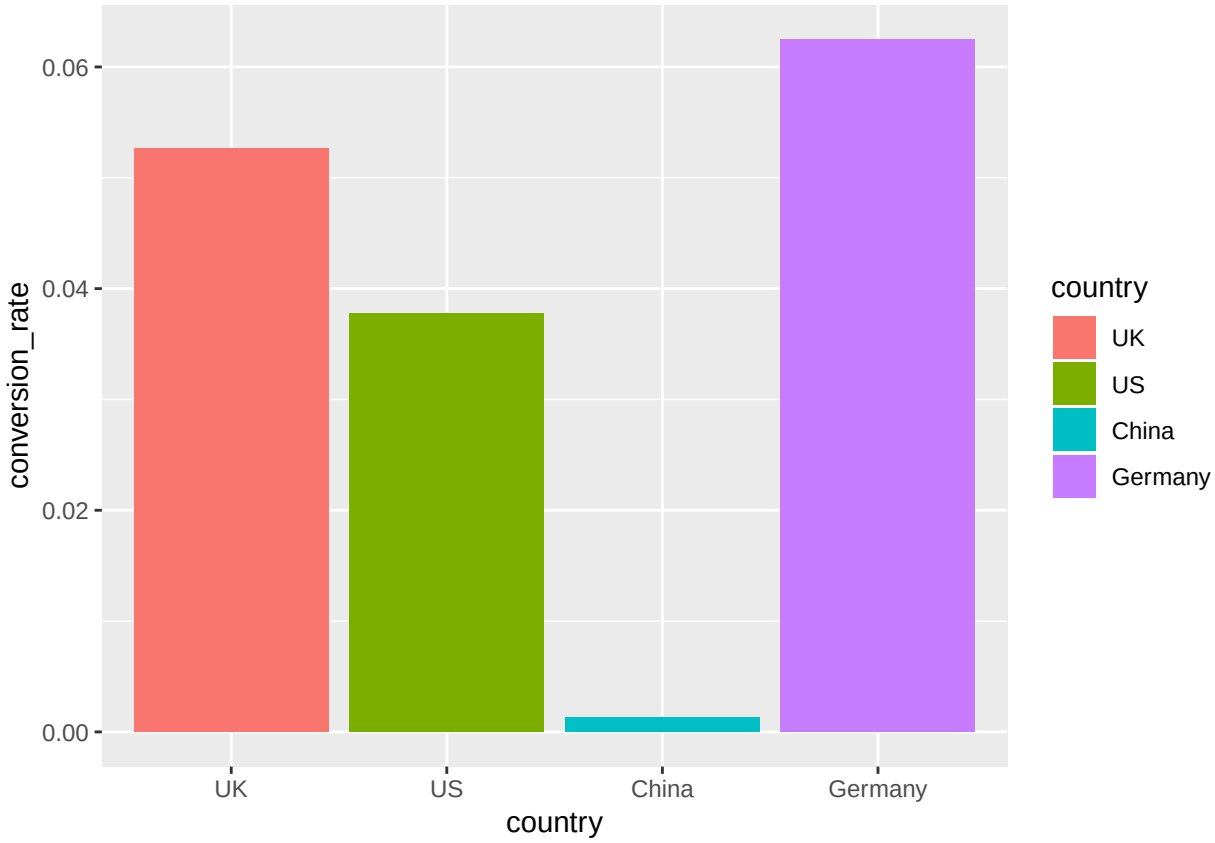
```
ggplot(data_country_users, aes(x=country, y=conversion_rate, fill = new_user))+  geom_col()
```




```
#Single Value
data_country <- conversion_data %>%
  group_by(country) %>%
  summarize(conversion_rate = mean(converted))

data_country
```

```
## # A tibble: 4 x 2
##   country conversion_rate
##   <fct>             <dbl>
## 1 UK               0.0526
## 2 US               0.0378
## 3 China            0.00133
## 4 Germany          0.0625
```

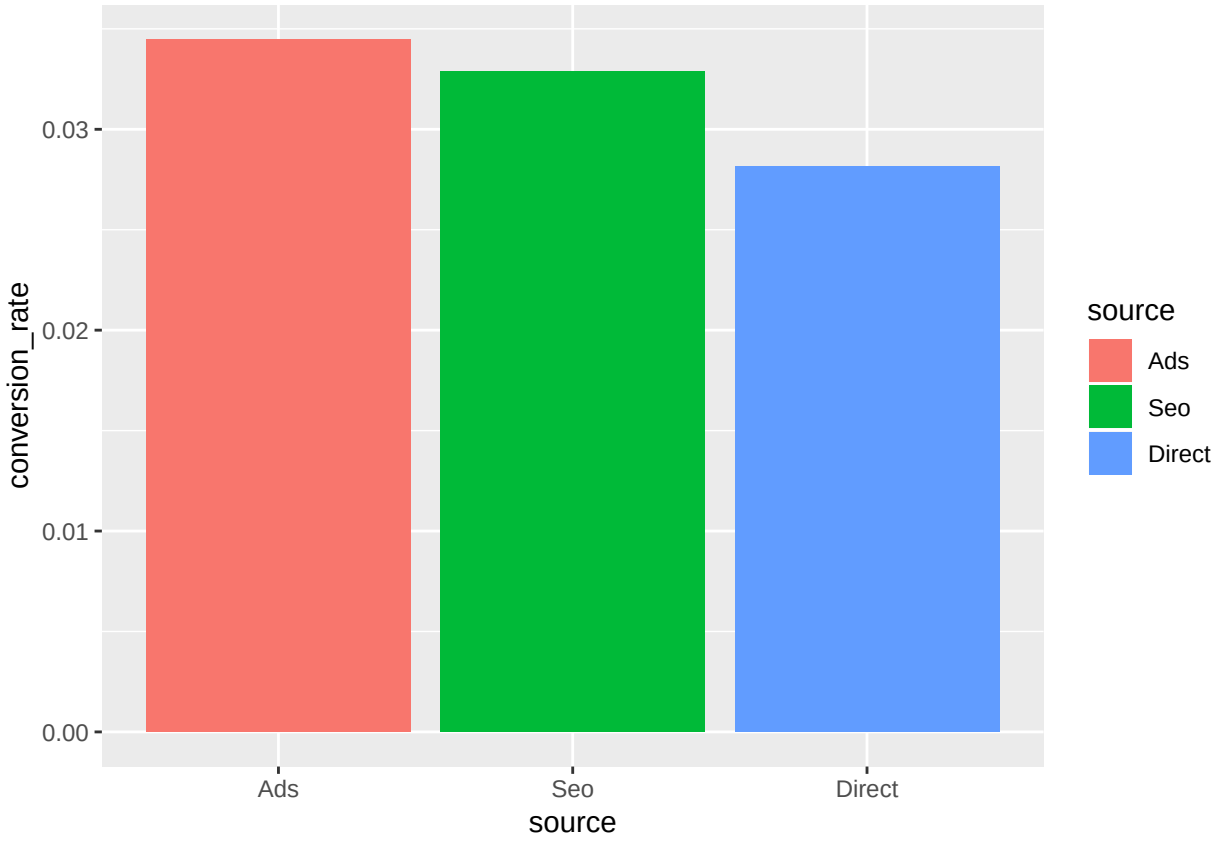
```
ggplot(data_country, aes(x=country, y=conversion_rate, fill = country))+  geom_col()
```




```
data_source <- conversion_data %>%
  group_by(source) %>%
  summarize(conversion_rate = mean(converted))

data_source
```

```
## # A tibble: 3 x 2
##   source conversion_rate
##   <fct>            <dbl>
## 1 Ads             0.0345
## 2 Seo             0.0329
## 3 Direct          0.0282
```

```
ggplot(data_source, aes(x=source, y=conversion_rate, fill = source))+  geom_col()
```



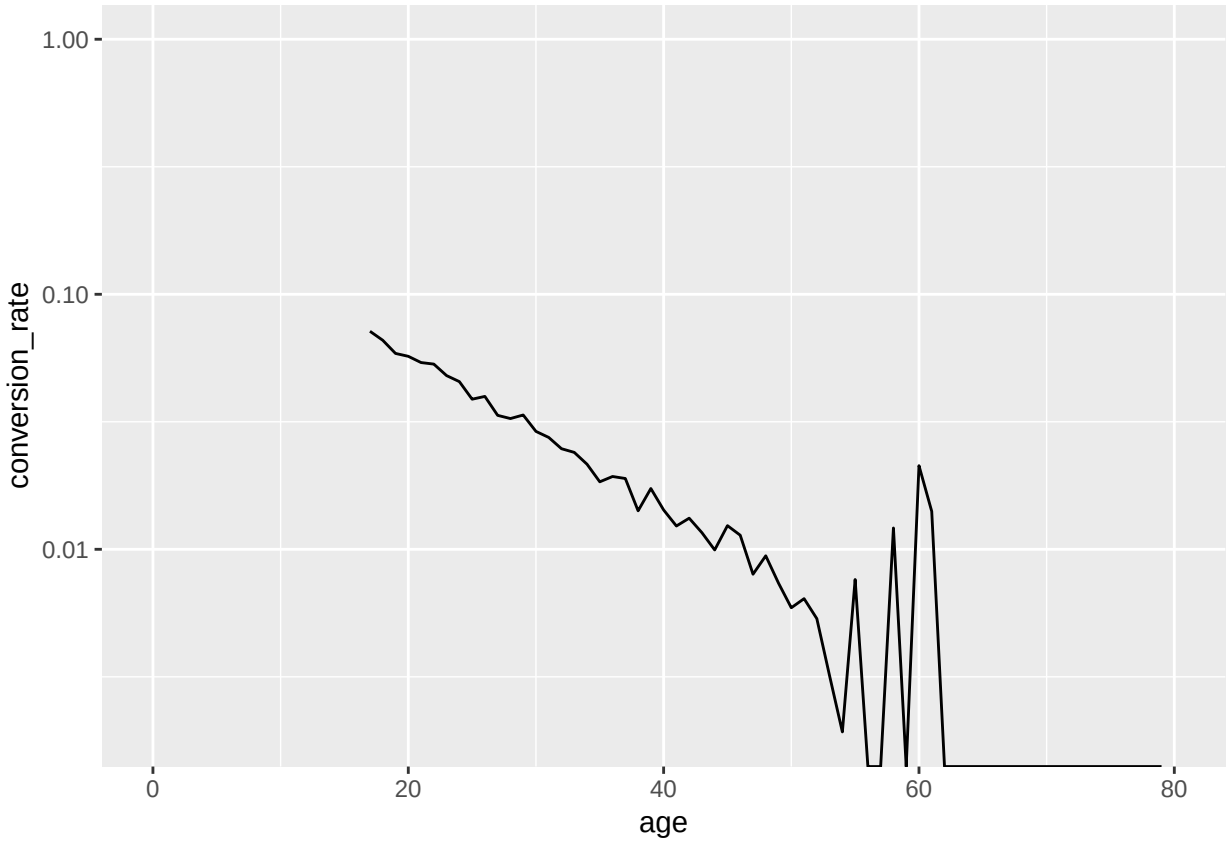

```
data_age <- conversion_data %>%
  group_by(age) %>%
  summarize(conversion_rate = mean(converted))

data_age
```

```
## # A tibble: 60 x 2
##      age conversion_rate
##    <int>           <dbl>
##  1    17          0.0716
##  2    18          0.0660
##  3    19          0.0586
##  4    20          0.0571
##  5    21          0.0540
##  6    22          0.0532
##  7    23          0.0480
##  8    24          0.0454
##  9    25          0.0388
## 10    26          0.0398
## # ... with 50 more rows
```

```
ggplot(data_age, aes(x=age, y=conversion_rate))+  geom_line() + xlim(0,80) + scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

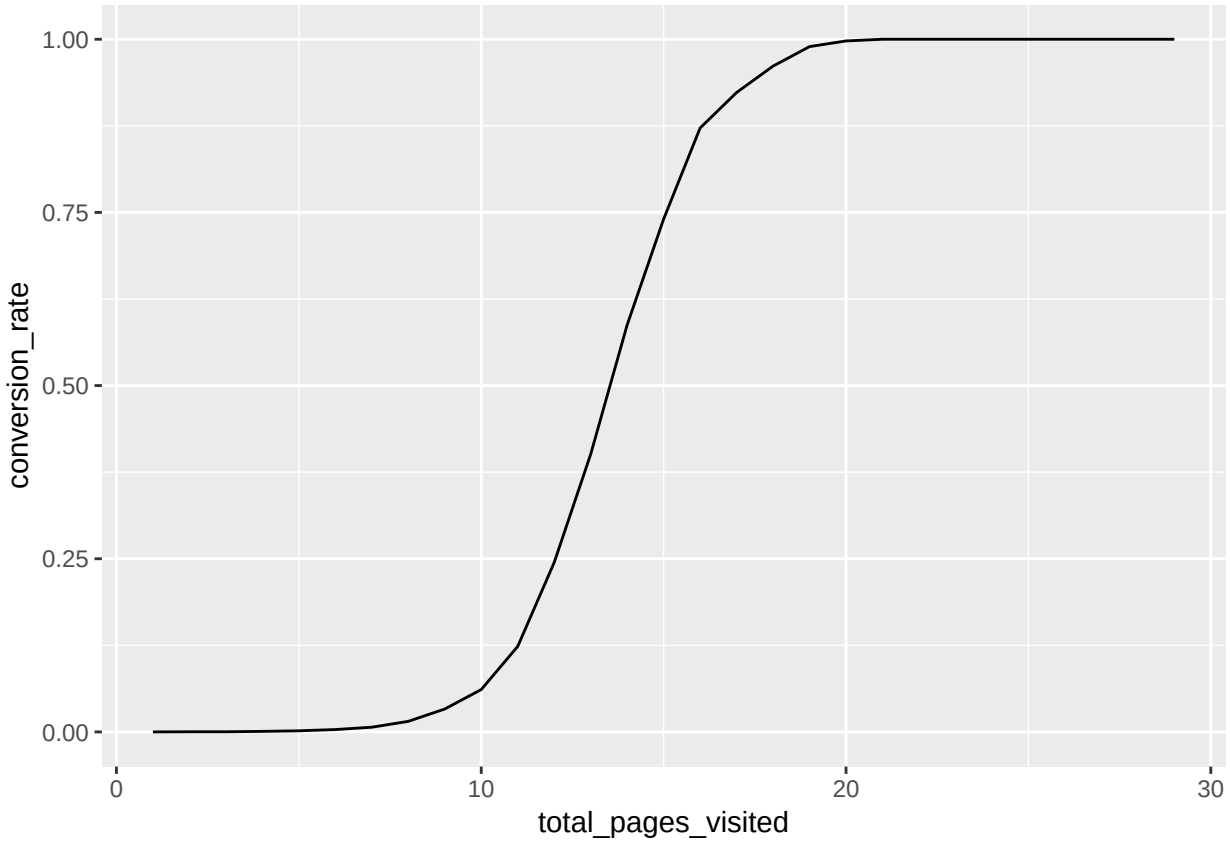
```
## Warning: Removed 2 rows containing missing values (geom_path).
```




```
data_pages <- conversion_data %>%
  group_by(total_pages_visited) %>%
  summarize(conversion_rate = mean(converted))

data_pages
```

```
## # A tibble: 29 x 2
##    total_pages_visited conversion_rate
##                  <int>           <dbl>
##  1                   1        0
##  2                   2        0.000228
##  3                   3        0.000251
##  4                   4        0.000780
##  5                   5        0.00157
##  6                   6        0.00344
##  7                   7        0.00677
##  8                   8        0.0152
##  9                   9        0.0331
## 10                  10        0.0612
## # ... with 19 more rows
```

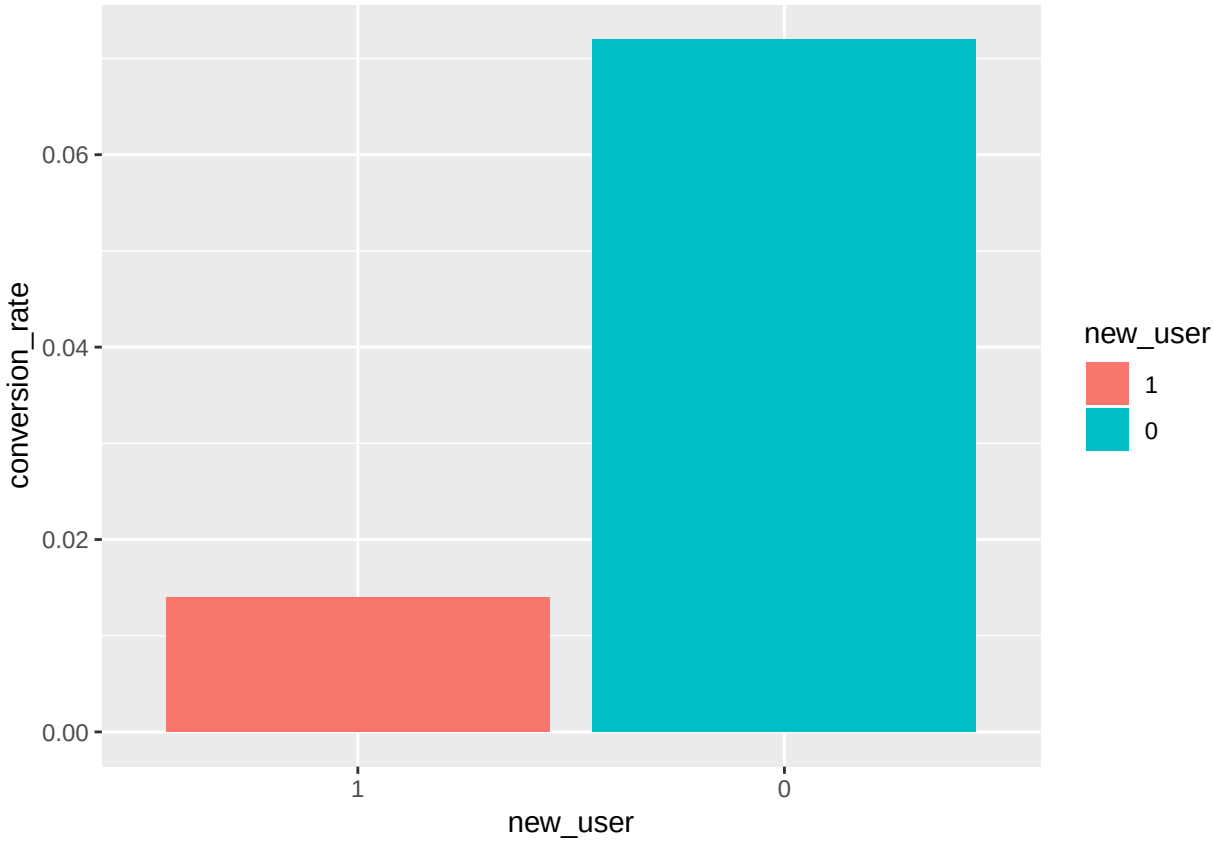
```
ggplot(data_pages, aes(x=total_pages_visited, y=conversion_rate))+  geom_line()
```



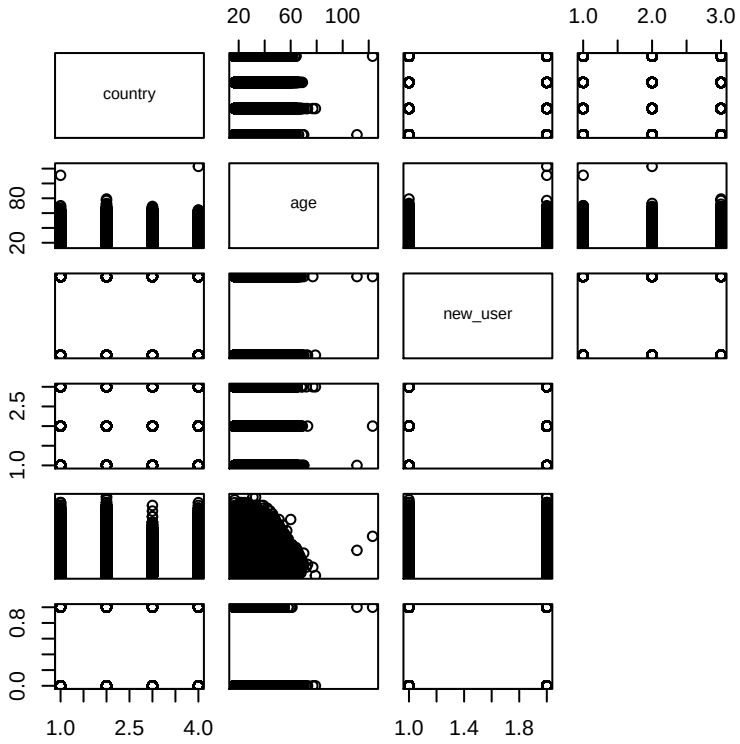

```
data_user <- conversion_data %>%
  group_by(new_user) %>%
  summarize(conversion_rate = mean(converted))

data_user
```

```
## # A tibble: 2 x 2
##   new_user conversion_rate
##   <fct>              <dbl>
## 1 1                 0.0140
## 2 0                 0.0720
```

```
ggplot(data_user, aes(x=new_user, y=conversion_rate, fill = new_user))+  geom_col()
```




```
pairs(conversion_data)
```



Taking care of abnormal values

I see that some ages are impossible to achieve (eg. 123), therefore, we have to remove these values. I have decided to remove these observations.

```
#Find the quantity of abnormal users
subset(conversion_data, age>79)
```

```
## # A tibble: 2 x 6
##   country   age new_user source total_pages_visited converted
##   <fct>   <int> <fct>    <fct>                <int>     <int>
## 1 Germany   123 0        Seo                     15         1
## 2 UK        111 0        Ads                     10         1
```

```
#filter them out
conversion_clean <- conversion_data %>%
  filter(age < 80)

conversion_clean
```

```
## # A tibble: 316,198 x 6
##    country   age new_user source total_pages_visited converted
##    <fct>   <int> <fct>    <fct>                <int>     <int>
##  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
##  7 China      27 1        Seo                      4         0
##  8 US         23 0        Ads                      4         0
##  9 UK         29 0        Direct                   4         0
## 10 US         25 0        Ads                      2         0
## # ... with 316,188 more rows
```

Recode Categorical Variables for modelling

```
conversion_wide <- conversion_clean %>%
  mutate(uk = recode(country, "UK" = 1,
                                .default = 0),
         usa = recode(country, "US" = 1,
                                .default = 0),
         china = recode(country, "China" = 1,
                                .default = 0),
         ads = recode(source, "Ads" = 1,
                                .default = 0),
         seo = recode(source, "Seo" = 1,
                                .default = 0)) %>%
  select(-country,-source)

conversion_wide
```

```
## # A tibble: 316,198 x 9
##      age new_user total_pages_vis~ converted    uk   usa china   ads   seo
##    <int> <fct>               <int>     <int> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1    25 1                       1         0     1     0     0     1     0
##  2    23 1                       5         0     0     1     0     0     1
##  3    28 1                       4         0     0     1     0     0     1
##  4    39 1                       5         0     0     0     1     0     1
##  5    30 1                       6         0     0     1     0     0     1
##  6    31 0                       1         0     0     1     0     0     1
##  7    27 1                       4         0     0     0     1     0     1
##  8    23 0                       4         0     0     1     0     1     0
##  9    29 0                       4         0     1     0     0     0     0
## 10    25 0                       2         0     0     1     0     1     0
## # ... with 316,188 more rows
```

Separate the data into training and testing sets

```
## Train with 75% of the sample size
smp_size <- floor(0.75 * nrow(conversion_wide))

set.seed(123)

train_ind <- sample(seq_len(nrow(conversion_wide)), size = smp_size)

train <- conversion_wide[train_ind, ]
test <- conversion_wide[-train_ind, ]

train
```

```
## # A tibble: 237,148 x 9
##      age new_user total_pages_vis~ converted    uk   usa china   ads   seo
##    <int> <fct>               <int>     <int> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1    33 1                       5         0     0     1     0     0     1
##  2    28 0                      12         1     0     1     0     0     0
##  3    57 1                       3         0     1     0     0     0     0
##  4    43 0                       7         0     0     1     0     0     1
##  5    27 1                       5         0     0     1     0     0     1
##  6    36 1                       8         0     0     0     1     1     0
##  7    27 1                       3         0     0     1     0     0     0
##  8    32 1                       2         0     0     1     0     1     0
##  9    17 0                       6         0     0     1     0     0     1
## 10    38 0                       9         0     0     1     0     1     0
## # ... with 237,138 more rows
```

```
test
```

```
## # A tibble: 79,050 x 9
##      age new_user total_pages_vis~ converted    uk   usa china   ads   seo
##    <int> <fct>               <int>     <int> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1    27 1                       3         0     0     0     1     1     0
##  2    24 1                       4         0     0     1     0     0     1
##  3    25 0                       3         0     0     0     1     0     1
##  4    24 1                       3         0     0     1     0     0     1
##  5    50 1                       6         0     0     0     1     0     1
##  6    23 1                       2         0     0     1     0     0     1
##  7    38 0                       5         0     0     1     0     0     1
##  8    35 0                       7         0     0     0     1     0     1
##  9    17 0                       1         0     0     1     0     0     1
## 10    27 1                       8         0     0     1     0     0     1
## # ... with 79,040 more rows
```

Logistic Regression

Because the response variable in this case (converted) is a categorical variable and the problem at hand is one of prediction, logistic regression is suitable for the matter.

```
model <- glm(converted ~.,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = converted ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9361  -0.0632  -0.0242  -0.0098   4.4109
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -8.429760   0.135819 -62.066  < 2e-16 ***
## age                 -0.072255   0.002742 -26.352  < 2e-16 ***
## new_user0            1.749745   0.041262  42.405  < 2e-16 ***
## total_pages_visited  0.758161   0.007189 105.468  < 2e-16 ***
## uk                  -0.276525   0.084902  -3.257 0.001126 **
```

```
## usa                 -0.643097   0.078262  -8.217  < 2e-16 ***
## china               -3.812302   0.149203 -25.551  < 2e-16 ***
## ads                  0.213031   0.056242   3.788 0.000152 ***
## seo                  0.123534   0.051412   2.403 0.016270 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 67175  on 237147  degrees of freedom
## Residual deviance: 19148  on 237139  degrees of freedom
## AIC: 19166
##
## Number of Fisher Scoring iterations: 10
```

```
anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: converted
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                237147      67175
## age                  1     2048    237146      65127 < 2.2e-16 ***
## new_user             1     4928    237145      60200 < 2.2e-16 ***
## total_pages_visited  1    39428    237144      20772 < 2.2e-16 ***
## uk                   1      183    237143      20589 < 2.2e-16 ***
## usa                  1      514    237142      20075 < 2.2e-16 ***
## china                1      913    237141      19162 < 2.2e-16 ***
## ads                  1        9    237140      19154  0.003341 **
## seo                  1        6    237139      19148  0.015904 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All of the variables have statistical significance. Therefore, we do not have to perform feature selection.

Predicting Conversions and Model Accuracy

```
fitted_results <- predict(model,test)

#Classifying the probabilities
fitted_results <- ifelse(fitted_results >= 0.5,1,0)

error <- mean(fitted_results != test$converted)
error
```

```
## [1] 0.0144845
```

```
accuracy = 1 - error
accuracy
```

```
## [1] 0.9855155
```

```
#Plot the confusion matrix
table(predicted = fitted_results,actual = test$converted)
```

```
##          actual
## predicted     0     1
##         0 76261   965
##         1   180  1644
```

Final insights and recommendations

The company should consider increasing targeted marketing for younger users, users in Germany and existing users due to the higher conversion rate for these specific groups.

People who are visiting a lot of pages also tend to have higher conversion rates. This is an indicator of intent to purchase. The company could also deploy targetted marketing strategies towards someone who has visited a lot of pages but has not made a purchase by sending them special offers on the products that have been viewed a lot by the customer.

China is doing exceptionally poor with regards to conversion rate. There could be a problem with translation issues as all of the other factors (marketing source) are relatively equally spread amongst countries. Finding out why China is doing so poorly and fixing the issue that was previously unrecognised could potentially significantly improve conversion rate and company performance in China.