# Shopping in VR

Christopher Davidson

2022-09-25

First of all I would like to thank the people who gave me this opportunity to prove my skills a data analyst and be able to showcase them. This is a Markdown document that will show all of the processes I went through whilst programming in R. Later, I exported the dataframe's I produced in R into PowerBI to be able to visualise the data in an easy way.

## Intro

I was asked to do a case study into Users using VR to shop. I was given a CSV containing the data from the time when these users were in VR. The CSV file does not contain time stamps when people used VR, however it does contain the amount of time users spent looking at certain products in VR. This was useful to understand how interested they were in a certain product. I am to work in this fictional team consisting of the Product Owner, a UX designer, and another data analyst. Meaning that I will have to present my findings in a way that everyone understands.

I used R to prepare and analyse the data but then used PowerBI to visualise the data later. I thought it would be easier this way.

## Ask

In this setup, I was already given the questions. They are as follows:

1. Who is the main target group? Which segments do you identify?
2. What kind of data would I want to improve my analysis and back-up the insights I mentioned and why?
3. The team wants to develop new features that is personalised for each target market. Which target market should they focus on first?
4. Some users recorded whether they had children or not and others did not. The team is wanting to increase children product sales. They want to know which characteristics a user has that shows that they have children.

## Process

The data collected was already very clean after initially looking at it through dplyr's glimpse and skimr's skim_without_charts. Nothing was coming out as unusual.

```
glimpse(CustomerData)
```

```
## Rows: 537,577
## Columns: 12
## $ CustomerID    <dbl> 1000001, 1000001, 1000001, 1000001, 1000002, 1000003, 10~
## $ ItemID        <chr> "P00069042", "P00248942", "P00087842", "P00085442", "P00~
## $ Sex           <chr> "F", "F", "F", "F", "M", "M", "M", "M", "M", "M", "M", "~
## $ Age           <chr> "0-17", "0-17", "0-17", "0-17", "55+", "26-35", "46-50",~
## $ Profession    <dbl> 10, 10, 10, 10, 16, 15, 7, 7, 7, 20, 20, 20, 20, 20, 9, ~
## $ CityType      <chr> "A", "A", "A", "A", "C", "A", "B", "B", "B", "A", "A", "~
## $ YearsInCity   <chr> "2", "2", "2", "2", "4+", "3", "2", "2", "2", "1", "1", ~
## $ HaveChildren  <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TR~
## $ ItemCategory1 <dbl> 3, 1, 12, 12, 8, 1, 1, 1, 1, 8, 5, 8, 8, 1, 5, 4, 2, 5, ~
## $ ItemCategory2 <dbl> NA, 6, NA, 14, NA, 2, 8, 15, 16, NA, 11, NA, NA, 2, 8, 5~
## $ ItemCategory3 <dbl> NA, 14, NA, NA, NA, NA, 17, NA, NA, NA, NA, NA, NA, 5, 1~
## $ Amount        <dbl> 8370, 15200, 1422, 1057, 7969, 15227, 19215, 15854, 1568~
```

```
skim_without_charts(CustomerData)
```

Table 1: Data summary

| | |
|---|---|
| Name | CustomerData |
| Number of rows | 537577 |
| Number of columns | 12 |
| Column type frequency: | |
| character | 5 |
| logical | 1 |
| numeric | 6 |
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| ItemID | 0 | 1 | 8 | 9 | 0 | 3623 | 0 |
| Sex | 0 | 1 | 1 | 1 | 0 | 2 | 0 |
| Age | 0 | 1 | 3 | 5 | 0 | 7 | 0 |
| CityType | 0 | 1 | 1 | 1 | 0 | 3 | 0 |
| YearsInCity | 0 | 1 | 1 | 2 | 0 | 5 | 0 |

**Variable type: logical**

| skim_variable | n_missing | complete_rate | mean | count |
|---|---|---|---|---|
| HaveChildren | 20170 | 0.96 | 0.41 | FAL: 304366, TRU: 213041 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 |
|---|---|---|---|---|---|---|---|---|---|
| CustomerID | 0 | 1.00 | 1002991.85 | 1714.39 | 1000001 | 1001495 | 1003031 | 1004417 | 1006040 |

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 |
|---|---|---|---|---|---|---|---|---|---|
| Profession | 0 | 1.00 | 8.08 | 6.52 | 0 | 2 | 7 | 14 | 20 |
| ItemCategory1 | 0 | 1.00 | 5.30 | 3.75 | 1 | 1 | 5 | 8 | 18 |
| ItemCategory2 | 166986 | 0.69 | 9.84 | 5.09 | 2 | 5 | 9 | 15 | 18 |
| ItemCategory3 | 373299 | 0.31 | 12.67 | 4.12 | 3 | 9 | 14 | 16 | 18 |
| Amount | 0 | 1.00 | 9333.86 | 4981.02 | 185 | 5866 | 8062 | 12073 | 23961 |

After checking that there wasn't anything unusual. I decided to continue with two cleaning functions just in case.

```
clean_names(CustomerData) #This is used to ensure that all the columns names are compatible for R to un
```

```
## # A tibble: 537,577 x 12
##    custome~1 item_id sex   age   profe~2 city_~3 years~4 have_~5 item_~6 item_~7
##        <dbl> <chr>   <chr> <chr>   <dbl> <chr>   <chr>   <lgl>     <dbl>   <dbl>
##  1   1000001 P00069~ F     0-17       10 A       2       FALSE         3      NA
##  2   1000001 P00248~ F     0-17       10 A       2       FALSE         1       6
##  3   1000001 P00087~ F     0-17       10 A       2       FALSE        12      NA
##  4   1000001 P00085~ F     0-17       10 A       2       FALSE        12      14
##  5   1000002 P00285~ M     55+        16 C       4+      FALSE         8      NA
##  6   1000003 P00193~ M     26-35      15 A       3       FALSE         1       2
##  7   1000004 P00184~ M     46-50       7 B       2       TRUE          1       8
##  8   1000004 P00346~ M     46-50       7 B       2       TRUE          1      15
##  9   1000004 P00972~ M     46-50       7 B       2       TRUE          1      16
## 10   1000005 P00274~ M     26-35      20 A       1       TRUE          8      NA
## # ... with 537,567 more rows, 2 more variables: item_category3 <dbl>,
## #   amount <dbl>, and abbreviated variable names 1: customer_id, 2: profession,
## #   3: city_type, 4: years_in_city, 5: have_children, 6: item_category1,
## #   7: item_category2
```

```
get_dupes(CustomerData) #This ensures that none of the rows are duplicated and eliminates any that are
```

```
## No variable names specified - using all columns.
```

```
## No duplicate combinations found of: CustomerID, ItemID, Sex, Age, Profession, CityType, YearsInCity,
```

```
## # A tibble: 0 x 13
## # ... with 13 variables: CustomerID <dbl>, ItemID <chr>, Sex <chr>, Age <chr>,
## #   Profession <dbl>, CityType <chr>, YearsInCity <chr>, HaveChildren <lgl>,
## #   ItemCategory1 <dbl>, ItemCategory2 <dbl>, ItemCategory3 <dbl>,
## #   Amount <dbl>, dupe_count <int>
```

### Analyse

Afterwards it was time to analyse and this took some time. First, I was wanting to look at the data I had available. Each header telling me what was contained in that column.

```
glimpse(CustomerData)
```

```
## Rows: 537,577
## Columns: 12
## $ CustomerID    <dbl> 1000001, 1000001, 1000001, 1000001, 1000002, 1000003, 10~
## $ ItemID        <chr> "P00069042", "P00248942", "P00087842", "P00085442", "P00~
## $ Sex           <chr> "F", "F", "F", "F", "M", "M", "M", "M", "M", "M", "M", "~
## $ Age           <chr> "0-17", "0-17", "0-17", "0-17", "55+", "26-35", "46-50",~
## $ Profession    <dbl> 10, 10, 10, 10, 16, 15, 7, 7, 7, 20, 20, 20, 20, 20, 9, ~
## $ CityType      <chr> "A", "A", "A", "A", "C", "A", "B", "B", "B", "A", "A", "~
## $ YearsInCity   <chr> "2", "2", "2", "2", "4+", "3", "2", "2", "2", "1", "1", ~
## $ HaveChildren  <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TR~
## $ ItemCategory1 <dbl> 3, 1, 12, 12, 8, 1, 1, 1, 1, 8, 5, 8, 8, 1, 5, 4, 2, 5, ~
## $ ItemCategory2 <dbl> NA, 6, NA, 14, NA, 2, 8, 15, 16, NA, 11, NA, NA, 2, 8, 5~
## $ ItemCategory3 <dbl> NA, 14, NA, NA, NA, NA, 17, NA, NA, NA, NA, NA, NA, 5, 1~
## $ Amount        <dbl> 8370, 15200, 1422, 1057, 7969, 15227, 19215, 15854, 1568~
```

I figured out what I wanted to do. I wanted to look at the different type of users and what they were up to. That included sepearate dataframes for Age, Sex, Profession,children ,City, and the amount of time lived in that city. Whether that made a difference or not. I knew later I would want to go deeper into analysing users with children but for now I looked at the basics.

```
"Age" <- `CustomerData` %>% #Have a look to see which age group is using it most
  group_by(`Age`) %>%
  summarise(AgeCount =  n_distinct(CustomerID))
print(Age)
```

```
## # A tibble: 7 x 2
##   Age   AgeCount
##   <chr>    <int>
## 1 0-17       218
## 2 18-25     1069
## 3 26-35     2053
## 4 36-45     1167
## 5 46-50      531
## 6 51-55      481
## 7 55+        372
```

I realised at this point that numbers were so low, so I just wanted to make sure that the data I was looking at was correct.

```
`CustomerData` %>% #Only 5891 customers out of 537,577 rows. I'll double check these numbers with the s
  summarise(CustCount = n_distinct(CustomerID))
```

```
## # A tibble: 1 x 1
##   CustCount
##       <int>
## 1      5891
```

There was only 5891 customers out of 537,577 rows. I double checked the data using the =COUNTUNIQUE function in sheets and found that I got the same number. It was correct, there was just under 6000 people who tested this product.

From the data, it was also clear that **it was mostly 26-35 year olds that wanted to use VR**. You can see this more clearly in the report I created in PowerBI showing all the visuals.

```r
"AgeTime" <- `CustomerData` %>% #The older people get, the longer they look at the article
  group_by(`Age`) %>%
  summarise(AvgTime = mean(Amount))
print(AgeTime)
```

```
## # A tibble: 7 x 2
##   Age   AvgTime
##   <chr>   <dbl>
## 1 0-17    9020.
## 2 18-25   9235.
## 3 26-35   9315.
## 4 36-45   9401.
## 5 46-50   9285.
## 6 51-55   9621.
## 7 55+     9454.
```

This was interesting. **The older people got, the more time they would spend looking at individual products.**

```r
"Sex" <- `CustomerData` %>% #Over double the amount of people using this VR experience are male and men
  group_by(`Sex`) %>%
  summarise(SexCount =  n_distinct(CustomerID),AvgTime = mean(Amount)) %>%
  mutate(CountPerc = SexCount / sum(SexCount)*100) %>%
  mutate(TimePerc = AvgTime / sum(AvgTime)*100)
print(Sex)
```

```
## # A tibble: 2 x 5
##   Sex   SexCount AvgTime CountPerc TimePerc
##   <chr>    <int>   <dbl>     <dbl>    <dbl>
## 1 F         1666   8810.      28.3     48.1
## 2 M         4225   9505.      71.7     51.9
```

I decided to put a percentage in this one that would compare the sum of the column with the smaller part. Showing that around **70% of the users using VR are men and just under 30% were female**. **Men even spent a little longer on average on looking at the product than women.**

```r
"Prof" <-`CustomerData` %>%
  group_by(`Profession`) %>%
  summarise(Count = n_distinct(CustomerID), AvgTime = (mean(Amount)))%>%
  mutate(CountPerc = (Count / sum(Count)*100)) %>%
  mutate(TimePerc = (AvgTime / sum(AvgTime)*100)) %>%
  arrange(desc(CountPerc))
print(Prof)
```

```
## # A tibble: 21 x 5
##    Profession Count AvgTime CountPerc TimePerc
##         <dbl> <int>   <dbl>     <dbl>    <dbl>
##  1          4   740   9279.     12.6      4.74
##  2          0   688   9187.     11.7      4.70
##  3          7   669   9502.     11.4      4.86
##  4          1   517   9018.      8.78     4.61
```

```
## 5          17   491   9906.       8.33      5.06
## 6          12   376   9883.       6.38      5.05
## 7          14   294   9569.       4.99      4.89
## 8          20   273   8881.       4.63      4.54
## 9           2   256   9026.       4.35      4.61
## 10         16   235   9457.       3.99      4.84
## # ... with 11 more rows
```

This was a large file, but generally I could see that there were **people with certain professions who would be more interested in VR.** However, *the dataset was so small and it was a test run in the stores, there can be other contributing factors at play. These could include people with shift work not being able to make it on the day the store is showcasing VR or the other way around. Perhaps there could be a lot of co-workers who work in the store and surrounding stores whoc used the VR on their breaks.* If I was to look into this again, I would want to know the types of professions these people do and see if the data coincides with professional interests. For example, 3D designers could be interested in VR because the want to look at the technology progression so they could perhaps buy a VR headset for work. Or perhaps game creators are just naturally interested in the technology.

```
"City" <- `CustomerData` %>%
  group_by(`CityType`) %>%
  summarise(Count = n_distinct(CustomerID), AvgTime = (mean(Amount)))
print(City)
```

```
## # A tibble: 3 x 3
##   CityType Count AvgTime
##   <chr>    <int>   <dbl>
## 1 A         1045   8958.
## 2 B         1707   9199.
## 3 C         3139   9844.
```

If this was a real study, I would ask what they City Types represent. Whether that be particular cities, city density and so on. Or if it is something else. However, the data clearly shows that **people are a lot more interested in VR in type C.** However, the tricky part is that we don't know details. *Details we would want to understand this would be what City Types represent, the amount of time VR was shown in these cities, the amount of staff available to assist customers with VR and so on.*

```
"Local" <-`CustomerData` %>%
  group_by(`YearsInCity`) %>%
  summarise(CountA = n_distinct(ifelse(CityType == "A",CustomerID, NA),na.rm = T),AvgTimeA = mean(ifels
print(Local)
```

```
## # A tibble: 5 x 9
##   YearsInCity CountA AvgTimeA CountB AvgTimeB CountC AvgTimeC TCount TAvgTime
##   <chr>        <int>    <dbl>  <int>    <dbl>  <int>    <dbl>  <int>    <dbl>
## 1 0              147 1002916.    211 1003020.    414 1003140.    772    9247.
## 2 1              370 1002915.    608 1003130.   1108 1003006.   2086    9320.
## 3 2              183 1003183.    342 1003059.    620 1002960.   1145    9398.
## 4 3              180 1002492.    295 1002920.    504 1003158.    979    9351.
## 5 4+             165 1002972.    251 1002918.    493 1002858.    909    9346.
```

Next was just a little bit more complicated. I wanted to check the time that each user had lived in the city and how that was to effect the likelihood of them wanting to try VR. As you can see from the data frame or in the PowerBI report, you can clearly see that **people who have lived in the city for more than a year are more likely to try VR.** I know that this is not a report to speculate in, however, this probably is due to people who have just moved in are looking to settle so too busy, people who are 2+ years area already settled so not looking for any new experiences. Each of the counts with letters represent different cities and from the data we can see that it doesn't matter which city you live in, you are still more likely to try VR if you have lived in that city for 1-2 years.

```
"TKids" <- `CustomerData` %>%
  group_by(`HaveChildren`) %>%
  drop_na(HaveChildren) %>%
  summarise(Count = n_distinct(CustomerID), AvgTime = (mean(Amount))) %>%
  mutate(CountPerc = (Count / sum(Count)*100)) %>%
  mutate(TimePerc = (AvgTime / sum(AvgTime)*100))
print(TKids)
```

```
## # A tibble: 2 x 5
##   HaveChildren Count AvgTime CountPerc TimePerc
##   <lgl>        <int>   <dbl>     <dbl>    <dbl>
## 1 FALSE         3280   9334.      57.8     50.0
## 2 TRUE          2399   9332.      42.2     50.0
```

Now here comes the simple question of how many people who used VR have kids. I calculated the percentage on the total sum of the other columns since it would give me a more accurate reading.It is clear that just **under 60% of people who used VR didn't have children and just over 40% did.**

```
"KidCount" <- `CustomerData` %>% #Tried with item ID, ended up with 3623 rows, so trying with category
  group_by(`ItemCategory1`) %>%
  summarise(WithKids = n_distinct(ifelse(HaveChildren == TRUE,CustomerID, NA),na.rm = T),WOKids = n_dis
  mutate(WithKidsPerc = WithKids / sum(WithKids)*100) %>%
  mutate(WOKidsPerc = WOKids / sum(WOKids)*100) %>%
  mutate(KidsPercDif = WithKidsPerc-WOKidsPerc) %>%
  arrange(KidsPercDif)
print(KidCount)
```

```
## # A tibble: 18 x 6
##    ItemCategory1 WithKids WOKids WithKidsPerc WOKidsPerc KidsPercDif
##            <dbl>    <int>  <int>        <dbl>      <dbl>       <dbl>
##  1             3     1489   2198        7.00       7.51      -0.518
##  2            11     1418   2024        6.66       6.92      -0.257
##  3             2     1713   2417        8.05       8.26      -0.214
##  4            15      967   1380        4.54       4.72      -0.174
##  5             4     1337   1870        6.28       6.39      -0.111
##  6             9      156    235        0.733      0.803     -0.0704
##  7             1     2331   3224       11.0       11.0       -0.0697
##  8            16     1252   1740        5.88       5.95      -0.0660
##  9            13      910   1257        4.28       4.30      -0.0217
## 10             6     1654   2261        7.77       7.73       0.0416
## 11             7      595    805        2.80       2.75       0.0436
## 12             5     2345   3193       11.0       10.9        0.102
## 13            14      408    524        1.92       1.79       0.126
## 14             8     2313   3139       10.9       10.7        0.136
## 15            18      538    689        2.53       2.36       0.172
## 16            10      968   1260        4.55       4.31       0.241
```

```
## 17              17      202     206         0.949       0.704       0.245
## 18              12      686     827         3.22        2.83        0.396
```

This is something I didn't use in my final report since it is showing a **0.5% difference between people with and without children.** I considered this statistically insignificant and thought this data isn't useful.

```
"KidTime" <- `CustomerData` %>% #I'm not comfortable with these variables. These numbers are too tight.
  group_by(`ItemCategory1`) %>%
  summarise(WithKids = mean(ifelse(HaveChildren == TRUE,Amount, NA),na.rm = T),WOKids = mean(ifelse(Have
  mutate(WithKidsPerc = WithKids / (WithKids+WOKids)*100) %>%
  mutate(WOKidsPerc = WOKids / (WithKids+WOKids)*100) %>%
  mutate(KidsPercDif = WithKidsPerc-WOKidsPerc) %>% #Something has gone wrong with this difference, how
  arrange(KidsPercDif)
print(KidTime)
```

```
## # A tibble: 18 x 6
##    ItemCategory1 WithKids WOKids WithKidsPerc WOKidsPerc KidsPercDif
##            <dbl>    <dbl>  <dbl>        <dbl>      <dbl>       <dbl>
##  1             9   15064. 15919.         48.6       51.4      -2.76
##  2            11    4614.  4721.         49.4       50.6      -1.14
##  3            18    2950.  3001.         49.6       50.4      -0.861
##  4            13     720.   725.         49.8       50.2      -0.383
##  5             6   15811. 15867.         49.9       50.1      -0.177
##  6             7   16317. 16364.         49.9       50.1      -0.145
##  7            10   19703. 19663.         50.1       49.9       0.100
##  8            12    1352.  1348.         50.1       49.9       0.154
##  9            15   14790. 14744.         50.1       49.9       0.154
## 10             1   13640. 13596.         50.1       49.9       0.161
## 11            16   14802. 14736.         50.1       49.9       0.223
## 12             8    7521.  7479.         50.1       49.9       0.279
## 13             5    6260.  6222.         50.2       49.8       0.305
## 14            17   10148. 10076.         50.2       49.8       0.356
## 15            14   13188. 13082.         50.2       49.8       0.406
## 16             2   11360. 11178.         50.4       49.6       0.806
## 17             3   10198. 10010.         50.5       49.5       0.927
## 18             4    2358.  2305.         50.6       49.4       1.14
```

```
"KidTime2" <- `CustomerData` %>%
  group_by(`ItemCategory2`) %>%
  summarise(WithKids = mean(ifelse(HaveChildren == TRUE,Amount, NA),na.rm = T),WOKids = mean(ifelse(Have
  mutate(WithKidsPerc = WithKids / (WithKids+WOKids)*100) %>%
  mutate(WOKidsPerc = WOKids / (WithKids+WOKids)*100) %>%
  mutate(KidsPercDif = WithKidsPerc-WOKidsPerc) %>%
  arrange(KidsPercDif)
print(KidTime2)
```

```
## # A tibble: 18 x 6
##    ItemCategory2 WithKids WOKids WithKidsPerc WOKidsPerc KidsPercDif
##            <dbl>    <dbl>  <dbl>        <dbl>      <dbl>       <dbl>
##  1             7    6791.  6917.         49.5       50.5      -0.922
##  2            15   10264. 10419.         49.6       50.4      -0.754
##  3             8   10231. 10323.         49.8       50.2      -0.446
```

```
## 4              16   10256. 10333.          49.8       50.2     -0.371
## 5              10   15613. 15656.          49.9       50.1     -0.137
## 6              17    9407.  9402.          50.0       50.0      0.0268
## 7              14    7107.  7094.          50.0       50.0      0.0912
## 8               6   11534. 11503.          50.1       49.9      0.137
## 9               9    7314.  7273.          50.1       49.9      0.281
## 10              2   13678. 13600.          50.1       49.9      0.286
## 11             NA    7737.  7655.          50.3       49.7      0.535
## 12             18    9426.  9325.          50.3       49.7      0.537
## 13             12    7008.  6925.          50.3       49.7      0.591
## 14             11    9005.  8873.          50.4       49.6      0.739
## 15              4   10332. 10134.          50.5       49.5      0.967
## 16              3   11364. 11140.          50.5       49.5      0.992
## 17              5    9157.  8969.          50.5       49.5      1.04
## 18             13    9879.  9534.          50.9       49.1      1.77
```

```r
"KidTime3" <-`CustomerData` %>%
  group_by(`ItemCategory3`) %>%
  summarise(WithKids = mean(ifelse(HaveChildren == TRUE,Amount, NA),na.rm = T),WOKids = mean(ifelse(Have
  mutate(WithKidsPerc = WithKids / (WithKids+WOKids)*100) %>%
  mutate(WOKidsPerc = WOKids / (WithKids+WOKids)*100) %>%
  mutate(KidsPercDif = WithKidsPerc-WOKidsPerc) %>%
  arrange(KidsPercDif)
print(KidTime3)
```

```
## # A tibble: 16 x 6
##    ItemCategory3 WithKids  WOKids WithKidsPerc WOKidsPerc KidsPercDif
##            <dbl>    <dbl>  <dbl>        <dbl>      <dbl>       <dbl>
## 1              3   13749. 14104.         49.4       50.6     -1.28
## 2             11   12029. 12198.         49.7       50.3     -0.698
## 3             14    9980. 10113.         49.7       50.3     -0.663
## 4              6   13129. 13236.         49.8       50.2     -0.403
## 5              8   13009. 13052.         49.9       50.1     -0.162
## 6              9   10433. 10438.         50.0       50.0     -0.0249
## 7             NA    8314.  8300.         50.0       50.0      0.0818
## 8             16   12017. 11956.         50.1       49.9      0.252
## 9             13   13221. 13146.         50.1       49.9      0.287
## 10            15   12393. 12316.         50.2       49.8      0.309
## 11            17   11832. 11748.         50.2       49.8      0.359
## 12            18   11052. 10936.         50.3       49.7      0.527
## 13             5   12226. 12067.         50.3       49.7      0.653
## 14             4    9867.  9730.         50.3       49.7      0.700
## 15            10   13667. 13372.         50.5       49.5      1.09
## 16            12    8861.  8648.         50.6       49.4      1.21
```

All these dataframes are looking at the different product catagories and explores the likelihood of people looking at these products categories when they have kids or not. It turns out that there wasn't a single catagory in any of Category types that really stood out as having one group or the other looking at those items more. **The most the difference even got was 2% in KidTime2 which I didn't consider enough to chase.**

```r
"KidTimeArt" <-`CustomerData` %>% #Tried with item ID, ended up with 3623 rows, so trying with category
  group_by(`ItemID`) %>%
  summarise(WithKids = mean(ifelse(HaveChildren == TRUE,(Amount), NA),na.rm = T),WOKids = mean(ifelse(Ha
  mutate(WithKidsPerc = (WithKids / (WithKids+WOKids)*100)) %>%
  mutate(WOKidsPerc = (WOKids / (WithKids+WOKids)*100)) %>%
  mutate(KidsPercDif = WithKidsPerc-WOKidsPerc)%>%
  mutate(Dif = WithKids-WOKids) %>%
  arrange(desc(KidsPercDif)) #I want to put this in Ascending to check the Null's
  #arrange(Dif) #I had to scroll down for the Nulls, however the Null's were so small, I thought I shou
#Success, the item with ID P00131842 was looked at 14159 more miliseconds by people with kids. I should
print(KidTimeArt)
```

```
## # A tibble: 3,623 x 7
##    ItemID    WithKids WOKids WithKidsPerc WOKidsPerc KidsPercDif    Dif
##    <chr>        <dbl>  <dbl>        <dbl>      <dbl>       <dbl>  <dbl>
##  1 P00175142    7989   2193          78.5       21.5        56.9  5796
##  2 P00077142    7866   2222          78.0       22.0        55.9  5644
##  3 P00309742    6096.  1729          77.9       22.1        55.8  4368.
##  4 P00161342   15262   4516          77.2       22.8        54.3 10746
##  5 P00131842   20468.  6308          76.4       23.6        52.9 14160.
##  6 P00138742   12779   4348          74.6       25.4        49.2  8431
##  7 P00261942   11425   3977          74.2       25.8        48.4  7448
##  8 P00293442     571.   200          74.1       25.9        48.1   371.
##  9 P00152342    8565   3010.         74.0       26.0        48.0  5554.
## 10 P00247842    6025   2195          73.3       26.7        46.6  3830
## # ... with 3,613 more rows
```

I instead looked at each individual article. **I found there were quite a few articles where people with kids would look at those longer than people without.** This could well predict whether people do have kids or not if we don't have that data.

```r
"KidAgeCount" <-`CustomerData` %>% #The older people get, the more likely they are to have kids
  group_by(`Age`) %>%
  summarise(CountKids = n_distinct(ifelse(HaveChildren == TRUE,CustomerID, NA),na.rm = T),CountWOKids =
  mutate(WithKidsPerc = (CountKids / (CountKids+CountWOKids))*100) %>%
  mutate(WOKidsPerc = (CountWOKids / (CountKids+CountWOKids))*100) %>%
  mutate(dif = WithKidsPerc - WOKidsPerc)
print(KidAgeCount)
```

```
## # A tibble: 7 x 6
##   Age   CountKids CountWOKids WithKidsPerc WOKidsPerc   dif
##   <chr>     <int>       <int>        <dbl>      <dbl> <dbl>
## 1 0-17          0         211          0          100  -100
## 2 18-25       239         785         23.3       76.7 -53.3
## 3 26-35       783        1198         39.5       60.5 -20.9
## 4 36-45       448         675         39.9       60.1 -20.2
## 5 46-50       363         152         70.5       29.5  41.0
## 6 51-55       334         128         72.3       27.7  44.6
## 7 55+         232         131         63.9       36.1  27.8
```

I then looked at people of differenct age groups and the liklihood of them having kids. It was immediately clear **The older the user gets, the more likely they are to have kids. Dropping off when we looked at people over the age of 55.** You can clearly see this in the report.

```r
"KidCityYearsCount" <-`CustomerData` %>% #There doesn't seem to be a difference when looking at years i
  group_by(`YearsInCity`) %>%
  summarise(CountKids = (n_distinct(ifelse(HaveChildren == TRUE,CustomerID, NA),na.rm = T)),CountWOKids
  mutate(WithKidsPerc = (CountKids / sum(CountKids))*100) %>%
  mutate(WOKidsPerc = (CountWOKids / sum(CountWOKids))*100) %>%
  mutate(dif = WithKidsPerc - WOKidsPerc)
print(KidCityYearsCount)
```

```
## # A tibble: 5 x 6
##   YearsInCity CountKids CountWOKids WithKidsPerc WOKidsPerc      dif
##   <chr>           <int>       <int>        <dbl>      <dbl>    <dbl>
## 1 0                 300         446         12.5       13.6 -1.09
## 2 1                 891        1120         37.1       34.1  2.99
## 3 2                 469         641         19.5       19.5  0.00713
## 4 3                 385         554         16.0       16.9 -0.842
## 5 4+                354         519         14.8       15.8 -1.07
```

The last dataframe was looking deeper into this idea of the users having children and whether the amount of time the user has lived in the city makes a difference as to whether they had children or not. The result was **a very small difference between years in the city and whether or not the people were more likely to have kids**. I was going this way because I was thinking whether users who have just moved to the city were more likely to be travellers who never settled. But I couldn't find any data to back that theory up.

### Share

I exported all the DataFrames as CSV files to be pulled into PowerBI to throw together the report that shows all the visuals. I realised that this is probably the most efficent way of doing things since writing out code to show the visuals is very time consuming.

```r
write.csv(Age,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/Age.csv",row.names = F
write.csv(AgeTime,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/AgeTime.csv",row.na
write.csv(Sex,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/Sex.csv",row.names = F
write.csv(Prof,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/Prof.csv",row.names =
write.csv(City,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/City.csv",row.names =
write.csv(Local,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/Local.csv",row.names
write.csv(TKids,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/TKids.csv",row.names
write.csv(KidTimeArt,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/KidTimeArt.csv"
write.csv(KidAgeCount,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/KidAgeCount.cs
write.csv(CustomerData,"C:/Users/chris/Documents/R/IkeaInterview220922/Ikea Interview/CSV/CustomerData.
```

#### Answering Question 1 - Who is the main target group? Which segments do you identify?:

The Segments I identify are:

1. Men used VR more often and spent more time looking at the products
2. Aged 26-35, the younger also enjoyed VR and so did the older equally. Then it wavered off after the age of 50.
3. The professions that enjoyed VR most included 4,0,7 and 1.

4. The city type that enjoyed VR the most was type C, more of them using it and looking at products longer. Followed by B, then at the bottom was A
5. The majority of people trying out VR don't have kids
6. There are specific products that people with kids prefer to look at but the list is too long to list here
7. People between the ages of 51 to 55 appear to spend more time on average looking at each product in VR

**Answering Question 3 - The team wants to develop new features that is personalised for each target market. Which target market should they focus on first?**

The right target group to create VR products for are males aged 26-35 living in cities catagorised as "C" having lived there for over 1 year but below 2 years, doesn't have children and profession falls into catagory 4.

**Answering Question 4 - Some users recorded whether they had children or not and others did not. The team is wanting to increase children product sales. They want to know which characteristics a user has that shows that they have children.**

I looked into a variety of different tells that could tell us if someone had kids or not. The two segments that really stood out to me are as follows:

1. There were certain products users with kids would spend longer looking at. The list is too long to list here.
2. People were most likely to have kids if they were between the ages of 51-55 and the chance of people having kids goes down with age until its extremely unlikely that people have kids if they are a teenager.

## Act

Question 2 asked me what kind of data would I want to improve my analysis and back-up the insights I mentioned and why? There are several bits of data I would like:

1. I would like to know the **time and date** when the users were trying out the VR headset to pull in other factors that could contribute to why certain groups were more likely to try VR then others
2. I would also love to have the data **revealing the professions each of the numbers represent** to know more certainly about whether they are working in the shop or surrounding shops and other factors.
3. I would like to know **what each of the City Types represent**, to understand whether they were multiple cities or just one. Whether the type was judged on density of the city, location or something else. All of this data would help build a better profile on the customer which is using the VR. Knowing the types of connections that city has and other variables that may not have been considered when assigning this city type.

The dataset was quite small with only a couple thousand participants. However, if I was working for this company, I would hope that these experiments would continue before they release the final product. I still feel that there is some more data needed to get a clear picture as to who our target market should be. This includes time and date when the VR was tried and more details on parts assigned codes instead of named. But overall, I do feel that the suggestions I included would help any team trying to improve their product.

Thankyou for reading.