Comparing Groups: Tables and Visualizations

Luis Francisco Gómez López

FAEDIS

2024-08-01

UNIVERSIDAD MILITAR
NUEVA GRANADA

Table of contents I

1. Please Read Me

2. Purpose

3. Consumer segmentation survey

4. Acknowledgments

- This presentation is based on (Chapman and Feit 2019, chap. 5)

- Use descriptive summaries by groups and visualize them to investigate differences between groups

- **age**: age of the consumer in years
- **gender**: if the consumer is male of female
- **income**: yearly disposable income of the consumer
- **kids**: number of children of the consumer
- **ownHome**: if the consumer owns a home
- **subscribe**: if the consumer is subscribed or not
- **Segment**: market segment assigned by a clustering algorithm (Chapman and Feit 2019, chap. 11), expert assignment or a segmentation typing tool

- **Segment**:
  - **Moving up**: consumers experiencing upward mobility in terms of their socioeconomic status
  - **Suburb mix**: consumers living in suburban areas
  - **Travelers**: consumers who prioritize experiences and adventures
  - **Urban Hip**: consumers interested in urban culture, artistic expression, and modern trends

- **Import data**

```r
segmentation <- read_csv(file = "http://goo.gl/qw303p")
segmentation |> head(n = 5)
```

```
# A tibble: 5 x 7
    age gender income  kids ownHome subscribe Segment
  <dbl> <chr>   <dbl> <dbl> <chr>   <chr>     <chr>
1  47.3 Male   49483.     2 ownNo   subNo     Suburb mix
2  31.4 Male   35546.     1 ownYes  subNo     Suburb mix
3  43.2 Male   44169.     0 ownYes  subNo     Suburb mix
4  37.3 Female 81042.     1 ownNo   subNo     Suburb mix
5  41.0 Female 79353.     3 ownYes  subNo     Suburb mix
```

- **Transform data**

```
segmentation <- segmentation |>
  mutate(gender = factor(gender, ordered = FALSE),
         kids = as.integer(kids),
         ownHome = factor(ownHome, ordered = FALSE),
         subscribe = factor(subscribe, ordered = FALSE),
         Segment = factor(Segment, ordered = FALSE))

segmentation |> head(n = 5)
```

```
# A tibble: 5 x 7
    age gender income  kids ownHome subscribe Segment
  <dbl> <fct>   <dbl> <int> <fct>   <fct>     <fct>
1  47.3 Male   49483.     2 ownNo   subNo     Suburb mix
2  31.4 Male   35546.     1 ownYes  subNo     Suburb mix
3  43.2 Male   44169.     0 ownYes  subNo     Suburb mix
4  37.3 Female 81042.     1 ownNo   subNo     Suburb mix
5  41.0 Female 79353.     3 ownYes  subNo     Suburb mix
```

- **Basic Formula Syntax**
  - $\sim$ and $+$: operators
  - y: response variable
  - x, z: explanatory variables
  - $y \sim x + z$: a formula which means that y depends on x and z
    - $+$ is used to indicate the addition of predictor variables to the right of the formula
    - Be careful not to confuse the arithmetic operator $+$ with $+$ within a formula

```
?`+` # Arithmetic Operators
?formula # operators in a formula
```

- **Descriptives for n-Way Groups: the base R way**
  - Split data into n subsets and compute summary statistics

```
aggregate(x = income ~ Segment + ownHome,
          data = segmentation, FUN = mean)
```

```
     Segment ownHome   income
1  Moving up   ownNo 54497.68
2 Suburb mix   ownNo 54932.83
3  Travelers   ownNo 63188.42
4  Urban hip   ownNo 21337.59
5  Moving up  ownYes 50216.37
6 Suburb mix  ownYes 55143.21
7  Travelers  ownYes 61889.12
8  Urban hip  ownYes 23059.27
```

- **Descriptives for n-Way Groups: the base R way**
    - Split data into n subsets and compute summary statistics

```
aggregate(x = kids ~ Segment + ownHome,
          data = segmentation, FUN = sum)
```

```
     Segment ownHome kids
1  Moving up   ownNo   82
2 Suburb mix   ownNo   90
3  Travelers   ownNo    0
4  Urban hip   ownNo   43
5  Moving up  ownYes   52
6 Suburb mix  ownYes  102
7  Travelers  ownYes    0
8  Urban hip  ownYes   12
```

- **Descriptives for n-Way Groups: the tidyverse way**
  - Split data into n subsets and compute summary statistics

```
segmentation |>
  group_by(Segment, ownHome) |>
  summarise(mean_income = mean(income))
```

```
# A tibble: 8 x 3
# Groups:   Segment [4]
  Segment    ownHome mean_income
  <fct>      <fct>         <dbl>
1 Moving up  ownNo        54498.
2 Moving up  ownYes       50216.
3 Suburb mix ownNo        54933.
4 Suburb mix ownYes       55143.
5 Travelers  ownNo        63188.
6 Travelers  ownYes       61889.
7 Urban hip  ownNo        21338.
8 Urban hip  ownYes       23059.
```

- **Descriptives for n-Way Groups: the tidyverse way**
  - Split data into n subsets and compute summary statistics

```
segmentation |>
  group_by(Segment, ownHome) |>
  summarise(sum_kids = sum(kids))
```

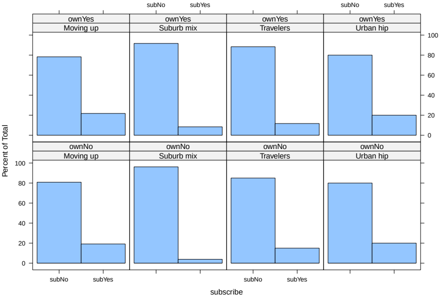
```
# A tibble: 8 x 3
# Groups:   Segment [4]
  Segment    ownHome sum_kids
  <fct>      <fct>      <int>
1 Moving up  ownNo         82
2 Moving up  ownYes        52
3 Suburb mix ownNo         90
4 Suburb mix ownYes       102
5 Travelers  ownNo          0
6 Travelers  ownYes         0
7 Urban hip  ownNo         43
8 Urban hip  ownYes        12
```

- **Basic Formula Syntax**
  - $\sim$, $+$ and $|$: operators
  - y: response variable
  - x: explanatory variable
  - z: grouping variable
  - $y \sim x|z$: y depends on x based on different groups defined by z
    - $|$ is used to separate the grouping variable from the explanatory variable
    - Be careful not to confuse the logical operator $|$ with $|$ within a formula

```
?`|` # Logical Operators
?lattice::xyplot # operators in a formula (you need first to install the package lattice)
```

- **Visualization by group as frequencies: the lattice way**

```r
library(lattice)
histogram(~ subscribe | Segment + ownHome, data = segmentation)
```



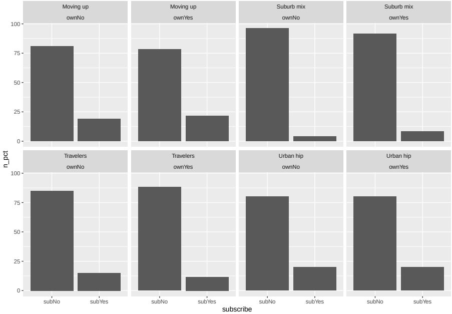
- **Visualization by group as frequencies: the tidyverse way**

```
# Prepare data
subscriber_by_segment_home_ownership <- segmentation |>
  count(subscribe, Segment, ownHome) |>
  group_by(Segment, ownHome) |>
  mutate(n_pct = (n / sum(n)) * 100) |>
  ungroup()
subscriber_by_segment_home_ownership
```

```
# A tibble: 16 x 5
   subscribe Segment    ownHome     n n_pct
   <fct>     <fct>      <fct>   <int> <dbl>
 1 subNo     Moving up  ownNo      38 80.9
 2 subNo     Moving up  ownYes     18 78.3
 3 subNo     Suburb mix ownNo      50 96.2
 4 subNo     Suburb mix ownYes     44 91.7
 5 subNo     Travelers  ownNo      17 85
 6 subNo     Travelers  ownYes     53 88.3
 7 subNo     Urban hip  ownNo      32 80
 8 subNo     Urban hip  ownYes      8 80
 9 subYes    Moving up  ownNo       9 19.1
10 subYes    Moving up  ownYes      5 21.7
11 subYes    Suburb mix ownNo       2  3.85
12 subYes    Suburb mix ownYes      4  8.33
13 subYes    Travelers  ownNo       3 15
14 subYes    Travelers  ownYes      7 11.7
15 subYes    Urban hip  ownNo       8 20
16 subYes    Urban hip  ownYes      2 20
```

- **Visualization by group as frequencies: the tidyverse way**

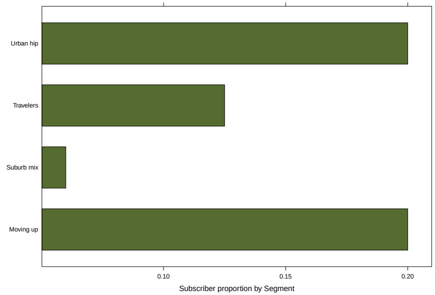
```
subscriber_by_segment_home_ownership |>
  ggplot() +
  geom_col(aes(x = subscribe, y=n_pct)) +
  facet_wrap(facets = vars(Segment,ownHome),
             nrow = 2, ncol = 4)
```



- **Visualization by group as proportions: the lattice way**

```r
# Prepare data
prop_table <- table(segmentation$subscribe,
                    segmentation$Segment) |>
  prop.table(margin = 2) |>
  _[2, ] # You can use _ as a placeholder. Check ?pipeOp
```

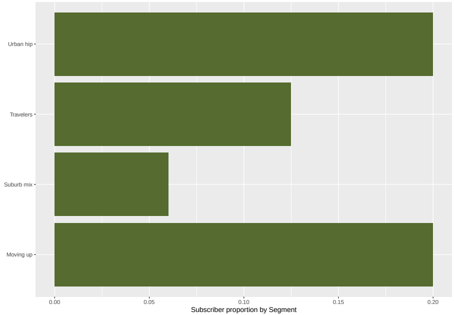
```r
barchart(prop_table,
         xlab='Subscriber proportion by Segment', col='darkolivegreen')
```



- **Visualization by group as proportions: the tidyverse way**

```r
# Prepare data
prop_table <- segmentation |>
  count(subscribe, Segment) |>
  group_by(Segment) |>
  mutate(n_pct = n / sum(n)) |>
  filter(subscribe == 'subYes')
```

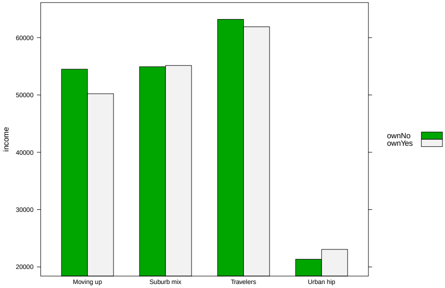
```r
prop_table |> ggplot() +
  geom_col(aes(x=n_pct, y=Segment),
           fill='darkolivegreen') +
  labs(x='Subscriber proportion by Segment',
       y=NULL)
```



- **Visualization by group with continuous data: the lattice way**

```
# Prepare data
seg_income_agg <- aggregate(income ~ Segment + ownHome,
                            data=segmentation, FUN = mean)
```

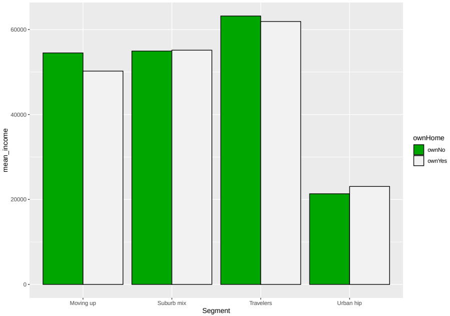
```
barchart(income ~ Segment, data = seg_income_agg,
         groups=ownHome, auto.key=TRUE, # Add groups
         par.settings=simpleTheme(col=terrain.colors(n = 2))) # Change default colors
```



- **Visualization by group with continuous data: the tidyverse way**

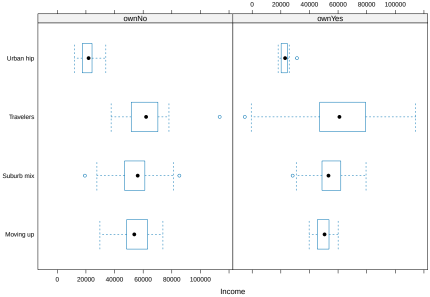
```r
# Prepare data
seg_income_agg <- segmentation |>
  group_by(Segment, ownHome) |>
  summarise(mean_income = mean(income)) |>
  ungroup()
```

```r
seg_income_agg |> ggplot() +
  geom_col(aes(x=Segment, y=mean_income, fill=ownHome),
           position = position_dodge(), color='black') +
  scale_fill_manual(values=terrain.colors(n = 2))
```



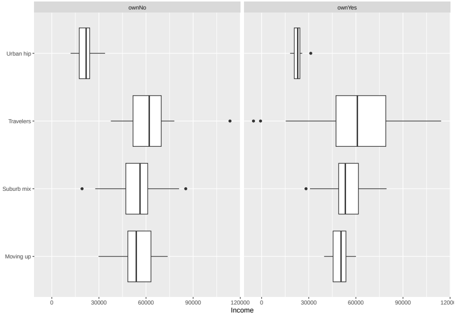
- **Visualization by group with continuous data: the lattice way**

```
bwplot(Segment ~ income | ownHome,
       data = segmentation,
       xlab = 'Income')
```



- **Visualization by group with continuous data: the lattice way**

```r
segmentation |> ggplot() +
  geom_boxplot(aes(x=income, y=Segment)) +
  facet_wrap(facets = vars(ownHome)) +
  labs(x='Income',
       y=NULL)
```



- To my family that supports me
- To the taxpayers of Colombia and the **UMNG students** who pay my salary
- To the **Business Science** and **R4DS Online Learning** communities where I learn **R** and π**-thon**
- To the **R Core Team**, the creators of **RStudio IDE**, **Quarto** and the authors and maintainers of the packages **tidyverse** and **tinytex** for allowing me to access these tools without paying for a license
- To the **Linux kernel community** for allowing me the possibility to use some **Linux distributions** as my main **OS** without paying for a license

References I

Chapman, Chris, and Elea McDonnell Feit. 2019. *R For Marketing Research and Analytics*. 2nd ed. 2019. Use R! Cham: Springer International Publishing : Imprint: Springer. https://doi-org.ezproxy.umng.edu.co/10.1007/978-3-030-14316-9.