Categorical data

Introduction

Goals

In this tutorial we are going to learn

- how to describe categorical data
- how to compute frequencies
- how to compute proportions
- what visualisations are possible
- Interpret a 2-way chart of category relationships

using 

Warm-up quiz

This tutorial will show you how to work with categorical data. To warm up, try these three seemingly simple questions:

Which of these variables are categorical?

- [ ] Passport number: 0166534, 0188674, 0754425, G876102
- [ ] Grades: A+, A, A-, B+, B, B-, C+, C, C-, D
- [x] Times of day: morning, afternoon, evening, night
- [ ] Women's clothing sizes: 8, 10, 12, 14, 16, 18
- [ ] Men's clothing sizes: XS, S, M, L, XL, XXL
- [ ] Scores: 94, 87, 84, 76, 73, 69, 64, 56, 51, 45
- [x] Insurance policy number: 5634221J, 766925L, 692545A, 494801Y
- [x] Times of day: 9, 15, 19, 1

Submit Answer

What might it mean if another student has a student id number that is next, in numerical sequence, to your own?

✗ We can expect to get similar grades

✗ The other student is bigger in some sense than me

✓ Our enrollments were processed at much the same time

✗ We are much more similar compared to students with bigger numerical differences

Correct!

Sometimes we create labels using digits instead of characters. The size of the number is meaningless; only its uniqueness matters.

One of these sequences is not like the others...

✗ 1, 0, 1, 0, 0

✗ green, red, green, red, red

✗ oui, non, oui, non, non

✓ ✓, ✓, ✓, ✗, ✗

✗ yes, no, yes, no, no

✗ 3, 0, 3, 0, 0

✗ TRUE, FALSE, TRUE, FALSE, FALSE

Correct!

The important thing here is the pattern of values, which can be represented in numbers or categories.

Glossary

- **Categorical variable**: A variable whose values are not numbers whose magnitude is meaningful. Also known as a “Factor” or “Qualitative” or “Nominal” variable.
- **Categories**: The unique values of a categorical variable. Also known as “Levels”
- **Cardinality**: The count of unique levels of a variable.
- **Ordinal Variable**: A categorical variable whose levels have ranking
- **Cyclic Variable**: A categorical variable whose levels can be drawn onto a clock face. The levels loop around so that the last level is most similar to the first.

Examples

Example 1

Variable *CustService* has values: Good, Good, Bad, Good, Excellent, No-opinion, Bad, No-opinion, Excellent, No-opinion, Bad

The **levels** of *Mood* are, in ascending order: Bad, No-opinion, Good, Excellent The **cardinality** of *Mood* is 4
The number of cases is 11

Example 2

Variable *Mood* has values: Happy, Happy, Sad, Happy, Excited, Anxious, Sad, Anxious, Excited

The **levels** of *Mood* are, in alphabetic order: Anxious, Excited, Happy, Sad
The **cardinality** of *Mood* is 4
The number of cases is 9

Example 3

Variable *Taste* has values: Bitter, Bitter, Sweet, Bitter, Sour, Salt, Sweet, Salt, Sour, Sweet, Umami, Umami, Bitter

How many levels does variable *Taste* have?

✗ 6
✗ 13
✓ 5
✗ 4
✗ 12

Correct!

The 5 levels are Bitter, Salt, Sour, Sweet, Umami

Is the *Taste* variable an ordinal variable?

✓ No, as there is no order to tastes
✗ Yes, as there is an order to tastes

Correct!

Example 4

Variable *Season* has values: Summer, Summer, Winter, Summer, Autumn, Spring, Winter, Spring, Autumn, Winter, Summer

What kind of categorical variable is *Seasons*?

✓ Cyclic, as levels repeat every year

✗ Nominal, as the everything preceeds everything else

✗ Ordinal, as their order is Winter, Spring, Summer, Autumn

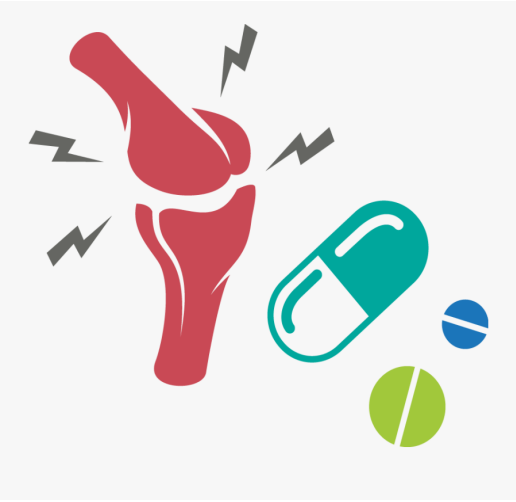
Correct!

Cyclical is the best answer because this allows the first level to be similar to the last level

Categorical Tabulation

To explore how to compute frequencies and proportions using R, we shall need a data set.

Arthritis data



The “Arthritis” data set is inbuilt to the `vcd` package. We shall begin by looking at its structure. The structure is revealed by the `str()` function. Press the “Run Code” button.

R-code Start Over Run Code

```
1 library(vcd)
2 data(Arthritis)
3 str(Arthritis)
```

```
'data.frame':   84 obs. of  5 variables:
 $ ID       : int  57 46 77 17 36 23 75 39 33 55 ...
 $ Treatment: Factor w/ 2 levels "Placebo","Treated": 2 2 2 2 2 2 2 2 2 2 ...
 $ Sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
 $ Age      : int  27 29 30 32 46 58 59 59 63 63 ...
 $ Improved : Ord.factor w/ 3 levels "None"<"Some"<..: 2 1 1 3 3 3 1 3 1 1 ...
```

As you can see, there are 3 categorical variables. In R, these are labelled “factors”. The last of these, *Improved*, is an ordinal factor. This means that the various levels have rank; they can be judged to be bigger or smaller than each other. Each factor has its number of levels (cardinality) shown in the summery above. Notice, also, that there are 84 cases and 5 variables.

The levels of “Improved” can be revealed using the following code snippet:

R-code Start Over Run Code

```
1 levels(Arthritis$Improved)
2
3
```

```
[1] "None"   "Some"   "Marked"
```

We can interpret *Improved* levels as “None” < “Some” < “Marked”, as if they were numbers.

The number of levels of “Improved” can be revealed using the following code snippet:

R-code Start Over Run Code

```
1 nlevels(Arthritis$Improved)
2
3
```

```
[1] 3
```

Frequencies

Frequencies are the counts of the categories present in some data.

Frequencies of the treatment types can be computed using the `table()` function. In this code snippet we shall pipe the data from one command to another using the ” %>%” operator. The `table()` function will only see the *Treatment* variable of the **Arthritis** data set. Press “Run code” to see the output.

R-code Start Over Run Code

```
1 Arthritis %>%
2   select(Treatment) %>%
3   table()
```

```
Treatment
Placebo Treated
     43      41
```

Notice that the numbers above sum to 84 which is the number of cases in our data set.

We can do the same for the **Sex** variable. Please edit the snippet to report the frequencies of the *Sex* variable.

R-code Start Over Run Code Submit Answer

```
1 Arthritis %>%
2   select(Sex) %>%
3   table()
```

```
Sex
Female   Male
    59     25
```

Terrific!

We can also perform frequency tabulation across more than one variable. The code below will create a 2 by 2 grid of counts.

R-code Start Over Run Code

```
1 Arthritis %>%
2   select(Treatment, Sex) %>%
3   table()
```

```
         Sex
Treatment Female Male
  Placebo     32   11
  Treated     27   14
```

Proportions

Proportions are similar to frequencies except rather than ranging between 0 and 84 (in this example), proportions will range between 0 and 1.

Calculating proportions can be achieved by dividing any counts by 84. The last line produces a 2 by 2 table of results, every element of which is then divided by 84.

R-code Start Over Run Code

```
Arthritis %>%
  select(Treatment, Sex) %>%
  table() / 84
```

```
         Sex
Treatment    Female      Male
  Placebo 0.3809524 0.1309524
  Treated 0.3214286 0.1666667
```

Now the sum of these numbers should come to 1 rather than 84.
Another way to do this, is to use the `proportions()` function. In the second version of the code, below, we make no assumption that there are 84 cases. Press the “Run Code” button.

R-code Start Over Run Code

```
Arthritis %>%
  select(Treatment, Sex) %>%
  table() %>%
  proportions()
```

```
         Sex
Treatment    Female      Male
  Placebo 0.3809524 0.1309524
  Treated 0.3214286 0.1666667
```

In R it is common to have many ways of achieving the same result. Sometimes there is no “best” way and it comes down to personal preference.

Let’s see what happens when we open up our analysis of categorical variables to all three factors.

R-code Start Over Run Code

```
1 Arthritis %>%
2   select(Treatment, Sex, Improved) %>%
3   table() %>%
4   proportions()
```

```
, , Improved = None

         Sex
Treatment     Female       Male
  Placebo 0.22619048 0.11904762
  Treated 0.07142857 0.08333333

, , Improved = Some

         Sex
Treatment     Female       Male
  Placebo 0.08333333 0.00000000
  Treated 0.05952381 0.02380952

, , Improved = Marked

         Sex
Treatment     Female       Male
  Placebo 0.07142857 0.01190476
  Treated 0.19047619 0.05952381
```

Visualising Single Variables

Single variables can be plotted as:

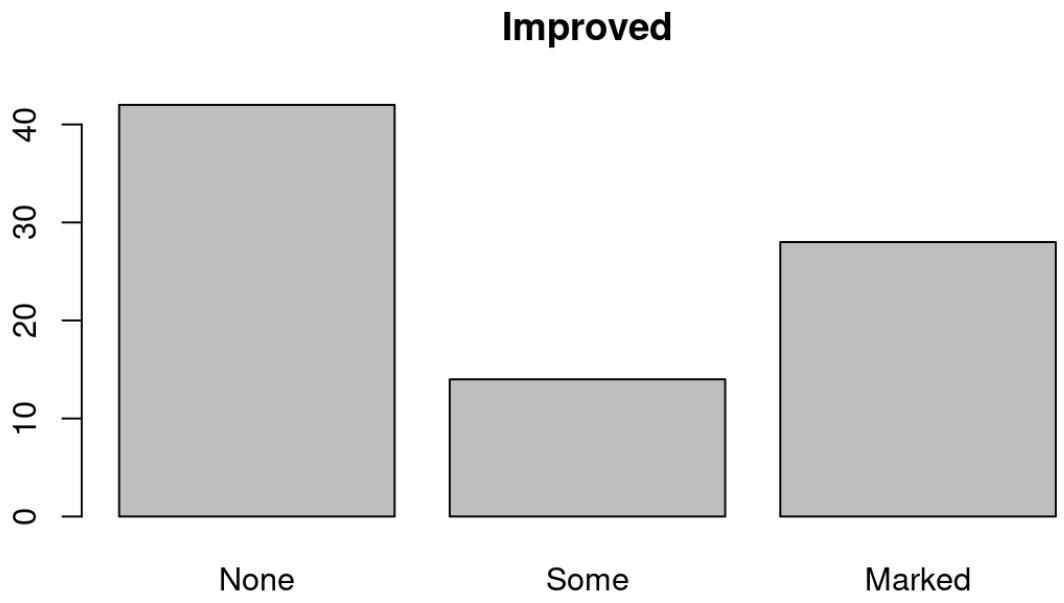
- Bar chart
- Pie chart

Bar chart

The `plot()` function , when given a categorical variable, knows that a histogram representation is required. Notice that the y axis is the frequency. Press “Run Code”.

R-code Start Over Run Code

```
1 library(graphics)
2 Arthritis %>%
3   select(Improved) %>%
4   table() %>%
5   barplot(main = "Improved")
```

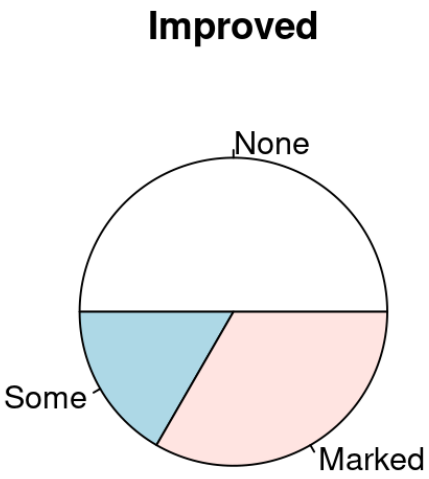


Pie chart

The same information can generate a pie chart. Please add a chart main title of “Improved” and submit the answer.

R-code Start Over Run Code Submit Answer

```
1 Arthritis %>%
2   select(Improved) %>%
3   table() %>%
4   graphics::pie(main = "Improved")
```



Lovely job!

Visualising Two Variables

Single variables can be plotted as:

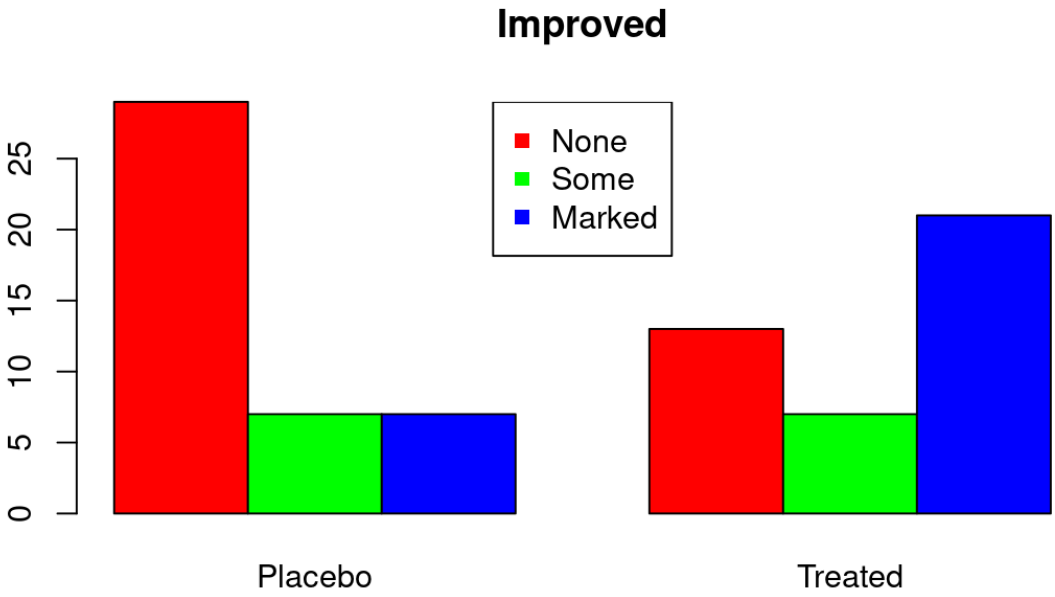
- Bar chart
- Stacked bar chart
- Mosaic chart

Bar chart

To show this data in a stacked bar, a legend and colours are needed to make sense of the information.

R-code Start Over Run Code

```
1 Arthritis %>%
2   select(Improved, Treatment) %>%
3   table() %>%
4   barplot(beside = TRUE, col = c("red","green", "blue"), main = "Improved")
5 legend("top", legend = c("None", "Some", "Marked"), col = c("red","green", "blue"), pch=
```

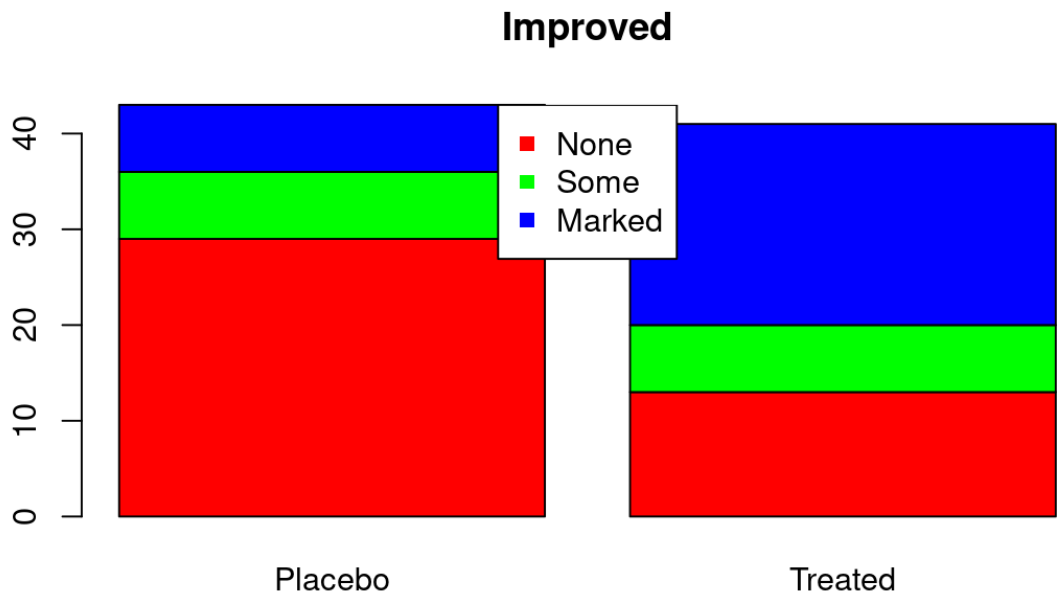


Stacked bar chart

To show this data in a stacked bar, a legend and colours are needed to make sense of the information.

R-code Start Over Run Code

```
1 Arthritis %>%
2   select(Improved, Treatment) %>%
3   table() %>%
4   barplot(col = c("red","green", "blue"), main = "Improved")
5 legend("top", legend = c("None", "Some", "Marked"), col = c("red","green", "blue"), pch=
```

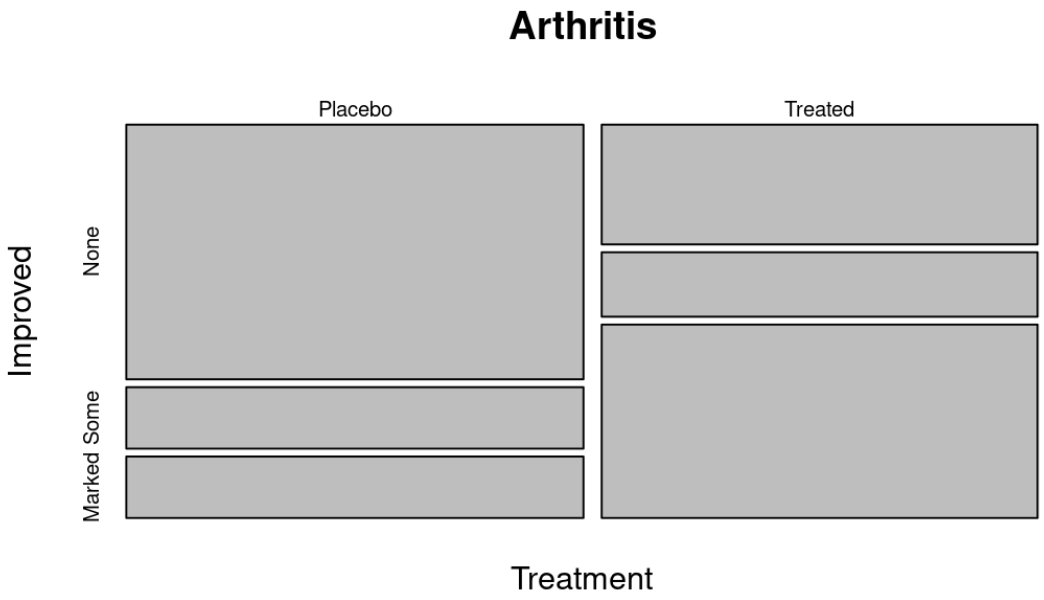


Mosaic chart

The code below shows how to plot a mosaic chart using the **graphics** package. In this chart the frequencies are related to the areas of the chart regions.

R-code Start Over Run Code

```
1 Arthritis %>%
2   select(Treatment, Improved) %>%
3   table() %>%
4   mosaicplot(main = "Arthritis")
```



Visualising Many Variables

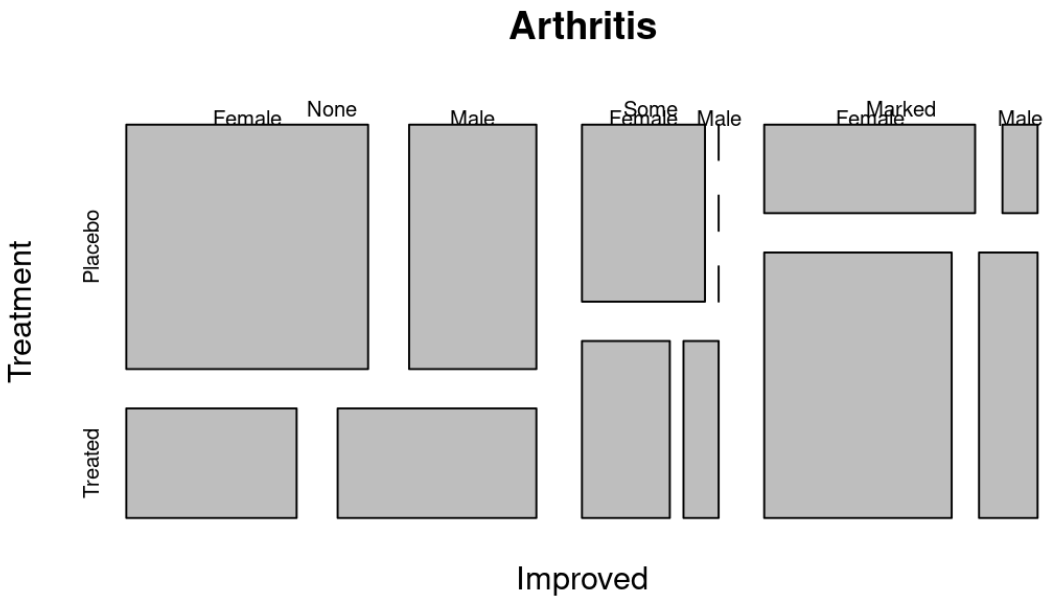
The Mosaic chart becomes the only feasible chart as the number of categorical variables gets above 2. After 4 categorical variables, even a mosaic chart becomes hard to understand.

Mosaic chart

You can see visually that Males+Placebo+Some is a heavily under represented intersection of characteristics.

R-code Start Over Run Code

```
1 Arthritis %>%
2   select(Improved, Treatment, Sex) %>%
3   table() %>%
4   mosaicplot(main = "Arthritis")
```



Practical Issues

Missing values

What happens when a categorical variable has missing values? Let's do an experiment to find out. We shall set the 12th value of the *Sex* variable to NA. This variable now has a missing value.

R-code Start Over Run Code

```
1 Arthritis2 <- Arthritis
2 Arthritis2$Sex[12] <- NA
3 Arthritis2$Sex
```

```
 [1] Male   Male   Male   Male   Male   Male   Male   Male   Male   Male
[11] Male   <NA>   Male   Male   Female Female Female Female Female Female
[21] Female Female Female Female Female Female Female Female Female Female
[31] Female Female Female Female Female Female Female Female Female Female
[41] Female Male   Male   Male   Male   Male   Male   Male   Male   Male
[51] Male   Male   Female Female Female Female Female Female Female Female
[61] Female Female Female Female Female Female Female Female Female Female
[71] Female Female Female Female Female Female Female Female Female Female
[81] Female Female Female Female
Levels: Female Male
```

The count of missing values Sex can be counted using `is.na()` and `sum()` functions

R-code Start Over Run Code

```
Arthritis2 %>%
  select(Sex) %>%
  is.na() %>%
  sum()
```

```
[1] 1
```

The `levels()` ignores missing values. The levels of *Sex* are:

R-code Start Over Run Code

```
levels(Arthritis2$Sex)


```

```
[1] "Female" "Male"
```

The `table()` function only sees 83 cases now. The frequencies of *Sex* are:

R-code Start Over Run Code

```
Arthritis2 %>%
  select(Sex) %>%
  table()
```

```
Sex
Female   Male
    59     24
```

An alternative is to allow missing values to be counted as well. The `table()` function has a parameter called **useNA = "ifany"** which allows this behaviour.

R-code Start Over Hints Run Code Submit Answer

```
Arthritis2 %>%
  select(Sex) %>%
  table(useNA = "ifany")
```

```
Sex
Female   Male   <NA>
    59     24      1
```

Terrific!

Outliers

Outliers for quantitative variables are values that do not seem to belong to the distribution of values. Does this apply to categorical data? In other words, can categorical variables have outlier values?

Given that the categories are supposed to be known in advance, how is possible to have a value that does not seem to belong?

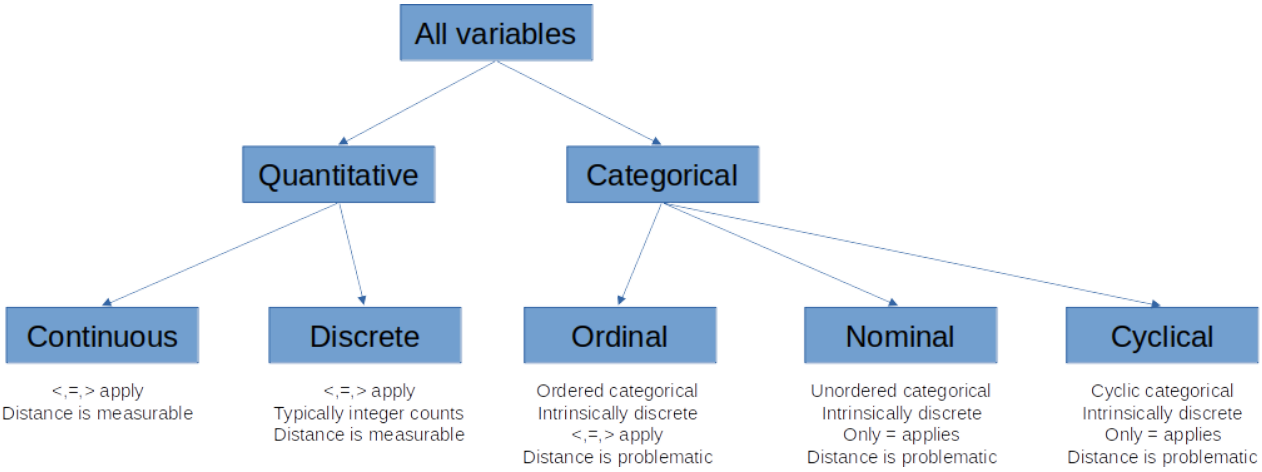
Categorical data can exhibit the following characteristics:

- Unusual categories: Categories with unusually low or unusually high proportions compared to the others. The low proportion ones seem be a bit like outliers.
- Novel categories: When the population is randomly sampled further, some new categories begin to appear that were not seen previously. Generally these have low proportions (or they would have been found sooner.)
- Unusual Intersections of categorical variables: Category intersections with unusually low or unusually high proportions compared to the others. The low proportion ones seem be a bit like outliers. The *Male:Placebo:Some* intersection of *Arthritis* was a case of this idea.

In an experimental study (as opposed to an Observational study) the design goal may be to ensure all categories (and all intersections of categories) have similar proportions.

Wrap Up

Hierarchy of types



Hierarchy of data types

Code

In this tutorial you have been exposed to the following R functions:

Function	Package	Description
str()	base	Report structure
levels()	base	Report categories
nlevels()	base	Number of categories
select()	dplyr	Choose a subset of variables
table()	base	Tabulate
proportions()	base	Turn counts into proportions
is.na()	base	Whether values are NA (missing)

Function	Package	Description
sum()	base	Sums up the values of a variable
pie()	graphics	Create a pie chart
barplot()	graphics	Create a bar chart
legend()	graphics	Create a chart legend
mosaicplot()	graphics	Create a mosaic chart

The end

In case this on-line tutorial is not available in future, you may want to keep a PDF copy of this material for reference purposes.

Export as PDF

Topic not yet completed...