Categorical Data Analysis

S20426

2025-04-25

Nominal Data:

Definition:

Nominal data is used to label or categorize data without any inherent order or ranking.

Examples:

Gender (male, female), eye color (blue, brown, green), types of fruits (apple, banana, orange).

Characteristics:

The order of categories is not meaningful, and there’s no concept of “more” or “less”.

Ordinal Data:

Definition:

Ordinal data represents categories with a meaningful order or ranking, but the intervals between categories are not necessarily equal.

Examples:

Education level (high school, bachelor’s, master’s), rating on a scale (1-5 stars), survey responses (strongly disagree, disagree, neutral, agree, strongly agree).

Characteristics:

The order of categories is important, but the difference between adjacent categories might not be equal.

Binary Data:

Definition:

Binary data is a type of nominal data with only two possible values or categories, often representing a choice between “yes” and “no,” “true” and “false,” or “0” and “1”.

Examples:

Whether a customer has purchased a product (yes or no), whether a light switch is on or off, the result of a coin flip (heads or tails).

Characteristics:

It’s a specific type of nominal data where the categories are mutually exclusive and exhaustive.

Why use factors?

Factors are especially useful in statistical modeling, plotting, and when you want to treat data as categories rather than continuous values.

```
#library(catdata)
data(knee)
head(knee)
```

```
##   N Th Age Sex R1 R2 R3 R4
## 1 1  1  28   1  4  4  4  4
## 2 2  1  32   1  4  4  4  4
## 3 3  1  41   1  3  3  3  3
## 4 4  2  21   1  4  3  3  2
## 5 5  2  34   1  4  3  3  2
## 6 6  1  24   1  3  3  3  2
```

N : Patient’s number

Th : Therapy (placebo = 1, treatment = 2)

Age : Age in years

Sex : Gender (male = 0, female = 1)

R1 : Pain before treatment (no pain = 1, severe pain = 5)

R2 : Pain after three days of treatment

R3 : Pain after seven days of treatment

R4 : Pain after ten days of treatment

Check the structure of the data

```
str(knee)
```

```
## 'data.frame':    127 obs. of  8 variables:
##  $ N  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Th : int  1 1 1 2 2 1 2 2 2 1 ...
##  $ Age: int  28 32 41 21 34 24 28 40 24 39 ...
##  $ Sex: num  1 1 1 1 1 1 1 1 0 0 ...
##  $ R1 : int  4 4 3 4 4 3 4 3 4 4 ...
##  $ R2 : int  4 4 3 3 3 3 3 2 4 4 ...
##  $ R3 : int  4 4 3 3 3 3 3 2 4 4 ...
##  $ R4 : int  4 4 3 2 2 2 2 2 3 3 ...
```

Convert into factor variables

```
knee$Th <- as.factor(knee$Th)
knee$Sex <- as.factor(knee$Sex)
str(knee)
```

```
## 'data.frame':    127 obs. of  8 variables:
##  $ N  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Th : Factor w/ 2 levels "1","2": 1 1 1 2 2 1 2 2 2 1 ...
##  $ Age: int  28 32 41 21 34 24 28 40 24 39 ...
##  $ Sex: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 1 1 ...
##  $ R1 : int  4 4 3 4 4 3 4 3 4 4 ...
##  $ R2 : int  4 4 3 3 3 3 3 2 4 4 ...
##  $ R3 : int  4 4 3 3 3 3 3 2 4 4 ...
##  $ R4 : int  4 4 3 2 2 2 2 2 3 3 ...
```

```
## you can see how many levels in the columns
```

Changing factor levels

```
levels(knee$Th) <- c("Placebo","Treatment")
levels(knee$Sex) <-c("Male","Female")
head(knee)
```

```
##   N        Th Age    Sex R1 R2 R3 R4
## 1 1   Placebo  28 Female  4  4  4  4
## 2 2   Placebo  32 Female  4  4  4  4
## 3 3   Placebo  41 Female  3  3  3  3
## 4 4 Treatment  21 Female  4  3  3  2
## 5 5 Treatment  34 Female  4  3  3  2
## 6 6   Placebo  24 Female  3  3  3  2
```

Creating tabulated summaries

```
t_1=table(knee$Th)
t_1
```

```
##
##   Placebo Treatment
##        63        64
```

```
prop.table(t_1)
```

```
##
##   Placebo Treatment
##  0.496063  0.503937
```

```
t_2=table(knee$Th,knee$Sex)
t_2
```

```
##
##             Male Female
##   Placebo     17     46
##   Treatment   21     43
```

```
prop.table(t_2)
```

```
##
##                  Male    Female
##   Placebo   0.1338583 0.3622047
##   Treatment 0.1653543 0.3385827
```

Using CrossTable function in gmodels package

```
library(gmodels)
```

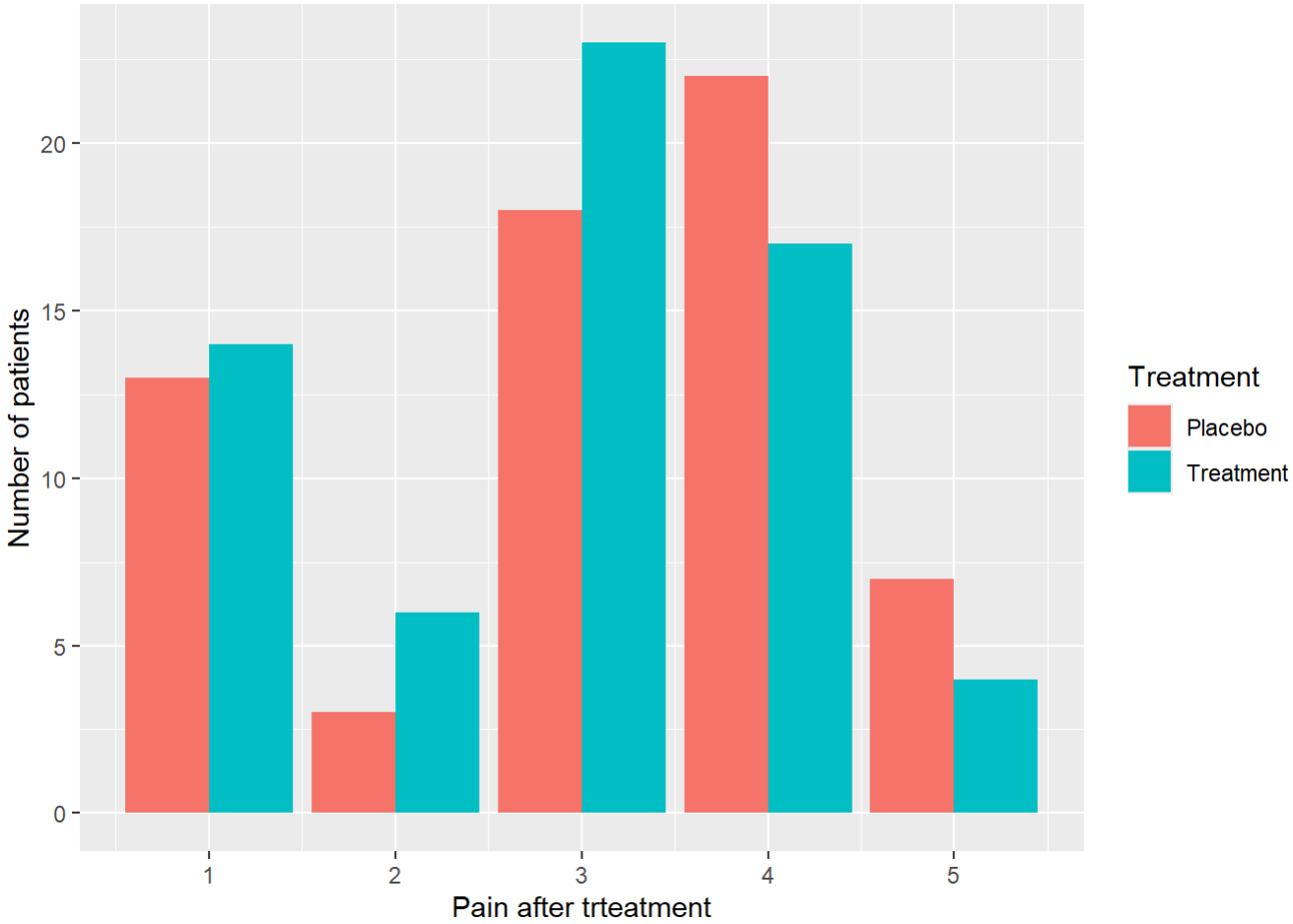
```
## Warning: package 'gmodels' was built under R version 4.3.3
```

```
CrossTable(table(knee$Th,knee$Sex))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  127
##
##
##              |
##              |      Male |    Female | Row Total |
## -------------|-----------|-----------|-----------|
##      Placebo |        17 |        46 |        63 |
##              |     0.182 |     0.078 |           |
##              |     0.270 |     0.730 |     0.496 |
##              |     0.447 |     0.517 |           |
##              |     0.134 |     0.362 |           |
## -------------|-----------|-----------|-----------|
##    Treatment |        21 |        43 |        64 |
##              |     0.179 |     0.076 |           |
##              |     0.328 |     0.672 |     0.504 |
##              |     0.553 |     0.483 |           |
##              |     0.165 |     0.339 |           |
## -------------|-----------|-----------|-----------|
## Column Total |        38 |        89 |       127 |
##              |     0.299 |     0.701 |           |
## -------------|-----------|-----------|-----------|
##
##
```

Categorical Data Visulization

```
#library(ggplot2)
ggplot(knee, aes(x = R2, fill = Th)) + geom_bar(position = "dodge") +
  labs(x = "Pain after trteatment",
       y = "Number of patients",
       fill = "Treatment")
```



Chi-square goodness of fit test

A statistical hypothesis test used to determine whether a variable is likely to come from a specified distribution or not.

In Knee injuries dataset, let's check whether the patients were randomly allocated to the treatment and placebo groups.

Null hypothesis: Ptrt=Pplc=0.5

```
probabilities <- c(Treatment = .5, Placebo = .5)
probabilities
```

```
## Treatment   Placebo
##       0.5       0.5
```

```
library(lsr)
goodnessOfFitTest(x=knee$Th) # No need to input probabilities if they are equal
```

```
##
##      Chi-square test against specified probabilities
##
## Data variable:   knee$Th
##
## Hypotheses:
##    null:        true probabilities are as specified
##    alternative: true probabilities differ from those specified
##
## Descriptives:
##           observed freq. expected freq. specified prob.
## Placebo               63           63.5             0.5
## Treatment             64           63.5             0.5
##
## Test results:
##    X-squared statistic:  0.008
##    degrees of freedom:  1
##    p-value:  0.929
```

Test results:

Chi-square statistic = 0.008 This is a very small number, meaning the observed and expected counts are almost the same.

Degrees of freedom (df) = 1

Since there are 2 categories (Treatment and Placebo), df = 2 - 1 = 1.

p-value = 0.929

This is a very high p-value.

Interpretation:

Since the p-value is much greater than 0.05, we fail to reject the null hypothesis. that means there's no significant difference between observed and expected group sizes.

Another Method

```
chisq.test(x=table(knee$Th))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(knee$Th)
## X-squared = 0.007874, df = 1, p-value = 0.9293
```

Chi-square test of Independence

A hypothesis test used to determine whether two categorical or nominal variables are likely to be related or not.

In Knee injuries dataset, let's check whether the variables Th and R2 are independent or not

```
#library(lsr)
knee$R2<-as.factor(knee$R2)
associationTest( formula = ~Th+R2, data = knee )
```

```
## Warning in associationTest(formula = ~Th + R2, data = knee): Expected
## frequencies too small: chi-squared approximation may be incorrect
```

```
##
##      Chi-square test of categorical association
##
## Variables:   Th, R2
##
## Hypotheses:
##    null:        variables are independent of one another
##    alternative: some contingency exists between variables
##
## Observed contingency table:
##            R2
## Th           1  2  3  4  5
##   Placebo   13  3 18 22  7
##   Treatment 14  6 23 17  4
##
## Expected contingency table under the null hypothesis:
##            R2
## Th             1    2    3    4    5
##   Placebo   13.4 4.46 20.3 19.3 5.46
##   Treatment 13.6 4.54 20.7 19.7 5.54
##
## Test results:
##    X-squared statistic:  3.098
##    degrees of freedom:  4
##    p-value:  0.542
##
## Other information:
##    estimated effect size (Cramer's v):  0.156
##    warning: expected frequencies too small, results may be inaccurate
```

Another Method (Independence)

```
T3=table(knee$Th,knee$R2)
chisq.test(T3)
```

```
## Warning in chisq.test(T3): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  T3
## X-squared = 3.0983, df = 4, p-value = 0.5415
```

Assumptions of chi-square test

Expected frequencies are sufficiently large.

If this assumption is violated If your expected cell counts are too small, check out the Fisher exact test.

As can be seen it does not calculate a test statistic.

```
T3=table(knee$Th,knee$R2)
fisher.test(T3)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  T3
## p-value = 0.5641
## alternative hypothesis: two.sided
```

Observations are independent.

If observations are not independent It may be possible to use the McNemar test or the Cochran test.

```
R2.merge=factor(ifelse(knee$R2==1 | knee$R2==2,1,2))
R3.merge=ifelse(knee$R3==1 | knee$R3==2,1,2)
T4=table(R2.merge,R3.merge)
mcnemar.test(T4)
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  T4
## McNemar's chi-squared = 9.0909, df = 1, p-value = 0.002569
```

Odds Ratio and 95% CI

```
library(vcd) # install the package first
```

```
## Warning: package 'vcd' was built under R version 4.3.3
```

```
## Loading required package: grid
```

```
T5 <-table(knee$R4,knee$Th)
odds.2cb <- oddsratio(T5,log=F) # computes the odds ratio
summary(odds.2cb) # summary displays the odds ratio
```

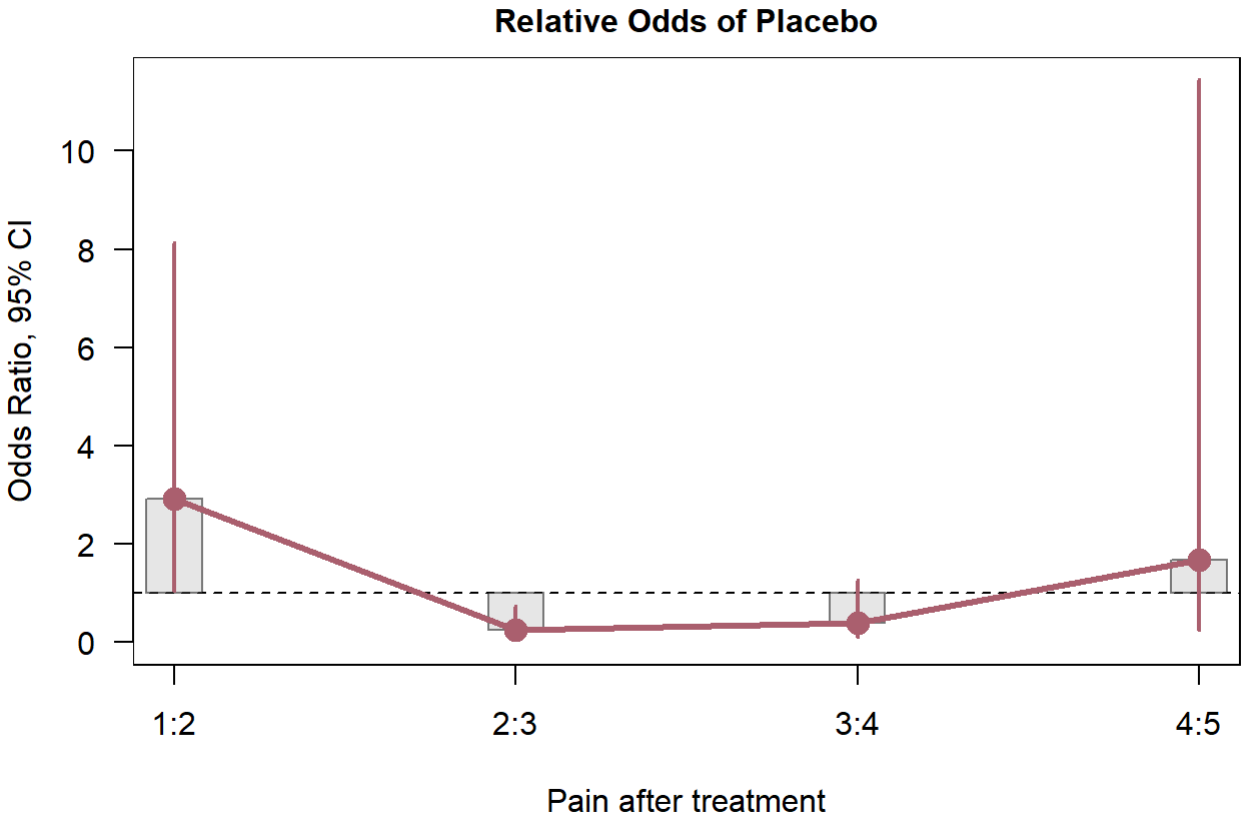
```
##
## z test of coefficients:
##
##                        Estimate Std. Error z value Pr(>|z|)
## 1:2/Placebo:Treatment  2.90789    1.52468  1.9072  0.05649 .
## 2:3/Placebo:Treatment  0.24176    0.13799  1.7520  0.07978 .
## 3:4/Placebo:Treatment  0.38182    0.23506  1.6243  0.10430
## 4:5/Placebo:Treatment  1.66667    1.63865  1.0171  0.30911
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(odds.2cb) # displays the confidence intervals
```

```
##                            2.5 %     97.5 %
## 1:2/Placebo:Treatment 1.04057283  8.1261509
## 2:3/Placebo:Treatment 0.07898152  0.7400092
## 3:4/Placebo:Treatment 0.11424274  1.2760997
## 4:5/Placebo:Treatment 0.24263538 11.4483623
```

Plot the odds ratio and their respective confidence intervals.

```
plot(odds.2cb, main = "Relative Odds of Placebo", xlab = "Pain after treatment", ylab = "Odds Ratio, 95% CI")
```


Relative Odds of Placebo
Odds Ratio, 95% CI
0
2
4
6
8
10
1:2
2:3
3:4
4:5
Pain after treatment