Introduction to Statistics - Young Researchers Fellowship Program

Lecture 2 - More on descriptive statistics & statistical data visualization

Daniel Sánchez Pazmiño

Laboratorio de Investigación para el Desarrollo del Ecuador

September 2024

Recap

- So far, we covered univariate descriptive statistics:
  - Measures of central tendency
  - Measures of dispersion
  - Measures of position
  - Measures of distributional shape
- We must also look at descriptive statistics in other contexts:
  - Categorical data descriptive stats
  - Bivariate descriptive stats (measures of association)
  - Statistical data visualization: boxplots, histograms, scatter plots, etc.

Categorical data descriptive statistics

Describing categorical data

- Our univariate descriptive statistics applied quite well to numerical data.
- However, for categorical data, would we be able to calculate a mean?
  - No, because categories are not numbers.
- There are specific descriptive stats, some of them which mirror numerical data stats, which should be reviewed for categorical data.
  - The frequency of each category
  - Frequency tables
  - Relative frequencies

Frequency of ocurrence

- The frequency of ocurrence of a category is the number of times it appears in the dataset.

$$f = \sum_{i=1}^{n} I(x_i = c)$$

where f is the frequency of category c, n is the number of observations, and I is the indicator function. - $I(x_i = c)$ is 1 if $x_i = c$ and 0 otherwise.

- This can be called the *absolute frequency* of a category.

Frequency of ocurrence

- Notice that if a variable in a dataset is categorical, it may have two or more categories within itself.
  - `sex` may have two categories: `male` and `female`
  - `ethnicity` may have multiple categories: `mestizo`, `afroecuadorian`, `indigenous`, etc.
- Each category of a categorical variable would have its own frequency of ocurrence.

Relative frequency

- The relative frequency of a category is the proportion of times it appears in the dataset.

$$rf = \frac{f}{n}$$

where rf is the relative frequency of category c, f is the frequency of category c, and n is the number of observations.

- This is given to you in *proportion* form.
  - For example, if the relative frequency of `male` is 0.6, then 60% of the dataset falls under the `male` category.
  - Proportions are always between 0 and 1.
  - Find a percentage by multiplying by 100, however, it is recommended to keep it in proportion form for easier calculations.

Frequency tables

- A frequency table is a table that shows the frequency of each category in a categorical variable.
- It is a way to summarize the distribution of a categorical variable.
- For example, consider the SUPERCIAS dataset. We can calculate the frequency of each category in the `region` variable.

Var1	Freq
COSTA	105744
GALÁPAGOS	1340
ORIENTE	7257
SIERRA	95277

Frequency tables

- A frequency table can be presented with both the absolute frequency and the relative frequency.
- The relative frequency is calculated by dividing the absolute frequency by the total number of observations.
- The relative frequency is a proportion, so it is always between 0 and 1.

Frequencies with R

- We can use the `table()` function in R to calculate the frequency of ocurrence of each category in a categorical variable (i.e. a table of frequencies).
  - Works similarly to the numerical data `table()` function.
- Alternatively, use `count` from `dplyr` to calculate the frequency of ocurrence of each category in a categorical variable.
  - This is a shorthand for `group_by()` and `summarize()` for a variable which isn't numerical.
- We may extract a specific category frequency by subsetting the table or using `pull()` from `dplyr`.

Example: SUPERCIAS dataset

- The code for the previous frequency table is as follows:

```
supercias$region %>%
    table()
```

.

```
   COSTA GALÁPAGOS   ORIENTE    SIERRA
  105744      1340      7257     95277
```

Example: SUPERCIAS dataset

- A tidyverse workflow for the frequency table is as follows:

```
## Relative frequencies

supercias %>%
    count(region)
```

```
# A tibble: 4 x 2
  region          n
  <chr>       <int>
1 COSTA      105744
2 GALÁPAGOS    1340
3 ORIENTE      7257
4 SIERRA      95277
```

R implementation for relative frequencies

- For a relative frequency table, we may add an additional column to the frequency table with `mutate()`.
  - This column will be the relative frequency of each category.
- A base R implementation would be passing the `table()` call to `prop.table()`.

Example: SUPERCIAS dataset

- The code for the relative frequency table is as follows:

```
supercias$region %>%
    table() %>%
    prop.table()
```

.

COSTA	GALÁPAGOS	ORIENTE	SIERRA
0.504460495	0.006392581	0.034620119	0.454526806

Example: SUPERCIAS dataset

- A tidyverse workflow for the relative frequency table is as follows:

```
supercias %>%
    count(region) %>%
    mutate(relative_frequency = n / sum(n))
```

```
# A tibble: 4 x 3
  region          n relative_frequency
  <chr>       <int>              <dbl>
1 COSTA      105744            0.504
2 GALÁPAGOS    1340            0.00639
3 ORIENTE      7257            0.0346
4 SIERRA      95277            0.455
```

- Note how the denominator, n, is the sum of the frequencies, `sum(n)`.

Dichotomous variables

- A dichotomous variable is a categorical variable with only two categories, which in some cases can be represented as 0 and 1.
  - These are also called binary or dummy variables.
- For example, `sex` can be represented as `male` and `female`, which can be coded as 0 and 1, respectively.
  - It's important to read the variables dictionary in a dataset to understand the coding of dichotomous variables.

Dichotomous variables

- The reason why dichotomous variables are important is that they can be used in statistical models.
  - It is beneficial to understand the category of interest as a 1 and the other category as a 0.
  - We will talk more about these in other lectures and the Econometrics module.
- For now, know that *if you take the mean of a dichotomous variable, you are calculating the proportion of the category of interest* in the dataset.

$$f_{dic} = \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

where $\bar{x}$ is the mean of the dichotomous variable dic, n is the number of observations, and x_i is the value of the dichotomous variable for observation i.

Dealing with dichotomous variables in R

- If a variable is dichotomous, we may want to recode it to its original values for better interpretation.
  - For example, `1` and `0` can be recoded to `male` and `female`, respectively.
- This can be done using `case_when()` from `dplyr` in a `mutate()` call.
- This would also allow you to do the reverse, recoding a categorical variable to a dichotomous variable.

Dealing with dichotomous variables in R

- Other solutions exist for recoding dichotomous variables, such as `recode()` from `dplyr` or `if_else()` from `dplyr`.
- However, R allows for the use of factors, which are a much more effective way to deal with categorical variables for statistical models.
  - These mantain the categories and their levels (order, if applicable or a numerical value) at the same time.
- We can convert a dichotomous variable to a factor using `as.factor()`.
  - This is a base R solution.
- The `forcats` package from the tidyverse is a specialized package for dealing with factors.
  - It has functions for reordering levels, recoding levels, and other factor-related tasks.

Example: Dichotomous variable in SUPERCIAS dataset

- We can manually create dummies for dichotomous variables in the SUPERCIAS dataset through a `mutate()` call and `if_else()`.
  - For example, we can create a dummy for `region` being `SIERRA`.

```
supercias_dummies <-
    supercias %>%
    mutate(region_sierra = if_else(region == "SIERRA", 1, 0))
```

- The proportion of `SIERRA` in the dataset can be calculated by taking the mean of the dummy variable.

```
supercias_dummies$region_sierra %>%
    mean()
```

```
[1] 0.4545268
```

Example: Dichotomous variable in SUPERCIAS dataset

- We can verify the proportion above is correct with a frequency table.

```
table(supercias_dummies$region) %>% prop.table()
```

```
      COSTA   GALÁPAGOS     ORIENTE      SIERRA
0.504460495 0.006392581 0.034620119 0.454526806
```

Descriptive statistics for bivariate data

Measures of association

- Measures of association are used to describe the relationship between two variables.
  - Also called bivariate descriptive statistics.
- These are:
  - Covariance
  - Correlation
  - Contingency tables

Covariance

- Covariance measures association between two variables in terms of their deviation from the mean.

$$cov(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

where $cov(x, y)$ is the covariance between variables x and y, n is the number of observations, x_i and y_i are the values of the variables for observation i, and $\bar{x}$ and $\bar{y}$ are the means of the variables.

- Covariance can be positive, negative, or zero.
  - Positive covariance means that as one variable increases, the other variable also increases.
  - Negative covariance means that as one variable increases, the other variable decreases.
  - Zero covariance means that there is no relationship between the variables.

Covariance

- Covariance is not standardized, so it is difficult to interpret.
  - It is affected by the scale of the variables.
  - It is difficult to compare covariances across different datasets.
- One should simply interpret the sign of the covariance, not the magnitude.

Correlation

- Correlation is a standardized version of covariance.
  - It is a measure of the strength and direction of the **linear relationship** between two variables.

$$corr(x, y) = \frac{cov(x, y)}{s_x s_y}$$

- The correlation coefficient is always between -1 and 1.
  - A correlation of 1 means that the variables are perfectly positively correlated.
  - A correlation of -1 means that the variables are perfectly negatively correlated.
  - A correlation of 0 means that there is no linear relationship between the variables.

Correlation

- Notice that the correlation coefficient won't capture non-linear relationships.
  - For example, a correlation of 0 doesn't mean that there is no relationship between the variables, just that there is no linear relationship.

R implementation for covariance and correlation

- The `cov()` function in R calculates the covariance between two variables.
  - It is a base R function.
- The `cor()` function in R calculates the correlation between two variables.
  - It is a base R function.
- Both functions take two vectors as arguments.

Example with `mtcars`

- We can calculate the covariance between `mpg` and `wt` in the `mtcars` dataset.

```
mtcars %>%
    select(mpg, wt) %>%
    cov()
```

```
           mpg        wt
mpg 36.324103 -5.116685
wt  -5.116685  0.957379
```

Example with `mtcars`

- We can calculate the correlation between `mpg` and `wt` in the `mtcars` dataset.

```
mtcars %>%
    select(mpg, wt) %>%
    cor()
```

```
           mpg        wt
mpg  1.0000000 -0.8676594
wt  -0.8676594  1.0000000
```

Correlation matrices

- A correlation matrix is a table that shows the correlation between each pair of variables in a dataset.
- It is a way to summarize the relationships between variables in a dataset
- Often useful as an exploratory tool to understand the relationships between variables before more complex statistical analysis.

Example with mtcars

```
mtcars %>%
    select(mpg, disp, hp, wt) %>%
    cor()
```

```
            mpg       disp         hp         wt
mpg   1.0000000 -0.8475514 -0.7761684 -0.8676594
disp -0.8475514  1.0000000  0.7909486  0.8879799
hp   -0.7761684  0.7909486  1.0000000  0.6587479
wt   -0.8676594  0.8879799  0.6587479  1.0000000
```

Contingency tables

- A contingency table is a table that shows the frequency of each combination of categories in two categorical variables.
- These are also called cross-tabulation tables.

Example with SUPERCIAS data

- We can calculate a contingency table for the `region` and `ultimo_balance` variables in the SUPERCIAS dataset.

```
supercias %>%
    count(region, ultimo_balance)
```

```
# A tibble: 108 x 3
   region ultimo_balance     n
   <chr>  <chr>          <int>
 1 COSTA  1995             122
 2 COSTA  1996             121
 3 COSTA  1997             389
 4 COSTA  1998             310
 5 COSTA  1999             218
 6 COSTA  2000             266
 7 COSTA  2001             200
 8 COSTA  2002             206
 9 COSTA  2003             253
```

R implementations

- We may also feed two columns to the `table()` function to calculate a contingency table.
- Depending on the context, you may want to modify the `margin` argument.
- For example, `margin = 1` would give you the relative frequency of each row.
- `margin = 2` would give you the relative frequency of each column.
- The default is `margin = NULL`, which gives you the relative frequency of the entire table.

Statistical data visualization

Introduction

- Data visualization is an important part of data analysis itself, however, statistical data visualizations are specificially designed to work well with the methods we've just learned in descriptive statistics.
- The following are common data visualization used for descriptive statistics:
  - Histograms/density plots
  - Boxplots
  - Cumulative distribution plots
  - Scatter plots

Histograms

- A histogram is a graphical representation of the distribution of a numerical variable.
- It typically consists of bars that represent the frequency of each interval of the variable.
- The height of the bars represents the frequency of the interval.
- Very important for evaluating distributional shape!

Histograms in R

- The `hist()` function in R is used to create histograms.
  - This is a base R function.
- A tidy alternative is the `geom_histogram()` function from `ggplot2`.
- The `geom_density()` function from `ggplot2` can be used to create a density plot, which is a smoothed version of a histogram.

Example with mtcars

- We can create a histogram for the mpg variable in the mtcars dataset.

```
mtcars %>%
    ggplot(aes(x = mpg)) +
    geom_histogram()
```

Boxplots

- A boxplot is a graphical representation of the distribution of a numerical variable.
- It consists of a box that represents the interquartile range (IQR) of the variable.
- The line in the box represents the median of the variable.
- The whiskers represent the range of the variable, excluding outliers.

Boxplots in R

- The `boxplot()` function in R is used to create boxplots with base R.
- The tidy alternative is the `geom_boxplot()` function from `ggplot2`.

Example with mtcars

```
mtcars %>%
    ggplot(aes(x = 1, y = mpg)) +
    geom_boxplot()
```

Scatter plots

- A scatter plot is a graphical representation of the relationship between two numerical variables.
- It consists of points that represent the values of the two variables.
- The x-axis represents one variable, and the y-axis represents the other variable.
- Scatter plots are useful for identifying relationships between variables.

Scatter plots in R

- The `plot()` function in R is used to create scatter plots with base R.
- The tidy alternative is the `geom_point()` function from `ggplot2`.
  - Sometimes, we include a `geom_smooth()` function to add a regression line to the scatter plot (the "trend line").

Example with mtcars

```
mtcars %>%
    ggplot(aes(x = wt, y = mpg)) +
    geom_point() +
    geom_smooth(method = "lm")
```

Cumulative frequency/distribution plots

- A cumulative frequency plot is a graphical representation of the cumulative frequency of a numerical variable.
- It consists of a line that represents the cumulative frequency of the variable.
- The x-axis represents the variable, and the y-axis represents the cumulative frequency.
- Cumulative frequency plots are useful for understanding the distribution of a variable.
- Typically, we may want to group the variable into intervals to create a cumulative frequency plot.

Cumulative frequency in a table of frequencies

- We can calculate the cumulative frequency of a variable by summing the frequencies of the variable up to a certain point.
- This can be done by creating a cumulative sum of the frequencies.
- The cumulative frequency of a variable is the sum of the frequencies of the variable up to that point.

Example with SUPERCIAS's capital_suscrito

- We can calculate the cumulative frequency of the capital_suscrito variable in the SUPERCIAS dataset.
- We will first group the variable into intervals, then calculate the cumulative frequency.

```
supercias_with_cumulative <-
    supercias %>%
    mutate(capital_suscrito_interval = cut(capital_suscrito, bre
    count(capital_suscrito_interval) %>%
    mutate(cumulative_frequency = cumsum(n))
```

Example with SUPERCIAS's capital_suscrito

```
# A tibble: 11 x 3
   capital_suscrito_interval      n cumulative_frequency
   <fct>                      <int>                <int>
 1 (0,1e+07]                208714               208714
 2 (1e+07,2e+07]               229               208943
 3 (2e+07,3e+07]                93               209036
 4 (3e+07,4e+07]                45               209081
 5 (4e+07,5e+07]                16               209097
 6 (5e+07,6e+07]                14               209111
 7 (6e+07,7e+07]                14               209125
 8 (7e+07,8e+07]                 5               209130
 9 (8e+07,9e+07]                 4               209134
10 (9e+07,1e+08]                 8               209142
11 <NA>                        476               209618
```

Example with SUPERCIAS's capital_suscrito

- The relative frequency of each cumulative step can be calculated by dividing the cumulative frequency by the total number of observations.
- This is the cumulative relative frequency. It will be between 0 and 1 for each step.
- The cumulative relative frequency for the last step will always be 1.