Summer School of Modern Methods in Biostatistics and Epidemiology

Basics of Stata

Robert Thiesmeier
Karolinska Institutet

robert.thiesmeier@ki.se

Course description

This course aims at introducing the fundamental elements of the statistical software Stata. The course follows a practical syllabus where students learn basic Stata commands by directly applying them to health related data questions. The lectures will demonstrate the use of Stata code. At the end of this one-day course, students should be able to:

- Write reproducible code in a do-file
- Describe, manage, summarize and restructure data
- Visualise, analyse, and program data

Course overview

Morning

09.00 – 10.30	*Lecture and practical I*: Introducing Stata
10.30 – 11.00	Coffee break
11.00 – 13.00	*Lecture and practical II*: Handling data
13.00 – 14.00	**Lunch**

Afternoon

14.00 – 15.00	*Lecture and practical III*: Visualising data
15.00 – 15.30	Coffee break
15.30 – 17.00	*Lecture and practical IV*: Analysing and programming data

Introducing Stata

How to get started with Stata?

Stata is a complete, integrated software package that is broadly used for data science including data manipulation, visualization, statistics, programming, and reproducible reporting.

It is easy to use: https://www.stata.com/why-use-stata/

Stata has a lot of resources that are free to use:
https://www.stata.com/links/resources-for-learning-stata/

Speaking Stata

As with any language, it will take some time to get used to speaking and writing Stata.

Those with experience in other languages such as R, Python, SAS, SPSS might find it easier to follow.

Each command is like a vocabular – documentation for each command is available for the `syntax` of the command including options and examples.

The Stata Base Reference Manual contains almost 4000 pages incl. methods, formulas, and examples. Every command has documentation – **`help`**

Speaking Stata: operators

Stata works with multiple operators that you can use to specify certain conditions:

Arithmetic	**Logical**	**Relational**
+ addition	**&** and	**>** greater than
- subtraction	**|** or	**<** less than
* multiplication	**!** not	**>=** **>** or equal
/ division		**<=** **<** or equal
^ power		**==** equal
- negation		**!=** not equal
= equal		

Motivating example

We will use a birth cohort dataset including data on children's birthweight and information on maternal smoking and medication prescription during pregnancy, and height and weight of the mother. The dataset can be accessed in Stata or Excel format.

Lectures: `use data_birthcohort.dta, clear`
Practicals: `use data_tutorial.dta, clear`

Basic commands for data management

1. How do you open a new writing file? (**`doedit`**)
2. How do you set a working directory? (**`pwd, cd`**)
3. How do you open a data set? (**`use, import`**)
4. How do take a first look at the data?
   (**`codebook, list, browse, if, sort, count`**)
5. How do you describe a variable? (**`describe`**)
6. How do you obtain summary information? (**`summarize`**)
7. How do you add new information to your existing data set?
   (**`merge, append`**)
8. How do you input data yourself? (**input**)
9. How do you save your data? (**`save, export, replace`**)

Practical I

How to get the data

Download the dataset for the practical:

```
use
"https://raw.githubusercontent.com/robertthiesmeier/summer_school_2025/m
ain/data_tutorial.dta", clear
```

Download the dataset for the lecture:

```
use
"https://raw.githubusercontent.com/robertthiesmeier/summer_school_2025/m
ain/data_birthcohort.dta", clear
```

Practical session

1. Open and briefly describe the content of the data (number of participants, variables etc.) (`data_tutorial.dta`)
2. Describe the baseline age of participants (unique values, range, min, max)
3. List `id`, `CHD status`, and `age` of those subjects 59 years of age or more and who did not smoke cigarettes.
4. Export the dataset into an Excel file.
5. List `CHD status, smoke, age` of the first 5 participants. Input the data with the `input` command.

Handling data

Handling data in Stata

We continue to work with the birth cohort data set.

1. How do we count data? (**`count`**)
2. How can we display results and expressions? **`(display)`**
3. How do you produce one-way/two-way tables? (**`tabulate`**)
4. How do you create a stratified table with descriptive results? (**`tabstat`**)
5. How do you build a new variable? (**`rename`**, **`generate`**, **`egen`**)
6. How do you define and attach labels? (**`label`**)
7. How do you replace or drop values? (**`replace, drop, keep`**)

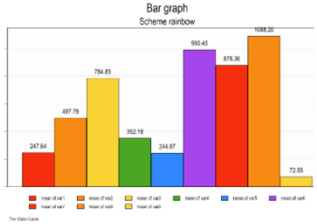
Practical II

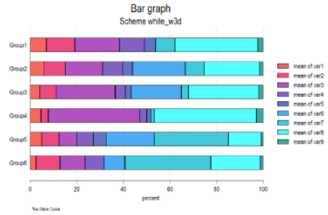
Practical session

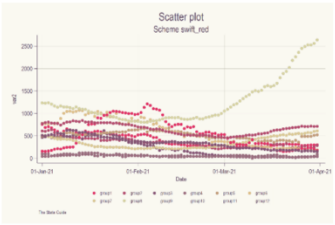
1. Count the number of new CHD cases occurred during the follow-up time. Display the incidence of CHD as percentage.
2. Produce a one-way table of counts of smoking status. Display the percentage of smokers.
3. Count the number of CHD cases among smokers and non-smokers.
4. Count the individuals smoking more than 20 cigarettes per day.
5. Describe the distribution of baseline age.
6. Produce a two-way table of counts between categorized age and CHD. Display the percentage of incidence CHD by categories of age.
7. Produce a table of descriptive statistics (median, 25th percentile, and 75th percentile) of systolic blood pressure by CHD status.
8. Dichotomize diastolic blood pressure (above/below median) and examine the observed CHD risk in the two groups.

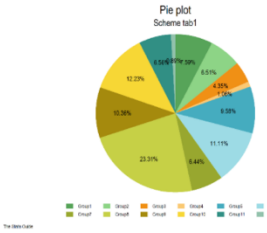
Visualising data

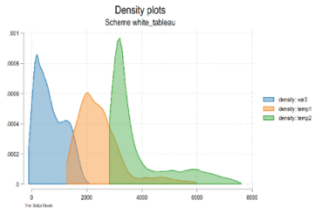
Data visualisation with Stata

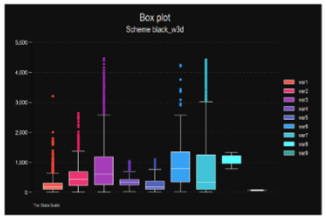


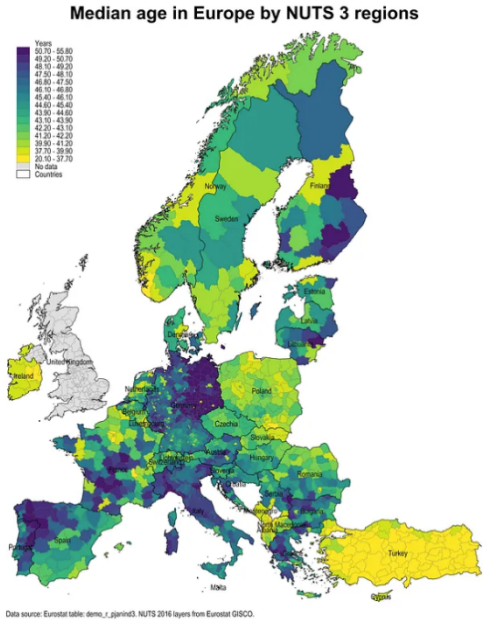


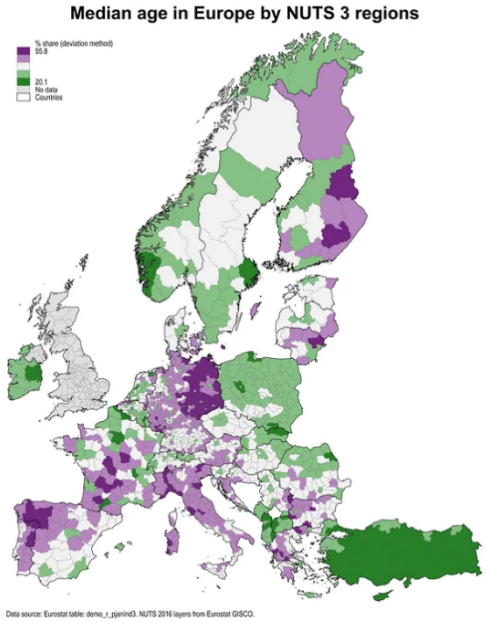


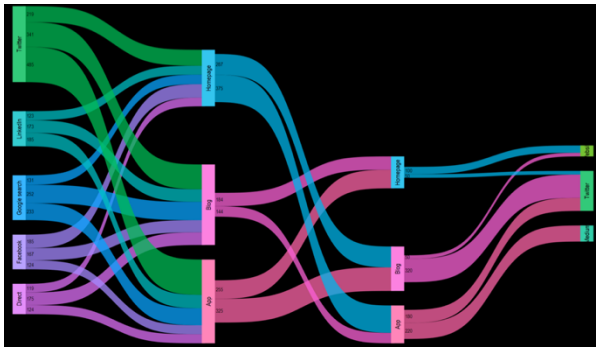












The anatomy of a graph

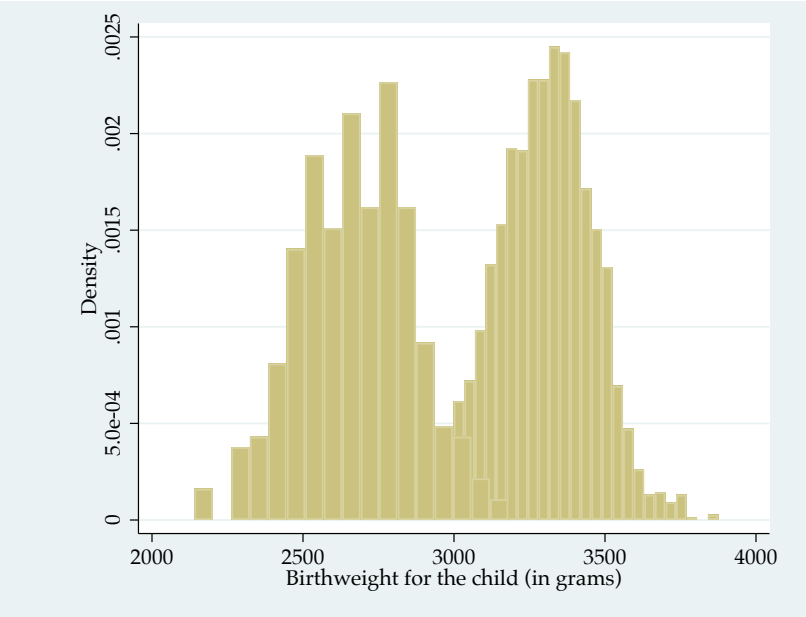
An overview for Stata graphs:
https://www.stata.com/support/faqs/graphics/gph/stata-graphs/

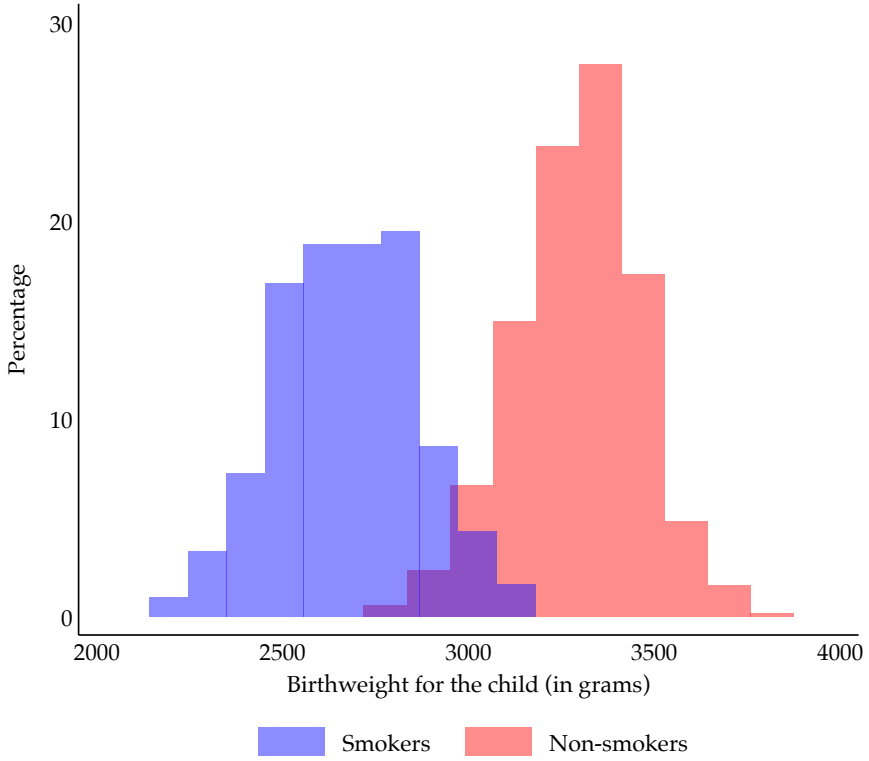
User-written packages for more colour-palettes (e.g., `schemepack`).

`twoway graph` is a family of graphs with a lot of options that produces publication-ready figures.

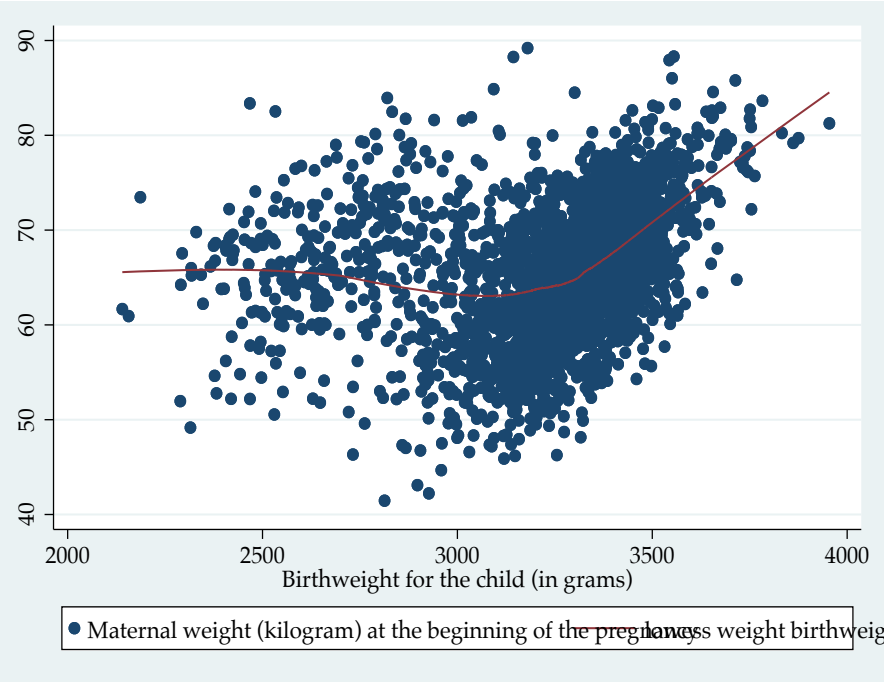
```
twoway ///
    (plottyple 1, options) ///
    (plottyple 2, options ///
    … add as many types as you want to
    , options for the overall graph
```

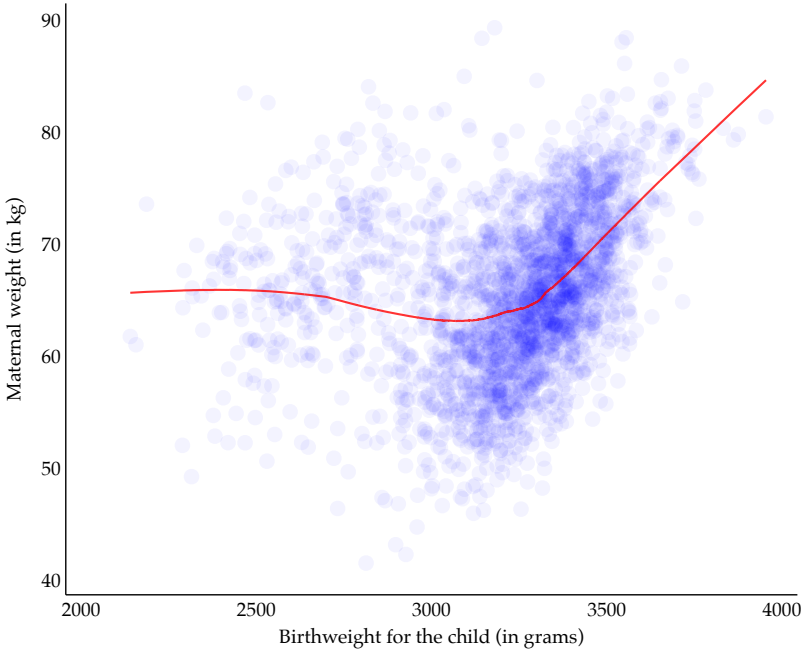
Visualising data in Stata





Visualising data in Stata





Visualising data in Stata

1. How do you create a scatter plot? (**`scatter`**)
2. How do you plot a smoothed average? (**`lowess`**)
3. How do you plot a bar chart? (**`bar`**)
4. How do you create a histogram? (**`histogram`**)
5. How do you plot a kernel density? (**`kdensity`**)
6. How do you combine multiple graphs? (**`twoway`**)
7. How do you change options? (**`ylabel, xlabel, ytitle, xtitle, legend, color`**)

Practical III

Practical session

1. Produce a scatter plot of weight and height. Use options to control various elements of the graph.
2. Overlay a smoothed trend (`lowess`) with the scatter plot. Remove the legend.
3. Produce a histogram of the body mass index distribution. Control various elements of the graph.
4. Overlay a smoothed histogram (kernel density) of the total cholesterol distribution among cases and non-cases of CHD. Control various elements of the graph.
5. Produce a `twoway` bar chart showing the CHD risk in the sample as function of BMI categorized.

Analysing and programming data

Regression analysis

The aim of statistics is often inference.

Stata offers a variety of different statistical models and options.

For regression analyses, a simplified structure looks as follows:

Regression method outcome [**covariate 1 covariate 2** …] [, **options**]

```
logit depvar [indepvars] [, options]
linear depvar [indepvars] [, options]
qreg depvar [indepvars] [, options]
stcox [indepvars] [, options]
```

Programming data

Basic commands for programming

Stata can be used for simple and advanced programming. Some basic commands are shown in the table below.

do-file	`do` `run`
saved results	`return list` `ereturn list`
Macros	`local`
Looping	`foreach` `forvalues`
Programming	`scalar`

Basic commands for programming

We will look at a few commands in practice.

1. How to access saved results? (**`return`**)
2. How to store content? (**`scalar`**, macros)
3. How to create a loop? (**`foreach, forvalues`**)
4. How to create a program? (**`program`**)

Practical IV

Practical Session

1. Run a logistic regression model (`logit`) on `chd69` as the outcome and `smoke` as the main exposure. Access and save the coefficient for `smoke` from the estimation results.
2. Create a list of covariates (`smoke, age, sbp, bmi`) and loop them over to run a univariate logistic regression with `chd69` as the outcome.
3. Write a short program that takes one covariate as input, runs a univariate logistic regression, and displays the odds ratio.

Final thoughts

At the end of this course, you should be able to:

- ✅ Write reproducible code in a do-file
- ✅ Describe, manage, summarize and restructure data
- ✅ Visualise, analyse, and program data

…Learning a new language takes time

…Practice often and repeat the basics

…Enjoy!