Crash course on data visualisation with Gretl

In this exercise, you will learn how to create different types of plots using Gretl. You will learn how to create basic plots such as histograms, scatter plots, and box plots etc. You will also learn how to customise these plots to make them more informative and visually appealing.

We are going to use the *housing* dataset. This dataset contains 546 observations on house sales in Windsor, Canada. The data is stored in the file `housing.csv` or `housing.gdt` (Gretl's native datatype).

Variables of the dataset

- `price`: sale price of a house
- `lotsize`: lot size of a property in square feet
- `bedrooms`: number of bedrooms
- `bathrms`: number of full bathrooms
- `stories`: number of stories excluding basement
- `driveway`: does the house have a driveway? 1 if True, otherwise 0
- `recroom`: does the house have a recreation room? 1 if True, otherwise 0
- `fullbase`: does the house have a full finished basement? 1 if True, otherwise 0
- `gashw`: does the house use gas for hot water heating? 1 if True, otherwise 0
- `airco`: does the house have central air conditioning? 1 if True, otherwise 0
- `garagepl`: number of garage places
- `prefarea`: is the house located in the preferred neighbourhood of the city? 1 if True, otherwise 0

Load some useful packages

For this exercise, we will use the some user-contributed Gretl packages. These can be obtained from the Gretl server as follows (only needed once!):

```
pkg install fdensity
pkg install PairPlot
pkg install Datatools
```

Next, you need to load the packages into memory. You can do this by running the following code:

```
include fdensity.gfn
include PairPlot.gfn
include Datatools.gfn
```

Load the dataset and have a first look

First, you need to load the dataset into Gretl. You can use the following code to load the dataset. **Note:** You need to replace `<PATH TO FOLDER>` with the actual path to the folder where the dataset is stored.

```
string FILENAME = "<PATH TO FOLDER>\housing.gdt"

open "@FILENAME"
```

Let's have a first look at the dataset. You can use the following code to display the first and last few observations of the dataset.

```
print dataset --byobs --range=1:5
print dataset --byobs --range=-5:
```

Visualising distributions

Histograms

Histograms are a great way to visualise the distribution of a single variable. You can use the following code to create a histogram of the variable `price`.

```
freq price --plot=display
```

This will create the following histogram:

As you can see, the histogram shows the distribution of the variable `price`. The x-axis represents the price of the houses, while the y-axis represents the frequency of the prices. The histogram is divided into bins, and the height of each bin represents the frequency of the prices in that bin.

Most of the prices are between 40,000 and 80,000. There are a few houses with prices above 100,000.

Unfortunately, currently you can't really manipulate the histogram in Gretl. However, there exist alternatives as we are going to see.

Boxplots

Boxplots are a powerful tool to visualise and analyse the distribution of a single variable. You can use the following code to create a boxplot of the variable `price`.

```
boxplot price --output=display
```

This will produce a simple boxplot.

Let's try to improve the boxplot by adding some customisation. You can use the following code to create a customised boxplot of the variable `price`.

```
boxplot price --output=display \
 { set title "Boxplot of Price";\
   set ylabel "Price";\
   set ytics 20000;\
   set grid;}
```

The customised boxplot looks like this:

"The cheapest house costs $25,000 and the most expensive around $190,000. 25% of the houses cost a maximum of $50,000 and the top 25% of properties cost at least $80,000. The middle 50% of houses cost between $50,000 and $80,000. There are a few extremely high-priced houses that cost more than $125,000."

Grouped boxplots

Often you want to compare the distribution of a variable across different groups. You can use the following code to create a grouped boxplot of the variable `price` by the variable `prefarea`.

```
boxplot price prefarea --factorized --output=display \
 { set title "Boxplot of Price grouped by 'prefarea'";\
   set ylabel "Price";\
   set ytics 20000;\
   set grid;}
```

Houses in preferred residential areas tend to be more expensive than houses in other areas. The middle 50% of houses in preferred areas cost between $70,000 and $100,000, but in non-preferred areas, they only cost between $50,000 and $75,000. Interestingly, there are also houses in non-preferred areas that cost between $110,000 and about $175,000.

Visualising categorical variables

So called swarm plots are a great way to visualise the distribution of a continuous variable across different categories. You can use the following code to create a swarm plot of the variable `lotsize` by the variable `prefarea`.

```
gnuplot lotsize prefarea --fit=none --output=display \
  { set jitter overlap 0.05;}
```

The resulting plot is:

It is evident that the majority of houses in non-preferred areas have rather small plots between 2,500 and 5,000 square feet. In preferred areas, the plots tend to be larger and range between 5,000 and 7,500

square feet. However, there are also houses in non-preferred areas with very large plots of more than 7,500 square feet.

Kernel density plots

Instead of a discrete histogram, you can also create a continuous kernel density plot.

Built-in command

You can use the following code to create a kernel density plot of the variable `price`.

```
kdplot price --output=display
```

This will produce the following kernel density plot:

Grouped kernel density plots

You can also create grouped kernel density plots to compare the distribution of a variable across different groups.

For this, we make use of the `fdensity` package. You can use the following code to create a grouped kernel density plot of the variable `price` by the variable `airco`.

```
include fdensity.gfn  # load package into memory

# Customize the plot
bundle Custom = _(add_opts = "set grid; set key top right; set logscale x
2")
fdensity(price, airco, Custom)
```

The resulting plot is:

Scatter plots

Scatter plots are a great way to visualise the relationship between two continuous variables. You can use the following code to create a scatter plot of the variables `lotsize` and `price`.

```
gnuplot price lotsize --fit=linear --output=display \
  {set title "Relationship between price and lotsize";\
   set grid;}
```

This creates the following scatter plot:

Grouped scatter plots

You can also create grouped scatter plots to compare the relationship between two continuous variables across different groups.

The following code creates a grouped scatter plot of the variables `price` and `lotsize` by the variable `airco`:

```
gnuplot price lotsize airco --dummy --output=display \
  {set title "Relationship between price and lotsize by airco";\
   set grid;}
```

And the plot looks like this: