Lab

Building plots: facets

Lab objectives:

- To plot several variables in the same plot.
- To create visualisations using facets.

Type all the examples in this lab and check the results to ensure you understand the concepts being introduced. Exercises are at the end of this lab.

ggplot2

In this lab, we are using ggplot2. Ensure that you load this library.

Data

Note: in this lab, as in previous labs, we will assume that we have perfect data which we do not need to transform. The datasets to be downloaded have already been manipulated to give the perfect data for our purposes.

You need the following files from CampusMoodle **which were shared in previous weeks**:

- *provenOilReserveWEurope.csv*: contains data regarding proven oil reserves for western European countries.
- *channels*.csv: contains perceived sensation vs stimuli for various channels.
- *ComsumptionTransposed*.csv.
- *ConsumptionBySource.csv.*

Load these datasets.

Plotting several variables

Load the channels dataset. We would like to see how electric, length and area compare in terms of size of the stimuli vs amount of sensation perceived.

Try the following

```
p <- ggplot(channels, aes(n))
p <- p + geom_line(aes(y = electric, colour= "electric"))
p <- p + geom_line(aes(y = length, colour ="length"))
p <- p + geom_line(aes(y = area, colour= "area"))
p <- p+labs( colour="channels")
p
```

Note that in the above code colour = "electric" means attach label "electric" to the legend for this line's colour.
Note also that the second to last statement (the one about the labs setting colour) attaches title "channels" to the colour legend.

The above plot lacks a title. We can include one by inserting the following line as the 2nd last line:

```
p <- p + labs(title="Different channels sensation according to level of stimuli")
```

Unfortunately, the plot we obtain has "electric" as the label for the y axis. We can change the label, as explained in lab 4. For example, by adding the code below as the 2nd last line

```
P <- p + labs(x= "Size of stimuli", y= "Amount of sensation")
p
```

Try the code above but replace the legend labels with more explanatory labels, e.g.

```
p <- ggplot(channels, aes(n))
p <- p + geom_line(aes(y = electric, colour = "electric current"))
p <- p + geom_line(aes(y = length, colour = "object's length"))
p <- p + geom_line(aes(y = area, colour = "object's area"))
p <- p + labs(colour="channels")
p <- p + labs(title="Different channels sensation according to level of stimuli")
p <- p + labs(x= "Size of stimuli", y= "Amount of sensation")
p
```

The plot above is not good for appreciating the difference between the channels where the sensation is small when compared to electricity. Change the y axis upper limit to 50 so that you can appreciate the difference. For example,

```
p <- ggplot(channels, aes(n))
p <- p + geom_line(aes(y = electric, colour = "electric current"))
p <- p + geom_line(aes(y = length, colour = "object's length"))
p <- p + geom_line(aes(y = area, colour = "object's area"))
p <- p + labs(colour="channels")
p <- p + labs(title="Different channels sensation according to level of stimuli")
p <- p + labs(x= "Size of stimuli", y= "Amount of sensation")
p <- p + ylim(0,50)
p
```

You will get a warning explaining that some values are lost – the "electric" values which are outside the plot area.

The alternatives are to exclude the "offending electric " column from the dataset or to restrict the rows to those where no value exceeds 50 for electric.

To remove electric

```
channels$electric <-NULL
p <- ggplot(channels, aes(n))
p <- p + geom_line(aes(y = length, colour = "object's length"))
p <- p + geom_line(aes(y = area, colour = "object's area"))
p <- p + labs(colour="channels")
p <- p + labs(title="Different channels sensation according to level
of stimuli")
p <- p + labs(x= "Size of stimuli", y= "Amount of sensation")
p
```

To restrict the rows shown to those with no electric value over 50 and put the data in variable channels2

```
channels2 <- channels |> filter(electric < 50)
```

We can now plot this new dataset

```
p <- ggplot(channels2, aes(n))
p <- p + geom_line(aes(y = electric, colour = "electric current"))
p <- p + geom_line(aes(y = length, colour = "object's length"))
p <- p + geom_line(aes(y = area, colour = "object's area"))
p <- p + labs(colour="channels")
p <- p + labs(title="Different channels sensation according to level
of stimuli")
p <- p + labs(x= "Size of stimuli", y= "Amount of sensation")p
```

Facets

Load the provenOilReserveWEurope dataset.

Plot the various variables which you want to compare in separate plots. For example, try

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_line(aes(y = MT.Barrels))
p <- p + facet_grid(. ~ Country)
p <- p + labs(title="Proven oil reserves in Western Europe", y =
"Billion barrels of oil equivalent")
```

```
p
```

It may be easier to compare the plots if they are placed vertically in several rows instead of horizontally in one row. Try

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_line(aes(y = MT.Barrels))
p <- p + facet_grid(Country ~ .)
p <- p + labs(title="Proven oil reserves in Western Europe", y =
"Billion barrels of oil equivalent")
p
```

We could make each plot a different colour, according to the value of the Country field.

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_line(aes(y = MT.Barrels, colour=Country))
p <- p + facet_grid(Country ~ .)
p <- p + labs(title= "Proven oil reserves in Western Europe",
           y = "Billion barrels of oil equivalent",
           colour = "European country")
p
```

We could let R decide which "y" scale is best for each of the country's plot using the "scales" option. For example,

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_line(aes(y = MT.Barrels, colour=Country))
p <- p + facet_grid(Country ~ ., scales="free_y")
p <- p + labs(title= "Proven oil reserves in Western Europe",
           y = "Billion barrels of oil equivalent",
           colour = "European country")
p
```

Although this makes it more difficult to compare values between countries, it makes it easier to see what is happening with each country.

We could emphasize the different countries by changing the linetype for each of them. For example,

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_line(aes(y = MT.Barrels,
                       colour=Country,linetype=Country))
p <- p + facet_grid(Country ~ ., scales = "free")
```

```
p <- p + labs(title= "Proven oil reserves in Western Europe",
           y = "Billion barrels of oil equivalent",
           colour = "European country",
           linetype = "European country")
p
```

We could replace the lines by points, e.g.

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_point(aes(y = MT.Barrels, colour=Country))
p <- p + facet_grid(Country ~ ., scales="free_y")
p <- p + labs(title= "Proven oil reserves in Western Europe",
           y = "Billion barrels of oil equivalent",
           colour = "European country")
p
```

Or we could have lines and points together

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_point(aes(y = MT.Barrels, colour=Country))
p <- p + geom_line(aes(y = MT.Barrels, colour=Country))
p <- p + facet_grid(Country ~ ., scales="free_y")
p <- p + labs(title="Proven oil reserves in Western Europe",
           y = "Billion barrels of oil equivalent",
           colour = "European country")
p
```

We may wish to give more vertical (y axis) space to some of the plots. We can do this by defining the "space" as "free_y". Other possibilities include "free_x" (for flexible x axis space) and "free (for flexible x and y axes).

```
p <- ggplot(ProvenOilReserveWEurope, aes(Year))
p <- p + geom_point(aes(y = MT.Barrels, colour=Country))
p <- p + geom_line(aes(y = MT.Barrels, colour=Country))
p <- p + facet_grid(Country ~ ., scales = "free", space="free_y")
p <- p + labs(title= "Proven oil reserves in Western Europe",
           y = "Billion barrels of oil equivalent",
           colour = "European country")
p
```

Exercises

1. Produce a scatterplot Oil, Gas, Coal and Renewables using the consumptionTransposed dataset. The unit of measure is million tons of oil equivalent for each of the energy sources Ensure appropriate titles and labels. Is a scatterplot the most suitable plot or would a line plot be better?
2. Visualise the consumptionBySource dataset. Use facets according to the value of Source. Use a line geom, where the colour of the line depends on the value of the source. Ensure that the Y axis scale is free (i.e. it varies according to the values of the facet being displayed.) Ensure appropriate titles and axis labels are presented.
3. Repeat the exercise above, but this time use a scatterplot (point geom) where the shape is a triangle. Use both `colour` and `fill` to specify the colour of the triangles (within the `aes` of `geom_point.` Use also `colour` and `fill` to specify the legend within l`abs`. Discuss which visualisation is best, the one in exercise 2 or this one. Justify your answer.