The GGFormula system for graphics based on a dataset

- There are many different graphics systems in R. GGFormula is an extension of the ggplot[1] package which provides a nicer formula-based[2] but also pipe-based[3] workflow.
- The basic syntax for ggformula graphics is as follows:

```
gf_graphtype(formula, data=..., other parameters)
gf_graphtype(formula, data=..., other parameters) %>% otherCustomizations
dataset %>% gf_graphtype(formula, other parameters) ...
```

- The ggformula methods allow for the dataset to be entered either via pipe system or via the data =... parameter.
- The parameters can be used to customize the graph[4], and the choices differ somewhat from method to method, although there are a number of common entries.
- A lot of graph addons[5] can be used via the piping mechanism, to add elements to a graph (or to overlay two graphs).
- Here is a list of the most standard basic graph commands. Most of these would generate a horizontal plot instead of a vertical one by addin the letter h to the end of the name.
  - gf_bar and gf_barh generate barcharts
  - gf_percents, gf_percentsh, gf_props and gf_propsh use proportions and percents instead of counts.
  - gf_histogram and gf_histogramh generate histograms
  - gf_dhistogram and gf_dhistogramh generate histograms based on density instead of count. These are well set up to overlay with density plots.
  - gf_point generates xy-plots.
  - gf_boxplot and gf_boxploth generate boxplots.
  - gf_dens and gf_density generate so-called “density plots”.

Example:

```
library(hanoverbase)
driving <- read_csv("driving.csv", col_types = cols(day = col_date(format = "%Y/%m/%d"),
                                                    arrTime=col_time(format="%H:%M"),
                                                    leaveTime=col_time(format="%H:%M")))
driving %>% gf_barh(~weekDay)
driving %>% gf_percentsh(~weekDay)
driving %>% gf_histogram(~miles)
driving %>% gf_histogram(~miles|direction)
driving %>% gf_dhistogram(~miles|direction) %>% gf_dens(~miles|direction)
driving %>% filter(miles <= 48) %>% gf_density(~miles, fill=~direction)
driving %>% filter(miles <= 48) %>% gf_boxploth(direction~miles)
driving %>% gf_boxploth(~miles|direction)
driving %>% filter(miles <= 48) %>% gf_point(time~miles, color=~direction)
```

[1] ggplot.html
[2] formulas.html
[3] piping.html
[4] ggformulaParameters.html
[5] ggformulaAddons.html