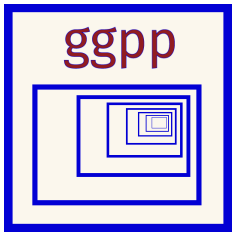

Plot insets and data labels with ggpp :: CHEAT SHEET

Basics

ggpp extends **ggplot2** and the **grammar of graphics** that it is based on. Plot construction remains unchanged but new **geometries**, **scales** and **positions** support the use of the same grammar as for observations for *data labels* and *annotations*. In the case of data labels, which are linked to observations, their, usually displaced, positions are meaningful in data units. Textual annotations and insets are, in contrast, linked to a plot as a whole, and their location within the plotting area decided on graphic design basis. True annotations and insets can be easily separated from the plot, while, data labels, cannot.

Plots, tables and grobs as labels

A different plot, a table or an illustration set within the plotting area of a larger plot is called an *inset* when they are annotations, but if small, they can also function as data labels.

geom_plot(), **geom_table()**, and **geom_grob()** differ from ggplot's **geom_text()** mainly in that instead working by mapping a `character` vector to the `label` aesthetic, they require a list of gg objects, a list of `data.frame` objects or a list of grob objects to be mapped to `label`. While all geoms return trees or lists of grobs (or graphical objects) as defined in package 'grid', **geom_grob()** accepts a list of grobs or grob trees mapped to the `label` aesthetic and collects them into a plot layer. (grobs can be built in advance using drawing functions or constructed by converting raster or vector graphics objects from other formats into grobs.)

As in **geom_text()** `hjust` and `vjust` control the justification of each inset in whole. Rotation with `angle` is supported. Size can be adjusted in proportion to the plotting area with aesthetics `vp.width` and `vp.height`. Aesthetic `size` is not supported. **geom_table()** accepts table themes (**ttheme**s), of which several are provided. **stat_fmt_tb()** makes it possible to select part of a data frame and/or replace headings when adding it as an inset with **geom_table()**.

ttheme_gtdefault()
ttheme_gtminimal()
ttheme_gtbw()
ttheme_gtplain()
ttheme_gtdark()
ttheme_gtlight()
ttheme_gtsimple()
ttheme_gtstripes()

Annotations

True annotations can be added using data coordinates or native plot coordinates (NPC) depending on the situation. NPC coordinates are specially useful when adding annotations and insets to plots in which the limits of the plotting area vary among panels or when plotting different data sets. New geometries and *pseudo*-scales from **'ggpp'** extend the grammar of graphics to support NPC similarly as it supports observations plotted according to data cooedinates. Layers created with geometries with names ending in _npc use the *pseudo*-aesthetics npcx and npcy instead of aesthetics x and y, both with values in the range $0 \ldots 1$ and no support for transformations. These geometries add a layer with annotations at positions relative to the extent of the plotting area. A wrapper on **annotate()** from **'ggplot2'** adds to it support for npcx and npcy.

geom_text_npc()
geom_text_npc()
geom_plot_npc()
geom_table_npc()
geom_grob_npc()
annotate()
scale_npcx_continuous()
scale_npcy_continuous()

Data labels

If the coordinates are displaced with a position function that keeps the original ones, these geometries can draw a line segment or an arrow to link the added graphical elements to the original position, before displacement, position. They also differ from ordinary geometries in that the mapped colour, fill and transparency can be applied selectively to different elements of the graphical objects plotted.

geom_label_s()
geom_text_s()
geom_point_s()

Reference lines and marginal marks

It is frequently useful to highlight specific values along the **x** or **y** axes, at the edge of the plotting area or across the plotting area, similarly to how **geom_rug()**, and **geom_hline()** or **geom_vline()** work, respectively.

geom_x_margin_point()
geom_y_margin_point()
geom_x_margin_arrow()
geom_y_margin_arrow()

geom_x_margin_grob()
geom_y_margin_grob()
geom_quadrant_lines()
geom_vhlines()

Subsetting data labels

When adding data labels to data-dense scatter plots we may want to only add labels to observations on the edges of the cloud of observations. If we use the repulsive geometries **geom_text_repel()** and **geom_label_repel()** from 'ggrepel' even if we do not add labels to all observations, we want the labels no to occlude any observation, labelled or not. These geometries will not include a data label in a layer if the text is "" but will take into account the position of the observation when repulsing other labels. To automate the selection of the observations to be labelled, we use statistics **stat_dens1d_labels()** or **stat_dens2d_labels()** compute the local density of observations, and replace some text labels by "" based on the density. Alternatively, we can use a *filter* statistic to subset (or filter) the rows in **data** based on the local density, computed globally for a plot panel or by group.

stat_dens1d_labels()
stat_dens1d_filter()
stat_dens1d_filter_g()
stat_dens2d_labels()
stat_dens2d_filter()
stat_dens2d_filter_g()

Summaries

In most cases **stat_summary()** from 'ggplot2' has the capabilities we need. Package 'ggpp' adds statistics that cater for special cases: summarizing both **x** and **y** in the same layer and computing summaries per quadrant of a plot. Finally **stat_apply_group()** applies a function to the **data** by groups, making it possible to apply functions to compute running and cumulative summaries.

stat_panel_counts()
stat_group_counts()
stat_quadrant_counts()
stat_apply_group()
stat_summary_xy()
stat_centroid()

Computed nudging

In package 'ggplot2' nudging is applied with function **position_nudge()** based on constant values supplied by the user, and the new positions replace the original ones in **data**. Package 'ggpp' provides positions with which the displacements are computed either based on the observations being plotted or based on reference positions supplied by the users, and additionally the original positions renamed in **data** as `x_orig` and `y_orig`.

position_nudge_to()
position_nudge_center()
position_nudge_line()

Combined positions

There are cases when it is useful to apply nudging in addition to jitter, dodging or stacking. A typical case is adding data labels. In this case the position functions apply two separate displacements of the position, one after another. The ones kept as `x_orig` and `y_orig` can be selected bt the user to be either the initial one or the one after the first displacement.

position_stack_nudge()
position_fill_nudge()
position_dodge_nudge()
position_dodge2_nudge()
position_jitter_nudge()

Positions with *memory*

Contrary to positions from 'ggplot2', all positions defined in package 'ggpp' retain in **data** a copy of the original positions of the observations renamed as `x_orig` and `y_orig`, and are thus compatible both with the repulsive geometries from package 'ggrepel' and all geometries from **'ggplot2'** and its extensions. When they are used with **geom_text_repel()** and **geom_label_repel()** from **'ggrepel'** or with the non-repulsive geometries from package **'ggpp'** segments or arrows connecting with the original positions of the observations to the displaced data labels are drawn.

position_stack_keep()
position_fill_keep()
position_dodge_keep()
position_dodge2_keep()
position_jitter_keep()

Avoiding overlaps

Automatically avoiding overlaps among data labels and between data labels and observations is not an easy task. The repulsive geometries from package **'ggrepel'** frequently, but not always, do a very good job on their own. However, in other cases combining these geometries with computed nudging by **positions** and density-based subsetting by **statistics** from **'ggpp'** can improve the outcome. The development of packages **'ggrepel'** and **'ggpp'** has been done in coordination, to allow them to work synergistically. I acknowledge the contributions by Kamil Slowikosky, the author of **'ggrepel'**, to **'ggpp'** as well as his willingness to make the needed changes to **'ggrepel'** to ensure inter-compatibility.