Package: mobileCharts (via r-universe)

September 24, 2024

Title Mobile Charts

Version 0.0.1

Description Create charts optimized for mobile.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports shiny, purrr, rlang, ggplot2, magrittr, htmltools, htmlwidgets

RoxygenNote 7.0.2

Repository https://tanho63.r-universe.dev

RemoteUrl https://github.com/tanho63/mobileCharts

RemoteRef HEAD

RemoteSha ece3a75fadbcf74ed812fb24dd410e6f721bf29c

Contents

axis *Axis*

Description

Customise axis.

Usage

```
mobile_hide_axis(m)

mobile_axis_x(m, ...)

mobile_axis_y(m, ...)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`...`	Named options from the official documentation.

Examples

```
df <- data.frame(
 label = c("banana", "apple", "cake"),
 value = c(50, 30, 20),
 x = "1" # phoney variable
)

mobile(df, aes(x, value, color = label, adjust = stack)) %>%
 mobile_bar() %>%
 mobile_coord("polar", transposed = TRUE) %>%
 mobile_hide_axis()
```

geom *Geoms*

Description

Add a geometry.

Usage

```
mobile_bar(m, ..., data = NULL, inherit_aes = TRUE)

mobile_point(m, ..., data = NULL, inherit_aes = TRUE)

mobile_path(m, ..., data = NULL, inherit_aes = TRUE)

mobile_line(m, ..., data = NULL, inherit_aes = TRUE)

mobile_area(m, ..., data = NULL, inherit_aes = TRUE)

mobile_polygon(m, ..., data = NULL, inherit_aes = TRUE)

mobile_schema(m, ..., data = NULL, inherit_aes = TRUE)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`...`	Aesthetics.
`data`	A data.frame.
`inherit_aes`	Whether to inherit the aesthetics from `mobile`.

Examples

```
mobile(cars) %>%
 mobile_point(aes(speed, dist))
```

`mobile` *Initialise*

Description

Initialise a chart.

Usage

```
mobile(data = NULL, ..., width = NULL, height = NULL, elementId = NULL)
```

Arguments

`data`	A data.frame.
`...`	Mobile chart aesthetics.
`width, height`	Must be a valid CSS unit (like `'100%'`, `'400px'`, `'auto'`) or a number, which will be coerced to a string and have `'px'` appended.
`elementId`	ID of element holding the mobile chart.

Examples

```
mobile(cars) %>%
 mobile_point(aes(speed, dist))
```

---

mobile-shiny *Shiny bindings for mobile*

Description

Output and render functions for using mobile within Shiny applications and interactive Rmd documents.

Usage

```
mobileOutput(outputId, width = "100%", height = "400px")

render_mobile(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	output variable to read from
`width, height`	Must be a valid CSS unit (like `'100%'`, `'400px'`, `'auto'`) or a number, which will be coerced to a string and have `'px'` appended.
`expr`	An expression that generates a mobile
`env`	The environment in which to evaluate `expr`.
`quoted`	Is `expr` a quoted expression (with `quote()`)? This is useful if you want to save an expression in a variable.

---

mobile_animate *Animate*

Description

Animate various aspects of the chart.

Usage

```
mobile_animate(m, aspect, ...)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`aspect`	Aspect to animate, see section below for valid values.
`...`	Named options from the official documentation.

Aspect

Valid aspect:

- `axis-label`
- `axis-grid`
- `axis-tick`
- `axis-line`
- `line`
- `area`
- `bar`
- `path`
- `point`
- `polygon`
- `schema`

Examples

```
mobile(cars, aes(speed, dist)) %>%
 mobile_point() %>%
 mobile_animate(
   "point",
   appear = list(
     animation = 'fadeIn',
     duration = 5000
   )
 )
```

`mobile_coord` *Coordinates*

Description

Change chart coordinates.

Usage

```
mobile_coord(m, type = c("rect", "polar"), ...)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`type`	Type of coordinate, `rect` for cartesian or `polar`.
`...`	Named options from the official documentation.

Examples

```
mobile(df, aes(label, value, color = label)) %>%
 mobile_bar() %>%
 mobile_coord("polar", transposed = TRUE, innerRadius = .7) %>%
 mobile_hide_axis()
```

`mobile_interaction` *Add Interactions*

Description

Add interactions on pan, pinch, bar selection or pie selection.

Usage

```
mobile_interaction(m, interaction, ...)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`interaction`	Name of the interaction.
`...`	Named options from the official documentation.

Interactions

- `bar-select`
- `pie-select`
- `pinch`
- `pan`

Examples

```
df <- data.frame(
 x = letters[1:5],
 y = runif(5)
)

mobile(df, aes(x, y)) %>%
 mobile_bar() %>%
 mobile_interaction("bar-select")

df <- data.frame(
 x = 1:20,
 y = runif(20, 1, 10)
)
```

```
mobile(df, aes(x, y)) %>%
 mobile_bar() %>%
 mobile_interaction("pan", limitRange = list(x = list(min = -20, max = 40)))
```

`mobile_legend` *Legend*

Description

Customise the legend of the chart.

Usage

```
mobile_legend(m, ...)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`...`	Named options from the official documentation.

Examples

```
mobile(cars) %>%
 mobile_point(aes(speed, dist)) %>%
 mobile_legend(position = "right")
```

`mobile_options` *General Options*

Description

Pass general options to the chart.

Usage

```
mobile_options(m, padding = "auto", animate = TRUE)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`padding`	An integer or a list containing 4 integers e.g.: `list(40, 10, 'auto', 'auto')` giving padding on top, right, bottom, and left.
`animate`	Set to `FALSE` to disable animations.

mobile_scroll *Scroll*

Description

Add a scrollbar to let user pan charts.

Usage

```
mobile_scroll(m, mode = c("x", "y", "xy"), ...)
```

Arguments

m	An object of class mobile as returned by mobile.
mode	Axis to apply scroll to.
...	Named options from the official documentation.

Examples

```
# range for pan
lmt <- list(x = list(min = -20, max = 20))

mobile(df, aes(x, y)) %>%
 mobile_bar() %>%
 mobile_interaction("pan", limitRange = lmt) %>%
 mobile_scroll(mode = "x", xStyle = list(offsetY = -5))
```

mobile_tooltip *Tooltip*

Description

Add a tooltip. Note that the visualisations being optimized for mobile there is no such thing as hover: tooltips show on click.

Usage

```
mobile_tooltip(m, ...)
```

Arguments

m	An object of class mobile as returned by mobile.
...	Named options from the official documentation.

Examples

```
mobile(cars) %>%
 mobile_point(aes(speed, dist)) %>%
 mobile_tooltip(snap = TRUE)
```

scale *Scale*

Description

Scale variables.

Usage

```
mobile_scale_x(m, ...)

mobile_scale_y(m, ...)
```

Arguments

`m`	An object of class `mobile` as returned by `mobile`.
`...`	Named options from the official documentation.

Examples

```
mobile(cars, aes(speed, dist)) %>%
 mobile_point() %>%
 mobile_scale_x(min = -10, tickCount = 20)
```

theme *Theme*

Description

Set a theme

Usage

```
mobile_theme(m, colors, ..., background_color = NULL)

mobile_ggplot_theme(m, n = 3, ...)
```

Arguments

`m` An object of class `mobile` as returned by `mobile`.

`colors` A vector of colors to use as palette.

`...` Any other options.

`background_color` Background color of plot, if `NULL` will be transparent.

`n` Number of colors to generate.

Index