07 Erstellung von Grafiken mit ggplot2

Martin Hanewald

2019-02-19

Packages

```
library(tidyverse)
library(knitr)
library(DT)
```

Überblick

Das Paket `ggplot2` ist die meistgenutze Grafikbibliothek in R. Sein modularer Aufbau in `aesthetics`, `coordinates` und `geometries` erlaubt beliebige Freiheit in der Gestaltung von Plots.

Dataset

Datensatz `midwest` aus dem Package `ggplot2` enthält Daten einer Volkszählung.

```
data(midwest)
# Umwandlung einiger Variablen in Datentyp 'factor'
midwest <- midwest %>% mutate_at(vars(county, state, inmetro, category), as.factor)
# Show sample
sample_n(midwest, 10) %>% DT::datatable(width = 700, options=list(scrollX = TRUE))
```

Show 10 ▾ entries Search:

	PID	county	state	area	poptotal	popdensity	popwhite	popbla
1	691	HAMILTON	IN	0.024	108936	4539	106764	
2	705	KOSCIUSKO	IN	0.032	65294	2040.4375	64058	
3	2014	AUGLAIZE	OH	0.024	44585	1857.70833	44225	
4	1278	WAYNE	MI	0.035	2111687	60333.9143	1212007	849
5	742	TIPTON	IN	0.016	16119	1007.4375	15990	
6	1236	KALKASKA	MI	0.033	13497	409	13321	
7	3003	GREEN	WI	0.034	30339	892.323529	30173	
8	670	CARROLL	IN	0.022	18809	854.954545	18720	
9	2995	DOOR	WI	0.028	25690	917.5	25387	
10	2095	WOOD	OH	0.037	113269	3061.32432	109303	1

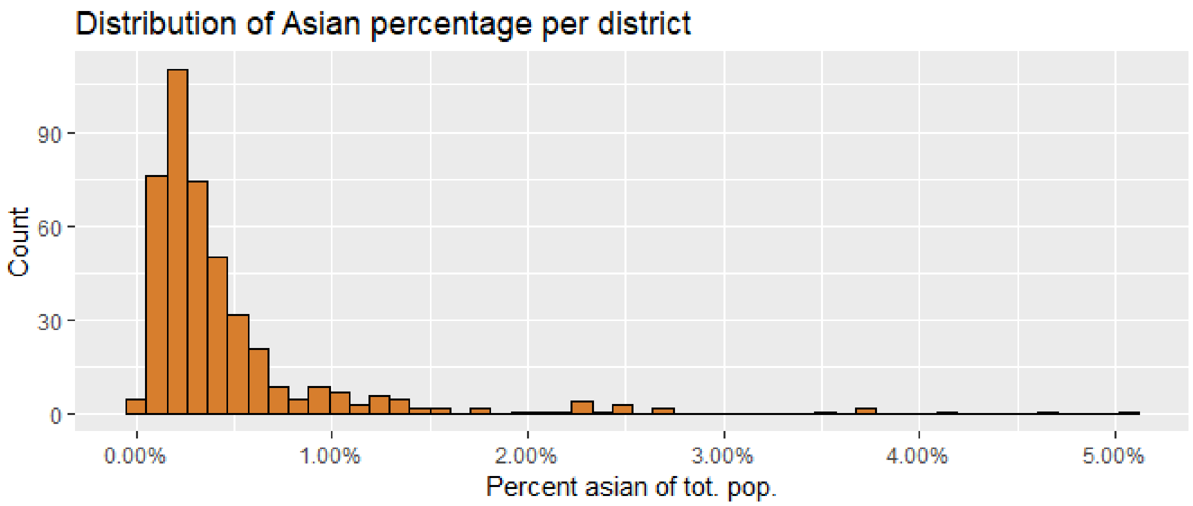
Showing 1 to 10 of 10 entries

Previous 1 Next

Histogram

Verteilung einer numerischen Variable

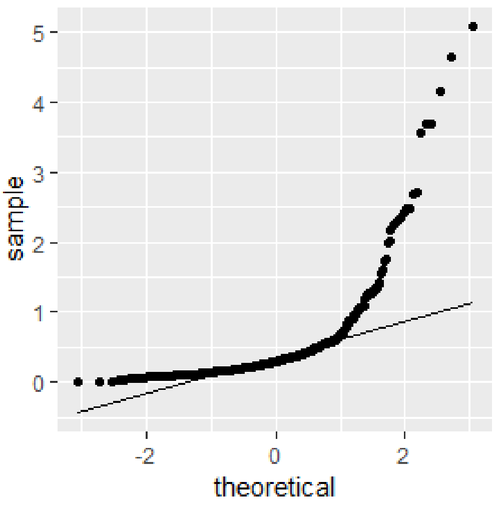
```
midwest %>%
    ggplot(aes(x = percasian / 100)) +
    geom_histogram(bins = 50, color=1, fill='#d77e2d') +
    scale_x_continuous(labels=scales::percent) +
    labs(x='Percent asian of tot. pop.', y = 'Count',
        title='Distribution of Asian percentage per district')
```

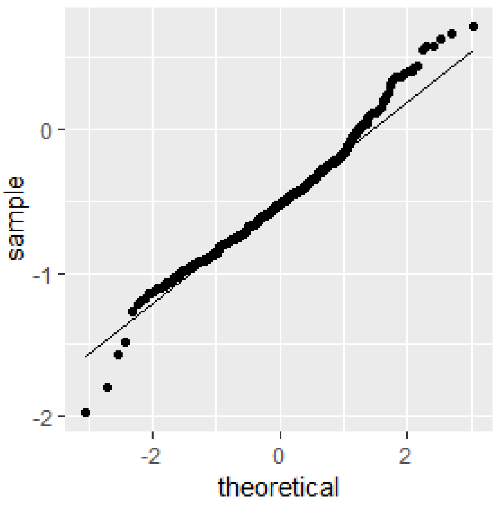


Quantilsplot

Vergleich einer Verteilung mit Normalverteilung

```
midwest %>%
    ggplot(aes(sample=percasian)) + geom_qq() + stat_qq_line()

midwest %>%
    ggplot(aes(sample=log10(percasian))) + geom_qq() + stat_qq_line()
```

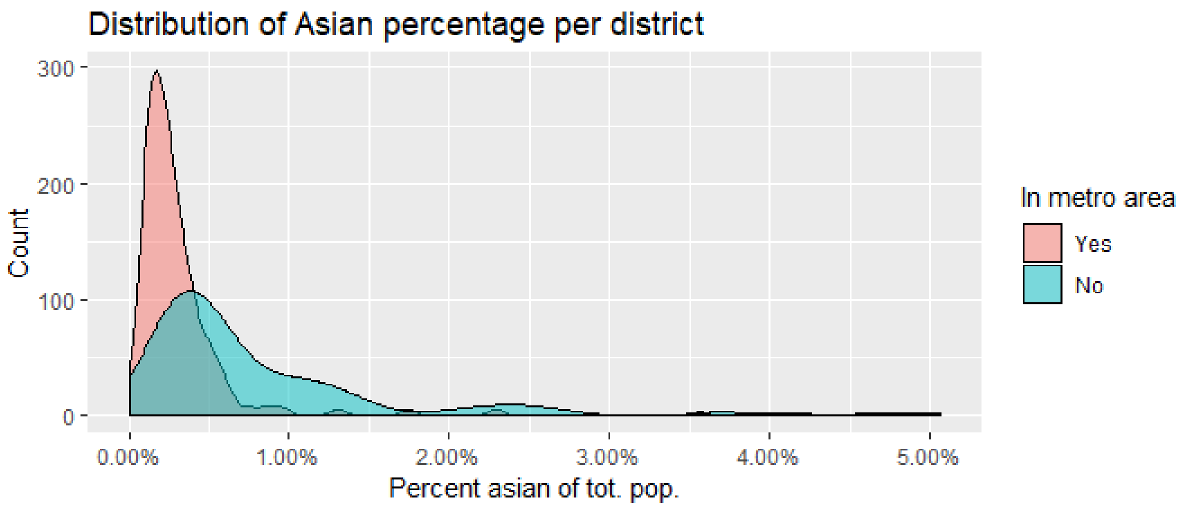




Density plot

Verteilung einer numerischen Variable im Vergleich mit wenigen Kategorien

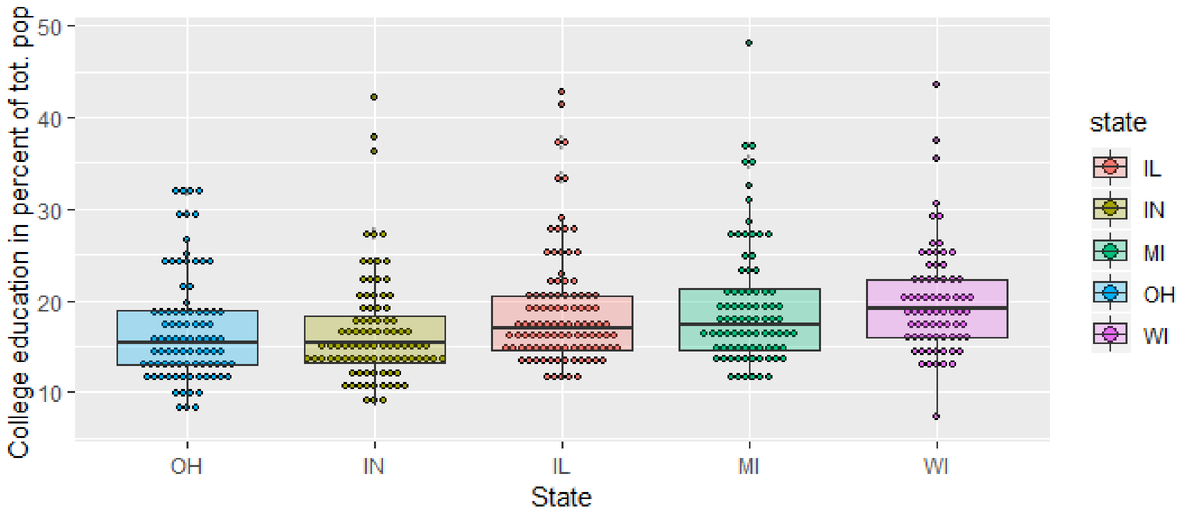
```
midwest %>%
    ggplot(aes(x = percasian / 100, fill = inmetro)) +
    geom_density(alpha=.5) +
    scale_x_continuous(labels = scales::percent) +
    scale_fill_discrete(labels = c('Yes', 'No')) +
    labs(x='Percent asian of tot. pop.', y = 'Count',
        title='Distribution of Asian percentage per district',
        fill = 'In metro area')
```



Boxplot / Dotplot

Verteilung einer numerischen Variable über mehrere Kategorien

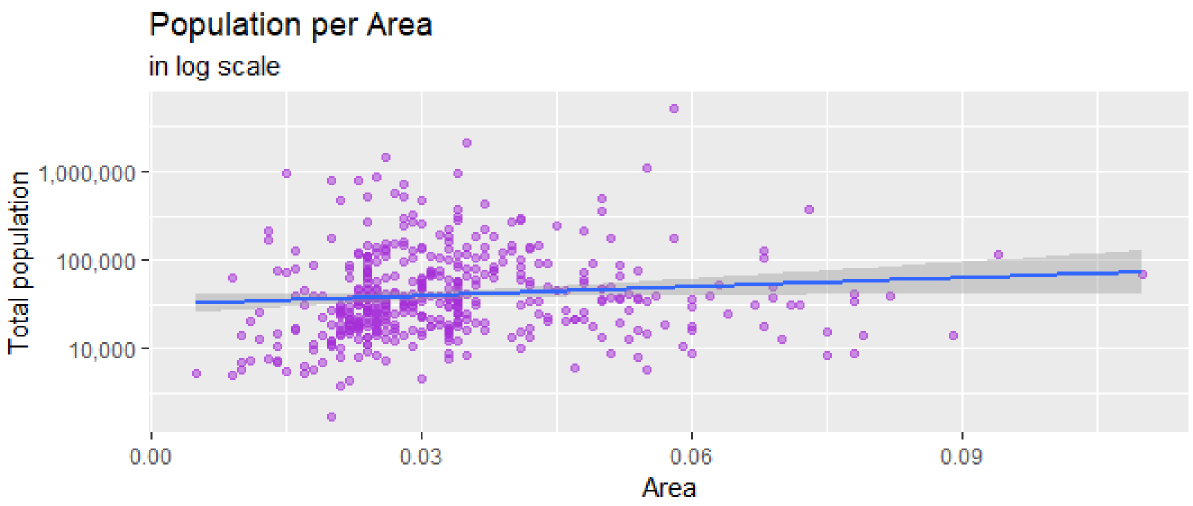
```
midwest %>%
    ggplot(aes(x = state %>% fct_reorder(percollege), y = percollege, fill=state)) +
    geom_dotplot(binaxis = 'y', stackdir = 'center', dotsize=.6) +
    geom_boxplot(alpha = .3, outlier.size = 0) +
    labs(x = 'State', y = 'College education in percent of tot. pop.')
```



Scatterplot

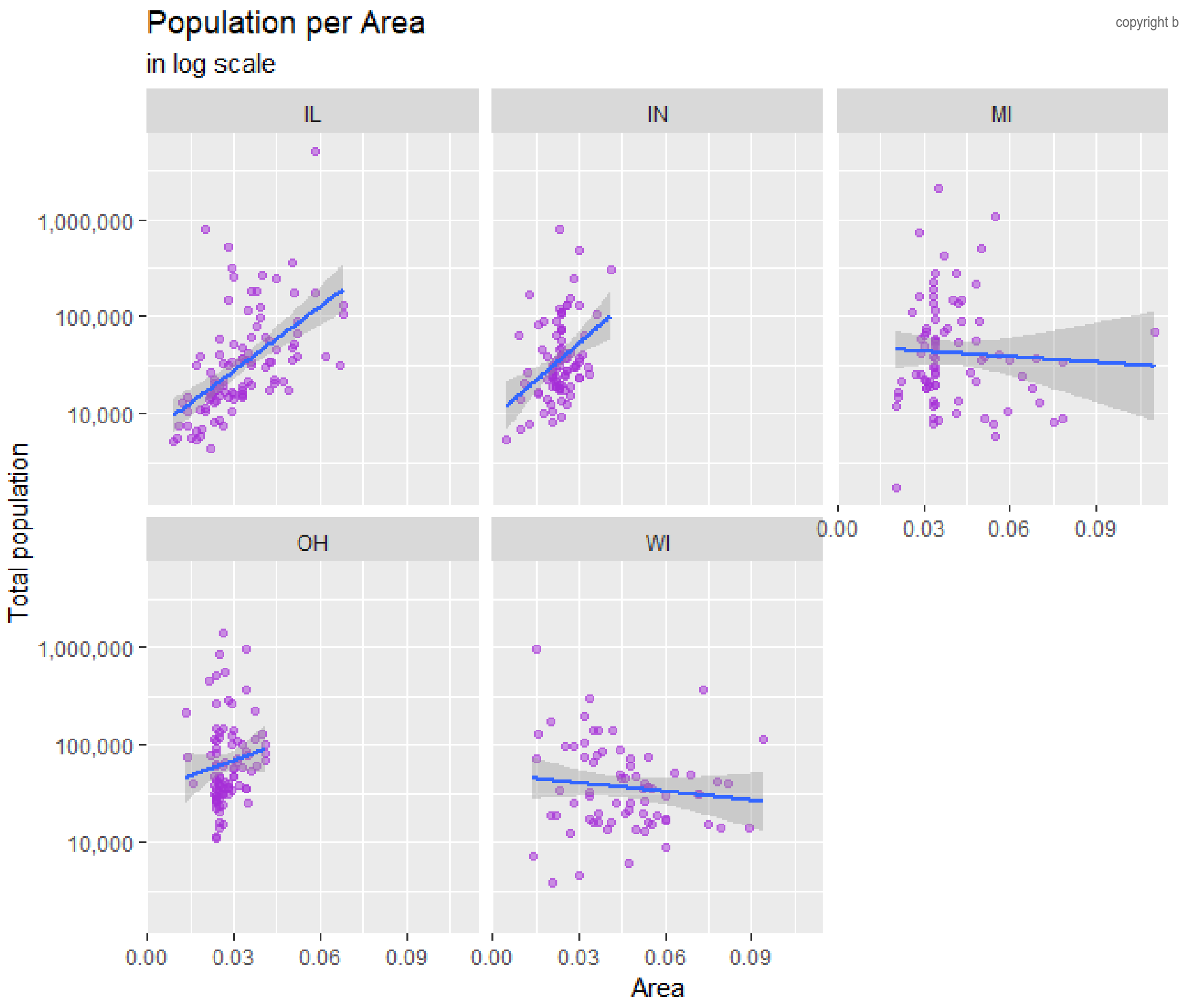
Relation zwischen zwei numerischen Variablen

```
midwest %>%
    ggplot(aes(x=area, y=poptotal)) +
    geom_point(alpha=.5, color='#a52dd7') +
    geom_smooth(method="lm") +
    scale_y_log10(labels= scales::comma) +
    labs(x = 'Area', y = 'Total population',
         title='Population per Area',
         subtitle = 'in log scale')
```



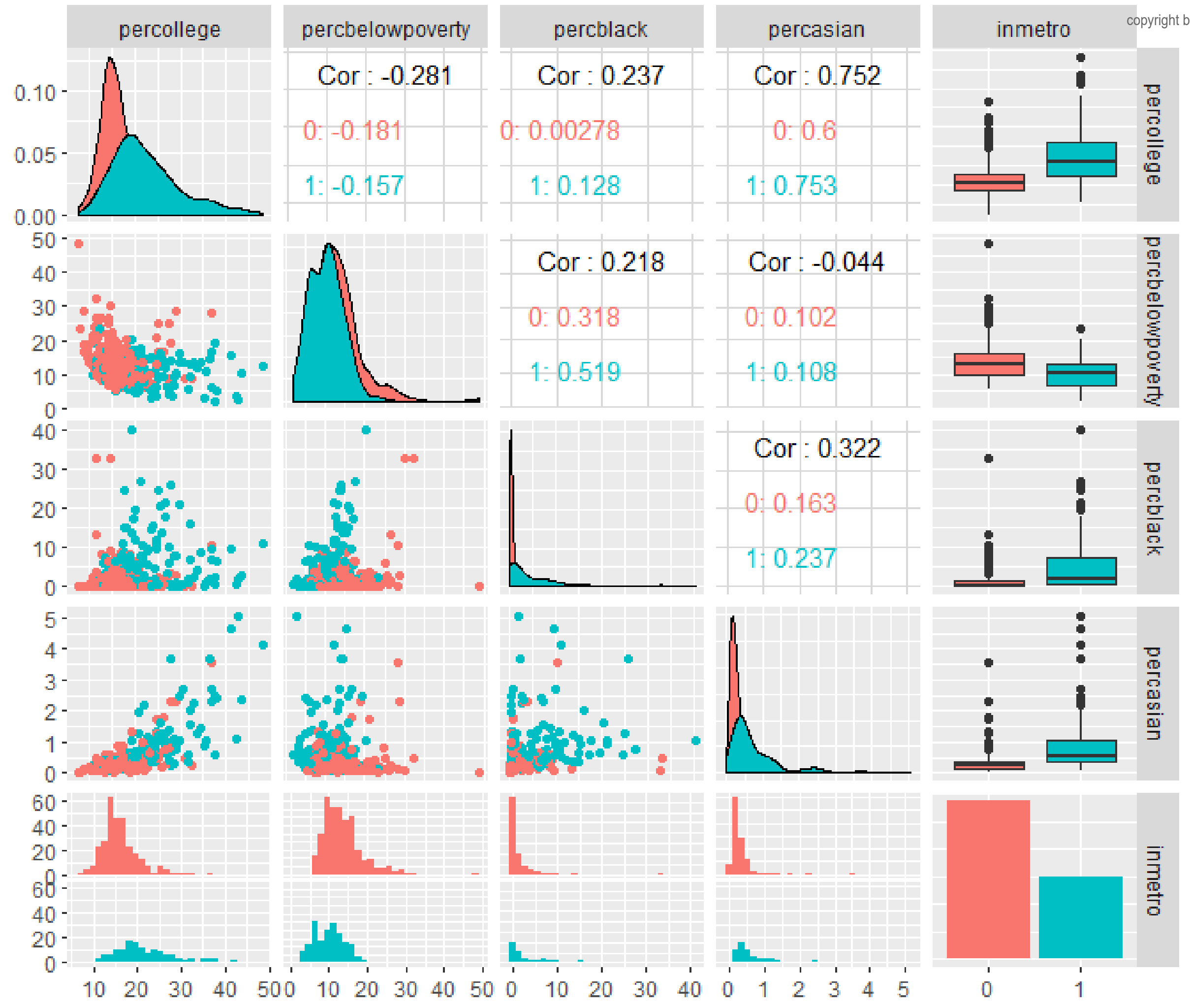
Als *facet-plot* unterschieden nach *state*

```
midwest %>%
    ggplot(aes(x=area, y=poptotal)) +
    geom_point(alpha=.5, color='#a52dd7') +
    geom_smooth(method="lm") +
    scale_y_log10(labels= scales::comma) +
    labs(x = 'Area', y = 'Total population',
         title='Population per Area',
         subtitle = 'in log scale') +
    facet_wrap(vars(state))
```



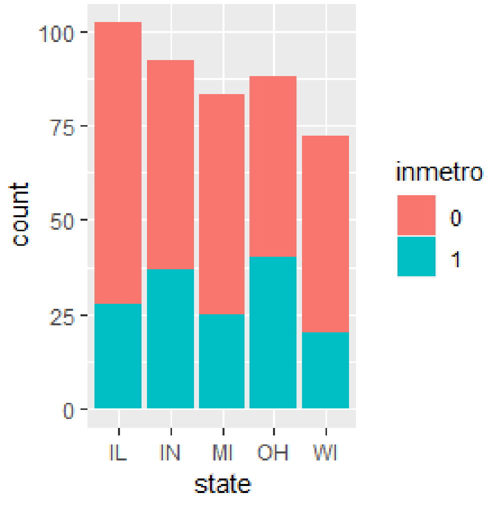
Matrix-Scatterplot

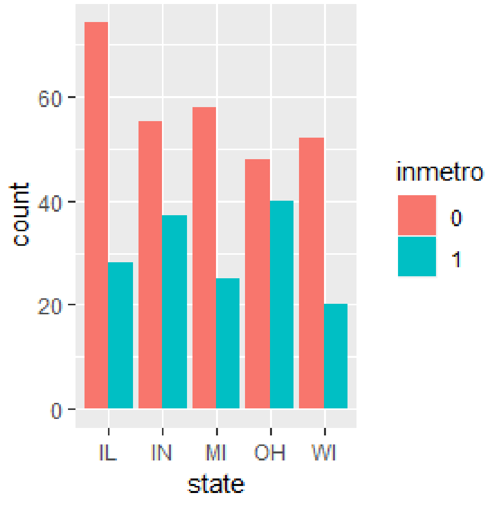
```
library(GGally)

midwest %>%
    select(percollege, percbelowpoverty, percblack, percasian, inmetro) %>%
    GGally::ggpairs(mapping=aes(color=inmetro))
```



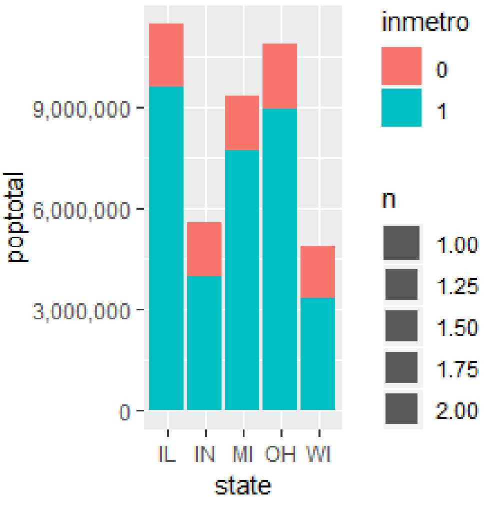
Barplots

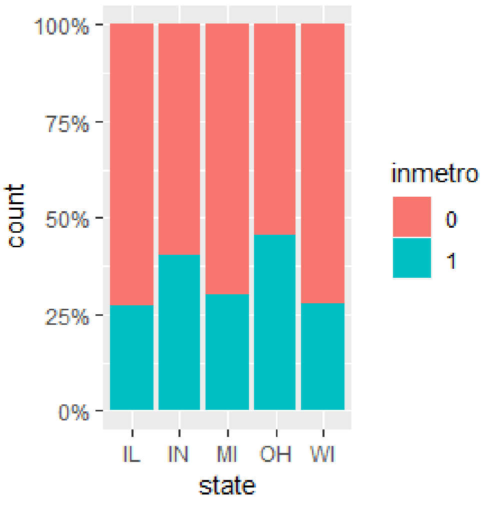
```
midwest %>%
    ggplot(aes(state, fill = inmetro)) + geom_bar()

midwest %>%
    ggplot(aes(state, fill = inmetro)) + geom_bar(position='dodge')
```





```
midwest %>%
    ggplot(aes(state, fill = inmetro, y = poptotal)) + geom_bar(stat='sum') +
    scale_y_continuous(labels=scales::comma)

midwest %>%
    ggplot(aes(state, fill = inmetro)) + geom_bar(position='fill') +
    scale_y_continuous(labels = scales::percent)
```

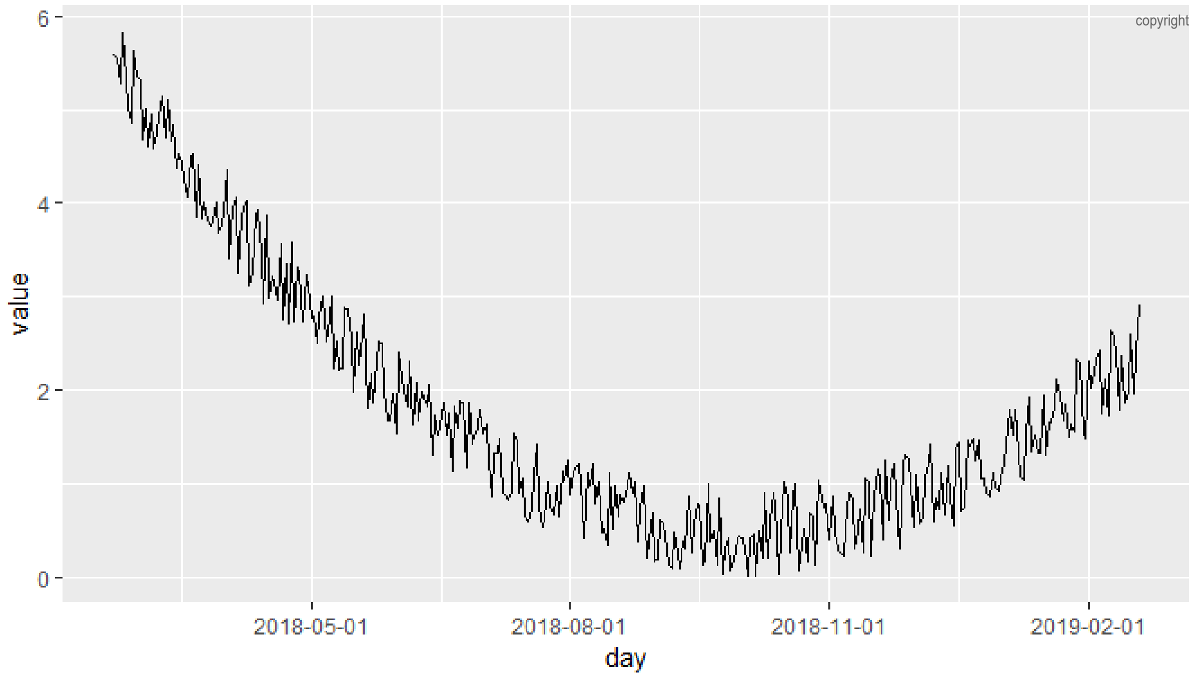




Timeseries

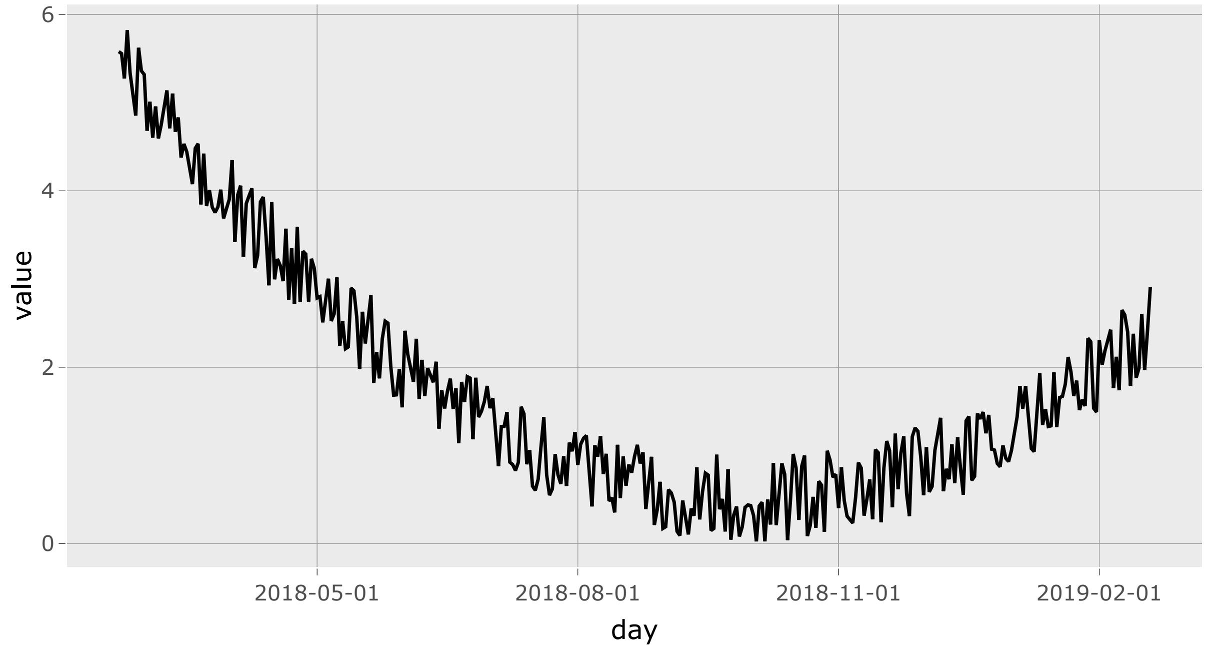
```
# Build a Time series data set
day <- Sys.Date() - 0:364
value <- runif(365) + seq(-140, 224)^2 / 10000
tsdata <- tibble(day, value)
```

```
p <- tsdata %>%
    ggplot(aes(day, value)) +
    geom_line() +
    scale_x_date(
        #date_labels  = '%Y-%m',
        date_breaks = '3 months'
        )
p
```



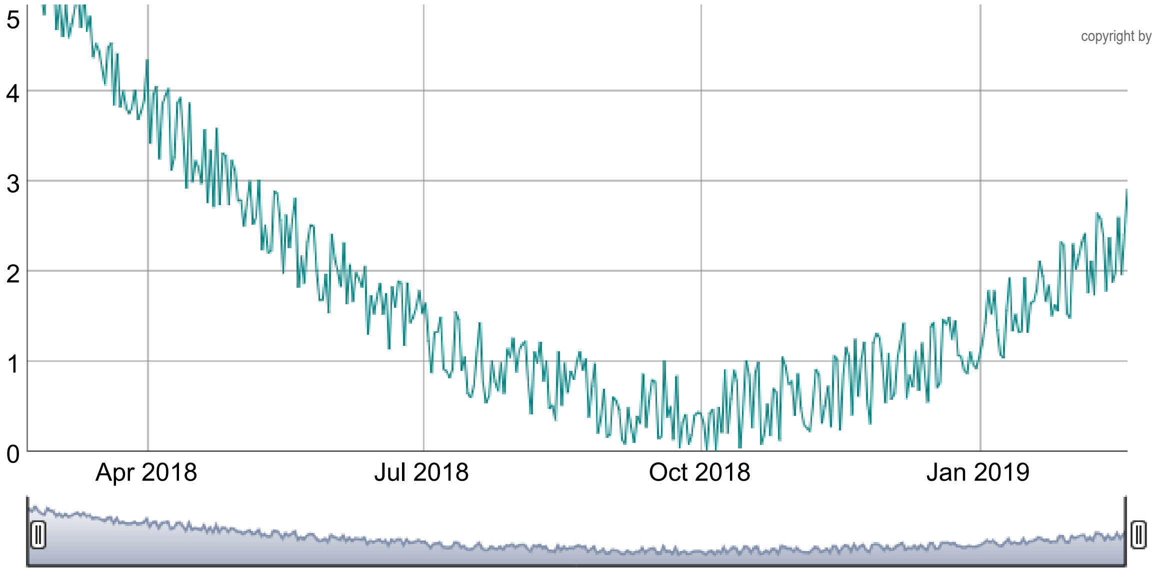
Interaktivität mit `ggplotly`

```r
library(plotly)
ggplotly(p)
```



Interaktivität mit `dygraphs`

```r
library(dygraphs)

xtsdata <- tsdata %>%
    as.data.frame() %>%
    column_to_rownames("day") %>%
    xts::as.xts()

xtsdata %>% dygraph() %>% dyRangeSelector()
```



Weitere Beispiele

https://www.r-graph-gallery.com/