Untitled

2023-10-17

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(gridExtra)
```

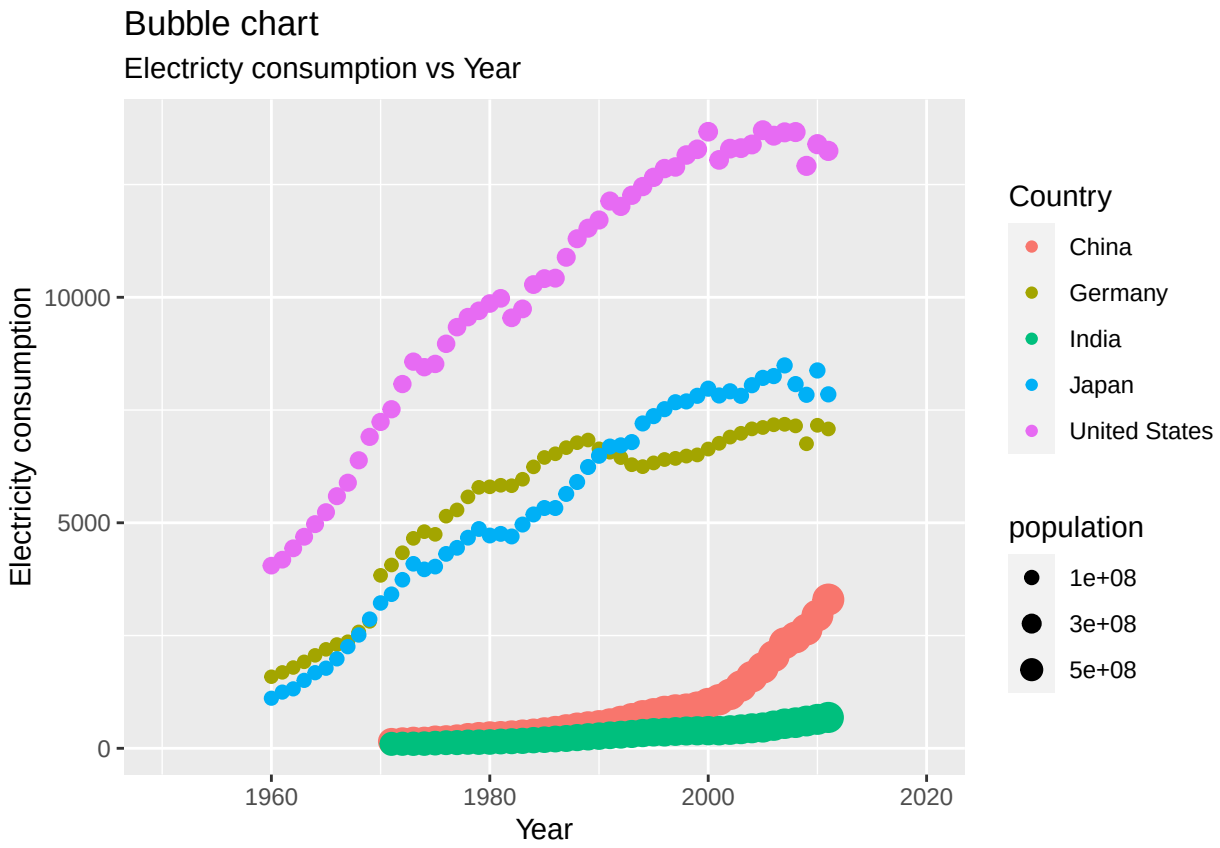
```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
df <- read_csv("../data/gapminder-data.csv")
```

```
## New names:
## Rows: 1512 Columns: 10
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): Country dbl (9): ...1, Year, gdp_per_capita,
## Electricity_consumption_per_capita, und...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
dfs <- subset(df, Country %in% c("Germany", "India", "China", "United States", "Japan"))
ggplot(dfs, aes(x=Year, y=Electricity_consumption_per_capita)) +
  geom_point(aes(size=population, color=Country)) +
  coord_cartesian(xlim=c(1950, 2020)) +
  labs(subtitle="Electricty consumption vs Year", title="Bubble chart") +
  ylab("Electricity consumption") +
  scale_size(breaks=c(0, 1e+8, 0.3e+9, 0.5e+9, 1.5e+9), range=c(1, 5))
```

```
## Warning: Removed 842 rows containing missing values (`geom_point()`).
```




```
df <- read_csv("../data/LoanStats.csv")
```

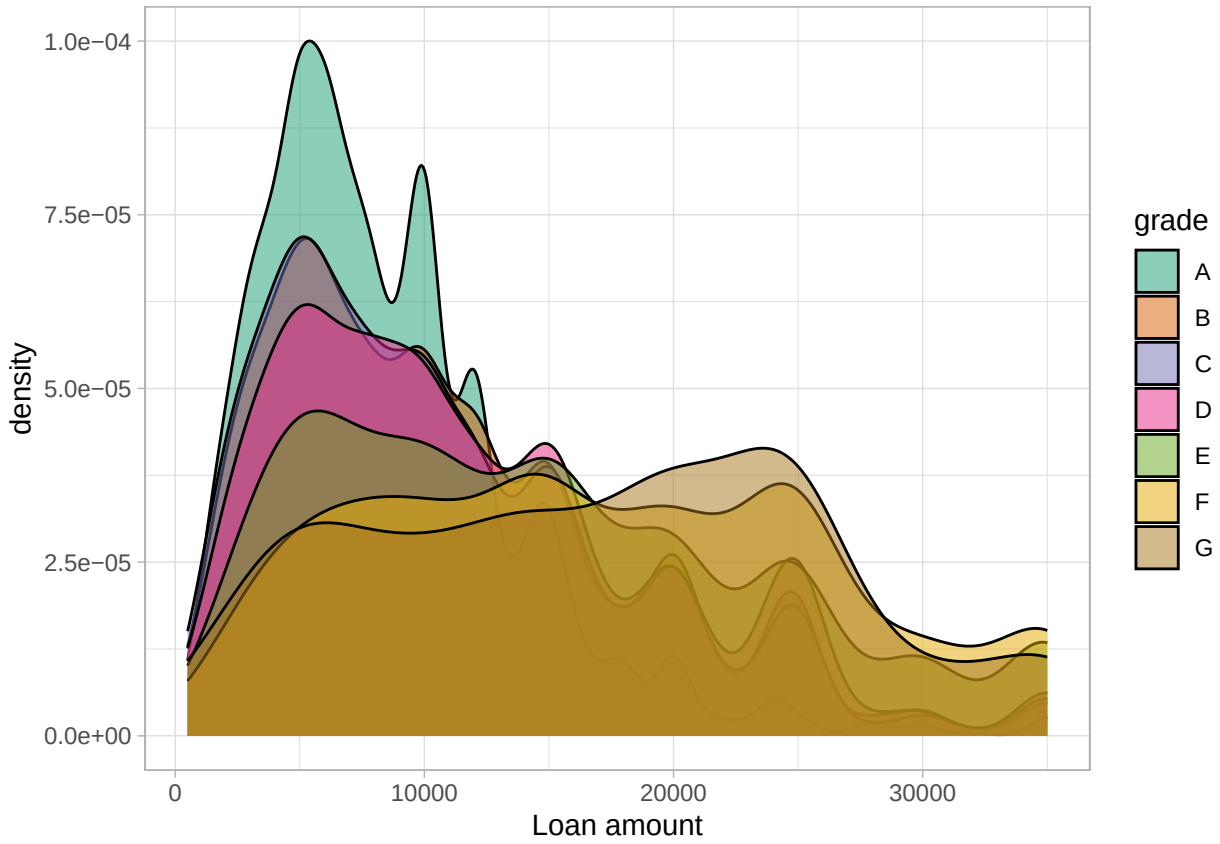
```
## Rows: 42542 Columns: 145
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (29): id, term, int_rate, grade, sub_grade, emp_title, emp_length, home_...
## dbl (34): loan_amnt, funded_amnt, funded_amnt_inv, installment, annual_inc, ...
## lgl (82): member_id, url, initial_list_status, mths_since_last_major_derog, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df <- df %>% drop_na(grade)
ggplot(df, aes(x=loan_amnt)) +
  geom_density() +
  facet_wrap(~grade) +
  xlab("Loan amount")
```


```
df <- read_csv("../data/LoanStats.csv")
```

```
## Rows: 42542 Columns: 145
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (29): id, term, int_rate, grade, sub_grade, emp_title, emp_length, home_...
## dbl (34): loan_amnt, funded_amnt, funded_amnt_inv, installment, annual_inc, ...
## lgl (82): member_id, url, initial_list_status, mths_since_last_major_derog, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

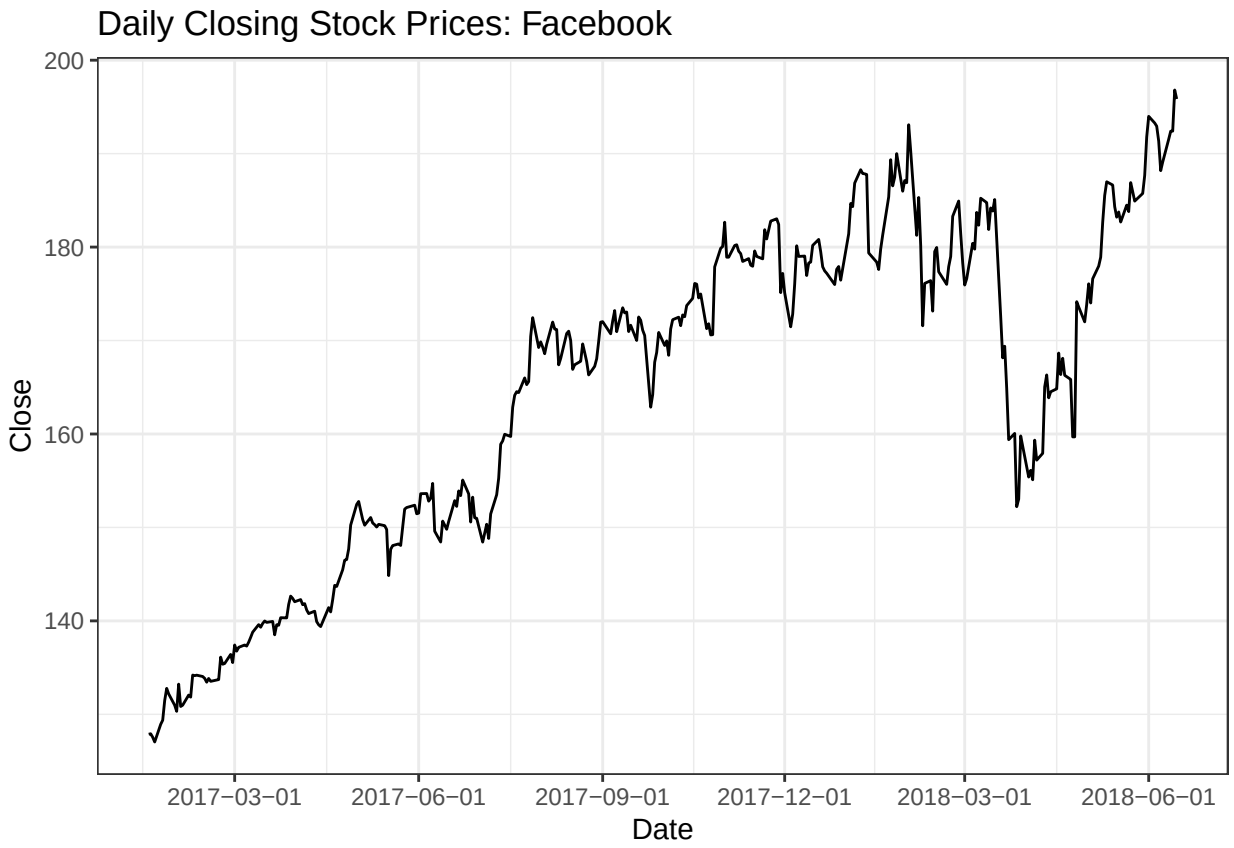
```
df <- df %>% drop_na(grade)
ggplot(df, aes(x=loan_amnt)) +
  geom_density(aes(fill=grade), alpha=1/2) +
  scale_fill_brewer(palette="Dark2") +
  xlab("Loan amount") +
  theme_light()
```




```
df_fb <- read_csv("../data/FB.csv")
```

```
## Rows: 357 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (6): Open, High, Low, Close, Adj Close, Volume
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_fb$Date <- as.Date(df_fb$Date)
ggplot(df_fb, aes(x=Date, y=Close, group=1)) +
  geom_line(color="black", na.rm=TRUE) +
  ggtitle("Daily Closing Stock Prices: Facebook") +
  theme(plot.title=element_text(lineheight=0.7, face="bold")) +
  scale_x_date(date_breaks="3 month") +
  theme_bw()
```



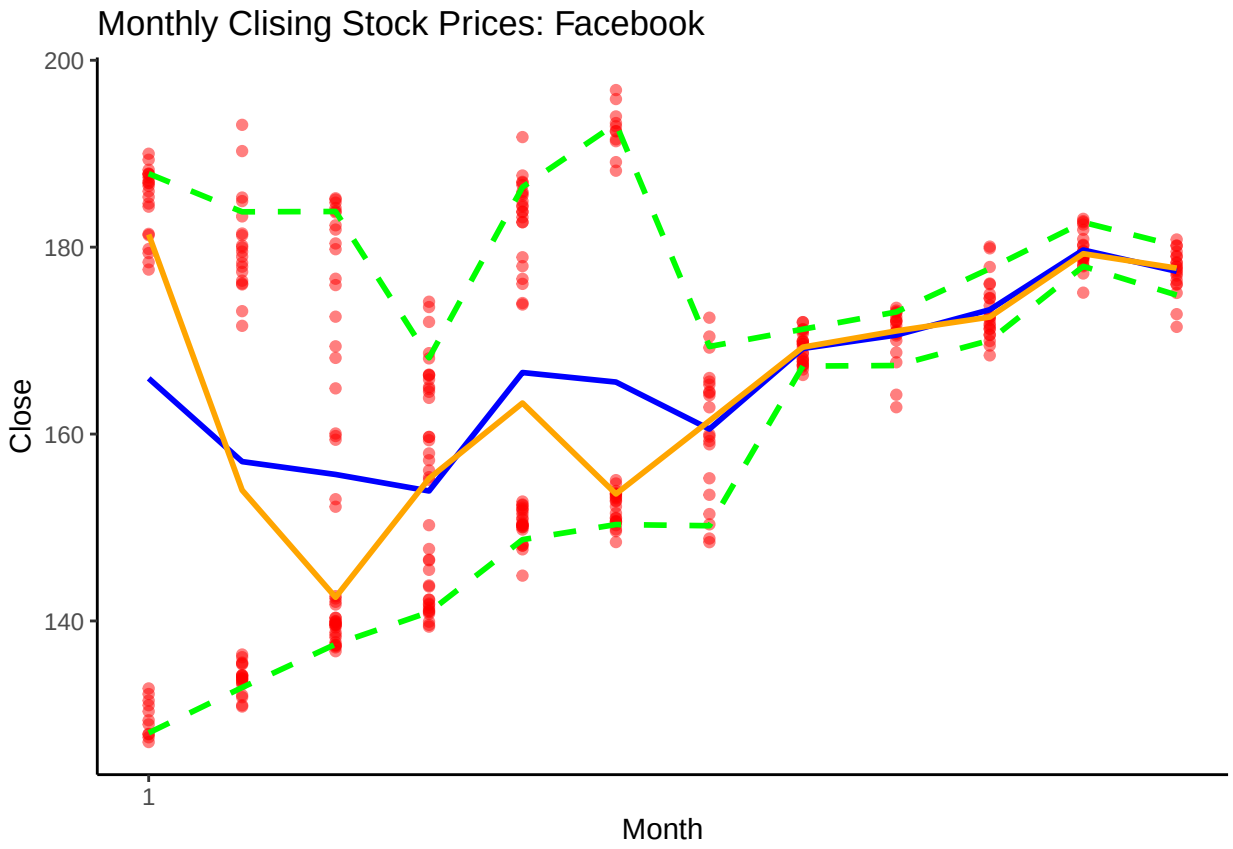

```
df_fb <- read_csv("../data/FB.csv")
```

```
## Rows: 357 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (6): Open, High, Low, Close, Adj Close, Volume
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_fb$Date <- as.Date(df_fb$Date)
df_fb$Month <- strftime(df_fb$Date, "%m")
df_fb$Month <- as.numeric(df_fb$Month)
ggplot(df_fb, aes(Month, Close)) +
  geom_point(color="red", alpha=1/2, position=position_jitter(h=0.0, w=0.0)) +
  stat_summary(geom="line", fun="mean", color="blue", size=1) +
  stat_summary(geom="line", fun="median", color="orange", size=1) +
  stat_summary(geom="line", fun="quantile", fun.args=list(probs=0.1), linetype=2, color="green", size=1
  stat_summary(geom="line", fun="quantile", fun.args=list(probs=0.9), linetype=2,  color="green", size=
  scale_x_continuous(breaks=c(0, 13, 1)) +
  ggtitle("Monthly Clising Stock Prices: Facebook") +
  theme_classic()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

```
## generated.
```




```
df <- read_csv("../data/gapminder-data.csv")
```

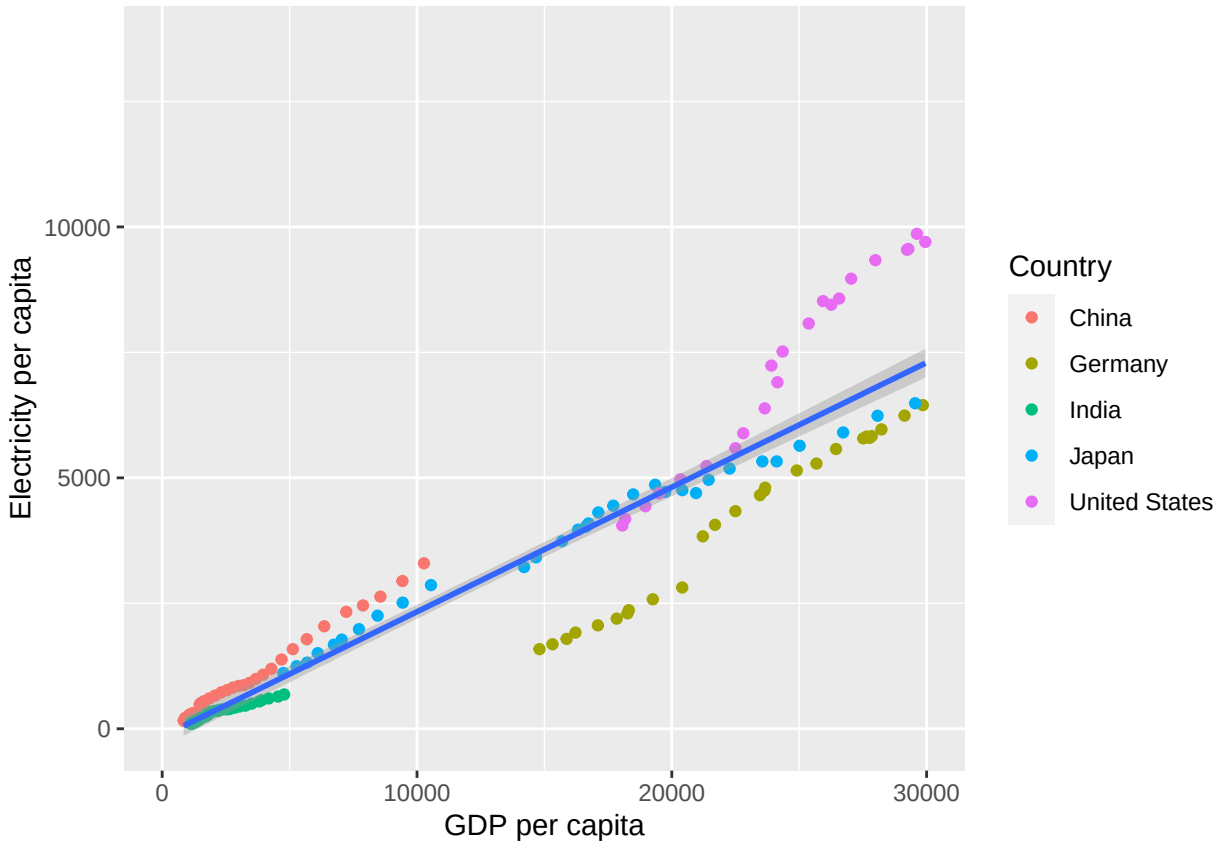
```
## New names:
## Rows: 1512 Columns: 10
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): Country dbl (9): ...1, Year, gdp_per_capita,
## Electricity_consumption_per_capita, und...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
dfs <- subset(df, Country %in% c("Germany", "India", "China", "United States", "Japan"))
ggplot(dfs, aes(gdp_per_capita, Electricity_consumption_per_capita)) +
  geom_point(aes(color=Country)) +
  xlim(0, 30000) +
  xlab("GDP per capita") +
  ylab("Electricity per capita") +
  geom_smooth(method=lm, level=0.95)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 919 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 919 rows containing missing values (`geom_point()`).
```




```
df <- read_csv("../data/gapminder-data.csv")
```

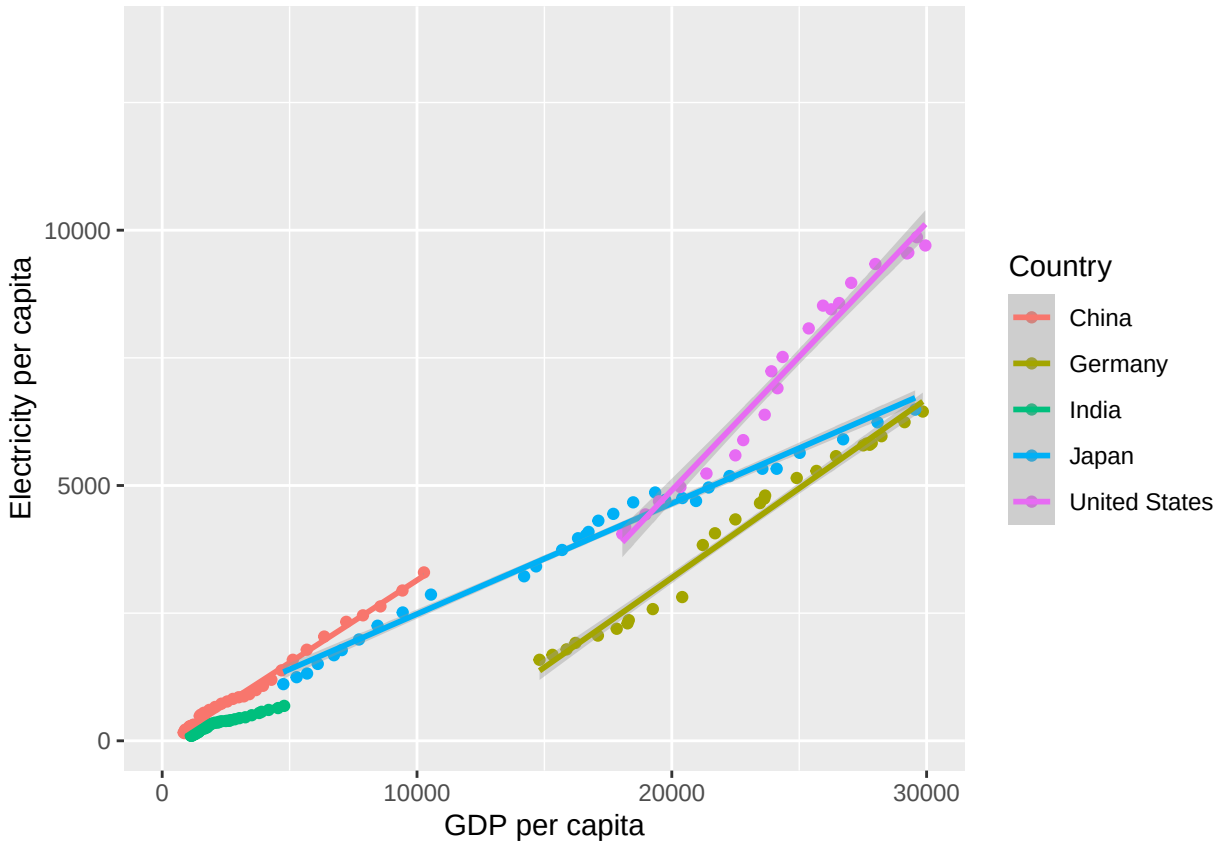
```
## New names:
## Rows: 1512 Columns: 10
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): Country dbl (9): ...1, Year, gdp_per_capita,
## Electricity_consumption_per_capita, und...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
dfs <- subset(df, Country %in% c("Germany", "India", "China", "United States", "Japan"))
ggplot(dfs, aes(gdp_per_capita, Electricity_consumption_per_capita, color=Country)) +
  geom_point() +
  xlim(0, 30000) +
  xlab("GDP per capita") +
  ylab("Electricity per capita") +
  geom_smooth(method=lm, level=0.95)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

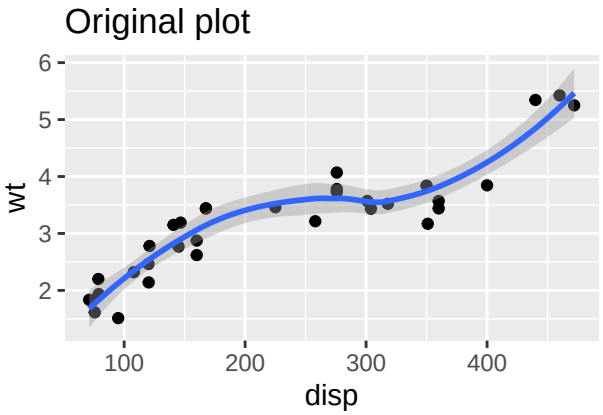
```
## Warning: Removed 919 rows containing non-finite values (`stat_smooth()`).
```

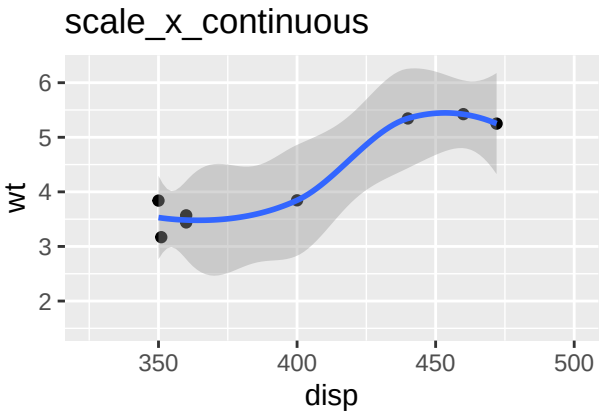
```
## Warning: Removed 919 rows containing missing values (`geom_point()`).
```

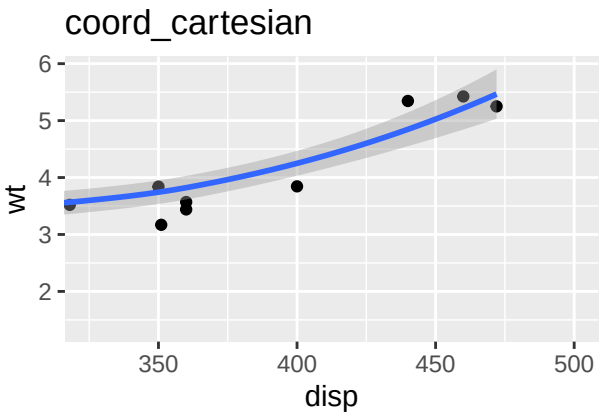



```
p <- ggplot(mtcars, aes(disp, wt)) +
  geom_point() +
  geom_smooth()
p1 <- p + scale_x_continuous(limits=c(325, 500)) +
  ggtitle("scale_x_continuous")
p2 <- p + coord_cartesian(xlim=c(325, 500)) +
  ggtitle("coord_cartesian")
p <- p + ggtitle("Original plot")
grid.arrange(p, p1, p2, ncol=2)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## Warning: Removed 24 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 24 rows containing missing values (`geom_point()`).
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



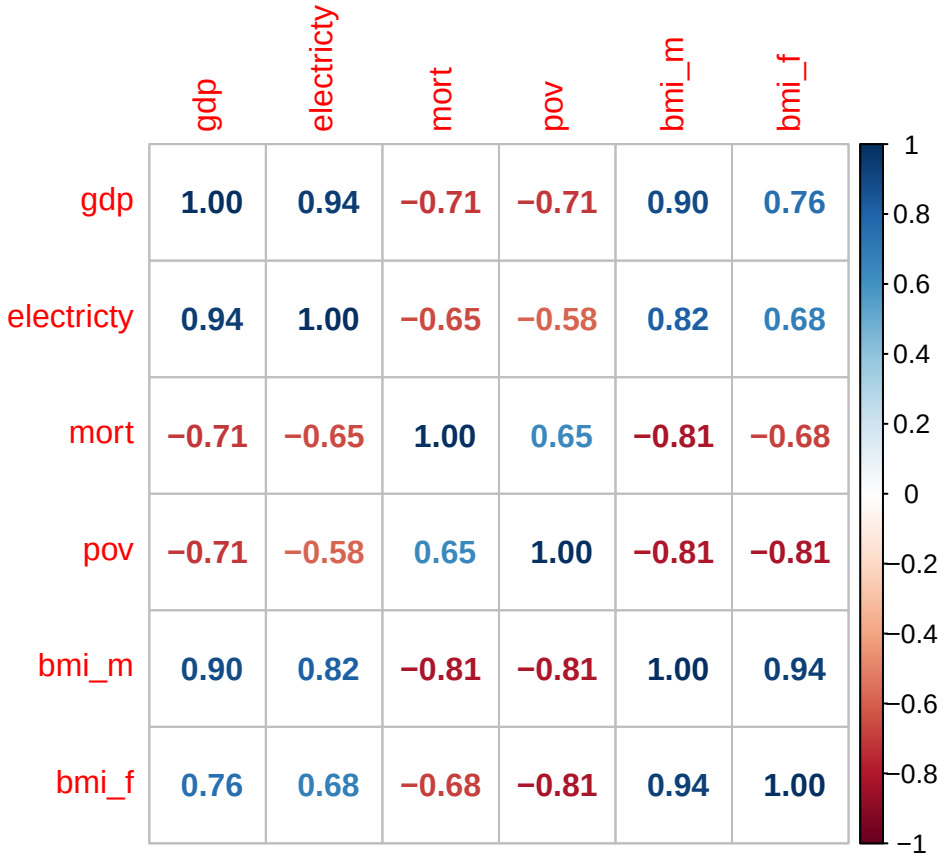




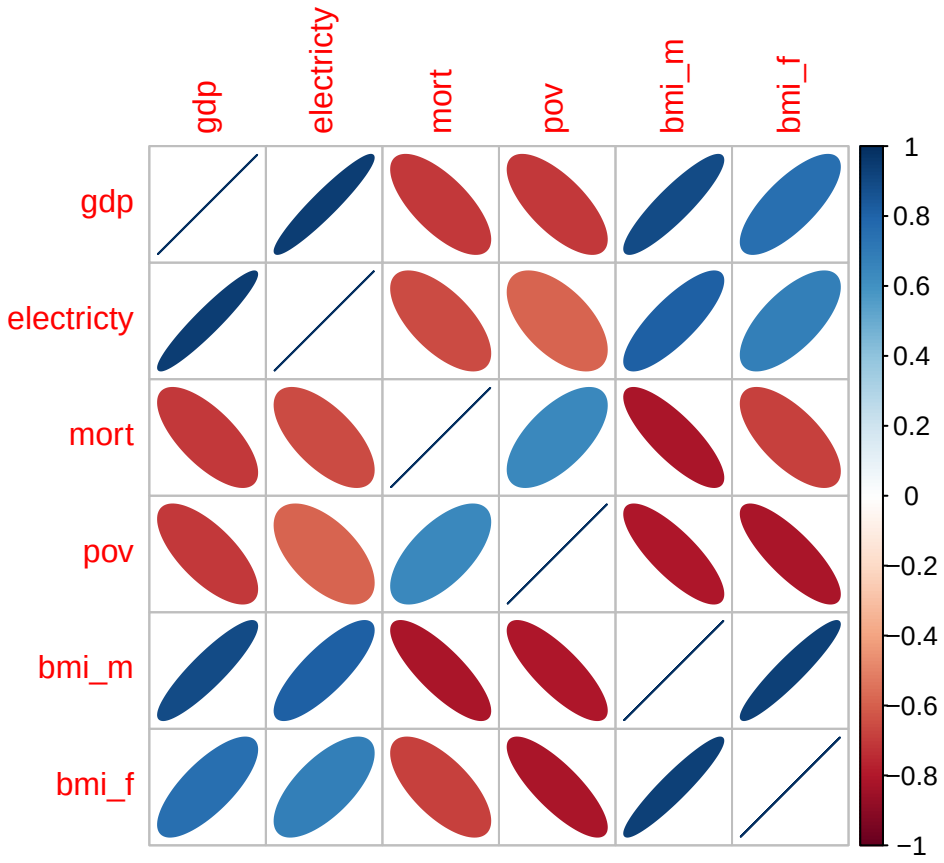

```
df <- read_csv("../data/gapminder-data.csv")
```

```
## New names:
## Rows: 1512 Columns: 10
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): Country dbl (9): ...1, Year, gdp_per_capita,
## Electricity_consumption_per_capita, und...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

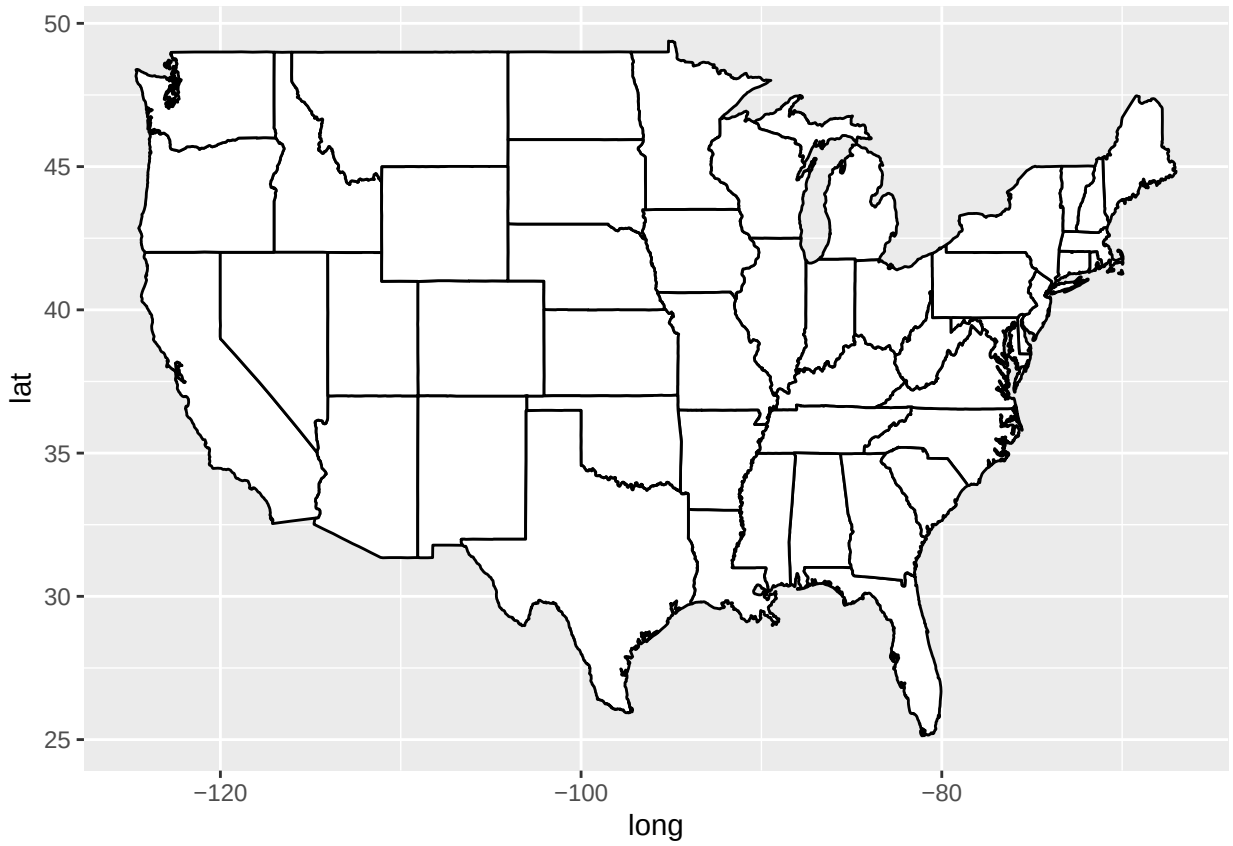
```
df <- df[, colnames(df)[4:9]]
df <- na.omit(df)
colnames(df) <- c("gdp", "electricty", "mort", "pov", "bmi_m", "bmi_f")
M <- cor(df)
corrplot(M, method="number")
```



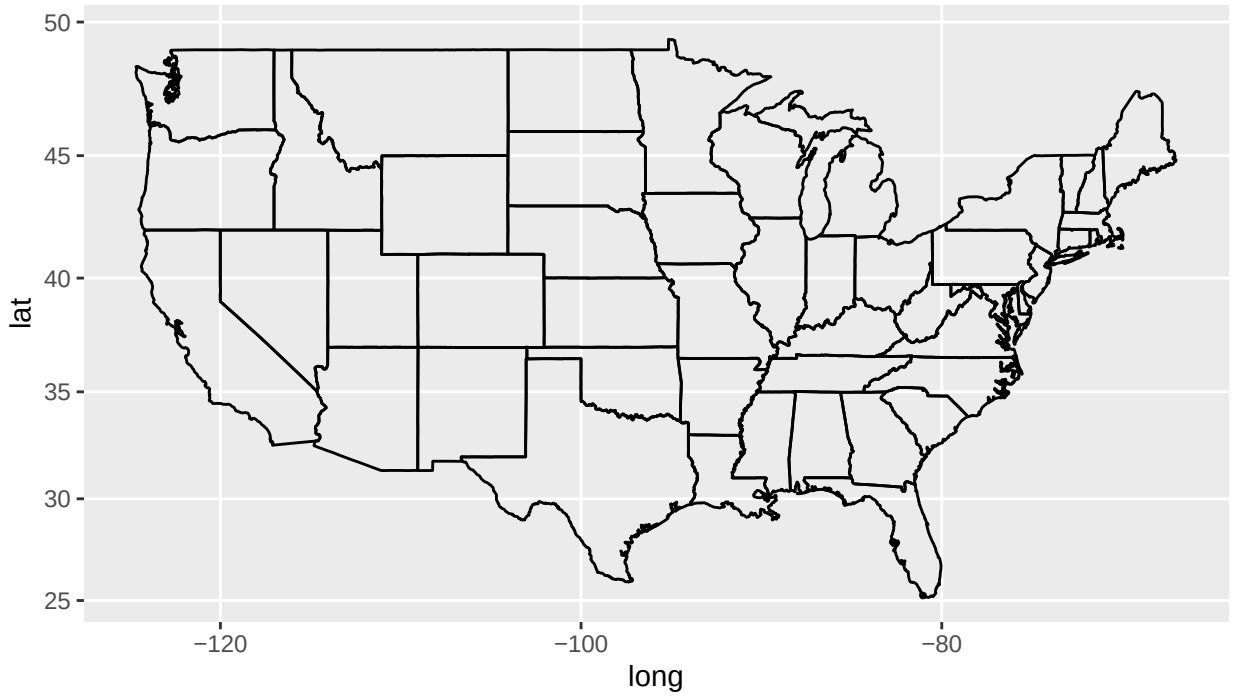

```
corrplot(M, method="ellipse")
```



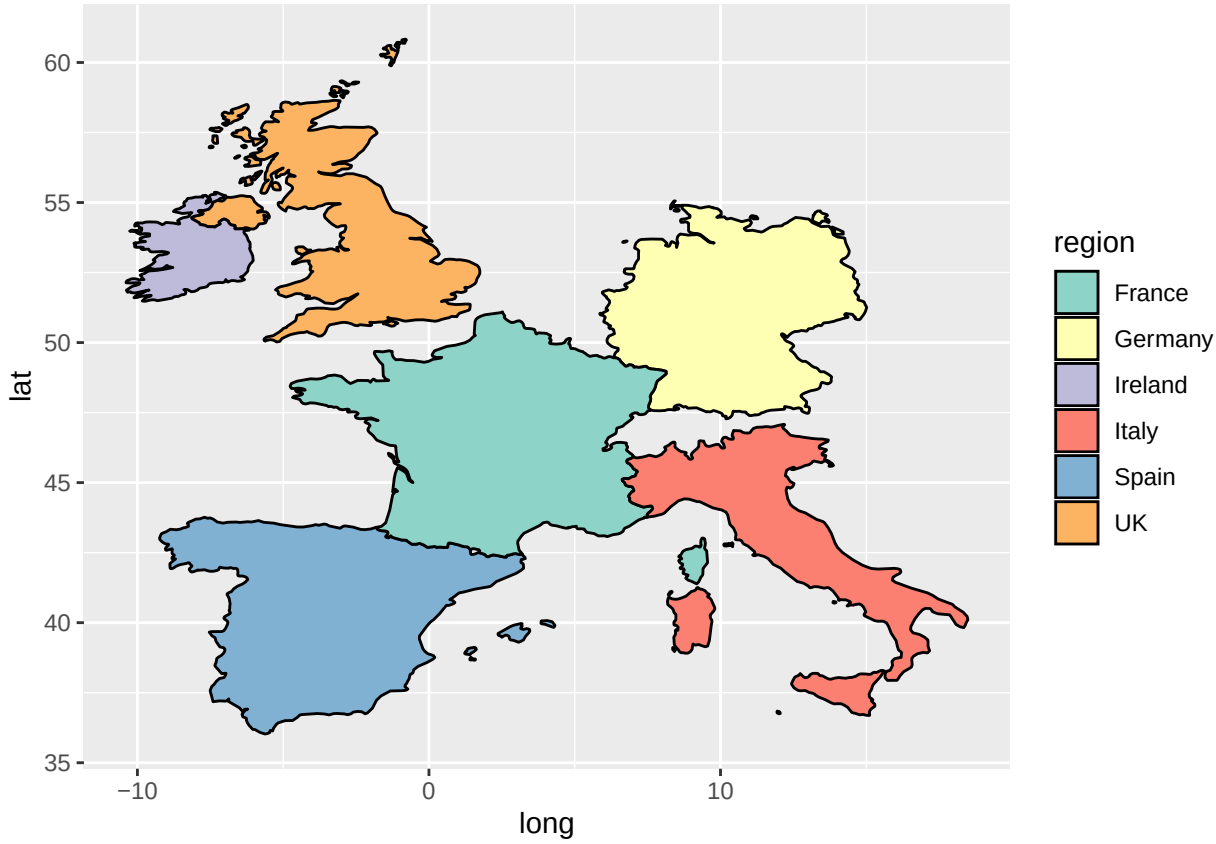

```
states_map <- map_data("state")
ggplot(states_map, aes(x=long, y=lat, group=group)) +
  geom_polygon(fill="white", color="black")
```




```
ggplot(states_map, aes(x=long, y=lat, group=group)) +
  geom_path() +
  coord_map("mercator")
```



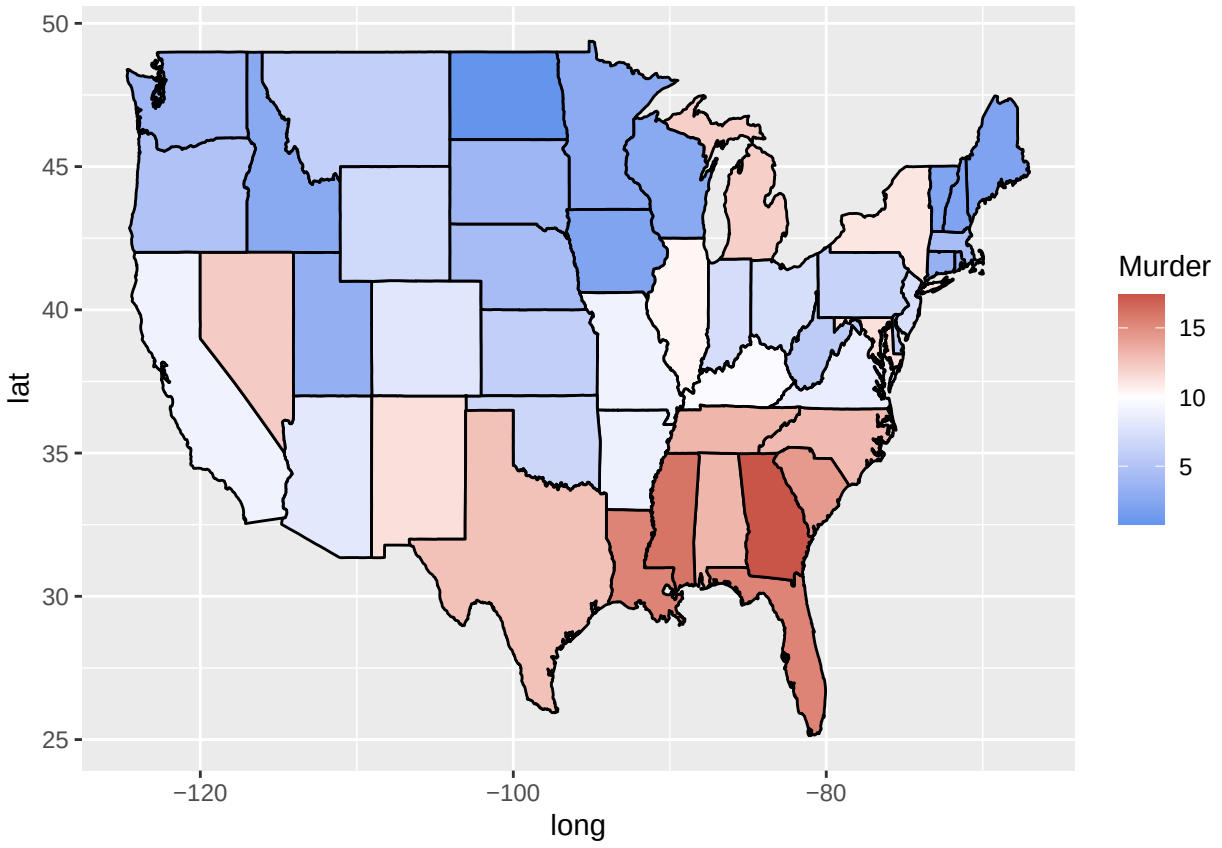
```
europe <- map_data("world", region=c("Germany", "Spain", "Italy", "France", "UK", "Ireland"))
ggplot(europe, aes(x=long, y=lat, group=group, fill=region)) +
  geom_polygon(color="black") +
  scale_fill_brewer(palette="Set3")
```




```
arrests <- as_tibble(USArrests, rownames=NA) %>%
  rownames_to_column("region") %>%
  mutate(region = tolower(region))
states_arrests <- states_map %>% left_join(arrests)
```

```
## Joining with `by = join_by(region)`
```

```
ggplot(states_arrests, aes(x=long, y=lat, group=group)) +
  geom_polygon(aes(fill=Murder), color="black") +
  scale_fill_gradient2(low="cornflowerblue",
                       high="firebrick",
                       midpoint=10)
```


50
45
40
35
30
25
lat
−120
−100
−80
long
Murder
15
10
5

“‘