Ggplot2 tutorial - Command lines

Ariane Ducellier

7/18/2023

```
library(corrplot)
```

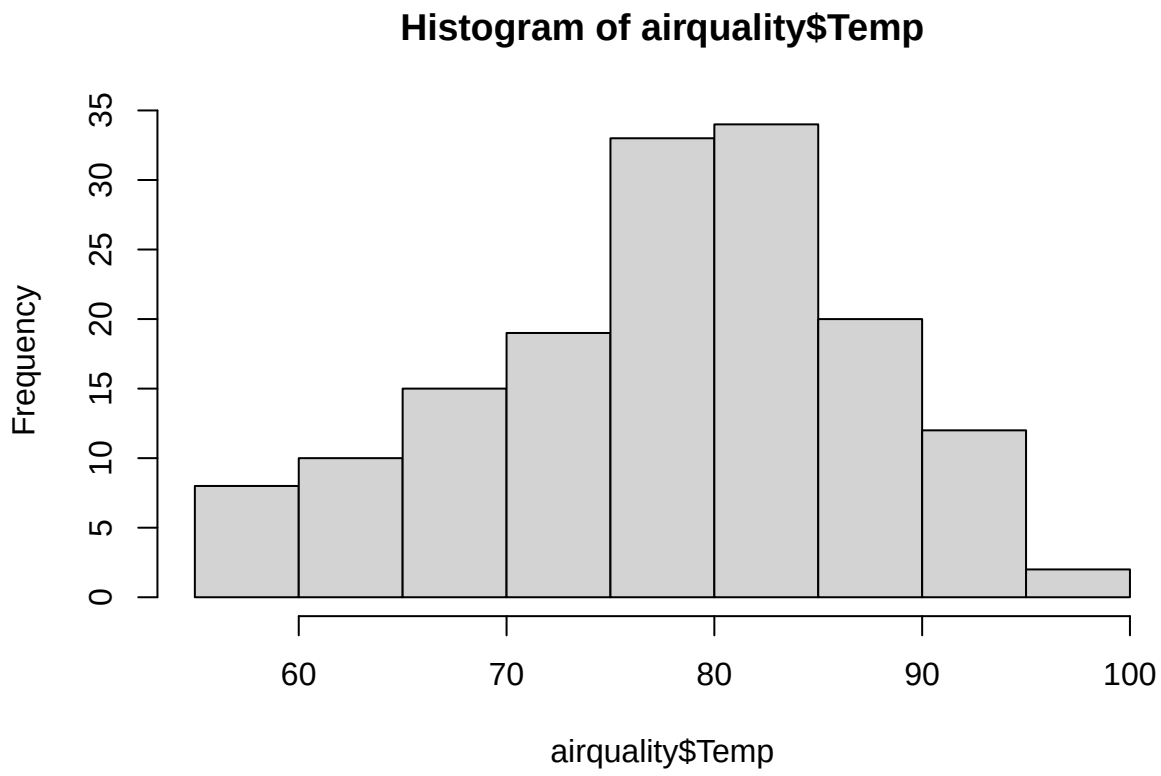
```
## corrplot 0.92 loaded
```

```
library(ggplot2)
library(gridExtra)
library(Lock5Data)
library(maps)
library(mapproj)
```

Part 1 - Basic plotting in ggplot2

Histograms

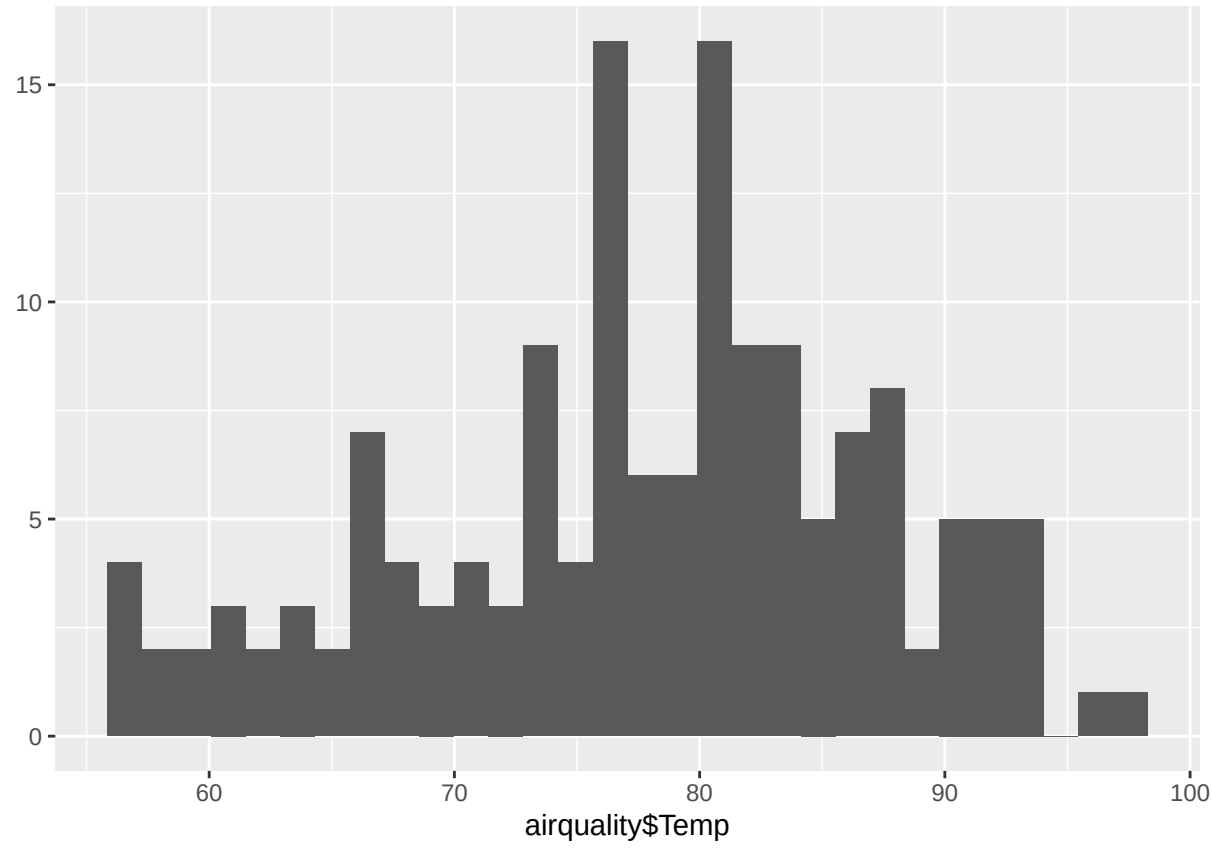
```
hist(airquality$Temp)
```




```
qplot(airquality$Temp)
```

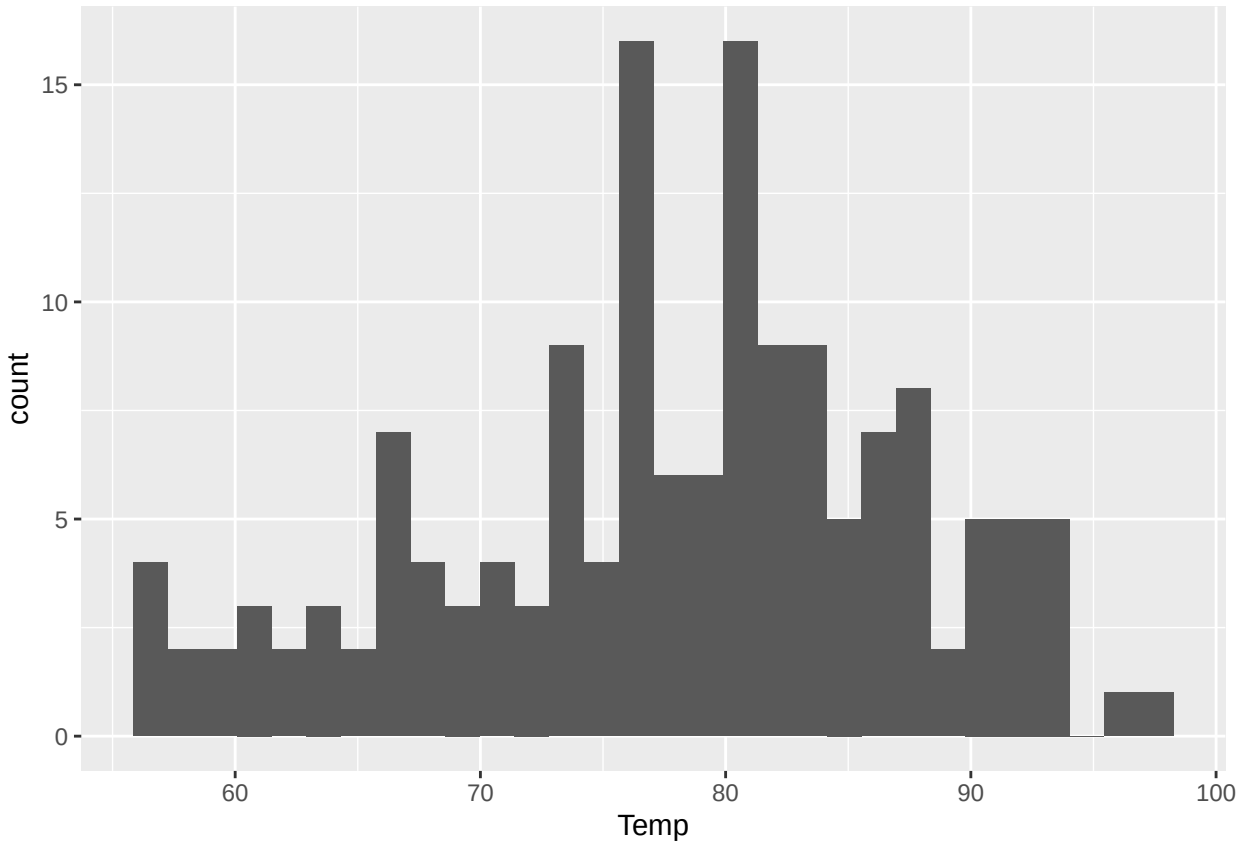
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



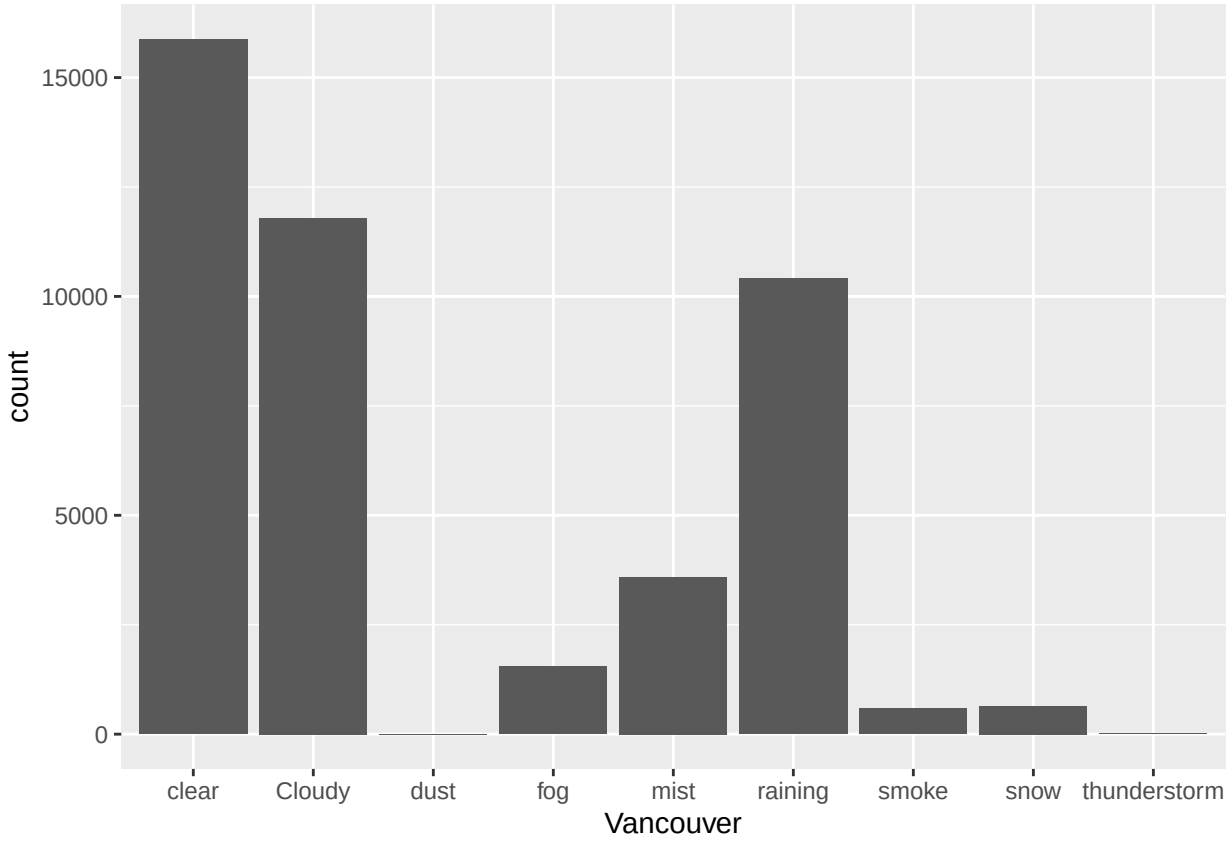

```
ggplot(airquality, aes(x=Temp)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



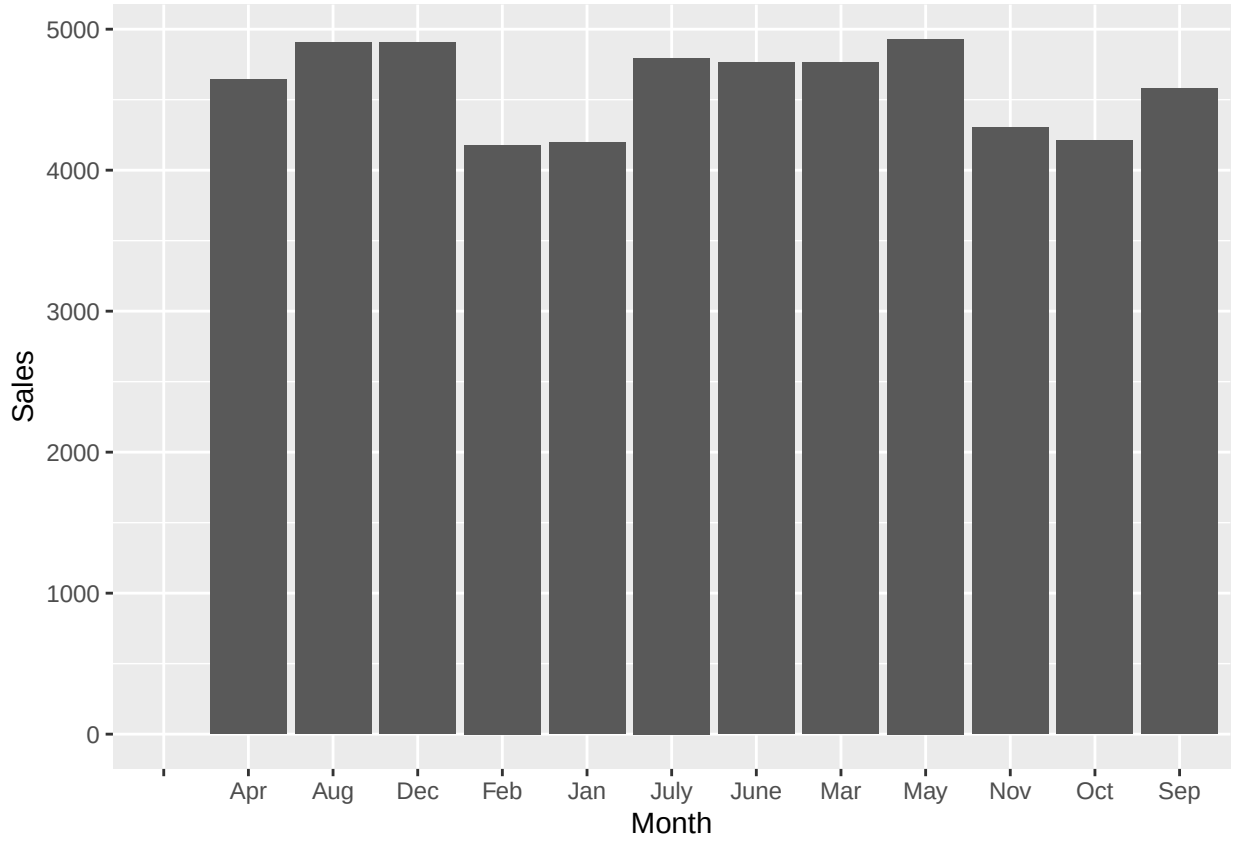
Bar plots

```
df_desc <- read.csv("../data/historical-hourly-weather-data/weather_description.csv")
ggplot(df_desc, aes(x=Vancouver)) + geom_bar()
```




```
ggplot(RetailSales, aes(x=Month, y=Sales)) + geom_bar(stat="identity")
```

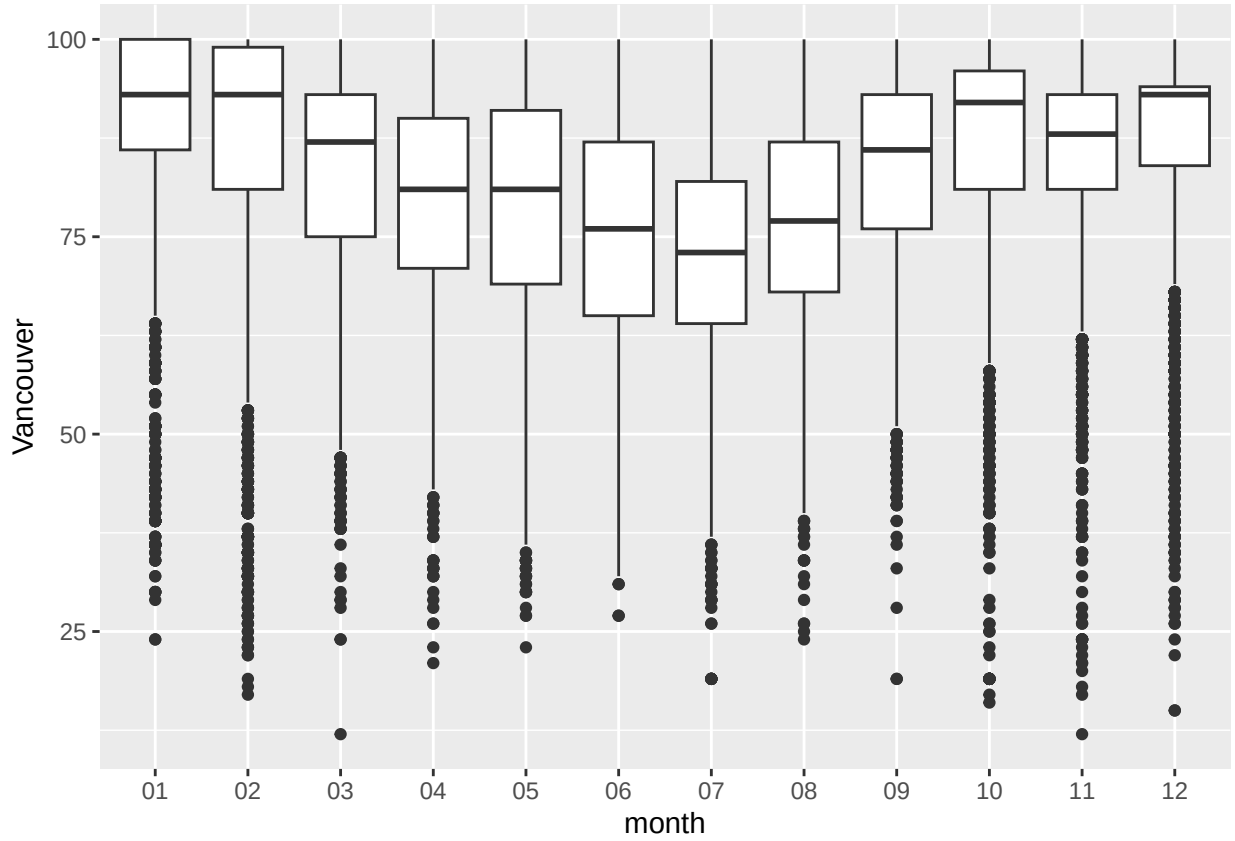
```
## Warning: Removed 15 rows containing missing values (‘position_stack()‘).
```



Box plots

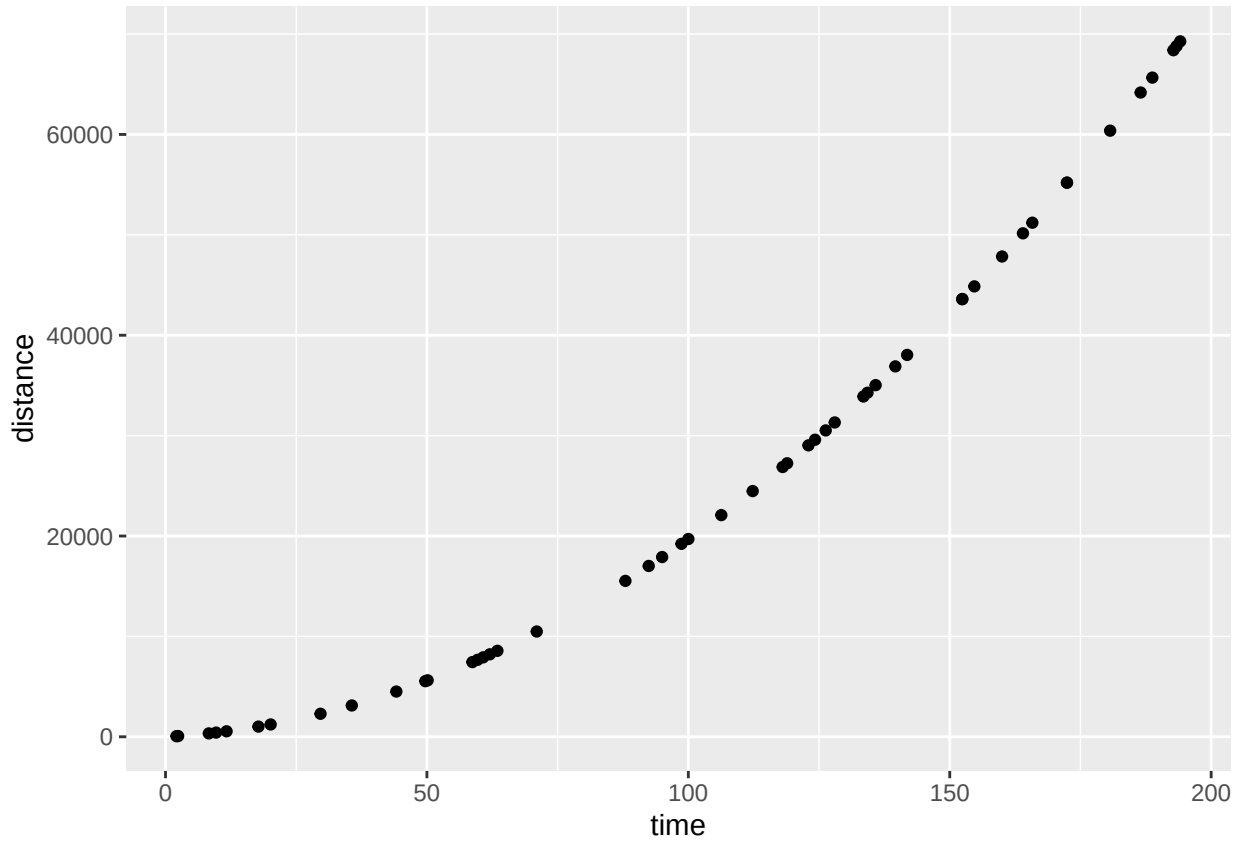
```
df_hum <- read.csv("../data/historical-hourly-weather-data/humidity.csv")
df_hum$datetime <- as.character(df_hum$datetime)
df_hum$month <- substr(df_hum$datetime, 6, 7)
ggplot(df_hum, aes(x=month, y=Vancouver)) + geom_boxplot()
```

```
## Warning: Removed 1826 rows containing non-finite values (‘stat_boxplot()‘).
```

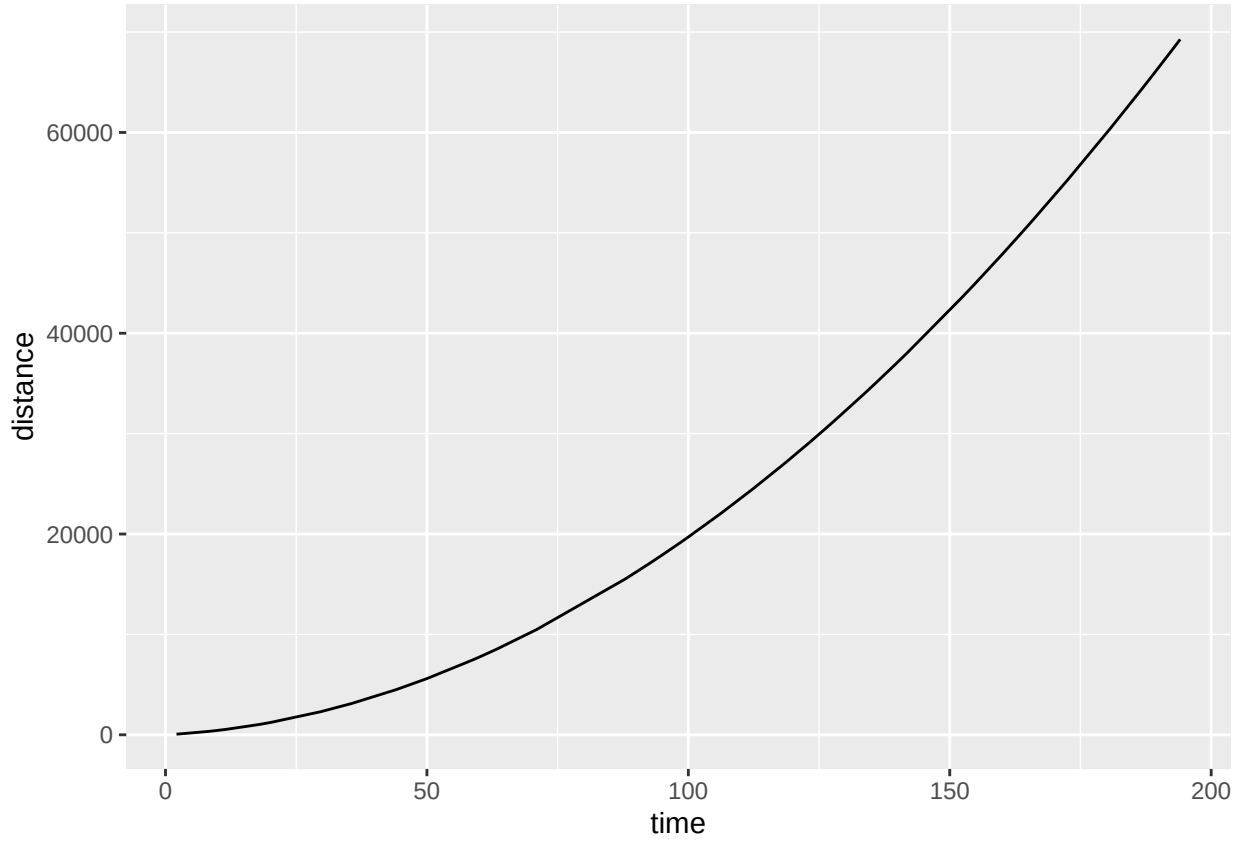


Scatter plots and line plots

```
a = 3.4
v0 = 27
time <- runif(50, min=0, max=200)
distance <- sapply(time, function(x) v0 * x + 0.5 * a * x^2)
df <- data.frame(time,distance)
ggplot(df, aes(x=time, y=distance)) + geom_point()
```




```
ggplot(df, aes(x=time, y=distance)) + geom_line()
```

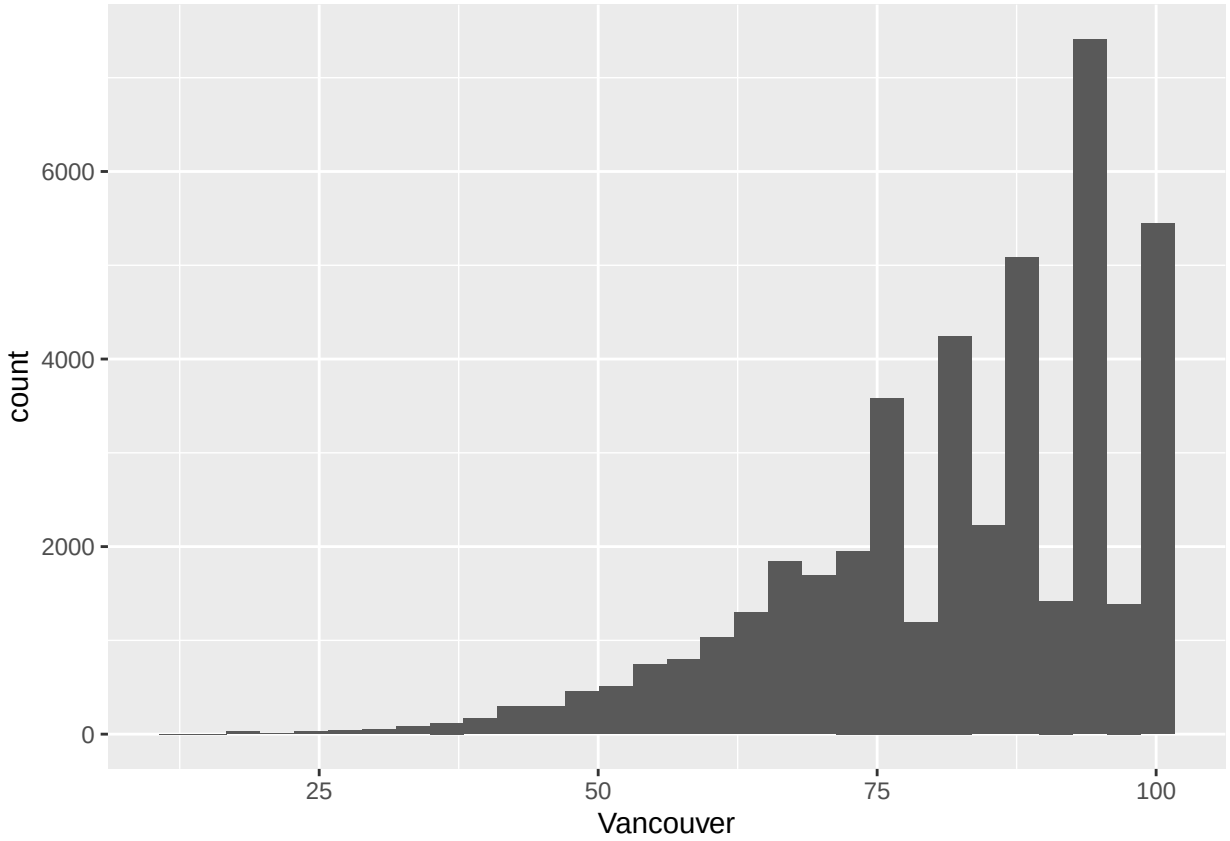


Changing histogram defaults and adding aesthetics

```
df_hum <- read.csv("../data/historical-hourly-weather-data/humidity.csv")
ggplot(df_hum, aes(x=Vancouver)) + geom_histogram()
```

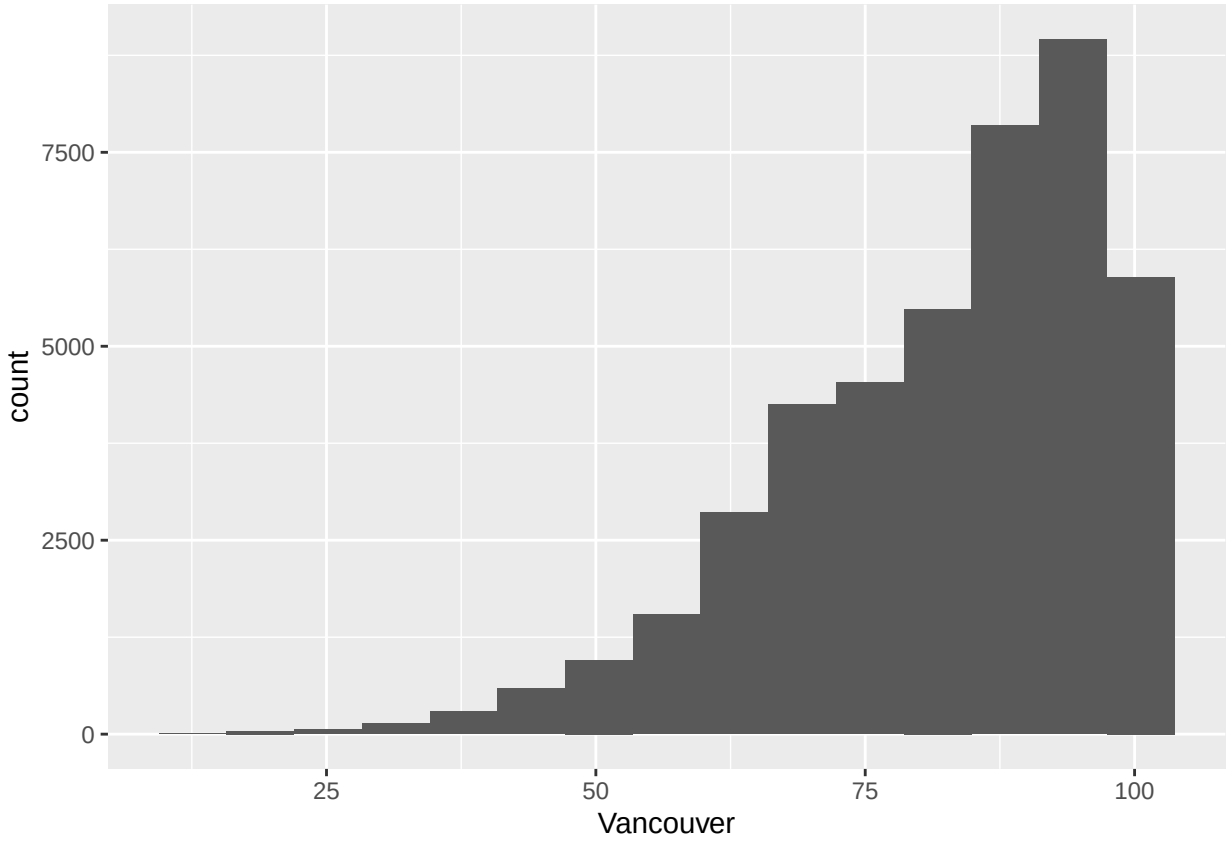
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1826 rows containing non-finite values (`stat_bin()`).
```



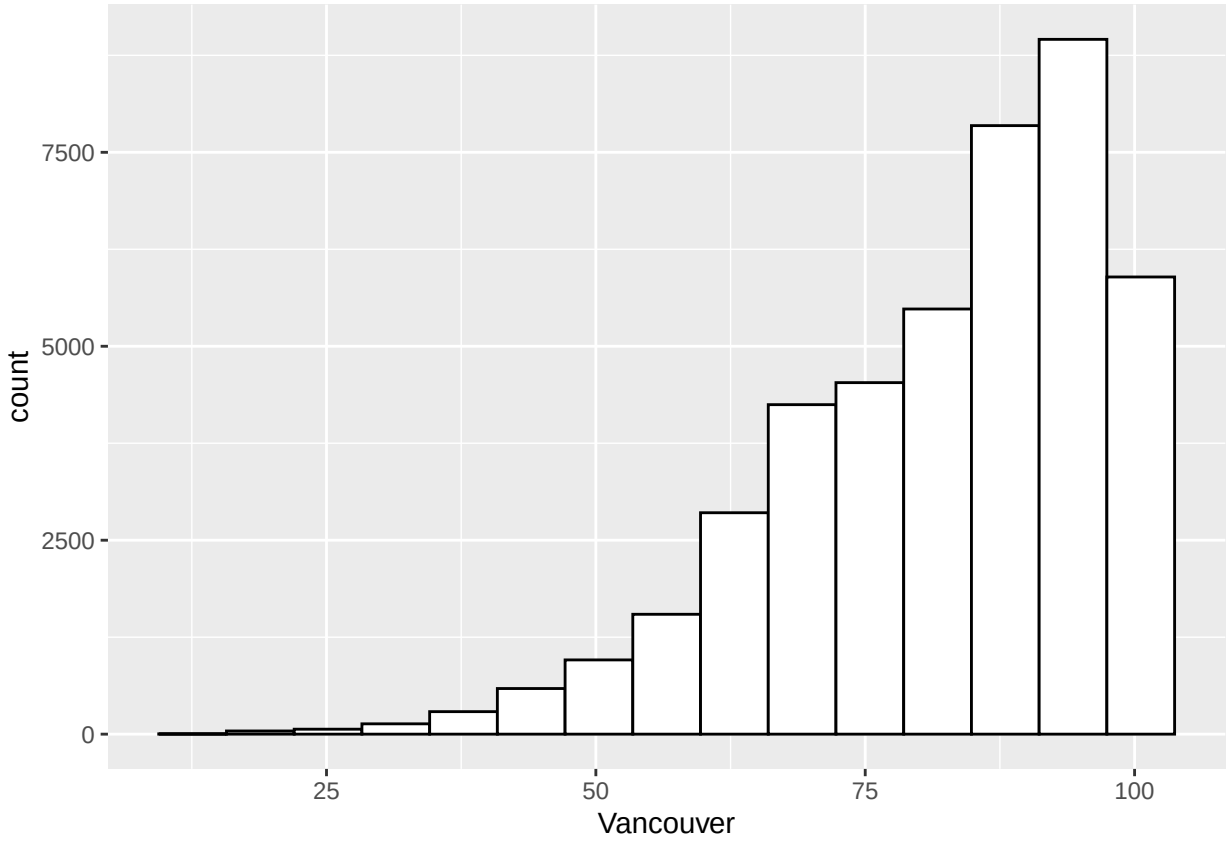

```
ggplot(df_hum, aes(x=Vancouver)) + geom_histogram(bins=15)
```

```
## Warning: Removed 1826 rows containing non-finite values (‘stat_bin()‘).
```



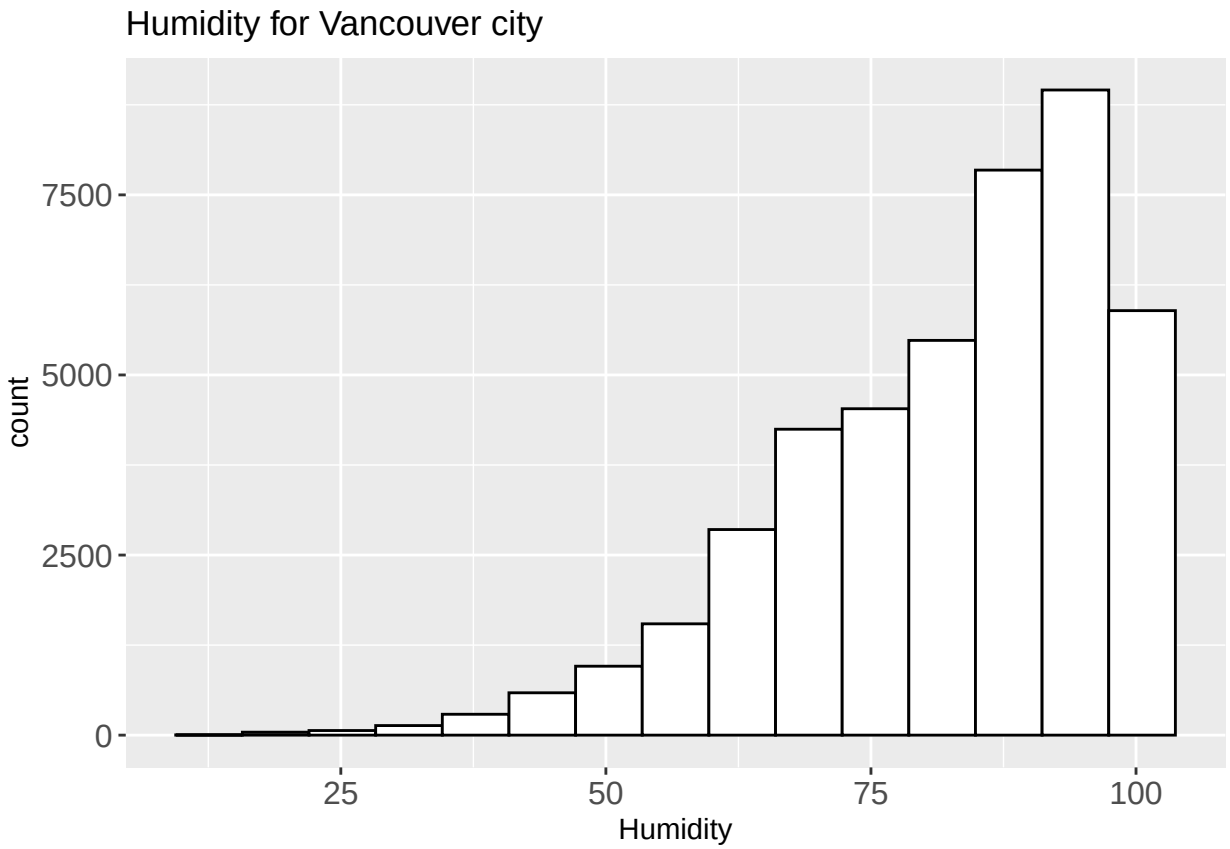

```
ggplot(df_hum, aes(x=Vancouver)) + geom_histogram(bins=15, fill="white", color=1)
```

```
## Warning: Removed 1826 rows containing non-finite values (‘stat_bin()‘).
```




```
ggplot(df_hum, aes(x=Vancouver)) +
geom_histogram(bins=15, fill="white", color=1) +
ggtitle("Humidity for Vancouver city") +
xlab("Humidity") +
theme(axis.text.x=element_text(size=12), axis.text.y=element_text(size=12))
```

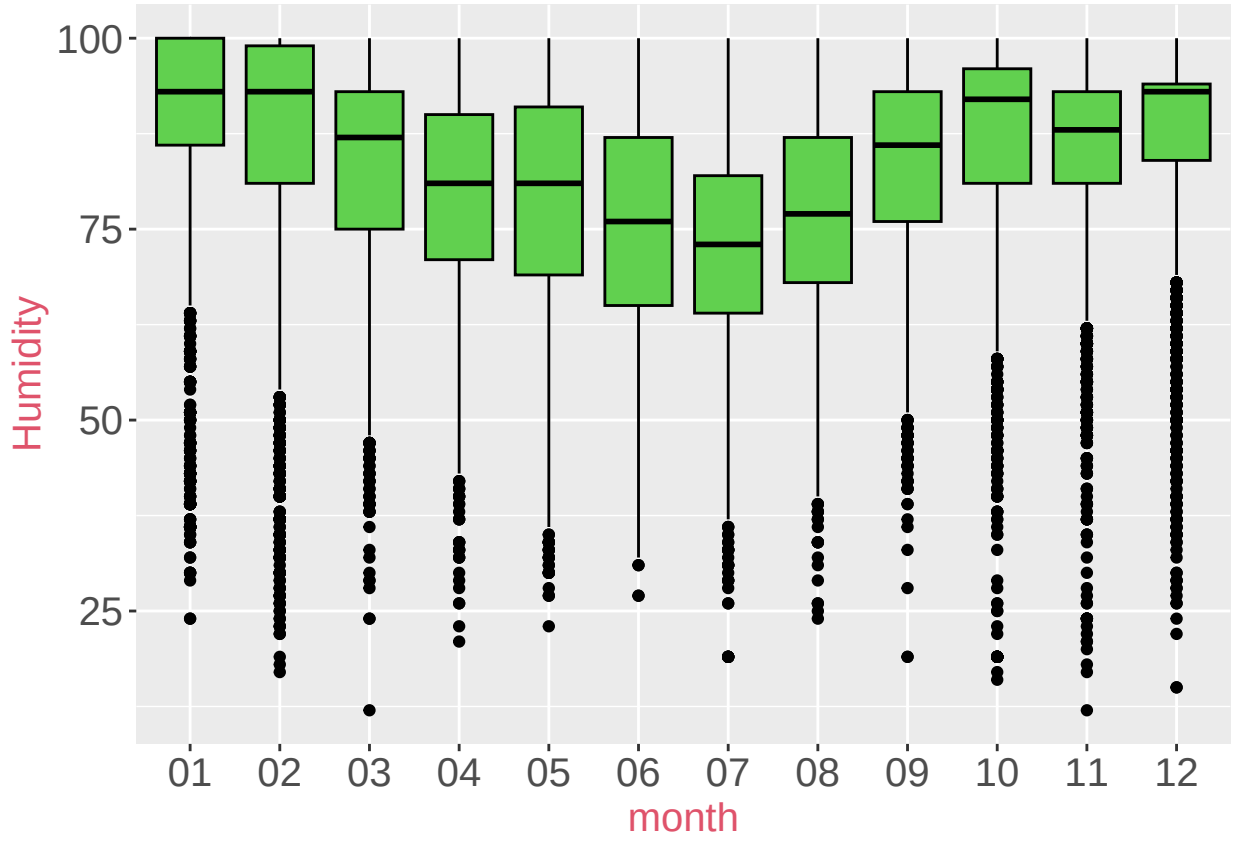
```
## Warning: Removed 1826 rows containing non-finite values (‘stat_bin()‘).
```



Changing boxplot defaults and adding aesthetics

```
df_hum <- read.csv("../data/historical-hourly-weather-data/humidity.csv")
df_hum$datetime <- as.character(df_hum$datetime)
df_hum$month <- substr(df_hum$datetime, 6, 7)
ggplot(df_hum, aes(x=month, y=Vancouver)) +
geom_boxplot(color=1, fill=3) +
ylab("Humidity") +
theme(axis.text.x=element_text(size=15),
axis.text.y=element_text(size=15),
axis.title.x=element_text(size=15, color=2),
axis.title.y=element_text(size=15, color=2))
```

```
## Warning: Removed 1826 rows containing non-finite values (‘stat_boxplot()‘).
```



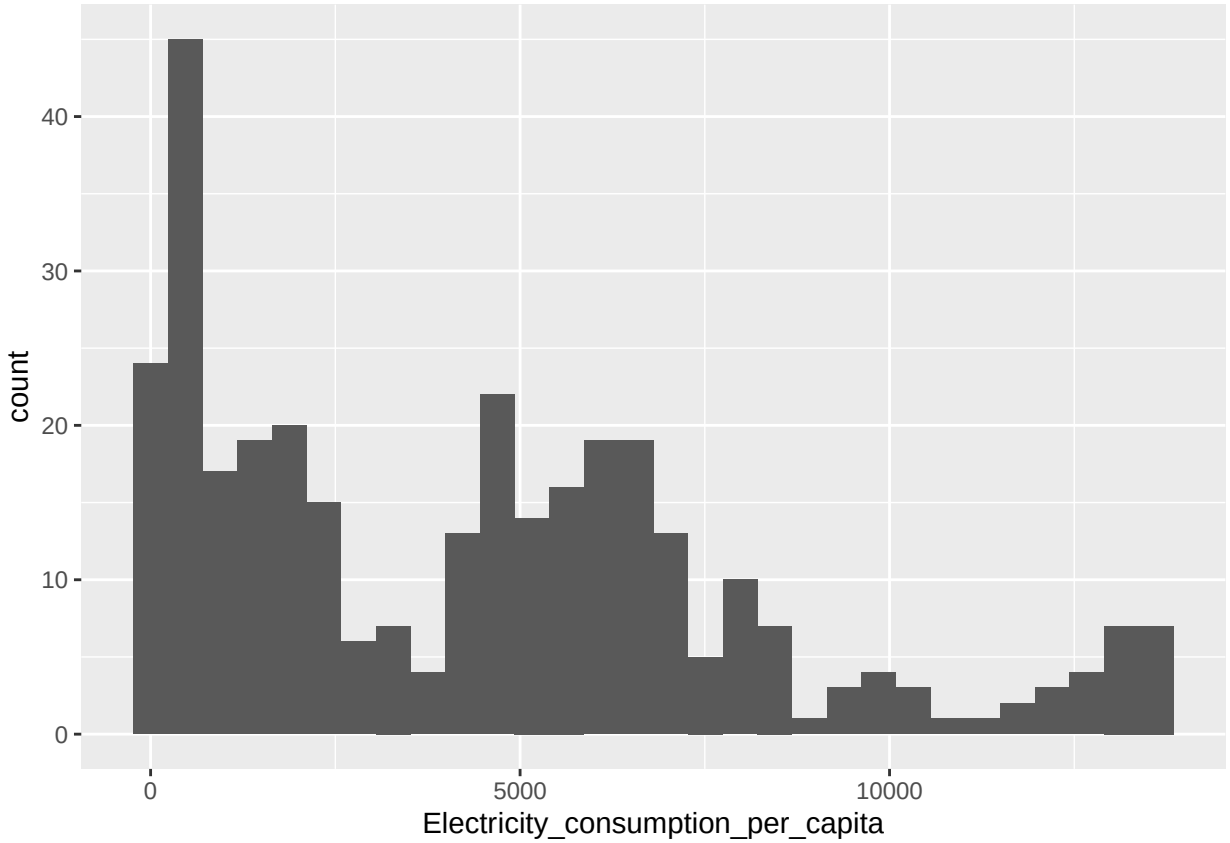
Part 2 - Grammar of graphics and visual components

Layers

```
df <- read.csv("../data/gapminder-data.csv")
p1 <- ggplot(df, aes(x=Electricity_consumption_per_capita))
p2 <- p1 + geom_histogram()
p3 <- p1 + geom_histogram(bins=15)
p4 <- p3 + xlab("Electricity consumption per capita")
p2
```

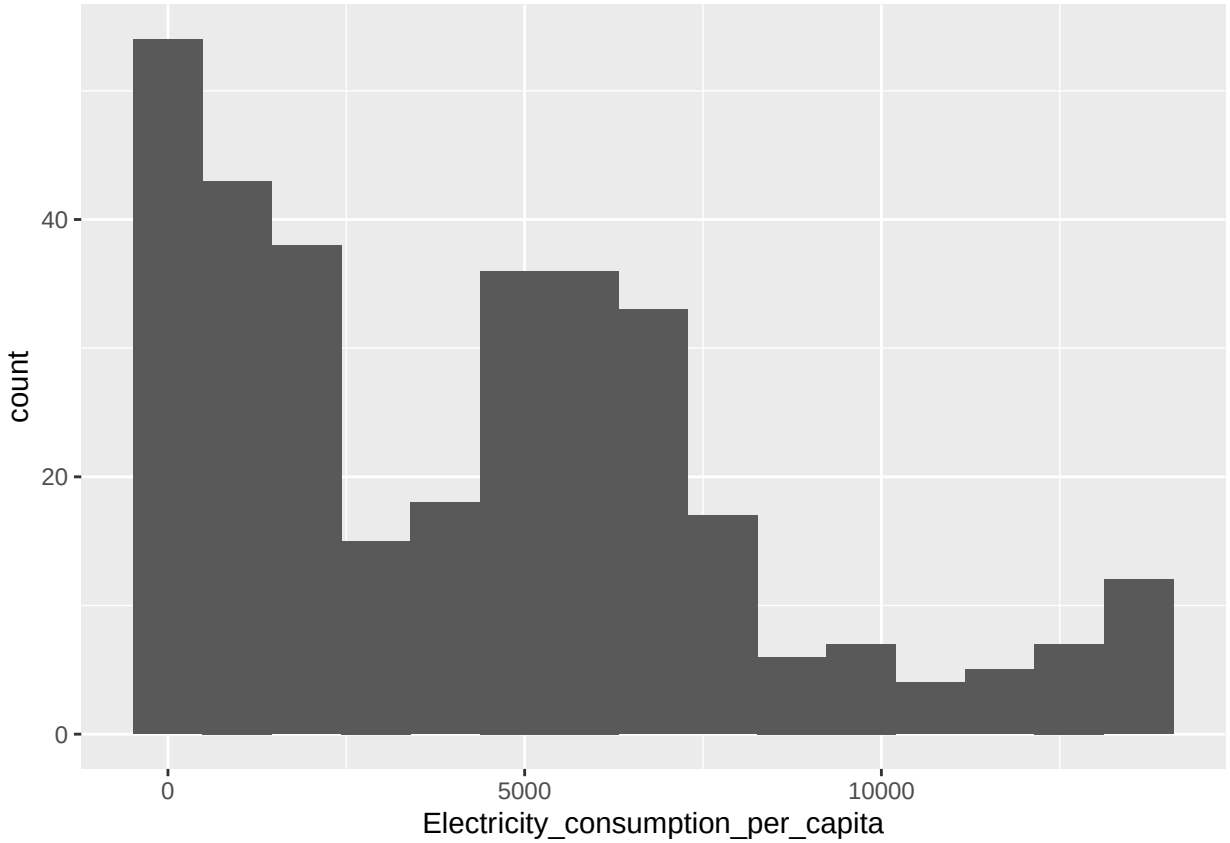
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1181 rows containing non-finite values (`stat_bin()`).
```



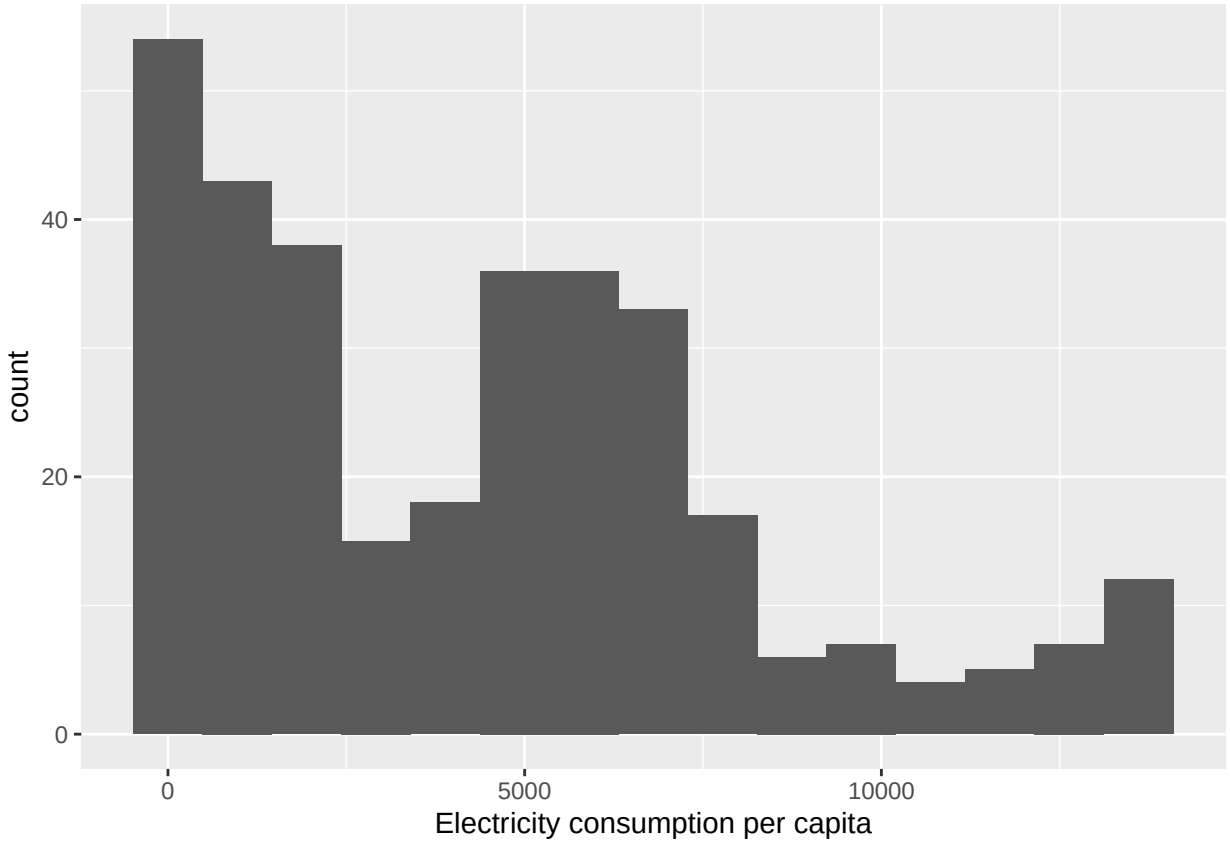

```
p3
```

```
## Warning: Removed 1181 rows containing non-finite values (‘stat_bin()‘).
```




```
p4
```

```
## Warning: Removed 1181 rows containing non-finite values (‘stat_bin()‘).
```



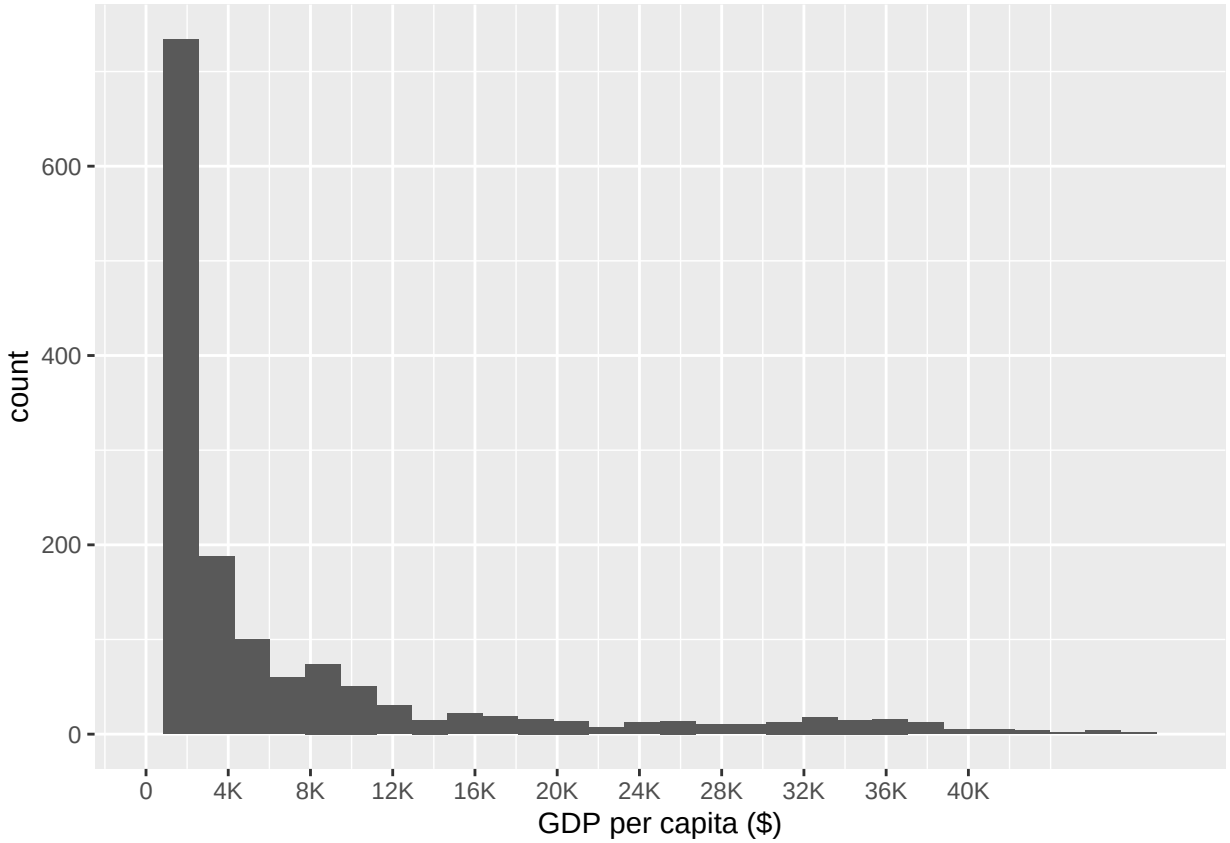
Scales

```
p1 <- ggplot(df, aes(x=gdp_per_capita))
p2 <- p1 + geom_histogram()
p3 <- p2 + scale_x_continuous(name='GDP per capita ($)',
                              limits=c(0, 50000),
                              breaks=seq(0, 40000, 4000),
                              labels=c('0', '4K', '8K', '12K', '16K', '20K',
                                       '24K', '28K', '32K', '36K', '40K'))
p3
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

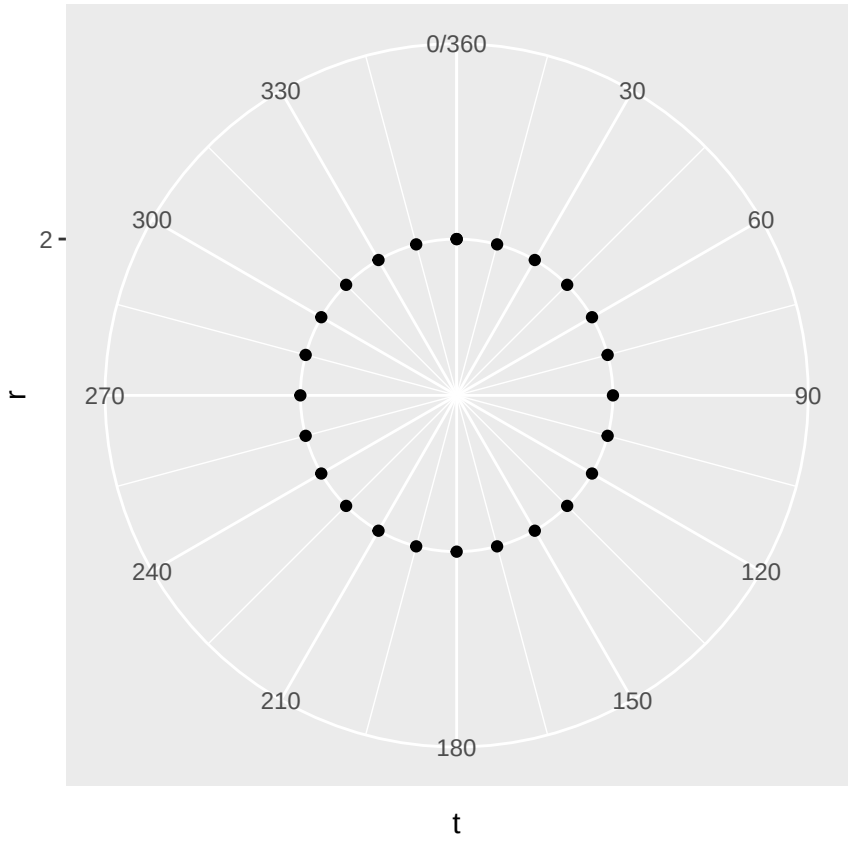
```
## Warning: Removed 7 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



Polar coordinates

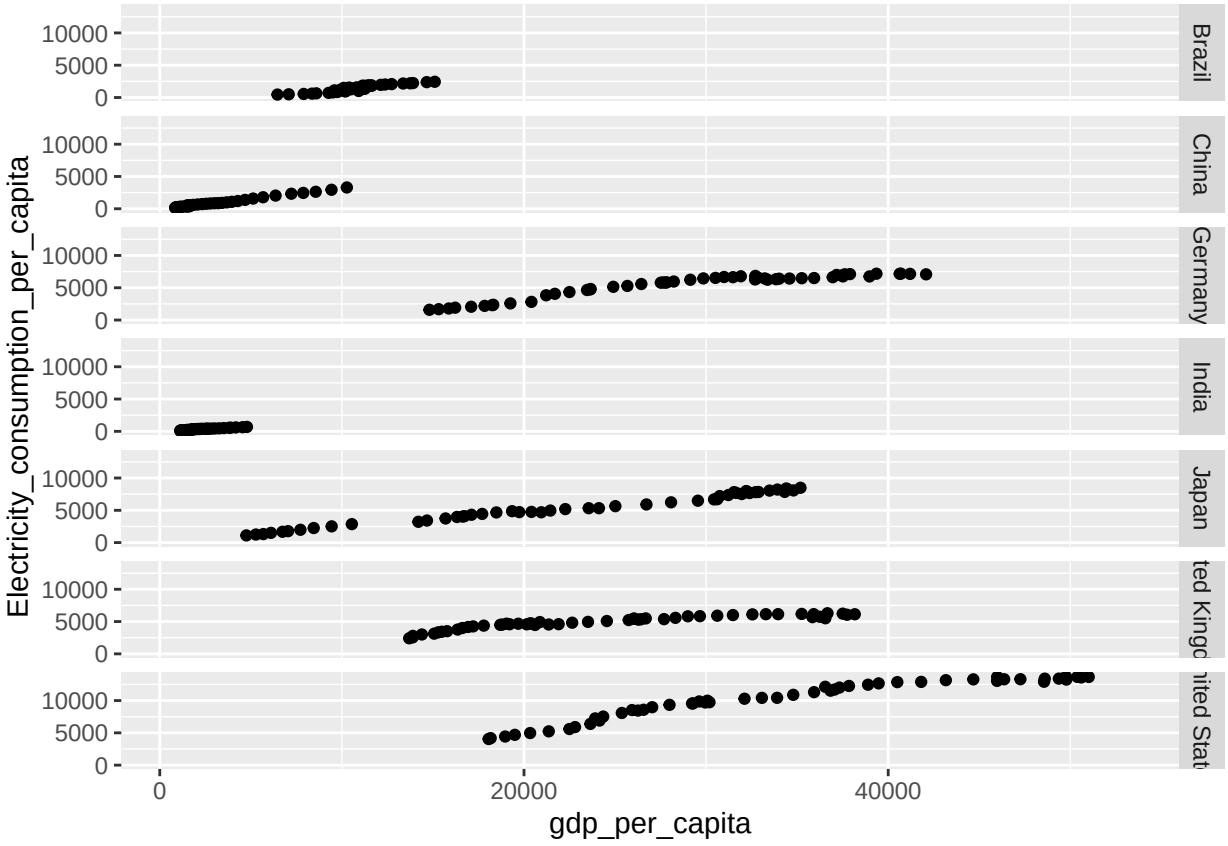
```
t <- seq(0, 360, by=15)
r <- 2
qplot(r, t) +
coord_polar(theta="y") +
scale_y_continuous(breaks=seq(0, 360, 30))
```



Facets

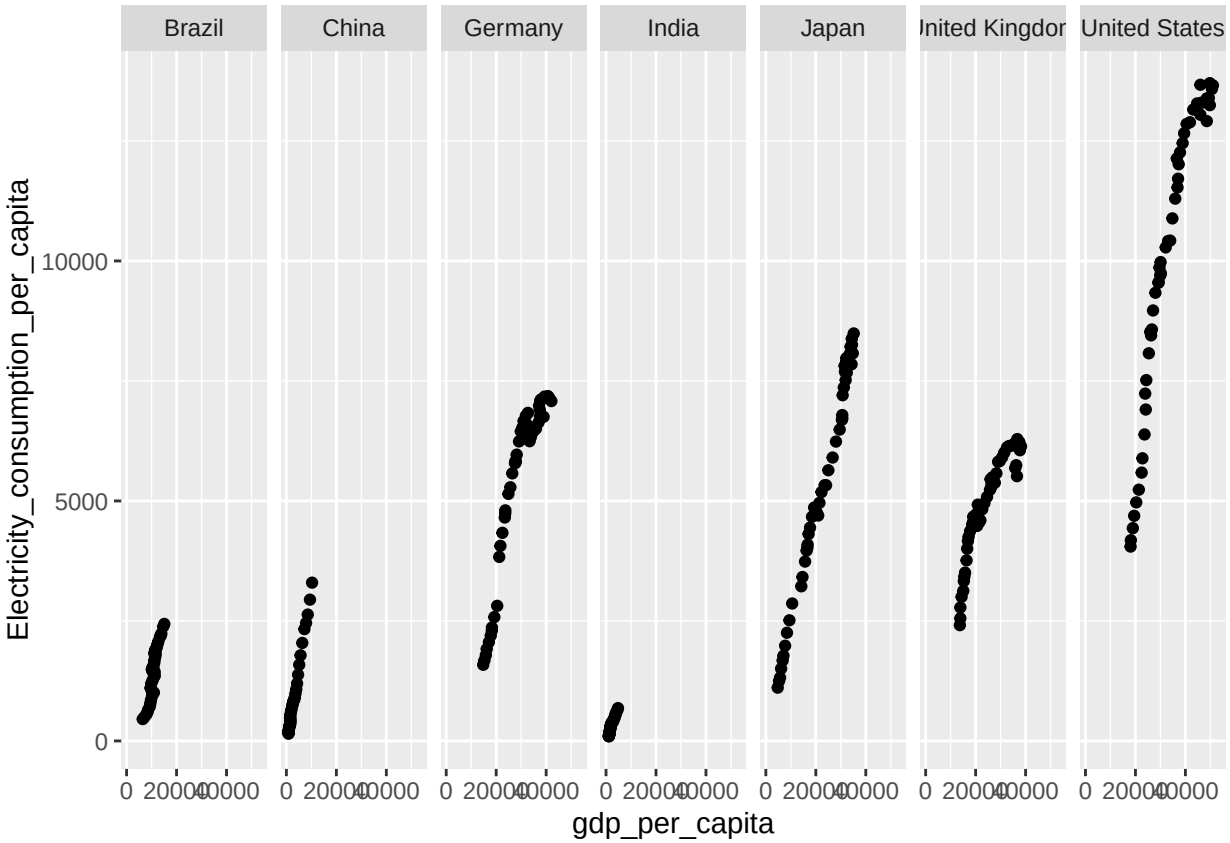
```
p <- ggplot(df, aes(x=gdp_per_capita, y=Electricity_consumption_per_capita)) + geom_point()
p + facet_grid(Country ~ .)
```

```
## Warning: Removed 1181 rows containing missing values (‘geom_point()‘).
```



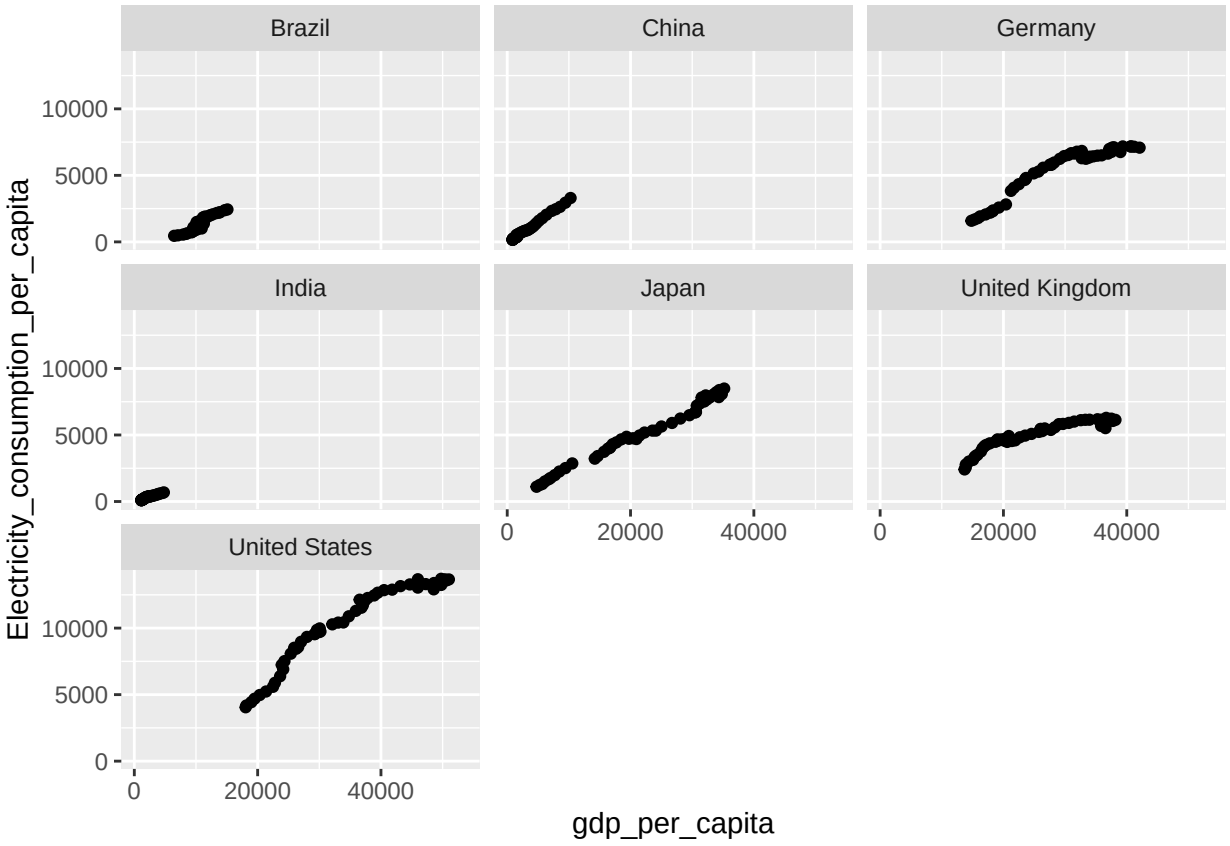

```
p + facet_grid(. ~ Country)
```

```
## Warning: Removed 1181 rows containing missing values (‘geom_point()‘).
```




```
p + facet_wrap(~Country)
```

```
## Warning: Removed 1181 rows containing missing values (‘geom_point()‘).
```



Shapes and colors

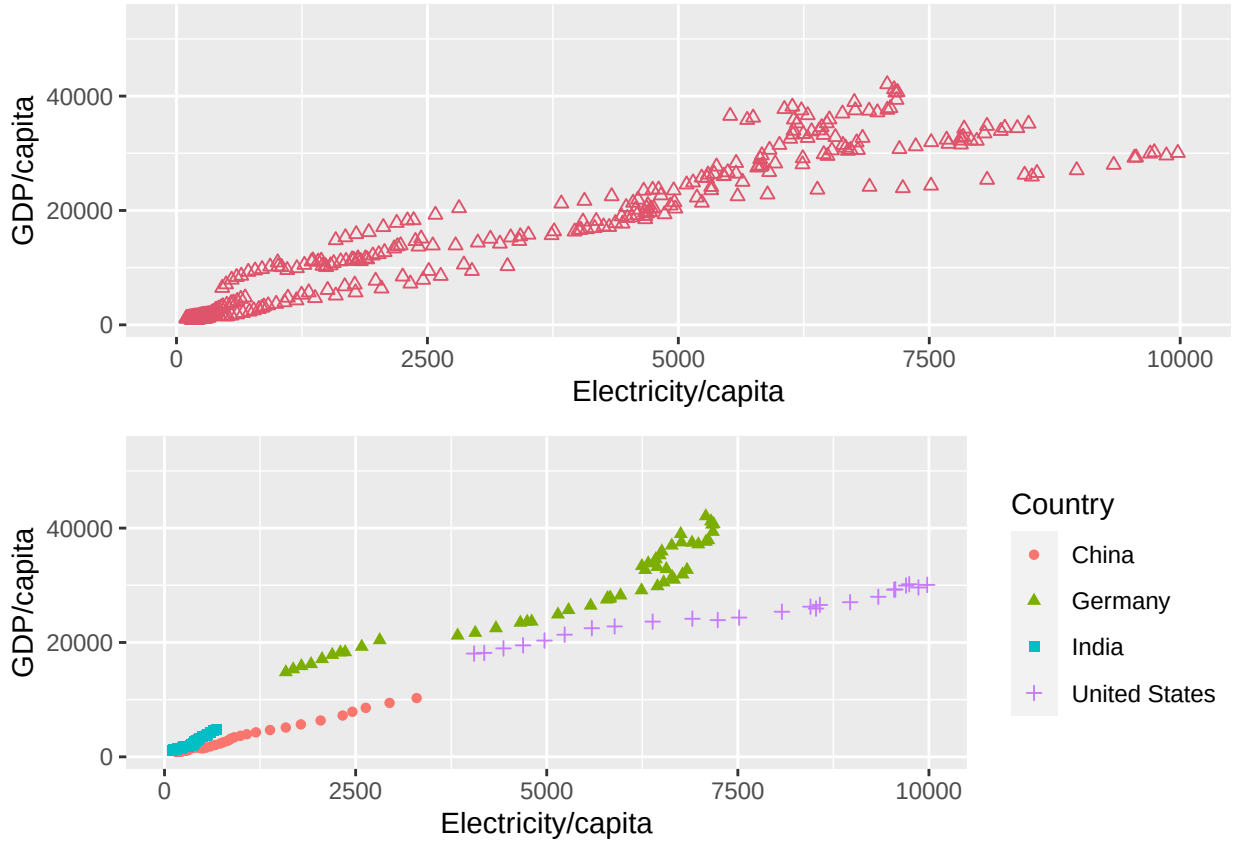
```
dfs <- subset(df, Country %in% c("Germany", "India", "China", "United States"))
var1 <- "Electricity_consumption_per_capita"
var2 <- "gdp_per_capita"
name1 <- "Electricity/capita"
name2 <- "GDP/capita"
p1 <- ggplot(df, aes_string(x=var1, y=var2)) +
geom_point(color=2, shape=2) +
xlim(0, 10000) + xlab(name1) + ylab(name2)
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
p2 <- ggplot(dfs, aes_string(x=var1, y=var2)) +
geom_point(aes(color=Country, shape=Country)) +
xlim(0, 10000) + xlab(name1) + ylab(name2)
grid.arrange(p1, p2, nrow = 2)
```

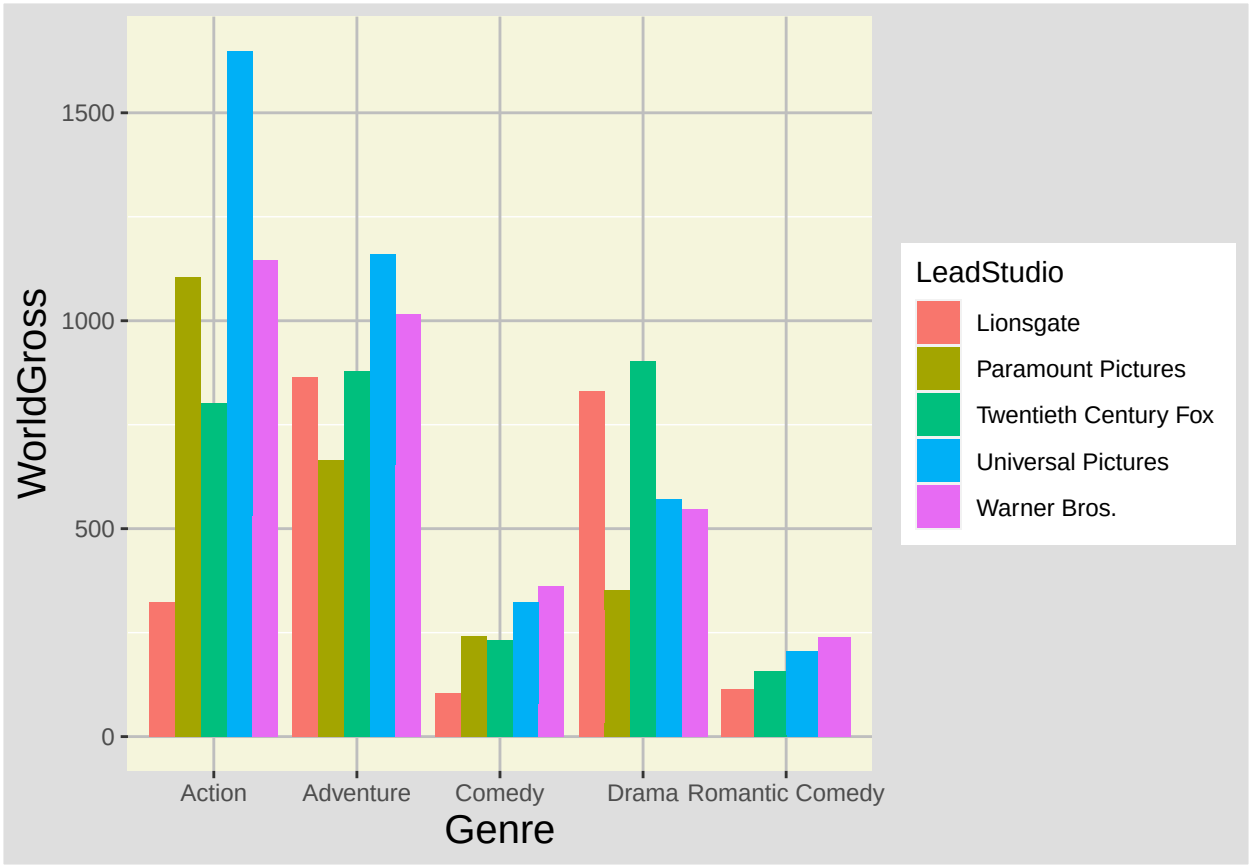
```
## Warning: Removed 1209 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 706 rows containing missing values (`geom_point()`).
```

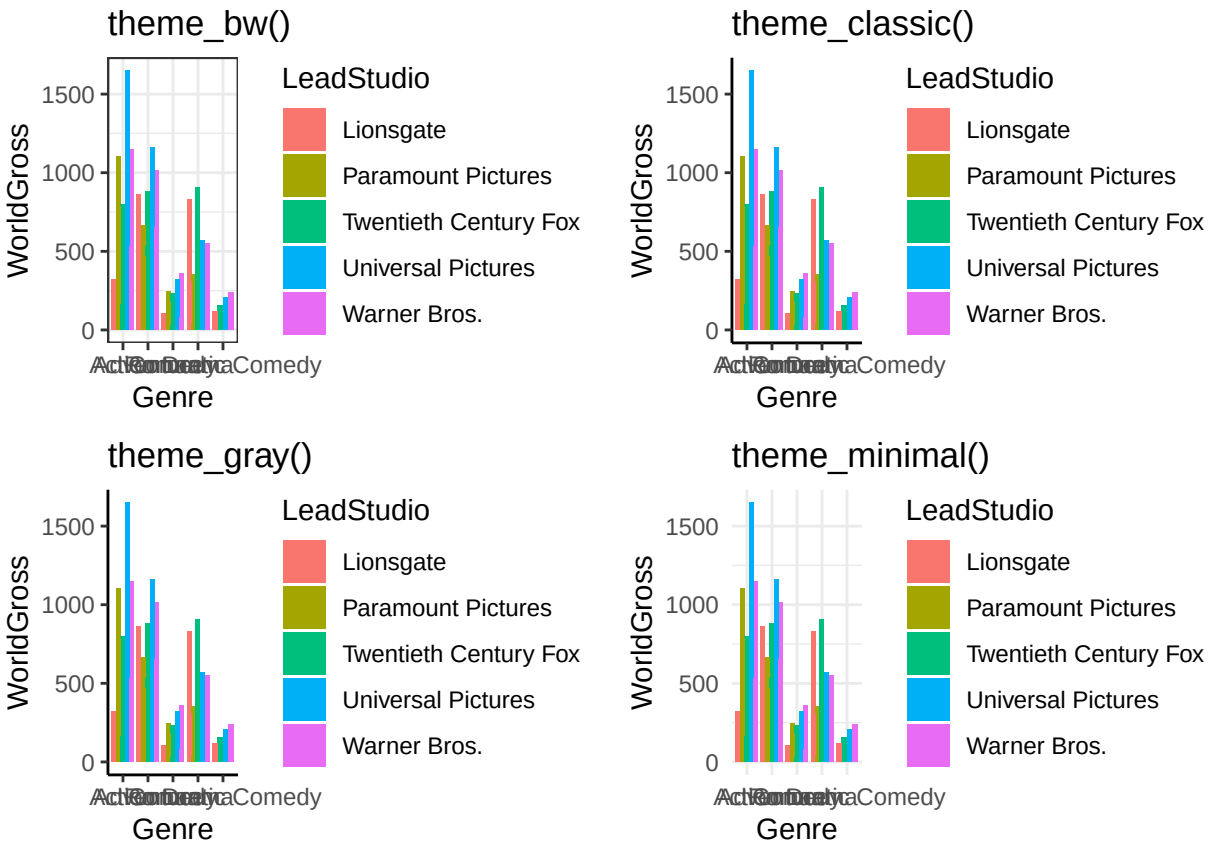


Themes

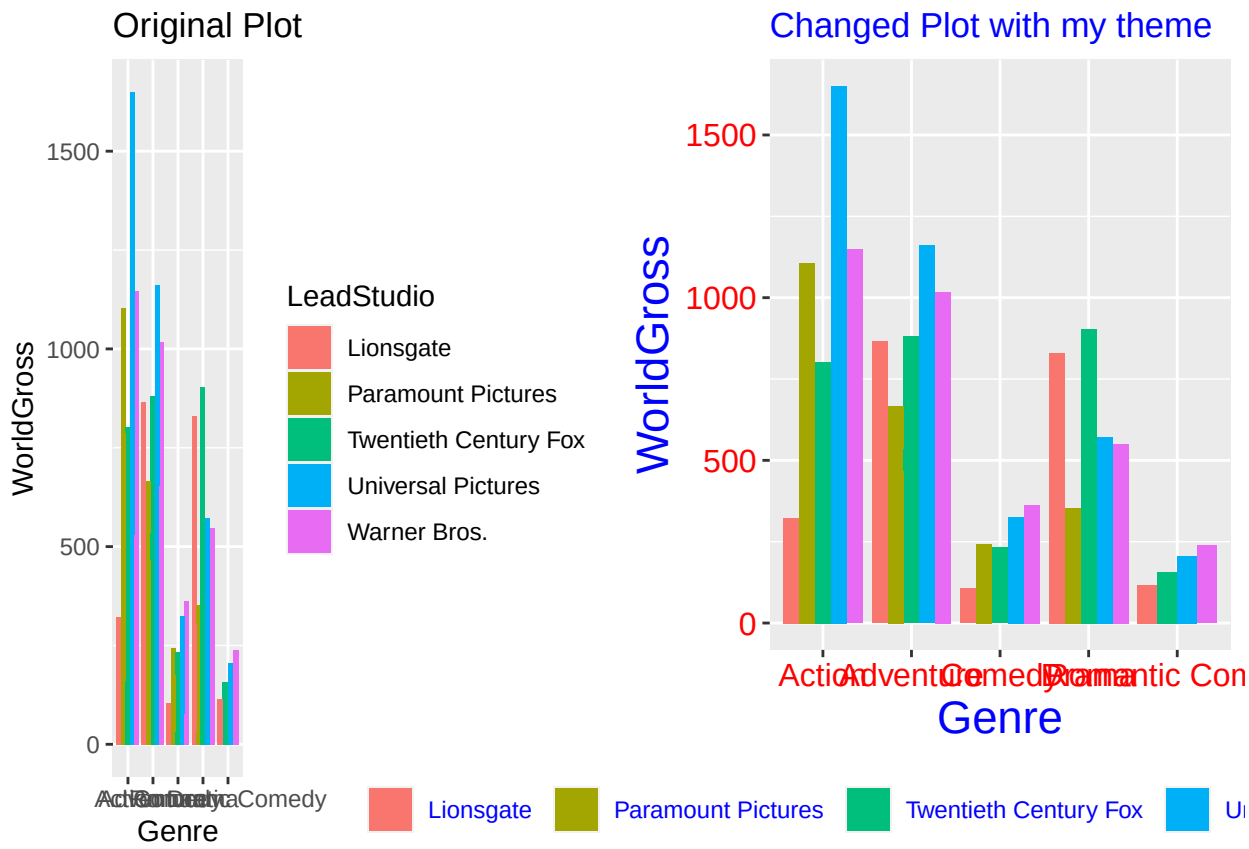
```
dfn <- subset(HollywoodMovies,
              Genre %in% c("Action", "Adventure", "Comedy", "Drama",
              "Romantic Comedy") &
              LeadStudio %in% c("Lionsgate ", "Paramount Pictures ",
              "Twentieth Century Fox ", "Universal Pictures ", "Warner Bros. "))
p1 <- ggplot(dfn, aes(x=Genre, y=WorldGross))
p2 <- p1 + geom_bar(aes(fill=LeadStudio), stat="Identity", position="dodge")
p3 <- p2 + theme(axis.title.x=element_text(size=15),
                 axis.title.y=element_text(size=15),
                 plot.background=element_rect(fill="gray87"),
                 panel.background=element_rect(fill="beige"),
                 panel.grid.major=element_line(color="Gray", linetype=1))
p3
```



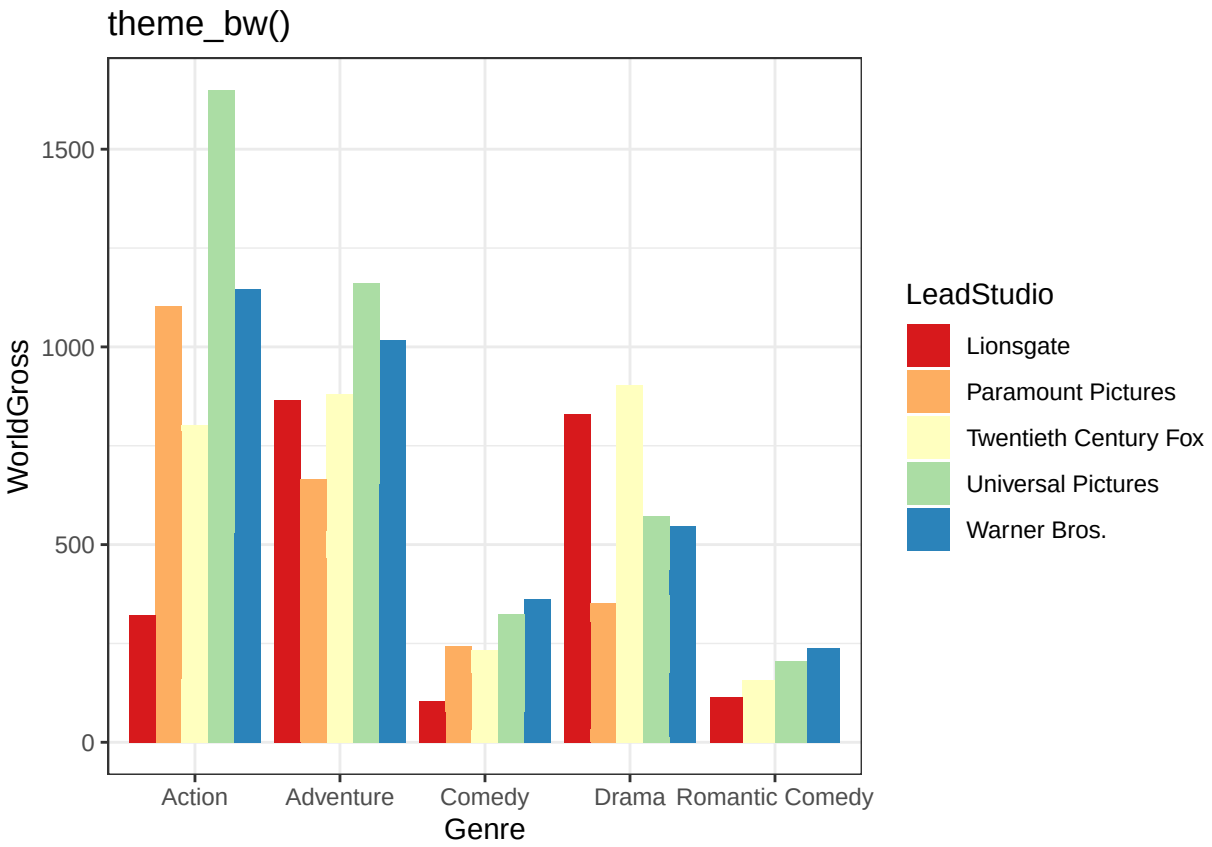

```
p4 <- p2 + theme_bw() + ggtitle("theme_bw()")
p5 <- p2 + theme_classic() + ggtitle("theme_classic()")
p6 <- p2 + theme_classic() + ggtitle("theme_gray()")
p7 <- p2 + theme_minimal() + ggtitle("theme_minimal()")
grid.arrange(p4, p5, p6, p7, nrow=2, ncol=2)
```



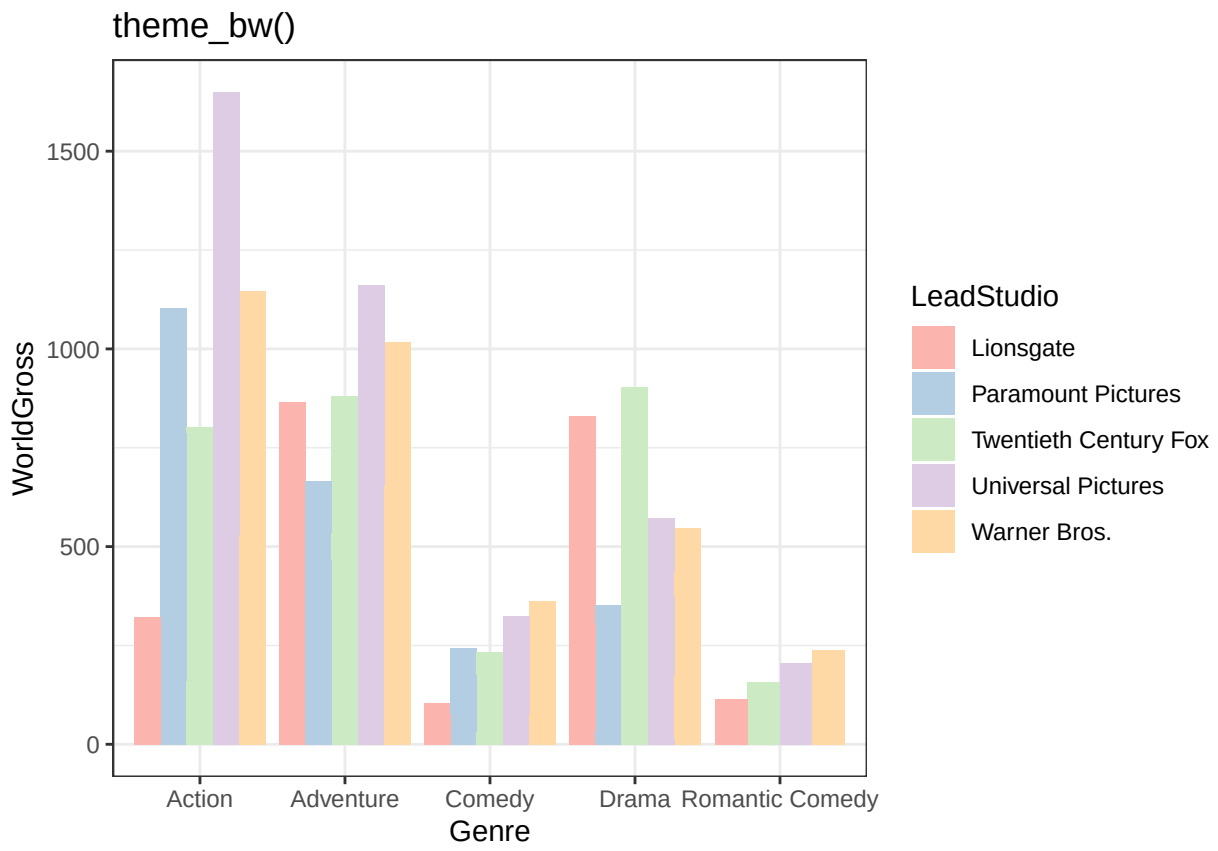

```
mytheme <- theme(legend.title=element_blank(),
legend.position="bottom",
text = element_text(color="Blue"),
axis.text=element_text(size=12, color="Red"),
axis.title=element_text(size=rel(1.5)))
p2 <- p2 + ggtitle("Original Plot")
p8 <- p2 + mytheme + ggtitle("Changed Plot with my theme")
grid.arrange(p2, p8, ncol=2)
```



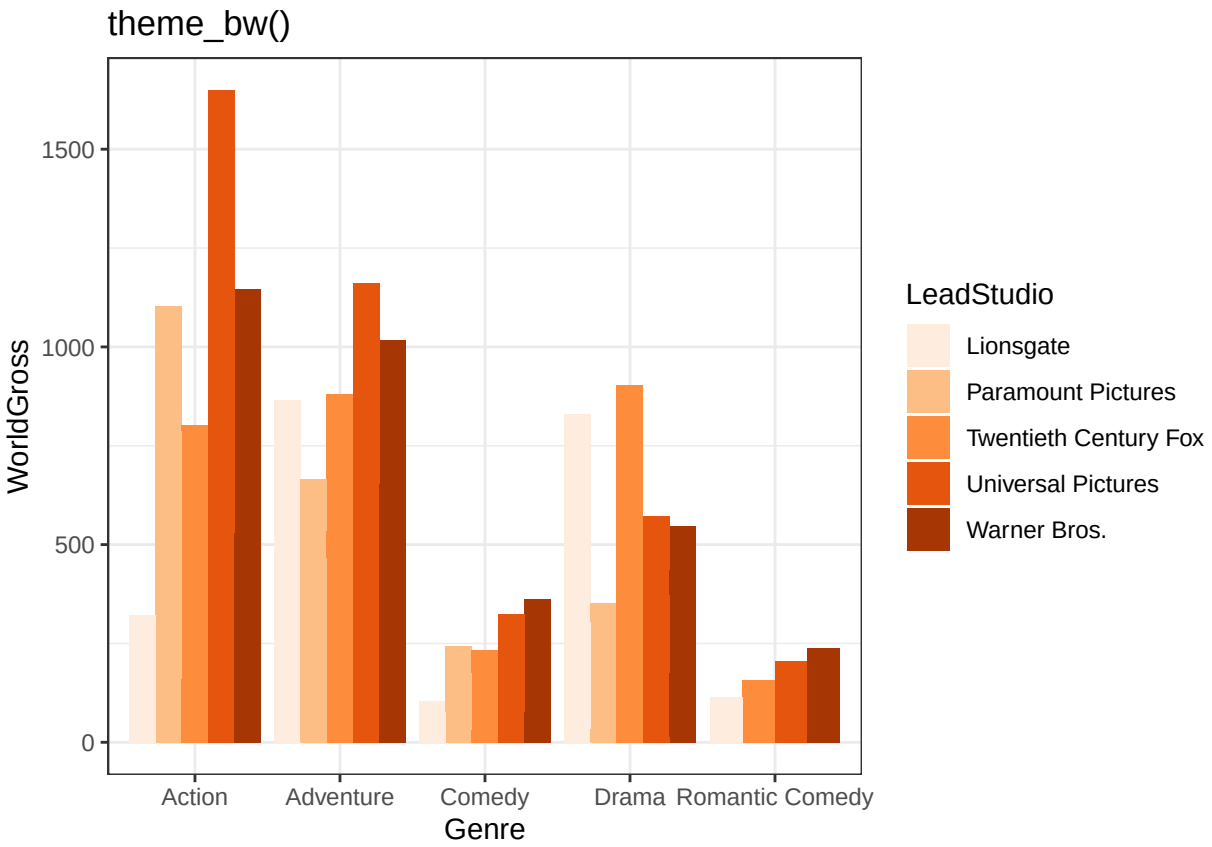

```
p4 + scale_fill_brewer(palette="Spectral")
```




```
p4 + scale_fill_brewer(palette="Pastel1")
```




```
p4 + scale_fill_brewer(palette="Oranges")
```

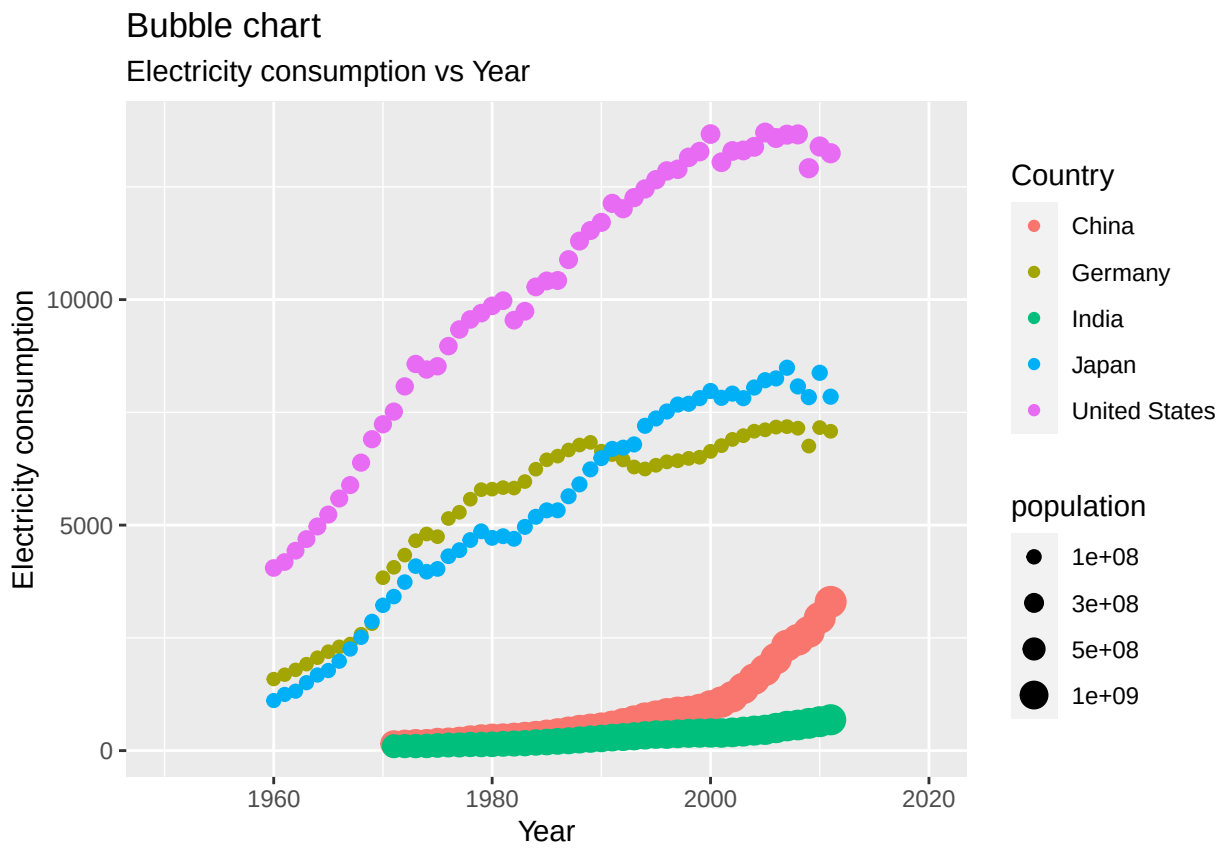


Part 3 - Advanced Geoms and Statistics

Bubble charts

```
df <- read.csv("../data/gapminder-data.csv")
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=
coord_cartesian(xlim=c(1950, 2020)) +
labs(subtitle="Electricity consumption vs Year", title="Bubble chart") +
ylab("Electricity consumption") +
scale_size(breaks=c(0, 1e+8, 0.3e+9, 0.5e+9, 1e+9, 1.5e+9), range=c(1, 5))
```

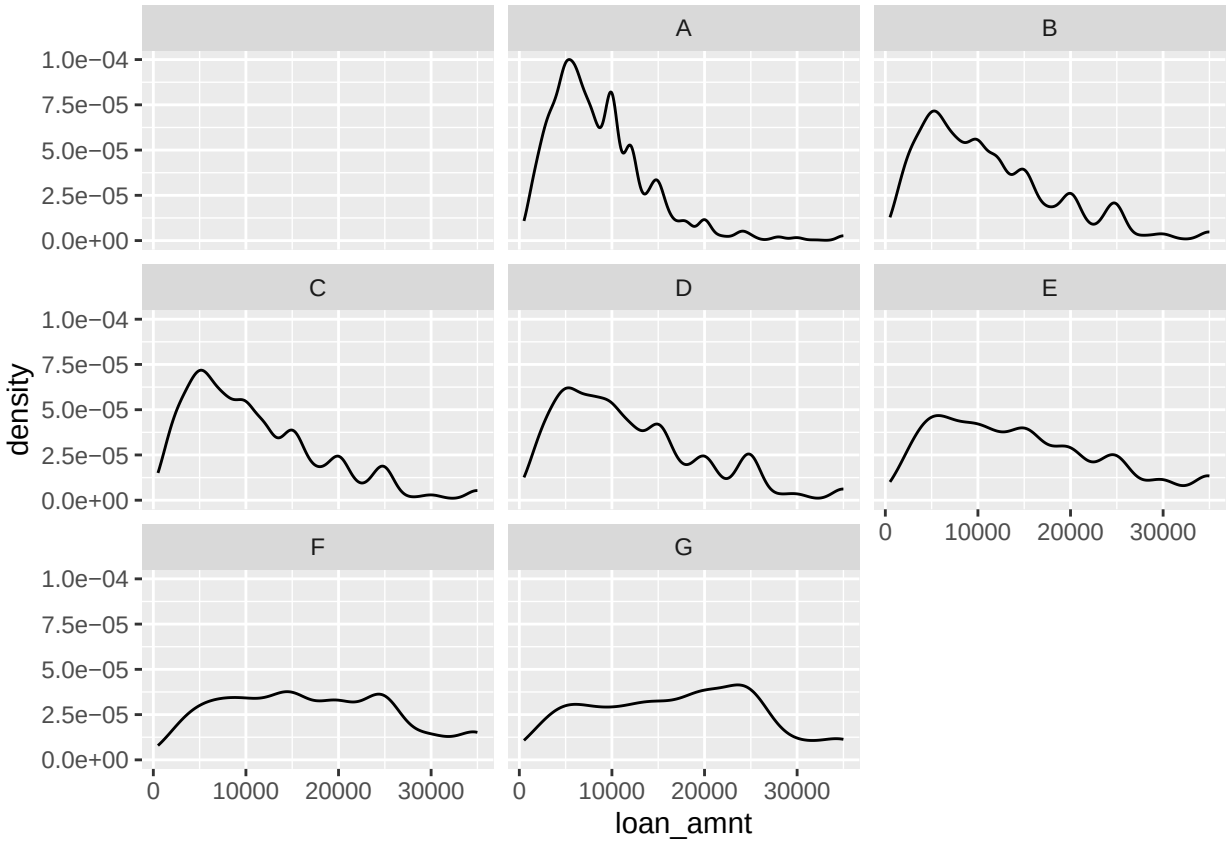
```
## Warning: Removed 842 rows containing missing values (`geom_point()`).
```



Density plots

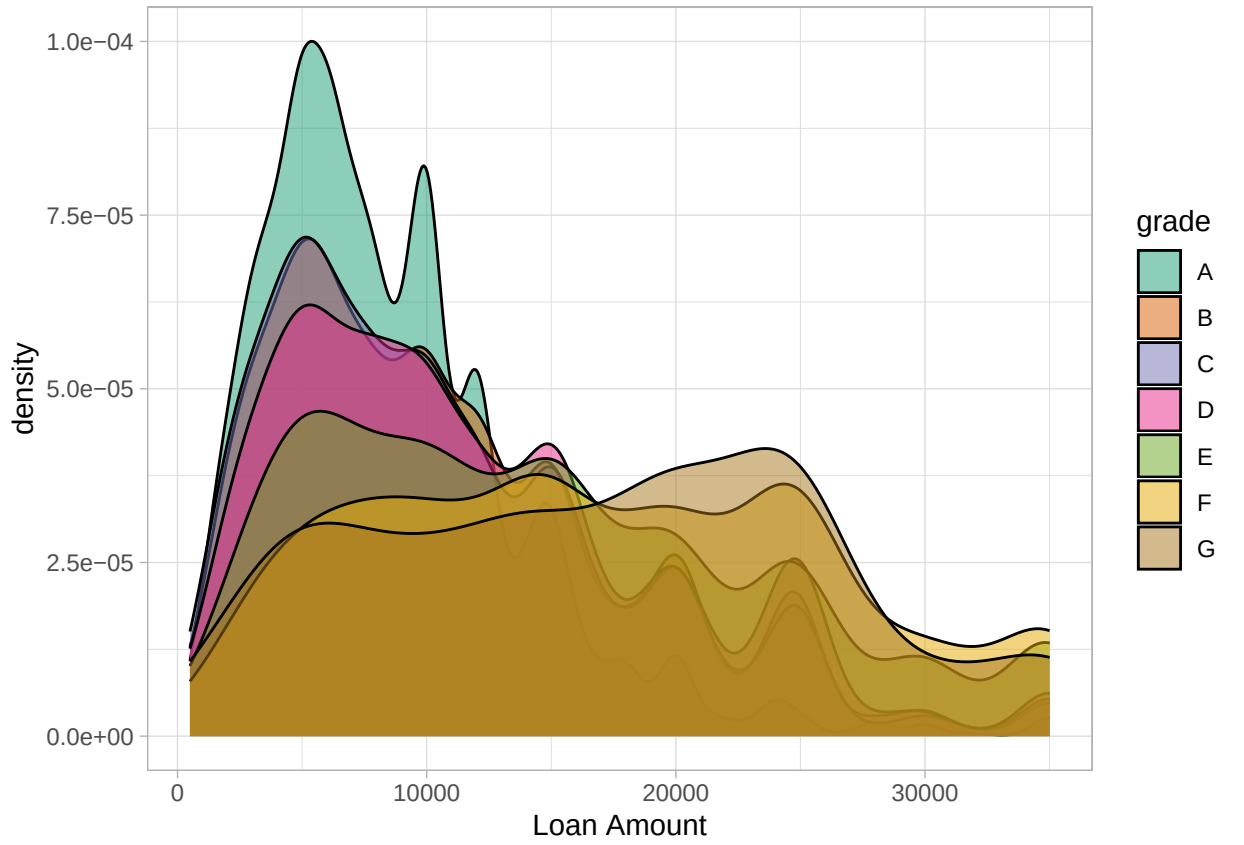
```
df <- read.csv("../data/LoanStats.csv")
ggplot(df, aes(x=loan_amnt)) + geom_density() + facet_wrap(~grade)
```

```
## Warning: Removed 7 rows containing non-finite values (‘stat_density()‘).
```



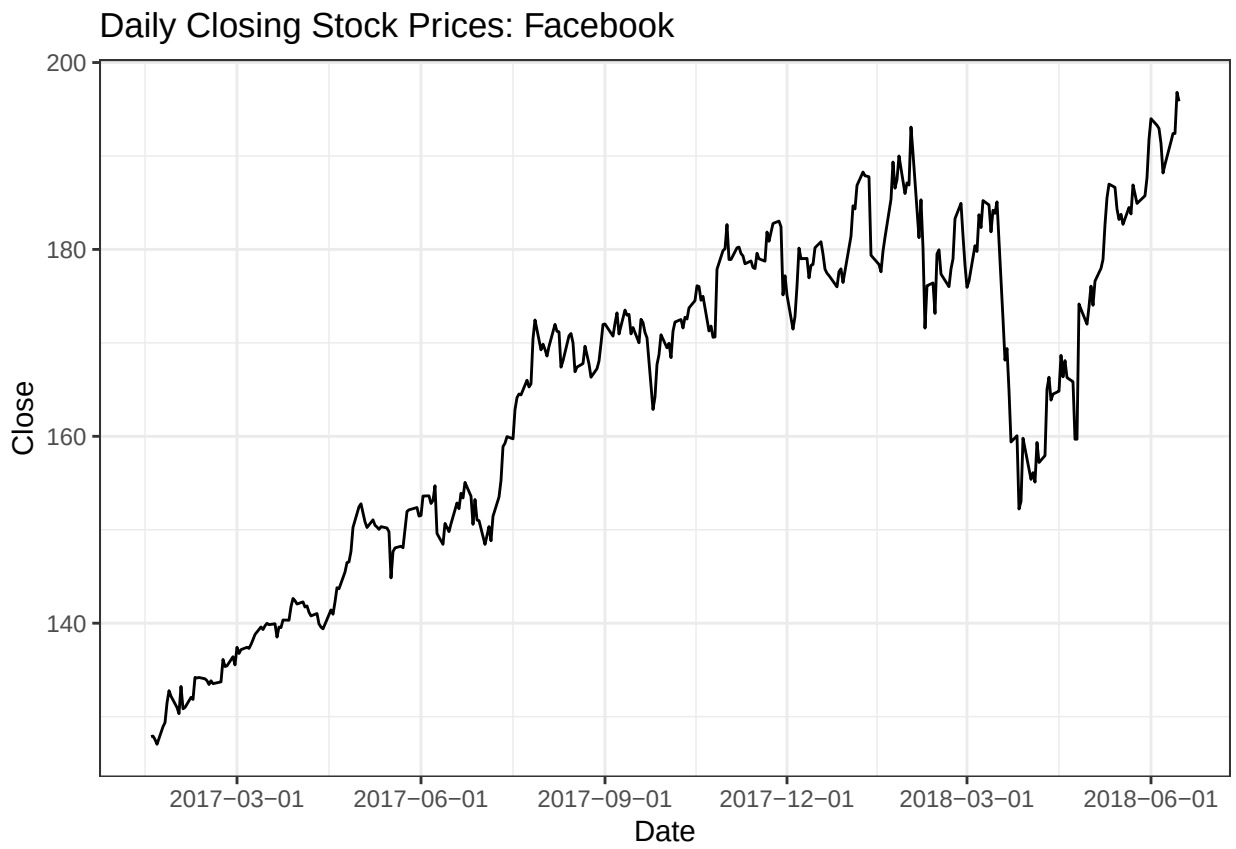

```
df <- read.csv("../data/LoanStats.csv")
ggplot(df, aes(x=loan_amnt)) + geom_density(aes(fill=grade), alpha=1/2) +
scale_fill_brewer(palette="Dark2") + xlab("Loan Amount") + theme_light()
```

```
## Warning: Removed 7 rows containing non-finite values (‘stat_density()‘).
```



Time series

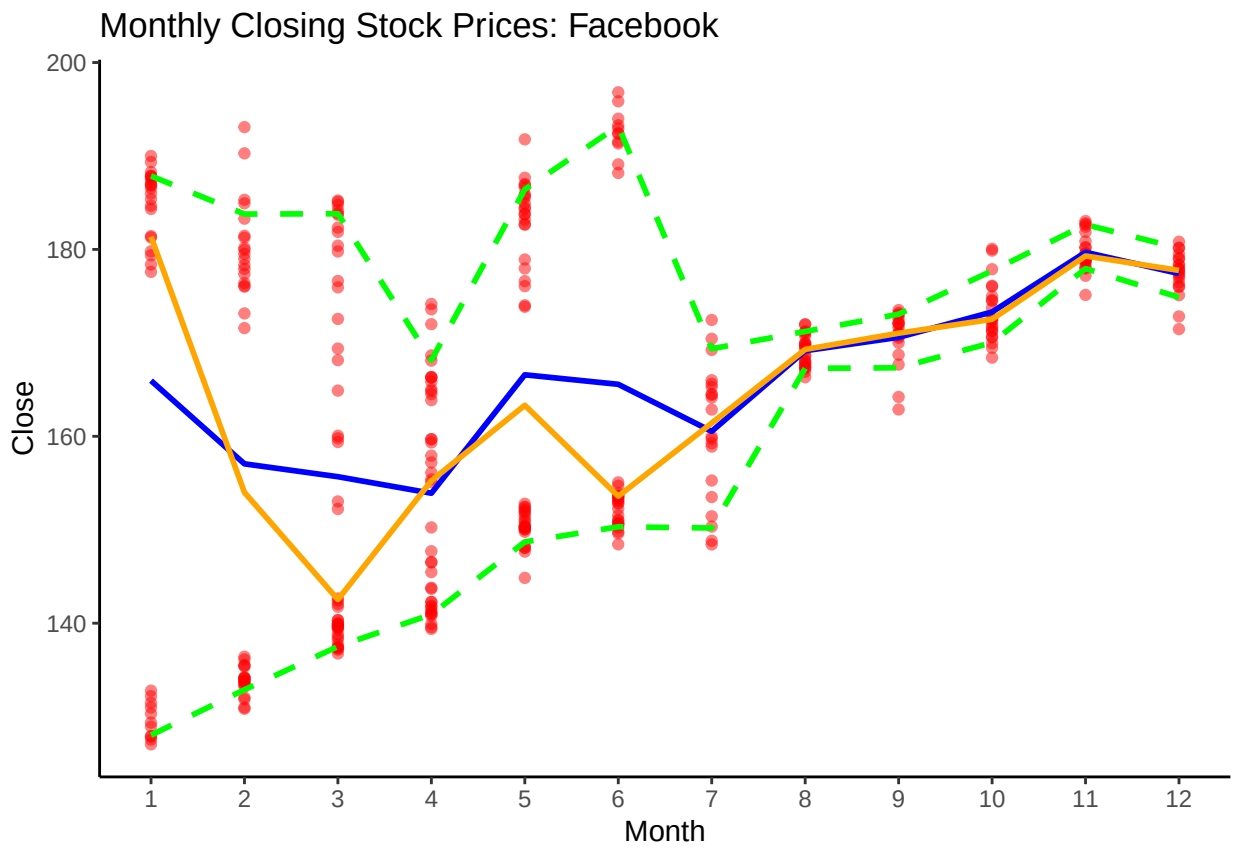
```
df_fb <- read.csv("../data/FB.csv")
df_fb$Date <- as.Date(df_fb$Date)
ggplot(df_fb, aes(x=Date, y=Close, group=1)) +
geom_line(color="black", na.rm=TRUE) +
ggtitle("Daily Closing Stock Prices: Facebook") +
theme(plot.title = element_text(lineheight=.7, face="bold")) +
scale_x_date(date_breaks='3 month') +
theme_bw()
```



Statistical summaries

```
df_fb <- read.csv("../data/FB.csv")
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=.1), linetype=2, color="green", size=1) +
stat_summary(geom="line", fun="quantile", fun.args=list(probs=.9), linetype=2, color="green", size=1) +
scale_x_continuous(breaks=seq(0, 13, 1)) +
ggtitle("Monthly Closing 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.
```



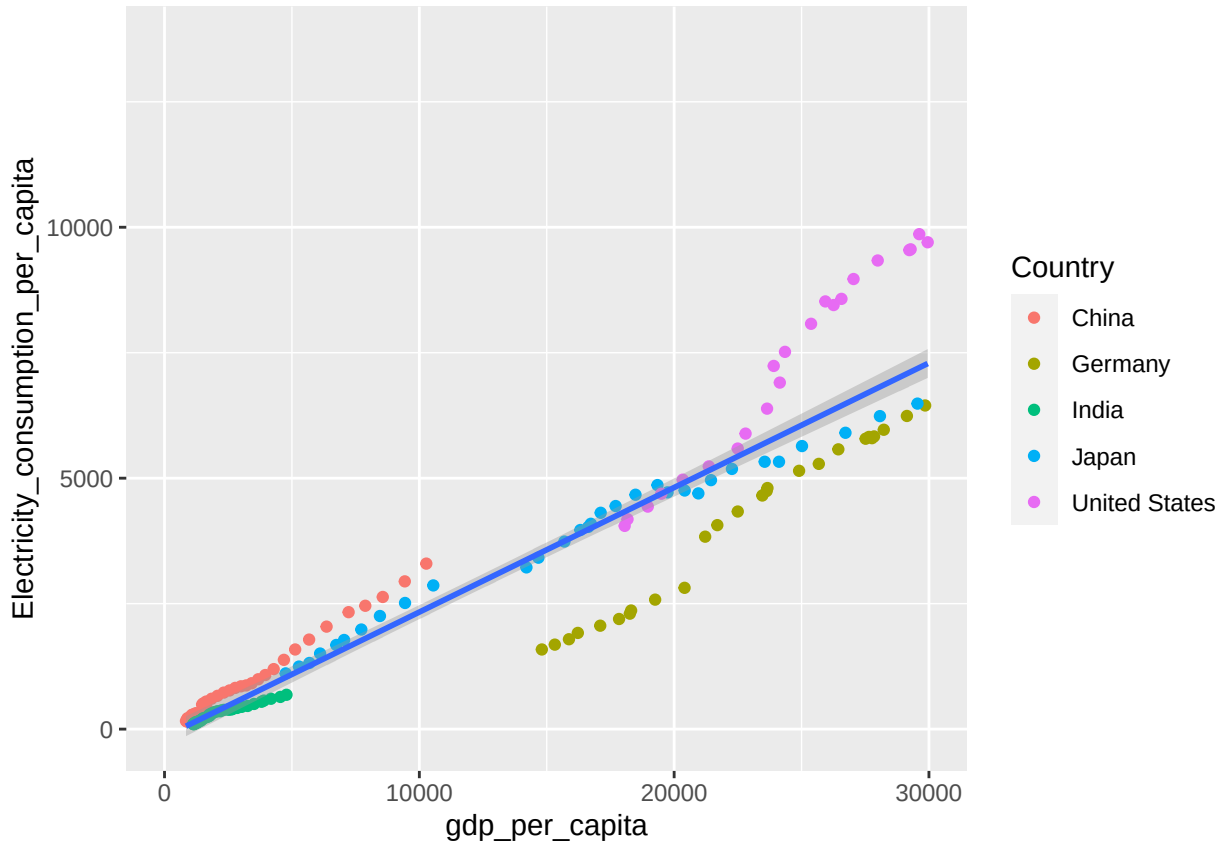
Linear regression

```
df <- read.csv("../data/gapminder-data.csv")
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) +
stat_smooth(method=lm)
```

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

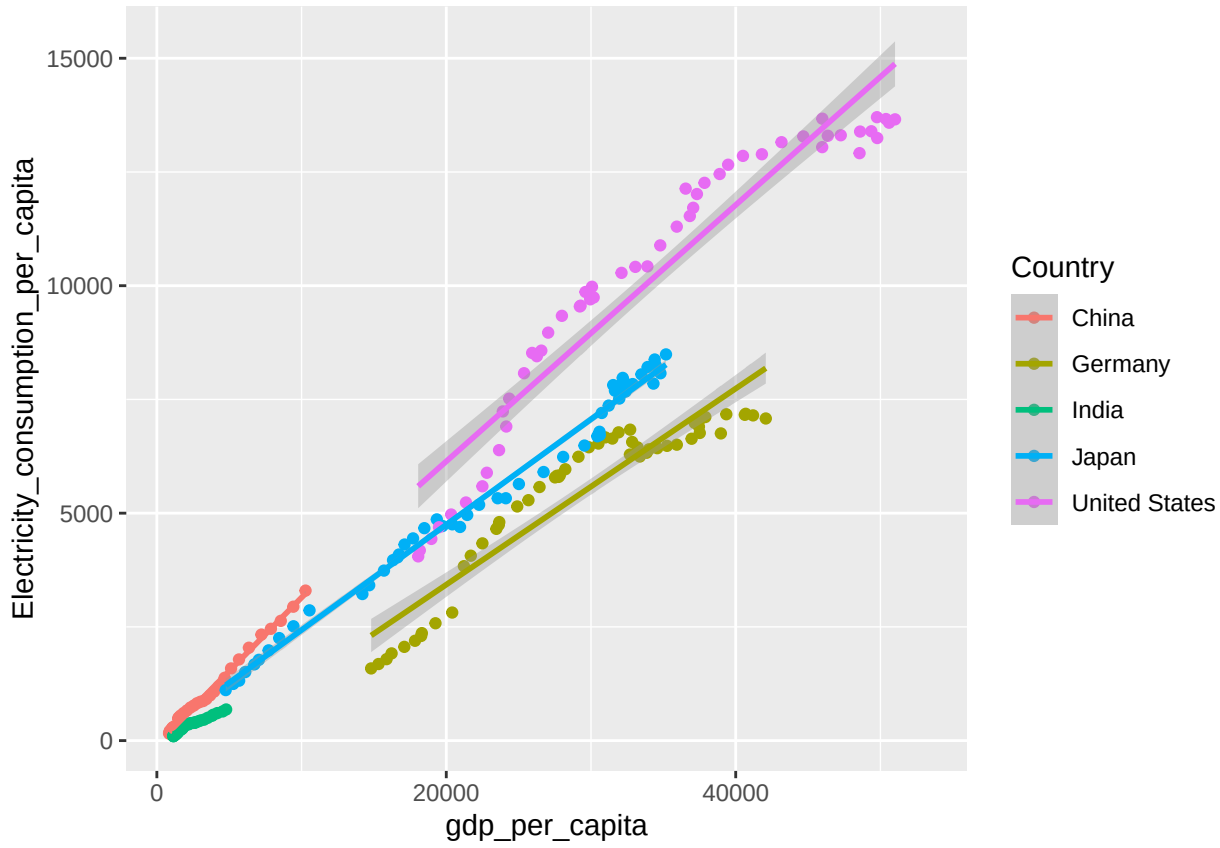
```
## Warning: Removed 919 rows containing non-finite values (`stat_smooth()`).
```

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



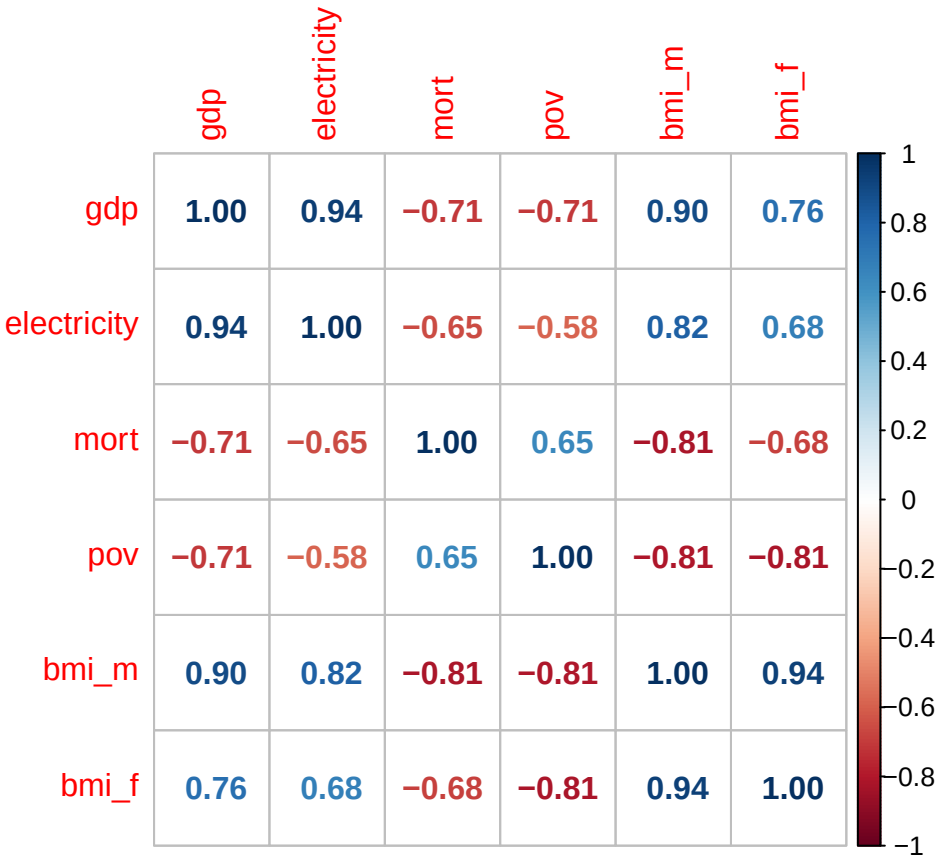

```
df <- read.csv("../data/gapminder-data.csv")
dfs <- subset(df, Country %in% c("Germany", "India", "China", "United States", "Japan"))
ggplot(dfs, aes(gdp_per_capita, Electricity_consumption_per_capita, color=Country)) +
geom_point() +
stat_smooth(method=lm)
```

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

## Warning: Removed 842 rows containing non-finite values (`stat_smooth()`).

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

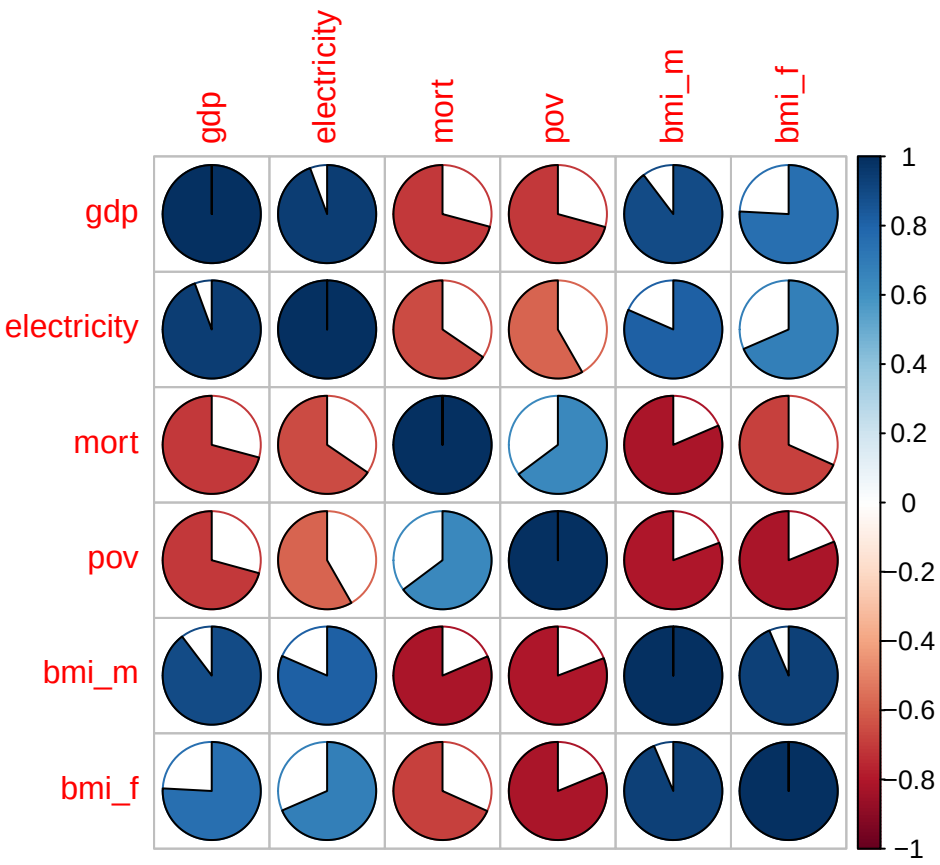


Correlations

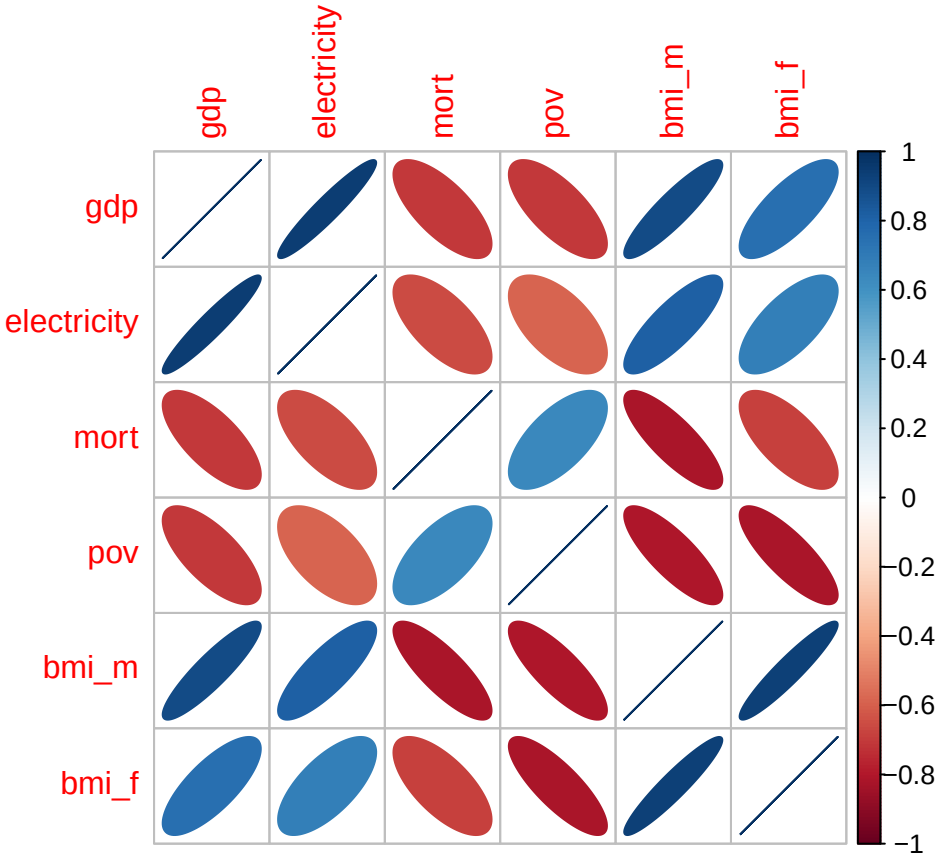
```
df <- read.csv("../data/gapminder-data.csv")
df <- df[, colnames(df)[4:9]]
df <- na.omit(df)
colnames(df) <- c("gdp", "electricity", "mort", "pov", "bmi_m", "bmi_f")
M <- cor(df)
corrplot(M, method="number")
```




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

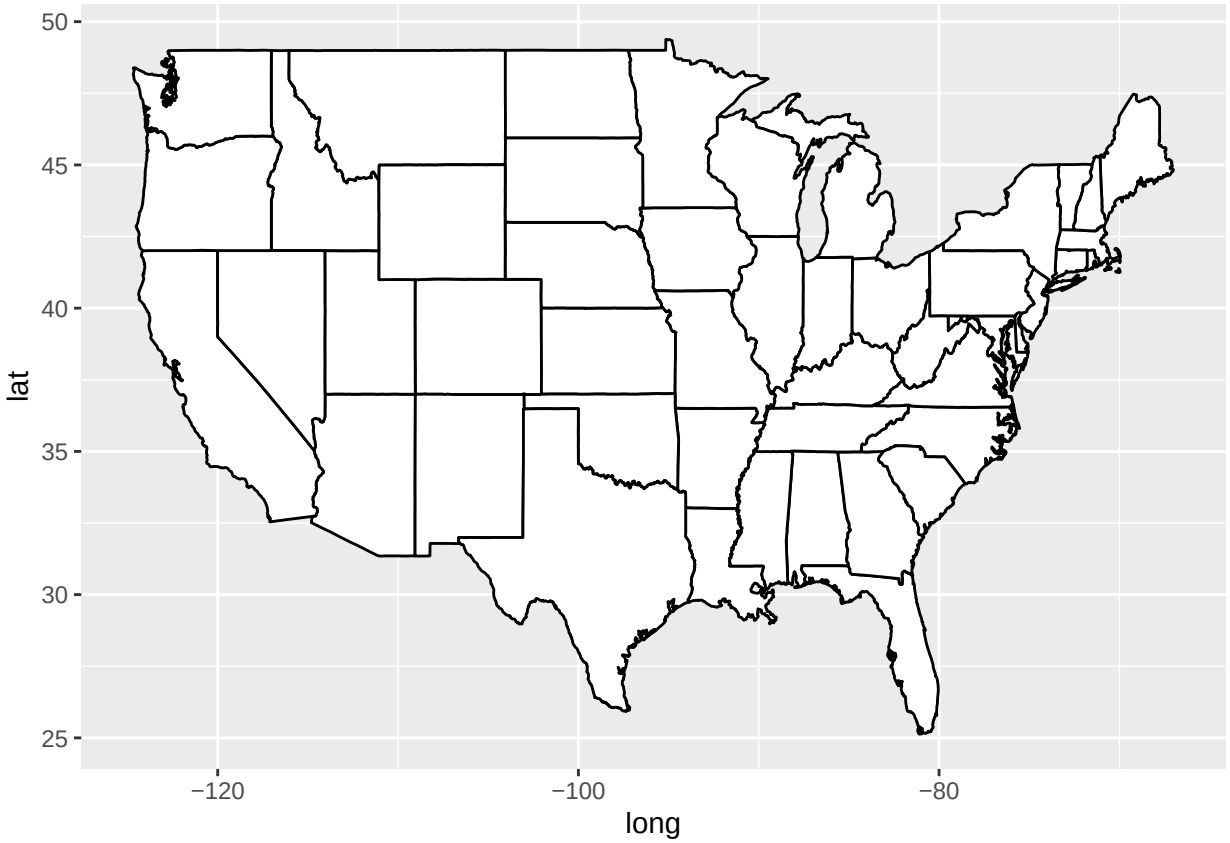



```
corrplot(M, method="ellipse")
```

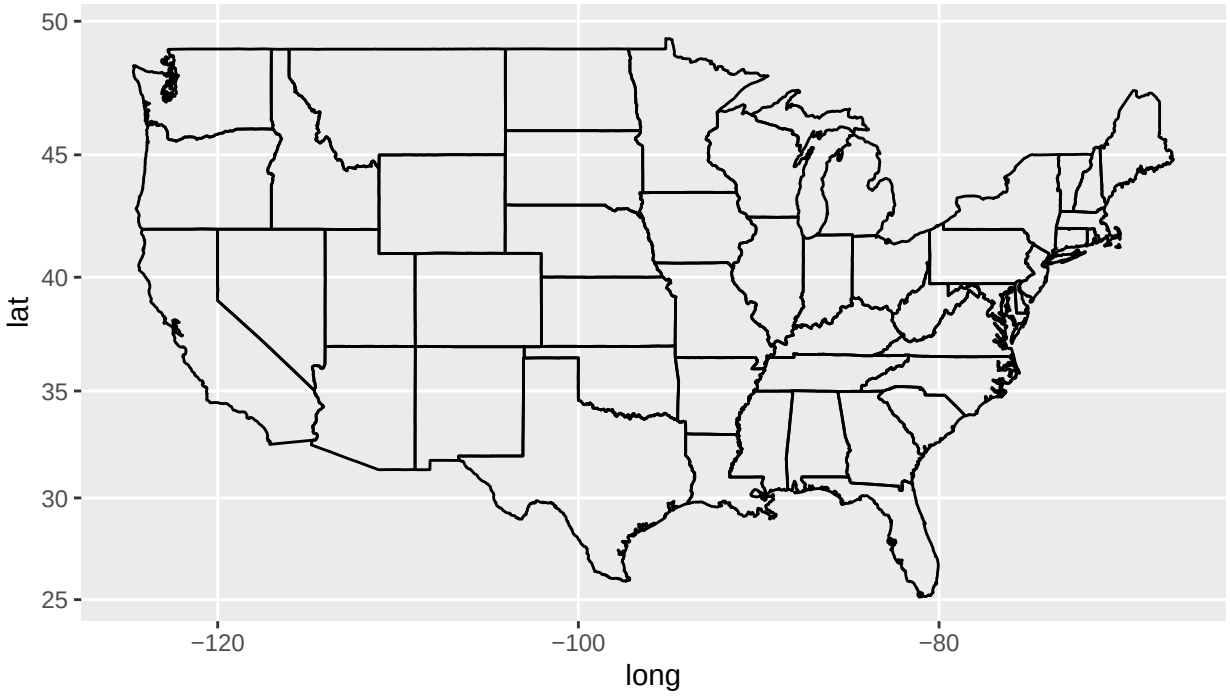


Maps

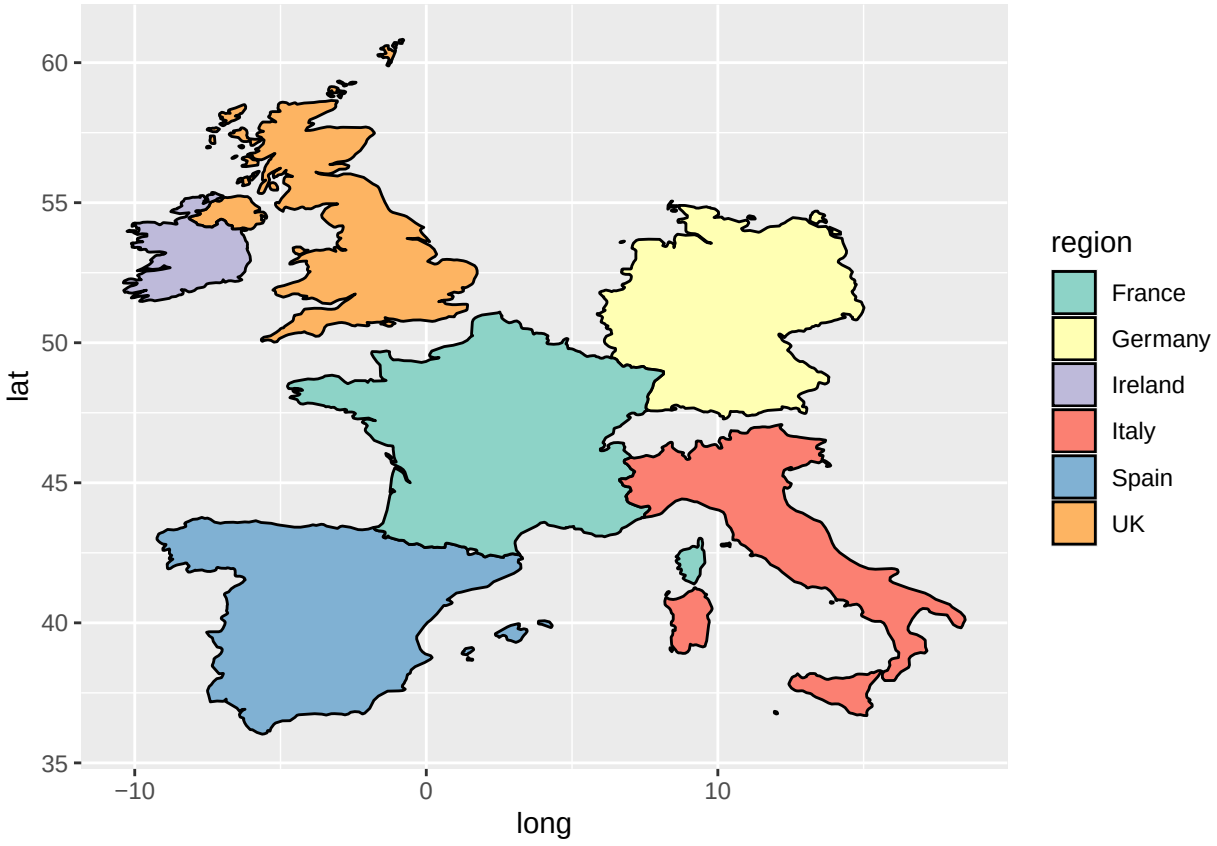
```
states_map <- map_data("state")
ggplot(states_map, aes(x=long, y=lat, group=group)) +    geom_polygon(fill="white", colour="black")
```




```
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")
```


60
55
50
45
40
35
lat
−10
0
10
long
region
France
Germany
Ireland
Italy
Spain
UK