

# Ggplot2 tutorial - Command lines Part 2

Ariane Ducellier

10/12/2023

Load R packages

```
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(ggpubr)
```

```
library(gridExtra)
```

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

```
library(Lock5Data)
```

## Part 2 - Grammar of Graphics and Visual Components

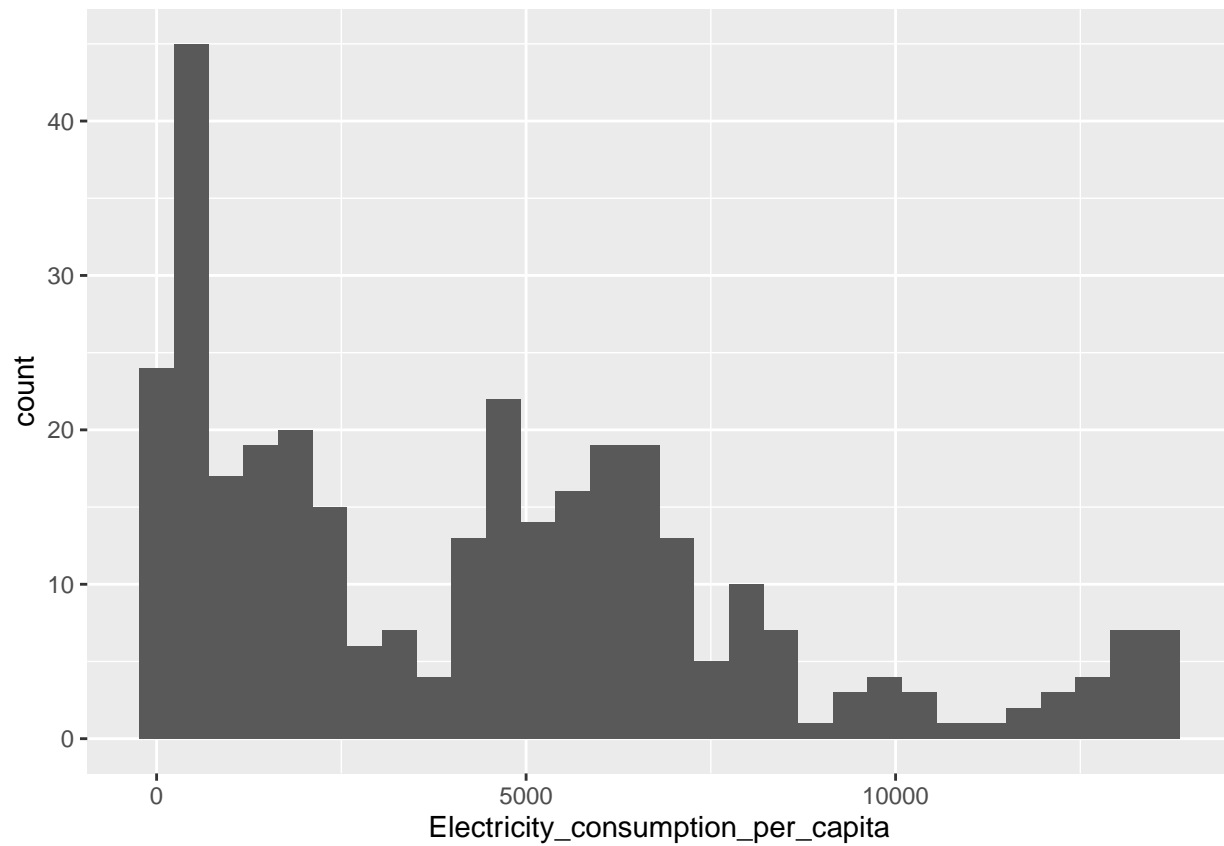
### Layers

```
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`
```

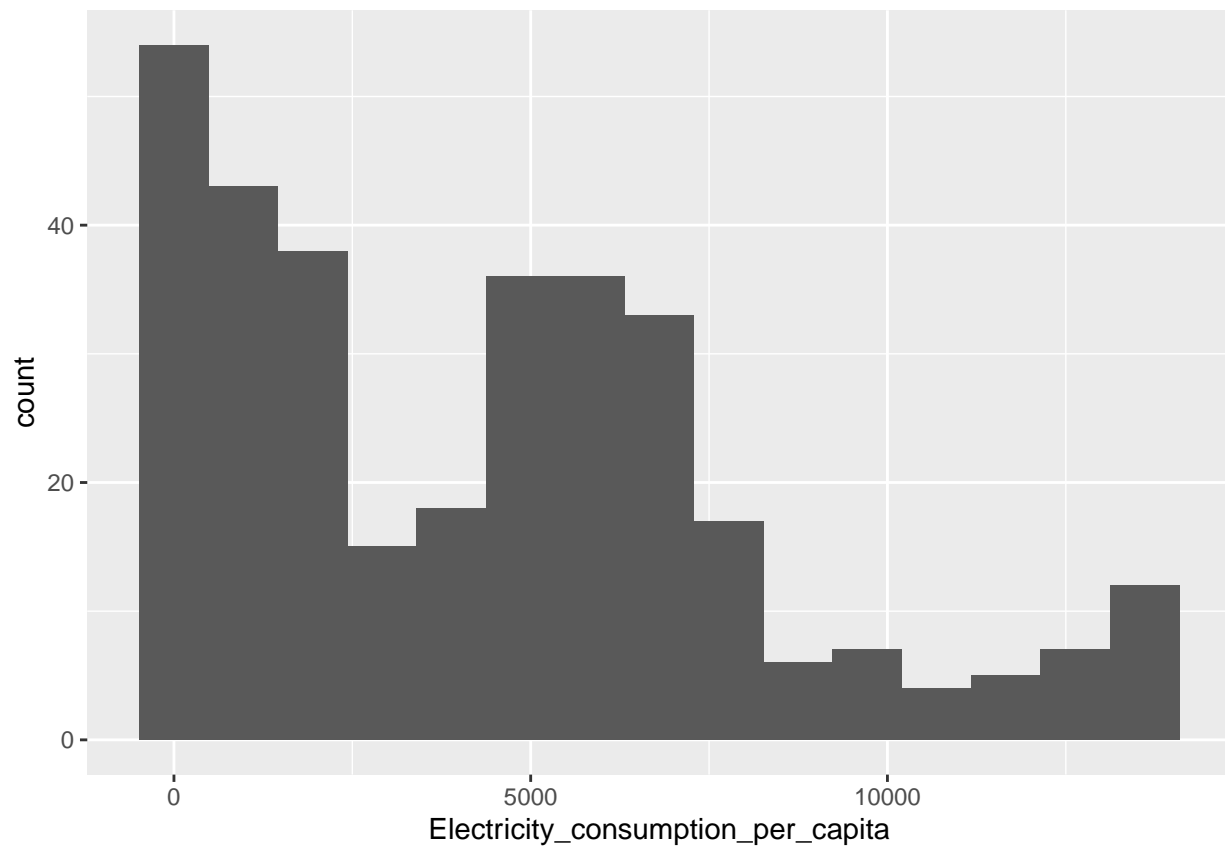
```
p1 <- ggplot(df, aes(x=Electricity_consumption_per_capita))
p2 <- p1 + geom_histogram()
p2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 1181 rows containing non-finite values (`stat_bin()`).
```



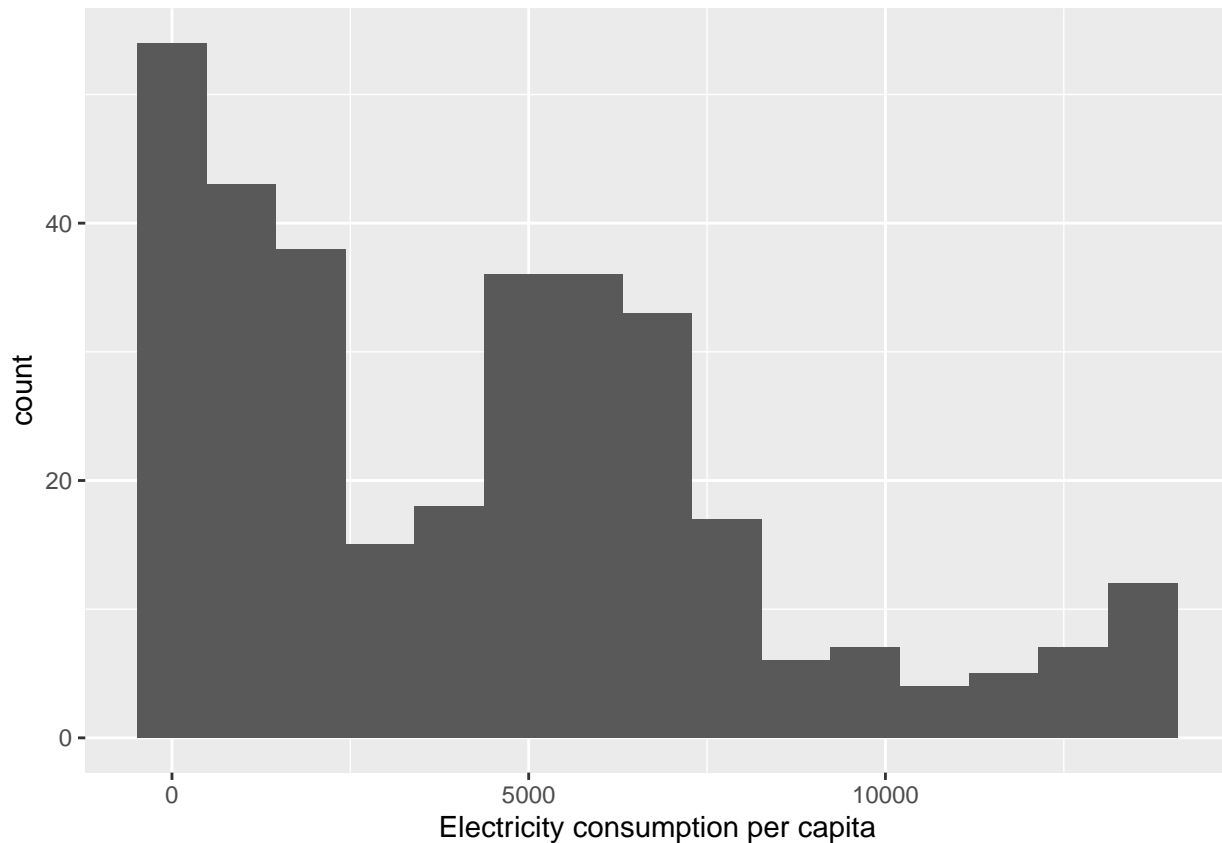
```
p3 <- p1 + geom_histogram(bins=15)
p3
```

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



```
p4 <- p3 + xlab("Electricity consumption per capita")  
p4
```

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



## Scales

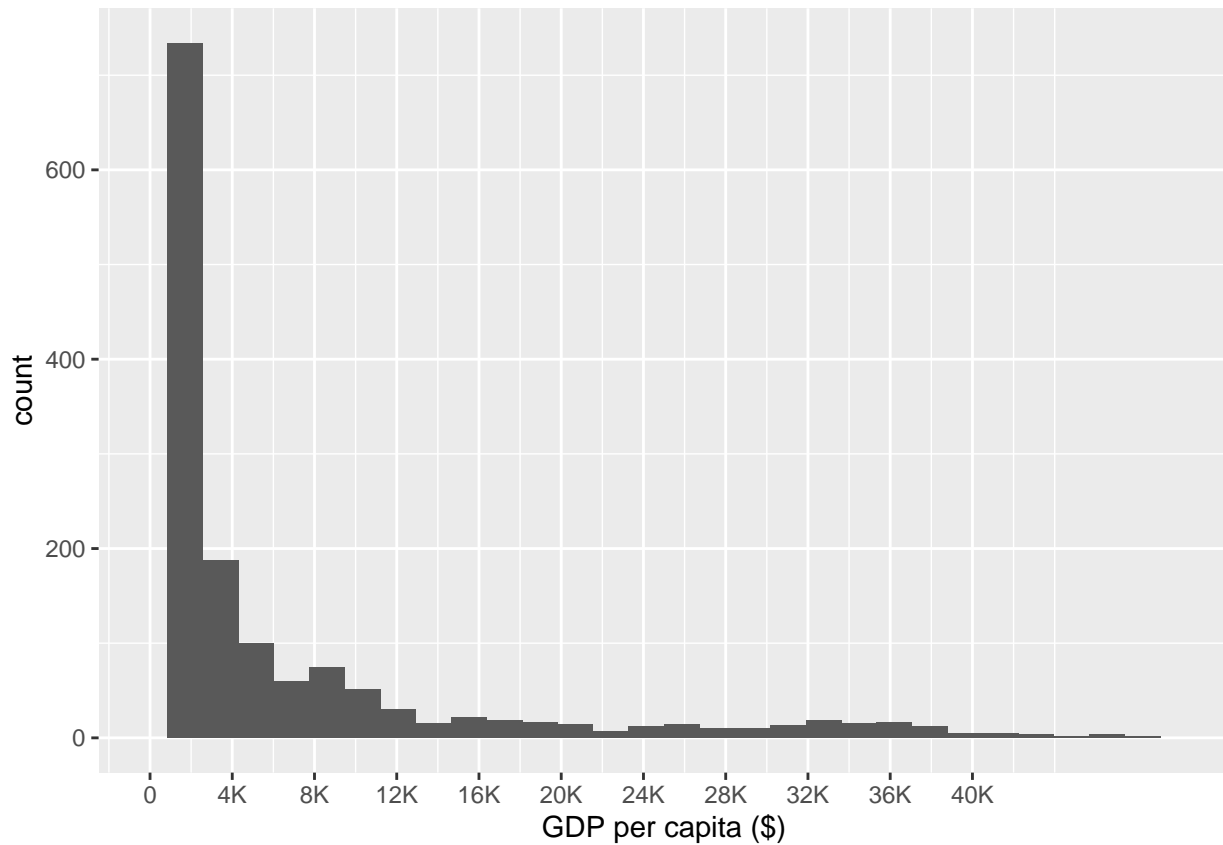
```
df <- read_csv("../data/gapminder-data.csv")

## 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`

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

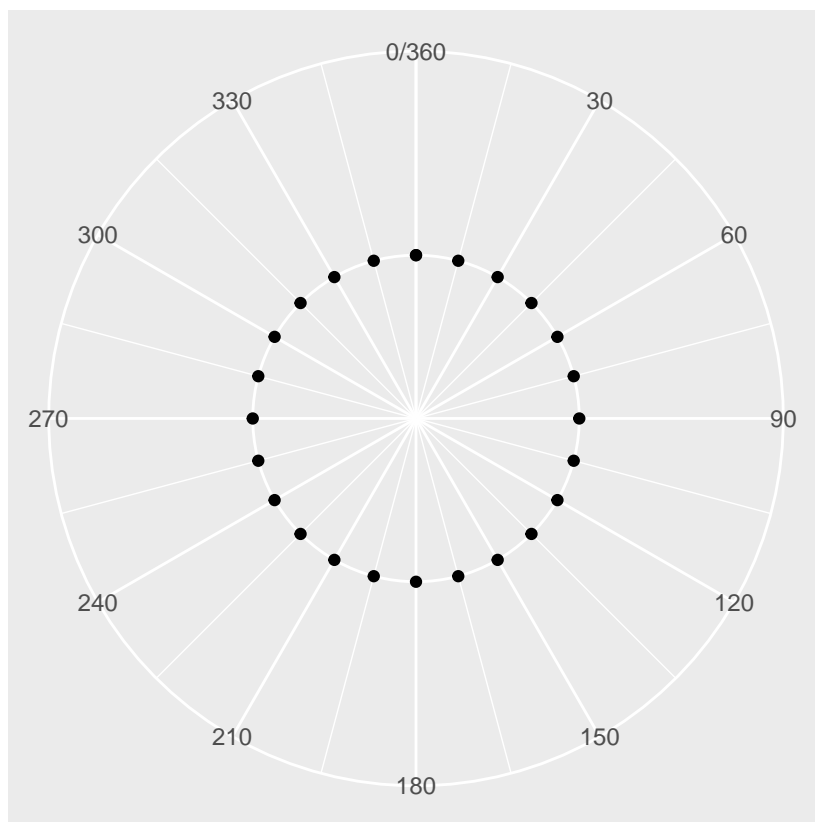
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 7 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



## 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)) +
  theme(axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank())
```

```
## 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.
```



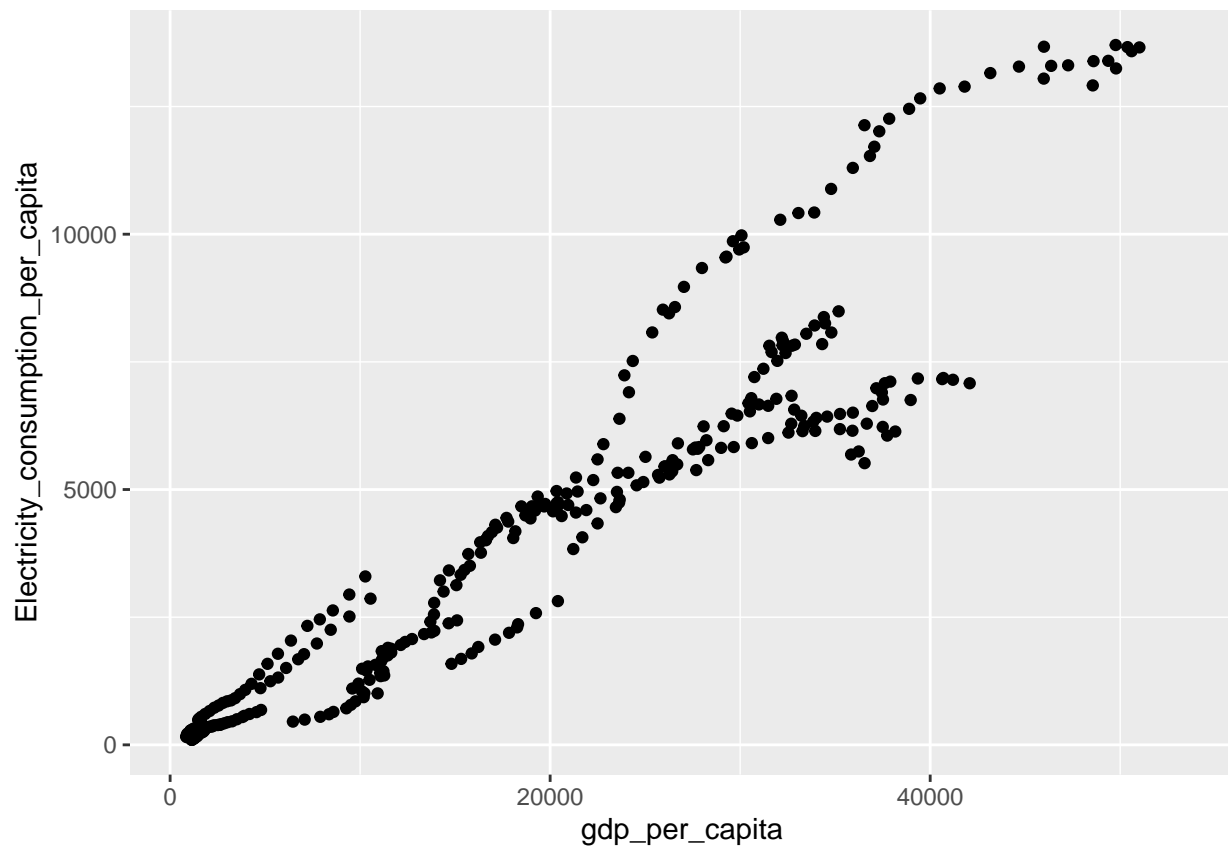
## Facets

```
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`
```

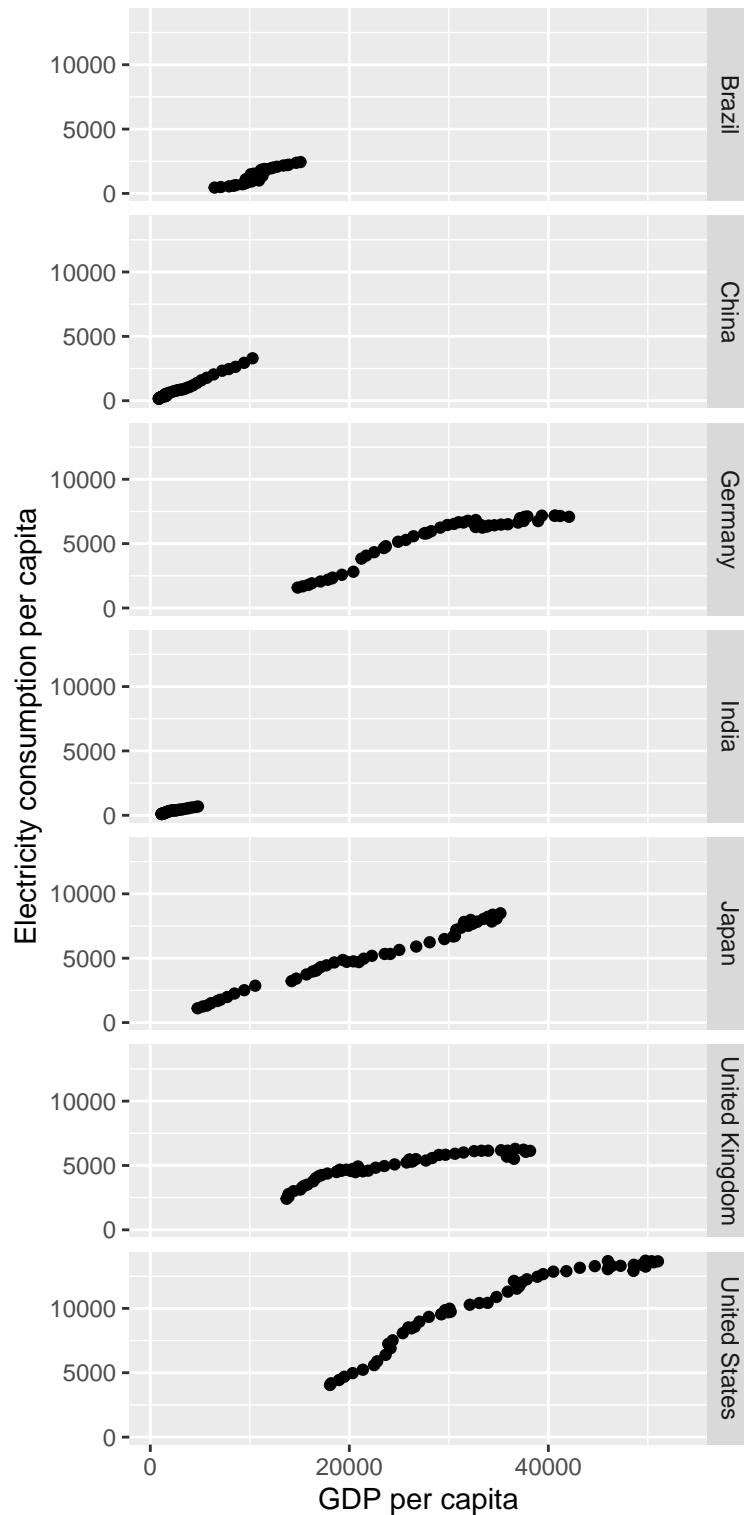
```
p <- ggplot(df, aes(x=gdp_per_capita, y=Electricity_consumption_per_capita)) + geom_point()
p
```

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



```
p + facet_grid(factor(Country, stringr::str_sort(unique(Country))) ~ .) +
  xlab("GDP per capita") +
  ylab("Electricity consumption per capita")
```

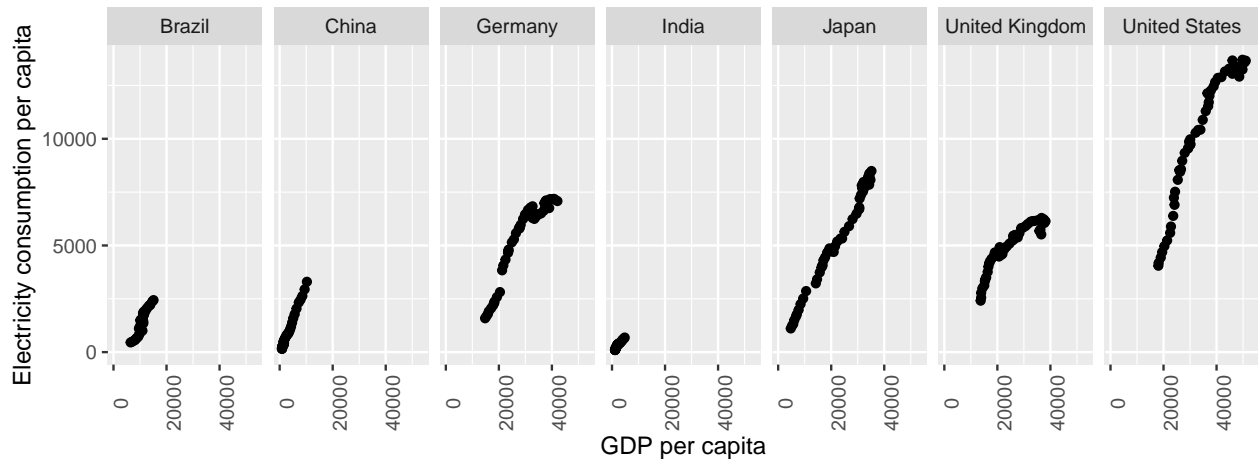
```
## Warning: Removed 1181 rows containing missing values (`geom_point()`).
```



```
p + facet_grid(. ~ factor(Country, stringr::str_sort(unique(Country)))) +
  xlab("GDP per capita") +
  ylab("Electricity consumption per capita") +
  theme(axis.text.x=element_text(angle=90))
```

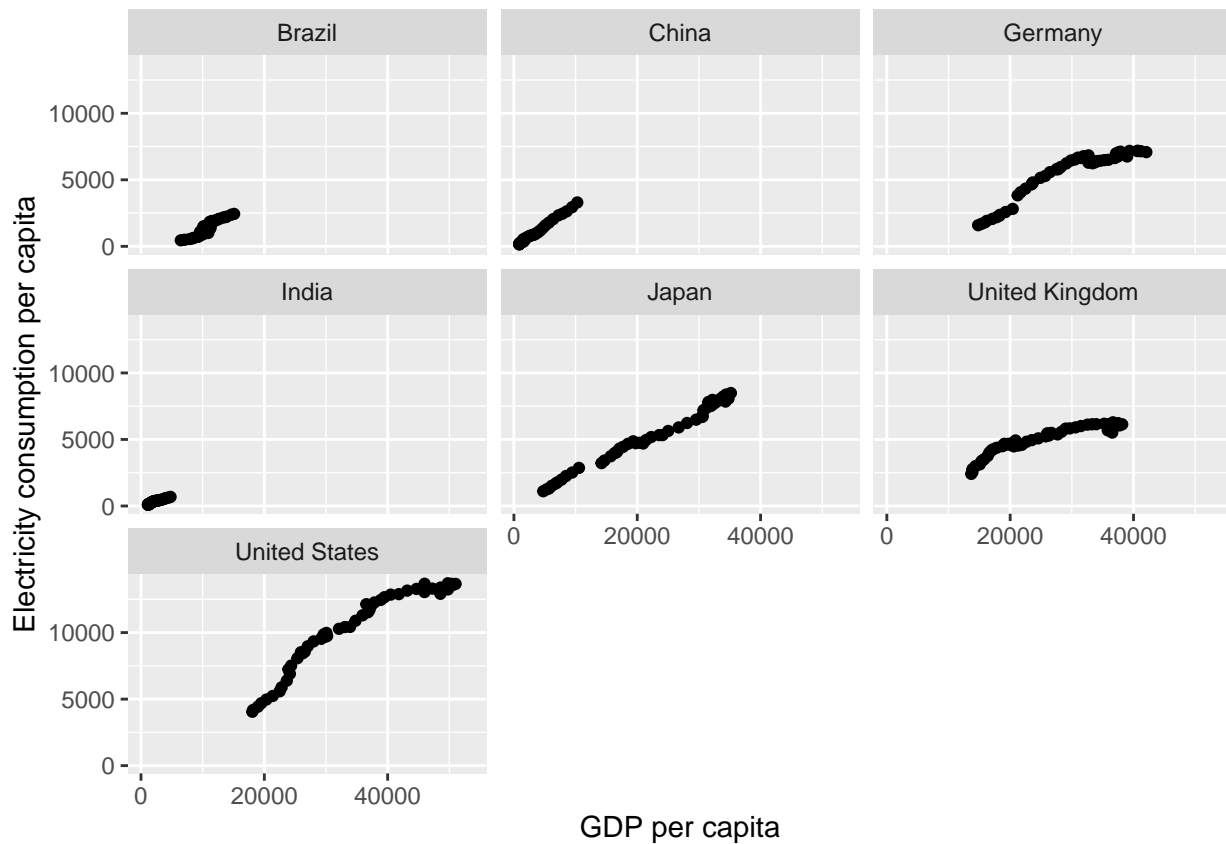
```
## Warning: Removed 1181 rows containing missing values (`geom_point()`).
```





```
p + facet_wrap(~Country) +  
  xlab("GDP per capita") +  
  ylab("Electricity consumption per capita")
```

## 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"
ggplot(df, aes_string(x=var1, y=var2)) +
  geom_point(color=2, shape=2) +
  xlim(0, 10000) +
  xlab(name1) +
  ylab(name2)

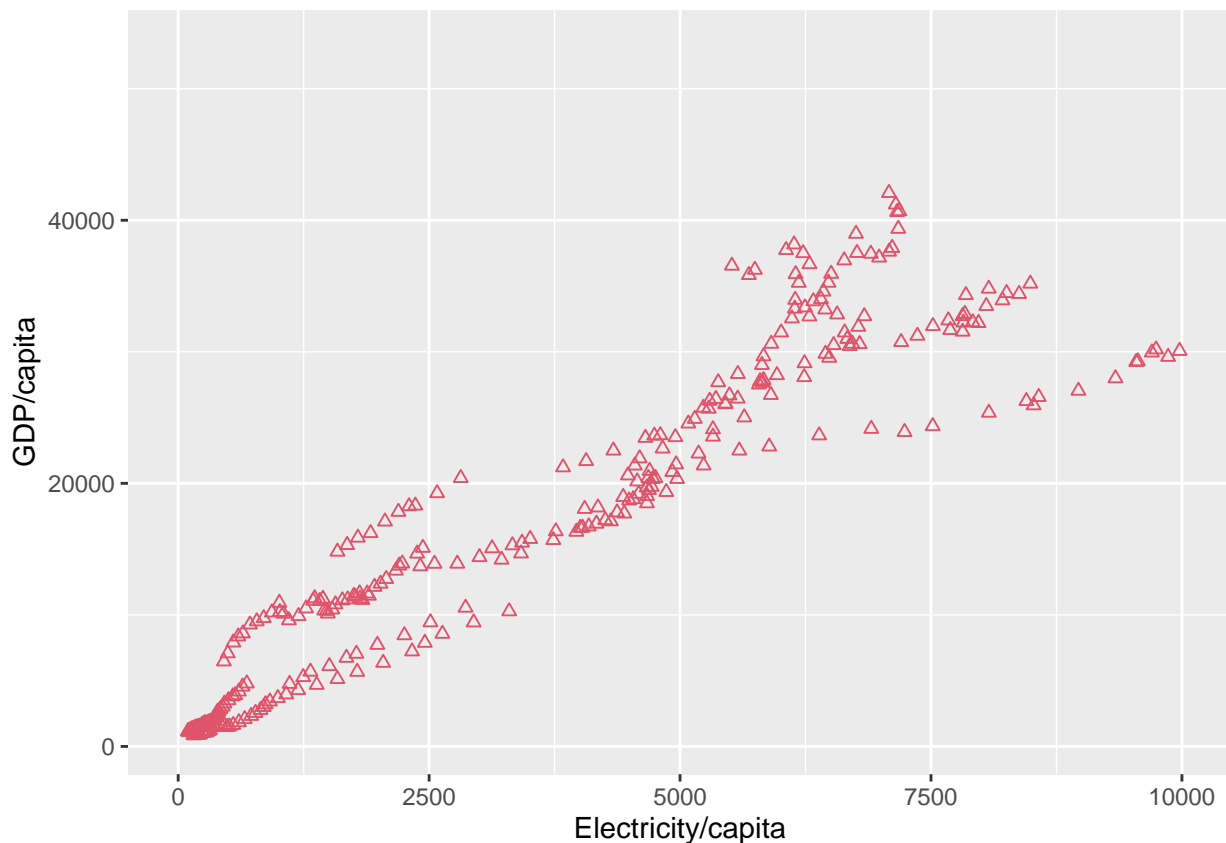
```

```

## 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.

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

```



```

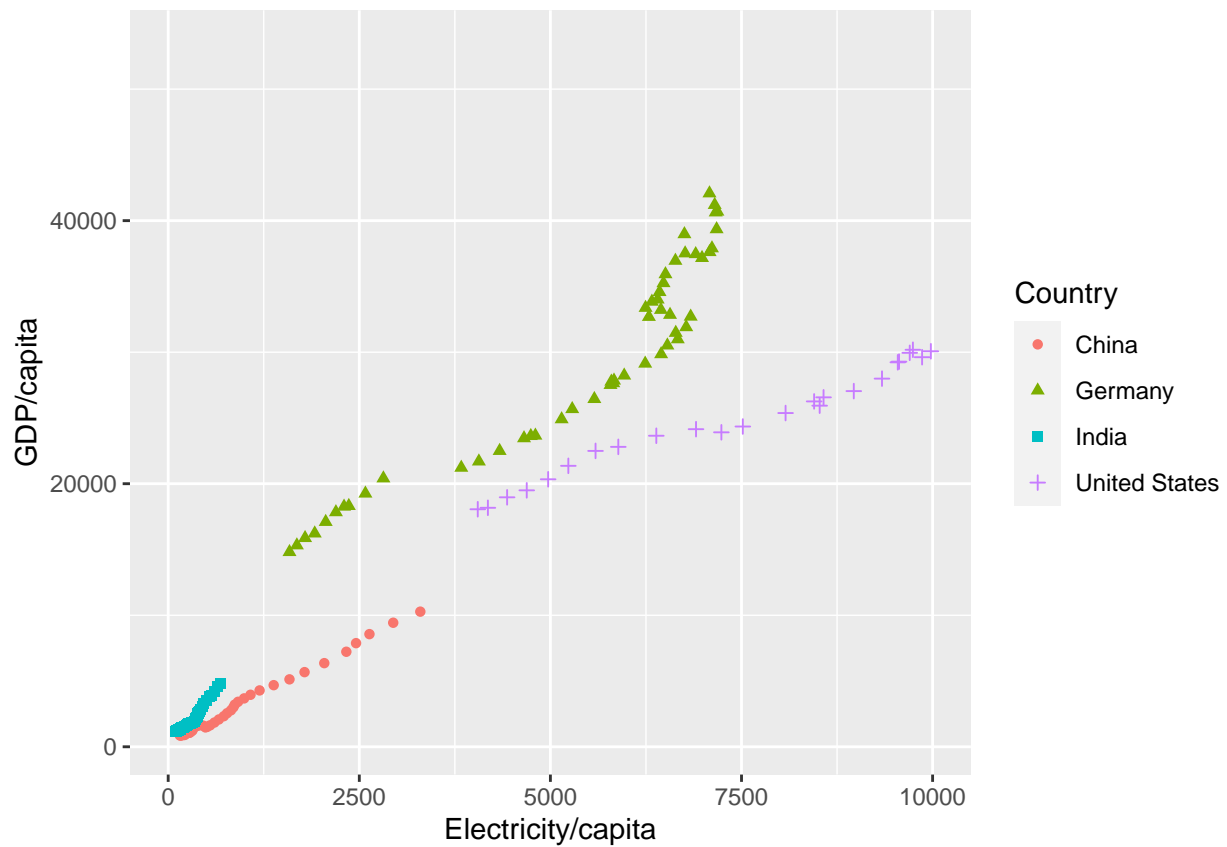
ggplot(dfs, aes_string(x=var1, y=var2)) +
  geom_point(aes(color=Country, shape=Country)) +
  xlim(0, 10000) +
  xlab(name1) +
  ylab(name2)

```

```

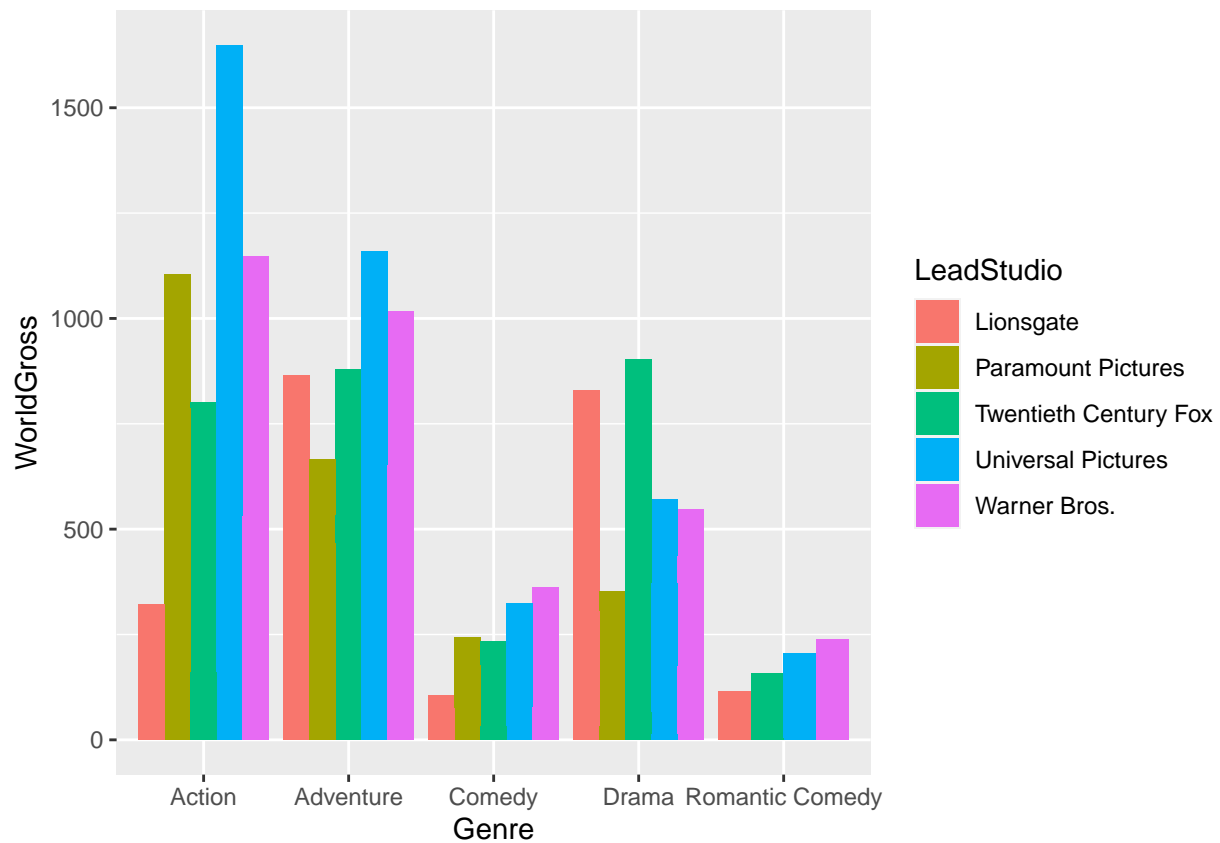
## Warning: Removed 706 rows containing missing values (`geom_point()`).

```



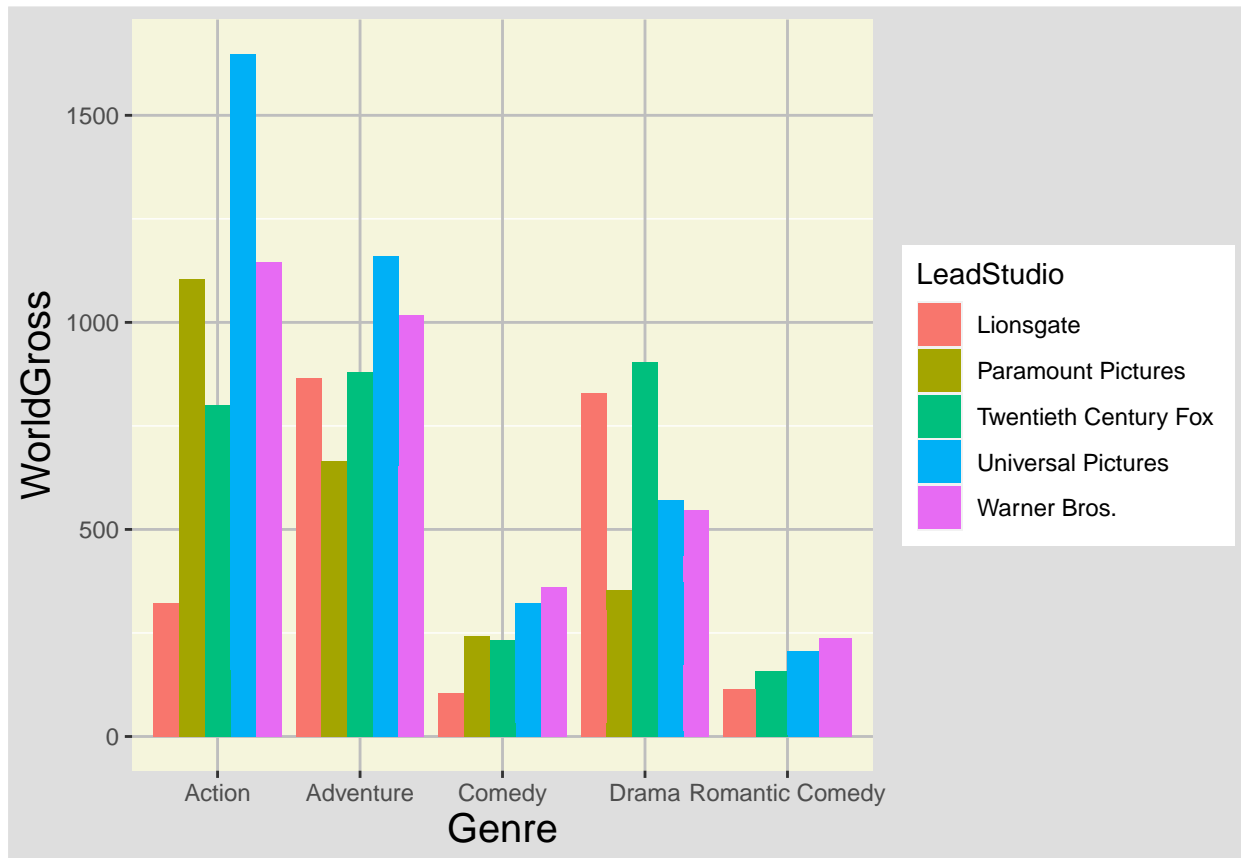
## Themes

```
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")
p2
```

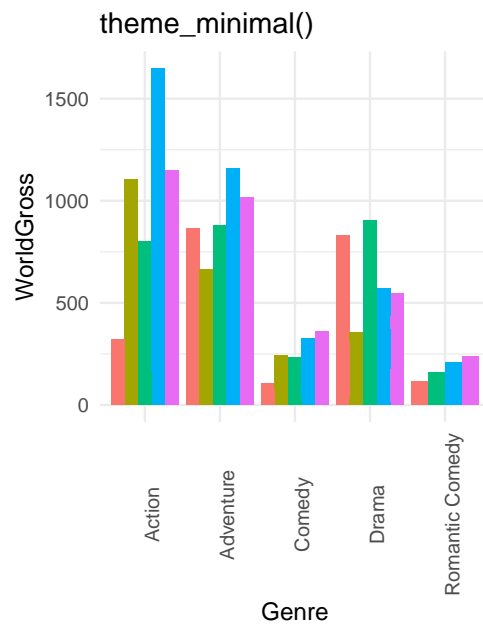
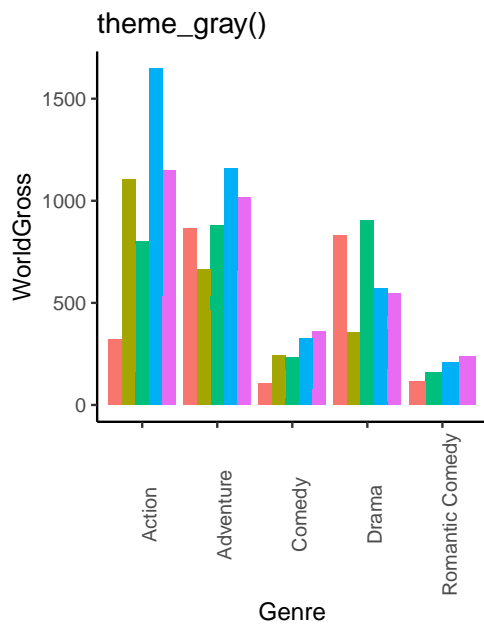
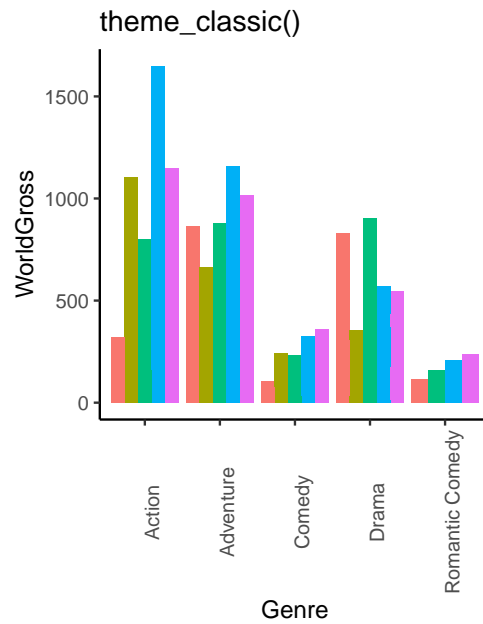
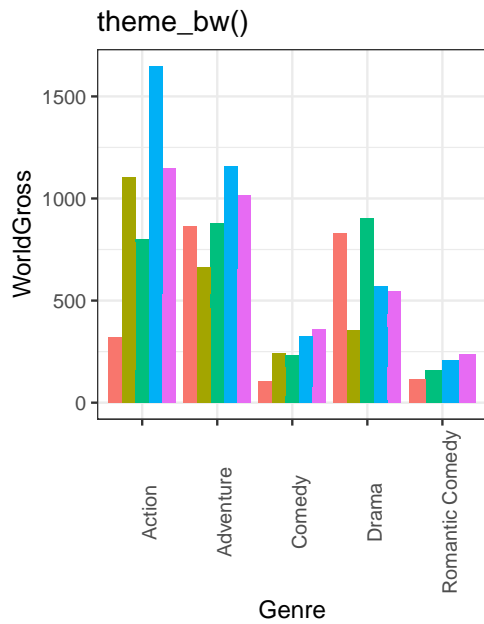


```
p3 <- p2 + theme(axis.title.x=element_text(size=15),
  axis.title.y=element_text(size=15),
  plot.background=element_rect(fill="gray87"),
  panel.background=element_rect(fill="beige"),
  panel.grid.major=element_line(color="Gray", linetype=1))
```

p3



```
p4 <- p2 +
  theme_bw() +
  ggtitle("theme_bw()") +
  theme(axis.text.x=element_text(angle=90))
p5 <- p2 +
  theme_classic() +
  ggtitle("theme_classic()") +
  theme(axis.text.x=element_text(angle=90))
p6 <- p2 +
  theme_classic() +
  ggtitle("theme_gray()") +
  theme(axis.text.x=element_text(angle=90))
p7 <- p2 +
  theme_minimal() +
  ggtitle("theme_minimal()") +
  theme(axis.text.x=element_text(angle=90))
ggarrange(p4, p5, p6, p7, ncol=2, nrow=2, common.legend = TRUE, legend="right")
```

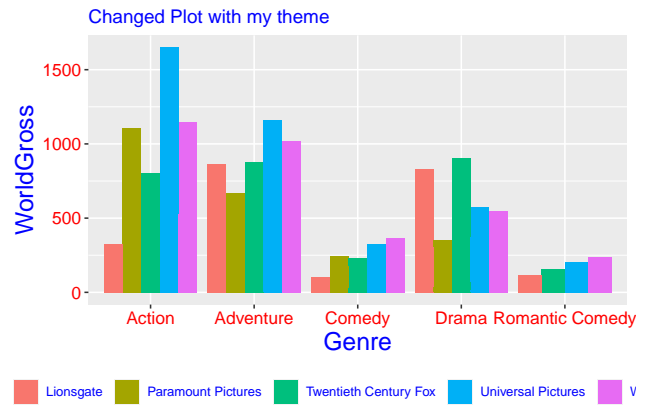
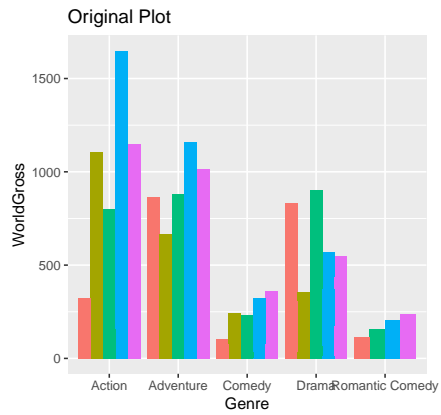


LeadStudio

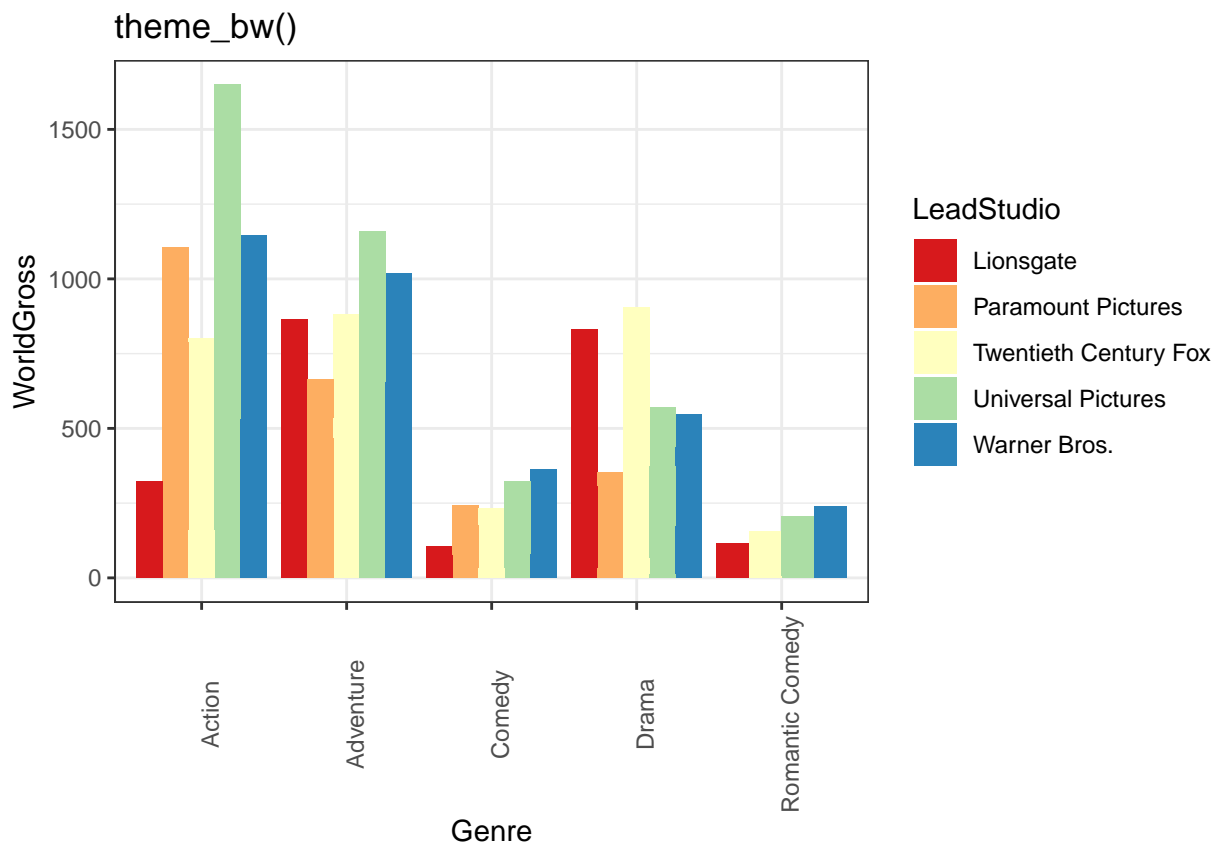
- Lionsgate
- Paramount Pictures
- Twentieth Century Fox
- Universal Pictures
- Warner Bros.

```
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)))
```

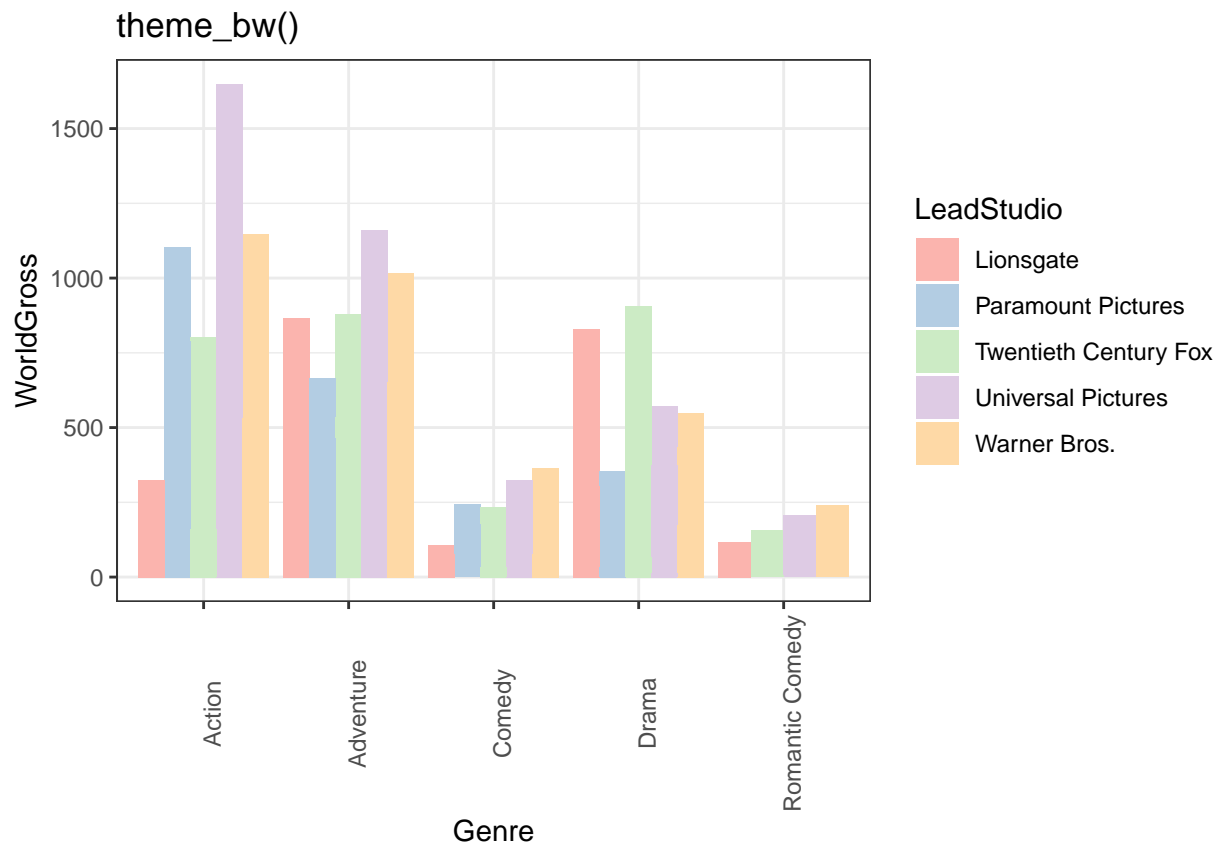
```
p2 <- p2 +
  ggtitle("Original Plot")
p8 <- p2 +
  mytheme +
  ggtitle("Changed Plot with my theme")
grid.arrange(p2, p8, ncol=2)
```



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



```
p4 + scale_fill_brewer(palette="Pastell1")
```



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



