For Beginners

r_ladies_helsinki_ggplot2 - RStudio

```
#this is R-Ladies Helsinki
A <- 10
a <- 3

myNumbers <- c(1:10)
myNumbers
rep(myNumbers, times = 3)

```

```
> #this is R-Ladies Helsinki
> A <- 10
> a <- 3
> a
[1] 3
> a
[1] 3
> A
[1] 10
>
```

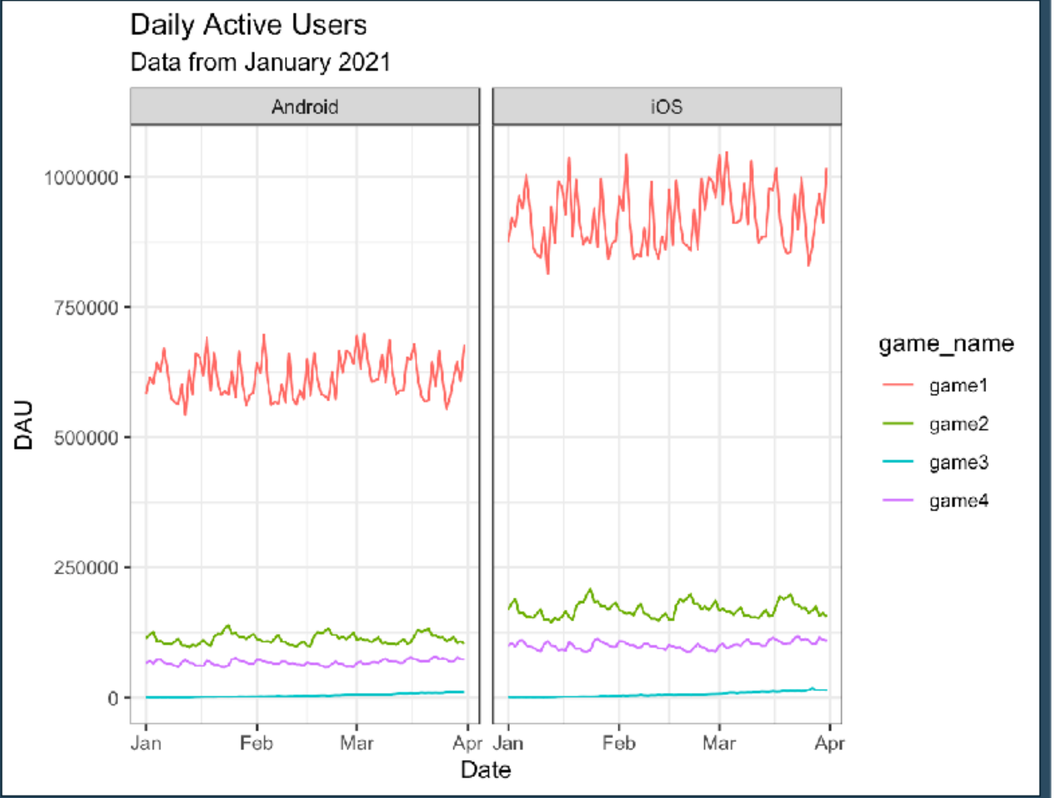
Environment | History | Connections | Tutorial

Data

game_data	720 obs. of 6 variables

Values

a	3
A	10



Some basics about R!

R commands;

- are case sensitive
- can be separated either by a semi-colon (‘;’), or by a newline
- #comment

Assignment, Basic Operators

Assignments

- use **<-**

Basic arithmetic operators

- +,-,*,/,^, %%

Logical operators

- <, >, <=, >=, ==, !=, !x, x & y, x | y

Others

- sum, sqrt, min, max, mean, var, sd, abs, summary

Tidyverse

```
install.packages("tidyverse")

library(tidyverse)
```

Loading today's data set!

```
install.packages("tidyverse")

library(tidyverse)
```

Option 1

```
game_data <- read_csv("https://tinyurl.com/3a5hxnra")
```

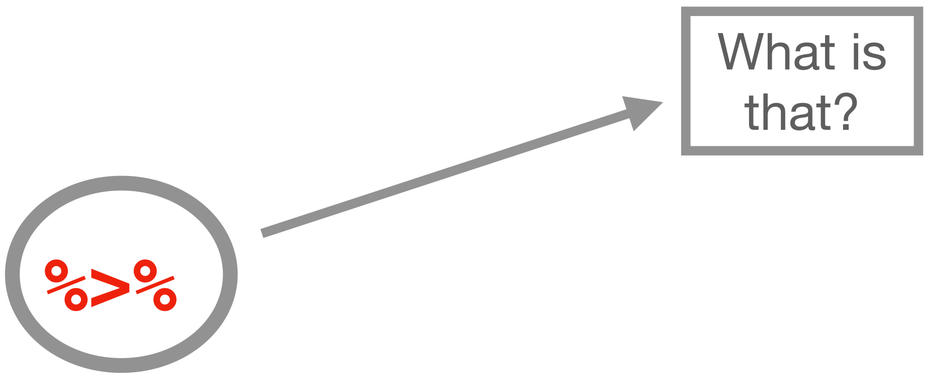
Get familiar today's data set!

Option 1

```
glimpse(game_data)
```

Option 2

```
game_data %>%
        View()
```



If you have recent version of dplyr, you'll see this:

|>

instead of

%>%

Pipe operator %>% or |>

- Chaining multiple functions

- Giving an input to function —> using it's output as another input to another function

- cmd + shift + m

- kntr + shft + m

Get familiar today's data set!

Option 1

```
glimpse(game_data)
```

```
> glimpse(game_data)
Rows: 720
Columns: 6
$ date              <date> 2021-01-01, 2021-01-01, 2021-01-01, 2021-01-…
$ game_name         <chr> "game2", "game4", "game3", "game1", "game2", …
$ genre             <chr> "puzzle", "rpg", "rpg", "puzzle", "puzzle", "…
$ daily_active_users <dbl> 169546, 99376, 676, 875956, 181696, 105046, 5…
$ in_app_revenue    <dbl> 409.183, 252.451, 2.109, 1984.495, 420.468, 2…
$ platform          <chr> "iOS", "iOS", "iOS", "iOS", "iOS", "iOS", "iO…
```

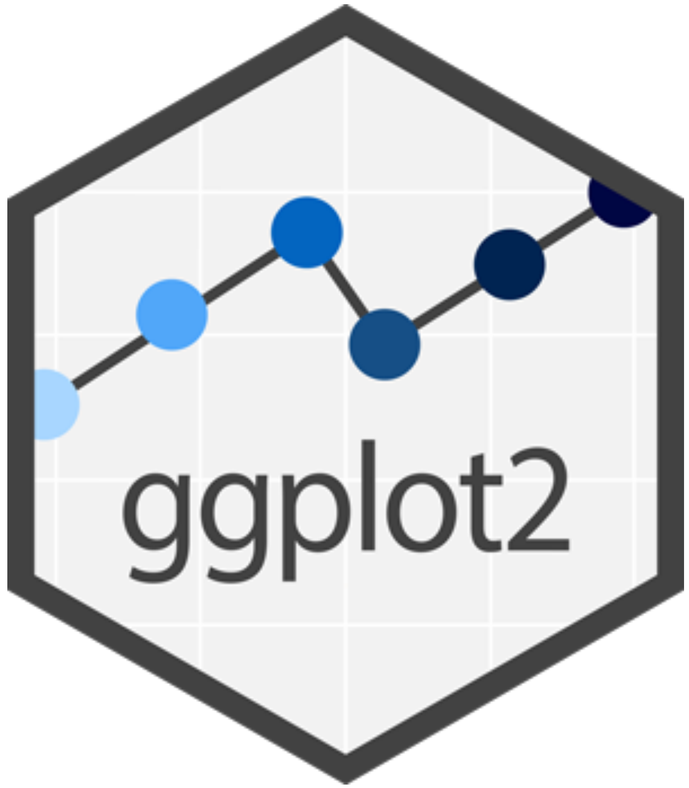
Get familiar today's data set!

date	game_name	genre	daily_active_users	in_app_revenue	platform
2021-01-01	game2	puzzle	169546	409.183	iOS
2021-01-01	game4	rpg	99376	252.451	iOS
2021-01-01	game3	rpg	676	2.109	iOS
2021-01-01	game1	puzzle	875956	1984.495	iOS
2021-01-01	game2	puzzle	113168	619.304	Android
2021-01-01	game4	rpg	66388	382.088	Android
2021-01-01	game3	rpg	588	3.192	Android
2021-01-01	game1	puzzle	584108	3003.560	Android
2021-01-02	game2	puzzle	181696	420.468	iOS
2021-01-02	game4	rpg	105046	265.697	iOS
2021-01-02	game3	rpg	586	2.553	iOS
2021-01-02	game1	puzzle	922306	2100.231	iOS
2021-01-02	game2	puzzle	121268	636.384	Android
2021-01-02	game4	rpg	70168	402.136	Android
2021-01-02	game3	rpg	528	3.864	Android
2021-01-02	game1	puzzle	615008	3178.728	Android

```
game_data %>%
  View()
```

ggplot2

- A system to create visualisations based on Grammar of Graphics.



Grammar of Graphics 🧐

- **Types of geometry:** histogram, bar, point, box, line — `geom_`
- **Mapping variables:** axes, shape, size, color — `aes()`
- **Panels:** facets for variables with multiple values — `facet_`
- **Labels:** title, subtitle, axes labels, caption — `labs()`
- **Themes:** built-in themes or create your own — `theme_`

Thought Process

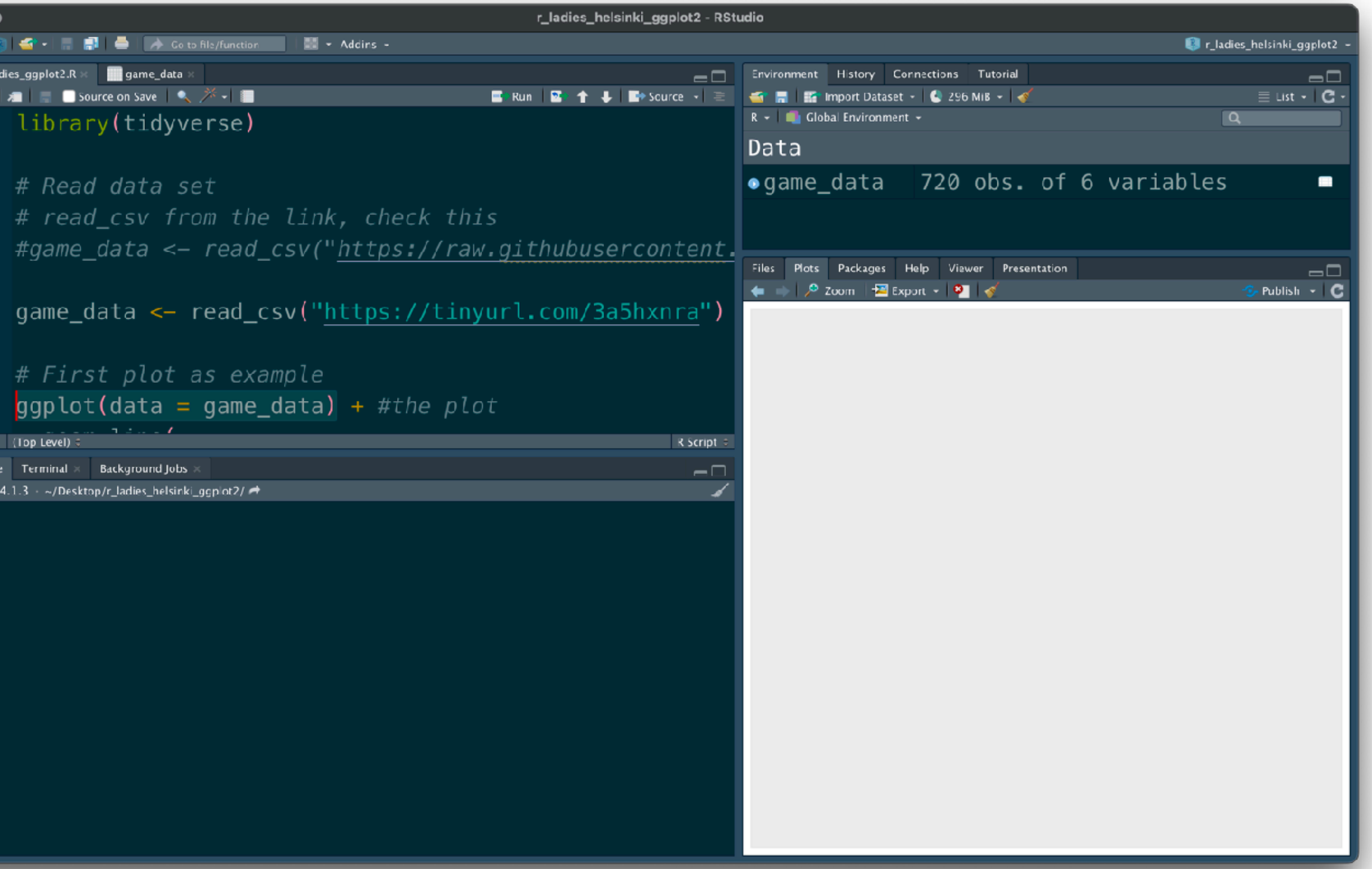
- You have some data

- You have some questions

- You imagine a plot *(a way to present your ideas, findings)*

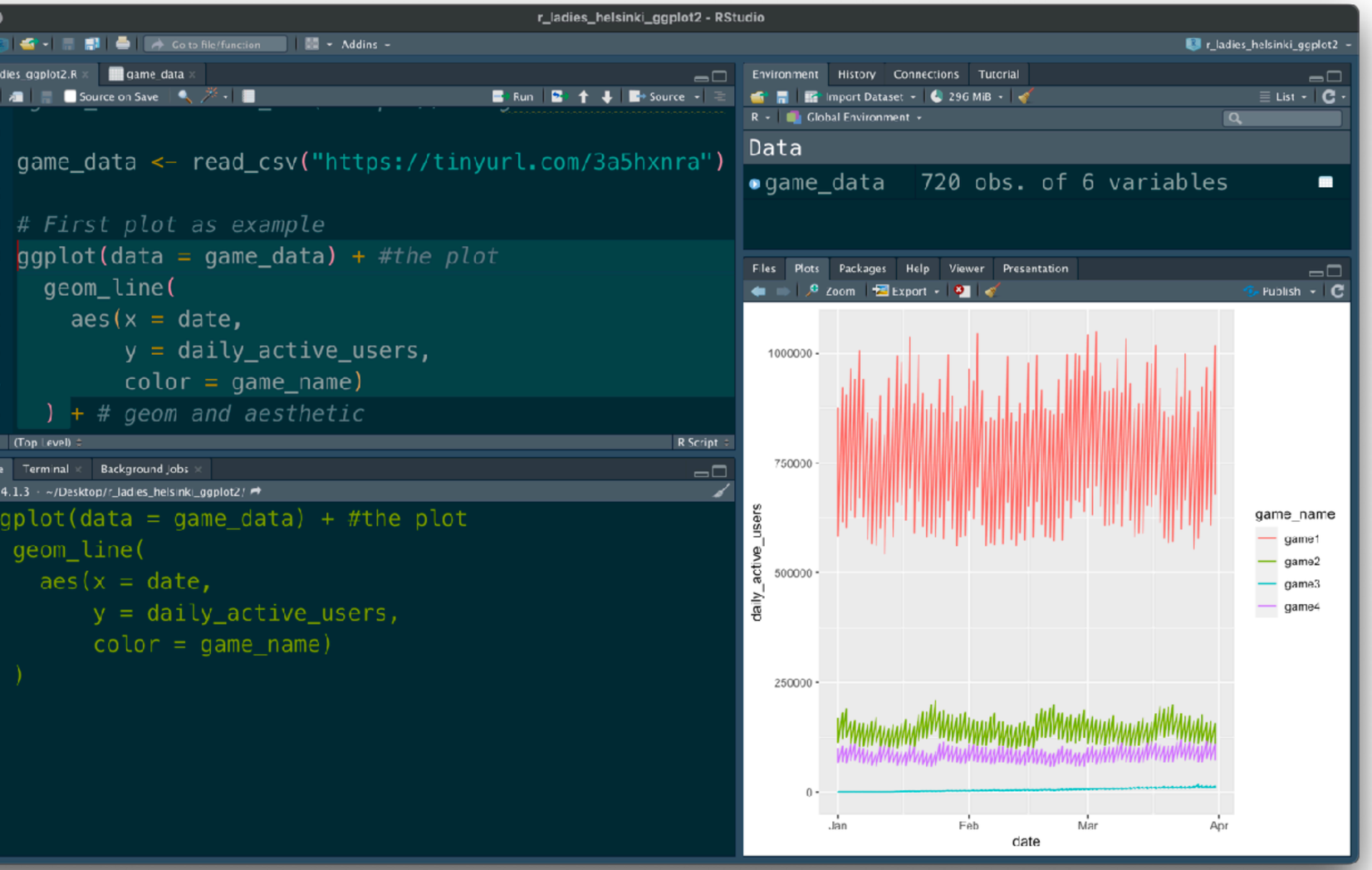
- Prepare your data and make the chart with ggplot2

Grammar of Graphics 🧐



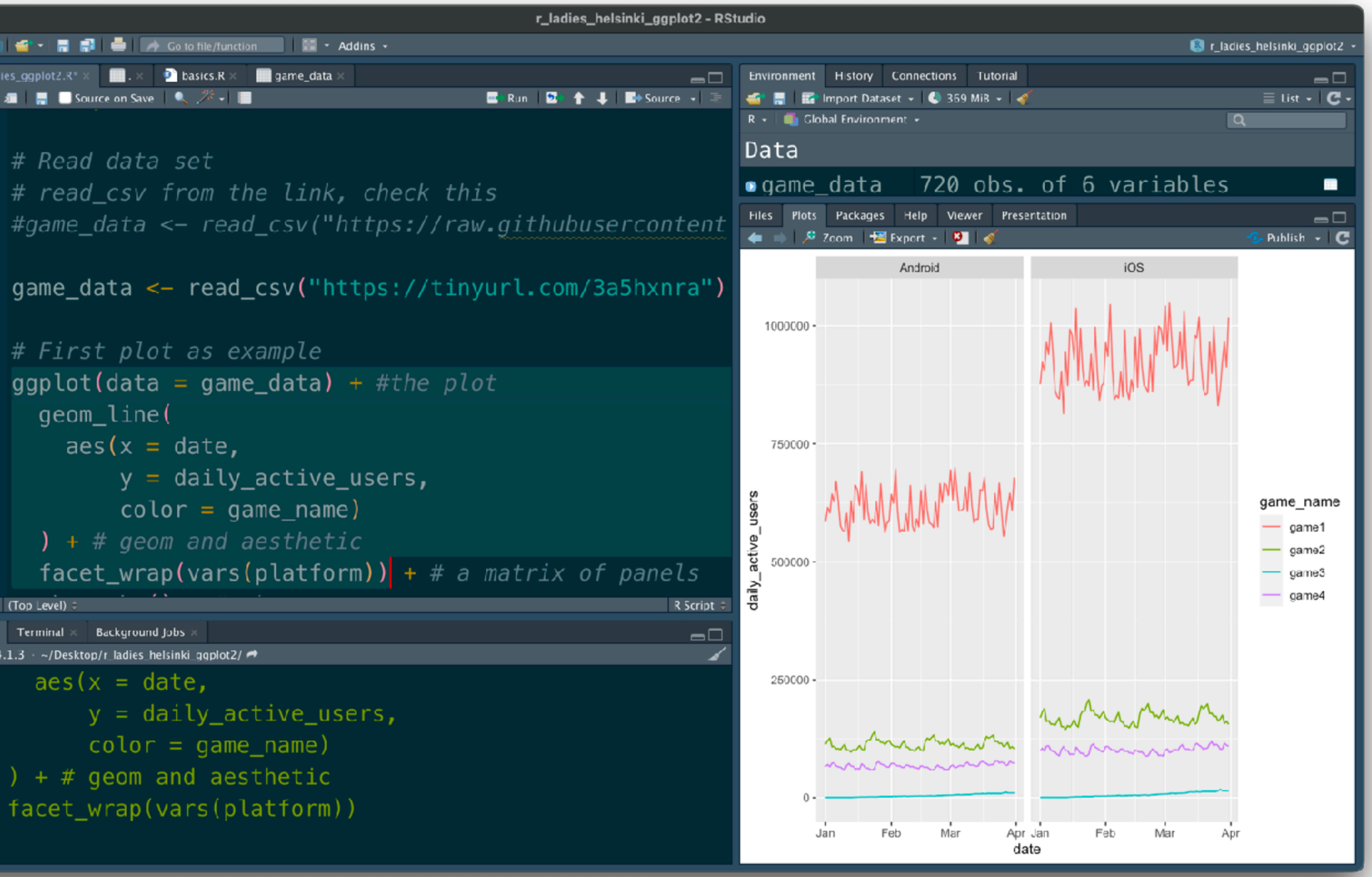

```
ggplot(data = game_data)
```

Grammar of Graphics 🧐



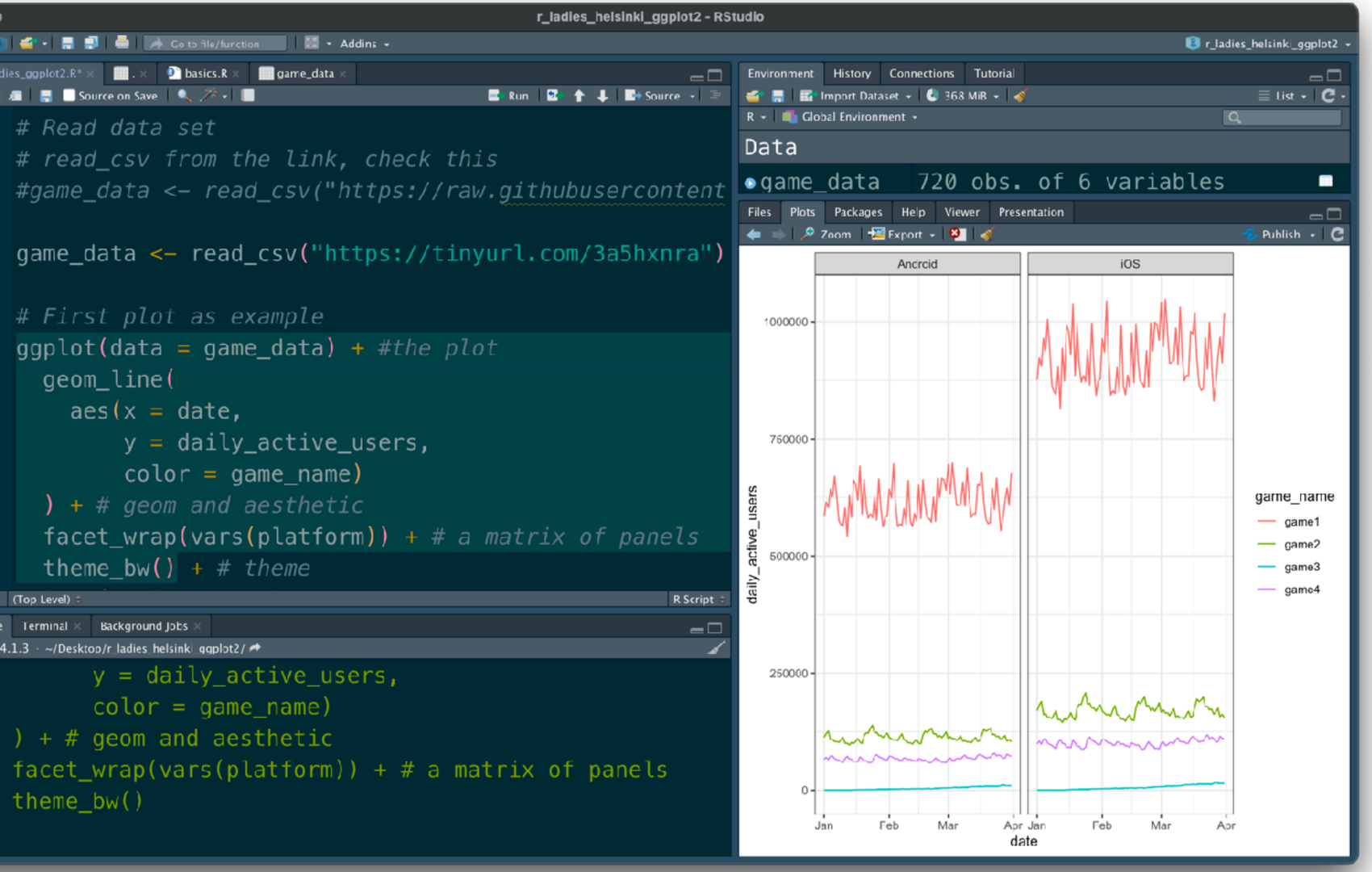

```
ggplot(data = game_data) +
geom_line(
  aes(x = date,
      y = daily_active_users,
      color = game_name)
   )
```

Grammar of Graphics 🧐🤨



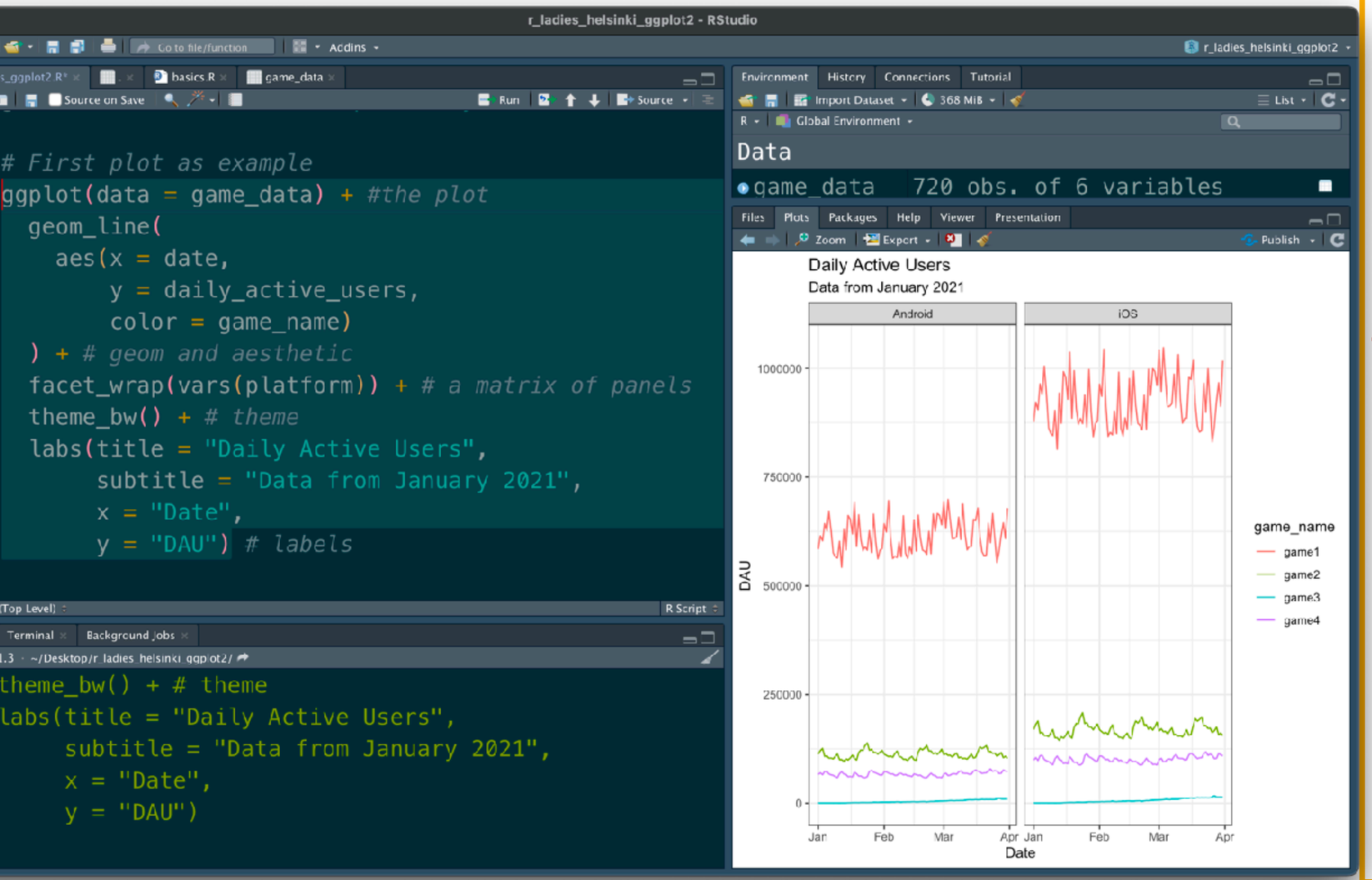

```
ggplot(data = game_data) +
geom_line(
   aes(x = date,
       y = daily_active_users,
       color = game_name)) +
facet_wrap(vars(platform))
```

Grammar of Graphics 🧐




```
ggplot(data = game_data) +
geom_line(
  aes(x = date,
      y = daily_active_users,
      color = game_name)) +
 facet_wrap(vars(platform)) +
 theme_bw()
```

Grammar of Graphics 🧐




```
ggplot(data = game_data) +
geom_line(
    aes(x = date,
        y = daily_active_users,
        color = game_name)) +
facet_wrap(vars(platform)) +
theme_bw() +
  labs(
   title = "Daily Active Users",
   subtitle = "Data from January 2021",
       x = "Date",
       y = "DAU")
```

Another Line Chart

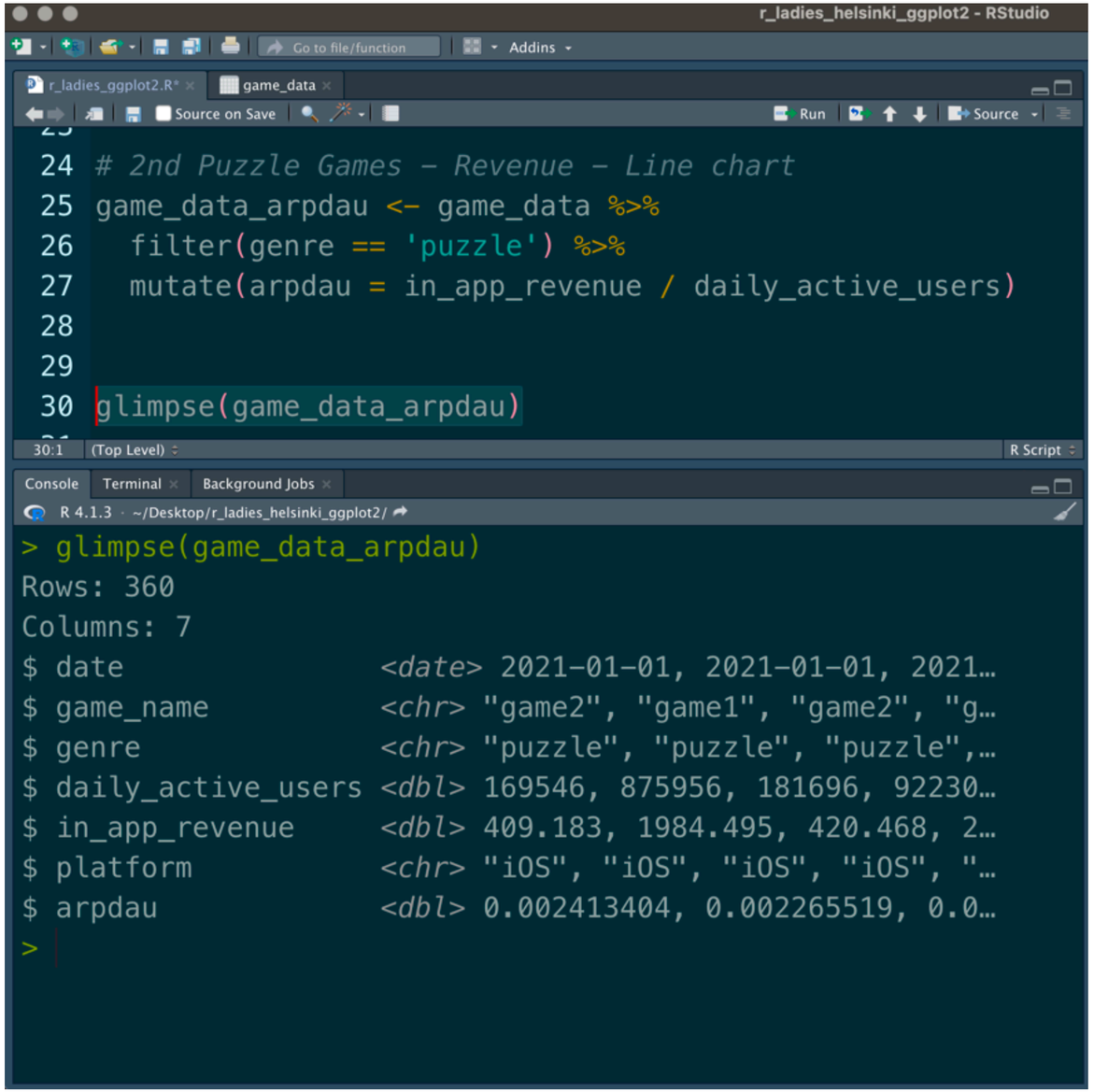
#Prep your data

- We want to know only puzzle games' daily

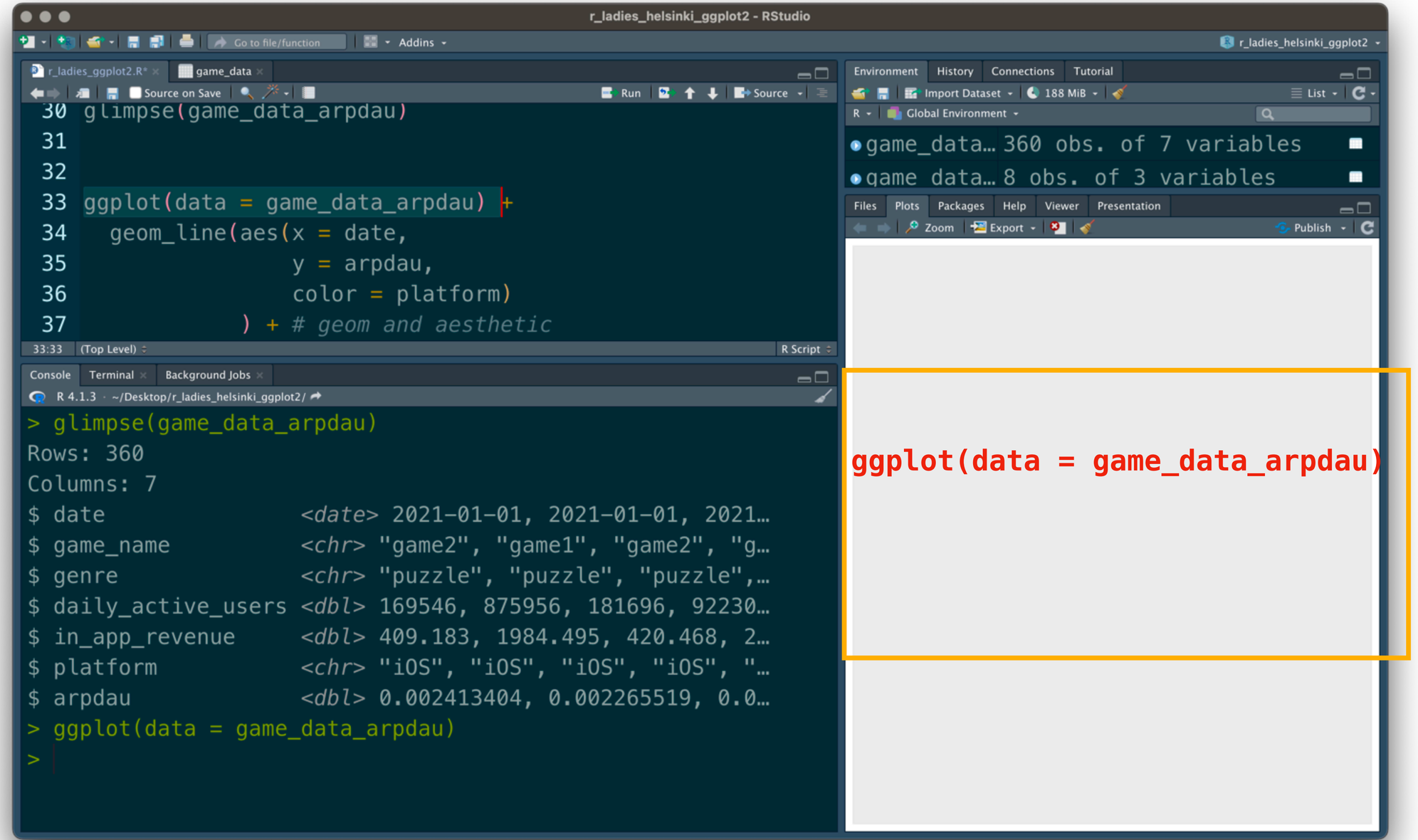
ARPDAU for each platform

- For that, we need to filter our data for puzzle

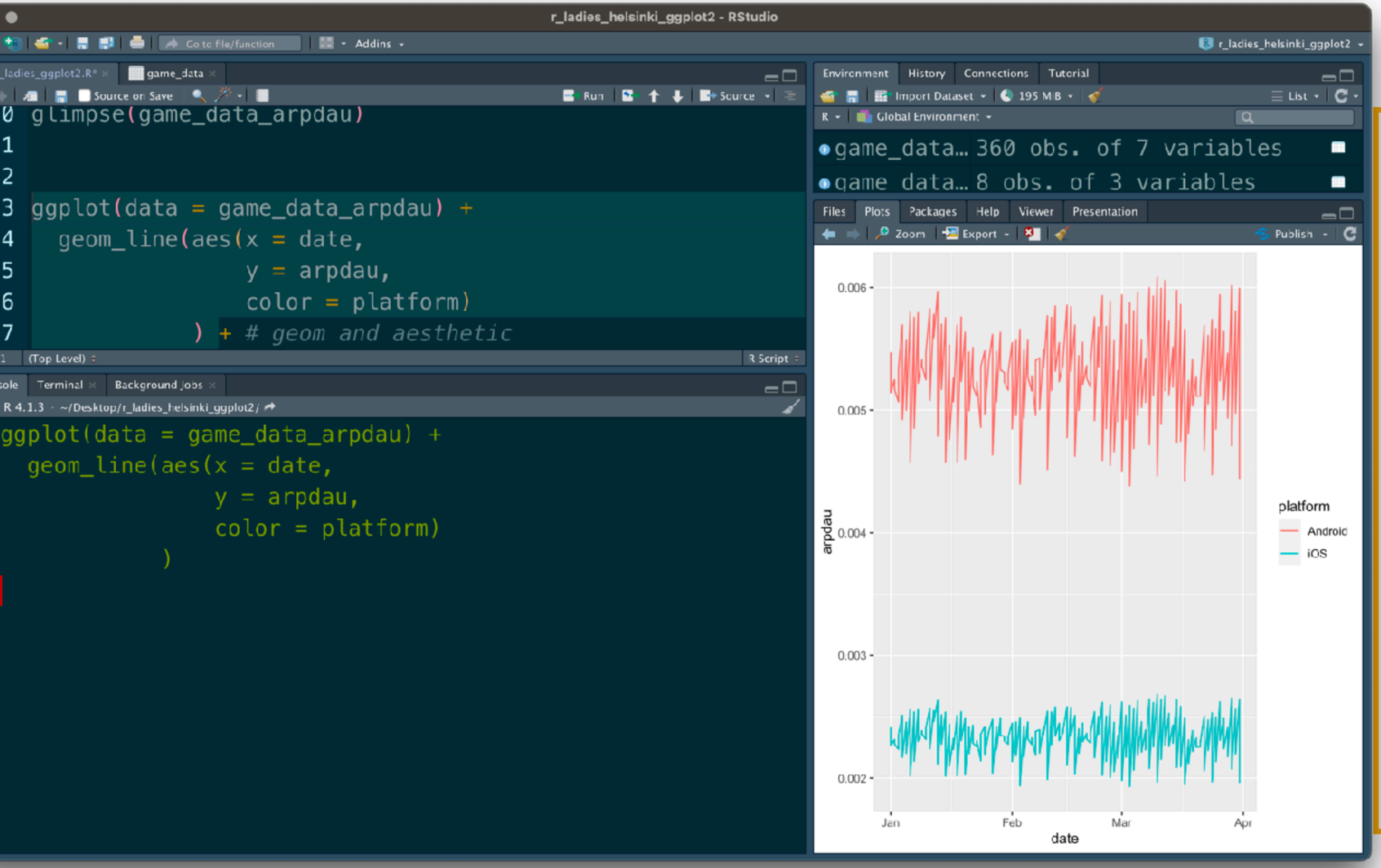
genre and mutate a new column for ARPDAU



Line chart: add data

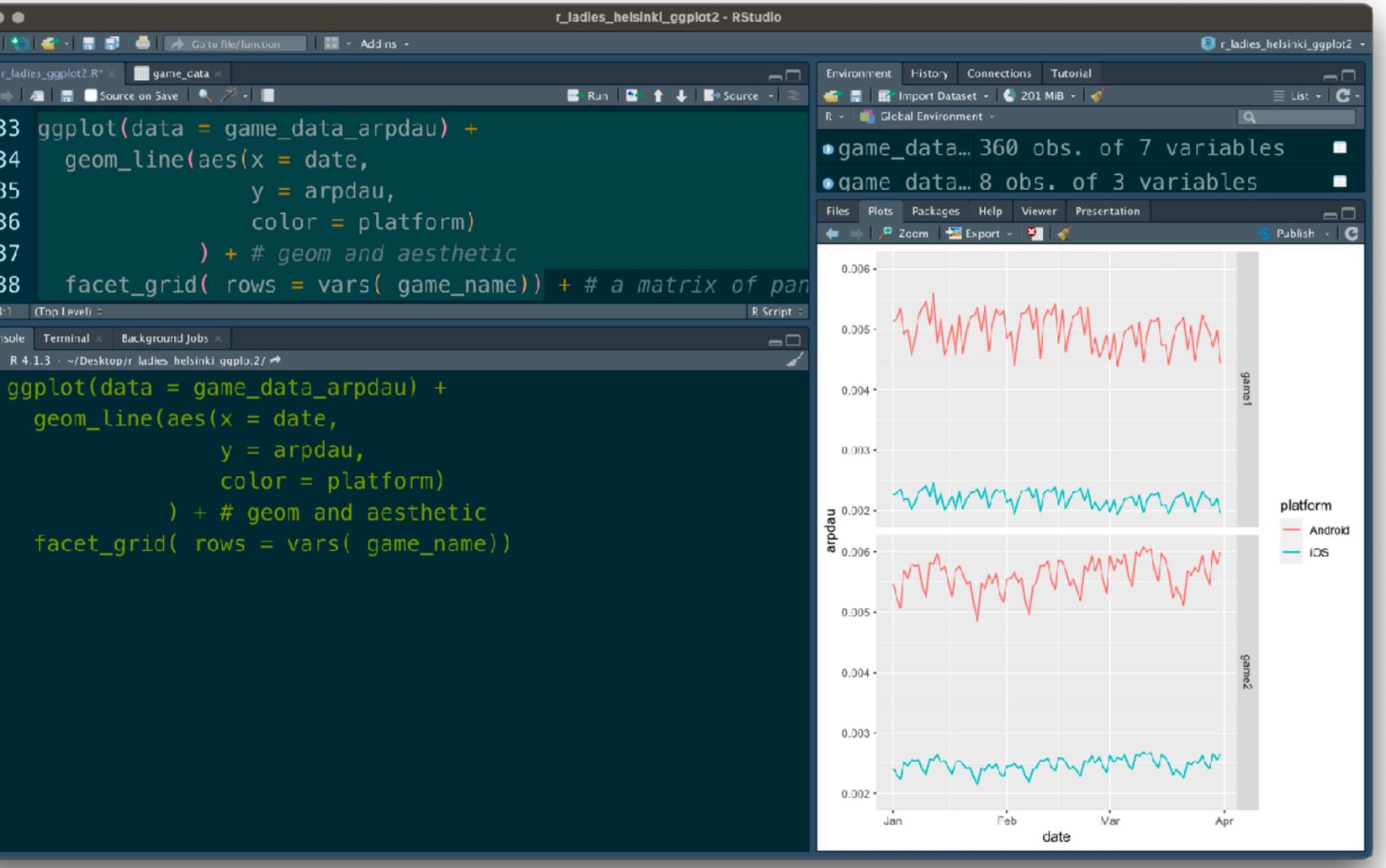


Line chart: add geom



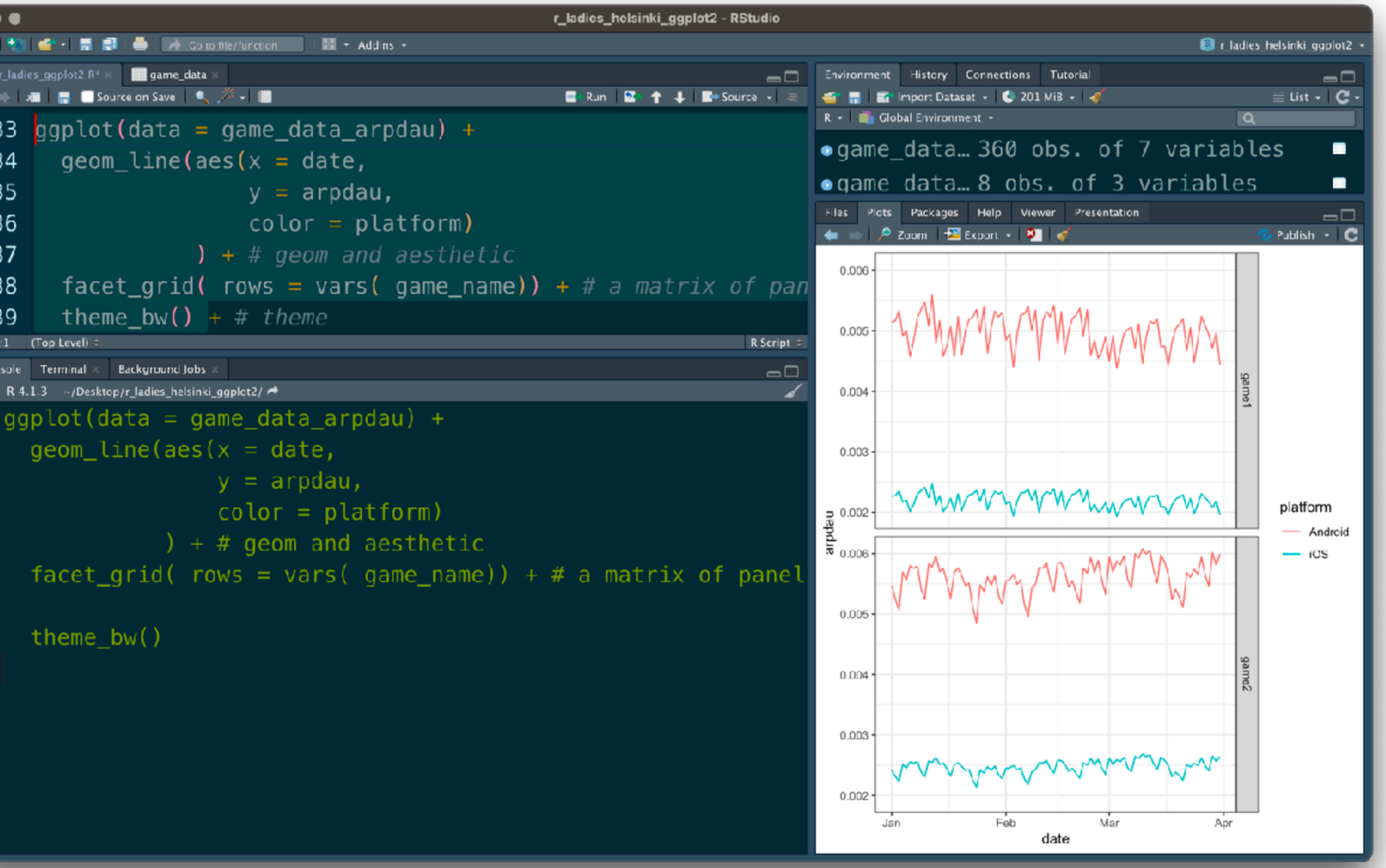

```
ggplot(data = game_data_arpdau) +
geom_line(aes(x = date,
              y = arpdau,
              color = platform)
          )
```

Line chart: add facets



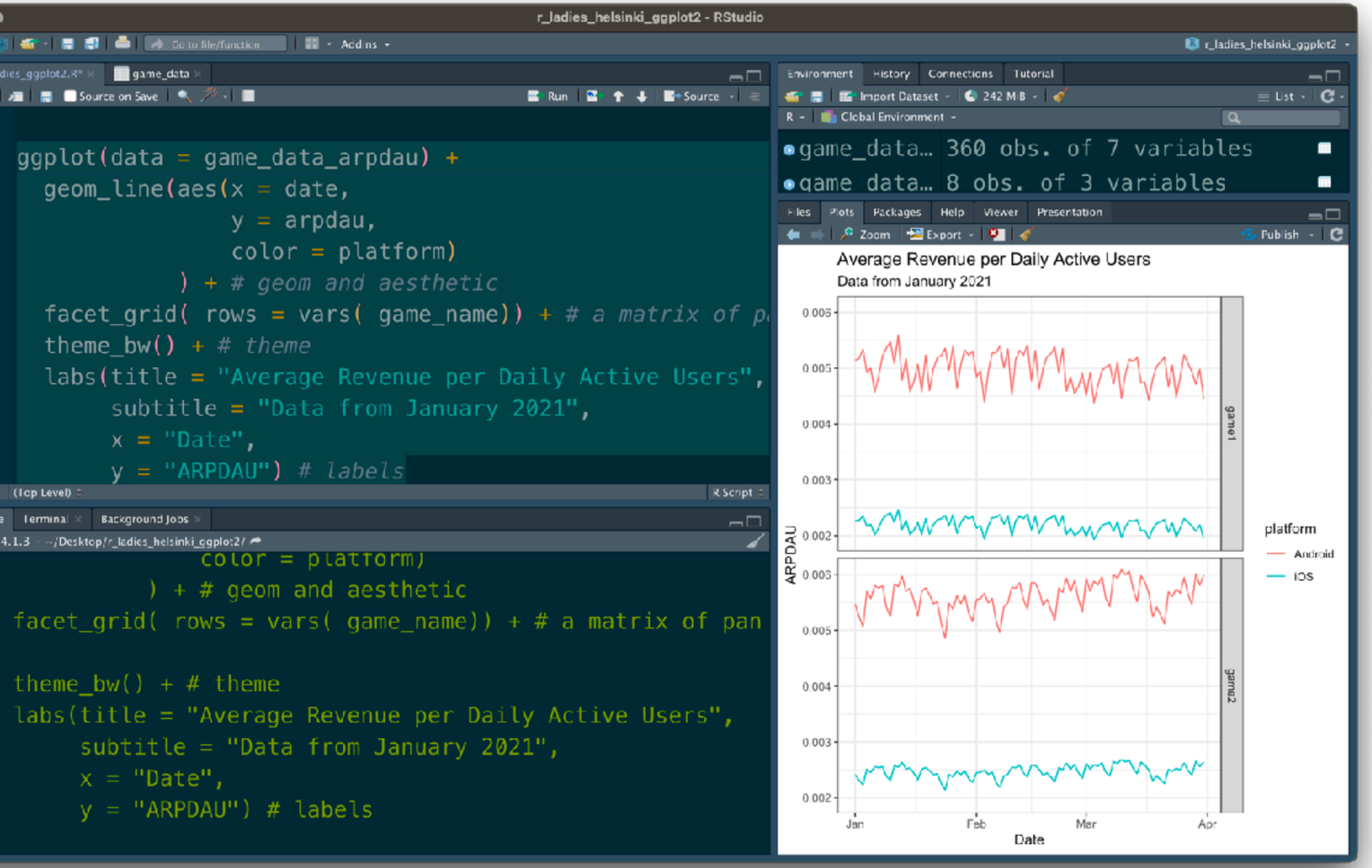

```
ggplot(data = game_data_arpdau) +
geom_line(aes(x = date,
              y = arpdau,
              color = platform)
          ) +
facet_grid( rows = vars( game_name))
```

Line chart: change theme




```
ggplot(data = game_data_arpdau) +
geom_line(aes(x = date,
              y = arpdau,
              color = platform)
          ) +
facet_grid( rows = vars( game_name)) +
theme_bw()
```

Line chart: add labels




```
ggplot(data = game_data_arpdau) +
geom_line(aes(x = date,
              y = arpdau,
              color = platform)
             ) +
facet_grid( rows = vars( game_name)) +
theme_bw() +
labs(
    title = "Average Revenue per DAU",
    subtitle = "Data from January 2021",
       x = "Date",
       y = “ARPDAU")
```

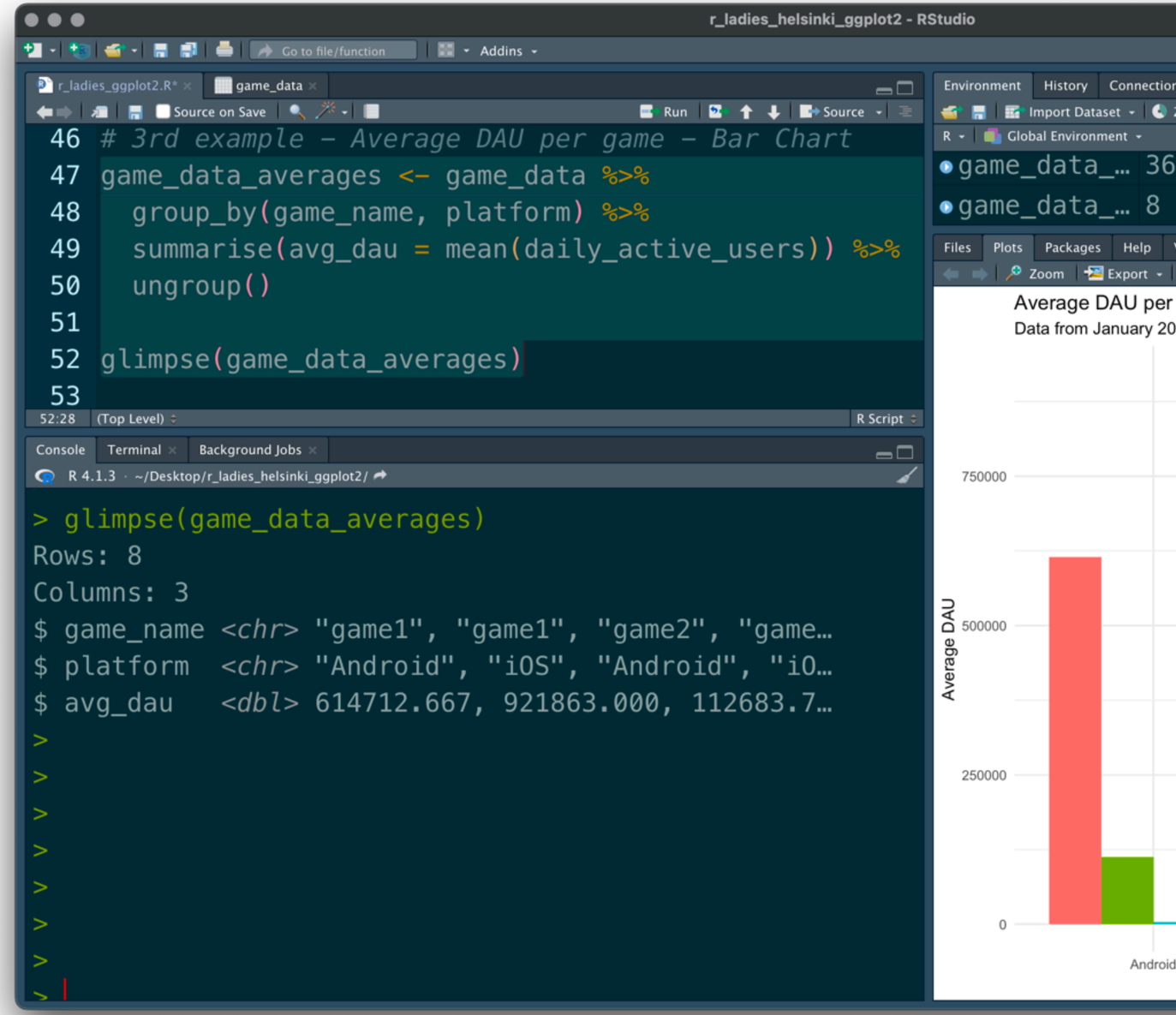
Time to create a bar chart

Create a Bar Chart

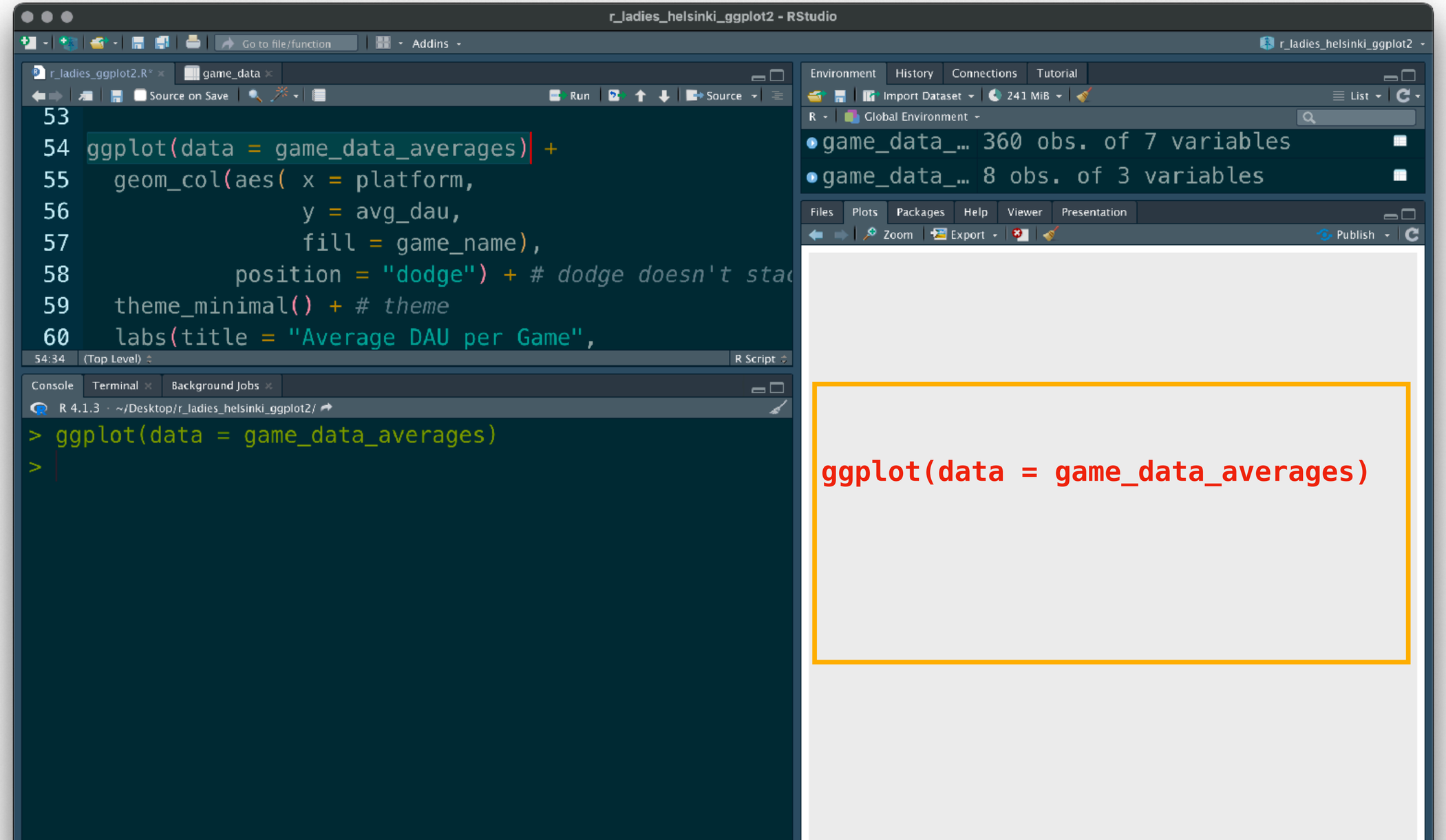
#Prep your data

- We want to know Average Daily Active Users per game and per platform.

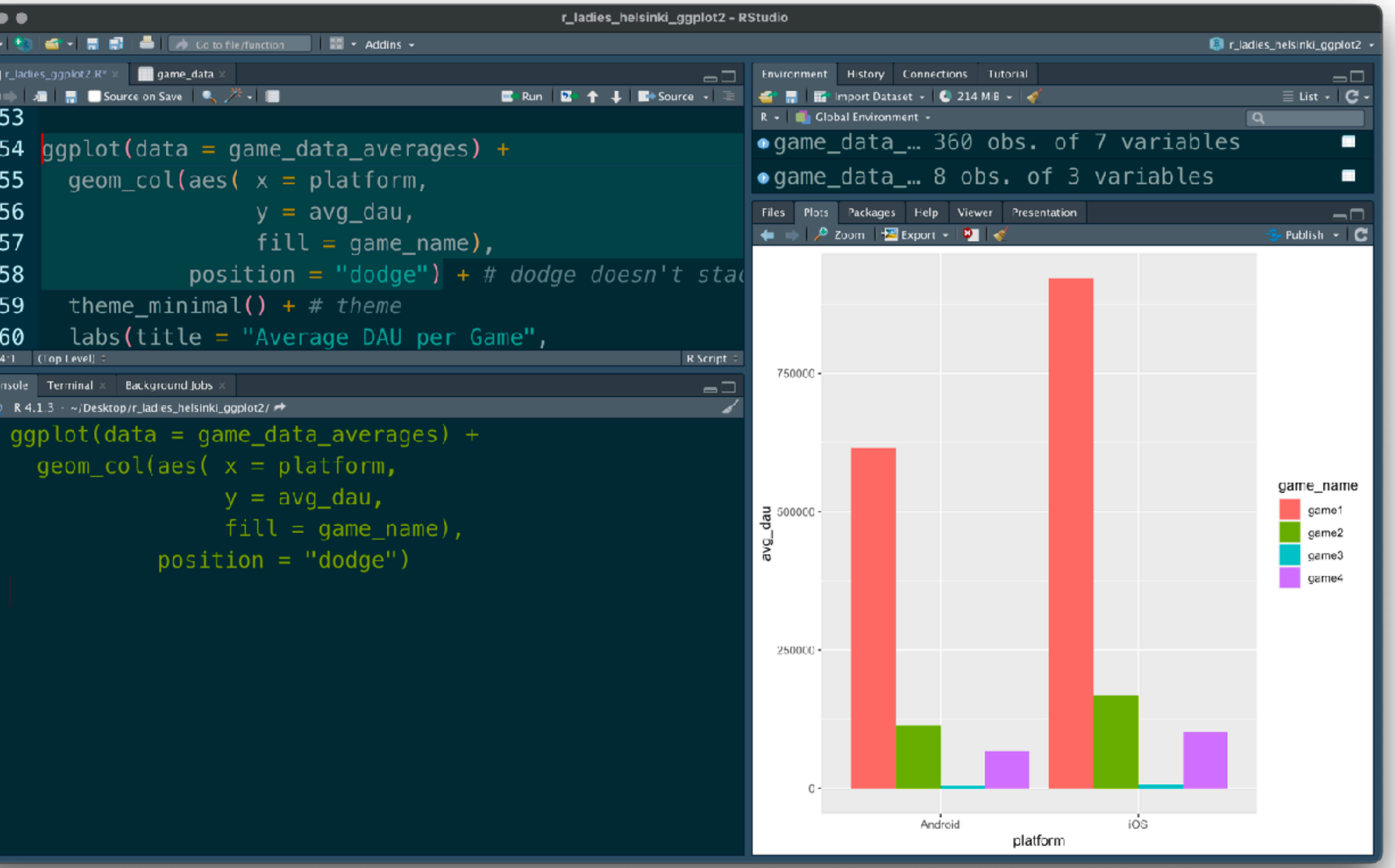
- For that, group our data by game, platform then summarise it for average values.



Bar Chart: add data

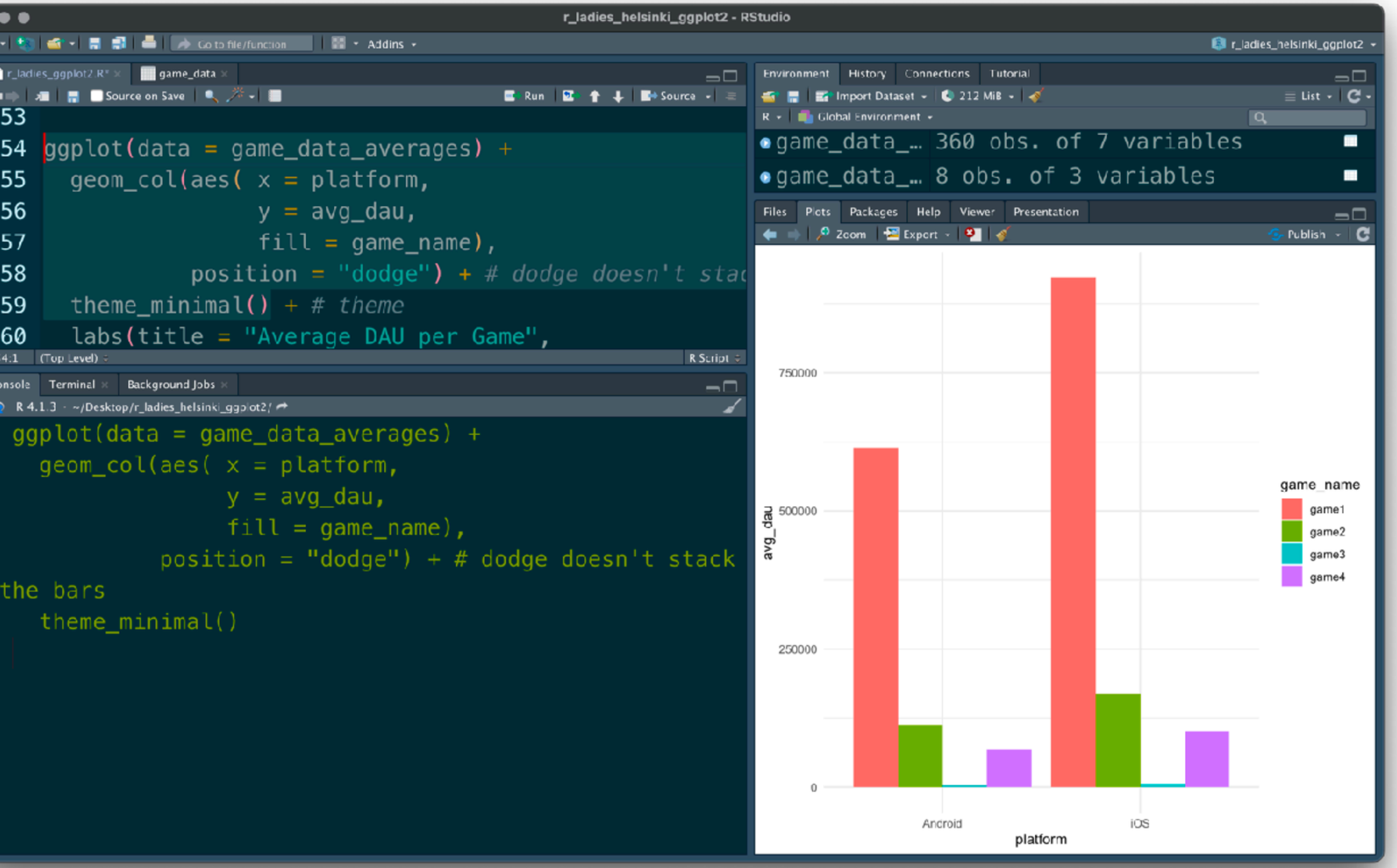


Bar Chart: add geom



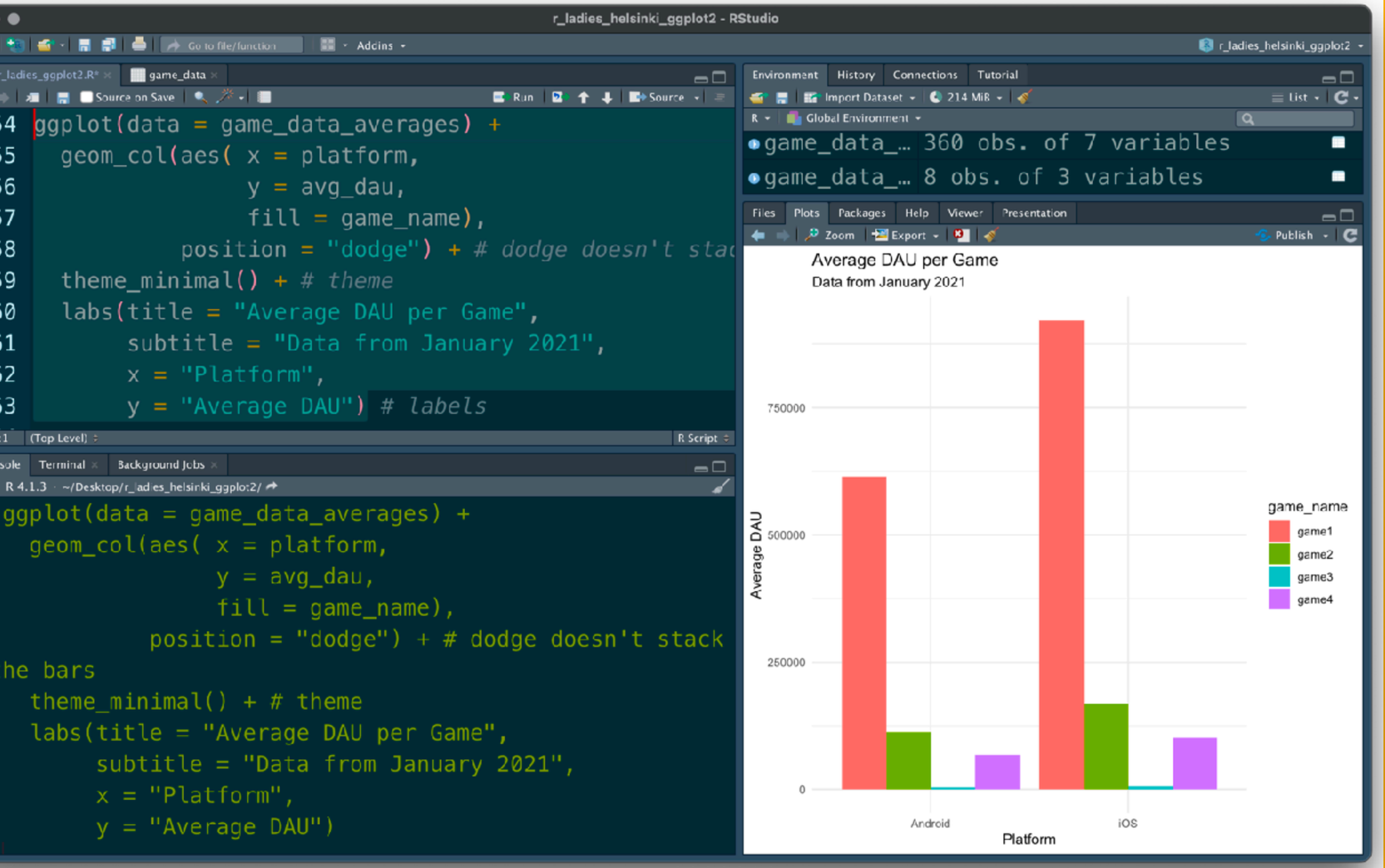

```
ggplot(data = game_data_averages) +
geom_col(aes( x = platform,
              y = avg_dau,
              fill = game_name),
              position = “dodge”)
```

Bar Chart: change theme




```
ggplot(data = game_data_averages) +
geom_col(aes( x = platform,
              y = avg_dau,
              fill = game_name),
              position = “dodge”) +
theme_minimal()
```

Bar Chart: add labels




```
ggplot(data = game_data_averages) +
geom_col(aes( x = platform,
              y = avg_dau,
              fill = game_name),
              position = “dodge”) +
theme_minimal() +
labs(title = "Avg DAU per Game",
    subtitle = "Data from Jan 2021",
    x = "Platform",
    y = "Average DAU")
```

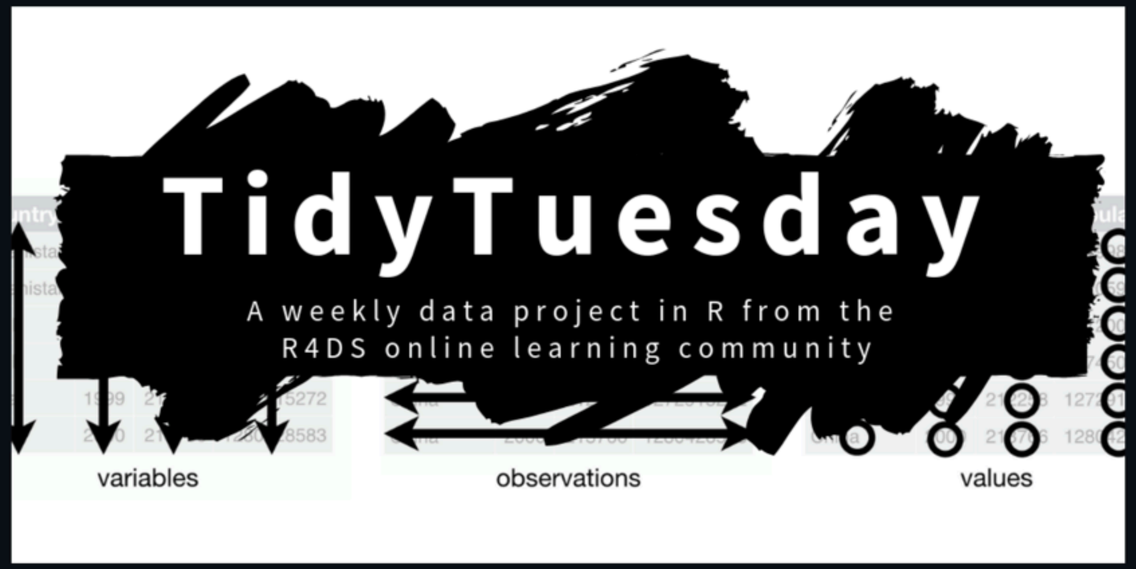
There is more to learn

- geom_histogram
- geom_point
- geom_jitter
- geom_density
- geom_boxplot
- geom_smooth
- geom_violin
- geom_text

- library(scales)
- library(ggthemes)
- library(ggExtra)
- library(plotly)

Check on #TidyTuesday

README.md



A weekly social data project in R

A weekly data project aimed at the R ecosystem. As this project was borne out of the `R4DS Online Learning Community` and the `R for Data Science` textbook, an emphasis was placed on understanding how to summarize and arrange data to make meaningful charts with `ggplot2`, `tidyr`, `dplyr`, and other tools in the `tidyverse` ecosystem. However, any code-based methodology is welcome - just please remember to share the code used to generate the results.

- https://github.com/rfordatascience/tidytuesday