

For Beginners

File Edit View Insert Cell Help

Go to file/function Addins

r_ladies_ggplot2.R Untitled1* game_data

Run Source

```
1 #this is R-Ladies Helsinki
2 A <- 10
3 a <- 3
4
5 myNumbers <- c(1:10)
6 myNumbers
7 rep(myNumbers, times = 3)
8
```

8:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.1.3 ~/Desktop/r_ladies_helsinki_ggplot2/

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

Environment History Connections Tutorial

Import Dataset 198 MiB

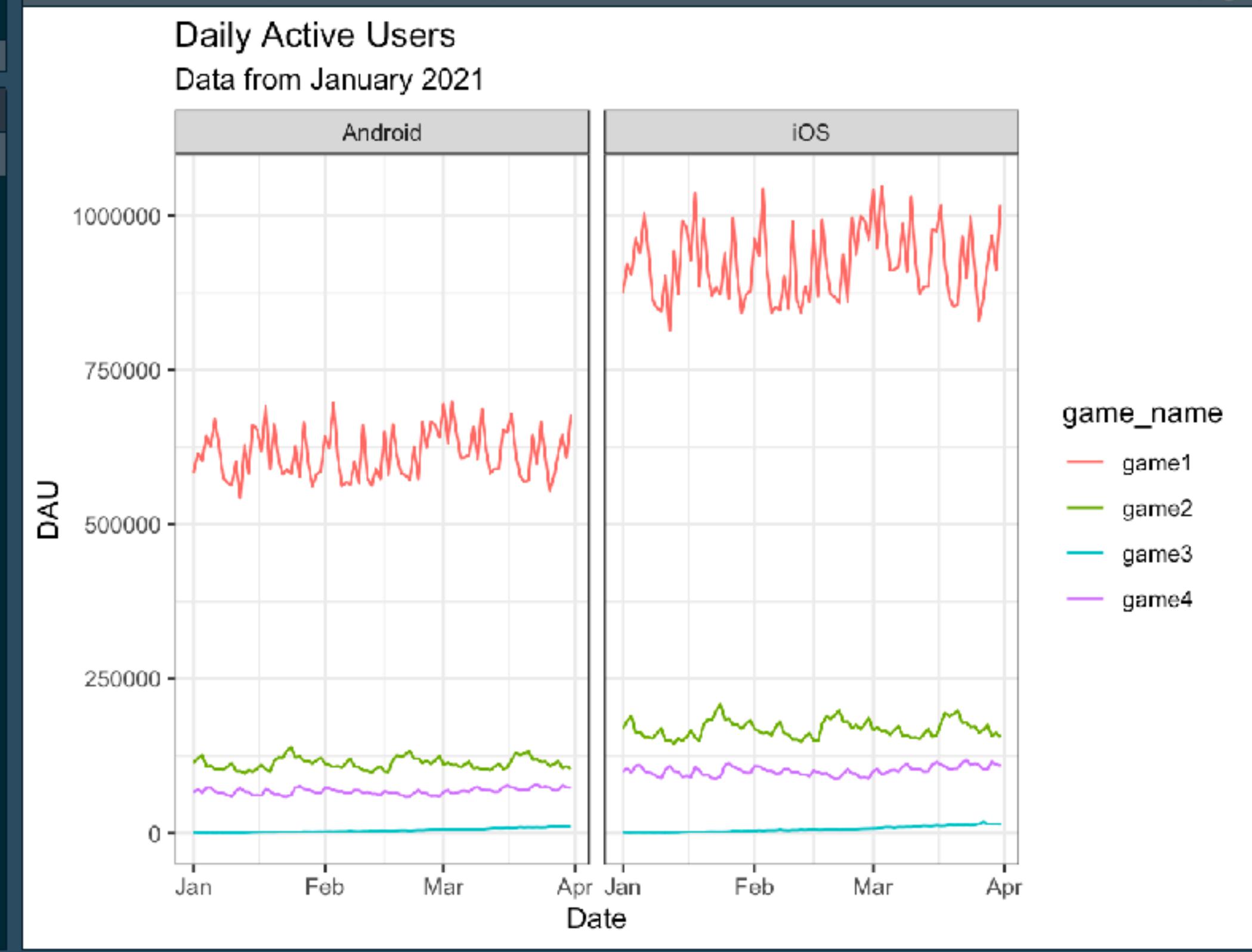
R Global Environment

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()
```

What is
that?

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", "i0...
```

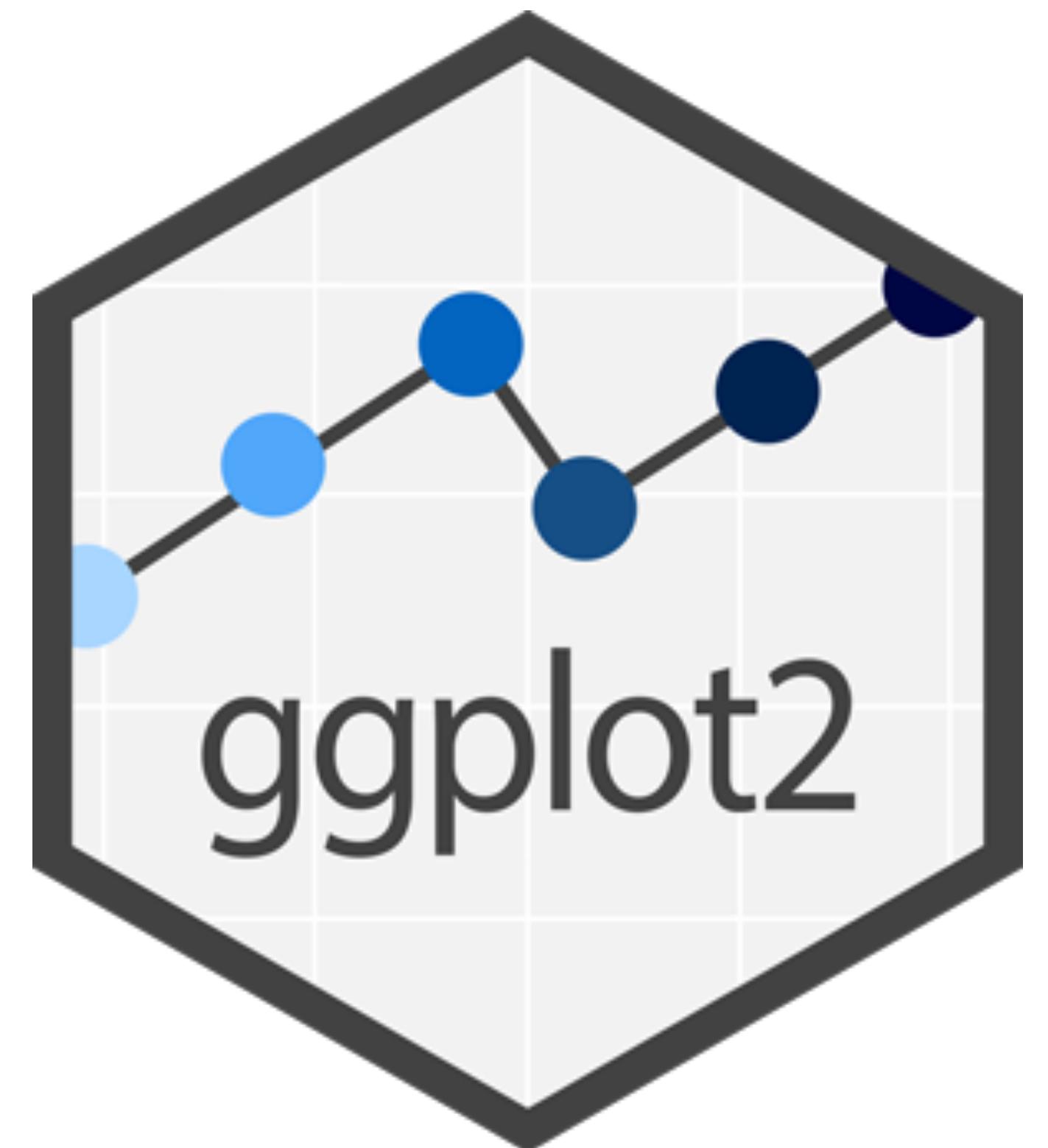
Get familiar today's data set!

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

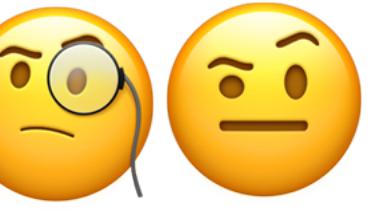
game_data %>%
View()

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

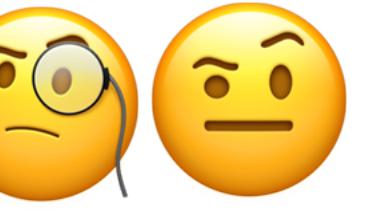
theme_

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



```
library(tidyverse)

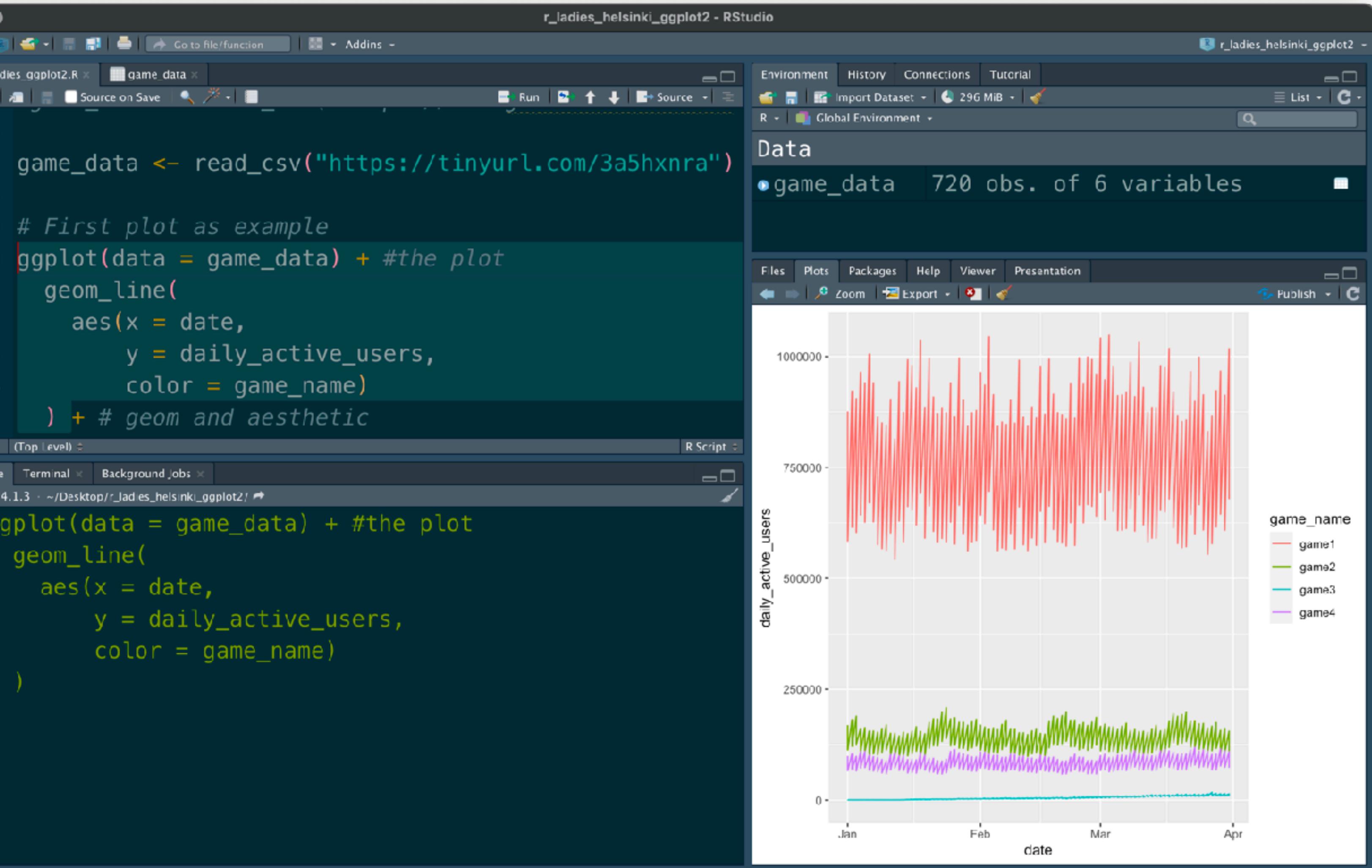
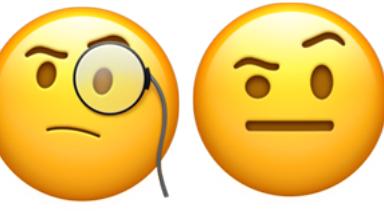
# Read data set
# read_csv from the link, check this
#game_data <- read_csv("https://raw.githubusercontent.com/r-ladies-helsinki/grammar-of-graphics/master/data/game_data.csv")

game_data <- read_csv("https://tinyurl.com/3a5hxnr")

# First plot as example
ggplot(data = game_data) + #the plot
```

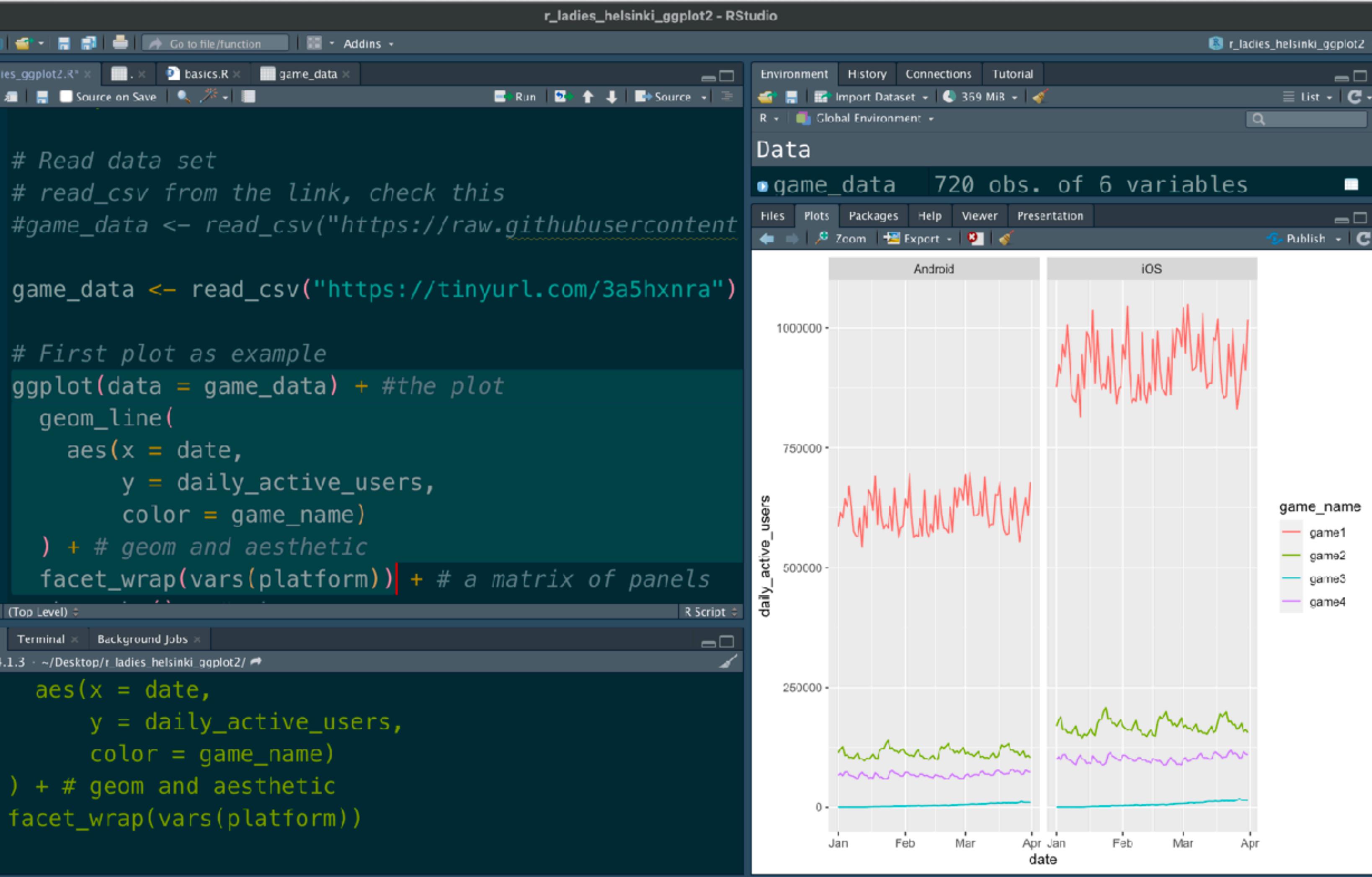
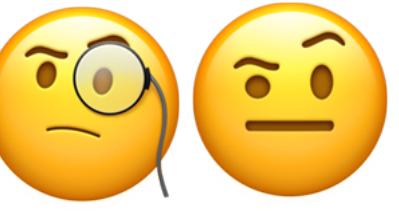
ggplot(data = game_data)

Grammar of Graphics



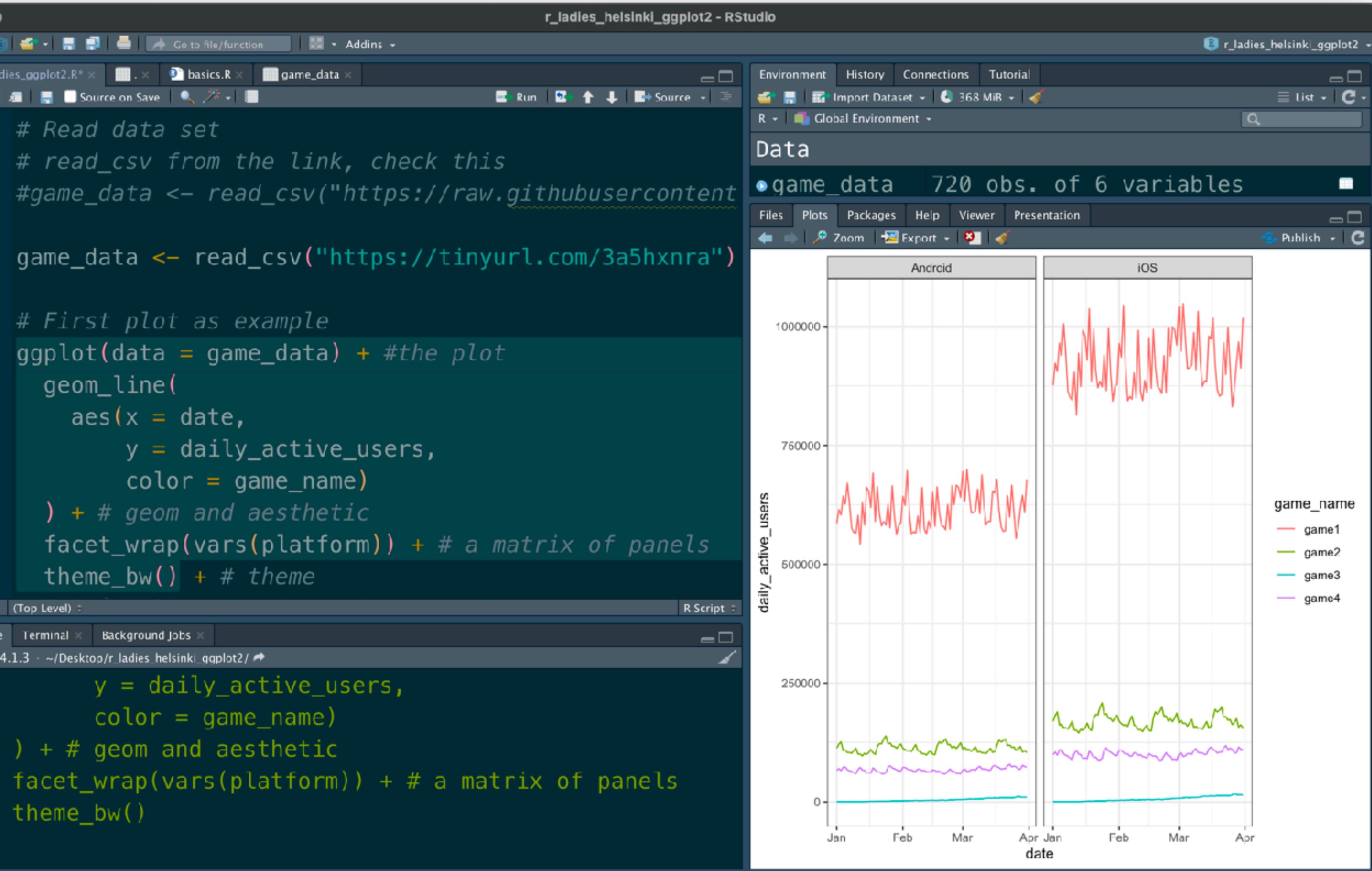
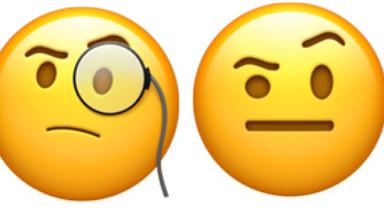
```
ggplot(data = game_data) +  
  geom_line(  
    aes(x = date,  
        y = daily_active_users,  
        color = game_name)  
  )
```

Grammar of Graphics



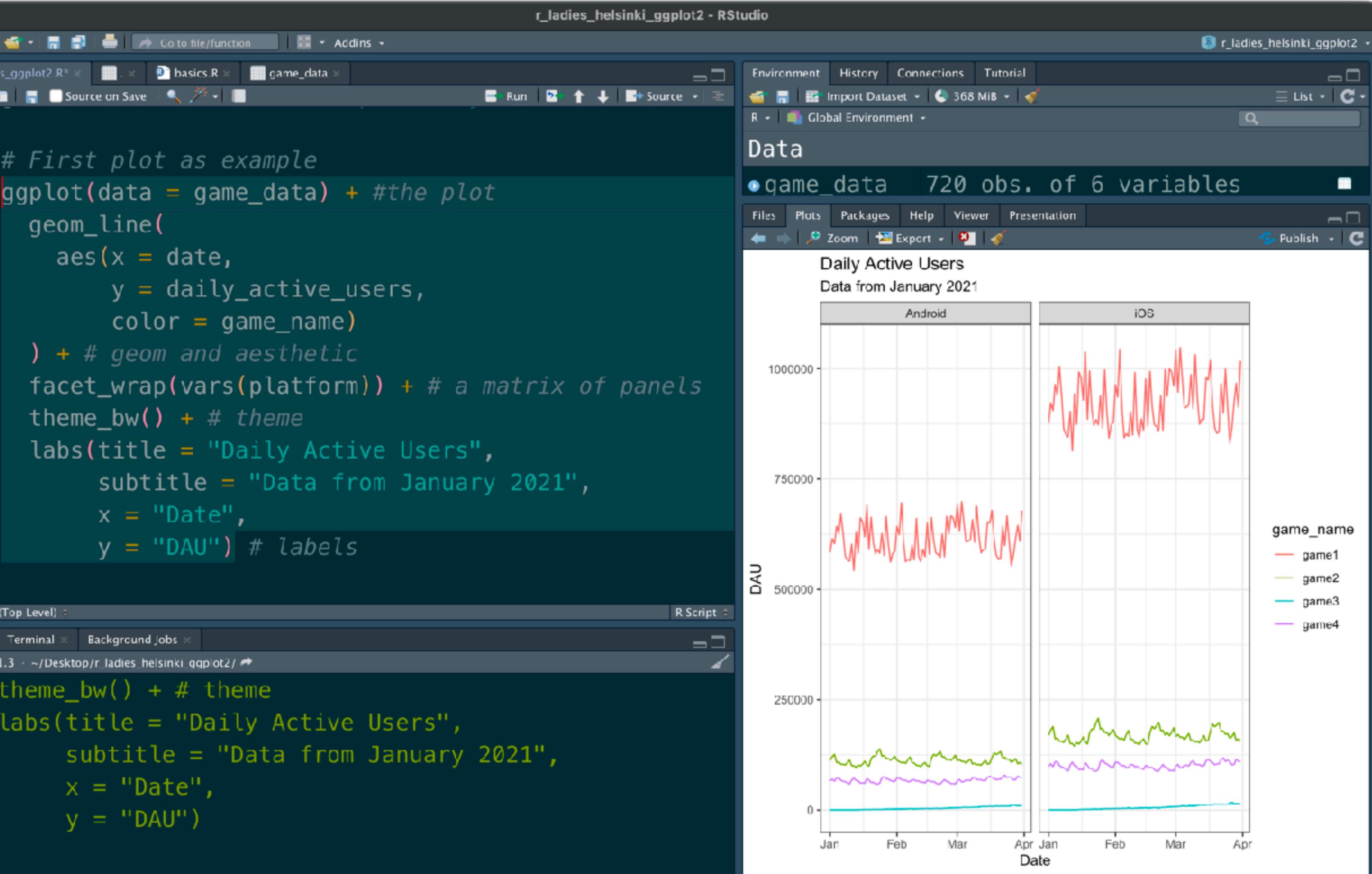
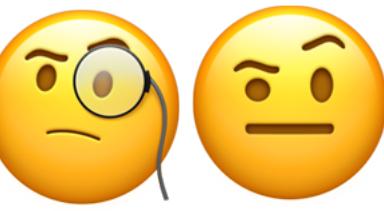
```
ggplot(data = game_data) +
  geom_line(
    aes(x = date,
        y = daily_active_users,
        color = game_name)) +
  facet_wrap(vars(platform))
```

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

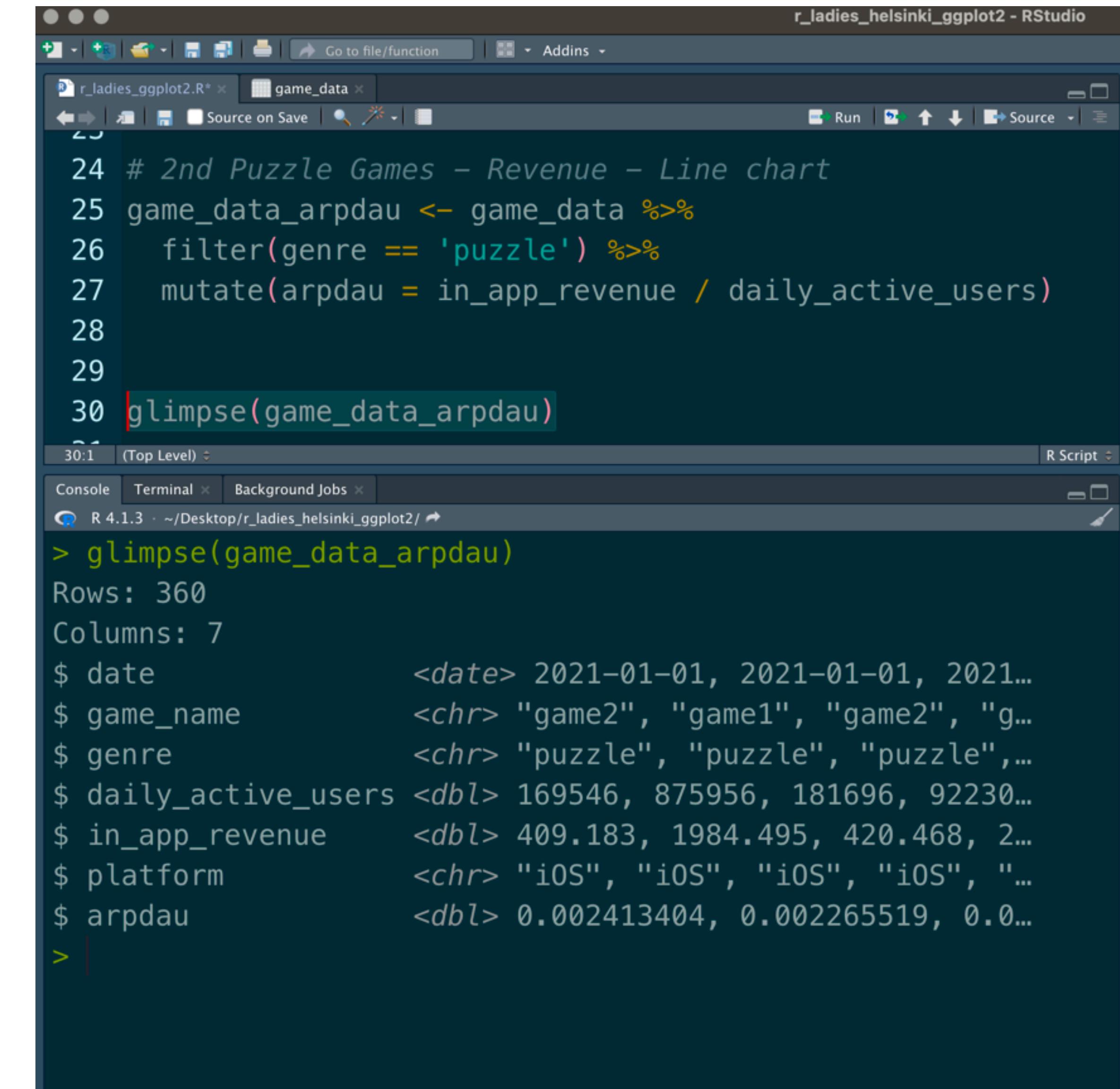
#Prep your data

- 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



The screenshot shows an RStudio interface with the following details:

- Code Editor:** The script pane displays R code for preparing data. The last line shown is `30 glimpse(game_data_arpdau)`.
- Console:** The output pane shows the results of running the code. It includes:
 - The command: `> glimpse(game_data_arpdau)`
 - The output: Rows: 360, Columns: 7
 - A table of column types and values:

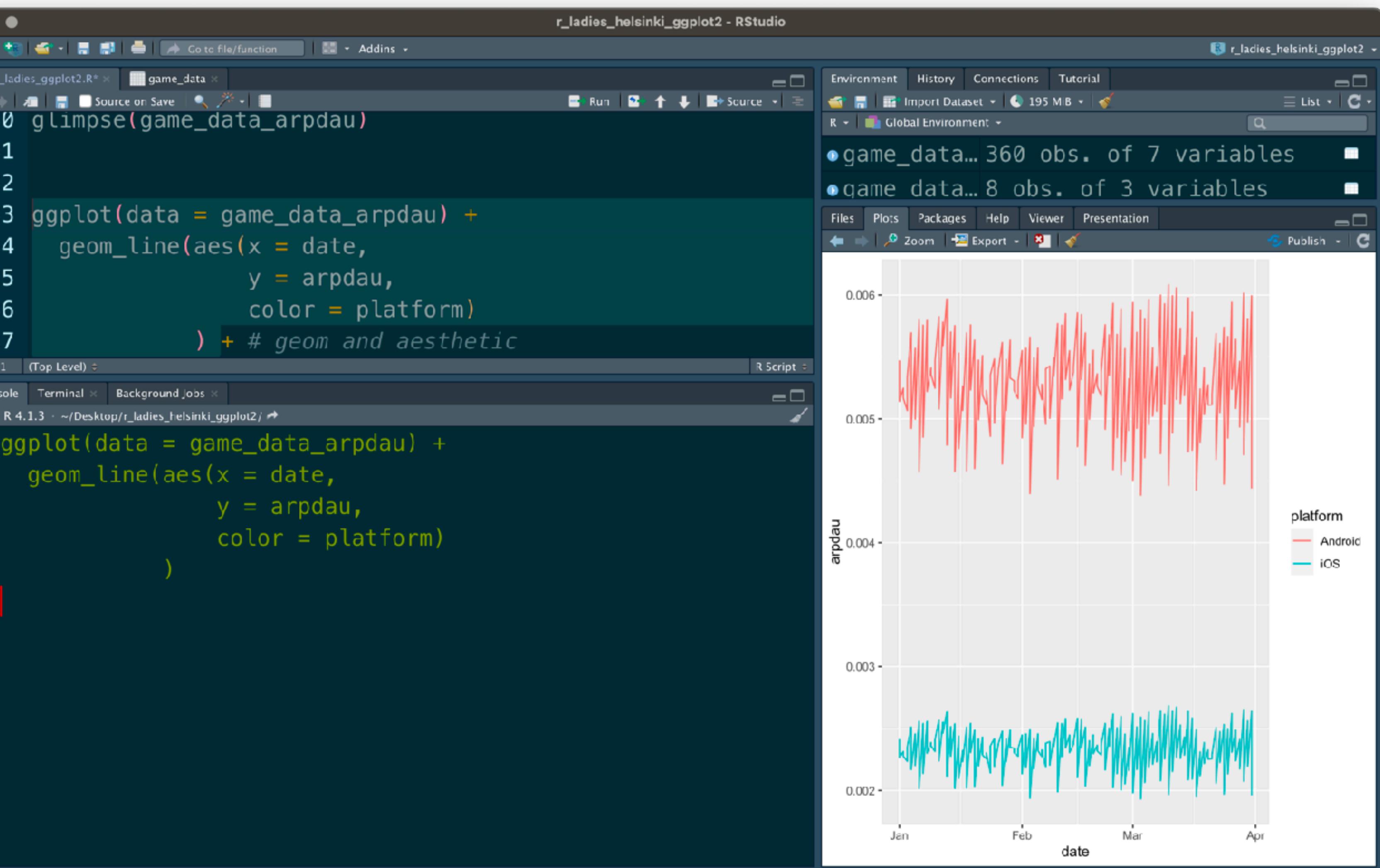
Column	Type	Values
\$ date	<date>	2021-01-01, 2021-01-01, 2021...
\$ game_name	<chr>	"game2", "game1", "game2", "g...
\$ genre	<chr>	"puzzle", "puzzle", "puzzle",...
\$ daily_active_users	<dbl>	169546, 875956, 181696, 92230...
\$ in_app_revenue	<dbl>	409.183, 1984.495, 420.468, 2...
\$ platform	<chr>	"iOS", "iOS", "iOS", "iOS", "...
\$ arpdau	<dbl>	0.002413404, 0.002265519, 0.0...

Line chart: add data

The screenshot shows the RStudio interface with the following components:

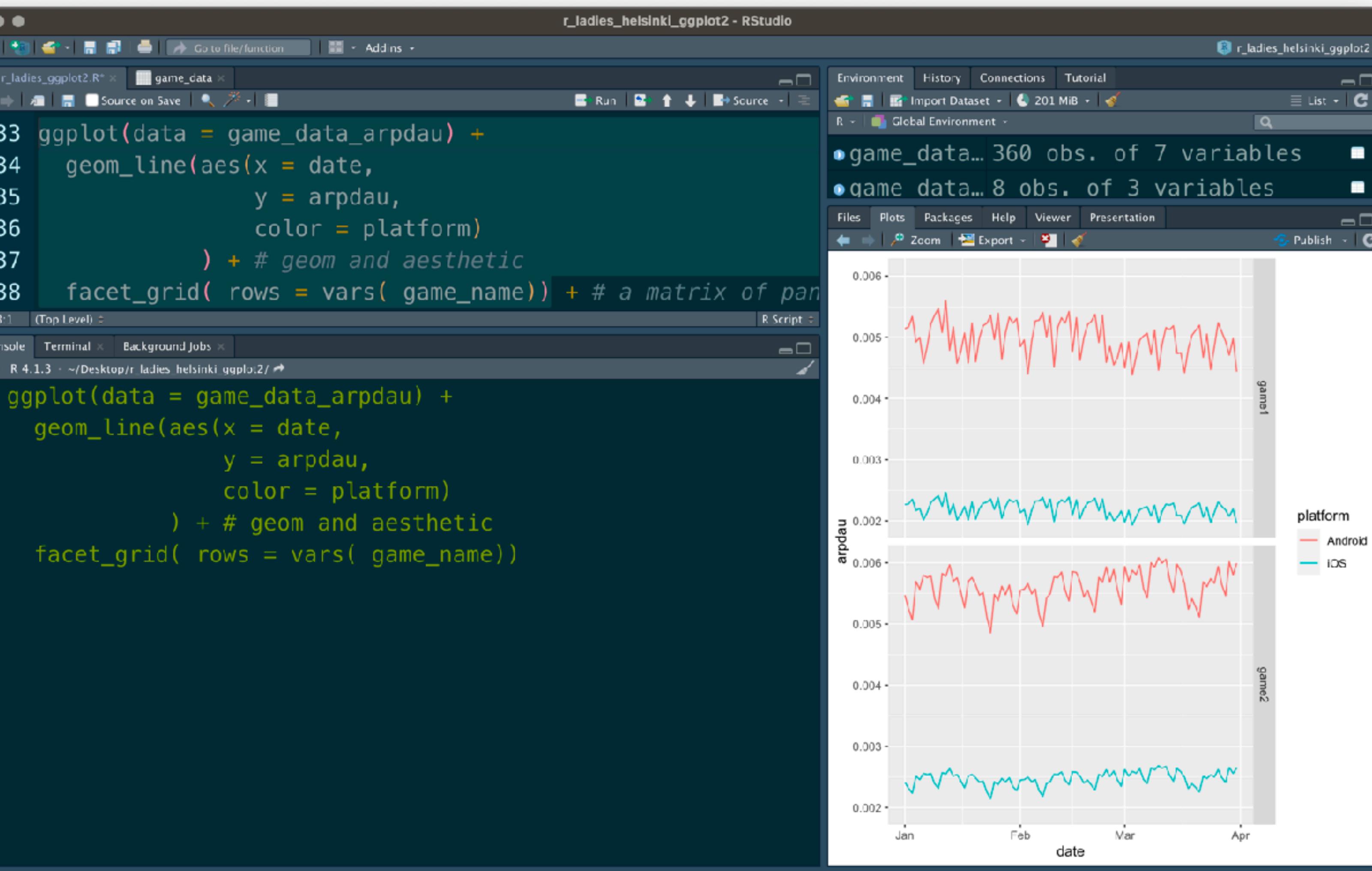
- Code Editor:** Displays the R script `r_ladies_helsinki_ggplot2.R`. The code includes:
 - Line 30: `glimpse(game_data_arpdau)`
 - Line 33: `ggplot(data = game_data_arpdau) + geom_line(aes(x = date, y = arpdau, color = platform))`
 - Line 37: A comment `) + # geom and aesthetic`
- Console:** Shows the output of the command `glimpse(game_data_arpdau)`, which provides details about the `game_data_arpdau` dataset:
 - Rows: 360
 - Columns: 7
 - Structure:
 - \$ date: `<date>` 2021-01-01, 2021-01-01, 2021...
 - \$ game_name: `<chr>` "game2", "game1", "game2", "g...
 - \$ genre: `<chr>` "puzzle", "puzzle", "puzzle",...
 - \$ daily_active_users: `<dbl>` 169546, 875956, 181696, 92230...
 - \$ in_app_revenue: `<dbl>` 409.183, 1984.495, 420.468, 2...
 - \$ platform: `<chr>` "iOS", "iOS", "iOS", "iOS", "
 - \$ arpdau: `<dbl>` 0.002413404, 0.002265519, 0.0...
- Environment:** Shows the `game_data_arpdau` dataset in the Global Environment, with 360 observations and 7 variables.
- Plots:** A large white area representing the plot canvas, with the ggplot command highlighted in red and enclosed in a yellow box.

Line chart: add geom



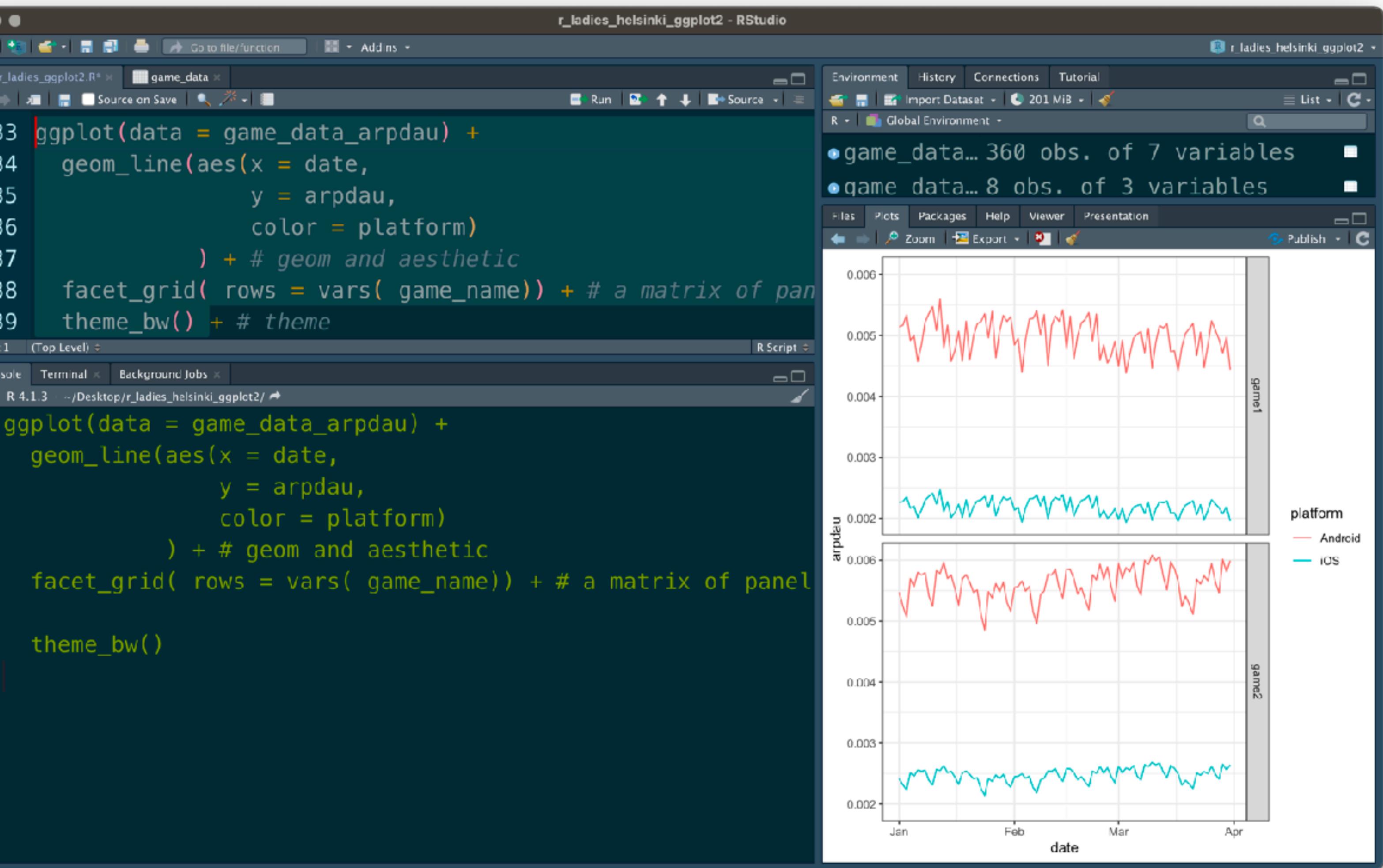
```
ggplot(data = game_data_arpdau) +
  geom_line(aes(x = date,
                y = arpdau,
                color = platform))
)
```

Line chart: add facets



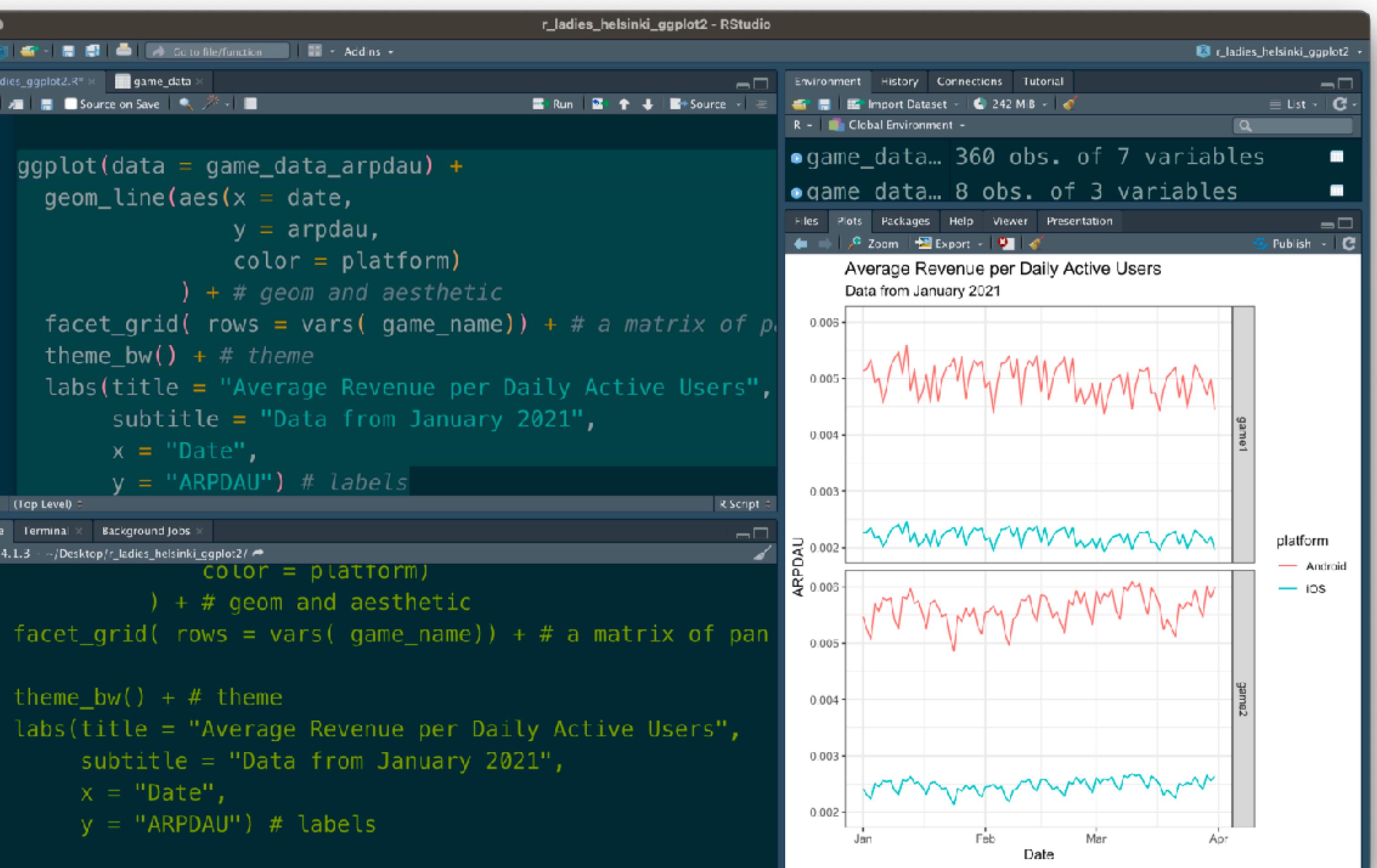
```
ggplot(data = game_data_arpdau) +  
  geom_line(aes(x = date,  
    y = arpdau,  
    color = platform))  
  ) +  
  facet_grid( rows = vars( game_name ))
```

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

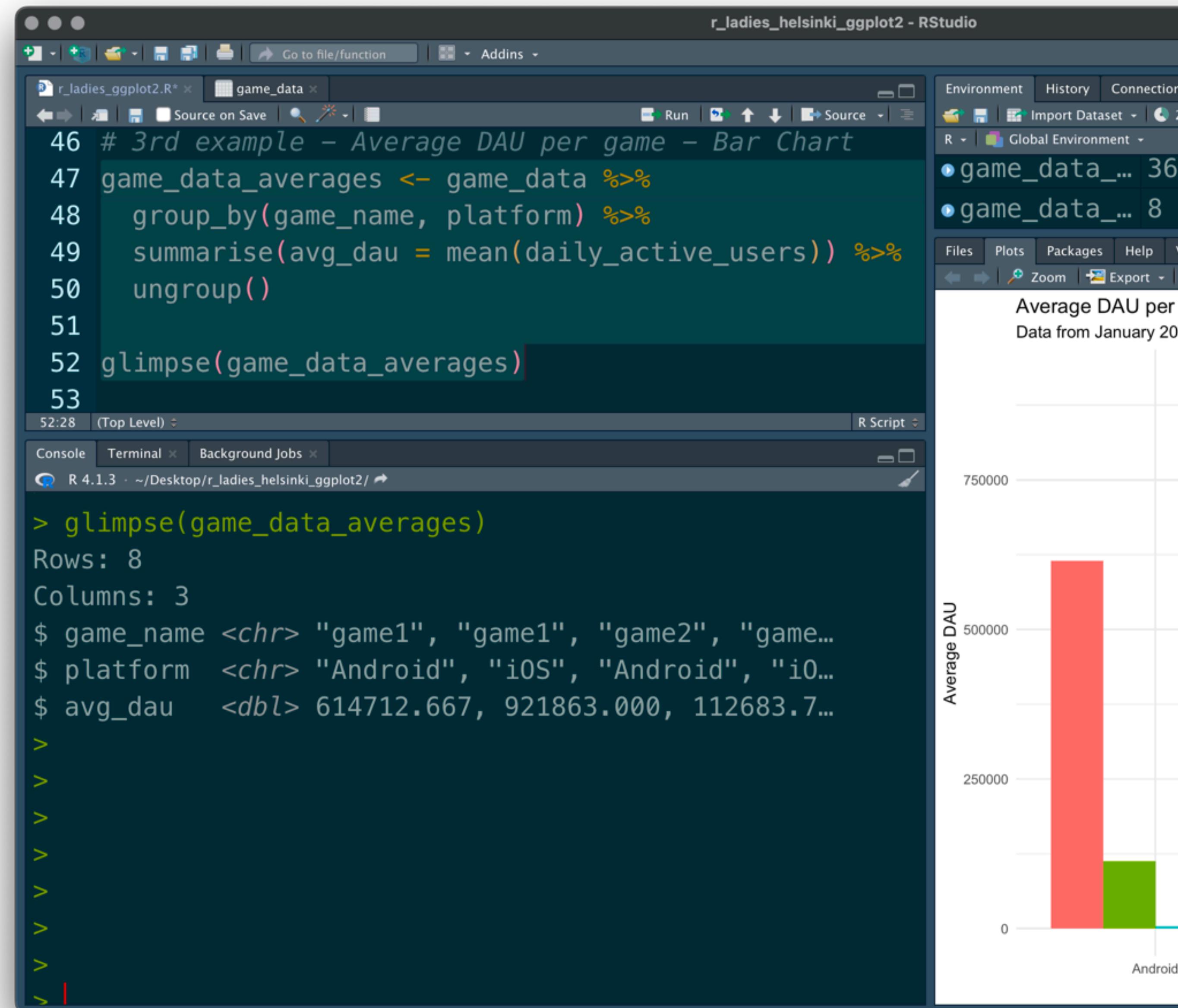
- We want to know **Average Daily Active**

Users per game and per platform.

- For that, **group** our data by game,

platform then **summarise** it for average

values.



Bar Chart: add data

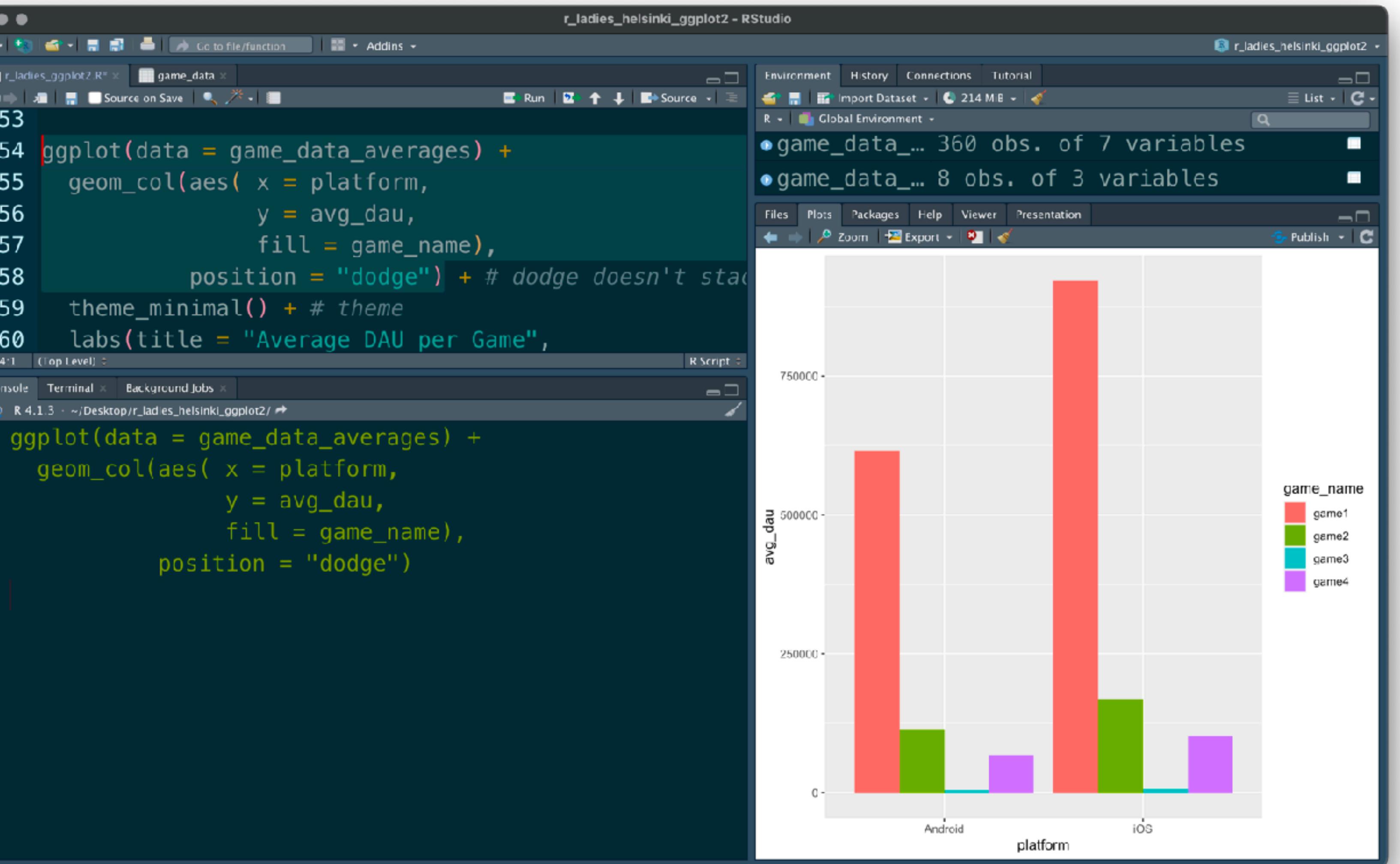
The screenshot shows the RStudio interface with the following components:

- Top Bar:** Shows the title "r_ladies_helsinki_ggplot2 - RStudio".
- File Explorer:** Shows files "r_ladies_ggplot2.R" and "game_data".
- Environment:** Shows the global environment with two objects:
 - "game_data_..." with 360 obs. of 7 variables.
 - "game_data_..." with 8 obs. of 3 variables.
- Plots:** A large white area representing the plot canvas.
- Code Editor:** Displays R code in a script named "r_ladies_ggplot2.R". The code defines a ggplot object with various layers and themes.

```
53
54 ggplot(data = game_data_averages) +
55   geom_col(aes( x = platform,
56                 y = avg_dau,
57                 fill = game_name),
58             position = "dodge") + # dodge doesn't stack
59   theme_minimal() + # theme
60   labs(title = "Average DAU per Game",
```
- Console:** Shows the command "ggplot(data = game_data_averages)" entered in the R console.
- Status Bar:** Shows "R 4.1.3 ~/Desktop/r_ladies_helsinki_ggplot2/".

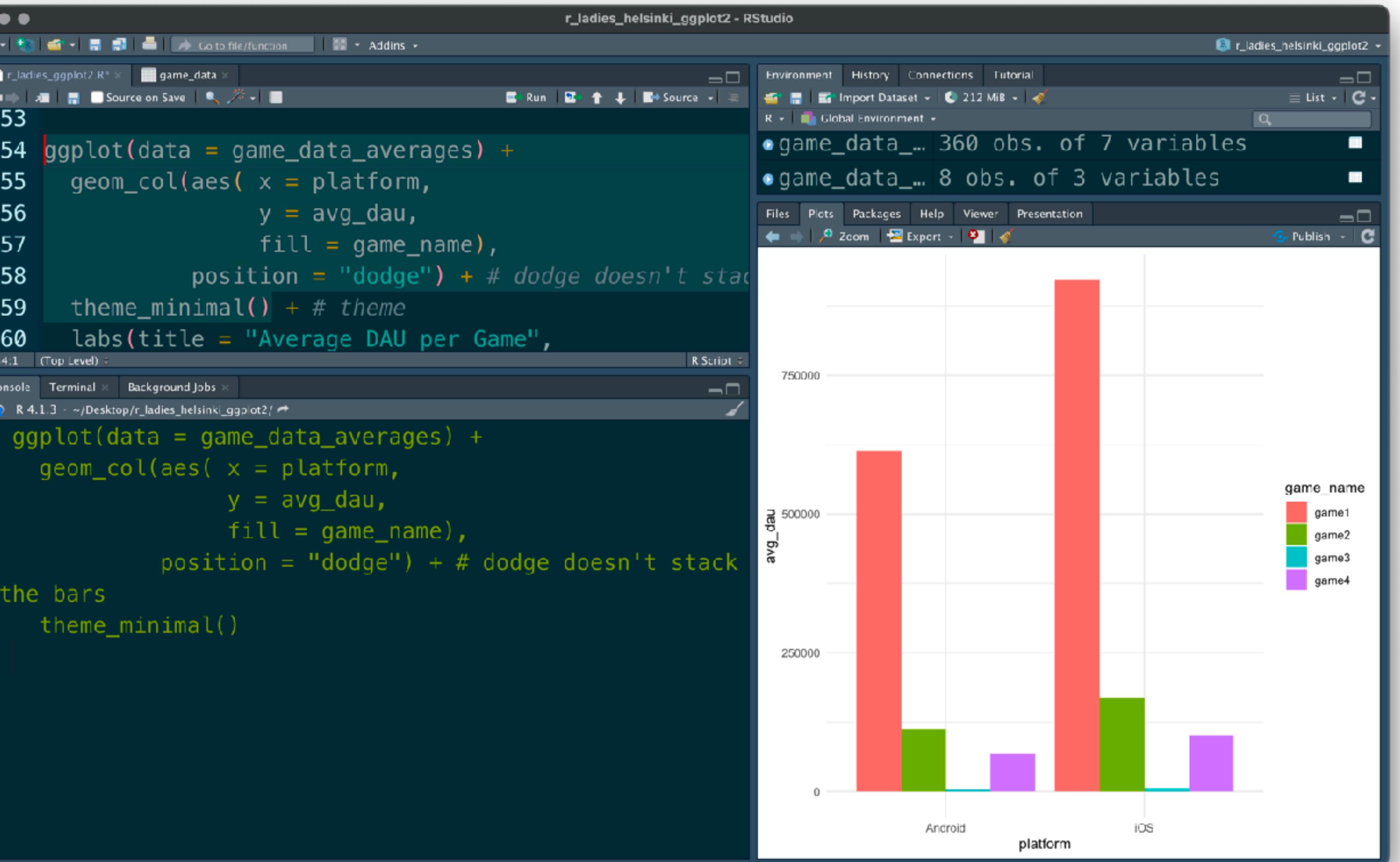
A red rectangular box highlights the line of code "ggplot(data = game_data_averages)".

Bar Chart: add geom



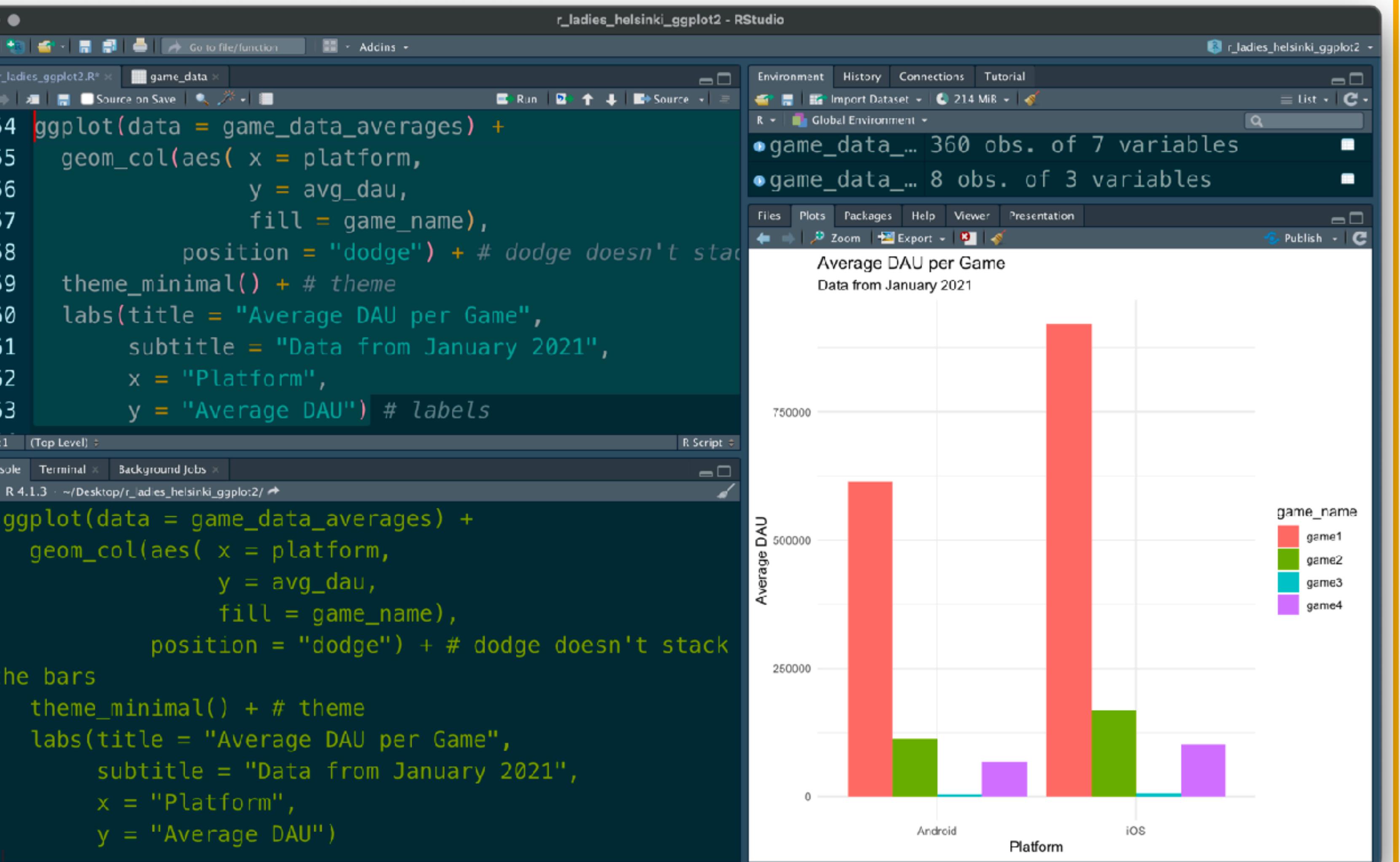
```
ggplot(data = game_data_averages) +
  geom_col(aes( x = platform,
                y = avg_dau,
                fill = game_name),
  position = "dodge")
```

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



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