Introduction to R – Data Visualization

Matt Blyth

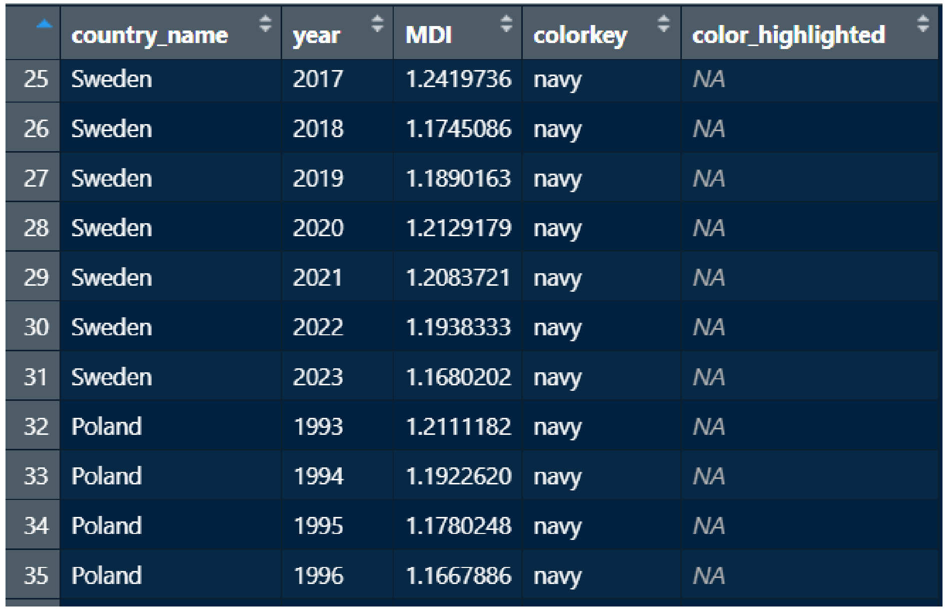
Session 3

Agenda

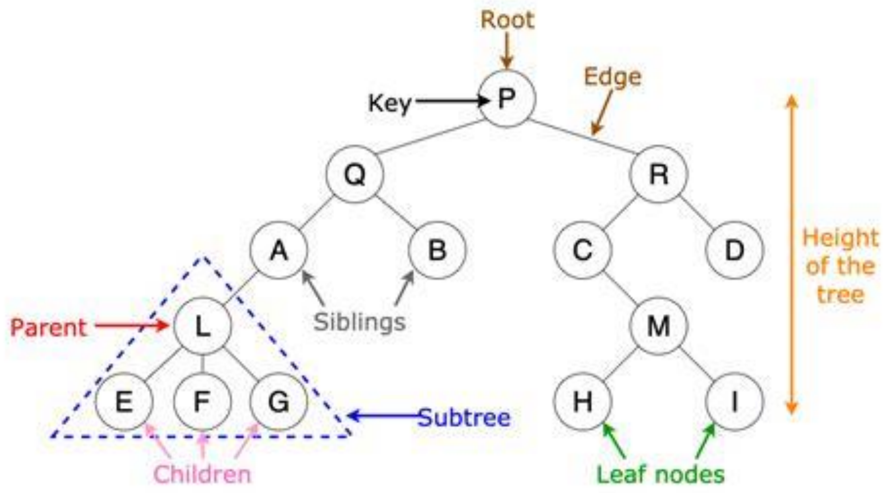
- Data visualization using ggplot2
- Tables
- Practice

Review of tidy data

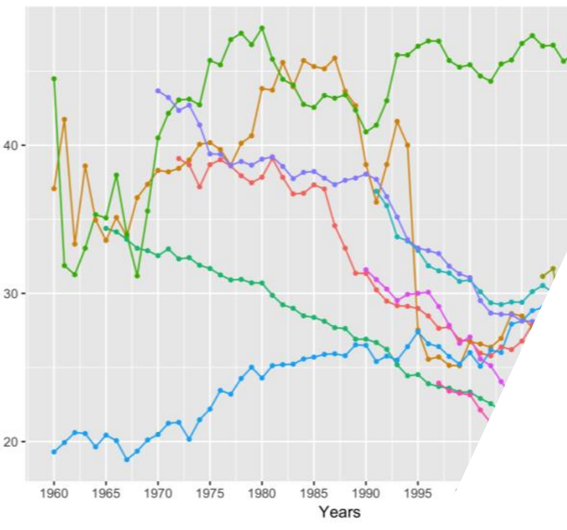
- Variables have their own columns
- Observations have their own rows
- Rectangular data structure
  - E.g. something like a spreadsheet, rather than a tree or other hierarchical data

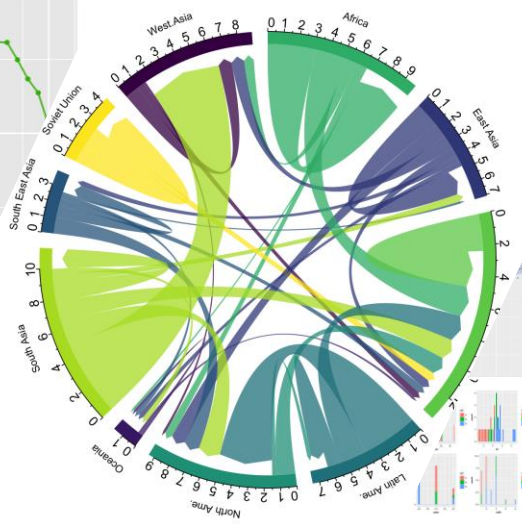


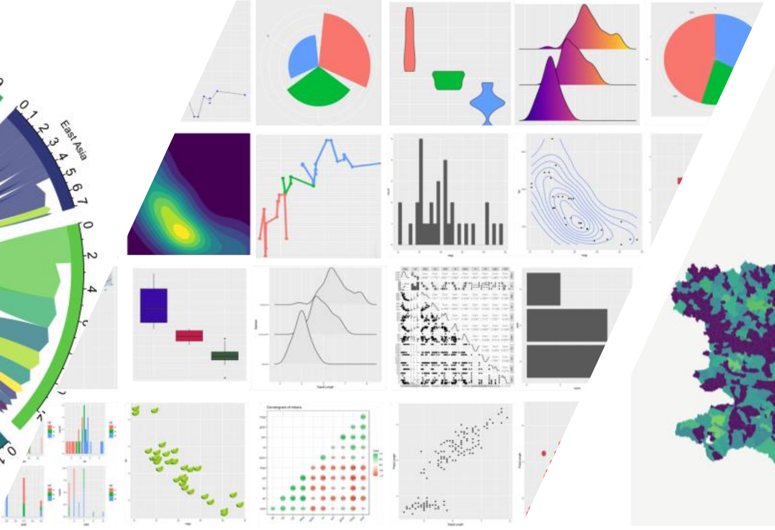
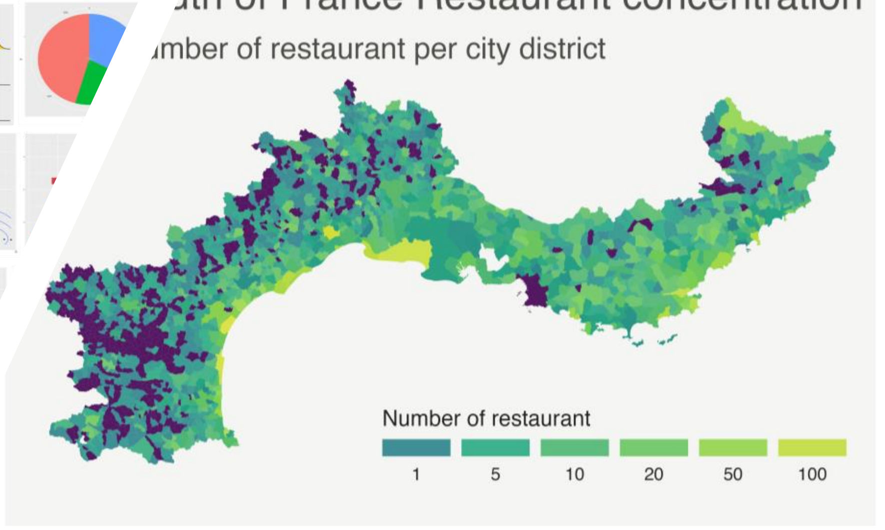
	country_name	year	MDI	colorkey	color_highlighted
25	Sweden	2017	1.2419736	navy	NA
26	Sweden	2018	1.1745086	navy	NA
27	Sweden	2019	1.1890163	navy	NA
28	Sweden	2020	1.2129179	navy	NA
29	Sweden	2021	1.2083721	navy	NA
30	Sweden	2022	1.1938333	navy	NA
31	Sweden	2023	1.1680202	navy	NA
32	Poland	1993	1.2111182	navy	NA
33	Poland	1994	1.1922620	navy	NA
34	Poland	1995	1.1780248	navy	NA
35	Poland	1996	1.1667886	navy	NA



https://towardsdatascience.com





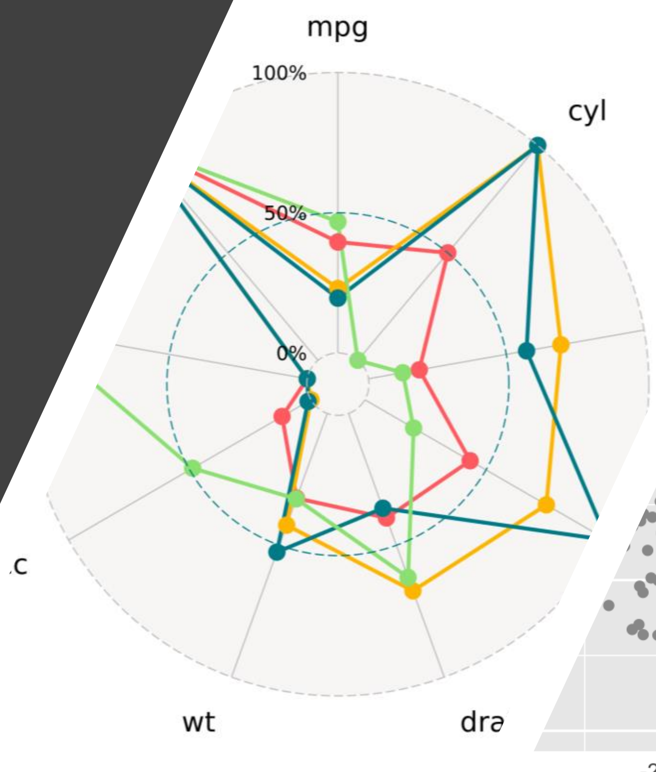
th of France Restaurant concentration
mber of restaurant per city district
Number of restaurant
1
5
10
20
50
100

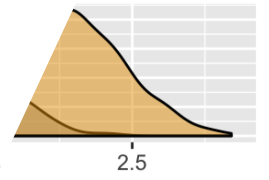
Why visualize?

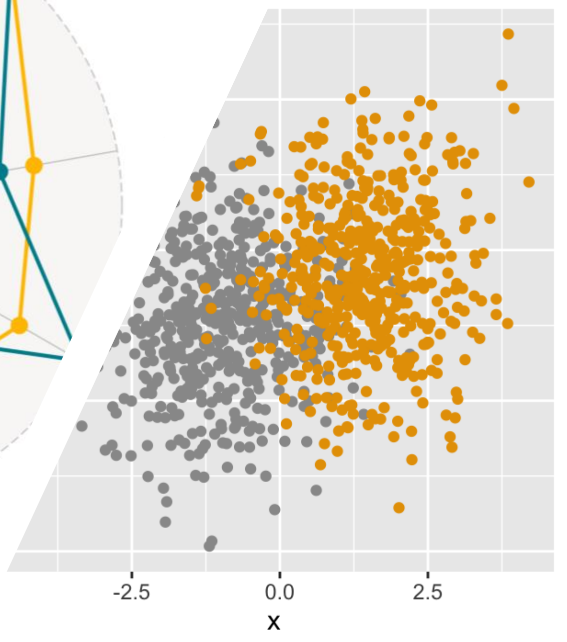
- Communication
- Better understanding
- Stand out – presentation matters

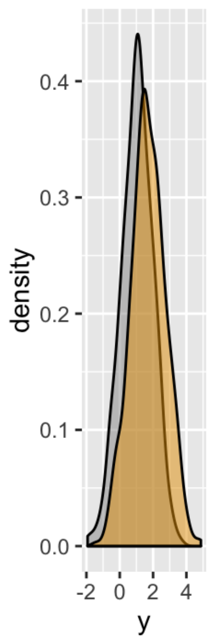
Image sources:

https://opensourcebiology.eu/2022/03/08/a-comprehensive-guide-on-ggplot2-in-r/
https://r-graph-gallery.com/
https://www.storybench.org/getting-started-data-visualization-r-using-ggplot2/
https://www.sthda.com
https://exts.ggplot2.tidyverse.org/ggradar.html









ggplot2

- Commonly used R package, included in tidyverse
- Clean, professional graphics
- Flexible, with high potential for customization
- Many add-ons are available for specific chart types

“Grammar of Graphics”

- At minimum:
  - Data
  - Mapping
  - Layer
- Additional options:
  - Scales
  - Facets
  - Coordinates
  - Theme

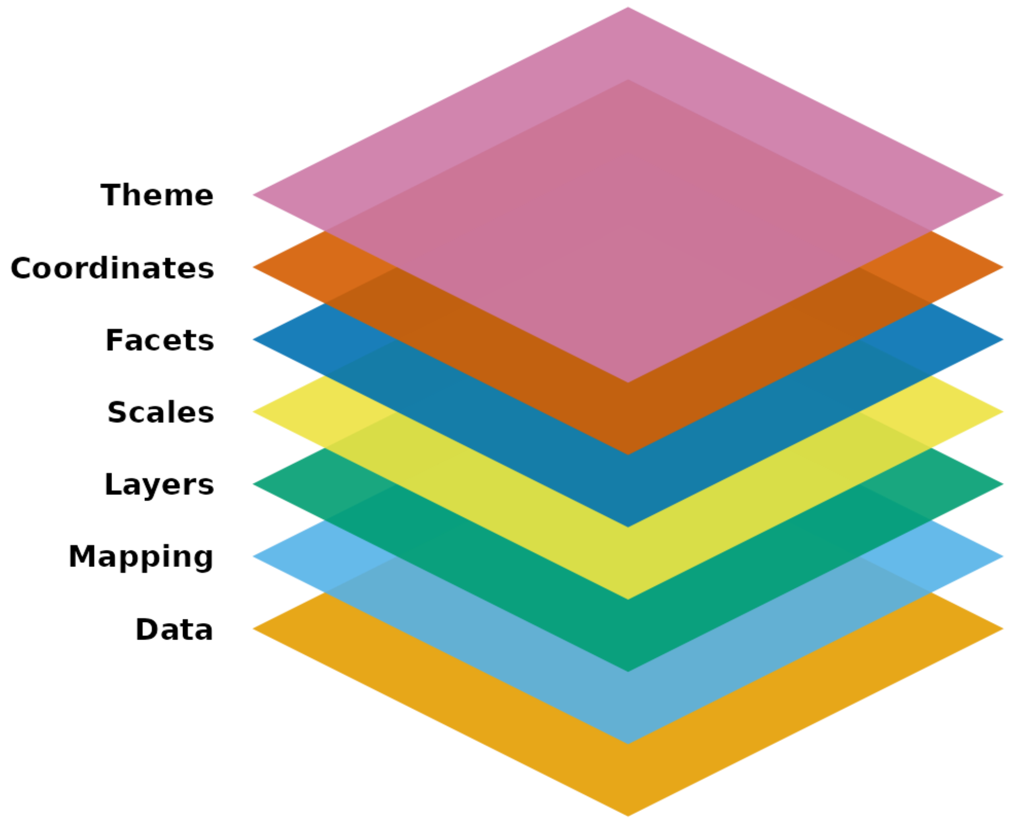
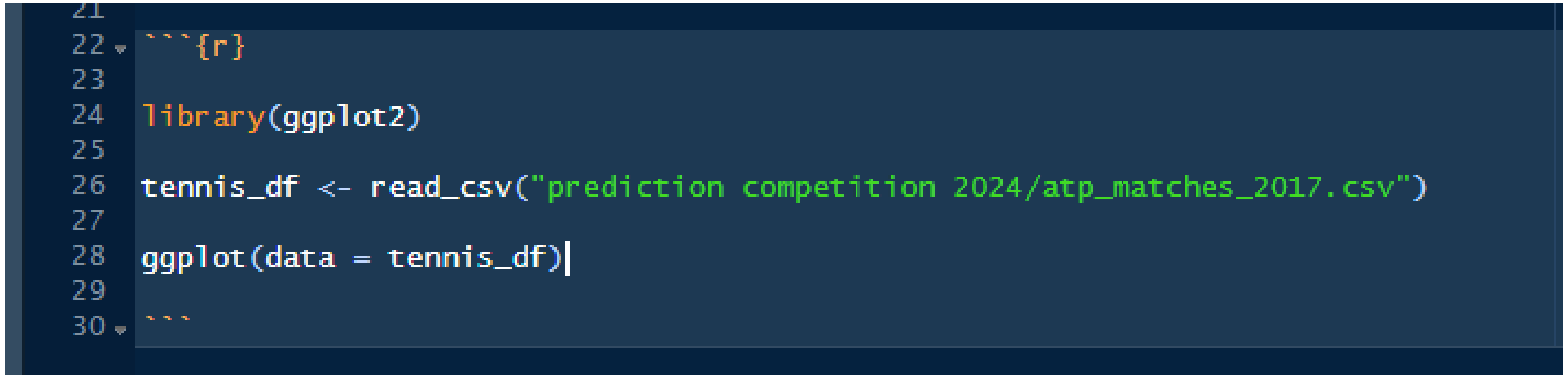


Image source and adapted information for this section: https://ggplot2.tidyverse.org/articles/ggplot2.html

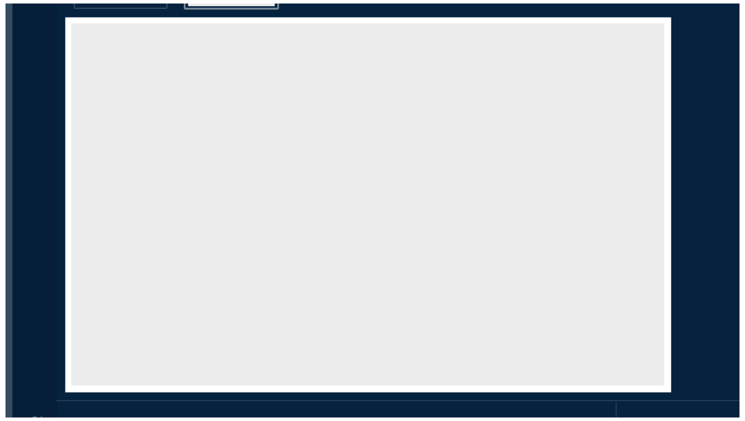
Data

- In general, want data in tidy format

- Often need to restructure based on the parameters of the layers to be used

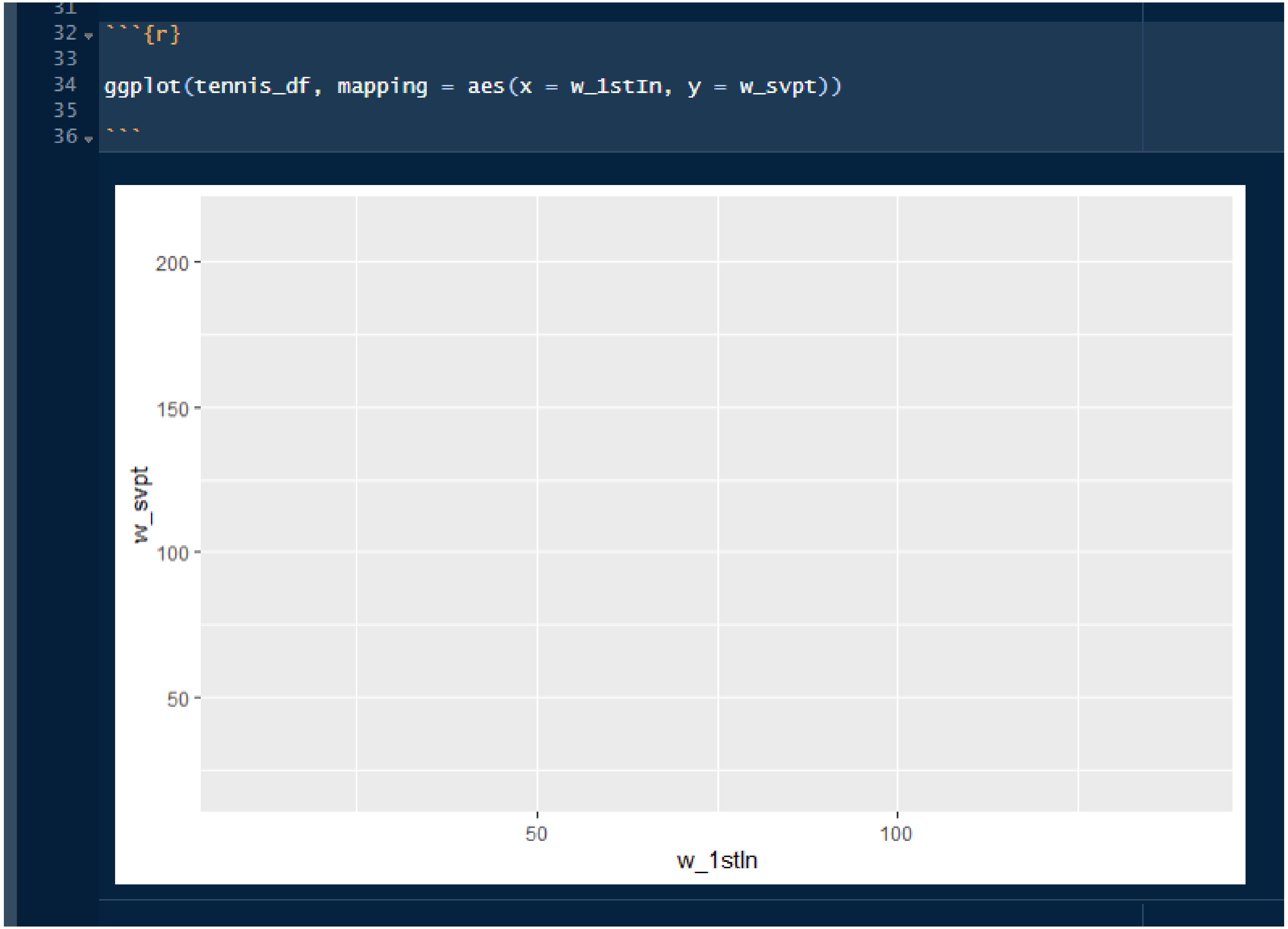


~~~
```{r}

library(ggplot2)

tennis_df <- read_csv("prediction competition 2024/atp_matches_2017.csv")

ggplot(data = tennis_df)

```
~~~

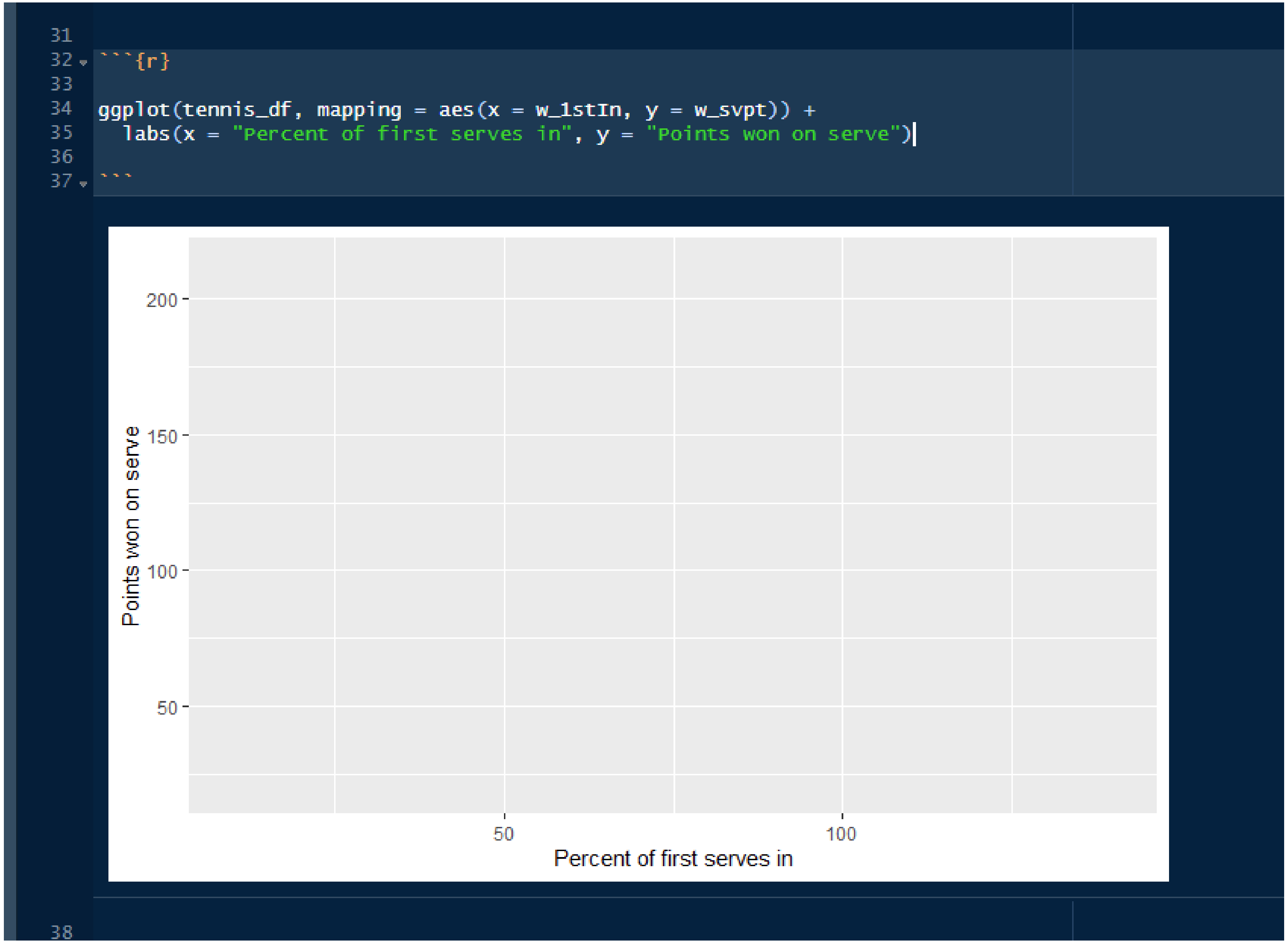


Mapping

- Translate from layout of data to where things will show up in plot
- This includes…
  - x and y (axis)
  - group
  - fill
  - color
  - shape
  - size
- Use aes() function within ggplot()



~~~
```{r}

ggplot(tennis_df, mapping = aes(x = w_1stIn, y = w_svpt))

```
~~~

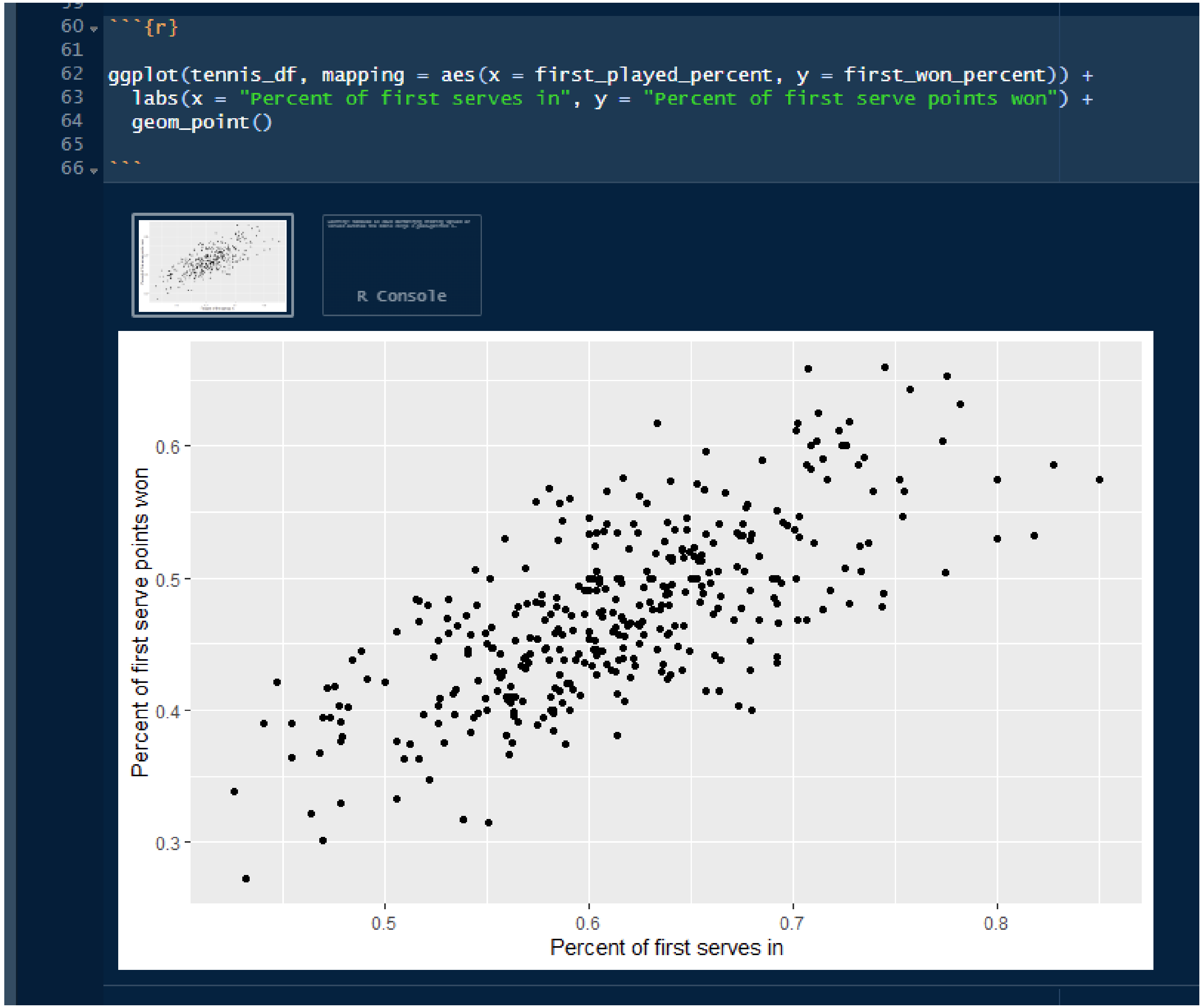


```{r}

ggplot(tennis_df, mapping = aes(x = w_1stIn, y = w_svpt)) +
  labs(x = "Percent of first serves in", y = "Points won on serve")

```

Layers

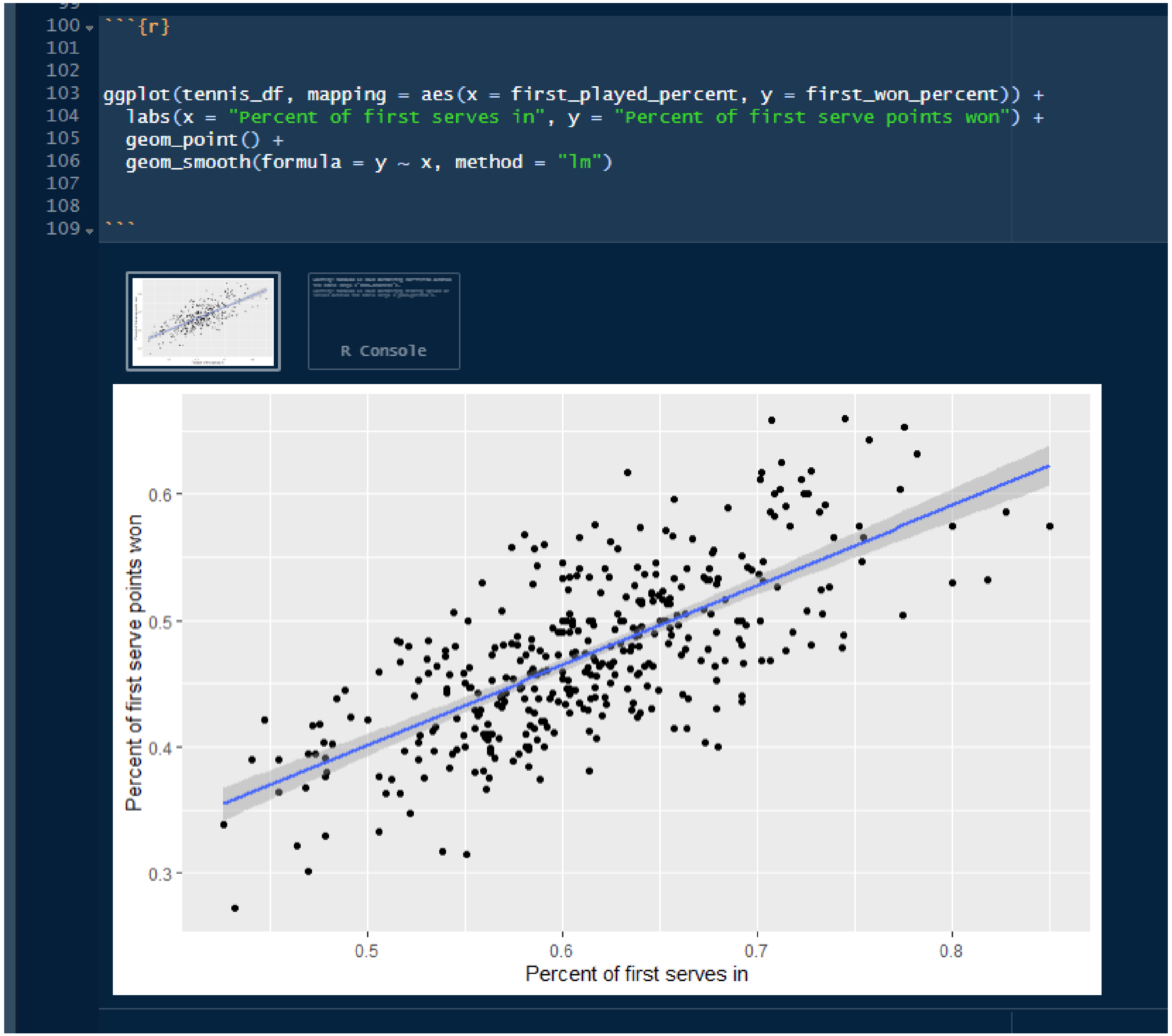
- Geometry
  - geom_ functions
- Transformations
  - stat_ functions
- Position
  - position =

geom_point()



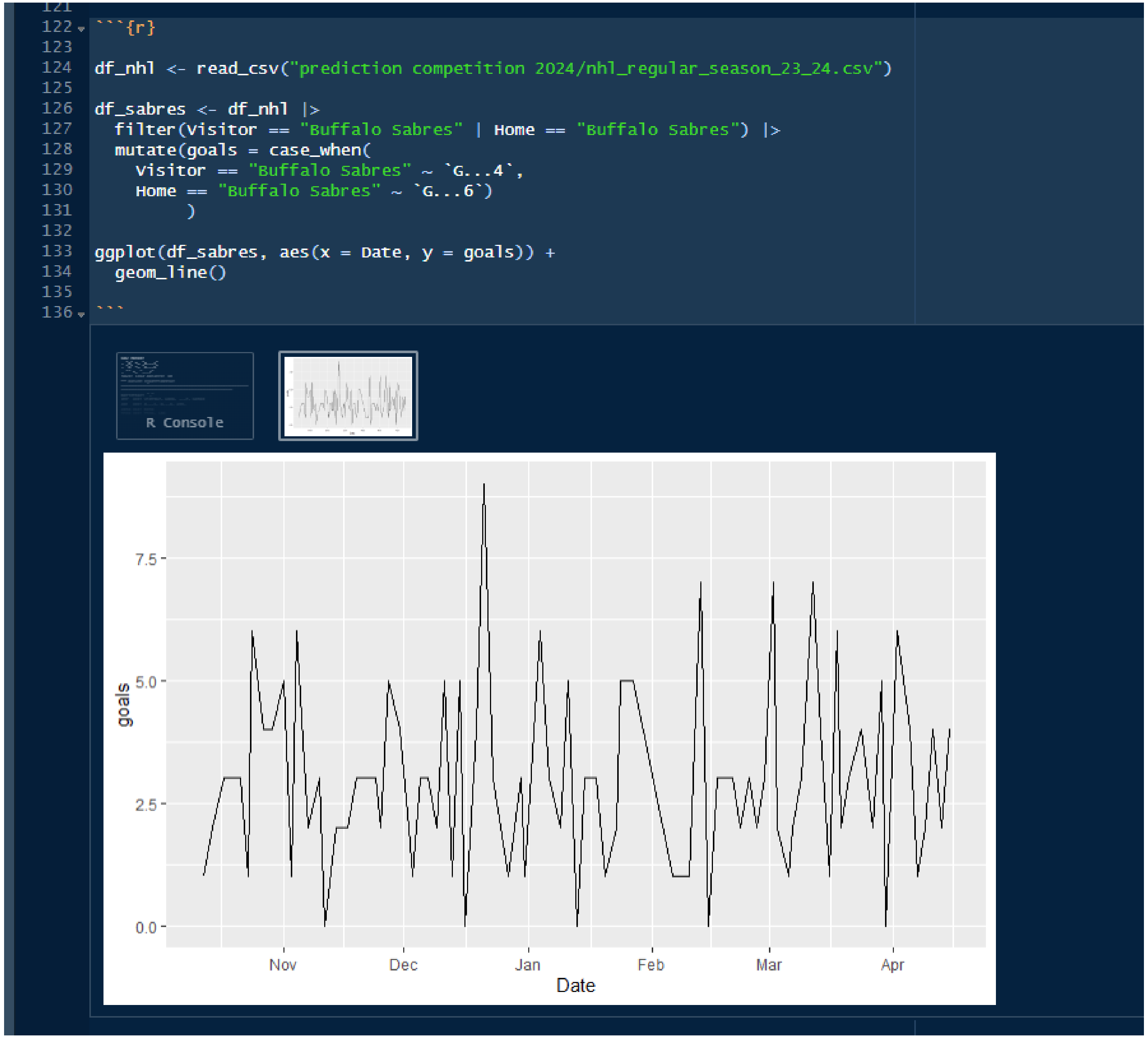
```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point()

```

geom_smooth()



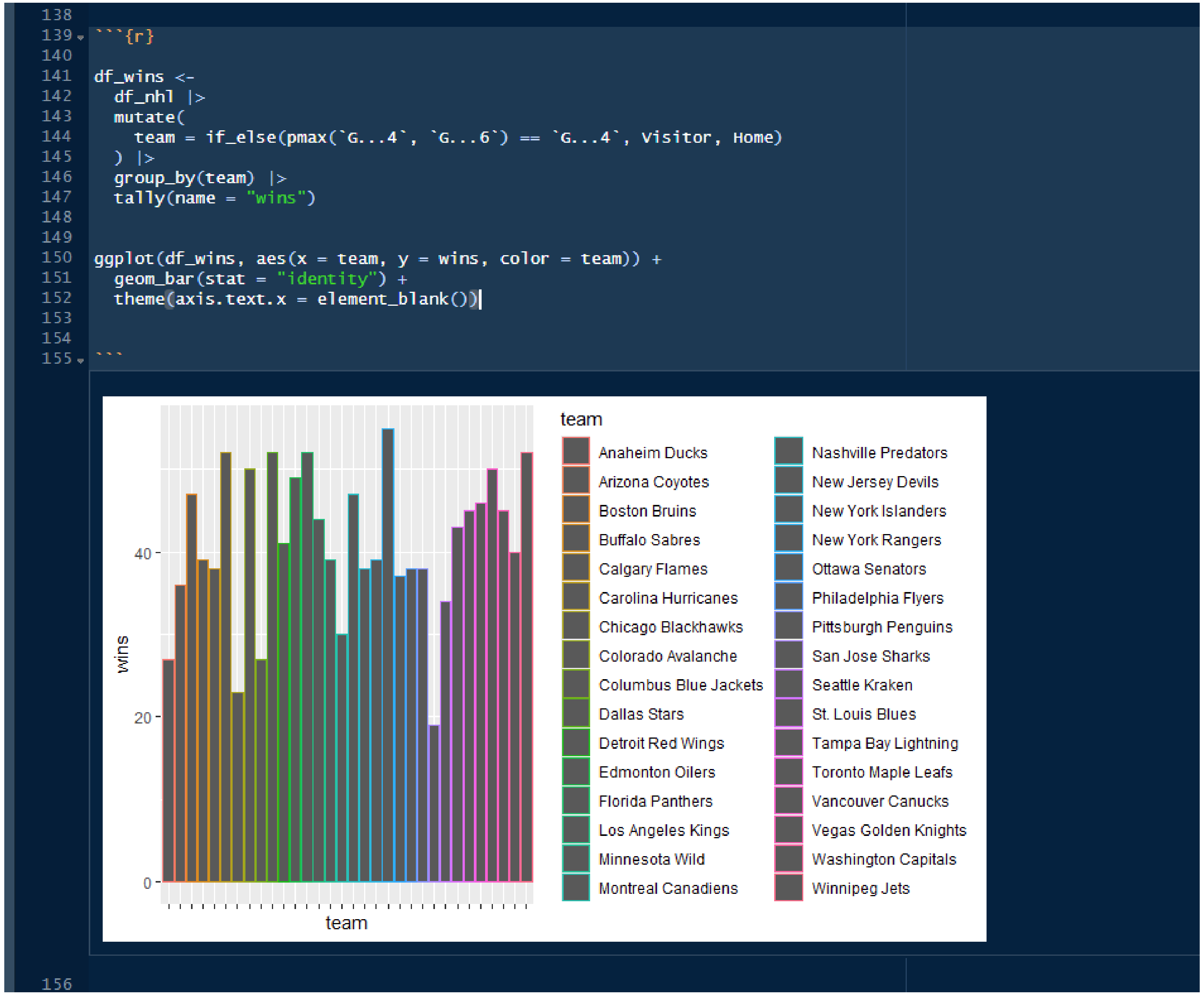
~~~
```{r}


ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "lm")


```
~~~

geom_line()



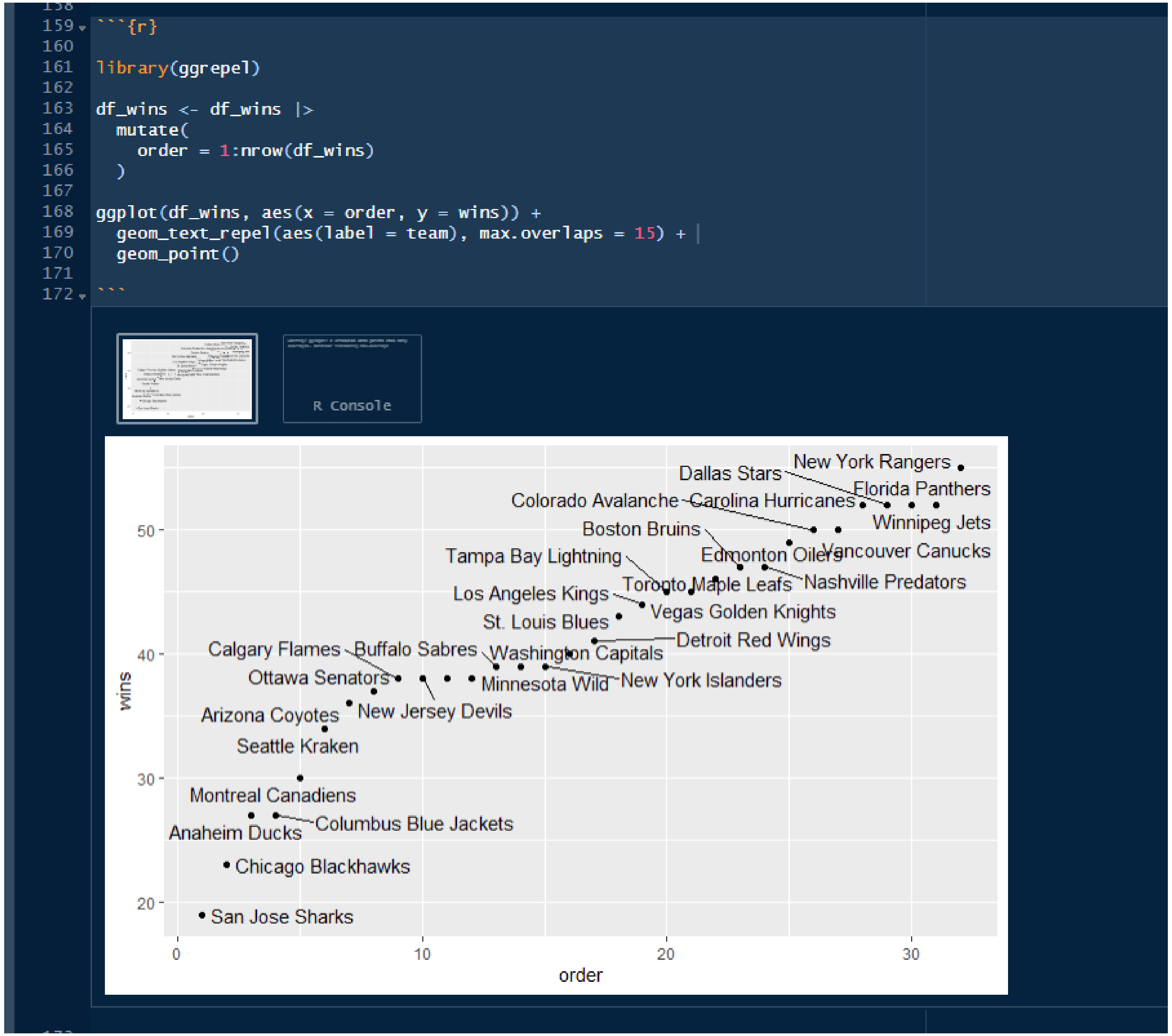
~~~r
```{r}

df_nhl <- read_csv("prediction competition 2024/nhl_regular_season_23_24.csv")

df_sabres <- df_nhl |>
  filter(Visitor == "Buffalo Sabres" | Home == "Buffalo Sabres") |>
  mutate(goals = case_when(
    Visitor == "Buffalo Sabres" ~ `G...4`,
    Home == "Buffalo Sabres" ~ `G...6`)
         )

ggplot(df_sabres, aes(x = Date, y = goals)) +
  geom_line()

~~~
```
```

geom_bar()



~~~
```{r}

df_wins <-
  df_nhl |>
  mutate(
    team = if_else(pmax(`G...4`, `G...6`) == `G...4`, Visitor, Home)
  ) |>
  group_by(team) |>
  tally(name = "wins")


ggplot(df_wins, aes(x = team, y = wins, color = team)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_blank())


```
~~~

geom_text()



~~~
```{r}

library(ggrepel)

df_wins <- df_wins |>
  mutate(
    order = 1:nrow(df_wins)
  )

ggplot(df_wins, aes(x = order, y = wins)) +
  geom_text_repel(aes(label = team), max.overlaps = 15) +
  geom_point()

```
~~~

Other geoms

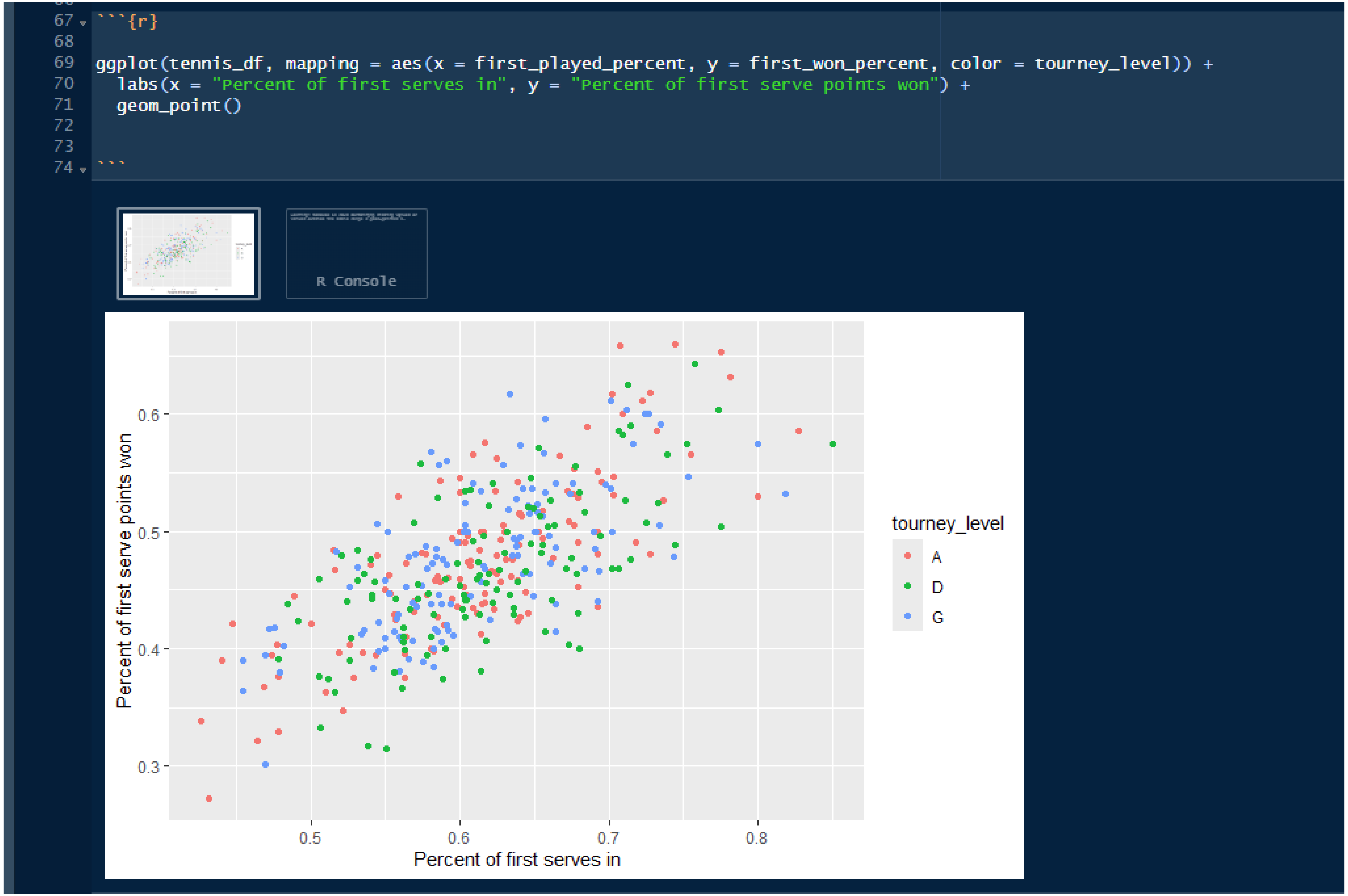
- geom_density()
- geom_area()
- geom_label()
- geom_segment()
- etc

Scales

- Colors, axes, position, shape, size

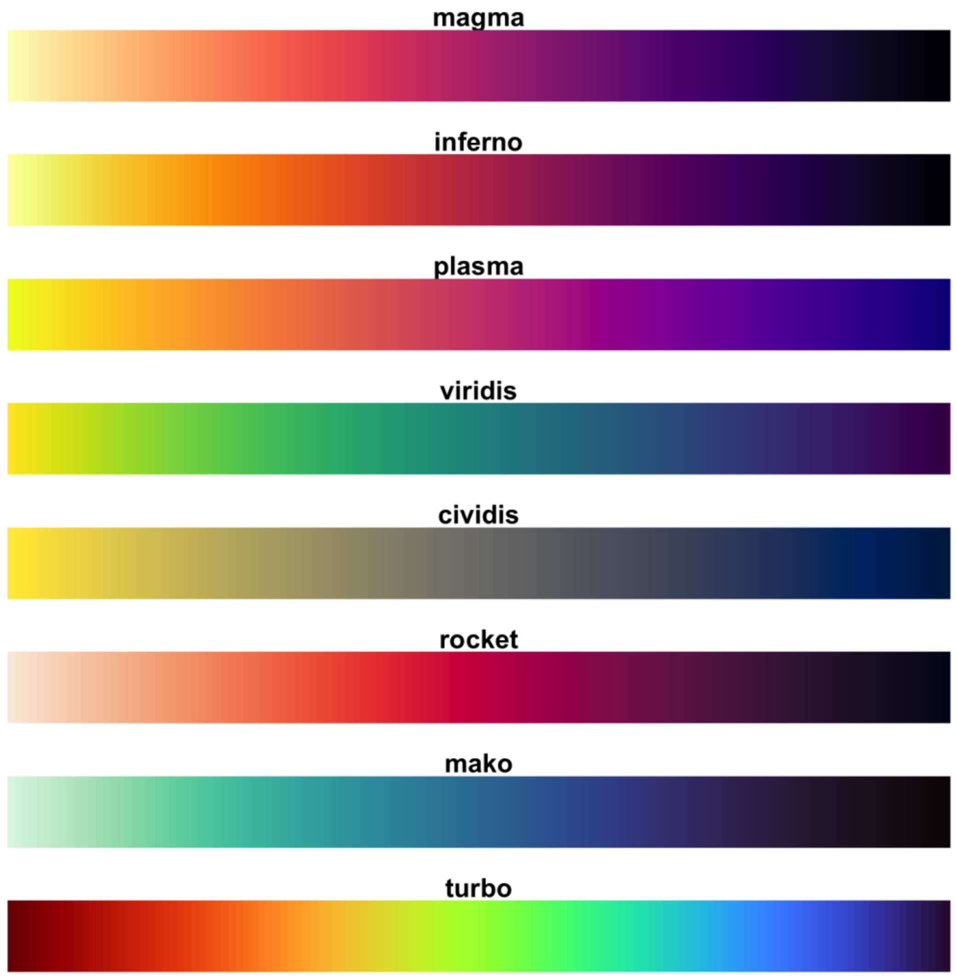
- scale_AES_TYPE ()
  - Where aes() goes back to the mapping function at the beginning, could be color, x, y ,etc
  - Type depends on what kind of scale you want, such as discrete, continuous, etc
  - brewer (discrete) and distiller (continuous) have a variety of preset palettes

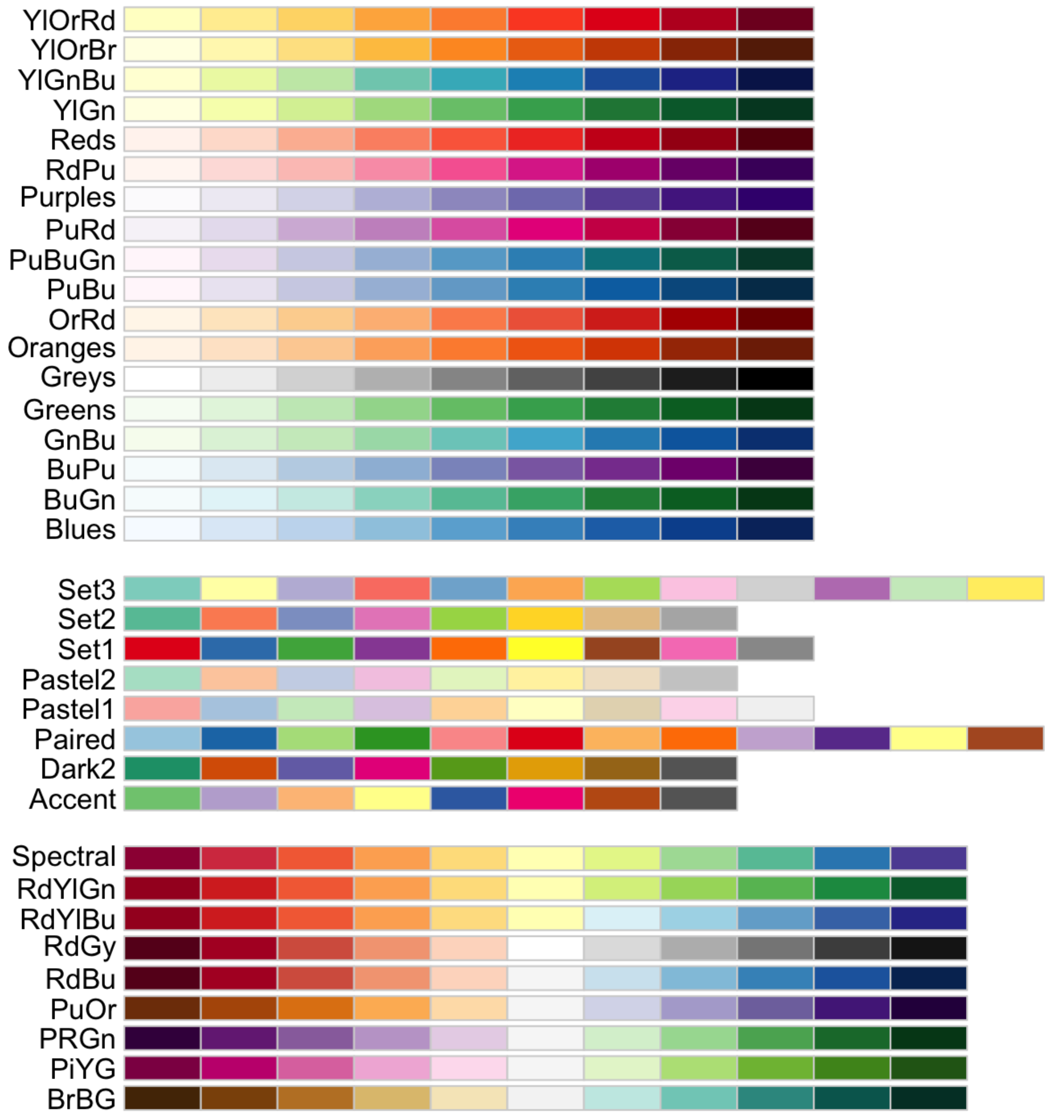
- This can get complicated and/or confusing
  - Many resources available to help

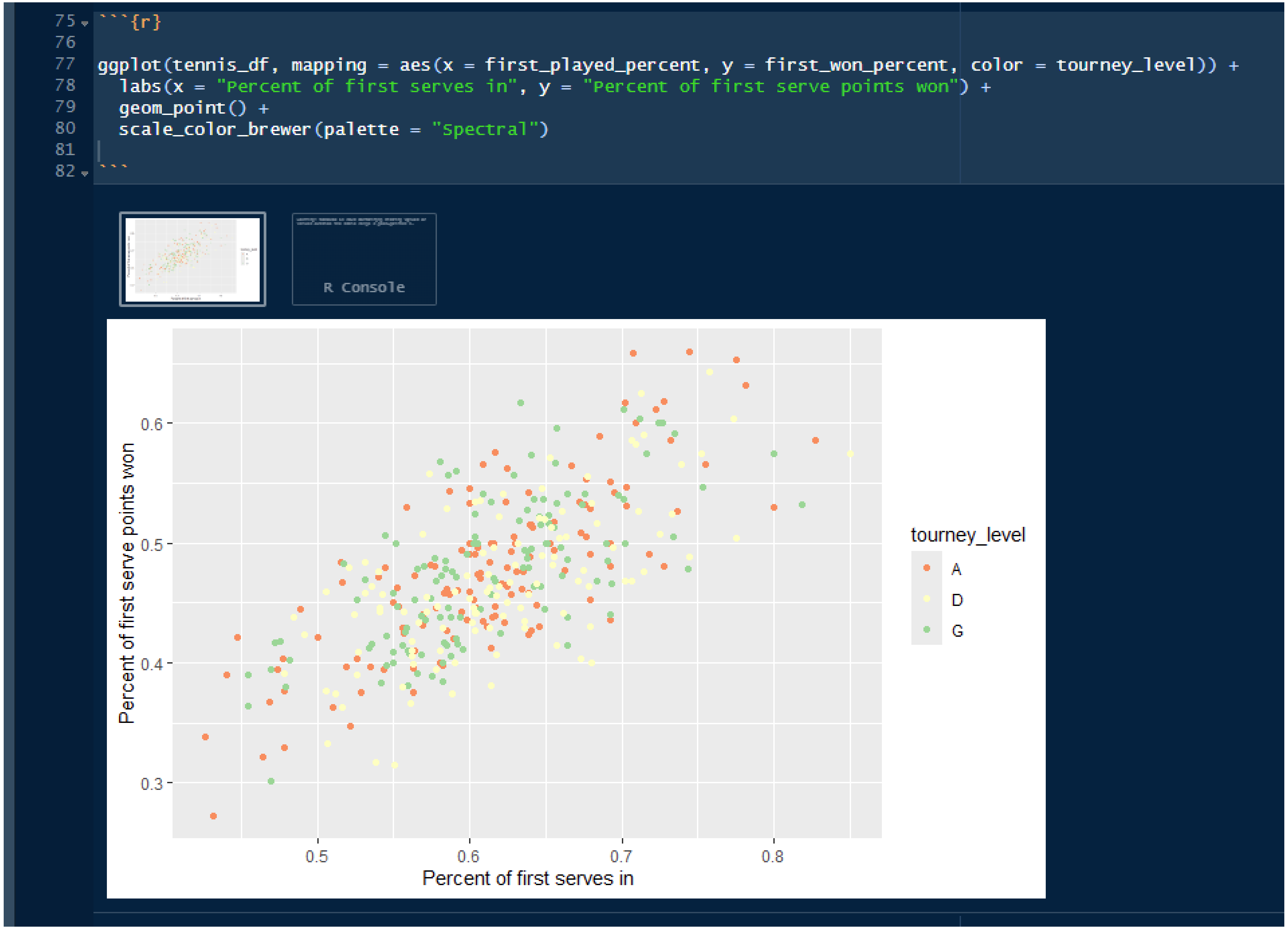


~~~
```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent, color = tourney_level)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point()


```
~~~

Scales: color palettes



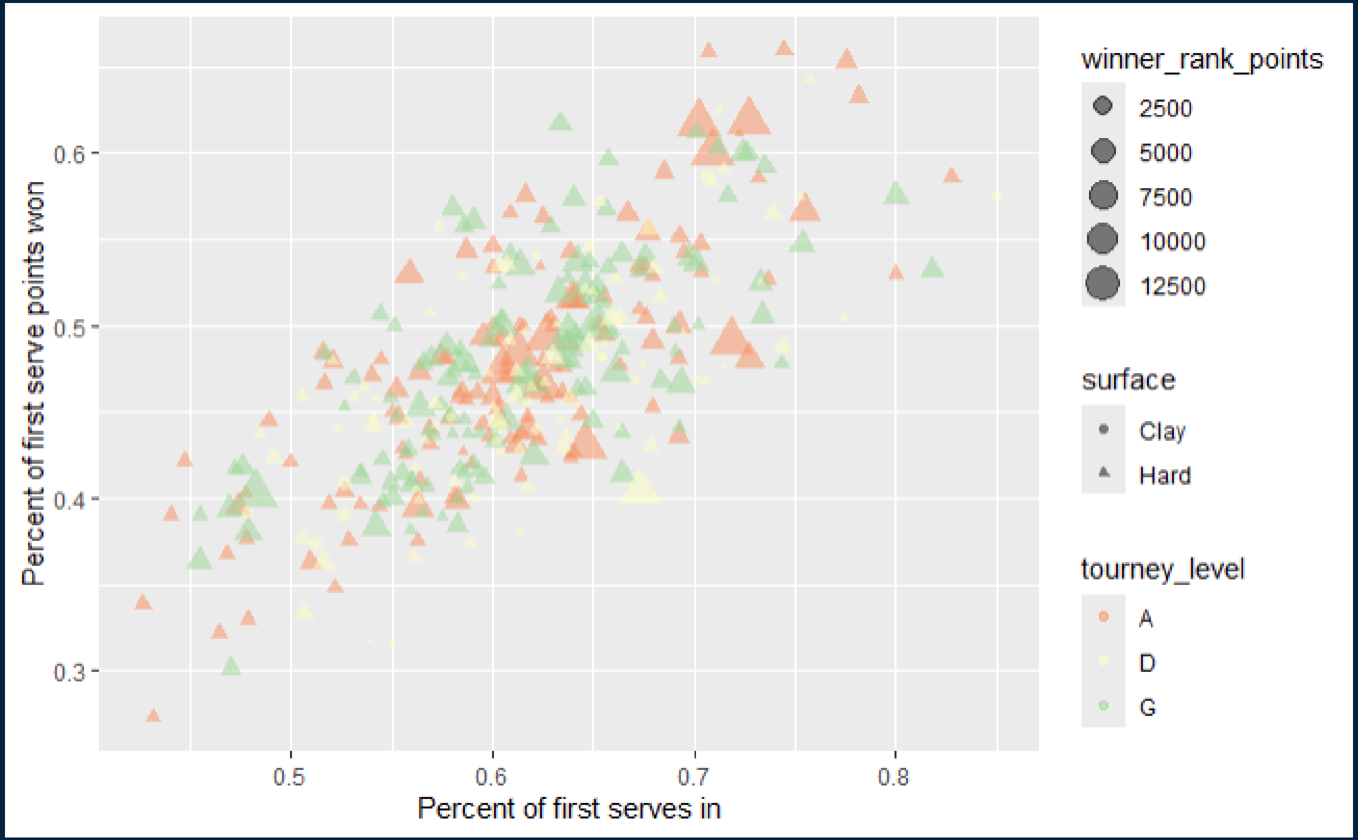




```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent, color = tourney_level)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point() +
  scale_color_brewer(palette = "Spectral")

```

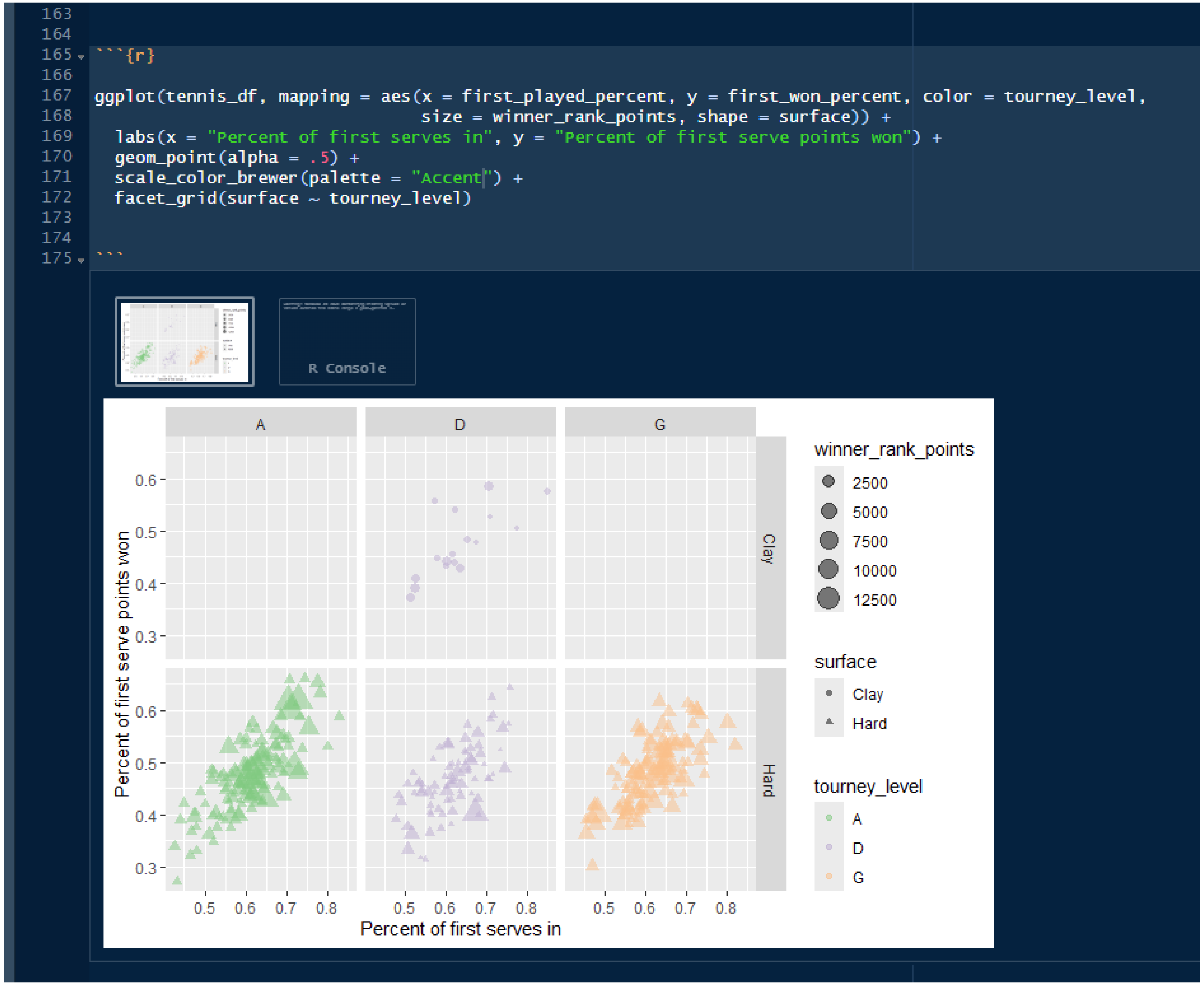
```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent, color = tourney_level,
                                size = winner_rank_points, shape = surface)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point(alpha = .5) +
  scale_color_brewer(palette = "Spectral")

```

R Console



Facets

- Break plots up into multiple panels
- Facets indicated in facet_{something}() functions



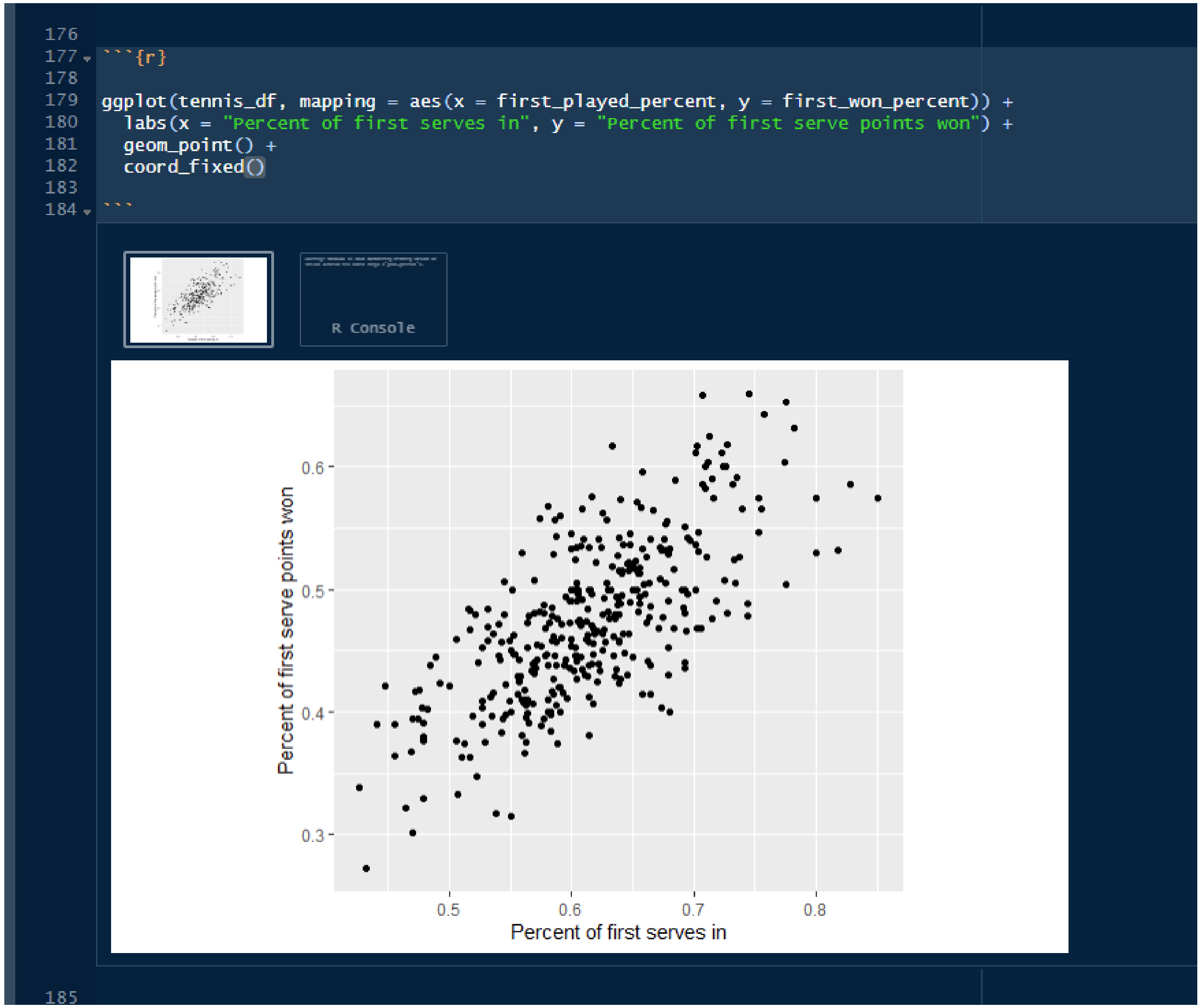
~~~
```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent, color = tourney_level,
                                size = winner_rank_points, shape = surface)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point(alpha = .5) +
  scale_color_brewer(palette = "Accent") +
  facet_grid(surface ~ tourney_level)


```
~~~

Coordinates

- Can be used for mapping, non-Cartesian coordinate systems

- Helps with some display

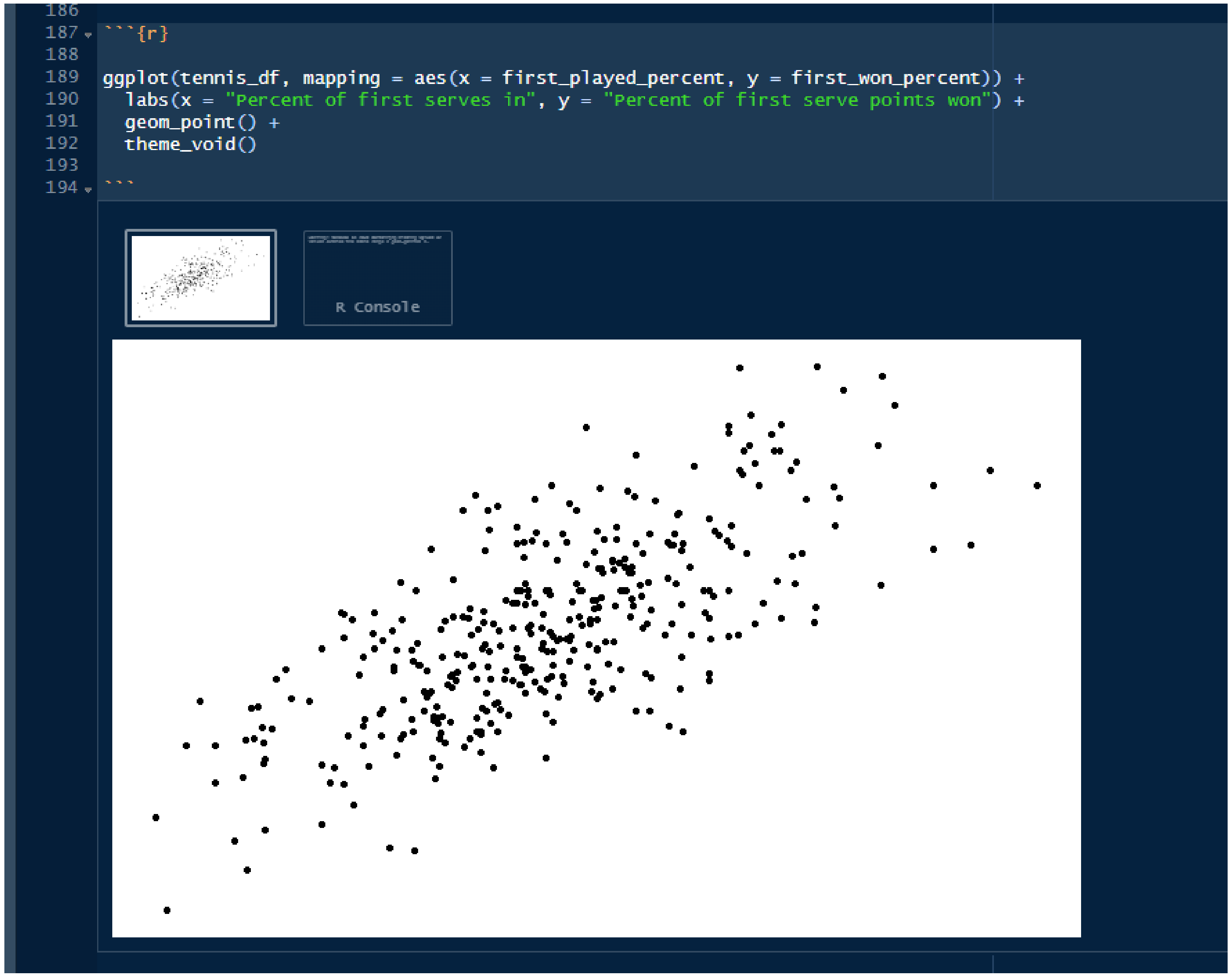
- In image, fixed aspect ratio



~~~
```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point() +
  coord_fixed()

```
~~~

Theme

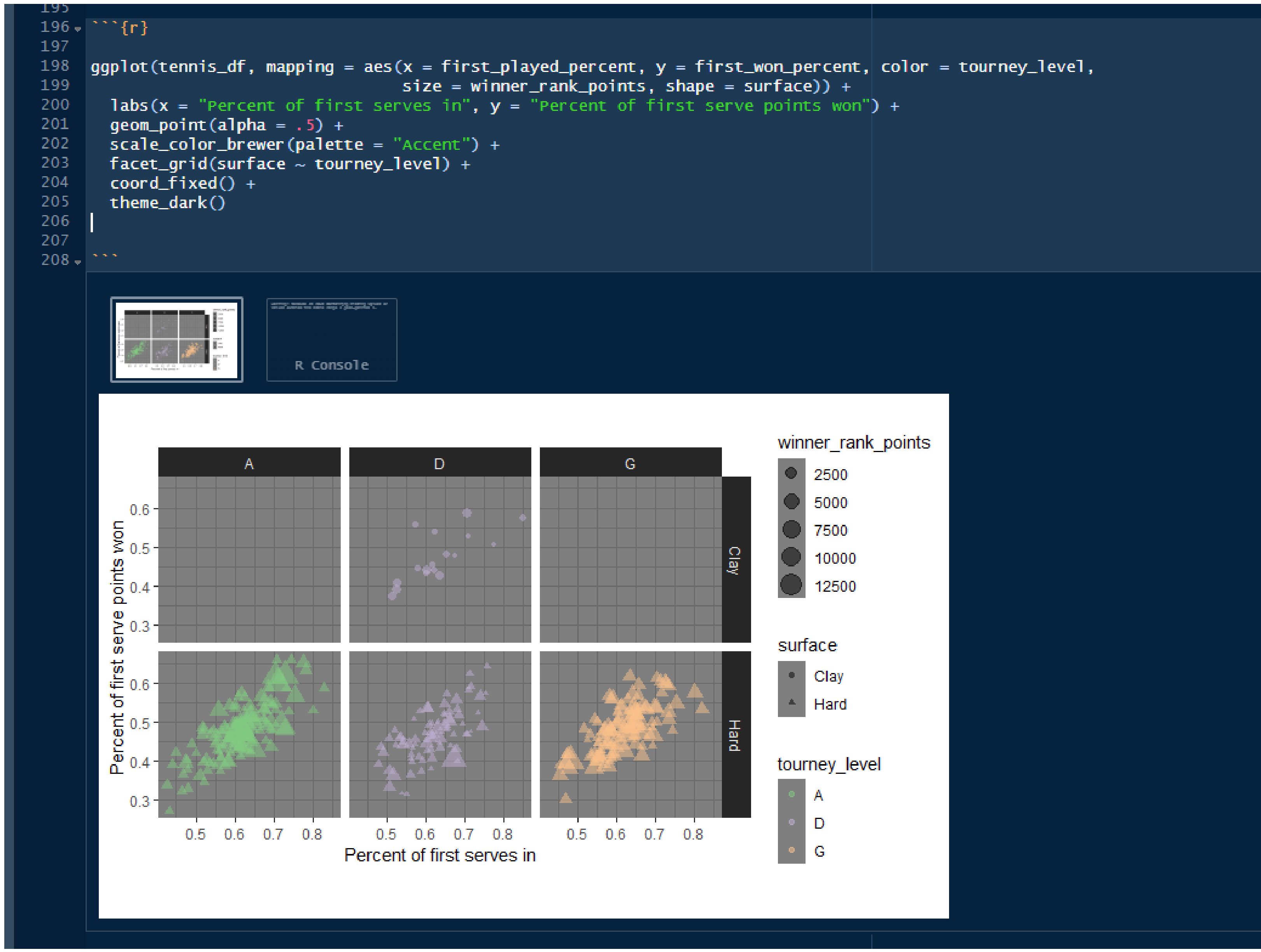
- Changes overall appearance
- Many prebuilt themes, can also create your own
- More fine control over what is displayed



~~~
```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point() +
  theme_void()

```
~~~

Putting everything together

- Each piece builds on the previous ones, allowing for complex graphics

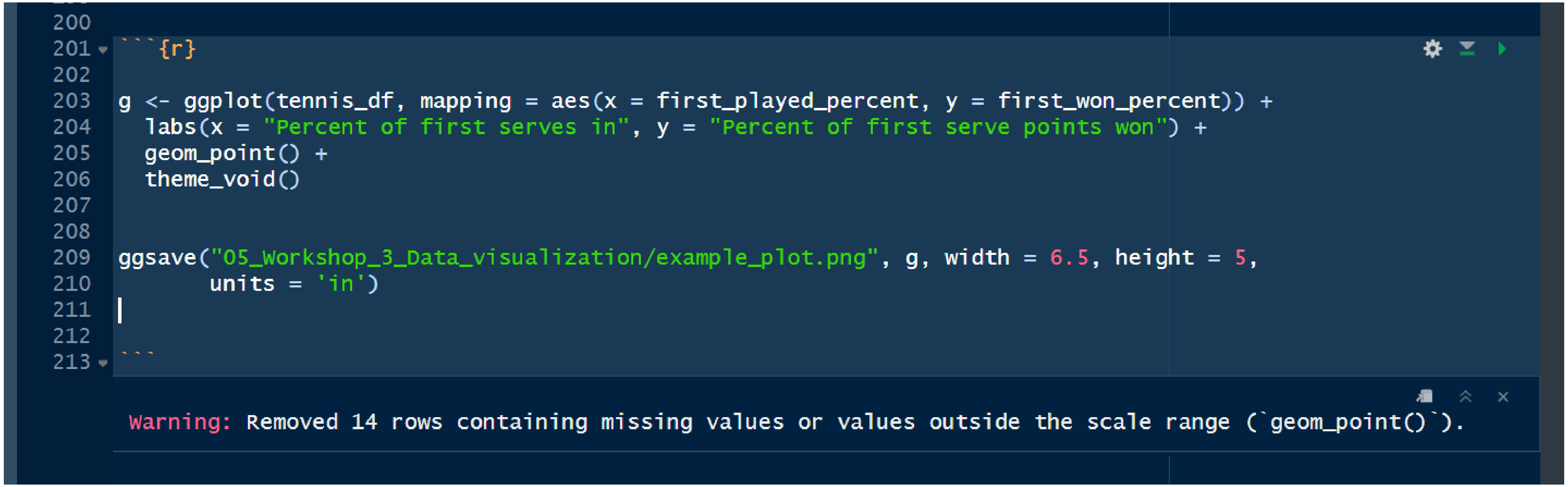


~~~
```{r}

ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent, color = tourney_level,
                                size = winner_rank_points, shape = surface)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point(alpha = .5) +
  scale_color_brewer(palette = "Accent") +
  facet_grid(surface ~ tourney_level) +
  coord_fixed() +
  theme_dark()


```
~~~

Saving plots

- Plots can be saved in R as objects

- ggsave() function will save it to the specified location in your computer



~~~
```{r}

g <- ggplot(tennis_df, mapping = aes(x = first_played_percent, y = first_won_percent)) +
  labs(x = "Percent of first serves in", y = "Percent of first serve points won") +
  geom_point() +
  theme_void()


ggsave("05_Workshop_3_Data_visualization/example_plot.png", g, width = 6.5, height = 5,
       units = 'in')


```
~~~

```
Warning: Removed 14 rows containing missing values or values outside the scale range (`geom_point()`).
```

LLMs

- Chat GPT or similar models can be helpful for setting up an outline when getting started- but be careful to understand how the different pieces work if you want to be able to fully control the output
- Also can be limiting – you still need a vision for how it should turn out!

Tables

- How to report?
- In R
- LaTeX
- Word, etc

In RStudio

- Quarto
- R Notebook
- knittr and creating pdfs from R files
- Regression – stargazer
- General tables from dataframe – kable, kableExtra

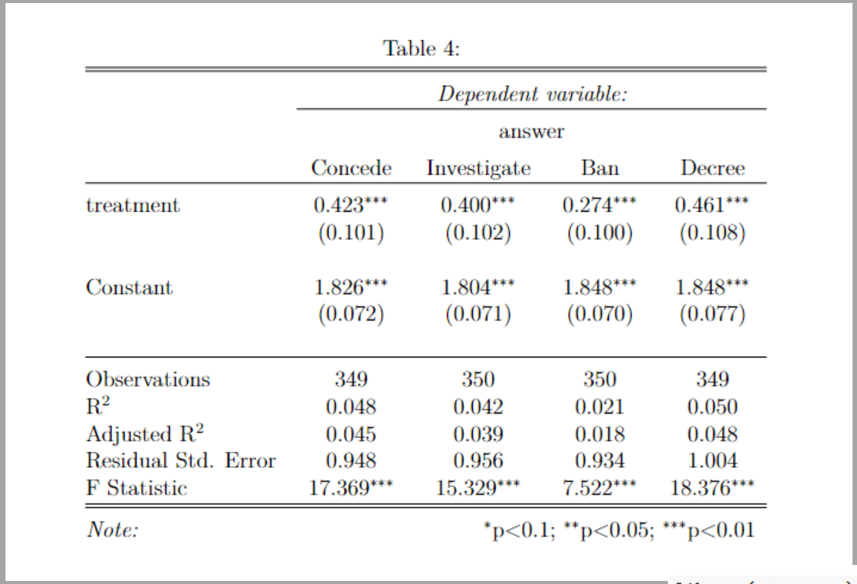
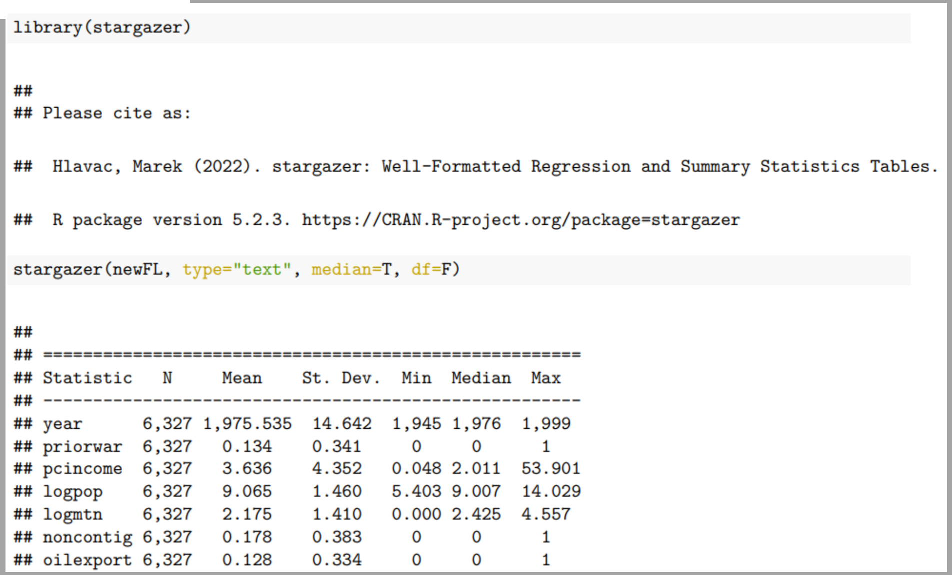


Table 4:

	Dependent variable:			
	answer			
	Concede	Investigate	Ban	Decree
treatment	0.423***	0.400***	0.274***	0.461***
	(0.101)	(0.102)	(0.100)	(0.108)
Constant	1.826***	1.804***	1.848***	1.848***
	(0.072)	(0.071)	(0.070)	(0.077)
Observations	349	350	350	349
R^2	0.048	0.042	0.021	0.050
Adjusted R^2	0.045	0.039	0.018	0.048
Residual Std. Error	0.948	0.956	0.934	1.004
F Statistic	17.369***	15.329***	7.522***	18.376***

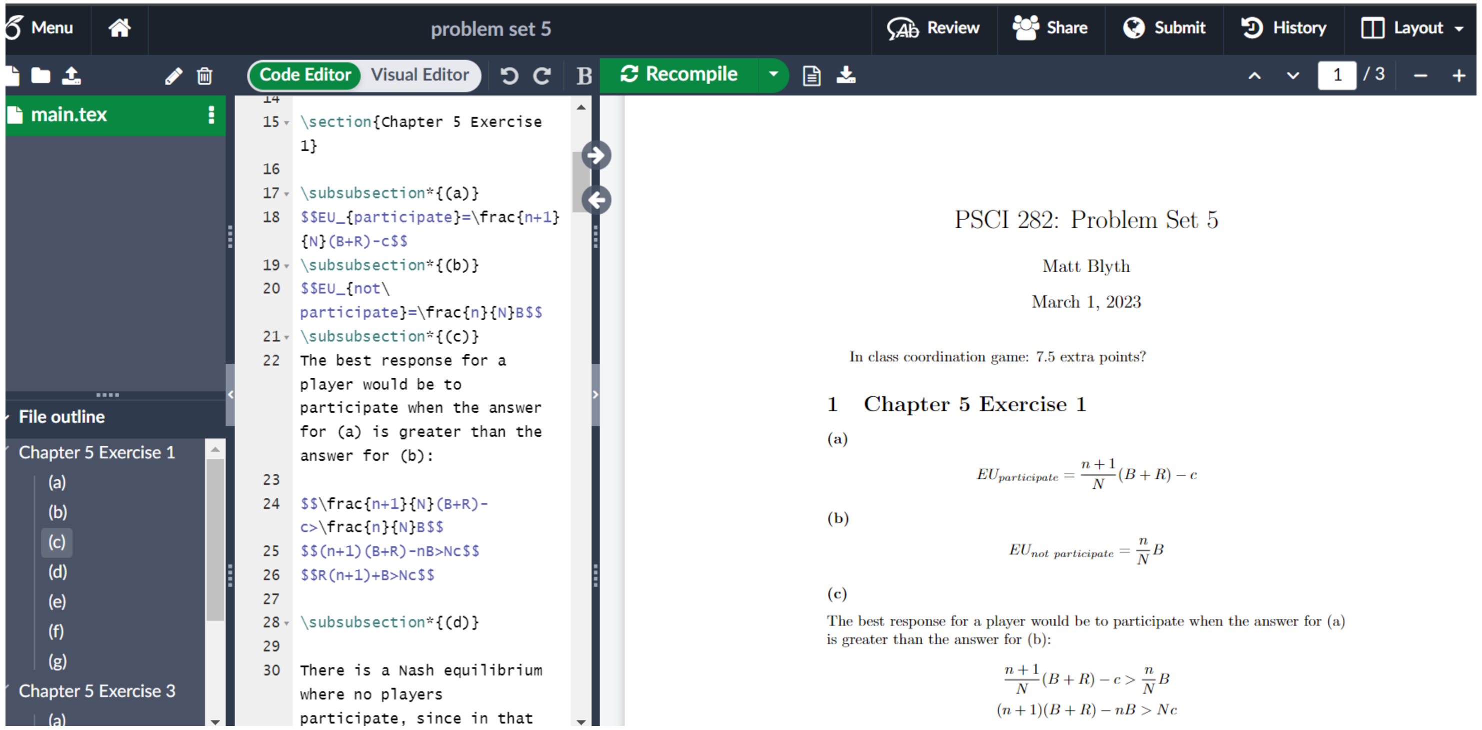
Note: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$



```
library(stargazer)

##
## Please cite as:

##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

stargazer(newFL, type="text", median=T, df=F)

##
## =======================================================
## Statistic   N     Mean    St. Dev.  Min  Median  Max
## -------------------------------------------------------
## year      6,327 1,975.535  14.642  1,945 1,976  1,999
## priorwar  6,327   0.134    0.341     0     0      1
## pcincome  6,327   3.636    4.352   0.048 2.011  53.901
## logpop    6,327   9.065    1.460   5.403 9.007  14.029
## logmtn    6,327   2.175    1.410   0.000 2.425  4.557
## noncontig 6,327   0.178    0.383     0     0      1
## oilexport 6,327   0.128    0.334     0     0      1
```

LaTeX

- Easiest to use Overleaf
  - or .Rmd, .qmd
- Well integrated with other platforms



```latex
\section{Chapter 5 Exercise 1}

\subsubsection*{(a)}
$$EU_{participate}=\frac{n+1}{N}(B+R)-c$$
\subsubsection*{(b)}
$$EU_{not\ participate}=\frac{n}{N}B$$
\subsubsection*{(c)}
The best response for a player would be to participate when the answer for (a) is greater than the answer for (b):

$$\frac{n+1}{N}(B+R)-c>\frac{n}{N}B$$
$$(n+1)(B+R)-nB>Nc$$
$$R(n+1)+B>Nc$$

\subsubsection*{(d)}

There is a Nash equilibrium where no players participate, since in that
```

PSCI 282: Problem Set 5

Matt Blyth

March 1, 2023

In class coordination game: 7.5 extra points?

1 Chapter 5 Exercise 1

(a)

$$EU_{participate} = \frac{n+1}{N}(B+R) - c$$

(b)

$$EU_{not\ participate} = \frac{n}{N}B$$

(c)

The best response for a player would be to participate when the answer for (a) is greater than the answer for (b):

$$\frac{n+1}{N}(B+R) - c > \frac{n}{N}B$$
$$(n+1)(B+R) - nB > Nc$$

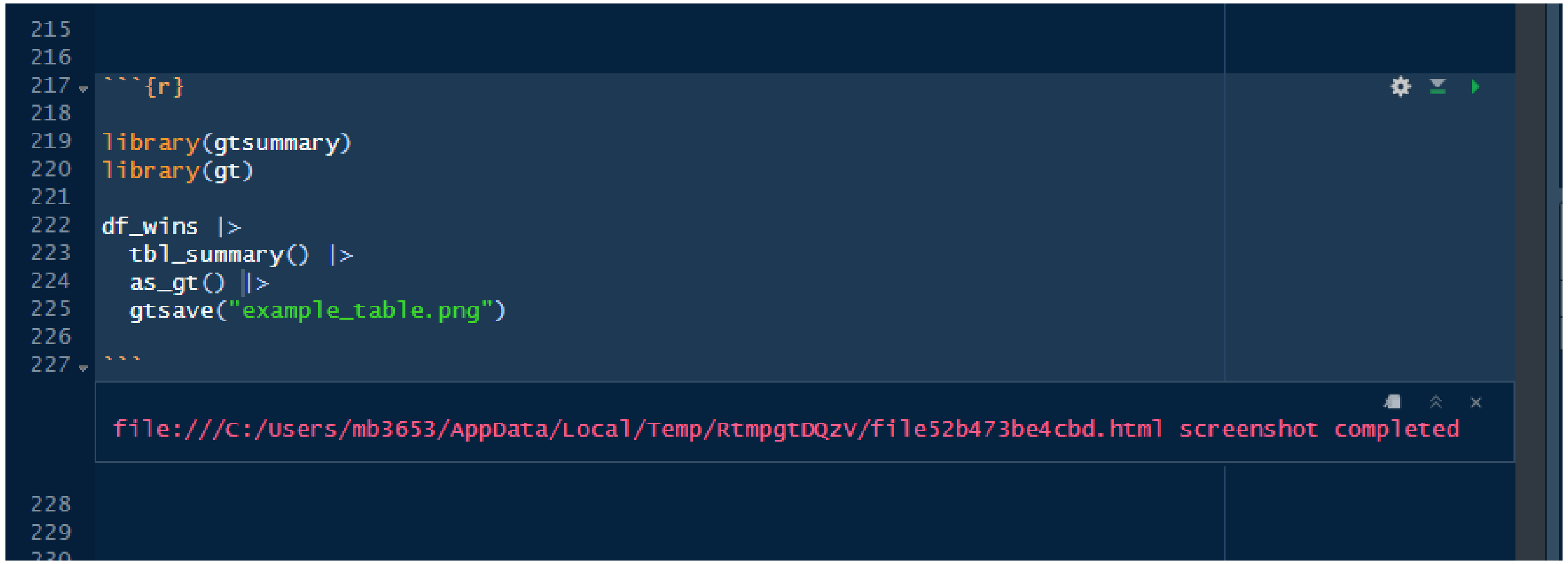
- High quality formatting, easier to type math
- Takes some getting used to, but many resources are available

Word

- In R, save plot as an image (png, etc)

- Then add to word or other program

- Useful packages: gtsummary/gt, modelsummary

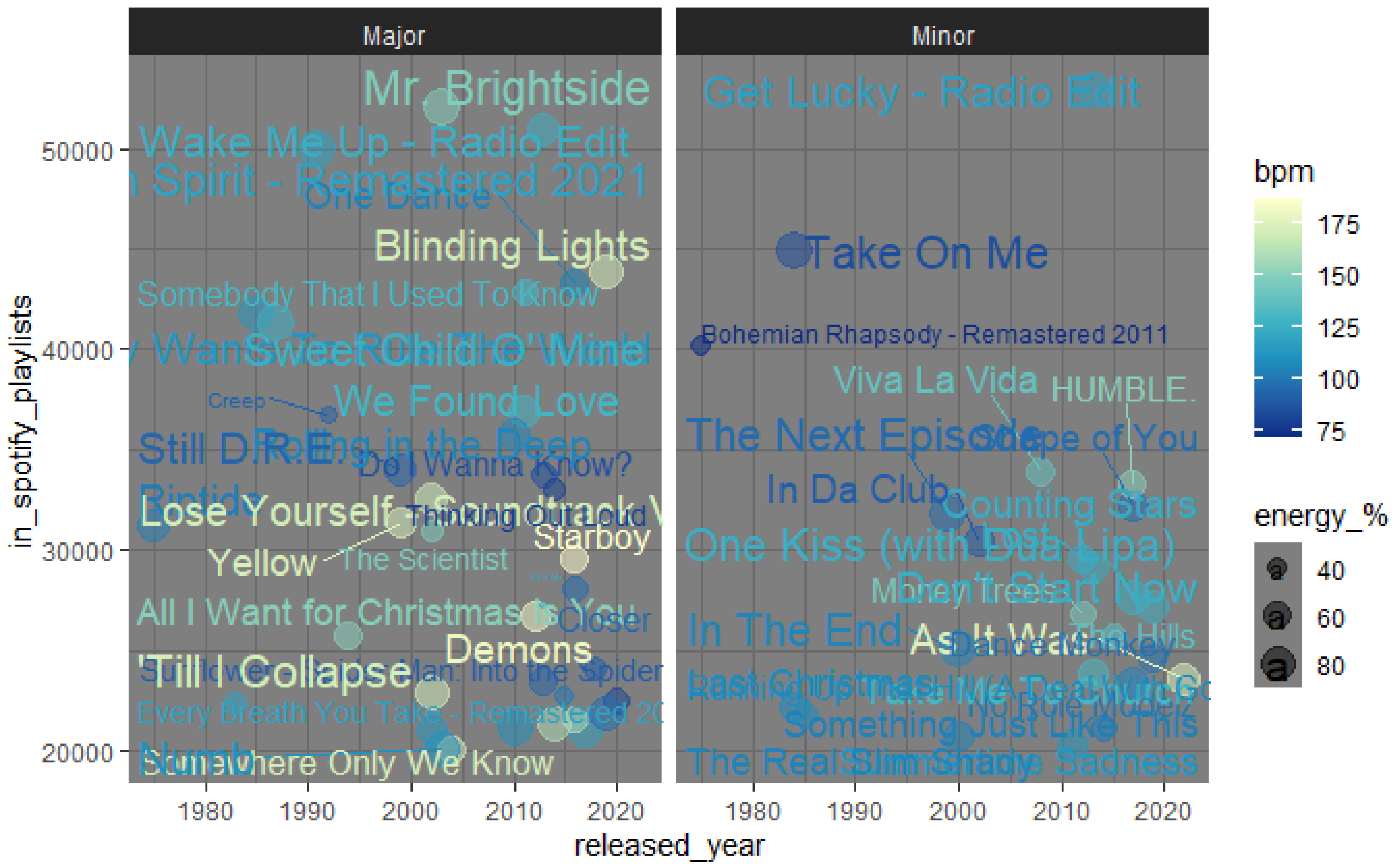


~~~
```{r}

library(gtsummary)
library(gt)

df_wins |>
  tbl_summary() |>
  as_gt() |>
  gtsave("example_table.png")

```
~~~

```
file:///C:/Users/mb3653/AppData/Local/Temp/RtmpgtDQzV/file52b473be4cbd.html screenshot completed
```

Practice: try to recreate this graph

- Download and load Spotify data
- I filtered the data to only include songs that are in 20,000 or more playlists
- Packages: tidyverse, ggrepel
- Think in terms of each piece of a ggplot that I walked through
- facet_grid(. ~ *varname*) is the syntax for one variable (comparing to the previous example in the slides)
- Try scale_color_distiller() and a preexisting palette
- Column names with special characters, like % signs should be typed within ` ` quotes in R



Additional resources

- https://ggplot2-book.org/ (in-depth description)
- https://ggplot2.tidyverse.org/ (package website)
- https://exts.ggplot2.tidyverse.org/gallery/ (additional graph types)
- https://r-graph-gallery.com/ (more graphs)
- https://rstudio.github.io/cheatsheets/data-visualization.pdf (‘cheatsheet’)
- https://rstudio.github.io/cheatsheets/html/data-visualization.html (another summary)
- See also links on other slides for more information about specific topics

Acknowledgements

- Content adapted from:
  - Wickham et al. (https://ggplot2.tidyverse.org/articles/ggplot2.html)
  - ZfS-Kurs “Einführung in die Statistik R” by Veronica Kunz
- Data sources:
  - NHL (https://www.quanthockey.com/)
  - ATP Tour (https://www.kaggle.com/datasets/gmadevs/atp-matches-dataset)
  - Spotify (https://www.kaggle.com/datasets/abdulszz/spotify-most-streamed-songs)
- Additional thanks to Niklas Hähn, Curt Signorino, Josh Kalla, ISPS/CSAP