Wrapping up: hosting and best practices

Lyndsie Wszola

Before we begin

More shiny content:

https://lsw5077.github.io/shiny_workshop/

Libraries:

```
library(shiny)
library(tidyverse)
```

Agenda

- Best practices: general R
- Best practices: shiny
- Hosting: set your app free!

Best practices: general R

Any fool can write code that a computer can understand. Good programmers write code that humans can understand.

– Martin Fowler, 2008.

Best practices: general R

☺

Best practice

Best practices: general R

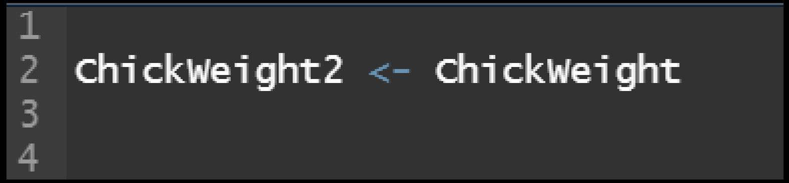
☺

Best practice

☹

Less-than-best practice

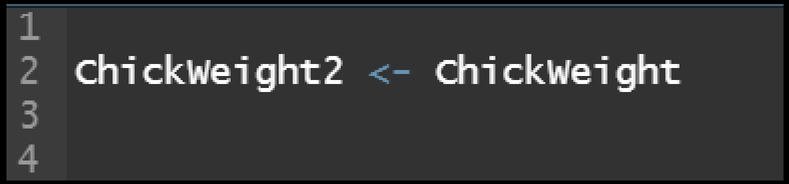
Best practices: Names and comments



```
ChickWeight2 <- ChickWeight
```

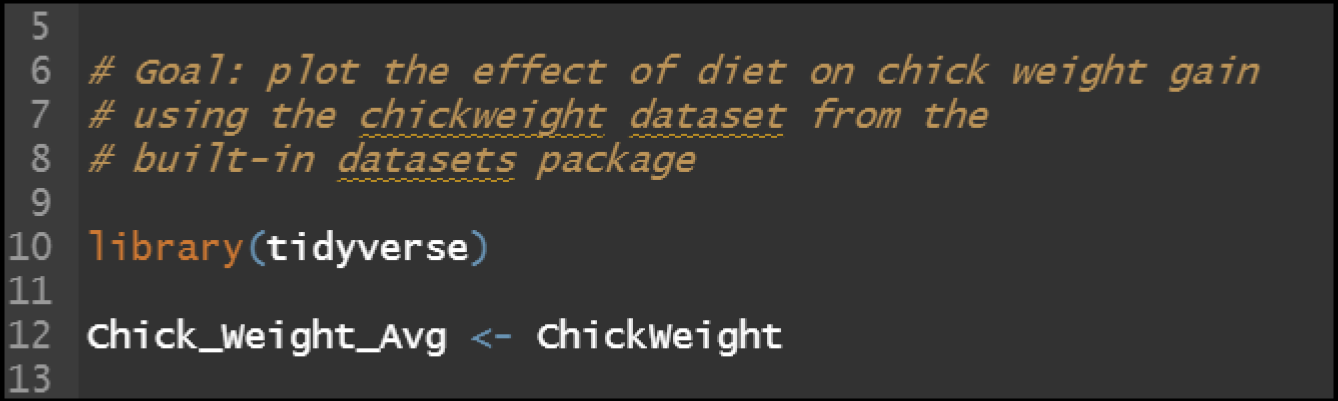
☹

Best practices: Names and comments



```r

ChickWeight2 <- ChickWeight


```

☹

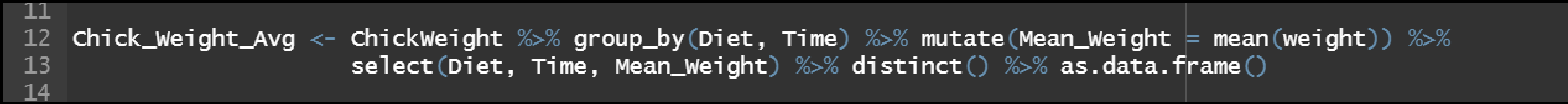


```r

# Goal: plot the effect of diet on chick weight gain
# using the chickweight dataset from the
# built-in datasets package

library(tidyverse)

Chick_Weight_Avg <- ChickWeight

```

☺

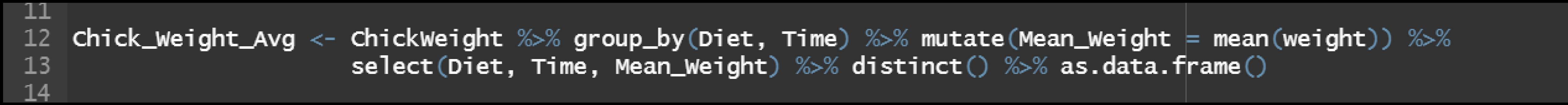
Best practices: Formatting

☹

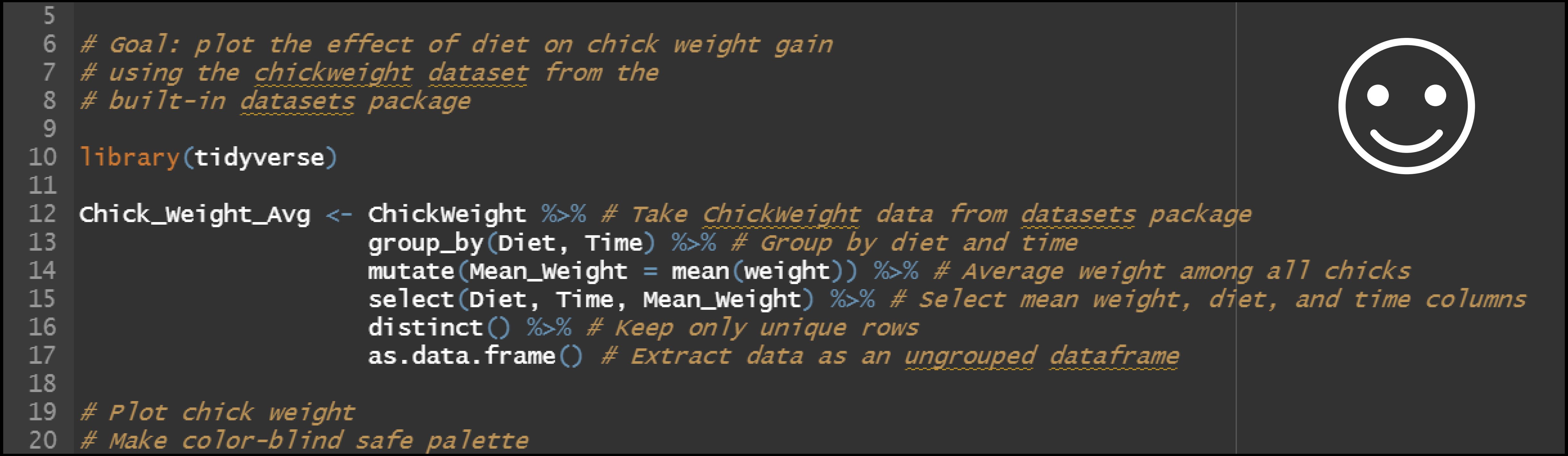


```
Chick_Weight_Avg <- ChickWeight %>% group_by(Diet, Time) %>% mutate(Mean_Weight = mean(weight)) %>%
                    select(Diet, Time, Mean_Weight) %>% distinct() %>% as.data.frame()
```

Best practices: Formatting

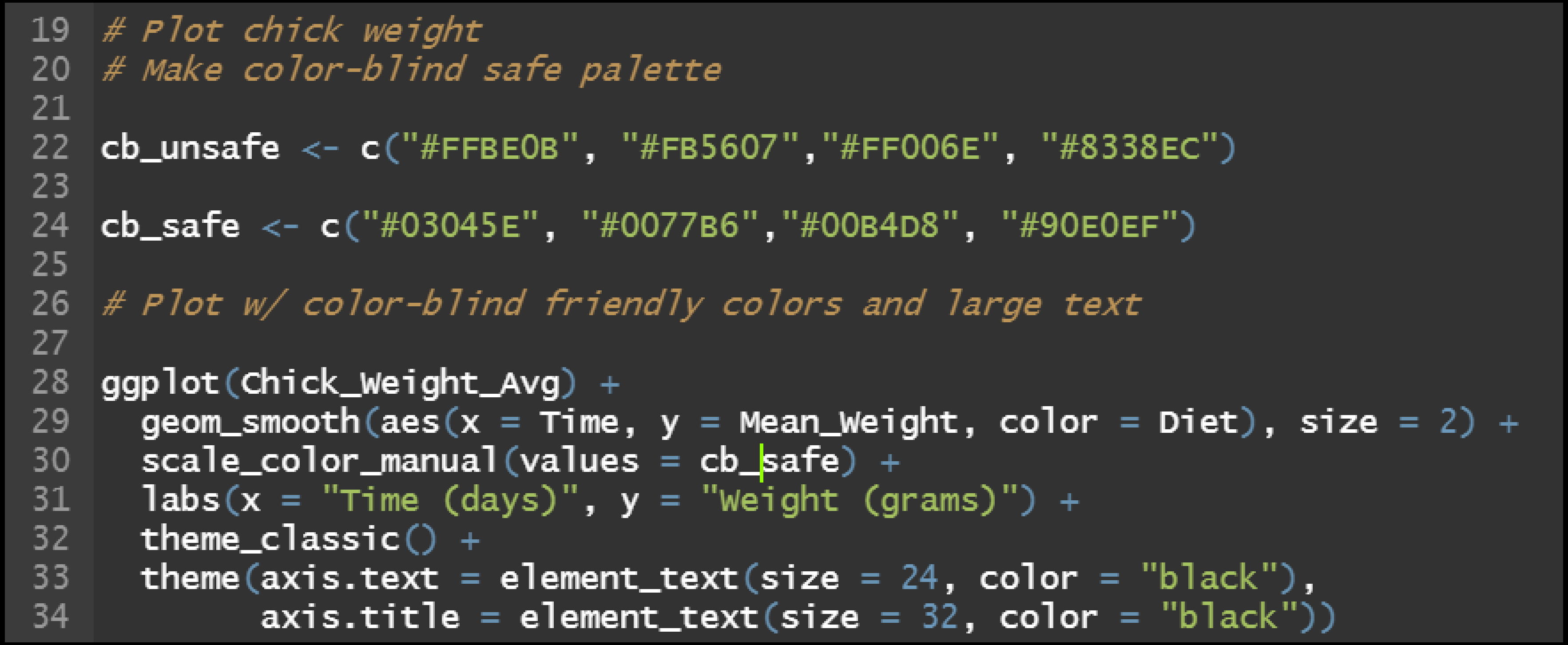


```
Chick_Weight_Avg <- ChickWeight %>% group_by(Diet, Time) %>% mutate(Mean_Weight = mean(weight)) %>%
                    select(Diet, Time, Mean_Weight) %>% distinct() %>% as.data.frame()
```



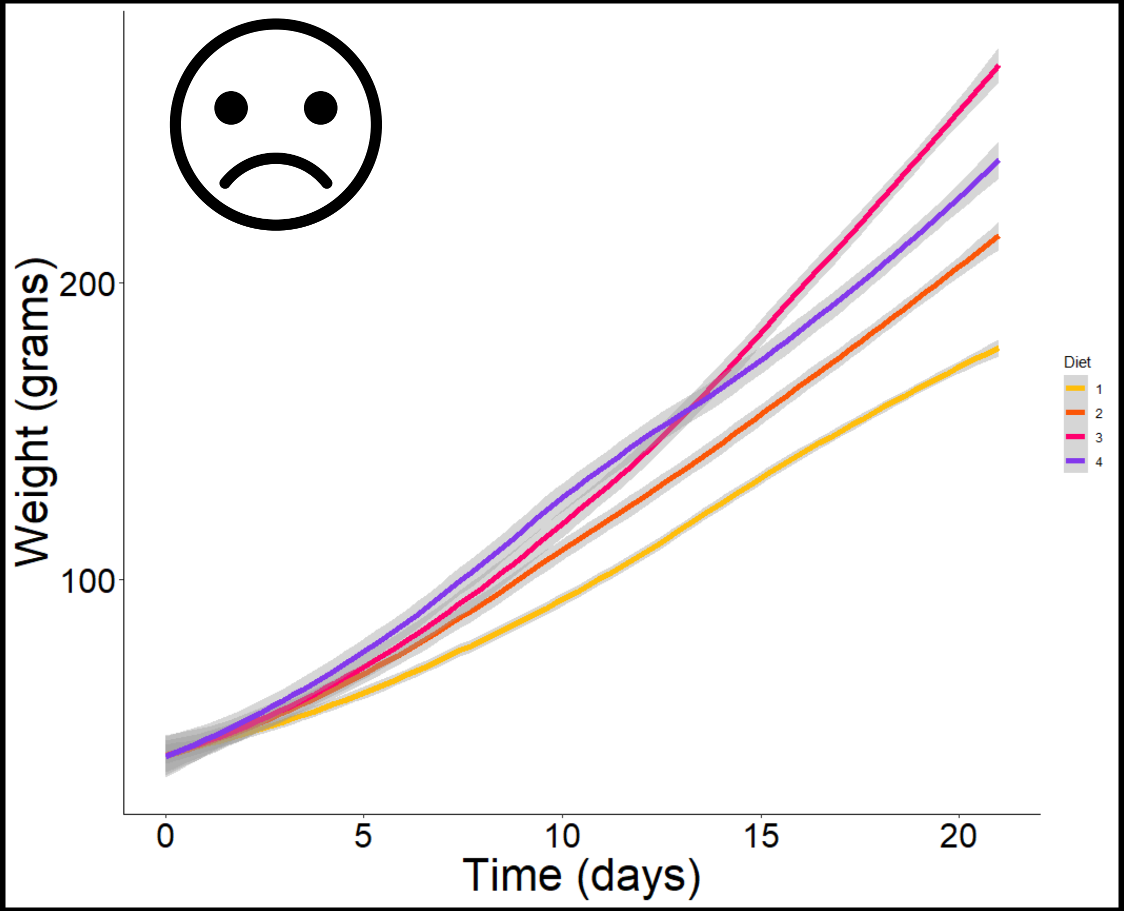
```

# Goal: plot the effect of diet on chick weight gain
# using the chickweight dataset from the
# built-in datasets package

library(tidyverse)

Chick_Weight_Avg <- ChickWeight %>% # Take ChickWeight data from datasets package
                    group_by(Diet, Time) %>% # Group by diet and time
                    mutate(Mean_Weight = mean(weight)) %>% # Average weight among all chicks
                    select(Diet, Time, Mean_Weight) %>% # Select mean weight, diet, and time columns
                    distinct() %>% # Keep only unique rows
                    as.data.frame() # Extract data as an ungrouped dataframe

# Plot chick weight
# Make color-blind safe palette
```

Best practices: Accessibility

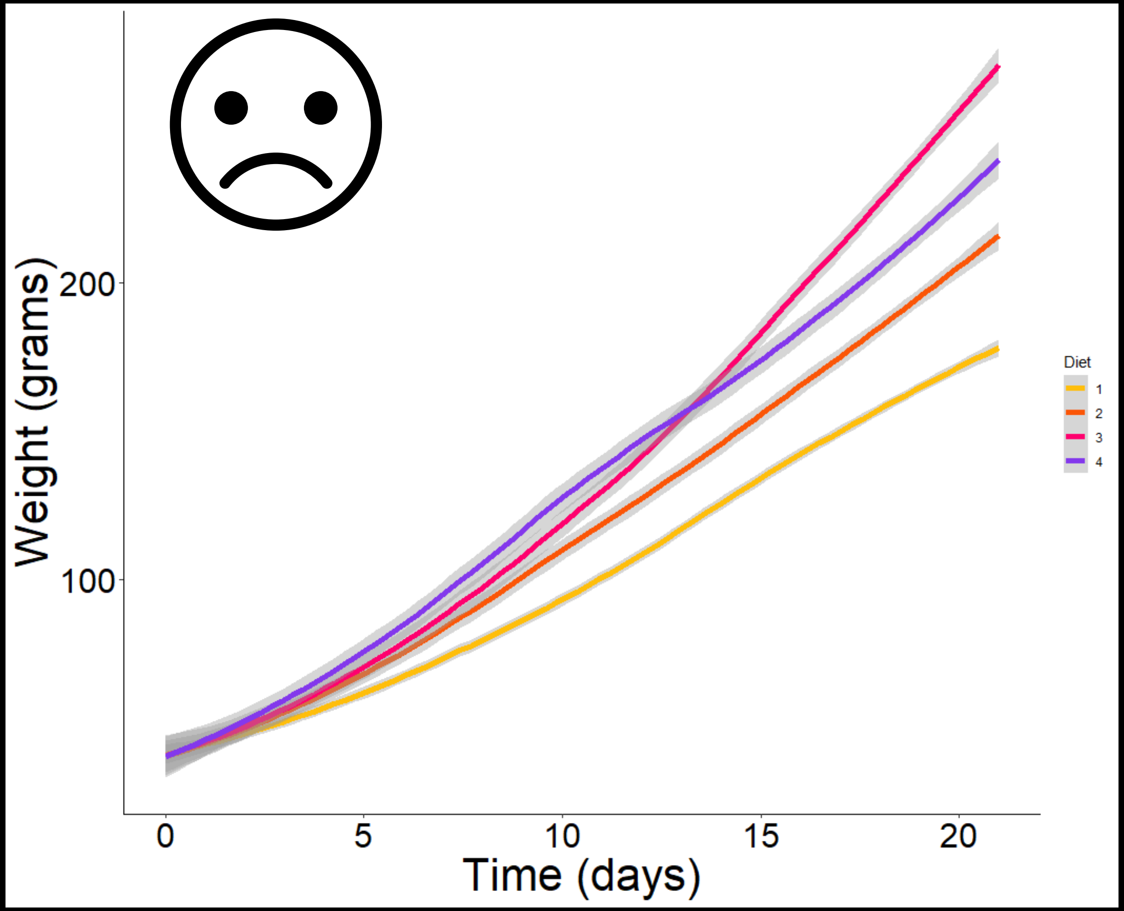


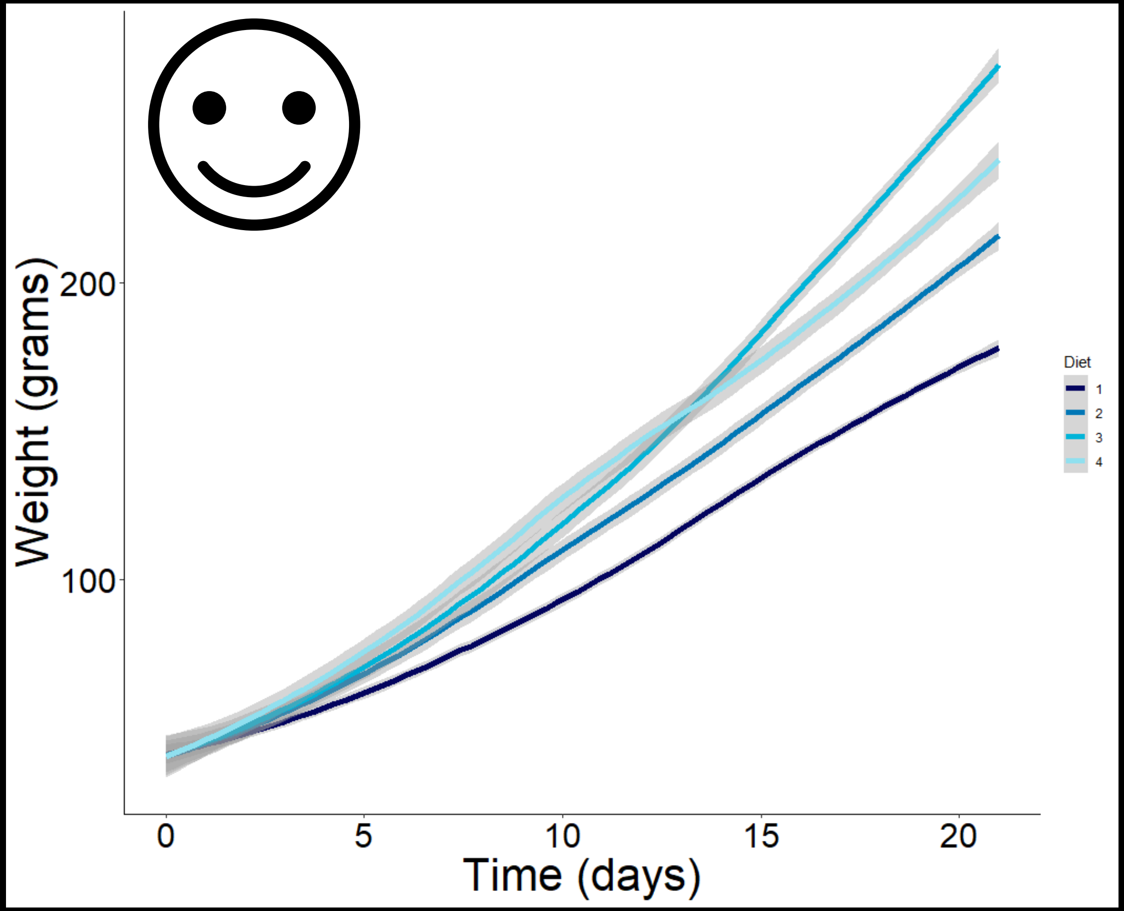
```r
# Plot chick weight
# Make color-blind safe palette

cb_unsafe <- c("#FFBE0B", "#FB5607","#FF006E", "#8338EC")

cb_safe <- c("#03045E", "#0077B6","#00B4D8", "#90E0EF")

# Plot w/ color-blind friendly colors and large text

ggplot(Chick_Weight_Avg) +
  geom_smooth(aes(x = Time, y = Mean_Weight, color = Diet), size = 2) +
  scale_color_manual(values = cb_safe) +
  labs(x = "Time (days)", y = "Weight (grams)") +
  theme_classic() +
  theme(axis.text = element_text(size = 24, color = "black"),
        axis.title = element_text(size = 32, color = "black"))
```

Best practices: Accessibility

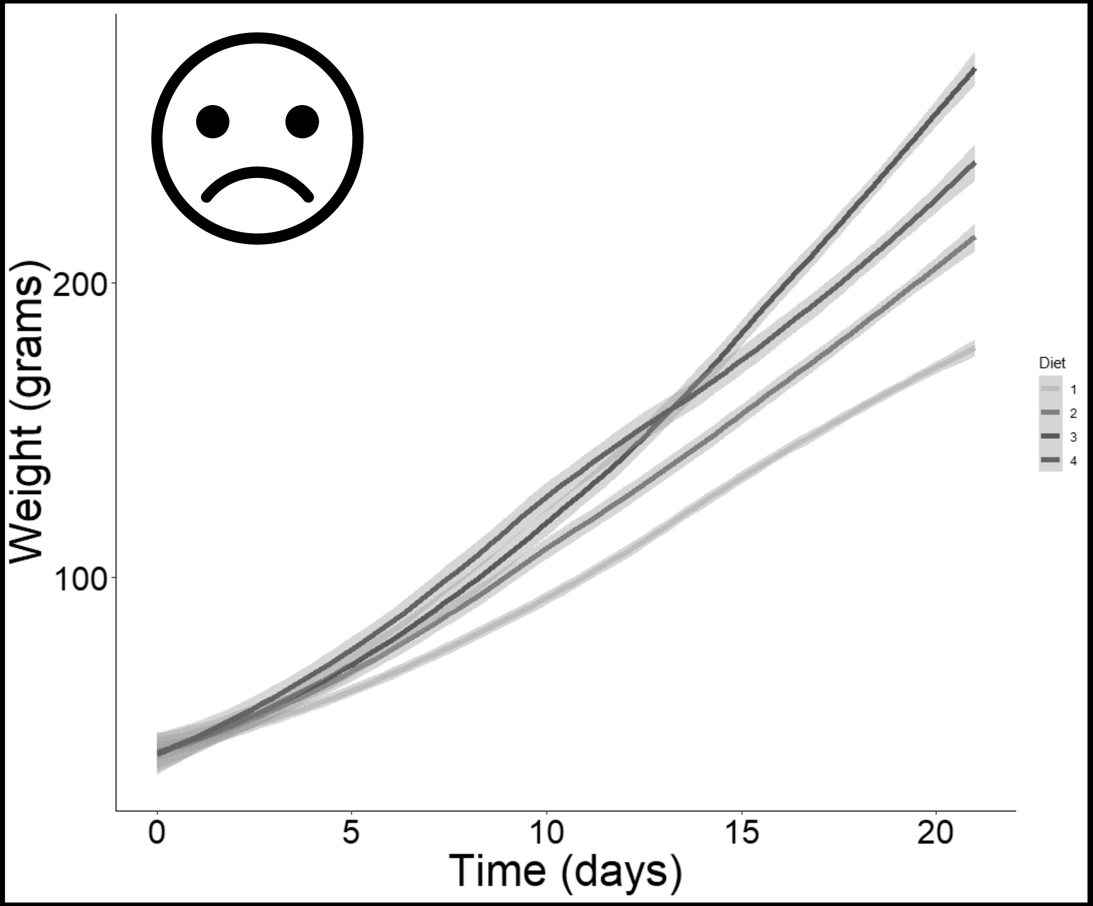


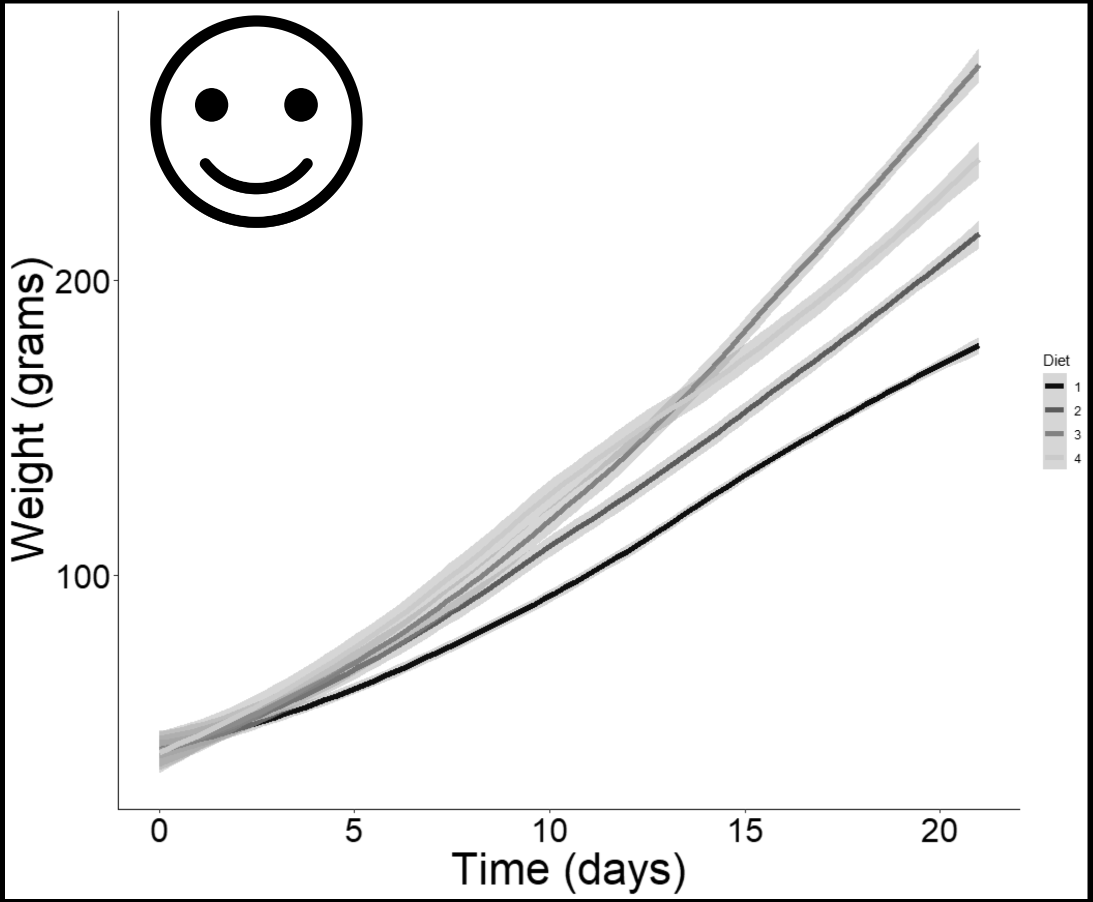
Best practices: Accessibility



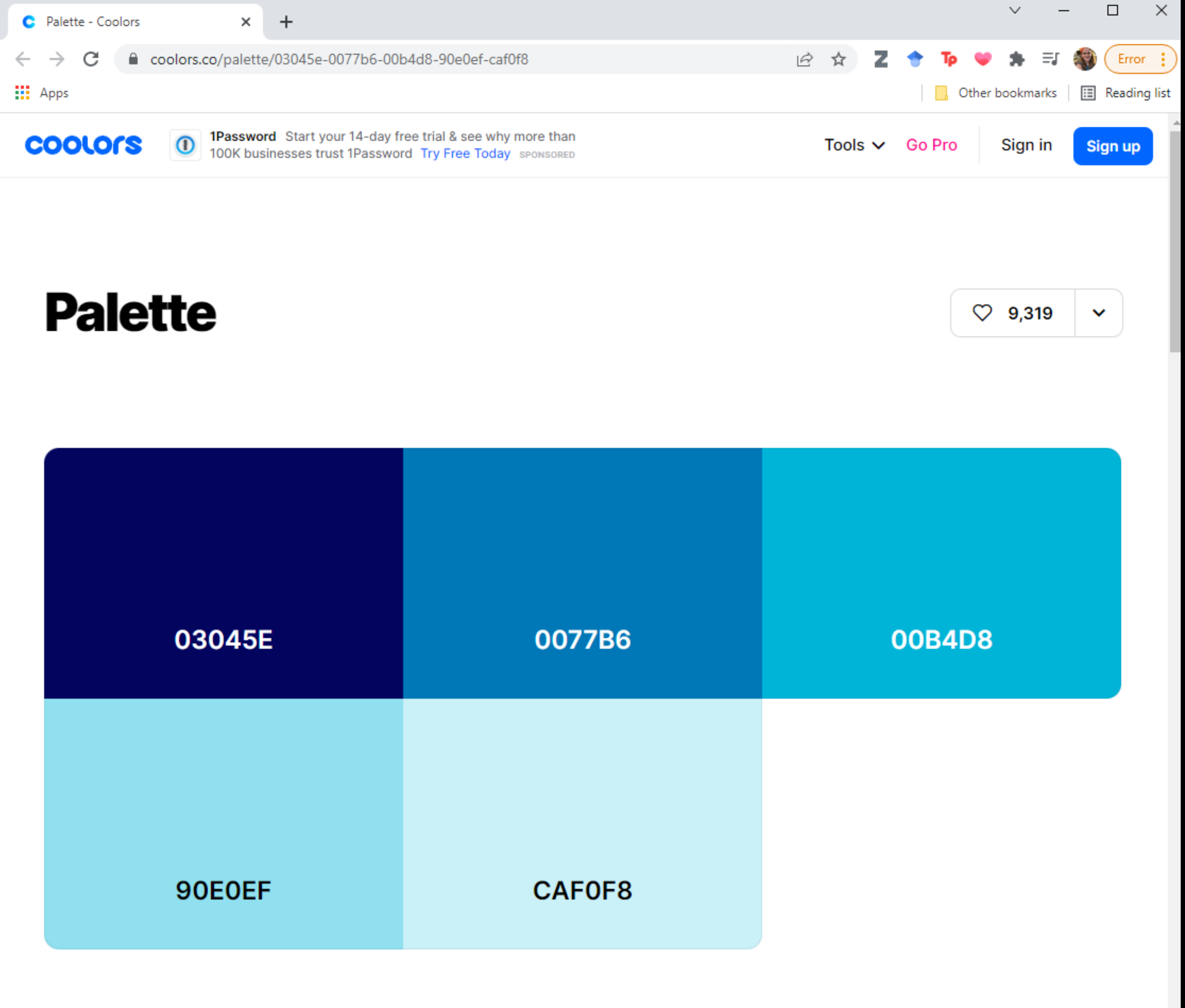


Best practices: Accessibility





Best practices: Accessibility



Palette

9,319

03045E

0077B6

00B4D8

90E0EF

CAF0F8

Best practices: R

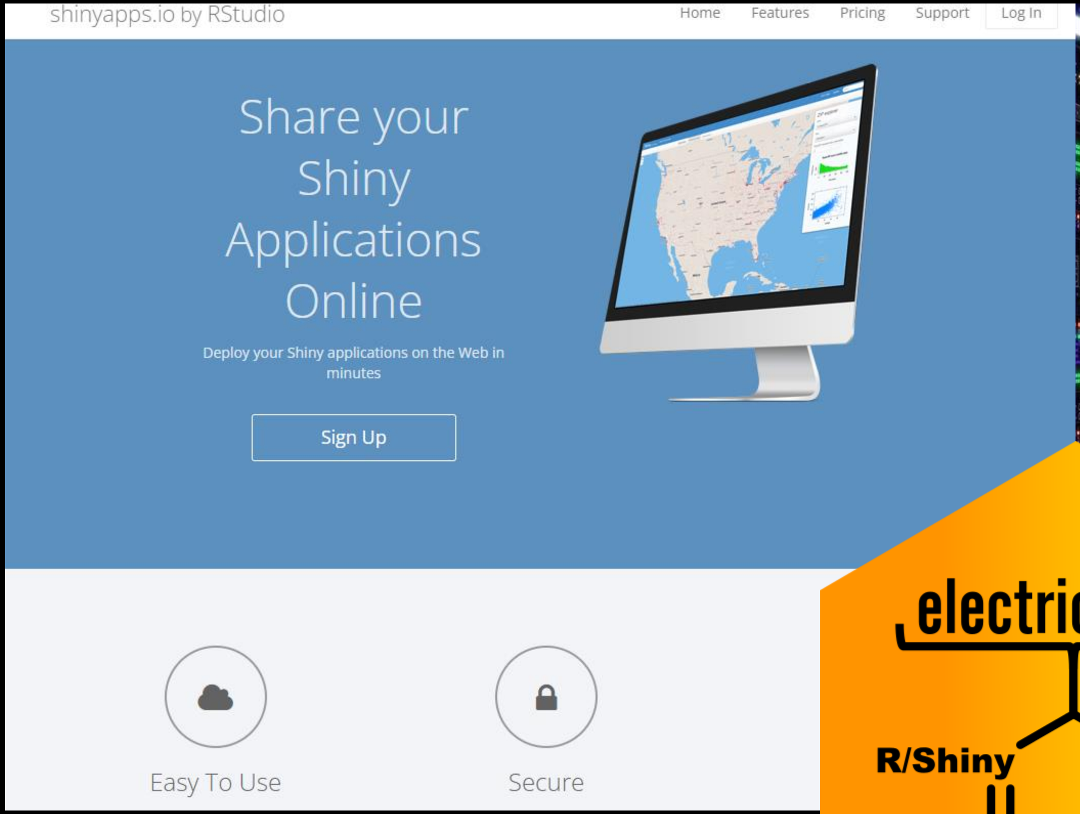
- Human-readable names
- Extensive comments
- Breathable formatting
- Accessible figures

Best practices: Shiny

- Image accessibility
- Citations

Do. Things. Once.

Hosting: options



Share your Shiny Applications Online

Deploy your Shiny applications on the Web in minutes

Easy To Use

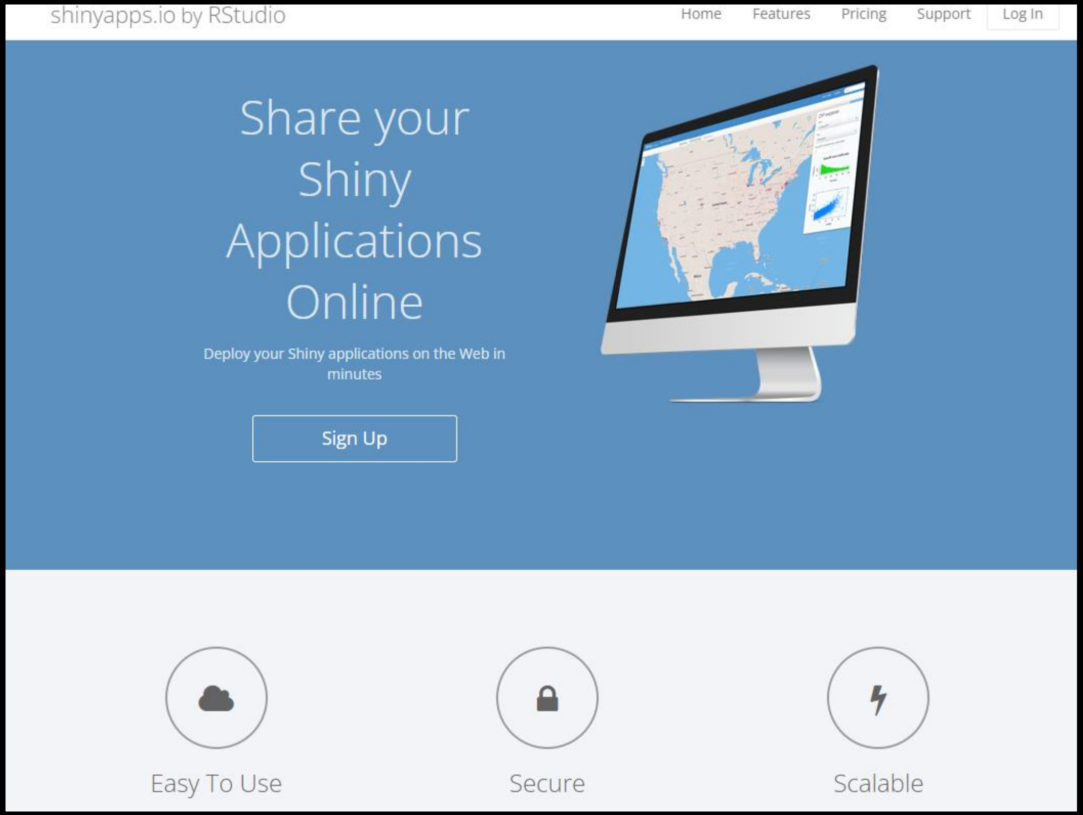
Secure

electricShine

R/Shiny

Electron

Hosting: shinyapps.io



Share your Shiny Applications Online

Deploy your Shiny applications on the Web in minutes

Pros:

- Easy!
- Error logging
- Usage stats

Cons:

- Limited apps/server time
- Potentially expensive

Hosting: your own server

Pros:

- Cheap
- Full control

Cons:

- Needs IT expertise
- Requires Linux servers

Hosting: Internal deployment

Pros:

- Secure
- Can share apps offline

Cons:

- Only works on internal systems

Hosting: shinyapps.io