{poorman}

A dependency free recreation of {dplyr}

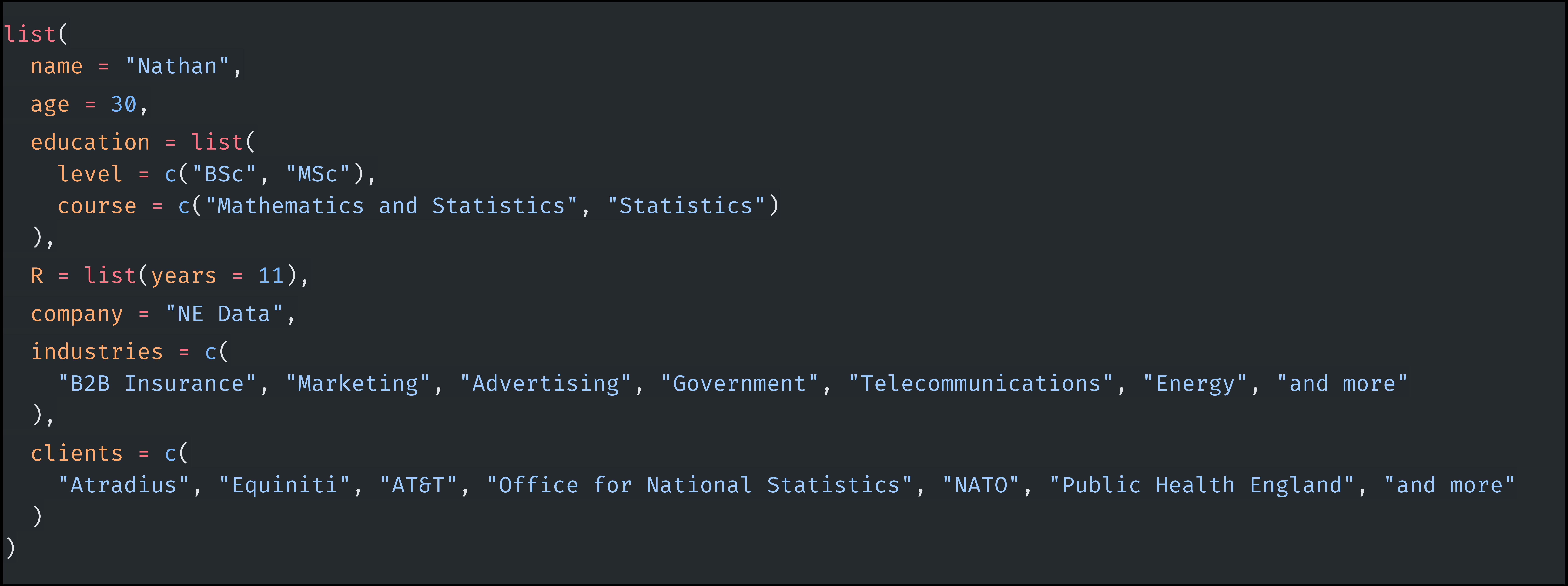
Nathan Eastwood 30/07/2020

Today’s Talk

- Who Am I?
- What Is {poorman}?
- What Does {poorman} Include?
- Let’s Manipulate Some Data!
- Why Develop {poorman} and other FAQs
- What Does The Future Hold For {poorman}?

Who Am I?

A PROGRAMMER n. [proh-gram-er]; see also WIZARD, MAGICIAN



```r
list(
  name = "Nathan",
  age = 30,
  education = list(
    level = c("BSc", "MSc"),
    course = c("Mathematics and Statistics", "Statistics")
  ),
  R = list(years = 11),
  company = "NE Data",
  industries = c(
    "B2B Insurance", "Marketing", "Advertising", "Government", "Telecommunications", "Energy", "and more"
  ),
  clients = c(
    "Atradius", "Equiniti", "AT&T", "Office for National Statistics", "NATO", "Public Health England", "and more"
  )
)
```

https://github.com/nathaneastwood/poorman

BBC three HD
L
C
O
WHEN'S THE DENTIST?
2:30
SERIOUSLY, FB!
What is this?!

What Is {poorman}?

An awesome R package!

- {poorman} is a feature rich, grammar of data manipulation R 📦
- It unapologetically 💁‍♂️ tries to recreate the {dplyr} API exactly
- Written entirely using {base}, i.e. 0️⃣ dependencies
- Installation time is ~6 seconds 💯
- Many of the {dplyr} verbs are included
- A version of the pipe, %>%, is included

https://github.com/nathaneastwood/poorman

What Does {poorman} Include?

Lots of core goodies!

```
select(), rename(), pull(), relocate()
mutate(), transmute()
arrange()
filter(), slice()
summarise() / summarize()
group_by(), ungroup()

inner_join(), left_join(), right_join(), full_join()
anti_join(), semi_join()

%>%
```

https://github.com/nathaneastwood/poorman

What Does {poorman} Include?

As well as lots of other great features!

```
# Vector functions:
between(), coalesce(), desc(), if_else(), lag(), lead(), n_distinct(), na_if(),
near(), recode(), recode_factor()

# Window functions:
cume_dist(), dense_rank(), min_rank(), ntile(), percent_rank(), row_number()
```

https://github.com/nathaneastwood/poorman

But Wait! There’s More!

Not so poor after all!

- {poorman} also includes its own version of {tidyselect}

```
starts_with(), ends_with(), contains(), matches(), all_of(), any_of(), everything(),
last_col(), where()
```

- There are also features from 🧹 {tidyr}

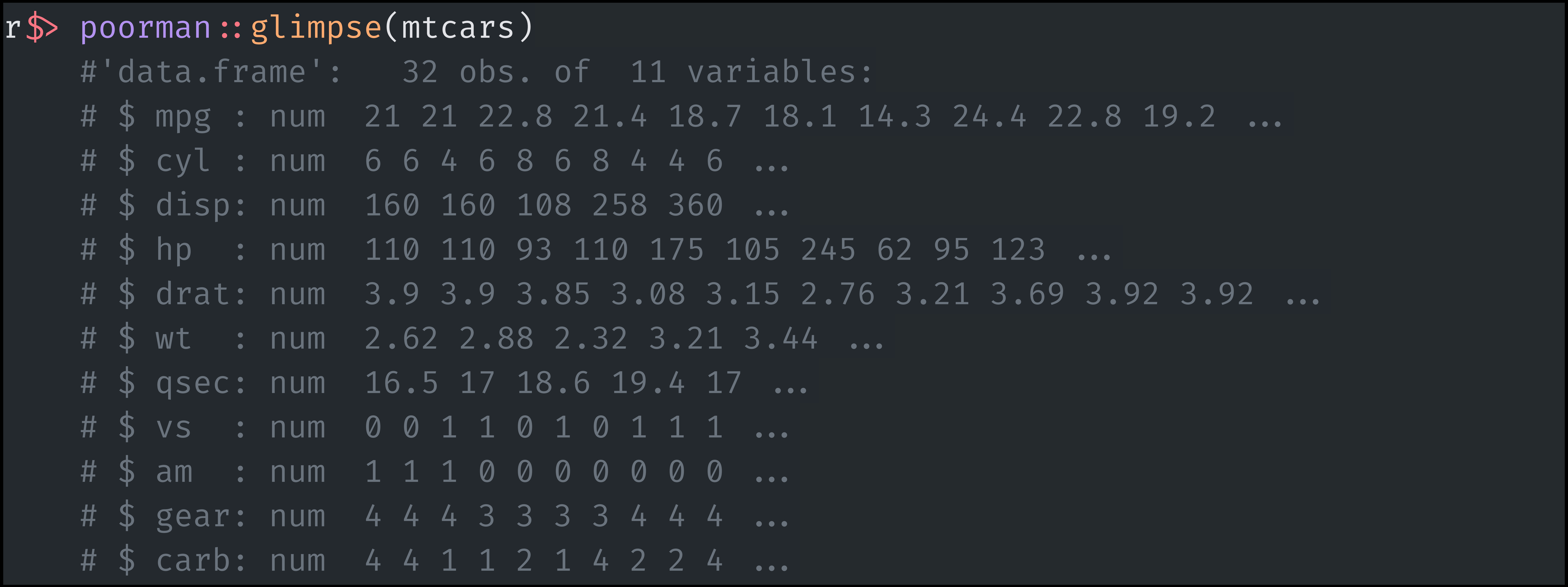
```
replace_na(), rownames_to_column()
```

- And more!

https://github.com/nathaneastwood/poorman

Let’s Manipulate Some Data

We’ll Use mtcars

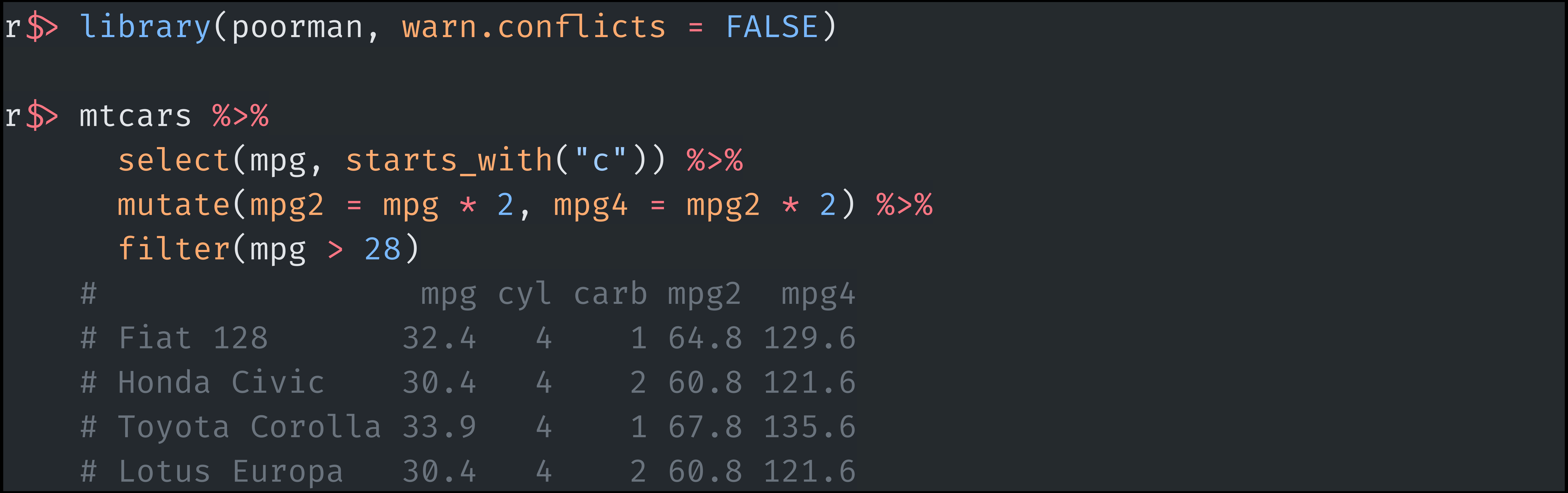


```
r$> poorman::glimpse(mtcars)
    #'data.frame':   32 obs. of  11 variables:
    # $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
    # $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
    # $ disp: num  160 160 108 258 360 ...
    # $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
    # $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
    # $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
    # $ qsec: num  16.5 17 18.6 19.4 17 ...
    # $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
    # $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
    # $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
    # $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

https://github.com/nathaneastwood/poorman

Let’s Manipulate Some Data

data THEN select( ) THEN mutate( ) THEN filter( )



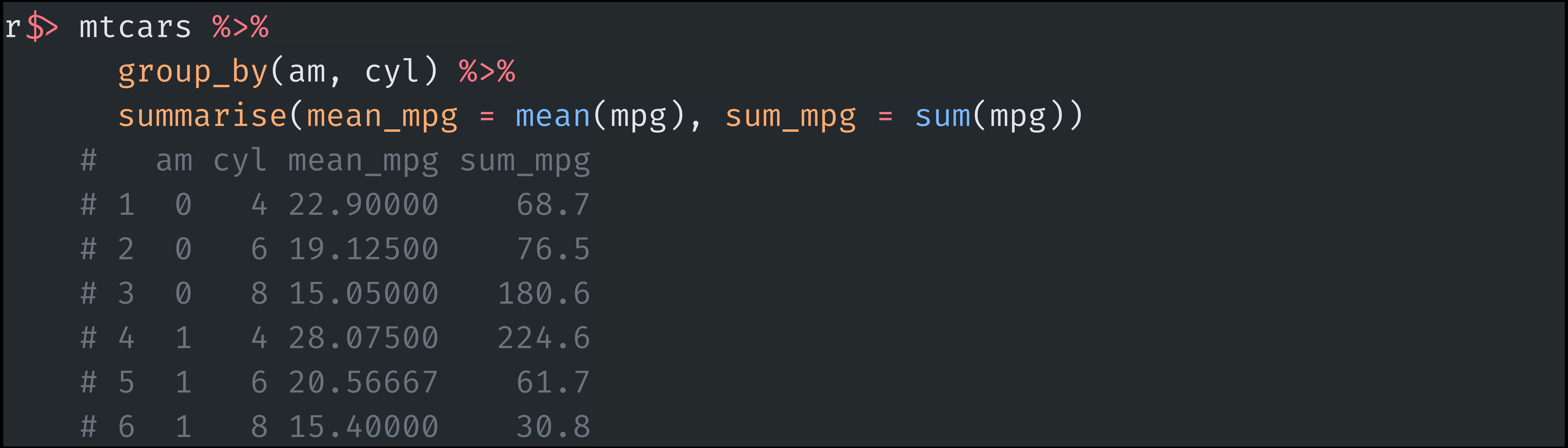
```
r$> library(poorman, warn.conflicts = FALSE)

r$> mtcars %>%
      select(mpg, starts_with("c")) %>%
      mutate(mpg2 = mpg * 2, mpg4 = mpg2 * 2) %>%
      filter(mpg > 28)
    #                 mpg cyl carb mpg2  mpg4
    # Fiat 128       32.4   4    1 64.8 129.6
    # Honda Civic    30.4   4    2 60.8 121.6
    # Toyota Corolla 33.9   4    1 67.8 135.6
    # Lotus Europa   30.4   4    2 60.8 121.6
```

https://github.com/nathaneastwood/poorman

Let’s Manipulate Some Data

data THEN group_by( ) THEN summarise( )

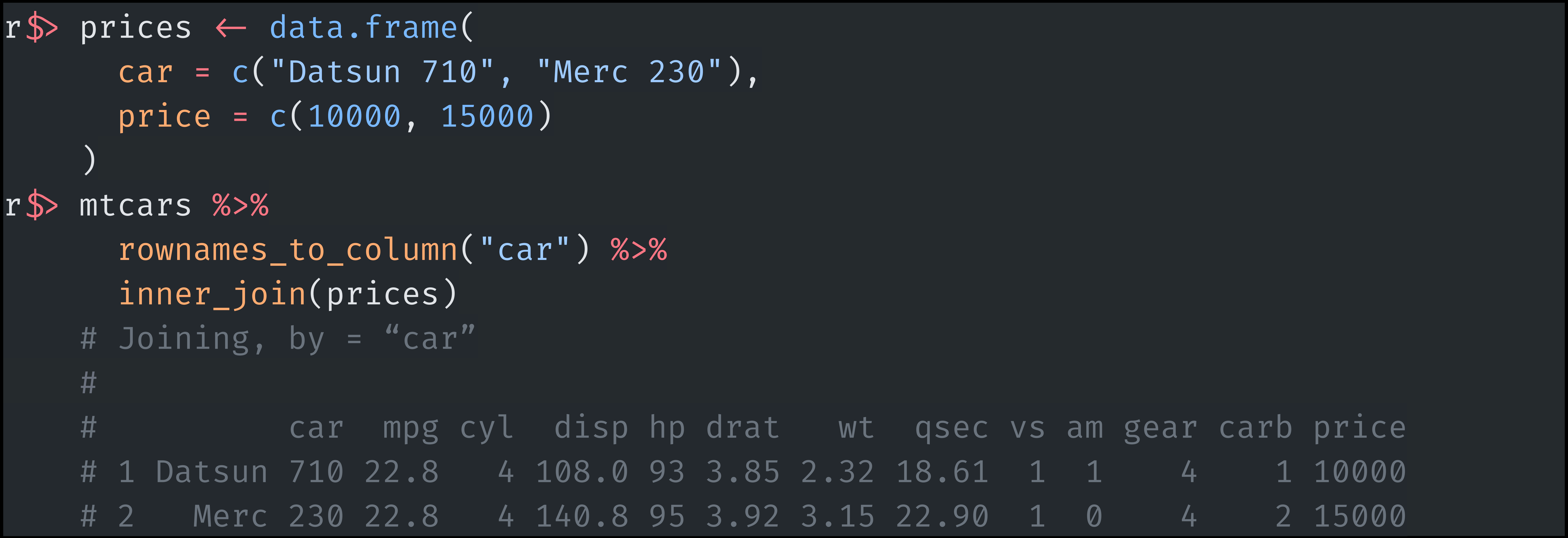


```
r$> mtcars %>%
      group_by(am, cyl) %>%
      summarise(mean_mpg = mean(mpg), sum_mpg = sum(mpg))
    #    am cyl mean_mpg sum_mpg
    # 1   0   4 22.90000    68.7
    # 2   0   6 19.12500    76.5
    # 3   0   8 15.05000   180.6
    # 4   1   4 28.07500   224.6
    # 5   1   6 20.56667    61.7
    # 6   1   8 15.40000    30.8
```

https://github.com/nathaneastwood/poorman

Let’s Manipulate Some Data

data THEN rownames_to_column( ) THEN inner_join( )



```
r$> prices ← data.frame(
      car = c("Datsun 710", "Merc 230"),
      price = c(10000, 15000)
    )
r$> mtcars %>%
      rownames_to_column("car") %>%
      inner_join(prices)
    # Joining, by = “car”
    #
    #          car  mpg cyl  disp hp drat   wt  qsec vs am gear carb price
    # 1 Datsun 710 22.8   4 108.0 93 3.85 2.32 18.61  1  1    4    1 10000
    # 2   Merc 230 22.8   4 140.8 95 3.92 3.15 22.90  1  0    4    2 15000
```

https://github.com/nathaneastwood/poorman

Demo

Why Develop {poorman}?

Quite simply, because it was fun!

- {poorman} grew organically 🌱
- It was an interesting challenge
- As a freelance developer 👨‍💻, it shows off my skills nicely
- {poorman} provides a platform to show and teach 👨‍🏫 {base} skills

Why Develop {poorman}?

But more seriously…

- In the corporate world, dependencies can be bad for business 🤨
- I don’t always need the fancy backends provided by {dbplyr}
- I wanted to use the API in my packages without forcing further dependencies 🎁 on my users
- Installation times ⏱️
- I wanted a source that was easier to understand and which provided nicer tracebacks
- I wanted to challenge a common misconception that {base} R is not as powerful 💪, or as good, or as useful as {dplyr}

https://github.com/nathaneastwood/poorman

“I’d seen my father. He was a poor man, and I watched him do astonishing things.”

Sidney Poitier

How Does {poorman} Compare In Terms Of Speed?

When this thing gets up to 88 mph, you're gonna see some serious s***

- I haven’t ran any benchmarks 😔
- There are plenty of benchmarks comparing {dplyr} and {base} online 😃
- {poorman} will perform similarly to those benchmarks 😏
- If speed is a concern of yours, I’d recommend considering {data.table} 🤯

What Does The Future Hold For {poorman}?

A great many things!

- More features from {dplyr} and the wider {tidyverse}, e.g. tidyr::pivot_wide( ) ➡️ and tidyr::pivot_longer( ) ⬇️
- Possible different backends (C++, {data.table}) which are user choices
- A poor man’s alternative to {rlang}
- … and documentation 📘

https://github.com/nathaneastwood/poorman

Thanks For Listening!

(shameless plugs go here)

- Don’t forget to ⭐ the repo: https://github.com/nathaneastwood/poorman
- Or 🐦 about it and @nathaneastwood_
- If you need an R developer, get in touch: nathan.eastwood@icloud.com
- Or if you’d like to chat more about R, {poorman}, dependencies, etc. then feel free to message me