PS1_Collatz

AUTHOR
1093122

Exercise 1 collatz_sequence function

a. Warm up

When n = 3, the next number in the sequence is 10.

```
n <- 3

if (n %% 2 == 0) {
  result <- n/2 # divide by 2 if it is even
} else{
  result <- n*3 + 1 # if it is odd
}
print(result)
```

```
[1] 10
```

b. Store the first two integers of the sequence for 3 into seq_3

```
seq_3 <- c(3)
seq_3 <- c(seq_3, result)
print(seq_3)
```

```
[1]  3 10
```

c. Get seq_5 by using while()

```
n <- 5
seq_5 <- c(5)
while (n != 1) {
  if (n %% 2 == 0) {
  n <- n/2
} else{
  n <- n*3 + 1
}
  seq_5 <- c(seq_5, n)
}

print(seq_5)
```

```
[1]  5 16  8  4  2  1
```

d. function collatz_sequence

```
# modify c and turn it into a function
collatz_sequence <- function(n){
  seq <- c(n)
while (n != 1) {
  if (n %% 2 == 0) {
  n <- n/2
} else{
  n <- n*3 + 1
}
  seq <- c(seq, n)
}

return(seq)
}

print(collatz_sequence(5))
```

```
[1]  5 16  8  4  2  1
```

Exercise 2 collatz_holds function to check if Collatz conjecture holds

a. use for loop to iteratre over all integers between 3 and 10

```
a <- 3
b <- 10
# call collatz_sequence function for every number in a:b and print it
for (i in a:b) {
 print(collatz_sequence(i))
}
```

```
[1]  3 10  5 16  8  4  2  1
[1] 4 2 1
[1]  5 16  8  4  2  1
[1]  6  3 10  5 16  8  4  2  1
 [1]  7 22 11 34 17 52 26 13 40 20 10  5 16  8  4  2  1
[1] 8 4 2 1
 [1]  9 28 14  7 22 11 34 17 52 26 13 40 20 10  5 16  8  4  2  1
[1] 10  5 16  8  4  2  1
```

b. write a function that can do the same for arbitrary a and b

```
collatz_holds <- function(a, b){
  for (i in a:b) {
 print(collatz_sequence(i))
}
}
collatz_holds(3, 10)
```

```
[1]  3 10  5 16  8  4  2  1
[1] 4 2 1
[1]  5 16  8  4  2  1
[1]  6  3 10  5 16  8  4  2  1
 [1]  7 22 11 34 17 52 26 13 40 20 10  5 16  8  4  2  1
[1] 8 4 2 1
 [1]  9 28 14  7 22 11 34 17 52 26 13 40 20 10  5 16  8  4  2  1
[1] 10  5 16  8  4  2  1
```

c. Modify the function so it only prints out the last element of each sequence by using tail()

```
collatz_holds <- function(a, b){
  for (i in a:b) {
 print(tail(collatz_sequence(i), 1))
}
}
collatz_holds(3, 10)
```

```
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
```

d. Modify the function: use if-else statement to check whether this element equals one and store the appropriate TRUE/FALSE value in a vector

```
collatz_holds <- function(a, b){
```

```
  v <-  c()
  for (i in a:b) {
if (tail(collatz_sequence(i), 1) == 1){
  v <- c(v, TRUE)
} else{
  v <- c(v, FALSE)
}
  }
  return(v)
}

print(collatz_holds(3, 10))
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Exercise 3: write a function called longest_collatz that returns two numbers for any starting integer over [a,b]: the longest Collatz sequence, and the length of that sequence

```
collatz_holds <- function(a, b){
  v <-  c()
  for (i in a:b) {
    l <- length(collatz_sequence(i))
    v[i-a+1] <- l
    names(v)[i-a+1] <- i
  }
  max_integer <- names(v)[which.max(v)]
  max_length <- max(v)
  return(c(max_integer, max_length))
}
collatz_holds(6, 100)
```

```
[1] "97"  "119"
```

Exercise 4: write a function get_collatz_steps() that takes one argument n_max and returns a tibble with two columns: n and steps.

```
library(tibble)
get_collatz_steps <- function(n_max){
  my_tibble <- tibble(
    n = numeric(n_max),
    steps = numeric(n_max)
  )
  for (i in 1:n_max){
   my_tibble$n[i] <- i
   my_tibble$steps[i] <- length(collatz_sequence(i))
  }
   return(my_tibble)
}
print(get_collatz_steps(500))
```

```
# A tibble: 500 × 2
      n steps
  <dbl> <dbl>
1     1     1
2     2     2
3     3     8
4     4     3
5     5     6
6     6     9
7     7    17
8     8     4
9     9    20
```

```
10    10     7
# ℹ 490 more rows
```

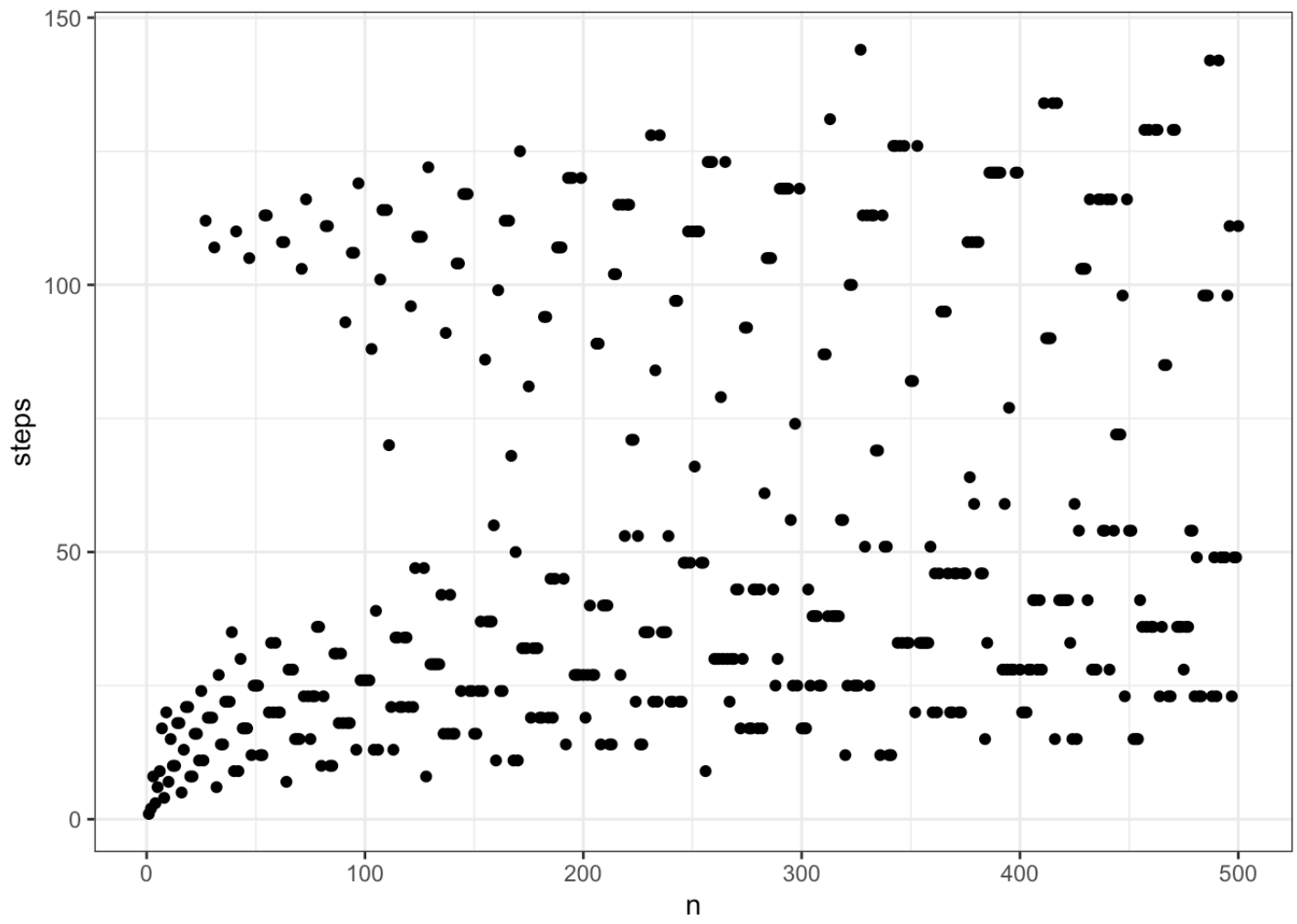
use ggplot2 to create a scatterplot with n on the x-axis and steps on the y-axis

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ purrr     1.0.4
✔ forcats   1.0.0     ✔ readr     2.1.5
✔ ggplot2   3.5.2     ✔ stringr   1.5.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```
data <- get_collatz_steps(500)

data |>
  ggplot(mapping = aes(x = n, y = steps)) +
  geom_point() +
  theme_bw()
```



PS1_Lakisha

AUTHOR
1093122

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.2     ✔ tibble    3.2.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```
bm <- read_csv("https://ditraglia.com/data/lakisha_aer.csv")
```

```
Rows: 4870 Columns: 65
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (10): id, ad, firstname, sex, race, city, kind, expminreq, schoolreq, ow...
dbl (55): education, ofjobs, yearsexp, honors, volunteer, military, empholes...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Exercise 1

a.

There is significant racial inequality in the US labour market and BM try to find whether it is a result of racial bias or other characteristics that are unobservable to the researchers but observable to the employers.

b.

They create a bank of fictitious resumes to help-wanted ads in Boston and Chicago newspapers and measure callback for interview for each sent resume. The resumes are randomly assigned African-American or White-sounding names. They also produce high-quality and low-quality resumes to study how responsive callbacks are to credentials for African-American names compared with White names.The resumes are also sent to a large spetrum of job quality and industries.

c.

They find large racial differences in callback rates. White names receive 50 percent more callbacks for interviews. Callbacks are also more responsive to resume quality for White names than for African-American ones. Applicants with African-American names can find it difficult to overcome this difficulty by improving their observable skills and credentials. The racial gap is uniform across occupation, industry, and employer size. Hence, training program might not be enough to alleviate this racial gap and other forces might be needed.

Exercise 2

a.

It has 4870 rows and 65 columns.

```
bm
```

```
# A tibble: 4,870 × 65
   id    ad    education ofjobs yearsexp honors volunteer military empholes
   <chr> <chr>     <dbl>  <dbl>    <dbl>  <dbl>     <dbl>    <dbl>    <dbl>
 1 b     1             4      2        6      0         0        0        1
```

```
 2 b     1             3      3        6      0         1         1         0
 3 b     1             4      1        6      0         0         0         0
 4 b     1             3      4        6      0         1         0         1
 5 b     1             3      3       22      0         0         0         0
 6 b     1             4      2        6      1         0         0         0
 7 b     1             4      2        5      0         1         0         0
 8 b     1             3      4       21      0         1         0         1
 9 b     1             4      3        3      0         0         0         0
10 b     1             4      2        6      0         1         0         0
# ℹ 4,860 more rows
# ℹ 56 more variables: occupspecific <dbl>, occupbroad <dbl>,
#   workinschool <dbl>, email <dbl>, computerskills <dbl>, specialskills <dbl>,
#   firstname <chr>, sex <chr>, race <chr>, h <dbl>, l <dbl>, call <dbl>,
#   city <chr>, kind <chr>, adid <dbl>, fracblack <dbl>, fracwhite <dbl>,
#   lmedhhinc <dbl>, fracdropout <dbl>, fraccolp <dbl>, linc <dbl>, col <dbl>,
#   expminreq <chr>, schoolreq <chr>, eoe <dbl>, parent_sales <dbl>, …
```

b.

“sex” column contains the sex of the individual indicated by the resume. “f” stands for female and “m” stands for male. “race” column contains the race of the individual implied by the resume. “w” stands for white and “b” stands for black. “firstname” column contains the fictitious name included in the resume.

```
bm[c("sex", "race", "firstname")]
```

```
# A tibble: 4,870 × 3
   sex   race  firstname
   <chr> <chr> <chr>
 1 f     w     Allison
 2 f     w     Kristen
 3 f     b     Lakisha
 4 f     b     Latonya
 5 f     w     Carrie
 6 m     w     Jay
 7 f     w     Jill
 8 f     b     Kenya
 9 f     b     Latonya
10 m     b     Tyrone
# ℹ 4,860 more rows
```

c.

```
# use mutate to create the column. use ifelse to generate TRUE/FALSE values depending on the conditi
bm <- bm |>
  mutate(female = ifelse(sex == "f", TRUE, FALSE),
         black = ifelse(race == "b", TRUE, FALSE))
bm
```

```
# A tibble: 4,870 × 67
   id    ad    education ofjobs yearsexp honors volunteer military empholes
   <chr> <chr>     <dbl>  <dbl>    <dbl>  <dbl>     <dbl>    <dbl>    <dbl>
 1 b     1             4      2        6      0         0        0        1
 2 b     1             3      3        6      0         1        1        0
 3 b     1             4      1        6      0         0        0        0
 4 b     1             3      4        6      0         1        0        1
 5 b     1             3      3       22      0         0        0        0
 6 b     1             4      2        6      1         0        0        0
 7 b     1             4      2        5      0         1        0        0
 8 b     1             3      4       21      0         1        0        1
 9 b     1             4      3        3      0         0        0        0
10 b     1             4      2        6      0         1        0        0
# ℹ 4,860 more rows
# ℹ 58 more variables: occupspecific <dbl>, occupbroad <dbl>,
#   workinschool <dbl>, email <dbl>, computerskills <dbl>, specialskills <dbl>,
#   firstname <chr>, sex <chr>, race <chr>, h <dbl>, l <dbl>, call <dbl>,
#   city <chr>, kind <chr>, adid <dbl>, fracblack <dbl>, fracwhite <dbl>,
#   lmedhhinc <dbl>, fracdropout <dbl>, fraccolp <dbl>, linc <dbl>, col <dbl>,
#   expminreq <chr>, schoolreq <chr>, eoe <dbl>, parent_sales <dbl>, …
```

Exercise 3

a.

The experimenters created their bank of resumes by starting with real resumes posted on job search websites from job seekers in Boston and Chicago. They altered the resumes by replacing school and employer names with equivalent institutions from the opposite city (e.g., Boston resumes were adapted for Chicago and vice versa). They further diversified the resumes by varying fonts, layouts, and cover letter styles.

b.

The resumes were classified into two groups: high-quality and low-quality. This classification was based on subjective criteria such as labor market experience, career profile, employment gaps, and listed skills. To reinforce the quality distinction, additional features were added to high-quality resumes, like summer jobs, volunteer experience, extra computer skills, certifications, foreign language skills, honors, or military experience.

c.

To generate identities for the fictitious applicants, the experimenters randomly assigned racially distinctive names, using birth certificate data to ensure name authenticity. They also assigned phone numbers (with race/gender-matched voicemail recordings), randomized postal addresses from real streets in Boston and Chicago, and created neutral email addresses for some resumes. This process ensured that the only systematic differences between applicants were their perceived race and resume quality.

Exercise 4

a.

The proportion of female for each race is around 77%. Hence, sex is balanced across race.

```
library(dplyr)
# group by race and calculate the proportion of female in each group
bm |>
  group_by(race) |>
  summarize(female_percent = mean(female))
```

```
# A tibble: 2 × 2
  race  female_percent
  <chr>          <dbl>
1 b              0.775
2 w              0.764
```

b.

Computer skill are roughly balanced with around 80% of individuals in each group with computer skills

```
# group by race and calculate the proportion of individuals with computer skills
bm |>
  group_by(race) |>
  summarise(cs_percent = mean(computerskills))
```

```
# A tibble: 2 × 2
  race  cs_percent
  <chr>      <dbl>
1 b          0.832
2 w          0.809
```

c.

The average education level is very close between the two groups and the number of previous jobs are close too. The average number of previous jobs is around 3.6.

```
bm |>
  group_by(race) |>
  summarise(education_avg = mean(education),
            ofjobs_avg = mean(ofjobs))
```

```
# A tibble: 2 × 3
  race  education_avg ofjobs_avg
  <chr>         <dbl>      <dbl>
1 b              3.62       3.66
2 w              3.62       3.66
```

d.

White individuals have slightly higher years of education on average but they also have higher standard deviation. Both groups have similar mean and variability.

```
bm |>
  group_by(race) |>
  summarise(yearsexp_mean = mean(yearsexp),
            yearsexp_sd = sd(yearsexp))
```

```
# A tibble: 2 × 3
  race  yearsexp_mean yearsexp_sd
  <chr>         <dbl>       <dbl>
1 b              7.83        5.01
2 w              7.86        5.08
```

e.

We want to ensure that the treatment (African-American sounding names) are randomly assigned across the resumes. Hence, we want to check that the characteristics of the two groups are similar to each other on average.

f.

The proportion of males with computer skills are significantly lower than that for females. The average level of education for male is higher than female. This is because the authors use nearly exclusively female names for administrative and clerical jobs to increase callback rates. These jobs typically require computer skills but lower education. Since the focus of the research is on race differentials, the gender imbalance in certain characteristics does not invalidate its conclusion as long as gender is balanced across each race group. We have shown this is the case in (a).

```
bm |>
  group_by(sex) |>
  summarise(cs_avg = mean(computerskills),
            edu_avg = mean(education))
```

```
# A tibble: 2 × 3
  sex   cs_avg edu_avg
  <chr>  <dbl>   <dbl>
1 f      0.868    3.58
2 m      0.662    3.73
```

Exercise 5

a.

Calculate the average callback rate for all resumes

The average callback rate for all resumes in Table 1 of the paper

$$9.65 * 0.5 + 6.45 * 0.5 = 8.05$$

This is almost identical to the average callback rate I obtained below.

```
bm |>
  summarise(callback_avg = mean(call)*100)
```

```
# A tibble: 1 × 1
  callback_avg
         <dbl>
1         8.05
```

b.

On average, White-sounding names are 50% more likely than African-American names to get a callback. This suggests there is still significant differential treatment by race, even if the other characteristics of the two groups are controlled to be very similar.

```
bm |>
  group_by(race) |>
  summarise(callback_avg = mean(call)*100)
```

```
# A tibble: 2 × 2
  race  callback_avg
  <chr>        <dbl>
1 b             6.45
2 w             9.65
```

c.

The results suggest that within each sex, African Americans have a lower callback rate than White. The racial gap (percentage difference) is slightly larger for males than for females. Within each race, female obtains higher callback rates, likely due to the reason mentioned in 4.f.

```
bm |>
  group_by(race, sex) |>
  summarise(callback_avg = mean(call)*100)
```

```
`summarise()` has grouped output by 'race'. You can override using the
`.groups` argument.

# A tibble: 4 × 3
# Groups:   race [2]
  race  sex   callback_avg
  <chr> <chr>        <dbl>
1 b     f             6.63
2 b     m             5.83
3 w     f             9.89
4 w     m             8.87
```

Exercise 6

a.

```
callback_b <- bm |>
  filter(black == TRUE) |>
  pull(call)

callback_w <- bm |>
  filter(black == FALSE) |>
  pull(call)
```

b.

Test statistics used:

$$t = \frac{(\overline{X}_1 - \overline{X}_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

Assume unequal varainces, compute the degrees of freedom as follows:

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}}$$

```
# obtain sample size, mean and variance for black and white groups
```

```
  sizeb <- length(callback_b)
  meanb <- mean(callback_b)
  varb <- var(callback_b)


  sizew <- length(callback_w)
  meanw <- mean(callback_w)
  varw <- var(callback_w)

# construct test statistics
  test_stats <- (meanw - meanb)/sqrt(varw/sizew + varb/sizeb)
  test_stats
```

```
[1] 4.114705
```

```
# calculate the degrees of freedom
  numerator <- (varw/sizew + varb/sizeb)^2
  denominator <- ((varw/sizew)^2)/(sizew-1) + ((varb/sizeb)^2)/(sizeb-1)
  df <- numerator/denominator
  df
```

```
[1] 4711.602
```

c.

critical value for a two-sided t-test with $\alpha = 0.05$

for each tail, p = 0.025

```
qnorm(0.975)
```

```
[1] 1.959964
```

d.

The result is consistent with Table 1 in which the p-value is almost zero.

```
pnorm(test_stats, lower.tail = FALSE)
```

```
[1] 1.938372e-05
```

e.

We reject the null hypothesis of no difference in callback rates between black- and white-sounding names at 5% significance level. This suggest the difference we found is significant and differential treatment according to race is still prominent in the US labor market.

Exercise 7

a.

Names can carry information about cultural identity and group affiliation. In 1970s, Blacks in racially isolated neighborhoods began to adopt increasingly distinctive names under the influence of Black Power movement while others move towards more assimilative names.It also contains information about socioeconomic status. The study finds that, over time, distinctively Black names became strong predictors of lower socioeconomic status.

b.

The experiment tracks callback rates for interviews, not final hiring decisions or wage outcomes. Although fewer callbacks likely lead to fewer job offers, the study cannot directly measure disparities in actual hiring rates or earnings. This limitation restricts its scope in fully capturing labor market discrimination.

The study signals race indirectly through names. Employers might miss the racial cues in names, leading to underestimation in discrimination. The finding apply specifically to people with racially distinctive names, not all African-Americans since some of them might have neutral names.

The experiment focuses on newspaper ads, just one of many job search channels. It does not account for social

networks, which are another key channel. If African-Americans rely more on networks or if employers using networks discriminate less, the results may not fully represent labor market dynamics.

c.

Distinctive black-sounding names can be correlated with determinants of productivity not captured by a resume. They can also be correlated with lower socioeconomic status, which could cause employers to be biased against perceived socioeconomic background rather than race alone.

d.

Taste-Based Discrimination: This occurs when an employer discriminates against a group due to personal prejudice or a “taste” for preferring one group over another, regardless of productivity or qualifications. This suggests they are willing to incur costs to avoid hiring from a disfavored group.

Statistical Discrimination: This occurs when an employer uses group-based stereotypes or statistical averages to make decisions about individuals, due to incomplete information about their true productivity.

In the BM study, they found significant callback gap between White and Black names, even for identical resumes. This is consistent with taste-based discrimination. If employers have a prejudice against Black individuals and they have negative perception about distinctive black names, they may systematically reject resumes with Black names out of personal bias, regardless of qualifications. However, the study does not directly measure employers’ preferences and hence the taste-based discrimination is only inferred.

The BM results are also consistent with statistical discrimination.Employers may use Black names as proxies for unobservable characteristics of the individual such as educational quality.They might also infer the socioeconomic backgrounds from names and associate Black names with lower average productivity, which leads to lower callback rates for black-sounding names. However, Fryer and Levitt’s data show that Black names are not causally linked to worse life outcomes. If names were strong signals of productivity, we should expect impact on labour market outcomes, even after controlling for socioeconomic backgrounds.

e.

In the 1960s, Black and White Americans chose relatively similar names for their children. However, during the early 1970s, there was a drastic change in naming patterns, particularly among Blacks in racially segregated neighborhoods, who began giving their children highly distinctive names. This shift in naming patterns is likely to be triggered by the rise of Black Power movement. Among Black children born after the 1970s, a child’s name became a strong indicator of socioeconomic status. Fryer and Levitt also found no evidence that having a distinctively Black name causally harms a person’s life outcomes after controlling for background characteristics such as family and neighborhood circumstances.

f.

One possibility is that discrimination can be stage dependent.The discrimination observed by BM occurs primarily at the resume-screening stage, where names are the only racial cue. Fryer and Levitt suggest that once race is directly observed in interviews or hiring, the effect of names may diminish because employers can assess race directly, and other factors become more important. If discrimination primarily occurs at the callback stage, it may not significantly affect long-term outcome as suggested in Fryer and Levitt’s findings.

Another point is that black names may signal productivity beyond resume information. Black names may correlate with unobservable productivity factors not captured in BM’s resumes. Employers use names for statistical discrimination, assuming Black-named candidates are less productive on average due to socioeconomic status (SES) correlations. Fryer and Levitt’s controls for birth circumstances account for these factors, explaining why they find no negative name effect on outcomes.

PS1_FREDR

AUTHOR
1093122

Exercise 1: Getting familiar with the FRED API

Install the fredr package and load the library; Set API key

```
library(tidyverse)
library(fredr)
library(tibble)
my_key <- Sys.getenv("api_key")
fredr_set_key(my_key)
```

get access to GDP quarterly data from 1947

```
fredr(series_id = "GDP")
```

```
# A tibble: 317 × 5
   date       series_id value realtime_start realtime_end
   <date>     <chr>     <dbl> <date>         <date>
 1 1946-01-01 GDP         NA  2025-04-30     2025-04-30
 2 1946-04-01 GDP         NA  2025-04-30     2025-04-30
 3 1946-07-01 GDP         NA  2025-04-30     2025-04-30
 4 1946-10-01 GDP         NA  2025-04-30     2025-04-30
 5 1947-01-01 GDP        243. 2025-04-30     2025-04-30
 6 1947-04-01 GDP        246. 2025-04-30     2025-04-30
 7 1947-07-01 GDP        250. 2025-04-30     2025-04-30
 8 1947-10-01 GDP        260. 2025-04-30     2025-04-30
 9 1948-01-01 GDP        266. 2025-04-30     2025-04-30
10 1948-04-01 GDP        273. 2025-04-30     2025-04-30
# ℹ 307 more rows
```

Modify the API call to retrieve quarterly GDP data from January 01, 2000, and save it in a tibble called gdp.

```
gdp <- fredr(series_id = "GDP",
             observation_start = as.Date("2000-01-01"))
```

Find out what other time series fredr has

```
fredr_category_children(category_id = 0)
```

```
# A tibble: 8 × 3
     id name                                    parent_id
  <int> <chr>                                       <int>
1 32991 Money, Banking, & Finance                       0
2    10 Population, Employment, & Labor Markets         0
3 32992 National Accounts                               0
4     1 Production & Business Activity                  0
5 32455 Prices                                          0
6 32263 International Data                              0
7  3008 U.S. Regional Data                              0
8 33060 Academic Data                                   0
```

Retrieve unemployment rates

```
fredr_series_search_text("unemployment")
```

```
# A tibble: 1,000 × 16
   id        realtime_start realtime_end title observation_start observation_end
   <chr>     <chr>          <chr>        <chr> <chr>             <chr>
 1 UNRATE    2025-05-07     2025-05-07   Unem… 1948-01-01        2025-04-01
 2 UNRATENSA 2025-05-07     2025-05-07   Unem… 1948-01-01        2025-04-01
 3 UNEMPLOY  2025-05-07     2025-05-07   Unem… 1948-01-01        2025-04-01
 4 NROU      2025-05-07     2025-05-07   Nonc… 1949-01-01        2035-10-01
 5 CCSA      2025-05-07     2025-05-07   Cont… 1967-01-07        2025-04-19
```

```
 6 CCNSA     2025-05-07     2025-05-07   Cont… 1967-01-07        2025-04-19
 7 LNS14000… 2025-05-07     2025-05-07   Unem… 1972-01-01        2025-04-01
 8 LNU03000… 2025-05-07     2025-05-07   Unem… 1948-01-01        2025-04-01
 9 LNU04000… 2025-05-07     2025-05-07   Unem… 1972-01-01        2025-04-01
10 U6RATE    2025-05-07     2025-05-07   Tota… 1994-01-01        2025-04-01
# ℹ 990 more rows
# ℹ 10 more variables: frequency <chr>, frequency_short <chr>, units <chr>,
#   units_short <chr>, seasonal_adjustment <chr>,
#   seasonal_adjustment_short <chr>, last_updated <chr>, popularity <int>,
#   group_popularity <int>, notes <chr>
```

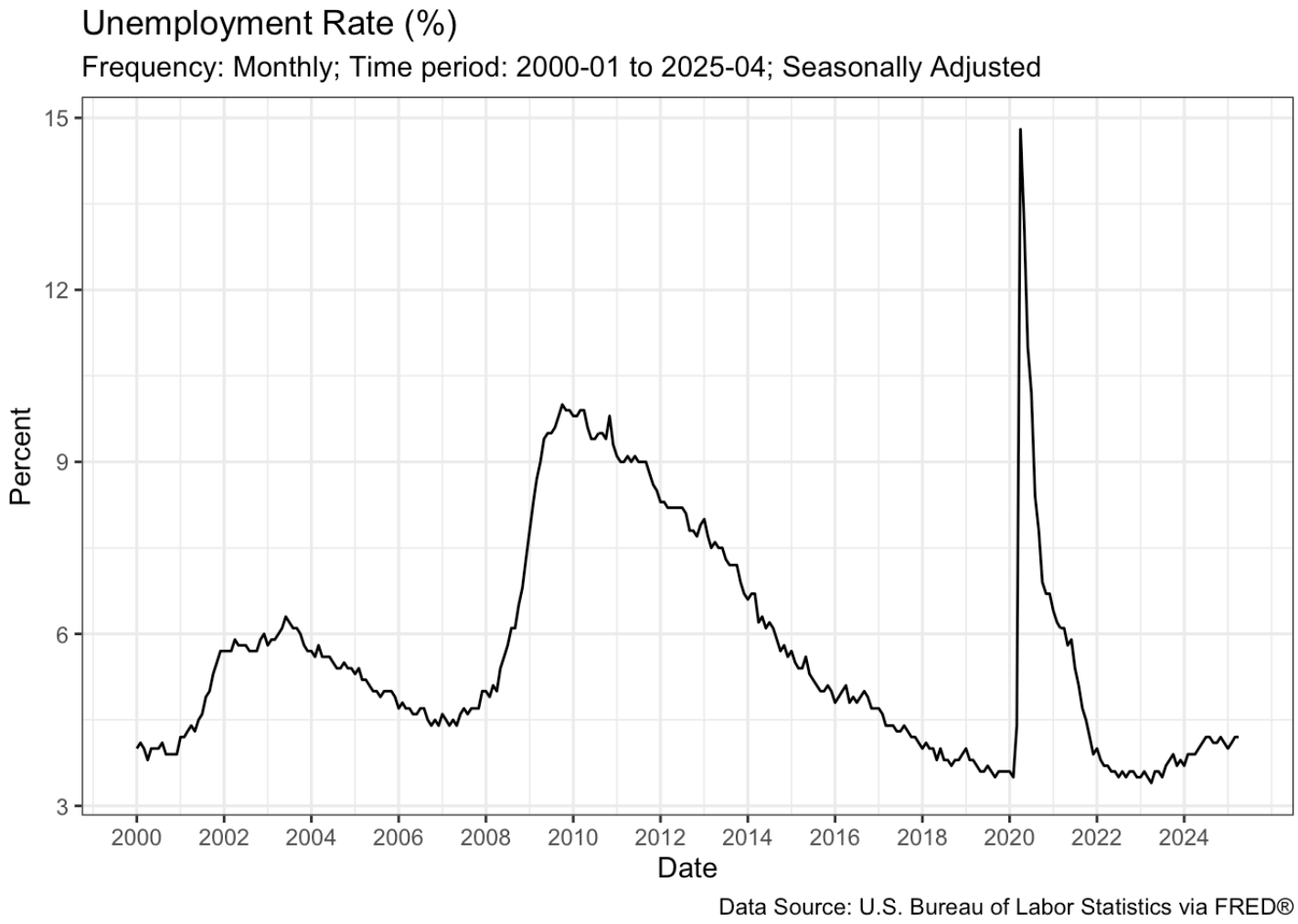
```
u <- fredr(series_id = "UNRATE", observation_start = as.Date("2000-01-01"))
u
```

```
# A tibble: 304 × 5
   date       series_id value realtime_start realtime_end
   <date>     <chr>     <dbl> <date>         <date>
 1 2000-01-01 UNRATE      4   2025-05-02     2025-05-02
 2 2000-02-01 UNRATE      4.1 2025-05-02     2025-05-02
 3 2000-03-01 UNRATE      4   2025-05-02     2025-05-02
 4 2000-04-01 UNRATE      3.8 2025-05-02     2025-05-02
 5 2000-05-01 UNRATE      4   2025-05-02     2025-05-02
 6 2000-06-01 UNRATE      4   2025-05-02     2025-05-02
 7 2000-07-01 UNRATE      4   2025-05-02     2025-05-02
 8 2000-08-01 UNRATE      4.1 2025-05-02     2025-05-02
 9 2000-09-01 UNRATE      3.9 2025-05-02     2025-05-02
10 2000-10-01 UNRATE      3.9 2025-05-02     2025-05-02
# ℹ 294 more rows
```

Exercise 2: Plotting macroeconomic patterns

a. Plot the monthly unemployment rates stored in u using ggplot2

```
u |>
  ggplot(aes(x = date, y = value)) +
  geom_line() +
  scale_x_date(
  breaks = seq(as.Date("2000-01-01"), as.Date("2025-01-01"), by = "2 years"),
  date_labels = "%Y"
)+
  xlab("Date") +
  ylab("Percent")+
  labs(title = "Unemployment Rate (%)",
       subtitle = "Frequency: Monthly; Time period: 2000-01 to 2025-04; Seasonally Adjusted",
       caption = "Data Source: U.S. Bureau of Labor Statistics via FRED®") +
  theme_bw()
```

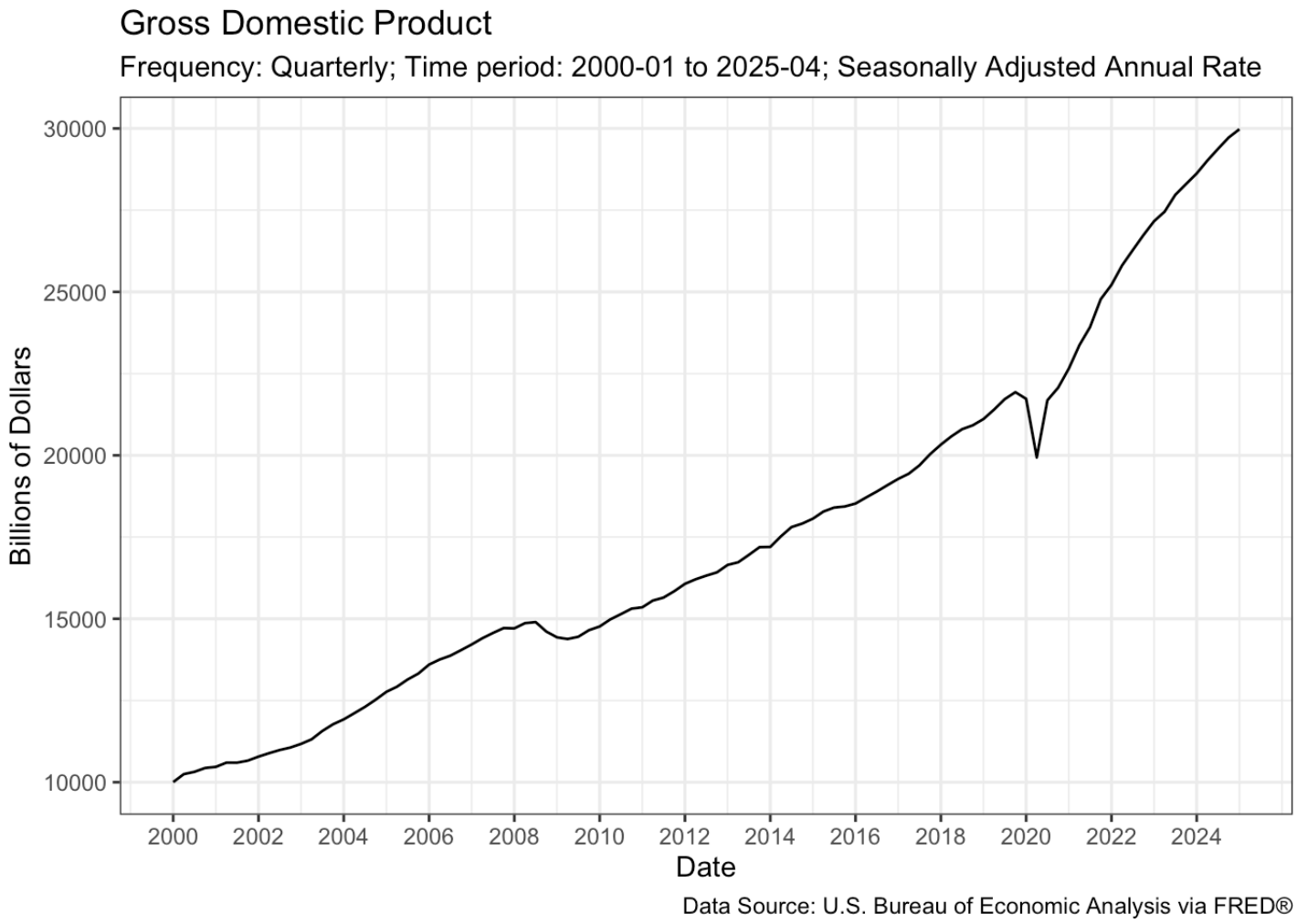


b. Description of unemployment rates change during the business cycles.

Unemployment rates increase significantly during recessions. In 2001, it increased from around 4% to 6%. In 2008, it increased from 5% to around 10%. During the Covid-19 recession in 2020, it increased sharply from 3% to around 15%.Unemployment then decreases during economic recoveries and economic booms, as shown by the downward trend after each recession.

c. GDP Plot

```
gdp |>
  ggplot(aes(x = date, y = value)) +
  geom_line() +
  scale_x_date(
  breaks = seq(as.Date("2000-01-01"), as.Date("2025-01-01"), by = "2 years"),
  date_labels = "%Y"
)+
  xlab("Date") +
  ylab("Billions of Dollars") +
  labs(title = "Gross Domestic Product",
       subtitle =
      "Frequency: Quarterly; Time period: 2000-01 to 2025-04; Seasonally Adjusted Annual Rate",
       caption = "Data Source: U.S. Bureau of Economic Analysis via FRED®") +
theme_bw()
```



d. Create quarter-over-quarter growth and annualised QoQ for plotting GDP growth

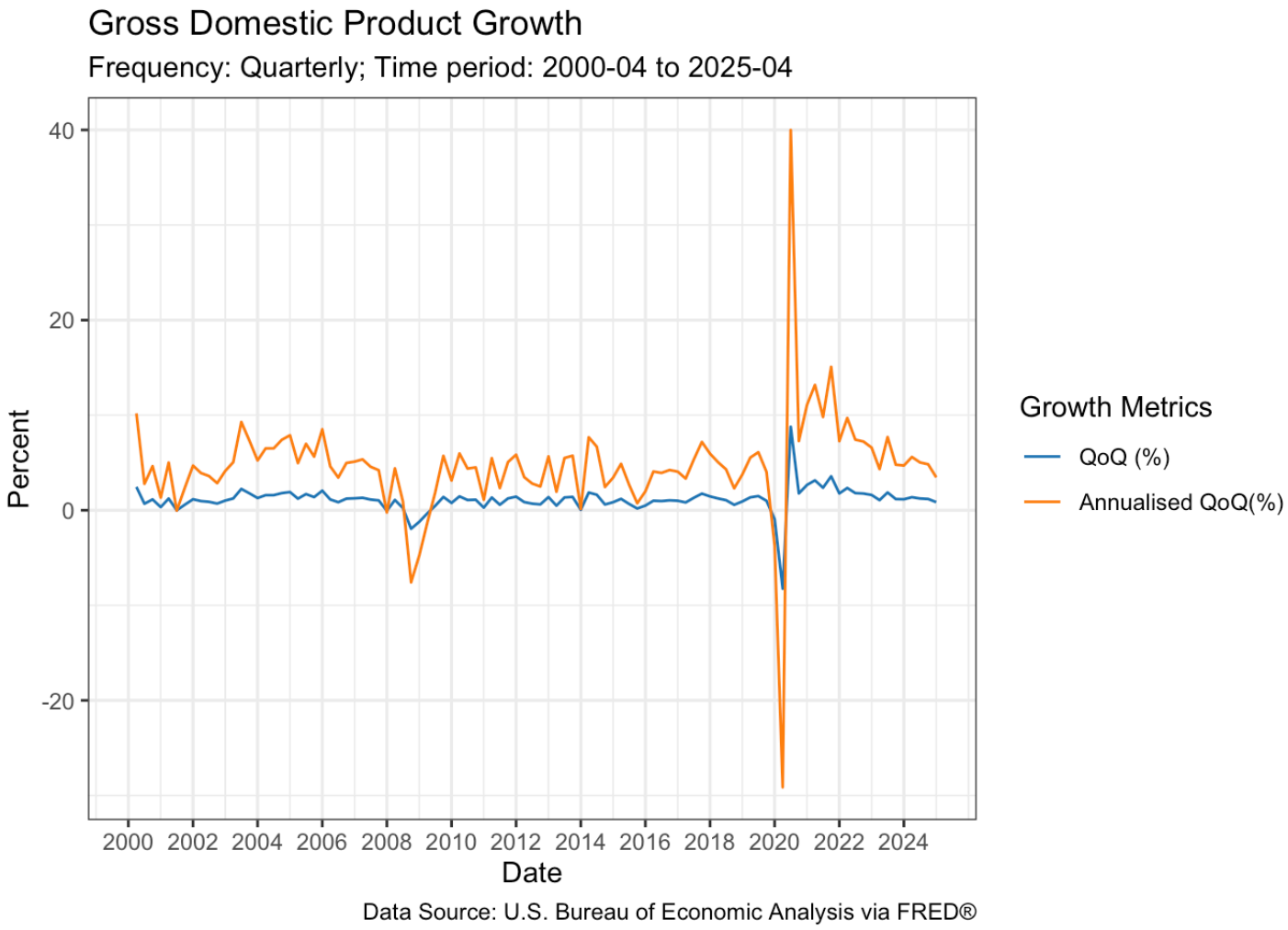
```
gdp <- gdp |>
  mutate(qoq = ((value - lag(value))/lag(value)),
         qoq_annualised = (1 + qoq )^4 - 1, # as if the qoq growth continues for a year, what would b
         qoq = qoq*100,
         qoq_annualised = qoq_annualised*100)
```

We do not have a previous period for the first observation of this dataset (2000-01-01). Therefore, for the first period, we do not obtain a quarter-over-quarter growth and the values are missing.

e. Plot both measures of quarterly growth in the same plot

```
gdp |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = qoq, color = "qoq")) +
  geom_line(aes(y = qoq_annualised, color = "qoq_annualised"))+
  scale_x_date(
  breaks = seq(as.Date("2000-01-01"), as.Date("2025-01-01"), by = "2 years"),
  date_labels = "%Y"
)+
  xlab("Date") +
  ylab("Percent")+
  labs(title = "Gross Domestic Product Growth",
       subtitle = "Frequency: Quarterly; Time period: 2000-04 to 2025-04",
       caption = "Data Source: U.S. Bureau of Economic Analysis via FRED®") +
  scale_color_manual(name = "Growth Metrics",
                     values = c("qoq" = "#1f77b4",
                      "qoq_annualised" = "#ff7f0e"),
                     labels = c("qoq" = "QoQ (%)",
                                "qoq_annualised" = "Annualised QoQ(%)"))+
  theme_bw()
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_line()`).
Removed 1 row containing missing values or values outside the scale range
(`geom_line()`).
```



f. Pull QoQ growth from FRED API directly and fix the missing values

```
gdp_growth <- fredr(series_id = "GDP", units = "pch", observation_start = as.Date("2000-01-01"))
gdp_growth
```

```
# A tibble: 101 × 5
   date       series_id    value realtime_start realtime_end
   <date>     <chr>        <dbl> <date>         <date>
 1 2000-01-01 GDP        1.03    2025-05-01     2025-05-01
 2 2000-04-01 GDP        2.45    2025-05-01     2025-05-01
 3 2000-07-01 GDP        0.687   2025-05-01     2025-05-01
 4 2000-10-01 GDP        1.14    2025-05-01     2025-05-01
 5 2001-01-01 GDP        0.330   2025-05-01     2025-05-01
 6 2001-04-01 GDP        1.23    2025-05-01     2025-05-01
 7 2001-07-01 GDP       -0.00925 2025-05-01     2025-05-01
 8 2001-10-01 GDP        0.589   2025-05-01     2025-05-01
 9 2002-01-01 GDP        1.15    2025-05-01     2025-05-01
10 2002-04-01 GDP        0.964   2025-05-01     2025-05-01
# ℹ 91 more rows
```

The quarterly growths rates match to my qoq variable.

```
gdp$qoq[1] <- gdp_growth$value[1]
gdp$qoq_annualised[1] <- ((1+gdp$qoq[1]*0.01)^4 - 1)*100
gdp
```

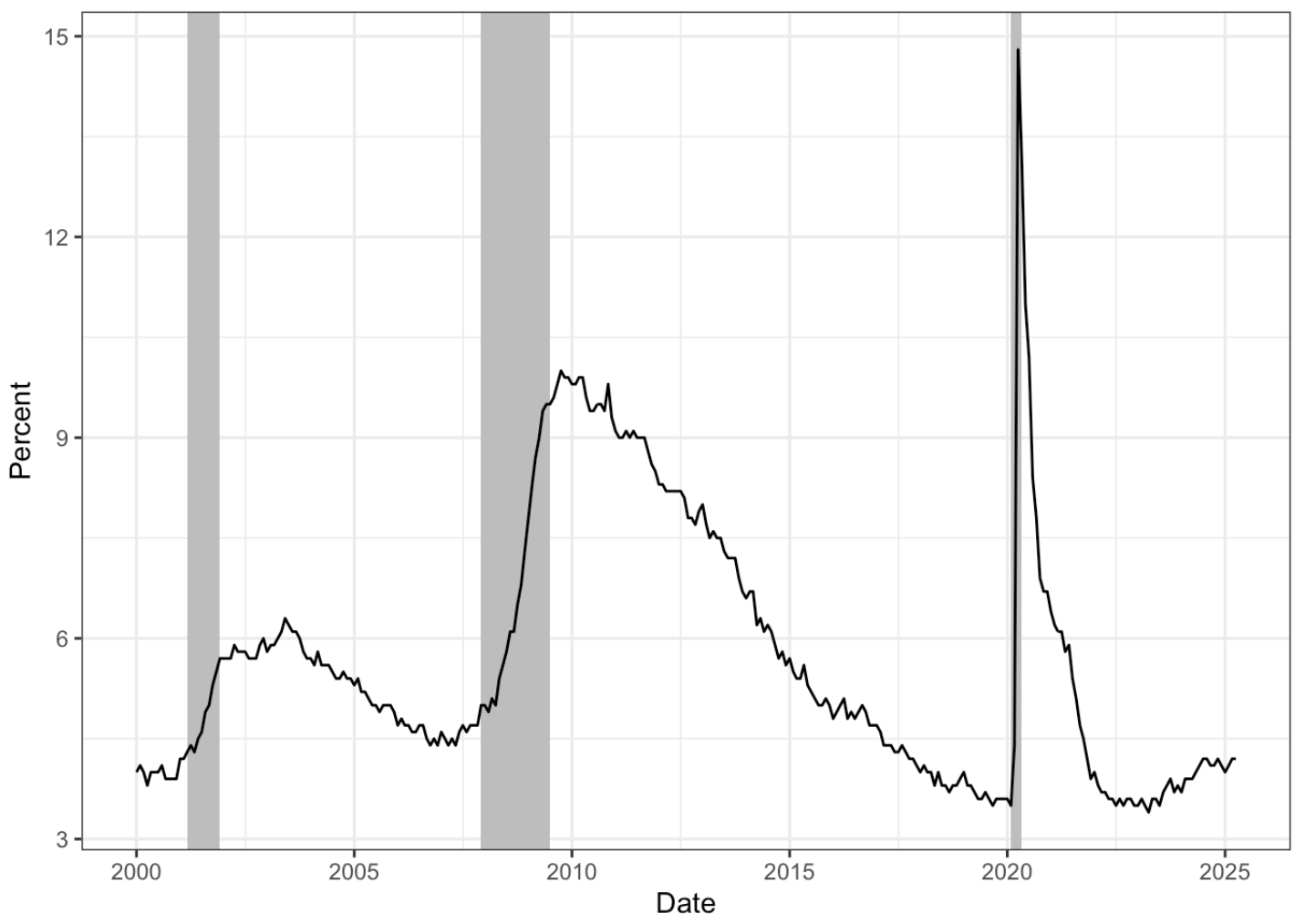
```
# A tibble: 101 × 7
   date       series_id  value realtime_start realtime_end      qoq
   <date>     <chr>      <dbl> <date>         <date>          <dbl>
 1 2000-01-01 GDP       10002. 2025-05-02     2025-05-02    1.03
 2 2000-04-01 GDP       10248. 2025-05-02     2025-05-02    2.45
 3 2000-07-01 GDP       10318. 2025-05-02     2025-05-02    0.687
 4 2000-10-01 GDP       10436. 2025-05-02     2025-05-02    1.14
 5 2001-01-01 GDP       10470. 2025-05-02     2025-05-02    0.330
 6 2001-04-01 GDP       10599  2025-05-02     2025-05-02    1.23
 7 2001-07-01 GDP       10598. 2025-05-02     2025-05-02   -0.00925
 8 2001-10-01 GDP       10660. 2025-05-02     2025-05-02    0.589
 9 2002-01-01 GDP       10784. 2025-05-02     2025-05-02    1.15
10 2002-04-01 GDP       10887. 2025-05-02     2025-05-02    0.964
# ℹ 91 more rows
# ℹ 1 more variable: qoq_annualised <dbl>
```

Exercise 3: Spotting recessions

```
recessions <- tibble(
  start = as.Date(c("2001-03-01", "2007-12-01", "2020-02-01")),
  end   = as.Date(c("2001-11-30", "2009-06-30", "2020-04-30"))
)
```

a. Plot unemployment rates and add shading for recession periods

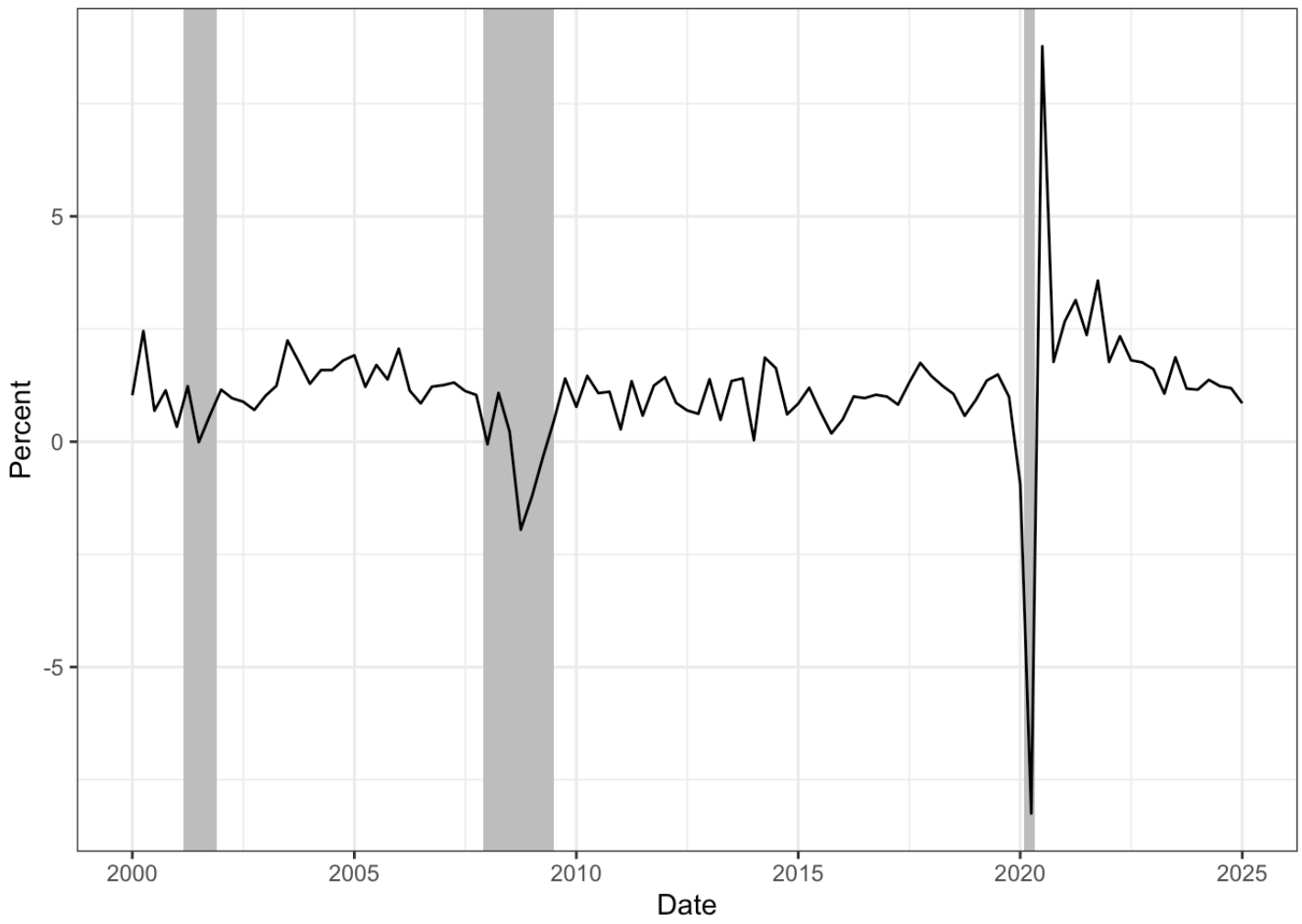
```
uplot <-  u |>
  ggplot(aes(x = date, y = value)) +
   geom_rect(
    data = recessions,
    aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf),
    fill = "grey",
    inherit.aes = FALSE) +
  geom_line() +
    scale_x_date(
  breaks = seq(as.Date("2000-01-01"),
               as.Date("2025-01-01"),
               by = "5 years"),
  date_labels = "%Y"
)+
  xlab("Date") +
  ylab("Percent")+
  theme_bw()
uplot
```



b. Plot QoQ GDP growth with recession shading using geom_rect()

```
gdp_growth_plot <- gdp_growth |>
  ggplot(aes(x = date, y = value)) +
   geom_rect(
    data = recessions,
    aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf), # Inf / -Inf to cover the whole column
    fill = "grey", # use grey color
    inherit.aes = FALSE) +
  geom_line() +
  scale_x_date(
  breaks = seq(as.Date("2000-01-01"),
```

```
              as.Date("2025-01-01"), by = "5 years"),
  date_labels = "%Y"
)+
  xlab("Date") +
  ylab("Percent")+
  theme_bw()
gdp_growth_plot
```



c. Combine the plots next to each other

```
library(patchwork)
uplot + gdp_growth_plot+
  plot_annotation(title = "Unemployment (Left, %) and GDP Growth (Right, %)",
                  caption = "Data Source: FREDR and NBER")
```