

PS2_Football

AUTHOR
1093122

Exercise 1

```
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

```
library(dplyr)  
football <- read_csv('https://ditraglia.com/data/fair_football.csv')
```

Rows: 1582 Columns: 10
— Column specification —
Delimiter: ","
dbl (10): SPREAD, H, MAT, SAG, BIL, COL, MAS, DUN, REC, LV

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.

```
football
```

```
# A tibble: 1,582 × 10  
  SPREAD    H  MAT  SAG  BIL  COL  MAS  DUN  REC  LV  
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1     34     1    7   31   28   17   38   14    0   24  
2     29    -1   34   29   10   41   26   18  33.3  13.5  
3     10    -1  -16  -23  -33    5  -12  -25   8.33 -10.5  
4    -11     1    2   -8   -8   -7   -2   -4    0    3  
5     35    -1   35   35   38   25   25   28  25    5  
6     -2     1   29   36   17   25   20   11  33.3  11.5  
7     11     1   35   39   28   40   30   34  41.7   10  
8     20     1   29   13   12   37   13   26  25    7.5  
9      7     1   40   41   -7   45   36   43  66.7  11.5  
10    20    -1   61   37   36   80   51   35  75   11  
# i 1,572 more rows
```

calculate the home field advantage

```
# only need games where H is 1 or -1  
home <- football |>  
  filter(H == 1 | H == -1)  
  
home |>  
  mutate(home_win = ((H == 1 & SPREAD > 0) | (H == -1 & SPREAD < 0))) |>  
  summarise(home_win_percent = mean(home_win))
```

```
# A tibble: 1 × 1  
  home_win_percent  
  <dbl>  
1           0.586
```

```
home <- home |>  
  mutate(homescore_more = if_else(H == 1, SPREAD, -SPREAD))
```

```
home |>
  summarise(more_points_avg = mean(homescore_more))
```

```
# A tibble: 1 × 1
  more_points_avg
      <dbl>
1             4.86
```

Exercise 2

The intercept is given by the difference between the averages of SPREAD and H. Given the designation of “Team A” and “Team B” is completely arbitrary, H is expected to be zero on average, the intercept is simply the average of SPREAD. Since the football game is a zero-sum game, the average of SPREAD is also expected to be zero. A non-zero intercept would imply a consistent bias in one direction, which implies the team labeling is not arbitrary. For all the other predictor variables, their values should also be equal to zero on average. The numbers obtained by reversing team A and B are symmetric with opposite signs. Hence, the intercept should be zero in any regression predicting SPREAD given completely arbitrary designation of teams.

Exercise 3

Interpretation: The estimated coefficient on H represents that if team A is the home team, it earns 4.857 points more than team B on average.
Inference: The hypothesis test below suggests that the coefficient is significantly different from zero with $p < 0.001$.
Model fit: both multiple R-squared and adjusted R-squared are close to zero, which suggests a relatively poor fit.

```
# Regress SPREAD on H without intercept
reg1 <- lm(SPREAD ~ H-1, football)
summary(reg1)
```

Call:
lm(formula = SPREAD ~ H - 1, data = football)

Residuals:

Min	1Q	Median	3Q	Max
-61.143	-6.143	6.143	17.857	68.143

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
H	4.857	0.537	9.044	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.66 on 1581 degrees of freedom
Multiple R-squared: 0.04919, Adjusted R-squared: 0.04859
F-statistic: 81.8 on 1 and 1581 DF, p-value: < 2.2e-16

Hypothesis testing with $H_0 : \beta_1 = 0$

We reject the null hypothesis.

```
library(car)
linearHypothesis(reg1, "H = 0")
```

Linear hypothesis test:
H = 0

Model 1: restricted model
Model 2: SPREAD ~ H - 1

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1582	709639				
2	1581	674729	1	34910	81.801	< 2.2e-16 ***

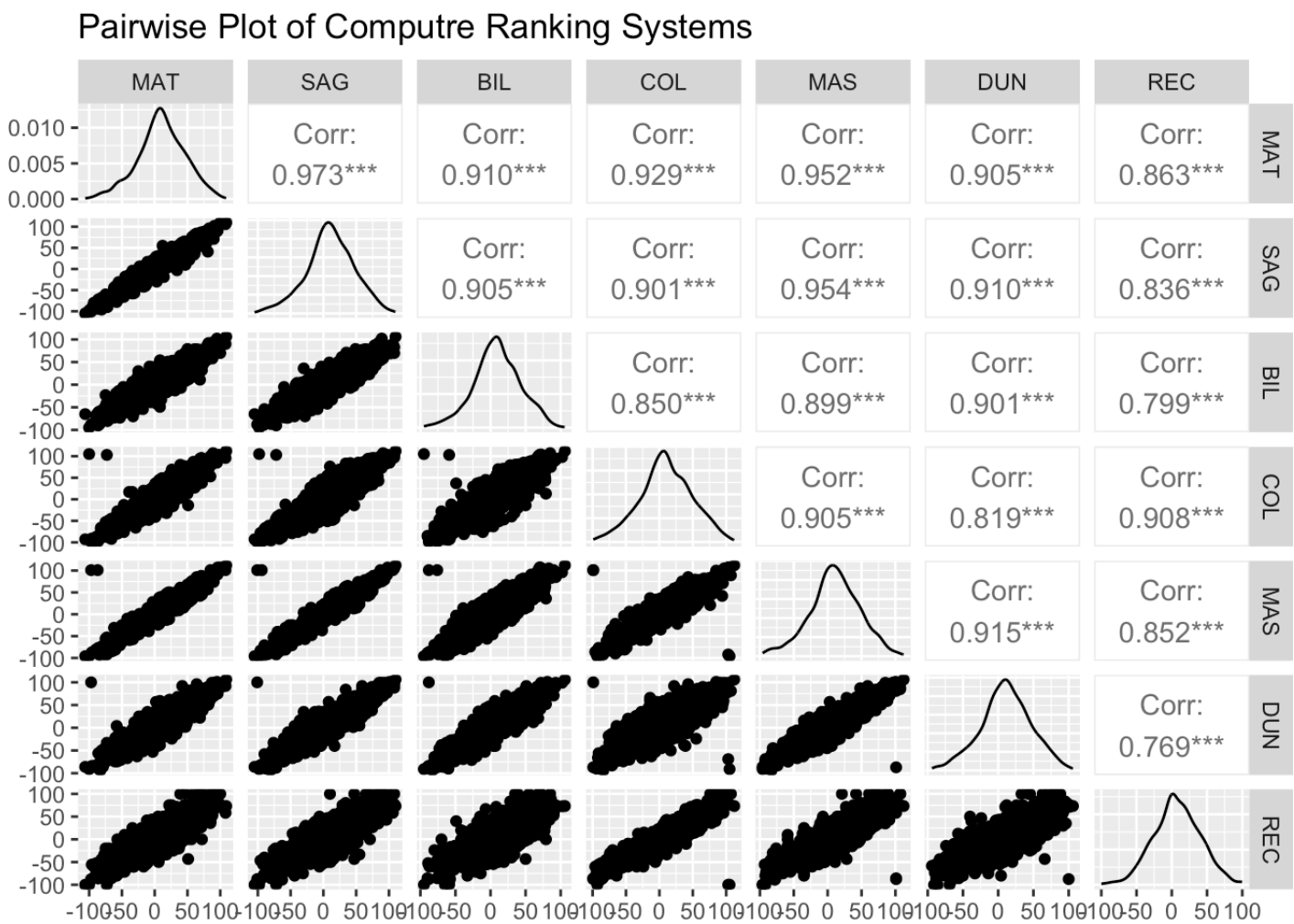
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Exercise 4

The lower triangle: it is the scatterplot between variable pairs.
The upper triangle: it gives the correlation coefficients.
Diagonal: Univariate distribution - it gives the histograms of each individual variable.

Interpretation: all the variables are positively correlated with each other. This positive relationship is strong as the correlation coefficients are close to 1. The histogram shows the distribution of each variable tends to be symmetric around zero, which corresponds to the random labeling of teams.

```
library(GGally)
football |>
  ggpairs(columns = c("MAT", "SAG","BIL", "COL", "MAS", "DUN", "REC"),
    title = "Pairwise Plot of Computre Ranking Systems")
```



Exercise 5

Statistical Inference: based on the t-test below, use a significance level $\alpha = 10\%$, variables MAT and MAS are not statistically significant. The same result is given by the F-test with $H_0 : \beta_{MAT} = \beta_{MAS} = 0$. Hence, these two variables may not add additional predictive information.

```
reg2 <- lm(SPREAD ~ . - LV - 1, football)
summary(reg2)
```

Call:
lm(formula = SPREAD ~ . - LV - 1, data = football)

Residuals:

Min	1Q	Median	3Q	Max
-53.542	-9.134	2.150	11.736	56.963

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
H	4.267073	0.436668	9.772	< 2e-16 ***
MAT	-0.099306	0.060804	-1.633	0.102624
SAG	0.248165	0.054817	4.527	6.43e-06 ***

```
BIL  0.080436    0.034244    2.349 0.018953 *
COL -0.062588    0.035894   -1.744 0.081410 .
MAS -0.007075    0.044624   -0.159 0.874047
DUN  0.118512    0.033769    3.509 0.000462 ***
REC  0.080412    0.030460    2.640 0.008374 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 16.53 on 1574 degrees of freedom
Multiple R-squared: 0.3942, Adjusted R-squared: 0.3911
F-statistic: 128 on 8 and 1574 DF, p-value: < 2.2e-16

Hypothesis Testing with $H_0 : \beta_{MAT} = \beta_{MAS} = 0$

We do not reject the null hypothesis.

```
linearHypothesis(reg2, c("MAT = 0", "MAS = 0"))
```

Linear hypothesis test:
MAT = 0
MAS = 0

Model 1: restricted model
Model 2: SPREAD ~ (H + MAT + SAG + BIL + COL + MAS + DUN + REC + LV) -
LV - 1

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1576	430638				
2	1574	429879	2	758.07	1.3878	0.2499

Re-estimate the model by removing these two variables

```
reg3 <- lm(SPREAD ~ . - LV -MAT -MAS - 1, football)
summary(reg3)
```

Call:
lm(formula = SPREAD ~ . - LV - MAT - MAS - 1, data = football)

Residuals:

	Min	1Q	Median	3Q	Max
	-53.379	-9.159	2.226	11.953	60.007

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
H	4.31812	0.43495	9.928	< 2e-16 ***
SAG	0.18662	0.03809	4.899	1.06e-06 ***
BIL	0.07203	0.03387	2.127	0.033587 *
COL	-0.08575	0.03279	-2.615	0.009014 **
DUN	0.10866	0.03151	3.449	0.000578 ***
REC	0.07666	0.03017	2.541	0.011141 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.53 on 1576 degrees of freedom
Multiple R-squared: 0.3932, Adjusted R-squared: 0.3908
F-statistic: 170.2 on 6 and 1576 DF, p-value: < 2.2e-16

Yes, it is possible to make better predictions as long as each computer ranking system provides statistically significant incremental predictive power. Each ranking system may contain independent information that would improve the prediction. Based on the result, SAG, DUN, COL, BIL and REC are consistently significant, which means they provide non-redundant information and would help improve predictive performance.

Exercise 6

Neither H nor any of the ranking systems is statistically significant after including LV in the regression. Hence, they do not carry independent information beyond that contained in LV. LV is the only statistically significant variable in

the regression.

```
reg4 <- lm(SPREAD ~ LV + H + SAG + BIL + COL + DUN + REC - 1, football)
summary(reg4)
```

Call:
lm(formula = SPREAD ~ LV + H + SAG + BIL + COL + DUN + REC -
1, data = football)

Residuals:
Min 1Q Median 3Q Max
-60.379 -8.469 1.564 11.285 54.636

Coefficients:
Estimate Std. Error t value Pr(>|t|)
LV 1.051782 0.076071 13.826 <2e-16 ***
H 0.729503 0.485981 1.501 0.134
SAG 0.018065 0.037994 0.475 0.635
BIL -0.027867 0.032797 -0.850 0.396
COL -0.005476 0.031518 -0.174 0.862
DUN -0.024891 0.031290 -0.795 0.426
REC 0.018585 0.028804 0.645 0.519

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.61 on 1575 degrees of freedom
Multiple R-squared: 0.4588, Adjusted R-squared: 0.4564
F-statistic: 190.8 on 7 and 1575 DF, p-value: < 2.2e-16

Hypothesis testing with $H_0 : \beta_H = \beta_{SAG} = \beta_{BIL} = \beta_{COL} = \beta_{DUN} = \beta_{REC} = 0$

We do not reject the null hypothesis.

```
linearHypothesis(reg4, c("H = 0", "SAG = 0", "BIL = 0", "COL = 0", "DUN = 0", "REC = 0"))
```

Linear hypothesis test:
H = 0
SAG = 0
BIL = 0
COL = 0
DUN = 0
REC = 0

Model 1: restricted model
Model 2: SPREAD ~ LV + H + SAG + BIL + COL + DUN + REC - 1

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1581	385883				
2	1575	384026	6	1856.8	1.2692	0.2684

Exercise 7

The market is efficient if it fully and correctly reflect all public information. Part 6 shows that the other variables do not contain any independent information that is not already captured by LV. This means the betting market has contained all the public information available. If the betting markets are efficient, the slope in a regression using LV alone should be 1 as LV should give the best prediction of SPREAD. In the regression below, it is shown that the coefficient on LV is close to 1 and is statistically significant. The R-squared values using LV alone is around 0.46. This means around 46% of the variability in actual spread can be explained by LV alone, which shows stronger accuracy compared with the previous settings without LV. The root mean squared error is essentially given by the residual standard error below: on average LV predictions are off by about 15.62 points.

```
reg5 <- lm(SPREAD ~ LV-1, football)
summary(reg5)
```

```
Call:
lm(formula = SPREAD ~ LV - 1, data = football)

Residuals:
    Min       1Q   Median       3Q      Max
-61.244  -9.065   1.043  10.910  54.234

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
LV  1.01436    0.02785   36.42  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.62 on 1581 degrees of freedom
Multiple R-squared:  0.4562,    Adjusted R-squared:  0.4559
F-statistic: 1326 on 1 and 1581 DF,  p-value: < 2.2e-16
```

Hypothesis testing with $H_0 : \beta_{LV} = 1$

We do not reject the null hypothesis.

```
linearHypothesis(reg5, "LV = 1")
```

Linear hypothesis test:
LV = 1

Model 1: restricted model
Model 2: SPREAD ~ LV - 1

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1582	385948				
2	1581	385883	1	64.908	0.2659	0.6061

Exercise 8

```
library(modelsummary)
regressions <- list(reg1, reg2, reg3, reg4, reg5)
modelsummary(regressions, gof_omit = 'Log.Lik|R2 Adj.|AIC|BIC|F', fmt = 2,
              notes = 'Source: Fair & Oster (2007).')
```

	(1)	(2)	(3)	(4)	(5)
H	4.86	4.27	4.32	0.73	
	(0.54)	(0.44)	(0.43)	(0.49)	
MAT		-0.10			
		(0.06)			
SAG		0.25	0.19	0.02	
		(0.05)	(0.04)	(0.04)	
BIL		0.08	0.07	-0.03	
		(0.03)	(0.03)	(0.03)	
COL		-0.06	-0.09	-0.01	
		(0.04)	(0.03)	(0.03)	
MAS		-0.01			
		(0.04)			
DUN		0.12	0.11	-0.02	

		(0.03)	(0.03)	(0.03)	
REC		0.08	0.08	0.02	
		(0.03)	(0.03)	(0.03)	
LV				1.05	1.01
				(0.08)	(0.03)
Num.Obs.	1582	1582	1582	1582	1582
R2	0.049	0.394	0.393	0.459	0.456
RMSE	20.65	16.48	16.50	15.58	15.62

Source: Fair & Oster (2007).

PS2_Monte_Carlo

AUTHOR
1093122

Exercise 1

a.

Lehmer random number generator produces $\{x_n\}$ using the following:

$$x_{n+1} = ax_n \bmod m$$

$\{\frac{x_n}{m}\}$ gives the sequence of iid Uniform (0,1) draws.

```
# Define m and a
m <- 2^16 + 1
a <- 75

# Seed
x0 <- 42

# x %% y is the remainder when x is divided by y
x1 <- (a * x0) %% m

# next element of the sequence
print(x1)
```

[1] 3150

b.

Divide the Lehmer sequence by m to transform it to be within the interval [0,1]. To convert it into a [3, 5] interval, divide the Lehmer sequence by m, multiplied it by two and add three.

$$\{2 \times \frac{x_n}{m} + 3\}$$

```
Lehmer_seq <- c(42, 3150, 39639, 23760, 12501, 20057)

# convert into [0,1] interval
print(Lehmer_seq/m)
```

[1] 0.0006408594 0.0480644521 0.6048339106 0.3625432962 0.1907472115
[6] 0.3060408624

```
# convert into [3,5] interval
print(2*Lehmer_seq/m + 3)
```

[1] 3.001282 3.096129 4.209668 3.725087 3.381494 3.612082

c.

```
runif_zx81 <- function(seed, n, min = 0, max = 1){
  # Set the a and m parameters as specified.
  m <- 2^16 + 1
  a <- 75
  # Add warning messages in case the seed input is negative or larger than m.
  if (seed < 0) {
    "The seed is negative!"
  }
  if (seed > m) {
    "The seed is bigger than m!"
  }
  # Initialize an empty vector to save the draws, and save the start of the
  # sequence as the first draw. Hint: how do you access elements of vectors?
  seq <- c()
```



```
seq[1] <- seed
# Run a for loop to construct as many elements of the sequence as specified by
# the number of draws n.
for (i in 2:n){
  x <- (a * seq[i-1]) %% m
  seq[i] <- x
}
# Adjust the interval of your vector to run from min to max.
seq <- (max - min)*(seq/m) + min
# Return the vector of pseudorandom numbers.
seq
}
```

d.

```
# seed = 42 and n = 1000
data <- runif_zx81(66, 1000)
```

The count of data points in each bin is around 10 using a bin size = 100, which suggests the data points are uniformly distributed between 0 and 1.

```
# histogram
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		

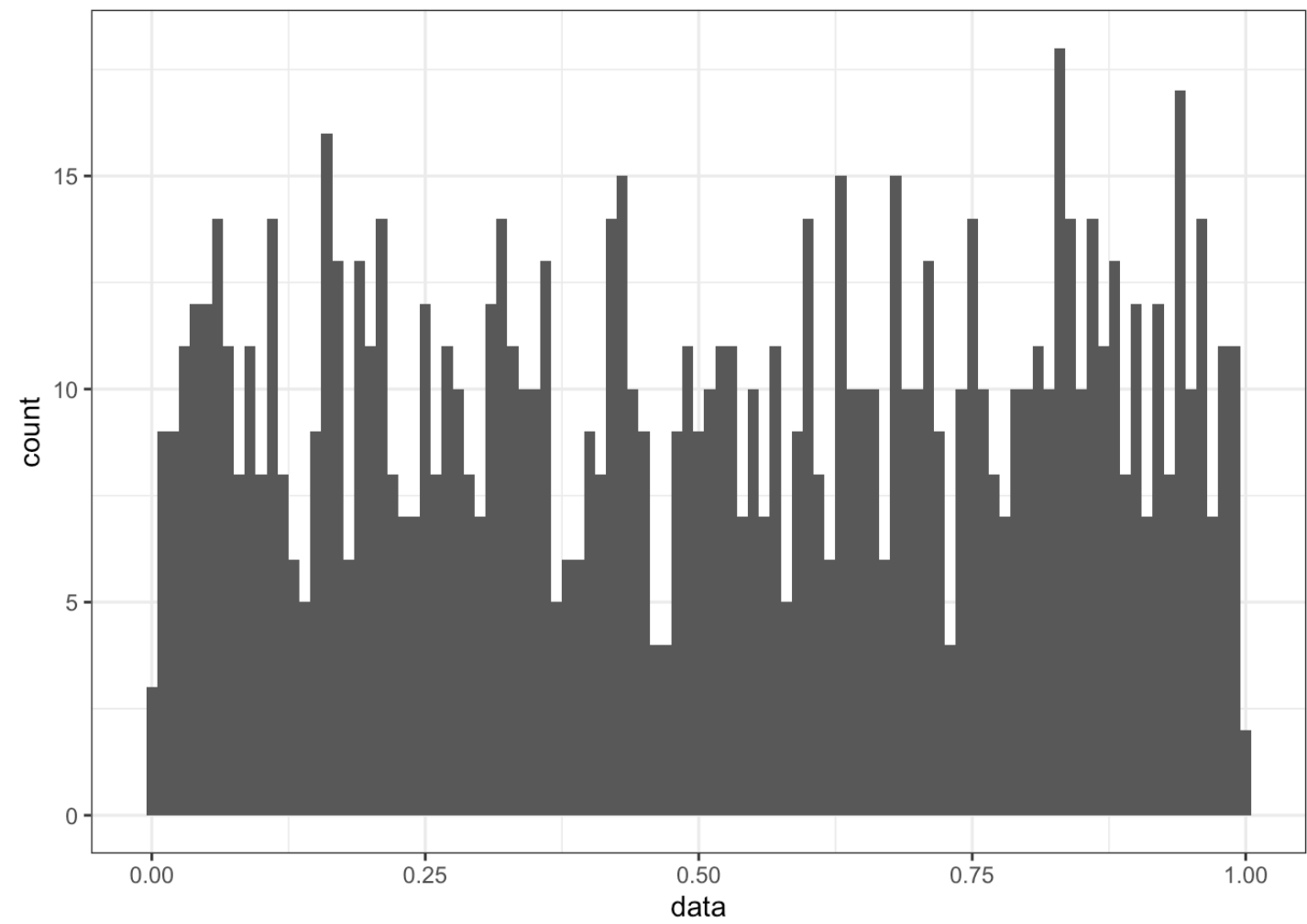
— Conflicts — tidyverse_conflicts() —

✖ dplyr::filter() masks stats::filter()

✖ dplyr::lag() masks stats::lag()

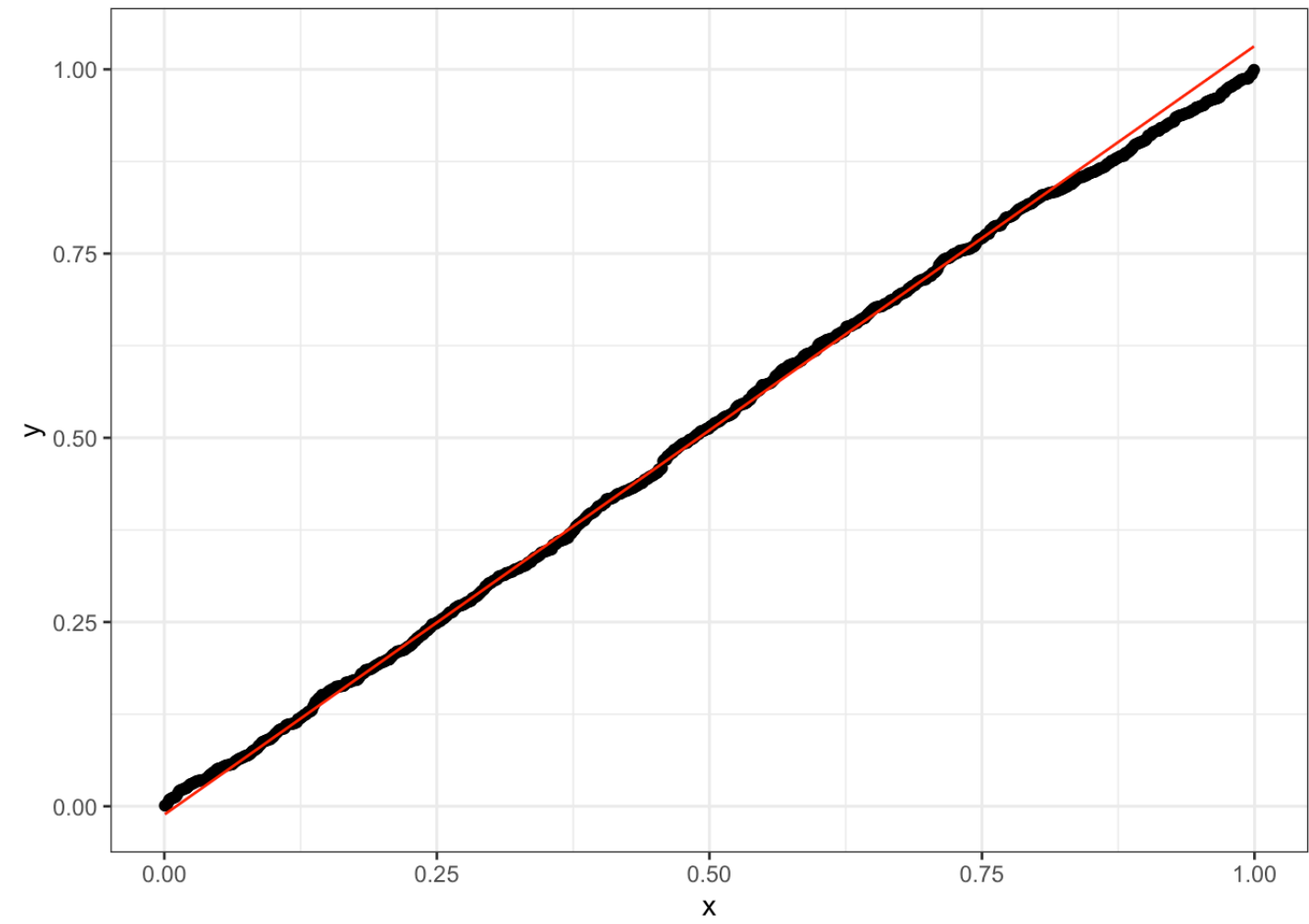
i Use the conflicted package (<<http://conflicted.r-lib.org/>>) to force all conflicts to become errors

```
data <- tibble(data)
data |>
  ggplot(aes(x = data)) +
  geom_histogram(
    binwidth = 0.01
  ) +
  theme_bw()
```



The QQ plot lies quite closely to the 45-degree line. This means the sample quantile values closely match the theoretical quantile values of a uniform distribution.

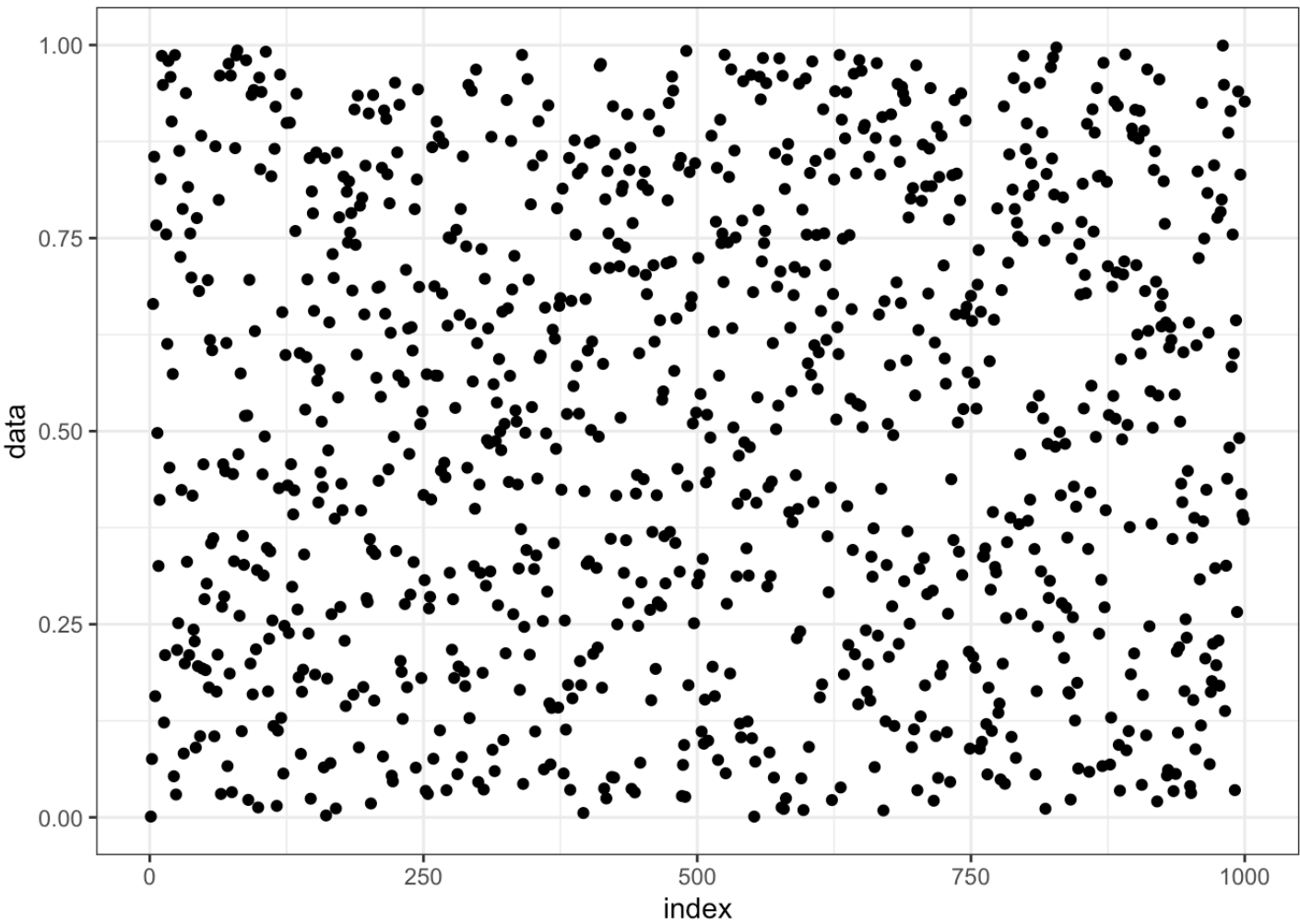
```
# create the QQ plot and use the uniform distribution over [0, 1] as a reference
ggplot(data, aes(sample = data)) +
  stat_qq(distribution = qunif, dparams = list(min = 0, max = 1)) +
  stat_qq_line(distribution = qunif, dparams = list(min = 0, max = 1), color = "red") +
  theme_bw()
```



The points seem to be distributed evenly and randomly between 0 and 1. There is no trend or clustering over the index.

```
# time series plot
data |>
  mutate(index = row_number()) |>
  ggplot(aes(x = index, y = data)) +
  geom_point() +
```

theme_bw()



Exercise 2

a.

```
# Pseudocode for 2.a
# construct U1 and U2 with runif_zx81

# Construct the two new random variables using U1 and U2

# Construct Z1 and Z2 with the new random variables as the Box-Muller Transform suggests
# Z1 and Z2 are standard, normally distributed random variables

# Transform Z1 and Z2 to get V = Z*sd + mean

# Combine V1 and V2 into a new vector Z
```

b.

```
unif_seq <- c(0.5600805, 0.5767570, 0.8858708, 0.9313472, 0.7665961, 0.9763004)

# Divide the unif_seq into two equal halves
U1 <- unif_seq[1:3]
U2 <- unif_seq[4:6]

# Construct the two new random variables using U1 and U2
R <- sqrt(-2*log(U1))
theta <- 2*pi*U2

# Construct Z1 and Z2 with the new random variables as the Box-Muller Transform suggests
Z1 <- R*cos(theta)
Z2 <- R*sin(theta)

# Transform Z1 and Z2 to get V = Z*sd + mean
V1 <- Z1*0.5 + 2
V2 <- Z2*0.5 + 2

# Combine V1 and V2 into a new vector Z
Z <- c(V1, V2)
```

Z

[1] 2.489050 2.054601 2.243431 1.774907 1.478286 1.963481

```

rnorm_zx81 <- function(seed, n, mean, sd) {
  # construct U1 and U2 with runif_zx81
  if (n %% 2 == 0){
    # if n is even
    unif_seq <- runif_zx81(seed, n)
    U1 <- unif_seq[1:(n/2)]
    U2 <- unif_seq[(n/2+1):n]
  } else{
    # if n is odd
    unif_seq <- runif_zx81(seed, n+1)
    U1 <- unif_seq[1:((n+1)/2)]
    U2 <- unif_seq[((n+1)/2 + 1) : (n+1)]
  }

  # Construct the two new random variables using U1 and U2
  R <- sqrt(-2*log(U1))
  theta <- 2*pi*U2

  # Construct Z1 and Z2 with the new random variables as the Box-Muller Transform suggests
  # Z1 and Z2 are standard, normally distributed random variables
  Z1 <- R*cos(theta)
  Z2 <- R*sin(theta)

  # Transform Z1 and Z2 to get V = Z*sd + mean
  V1 <- Z1*sd + mean
  V2 <- Z2*sd + mean

  # remove one element from V1 if n is odd
  if (n %% 2 != 0) {
    V1 <- V1[-1]
  }

  # Combine V1 and V2 into a new vector Z
  Z <- c(V1, V2)
  Z
}

```

d.

```

# Generate 1000 random draws from rnorm_zx81()
data_n <- rnorm_zx81(42, 1000, 0, 1)

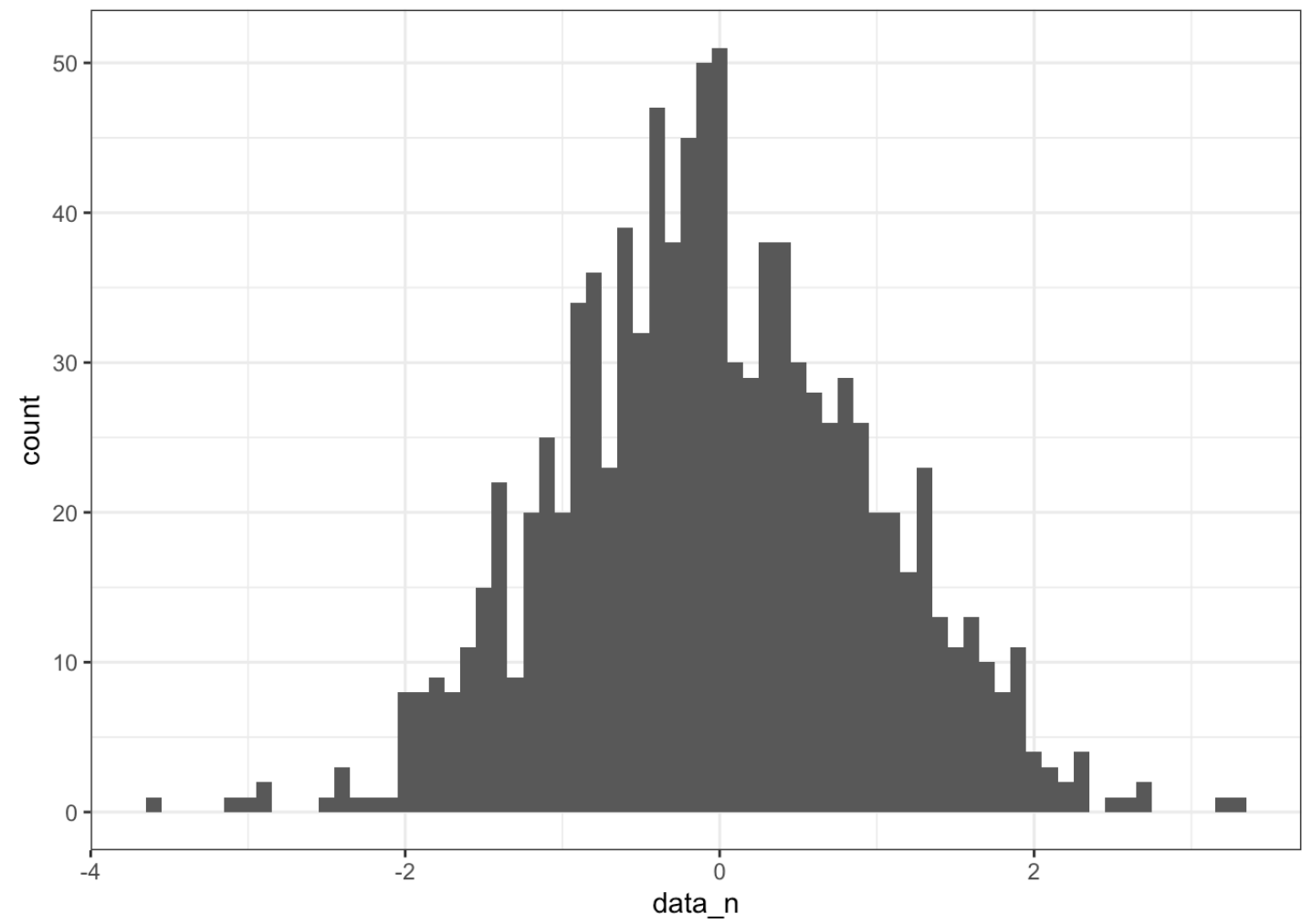
```

The histogram appears to be bell-shaped and symmetric, which matches the pdf of a normal distribution.

```

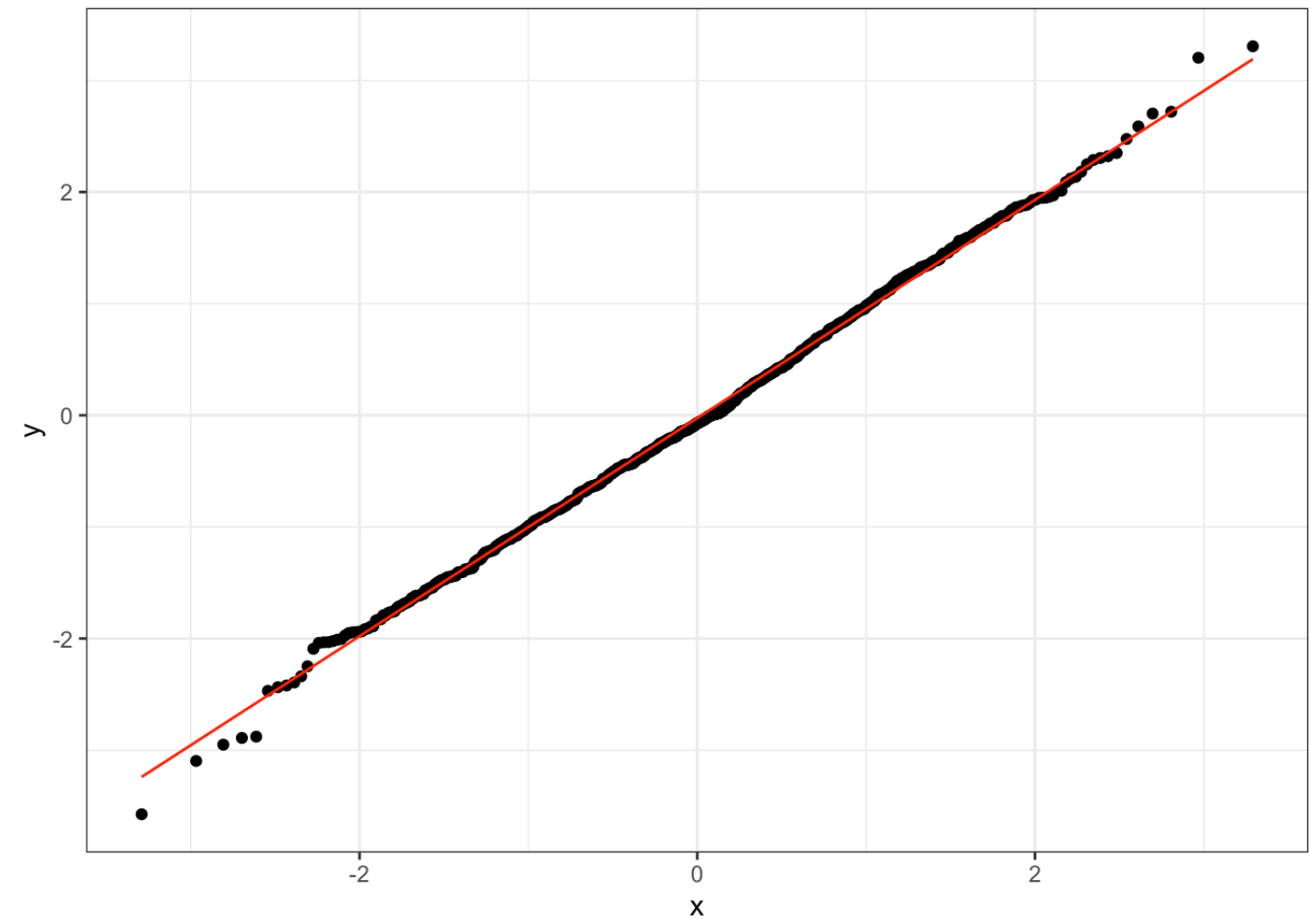
# histogram
data_n <- tibble(data_n)
data_n |>
  ggplot(aes(x = data_n)) +
  geom_histogram(
    binwidth = 0.1
  ) +
  theme_bw()

```



The QQ plot lies very closely to the 45-degree line. This means the sample quantile values closely match the theoretical quantile values of a standard normal distribution.

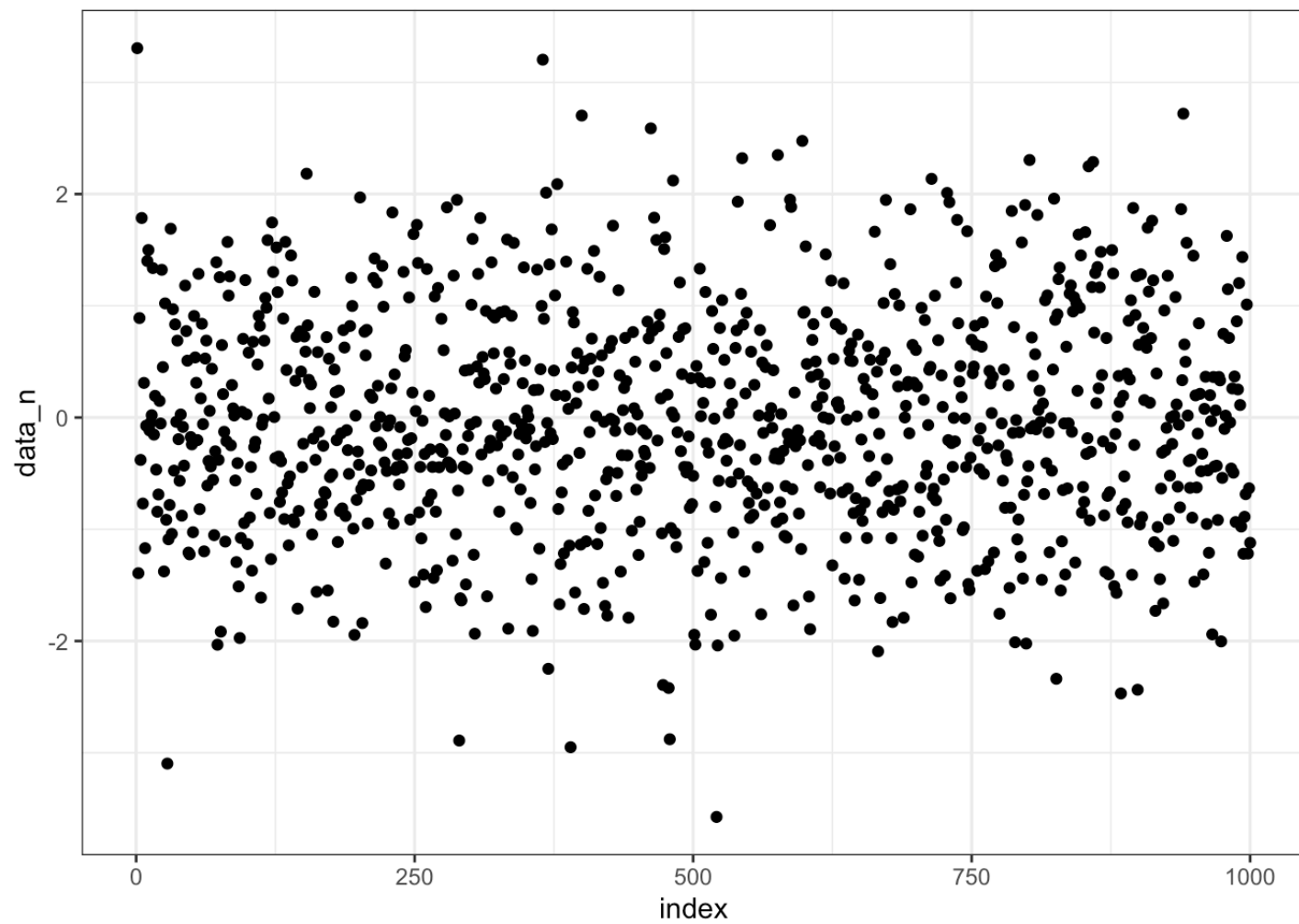
```
# QQ plot with standard normal distribution as the reference theoretical distribution
ggplot(data_n, aes(sample = data_n)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  theme_bw()
```



The time series plot shows that the data points are randomly distributed over the index without any trend or clustering. Hence, the draws appear to be independent.

```
# time series plot
data_n |>
  mutate(index = row_number()) |>
  ggplot(aes(x = index, y = data_n)) +
  geom_point() +
```

```
theme_bw()
```



Exercise 3

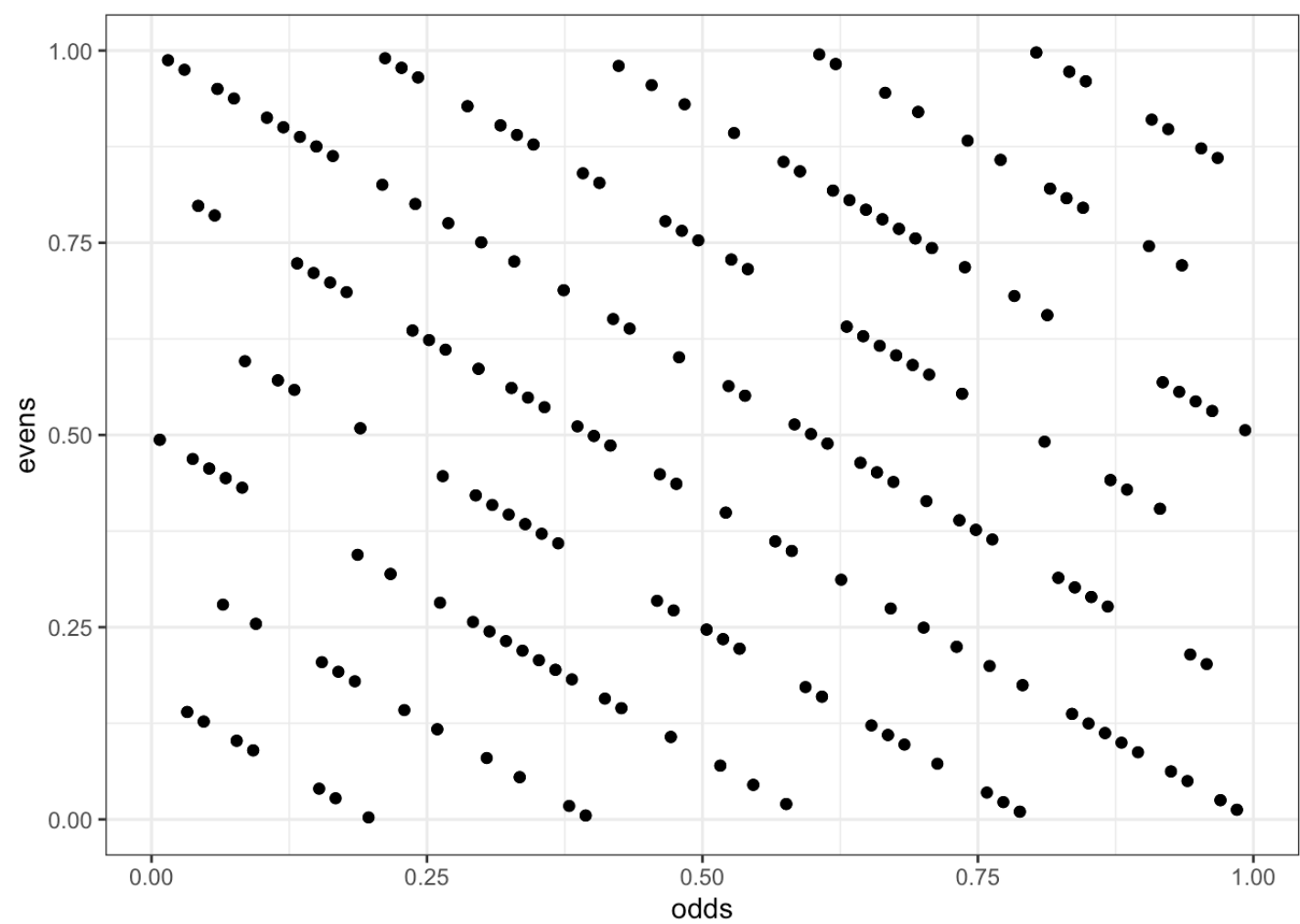
```
# write a new equation with the same function as runif_zx81 but sets a = 66 and m = 401
runif_3 <- function(seed, n, min = 0, max = 1){
  # Set the a and m parameters as specified.
  m <- 401
  a <- 66
  # Add warning messages in case the seed input is negative or larger than m.
  if (seed < 0) {
    "The seed is negative!"
  }
  if (seed > m) {
    "The seed is bigger than m!"
  }
  # Initialize an empty vector to save the draws, and save the start of the
  # sequence as the first draw. Hint: how do you access elements of vectors?
  seq <- c()
  seq[1] <- seed
  # Run a for loop to construct as many elements of the sequence as specified by
  # the number of draws n.
  for (i in 2:n){
    x <- (a * seq[i-1]) %% m
    seq[i] <- x
  }
  # Adjust the interval of your vector to run from min to max.
  seq <- (max - min)*(seq/m) + min
  # Return the vector of pseudorandom numbers.
  seq
}
```

```
# set seed = 42 and generate 1000 draws
data3 <- runif_3(42, 1000)
```

```
# extract odd and even indexed values and make a new tibble with these two variables
odds <- data3[seq(1, length(data3), 2)]
evens <- data3[seq(2, length(data3), 2)]
data3 <- tibble(odds = odds, evens = evens)
```

The scatter plot shows points lying on a number of parallel downward-sloping lines. This contradicts true randomness, where points should be uniformly scattered. This corresponds to theorem 1 in Marsaglia, 1968 that the Lehmer random number generator would give points lying on a set of parallel hyperplanes and the number of hyperplanes is also bounded. The plot below is a 2D demonstration of this defect where the hyperplanes are given by the parallel lines. There are also many systems of parallel hyperplanes which contain all of the points. Therefore, pseudo-random numbers form MCGs are not truly random in higher dimensions.

```
# make the 2D scatter plot
data3 |>
  ggplot(aes(x = odds, y = evens)) +
  geom_point() +
  theme_bw()
```



PS2_Wells

AUTHOR
1093122

Exercise 1

a.

```
# load the data and store it in a tibble
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

```
library(dplyr)
wells <- read_csv("https://ditraglia.com/data/wells.csv")
```

Rows: 3020 Columns: 5
— Column specification —
Delimiter: ","
dbl (5): switch, arsenic, dist, assoc, educ

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.

```
wells
```

```
# A tibble: 3,020 × 5
  switch arsenic dist assoc educ
  <dbl>   <dbl> <dbl> <dbl> <dbl>
1     1     2.36  16.8     0     0
2     1     0.71  47.3     0     0
3     0     2.07  21.0     0    10
4     1     1.15  21.5     0    12
5     1     1.1   40.9     1    14
6     1     3.9   69.5     1     9
7     1     2.97  80.7     1     4
8     1     3.24  55.1     0    10
9     1     3.28  52.6     1     0
10    1     2.52  75.1     1     0
# i 3,010 more rows
```

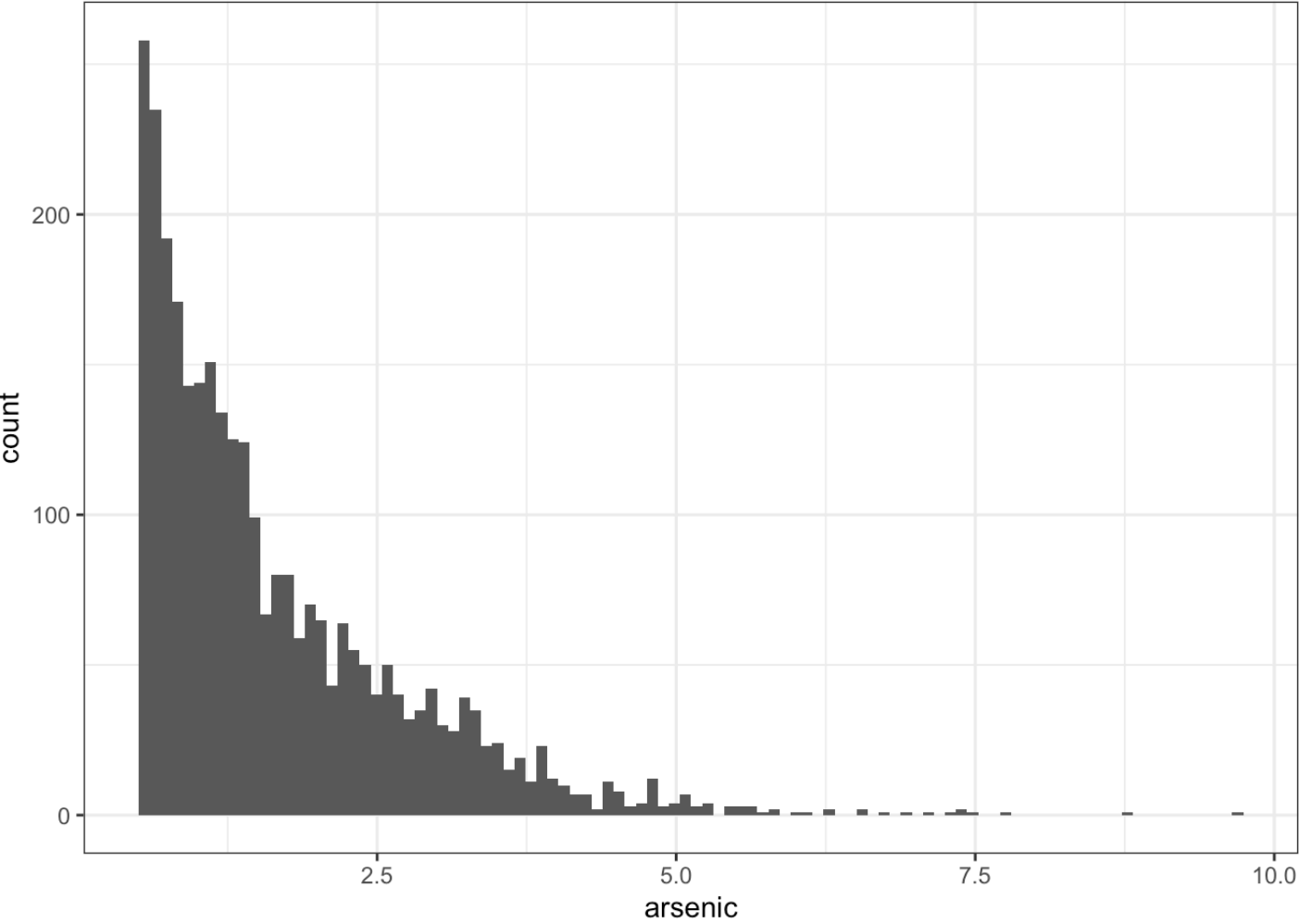
b.

```
# create the natural logarithm of arsenic
wells <- wells |>
  mutate(larsenic = log(arsenic))
```

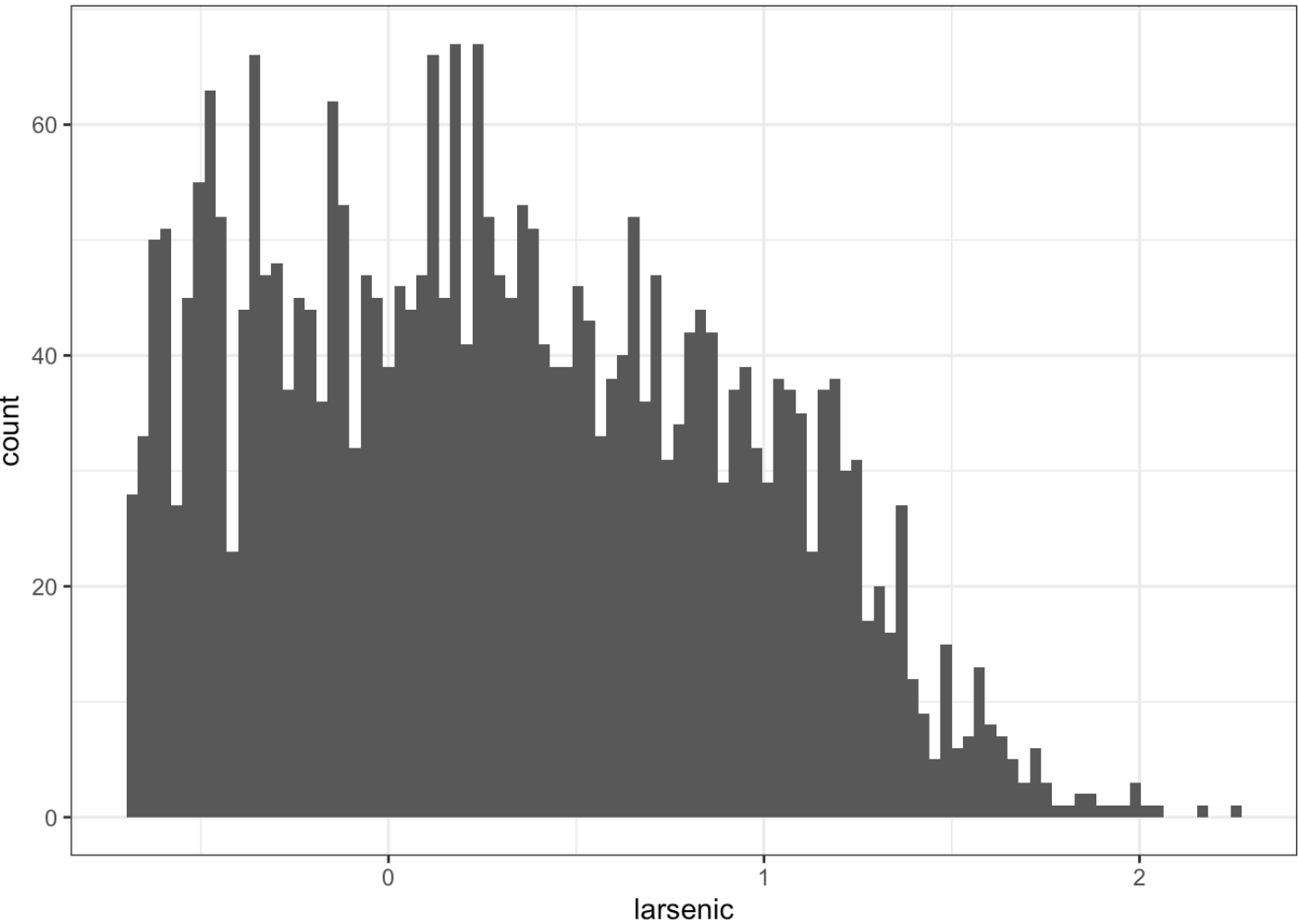
c.

The distribution of arsenic is skewed to the right, which means the right tail of the distribution is longer than the left tail. Most of the data is clustered around the lower end and a few larger values pull the tail to the right. The distribution for larsenic have shorter tails and data clusters near the middle. This is because the natural log transformation compresses extreme values due to its concave property and the rate of increase in larsenic would be smaller as arsenic increases. Therefore, the distribution for larsenic is more symmetric compared with arsenic.


```
# Use ggplot2 to make a histogram of arsenic and larsenic
wells |>
  ggplot(aes(x = arsenic)) +
  geom_histogram(bins = 100) +
  theme_bw()
```



```
wells |>
  ggplot(aes(x = larsenic)) +
  geom_histogram(bins = 100) +
  theme_bw()
```



C.

```
# Measure the distance in hundreds of meters
```

```
wells <- wells |>
  mutate(dist100 = dist /100)
```

d.

```
# zeduc: z-score of educ
wells <- wells |>
  mutate(zeduc = (educ - mean(educ))/sd(educ))
```

Exercise 2

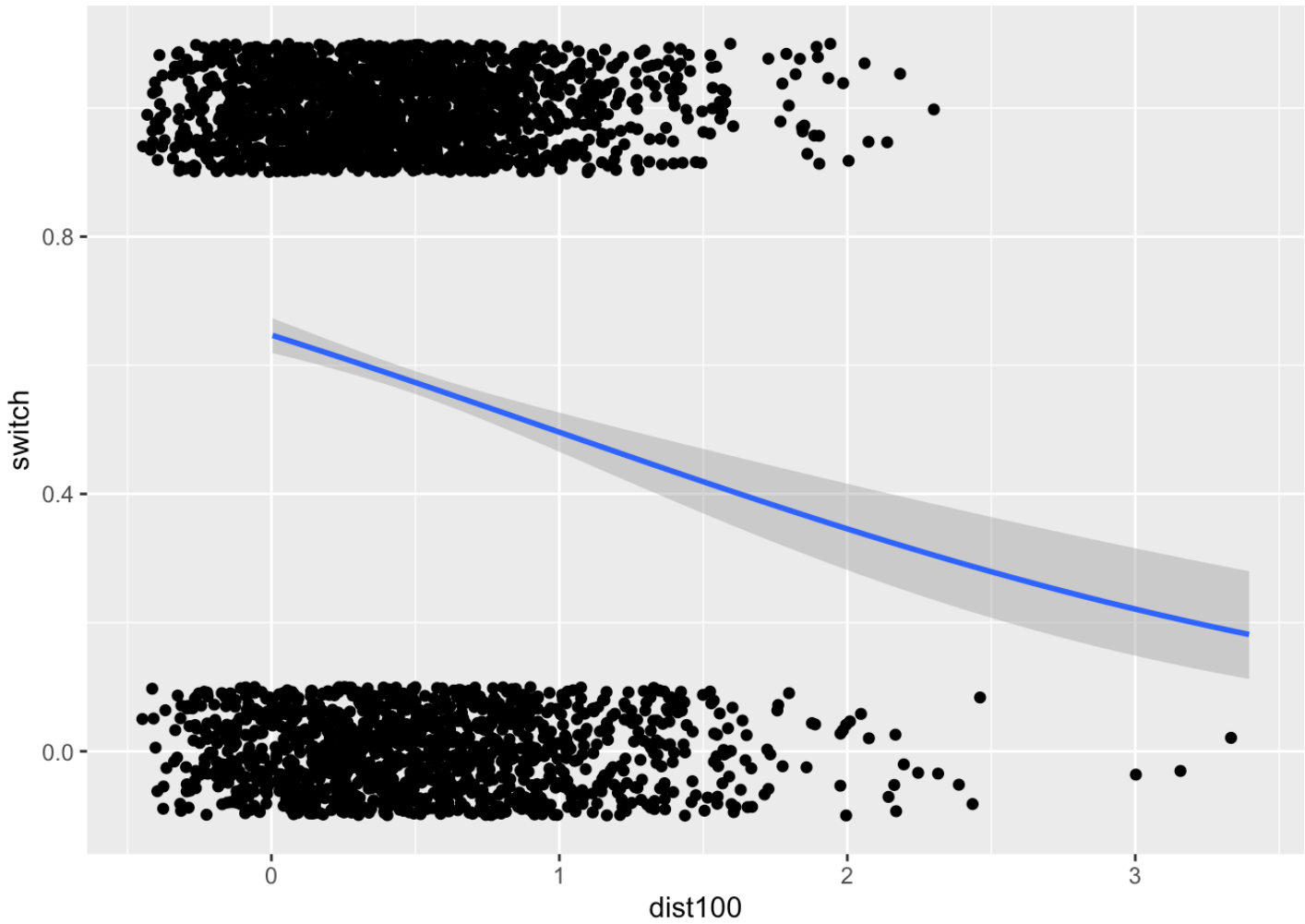
a.

```
# Run a logistic regression using dist100 to predict switch and store the result in an object called fit1
fit1 <- glm(switch ~ dist100, family = binomial(link = "logit"), wells)
```

b.

```
# Use ggplot2 to plot the logistic regression function from part (a) along with the data
ggplot(wells, aes(dist100, switch)) +
  # stat_smooth() plots the predicted probabilities of switching given dist100
  stat_smooth(method = "glm", method.args = list(family = 'binomial')) +
  geom_jitter(width = 0.5,
              height = 0.1)
```

`geom_smooth()` using formula = 'y ~ x'



c.

The test below suggests that dist100 is a statistically significant predictor of switch as p-value < 0.001. The sign of the coefficient is negative: as the distance to closest known safe well increases, the probability of switching decreases. This makes sense since it is more difficult to reach a safe well given longer distance.

```
summary(fit1)
```

Call:

```
glm(formula = switch ~ dist100, family = binomial(link = "logit"),
    data = wells)
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.60596     0.06031  10.047  < 2e-16 ***
dist100      -0.62188     0.09743  -6.383 1.74e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 4118.1  on 3019  degrees of freedom
Residual deviance: 4076.2  on 3018  degrees of freedom
AIC: 4080.2
```

Number of Fisher Scoring iterations: 4

d.

```
# Estimate of P(switch = 1 | dist100 = mean(dist100))
p_average <- predict(fit1, newdata = data.frame(dist100 = mean(wells$dist100)), type = 'response')
p_average
```

```
1
0.5757602
```

e.

For an average household in the dataset, a unit increase in hundreds meters from the closest safe well decreases the probability of switching by about 15%. Compared to the maximum marginal effect, the difference is not very large. This means the mean of dist100 is close to zero.

```
marginal_effect_average <- coef(fit1)["dist100"] * p_average * (1-p_average)
marginal_effect_average
```

```
dist100
-0.1519011
```

```
max_effect <- coef(fit1)["dist100"]/4
max_effect
```

```
dist100
-0.1554705
```

Exercise 3

a.

```
wells <- wells |>
  mutate(p1 = predict(fit1, type = "response"))
wells
```

A tibble: 3,020 × 9

	switch	arsenic	dist	assoc	educ	larsenic	dist100	zeduc	p1
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	2.36	16.8	0	0	0.859	0.168	-1.20	0.623
2	1	0.71	47.3	0	0	-0.342	0.473	-1.20	0.577
3	0	2.07	21.0	0	10	0.728	0.210	1.29	0.617
4	1	1.15	21.5	0	12	0.140	0.215	1.79	0.616
5	1	1.1	40.9	1	14	0.0953	0.409	2.28	0.587
6	1	3.9	69.5	1	9	1.36	0.695	1.04	0.543
7	1	2.97	80.7	1	4	1.09	0.807	-0.206	0.526
8	1	3.24	55.1	0	10	1.18	0.551	1.29	0.565
9	1	3.28	52.6	1	0	1.19	0.526	-1.20	0.569
10	1	2.52	75.1	1	0	0.924	0.751	-1.20	0.535

i 3,010 more rows

b.

```
wells <- wells |>
  mutate(pred1 = ifelse(p1 > 1/2, 1, 0))
wells
```

A tibble: 3,020 × 10

	switch	arsenic	dist	assoc	educ	larsenic	dist100	zeduc	p1	pred1
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	2.36	16.8	0	0	0.859	0.168	-1.20	0.623	1
2	1	0.71	47.3	0	0	-0.342	0.473	-1.20	0.577	1
3	0	2.07	21.0	0	10	0.728	0.210	1.29	0.617	1
4	1	1.15	21.5	0	12	0.140	0.215	1.79	0.616	1
5	1	1.1	40.9	1	14	0.0953	0.409	2.28	0.587	1
6	1	3.9	69.5	1	9	1.36	0.695	1.04	0.543	1
7	1	2.97	80.7	1	4	1.09	0.807	-0.206	0.526	1
8	1	3.24	55.1	0	10	1.18	0.551	1.29	0.565	1
9	1	3.28	52.6	1	0	1.19	0.526	-1.20	0.569	1
10	1	2.52	75.1	1	0	0.924	0.751	-1.20	0.535	1

i 3,010 more rows

c.

```
wells <- wells |>
  mutate(incorrect = ifelse(switch != pred1, 1, 0))

wells |>
  summarise(error_rate = mean(incorrect))
```

A tibble: 1 × 1

error_rate
<dbl>
1 0.405

d.

```
conf_matrix <- table(Actual = wells$switch, Predicted = wells$pred1)
conf_matrix
```

	Predicted	
Actual	0	1
0	194	1089
1	133	1604

e.

```
sensitivity <- (conf_matrix[2,2])/(conf_matrix[1, 2] + conf_matrix[2, 2])
specificity <- (conf_matrix[1,1])/(conf_matrix[1,1] + conf_matrix[2, 1])
sensitivity
```

[1] 0.5956183

```
specificity
```

[1] 0.5932722

f.

The most common value for switch is 1 as mean of switch > 0.5. The error rate would be $1 - \text{mean}(\text{switch}) = 0.42$. The error rate using prediction values is around 0.4 in (c). The probability of giving false positives or false negatives is also around 0.4 as given in (e). The difference in error rate is not very large.

```
print(mean(wells$switch))
```

[1] 0.5751656

```
print(1-mean(wells$switch))
```

[1] 0.4248344

Exercise 4

a-c.

```
fit2 <- glm(switch ~ larsenic, family = binomial(link = "logit"), wells)
fit3 <- glm(switch ~ zeduc, family = binomial(link = "logit"), wells)
fit4 <- glm(switch ~ dist100 + larsenic + zeduc, family = binomial(link = "logit"), wells)
```

d.

```
library(modelsummary)
fits <- list(fit1, fit2, fit3, fit4)
modelsummary(fits, gof_omit = 'Log.Lik|R2 Adj.|AIC|BIC|F', fmt = 2, title = "Logistic Regression Results")
```

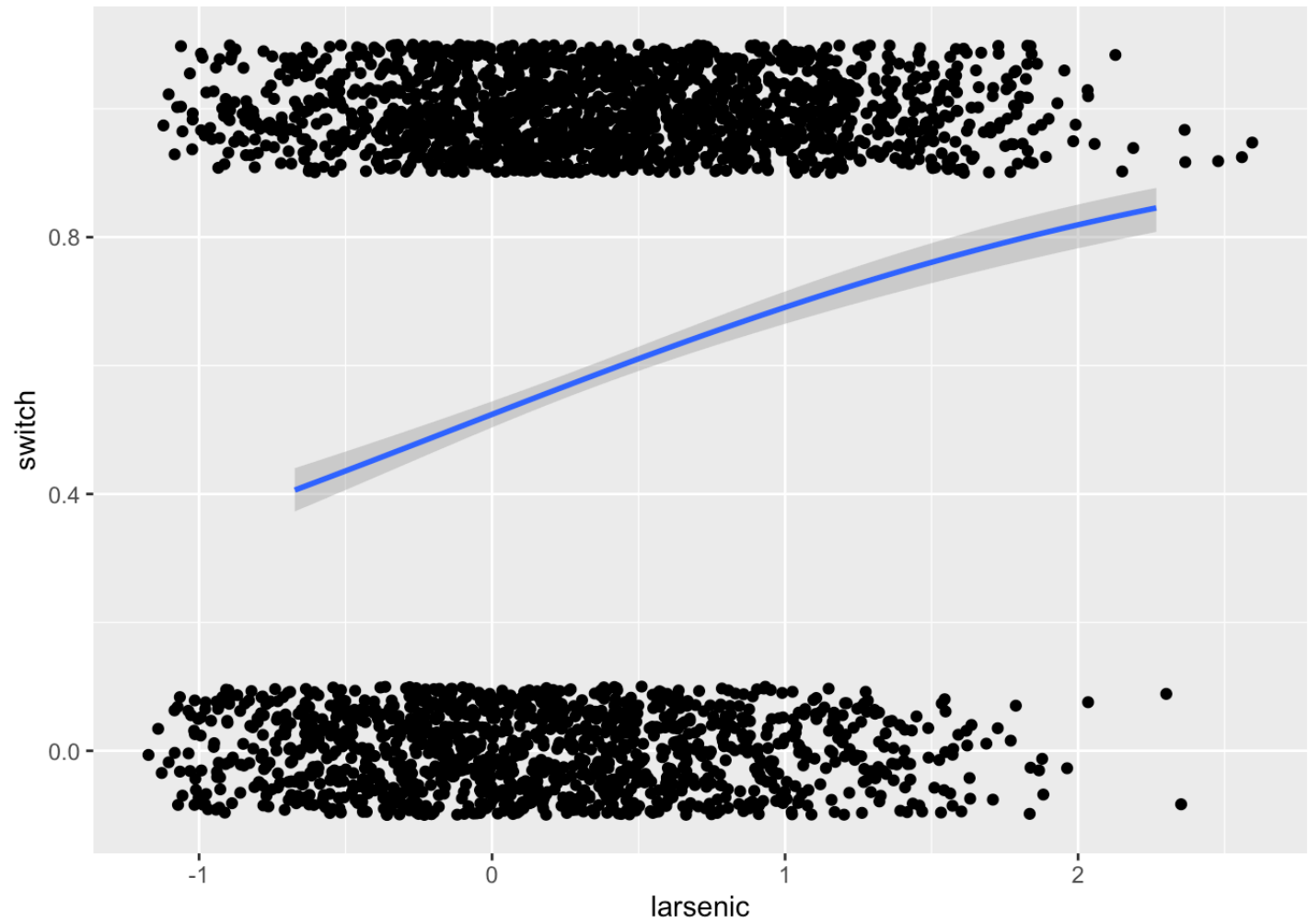
Logistic Regression Results table				
	(1)	(2)	(3)	(4)
(Intercept)	0.61	0.10	0.30	0.53
	(0.06)	(0.04)	(0.04)	(0.06)
dist100	-0.62			-0.98
	(0.10)			(0.11)
larsenic		0.71		0.89
		(0.06)		(0.07)
zeduc			0.16	0.17
			(0.04)	(0.04)
Num.Obs.	3020	3020	3020	3020
RMSE	0.49	0.48	0.49	0.47
Note: This table summarises the estimates from the four logistic regressions. Standard errors are given in the parentheses				

Exercise 5

a.

```
# Repeat 2.b for fit2
ggplot(wells, aes(larsenic, switch)) +
  # stat_smooth() plots the predicted probabilities of switching given dist100
  stat_smooth(method = "glm", method.args = list(family = 'binomial')) +
  geom_jitter(width = 0.5,
              height = 0.1)
```

`geom_smooth()` using formula = 'y ~ x'



larsenic is a statistically significant predictor of switch as shown below. The sign of the coefficient is positive: this means the probability of switching increases as the arsenic level increases. This makes sense as the water is more damaging to human body, which incentivises people to switch.

```
# repeat 2.c for fit2
summary(fit2)
```

Call:
glm(formula = switch ~ larsenic, family = binomial(link = "logit"),
data = wells)

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.09619	0.04137	2.325	0.0201 *
larsenic	0.70765	0.06404	11.050	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

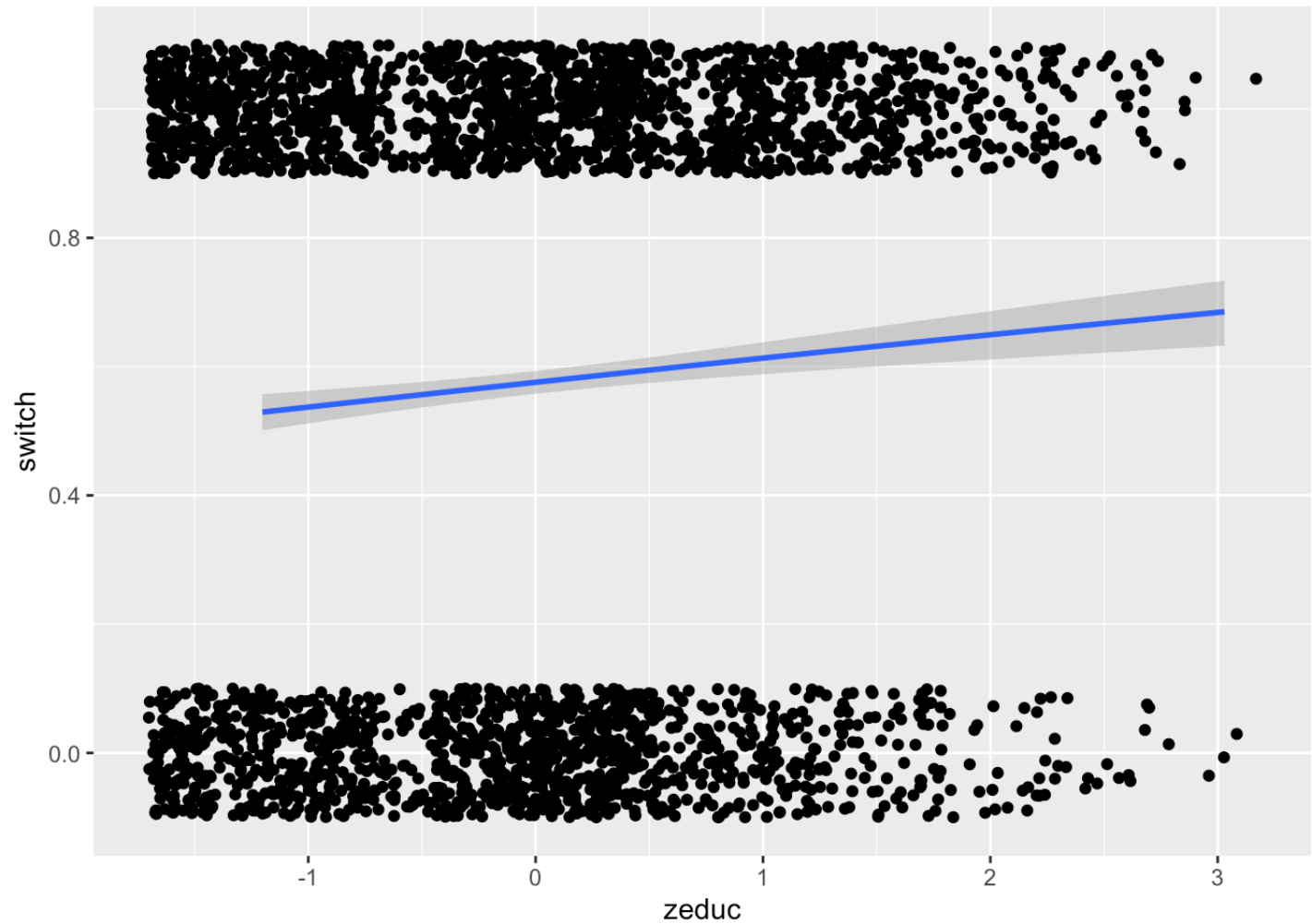
Null deviance: 4118.1 on 3019 degrees of freedom
Residual deviance: 3989.3 on 3018 degrees of freedom
AIC: 3993.3

Number of Fisher Scoring iterations: 4

b.

```
# Repeat 2.b for fit3
ggplot(wells, aes(zeduc, switch)) +
  # stat_smooth() plots the predicted probabilities of switching given dist100
  stat_smooth(method = "glm", method.args = list(family = 'binomial')) +
  geom_jitter(width = 0.5,
              height = 0.1)
```

`geom_smooth()` using formula = 'y ~ x'



zeduc is a statistically significant predictor of switch. The sign of the coefficient is positive. This means households with higher levels of education are more likely to switch. Intuitively, people with higher education are more likely to realise the damage of arsenic and hence are more likely to switch.

```
# repeat 2.c for fit3
summary(fit3)
```

Call:
glm(formula = switch ~ zeduc, family = binomial(link = "logit"),
data = wells)

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.30490	0.03693	8.256	< 2e-16 ***
zeduc	0.15600	0.03726	4.187	2.83e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 4118.1 on 3019 degrees of freedom
Residual deviance: 4100.4 on 3018 degrees of freedom
AIC: 4104.4

Number of Fisher Scoring iterations: 4

C.

```
p_mean <- predict(fit4, newdata = data.frame(dist100 = mean(wells$dist100), larsenic = mean(wells$larsenic)), type = "response")
p_mean
```

1
0.5820232

If the distance to the closest safe well increases by 100 meters, switching probability would decrease by 24%. An 1% increase in the level of arsenic increases the probability of switching by about 0.22%. One standard deviation increase in education level would increase the switching probability by 4.2%. The marginal effect for an average household is close to the maximum marginal effect for all predictors. This means the intercept and weighted sum of mean predictors balance out to zero.

```
marginal_effect_mean_dist100 <- coef(fit4)["dist100"] * p_mean * (1-p_mean)
marginal_effect_mean_larsenic <- coef(fit4)["larsenic"] * p_mean * (1-p_mean)
marginal_effect_mean_zeduc <- coef(fit4)["zeduc"] * p_mean * (1-p_mean)
max_effect_fit4 <- c(coef(fit4)["dist100"], coef(fit4)["larsenic"], coef(fit4)["zeduc"])*0.25
marginal_effect <- c(marginal_effect_mean_dist100,
marginal_effect_mean_larsenic,
marginal_effect_mean_zeduc)

marginal_effect
```

dist100	larsenic	zeduc
-0.23814696	0.21625074	0.04212322

```
max_effect_fit4
```

dist100	larsenic	zeduc
-0.24473302	0.22223125	0.04328816

Exercise 6

a.

```
wells <- wells |>
  mutate(p4 = predict(fit4, type = "response"),
         pred4 = ifelse(p4 > 1/2, 1, 0))

wells <- wells |>
  mutate(incorrect4 = ifelse(switch != pred4, 1, 0))

wells|>
  summarise(error_rate4 = mean(incorrect4))
```

```
# A tibble: 1 × 1
  error_rate4
    <dbl>
1      0.370
```

```
conf_matrix4 <- table(Actual = wells$switch, Predicted = wells$pred4)
conf_matrix4
```

	Predicted	
Actual	0	1
0	546	737
1	379	1358

```
sensitivity4 <- (conf_matrix4[2,2])/(conf_matrix4[1, 2] + conf_matrix4[2, 2])
specificity4 <- (conf_matrix4[1,1])/(conf_matrix4[1,1] + conf_matrix4[2, 1])
sensitivity4
```

[1] 0.64821

```
specificity4
```

[1] 0.5902703

The most common value for switch is 1 as mean of switch > 0.5. The error rate would be $1 - \text{mean}(\text{switch}) = 0.42$. The error rate using prediction values of fit4 is around 0.37. The probability of giving false positives is around 0.35 or false negatives is around 0.4. Fit4 gives a lower error rate compared with the null model and fit1.

```
# error rate by using the most common value
print(1-mean(wells$switch))
```

[1] 0.4248344


```
1-sensitivity4
```

```
[1] 0.35179
```

```
1-specificity4
```

```
[1] 0.4097297
```

b.

In terms of the overall error rate, fit4 is around 4% lower than fit1. For false positive rates, fit4 is around 5% lower than fit1. For false negatives, fit4 performs almost the same as fit1. Hence, fit4 performs better overall than fit1 and this is mainly because it has a lower rate for giving false positives.

```
# in-sample predictive performance of fit1 and fit4
# error rate
error_rate1 <- mean(wells$incorrect)
error_rate4 <- mean(wells$incorrect4)
error_rate14 <- c(error_rate1, error_rate4)
error_rate14
```

```
[1] 0.4046358 0.3695364
```

```
# false positive
false_positive <- c(1-sensitivity, 1-sensitivity4)
false_positive
```

```
[1] 0.4043817 0.3517900
```

```
# false negative
false_negative <- c(1-specificity, 1-specificity4)
false_negative
```

```
[1] 0.4067278 0.4097297
```

PS2_NSW

AUTHOR
1093122

Exercise 1

a.

```
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
```

```
library(haven)
library(dplyr)
library(broom)

experimental <- read_dta("https://users.nber.org/~rdehejia/data/nsw_dw.dta")
```

b.

```
experimental <- experimental |>
  rename(earnings74 = re74,
         earnings75 = re75,
         earnings78 = re78)
experimental
```

```
# A tibble: 445 × 11
  data_id      treat   age education black hispanic married nodegree earnings74
  <chr>      <dbl> <dbl>    <dbl> <dbl>   <dbl>   <dbl>   <dbl>      <dbl>
1 Dehejia-Wah...     1    37         11     1     0       1         1         0
2 Dehejia-Wah...     1    22          9     0     1       0         1         0
3 Dehejia-Wah...     1    30         12     1     0       0         0         0
4 Dehejia-Wah...     1    27         11     1     0       0         1         0
5 Dehejia-Wah...     1    33          8     1     0       0         1         0
6 Dehejia-Wah...     1    22          9     1     0       0         1         0
7 Dehejia-Wah...     1    23         12     1     0       0         0         0
8 Dehejia-Wah...     1    32         11     1     0       0         1         0
9 Dehejia-Wah...     1    22         16     1     0       0         0         0
10 Dehejia-Wah...    1    33         12     0     0       1         0         0
# i 435 more rows
# i 2 more variables: earnings75 <dbl>, earnings78 <dbl>
```

c.

```
experimental <- experimental |>
  mutate(race = case_when(black == 1 ~ "black",
                          hispanic == 1 ~ "hispanic",
                          TRUE ~ "white"))
  ) |>
  select(-black, -hispanic)
experimental
```

```
# A tibble: 445 × 10
  data_id      treat   age education married nodegree earnings74 earnings75
  <chr>      <dbl> <dbl>    <dbl>   <dbl>   <dbl>      <dbl>      <dbl>
```

```
      <chr>      <dbl> <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
1 Dehejia-Wahba S...      1      37          11          1          1          0          0
2 Dehejia-Wahba S...      1      22           9          0          1          0          0
3 Dehejia-Wahba S...      1      30          12          0          0          0          0
4 Dehejia-Wahba S...      1      27          11          0          1          0          0
5 Dehejia-Wahba S...      1      33           8          0          1          0          0
6 Dehejia-Wahba S...      1      22           9          0          1          0          0
7 Dehejia-Wahba S...      1      23          12          0          0          0          0
8 Dehejia-Wahba S...      1      32          11          0          1          0          0
9 Dehejia-Wahba S...      1      22          16          0          0          0          0
10 Dehejia-Wahba S...      1      33          12          1          0          0          0
# i 435 more rows
# i 2 more variables: earnings78 <dbl>, race <chr>
```

d.

```
experimental <- experimental |>
  mutate(treat = ifelse(treat == 1, "treated", "non_treated"),
         degree = ifelse(nodegree == 1, "no_high_school", "high_school"),
         marriage = ifelse(married, "married", "unmarried")
  ) |>
  select(-nodegree, -married)
experimental
```

```
# A tibble: 445 × 10
  data_id  treat  age education earnings74 earnings75 earnings78 race  degree
  <chr>    <chr> <dbl>      <dbl>      <dbl>      <dbl>      <dbl> <chr> <chr>
1 Dehejia-... trea...   37         11          0          0      9930. black no_hi...
2 Dehejia-... trea...   22          9          0          0      3596. hisp... no_hi...
3 Dehejia-... trea...   30         12          0          0     24909. black high_...
4 Dehejia-... trea...   27         11          0          0      7506. black no_hi...
5 Dehejia-... trea...   33          8          0          0       290. black no_hi...
6 Dehejia-... trea...   22          9          0          0      4056. black no_hi...
7 Dehejia-... trea...   23         12          0          0         0 black high_...
8 Dehejia-... trea...   32         11          0          0      8472. black no_hi...
9 Dehejia-... trea...   22         16          0          0      2164. black high_...
10 Dehejia-... trea...   33         12          0          0     12418. white high_...
# i 435 more rows
# i 1 more variable: marriage <chr>
```

e.

```
experimental <- experimental |>
  mutate(employment74 = ifelse(earnings74==0, "unemployed", "employed"),
         employment75 = ifelse(earnings75==0, "unemployed", "employed"))

experimental
```

```
# A tibble: 445 × 12
  data_id  treat  age education earnings74 earnings75 earnings78 race  degree
  <chr>    <chr> <dbl>      <dbl>      <dbl>      <dbl>      <dbl> <chr> <chr>
1 Dehejia-... trea...   37         11          0          0      9930. black no_hi...
2 Dehejia-... trea...   22          9          0          0      3596. hisp... no_hi...
3 Dehejia-... trea...   30         12          0          0     24909. black high_...
4 Dehejia-... trea...   27         11          0          0      7506. black no_hi...
5 Dehejia-... trea...   33          8          0          0       290. black no_hi...
6 Dehejia-... trea...   22          9          0          0      4056. black no_hi...
7 Dehejia-... trea...   23         12          0          0         0 black high_...
8 Dehejia-... trea...   32         11          0          0      8472. black no_hi...
9 Dehejia-... trea...   22         16          0          0      2164. black high_...
10 Dehejia-... trea...   33         12          0          0     12418. white high_...
# i 435 more rows
# i 3 more variables: marriage <chr>, employment74 <chr>, employment75 <chr>
```

f.

```
experimental <- experimental |>
  select(-data_id)
```

```
experimental
```






```
# A tibble: 445 × 11
  treat    age education earnings74 earnings75 earnings78 race  degree marriage
<chr>  <dbl>    <dbl>    <dbl>    <dbl>    <dbl>  <chr> <chr>   <chr>
1 treat...  37      11        0        0    9930. black no_hi... married
2 treat...  22       9        0        0    3596. hisp... no_hi... unmarri...
3 treat...  30      12        0        0   24909. black high_... unmarri...
4 treat...  27      11        0        0    7506. black no_hi... unmarri...
5 treat...  33       8        0        0     290. black no_hi... unmarri...
6 treat...  22       9        0        0   4056. black no_hi... unmarri...
7 treat...  23      12        0        0      0 black high_... unmarri...
8 treat...  32      11        0        0   8472. black no_hi... unmarri...
9 treat...  22      16        0        0   2164. black high_... unmarri...
10 treat... 33      12        0        0  12418. white high_... married
# i 435 more rows
# i 2 more variables: employment74 <chr>, employment75 <chr>
```

```
# define the cleanup function for data cleaning later
cleanup <- function(experimental){
  experimental <- experimental |>
  rename(earnings74 = re74,
         earnings75 = re75,
         earnings78 = re78)
  experimental <- experimental |>
  mutate(race = case_when(black == 1 ~ "black",
                          hispanic == 1 ~ "hispanic",
                          TRUE ~ "white")
         ) |>
  select(-black, -hispanic)
  experimental <- experimental |>
  mutate(treat = ifelse(treat == 1, "treated", "non_treated"),
         degree = ifelse(nodegree == 1, "no_high_school", "high_school"),
         marriage = ifelse(married, "married", "unmarried")
         ) |>
  select(-nodegree, -married)
  experimental <- experimental |>
  mutate(employment74 = ifelse(earnings74==0, "unemployed", "employed"),
         employment75 = ifelse(earnings75==0, "unemployed", "employed"))
  experimental <- experimental |>
  select(-data_id)
}
```

Exercise 2

a.

```
library(modelsummary)
datasummary_skim(experimental, type = "numeric")
```

	Unique	Missing Pct.	Mean	SD	Min	Median	Max	Histogram
age	34	0	25.4	7.1	17.0	24.0	55.0	
education	14	0	10.2	1.8	3.0	10.0	16.0	
earnings74	115	0	2102.3	5363.6	0.0	0.0	39570.7	
earnings75	155	0	1377.1	3151.0	0.0	0.0	25142.2	
earnings78	308	0	5300.8	6631.5	0.0	3701.8	60307.9	

```
datasummary_skim(experimental, type = "categorical")
```

N %

	treat	non_treated	260	58.4
		treated	185	41.6
race	black		371	83.4
	hispanic		39	8.8
	white		35	7.9
degree	high_school		97	21.8
	no_high_school		348	78.2
marriage	married		75	16.9
	unmarried		370	83.1
employment74	employed		119	26.7
	unemployed		326	73.3
employment75	employed		156	35.1
	unemployed		289	64.9

b.

The two groups are similar in terms of their average age, education and earnings in 1974. The average earnings for the treated group are slightly higher in 1975 and much higher in 1978 than the untreated group. The race composition in treated and untreated groups are similar while there is a slightly higher proportion of hispanic people and a lower proportion of white people in the non-treated group. A higher proportion of people in the treated group have high school degree compared with the non-treated group. The two groups also have similar marriage patterns and employment status in 1974. In 1975, a higher proportion of people from the treated group are employed. There seems to be no obvious selection bias based on observable characteristics.

```
datasummary_balance(experimental
                    ~ treat, data = experimental)
```

Warning: Please install the `estimatr` package or set `dinm=FALSE` to suppress this warning.

		non_treated (N=260)		treated (N=185)	
		Mean	Std. Dev.	Mean	Std. Dev.
age		25.1	7.1	25.8	7.2
education		10.1	1.6	10.3	2.0
earnings74		2107.0	5687.9	2095.6	4886.6
earnings75		1266.9	3103.0	1532.1	3219.3
earnings78		4554.8	5483.8	6349.1	7867.4
		N	Pct.	N	Pct.
race	black	215	82.7	156	84.3
	hispanic	28	10.8	11	5.9
	white	17	6.5	18	9.7
degree	high_school	43	16.5	54	29.2
	no_high_school	217	83.5	131	70.8
marriage	married	40	15.4	35	18.9
	unmarried	220	84.6	150	81.1

employment74	employed	65	25.0	54	29.2
	unemployed	195	75.0	131	70.8
employment75	employed	82	31.5	74	40.0
	unemployed	178	68.5	111	60.0

C.

The entire 95% confidence interval lies above 0. This indicates the NSW program significantly increased participants’ earnings.

```
# calculate the group means for the treated and non-treated group
group_means <- experimental |>
  group_by(treat) |>
  summarize(
    mean_earnings78 = mean(earnings78),
    se = sd(earnings78)/sqrt(n()),
    n = n()
  )

# Compute ATE
ate <- group_means$mean_earnings78[2] - group_means$mean_earnings78[1]

# Pooled standard error
se_ate <- sqrt(sum(group_means$se^2))

# Construct confidence interval
ci_lower <- ate - 1.96 * se_ate
ci_higher <- ate + 1.96 * se_ate
ci <- c(ci_lower = ci_lower, ci_higher = ci_higher)
ci
```

```
ci_lower ci_higher
479.1892  479.1892
```

Exercise 3

a.

```
cps_controls <- read_dta("https://users.nber.org/~rdehejia/data/cps_controls.dta")
cps_controls
```

```
# A tibble: 15,992 × 11
  data_id treat  age education black hispanic married nodegree  re74  re75
  <chr>   <dbl> <dbl>   <dbl> <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl>
1 CPS1      0   45      11      0      0      1      1 21517. 25244.
2 CPS1      0   21      14      0      0      0      0  3176.  5853.
3 CPS1      0   38      12      0      0      1      0 23039. 25131.
4 CPS1      0   48       6      0      0      1      1 24994. 25244.
5 CPS1      0   18       8      0      0      1      1  1669. 10728.
6 CPS1      0   22      11      0      0      1      1 16366. 18449.
7 CPS1      0   48      10      0      0      1      1 16805. 16355.
8 CPS1      0   18      11      0      0      0      1  1144.  3620.
9 CPS1      0   48       9      0      0      1      1 25862. 25244.
10 CPS1     0   45      12      0      0      1      0 25862.     0
# i 15,982 more rows
# i 1 more variable: re78 <dbl>
```

b.

```
cps_controls <- cleanup(cps_controls)
cps_controls
```

```
# A tibble: 15,992 × 11
  treat    age education earnings74 earnings75 earnings78 race  degree marriage
<chr>  <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <chr> <chr>  <chr>
1 non_t...  45      11    21517.    25244.    25565. white no_hi... married
2 non_t...  21      14     3176.     5853.    13496. white high_... unmarri...
3 non_t...  38      12    23039.    25131.    25565. white high_... married
4 non_t...  48       6    24994.    25244.    25565. white no_hi... married
5 non_t...  18       8     1669.    10728.     9861. white no_hi... married
6 non_t...  22      11    16366.    18449.    25565. white no_hi... married
7 non_t...  48      10    16805.    16355.    18059. white no_hi... married
8 non_t...  18      11     1144.     3620.    15739. white no_hi... unmarri...
9 non_t...  48       9    25862.    25244.    25565. white no_hi... married
10 non_t...  45      12    25862.         0     3925. white high_... married
# i 15,982 more rows
# i 2 more variables: employment74 <chr>, employment75 <chr>
```

C.

```
experimental_treated <- experimental |>
  filter(treat == "treated")
composite <- bind_rows(cps_controls, experimental_treated)
```

d.

```
balance_table <- datasummary_balance(
  ~ treat, # Compare groups
  data = composite,
  title = "Balance Table: NSW-treated vs CPS-controls",
  dinm = FALSE
)

# Print the table
balance_table
```

		non_treated (N=15992)		treated (N=185)	
		Mean	Std. Dev.	Mean	Std. Dev.
age		33.2	11.0	25.8	7.2
education		12.0	2.9	10.3	2.0
earnings74		14016.8	9569.8	2095.6	4886.6
earnings75		13650.8	9270.4	1532.1	3219.3
earnings78		14846.7	9647.4	6349.1	7867.4
		N	Pct.	N	Pct.
race	black	1176	7.4	156	84.3
	hispanic	1152	7.2	11	5.9
	white	13664	85.4	18	9.7
degree	high_school	11261	70.4	54	29.2
	no_high_school	4731	29.6	131	70.8
marriage	married	11382	71.2	35	18.9
	unmarried	4610	28.8	150	81.1
employment74	employed	14079	88.0	54	29.2
	unemployed	1913	12.0	131	70.8
employment75	employed	14244	89.1	74	40.0

unemployed	1748	10.9	111	60.0
------------	------	------	-----	------

Balance Table: NSW-treated vs CPS-controls

e.

The CPS controls appear to be very different from the treated group for all the observable characteristics listed above. The raw differences in earnings78 can not be interpreted as the average treatment effect as there is likely to be a selection bias and other characteristics such as race and education are likely to confound the treatment effect estimate.

Exercise 4

a.

The estimated treatment effect is \$1066 when regressing earnings on treatment status and covariates in the composite dataset with a p-value of 0.054. This suggest weak evidence of a positive effect. However, the average treatment effect obtained from the randomised experiment is larger (\$1794) and statistically significant at 5% significant level as shown in 2.c.

```
reg1 <- lm(earnings78~., composite)
summary(reg1)
```

Call:

lm(formula = earnings78 ~ ., data = composite)

Residuals:

Min	1Q	Median	3Q	Max
-25010	-3521	1283	3772	53738

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.122e+03	4.886e+02	10.483	< 2e-16 ***
treattreated	1.066e+03	5.536e+02	1.926	0.054092 .
age	-9.446e+01	5.995e+00	-15.755	< 2e-16 ***
education	1.745e+02	2.867e+01	6.086	1.18e-09 ***
earnings74	2.913e-01	1.274e-02	22.869	< 2e-16 ***
earnings75	4.414e-01	1.286e-02	34.323	< 2e-16 ***
racehispanic	5.794e+02	2.878e+02	2.013	0.044085 *
racewhite	8.098e+02	2.127e+02	3.808	0.000141 ***
degreeno_high_school	3.411e+02	1.777e+02	1.919	0.054993 .
marriageunmarried	-1.534e+02	1.427e+02	-1.075	0.282184
employment74unemployed	3.515e+02	2.315e+02	1.519	0.128845
employment75unemployed	-1.621e+03	2.396e+02	-6.766	1.37e-11 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6993 on 16165 degrees of freedom
Multiple R-squared: 0.4774, Adjusted R-squared: 0.4771
F-statistic: 1343 on 11 and 16165 DF, p-value: < 2.2e-16

b.

The assumption of conditional independence is needed. After controlling for observable covariates, treatment assignment must be independent of potential outcomes. This requires that all confounders are observed and included in the model.

c.

The ATE conditional on covariates is expressed as:

ATE = E[Y | D = 1, X] - E[Y | D = 0, X]

Under **unconfoundedness**, the assignment of the treatment can be seen as random. Hence, D is mean independent of (Y0, Y1), we can write ATE over the full population as:

$$ATE = \mathbb{E}[\mathbb{E}[Y_1 \mid X] - \mathbb{E}[Y_0 \mid X]]$$

Using the linear conditional expectations above:

$$\mathbb{E}[Y_1 \mid X] = \alpha_1 + X'\beta_1 \mathbb{E}[Y_0 \mid X] = \alpha_0 + X'\beta_0$$

So,

$$\mathbb{E}[Y_1 \mid X] - \mathbb{E}[Y_0 \mid X] = (\alpha_1 - \alpha_0) + X'(\beta_1 - \beta_0)$$

Now take the **expectation over the distribution of (X)**:

$$ATE = (\alpha_1 - \alpha_0) + \mathbb{E}[X'](\beta_1 - \beta_0)$$

d.

We are given that the conditional expectation of the potential outcomes is:

$$\mathbb{E}[Y_d \mid X] = \alpha_d + X'\beta_d \quad \text{for } d = 0, 1$$

Using the switching equation:

$$Y = D \cdot Y_1 + (1 - D) \cdot Y_0,$$

we can compute:

$$\mathbb{E}[Y \mid D, X] = D \cdot (\alpha_1 + X'\beta_1) + (1 - D)(\alpha_0 + X'\beta_0)$$

Expanding this, we get:

$$\mathbb{E}[Y \mid D, X] = \alpha_0 + X'\beta_0 + D \cdot [(\alpha_1 - \alpha_0) + X'(\beta_1 - \beta_0)]$$

Therefore, we should run a regression to estimate ATE as follows:

$$Y = \alpha + \tau D + X'\gamma + \varepsilon$$

Additional assumptions:

- (1) It is a linear model.
- (2) Unconfoundedness: $(Y_0, Y_1) \perp\!\!\!\perp D \mid X$

Exercise 5

a.

In the propensity score model, we should exclude earnings78 from the list of covariates because it is a post-treatment outcome variable. Including it would introduce post-treatment bias, violating the assumption that the propensity score is based solely on pre-treatment characteristics. Only covariates determined before treatment assignment should be included when estimating the propensity score.

```
logit_data <- composite |>
  mutate(
    treat = ifelse(treat == "treated", 1, 0),

    # Convert categorical vars to factor first
    race = as.factor(race),
    degree = as.factor(degree),
    marriage = as.factor(marriage),
    employment74 = as.factor(employment74),
    employment75 = as.factor(employment75)
  )

reg2 <- glm(treat ~ . - 1 - earnings78, data = logit_data, family = binomial(link = "logit"))
summary(reg2)
```

Call:

```
glm(formula = treat ~ . - 1 - earnings78, family = binomial(link = "logit"),
    data = logit_data)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
age	-1.790e-02	1.064e-02	-1.683	0.092421 .
education	1.949e-02	4.859e-02	0.401	0.688381
earnings74	6.253e-05	2.845e-05	2.197	0.027990 *
earnings75	-1.773e-04	3.600e-05	-4.923	8.51e-07 ***
raceblack	-3.049e+00	8.383e-01	-3.637	0.000275 ***
racehispanic	-5.500e+00	8.721e-01	-6.306	2.86e-10 ***
racewhite	-7.335e+00	8.716e-01	-8.416	< 2e-16 ***
degreeno_high_school	9.028e-01	2.751e-01	3.282	0.001031 **
marriageunmarried	9.943e-01	2.413e-01	4.120	3.79e-05 ***
employment74unemployed	1.572e+00	2.639e-01	5.956	2.58e-09 ***
employment75unemployed	2.347e-01	2.386e-01	0.984	0.325164

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 22426.08 on 16177 degrees of freedom
Residual deviance: 950.45 on 16166 degrees of freedom
AIC: 972.45

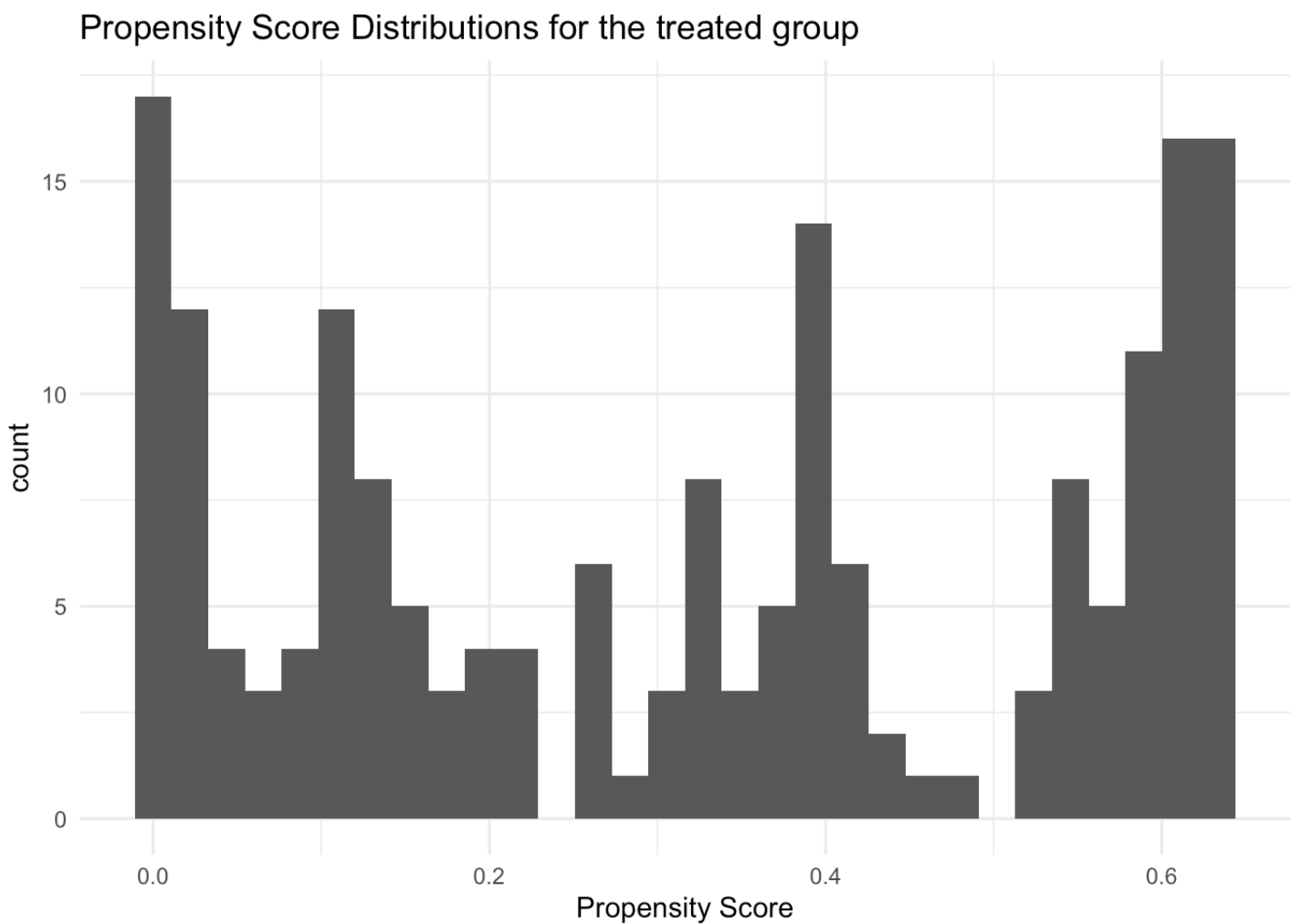
Number of Fisher Scoring iterations: 10

```
logit_data <- augment(reg2, type.predict = "response") |>
  rename(propensity_score = .fitted)
```

b.

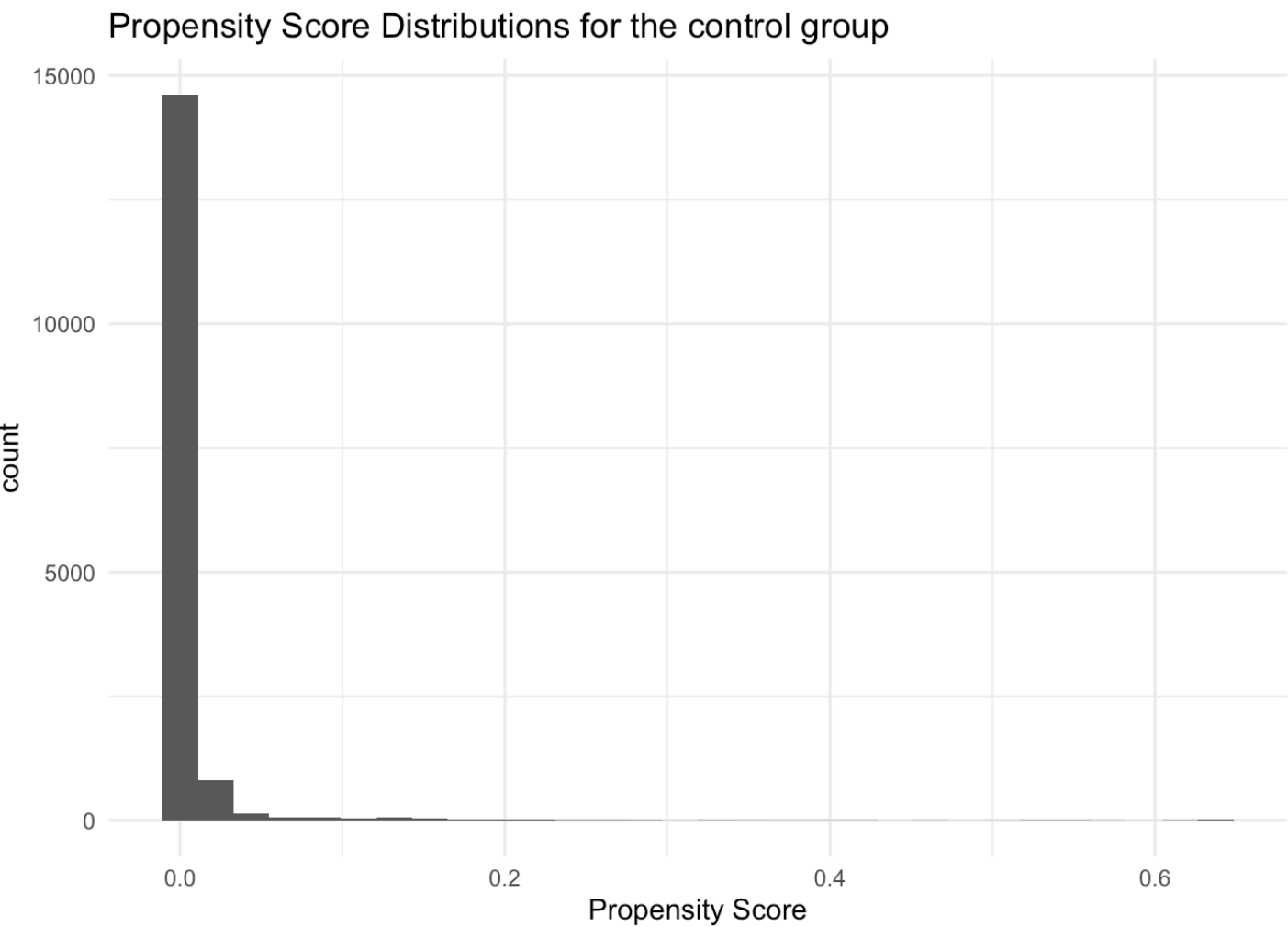
```
treated <- logit_data |>
  filter(treat == 1)

ggplot(treated, aes(x = propensity_score)) +
  geom_histogram(position = "identity", bins = 30) +
  labs(
    title = "Propensity Score Distributions for the treated group",
    x = "Propensity Score",
  ) +
  theme_minimal()
```



```
control <- logit_data |>
  filter(treat == 0)
```

```
ggplot(control, aes(x = propensity_score)) +
  geom_histogram(position = "identity", bins = 30) +
  labs(
    title = "Propensity Score Distributions for the control group",
    x = "Propensity Score",
  ) +
  theme_minimal()
```



```
# extra summary statistics
logit_data %>%
  group_by(treat) %>%
  summarise(
    mean_ps = mean(propensity_score),
    sd_ps = sd(propensity_score),
    min_ps = min(propensity_score),
    max_ps = max(propensity_score)
  )
```

A tibble: 2 × 5

	treat	mean_ps	sd_ps	min_ps	max_ps
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	0.00785	0.0434	0.00000470	0.638
2	1	0.322	0.226	0.000512	0.633

The result highlights the substantial difference in the distribution of propensity scores for the control and treated group. The histogram of the control group’s propensity scores shows an extreme concentration near 0, while the treated group’s scores are widely distributed between 0 and 0.65. This means people in the treated group are more likely to receive treatment than the control group, which suggests substantial differences in observed covariates between groups.

C.

```
# Compute IPW weights
logit_data <- logit_data |>
  mutate(
    weight_treat = treat / propensity_score,
    weight_control = (1 - treat) / (1 - propensity_score)
  )

# Compute psw
psw_estimate <- mean(
```

```
logit_data$weight_treat * logit_data$earnings78 -  
logit_data$weight_control * logit_data$earnings78  
)  
  
psw_estimate
```

```
[1] -10548.41
```

d.

```
# Step 1: Trim observations with extreme propensity scores  
logit_trimmed <- logit_data %>%  
  filter(propensity_score >= 0.1, propensity_score <= 0.9)  
  
# Step 2: Compute psw  
logit_trimmed <- logit_trimmed %>%  
  mutate(  
    weight_treat = treat / propensity_score,  
    weight_control = (1 - treat) / (1 - propensity_score)  
  )  
  
# Step 3: Compute trimmed psw estimator  
psw_estimator_trimmed <- mean(  
  logit_trimmed$weight_treat * logit_trimmed$earnings78 -  
  logit_trimmed$weight_control * logit_trimmed$earnings78  
)  
  
psw_estimator_trimmed
```

```
[1] 2047.641
```

e.

The difference between parts (c) and (d) arises from the presence of extreme propensity scores in the untrimmed sample. In part (c), these lead to very large inverse weights, especially among control units with scores near 0. This results in a highly volatile and unreliable ATE estimate. In contrast, part (d) trims these problematic observations and ensures that only observations with sufficient overlap are used for causal inference, where treated and control units are more comparable. This produces a much more stable and credible estimate of the treatment effect.