Homework 4 Final

Lekh Shetty

Link to the Github repository

❗ Due: Sun, Apr 2, 2023 @ 11:59pm

Please read the instructions carefully before submitting your assignment.

1. This assignment requires you to only upload a `PDF` file on Canvas
2. Don't collapse any code cells before submitting.
3. Remember to make sure all your code output is rendered properly before uploading your submission.

Please add your name to the author information in the frontmatter before submitting your assignment

We will be using the following libraries:

```
packages <- c(
  "dplyr",
  "readr",
  "tidyr",
  "tidyverse",
  "purrr",
  "stringr",
  "corrplot",
  "car",
  "caret",
  "torch",
  "nnet",
  "broom"
)
```

```r
#renv::install(packages)
sapply(packages, require, character.only=T)
```

```
Loading required package: dplyr


Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Loading required package: readr

Loading required package: tidyr

Loading required package: tidyverse

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v purrr     1.0.1
v ggplot2   3.4.1     v stringr   1.5.0
v lubridate 1.9.2     v tibble    3.2.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becon
Loading required package: corrplot

corrplot 0.92 loaded

Loading required package: car

Loading required package: carData
```

```
Attaching package: 'car'


The following object is masked from 'package:purrr':

    some


The following object is masked from 'package:dplyr':

    recode


Loading required package: caret

Loading required package: lattice


Attaching package: 'caret'


The following object is masked from 'package:purrr':

    lift


Loading required package: torch

Loading required package: nnet

Loading required package: broom

    dplyr     readr     tidyr tidyverse     purrr   stringr  corrplot       car
     TRUE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
    caret     torch      nnet     broom
     TRUE      TRUE      TRUE      TRUE
```

Question 1

30 points

Automatic differentiation using `torch`

1.1 (5 points)

Consider $g(x, y)$ given by

$$g(x, y) = (x - 3)^2 + (y - 4)^2.$$

Using elementary calculus derive the expressions for

$$\frac{d}{dx} g(x, y), \quad \text{and} \quad \frac{d}{dy} g(x, y).$$

Using your answer from above, what is the answer to

$$\frac{d}{dx} g(x, y)\Bigg|_{(x=3, y=4)} \quad \text{and} \quad \frac{d}{dy} g(x, y)\Bigg|_{(x=3, y=4)} ?$$

Define $g(x, y)$ as a function in R, compute the gradient of $g(x, y)$ with respect to $x = 3$ and $y = 4$. Does the answer match what you expected?

```
x_tensor <- torch_tensor(3, requires_grad = TRUE)
y_tensor <- torch_tensor(4, requires_grad = TRUE)

result <- torch_sum((x_tensor - 3)^2 + (y_tensor - 4)^2)

result$backward()

x_tensor$grad
```

```
torch_tensor
 0
[ CPUFloatType{1} ]
```

```
y_tensor$grad
```

```
torch_tensor
 0
[ CPUFloatType{1} ]
```

As seen from the above calculations the values of d/dx and d/dy equal to 0, as expected if we were to calculate the gradients ourselves

---

1.2 (10 points)

1.3 (5 points)

Define $f(z)$ as a function in R, and using the `torch` library compute $f'(-3.5)$.

```r
#library(torch)

f <- function(z) {
  return (z^4 - 6*z^2 - 3*z + 4)
}
```

```r
z <- torch_tensor(-3.5, requires_grad=TRUE)
output <- f(z)
output$backward()
z$grad
```

```
torch_tensor
-132.5000
[ CPUFloatType{1} ]
```

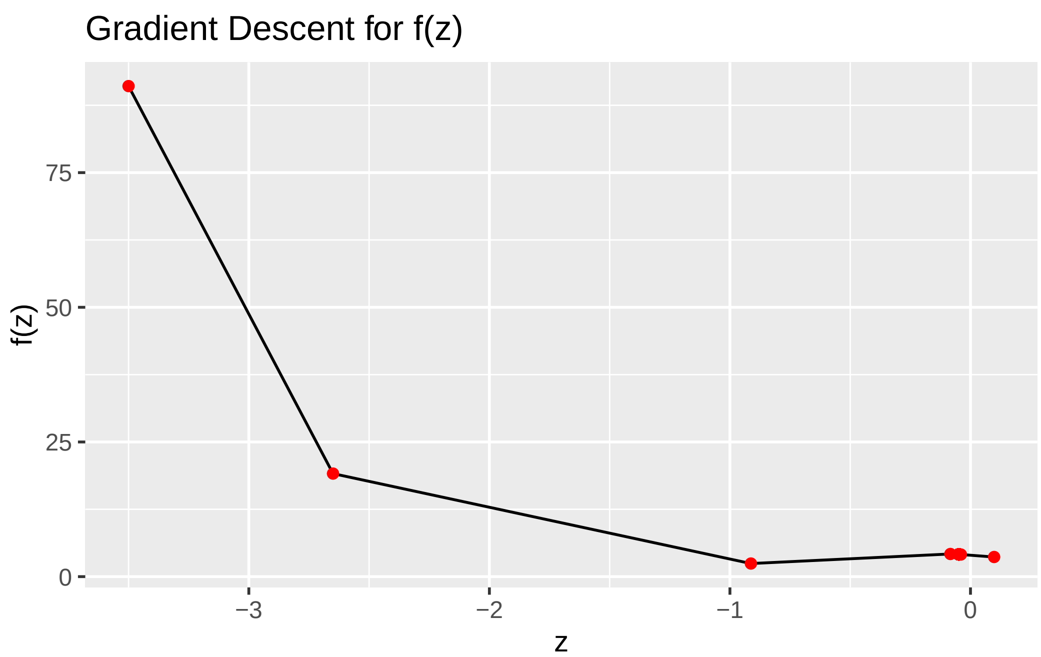
1.4 (5 points)

For the same function f, initialize $z[1] = -3.5$, and perform $n = 100$ iterations of **gradient descent**, i.e.,

> $z[{k+1}] = z[k] - \eta f'(z[k]) \ \ \ \ $ for $k = 1, 2, ..., 100$

Plot the curve f and add taking $\eta = 0.02$, add the points $\{z_0, z_1, z_2, ... z_{100}\}$ obtained using gradient descent to the plot. What do you observe?

```r
f <- function(z) {
  return (z^4 - 6*z^2 - 3*z + 4)
}

z <- torch_tensor(-3.5, requires_grad=TRUE)
eta <- 0.02
z_list <- list(z$detach())
for (i in 1:100) {
  output <- f(z)
  output$backward()
  z$detach_()
  z = eta * z$grad
  z$requires_grad_()
  z_list[[i+1]] <- z$detach()
}

z_vals <- unlist(lapply(z_list, function(x) as.numeric(x)))
f_vals <- unlist(lapply(z_list, function(x) f(x)$item()))
df <- data.frame(z=z_vals, f=f_vals)

ggplot(data=df, aes(x=z, y=f)) +
  geom_line() +
  geom_point(data=df, aes(x=z, y=f), color="red") +
  ggtitle("Gradient Descent for f(z)") +
  xlab("z") +
  ylab("f(z)")
```



1.5 (5 points)

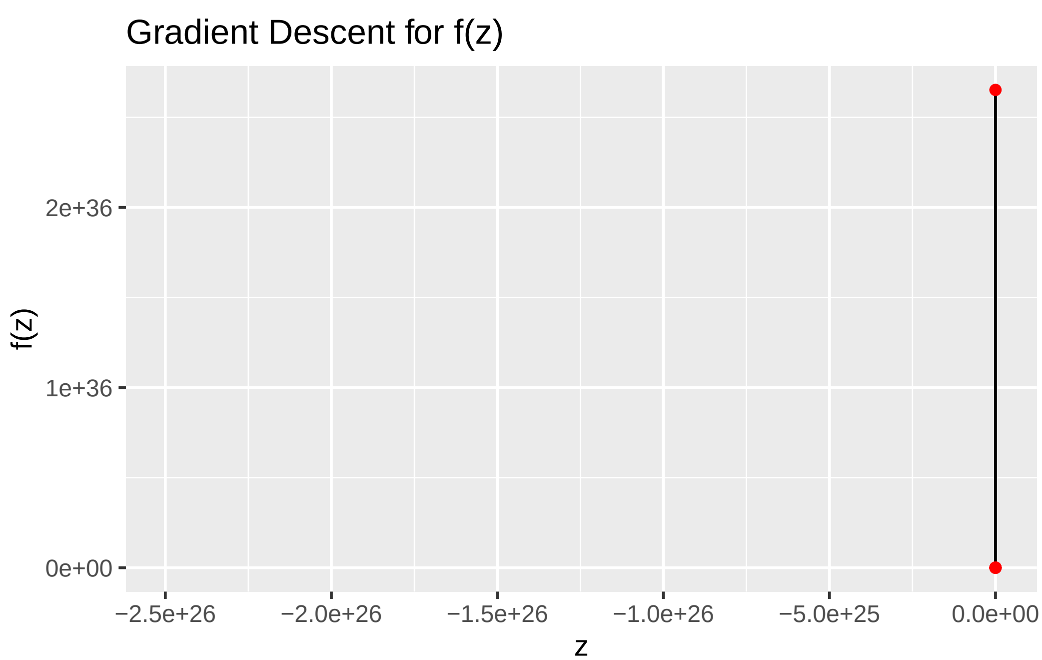
Redo the same analysis as **Question 1.4**, but this time using $\eta = 0.03$. What do you observe? What can you conclude from this analysis

```
f <- function(z) {
  return (z^4 - 6*z^2 - 3*z + 4)
}

z <- torch_tensor(-3.5, requires_grad=TRUE)
eta <- 0.03
z_list <- list(z$detach())
for (i in 1:100) {
  output <- f(z)
  output$backward()
  z$detach_()
  z = eta * z$grad
  z$requires_grad_()
  z_list[[i+1]] <- z$detach()
}
```

```r
z_vals <- unlist(lapply(z_list, function(x) as.numeric(x)))
f_vals <- unlist(lapply(z_list, function(x) f(x)$item()))
df <- data.frame(z=z_vals, f=f_vals)

ggplot(data=df, aes(x=z, y=f)) +
  geom_line() +
  geom_point(data=df, aes(x=z, y=f), color="red") +
  ggtitle("Gradient Descent for f(z)") +
  xlab("z") +
  ylab("f(z)")
```

```
Warning: Removed 95 rows containing missing values (`geom_line()`).
```

```
Warning: Removed 95 rows containing missing values (`geom_point()`).
```



—

Question 2

50 points

Logistic regression and interpretation of effect sizes

For this question we will use the **Titanic** dataset from the Stanford data archive. This dataset contains information about passengers aboard the Titanic and whether or not they survived.

2.1 (5 points)

Read the data from the following URL as a tibble in R. Preprocess the data such that the variables are of the right data type, e.g., binary variables are encoded as factors, and convert all column names to lower case for consistency. Let's also rename the response variable `Survival` to `y` for convenience.

```
url <- "https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv"

df <- read_csv(url, col_types = cols(
  Survived = col_factor(),
  Pclass = col_factor(),
  Name = col_character(),
  Sex = col_factor(),
  Age = col_double(),
  Siblings = col_double(),
  Parents = col_double(),
  Fare = col_double()
))
```

```
Warning: The following named parsers don't match the column names: Siblings,
Parents
```

```
colnames(df)[1]  <- "y"
colnames(df)[6]  <- "Siblings"
colnames(df)[7]  <- "Parents"

df$Sex<-ifelse(df$Sex=="male",1,0)
```

```r
names(df) <- tolower(names(df))

df
```

```
# A tibble: 887 x 8
   y     pclass name                          sex   age siblings parents  fare
   <fct> <fct>  <chr>                       <dbl> <dbl>    <dbl>   <dbl> <dbl>
 1 0     3      Mr. Owen Harris Braund          1    22        1       0  7.25
 2 1     1      Mrs. John Bradley (Florence ~   0    38        1       0 71.3
 3 1     3      Miss. Laina Heikkinen           0    26        0       0  7.92
 4 1     1      Mrs. Jacques Heath (Lily May~   0    35        1       0 53.1
 5 0     3      Mr. William Henry Allen         1    35        0       0  8.05
 6 0     3      Mr. James Moran                 1    27        0       0  8.46
 7 0     1      Mr. Timothy J McCarthy          1    54        0       0 51.9
 8 0     3      Master. Gosta Leonard Palsson   1     2        3       1 21.1
 9 1     3      Mrs. Oscar W (Elisabeth Vilh~   0    27        0       2 11.1
10 1     2      Mrs. Nicholas (Adele Achem) ~   0    14        1       0 30.1
# i 877 more rows
```

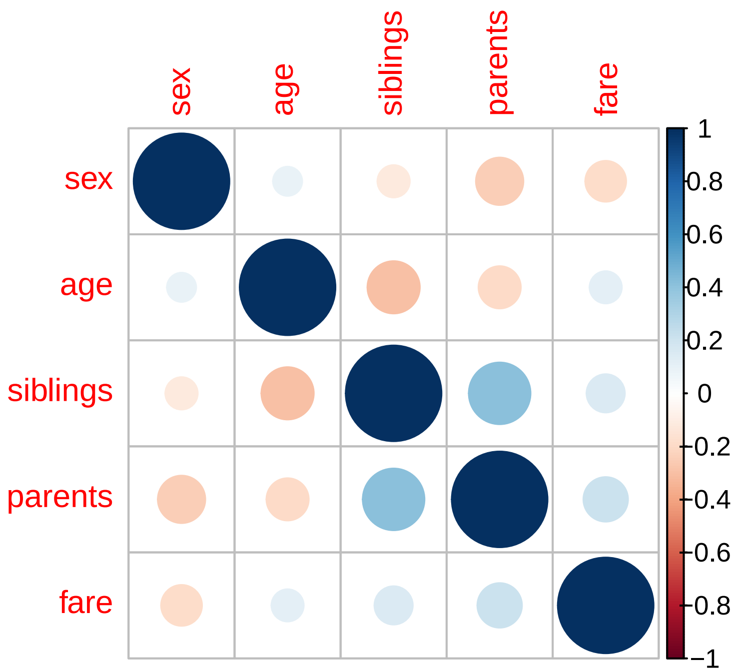
```r
#head(df)
summary(df)
```

```
 y       pclass      name                sex              age
 0:545   3:487   Length:887         Min.   :0.000   Min.   : 0.42
 1:342   1:216   Class :character   1st Qu.:0.000   1st Qu.:20.25
         2:184   Mode  :character   Median :1.000   Median :28.00
                                    Mean   :0.646   Mean   :29.47
                                    3rd Qu.:1.000   3rd Qu.:38.00
                                    Max.   :1.000   Max.   :80.00
    siblings          parents            fare
 Min.   :0.0000   Min.   :0.0000   Min.   :  0.000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:  7.925
 Median :0.0000   Median :0.0000   Median : 14.454
 Mean   :0.5254   Mean   :0.3833   Mean   : 32.305
 3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 31.137
 Max.   :8.0000   Max.   :6.0000   Max.   :512.329
```

2.2 (5 points)

Visualize the correlation matrix of all numeric columns in `df` using `corrplot()`

```
library(corrplot)

df %>%
  select_if(is.numeric) %>%
  cor() %>%
  corrplot()
```



2.3 (10 points)

Fit a logistic regression model to predict the probability of surviving the titanic as a function of:

- `pclass`
- `sex`
- `age`
- `fare`
- `# siblings`
- `# parents`

```r
full_model <- glm(y ~ pclass + sex + age + fare + siblings + parents, data = df, family =
summary(full_model)
```

```
Call:
glm(formula = y ~ pclass + sex + age + fare + siblings + parents,
    family = binomial(), data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7773  -0.5991  -0.3984   0.6131   2.4412

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.759755   0.283563   6.206 5.44e-10 ***
pclass1      2.350022   0.304666   7.713 1.22e-14 ***
pclass2      1.188532   0.229527   5.178 2.24e-07 ***
sex         -2.756710   0.200642 -13.739  < 2e-16 ***
age         -0.043410   0.007790  -5.573 2.51e-08 ***
fare         0.002823   0.002468   1.144  0.25277
siblings    -0.401572   0.110795  -3.624  0.00029 ***
parents     -0.106884   0.118767  -0.900  0.36815
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.77  on 886  degrees of freedom
Residual deviance:  780.93  on 879  degrees of freedom
AIC: 796.93

Number of Fisher Scoring iterations: 5
```

2.4 (30 points)

Provide an interpretation for the slope and intercept terms estimated in `full_model` in terms of the log-odds of survival in the titanic and in terms of the odds-ratio (if the covariate is also categorical).

> Recall the definition of logistic regression from the lecture notes, and also recall how we interpreted the slope in the linear regression model (particularly when the covariate was categorical).

```r
print("The intercept term: if a passenger is a male in third class with no siblings, no pa
```

```
[1] "The intercept term: if a passenger is a male in third class with no siblings, no parents
```

—

Question 3

> 70 points
>
> Variable selection and logistic regression in `torch`

3.1 (15 points)

Complete the following function `overview` which takes in two categorical vectors (`predicted` and `expected`) and outputs:

- The prediction accuracy
- The prediction error
- The false positive rate, and
- The false negative rate

```r
overview <- function(predicted, expected){
    accuracy <- sum(predicted == expected) / length(expected)
    error <- 1 - accuracy
    total_false_positives <- sum(predicted == 1 & expected == 0)
    total_true_positives <- sum(predicted == 1 & expected == 1)
    total_false_negatives <- sum(predicted == 0 & expected == 1)
    total_true_negatives <- sum(predicted == 0 & expected == 0)
    false_positive_rate <- total_false_positives / (total_false_positives + total_true_neg
    false_negative_rate <- total_false_negatives / (total_false_negatives + total_true_pos
    return(
        data.frame(
```

```
                accuracy = accuracy,
                error=error,
                false_positive_rate = false_positive_rate,
                false_negative_rate = false_negative_rate
            )
        )
}
```

You can check if your function is doing what it's supposed to do by evaluating

```
overview(df$y, df$y)
```

```
  accuracy error false_positive_rate false_negative_rate
1        1     0                   0                   0
```

and making sure that the accuracy is 100% while the errors are 0%.

3.2 (5 points)

```
full_model_prob <- predict(full_model, type="response")
full_model_pred <- ifelse(full_model_prob >= 0.5, 1, 0)

full_model_overview <- overview(full_model_prob, df$y)
full_model_overview
```

```
  accuracy error false_positive_rate false_negative_rate
1        0     1                 NaN                 NaN
```

3.3 (5 points)

Using backward-stepwise logistic regression, find a parsimonious alternative to `full_model`, and print its `overview`

```
step_model <- step(full_model, direction = "backward",scope=formula(full_model)) # Insert
```

```
Start:  AIC=796.93
y ~ pclass + sex + age + fare + siblings + parents

           Df Deviance     AIC
- parents   1   781.75  795.75
- fare      1   782.37  796.37
<none>          780.93  796.93
- siblings  1   796.79  810.79
- age       1   815.20  829.20
- pclass    2   847.84  859.84
- sex       1  1020.26 1034.26

Step:  AIC=795.75
y ~ pclass + sex + age + fare + siblings

           Df Deviance     AIC
- fare      1   782.82  794.82
<none>          781.75  795.75
- siblings  1   801.56  813.56
- age       1   815.88  827.88
- pclass    2   852.19  862.19
- sex       1  1024.08 1036.08

Step:  AIC=794.82
y ~ pclass + sex + age + siblings

           Df Deviance     AIC
<none>          782.82  794.82
- siblings  1   801.59  811.59
- age       1   818.25  828.25
- pclass    2   900.80  908.80
- sex       1  1031.69 1041.69
```

```
summary(step_model)
```

```
Call:
glm(formula = y ~ pclass + sex + age + siblings, family = binomial(),
    data = df)

Deviance Residuals:
```

```
    Min       1Q   Median       3Q      Max
-2.7637  -0.5883  -0.3930   0.6136   2.4543

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.752932   0.274653   6.382 1.74e-10 ***
pclass1      2.541237   0.258324   9.837  < 2e-16 ***
pclass2      1.219533   0.228650   5.334 9.63e-08 ***
sex         -2.738024   0.195796 -13.984  < 2e-16 ***
age         -0.043918   0.007757  -5.662 1.50e-08 ***
siblings    -0.409624   0.105495  -3.883 0.000103 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.77  on 886  degrees of freedom
Residual deviance:  782.82  on 881  degrees of freedom
AIC: 794.82

Number of Fisher Scoring iterations: 5
```

```r
step_predictions <-  predict(step_model, type = "response")
step_predictions <- ifelse(step_predictions >= 0.5, 1, 0)

overview(step_predictions, df$y)
```

```
   accuracy     error false_positive_rate false_negative_rate
1 0.8049605 0.1950395            0.133945           0.2923977
```

3.4 (15 points)

Using the `caret` package, setup a 5-fold cross-validation training method using the `caret::trainConrol()` function

```r
controls <- trainControl(method="cv", number=5) #insert your code here
```

Now, using `control`, perform 5-fold cross validation using `caret::train()` to select the optimal λ parameter for LASSO with logistic regression.

Take the search grid for λ to be in $\{2^{-20}, 2^{-19.5}, 2^{-19}, \dots, 2^{-0.5}, 2^{0}\}$.

```
# Insert your code in the ... region
library(glmnet)
```

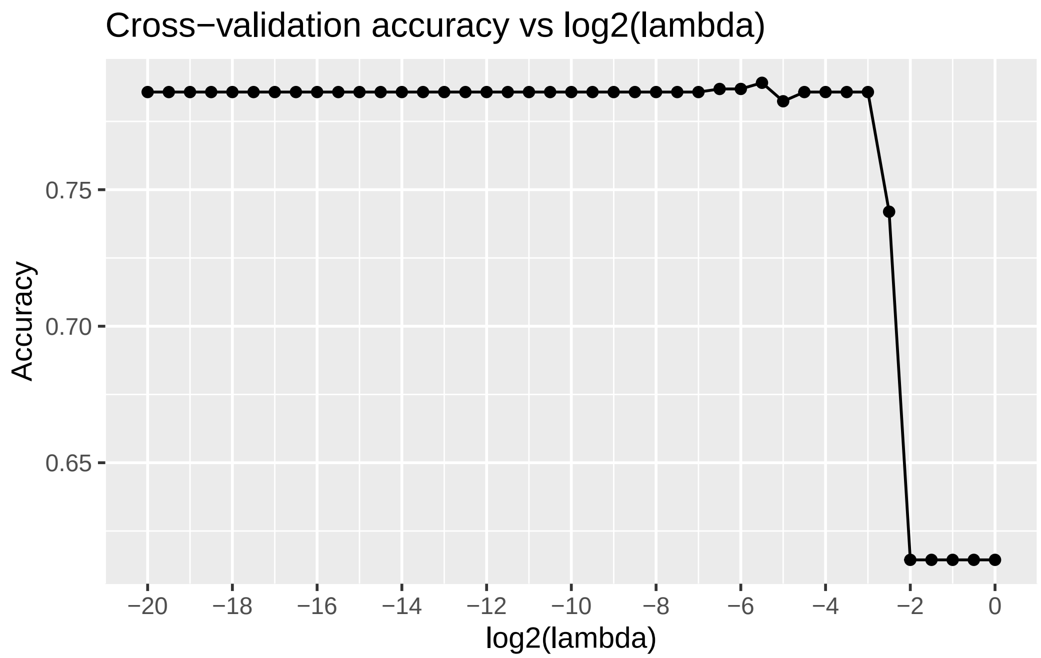
```
Loading required package: Matrix


Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack

Loaded glmnet 4.1-7
```

```
X <- model.matrix(y ~ ., data = df)
y <- df$y

lasso_fit <- train(
  x = X,
  y = y,
  method = "glmnet",
  trControl = controls,
  tuneGrid = expand.grid(
    alpha = 1,
    lambda = 2^seq(-20, 0, by = 0.5)
    ),
  family = "binomial"
)
```

Using the information stored in `lasso_fit$results`, plot the results for cross-validation accuracy vs. $log_2(\lambda)$. Choose the optimal λ^*, and report your results for this value of λ^*.

```
library(ggplot2)

ggplot(data = lasso_fit$results, aes(x = log2(lambda), y = Accuracy)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = seq(-20, 0, by = 2)) +
```

```
labs(title = "Cross-validation accuracy vs log2(lambda)", x = "log2(lambda)", y = "Accur
```



3.6 (5 points)

Create a summary table of the `overview()` summary statistics for each of the 4 models we have looked at in this assignment, and comment on their relative strengths and drawbacks.

—

::: {.callout-note collapse="true"} ## Session Information

Print your `R` session information using the following command

```
sessionInfo()
```

```
R version 4.2.3 (2023-03-15 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:
[1] stats     graphics  grDevices datasets  utils     methods   base

other attached packages:
 [1] glmnet_4.1-7     Matrix_1.5-3     broom_1.0.4      nnet_7.3-18
 [5] torch_0.9.1      caret_6.0-94     lattice_0.20-45 car_3.1-2
 [9] carData_3.0-5    corrplot_0.92    lubridate_1.9.2 forcats_1.0.0
[13] stringr_1.5.0    purrr_1.0.1      tibble_3.2.1     ggplot2_3.4.1
[17] tidyverse_2.0.0 tidyr_1.3.0      readr_2.1.4      dplyr_1.1.1

loaded via a namespace (and not attached):
 [1] nlme_3.1-162         bit64_4.0.5          tools_4.2.3
 [4] backports_1.4.1      utf8_1.2.3           R6_2.5.1
 [7] rpart_4.1.19         colorspace_2.1-0     withr_2.5.0
[10] tidyselect_1.2.0     processx_3.8.0       curl_5.0.0
[13] bit_4.0.5            compiler_4.2.3       cli_3.6.1
[16] labeling_0.4.2       scales_1.2.1         proxy_0.4-27
[19] callr_3.7.3          digest_0.6.31        rmarkdown_2.21
[22] coro_1.0.3           pkgconfig_2.0.3      htmltools_0.5.5
[25] parallelly_1.35.0    fastmap_1.1.1        rlang_1.1.0
[28] rstudioapi_0.14      shape_1.4.6          generics_0.1.3
[31] farver_2.1.1         jsonlite_1.8.4       vroom_1.6.1
```

```
[34] ModelMetrics_1.2.2.2 magrittr_2.0.3       Rcpp_1.0.10
[37] munsell_0.5.0        fansi_1.0.4          abind_1.4-5
[40] lifecycle_1.0.3      stringi_1.7.12       pROC_1.18.0
[43] yaml_2.3.7           MASS_7.3-58.2        plyr_1.8.8
[46] recipes_1.0.5        grid_4.2.3           parallel_4.2.3
[49] listenv_0.9.0        crayon_1.5.2         splines_4.2.3
[52] hms_1.1.3            knitr_1.42           ps_1.7.3
[55] pillar_1.9.0         future.apply_1.10.0  reshape2_1.4.4
[58] codetools_0.2-19     stats4_4.2.3         glue_1.6.2
[61] evaluate_0.20        data.table_1.14.8    renv_0.17.2
[64] vctrs_0.6.1          tzdb_0.3.0           foreach_1.5.2
[67] gtable_0.3.3         future_1.32.0        xfun_0.38
[70] gower_1.0.1          prodlim_2019.11.13   e1071_1.7-13
[73] class_7.3-21         survival_3.5-3       timeDate_4022.108
[76] iterators_1.0.14     hardhat_1.3.0        lava_1.7.2.1
[79] timechange_0.2.0     globals_0.16.2       ipred_0.9-14
```