Week 10 Quiz Material

When copy and pasting from a code block, or from your local `R` session, be sure to include all available digits for any numeric answer. It would be best to copy and paste values that were returned using printing methods that do not round results. (Notably the direct output from calling `summary()` .) Also, do not modify the default digits option in the code blocks or your local `R` session.

Practice

Exercise 1

```
# preamble
```

```
# starter
```

Consider a categorical response Y which takes possible values 0 and 1 as well as two numerical predictors X_1 and X_2. Recall that

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

Consider the model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

together with parameters

- $\beta_0 = 2$
- $\beta_1 = -1$
- $\beta_2 = -1$

Calculate $P[Y = 1 \mid X_1 = 1, X_2 = 0]$.

```
# solution
eta = 2 + -1 * 1 + 0 * -1
p = 1 / (1 + exp(-eta))
p
```

```
## [1] 0.7310586
```

- Hint: As stated, the response is the log-odds. You need to un-transform to obtain the desired conditional probability.

Exercise 2

```
# preamble
make_sim_data = function(n = 100) {
  x1 = rnorm(n = n)
  x2 = rnorm(n = n, sd = 2)
  x3 = rnorm(n = n, sd = 3)
  x4 = rnorm(n = n)
  x5 = rnorm(n = n)
  x6 = rnorm(n = n)
  x7 = rnorm(n = n)
  eta = -1 + 0.75 * x2 + 2.5 * x6
  p = 1 / (1 + exp(-eta))
  y = rbinom(n = n, 1, prob = p)
  data.frame(y, x1, x2, x3, x4, x5, x6, x7)
}

set.seed(1)
quiz_data = make_sim_data()
```

```
# starter
quiz_data
```

Recall that

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

Use the data available in the above code chunk stored in `quiz_data` to fit the model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7.$$

Report the value of the estimate for β_2.

```
# solution
fit = glm(y ~ ., data = quiz_data, family = "binomial")
unname(coef(fit)["x2"])
```

```
## [1] 0.8839991
```

- Hint: By default, `glm()` is fitting ordinary linear regression, assuming a numeric response.
- Hint: Be sure to set `family = "binomial"` in order to fit logistic regression.

Exercise 3

```
# preamble
make_sim_data = function(n = 100) {
  x1 = rnorm(n = n)
  x2 = rnorm(n = n, sd = 2)
  x3 = rnorm(n = n, sd = 3)
  x4 = rnorm(n = n)
  x5 = rnorm(n = n)
  x6 = rnorm(n = n)
  x7 = rnorm(n = n)
  eta = -1 + 0.75 * x2 + 2.5 * x6
  p = 1 / (1 + exp(-eta))
  y = rbinom(n = n, 1, prob = p)
  data.frame(y, x1, x2, x3, x4, x5, x6, x7)
}

set.seed(1)
quiz_data = make_sim_data()
```

```
# starter
quiz_data
```

Use the data available in the above code chunk stored in `quiz_data` to fit the model

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7.$$

Use a Wald test to test $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$. Report the p-value of this test.

```
# solution
fit = glm(y ~ ., data = quiz_data, family = "binomial")
coef(summary(fit))["x3", "Pr(>|z|)"]
```

```
## [1] 0.2395128
```

- Hint: Performing this test in `R` is extremely similar to a t-test for ordinary linear regression.

Exercise 4

```
# preamble
make_sim_data = function(n = 100) {
  x1 = rnorm(n = n)
  x2 = rnorm(n = n, sd = 2)
  x3 = rnorm(n = n, sd = 3)
  x4 = rnorm(n = n)
  x5 = rnorm(n = n)
  x6 = rnorm(n = n)
  x7 = rnorm(n = n)
  eta = -1 + 0.75 * x2 + 2.5 * x6
  p = 1 / (1 + exp(-eta))
  y = rbinom(n = n, 1, prob = p)
  data.frame(y, x1, x2, x3, x4, x5, x6, x7)
}

set.seed(1)
quiz_data = make_sim_data()
```

```
# starter
quiz_data
```

Use the data available in the above code chunk stored in `quiz_data` to fit the model

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7.$$

Using this as an initial model, use BIC and a backwards stepwise procedure to select a reduced model. Use likelihood ratio test to compare the initial model and the selected model. Report the p-value of this test.

```
# solution
fit = glm(y ~ ., data = quiz_data, family = "binomial")
fit_selected = step(fit, k = log(nrow(quiz_data)), trace = 0)
anova(fit_selected, fit, test = "LRT")[2, "Pr(>Chi)"]
```

```
## [1] 0.7695132
```

- Hint: You will need to use the `step()` function and specify the argument `k` .
- Hint: You will need to use the `anova()` function with the argument `test = "LRT"` .

Exercise 5

```
# preamble
make_sim_data = function(n = 100) {
  x1 = rnorm(n = n)
  x2 = rnorm(n = n, sd = 2)
  x3 = rnorm(n = n, sd = 3)
  x4 = rnorm(n = n)
  x5 = rnorm(n = n)
  x6 = rnorm(n = n)
  x7 = rnorm(n = n)
  eta = -1 + 0.75 * x2 + 2.5 * x6
  p = 1 / (1 + exp(-eta))
  y = rbinom(n = n, 1, prob = p)
  data.frame(y, x1, x2, x3, x4, x5, x6, x7)
}

set.seed(1)
quiz_data = make_sim_data()
```

```
# starter
quiz_data
# fit the model here
set.seed(1)
# calculate the metric here
```

Use the data available in the above code chunk stored in `quiz_data` to fit the model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7.$$

Calculate the 5-fold cross-validation misclassification rate when using this model as a classifier that seeks to minimize the misclassification rate. Since the data will be split randomly, use the seed provided after fitting the model. Also, use the relevant function from the `boot` package to ensure your calculation uses the same splits for grading purposes. (Even with the same seed, the splits could be done differently.)

```
# solution
fit = glm(y ~ ., data = quiz_data, family = "binomial")
set.seed(1)
boot::cv.glm(quiz_data, fit_selected, K = 5)$delta[1]
```

```
## [1] 0.1267324
```

- Hint: Use the `cv.glm()` function from the `boot` package.
- Hint: Extract the element `delta[1]` from the call to `cv.glm()` .

Graded

Exercise 1

```
# preamble
```

```
# starter
```

Consider a categorical response Y which takes possible values 0 and 1 as well as three numerical predictors X_1, X_2, and X_3. Recall that

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

Consider the model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

together with parameters

- $\beta_0 = -3$
- $\beta_1 = 1$
- $\beta_2 = 2$
- $\beta_2 = 3$

Calculate $P[Y = 0 \mid X_1 = -1, X_2 = 0.5, X_2 = 0.25]$

```
# solution
eta = -3 + (1 * -1) + (2 * 0.5) + (3 * 0.25)
p = 1 / (1 + exp(-eta))
1 - p
```

```
## [1] 0.9046505
```

Exercise 2

```
# preamble
```

```
# starter
```

For Exercises 2 - 7, use the built-in `R` dataset `mtcars`. We will use this dataset to attempt to predict whether or not a car has a manual transmission.

Recall that

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

Fit the model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

where

- Y is `am`
- x_1 is `mpg`
- x_2 is `hp`
- x_3 is `qsec`

Report the value of the estimate for β_3.

```
# solution
fit_full = glm(am ~ mpg + hp + qsec, data = mtcars, family = "binomial")
unname(coef(fit_full)["qsec"])
```

```
## [1] -4.040952
```

Exercise 3

```
# preamble
```

```
# starter
```

Using the model fit in Exercise 2, estimate the change in log-odds that a car has a manual transmission for an increase in fuel efficiency of one mile per gallon.

```
# solution
fit_full = glm(am ~ mpg + hp + qsec, data = mtcars, family = "binomial")
unname(coef(fit_full)["mpg"])
```

```
## [1] 2.29643
```

Exercise 4

```
# preamble
```

```
# starter
```

Using the model fit in Exercise 2, estimate the log-odds that a car has a manual transmission for a car with a fuel efficiency of 19 miles per gallon, 150 horsepower, and a quarter mile time of 19 seconds.

```
# solution
fit_full = glm(am ~ mpg + hp + qsec, data = mtcars, family = "binomial")
a_car = data.frame(mpg = 19, hp = 150, qsec = 19)
predict(fit_full, a_car, type = "link")
```

```
##         1
## -8.338686
```

Exercise 5

```
# preamble
```

```
# starter
```

Using the model fit in Exercise 2, estimate the probability that a car with a fuel efficiency of 22 miles per gallon, 123 horsepower, and a quarter mile time of 18 seconds has a manual transmission.

```
# solution
fit_full = glm(am ~ mpg + hp + qsec, data = mtcars, family = "binomial")
a_car = data.frame(mpg = 22, hp = 123, qsec = 18)
predict(fit_full, a_car, type = "response")
```

```
##        1
## 0.916414
```

Exercise 6

```
# preamble
```

```
# starter
```

Use a likeliood ratio test to test

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0$$

for the model fit in Exercise 2. Report the test statistic of this test.

```
# solution
fit_null = glm(am ~ 1, data = mtcars, family = "binomial")
fit_full = glm(am ~ mpg + hp + qsec, data = mtcars, family = "binomial")
anova(fit_null, fit_full, test = "LRT")[2, "Deviance"]
```

```
## [1] 35.74953
```

Exercise 7

```
# preamble
```

```
# starter
```

Recall that

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

Fit the model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

where

- Y is `am`
- x_1 is `mpg`
- x_2 is `hp`
- x_3 is `qsec`

Use a Wald test to test $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$. Report the p-value of this test.

```
# solution
fit_full = glm(am ~ mpg + hp + qsec, data = mtcars, family = "binomial")
coef(summary(fit_full))["hp", "Pr(>|z|)"]
```

```
## [1] 0.8771707
```

Exercise 8

```
# preamble
```

```
# starter
library(MASS)
```

For Exercises 8 - 15, we will use two related diabetes datasets about the Pima Native Americans (https://en.wikipedia.org/wiki/Pima_people) from the `MASS` package; `Pima.tr` and `Pima.te`. For details, use `?MASS::Pima.tr`. They are essentially a train (`Pima.tr`) and test (`Pima.te`) dataset that are pre-split.

Recall that

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

Use to training data to fit the model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

where

- Y is a binary categorical variable that takes the value 1 when an individual is diabetic according to WHO criteria, 0 if not
- x_1 is `glu`
- x_2 is `ped`

Report the estimate of β_4.

Hint: You do not need to create a response variable with values 1 and 0, instead you can use the factor variable `type`.

```
# solution
library(MASS)
small_polynomial = glm(type ~ glu + I(glu^2) + ped + I(ped^2) + glu:ped,
                       data = Pima.tr, family = "binomial")
unname(coef(small_polynomial)["I(ped^2)"])
```

```
## [1] -0.3595626
```

Exercise 9

```
# preamble
```

```
# starter
library(MASS)
```

Use the model fit in Exercise 8 to obtain a predicted probability of diabetes for each of the individuals in the test dataset (`Pima.te`). What proportion of these probabilities are larger than 0.80?

```
# solution
library(MASS)
small_polynomial = glm(type ~ glu + I(glu^2) + ped + I(ped^2) + glu:ped,
                       data = Pima.tr, family = "binomial")
mean(predict(small_polynomial, Pima.te, type = "response") > 0.80)
```

```
## [1] 0.07228916
```

Exercise 10

```
# preamble
```

```
# starter
library(MASS)
```

Fit an additive logistic regression to model the probability of diabetes using the train dataset, `Pima.tr` , which uses all available predictors in the dataset. Using this as an initial model, use AIC and a backwards stepwise procedure to select a reduced model. How many predictors are used in this reduced model?

```
# solution
library(MASS)
additive = glm(type ~ ., data = Pima.tr, family = "binomial")
additive_selected = step(additive, trace = 0)
length(coef(additive_selected)) - 1
```

```
## [1] 5
```

Exercise 11

```
# preamble
```

```
# starter
library(MASS)
```

Fit a logistic regression to model the probability of diabetes using the train dataset, `Pima.tr` , which uses all available predictors in the dataset as well as all possible two-way interactions. Using this as an initial model, use AIC and a backwards stepwise procedure to select a reduced model. What is the deviance of this reduced model?

```
# solution
library(MASS)
interaction = glm(type ~ . ^ 2, data = Pima.tr, family = "binomial")
interaction_selected = step(interaction, trace = 0)
deviance(interaction_selected)
```

```
## [1] 162.6924
```

Exercise 12

```
# preamble
```

```
# starter
library(MASS)
library(boot)

# fit the models here

set.seed(42)
# get cross-validated results for the polynomial model here
set.seed(42)
# get cross-validated results for the additive model here
set.seed(42)
# get cross-validated results for the model selected from additive model here
set.seed(42)
# get cross-validated results for the interaction model here
set.seed(42)
# get cross-validated results for the model selected from interaction model here
```

Obtain 5-fold cross-validated misclassification rates for each of the previous 5 models used as classifiers that seek to minimize the misclassification rate. (The models from Exercises 8, 10, and 11) Since the data will be split randomly, use the seeds provided to obtain the cross-validated results after fitting the models. Also, use the relevant cross-validation function from the `boot` package to ensure your calculation uses the same splits for grading purposes. (Even with the same seed, the splits could be done differently.)

Report the best cross-validated misclassification rate of these five.

```
# solution
library(MASS)
library(boot)

small_polynomial = glm(type ~ glu + I(glu^2) + ped + I(ped^2) + glu:ped,
                       data = Pima.tr, family = "binomial")
additive = glm(type ~ ., data = Pima.tr, family = "binomial")
additive_selected = step(additive, trace = 0)
interaction = glm(type ~ . ^ 2, data = Pima.tr, family = "binomial")
interaction_selected = step(interaction, trace = 0)

set.seed(42)
res1 = cv.glm(Pima.tr, small_polynomial, K = 5)$delta[1]
set.seed(42)
res2 = cv.glm(Pima.tr, additive, K = 5)$delta[1]
set.seed(42)
res3 = cv.glm(Pima.tr, additive_selected, K = 5)$delta[1]
set.seed(42)
res4 = cv.glm(Pima.tr, interaction, K = 5)$delta[1]
set.seed(42)
res5 = cv.glm(Pima.tr, interaction_selected, K = 5)$delta[1]

min(res1, res2, res3, res4, res5)
```

```
## [1] 0.1597213
```

Exercise 13

```
# preamble
```

```
# starter
library(MASS)
```

Using the additive model previously fit to the training dataset, create a classifier that seeks to minimize the misclassification rate. Report the misclassification rate or this classifier in the test dataest.

```
# solution
library(MASS)
additive = glm(type ~ ., data = Pima.tr, family = "binomial")
mean(ifelse(predict(additive, Pima.te) > 0, "Yes", "No") != Pima.te$type)
```

```
## [1] 0.1987952
```

Exercise 14

```
# preamble
# starter
library(MASS)
```

Using the additive model previously fit to the training dataset, create a classifier that seeks to minimize the misclassification rate. Report the sensitivity of this classifier in the test dataset.

```
# solution
library(MASS)
additive = glm(type ~ ., data = Pima.tr, family = "binomial")
positives = table(
  pred = ifelse(predict(additive, Pima.te) > 0, "Yes", "No"),
  act = Pima.te$type)[, 2]
unname(positives[2] / sum(positives))
```

```
## [1] 0.6055046
```

Exercise 15

```
# preamble
```

```
# starter
library(MASS)
```

Using the additive model previously fit to the training dataset, create a classifier that classifies an individual as diabetic if their predicted probability of diabetes is greater than 0.3. Report the sensitivity of this classifier in the test dataset.

```
# solution
library(MASS)
additive = glm(type ~ ., data = Pima.tr, family = "binomial")
positives = table(
  pred = ifelse(predict(additive, Pima.te, type = "response") > 0.3, "Yes", "No"),
  act = Pima.te$type)[, 2]
unname(positives[2] / sum(positives))
```

```
## [1] 0.7981651
```