```r
rm(list = ls())
# 10. ----
# (a) ----
library(ISLR)

# Do this to put it in the global environment. (Otherwise, this isn't technically necessary)
data(Weekly)

summary(Weekly)
# Alternatively...
stargazer::stargazer(Weekly, type = "text")

plot(Weekly$Year, Weekly$Volume)

library(ggplot2)
ggplot(data = Weekly, aes(x = Year)) +
  geom_point(aes(y = Volume))

Weekly %>%
  tidyr::gather(lag, value, Lag1:Lag5) %>%
  ggplot(aes(x = Year, y = value, color = lag)) +
  geom_point(alpha = 0.1, position = "jitter") +
  geom_smooth(method = lm, size = 2) +
  facet_wrap(~lag, scales = "free")


pairs(Weekly)
# Altneratively...
# GGally::ggpairs(Weekly)

# Error with this...
# cor(Weekly)
cor(Weekly[-9])
cor(Weekly[, -9])
round(cor(Weekly[, -9]), 2)
# Altneratively...
GGally::ggcorr(Weekly[-9], label = TRUE)

# (b) ----
# Note that `Year` is in the first column.
logit_fit <-
  glm(Direction ~ . - Year - Today, data = Weekly, family = binomial)
logit_fit_v2 <-
  glm(Direction ~ ., data = Weekly[, c(2:7, 9)], family = binomial)
summary(logit_fit)

# (c) ----
# Note that this should be 'newdata', not 'data'.
logit_probs <- predict(logit_fit, newdata = Weekly, type = "response")
logit_preds <- ifelse(logit_probs > 0.5, "Up", "Down")
logit_preds_v2 <- rep("Down", length(logit_probs))
logit_preds_v2[logit_probs > 0.5] <- "Up"

conf_mat <- table(predicted = logit_preds, Weekly$Direction)
conf_mat

# Overall accuracy.
(conf_mat[1,1] + conf_mat[2,2]) / sum(conf_mat)
# Accuracy when predicting 'Down'
conf_mat[1,1] / (conf_mat[1,1] + conf_mat[1,2])
# Accuracy when predicting 'Up'
conf_mat[2,2] / (conf_mat[2,1] + conf_mat[2,2])

caret::confusionMatrix(conf_mat, positive = "Up")

# (d) ----
trn_log_idx <- Weekly$Year %in% (1990:2008)
trn_log_idx_v2 <- Weekly$Year < 2009

Weekly_trn <- Weekly[trn_log_idx, ]
Weekly_tst <- Weekly[!trn_log_idx, ]

logit_fit_trn <- glm(Direction ~ Lag2, data = Weekly_trn, family = binomial)
logit_fit_trn_v2 <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset =
trn_log_idx)
logit_fit_trn_v3 <- glm(Direction ~ Lag2, data = Weekly[trn_log_idx, ], family = binomial)
logit_probs_tst <- predict(logit_fit_trn, newdata = Weekly_tst, type = "response")
# This doesn't work...
# logit_probs_tst_v2 <- predict(logit_fit_trn, newdata = Weekly, type = "response", subset =
!trn_log_idx)
logit_probs_tst_v3 <- predict(logit_fit_trn, newdata = Weekly[!trn_log_idx, ], type =
"response")
logit_preds_tst <- ifelse(logit_probs_tst > 0.5, "Up", "Down")

conf_mat_tst <-
  table(logit_preds_tst, Weekly_tst$Direction)
conf_mat_tst

(conf_mat_tst[1,1] + conf_mat_tst[2,2]) / sum(conf_mat_tst)

# Alternatively...
compute_class_err <- function(actual, predicted) {
  mean(actual != predicted)
}

1 - compute_class_err(logit_preds_tst, Weekly_tst$Direction)

logit_tst_roc <- pROC::roc(Weekly_tst$Direction ~ logit_probs_tst, plot = TRUE, print.auc =
TRUE)

# Aside...
create_fmla <- function(var_y, vars_x) {
  fmla <- paste0(var_y, " ~ ", paste(vars_x, collapse = " + "))
  fmla <- as.formula(fmla)
}

# 13. ----
library(MASS)
data(Boston)
summary(Boston)
stargazer::stargazer(Boston, type = "text")
median(Boston$crim)
crim1 <- ifelse(Boston$crim > median(Boston$crim), 1, 0)
Boston2 <- data.frame(crim1, Boston)
Boston2

GGally::ggcorr(Boston, label = TRUE)
cor(Boston)
sort(abs(cor(Boston)[1, ]), decreasing = TRUE)

set.seed(42)
trn_idx <- sample(1:nrow(Boston), nrow(Boston) * 0.7, replace = FALSE)
Boston_trn <- Boston2[trn_idx, ]
Boston_tst <- Boston2[-trn_idx, ]

logit_fit_trn_1 <- glm(crim1 ~ ., data = Boston_trn, family = binomial)
summary(logit_fit_trn_1)
logit_fit_trn_2 <- glm(crim1 ~ rad + tax + lstat + nox, data = Boston_trn, family = binomial)
summary(logit_fit_trn_2)
logit_fit_trn_3 <- glm(crim1 ~ rad + nox, data = Boston_trn, family = binomial)
summary(logit_fit_trn_3)

lm_fit_trn_1 <- lm(crim1 ~ ., data = Boston_trn, family = binomial)
lm_step <- step(lm_fit_trn_1)
summary(lm_step)

logit_fit_trn_4 <- glm(crim1 ~ nox + age + rad + medv, data = Boston_trn, family = binomial)
summary(logit_fit_trn_4)

logit_fit_tst_3_probs <- predict(logit_fit_trn_3, newdata = Boston_tst, type = "response")
logit_fit_tst_3_preds <- ifelse(logit_fit_tst_3_probs > 0.5, 1, 0)

logit_fit_tst_4_probs <- predict(logit_fit_trn_4, newdata = Boston_tst, type = "response")
logit_fit_tst_4_preds <- ifelse(logit_fit_tst_4_probs > 0.5, 1, 0)

conf_mat_fit_tst_3 <- table(logit_fit_tst_3_preds, Boston_tst$crim1)
conf_mat_fit_tst_3
1 - compute_class_err(logit_fit_tst_3_preds, Boston_tst$crim1)
1 - compute_class_err(logit_fit_tst_4_preds, Boston_tst$crim1)
```

```
lm.step(crim1 ~ ., data = Boston_trn)

# trn_control_cv_cls_kappa <-
#   caret::trainControl(method = "cv",
#                       number = 10,
#                       classProbs = TRUE)
# Boston_trn_caret <- Boston_trn
# Boston_trn_caret$crim1 <- factor(ifelse(Boston_trn_caret$crim1 == 1, "yes", "no"), levels =
c("no", "yes"))
# crim1_caret_tst <- Boston_trn_caret$crim1
# set.seed(42)
# logit_cv_1_kappa <-
#   caret::train(
#     crim1 ~ .,
#     data = Boston_trn_caret,
#     method = "glm",
#     trControl = trn_control_cv_cls_kappa,
#     metric = "Kappa"
#   )
# summary(logit_cv_1_kappa)
# logit_cv_1_kappa
# logit_cv_1_kappa$result
# postResample(predict(logit_cv_1_kappa, Boston_trn_caret), crim1_caret_tst)
# confusionMatrix(predict(logit_cv_1_kappa, crim1_caret_tst), crim1_caret_tst)
```