STAA 577: HW4

Matthew Stoebe

Problem 1

Cross Validation is a better way to estimate the models error on unseen data as it splits hte data into multiple train-test splits to fit and test k models on different test sets. This helps reduce the bias of our test set evaluation.

Problem 2

- k-fold compared to validation set
  - adv: More robust method of validation as we validate against many different training sets
  - adv: Reduced bias
  - disadv: Higher computational cost as we most fit k models
- k-fold compared to LOOCV
  - adv: Lower computational cost as we need to fit 1 model for each row in LOOCV
  - adv: Lower Variance
  - disadv: Slightly more biased

Problem 3

##a these probabilities dont shift over time because we are sampling with replacement so it is just 1- p(it is the jth obs) = 1-(1/n) = (n-1)/n

#b This is the same as above becuase the draws are done with replacement

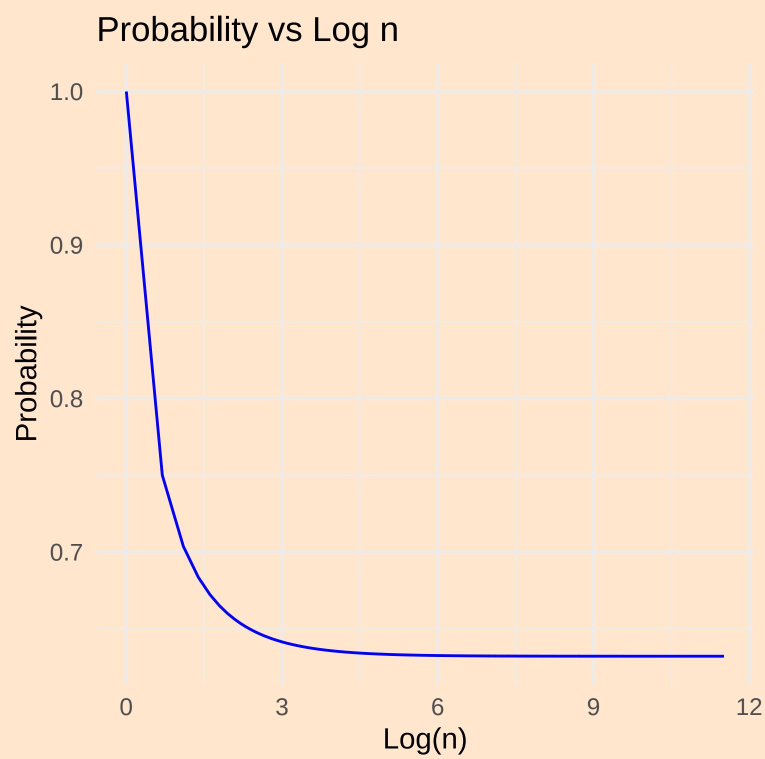
#c bootstrap sample consists of n independant draws so the probability that the jth observation is not in the sample is the probability that it is not in a single sample to the power of n.

#d this is the complement of jth observation not being in the sample so we can do 1-((n-1)/n)^n. thus:~.672

#e ~.634

#f ~.632

#g



Problem 4a

No output needed

Problem 4b and c

```
# Problem 4b
loocv_accuracy
```

```
## [1] 0.8349835
```

LOOCV accuracy: 0.8349835

Problem 4d

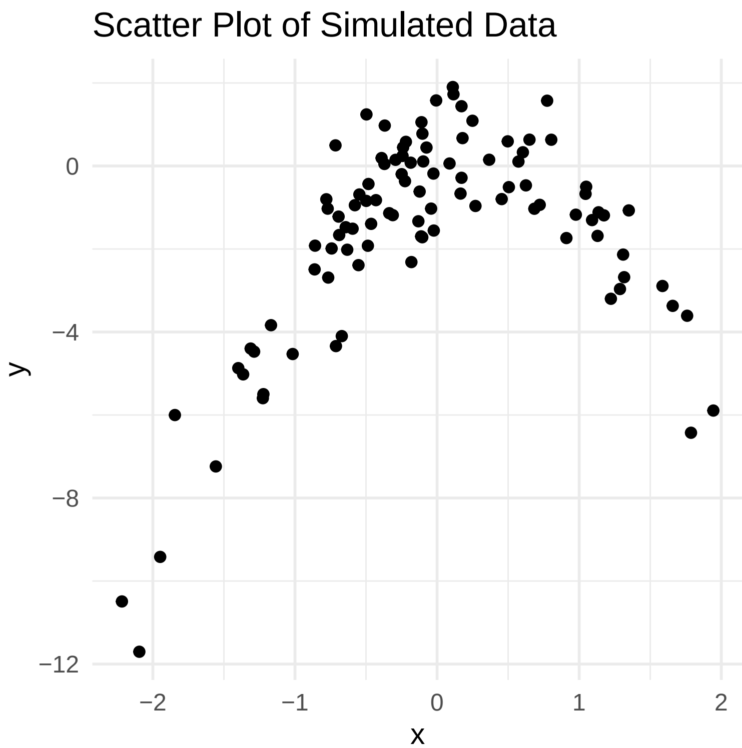
```
## [1] 0.8349835
```

No output needed. Just verifying the LOOCV accuracy.

Problem 5a

No output needed.

Problem 5b



- i see a curve that increases when x < 0 and then decreases x > 0. this sort of parabolic relationship indicates that we are going to need some x^2 type terms to model it well

Problem 5c

```
##       Model     RMSE
## 1    Linear 2.317004
## 2 Quadratic 1.039291
## 3     Cubic 1.024239
## 4   Quartic 1.050147
```

- Model (a) RMSE: 2.33
- Model (b) RMSE: 1.06
- Model (c) RMSE: 1.035
- Model (d) RMSE: 1.031

Problem 5d

- The Quartic model has teh lowest MSE, but it is only marginally better than quadratic and cubic. This makes sense as it is simply a more flexible model. That said, the main performance gain actually came from moving linear to quadratic as this allwoed the model to actually fit the curve. With that in mind, i would just take the quadratic model for simplicity and explanability

Problem 5e

```
##
## Call:
## lm(formula = y ~ x, data = simdata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.866 -1.045  0.538  1.546  3.343
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.5612     0.2340  -6.672 1.52e-09 ***
## x             1.0866     0.2614   4.156 6.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.334 on 98 degrees of freedom
## Multiple R-squared:  0.1499, Adjusted R-squared:  0.1412
## F-statistic: 17.27 on 1 and 98 DF,  p-value: 6.927e-05
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2), data = simdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.65820 -0.64347 -0.01999  0.58721  2.48853
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.009671   0.129894   0.074    0.941
## x            0.973753   0.115931   8.399 3.77e-13 ***
## I(x^2)      -1.969616   0.098171 -20.063  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.034 on 97 degrees of freedom
## Multiple R-squared:  0.8349, Adjusted R-squared:  0.8315
## F-statistic: 245.3 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2) + I(x^3), data = simdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52564 -0.66283  0.07511  0.63083  2.33127
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06008    0.13356   0.450    0.654
## x            1.25708    0.22400   5.612 1.93e-07 ***
## I(x^2)      -2.02577    0.10474 -19.340  < 2e-16 ***
## I(x^3)      -0.13072    0.08862  -1.475    0.143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.028 on 96 degrees of freedom
## Multiple R-squared:  0.8386, Adjusted R-squared:  0.8335
## F-statistic: 166.2 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2) + I(x^3) + I(x^4), data = simdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51026 -0.67711  0.07539  0.58234  2.37631
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.11088    0.15456   0.717    0.475
## x            1.22949    0.22853   5.380 5.34e-07 ***
## I(x^2)      -2.20478    0.29130  -7.569 2.41e-11 ***
## I(x^3)      -0.10631    0.09629  -1.104    0.272
## I(x^4)       0.05316    0.08069   0.659    0.512
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.031 on 95 degrees of freedom
## Multiple R-squared:  0.8393, Adjusted R-squared:  0.8325
## F-statistic: 124.1 on 4 and 95 DF,  p-value: < 2.2e-16
```

We once again see that only the first two terms are signifficant. there is marginal improvement from adding the cubic and quartic terms in terms of MSE, but the coefficients are not signifficant. This aligns with my previous answer

Problem 5f

```
## [1] 0.9617447
```

The result is different because the random seed is different. If this new seed only affected the kfolds this large of a decrease in RMSE would be suppriseing but because the random seed is affecting both data generation and kfold, the new rmse value is reallly not comparable to the old one.

Problem 5g

Confused here, I dont see a 5G in the homework

Appendix

```r
library(knitr)
# install the tidyverse library (do this once) install.packages('tidyverse')
library(tidyverse)
# set chunk and figure default options
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE, fig.width = 4,
    fig.height = 4, tidy = TRUE)
library(ggplot2)

n_vals <- 1:1e+05
prob_in_bootstrap <- 1 - (1 - 1/n_vals)^n_vals

plot_data <- data.frame(n = n_vals, prob = prob_in_bootstrap)

ggplot(plot_data, aes(x = log(n), y = prob)) + geom_line(color = "blue") + labs(title = "Probability vs
    x = "Log(n)", y = "Probability") + theme_minimal()

# Problem 4a

train <- read.csv("heart_training.csv")
test <- read.csv("heart_test.csv")

heart <- rbind(train, test)

heart$sex <- factor(heart$sex)
heart$cp <- factor(heart$cp)
heart$exang <- factor(heart$exang)
```

```r
heart$restecg <- factor(heart$restecg)

n <- nrow(heart)
correct <- rep(NA, n)

for (i in 1:n) {
    trainData <- heart[-i, ]
    testData <- heart[i, , drop = FALSE]

    model <- glm(target ~ age + sex + cp + trestbps + thalach + exang + oldpeak +
        ca + thal, data = trainData, family = binomial)

    prob <- predict(model, newdata = testData, type = "response")
    pred <- ifelse(prob > 0.5, 1, 0)

    correct[i] <- as.numeric(pred == testData$target)
}

loocv_accuracy <- mean(correct)


# Problem 4b
loocv_accuracy

# Problem 4d
library(boot)

errorfun <- function(obs, phat = 0) {
    mean(round(phat) != obs)
}

glmOut <- glm(target ~ age + sex + cp + trestbps + thalach + exang + oldpeak + ca +
    thal, data = heart, family = binomial)

cvOut <- cv.glm(data = heart, glmOut, cost = errorfun, K = n)

accur = 1 - cvOut$delta[1]
accur
# Problem 5a
set.seed(577)
x <- rnorm(100)
y <- x - 2 * x^2 + rnorm(100)
simdata <- data.frame(x = x, y = y)
# Problem 5b

ggplot(simdata, aes(x = x, y = y)) + geom_point() + theme_minimal() + labs(title = "Scatter Plot of Sim
    x = "x", y = "y")

# Problem 5c
library(caret)
trainCtrlOpts <- trainControl(method = "cv", number = 10)

model1 <- train(y ~ x, data = simdata, method = "lm", trControl = trainCtrlOpts)
```

```r
rmse1 <- model1$results$RMSE

model2 <- train(y ~ x + I(x^2), data = simdata, method = "lm", trControl = trainCtrlOpts)
rmse2 <- model2$results$RMSE

model3 <- train(y ~ x + I(x^2) + I(x^3), data = simdata, method = "lm", trControl = trainCtrlOpts)
rmse3 <- model3$results$RMSE

model4 <- train(y ~ x + I(x^2) + I(x^3) + I(x^4), data = simdata, method = "lm",
    trControl = trainCtrlOpts)
rmse4 <- model4$results$RMSE

rmse_values <- data.frame(Model = c("Linear", "Quadratic", "Cubic", "Quartic"), RMSE = c(rmse1,
    rmse2, rmse3, rmse4))
rmse_values

# Problem 5e
lm1 <- lm(y ~ x, data = simdata)
summary(lm1)

lm2 <- lm(y ~ x + I(x^2), data = simdata)
summary(lm2)

lm3 <- lm(y ~ x + I(x^2) + I(x^3), data = simdata)
summary(lm3)

lm4 <- lm(y ~ x + I(x^2) + I(x^3) + I(x^4), data = simdata)
summary(lm4)

# Problem 5f
set.seed(1500)
x_new <- rnorm(100)
y_new <- x_new - 2 * x_new^2 + rnorm(100)
simdata_new <- data.frame(x = x_new, y = y_new)

model2_new <- train(y ~ x + I(x^2), data = simdata_new, method = "lm", trControl = trainCtrlOpts)
rmse2_new <- model2_new$results$RMSE
rmse2_new
```