Conceptual Exercises

Import notebook functions

```
from notebookfuncs import *
```

Exercise 1

Using basic statistical properties of the variance, as well as single-variable calculus, derive

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}^2}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}^2}$$

.

In other words, prove that α given by the equation above does indeed minimize $Var(\alpha X + (1-\alpha)Y)$.

By statistical properties of the variance, we have

$$\begin{aligned}
&Var(aX + bY) = Var(aX) + Var(bY) + 2Cov(aX, bY) \\
&= a^2 Var(X) + b^2 Var(Y) + 2abCov(X, Y) \\
&\therefore Var(\alpha X + (1-\alpha)Y) \\
&= \alpha^2 Var(X) + (1-\alpha)^2 Var(Y) + 2\alpha(1-\alpha)Cov(X, Y) \\
&= \alpha^2 Var(X) + Var(Y) + \alpha^2 Var(Y) - 2\alpha Var(Y) + 2\alpha Cov(X.Y) - 2\alpha^2 Cov(X, Y) \\
&= \alpha^2 Var(X) + \alpha^2 Var(Y) - 2\alpha^2 Cov(X, Y) - 2\alpha Var(Y) + 2\alpha Cov(X.Y) + Var(Y) \\
&= \alpha^2(Var(X) + Var(Y) - 2Cov(X, Y)) - 2\alpha(Var(Y) - Cov(X, Y)) + Var(Y)
\end{aligned}$$

Using single-variable calculus and differentiating the above equation w.r.t α and equating it to zero to discover the value of α that minimizes the variance, we have

$$
\begin{aligned}
&2\alpha(Var(X) + Var(Y) - 2Cov(X,Y)) - 2(Var(Y) - Cov(X,Y)) = 0 \\
&\implies 2\alpha(Var(X) + Var(Y) - 2Cov(X,Y)) = 2(Var(Y) - Cov(X,Y)) \\
&\implies \alpha(Var(X) + Var(Y) - 2Cov(X,Y)) = Var(Y) - Cov(X,Y) \\
&\implies \alpha = \frac{Var(Y) - Cov(X,Y)}{Var(X) + Var(Y) - 2Cov(X,Y)} \\
&\implies \alpha = \frac{\sigma_Y^2 - \sigma_{XY}^2}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}^2}
\end{aligned}
$$

Exercise 2

We will now derive the probability that a given observation is part of a bootstrap sample. Suppose that we obtain a bootstrap sample from a set of n observations.

(a)

What is the probability that the first bootstrap observation is not the j_{th} observation from the original sample? Justify your answer.

We are obtaining an observation from a set of n observations. ∴ the probability that the first bootstrap observation is the j_{th} observation is $\frac{1}{n}$. ∴ the probability that it is not is $1 - \frac{1}{n}$.

(b)

What is the probability that the second bootstrap observation is not the j_{th} observation from the original sample?

We are sampling with replacement since bootstrap implies the same, the probability that the second observation is not the j_{th} observation is $(1 - \frac{1}{n})$ as well.

(c)

Argue that the probability that the j_{th} observation is not in the bootstrap sample is $(1 - \frac{1}{n})^n$.

Sampling using bootstrap method involves sampling with replacement from a dataset with n items. Each observation has probability of 1/n of being selected at each draw and thus (1 - 1/n) probability of not being chosen at each draw. Since each draw is independent of each

other, the probability that a select observation is not selected in the entire bootstrap sample is the product of their individual draw probabilities and is equivalent to the following:

$$\prod_{i=1}^{n}(1-\frac{1}{n})$$
$$=(1-\frac{1}{n})^n$$

Thus, the probability that the j_{th} observation is not in the entire bootstrap sample of size n is $(1-\frac{1}{n})^n$.

(d)

When n = 5, what is the probability that the j_{th} observation is in the bootstrap sample?

```
import sympy
from sympy import Symbol, N
n = Symbol('n')
prob_j = 1 - (1 - 1 / n) ** n
N(prob_j.subs(n, 5))
```

0.67232

(e)

When n = 100, what is the probability that the j_{th} observation is in the bootstrap sample?

```
N(prob_j.subs(n, 100))
```

0.633967658726771

(f)

When n = 10, 000, what is the probability that the j_{th} observation is in the bootstrap sample?

```
N(prob_j.subs(n, 10000))
```

0.63213895356707

(g)

Create a plot that displays, for each integer value of n from 1 to 100, 000, the probability that the j_{th} observation is in the bootstrap sample. Comment on what you observe.

I've drawn the x-axis as log-scale since the curve appears to be just two lines, one vertical and the other horizontal over the linear values. I've also drawn the 0.632 bootstrap limit as $(1 - \frac{1}{e})$.

We can prove that $\lim_{n\to\infty}(1 - 1/n)^n = 1/e$

$$\begin{aligned} \text{Let } y &= (1 - 1/n)^n \\ \implies \ln y &= n \ln (1 - 1/n) \\ \implies \lim_{n\to\infty} y &= \lim_{n\to\infty} n \ln (1 - 1/n) \\ &= \lim_{n\to\infty} \frac{\ln (1 - 1/n)}{(1/n)} \end{aligned}$$

Now, by L'Hospital's rule that

$$\lim_{n\to\infty} \frac{f(x)}{g(x)} = \lim_{n\to\infty} \frac{f^{'}(x)}{g^{'}(x)}$$

We have

$$\begin{aligned} &\frac{d}{dn}\Big(\ln (1 - 1/n)\Big) \\ &= \frac{1}{1 - 1/n}(1/n^2) = \frac{n}{n-1}(1/n^2) \end{aligned}$$

And

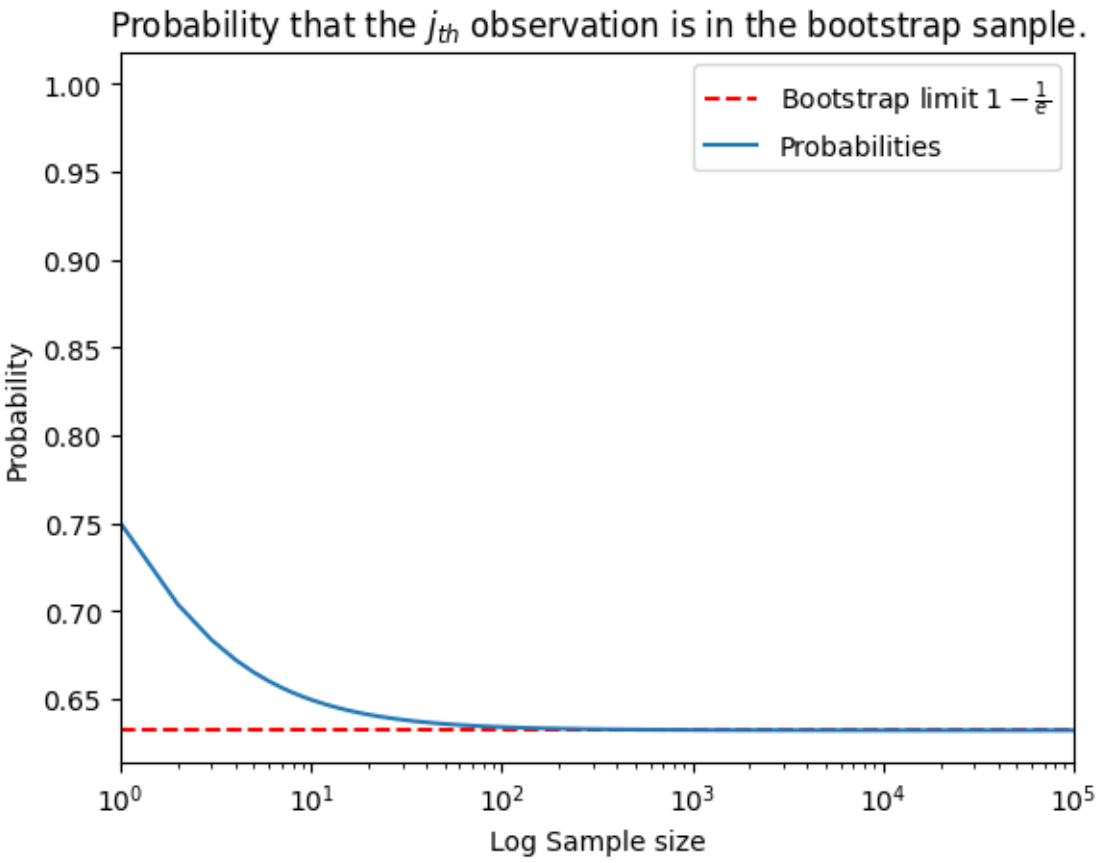
$$\begin{aligned} &\frac{d}{dn}\Big(1/n\Big) \\ &= -(1/n^2) \end{aligned}$$

Substituting and canceling, we have

$$\begin{aligned} &\lim_{n\to\infty} (-1)\frac{n}{n-1} \\ \text{Since } \lim_{n\to\infty} \frac{n}{n-1} \text{ tends to } 1 \quad &\text{(Again, by L'Hospital's rule)} \\ \implies \lim_{n\to\infty} \ln y &= (-1)(1) = -1 \\ \implies y &= e^{-1} = 1/e \end{aligned}$$

Thus, the bootstrap probability limit tends to $1 - (1 - 1/n)^n \approx 1 - (1/e)$.

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import mode
from math import e
probs = np.zeros(100000)
for i in range(100000):
  n = i + 1
  probs[i] = 1 - (1 - 1 / n) ** n
plt.title("Probability that the $j_{th}$ observation is in the bootstrap sanple.")
plt.xlabel("Log Sample size")
plt.ylabel("Probability");
plt.gca().set_xscale("log")
plt.gca().set_xlim((1, 100000))
plt.gca().axhline(1 - 1/e, linestyle='--', label="Bootstrap limit $1 - \\frac{1} {e}$", c="r'
plt.plot(probs, label="Probabilities")
plt.legend();
```




```
printmd("Here, we can see that as the size of the bootstrap sample increases, the probability
printmd(f"**{mode(probs)[0]:.4f}**.")
```

Here, we can see that as the size of the bootstrap sample increases, the probability of selecting the j_{th} samples stabilizes at

0.6321.

This limit of 0.632 is also termed as the '0.632 bootstrap rule'.

(h)

We will now investigate numerically the probability that a bootstrap sample of size n = 100 contains the j_{th} observation. Here j = 4. We first create an array store with values that will subsequently be overwritten using the function `np.empty()`. We then repeatedly create bootstrap samples, and each time we record whether or not the fifth observation is contained in the bootstrap sample.

```
rng = np.random.default_rng(10)
store = np.empty(10000)
for i in range(10000):
  store[i] = np.sum(rng.choice(100, size=100, replace=True) == 4) > 0
np.mean(store)
```

Comment on the results obtained.

```
rng = np.random.default_rng(10)
store = np.empty(10000)
for i in range(10000):
  # This stores the number of bootstrap samples that contain 4
  store[i] = np.sum(rng.choice(100, size=100, replace=True) == 4) > 0
np.mean(store)
```

```
0.6362
```

Note: We had to change the code above so that 100 numbers are sampled from the range 0 - 99 and thus 10000 bootstrap samples are generated. The earlier code only sampled 1 element at a time.

```
printmd(f"Thus, we see that the probability closely matches what we obtained theoretically at
```

Thus, we see that the probability closely matches what we obtained theoretically at 0.6362.

Exercise 3

We now review k-fold cross-validation.

(a)

Explain how k-fold cross-validation is implemented.

The purpose of cross-validation is to estimate a test statistic or metric. This is one of the methods utilized to estimate a test statistic when there is no test or validation dataset available to estimate the metric. The metric could be MSE or Accuracy or any other metric. It doesn't really matter.

K-fold cross-validation involves splitting the input dataset into k folds (usually 5 or 10) of equal size. One of the folds is held out to be used as the validation dataset while the model is trained on the remaining k - 1 folds. This process is repeated k times and the test performance is recorded and averaged. This metric is the cross-validation or out-of-sample metric.

(b)

What are the advantages and disadvantages of k-fold cross-validation relative to:

i. The validation set approach?

The validation set approach involves randomly dividing the available set of observations into two parts, a training set and a validation set or hold-out set. The model is fit on the training set, and the fitted model is used to predict the set responses for the observations in the validation set. The resulting validation hold-out set set error rate provides an estimate of the test error rate.

Advantages:

1. The validation set approach estimate can be highly variable since it depends on which observations are selected in the training set. This compares unfavorably with the k-fold cross-validation approach where the test variance is smaller.
2. In the validation approach, only half the observations — those that are included in the training set rather than in the validation set — are used to fit the model. Statistical methods perform worse when trained on fewer observations. The validation set error rate thus overestimates the test error rate for the model fit on the entire data set.
3. All the data is used to train the models unlike as in the Validation set approach.

Disadvantages:

1. The validation set approach is conceptually simple and easier to explain as well as implement. The k-fold cross-validation approach is not as simple to explain or understand.
2. The validation set approach is computationally inexpensive and has to be performed just once. The k-fold approach depends on the value chosen for k, usually 5 or 10 — empirically proven to give the best results.

ii. LOOCV?

Leave-one-out cross-validation (LOOCV) attempts to address the Validation Set method's drawbacks. Here, a single observation is used as the validation dataset and the remaining observations make up the training dataset.The procedure is repeated n times leaving out one observation at a time. This process produces n metrics which are then averaged out.

Advantages

1. Computationally less expensive than LOOCV. In 10-fold cross-validation, the model has to be fit only 10 times. In LOOCV, it would be n times where n is the dataset size.

2. While LOOCV would be less biased than K-fold, the trade-off is that there is less variability in the estimates from a k-fold cross-validation procedure. LOOCV has higher variance than does k-fold CV with $k < n$. When we perform LOOCV, we are averaging the outputs of n fitted models, each of which is trained on an almost identical set of

observations; therefore, these outputs are highly (positively) correlated with each other. In contrast, when we perform k-fold CV with k < n, we are averaging the outputs of k fitted models that are somewhat less correlated with each other, since the overlap between the training sets in each model is smaller. Since the mean of many highly correlated quantities has higher variance than does the mean of many quantities that are not as highly correlated, the test error estimate resulting from LOOCV has higher variance than does the test error estimate resulting from k-fold CV.

3. The cross-validation approach tends to underestimate the the actual test error rate. However, sometimes we are more interested in the location of the minimum point in the estimated MSE curve. That's because we may perform cross-validation on a number of statistical learning methods or on a single method with varying degrees of flexibility. Hence, the location of the minimum point is important and not it's actual value.

Disadvantages

1. With least squares linear or polynomial regression, the cost of LOOCV is the same as that of a single fit when we use the following equation:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{yi}}{1 - h_i}\right)^2$$

   This relies on calculating the leverage for each point which reflects how much an observation influences its own fit. The leverage lies between 1/n and 1 and thus the high leverage points' residuals are inflated by exactly that amount in the equation above. This magic formula does not hold in general and hence LOOCV has to be computed n times for other statistical methods.

2. There is an *element of randomness* in the k-fold approach depending on how data is split into k-folds. LOOCV does not have this drawback.

Exercise 4

Suppose that we use some statistical learning method to make a prediction for the response Y for a particular value of the predictor X. Carefully describe how we might estimate the standard deviation of our prediction.

The bootstrap is a widely applicable and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method. The bootstrap's power lies in the fact that it can be easily applied to a wide range of statistical learning methods, including some for which a measure of variability is otherwise difficult to obtain and is not automatically output by statistical software.

The bootstrap can be performed on a dataset with n observations. We randomly select n observations from this dataset with replacement. We can use this new dataset to make a

prediction for a particular value of the predictor X. We can perform this several times, say B, a large number. The array of B values for the predictor is our result. We then calculate the standard deviation of these predictions. This is the standard deviation of our bootstrapped estimates for the prediction. This serves as an estimate for the standard deviation of the prediction obtained from the original dataset.

```
allDone();
```

```
<IPython.lib.display.Audio object>
```