Dylan DeNicola, Natalie Cardoso
COGS 4210, RPI
Spring 2024

Homework 1

Problem 1:

1. **FALSE**
2. TRUE
3. **FALSE**
4. **FALSE**
5. **FALSE**
6. TRUE
7. **FALSE**
8. TRUE

Problem 2:

1. GitHub Repository: https://github.com/nataliecar/cognitive-modeling/tree/main
2. Merge conflict created and resolved in GitHub commits!
3. Explain differences:
   (a) git restore
   Used to restore the working tree by bringing back the files to their last committed state.
   (b) git checkout
   Used to change branches. Also, can discard uncommitted changes from the previous branch.
   (c) git reset
   Used to reset the current branch to a previous commit. Removes the record of all commits after the specified reset.
   (d) git revert
   Used to create a new commit that reverts undoes a previous commit. Keep the record of commits intact.

Problem 3:

Expected Return = (Probability-of-Market-Going-Up * Return-if-Market-Goes-Up) +

(Probability-of-Market-Going-Down * Return-if-Market-Goes-Down)

= (0.8 * 0.01) + (0.2 * (-0.10)) = 0.008 - 0.02 = **-0.012**

The expectation of -0.012 indicates a potential loss, therefore I would not invest in this market.

Set the Expected Return to 0 (i.e. breaking even):

0 = (p * 0.01) + ((1 - p) * (-0.10)) = 0.01p - 0.10 + 0.10p

0.10 = 0.11p

p = 0.909 = **90.9%**

A limitation of expectations when making single-shot, real-life decisions is that they do not consider risk associated with the outcomes. In this example, in the possibility the market went down oil prices would drop ten times the amount they would rise if the market went up. Real-life decision require a more comprehensive analysis that include both expected values and risk.

Problem 4:

1. We will show the following: $Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$

$$Var[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$$
$$Var[X] = \mathbb{E}[X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2]$$
$$Var[X] = \mathbb{E}[X^2 - \mathbb{E}[X](2X - \mathbb{E}[X])]$$
$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[\mathbb{E}[X](2X - \mathbb{E}[X])]$$
$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]\mathbb{E}\big[2X - \mathbb{E}[X]\big]$$
$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]\big(\mathbb{E}[2X] - \mathbb{E}\big[\mathbb{E}[X]\big]\big)$$
$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X](2\mathbb{E}[X] - \mathbb{E}[X])$$
$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]\mathbb{E}[X]$$
$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

2. We will show the following: $Var[\alpha X + \beta] = \alpha^2 Var[X]$

$$Var[\alpha X + \beta] = \mathbb{E}[(\alpha X + \beta - \mathbb{E}[\alpha X + \beta])^2]$$
$$Var[\alpha X + \beta] = \mathbb{E}[(\alpha X + \beta - \alpha\mathbb{E}[X] - \beta)^2]$$
$$Var[\alpha X + \beta] = \mathbb{E}[(\alpha X - \alpha\mathbb{E}[X])^2]$$
$$Var[\alpha X + \beta] = \alpha^2\mathbb{E}[(X - \mathbb{E}[X])^2]$$
$$Var[\alpha X + \beta] = \alpha^2 Var[X]$$

3. We will show the transformation: $X \sim Normal(\mu = 0, \sigma = 1) \rightarrow \tilde{X} \sim Normal(\mu = 3, \sigma = 5)$
   Suppose we have a normal distribution X_0 with μ_0 and σ_0. We want to get to a new normal distribution X_1 with μ_1 and σ_1.
   We can apply the following linear transformation to do this:

$$X_1 = \mu_1 + \left(\frac{\sigma_1}{\sigma_0}\right) * (X_0 - \mu_0)$$

   We can use this linear transformation with $X = X_0$ and $\tilde{X} = X_1$:

$$\tilde{X} = 3 + \left(\frac{5}{1}\right) * (X - 0)$$
$$\tilde{X} = 3 + 5 * X$$

   There is a Jupyter notebook on the GitHub that demonstrates this result.

Problem 5:

Let S be a random variable that signifies if the statement is the truth or a lie. We have that $R_S = \{truth, lie\}$. Let Y be the probability that "yes" response is true.

We know the following information:

$$P(S = truth) = \frac{1}{3}$$

Homework 1

$$P(S = lie) = \frac{2}{3}$$

$$P(Y \mid S = truth) = \frac{1}{3}$$

$$P(Y \mid S = lie) = \frac{2}{3}$$

We need to find the probability of the statement being true. Thus, we need to find $P(S = truth \mid Y)$.

We can do this using Bayes' Rule:

$$P(S = truth \mid Y) = \frac{P(Y \mid S = truth)P(S = truth)}{P(Y)}$$

We can find $P(Y = truth)$ using the sum and product rules:

$$P(Y) = \sum_{S \in R_S} P(Y \mid S)P(S) = P(Y \mid S = truth)P(S = truth) + P(Y \mid S = lie)P(S = lie)$$

$$P(Y) = \left(\frac{1}{3}\right)\left(\frac{1}{3}\right) + \left(\frac{2}{3}\right)\left(\frac{2}{3}\right) = \frac{5}{9}$$

Therefore, we have the following probability:

$$P(S = truth \mid Y) = \frac{\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)}{\left(\frac{5}{9}\right)} = \frac{1}{5}$$

There is a probability of $\frac{1}{5}$ that the statement was true.

Problem 6: There is a Jupyter notebook on the GitHub that contain this problem's solution.

Problem 7:

First, we will discuss how the posterior probability changes as a function of the prior probability. We see from the simulation that the curve of this relationship is logarithmic. This means that as the prior probability initially increases, the posterior probability quickly increases, but as the prior probability gets larger, the posterior probability increases at a slower rate.

Next, we will discuss how the posterior probability changes as a function of the sensitivity. We see from the simulation that the curve of this relationship is linear. This means that as the sensitivity increases, the posterior probability quickly increases, and that the rate at which the posterior probability increases stays constant.

Lastly, we will discuss how the posterior probability changes as a function of the specificity. We see from the simulation that the curve of this relationship is exponential. This means that as the prior probability initially increases, the posterior probability slowly increases, but as the prior probability gets larger, the posterior probability increases at a faster rate.

There is a Jupyter notebook on the GitHub showing these 2D graphs as well as the 3D graphs for the bonus.

Problem 8: There is a Jupyter notebook on the GitHub that contain this problem's solution.

Problem 9:

We used ChatGPT 3.5 to generate the function. We also specified that it does not use any SciPy functions when creating its function. It performed very well because it gave practically the exact same answer as SciPy's function did. The differences between ChatGPT and SciPy were on orders of magnitude such around 10^{-17}. This good performance was consistent across all three Gaussian types. A Jupyter notebook with ChatGPT's function and the comparisons to SciPy are on the GitHub.