Name: Richard Duncan

Problem	Points	Score
1a - done	10	
1b - done	10	
1c - done	10	
1d - done	10	
2a - done	10	
2b - done	10	
2c - done	10	
3a - done	10	
3b - done	10	
3c - done	10	
3d - done	10 (if rework is all correct)	
Total	100	

Notes:

1. The exam is closed books/closed notes - except for one page (double-sided) of notes.

2. Please show ALL work. Answers with no supporting explanations or work will be given no credit.

3. Please indicate clearly your answer to the problem. If I can’t read it (and I am the judge of legibility), it is wrong. If I can’t follow your solution (and I get lost easily), it is wrong. All things being equal, neat and legible work will get the higher grade:)

Problem No. 1: General Concepts

(a) Explain the relationship between mutual information and relative entropy. Do not simply write the equations or restate them in words — show some insight. (Hint: recall the discussion in Chap. 1.)

Mutual information is the degree to which one variable is determined once we know another variable, it is a measure of the dependence between the two random variables. Relative entropy is the distance between two distributions. Mutual information is a special case of relative entropy, useful only when the relative entropy is sufficiently small and the distributions overlap. Relative entropy is a more general because it will always have a meaningful value. If two distributions are independent, then the mutual information is zero. But, independent distributions p_0 and q_0 may be closer together than independent distributions p_1 and q_1. Therefore, algorithmically it makes more sense to minimize relative entropy rather than maximize mutual information, since relative entropy will always be defined.

(b) Explain the impact of information theory on the data compression and data transmission problems (Hint: recall the discussion in Chap. 1.)

Before Shannon upset the telecommunication world with his information theory, it was thought that increasing transmission speed would increase the probability of errors. Information theory, however, proved that up to a certain limit (channel capacity, not yet formally defined in class) we can achieve error free communication. In the 1940's, Shannon characterized the analog voice POTS network and found a very close estimate to the highest transmission rates we are now achieving of 33.6 kps —50 years later. While he did not give us the code, he set the mathematical limit.

Also, he introduced the concept of entropy. Entropy is the lower limit on data compression, which is a measure of the underlying uncertainty in data. He named it entropy due to its similarity to the thermodynamics concept.

(c) Prove the chain rule for relative entropy.

The chain rule for relative entropy states:
$D(p(x, y) \| q(x, y)) = D(p(x) \| q(x))) + D(p(y|x) \| q(y|x))$.
In words, the distance between the joint distributions $p(x, y)$ and $q(x, y)$ is equivalent to the distance between the distributions $p(x)$ and $p(y)$ plus the distance between the conditional distributions $p(y|x)$ and $q(y|x)$.

The proof for this theorem is very strait forward. Relative entropy is defined as
$$D(p(x) \| q(x)) = \sum_x p(x) \cdot \log\frac{p(x)}{q(x)}.$$

Substituting in the joint distributions, we have:
$$D(p(x, y) \| q(x, y)) = \sum_x \sum_y p(x, y) \cdot \log\frac{p(x, y)}{q(x, y)}$$

Since probability theory gives us that $p(x, y) = p(x) \cdot p(y|x)$,

$$D(p(x, y) \| q(x, y)) = \sum_x \sum_y p(x, y) \cdot \log\frac{p(x) \cdot p(y|x)}{q(x) \cdot q(y|x)}.$$

Through the logarithmic identity $\log xy = \log x + \log y$, we can split the sum into

$$D(p(x, y) \| q(x, y)) = \sum_x \sum_y p(x, y) \cdot \log\frac{p(x)}{q(x)} + \sum_x \sum_y p(x, y) \cdot \log\frac{p(y|x)}{q(y|x)}$$

The last step in the proof is to get rid of the joint pdfs. The first term can be simplified via the marginal pdf of x, specifically $p_x(x) = \sum_y p_{xy}(x, y)$. For the second term, we again apply $p(x, y) = p(x) \cdot p(y|x)$. Removing the joint pdfs, we are left with.

$$D(p(x, y) \| q(x, y)) = \sum_x p(x) \cdot \log\frac{p(x)}{q(x)} + \sum_x \sum_y p(x) \cdot p(y|x) \cdot \log\frac{p(y|x)}{q(y|x)}.$$

(proof completed on the next page)

Rearranging the order of summation in the second term allows us to apply $\sum_x p(x) = 1$, leaving

$$D(p(x, y) \| q(x, y)) = \sum_x p(x) \cdot \log \frac{p(x)}{q(x)} + \sum_y p(y|x) \cdot \log \frac{p(y|x)}{q(y|x)} \quad .$$

$$= D(p(x) \| q(x)) + D(p(y|x) \| q(y|x))$$

(d) Prove the independence bound on entropy:

Let $X_1, X_2, \ldots, X_n$ be drawn from $p(x_1, x_2, \ldots, x_n)$. The independence bound on entropy states that the highest possible entropy for this set of variables is when they are all independent. The proof is simple if we use the chain rule of entropy and the fundamental concept that conditioning reduces entropy.

By the chain rule of entropy, $H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i | X_{i-1}, \ldots, X_1)$.

If X and Y are independent, then $H(X|Y) = H(X)$. Stated in words, knowing Y doesn't tell us anything at all about X. This the equality condition of the fundamental notion that conditioning reduces entropy, $H(X|Y) \le H(X)$.

So, if each X_i is independent from $X_{i-1}, X_{i-2}, \ldots, X_1$, then for each i we can simplify $H(X_i | X_{i-1}, \ldots, X_1) = H(X_i)$.

Independence is the worst case, in general $H(X_i | X_{i-1}, \ldots, X_1) \le H(X_i)$.

This leaves us with,

$$H(X_1, X_2, \ldots, X_n) \le \sum_{i=1}^{n} H(X_i),$$

with equality when each X_i is independent.

Problem No. 2: Calculations

(a) What is the relationship between $H(Y)$ and $H(X)$ when $Y = \cos X$ and X is a random variable that takes on a finite number of values.

If X and Y are random variables and Y is a function of X such that $Y = \cos X$, then

$$H(Y) = -\sum_{y} p(Y) \cdot \log p(Y) = -\sum_{x} p(\cos X) \cdot \log p(\cos X)$$

Since Y is a function of X, $H(Y|X) = 0$, in other words Y is completely determined once we know X.

The mutual information, $I(Y;X) = H(Y) - H(Y|X) = H(Y) = I(X;Y)$, since $H(Y|X) = 0$.

However, $I(X;Y) = H(X) - H(X|Y) \leq H(X)$, since $H(X|Y) \geq 0$. This is true because all values of X do not map onto a unique value of Y, the cosine operator is not always a one-to-one transformation. I include the equality condition in these expressions because certain ranges of X will result in a one-to-one operator in which $H(X|Y) = 0$, one such range is $-\pi < X \leq \pi$. Other ranges less than one period of the cosine wave are also acceptable. For any range greater than 2π, however, knowing Y does not completely determine the value of X.

In conclusion, $H(X) \geq H(Y)$.

(b) Show that if $H(Y|X) = 0$, then Y is a function of X, i.e., for all x with $p(x) > 0$, there is only one possible value of y with $p(x, y) > 0$.

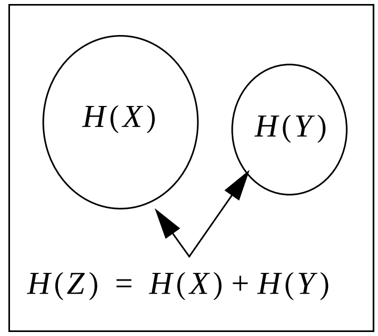
Intuitively, if $H(Y|X) = 0$, then there is no uncertainty left in Y once we know X, hence Y must be a function of X.

We will prove this through contradiction. Assume that x_0 is some value of x such that $p(x_0) > 0$ and for two different values of y (y_0 and y_1) $p(x_0, y_0) > 0$ and $p(x_0, y_1) > 0$. Then $p(x_0) \geq p(x_0, y_0) + p(x_0, y_1) > 0$, and neither $p(y_0|x_0)$ nor $p(y_1|x_0)$ are equal to 0 or 1. Thus the conditional entropy, $H(Y|X) = -\sum_x p(x) \cdot \sum_y p(y|x) \cdot \log p(y|x)$, can be expressed as $H(Y|X) \geq p(x_0) \cdot (-p(y_0|x_0) \cdot \log p(y_0|x_0) - p(y_1|x_0) \cdot \log p(y_1|x_0))$. Since $-t \cdot \log t \geq 0$ for all $0 \leq t \leq 1$, more strictly $-t \cdot \log t > 0$ for all $0 < t < 1$.

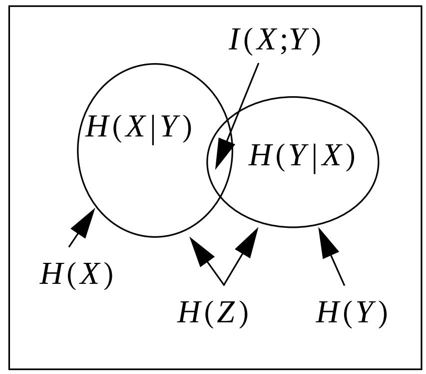
Applying this logarithmic identity, $H(Y|X) > 0$ since $p(y_0|x_0)$ and $p(y_1|x_0)$ are not equal to 0 or 1.

Since $H(Y|X) > 0$ contradicts the problem statement that $H(Y|X) = 0$, for all x with $p(x) > 0$, there is only one possible value of y with $p(x, y) > 0$.

(c) X **and** Y **are random variables, and let** $Z = X + Y$**. Show that** $H(Z|X) = H(Y|X)$**.**



In words, $H(Z|X)$ is the uncertainty of Z once we know X, and $H(Y|X)$ is the uncertainty of Y once we know X. Once we know X, the only source of uncertainty left for Z is in Y, obviously bounded by $H(Y)$. This can be expressed mathematically by $H(Z|X) \le H(Y)$, with equality when X and Y are independent. If X and Y are independent, the Venn diagram is shown by the topmost figure on the right. In the independent case, $I(X;Y) = 0$. However, if X and Y are not independent, then the second Venn diagram is more appropriate, with $I(X;Y) \ge 0$.



Recalling the definition of conditional entropy, $H(Z|X) = \sum p(x) \cdot H(Z|X = x)$.

Expanding $H(Z|X = x)$ by the definition of entropy,

$$H(Z|X) = -\sum_x p(x) \cdot \sum_z p(Z = z|X = x) \cdot \log(Z = z|X = x)$$

Since $Z = X + Y$, then $Y = Z - X$ and $p(Z = z|X = x) = p(Y = z - x|X = x)$. Hence, $H(Z|X) = -\sum_x p(x) \cdot \sum_y p(Y = z - x|X = x) \cdot \log(Y = z - x|X = x)$ (note the mistake in Cover's solution manual, page 27).

This can be rewritten as a simple conditional entropy, $H(Z|X) = \sum p(x) \cdot H(Y|X = x) = H(Y|X)$.

Further, if X and Y are independent, then $H(Y) = H(Y|X) = H(Z|X)$.

Problem No. 3: A Deck of Cards

(a) A deck of cards contains 52 cards, evenly distributed between hearts, diamonds, spades, and clubs. Which has higher entropy: drawing two red cards with or without replacement.

Intuitively, with replacement has a higher entropy. If we keep our first card, there are fewer possible events in the equally likely probability model for the second draw, and therefore a lower entropy. A more mathematically rigorous explanation can be expressed by looking at both cases.

Case 1) with replacement. Each draw is an independent and you are equally likely to draw any of the 52 cards. So, if A is the event of drawing two red cards (and the complement $\bar{A}$ is the probability of not drawing two red cards), and since the probability in any full deck of drawing a red card is $\frac{26}{52} = \frac{1}{2}$, then $p_1(A) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ and $p_1(\bar{A}) = 1 - p_1(A) = \frac{3}{4}$. Since we know the entire pdf, we can directly compute the entropy as:

$$H(p_1) = -\sum_{p_1} p_1 \cdot \log p_1 = p_1(A) \cdot \log \frac{1}{p_1(A)} + p_1(\bar{A}) \cdot \log \frac{1}{p_1(\bar{A})}$$

$$= \frac{1}{4} \cdot \log 4 + \frac{3}{4} \cdot \log \frac{4}{3} = 0.8113$$

Case 2) without replacement. The first draw is an equally likely event, identical to the first draw in case 1. The second draw, however, depends on the first draw. If we drew a red card on the first draw, then there are only 25 red cards left for the second draw. So, $p_2(A) = \frac{26}{52} \cdot \frac{25}{51} = 0.245$, since it is pointless to consider the case where we draw a black card on the first draw since that violates the conditions of the event. Again, $p_2(\bar{A}) = 1 - p_2(A) = 0.755$. Again, we can directly compute the entropy as:

$$H(p_2) = -\sum_{p_1} p_2 \cdot \log p_2 = p_2(A) \cdot \log \frac{1}{p_2(A)} + p_2(\bar{A}) \cdot \log \frac{1}{p_2(\bar{A})} \quad .$$

$$= -0.245 \cdot \log 0.245 + -0.755 \cdot \log 0.755 = 0.8034$$

Since $H(p_1) > H(p_2)$, with replacement has a higher entropy and my intuition has been verified.

I also analyzed this empirically as a quick proof of concept for further use of the software:

```
n_games = 1000000, w replacement =    0.25012, H =    0.81147
n_games = 1000000, w out replace =    0.24494, H =    0.80316
```

These results are very close to the theoretical values, but how close? This is beyond the scope of the problem, but needless to say empirical results on this easily modeled problem should be helpful for statistical evaluation of the program, such as statistical significance testing.

(b) Suppose the first card drawn is a king. What are the chances that the next card drawn will result in a sum under 21. Prove that you are better off knowing that the first card drawn was a king. I want you to show, using entropy, that you are better off knowing the value of a card (as opposed to not knowing it). In other words, knowing the value of the card, the entropy must be less for any given sum.

Since there are three problems about BlackJack, let's start with some fundamental rules and assumptions. BlackJack is a game in which players draw cards in an attempt to get the highest score under 21. Number cards are worth their cardinal values, face cards are worth 10 points, and the ace is worth either 1 or 11 points. Further, drawing 5 cards which sum to less than 22 points is an assured win. Since all of these problems deal with high values and a low number of cards, the rule of 5 can be safely ignored. To simplify the game somewhat for the purposes of this discussion, the Ace will always be counted as 11 points. These simplifications are acceptable because the point of these exercises is to show trends rather than actual values. We aren't going to try to make money off BlackJack anyway.

Let $X_1, X_2, \ldots, X_n$ be random variables representing the value in points of each card drawn. The first card is drawn from the distribution $p(X)$ given in the table on the right. The entropy of the first card drawn is simple,

$$H(X_1) = -\left[9 \cdot \left(\frac{1}{13} \cdot \log\frac{1}{13}\right) + \frac{4}{13} \cdot \log\frac{4}{13}\right] = 3.085 bits$$

The entropy of the first card drawn is independent of the number of decks used, as the numerator and denominator of each term are multiplied by the same number. Empirically,

```
n_games = 100000, H(draw1) =    3.08419
```

If we wanted to draw our second card without looking at the first, the uncertainty for the second card is the same as that for the first, another $3.085 bits$. So, playing blind the entropy for two cards is just the sum of two independent entropies, or $6.17 bits$.

Table 1: pdf of independent draws

X	$p(X)$
2	4/52 = 1/13
3	1/13
4	1/13
5	1/13
6	1/13
7	1/13
8	1/13
9	1/13
10	(10, J, Q, K) * 1/13 = 4/13
11	1/13

However, if we know that the first card drawn was a King, the second card is no longer an independent event. Also, the number of decks now plays a factor in the distribution — the one card removed from possibilities prevents this number from acting as a multiplicative identity. For n decks, the pdf can is computed much the same way except each $Pr\{X \neq 10\}$ is slightly higher and $Pr\{X = 10\}$ is slightly lower. Specifically $Pr\{X_2 \neq 10\} = \frac{4 \cdot n}{52 \cdot n - 1}$ and $Pr\{X_2 = 10\} = \frac{4 \cdot 4 \cdot n - 1}{52 \cdot n - 1}$. The entropy of this distribution is

$$H(X_2) = -\left[9 \cdot \frac{4 \cdot n}{52 \cdot n - 1} \cdot \log\frac{4 \cdot n}{52 \cdot n - 1} + \frac{16 \cdot n - 1}{52 \cdot n - 1} \cdot \log\frac{16 \cdot n - 1}{52 \cdot n - 1}\right].$$

For a single deck,

$$H(X_2)\Big|_{n=1} = -\left[9 \cdot \frac{4}{51} \cdot \log\frac{4}{51} + \frac{15}{51} \cdot \log\frac{15}{51}\right] = 3.11 bits$$

This is a very interesting result, which at first glance is non intuitive. What happened is that we started with a non-uniform distribution (even 1/13 probabilities for most values, one spike of 4/13 at 10 points). Knowing that the first card is a King, we reduce this spike and cause the distribution to be more uniform. Since maximum entropy exists at a uniform distribution, evening out the distribution increases the entropy.

Another way to look at this is the Duncan guessing game (how's that for gratuitous). What is the probability that you can correctly guess the value in points of a card, without replacement? Given a full deck, you would place your bet on the most likely outcome, 10, with a 4/13 probability. If this card was a 10 point card, then for the next draw you are at a disadvantage — the most likely outcome is less likely than it used to be, hence the entropy is greater. If the card was not a 10 point card, then the most likely choice is even more likely, hence the entropy is lessened. While this is a very interesting result **(which is worth another point for Rick)**, it is a long way from where we need to be in solving this problem. Now back to playing with a full deck...

A much simpler solution is to consider a binary distribution. In drawing a single card, let A be 1 if the first card drawn is greater than or equal to 10, 0 otherwise. The 10, J, Q, K, or Ace are the only cards worth 10 or more points, so

$$A = \begin{cases} 0 & ,p(A=0) = 8/13 \\ 1 & ,p(A=1) = 5/13 \end{cases}, \text{ and } H(A) = H(8/13, 5/13) = 0.9612 bits.$$

What is the probability of two cards resulting in a sum under 21? It is easier to look at what cards yield a sum larger than 20, mainly (10 J Q K, A), (A, 10 J Q K), or (A, A). If we let B be 1 if the sum is greater than 20, 0 otherwise, then

$$\begin{cases} \bar{B} & ,p(\bar{B}) = 1 - P(B) \\ B & ,p(B) = \frac{4}{13} \cdot \frac{4}{51} + \frac{1}{13} \cdot \frac{16}{51} + \frac{1}{13} \cdot \frac{3}{51} \end{cases}. \text{Further,} H(B) = H(0.05279, 0.9472) = 0.2981 bits.$$

What does this number mean? I think it means that the uncertainty of you NOT getting a sum over 20 is very small, meaning you are nearly assured to get a sum under 21 points in drawing two cards.

If we know that the first card drawn is a King, however, we can do two things to $p(B)$. First, we know that only the first case is valid, so the second and third terms in the summation are dropped.

Secondly, we don't have to multiply by the probability of the first card being worth 10 points (this is 1, it is given). Hence, $p(B) = 4/51$ and
$H(B|\text{first card King}) = H(0.0784, 0.9215) = 0.3966 bits$.

Again, this number is the uncertainty that you WON'T get your 21 points, the more uncertain this negative event the more likely we are to get 21 points. So, knowing that the first card is a king means that we are less certain we won't meet our goal.

An alternate example may help to illustrate this line of thinking. Given my skills in basketball, if I were to make the statement "I will make 10 free-throws in a row," you could be fairly certain I would fail. The closer you are to certainty, the lower the entropy. If I make the first basket, you would still probably not doubt my eventual failure. However, if I happened make 5 baskets in a row, your certainty in my athletic incompetency will be lessened, thereby increasing the entropy. You might start to think that I actually had some chance of making 10 baskets.

When the odds are stacked against you, you want as much entropy as possible. At the risk of getting philosophical (I've been working on this problem a LONG time), I'll hazard the statement *entropy bounds luck*.

This philosophy is all well and good, but it turns out that I think I had just misinterpreted the problem to begin with. A MUCH simpler solution follows:

Let X be a random variable equal to the value in points of the first card drawn. Let Y be a random variable equal to the value in points of the second card drawn. $Z = X + Y$ is your score in points for the two draws. As I proved earlier, $H(X) = 3.085 bits$. There are two reasons X and Y are not independent; the two cards are drawn without replacement (see solution, 3a), and the value of the Ace may change given the value of the second drawn card. If we tried to look at $H(Y)$ without any knowledge of X, we would again arrive at $3.085 bits$ (it doesn't matter to the pdf if the card is sitting in the deck or on the table, we still don't know what it is). Mathematically, $H(Z)$ and $H(Y)$ get very complicated, but they can easily be computed through empirical means.

First, let's verify $H(X)$. Over 1 million iterations, we obtain:

```
            p[1] =    0.00000
            p[2] =    0.07690
            p[3] =    0.07670
            p[4] =    0.07709
            p[5] =    0.07701
            p[6] =    0.07691
            p[7] =    0.07681
            p[8] =    0.07684
            p[9] =    0.07694
            p[10] =    0.30746
            p[11] =    0.07735
    n_games = 1000000, H(draw1) =    3.08552
```

Now, let's try $H(Z)$. Drawing two cards without replacement, the points will be distributed by:

```
        p[1] =    0.00000
        p[2] =    0.00000
        p[3] =    0.00000
        p[4] =    0.00449
        p[5] =    0.01200
        p[6] =    0.01652
        p[7] =    0.02404
        p[8] =    0.02883
        p[9] =    0.03614
        p[10] =    0.04051
        p[11] =    0.04850
        p[12] =    0.09372
        p[13] =    0.09638
        p[14] =    0.08880
        p[15] =    0.08468
        p[16] =    0.07713
        p[17] =    0.07204
        p[18] =    0.06508
        p[19] =    0.06034
        p[20] =    0.10218
        p[21] =    0.04862
n_games = 1000000, H(draw2) =    3.92347
```

If we constrain Z by knowing the king, $H(Z|X = 10)$ is found to be:

```
        p[1,..,10] =    0.00000
        p[11] =    0.00000
        p[12] =    0.07831
        p[13] =    0.07833
        p[14] =    0.07853
        p[15] =    0.07846
        p[16] =    0.07820
        p[17] =    0.07843
        p[18] =    0.07843
        p[19] =    0.07848
        p[20] =    0.29427
        p[21] =    0.07855
n_games = 1000000, H(draw1,K) =    3.11128
```

So, knowing the first card drawn is a king lowers the entropy of drawing two cards. The Duncan guessing game, Rick's free throws, and luck all amounted to a misinterpretation. I was comparing the entropy of one draw (X) to the entropy of two draws $H(Z|X = 10)$. I wanted to state that since the entropy of the first draw is 0 when it is certain (we know it's a King), then the entropy of the combination would be just the sum of the 0 entropy and an independent distribution. The reason this doesn't work is that the two events are not independent!

(c) You are playing blackjack with a dealer who is using one or two decks of cards. The player to your right has a king exposed on the table, and tells you he has reached a score of 20. Can you use information learned in this class to determine what your safest betting strategy might be? Or is this simply a straightforward probability calculation?

Assuming the player on my right is wearing some mark of clergy and I wish to trust him, I can use this information to my advantage. Since one player has 20 points, the game is simplified—either get 21 points, draw 5 cards under 22 points, or lose. The chances of beating a score of 20 are pretty slim, so my best bet is to keep my money until the next hand.

At the start of any new game (without such rules as dealer wins ties, etc.), both you and your opponent have an equal chance of winning the game. Assuming 3 players (yourself, the player on your right, and the dealer), the entropy of this distribution is $H(1/3, 1/3, 1/3) = 1.58 bits$. Since professional dealers always hit on 16 and stay at 17, it can probably be inferred that considerable analysis has been performed and staying at 17 maximizes a players chances at winning (or at least tying the winning score, which causes the house to win). So, a score of 20 is far above the average, and the player is very likely to win. Therefore, the entropy is reduced since the player on the right is now favored by probability, and I should keep my money until the next hand.

How do the multiple decks factor into this? If it was possible to use the information that a king was on the table with one deck to my advantage, then the dealer using two decks diminishes this usefulness. We are drawing cards without replacement. Each card drawn reduces the sample space, therefore making it easier to guess which card will be drawn next (though not necessarily the value in points of the next card, as shown in 3(b)). The extreme of this is to draw cards from an infinite number of decks. Knowing what cards are not in the deck (what cards are already drawn) does not help at all, since the pdf is equivalent to that of drawing with replacement.

This experiment is setup by having BJplay count occurrences of the dealer with 20 points, with and without a king as her first card. Any time the dealer has 20 points the outcome is bleak for the gambler. Without knowing what cards she has (only that she has 20 points), you end up with:

```
n_deck = 1, n_games = 126126, rate(D20) =   -0.67295
n_deck = 2, n_games = 127758, rate(D20) =   -0.67552
n_deck = 4, n_games = 128643, rate(D20) =   -0.67321
```

We actually seem to be a little better off when the dealer has a King, but the reason is not what you might expect. The standard table we follow for hit/stay/double is based on both the player's point total and the dealer's upcard. In this case, we force the dealer's first card to be a King, so the player moves more assertively.

```
n_deck = 1, n_games = 315587, rate(D20),K =   -0.58499
n_deck = 2, n_games = 322269, rate(D20),K =   -0.57783
n_deck = 4, n_games = 326524, rate(D20),K =   -0.57172
```

The problem with these numbers is that the player is not fully using the information given. These are just the statistics of what happens when the dealer has 20 points, but the player is still using the standard table to decide when to be hit or not. If we adjust the table to always hit when less than 20, we get a better distribution for our hero. Specifically,

```
n_deck = 1, n_games = 254719, rate(DS20) =    0.32173
n_deck = 2, n_games = 259582, rate(DS20) =    0.31261
n_deck = 4, n_games = 263189, rate(DS20) =    0.30886
```

This is such a significant swing it leads me to believe there must be a bug in the software, or we are hitting some sort of repetition in the random number generator. Given that the dealer's first card is a king, we get:

```
n_deck = 1, n_games = 303039, rate(DS20),K =   -0.40814
n_deck = 2, n_games = 310175, rate(DS20),K =   -0.40882
n_deck = 4, n_games = 314528, rate(DS20),K =   -0.40803
```

These are much more reasonable numbers. These numbers are hard to collect given the current software framework because I have to run many more experiments than what I collect data for. I run the software as normal, but only count instances where the scores are what we are interested in. Forcing the first card to be a king, 1 million iterations give us a sufficient amount of samples. With a completely open system, though, we needed to run 5 million iterations. Modifications to the simulator framework would better facilitate this collection, but are beyond the scope of this test problem.

(d) Suppose that two cards have been drawn by yourself and another player. The two cards you are holding sum to 17. Prove that if I tell you the other player's first card was a king, you are better able to make an intelligent bet (meaning you can better predict whether your sum will be greater than the other player's sum). Try to do this using entropy or other such concepts.

The table below shows a general BlackJack betting strategy. The columns represent what card the dealer has showing (in points), the rows show what your current score is. The three choices are to hit, stay, or double (H,S,D, respectively). To hit is to draw a card, to stay is to not draw a card, and to double is to double your wager and draw one last card. This strategy will lead you to lose 1.7 percent of the time. This may not seem that good, but any modifications only lead to a worse rate. For example, if we change all the "double" entries to simple "hits," the winning percentage will drop to -2.1%. Instructing the player to always hit on a 12 will cost another few tenths of a percentile. I found this table on the Yahoo games website, it would be interesting to try to generate such a table myself. Incidental, my login at the Yahoo games website, following the optimal probability model, is 15 thousand imaginary cyberdollars in the hole, so use extreme caution :)

your points	Dealer's upcard									
	2	3	4	5	6	7	8	9	10	A
17-21	S	S	S	S	S	S	S	S	S	S
13-16	S	S	S	S	S	H	H	H	H	H
12	H	H	S	S	S	H	H	H	H	H
11	D	D	D	D	D	D	D	D	D	H
10	D	D	D	D	D	D	D	D	H	H
9	H	D	D	D	D	H	H	H	H	H
4-11	H	H	H	H	H	H	H	H	H	H

Table 1: A General BlackJack Player's Strategy

If the cards you hold sum to 17, every BlackJack statistic sides with staying. So, What are the odds of winning a game when you have 17 points and you stay? What about 18, 19, or 20 points? For this experiment, the simulator was modified slightly to tag when the player's score was equal to X points, building the histograms only from those values. The output is as follows:

```
n_deck = 4, n_games = 119920, rate(17) =   -0.28047
n_deck = 4, n_games = 114437, rate(18) =    0.01620
n_deck = 4, n_games = 109002, rate(19) =    0.31699
n_deck = 4, n_games = 153436, rate(20) =    0.65805
```

So, if I have 17 showing, my chances aren't that good. Too bad I have already put down my money at that point.

Now I run the experiment again, this time giving the dealer the King as her first card. What are my chances this time around?

```
n_deck = 4, n_games = 136204, rate(17),K =   -0.43637
n_deck = 4, n_games = 129279, rate(18),K =   -0.19755
n_deck = 4, n_games = 123448, rate(19),K =    0.04233
n_deck = 4, n_games = 167221, rate(20),K =    0.49858
```

Even worse! Once again, too bad I already have my money down on this game. All numbers above use 4 decks. There is little difference observed when the number of decks is varied:

```
n_deck = 1, n_games = 124844, rate(17) =   -0.29843
n_deck = 2, n_games = 123846, rate(17) =   -0.30012
n_deck = 4, n_games = 119920, rate(17) =   -0.28047

n_deck = 1, n_games = 138082, rate(17),K =   -0.43262
n_deck = 2, n_games = 136208, rate(17),K =   -0.43635
n_deck = 4, n_games = 136204, rate(17),K =   -0.43637
```

In theory, though, as the number of decks increases the knowledge that the dealer’s first card is a king should less affect the probability.

Supplemental points:

```
Date: Thu, 25 Feb 1999 15:33:32 -0600 (CST)
X-Authentication-Warning: isip23.isip.msstate.edu: picone set sender to
picone@isip.msstate.edu using -f
From: Joe Picone - The Terminal Man <picone@ISIP.MsState.EDU>
To: duncan@isip.msstate.edu
CC: team_leaders@ISIP00.ISIP.MsState.Edu
In-reply-to: <199902252131.PAA25717@isip24.isip.msstate.edu>
(duncan@ISIP.MsState.EDU)
Subject: Re: mail command
Reply-to: picone@ISIP.MsState.EDU
Content-Type: text
Content-Length: 126

> > But there must be a better way.
>
> Yeah, use a prefix code!
>

That got you one more point on the first exam :)

-Joe
```

```
Date: Sun, 28 Feb 1999 12:28:14 -0600 (CST)
X-Authentication-Warning: isip23.isip.msstate.edu: picone set sender to
picone@isip.msstate.edu using -f
From: Joe Picone - The Terminal Man <picone@ISIP.MsState.EDU>
To: duncan@isip.msstate.edu
CC: hamaker@isip.msstate.edu, students@ISIP00.ISIP.MsState.Edu
In-reply-to: <199902281823.MAA06540@isip24.isip.msstate.edu>
(duncan@ISIP.MsState.EDU)
Subject: Re: fromLong, toDouble, etc.
Reply-to: picone@ISIP.MsState.EDU
Content-Type: text
Content-Length: 636

> I consider this to be an issue of invertability and/or mutual
> information. If you can fully describe Graph2d by these arguments and
>
>  I(Graph2d; Graph2dParameters) = H(Graph2dParameters) = H(Graph2d)

I think you get yet another point on the first exam.

> and have x1 == x2 (nominal formatting aside). If it isn’t this
> complete, you can’t use the name get() or assign(). Move to
> getSomething() or setSomething(). Notice we don’t use
> assignParameters(), since assign implies completeness.

At least you didn’t use the term NP complete :)

I believe what is missing is the notion of datasets within the Graph object.

-Joe
```