1A.

```
x = c(67,73, 64,77 ,71, 89, 72, 77, 80, 78)

x
 [1] 67 73 64 77 71 89 72 77 80 78
> meanX = sum(x) / length(x)
> varX = sum((x - meanX)^2) / length(x) -1
> varX
[1] 44.16

> sdX = sqrt(varX)
> sdX
[1] 6.645299

> var(x)
[1] 50.17778
> sdX
[1] 6.645299
```

1B.

```
Y = c(1.5, 2, 1, 3, 3, 9, 3, 2, 6, 8)

> numerr = sum((x - mean(x)) * (Y - mean(Y)))
> denomi = sqrt(sum((x - mean(x))^2) * sum((Y - mean(Y))^2))
> xpearson = numerr / denomi
> xpearson
[1] 0.8358709

> rankx = rank(x)
> rankY = rank(Y)
> numerr = sum((rankx - mean(rankx)) * (rankY - mean(rankY)))
> denomi = sqrt(sum((rankx - mean(rankx))^2) * sum((rankY - mean(rankY))^2))
> spearman = numerr / denomi
> spearman
[1] 0.8364835
```

2A.

```
> prob = 0.06
> q = 1 - prob
> probOneOrMore = 1 - q^n
> ceiling(log(1 - 0.85) / log(1 - probOneOrMore))
[1] 31



> p1 = 0.05
> n1 = 50
```

2B.

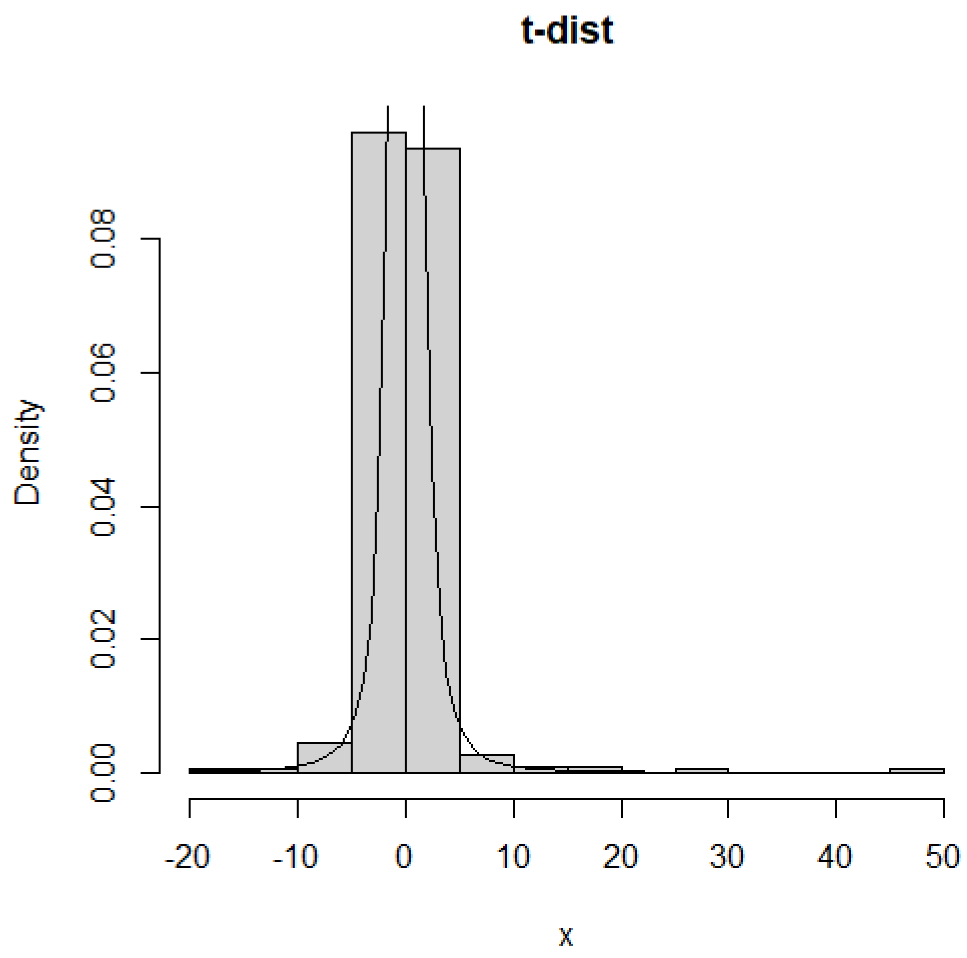
```
> prob01 = (1 - p1)^n1
> prob11 = n1 * p1 * (1 - p1)^(n1-1)
> probAtLeast21 = 1 - (prob01 + prob11)
>
> p2 = 0.25
> n2 = 10
> prob02 = (1 - p2)^n2
> prob12 = n2 * p2 * (1 - p2)^(n2-1)
> probAtLeast22 = 1 - (prob02 + prob12)
> probAtLeast21
[1] 0.7205682
> probAtLeast22
[1] 0.7559748
```

So the second option would be the better choice.

3A

I had a bunch of warnings here so I may have messed up but I believe my logic is correct, I forgot how the replicate thingy worked but I think this works as well.

```
> x = numeric(1000)
> for (i in 1:1000) {
+    y = rnorm(2)
+    x[i] = y[1]/sqrt(y[2]/2)
+ }
There were 50 or more warnings (use warnings() to see the first 50)
>
> hist(x, freq = FALSE, main = "t-dist")
> curve(dt(x, df = 2), add = TRUE, col = "black")
>
```



3B

```
> pt(1, dff = 2) - pt(-1, df = 2)
Error in pt(1, dff = 2) : unused argument (dff = 2)
> pt(1, df = 2) - pt(-1, df = 2)
[1] 0.5773503
```

4A.

I'm bad at these so forgive me

Win the third set, then win either the fourth or 5th set.
Win the fourth set, then win the 5th set.

Win the 5th set.

Are the possibilities for winning, i think?

Prob for Not winning the rest of the sets is (1-p)^2, so the probability of winning at least one of the sets is 1 - (1-p)^2. That means the probability of scenario 1 is p(1 - (1-p)^2).

Prob of scenario two is the probability of winning the fourth set P times the probability of winning the 5th se tP. Therefore, the probability of scen two is p^2.

The probability of scenario 3 is the probability of winning the fifth set P so it should just be P

So, the probability that the player wins the match given that he/she won the first two sets is:

p(1 - (1-p)^2) + p^2 + p

Again I'm going off the stuff in class 3 so I highly doubt I'm correct as I was confused there.

4B.

Given set win chance = p

Assuming these are the possible outcomes

The player wins the next set and the match.

The player wins the next set but loses the match.

The player loses the next set and the match.

Prob of winning next set is P, prob of losing is 1- p

player loses 10% of the matches in which they won the first two sets.

probability of winning the match given that the player won the first two sets is:

P(win match | win fir 2 sets) =
P(win 3 sets and the match | win first 2 sets) +
P(win 3 sets but lose the match | win first 2 sets)

Like 90% sure i gotta do these seperatley

First = $p * p * p$

Second = $p * p * (1-p) * 0.1$

So

$p * p * p + p * p * (1-p) * 0.1$

Then set it to 1/2

$p * p * p + p * p * (1-p) * 0.1 = 1/2$

$10p^3 + p^2 - p^3 = 5$

$9p^3 + p^2 - 5 = 0$

Which should be around 0.7866

5A.

```
> heights = c(158, 176, 177, 178, 184, 184, 187, 187)

> len = length(heights)

> meanHeight = mean(heights)

> stdError = sd(heights) / sqrt(len)

> tCrit = qt(0.9, df = len- 1)

> lower = meanHeight - tCrit * stdError

> lower

[1] 174.1184

> upper
```

```
[1] 183.6316
```

So 174.1184,183.6316

5B.

```
> heights = c(158, 176, 177, 178, 184)
> len = length(heights)
> meanHeight = mean(heights)
> stdError = sd(heights) / sqrt(n)
> tCrit = qt(0.8, df = len - 1)
> lower = meanHeight - tCrit * stdError
> upper = meanHeight + tCrit * stdError
> lower
[1] 171.3438
> upper
[1] 177.8562
```

So 171.3438,177.8561

5C.

```
> heights = c(158, 176, 177, 178, 184)
> len = length(heights)
> meanHeight = mean(heights)
> stdError = sd(heights) / sqrt(n)
> tCrit = .60
> lower = meanHeight - tCrit * stdError
> upper = meanHeight + tCrit * stdError
> lower

[1] 172.5237

> upper

[1] 176.6763
```

So 172.5237,176.6753

Not sure about this one either but oh well.

By signing below, I assert that I did not receive live help from any human, artificial intelligence, or material found on the web to answer the questions on this exam.

Student Signature (typed): Kaden Bilyeu