```yaml
  Course: Statistical_Inference
  Lesson: Variance

- Class: text
  Output: "Variance. (Slides for this and other Data Science courses may be found at github https://github.com/DataScienceSpecialization/courses/. If you care to use them, they must be downloaded as a zip file and viewed locally. This lesson corresponds to 06_Statistical_Inference/05_Variance.)"

- Class: text
  Output: In this lesson, we'll discuss variances of distributions which, like means, are useful in characterizing them. While the mean characterizes the center of a distribution, the variance and its square root, the standard deviation, characterize the distribution's spread around the mean. As the sample mean estimates the population mean, so the sample variance estimates the population variance.

- Class: text
  Output: The variance of a random variable, as a measure of spread or dispersion,  is, like a mean, defined as an expected value. It is the expected squared distance of the variable from its mean. Squaring the distance makes it positive so values less than and greater than the mean are treated the same. In mathematical terms, if X comes from a population with mean mu, then

- Class: text
  Output: Var(X) = E( (X-mu)^2 ) = E( (X-E(X))^2 ) =  E(X^2)-E(X)^2

- Class: text
  Output: So variance is the difference between two expected values. Recall that E(X), the expected value of a random variable from the population, is mu, the mean of that population.

- Class: text
  Output: Higher variance implies more spread around a mean than lower variance.

- Class: text
  Output: Finally, it's easy to show from the definition and the linearity of expectations that, if a is a constant, then Var(aX)=a^2*Var(X). This will come in handy later.

#- Class: video
#  Output: Would you like to see the equation proving this? You'll need an internet connection to see it.
#  VideoLink: "http://wilcrofter.github.io/slidex/markDown/varAX.html"

- Class: figure
  Output: If you're interested, here's the proof. You might have to stretch out your plot window to make it clearer.
  Figure: plotVform.R
  FigureType: new

- Class: text
  Output:  Let's practice computing the variance of a dice roll now. First we need to compute E(X^2). From the definition of expected values, this means we'll take a weighted sum over all possible values of X^2. The weight is the probability of X occurring.

- Class: cmd_question
  Output: For convenience, we've defined a 6-long vector for you, dice_sqr, which holds the squares of the integers 1 through 6. This will give us the X^2 values. Look at it now.
  CorrectAnswer: dice_sqr
  AnswerTests: omnitest(correctExpr='dice_sqr')
  Hint: Type dice_sqr at the command prompt.

- Class: cmd_question
  Output: Now we need weights. For these we can use any of the three PDF's, (dice_fair, dice_high, and dice_low) we defined in the previous lesson. Using R's ability to multiply vectors componentwise and its function 'sum' we can easily compute E(X^2) for any of these dice. Simply sum the product dice_sqr * PDF.  Try this now with
```

```
    dice_fair and put the result in a variable ex2_fair.
    CorrectAnswer: ex2_fair <- sum(dice_fair * dice_sqr)
    AnswerTests: expr_creates_var('ex2_fair'); ANY_of_exprs('ex2_fair <- sum(dice_fair * dice_sqr)','ex2_fair <- sum(dice_sqr * dice_fair)')
    Hint: Type 'ex2_fair <- sum(dice_fair * dice_sqr)' at the command prompt.

  - Class: cmd_question
    Output: Recall that the expected value of a fair dice roll is 3.5. Subtract the square of that from ex2_fair to compute the sample variance.
    CorrectAnswer: ex2_fair-3.5^2
    AnswerTests: ANY_of_exprs('ex2_fair-3.5^2','ex2_fair-3.5*3.5','ex2_fair-(3.5^2)','ex2_fair-(3.5*3.5)')
    Hint: Type 'ex2_fair-3.5^2' at the command prompt.

  - Class: cmd_question
    Output: Now use a similar approach to compute the sample variance of dice_high in one step. Sum the appropriate product and subtract the square of the mean. Recall that edh holds the expected value of dice_high.
    CorrectAnswer: sum(dice_high * dice_sqr)-edh^2
    AnswerTests: ANY_of_exprs('sum(dice_high * dice_sqr)-edh^2','sum(dice_sqr * dice_high)-edh^2','sum(dice_high * dice_sqr)-edh*edh','sum(dice_sqr * dice_high)-edh*edh')
    Hint: Type 'sum(dice_high * dice_sqr)-edh^2' at the command prompt.

  - Class: text
    Output: Note that when we talk about variance we're using square units. Because it is often more useful to use measurements in the same units as X we define the standard deviation of X as the square root of Var(X).

  - Class: figure
    Output: Here's a figure from the slides. It shows several normal distributions all centered around a common mean 0, but with different standard deviations. As you can see from the color key on the right, the thinner the bell the smaller the standard deviation and the bigger the standard deviation the fatter the bell.
    Figure: normalVar.R
    FigureType: new

  - Class: text
    Output: Just as we distinguished between a population mean and a sample mean we have to distinguish between a population variance sigma^2 and a sample variance s^2. They are defined similarly but with a slight difference. The sample variance is defined as the sum of n squared distances from the sample mean divided by (n-1), where n is the number of samples or observations. We divide by n-1 because this is the number of degrees of freedom in the system. The first n-1 samples or observations are independent given the mean. The last one isn't independent since it can be calculated from the sample mean used in the formula.

  - Class: text
    Output:  In other words, the sample variance is ALMOST the average squared deviation from the sample mean.

  - Class: text
    Output: As with the sample mean, the sample variance is also a random variable with an associated population distribution. Its expected value or mean is the population variance and its distribution gets more concentrated around the population variance with more data. The sample standard deviation is the square root of the sample variance.

  - Class: figure
    Output: To illustrate this point, consider this figure which plots the distribution of 10000 variances, Each variance was computed on a sample of standard normals of size 10. The vertical line indicates the standard deviation 1.
    Figure: moreData1.R
    FigureType: new

  - Class: figure
    Output: Here we do the same experiment but this time (the taller lump) each of the 10000 variances is over 20 standard normal samples. We've plotted over the first plot
```

```
    (the shorter lump) and you can see that the distribution of the variances is getting tighter and shifting closer to the vertical line.
    Figure: moreData2.R
    FigureType: new

  - Class: figure
    Output: Finally, we repeat the experiment using 30 samples for each of the 10000 variances. You can see that with more data, the distribution gets more concentrated around the population variance it is trying to estimate.
    Figure: moreData3.R
    FigureType: new

  - Class: text
    Output: Now recall that the means of unbiased estimators equal the values they're trying to estimate. We can infer  from the above that the sample variance is an unbiased estimator of population variance.

  - Class: text
    Output: Recall that the average of random samples from a population is itself a random variable with a distribution centered around the population mean. Specifically, E(X') = mu, where X' represents a sample mean and mu is the population mean.

  - Class: text
    Output:  We can show that, if the population is infinite, the variance of the sample mean is the population variance divided by the sample size. Specifically,  Var(X') = sigma^2 / n. Let's work through this in four short steps.

  - Class: mult_question
    Output: Which of the following does Var(X') equal? Here X' represents the sample mean and 'Sum(X_i)' represents the sum of the n samples X_1,...X_n. Assume these samples are independent.
    AnswerChoices: Var(1/n * Sum(X_i)); E(1/n * Sum(X_i)); mu; sigma
    CorrectAnswer: Var(1/n * Sum(X_i))
    AnswerTests: omnitest(correctVal='Var(1/n * Sum(X_i))')
    Hint: Which of the choices has both Var and the definition of mean in it?

  - Class: mult_question
    Output: Which of the following does Var(1/n * Sum(X_i)) equal?
    AnswerChoices: 1/n^2*Var(Sum(X_i)); 1/n^2*E(Sum(X_i)); mu/n^2; sigma/n
    CorrectAnswer: 1/n^2*Var(Sum(X_i))
    AnswerTests: omnitest(correctVal='1/n^2*Var(Sum(X_i))')
    Hint: Remember that fact about Var that we said would be useful before? Now is the time to use it.

  - Class: mult_question
    Output: Recall that Var is an expected value and expected values are linear. Also recall that our samples X_1, X_2,...,X_n are independent. What does Var(Sum(X_i)) equal?
    AnswerChoices: Sum(Var(X_i)); E(Sum(X_i)); E(mu); Var(sigma)
    CorrectAnswer: Sum(Var(X_i))
    AnswerTests: omnitest(correctVal='Sum(Var(X_i))')
    Hint: By linearity, the variance of the sum equals the sum of the variance.

  - Class: mult_question
    Output: Finally, each X_i comes from a population with variance sigma^2. What does Sum(Var(X_i)) equal? As before, Sum is taken over n values.
    AnswerChoices: n*(sigma)^2; n*mu; n*E(mu); (n^2)*Var(sigma)
    CorrectAnswer: n*(sigma)^2
    AnswerTests: omnitest(correctVal='n*(sigma)^2')
    Hint: Var(X_i) is the constant value sigma^2 and we're summing over n of them.

  - Class: text
    Output: So we've shown that Var(X')=Var(1/n*Sum(X_i))=(1/n^2)*Var(Sum(X_i))=(1/n^2)*Sum(sigma^2)=sigma^2/n for infinite populations when our samples are independent.

  - Class: text
    Output: The standard deviation of a statistic is called its standard error, so the
```

```
  standard error of the sample mean is the square root of its variance.

- Class: text
  Output: We just showed that the variance of a sample mean is sigma^2 / n and we estimate it with s^2 / n. It follows that its square root, s / sqrt(n), is the standard error of the sample mean.

- Class: text
  Output: The sample standard deviation, s, tells us how variable the population is, and s/sqrt(n), the standard error, tells us how much averages of random samples of size n from the population vary. Let's see this with some simulations.

- Class: cmd_question
  Output: The R function rnorm(n,mean,sd) generates n independent (hence uncorrelated) random normal samples with the specified mean and standard deviation. The defaults for the latter are mean 0 and standard deviation 1. Type the expression sd(apply(matrix(rnorm(10000),1000),1,mean)) at the prompt.
  CorrectAnswer: sd(apply(matrix(rnorm(10000),1000),1,mean))
  AnswerTests: omnitest(correctExpr='sd(apply(matrix(rnorm(10000),1000),1,mean))')
  Hint: Type 'sd(apply(matrix(rnorm(10000),1000),1,mean))' at the command prompt.

- Class: cmd_question
  Output: This returns the standard deviation of 1000 averages, each of a sample of 10 random normal numbers with mean 0 and standard deviation 1. The theory tells us that the standard error, s/sqrt(n), of the sample means indicates how much averages of random samples of size n (in this case 10) vary. Now compute 1/sqrt(10) to see if it matches the standard deviation we just computed with our simulation.
  CorrectAnswer: 1/sqrt(10)
  AnswerTests: omnitest(correctExpr='1/sqrt(10)')
  Hint: Type '1/sqrt(10)' at the command prompt.

- Class: mult_question
  Output: Pretty close, right? Let's try a few more. Standard uniform distributions have variance 1/12. The theory tells us the standard error of means of independent samples of size n would have which standard error?
  AnswerChoices: 1/(12*sqrt(n)); 12/sqrt(n); 1/sqrt(12*n); I haven't a clue
  CorrectAnswer: 1/sqrt(12*n)
  AnswerTests: omnitest(correctVal='1/sqrt(12*n)')
  Hint: In this case s is the sqrt(1/12). Divide this by sqrt(n).

- Class: cmd_question
  Output:  Compute 1/sqrt(120). This would be the standard error of the means of uniform samples of size 10.
  CorrectAnswer: 1/sqrt(120)
  AnswerTests: omnitest(correctExpr='1/sqrt(120)')
  Hint: Type '1/sqrt(120)' at the command prompt.

- Class: cmd_question
  Output: Now check it as we did before. Use the expression sd(apply(matrix(runif(10000),1000),1,mean)).
  CorrectAnswer: sd(apply(matrix(runif(10000),1000),1,mean))
  AnswerTests: omnitest(correctExpr='sd(apply(matrix(runif(10000),1000),1,mean))')
  Hint: Type 'sd(apply(matrix(runif(10000),1000),1,mean))' at the command prompt.

- Class: mult_question
  Output: Pretty close again, right? Poisson(4) are distributions with variance 4; what standard error would means of random samples of n Poisson(4) have?
  AnswerChoices: 2/sqrt(n); 1/sqrt(2*n); 2*sqrt(n); I haven't a clue
  CorrectAnswer:  2/sqrt(n)
  AnswerTests: omnitest(correctVal='2/sqrt(n)')
  Hint: In this case s is 2. Divide this by sqrt(n).

- Class: cmd_question
  Output:  We'll do another simulation to test the theory. First, assume you're taking averages of 10 Poisson(4) samples and compute the standard error of these means. Use the formula you just chose.
  CorrectAnswer: 2/sqrt(10)
  AnswerTests: omnitest(correctExpr='2/sqrt(10)')
  Hint: Type '2/sqrt(10)' at the command prompt.
```

```
- Class: cmd_question
  Output: Now check it as we did before. Use the expression sd(apply(matrix(rpois(10000,4),1000),1,mean)).
  CorrectAnswer: sd(apply(matrix(rpois(10000,4),1000),1,mean))
  AnswerTests: omnitest(correctExpr='sd(apply(matrix(rpois(10000,4),1000),1,mean))')
  Hint: Type 'sd(apply(matrix(rpois(10000,4),1000),1,mean))' at the command prompt.

- Class: mult_question
  Output: Like magic, right? One final test. Fair coin flips have variance 0.25; means of random samples of n coin flips have  what standard error?
  AnswerChoices: 2/sqrt(n); 1/sqrt(2*n); 2*sqrt(n); 1/(2*sqrt(n)); I haven't a clue
  CorrectAnswer:  1/(2*sqrt(n))
  AnswerTests: omnitest(correctVal='1/(2*sqrt(n))')
  Hint: In this case s is 1/2 which is the sqrt of 1/4, the variance. Divide this by sqrt(n).

- Class: cmd_question
  Output:  You know the drill. Assume you're taking averages of 10 coin flips and compute the standard error of these means with the theoretical formula you just picked.
  CorrectAnswer: 1/(2*sqrt(10))
  AnswerTests: omnitest(correctExpr=' 1/(2*sqrt(10))')
  Hint: Type ' 1/(2*sqrt(10))' at the command prompt.

- Class: cmd_question
  Output: Now check it as we did before. Use the expression sd(apply(matrix(sample(0:1,10000,TRUE),1000),1,mean)).
  CorrectAnswer: sd(apply(matrix(sample(0:1,10000,TRUE),1000),1,mean))
  AnswerTests: omnitest(correctExpr='sd(apply(matrix(sample(0:1,10000,TRUE),1000),1,mean))')
  Hint: Type 'sd(apply(matrix(sample(0:1,10000,TRUE),1000),1,mean))' at the command prompt.

- Class: text
  Output: Finally, here's something interesting. Chebyshev's inequality helps interpret variances. It states that the probability that a random variable X is at least k standard deviations from its mean is less than 1/(k^2). In other words, the probability that X is at least 2 standard deviations from the mean is less than 1/4, 3 standard deviations 1/9, 4 standard deviations 1/16, etc.

- Class: text
  Output: However this estimate is quite conservative for random variables that are normally distributed, that is, with bell-curve distributions. In these cases, the probability of being at least 2 standard deviations from the mean is about 5% (as compared to Chebyshev's upper bound of 25%) and the probability of being at least 3 standard deviations from the mean is roughly .2%.

- Class: mult_question
  Output: Suppose you had a measurement that was 4 standard deviations from the distribution's mean. What would be the upper bound  of the probability of this happening using Chebyshev's inequality?
  AnswerChoices: 6%; 0%; 11%; 25%; 96%
  CorrectAnswer: 6%
  AnswerTests: omnitest(correctVal='6%')
  Hint: Chebyshev's inequality estimates that probability as 1/16. Convert this to a probability.


- Class: mult_question
  Output: Now to review. The sample variance estimates what?
  AnswerChoices: population variance; sample mean; sample standard deviation; population
  CorrectAnswer: population variance
  AnswerTests: omnitest(correctVal='population variance')
  Hint: Which choice has the word variance in it?

- Class: mult_question
  Output: The distribution of the sample variance is centered at what?
  AnswerChoices: population variance; sample mean; sample standard deviation; population
  CorrectAnswer: population variance
```

```yaml
  AnswerTests: omnitest(correctVal='population variance')
  Hint: What is the sample variance estimating?

- Class: mult_question
  Output: True or False - The sample variance gets more concentrated around the population variance with larger sample sizes
  AnswerChoices: True; False
  CorrectAnswer: True
  AnswerTests: omnitest(correctVal='True')
  Hint: Is more data better than less data?

- Class: mult_question
  Output: The variance of the sample mean is the population variance divided by ?
  AnswerChoices: n; n^2; sqrt(n); I haven't a clue
  CorrectAnswer: n
  AnswerTests: omnitest(correctVal='n')
  Hint: Remember the 4 step proof starting with Var(X')=...? The last step had an n divided by an n^2.

- Class: mult_question
  Output: The standard error of the sample mean is the sample standard deviation s divided by ?
  AnswerChoices: n; n^2; sqrt(n); I haven't a clue
  CorrectAnswer: sqrt(n)
  AnswerTests: omnitest(correctVal='sqrt(n)')
  Hint: Remember the many many examples we went through. The sqrt(n) figured prominently in them.

- Class: text
  Output: Congrats! You've concluded this vary long lesson on variance. We hope you liked it vary much.

- Class: mult_question
  Output: "Would you like to receive credit for completing this course on
    Coursera.org?"
  CorrectAnswer: NULL
  AnswerChoices: Yes;No
  AnswerTests: coursera_on_demand()
  Hint: ""
```