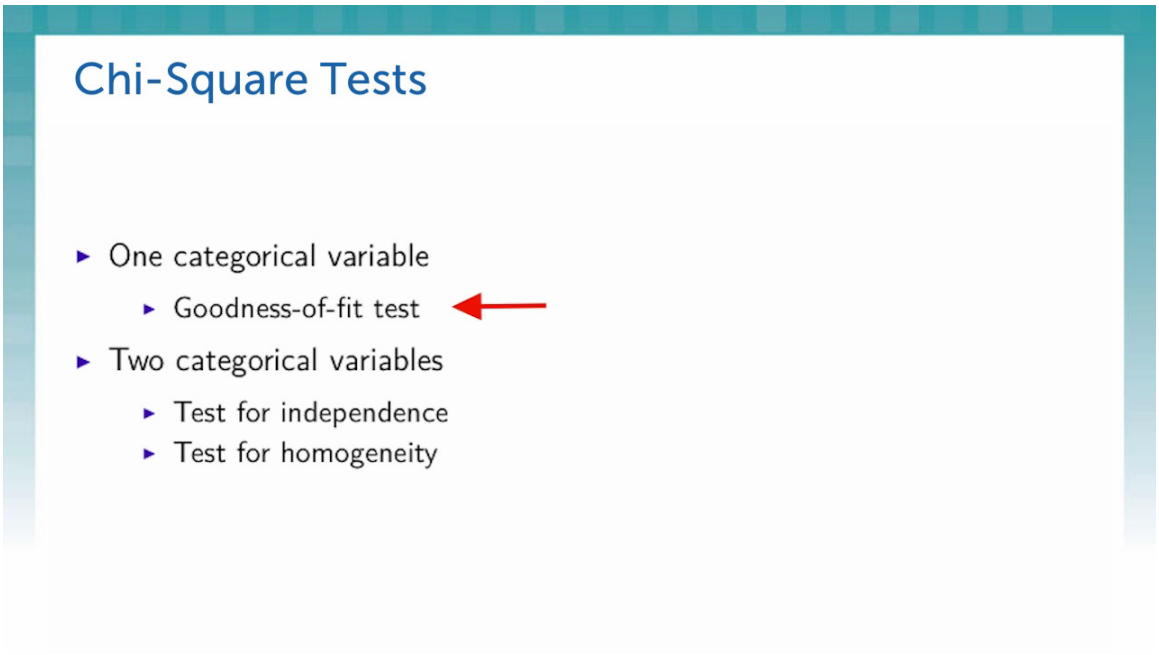
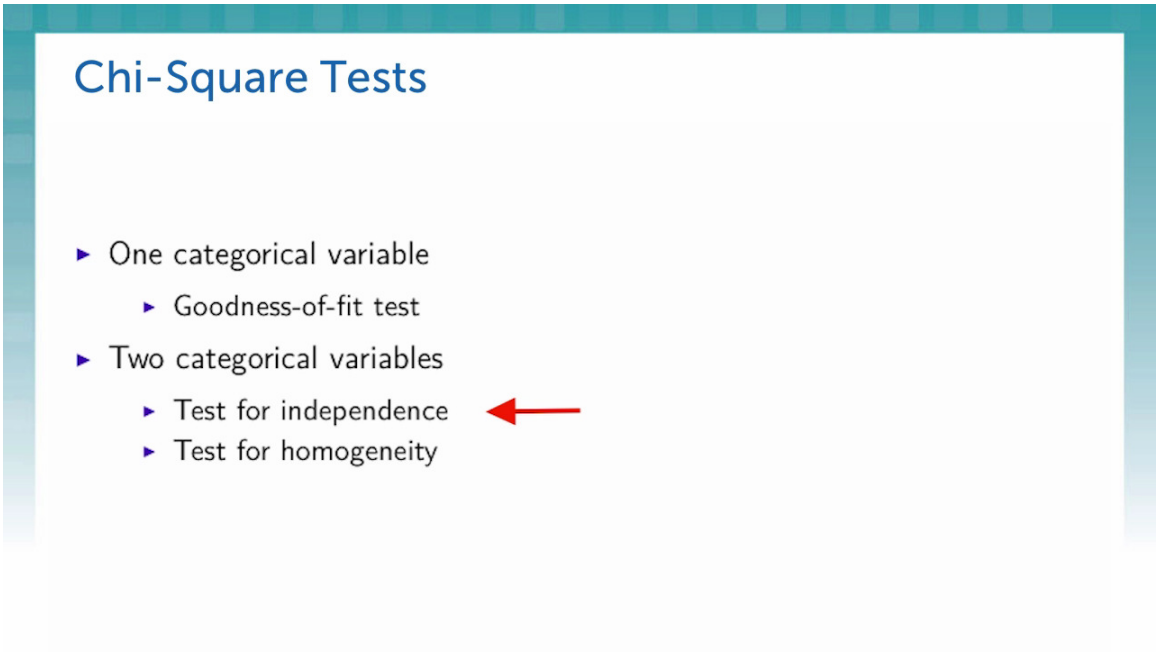
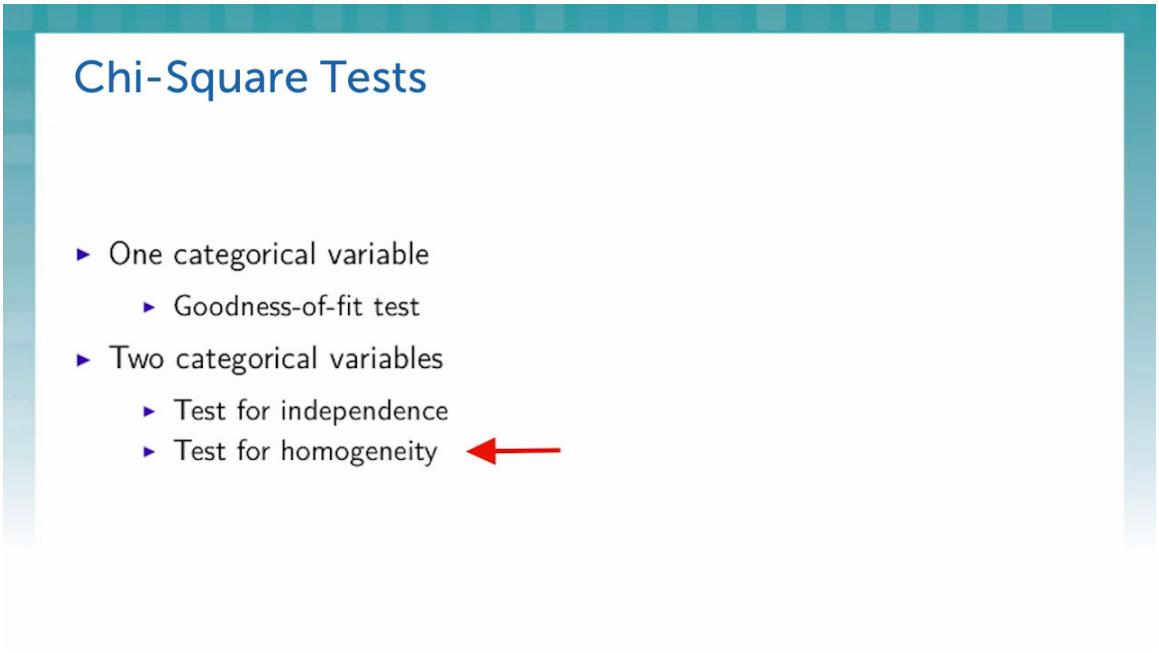

DS 705
Statistical Methods
Inference for Categorical Data – Part 2



When just one categorical variable is under consideration, the chi-square test for goodness of fit can be used to test the hypothesis that the sample was drawn from a specified distribution versus the alternative that it was not. You may recall that the Shapiro-Wilk test for normality is also a goodness-of-fit test.



For two categorical factors, the chi-square statistic can be used to test for the independence of the two factors versus the alternative that the factors are associated. With the test for independence, the sampling scheme must be that a random sample has been drawn from the population of interest, thus making the row and column totals random counts.



And the chi-square test for homogeneity has identical computations for the test statistic and p-value as for the test for independence but differs in the sampling scheme, which also affects how the hypotheses are set up and how to interpret the results.

For the test of homogeneity, independent samples are drawn from each sub-population defined by one categorical factor in order to compare the distributions of the other categorical variable. The null hypothesis would be that all of the distributions are the same against the alternative that they are not all the same.

Example A: Chi-square GOF Test

Suppose it is reported in a media release that 24% of all personal loans are for home mortgages, 38% were for automobile purchases, 18% were for credit card loans, and the rest were for other types of loans. Records for a random sample of 55 loans was obtained and each was classified into one of these categories. The results are in the following table.

	Mortgage	Auto	Credit	Other
Number of loans	24	21	6	4

GOF Test: The Request

Conduct the appropriate test to determine if the distribution reported in the media release for the frequency of the types of loans fits the actual distribution of types of loans in the population.

Use $\alpha = 0.01$

GOF Test: The Hypotheses

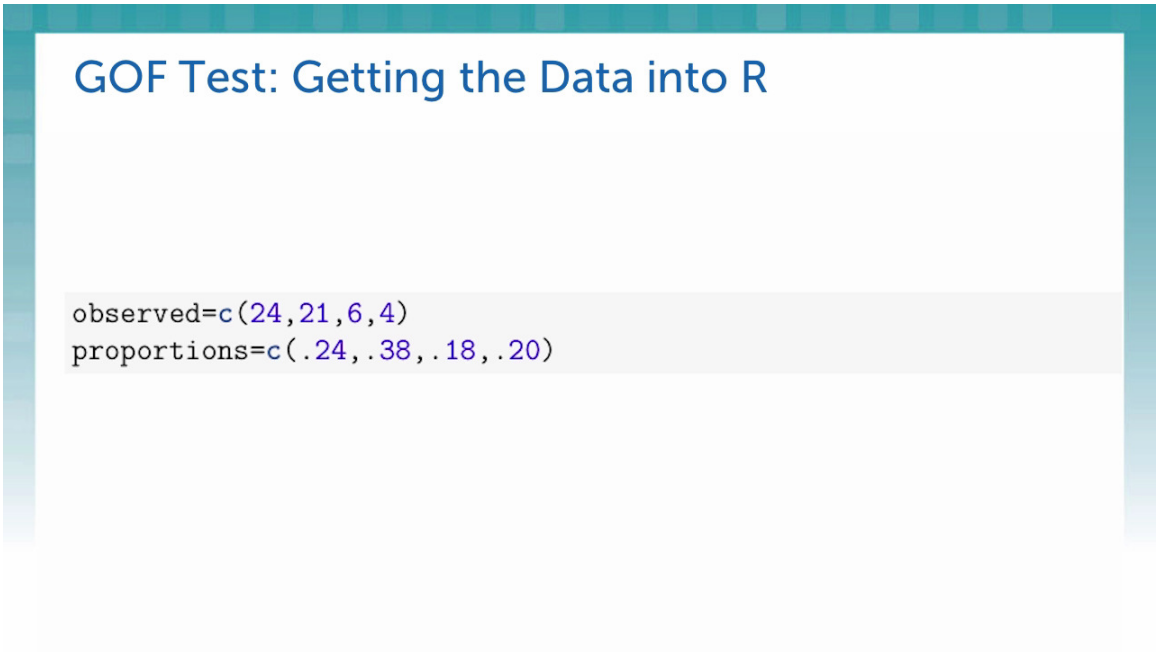
H_0: $\pi_{Mortgage} = 0.24, \pi_{Auto} = 0.38, \pi_{Credit} = 0.18, \pi_{Other} = 0.20$

H_a: At least one π_i differs from another

Verbally, the null hypothesis is claiming that the distribution claimed by the media release is correct. The alternative hypothesis is simply that the distribution is not correct since at least one of the hypothesized probabilities is not right.

Many times, the chi-square goodness-of-fit test is used to determine if the categories have equal probabilities, like testing to see if a die is fair, for example. In those cases, it isn't necessary to specify the proportions because they are self-evident.

If a six-sided die is equally balanced, then each outcome should have a probability of 1 our of 6, or 1/6. If we're testing to see if the proportions of loans are equally likely here, the null hypothesis probabilities would all be 1/4 since there are four categories.



We simply create a vector that contains the observed cell counts-- here I named it observed-- and another vector holding the hypothesized proportions, which I called proportions. You've probably noticed by now that we're using the terms "proportions" and "probabilities" interchangeably.

In this test, our presumption is that the underlying variable has a multinomial probability distribution with the probability specified by the null hypothesis. Multinomial distributions are characterized by having n identical independent trials each having k possible outcomes, where the probabilities of each of the k outcomes remains constant from trial to trial.

GOF Test: Getting the Test Statistic & P-value in R

```
chisq.test(x=observed,p=proportions)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 14.828, df = 3, p-value = 0.00197
```

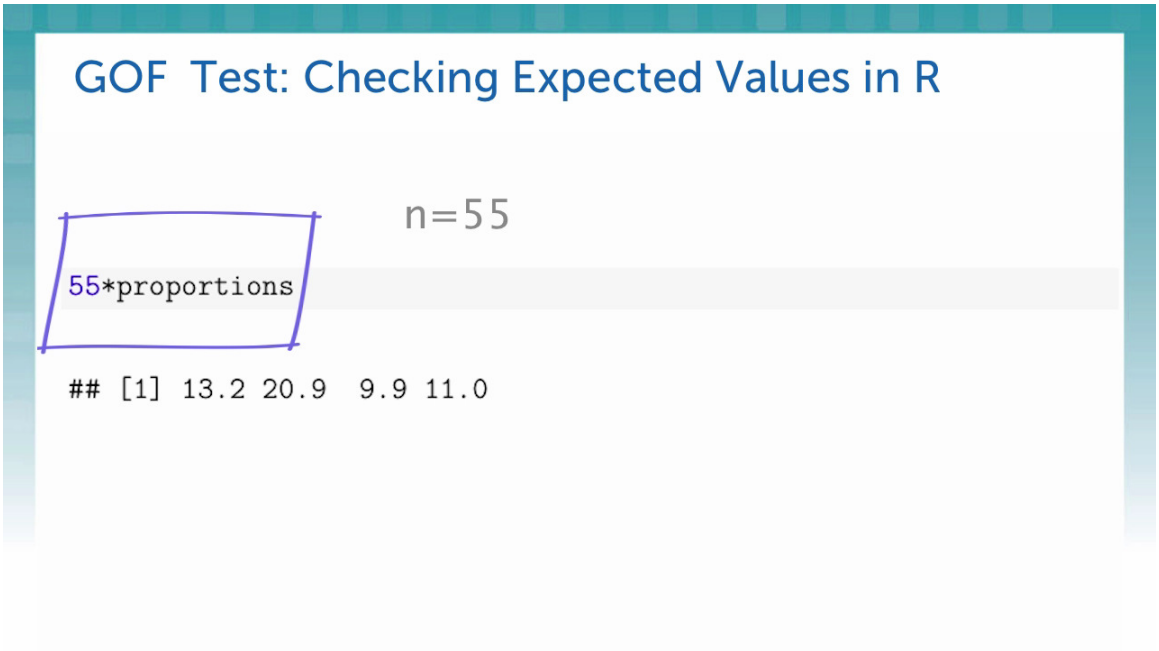
One quick check to see that we've coded it right is to look at the degrees of freedom. It should be the number of categories minus 1. Since there were four loan categories being tested and we see the degrees of freedom given as 3, we should start to get warm fuzzies about now.

GOF Test: Getting the Test Statistic & P-value in R

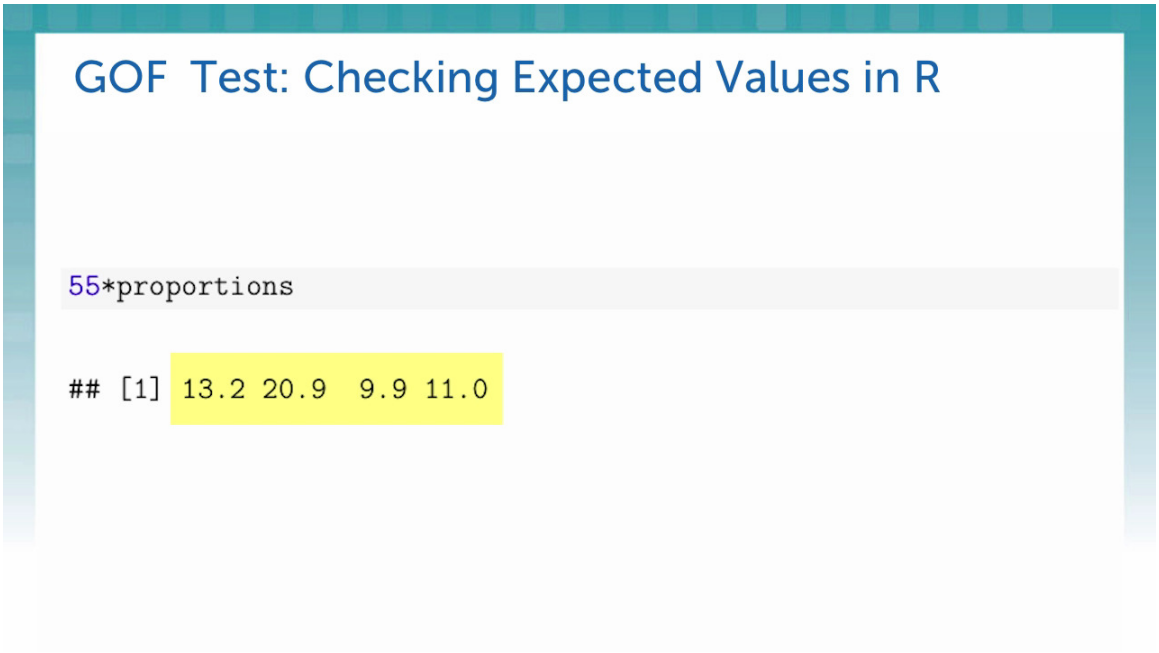
```
chisq.test(x=observed,p=proportions)


##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 14.828, df = 3, p-value = 0.00197
```

(The level of significance was 0.01)

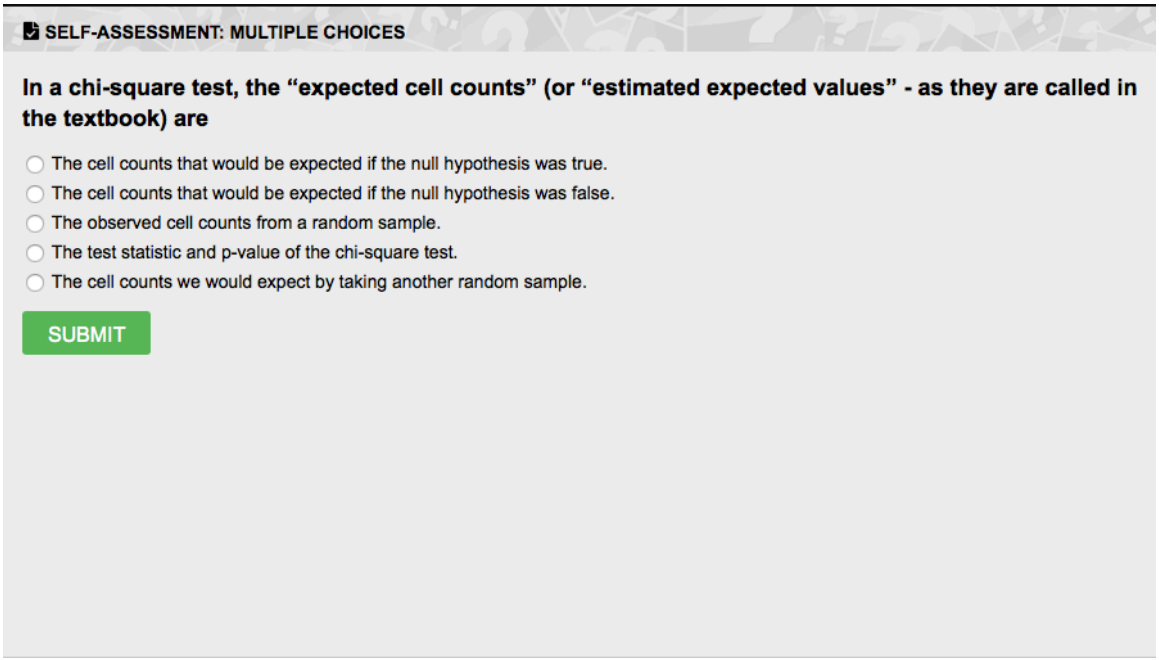
What should we conclude? Was the media report correct? No. According to the sample data resulting in a test statistic of 14.828 and a p-value of 0.00197, which is less than 0.01, we should reject the null hypothesis and claim that the actual distribution for the types of personal loans is different from what was reported.



With a smaller sample size like this one, it would behoove us to check the sample size requirement for this chi-square test. You see the expected cell frequencies are easily obtained by multiplying the vector of the hypothesized proportions by the sample size. Notice also that the requirement isn't that the observed cell counts are all at least 5, but that the expected counts are all at least 5.



So even though there was an observed cell frequency of 4 here, our sample was still large enough to trust the chi-square test for goodness of fit. At least we can trust it to the extent that we didn't make a type I error by rejecting the null hypothesis, which was controlled at the 1% level of significance.



Question 1

See correct answer at the end of the transcripts.

Example B: Health Exam Data

The Age Group and Region for the first 6 out of 80 subjects is as follows

```
##   AgeGroup    Region
## 1    36-65      West
## 2    36-65     South
## 3      65+   Midwest
## 4    36-65      West
## 5    36-65 Northeast
## 6      65+   Midwest
```

Instead of having the counts as basic summary statistics for our categorical variables, we may have a large data frame that contains the individual observations. That's OK. R will know just what to do with them, and they can be entered into the chisq.test function in the same way as the vectors or matrices containing the frequencies.

Example B: Health Exam Data

To see the crosstabs, use the 'table' function in R

```
table(AgeGroup,Region)
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35       6         9     5    8
##    36-65       4         7    13    8
##    65+         6         6     2    6
```

When the data comes as individual observations in a data frame, it is a good idea to just look at the counts to get a feel for what relationship might exist between the factors and to make sure there aren't any unexpected surprises in your data set.

Example B: Health Exam Data

Whether it is a test for independence or homogeneity, the R code is the same.

```
chisq.test(AgeGroup,Region,data=HealthExam)
```

The chisq.test function can be used with vectors or matrices containing the contingency table frequencies in the same way that was shown for the prop.test function previously in this presentation. However, when categorical data is listed out in a data frame, the variables can be loaded directly into the chisq.test function by their names in the data frame.

Example B: Health Exam Output from `chisq.test`

Since the 80 people selected in this study randomly fell into the age categories and geographic regions, the chi-square test here is for independence (not homogeneity).

```
##
##  Pearson's Chi-squared test
##
## data:  AgeGroup and Region
## X-squared = 8.188, df = 6, p-value = 0.2247
```

Chi-square Test for Health Exam Data

H_0: Age Group and Region are independent.
H_a: Age Group and Region are associated.

Conclusion: Do not reject H_0 at $\alpha = 0.05$. There is insufficient evidence in this sample to claim that Age Group and Region are associated for the population of U.S. adults ($P = 0.2247$).

You see the conclusion here is not to reject the null hypothesis. But wait. Some of those cell counts were pretty small. We should check the expected cell counts to see if any are under 5.

Expected Cell Counts for Health Exam Data

```
result=chisq.test(AgeGroup,Region)
result$expected
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35     5.6       7.7     7  7.7
##    36-65     6.4       8.8     8  8.8
##    65+       4.0       5.5     5  5.5
```

To get the expected cell counts, you see that it's necessary to assign the chisq.test output to an object in R and then call from that object the expected values using this code here—result$expected.

Expected Cell Counts for Health Exam Data

```
result=chisq.test(AgeGroup,Region)
result$expected
```

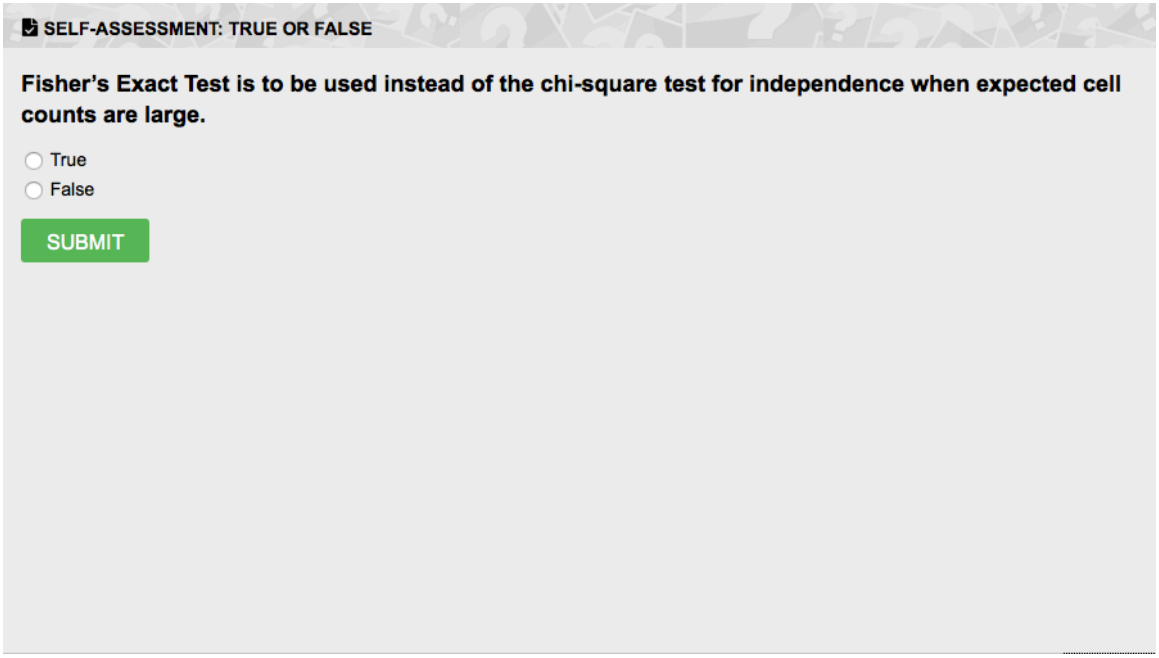
```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35     5.6       7.7     7  7.7
##    36-65     6.4       8.8     8  8.8
##    65+       4.0       5.5     5  5.5
```

Do you see the expected cell frequency for the 65-and-over age group in the Midwest region? It's 4. Well, it is only one cell count, and it is very close to 5. Even so, using the chi-square distribution for the test statistic may not be such a good approximation, even to the extent that we should a least look at another test, one that can handle small expected cell frequencies. Fisher's exact test is just the one. It can handle tables larger than 2-by-2. Let's see what it says about the Health Exam data set.

Fish Exact Test for Health Exam Data – More Than a 2x2 Table

```
fisher.test(AgeGroup,Region)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  AgeGroup and Region
## p-value = 0.2443
## alternative hypothesis: two.sided
```



Question 2

See correct answer at the end of the transcripts.

Row Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,1)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    21.4      32.1  17.9 28.6 → 100%
##    36-65    12.5      21.9  40.6 25.0 → 100%
##    65+      30.0      30.0  10.0 30.0 → 100%
```

If I was interested in looking at the distribution of people in the four geographic regions for each age group, based on the way the contingency table's arranged I would need row percents. That is, the rows would need to add up to 100%.

Row Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,1)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    21.4      32.1  17.9 28.6
##    36-65    12.5      21.9  40.6 25.0
##    65+      30.0      30.0  10.0 30.0
```

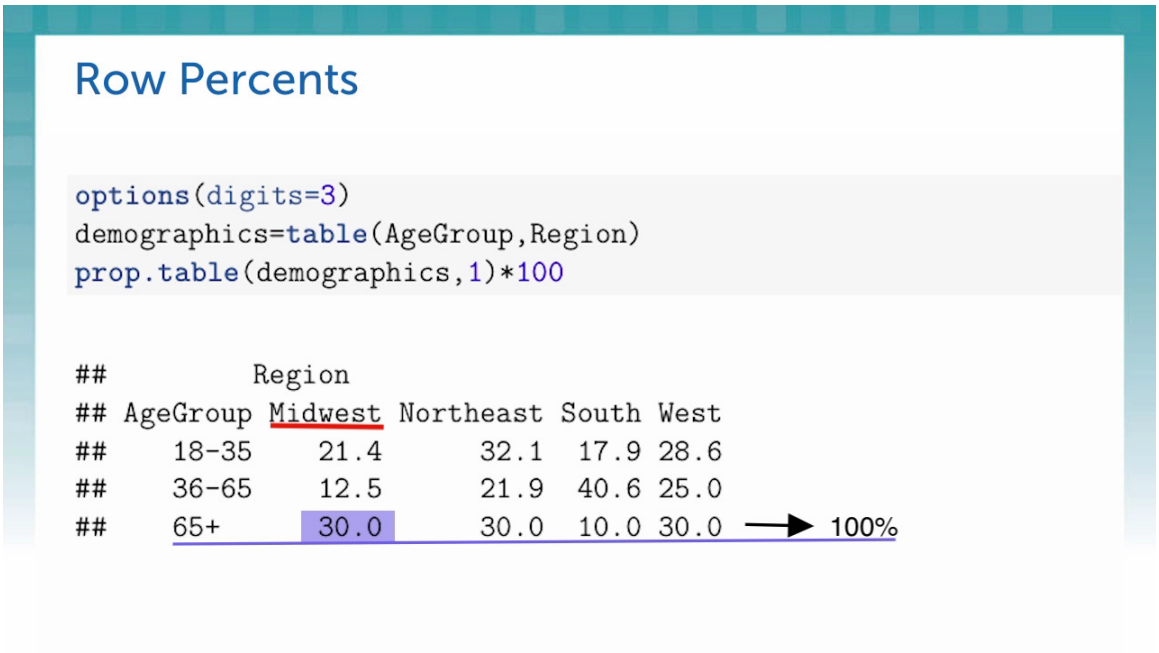
→ 100%

Row Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,1)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    21.4      32.1  17.9 28.6
##    36-65    12.5      21.9  40.6 25.0
##    65+      30.0      30.0  10.0 30.0
```

→ 100%



Comparisons of percentages among age groups can now be made for each region. So I can say something like, 21.4% of all the people in the sample age 18 to 35 live in the Midwest, while only 12.5% of the 36 to 65-year-olds live in the Midwest, and 30% of people over 65 live in the Midwest.

These percentages may seem far apart. But they weren't different enough for our chi-square test or our Fisher exact test to reject the null hypothesis of independence. Even though there were 80 people in the sample, it's still a relatively small sample, and so it has a low power to detect differences.

Row Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,1)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    21.4      32.1  17.9 28.6
##    36-65    12.5      21.9  40.6 25.0
##    65+      30.0      30.0  10.0 30.0
```

Prop.table will give them as proportions. And multiplying by 100 converts them to percents, which may be easier for us to think about.

Row Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,1)*100
```

1 for row percents

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    21.4      32.1  17.9 28.6
##    36-65    12.5      21.9  40.6 25.0
##    65+      30.0      30.0  10.0 30.0
```

The number 1 in the prop.table function is what directs R to compute row percents. Just try to remember that in matrix notation, the rows get mentioned first. So a 1 is appropriate.

Row Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,1)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    21.4      32.1  17.9 28.6
##    36-65    12.5      21.9  40.6 25.0
##    65+      30.0      30.0  10.0 30.0
```

Setting Options to 3 is a global setting, which will print numerical values to three digits. The default in R is seven digits, and this makes the table easier to read.

PS-- don't get too excited about the actual numbers in this table. They're more for an academic exercise than for their accuracy in describing current US demographics. In reality, I'm guessing that more than 10% of people over age 65 live in the South. But I digress.

Column Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,2)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    37.5      40.9  25.0 36.4
##    36-65    25.0      31.8  65.0 36.4
##    65+      37.5      27.3  10.0 27.3
##             100%      100%  100% 100%
```

If I was interested in looking at the distributions of people in the four geographic regions, I would use column percents, since this contingency table is arranged with region defining the columns. Notice for column percents, it's the columns that add up to 100%.

Comparisons of percentages among geographic regions can now be made for each age group.

Column Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,2)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    37.5      40.9  25.0 36.4
##    36-65    25.0      31.8  65.0 36.4
##    65+      37.5      27.3  10.0 27.3
```

100%

Column Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,2)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    37.5      40.9  25.0 36.4
##    36-65    25.0      31.8  65.0 36.4
##    65+      37.5      27.3  10.0 27.3
```

100%

Column Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,2)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    37.5      40.9  25.0 36.4
##    36-65    25.0      31.8  65.0 36.4
##    65+      37.5      27.3  10.0 27.3
```

100%

Column Percents

```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,2)*100
```

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    37.5      40.9  25.0 36.4
##    36-65    25.0      31.8  65.0 36.4
##    65+      37.5      27.3  10.0 27.3
```

100%

So I can say something like, 37.5% of all the people in the sample in the Midwest are 18 to 35 years old. Well, 40.9% in the Northeast are 18 to 35. 25% in the South are 18 to 35. And 36.4% in the West are 18 to 35 years old.

Column Percents

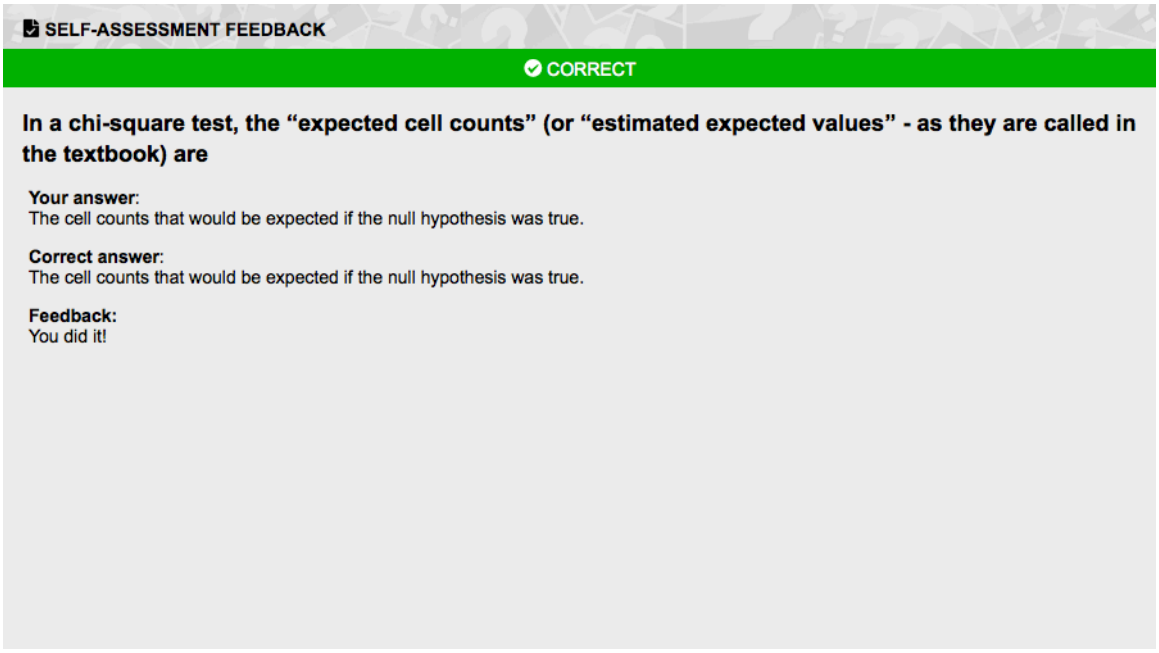
```
options(digits=3)
demographics=table(AgeGroup,Region)
prop.table(demographics,2)*100
```

2 for column percents

```
##          Region
## AgeGroup Midwest Northeast South West
##    18-35    37.5      40.9  25.0 36.4
##    36-65    25.0      31.8  65.0 36.4
##    65+      37.5      27.3  10.0 27.3
```

The number 2 in the prop.table function is what directs R to compute column percents. In matrix notation, the rows get mentioned first, and the columns get mentioned second. So a 2 indicates that we want column percents.

Question 1 answer:



Question 2 answer: