Introductory Statistics with R

Clay Ford

Spring 2020

Selected topics

- Summaries
- Counts and Proportions
- Means and Medians
- Comparing two proportions
- Comparing two means
- Comparing more than two means
- Association between categorical variables
- Linear association between numeric variables
- Simple Linear Regression

Summaries

Use on data frames or vectors in a data frame

```r
summary(acs12)
summary(acs12$income)
```

Look for...

- Unusual values (too low, too high)
- missing data (NA's)
- big differences between mean and median (skewed data?)
- consistent Factor level names ("male" vs "Male")
- order of Factor levels
- excess zeroes, or some other value
- wrong data types (numbers stored as Factors, etc)

Counts and Proportions

`table` returns counts of unique values in a vector

```
table(acs12$employment)
```

`prop.table` returns proportions of counts in a table

```
# %>% from dplyr or magrittr packages
table(acs12$employment) %>% prop.table()
```

`mosiac::prop.test` returns confidence intervals of proportions

```
mosaic::prop.test(acs12$employment == "employed")
mosaic::prop.test(acs12$income > 100000)
mosaic::prop.test(acs12$hrs_work > 40)
```

Counts and Proportions

`binom::binom.test` returns confidence intervals of all proportions in a table

```
binom.confint(x = table(acs12$employment),
              n = sum(table(acs12$employment)),
              method = "prop.test")
```

Leaving out the `method` argument returns 11 different CI estimates.

Confidence intervals

95% Confidence Interval theory:

- sample the data
- calculate a 95% confidence interval
- repeat many times

About 95% of confidence intervals will contain the “true” value you’re estimating.

Means and Medians

The median is the middle of the sorted data. The mean is the "balance point" of the data. Symmetric data have similar means and medians.

```
# specify na.rm = TRUE to ignore missing data
mean(acs12$hrs_work, na.rm = TRUE)
median(acs12$hrs_work, na.rm = TRUE)
```

`t.test` returns a CI of a mean.
`wilcox.test` returns a CI for the median.

```
t.test(acs12$hrs_work)
wilcox.test(acs12$hrs_work, conf.int = TRUE)
```

Comparing two proportions

`xtabs` cross tabulates categorical variables:

```
xtabs(~ citizen + disability, data = acs12)
```

`addmargins` adds margin totals in the specified dimension (1 = rows, 2 = columns)

```
xtabs(~ citizen + edu, data = acs12) %>%
  addmargins(margin = 2)
```

Comparing two proportions

Use `prop.test()` to test the hypothesis that the proportions are equal. The first argument `x` takes number of "successes" for each group; the second argument `n` takes total number in each group

```
tab <- xtabs(~ citizen + edu, data = acs12) %>%
  addmargins(margin = 2)
prop.test(x = tab[,"college"], n = tab[,"Sum"])
```

Hypothesis Tests

- There are two competing hypotheses:
  - The Null: this is usually “no difference” or 0
  - The Alternative: this is something like “different” or “not equal to 0”
- A hypothesis test returns a p-value
- A p-value is the probability of getting the result we got, or more extreme, given the null hypothesis
- Small p-values, say less than 0.05, are good evidence that we can reject the null hypothesis

Some p-value advice

- A p-value is not the probability that null hypothesis is true
- Scientific conclusions or business decisions should not be based only on whether a p-value passes a specific threshold
- A p-value does not measure the size of an effect or the importance of a result
- A p-value is simply the probability that a statistical summary of the data would be equal to or more extreme than its observed value if the null hypothesis were true

See The ASA Statement on p-Values

Comparing two means

The `t.test` function tests the null hypothesis that two means are the same. It assumes each sample came from Normally distributed populations.

```
t.test(income ~ gender, data = acs12)
```

The `wilcox.test` function tests the null hypothesis that two samples came from the same distribution.

```
wilcox.test(income ~ gender, data = acs12)
```

It is sometimes suggested as an alternative to the t-test if the normality assumption is suspect, but it really is a different test.

Comparing more than 2 means

The ANOVA procedure is usually emmployed to determine if there is a statistically significant difference between more than 2 means.

```
aov.out <- aov(log(income) ~ race,
               data = subset(acs12, income > 0))
summary(aov.out)
```

A low p-value provides evidence that the means differ. If this is the case, we usually follow-up the test with a *post-hoc* procedure such as Tukey's HSD.

```
tukey.out <- TukeyHSD(aov.out)
plot(tukey.out)
```

This creates a set of confidence intervals on the differences between the means.

Association between categorical variables

The `chisq.test` function tests the null hypothesis that two categorical variables are not related.

```
xtabs(~ race + employment, data = acs12) %>%
  chisq.test()
```

A small p-value provides evidence against the null hypothesis of no association. However it does not tell us where the association is or the magnitude of the association. For this a mosaic plot is useful.

```
xtabs(~ employment + race, data = acs12) %>%
  mosaicplot(shade = TRUE)
```

Linear association between numeric variables

- Correlation summarizes the strength and direction of a linear relationship between two numeric variables
- Ranges from -1 to 1
  - -1 is a perfectly negative relationship (as one goes up, the other goes down)
  - 0 means no relationship
  - 1 is a perfectly positive relationship (as one goes up, the other goes up)
- Correlation is not the same as cause
- Examine a scatter plot when calculating correlation to determine whether or not it's appropriate

Correlation

The `cor.test` function returns a confidence interval on the correlation

```
mosaic::cor.test(age ~ income,
                 use = "pairwise.complete.obs",
                 data = acs12)
```

If you have missing data and want to proceed with calculating correlation, set `use = "pairwise.complete.obs"` to use all available pairs of data.

Simple Linear Regression

Simple linear regression is basically summarizing the relationship between two variables as a straight line, using the familiar slope-intercept formula:

$$y = a + bx$$

This implies we can approximate the mean of y for a given value of x by multiplying x by some number and adding a constant value.

Simple linear regression

The `lm` function fits linear models. The formula `income ~ hrs_work` translates to "regress `income` on `hrs_work`", or "model `income` as a function of `hrs_work`".

```
mod <- lm(income ~ hrs_work, data = acs12)
summary(mod)
```

Calling `plot` on the model object produces diagnostic plots to assess various assumptions. Two of interest:

```
plot(mod, which = 1)
plot(mod, which = 2)
```

The first plot assess the constant variance assumption. The second assesses the Normality assumption.

References

The library provides access to many books (electronic and hard copy) that introduce statistics with R.

- Introductory Statistics : A conceptual approach using R (Ware, et al)
- Statistics and Data with R: An applied approach through examples (Cohen and Cohen)
- Learning Statistics with R (Navarro)
- Introductory Statistics with R (Dalgaard)
- A Course in Statistics with R (Tattar)

Of course Google is your friend.

Thanks for coming

- For statistical consulting: statlab@virginia.edu
- Sign up for more workshops or see past workshops: http://data.library.virginia.edu/training/
- Register for the Research Data Services newsletter to be notified of new workshops: http://data.library.virginia.edu/newsletters/