

# Statistics with R

## *Regression and ANOVA*

Scott Hetzel

University of Wisconsin – Madison

Summer Institute for Training in Biostatistics (2009)

Derived from: “Introductory Statistics with R” by: Peter Dalgaard

and from previous notes by Deepayan Sarkar, Ph.D

# What we Discussed Last Time

---

Last time we discussed the functions used for making inferences on count and tabular data. Functions used:

- `binom.test` to test binomial probability of success
- `prop.test` to test multiple proportions equality
- `fisher.test` to test for association, mostly used for 2 x 2 tables. Inference is based on odds ratio
- `chisq.test` to test for association, mostly used for larger dimensioned tables.

Dr. Gangnon discussed the functions used for making inferences on continuous data. Functions used:

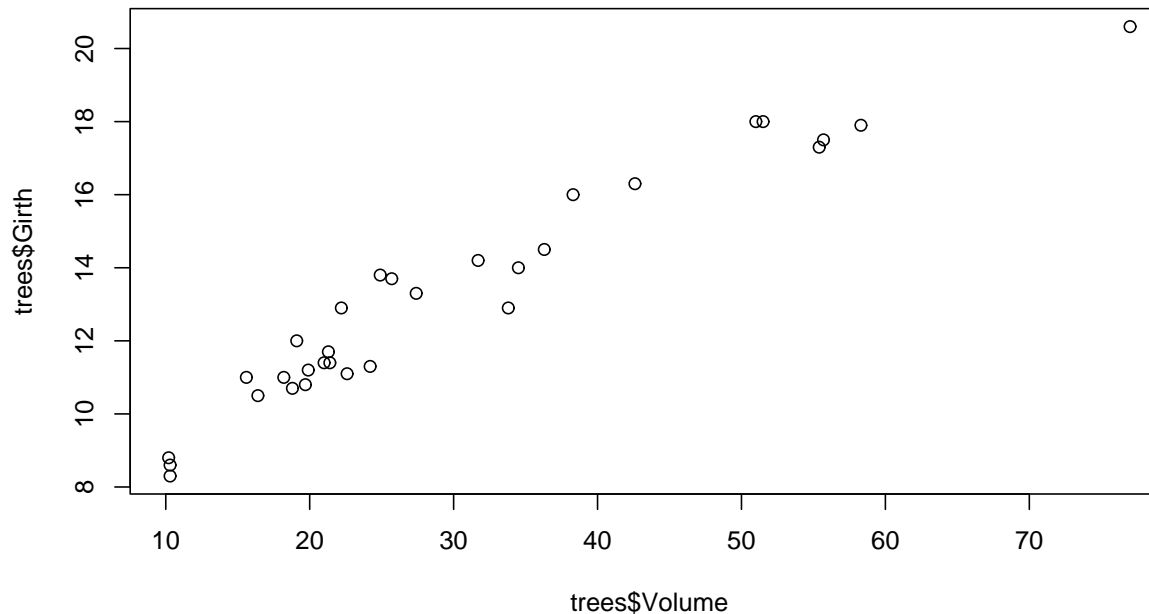
- `t.test` to test:
  - $H_0 : \mu = \mu_0$
  - $H_0 : \mu_1 = \mu_2$
  - $H_0 : \delta = 0$
- `wilcox.test` to test about the median:
  - $H_0 : M = M_0$
  - $H_0 : M_1 = M_2$
  - $H_0 : M_\delta = 0$

# Correlation Coefficient

---

When examining the relationship between two continuous variables, one of the simplest methods of quantifying this relationship is through the linear correlation coefficient. It summarizes what direction the association is and how strongly the linear association is between two continuous variables.

```
> cor(trees$Volume, trees$Girth)
[1] 0.9671194
> plot(trees$Volume, trees$Girth)
```



# Linear Regression

---

Simple linear regression is a technique that is used to explore the nature of the relationship between two continuous random variables. More specifically simple linear regression is used to investigate the change in one variable, called the *response*, which occurs because of a change in another variable, called the *explanatory variable*. Functions that we will use in **R** for linear regression are:

- `lm()`
- `plot()`
- `abline()`
- `resid()`
- `predict()`

# Linear Regression (Cont.)

---

`lm()`, standing for Linear Model, fits a linear model to the data using the Least Squares Method. The model description should look like:

Model: Response  $\sim$  Explanatory1 + Explanatory2 + ...

The default is to have the intercept in the model. If you put -1 at the end of the model statement, this removes the intercept from the model.

The general format is:

```
> lm1 <- lm(Model, data=name of data)
```

See `args(lm)` or `?lm()` for more details.

# Example of Simple Regression

---

```
> treeLM <- lm(Volume ~ Girth, data=trees)
```

```
> treeLM
```

Call:

```
lm(formula = Volume ~ Girth, data = trees)
```

Coefficients:

```
(Intercept)  Girth  
   -36.943    5.066
```

This does not help us very much at all. This output only gives us the estimates for  $\beta_0$  and  $\beta_1$ . Using `summary(treeLM)` will give us more meaningful output.

# Regression Example (Cont.)

---

```
> summary(treeLM)
```

```
Call:
```

```
lm(formula = Volume ~ Girth, data = trees)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-8.0654	-3.1067	0.1520	3.4948	9.5868

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	-36.9435	3.3651	-10.98	7.62e-12	***
Girth	5.0659	0.2474	20.48	< 2e-16	***

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 4.252 on 29 degrees of freedom
```

```
Multiple R-Squared: 0.9353, Adjusted R-squared: 0.9331
```

```
F-statistic: 419.4 on 1 and 29 DF, p-value: < 2.2e-16
```

# Summarizing `summary(treeLM)`

---

- Residuals table: Not so much the table that concerns us, but the distribution of the residuals. We will talk more about this later in the diagnostic checking of assumptions.
- Coefficients table: Again the estimates:  $\hat{\beta}_0$  and  $\hat{\beta}_1$  are given along with standard error. The p-values are for the two tests:  $H_0 : \beta_0 = 0$  and  $H_0 : \beta_1 = 1$  The p-values are highly significant, so we would reject those null hypotheses.
- Residual Standard Error: Is the square root of the Mean Square Error. Found by: `sqrt(deviance(treeLM)/df.residual(treeLM))`
- Multiple R-Squared: Means that 93.53% of the total response variation is due to the linear association between the variables. Notice that the square root of 0.9353 is the correlation coefficient 0.9671.
- F-statistic p-value: P-value for the test of two models. This model versus a model with only the intercept. In the case of only one explanatory variable this reduces to the same t-test for  $\beta_1$ .

# Assumptions for Linear Regression

---

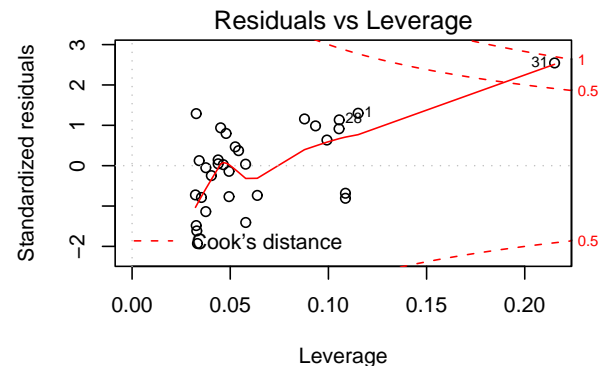
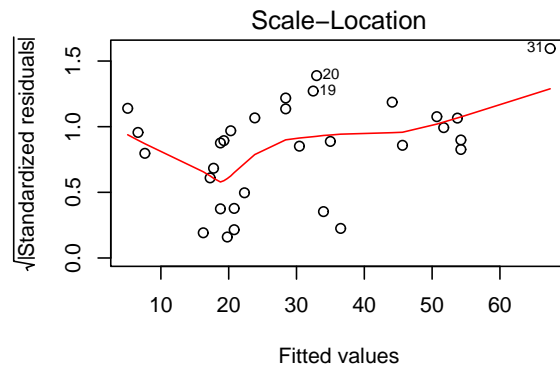
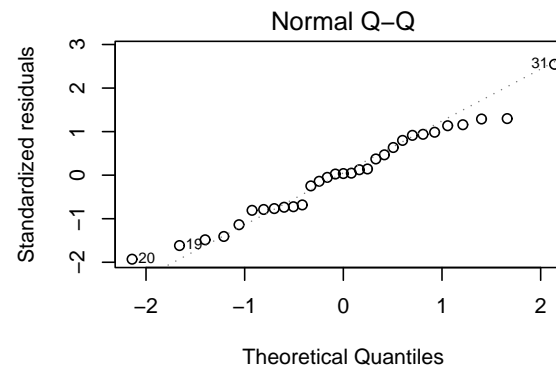
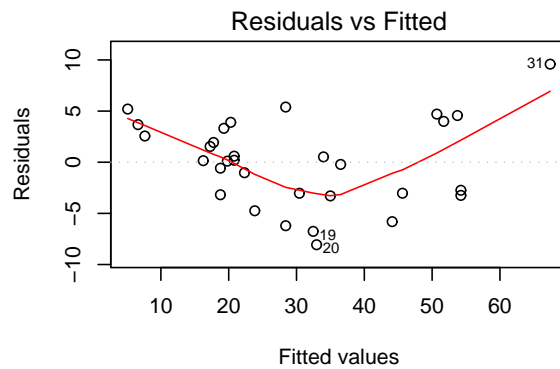
- $\varepsilon_i$  are independent of each other,  $i = 1, 2, \dots, n$
- $\varepsilon_i$  are normally distributed with mean 0, and equal variance,  $\sigma^2$

Independence in the errors is the same as independence in the responses,  $y_i$ . This is hard to explicitly check but is normally taken care of by a good study design.

The assumption of normality with equal variance is checked by looking at the residuals versus fitted values plot and the Q-Q normal plot. This can be done in one command in **R**. `plot(treeLM)`.

# Checking Normality of Residuals

```
> layout(matrix(c(1,2,3,4), nrow=2, byrow=T))  
> plot(treeLM)
```

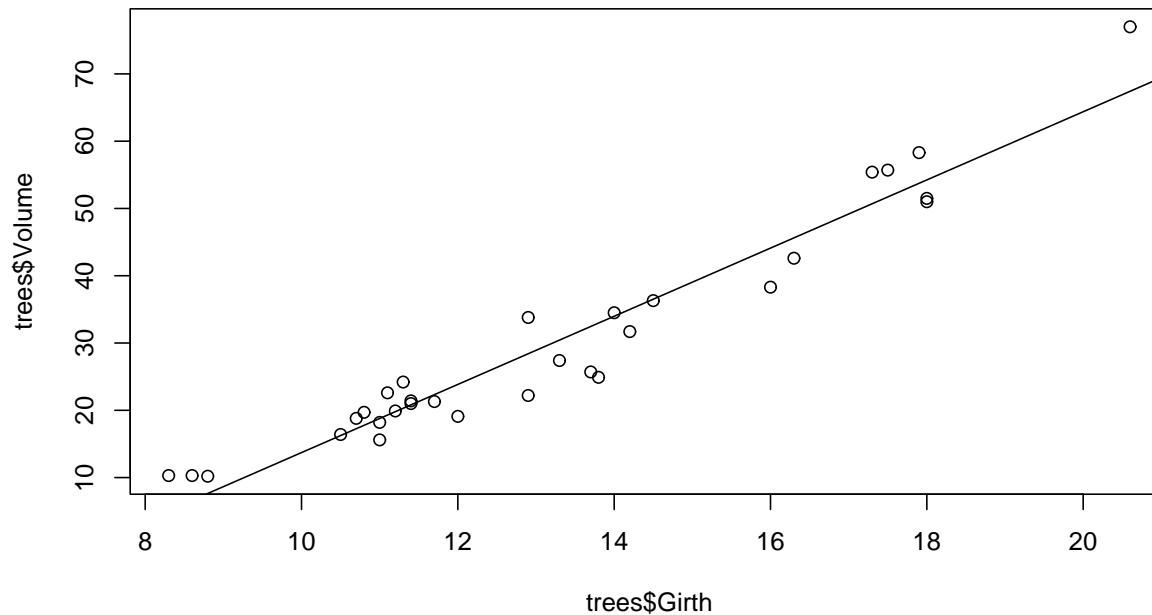


# Simple Linear Regression Plot

---

A nice graphical representation of a simple linear regression is to use the `plot` function and the `abline` function to place the line on the scatter plot.

```
> plot(trees$Girth, trees$Volume)
> abline(treeLM)
```



# Linear Regression Concluded

---

There are functions in **R** that will allow you to look at the residuals and predicted values as vectors. Also helpful can be the `pairs` function which plots scatterplots of all the combinations of two variables in a data frame. Not too useful if the number of variables is large because plots will be too small.

```
> resid(treeLM) # summary(treeLM)$resid will work too
```

```
> predict(treeLM) # fitted(treeLM) will do the same thing
```

```
> pairs(stackloss, panel=panel.smooth)
```

# Exercises in Using R

---

Using `stack.loss` as the response variable from the data set `stackloss`:

1. Fit simple linear regression models using the remaining variables as the explanatory variable
2. Which variables have an estimated slope that is significantly different than 1?
3. Which variable has the highest correlation with `stack.loss`?
4. Check the assumptions of the regression analysis. Do any of the models need a transformation to the data to better achieve the assumptions?

# Exercises in Using R Answers

---

1. 

```
> attach(stackloss)
> AirLM <- lm(stack.loss ~ Air.Flow)
> WaterLM <- lm(stack.loss ~ Water.Temp)
> AcidLM <- lm(stack.loss ~ Acid.Conc.)
```
2. 

```
> c(summary(AirLM)$coef[2,4], summary(WaterLM)$coef[2,4],
+ summary(AcidLM)$coef[2,4])
[1] 3.774296e-09 2.028017e-07 7.252300e-02
```
3. 

```
> c(summary(AirLM)$r.sq, summary(WaterLM)$r.sq,
+ summary(AcidLM)$r.sq)
[1] 0.8457809 0.7665080 0.1598637
```
4. 

```
> plot(AirLM)
> plot(WaterLM)
> plot(AcidLM)
```

No output will be shown here, but looking at it yourself, you can see that AirLM seems the best to hold the assumptions, where constant variance is violated in WaterLM and AcidLM. Transformations or polynomial regression should be used on the latter two variables.

# Analysis of Variance - ANOVA

---

As previously discussed, comparing means between two continuous variables, more commonly a continuous variable split into two groups by some binary factor, is done in **R** by the functions `t.test` or `wilcox.test` depending on normality of the continuous variable. But what if we want to compare means when the factor has more than two levels? This is done by using the one-way ANOVA method which is very easy to run in **R**.

Useful functions in **R** for ANOVA:

- `anova()`
- `lm()` The same as what we used in Regression
- `pairwise.t.test()`
- `bartlett.test()`

Hypothesis:

- $H_0 : \mu_1 = \mu_2 = \dots = \mu_k$
- $H_1 : \text{At least one } \mu_i \text{ is not equal to a different } \mu_j$

# ANOVA (Cont.)

---

So if this is a method for comparing means, why is it called Analysis of **Variance**?

The total variation of the observations from the grand mean can be split into two sources of variation. Namely, the variation within groups (Error Sum of Squares)  $SS_{Error} = \sum_i \sum_j (x_{ij} - \bar{x}_{.})$  and the variation between groups (Treatment Sum of Squares)  $SS_{TRT} = \sum_i n_i (\bar{x}_i - \bar{x}_{.})^2$ . The ANOVA test statistic is:  $F = \frac{MS_{TRT}}{MS_{Error}}$ .

Where,  $MS_{TRT}$  is  $\frac{SS_{TRT}}{k-1}$  and  $MS_{Error}$  is  $\frac{SS_{Error}}{N-k}$ . The denominator is the pooled variance obtained by combining the individual group variances and this is an estimate of  $\sigma^2$ . If there is no group effect, then the variance of the group means from the grand mean, which is estimated by the numerator, should also estimate  $\sigma^2$ . With a group effect, the group means will be considerably different than the grand means hence  $MS_{TRT}$  will be larger, making the F statistic large enough to reject the null hypothesis. So the inference is based upon comparing estimates of variance, hence analysis of variance. However, the reason for obtaining a significant test statistic is because of the differences in group means.

- Assume k independent and normally distributed random variables one per group
- Assume  $\sigma_1 = \sigma_2 = \dots = \sigma_k$

# ANOVA (Cont.)

---

To run the one-way ANOVA in **R** we need to make sure we have a numeric variable that is broken into multiple groups by a factor variable.

Then to conduct the test we first have to set up a linear model like we did in regression analysis. Like so:

```
> anvl <- lm(num.var ~ factor.var, data = yourDF)
```

A word of caution: If your factor is a numeric, i.e. 1, 2, 3, ..., the program will then read it as a regression analysis and not a comparison of means. In this case you can use the `as.factor()` function to coherse the numeric into a factor.

To obtain the ANOVA output:

```
> anova(anvl)
```

# ANOVA Example

---

```
> attach(PlantGrowth)
> str(PlantGrowth)
'data.frame': 30 obs. of 2 variables:
 $ weight: num 4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 ...
 $ group : Factor w/ 3 levels "ctrl","trt1",...: 1 1 1 1 1 ...
```

```
> anova(lm(weight ~ group))
Analysis of Variance Table
```

```
Response: weight
```

	DF	Sum Sq	Mean Sq	F value	Pr(>F)
group	2	3.7663	1.8832	4.8461	0.01591
Residuals	27	10.4921	0.3886		

```
> anova(lm(weight ~ as.numeric(group))) # What is this testing?
Analysis of Variance Table
```

```
Response: weight
```

	DF	Sum Sq	Mean Sq	F value	Pr(>F)
as.numeric(group)	1	1.2202	1.2202	2.6204	0.1167
Residuals	28	13.0382	0.4657		

# Multiple Comparisons

---

So after finding that there is at least one difference between two of the groups means, what is the first question the investigator is going to ask you? "Which groups are different?" This can be figured out by using `pairwise.t.test`.

- `pairwise.t.test(num.var, factor.var, p.adj="bonf")`
- The output for this test is a matrix of adjusted p-values for the individual t-tests. P-values less than 0.05 are where there are significantly different means.

P-values need to be adjusted because of the multiple tests on the same data. We will not go into detail, however, when running multiple tests the error rate becomes inflated hence the probability of rejecting the null when there truly is no difference ( $\alpha$ ) for the tests is larger than 0.05. There are multiple adjustment methods. Bonferroni's is conservative meaning that only clear cut and true differences will be detected. In **R** the Bonferroni correction multiplies the unadjusted p-values by the number of two-way comparisons. See `?p.adjust` for explanations for the others. The default is "holm", which according to `?p.adjust` "dominates" the Bonferroni method.

# ANOVA Example Revisted

---

```
> pairwise.t.test(weight, group)
Pairwise comparisons using t tests with pooled SD
data:  weight and group
      ctrl  trt1
trt1 0.194  -
trt2 0.175 0.013
P value adjustment method:  holm
```

# Checking Assumptions

---

Remember the two assumptions for the ANOVA model:

- Assume  $k$  independent and normally distributed random variables one per group
- Assume  $\sigma_1 = \sigma_2 = \dots = \sigma_k$

To check normality assumption will be to check the values in each group individually by plotting them in a [qqnorm](#) plot and seeing if the dots are reasonably close to the  $45^\circ$  line. Also, you can plot histograms of the data separated by groups and see if they follow a normal bell curve.

# Checking Assumptions

---

To check whether a variable has the same variance in all groups we can use the `bartlett.test` function in **R**.

```
> bartlett.test(weight ~ group)
Bartlett test of homogeneity of variances
data:  weight by group
Bartlett's K-squared = 2.8786, df = 2, p-value = 0.2371
```

If Bartlett test's p-value is less than 0.05 then the assumption of equal variances would be rejected. `oneway.test` is a function in **R** that runs a similar procedure without the constraints of the equal variance assumption. Read 6.1.2 in the text for more details.

# Exercises in Using R

---

Revisiting data set ChickWeight from Lecture 3. Remember I asked for a graphical representation to see if there is a difference in weights based on diets. Now we can do the formal test.

1. Run ANOVA to see if there is a difference in mean weights at the end of the study based on diets.
2. What is the p-value of the test?
3. If p-value is significant, which it is because I wouldn't ask this question if it wasn't, where is the sig. differences?
4. Is the assumption of equal variance valid?

# Exercises in Using R Answers

---

- ```
> Day21CW <- subset(ChickWeight, subset=ChickWeight$Time==21)
> anv1 <- anova(lm(Day21CW$weight ~ Day21CW$Diet))
```
- ```
> anv1$Pr[1]
[1] 0.006857959
```
- ```
> pairwise.t.test(Day21CW$weight, Day21CW$Diet)
Pairwise comparisons using t tests with pooled SD
data: Day21CW$weight and Day21CW$Diet
      1      2      3
1  0.4786  -    -
2  0.0053  0.2355 -
3  0.1391  0.5731 0.5731
P value adjustment method: holm
```
- ```
> bartlett.test(Day21CW$weight ~ Day21CW$Diet)
Bartlett test of homogeneity of variances
data: Day21CW$weight by Day21CW$Diet
Bartlett's K-squared = 3.0524, df = 3, p-value = 0.3836
```