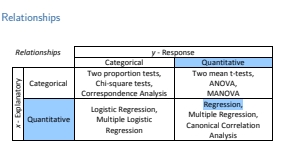
Linear Regression and Correlation

DS705

Relationships

Relationships		*y* - Response	
		Categorical	Quantitative
x - Explanatory	Categorical	Two proportion tests, Chi-square tests, Correspondence Analysis	Two mean t-tests, ANOVA, MANOVA
	Quantitative	Logistic Regression, Multiple Logistic Regression	Regression, Multiple Regression, Canonical Correlation Analysis



- much of statistics is about exploring relationships between variables
- In ANOVA for instance we could study the relationship between the type of curriculum taught and the reading score achieved by students.
  - the explanatory or independent variable x is the type of curriculum
  - the response or dependent variable y is the reading score
- This unit is about simple linear regression when there is a single quantitative explanatory variable and a single quantitative response variable
- In future units, you'll learning about having multiple explanatory variables and/or response variables

Explore Relationships and Make Predictions

Manufacturing example: producing more items requires more time

$x =$ number of items, $y =$ production time (minutes)

- Model the relationship.
- Make predictions.

Explore Relationships and Make Predictions

Manufacturing example: producing more items requires more time

$x =$ number of items, $y =$ production time (minutes)

- Model the relationship.
- Make predictions.

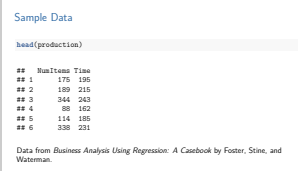
- no audio

Sample Data

```
head(production)
```

```
##   NumItems Time
## 1      175  195
## 2      189  215
## 3      344  243
## 4       88  162
## 5      114  185
## 6      338  231
```

Data from *Business Analysis Using Regression: A Casebook* by Foster, Stine, and Waterman.


Sample Data

```
head(production)
```

```
##   NumItems Time
## 1      175  196
## 2      189  215
## 3      344  243
## 4       88  162
## 5      114  185
## 6      338  231
```

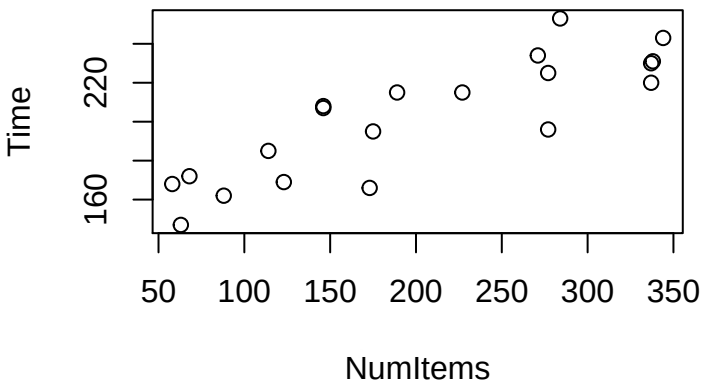
Data from *Business Analysis Using Regression: A Casebook* by Foster, Stine, and Waterman.

- the production data is stored in a dataframe with two columns NumItems and Time
- each row represents a single observation in which both the number of times and the production time were recorded

Plot the data

Always start with a scatterplot:

```
with(production,plot(NumItems,Time))
```



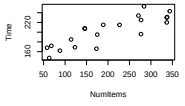
Linear Regression and Correlation

Plot the data

Plot the data

Always start with a scatterplot:

```
with(production,plot(NumItems,Time))
```



- is there a linear trend?
- real data almost never lies perfectly along a line
- to use a linear model, the data should look like a line with some added noise or jitter

Correlation

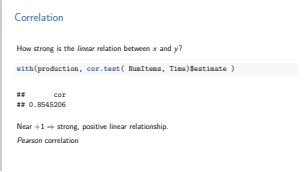
How strong is the *linear* relation between x and y?

```
with(production, cor.test( NumItems, Time)$estimate )
```

```
##       cor
## 0.8545206
```

Near $+1 \Rightarrow$ strong, positive linear relationship.

Pearson correlation


Correlation

How strong is the *linear* relation between x and y?

```
with(production, cor.test( RunItems, Time)$estimate )
```

```
##       cor
## 0.8545206
```

Near $+1 \Rightarrow$ strong, positive linear relationship.

Pearson correlation

- correlation is useful only for linear relationships.
- we verified this was a linear relationship by inspecting the scatter plot.
- cor.test produces more than simply the correlation coefficient, we'll look at some of the other stuff later.
- this is the classical *Pearson* correlation coefficient, there are other ways to quantify correlation that are based on ranking the data

Desired Model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

- x = number of items
- y = production time in minutes
- $\hat{y}$ predicted value of y
- β_0 estimated y-intercept
- β_1 estimated slope of line

Linear Regression and Correlation

Desired Model

Desired Model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

- x = number of items
- y = production time in minutes
- $\hat{y}$ predicted value of y
- β_0 estimated y-intercept
- β_1 estimated slope of line

- no audio

Confusion alert: too many y's

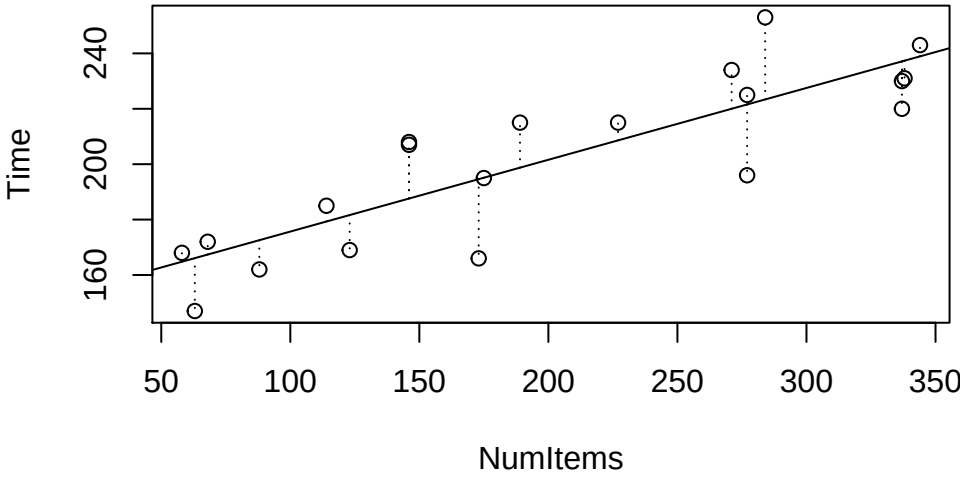
- $\hat{y}$: response values predicted estimated model
- y: theoretical response values from the true model
- y_i: observed values of the response variable

Confusion alert: too many y's

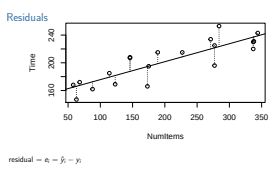
- $\hat{y}$: response values predicted estimated model
- y: theoretical response values from the true model
- y_i: observed values of the response variable

- no audio

Residuals



residual $= e_i = \hat{y}_i - y_i$



- finding the line is all about the residuals
- a residual is the vertical difference between an observed y-value and the predicted y-value from the line
- the least squares regression line is found by choosing the line that minimizes the sum square residuals
- you should read about the equations used for finding the slope and intercept in your book
- we'll see how to find the model and plot the line in the next few slides

Least Squares Regression Concept

Insert video here

Add clickable link in lower box

`http://hspm.sph.sc.edu/courses/J716/demos/LeastSquares/LeastSquaresDemo.html`

Least Squares Regression Concept

Insert video here

Add clickable link in lower box

http://hsph.sph.sc.edu/courses/J716/demos/LeastSquares/LeastSquaresDemo.html

video slide . . . narrated presentation using the java applet

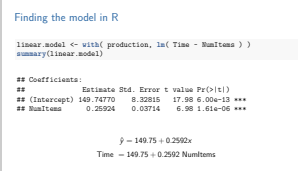
Finding the model in R

```
linear.model <- with( production, lm( Time ~ NumItems ) )
summary(linear.model)
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 149.74770    8.32815   17.98 6.00e-13 ***
## NumItems      0.25924    0.03714    6.98 1.61e-06 ***
```

$$\hat{y} = 149.75 + 0.2592x$$

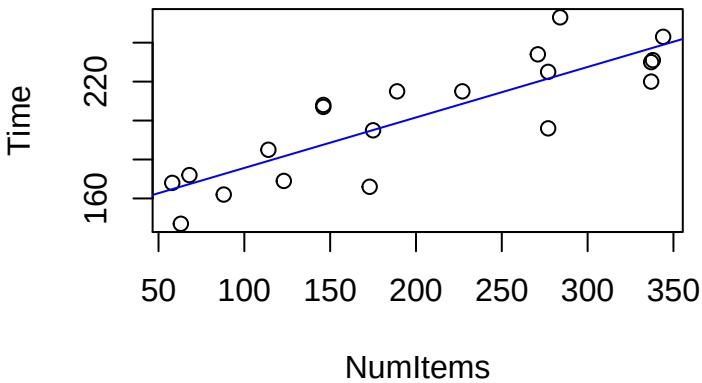
$$\text{Time} = 149.75 + 0.2592 \text{ NumItems}$$



- the summary command actually reports a whole lot more than the coefficients of the model and we'll look at more of it later, for now we are just interested in the coefficients of the model
- (Highlight the block (Intercept) 149.74770 and NumItems 0.25924)

Plotting the least-squares line

```
with( production, plot( NumItems, Time) )
abline( linear.model, col = 'blue' )
```

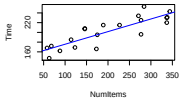


Linear Regression and Correlation

Plotting the least-squares line

Plotting the least-squares line

```
with( production, plot( NumItems, Time) )
abline( linear.model, col = 'blue' )
```



- no audio

Extracting Coefficients

```
linear.model$coef[1]
```

```
## (Intercept)
##    149.7477
```

```
linear.model$coef[2]
```

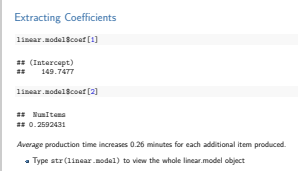
```
##  NumItems
## 0.2592431
```

Average production time increases 0.26 minutes for each additional item produced.

- Type `str(linear.model)` to view the whole linear.model object

Linear Regression and Correlation

Extracting Coefficients

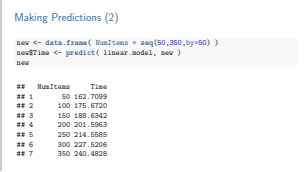


The y-values predicted from the model are estimates of the *average* response value for each x

Making Predictions (2)

```
new <- data.frame( NumItems = seq(50,350,by=50) )
new$Time <- predict( linear.model, new )
new
```

```
##   NumItems     Time
## 1       50 162.7099
## 2      100 175.6720
## 3      150 188.6342
## 4      200 201.5963
## 5      250 214.5585
## 6      300 227.5206
## 7      350 240.4828
```


Making Predictions (2)

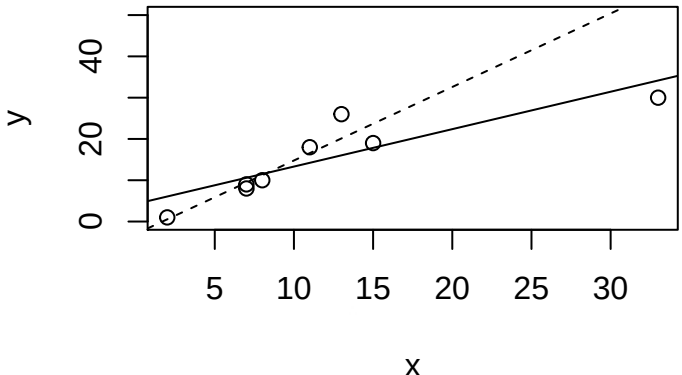
```
new <- data.frame( NumItems = seq(50,350,by=50) )
newTime <- predict( linear.model, new )
new
```

```
##   NumItems     Time
## 1       50 162.7099
## 2      100 175.6720
## 3      150 188.6342
## 4      200 201.5963
## 5      250 214.5585
## 6      300 227.5206
## 7      350 240.4828
```

- the predict function requires a dataframe containing the values of the explanatory variable to be used for making predictions
- the estimated average time to produce 100 items is about 176 minutes
- think of this estimate like a sample mean, it is a point estimate of the population mean production time
- for a particular production run that produces the 100 items, the actual time is expected to vary significantly around 176 minutes

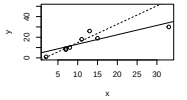
Outliers and Influential Observations

```
x <- c(2,7,7,8,11,13,15,33); y <- c(1,9,8,10,18,26,19,30)
plot( x, y, ylim=c(0,50) )
mod1 <- lm( y~x ); mod2 <- lm( y[-8] ~ x[-8] )
abline(mod1); abline(mod2,lty='dashed')
```



Outliers and Influential Observations

```
x <- c(2,7,7,8,11,13,15,33); y <- c(1,9,8,10,18,26,19,30)
plot( x, y, ylim=c(0,50) )
mod1 <- lm( y~x ); mod2 <- lm( y[-8] ~ x[-8] )
abline(mod1); abline(mod2,lty='dashed')
```



- as always outliers are any observations that are outside the pattern, the point (33,30) is an outlier
- an outlier doesn't have to be influential and an influential point need not be an outlier
- to see if a point is influential, plot the line with all the data, this is the solid line
- and plot the line with the point (33,30) removed resulting in the dashed line
- since the line is much different with the point removed, the point (33,30 is an influential point
- we should determine if we want to restrict our attention to x values between 2 and 15 in which case the dashed line is fine, or if we need to extend the model further then we probably need additional data before we can assess whether a linear model is appropriate
- your book distinguishes between outliers, influential points, and points with so-called "leverage" ... I don't find these distinctions important. If a point doesn't seem to fit

Inference for Regression

- estimate population slope
- estimate population correlation
- test for significant linear relationship / correlation
- estimate average response at given x
- estimate future individual response at given x

Inference for Regression

- estimate population slope
- estimate population correlation
- test for significant linear relationship / correlation
- estimate average response at given x
- estimate future individual response at given x

- these are typical inference procedures for linear regression, but traditional, parametric approaches to these procedures are based on some assumptions or requirements about the data
- if those requirements aren't met, the parametric approaches may be invalid or inaccurate
- we'll discuss those assumptions and how to check them in the next several slides

Simple Linear Regression Model

Simple Linear Regression:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \qquad \epsilon_i \overset{\text{ind}}{\sim} N(0, \sigma_\epsilon)$$

$x =$ explanatory, independent, predictor , $y =$ response, depdendent

Assumptions / Requirements:

1. errors have mean 0
2. errors have the same variance for all x
3. errors are independent of each other
4. errors are normally distributed.

Simple Linear Regression Model

Simple Linear Regression:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \qquad \epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon)$$

$x =$ explanatory, independent, predictor , $y =$ response, dependent

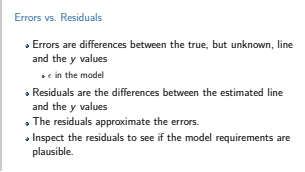
Assumptions / Requirements:

1. errors have mean 0
2. errors have the same variance for all x
3. errors are independent of each other
4. errors are normally distributed.

- of course you can fit a line to any data you like, but if you want to make inferences based on the linear relationship, you need need to first verify that these requirements are met.
- if the requirements are not met, then more advanced techniques, beyond those discussed below are needed.

Errors vs. Residuals

- Errors are differences between the true, but unknown, line and the y values
  - ϵ in the model
- Residuals are the differences between the estimated line and the y values
- The residuals approximate the errors.
- Inspect the residuals to see if the model requirements are plausible.


Errors vs. Residuals
Errors are differences between the true, but unknown, line and the y values
ϵ in the model
Residuals are the differences between the estimated line and the y values
The residuals approximate the errors.
Inspect the residuals to see if the model requirements are plausible.

- no audio

Check Requirements before Inference

Assumptions / Requirements:

1. errors have mean 0
2. errors have the same variance for all x
3. errors are independent of each other
4. errors are normally distributed.

To make things simpler extract all the info. first:

```
resids <- linear.model$resid # extract residuals from model
NumItems <- production$NumItems
Time <- production$Time
TimeFit <- linear.model$fitted.values
```

Check Requirements before Inference

Assumptions / Requirements:

1. errors have mean 0
2. errors have the same variance for all x
3. errors are independent of each other
4. errors are normally distributed.

To make things simpler extract all the info. first:

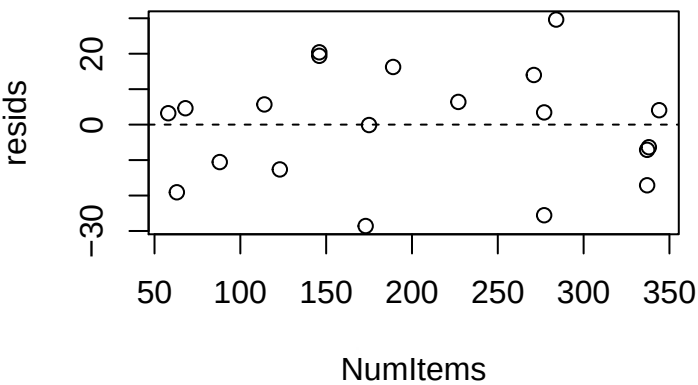
```
resids <- linear.model$resid # extract residuals from model
NumItems <- production$NumItems
Time <- production$Time
TimePit <- linear.model$fitted.values
```

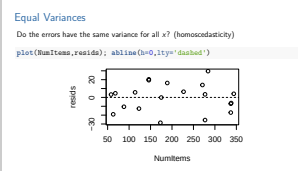
- We are going to use the residuals as surrogates for the errors and by inspecting the residuals see if each requirement seems to be satisfied
- The first requirement is automatic, since mathematically the residuals always add to zero.

Equal Variances

Do the errors have the same variance for all x? (homoscedasticity)

```
plot(NumItems,resids); abline(h=0,lty='dashed')
```



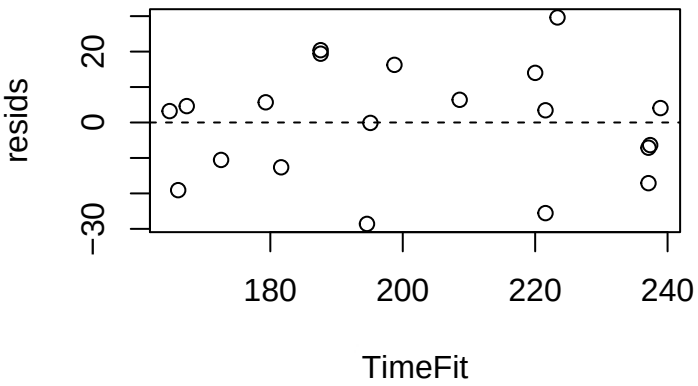


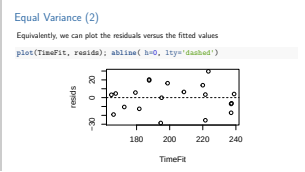
- we want to see the same amount of spread, vertically, at all values of x
- some authors call this equal variances, others say the variance is constant, still others say the variance is homoscedastic
- this certainly seems to be true for this data
- the next slide shows an example of the kind of residual plot we do not want to see

Equal Variance (2)

Equivalently, we can plot the residuals versus the fitted values

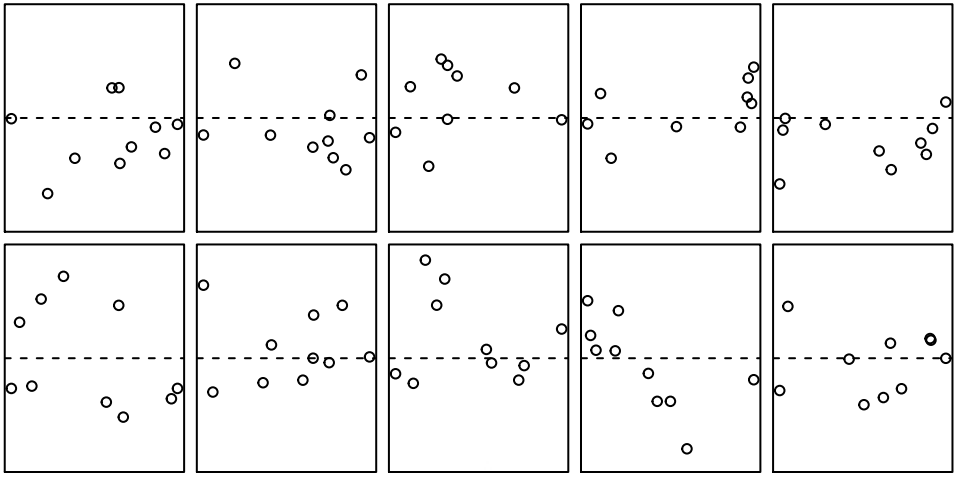
```
plot(TimeFit, resids); abline( h=0, lty='dashed')
```

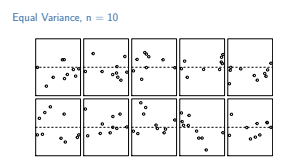




- the last two plots look exactly the same and for simple linear regression they are identical
- in multiple regression there are multiple explanatory variables x so we look at only the plot if the residuals versus the fitted values

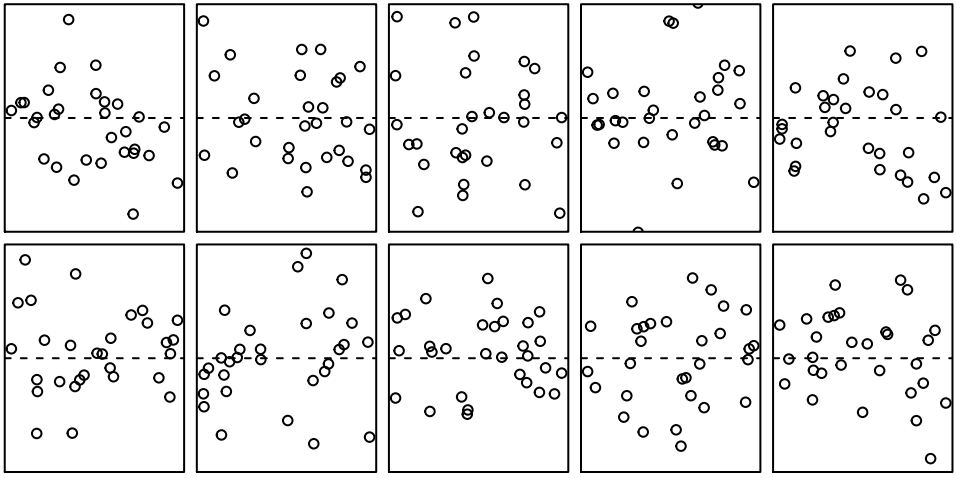
Equal Variance, n = 10

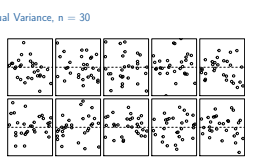




- on this and the next couple of slides are multiple residual plots where the residuals are sampled from a normal distribution with constant variance so that the equal variance condition is satisfied.
- inspect these plots to get an idea of what equal variances looks like
- we start with 10 samples of size 10, notice it's very hard to tell with small samples
- so unless the variances are quite different at different values of x, then assume equal variances

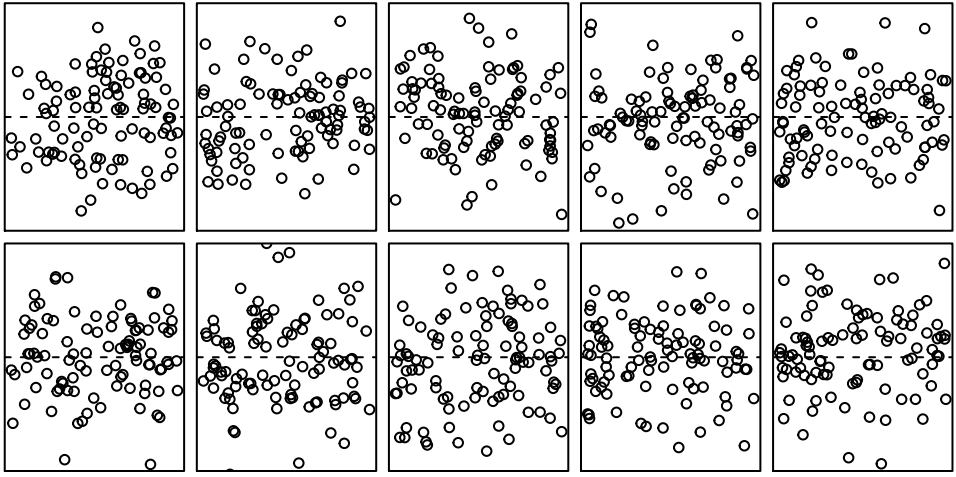
Equal Variance, n = 30

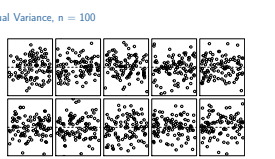




the variances are constant here too, it's easier to see with larger samples

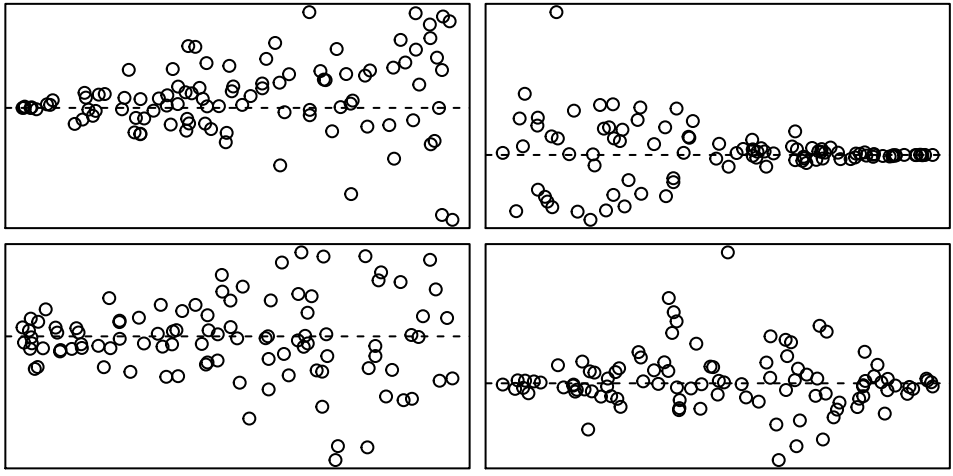
Equal Variance, n = 100

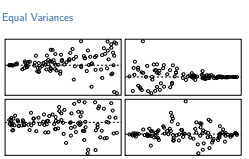




- no audio

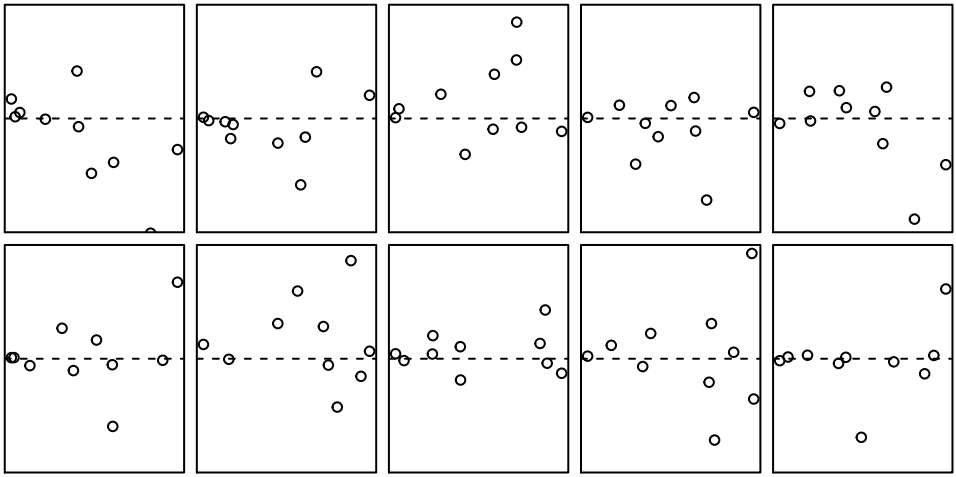
Not Equal Variances

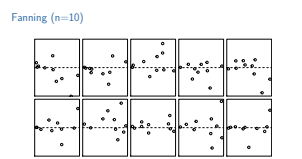




- Here are some examples of residual plots showing what unequal variances look like.
- The first three of these plots show residuals that exhibit "fanning"
- "fanning" is where the variance of the residuals increases, or decreases, as a function of the explanatory variable
- The fourth plot shows a diamond shape where the variance decreases as we move away from the middle
- these residual plots indicate data sets that violate the requirements for the simple linear regression model, while it's still reasonable to fit a line to the data, we shouldn't try statistical inference with the standard procedures discussed here.
- it might be possible to transform the data or use bootstrapping. WE won't cover bootstrapping for linear regression, but an example of transformation is at the end of these slides.

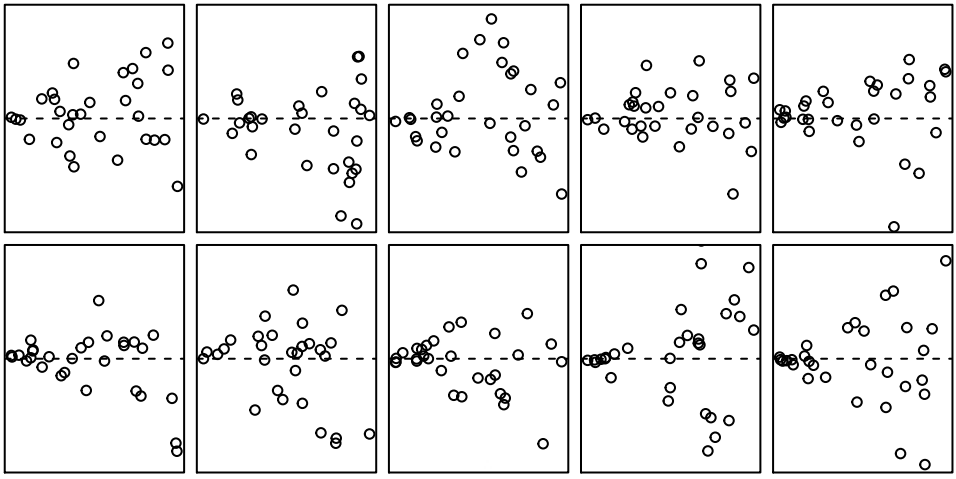
Fanning (n=10)

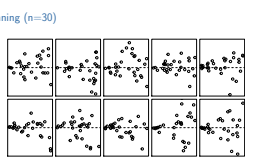




- again it can be very hard, for small samples, to see if the equal variances requirement is met, but these residual plots give you an idea of what it looks like
- the best thing to do is get more data!

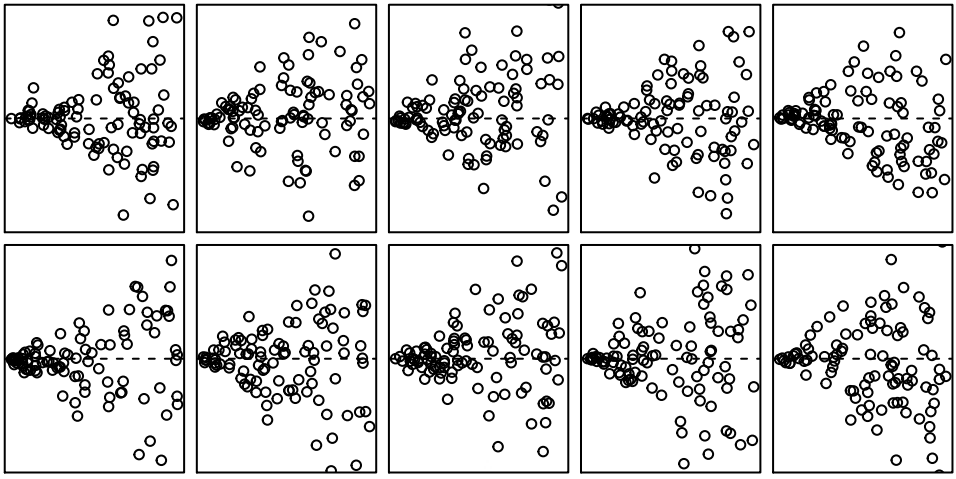
Fanning (n=30)

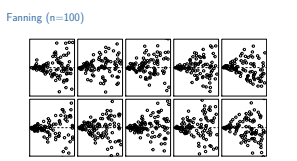




- no audio

Fanning (n=100)





- no audio

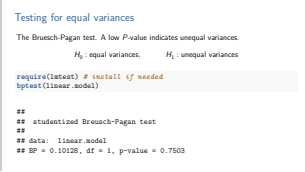
Testing for equal variances

The Bruesch-Pagan test. A low P-value indicates unequal variances.

$$H_0 : \text{equal variances}, \qquad H_1 : \text{unequal variances}$$

```
require(lmtest) # install if needed
bptest(linear.model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linear.model
## BP = 0.10128, df = 1, p-value = 0.7503
```


Testing for equal variances

The Bruesch-Pagan test. A low *P*-value indicates unequal variances.

H_0 : equal variances, H_1 : unequal variances

```
require(lmtest) # install if needed
bptest(linear.model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linear.model
## BP = 0.10128, df = 1, p-value = 0.7503
```

- there are several hypothesis tests one can use to determine if the equal variances condition is violated
- this one, called the Bruesch-Pagan test is described on page 800 of Ott's book.
- just like testing for normality, you should not rely on a test for unequal variances very heavily
- if you have a very small sample, the test won't have much power and will only detect severe unequal variances
- if you have a very large sample, the test will flag very slightly unequal variances as statistically significant even though they likely have little impact on the linear regression assumptions
- usually, a visual inspection of the residual plots is adequate, if there is a question, try to get more data

Independence of errors

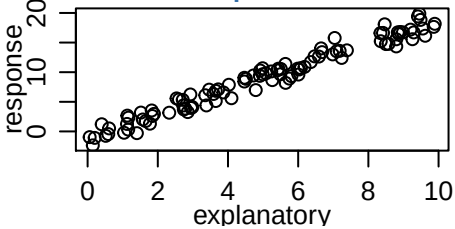
- Errors should have no dependence on order, time, or space
- Lack of independence includes:
  - clusters or patterns
  - serial correlation (order or time dependence)
  - spatial association
- Plots
  - residuals vs explanatory variable(s)
  - residuals vs order (and/or time)

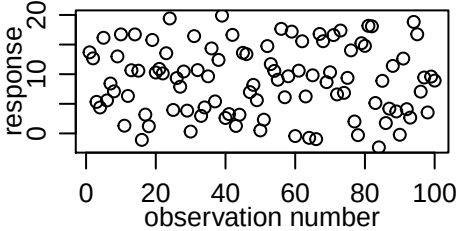
Independence of errors

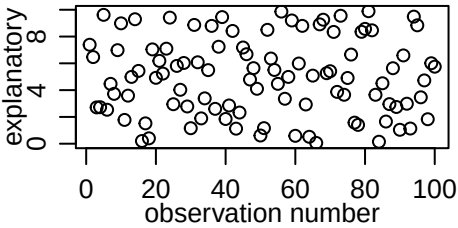
- Errors should have no dependence on order, time, or space
- Lack of independence includes:
  - clusters or patterns
  - serial correlation (order or time dependence)
  - spatial association
- Plots
  - residuals vs explanatory variable(s)
  - residuals vs order (and/or time)

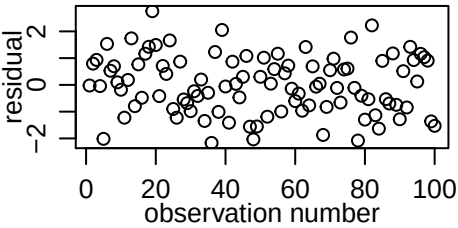
- we can use the residuals as surrogates for errors and inspect the residuals to see if there is reason to believe the residuals are dependent
- essentially, the residuals should not exhibit any dependence on the explanatory variables or on the order they were selected . . .

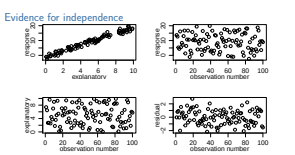
Evidence for independence





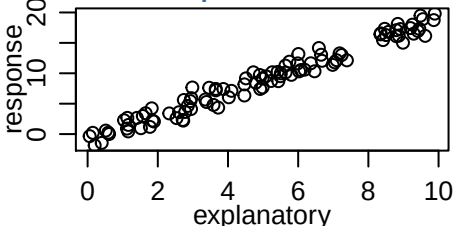


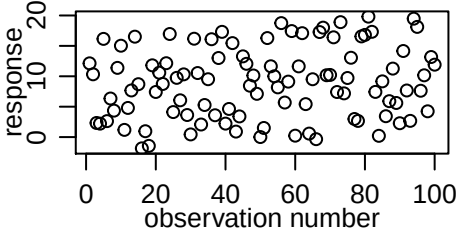


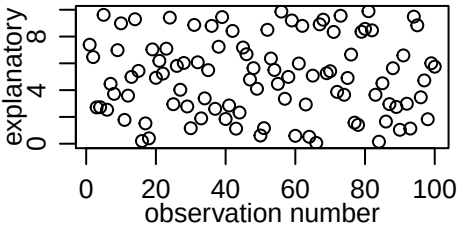


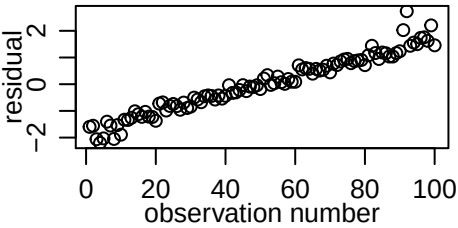
- the first plot of the response versus the explanatory variable shows a nice linear trend that we want to see for simple linear regression
- it also shows that residuals, the vertical deviations from the line, don't depend on x, of course you plot just the residuals versus x to verify that, but we haven't shown that plot here
- the next two plots show you that both the response and explanatory variables don't exhibit any dependence on the order of the observations. This doesn't have to be the case since it would be reasonable to put the x's in some order
- but the last plot, of residual versus x, shouldn't exhibit a different trend than either of plots 2 and 3 ... since x and y both don't depend on order, the residuals shouldn't depend on order either.
- these plots suggest that the independence of errors requirement is plausible for this data

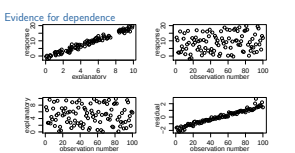
Evidence for dependence







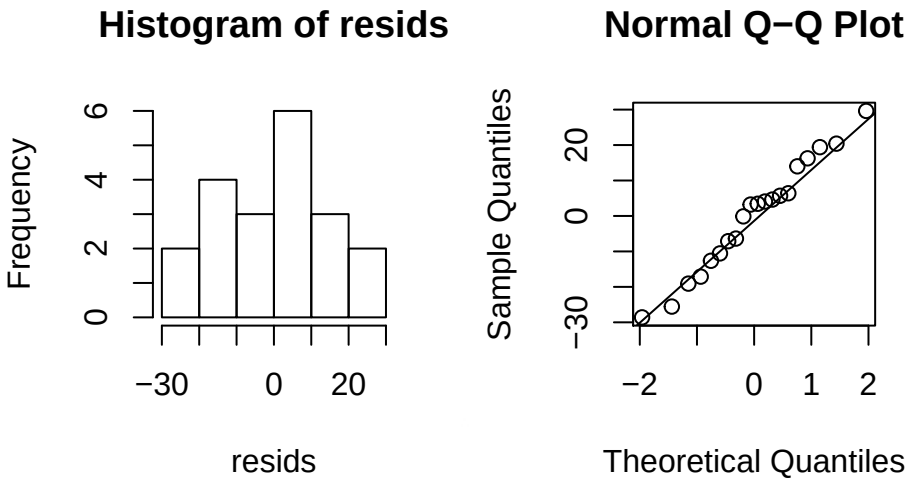




- these plots are just like the ones on the previous slide, but the fourth plot looks different
- notice that x and y are both independent of the observation number, but the residuals depend strongly on the order of the observations ... there is a relationship among the residuals so the requirement of independence may be violated and we should proceed with caution ... if the pairs were artificially sorted in order of increasing residuals, then there isn't a problem, but if the observations are in chronological order then that is a pattern of dependence we can't ignore in which case the requirements for the simple linear regression model are not met.

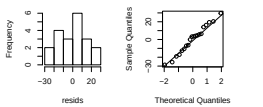
Normality of Error Distribution

```
par(mfrow=c(1,2)); hist( resids); qqnorm( resids); qqline( resids)
```



Normality of Error Distribution

```
par(mfrow=c(1,2)); hist( resids); qqnorm( resids); qqline( resids)
```



- the histogram and normal quantile plot both suggest that the residuals could be normally distributed so its plausible that the errors are normally distributed
- having verified that are data seems to be the simple linear regression model, we can continue on to doing statistical inference for this problem

What if requirements are violated?

Alternatives to the simple linear regression model include:

- nonparametric procedures based on rank
- bootstrapping
- Generalized Linear Model

Beyond the scope of this class . . .

What if requirements are violated?

Alternatives to the simple linear regression model include:

- nonparametric procedures based on rank
- bootstrapping
- Generalized Linear Model

Beyond the scope of this class . . .

- bootstrap estimates of the population slope, population intercept, and population mean response for a fixed x are actually pretty easy to get, bootstrapping the prediction intervals is more complicated, I'd love to show you all the bootstrap stuff because it's some of my favorite material, but we don't have time to do it all.
- the generalized linear model is a framework that allows
  - more complicated functions of the data and parameters using what are called link functions
  - different probability distributions for the error terms
  - makes it possible to model unequal variances

If the requirements are met

- then proceed to statistical inference using the classical methods described here and in the book

If the requirements are met

- then proceed to statistical inference using the classical methods described here and in the book

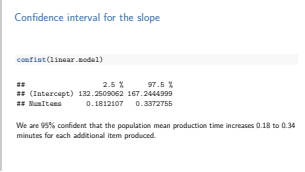
- no audio

Confidence interval for the slope

```
confint(linear.model)
```

```
##                   2.5 %      97.5 %
## (Intercept) 132.2509062 167.2444999
## NumItems      0.1812107   0.3372755
```

We are 95% confident that the population mean production time increases 0.18 to 0.34 minutes for each additional item produced.


Confidence interval for the slope

```
confint(linear.model)
```

```
##                 2.5 %     97.5 %
## (Intercept) 132.2509062 167.2444999
## NumItems      0.1812107   0.3372755
```

We are 95% confident that the population mean production time increases 0.18 to 0.34 minutes for each additional item produced.

- Usually we are most interested in the population slope, beta_1, whose confidence interval estimate is given on the second line,
- but if needed the CI for the population intercept, beta_0, is given on the first line
- the intercept here suggests that 132 to 167 minutes are required if no items are produced . . . that hardly seems reasonable, since all of the data is for production runs with 50 to 350 items produced, we shouldn't expect the model to give us meaningful information for 0 items produced . . . this is an example of extrapolation

Confidence interval for the correlation

```
with(production, cor.test( NumItems, Time)$conf.int )
```

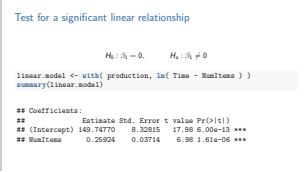
```
## [1] 0.6625316 0.9411514
## attr(,"conf.level")
## [1] 0.95
```

Test for a significant linear relationship

$$H_0 : \beta_1 = 0, \qquad H_a : \beta_1 \neq 0$$

```
linear.model <- with( production, lm( Time ~ NumItems ) )
summary(linear.model)
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 149.74770    8.32815   17.98 6.00e-13 ***
## NumItems      0.25924    0.03714    6.98 1.61e-06 ***
```



- this interval estimate is less useful in most applications, but just in case you need it you can
- see that is very easy to get a confidence interval estimate of the population correlation
- this verifies that is moderate or strong positive correlation between the number of items produced and the production time.

- as part of computing the linear model R does a hypothesis test to determine if the unknown population slope is different than 0
- The second line of the coefficients that begins with NumItems shows us
  - the estimated slope .25924
  - the standard error of the slope

Checking for practical significance (effect size)

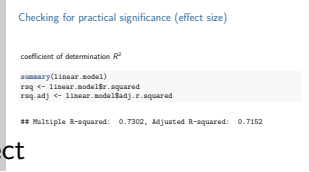
coefficient of determination R^2

```
summary(linear.model)
rsq <- linear.model$r.squared
rsq.adj <- linear.model$adj.r.squared
```

```
## Multiple R-squared:  0.7302, Adjusted R-squared:  0.7152
```

Linear Regression and Correlation

Checking for practical significance (effect size)



- $R^2 \approx$ proportion of total variation in y that is explained by the linear relationship with x.
- the adjusted R^2 is an unbiased estimate of the population coefficient of determination and is usually very similar to the regular R^2
- the adjusted R^2 suggests that about 72% of the total variation in production times is explained by the linear relationship between production time an the number of items produced. This also means that 28% of the variation is unexplained by the linear relationship.
- in this case the model has strong predictive power, but a low coefficient of determination might indicate that a model has little practical significance
- R^2 is often used as a standardized effect size for linear regression, its possible to have a tiny P-value indicating a statistically significant linear relationship and also have a small R^2 indicating that the linear relationship doesn't explain much ... the exact

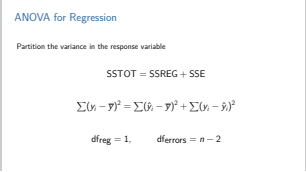
ANOVA for Regression

Partition the variance in the response variable

$$\text{SSTOT} = \text{SSREG} + \text{SSE}$$

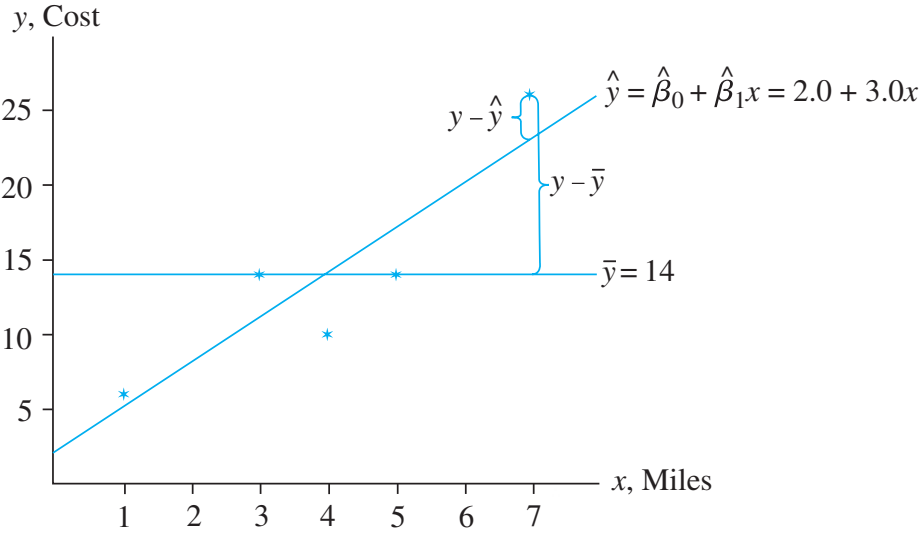
$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum (y_i - \hat{y}_i)^2$$

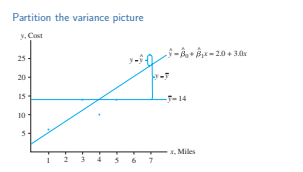
$$\text{df}_{\text{reg}} = 1, \qquad \text{df}_{\text{errors}} = n - 2$$



- the test for a significant linear relationship that was reported a few slides ago is based on the t-test statistic, but the F-test is equivalent in the case of simple linear regression
- for multiple regression, only the F-test works so its worth exploring here
- the underlying idea is important also ...
- in ANOVA analysis of multiple population means, the Sum Squares total variation in the response variable is partitioned into two components: the sum squares groups and the sum squares residuals
- in ANOVA analysis for linear regression the sum squares total variation in the response variable is partitioned into the sum squares regression which is similar to the sum squares groups from before and the sum squares residuals
- the next slide helps explain these two sources of variation

Partition the variance picture





make this into a video slide and show the two error components

ANOVA Table for Regression

Source	df	SS	MS	F	P-value
Regression	1	$SSREG$	$MSREG = \frac{SSREG}{1}$	$F_0 = \frac{MST}{MSE}$	$P(F_{1,n-2} > F_0)$
Error	$n-2$	SSE	$MSE = \frac{SSE}{n-2}$		
Total	$n-1$	$SSTOT$			

ANOVA Table for Regression

Source	df	SS	MS	F	P-value
Regression	1	$SSREG$	$MSREG = \frac{SSREG}{1}$	$F_0 = \frac{MST}{MSE}$	$P(F_{1,n-2} > F_0)$
Error	$n-2$	SSE	$MSE = \frac{SSE}{n-2}$		
Total	$n-1$	$SSTOT$			

- once you have the sum squares and the degrees of freedom the remainder of the anova table follows just like it did with our past ANOVA analysis
- the degrees of freedom for the sum squares is one less than the number of parameters in the model
- notice that sum-squares regression has only 1 degree of freedom, whenever the degrees of freedom for the numerator is 1 in an F distribution, the F test statistic is just the square of the t-test statistic
- in multiple regression there are more parameters in the model and the degrees of freedom for the numerator will be larger than one, so the F test statistic will no longer be equivalent to a t.

ANOVA for Regression Example

```
linear.model <- with( production, lm( Time ~ NumItems ) )
anova(linear.model)
```

```
## Analysis of Variance Table
##
## Response: Time
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## NumItems   1 12868.4 12868.4  48.717 1.615e-06 ***
## Residuals 18  4754.6   264.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA for Regression Example

```
linear.model <- with( production, lm( Time ~ NumItems ) )
anova(linear.model)
```

```
## Analysis of Variance Table
##
## Response: Time
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## NumItems   1 12868.4 12868.4  48.717 1.615e-06 ***
## Residuals 18  4754.6   264.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- here we have the anova table for the production example
- you might notice that the P-value is the same as it was, except for rounding, when we reported the t-test a few slides ago
- if you were to square the t-test statistic from before it would be the same as F here.

Confidence Interval for Population Mean Response

At a production level of 300 items, what is the average production time?

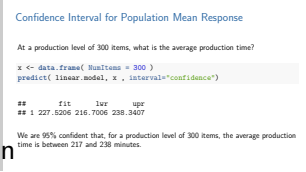
```
x <- data.frame( NumItems = 300 )
predict( linear.model, x , interval="confidence")
```

```
##        fit      lwr      upr
## 1 227.5206 216.7006 238.3407
```

We are 95% confident that, for a production level of 300 items, the average production time is between 217 and 238 minutes.

Linear Regression and Correlation

Confidence Interval for Population Mean Response



Confidence Interval for Population Mean Response

At a production level of 300 items, what is the average production time?

```
x <- data.frame( NumItems = 300 )
predict( linear.model, x , interval="confidence")
```

```
##        fit      lwr      upr
## 1 227.5206 216.7006 238.3407
```

We are 95% confident that, for a production level of 300 items, the average production time is between 217 and 238 minutes.

-no audio

Prediction Interval for New Observed Value of Response

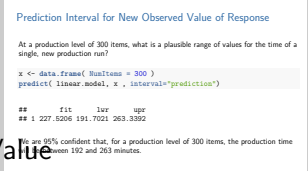
At a production level of 300 items, what is a plausible range of values for the time of a single, new production run?

```
x <- data.frame( NumItems = 300 )
predict( linear.model, x , interval="prediction")
```

```
##        fit       lwr       upr
## 1 227.5206 191.7021 263.3392
```

We are 95% confident that, for a production level of 300 items, the production time will be between 192 and 263 minutes.

Prediction Interval for New Observed Value of Response


Prediction Interval for New Observed Value of Response

At a production level of 300 items, what is a plausible range of values for the time of a single, new production run?

```
x <- data.frame( NumItems = 300 )
predict( linear.model, x , interval="prediction")
```

```
##        fit      lwr      upr
## 1 227.5206 191.7021 263.3392
```

We are 95% confident that, for a production level of 300 items, the production time is between 192 and 263 minutes.

- as discussed in your text book, the prediction interval is saying that if we added one more observation to the data set, with the x value of 300, then the e y-value will be between 192 and 263.
- the 95% says that for 95% of samples you'll get a prediction interval that actually contains the next observed response value
- the prediction interval is often more relevant than the confidence interval because the prediction interval is telling us what y-values we can expect to observe at a given level of x, whereas the confidence interval tells us the *average* value of y to expect.

Confidence Bands - the code

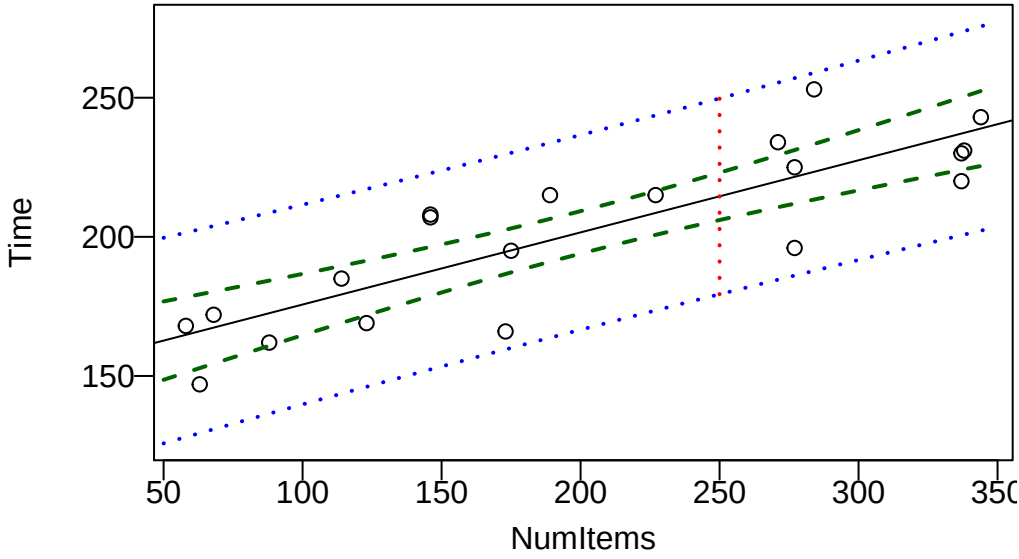
```
xplot <- data.frame( NumItems = seq( 50, 3, length=200) )
fittedC <- predict(linear.model,xplot,interval="confidence")
fittedP <- predict(linear.model,xplot,interval="prediction")

# scatterplot
ylimits <- c(min(fittedP[,"lwr"]),max(fittedP[,"upr"]))
plot(NumItems,Time,ylim=ylimits)
abline(linear.model)

# plot the confidence and prediction  bands
lines(xpts, fittedC[, "lwr"], lty = "dashed",col='darkgreen')
lines(xpts, fittedC[, "upr"], lty = "dashed",col='darkgreen')
lines(xpts, fittedP[, "lwr"], lty = "dotted",col='blue')
lines(xpts, fittedP[, "upr"], lty = "dotted",col='blue')
```

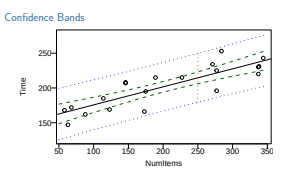
Confidence Bands - the code

```
xplot <- data.frame( NumItems = seq( 50, 3, length=200) )
fittedC <- predict(linear.model,xplot,interval="confidence")
fittedP <- predict(linear.model,xplot,interval="prediction")

# scatterplot
ylimits <- c(min(fittedP[,"lwr"]),max(fittedP[,"upr"]))
plot(NumItems,Time,ylim=ylimits)
abline(linear.model)

# plot the confidence and prediction  bands
lines(xpts, fittedC[, "lwr"], lty = "dashed",col='darkgreen')
lines(xpts, fittedC[, "upr"], lty = "dashed",col='darkgreen')
lines(xpts, fittedP[, "lwr"], lty = "dotted",col='blue')
lines(xpts, fittedP[, "upr"], lty = "dotted",col='blue')
```

- a nice summary plot is to show the scatterplot of the data,
- the fitted model
- and both confidence and prediction intervals as each value of x
- the OTT textbook has exactly this kind of plot on page 598.

Confidence Bands



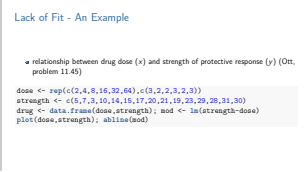


- to interpret this plot look at the vertical red line shown where the number of items is 250.
- the vertical red line crosses through the dashed green lines at the lower and upper 95% confidence limits for the population mean production time when 250 items are produced
- the vertical red line intersects the dotted blue lines at the lower and upper 95% prediction limits for the value of a new Production time at a production level of 250 minutes.
- notice how the width of the intervals gets larger as we move away from the center of the plot
- think of this as an extrapolation penalty, the uncertainty increases as we move away from the middle of the data, this is reflected in the term containing the sample mean x value in the formulas for the standard error

Lack of Fit - An Example

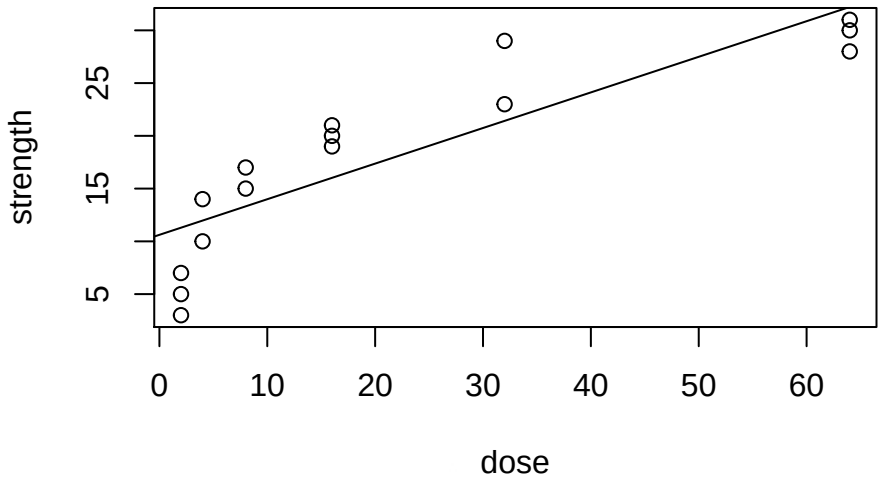
- relationship between drug dose (x) and strength of protective response (y) (Ott, problem 11.45)

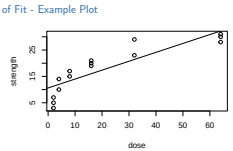
```
dose <- rep(c(2,4,8,16,32,64),c(3,2,2,3,2,3))
strength <- c(5,7,3,10,14,15,17,20,21,19,23,29,28,31,30)
drug <- data.frame(dose,strength); mod <- lm(strength~dose)
plot(dose,strength); abline(mod)
```



- Ott describes the lack of fit test on pages 602 and 603
- this test is a special case of a general F test that is used for comparing nested linear models

Lack of Fit - Example Plot





- in this example we are trying the predict the strength of response to a drug from the dose of the drug
- visually we can see that while the trend is for the strength of response to increase as the dose increases, but it isn't exactly linear
- in this case the strength is increasing, but the rate of increase is decreasing as the dose increases
- in other cases the lack of fit is more subtle, see the example on page 604 of Ott.

RSquare doesn't tell the whole story

```
drug.model <- with( drug, lm( strength ~ dose ) )
summary(drug.model)
```

```
## Multiple R-squared:  0.7581, Adjusted R-squared:  0.7394
```

RSquare doesn't tell the whole story

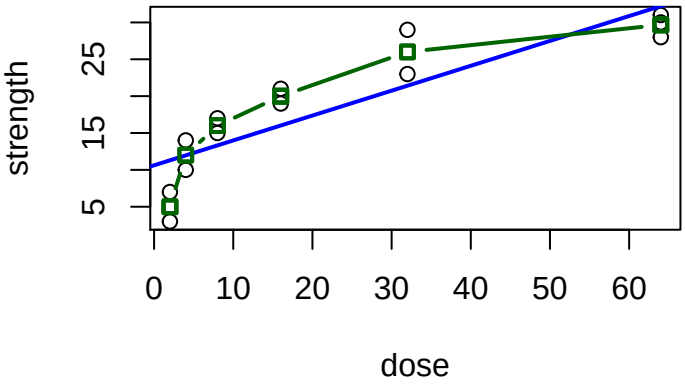
```
drug.model <- with( drug, lm( strength ~ dose ) )
summary(drug.model)
```

```
## Multiple R-squared:  0.7581, Adjusted R-squared:  0.7394
```

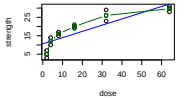
- The R-squared and adjusted R-squared both suggest that about 74 to 76 percent of the overall variation in strength of response is due to the linear relationship with dose,
- but visually we see that there should be a better model than a straight line . . . we don't really need the lack of fit test to tell that here, but it isn't always so obvious
- we'll go ahead with the lack of fit test just so we can see how it works in R

The Lack of Fit F-test

- requires some x values to have multiple observed y values
- compares linear model to a "full" model that fits through mean of each group



The Lack of Fit F-test
- requires some x values to have multiple observed y values
- compares linear model to a "full" model that fits through mean of each group



- we can only use this test if we have at least a few values of x for which multiple values of y are recorded similar to what we have in this drug study
- essentially we're going to compare the variation of the residuals about the line to the variation within each group
- loosely, we do this by fitting a model that goes through the mean of each group and comparing the residuals from the linear model to the residuals from the full model

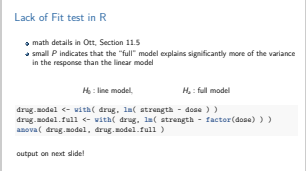
Lack of Fit test in R

- math details in Ott, Section 11.5
- small P indicates that the "full" model explains significantly more of the variance in the response than the linear model

H_0 : line model, H_a : full model

```
drug.model <- with( drug, lm( strength ~ dose ) )
drug.model.full <- with( drug, lm( strength ~ factor(dose) ) )
anova( drug.model, drug.model.full )
```

output on next slide!



- this is a case where R makes something very simple that seems complicated on paper
- find the linear model based on the least squared line
- find another linear model which fits to the mean of each group as in an ANOVA for multiple means, that's represented by the green curve here . . . note, linear model refers to how the unknown coefficients appear in the model and not necessarily to a straight line
- finally, use the general f-test, implemented in ANOVA to compare the two models. this works as long as the models are nested . . . in this case the linear model assumes that the population means are related by the line equation so this is a special case of the more general model that assumes no special relationship between the means, that is drug.model is nested inside of drug.model.full
- the general F test can be used to compare any pair of nested models

Lack of Fit test in R

```
## Analysis of Variance Table
##
## Model 1: strength ~ dose
## Model 2: strength ~ factor(dose)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     13 284.947
## 2      9  42.667  4    242.28 12.777 0.0009388 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Small $P \Rightarrow$ linear model not a good fit.
- Too much response variance not captured by the model.

Lack of Fit test in R

```
## Analysis of Variance Table
##
## Model 1: strength ~ dose
## Model 2: strength ~ factor(dose)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     13 284.947
## 2      9  42.667  4   242.28 12.777 0.0009388 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

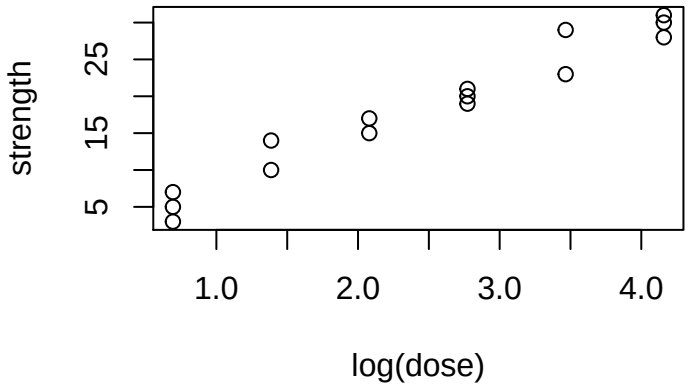
- Small $P \Rightarrow$ linear model not a good fit.
- Too much response variance not captured by the model.

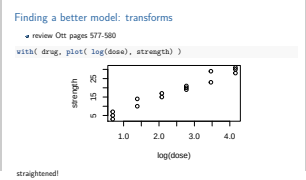
- the tiny P value confirms what we already knew, the full model fits better, or equivalently, there is a lack of fit the simple line model
- we can develop a more complex model or perhaps find a simple model which fits better
- be careful with more complex models, a very complex model with many parameters might fit the data perfectly but still be useless for making predictions for different values of x, this is called overfitting

Finding a better model: transforms

- review Ott pages 577-580

```
with( drug, plot( log(dose), strength) )
```





- Ott discusses when to try different transformations, but since the data looks like a square root or logarithmic curve, those are both reasonable things to try
- this shows the the logarithm of the dose and the strength of response are about as linear as possible

Fitting the transformed model

```
drug.model.logx <- with( drug, lm( strength ~ log(dose) ) )
(b0 <- drug.model.logx$coef[1])
```

```
## (Intercept)
##   0.9650838
```

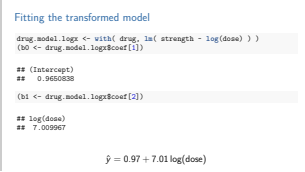
```
(b1 <- drug.model.logx$coef[2])
```

```
## log(dose)
##  7.009967
```

$$\hat{y} = 0.97 + 7.01 \log(\text{dose})$$

Linear Regression and Correlation

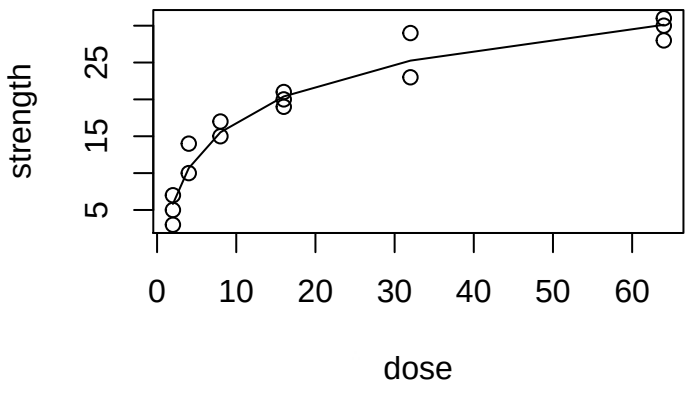
Fitting the transformed model



- no audio

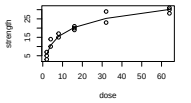
Transformed model plot

```
with( drug, plot( dose, strength) )
points( dose, b0 + b1* log(dose), type = 'l')
```



Transformed model plot

```
with( drug, plot( dose, strength) )
points( dose, b0 + b1* log(dose), type = '1')
```



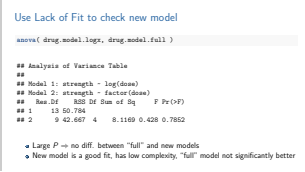
- no audio

Use Lack of Fit to check new model

```
anova( drug.model.logx, drug.model.full )
```

```
## Analysis of Variance Table
##
## Model 1: strength ~ log(dose)
## Model 2: strength ~ factor(dose)
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1     13 50.784
## 2      9 42.667  4    8.1169 0.428 0.7852
```

- Large $P \Rightarrow$ no diff. between "full" and new models
- New model is a good fit, has low complexity, "full" model not significantly better


Use Lack of Fit to check new model

```
anova( drug.model.logx, drug.model.full )
```

```
## Analysis of Variance Table
##
## Model 1: strength ~ log(dose)
## Model 2: strength ~ factor(dose)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     13 50.794
## 2      9 42.667  4    8.1169 0.428 0.7852
```

- Large $P \Rightarrow$ no diff. between "full" and new models
- New model is a good fit, has low complexity, "full" model not significantly better

- you can transform any data to straighten it and/or make the variances the same for all values few the explanatory variable
- the lack of fit test, which we used here, only applies if the there are multiple response values for at least a few values of the explanatory variable, the more repeated measurements, the more powerful the test.