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

Explore Relationships and Make Predictions

Manufacturing example: producing more items requires more time

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

- Model the relationship.
- Make predictions.

Sample Data

```r
head(production)
```

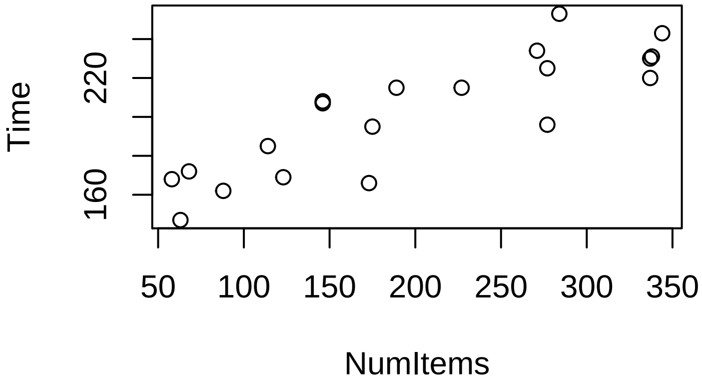
```
##   NumItems Time
## 1      175  195
## 2      189  215
## 3      344  243
## 4       88  162
## 5      114  185
## 6      338  231
```

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

Plot the data

Always start with a scatterplot:

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



Correlation

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

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

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

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

Pearson correlation

Desired Model

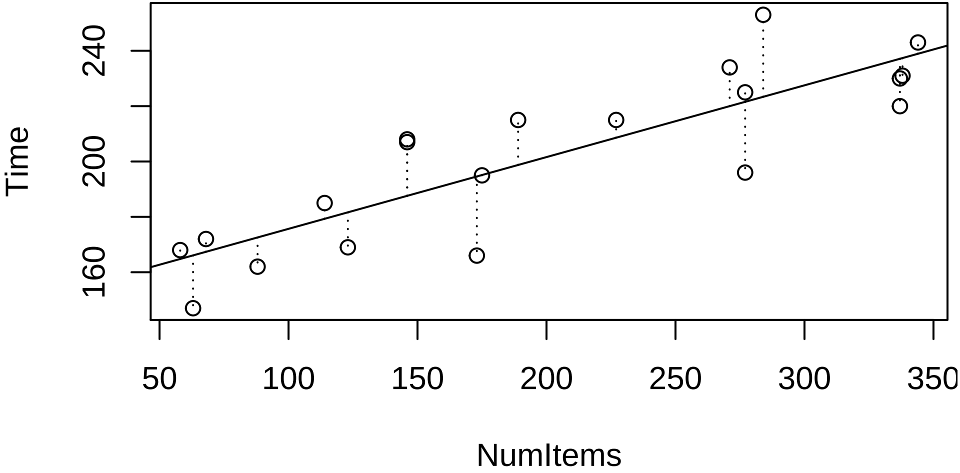
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

- 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

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

Residuals



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

Least Squares Regression Concept

Insert video here

Add clickable link in lower box

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

Finding the model in R

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

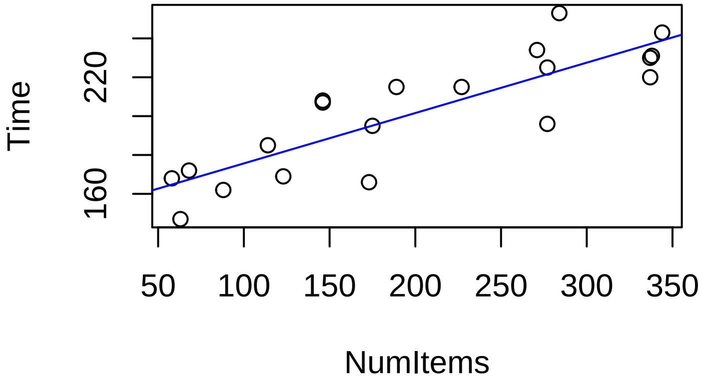
```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 149.74770    8.32815   17.98 6.00e-13 ***
## NumItems      0.25924    0.03714    6.98 1.61e-06 ***
```

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

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

Plotting the least-squares line

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



Extracting Coefficients

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

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

```r
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

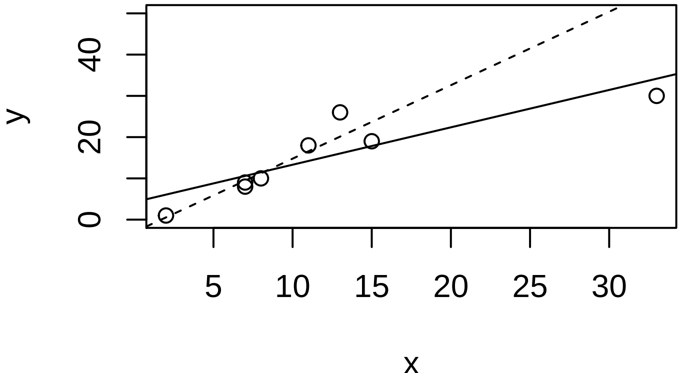
Making Predictions (2)

```r
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
```

Outliers and Influential Observations

```r
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')
```



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

Simple Linear Regression Model

Simple Linear Regression:

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

$$x = \text{ explanatory, independent, predictor }, y = \text{ response, depdendent}$$

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.

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.

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.

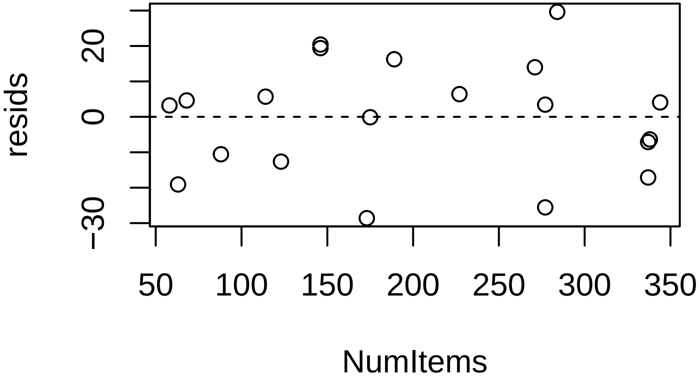
To make things simpler extract all the info. first:

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

Equal Variances

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

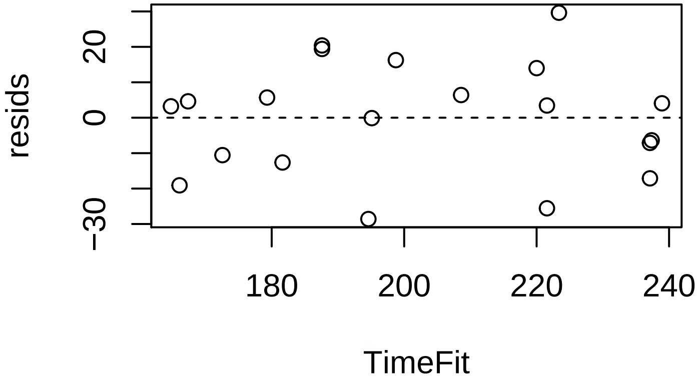
```
plot(NumItems,resids); abline(h=0,lty='dashed')
```



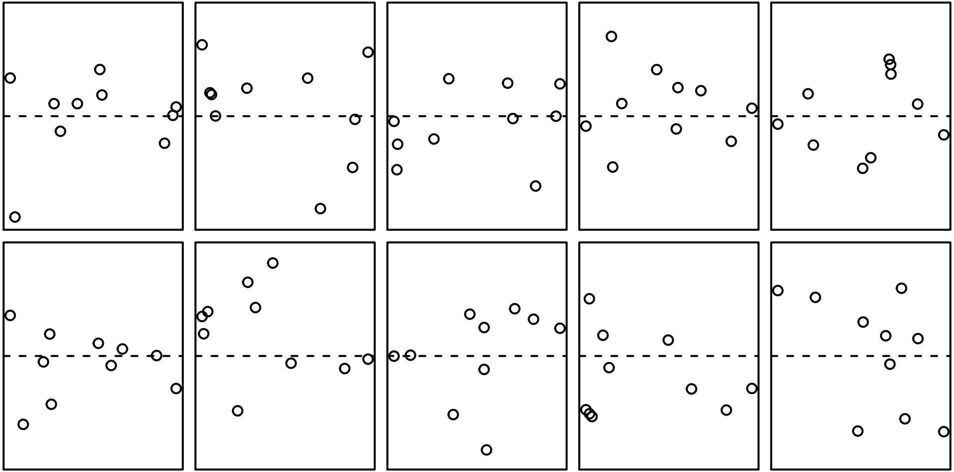
Equal Variance (2)

Equivalently, we can plot the residuals versus the fitted values

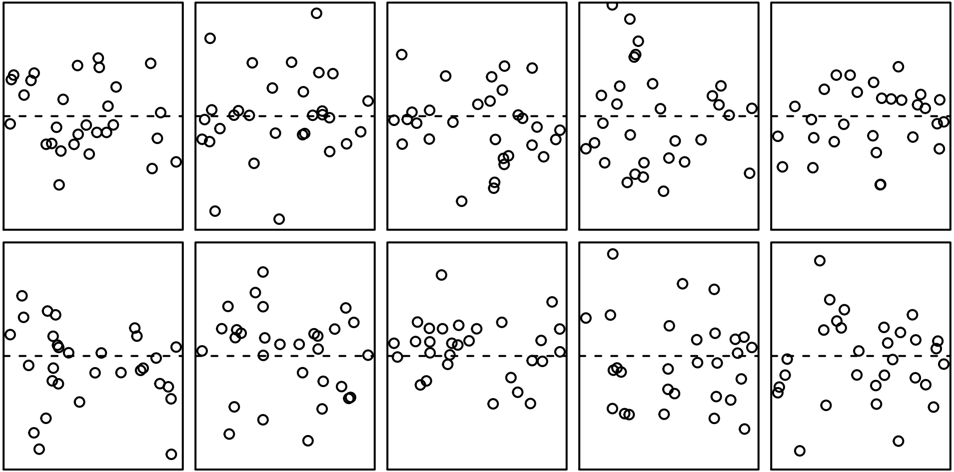
```
plot(TimeFit, resids); abline( h=0, lty='dashed')
```



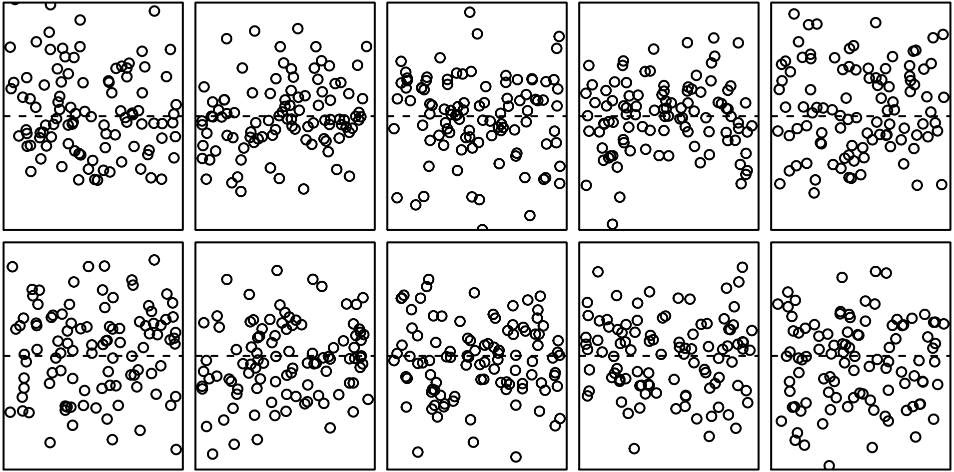
Equal Variance, n = 10



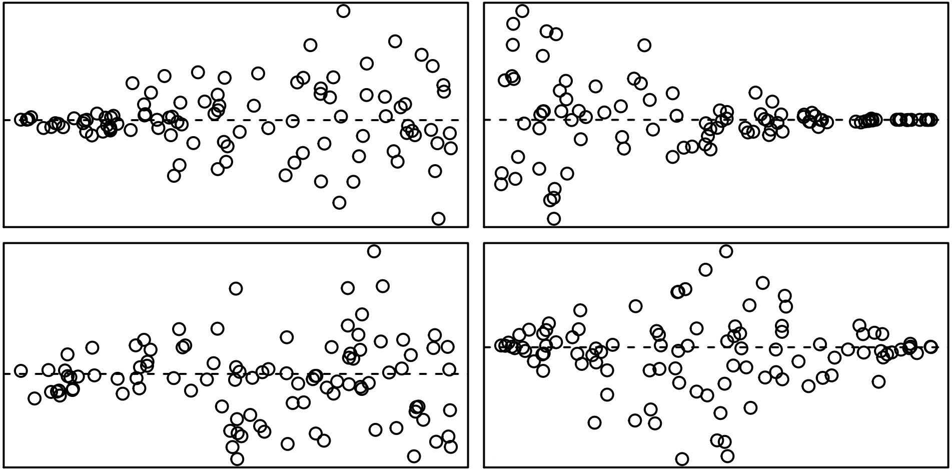
Equal Variance, n = 30



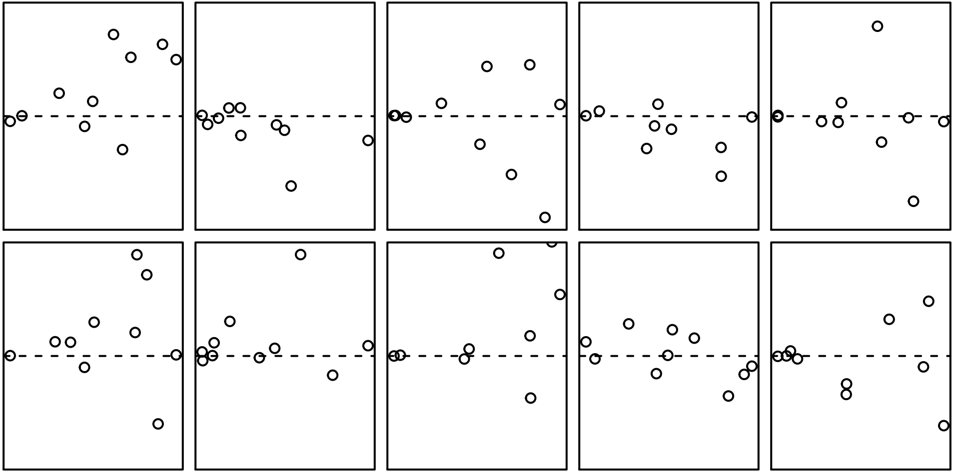
Equal Variance, n = 100



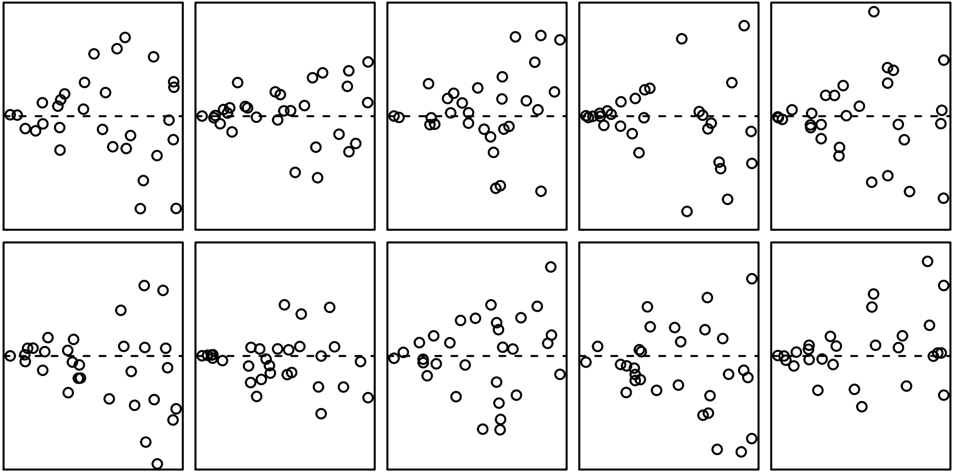
Not Equal Variances



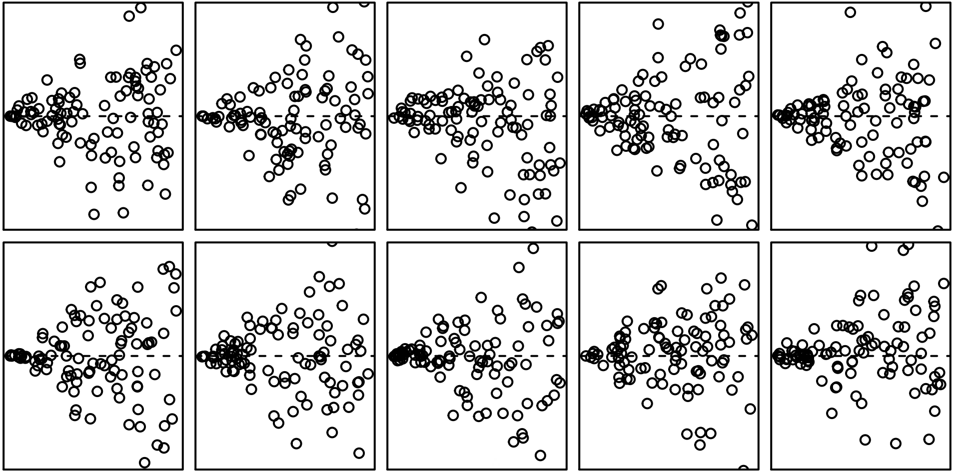
Fanning (n=10)



Fanning (n=30)



Fanning (n=100)



Testing for equal variances

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

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

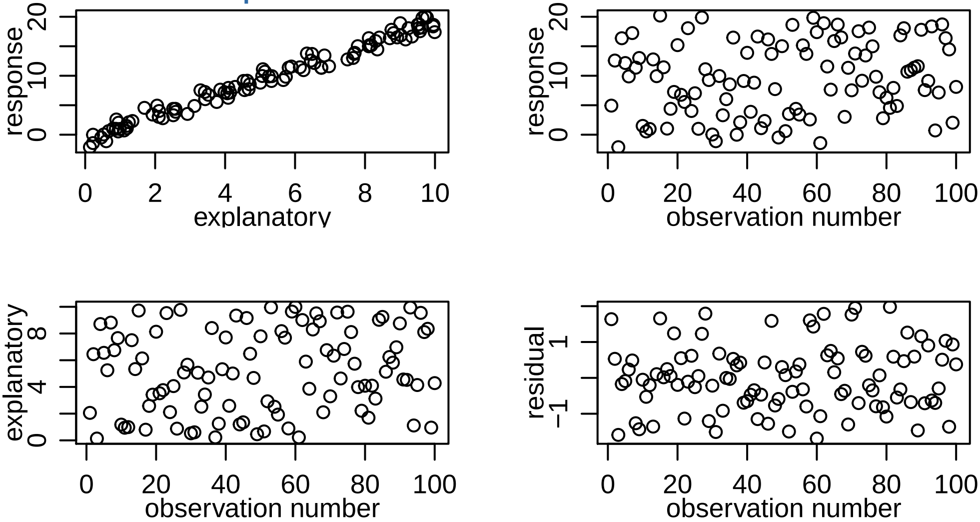
```
require(lmtest) # install if needed
bptest(linear.model)
```

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

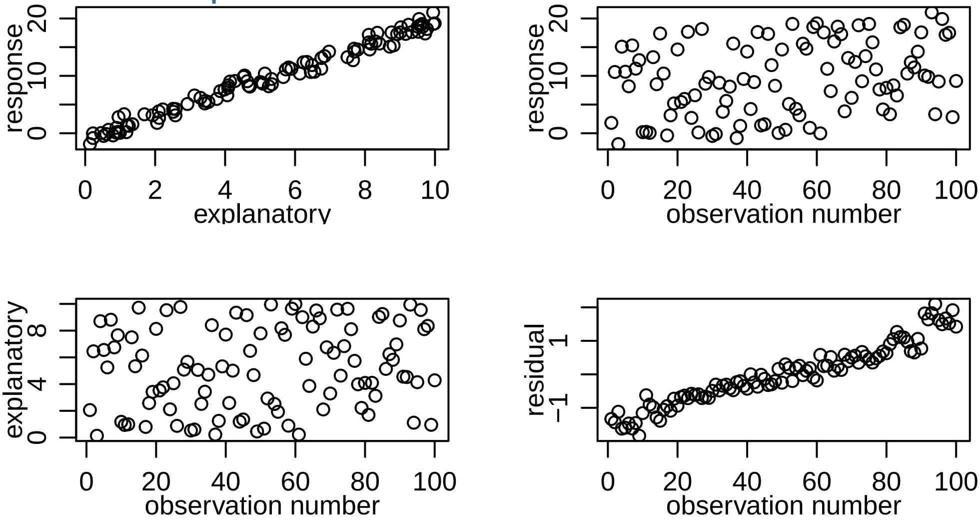
Independence of errors

- 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)

Evidence for independence

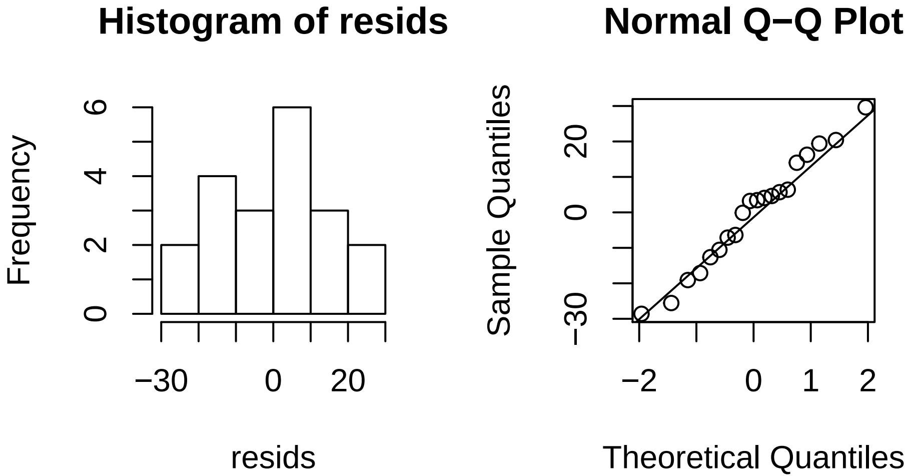


Evidence for dependence



Normality of Error Distribution

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



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 ...

If the requirements are met

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

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 correlation

```r
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

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

```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 ***
```

Checking for practical significance (effect size)

coefficient of determination R^2

```r
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
```

ANOVA for Regression

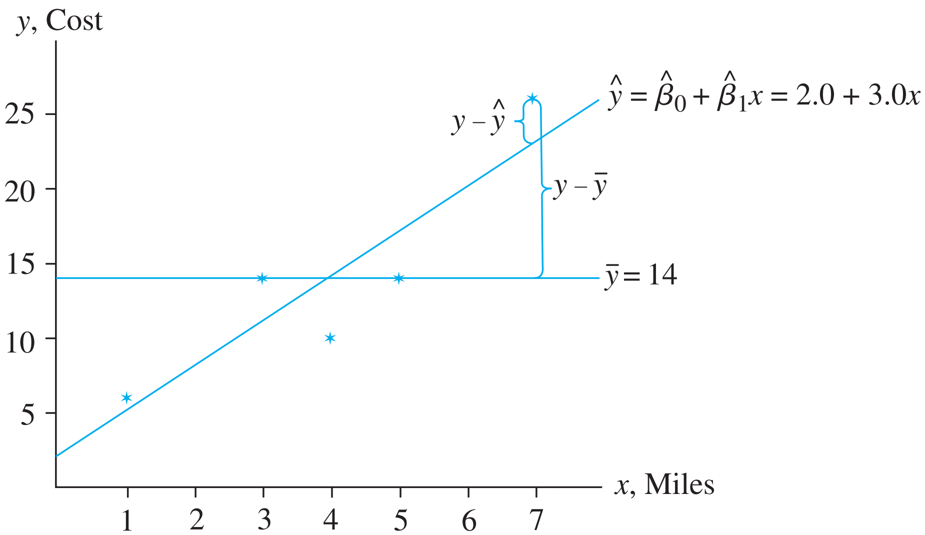
Partition the variance in the response variable

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

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

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

Partition the variance picture



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 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
```

Confidence Interval for Population Mean Response

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

```r
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.

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.

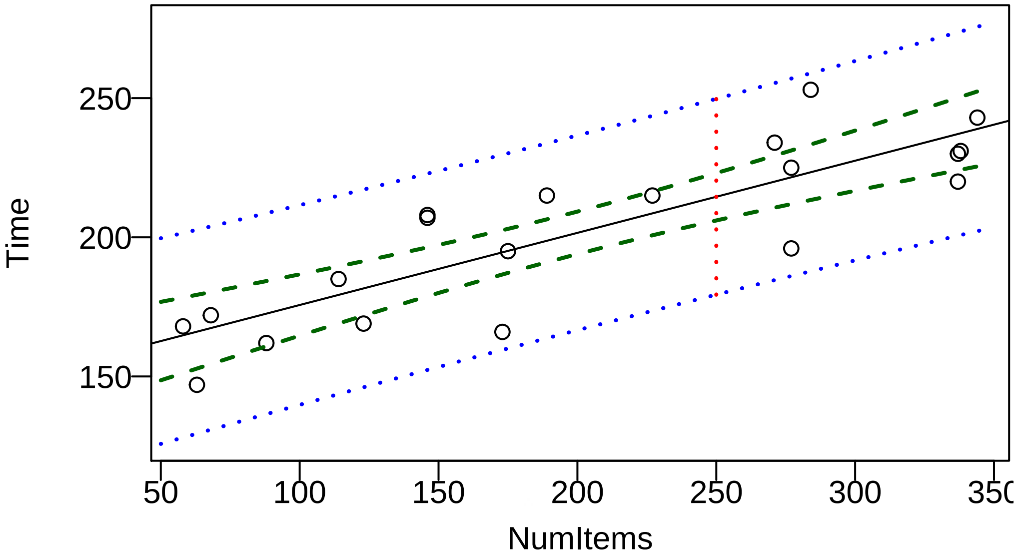
Confidence Bands - the code

```
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

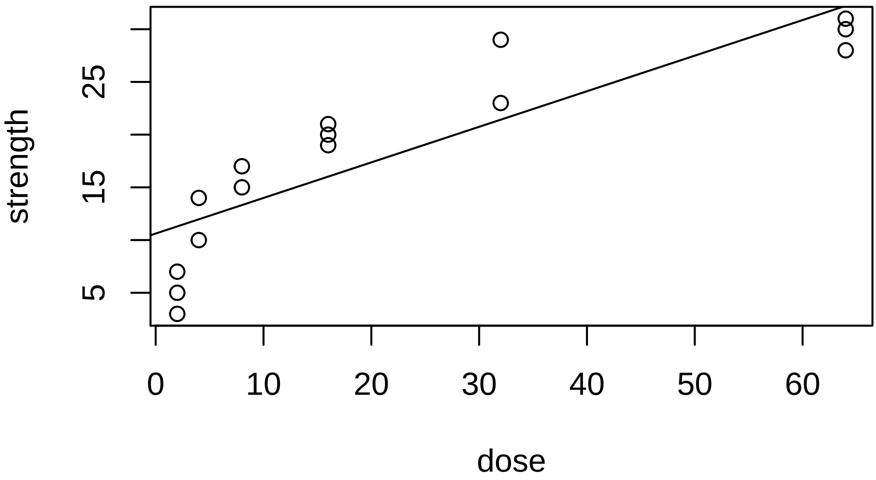


Lack of Fit - An Example

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

```r
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)
```

Lack of Fit - Example Plot



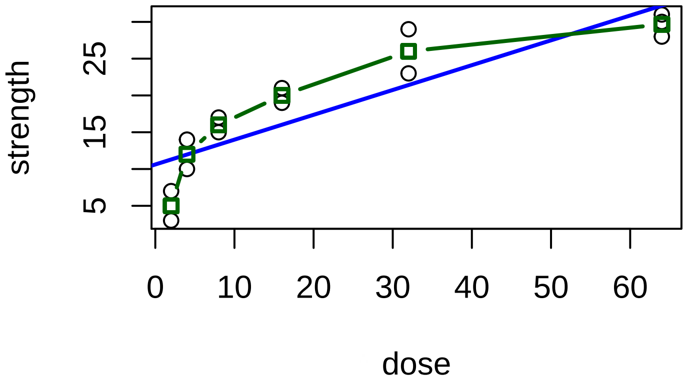
RSquare doesn't tell the whole story

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

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

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



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

$$H_0 : \text{line model}, \qquad H_a : \text{full model}$$

```r
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!

Lack of Fit test in R

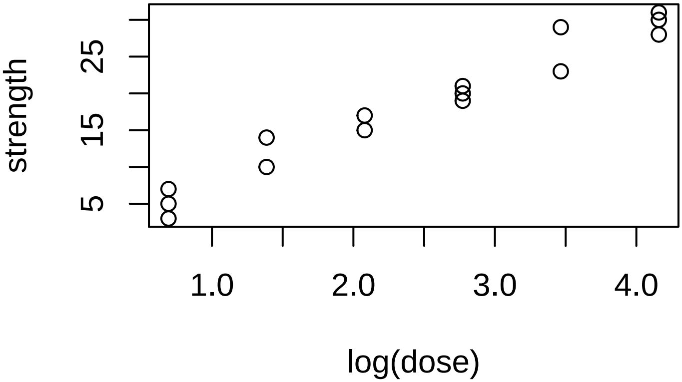
```
## Analysis of Variance Table
##
## Model 1: strength ~ dose
## Model 2: strength ~ factor(dose)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     13 284.947
## 2      9  42.667  4    242.28 12.777 0.0009388 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

Finding a better model: transforms

- review Ott pages 577-580

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



Fitting the transformed model

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

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

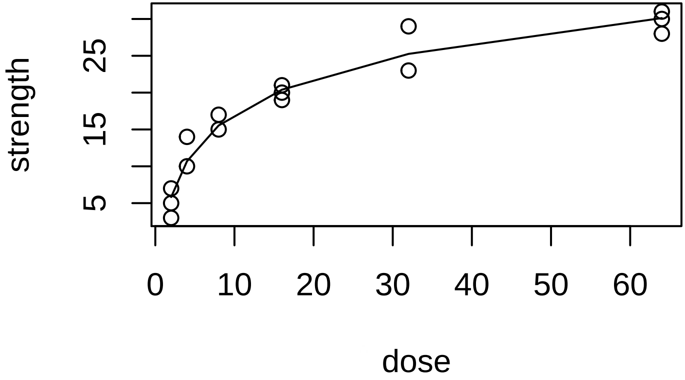
```r
(b1 <- drug.model.logx$coef[2])
```

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

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

Transformed model plot

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



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