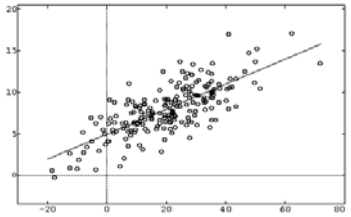
# Regression

## Regression

- Dependent=endogenous=y continuous
- Independent=exogenous=x=explanatory continuous
- y=a+bx
- We're doing statistics now=need error model
  - y=a+bx+ε where ε is ~N(0,σ)

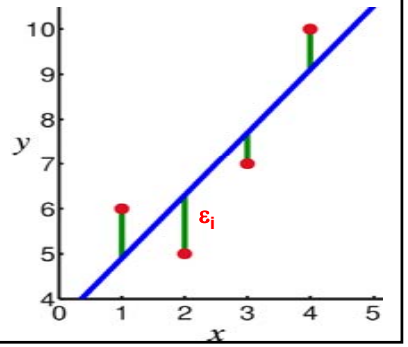
## Regression



- Is overdetermined
  - 3 variables to estimate (a,b, σ)
  - Usually dozens of data points
- Regression of 2 points
  - a,b→line going through pionts
  - 3 points estimate sigma too
- What criteria to use?

## Least squares

- Gauss 1794
- $\Sigma\varepsilon_i$ ?
  - But + & - cancel
- $\Sigma|\varepsilon_i|$?
  - But not nice math
- $\Sigma\varepsilon_i^2$
  - Bingo!



## Solving OLS

- $y=a+bx+\varepsilon$
- $\varepsilon = y-(a+bx)$
- $\Sigma\varepsilon^2 = \Sigma\,(y-(a+bx))^2 = \Sigma y^2-2y(a+bx)+(a+bx)^2$
  - $=\Sigma\, y^2-2ya-2ybx+a^2+2abx+(bx)^2$
- Minimize $\varepsilon^2$ wrt $\beta$ (take derivative, set to zero, solve)
- $d\Sigma\varepsilon^2/db = -\Sigma 2yx+\Sigma 2ax+\Sigma 2bx^2=0 \rightarrow \Sigma x[y-a+bx]=0$
  - → $b=cov(x,y)/var(x)$
- $d\Sigma\varepsilon^2/da=-\Sigma 2y+\Sigma 2a+\Sigma 2bx=0$
  - $\Sigma y-Na-\Sigma bx=0$ → $Na=\Sigma y-\Sigma bx \rightarrow a=\bar{y}-b\bar{x}$ → goes through $(\bar{x},\bar{y})$

## A second justification

- $P\{y|x\}=P\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4,\ldots\}$
  - $=P\{\varepsilon_1\}P\{\varepsilon_2\}P\{\varepsilon_3\}P\{\varepsilon_4\}\ldots$
- $\varepsilon$ is $\sim N(0,\sigma)$
  - $\sim c\exp(-k(y_i-a-bx_i)^2/\sigma^2)$
- So $P\{y|x\}=c\exp(-k(y_i-a-bx_i)^2/\sigma^2)*c\exp(-k(y_i-a-bx_i)^2/\sigma^2)*\ldots$
  - $=c^N\exp(-\Sigma k(y_i-a-bx_i)^2/\sigma^2)$
- So $\ln P\{y|x\}=\ln [c^N \Pi\exp(-k(y_i-a-bx_i)^2/\sigma^2)]$
  - $=N\ln c -\Sigma k(y_i-a-bx_i)^2/\sigma^2$
- Fortunately maximizing P same as maximizing ln P
  - Take derivative with respect to a, b and set to zero
  - Constants fall out – get least squares!
- Least squares is maximum probability! Assuming $\varepsilon$
  - Normal
  - Independent
  - Constant $\sigma$

## Results summary

- b=cov(x,y)/var(x)
  - b=cor(x,y)std(y)/std(x)
  - (cor(x,y)=cov(x,y)/std(x)std(y))
- a, b each have a t-distribution

## Sum of squares

- $SS_{resid} = \sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2$ = unexplained variance
- $SS_{model} = \sum_{i=1}^{n}\left(\hat{y}_i-\bar{y}\right)^2$ = explained variance
- $SS_{total} = \sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2$ =n Var(y)

- $SS_{total} = SS_{resid} + SS_{model}$ !
- $r^2 = SS_{model}/SS_{total}$ = % variance explained
  - $SS_{resid}/SS_{total} = 1-SS_{model}/Ss_{total} = 1-$% unexplained
- In linear univariate case $SS_{model}/SS_{total} = r^2 = cor(x,y)^2$ !

## Hypothesis testing

- b=0
  - Use t-test with 95% confidence interval around estimated b
- Model explains more than residuals
  - $(SS_{model}/1) / (SS_{resid}/(n-2)) >> 1$
  - F-test with 1,n-2 degrees freedom
    - Recall F is ratio of sums of normals!
- For univariate linear case two are equal
  - More generally first tests 1 coefficient, 2nd tests whole model

## Assumptions of OLS

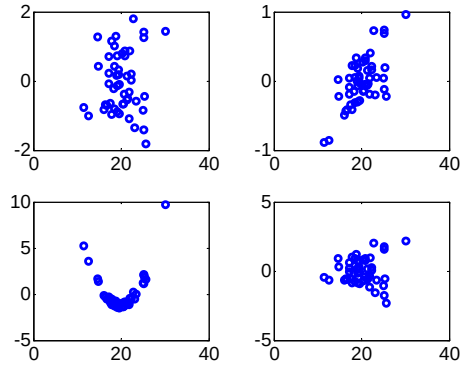
1. Structural:
   - Y is continuous
   - Y depends on 1 variable x
   - Linear relationship
2. Normality: $\varepsilon_i \sim N(0,\sigma_i)$
3. Independence: $Cov(\varepsilon_i,\varepsilon_j)=0$
4. Homoscedasticity: $\sigma_i=\sigma_j$
5. (Non-collinearity: $Cov(x_i,x_j)=0$)

## Residual plots for regression

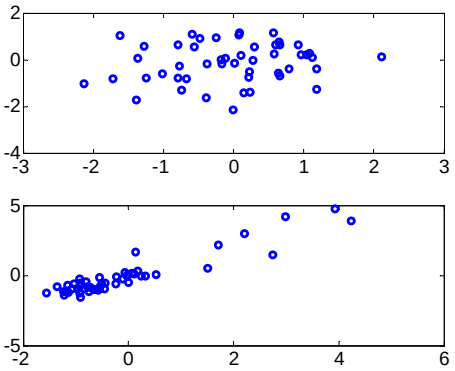
Check 3 other assumptions
- Independence of error terms
- Homoscedasticity
- Linear model appropriate

- Two main plots
  - $\varepsilon_i$ vs. $y_i$
    - Detect heteroscedasticity, nonlinearity, some independence
  - $\varepsilon_i$ vs. $\varepsilon_{i-1}$
    - Detect independence
    - Also Durbin-Watson statistic
- Also
  - $\varepsilon_i$ vs. time of collection, collateral variables

## Sample residual plots ($\varepsilon_i$ vs. $y_i$)



## Resid vs. Resid ($\varepsilon_i$ vs $\varepsilon_{i-1}$)




## If heteroscedastic

- ANOVA is robust *if* design is nearly balanced
- Regression moderately robust, but one side of regression figures more heavily into estimates - bias
- Often a good transformation will fix the problem

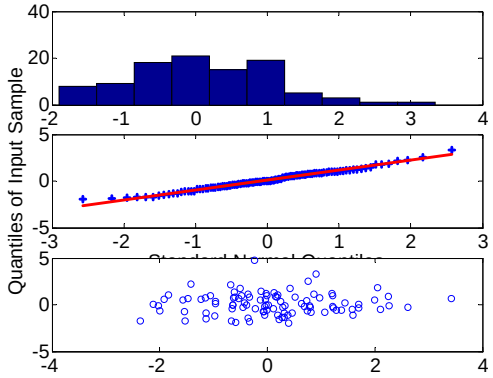
## Transformation

- Can apply to dependent or independent
- Common transformations:
  - Y'=log(y) – extremely common in biology
    - Body mass, abundance, etc
    - Many stats programs use ln, but $\log_{10}$ easier to plot
  - Y'=sqrt(y) – if expect y to be a Poisson count
  - Y'=1/Y (reciprocal) – common for rates (e.g. # offspring/female)
  - Y'=arcsin(sqrt(y)) – for proportions (0-1)
  - Box-Cox: $y'=(y^{\lambda}-1)/\lambda$ matches 3 above if $\lambda$=0, ½, -1) – most stats packages can calculate "best" $\lambda$
- Report backtransformed parameters

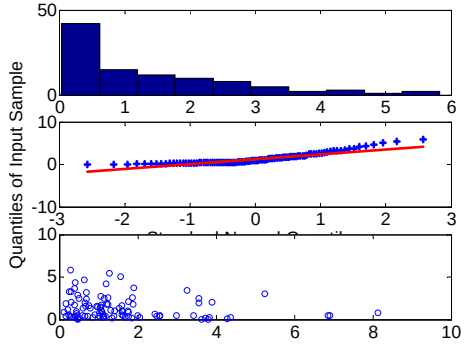
## Diagnostics for normality

- Histogram plot of residuals
- QQPlot (quantile-quantile plot)
- Residual plots
  - Should be a symmetric oval with thinning at edges – w/ 20-30+ points edges should be well traced
- Applies to ANOVA & regression

## Normal plots




## Non-normal (needs log transform)




## If non-normal

- As long as you have a peak with reasonable symmetry, you're probably OK – very robust
- If data is heavily skewed
  - Non-paremetric tests
  - Transform the data (e.g. log)
  - GLIM

## Outliers

- The normal distribution assumes tails are "small" – extreme outliers (~3+ standard deviations out) "shouldn't" occur
- Can heavily skew estimates
- Detecting outliers
  - Obvious in residual plots
    - Numeric lists as well
  - Outside the whiskers in Box plots
  - Also calculate leverage or influence
    - Degree of effect on regression slope

## What to do about outliers

- Revisit the paper trail for that data point
  - Most often a data entry or other human error
  - Simply correct
- Revisit notes about that site/experiment/data point
- Remove if:
  - Has high leverage and care about estimating parameters
  - Obvious experimental issue
- **If you remove – it is grossly unethical to fail to report this – OK to report and explain why**
- A priori filter criteria can also be helpful

## Regression summary

- OLS = Ordinary Least Squares
  - Minimize sum of squares to fit line through cloud
  - = maximum probability
  - Simple equations for a, b
  - Sum of squares → variance partitioning
  - Two null hypotheses converge in this case (t-test vs. ANOVA)
- 5 assumptions
  - Test with QQ plots, histograms & residual plots
  - Transform can help meet requirements

## R Introduction

## R basics

- A simple calculation
  - storagename<-expression;
  - x<-3+4
  - Inf NaN # not a number
- Building vectors
  - V<-c(1,3,4) #vector
  - V<-seq(lo,hi,step) v<-(length=n, from=lo, by=step)
  - V<-rep(3,times=10)
- Boolean vectors
  - V>3
  - V>3 & v<10
  - v1[v2>3] # picks out elements of v that are true
- Statistical vectors
  - v=ordered(v,levels=c("first","second","third"))
  - v=factor(v) #unordered discrete

## Making R repeatable

- Results (variables)
  - List results: ls()
  - Save results: save.image("c:/path/name")
  - Load results: load("file")
- Commands (things typed)
  - Save commands: savehistory("file")
  - Edit commands:edit(file="c:/temp/class.r")
  - Run commands from file: source("/851/birds.r", echo=TRUE,print.eval=TRUE)

## R intro II - dataframes

- A dataframe is a "fancy" array for holding stats data
  - Has column (=variable) & possible row (=case/data point) labels
- Loading data – have a comma separated text file (can be on URL)

```
birds<-read.csv("c:/851/birds.csv",header=TRUE) #load data into
   datatframe; also sep=';
```

- Looking at a data frame

```
summary(birds)  #summary statistics on data
names(birds) #quick way to see variables
```

- Getting one variable/column

```
birds$SpeciesName #one way to access a variable
birds$Mass

attach(birds) # a quick way to make all variables accessible
Mass
detach(birds)
```

## More data frames

- General syntax
  - dataframename$varname[row subscripting]
  - d[1:10] # all variables, 1st 10 rows
  - d$Mass[3] #3rd row
  - d$Mass[3:9] #7 rows starting at 3
  - d$Mass[-1] #last row
  - d$Mass[d$RangeSize>1000]
  - d$LnMass=log(d$Mass) #add new column
- Group calculations
  - table(d$var1,d$var2) #crosstab
  - tapply(d$numericvar,list(d$factor1,d$factor2),mean/std) #gives stats by group

## R intro III – plotting/summarizing data

```
summary(birds)
pairs(birds)
hist(birds$Mass)
hist(log(birds$mass))
plot(density(log(bird$Mass),na.rm=TRUE))
#
#plotting relationships
#
pairs(birds)
```

## Formulas in R

- Key concept in R is the formula
- Written as "dependent~independents"
- "+" includes additional terms
- Constant term automatically included
  - -1 omits constant (e.g. y~x-1) (or +0)
- Interactions via : & *
  - ":" interaction only Y~x:z
  - "*" includes subterms y~x*z = y~x+z+x:z
- Polynomial:
  - (var1+var2)^2 or even (a+b)*(c+d)
  - poly(var1,var2,2)
- Arithmetic
  - log(var) etc. allowed
  - I(x+y) does actual arithmetic
- Usually can specify dataframe after formula

## Plotting with formulas

plot(Mass~Passerine,data=birds)

boxplot(log(Mass)~Passerine,data=birds)

plot(TotalAbund~Mass,data=birds)

interaction.plot(birds$Invasive,birds$Aquatic,log(birds$Mass))

## Solving OLS with formulas

- Use the "lm" function
- m<-lm(y~mass)
- Usually work with a dataframe, so
  - m<-lm(y~mass,data=mydata)
- What can you do with m
  - print(m)
  - plot(m)
  - summary(m)
  - predict(m,newdata)
  - resid(m) or m$resid
  - coef(m)
  - fitted(m) # predicted
  - model.matrix(m)
  - abline(m) #after plot of formula)

## R examples

```
#
#fitting models
#
ma<-lm(Mass~Aquatic,data=birds) #run a model
summary(ma) #summarize it
plot(ma) #plot it
ma<-lm(log(Mass)~Aquatic,data=birds) #oops - need log transform
summary(ma) #despite enormous variance, is a significant difference
plot(ma)
ma<-lm(log(Mass)~Invasive,data=birds) #no difference
summary(ma)
ra<-lm(Rangesize~TotalAbund,data=birds)
summary(ra)
plot(ra)
fitted(ra)
```