MSc Business Analytics

Financial Modelling and Analysis

Chapter 4: Regressions and herding.

Instructor: Roman Matkovskyy

Twitter: @matkovskyy

Outline

- **Regression**:
  - Which frequency is better?
  - Multiple regression
  - Standard Errors, t-Values, and p-Values
  - Analysis of Variance, Sums of Squares, and R^2
  - Model selection (AIC/BIC)
  - Collinearity and Variance Inflation
- Troubleshooting: Regression Diagnostics (leverages, internally/externally studentized residuals, Cook's Distance)
- Non-linear regression
- Transformations (Transform-Both-Sides Regression, only the Response, Variance Stabilizing Transformations, Box-Cox Transformation)
- Binary regressions.
- **Irrationality and modelling of herding**
  - Linear regression
  - TV-regression
  - Bayesian regression
  - Markov regime switching regression
  - Quantile regression

Basics

- Regression is one of the most widely used of all statistical methods.
- For univariate regression, the available data are **one response variable and *p* predictor variables**, **all measured on each of *n* observations**.
- We let ***Y* denote the response variable and $X_1, \ldots, X_p$ be the predictor or explanatory variables**.
- Also, Y_i **and** $X_{i,1}, \ldots, X_{i,p}$ **are the values of these variables for the *i*th observation.**
- The goals of regression modeling include the investigation of **how *Y* is related to *X*1, . . . , *Xp*, estimation of the conditional expectation of *Y* given *X*1, . . . , *Xp*, and prediction of future *Y* values when the corresponding values of *X*1, . . . , *Xp* are already available**. These goals are closely connected.

Basic, cont.

- A regression model is stated in terms of a **connection between the predictors X and the response Y** .
- Let C(Y|X) denote a property of the distribution of Y given X (as a function of X).
- For example, **C(Y|X) could be E(Y|X), the expected value or average of Y given X, or C(Y|X) could be the probability that Y = 1 given X (where Y = 0 or 1).**
- We define a regression function as **a function that describes interesting properties of Y that may vary across individuals in the population**.
- X describes the list of factors determining these properties. Stated mathematically, a general regression model is given by **C(Y|X) = g(X).**
- We start with the models that, after a certain transformation, are linear in the unknown parameters.
- The general linear regression model is given by C(Y |X) = g(Xβ).
- For example, the ordinary linear regression model is C(Y|X) = E(Y|X) = Xβ

Basic, cont.

- The ***multiple linear regression*** model relating *Y* to the predictor or regressor variables is

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p} + \epsilon_i,$$

- where ϵ_i is called **the noise, disturbances, or errors**. The ϵ_i are often called "errors" because they are the prediction errors when Y_i is predicted by $\beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}$.
- The parameter **β_0 is the intercept**. The regression coefficients $\beta 1, \ldots, \beta p$ are the slopes. **More precisely, βj is the partial derivative of the expected response with respect to the *j*th predictor**:

$$\beta_j = \frac{\partial E\left(Y_i \middle| X_{i,1}, \ldots, X_{i,p}\right)}{\partial X_{i,j}}$$

- Therefore, **βj is the change in the expected value of Y_i when $X_{i,j}$ changes one unit**.
- It is assumed that the noise,**$\epsilon_1, \ldots, \epsilon_n$, is i.i.d. (independent and identically distributed random)** with mean 0 and constant variance σ^2.
- Often the **$\epsilon_i s$ are assumed to be normally distributed**, **which with implies Gaussian white noise**.

The assumptions of the linear regression

- **linearity of the conditional expectation**:

$$E(Yi \mid X_{i,1}, \ldots, X_{i,p}) = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}.$$

- **independent noise:** $\epsilon_1, \ldots, \epsilon_n$ are independent;
- **constant variance:** Var(i) = σ^2 for all i;
- **Gaussian noise:** ϵ_i is normally distributed for all i.
- We will discuss methods for checking these assumptions, the consequences of their violations, and possible remedies when they do not hold.

Straight-Line Regression

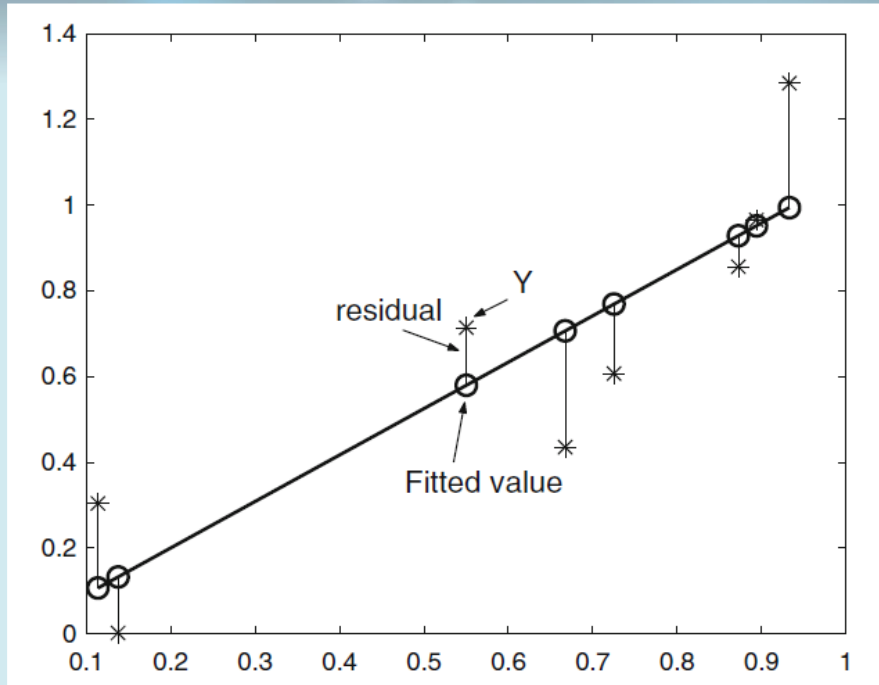
- *Straight-line regression* is linear regression with only one predictor variable.
- The model is

$$Yi = \beta 0 + \beta 1 Xi + \epsilon_i ,$$

- where β0 and β1 are the unknown intercept and slope of the line and ϵ_i is called the noise or error.
- The regression coefficients can be estimated by the *method of least squares*.
- The least-squares estimates are the values of β0 and β1 that minimize

$$\sum_{i=1}^{n} \{Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_i)\}^2$$

- Geometrically, **we are minimizing the sum of the squared lengths of the vertical lines i**n Fig. The data points are shown as asterisks.
- The vertical lines connect the data points and the predictions using the linear equation.
- The predictions themselves are called the *fitted values* or "*y*-hats" and shown as open circles.
- The differences between the *Y* -values and the fitted values are called the *residuals*.



Least-squares estimation. The vertical lines connect the data (*) and the fitted values (o) represent the residuals. The least-squares line is defined as the line making the sum of the squared residuals as small as possible.

Application –Interest rate

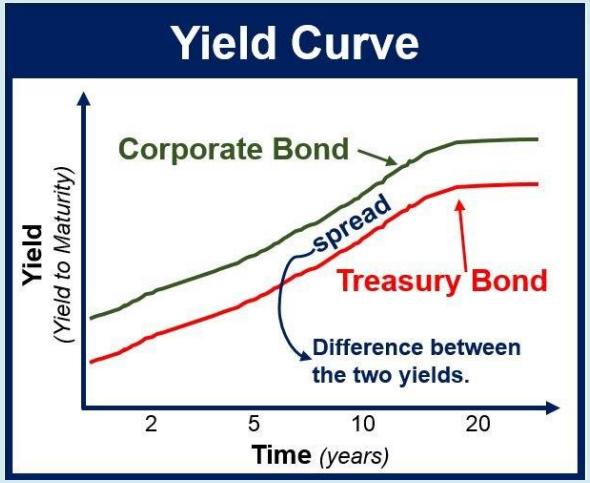
- It is nearly impossible to get through the day without seeing some reference to interest rates on saving or borrowing money.
- You may see interest rates being offered on savings accounts by depository institutions, interest rates on new and used automobiles, and even the rate at which you could borrow for a loan to pay your tuition or to purchase a home.
- Your cost of borrowing will generally be higher when you are just starting your working career and your credit quality has not yet been established.

Application: *Weekly interest rates*

- The interest rate is the amount (in %) a lender charges for the use of assets expressed as a percentage of the principal.
- We use the following interest rates
- 10-year Treasury constant maturity rate:
  - **An index published by the Federal Reserve Board based on the average yield of a range of Treasury securities, all adjusted to the equivalent of a 10-year maturity**.
  - Yields on Treasury securities at constant maturity are determined by the U.S. Treasury from the daily yield curve.
  - The Treasury yield curve is estimated daily using a cubic spline model.
  - That is based on the closing market-bid yields on actively traded Treasury securities in the over-the-counter market.
  - **A bid refers to the price at which a market maker** (an investor, trader, or dealer**) is willing to buy**. [***The ask is the price a seller is willing to accept for a security, which is often referred to as the offer price***. The spread between the bid and the ask is a reliable indicator of supply and demand]
- **10-year Treasury constant maturity rate is used as a reference point to establish the price of other securities such as corporate bonds.**
- Treasury securities are considered **risk-free** since they are backed by the U.S. government. This rate, and an added margin based upon the risk involved, is used in pricing various debt securities.

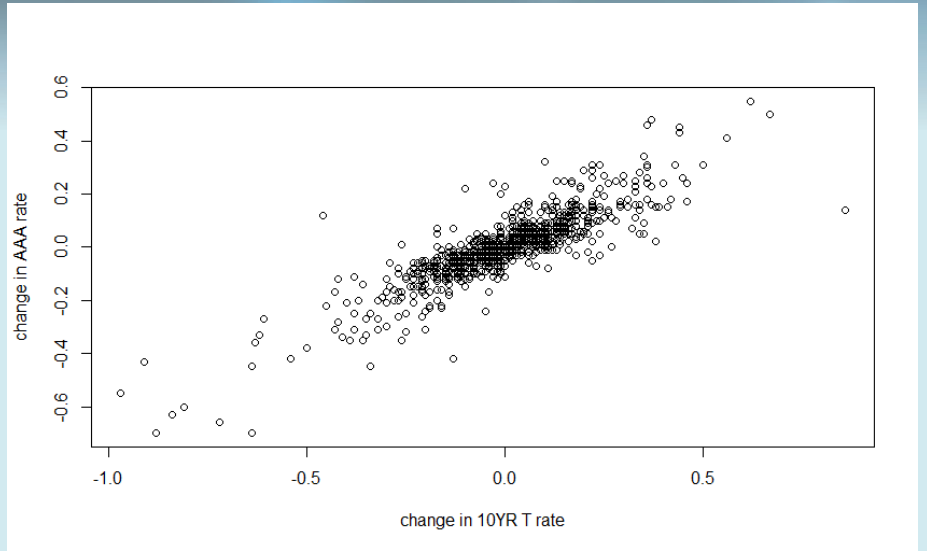
Application: *Weekly interest rates*

- the Moody’s corporate AAA bond yield, also known as "Moody's Aaa" is an investment bond that acts as an index of the performance of all bonds given an Aaa rating by Moody's Investors Service.
- This corporate bond is often used in macroeconomics as an alternative to the federal ten-year Treasury Bill as an indicator of the interest rate.
  - Bonds are units of corporate debt issued by companies and securitized as tradeable assets.
  - A bond is referred to as a fixed-income instrument since bonds traditionally paid a fixed interest rate (coupon) to debtholders. Variable or floating interest rates are also now quite common.
  - Bond prices are inversely correlated with interest rates: when rates go up, bond prices fall and vice-versa.
  - Bonds have maturity dates at which point the principal amount must be paid back in full or risk default.



Application: *Weekly interest rates — least-squares estimates*

- Weekly interest rates from February 16, 1977, to December 31, 1993, were obtained from the Federal Reserve Bank of Chicago. Figure is a plot of changes in the 10-year Treasury constant maturity rate (X) and changes in the Moody's seasoned corporate AAA bond yield (Y). The plot looks linear, so we try linear regression using R's `lm()` function.
- The code **`aaa_dif ~ cm10_dif`** is an example of a formula in R with the outcome variable to the left of "~" and the explanatory variables to the right of "~."
- In this example, there is only one explanatory variable. In cases where there are multiple explanatory variables, they are separated by "+". Here is the output.




```
Call:
lm(formula = aaa_dif ~ cm10_dif)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3894 -0.0330  0.0001  0.0293  0.4034

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.000109   0.002221   -0.05     0.96
cm10_dif     0.615762   0.012117   50.82   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0659 (ϵ_i) on 878 degrees of freedom
Multiple R-squared:  0.746,      Adjusted R-squared:  0.746
F-statistic: 2.58e+03 on 1 and 878 DF,  p-value: <2e-16
```

```
dat = read.table(file="WeekInt.txt",header=T)
attach(dat)
cm10_dif = diff( cm10 ) # 10-year Treasury constant maturity rate
aaa_dif = diff( aaa ) # the Moody's seasoned corporate AAA bond yield

par(mfrow=c(1,1)) # one graph in the window
plot(cm10_dif,aaa_dif,xlab="change in 10YR T rate", ylab="change in AAA
rate") # plot
summary(lm(aaa_dif ~ cm10_dif)) # show the summary of the regression
estimate

#another way
results = lm(aaa_dif ~ cm10_dif)
summary (results)
```

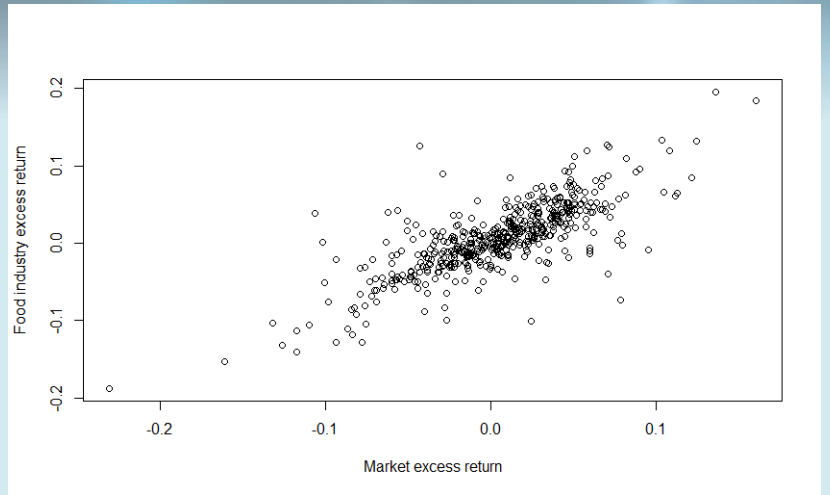
Modelling of Excess return

- **The excess return on a security or market index is the return minus the risk-free interest rate.**
- investors hope for positive excess return because it provides an investor with more money than they could have achieved by investing elsewhere.
- An important application of linear regression in finance is the regression of the excess return of an asset or market sector on the excess return of the entire market.

Application: *Excess returns on the food sector and the market portfolio*

- In this example, we will regress the excess monthly return of the food sector (rfood) on the excess monthly return of the market portfolio (rmrf).
- The data are in R's Capm data set in the Ecdat package and are plotted in Fig. The returns are expressed as percentages in the data set but have been converted to fractions in this example.

```
data(Capm,package="Ecdat")
attach(Capm)
rfood2 = rfood/100
rmrf2 = rmrf/100
plot(rmrf2,rfood2,ylab="Food industry excess return",
     xlab="Market excess return")
results = lm(rfood2~rmrf2)
summary(results)
```



- The output from lm is below the figure.
- Thus, the fitted regression equation is

rfood = 0.00339 + 0.78342 rmrf +ϵ, and σ = **0.0289**.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.00339    0.00128    2.66   0.0081 **
rmrf2        0.78342    0.02835   27.63   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1

Residual standard error: 0.0289 on 514 degrees of
freedom
Multiple R-squared:  0.598,   Adjusted R-squared:
0.597
F-statistic:  763 on 1 and 514 DF,  p-value: <2e-16
```

Variance of $\hat{\beta}_1$

- It is useful to have a formula for **the variance of an estimator to show how the estimator's precision depends on various aspects of the data such as the sample size and the values of the predictor variables**.
- Fortunately, it is easy to derive a formula for the variance of $\hat{\beta}_1$.
- We can write $\hat{\beta}_1$ as a weighted average of the responses

$$\hat{\beta}_1 = \sum_{i=1}^{n} w_i Y_i$$

- where w_i is the weight given by

$$w_i = \frac{X_i - \bar{X}}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

- We consider $X_1, \ldots, X_n$ as fixed, so if they are random we are conditioning upon their values. From the assumptions of the regression model, it follows that $\text{Var}(Yi|X1, \ldots, Xn) = \hat{\sigma}_\in$ and $Y_1, \ldots, Y_n$ are conditionally uncorrelated.
- Therefore,

$$Var(\hat{\beta}_1 | X_1, \ldots, X_n) = \sigma_\epsilon^2 \sum_{i=1}^{n} w_i^2 = \frac{\sigma_\epsilon^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2} = \frac{\boldsymbol{\sigma_\epsilon^2}}{(\boldsymbol{n-1})\boldsymbol{s_X^2}}$$

- the numerator σ_ϵ^2 is simply the variance of the ϵ_i. Reason - **More variability in the noise means more variable estimators**.
- The denominator shows us that the variance of β1 is inversely proportional to (n−1) and to s_X^2 (the sample variance of X)
- **So the precision of β1 increases as $\boldsymbol{\sigma_\epsilon^2}$ is reduced, *n* is increased, or $\boldsymbol{s_X^2}$ is increased**.
- Why does increasing s_X^2 decrease Var(β1/*X*1, . . . , *Xn*)? The reason is that increasing s_X^2 means that the *Xi* are spread farther apart, which makes the slope of the line easier to estimate.

Optimal sampling frequencies

- Suppose that we have, *Xt* and *Yt*, and we wish to regress *Yt* on *Xt*.
- **A significant practical question is whether one should use daily or weekly data, or perhaps even monthly or quarterly data.**
- **Does it matter which sampling frequency we use?**
- The answer is "yes" and **the highest possible sampling frequency gives the most precise estimate of the slope**.
- To understand why this is so, we compare daily and weekly data. Assume that the *Xt* and *Yt* are white noise sequences.
- **Since a weekly log return is simply the sum of the five daily log returns within a week, σ_ϵ^2 and s_X^2 will each increase by a factor of five if we change from daily to weekly log returns, so the ratio σ_ϵ^2/s_X^2 will not change.**
- However, **by changing from daily to weekly log returns, (n − 1) is reduced by approximately a factor of five. The result is that Var(β1/*X*1, . . . , *Xn*) is approximately five times smaller using daily rather than weekly log returns.**
- **Similarly, Var(β1|*X*1, . . . , *Xn*) is about four times larger using monthly rather than weekly returns**.
- The obvious **conclusion is that one should use the highest sampling frequency available**, which is often daily returns.
- We have assumed that the *Xt* and *Yt* are white noise in order to simplify the calculations, but this conclusion still holds if they are stationary but autocorrelated.
- For time series models, quite often, high freqency data are used. The down side of it is higher noise.

Multiple Linear Regression

- The multiple linear regression model is

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p} + \epsilon_i .$$

- The least-squares estimates are the values β0, β1, . . . , βp that minimize

$$\sum_{i=1}^{n} \left\{ Y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 X_{i,1} + \cdots + \hat{\beta}_p X_{i,p} \right) \right\}^2$$

- An unbiased estimate of σ_ϵ^2 is

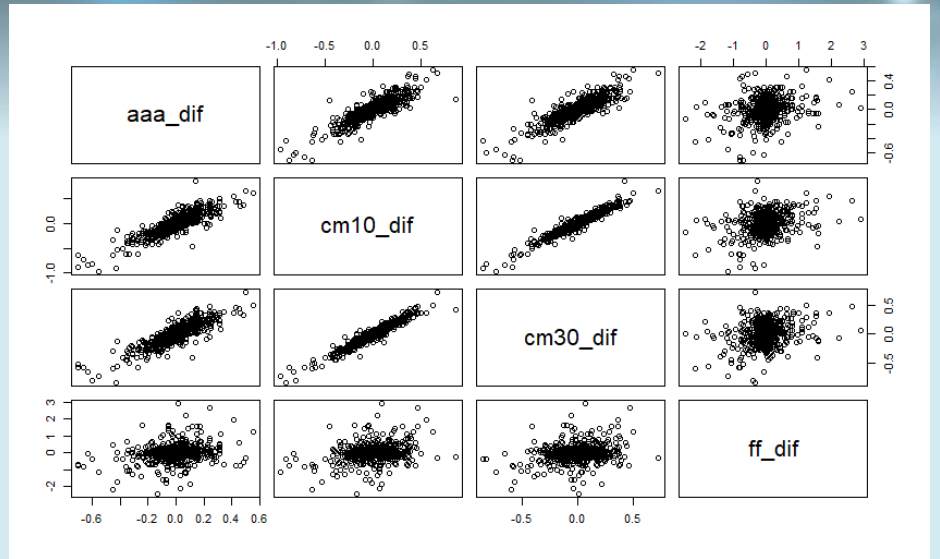
$$\hat{\sigma}_\epsilon^2 = \frac{\sum_{i=1}^{n} \widehat{\epsilon_i^2}}{n - 1 - p}$$

- The denominator in is the sample size minus the number of regression coefficients that are estimated.

Application: Multiple linear regression with interest rates

- As an example, we continue the analysis of the weekly interest-rate data but now with changes in the 30-year Treasury rate (cm30_dif) and changes in the Federal funds rate (ff_dif) as additional predictors. Thus $p = 3$.
- The 30-10 Treasury Yield Spread is the difference between the 30 year treasury rate and the 10 year treasury rate.
  - A 30-10 treasury spread that approaches 0 signifies a **"flattening" yield curve** (A flattening yield curve is when short-term and long-terms see no discernible change in rates. This makes long-term bonds less attractive to investors). Such a curve can be considered **a psychological marker, one that could mean investors are losing faith in a long-term market's growth potential**.
  - if the **spread goes negative**, this indicates a flight to safety that **can signal a lack of confidence in the strength of the economy.**
- The fed funds rate **is the interest rate banks pay for overnight borrowing in the federal funds market** (the rate at which commercial banks borrow and lend their excess reserves to each other overnight).
- It is also used to influence other interest rates, such as credit cards, mortgages, and bank loans.
- It also affects the value of the U.S. dollar and other household and business assets.
- Figure is a scatterplot matrix of the four time series. There is a strong linear relationship between all pairs of aaa_dif, cm10_dif, and cm30_dif, but ff_dif is not strongly related to the other series. The code is

- `summary(lm(aaa_dif ~ cm10_dif + cm30_dif + ff_dif))`



Scatterplot matrix of the changes in four weekly interest rates. The variable aaa dif *is the response*

```
Coefficients:
             Estimate     Std. Error t  value       Pr(>|t|)
(Intercept)  -9.07e-05    2.18e-03     -0.04         0.97
cm10_dif     3.55e-01     4.51e-02     7.86          1.1e-14 ***
cm30_dif     3.00e-01     5.00e-02     6.00          2.9e-09 ***
ff_dif       4.12e-03     5.28e-03     0.78          0.44

Residual standard error: 0.0646 on 876 degrees of freedom
Multiple R-squared:  0.756, Adjusted R-squared:  0.755
F-statistic:  906 on 3 and 876 DF,  p-value: <2e-16
```

Standard Errors, t-Values, and p-Values

```
Coefficients:
                Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)    -9.07e-05   2.18e-03    -0.04     0.97
cm10_dif        3.55e-01   4.51e-02     7.86     1.1e-14 ***
cm30_dif        3.00e-01   5.00e-02     6.00     2.9e-09 ***
ff_dif          4.12e-03   5.28e-03     0.78     0.44

Residual standard error: 0.0646 on 876 degrees of freedom
Multiple R-squared:  0.756, Adjusted R-squared:  0.755
F-statistic:  906 on 3 and 876 DF,  p-value: <2e-16
```

- As noted before, the estimated coefficients are $\hat{\beta}_0 = -9.07 \times 10^{-05}$, $\hat{\boldsymbol{\beta}}_{\mathbf{1}} = \mathbf{0.355}$, $\hat{\beta}_2 = 0.300$, and $\hat{\beta}_3 = 0.00412$. Each of these coefficients has three other statistics associated with it.
- **The standard error (SE),** which is the estimated standard deviation of the least-squares estimator, tells us the precision of the estimator.
- The ***t*-value**, **is the *t*-statistic for testing that the coefficient is 0. The *t*value is the ratio of the estimate to its standard error**. For example, for cm10_dif, the *t*-value is 7.86 = **0.355/0.0451**.
- The ***p*-value** (Pr > |t| in the lm() output), associated with **testing the null hypothesis that the coefficient is 0 versus the alternative that it is not 0**.
  - If a *p*-value for a slope parameter is small, as it is here for $\hat{\beta}_1$, then this is evidence that the corresponding coefficient is *not* 0, which means that the predictor has a *linear* relationship with the response.

Standard Errors, t-Values, and p-Values, Cont.

- It is important to keep in mind that the ***p*-value only tells us if there is a linear relationship.**
- The existence of a linear relationship **between *Yi* and *Xi,j* means only that the linear predictor of *Yi* has a nonzero slope on *Xi,j* , or, equivalently, that partial correlation between *Xi,j* and *Yi* is not zero**. (*The partial correlation between two variables is their correlation when all other variables are held fixed*.)
- **When the *p*-value is small (so a linear relationship exists), there could also be a strong nonlinear deviation from the linear relationship.** Moreover, **when the *p*-value is large (so no linear relationship exists), there could still be a strong nonlinear relationship.**
- Because of the potential for nonlinear relationships to go undetected in a linear regression analysis, graphical analysis of the data and residual analysis are essential.
- The *p*-values for β1 and β2 are *very* small, so we can conclude that these slopes are *not* 0. The *p*-value is large (0.97) for β0, so we would not reject the hypothesis that the intercept is 0.
- Similarly, we would not reject the null hypothesis that β3 is zero. Stated differently, we can accept the null hypothesis that, conditional on cm10_dif and cm30_dif, aaa_dif and ff_dif are not linearly related. This result should *not* be interpreted as stating that aaa_dif and ff_dif are unrelated, but only that ff_dif is not useful for predicting aaa_dif when cm10_dif and cm30_dif are included in the regression model. (In fact, aaa_dif and ff_dif have a correlation of 0.25 (this is the full, not partial, correlation) and the linear regression of aaa_dif on ff_dif alone is highly significant; the *p*-value for testing that the slope is zero is $5.158 \times 10{-}14$.)
- **Since the Federal Funds rate is a short-term (overnight) rate, it is not surprising that ff_dif is less useful than changes in the 10- and 30-year Treasury rates for predicting aaa_dif.**

Standard Errors, t-Values, and p-Values, Cont.

- For regression with one predictor variable, the standard error of $\hat{\beta}_1$ is $\mathrm{SE} = \dfrac{\hat{\sigma}_\epsilon}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}}$.
- When there are more than two predictor variables, formulas of standard errors are more complex and are facilitated by the use of matrix notation.
- Because standard errors can be computed with standard software, the formulas are not needed for applications and so are postponed.

Analysis of Variance, Sums of Squares, and R^2

- Certain results of a regression fit are often displayed in an *analysis of variance table*, also called the ANOVA or AOV table.
- **The idea behind the ANOVA table is to describe how much of the variation in *Y* is predictable if one knows *X*1, . . . , *Xp*.** Here is the ANOVA table for the model from the previous Example:

```
> anova(lm(aaa_dif ~ cm10_dif + cm30_dif + ff_dif))
Analysis of Variance Table
Response: aaa_dif
           Df    Sum Sq     Mean Sq    F value    Pr(>F)
cm10_dif   1     11.21      11.21      2682.61    < 2e-16 ***
cm30_dif   1     0.15       0.15       35.46      3.8e-09 ***
ff_dif     1     0.0025     .0025      0.61       0.44
Residuals 876    3.66  0.0042
```

- The total variation in *Y* can be partitioned into two parts: **the variation that can be predicted by *X*1, . . . , *Xp* and the variation that cannot be predicted**.

Analysis of Variance, Sums of Squares, and R^2, cont.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
cm10_dif	1	11.21	11.21	2682.61	< 2e-16 ***
cm30_dif	1	0.15	0.15	35.46	3.8e-09 ***
ff_dif	1	0.0025	.0025	0.61	0.44
Residuals	876	3.66	0.0042		

- **The variation that can be predicted is measured by the regression sum of squares**, which is **Regression** $SS = \sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2$
- **The regression sum of squares** for the model that uses only cm10_dif is in the first row of the ANOVA table and is **11.21**. **The entry, 0.15, in the second row is the increase in the regression sum of squares when cm30_dif is added to the model.**
- Similarly, **0.0025** is the increase in the regression sum of squares when ff_dif is added. Thus, rounding to two decimal places, **11.36** = **11.21 + 0.15 + 0.00** **is the regression sum of squares with all three predictors in the model**.
- **The amount of variation in *Y* that cannot be predicted by a linear function of *X1, . . . , Xp* is measured by the residual error sum of squares**, which is the sum of the squared residuals; i.e.,

$$\text{residual error } SS = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

- In the ANOVA table, **the residual error sum of squares** is in the last row and is **3.66**.
- The **total variation is measured by the total sum of squares (total SS),** which is the sum of the squared deviations of *Y* from its mean; that is,

$$\text{Total } SS = \sum_{i=1}^{n}(Y_i - \overline{Y}_i)^2$$

- It can be shown algebraically that **total SS = regression SS + residual error SS**. Therefore, the total SS is **11.36** + **3.66** = **15.02**

Analysis of Variance, Sums of Squares, and R^2, cont.

- R-squared, denoted by R^2, is

$$R^2 = \frac{\boldsymbol{regression\ SS}}{\boldsymbol{total\ SS}} = 1 - \frac{\boldsymbol{residual\ error\ SS}}{\boldsymbol{total\ SS}}$$

- and measures the proportion of the total variation in *Y* that can be linearly predicted by *X*.
- In the example, R^2 is 0.746 = 11.21/15.02 if only cm10_dif is the model and is 11.36/15.02 = 0.756 if all three predictors are in the model.
- *R* can be viewed as the "multiple" correlation between *Y* and many *X*s.
- **The residual error sum of squares is also called the error sum of squares or sum of squared errors and is denoted by SSE**

Analysis of Variance, Sums of Squares, and R^2, cont.

- It is important to understand that **sums of squares** in an ANOVA table **depend upon the order of the predictor variables in the regression, because the sum of squares for any variable is the increase in the regression sum of squares when that variable is added to the predictors already in the model.**
- The table below has the same variables as before, but the order of the predictor variables is reversed.
- Now that ff_dif is the first predictor, its sum of squares is much larger than before and its *p*-value is highly significant; before it was nonsignificant, only 0.44. The sum of squares for cm30_dif is now much larger than that of cm10_dif, the reverse of what we saw earlier, since cm10_dif and cm30_dif are highly correlated and the first of them in the list of predictors will have the larger sum of squares.

```
> anova(lm(aaa_dif~ff_dif+cm30_dif+cm10_dif))
Analysis of Variance Table
Response: aaa_dif
               Df      Sum Sq  Mean Sq F value  Pr(>F)
ff_dif         1       0.94    0.94    224.8    < 2e-16 ***
cm30_dif       1       10.16   10.16   2432.1   < 2e-16 ***
cm10_dif       1       0.26    0.26    61.8      1.1e-14 ***
Residuals      876     3.66    0.00
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The lesson here is that **an ANOVA table is most useful for assessing the effects of adding predictors in some natural order**. Since AAA bonds have maturities closer to 10 than to 30 years, and since the Federal Funds rate is an overnight rate, it made sense to order the predictors as cm10_dif, cm30_dif, and ff_dif as done initially.

Degrees of Freedom (DF)

- **There are degrees of freedom (DF) associated with each of these sources of variation (observations).**
- **Degrees of Freedom refers to the maximum number of logically independent values, which are values that have the freedom to vary, in the data sample.**
- **The total degrees of freedom is *n* − 1. The residual error degrees of freedom is *n* − *p* − 1.**

- Here is a way to think of degrees of freedom.
- **Initially, there are *n* degrees of freedom, one for each observation**. Then **one degree of freedom** is allocated to estimation of the intercept. This leaves a total of $n - 1$ degrees of freedom for estimating the effects of the *X* variables and σ_ϵ^2
- **Each regression parameter uses one degree of freedom for estimation**. Thus, there are $(n - 1) - p$ degrees of freedom remaining for estimation of σ_ϵ^2 using the residuals. There is an elegant geometrical theory of regression where the responses are viewed as lying in an *n*-dimensional vector space and degrees of freedom are the dimensions of various subspaces.

Mean Sums of Squares (MS) and F-Tests

- As just discussed, every sum of squares in an ANOVA table has an associated degrees of freedom.
- The ratio of the sum of squares to the degrees of freedom is the mean sum of squares:

mean sum of squares = sum of squares/degrees of freedom

- The residual mean sum of squares is the unbiased estimate σ_ϵ^2

$$\hat{\sigma}_\epsilon^2 = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n-1-p} = \textbf{\textit{residual mean sum of squares}} =$$

= **residual error SS/residual degrees of freedom**

Testing two models

- Other mean sums of squares are used in testing.
- **Suppose we have two models, I and II,** and the predictor variables(Xs) in model I are a subset of those in model II, so that model I is a submodel of II.
- A common null hypothesis is that the data are generated by model I.
- Equivalently, in model II the slopes are zero for variables not also in model I.
- To test this hypothesis, we use the **excess regression sum of squares of model II relative to model I (SS- sum of squares):**

SS(II/I) = regression SS for model II – regression SS for model I = residual SS for model I – residual SS for model II.

Testing two models, cont.

- The degrees of freedom for SS(**II** | **I**) is the number of extra predictor variables (Xs) in model II compared to model I.
- The mean square is denoted as MS(**II** | **I**).
- Stated differently, if *p* **I** and *p* **II** are the number of parameters in models **I** and **II**, respectively, then df (**II**| **I**)= *p* **II** − *p* **I** and MS(**II** | **I**) = SS(**II** | **I**)/df(**II** | **I**). The *F*-statistic for testing the null hypothesis is

$$F = \frac{MS(II|I)}{\hat{\sigma}_{\epsilon}^{2}}$$

- **where $\hat{\sigma}_{\epsilon}^{2}$ is the mean residual sum of squares for model II**.
- Under the null hypothesis, the *F*-statistic has an *F*-distribution with df II/ I and *n* − *p* II − 1 degrees of freedom and the **null hypothesis is rejected if the *F*-statistic exceeds the *α*-upper quantile of this *F*-distribution**. (H0 is that the data are generated by model I)

Exercise: Weekly interest rates—Testing the one-predictor versus three predictor model

- **In this example, the null hypothesis is that, in the three-predictor model, the slopes for cm30_dif and ff_dif are zero**.
- The *F*-test can be computed using R's `anova` function. The output

```
## Testing models
fit1 = lm(aaa_dif ~ cm10_dif)
fit2 = lm(aaa_dif~cm10_dif+cm30_dif)
fit3 = lm(aaa_dif~cm10_dif+cm30_dif+ff_dif)
anova(fit1, fit3)
Analysis of Variance Table

Model 1: aaa_dif ~ cm10_dif
Model 2: aaa_dif ~ cm10_dif + cm30_dif + ff_dif
  Res.Df  RSS Df Sum of Sq  F  Pr(>F)
1    878 3.81
2    876 3.66  2     0.151 18 2.1e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> anova(fit2, fit3)
Analysis of Variance Table

Model 1: aaa_dif ~ cm10_dif + cm30_dif
Model 2: aaa_dif ~ cm10_dif + cm30_dif + ff_dif
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1    877 3.66
2    876 3.66  1   0.00254 0.61   0.44
```

- In the last row, the entry 2 in the "Df" column is the difference between the two models in the number of parameters and 0.15 in the "Sum of Sq" column **is the difference between the residual sum of squares (RSS) for the two models.**
- The very small *p*-value (2.1×10^{-8}) leads us to reject the null hypothesis and says that the result is "**highly significant**." (statistical significance), i.e. Model 2 is better
- When the sample size is as large as it is here, it is common to reject the null hypothesis.
  - The reason for this is that the null hypothesis is rarely true exactly, and with a large sample size it is highly likely that even a small deviation from the null hypothesis will be detected. *Statistically significance must be distinguished from practical significance.*
- The adjusted R_2 values for the two- and three-variable models are very similar, 0.756 and 0.755, respectively. Therefore, the rejection of the two-variable model may not be of practical importance.
- The large p-value (**0.44**) leads us to accept the null hypothesis. Model 1 is better.

Model Selection

- When there are many potential predictor variables, often we wish to find a subset of them that provide a parsimonious regression model.
- ***F*-tests are not very suitable for model selection**. One problem is that there are many possible *F*-tests and the joint statistical behavior of all of them is not known.
- For model selection, it is more appropriate to use a model selection criterion such as **AIC (Akaike information criterion) or BIC (Bayesian information criterion)**.
- For linear regression models, AIC is

$$AIC = n\log(\hat{\sigma}^2) + 2(1+p)$$

- where 1+*p* is the number of parameters in a model with *p* predictor variables; the intercept gives us the final parameter.
- BIC replaces **2(1 + *p*)** in AIC by **log(*n*)(1+*p*).**
- **With AIC, and BIC, smaller values are better, but for adjusted R^2, larger values are better.**

Model Selection, cont.

- In addition to AIC and BIC, there are two model selection criteria specialized for regression.
- One is adjusted R^2.
- Another is *Cp (Mallows Cp: A variant of AIC developed by Colin Mallows)*.
  - *Cp* is related to AIC and usually *Cp* and AIC are minimized by the same model. The primary reason for using *Cp* instead of AIC is that some regression software computes only *Cp*, not AIC—this is true of the `regsubsets()` function in R's *leaps* package which will be used in the following example.
  - To define *Cp*, suppose there are *M* predictor variables. Let $\hat{\sigma}^2_{\epsilon,M}$ be the estimate of *variance* using all of them, and let squared estimate of errors, SSE(*p*), be the sum of squares for residual error for a model with some subset of only $p \leq M$ of the predictors (as usual, *n* is the sample size):

$$C_p = \frac{SSE(p)}{\hat{\sigma}^2_{\epsilon,M}} - n + 2(p+1)$$

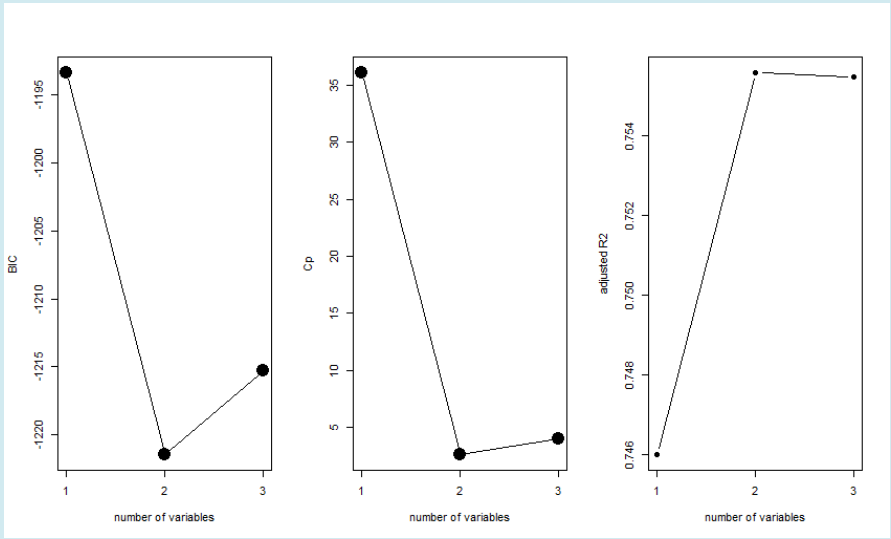
  - Of course, *Cp* will depend on which particular model is used among all of those with *p* predictors, so the notation "*Cp*" may not be ideal.

Model Selection, cont.

- **Model choice should be guided by economic theory and practical considerations, as well as by model selection criteria**.
- It is important that the **final model makes sense** to the user.
- Subject-matter expertise might lead to adoption of a model not optimal according to the criterion being used but, instead, to a model slightly below optimal but more parsimonious or with a better economic rationale.

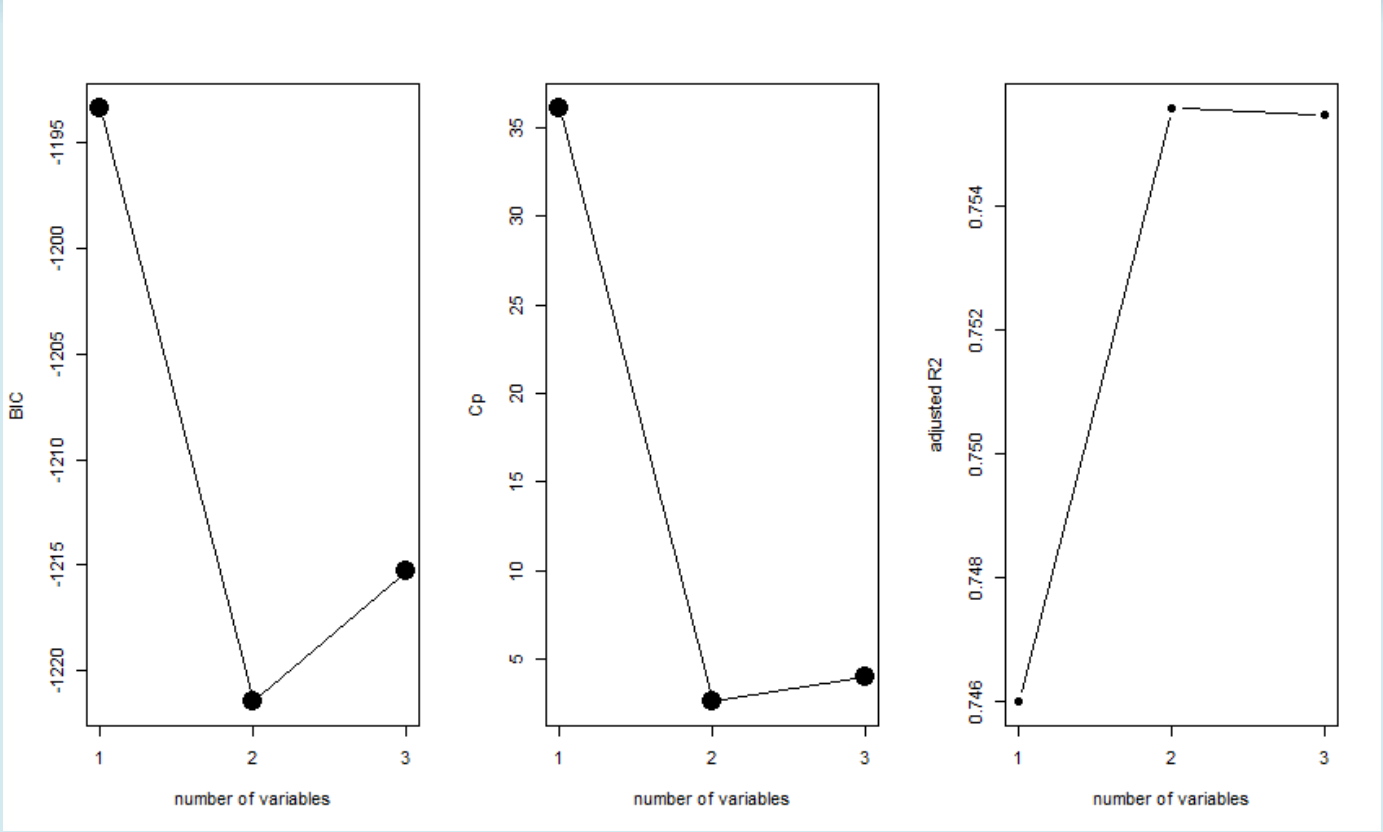
Exercise: *Weekly interest rates—Model selection by AIC and BIC*

- Figure contains plots of **the number of predictors in the model versus the optimized value of a selection criterion**. By "optimized value," we mean the best value among all models with the given number of predictor variables.
- **"Best" means smallest for BIC and *Cp* and largest for adjusted *R*2.**
- There are three plots, one for each of BIC, *Cp*, and adjusted *R*2. All three criteria are optimized by two predictor variables.
- There are three models with two of the three predictors. The one that optimized the criteria1 is the model with cm10_dif and cm30_dif, as can be seen in the following output from `regsubsets`.
- Here "*" indicates a variable in the model and " " indicates a variable not in the model, so the three rows of the table indicate that the best one-variable model is cm10_dif and the best two-variable model is cm10_dif and cm30_dif—the third row does not contain any real information since, with only three variables, there is only one possible three-variable model.




```
library(leaps)
subsets = regsubsets(aaa_dif~.,
data=as.data.frame(cbind(cm10_dif,cm30_dif,
ff_dif)),nbest=1)
b = summary(subsets)
b

Subset selection object
Call: regsubsets.formula(aaa_dif ~ ., data
= as.data.frame(cbind(cm10_dif,
    cm30_dif, ff_dif)), nbest = 1)
3 Variables  (and intercept)
          Forced in Forced out
cm10_dif      FALSE      FALSE
cm30_dif      FALSE      FALSE
ff_dif        FALSE      FALSE
1 subsets of each size up to 3
Selection Algorithm: exhaustive
         cm10_dif cm30_dif ff_dif
1  ( 1 ) "*"      " "      " "
2  ( 1 ) "*"      "*"      " "
3  ( 1 ) "*"      "*"      "*"
```

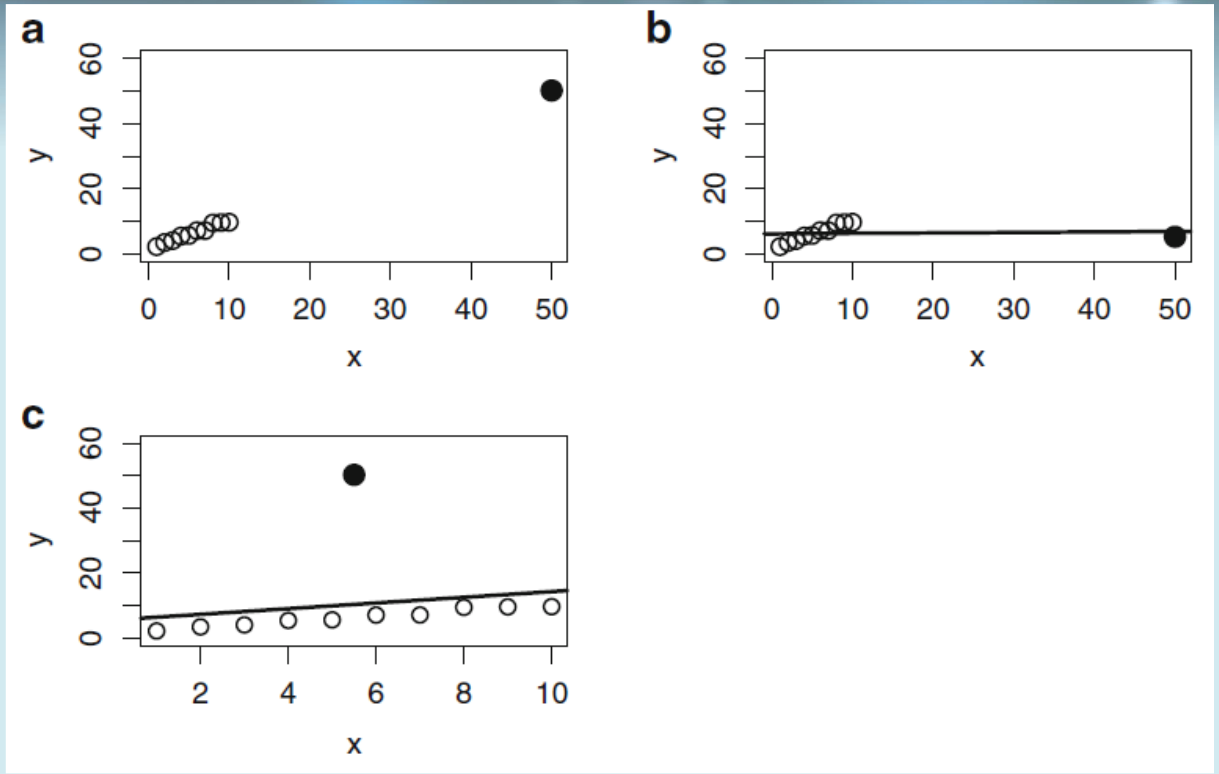
Exercise: *Weekly interest rates—Model selection by AIC and BIC, cont.*



Regression Diagnostics

Three important tools will be discussed for diagnosing problems with the model or the data:

- leverages (a);
- externally studentized residuals (b); and
- Cook's Distance (Cook's D), which quantifies the overall influence of each observation on the fitted values.



a) Linear regression with a high-leverage point that is not a residual outlier (solid circle).
b) Linear regression with a high-leverage point that is a residual outlier (solid circle).
c) Linear regression with a low-leverage point that is a residual outlier (solid circle). Least-squares fits are shown as solid lines.

Leverages

- **The *leverage* of the *i*th observation, denoted by H_{ij}, measures how much influence *Yi* has on its own fitted value $\hat{Y}_i$. We will not go into the algebraic details.**
- An important result is that there are weights H_{ij} depending on the values of the predictor variables but *not* on *Y*1, . . . , *Yn* such that

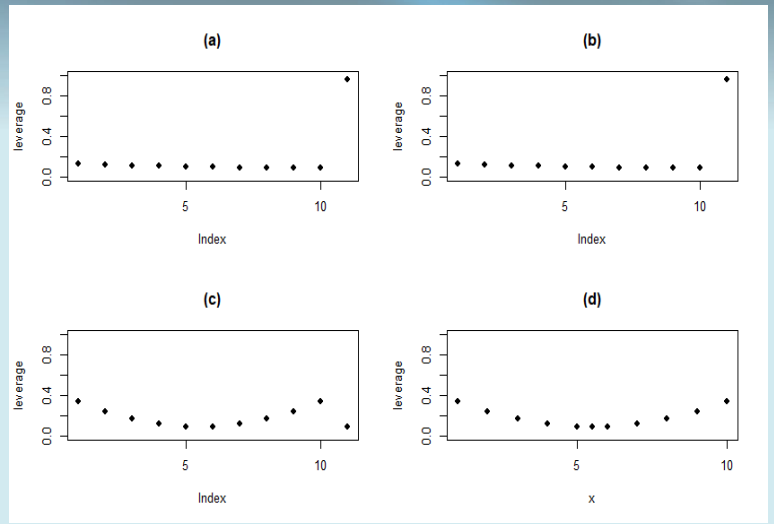
$$\hat{Y}_i = \sum_{j=1}^{n} H_{ij} Y_j$$

- In particular, H_{ij} is the weight of Y_i in the determination of $\hat{Y}_i$.
- **It is a potential problem if H_{ij} is large since then $\hat{Y}_i$ is determined too much by Y_i itself and not enough by the other data**.
- **A high value of H_{ij} means a fitted value with low accuracy**.
- **The leverage value H_{ii} is large when the predictor variables for the *i*th case are atypical of those values in the data, for example, because one of the predictor variables for that case is extremely outlying**.
- the average of $H_{11}, \ldots, H_{nn}$ is (p + 1)/n, where p + 1 is the number of parameters (one intercept and p slopes) and that therefore 0 < H_{ii} < 1. A value of H_{ii} exceeding **2(*p*+1)/*n***, that is, over twice the average value, is generally considered to be too large and therefore a cause for concern (Belsley et al. 1980).
- **The square matrix with *i, j*th element equal to H_{ij} is called the hat matrix since it converts Y_j, *j* = 1, . . . , *n*, to $\hat{Y}_i$. The H_{ij} are sometimes called the *hat diagonals*.**

Leverage: example

Figure plots the leverages for the three cases.

```
# Leverages example
set.seed(99) # to insure replicability
x = 1:11 # create a variable x with values from 1 to 11
x[11] = 50 # replace the 11th element with 50 (anomaly)
y=1+x+rnorm(11) # generate y that depends on x
y2 = y
y2[11] = y[11]-45
x2 = x
x2[11] = 5.5

# create cexx with 10 values of 21, and the last element is 19
cexx = c(rep(21,10),19)

# hatvalues () calculates hatvalues

# the plots from teh slides
par(mfrow=c(2,2),lwd=1,pch=19)
plot(hatvalues(lm(y~x)),ylab="leverage",main="(a)",ylim=c(0,1))
plot(hatvalues(lm(y2~x)),ylab="leverage",main="(b)",ylim=c(0,1))
plot(hatvalues(lm(y~x2)),ylab="leverage",main="(c)",ylim=c(0,1))
plot(x2,hatvalues(lm(y~x2)),xlab="x",ylab="leverage",
main="(d)",ylim=c(0,1))
```



***a*)–(*c*)** *Leverages plotted again case number (index).*
*Panels (**a**) and (**b**) are identical because leverages do not depend on the response values.*
*Panel (**d**) plots the leverages in (**c**) against X_i.*

- Because the leverages depend only on the *X*-values, the leverages are the same in panels (a) and (b).
- In both panels, the high-leverage point has a leverage equal to 0.960.
- In these examples, the rule-of-thumb cutoff point for high leverage is only $2(p + 1)/n = 2 * 2/11 = 0.364$, so 0.960 is a huge leverage and close to the maximum possible value of 1.
- In panel (c), none of the leverages is greater than 0.364.

Residuals

- The ***raw residual*** is $\hat{\epsilon}_i = Y_i - \hat{Y}_i$.
- **Under ideal circumstances such as a reasonably large sample and no outliers or high-leverage points, the raw residuals are approximately *N*(0, $\hat{\sigma}_\epsilon^2$)** so absolute values greater than $2\hat{\sigma}_\epsilon^2$ are outlying and greater than $3\hat{\sigma}_\epsilon^2$ are extremely outlying.
- However, circumstances are often not ideal. **When residual outliers occur at high-leverage points, they can so distort the least-squares fit that they are not seen to be outlying.**
- **The problem in these cases is that $\hat{\epsilon}_i$ is not close to ϵ_i because of the bias in the least-squares fit. The bias is due to residual outliers themselves.**
- **The standard error of $\hat{\epsilon}_i$ is $\hat{\sigma}_\epsilon\sqrt{1 - H_{ii}}$, so the raw residuals do not have a constant variance, and those raw residuals with large leverages close to 1 are much less variable than the others.**
- To fix the problem of nonconstant variance, one can use the ***standardized residual***, sometimes called the ***internally studentized residual***, which is

$$\frac{\hat{\epsilon}_i}{\hat{\sigma}_\epsilon\sqrt{1 - H_{ii}}}$$

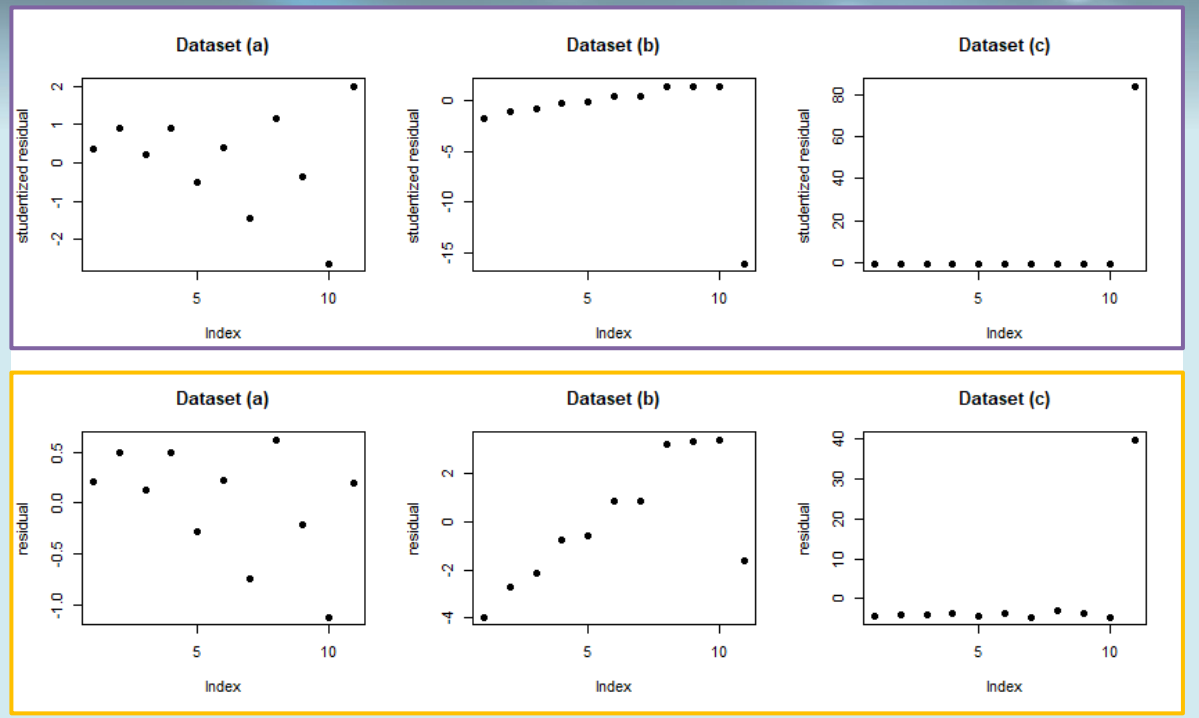
- There is still another problem with standardized residuals. **An extreme residual outlier can inflate $\hat{\sigma}_\epsilon$, causing the standardized residual for the outlying point to appear too small.**
- The solution is to redefine the *i*th studentized residual with an $\hat{\sigma}_\epsilon$ that does not use the *i*th data point.
- Thus, the ***externally studentized residual***, often called ***rstudent***, is defined to be

$$\hat{\epsilon}_i / \hat{\sigma}_{\epsilon\,(-i)}\sqrt{1 - H_{ii}}$$

where $\hat{\sigma}_{\epsilon\,(-i)}$ is the estimate of σ computed by fitting the model to the data with the *i*th observation deleted (The notation (–*i*) means the deletion of the *i*th observation)

Residuals - example

- The top row of Fig. shows the externally studentized residuals Case #11 is correctly identified as a residual outlier in data sets (b) and (c) and also correctly identified in data set (a) as not being a residual outlier. The bottom row of Fig. shows the raw residuals, rather than the externally studentized residuals.
- It is not apparent from the raw residuals that in data set (b), case #11 is a residual outlier.
- This shows the inappropriateness of raw residuals for the detection of outliers, especially when there are high-leverage points.




```
# Residuals example - Externally studentized residuals - the function # -
rstudent()
par(mfrow=c(2,3),lwd=1,pch=19)
plot(rstudent(lm(y~x)),ylab="studentized residual",main="Dataset (a)")
plot(rstudent(lm(y2~x)),ylab="studentized residual",main="Dataset (b)")
plot(rstudent(lm(y~x2)),ylab="studentized residual",main="Dataset (c)")
plot(residuals(lm(y~x)),ylab="residual",main="Dataset (a)")
plot(residuals(lm(y2~x)),ylab="residual",main="Dataset (b)")
plot(residuals(lm(y~x2)),ylab="residual",main="Dataset (c)")
```

Top row: *Externally studentized residuals for the data sets used before (see the code); Case #11 is an outlier in data sets (**b**) and (**c**) but not in data set (**a**).*
Bottom row: *Raw residuals for the same three data sets as in the top row. For data set (**b**), the raw residual does not reveal that case #11 is outlying.*

Cook's Distance

- **A high-leverage value or a large absolute externally studentized residual indicates only a *potential* problem with a data point**.
- Neither tells how much influence the data point actually has on the estimates.
- For that information, we can use *Cook's distance*, often called *Cook's D*, **which measures how much the fitted values change if the *i*th observation is deleted**.
- **We say that Cook's D measures influence, and any case with a large Cook's D is called a high-influence case. Leverage and studentized residual alone do not measure influence.**

Cook's Distance, cont.

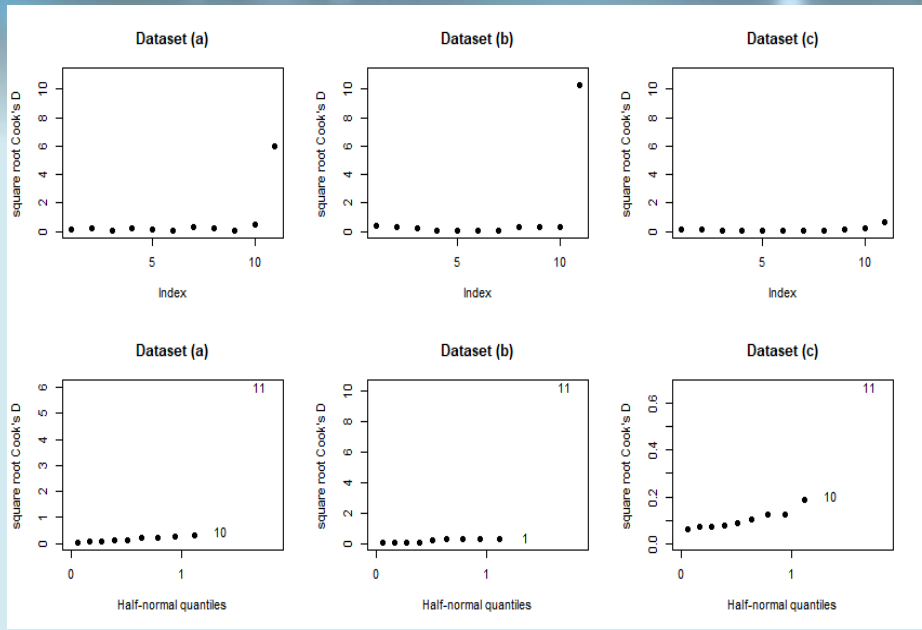
- Let $\hat{Y}_{j\,(-i)}$ be the *j*th fitted value using estimates of the βs obtained with the *i*th observation deleted.
- Then Cook's D for the *i*th observation is

$$\frac{\sum_{j=1}^{n}\left\{\hat{Y}_j - \hat{Y}_{j\,(-i)}\right\}^2}{(p+1)s^2}$$

- The numerator in is the sum of squared changes in the fitted values when the *i*th observation is deleted.
- The denominator standardizes this sum by dividing by the number of estimated parameters and an estimate of $\hat{\sigma}_{\epsilon}^2$
- **One way to use Cook's D is to plot the values of Cook's D against case number and look for unusually large values.**
- However, it can be difficult to decide which, if any, values of Cook's D are outlying. Of course, some Cook's D values will be larger than others, but are any so large as to be worrisome?
- To answer this question, a half-normal plot of values of Cook's D, or perhaps of their square roots, can be useful. Neither Cook's D nor its square root is normally distributed, so one does not check for linearity.
- Instead, one looks for values that are "detached" from the rest.

Example: Cook's Distance

- The three columns of Fig. show the values of square roots of Cook's D for the three simulated data as before.
- In the top row, the square roots of Cook's D values are plotted versus case number (index).
- The bottom row contains half-normal plots of the square roots of the Cook's D values.
- In all panels, case #11 has the largest Cook's D, indicating that one should examine this case to see if there is a problem.
- In data set (a), case #11 is a high-leverage point and has high influence despite not being a residual outlier.
- In data set (b), where case #11 is both a high-leverage point and a residual outlier, the value of Cook's D for this case is very large, larger than in data set (a).
- In data set(c), where case #11 has low leverage, all 11 Cook's D values are reasonably small, at least in comparison with data sets (a) and (b), but case #11 is still somewhat outlying.




```
par(mfrow=c(2,3),cex.axis=1,cex.lab=1,lwd=1,pch=19)
plot(sqrt(cooks.distance(lm(y~x))),ylab=("square root Cook's D"),cex=1,main="Dataset (a)",
ylim=c(0,11))
plot(sqrt(cooks.distance(lm(y2~x))),ylab=("square root Cook's D"),cex=1,main="Dataset (b)",
ylim=c(0,11))
plot(sqrt(cooks.distance(lm(y~x2))),ylab=("square root Cook's D"),cex=1,main="Dataset (c)",
ylim=c(0,11))

halfnorm(sqrt(cooks.distance(lm(y~x))),ylab=("square root Cook's D"),cex=1,main="Dataset (a)",
xlim=c(0,1.85))
halfnorm(sqrt(cooks.distance(lm(y2~x))),ylab=("square root Cook's D"),cex=1,main="Dataset (b)",
xlim=c(0,1.85))
halfnorm(sqrt(cooks.distance(lm(y~x2))),ylab=("square root Cook's D"),cex=1,main="Dataset (c)",
xlim=c(0,1.85))
```

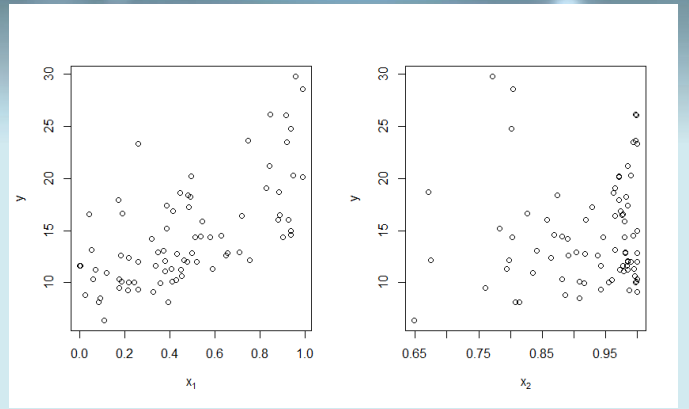
Top row: *Square roots of Cook's D for the simulated data plotted against case number.* **Bottom row:** *Half-normal plots of square roots of Cook's D.* **Data set (a)** *Case #11 has high leverage. It is not a residual outlier but has high influence nonetheless.* **Data set (b)** *Case #11 has high leverage and is a residual outlier. It has higher influence (as measured by Cook's D) than in data set* **(a)**. **Data set (c)** *Case #11 has low leverage but is a residual outlier. It has much lower influence than in data sets* **(a)** *and* **(b)**. **Note:** *In the top row, the vertical scale is kept constant to emphasize differences among the three cases.*

Checking Model Assumptions: Nonlinearity

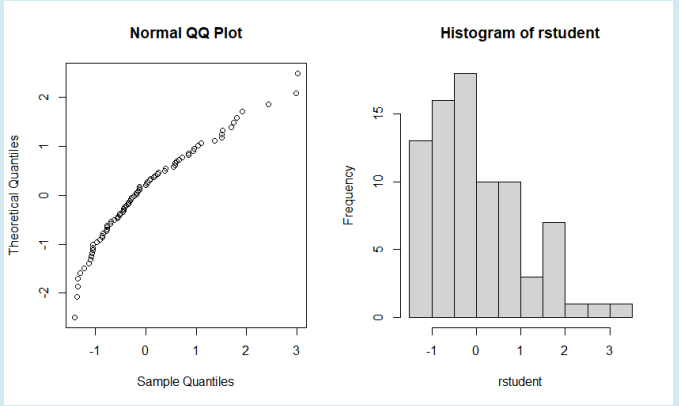
- *If a plot of the residuals versus a predictor variable shows a systematic nonlinear trend, then this is an indication that the effect of that predictor on the response is nonlinear*.
- **Nonlinearity causes biased estimates and a model that may predict poorly. Confidence intervals, which assume unbiasedness, can be seriously in error if there is nonlinearity.**
- The value $100(1 - \alpha)\%$ is called the nominal value of the coverage probability of a confidence interval and is guaranteed to be the actual coverage probability only if all modelling assumptions are met.
- Response transformation, polynomial regression, and nonparametric regression (e.g., splines and loess) are common solutions to the problem of nonlinearity.

Checking Model Assumptions: Nonlinearity, cont.

- Data were simulated to illustrate some of the techniques for diagnosing problems.
- In the example there are two predictor variables, X_1 and X_2.
- The assumed model is multiple linear regression, $Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + e_i$.
- *Upper row figure, shows the responses plotted against each of the predictors, suggests that the errors are heteroskedastic because there is more vertical scatter on the right sides of the plots.*
- Otherwise, it is not clear whether there are other problems with the data or the model.
  - **The point here is that plots of the raw data often fail to reveal all problems.**
- Lower row figure contains a normal plot and a histogram of the residuals—the externally standardized residuals (rstudents) are used in all examples of this chapter.
- *Notice the right skewness which suggests that a response transformation to remove right skewness, such as, a square-root or log transformation, should be investigated*.



Simulated data. Responses plotted against the two predictor variables



Simulated data. Normal plot and histogram of the studentized residuals. Right skewness is evident and perhaps a square root or log transformation of Y would be helpful.

Why do we need transformation?

- A regression model is composed by a **deterministic and a random component**, which rely on different assumptions.
- Among others, these assumptions can be summarized as follows:
- **Normality (N):** The conditional distribution of y given x follows a normal distribution. This is an optional, but often desired assumption.
- **Homoscedasticity (H):** The conditional variance of y given x is constant.
- **Linearity (L):** The conditional expectation of the outcome variable y given the continuous covariates x is a linear function in x.

Nonlinear Regression

- Often we can derive a theoretical model relating predictor variables and a response, but the model we derive is not linear.
- In particular, models derived from economic theory are commonly used in finance are not linear.
- The nonlinear regression model is

$$Y_i = f(X_i; \beta) + \epsilon_i$$

where Y_i is the response measured on the i-th observation, Xi is a vector of observed predictor variables for the i-th observation, f(· ; ·) is a known function, β is an unknown parameter vector, and $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. with mean 0 and variance σ2.

The least-squares estimate $\hat{\beta}$ minimizes

$$\sum_{i=1}^{n} \{Y_i - f(X_i; \beta)\}^2$$

The predicted values are $\hat{Y}_i = f(X_i; \hat{\beta})$ and the residuals are $\hat{\epsilon}_i = Y_i - \hat{Y}_i$.

- Since the model is nonlinear, finding the least-squares estimate requires nonlinear optimization.
- Because of the importance of nonlinear regression, almost every statistical software package will have routines for nonlinear least squares estimation. This means that most of the difficult programming has already been done for us.

Zero-coupon bonds

- **Zero-coupon bond (the discount bond) is a type of bonds that does not pay any regular interest payments to the investor (bond is a fixed income instrument that represents a loan made by an investor to a borrower -typically corporate or governmental).**
- Instead, **you purchase the bond for a discount and then when it matures, you can get back the face value of the bond.**
  - Thus, money invested in Zero Coupon Bond does not generate a regular interest during the tenure.
- This is a long-term type of investment that can provide nice yields (a risk-free interest over a long period of time).
- Zero-coupon bonds can help investors to avoid gift taxes, but they also create phantom income tax issues. (Phantom income is income that a business owner has to pay taxes on despite not having received any cash)
- Zero coupon bonds have a **higher default risk than traditional bonds**. The reason behind this is that companies do not have to make regular interest payments to the investors. They just keep all of the money and do with it as they please.
- **Price of bond = Face Value / (1 + r) n**
  - **Face value** = Future value or maturity value of the bond
  - **r** = Required rate of return or interest rate
  - **n** = Number of years until maturity

Example: Simulated bond prices

- Consider prices of par $1000 zero-coupon bonds issued by a particular borrower, perhaps the Federal government or a corporation. Suppose that there are several times to maturity, being denoted by Ti. Suppose also that the yield to maturity is a constant, say r.

- The rate r is determined by the market and can be estimated from prices. Under the assumption of a constant value of r, the present price of a bond with maturity Ti is

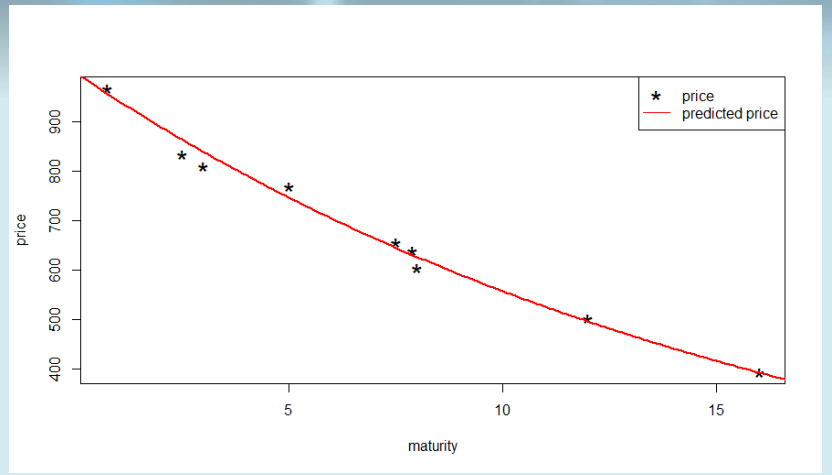
$$P_i = 1000 \exp(-rT_i)$$

Example: Simulated bond prices, cont

- In $P_i = 1000\exp(-rT_i)$ there is some random variation in the observed prices.
- One reason is that **the price of a bond can only be determined by the sale of the bond, so the observed prices have not been determined simultaneously.**
- **Prices that may no longer reflect current market values are called stale.**
- **Each bond's price was determined at the time of the last trade of a bond of that maturity, and r may have had a somewhat different value then. It is only as a function of time to maturity that r is assumed constant, so r may vary with calendar time.**
- Thus, we augment model by including a noise term to obtain the regression model $P_i = 1000\exp(-rT_i) + \boldsymbol{\epsilon_i}$
- An estimate of r can be determined by least squares, that is, by minimizing over r the sum of squares:
- $\sum_{i=1}^{n}\{P_i - 1000\exp(-rT_i)\}^2$
- The least-squares estimator is denoted by $\hat{\beta}$.

Example: Simulated bond prices, cont

- Since it is unlikely that market data will have a constant r, this example uses simulated data.
- The data were generated with r fixed at 0.06 and plotted in Fig.
- The nonlinear least-squares estimate of r was found using R's **`nls()`** function.
- Nonlinear optimization requires starting values for the parameters, and a starting value of 0.04 was used for r.



- The output is:

Formula: price ~ 1000 * exp(-r * maturity)

Parameters:

	Estimate	Std. Error	t value	Pr(>|t|)
r	0.05850	0.00149	39.3	1.9e-10 ***

Residual standard error: 20 on 8 degrees of freedom

Number of iterations to convergence: 4

Achieved convergence tolerance: 5.53e-08

```
bondprices = read.table("bondprices.txt", header = TRUE)
attach(bondprices)
fit = nls(price ~ 1000 * exp(-r * maturity), start =
list(r = 0.04))
summary(fit)
par(mfrow=c(1,1))
plot(maturity,price,pch="*",cex = 2)
grid = seq(0, 20, length=201)
price_grid = 1000*exp(-0.0585*grid)
lines(grid,price_grid, lwd = 2, col = "red")
legend("topright",c("price","predicted
price"),pch=c("*",NA), col = c("black","red"),
lty=c(NA,1),pt.cex=c(2,1))
```

Example: Simulated bond prices, cont

- How to select a starting point for the **`nls()` `function`**:
- Estimate the linear model:

```
temp = lm(log(price)~log(maturity))
summary (temp)
exp((temp$coefficients[2]))
fit2 = nls(price ~ 1000 * exp(-r *
maturity), start = list(r =
exp((temp$coefficients[2]))))
summary(fit2)
```

Coefficients from the log linear function:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.93941    0.08753  79.283 1.34e-11 ***
log(maturity) -0.26866    0.04698  -5.719 0.000721 ***
```

Coefficients from the nonlinear function

Parameters:

```
  Estimate Std. Error t value Pr(>|t|)
r 0.058498   0.001488   39.31 1.93e-10 ***
```

Another advanced way: the Levenberg-Marquardt Nonlinear Least-Squares. It is more robust and if the initial guess is far from the mark, the algorithm can still find an optimal solution.

```
library (minpack.lm)
fit3 <- nlsLM(price ~ 1000 * exp(-r * maturity), start = list(r = 10))
summary (fit3)
```

Nonlinear vs linear

- In nonlinear regression the **form of the regression function is nonlinear** but known up to a few unknown parameters.
- For example, the regression function has an exponential form in the last example ($1000\exp(-rT_i) + \epsilon_i$).
- For this reason, nonlinear regression would best be **called nonlinear parametric regression to distinguish it from nonparametric regression, where the regression function is also nonlinear but not of a known parametric form**.
- **Polynomial regression** may appear to be nonlinear since polynomials are nonlinear functions. For example, the quadratic regression model

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i$$

is nonlinear in X_i. However, by defining X_i^2 as a second predictor variable, this model is linear in (X_i, X_i^2) and therefore is an example of multiple linear regression. What makes that model linear is that **the right-hand side is a linear function of the parameters $\boldsymbol{\beta_0}$, $\boldsymbol{\beta_1}$, and $\boldsymbol{\beta_2}$, and therefore can be interpreted as a linear regression with the appropriate definition of the variables.**

- In contrast, the exponential model

$$Y_i = \beta_0 e^{\beta_1 X_i} + \epsilon_i$$

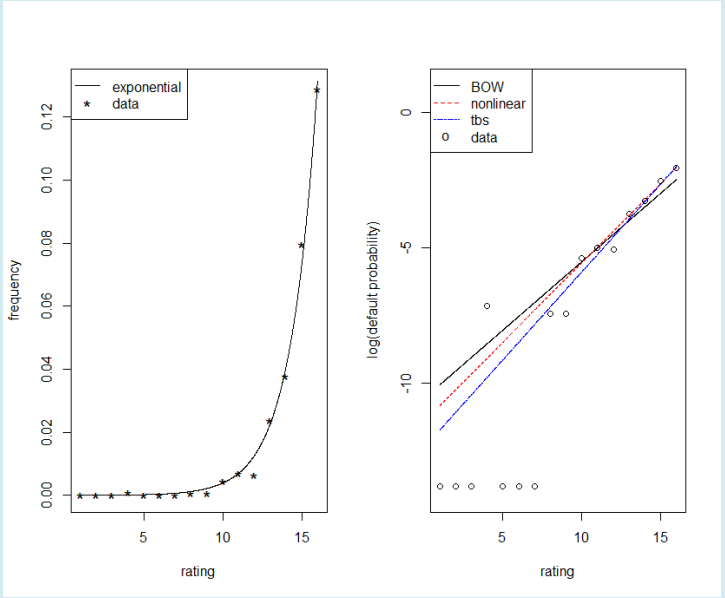
is nonlinear in the parameter β_1, so it cannot be made into a linear model by redefining the predictor variable.

Example: Estimating default probabilities

- This example illustrates both nonlinear regression and the detection of heteroskedasticity by residual plotting.
- **Credit risk is the risk to a lender that a borrower will default on contractual obligations, for example, that a loan will not be repaid in full.**
- A key parameter in the determination of credit risk is **the probability of default**.
- Bluhm, Overbeck, and Wagner (2003) illustrate how one can calibrate Moody's credit rating to estimate default probabilities.
- These authors use observed default frequencies for bonds in each of 16 Moody's ratings from Aaa (best credit rating) to B3 (worse rating).
- They convert the credit ratings to a 1 to 16 scale (Aaa = 1, . . . , B3 = 16).
- Figure on the right hand side, shows default frequencies (as fractions, not percentages) plotted against the ratings. The data are from Bluhm, Overbeck, and Wagner (2003).
- The relationship is clearly nonlinear. Bluhm, Overbeck, and Wagner used a nonlinear model, specifically

$$\Pr\{default|rating\} = \exp\{\beta_0 + \beta_1 rating\}$$

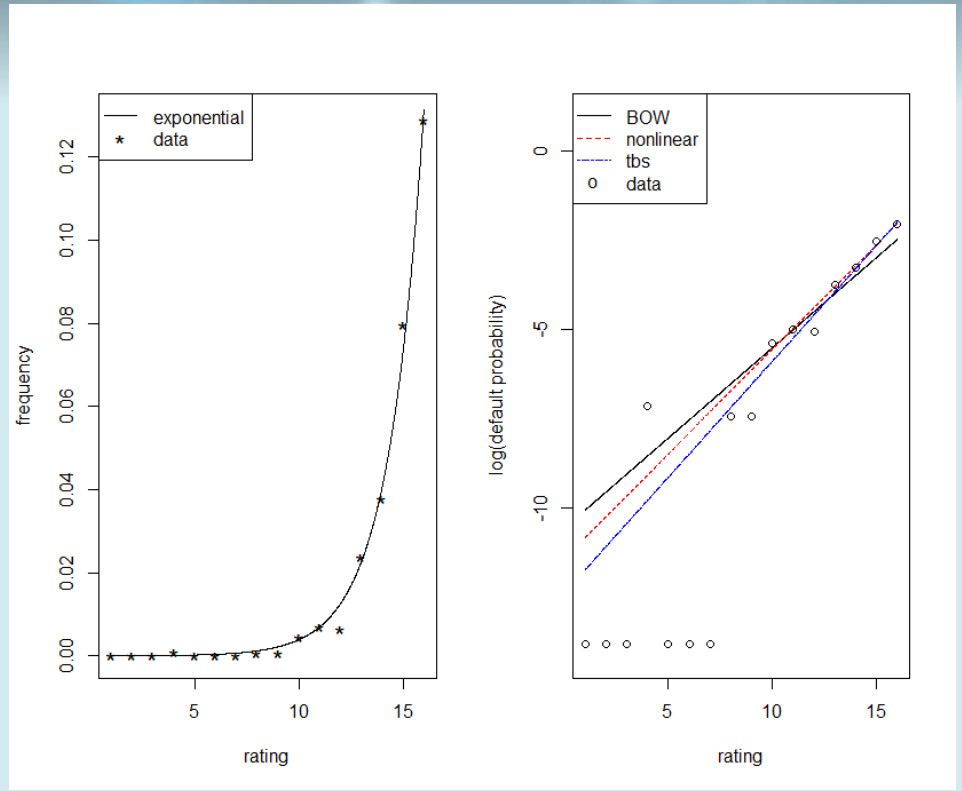
- To use this model they fit a linear function to the logarithms of the default frequencies. **One difficulty with doing this is that six of the default frequencies are zero giving a log transformation of −∞.**



Bluhm, C., Overbeck, L., and Wagner, C. (2003) *An Introduction to Credit Risk Modelling*, Chapman & Hall/CRC, Boca Raton, FL.

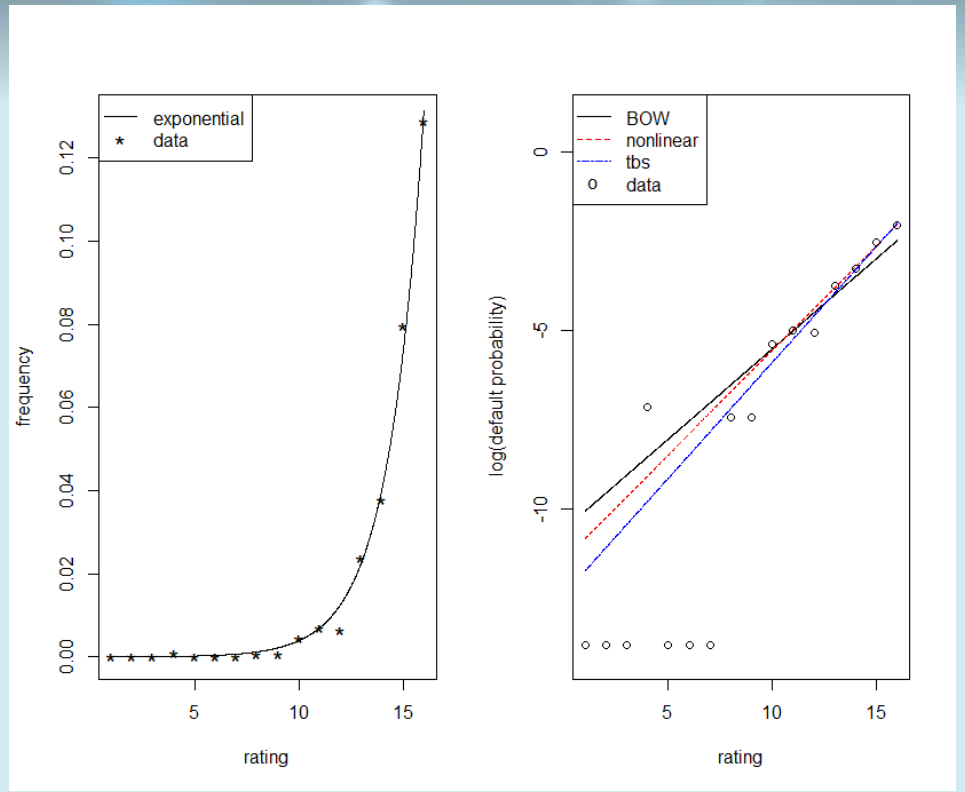
Example: Estimating default probabilities, cont

- (*left*) Default frequencies with an exponential fit. “Rating” is a conversion of the Moody’s rating to a 1 to 16-point scale as follows: 1 = Aaa, 2 = Aa1, 3 = Aa3, 4 = A1, . . . , 16 = B3.
- (*right*) Estimation of default probabilities by Bluhm, Overbeck, and Wagner’s (2003) linear regression with ratings removed that have no observed defaults (**BOW**) and by nonlinear regression with all data (**nonlinear**).
- Because some default frequencies are zero, when plotting the data on a semilog plot,10^{-6} was added to the default frequencies (see in the code). This constant was not added when estimating default frequencies, only for plotting the raw data.
- The six observations along the bottom of the plot are the ones removed by Bluhm, Overbeck, and Wagner. “TBS” is the transform-both-sides estimate, which will be discussed soon.



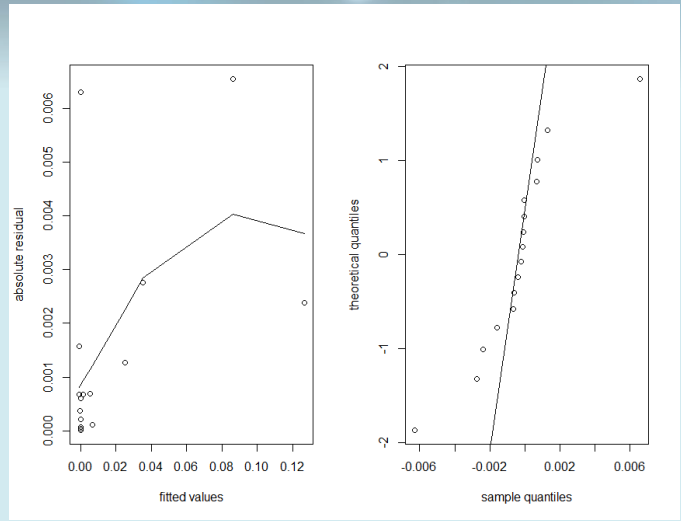
Example: Estimating default probabilities, cont

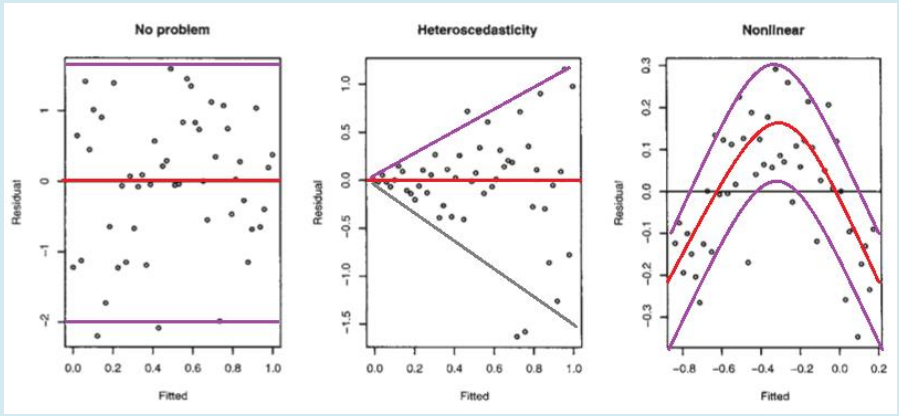
- Bluhm, Overbeck, andWagner (2003) **address the issue of log transformation of 0 by labeling default frequencies equal to zero as "unobserved" and not using them in the estimation process.**
- The problem with their technique is that **they have deleted the data with the lowest observed default frequencies.**
- **This biases their estimates of default probabilities in an upward direction**.
- Bluhm, Overbeck, and Wagner argue that an observed default frequency of zero does not imply that the true default probability is zero.
- This is certainly true. However, **the default frequencies, even when they are zero, are unbiased estimates of the true default probabilities**.
- We can avoid the bias of their method by using nonlinear regression with model.
- **The advantage of fitting $\Pr\{default|rating\} = \exp\{\beta_0 + \beta_1 rating\}$ by nonlinear regression is that it avoids the use of a logarithm transformation thus allowing the use of all the data, even data with a default frequency of zero.**
- The fits by the Bluhm, Overbeck, and Wagner method and by nonlinear regression using model $\Pr\{default|rating\} = \exp\{\beta_0 + \beta_1 rating\}$ are shown in Fig (*Right panel*) with a log scale on the vertical axis so that the fitted functions are linear. **Notice that at good credit ratings the estimated default probabilities are lower using nonlinear regression compared to Bluhm, Overbeck, and Wagner's biased method.**
- The differences between the two sets of estimated default probabilities can be substantial. Bluhm, Overbeck, and Wagner estimate the default probability of an Aaa bond as 0.005 %. In contrast, the unbiased estimate by nonlinear regression is only 40% of that figure, specifically, 0.0020 %.
- Thus, the bias in the Bluhm, Overbeck, and Wagner estimate leads to a substantial overestimate of the credit risk of Aaa bonds and similar overestimation at other good credit ratings



Example: Estimating default probabilities, cont

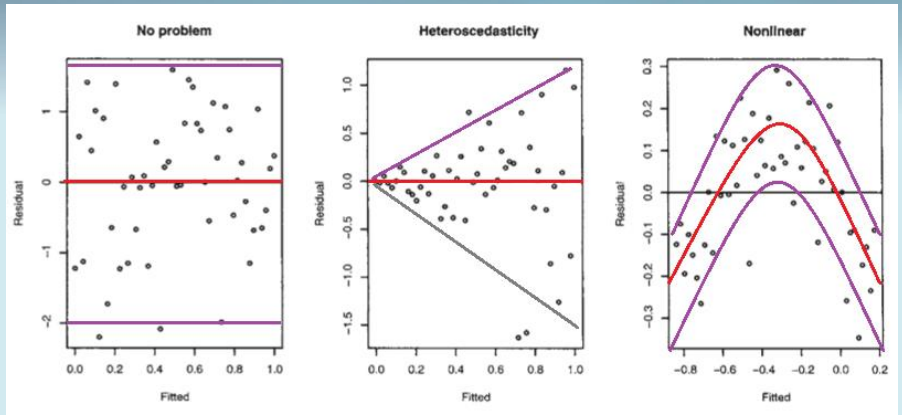
- A plot of the absolute residuals versus the fitted values in Fig. Top Left gives a clear indication of heteroskedasticity.
  - Heteroskedasticity cause inefficient estimates
  - Later, we will fix this problem by a variance-stabilizing transformation.
- Figure on the right hand side is a normal probability plot of the residuals.
- Outliers with both negative and positive values can be seen.
- These are due to the nonconstant variance and are not necessarily a sign of nonnormality.
- This plot illustrates the danger of attempting to interpret a normal plot when the data have a nonconstant variance.
- One should apply a variance-stabilizing transformation first before checking for normality.





The example: the conditional mean (red) and conditional mean ± (roughly!) twice the conditional standard deviation (purple)

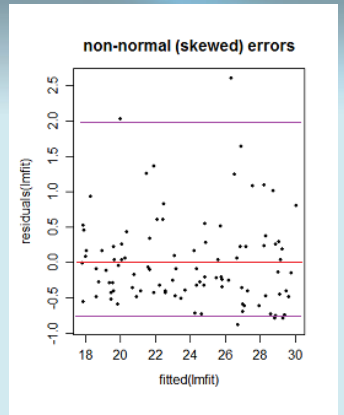
Examples of the figures fitted values vs residuals



The example: the conditional mean (red) and conditional mean ± (roughly!) twice the conditional standard deviation (purple).
The first figure shows no issues with the model.

The second plot shows the mean residual doesn't change with the fitted values (and so is doesn't change with x), but the spread of the residuals (and hence of the y's about the fitted line) is increasing as the fitted values (or x) changes. That is, the spread is not constant. Heteroskedasticity.

The third plot shows that the residuals are mostly negative when the fitted value is small, positive when the fitted value is in the middle and negative when the fitted value is large. That is, the spread is approximately constant, but the conditional mean is not - the fitted line doesn't describe how y behaves as x changes, since the relationship is curved (non-linear).



The purple lines still represent a (very) roughly 95% interval, but it's no longer symmetric.

Transformation

- Suppose we have a theoretical model that states that in the absence of any noise, $Y_i = f(X_i; \beta)$.
- This model is identical to the model $h\{Y_i\} = h\{f(X_i; \beta)\}$.

where ***h*** is any one-to-one function, such as, a strictly increasing function.

- In the absence of noise, one choice of *h* is as good as any other and one might as well stick with model $Y_i = f(X_i; \beta)$, but when noise exists, this is no longer true.
- When we have noisy data,

$$h\{Y_i\} = h\{f(X_i; \beta)\} + \boldsymbol{\epsilon_i}$$

can be converted to the nonlinear regression model. It is called the **transform-both-sides (TBS) regression model** because both sides have been transformed by the same function *h*.

- Typically, *h* will be one of the Box–Cox transformations and *h* is chosen to stabilize the variation and to induce nearly normally distributed errors.
- To estimate β for a fixed *h*, one minimizes

$$\sum_{i=1}^{n} \left[h\{Y_i\} - h\{f(X_i; \hat{\beta})\}\right]^2$$

- Various choices of h can be compared by residual plots. The h that gives approximately normally distributed residuals with a constant variance is used for the final analysis

Transforming Only the Response

- The so-called Box–Cox transformation model is

$$Y_i^{(a)} = \beta_0 + X_{i,1}\beta_1 + \cdots + X_{i,p}\beta_p + \epsilon_i,$$

where $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. N(0, σ^2) for some σ.

- **If only the response is transformed.**
- The goal of transforming the response is to achieve three objectives:

  1. a simple model: $Y_i^{(a)}$ is linear in predictors $X_{i,1}, \ldots, X_{i,p}$ and in the parameters $\beta_1, \ldots, \beta_p$;

  2. constant residual variance; and

  3. Gaussian noise.

- In contrast, 2 and 3 but not 1 are the goals of the TBS model (transform both sides).

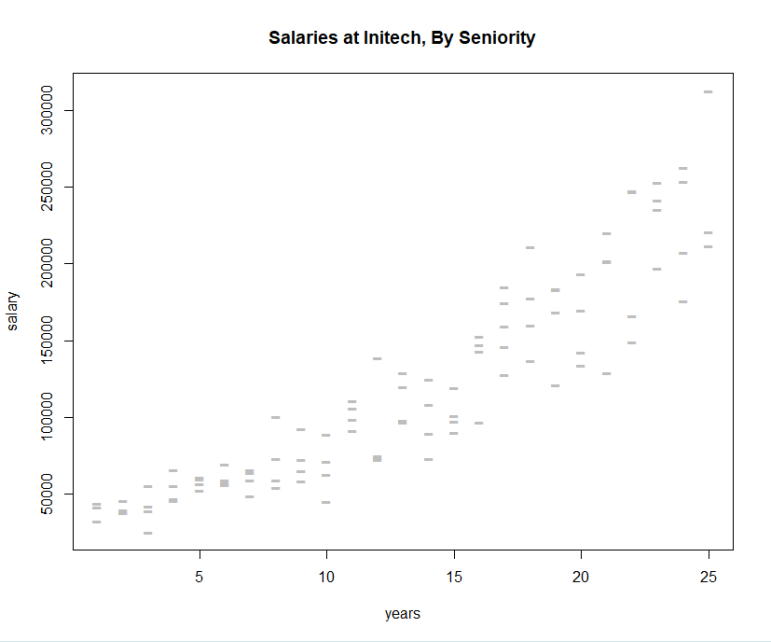
Transforming Only the Response, cont.

- This model $Y_i^{(a)} = \beta_0 + X_{i,1}\beta_1 + \cdots + X_{i,p}\beta_p + \epsilon_i$ was introduced by Box and Cox (1964) who suggested estimation of α by maximum likelihood.
  - Maximum likelihood estimation is a method that determines values for the parameters of a model. The parameter values are found such that they maximise the likelihood that the process described by the model produced the data that were actually observed.
  - The method of maximum likelihood selects the set of values of the model parameters that maximizes the likelihood function. Intuitively, this maximizes the "agreement" of the selected model with the observed data.
- The function `boxcox()` in R's MASS package will compute the profile log-likelihood for α along with a confidence interval.
- Usually, α is taken to be some round number, e.g., −1, −1/2, 0, 1/2,or 1, in the confidence interval.
- The reason for selecting one of these numbers is that then the transformation is readily interpretable, that is, it is the square root, log, inverse, or some other familiar function.
- After α has been selected in this way, $\beta_0, \ldots, \beta_p$ and σ^2 can be estimated by regressing $Y_i^{(a)}$ on $X_{i,1}, \ldots, X_{i,p}$.

Example : transformation

- Let's look at some salary data from some company Initech.
- We will try to model salary as a function of years of experience. The data can be found in initech.csv

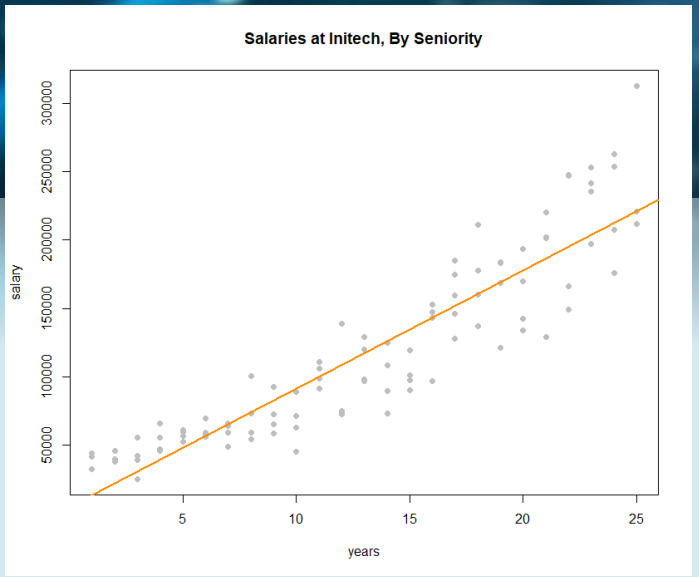
```
initech = read.csv("initech.csv")
plot(salary ~ years, data = initech, col = "grey", pch = 45, cex = 2, main = "Salaries at
Initech, By Seniority")
initech_fit = lm(salary ~ years, data = initech) # We first fit a simple linear model
summary(initech_fit)

Call:
lm(formula = salary ~ years, data = initech)

Residuals:
   Min     1Q Median     3Q    Max
-57225 -18104    241  15589  91332

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)     5302       5750    0.92     0.36
years           8637        389   22.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27400 on 98 degrees of freedom
Multiple R-squared:  0.834,        Adjusted R-squared:  0.832
F-statistic:  493 on 1 and 98 DF,  p-value: <2e-16
```



This model appears significant. Your interpretation?

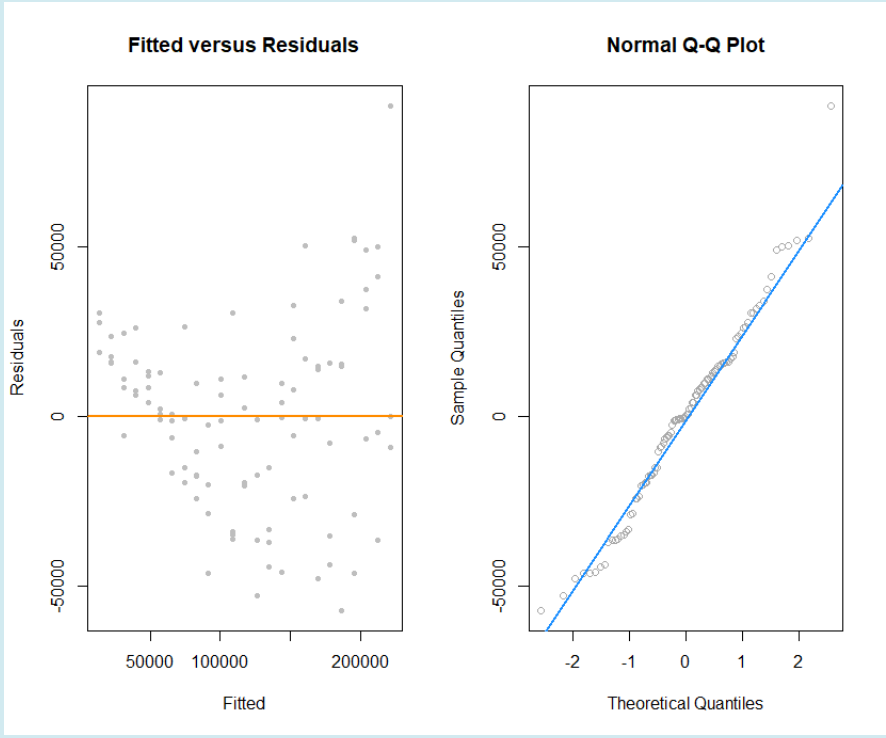
Example : transformation

```
plot(salary ~ years, data = initech, col = "grey", pch = 20, cex = 1.5,
     main = "Salaries at Initech, By Seniority")
abline(initech_fit, col = "darkorange", lwd = 2)
```



- Adding the fitted line to the plot, we see that the linear relationship appears correct (Figure on Top)

```
par(mfrow = c(1, 2))
plot(fitted(initech_fit), resid(initech_fit), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
qqnorm(resid(initech_fit), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(initech_fit), col = "dodgerblue", lwd = 2)
```



- However, from the fitted versus residuals plot it appears there is non-constant variance (there is heteroskedasticity).
  - Specifically, the variance increases as the fitted value increases.
- Though the residuals are close no normal in terms of distribution

- Thus – we need to stabilize variance!

Example : Variance Stabilizing Transformations

- Recall the fitted value is our estimate of the mean at a particular value of x. Under our usual assumptions, $\epsilon \sim N(0,\sigma^2)$ and thus $Var[Y|X=x]=\sigma^2$, which is a constant value for any value of x.
- However, here we see that the variance is a function of the mean, $Var[Y|X=x]=h(E[Y|X=x])$.
- In this case, h is some increasing function.
- In order to correct this, we would like to **find some function of Y, g(Y), such that, $Var[g(Y)|X=x]=c$ where c is a constant that does not depend on the mean**.
- A transformation that accomplishes this is called a **variance stabilizing transformation.**
- **A common variance stabilizing transformation (VST) when we see increasing variance in a fitted versus residuals plot is log (Y) is a log transformation.**
  - If the values of a variable range over more than one order of magnitude and the variable is strictly positive, then replacing the variable by its logarithm is likely to be helpful.
- *A reminder, that for our purposes, log and ln are both the natural log. R uses log to mean the natural log, unless a different base is specified.*
- We will now use a model with a log transformed response for the *Initech* data, $\mathbf{\log(Y_i)=\beta_0+\beta_1 x_i+\epsilon_i}$.
  - **Note, if we re-scale the model from a log scale back to the original scale of the data, we now have $Y_i=\exp(\beta_0+\beta_1 x_i)\cdot\exp(\epsilon_i)$ which has the errors entering the model in a multiplicative fashion.**
- Fitting this model in R requires only a minor modification to our formula specification.

Example : transformation - Variance Stabilizing Transformations, cont

```
initech_fit_log = lm(log(salary) ~ years, data = initech)
```

- Note that while log(y) is considered the new response variable, we do not actually create a new variable in R, but simply transform the variable inside the model formula.

```
plot(log(salary) ~ years, data = initech, col = "grey", pch = 20, cex = 1.5,main = "Salaries at Initech, By Seniority")
abline(initech_fit_log, col = "darkorange", lwd = 2)
```

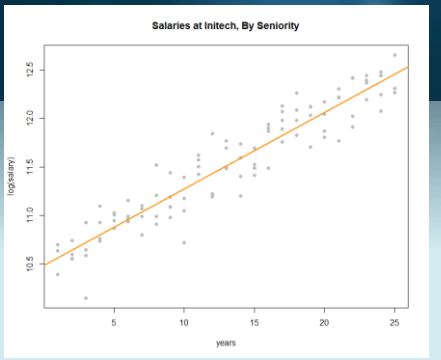
- Plotting the data on the transformed log scale and adding the fitted line, the relationship again appears linear, and we can already see that the variation about the fitted line looks constant.

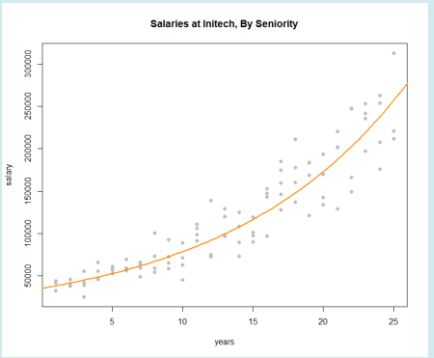
```
plot(salary ~ years, data = initech, col = "grey", pch = 20, cex = 1.5, main = "Salaries at Initech, By Seniority")
curve(exp(initech_fit_log$coef[1] + initech_fit_log$coef[2] * x),from = 0, to = 30, add = TRUE, col = "darkorange", lwd = 2)
```

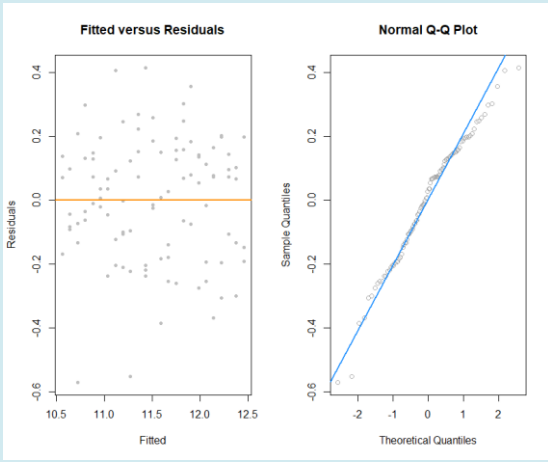
- By plotting the data on the original scale, and adding the fitted regression, we see an exponential relationship. However, this is still a *linear* model, since the new transformed response, log(y), is still a *linear* combination of the predictors.
- The fitted versus residuals plot looks much better. It appears the constant variance assumption is no longer violated.

```
par(mfrow = c(1, 2))
plot(fitted(initech_fit_log), resid(initech_fit_log), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
qqnorm(resid(initech_fit_log), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(initech_fit_log), col = "dodgerblue", lwd = 2)
```

- Comparing the Root Mean Square Error (RMSE) using the original [`sqrt(mean((initech$salary - fitted(initech_fit))^2))`] and transformed response [`sqrt(mean((initech$salary - exp(fitted(initech_fit_log)))^2))`], we also see that the log transformed model simply fits better, with a smaller average squared error.
- Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. **Lower values of RMSE indicate better fit.**







Example : transformation - Variance Stabilizing Transformations, cont

- Comparing the RMSE using the original and transformed response, we also see that the log transformed model simply fits better, with a smaller average squared error.

```
sqrt(mean(resid(initech_fit) ^ 2))
## [1] 27080.16
sqrt(mean(resid(initech_fit_log) ^ 2))
## [1] 0.1934907
```

- That isn't fair, this difference is simply due to the different scales being used. Transforming the fitted values of the log model back to the data scale, we do indeed see that it fits better!

```
sqrt(mean((initech$salary - fitted(initech_fit)) ^ 2))
## [1] 27080.16
sqrt(mean((initech$salary - exp(fitted(initech_fit_log))) ^ 2))
## [1] 24280.36
summary(initech_fit_log)
##
## Call:
## lm(formula = log(salary) ~ years, data = initech)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57022 -0.13560  0.03048  0.14157  0.41366
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.48381    0.04108  255.18   <2e-16 ***
## years        0.07888    0.00278   28.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1955 on 98 degrees of freedom
## Multiple R-squared:  0.8915, Adjusted R-squared:  0.8904
## F-statistic: 805.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

- Again, the transformed response is a *linear* combination of the predictors, $\log(\hat{y}(x)) = \hat{\beta}_0 + \hat{\beta}_1 x = 10.484 + 0.079x$
- But now, if we re-scale the data from a log scale back to the original scale of the data, we now have $\hat{y}(x) = \exp(\hat{\beta}_0)\exp(\hat{\beta}_1 x) = \exp(10.483)\exp(0.079x)$
- **We see that for every one additional year of experience, average salary increases $\exp(0.079) = 1.0822$ times.**
- While using a log transform is possibly the most common response variable transformation, many others exist.
- We will now consider a family of transformations and choose the best from among them, which includes the log transform.

Example : transformation - Box-Cox Transformations

- The Box-Cox method considers a family of transformations on strictly positive response variables,

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, if\ \lambda \neq 0 \\ \log(y), if\ \lambda = 0 \end{cases} \quad (1)$$

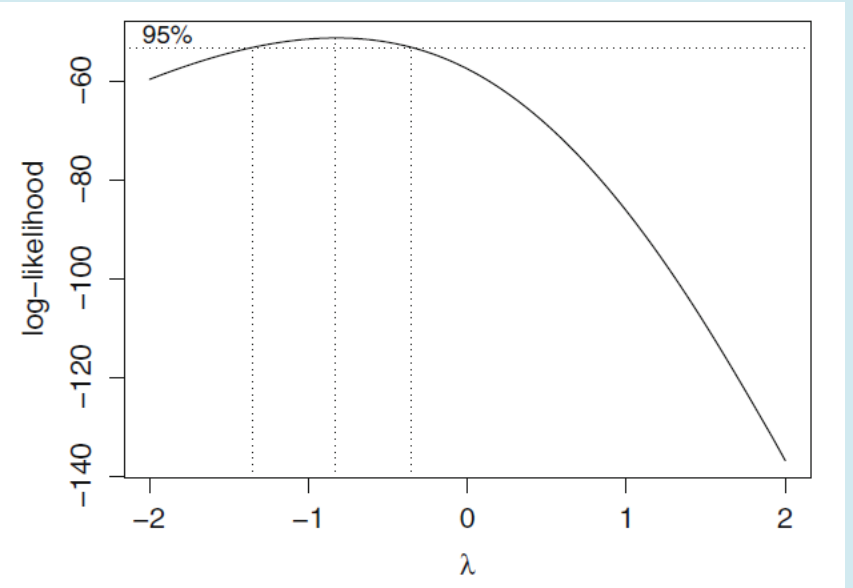
- The λ parameter is chosen by numerically maximizing the log-likelihood,

$$L(\lambda) = -\frac{n}{2}\log\left(\frac{RSS_{\lambda}}{n}\right) + (\lambda - 1)\sum \log(y_i) \quad (2)$$

- Where RSS is Residual Sum of Squares.
- A 100(1-a)% confidence interval for λ is,

$$\left\{\lambda: L(\lambda) > L(\hat{\lambda}) - \frac{1}{2}\chi^2_{1,a}\right\}$$

which R will plot for us to help quickly select an appropriate λ value.



Thus, our goal is to calculate λ, and then

Example : transformation - Box-Cox Transformations, cont.

```
library(MASS); library(faraway)
```

- Here we need the MASS package for the `boxcox()` function, and we will consider a couple of datasets from the faraway package.
- First we will use the savings dataset as an example of using the Box-Cox method to justify the use of no transformation.
- We fit an additive multiple regression model with `sr` (a saving rate) as the response and each of the other variables as predictors.

```
savings_model = lm(sr ~ ., data = savings)
```

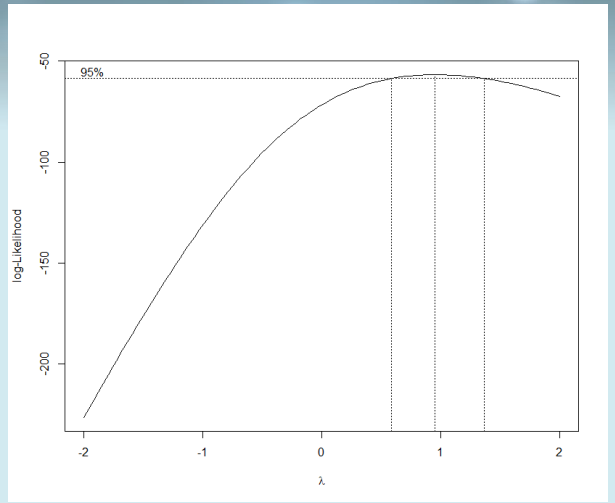
- We then use the boxcox() function to find the best transformation of the form considered by the Box-Cox method.

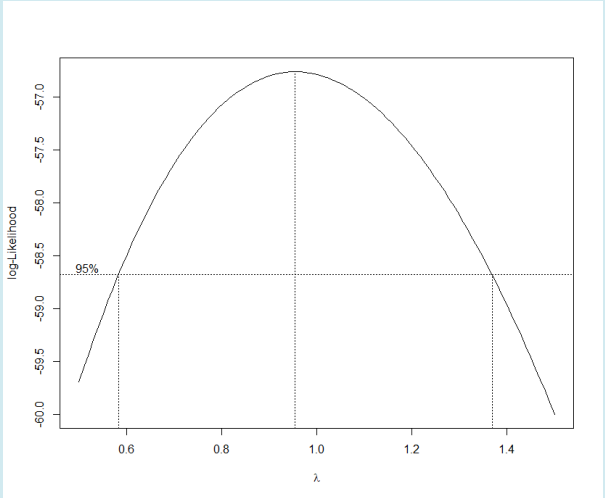
```
boxcox(savings_model, plotit = TRUE)
```

- R automatically plots the log-Likelihood as a function of possible λ values. It indicates both the value that maximizes the log-likelihood, as well as a confidence interval for the λ value that maximizes the log-likelihood.

```
boxcox(savings_model, plotit = TRUE, lambda = seq(0.5, 1.5, by = 0.1))
```

- Note that we can specify a range of λ values to consider and thus be plotted. We often specify a range that is more visually interesting. Here we see that $\lambda = 1$ is both in the confidence interval, and is extremely close to the maximum. This suggests a transformation of the form $\frac{y^{\lambda}-1}{\lambda} = \frac{y^{1}-1}{1} = y - 1$.
- This is essentially not a transformation. It would not change the variance or make the model fit better. By subtracting 1 from every value, we would only change the intercept of the model, and the resulting errors would be the same

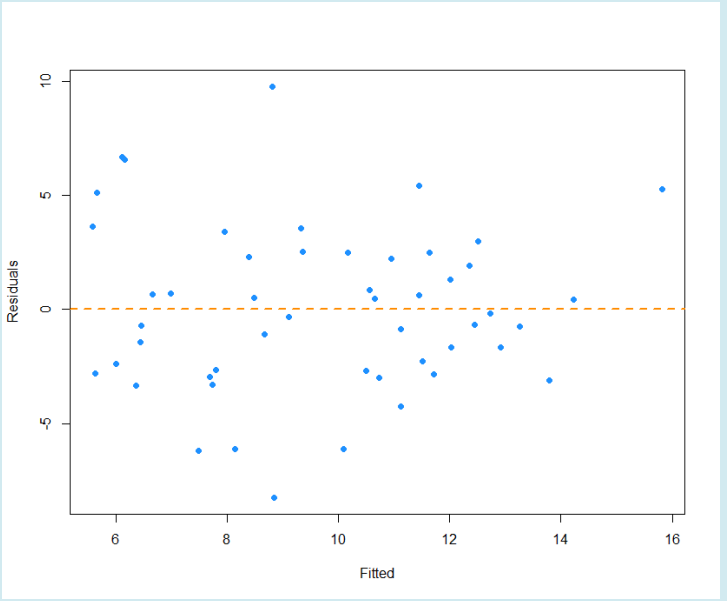




Example : transformation - Box-Cox Transformations, cont.

```
plot(fitted(savings_model), resid(savings_model), col = "dodgerblue", pch = 20, cex = 1.5, xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```

- Looking at a fitted versus residuals plot verifies that there likely are not any issue with the assumptions of this model, which Breusch-Pagan (for heteroskedasticity) and Shapiro-Wilk tests verify:




```
bptest(savings_model)
studentized Breusch-Pagan test
data:  savings_model
BP = 5, df = 4, p-value = 0.3
```

Since the p-value is not less than 0.05, **we fail to reject the null hypothesis (H0=homoscedasticity, implying that variance does not depend on regressors).** Thus, we do not have sufficient evidence to say that heteroscedasticity is present in the regression model.

```
shapiro.test(resid(savings_model))
Shapiro-Wilk normality test
data:  resid(savings_model)
W = 1, p-value = 0.9
```

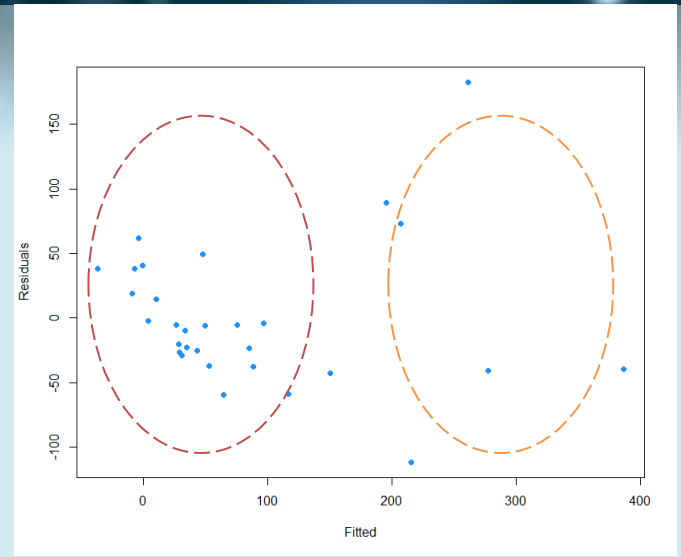
- From the output, the **p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution**. In other words, we can assume the normality of the residuals.

- Given that our residuals are normally distributed and are without heteroscedasticity, that caused our λ was close to 1

Example : Box-Cox transformation (gala dataset)

- Now we will use the gala dataset as an example of using the Box-Cox method to justify a transformation other than log.
- We fit an additive multiple regression model with Species as the response and most of the other variables as predictors.

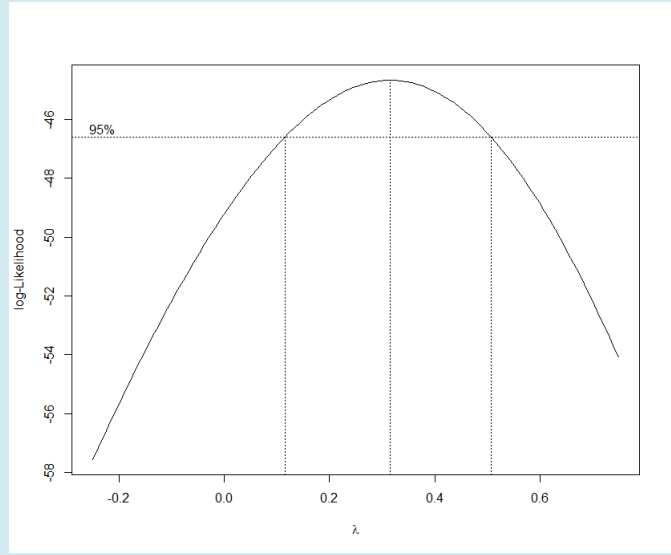
```
gala_model = lm(Species ~ Area + Elevation + Nearest + Scruz +
Adjacent, data = gala)
plot(fitted(gala_model), resid(gala_model), col = "dodgerblue",
pch = 20, cex = 1.5, xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```



- Even though there is not a lot of data for large fitted values, it still seems very clear that the constant variance assumption is violated.

```
boxcox(gala_model, plotit = TRUE)# general impression
boxcox(gala_model, lambda = seq(-0.25, 0.75, by = 0.05), plotit
= TRUE)
```

Using the Box-Cox method, we see that λ=0.3 is both in the confidence interval, and is extremely close to the maximum, which suggests a transformation of the form $\frac{y^{0.3}-1}{0.3}$

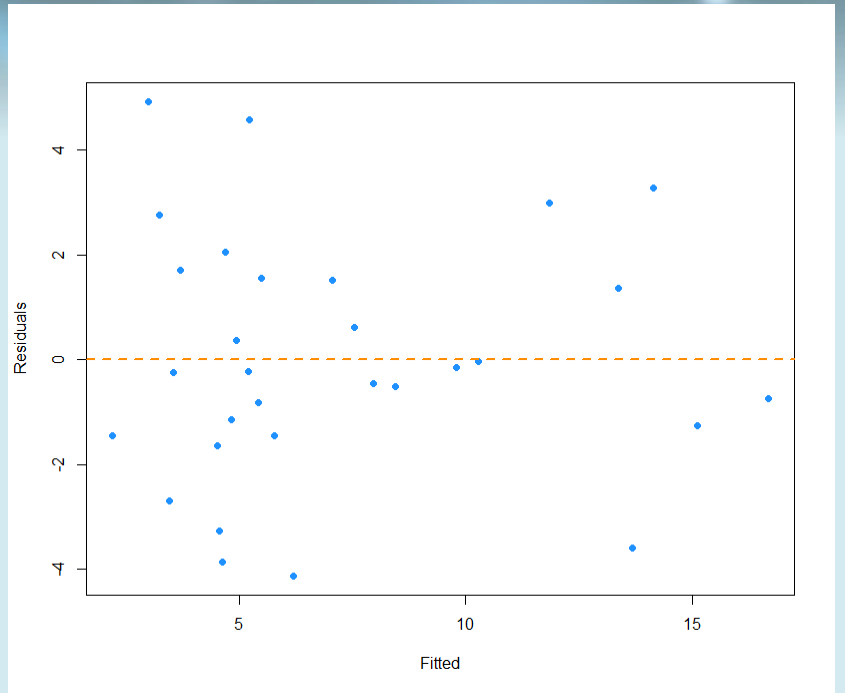


- We then fit a model with this transformation applied to the response.

```
gala_model_cox = lm((((Species ^ 0.3) - 1) / 0.3) ~ Area +
Elevation + Nearest + Scruz + Adjacent, data = gala)
```

Example : Box-Cox transformation (gala dataset), cont.

```
plot(fitted(gala_model_cox),
resid(gala_model_cox), col =
"dodgerblue",      pch = 20, cex = 1.5,
xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col =
"darkorange", lwd = 2)
```



- The resulting fitted versus residuals plot looks much better!

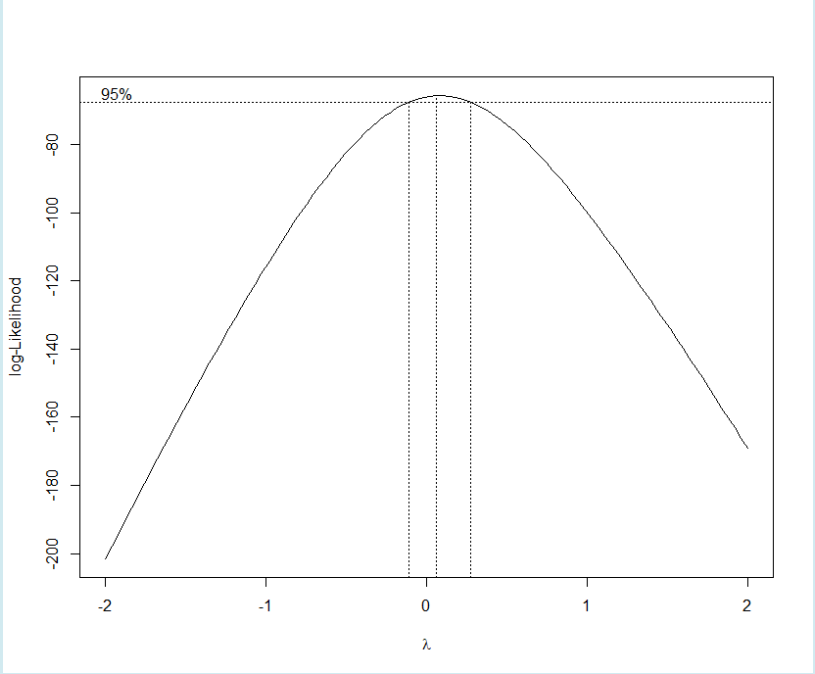
- Lastly, we return to the initech data, and the initech_fit model we had used earlier. Recall, that this was the untransformed model, that we used a log transform to fix.

Example : Box-Cox transformation -> initech_fit model

`boxcox(initech_fit)`

Using the Box-Cox method, we see that λ=0 is both in the interval, and extremely close to the maximum, which suggests a transformation of the form log(y)

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, if\ \lambda \neq 0 \\ \log(y), if\ \lambda = 0 \end{cases}$$



So the Box-Cox method justifies our previous choice of a Log transform!

How to make the things simpler?

- The R package `trafo` offers a simple user friendly framework for selecting a suitable transformation depending on the user needs.

- The collection of selected transformations and estimation methods in the package trafo complement and enlarge the methods that are existing in R so far.

trafo

Data-driven transformations

- Table provides information about the range of the dependent variable that is supported by the transformation.
- **Some transformations are only suitable for positive values of y.**
- This is generally true for the *logarithmic and Box-Cox transformations*.
- **However, in case that the dependent variable contains negative values, the values are shifted by a deterministic shift *s* such that y + s > 0 by default in package trafo**.
- Furthermore, the tables emphasize which assumptions the transformation helps to achieve.
- These are general suggestions and the actual success always also depends on the data

Transformation	Source	Formula	Support	N	H	L
Box-Cox (shift)	Box and Cox (1964)	$\begin{cases} \frac{(y+s)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0; \\ \log(y+s) & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗	✗	✗
Log-shift opt	Feng, Hannig, and Marron (2016)	$\log(y + \lambda)$	$y \in \mathbb{R}$	✗	✗	✗
Bickel-Docksum	Bickel and Doksum (1981)	$\frac{\lvert y\rvert^\lambda \text{Sign}(y) - 1}{\lambda}$ if $\lambda > 0$	$y \in \mathbb{R}$	✗	✗	
Yeo-Johnson	Yeo and Johnson (2000)	$\begin{cases} \frac{(y+1)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, y \geq 0; \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0; \\ \frac{(1-y)^{2-\lambda} - 1}{\lambda - 2} & \text{if } \lambda \neq 2, y < 0; \\ -\log(1-y) & \text{if } \lambda = 2, y < 0. \end{cases}$	$y \in \mathbb{R}$	✗	✗	
Square Root (shift)	Medina *et al.* (2018)	$\sqrt{y + \lambda}$	$y \in \mathbb{R}$	✗	✗	
Manly	Manly (1976)	$\begin{cases} \frac{e^{\lambda y} - 1}{\lambda} & \text{if } \lambda \neq 0; \\ y & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗	✗	
Modulus	John and Draper (1980)	$\begin{cases} \text{Sign}(y) \frac{(\lvert y\rvert + 1)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0; \\ \text{Sign}(y) \log(\lvert y\rvert + 1) & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗		
Dual	Yang (2006)	$\begin{cases} \frac{(y^\lambda - y^{-\lambda})}{2\lambda} & \text{if } \lambda > 0; \\ \log(y) & \text{if } \lambda = 0. \end{cases}$	$y > 0$	✗		
Gpower	Kelmansky, Martínez, and Leiva (2013)	$\begin{cases} \frac{(y + \sqrt{y^2+1})^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0; \\ \log(y + \sqrt{y^2+1}) & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗		

N – normality, H – homoskedasticity, L-linearity

trafo, cont

Diagnostic checks provided in the package trafo

Assumption	Diagnostic check	Fast check
Normality	Skewness and kurtosis	✗
	Shapiro-Wilk test	✗
	Quantile-quantile plot	
	Histograms	
Homoscedasticity	Breusch-Pagan test	✗
	Residuals vs. fitted plot	
	Scale-location	
Linearity	Scatter plots between y and x	✗
	Observed vs. fitted plot	

Trafo, cont

Core functions of package trafo

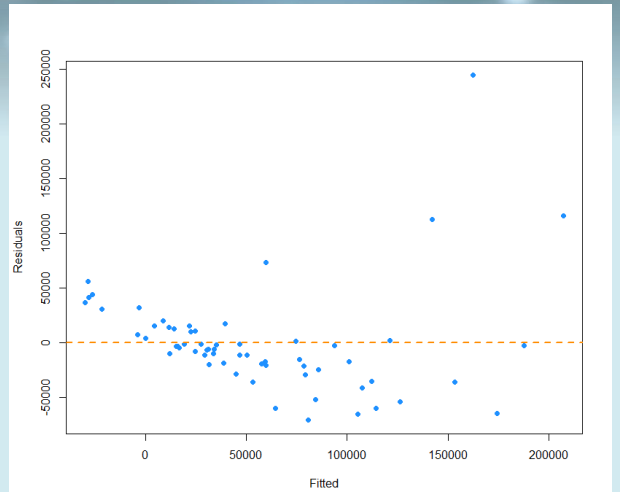
Function	Description
`assumptions()`	Enables a fast check which transformation is suitable.
`trafo_lm()`	Compares the untransformed model with a transformed model.
`trafo_compare()`	Compares two differently transformed models.
`diagnostics()`	Returns information about the transformation and different diagnostics checks in form of tests.
`plot()`	Returns graphical diagnostics checks.

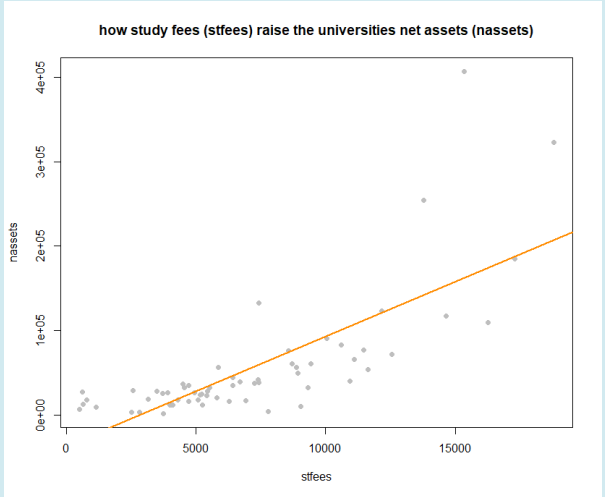
Example – use trafo

```
library(Ecdat)
data(University)
```

- A practical question for the head of a university could be how study fees (stfees) raise the universities net assets (nassets).
- Both variables are metric. Thus, a linear regression could help to explain the relation between these two variables.
- A linear regression model can be conducted in R using the lm function.

```
linMod <- lm(nassets ~ stfees, data = University)
Call:
lm(formula = nassets ~ stfees, data = University)

Residuals:
   Min     1Q Median     3Q    Max
-70529 -20410  -5350  11948 244354

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -36446.66   12078.93   -3.02   0.0037 **
stfees          12.96       1.47    8.80  2.1e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 48500 on 60 degrees of freedom
Multiple R-squared:  0.564,             Adjusted R-squared:  0.556
F-statistic: 77.5 on 1 and 60 DF,  p-value: 2.13e-12
```






```
plot(fitted(linMod), resid(linMod), col = "dodgerblue", pch = 20, cex = 1.5, xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)

plot(nassets ~ stfees, data = University, col = "grey", pch = 20, cex = 1.5, main = "how study fees (stfees) raise the
universities net assets (nassets)")
abline(linMod, col = "darkorange", lwd = 2)
```

Example – use trafo: Finding a suitable transformation

- The reliability of the linear regression model depends on assumptions.
- Amongst others, normality, homoscedasticity, and linearity are assumed. Now we focus on **presenting how you can decide and assess, if and which, transformations help to fulfil these model assumptions**.
- Thus, a first fast check of these model assumptions can be used in the package `trafo` in order to find out if the untransformed model meets these assumptions or if using a transformation seems suitable.
- The fast check can be conducted by the function `assumptions`. This function returns the skewness, the kurtosis and the Shapiro-Wilk test for normality, the Breusch-Pagan test for homoscedasticity and scatter plots between the dependent and the explanatory variables for checking the linear relation.
- All possible arguments of the function assumptions are summarized in the next table. In the following, we only show the returned normality and homoscedasticity tests. The results are ordered by the highest p value of the Shapiro-Wilk and Breusch-Pagan test.

Example – use trafo: Finding a suitable transformation

```
assumptions(linMod)
Test normality assumption
              Skewness Kurtosis Shapiro_W Shapiro_p
logshiftopt     -0.420     4.06     0.974    0.2132
boxcox          -0.489     4.22     0.962    0.0527
bickeldoksum    -0.489     4.22     0.962    0.0527
gpower          -0.489     4.22     0.962    0.0527
modulus         -0.489     4.22     0.962    0.0527
yeojohnson      -0.489     4.22     0.962    0.0527
dual            -0.484     4.22     0.962    0.0519
sqrtshift        0.645     5.28     0.950    0.0139
log             -1.165     5.12     0.914    0.0004
neglog          -1.165     5.12     0.914    0.0004
glog            -1.165     5.12     0.914    0.0004
untransformed    2.450    12.71     0.792    0.0000
reciprocal      -3.726    19.05     0.568    0.0000

Test homoscedasticity assumption
             BreuschPagan_V BreuschPagan_p
modulus               0.103          0.7477
yeojohnson            0.103          0.7477
boxcox                0.103          0.7476
bickeldoksum          0.104          0.7476
gpower                0.103          0.7476
dual                  0.113          0.7369
logshiftopt           0.115          0.7341
neglog                0.716          0.3976
log                   0.716          0.3975
glog                  0.716          0.3975
reciprocal            1.611          0.2044
sqrtshift             5.462          0.0194
untransformed         9.824          0.0017
```

- Following the Shapiro-Wilk test, the best transformation to fulfil the normality assumption is the log-shift opt transformation followed by the Box-Cox, Bickel-Doksum, gpower, modulus and Yeo-Johnson transformation.
- The similarity or even equality of the test results for different transformations is due to the same functional form in the case of positive values as e.g., the Box-Cox and Bickel-Doksum transformation, or to the rounding at four decimals.
- For improving the homoscedasticity assumption, all transformations help except the square root (shift) transformation.
- We choose the Box-Cox transformation for the further illustrations even though some other transformations would be suitable as well.

Example – use trafo: Comparing the untransformed model with a transformed model

- For a more detailed comparison of the transformed model with the untransformed model, a function called trafo_lm (for the arguments see help) :

```
linMod_trafo <- trafo_lm(linMod)
```

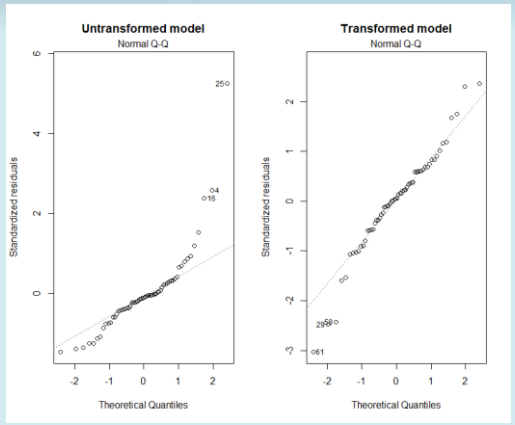
- *The Box-Cox transformation is the default option* such that only the lm object needs to be given to the function.
- The object linMod_trafo is of class trafo_lm and the user can conduct the methods print, summary and plot in the same way as for an object of class lm.

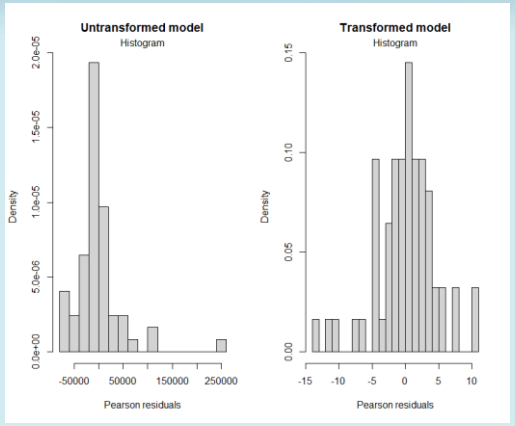
```
diagnostics(linMod_trafo)
Diagnostics: Untransformed vs transformed model

Transformation:  boxcox
Estimation method:  ml
Optimal Parameter:  0.189

Residual diagnostics:

Normality:
Pearson residuals:
                    Skewness Kurtosis Shapiro_W Shapiro_p
Untransformed model    2.450    12.71     0.792  6.02e-08
Transformed model     -0.489     4.22     0.962  5.27e-02

Heteroscedasticity:
                    BreuschPagan_V BreuschPagan_p
Untransformed model          9.824        0.00172
Transformed model            0.104        0.74763
```

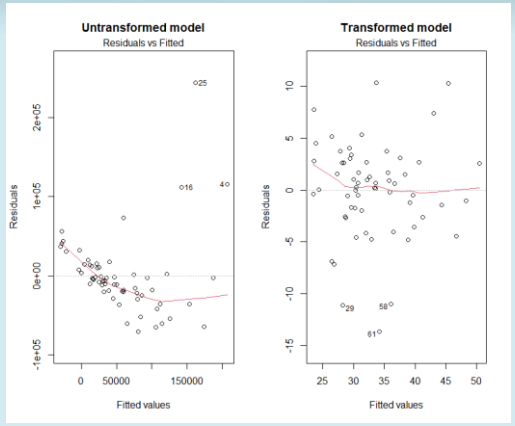
- The rest part of the return shows information of the applied transformation.
- As chosen, the Box-Cox transformation is used with the optimal transformation parameter around **0.19** which is estimated using the maximum likelihood approach that is also set as default.
- *Thus, the optimal transformation parameter differs from 0, which would be equal to the logarithmic transformation, and 1, which means that no transformation is optimal.*
- The Shapiro-Wilk test rejects normality of the residuals of the untransformed model but it does not reject normality for the residuals of the transformed model on a 5% level of significance (**5.27e-02**).
- Furthermore, the skewness (**-0.489**) shows that the residuals in the transformed model are more symmetric and the kurtosis (**4.22**) is closer to 3, the value of the kurtosis of the normal distribution.
- The results of the Breusch-Pagan test clearly show that homoscedasticity is rejected in the untransformed model (p=**0.00172**) but not in the transformed model (p= **0.74763**.

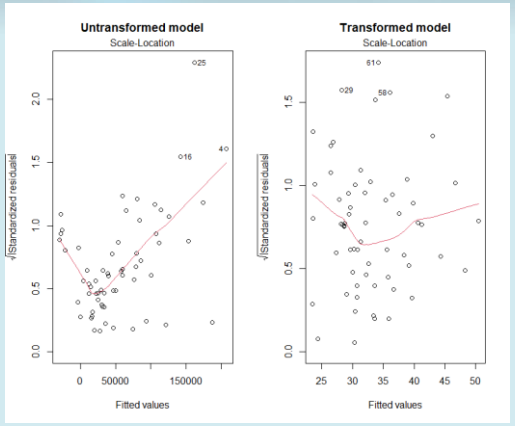
Example – use trafo: Comparing the untransformed model with a transformed model, cont

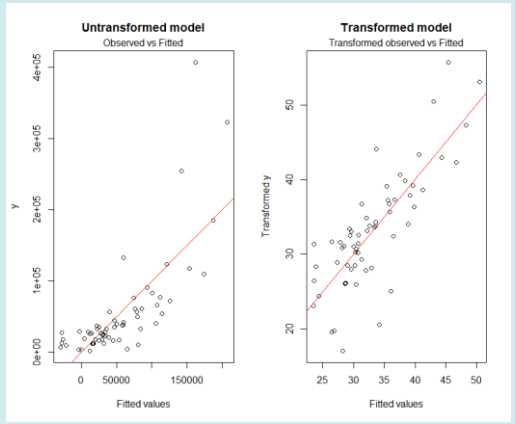
```
plot (linMod_trafo)
```

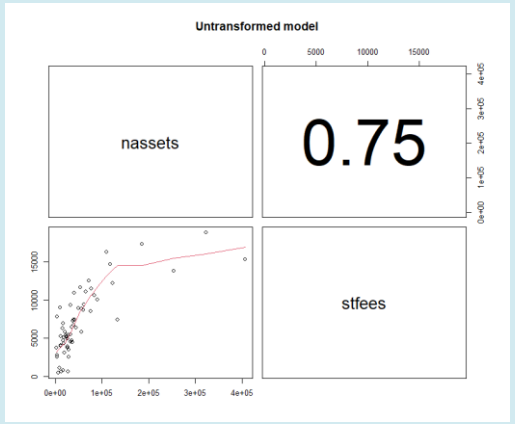


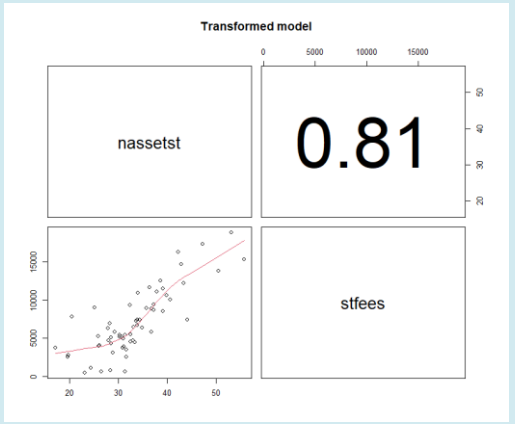












a

b

Selection of obtained diagnostic plots by using plot(linMod_trafo).

(a) shows the scatter plot of the untransformed net assets and the study fees

(b) (b) shows scatter plot of the transformed net assets and the study fees. The numbers specify the correlation coefficient between the dependent and independent variable.

Example – use trafo: more options

- While we only show the example with the default transformation, the user can also easily change the transformation and the estimation method.
- For instance, the user could choose the log-shift opt transformation with the skewness minimization as estimation method.

```
linMod_trafo2 <- trafo_lm(object = linMod,
trafo = "logshiftopt", method = "skew")
```

Binary regression

Binary Regression

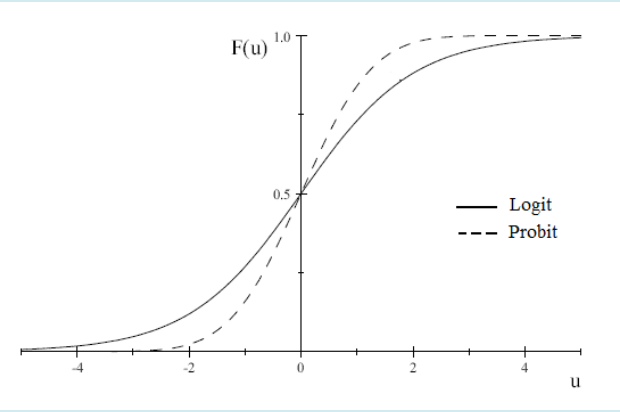
- **A binary response Y can take only two values, 0 or 1, which code two possible outcomes,** for example, that a company goes into default on its loans or that it does not default.
- **A binary regression models the conditional probability that a binary response is 1, given the values of the predictors $X_{i,1}, \ldots, X_{i,p}$.**
- Since a probability is constrained to lie between 0 and 1, a linear model is not appropriate for a binary response.
- However, linear models are so convenient that one would like a model that has many of the features of a linear model.
- This has motivated the development of generalized linear models, often called **GLMs**.

The main types of the model

- **Logit models are used for discrete outcome modeling. This can be for binary outcomes (0 and 1) or for three or more outcomes (multinomial logit).** The logit model operates under the logit distribution (i.e., Gumbel distribution) and is preferred for large sample sizes. Logit has easier interpretation than probit.

- **Probit** is used **when the dependent variable is binary (true/false one/zero).** It is similar to Logit. However, for three or more outcomes (in this context, it's typically ranking or ordering) it operates much differently. (Log-normal distribution*).
- **Probit models** can be generalized to account for non-constant error variances in more advanced econometric settings (known as heteroskedastic probit models) and hence are used in some contexts by economists and political scientists.

- **Tobit** is used when **the dependent variable is continuous but bounded / cut off at one end**. A typical example is wage information where there is a minimum wage - the wage data is bounded at the minimum.
- **Tobit** relies on the underlying y being normally distributed, and your standard errors being homescedastic



Advanced:

- The logit model uses something called the cumulative distribution function of the logistic distribution. The probit model uses something called the cumulative distribution function of the standard normal distribution to define f(∗)
- The link function:
  - In Logit: $Pr(Y=1|X)=[1+\exp(-X'\beta)]^{-1}$
  - In Probit: $Pr(Y=1|X)=\Phi(X'\beta)$ (Cumulative normal pdf)

Simple binary response model (logistic model)

- **The logistic regression model for a binary response variable Y and predictor variable X is**:

$$logit\,[Y = 1|X] = logit\,(P) = \log\frac{P}{1-P} = a + \beta X,$$ (1) where the logit function is defined as

- where *P* = Prob{Y = 1|X}. Thus the model is a linear regression model in the log odds that *Y* = 1 since logit(*P*) is a weighted sum of the *X*s. If all effects are additive (i.e., no interactions are present), the model assumes that for every predictor *Xj*, logit{Y = 1|X} = β0 + β1X1 + . . . + βjXj + . . . + βkXk .
- The parameter βj is then the change in the log odds per unit change in Xj if Xj represents a single factor that is linear and does not interact with other factors and if all other factors are held constant. Instead of writing this relationship in terms of log odds, it could just as easily be written in terms of the odds that *Y* = 1: odds{Y = 1|X} = exp(Xβ),

- **Since, β is a difference in the logarithm of the odds of Y = 1 for one-unit differences in X, e^{β} is the odds ratio of Y = 1**

for a one-unit difference in X.

- **When the odds ratio is greater than 1, it means a positive relationship. Respectively, when an odds ratio is less than**

1 it implies a negative relationship.

Example: Who gets a credit card?

- In this example, we will analyze the data in the CreditCard data set in R's AER package. The following variables are included in the data set:

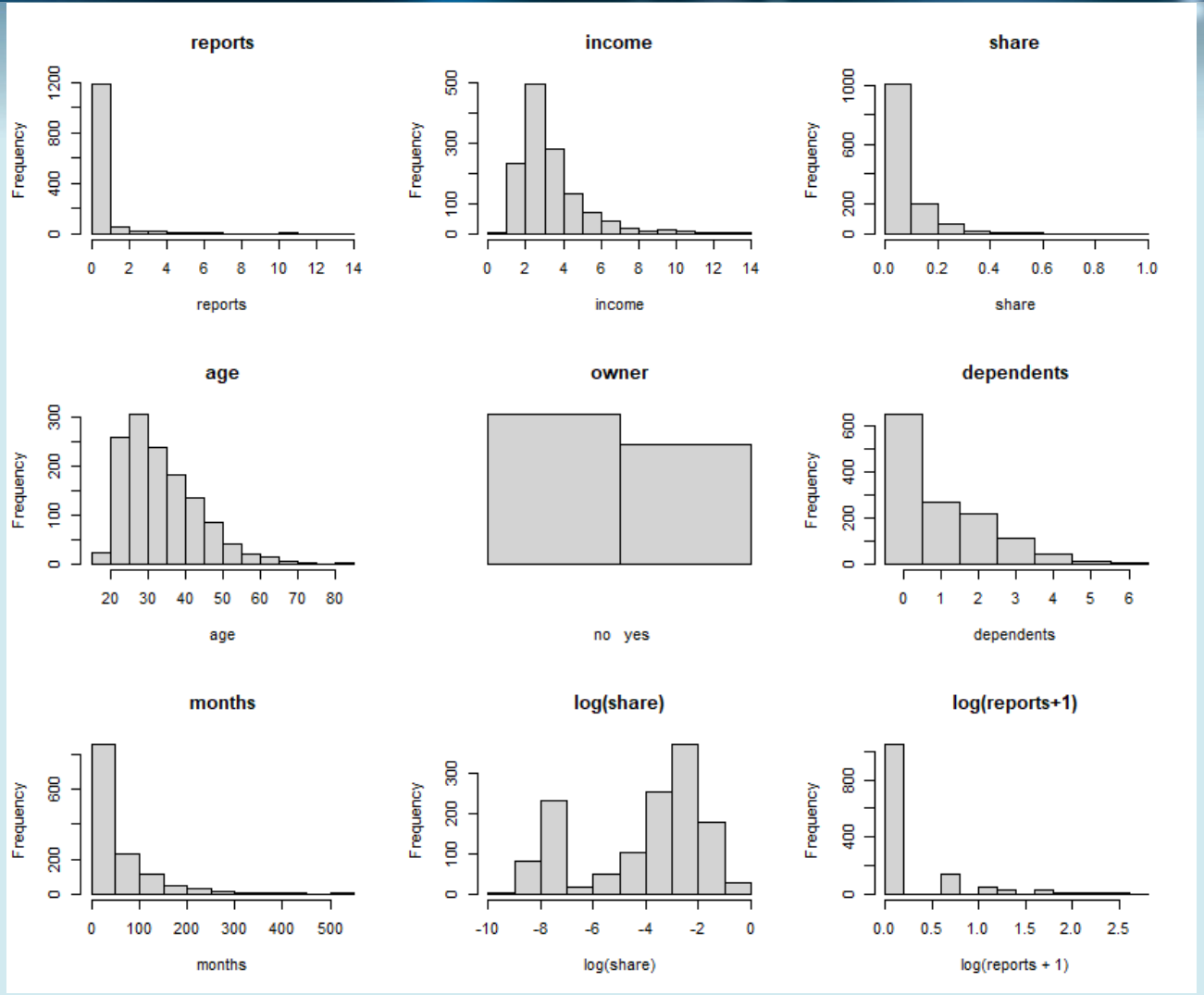
1. card = Was the application for a credit card accepted? (Y)
2. reports = Number of major derogatory reports
3. income = Yearly income (in USD 10,000)
4. age = Age in years plus 12ths of a year
5. owner = Does the individual own his or her home?
6. dependents = Number of dependents
7. months = Months living at current address
8. share = Ratio of monthly credit card expenditure to yearly income
9. selfemp = Is the individual self-employed?
10. majorcards = Number of major credit cards held
11. active = Number of active credit accounts
12. expenditure = Average monthly credit card expenditure

The first variable, *card*, is binary and will be the response.
Variables 2–8 will be used as predictors.

The goal of the analysis is to discover which of the predictors influences the probability that an application is accepted. R's documentation mentions that there are some values of the variable age under one year.
These cases must be in error and they were deleted from the analysis

Example: Who gets a credit card?, cont.

- Figure contains histograms of the predictors. **The variable share is highly right-skewed, so log(share) will be used in the analysis.**
- The variable "reports" is also extremely right-skewed; most values of reports are 0 or 1 but the maximum value is 14. **To reduce the skewness, log(reports+1) will be used instead of reports. The "1" is added to avoid taking the logarithm of 0.**
- There are no assumptions in regression about the distributions of the predictors, so skewed predictor variables can, in principle, be used.
- However, **highly skewed predictors have high-leverage points and are less likely to be linearly related to the response**.
- It is a good idea at least to consider transformation of highly skewed predictors. In fact, the logistic model was also fit with "reports" and "share" untransformed, but this increased AIC by more than 3 compared to using the transformed predictors (you can replicate it at home)



Example: Who gets a credit card?, cont.

- First, **a logistic regression model is fit with all seven predictors using the `glm()` function.**
  - Generalized linear model is defined in terms of linear predictor : $\eta = X\beta$
  - That is passed through the link function g: $g(E(Y|X)) = \eta$
  - It models the relation between the dependent variable Y and independent variables $X = X_1, X_2, \ldots, X_k$. More precisely, it models a conditional expectation of Y given X, $E(Y|X) = \mu = g^{-1}(\eta)$
- The R code is:

```
fit1= glm(card~log(reports+1)+income+log(share)+age+owner+dependents+months,
family="binomial",data=CreditCard_clean)
```

- `family="binomial"` – **If your outcome is binary (zeros and ones), proportions of "successes" and "failures" (values between 0 and 1), or their counts, you can use Binomial distribution**, i.e. the logistic regression model.
  - If there is more then two categories, you would use multinomial distribution in multinomial regression.
- if your outcome is continuous and unbounded, then the most "default" choice is the *Gaussian distribution (a.k.a. normal distribution),* i.e. the standard linear regression (unless you use other link function then the default identity link).
- If you are dealing with continuous non-negative outcome, then you could consider the *Gamma distribution*, or *Inverse Gaussian distribution*.
- If your outcome is discrete, or more precisely, you are dealing with counts (how many times something happen in given time interval), then the most common choice of the distribution to start with is *Poisson distribution*.
  - The problem with Poisson distribution is that it is rather inflexible in the fact that it assumes that mean is equal to variance, if this assumption is not met, you may consider using quasi-Poisson family, or negative binomial distribution (see also Definition of dispersion parameter for quasipoisson family).

Example: Who gets a credit card?, cont.

```
summary(fit1)
Call:
glm(formula = card ~ log(reports + 1) + income + log(share) +
    age + owner + dependents + months, family = "binomial", data = CreditCard_clean)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.63787   0.00004   0.00044   0.00225   2.87105

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      21.473930   3.674325   5.844 5.09e-09 ***
log(reports + 1) -2.908644   1.097604  -2.650  0.00805 **
income            0.903315   0.189754   4.760 1.93e-06 ***
log(share)        3.422980   0.530499   6.452 1.10e-10 ***
age               0.022682   0.021895   1.036  0.30024
owneryes          0.705171   0.533070   1.323  0.18589
dependents       -0.664933   0.267404  -2.487  0.01290 *
months           -0.005723   0.003988  -1.435  0.15130
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1398.53  on 1311  degrees of freedom
Residual deviance:  139.79  on 1304  degrees of freedom
AIC: 155.79

Number of Fisher Scoring iterations: 11
```

Several of the regressors have large p-values, so stepAIC() was used to find a more parsimonious model.

Example: Who gets a credit card?, cont.

```
stepAIC(fit1) # Choose a model by AIC in a Stepwise Algorithm

Step:  AIC=154.22
card ~ log(reports + 1) + income + log(share) + dependents

                   Df Deviance     AIC
<none>                  144.22  154.22
- dependents        1   150.28  158.28
- log(reports + 1)  1   164.18  172.18
- income            1   173.62  181.62
- log(share)        1  1079.61 1087.61

Call:  glm(formula = card ~ log(reports + 1) + income + log(share) +
    dependents, family = "binomial", data = CreditCard_clean)

Coefficients:
     (Intercept)  log(reports + 1)            income        log(share)        dependents
         21.3224           -2.8953            0.8717            3.3102           -0.5506

Degrees of Freedom: 1311 Total (i.e. Null);  1307 Residual
Null Deviance:         1399
Residual Deviance: 144.2    AIC: 154.2
```

Example: Who gets a credit card?, cont.

- Below is the fit using the model selected by stepAIC().
- For convenience later, each of the regressors was mean-centered; "_c" appended to a variable name indicates centering.

```
glm_fit02 = glm(card~log_reports_c+income_c+log_share_c+dependents_c,
family="binomial",data=CreditCard_clean)
Call:
glm(formula = card ~ log_reports_c + income_c + log_share_c + dependents_c, family = "binomial",
data = CreditCard_clean)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.50664   0.00006   0.00061   0.00280   2.79345

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)     9.5238     1.7213   5.533 3.15e-08 ***
log_reports_c  -2.8953     1.0866  -2.664  0.00771 **
income_c        0.8717     0.1724   5.056 4.28e-07 ***
log_share_c     3.3102     0.4942   6.698 2.11e-11 ***
dependents_c   -0.5506     0.2505  -2.198  0.02793 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1398.53  on 1311  degrees of freedom
Residual deviance:  144.22  on 1307  degrees of freedom
AIC: 154.22

Number of Fisher Scoring iterations: 11
```

- **It is important to understand what the logistic regression model is telling us about the probability of an application being accepted.**

- Qualitatively, we see that the probability of having an application accepted increases with income and share and decreases with reports and dependents.
- To understand these effects look at the formulas:

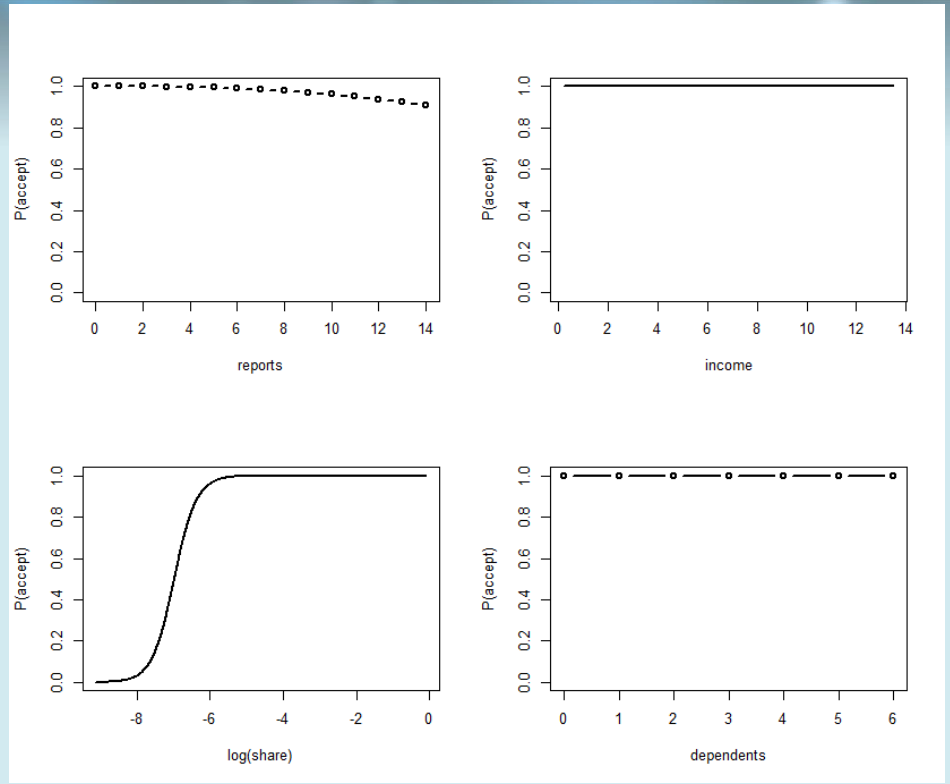
$$\log\left(\frac{p(x)}{1-p(x)}\right) = 9.52 - 2.895\text{logreports}_c + 0.8717\text{income}_c + 3.31\text{logshare}_c - 0.55\text{dependents\_c}$$

Thus, your response is:

$$\hat{p}(x) = \frac{e^{\hat{\eta}(x)}}{1+e^{\hat{\eta}(x)}} = \frac{e^{9.52-2.895\text{logreports}_c+0.8717\text{income}_c+3.31\text{logshare}_c-0.55\text{dependents\_c}}}{1+e^{(9.52-2.895\text{logreports}_c+0.8717\text{income}_c+3.31\text{logshare}_c-0.55\text{dependents\_c})}}$$

Example: Who gets a credit card?, cont.

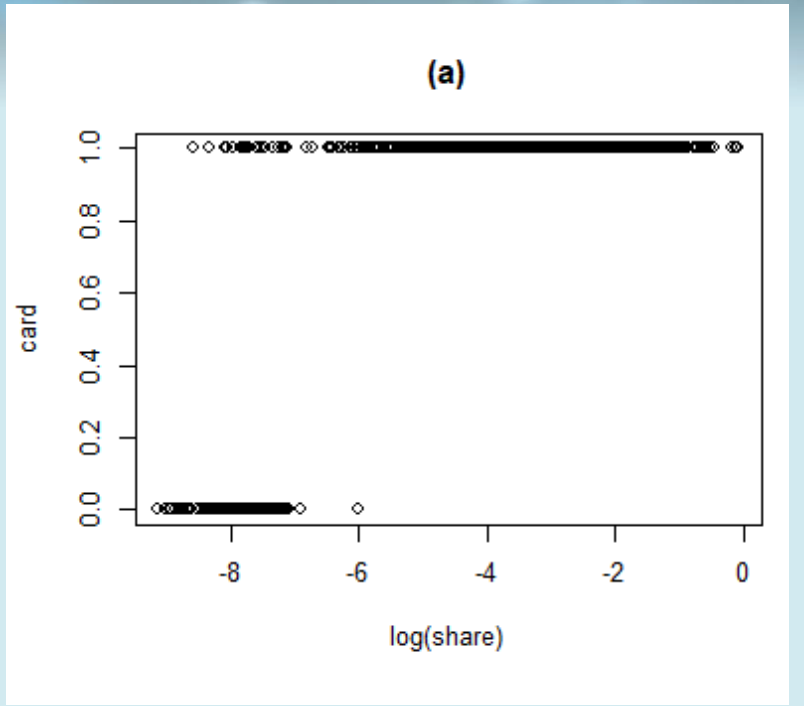
- Figure plots the probability that a credit card application is accepted as functions of reports, income, log(share), and dependents.
- **In each plot, the other variables are fixed at their means.**
- The variable with the largest effect is share, the ratio of monthly credit card expenditure to yearly income.
- We see that applicants who spend little of their income through credit cards are unlikely to have their applications accepted.



Plots of probabilities of a credit card application being accepted as functions of single predictors with other predictors fixed at their means. The variables vary over their ranges in the data

Example: Who gets a credit card?, cont.

- In Fig. is a plot of card, which takes value 0 if an application is rejected and 1 if it is accepted, versus log(share).
- It should be emphasized that panel (**a) is a plot of the data, not a fit from the model.**
- **We see that an application is always accepted if log(share) exceeds –6, which translates into share exceeding 0.0025 (**the ratio of monthly credit card expenditure to yearly income**).**
- Thus, **in this data set, among the group of applicants whose average monthly credit card expenses exceeded 0.25% of yearly income, all credit card applications were accepted.**



Plots of probabilities of a credit card application being accepted as functions of single predictors with other predictors fixed at their means. The variables vary over their ranges in the data

OTHER (USEFUL) REGRESSIONS AND THEIR APPLICATION

Example: herding

- In the aftermath of several crises, “herd” has again become a popular term.
- Herding is met in very different settings from neurology and zoology, to sociology, psychology, economics, and finance.
- Generally speaking, in economics and finance with the term **herding or herd behavior** we mean ***the process where economic agents are imitating each other actions and/or base their decisions upon the actions of others***.
- “Investors and fund managers are portrayed as herds that charge into risky ventures without adequate information and appreciation of the risk-reward trade-offs and, at the first sign of trouble, flee to safer havens.”
- **Herding by market participants exacerbates volatility, destabilizes markets**.

What is herding?

- For an investor to imitate others, she must be aware of and be influenced by others' actions.
- Intuitively, **an individual can be said to herd if she would have made an investment without knowing other investors' decisions, but does not make that investment when she finds that others have decided not to do so**.
- Alternatively, **she herds when knowledge that others are investing changes her decision from not investing to making the investmen***t*.

The main reasons to herd

- Others may know something about the return on the investment and their actions reveal this information.
- market participants may infer information from the actions of previous participants
- investors may react to the arrival of fundamental information
- analysts may herd in order to protect reputation
- institutional investors may herd for reasons related to remuneration
- investors may simply be irrational and herd behavior can arise as the consequence of psychological and/or social conventions.

Why is it not good?

- When investors are influenced by others' decisions, they may herd on an investment decision that is wrong for all of them.
- Suppose that **100 investors each have their own assessments, possibly different, about the profitability of investing in an emerging market**.
- For concreteness, **suppose that 20 of the investors believe that this investment is worthwhile** and the remaining **80 believe that it is not**.
- **Every investor knows only her own estimate of the profitability of this investment**; she does not know the assessments of others' or which way a majority of them are leaning.
- **If these investors pooled their knowledge and assessments, they would collectively decide that investing in the emerging market is not a good idea.**
- But they **do not share their information and assessments with each other**.
- Moreover, these 100 investors **do not take their investment decisions at the same time**.
- Suppose that **the first few investors** who decide **are among the 20 optimistic investors** and they make a decision **to enter the emerging market**.
- Then several of **the 80 pessimistic investors may revise their beliefs and also decide to invest**.
- This, in turn, could have **a snowballing effect**, and lead to most of the 100 individuals investing in the emerging market.
- Later, when the unprofitability of the decision becomes clear, these investors exit the market.

Theories of herding

- Some authors argue that under certain circumstances **herding is a rational choice**. For instance,
  - money managers may *mimic the actions of other* money managers
  - in order to *preserve reputation* and/or compensation
  - *younger analysts know that if they make bold forecasts and deviate from the consensus they are more likely to be fired*
  - during a bank crisis depositors contribute to runs on banks because they see long lines of other depositors outside banks and know that if they do not join the line early there may be no funds left for them

Theories of herding, cont

- Furthermore, many economists present formal models on how investor sentiment may affect investor trading patterns and behavior and lead to systematic asset misspricings.
- For instance, Barberis, Shleifer and Vishny (1998) suggest **a model of investor sentiment that predicts investor overreaction and/or underreaction to information**.
- Their results are consistent with **empirical evidence on the shortcomings of personal judgment under uncertainty.**
- Hong and Stein (1999) propose a theory with **two type of boundedly rational market participants**.
  - In this model, *short-run price underreaction is due to slow information dissemination that is exploited by momentum traders which, in turn, leads to long-term overreaction*.
- Daniel, Hirshleifer, and Subrahmanyam (1998) suggest that **investors are overconfident regarding their private information and suffer from biased selfattribution**.
  - These biases lead to autocorrelations, excess volatility and return predictability.

Theories of herding: Information cascades

- The above example shows several aspects of information cascades or herd behavior arising from informational differences.
- First, **the actions** (and the assessments) of investors who decide early **may be crucial in determining which way the majority will decide**.
- Second, **the decision** that investors herd on **may well be incorrect**.
- Third, **if** investors take **a wrong decision**, **then with experience and/or the arrival of new information, they are likely to eventually reverse their decision starting a herd in the opposite direction**.
- **This, in turn, increases volatility in the market**.

Theories of herding: Information cascades, cont.

- **An informational cascade takes place when it is optimal for individuals to follow the observable actions of individuals before them, disregarding their own information** (Bikhchandani, Hirshleifer, and Welch, 1992).
- For example, for investors that enter the market at a later stage it may be an optimal decision to ignore their own private information and mimic the trading behavior of previous investors since they may infer that previous investors possess private information.
- Informational cascades may have an influence over perfectly rational individuals and lead to the creation of bubbles.
- A decision model where it is rational for decision makers to look at the decisions made by previous decision makers since previous decision makers may possess related information that is important is analyzed by Banerjee (1992).

Theories of herding: Information cascades, cont.

- A general sequential choice model where a decision maker will act only on the information obtained from previous decisions ignoring private information (as will later decision makers) is discussed by Bikhchandani, Hirshleifer, and Welch (1992)
- They argue that, **irrespective of the social desirability of the outcome, the reasoning may be entirely rational** (see also, Welch, 1992).
- Note that informational cascades may be linked with partial or complete information aggregation blockages, increased fragility to even small informational shocks, fads and stampedes (Hirshleifer and Teoh, 2003; among others).

Theories of herding: Information cascades, cont.

- Avery and Zemsky (1998) find that herding in the form of an informational cascade is not possible, if simple information structures and a price mechanism are assumed**.**
- **In case of complicated information structures, however, herding is possible and it may affect asset prices only when the market is uncertain for both the asset value and the information of the average trader.**
- In a laboratory experiment, Cipriani and Guarino (2005) study herding in financial markets and their results are in line with the results of Avery and Zemsky, i.e. **when the subjects are trading for informational reasons in a frictionless market herding occurs rarely**, although in some cases they observe that subjects follow a contrarian strategy or choose to ignore private information.

Theories of herding: Spurious herding

- Herding results from an obvious intent by investors to copy the behavior of other investors.
- This should be distinguished from "**spurious herding**" where groups facing similar decision problems and information sets take similar decisions.
  - Thus, "spurious" herding where investors face a similar fundamental-driven information set and thus make similar decisions
- "**intentional**" herding where investors have an intention to copy the behavior of others.
- The "**spurious herding**" may lead to an efficient outcome while the "**intentional**" may not
- Empirically distinguishing "spurious herding" from "intentional" herding is easier said than done and may even be impossible, since typically, a multitude of factors have the potential to affect an investment decision.
- Fundamentals-driven spurious herding out of equities could arise if, for example, interest rates suddenly rise and stocks become less attractive investments.
  - Investors under the changed circumstances may want to hold a smaller percentage of stocks in their portfolio. This is not herding according to the definition above because investors are not reversing their decision after observing others. Instead, they are reacting to commonly known public information, which is the rise in interest rates.

Theories of herding: a need to share the blame

- Scharfstein and Stein (1990) argue that reputation concerns in labor markets with no perfect information and **a need to share the blame** when things go bad may lead managers to follow each other's actions.
- They present a learning model where the labor market is able to update its understanding of the manager's ability from the investment decisions a manager is making.
- **Thus, manager concern for labor market reputation may lead to rational (but socially inefficient) herd behavior, i.e. managers may ignore significant private information and follow the decisions of other managers.**
- In other words, **herding may be viewed as insurance against manager underperformance** (Rajan, 2006).

Theories of herding: the perception of analyst abilities affects analyst compensation

- In Trueman (1994) **the perception of analyst abilities affects analyst compensation**.
- **Trueman's theoretical model indicates** that the earnings forecasts of analysts do not necessarily reflect in an unbiased manner their private information, but rather **there is a tendency to release forecasts closer to prior earnings expectations.**
- Furthermore, **analysts also tend to forecast earnings similar to those previously announced by other analysts in an effort to copy higher ability and obtain higher compensation, even when private analyst information does not justify this behavior**.

Theories of herding: high reputation or low ability

- Graham (1999) develops a model where analysts are more likely to herd:
  - when they **are characterised by high reputation or low ability** (e.g. **high reputation analysts have greater incentives to hide in the consensus in order to protect their reputation**),
  - **when there is strong public information inconsistent with analyst private information**, and
  - when private information signals across analysts exhibit positive correlation.

Theories of herding: short horizons

- Froot, Scharfstein, and Stein (1992) show that **if speculators have short horizons they may herd on the same information trying to learn what other informed investors know**.
- Their model allows for some speculators to have short trading horizons, which implies that they may allocate research resources in a non-optimal way.
  - *This may violate informational efficiency* (although at the pricing stage the market may be efficient) *in the sense that investors may have the tendency to concentrate on one information source* (that may be of poor quality or have no relation to fundamentals) *rather than employ a diverse set of information sources.*
  - As a growing number of speculators acquire this information it will be disseminated in the market and thus it is profitable to acquire this information at an early stage (positive informational spillovers); in their model there may be multiple herding equilibria.

Theories of herding: irrationality

- **Other authors suggest that investors (or a subset of) are irrational and that the existence of such irrational investors may give rise to bubble-like phenomena and herd behavior.**
- Furthermore, non-rational herd behavior can arise as the consequence of psychological stimuli and restraints, such as pressure from social circles and/or social conventions.
  - For instance, Keynes (1936) argues that investors are affected by sociological factors (e.g. social conventions) that may drive market participants to imitate the actions of others during periods of uncertainty.
- Baddeley, Curtis and Wood (2004) demonstrate that even experts may resort to herd behavior, given information scarcity, asymmetry and the employment of common heuristic rules.

Theories of herding: arbitragers

- Shleifer and Summers (1990), **distinguish between arbitrageurs who are fully rational and noise traders** (Black, 1986), i.e. **irrational investors who act on noise and whose trading behavior suffers from systematic biases**.
- They suggest that **some shifts in investor demand for assets and changes in investor sentiment appear to be irrational and not justified by fundamentals**, e.g. investors' reaction to pseudo-signals such as advice by "financial gurus".
- Consider the case where a fraction of investors follows trends.
  - Arbitrageurs, instead of opposing this bandwagon, rationally decide to jump on it.
  - The new higher demand will lead prices even higher and further away from fundamentals, attract more irrational investors, and the rational arbitragers will exit when prices are near the top in order to collect their profits.
  - *In other words, the behavior of rational arbitragers, in the short run, will nourish the irrational price bubble*.

Measuring herding in financial markets

- How is herd behavior measured in empirical studies?
- Generally speaking, we can classify empirical methodologies into two main categories:
  - studies that rely on micro-data or proprietary data and investigate whether specific investor types herd,
  - and studies that rely on aggregate price and market activity data and investigate herding towards the market consensus.

Measuring herding in financial markets, cont.

- We estimate herding behavior by means of the Chang et al. (2000) approach.
- Despite the availability of alternative models (Bohl et al. 2013; Lee, 2017; Clements et al., 2017), we choose this approach given its wide use in literature to be able to ensure comparability of our results with those of prior studies.
- Chang et al. (2000) argue that **if investors tend to follow aggregate market behavior during periods of large average price movements, then the linear and increasing relation between dispersion and market return will no longer hold and it can become non-linearly increasing or even decreasing**.
- Thus, they utilize a non-linear regression specification to estimate the relation **between the cross-sectional absolute deviation of returns and the market return**.
- **If investors trust market expectations and follow them, investors' return will not deviate from market return, whereas dispersion level or variance between individuals' return and market return, in light of adopting herd behavior by investors, will be zero**.
- **When stock return differs from market return, dispersion increases, and in case investors follow market's expectations, dispersion will become significantly less than the mean.**

Measuring herding in financial markets, cont.

- They propose a test of herding behavior that also requires a parameter to capture any possible non-linearity in the relation between asset return dispersions and the market return.
- We use the **cross-sectional absolute deviation** at time t (CSADt)
- **CSAD is estimated as the average AVD (Absolute Value of Deviation) of each stock relative to the return of the equally-weighted market portfolio**.
- **The notion behind this approach is that if herding is present during periods of extreme market conditions then there should be a less than proportional increase (or even decrease) in the CSAD measure.**
- Note here that CSAD is not used as a metric for herding: **herding is identified through the relationship between CSAD and the market return.**

Measuring herding in financial markets, cont.

- The following specification for estimates is used:

$$CSAD_{m,t} = \beta_0 + \beta_1 \left|R_{m,t}\right| + \beta_2 R_{m,t}^2 + e_t$$

where $\left|R_{m,t}\right|$ is the average absolute market return of all actively traded selected equities in a market, at time t, $CSAD_{m,t}$ is the Cross Sectional Absolute Deviation of returns and is calculated as follows:

$$CSAD_{m,t} = \frac{\sum_{i=1}^{n} \left|R_{i,t} - R_{m,t}\right|}{n}$$

where $R_{i,t}$ is the first logarithmic difference of closing prices for an equity *i* at time *t*

$$R_{i,t} = InP_t - InP_{t-1}$$

Measuring herding in financial markets, cont.

- **If herding is not present in a market, the relationship between the cross-sectional return dispersion, $CSAD_{m,t}$, and absolute market returns, $|R_{m,t}|$, would be expected to be positive and linear, implying that β_1 would be expected to be significantly positive, while β_2 insignificant.**
- On the contrary, **in the presence of herding, when values of $|R_{m,t}|$ are high and thus substantial market movements are observed, the relationship between $CSAD_{m,t}$ and $|R_{m,t}|$ would be non-linear, implying that β_2 would be negative and significant.**
- Thus, **herding lowers cross-sectional dispersion of returns compared to the case of rational pricing.**
- Bernales et al. (2019) postulate that herding can be considered to be stronger when β_1 is negative, implying a negative relationship between the cross-sectional deviation of the cryptocurrency's return and the magnitude of respective market returns.

Measuring herding in financial markets, cont.

- To assess herding on up/down market days $CSAD_{m,t} = \beta_0 + \beta_1 \left|R_{m,t}\right| + \beta_2 R_{m,t}^2 + e_t$ is extended:

$$\boldsymbol{CSAD_{m,t} = \beta_0 + \beta_1 D^{up} \left|R_{m,t}\right| + \beta_2 (1 - D^{up}) \left|R_{m,t}\right| + \beta_3 D^{up} R_{m,t}^2 + \beta_4 (1 - D^{up}) R_{m,t}^2 + e_t}$$

- where D^{up} is equal to one (zero) on days with positive (negative) values of $R_{m,t}$.
- Significantly negative values of β_3 (β_4) would indicate the presence of herding on days of positive (negative) average market performance.

Parameters estimation

- It is a regression!
  - Thus you can apply any regression method that satisfies the data you have in your hands.
- First, non-linear OLS then estimate the classic Newey-West (Newey & West, 1987) Heteroscedasticity and Autocorrelation consistent (HAC) estimators to linear regressions using Bartlett kernel weights as described in Newey & West (1987, 1994). (because of time-series regressions). Try different methods you can find.

Example in R:

```
require (xts) # the package to work with xts objects
require (zoo) # the package to work with zoo objects
library (PerformanceAnalytics) # the package to calculate return automatically
require (sandwich)
require(lmtest)

## load the data
close.prices = read.csv("close.prices.csv", header = TRUE) # import the datafile
close.prices$Index = as.POSIXct(close.prices$Index,format="%d/%m/%Y %H", tz = "") # change the type of the first column
close.prices.xts <- xts(close.prices[,-1], order.by=close.prices[,1]) # transform the dataframe into xts object
close.prices.zoo  <- as.zoo(close.prices.xts) # transfporm into zoo object
```

Example in R:

```
## calculate the return
return = Return.calculate(
close.prices.xts , method = "log")

# a function to create CSAD
exchange.herd = function(return)
{
  n=ncol(return)
  Rm = rowMeans(return)
  Rm = rowMeans(return)
  temp_dif =abs(return-Rm)
  temp_sum = rowSums(temp_dif)
  CSAD = temp_sum / ncol(return)
  CSAD = cbind (CSAD, Rm)
  return (CSAD)
}
f = exchange.herd(return)
head (f)
```

```
CSAD.df = fortify.zoo(f)
CSAD.df$Rm2 = CSAD.df$Rm^2
CSAD.df = CSAD.df[-c(1),]
head (CSAD.df)
tail (CSAD.df)

y = CSAD.df$CSAD
x1 = abs (CSAD.df$Rm)
x2 =CSAD.df$Rm2
variables.ready = cbind (y, x1, x2)

#Linear model
linearMod <- lm(y~x1+x2)  # build linear regression
model on full data
print(linearMod)
summary(linearMod)

#Newey-West Heteroscedasticity and Autocorrelation
consistent (HAC) estimators
coeftest(linearMod,vcov=NeweyWest(linearMod,verbose=T))
```

Time varying regression

TV-regression

- Time-Varying Regressions, TVR (Bollerslev, Patton and Quaedvlieg, 2016; Casas, Mao, and Veiga, 2018) can be estimated to assess evolution over time.
- A classical linear model can be expressed as $y_t = x_t^T \beta + u_t$, where *t*=1,…, *T*, y_t is a dependent variable, $x_t = (x_{1t}, x_{2t}, \dots, x_{dt})^T$ is a vector of repressors at time *t*, $\beta = (\beta_0, \beta_1, \dots, \beta_d)^T$ is a vector of coefficients and u_t is the error term. If the coefficients are allowed to vary over time.
- The time-varying coefficient model (TV-LM) can be specified as follows:

$$y_t = x_t^T \beta(z_t) + u_t, t = 1, \dots, T$$

- where z_t is the smoothing variable, transforming coefficients to be a function of z_t:

$$\beta(z_t) = (\beta_0(z_t), \beta_1(z_t), \dots, \beta_d(z_t))^T.$$

z_t can be defined in two ways.

- First, as unknown functions of time, $\beta(z_t) = f(\tau)$, as proposed in Robinson (1989), and further developed by Cai (2007) and Chen, Gao, Li, and Silvapulle (2017).
- Second, this variable can be defined as unknown functions of a random variable, $\beta(z_t) = f(z_t)$, developed by Hastie and Tibshirani (1993), Cai, Fan, and Yao (2000); Chang and Martinez-Chombo (2003), Cai, Li, and Park (2009), Sun, Cai, and Li (2013) and Gao and Phillips (2013).

TV-regression, cont.

- The estimation is done by combining OLS and the local polynomial kernel estimator (Fan and Gijbels 1996).
- Given that $\beta(\cdot)$ is twice differentiable, an approximation of $\beta(z_t)$ can be expressed by means of the Taylor rule, $\beta(z_t) = \beta(z) + \beta(z)^{(1)}(z_t - z)$, where $\beta^{(1)(z)} = d\beta(z)/dz$ is the first derivative.
- The following minimization problem should be solved:

$$\arg\min_{\theta_0,\theta_1} \sum_{t=1}^{T} [y_t - x_t^T\theta_0 - (z_t - z)x_t^T\theta_1]^2 K_b(z_t - z)$$

This approach can be fit to a set of weighted local regressions with an optimally chosen window size, defined by the bandwidth *b*. Using the weights derived from the kernel $K_b(z_t - z) = b^{-1}K\left(\frac{z_t - z}{b}\right)$, yields the following local estimator:

$$\begin{pmatrix} \hat{\beta}_t \\ \hat{\beta}_t^{(1)} \end{pmatrix} = \begin{pmatrix} S_{T,0}(z_t) & S_{T,1}^T(z_t) \\ S_{T,1}(z_t) & S_{T,2}(z_t) \end{pmatrix}^{-1} \begin{pmatrix} T_{T,0}(z_t) \\ T_{T,1}(z_t) \end{pmatrix}$$

where,

$$S_{T,s}(z_t) = \frac{1}{T}\sum_{i=1}^{T} X_i' X_i (z_i - z_t)^s K\left(\frac{z_i - z_t}{h}\right)$$

$$T_{T,s}(z_t) = \frac{1}{T}\sum_{i=1}^{T} X_i' (z_i - z_t)^s K\left(\frac{z_i - z_t}{h}\right) y_i$$

TV-regression in R, example

Several markets, e.g., USD, EUR, GBP etc Crytptocurrrency markets (BTC, ETH, LTC).

```
return = Return.calculate( close.prices.zoo , method = "log")
library(pastecs)
descriptive.stat.return = stat.desc(return)
```

	USD_Bitfinex_BTCUSD_Raw	USD_Bitfinex_ETHUSD_Raw	USD_Bitfinex_LTCUSD_Raw	USD_Bitstamp_BTCUSD_Raw	USD_Bitstamp_ETHUSD_Raw	USD_Bitstamp_LTCUSD_Raw	USD_Coinbase_BTCUSD_Raw	USD_Coinbase_ETHUSD_Raw	USD_Coinbase_LTCUSD_Raw	USD_Kraken_BTCUSD_Raw	USD_Kraken_ETHUSD_Raw	USD_Kraken_LTCUSD_Raw
nbr.val	10953	10953	10953	10953	10953	10953	10953	10953	10953	10953	10953	10953
nbr.null	61	81	147	61	81	147	61	81	147	61	81	147
nbr.na	1	1	1	1	1	1	1	1	1	1	1	1
min	-0.17881	-0.23083	-0.20449	-0.17881	-0.23083	-0.20449	-0.17881	-0.23083	-0.20449	-0.17881	-0.23083	-0.20449
max	0.182841	0.162843	0.175404	0.182841	0.162843	0.175404	0.182841	0.162843	0.175404	0.182841	0.162843	0.175404
range	0.361647	0.39367	0.379899	0.361647	0.39367	0.379899	0.361647	0.39367	0.379899	0.361647	0.39367	0.379899
sum	0.495402	-0.03262	0.212198	0.495402	-0.03262	0.212198	0.495402	-0.03262	0.212198	0.495402	-0.03262	0.212198
median	4.26E-05	0	0	4.26E-05	0	0	4.26E-05	0	0	4.26E-05	0	0
mean	4.52E-05	-2.98E-06	1.94E-05	4.52E-05	-2.98E-06	1.94E-05	4.52E-05	-2.98E-06	1.94E-05	4.52E-05	-2.98E-06	1.94E-05
SE.mean	7.95E-05	9.41E-05	0.000105	7.95E-05	9.41E-05	0.000105	7.95E-05	9.41E-05	0.000105	7.95E-05	9.41E-05	0.000105
CI.mean.0.95	0.000156	0.000184	0.000206	0.000156	0.000184	0.000206	0.000156	0.000184	0.000206	0.000156	0.000184	0.000206
var	6.92E-05	9.69E-05	0.000121	6.92E-05	9.69E-05	0.000121	6.92E-05	9.69E-05	0.000121	6.92E-05	9.69E-05	0.000121
std.dev	0.008317	0.009845	0.011016	0.008317	0.009845	0.011016	0.008317	0.009845	0.011016	0.008317	0.009845	0.011016

Example in R:

```
## calculate the return
return = Return.calculate(
close.prices.xts , method = "log")
```

$$CSAD_{m,t} = \frac{\sum_{i=1}^{n} |R_{i,t} - R_{m,t}|}{n}$$

```
# a function to create CSAD
exchange.herd = function(return)
{
  n=ncol(return)
  Rm = rowMeans(return)
  Rm = rowMeans(return)
  temp_dif =abs(return-Rm)
  temp_sum = rowSums(temp_dif)
  CSAD = temp_sum / ncol(return)
  CSAD = cbind (CSAD, Rm)
  return (CSAD)
}
f = exchange.herd(return)
head (f)
```

```
CSAD.df = fortify.zoo(f)
CSAD.df$Rm2 = CSAD.df$Rm^2
CSAD.df = CSAD.df[-c(1),]
head (CSAD.df)
tail (CSAD.df)

y = CSAD.df$CSAD
x1 = abs (CSAD.df$Rm)
x2 =CSAD.df$Rm2

#Linear model
linearMod <- lm(y~x1+x2)  # build linear regression
model on full data
print(linearMod)
summary(linearMod)
```

Simple regression in R, example

$$CSAD_{m,t} = \beta_0 + \beta_1 |R_{m,t}| + \beta_2 R_{m,t}^2 + e_t$$

```
linearMod <- lm(y~x1+x2, data= hourly)
> summary(linearMod)

Call:
lm(formula = y ~ x1 + x2, data = hourly)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0143816 -0.0009332 -0.0002893  0.0005789  0.0232083

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.106e-03  2.415e-05   45.79   <2e-16 ***
x1           2.527e-01  3.577e-03   70.65   <2e-16 ***
x2          -8.912e-01  4.056e-02  -21.97   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.001847 on 10950 degrees of freedom
Multiple R-squared:  0.3929,         Adjusted R-squared:  0.3928
F-statistic:  3543 on 2 and 10950 DF,  p-value: < 2.2e-16
```

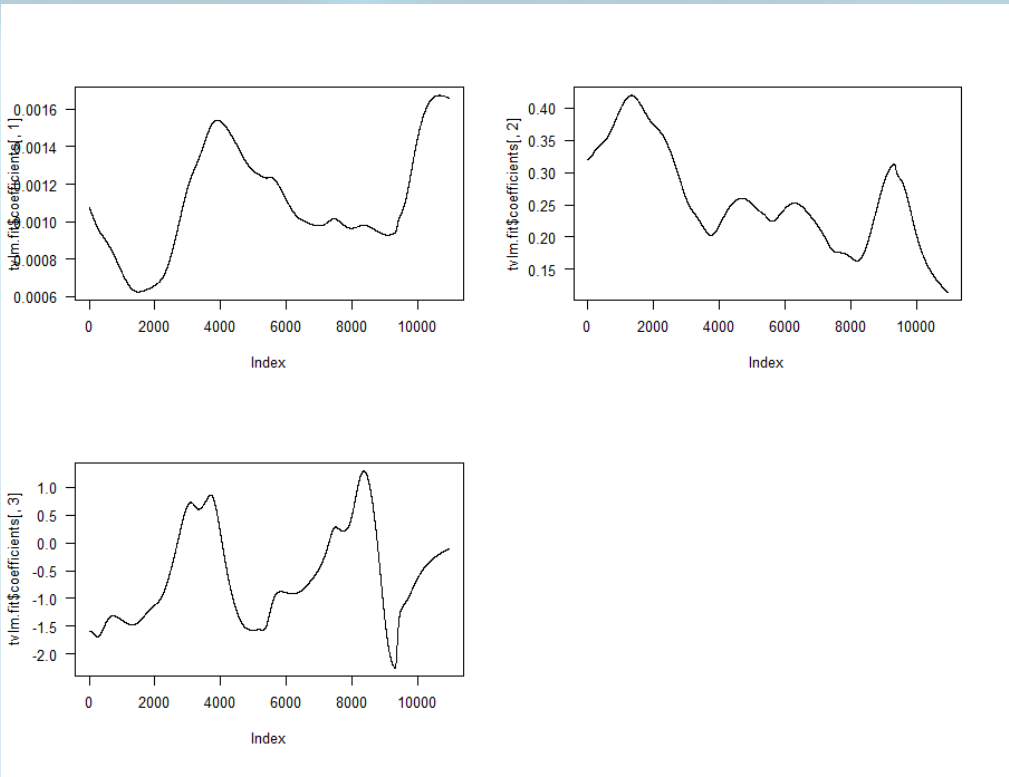
TV regression, R

- We will be using `tvLM {tvReg}`

Function	Class	Methods for class	Functions for class
tvLM	tvlm	bw, coef, confint, fitted, forecast, plot, predict, print, resid, summary, tvOLS	
tvAR	tvar	bw, coef, confint, fitted, forecast, plot, predict, print, resid, summary, tvOLS	
tvPLM	tvPLM	coef, confint, fitted, forecast, plot, predict, print, resid, summary,	tvRE, tvFE
tvSURE	tvsure	bw, coef, confint, fitted, forecast, plot, predict, print, resid, summary, tvGLS	
tvVAR	tvvar	bw, coef, confint, fitted, forecast, plot, predict, print, resid, summary, tvOLS	tvAcoef, tvBcoef, tvIRF, tvPhi, tvPsi
tvIRF	tvirf	coef, confint, plot, print summary	

TV regression, R

```
tvlm.fit <- tvLM(y~x1+x2, bw = NULL  )
> tvlm.fit <- tvLM(y~x1+x2, bw = NULL  )
Calculating regression bandwidth... bw =  0.1068501
> tvlm.fit

Class:  tvlm

Mean of coefficient estimates:
==============================
(Intercept)          x1          x2
     0.0011      0.2592     -0.6086

Bandwidth:  0.1069
#It represents the bandwidth in the estimation of trend
coefficients. If NULL, it is selected by cross
validation.

> par(mfrow=c(2,2))
> plot(tvlm.fit$coefficients[,1], type="l")
> plot(tvlm.fit$coefficients[,2], type="l")
> plot(tvlm.fit$coefficients[,3], type="l")
```



Other types of TV models

- Generalised additive models (GAM), introduced by Hastie and Tibshirani (1993). In R packages `{gam}` and `{mgcv}`
  - The GAM is a family of semiparametric models that extends parametric linear models by allowing for non-linear relationships of the explanatory variables and still retaining the additive structure of the model.
- State-space model (Kalman filter):
  - Package {`KFAS`} follows Durbin and Koopman (2001) tradition - time-varying linear Gaussian state space models, both univariate and multivariate, univariate exponential family (Poisson and Binomial) linear state space models. Simulation-based Bayesian inference, via Markov chain Monte Carlo methods.
  - package {`dlm`} follows West and Harrison (1997) tradition. It was developed to accompany Petris, Petrone, and Campagnoli (2009); The main focus is on Bayesian analysis, but maximum likelihood estimation of unknown parameters is also supported. Time-varying linear Gaussian state space models, both univariate and multivariate, can be also estimated.
- https://www.jstatsoft.org/article/download/v041i04/488

Bayesian regression

General logic

The Bayesian approach is more comprehensive.
It returns no single value, but a whole probability distribution for the unknown parameter θ conditional on data.
This probability distribution, $\mathrm{P}(\theta|data)$, is called posterior.
The posterior comes from one of the most celebrated works of Thomas Bayes that you have probably met before,

$$\mathrm{P}(\theta|data) = \frac{P(data|\theta) \times P(\theta)}{\int P(data|\theta) \times P(\theta) d\theta}$$

or, in plain words, $Posterior = \frac{Likelihood \times Prior}{Average\ Link}$
The posterior can be computed from three key ingredients:
A likelihood distribution, $P(data|\theta)$;
A prior distribution, $P(\theta)$;
The 'average likelihood', $\int P(data|\theta) \times P(\theta) d\theta = P(data)$.

Bayes theorem updates some prior belief by accounting the observed data
The following reconstruction of the theorem in three simple steps will seal the gap between frequentist and bayesian perspectives.

Bayesian regression in R, example

$$CSAD_{m,t} = \beta_0 + \beta_1|R_{m,t}| + \beta_2 R_{m,t}^2 + e_t$$

```
Bayesian estimates
# Bayesian models
library (brms)
hourly = cbind(y, x1, x2)
model <- brm(formula = y ~ x1+x2,
             data    = hourly,
             seed    = 123)
>summary(model)

Family: gaussian
  Links: mu = identity; sigma = identity
Formula: y ~ x1 + x2
   Data: hourly (Number of observations: 10953)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.00      0.00     0.00     0.00 1.00     2106     2481
x1            0.23      0.00     0.22     0.24 1.00     1250     1659
x2           -0.35      0.05    -0.46    -0.25 1.00     1097     1141

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.00      0.00     0.00     0.00 1.00     1687     1817

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the
potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

- `brm{brms}` implements Bayesian multilevel models in R using the probabilistic programming language Stan implementing Hamiltonian Monte Carlo (Duane, Kennedy, Pendleton, and Roweth 1987; Neal 2011).
- Many distributions and link functions are supported, allowing users to use linear, robust linear, binomial, Poisson, survival, response times, ordinal, quantile, zero-inflated, hurdle, and even non-linear models all in a multilevel context.
- modeling options include autocorrelation of the response variable, user defined covariance structures, censored data;

$$y_i \sim D(f(\eta_i), \theta)$$

- Where the prediction of the response y is through the linear combination η of predictors transformed by the inverse link function f assuming a certain distribution D for y.
- The parameter θ describes family specific parameters that typically do not vary across data points, such as the standard deviation σ in normal models or the shape α in Gamma or negative binomial models. The linear predictor can generally be written as

$$\eta = X\beta + Zu$$

- In this equation, β and u are the coefficients at population-level and group-level respectively and X, Z are the corresponding design matrices. The response y as well as X and Z make up the data, whereas β, u, and θ are the model parameters being estimated. The coefficients β and u may be more commonly known as fixed xed and random effects
- Details: https://cran.r-project.org/web/packages/brms/vignettes/brms_overview.pdf

Bayesian regression in R, example

$$CSAD_{m,t} = \beta_0 + \beta_1|R_{m,t}| + \beta_2 R_{m,t}^2 + e_t$$

```
Bayesian estimates
# Bayesian models
library (brms)
hourly = cbind(y, x1, x2)
model <- brm(formula = y ~ x1+x2,
             data    = hourly,
             seed    = 123)
>summary(model)

Family: gaussian
  Links: mu = identity; sigma = identity
Formula: y ~ x1 + x2
   Data: hourly (Number of observations: 10953)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.00      0.00     0.00     0.00 1.00     2106     2481
x1            0.23      0.00     0.22     0.24 1.00     1250     1659
x2           -0.35      0.05    -0.46    -0.25 1.00     1097     1141

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.00      0.00     0.00     0.00 1.00     1687     1817

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the
potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

- In general, every parameter is summarized using the mean (Estimate) and the standard deviation (Est.Error) of the posterior distribution as well as two-sided 95% Credible intervals
- (l-95% CI and u-95% CI) based on quantiles.
- The Rhat value provides information on the convergence of the algorithm (cf., Gelman and Rubin, 1992). If Rhat is considerably greater than 1 (i.e., > 1.1), the chains have not yet converged and it is necessary to run more iterations and/or set stronger priors.

Other types of Bayesian estimates

- Please visit https://cran.r-project.org/web/views/Bayesian.html for the list of the packages

Markov regime switching regression

Markov regime-switching model

- Markov-switching models are widely used in the social sciences.
  - For instance, in economics, the growth rate of Gross Domestic Product is modeled as a switching process to capture the asymmetrical behavior observed over expansions and recessions (Hamilton 1989).
  - Other economic examples include modeling interest rates (Garcia and Perron 1996) and exchange rates (Engel and Hamilton 1990). In finance, Kim, Nelson, and Startz (1998) model monthly stock returns, while Guidolin (2011b, 2011a) provide many applications of these models to returns, portfolio choice, and asset pricing.
- Markov-switching models are used for series that are believed to transition over a finite set of unobserved states, allowing the process to evolve differently in each state.
  - The transitions occur according to a Markov process.
  - The time of transition from one state to another and the duration between changes in state is random.
- For example, these models can be used to understand the process that governs the time at which economic growth transitions between expansion and recession and the duration of each period.

Markov regime-switching model, cont.

- a simple regime switching switching model, which constitutes a special case of hidden Markov models (HMMs).
- These models allow for greater flexibility to accommodate for non-stationarity (we will speak about it later) in the time series data.
- From application point of view, these models are very useful in assessing the state of the economy/market.
- Suppose we have a x_t stochastic process that has the following conditional distribution for t=1,...,t:

$$x_t|x_t = \mathrm{s} \sim \mathrm{N}(\mu_s, \sigma_s^2)$$

- this indicates if we know the current state s_t, then we know that x_t follows a normal distribution with mean μ_s and volatility σ_s^2.
- Nonetheless, in practice, we do not know the state nor we observe its realization.
  - We can, however, infer it from the data.
- This requires making assumption about the process s_t, which usually assumed to follow a first order Markov process. Specifically, the probability to transition to state *j* at time *t*, given that we were in state *i* at *t−1* is given by

$$\mathrm{P}(s_t = j|s_{t-1} = i) = p_{ij}$$

- the probability to transition in one step depends only on the previous state, hence, the Markovian property.
- it is common to represent these transition probability in a matrix, known as transition matrix, denoted by P.
- In particular, for *S* states, p_{ij} denotes the *i*th and *j*th column of P, for all *i,j*=1,..,S.
- For instance, for S=2, the transition matrix is given by

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

- where p_{12}=1−p_{11} and p_{21}=1−p_{22}. In this case, the transition matrix is identified using two parameters only, p_{11} and p_{22} which denote the persistence of state 1 and 2, respectively.
- `msmFit {MSwM}`

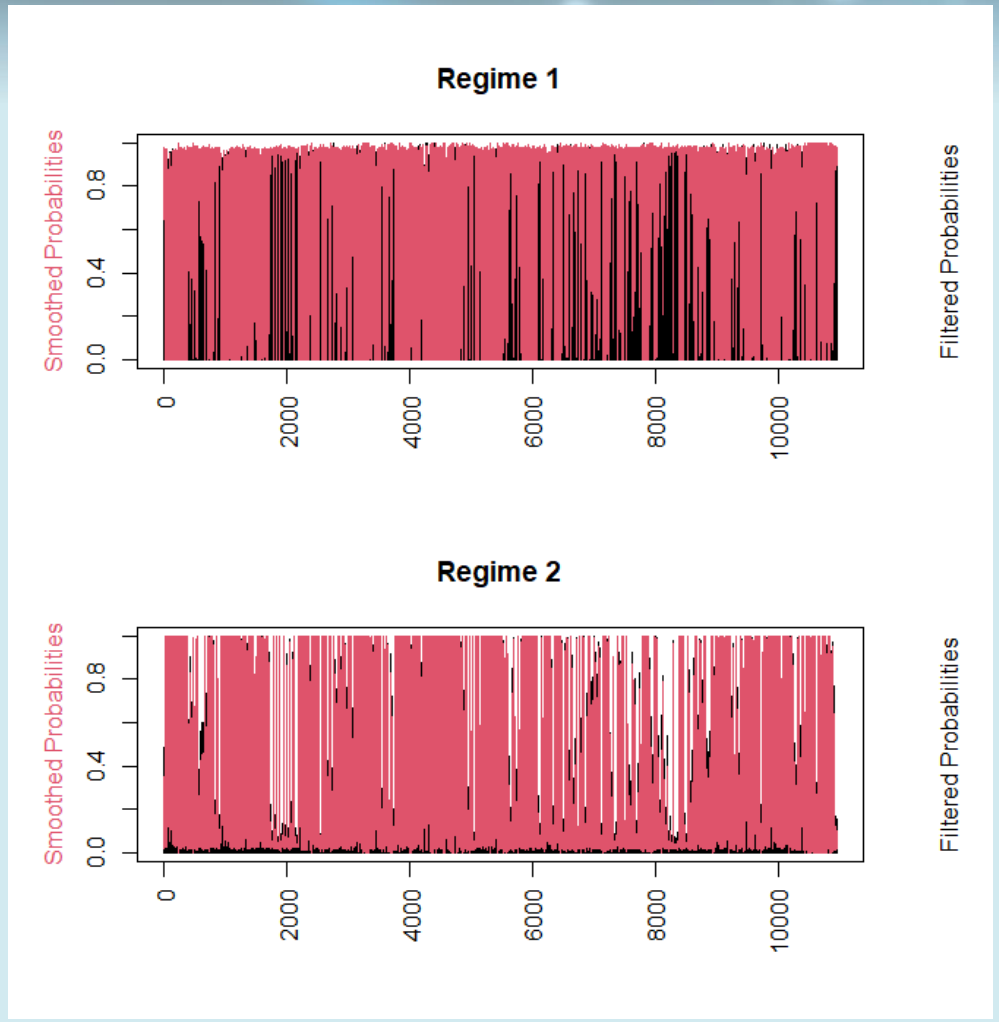
Markov regime-switching model, cont

```
# Markow regime-switching model
nstates <- 2
msEuro = msmFit(linearMod, k = nstates, sw = rep(TRUE, 4))
summary(msEuro)
plotProb(msEuro ,which=1)
```

```
> summary(msEuro)
Markov Switching Model
Call: msmFit(object = linearMod, k = nstates, sw = rep(TRUE, 4))

      AIC      BIC logLik
  -111926 -111826  55969

Coefficients:

Regime 1
---------
                Estimate Std. Error t value Pr(>|t|)
(Intercept)(S)     0.001      0.000     Inf   <2e-16 ***
x1(S)              0.156      0.005    31.2   <2e-16 ***
x2(S)             -1.867      0.115   -16.2   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.000972
Multiple R-squared: 0.255

Standardized Residuals:
      Min        Q1       Med        Q3       Max
-4.28e-03 -5.71e-04  1.42e-05  6.71e-04  3.12e-03

Regime 2
---------
                Estimate Std. Error t value Pr(>|t|)
(Intercept)(S)     0.003      0.000     Inf   <2e-16 ***
x1(S)              0.319      0.010    31.9   <2e-16 ***
x2(S)             -1.055      0.103   -10.2   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.00274
Multiple R-squared: 0.519

Standardized Residuals:
      Min        Q1       Med        Q3       Max
-0.012739 -0.000609 -0.000382 -0.000233  0.020278

Transition probabilities:
         Regime 1  Regime 2
Regime 1 0.893     0.387
Regime 2 0.107     0.613
```

Markov regime-switching model, cont.

```
> msEuro
Markov Switching Model
Call: msmFit(object = linearMod, k = nstates, sw =
rep(TRUE, 4))

        AIC       BIC   logLik
  -113645.1 -113545.5 56828.55

Coefficients:
        (Intercept)(S)     x1(S)     x2(S)       Std(S)
Model 1    0.001142539 0.1187830 -0.570413 0.0008599728
Model 2    0.002242006 0.3356749 -1.264380 0.0025122032

Transition probabilities:
         Regime 1  Regime 2
Regime 1 0.881343 0.3897501
Regime 2 0.118657 0.6102499

> plotProb(msEuro ,which=1)
```



Quantile regression

Quantile Regression

- Quantile Regression method was proposed by Koenker and Bassett in 1978 because linear regression models are not flexible against extreme values;

$$Y_q = \beta_0 + \beta X + \epsilon$$

- The difference from linear regression is that the dependent variable depends on the quartile value.
- In R – rq {quantreg}

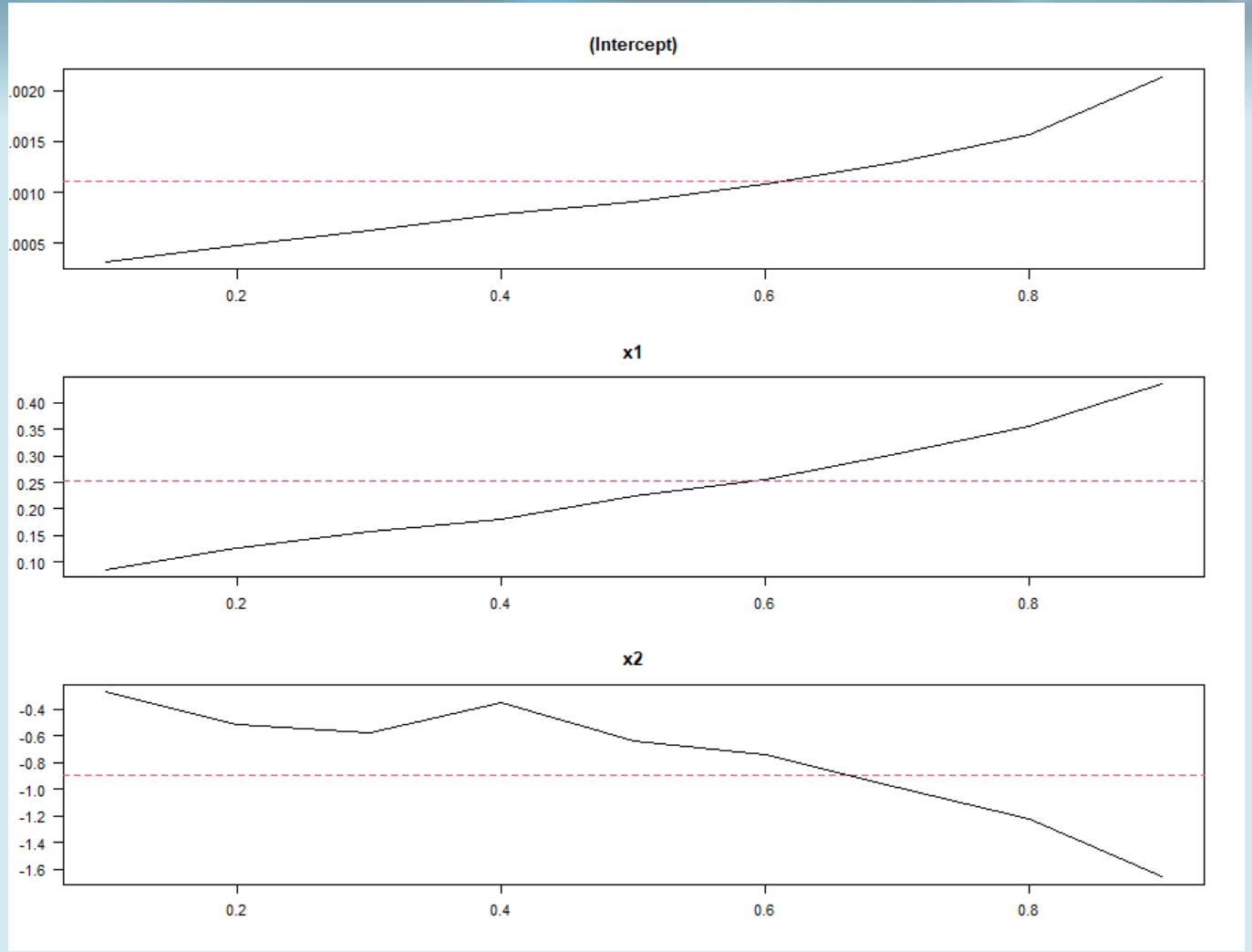
Quantile Regression

```
taus<-seq(from = .1, to = .9, by = .1)
coef0 <- rq( y ~ x1+x2, tau=taus);
summary (coef0);
plot (coef0, type = "l")
```

```
Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.1

Coefficients:
            Value    Std. Error t value  Pr(>|t|)
(Intercept)  0.00031  0.00003   11.75177  0.00000
x1           0.08622  0.01256    6.86341  0.00000
x2          -0.26696  0.65625   -0.40680  0.68417

Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.2

Coefficients:
            Value    Std. Error t value  Pr(>|t|)
(Intercept)  0.00047  0.00002   25.92344  0.00000
x1           0.12752  0.00732   17.41718  0.00000
x2          -0.51005  0.24976   -2.04214  0.04116

Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.3

Coefficients:
            Value    Std. Error t value  Pr(>|t|)
(Intercept)  0.00062  0.00003   21.09044  0.00000
x1           0.15850  0.01250   12.68155  0.00000
x2          -0.57732  0.60356   -0.95652  0.33883

Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.4

Coefficients:
            Value    Std. Error t value  Pr(>|t|)
(Intercept)  0.00079  0.00002   42.81287  0.00000
x1           0.18065  0.00699   25.84411  0.00000
x2          -0.35343  0.20436   -1.72943  0.08376

Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.5

Coefficients:
            Value    Std. Error t value  Pr(>|t|)
(Intercept)  0.00091  0.00002   45.65603  0.00000
x1           0.22360  0.00669   33.42416  0.00000
x2          -0.64017  0.15287   -4.18767  0.00003
```

Quantile Regression

```
Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.6

Coefficients:
            Value     Std. Error t value   Pr(>|t|)
(Intercept)  0.00108  0.00002    46.50253  0.00000
x1           0.25630  0.00769    33.31076  0.00000
x2          -0.73793  0.20071    -3.67667  0.00024

Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.7

Coefficients:
            Value     Std. Error t value   Pr(>|t|)
(Intercept)  0.00130  0.00003    44.77076  0.00000
x1           0.30435  0.01060    28.71632  0.00000
x2          -0.98278  0.39085    -2.51445  0.01194

Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.8

Coefficients:
            Value     Std. Error t value   Pr(>|t|)
(Intercept)  0.00157  0.00004    42.33133  0.00000
x1           0.35656  0.01306    27.29868  0.00000
x2          -1.21659  0.39016    -3.11820  0.00182

Call: rq(formula = y ~ x1 + x2, tau = taus)

tau: [1] 0.9

Coefficients:
            Value     Std. Error t value   Pr(>|t|)
(Intercept)  0.00213  0.00006    36.53524  0.00000
x1           0.43485  0.01773    24.52756  0.00000
x2          -1.65431  0.44358    -3.72948  0.00019
```



Optional: other methods of herding estimation

- Information Cascades:
- This model proposes that individuals make decisions based on the actions of others, rather than their own private information.
- As people observe others' actions or get information about sentiments, they may infer that those actions are based on superior information and choose to mimic them, leading to a cascade effect.

Optional: other methods of herding estimation

- To estimate information cascades using R, you can create a simple simulation of individuals making decisions based on the actions of others.
- In this example, we'll simulate a group of individuals deciding whether to invest (1) or not invest (0) in a financial asset.
- Each individual receives a private signal about the true value of the investment and makes a decision based on their own signal and the decisions of others before them. (see `InformationCascade.R`)

Optional: other methods of herding estimation

```
# Set parameters
num_individuals <- 100
# True value of the investment (1: good, 0: bad)
true_value <- 1

# Probability that an individual's private signal matches the true value
signal_accuracy <- 0.6

# Generate private signals
set.seed(42)
private_signals <- rbinom(num_individuals, 1, ifelse(true_value == 1, signal_accuracy, 1 - signal_accuracy))

# Initialize decision vector
decisions <- numeric(num_individuals)

# Function to calculate the likelihood ratio of the observed decisions
calc_likelihood_ratio <- function(decisions_so_far, signal_accuracy) {
  num_invest <- sum(decisions_so_far)
  num_not_invest <- length(decisions_so_far) - num_invest
  (signal_accuracy / (1 - signal_accuracy))^num_invest * ((1 - signal_accuracy) / signal_accuracy)^num_not_invest
}
```

```
# Simulate the decision-making process
for (i in seq_along(private_signals)) {
  if (i == 1) {
   # First individual makes a decision based solely on their private #signal
    decisions[i] <- private_signals[i]
  } else {
    # Subsequent individuals consider the decisions of those before #them
    decisions_so_far <- decisions[1:(i - 1)]
    likelihood_ratio <- calc_likelihood_ratio(decisions_so_far, signal_accuracy)

    # Compare the likelihood ratio to the individual's private signal
    if (private_signals[i] == 1 && likelihood_ratio > 1) {
      decisions[i] <- 1
    } else if (private_signals[i] == 0 && likelihood_ratio < 1) {
      decisions[i] <- 0
    } else {
      decisions[i] <- private_signals[i]
    }
  }
}
```