|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Multiple comparison | For a factor w a levels, we want to find in the contrast space , which contrasts are significantly nonzero  2 situations: a) general exploring of all contrasts to discover significant ones  b) Certain important contrasts that we are interested in, so only need to check if these are significant  Multiple comparison ensures prob of making any false discovery is controlled at a desired level | | | | | | |
| Family-wise type I error rate | In multiple comparison problem, we investigate many contrasts which form a family.  If a contrast is "actually" 0 but turn out to be "significant", these apparently significant contrasts are called artifacts  To avoid claiming artifacts as significant contrasts, the family-wise type I error rate must be controlled  Denote family of contrasts by . For any j , let Tj be test statistic & decision rule to reject H0 be Tj ≥ cj for a critical value cj  Family-wise type I error rate is P()  If for each individual contrast the type I error rate is controlled at , i.e. P(Tj ≥ cj) = , then P() > P(Tj ≥ cj) = , then family-wise type I error rate cannot be controlled at level | | | | | | |
| General exploration | Find if there is any contrasts of grp means which are statistically significant.  For main effects: grp means are those at the level of a factor  For interaction effect: grp means are those at the level combination of 2 factor  So let be the vector of the grp means. Let **c** denot a contrast vector of and denote set of all possible contrasts.  Then H0: **c**T= 0 for all **c** . H0 can be expressed in terms of the basis contrasts:  Let be the vector of basis contrasts. H0 is then equivalent to **a**T = 0 for any **a**  In main effects: = (, ..., ) or (,..., )  In interaction effects: = ()i=2,...,a; j=2,...,b | | | | | | |
| Scheffe's soln  For an individual contrast **a**T, test statistic T**a** = , where is estimate of , is estimated var matrix of  Need to find critical value s.t. P() ≤  To control family-wise type I error rate at level , Scheffe's soln is to take for all j, cj = = , where m1 is num of components of and m2 is df of , is the upper quantile of the dist. ( aka Scheffe's criterion) | | | | | | |
| Derivation  Note  So = =  = (from = (AB-1)) = (since (AB) = (BA) if A and B are compatible) = (since eigenvalue of scalar is just scalar) = (since = )  And follows . Thus = , i.e. = , m­2 = n-a for 1 way ANOVA | | | | | | |
| One-way ANOVA | If multiple comparison is done directly w contrasts of level means (and not in terms of indep contrasts, i.e. Scheffe's), the test statistic for contrast is and =  Note =  Let **b** = , **c** = (c1,...,ca)T, D = diag()  Then = = **b**TD-1**b** = SSA =  And = (a-1)MSA/MSE (since SSA = (a-1)MSA, = MSE) (where MSA/MSE is )  Then can reach same conclusion as Scheffe's soln? | | | | | | |
| Let T**c** = , H0: = 0.  Then the simultaneous p-value of 2-sided test is P() = P() = P()  For 1-sided test: p-value is (1/2)P() and critical value is = , | | | | | | |
| Individual & Simultaneous CI | | | | Simultaneous CI: CI which covers all params  Simultaneous CI w confidence coefficient 1- for all j require that P() ≥ 1-  Scheffe's simultaneous CI: P = 1 – P = 1-  Simulateous CI for the contrasts is ± | | | |
| Approach for general exploring | 1) Conduct overall significance test to see if there is any effect on Y (if no -> stop)  2) If significance test is significant, find particular significant effects (impossible to investigate all possible contrasts, so just look at rather diff in summary data or look at estimated regression coefficients) | | | | | | |
| Example | SSA = . SSE = . SST = ()S2. MSA = SSA/(a-1). MSE = SSE/(n-a). F-ratio = MSA/MSE  1) So if p-value = P(Fa-1,n-a > F-ratio) = ... < .... Then there is significant effect of factor A  2) Say we want to check . Then **c** = (1,0,-1,0) if factor A has 4 levels.  So = . SE() = = . T = /SE(). p-value = P(Fa-1,n-a ≥ T2/(a-1))  Can calculate intervals for : ± SE()  If interval contains 0 -> contrast not sig; interval don't contain 0 -> contrast sig | | | | | | |
| Multiple comparison through LRM | LRM for one-way ANOVA is , i = 1,..,n where = , k ≥ 2  For any contrast, we have = = =  Test statistic: TC = , where = (c2,...,ca)T, is estimated covariance matrix of (,...,)  TC is same as | | | | | | |
| Example2  (General Exploring) | levels <- factor(kronecker(c(1:4), rep(1,3))  fit <- lm(y~package); summary(fit); anova(fit)  b <- fit$coef[-1], V=vcov(fit)[-1,-1]  C <- matrix(c(1,0,0, 0,0,1, 1,-1,0,...), ncol=3, byrow=TRUE)  L <- C%\*%b; VC <- C%\*%V%\*%t(C); sd <- sqrt(diag(VC));  T <- L/sd; p.value <- pf(T^2/(a-1), a-1, nERR, lower.tail=FALSE) | | | | | | #4 levels, 3 observations each  #F-test p-value: effect of level on y is significant  #General exploring: if beta p-value sig or diff btw estimate is larger than a sig estimate, explore those contrast (e.g. C = , , ,...)  #sd need diag as C might not be sq matrix  #a = num of levels, nERR = n-a |
| Conclude if diff btw packages is signifiicant by p-value | | | | | | |
| Pairwise contrasts | Only interested in certain pairwise contrasts, , 1 ≤ k < j ≤ a. Overall significance F-test not necessary  Only need to find s.t P = | | | | | | |
| Studentized range dist | Let , i=1,...,a be sample means of a samples w equal sizes (ni = n)  Studentized range statistic is =  Let denote upper -quantile of the Studentized range dist, aka Tukey's criterion for pairwise comparison at level | | | | | | |
| Pairwise comparison procedure | For studentized range dist, Q-statistic for is Qij = , where  Contrast is significant at level if Qij > | | | | | | |
| Diff btw Q-statistic & t-statistic | Note T**c** = . And for contrast , TC = Tij = . So |Tij| = Qij/  t-statistics can be used for pairwise comparison using Tukey's criterion, but |Tij| must be compared w  i.e. contrast is significant at level if Qij > OR |Tij| > | | | | | | |
| Equivalent form in terms of regression coefficient | | | W conventional defn of dummy vars, the regression coefficients = , i = 2,...,a OR = , i,j > 1  t-statistics: Tij = | | | | |
| Tukey's simultaneous CI | | | For = : ± sd()  For = , i,j > 1: () ± | | | | |
| Example pairwise comparison | | | | | | Cij = , , Qij > -> diff is sig for | |
| Example2 pairwise comparison & Tukey's CI | | | | | C <- matrix(c(1,0,0, 0,1,0, ...), ncol = a-1, byrow=TRUE)  L <- C%\*%b; VC <- C%\*%V%\*%t(C); sd = sqrt(diag(VC); T = L/sd  cbind(L, sd, T). compare w to check significance of difference  Tukey's CI: cbind(L – sd\*, L + sd\*) | | |
| Bonferroni's Mtd | | If there are only k contrasts we are interest in, the overall type I error rate for the k contrasts can be controlled by Bonferroni's Mtd: , where is type I error rate for contrast j  Each can be specified, but in general just use = /k. Critical value = Tn-a(/k\*2). Check |Tj| > for 2-sided test | | | | | |
| Rationale: P() ≤ , where P() ≤ ≤  Thus Bonferroni method is a conservative method | | | | | |
| p-values: p-value for jth test is pj = kP(Tj ≥ ) where is observed value of Tj and prob computed under dist of Tj  If critical value for all individual test is set at , then = P(Tj ≥ ) for j = 1,...,k and overall type I error rate is = k  i.e. p-value is the overall type I error rate when critical value is observed value of the statistic | | | | | |
| Example3 | Contrast interested: ;  Use regression approach and get estimated regression coefficients (col vector), var()  Contrast in terms of 's: ; OR **c**1, **c**2, where  c1 = (0.5,-0.5,-0.5); c2 = (-0.5,0.5,-0.5)  L1 = c1%\*%b; s1 = sqrt(c1%\*%V%\*%t(c1)); T1 = L1/s1  L2 = c2%\*%b; s2 = sqrt(c2%\*%V%\*%t(c2)); T2 = L2/s2  T = c(T1, T2); k\*2\*pt(abs(T), 15, lower.tail=FALSE). Then conclude based on p-value if contrast is significant | | | | | | |
| Summary | General exploration: Scheffe's criterion  Pairwise comparison: all 3 mtds can be used, but Tukey's Mtd more efficient than Scheffe's and Bonferroni's  Pre-specified contrasts: Scheffe's and Bonferroni's. Bonferroni more efficient when num of pre-specified contrasts is small  For pre-specified contrasts: can just compute both Scheffe's and Bonferroni's, smaller criteria value -> more efficient | | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LRM w both factor & quant-itative predictor | ANOCOV (analysis of covariance) model is a LRM w both factor & quantitative predictors,  where uj, j = 2,...,l are dummy variables representing a factor predictor and x is a quantitative predictor, is diff of Y btw type j and type 1 when X is same for both types, is effect of X on Y (i.e. when X change by a unit, Y has an expected change of )  In general ANOCOV model could have more than 1 factor and quantitative predictor. There could be both main and interaction effects | | | | |
| Traditionally, 1) ANOCOV = comparison of treatment effects of factor predictors, adjusting for effect of certain concomitant variables (which have effect on response as well) so comparison is more efficient  2) Comparison of regression fns: r/s btw response var and quantitative predictors is studied in diff categories. Regression fn is then compared | | | | |
| ANOCOV is based on adjusted SS's (which adjust for effect of concomitant vars). But adjusted SS's have complicated formulae and have similar limitations as traditional ANOVA  W regression approach, when estimating factor effects, effect of concomitant vars is auto adjusted, & avoids drawbacks of traditional ANOVA | | | | |
| Example | | data = read.table('', header=TRUE); data$category = factor(data$category)  fit = lm(y~x+category, data=data); summary(fit); anova(fit)  Note ANOVA table produced depends on order of var, only 2nd variable has valid SS, F and p-values. Recommended to put concomitant var first followed by factor | | | |
| Correct F value can be obtained through Wald statistics regardless of order  If lm(y~category+x), F value for category: b = fit$coef[c(2,...,k)]; V = vcov(fit)[c(2,...,k), c(2,...,k)]; F = t(b)%\*%solve(V)%\*%b/(k-1) | | | |
| If lm(y~x+category): from ANOVA table, category has significant effect on Y, since F-statistic for category has p-value of ...  From summary table, category .. are significantly diff from type 1, since p-value are ... (using t test)  Category ... is worse than category ..., since estimate of is ...  X has a significant effect on Y since p-value from summary table is ... | | | |
| Example2 | | 2 factors & quantitative predictor:  (bias + factor A + factor B + interaction AB + concomitant X + error)  EY = . So | | | |
| data <- read.table('', header = TRUE); data$A <- factor(data$A); data$B <- factor(data$B)  AB.fit <- lm(y~x+A\*B, data = data); anova(AB.fit)  BA.fit <- lm(y~x+B\*A, data = data); anova(BA.fit)  summary(AB.fit); vcov(AB.fit) | | | #anova table correct for x and interaction  # B has correct p-value  # A has correct p-value  # all values in summary table and vcov correct |
| From summary table, can get , , , estimate and p-value and make inference from there (meaning of var explained above)  To get or estimate and p-value: vcov(AB.fit)[c(3,4,5), c(3,4,5)]  For : test statistic, T = ()/[var() + var() + 2cov(,)]1/2. p-value = 2P(tn-ab-1 > T) (additional -1 due to X) | | | |
| Comparison of regression lines | | | Find whether regression lines have same intercept and whether have same slope -> Use ANOCOV model w interaction btw factor and quantitative predictor: Y = , i.e.  Category 1: Y = . Category i ≥ 2: . So just making inference on , and | | |
| Example 3 | | Y = (since only have 2 categories)  data <- read.table('', header = TRUE); y <- data$y; x <- data$x; u <- data$u; u <- factor(u)  fit <- lm(y~x\*U); summary(fit)  Can make inference on for if X has significant linear r/s w Y  Inference on and , if significant -> difference of regression lines are significant  Depending on sign of slope, can say whether category has a positive/negative effect on Y | | | |
| LRM w non-linear predictor terms | | | | In LRM: Y = + X1 +...+ Xp + , X need not be diff predictor variables, could be non-linear fns of predictor variables  E.g. Y = +Z+ Z2 +...+ Zp + | Y = + (1/X) + (inverse model) | log(Y) = + log(x) + log(v)+ (log model) | |
| Polynomial regression models | | yi = + x1i + x2i + + + , i = 1,...,n (model can have more than 1 predictor variables)  yi = + + + + + (centralization to multicollinearity of model) | | | |
| Example 4 | | data <- read.table('', header=TRUE); y <-data$y; x <- data$x; x.c <- x-mean(x)  fit <- lm(y~x.c + I(x.c^2) + I(x.c^3)); summary(fit)  By looking at F-statistic p-value -> y and x have a significant cubic r/s  Although w and w/o centralization F-statistic, R2, residual standard error and I(x.c^3) all have same values, I(x.c^2), x.c, and intercept have larger SE w/o centralization due to multicollinearity | | | |
| Piece-wise linear models | | R/s btw Y and X might be diff over diff ranges of X  Hence use Auxiliary X truncated at point X = c: = (X - c)+ =  Model from yi = + Xi + becomes yi = + Xi + + (piece-wise linear in X)  Slope of model changes from to at point X = c. Can continue adding more auxiliary terms if r/s diff at diff points | | | |
| Example 7 | | data <- read.csv('', header=TRUE); price <- data$x, n <- length(price), return <- log(price[1:(n-1)]) - log(price[2:n]); m <- n-1  r1 <- return[2:m]; r2 <- return[1:m-1]  reg0 <- lm(r2~r1); summary(reg0)  r1A <- r1\*(r1>0); reg1 <- lm(r2~r1+r1A); summary(reg1); I = order(r1); Expected <- reg1$coeff[1]+reg1$coeff[2]\*r1+reg1$coeff[3]\*r1A  plot(r1[I], Expected[I], type = 'l', xlab='Previous return', ylab='Expected next return')  Estimated coefficient of r1A has p-value ... -> significant.  Piece-wise linear model increase the R^2 from ... to ... (reg0 vs reg1), implying it explain an additional significant amt of variation in r2 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Predictors | | Relevant predictor: can explain a proportion of the variation of Y which cannot otherwise be explained by other predictors  Relevant predictor is either a causal variable OR a surrogate of certain causal variables not observed  Irrelevant predictor is neither a causal variable nor a surrogate of causal variable, BUT might still have correlation w Y.  This "fake" correlation aka spurious correlation. Might be that causal variable causes variation in Y and in the irrelevant predictor  Irrelevant predictor might have correlation w Y due to data structure (small-n-large-p) where num of predictors > num of observations | | |
| In most cases, model selection is equivalent to variable selection  Variable selection is too identify the relevant predictors (causal or surrogate of causal predictors)  Variable selection emphasizes accuracy of predictor selection, but model selection emphasizes of accuracy of prediction | | |
| Under / Over- Fitting | | | Under-fitting model: model which does not contain all relevant predictors  Over-fitting model: model containing irrelevant predictiors in addition to all relevant predictors | |
| Effect of under/over fitting on LSE | | Suppose design matrix is partitioned as X = (X1, X2) and includes all relevant predictors.  **y** = X + = (full model). **y** = (reduced model)  Under full model, , E = , Var() = . Var() =  Under reduced model, = , Var() = , E =  Under reduced model, LSE of is larger and have a bias of  Under reduced model, var of LSE of is smaller since > (then inverse it to get < )  On the other hand, over-fitting increase variance, but decrease bias | | |
| Effect of under/over fitting on prediction | | Let be the parameter of model M w design matrix XM. The estimator of E**y** = under model M is  and E and Var() =  , where M is num of cols of XM (usually num of predictors in model M + 1)  If we use model M to predict n unobserved y's w the same XM, the sum of prediction squared error (SPSE) is SPSE =  Under fitting increase bias of prediction, decr variance. Over fitting incr variance of prediction, decr bias | | |
| Principle of variable selection | | Accuracy of prediction is measured by SPSE which consists of a variance and bias component  Accuracy of estimate is measured by MSE which also consists of a variance and bias component. MSE =  Principle of variable selction is to select variables to balance variance and bias so that SPSE or MSE is minimized  SPSE or MSE cannot be practically computed, so model selection criterion uses a surrogate of SPSE or MSE in some sense | | |
| Selection Criteria | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | R2 and | Mallow's Cp | AIC | BIC | EBIC | CV | | Suitable for | models w same num of predictors | Good estimate of | Minimize Kullback-Leibler dist |  | models in small-n-large-p problems | based on estimated prediction error | | Better | Higher | Lower | Lower | Lower | Lower | Lower | | | | |
| R2 and | | R2 = , = R2 – (1 – R2) both estimate proportion of pop variation of response Y explained by predictors in regression model  However, R2 always incr when num of predictors incr. Does not strike a balance btw variance and bias  Though not strictly increasing as num of predictors incr, it still incr when predictor having a small contribution to explained variation is added to the model. | | |
| Mallow's Cp (Complexity parameter) | | Ideally, use to estimate SPSE. But don't have yn+i available. So use to estimate SPSE and find that  E = SPSE - 2|M|  Adjusting for bias, (unbiased estimate of SPSE)  Mallow's Cp = . Minimizing Cp is equivalent to minimizing  If model M is correct model, then ECp ≈ |M| | | |
| Akaike's information criterion (AIC) | | AIC minimizes the Kullback-Leibler dist btw model M (gM(y)) and the true model (f(y) approximated by the empirical dist) given by  I(f,gM) =  AIC approximates the 2nd component above (since can ignore constant = 1st term)  AIC = , where are MLE under model M, L is likelihood fn and jM is num of predictors in M  For multiple LRM w normality assumption, AIC = , where and C = n(ln(2π) + 1) | | |
| Bayesian information criterion (BIC) | | Let be the set of all possible models, Pr a prior probability measure on . Let Pr(M) be the prior on a model M and the prior on parameters of model M  The marginal density of data **y** given model M is m(**y**|M) =  The posterior probability of model M is p(M|**y**) =  If Pr(M) is taken as constant, BIC = | | |
| Extended Bayesian information criterion (EBIC) | | Let model space be partitioned according to num of predictors contained in the models as , where is set of all models containing j predictors. If Pr is the constant prior, then Pr() , where p is total num of predictors  When p is large, the prior probability of models w more predictors will be larger than those w less predictors. So BIC will select models w more predictors. EBIC accounts for this problem, by adding a non-constant prior on models  EBIC = , | | |
| Cross Validation (CV) | | Model M fitted using data (**y**, X). To assess goodness of model, ideally use another data set, () and validate by prediction error . But practically, same dataset is split into training and testing data. So wastes information and is against principle of sufficiency  Leave-out-one CV: Training data is n-1 observations, test data is 1 observation  k-fold CV: Whole data divided into k parts, each time, one part is test data, remaining k-1 parts for training data  Leave-out-one CV score: CV = , where is the estimate of by leaving out the ith data point (yi, **x**i)  k-fold CV score: CVk = , where (**y**j, Xj) is the jth part of data and is testing data, and is estimate obtained by training data | | |
| R codes | | AIC(model); BIC(model). If arg k=log(n) is set in AIC, then would become computing BIC  For CV, use package boot  glmfit = glm(Y~X1+X2+..., data = ...)  Leave-out-one Score: CV.1.f = cv.glm(data, glmfit)$delta[2]  k-fold score: CV.k.f = cv.glm(data, glmfit, K=k)$delta[2]  AIC(glmfit); BIC(glmfit) | | |
| Model selection strategy | | Naive method: all subset selection. Not practical as for p predictors, there are 2p possible models | | |
| 1) Remove redundant predictors (when p is not very large)  i. Fit full model, remove all predictors w p-value bigger than a certain level  ii. Fit full model, remove predictor w largest p-value which is > , then re-fit model w remaining predictors, repeat this until no predictor  has p-value >  If covariates are not highly correlated, the 2 options produce the same selected model.  If high correlation among covariates exists, ii. preferred | | |
| 2) Forward selection (sequential procedure)  Starts w null model M0 w no predictors, then add predictors 1 at a time, choosing predictor having largest contribution to reduce the residual sum of squares  Compare new model w old model by certain criterion. If new model better than old model -> continue; otherwise stop.  Criterion can be any except R2 and | | |
| 3) Backward Selection (inverted sequential method)  Start w full model MF w all predictors, then reduce model by removing predictors one at a time, choosing predictor w smallest contribution to reduce residual sum of squares to be removed.  Compare reduced model w previous model by AIC. If AIC of new < AIC of old -> continue; otherwise stop | | |
| 4) Stepwise selection (mixture of forward & backward selection. Can be done upwards OR downwards)  Upward stepwise selection; Start w null model. Add predictor to model, perform backward procedure until no predictor can be removed. Proceed to next forward step. Repeat  Downward stepwise selection: Start w full model. Remove predictor from model, perform forward procedure until no predictors can be added. Proceed to next backward step. Repeat | | |
| R code for selection | | library(MASS)  fitted = lm(y~...)  stepAIC(fitted, scope=list(upper = ~x1+x2+x3+x4+x5, lower = ~1), direction = ) | | #null model or full model  direction can be "both", "forward", or "backward" |
| Penalized likelihood approach | | For LRM **y** = X+ , the penalized likelihood approach select variables by minimizing , where is the penalty function, is the penalty parameter whose value is to be chosen  Procedure: specify sequence of values, at each value, carry out the penalized minimization, which yields a model w certain selected variables. Selection criteria is used to select the model  If purpose to obtain model for prediction -> CV. If purpose to identify important variables -> EBIC | | |
| Common penalty functions | | 1) LASSO penalty:  2) Adaptive LASSO penalty: , where wj is taken as . If p n, being the OLS (ordinary least square) estimator in multiple LRM. If p is close to or > n, is the OLS estimate is the marginal LRM  SCAD penalty: for near 0, and equals a constant C for large , the two parts are connected by a smooth function  MCP penalty: for large , it is a constant. Smoothly decreases to 0 w = as its asymptote when approaches 0 | | |
| Rationale of penalized likelihood approach | | E.g. LASSO (least absolute shrinkage and selection operator) estimates by minimizing  If = 0, LASSO estimator is same as LSE. If = ∞, all components of are estimated as 0  For a certain nonzero , some of the components will be estimated as nonzero, and others 0. The nonzero ones are shrunk version of LSE  Variables w nonzero estimated coefficients are the selected variables | | |
| R code for penalized likelihood approach | library(iterators); library(foreach); library(Matrix); library(shape); library(glmnet)  x = as.matrix(data$x); y = as.vector(data$y)  fit1 <- cv.glmnet(x,y)  coef(fit, s = fit1$lambda.min) | | | Note x here is w/o bias col (i.e. x = (X1 ... Xp))  #fit1$lambda.min is value producing smallest CV score |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model diagnostics | Assumptions made for model might not be true, leading to discrepancies. There are 2 types of discrepancies – systematic and local | | | | | | | | | | |
| Fitted values are , i = 1,...,n. Hat matrix is H = X(XTX)-1XT  Hat values (ith diagonal elem of H), = hat value of ith observation, where is the ith row of X | | | | | | | | | | |
| Partitioning X as X = (**1** Z), (XTX)-1 =  A22 = = sample covariance matrix, A11 = , A12 = , A21 =  Let ith row vector **x**i of X be and . Then A11 =  Then hii = = = = = . hii is + Mahalanobis dist of ith observation in x-space to centroid of sample | | | | | | | | | | |
| Pearson's residuals are ei = yi – , i = 1,...,n. Let **e** = (e1,...,en)T. Then **e** = **y** – = (I – H)**y**. E**e** = **0**, var(**e**) = (I – H). var(ei) = (I – hii)  Under normality assumption, **e** ~ N(**0**, (I – H)) | | | | | | | | | | |
| Studentized residuals,  Studentized deleted residuals, , where is the predicted value by fitted model w ith observation deleted, is the counter part of when ith observation is deleted  Cook's distance, di = , where is estimate of when ith observation is removed from data  Variance Inflation Factor (VIF). In regression model w p predictors X1,...,Xp. Let be the coefficient of determination of model Xk = . VIF of Xk is VIFk = | | | | | | | | | | |
| R codes: glm.fit = glm(y~x1+x2+x3+x4+x5+..., data = )  fitted = glm.fit$fitted.values; res = residuals(glm.fit, type = "pearson")  hat = hatvalues(glm.fit, type = "diagonal"); infl = influence(glm.fit, do.coef = FALSE)  rsta = rstandard(glm.fit, infl, type = "pearson");  rstu = rstudent(glm.fit, infl, type="pearson")  cook.d = cooks.distance(glm.fit, infl, res = infl$pear.res, dispersion = summary(glm.fit$dispersion, hat = infl$hat)  round(cbind(fitted, res, hat, rsta, rstu, cook.d), 4) | | | | | | | | | | #Pearson's residuals  #hat values  # studentized residuals  # studentized deleted residuals  #cooks distance |
| Systematic discrepancies | | | Caused by regression fn not linear, error terms don't have constant variance, error terms not indep, error terms don't have normal dist, impt predictors are omitted from model. Use residual plots to check for systematic error | | | | | | | | |
| If no discrepancies, residuals would appear like iid random errors w mean 0. In any residual plot, points are scattered evenly within a horizontal band around 0 | | | | | | | | |
| Check non-linearity | Plot Pearson's residual against fitted values  Plot Pearson's residual against predictor variables  Scatter plot of response against predictor variables | | | | | If any of the plots show a non-linear trend -> regression fn is not linear  For residual vs predictors: no trend -> no obvious discrepancy in regression fn | | | | | |
| Check homogeneity | | | Plot Pearson's residual against fitted values  Plot Pearson's residual against predictor variables | | | | If vertical range of residuals have obvious change along x-axis -> variances are not constant / not homogeneous | | | | |
| Check independence | | | | Plot residual against time/space | | | If indep -> should be constant horizontal trend | | | | |
| Check normality | Plot of studentized residuals through dist plot (box plot, histogram) OR normal probability plot of residuals OR QQ plot  If normality holds, points in QQ plot shld fall on straight line y = x | | | | | | | |  | | |
| Heavy tailed pattern | If dist of r.v. Y is skewed to the right (positive skew) relative to normal dist, then P(Y ≤ c ) ≤ P(Z ≤ c) for all c  Let yq and zq denote q-quantile of Y and Z, then P(Y ≤ yq) = P(Z ≤ zq) ≥ P(Y ≤ zq). Then P(Y ≤ yq) ≥ P(Y ≤ zq)  Hence yq ≥ zq for all q. In Q-Q plot where yq is plotted against zq, the point (zq, yq) is above the point (zq, zq) | | | | | | | | |  | |
| Check missing predictors | | | | | Plot residual against other predictors not included in the model | | | | | If any of the plot show a trend -> that predictor is missing | |
| Outliers | Leverage: whether point is far away from major cluster in x-space  Since = Tr(H) = p. Point is high leverage if hii > | | | | | | | | |  |  |  |  | | --- | --- | --- | --- | |  | a | b | c | | Leverage | low | high | high | | Consistency | No | Yes | No | | Influence | low | low | high | | | |
| Consistency: whether point is consistent in terms of fitting in the (x,y)-space  Studentized deletion residuals are the standardized prediction errors,  Find points w highest || values -> possible outliers | | | | | | | |
| Influence: whether point highly affects fitting of model  Find points w highest Cook's distance -> possible outliers | | | | | | | |
| Assessment of outliers | | Informal test of outliers: normal probability plots of studentized deletion residual, leverage h­i and Cook's distance di  Formal test: To assess (yi, **x**i), introduce the dummy variable u = .  Significance of coefficient of u in the linear predictor indicates ith point is an outlier. | | | | | | | | | |
| Example | Identify possible outliers from informal test: e.g. point 37,38,9,22,27,28  u1 = rep(0,n); u2=u3=u4=u5=u6=u1  u1[37]=1; u2[38]=1; u3[9]=1; u4[22]=1; u5[27]=1; u6[28]=1;  fitted = glm(Y~x1+x2+x3+x4+u1+u2+u3+u4+u5+u6, data = )  summary(fitted) | | | | | | | Significance of outliers judged by Bonferroni's criterion  Since testing 6 outliers, k = 6  Bonferroni's p-values = 6\*p-values in summary table of u1,...,u6  If Bonferroni's p-value < 0.05 -> outlier is significant | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. LRM w unequal variances | When 's don't have common variance, an unequal variance model is considered: ,  wherehas the variance matrix , where wi's are unequal weights (if common variance: ) | | | | | | |
| If wi's are known, the unequal variance model can be transformed into an equal variance model. Let  W = . Multiply by W1/2, then **.** Let **,** ,  Then var() . And model is a constant variance model | | | | | | |
| Minimizing , we obtain the estimate of as = weighted LSE (WLSE)  can be expressed explicitly as . The weight wi reflect the relative importance of in the estimation. The larger the variance of ith term, the smaller the corresponding weight, since weight is inversly proportional to the variance | | | | | | |
| For WLSE , E = , var() = . And  is estimated as . Inference on is made in same way as in normal LRM | | | | | | |
| Estimation of unknown weights | If no replicates of predictor values, Weights can be estimated by the following procedure:  1. Fit regression model by unweighted least squares and obtain residuals **r** and fitted values  2. Regress log of absolute residual on log of fitted values, e.g. ln |ri| = .  If smallest < 0, replace by , for some positive constant c  3. Weights are estimated as OR . Constant don't really matter | | | | | | |
| If there are replicates of predictor values, sample variance can be used in estimation of weights  0. Naive method: weight = 1/s^2, s = sample SD  1. Estimate variance as a function of the mean by using the regression model: ln si = , where si is sample sd, and is sample mean  2. The weight for the ith predictor value is | | | | | | |
| Example (no replicates) | data = read.table("", header=TRUE)  fit = lm(y~x); r = fit$resid; y.h = fit$fitted  plot(y.h, r, xlab="Fitted values", ylab="Residuals")  wt.fit = lm(log(abs(r))~log(y.h))  s = exp(wt.fit$fitted); w=1/s^2  w.fit = lm(y~x, weight=w);  r = w.fit$resid; y.h = w.fit$fitted  r.w = r\*sqrt(w); y.w = y.h\*sqrt(w);  plot(y.w, r.w, xlab="Weighted fitted values", ylab="weighted residuals") | | | | #check variance not constant as fitted value increase  #weights  #weighted model: **,**  #fitted values: , residuals:  #fitted values **,** residuals are | | |
| Model w weighted estimates usualy have smaller SE. | | | | Comparing R2don't make much sense | | |
| Example 2 (w replicates) | plot(ybar, sd) | | | #no constant variance | | | |
| Consider regression model w as response variable  Individual observation in are iid from a pop w mean and var . var of () is then | | | 2 types of weights to consider:  1. If 's don't have any r/s w , weights can be taken as  2. If have r/s, , weights can be taken as | | | |
| fit1 = lm(y~x1+x2+x3)  r0 = fit1$resid; y0 = fit1$fitted; plot(y0,r0) | | | # Unweighted  #no constant variance | | | |
| Weight taken as n/s^2  w2 = n/s^2  fit2 = lm(y~x1+x2+x3, weight = w2)  r2 = fit2$resid; y2=fit2$fitted;  r2.w = r2\*sqrt(w2); y2.w = y2\*sqrt(w2); plot(y2.w, r2.w) | | | Weight taken as n/  w.fit = lm(log(s)~log(y)); log.s = w.fit4fitted; s3 = exp(log.s)  w3=n/s3^2 #group individual variance as a fn of mean  fit3 = lm(y~x1+x2+x3, weight = w3); r3 = fit3$resid; y3=fit3$fitted  r3.w = r3\*sqrt(w3); y3.w = y3\*sqrt(w3); plot(y3.w, r3.w) | | | |
| By looking at both plots, n/s^2 and n/ seem to provide appropriate weights (null pattern) by residual plots  So, compare estimated sd of estimated params of both weights. Weight w smaller sd is preferred  Cross validation is ultimate criterion for choice of diff weights | | | | | | |
| Example 3 | Preliminary analysis:  C=factor(C); fit = glm(y~C+x1+x2); r = fit$resid; yhat = fit$fitted  par(mfrow=c(2,2)); plot(yhat,r); plot(C,r); plot(x1,r); plot(x2,r)  hat = hatvalues(fit, type="diagonal"); infl = influence(fit, do.coef = FALSE)  rsta = rstandard(fit, infl, type="pearson"); rstu = rstudent(fit, infl, type="pearson")  cook.d = cooks.distance(fit, infl, res = infl$pear.res, dispersion = summary(fit)$dispersion, hat=infl$hat)  par(mfrow=c(2,2)); qqnorm(hat); qqnorm(rstu); qqnorm(cook.d)  n = length(cook.d); cook.d[order(cook.d)][(n-3):n]  rstu[order(rstu)][c(1:3, (n-3):n)] | | | | | | Analysis show variances are not constant and observations 23 and 43 are influential outliers. So shall remove these 2 points from further analysis |
| data = data[-c(23,43),]  fit = glm(y~C+x1+x2, data = data);r = fit$resid; yhat = fit$fitted  weight = lm(log(abs(r))~log(yhat)); s = exp(weight$fitted); w = 1/s^2  w.fit = lm(y~C+x1+x2, data = data, weight = weight)  r.w = w.fit$resid\*sqrt(weight); y.w = w.fit$fitted\*sqrt(weight)  t.uw = lm(abs(r)~yhat); t.w = lm(abs(r.w~y.w))  par(mfrow=c(1,2))  plot(yhat,abs(r));abline(t.uw); plot(y.w,abs(r.w));abline(t.w) | | | #unweighted vs weighted  #unweighted vs weighted | | | |
| 2. Multi-collinearity & its effects | | Correlation among predictor variables such as pariwise correlation, linear dependence of one predictor on another predictor,...  If 1 predictor is perfectly linearly dependent on the other predictors, XTX would be singular, i.e. non-invertible  Practically, although perfect linear dependence will not occur, high multicollinearity can cause XTX to be nearly singular (i.e. having large condition number ), which renders the LSE extremely unstable  Serious multicollinearity greatly increases variance of LSE and make LSE inaccurate and useless | | | | | |
| Informal Diagnostics for multi-collinearity | | Large changes in estimated regression coefficients when a predictor is added/deleted, or an observation is altered or deleted  Nonsignificant results in individual test on the regression coefficients for known important predictor variables  Estimated regression coefficients w an algebraic sign that is opp of that expected from theoretical consideration or prior experience  Large coefficients of sample correlation btw pairs of predictors in the correlation matrix (linear trend, correlation > 0.6) | | | | | |
| 1) Formal diagnostic – VIF | Recall LSE of and its variance are = , var() = , where is the projection matrix of , being the submatrix of X w/o its jthcolumn  Note is the SSEj when **x**j is regressed on . Thus SSEj = SSTj – SSRj = SSTj, where SSTj = and is the coefficient of multiple determination of **x**j is regressed on  If **x**j is uncorrelated w -> = 0 and the variance of = /SSTj  If **x**j is correlated w -> the variance is inflated by a factor VIFj = (Variance inflation factor) | | | | | | |
| R code | library(car)  vif(object)  If vif value < 5 -> ok. If vif value > 5 -> have multicollinearity  Remove predictors w largest VIF from full fitted model (lm(y~x1+x2+x3, x=TRUE, data = data) one at time until all VIF < 5 | | vifChen = function(object) { #object must be lm object w argument x = TRUE)  X = object$x[,-1]; V = vcov(object)[-1,-1]; n = dim(X)[1]  sigma = summary(object)$sigma; v = diag(V); S = diag(var(X))\*(n-1)  vif = v\*S/sigma^2  return(vif)  } | | | | |
| Now largest VIF is < 5. So can proceed to model selection  Can use forward selection (excluding those predictors already removed).  Plot residual vs fitted value (check common variance); plot QQ plot of hat values (check for outliers w high leverage); QQplot of standardized/studentized residual (check for inconsistent points); QQplot of cook's distance (check for high influence)  If all ok -> model from forward selection is adequate. Predictors a,b,c... are significant as p-value < 0.05 | | | | | | |
| 2) Ridge regression | Since consequence of multicollinearity is XTX nearly singular, to remedy: add diagonal matrix to XTX, i.e. XTX + I (which is invertible)  Ridge regression estimator , where > 0 is a parameter to be chosen  is also the minimizer of the penalized sum of squares: | | | | | | |
| E = -> estimator is biased. Bias is  Let Q be the orthogonal matrix s.t. XTX = QQT, where = Diag(,...,). Thus QT  Let denote the jth component of QT. We have . Thus bias increase as increase | | | | | | |
| Variance matrix of = var() = = QT  Thus tr(var()) = . The variance decrease as increase | | | | | | |
| Sum of mean squares errors of is given by MSE = .  Need to get balance btw bias and variance by minimizing MSE  MSE cannot be readily used as a criterion as it involves unknowns and . Can select by Cross validation (CV)  Let be the ridge regression estimate of w parameter by deleting the ith observations. The CV score is given by CV() = , where is the ith row vector of the design matrix X  Best is the minimizer of CV() | | | | | | |
| R code | library(MASS)  regRidge = lm.ridge(y~x1+x2+x3+x4+x5, data = data, lambda = seq(0, 0.1, 0.005))  CV = regRidge$GCV; lambda = regRidge$lambda  plot(lambda, CV, type="l"); lambda.best = lambda[order(CV)[1]]  ridge.fit = lm.ridge(y~x1+x2+x3+x4+x5, data = data, lambda = lambda.best) | | | | | #create seq of lambda values to test  #Generalized CV; lambda values  #get best lambda | |
| Note ridge regression mainly used for building model for prediction. Cannot be used to assess importance or effects of predictor variables. The estimates from model cannot be used to construct CI or conduct hypo testing  If need to make inference on effects of predictors -> use strategy of removing predictors w large VIF | | | | | | |
| 3. Non-normality (remedy w variable transfor-mation) | If normal dist -> variance don't depend on mean (i.e. constant variance in regression models)  The violation of normality usually goes tgt w violation of constancy of variance  A variance stabilization transformation can help to rectify both discrepancy in normality and constancy of variance  If r/s btw variance and mean is known -> the desired variance stabilization transformation can be derived  To find the transformation -> use Box-Cox transformation | | | | | | |
| If variance depend on mean , i.e. = V(), a transformation can be found s.t. variance of transformed variable is approx indep of mean  Let h(Y) be a transformation. Use taylor series to expand h(Y) at , as h(Y) ≈ h() + h'()(Y-), where h() is a constant  Treating h() as mean of h(Y), var(h(Y)) = E[h(Y) - h()]2 ≈ [h'()]2E(Y - )2 = [h'()]2V()  WLOG, setting [h'()]2V() = 1, h'() = -> h() = , aka Variance Stabilization transformation | | | | | | |
| E.g. for proportion data | | Proportion response similar to binomial r.v.: variance depend on mean (np(1-p)) &  variance is smaller if mean is closer to either 0 or 1; larger if mean is closer to 0.5  So variance of proportion response can be approximated by V() = c(1-) -> h(Y) = = sin-1 | | | | | |
| y = c(1,2,3,4,5....); g = factor(c(rep(1,n1), rep(2,n2)); original.fit = lm(y~g); r.o = original.fit$resid; y.o = original.fit$fitted  y.t = asin(sqrt(y)); tra.fit = lm(y.t~g); r.t = tra.fit$resid; y.t = tra.fit$fitted  par(mfrow=c(2,1)); plot(y.o, r.o, main = "Residual plot of original fitting"); plot(y.t, r.t) #can see original plot has unequal variance  summary(org.fit); summary(tra.fit) #only use summary table of tranformed model for inferences | | | | | |
| Analysis can also be done using 2 sample t-test w test statistic, T = (n1 ≠ n2), where s =  Under normality assumption and H0: diff = 0: T~tn-a (a being num of grps) | | | | | |
| E.g. for count data | Count variable can be approximated by Poisson r.v. For Poisson dist, V() = . So variance is proportional to mean, i.e. V() = c (can also do quick check w sample data, s/sqrt(Y) for each group to check is more or less the same)  Variance stabilization transformation for count response Y is h(Y) = = | | | | | | |
| E.g. w h(Y) = | yy = sqrt(y); tran.fit = glm(yy~x1+x2+x3+x4+x5, data = data1)  In QQ plot of cook's distance, if some points don't follow smooth curve (don't need y = x) -> likely outliers  data2 = data1[-c(7,37,45)] #outliers; tran.fit2 = glm(yy~x1+x2+x3+x4+x5, data = data2) | | | | | | |
| After checking diagnostics again -> non constantcy of variance solve. Check VIF -> no serious multicollinearity  Can make inference now. | | | | | | |
| Box-Cox transfor-mation | If Variable transformation don't work?  For certain non-normal cts r.v., transformation cannot be determined solely by data type  Box-Cox transformation: h(Y) = . can be determined by data  When = 0, the Box-Cox transformation is given by h(Y) = ln(Y), since = ln(Y) | | | | | | |
| Let , where and are the SD and mean of ith treatment effect. The in Box-Cox transformation is determined by variance stabilization transformation   |  |  |  |  | | --- | --- | --- | --- | |  | (just use closest half int, e.g. = 0.04 -> use 0.05) | = 1 - | Transformation | |  | 3 | -2 | reciprocal squared | |  | 2 | -1 | reciprocal | |  | 3/2 | -1/2 | reciprocal sqrt | |  | 1 | 0 | log | |  | 1/2 | 1/2 | sqrt | | constant | 0 | 1 | no transformation | |  | -1/2 | 3/2 | 3/2 power | |  | -1 | 2 | square | | | | | | | |
| Determination of  Method 1: If observations are grouped, for each group, compute and . Fit regression model ln si = + ln + ei  If observations not group, fit ln |ri| = + ln + ei  For both, = estimate of  Method 2: Only for grouped observations. Select a few values, say , k = 1,...,K. For each k, compute Rk =  Select with smallest Rk | | | | | | |
| Direct determination of  For grouped observations: select a few values, say , k = 1,...,K  For each k, make the transformation yij . With the transformed data, compute , i = 1,...,g  Select s.t is closest to 1  For non-grouped observations: select a few values, say , k = 1,...,K  For each k, make the transformation yij . Analyze the regression models w as response variable.  Select s.t MSE() is smallest | | | | | | |
| Box-Cox e.g. find | fit.o = lm(y~x); r.o = fit.o$resid; y.o = fit.o$fitted; plot(y.o, r.o, main="Residual plot of original fit")  Method 1: try log(s)~log() OR log(|r.o|) ~ log(y.o). In both cases (estimate of ) ≈ 2, hence = -1  Method 2: alpha = c(-1,-0.5,0.5,1,1.5,2,3); R = NULL  for (i in 1:length(alpha)) {  R[i] = max(sd/ybar^alpha[i])/min(sd/ybar^alpha[i])}  Select with smallest Rk which is 2. Note if smallest R at endpoints -> need to consider more | | | | | | |
| yy = 1/y; fit.inv = lm(yy~x)  r.inv = fit.inv$resid; y.inv = fit.inv$fitted; plot(y.inv, r.inv, main = "Residual plot of transformed fit")  Since p-value of F-statistic is < 0.05, x is significant.  All factor are significantly diff from factor 1 as seen in summary table p-value  To test other pairwise compairson, need to extract covariance matrix of coefficients estimates | | | | | | |
| Box-Cox e.g. find | Computation of transformed s1/s2: s1 = NULL; s2 = NULL; lambda = c(-1,-0.5,0.001,0.5); #note lambda cannot be 0  for (i in 1:4) {  z1 = (y1^lambda[i]-1)/lambda[i]; z2 = (y2^lambda[i]-1)/lambda[i]  s1[i] = sqrt(var(z1)); s2[i] = sqrt(var(z2)) }  s.ratio = s1/s2. Select s.t ratio is closest to 1, which gives reciprocal transformation ( = -1) | | | | | | |
| y.t = 1/y; fit.o = lm(y~x); fit.t = lm(y.t~x)  summary(fit.o); summary(fit.t)  Difference is not significant w p-value > 0.05, based on transformed data | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Data w non-normal dist | | Binary response variable, Count response variable (but num of unique counts is small) | |
| MLR to GLIM | MLR assumptions:  1. Y follows normal dist  2. Covariates X­1,...,Xp affect dist of Y through a linear predictor:  3. The linear predictor is associated w mean of Y: where = EY | | Generalization of MLR (Generalized linear model/GLIM):  1. Distribution: Y follows a dist from exponential family  2. Linear predictor: Covariates affect dist of Y collectively through a linear predictor:  3. Link fn: The linear predictor is associated w dist of Y by a monotone link fn: g |
| Exponential family | Y follows an exponential family dist if its density fn has the form: , where is a monotone fn and is the dispersion param  Properties of exponential families: EY = . Var(Y) =  Since var(Y) > 0, is increasing and invertible, and hence | | |
| E.g. Benoulli: , where ,  Poisson: , where ,  Normal dist also part of exponential family | | |
| GLIM and its likelihood fn | GLIM: y . Since , , where h is the inverse fn of link fn g, and , eventually, which is a composite fn of  With data {(yi, **x**i), i = 1,...,n}, the log likelihood fn for , ignoring components free of **,** is given by | | |
| Estimates of is obtained by MLE. MLE is implemented by algo of Newton mtd with Fisher scoring: , where , . Iterate eqn until convergence  Above algo is same as iterated weighted least squares (IWLS) procedure which iterates: , where W is a diagonal matrix w wi = and **z** is a vector w zi = | | |
| By general properties of MLE, asymptotically, , where  For GLIM,  The asymptotic dist of is . Inference on is based on the asymptotic dist | | |
| For any component of , asymptotically, ~ N(0,1)  For testing H0: = 0, H1: ≠ 0, where is a subvector of , the test statistic is given by the Wald statistic (rather than a F-statstic): W = , which has an asymptotic -dist with df dim() | | |
| glm(formula, family = family(link), ...)  Some family(link) e.g.: binomial(logit), gaussian(identity), Gamma(inverse), inverse.gaussian(1/mu^2), poisson(log), quasi(link = "identity", variance = "constant"), quasibiniomial(logit), quasipoisson(log) | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |