|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IMPORTANT MODELS AND TESTS AND THERE APPLICATION** | | | | | | |
| **TEST OR MODEL NAME** | **APPLICATION** | **COMPONENTS** | **NULL HYPOTHESIS** | **HOW TO USE TEST STATISTIC OR P-VALUE** | **REMEDIAL MEASURE?** | **R CODE AND LIBRARY** |
| 1. **Simple linear regression** | * To check significant variables in the model. * To check the goodness of fit. | * RSE- How far is the fit from the points * F stat- Check if reg coef are non-zero. * AIC and SIC- select lowest valued model. | * Value of the **particular** coefficient is zero. * For F test- **all** explanatory variables have no impact. | * Smaller t-test= reject null OR p-val<0.05. * Greater the F value, rejecting Ho. | * Discard insignificant variables. | model=lm(y~x.,data=data1)  summary(model) |
| 1. **Durbin-Watson test** | * To test for autocorrelation between error terms | * DW ‘d’ statistic | * No autocorrelation exists | * If P-value<0.05, reject Ho | * If autocorrelation exists, use first difference or logarithmic change of the variable. | library(lmtest)  dwtest(model) |
| 1. **Breusch-Godfrey test** | * To test for autocorrelation between error terms |  | * No autocorrelation exists | * If P-value<0.05, reject Ho | * If autocorrelation exists, use first difference or logarithmic change of the variable. | library(lmtest)  bgtest(model) |
| 1. **VIF and tolerance** | * To detect multicollinearity |  |  | * Remove variables with VIF values between 5-10. * Consider the reciprocals | * Drop variables with high correlation and run regression on the new data. | library(car)  vif(model) |
| 1. **Breusch-pagan and White’s test** | * To detect Heteroscedasticity |  | * Error variance is homoscedastic | * If P-value<0.05, reject Ho | * Best way is to take the log of the dependent variable and carry out regression using the rest of the regressors. | Library(car)  bptest(model) |
| 1. **Ramsey Reset test and Lagrange’s Multiplier test** | * To check omission of relevant variables. | * Reset is called the F statistic * Chi-sq statistic | * Original model is correct | * Small statistic value, accept Ho OR p-value<0.05. | * Include variable if model fit is better otherwise discard or transform it. | Library(lmtest)  resettest(formula, power = 2:3, type = c("fitted", "regressor","princomp"), data = list()) |
| 1. **Jarque-Bera test** | * Normality test for errors | * X-sqrd and p-value | * Errors are normally distributed. | * If chi-sq stat > p-value then accept Ho OR p-value >0.05. | * Used for large samples, won’t affect. | Library(tseries)  jarque.bera.test(x) |
| 1. **Logit and Probit Models** | * Dichotomous and Binary variable regression models.   Eg: Gender, employed or unemployed etc. | * Coef-log of odds in favour of Y change by a unit change in X. * Coef \* normal density function, gives the probability * Pseudo and count R square | * Value of the **particular** coefficient is zero. | * Lower the values of null and residual deviance better the fit | * Probit is better since it has lower variance. But both can be used. | model=glm(yr~.,data=subdata,family = binomial(link = "logit"))  summary(model)  model2=glm(y~.+x\*x1=subdata,family = binomial(link = "probit"))  summary(model2) |
| 1. **Multinomial regression models** | * Polytomous or multiple category regression models   Eg: choice of car, choice of cereal etc. | * Chooser specific MLM: depend individual to individual. * Choice specific CLM: how features affect the choice of an individual. * Mixed | * Value of the **particular** coefficient is zero. | * Smaller t-test= reject null OR p-val<0.05. |  | test <- multinom(y~.,data =data1)  summary(test) |
| 1. **Ordinal regression models** | * Ordered data or ranked data   Eg: Likert type questionnaires. | * Compute odds ratio by exp(coefficient) | * Proportionality assumption, parallel reg lines. | * If chi-sq stat > p-value then accept Ho OR p-value >0.05. | * Use Maximum likelihood method | library(MASS)  library(ordinal)  fit=polr(y~x+x1,data=data) |
| 1. **Tobit and truncated models** | * Censored and truncated data. | * Coeff- direct effect of X on Y is inferred * LogLik- select model with maximum value | * Value of the **particular** coefficient is zero. | * Smaller t-test= reject null OR p-val<0.05. |  | library(survival)  fit=survreg(Surv(hours, hours>0, type='left') ~.,data=data, dist='gaussian')  summary(fit) |
| 1. **Overdispersion test** | * To check the equidispersion property PRM i.e mean=variance | * Dispersion * P-value | * The property of equidispersion holds | * Check dispersion, if non-zero. * P-value<0.05 reject Ho. | * Use Negative binomial regression model or quassi-poisson model | library(AER)  dispersiontest(model2, trafo = NULL, alternative = c("greater")) |
| 1. **Stationarity** | * For stationary time series. | * UR/ ADF test | * The given time series is not stationary | * Accept Ho if p-value>0.05 | * Use first differencing to convert the time series. | library(tseries)  adf.test(x) |
| 1. **Engle-Granger test** | * Testing cointegration or long run relationship. | * Test statistic * Tau | * The given time series is not stationary and there is no cointegration. | * Reject Ho if test statistic > tau at 5% LOS. | * Use Johansen test | library(urca)  ur=ur.df(res,type="none")  summary(ur) |
| 1. **Johansen test** | * Testing cointegration or long run relationship | * Eigen values * r: rank | * r<=1 * r=0 | * Reject Ho if test statistic > critical values at 5% LOS. * Rank is > than the one given in Ho. | * If no cointegration exists, regression is spurious. Either transform or discard the variable. | library(urca)  cointest=ca.jo(cmbdata,K=2,type = "eigen", ecdet = "const", spec = "transitory")  summary(cointest) |
| 1. **Box-Ljung test** | * To check presence of autocorrelation | * P-value | * No serial correlation/autocorrelation | * Accept if p-value>0.05 | * If autocorrelation exists, use first difference or logarithmic change of the variable. | Library(tseries)  box.test (x, lag = 1) |
| 1. **Granger Causality test** | * To check which variable precedes which. |  | * X causes Y and Y causes X | * P-value<0.05, reject Ho |  | Library(var)  causality(var,cause = "pdi\_dif")$Granger |
| 1. **Outliers test** | * To detect outliers | * Values of outliers |  |  | * Use robust regression | Library(car)  outlierTest(fit)  library(robustbase)  ltsReg(x1~. , data) |

<https://github.com/akshat3096>