**Krotov Egor**

**BIG DATA SYSTEMS 181**

Assignment 2 on Descriptive Statistics

1. *Assignment on Confidence Intervals and Hypothesis Testing*

For the assignment I took the iROBOT company stock prices from Yahoo Finance. Typical stock info set has many variables: volume, open&close prices, max&min of the day. To represent most valuable meaning, I selected the Close prices of stocks from 1st January of 2010 to present day. Data set of the chosen prices has 2207 observation.

*> library(quantmod)*

*> dataIRBT <- getSymbols("IRBT", src = "yahoo", from ="2010-01-01", auto.assign = FALSE)*

*> colMeans(is.na(dataIRBT))\*100*

*IRBT.Open IRBT.High IRBT.Low IRBT.Close IRBT.Volume IRBT.Adjusted*

*0 0 0 0 0 0*

*> data <- dataIRBT$IRBT.Close*

No NA in the Close Prices data and other columns.

1. Obtain a 97% confidence interval for the population mean.

*> t.test(data, conf.level=0.97)*

*One Sample t-test*

*data: data*

*t = 86.598, df = 2202, p-value < 2.2e-16*

*alternative hypothesis: true mean is not equal to 0*

*97 percent confidence interval:*

*38.84455 40.84276*

*sample estimates:*

*mean of x*

*39.84365*

The conclusion of this result is that with a probability of 97% iROBOT company stocks in a time of past 8 years was having price between 38.84455 - 40.84276USD.

(b) Perform a t-test on whether the population mean is equal to the sample median. Clearly state the null and alternative hypotheses provide the p-value.

*> summary(data)*

*Index IRBT.Close*

*Min. :2010-01-04 Min. : 14.52*

*1st Qu.:2012-03-10 1st Qu.: 25.91*

*Median :2014-05-20 Median : 33.15*

*Mean :2014-05-18 Mean : 39.84*

*3rd Qu.:2016-07-26 3rd Qu.: 40.82*

*Max. :2018-10-02 Max. :117.71*

Calculating the Median to perform the t.test. Which is 33.15 for this data set. Hypotheses for test will be:

H0: The population mean is equal to the sample median 33.15.

H1: The population mean is different of the sample median 33.15.

*> t.test(data, mu=33.15)*

*One Sample t-test*

*data: data*

*t = 14.548, df = 2202, p-value < 2.2e-16*

*alternative hypothesis: true mean is not equal to 33.15*

*95 percent confidence interval:*

*38.94138 40.74593*

*sample estimates:*

*mean of x*

*39.84365*

Gotten p-value is very small. It means that null hypotheses fully rejected.

(c) Obtain a 95% confidence interval for the population standard deviation.

*> df=length(data)-1*

*> varIRBT=var(data)*

*> lower=varIRBT\*df/qchisq(0.05/2,df,lower.tail=FALSE)*

*> upper=varIRBT\*df/qchisq(1-0.05/2,df,lower.tail=FALSE)*

*> c(lower=sqrt(lower),std.dev=sqrt(varIRBT),upper=sqrt(upper))*

*lower std.dev upper*

*20.97594 21.59528 22.25257*

Confidence interval of 95% for the population standard deviation is 20.97594 – 22.25257. The sample standard deviation is 21.59528.

(d) Find some dataset with a categorical variable. For that variable, compute the proportion of some level. Obtain a 99% confidence interval for that proportion.

I took epil$trt from MASS package. It has data about 112 placebo and 124 progabide drugs, with 236 observation in summary.

*> prop.test(112,236,conf.level=0.99)*

*1-sample proportions test with continuity correction*

*data: 112 out of 236, null probability 0.5*

*X-squared = 0.51271, df = 1, p-value = 0.474*

*alternative hypothesis: true p is not equal to 0.5*

*99 percent confidence interval:*

*0.3906510 0.5599267*

*sample estimates:*

*p*

*0.4745763*

With probability of 99%, the confidence interval for this proportion is 0.3906510 - 0.5599267.

(e) Perform a hypothesis test on whether the population proportion is equal to 1/2. Clearly state the null and alternative hypotheses provide the p-value.

Let us assume hypotheses:

H0: Half of the population is drugs, the other on is placebo.

H1: More than the half is drugs, or more that the half is placebo.

*> prop.test(x=112,n=236,p=0.5)*

*1-sample proportions test with continuity correction*

*data: 112 out of 236, null probability 0.5*

*X-squared = 0.51271, df = 1, p-value = 0.474*

*alternative hypothesis: true p is not equal to 0.5*

*95 percent confidence interval:*

*0.4097148 0.5402790*

*sample estimates:*

*p*

*0.4745763*

Here we can see that the p-value is 0.474 which more 0.05, that means we reject H0 and accept H1. Formally in this dataset bigger half is drugs, or placebo.

(f) Come up with some data for calculating the confidence intervals between proportions of two populations (in fact, you need just four numbers). Obtain a 99% confidence interval for the difference between proportions.

I took ObamaApproval data set from UsingR package.

H0: The confidence interval between proportions in the two years are the same

H1: The confidence interval between proportions in the two years are different

*> sum(ObamaApproval$approve[ObamaApproval$year==2013])*

*[1] 3360*

*> sum(ObamaApproval$disapprove[ObamaApproval$year==2013])*

*[1] 3277*

*> sum(ObamaApproval$approve[ObamaApproval$year==2010])*

*[1] 8200*

*> sum(ObamaApproval$disapprove[ObamaApproval$year==2010])*

*[1] 8416*

*> prop.test(x=c(3360,6637), n=c(8200,16616), conf.level=0.99)*

*2-sample test for equality of proportions with continuity correction*

*data: c(3360, 6637) out of c(8200, 16616)*

*X-squared = 2.3889, df = 1, p-value = 0.1222*

*alternative hypothesis: two.sided*

*99 percent confidence interval:*

*-0.006842106 0.027485741*

*sample estimates:*

*prop 1 prop 2*

*0.4097561 0.3994343*

Confidence interval 99% for two dataset are -0.006842106 to +0.548793323. This means that we are not sure if the confidence interval between proportions are equal because the 0 is also in the interval. The p-value is 0.1222 so we can’t reject the Null hypothesis, because the zero included in the interval.

(g)Perform an appropriate hypothesis test for the difference between proportions (perhaps, using imaginary data). Draw a conclusion.

I did not added any imaginary data for my dataset. The initial data is already good enough to demonstrate separation between years 2010 and 2013. Let’s assume hypotheses:

H0: The proportions in the two years are the same

H1: The proportions in the two years are different

*> prop.test(x=c(3360,6637), n=c(8200,16616), conf.level=0.95, alt="less")*

*2-sample test for equality of proportions with continuity correction*

*data: c(3360, 6637) out of c(8200, 16616)*

*X-squared = 2.3889, df = 1, p-value = 0.9389*

*alternative hypothesis: less*

*95 percent confidence interval:*

*-1.00000000 0.02131514*

*sample estimates:*

*prop 1 prop 2*

*0.4097561 0.3994343*

The 95% confidence interval is -1.0 to +0.02131514. By this range, we reveal that proportion are almost perfectly equal. Also p-value is 0.9389. This is strong argument to accept H0 hypothesis about equality of proportions.

1.2 (a) Perform the Jarque-Bera for normality. State clearly the null and alternative hypothesis.

For the assignment I took three company stock prices from Yahoo Finance – SWKS, SPLK, NXPI. To represent most valuable meaning, I selected the Close prices of stocks from 1st January of 2010 to present day. Hypotheses to be tested:

H0: The distribution of the stocks prices is normal distribution

H1: The distribution of stocks pries is not normal distribution

*> x= getSymbols("SPLK", src = "yahoo", from ="2010-01-01", auto.assign = FALSE)*

*> y= getSymbols("NXPI", src = "yahoo", from ="2010-01-01", auto.assign = FALSE)*

*> z= getSymbols("SWKS", src = "yahoo", from ="2010-01-01", auto.assign = FALSE)*

*> z=as.numeric(z$SWKS.Close)*

*> x=as.numeric(x$SPLK.Close)*

*> y=as.numeric(y$NXPI.Close)*

*> xpi=x[2:length(x)]*

*> xpmi=x[1:length(x)-1]*

*> ypi=y[2:length(y)]*

*> ypmi=y[1:length(y)-1]*

*> zpi=z[2:length(z)]*

*> zpmi=z[1:length(z)-1]*

*> log\_y=log(ypi)-log(ypmi)*

*> log\_z=log(zpi)-log(zpmi)*

*> log\_x=log(xpi)-log(xpmi)*

*> jarque.bera.test(log\_x)*

*Jarque Bera Test*

*data: log\_x*

*X-squared = 8612.9, df = 2, p-value < 2.2e-16*

*> jarque.bera.test(log\_y)*

*Jarque Bera Test*

*data: log\_y*

*X-squared = 4326.1, df = 2, p-value < 2.2e-16*

*> jarque.bera.test(log\_z)*

*Jarque Bera Test*

*data: log\_z*

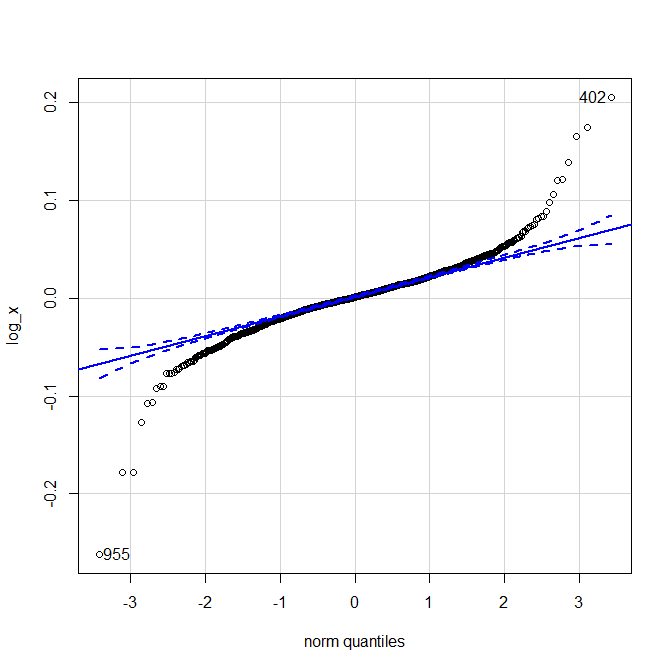
*X-squared = 3347, df = 2, p-value < 2.2e-16*

For all chosen datasets, by the p-value, we can reject H1 hypothesis, and conclude that that is not normal distribution.

(b) Check whether the (univariate) empirical distribution of log returns for each stock is normal by examining the QQ-plot. Use the command qq.plot() from car package instead of the built-in function. Discuss whether the observations are within the confidence interval.

> qqPlot(log\_x)

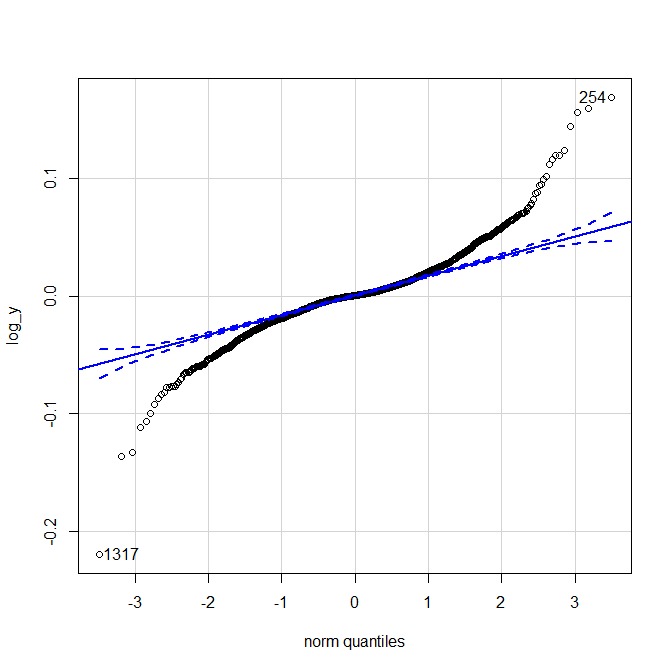
[1] 955 402



Empirical distribution of log returns for stock is are out of the confidence interval, except the range of -1 to +1.5.

*> qqPlot(log\_y)*

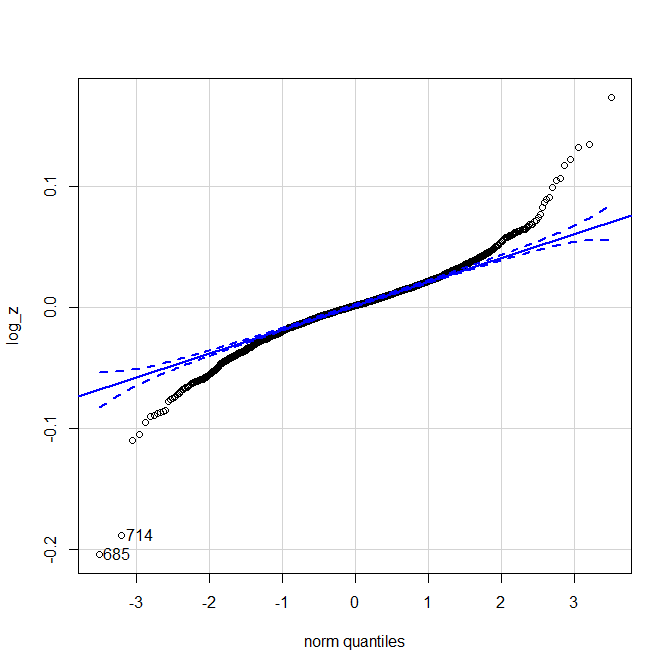
*[1] 1317 254*



Empirical distribution of log returns for stock is are out of the confidence interval, except the range of -1 to +1.

*> qqPlot(log\_z)*

*[1] 685 714*



Empirical distribution of log returns for stock is are out of the confidence interval, except the range of -1 to +1.5.

1.3 Use a built-in set from 2 to perform the x^2-test for homogeneity (uniform distribution). Describe the data and discuss the result.

For this task I’m using reddrum dataset of usingR. The hypotheses are:

H0: Uniform distribution of data are homogeny   
H1: Uniform distribution of data are not homogeny

*> chisq.test(reddrum$length, p=rep(1/length(reddrum$length),length(reddrum$length)))*

*Chi-squared test for given probabilities*

*data: reddrum$length*

*X-squared = 55.778, df = 99, p-value = 0.9999*

For no mistake with that absolute 0.9999 p-value,

1.4 Get a two-way contingency table from sources 3. Conduct a x^2-test for association (independence) between the variables.

|  |  |  |
| --- | --- | --- |
|  | *Короткие волосы* | *Длинные волосы* |
| *Молодые девушки* | *57* | *225* |
| *Пожилые женщины* | *371* | *70* |

*> hair\_age=rbind(c(57,225),c(371,70))*

*> chisq.test(hair\_age)*

*Pearson's Chi-squared test with Yates' continuity correction*

*data: hair\_age*

*X-squared = 288.27, df = 1, p-value < 2.2e-16*

The results are pretty obvious: young women prefer to have long hair, and with the ages they are starting to prefer short hair. P-value for this test is close to 0. It means that variables are not independent.

1. *Assignment on Regression and Classification*

1. Simple regression.

I took UsingR smokyph. Description: Water pH levels at 75 water samples in the Great Smoky Mountains.

*> data(package="UsingR")*

*> x=smokyph$waterph*

*> y=smokyph$elev*

*(a)* Build a simple regression model (command lm). Provide the estimates of the model's parameters. Draw the scatter plot and the regression line.

*> lm(y~x)*

*Call:*

*lm(formula = y ~ x)*

*Coefficients:*

*(Intercept) x*

*2.1336 -0.2132*

*> summary(lm(y~x))*

*Call:*

*lm(formula = y ~ x)*

*Residuals:*

*Min 1Q Median 3Q Max*

*-0.32398 -0.15061 -0.02379 0.14187 0.56742*

*Coefficients:*

*Estimate Std. Error t value Pr(>|t|)*

*(Intercept) 2.13365 0.36304 5.877 1.16e-07 \*\*\**

*x -0.21323 0.05075 -4.201 7.41e-05 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*Residual standard error: 0.1918 on 73 degrees of freedom*

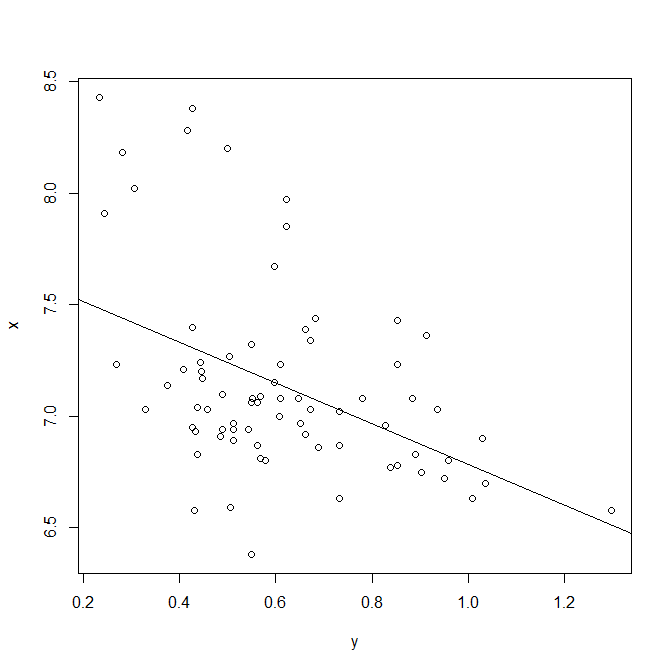
*Multiple R-squared: 0.1947, Adjusted R-squared: 0.1837*

*F-statistic: 17.65 on 1 and 73 DF, p-value: 7.413e-05*

*> plot(y~x)*

*> res=lm(y~x)*

*> abline(res)*



Scatter plot shows that data not laying on the regression line.

The lm command in R resulted into the following simple linear regression line: y(elevlation) = 2.1336 - 0.2132\*x(waterph)

(b) Analyze the summary statistics (command summary()) focusing on:

i. The t-test for the slope. ii. The F-test. iii. R^2 coefficient.

The p-values 1.16e-07 and 7.413e-05 is very low that indicates the rejection of H0 (intercept and slope are equal to 0).

The F-statistic p-value: 7.413e-05 means the same as t-test that the H0 (slope is equal to 0) can be rejected in favor of H1 (slope is not equal to 0).

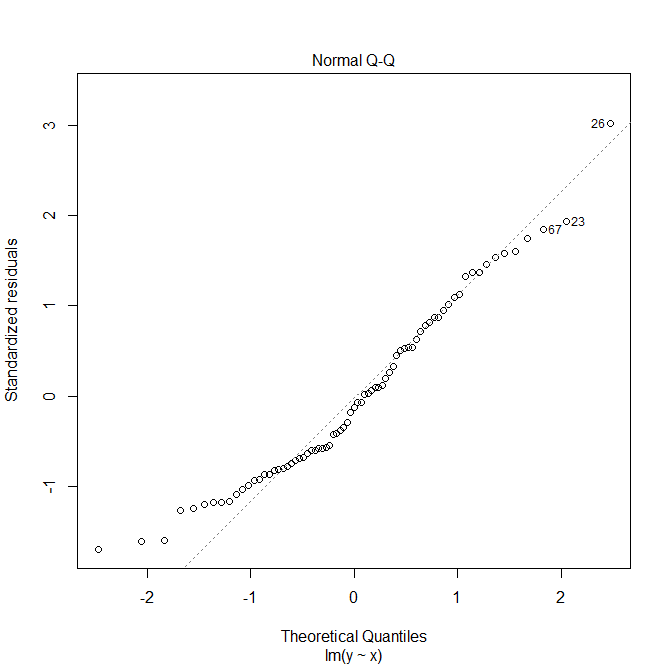
R-squared: 0.1837 that means that the data points are not lying on the regression line.

(c) Plot the residuals against fitted values and comment on the model's adequacy. Examine the qq-plot for the residuals.

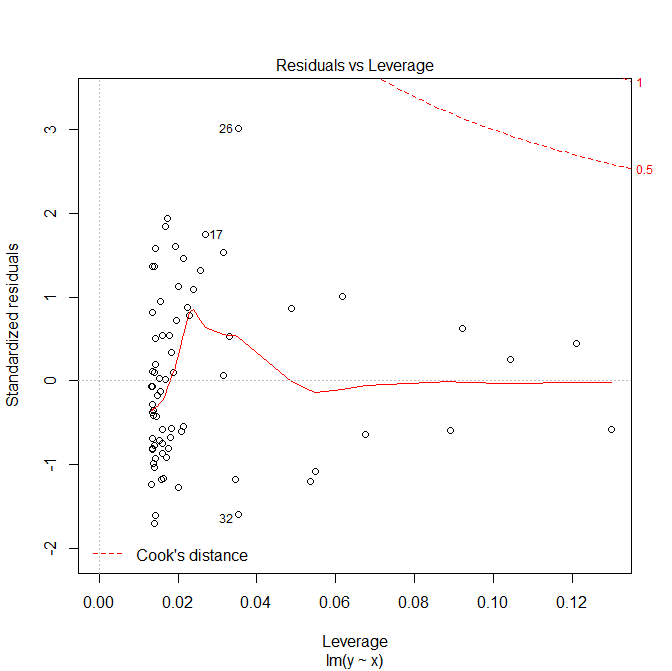
*> plot(res)*



It can be concluded that the residuals (distances from actual data points to regression line) are big.

**

From the qq-plot we can say that the data is not normally distributed.



(d) Make predictions for several new values of the independent variable. For each predicted value, compute and plot the confidence intervals for the mean and single value.

Based on type of my data I did prediction for variable x:

*> values = seq(6, 9, 0.2)*

*> predict(da, new=data.frame(x = values))*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Value | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | 8 | 9 | 10 | 11 |
| Prediction | 0. *8542* | | 0. *8116* | | 0. *7689* | | 0. *7263* | | 0. *6836* | | 0. *6410* | | 0. *5983* | 0. *5557* | 0. *5130* | 0. *4704* | 0. *4277* |
| Value | | 12 | | 13 | | 14 | | 15 | | 16 | |
| Prediction | | 0. *3851* | | 0. *3424* | | 0. *2998* | | 0. *2572* | | 0. *2145* | |

*> meanConfInt = predict(res, new=data.frame(x = values), int="conf")*

*> observConfInt = predict(res, new=data.frame(x = values), int="predict")*

*> plot(x,y)*

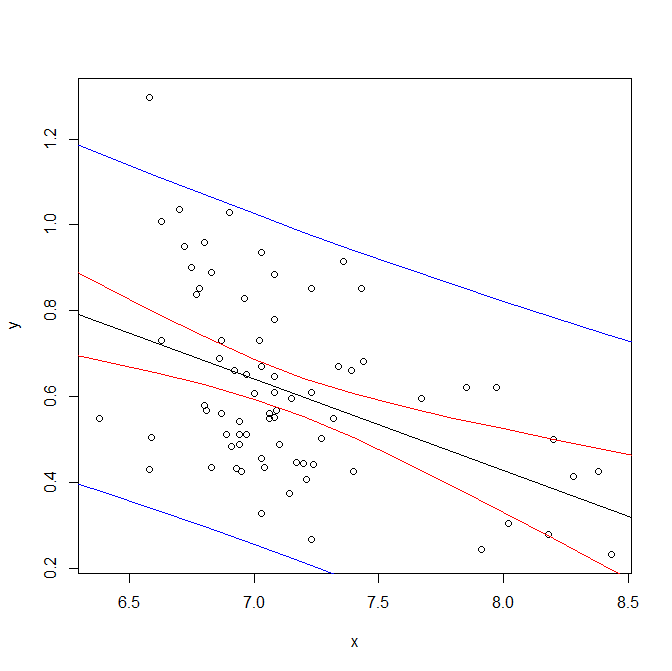
*> abline(res)*

*> lines(values, observConfInt[,2], col='blue')*

*> lines(values, observConfInt[,3], col='blue')*

*> lines(values, meanConfInt[,2], col = 'red')*

*> lines(values, meanConfInt[,3], col = 'red')*



Plot describes the area between two blue lines shows with 95% confidence possible observations for the values between 6 and 9 with step of 0.2. Area between two red lines indicate 95% confidence interval for the mean of possible values, which is potential regression lines. One of 75 original observation is out of range of lines, the rest lays inside the lines confidently.

1. Multivariate regression.

I took ISwR heart.rate. Description: German on fragments of glass collected in forensic work.

1. Choose the response and explanatory variables.

*> data = fgl*

*> x1=data$Na*

*> x2=data$Mg*

*> x3=data$Al*

*> x4=data$Si*

*> y=data$RI*

Variables «Na» as x1, «Mg» as x2, «Al» as x3, «Si» as x4 will be used for multivariate regression model for predicting refractive index.

1. Build a multivariate linear model (command lm). Provide the estimates of the model's parameters.

*> ML=lm(y~x1+x2+x3+x4)*

lm used to derive the multivariate linear model.

1. Analyze the summary statistics (command summary()).

*> summary(ML)*

*Call:*

*lm(formula = y ~ x1 + x2 + x3 + x4)*

*Residuals:*

*Min 1Q Median 3Q Max*

*-8.4997 -0.4776 0.0925 0.6624 4.4874*

*Coefficients:*

*Estimate Std. Error t value Pr(>|t|)*

*(Intercept) 217.32738 9.99353 21.747 <2e-16 \*\*\**

*x1 -1.15087 0.12724 -9.045 <2e-16 \*\*\**

*x2 -1.35308 0.08253 -16.395 <2e-16 \*\*\**

*x3 -4.08851 0.22784 -17.944 <2e-16 \*\*\**

*x4 -2.64266 0.13142 -20.108 <2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*Residual standard error: 1.447 on 209 degrees of freedom*

*Multiple R-squared: 0.7771, Adjusted R-squared: 0.7728*

*F-statistic: 182.2 on 4 and 209 DF, p-value: < 2.2e-16*

*> stepAIC(ML)*

*Start: AIC=163.21*

*y ~ x1 + x2 + x3 + x4*

*Df Sum of Sq RSS AIC*

*<none> 437.87 163.21*

*- x1 1 171.41 609.28 231.91*

*- x2 1 563.15 1001.02 338.16*

*- x3 1 674.62 1112.49 360.75*

*- x4 1 847.12 1284.99 391.60*

*Call:*

*lm(formula = y ~ x1 + x2 + x3 + x4)*

*Coefficients:*

*(Intercept) x1 x2 x3 x4*

*217.327 -1.151 -1.353 -4.089 -2.643*

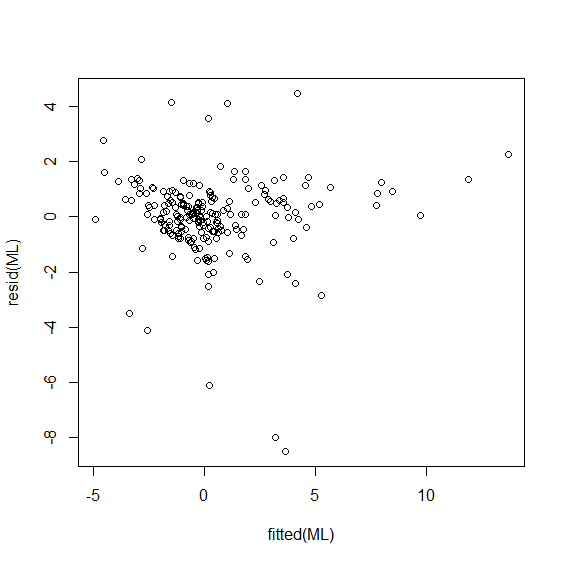
The summary of the model indicated very low p-values for all intercept and the four variables (slopes) that indicate that none of the parameters are equal to 0 (H0 hypothesis rejected in favor to H1).

F-test’s p-value is < 2.2e-16. Therefore we can reject H0 (x1=x2=x3=x4=0) in favor to H1 (at least one of x1, x2, x3, x4 are not = 0)

Both Multiple R-squared: 0.7771 and Adjusted R-squared: 0.7728 are close to 1 that indicate good quality of the model.

1. Plot the residuals against \_fitted values and comment on the model's adequacy.

*> plot(fitted(ML), resid(ML))*



Apart from some outliers the residual values lie between -3 and 2 that are on an very acceptable level.

1. Play with your model by adding or removing the explanatory variables. Alternatively, add a non-linear term(s) to your model: Choose the best one by the partial F-test criterion (command anova). Choose the best one by the AIC criterion (command stepAIC). For each model, watch the value of the adjusted R2.

*> ML1=lm(y~x2+x3+x4)*

*> anova(ML,ML1)*

*Analysis of Variance Table*

*Model 1: y ~ x1 + x2 + x3 + x4*

*Model 2: y ~ x2 + x3 + x4*

*Res.Df RSS Df Sum of Sq F Pr(>F)*

*1 209 437.87*

*2 210 609.28 -1 -171.41 81.815 < 2.2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*> stepAIC(ML1)*

*Start: AIC=231.91*

*y ~ x2 + x3 + x4*

*Df Sum of Sq RSS AIC*

*<none> 609.28 231.91*

*- x2 1 447.20 1056.48 347.70*

*- x3 1 686.97 1296.25 391.47*

*- x4 1 769.65 1378.93 404.70*

*Call:*

*lm(formula = y ~ x2 + x3 + x4)*

*Coefficients:*

*(Intercept) x2 x3 x4*

*191.159 -1.168 -4.125 -2.501*

*> ML2=lm(y~x1+x4)*

*> anova(ML, ML2)*

*Analysis of Variance Table*

*Model 1: y ~ x1 + x2 + x3 + x4*

*Model 2: y ~ x1 + x4*

*Res.Df RSS Df Sum of Sq F Pr(>F)*

*1 209 437.87*

*2 211 1283.04 -2 -845.17 201.71 < 2.2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*> stepAIC(ML2)*

*Start: AIC=389.28*

*y ~ x1 + x4*

*Df Sum of Sq RSS AIC*

*<none> 1283.0 389.28*

*- x1 1 104.18 1387.2 403.98*

*- x4 1 609.03 1892.1 470.40*

*Call:*

*lm(formula = y ~ x1 + x4)*

*Coefficients:*

*(Intercept) x1 x4*

*170.8714 -0.8585 -2.1885*

*> summary(ML2)*

*Call:*

*lm(formula = y ~ x1 + x4)*

*Residuals:*

*Min 1Q Median 3Q Max*

*-10.2894 -1.3909 -0.2099 1.0819 9.1621*

*Coefficients:*

*Estimate Std. Error t value Pr(>|t|)*

*(Intercept) 170.8714 16.3196 10.470 < 2e-16 \*\*\**

*x1 -0.8585 0.2074 -4.139 5.04e-05 \*\*\**

*x4 -2.1885 0.2187 -10.008 < 2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*Residual standard error: 2.466 on 211 degrees of freedom*

*Multiple R-squared: 0.3469, Adjusted R-squared: 0.3407*

*F-statistic: 56.03 on 2 and 211 DF, p-value: < 2.2e-16*

*> ML3=lm(y~x1+x2+x4)*

*> anova(ML, ML3)*

*Analysis of Variance Table*

*Model 1: y ~ x1 + x2 + x3 + x4*

*Model 2: y ~ x1 + x2 + x4*

*Res.Df RSS Df Sum of Sq F Pr(>F)*

*1 209 437.87*

*2 210 1112.49 -1 -674.62 322 < 2.2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*> stepAIC(ML3)*

*Start: AIC=360.75*

*y ~ x1 + x2 + x4*

*Df Sum of Sq RSS AIC*

*<none> 1112.5 360.75*

*- x2 1 170.55 1283.0 389.28*

*- x1 1 183.76 1296.2 391.47*

*- x4 1 714.70 1827.2 464.94*

*Call:*

*lm(formula = y ~ x1 + x2 + x4)*

*Coefficients:*

*(Intercept) x1 x2 x4*

*193.6364 -1.1914 -0.6573 -2.4161*

*> summary(ML3)*

*Call:*

*lm(formula = y ~ x1 + x2 + x4)*

*Residuals:*

*Min 1Q Median 3Q Max*

*-12.677 -1.005 -0.114 0.885 7.054*

*Coefficients:*

*Estimate Std. Error t value Pr(>|t|)*

*(Intercept) 193.6364 15.7520 12.293 < 2e-16 \*\*\**

*x1 -1.1914 0.2023 -5.890 1.52e-08 \*\*\**

*x2 -0.6573 0.1159 -5.674 4.59e-08 \*\*\**

*x4 -2.4161 0.2080 -11.615 < 2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*Residual standard error: 2.302 on 210 degrees of freedom*

*Multiple R-squared: 0.4337, Adjusted R-squared: 0.4256*

*F-statistic: 53.6 on 3 and 210 DF, p-value: < 2.2e-16*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Pr(>F) compared to ML | AIC | Adjusted R2 |
| ML=lm(y~x1+x2+x3+x4) |  | 163.21 | 0.7728 |
| ML1=lm(y~x2+x3+x4) | 2.2e-16 | 231.91 | 0.6854 |
| ML2=lm(y~x1+x4) | 2.2e-16 | 389.28 | 0.3407 |
| ML3=lm(y~x1+x2+x4) | 2.2e-16 | 360.75 | 0.4256 |

We can see that none of the independent variables are likely to be equal to 0 due to low Pr(>F) values. According to lowest AIC value the best model is the initial one with all four explanatory variables. Adjusted R2 is the biggest for the first model, hence the best one according to this criteria.

1. Logistic regression.

I took mmr\_levee.dat data set from users.stat.ufl.edu. Description: Factors Relating to Levee Failures on the Middle Mississippi River.

*> river <- read.table("C:/Users/EGOR/Downloads/mmr\_levee.dat", header = FALSE)*

1. Build a logistic regression model (command glm). Comment on the significance of the coefficients.

*> river.fail=river$V1*

*> river.width=river$V7*

*> river.mile=river$V3*

*> river.flway=riverV8*

*> river.sin=river$V12*

*> river.log=glm(river.fail~river.mile+river.width+river.flway+river.sin,family=binomial)*

*> summary(river.log)*

*Call:*

*glm(formula = river.fail ~ river.mile + river.width + river.flway +*

*river.sin, family = binomial)*

*Deviance Residuals:*

*Min 1Q Median 3Q Max*

*-1.72134 -1.12970 -0.00934 1.13423 1.66943*

*Coefficients:*

*Estimate Std. Error z value Pr(>|z|)*

*(Intercept) -0.2731078 1.4838181 -0.184 0.854*

*river.mile 0.0012426 0.0047382 0.262 0.793*

*river.width -0.0010011 0.0006440 -1.554 0.120*

*river.flway 0.0001076 0.0002074 0.519 0.604*

*river.sin 0.7152453 0.8342745 0.857 0.391*

*(Dispersion parameter for binomial family taken to be 1)*

*Null deviance: 97.041 on 69 degrees of freedom*

*Residual deviance: 93.722 on 65 degrees of freedom*

*AIC: 103.72*

*Number of Fisher Scoring iterations: 4*

After building a logistic regression model with four independent variables (Fail of dam, Mileage, Width, Flows and Turns) in order to classify dam fail column, we get from the summary of the model.

We can see that river.mile, river.sin, river.flway correlated with good value, the river.width correlated with less good value.

1. Use stepAIC command to select the best model.

*> stepAIC(river.log)*

*Start: AIC=103.72*

*river.fail ~ river.mile + river.width + river.flway + river.sin*

*Df Deviance AIC*

*- river.mile 1 93.791 101.79*

*- river.flway 1 93.994 101.99*

*- river.sin 1 94.492 102.49*

*<none> 93.722 103.72*

*- river.width 1 96.294 104.29*

*Step: AIC=101.79*

*river.fail ~ river.width + river.flway + river.sin*

*Df Deviance AIC*

*- river.flway 1 94.019 100.02*

*- river.sin 1 94.522 100.52*

*<none> 93.791 101.79*

*- river.width 1 96.309 102.31*

*Step: AIC=100.02*

*river.fail ~ river.width + river.sin*

*Df Deviance AIC*

*- river.sin 1 94.692 98.692*

*<none> 94.019 100.019*

*- river.width 1 96.388 100.388*

*Step: AIC=98.69*

*river.fail ~ river.width*

*Df Deviance AIC*

*<none> 94.692 98.692*

*- river.width 1 97.041 99.041*

*Call: glm(formula = river.fail ~ river.width, family = binomial)*

*Coefficients:*

*(Intercept) river.width*

*0.8538451 -0.0008459*

*Degrees of Freedom: 69 Total (i.e. Null); 68 Residual*

*Null Deviance: 97.04*

*Residual Deviance: 94.69 AIC: 98.69*

We observe model with lowest AIC 98.69 and optimal model with considering AIC will be

formula = river.fail ~ river.width

1. Make a prediction based on the entire dataset. State the threshold of acceptance. Compare the forecast with the actual observations.

*> f=predict(river.log, type = "response")*

*> dif=abs(river.fail-f)*

*> length(dif[dif>0.5])*

*[1] 27*

Threshold of acceptance will be 0,5 and above.

After making the forecast on the entire dataset and comparing with the actual values 27 out of 70 correct predictions were made. Overall accuracy of the model is (1853/2287)\*100 = 38% which is low.

1. Divide the entire set into training and test subsets. Rebuild the model using only the training subset. Make predictions for the test subset.

*> river.test.log=glm(V1~V3+V7+V8+V12, data=river.test, family=binomial)*

*> tr.index = sample(1:nrow(river.test), nrow(river.test)\*0.8)*

*> trSet = river.test[tr.index, ]*

*> testSet = river.test[-tr.index, ]*

*> river.test2 = glm(V1 ~ V3+V7 + V8 + V12, trSet, family = binomial)*

*> fitted\_results\_test = predict(river.test2, newdata= testSet, type = "response")*

*> fitted\_results\_test = ifelse(fitted\_results\_test > 0.5,1,0)*

*> fitted\_results\_test*

*5 10 12 17 19 23 24 26 35 36 44 49 50 68*

*0 1 0 0 1 0 0 0 0 0 0 1 1 1*

After rebuilding the model on the training subset and making forecasts on the test subset, only nine prediction was made. Accuracy rate on the test set is (5/14)\*100=35% that indicates “badness” of the model.

1. Discriminant analysis.
2. Conduct the linear discriminant analysis (command lda, package MASS) using training and test subsets. Compare the forecast with the actual observations. Comment on the results.

*> river.log2*

*Call:*

*lda(V1 ~ V3 + V7 + V8 + V12, data = trSet)*

*Prior probabilities of groups:*

*0 1*

*0.5357143 0.4642857*

*Group means:*

*V3 V7 V8 V12*

*0 93.21733 1099.7563 2819.556 1.181857*

*1 107.62692 901.4454 2738.352 1.195404*

*Coefficients of linear discriminants:*

*LD1*

*V3 0.0110944609*

*V7 -0.0022052227*

*V8 0.0004095159*

*V12 0.9575115893*

*> river.log2=lda(V1 ~ V3+V7 + V8 + V12,testSet)*

*> river.log2p = predict(river.log2, trSet)$class*

*> river.log2p*

*[1] 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1*

*[54] 1 0 0*

*Levels: 0 1*

The LDA output is 0.535 and 0.464, it means that 53,5% of the training observations

correspond presence of Mississippi river dam fails.

1. Conduct the quadratic discriminant analysis (command qda). Comment.

*> river.qda=qda(V1 ~ V3+V7 + V8 + V12,trSet)*

*> predict(river.qda,testSet)*

*$`class`*

*[1] 0 1 0 0 0 0 0 0 0 0 1 0 0 0*

*Levels: 0 1*

*$posterior*

*0 1*

*5 0.695364439 0.30463556*

*10 0.006403732 0.99359627*

*12 0.513002794 0.48699721*

*17 0.809582860 0.19041714*

*19 0.903176176 0.09682382*

*23 0.644381096 0.35561890*

*24 0.653566392 0.34643361*

*26 0.624896925 0.37510308*

*35 0.567593412 0.43240659*

*36 0.736870520 0.26312948*

*44 0.353587882 0.64641212*

*49 0.646816203 0.35318380*

*50 0.627204108 0.37279589*

*68 0.559865932 0.44013407*

The quadratic discriminant regression model showed slightly result by misclassifying 3 observations. Accuracy = 78%

1. The KNN classifier..
2. Conduct the KNN classification (command knn(), package class) using training and test subsets. Compare the forecast with the actual observations. Comment on the results.
3. Play with the number of nearest neighbors K.

*> train=river[1:49,c("V1","V3","V7","V8","V12")]*

*> test=river[50:70,c("V1","V3","V7","V8","V12")]*

*> res=river[1:49,c("V1")]*

*> knn(train,test,res,10)*

*[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1*

*Levels: 0 1*

*> knn(train,test,res,5)*

*[1] 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1*

*Levels: 0 1*

*> knn(train,test,res,3)*

*[1] 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1*

*Levels: 0 1*

*> knn(train,test,res,1)*

*[1] 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1*

*Levels: 0 1*

The best result is with the number of neighbors considered: 1

Accuracy 80%.

1. Compare the quality of classification obtained by algorithms 3-5 for the test subset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistic | Linear Discriminant | Quadratic Discriminant | KNN |
| Accuracy % | 33 | 53,5 | 78 | 80 |

Judging solely from the accuracy level, we could conclude that the KNN and Quadratic discriminant models are the best ones. However, since the data availability was limited (only 49 observations for training subset) it is hard to conclude whether either of those models would behave with similar accuracy on actual data. Therefore, it would make sense to train the models and compare them based on larger datasets.