# R Shortcuts for Statistics

Useful commands

| | |
|---|---|
| which() | filter data frame or factor by creating a filter list |
| table(df) | number of entries for each value in dataframe |
| prop.table | table in terms of probabilities |
| str(df) | Type of variables present at df |
| mean(factor/df) | means of dataframe columns or factor |
| sd(factor/df) | standard deviation of dataframe columns or factor |
| paste0 | concatenate strings |
| Levels | return levels of attributes |
| | |
| | |

Tests

| Name | Use Case | Comments |
|---|---|---|
| acf() | | |
| dwtest() | | |
| shapiro.test() | Test if data is normally distributed | |
| t.test() | | |
| chisq() | | |
| varTest() | | |
| cor(df[,c(4,1:3)],method="pearson") | | Numeric Insights Parametric version for normal-like |
| cor(df[,c(4,1:3)],method="spearman") | | Numeric Insights Non Parametric |
| cor.test(df$prestige,df$income,method="pearson") | Test Hypothesis to test whether my population rho equals 0 or not | Inferential |
| cor.test(df$prestige,df$income,method="spearman",data=df) | Test Hypothesis to test whether my population rho equals 0 or not | |

| Question | Solution |
|---|---|
| Is serial correlation present? | acf(df$cal)<br>library(lmtest)<br>dwtest(df$cal~1) |
| Determine univariant severe outliers. | See Excel |
| Multivariate Outliers: | See Excel, then: |

| | | | |
|---|---|---|---|
| | abline( h=res.mout$cutoff, lwd=2, col="red")<br>abline( v=res.mout$cutoff, lwd=2, col="red")<br><br>llmout <- which( ( res.mout$md > res.mout$cutoff ) & (res.mout$rd > res.mout$cutoff) );llmout<br>df[llmout,]<br>res.mout$md[llmout]<br>df$mout <- 0<br>df$mout[ llmout ] <- 1<br>df$mout <- factor( df$mout, labels = c("MvOut.No","MvOut.Yes")) | | |
| Using EDA which are the most associated variables with the numeric response variable? Use also FactoMineR profiling tools at 99% significance level | See Excel for FactoMineR<br><br>EDA:<br>plot(df[,c(TARGET_VAR,EXPLAIN_VARS)])<br>cor(df[,c(TARGET_VAR, EXPLAIN_VARS)], method="spearman")<br>corrplot(cor(df[,c(9,3:8)], method="spearman"), is.corr=T) | | |
| determine the most relevant global associations at 99% CI for categorial Target var. | See Excel | | |
| Say Something about distribution that was assumed. Graphical and inferential | graphical<br>hist(df$cal,30)<br>hist(log(df$cal),30) | Inferential<br>shapiro.test( log(df$cal) ) | |
| Num_var variate dispersion behavior according to the categ_var. numeric, graphics and inferential | Graphical<br>Boxplot( num_var~cat_var, data = df ) | Inferential<br>See Excel for correct test (29-33) | Numerical<br>tapply( df$cal, df$brand, sd ) |
| Num_var variate mean behavior according to the cat_var. numeric, graphics and inferential | Boxplot( cal~brand, data = df ) | See Excel for correct test (23-26) | tapply( df$cal, df$brand, mean ) |
| which brands show a remarkable difference in mean behavior among them. Use one-sided tests | pairwise.wilcox.test( df$cal, df$brand, alternative="less" )<br>pairwise.wilcox.test( df$cal, df$brand, alternative="greater" ) | | |
| test at the 1% level the null hypothesis that the population standard deviation is not larger than 0.15cal against the alternative that it is | tapply(df$cal,df$brand,sd)<br>table(df$brand)<br>ss <- 0.16362880<br># H0: sigma^2= 0.15^2  H1: sigma > 0.15^2 Normal population  (n-1)ss^2/sigma^2 ~ X2(n-1)<br># (n-1)ss^2/sigma^2<br><br>chi<-(29-1)*(ss^2)/(0.15^2);chi<br>1-pchisq(chi,28) # pvalue > 0.01 H0 can not be rejected<br><br>Or | | |

| | |
|---|---|
| | varTest(df[which(df$brand == "A"),]$cal, alternative="greater", conf.level =0.99, sigma.squared = 0.15^2) |
| 99% upper threshold for the number of calories for brand A population variance. Normal distribution for calories is assumed to hold. | varTest(x, sigma.squared=0.15^2, alternative="less",conf.level=0.99)<br>sqrt(variance) to obtain standard deviation |
| Build a 99% confidence interval for the difference in the mean of 100 g calories between brands A and C. Assume that equal variances in the population calories per brand does not hold | ll <- which( df$brand %in% c("A","C"))<br>dff <- df[ll,]<br><br>t.test(dff$cal~dff$brand, conf.level=0.99) – used to create confidence interval<br>fligner.test(dff$cal,dff$brand, conf.level=0.99) – used to check if variances are the same<br>t.test(dff$cal~dff$brand, conf.level=0.99, var.equal = T) |
| Out of 100 people, 60 prefer A to C. Determine a 99% confidence interval for the population proportion that favors A in front of C. Test the null hypothesis that selecting A and C has equal probability. | prop.test(60, n=100, p=0.5, conf.level=0.99, correct=F) |
| Determine a 99% confidence interval for the difference in the population proportion that favors A in front of C accounting the two surveys. Test the null hypothesis that selecting A brand has a lower probability in the second of the surveys | prop.test(c(60,110), n=c(100,200), conf.level=0.99, correct=F, alternative="greater") |