Categorical Variable and Dummy Coding

Code

Hide

```
library(car)
```

```
Warning message:
In read.dcf(file.path(p, "DESCRIPTION"), c("Package", "Version")) :
  cannot open compressed file '/home/atrides/R/x86_64-pc-linux-gnu-library/
4.0/tinytex/DESCRIPTION', probable reason 'No such file or directory'
```

Hide

```
library(QuantPsyc)
library(boot)
library(dplyr) # data mainpulation
library(cowplot)
library(DAAG)
library(ggplot2)
```

Hide

```
df<- read.delim('/home/atrides/Desktop/R/statistics_with_R/07_Regression/Dat
a_Files/GlastonburyFestivalRegression.dat', header=TRUE)
df<- na.omit(df)
head(df)
```

	ticknumb	music	day1	day2	day3	change
	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	2111	Metaller	2.65	1.35	1.61	-1.04
2	2229	Crusty	0.97	1.41	0.29	-0.68
10	2504	No Musical Affiliation	1.11	0.44	0.55	-0.56
12	2510	Crusty	0.82	0.20	0.47	-0.35
14	2515	No Musical Affiliation	1.76	1.64	1.58	-0.18
21	2549	Crusty	2.17	0.70	0.76	-1.41

6 rows

Hide

```
is.factor(df$music)
```

```
[1] FALSE
```

Hide

```
df$music<- as.factor(df$music)

is.factor(df$music)
```

```
[1] TRUE
```

Hide

```
contrasts(df$music)<- contr.treatment(4,base=4)

attr(df$music, "contrasts")
```

```
                      1 2 3
Crusty                1 0 0
Indie Kid             0 1 0
Metaller              0 0 1
No Musical Affiliation 0 0 0
```

Hide

```
# setting contrasts manually, preferable

crustyVSnma<- c(1,0,0,0)
indieVSnma<-  c(0,1,0,0)
metalVSnma<-  c(0,0,1,0)

contrasts(df$music)<- cbind(crustyVSnma, indieVSnma, metalVSnma)

attr(df$music, 'contrasts')
```

```
                      crustyVSnma indieVSnma metalVSnma
Crusty                          1          0          0
Indie Kid                       0          1          0
Metaller                        0          0          1
No Musical Affiliation          0          0          0
```

Hide

```
# Doing Regression
fest<- lm(change~music, data=df)
summary(fest)
```

```
Call:
lm(formula = change ~ music, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.82569 -0.50489  0.05593  0.42430  1.59431

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.55431    0.09036  -6.134 1.15e-08 ***
musiccrustyVSnma -0.41152    0.16703  -2.464   0.0152 *
musicindieVSnma  -0.40998    0.20492  -2.001   0.0477 *
musicmetalVSnma   0.02838    0.16033   0.177   0.8598
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6882 on 119 degrees of freedom
Multiple R-squared:  0.07617,   Adjusted R-squared:  0.05288
F-statistic:  3.27 on 3 and 119 DF,  p-value: 0.02369
```

Hide

```
# what does these coeffiecients represent
# this actually represents the difference in the change in hygiene scores if
a person has no musical affiliation,
# compared to someone who is a crusty, indie, metal, Rescpectively


# see https://stackoverflow.com/questions/3505701/grouping-functions-tapply-
by-aggregate-and-the-apply-family
# note: tapply - For when you want to apply a function to subsets of a vecto
r and the subsets
# are defined by some other vector, usually a factor.
```

Hide

```
print(round(tapply(df$change, df$music, mean, na.rm=TRUE), 3))
```

```
               Crusty              Indie Kid               Metaller No Musi
cal Affiliation
               -0.966                 -0.964                 -0.526
-0.554
```

Hide

```
cat("AIC_model: ", AIC(fest),"\nBIC_model: ",  BIC(fest))
```

```
AIC_model:  263.0618
BIC_model:  277.1228
```

Hide

```
anova(fest)
```

```
Analysis of Variance Table

Response: change
           Df Sum Sq Mean Sq F value  Pr(>F)
music       3  4.646 1.54882  3.2704 0.02369 *
Residuals 119 56.358 0.47359
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
# outliers and influential cases

df$residuals <-resid(fest)
df$standarized.residuals <- rstandard(fest)
df$studentized.residuals <- rstudent(fest)
df$cooks <- cooks.distance(fest)
df$dfbeta <- dfbeta(fest)
df$dffit <- dffits(fest)
df$leverage <- hatvalues(fest)
df$covratio <- covratio(fest)
df$fitted.values<- fitted.values(fest)


large_resid <- dplyr::filter(df, standarized.residuals>2 | standarized.resid
uals< -2)
large_resid
```

Hide

```
# now lets see cooks distance , leverage , covariance ratio for 'these' case
s
k = 3 #number of predictors
n = 123 #number of objervations

average_leverage = (k+1)/n
average_leverage
```

```
[1] 0.03252033
```

Hide

```
cvr_low<- 1-3*average_leverage
cvr_high<- 1+3*average_leverage

large_resid$cov_ideal_low <- cvr_low
large_resid$cov_ideal_high <- cvr_high

large_resid
```

Hide

```
# no serious influencers or outliers
```

Hide

```
# Assumptions Check
car::durbinWatsonTest(fest)  # assumption of independent errors
```

```
 lag Autocorrelation D-W Statistic p-value
   1      0.04948997      1.893407   0.552
 Alternative hypothesis: rho != 0
```

Hide

```
# here car::vif(fest) will not be used
# Not with vif() in the car package, which wants to compute generalized vari
ance inflation factors (GVIFs)
# for multi-df terms in the model. Single-df VIFs are pretty simple, so you
could just write your own function.
# Alternatively, there are other packages on CRAN, such as DAAG, that comput
e VIFs, so you might try one of these.
DAAG::vif(fest)     # assumptions of multicollinearity
```

```
musiccrustyVSnma  musicindieVSnma  musicmetalVSnma
          1.1379           1.1001           1.1438
```

Hide

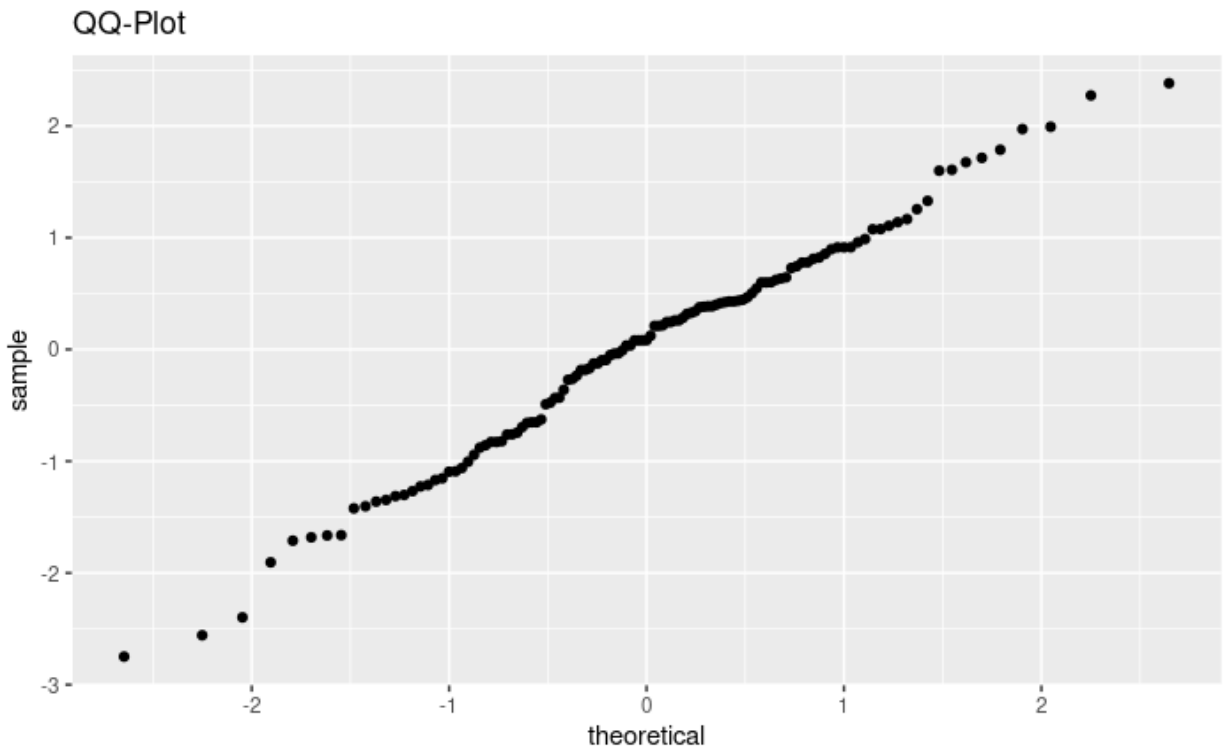
```
# https://r.789695.n4.nabble.com/Variance-Inflation-Factor-VIC-with-a-matrix
-td4643734.html
```

Hide

```
df$fitted.values<- fitted.values(fest)
df$std_fitted.values<- (fitted.values(fest)-mean(fitted.values(fest)))/sd(fi
tted.values(fest))
resid_plot<- ggplot(df, aes(standarized.residuals,std_fitted.values))
resid_plot<- resid_plot+geom_point()+geom_smooth(formula='y~x',method = "l
m",alpha=0.1)



resid_qq<- ggplot(df, aes(sample=studentized.residuals))
resid_qq<- resid_qq+stat_qq()+ggtitle('QQ-Plot')
resid_qq
```

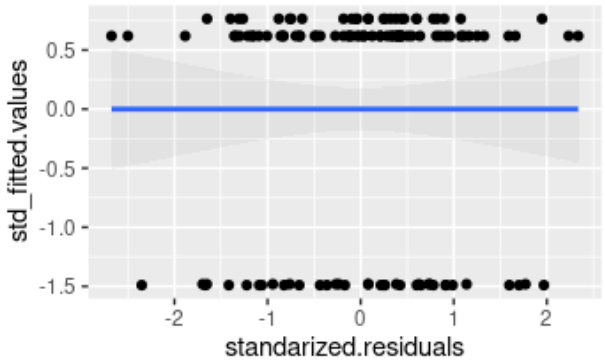


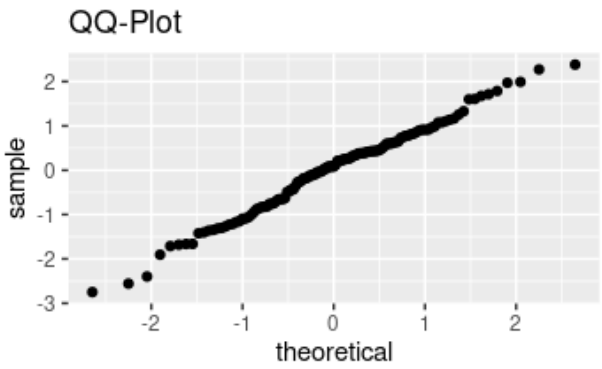
Hide

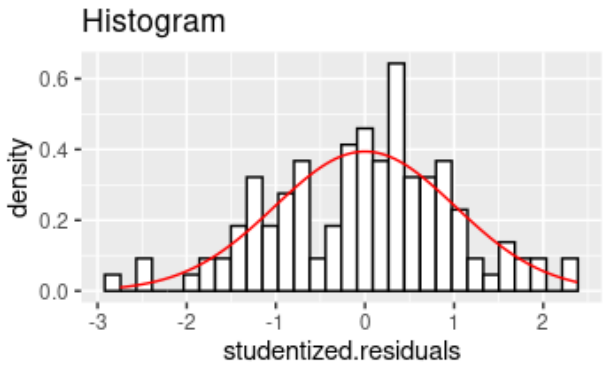
```
histresid<- ggplot(df, aes(studentized.residuals))
histresid<- histresid+geom_histogram(aes(y=..density..),colour='black', fill
='white')+
  ggtitle('Histogram')+
  stat_function(fun = dnorm, args = list(mean=0, sd=sd(df$studentized.residu
als, na.rm = TRUE)), colour='red')


plot_grid(resid_plot, resid_qq, histresid,ncol=2, nrow=2 )
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```







Hide

```
# this assumption was also met
```