Regression Diagnostics

DATA 712
City University of New York, Queens College
Joseph N. Cohen
October 1, 2019

CONTENTS

Regression Diagnostics: Introduction 2
Data & Models Used in this Module 2
Data 2
Getting Started 3
Review: Implementing and Interpreting Models 4
Diagnosing Models 6
Test #1: Error Distribution 7
Explained 7
Testing 7
Redresses 9
Test #2: Heteroscedasticity 9
Explained 9
Testing 9
Redresses 11
Test #3. Collinearity 11
Explained 11
Testing 11
Rederesses 12
Outliers 12
Explained 12
Testing 13
Rederesses 14
Linearity 15
Explained 15
Testing 15
Redresses 16
Redresses 16
Variable Transformation 16
Interpreting Models with Logged Variables 18
Model Respecification 19

REGRESSION DIAGNOSTICS: INTRODUCTION

OLS regression relies on several assumptions. If these assumptions are violated, then the model's results can be distorted.

Before accepting the results of an OLS model, we perform *diagnostics*, which ensure that these assumptions hold. If these assumptions do not hold, then we have reason to doubt the model's results.

Remember: A regression isn't done until its diagnostics have been passed. If your diagnostics fail, your model may need adjustments.

Diagnostics Considered in this Module

We will consider five assumptions that should be considered when diagnosing an OLS model:

- Normally-Distributed, Mean Zero Errors
- Homoskedasticity
- No Collinearity
- No Outliers
- Linearity

DATA & MODELS USED IN THIS MODULE

Data

In this example, we will work with data that probes the relationship between family structure, labor markets, and government policies, using data from the Organization for Economic Cooperation and Development (OECD). Data is from 2010. It is stored as an Excel spreadsheet called **"OECD Family Data.xlsx"**

For your convenience, I've printed a codebook below:

Variable	Description
cname	Country Name
ccode	Country Code
electricity	Access to electricity (% of population) [EG.ELC.ACCS.ZS]
teen.birth	Adolescent fertility rate (births per 1,000 women ages 15-19) [SP.ADO.TFRT]
hs.dropout	Adolescents out of school (% of lower secondary school age) [SE.SEC.UNER.LO.ZS]
hiv.young	Adults (ages 15+) and children (ages 0-14) newly infected with HIV [SH.HIV.INCD.TL]
age.dep	Age dependency ratio (% of working-age population) [SP.POP.DPND]
fertile.land	Arable land (% of land area) [AG.LND.ARBL.ZS]
military.people	Armed forces personnel (% of total labor force) [MS.MIL.TOTL.TF.ZS]
birth.rate	Birth rate, crude (per 1,000 people) [SP.DYN.CBRT.IN]
injury.deaths	Cause of death, by injury (% of total) [SH.DTH.INJR.ZS]
primary.dropout	Children out of school (% of primary school age) [SE.PRM.UNER.ZS]
death.rate	Death rate, crude (per 1,000 people) [SP.DYN.CDRT.IN]
gdp.pc	GDP per capita, PPP (constant 2011 international $) [NY.GDP.PCAP.PP.KD]
gov.exp	Expense (% of GDP) [GC.XPN.TOTL.GD.ZS]
mat.mortality	Maternal mortality ratio (modeled estimate, per 100,000 live births) [SH.STA.MMRT]
military.exp	Military expenditure (% of GDP) [MS.MIL.XPND.GD.ZS]
open.toilet	People practicing open defecation (% of population) [SH.STA.ODFC.ZS]
doctors	Physicians (per 1,000 people) [SH.MED.PHYS.ZS]
pop.density	Population density (people per sq. km of land area) [EN.POP.DNST]
poverty.extreme	Poverty gap at $1.90 a day (2011 PPP) (%) [SI.POV.GAPS]
poverty.severe	Poverty gap at $3.20 a day (2011 PPP) (%) [SI.POV.LMIC.GP]
poverty.strong	Poverty gap at $5.50 a day (2011 PPP) (%) [SI.POV.UMIC.GP]
hiv.total	Prevalence of HIV, total (% of population ages 15-49) [SH.DYN.AIDS.ZS]
starvation	Prevalence of undernourishment (% of population) [SN.ITK.DEFC.ZS]
internet	Secure Internet servers (per 1 million people) [IT.NET.SECR.P6]
smoking	Smoking prevalence, total (ages 15+) [SH.PRV.SMOK]
suicide	Suicide mortality rate (per 100,000 population) [SH.STA.SUIC.P5]

Getting Started

Let's get the session started. We load the data into memory, assigning it to object **dat**:

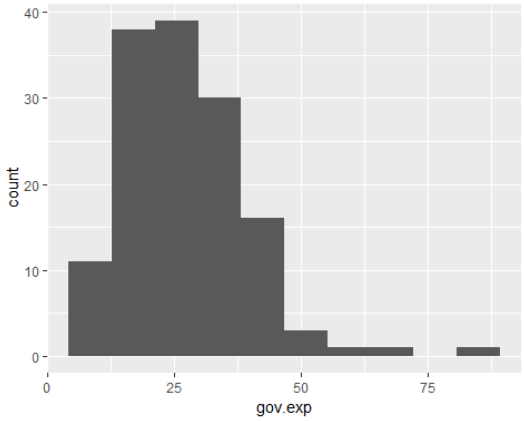
```
rm(list=ls())
gc()
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 509131 27.2    1150084 61.5   609151 32.6
## Vcells 988395  7.6    8388608 64.0  1605885 12.3
directory <- "C:/Users/jncohen/Dropbox/Teaching/DATA 712/Week 6"
setwd(directory)

library(readxl)
dat <- read_xlsx("WDI 2015 Extract.xlsx", sheet = 1)

head(dat)
## # A tibble: 6 x 28
##   cname ccode electricity teen.birth hs.dropout hiv.young age.dep
##   <chr> <chr>       <dbl>      <dbl>      <dbl>     <dbl>   <dbl>
## 1 Afgh~ AFG          71.5       73.1      NA           NA    88.8
## 2 Alba~ ALB         100         20.7       2.03       100    44.0
## 3 Alge~ DZA          99.9       10.7      NA         1200    52.7
## 4 Amer~ ASM          NA         NA        NA           NA    NA
## 5 Ando~ AND         100         NA        NA           NA    NA
## 6 Ango~ AGO          42        157.       NA        26000    97.6
## # ... with 21 more variables: fertile.land <dbl>, military.people <dbl>,
## #   birth.rate <dbl>, injury.deaths <dbl>, primary.dropout <dbl>,
## #   death.rate <dbl>, gdp.pc <dbl>, gov.exp <dbl>, mat.mortality <dbl>,
## #   military.exp <dbl>, open.toilet <dbl>, doctors <dbl>,
## #   pop.density <dbl>, poverty.extreme <dbl>, poverty.severe <dbl>,
## #   poverty.strong <dbl>, hiv.total <dbl>, starvation <dbl>,
## #   internet <dbl>, smoking <dbl>, suicide <dbl>
```

REVIEW: IMPLEMENTING AND INTERPRETING MODELS

Recall from our last lesson that one can implement an OLS model using the **lm()** function. Here is a brief walk through of the process.

Remember that we use OLS with continuous outcomes. In this exercise, we will model differences in government expenditures (% GDP). This measure captures the size of government spending relative to the overall size of the economy. A higher number means that the government is larger, relative to the economy. Here is the distribution of our data set's 140 reported values:

```
library(ggplot2)
ggplot(dat, aes(x = gov.exp)) + geom_histogram(bins = 10) +
  scale_y_continuous(breaks=seq(0,90,10))
```



Let's work with a model that tries to predict the size of a government's public sector using (1) the per capita GDP (a metric often used to capture a country's overall level of economic development), and (2) whether or not that country is "highly militarized" (a ratio of military to labor force size greater than 3%), "moderately militarized" (a ratio between 1% and 3%), or "non-militarized" (below 1%).[1] I create the second variable using the function **ifelse()**:

```
dat$militarized <- ifelse(dat$military.people > 3, 2,
                          ifelse(dat$military.people > 1, 1, 0))
table(dat$militarized)
##
##   0   1   2
## 102  49  14
```

[1] These are arbitrary thresholds. This variable is being used in this analysis to illustrate how to interpret a discrete variable.

And then run the regression:

```
reg.1 <- lm(gov.exp ~ gdp.pc + factor(dat$militarized), data = dat)
summary(reg.1)
##
## Call:
## lm(formula = gov.exp ~ gdp.pc + factor(dat$militarized), data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.936  -7.058  -0.662   7.232  21.265
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 20.6163324  1.4620607   14.10   <2e-16 ***
## gdp.pc                       0.0002221  0.0000469    4.74    6e-06 ***
## factor(dat$militarized)1     4.4953439  1.9277133    2.33    0.021 *
## factor(dat$militarized)2    -1.5357448  3.7770022   -0.41    0.685
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.57 on 119 degrees of freedom
##   (94 observations deleted due to missingness)
## Multiple R-squared:  0.182,  Adjusted R-squared:  0.162
```

Remember that there are four basic descriptors of a regression model's results:

- *Coefficient sign.* Whether our model finds a positive or negative relationship between a predictor and an outcome.
- *Coefficient size.* Our model's predicted change in an outcome variable for each one-unit increase in our predictor.
- *Coefficient significance.* Our model's estimate of the probability that we could get an effect this extreme if the true effect was zero. We conventionally treat $p < 0.05$ as minimal (though not necessarily rock-solid) evidence that an effect is "significant".
- *Model fit.* Which you can understand as the proportion of total variation in the outcome variable that is "captured" by differences in our units' predictor scores.

Here, the model predicts that a country will have 0.206 ratio of government expenditures to GDP, plus 0.0002 percentage points for every additional dollar of per capita GDP. So a country like the United States, with a per capita GDP of about $53 thousand is expected to have a government expenditure to GDP ratio of 30%, whereas a poorer country like Sudan (per capita GDP $4,262) is predicted to have a ratio of 20%.

While the model registers a minimally significant coefficient for the intermediate category of our "militarized" coefficient, I interpret these findings as suggesting that it is overall not important. First off, these categories were arbitrary, so we didn't have a good theoretical reason for expecting them to be significant. Second, the combination of the fact that this variable is only marginally significant while the "highly militarized" category is so clearly insignificant makes me suspect that this is a situation in which people can get "fooled by randomness". Modeling is a probabilistic

enterprise, and one of the hazards is that variables can randomly register as significantly related to each other, even if they're not. One remedy to this problem is to withhold belief in a predictor until its predictive power has been established in multiple environments, like with different data or different models.

The R-Squared on this model is about 0.18. It is nothing extraordinary, but you see models of this caliber in the social sciences literature.

So the conclusion that we might draw here is that this test supports the idea that countries with higher GDPs have higher government spending levels.

DIAGNOSING MODELS

The next step in the process is diagnosing the model. This means that we (1) run diagnostic checks that ensure our model satisfies some important assumptions, and (2) adjust our data or models to satisfy these assumptions, if possible.

Note: The examples presented in this note are a bit more drawn out than necessary. By this, I mean that I calculate diagnostic variables and generate plots using more basic codes, without capitalizing on packages that do a lot of this automatically. My reason for doing this is to offer students a more explicit look at what these diagnostic statistics and graphs mean.

That being said there are two very useful commands that can help you diagnose your OLS regressions with more expediency. The first is **ls.fit()**, which is part of R's base package. This command allows you to create an object with the diagnostic variables that are formulated and used below:

```
diags.1 <- ls.diag(reg.1)
names(diags.1)
##  [1] "std.dev"      "hat"          "std.res"      "stud.res"
##  [5] "cooks"        "dfits"        "correlation"  "std.err"
##  [9] "cov.scaled"   "cov.unscaled"
```

The second is the **plot()** function, which can be used with an OLS model to create many of the diagnostic plots that we'll examine below:

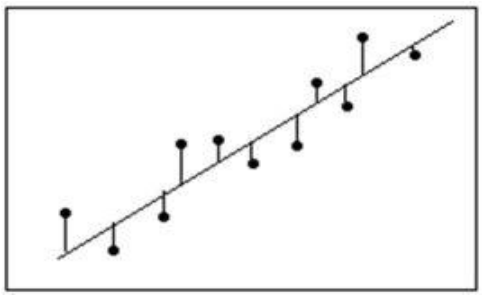
```
plot(reg.1)
```

The output has been omitted for brevity's sake, but try it on your own computer.

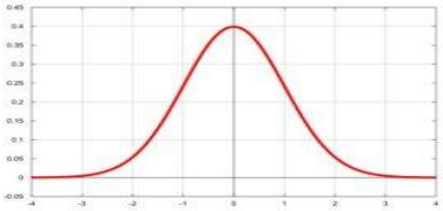
Test #1: Error Distribution

Explained

Recall from the previous module that *errors* our observations' deviation from the regression line. They represent our model's failure to predict our observations perfectly.



The first assumption underpinning our OLS model is that our observations have a normal distribution with a mean of zero. Such a distribution would look something like this:



Another way of understanding this assumption is that:

1. On average, the model doesn't systematically over- or under-estimate in making predictions
2. The model is more typically off by smaller, rather than larger, magnitudes

If this assumption is violated, our standard error and significance estimates will be distorted.

Testing

To test this assumption, we need to produce a new variable denoting each observation's error in Model 1. We will be working with these observations' **studentized residuals**,which are the residuals divided by estimates of their standard deviation. This is done to standardize our residual estimates, such that differing error variability across observations does not affect our estimates. Studentized residuals can be generated using the **stures()** command in the ***MASS*** package.

```
library(MASS)
resid.1 <- studres(reg.1)
```

Zero-Mean Assumption: We can check the zero-mean assumption using **summary()**:

```
summary(resid.1)
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.715000 -0.754000  0.006537 -0.007000  0.927900  1.660000
```

Looks OK here.

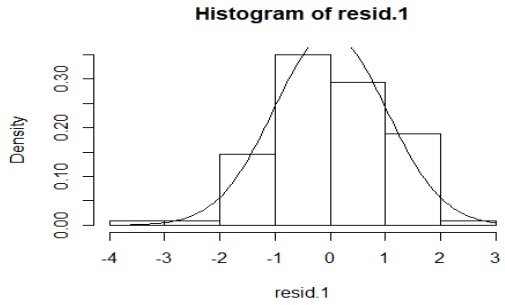
Normality Assumption via Histogram: We can eyeball the normality assumption by looking at a historgram of the residuals:

```
hist(resid.1, freq = FALSE)
```

The histogram looks *sort of* normal-looking, with lots of observations in the middle and fewer moving outward. Still, this is all very impressionistic.

You can overlay a normal curve on this histogram as follows:

```
mean.r1 <- mean(resid.1)
sd.r1 <- sd(resid.1)
hist(resid.1, freq = FALSE)
curve(dnorm(x, mean=mean.r1, sd=sd.r1), add=TRUE)
```

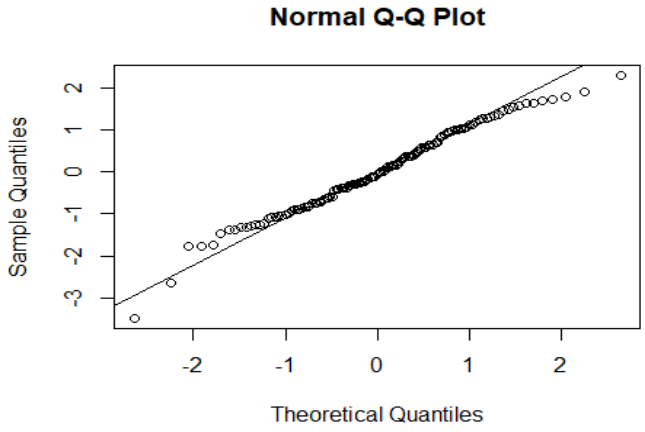


Not bad, though the tails look a big.

QQplot: Alternatively, we can plot the residuals on what is known as a QQplot. A QQplot has a diagonal line that denotes where residual values would lie if they were perfectly normally distributed. The dots represent where the actual residuals lie. The distribution of the residuals deviates from a normal curve to the extent that the dots deviate from the diagonal line in the figure.

The syntax is:

```
qqnorm(resid.1)
qqline(resid.1)
```



The plot suggests that the distribution's more extreme values are larger than would be expected in a perfectly normal distribution.

Shapiro-Wilk Normality Test: One alternative to these graph-based tests is the Shapiro-Wilks normality test. It test the null hypothesis that the residuals are distributed normally. The command is implemented as **shapiro-test()** in the base package:

```
##
##  Shapiro-Wilk normality test
##
## data:  resid.1
## W = 0.99, p-value = 0.2
```

A significant test means we have a problem. In this case, the test does not reject the null. It does not provide strong evidence that the errors depart from our assumption of being Normally distributed. I would not put a lot of weight into these tests, but want you to know what they are in case someone mentions them to you.

Redresses

If the zero-mean it, it normally-distributed assumption is violated, then we might consider the following rederesses:

- **Transforming Variables.** Rescaling outcome or predictor variables. Useful for variables with serious skews.
- **Model Re-Specification:** Adding or subtracting predictors
- **Robust Standard Errors:** Model that accounts for uncertainty in error distribution

Each of these rederesses is reviewed below.

Test #2: Heteroscedasticity

Explained

The second assumption underpinning OLS models is that error variance is constant across all predicted values. Another way of explaining this is to say that the model should have the same predictive accuracy for high, medium, and low dependent variable scores. The model should it be spot-on for one set of values parenthesis say, middle range values), but all over the map for others (like more extreme dependent variable scores).

If error variance is in constant across predicted values, then the model suffers from the problem of **heteroscedasticity**, or non-constant error variance.

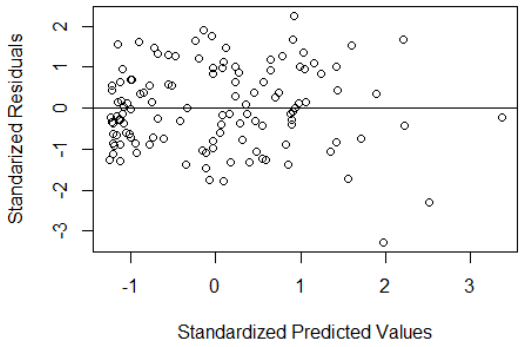
A model with heteroskedastic errors will have distorted significance estimates.

Testing

As with our assumptions about error distribution, heteroscedasticity can be tested by graphical and numeric means.In the graphical test, we plot our standardized residuals against standardized predicted values. In other words, we create a graph that sees whether our error terms vary with the model's predictions.

```
#Calculate predicted values
p.1 <- predict(reg.1)

#Standardized predicted values
std.p.1 <- (p.1 - mean(p.1))/sd(p.1)
```

```r
#Calculate residuals
r.1 <- resid(reg.1)

#Standardize Residuals
std.r.1 <- (r.1 - mean(r.1))/sd(r.1)

# Plot the two as a scatterplot with an additional like along y (residuals) =
 0
plot(std.p.1,std.r.1,xlab="Standardized Predicted Values",ylab="Standarized R
esiduals")
abline(0,0)
```



When you look at these graphs, you are looking for a solid band of variance around the Y = 0 line. You don't want the spread of error values to be different at different predicted values.

Again, we are dealing with a small sample in this example. It kind of looks like the errors are evenly spread out across predicted values. With this test, problems are more obvious when you have huge spreads of errors on extreme values when more moderate predicted values have low errors, or vice-versa.

Breusch-Pagan Test. An alternative, numeric test is the test. It is implemented as the command **bptest()** in the ***lmtest*** package. This is a chi-squared test of the null hypothesis that error variance is constant across predicted values. A very low p-value (say below 0.05) implies that the errors are heteroskedastic, and thus the models assumptions are violated.

```r
library(lmtest)
bptest(reg.1)
##
##  studentized Breusch-Pagan test
##
## data:  reg.1
## BP = 14, df = 3, p-value = 0.003
```

Here, the test p-value is well below our 0.05 (or whatever), and so we assume the OLS model's underlying assumptions are violated with respect to homoscedasticity. In other words, a significant test means we have a problem.

Redresses

Our redresses are similar to those in the previous section:

- Re-specify Model
- Transform Variables
- Robust Standard Errors

Test #3. Collinearity

Explained

Variables are "collinear" when their affect on the dependent variable is highly correlated. This is generally a threat when the predictors themselves are highly correlated. The problem with collinearity is that the OLS model cannot discern the independent effects of these two or more related variables well. Their effects are so similar that the model has difficulty finding sufficient incidences in which one of fact took hold without the other, making it hard to discern independent effects empirically.

The problem with collinearity is that it can produce distorted estimates. Sometimes, collinear variables appear as large, highly-significant effects. In other words, overly-similar predictors can create an illusion of a strong, powerful effect where none exists.

Testing

One way to test for collinearity is by calculating the Variance Inflation Factor. This can be done using the **vif()**command from the ***usdm*** package.

The **vif()** command needs are matrix of the model's predictors. An easy way to get this matrix is to create an object using the "model" sub object that was produced when creating your linear model. We'll need a model with multiple predictors:

```
reg.2 <- lm(gov.exp ~ gdp.pc + military.exp  + age.dep, data = dat)
names(reg.2)
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "na.action"     "xlevels"       "call"          "terms"
## [13] "model"
head(reg.2$model)
##    gov.exp gdp.pc military.exp age.dep
## 1    37.02   1767       0.9935   88.77
## 2    24.45  10971       1.1623   44.02
## 6    20.82   6645       3.1054   97.58
## 8    25.01  19244       0.8501   56.55
## 9    25.08   8172       4.2392   44.41
## 11   26.74  44054       1.9585   51.09
```

We only want the predictors, so we're going to create a new object of model predictors by using this "model" sub-object and eliminating the first column, which had dependent variabel values:

```
pred.2 <- reg.2$model[-1,]
```Then, we can produce VIF values for the model's predictors:

```
library(usdm)
## Loading required package: sp
## Loading required package: raster
##
## Attaching package: 'raster'
## The following objects are masked from 'package:MASS':
##
##     area, select
vifs.2 <- vif(pred.2)
vifs.2
##      Variables   VIF
## 1      gov.exp 1.340
## 2       gdp.pc 1.469
## 3 military.exp 1.014
## 4      age.dep 1.344
```

Conventionally, variables are treated as potentially-collinear if their VIFs approach or exceed 4. If a VIF is greater than 10, it is probably a serious problem.

Here, the test suggests that collinearity is probably not a problem.

Rederesses

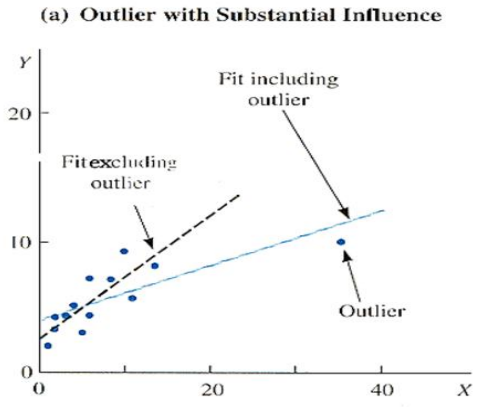
Collinearity is a problem that's more difficult to resolve. Some potential rederesses:

- Consider excluding one of the collinear predictors. Check the effects of using one predictor versus the others.
- Consider consolidating collinear predictors into an index. Collinear variables may be touching upon a broader concept that you can operationalize using multiple indicators.

Outliers

Explained

Outliers are extreme measures. OLS estimates are vulnerable to the affect of extreme observations. Extreme observations do not necessarily distort a models estimates. For example, if extreme predictor scores lead to extreme outcome scores that are consistent with patterns established among more moderate observations, then outliers need not affect our overall model. Problems emerge when outliers "break the rules", in the sense of defying the patterns that prevail among the more "normal" observations. Under these circumstances, the entire model can be distorted.

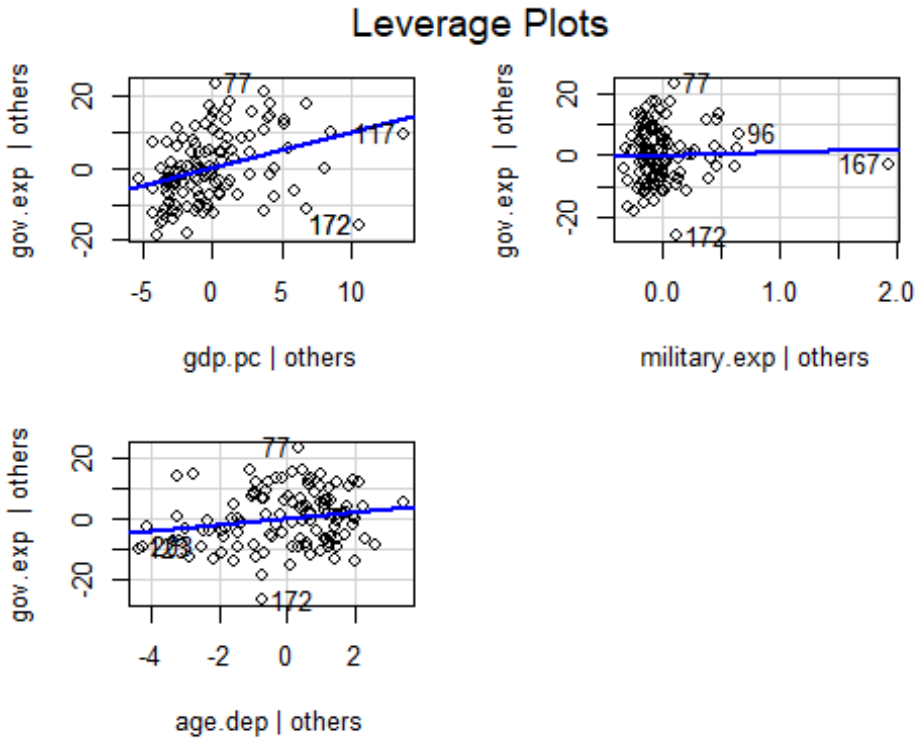


Testing

We can test for outliers using graphical or numeric tests.

Leverage Plots: One way to ascertain potential influential observations is through leverage plots. These plots give the partial relationship between the model outcome and individual predictors (net of other variables in the model). It is implemented as the **leveragePlots()** command in the ***car*** package.

```
library(car)
## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:usdm':
##
##     vif
leveragePlots(reg.2)
```



Here, the most obvious potential problem observation is 167 for military spending (a country with very high military spending -- Saudi Arabia) and 172 whole low government spending and high GDP seems typical (Singapore).

Bonferroni Outlier Test: Alternatively, we can test for influential outliers numerically using a Bonferroni Outlier Test, **outlierTest()** in the ***car*** package.

```
outlierTest(reg.2)
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferonni p
## 172   -3.214           0.001708       0.1998
```

It does not find potentially significant outliers.

Cook's Distance is an alternative method for demarcating potential outliers. It is implemented using the *cooks.distance()** command. Cook's Distance scores above 4/N should be considered as potential outliers.

```
cd <- cooks.distance(reg.2)
cd > 4/nrow(reg.2$model)

##     1     2     6     8     9    11    12    13    16    18    19    20
##  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    25    26    27    30    31    33    34    35    36    38    41    42
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    43    47    48    49    52    53    54    57    59    60    63    65
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    67    68    69    71    73    74    75    77    81    86    88    90
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    91    93    94    96    97    98    99   100   101   102   105   107
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##   108   110   111   112   116   117   120   121   123   124   127   128
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   130   132   134   136   137   139   140   142   143   146   148   153
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   154   155   156   157   158   161   162   163   167   168   170   172
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
##   174   175   178   180   181   186   188   189   192   193   194   195
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
##    197    199    203    204    206    207    208    216    217
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

This suggests that observations #1, 167, 172, 189, and 194 are potential outliers.

Rederesses

- Drop outliers on grounds that it is an atypical case. Be careful with this one - there is a fine line between excluding atypical cases and systematically eliminating cases that do not fit your theory.
- Variable transformation
- Model respecification
- Top/bottom coding

Linearity

Explained

The presumption of "linearity" presumes that the relationship between dependent and independent variables are linear. In other words, it presumes that the fact of adding +1 to the independent variable has the same predicted effect on the dependent variable, regardless of whether or not our independent score is low, middle-range, or high.

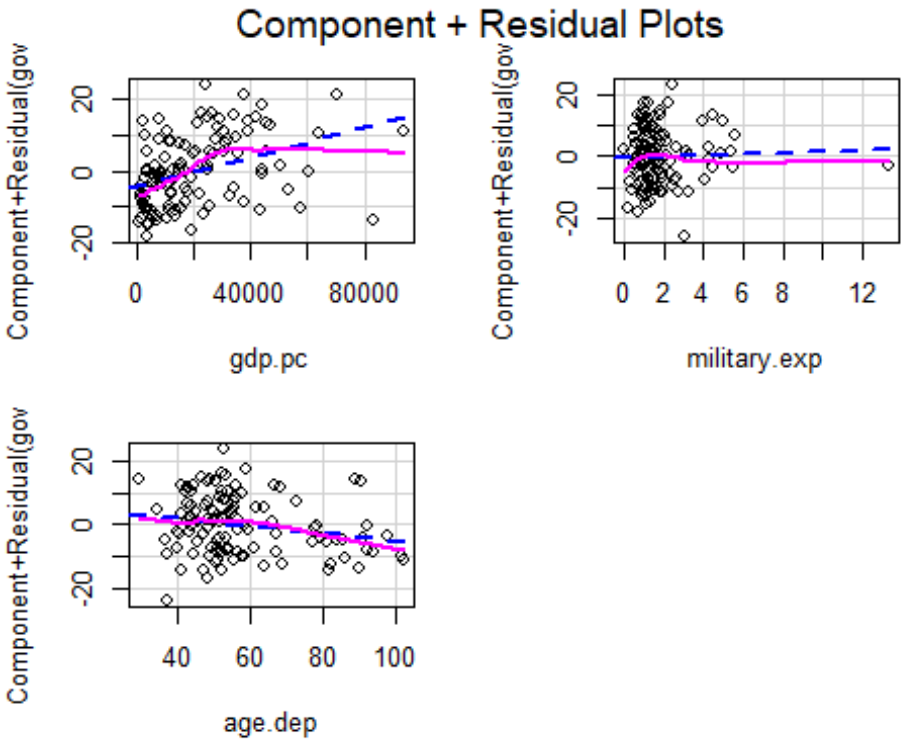
Coefficients are reported as linear, and they can misrepresent non-linear relationships.

Testing

Linearity can be assessed graphically using a Component-Residual Plot. These plots overlay a red line denoting an estimated linear relationship between a predictor and outcome (net of other factors in the model) across all values, alongside a green line denoting this relationship locally.

This is implemented using the **crPlots()** command in the ***car*** package.

```
crPlots(reg.2)
```



If the purple and blue lines are roughly similar, then the linearity condition is more likely to be satisfied. If they diverge substantially, then the relationship may be nonlinear. Here, there's clearly a linearity concern with the variable "gdp.pc", and it's too hard to say anything about "military.exp" with that outlier.

Redresses

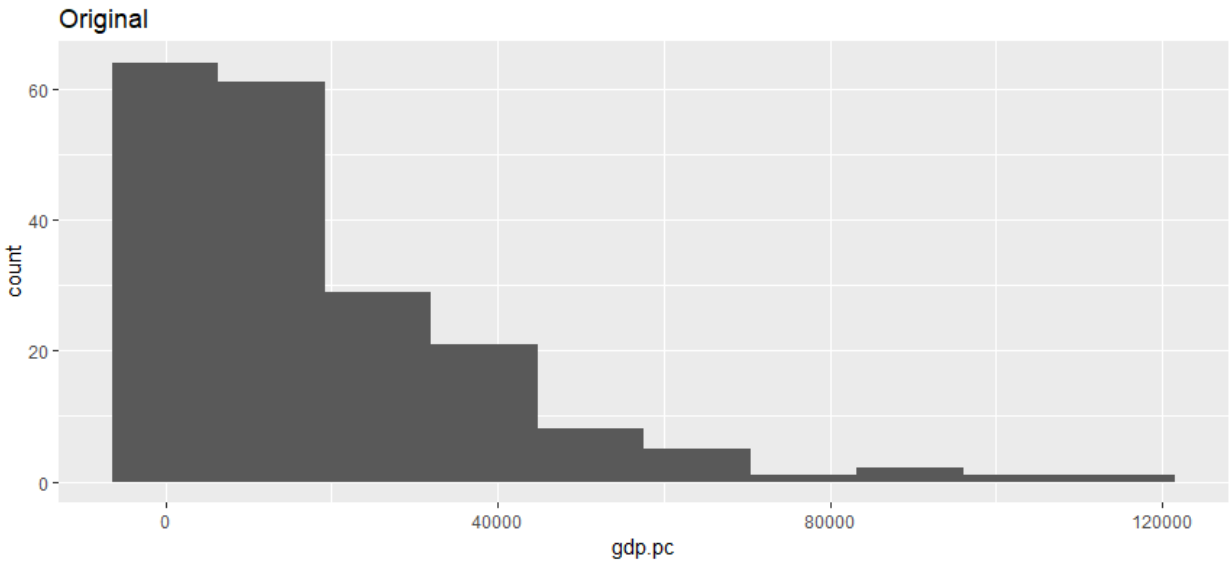
We have a few rederesses to a violation of the linearity assumption:

- Variable transformation
- Model respecification
- Quadratic predictors

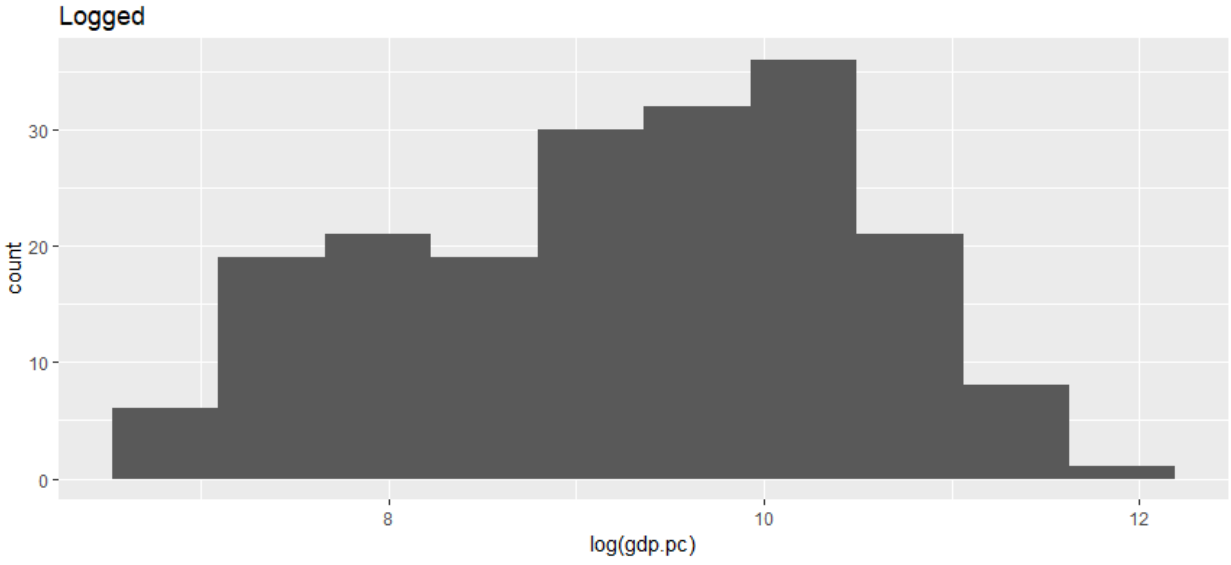
REDRESSES

Variable Transformation

Variable transformation means re-scaling a variable. For example, analysts often transform right-skewed variables (where values are clustered at low levels, but with large positive outliers) by logging them. Such a variable - per capita GDP among the world's countries - is depicted below:



If we were to log that variable, it would look something like this:



The log of that variable has values that are more evenly spread out. Using these transformed values instead of their originals will help insulate us from the threat of influential outliers, can produce linear estimates, and help create balanced, constant errors.

If we re-run the model with a transformed per capita GDP predictor:

```
dat$l.gdppc <- log(dat$gdp.pc)
reg.3 <- lm(gov.exp ~ l.gdppc + military.exp  + age.dep, data = dat)
summary(reg.3)
##
## Call:
## lm(formula = gov.exp ~ l.gdppc + military.exp + age.dep, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.947  -5.668   0.494   5.075  21.302
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -26.1234    12.6935   -2.06    0.042 *
## l.gdppc        5.4383     1.0187    5.34  4.9e-07 ***
## military.exp   0.1392     0.5057    0.28    0.784
## age.dep        0.0254     0.0696    0.36    0.716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.65 on 113 degrees of freedom
##   (100 observations deleted due to missingness)
## Multiple R-squared:  0.306,  Adjusted R-squared:  0.287
## F-statistic: 16.6 on 3 and 113 DF,  p-value: 5.39e-09
```

Although not reported here, the R-Squared rose considerably, suggesting model fit improved by logging the per capita GDP variable. But how do we interpret these variables? In short:

- Logged predictor – compare two concrete data points
- Logged outcome – coefficient is the change in the outcome for a +1 addition to the predictor
- Both – coefficient is percent change in the outcome for a +1% change in the predictor

Let's elaborate below:

Interpreting Models with Logged Variables

Scenario #1: Predictor logged, outcome not

For example:

$$\text{nchildren} = \alpha + \beta * \log(\text{income}) + \epsilon$$

Interpretation:

- β = effect on nchildren if log(income) + 1
- Problem: Hard to make concrete sense of adding +1 to a logged variable.
- We want to make sense of adding +1 (or whatever) to income, not its log.

Solution:
Explain the predicted effects between two X values of your choosing.

Assume β = 0.5, then the predicted difference between a family earning $50k and $100k is:

β * [log(X1) - log(X2)] 0.5 * [log(100,000) - log(50,000)] 0.5 * [log(100,000/50,000)] 0.5 * log(2) = 0.34 children

Scenario #2: Outcome logged, predictor not

For example:

$$\log(\text{income}) = \alpha + \beta * \text{children} + \epsilon$$

Interpretation:

β = effect on log income if children + 1 (β * 100) can be interpreted as % change in income with children + 1

For example:

- = 0.07 predicts a +7% change income with each additional child
- = 1.01 predicts a +101% change in income with each additional child

Scenario #3: Logged Outcome and Predictor

For example:

$$\log(\text{income}) = \alpha + \beta * \log(\text{yrsofeducation}) + \epsilon$$

Interpretation:

- β = % change in Y with +1% change in X
- β = 0.005 for income will rise 0.5% as years of schooling rises by +1%

Model Respecification

Model respecification means changing - and hopefully improving - the model you are testing. One common way to respecify model is to add or subtract variables. For example, we may wish to eliminate variables that are not important - maybe they are adding noise to the model. Adding new variables is straightforward, but choosing which variables to drop can be a little more complicated

Which variables should be dropped? Those with insignificant coefficients are obvious candidates, but not all insignificant variables are unimportant - sometimes are strong predictors rely on the presence of non-significant predictors in the model (their effect might be strong, but only net of another factor that may not be influential in and of itself).

One way to test whether or not we can safely drop an insignificant variable from the model is to use a likelihood ratio test, which tests whether the models estimates are significantly different if a predictor is removed. Likelihood ratio tests are implemented using the **lrtest()** command from the ***lmtest*** package.

Likelihood ratio tests compared to models - a larger one (which includes all of the variables) and a "trimmed" model (which excludes the variables we want to drop). We must create an object with the trimmed model and then tested:

```
#Recall that our model 3 is:
reg.3$call
## lm(formula = gov.exp ~ l.gdppc + military.exp + age.dep, data = dat)
#We need to create a data set with only the observations in regression 3 (som
e observations will be dropped for missing values)

#Create a trimmed model using the data
reg.3.t <- lm(gov.exp ~ l.gdppc, data = reg.3$model)
summary(reg.3.t)
##
## Call:
## lm(formula = gov.exp ~ l.gdppc, data = reg.3$model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.841  -5.794   0.404   5.297  21.423
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -22.235      6.967   -3.19   0.0018 **
## l.gdppc        5.207      0.734    7.09  1.1e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.58 on 115 degrees of freedom
## Multiple R-squared:  0.304,  Adjusted R-squared:  0.298
## F-statistic: 50.3 on 1 and 115 DF,  p-value: 1.14e-10
lrtest(reg.3, reg.3.t)
## Likelihood ratio test
##
```

```
## Model 1: gov.exp ~ l.gdppc + military.exp + age.dep
## Model 2: gov.exp ~ l.gdppc
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1   5   -416
## 2   3   -417 -2  0.21        0.9
```

The trimmed model looks pretty similar, and the likelihood ratio test results are not significant, which implies that we can drop the variables without substantially altering the effects of those that remain in the model. We have trimmed Model 3.