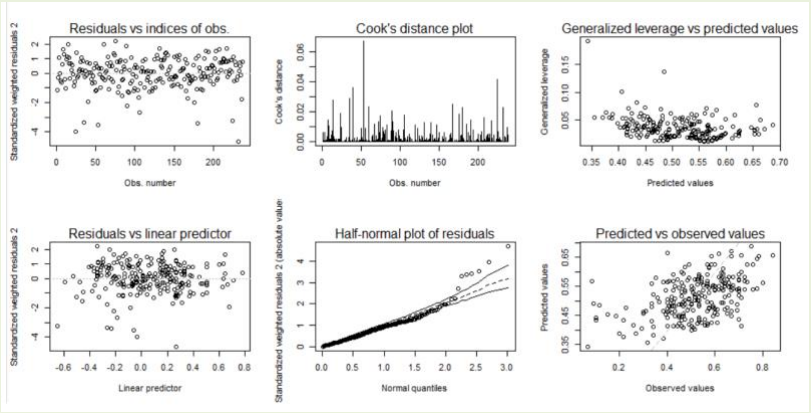
BIOSTATS 640 – Introduction to R
Fall 2023
https://people.umass.edu/biep640w/webpages/demonstrations.html



https://stackoverflow.com/questions/56316077/adding-labels-to-diagnostic-plots-in-r

08
Regression Diagnostics for
Normal Theory Regression in R
October 27, 2023

Dataset used
hersdata_small.xlsx

Packages used: `tidyverse, stargazer, car, ggplot2, ggfortify, lmtest, stargazer, gridExtra`

1. Introduction to The Heart and Estrogen/progestin Replacement Study (HERS)

hersdata_small.xlsx

Source

Hulley et al (1998) Randomized trial of estrogen plus progestin for secondary prevention of heart disease in postmenopausal women. The Heart and Estrogen/progestin Replacement Study. *Journal of the American Medical Association*, **280**(7), 605-613

The Heart and Estrogen/Progestin Replacement Study (HERS) was a randomized clinical trial of hormone therapy (estrogen plus progestin) for the reduction of cardiovascular disease risk in post-menopausal women with established coronary disease. Study participants were n=2,763 women who were: (1) post-menopausal (2) with coronary disease; and (3) with an intact uterus.

The data set for this illustration is a simple random sample of n=1000. A subset of the variables are considered:

Data dictionary/Codebook *(Partial)*

Variable	Label	Type	Codings
age	Age, years	numeric	Continuous, range, [45:79]
BMI	Body Mass index (kg/m^2)	numeric	Continuous, range, [15.21:54.13]
glucose	Fasting glucose (mg/dL)	numeric	Continuous, range, [29:298]
LDL	LDL cholesterol (mg/dL)	numeric	Continuous, range, [44.4:393.4]
drinkany	Any current alcohol use	numeric	1 = yes 0 = no
exercise	Exercise at least 3x/week	numeric	1 = yes 0 = no
HT	Randomization	numeric	1 = hormone therapy 0 = placebo
physact	Comparative ("compared to other women your age") physical activity	Numeric	1 = much less active 2 = somewhat less active 3 = about as active 4 = somewhat more active 5 = much more active
statins	Statin use	Numeric	1 = yes 0 = no
diabetes	Diabetes	Numeric	1 = yes 0 = no

2. Highlights of Lesson 07
Introduction to Multiple Linear Regression in R

Multiple Linear Regression

Fit model. Save as object.
```
fitobject <- lm(yvar ~ xvar1 + xvar2 + xvar3, data=dataname)
```
Example:
```
mfull <- lm(ln_sbp ~ ln_bmi + ln_scl, data=mydata)
```

Return names of model fit object.
```
names(fitobject)
```
Example:
```
names(mfull)
```

Show model output.
```
summary(fitobject)
```
Example:
```
summary(mfull)
```

Show show regression estimates and confidence intervals.
```
cbind(coef(fitobject),confint(fitobject))
```
Example:
```
cbind(coeff(mfull), confint(mfull))
```

Show analysis of variance table.
```
anova(fitobject)
```
Example:
```
anova(mfull)
```

Compare nested/hierarchial Models. Partial F-Test.
```
m_reduced <- lm(yvar ~ controlvar1 + controlvar2, data=dataname)
m_full <- lm(yvar ~ controlvar1 + controlvar2 + extravar1 + extravar2 + extravar3,
data=dataname)
anova(m_reduced,m_full)
```
Example:
```
m_reduced <- lm(ln_sbp ~ ln_bmi, data=mydata)
m_full <- lm(ln_sbp ~ ln_bmi + ln_scl, data=mydata)
anova(m_reduced, m_full)
```

Nice tabular side-by-side comparison of models using package {stargazer}
```
library(stargazer)
stargazer(model1, model2, model3, type="text")
```
Example:
```
stargazer(m_reduced, m_full, type="text")
```

3. At a Glance/Quick Single Commands for Multiple Diagnostics

plot( ). No package necessary

Command	Plot Produced
plot(fit, which=1)	X = fitted value Y = residual
plot(fit, which=2)	X = theoretical normal quantile Y = studentized residual
plot(fit, which=3)	X = fitted value Y = square root (standardized residual)
plot(fit, which=4)	X = observation number Y = Cook's Distance
plot(fit, which=5)	X = leverage Y = standardized residual
plot(fit, which=6)	X = leverage Y = Cook's Distance
plot(fit)	Default is four plots: which=1, which=2, which=3, and which=5

residualPlots( ) in package {car}

This command also provides, for each predictor X, a t-test of NULL: "no curvature" quadratic X^2 is not statistically significant. It also provides the Tukey test of NULL: "the model is additive"

Command	Plots Produced
residualPlots(fit)	For each predictor: X = predictor Y = residual And also: X = fitted Y = residual
residualPlots(fit, ~X1)	For single predictor of interest: X = predictor Y = residual And also: X = fitted Y = residual
residualPlots(fit, ~1)	X = fitted Y = residual *ONLY*

autoplot( ) in package {ggfortify}.

To be safe you might need to have library(ggplot2)

Command	Plot Produced
autoplot(fit, which=1, option, option)	X = fitted value Y = residual
autoplot(fit, which=2, option, option)	X = theoretical normal quantile Y = studentized residual
autoplot(fit, which=3, option, option)	X = fitted value Y = square root (standardized residual)
autoplot(fit, which=4, option, option)	X = observation number Y = Cook's Distance
autoplot(fit, which=5, option, option)	X = leverage Y = standardized residual
autoplot(fit, which=6, option, option)	X = leverage Y = Cook's Distance
autoplot(fit, which=1:4, option, option)	Note - You can select which plots you want.

4. Regression Diagnostics
Model Specification

Preliminary: Fit the model and show

In this illustration, we are following the solutions to the 3rd homework for Unit 5 - Normal Theory Regression. Here:

- Y = glucose
- Three predictors: X_1=age, X_2=BMI and X_3=drinkany.
- We consider only those for whom diabetes=0

$$glucose = \beta_0 + \beta_1 \cdot age + \beta_2 \cdot BMI + \beta_3 \cdot drinkany$$

```
import source data
library(readxl)
source <- read_excel("hersdata_small.xlsx")
```

```
create data ready for analysis
library(tidyverse)

ready <- source %>%
         filter(diabetes==0) %>%                    # filter( ) to select observations rows( )
         select(id, glucose,age,BMI,drinkany) %>%   # select( ) to select variables columns( )
         mutate(drinkanyf = factor(drinkany,        # mutate(NEWVARIBLE = ) to create new variable
                                levels=c(0,1),
                                labels=c("0 = no", "1 = yes"))) %>%
         na.omit()                                  # na.omit( )to output complete data ONLY

ready <- as.data.frame(ready)
glimpse(ready)

## Rows: 748
## Columns: 6
## $ id        <dbl> 1, 4, 5, 6, 7, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 22,…
## $ glucose   <dbl> 115, 96, 109, 108, 111, 90, 90, 108, 107, 80, 90, 92, 94, 10…
## $ age       <dbl> 76, 62, 54, 58, 69, 70, 63, 64, 66, 65, 71, 72, 76, 73, 65, …
## $ BMI       <dbl> 21.68, 26.93, 38.14, 33.70, 26.20, 26.84, 33.31, 22.42, 27.2…
## $ drinkany  <dbl> 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, …
## $ drinkanyf <fct> 1 = yes, 0 = no, 1 = yes, 0 = no, 0 = no, 0 = no, 1 = yes, 1…
```

Fit model. Show.

```
library(stargazer)
fit <- lm(glucose ~ age + BMI + drinkany, data=ready)

stargazer(fit,type="text",                                          # MUST have option type="text"
          font.size="small",                                        # additional options as you like
          align=TRUE,
          ci=TRUE,
          intercept.bottom=FALSE,
          covariate.labels=c("Intercept",
                             "age:  Age (years)",
                             "BMI:  Body Mass Index (kg/m2)",
                             "drinkany:  Any current alcohol use"),
           dep.var.labels=c("Y = glucose"),
           title="Fitted Model (fit):  Betas (95% CI)")fit <- lm(glucose ~ age + BMI + drinkany, data=ready)
```

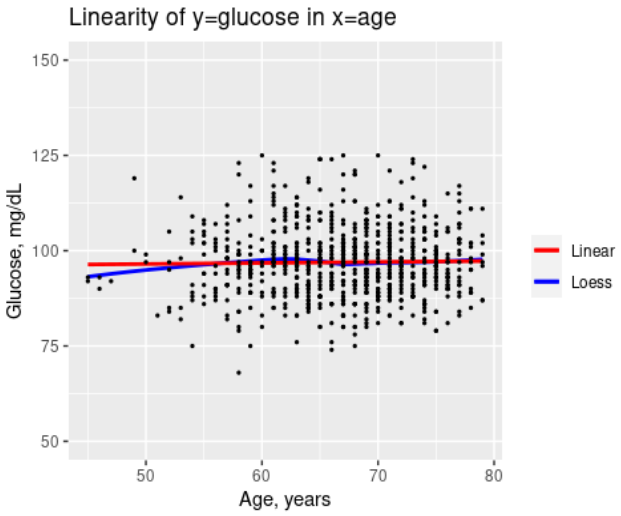
```
Fitted Model (fit): Betas (95% CI)
=============================================================
                                    Dependent variable:
                                -----------------------------
                                       Y = glucose
-------------------------------------------------------------
Intercept                                80.073***
                                      (72.120, 88.026)

age: Age (years)                          0.056
                                      (-0.044, 0.156)

BMI: Body Mass Index (kg/m2)              0.484***
                                      (0.356, 0.612)

drinkany: Any current alcohol use         -0.388
                                      (-1.713, 0.938)

-------------------------------------------------------------
Observations                                748
R2                                          0.071
Adjusted R2                                 0.067
Residual Std. Error                   9.156 (df = 744)
F Statistic                         18.846*** (df = 3; 744)
=============================================================
Note:                             *p<0.1; **p<0.05; ***p<0.01
```

4.1. Linearity

Goal	At each vector level "$\underline{x} = [x_1, x_2, \ldots, x_p]$ " of the predictor vector $\underline{X}$, the mean of $Y_{\underline{X}}$ lies on a line, $\mu_{Y|\underline{x}} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$
What to watch out for	Departure from linearity

ggplot( ) in package {ggplot2}: XY Scatterplot with Overlay Line and Loess.

```
library(ggplot2)
# get min and max of Y for setting a common y-axis
min(ready$glucose)
## [1] 68

max(ready$glucose)
## [1] 125

# linearity in age
ggplot(data=ready) +
    aes(y=glucose) +
    aes(x=age) +

    geom_smooth(method="loess", aes(color="Loess"), se=FALSE) +                    # Loess smooth w no CI
    geom_smooth(method="lm", aes(color="Linear"), se=FALSE) +                      # linear fit w no CI
    geom_point(size=0.5) +                                                         # X-Y scatter
    scale_colour_manual(name="", values=c("red","blue")) +
    scale_y_continuous(limits = c(50,150), breaks = seq(50,150, by=25))  +         # set y-axis explicitly
    ggtitle("Linearity of y=glucose in x=age") +
    xlab("Age, years") +
    ylab("Glucose, mg/dL")
```



Interpretation: Linearity of Y=glucose in X=age can reasonably be assumed.

4.2. Multicollinearity

Multicollinearity occurs when the predictors are themselves linearly related. If extensive, multicollinearity is a problem because each predictor on its own possesses too little independent information for the prediction of outcome. A measure of this is the variance inflation factor statistic, VIF. The VIF for the i^{th} variable is defined:

$$VIF_i = \frac{1}{\sqrt{1 - R^2_{\text{regression of ith on all other predictors}}}}$$

Briefly, to obtain the VIF for a particular predictor, that predictor is regressed on all the other predictors and an R-squared is obtained. The VIF for the predictor is then obtained as follows. Values of VIF < 10 are considered acceptable (translation: no worries!):

Goal	Values of VIF < 10 suggest all is well. Note: Some software packages report the inverse VIF^{-1}. All is well if $VIF^{-1} > .10$
What to watch out for	Values of VIF > 10 or, values of the inverse, $VIF^{-1} < .10$

vif(fit) in package {car}

```
library(car)
vif(fit)
```

```
##      age      BMI drinkany
## 1.017763 1.013782 1.015087
```

Interpretation: All is well. All the VIF are much less than 10.

4.3. Partial Regression Plot/Added Variables Plot

A **partial regression plot**, also called an **added variables plot**, provides a visual assessment of a candidate predictor's significance for the prediction of the outcome, **after controlling for other predictors already in the model**.

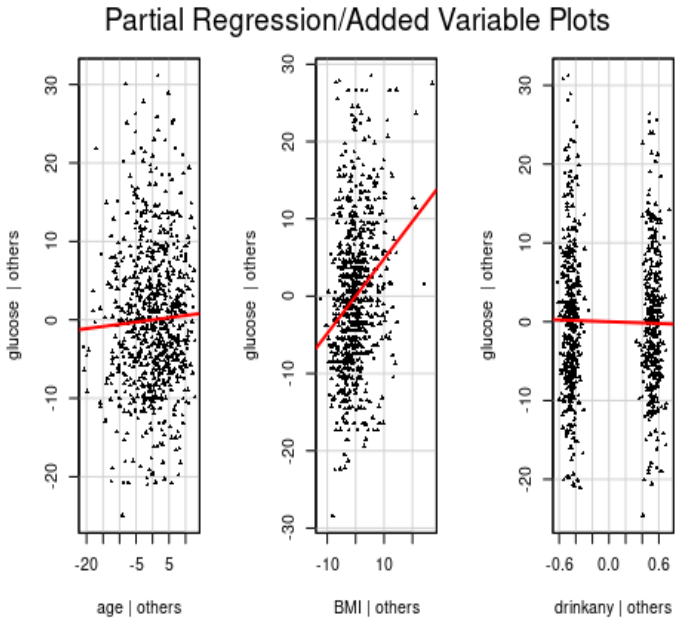
- **Vertical axis**: Residuals for model of Y on the control variables
- **Horizontal axis**: Residuals for model of candidate predictor on the control variables

By plotting residuals after adjustment for the control variables, you get an assessment of the nature and significance of the candidate predictor for the prediction of the outcome with the control variables "adjusted out."

Goal	To visualize the independent influence of each predictor, controlling for all the other predictors.
What to watch out for	- A flat line suggests the candidate predictor is not important - Linearity suggests that the candidate predictor should be added to the model - Curvilinearity suggests that the candidate predictor should be included but perhaps with some additional terms (e.g., as a quadratic)

avPlots(fit, option, option) in package {car}

```
library(car)
avPlots(fit,
        id=FALSE,                                          # suppress id's of extreme residuals
        pch=17,                                            # plotting character is diamond
        cex=0.5,                                           # point size
        col.lines="red",                                   # color of partial regression line
        main="Partial Regression/Added Variable Plots",
        layout=c(1,3))                                     # 3 graphs in 1 row, 3 columns
```



Interpretation: These pictures suggest that, *after controlling for other predictors in the model*: (1) age is linearly related to glucose; (2) BMI is linearly related to glucose; but that (3) current alcohol use (yes/no) is not associated with glucose.

4.4. Omitted Variables

Another issue in variable selection is the possibility of ***model misspecification*** which occurs if the predictors are not modeled correctly (e.g., linearity in the predictor is insufficient) or important predictors are missing. The **Ramsey test** tests the null hypothesis the curent model is adequately specified.

Goal	Null Hypothesis: No omitted variables All is well if p-value is NOT statistically significant
What to watch out for	Statistical significance (small p-value) suggests either that important variables have been omitted and/or that the model has been misspecified with respect to its included predictors.

resettest(fit, power=2, type="regressor") in package {lmtest}

```
library(lmtest)
resettest(fit, power=2, type="regressor")

##
##  RESET test
##
## data:  fit
## RESET = 0.18324, df1 = 3, df2 = 741, p-value = 0.9078
```

Interpretation: Do NOT reject the null hypothesis of adequate model specification (p-value = .91). We have no statistically significant evidence that the model is misspecified (either with respect to its included predictors or with respect to omitting important predictors).

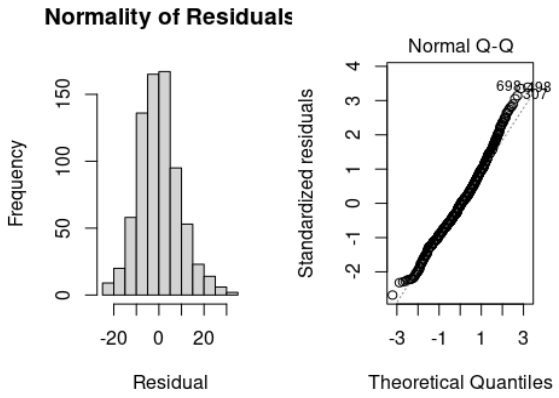
4.5. Normality of Residuals

A key assumption of normal theory linear regression is that the distribution of the outcome Y at each level of the predictor is normal (and with constant variance, as discussed below). When this assumption is met, ***the distribution of the residuals is distributed Normal with mean = 0 and constant variance***.

Goal	All is well if the histogram of the residuals can reasonably be assumed to be distributed Normal (and with constand variance, as discussed below) and if the QQ plot is a straight line, or reasonably so.
What to watch out for	Histogram: Departures from a reasonably shaped bell distribution QQ Plot: Departures from straight line.

hist(saved.residuals) and plot(fit, which=2)

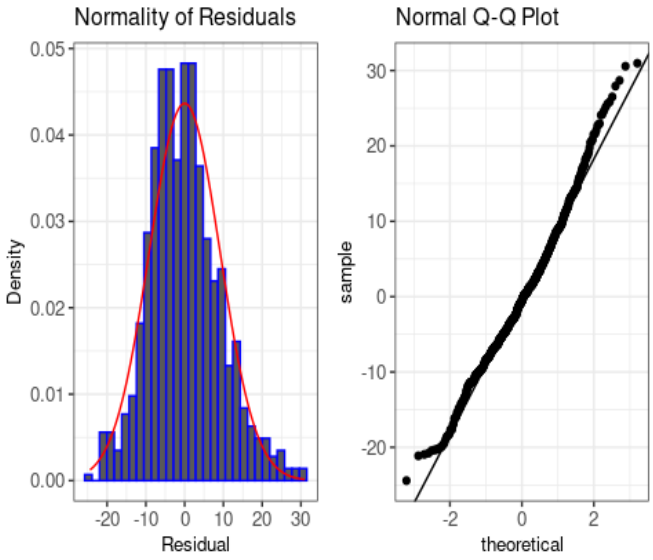
```
ready$fit.resid <- resid(fit)                          # resid( ) to get residuals from fit
par(mfrow = c(1,2))                                    # set graph to be 2 panes (1 row, 2 col)
hist(ready$fit.resid,                                  # histogram of residuals (look for normality)
     main="Normality of Residuals",
     xlab="Residual")
plot(fit, which = 2)                                   # qqplot (look for straight line)
```



Interpretation: Not bad. The bell shape of the histogram is consistent with normality. The normal QQ plot is (mostly) linear, which is also what we look for in assessing normality of the residuals.

ggplot( ) in package {ggplot2} and grid.arrange( ) in package {gridExtra}

```
par(mfrow = c(1,1))                                          # return graph setting to single pane
library(car)
library(ggplot2)
library(gridExtra)


# panel 1 = plot of residuals w overlay normal
p1 <- ggplot(data=ready) +
        aes(x=fit.resid) +
        geom_histogram(colour="blue",
                       aes(y=..density..)) +
        stat_function(fun=dnorm,
                      color="red",
                      args=list(mean=mean(ready$fit.resid),
                                sd=sd(ready$fit.resid))) +
        ggtitle("Normality of Residuals") +
        xlab("Residual") +
        ylab("Density") +
        theme_bw() +
        theme(axis.text = element_text(size = 10),
              axis.title = element_text(size = 10),
              plot.title = element_text(size = 12))


# panel 2 = quantile-quantile plot
p2 <- ggplot(data=ready) +
         aes(sample=fit.resid) +
         stat_qq() +
         stat_qq_line() +
         ggtitle("Normal Q-Q Plot") +
         theme_bw() +
         theme(axis.text = element_text(size = 10),
               axis.title = element_text(size = 10),
               plot.title = element_text(size = 12))

gridExtra::grid.arrange(p1, p2, ncol=2)
```



Interpretation: Same. Thiis just a prettier picture.

```
shapiro.test(saved.residuals)
```

```
shapiro.test(ready$fit.resid)

##
##  Shapiro-Wilk normality test
##
## data:  ready$fit.resid
## W = 0.9873, p-value = 0.000004419
```

Interpretation: This is a nice example of how sample sizes that are very large (here, n=748) can produce statistical significance when, in reality, the data themselves do not suggest a meaningful departure from the null. A great reminder of the importance of looking at the data!

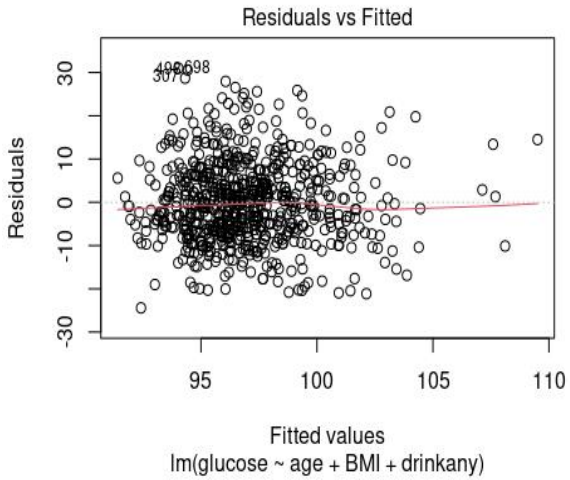
4.6. Constant Variance of Residuals

Here, we address the second aspect of the assumption of normality: **constant variance**. This is addressed by assessing the ***constancy of variance of the residuals.***

Goal	A plot of residuals (vertical) by fitted values (horizontal) should display an even band, centered at 0
What to watch out for	Any sort of wedge shape scatter suggests non-constant variance But also, any sort of curvilinear scatter is noteworthy too; it suggests model misspecification.

```
plot(fit, which=1)
```

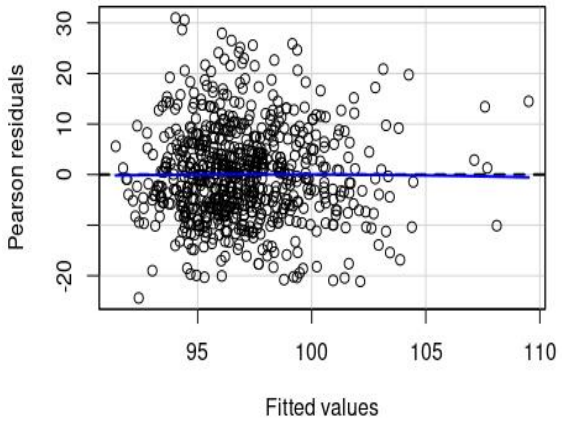
```
plot(fit, which = 1)                                    # which=1 to plot X=predicted v Y=residual
```



Interpretation: Looks okay.

residualPlots(fit, ~1, fitted=TRUE) in package {car}

```
library(car)
residualPlots(fit, ~ 1, fitted=TRUE)            # residualPlots() will also provide a test of the null
```

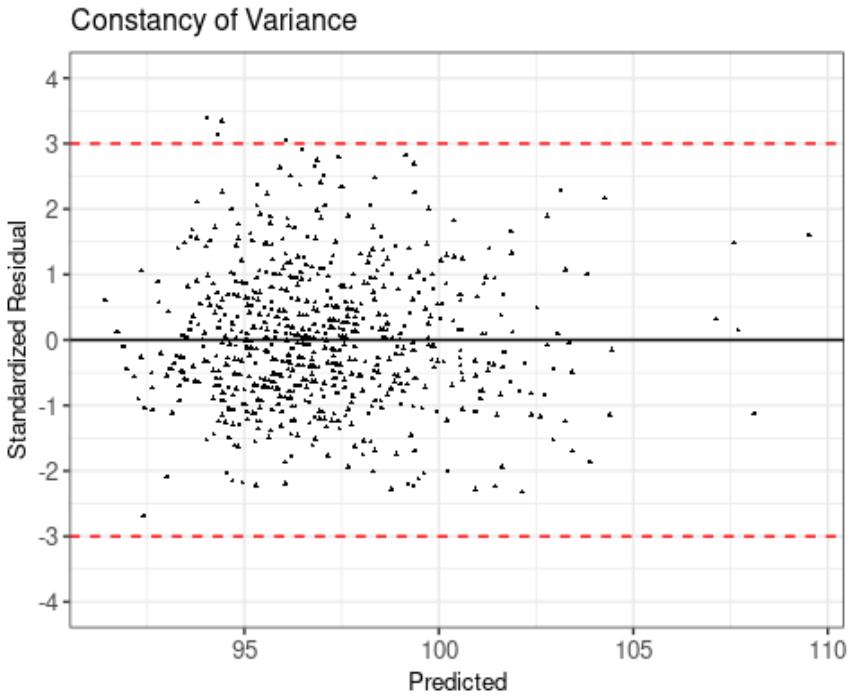



```
##            Test stat Pr(>|Test stat|)
## Tukey test   -0.1537           0.8778
```

Interpretation: Do NOT reject the null hypothesis of constant variance (p-value = .88) The picture looks similar to previous picture. While this procedure is convenient in providing both a graph and a statistical hypothesis test, the first picture has the advantage of labeling of the X and Y axes more explicitly.

ggplot( ) in package {ggplot2}

```
library(ggplot2)

ready$yhat <- fitted(fit)                              # fitted( ) to get predicted values
ready$estandard <- rstandard(fit)                      # rstandard( ) to get standardized residuals


ggplot(data=ready) +
     aes(x=yhat) +
     aes(y=estandard) +

     geom_point(size=0.5, pch=17) +                                    # pch=17 for diamonds
     geom_hline(yintercept=0, color="black") +                         # line at expected residual = 0
     geom_hline(yintercept=3, linetype="dashed", color="red") +        # line at +3 std
     geom_hline(yintercept=-3,linetype="dashed", color="red") +        # line at -3 std

     scale_y_continuous(limits = c(-4,4), breaks = seq(-4, 4, by=1))  +      # set y-axis explicitly

     ggtitle("Constancy of Variance") +
     xlab("Predicted") +
     ylab("Standardized Residual") +
     theme_bw() +
     theme(axis.text = element_text(size = 10),
           axis.title = element_text(size = 10),
           plot.title = element_text(size = 12))
```



NOTE! I plotted the standardized residuals because I link to think in terms of Z-scores (approx). I also provided reference lines at ± 3 standard deviations away from the expected value of 0. We can see that there is, really, not much of a problem.

ncvTest(fit) in package {car}

```
library(car)
ncvTest(fit)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.832514, Df = 1, p = 0.17583
```

Interpretation: Do NOT reject the null hypothesis of constant variance (p-value = .18) I'm not sure why the p-values for the 2 tests of non-constant variance are so different (.88 versus .18). I'll have to look into that. Mercifully, the conclusion is the same.

5. Regression Diagnostics
Case Analysis

5.1. Outliers (Y "Unusualness")

Outliers are observations that are unusual in the Y-sense. They may or may not influence the fitted model. But it's good to take a look. The Bonferroni test examines the largest studentized residual. For this particular studentized residual it peforms a t-test of the null hypothesis that it is not statistically significantly different from the other studentized residuals.

Goal	All is well if most of the | studentized residuals | are $\leq$ 2-3 or so.
What to watch out for	| Studentized residuals | $\geq$ 3 or so.

outlierTest(fit) in package {car}

```
library(car)

# outlierTest() for detecting observations with large standardized residuals
outlierTest(fit)

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 698  3.41559         0.00067102      0.50193
```

Interpretation: In my opinion, a statistical test investigating a single outlier is not very useful and that it would be better to inspect the plot on the previous page.

5.2 Leverage (X "Unusualness")

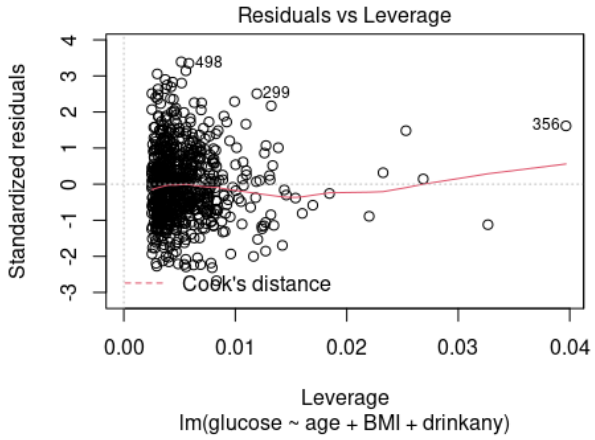
High leverage observations are observations that are unusual in the X-sense. They may or may not influence the fitted model.

Goal	All is well if you don't see any "X-unusual" observations.
What to watch out for	Histogram: Departures from a reasonably shaped bell distribution QQ Plot: Departures from straight line.

High leverage observations are observations that are unusual in the X-sense. They may or may not influence the fitted model. By any means you like, assess the fitted model with respect to leverage.

```
plot(fit, which=5)
```

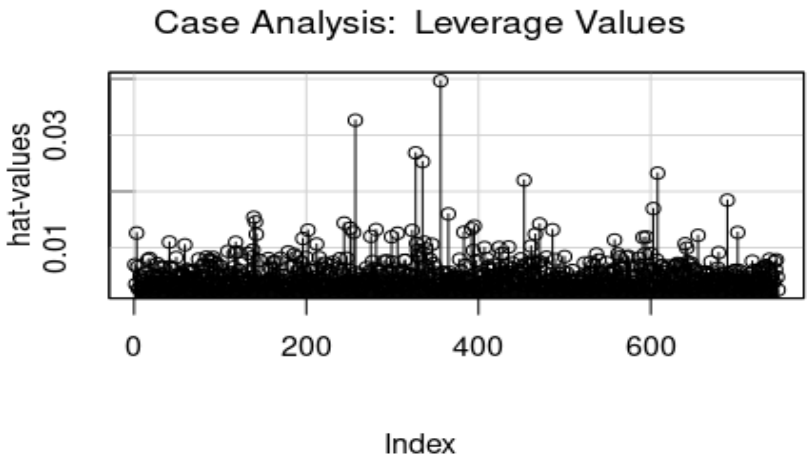
```
# leverage, basic plot
plot(fit, which = 5)                          # which=5 to plot Y=standardized residual v X=leverage
```



Interpretation: The usefulness of this graph is that it shows you the observations that are unusual in BOTH the X-sense (leverage) and the Y-sense (studentized residual). Keep in mind, however, this may or may not mean that the point is influential in determining the estimates of the betas.

influenceIndexPlot(fit, vars=c("hat"), id=FALSE, option, option) in package {car}

```
library(car)

# Leverage, fancy
influenceIndexPlot(fit, vars=c("hat"),          # vars=c("CHOOSE") from "Studentized","Bonf","hat", "Cook"
                   id=FALSE,
                   main="Case Analysis:  Leverage Values")
```



Interpretation: All's well; nothing to worry about. The values of the leverage are all much much less than 1.

5.3. Influential Observations

Influential observations do impact the fit! Their inclusion in the model changes the estimated betas. There are several approaches to detect influential observations. Among the most commonly used is the calculation of ***Cook's distance***. Briefly, the Cook's distance is a summary measure of the discrepancy in the estimation betas in two models, one with the observation included and the other with the observation not included. A plot of study id versus Cook's distance makes their detection easy; simply look for spikes! Several thresholds/cutoffs have been suggested for the identification of influential observations.

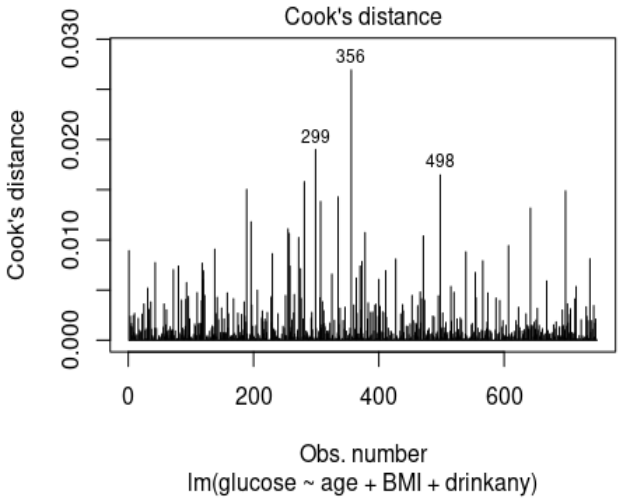
Some suggestions:

- Look at the plot first; where you see spikes, these observations may be influential (take care, however, to notice the range of Cook's distances by examining the y-axis scale provided);
- A Cook's distance > 1 is worth exploring further;
- A Cook's distance > .5 is of mild interest.

Goal	All is well if all Cook's distances are small (<= 1)
What to watch out for	Histogram: Departures from a reasonably shaped bell distribution QQ Plot: Departures from straight line.

```
plot(fit, which=4)
```

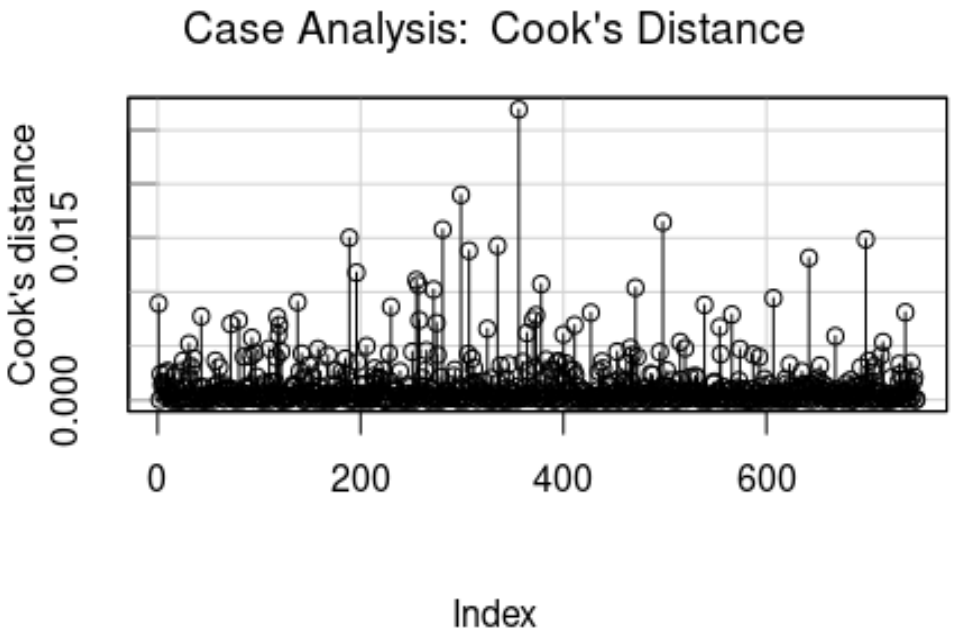
```
# cook's distance, basic plot
plot(fit, which = 4)                              # which=4 to plot X=observation ID v Y=Cook distance
```



Interpretation: AGAIN! Take a look at the Y-axis. The values of the Cook's distances are all much much less than 1. We have nothing to worry about vis a vis influence!

influenceIndexPlot(fit, vars=c("Cook"), id=FALSE, option, option) in package {car}

```
library(car)

# cook's distance, fancy
influenceIndexPlot(fit, vars=c("Cook"),                          # choose: "Studentized","Bonf","hat", "Cook"
                   id=FALSE,
                   main="Case Analysis:  Cook's Distance")
```



Interpretation: Of the two graphs of Cook's distances, I prefer the first because the axes are more clearly labeled.

6. Additional Resources

__1. Introduction to `{broom}`
https://cran.r-project.org/web/packages/broom/vignettes/broom.html

__2. Linear Regression Diagnostics with `{broom}`, `{ggplot2}` and `{regressinator}`
https://cran.r-project.org/web/packages/regressinator/vignettes/linear-regression-diagnostics.html

__3. STHDA Linear Regression Assumptions and Diagnostics in R: Essentials
http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/