Homework #8
Unit 5 – Regression and Correlation (3 of 3)
Practice Problems
Solutions

Before you begin. Download from the course website
hersdata_small.xlsx

Description of Dataset

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 homework is a random sample of n=1000 from the data set **`hersdata_small.xlsx`**. The following variables are considered:

Data dictionary/Codebook *(Partial)*

Variable	Label	Type	Codings
age	Age, years	numeric	Continuous, range, [45:79]
BMI	Body Mass index (kg/m²)	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

The exercises in this assignment give you practice performing regression diagnostics.
They are ***not*** an illustration of an entire regression analysis, beginning with data exploration followed by a series of model estimation followed by diagnostics.

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

Preliminaries

After importing **`hersdata_small.xlsx`**, create a new dataframe called **`ready`** that you will use for multiple predictor regression.

```
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 choose rows/observations
         select(id, glucose,age,BMI,drinkany) %>%             # select( ) to choose columns/variables
         mutate(drinkanyf = factor(drinkany,                  # mutate( ) and factor( ) to create categorical var
                                   levels=c(0,1),
                                   labels=c("0 = no", "1 = yes"))) %>%
         na.omit()

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…
```

#1

Fit. Using as your dependent variable Y=glucose, fit the following 3 predictor model:

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

Check: You should get the following prediction equation.

$$Predicted\ glucose = 80.07 + 0.056 \cdot age + 0.484 \cdot BMI - 0.388 \cdot drinkany$$

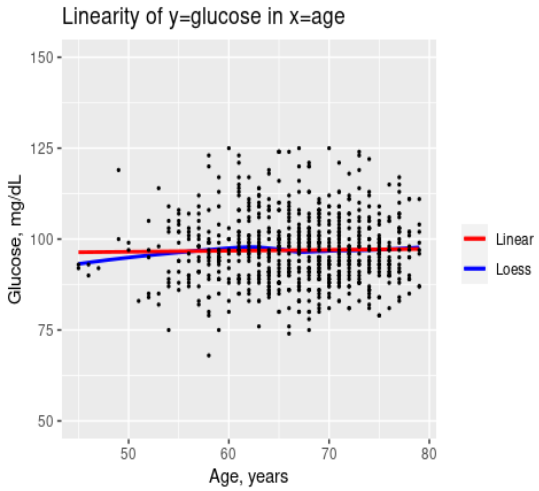
```
Q1. fit model. show.

fit <- lm(glucose ~ age + BMI + drinkany, data=ready)       # lm( yvar ~ x1 + X2, data=FILLIN) to fit model
summary(fit)                                                # summary(MODEL_OBJECT) to show results

##
## Call:
## lm(formula = glucose ~ age + BMI + drinkany, data = ready)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.4162  -6.3124  -0.5711   5.3248  30.9722
##



## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept) 80.07318    4.05782  19.733 < 0.0000000000000002 ***
## age          0.05605    0.05089   1.101                0.271
## BMI          0.48365    0.06526   7.411    0.000000000000341 ***
## drinkany    -0.38757    0.67637  -0.573                0.567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.156 on 744 degrees of freedom
## Multiple R-squared:  0.07063,    Adjusted R-squared:  0.06688
## F-statistic: 18.85 on 3 and 744 DF,  p-value: 0.000000000008651
```

$$\textit{Predicted glucose} = 80.07 + 0.056 \cdot \textit{age} + 0.484 \cdot \textit{BMI} - 0.388 \cdot \textit{drinkany}$$

#2.

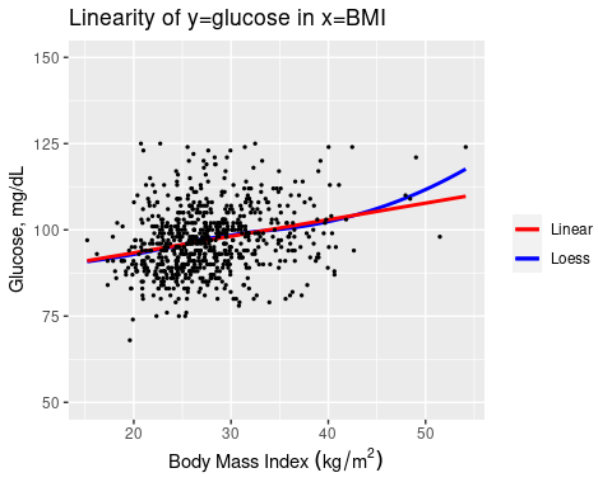
Linearity of Y in the predictors. Normal theory linear regression makes the assumption that, at each level of the predictor (X), the distribution of the outcome Y (this is known as the conditional distribution of Y given X) is distributed Normal with ***means that lie on a line (simple linear regresssion) or a plane (multiple linear regression)***. By any means you like, produce graphs to assess the linearity of Y = glucose in age and the linearity of Y=glucose in BMI.

```
Q2. Regression Diagnostics, model related. Linearity.
library(ggplot2)
# get min and max of Y for setting a common y-axis                                  # convenient for ggplot below
min(ready$glucose)
## [1] 68

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

# linearity in age
ggplot(data=ready) +                              # required layer
    aes(y=glucose) +                              # required layer
    aes(x=age) +                                  # required layer.  Note:  could also do aes(x=age, y=glucose)


    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")) +                        # optional aesthetics
    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.

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

    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))  +       # Recommended:set y-axis explicitly

    ggtitle("Linearity of y=glucose in x=BMI") +
    xlab(expression("Body Mass Index "(kg/m^2))) +
    ylab("Glucose, mg/dL")
```

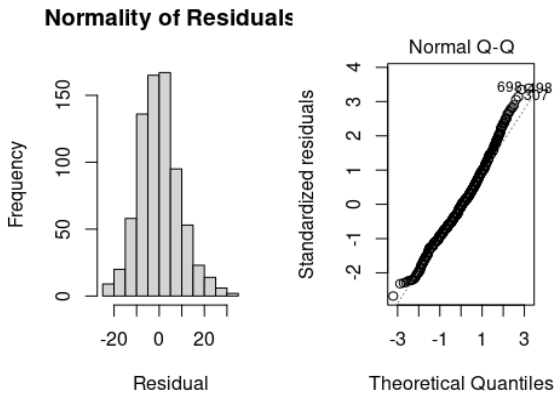


Interpretation: Here, it is a little less clear. Possibly, linearity of Y=glucose in X=BMI is reasonable. However, we do see some curvature in the loess smooth. But this is based on very few observations. Stay tuned. We'll explore this again in a partial F test..

#3.

Y is distributed normal with constant variance. Another assumption of linear regression is that the distribution of the outcome Y at each level of the predictor is normal with constant variance.. When this assumption is met, ***the distribution of the residuals is distributed Normal with mean = 0 and constant variance***. Thus, assessment of this assumption involves examination of the residuals after fitting the model. Consider the model you fit in exercise #1. By any means you like, assess the assumption of normality of the residuals.

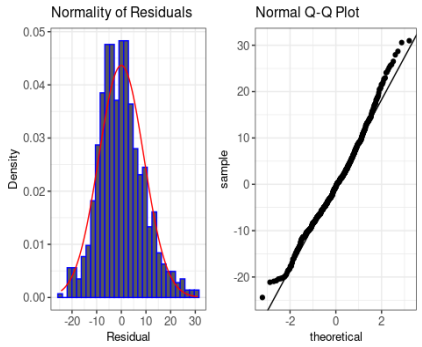
```
Q3. Regression diagnostics, model related. Normality
library(car)
library(ggplot2)
library(gridExtra)

ready$fit.resid <- resid(fit)                                   # resid(MODELOBJECT) to get residuals from fit

# Normality of residuals, basic plot
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.

```
par(mfrow=c(1,1))                                               # return graph setting to single panel!!!!


# panel 1 - Normality of residuals, ggplot
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() +
          geom_abline(intercept=mean(ready$fit.resid),
                      slope = sd(ready$fit.resid)) +
          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.

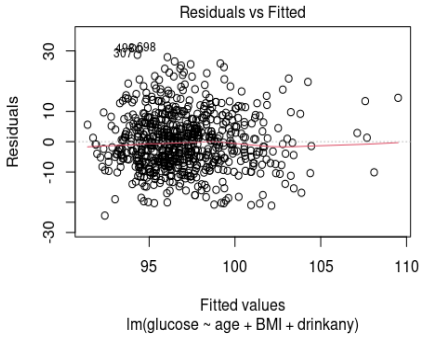
```
# test of normality of residuals
shapiro.test(ready$fit.resid)

##
##  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.

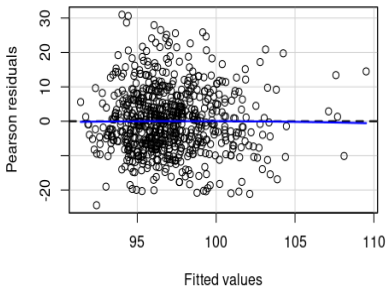
Y is distributed normal with constant variance Consider the model you fit in exercise #1. By any means you like test the null hypothesis of constancy of variance of the residuals.

```
Q4. Regression diagnostics, model related. Constant variance
library(car)
library(ggplot2)

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

# constancy of variance, basic plot
plot(fit, which = 1)                                         # which=1 plots X=predicted v Y=residual
```



Interpretation: For constant variance, we want to see an even band of residual scatter, centered at zero. Looks pretty good here.

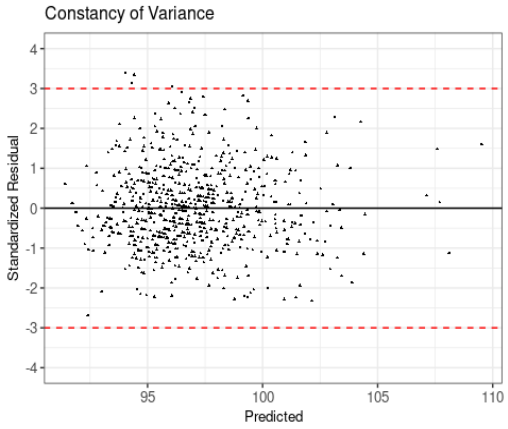
```
# constancy of variance, using in {car}
residualPlots(fit, ~ 1, fitted=TRUE)                    # residualPlots() will also provide a test of the null
```




```
##            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.

```
# constancy of variance, ggplot1
ggplot(data=ready) +
     aes(x=yhat) +
     aes(y=estandard) +

     geom_point(size=0.5, pch=17) +                                    # pch=17 for shape (=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))  +   # RECOMMENDED: 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! In this ggplot, I plotted the standardize 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.

```
# test of constant variance (NULL: residual variance is constant)
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.

Partial F-Test. In question 2, where you assessed normality of Y=glucose in the predictor X=BMI, the loess smoother suggested that, possibly, the relationship of Y=glucose to body mass index might be modeled better as a quadratic, namely with two predictors: BMI and BMI^2. Create a new predictor that is BMI^2. Then, perform a partial F test of the null hypothesis that, controlling for linearity in BMI, there is no additional statistical significance in BMI^2 in explaining the variability in outcomes.

Q5. Partial F-test of BMI^2, controlling for BMI.

```
# Partial F for extra inclusion of BMI squared
ready$BMIsq <- ready$BMI^2                                    # create BMI squared
reduced <- lm(glucose ~ age + drinkany + BMI, data=ready)
full <- lm(glucose ~ age + drinkany + BMI + BMIsq, data=ready)
anova(reduced,full)                                           # partial F-test of hierarchical models

## Analysis of Variance Table
##
## Model 1: glucose ~ age + drinkany + BMI
## Model 2: glucose ~ age + drinkany + BMI + BMIsq
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    744 62375
## 2    743 62375  1  0.043192 0.0005 0.9819
```

Interpretation: Do NOT reject the null hypothesis that, controlling for BMI, BMI^2 is NOT statistically significant in explaining the variability in Y=glucose (p-value = .98). This confirms what we suspected, given what we saw in the graph (question 2), where the number of observations producing a slight curvature was very small.

#6.

Multicollinearity and the assessment of variance inflation. ***Multicollinearity*** is said to be present when the predictors are themselves linearly related. While some multicollinearity might be reasonably expected, if it is too extensive, each predictor on its own possesses too little independent information for the prediction of outcome. The result is regression coefficients with very large variances, or variance inflation. A measure of this is the variance inflation factor statistic, **VIF**. Briefly, to obtain the VIF for a particular predictor, that predictor is regressed on all the other predictors "i" 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!):

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

By any means you like, produce a table of VIF values for the 3 predictors in the model you fit in exercise #1.

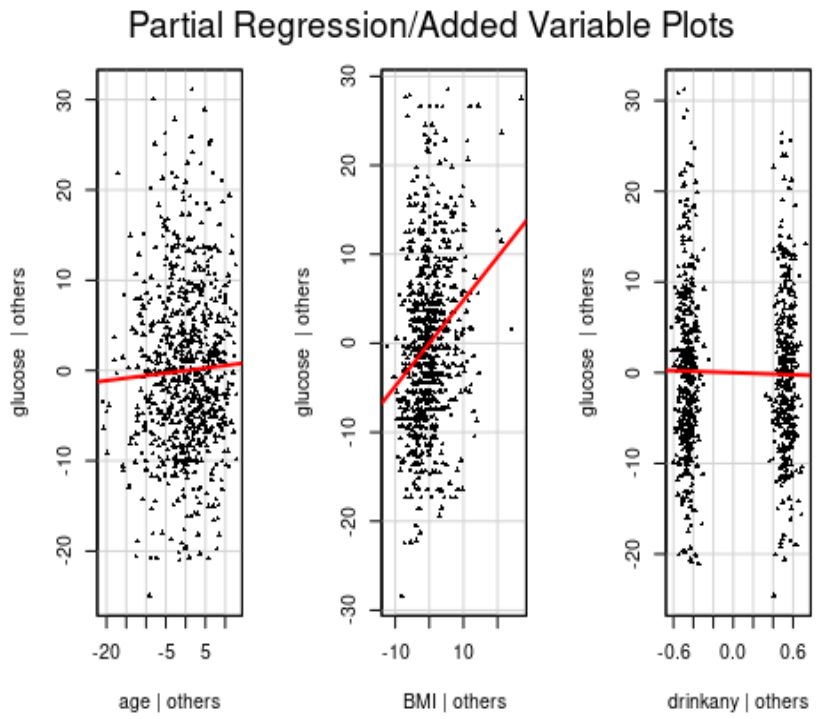
```
Q6. Variance Inflation Factor (VIF).
library(car)                                                    # vif() in package {car}
# Variance inflation factors (VIF).  Look for VIF < 10.
vif(fit)

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

Interpretation: All the VIF are much less than 10 which suggests that we do not need to worry about multicollinearity.

#7.

Partial regression plots (also called "added variables plot"). In a partial regression/added variable plot, the extra significance of a new predictor, controlling for the variables already in the model is examined by plotting the residuals of the new predictor on the control variables on the horizontal axis versus the residuals of Y on the control variables on the vertical axis. In this way, the influence of the control variables is "adjusted out". The slope of the scatter in this plot is a visual of the adjusted slope that will be obtained for the new predictor upon its inclusion in the model. Nice!

```
Q7. Partial Regression/Added variable Plot
library(car)                                                    # avPlots in {car}
# Partial Regression/Added variables plots                                          .
avPlots(fit,
        id=FALSE,                                               # suppress id's of extreme residuals
        pch=17,                                                 # pch= for plotting character (17=diamond)
        cex=0.5,                                                # cex= for point size
        col.lines="red",                                        # col.lines= for color of lines
        main="Partial Regression/Added Variable Plots",
        layout=c(1,3))                                          # c(#rows, # columns), here 1 row, 3 cols
```



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.

#8.

Model misspecification. ***Model misspecification*** can occur in a variety of ways; e.g., if some 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. By any means you like, perform the Ramsey test for the model you fit in exercise #1.

```
# Test for model misspecification (NULL: model is adequately specified)
library(lmtest)                                                    # resettest() in package {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).

#9.

Outliers. ***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. By any means you like, assess the model you fit in exercise #1 with respect to outliers.

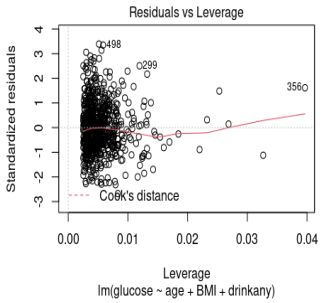
```
Q9. Regression diagnostics, case related. Outliers
library(car)                                                    # outlierTest() in package {car}

# 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: Take another look at the ggplot on page 7 (X=predicted outcome, Y=studentized residual). The graph shows 3 observations (or so) with studentized residuals more than 3 standard deviations from their expected value of 0. Sure enough, here we see that the largest is equal to 3.42. In a t-test that does NOT adjust for multiple comparisons, the p-value is highly statistically significant (p-value=.0007). After adjustment for multiple comparisons, it is no longer statistically significant. In my opinion, the graph on page 7 is more useful than this outlier test.

#10.

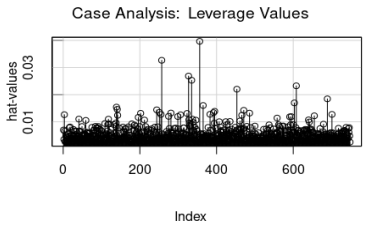
High leverage observations. ***High leverage observations*** are observations that are unusual in the X-sense. They may or may not influence the fitted model. By any means you like, assess model you fit in exercise #1 with respect to leverage.

```
Q10. Regression diagnostics, case related. Leverage
library(car)                                                    # influenceIndexPlot() in package {car}

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



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. Still, it's good to take a look.

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

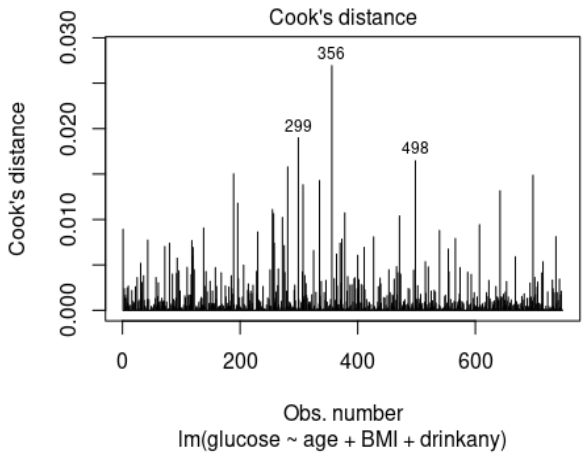


Interpretation: Take a look at the Y-axis. The values of the leverage are all much much less than 1. We have nothing to worry about vis a vis leverage!

#11.

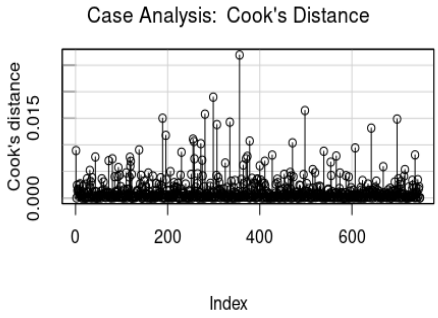
Influential observations. ***Influential observations*** do impact the fit! Their inclusion in the model changes the estimated betas. There are several approaches to detect influential observations. Among the most commonly used is the calculation of ***Cook's distance***. Briefly, the Cook's distance 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. My suggested guidelines are these: (1) 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); (2) A Cook's distance > 1 is worth exploring further; (2) A Cook's distance > .5 is of mild interest. By any means you like, construct a plot of Cook's distances for the model you fit in exercise #1.

```
# cook's distance, basic plot
plot(fit, which = 4)                # which=4 X=observation # 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!

```
# 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.

Good to know – Single commands for bundles of diagnostics.

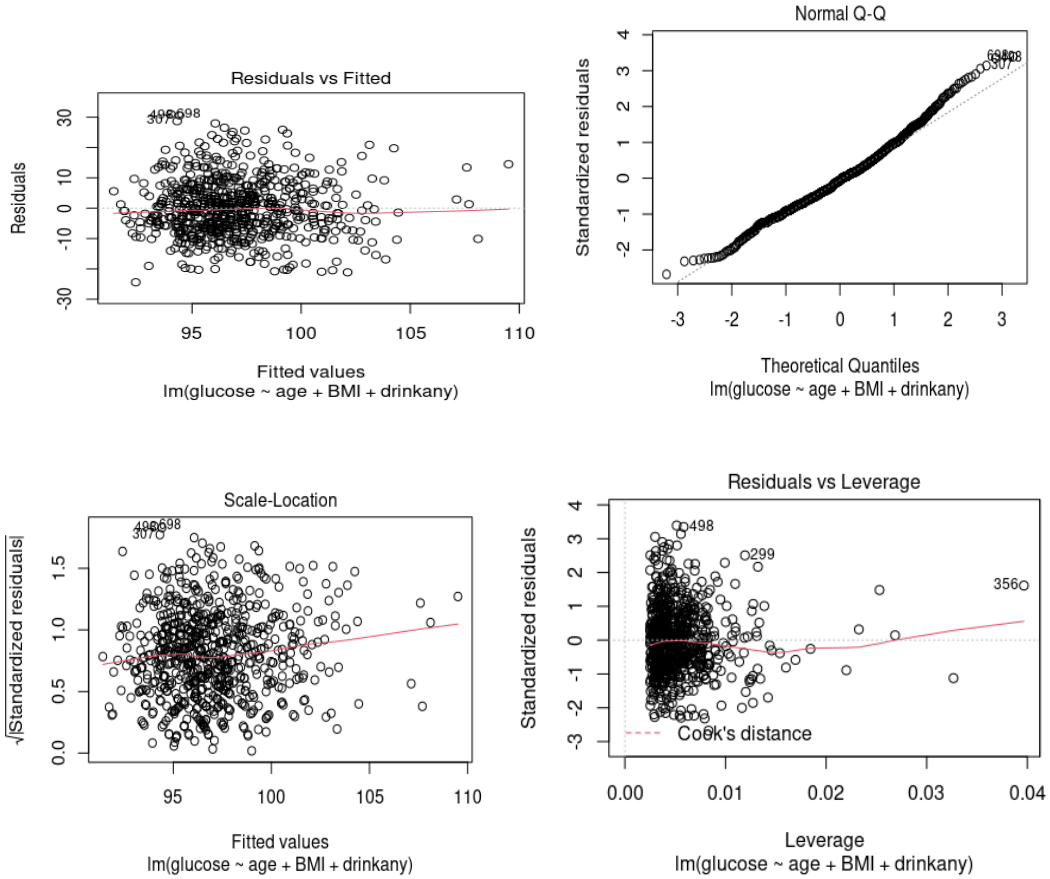
There exist several commands which will produce several diagnostic plots all at once. How convenient is that!. Here are some examples. You're welcome.

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

Bundles of diagnostics with single commands

```
library(car)
library(ggplot2)
library(ggfortify)

# Using base package
plot(fit)                                   # Plot(fit) produces four plots:  which=1, 2, 3, and 5
```

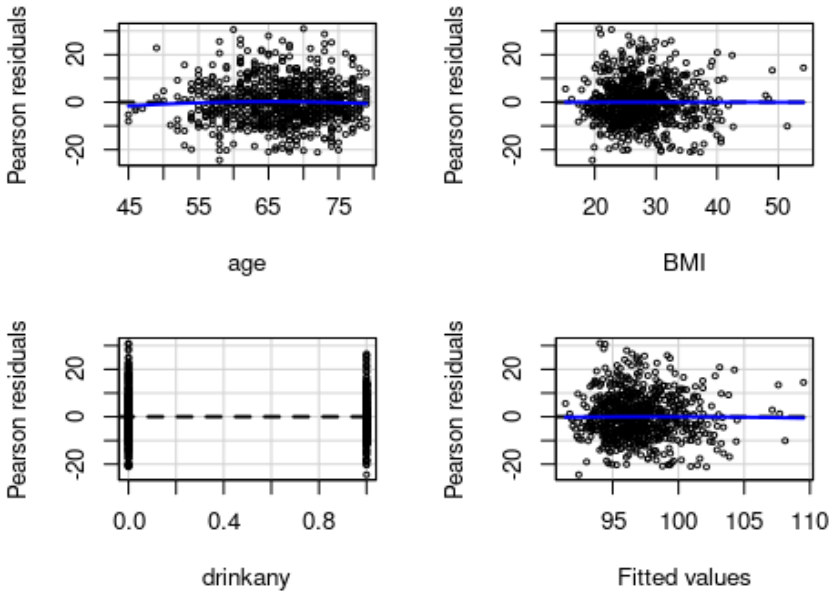


residualPlots( ) in package {car}

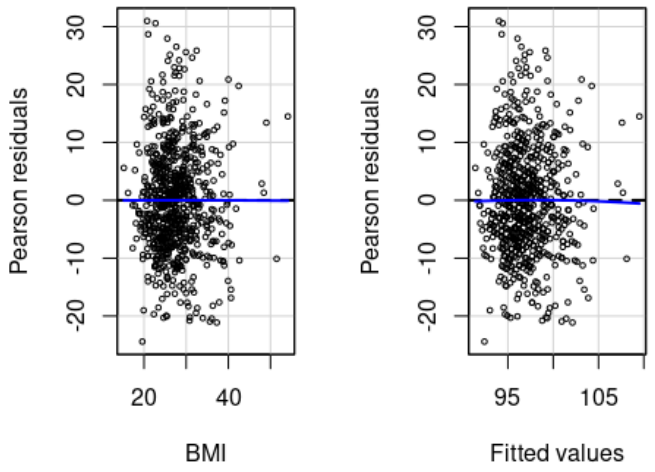
This command also provides, for each predictor X, a t-test of NULL: "no curvature" quadratic X^2 is not statistically significant. It also provides the Tukey test of NULL: "the model is additive"

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*

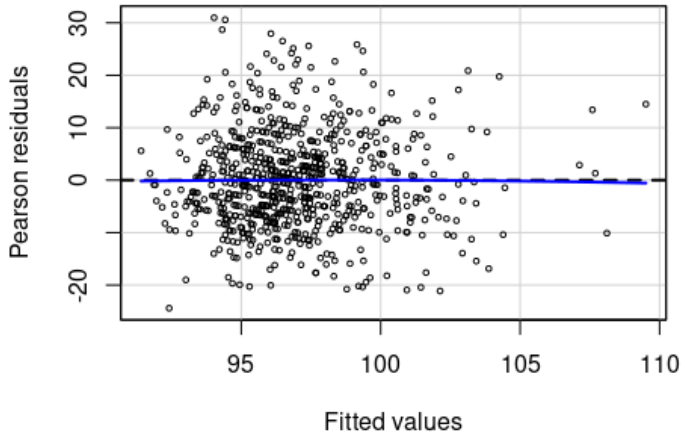
```
# Using package {car}.                                    # cex=.5 makes point size smaller
residualPlots(fit, cex=.5, fitted=TRUE)                   # all predictors
```




```
##            Test stat Pr(>|Test stat|)
## age          -0.7414           0.4587
## BMI          -0.0227           0.9819
## drinkany      0.8134           0.4163
## Tukey test   -0.1537           0.8778

residualPlots(fit, ~BMI, cex=.5, fitted=TRUE)             # just one predictor
```




```
##            Test stat Pr(>|Test stat|)
## BMI          -0.0227           0.9819
## Tukey test   -0.1537           0.8778

residualPlots(fit, ~1, cex=.5, fitted=TRUE)                    # fitted v residuals only
```

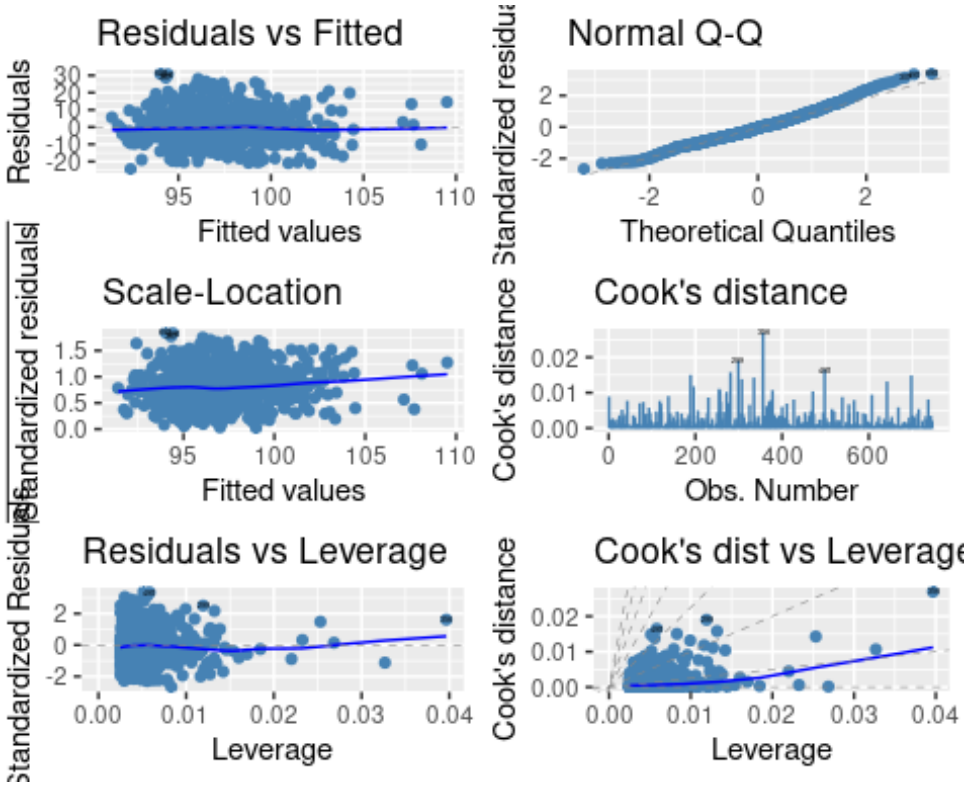



```
##            Test stat Pr(>|Test stat|)
## Tukey test   -0.1537           0.8778
```

autoplot( ) in package {ggpfortify}.
To be safe you might need to have library(ggplot2)

Command	**Plot Produced**
which=1	X = fitted value Y = residual
which=2	X = theoretical normal quantile Y = studentized residual
which=3	X = fitted value Y = square root (standardized residual)
which=4	X = observation number Y = Cook's Distance
which=5	X = leverage Y = standardized residual
which=6	X = leverage Y = Cook's Distance

```
# Using package {ggfortify}
autoplot(fit, which = 1:6, ncol = 2, label.size = 1,
         colour = "steelblue")
```



Last but not least!
How to Plot Predicted Means from a Fit, holding other covariates at their means

```
REPORTING Plot marginal prediction v age (other vars set to means)
library(ggplot2)

#1. Predicted mean Y v X=age with other vars set to their means
#1a.  Data for plot
newage <- data.frame(age=c(45,55,65,75),                                    # X = values of age
                     BMI=rep(mean(ready$BMI),4),                            # BMI at its mean
                     drinkany=rep(mean(ready$drinkany),4))                  # drinkany at its mean
yhat1 <- predict(fit, newdata=newage, interval="confidence")                # Yhat1 = predicted fit w CI
age <- newage$age                                                           # To obtain variable name
plotdata <- cbind(age, yhat1)                                               # cbind() will yield dataframe

# Get names of columns/variables in plotdata for use in ggplot
# names(plotdata)                                                           # Names are fit, lwr, upr

#1b.  Plot.
ggplot(data=plotdata) +                                                     # data=DATAFRAME, required
  aes(x=age, y=fit) +
  geom_point() +                                                            # Points are predicted means
  stat_smooth(method = lm, size=0.5, color="black") +
     geom_line(aes(y = lwr), color = "blue", linetype = "dashed") +  # lower CI
     geom_line(aes(y = upr), color = "blue", linetype = "dashed") +  # upper CI

   ggtitle("Predicted Mean Glucose (95% CI) by Age")  +
         xlab("Age, years") +
         ylab("Glucose, mg/dL")  +
         labs(caption = "Covariates bmi and drinkany at their means") +          # add a footnote!
         theme(plot.caption = element_text(hjust = 0, face = "italic"))          # position footnote lower left
```