Module 3 Worksheet

Prepared by: Katherine Daignault and the STA302H1 Fall 2025 Teaching Team

Worksheet Information

Goal of the worksheet:

The Module 3 worksheet is an opportunity to practice creating residual plots for a multiple linear model and interpreting patterns in the plots. By completing this worksheet, each student will be developing the skills to achieve the following weekly learning objectives:

- Recognize distinct patterns in appropriate residual plots to correctly conclude which assumption is violated.
- Detect situations in the data or population that increase the chance of violated assumptions.
- Create appropriate residual plots to evaluate model assumptions for a given dataset using software.
- Report the results and implications of a residual plot analysis.

This worksheet, in addition to the remainder of the class time, is **important practice for completing questions on the term test and final exam, and for your final project**.

Preparation assumed:

For hybrid sections: As part of the flipped design of the course, it is assumed that each student is attending this lecture having completed the following pre-class preparation:

- Watched the Module 3 Videos, attempted the Pre-Class Quiz, and accessed the code provided in the Guided Practice

For in person sections:

- Please complete this worksheet after attending your in-person lectures for each week. If you did not attend class, please review the annotated slides posted on Quercus which will be posted on each week's Quercus Page.
- Additional R/Coding resources are linked here.

How to complete this worksheet:

- Students may work in groups of 2-3 if desired. However **each student** must submit their worksheet to MarkUs to receive their completion credit. It is recommended that each student work on their **own copy** of the assignment.
- All the code and course knowledge needed to complete this worksheet has been provided in the pre-class materials. It may help to have these open while working on this document.
- Follow the instructions provided in each question to complete the code.
- DO NOT change the names of the variables that store your final answers.
- When in doubt about a question in the worksheet or your code, ask a TA or the instructor during office hours or on the discussion board.

Steps for submitting to MarkUs:

1. Go to MarkUs and log in using your UofT credentials.
2. Select Worksheet 2 from the assignment list.
3. Under Submissions, upload your Rmd file and select the file name from the list.
4. Go to Automated Testing and select Run Tests to check your worksheet answers.
5. You can submit as many times as you want, but only your latest submission before the deadline will be counted.
6. It is recommended you submit your file to MarkUs after you complete each activity to check your answers before moving on. You can submit multiple times to check your work, as your autograding tokens regenerate over time.

What to do if a test fails on MarkUs

1. Don't panic. Your work won't really be graded until the deadline, so start early to make sure you have lots of time to resolve issues before the deadline.
2. Read the message to get hints about what the problem is. For example "variable X not present" means that you may have a typo in your variable name.
3. Double check the instructions for each question to ensure you are entering an answer in the correct format.
4. Search on the discussion board to see if other classmates have encountered a similar error (and if not, consider posting a screenshot of the error message).
5. Come to TA or instructor office hours with your issue.

The due date for MarkUs Worksheet 3 is Tuesday, September 23, at 11:59pm

The NHANES survey data, available in R, is a very large dataset with many possible variables we could use as responses or predictors (for full details on the dataset, see https://cran.r-project.org/web/packages/NHANES/NHANES.pdf). For today, we will be using a subset of these data:

- only the 2009-2010 survey results
- results for adults aged 18 years or older.

We will study some of the demographic and lifestyle characteristics that affect the Body Mass Index (BMI) of individuals in the United States.

1) Load the dataset and subset

The variables we will use today are

- BMI: the body mass index of an individual as the response
- Age: age of the individual (rounded to an integer)
- Gender: Male or Female designation for gender (this survey predates when other options for this category were provided)
- Race1: a self-identified race with options White, Black, Hispanic, Mexican, Other
- HHIncomeMid: The midpoint of a household income bracket
- SleepHrsNight: Hours of sleep per night on weekdays, self-reported
- Smoke100n: Whether the individual has smoked more than 100 cigarettes in their life

We will use all variables except BMI as predictors. We will also keep track of the Survey Year to make sure we are only looking at 2009-2010, and the ID variable that gives each unique individual a numerical identifier.

To access the data, run the code chunk below:

```r
# accesses the package where the data lives
#install.packages("NHANES")
library(NHANES)

# loads the dataset into R
data(NHANES)

# shows how many rows and columns are in the dataset
dim(NHANES)
```

```
## [1] 10000    76
```

To subset the data to include *only* the variables we wish to use and *only* the individuals that meet our inclusion criteria (i.e. 2009-2010 survey and 18 years or older), you will replace `NULL` in each line with the subsetting command `subset()`.

To use `subset()`, follow this structure: `subset(dataset to be subsetted, condition of rows to keep)`. You may need to cast your data to a datafrane first before using `subset()`. For our goal, use these filtering conditions:

1. subset ‘new2‘ to include only the value of ‘SurveyYr‘ equal to the level 2009-2010
2. subset ‘new‘ to include only individuals with ‘Age‘ greater or equal to 18

Lastly, replace `NULL` in `dim1` and `dim2` with the vector produced by the function `dim(data)`, which displays the number of rows and columns in `data`.

```r
# removes all other variables that are not those we will use
new1 <- subset(as.data.frame(NHANES),
               select = c(ID, SurveyYr, Gender, Age, Race1,
                          HHIncomeMid, Smoke100n, SleepHrsNight, BMI))

# removes any row that contains a missing value
new1 <- new1[complete.cases(new1),]

# restrict to survey year 2009-2010 using condition SurveyYr=="2009_10"
table(new1$SurveyYr)
```

```
##
## 2009_10 2011_12
##    3283    3276
```

```r
new2 <- NULL
dim1 <- NULL

# restrict new2 to adults 18 years and up using condition Age>=18
new <- NULL
dim2 <- NULL
```

2) Fit a multiple linear model

Now, fit the multiple linear regression model using BMI as response, and Gender, Age, Race1, HHIncomeMid, Smoke100n, and SleepHrsNight as predictors by replacing `NULL` in the variable `og_model` with the code needed to create the model described above. Be sure to specify that you are using the dataset `new` that we created in Task 1.

Finally, replace `NULL` in the variable `intercept` with the estimated intercept from `og_model`, rounded to the nearest integer (if using `coef(model)` to extract the intercept, you need to embed it inside `as.numeric()` to turn the output into a number).

```r
# fit the requested model
og_model <- NULL

# replace NULL with the intercept value of the model (rounded to the nearest integer)
intercept <- NULL
```

Question: What is the interpretation of this value in the context of the data? Does this value have any real-life significance?
TYPE YOUR ANSWER BELOW:

3) Check the assumptions of this model

To begin, create the Residual vs Fitted Value plot by replacing the `NULL` values in `y_value` and `x_value` with the values to be plotted on the y and x axes respectively. You'll want to use `fitted()` and `resid()` to extract the information from `og_model`. This information will be used to make the residual plot.

Be sure to use the functions fitted() and resid() for this!

```r
# replace NULL with appropriate values to be used in y and x axes of plot respectively.
y_value <- NULL
x_value <- NULL

# plots the residual vs fitted plot
# plot(x = x_value, y = y_value, main="Residual vs Fitted", xlab="Fitted", ylab="Residuals")
```

Next, we create a plot of Residuals vs each predictor, focusing on only the quantitative predictors. The code below creates these plots and uses your `y_value` from before, so double check that this corresponds to your residuals. Simply uncomment (i.e. delete the #s) the plot code and run to view the plots.

```
# use this command to plot these in a single grid
par(mfrow=c(1,3))

# uncomment below code to run plots
# plot(x = new$Age, y = y_value, main="Residual vs Age", xlab="Age", ylab="Residual")
# plot(x = new$HHIncomeMid, y = y_value, main="Residual vs Household Income", xlab="Household income", ylab="R
# plot(x = new$SleepHrsNight, y = y_value, main="Residual vs Hours Slept", xlab="Hours Slept", ylab="Residual"
```

Thirdly, we check assumptions involving categorical predictors using boxplots. The code below creates these plots and uses your `y_value` from before, so double check that this corresponds to your residuals. Simply uncomment the plot code and run to view the plots.

```
# make another 1 by 3 grid
par(mfrow=c(1,3))

# uncomment below code to run plots
# boxplot(y_value ~ new$Gender , main="Residual vs Gender", xlab="Gender", ylab="Residuals")
# boxplot(y_value ~ new$Race1 , main="Residual vs Race", xlab="Race", ylab="Residuals")
# boxplot(y_value ~ new$Smoke100n , main="Residuals vs Smoke 100 cigs", xlab=">100 Cigarettes Smoked", ylab="R
```

Finally, create your Normal QQ plots by using the function `qqnorm()` and then adding the diagonal line with `qqline()`. These functions will need you to input the residuals of `og_model`. You should use the `resid()` function to extract these.

```
# use qqnorm() and qqline() with residuals to display QQ plot
# qqnorm(...)
# qqline(...)
```

Question: What patterns, if any, do you see in these plots?
TYPE YOUR ANSWER BELOW:

4) Check the additional conditions

In order for your conclusion above to be valid, you must further confirm that no other functional relationships among variables are causing these patterns to occur.

We can check condition 1 first, by making a scatterplot of our response versus fitted values. If we observe random scatter around the diagonal or a simple curving pattern, we can say this condition is satisfied.

Replace `NULL` in both `y_value2` and `x_value2` with the response and fitted values needed to create the plot. Then uncomment the plot code to run. Remember that our dataset was called `new`, our model was called `og_model`, and we can extract fitted values using `fitted()`.

```
# replace NULL with the variables to be displayed on the x and y axis
y_value2 <- NULL
x_value2 <- NULL

# uncomment the code below to see plot
# plot(x = x_value2, y = y_value2, main="Response vs Fitted", xlab="Fitted", ylab="BMI")
# abline(a = 0, b = 1, lty=2, col="red")
```

Lastly, run the below code to display the condition 2 plot, all pairwise plots between the predictors. The function `pairs()` is used and we restrict it to plot only the variables stored in columns 3 to 8 using `new[,3:8]`. Simply uncomment the plot code and run to view the plots.

```
# uncomment to run the code
# pairs(new[,3:8])
```

Question: Are your conclusions about model violations still valid?
TYPE YOUR ANSWER BELOW:

END OF WORKSHEET - BE SURE TO SUBMIT YOUR WORKSHEET ON MARKUS TO RECEIVE COMPLETION CREDIT