Unit 9
R for Normal Theory Regression

"Assume that a statistical model such as a linear model is a good first start only"

- Gerald van Belle

Normal theory regression analysis explores the relationship of ***one*** outcome that is continuous (e.g. Y = birth weight) with one or more predictors that can be continuous or discrete (e.g. X_1 = months gestation, X_2 = yes/no indicator of mother's smoking status, X_3 = mother's systolic blood pressure, and so on).

> In **simple linear regression**, the number of predictors is **one** and **continuous** (eg X=mother's systolic blood pressure).
>
> In **multiple linear regression**, the number of predictors is **two or more** and can be both **continuous and discrete**

The **goal** is to explain the variation in the outcomes (the Y variable) with a "good" model that is a function of the predictors (the X variables) that is as "small" as possible. The challenge is in how to achieve both "good" (close fit) and "small" (parsimony) simultaneously.

Ultimately, we don't know if our model is correct and most likely it is not. Nevertheless, a model that is "good" and "small" has a variety of **uses**:

> **Hypothesis Tests and Confidence Intervals**
> We can ask such questions as: "Is the experimental treatment is associated with a statistically significant benefit?"
>
> **Prediction**
> We can use the estimating equation to make confidence interval predictions such as: the survival time following surgery of a future patient undergoing coronary bypass surgery.
>
> **Insights into Nature**
> Sometimes, the fitted model derives from a physical-equation. An example is Michaelis-Menton kinetics. A Michaelis-Menton model is fit to the data for the purpose of estimating the actual rate of a particular chemical reaction.

Table of Contents

Learning Objectives

When you have finished this unit, you should be able to:

- **Define** the simple and multiple linear regression models;
- State and explain the **assumptions** for normal theory linear regression analysis;
- Use R to **explore a data set** (numerical descriptions, scatterplots, etc) prior to model estimation;
- Use R to **create design variables** for use in the modeling of categorical explanatory variables;
- Use R to **fit (estimate)** a normal theory regression model;
- **Interpret a fitted model**, including the regression coefficients, standard errors, R^2, sums of squares, analysis of variance, t-tests, and F-tests;
- Explain **confounding** and **effect modification**;
- Use R to **assess confounding and modification** in a normal theory regression;
- Use R to perform **hypothesis tests** and obtain **confidence intervals**;
- Use R to produce **post-estimation graphical summaries** of model fit;
- Use R to perform **regression diagnostics** to assess model adequacy for a normal theory regression; and
- **Write a 1-2 paragraph interpretation** of a normal theory regression analysis.

Packages Used (one-time installation):

```
- stargazer
- summarytools
- Hmisc
- ggplot2
- GGally
- psych
- car
- lmtest
- tidyverse
- gtools
```

1. Introduction

1.1 Settings Where Regression Might Be Considered

Example #1
Are Emergency Calls to the New York Auto Club Related to the Weather?

Source:
Chatterjee, S; Handcock MS and Simonoff JS A Casebook for a First Course in Statistics and Data Analysis. New York, John Wiley, 1995, pp 145-152.

Are calls to the New York Auto Club related to the weather, with more calls occurring during bad weather? To explore this possibility, the NY Auto Club obtained observations on numbers of calls to the New York Auto Club (Y=calls) together with several kinds of information about the weather on the day of the call. Among the analyses they performed was a **simple linear regression** with outcome (dependent) variable Y and predictor (explanatory) variable X, both continuous, defined:

Y = calls (number of calls)
X = low (the lowest temperature of the day).

Dear reader: Strictly speaking, the variable Y=calls is discrete, not continuous. In this example, however, the sample size was large and the distribution of calls was approximated well with the assumption of normality. So, the normal theory linear regression went forward!

Example #2
Does the expression of p53 change with parity and age?

Source:
Matthews et al. Parity Induced Protection Against Breast Cancer 2007.

P53 is a human gene that is a tumor suppressor gene. Malfunctions of this gene have been implicated in the development and progression of many cancers, including breast cancer. Matthews et al were interested in exploring the relationship of Y=p53 expression to parity and age at first pregnancy, *after adjustment for* other, established, risk factors for breast cancer, including: age at first mensis, family history of breast cancer, menopausal status, and history of oral contraceptive use.

- Among the initial analyses, a **simple linear regression** might be performed to obtain a thorough understanding of the relationship of p53 expression and age. Both the outcome (Y) and the predictor (X) are continuous.

  Y = p53 expression
  X = Age

- A **multiple linear regression** might then be performed to see if age and parity *retain* their predictive significance, *after controlling for* the other, known, risk factors for breast cancer. Thus, the analysis would consider one outcome variable (Y) and 6 predictor variables (X_1, X_2, X_3, X_4, X_5, X_6):

  Y = p53
  X_1 = Age
  X_2 = Parity
  X_3 = Age at first mensis
  X_4 = Family history of breast cancer
  X_5 = Menopausal status
  X_6 = History of oral contraceptive use

Example #3
Does Air Pollution Reduce Lung Function?

Source:
Detels et al (1979) The UCLA population studies of chronic obstructive respiratory disease. I. Methodology and comparison of lung function in areas of high and low pollution. Am. J. Epidemiol. 109: 33-58.

Detels et al (1979) investigated the relationship of lung function to exposure to air pollution among residents of Los Angeles in the 1970's. Baseline and follow-up measurements of exposure and lung function were obtained. Also obtained were measurements of selected other variables that the investigators suspected might confound or modify the effects of pollution on lung function: age, sex, height, weight, etc. Afifi, Clark and May (2004) consider portions of this data in their 2004 text, *Computer-Aided Multivariate Analysis, Fourth Edition* (Chapman & Hall)

- It is already known that a person's FEV is related to their height. Thus, an analysis of the effects of air pollution might begin with a **simple linear regression** analysis of the relationship between FEV and height before moving on to an examination of the effects of exposure to air pollution:

  Y = FEV, liters
  X = Height, inches

- A **multiple linear regression** might then be performed to determine the nature and strength of exposure to pollution for the prediction of lung function, *taking into account* the role of height and other influences on lung function, such as age, smoking, etc. For example, the relationship of lung function to exposure to air pollution might be different for smokers and non-smokers; this would be an example of effect modification (interaction). It might also be the case that the relationship of lung function to exposure to air pollution is confounded by height. Here, we would have something like:

  Y = FEV, liters
  X_1 = Exposure to air pollution
  X_2 = Height, inches
  X_3 = Smoking (1=yes, 0=no)

Example #4
Exercise and Glucose for the Prevention of Diabetes

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 Study. JAMA 280(7): 605-13.

In the HERS study, Hulley et al. (1998) sought to determine if exercise, a modifiable behavior, might lower the risk of diabetes in non-diabetic women who were at risk of developing the disease. The question is a complex one because there are many risk factors for diabetes. Moreover, the type of woman who chooses to exercise may be related in other ways to risk of diabetes, apart from the fact of her exercise habit. For example, women who exercise regularly are typically younger and have lower body mass index (BMI); these characteristics also confer a risk benefit with respect to diabetes. Finally, the benefit of exercise may be mediated through a reduction of body mass index. Vittinghoff, Glidden, Shiboski and McCullogh (2005) consider portions of this data in their 2005 text, *Regression Methods in Biostatistics: Linear.Logistic, Survival and Repeated Measures Models* (Springer).

- A **multiple linear regression** was performed to assess the benefit of exercising at least three times/week, compared to no exercise, on blood glucose, after controlling for other factors associated with blood glucose levels. Thus, here we would have something like:

  Y = Glucose, mg/dL
  X_1 = Exercise (1=yes if 3x/week or more, 0 = no)
  X_2 = Age, years
  X_3 = Body Mass Index (BMI)
  X_4 = Alcohol Use (1=yes, 0=no)

1.2 Review - What is Statistical Modeling

George E.P. Box, a very famous statistician, once said, ***"All models are wrong, but some are useful."*** Incorrectness notwithstanding, we do statistical modeling for a very good reason: we seek an understanding of the natures and strengths of the relationships (if any) that might exist in a set of observations that co-vary.

For any set of observations, theoretically, lots of models are possible. So, how to choose? The **goal** of statistical modeling is to obtain a model that is simultaneously **minimally adequate** and a **good fit**. **The model should also make sense.**

Minimally adequate

- Each predictor is "important" in its own right
- Each extra predictor is retained in the model only if it yields a significant improvement (in fit and in variation explained).
- The model should not contain any redundant parameters.

Good Fit

- The amount of variability in the outcomes (the Y variable) explained is a lot
- The outcomes that are predicted by the model are close to what was actually observed.

The model should also make sense

- A preferred model is one based on "subject matter" considerations
- The preferred predictors are the ones that are simply and conveniently measured.

It is *not* possible to choose a model that is simultaneously minimally adequate and a perfect fit. Model estimation and selection must achieve an appropriate balance.

1.3 A General Approach for Model Development

There are ***no*** rules ***nor a single best strategy***. Different study designs and research questions call for different approaches for model development. ***Tip –*** Before you begin model development, make a list of your study design, research aims, outcome variable, primary predictor variables, and covariates.

As a general suggestion, the following approach has the advantages of providing a reasonably thorough **exploration of the data and relatively little risk of missing something important**

Preliminary – Be sure you have: (1) checked, cleaned and described your data, (2) screened the data for possible associations, and (3) thoroughly explored the bivariate (also called "single predictor", "unadjusted", "crude") relationships.

Step 1 – Fit the "maximal" model.
The maximal model is the large model that contains all the explanatory variables of interest as predictors. This model also contains all the covariates that might be of interest. It also contains all the interactions that might be of interest. Note the amount of variation explained.

Step 2 – Begin simplifying the model.
Inspect each of the terms in the "maximal" model with the goal of removing the predictor that is the least significant. Drop from the model the predictors that are the least significant, beginning with the higher order interactions (**Tip** -interactions are complicated and we are aiming for a simple model). Fit the reduced model. Compare the amount of variation explained by the reduced model with the amount of variation explained by the "maximal" model.

If the deletion of a predictor has little effect on the variation explained
Then leave that predictor out of the model.
And inspect each of the terms in the model again.

If the deletion of a predictor has a significant effect on the variation explained
Then put that predictor back into the model.

Step 3 – Keep simplifying the model.
Repeat step 2, over and over, until the model remaining contains nothing but significant predictor variables.

Beware of some important caveats

- Sometimes, you will want to keep a predictor in the model regardless of its statistical significance (an example is randomization assignment in a clinical trial)
- The order in which you delete terms from the model matters
- You still need to be flexible to considerations of biology and what makes sense.

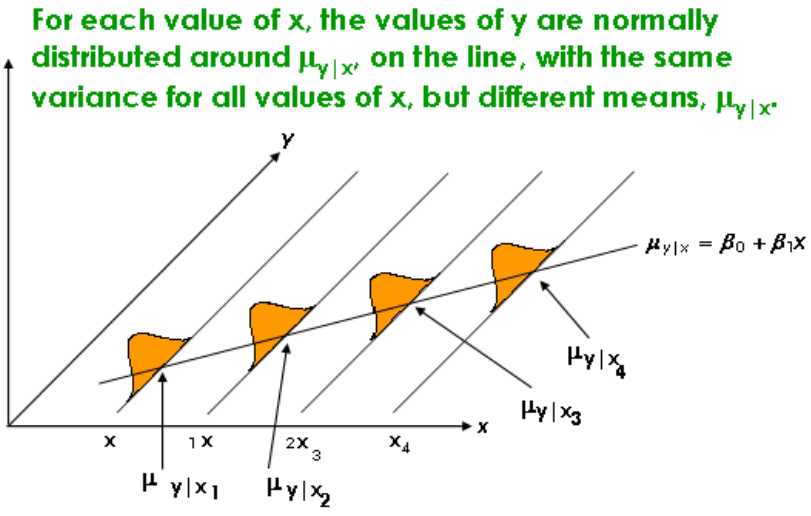
1.4 Review - Normal Theory Regression

Normal theory regression analysis is used to investigate possibly complex relationships when:

- The outcome is a **single continuous variable (Y)** that can reasonably be assumed to be **distributed normal**; and
- The outcome is potentially related to possibly **several predictor variables (X_1, X_2, …, X_p)** which can be **continuous or discrete**; and
- Some of the predictor variables might **confound** the prediction role of other explanatory variables; and
- Some of the predictor-outcome relationships may be different (are modified by) depending on the level of one or more different predictor variables (**interaction**)

Simple Linear Regression:

A simple linear regression model is one for which the mean μ (the average value) of **one continuous, and normally distributed, outcome** random variable Y (e.g. **Y= FEV**) varies linearly with changes in **one continuous predictor** variable X (e.g. **X=Height**). It says that the expected values of the outcome Y, as X changes, lie on a straight line ("regression line").



Assumptions

1. The outcomes $Y_1, Y_2, \cdots, Y_n$ are **independent**.

2. The values of the predictor variable X are fixed and measured without error.

3. At each value of the predictor variable X=x, the distribution of the outcome Y is **normal** with

$$\text{mean} = \mu_{Y|X=x} = \beta_0 + \beta_1 x$$
$$\text{variance} = \sigma_{Y|x}^2.$$

Model

These assumptions mean that we are considering the following **model**. For individual "i",

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \text{ where}$$

1. The errors $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$ are **independent**.

2. Each error ε_i is distributed is **normal** with

$$\text{mean} = 0$$
$$\text{variance} = \sigma_{Y|x}^2.$$

Multiple Linear Regression:

In multiple linear regression, there is still just **one outcome variable, continuous**. The term "multiple" refers to there being **more than one predictor variable**.

Definition

A multiple linear regression model is a particular model in which the mean μ of **one continuous** outcome random variable Y (e.g. **Y= FEV**) varies linearly with changes in two or more predictor variables X_1, X_2, etc. (e.g. **X_1=Height, X_2 = Smoking (1=yes, 0 = no)**. The predictor variables can be continuous, discrete, or both. A multiple linear regression model says that the expected values (μ) of the outcome Y, as X_1, X_2, etc change, lie on a plane ("regression plane").

Assumptions

The assumptions required are an extension of those for simple linear regression.

1. The outcomes $Y_1, Y_2, \cdots, Y_n$ are **independent**.
2. The values of the predictor variables $X_1 \cdots X_p$ are fixed and measured without error.
3. For each fixed profile of values, $x_1, x_2, \ldots, x_p$, of the p predictor variables $X_1 \cdots X_p$ (written using vector notation $\underline{X}=\underline{x}$), the distribution of values of Y is **normal** with

$$\text{mean} = \mu_{Y|\underline{X}=\underline{x}} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$\text{variance} = \sigma^2_{Y|\underline{X}=\underline{x}}.$$

Model

Our model is now:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

- p = # predictors, ***apart*** from the intercept
- Each $X_1 \cdots X_p$ can be either discrete or continuous.

2. Example of R to Perform Normal Theory Regression

How to follow along:
Download from the course website.
framingham_1000.Rdata

Source:
Levy (1999) National Heart Lung and Blood Institute. Center for Bio-Medical Communication. Framingham Heart Study

Description:
Cardiovascular disease (CVD) is the leading cause of death and serious illness in the United States. In 1948, the Framingham Heart Study - under the direction of the National Heart Institute (now known as the National Heart, Lung, and Blood Institute or NHLBI) was initiated. The objective of the Framingham Heart Study was to identify the common factors or characteristics that contribute to CVD by following its development over a long period of time in a large group of participants who had not yet developed overt symptoms of CVD or suffered a heart attack or stroke.

Here we will use a subset of the data comprised of information on 9 variables in a subset of n=1000.
Note – some of the variables shown here will be created in the pages that follow.

Variable	Label	Codings
sbp	Systolic Blood Pressure (mm Hg)	
ln_sbp	Natural logarithm of sbp	ln_sbp=ln(sbp)
age	Age, years	
bmi	Body Mass index (kg/m^2)	
ln_bmi	Natural logarithm of bmi	ln_bmi=ln(bmi)
sex	Sex at birth	1=male 2=female
female	Female Indicator	0 = male 1 = female
scl	Serum Cholesterol (mg/100 ml)	
ln_scl	Natural logarithm of scl	ln_scl=ln(scl)

Multiple Regression Variables:
Outcome Y = ln_sbp
Predictor Variables: ln_bmi, ln_scl, age, sex

Research Question:
From among these 4 "candidate" predictors, what are the important "risk" factors and what is the nature of their association with Y=ln_sbp?

The basic steps in this illustration are the following and correspond to the general approach to model development introduced on page 8.

Plan of Illustration

Step 1 – Exploratory Data Analysis, Indicator Variables, and Interactions.
Examine descriptive statistics, assess normality of the dependent variable, consider a "normalizing" transformation if needed, create indicator variables, create interaction variables

Step 2 – Examine Bivariate Relationships.
Look at the relationship of the dependent variables (Y) with each of the candidate predictor variables (X). Look at these relationships graphically and test correlations. Consider transformations of the predictor variables if needed.

Step 3 – Fit Models and Choose "Tentative" Final Model.
Fit an initial model. Fit alternative models. Compare competing models with partial F-tests and side-by-side comparisons of estimated regression coefficients, percent variance explained (R-squared), and mean squared error. Choose a "tentative" final model.

Step 4 – Regression Diagnostics.
Fit again the "tentative" final model; this is a necessary preliminary to doing most regression diagnostics. Check model assumptions. Check model adequacy.

Step 5 – Repeat steps #3 and #4 as needed.

Step 6 – Report Regression Results.
Produce appropriate tabulations of regression results. Produce graphical summaries of the "final" model. Interpret.

Step 1 – Exploratory Data Analysis, Indicator Variables, and Interactions.

Examine descriptive statistics, assess normality of the dependent variable, consider a "normalizing" transformation if needed, create indicator variables, create interaction variables.

```
# NOTE: In the comments, I will use { } to denote the package that contains the command being used.

options(scipen=1000)                        # turn off scientific notation
rm(list=ls())                               # clear the decks
setwd("/Users/cbigelow/Desktop/")           # set the working directory )
load(file="framingham_1000.Rdata")          # load data (this code assumes data is in working directory)

framingham <- framingham_1000               # would rather work with shorter dataframe name - cb

summary(framingham)                         # {base} summary( ) to examine descriptive statistics
```

```
##     sex           sbp             scl             age
##  Men  :443   Min.   : 80.0   Min.   :115.0   Min.   :30.00
##  Women:557   1st Qu.:116.0   1st Qu.:197.0   1st Qu.:38.75
##              Median :128.0   Median :225.0   Median :45.00
##              Mean   :132.3   Mean   :227.8   Mean   :45.92
##              3rd Qu.:144.0   3rd Qu.:255.0   3rd Qu.:53.00
##              Max.   :270.0   Max.   :493.0   Max.   :66.00
##                              NA's   :4
##       bmi              id           ln_bmi          ln_sbp
##  Min.   :16.40   Min.   :   1   Min.   :2.797   Min.   :4.382
##  1st Qu.:23.00   1st Qu.:1246   1st Qu.:3.135   1st Qu.:4.754
##  Median :25.10   Median :2488   Median :3.223   Median :4.852
##  Mean   :25.57   Mean   :2410   Mean   :3.230   Mean   :4.872
##  3rd Qu.:27.80   3rd Qu.:3605   3rd Qu.:3.325   3rd Qu.:4.970
##  Max.   :43.40   Max.   :4697   Max.   :3.770   Max.   :5.598
##  NA's   :2                      NA's   :2
##      ln_scl
##  Min.   :4.745
##  1st Qu.:5.283
##  Median :5.416
##  Mean   :5.410
##  3rd Qu.:5.541
##  Max.   :6.201
##  NA's   :4
```

```
library(stargazer)
stargazer::stargazer(framingham, type="text", median=TRUE)  # {stargazer} stargazer( ) for tidy descriptives
```

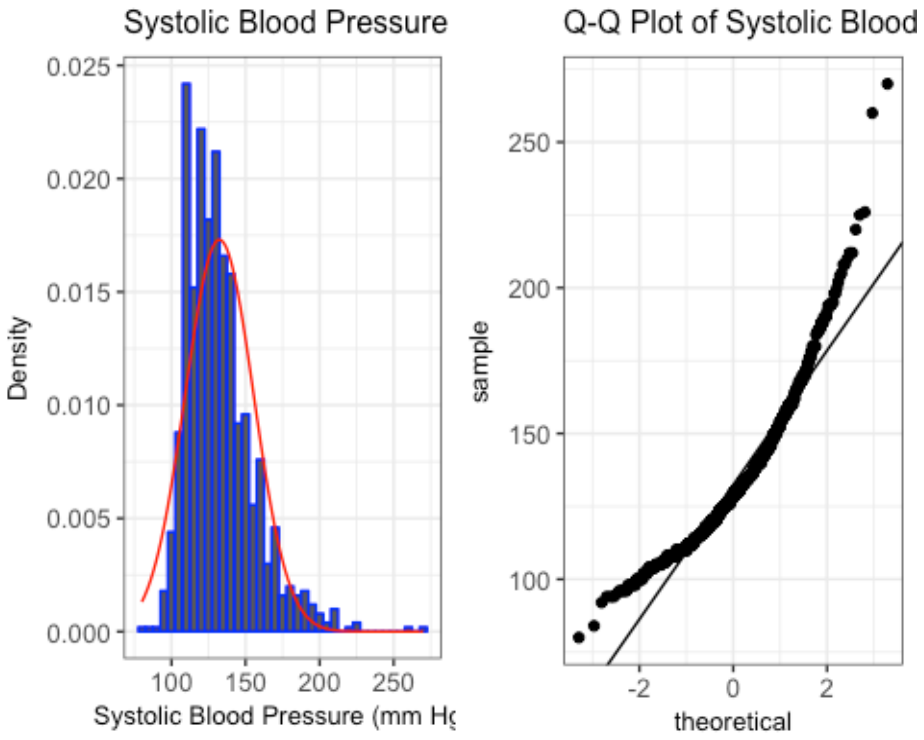
```
##
## ==========================================================
## Statistic   N     Mean    St. Dev.   Min   Median   Max
## ----------------------------------------------------------
## sbp       1,000  132.350   23.043     80     128    270
## scl        996   227.846   45.087    115     225    493
## age       1,000  45.922     8.545     30     45      66
## bmi        998   25.566     3.848   16.400 25.100  43.400
## id        1,000 2,410.031 1,363.439   1    2,487.5 4,697
## ln_bmi     998    3.230     0.147   2.797   3.223  3.770
## ln_sbp    1,000   4.872     0.163   4.382   4.852  5.598
## ln_scl     996    5.410     0.195   4.745   5.416  6.201
## ----------------------------------------------------------
```

```
library(summarytools)
summarytools::freq(framingham$sex)            # {summarytools} freq( ) for discrete vars

## Frequencies
## framingham$sex
## Type: Factor (unordered)
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##         Men    443     44.30          44.30     44.30          44.30
##       Women    557     55.70         100.00     55.70         100.00
##        <NA>      0                               0.00         100.00
##       Total   1000    100.00         100.00    100.00         100.00


# Prior to choosing Y=ln_sbp, I considered modeling as my Y the variable sbp
library(summarytools)
summarytools::descr(framingham$sbp, stats = c("n.valid","mean", "sd", "min","q1", "med", "q3", "max","CV"),
transpose = TRUE)

## Descriptive Statistics
## framingham$sbp
## N: 1000
##
##             N.Valid     Mean   Std.Dev     Min       Q1   Median       Q3      Max     CV
## --------- --------- -------- --------- ------- -------- -------- -------- -------- ------
##       sbp   1000.00   132.35     23.04   80.00   116.00   128.00   144.00   270.00   0.17


shapiro.test(framingham$sbp)                    # I tested normality of sbp (Null:  Distribution is normal)
##
##  Shapiro-Wilk normality test
##
## data:  framingham$sbp
## W = 0.92121, p-value < 0.00000000000000022
```

Interpretation: The null hypothesis of normality of the distribution of sbp is rejected (p << .00001)

```
# I also plotted the distribution of sbp (Goal: to see if it looked at least approximately normal)
library(ggplot2)
library(gridExtra)

# p1 = histogram w overlay normal
# ggplot(DATAFRAME, aes(x=VARIABLENAME)) + stuff below
p1 <- ggplot2::ggplot(data=framingham, aes(x=sbp)) +
        geom_histogram(binwidth=5, colour="blue",
                       aes(y=..density..)) +
        stat_function(fun=dnorm,
                      color="red",
                      args=list(mean=mean(framingham$sbp),
                                sd=sd(framingham$sbp))) +
        ggtitle("Systolic Blood Pressure (sbp)") +
        xlab("Systolic Blood Pressure (mm Hg)") +
        ylab("Density") +
        theme_bw() +
        theme(axis.text = element_text(size = 10),
             axis.title = element_text(size = 10),
             plot.title = element_text(size = 12))
```

```
# p2 = quantile-quantile plot
p2 <- ggplot2::ggplot(data=framingham, aes(sample=sbp)) +
          stat_qq() +
          geom_abline(intercept=mean(framingham$sbp),
               slope = sd(framingham$sbp)) +
          ggtitle("Q-Q Plot of Systolic Blood Pressure (sbp)") +
          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)
```



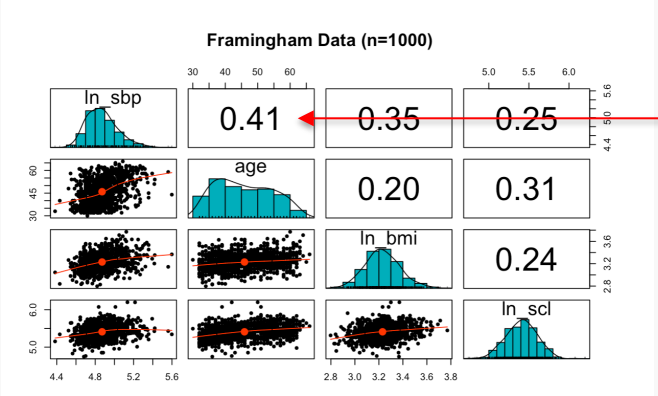
Interpretation: This confirms what the Shapiro-Wilk test suggests. The null hypothesis of normality of the distribution of sbp is not supported.
Interpretation: The distribution of Y=sbp departs from normality → confirming that we should consider a transformation.

ln_sbp
was used as the dependent variable

```
library(summarytools)
library(Hmisc)

framingham$female <- as.numeric(framingham$sex == "Women", na.rm=TRUE)   # Create 0/1 female from factor sex
summarytools::ctable(framingham$sex,framingham$female,prop = 'n', totals = FALSE)   # Check

## Cross-Tabulation
## Variables: sex * female
## Data Frame: framingham
##
## ------- -------- ----- -----
##           female     0     1
##     sex
##     Men            443     0
##   Women              0   557
## ------- -------- ----- -----

Hmisc::label(framingham$female) <- "female01"                            # Label variable

framingham$ageXfemale <- framingham$age*framingham$female                # Create interaction age x female
Hmisc::label(framingham$ageXfemale) <- "AGE x FEMALE interaction"        # Label interaction
framingham$lnsclXfemale <- framingham$ln_scl*framingham$female
Hmisc::label(framingham$lnsclXfemale) <- "ln(scl) x FEMALE interaction"
framingham$lnbmiXfemale <- framingham$ln_bmi*framingham$female
Hmisc::label(framingham$lnbmiXfemale) <- "ln(bmi) x FEMALE interaction"
```

Step 2 – Examine Bivariate Relationships.
Look at the relationship of the dependent variables (Y) with each of the candidate predictor variables (X). Look at these relationships graphically and test correlations. Consider transformations of the predictor variables if needed.

```
library(psych)
myvars <- c("ln_sbp","age","ln_bmi","ln_scl")
psych::pairs.panels(framingham[myvars],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE, # show correlation ellipses

             main="Framingham Data (n=1000)")
```



KEY:
0.41 = correlation(ln_sbp, age)

Create a dataset that is complete on all the variables of interest
Why: So that comparisons of models will be based on the same observations

```
complete <- na.omit(framingham, cols=c("ln_sbp", "ln_bmi", "age", "female", "lnbmiXfemale", "lnsclXfemale","ageXfemale"))
```

Step 3 – Fit Models and Choose "Tentative" Final Model.

Fit an initial model. Fit alternative models. Compare competing models with partial F-tests and side-by-side comparisons of estimated regression coefficients, percent variance explained (R-squared), and mean squared error. Choose a "tentative" final model.

```
# lm( ) to fit a linear regression model
# KEY: modelname <- lm(data=dataframe, YVAR ~ predictor1 + predictor2 + etc)
m_maximal <- lm(data=complete, ln_sbp ~ ln_bmi + ln_scl + age + female + lnbmiXfemale + lnsclXfemale + ageXfemale)
m_2 <- lm(data=complete, ln_sbp ~ ln_bmi + ln_scl + age + female + ageXfemale)
m_3 <- lm(data=complete, ln_sbp ~ ln_bmi)
m_4 <- lm(data=complete, ln_sbp ~ ln_scl)
m_5 <- lm(data=complete, ln_sbp ~ age + female + ageXfemale)


# {stargazer} stargazer() to display models side by side
stargazer::stargazer(m_maximal,m_2,m_3,m_4,m_5,type="text",font.size="small", align=TRUE, omit.stat=c("f", "ser"))
```

```
============================================================
                            Dependent variable:
              ----------------------------------------------
                                    ln_sbp
                (1)       (2)       (3)       (4)       (5)
------------------------------------------------------------
ln_bmi        0.304*** 0.271***  0.388***
              (0.055)   (0.032)  (0.033)

ln_scl         0.059    0.056**            0.211***
              (0.037)   (0.025)            (0.026)

age           0.004*** 0.004***                      0.004***
              (0.001)   (0.001)                      (0.001)

female         -0.011  -0.217***                    -0.327***
              (0.304)   (0.051)                      (0.051)

lnbmiXfemale   -0.051
              (0.067)

lnsclXfemale   -0.009
              (0.050)

ageXfemale    0.005*** 0.005***                      0.007***
              (0.001)   (0.001)                      (0.001)

Constant      3.396*** 3.521***  3.618*** 3.730*** 4.701***
              (0.234)   (0.159)  (0.106)  (0.139)   (0.039)

------------------------------------------------------------
Observations    994       994      994      994       994
R2             0.267     0.266    0.123    0.064     0.203
Adjusted R2    0.261     0.262    0.122    0.063     0.200
============================================================
Note:                            *p<0.1; **p<0.05; ***p<0.01
```

KEY:
.304 = beta for ln_bmi
.055 = SE(beta)
*** = "significant at type I error = .01"

KEY – Model (1):
Fitted Model:

$$\widehat{ln\_sbp} = 3.4 + 0.30*ln\_bmi + 0.06*ln\_scl + 0.003*age - 0.01*female - 0.05*lnbmi\_female - 0.009lnscl\_female + 0.005*age\_female$$

R2 = .267:
26.7% of the variability in Y=ln_sbp is explained by the fitted model

```
# anova(REDUCEDMODEL, FULLMODEL) to obtain partial F Tests
paste("Partial F-test, 2df:  Null:  lnbmiXfemale=0 lnsclXfemale=0")
anova(m_2, m_maximal)
```

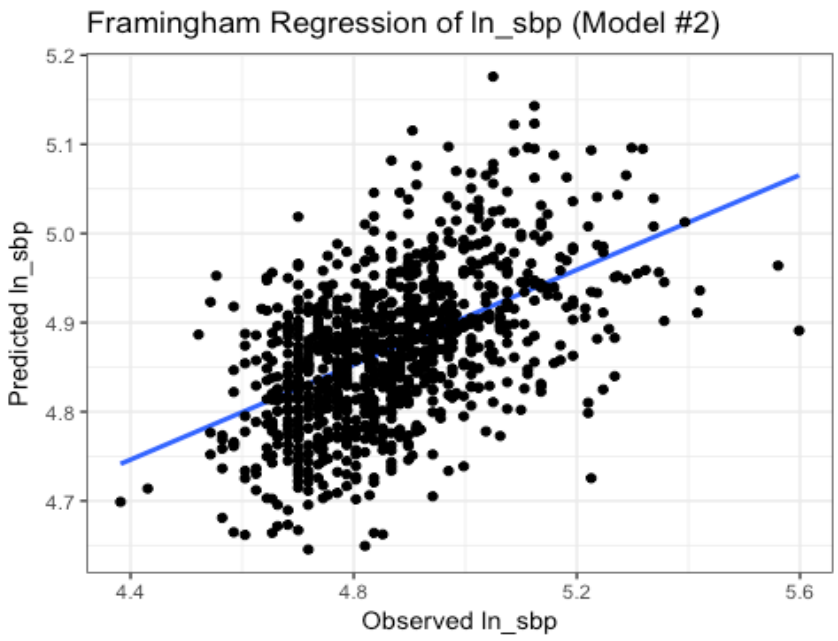
```
[1] "Partial F-test, 2df:  Null:  lnbmiXfemale=0 lnsclXfemale=0"
Analysis of Variance Table

Model 1: ln_sbp ~ ln_bmi + ln_scl + age + female + ageXfemale
Model 2: ln_sbp ~ ln_bmi + ln_scl + age + female + lnbmiXfemale + lnsclXfemale + ageXfemale

  Res.Df    RSS Df Sum of Sq        F     Pr(>F)
1    988 19.314
2    986 19.301  2  0.013173   0.3365 0.7144
```

Interpretation - okay to DROP lnbmi_female and lnscl_female (Partial F = 0.34, p-value = .71) ns
So, model 2 is our "tentative" final model. Model "(2)" is the second column in the table on the previous page

Model (2) is our tentative final model

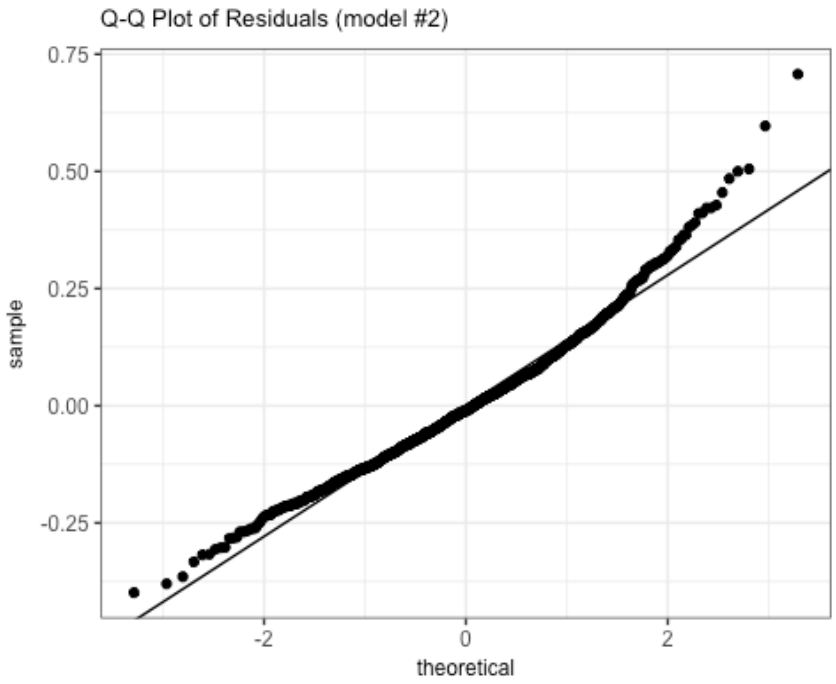
Step 4 – Regression Diagnostics.
Tip – You must fit your model before any diagnostics on it. The diagnostics you run are called "post-estimation" commands (that makes sense, yes?). Fit again the "tentative" final model; this is a necessary preliminary to doing most regression diagnostics. Check model assumptions. Check model adequacy.

```
# DIAGNOSTIC:  Plot of observed v predicted
# Look for:  Points along a straight line ("all is well")
library(ggplot2)
library(Hmisc)
m_best <- lm(data=complete, ln_sbp ~ ln_bmi + ln_scl + age + female + ageXfemale)
complete$yhat <- predict(m_best)
Hmisc::label(complete$yhat) <- "Predicted ln(sbp)"
ggplot(data=complete, aes(x=ln_sbp,y=yhat)) +
   geom_smooth(method=lm, se=FALSE) +
   geom_point() +
   xlab("Observed ln_sbp") +
   ylab("Predicted ln_sbp") +
   ggtitle("Framingham Regression of ln_sbp (Model #2)") +
   theme_bw()
```



Interpretation – Not bad! Ideally the scatter lies on the line defined by 45 degrees. We expect some widening of the confidence intervals at the ends of the range but not too much. What we see here is reasonable.

```
# DIAGNOSTIC:  Normality of Residuals:  QQ plot and Shapiro-Wilk Test of Null: normality ("all is well")
# Look for:  Points along a straight line ("all is well")

library(ggplot2)
complete$residuals <- residuals(m_best)
ggplot2::ggplot(data=complete, aes(sample=residuals)) +
    stat_qq() +
    geom_abline(intercept=mean(complete$residuals), slope = sd(complete$residuals)) +
    ggtitle("Q-Q Plot of Residuals (model #2)") +
    theme_bw() +
    theme(axis.text = element_text(size = 9),
        axis.title = element_text(size = 9),
        plot.title = element_text(size = 10))
```




```
options(scipen=1000)
shapiro.test(complete$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  complete$residuals
## W = 0.9775, p-value = 0.000000000028    # Okay.  Normality rejected.  But plot looks so so. We forge on
```

Interpretation – Here too, we hope to see a scatter on the 45 degree line. Not bad!

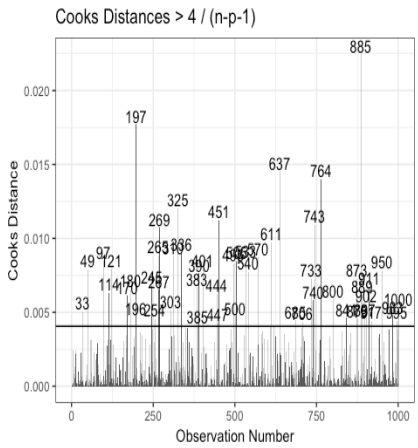
```
# DIAGNOSTIC:  Ramsay test of omitted variables, Null: No omissions ("all is well")
library(lmtest)
lmtest::resettest(m_best,power=2,type="regressor")
##  RESET test
##
## data:  m_best
## RESET = 0.42467, df1 = 5, df2 = 983, p-value = 0.8317
```

Interpretation – Ramsey test is NOT significant (p=.83) suggesting we're okay!

```
# DIAGNOSTIC:  Assessment of multicollinearity (we hope there is not multicollinearity)
# Look for VIF < 10
library(car)
car::vif(m_best)

##     ln_bmi     ln_scl        age     female ageXfemale
##   1.115511   1.175531   2.378150  32.394888  34.116761
```

Interpretation – female and ageXfemale appear to be collinear suggesting some concern about the extent to which there is adequacy of range of age in the 2 genders.

```
# DIAGNOSTIC:  Cook's Distances (looking for influential observations
# Potentially influential observations have Cook distance > 4/(n-p-1).  Other definitions possible
library(Hmisc)
library(ggplot2)

complete$ID <- as.numeric(row.names(complete))        # create a variable containing observation #.  Label it
Hmisc::label(complete$ID) <- "Observation Number"     # Label
complete$cooks <- cooks.distance(m_best)              # Create a variable containing Cook Distance values.

cutoff <- 4/((nrow(complete)-length(m_best$coefficients)-2))    # Calculate "threshold" of high Cook Distance
ggplot2::ggplot(data=complete, aes(x=ID, y=cooks)) +       # Plot X=ID versus Y=Cook
 geom_bar(stat="identity", position="identity") +
 xlab("Observation Number") +
 ylab("Cooks Distance") +
 geom_hline(yintercept=cutoff) +
 geom_text(aes(label=ifelse((cooks>cutoff), ID, "")),
       vjust=-0.2, hjust=0.5) +
 ggtitle("Cooks Distances > 4 / (n-p-1)") +
theme_bw()
```



Interpretation: There are some subject ID whose inclusion in the regression may be influential; eg – ID =197 and 885.

3. Exploratory Data Analysis, Indicator Variables and Interactions

Examine the data to assess:

1. The range and pattern of variability in the outcome variable, Y
2. The range and pattern of variability in the predictor variable X
3. The nature and strength of the presumed linear relationship, Y on X
4. The occurrence of unusual data points requiring further examination; these could be either important data points that are influential or errors.

3.1 Exploratory Data Analysis

3.1.a. Numerical Descriptives on Every Variable and pairwise correlations

{ Package } command	Example
METHOD I: {base} summary(dataframename)	summary(framingham)
METHOD II: {stargazer} stargazer(dataframename,type="text", median=TRUE) What could go wrong: If the input is not a dataframe (for example, it is a tibble), you will only get a header. Solution: dataframename <- as.data.frame(dataframename)	library(stargazer) framingham <- as.data.frame(framingham) stargazer(framingham,type="text", median=TRUE)
METHOD III: {summarytools} summarytools::descr(dataframename)	
METHOD IV: {psych} psych::describe(dataframename)	
METHOD V: {summarytools} print(dfSummary(dataframename))	library(summarytools) print(dfSummary(Isoproterenol))
TABLE OF PAIRWISE CORRELATIONS using cor( ) {stats} myvars <- c("var1", "var2", "var3") cor(dataframename[myvars],method=c("pearson")) To report just 2 digits after decimal temp <- car(dataframename[myvars]) round(temp,digits=2) Other methods are: "spearman", "kendall"	myvars <- c("ln_sbp","age","ln_bmi","ln_scl") temp <- cor(complete[myvars],method=c("pearson")) round(temp,digits=2)

3.1.b. Graphical Descriptives on Every Variable

{ Package } command	Example
METHOD I: {base} pairs(dataframename, main="TITLE") Option – set the size of the points pch=## Option – show upper half only lower.panel=NULL	pairs(framingham,pch=1)
METHOD II: {base} plot(data=dataframename, ~ var+var+var+var, col=rgb(0,0,1,.15), pch=1, main="TITLE")	plot(data=framingham_1000, ~ ln_sbp+age+ln_bmi+ln_scl, col=rgb(0,0,1,.15), pch=1)
METHOD III (my favorite – cb): {psych} myvars <- c("var1", "var2","var3") psych::pairs.panels(dataframename[myvars], method="pearson", hist.col="COLORNAME", density=TRUE, ellipses=TRUE)	library(psych) myvars <- c("ln_sbp","age","ln_bmi","ln_scl") pairs.panels(framingham_1000[myvars], method = "pearson", hist.col = "#00AFBB", density = TRUE, ellipses = TRUE)
METHOD IV {car} car::scatterplotMatrix(data=dataframename, ~ var1 + var2 + var3, diagonal=list(method="histogram",breaks="FD"), main="TITLE")	library(car) car::scatterplotMatrix(data=framingham_1000, ~ ln_sbp + age + ln_bmi + ln_scl, diagonal=list(method="histogram",breaks="FD"), main="Framingham Data (n=1000)")
METHOD V: {GGally} GGally::ggscatmat(data=dataframe, columns=c("var1","var2","var3","var3")) + ggtitle("YOURTITLE") + theme_bw() Tip: Set var1 to be your dependent variable y	library(GGally) GGally::ggscatmat(data=framingham, columns=c("ln_sbp","age","ln_bmi","ln_scl")) + ggtitle("Framingham Data (n=1000)") + theme_bw()

3.1.c. Assess Normality of Dependent Variable

{ Package } command	Example
Shapiro-Wilk Test (Null: Normality) {base} `shapiro.test(dataframenamee$variable)`	`shapiro.test(framingham$sbp)`

Plot I – Histogram w overlay normal
{ggplot2}
`ggplot2::ggplot(data=dataframe, aes(x=var)) + geom_histogram(binwidth=5, colour="blue", aes(y=..density..)) + stat_function(fun=dnorm, color="red", args=list(mean=mean(dataframe$var), sd=sd(dataframe$var)))`
	`library(ggplot2)`
	`p1 <- ggplot2::ggplot(data=framingham, aes(x=sbp)) +`
	`  geom_histogram(binwidth=5, colour="blue",`
	`  aes(y=..density..)) +`
	`  stat_function(fun=dnorm,`
	`  color="red",`
	`  args=list(mean=mean(framingham$sbp),`
	`  sd=sd(framingham$sbp))) +`
	`  ggtitle("Systolic Blood Pressure (sbp)") +`
	`  xlab("Systolic Blood Pressure (mm Hg)") +`
	`  ylab("Density") +`
	`  theme_bw() +`
	`  theme(axis.text = element_text(size = 10),`
	`  axis.title = element_text(size = 10),`
	`  plot.title = element_text(size = 12))`

Plot II – QQ Plot
{ggplot2}
`ggplot2::ggplot(data=dataframe, aes(sample=var)) + stat_qq() + geom_abline(intercept=mean(dataframe$var), slope = sd(dataframe$var))`
	`library(ggplot2)`
	`library(gridExtra)`
	`p2 <- ggplot2::ggplot(data=framingham, aes(sample=sbp)) +`
	`  stat_qq() +`
	`  geom_abline(intercept=mean(framingham$sbp),`
	`  slope = sd(framingham$sbp)) +`
	`  ggtitle("Q-Q Plot of Systolic Blood Pressure (sbp)") +`
	`  theme_bw() +`
	`  theme(axis.text = element_text(size = 10),`
	`  axis.title = element_text(size = 10),`
	`  plot.title = element_text(size = 12))`
	
	Extra- the following puts the 2 plots into one plot
	`gridExtra::grid.arrange(p1, p2, ncol=2)`

3.2 How to Create Indicator Variables

NEVER!!!
Use a NOMINAL Predictor in a lm( ) Command

The estimated slope will be meaningless.

Example:
Party (1=Republican, 2=Democratic, 3 = Libertarian, 4 = Green) is a nominal variable. Because the numbers "1", "2", "3" and "4" are just labels, a unit change in race has no meaning. Therefore, an estimated slope for party also has no meaning.

Review of BIOSTATS 640 Unit 5 (Normal Theory Regression): How to Model Discrete Predictors

(1) A discrete predictor might be **nominal** (eg. – race) or **ordinal** (eg – age, grouped)
(2) Note the **number of levels** (eg – party has 4 levels)
(3) Choose one level to be the **referent** (eg – the group "1=Republican", if this is the most numerous)
(3) K levels require **(K-1) design variables** (eg – For race, we need [4-1]=3 design variables)
(4) Use **ONLY design** variables as predictors.

How to Create a 0/1 Variable in R

Note: The following is a duplicate of the Unit 4 (Introduction to R) notes, page 63
There are at least three (3) ways to do this. Caution! – Take care that you handle missing values correctly.

```
# METHOD I – Most explicit and easiest to understand
newvariable <- NA
newvariable[conditionfor0] <- 0
newvariable[conditionfor1] <- 1
newvariable <- as.numeric(newvariable)

# Method II – Utilizes 0/1 logical operator. Condition true is coded 1. Code false 0.
newvariable <- as.numeric(conditionfor1, na.rm=TRUE)


#  Method III – Utilizes "if/else".  If true is coded 1.   Else is coded 0
newvariable <- with(data=dataframe,ifelse(conditionfor1,1,0), na.rm=TRUE)
```

```
> # EXAMPLE:  Method I
> ivf$female01 <- NA
> ivf$female01[ivf$sex=="male"] <- 0
> ivf$female01[ivf$sex=="female"] <- 1
> ivf$female01 <- as.numeric(ivf$female01)
> table(ivf$sex,ivf$female01)

          0   1
    male 326   0
  female   0 315

> # EXAMPLE:  Method II
> ivf$female01 <- as.numeric(ivf$sex == "female", na.rm=TRUE)
> table(ivf$sex,ivf$female01)

          0   1
    male  326   0
  female   0 315

> # EXAMPLE:  Method III
> # KEY:  ifelse(CONDITION,VALUETRUE,VALUEFALSE)
> ivf$female01 <- with(data=ivf,ifelse(sex=="female",1,0), na.rm=TRUE)
> table(ivf$sex,ivf$female01)

          0   1
    male 326   0
  female   0 315
```

3.3 How to Create Interactions

Review of BIOSTATS 640 Unit 5 (Normal Theory Regression): Interactions and Effect Modification

Note: The following is adapted from BIOSTATS 640 Unit 5

Sometimes the nature of an X-Y relationship is ***different (meaning the slope is different)***, depending on the level of some third variable which, for now, we'll call Z. This is **interaction.** To capture how an X-Y relationship is "different" (or "modified"), depending on the level of Z, we can define an **interaction variable** and then incorporate it as an additional predictor in the model.

Interaction of predictor X with third variable Z = XZ = X*Z

Example: Y = ln_sbp
X = ln_bmi
Z = 0/1 indicator of vertebral fracture (Z=0 not present and Z=1 for present)
XZ = [X] * [Z] = interaction of X and Z

Our full model is thus the following:

$$Y = \beta_0 + \beta_1 Z + \beta_2 X + \beta_3 XZ$$

Key to the betas:

β_0 = intercept for referent (the referent group are patients with $Z = 0$, the non-vertebral fracture folks)
β_1 = CHANGE in INTERCEPT (associated with Z=1, that is - associated with vertebral fracture)
β_2 = slope of change in Y per unit X for referent group
β_3 = CHANGE in SLOPE associated with Z=1 (that is - associated with vertebral fracture)

What does the model become when Z=0?
This yields the model for the non-vertebral fractures patients. Insertion of Z=0 yields

$$Y = \beta_0 + \beta_2 X$$
$$\text{Intercept} = \beta_0$$
$$\text{Slope} = \beta_2$$

What does the model become when Z=1.
This yields the model for the vertebral fractures patients. Insertion of Z=1 yields

$$Y = [\beta_0 + \beta_1] + [\beta_2 + \beta_3]X$$
$$\text{Intercept} = [\beta_0 + \beta_1]$$
$$\text{Slope} = [\beta_2 + \beta_3]$$

How to Create an interaction in R

```
library(Hmisc)

newvariable <- variable1*variable2
```

```
# EXAMPLE -

library(Hmisc)

framingham$ageXfemale <- framingham$age*framingham$female                # create interaction

Hmisc::label(framingham$ageXfemale) <- "AGE x FEMALE interaction"        # label interaction
```

3.4 How to Create Quartiles (or other groupings)

Note: The following is a duplicate of the Unit 4 (Introduction to R) notes, page 65

Perhaps the simplest is to install (one time) the package **gtools** and use the function **quantcut( )**. But you could also do it using the base package. Here, however, you need to be sure to tell R to put the smallest observation into the first bin by including the option **include.lowest=TRUE**.

```
library(gtools)

# METHOD I:  With gtools package and command quantcut( ) and option labels=.
newvariable <- quantcut(sourcevariable, q=#, labels=c("label","label"),na.rm=TRUE)


# METHOD II:  With base package using seq( ) and labeling
newvariable <- with(data=dataframe, cut(sourcevariable,
                                breaks=quantile(sourcevariable, probs=seq(0,1, by=0.2),
na.rm=TRUE),
                                labels=c("Q1","Q2", "Q3", "Q4","Q5"),
                                include.lowest=TRUE))

# METHOD III:  With base package specifying cutoffs explicitly and option labels=
newvariable <- with(data=dataframe, cut(sourcevariable,
                                breaks=quantile(sourcevariable, probs=(c(0, .2, .4, .6, .8,
1)),
                                na.rm=TRUE),
                                include.lowest=TRUE))
```

```
> # Method I
> library(gtools)
> ivf$qage <- quantcut(ivf$matage, q=5, na.rm=TRUE)
> table(ivf$qage)
[23,31] (31,33] (33,35] (35,38] (38,43]
    166     112     123     163      77

> # Method II
> ivf$qage <- with(data=ivf, cut(matage,
                               breaks=quantile(matage, probs=seq(0,1, by=0.2), na.rm=TRUE),
                               labels=c("Q1","Q2", "Q3", "Q4","Q5"),
                               include.lowest=TRUE))

> table(ivf$qage)
 Q1  Q2  Q3  Q4  Q5
166 112 123 163  77

> # Method III
>  ivf$qage <- with(data=ivf, cut(matage,
                                breaks=quantile(matage, probs=(c(0, .2, .4, .6, .8, 1)), na.rm=TRUE),
                                include.lowest=TRUE))

> table(ivf$qage)
[23,31] (31,33] (33,35] (35,38] (38,43]
    166     112     123     163      77
```

4. Simple Linear Regression (Bivariate Analyses)

{ Package } command	Example
# FIT {stats} MODELNAME <- lm(YVAR ~ XVAR, data=DATAFRAME)	model_simple <- lm(calls ~ low, data=ersdata)
# Basic report of fit {stats} summary(MODELNAME)	summary(model_simple)
# Obtain betas (regression coefficients/slopes) {stats} coefficients(MODELNAME)	coefficents(model_simple)
# 95% and 99% CI for the betas {stats} confint(MODELNAME) confint(MODELNAME)level=0.99)	confint(model_simple)
# Obtain variance-covariance matrix of betas {stats} vcov(MODELNAME)	vcov(model_simple)
# Analysis of Variance Table {stats} anova(MODELNAME)	anova(model_simple)
# Obtain R-squared = % Variance Explained {stats} summary(MODELNAME)$r.squared	summary(model_simple)$r.squared
# Obtain predicted values {stats} predict(MODELNAME)	
# Obtain ***residuals*** {stats} residuals(MODELNAME)	
# PLOT – fit with overlay scatter {ggplot2} ggplot2::ggplot(**dataframe**, aes(x=XVAR, y=YVAR)) + geom_smooth(**method=lm, level=.95, se=TRUE**) + geom_point() *Recommended* - Do geom_smooth( ) first - Do geom_points( ) last so that data points are plotted on top of the fit	library(ggplot2) ggplot2::ggplot(ersdata, aes(x=low, y=calls)) + geom_smooth(method=lm, level=.95, se=TRUE) + geom_point() + xlab("Lowest Temperature Previous Day") + ylab("Number of Calls") + ggtitle("Simple Linear Fit w 95% CI of Means") + theme_bw()

4. Simple Linear Regression (Bivariate Analyses) - *continued*

{ Package } command	Example

# PLOT – 	
{ggplot2}
95% CI of mean, 95% CI predict + overlay scatter
{ggplot2}
 	
# Use command predict(modelname, interval="prediction") to extract fitted values 	
yhat <- predict(modelname,
interval="prediction")
 	
# Use command cbind(dataframe,newvariable) to append fitted values to dataframe 	
temp_df <- cbind(dataframename, yhat)
 	
ggplot(temp_df, aes(x=xvar, y=yvar)) +
geom_line(aes(y=lwr), color = "red",
linetype = "dashed") +
geom_line(aes(y=upr), color = "red",
linetype = "dashed") +
geom_smooth(method=lm, level=.95, se=TRUE) +
geom_point()
	library(ggplot2)
	
	yhat <- predict(model_simple, interval="prediction")
	temp_df <- cbind(ersdata, yhat)
	
	ggplot2::ggplot(temp_df, aes(x=low, y=calls)) +
	geom_line(aes(y=lwr), color = "red", linetype = "dashed") +
	geom_line(aes(y=upr), color = "red", linetype = "dashed") +
	geom_smooth(method=lm, level=.95, se=TRUE) +
	geom_point() +
	xlab("Lowest Temperature Previous Day") +
	ylab("Number of Calls") +
	ggtitle("Simple Linear Fit w 95% CI's of Mean and Individual Predictions") +
	theme_bw()

{stats}
# Plot - Residuals v Predicted 	
# Look for: Even band centered at y=0 	
plot(modelname,which=1)
	plot(model_simple,which=1)

{stats}
# Plot - Y=Standardized residuals v X=predicted 	
# Look for: Even band centered at y=0 	
plot(modelname,which=3)
	plot(model_simple,which=3)

{stats}
# Plot - Normal QQ plot 	
# Look for: Straight 45 degree line, slope = 1 	
plot(modelname, which=2)
	plot(model_simple, which=2)

{stats}
# Plot - Cooks Distance 	
# Look for: All small and nothing outstanding 	
plot(modelname, which=4)
	plot(model_simple, which=4)

{stats}
# Plot - Y=Cooks Distance v X=leverage 	
# Look for: no trend of any sort 	
plot(modelname, which=6)
	plot(model_simple, which=6)

5. Multiple Linear Regression and Choose "Tentative" Final Model

5.1 Review of Hierarchical Models

Two models, conveniently referred to as "reduced" and "full", are hierarchical if the all of the predictors in the "reduced" model are contained in the "full" model. Their comparison then addresses the question: are the additional variables in the "full" model significant after adjustment for all the variables in the "reduced" model? The comparison of hierarchical models is an essential tool in regression model development. **Hierarachical model comparison requires that the fitted models are to the SAME observations** –

5.2 Multiple Linear Regression Model Estimation in R

{ Package } command	Example
# Create dataset of observations complete on every variable of interest {base} complete <- na.omit(dataframename, cols=c("var1", "var2", "var3"))	complete <- na.omit(framingham, cols=c("ln_sbp", "ln_bmi", "age", "female", "lnbmiXfemale", "lnsclXfemale","ageXfemale"))
# Partial F-test of reduced v full # NULL: extra predictors are ns (all betas=0) {base} reduced <- lm(data=dataframe, yvar ~ var1 + var2 + … + varp) full <- lm(data=dataframe, yvar ~ var1 + …+ varp + …. Varpk) anova(reduced, full) KEY: (1) observations must be the same in both models (2) predictors in REDUCED model must be a subset of the predictors in the FULL moel	reduced <- lm(data=p53paper, p53 ~ pregnum) full <- lm(data=p53paper, p53 ~ pregnum + early + late) anova(reduced, full)

6. Regression Diagnostics: Model Assumptions and Model Adequacy

6.1 Linearity

Introduction to LOWESS.
"LOWESS" refers to *"locally weighted scatter plot smoother"*. In lowess smoothing, at each value of the predictor X, a regression model is fit to a subset of the data, in particular a collection of data points that are in the immediate neighborhood. You can set the percent of the dataset used at each value of X. The regression model fit is a polynomial. The result is a "smoothing". The smoothing is then compared with a regression model of interest (e.g. – linear) to see how close the two match. Here, we use a comparison of LOWESS smoothing with a linear fit to assess the reasonableness of the assumption of linearity.

{ Package } command	Example
# PLOT – LOWESS Smoothing, Linear and Scatter {ggplot2} ggplot(data=dataframe, aes(x=xvar,y=yvar)) + geom_point() + geom_smooth(method = "loess", aes(color="Loess"), se=FALSE) + geom_smooth(method = "lm", aes(color="Fitted values"), se=FALSE) + scale_colour_manual(name="",values=c("blue", "red")) Tip – Plot scatterplot of observations last so that they are on top.	library(ggplot2) ggplot2::ggplot(data=data.temp, aes(x=pregnum,y=p53)) + labs(x="Number of pregnancies", title="Assessment of Linearity") + geom_point() + geom_smooth(method = "loess",aes(color="Loess"), se=FALSE) + geom_smooth(method = "lm", aes(color="Fitted values"), se=FALSE) + scale_colour_manual(name="",values=c("blue", "red"))

6.2 Normality of Residuals

If all is well, the residuals = [observed – fitted] are reasonably assumed distributed Normal with mean=0 and constant variance.

{ Package } command	Example
# PLOT – QQ Plot of residuals # Look for: Straight line at 45 degrees {ggplot2} # Step 1 - Extract residuals from fitted model df$residuals <- residuals(modelname) # Step 2 – QQ Plot ggplot2::ggplot(data=df, aes(sample=residuals)) + stat_qq() + geom_abline(intercept=mean(df$residuals), slope = sd(df$residuals))	library(ggplot2) complete$residuals <- residuals(m_best) ggplot2::ggplot(data=complete, aes(sample=residuals)) + stat_qq() + geom_abline(intercept=mean(complete$residuals), slope = sd(complete$residuals)) + ggtitle("Q-Q Plot of Residuals (model #2)") + theme_bw() + theme(axis.text = element_text(size = 9), axis.title = element_text(size = 9), plot.title = element_text(size = 10))
# Shapiro-Wilk Test of Normality # Null: Residuals distributed normal shapiro.test(dataframe$var)	shapiro.test(complete$residuals)

6.3 Multicollinearity

Multicollinearity occurs when the predictor variables themselves are linearly interrelated. This is a problem because it makes it difficult to extract the separate effect of each predictor; the betas are unstable. Multicollinearity also has the effect of inflating the variances of the estimated betas. For example, if xvar1 and xvar2 are themselves highly linearly interrelated, then the variance of the beta for xvar1 will be inflated! We use the variance inflation factor (VIF) to assess the data for evidence of multicollinearity.

{ Package } command	Example
# Variance Inflation Factor (VIF) # Look for: All is well if VIF < 10 {car} vif(modelname)	library(car) car::vif(m_best)

6.4 Model Misspecification

A fitted model that fails to include an important explanatory variable is problematic: 1) our understanding of the outcomes is incomplete; 2) estimated associations may be biased due to confounding; and/or 3) model assumptions may be violated. The **Ramsay Test** tests the null hypothesis that predicted values from the fitted model are unrelated to powers of the fitted model, after adjustment for the predictor variables in the model. The test statistic is an F statistic. All is well (meaning we have not failed to include an important explanatory variable) if the null hypothesis is NOT rejected. Evidence of a failure to include one or more explanatory variables is reflected in a large F statistic value. Tip - I recommend that you also do a scatterplot of the squared standardized residuals versus the leverage values. Omission of an important explanatory variables is suggested by: 1) extreme values; and/or 2) any systematic pattern.

{ Package } command	Example
# Ramsay Test for Omitted Variables # Look for: All is well if null is NOT rejected {lmtest} resettest(modelname,power=2,type="regressor")	library(lmtest) lmtest::resettest(m_best,power=2,type="regressor")

6.5 Constant Variance

Recall what we are assuming with respect to homogeneity of variance of the outcome at each profile of values on the predictor variables. If homogeneity of variance is violated, this is hetroskedasticity. This is a problem because: 1) SE estimates may be in error and, in turn 2) hypothesis tests may be in error as well.

{ Package } command	Example
# Breusch-Pagan Test of Constant Variance # Look for: All is well if null is NOT rejected {car} ncvTest(modelname)	library(car) car::ncvTest(m_best)
# Plot X=predicted v Y=residual # Look for: Even band, centered at Y=0 {stats} plot(modelname,which = 1)	plot(m_best,which = 1, main = "Model Check - constant variance")

# Plot X=predicted v Y=residual 	
# Look for: Even band, centered at Y=0 	
{ggplot2}
 	
# Step 1 – Get predicted and residuals 	
xpredicted <- predict(modelname)
yresidual <- residuals(modelname)
 	
# Step 2 – Plot 	
ggplot(data=dataframe,
aes(x=xpredicted,y=yresidual)) +
geom_point()	

6.6 Outlying, High Leverage, and Influential Points

Recall: 1) **outliers** are observations with large residuals (observed – predicted) and 2) **high leverage values** are observations with extreme values on the predictor variables.

Cook's distance provides a measure of the influence of an individual data point on the fitted model and is a function of the values of both the residual and leverage. **Cook's Distance** = Change in estimated regression coefficient value, expressed in standard error units.

{ Package } command	Example

`# Plot X=Study ID v Y=Cook Distance`
`# Look for: Small and no spikes`
`{stats}`
 	
`plot(modelname,which = 4)`
	`plot(m_best,which = 4)`

`# Plot X=Study ID v Y=Cook Distance`
`# Look for: Small and no spikes`
`{ggplot2}`
 	
`# (if you need to) Create variable ID.`
`dataframe$ID <- as.numeric(row.names(dataframe))`
 	
`# Create variable w cooks distances (cooks)`
`dataframe$cooks <- cooks.distance(modelname)`
 	
`# Solve for threshold cook using cutoff = 4 / (n-p-1).`
`cutoff <- 4/((nrow(dataframe)-length(modelname$coefficients)-2))`
 	
`#Plot`
`ggplot(data=dataframe, aes(x=ID, y=cooks)) +`
`geom_bar(stat="identity", position="identity") +`
`geom_hline(yintercept=cutoff) +`
`geom_text(aes(label=ifelse((cooks > cutoff), ID, "")), vjust=-0.2, hjust=0.5)`
	`library(Hmisc)`
	`library(ggplot2)`
	
	`# Create variable ID. Label it.`
	`complete$ID <- as.numeric(row.names(complete))`
	`Hmisc::label(complete$ID) <- "Observation Number"`
	
	`# Create variable w Cooks distances (cooks)`
	`complete$cooks <- cooks.distance(m_best)`
	
	`# Solve for threshold cook using cutoff = 4 / (n-p-1).`
	`cutoff <- 4/((nrow(complete)-length(m_best$coefficients)-2))`
	
	`#Plot`
	`ggplot2::ggplot(data=complete, aes(x=ID, y=cooks)) +`
	`geom_bar(stat="identity", position="identity") +`
	`xlab("Observation Number") +`
	`ylab("Cooks Distance") +`
	`geom_hline(yintercept=cutoff) +`
	`geom_text(aes(label=ifelse((cooks>cutoff), ID, "")), vjust=-0.2, hjust=0.5) +`
	`ggtitle("Cooks Distances > 4 / (n-p-1)") +`
	`theme_bw()`

7. Reporting

I Highly Recommend! – Consider Use the command stargazer( ) to produce a summary display of your models, side-by-side.
KEY – The models being compared side-by-side MUST BE fit to the same observations.

```
library(stargazer)
# Basic:  Beta, SE(beta)
stargazer(model1,model2,type="text",font.size="small", align=TRUE, omit.stat=c("f","ser"))

# ---- Lots of options ------#
ci=TRUE                       # Report 95% CI of beta instead of SE(beta)
ci.level=.99                  # Change confidence level to desired value (here = .99)
report=("vc*p")               # Report p-value instead of SE(beta)
report=("vct*")               # Report t-statistic(and significance stars) instead of SE(beta)
omit.stat=c("f")              # Omit reporting of overall F statistic
omit.stat=c("f","ser","rsq") # Omit reporting of overall F, residual SE, R-squared
intercept.bottom=FALSE        # Move the intercept row to the TOP of the table
digits=1                      # Control the number of digits displayed (here = 1)
title="YOURTITLE"             # Provide a title

covariate.labels=             # Display variable labels instead of variable names
  c("label", "label")

dep.var.labels=               # Display dependent variable labels instead of variable names.
  c("label", "label")

column.labels =               # Label the models (these are your column headings)
 c("model1", "model2")
no.space=TRUE                 # Remove empty lines
font.size="small"             # Control font size (here I chose "small")
out="NAME.txt"                #  Save table to a ".txt" file
out="NAME.htm"                #  Save table to a ".htm" file (Word can read this just fine)
```

```
# Example
stargazer(m_4,m_5,type="text",font.size="small", align=TRUE, omit.stat=c("f", "ser"))
=========================================
               Dependent variable:
             ---------------------------
                         p53
                  (1)            (2)
-----------------------------------------
pregnum         0.492***      0.475***
                (0.125)        (0.114)

agecurr          -0.005
                (0.014)

menop            -0.280        -0.367
                (0.378)        (0.281)

Constant        2.704***      2.569***
                (0.440)        (0.208)

-----------------------------------------
Observations       67             67
R2               0.218          0.216
Adjusted R2      0.181          0.192
=========================================
Note:         *p<0.1; **p<0.05; ***p<0.01
```

Design ..... Data Collection ..... Data Management ..... Data Summarization ..... Statistical Analysis ..... Reporting