# Assignment 1

## Recommended R Reading

[W3 Schools R Tutorial](https://www.w3schools.com/r/default.asp): Free, no sign up required

* Sections “R Syntax”, “R Variables”, “R Data Types”, “R Vectors”, “R Data Frames”
  + This covers basic (but fundamental!) concepts in R. Each lesson is quite short.

Many beginner-level resources can be found by searching online for “R tutorial” if you’d like more practice.

[R for Data Science, Chapter 5 Data Transformation](https://r4ds.had.co.nz/transform.html)

* Sections “Introduction”, “Add new variables with mutate()”
  + This is meant as a supplement to the R lab code. Material in other sections will be covered in future lectures.

## HCUP Resources

See last slide of the R lab lecture notes.

## Task 1: Build familiarity with HCUP’s data elements documentation

Check the documentation for the following variables used in this assignment:

* AGE
* AHOUR
* AWEEKEND
* LOS
* LOS\_X
* PAY1
* PRDAY*n*

## Task 2: Practice setting up an R script

Following the example code used in class, create an R script for this assignment in your folder on the RIS server. Load the tidyverse package and set your working directory to the same folder. (You may need to load other packages to complete subsequent tasks; if so, write those statements in the same part of the script.) Read the data set containing the 1% sample of the Florida SID used in class.

## Task 3: Practice using View() in RStudio to directly inspect data

The slides for lecture 2 described the PRDAY*n* fields as representing the number of days relative to admission that a procedure occurred. Are procedures always listed in chronological order? That is, is it the case that PRDAY1 <= PRDAY2 <= PRDAY3 <= …? Using RStudio’s View feature, inspect the first few rows of data. Note that because View only shows 50 columns at a time, you will need to scroll over several times; the PRDAY*n* fields start at around column 140.

Write a comment in your script with your observation. (Yes or no is sufficient.)

## Task 4: Practice checking for missing values and defining new variables

Use the function summary to produce a numeric summary of AGE. Notice that one record has missing AGE in the 1% sample data set. In this task, we’ll define a variable called adult that takes the value 0 if AGE is between 0 and 17, 1 if AGE is 18 or greater, and NA if AGE is missing.

Complete the definition of adult in the code below by replacing “< >” with the correct values or expressions.

core1p <- mutate(

core1p,

adult = ifelse(is.na(AGE), < >,

ifelse(< >, 1, < >))

)

A frequency table of the new variable should return the following:

table(core1p$adult, useNA = "ifany")

0 1 <NA>

2313 100419 1

## Task 5: Practice defining variables based on the values of two or more existing variables

A research topic that has received some attention in the literature is whether patients admitted to hospitals outside of regular business hours experience poorer outcomes. In a meta-analysis, Zhou *et al.* (2016; doi:10.1161/JAHA.115.003102) found increased risk of mortality in hospital or shortly after discharge for these so-called ‘off-hours’ admissions across a wide array of conditions. One particular study by Magid and colleagues (2006; doi: 10.1001/jama.294.7.803) looked at the relationship between time of admission and in-hospital mortality in patients with ST-segment elevation myocardial infarction (STEMI). They defined off-hour admissions as those occurring on weekends or between 5 pm and 7 am on week days (that is, midnight to 7 am and 5 pm to midnight).

Using AHOUR and AWEEKEND, define a variable off\_hours based on this definition.

## Task 6: Practice categorizing a numeric variable.

Writing in the *American Journal of Cardiology*, Doshi *et al.* (2019; doi:10.1016/j.amjcard.2018.11.045) describe an analysis comparing outcomes in patients with atrial fibrillation covered by Medicaid and private insurance using data from the HCUP National Inpatient Database (NIS). They report higher rates of in-hospital mortality, lower rates of discharge to home, and slightly longer lengths of stay in Medicaid-insured patients (Table 3).

Following the example code used in class, redefine LOS as LOS if it was nonmissing and LOS\_X if it was missing. Then use cut to produce a new variable LOScat that categorizes length of stay: 0–2 days, 3–4 days, 5–8 days, and 9–488 days.

Produce a crosstabulation of PAY1 and LOScat. Do Medicaid patients tend to have longer hospital stays than privately insured patients? (No formal analysis is needed, just your observation.) Note that because the number of Medicaid and privately insured patients differs, you’ll want to compare the *proportion* of patients in the different categories, not just the counts.

## Task 7: Practice discovering new functions and reading R package documentation

Because RStudio’s View feature only shows 50 columns at a time, it can get tedious having to scroll across the screen to find a variable in the middle of the data set. When creating new variables and checking your work, one helpful technique is to rearrange the columns of the data set so that a few particularly important variables occupy the first few columns when View’ed.

Read the documentation for the function relocate: <https://dplyr.tidyverse.org/reference/relocate.html>. (Notice that it’s defined by the same package that provides select and mutate.)

Use relocate to move the columns VisitLink and DaysToEvent to the leftmost position in the 1% sample data set.