Working with data frames

One of the most important skills is picking out of a data frame the cases that you want. If \( d \) is a data frame and \( \text{cond} \) is a logical condition involving the variables of \( d \), then

\[
\text{subset}(d, \text{cond})
\]

is a data frame that consists of only those cases of \( d \) where the condition \( \text{cond} \) is true.

Load the \texttt{reshape} package. (Check the box by \texttt{reshape} in the package menu.) The dataframe \texttt{tips} has data on the tips that a waiter collected from many different parties.

Compute the following:

- The total number of parties that the waiter served
- The total of all the tips
- The largest tip
- The total of all tips made on Thursdays
- The mean tip
- The mean tip left by a female
- The number of parties served on Sundays

Adding new variables to data frames

The percentage of a tip is \( \text{tip/total\_bill} \). To add that to the data frame we could do one of the following

\[
\texttt{> tips$pct = tips$tip/tips$total\_bill} \\
\texttt{> tips = transform(tips, pct = tip/total\_bill)}
\]

Add a column that computes the average tip per person in the party. Compute

- The largest average tip per person
- The largest average tip per person on a Saturday

Using the dataframe \texttt{counties} add a new variable to the data frame that is equal to the natural log of the population of the county.

- What is the mean population of a US county?
- What is the mean of the log of US county populations?
- What is the log of the mean of US county populations?
Using functions

One possible source of confusion in learning to use R functions is that there are a variety of syntax paradigms depending on the kinds of objects the function wants to see and who wrote the function. Here are three types of function syntax that we have used, illustrated by some functions.

1. A function that wants a vector (or vectors) as arguments. An example is `IQR`. Most functions that operate on data can use this syntax.

   ```r
   > IQR(tips$tip)
   > IQR(counties$Population)
   ```

2. A function that wants a vector as argument and that supports the `data=` argument. `IQR` (and many base R functions) does not support the `data=` argument which is why we needed the data frame name in the vector name in the example above. But many of our favorite functions do (although only if we have loaded the appropriate packages).

   ```r
   > mean(tips$tip)
   > mean(tip, data = tips)
   > mean(tip, tips) # because the data argument is the second argument expected
   ```

3. A function that wants a formula as an argument. Such functions invariably also support the `data=` argument and should always be used this way.

   A formula looks like `y ~ z`. Here are some important things to keep in mind.

   (a) `y` is usually the response variable in a model or the variable to be plotted on the y axis in a graph. It is sometimes missing because it is not explicitly named but rather computed.

   (b) In general, the named objects in the formula are variables (either quantitative (vector) variables or categorical (factor) variables) that usually live in a particular data frame.

   (c) The formulas can have arithmetic symbols in them (like + signs) but the meaning of these symbols varies according to the kind of function that is being applied.

   (d) Some functions take either formulas or vectors (like `mean`) so it is important to understand what is being computed in each case.

   (e) If the first argument is a formula, almost always the second argument is the dataframe.

   ```r
   > mean(tip ~ sex, tips)
   > mean(~tip, tips)
   > mean(tip, tips)
   ```

Note that functions that have other inputs besides the variables to operate on do not have an especially consistent syntax for where those arguments go. For example, `qdata` requires a number and a vector in that order while `sample` requires a vector and a number!

   ```r
   > qdata(0.5, tip, tips)
   > sample(tips$tip, 5) # sample doesn't support formula form
   ```
A dictionary of our favorite functions

See if you can recall for each function what kind of argument that it wants and what kind of syntax that it supports.

1. Basic summary statistics


   > mean()
   > median()
   > var()
   > sd()
   > max()
   > min()
   > pdata()
   > qdata()

2. Models


   > mm()
   > fitted()
   > residuals()
   > tally()

3. Graphics


   > histogram()
   > dotplot()
   > densityplot()
   > bwplot()
   > mosaicplot()

4. Sampling


   > sample()
   > resample()

5. Working with data frames


   > str()
   > head()
   > subset()
   > transform()