R Cheat Sheet

1. **Computing a correlation**. `cor(x, y)`

2. **Computing a correlation with missing data**. `cor(x, y, use = "complete")`

3. **Subsetting a data frame**. You can use the `subset` function to subset a data frame according to a variable in that data frame. For example, if you have a data frame `x` and you only want the rows of the data frame that correspond to temperature (`tmpd`) < 50, then you can do

   ```
   new.x <- subset(x, tmpd < 50)
   ```

   Multiple logical statements can be used, so you can select only the rows that have temperature ≥ 50 but < 80 by doing

   ```
   new.x <- subset(x, tmpd >= 50 & tmpd < 80)
   ```

4. **Subsetting by date**. Subsetting a data frame by date can be done easily by using R's date/time functionality. For example, suppose you only wanted the part of a data frame that corresponded to everything after January 15, 1995. Then you could do

   ```
   subset(x, date >= as.Date("1995-01-15"))
   ```

   If you wanted everything after January 15, 1995 and before November 15, 1998, you could do

   ```
   subset(x, date >= as.Date("1995-01-15") & date < as.Date("1998-11-15"))
   ```

5. **Calculating quarters/seasons**. R's date/time functionality allows you to automatically calculate what quarter a given date is in. The function `quarters` returns a vector of "Q1", "Q2", "Q3", or "Q4" depending on whether a date is in the first, second, third, or fourth quarter. If you want to subset a data frame by only keeping the rows that correspond to the second quarter (April–June), you can do

   ```
   subset(x, quarters(date) == "Q2")
   ```

6. **Fitting a linear regression model**. Linear models can be fit with the `lm` function. Models are specified with the $\sim$ symbol. The variable on the left of the $\sim$ is the response and the variable on the right of $\sim$ is the predictor or dependent variable. So to fit a model such as

   $$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

   where ε_i is some "error", you can do

   ```
   fit <- lm(y ~ x)
   ```

   to get estimates of β_0 and β_1.

7. **Dividing a variable into ranges/categories**. The `cut` function can be used to divide a continuous variable into categories or ranges. For example, if you wanted to create a categorical variable for `x` where the ranges were `x` < 50, $50 \leq$ `x` < 80, and `x` ≥ 80, then you could do

   ```
   new.x <- cut(x, c(-Inf, 50, 80, Inf))
   ```

   You can even add labels to each of the categories, such as

   ```
   new.x <- cut(x, c(-Inf, 50, 80, Inf), labels = c("Cold", "Warm", "Hot"))
   ```