Scatter plots and linear regression

This worksheet is an interactive, guided module for learning the basics of scatter plotting, and of fitting linear models to data. It also shows how to read datafiles and produce a dataframe containing your data.

Let us begin by learning how to read datasets and create a dataframe.

Example: The comma-separated data file named "winter.csv" contains the following 4 variables: (1) name of a U.S. city; (2) the mean January temperatures in that city (in degrees F); (3) the latitude of the city; and (4) the January temperature in degrees c.
We will read the data, print it out and see what it looks like, and plot the mean January temperature vs. latitude. </FONT>

The commands below show how to do this.

In [6]:
```r
# Read the csv file
winterdat = read.csv(file="./winter.csv", header=TRUE, sep=",")
# Some points to note:
#    header=TRUE says my file has a header line at the top
#    TRUE must be in all upper-case
#    sep="," says to use comma as the separator

head(winterdat)   # just to print out and see the data
```

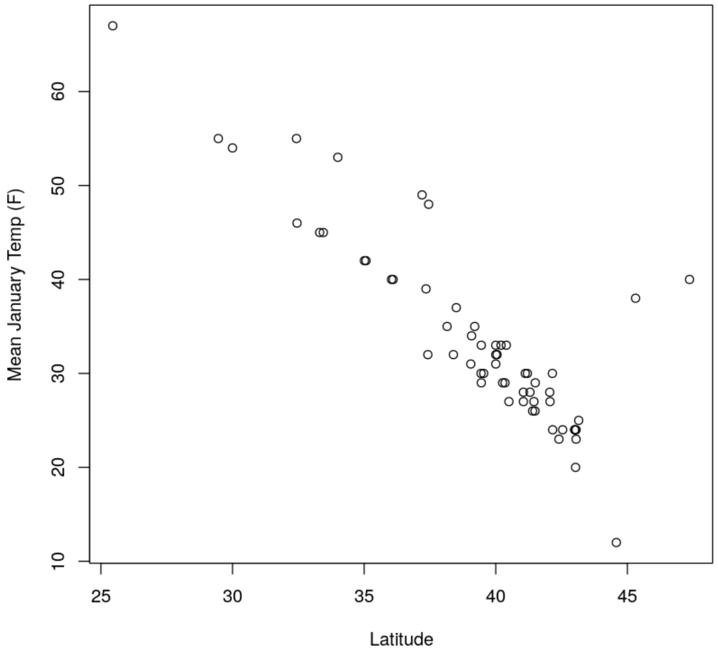
A data.frame: 6 × 4

	City	Mean_Jan_Temp_F	Latitude	Jan_degreesC
	<fct>	<int>	<dbl>	<dbl>
1	Akron, OH	27	41.05	-2.78
2	Albany-Schenectady-Troy, NY	23	42.40	-5.00
3	Allentown, Bethlehem, PA-NJ	29	40.35	-1.67
4	Atlanta, GA	45	33.45	7.22
5	Baltimore, MD	35	39.20	1.67
6	Birmingham, AL	45	33.31	7.22

The above "read.csv" command produces a dataframe named "winterdat". We can now compute summary stats, plot histograms, boxplots, etc. for the above variables. However, given the focus of the present tutorial, let's scatter plot the mean January temperature vs. latitude

```
In [2]: # Let's first create shorter names for the explanatory & response variables:
        xvar = winterdat$Latitude    # be careful and spell it exacly as shown in abo
        ve printout
        yvar = winterdat$Mean_Jan_Temp_F

        # Next, make a scatter plot:
        plot(xvar, yvar, xlab="Latitude", ylab="Mean January Temp (F)")

        # We can add the line of best-fit to the scatter plot, without
        # actually finding its equation (uncommet the next line to see it):
        # abline(lm(yvar ~ xvar))      # draw regression line on scatter plot

        # It is easy to find the correlation
        r = cor(xvar, yvar)
        cat("correlation=", r)  # This is one one way to print text and variables tog
        ether
```

```
correlation= -0.8573135
```

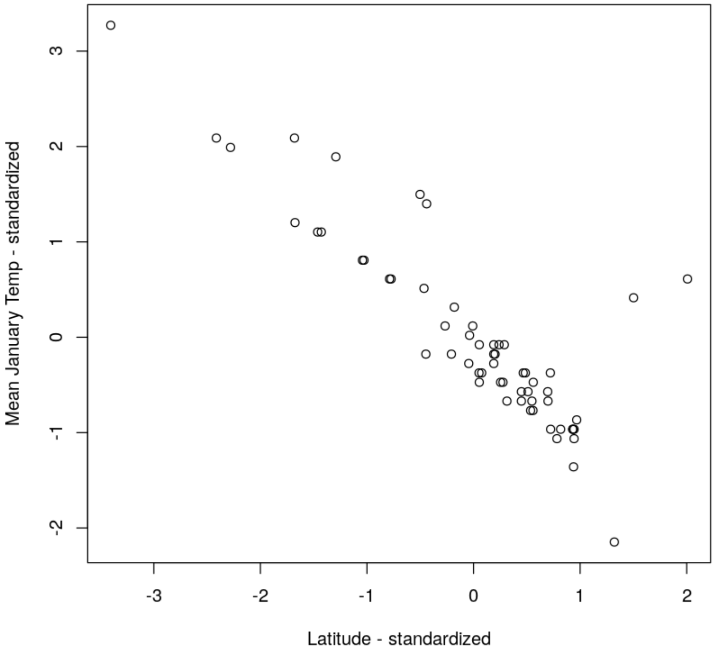


Nest, we will standardize the x, y variables and convert them to z-scores.
Here is how:

In [3]:

```
# Use the "scale" function to standardize as shown below.
# Note that we're using the names "zx", "zy" to store the
# z-scores after standardizing:
zx = scale ( xvar, center = TRUE, scale = TRUE )  # the "center" option subtracts the mean
zy = scale ( yvar, center = TRUE, scale = TRUE )  # the "scale"  option divides by the SD

# Now, let's scatterplot the standardized values and see what
# they look like
plot(zx, zy, xlab="Latitude - standardized", ylab="Mean January Temp - standardized")  # make scatter plot

# Let's verify that the correlation has remained the same
# even after standardizing:
r_after = cor(zx, zy)
cat("correlation after standardizing=", r_after)
```

```
correlation after standardizing= -0.8573135
```

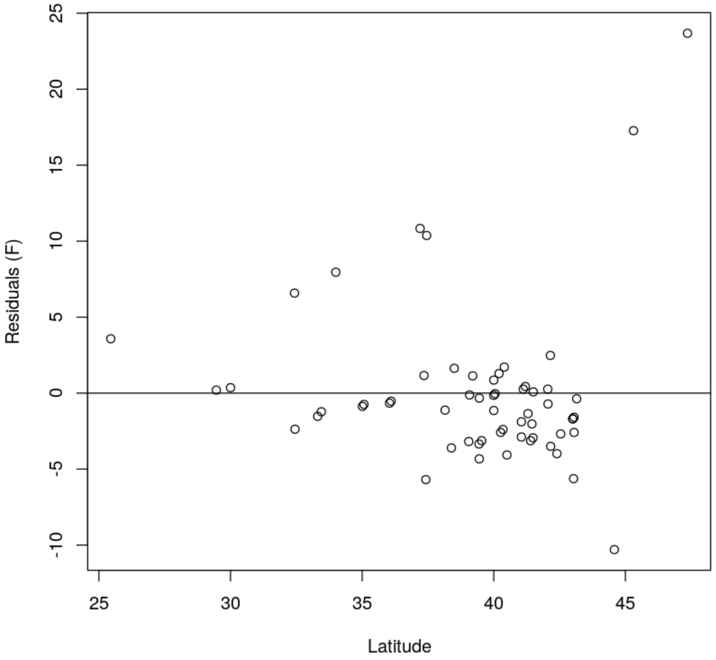


Next, we will let R compute a linear regression model for us.
Note that there are several, slightly different, variations on how to do this. Some examples are shown below.

In [4]:

```
# Method 1: Use the variables "xvar", "yvar" which have been
# extracted from original dataframe
lm(yvar ~ xvar)

# Method 2: Directly use the variables from original dataframe
lm(Mean_Jan_Temp_F ~ Latitude, data = winterdat)

# Method 3: Store the results of "lm" in a new variable
# and query that variable for information about the model
lmresults = lm(Mean_Jan_Temp_F ~ Latitude, data = winterdat)
summary ( lmresults )

# Notice that Method 3 gives more information about
# the results, including the R-squared value.
myres = residuals (lmresults )  # prints out values of the residuals
plot(xvar, myres, xlab="Latitude", ylab="Residuals (F)")
abline( 0, 0 )   # add horizontal reference line on x-axis
```

```
Call:
lm(formula = yvar ~ xvar)

Coefficients:
(Intercept)         xvar
     118.14        -2.15

Call:
lm(formula = Mean_Jan_Temp_F ~ Latitude, data = winterdat)

Coefficients:
(Intercept)     Latitude
     118.14        -2.15

Call:
lm(formula = Mean_Jan_Temp_F ~ Latitude, data = winterdat)

Residuals:
     Min       1Q   Median       3Q      Max
-10.2978  -2.6353  -0.8719   0.3965  23.6789

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  118.139      6.743   17.52   <2e-16 ***
Latitude      -2.150      0.171  -12.57   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.272 on 57 degrees of freedom
Multiple R-squared:  0.735,     Adjusted R-squared:  0.7303
F-statistic: 158.1 on 1 and 57 DF,  p-value: < 2.2e-16
```



Exercise:
The U.S. Center for Disease Control and Prevention (CDC) publishes state by state data on mortality rates by different causes, including deaths by firearms. Using this in conjunction with gun ownership data in each state, we can explore the association, if any, between firearm deaths and gun ownership. The file "firearms2013.csv" contains these data for the year 2013. Carry out the following tasks:

1. Read the file into R
2. Make a sctterplot of "deaths_per_100k" vs "gun_ownership_rate". Be sure to label your axes.
3. Compute the correlation between those two variables
4. Plot the same two variables in standardized form
5. Construct a linear regression model to predict firearm deaths from gun ownership rate. Plot the model together with the original data.
6. Plot the residuals.
7. Find the R^2 value.
8. Write a short paragraph discussing the quality and appropriateness of the linear model, based on the scatter plot, correlation, R^2, etc. Are there any conclusions you can draw from the model?

Turn in a printed copy of a PDF

```
In [ ]:
```