Intro to R Live Coding

Brian Anderson

Libraries

How to install a package (library)

```r
install.packages("tidyverse")
```

Once installed, they are always there, but to access the library in a session, you need to load it into memory.

```r
# Load library into memory
library(tidyverse)
```

Simulated Data

There are lots of ways to load data into R—basically, any type of data from any source can find its way into R. For our purposes, we're going to create our own.

Lets create data from a multiple regression model:

$$y = \alpha + \beta_x + \beta_w + \epsilon$$

Lets set some values, and we'll simulate a 200 observation dataset.

$$y = 3.5 + .65_x + .3_w + \mathcal{N}(0, 3)$$

```r
set.seed(1)
obs <- 200

df <- tibble(x = rnorm(obs, 0, 2),
             w = rnorm(obs, 0, 3))

df
```

```
## # A tibble: 200 x 2
##         x       w
##     <dbl>   <dbl>
##  1 -1.25   1.23
##  2  0.367  5.07
##  3 -1.67   4.76
##  4  3.19  -0.993
##  5  0.659 -6.86
##  6 -1.64   7.49
##  7  0.975  2.00
##  8  1.48   1.62
##  9  1.15  -0.0402
## 10 -0.611  1.53
## # ... with 190 more rows
```

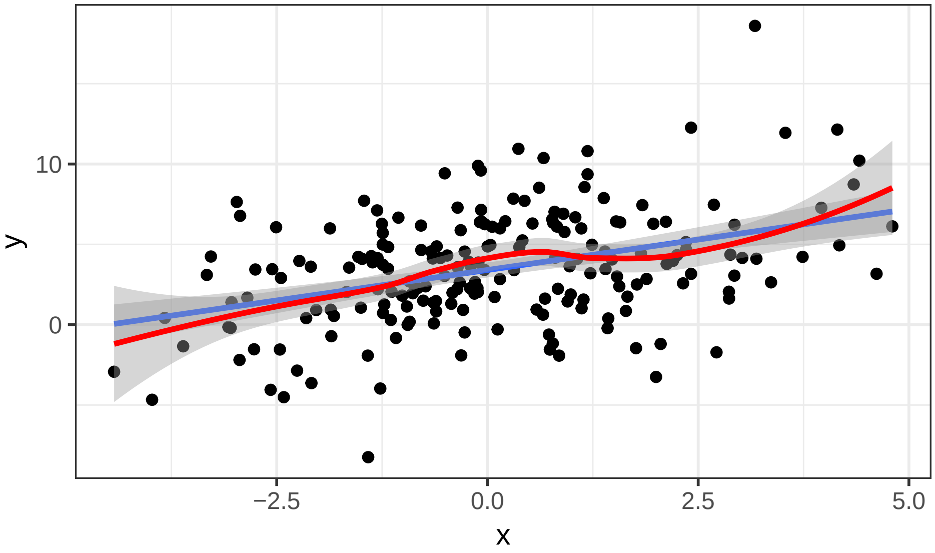
Next lets create our outcome variable:

```r
# Define our model parameters
a <- 3.5
b_x <- .65
b_w <- .3

df <- df %>%
  mutate(y = a + (b_x * x) + (b_w * w) + rnorm(obs, 0, 3))

df
```

```
## # A tibble: 200 x 3
##         x       w      y
##     <dbl>   <dbl>  <dbl>
##  1 -1.25   1.23    6.28
##  2  0.367  5.07   10.9
##  3 -1.67   4.76    2.03
##  4  3.19  -0.993   4.10
##  5  0.659 -6.86    0.623
##  6 -1.64   7.49    3.55
##  7  0.975  2.00    3.63
##  8  1.48   1.62    4.06
##  9  1.15  -0.0402  8.56
## 10 -0.611  1.53    1.47
## # ... with 190 more rows
```

Basic Visualizations

```r
ggplot(data = df, aes(y = y, x = x)) +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_smooth(method = "loess", color = "red") +
  theme_bw()
```



Linear Model

```r
y.model <- lm(y ~ x + w, data = df)
summary(y.model)
```

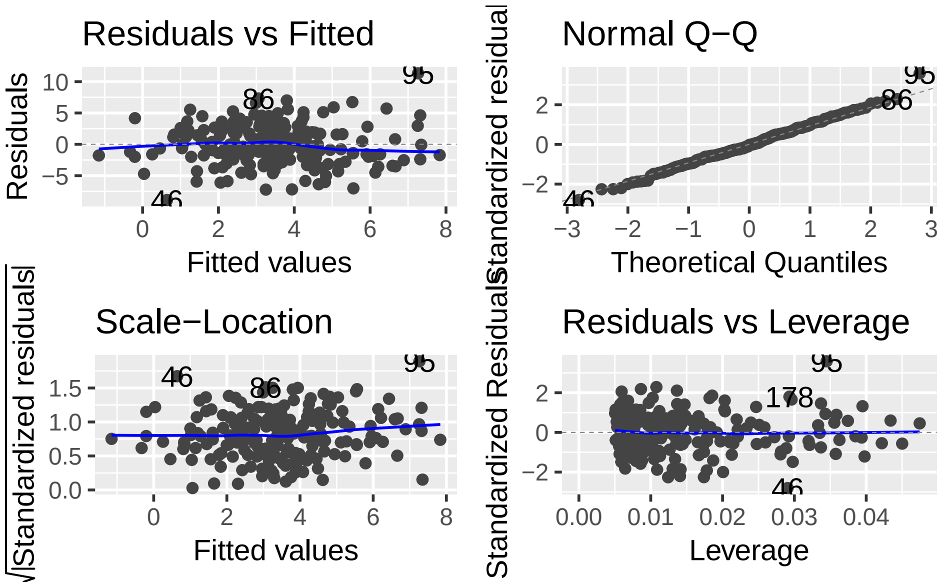
```
##
## Call:
## lm(formula = y ~ x + w, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8859 -1.9847 -0.1023  2.0382 11.3307
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.36942    0.22790  14.785  < 2e-16 ***
## x            0.76685    0.12279   6.245 2.55e-09 ***
## w            0.27373    0.07525   3.638 0.000351 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.218 on 197 degrees of freedom
## Multiple R-squared:  0.2066, Adjusted R-squared:  0.1985
## F-statistic: 25.65 on 2 and 197 DF,  p-value: 1.259e-10
```

Regression Diagnostics

Install a new package

```r
install.packages("ggfortify")
```

```r
library(ggfortify)
autoplot(y.model)
```



Formatting Results

Tidying Model Results

```
install.packages("broom")
```

Creating a manipulable dataframe

```
library(broom)
y.tidy <- tidy(y.model)
y.tidy
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    3.37     0.228      14.8  9.08e-34
## 2 x              0.767    0.123       6.25 2.55e- 9
## 3 w              0.274    0.0752      3.64 3.51e- 4
```

Formatting Results Tables

```
install.packages("sjPlot")
```

Create an html table for copy and pasting…

```
library(sjPlot)
tab_model(y.model)
```

Summary Statistics

A note on correlation matrices.

```
install.packages("skimr")
```

Descriptive statistics

```
library(skimr)
skim(df)
```

```
## Skim summary statistics
##  n obs: 200
##  n variables: 3
##
## -- Variable type:numeric ---------------------------------------------------------------------------
##  variable missing complete   n  mean    sd     p0    p25     p50  p75   p100
##        w       0      200  200 0.12   3.03 -8.67 -1.49  0.0098 2.2    7.95
##        x       0      200  200 0.071  1.86 -4.43 -1.23 -0.099  1.23   4.8
##        y       0      200  200 3.46   3.59 -8.24  1.36  3.47   5.99 18.59
##      hist
##
##
##
```