|  |  |  |
| --- | --- | --- |
| Word Equations (not R)**outcome = predictor + other stuff****Y = X + other stuff** | Other Basics print("Hello world!")  # arithmetic operations  sum(1, 2, 100)  +, -, \*, /  # logical operations  >, <, >=, <=, ==, !=, |, & | Extract and Save a Vector Y <- data\_set$variable |
| Basic Scatterplot gf\_point(Y ~ X, data = data\_set) | Predictions and Residuals prediction <- our\_function(X)  residual <- Y - our\_function(X) |
| Data Frame # view first/last six rows  head(data\_set)  tail(data\_set) | Custom Function our\_function <- function(X){-5.5 + 49\*X}  # evaluate a function  our\_function(.24) | # sum of residuals  sum(residual)  # sum of squared residuals  sum(residual^2) |
| Manipulate Data Frame # select multiple variables  select(data\_set, Y1, Y2)    # save new data frame  new\_data <- select(data\_set, Y1, Y2)  # find rows that meet condition  filter(data\_set, Y > 300) | # arrange rows by variable  arrange(data\_set, Y)    # create a new variable in data frame  mutate(data\_set, Y3 = Y1 - Y2) | Statistics with Vectorsmean(Y) sse(Y, our\_function(X))  mse(Y, our\_function(X)) rmse(Y, our\_function(X)) |
| Best-Fitting Linear Model # use one explanatory variable lm(Y ~ X, data = data\_set) |
| Visualizations # basic scatterplot  gf\_point(Y ~ X, data = data\_set,  color = "purple")    # add points to a scatterplot  gf\_point(Y ~ X, data = data\_set) %>%  gf\_point(6 ~ 0.22, color = "red")    # scatterplot with categorical X  gf\_point(Y ~ X, data = data\_set) | Multivariate Visualizations # add predictor variables  gf\_point(Y ~ X1, data = data\_set,  size = 4, color = ~X2, shape = ~X3)    # separate facets of scatterplots  gf\_point(Y ~ X1, data = data\_set) %>%  gf\_facet\_wrap(~X3)    # add a line to a scatterplot  gf\_point(Y ~ X, data = data\_set) %>%  gf\_abline(intercept = -5.5, slope = 49) | # build custom function  our\_function <- function(X){-5.5 + 49\*X}  # add function and its predictions to graph  gf\_point(Y ~ X, data = data\_set) %>%  gf\_function(our\_function) %>%  gf\_point(our\_function(0.18) ~ 0.18) %>%  gf\_point(our\_function(0.20) ~ 0.20) %>%  gf\_point(our\_function(0.22) ~ 0.22)    # add function predictions for all data  gf\_point(Y ~ X, data = data\_set) %>%  gf\_point(our\_function(X) ~ X) |