|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Basics print("Hello world!")  # assign value to object  myNumber <- 5  # combine values into vector  myVector <- c(1, 2, 3)  # first element in vector  myVector[1]  # arithmetic operations  sum(1, 2, 100), +, -, \*, /  # logical operations  >, <, >=, <=, ==, !=, |, & | Summary Tables # compute five-number summary  favstats(~ Y, data = data\_set)  # create frequency table tally(data\_set$Y)  tally(~ Y, data = data\_set)  # tally by condition  tally(~ Y < 1900, data = data\_set)  # two-way frequency table  tally(Y ~ X, data = data\_set) | | Simulation # sample without replacement sample(data\_set, 6)  # sample with replacement resample(data\_set, 10)  # randomize sampling distribution  # of b1s, centered on 0  sdob1 <- do(1000) \*  b1(shuffle(Y) ~ X, data = data\_set)  # bootstrap sampling distribution  # of b1s, centered on sample b1 sdob1\_boot <- do(1000) \*  b1(Y ~ X, data = resample(data\_set)) | | | # return TRUE for  # middle 95% of distribution  middle(sdob1$b1, .95)  # randomize sampling distribution  # of PREs  sdoPRE <- do(1000) \*  PRE(shuffle(Y)~ X,  data = data\_set)  # randomize sampling distribution  # of Fs  sdoF <- do(1000) \*  fVal(shuffle(Y)~ X, data = data\_set) | |
| Data Frame # view first/last six rows head(data\_set)  tail(data\_set)  # structure of data frame str(data\_set) glimpse(data\_set)  # select variable (a column) data\_set$Y | # select multiple variables select(data\_set, Y1, Y2)  # select first row  data\_set[1, ]  # find rows that meet condition  data\_set[data\_set$Y > 40]  filter(data\_set, Y > 300) | | | # arrange rows by variable arrange(data\_set, Y)  # sort in a descending order arrange(data\_set, desc(Y))  # get rid of all cases with any  # missing values  na.omit(data\_set) | | | # convert quantitative variable  # to categorical  factor(data\_set$Y)  # convert categorical variable  # to quantitative as.numeric(data\_set$Y) |
| Fitting Models to Data # empty model  empty\_model <- lm(Y ~ NULL, data = data\_set)  # use one expanatory variable  one\_model <-  lm(Y ~ X, data = data\_set)  # extract the best fitting b1  b1(shuffle(Y) ~ X, data = data\_set)  # multivariate model  multi\_model <-  lm(Y ~ X1 + X2, data = data\_set)  # model predictions and residuals  data\_set$empty\_predict <- predict(empty\_model)  data\_set$empty\_resid <- resid(empty\_model) | | Comparing Models pre(Y ~ X, data = data\_set)  f(Y ~ X, data = data\_set)  # sample F for X2  f(Y ~ X1 + X2,  data = data\_set, predictor = ~X2)  # all the model comparisons that can be  # made in relation to the multivariate model generate\_models(multi\_model) | | | Evaluating Models of DGP # produce ANOVA table supernova(empty\_model) supernova(multi\_model)  # t-test, using pooled variance  t.test(Y ~ X, data =  data\_set, var.equal=TRUE)  # confidence interval  confint(lm(Y ~ X,  data = data\_set))  # pairwise comparison  # corrections: "Bonferroni" or "none" pairwise(one\_model, correction = "none") | | |
| Visualizations gf\_histogram(~ Y, data = data\_set) %>%  # change labels    gf\_labs(title = "Graph Title", x = "Y\_Name", y = "Frequency")  Chart, histogram  Description automatically generated  # faceted grid of histograms  gf\_histogram(~ Y, data = data\_set) %>%    gf\_facet\_grid(X ~ .)  Chart, histogram  Description automatically generated  gf\_histogram(~ Y, data = data\_set) %>%   gf\_facet\_grid(X ~ Z)  Chart, histogram  Description automatically generated | | gf\_boxplot(Y ~ X, data = data\_set)  Chart, box and whisker chart  Description automatically generated  gf\_jitter(Y ~ X, data = data\_set)  Chart, scatter chart  Description automatically generated  gf\_point(Y ~ X, data = data\_set) %>%  # add model predictions as red points   gf\_point(Y ~ X , shape = 1, size = 3,  color = "firebrick") %>%  # add best fitting model as a red line   gf\_model(one\_model, color = “red”)  Chart, scatter chart  Description automatically generated | | | gf\_point(Y ~ X, color = ~Z,  data = data\_set)  Chart, scatter chart  Description automatically generated  # sampling distribution of b1  gf\_histogram(~b1, data = sdob1,  fill = ~middle(b1, .95)) %>%  # modify the limits on x- and y-axes   gf\_lims(x = c(-12, 12), y = c(0, 70))  Chart, histogram  Description automatically generated  # F-distribution depicting p-value  sample\_F <- fVal(Y ~ X, data = data\_set)  xpf(sample\_F, df1 = 1, df2 = 42)  Chart  Description automatically generated | | |