|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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(~ PriceK, data = Ames)  # create frequency table tally(TipExperiment$Condition)  tally(~ Condition,  data = TipExperiment)  # tally by condition  tally(~ YearBuilt < 1900,  data = Ames)  # two-way frequency table  tally(GarageCars ~ GarageType,  data = Ames) | | Simulation # sample without replacement sample(TipExperiment, 6)  # sample with replacement resample(TipExperiment, 10)  # randomize sampling distribution  # of b1s, centered on 0  sdob1 <- do(1000) \*  b1(shuffle(Tip) ~ Condition,  data = TipExperiment)  # bootstrap sampling distribution  # of b1s, centered on sample b1 sdob1\_boot <- do(1000) \*  b1(Tip ~ Condition,  data = resample(TipExperiment)) | | | # return TRUE for  # middle 95% of distribution  middle(sdob1$b1, .95)  # randomize sampling distribution  # of PREs  sdoPRE <- do(1000) \*  PRE(shuffle(Tip)~ Condition,  data = TipExperiment)  # randomize sampling distribution  # of Fs  sdoF <- do(1000) \*  fVal(shuffle(Tip)~ Condition,  data = TipExperiment) | |
| Data Frame # view first/last six rows head(TipExperiment) tail(TipExperiment)  # structure of data frame str(TipExperiment) glimpse(TipExperiment)  # select variable (a column) TipExperiment$Tip | # select multiple variables select(Ames, PriceK, PriceR)  # select first row  TipExperiment[1, ]  # find rows that meet condition TipExperiment[TipExperiment$Tip > 40]  filter(Ames, PriceK > 300) | | | # arrange rows by variable arrange(TipExperiment, Tip)  # sort in a descending order arrange(TipExperiment, desc(Tip))  # get rid of all cases with any  # missing values  na.omit(Ames) | | | # convert quantitative variable  # to categorical factor(Ames$HasCentralAir)  # convert categorical variable  # to quantitative as.numeric(Ames$HasCentralAir) |
| Fitting Models to Data # empty model  empty\_model <- lm(PriceK ~ NULL, data = Ames)  # use one expanatory variable Neighborhood\_model <-  lm(PriceK ~ Neighborhood, data = Ames)  # extract the best fitting b1  b1(PriceK ~ Neighborhood, data = Ames)  # multivariate model  multi\_model <-  lm(PriceK ~ Neighborhood + HomeSizeK,  data = Smallville)  # model predictions and residuals  Ames$empty\_predict <- predict(empty\_model)  Ames$empty\_resid <- resid(empty\_model) | | Comparing Models pre(Tip ~ Condition, data = TipExperiment)  f(Tip ~ Condition, data = TipExperiment)  # sample F for HomeSizeK  f(PriceK\_N\_resids ~ Neighborhood + HomeSizeK,  data = Smallville, predictor = ~HomeSizeK)  # 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(Tip ~ Condition, data =  TipExperiment, var.equal=TRUE)  # confidence interval  confint(lm(Tip ~ Condition,  data = TipExperiment))  # pairwise comparison  # corrections: "Bonferroni" or "none" pairwise(game\_model, correction = "Tukey") | | |
| Visualizations gf\_histogram(~ PriceK, data = Ames) %>%  # change labels    gf\_labs(title = “Distribution of Home Sale  Prices in Ames”, x =  “Home Sale Price  (Thousands of $)”, y = “Frequency”)  Chart, histogram  Description automatically generated  # faceted grid of histograms  gf\_histogram(~ PriceK, data = Ames) %>%    gf\_facet\_grid(Neighborhood ~ .)  Chart, histogram  Description automatically generated  gf\_histogram(~ PriceK, data = Ames) %>%   gf\_facet\_grid(Neighborhood ~ Floors) | | gf\_boxplot(PriceK ~ Neighborhood, data = Ames)  Chart, box and whisker chart  Description automatically generated  gf\_jitter(PriceK ~ Neighborhood, data = Ames)  Chart, scatter chart  Description automatically generated  gf\_point(PriceK ~ HomeSizeK, data = Ames) %>%  # add model predictions as red points   gf\_point(prediction ~ HomeSizeK , shape = 1,  size = 3,  color = “firebrick”) %>%  # add best fitting model as a red line   gf\_model(HomeSizeK\_model, color =  “red”)  Chart, scatter chart  Description automatically generated | | | gf\_point(FEV ~ HEIGHT, color = ~Age,  data = fevdata)  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(Tip ~ Condition,  data = TipExperiment)  xpf(sample\_F, df1 = 1, df2 = 42)  Chart  Description automatically generated | | |