Exam R Cheat Sheet

|  |  |  |
| --- | --- | --- |
| Chapter | Topic | Code |
| 2 | Data – load existing datasets | data(*dataset name*) |
| 2 | Frequency tables | table(*column name*) |
| 2 | Frequency tables as percentages | table(*column name*) / length(*column name*) \* 100 |
| 2 | Grouped frequency table | stem(*column name*, scale = *#*) |
| 2 | Ggplot2 | library(ggplot2) |
| 2 | Histograms | myplot = ggplot(*dataset*, aes(*column name*))  myplot +  geom\_histogram(binwidth = #) |
| 2 | Frequency Polygons | myplot = ggplot(*dataset*, aes(*column name*))  myplot +  geom\_freqpoly(binwidth = #) |
| 3 | Scatterplots | myplot = ggplot(*dataset*, aes(*X axis column, Y axis column*))  myplot +  geom\_point() +  geom\_smooth(method =”lm”) |
| 3 | Theme coding for plots | theme = theme(panel.grid.major = element\_blank(),  panel.grid.minor = element\_blank(),  panel.background = element\_blank(),  axis.line.x = element\_line(colour = "black"),  axis.line.y = element\_line(colour = "black"),  legend.key = element\_rect(fill = "white")) |
| 3 | X and Y axis labels | xlab(“Text goes here”)  ylab(“Text goes here”) |
| 3 | Bar graph | myplot = ggplot(*dataset*, aes(*X axis column, Y axis column*))  myplot +  stat\_summary(fun.y = mean,  geom = "bar",  fill = "White",  color = "Black") +  stat\_summary(fun.data = mean\_cl\_normal,  geom = "errorbar",  position = position\_dodge(width = 0.90),  width = 0.2) |
| 4 | Combine data | mycolumn = c(*#, #, #*) |
| 4 | Mean | summary(*column name*)  mean(*column name*, na.rm = T) |
| 4 | Median | summary(*column name*)  median(*column name*, na.rm = T) |
| 4 | Mode | temp <- table(as.vector(*column*))  names(temp)[temp == max(temp)] |
| 4 | Range | summary(*column name*)  max(*column*, na.rm = T) – min(*column*, na.rm = T) |
| 4 | Unbiased variance | var(*column name*, na.rm = T) |
| 4 | Unbiased standard deviation | sd(*column name,* na.rm = T) |
| 4 | Biased variance | pop.var <- function(x) var(x) \* (length(x)-1) / length(x)  pop.sd <- function(x) sqrt(pop.var(x))  pop.var(*column name*) |
| 4 | Biased standard deviation | pop.var <- function(x) var(x) \* (length(x)-1) / length(x)  pop.sd <- function(x) sqrt(pop.var(x))  pop.sd(*column name*) |
| 4 | IQR | summary(*column name*)  IQR(*column name*, na.rm = T) |
| 6-7 | pnorm | pnorm(*Z score*, lower.tail = F) |
| 6-7 | qnorm | qnorm(*p value*, lower.tail = F) |
| 8 | Confidence interval | M + qnorm(*p value*, lower.tail = F)  M – qnorm(*p value,* lower.tail = F) |
| 8 | Power | ##enter numbers here  popmean = 80  popsd = 30  N = 5  alpha = .05  samplemean = 120  lower = F  ##auto calculate  popse = popsd / sqrt(N)  mneed = popmean + popse\*qnorm(alpha, lower.tail = lower)  z = (mneed - samplemean)/popse  pnorm(z, lower.tail = lower) |
| 9-11 | SD / SE | sd = sd(*dataset$column, na.rm = T*)  se = sd(*dataset$column*) / sqrt(length(*dataset$column*))  spooled = spooled = sqrt( ((n1-1)\*sd1^2 + (n2-1)\*sd2^2) / (n1+n2 - 2))  sdifference = sqrt((spooled^2/n1 + spooled^2/n2)) |
| 9-11  15-16 | Critical Scores | * Less test:   + qt(.05, df, lower.tail = T) * Greater test:   + qt(.05, df, lower.tail = F) * Difference test:   + qt(.05/2, df, lower.tail = T) |
| 9 | Single t | t.test(*dataset$column*,  mu = #,  alternative = “less” OR “greater” OR “two.sided”,  conf.level = .95 OR .99) |
| 9-11 | Confidence intervals | Lower = Msample – tcritical\*SE  Upper = Msample + tcritical\*SE |
| 9-11 | Cohen’s d | d.singlet(m = 10, u = 12, sd = 4, n = 20, a = .10, k = 3)  d.deptdiff(mdiff = .857, sddiff = 1.07, n = 7, a = .05, k = 2)  d.indt(m1 = 10, m2 = 20, sd1 = 4, sd2 = 5, n1 = 10, n2 = 10, a = .05, k = 2) |
| 10 | Dep t | difference = *data$column – data$column*  t.test(*data*$*column,*  *data$column,*  paired = T,  alternative = “less” OR “greater” OR “two.sided”,  conf.level = .95 OR .99) |
| 11 | Ind t | t.test(*data*$*column,*  *data$column,*  paired = F,  var.equal = T,  alternative = “less” OR “greater” OR “two.sided”,  conf.level = .95 OR .99) |
| 12 | ANOVA  restructure data | library(reshape)  longdata = melt(*dateset*,  measured = c("*column*", "*column*", "*column*"))  longdata$partno = 1:nrow(longdata)  longdata = na.omit(longdata) |
| 12 | ANOVA  Run the ANOVA | library(ez)  ezANOVA(data = longdata,  dv = value,  between = variable,  wid = partno,  type = 3,  return\_aov = T) |
| 12 | ANOVA  Critical score | qf(.05, *dfn*, *dfd*, lower.tail = F) |
| 12 | ANOVA  Post hoc | pairwise.t.test(longdata$value,  longdata$variable,  paired = F,  var.equal = T,  p.adjust.method = "bonferroni") |
| 15 | Correlation test | cor.test(*dataset*$*column*, *dataset*$*column*) |
| 16 | Regression test | output = lm(*DV* ~ *IV*, data = *dataset*)  summary(output) |
| 16 | Beta | install.packages("QuantPsyc")  library(QuantPsyc)  lm.beta(output) |
| 16 | Multiple regression graph | predicted = output$fitted.values  myplot = ggplot(*dataset*, aes(predicted*, Y axis column*))  myplot +  geom\_point() +  geom\_smooth(method =”lm”) |