

Common Sta 101 Commands for R

1 One quantitative variable

Summary statistics

```
summary(x)
# most summary statistics at once
mean(x)
# na.rm = TRUE to get rid of NA values
median(x)
# na.rm = TRUE to get rid of NA values
sd(x)
# na.rm = TRUE to get rid of NA values
```

Visualization

```
hist(x)
boxplot(x)
# horizontal = TRUE for horizontal plot
qqnorm(x)
qqline(x)
# for normal probability plot and straight line
```

2 One categorical variable

Summary statistics

```
table(x)
```

Visualization

```
barplot(table(x))
```

3 Two categorical variables

Summary statistics

```
table(x,y)
```

Visualization

```
barplot(table(x,y))  
  # beside = TRUE for side-by-side barplot  
  # legend = TRUE to include a color legend  
mosaicplot(table(x,y))
```

4 One categorical and one quantitative variable

y = quantitative
x = categorical

Summary statistics

```
by(y, x, summary)  
  # summary by group  
by(y, x, mean)  
  # mean by group  
  # na.rm = TRUE to get rid of NA values  
by(y, x, sd)  
  # sd by group  
  # na.rm = TRUE to get rid of NA values
```

Visualization

```
boxplot(y ~ x)
```

5 Two quantitative variables, Simple linear regression

Note: Out of scope for project 1.

Summary statistics

```
cor(x,y)  
  # use = "complete.obs" to get rid of NA values  
slr = lm(y ~ x)  
summary(slr)
```

```
# linear model and the model output
```

Visualization

```
plot(y ~ x)
```

6 Multiple linear regression

```
mlr = lm(y ~ x1 + x2 + ...)  
summary(mlr)  
# linear model and the model output
```

7 Regression diagnostics

```
# in the code below m is the regression model  
plot(m$residuals ~ x)  
# residuals vs. an explanatory variable  
plot(m$residuals ~ m$fitted)  
# residuals vs. fitted (predicted) values of y from the model  
plot(m$residuals)  
# residuals vs. order of data collection  
hist(m$residuals)  
# histogram of residuals  
qqnorm(m$residuals)  
qqline(m$residuals)  
# normal probability plot of residuals
```

8 Subsetting

```
subset(dataname, !is.na(x))  
# the data set "dataname", but only cases for which x is not NA  
subset(dataname, x == "levelA")  
# the data set "dataname", but only cases for which x is equal to "levelA"  
x[!is.na(x)]  
# the variable x, but only cases for which x is not NA  
y[!is.na(x)]  
# the variable y, but only cases for which x is not NA  
x[x < 30]  
# the variable x, but only cases for which x is less than 30
```

```
x[x != "levelA"]  
# the variable x, but only cases for which x does not equal "levelA"  
droplevels(x)  
# drops empty levels if you have removed all the cases from one level
```

9 Probability distributions

```
pnorm(q, mean, sd)  
# calculate area under the normal curve below q  
# for a normal distribution with given mean and sd  
dnorm(x, mean, sd)  
# calculate the normal probability density at x (can be a vector)  
# for a normal distribution with given mean and sd,  
# useful for plotting a normal curve over a histogram  
dbinom(x, size, prob)  
# calculate the probability for x successes in size trials,  
# where probability of success is prob
```

10 Plotting lines

```
abline(h = value)  
# add a horizontal line to an existing plot  
abline(v = value)  
# add a vertical line to an existing plot  
abline(lm(y~x))  
# overlays linear regression line on the scatterplot of y vs. x,  
# only works if plot(y ~ x) ran first
```

11 Sampling

```
sample(x, size, replace = FALSE)  
# sample from x size number of elements without replacement (default)  
# to sample with replacement replace = TRUE
```

12 Plotting options

These arguments can be passed to the `plot`, or `hist`, or other similar functions. To learn more about all plotting parameters, type `?par`.

```

main = "main title"
  # title of plot, to be placed in the top center
xlab = "x-axis label"
  # x-axis label
ylab = "y-axis label"
  # y-axis label
xlim = c(min,max)
  # x-axis limits
ylim = c(min,max)
  # y-axis limits

```

13 inference function

Use the following command to load the inference function:

```
source("http://stat.duke.edu/~mc301/R/inference.R")
```

```

inference(y, x, est, type, method, null, alternative, success, order, conflevel, siglevel,
          nsim, eda_plot, inf_plot, sum_stats)
  # y = response variable, categorical or numerical variable
  # x = explanatory variable, categorical (optional)
  # est = "mean", "median", or "proportion"
  # type = "ci" for confidence interval, or "ht" for hypothesis test
  # method = "theoretical" or "simulation"
  # null = (optional) null value, does not need to be defined for chi-square or ANOVA
  # alternative = (optional) "less", "greater", or "twosided"
  # success = (optional) if data is categorical, the name of the level that is
  #           defined as success
  # order = (optional) if group is defined, the order in which to subtract the groups
  # conflevel = (optional) for confidence intervals, 0.95 by default,
  #            can be any number between 0 and 1
  # siglevel = (optional) for hypothesis testing, 0.05 by default,
  #            can be any number between 0 and 1
  # nsim = (optional) number of simulations, 10000 by default,
  #        use lower if sample size is too high and simulations take a long time
  # eda_plot = (optional) FALSE to hide the exploratory analysis plot
  #           default: TRUE
  # inf_plot = (optional) FALSE to hide the inference plot
  #           default: TRUE
  # sum_stats = (optional) FALSE to hide summary statistics
  #           default: TRUE

```