Common Sta 101 Commands for R

1 inference function

Use the following command to load the inference function:

```
source("http://stat.duke.edu/courses/Fall12/sta101.001/labs/inference.R")
```

```
inference(data, group, est, type, method, null, alternative, success, order, conflevel,
siglevel, nsim)
  # data = response variable, categorical or numerical variable
  # group = 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
```

2 One quantitative variable

Summary statistics

```
summary(x)
  # most summary statitstics 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
```

3 One categorical variable

Summary statistics

```
table(x)
```

Visualization

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

4 Two quantitative variables

Note: Out of scope for project 1.

Summary statistics

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

Visualization

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

5 Two categorical variables

Summary statistics

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

Visualization

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

6 One categorical and one quantitative variable

y = quantitative
x = categorical

Summary statistics

```r
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

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

7 Subsetting

```r
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
```

8 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
```

9 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
```

10 Sampling

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

11 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
```