# Common R commands used in Data Analysis and Statistical Inference

# Data Analysis and Statistical Inference

## One numerical 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
```

## One categorical variable

Summary statistics

```
table(x)
```

Visualization

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

# 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))
```

# One categorical and one numerical variable

`y` = numerical `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)
```

# inference function

Use the following command to load the inference function:

```
source("http://bit.ly/dasi_inference")
```

```
inference(y, x, est, type, method, null, alternative, success, order, conflevel, s
iglevel,
          nsim, eda_plot, inf_plot, sum_stats)
  # y = response variable, categorical or numerical variable
  # x = explanatory variable, categorical (optional)
  # est = parameter to estimate: "mean", "median", or "proportion"
  # type = "ci" for confidence interval, or "ht" for hypothesis test
  # method = "theoretical" for CLT based,
  #       or "simulation" for simulation based (randomizatio/bootstrapping)
  # null = (optional) null value for a hypothesis test,
  #       does not need to be defined for chi-square or ANOVA
  # alternative = (optional) direction of the alternative hypothesis:
  #       "less", "greater", or "twosided"
  # success = (optional) if the response variable is categorical,
  #       the name of the level that is defined as success
  # order = (optional) if the explanatory variable is defined,
  #       the order of levels of the explanatory variable in which to subtract gro
ups
  # conflevel = (optional) for confidence intervals, default conflevel = 0.95
  # siglevel = (optional) for hypothesis testing, takes values between 0 and 1,
  #       default siglevel = 0.95
  #       (used only for ANOVA to determine if posttests are necessary)
  # boot_method: (optional) set method for bootstrap interval:
  #       "perc" for percentile, or "se" for standard error
  # nsim = (optional) number of simulations, default nsim = 10000,
  #       decrease number of simulations if simulations take too long
  # seed = set a seed for simulations
  # useful for space savings:
  #   sum_stats = (optional) TRUE/FALSE - print summary stats
  #   eda_plot = (optional) TRUE/FALSE - print EDA plot
  #   inf_plot = (optional) TRUE/FALSE - print inference plot
  # not of much use for regular user:
  #   simdist = (optional) TRUE/FALSE - return the simulation distribution
  #   inf_lines = (optional) TRUE/FALSE - print lines on the inference plot
  #               for ci bounds or p-value
```

# Two numerical 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)
```

# Multiple linear regression

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

# 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
```

# 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
```

## 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
```

## 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
```

# Sampling

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

# 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
```