Rotman

INTRO TO R PROGRAMMING

R Tutorial (RSM358) – Session 2

January 20, 2026 Prepared by Jay Cao / MDAL

Website: https://rmdal.github.io/r-intro-2026-winter/

Rotman School of Management
UNIVERSITY OF TORONTO

Any Questions about Lab 2.3 & A1 Coding?

- Loading the Auto data? (review the 3 options)
  - Option 1: From the ISLR2 library
  - Option 2: From Auto.data (tab separated data)
  - Option 3: From Auto.csv (comma separated data)

- Removing rows with missing data?
  - `na.omit()`

Any Questions about Lab 2.3 & A1 Coding?

- `summary()`, `range()`, `mean()`, `sd()` and dataframe column slicing
  - `Auto['mpg'] vs Auto$mpg vs Auto[['mpg']]`
  - `Auto[c('mpg', 'horsepower')]`
  - `range(Auto['mpg']) vs range(Auto$mpg)`
  - `range(Auto['mpg']) vs range(Auto[c('mpg', 'horsepower')]) ???`
  - How about `mean()`

- Knowing the type of an object is IMPORTANT
  - A functions takes in certain types of objects and returns certain types of objects
    - i.e., need to know it to invoke the function correctly
  - Basic Data Structures slides and its companion notebook from last session
    - `str()`, `typeof()`, `class()`

Any Questions about Lab 2.3 & A1 Coding?

- `Auto[1:3] vs Auto[1:3, ]`
- `Auto[1:3, ] vs Auto[-(1:3), ] vs Auto[-c(1:3), ]`

- More on dataframe slicing
  - Basic Data Structures notebooks from last session

- `plot()`
- `pairs()`

Any Questions about Lab 2.3 & A1 Coding?

- Data Simulation Example 1 (from Lab 2.3)

```
x <- rnorm(50)
y <- x + rnorm(50, mean = 50, sd = .1)
cor(x, y)
```

- Data Simulation Example 2 (from 3.7)

```
set.seed (1)
x1 <- runif (100)
x2 <- 0.5 * x1 + rnorm (100) / 10
y <- 2 + 2 * x1 + 0.3 * x2 + rnorm (100)
```

Lab 3.6 Linear Regression

- `my_lm <- lm(formula = …, data = …)`
- `plot()`
  - Two variable scatter plot: `plot(x, y)`
  - Regression line: `abline(my_lm)`
  - Post-regression diagnostic plot: `plot(my_lm)`
- `predict(object, new_data, interval, level=0.95)`
  - Confidence interval
    - `predict(my_lm, data.frame(x1 = (c(5, 10))), interval = "confidence")`
  - Prediction interval
    - `predict(my_lm, data.frame(x1 = (c(5, 10))), interval = "prediction")`

lm() R Regression Formula - 1

my_df

y	x1	x2	x3
18	8	307	130
16	8	304	150
…	…	…	…

lm() Regression Formula	Regression Formula

`lm(formula = y ~ x1 + x2 + x3, data = my_df)`
 	
`lm(formula = y ~ ., data = my_df)`
	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$

`lm(formula = y ~ . - x3, data = my_df)`
 	
`lm(formula = y ~ x1 + x2, data = my_df)`
	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$

`lm(formula = y ~ 0 + x1 + x2 + x3, data = my_df)`	$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$
…	…

Source: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/formula.html

lm() R Regression Formula - 2

my_df

y	x1	x2	x3
18	8	307	130
16	8	304	150
…	…	…	…

lm() Regression Formula	Regression Formula

`lm(formula = y ~ x1 * x2, data = my_df)`
 	
`lm(formula = y ~ x1 + x2 + x1:x2, data = my_df)`
 	
`lm(formula = y ~ x1 + x2 + I(x1 * x2), data = my_df)`
	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$

`lm(formula = y ~ x1 + I(x1^2), data = my_df)`	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \epsilon$
`lm(formula = y ~ x1 + log(x2), data = my_df)`	$Y = \beta_0 + \beta_1 X_1 + \beta_2 \ln(X_2) + \epsilon$
…	…

Source: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/formula.html