Rotman

INTRO TO R PROGRAMMING

R Tutorial (RSM358) – Session 3

January 29, 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 3.6?

- Loading the data provided by the textbook
  - Easy option: via the `ISLR2` package
  - Alternatively: `csv` files on the textbook resource page (some are in the zipped file)

- Good practise: Before loading a csv data file, inspect it as a text file
  - This helps to determine the potential arguments of `read.csv()`
    - E.g., `header = FALSE` or `na.strings = "?"` or `stringsAsFactors = T`

- `factor` column/variable (categorical variable)
  - Why the factor data type?
  - Convert a non-factor column to a factor column
    - E.g. `auto$origin = as.factor(auto$origin)`

Any Questions about Lab 3.6?

- Linear regression?

- Comment on the regression summary report (section 3.4)
  - Is there a relationship, how strong, positive or negative
  - Confidence and prediction interval

- Interpretation of coefficients
  - Section 3.1 and 3.2 for quantitative/continuous predictors
  - Section 3.3.1 for qualitative/categorical predictors

- Outlier and high leverage observations
  - Use post-regression diagnostic plot; section 3.3.3

Linear Regression

- `my_lm <- lm(formula = …, data = …)`
  - `E.g., my_lm <- lm(formula = y ~ x, data = my_data)`

- `summary(my_lm)`

- `plot()`
  - Two variable scatter plot: `plot(my_data$x, my_data$y)`
  - Regression line: `abline(my_lm)`
  - Post-regression diagnostic plot: `plot(my_lm)`

- `predict(object, new_data, interval, level=0.95)`
  - Confidence interval
    - `E.g., predict(my_lm, data.frame(x = (c(5, 10))), interval = "confidence")`
  - Prediction interval
    - `E.g., predict(my_lm, data.frame(x = (c(5, 10))), interval = "prediction")`

Create Data Frame using `data.frame()`

```
# create a data frame
df1 <- data.frame(
  x = 1:3,
  y = letters[1:3],
  z = c(1.1, 2.2, 3.3)
)
```

x	**y**	**z**
1	"a"	1.1
2	"b"	2.2
3	"c"	3.3

Understanding CI and PI - Setup

- True model, true Y, and true coefficients β_i,

$$Y = f(X) + \epsilon = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon, \text{ where } \mathbb{E}(\epsilon) = 0.$$

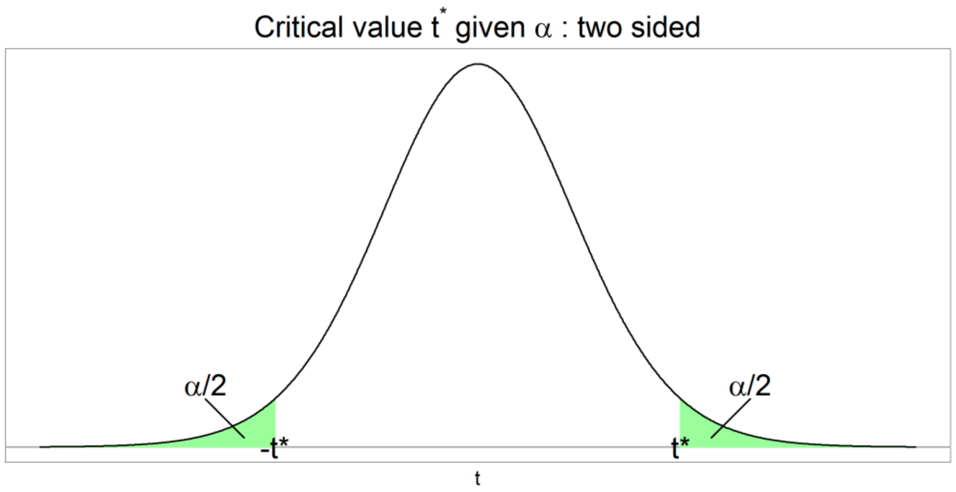
- Estimated model, predicted $\hat{Y}$, and estimated coefficients $\hat{\beta}_i$,

$$\hat{Y} = \hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$$

- R code

```
my_lm <- lm(formula = y ~ x1 + x2, data = my_df)
```

CI of β_i



- Under "usual/standard" assumptions,

$$\frac{\widehat{\beta}_i - \beta_i}{\widehat{SE}(\widehat{\beta}_i)} \sim t_{n-p},$$

So,

$$\mathbb{P}\left(-t_{n-p,\frac{\alpha}{2}} < \frac{\widehat{\beta}_i - \beta_i}{\widehat{SE}(\widehat{\beta}_i)} < t_{n-p,\frac{\alpha}{2}}\right) = 1 - \alpha,$$

Now, we get CI (centered around $\hat{\beta}_i$)

$$[\hat{\beta}_i - \widehat{SE}(\hat{\beta}_i) t_{n-p,\frac{\alpha}{2}}, \hat{\beta}_i + \widehat{SE}(\hat{\beta}_i) t_{n-p,\frac{\alpha}{2}}]$$

- Interpretation (e.g., $\alpha = 95\%$)?
  - "If we take repeated samples and construct the confidence interval for each sample, 95% of the intervals will contain the true unknown value of the parameter." – from you textbook

- R code: `confint(my_lm)`

Notations: t_{n-p} is t-dist w/ df $n-p$, n is # of obs., and p is # of parameters; α is significance level and $t_{n-p,\frac{\alpha}{2}}$ is critical value (the t^* in the graph) such that the prob of the t_{n-p} distribution to the right of it is $\frac{\alpha}{2}$; $\widehat{SE}(\hat{\beta}_i)$ is $SE(\hat{\beta}_i)$ in your textbook.

CI of $\mathbb{E}(Y) = f(X)$

- α level CI of $\mathbb{E}(Y) = f(X)$ at $X = x_0$ can be derived similarly
  - It's an interval centered at $\hat{f}(X = x_0)$

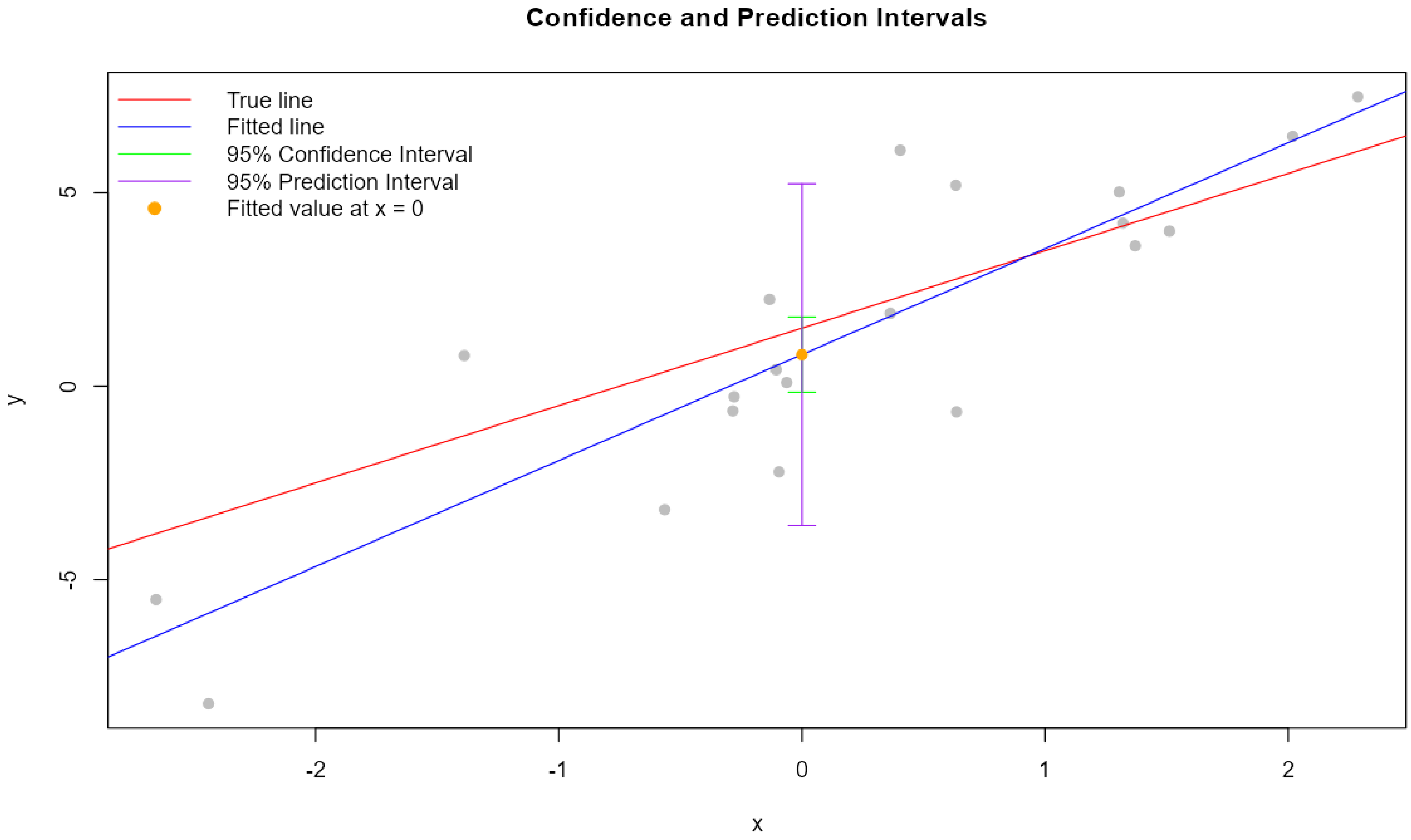
- Interpretation (say, $\alpha = 95\%$): "95% of intervals **of this form** will contain the true value of $f(X)$" – from your textbook
  - To be precise, $f(X = x_0)$
  - What does it mean "of this form"?
    - If we take repeated samples and construct the confidence interval for each sample, …

- R code
  - `predict(my_lm, new_data, interval = "confidence")`

PI of $Y = f(X) + \epsilon$

- α level PI of $Y = f(X) + \epsilon$ at $X = x_0$ can be derived similarly
  - Again, it's an interval centered at $\hat{f}(X = x_0)$
- Wider than the corresponding CI
  - because it accounts for the irreducible error
- Interpretation?
- R code
  - `predict(my_lm, new_data, interval = "prediction")`

CI of $\mathbb{E}(Y)$ and PI of Y at $X = 0$ and $\alpha = 95\%$



Note: Data are simulated with the true model being $Y = 1.5 + 2X + \epsilon$ where $\epsilon \sim N(0,2)$.

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

Lab 4.7 Logistic Regression

- `my_model <- glm(formula = …, data = …, family=binomial)`
- `summary(my_model)`

- `predict(my_model, newdata = …, type = "response")`
  - Set the argument `type = "response"` to get predicted probabilities, i.e., $P(Y = 1|X)$
  - Otherwise, `predict(my_model)` gives log odds (logit)
  - If the `newdata` argument is not supplied, the prediction is applied on the training data set
  - Use `contrast()` to find out which y category is set to 1.

- Construct confusing matrix
  - Convert probability prediction to binary prediction (cutoff prob.)
  - `table()`

Lab 4.7 Training & Test Set

- Training and test set split
  - For time series data, need to respect the time when splitting the data
    - That is, train on early data, test on late data
  - Otherwise, randomly split data to train and test

```
# randomly split Auto dataset into training and test set
num_rows <- nrow(Auto)
train_fraction <- 0.7
train_idx = sample(1:num_rows, size = round(num_rows * train_fraction))
train_data <- Auto[train_idx, ]
test_data <- Auto[-train_idx, ]
```