1 Linear Regresion - Lecture 1

1.1 EDA tells us

knowing the size of the data —— examining distributions of all variables using graphical and numerical summaries —— identifying missing values and potential outliers —— beginning to discover relationships between variables

1.2 Population vs. sample

```
set.seed(561)
dat_s <- sample_n(dat, 1000, replace = FALSE) #Sample
#Get summary statistics
dat_tax_SLR = dat_s |> select(assess_val, BLDG_METRE) |>
    gather() |>
    group_by(key) |>
    summarise(mean = mean(value, na.rm = TRUE),
              max = max(value),
              min = min(value),
              median = median(value, na.rm = TRUE),
              sd = sd(value, na.rm = TRUE))
```

2 Simple Linear Regression (SLR)

2.1 The conditional expectation

The conditional expected value is the best predictor given additional relevant information. SLR answers the question, given the size of property what can be the expected value?

2.2 Assumptions

Conditional expectation of the response is linearly related to the input variable and the line is the linear regression —— The random errors are independent and identically distributed: iid assumption —— The random errors have all the same variance: $Var(\epsilon_i) = \sigma^2$

$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$

β_0 is the population intercept, and β_1 is the population slope parameter measuring how Y changes with an associated change in X.

For each observation i, ε_i represents the random error term.

Intercept (β_0): The average value of Y when $X = 0$ is β_0

Slope (β_1): a one-unit increase in X (X) is associated with an expected increase of β_1 units in Y (Y).

Error Term (ϵ_i): Any distributional assumption made about the error term also affect the random variable Y (if normal, Y also normal)

3 Estimation of RL

3.1 Least squares estimation

Least Squares method minimizes the sum of the squares of the residuals. The residuals are the difference between the observed value of the response (y_i) and the predicted value of the response ($\hat{y}_i$) as $r_i = y_i - \hat{y}_i$

4 SLR with continuous variables

```
SL_reg <- lm(col1 ~ col2, data = data_frame)
tidy_SL_reg <- tidy(SL_reg) #Get estimate, std.error, p value
lm(response ~ .,data= df) # uses all variables in the df, except
     the response, as predictors
lm(response ~ input - 1,data= df) #forces the estimated intercept
     to be 0.
beta_0_hat = SL_reg$coef[1] # Extract coefficients
beta_1_hat = SL_reg$coef[2]
```

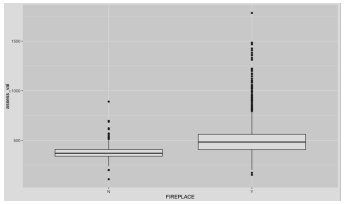
4.1 Estimated intercept β_0

Not interested because, the intercept value does not make sense for the model. many statistical properties do not hold for models without an intercept

5 SLR with categorical predictors

You have to use t-test, permutation test, or bootstrapping test to answer this kind of question. We use boxplots. X-axis is not numeric. We can still use this variable in a Linear model. We can reproduce the results of a t-test using lm.

```
Condition_plot <- dat_s %>% ggplot(aes(Condition, assess_val)) +
     geom_boxplot()
t.test(assess_val~Condition,dat_s,var.equal=T)
lm_F <- lm(assess_val~Condition,dat_s)
tidy(lm_F)
```



5.1 The conditional expectation

- the best predictor of X with a Condition: $E[Y \mid \text{Condition} = \text{Y}] = \mu_1$
- the best predictor of X without Condition: $E[Y \mid \text{Condition} = \text{N}] = \mu_0$
- and a two-sample t-test tests the difference between group means!!

$$H_0 : \mu_1 = \mu_0, \quad \text{or equivalently} \quad H_0 : \mu_1 - \mu_0 = 0$$

Since Condition is not numeric, we cant use in math formula, we instead use The dummy variable:

$$X_2 = \begin{cases} 1 & \text{if Condition} = \text{Y}, \\ 0 & \text{if Condition} = \text{N} \end{cases}$$

$$E[Y \mid X_2] = \beta_0 + \beta_2 X_2$$

- if Condition = N: $E[Y \mid X_2 = 0] = \beta_0$
- if Condition = Y: $E[Y \mid X_2 = 1] = \beta_0 + \beta_2$

Then

$$\beta_2 = E[Y \mid X_2 = 1] - E[Y \mid X_2 = 0] = \mu_1 - \mu_0$$

$H_0 : \beta_2 = 0$ is the same as the null hypothesis from the two-sample t-test.

5.2 The estimated intercept

It is the sample version of the conditional expectation (mean of the reference group)

5.3 The estimated slope

It is the sample version of the difference of the conditional expectations (or group means)

6 Uncertainty in the estimation - CI

We can think that the estimates are a (good) guess about the population parameters based on our data. However, the values of the estimates depend on the random sample used to compute them:

6.1 The standard errors

The variation of these estimates from sample to sample is measured by their standard deviation, which has a special name: the standard error (SE). In practice, we have different ways to measure the sample-to-sample variation of the estimated coefficients:

(i). take multiple samples from the population and compute multiple estimates as we did above, then compute their SE (but this is not a realistic option).

(ii). use a theoretical result (this is what lm does)

(iii). use bootstrapping as you did in DSCI 552 for other quantities!

7 Sampling Distribution

7.1 1-Bootstrapping

Bootstrapping refers to sampling from our original sample with replacement (also called resampling with replacement) to generate a many estimates and measure the sample-to-sample variation.

```
set.seed(0561)
lm_boot <- replicate(1000, {
  sample_n(dat_s, size = nrow(dat_s), replace = TRUE) %>%
    lm(X~Y,data=.) %>%
    .$coef
})
lm_boot <- data.frame(boot_intercept = lm_boot[1,], boot_slope =
     lm_boot[2,])
boot_s <- dat_s |>
    specify(assess_val~BLDG_METRE) |>
    generate(reps = 1000, type = 'bootstrap') |>
    calculate(stat = 'slope')
data.frame(coef_table %>% select(estimate),coef_table %>%
     select(std.error),
B_avg = lm_boot %>% summarize(intercept = mean(boot_intercept),
                              slope = mean(boot_slope)) %>%
                        round(3) %>% unlist(),
B_se = lm_boot %>% summarize(intercept = sd(boot_intercept),
                             slope= sd(boot_slope)) %>%
                       round(3) %>% unlist())
```

7.2 Using theoretical results - CLT

assumptions about the distribution of the error terms! —— the conditional distribution of the error terms is Normal, and so is the conditional distribution of the response! —— sample size is large —— lm uses this theoretical result to compute p-values and confidence intervals!

7.3 Confidence Intervals

```
tidy(lm_s, conf.int = TRUE)
quantile(slope_B,0.025) %>% round(2)
quantile(slope_B,0.975) %>% round(2)
(slope_s + qnorm(0.025) * sd(slope_B)) %>% round(2)
(slope_s - qnorm(0.025) * sd(slope_B)) %>% round(2)
```

7.4 Slope Coefficient Hypothesis

$H_0 : \beta_1 = 0$

$H_a : \beta_1 \neq 0$

From the regression output we get the following:

Slope estimate: tidy_SL_reg – estimate column

p-value: tidy_SL_reg – p-value column

If the p-value is far smaller than significance value α (0.05 usually):

p-value $< 0.05 \quad \Rightarrow \quad$ Reject H_0

Has is extremely strong statistical evidence that the slope coefficient differs from zero. We therefore conclude that col1 is a significant predictor of col2 in the population.

7.5 The range problem

The linear model assumes that the relationship between X and $E[Y—X]$ is linear, which may or may not be true

Sometimes, there's a linear association only in part of the data range. The linear model could still be useful when restricted to that specific range; We need to exercise caution when using the model outside the range of the data, as the relationship between X and Y may differ significantly.

8 Multiple Linear Regresion

$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$

$\varepsilon_i \sim N(0, \sigma^2) \quad$ for all i

$\text{Cov}(\varepsilon_i, \varepsilon_j) = 0 \quad$ for $i \neq j$

The MLR model is no longer a line, but a hyperplane in a $(p+1)$-dimensional space. In three dimensions (two predictors), it is a plane.

```
fit <- lm(assess_val ~ BLDG_METRE*FIREPLACE, data= dat_s)
ML_reg <- lm(col1 ~ col2 + col3 + col4, data = data_frame)
library(GGally)
ggpairs(data = dataFrame[, c("col1","col2","col3","col4",)])
```

8.1 "+" and "*" interaction in lm()

In an `lm()` formula, the `+` symbol adds a variable as a main effect only, meaning it includes the predictor in the model without forming any interaction terms. The `*` symbol expands to include both the main effects of the variables and their interaction term—for example, `x1 * x2` is equivalent to `x1 + x2 + x1:x2`. Thus, using `+` simply adds predictors, while using `*` fits a richer model that accounts for combined (interaction) effects between predictors.

```
# Examples in R:
lm(y ~ x1 + x2) # main effects only
lm(y ~ x1 * x2) # x1, x2, and interaction x1:x2
lm(y ~ x1:x2) # interaction only
format(3.904359e-02, scientific = FALSE) #convert exponents to dec
```

The bootstrap sampling distribution of the slope is approximately bell-shaped and symmetric, closely resembling a normal distribution. There are no major signs of skewness or heavy tails, indicating that the sampling distribution of the slope is well-approximated by the normal model commonly assumed in linear regression.