MONASH University

ETX2250/ETF5922: Data Visualization and Analytics

Linear Regression

Lecturer: *Lauren Kennedy*

Department of Econometrics and Business Statistics

lauren.kennedy1@monash.edu

Week 10

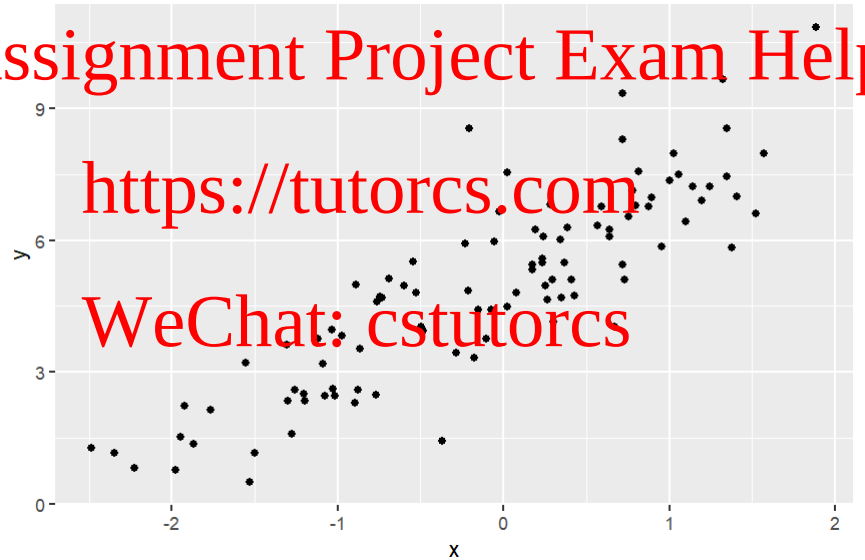
What we cover this week:

- An overview of regression
- Prediction vs Explanatory Models
- Assessing regression predictions
- Variable selection
- How to in R

Multiple linear Regression: An overview

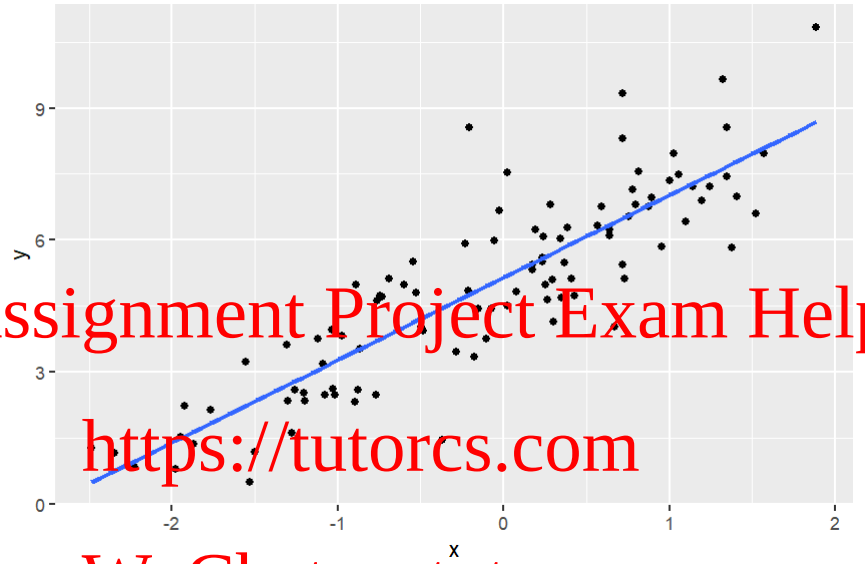
Simple linear regresion

- We have two variables, which we call x and y
- We want to understand the relationship between x and y
- We assume that relationship is linear.



- How would we answer this using summary values we've talked about before?

Fit a line



- Remember a line can be defined by:

$$y = mx + c$$

- In linear regression we use β_0 for the intercept term (c) and β_1 for the slope term m
- Note that this doesn't perfectly describe the data, so we add ϵ which describes residual unaccounted for variation

Inferenctial statistics

- Previously we've talked descriptive statistics. These are:
  - mean
  - standard deviation
  - median
  - correlation
- Now we're going to start to talk about *inferential* statistics. Now we want to move beyond to move describing the sample, and instead infer things about a true parameter.
- We estimate this with our sample. The estimate for β_0 denoted $\hat{\beta}_0$
- The value we calculate from our sample may be different if we were to have a different random sample. This means the $\hat{\beta}_0$ might be different in different samples.
- This means we have uncertainty when we estimate $\hat{\beta}_0$ and $\hat{\beta}_1$ a sample.

Least squares estimation

- How do we obtain this formula of a line from our observetd data?
- Assume we have a matrix X that looks like:

```
##   int           x
## 1   1  0.02140970
## 2   1 -0.07540132
## 3   1 -0.02550473
## 4   1 -2.34826490
## 5   1  0.26690802
## 6   1 -1.92429927
```

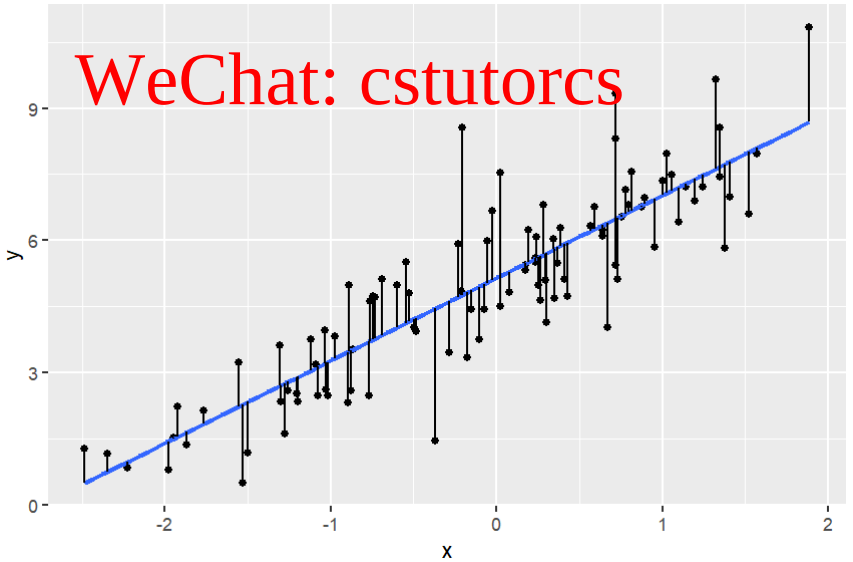

which represents all of our x observations, and a vector Y that has all of our y estimates. The estimate for $\{\hat{\beta}_0, \hat{\beta}_1$ is

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Minimising error

- This estimate minimizes the difference between the line of best fit and the observed data
- This difference is known as the residuals
- The thing we minimize is the sum of squared errors

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$



Representing parameter uncertainty: confidence intervals

- Every parameter estimate has a standard error associated
- We can create a **95% confidence interval** around a parameter estimate for $\hat{\beta}_0$ like this (assuming things like a large sample size)

$$(\hat{\beta}_0 - SE_{\beta_0} * 1.96, \hat{\beta}_0 + SE_{\beta_0} * 1.96)$$

- We say 95% of 95% confidence intervals contain the true parameter estimates

- The 1.96 comes from the .025 and .975 quantiles of a standard normal density function (.975-.025 = .95).

- We can have different levels of confidence by changing the quantile we calculate at. Common ones are:
  - 80% CI = 80% of all 80% confidence intervals contain the true parameter
  - 90% CI = 90% of all 90% confidence intervals contain the true parameter
- Which is wider?

Binary and factor level predictors

- What happens if x is binary?

$$y = \beta_0 + \beta_1 x_1$$

- Choose the 0 coded (or the first level if a factor with two levels) as a reference class. β_0 becomes the intercept for individuals in this reference class.
- β_0 the β_1 expected value where x_1 is 1.

Binary and factor level predictors

- What happens if x is a factor type variable?

$$y = \beta_0 + \beta_1 x_1$$

- The first level of x_1 is chosen as the reference class (there are other methods) + For l the number of levels in x_1 create $l-1$ dummy or indicator variables. I_1 if level 2, I_1 if level 3 etc

- The formula then becomes

$$y = \beta_0 + \beta_1 I_1 + \beta_2 I_2 + \ldots + \beta_{l-1} I_{l-1} + \epsilon$$

Multiple linear regression

- What if we have more than one predictor?
- We can assume an additive relationship and modify the model so it looks like this

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_m x_m + \epsilon$$

- Now the β_m term is interpreted as the increase we would see in y with a 1 unit increase in x_m whilst holding the value of $\{x_1, x_2, \ldots x_{m-1}\}$ constant.

- We can also have interaction terms (where the relation ship between y and x_1 moderated by x_2). We do not have time to cover this in depth.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$$

Assumptions

1. The noise parameter (ϵ) follows a normal distribution
2. The choice of predictors and their form is correct (linearity)
3. The observations are independent of each other
    - Describe a situation when they are not?
4. The variability of the outcome values are stable regardless of the values of the predictors (homoskedasticity)

Modelling aims: Explain vs Predict

Explain vs predict

- Explanatory models are concerned with estimating the average effect or relationship to the outcome
- A good explanatory model fits the data closely
- Explanatory models tend to use the entire dataset to fit the model, and focus on parameter estimates
- Predictive models are concerned with estimating the outcome for new observations.
- A good prediction model predicts new observations well
- Prediction models tend to be fit with a training dataset and tested on a test dataset (see week 6). A good prediction model predicts the test data well.

Representing uncertainty: prediction intervals

- The formula is a little more complex than a confidence interval so we won't cover it.
- However the interpretation is important. A prediction interval gives the uncertainty in our predictions of a new data point. A confidence interval gives us the uncertainty in our estimation of a parameter value.
- But some things remain the same. 95% of all prediction intervals should contain the actual future value.
- Interpret an 80% prediction interval.

Your turn!

Assessing Predictions

Violations to the assumptions of a linear regression

- Often when we teach linear regression, we devote considerable time to understanding how to check the assumptions of linear regression.
- We will spend some time in your tute investigating this, because it can be a useful way to understand these assumptions and what they mean.
- However, the assumption of following a normal distribution is not required when we consider a predictive model. This is because we do not need to calculate confidence intervals around the parameters.
- Instead, we simply need to predict, and by definition our estimates will have the smallest square error.
- This is not true for the other assumptions universally, but we can still obtain quite good predictions from models that violate predictions. B
- But what does it mean to be quite good?

Mean error

- Remember week 6 we talked about the error associated with classification. Now let's consider the error associated with regression
- For every data point $i$predict the outcome. Subtract this from the observed value and take the average of the differences.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$

- Here the sign of the error is retained, so we can obtain a sense of whether we consistently over or under predict.

Mean absolute error

- For every data point $i$predict the outcome. Subtract this from the observed value and take the average of the **absolute value** of differences.

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Here the sign of the error is **NOT** retained, so we can obtain a sense of overall magnitude of error.

Root mean square error

- For every data point $i$predict the outcome. Subtract this from the observed value and square it. Take the average of the squared differences, and then take the square root..

$$\sqrt{(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2)}$$

- This retains the same units as the outcome variable, which makes it easy to interpret

- These three measures can be used with training data and test data, just like with our measure of classification error.

Your turn!

Calulate:

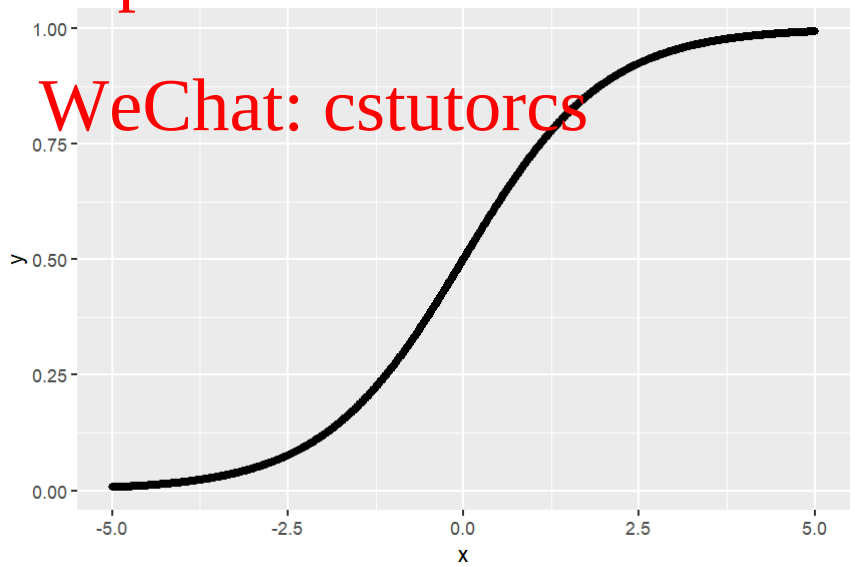
- Mean error
- Mean absolute error
- Root mean square error for the following:

predict	actual
10.2	23.3
34.6	35.7
24.3	25.7
12.3	15.4

What about classification

Logistic regression

- What if y could only take the values of 0 or 1?
- One option is to change the relationship of y and x so that we
  - Model $p(y = 1)$ rather than y directly
  - Model the relationship between $p(y = 1)$ and x using a non-linear relationship.
- One option is a logistic regression, which models the relationship using a logit link. This is one example of a wider family of regressions that can be used



Linear regression?

- You might see some people use linear regression even when y is binary.
- What is one challenge you would expect to see thinking about this from a prediction context?
- Oftentimes this is done in an explanatory context, where the probability of the outcome doesn't fall into extremely high or low probability.
- It might work in a prediction context, but much caution is needed!

Major challenges

What factors should we consider when including all of the variables?

- Is having that many predictors feasible (does having that many impact the quality of our measurement or the degree of missingness)?
- Multicolinearity (where two predictors have the same relationship with the outcome) can cause unstable regression estimates.
- Including predictors uncorrelated with the outcome leads to noisier estimates.
- Not including predictors related with the outcome leads to biased estimates.

Selecting vairables

- First use domain expertise.
- Then use practical determinations (some variables are exceptionally expensive to obtain)
- Lastly use computational power
- Exhaustive search
- Partial search

Choosing a model that fits well

- Need to capture how well the data fits the model on the training data
- Needs to penalise models with more parameters
- Much more advanced methods will also account for predictive power, but we focus on the training data for this section.
- One criterion is the adjusted R^2_{adj}

$$R^2_{adj} = 1 - \frac{n-1}{n-p-1}(1-R^2)$$

- n = number of data points
- p = number of variables
- $R^2 = 1 - \frac{RSS}{TSS}$

Which models should we select to compare?

- Exhaustive search: Compare every combination of variables
  - Often not practical
- Partial search: An algorithm helps to guide our search in models
- Forward selection:
- Backward selection:
- Stepwise regression

Forward selection

- Start with one predictor that produces the largest R^2
- Then add the next predictor that produces the largest change in R^2
- Continue to build the model until the change in R^2 not statistically significant (larger than what would be expected by chance)
- Main disadvantage: Misses pairs of predictors that perform well together but poorly seperately

Backward selection

- Start with all the predictors in a model
- Remove the least useful predictor (by statistical significance)
- Algorithm stops when all predictors are significant
- Main disadvantage: Computing the inital model with all predictors can be time-consuming and unstable

Stepwise regression

- Start as you would with forward selection
- At each step, also continue dropping variables that do not have a significant estimate.
- This can help to navigate the expense of straightforward backward selection

In R

Toyota Corolla data

Predict the price of used Toyota cars in a dealership from historic data.

```
toyota_sales <- read_csv(here::here("data/ToyotaCorolla.csv"))
head(toyota_sales)
```

```
## # A tibble: 6 x 39
##      Id Model Price Age_08_04 Mfg_Month Mfg_Year    KM Fuel_Type    HP Met_Co
##   <dbl> <chr> <dbl>     <dbl>     <dbl>    <dbl> <dbl> <chr>     <dbl>     <d
## 1     1 TOYO~ 13500        23        10     2002 46986 Diesel       90
## 2     2 TOYO~ 13750        23        10     2002 72937 Diesel       90
## 3     3 TOYO~ 13950        24         9     2002 41711 Diesel       90
## 4     4 TOYO~ 14950        26         7     2002 48000 Diesel       90
## 5     5 TOYO~ 13750        30         3     2002 38500 Diesel       90
## 6     6 TOYO~ 12950        32         1     2002 61000 Diesel       90
## # ... with 29 more variables: Color <chr>, Automatic <dbl>, CC <dbl>,
## #   Doors <dbl>, Cylinders <dbl>, Gears <dbl>, Quarterly_Tax <dbl>,
## #   Weight <dbl>, Mfr_Guarantee <dbl>, BOVAG_Guarantee <dbl>,
## #   Guarantee_Period <dbl>, ABS <dbl>, Airbag_1 <dbl>, Airbag_2 <dbl>,
```

Partition into test and training

- Decide to randomly split the data into training (70%) and test (30%).

```
toyota_sales <- toyota_sales%>%
  mutate(test_or_train =
           sample(c("test","train"),n(),replace = TRUE, prob = c(.3,.7)))
```


- Split the data into two datasets


```
toyota_train <- toyota_sales %>%
  filter(test_or_train == "train")
```



```
toyota_test <- toyota_sales %>%
  filter(test_or_train == "test")
```

Run a linear regression

- Use the training data to predict the outcome (price) by the age, fuel type, kilometers and whether it is an automatic or not.

```
model1 <- lm(Price ~ Age_08_04 + Fuel_Type+ KM + Automatic, data = toyota_train)
summary(model1)
```



```
##
## Call:
## lm(formula = Price ~ Age_08_04 + Fuel_Type + KM + Automatic,
##     data = toyota_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5660.9  -924.5   -47.8   831.2 11225.4
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
```

Make predictions

- Now we can use this model to predict. We will look at within sample (training data) and out of sample (test data predictions). First training data.

```
train_predict <-predict(model1, newdata = toyota_train,interval = "predict")

head(train_predict)
```



```
##        fit      lwr      upr
## 1 16405.89 13260.26 19551.51
## 2 15956.50 12812.96 19100.04
## 3 15500.61 12353.88 18647.34
## 4 14810.35 11667.33 17953.37
## 5 14979.89 11837.12 18122.66
## 6 15831.88 12700.51 18963.26
```

Assess predictions

- You'll cover the other diagnostics in your tute, but let's use mean error

```
toyota_train %>%
  mutate(predict_price = train_predict[,1])%>%
  summarise(mean_error = mean(Price -predict_price))
```



```
## # A tibble: 1 x 1
##   mean_error
##        <dbl>
## 1   2.90e-11
```

Assess predictions

- How often do the prediction intervals contain the true value?

```
toyota_train %>%
  mutate(predict_price = train_predict[,1],
         predict_low = train_predict[,2],
         predict_up = train_predict[,3],
         in_interval = Price>predict_low & Price<predict_up)%>%
  summarise(coverage = mean(in_interval))
```

```
## # A tibble: 1 x 1
##   coverage
##      <dbl>
## 1    0.958
```

What about the test data?

- Make predictions

```
test_predict <-predict(model1, newdata = toyota_test,interval = "predict")

head(test_predict)
```



```
##        fit      lwr      upr
## 1 16346.91 13200.60 19493.23
## 2 15937.37 12792.20 19082.54
## 3 14853.16 11711.14 17995.18
## 4 14830.40 11700.51 17960.28
## 5 16247.92 13115.88 19379.96
## 6 16181.31 13049.33 19313.29
```

Understand predictions

- How often do the prediction intervals contain the true value?
- What is the mean error?

```
toyota_test %>%
  mutate(predict_price = test_predict[,1],
         predict_low = test_predict[,2],
         predict_up = test_predict[,3],
         in_interval = Price>predict_low & Price<predict_up)%>%
  summarise(coverage = mean(in_interval),
            mean_error = mean(Price -predict_price))
```

```
## # A tibble: 1 x 2
##   coverage mean_error
##      <dbl>      <dbl>
## 1    0.937       29.4
```

Summary

- Linear regression marks the start of our work on prediction
- It assumes a linear relationship between predictors and outcome
- Important methods of assessing our predicitions are RMSE, ME, MSE
- However, there are challenges to linear regression in the number of predictors that be handled.
- It also assumes relatively strict form of the relationship (namely linear), and while the predictions can be quite robust to this, they are not always.
- In week 12 we build on this with tree based methodologies.

MONASH University

Lecturer: *Lauren Kennedy*

Department of Econometrics and Business Statistics

lauren.kennedy1@monash.edu

Week 10