# ETC3250/5250 Tutorial 1 Instructions

## Introduction to tidymodels

**prepared by Professor Di Cook**

**Week 1**

## 🎯 Objective

This tutorial aims to be a refresher on R, and to introduce you to working with `tidymodels`, an organised way to fit models as a part of data analysis.

## 🔧 Preparation

Get the packages installed:

```
install.packages(c("tidyverse","tidymodels","broom","dotwhisker","patchwork"))
```

This tutorial also assumes that you have previously used R before. If you are new to R you need to upskill yourself by working through the materials at https://learnr.numbat.space (https://learnr.numbat.space).

## 📖 Reading

Read the explanation and arguments for using tidymodels at https://rviews.rstudio.com/2020/04/21/the-case-for-tidymodels/ (https://rviews.rstudio.com/2020/04/21/the-case-for-tidymodels/).

Read through Emil Hitveldt's Linear Regression with tidymodels (https://emilhvitfeldt.github.io/ISLR-tidymodels-labs/linear-regression.html).

## 👋 Getting started

If you are in a zoom tutorial, say hello in the chat. If in person, do say hello to your tutor and to your neighbours.

## ⚙️ Exercise

The `nrc` data contains information collected on Statistics graduate programs in the USA. There are several ranking variables, and indicators of the departments' describing research, student and diversity, summarising many individual variables such as number of publications, student entrance scores and demographics. You can learn more about this data here (https://en.wikipedia.org/wiki/United_States_National_Research_Council_rankings).

The goal here is to follow the tidy models approach to fit a model for rank against indicators of research.

### 1

Load the libraries to complete the exercises.

```
# Load libraries
library(tidyverse)
library(tidymodels)
library(broom)
library(dotwhisker)
library(patchwork)
```

# 2

Read the data, simplify the names and select the relevant variables. You will want `rank = R.Rankings.5th.Percentile`, `research = Research.Activity.5th.Percentile`, `student = Student.Support.Outcomes.5th.Percentile` and `diversity = Diversity.5th.Percentile`.

```
# Read the data
nrc <- read_csv("https://iml.numbat.space/data/nrc.csv")

# Simplify names of and select variables to use
nrc <- nrc %>%
  mutate(rank = R.Rankings.5th.Percentile,
         research = Research.Activity.5th.Percentile,
         student = Student.Support.Outcomes.5th.Percentile,
         diversity = Diversity.5th.Percentile) %>%
  select(rank, research, student, diversity)
```

# 3

Make a plot of the observed response against predictors. What do you learn about the relationship between these variables?

```
# Make some plots of data
a1 <- ggplot(nrc, aes(x=research, y=rank)) + geom_point() +
  geom_smooth(se=FALSE)
a2 <- ggplot(nrc, aes(x=student, y=rank)) + geom_point() +
  geom_smooth(se=FALSE)
a3 <- ggplot(nrc, aes(x=diversity, y=rank)) + geom_point() +
  geom_smooth(se=FALSE)
a1 + a2 + a3
```

# 4

Set up the model. While it is unnecessary to set the mode for a linear regression since it can only be regression, we continue to do it in these labs to be explicit. The specification doesn't perform any calculations by itself. It is just a specification of what we want to do.

```
# Set up and fit model
lm_mod <-
  linear_reg() %>%
  set_engine("lm")
lm_mod
```

# 5

Fit the model. Once we have the specification we can fit it by supplying a formula expression and the data we want to fit the model on. The formula is written on the form `y ~ x` where `y` is the name of the response and `x` is the name of the predictors. The names used in the formula should match the names of the variables in the data set passed to data.

```
lm_fit <-
  lm_mod %>%
  fit(rank ~ research + student + diversity,
      data = nrc)
lm_fit
```

The result of this fit is a parsnip (https://parsnip.tidymodels.org) model object. This object contains the underlying fit as well as some parsnip-specific information. If we want to look at the underlying fit object we can access it and summarise it with

```
lm_fit %>%
  pluck("fit") %>%
  summary()
```

# 6

Report the coefficients of the model fit. We can use packages from the broom package to extract key information out of the model objects in tidy formats. The `tidy()` function returns the parameter estimates of a `lm` object. Explain the relationship between the predictors and the response variable. Is the interpretation of `research`, "the higher the value of research indicates higher value of rank"? This doesn't make sense, why?

```
tidy(lm_fit)
```

# 7

Make a dot and whisker plot of the coefficients, to visualise the significance of the different variables. Explain what you learn about the importance of the three explanatory variables for predicting the response.

```
tidy(lm_fit) %>%
  dwplot(dot_args = list(size = 2, color = "black"),
         whisker_args = list(color = "black"),
         vline = geom_vline(xintercept = 0, colour = "grey50", linetype = 2))
```

# 8

Report the fit statistics, using `broom::glance()`. What do you learn about the strength of the fit?

```
glance(lm_fit)
```

# 9

Explore the model fit visually. Plot the predicted values against observed, residuals against fitted, and predicted against each of the predictors. Summarise what you learn about the model fit.

```
# Plot the fit
nrc_all <- augment(lm_fit, nrc)
p1 <- ggplot(nrc_all, aes(x=.pred, y=rank)) + geom_point()

p2 <- ggplot(nrc_all, aes(x=.pred, y=.resid)) + geom_point()

p1 + p2
```

```
p3 <- ggplot(nrc_all, aes(x=research, y=.pred)) + geom_point()
p4 <- ggplot(nrc_all, aes(x=student, y=.pred)) + geom_point()
p5 <- ggplot(nrc_all, aes(x=diversity, y=.pred)) + geom_point()

p3 + p4 + p5
```

## 10

Generate a grid of new data values to predict, with all combinations of `research = c(10, 40, 70)`, `student = c(10, 40, 70)`, `diversity = c(10, 40, 70)`. Predict these values, as point and confidence intervals.

```
# Predict new data
new_points <- expand.grid(research = c(10, 40, 70),
                          student = c(10, 40, 70),
                          diversity = c(10, 40, 70))
new_points

mean_pred <- predict(lm_fit, new_data = new_points)
mean_pred

conf_int_pred <- predict(lm_fit,
                         new_data = new_points,
                         type = "conf_int")
conf_int_pred

new_points <- augment(lm_fit, new_points)
```

## 11

Make a plot of predicted values vs research for the observed data and the new data, with new data coloured differently. How do the predicted values compare?

# 🛑 Wrapping up

Talk to your tutor about what you think you learned today, what was easy, what was fun, what you found hard.

# 🦉 Got a problem you can't solve?

It is always good to try to suspect something simple is going wrong first. Most likely the error is a simple one, like a missing ")" or ",".

For deeper answers to questions about packages, analyses and functions, compiling an Rmarkdown document, or simply the error that is being generated, you can usually find an answer using google.

If you are still having problems, and you need someone to help you, the first step is to explain the problem clearly. Make a reproducible example of your problem, following the guidelines here (https://learnr.numbat.space/chapter3#3). Bring this to a consultation, or post on the discussion forum on moodle.

Externally, Q/A site: http://stackoverflow.com (http://stackoverflow.com) is a great place to get answers to tougher questions about R and also data analysis. You always need to check that someone hasn't asked it before, the answer might already be available for you. Remember these people that kindly answer questions on stackoverflow have day jobs too, and do this community support as a kindness to all of us.