# Statistical Thinking Assignment 2

Xinyi Cui, Janice Hsin Hsu, Pranali Angne, Sahinya Akila

10/17/2021

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
```

```
# Loading Libraries

library(printr)
library(tidyverse)
library(tidymodels)
library(broom)
library(splines)
library(dagitty)
library(ggdag)
library(knitr)
library(gtsummary)
library(kableExtra)
```

# Task 1: Estimating Neonatal Mortality

## Introduction

One of this century's global goals has been the reduction of childhood mortality across all countries. There has been enormous effort put into this goal at all levels from the united nations down to local interventions. The aim of this report is to produce a linear regression model to estimate the average neonatal mortality rate (NMR).

## Data

The source of the child mortality data is from the UN Inter-agency Group for Child Mortality Estimation.

It contains the following columns:

- `country_name`: Name of the country
- `year`: The year the data was measured
- `region`: The name of the continent the country is from
- `nmr`: The observed number of neonatal deaths per thousand live births (the neonatal mortality rate). This is measured either using a country's vital registration system (births and deaths register) or using some sort of high-quality survey.
- `u5mr`: The estimated under-five mortality rate
- `nmr_transformed`: log of the number of neonatal deaths per 1000 live births divided by the number of non-neonantal deaths per 1000 live births.

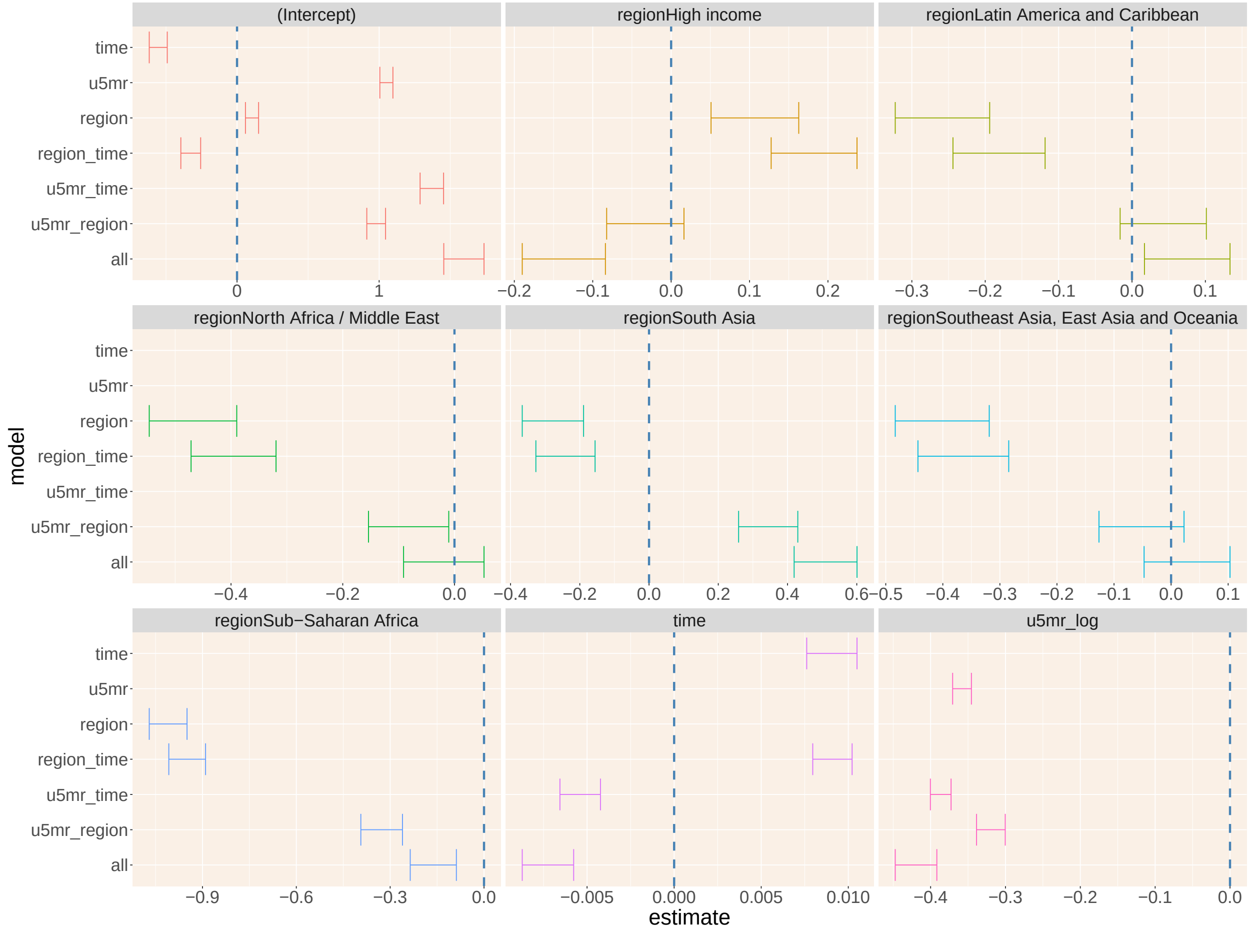
$$log(\frac{nmr}{u5mr - nmr})$$

```
# Reading the data
neonatal_mortality <- read_csv("neonatal_mortality.csv")

# Adding log ratio between neonatal mortality and non-neonatal mortality rate and time
nmr_data <- neonatal_mortality %>%
  mutate(u5mr_log = log(u5mr),
         transformed_nmr = log(nmr/(u5mr-nmr)),
         time = year - min(year))
```

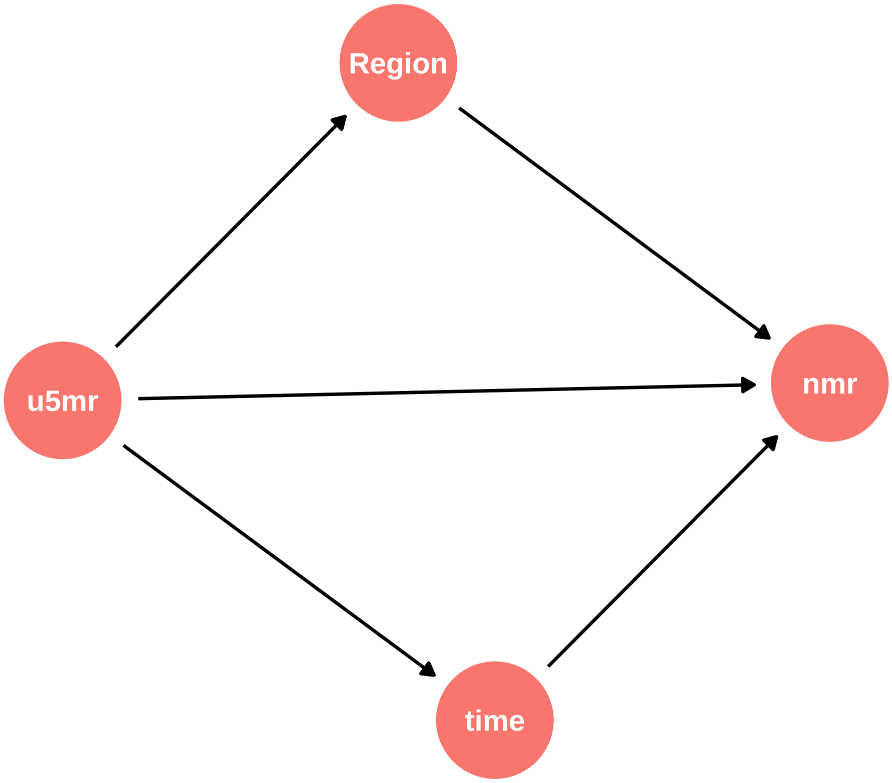
## Task 1.1: Linear Regression

**Task 1.1.1: Explain the choice of variables in your model (you should not use country_name, if you use U5MR, you should use it on the log-scale!). In particular you should consider whether an interaction effect should be used.**

```r
# Splitting the data
nmr_split <- initial_split(nmr_data, strata = region)
nmr_train <- training(nmr_split)
nmr_test <- testing(nmr_split)

# Choosing the variables by trying different combinations of variables
fit1 <- lm(transformed_nmr ~ u5mr_log + region + time, nmr_train) %>%
  tidy() %>%
  mutate(model = "all")

fit2 <- lm(transformed_nmr ~ u5mr_log + region, nmr_train) %>%
  tidy() %>%
  mutate(model = "u5mr_region")

fit3 <- lm(transformed_nmr ~ u5mr_log + time, nmr_train) %>%
  tidy() %>%
  mutate(model = "u5mr_time")

fit4 <- lm(transformed_nmr ~ region + time, nmr_train) %>%
  tidy() %>%
  mutate(model = "region_time")

fit5 <- lm(transformed_nmr ~ region, nmr_train) %>%
  tidy() %>%
  mutate(model = "region")

fit6 <- lm(transformed_nmr ~ u5mr_log, nmr_train) %>%
  tidy() %>%
  mutate(model = "u5mr")

fit7 <- lm(transformed_nmr ~ time, nmr_train) %>%
  tidy() %>%
  mutate(model = "time")

# joining all the data frames and calculating the upper and lower values
full_model <- list(fit1, fit2, fit3, fit4, fit5, fit6, fit7) %>%
  reduce(full_join) %>%
  mutate(upper = estimate + 1.96 * std.error,
         lower = estimate - 1.96 * std.error)

full_model %>% ggplot(aes(estimate, y = model)) +
  geom_errorbar(aes(xmin = lower, xmax = upper, color = term)) +
  geom_vline(aes(xintercept = 0),
             linetype = "dashed",
             size = 1.2,
             color = "steel blue") +
  facet_wrap(.~term, scale = "free_x") +
  scale_y_discrete(limits = c("all", "u5mr_region", "u5mr_time", "region_time", "region", "u5mr", "time
  theme(panel.background = element_rect(fill = "linen"), legend.position = "none", text = element_text(
```



```
# understanding relationship between the variables
dagify(nmr ~ u5mr,
       nmr ~ time,
       nmr ~ Region,
       time ~ u5mr,
       Region ~ u5mr
       ) %>%
  ggdag_adjust(node_size = 20) +
  theme_dag(legend.position = "none")
```



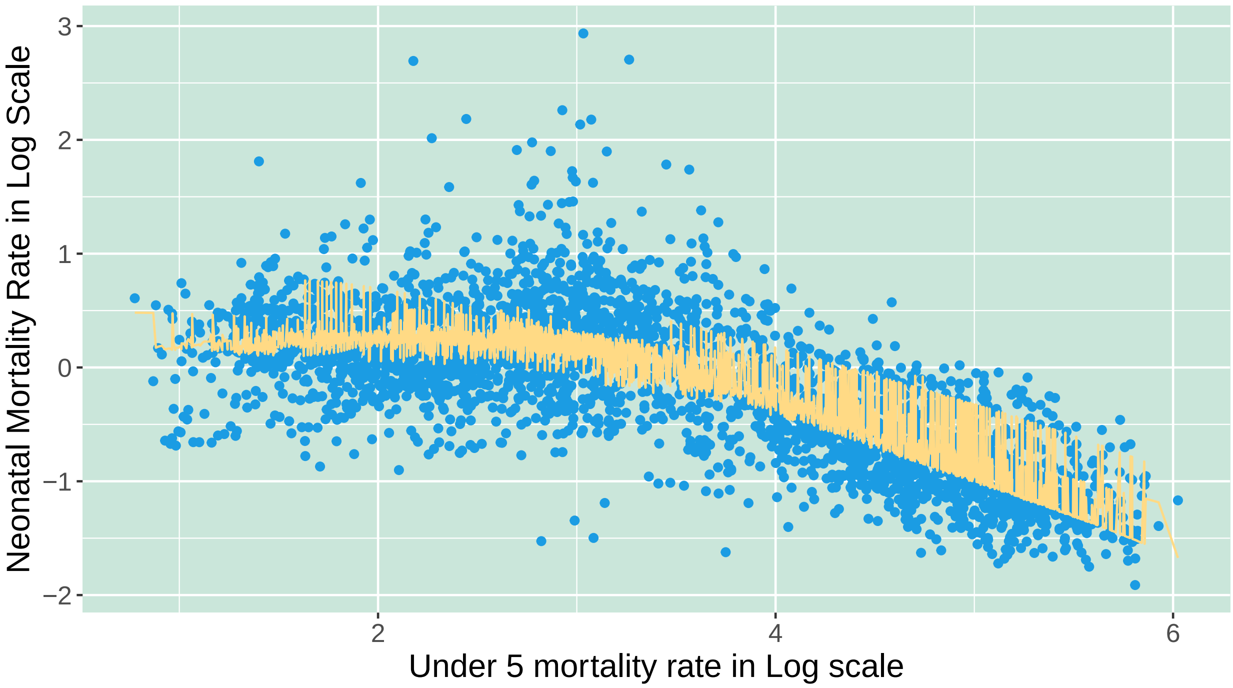
```r
# Fitting the model using the training data (with the interactive effect)
fit <- lm(transformed_nmr ~ u5mr_log + u5mr_log*region + u5mr_log*time, nmr_train)

summary(fit)
```

```
##
## Call:
## lm(formula = transformed_nmr ~ u5mr_log + u5mr_log * region +
##     u5mr_log * time, data = nmr_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74613 -0.22962 -0.01499  0.21162  2.80810
##
## Coefficients:
##                                                 Estimate Std. Error
## (Intercept)                                    1.5075489  0.1278860
## u5mr_log                                      -0.3939249  0.0353013
## regionHigh income                             -0.4280850  0.0761660
## regionLatin America and Caribbean              0.9244718  0.1060436
## regionNorth Africa / Middle East               0.3200096  0.1205871
## regionSouth Asia                               1.5084939  0.2944948
## regionSoutheast Asia, East Asia and Oceania           0.6913038  0.1373989
## regionSub-Saharan Africa                              1.3205693  0.1421459
## time                                                 -0.0124214  0.0018484
## u5mr_log:regionHigh income                            0.1537018  0.0288059
## u5mr_log:regionLatin America and Caribbean           -0.2560919  0.0320715
## u5mr_log:regionNorth Africa / Middle East            -0.1152158  0.0341583
## u5mr_log:regionSouth Asia                            -0.2622939  0.0651662
## u5mr_log:regionSoutheast Asia, East Asia and Oceania -0.2042708  0.0383515
## u5mr_log:regionSub-Saharan Africa                    -0.3544257  0.0348577
## u5mr_log:time                                         0.0021563  0.0004586
##                                                      t value Pr(>|t|)
## (Intercept)                                           11.788  < 2e-16 ***
## u5mr_log                                             -11.159  < 2e-16 ***
## regionHigh income                                     -5.620 2.07e-08 ***
## regionLatin America and Caribbean                      8.718  < 2e-16 ***
## regionNorth Africa / Middle East                       2.654 0.007999 **
## regionSouth Asia                                       5.122 3.19e-07 ***
## regionSoutheast Asia, East Asia and Oceania            5.031 5.13e-07 ***
## regionSub-Saharan Africa                               9.290  < 2e-16 ***
## time                                                  -6.720 2.14e-11 ***
## u5mr_log:regionHigh income                             5.336 1.02e-07 ***
## u5mr_log:regionLatin America and Caribbean            -7.985 1.93e-15 ***
## u5mr_log:regionNorth Africa / Middle East             -3.373 0.000752 ***
## u5mr_log:regionSouth Asia                             -4.025 5.83e-05 ***
## u5mr_log:regionSoutheast Asia, East Asia and Oceania  -5.326 1.07e-07 ***
## u5mr_log:regionSub-Saharan Africa                    -10.168  < 2e-16 ***
## u5mr_log:time                                          4.702 2.68e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4036 on 3261 degrees of freedom
## Multiple R-squared:  0.621,  Adjusted R-squared:  0.6193
## F-statistic: 356.2 on 15 and 3261 DF,  p-value: < 2.2e-16
```

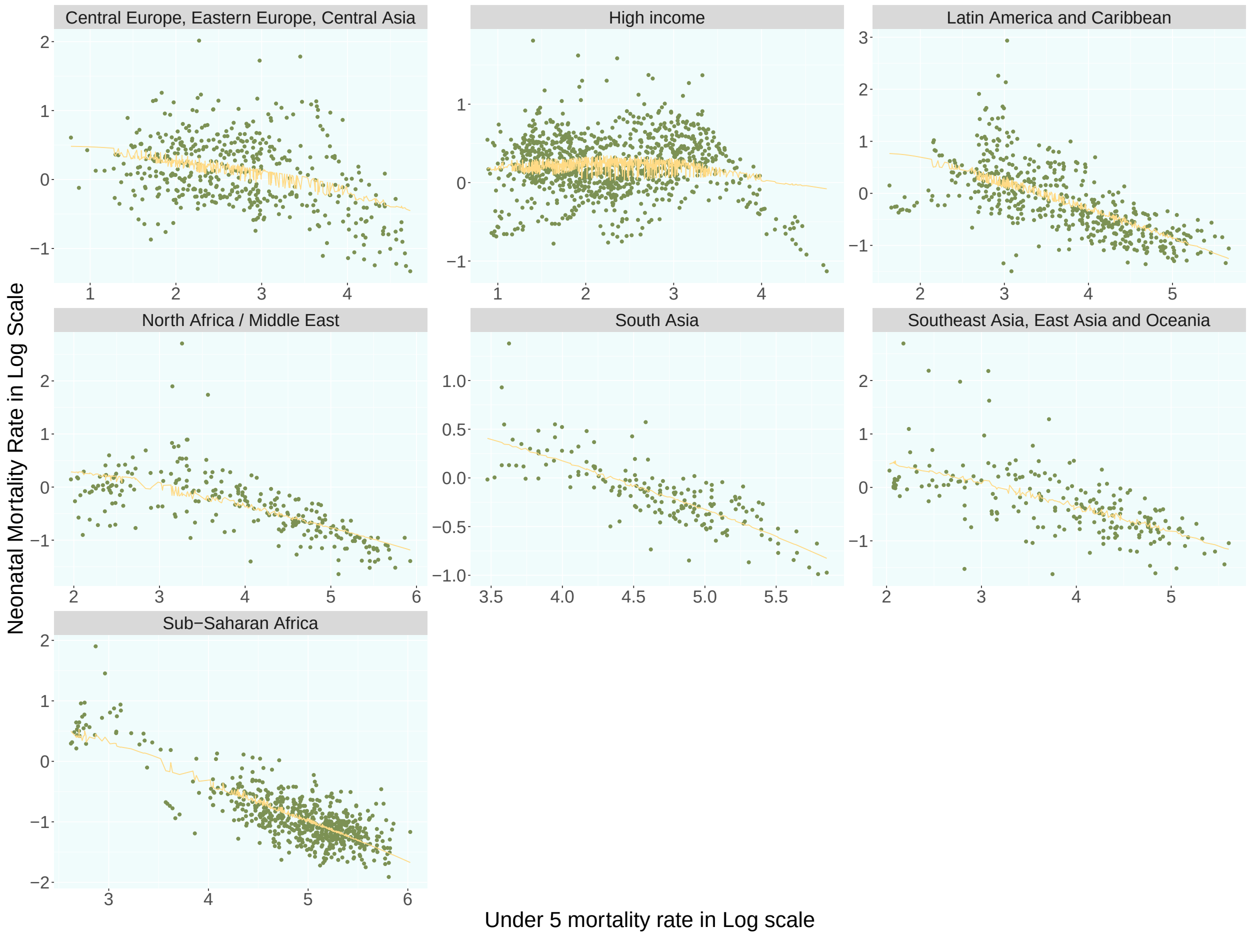
u5mr_log*region significant

**Task 1.1.2: Assess your linear model and comment on its fit. This should be done a) for all data simultaneously; b)for data in each region; and c) for data in a maximum of 3 countries that should be chosen to highlight different aspects of the fit diagnostics.**

```r
augment_fit <- fit %>%
  augment(data = nmr_train)
# part a
augment_fit %>%
  ggplot() +
  geom_point(aes(x = u5mr_log, y= transformed_nmr), color = "#1b9ce3")+
  geom_line(aes(x = u5mr_log, y = .fitted), color = "#ffda85") +
  theme(panel.background = element_rect(fill = "#cae6da"), text = element_text(size=15)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal Mortality Rate in Log Scale")
```

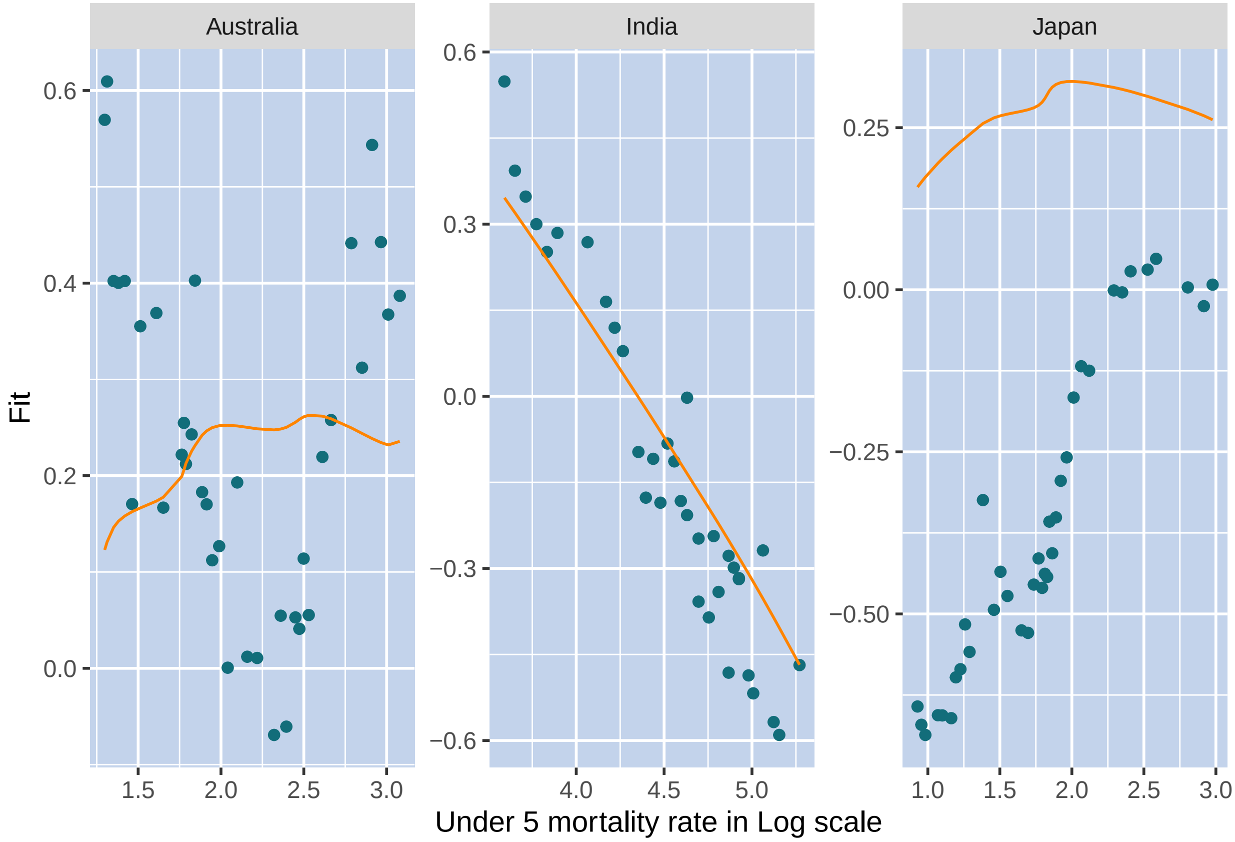


```r
# part b
augment_fit %>%
  ggplot() +
  geom_point(aes(x = u5mr_log, y= transformed_nmr), color = "#7c9154")+
  geom_line(aes(x = u5mr_log, y = .fitted), color = "#ffda85")+
  facet_wrap(~region, scale = "free") +
  theme(panel.background = element_rect(fill = "#f0fcfc"), text = element_text(size=25)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal Mortality Rate in Log Scale")
```



```
# part c

top3 <- nmr_data %>% filter(region %in% c("Central Europe, Eastern Europe, Central Asia", "High income"
  group_by(country_name) %>%
  summarise(count_europe_asia = n()) %>%
  arrange(desc(count_europe_asia)) %>% head(3)

augment_fit %>%
  filter(country_name %in% c("India", "Japan", "Australia")) %>%
  ggplot(aes(x = u5mr_log, y = transformed_nmr)) +
  geom_point(color = "#126d7a") +
  geom_line(aes(x = u5mr_log, y = .fitted), color = "#ff8400") +
  facet_wrap(~country_name, scales = "free") +
  theme(panel.background = element_rect(fill = "#c3d3eb")) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Fit")
```

| .metric | .estimator | .estimate |
|---|---|---|
| rmse | standard | 0.4324817 |
| rsq | standard | 0.5751634 |
| mae | standard | 0.2964029 |



**Task 1.1.3: Estimate the root mean square error and the mean absolute error on a test set. The test set should be produced using the argument strata = region.**

```
# Predict the test dataset
lm_pred <- tibble(pred = predict(fit, nmr_test))
lm_pred <- bind_cols(nmr_test, lm_pred)
lm_pred %>%
  metrics(truth = transformed_nmr,
          estimate = pred) %>%
  kbl(booktabs = T) %>%
  kable_styling(position = "center")
```