Statistical Thinking Assignment 2

Xinyi Cui (29645530), Janice Hsin Hsu (32195109), Pranali Angne (32355068), Sahinya Akila (29201128)

10/17/2021

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
options(digits = 2)
```

```
# Loading Libraries

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

options(scipen = 100000)
```

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 (https://childmortality.org/data).

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.

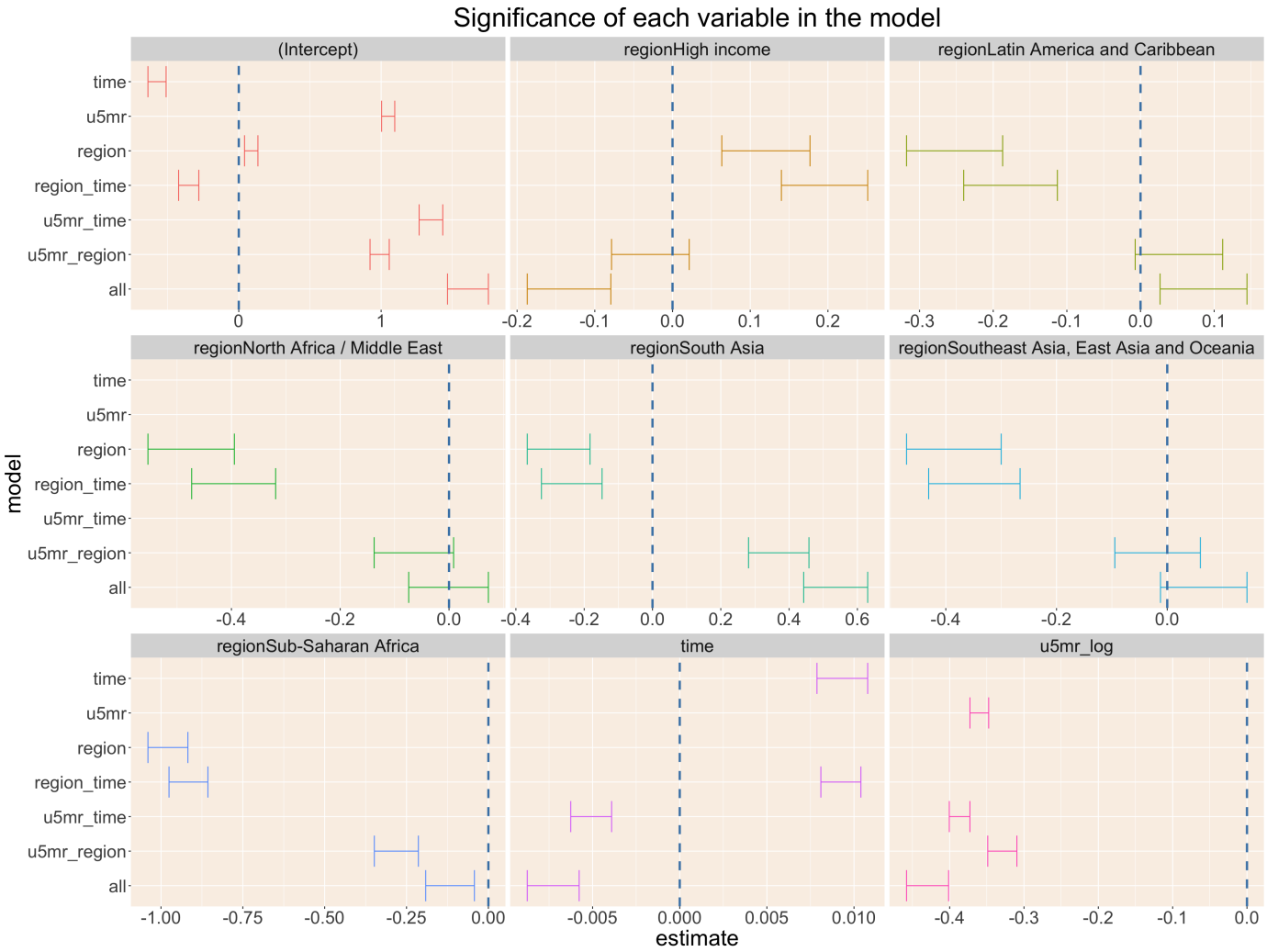
$$log(\frac{nmr}{u5mr - nmr})$$

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

# Adding log ratio between neonatal mortality and non-neonatal mortality rate and tim
e
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", "regi
on", "u5mr", "time"))+
  theme(panel.background = element_rect(fill = "linen"), legend.position = "none", te
xt = element_text(size=25), plot.title = element_text(hjust = 0.5)) +
  ggtitle("Significance of each variable in the model")
```



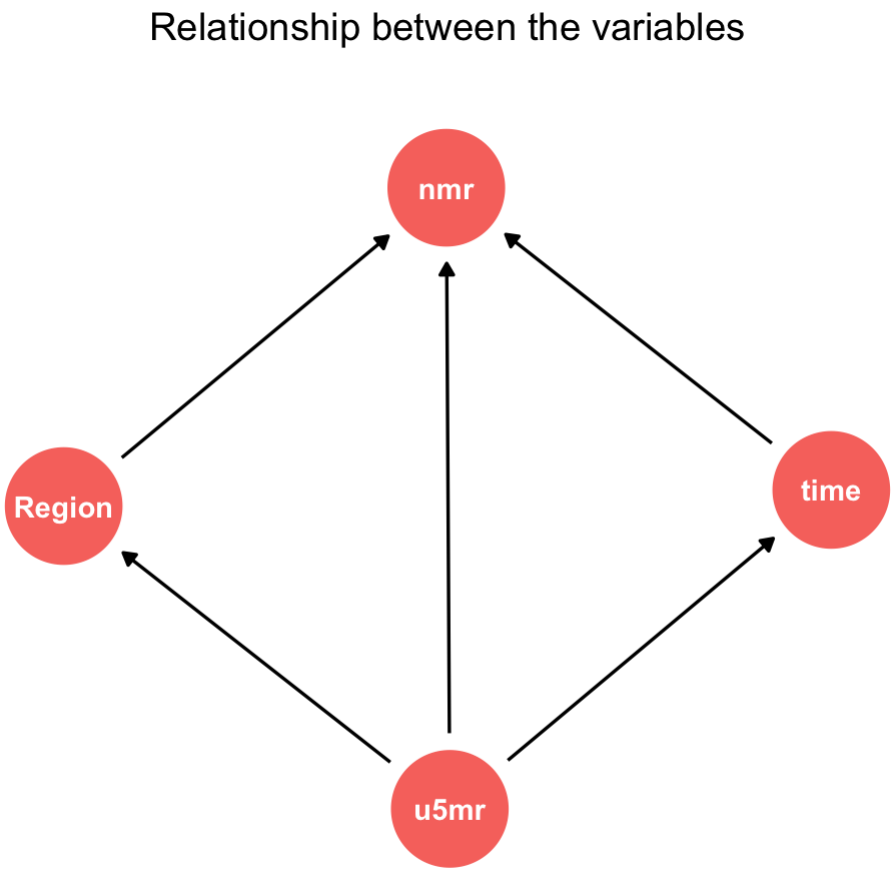
In order to choose appropriate variables for the linear model, we have created seven models which includes different variables. Then we fitted seven different models into the error bar graph that represents the significance of each variable in the model.

From the error bar we can see that the time variable is considered to be significant among all models under the 95% confidence interval, as both the maximum and the minimum are different from 0. Similarly, variables such as region, and u5mr can also be considered as significant. However, while looking at the region_time model which represents the interaction effect between region and time, we can see that the shifts from the region model are only significant for High income and Latin American region; and therefore, the interaction effect should not be used here.

For u5mr_region model, if we only consider u5mr or region, individually they are significant. When region is conditional on u5mr, the variable region turns from significant to insignificant, as it moves towards 0. Therefore, u5mr will affect region, but region have no effect on u5mr. We consider this is significant.

For u5mr_time model, as we mentioned before, u5mr itself is significant, while adding time, the significance level have not changed. However, when it is conditional on u5mr, time have been affected from positive values to negative significance values shown in graph t. Therefore, time u5mr will affect time. We also consider this is significant.

```
# 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") +
  ggtitle("Relationship between the variables") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# 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)
```

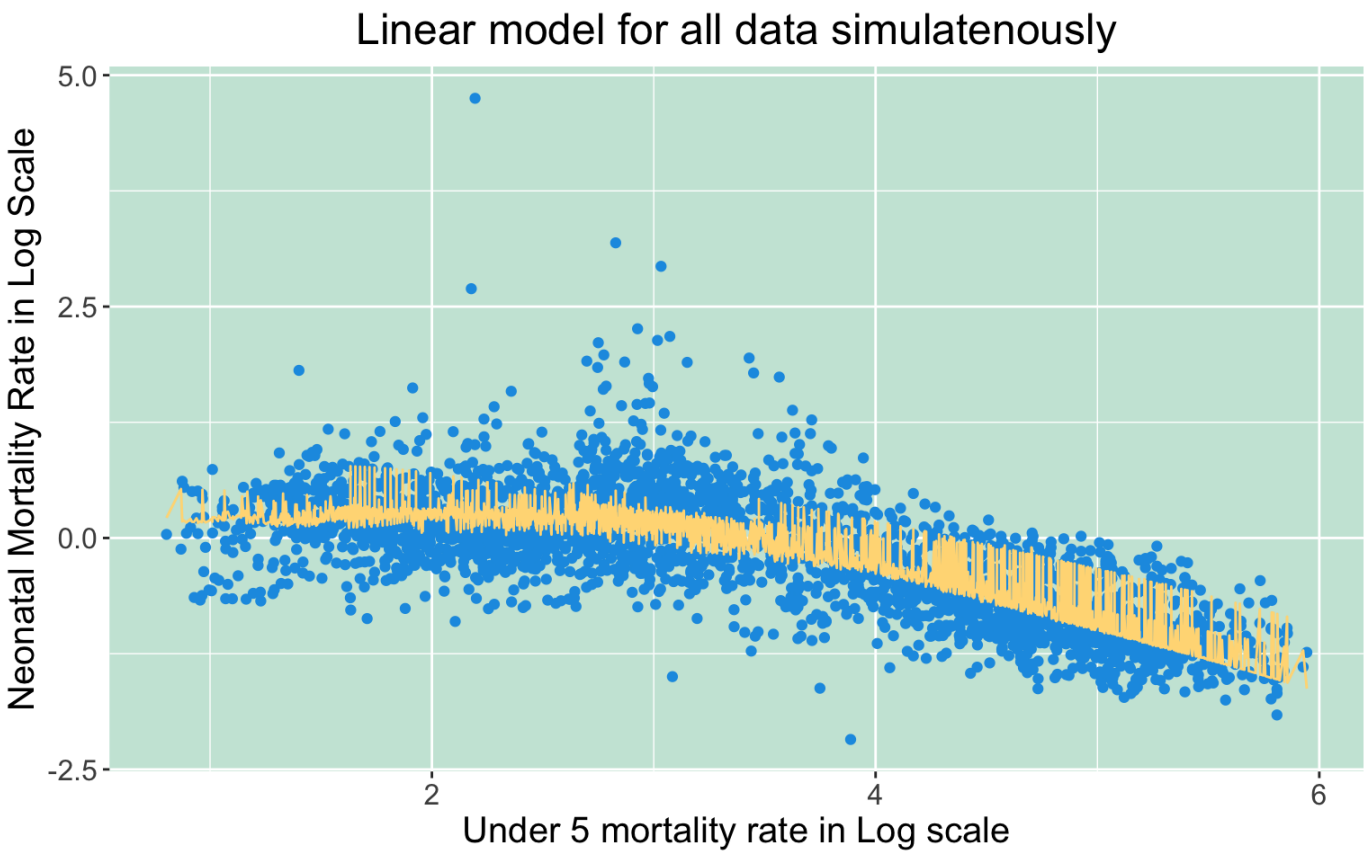
```
##
## Call:
## lm(formula = transformed_nmr ~ u5mr_log + u5mr_log * region +
##     u5mr_log * time, data = nmr_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.952 -0.233 -0.015  0.208  4.547
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                      1.561715   0.130787
## u5mr_log                                        -0.423426   0.036047
## regionHigh income                               -0.482298   0.077233
## regionLatin America and Caribbean                0.873946   0.107055
## regionNorth Africa / Middle East                 0.279148   0.127331
## regionSouth Asia                                 1.498821   0.313211
## regionSoutheast Asia, East Asia and Oceania      0.572819   0.148189
## regionSub-Saharan Africa                         1.340745   0.146079
## time                                            -0.012047   0.001893
## u5mr_log:regionHigh income                       0.176858   0.029228
## u5mr_log:regionLatin America and Caribbean      -0.233952   0.032418
## u5mr_log:regionNorth Africa / Middle East       -0.093161   0.035581
## u5mr_log:regionSouth Asia                       -0.244563   0.068875
## u5mr_log:regionSoutheast Asia, East Asia and Oceania -0.156841   0.040779
## u5mr_log:regionSub-Saharan Africa               -0.339928   0.035763
## u5mr_log:time                                    0.002099   0.000469
##                                                 t value
## (Intercept)                                       11.94
## u5mr_log                                         -11.75
## regionHigh income                                 -6.24
## regionLatin America and Caribbean                  8.16
## regionNorth Africa / Middle East                   2.19
## regionSouth Asia                                   4.79
## regionSoutheast Asia, East Asia and Oceania        3.87
## regionSub-Saharan Africa                           9.18
## time                                              -6.36
## u5mr_log:regionHigh income                         6.05
## u5mr_log:regionLatin America and Caribbean        -7.22
## u5mr_log:regionNorth Africa / Middle East         -2.62
## u5mr_log:regionSouth Asia                         -3.55
## u5mr_log:regionSoutheast Asia, East Asia and Oceania -3.85
## u5mr_log:regionSub-Saharan Africa                 -9.51
## u5mr_log:time                                      4.47
##                                                            Pr(>|t|)
## (Intercept)                                      < 0.0000000000000002 ***
## u5mr_log                                         < 0.0000000000000002 ***
## regionHigh income                                 0.00000000047952388 ***
## regionLatin America and Caribbean                 0.00000000000000046 ***
## regionNorth Africa / Middle East                                0.02843 *
## regionSouth Asia                                  0.00000178278818489 ***
## regionSoutheast Asia, East Asia and Oceania                     0.00011 ***
## regionSub-Saharan Africa                         < 0.0000000000000002 ***
## time                                              0.00000000022368145 ***
## u5mr_log:regionHigh income                        0.00000000160226743 ***
## u5mr_log:regionLatin America and Caribbean        0.00000000000065904 ***
## u5mr_log:regionNorth Africa / Middle East                       0.00888 **
```

```
## u5mr_log:regionSouth Asia                                          0.00039 ***
## u5mr_log:regionSoutheast Asia, East Asia and Oceania               0.00012 ***
## u5mr_log:regionSub-Saharan Africa                     < 0.0000000000000002 ***
## u5mr_log:time                                          0.00000793105352538 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.41 on 3261 degrees of freedom
## Multiple R-squared:  0.613,  Adjusted R-squared:  0.612
## F-statistic:  345 on 15 and 3261 DF,  p-value: <0.0000000000000002
```

The dag graph illustrates The information about the relationship we get from the error bar graph. The neonatal mortality rate (nmr) is the dependent variable, from the dag graph we can see that both time and region is piped. We do not want to add them, cause they will end up worth less effect which removes the correlation between u5mr and nmr because we are adding bias through time and region. u5mr is a fork therefore we consider to include it. Moreover, from the error bar we see the indirect affect to u5mr and time, therefore, to consider the interaction effect, we include u5mr*region and u5mr*time.

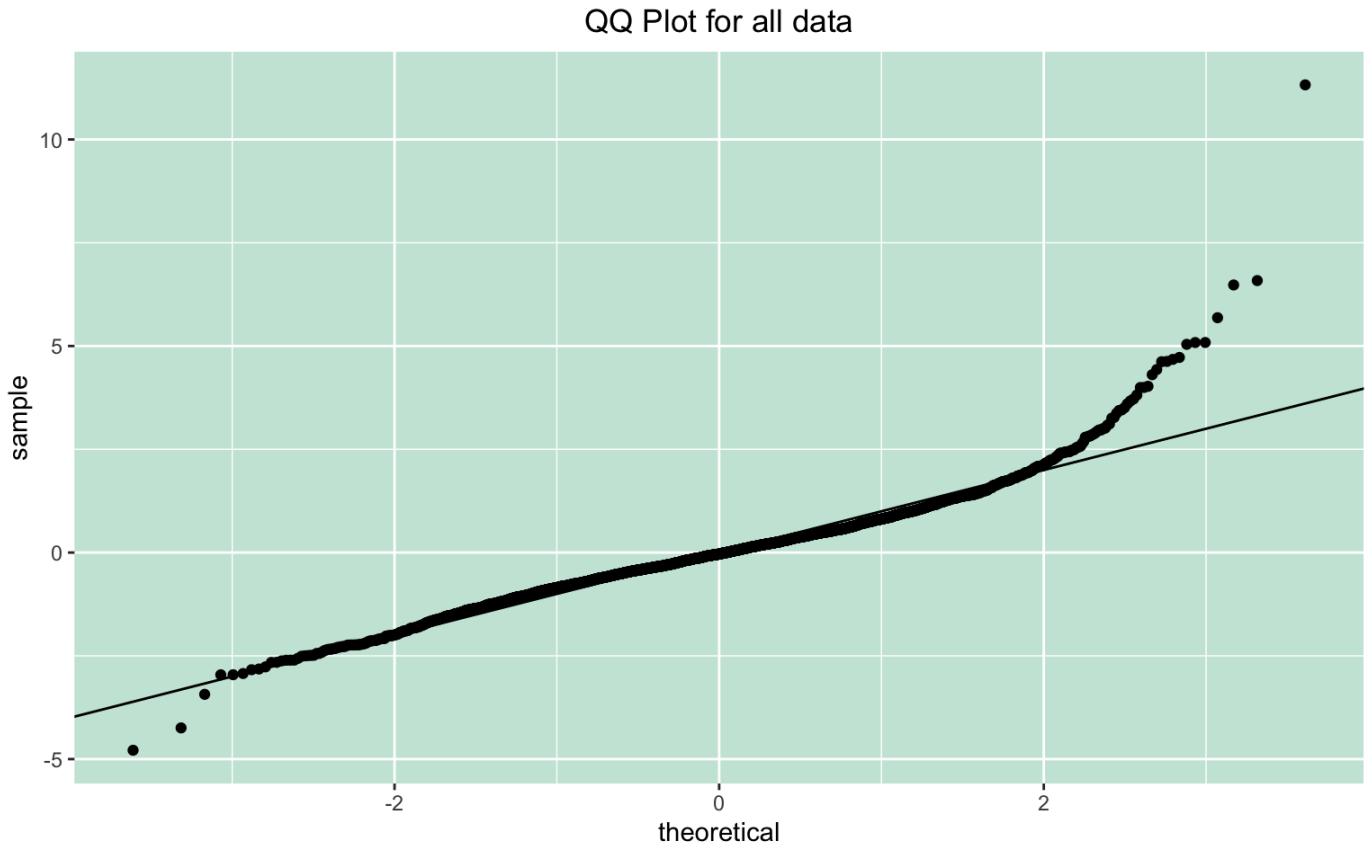
Task 1.1.2: Assess your linear model and comment on its fit. This should be done a) for all data simultaneously; b)for data in each region; and c) for data in a maximum of 3 countries that should be chosen to highlight different aspects of the fit diagnostics.

```
augment_fit <- fit %>%
  augment(data = nmr_train)
# Part a: For all data simultaneously
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=1
5),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal Mortality Rate in Log Scale") +
  ggtitle("Linear model for all data simulatenously")
```



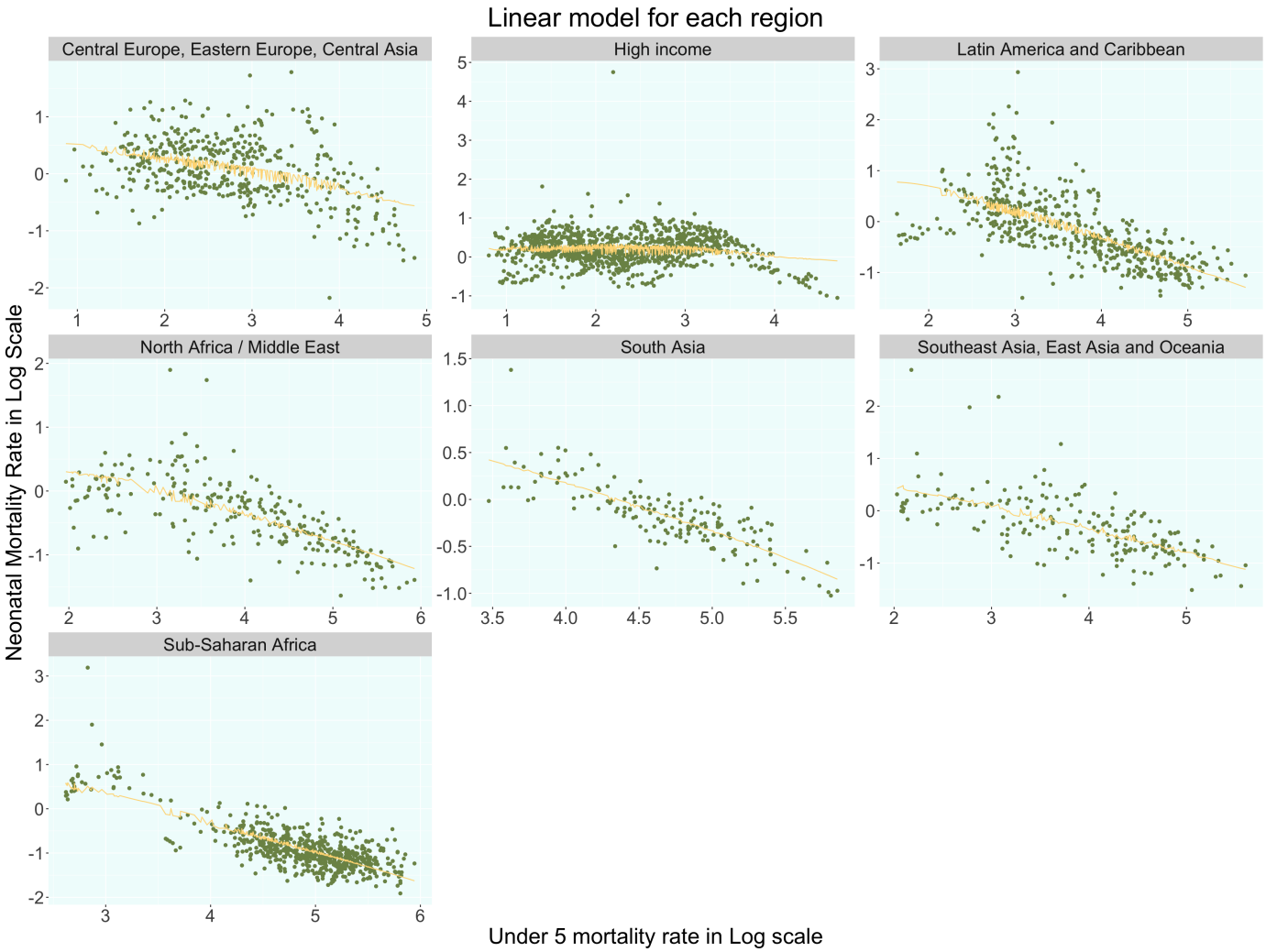
The fitted model fits fairly well with all the data simultaneously, as we can see from the graph that the fitted model and the actual data mostly overlap with each other.

```
augment_fit %>%
  ggplot(aes(sample=.resid/.sigma)) +
  geom_qq()+
  geom_abline(intercept=0, slope = 1) +
  theme(panel.background = element_rect(fill = "#cae6da"),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("QQ Plot for all data")
```



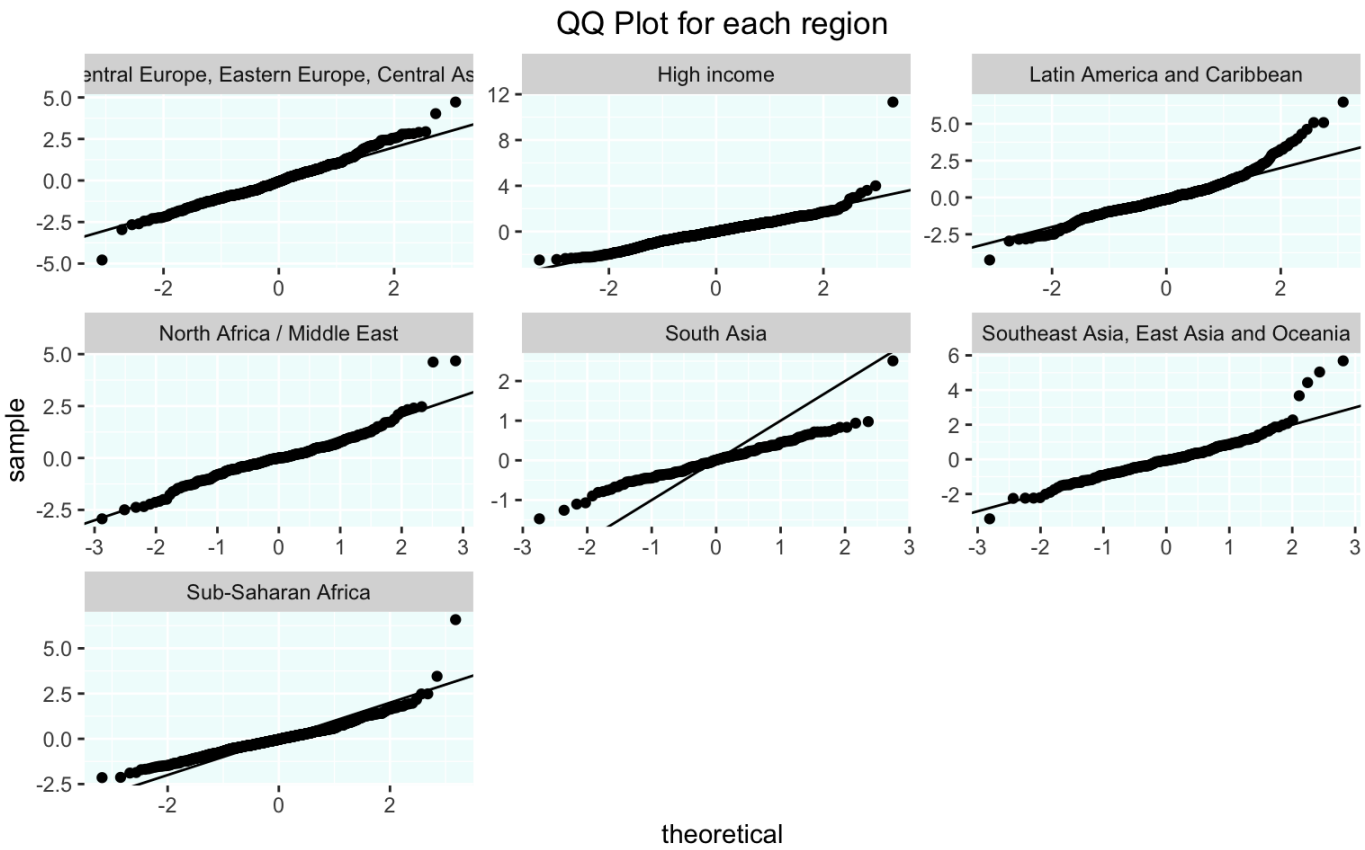
From the QQ-Plot, it can be observed that there is a heavy right tail which is common in log scaled variables. The normality does not equal 1.

```
# Part b: For data in each region
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=2
5),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal Mortality Rate in Log Scale") +
  ggtitle("Linear model for each region")
```



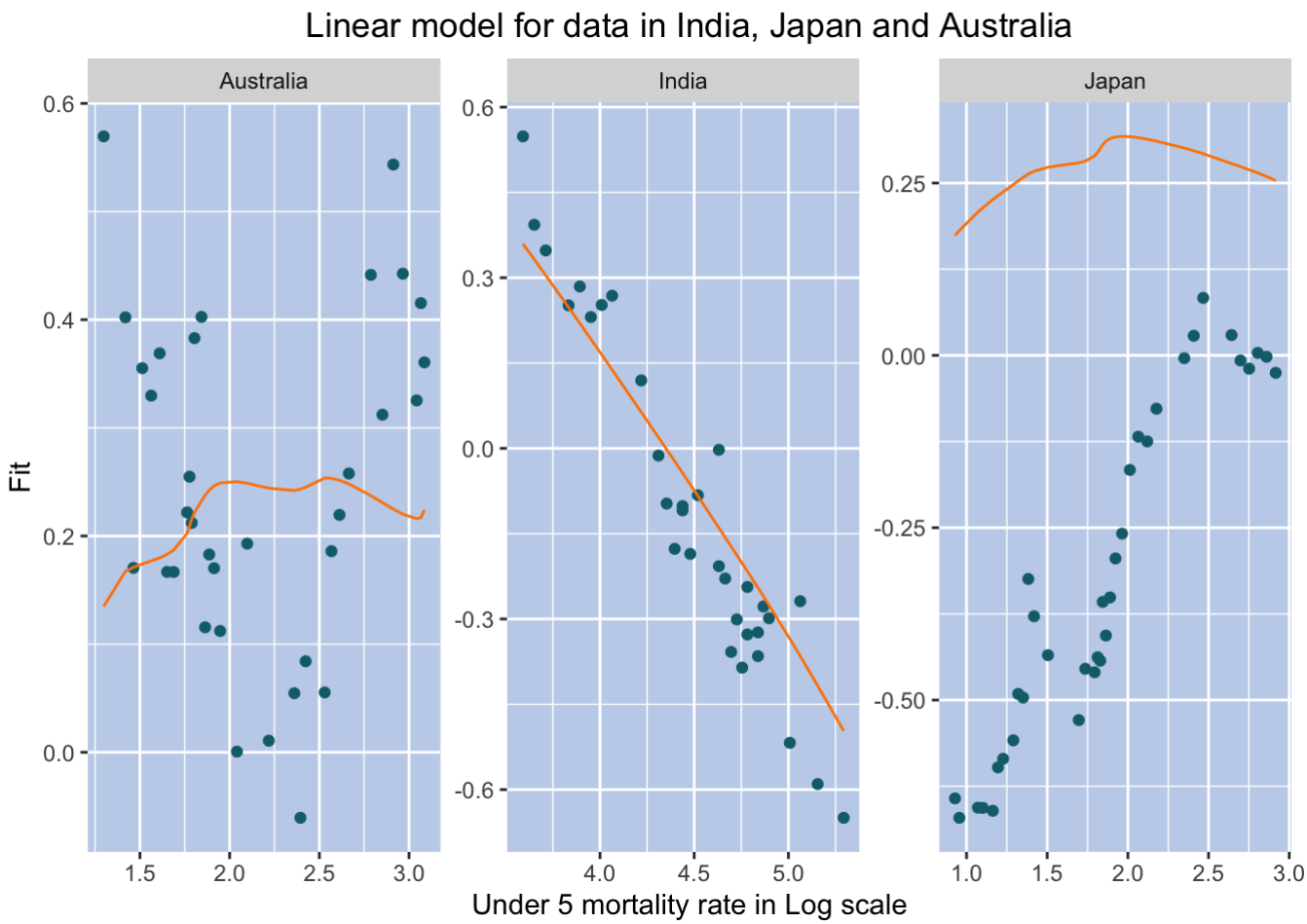
Similarly, the fitted model works well with data in each region as well, especially for high income countries and Latin America and Caribbean.

```
augment_fit %>%
  ggplot(aes(sample=.resid/.sigma)) +
  geom_qq()+
  geom_abline(intercept=0, slope = 1) +
  facet_wrap(~region, scale = "free") +
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("QQ Plot for each region")
```



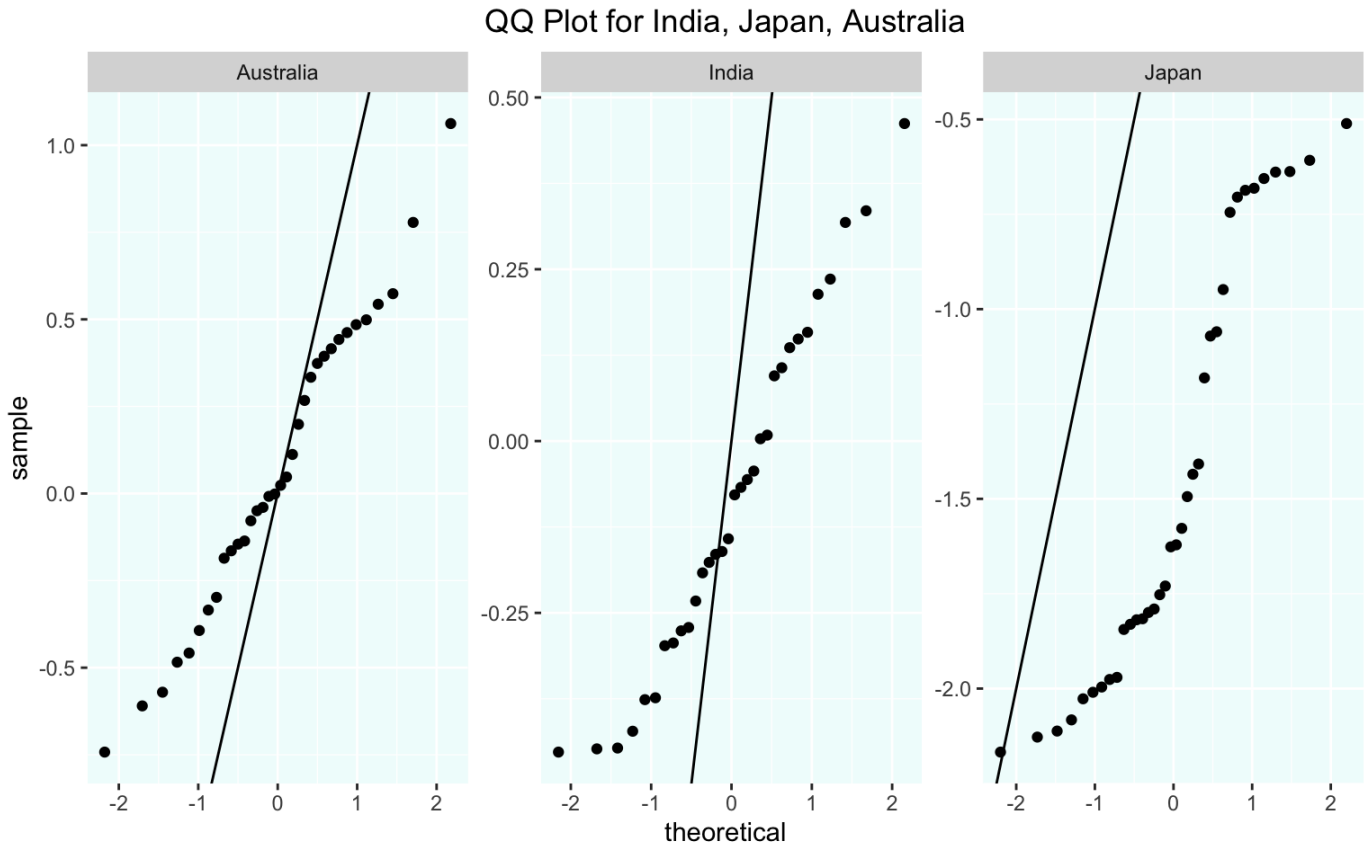
All regions except for South Asia, all other regions have a common distribution.

```
# Part c: For data in a maximum of 3 countries that should be chosen to highlight dif
ferent aspects of the fit diagnostics
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"),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Fit") +
  ggtitle("Linear model for data in India, Japan and Australia")
```



We choose India, Australia and Japan to highlight different aspects of the fit. Australia and Japan both are developed countries, and India is a developing country. For Australia and Japan, they fit terribly for the model, especially Japan, none of the points are on the model. On contrast, India fits better than the other two countries.

```
augment_fit %>%
  filter(country_name %in% c("India", "Japan", "Australia")) %>%
  ggplot(aes(sample=.resid/.sigma)) +
  geom_qq()+
  geom_abline(intercept=0, slope = 1) +
  facet_wrap(~country_name, scales = "free") +
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("QQ Plot for India, Japan, Australia")
```



The QQ-Plot shows that none of the countries show a common distribution.

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,
      caption = "Mean absolute error, Root mean Square Error, R-squared value for the
test set data") %>%
  kable_styling(position = "center")
```

Mean absolute error, Root mean Square Error, R-squared value for the test set data

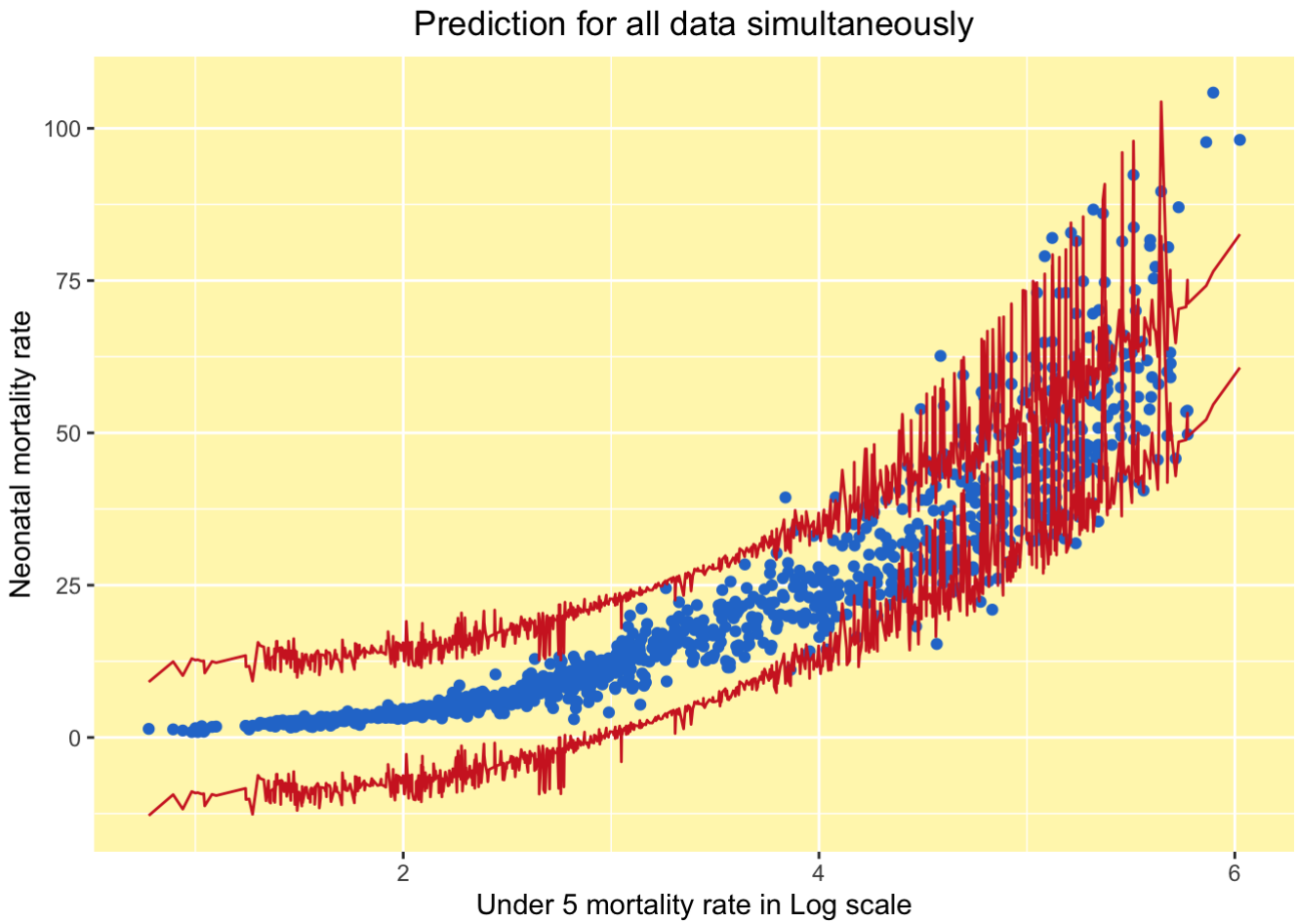
.metric	.estimator	.estimate
rmse	standard	0.42
rsq	standard	0.60
mae	standard	0.30

According to Cornell Statistical Consulting Unit, RMSE is the square root of the variance of the residuals, and “It indicates the absolute fit of the model to the data—how close the observed data points are to the model’s predicted values. (Grace-Martin, 2020) “ , and the lower value of RMSE indicates the better fit of the model

which the value we have here is 0.4122092, meaning the difference between predicted value and the observed value is a bit less than 50%. Moreover, the r-squared is 0.5986680, it means that the predicted model explains about 60% of the observed model. Additionally, MAE which stands for mean absolute error measures the average magnitude of the errors in a set of predictions, mae equals 0.3041219 meaning the errors between prediction and actual observations are about 30%.

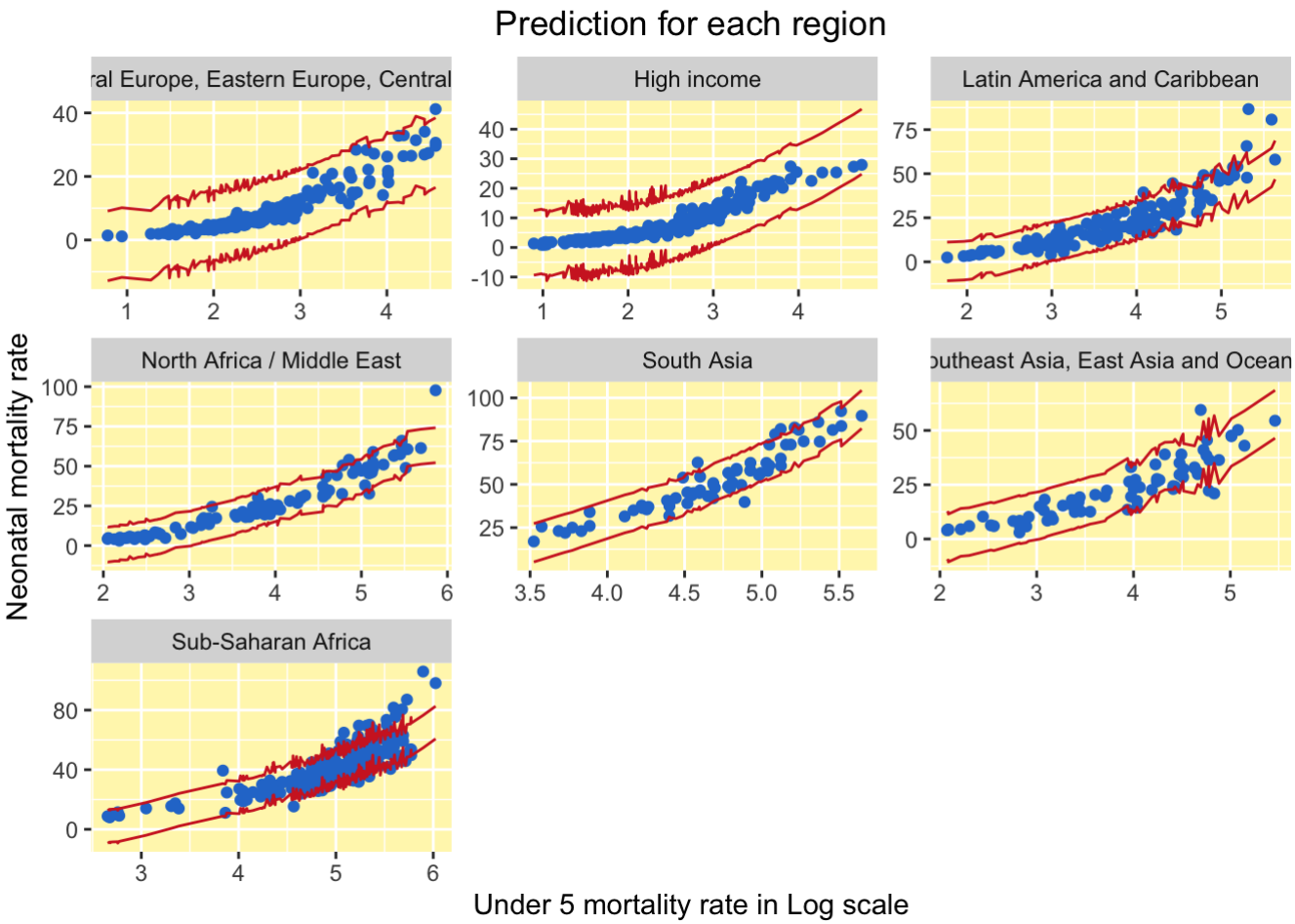
Task 1.1.4: Produce a prediction, with prediction intervals, of the NMR on its natural scale (aka not on the log-scale) and plot these a) for all data simultaneously; b) for data in each region; and c) for data in a maximum of 3 countries that show different aspects of the fit.

```
fit_natural <- lm(nmr ~ u5mr_log + u5mr_log*region + u5mr_log*time, nmr_train)
# Prediction for all data simultaneously
linear_prediction  <- as_tibble(predict(fit_natural, nmr_test, interval = "predictio
n")) %>%
  bind_cols(lm_pred)

linear_prediction  %>%
  ggplot(aes(x = u5mr_log, y = nmr))+
  geom_point(color = "#287ad1")+
  geom_line(aes(y = lwr), color = "#d12828")+
  geom_line(aes(y = upr), color = "#d12828") +
  theme(panel.background = element_rect(fill = "#fff6ba"),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal mortality rate") +
  ggtitle("Prediction for all data simultaneously")
```



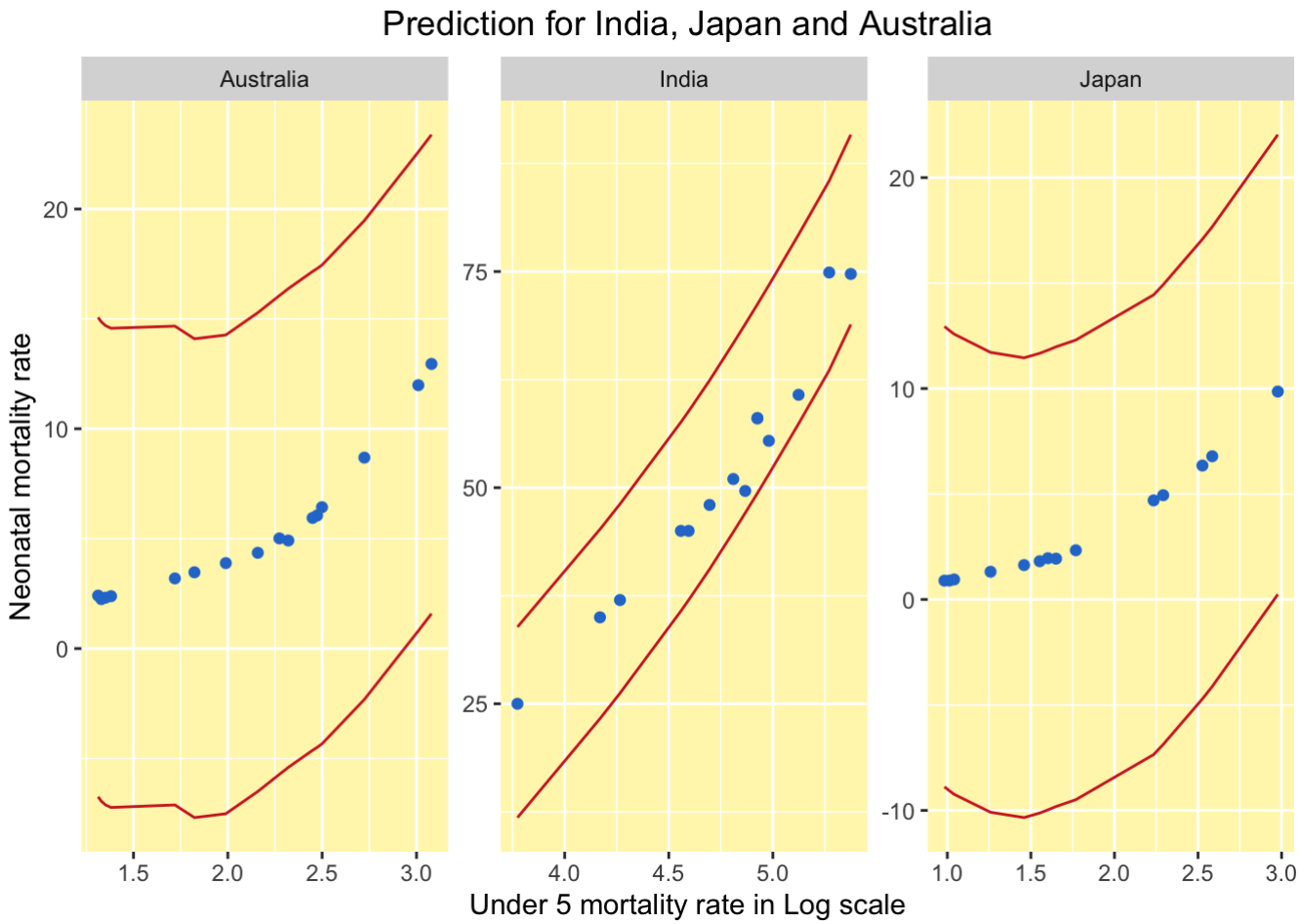
Surprisingly, putting NMR on a natural scale does not look really different from it is put on a log scale, only some values appear to be a little bit extreme. However, putting upper and lower values exclude these outliers.

```
#  Prediction for region
linear_prediction %>%
  ggplot(aes(x = u5mr_log, y = nmr))+
  geom_point(color = "#287ad1")+
  geom_line(aes(y = lwr),color = "#d12828") +
  geom_line(aes(y = upr),color = "#d12828")+
  facet_wrap(~ region, scale = "free")+
  theme(panel.background = element_rect(fill = "#fff6ba"),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal mortality rate") +
  ggtitle("Prediction for each region")
```



As for data in each region, Europe, High income countries, Latin America and Caribbean and Sub-Saharan Africa, the lower and upper scales contain most of the data, and the model performs well.

```
#  Prediction for 3 Countires
 linear_prediction %>%
  filter(country_name %in% c("India", "Japan", "Australia")) %>%
  ggplot(aes(x = u5mr_log, y = nmr))+
  geom_point(color = "#287ad1")+
  geom_line(aes(y = lwr), color = "#d12828") +
  geom_line(aes(y = upr), color = "#d12828")+
  facet_wrap(~ country_name, scale = "free")+
  theme(panel.background = element_rect(fill = "#fff6ba"),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal mortality rate") +
  ggtitle("Prediction for India, Japan and Australia")
```

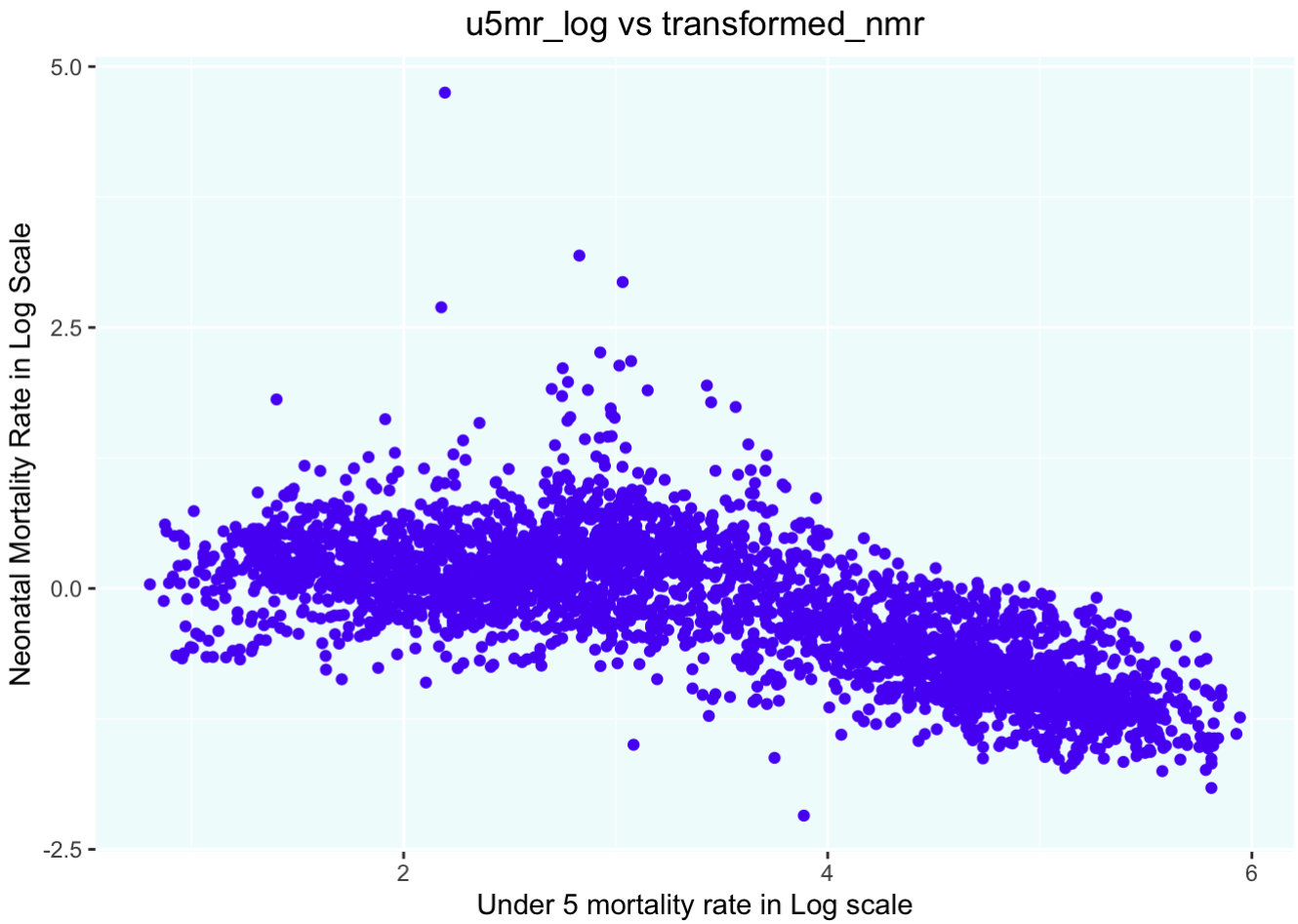


As for the countries, Australia, India and Japan, putting upper and lower scales looks like a better idea than the fit in the log scale, while for Japan, some outliers are still excluded.

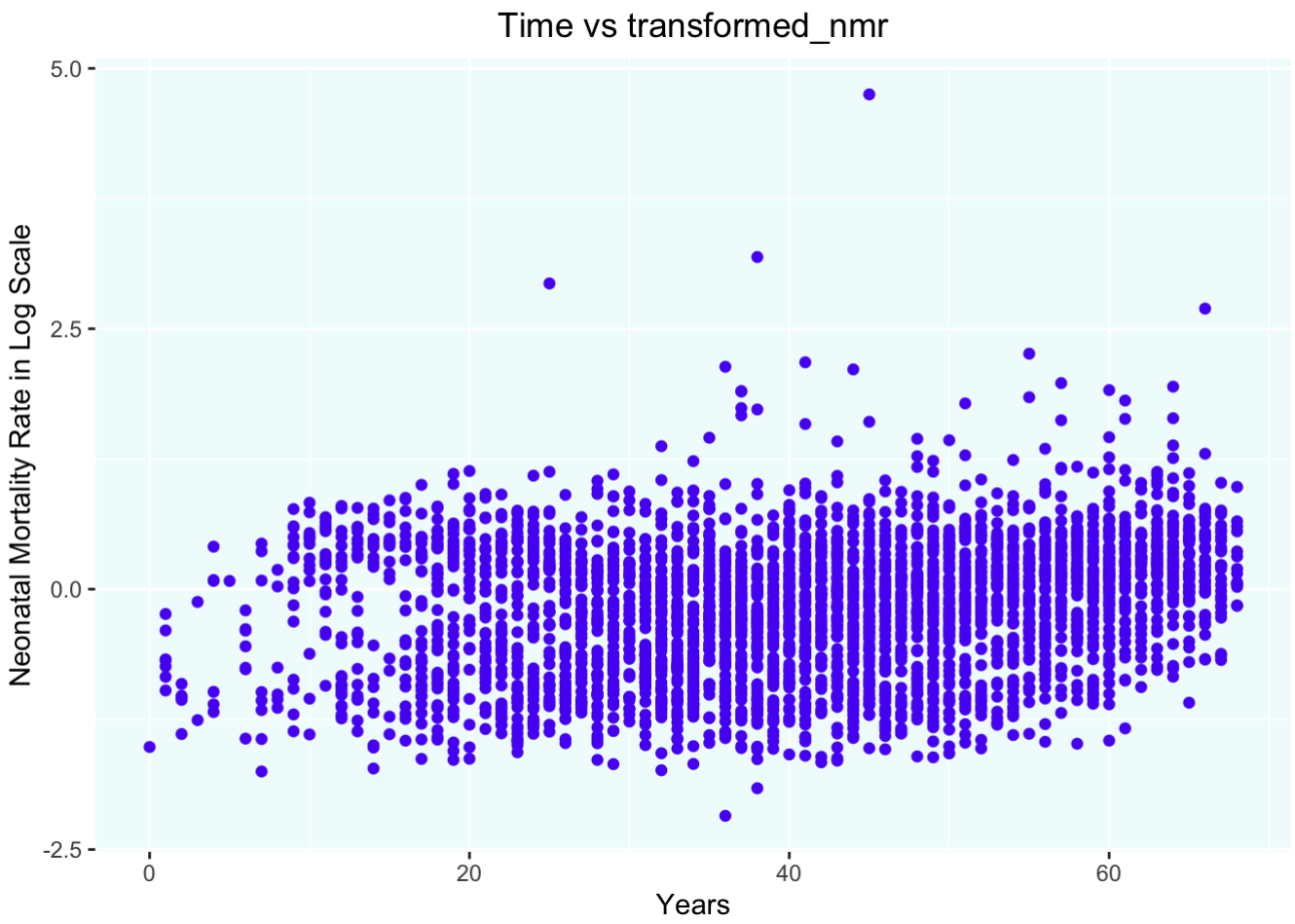
Task 1.2: Non-linear Regression

Task 1.2.1: Explain your choice of model, using appropriate visualisations to support your choice.

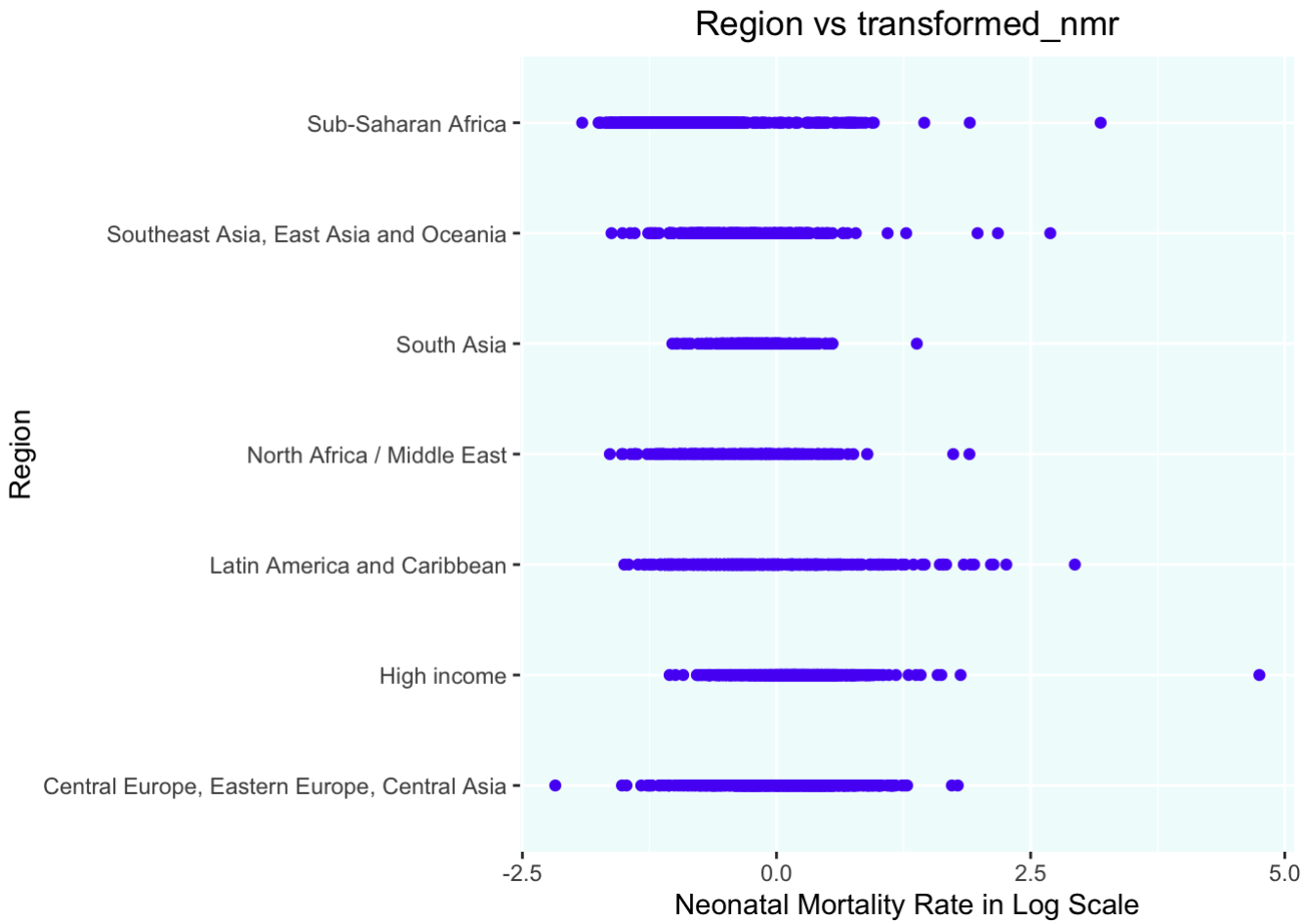
```
#non-linear variable

#non-linear variable part1 choose variable


# find which variables have non-linear relationship
#u5mr_log and transformed_nmr non linear relationship
ggplot(nmr_train)+
  geom_point(aes(x = u5mr_log, y = transformed_nmr), color = "#5905f5")+
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal Mortality Rate in Log Scale") +
  ggtitle("u5mr_log vs transformed_nmr")
```




```
#time and transformed_nmr, no non-linear relationship
ggplot(nmr_train)+
  geom_point(aes(x = time, y = transformed_nmr), color = "#5905f5")+
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  xlab("Years") +
  ylab("Neonatal Mortality Rate in Log Scale")+
  ggtitle("Time vs transformed_nmr")
```




```
#region and transformed_nmr, no non-linear
ggplot(nmr_train)+
  geom_point(aes(y = region, x = transformed_nmr), color = "#5905f5")+
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  ylab("Region") +
  xlab("Neonatal Mortality Rate in Log Scale") +
  ggtitle("Region vs transformed_nmr")
```



We first look for the variables with non-linear relationship. The first graph is u5mr_log and transformed_nmr, the second graph is time and transformed_nmr, the third is region and transformed_nmr, and it appears that only u5mr_log and transformed_nmr has non-linear relationship. We then fit a model with u5mr_log, u5mr_log*region and u5mr_log*time.

Task 1.2.2: Use cross-validation to select an appropriate number of basis functions for bs()

We use folds to split the data into groups.

```
# non-linear model
fit_non_linear <- lm(transformed_nmr ~ bs(u5mr_log, df= 15) + u5mr_log*region + u5mr_
log*time, data = nmr_train)

#fold data into groups
folds <- vfold_cv(nmr_data , v = 20, strata = region)

# create recipe
bs_recipe <- recipe(transformed_nmr ~ u5mr_log + region + time,
              data = nmr_train) %>%
  step_bs(u5mr_log, deg_free = tune()) %>%
  step_dummy(region) %>%
  step_interact(~starts_with("u5mr_log"):starts_with("region"))

# add recipe to the workflow
wf_bs <- workflow() %>%
  add_recipe(bs_recipe)

# create spec
spec <- linear_reg() %>%
  set_engine("lm")


wf_lm <- wf_bs %>%
  add_model(spec)

fit_lm <- wf_lm %>%
  tune_grid(resamples = folds)
```

```
lowest_rmse <- fit_lm %>%
  select_best("rmse")

lowest_rmse %>% knitr::kable()
```

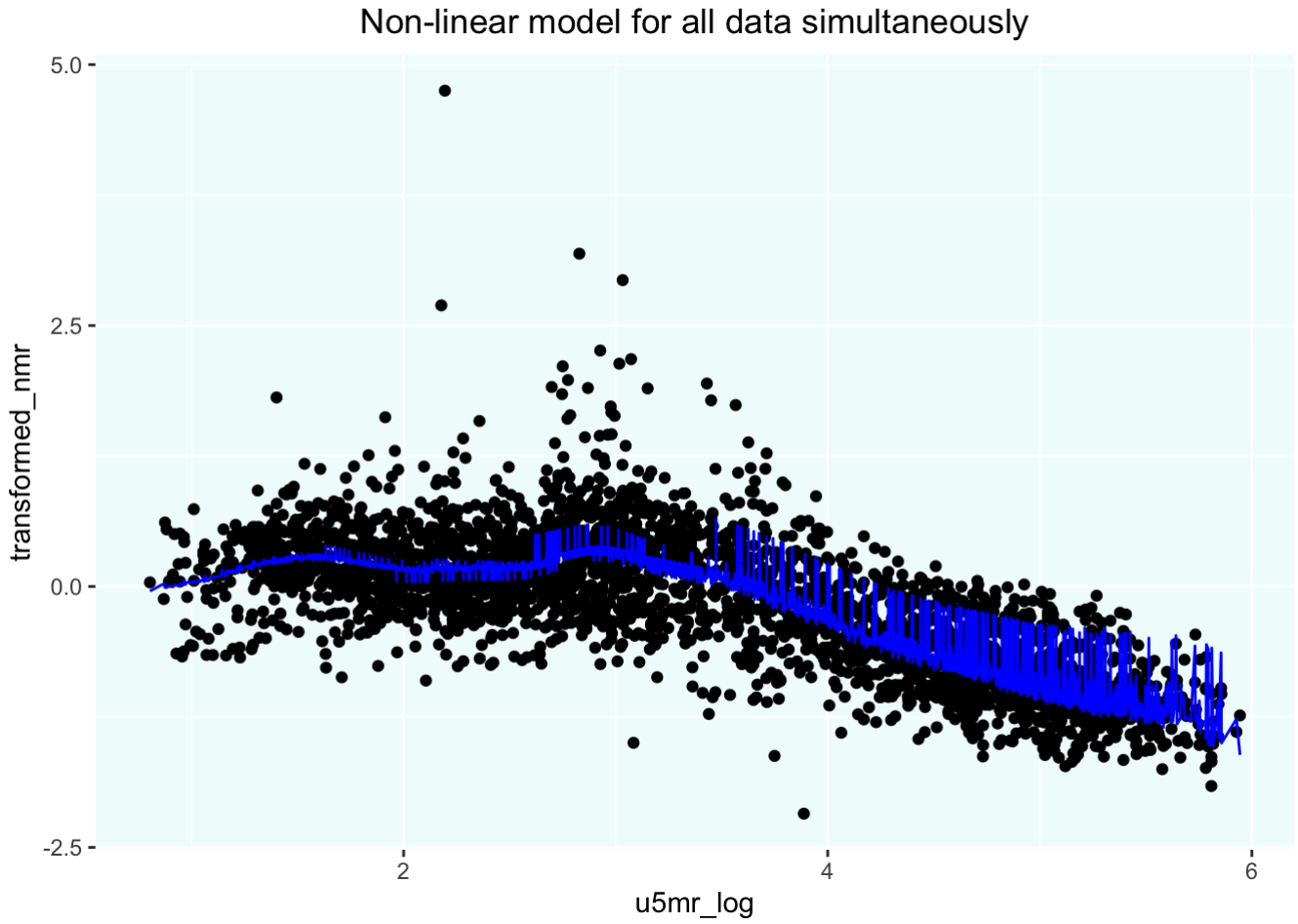
deg_free.config

8Preprocessor7_Model1

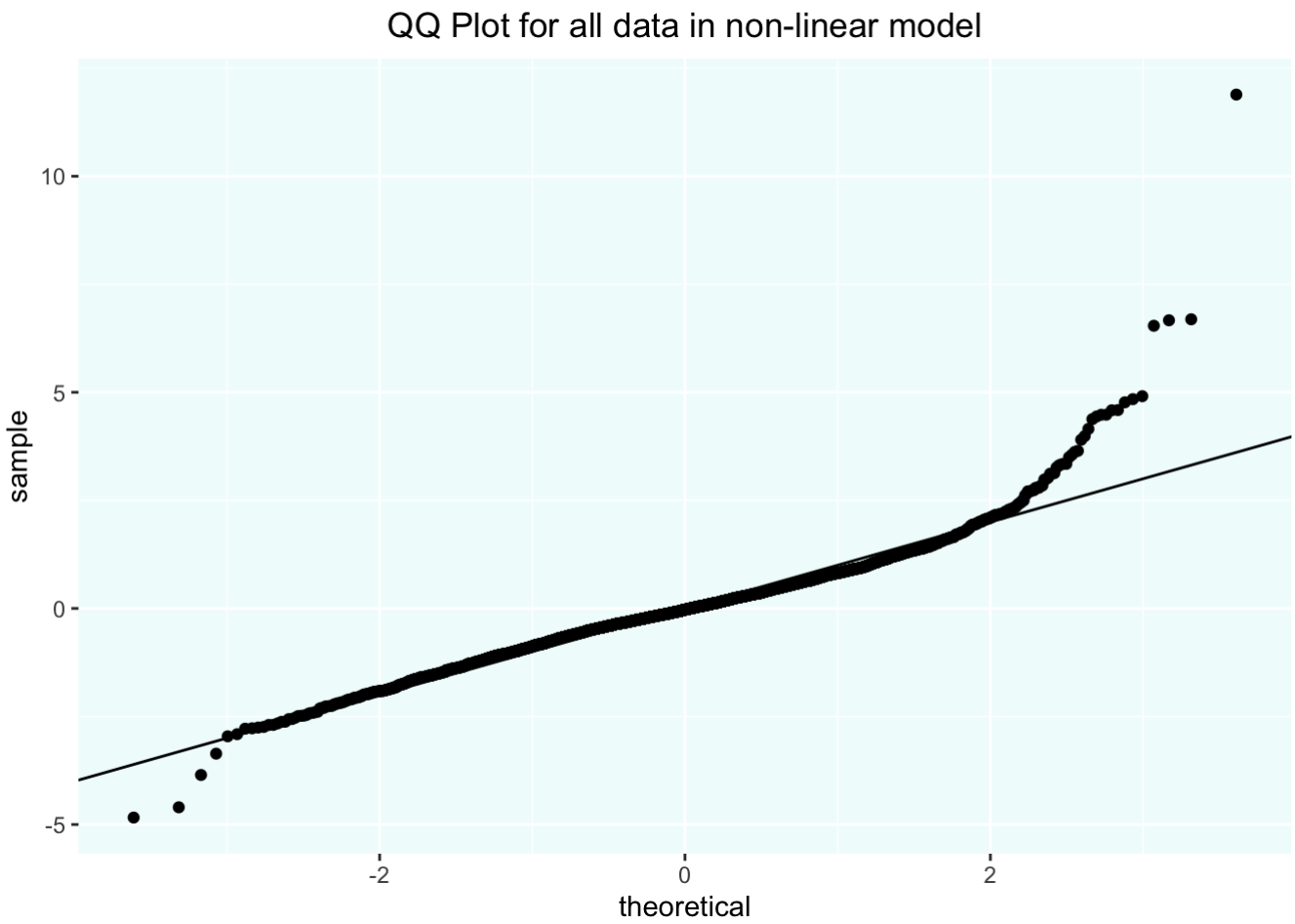
We used cross-validation to select an appropriate number of basis functions for bs(), we first split the data into groups by folds, then create recipes and add it to the workflow. In addition, we set up a spec to discover the value of lambda with the smallest average root square error.From the visualization of the error bar we find the lowest root mean square error to select the degree of freedom to be 8.

Task 1.2.3: For your final model, 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.

```
# non-linear all
augment_non_linear <- fit_non_linear %>%
  augment(data = nmr_train)


augment_non_linear %>%
  ggplot(aes(x = u5mr_log, y = transformed_nmr)) +
  geom_point()+
  geom_line(aes(x = u5mr_log, y = .fitted), color = "blue")+
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("Non-linear model for all data simultaneously")
```

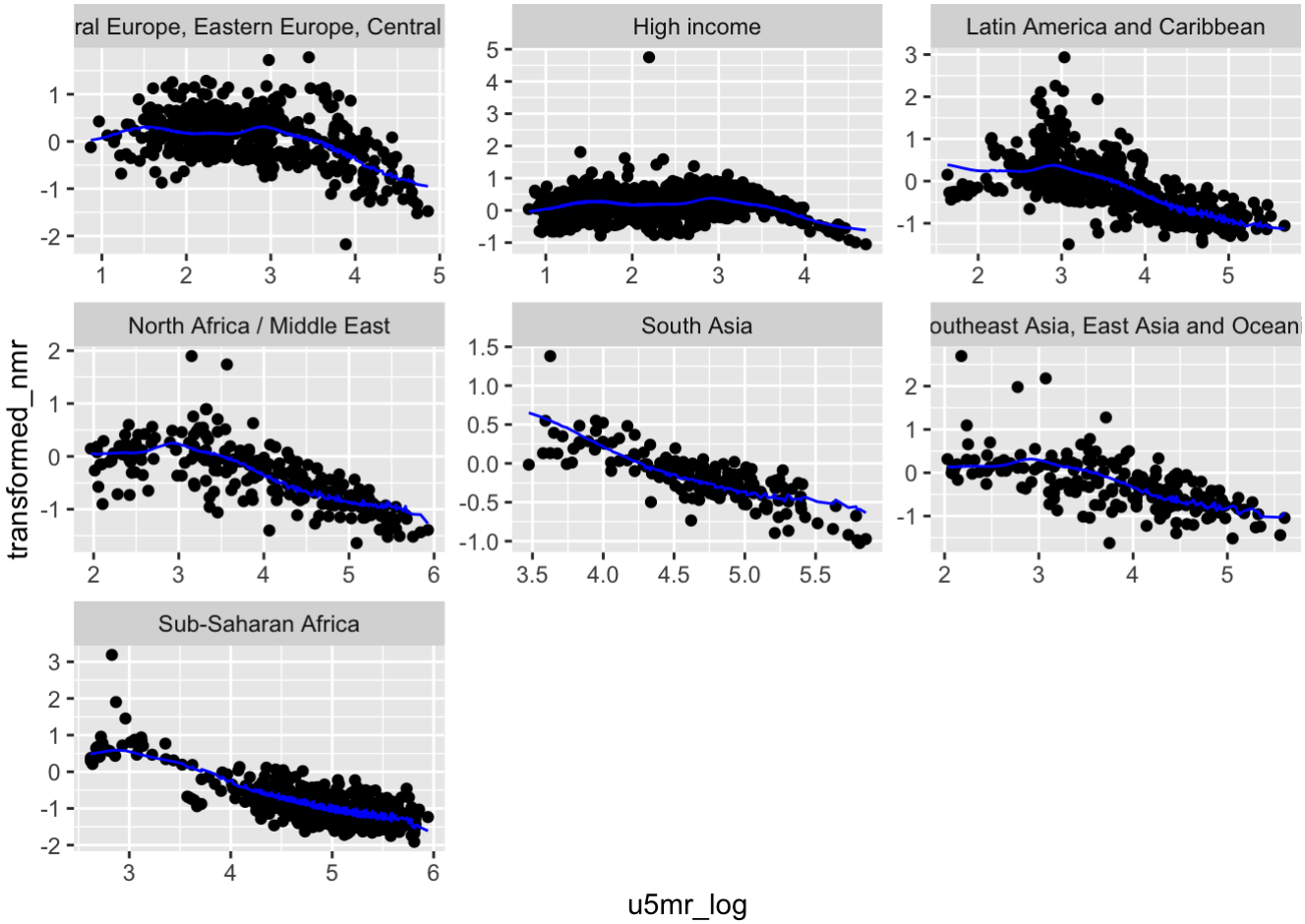



```
augment_non_linear %>%
  ggplot(aes(sample=.resid/.sigma)) +
  geom_qq()+
  geom_abline(intercept=0, slope = 1) +
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("QQ Plot for all data in non-linear model")
```

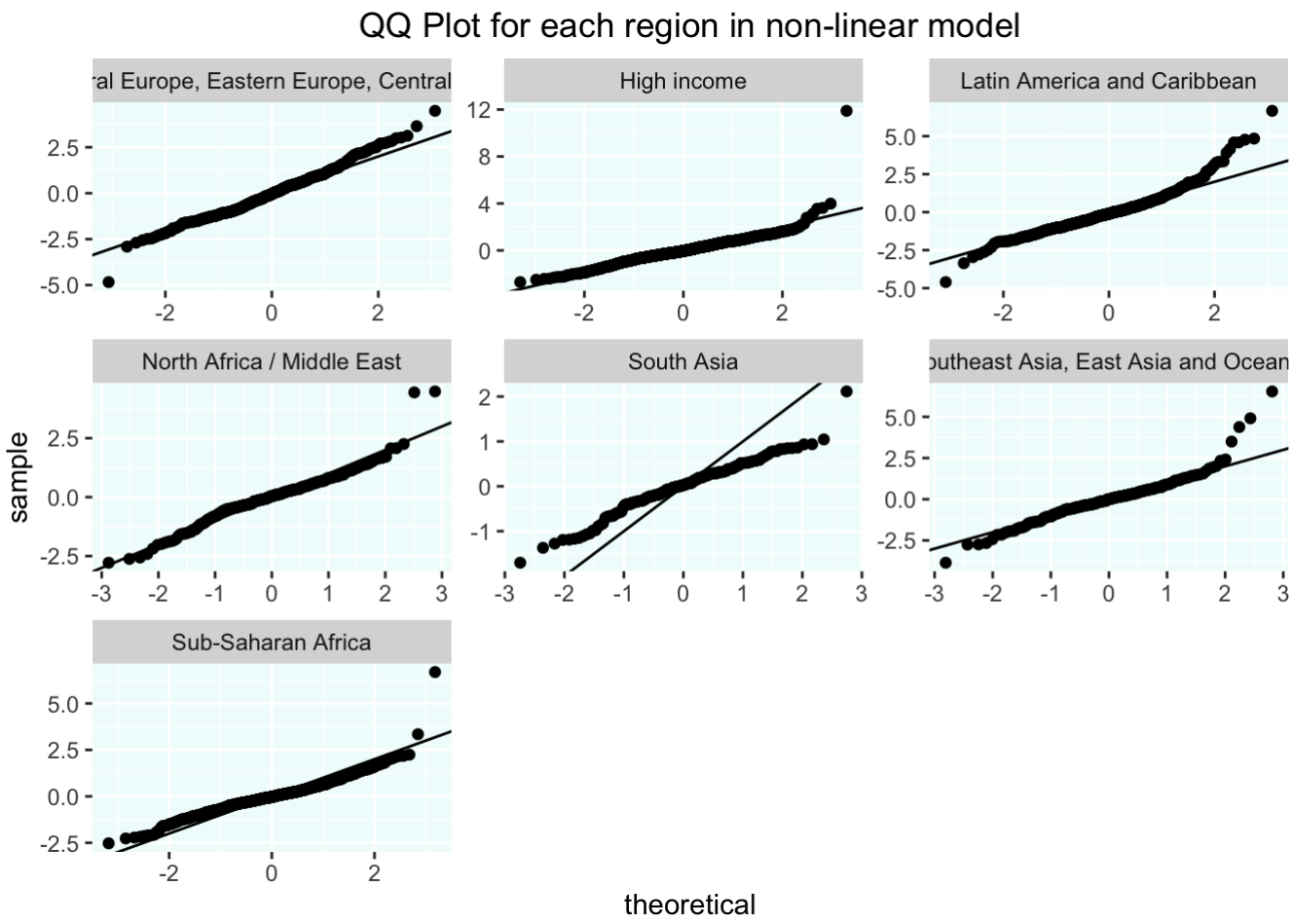


For the non-linear model, fitting the whole data simultaneously looks really similar to the linear model, both of them are good fit. Since the tails in QQ-Plot are not overlapping with the line, it indicates that the data has more extreme values than the values that would be obtained from a normal distribution.

```
#non- linear region
augment_non_linear %>%
  ggplot(aes(x = u5mr_log, y = transformed_nmr)) +
  geom_point()+
  geom_line(aes(x = u5mr_log, y = .fitted), color = "blue")+
  facet_wrap(~ region, scale = "free")
```

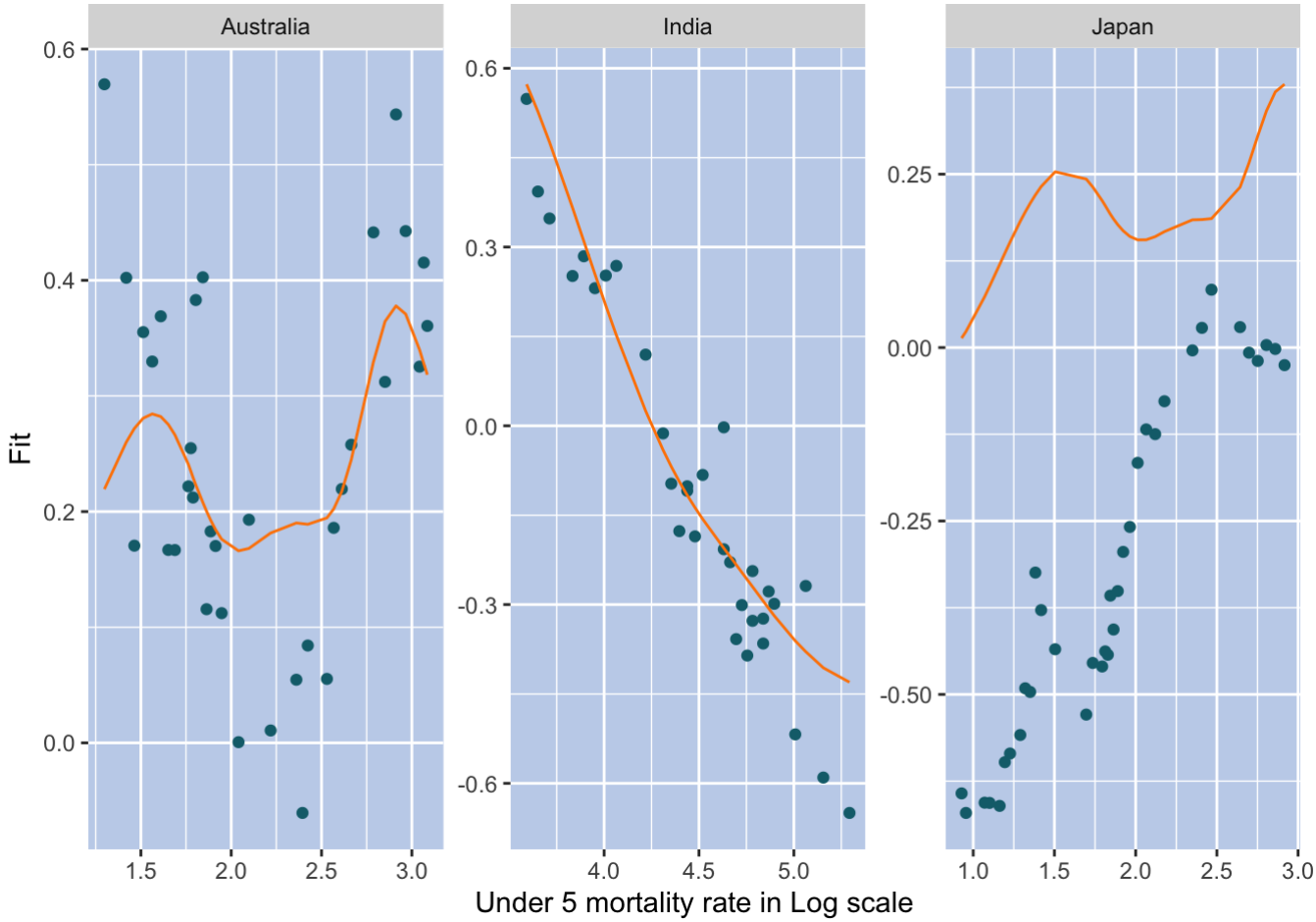



```
augment_non_linear %>%
  ggplot(aes(sample=.resid/.sigma)) +
  geom_qq()+
  geom_abline(intercept=0, slope = 1) +
  facet_wrap(~ region, scale = "free") +
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("QQ Plot for each region in non-linear model")
```

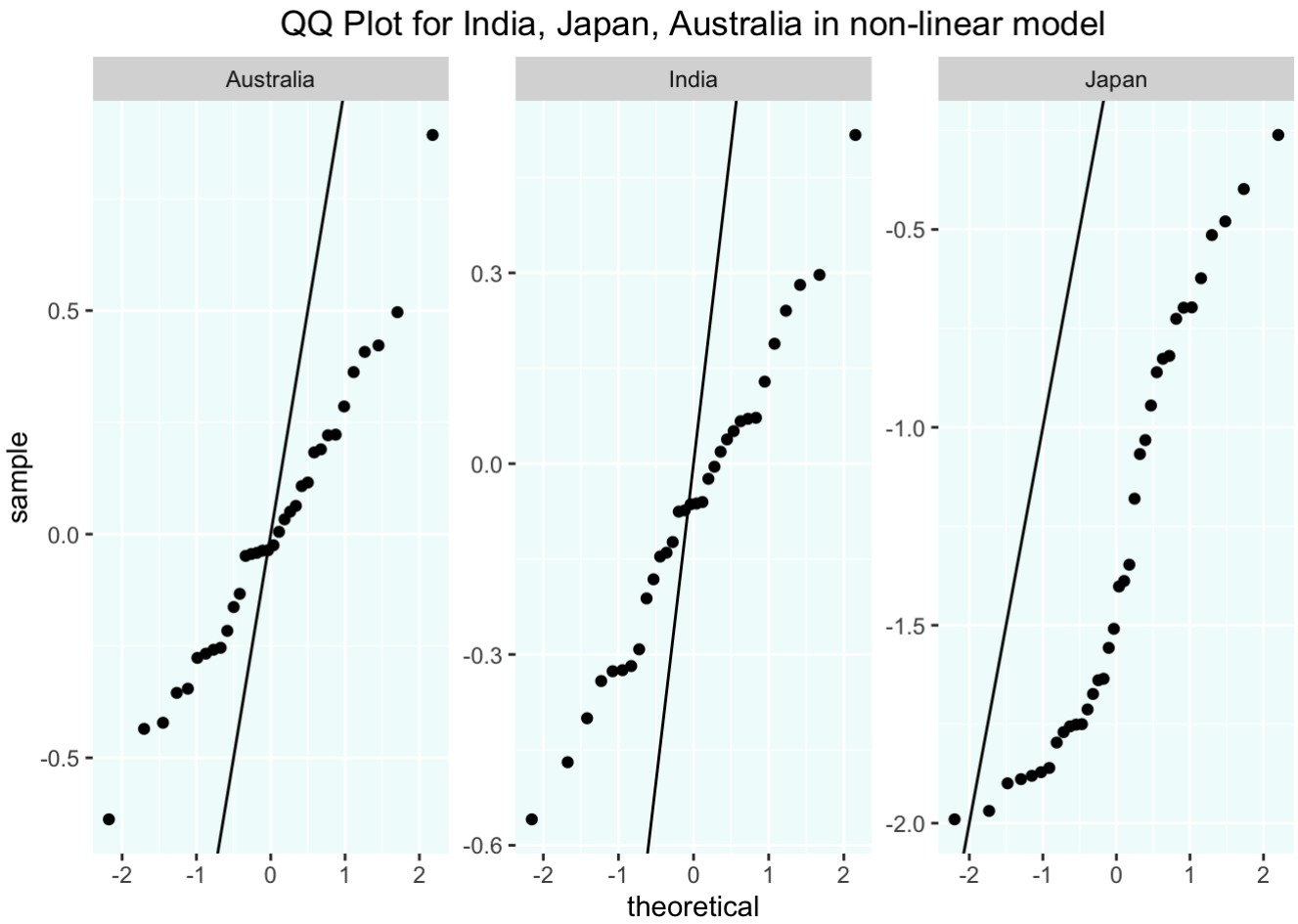


As for the data in each region, non-linear model performs much better than the linear model. It catches all the trends for every region. Based on QQ-Plot, Latin America and Caribbean, Southeast Asia, East Asia and Oceania and South Asia has more extreme values and are not normally distributed.

```
# 3 countries
augment_non_linear %>%
  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")
```




```
augment_non_linear %>%
  filter(country_name %in% c("India", "Japan", "Australia")) %>%
  ggplot(aes(sample=.resid/.sigma)) +
  geom_qq()+
  geom_abline(intercept=0, slope = 1) +
  facet_wrap(~country_name, scales = "free") +
  theme(panel.background = element_rect(fill = "#f0fcfc"),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("QQ Plot for India, Japan, Australia in non-linear model")
```



The non-linear model apparently fits better than the linear model for Australia, India and Japan. Take Australia for example, the model catches log_u5mr around 2.0 to 2.5 better, most of the value at this range are lower than others, and the fit model follows this trend while it is the opposite for the linear model. The predicted values differ largely from the observed values as shown in the QQ-Plot, as there are clearly some violations.

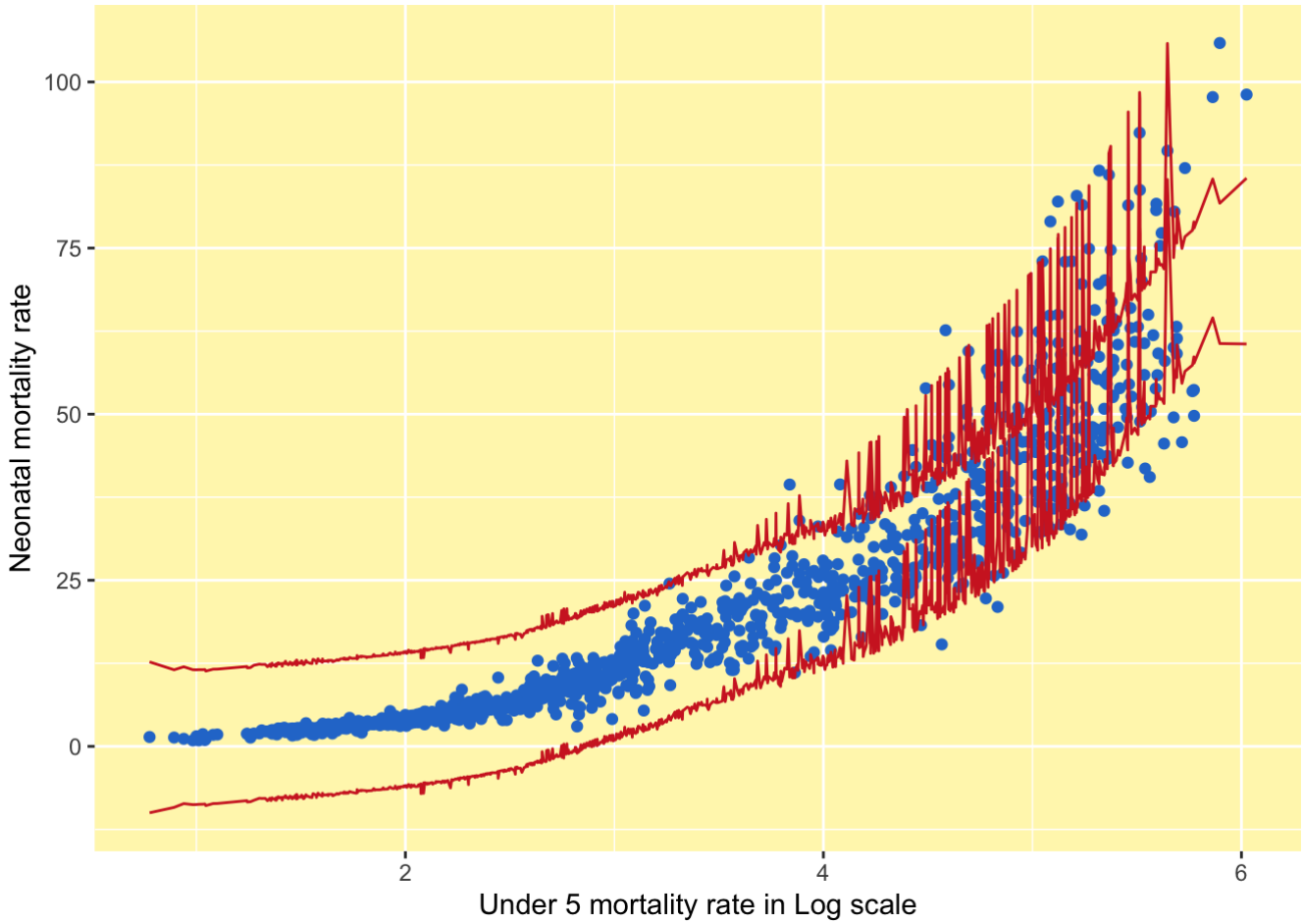
Task 1.2.4: Estimate the root mean square error and the mean absolute error using the same test set as before.

```
# Predict the test dataset
lm_pred <- tibble(pred = predict(fit_non_linear, 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")
```

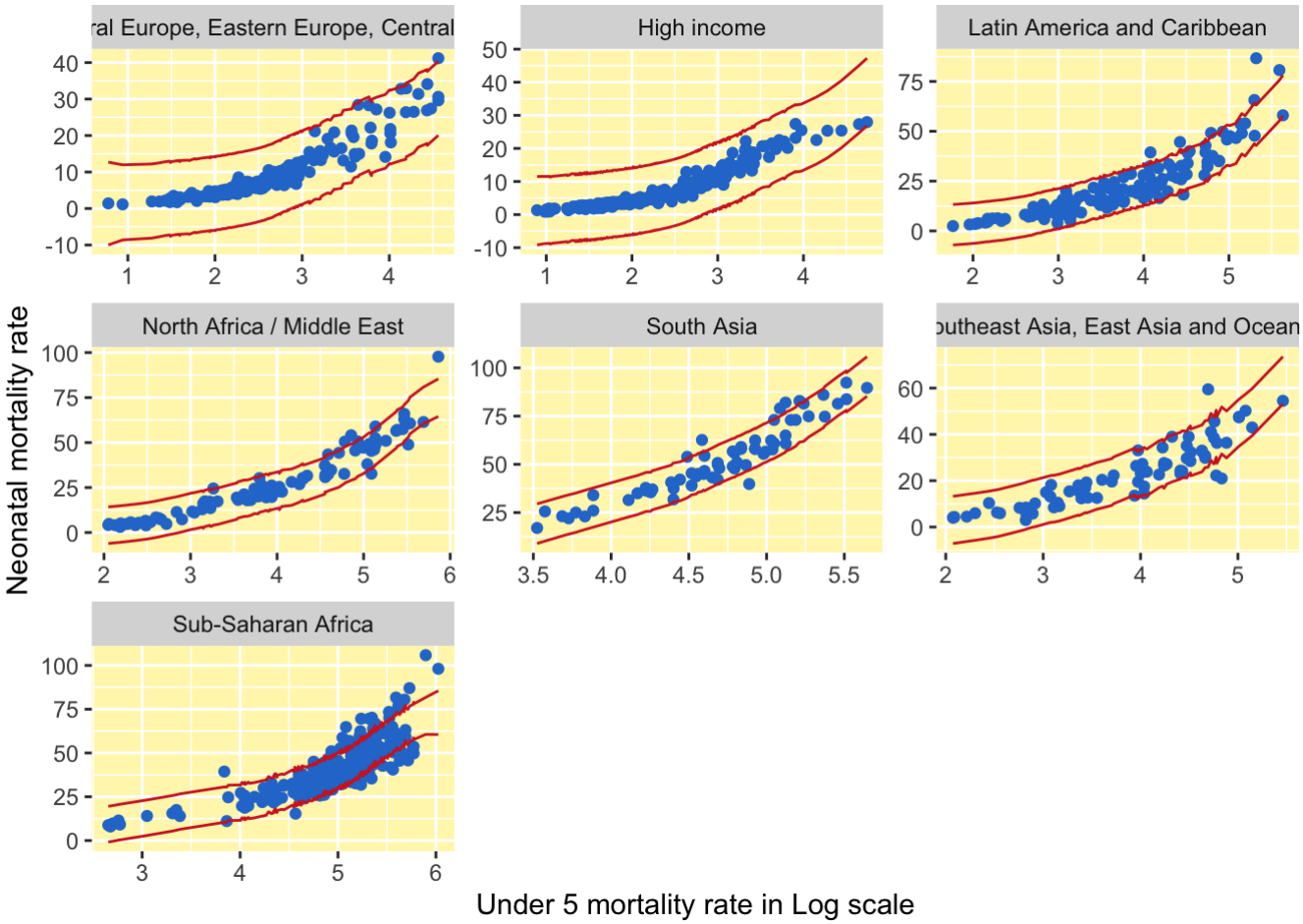
.metric	.estimator	.estimate
rmse	standard	0.40
rsq	standard	0.64
mae	standard	0.28

Task 1.2.5: Produce a prediction, with prediction intervals8, of the NMR on its natural scale (aka not on the log-scale) and plot these a) for all data simultaneously; b) for data in each region; and c) for data in a maximum of 3 countries that show different aspects of the fit.

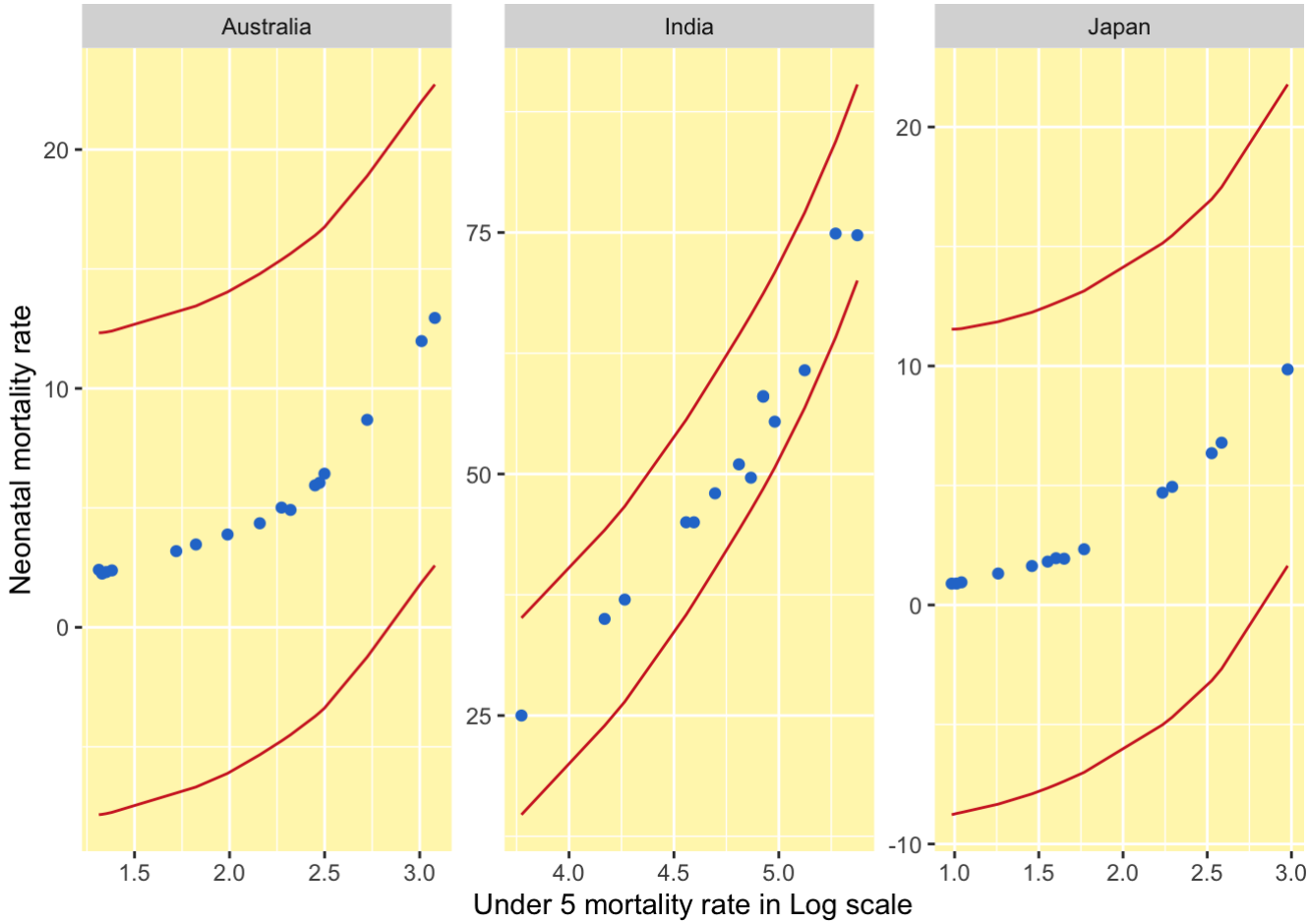
```
fit_non_linear_natural <- lm(nmr ~ bs(u5mr_log, df= 15) + u5mr_log*region + u5mr_log*
time, data = nmr_train)
# Prediction for all data simultaneously
non_linear_prediction <- as_tibble(predict(fit_non_linear_natural, nmr_test, interval
= "prediction")) %>%
  bind_cols(lm_pred)

non_linear_prediction %>%
  ggplot(aes(x = u5mr_log, y = nmr))+
  geom_point(color = "#287ad1")+
  geom_line(aes(y = lwr), color = "#d12828")+
  geom_line(aes(y = upr), color = "#d12828") +
  theme(panel.background = element_rect(fill = "#fff6ba")) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal mortality rate")
```



```
#  Prediction for region
non_linear_prediction %>%
  ggplot(aes(x = u5mr_log, y = nmr))+
  geom_point(color = "#287ad1")+
  geom_line(aes(y = lwr), color = "#d12828") +
  geom_line(aes(y = upr), color = "#d12828")+
  facet_wrap(~ region, scale = "free")+
  theme(panel.background = element_rect(fill = "#fff6ba")) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal mortality rate")
```




```
#  Prediction for 3 Countires
 non_linear_prediction %>%
  filter(country_name %in% c("India", "Japan", "Australia")) %>%
  ggplot(aes(x = u5mr_log, y = nmr))+
  geom_point(color = "#287ad1")+
  geom_line(aes(y = lwr), color = "#d12828") +
  geom_line(aes(y = upr), color = "#d12828")+
  facet_wrap(~ country_name, scale = "free")+
  theme(panel.background = element_rect(fill = "#fff6ba")) +
  xlab("Under 5 mortality rate in Log scale") +
  ylab("Neonatal mortality rate")
```



Conclusion for Task 1:

The first model is the linear regression model and second is the non-linear regression model. The MAE and the RMSE are significant, which means the errors of the predicted model cannot be ignored. The non-linear model is more appropriate model to explain the data. However, the R-squared value of the linear model and the non-linear model do not show much difference indicating that the transformed_nmr, u5mr_log, region and time variables are not highly dependent on each other. From the graph of the model and qq plots, especially from for each region and the three different countries, non-linear model have better fit over the linear model.

Task 2: The residual bootstrap

Introduction:

In this section we wrote a function that uses the residual bootstrap to compute an 80% confidence interval for the LASSO estimate for each regression parameters also check the coverage of each of the intervals.

```
#data
set.seed(123)

p <- 14
n <- 10000
n_sim <- 10

data <- matrix(rnorm(n*p), nrow = n, ncol = p) %>%
  as_tibble() %>%
  mutate(y =V1+2*V2-3*V3+V4+rnorm(n, sd = 0.4))

variables <- paste0("V", 1:14)
formula <- reformulate(variables, response = "y")
```

Write a function that uses a residual bootstrap to compute an 80% confidence interval for the LASSO estimate for each regression parameters.

```r
# write the function
library(glmnet)

lasso<- function(data, formula, n, n_sim){

# model with tune to find a good value of the penalty parameter
  spec_lasso_tune <- linear_reg(penalty = tune(), mixture = 1) %>%
    set_engine("glmnet")

  lambda_grid <- grid_regular(penalty(range = c(-5,5)), levels = 10)

#cross validation
  folds <- vfold_cv(data, v = 10)

#workflow
  rec <- recipe(formula, data = data)

  lasso_wf <- workflow() %>%
    add_recipe(rec) %>%
    add_model(spec_lasso_tune)

  lasso_wf_tune <- lasso_wf %>%
    tune_grid(resamples = folds,
              grid = lambda_grid)

#choose smallest mse
  smallest_rmse <- lasso_wf_tune %>%
    select_best("rmse")

#update lasso
  final_lasso <- finalize_workflow(lasso_wf, smallest_rmse)

  fit_lasso <- final_lasso %>%
    fit(data)

#Find intercept from v1 to v14 and predict estimate penality
  fit_beta <- final_lasso %>%
    fit(data) %>%
    tidy()

# compute residual (real - pred value)
  pred_value <- data %>%
    bind_cols(predict(fit_lasso, data))

  residuals <- pred_value %>%
    summarise(resid = y - .pred)

#Normalize the residuals
  normalized_residuals <- residuals %>%
    summarise(normalized_resid = resid - mean(resid))

#Re-sample the residuals with replacement to create non-parametric bootstrap data
  np_bs <- tibble(experiment = rep(1:n_sim, each = n),
                  index = sample(1:n, size = n*n_sim, replace = TRUE),
                  resid_star = normalized_residuals$normalized_resid[index],
                  pred_value[index,]) %>%  # corresponding predict y value
  mutate(y_hat = .pred + resid_star)
```

```
update_rec <- recipe(y_hat ~ ., data = np_bs[,-1])

experiment <-  np_bs %>%
  group_by(experiment) %>%
  nest()

fit_list <- list()

#Fit the LASSO to the bootstrap data
  for (i in 1:dim(experiment)[1]){

    #workflow
    updatde_lasso_wf <- workflow() %>%
    add_recipe(update_rec) %>%
    add_model(spec_lasso_tune)

    folds <- vfold_cv(experiment$data[[i]], v= 10)

    lasso_wf_tune <- updatde_lasso_wf%>%
    tune_grid(resamples = folds,
            grid = lambda_grid)

    smallest_rmse <- lasso_wf_tune %>%
    select_best("rmse")

  update_final_lasso <- finalize_workflow(updatde_lasso_wf, smallest_rmse)

  fit_list[[i]] <- update_final_lasso %>%
    fit(data = experiment$data[[i]]) %>%
    tidy() %>%
    mutate(exp = i)
  }

fit_beta_star <- bind_rows(fit_list) %>%
  rename(bootstrap_estimate = estimate)

#Compute the bias
bias <- fit_beta_star %>%
  group_by(exp) %>%
  mutate(delta = fit_beta$estimate - bootstrap_estimate) %>%
  left_join(fit_beta, by = "term")


#confidence interval
ci <- bias %>%
  group_by(term) %>%
  summarise(lwr = estimate +quantile(delta, 0.1),
            upr = estimate +quantile(delta, 0.9)) %>%
  unique()

coverage <- bias %>%
  left_join(ci, by = "term") %>%
  mutate(covered = ifelse(bootstrap_estimate > lwr  & bootstrap_estimate < upr, 1,0))
%>%
  group_by(term) %>%
  summarise(coverage = mean(covered))
```

```
return(list(ci, coverage))
}
```

This is an outcome of the sample data of 80% confidence interval for the LASSO estimate

```
# This is an outcome of the sample data of 80% confidence interval  for the LASSO est
imate
alist <- lasso(data, formula, n, n_sim)

ci <- alist[1]
coverage <- alist[2]

ci %>%
  kbl(booktabs = T,
      caption = "80% confidence interval for Lasso") %>%
  kable_styling(position = "center")
```

80% confidence interval for Lasso

term	lwr	upr
.pred	NA	NA
(Intercept)	0.00	0.00
index	NA	NA
resid_star	NA	NA
V1	-2.00	-2.00
V10	0.00	0.00
V11	0.00	0.00
V12	0.00	0.00
V13	0.00	0.00
V14	0.98	0.98
V2	2.99	2.99
V3	-2.98	-2.98
V4	0.99	0.99
V5	0.00	0.00
V6	0.00	0.00
V7	0.00	0.00
V8	0.00	0.00
V9	0.00	0.00
y	NA	NA

Write a function that checks the coverage of each of these intervals and Comment on whether or not the (modified) residual bootstrap achieves the nominal coverage

```
coverage %>%
  kbl(booktabs = T,
      caption = "Coverage check for each interval") %>%
  kable_styling(position = "center")
```

Coverage check for each interval

term	coverage
.pred	NA
(Intercept)	0.1
index	NA
resid_star	NA
V1	0.0
V10	0.0
V11	0.0
V12	0.0
V13	0.0
V14	0.0
V2	0.0
V3	0.0
V4	0.0
V5	0.0
V6	0.0
V7	0.0
V8	0.0
V9	0.0
y	NA

Conclusion for Task 2:

From the coverage table we can see that the modified residual bootstrap does not achieve the nominal coverage.