Data Wrangling Assignment 8

Yaniv Bronshtein

3/26/2021

Import the necessary libraries

```
library(gapminder)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggbeeswarm)
```

```
## Warning: package 'ggbeeswarm' was built under R version 4.0.4
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(rlist)
```

```
## Warning: package 'rlist' was built under R version 4.0.4
```

```
library(broom)
library(modelr)
```

```
##
## Attaching package: 'modelr'
```

```
## The following object is masked from 'package:broom':
##
##     bootstrap
```

PROBLEM 1:

For the gapminder data, perform the following operations, using the tidyr::nest() function and data frames with list-columns:

1.Fit a separate linear model of log10(gdpPercap) on year for each country. 2.Plot residuals against time, showing separate lines for each country in the same plot. Also, do this separately for each continent. 3.Create a continent-wise Beeswarmplot for (i) value of the estimated slope coefficient and (ii) value of the t-statistic (ratio of estimate and standard error). [Hint: You may need to revisit the materials on broom package]. Interpret the plots. 4.Identify the countries that have estimated negative slopes and p-values less than 0.05. What is the interpretation of the linear model fit for these countries? 5.Plot the year-wise log10(gdpPercap) for the countries identified in step d)

1.Fit separate linear model of log10(gdpPercap) on year for each country

```
gap_nested <- gapminder %>%
  group_by(country, continent) %>%
  nest()
```

Create a function country_lm to train an lm model on a dataframe

```
country_lm <- function(df) {
  lm(log10(gdpPercap) ~ year, data = df)
}
```

Apply country_lm() to every element in the data column of the gap_nested df Create a new column in gap_nested from this operation

```
gap_nested <-gap_nested %>%
  mutate(model = map(data, country_lm))

gap_nested
```

```
## # A tibble: 142 x 4
## # Groups:   country, continent [142]
##    country     continent data              model
##    <fct>       <fct>     <list>            <list>
##  1 Afghanistan Asia      <tibble [12 x 4]> <lm>
##  2 Albania     Europe    <tibble [12 x 4]> <lm>
##  3 Algeria     Africa    <tibble [12 x 4]> <lm>
##  4 Angola      Africa    <tibble [12 x 4]> <lm>
##  5 Argentina   Americas  <tibble [12 x 4]> <lm>
##  6 Australia   Oceania   <tibble [12 x 4]> <lm>
##  7 Austria     Europe    <tibble [12 x 4]> <lm>
##  8 Bahrain     Asia      <tibble [12 x 4]> <lm>
##  9 Bangladesh  Asia      <tibble [12 x 4]> <lm>
## 10 Belgium     Europe    <tibble [12 x 4]> <lm>
## # ... with 132 more rows
```

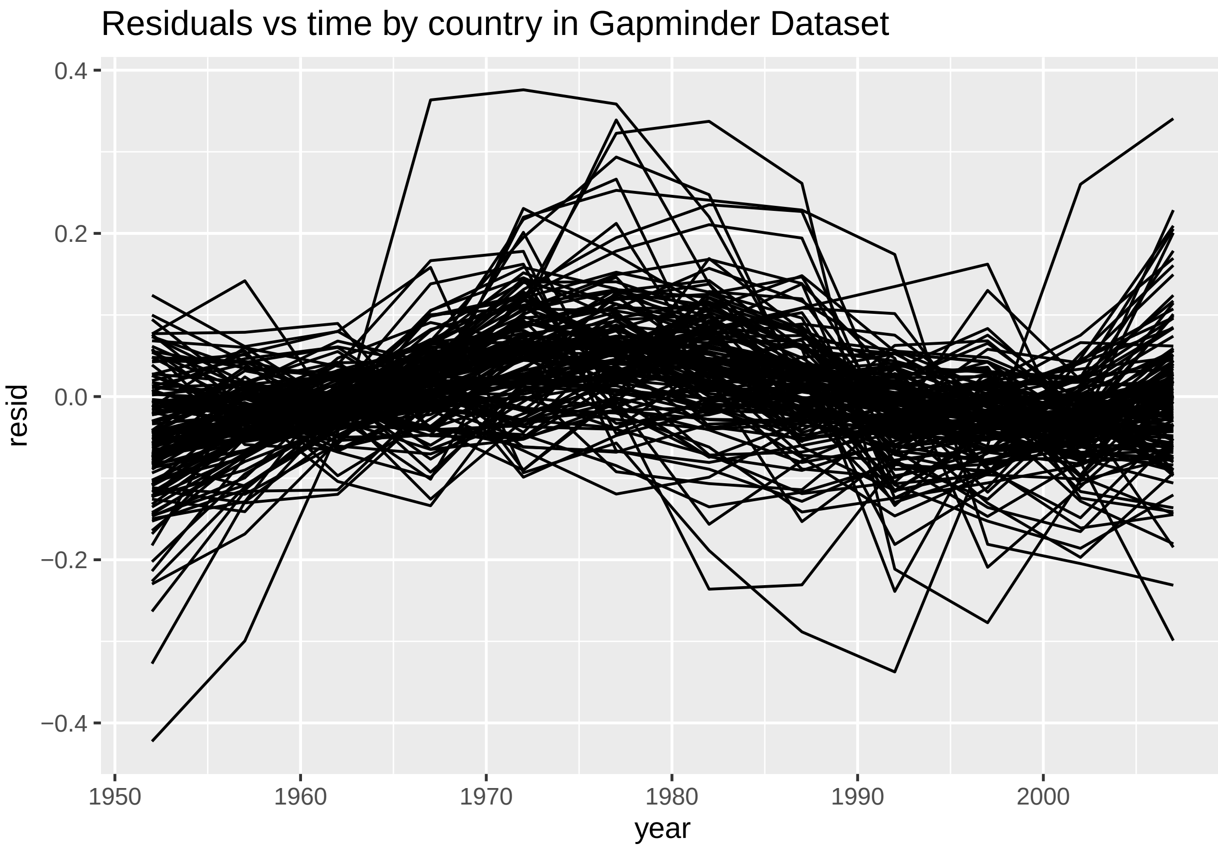
Use map2() to iterate simulataneously over the data and model columns and applying add_residuals to generate the resid column Then, use unnest() to extract the residual values

```
gap_nested_lm <- gap_nested %>%
  mutate(resid = map2(data, model, add_residuals))

resid <- unnest(gap_nested_lm, resid)
```

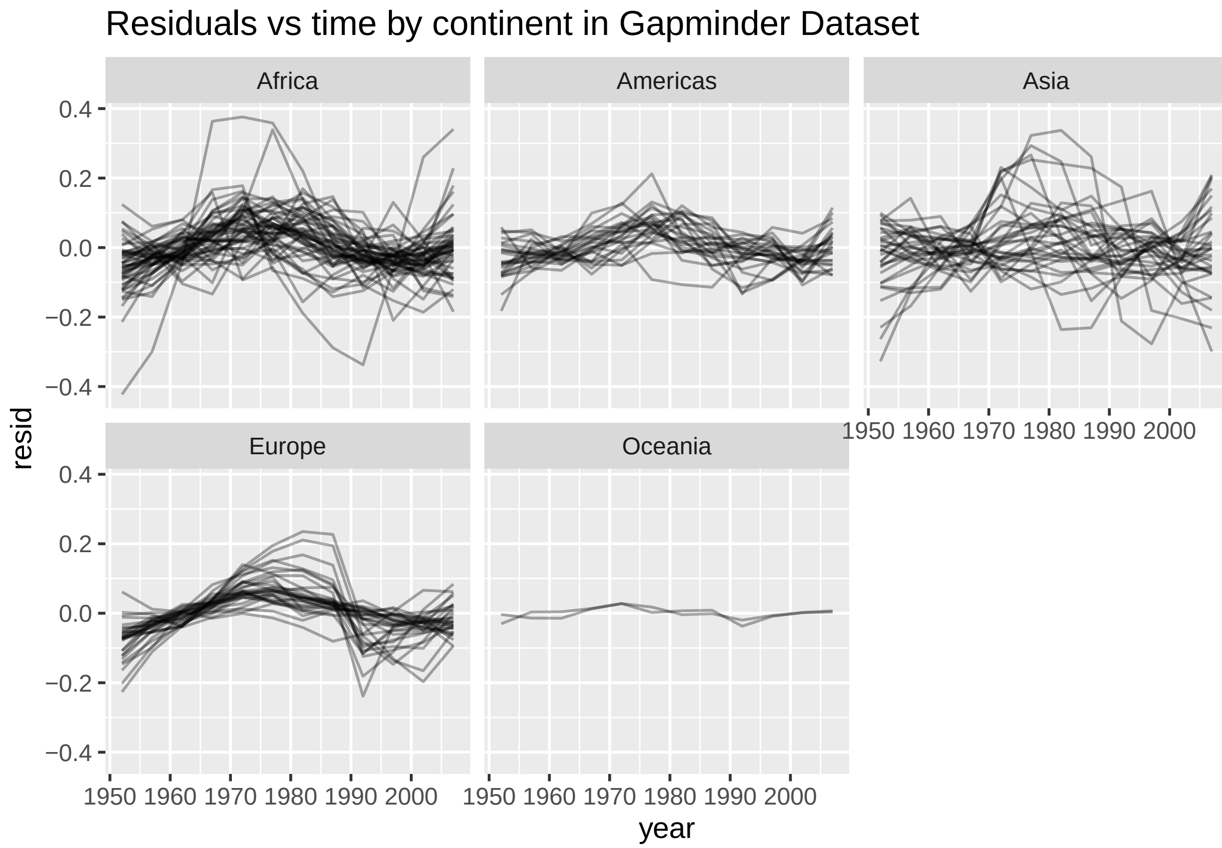
2.Plot residuals against time, showing separate lines for each country in the same plot.

```
ggplot(data = resid, mapping = aes(x = year, y = resid, group = country)) +
  geom_line() +
  labs(
    title = "Residuals vs time by country in Gapminder Dataset"
  )
```



Also do this separately for each continent

```
ggplot(data = resid, mapping = aes(x = year, y = resid, group = country)) +
  geom_line(alpha = 1/3) +
  labs(
    title = "Residuals vs time by continent in Gapminder Dataset"
  ) +
  facet_wrap(~continent)
```



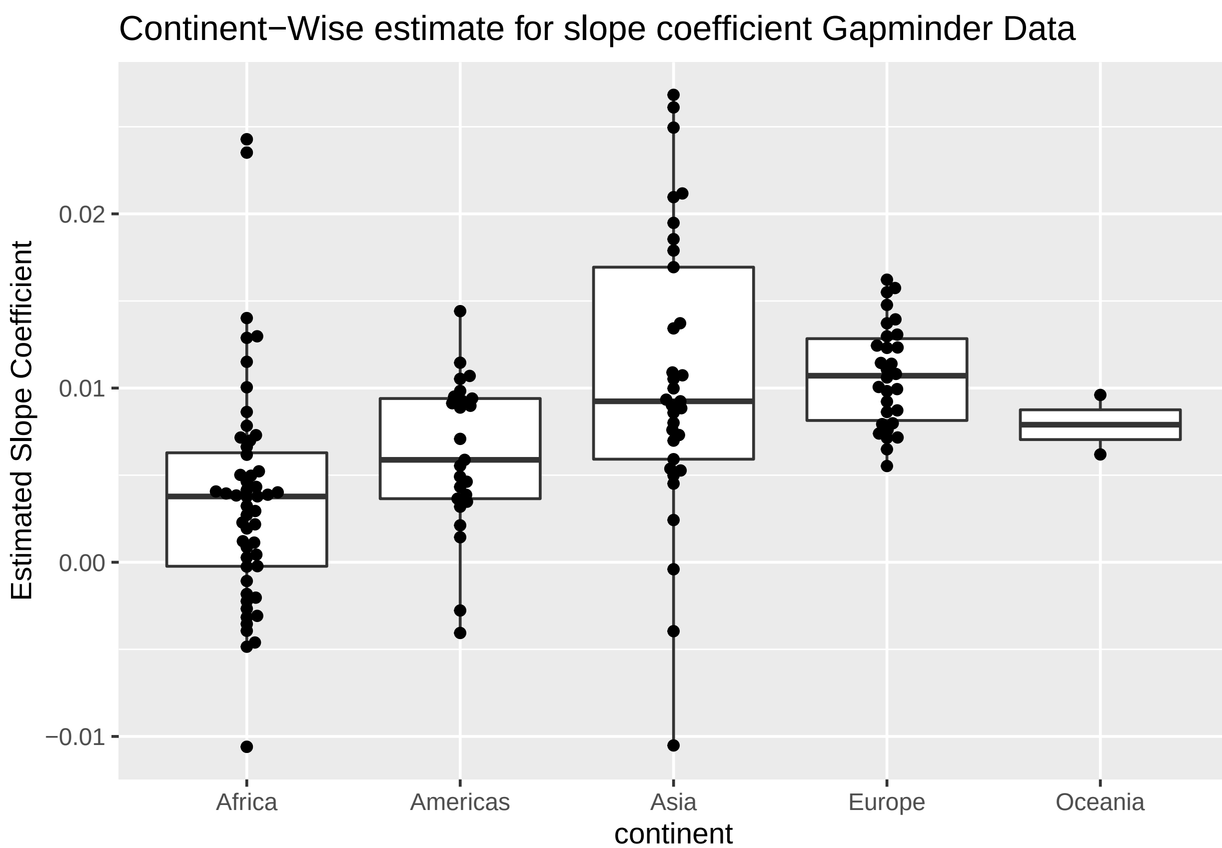
3. Create a continent-wise Beeswarmplot for (i) value of the estimated slope coefficient

```
#Apply tidy() to get statistics and estimate information
gap_nested_summary <- gap_nested %>%
  mutate(lm_tidy = map(model, tidy))

#Filter out the year after unnest() operation
summary_stat_bycontinent <- unnest(gap_nested_summary, lm_tidy) %>%
  filter(term == "year")
```

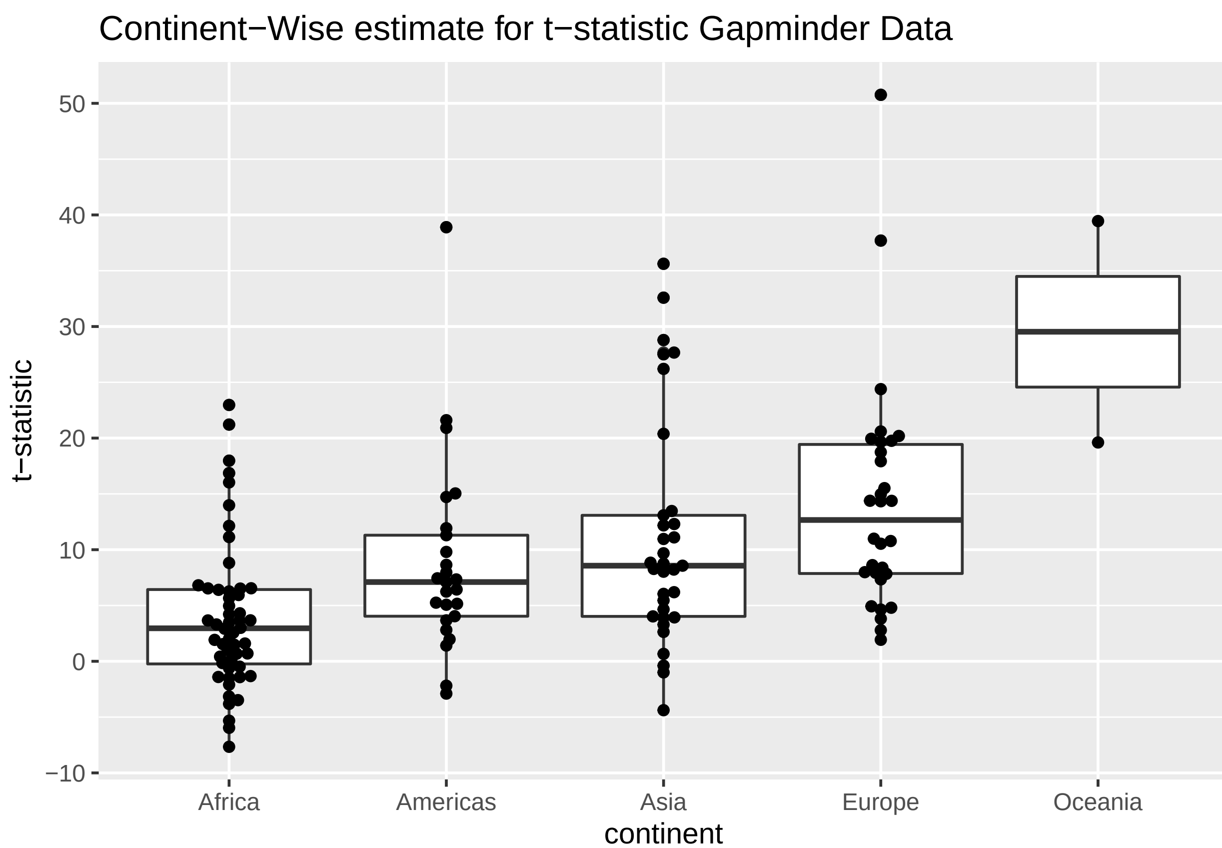
Generate the beeswarm() plot

```
ggplot(data = summary_stat_bycontinent,
       mapping = aes(continent, estimate)) +
  geom_boxplot() +
  geom_beeswarm() +
  labs(
    title = "Continent-Wise estimate for slope coefficient Gapminder Data",
    y = "Estimated Slope Coefficient"
  )
```



(ii). value of the t-statistic

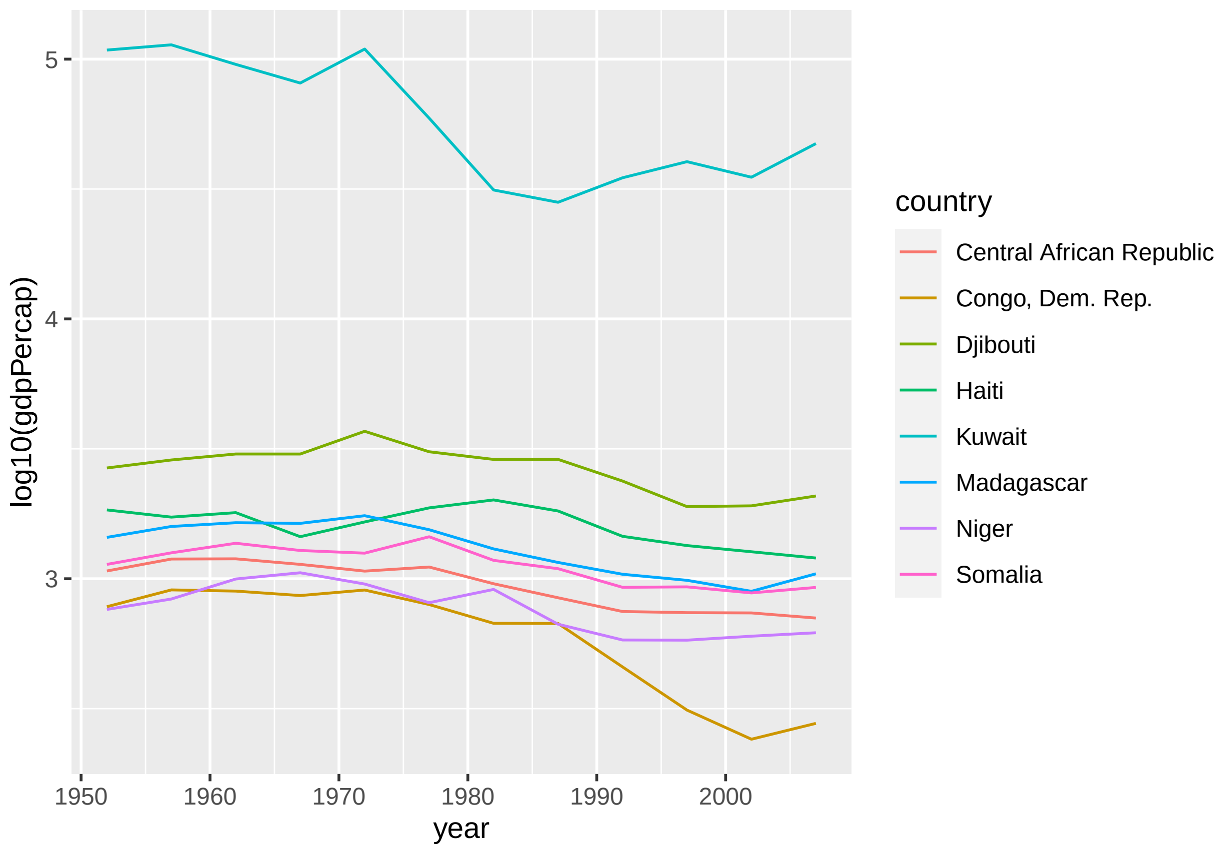
```
ggplot(data = summary_stat_bycontinent,
       mapping = aes(continent, statistic)) +
  geom_boxplot() +
  geom_beeswarm() +
  labs(
    title = "Continent-Wise estimate for t-statistic Gapminder Data",
    y = "t-statistic"
  )
```



4.Identify the countries that have estimated negative slopes and p-values less than 0.05. What is the interpretation of the linear model fit for these countries?

```
bad_fit <-
  summary_stat_bycontinent %>%
  filter(estimate < 0, p.value < 0.05)

gapminder %>%
  semi_join(bad_fit, by = "country") %>%
  ggplot(mapping = aes(x = year, y = log10(gdpPercap), color = country)) +
  geom_line()
```




```
#Interpretation:
#Kuwait
#Djibouti
#Haiti
#Madagascar
#Somalia
#Central African Republic
#Congo
#Niger

#Based on the graph, Kuwait had drastic drop in log10(gdpPercap)
#between 1972 and 1980 and never recovered. However it still
#maintains a logd10(gdpPercap) difference of at least 1 throughout
#the entire time range. Cog started off low but had the most dramatic
#drop in log10(gdpPercap) after around 1982.
#Aside from Kuwait and Congo, the countries maintained a log10(gdpPercap)
#between 2.5 and 3.5.
```

Problem 2

In the lecture, we discussed fitting of a linear model of mpg versus wt from the mtcars data and demonstrated evaluation of its out-of-sample performance with a k-fold cross validation. Repeat this analysis for a non-linear model mpg ~ a/wt + b, where a and b are model parameters and compare its performance with the linear model using an 8-fold cross validation. Let a=1 and b = 0 starting for nonlinear model

Determine mean RMSE for the linear model

```
#Create a column containing the result of training an lm model
mtcars_cv <- mtcars %>%
  crossv_kfold(k = 8) %>%
  mutate(model_lm = map(train, ~lm(mpg ~ wt, data = .)))
#Use map2_dbl() to extract the rmse for the 8 fold cross validation
mtcars_lm_mean_rmse <- mtcars_cv %$%
  map2_dbl(model_lm, test, rmse) %>%
  mean()

mtcars_lm_mean_rmse
```

```
## [1] 2.99776
```

Determine mean RMSE for nls

```
#map() does not work well with nls() so we manually create the training list
# from the result of the k-fold cross validation
train_nls <-list()
for (i in seq_along(length(mtcars_cv$train))) {
  idx <- mtcars_cv$train[[i]]$idx #extract the train indexes
  #For each fold, append the data associated to the train indexes to train_nls
  train_nls <- list.append(train_nls, mtcars_cv$train[[i]]$data[idx, ])
}

#Create a column containing the result of training an lm model
mtcars_cv <- mtcars_cv %>%
  mutate(train_nls = train_nls) %>%
  mutate(model_nls =
           map(.$train_nls, ~nls(mpg ~ a / wt + b, data = .,
                                 start = list(a = 1, b = 0))))
#Use map2_dbl() to extract the rmse for the 8 fold cross validation
mtcars_nls_mean_rmse <- mtcars_cv %$%
  map2_dbl(model_nls, test, rmse) %>%
  mean()

mtcars_nls_mean_rmse
```

```
## [1] 7.890702
```

Based on the results, the linear model is a much better fit due to the mean RMSE being roughly 1/4 that of the nls model