

Create a scatterplot showing population size on the y axis and year on the x axis. Add lines to show three different model fits for the data: a linear model, a LOESS model, and a cubic spline model.

```
model_colors <- brewer.pal(3,"Dark2")
#model_colors

one_country <- gapminder %>%
  filter(country == "New Zealand")
p <- ggplot(data = one_country, mapping = aes(x = year, y = pop))

ggplotly(p + geom_point() +
  geom_smooth(method = "lm",
    aes(color = "Linear Model", fill = "Linear Model")) +
  geom_smooth(method = "loess",
    aes(color = "LOESS", fill = "LOESS")) +
  geom_smooth(method = "lm", formula = y ~ splines::bs(x, df = 3),
    aes(color = "Cubic Spline", fill = "Cubic Spline")) +
  scale_color_manual(name = "Models", values = model_colors) +
  scale_fill_manual(name = "Models", values = model_colors) +
  theme(legend.position = "top") +
  labs(x = "Year", y = "Population", title = "New Zealand Population Growth"))

reg <- lm(data=one_country, year~pop)
#summary(reg)
#reg$coefficients[1]
#reg$coefficients[2]
pred.y <- reg$coefficients[1] + reg$coefficients[2]*one_country$pop
```

Using the gapminder data, looking at only data from the Americas, fit a linear regression to predict population as a function of year and country. Calculate predicted values for years ranging from 1950 to 2025, in intervals of five years. Include a 95% prediction interval. Plot your predictions for two countries of your choosing. In addition, create a plot of the residuals versus the fitted values.

```
# Data from the Americas only
smaller_gapminder <- gapminder %>%
  filter(continent == "Americas")

# Predict population as a function of year and country
Americas_pop <- lm(formula = pop ~ year * country,
  data = smaller_gapminder)
#summary(Americas_pop)

# Predict values for years ranging from 1950 to 2025 in intervals of 5 years
min_year <- 1950
max_year <- 2025

pred_df <- expand.grid(year = (seq(from = 1950,
  to = 2025,
  # intervals of 5 years, [(2016 - 1950)/5 + 1]
  length.out = 16)),
```

```

country = c( "Argentina","Cuba"))

#pred_df

pred_Americas_pop <- predict(object = Americas_pop,
                             newdata = pred_df,
                             interval = "predict") #95% prediction interval

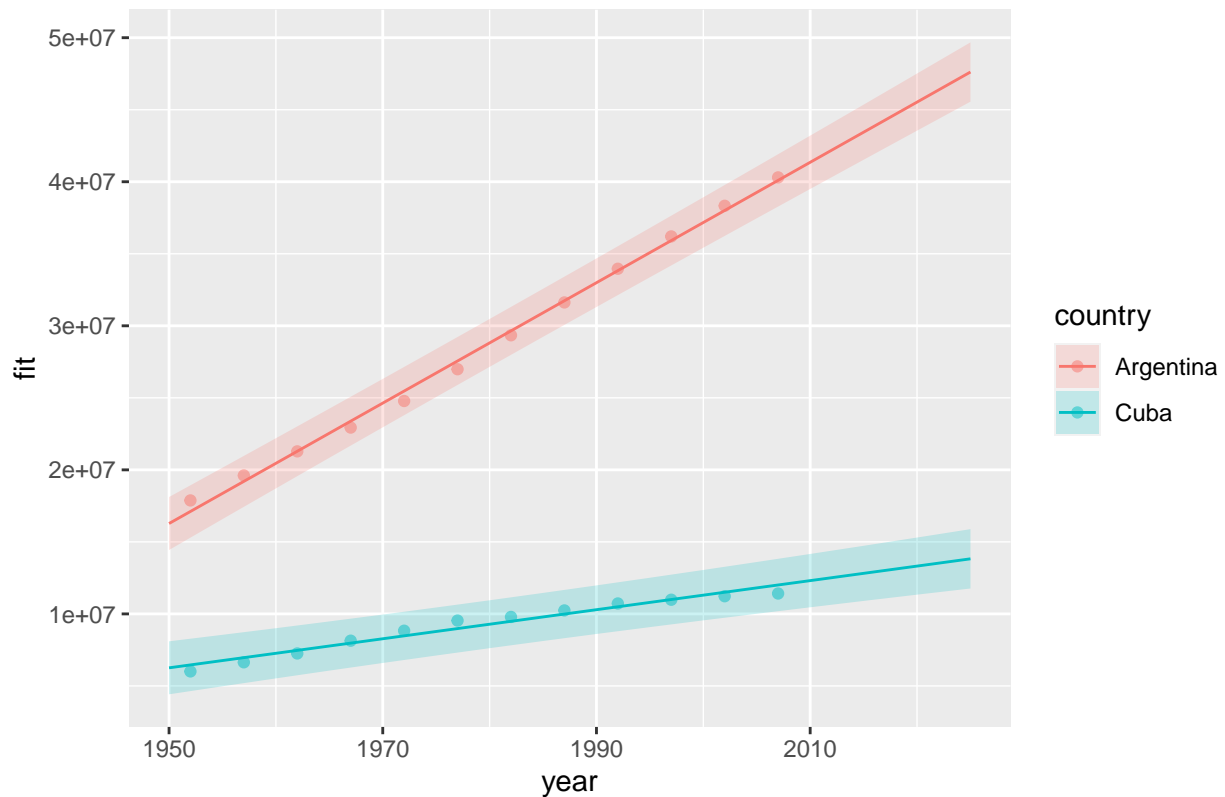
pred_df <- cbind(pred_df, pred_Americas_pop)

p <- ggplot(data = pred_df,
            aes(x = year,
                y = fit, ymin = lwr, ymax = upr,
                color = country,
                fill = country,
                group = country))

p + geom_point(data = subset(smaller_gapminder,
                             country %in% c("Cuba", "Argentina")),
              aes(x = year, y = pop,
                  color = country),
              alpha = 0.5,
              inherit.aes = FALSE) +
  geom_line() +
  geom_ribbon(alpha = 0.2, color = FALSE) +
  labs(title = "Argentina and Cuba projected population size from 1950 to 2025")

```

Argentina and Cuba projected population size from 1950 to 2025



```
# Data from Argentina and Cuba only
smaller_gapminder1 <- gapminder %>%
  filter(country %in% c( "Cuba", "Argentina"))
Americas_pop1 <- lm(formula = pop ~ year * country,
  data = smaller_gapminder1)

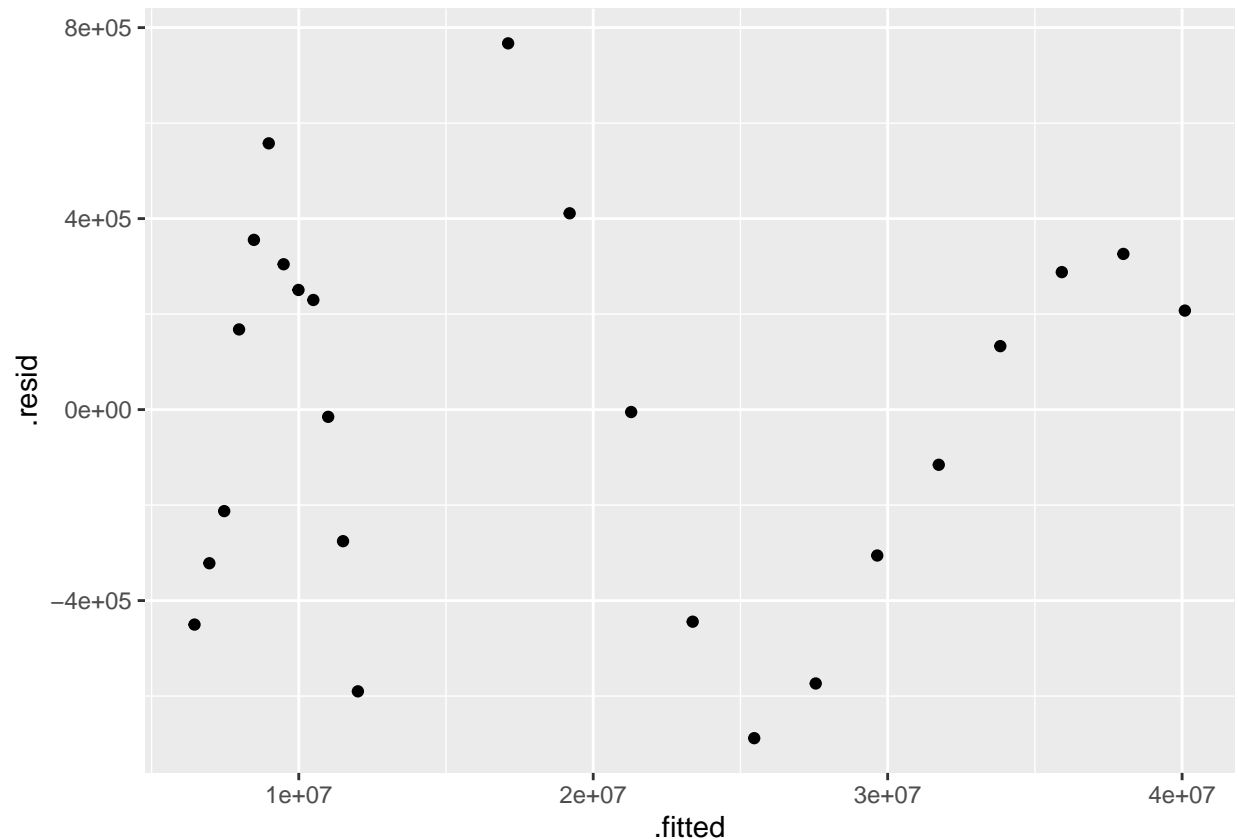
Americas_pop1_aug <- augment(Americas_pop1)
head(Americas_pop1_aug) %>% round_df()
```

```
## # A tibble: 6 x 9
##      pop year country      .fitted .resid .hat .sigma .cooksd .std.resid
##      <dbl> <dbl> <fct>      <dbl>  <dbl> <dbl>  <dbl>  <dbl>      <dbl>
## 1 17876956 1952 Argentina 17109890.  767066. 0.290 378998.  0.49      2.16
## 2 19610538 1957 Argentina 19199408.  411130. 0.22  419618.  0.09      1.11
## 3 21283783 1962 Argentina 21288926.   -5143. 0.17  433077.  0      -0.01
## 4 22934225 1967 Argentina 23378444. -444219. 0.13  419119.  0.05     -1.13
## 5 24779799 1972 Argentina 25467963. -688164. 0.1   399865.  0.08     -1.72
## 6 26983828 1977 Argentina 27557481. -573653. 0.09  410642.  0.05     -1.42
```

```
Americas_pop1_aug <- augment(Americas_pop1, data = smaller_gapminder1)
head(Americas_pop1_aug) %>% round_df()
```

```
## # A tibble: 6 x 12
##   country continent year lifeExp  pop gdpPercap .fitted .resid .hat .sigma
##   <fct>    <fct>    <dbl>  <dbl>  <dbl>    <dbl>  <dbl>  <dbl> <dbl>  <dbl>
## 1 Argenti~ Americas 1952   62.5 1.79e7   5911.  1.71e7  7.67e5 0.290 3.79e5
## 2 Argenti~ Americas 1957   64.4 1.96e7   6857.  1.92e7  4.11e5 0.22  4.20e5
## 3 Argenti~ Americas 1962   65.1 2.13e7   7133.  2.13e7 -5.14e3 0.17  4.33e5
## 4 Argenti~ Americas 1967   65.6 2.29e7   8053.  2.34e7 -4.44e5 0.13  4.19e5
## 5 Argenti~ Americas 1972   67.1 2.48e7   9443.  2.55e7 -6.88e5 0.1   4.00e5
## 6 Argenti~ Americas 1977   68.5 2.70e7  10079.  2.76e7 -5.74e5 0.09  4.11e5
## # ... with 2 more variables: .cooksd <dbl>, .std.resid <dbl>
```

```
p <- ggplot(data = Americas_pop1_aug,
mapping = aes(x = .fitted, y = .resid))
p + geom_point()
```



Using the gapminder data, use the nest function to fit a separate regression model for every country in the Americas, predicting population size as a function of year. Use geom\_pointrange() to display the slopes for each of these models, with error bars based on two standard errors.

```
out_le <- smaller_gapminder %>%
  group_by(country) %>%
  nest()

#out_le
fit_ols <- function(df) {
  lm(pop ~ year, data = df)
}

out_le <- smaller_gapminder %>%
  group_by(country) %>%
  nest() %>%
  mutate(model = map(data, fit_ols))

#out_le
fit_ols <- function(df) {
  lm(pop ~ year, data = df)
}

out_tidy <- smaller_gapminder %>%
  group_by(country) %>%
```

```

nest() %>%
mutate(model = map(data, fit_ols),
       tidied = map(model, tidy)) %>%
unnest(tidied, .drop = TRUE) %>%
filter(term %nin% "(Intercept)")

```

## Warning: The `'drop'` argument of `'unnest()'` is deprecated as of tidyr 1.0.0.  
## All list-columns are now preserved.

```

#out_tidy
p <- ggplot(data = out_tidy,
           mapping = aes(x = country, y = estimate,
                        ymin = estimate - 2*std.error,
                        ymax = estimate + 2*std.error,
                        group = country, color = country))

p + geom_pointrange(position = position_dodge(width = 1)) +
  scale_x_discrete(breaks = unique(smaller_gapminder$country)) +
  coord_flip() +
  theme(legend.position = "top") +
  labs(x = "Year", y = "Estimate", color = "Country")

```

