Assignment1_Mar25

Garrett Bullivant

2023-03-25

```
# load libraries
library(ggplot2)
library(tidyverse)
```

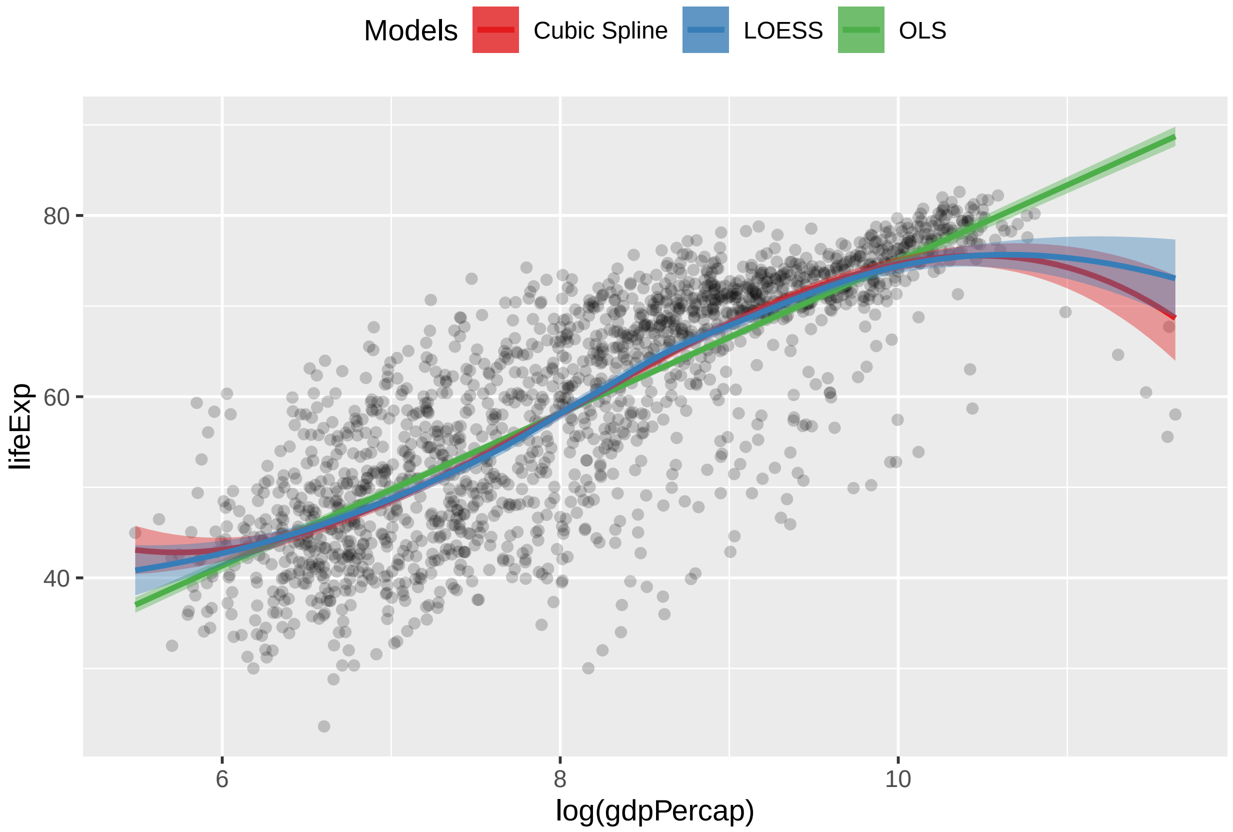
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(socviz)
library(gapminder)
library(ggrepel)

# build colour palette
model_colors <- RColorBrewer::brewer.pal(3, "Set1")
model_colors
```

```
## [1] "#E41A1C" "#377EB8" "#4DAF4A"
```

```
#
p0 <- ggplot(data = gapminder, mapping = aes(x = log(gdpPercap), y =
lifeExp))

p1 <- p0 + geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", aes(color = "OLS", fill = "OLS")) +
  geom_smooth(method = "lm", formula = y ~ splines::bs(x, df = 3),
              aes(color = "Cubic Spline", fill = "Cubic Spline")) +
  geom_smooth(method = "loess", aes(color = "LOESS", fill = "LOESS"))

# add a legend
p1 + scale_color_manual(name = "Models", values = model_colors) +
  scale_fill_manual(name = "Models", values = model_colors) +
  theme(legend.position = "top")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



```
# explore data
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

```
str(gapminder)
```

```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

```
# make our model object
out <- lm(formula = lifeExp ~ gdpPercap + pop + continent, data = gapminder)

# explore model
str(out)
```

```
## List of 13
##  $ coefficients : Named num [1:7] 4.78e+01 4.50e-04 6.57e-09 1.35e+01 8.19 ...
##   ..- attr(*, "names")= chr [1:7] "(Intercept)" "gdpPercap" "pop" "continentAmericas" ...
##  $ residuals    : Named num [1:1704] -27.6 -26.1 -24.5 -22.4 -20.3 ...
##   ..- attr(*, "names")= chr [1:1704] "1" "2" "3" "4" ...
##  $ effects      : Named num [1:1704] -2455.1 311.1 42.6 101.1 -17.2 ...
##   ..- attr(*, "names")= chr [1:1704] "(Intercept)" "gdpPercap" "pop" "continentAmericas" ...
##  $ rank         : int 7
##  $ fitted.values: Named num [1:1704] 56.4 56.4 56.5 56.5 56.4 ...
##   ..- attr(*, "names")= chr [1:1704] "1" "2" "3" "4" ...
##  $ assign       : int [1:7] 0 1 2 3 3 3 3
##  $ qr           :List of 5
##   ..$ qr   : num [1:1704, 1:7] -41.2795 0.0242 0.0242 0.0242 0.0242 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:1704] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:7] "(Intercept)" "gdpPercap" "pop" "continentAmericas" ...
##   .. ..- attr(*, "assign")= int [1:7] 0 1 2 3 3 3 3
##   .. ..- attr(*, "contrasts")=List of 1
##   .. .. ..$ continent: chr "contr.treatment"
##   ..$ qraux: num [1:7] 1.02 1.02 1 1.01 1.04 ...
##   ..$ pivot: int [1:7] 1 2 3 4 5 6 7
##   ..$ tol  : num 1e-07
##   ..$ rank : int 7
##   ..- attr(*, "class")= chr "qr"
##  $ df.residual  : int 1697
##  $ contrasts    :List of 1
##   ..$ continent: chr "contr.treatment"
##  $ xlevels      :List of 1
##   ..$ continent: chr [1:5] "Africa" "Americas" "Asia" "Europe" ...
##  $ call         : language lm(formula = lifeExp ~ gdpPercap + pop + continent, data = gapminder)
##  $ terms        :Classes 'terms', 'formula'  language lifeExp ~ gdpPercap + pop + continent
##   .. ..- attr(*, "variables")= language list(lifeExp, gdpPercap, pop, continent)
##   .. ..- attr(*, "factors")= int [1:4, 1:3] 0 1 0 0 0 0 1 0 0 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:4] "lifeExp" "gdpPercap" "pop" "continent"
##   .. .. .. ..$ : chr [1:3] "gdpPercap" "pop" "continent"
##   .. ..- attr(*, "term.labels")= chr [1:3] "gdpPercap" "pop" "continent"
##   .. ..- attr(*, "order")= int [1:3] 1 1 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(lifeExp, gdpPercap, pop, continent)
##   .. ..- attr(*, "dataClasses")= Named chr [1:4] "numeric" "numeric" "numeric" "factor"
##   .. .. ..- attr(*, "names")= chr [1:4] "lifeExp" "gdpPercap" "pop" "continent"
##  $ model        :'data.frame':   1704 obs. of  4 variables:
```

```
##   ..$ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##   ..$ gdpPercap: num [1:1704] 779 821 853 836 740 ...
##   ..$ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957
##   ..$ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##   ..- attr(*, "terms")=Classes ’terms’, ’formula’  language lifeExp ~ gdpPercap + pop + continent
##   .. .. ..- attr(*, "variables")= language list(lifeExp, gdpPercap, pop, continent)
##   .. .. ..- attr(*, "factors")= int [1:4, 1:3] 0 1 0 0 0 0 1 0 0 0 ...
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:4] "lifeExp" "gdpPercap" "pop" "continent"
##   .. .. .. .. ..$ : chr [1:3] "gdpPercap" "pop" "continent"
##   .. .. ..- attr(*, "term.labels")= chr [1:3] "gdpPercap" "pop" "continent"
##   .. .. ..- attr(*, "order")= int [1:3] 1 1 1
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. .. ..- attr(*, "predvars")= language list(lifeExp, gdpPercap, pop, continent)
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:4] "numeric" "numeric" "numeric" "factor"
##   .. .. .. ..- attr(*, "names")= chr [1:4] "lifeExp" "gdpPercap" "pop" "continent"
##  - attr(*, "class")= chr "lm"
```

```
summary(out)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap + pop + continent, data = gapminder)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.161  -4.486   0.297   5.110  25.175
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.781e+01  3.395e-01 140.819  < 2e-16 ***
## gdpPercap         4.495e-04  2.346e-05  19.158  < 2e-16 ***
## pop               6.570e-09  1.975e-09   3.326 0.000901 ***
## continentAmericas 1.348e+01  6.000e-01  22.458  < 2e-16 ***
## continentAsia     8.193e+00  5.712e-01  14.342  < 2e-16 ***
## continentEurope   1.747e+01  6.246e-01  27.973  < 2e-16 ***
## continentOceania  1.808e+01  1.782e+00  10.146  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 8.365 on 1697 degrees of freedom
## Multiple R-squared:  0.5821, Adjusted R-squared:  0.5806
## F-statistic: 393.9 on 6 and 1697 DF,  p-value: < 2.2e-16
```

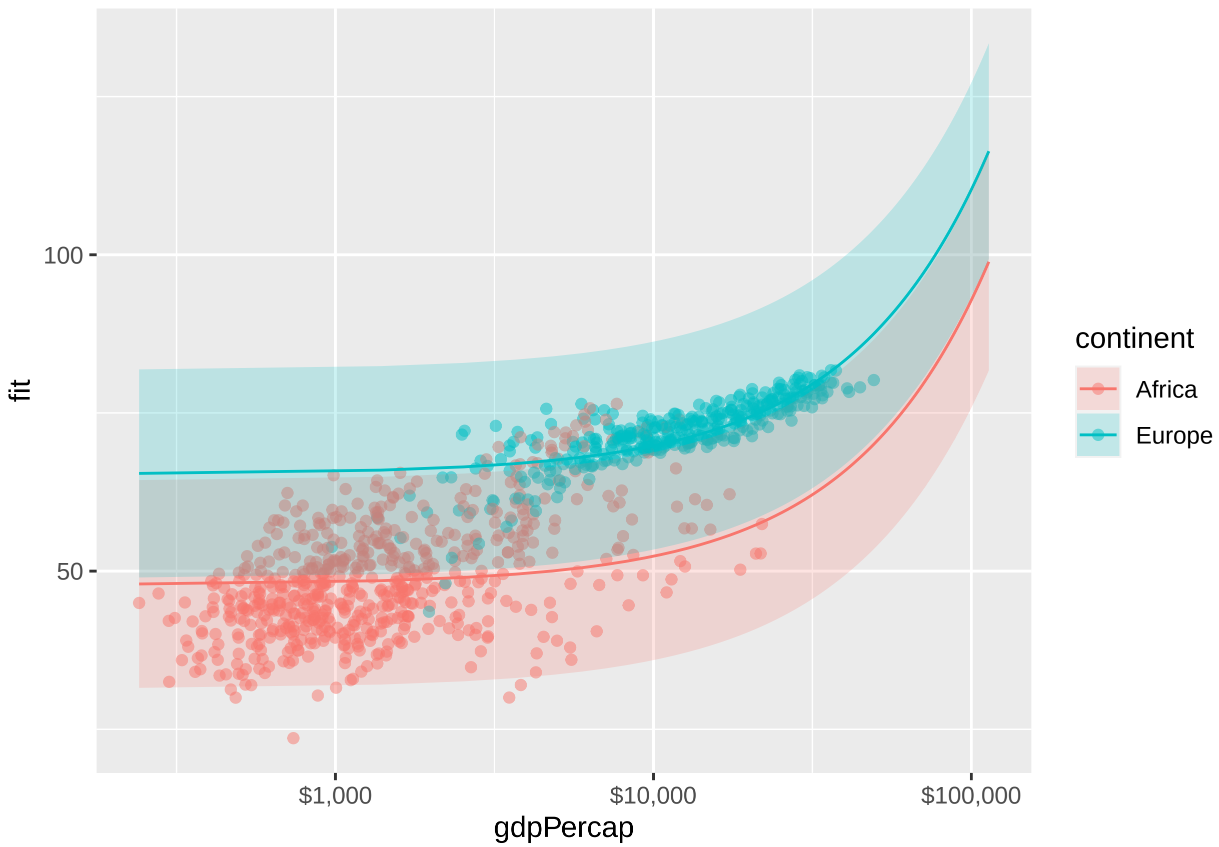
```
# make dataframe for predictions
min_gdp <- min(gapminder$gdpPercap)
max_gdp <- max(gapminder$gdpPercap)
med_pop <- median(gapminder$pop)

pred_df <- expand.grid(gdpPercap = (seq(from = min_gdp, to = max_gdp, length.out = 100)),
                       pop = med_pop,
                       continent = c("Africa", "Americas", "Asia", "Europe", "Oceania"))

# make predictions
pred_out <- predict(object = out, newdata = pred_df, interval = "predict")
head(pred_out)
```

```
##        fit      lwr      upr
## 1 47.96863 31.54775 64.38951
## 2 48.48298 32.06231 64.90365
## 3 48.99733 32.57670 65.41797
## 4 49.51169 33.09092 65.93245
## 5 50.02604 33.60497 66.44711
## 6 50.54039 34.11885 66.96193
```

```
#bind predictions and data
pred_df <- cbind(pred_df, pred_out)

head(pred_df)
```

```
##   gdpPercap     pop continent      fit      lwr      upr
## 1  241.1659 7023596    Africa 47.96863 31.54775 64.38951
## 2 1385.4282 7023596    Africa 48.48298 32.06231 64.90365
## 3 2529.6905 7023596    Africa 48.99733 32.57670 65.41797
## 4 3673.9528 7023596    Africa 49.51169 33.09092 65.93245
## 5 4818.2150 7023596    Africa 50.02604 33.60497 66.44711
## 6 5962.4773 7023596    Africa 50.54039 34.11885 66.96193
```

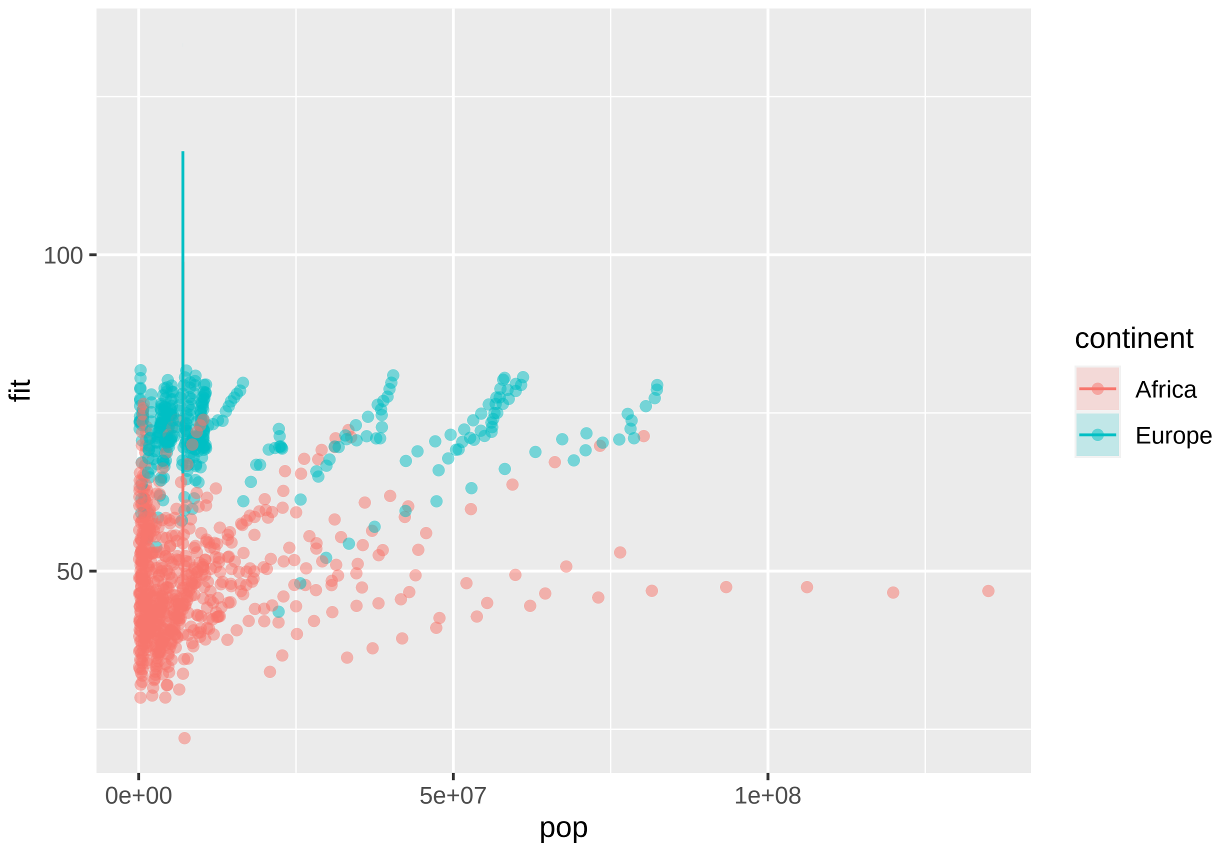
```
p <- ggplot(data = subset(pred_df, continent %in% c("Europe", "Africa")),
            aes(x = gdpPercap, y = fit, ymin = lwr, ymax = upr , color = continent, fill =
                  continent, group = continent))

p + geom_point(data = subset(gapminder, continent %in% c("Europe", "Africa")),
               aes(x = gdpPercap, y = lifeExp, color = continent),
               alpha = 0.5,
               inherit.aes = FALSE) +
  geom_line() +
  geom_ribbon(alpha = 0.2, color = FALSE) +
  scale_x_log10(labels = scales::dollar)
```



Activity

Research question - How is life expectancy influenced by the population of a given country?

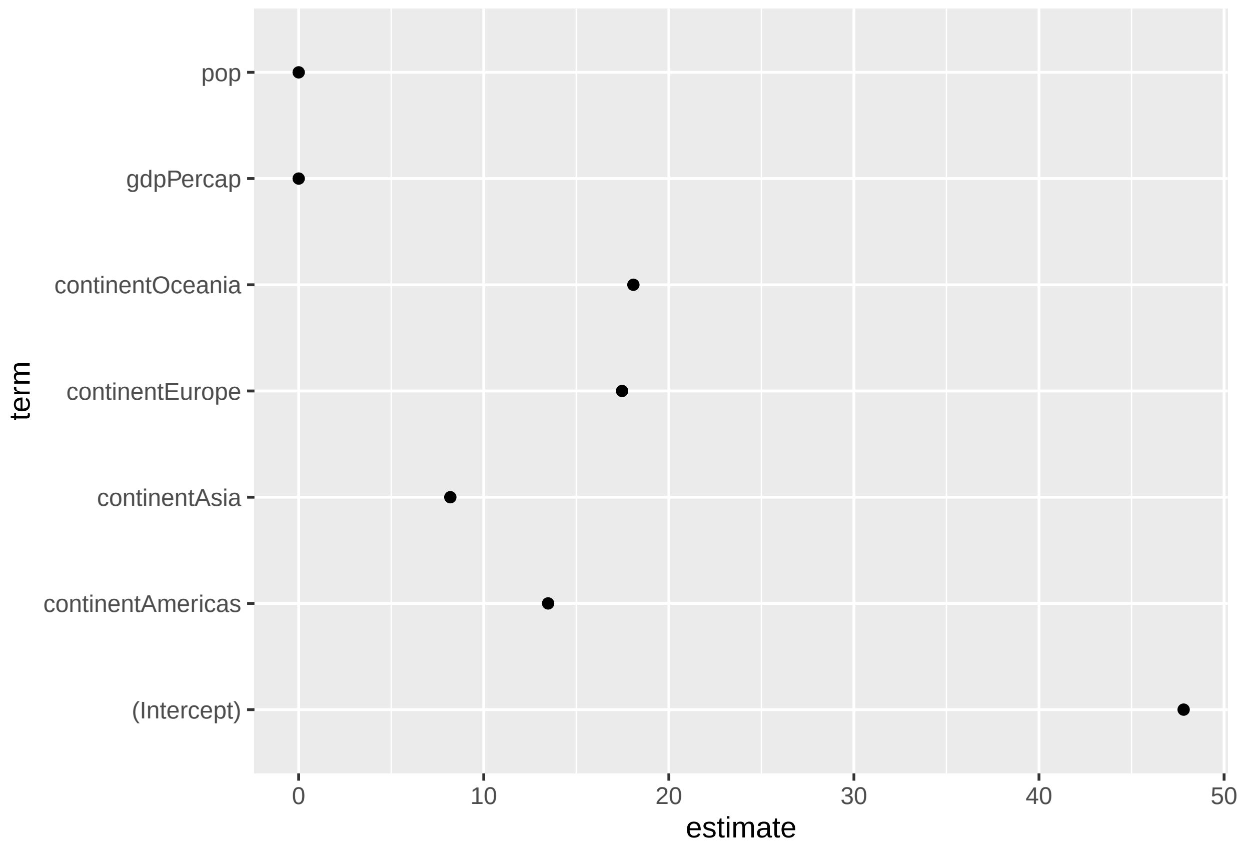
```
p <- ggplot(data = subset(pred_df, continent %in% c("Europe", "Africa")),
            aes(x = pop, y = fit, ymin = lwr, ymax = upr , color = continent, fill =
                  continent, group = continent))

p + geom_point(data = subset(gapminder, continent %in% c("Europe", "Africa")),
               aes(x = pop, y = lifeExp, color = continent),
               alpha = 0.5,
               inherit.aes = FALSE) +
  geom_line() +
  geom_ribbon(alpha = 0.2, color = FALSE)
```



```
# load broom library
#install.packages("broom")
library(broom)

# use tidy to gather component level stats
out_conf <- tidy(out)
out_conf |> round_df()
```

```
## # A tibble: 7 x 5
##   term                estimate std.error statistic p.value
##   <chr>                  <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)            47.8       0.34     141.         0
## 2 gdpPercap               0         0         19.2        0
## 3 pop                     0         0          3.33       0
## 4 continentAmericas      13.5       0.6       22.5        0
## 5 continentAsia           8.19      0.57      14.3        0
## 6 continentEurope        17.5       0.62      28.0        0
## 7 continentOceania       18.1       1.78      10.2        0
```

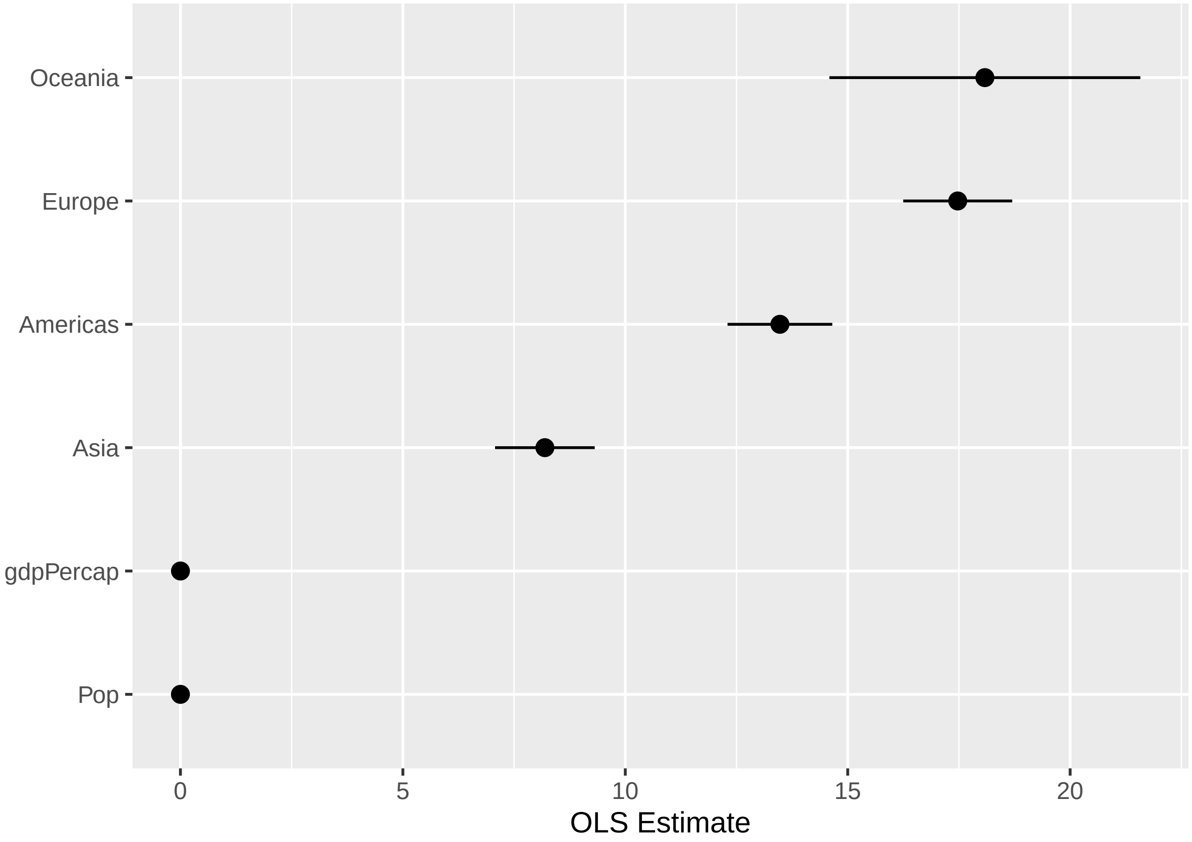
```
# plot component level stats
p <- ggplot(out_conf, mapping = aes(x = term, y = estimate))
p + geom_point() + coord_flip()
```



```r
# use confint to produce confidence intervals
out_conf <- tidy(out, conf.int = TRUE)
out_conf %>% round_df()
```

```
## # A tibble: 7 x 7
##   term              estimate std.error statistic p.value conf.low conf.high
##   <chr>                <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)          47.8       0.34    141.         0    47.2      48.5
## 2 gdpPercap             0         0         19.2       0     0         0
## 3 pop                   0         0          3.33      0     0         0
## 4 continentAmericas    13.5       0.6       22.5       0    12.3      14.6
## 5 continentAsia         8.19      0.57      14.3       0     7.07      9.31
## 6 continentEurope      17.5       0.62      28.0       0    16.2      18.7
## 7 continentOceania     18.1       1.78      10.2       0    14.6      21.6
```

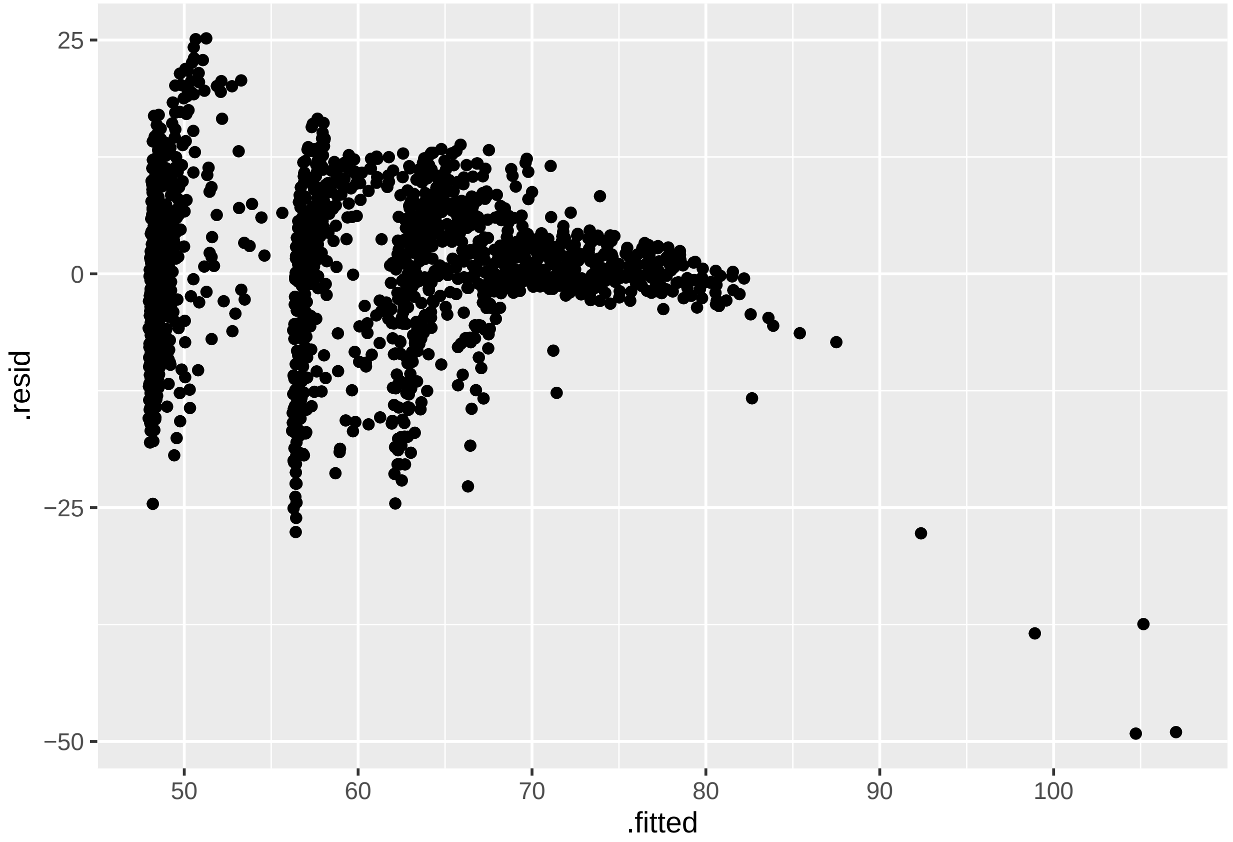
```r
# clean up our visualization with tidy
out_conf <- subset(out_conf, term %nin% "(Intercept)")
out_conf$nicelabs <- prefix_strip(out_conf$term, "continent")

# include confidence intervals
p <- ggplot(out_conf, mapping = aes(x = reorder(nicelabs, estimate), y = estimate, ymin = conf.low, yma
p + geom_pointrange() + coord_flip() + labs(x = "", y = "OLS Estimate")
```



```
# augment adds observation level statistics
out_aug <- augment(out)
head(out_aug) |> round_df()
```

```
## # A tibble: 6 x 10
##   lifeExp gdpPercap      pop conti~1 .fitted .resid  .hat .sigma .cooksd .std.~2
##     <dbl>     <dbl>    <dbl> <fct>     <dbl>  <dbl> <dbl>  <dbl>   <dbl>   <dbl>
## 1    28.8      779.  8425333 Asia       56.4  -27.6     0   8.34    0.01   -3.31
## 2    30.3      821.  9240934 Asia       56.4  -26.1     0   8.34    0      -3.13
## 3    32        853. 10267083 Asia       56.5  -24.5     0   8.35    0      -2.93
## 4    34.0      836. 11537966 Asia       56.5  -22.4     0   8.35    0      -2.69
## 5    36.1      740. 13079460 Asia       56.4  -20.3     0   8.35    0      -2.44
## 6    38.4      786. 14880372 Asia       56.5  -18.0     0   8.36    0      -2.16
## # ... with abbreviated variable names 1: continent, 2: .std.resid
```

```
# we can now plot observation level stats - residuals vs fitted values
p <- ggplot(data = out_aug, mapping = aes(x = .fitted, y = .resid))
p + geom_point()
```



```r
# finally we can use glance to gather model level statistics

glance(out) |> round_df()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squ~1 sigma stati~2 p.value    df logLik    AIC    BIC devia~3
##       <dbl>       <dbl> <dbl>   <dbl>   <dbl> <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
## 1      0.58        0.58  8.37    394.       0     6 -6034. 12084. 12127. 118754.
## # ... with 2 more variables: df.residual <dbl>, nobs <dbl>, and abbreviated
## #   variable names 1: adj.r.squared, 2: statistic, 3: deviance
```

```r
# Using broom for grpuped analysis
eu77 <- gapminder |> filter(continent == "Europe", year == 1977)
fit <- lm(lifeExp ~ log(gdpPercap), data = eu77)
summary(fit)
```

```
##
## Call:
## lm(formula = lifeExp ~ log(gdpPercap), data = eu77)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4956 -1.0306  0.0935  1.1755  3.7125
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       29.489      7.161   4.118 0.000306 ***
## log(gdpPercap)     4.488      0.756   5.936 2.17e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.114 on 28 degrees of freedom
## Multiple R-squared:  0.5572, Adjusted R-squared:  0.5414
## F-statistic: 35.24 on 1 and 28 DF,  p-value: 2.173e-06
```

```
# nesting data
out_le <- gapminder |>
  group_by(continent, year) |>
  nest()

out_le
```

```
## # A tibble: 60 x 3
## # Groups:   continent, year [60]
##    continent  year data
##    <fct>     <int> <list>
##  1 Asia       1952 <tibble [33 x 4]>
##  2 Asia       1957 <tibble [33 x 4]>
##  3 Asia       1962 <tibble [33 x 4]>
##  4 Asia       1967 <tibble [33 x 4]>
##  5 Asia       1972 <tibble [33 x 4]>
##  6 Asia       1977 <tibble [33 x 4]>
##  7 Asia       1982 <tibble [33 x 4]>
##  8 Asia       1987 <tibble [33 x 4]>
##  9 Asia       1992 <tibble [33 x 4]>
## 10 Asia       1997 <tibble [33 x 4]>
## # ... with 50 more rows
```

```
# we can now easily pick out data by continent and year
out_le |> filter(continent == "Europe" & year == 1977) |>
  unnest()
```

```
## Warning: `cols` is now required when using unnest().
## Please use `cols = c(data)`
```

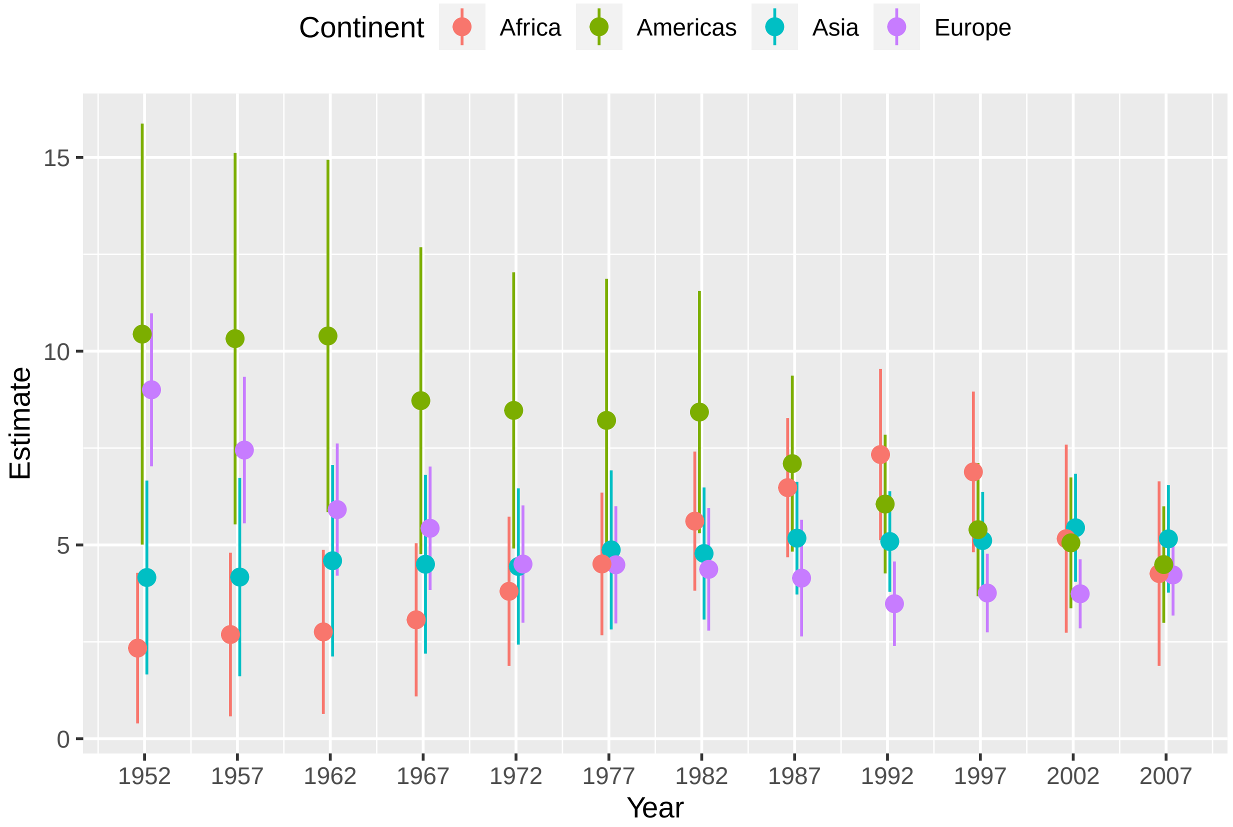
```
## # A tibble: 30 x 6
## # Groups:   continent, year [1]
##    continent  year country                lifeExp      pop gdpPercap
##    <fct>     <int> <fct>                    <dbl>    <int>     <dbl>
##  1 Europe     1977 Albania                   68.9  2509048     3533.
##  2 Europe     1977 Austria                   72.2  7568430    19749.
##  3 Europe     1977 Belgium                   72.8  9821800    19118.
##  4 Europe     1977 Bosnia and Herzegovina    69.9  4086000     3528.
##  5 Europe     1977 Bulgaria                  70.8  8797022     7612.
##  6 Europe     1977 Croatia                   70.6  4318673    11305.
##  7 Europe     1977 Czech Republic            70.7 10161915    14800.
##  8 Europe     1977 Denmark                   74.7  5088419    20423.
##  9 Europe     1977 Finland                   72.5  4738902    15605.
## 10 Europe     1977 France                     73.8 53165019    18293.
## # ... with 20 more rows
```

```
# create a function that fits our model to a dataframe
fit_ols <- function(df) {lm(lifeExp ~ log(gdpPercap), data = df)}

# apply model to each row
out_le <- gapminder |>
  group_by(continent, year) |>
  nest() |>
  mutate(model = map(data, fit_ols))
out_le
```

```
## # A tibble: 60 x 4
## # Groups:   continent, year [60]
##    continent  year data               model
##    <fct>     <int> <list>             <list>
##  1 Asia       1952 <tibble [33 x 4]> <lm>
##  2 Asia       1957 <tibble [33 x 4]> <lm>
##  3 Asia       1962 <tibble [33 x 4]> <lm>
##  4 Asia       1967 <tibble [33 x 4]> <lm>
##  5 Asia       1972 <tibble [33 x 4]> <lm>
##  6 Asia       1977 <tibble [33 x 4]> <lm>
##  7 Asia       1982 <tibble [33 x 4]> <lm>
##  8 Asia       1987 <tibble [33 x 4]> <lm>
##  9 Asia       1992 <tibble [33 x 4]> <lm>
## 10 Asia       1997 <tibble [33 x 4]> <lm>
## # ... with 50 more rows
```

```
# tidy up our data
out_tidy <- gapminder |> group_by(continent, year) |> nest() |>
  mutate(model = map(data, fit_ols), tidied = map(model, tidy)) |>
  unnest(tidied) |> filter(term %nin% "(Intercept)" & continent %nin% "Oceania")
```

```
p <- ggplot(data = out_tidy, mapping = aes(x = year, y = estimate, ymin = estimate - 2*std.error, ymax

p + geom_pointrange(position = position_dodge(width = 1)) +
  scale_x_continuous(breaks = unique(gapminder$year)) +
  theme(legend.position = "top") +
  labs(x = "Year", y = "Estimate", color = "Continent")
```



Activity How did this diagram of the Brooks use rational, moral, and emotional appeal to make a case to its audiences?

Using the diagrams to show pictures of people on the boat shows how horrible the conditions are instead of just using people/boats. It uses emotional persuasian to pull on empathy. Furthermore, I think this also uses moral appeal - it appeals to the audiences moral values.