432 Class 10

https://thomaselove.github.io/432-2025

2025-02-13

## Our Agenda

* World Happiness Report data ingest and cleanup
* Linear Regression with lm() and with ols()
  + Fitting Five Models with both lm() and ols()
  + Single and Multiple Imputation strategies
  + Spearman’s ; incorporating non-linear terms
  + Using “best subsets” searches to prune models
  + Evaluating / Displaying Model Fit in a Training Sample
  + Evaluating Model Predictions in a Test Sample

## Our R Setup

knitr::opts\_chunk$set(comment = NA)  
  
library(janitor); library(naniar)  
library(broom); library(gt); library(patchwork)  
library(car) ## variance inflation factor, boxCox plot  
library(caret) ## for confusion matrices  
library(cobalt) ## new today: to split factor into indicator variables  
library(cutpointr) ## new today: optimizing cutpoints  
library(mice) ## supporting multiple and simple imputation  
library(mosaic) ## for df\_stats() and favstats(), mostly  
library(olsrr) ## best subsets search for linear regression  
library(readxl) ## read in data from an Excel file  
library(rms) ## also loads Hmisc   
library(easystats); library(tidyverse)  
  
theme\_set(theme\_bw())

## Ingest the happy Data

happy <- read\_xlsx("c10/data/happy.xlsx", na = c("NA", "")) |>  
 janitor::clean\_names() |>  
 mutate(across(where(is.character), as\_factor),   
 iso3 = as.character(iso3),  
 country = as.character(country))  
  
dim(happy)

[1] 138 14

happy |> head()

# A tibble: 6 × 14  
 iso3 country ladder log\_gdp social life\_exp freedom generosity corruption  
 <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 AFG Afghanistan 1.45 NA 0.368 55.2 0.228 NA 0.738  
2 ALB Albania 5.44 9.69 0.691 69.2 0.872 0.0679 0.855  
3 ARG Argentina 6.39 9.99 0.892 67.3 0.832 -0.129 0.846  
4 ARM Armenia 5.68 9.73 0.819 68.2 0.819 -0.179 0.681  
5 AUS Australia 7.02 10.8 0.896 71.2 0.876 0.187 0.482  
6 AUT Austria 6.64 10.9 0.874 71.4 0.874 0.209 0.529  
# ℹ 5 more variables: pos\_affect <dbl>, neg\_affect <dbl>, region <fct>,  
# temp\_c <dbl>, pop\_dens <dbl>

## The happy data (1/3)

The data describe 14 characteristics of 138 countries included in the World Happiness Report 2024, much of which come from the Gallup World Poll (GWP).

| Variable | Description |
| --- | --- |
| iso3 | ISO-alpha3 code from [the United Nations](https://unstats.un.org/unsd/methodology/m49/) |
| country | Name as listed in World Happiness Report 2024 |
| ladder | National average response to “Please imagine a ladder, with steps numbered from 0 at the bottom to 10 at the top. The top of the ladder represents the best possible life for you and the bottom of the ladder represents the worst possible life for you. On which step of the ladder would you say you personally feel you stand at this time?” |
| log\_gdp | Natural log(GDP per capita) in purchasing power parity (PPP) at constant 2017 international dollar prices, from World Development Indicators |

## The happy data (2/3)

| Variable | Description |
| --- | --- |
| social | Social support = national average of 1/0 response to “If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?” |
| life\_exp | Nation’s healthy life expectancy at birth, extracted from the World Health Organization’s (WHO) Global Health Observatory data repository |
| freedom | National average of 1/0 responses to “Are you satisfied or dissatisfied with your freedom to choose what you do with your life?” |
| generosity | Residual of regressing national average 1/0 response to “Have you donated money to a charity in the past month?” on GDP per capita. |
| corruption | National average of 1/0 responses to two questions: “Is corruption widespread throughout the government or not” and “Is corruption widespread within businesses or not?” |

## The happy data (3/3)

| Variable | Description |
| --- | --- |
| pos\_affect | National average of 1/0 responses to three positive affect measures: “Did you smile or laugh a lot yesterday?”, and “Did you experience enjoyment during a lot of the day yesterday?”, “Did you learn or do something interesting yesterday?” |
| neg\_affect | National average of 1/0 responses to three negative affect measures: “Did you experience worry during a lot of the day yesterday?”, “Did you experience sadness …”, “Did you experience anger…” |
| region | United Nations Geoscheme (Continent), via [Wikipedia](https://en.wikipedia.org/wiki/List_of_countries_and_territories_by_the_United_Nations_geoscheme) |
| temp\_c | Mean yearly temperature (Celsius, 1991-2020) from [Wikipedia](https://en.wikipedia.org/wiki/List_of_countries_by_average_yearly_temperature) |
| pop\_dens | Population per square mile from [Wikipedia](https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population_density) |

* More details at [Statistical Appendix for Happiness Report](https://happiness-report.s3.amazonaws.com/2024/Ch2+Appendix.pdf)

# Data Management

## One region has only 2 countries…

We have five continents listed under region, but only 2 countries (Australia and New Zealand) located in Oceania. Let’s collapse that factor into the smallest other category.

happy <- happy |>   
 mutate(region4 = fct\_lump\_n(region, 3, other\_level = "Other"))  
  
happy |> tabyl(region4, region) |> gt() |> tab\_options(table.font.size = 24)

| region4 | Asia | Europe | Americas | Oceania | Africa |
| --- | --- | --- | --- | --- | --- |
| Asia | 40 | 0 | 0 | 0 | 0 |
| Europe | 0 | 39 | 0 | 0 | 0 |
| Africa | 0 | 0 | 0 | 0 | 37 |
| Other | 0 | 0 | 20 | 2 | 0 |

## Create two more categorical variables

We’re going to use two categories to represent temp\_c and three to represent the information in pop\_dens[[1]](#footnote-32).

happy <- happy |>   
 mutate(ftemp\_c = categorize(temp\_c, split = "median",   
 labels = c("cool", "warm"))) |>  
 mutate(fpop\_dens = categorize(pop\_dens, split = "quantile", n\_groups = 3,   
 labels = c("low", "med", "high")))  
  
happy |> tabyl(ftemp\_c, fpop\_dens) |>   
 adorn\_totals(where = c("row", "col")) |> adorn\_title()

fpop\_dens   
 ftemp\_c low med high Total  
 cool 24 25 20 69  
 warm 22 19 28 69  
 Total 46 44 48 138

## Sanity Checks

df\_stats(temp\_c ~ ftemp\_c, data = happy) |> gt() |>   
 tab\_options(table.font.size = 20) |>   
 fmt\_number(columns = mean:sd, decimals = 2) |>  
 opt\_stylize(style = 3, color = "blue")

| response | ftemp\_c | min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| temp\_c | cool | -4.03 | 8.60 | 10.49 | 13.07 | 20.99 | 11.01 | 5.68 | 69 | 0 |
| temp\_c | warm | 21.31 | 23.65 | 25.20 | 26.80 | 30.40 | 25.24 | 2.12 | 69 | 0 |

df\_stats(pop\_dens ~ fpop\_dens, data = happy) |> gt() |>   
 tab\_options(table.font.size = 20) |>   
 fmt\_number(columns = mean:sd, decimals = 2) |>  
 opt\_stylize(style = 3, color = "blue")

| response | fpop\_dens | min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pop\_dens | low | 5.7 | 30.0 | 58 | 105.25 | 140 | 64.70 | 42.72 | 46 | 0 |
| pop\_dens | med | 150.0 | 187.5 | 220 | 262.50 | 300 | 223.41 | 45.70 | 44 | 0 |
| pop\_dens | high | 310.0 | 380.0 | 635 | 1027.50 | 21400 | 1,363.54 | 3,102.89 | 48 | 0 |

## Data Codebook

data\_codebook(happy |> select(-iso3, -country))

select(happy, -iso3, -country) (138 rows and 15 variables, 15 shown)  
  
ID | Name | Type | Missings | Values | N  
---+------------+-------------+----------+---------------+-----------  
1 | ladder | numeric | 0 (0.0%) | [1.45, 7.7] | 138  
---+------------+-------------+----------+---------------+-----------  
2 | log\_gdp | numeric | 9 (6.5%) | [7.08, 11.68] | 129  
---+------------+-------------+----------+---------------+-----------  
3 | social | numeric | 0 (0.0%) | [0.37, 0.98] | 138  
---+------------+-------------+----------+---------------+-----------  
4 | life\_exp | numeric | 3 (2.2%) | [52.2, 74.6] | 135  
---+------------+-------------+----------+---------------+-----------  
5 | freedom | numeric | 2 (1.4%) | [0.23, 0.96] | 136  
---+------------+-------------+----------+---------------+-----------  
6 | generosity | numeric | 9 (6.5%) | [-0.27, 0.59] | 129  
---+------------+-------------+----------+---------------+-----------  
7 | corruption | numeric | 7 (5.1%) | [0.15, 0.95] | 131  
---+------------+-------------+----------+---------------+-----------  
8 | pos\_affect | numeric | 0 (0.0%) | [0.26, 0.84] | 138  
---+------------+-------------+----------+---------------+-----------  
9 | neg\_affect | numeric | 0 (0.0%) | [0.11, 0.52] | 138  
---+------------+-------------+----------+---------------+-----------  
10 | region | categorical | 0 (0.0%) | Asia | 40 (29.0%)  
 | | | | Europe | 39 (28.3%)  
 | | | | Americas | 20 (14.5%)  
 | | | | Oceania | 2 ( 1.4%)  
 | | | | Africa | 37 (26.8%)  
---+------------+-------------+----------+---------------+-----------  
11 | temp\_c | numeric | 0 (0.0%) | [-4.03, 30.4] | 138  
---+------------+-------------+----------+---------------+-----------  
12 | pop\_dens | numeric | 0 (0.0%) | [5.7, 21400] | 138  
---+------------+-------------+----------+---------------+-----------  
13 | region4 | categorical | 0 (0.0%) | Asia | 40 (29.0%)  
 | | | | Europe | 39 (28.3%)  
 | | | | Africa | 37 (26.8%)  
 | | | | Other | 22 (15.9%)  
---+------------+-------------+----------+---------------+-----------  
14 | ftemp\_c | categorical | 0 (0.0%) | cool | 69 (50.0%)  
 | | | | warm | 69 (50.0%)  
---+------------+-------------+----------+---------------+-----------  
15 | fpop\_dens | categorical | 0 (0.0%) | low | 46 (33.3%)  
 | | | | med | 44 (31.9%)  
 | | | | high | 48 (34.8%)  
---------------------------------------------------------------------

## Modeling Objective for Today

Predict ladder using some combination of these 11 predictors:

* log\_gdp, social, life\_exp, freedom, generosity,
* corruption, pos\_affect, neg\_affect, ftemp\_c,
* fpop\_dens, region4

We have complete data on ladder for all 138 countries.

n\_miss(happy$ladder)

[1] 0

## Check Variable Types

* Is each variable we’ll use either quantitative (<dbl> or <int>) or a factor (<fct>)?

glimpse(happy)

Rows: 138  
Columns: 17  
$ iso3 <chr> "AFG", "ALB", "ARG", "ARM", "AUS", "AUT", "AZE", "BHR", "BG…  
$ country <chr> "Afghanistan", "Albania", "Argentina", "Armenia", "Australi…  
$ ladder <dbl> 1.445909, 5.444691, 6.393229, 5.679090, 7.024582, 6.635664,…  
$ log\_gdp <dbl> NA, 9.688706, 9.993596, 9.729613, 10.846434, 10.930412, 9.6…  
$ social <dbl> 0.3684781, 0.6907526, 0.8921175, 0.8193378, 0.8964601, 0.87…  
$ life\_exp <dbl> 55.2, 69.2, 67.3, 68.2, 71.2, 71.4, 64.1, 65.6, 64.8, 71.2,…  
$ freedom <dbl> 0.2283012, 0.8715455, 0.8316838, 0.8193763, 0.8757688, 0.87…  
$ generosity <dbl> NA, 0.067885302, -0.129060909, -0.179444075, 0.187309042, 0…  
$ corruption <dbl> 0.7384709, 0.8554251, 0.8460935, 0.6807089, 0.4815805, 0.52…  
$ pos\_affect <dbl> 0.2605132, 0.5973493, 0.7201222, 0.5747167, 0.7310531, 0.71…  
$ neg\_affect <dbl> 0.4601669, 0.3142271, 0.3011622, 0.4226305, 0.2481628, 0.23…  
$ region <fct> Asia, Europe, Americas, Asia, Oceania, Europe, Asia, Asia, …  
$ temp\_c <dbl> 13.04, 12.44, 16.30, 7.59, 22.05, 7.44, 12.96, 27.69, 25.71…  
$ pop\_dens <dbl> 170.0, 260.0, 41.0, 240.0, 8.8, 280.0, 310.0, 4900.0, 3020.…  
$ region4 <fct> Asia, Europe, Other, Asia, Other, Europe, Asia, Asia, Asia,…  
$ ftemp\_c <fct> cool, cool, cool, cool, warm, cool, cool, warm, warm, cool,…  
$ fpop\_dens <fct> med, med, low, med, low, med, high, high, high, high, high,…

## Missing Data?

n\_miss(happy)

[1] 30

miss\_var\_summary(happy) |> filter(n\_miss > 0)

# A tibble: 5 × 3  
 variable n\_miss pct\_miss  
 <chr> <int> <num>  
1 log\_gdp 9 6.52  
2 generosity 9 6.52  
3 corruption 7 5.07  
4 life\_exp 3 2.17  
5 freedom 2 1.45

miss\_case\_table(happy)

# A tibble: 4 × 3  
 n\_miss\_in\_case n\_cases pct\_cases  
 <int> <int> <dbl>  
1 0 120 87.0   
2 1 8 5.80  
3 2 8 5.80  
4 3 2 1.45

## What’s the mode of our outcome?

happy |> tabyl(ladder) |>   
 adorn\_pct\_formatting() |> arrange(desc(n)) |> head(5)

ladder n percent  
 1.445909 1 0.7%  
 3.272092 1 0.7%  
 3.331648 1 0.7%  
 3.383398 1 0.7%  
 3.466578 1 0.7%

It appears we have nothing but unique values in our outcome.

identical(nrow(happy), n\_distinct(happy$ladder))

[1] TRUE

## Summarizing our outcome

mosaic::favstats(happy$ladder)

min Q1 median Q3 max mean sd n missing  
 1.445909 4.679697 5.86269 6.486904 7.698929 5.620811 1.139478 138 0

Hmisc::describe(happy$ladder)

happy$ladder   
 n missing distinct Info Mean pMedian Gmd .05   
 138 0 138 1 5.621 5.67 1.281 3.586   
 .10 .25 .50 .75 .90 .95   
 4.107 4.680 5.863 6.487 6.947 7.135   
  
lowest : 1.44591 3.27209 3.33165 3.3834 3.46658  
highest: 7.25479 7.38407 7.50419 7.56161 7.69893

* The ladder data are available for all 138 countries, with 138 distinct values.
* The mean ladder score is 5.62, with standard deviation 1.14 points.

## Which countries have the outlying ladder values?

* From our summary on the last slide, the range of ladder values goes from 1.45 to 7.70 points.

slice(happy, which.max(ladder)) |> select(iso3, country, ladder)

# A tibble: 1 × 3  
 iso3 country ladder  
 <chr> <chr> <dbl>  
1 FIN Finland 7.70

slice(happy, which.min(ladder)) |> select(iso3, country, ladder)

# A tibble: 1 × 3  
 iso3 country ladder  
 <chr> <chr> <dbl>  
1 AFG Afghanistan 1.45

## Single Imputation via mice

We’ll build a single imputation model, assuming MAR for the missing data, to help us fit our models.

happy\_si <- mice(happy, m = 1, seed = 43201, print = FALSE) |>  
 complete() |>  
 tibble()

Warning: Number of logged events: 47

dim(happy\_si)

[1] 138 17

prop\_miss\_case(happy\_si)

[1] 0

* Later, we’ll demonstrate both a mice-based method for multiple imputation, and another method using rms tools.

## Should we partition?

With just 138 countries available, we probably shouldn’t, but we will anyway here, so we can demonstrate some ideas.

1. Check that we have a unique iso3 code for each country?

identical(n\_distinct(happy\_si$iso3), nrow(happy\_si))

[1] TRUE

## Partitioning happy\_si

1. Partition happy\_si into samples of 80% training, 20% testing, using the data\_partition() tool from the **easystats** framework.

part\_si <- data\_partition(happy\_si, proportion = 0.8, seed = 43202)  
happy\_si\_train <- part\_si$p\_0.8  
happy\_si\_test <- part\_si$test  
  
dim(happy\_si\_train)

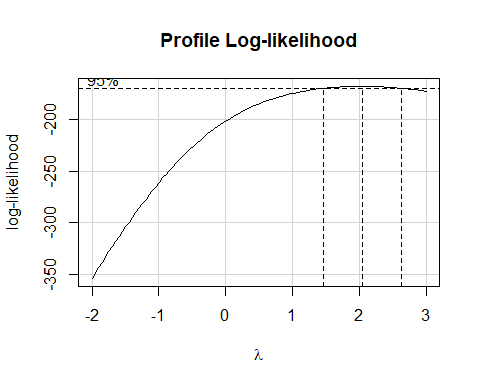
[1] 110 18

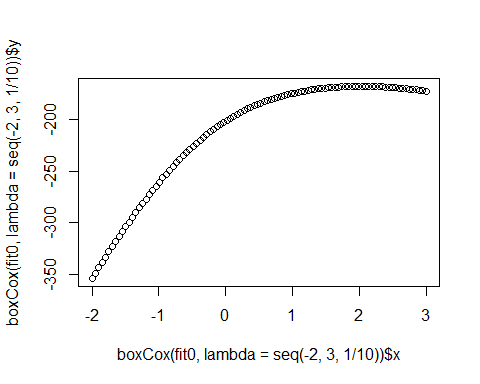
dim(happy\_si\_test)

[1] 28 18

## Consider outcome transformation?

fit0 <- lm(ladder ~ log\_gdp + social + life\_exp + freedom + generosity +   
 corruption + pos\_affect + neg\_affect + ftemp\_c +   
 fpop\_dens + region4, data = happy\_si\_train)  
  
plot(boxCox(fit0, lambda = seq(-2, 3, 1/10)))

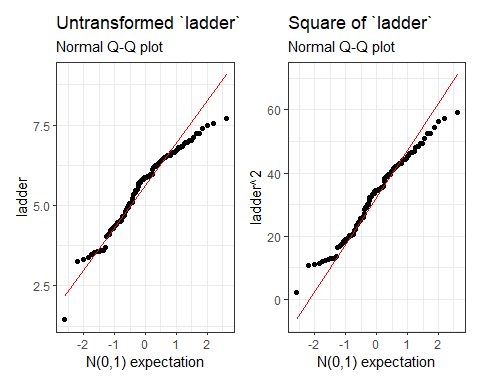




## Normality comparison?

* Do Normal Q-Q plots suggest a big improvement in Normality moving from ladder to ladder squared?

p1 <- ggplot(data = happy\_si\_train, aes(sample = ladder)) +  
 geom\_qq() + geom\_qq\_line(col = "red") +  
 labs(title = "Untransformed `ladder`", subtitle = "Normal Q-Q plot",  
 y = "ladder", x = "N(0,1) expectation")  
  
p2 <- ggplot(data = happy\_si\_train, aes(sample = ladder^2)) +  
 geom\_qq() + geom\_qq\_line(col = "red") +  
 labs(title = "Square of `ladder`", subtitle = "Normal Q-Q plot",  
 y = "ladder^2", x = "N(0,1) expectation")  
  
p1 + p2



## Evaluating Transformation’s Impact

* Consider residuals vs. fitted plots for models fit using ladder and ladder squared.

fit0 <- lm(ladder ~ log\_gdp + social + life\_exp + freedom + generosity +   
 corruption + pos\_affect + neg\_affect + ftemp\_c +   
 fpop\_dens + region4, data = happy\_si\_train)  
  
fit0\_aug <- augment(fit0)  
  
happy\_si\_train <- happy\_si\_train |> mutate(ladder2 = ladder^2)  
  
fitx2 <- lm(ladder2 ~ log\_gdp + social + life\_exp + freedom + generosity +   
 corruption + pos\_affect + neg\_affect + ftemp\_c +   
 fpop\_dens + region4, data = happy\_si\_train)  
  
fitx2\_aug <- augment(fitx2)

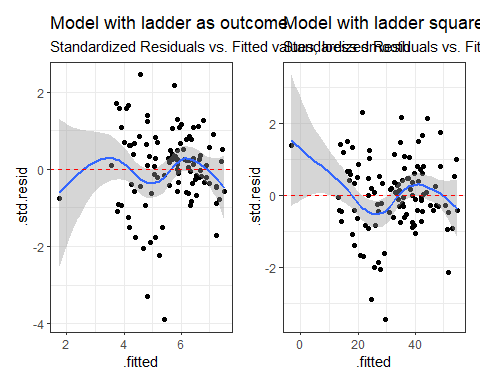
## Residuals vs. Fitted Values?

* Do residuals vs. fitted values plots suggest a big improvement in moving from ladder to ladder squared?

p1 <- ggplot(data = fit0\_aug, aes(x = .fitted, y = .std.resid)) +  
 geom\_point() +  
 geom\_smooth(method = "loess", formula = y ~ x, se = TRUE) +  
 geom\_hline(yintercept = 0, col = "red", lty = "dashed") +  
 labs(title = "Model with ladder as outcome",  
 subtitle = "Standardized Residuals vs. Fitted values; loess smooth")  
  
p2 <- ggplot(data = fitx2\_aug, aes(x = .fitted, y = .std.resid)) +  
 geom\_point() +  
 geom\_smooth(method = "loess", formula = y ~ x, se = TRUE) +  
 geom\_hline(yintercept = 0, col = "red", lty = "dashed") +  
 labs(title = "Model with ladder squared as outcome",  
 subtitle = "Standardized Residuals vs. Fitted values; loess smooth")

## Residuals vs. Fitted Plots

p1 + p2



# Fitting Five Models

## The Five Models We’ll Fit

We’ll predict ladder, without a transformation, using five sets of predictors, selected from the 11 available predictors.

We’ll build them in an order that’s relatively easy for us to think about, but label them (fit1, fit2, …) in a way that helps us later.

We’ll start with a model we’ll call fit4 that includes all 11 predictors, solely as main effects.

## Fitting all 11 predictors: the fit4 model

fit4\_lm <- lm(ladder ~ log\_gdp + social + life\_exp + freedom +  
 generosity + corruption + pos\_affect + neg\_affect +  
 ftemp\_c + fpop\_dens + region4,   
 data = happy\_si\_train)  
  
d <- datadist(happy\_si\_train)  
options(datadist = "d")  
  
fit4\_ols <- ols(ladder ~ log\_gdp + social + life\_exp + freedom +  
 generosity + corruption + pos\_affect + neg\_affect +  
 ftemp\_c + fpop\_dens + region4,   
 data = happy\_si\_train, x = TRUE, y = TRUE)

Do these functions produce the same coefficient values?

identical(as.numeric(fit4\_lm$coefficients),   
 as.numeric(fit4\_ols$coefficients))

[1] TRUE

## Harrell on “Spending df”

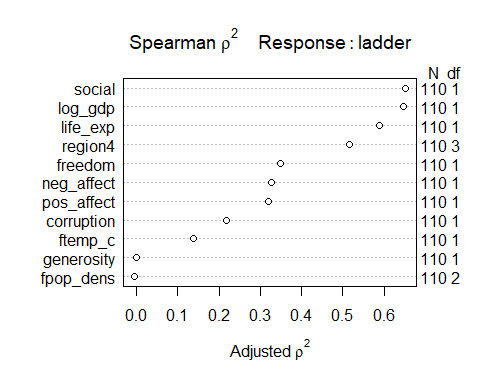
Given a fixed amount of information in the data available for the analysis there is an “information budget” that should be used judiciously: more important predictors should represented in a richer way (e.g. make the number of knots in splines proportional to the overall importance and complexity in the variable) than predictors that are less important.

“Spending df’s with no regrets”: to preserve the operating characteristics of formal inference, once assessed in the modeling process even the less important candidate predictors should remain in view and not elided. All candidate predictors get a portion of the “df budget”, but to a varying extent based on their predictive potential.

<https://hbiostat.org/rmsc/>

## Spearman plot

plot(spearman2(ladder ~ log\_gdp + social + life\_exp + freedom +  
 generosity + corruption + pos\_affect + neg\_affect +  
 ftemp\_c + fpop\_dens + region4, data = happy\_si\_train))



## What do we see in the Spearman plot?

Variables sorted from highest to lowest adjusted :

* log\_gdp (quantitative, so 1 df)
* social (quantitative, so 1 df)
* life\_exp (quantitative, so 1 df)
* region4 (three category, so 3 df)
* and the rest fall into lower tiers

## Non-Linear Terms We’ll Consider

Suppose we decide (a bit arbitrarily) to include the following non-linear terms in fit5:

* a four-knot restricted cubic spline in log\_gdp
* a polynomial of degree 2 in social
* an interaction (product term) between life\_exp and region4

really just to demonstrate methods for fitting and evaluating such things. We’ll call this model fit5 in what follows.

## Add non-linear terms: the fit5 model

fit5\_lm <- lm(ladder ~ rcs(log\_gdp,4) + poly(social,2) +   
 life\_exp \* region4 + freedom + generosity +   
 corruption + pos\_affect + neg\_affect +  
 ftemp\_c + fpop\_dens,  
 data = happy\_si\_train)  
  
# already set up the datadist for happy\_si\_train  
# note the need to use poly in lm() and pol in ols()  
  
fit5\_ols <- ols(ladder ~ rcs(log\_gdp,4) + pol(social,2) +   
 life\_exp \* region4 + freedom + generosity +   
 corruption + pos\_affect + neg\_affect +  
 ftemp\_c + fpop\_dens,   
 data = happy\_si\_train, x = TRUE, y = TRUE)

With the polynomial included, lm() and ols() look a little different, but there’s no meaningful difference in the models or the predictions they create.

## Two predictors + interaction: model fit2

Next, we’ll fit a model using log\_gdp and ftemp\_c and their interaction, which we’ll call fit2…

fit2\_lm <- lm(ladder ~ log\_gdp \* ftemp\_c,  
 data = happy\_si\_train)  
  
# `\*` indicates interaction of log\_gdp (quant.) and ftemp\_c (binary)  
# would be `+` if we just wanted the main effects (no interaction)  
  
  
fit2\_ols <- ols(ladder ~ log\_gdp \* ftemp\_c,   
 data = happy\_si\_train, x = TRUE, y = TRUE)

ols() and lm() still produce same model?

identical(as.numeric(fit2\_lm$coefficients),   
 as.numeric(fit2\_ols$coefficients))

[1] TRUE

## “Best Subsets” to search for good subsets

We’ll use Mallows’ statistic and a “best subsets” search to identify good subsets of 1-6 predictors from our main effects model fit4\_lm. We’ll use ols\_step\_best\_subset() from the **olsrr** package.

fit4\_bestsubs <- ols\_step\_best\_subset(fit4\_lm, metric = "cp", max\_order = 6)

* We want smaller values of , and we also want to catch big drops in
* Other available metric choices include: rsquare, adjr, aic, and sbic

## “Best Subsets” Results (see next slide)

fit4\_bestsubs

Best Subsets Regression   
--------------------------------------------------------------------  
Model Index Predictors  
--------------------------------------------------------------------  
 1 social   
 2 social freedom   
 3 social life\_exp freedom   
 4 social life\_exp freedom corruption   
 5 log\_gdp social freedom corruption region4   
 6 log\_gdp social life\_exp freedom corruption pos\_affect   
--------------------------------------------------------------------  
  
 Subsets Regression Summary   
-----------------------------------------------------------------------------------------------------------------------------------  
 Adj. Pred   
Model R-Square R-Square R-Square C(p) AIC SBIC SBC MSEP FPE HSP APC   
-----------------------------------------------------------------------------------------------------------------------------------  
 1 0.6456 0.6423 0.6294 109.0100 239.4141 -75.1831 247.5155 54.7568 0.5068 0.0047 0.3675   
 2 0.7324 0.7274 0.7166 58.3559 210.5157 -103.7936 221.3176 41.7374 0.3897 0.0036 0.2826   
 3 0.7878 0.7818 0.769 26.7435 186.9992 -126.3744 200.5016 33.4117 0.3147 0.0029 0.2282   
 4 0.8091 0.8018 0.7881 15.8313 177.3736 -135.3014 193.5765 30.3498 0.2884 0.0027 0.2091   
 5 0.8287 0.8170 0.7965 9.9086 171.4251 -144.0448 195.7294 27.4902 0.2686 0.0025 0.1910   
 6 0.8296 0.8197 0.8012 7.3710 168.8544 -142.4225 190.4583 27.6161 0.2669 0.0025 0.1935   
-----------------------------------------------------------------------------------------------------------------------------------  
AIC: Akaike Information Criteria   
 SBIC: Sawa's Bayesian Information Criteria   
 SBC: Schwarz Bayesian Criteria   
 MSEP: Estimated error of prediction, assuming multivariate normality   
 FPE: Final Prediction Error   
 HSP: Hocking's Sp   
 APC: Amemiya Prediction Criteria

## A briefer summary[[2]](#footnote-77)

| Index | Variables |  | AIC |  | Adj. |
| --- | --- | --- | --- | --- | --- |
| (1) | social | 107.8 | 229 | .6057 | .6021 |
| (2) | (1) + freedom | 56.8 | 200 | .7035 | .6980 |
| (3) | (2) + life\_exp | 21.8 | 173 | .7716 | .7652 |
| (4) | (3) + corruption | 14.8 | 167 | .7883 | .7802 |
| (5) | (4) + region4 | 9.1 | 161 | .8098 | .7967 |
| (6) | (5) + log\_gdp | 7.6 | 159 | .8162 | .8017 |

* Which of these seem most promising?

## fit1: 3 predictors, main effects

* Our fit1 will use the “best subsets” suggestion using 3 predictors: social, life\_exp and freedom
  + Why?
  + big drop in (and AIC) from two predictors to three,
  + = 0.7652, adjusted is 0.7498 so not much separation there

## fit1 model fitting

fit1\_lm <- lm(ladder ~ social + life\_exp + freedom,   
 data = happy\_si\_train)  
  
# already set up the datadist for happy\_si\_train  
  
fit1\_ols <- ols(ladder ~ social + life\_exp + freedom,   
 data = happy\_si\_train, x = TRUE, y = TRUE)

Do the models yield the same coefficients?

fit1\_lm$coefficients

(Intercept) social life\_exp freedom   
-4.40887436 3.69980008 0.07265312 2.98311052

fit1\_ols$coefficients

Intercept social life\_exp freedom   
-4.40887436 3.69980008 0.07265312 2.98311052

## fit3: 5 predictors + non-linearity

* Our fit3 will start with the “best subsets” suggestion using 5 predictors: social, life\_exp, freedom, corruption and region4
  + pretty low (and AIC) among these options, = 0.8098, adjusted is 0.7776
  + For demonstration purposes, we’ll also add an interaction between the main effect of life\_exp and region4 and a restricted cubic spline in 3 knots for social and also for life\_exp, and call that model fit3

## fit3 model fitting

fit3\_lm <- lm(ladder ~ rcs(social,3) + rcs(life\_exp, 3) + region4 +   
 life\_exp %ia% region4 + freedom + corruption,   
 data = happy\_si\_train)  
  
# %ia% for interaction using only main effect of life\_exp  
# already set up the datadist for happy\_si\_train  
  
fit3\_ols <- ols(ladder ~ rcs(social,3) + rcs(life\_exp, 3) + region4 +   
 life\_exp %ia% region4 + freedom + corruption,  
 data = happy\_si\_train, x = TRUE, y = TRUE)

And, again, the lm() and ols() fits are the same…

identical(as.numeric(fit3\_ols$coefficients),   
 as.numeric(fit3\_lm$coefficients))

[1] TRUE

## Our five models

| Model | ladder is predicted using… | Vars | NLT | df |
| --- | --- | --- | --- | --- |
| fit1 | social + life\_exp + freedom | 3 | 0 | 3 |
| fit2 | log\_gdp \* ftemp\_c | 2 | 1 | 3 |
| fit3 | rcs(social,3) + rcs(life\_exp, 3) + region4 + life\_exp %ia% region4 + freedom + corruption | 5 | 3 | 12 |
| fit4 | log\_gdp + social + life\_exp + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens + region4 | 11 | 0 | 14 |
| fit5 | rcs(log\_gdp,4) + poly(social,2) + life\_exp \* region4 + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens | 11 | 3 | 20 |

* Vars = number of variables (out of the 11 candidate predictors)
* NLT = number of non-linear terms (splines + polynomials + interactions)
* df = model degrees of freedom

# Estimation of coefficients, interpreting effect sizes

## fit1 Coefficients

model\_parameters(fit1\_ols, ci = 0.90, pretty\_names = FALSE, digits = 3)

Parameter | Coefficient | SE | 90% CI | t(106) | p  
--------------------------------------------------------------------  
Intercept | -4.409 | 0.683 | [-5.542, -3.276] | -6.455 | < .001  
social | 3.700 | 0.599 | [ 2.706, 4.694] | 6.176 | < .001  
life\_exp | 0.073 | 0.014 | [ 0.050, 0.096] | 5.261 | < .001  
freedom | 2.983 | 0.475 | [ 2.194, 3.772] | 6.274 | < .001

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

Effect of freedom in fit1?

* If we have two countries with the same values of social and life\_exp, but country A’s freedom score is one point higher than country B, our model fit1 predicts that, on average, the ladder score for country A will be 2.762 (90% CI: 1.982, 3.541) points higher than country B.

## Interpreting fit1 coefficients

* If we have two countries with the same values of life\_exp and freedom, but country A’s social score is one point higher than country B, our model fit1 predicts that, on average, the ladder score for country A will be 3.615 (90% CI: 2.699, 4.530) points higher than country B.
* If we have two countries with the same values of social and freedom, but country A’s life\_exp score is one point higher than country B, our model fit1 predicts that, on average, the ladder score for country A will be 0.072 (90% CI: 0.051, 0.093) points higher than country B.

## fit1 meaning of the intercept

* If we have a country with 0 values in social, life\_exp and freedom, our predicted ladder score, according to fit1, will be -4.140 (90% CI: -5.214, -3.065).
* Are there problems here?

df\_stats(~ ladder + social + life\_exp + freedom, data = happy\_si\_train) |>  
 gt() |> fmt\_number(columns = min:sd, decimals = 2) |>   
 opt\_stylize(style = 1, color = "gray")

| response | min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ladder | 1.45 | 4.72 | 5.86 | 6.51 | 7.70 | 5.61 | 1.18 | 110 | 0 |
| social | 0.37 | 0.70 | 0.83 | 0.89 | 0.98 | 0.79 | 0.13 | 110 | 0 |
| life\_exp | 52.20 | 61.55 | 66.20 | 69.92 | 74.60 | 65.31 | 5.53 | 110 | 0 |
| freedom | 0.23 | 0.74 | 0.82 | 0.89 | 0.96 | 0.79 | 0.12 | 110 | 0 |

## fit1 Coefficients (lm() vs. ols())

model\_parameters(fit1\_lm, ci = 0.90, pretty\_names = FALSE) |>   
 gt() |> fmt\_number(columns = -c(CI, df\_error), decimals = 3) |>  
 opt\_stylize(style = 6, color = "blue")

| Parameter | Coefficient | SE | CI | CI\_low | CI\_high | t | df\_error | p |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (Intercept) | -4.409 | 0.683 | 0.9 | -5.542 | -3.276 | -6.455 | 106 | 0.000 |
| social | 3.700 | 0.599 | 0.9 | 2.706 | 4.694 | 6.176 | 106 | 0.000 |
| life\_exp | 0.073 | 0.014 | 0.9 | 0.050 | 0.096 | 5.261 | 106 | 0.000 |
| freedom | 2.983 | 0.475 | 0.9 | 2.194 | 3.772 | 6.274 | 106 | 0.000 |

model\_parameters(fit1\_ols, ci = 0.90, pretty\_names = FALSE) |>   
 gt() |> fmt\_number(columns = -c(CI, df\_error), decimals = 3) |>  
 opt\_stylize(style = 6, color = "pink")

| Parameter | Coefficient | SE | CI | CI\_low | CI\_high | t | df\_error | p |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Intercept | -4.409 | 0.683 | 0.9 | -5.542 | -3.276 | -6.455 | 106 | 0.000 |
| social | 3.700 | 0.599 | 0.9 | 2.706 | 4.694 | 6.176 | 106 | 0.000 |
| life\_exp | 0.073 | 0.014 | 0.9 | 0.050 | 0.096 | 5.261 | 106 | 0.000 |
| freedom | 2.983 | 0.475 | 0.9 | 2.194 | 3.772 | 6.274 | 106 | 0.000 |

## Can we use tidy() instead?

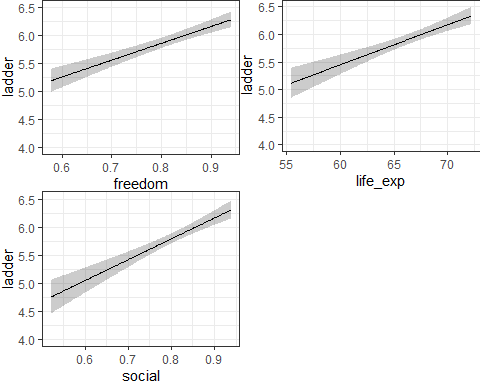
tidy(fit1\_lm, conf.int = TRUE, conf.level = 0.90) |>  
 gt() |> fmt\_number(decimals = 3) |>  
 opt\_stylize(style = 4, color = "cyan")

| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| --- | --- | --- | --- | --- | --- | --- |
| (Intercept) | -4.409 | 0.683 | -6.455 | 0.000 | -5.542 | -3.276 |
| social | 3.700 | 0.599 | 6.176 | 0.000 | 2.706 | 4.694 |
| life\_exp | 0.073 | 0.014 | 5.261 | 0.000 | 0.050 | 0.096 |
| freedom | 2.983 | 0.475 | 6.274 | 0.000 | 2.194 | 3.772 |

but the ols() fit doesn’t work with tidy().

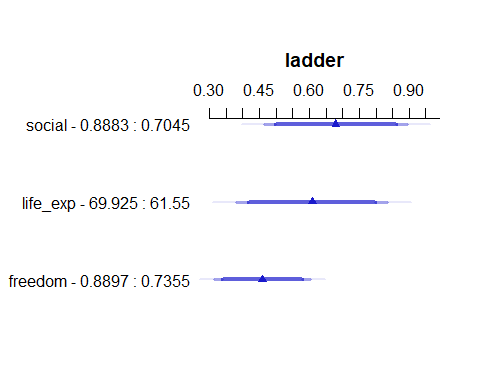
## fit1 prediction plots

ggplot(Predict(fit1\_ols, conf.int = 0.90))



## fit1 Effect Plot

plot(summary(fit1\_ols))



## fit1 Effects Summary

summary(fit1\_ols, conf.int = 0.90)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.9 Upper 0.9  
 social 0.70453 0.88829 0.18376 0.67987 0.110080 0.49721 0.86254   
 life\_exp 61.55000 69.92500 8.37500 0.60847 0.115660 0.41654 0.80040   
 freedom 0.73549 0.88971 0.15422 0.46006 0.073323 0.33839 0.58173

df\_stats(~ social + life\_exp + freedom, data = happy\_si\_train)

response min Q1 median Q3 max mean  
1 social 0.3684781 0.7045286 0.8294196 0.8882879 0.9787889 0.7874424  
2 life\_exp 52.2000000 61.5500000 66.2000000 69.9250000 74.6000000 65.3127273  
3 freedom 0.2283012 0.7354882 0.8152554 0.8897093 0.9648317 0.7927173  
 sd n missing  
1 0.1330279 110 0  
2 5.5254558 110 0  
3 0.1243859 110 0

## fit2 Coefficients

model\_parameters(fit2\_ols, ci = 0.90, pretty\_names = FALSE) |>   
 gt() |> fmt\_number(c(-CI, -df\_error), decimals = 3) |>   
 tab\_options(table.font.size = 24) |>  
 opt\_stylize(style = 6, color = "green")

| Parameter | Coefficient | SE | CI | CI\_low | CI\_high | t | df\_error | p |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Intercept | -5.283 | 1.248 | 0.9 | -7.354 | -3.212 | -4.234 | 106 | 0.000 |
| log\_gdp | 1.107 | 0.122 | 0.9 | 0.905 | 1.309 | 9.099 | 106 | 0.000 |
| ftemp\_c=warm | 4.445 | 1.469 | 0.9 | 2.007 | 6.883 | 3.026 | 106 | 0.003 |
| log\_gdp \* ftemp\_c=warm | -0.425 | 0.150 | 0.9 | -0.673 | -0.176 | -2.838 | 106 | 0.005 |

* Why pretty\_names = FALSE?
  + I like it better when the names in model\_parameters() precisely match what’s in my codebook.

## Can we interpret the fit2 interaction?

$$
\hat{\mbox{ladder}} = -1.919 + 0.788 \mbox{ log\_gdp} - \\
0.958 \mbox{ ftemp\_c = warm} - 0.104 \mbox{ log\_gdp } \times \mbox{ ftemp\_c = warm}
$$

Suppose countries A and B each have “cool” temperatures, but B’s log\_gdp is one unit larger than A’s. Our predicted ladder from fit2:

* for Country A: -1.919 + 0.788 (log\_gdp for A)
* for Country B: -1.919 + 0.788 (log\_gdp for A + 1)
* for Country B - Country A difference = 0.788

## Interpret the fit2 interaction term

$$
\hat{\mbox{ladder}} = -1.919 + 0.788 \mbox{ log\_gdp} - \\
0.958 \mbox{ ftemp\_c = warm} - 0.104 \mbox{ log\_gdp } \times \mbox{ ftemp\_c = warm}
$$

Suppose countries C and D each have “warm” temperatures, but D’s log\_gdp is one unit larger than C’s. From fit2:

* C: (-1.919 - 0.958) + (0.788 - 0.104) (log\_gdp for C)
* D: (-1.919 - 0.958) + (0.788 - 0.104) (log\_gdp for C + 1)
* D - C difference = 0.788 - 0.104 = 0.644

## Interpreting the fit2 interaction

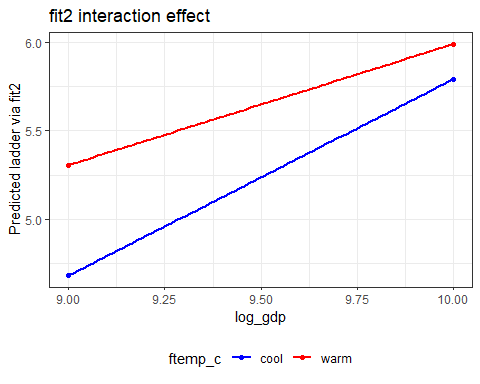
$$
\hat{\mbox{ladder}} = -1.919 + 0.788 \mbox{ log\_gdp} - \\
0.958 \mbox{ ftemp\_c = warm} - 0.104 \mbox{ log\_gdp } \times \mbox{ ftemp\_c = warm}
$$

| – | log\_gdp | ftemp\_c | Predicted ladder from fit1 |
| --- | --- | --- | --- |
| A | 9 | cool | -1.919 + 0.788 (9) - 0.958 (0) - 0.104 (9)(0) = 5.173 |
| B | 10 | cool | -1.919 + 0.788 (10) - 0.958 (0) - 0.104 (10)(0) = 5.961 |
| C | 9 | warm | -1.919 + 0.788 (9) - 0.958 (1) - 0.104 (9)(1) = 3.279 |
| D | 10 | warm | -1.919 + 0.788 (10) - 0.958 (1) - 0.104 (10)(1) = 3.963 |

* B - A = 5.961 - 5.173 = 0.788 compares 10 vs. 9 when temp = cool
* D - C = 3.963 - 3.279 = 0.684 compares 10 vs. 9 when temp = warm
* A - C = 5.173 - 3.279 = 1.894 compares cool vs. warm when log\_gdp = 9
* B - D = 5.961 - 3.963 = 1.998 compares cool vs. warm when log\_gdp = 10

## Picturing the fit2 Model

new\_dat <- tibble(log\_gdp = c(9, 10, 9, 10),   
 ftemp\_c = c("cool", "cool", "warm", "warm"))  
  
fit2\_aug <- augment(fit2\_lm, newdata = new\_dat)  
  
ggplot(data = fit2\_aug, aes(x = log\_gdp, y = .fitted,   
 col = ftemp\_c, group = ftemp\_c)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", formula = y ~ x, se = FALSE) +  
 scale\_color\_manual(values = c("blue", "red")) +  
 theme(legend.position = "bottom") +  
 labs(title = "fit2 interaction effect",  
 y = "Predicted ladder via fit2")



## fit2 Conclusions?

| Parameter | Coefficient | SE | CI | CI\_low | CI\_high | t | df\_error | p |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Intercept | -5.283 | 1.248 | 0.9 | -7.354 | -3.212 | -4.234 | 106 | 0.000 |
| log\_gdp | 1.107 | 0.122 | 0.9 | 0.905 | 1.309 | 9.099 | 106 | 0.000 |
| ftemp\_c=warm | 4.445 | 1.469 | 0.9 | 2.007 | 6.883 | 3.026 | 106 | 0.003 |
| log\_gdp \* ftemp\_c=warm | -0.425 | 0.150 | 0.9 | -0.673 | -0.176 | -2.838 | 106 | 0.005 |

* What is the log\_gdp effect?
  + Thanks to the inclusion of our interaction term (product term), **it depends** on ftemp\_c.
* What is the ftemp\_c effect? **It depends** on log\_gdp.
* Does the intercept’s value indicate something to us?

## happy\_si\_train’s quantities

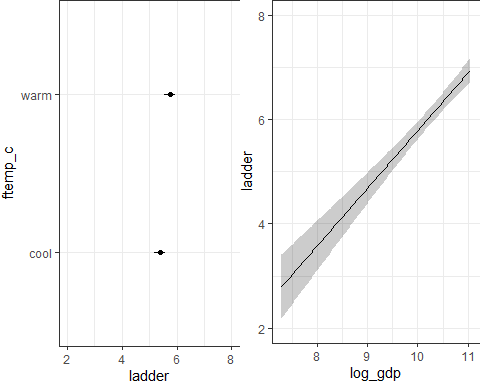
df\_stats(~ ladder + log\_gdp + social + life\_exp + freedom + generosity +   
 corruption + pos\_affect + neg\_affect, data = happy\_si\_train) |>  
 gt() |> fmt\_number(columns = min:sd, decimals = 2) |>  
 tab\_options(table.font.size = 20) |> opt\_stylize(style = 2, color = "gray")

Warning in formula.character(object, env = baseenv()): Using formula(x) is deprecated when x is a character vector of length > 1.  
 Consider formula(paste(x, collapse = " ")) instead.

| response | min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ladder | 1.45 | 4.72 | 5.86 | 6.51 | 7.70 | 5.61 | 1.18 | 110 | 0 |
| log\_gdp | 7.08 | 8.59 | 9.64 | 10.56 | 11.68 | 9.53 | 1.20 | 110 | 0 |
| social | 0.37 | 0.70 | 0.83 | 0.89 | 0.98 | 0.79 | 0.13 | 110 | 0 |
| life\_exp | 52.20 | 61.55 | 66.20 | 69.92 | 74.60 | 65.31 | 5.53 | 110 | 0 |
| freedom | 0.23 | 0.74 | 0.82 | 0.89 | 0.96 | 0.79 | 0.12 | 110 | 0 |
| generosity | -0.27 | -0.10 | 0.02 | 0.13 | 0.59 | 0.03 | 0.17 | 110 | 0 |
| corruption | 0.18 | 0.66 | 0.77 | 0.84 | 0.95 | 0.72 | 0.17 | 110 | 0 |
| pos\_affect | 0.26 | 0.58 | 0.66 | 0.74 | 0.83 | 0.65 | 0.11 | 110 | 0 |
| neg\_affect | 0.11 | 0.23 | 0.28 | 0.36 | 0.52 | 0.29 | 0.09 | 110 | 0 |

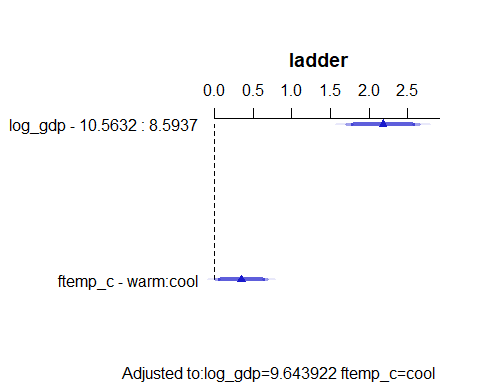
## fit2 prediction plots

ggplot(Predict(fit2\_ols, conf.int = 0.90), layout = c(1,2))



## fit2 Effect Plot

plot(summary(fit2\_ols))



## fit2 Effects Summary

summary(fit2\_ols, conf.int = 0.90)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.9 Upper 0.9  
 log\_gdp 8.5937 10.563 1.9695 2.1809 0.23969 1.783200 2.57860   
 ftemp\_c - warm:cool 1.0000 2.000 NA 0.3497 0.17062 0.066586 0.63282   
  
Adjusted to: log\_gdp=9.643922 ftemp\_c=cool

## fit3 Coefficients

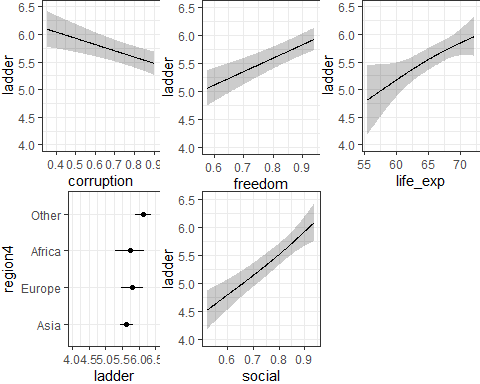
model\_parameters(fit3\_ols, ci = 0.90, pretty\_names = FALSE, digits = 3)

Parameter | Coefficient | SE | 90% CI | t(97) | p  
------------------------------------------------------------------------------------  
Intercept | -3.609 | 3.005 | [-8.600, 1.381] | -1.201 | 0.233   
social | 3.410 | 0.844 | [ 2.009, 4.811] | 4.043 | < .001  
social' | 0.377 | 0.937 | [-1.179, 1.933] | 0.402 | 0.688   
life\_exp | 0.081 | 0.052 | [-0.006, 0.168] | 1.539 | 0.127   
life\_exp' | -0.014 | 0.045 | [-0.089, 0.061] | -0.305 | 0.761   
region4=Europe | -0.530 | 3.980 | [-7.140, 6.080] | -0.133 | 0.894   
region4=Africa | 1.941 | 3.108 | [-3.219, 7.102] | 0.625 | 0.534   
region4=Other | 1.605 | 3.903 | [-4.877, 8.088] | 0.411 | 0.682   
life\_exp \* region4=Europe | 0.011 | 0.058 | [-0.085, 0.107] | 0.183 | 0.855   
life\_exp \* region4=Africa | -0.028 | 0.050 | [-0.112, 0.056] | -0.551 | 0.583   
life\_exp \* region4=Other | -0.017 | 0.058 | [-0.113, 0.079] | -0.289 | 0.774   
freedom | 2.394 | 0.526 | [ 1.520, 3.268] | 4.550 | < .001  
corruption | -1.136 | 0.382 | [-1.770, -0.501] | -2.973 | 0.004

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

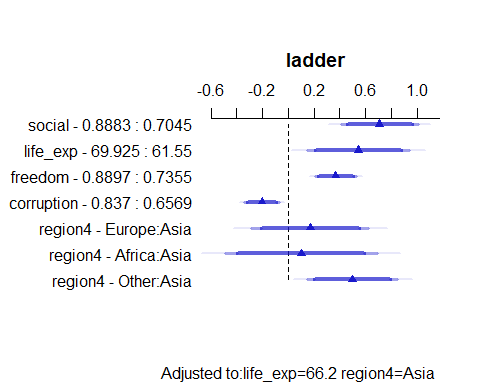
## fit3 prediction plots

ggplot(Predict(fit3\_ols, conf.int = 0.90))



## fit3 Effect Plot

plot(summary(fit3\_ols))



## fit3 Effects Summary

summary(fit3\_ols, conf.int = 0.90)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.9  
 social 0.70453 0.88829 0.18376 0.71013 0.153280 0.45557   
 life\_exp 61.55000 69.92500 8.37500 0.54609 0.202920 0.20909   
 freedom 0.73549 0.88971 0.15422 0.36921 0.081151 0.23444   
 corruption 0.65687 0.83695 0.18008 -0.20452 0.068793 -0.31876   
 region4 - Europe:Asia 1.00000 2.00000 NA 0.16978 0.232140 -0.21573   
 region4 - Africa:Asia 1.00000 3.00000 NA 0.10064 0.300710 -0.39875   
 region4 - Other:Asia 1.00000 4.00000 NA 0.50075 0.178850 0.20373   
 Upper 0.9  
 0.96469   
 0.88308   
 0.50398   
 -0.09027   
 0.55529   
 0.60003   
 0.79776   
  
Adjusted to: life\_exp=66.2 region4=Asia

## fit4 Coefficients

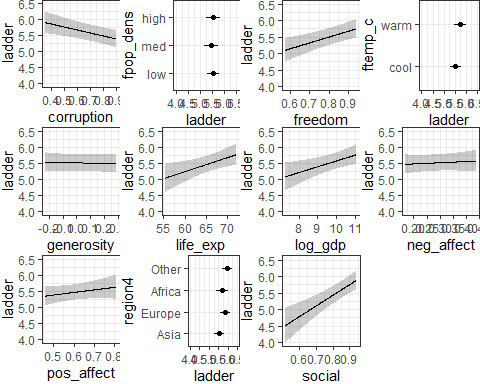
model\_parameters(fit4\_ols, ci = 0.90, pretty\_names = FALSE, digits = 3)

Parameter | Coefficient | SE | 90% CI | t(95) | p  
-------------------------------------------------------------------------  
Intercept | -3.225 | 1.359 | [-5.482, -0.968] | -2.374 | 0.020   
log\_gdp | 0.184 | 0.082 | [ 0.047, 0.320] | 2.239 | 0.027   
social | 3.263 | 0.829 | [ 1.887, 4.640] | 3.937 | < .001  
life\_exp | 0.044 | 0.021 | [ 0.009, 0.078] | 2.114 | 0.037   
freedom | 1.782 | 0.603 | [ 0.780, 2.784] | 2.953 | 0.004   
generosity | -0.092 | 0.339 | [-0.655, 0.470] | -0.272 | 0.786   
corruption | -0.932 | 0.355 | [-1.521, -0.343] | -2.629 | 0.010   
pos\_affect | 0.810 | 0.815 | [-0.544, 2.164] | 0.994 | 0.323   
neg\_affect | 0.372 | 0.824 | [-0.997, 1.740] | 0.451 | 0.653   
ftemp\_c=warm | 0.217 | 0.163 | [-0.055, 0.488] | 1.327 | 0.188   
fpop\_dens=med | -0.088 | 0.144 | [-0.327, 0.150] | -0.615 | 0.540   
fpop\_dens=high | 0.002 | 0.150 | [-0.247, 0.251] | 0.015 | 0.988   
region4=Europe | 0.333 | 0.173 | [ 0.046, 0.619] | 1.925 | 0.057   
region4=Africa | 0.148 | 0.183 | [-0.156, 0.452] | 0.809 | 0.420   
region4=Other | 0.413 | 0.202 | [ 0.077, 0.748] | 2.045 | 0.044

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

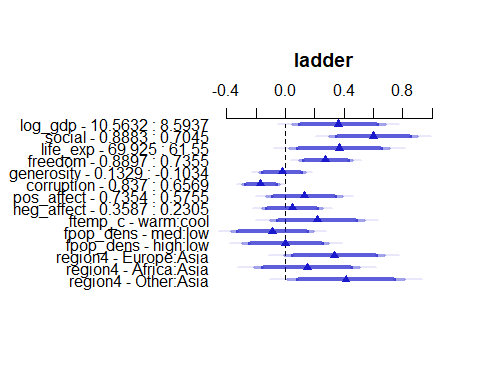
## fit4 prediction plots

ggplot(Predict(fit4\_ols, conf.int = 0.90))



## fit4 Effect Plot

plot(summary(fit4\_ols))



## fit4 Effects Summary

summary(fit4\_ols, conf.int = 0.90)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.9  
 log\_gdp 8.59370 10.56300 1.96950 0.3615700 0.161480 0.093340  
 social 0.70453 0.88829 0.18376 0.5996800 0.152320 0.346670  
 life\_exp 61.55000 69.92500 8.37500 0.3670700 0.173650 0.078630  
 freedom 0.73549 0.88971 0.15422 0.2748300 0.093070 0.120240  
 generosity -0.10340 0.13287 0.23626 -0.0217880 0.080027 -0.154720  
 corruption 0.65687 0.83695 0.18008 -0.1678800 0.063846 -0.273930  
 pos\_affect 0.57547 0.73544 0.15997 0.1296100 0.130390 -0.086978  
 neg\_affect 0.23054 0.35869 0.12815 0.0476340 0.105570 -0.127720  
 ftemp\_c - warm:cool 1.00000 2.00000 NA 0.2165700 0.163200 -0.054507  
 fpop\_dens - med:low 1.00000 2.00000 NA -0.0882950 0.143620 -0.326860  
 fpop\_dens - high:low 1.00000 3.00000 NA 0.0023078 0.150010 -0.246870  
 region4 - Europe:Asia 1.00000 2.00000 NA 0.3325200 0.172740 0.045580  
 region4 - Africa:Asia 1.00000 3.00000 NA 0.1481400 0.183040 -0.155890  
 region4 - Other:Asia 1.00000 4.00000 NA 0.4129400 0.201970 0.077460  
 Upper 0.9  
 0.629800  
 0.852680  
 0.655520  
 0.429430  
 0.111140  
 -0.061827  
 0.346190  
 0.222990  
 0.487650  
 0.150270  
 0.251490  
 0.619460  
 0.452170  
 0.748410

## fit5 Coefficients

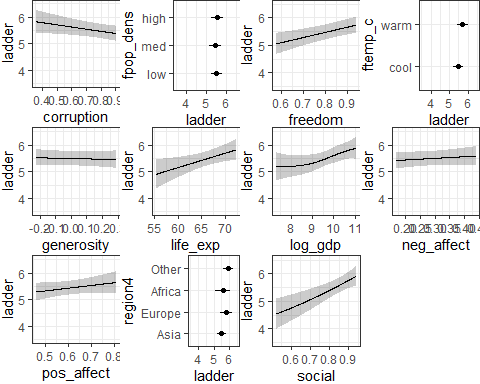
model\_parameters(fit5\_ols, ci = 0.90, pretty\_names = FALSE, digits = 3)

Parameter | Coefficient | SE | 90% CI | t(89) | p  
-----------------------------------------------------------------------------------  
Intercept | -2.194 | 2.307 | [-6.028, 1.641] | -0.951 | 0.344  
log\_gdp | 0.015 | 0.213 | [-0.339, 0.369] | 0.070 | 0.944  
log\_gdp' | 0.307 | 0.391 | [-0.343, 0.958] | 0.785 | 0.434  
log\_gdp'' | -1.154 | 2.181 | [-4.778, 2.471] | -0.529 | 0.598  
social | 1.022 | 3.910 | [-5.477, 7.521] | 0.261 | 0.794  
social^2 | 1.547 | 2.885 | [-3.249, 6.343] | 0.536 | 0.593  
life\_exp | 0.055 | 0.027 | [ 0.010, 0.099] | 2.028 | 0.046  
region4=Europe | 2.331 | 3.749 | [-3.901, 8.562] | 0.622 | 0.536  
region4=Africa | 1.157 | 2.397 | [-2.828, 5.141] | 0.483 | 0.631  
region4=Other | 3.499 | 3.817 | [-2.845, 9.842] | 0.917 | 0.362  
freedom | 1.868 | 0.650 | [ 0.788, 2.949] | 2.874 | 0.005  
generosity | -0.132 | 0.375 | [-0.756, 0.492] | -0.351 | 0.726  
corruption | -0.813 | 0.463 | [-1.583, -0.042] | -1.754 | 0.083  
pos\_affect | 1.024 | 0.911 | [-0.491, 2.538] | 1.124 | 0.264  
neg\_affect | 0.618 | 0.892 | [-0.864, 2.100] | 0.693 | 0.490  
ftemp\_c=warm | 0.210 | 0.173 | [-0.076, 0.497] | 1.220 | 0.226  
fpop\_dens=med | -0.051 | 0.151 | [-0.302, 0.200] | -0.336 | 0.738  
fpop\_dens=high | 0.024 | 0.157 | [-0.236, 0.285] | 0.156 | 0.877  
life\_exp \* region4=Europe | -0.031 | 0.054 | [-0.121, 0.060] | -0.561 | 0.576  
life\_exp \* region4=Africa | -0.016 | 0.039 | [-0.081, 0.049] | -0.414 | 0.680  
life\_exp \* region4=Other | -0.046 | 0.057 | [-0.140, 0.048] | -0.820 | 0.415

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

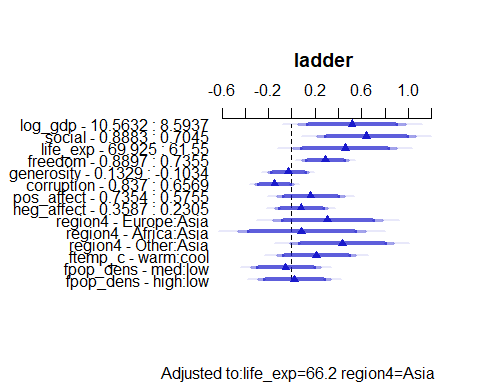
## fit5 prediction plots

ggplot(Predict(fit5\_ols, conf.int = 0.90))



## fit5 Effect Plot

plot(summary(fit5\_ols))



## fit5 Effects Summary

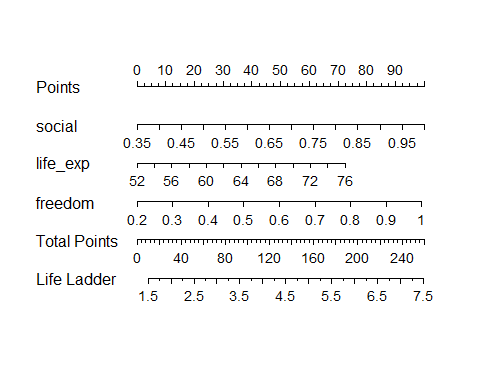
summary(fit5\_ols, conf.int = 0.90)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.9  
 log\_gdp 8.59370 10.56300 1.96950 0.517330 0.233410 0.129360  
 social 0.70453 0.88829 0.18376 0.640490 0.215580 0.282170  
 life\_exp 61.55000 69.92500 8.37500 0.457310 0.225510 0.082477  
 freedom 0.73549 0.88971 0.15422 0.288150 0.100270 0.121490  
 generosity -0.10340 0.13287 0.23626 -0.031164 0.088691 -0.178580  
 corruption 0.65687 0.83695 0.18008 -0.146360 0.083451 -0.285060  
 pos\_affect 0.57547 0.73544 0.15997 0.163740 0.145740 -0.078500  
 neg\_affect 0.23054 0.35869 0.12815 0.079188 0.114290 -0.110770  
 region4 - Europe:Asia 1.00000 2.00000 NA 0.308270 0.239050 -0.089066  
 region4 - Africa:Asia 1.00000 3.00000 NA 0.085235 0.279050 -0.378590  
 region4 - Other:Asia 1.00000 4.00000 NA 0.432400 0.225100 0.058252  
 ftemp\_c - warm:cool 1.00000 2.00000 NA 0.210450 0.172550 -0.076355  
 fpop\_dens - med:low 1.00000 2.00000 NA -0.050793 0.151130 -0.301990  
 fpop\_dens - high:low 1.00000 3.00000 NA 0.024415 0.156710 -0.236060  
 Upper 0.9   
 0.9053000  
 0.9988100  
 0.8321400  
 0.4548100  
 0.1162500  
 -0.0076478  
 0.4059900  
 0.2691500  
 0.7056000  
 0.5490600  
 0.8065500  
 0.4972500  
 0.2004100  
 0.2848900  
  
Adjusted to: life\_exp=66.2 region4=Asia

# Nomograms for our five models

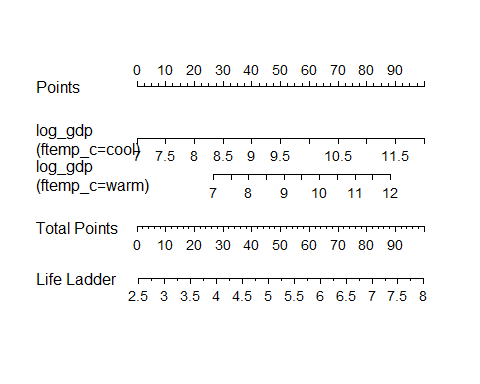
## fit1 nomogram

plot(nomogram(fit1\_ols), lplabel = "Life Ladder")



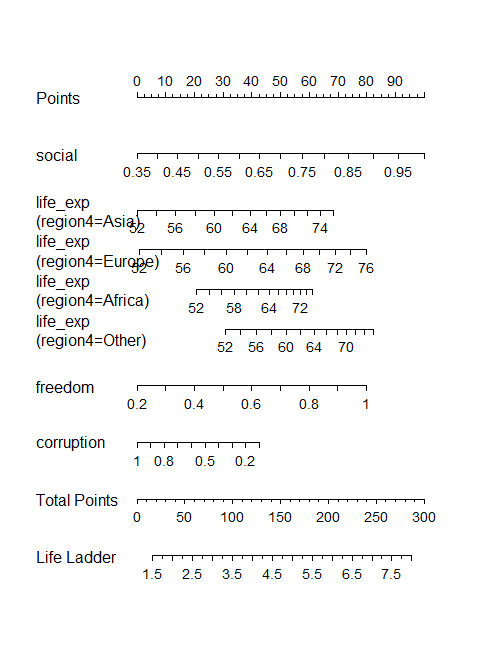
## fit2 nomogram

plot(nomogram(fit2\_ols), lplabel = "Life Ladder")



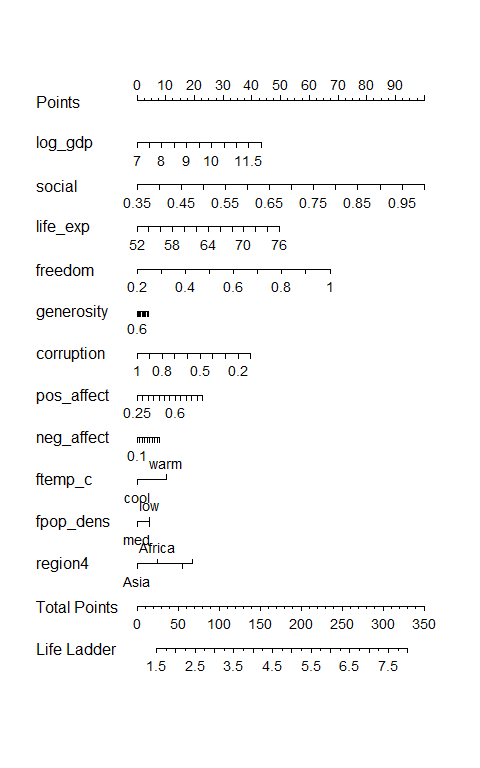
## fit3 nomogram

plot(nomogram(fit3\_ols), lplabel = "Life Ladder")



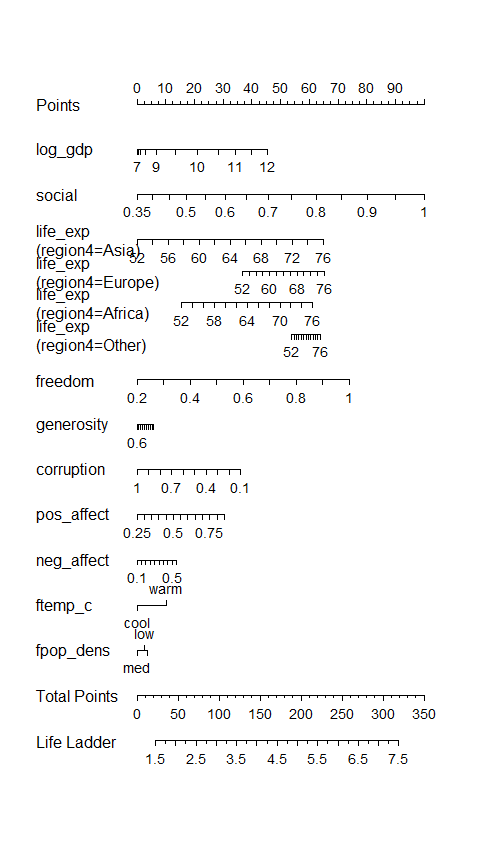
## fit4 nomogram

plot(nomogram(fit4\_ols), lplabel = "Life Ladder")



## fit5 nomogram

plot(nomogram(fit5\_ols), lplabel = "Life Ladder")



# ANOVA tables and measures

## Our five models

| Model | Predictors |
| --- | --- |
| fit1 | social, life\_exp, and freedom main effects |
| fit2 | log\_gdp, ftemp\_c and their interaction |
| fit3 | social, life\_exp, freedom, corruption, region4, 2 splines and an interaction |
| fit4 | main effects of all 11 predictors |
| fit5 | all 11 predictors plus a spline, a polynomial and an interaction |

## anova for fit1

anova(fit1\_lm)

Analysis of Variance Table  
  
Response: ladder  
 Df Sum Sq Mean Sq F value Pr(>F)   
social 1 97.942 97.942 322.508 < 2.2e-16 \*\*\*  
life\_exp 1 9.614 9.614 31.659 1.512e-07 \*\*\*  
freedom 1 11.956 11.956 39.368 7.799e-09 \*\*\*  
Residuals 106 32.191 0.304   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

* Total SS = 75.448+11.393+9.276+28.446 = 124.563
* So social accounts for 75.448 / 124.563 = 60.57% of the variation in ladder

## Effect Sizes, by

* : What proportion of the total variance in ladder is accounted for by each of the predictors, ignoring the other variables?

eta\_squared(fit1\_lm, partial = FALSE, ci = 0.90)

# Effect Size for ANOVA (Type I)  
  
Parameter | Eta2 | 90% CI  
-------------------------------  
social | 0.65 | [0.58, 1.00]  
life\_exp | 0.06 | [0.02, 1.00]  
freedom | 0.08 | [0.03, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

## Effect Sizes, by Partial

* Partial : **After** adjusting for the other predictors, what proportion of the *remaining* variance in ladder is accounted for by each predictor?

eta\_squared(fit1\_lm, partial = TRUE, ci = 0.90)

# Effect Size for ANOVA (Type I)  
  
Parameter | Eta2 (partial) | 90% CI  
-----------------------------------------  
social | 0.75 | [0.70, 1.00]  
life\_exp | 0.23 | [0.15, 1.00]  
freedom | 0.27 | [0.18, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

## lm() vs. ols() for ANOVA?

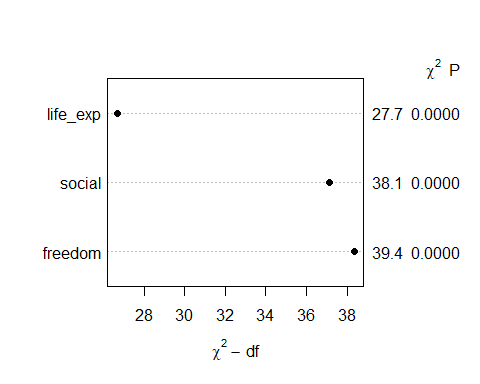
* Different arrangement, includes REGRESSION totals.

anova(fit1\_ols)

Analysis of Variance Response: ladder   
  
 Factor d.f. Partial SS MS F P   
 social 1 11.584115 11.5841147 38.14 <.0001  
 life\_exp 1 8.404431 8.4044312 27.67 <.0001  
 freedom 1 11.955681 11.9556806 39.37 <.0001  
 REGRESSION 3 119.512031 39.8373438 131.18 <.0001  
 ERROR 106 32.190987 0.3036886

## fit1 ANOVA plot with

plot(anova(fit1\_ols))



## Model fit2

| Model | Predictors |
| --- | --- |
| fit1 | social, life\_exp, and freedom main effects |
| fit2 | log\_gdp, ftemp\_c and their interaction |
| fit3 | social, life\_exp, freedom, corruption, region4, 2 splines and an interaction |
| fit4 | main effects of all 11 predictors |
| fit5 | all 11 predictors plus a spline, a polynomial and an interaction |

## anova for fit2

anova(fit2\_ols)

Analysis of Variance Response: ladder   
  
 Factor d.f. Partial SS MS   
 log\_gdp (Factor+Higher Order Factors) 2 76.206175 38.1030874  
 All Interactions 1 4.254008 4.2540080  
 ftemp\_c (Factor+Higher Order Factors) 2 5.943874 2.9719372  
 All Interactions 1 4.254008 4.2540080  
 log\_gdp \* ftemp\_c (Factor+Higher Order Factors) 1 4.254008 4.2540080  
 REGRESSION 3 95.713044 31.9043481  
 ERROR 106 55.989974 0.5282073  
 F P   
 72.14 <.0001  
 8.05 0.0054  
 5.63 0.0048  
 8.05 0.0054  
 8.05 0.0054  
 60.40 <.0001

## fit2 Effect Sizes, by

eta\_squared(fit2\_lm, partial = FALSE, ci = 0.90)

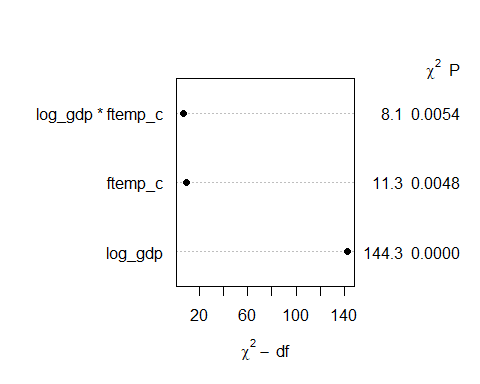
# Effect Size for ANOVA (Type I)  
  
Parameter | Eta2 | 90% CI  
-------------------------------------  
log\_gdp | 0.59 | [0.52, 1.00]  
ftemp\_c | 0.01 | [0.00, 1.00]  
log\_gdp:ftemp\_c | 0.03 | [0.00, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

eta\_squared(fit2\_lm, partial = TRUE, ci = 0.90)

# Effect Size for ANOVA (Type I)  
  
Parameter | Eta2 (partial) | 90% CI  
-----------------------------------------------  
log\_gdp | 0.62 | [0.55, 1.00]  
ftemp\_c | 0.03 | [0.00, 1.00]  
log\_gdp:ftemp\_c | 0.07 | [0.02, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

## fit2 ANOVA plot with

plot(anova(fit2\_ols))



## Model fit3

| Model | Predictors |
| --- | --- |
| fit1 | social, life\_exp, and freedom main effects |
| fit2 | log\_gdp, ftemp\_c and their interaction |
| fit3 | social, life\_exp, freedom, corruption, region4, 2 splines and an interaction |
| fit4 | main effects of all 11 predictors |
| fit5 | all 11 predictors plus a spline, a polynomial and an interaction |

## anova for fit3

anova(fit3\_ols)

Analysis of Variance Response: ladder   
  
 Factor d.f. Partial SS   
 social 2 9.26500440  
 Nonlinear 1 0.04336482  
 life\_exp (Factor+Higher Order Factors) 5 3.40232700  
 All Interactions 3 0.11919164  
 Nonlinear 1 0.02495166  
 region4 (Factor+Higher Order Factors) 6 2.80529242  
 All Interactions 3 0.11919164  
 life\_exp \* region4 (Factor+Higher Order Factors) 3 0.11919164  
 freedom 1 5.54493878  
 corruption 1 2.36756166  
 TOTAL NONLINEAR 2 0.06259064  
 TOTAL NONLINEAR + INTERACTION 5 0.15269452  
 REGRESSION 12 125.71891733  
 ERROR 97 25.98410113  
 MS F P   
 4.63250220 17.29 <.0001  
 0.04336482 0.16 0.6883  
 0.68046540 2.54 0.0332  
 0.03973055 0.15 0.9305  
 0.02495166 0.09 0.7609  
 0.46754874 1.75 0.1186  
 0.03973055 0.15 0.9305  
 0.03973055 0.15 0.9305  
 5.54493878 20.70 <.0001  
 2.36756166 8.84 0.0037  
 0.03129532 0.12 0.8899  
 0.03053890 0.11 0.9890  
 10.47657644 39.11 <.0001  
 0.26787733

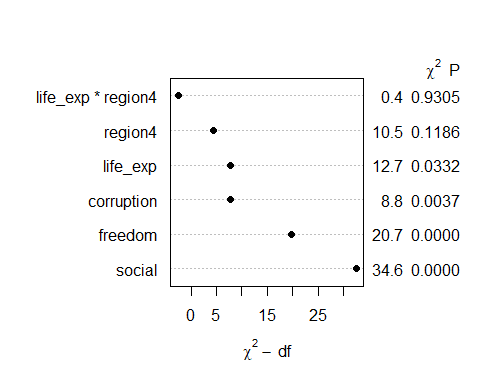
## fit3 Effect Sizes, by

eta\_squared(fit3\_lm, partial = FALSE, ci = 0.90)

# Effect Size for ANOVA (Type I)  
  
Parameter | Eta2 | 90% CI  
-------------------------------------------  
rcs(social, 3) | 0.66 | [0.59, 1.00]  
rcs(life\_exp, 3) | 0.05 | [0.00, 1.00]  
region4 | 0.02 | [0.00, 1.00]  
life\_exp %ia% region4 | 0.02 | [0.00, 1.00]  
freedom | 0.06 | [0.01, 1.00]  
corruption | 0.02 | [0.00, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

## fit3 ANOVA plot with

plot(anova(fit3\_ols))



## Model fit4

| Model | Predictors |
| --- | --- |
| fit1 | social, life\_exp, and freedom main effects |
| fit2 | log\_gdp, ftemp\_c and their interaction |
| fit3 | social, life\_exp, freedom, corruption, region4, 2 splines and an interaction |
| fit4 | main effects of all 11 predictors |
| fit5 | all 11 predictors plus a spline, a polynomial and an interaction |

## anova for fit4

anova(fit4\_ols)

Analysis of Variance Response: ladder   
  
 Factor d.f. Partial SS MS F P   
 log\_gdp 1 1.25356388 1.25356388 5.01 0.0275  
 social 1 3.87562434 3.87562434 15.50 0.0002  
 life\_exp 1 1.11727257 1.11727257 4.47 0.0371  
 freedom 1 2.18030115 2.18030115 8.72 0.0040  
 generosity 1 0.01853394 0.01853394 0.07 0.7860  
 corruption 1 1.72876208 1.72876208 6.91 0.0100  
 pos\_affect 1 0.24704310 0.24704310 0.99 0.3228  
 neg\_affect 1 0.05090620 0.05090620 0.20 0.6529  
 ftemp\_c 1 0.44033891 0.44033891 1.76 0.1877  
 fpop\_dens 2 0.14669868 0.07334934 0.29 0.7464  
 region4 3 1.51418790 0.50472930 2.02 0.1165  
 REGRESSION 14 127.94924169 9.13923155 36.55 <.0001  
 ERROR 95 23.75377677 0.25003976

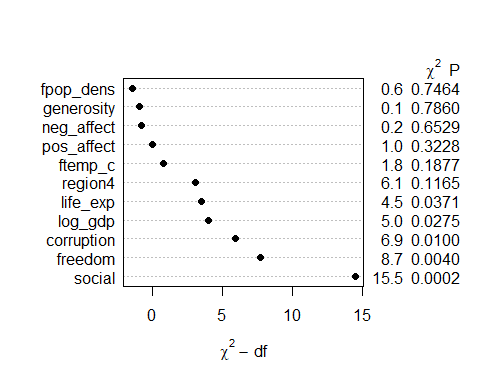
## fit4 Effect Sizes, by

eta\_squared(fit4\_lm, partial = FALSE, ci = 0.90)

# Effect Size for ANOVA (Type I)  
  
Parameter | Eta2 | 90% CI  
------------------------------------  
log\_gdp | 0.59 | [0.51, 1.00]  
social | 0.12 | [0.05, 1.00]  
life\_exp | 0.01 | [0.00, 1.00]  
freedom | 0.07 | [0.02, 1.00]  
generosity | 1.86e-04 | [0.00, 1.00]  
corruption | 0.01 | [0.00, 1.00]  
pos\_affect | 0.01 | [0.00, 1.00]  
neg\_affect | 1.33e-03 | [0.00, 1.00]  
ftemp\_c | 9.49e-04 | [0.00, 1.00]  
fpop\_dens | 7.32e-04 | [0.00, 1.00]  
region4 | 9.98e-03 | [0.00, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

## fit4 ANOVA plot with

plot(anova(fit4\_ols))



## Model fit5

| Model | Predictors |
| --- | --- |
| fit1 | social, life\_exp, and freedom main effects |
| fit2 | log\_gdp, ftemp\_c and their interaction |
| fit3 | social, life\_exp, freedom, corruption, region4, 2 splines and an interaction |
| fit4 | main effects of all 11 predictors |
| fit5 | all 11 predictors plus a spline, a polynomial and an interaction |

## anova for fit5

anova(fit5\_ols)

Analysis of Variance Response: ladder   
  
 Factor d.f. Partial SS MS   
 log\_gdp 3 1.65562721 0.55187574  
 Nonlinear 2 0.24516384 0.12258192  
 social 2 3.28660279 1.64330140  
 Nonlinear 1 0.07476777 0.07476777  
 life\_exp (Factor+Higher Order Factors) 4 1.25083512 0.31270878  
 All Interactions 3 0.23296391 0.07765464  
 region4 (Factor+Higher Order Factors) 6 1.37523133 0.22920522  
 All Interactions 3 0.23296391 0.07765464  
 freedom 1 2.14886805 2.14886805  
 generosity 1 0.03212648 0.03212648  
 corruption 1 0.80032779 0.80032779  
 pos\_affect 1 0.32845422 0.32845422  
 neg\_affect 1 0.12492141 0.12492141  
 ftemp\_c 1 0.38705306 0.38705306  
 fpop\_dens 2 0.07669877 0.03834939  
 life\_exp \* region4 (Factor+Higher Order Factors) 3 0.23296391 0.07765464  
 TOTAL NONLINEAR 3 0.40993454 0.13664485  
 TOTAL NONLINEAR + INTERACTION 6 0.59592510 0.09932085  
 REGRESSION 20 128.54516679 6.42725834  
 ERROR 89 23.15785167 0.26020058  
 F P   
 2.12 0.1032  
 0.47 0.6259  
 6.32 0.0027  
 0.29 0.5933  
 1.20 0.3157  
 0.30 0.8264  
 0.88 0.5124  
 0.30 0.8264  
 8.26 0.0051  
 0.12 0.7261  
 3.08 0.0829  
 1.26 0.2642  
 0.48 0.4902  
 1.49 0.2258  
 0.15 0.8632  
 0.30 0.8264  
 0.53 0.6661  
 0.38 0.8889  
 24.70 <.0001

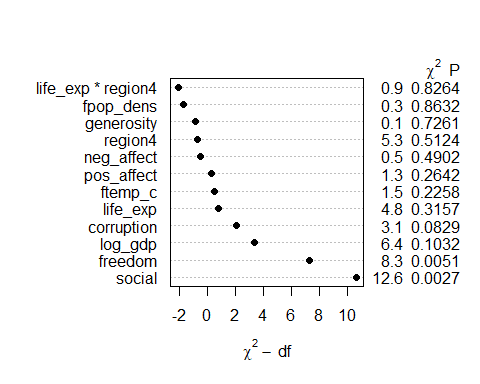
## fit5 Effect Sizes, by

eta\_squared(fit4\_lm, partial = FALSE, ci = 0.90)

# Effect Size for ANOVA (Type I)  
  
Parameter | Eta2 | 90% CI  
------------------------------------  
log\_gdp | 0.59 | [0.51, 1.00]  
social | 0.12 | [0.05, 1.00]  
life\_exp | 0.01 | [0.00, 1.00]  
freedom | 0.07 | [0.02, 1.00]  
generosity | 1.86e-04 | [0.00, 1.00]  
corruption | 0.01 | [0.00, 1.00]  
pos\_affect | 0.01 | [0.00, 1.00]  
neg\_affect | 1.33e-03 | [0.00, 1.00]  
ftemp\_c | 9.49e-04 | [0.00, 1.00]  
fpop\_dens | 7.32e-04 | [0.00, 1.00]  
region4 | 9.98e-03 | [0.00, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

## fit5 ANOVA plot with

plot(anova(fit5\_ols))



# Model Performance for our Five Models

## Our five models

| Model | ladder is predicted using… | df |
| --- | --- | --- |
| fit1 | social + life\_exp + freedom | 3 |
| fit2 | log\_gdp \* ftemp\_c | 3 |
| fit3 | rcs(social,3) + rcs(life\_exp, 3) + region4 + life\_exp %ia% region4 + freedom + corruption | 12 |
| fit4 | log\_gdp + social + life\_exp + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens + region4 | 14 |
| fit5 | rcs(log\_gdp,4) + poly(social,2) + life\_exp \* region4 + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens | 20 |

## Model Performance for fit1

model\_performance(fit1\_lm) |>  
 mutate(n = n\_obs(fit1\_lm), df = n - fit1\_lm$df.residual - 1) |>  
 mutate(model = "fit1") |> relocate(model, everything()) |>  
 gt() |> fmt\_number(AIC:Sigma, decimals = 3) |>   
 fmt\_number(n:df, decimals = 0) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| model | AIC | AICc | BIC | R2 | R2\_adjusted | RMSE | Sigma | n | df |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fit1 | 186.999 | 187.576 | 200.502 | 0.788 | 0.782 | 0.541 | 0.551 | 110 | 3 |

### Does it matter if instead we use the ols() fit?

* Yes, a little. model\_performance() for an ols() fit doesn’t show adjusted , so I’ll stick with the lm() fit.

## glance results for fit1

* glance(), like tidy(), doesn’t work for ols() fits.

glance(fit1\_lm) |> select(r.squared:df) |>   
 mutate(model = "fit1") |> relocate(model, everything()) |>  
 gt() |> fmt\_number(columns = -df, decimals = 3) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| model | r.squared | adj.r.squared | sigma | statistic | p.value | df |
| --- | --- | --- | --- | --- | --- | --- |
| fit1 | 0.788 | 0.782 | 0.551 | 131.178 | 0.000 | 3 |

glance(fit1\_lm) |> select(logLik:nobs) |>  
 gt() |> fmt\_number(columns = -c(df.residual, nobs), decimals = 3) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- |
| -88.500 | 186.999 | 200.502 | 32.191 | 106 | 110 |

## Model Performance for fit2

model\_performance(fit2\_lm) |>  
 mutate(n = n\_obs(fit2\_lm), df = n - fit2\_lm$df.residual - 1) |>  
 mutate(model = "fit2") |> relocate(model, everything()) |>  
 gt() |> fmt\_number(AIC:Sigma, decimals = 3) |>   
 fmt\_number(n:df, decimals = 0) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| model | AIC | AICc | BIC | R2 | R2\_adjusted | RMSE | Sigma | n | df |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fit2 | 247.883 | 248.460 | 261.385 | 0.631 | 0.620 | 0.713 | 0.727 | 110 | 3 |

* In glance() but not model\_performance()?

glance(fit2\_lm) |> select(statistic, p.value, logLik, deviance, df.residual) |>  
 gt() |> fmt\_number(columns = -df.residual, decimals = 3) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| statistic | p.value | logLik | deviance | df.residual |
| --- | --- | --- | --- | --- |
| 60.401 | 0.000 | -118.941 | 55.990 | 106 |

## Model Performance for fit3

model\_performance(fit3\_lm) |>  
 mutate(n = n\_obs(fit3\_lm), df = n - fit3\_lm$df.residual - 1) |>  
 mutate(model = "fit3") |> relocate(model, everything()) |>  
 gt() |> fmt\_number(AIC:Sigma, decimals = 3) |>   
 fmt\_number(n:df, decimals = 0) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| model | AIC | AICc | BIC | R2 | R2\_adjusted | RMSE | Sigma | n | df |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fit3 | 181.437 | 185.858 | 219.244 | 0.829 | 0.808 | 0.486 | 0.518 | 110 | 12 |

glance(fit3\_lm) |> select(statistic, p.value, logLik, deviance, df.residual) |>  
 gt() |> fmt\_number(columns = -df.residual, decimals = 3) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| statistic | p.value | logLik | deviance | df.residual |
| --- | --- | --- | --- | --- |
| 39.110 | 0.000 | -76.718 | 25.984 | 97 |

## Model Performance for fit4

model\_performance(fit4\_lm) |>  
 mutate(n = n\_obs(fit4\_lm), df = n - fit4\_lm$df.residual - 1) |>  
 mutate(model = "fit4") |> relocate(model, everything()) |>  
 gt() |> fmt\_number(AIC:Sigma, decimals = 3) |>   
 fmt\_number(n:df, decimals = 0) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| model | AIC | AICc | BIC | R2 | R2\_adjusted | RMSE | Sigma | n | df |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fit4 | 175.565 | 181.415 | 218.773 | 0.843 | 0.820 | 0.465 | 0.500 | 110 | 14 |

glance(fit4\_lm) |> select(statistic, p.value, logLik, deviance, df.residual) |>  
 gt() |> fmt\_number(columns = -df.residual, decimals = 3) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| statistic | p.value | logLik | deviance | df.residual |
| --- | --- | --- | --- | --- |
| 36.551 | 0.000 | -71.783 | 23.754 | 95 |

## Model Performance for fit5

model\_performance(fit5\_lm) |>  
 mutate(n = n\_obs(fit5\_lm), df = n - fit5\_lm$df.residual - 1) |>  
 mutate(model = "fit5") |> relocate(model, everything()) |>  
 gt() |> fmt\_number(AIC:Sigma, decimals = 3) |>   
 fmt\_number(n:df, decimals = 0) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| model | AIC | AICc | BIC | R2 | R2\_adjusted | RMSE | Sigma | n | df |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fit5 | 184.770 | 196.403 | 244.181 | 0.847 | 0.813 | 0.459 | 0.510 | 110 | 20 |

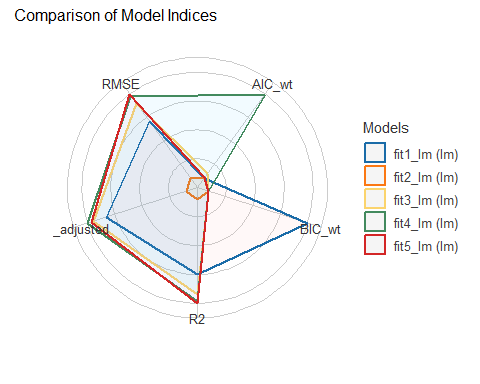
glance(fit5\_lm) |> select(statistic, p.value, logLik, deviance, df.residual) |>  
 gt() |> fmt\_number(columns = -df.residual, decimals = 3) |>  
 tab\_options(table.font.size = 24) |> opt\_stylize(style = 2, color = "blue")

| statistic | p.value | logLik | deviance | df.residual |
| --- | --- | --- | --- | --- |
| 24.701 | 0.000 | -70.385 | 23.158 | 89 |

# Comparing Performance in the Training Sample

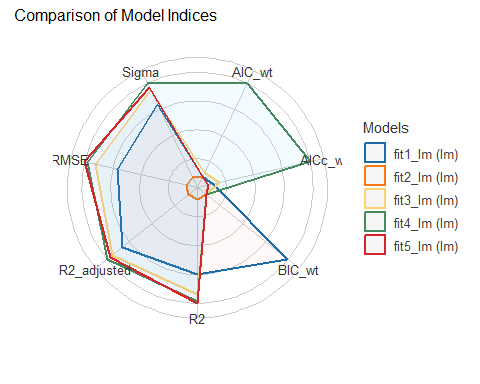
## Comparing across the five models

plot(compare\_performance(fit1\_lm, fit2\_lm, fit3\_lm, fit4\_lm, fit5\_lm,   
 metrics = "common"))



## Comparing across the five models

plot(compare\_performance(fit1\_lm, fit2\_lm, fit3\_lm, fit4\_lm, fit5\_lm,   
 metrics = "all"))



## Comparing across the five models

compare\_performance(fit1\_lm, fit2\_lm, fit3\_lm, fit4\_lm, fit5\_lm,   
 metrics = "common")

# Comparison of Model Performance Indices  
  
Name | Model | AIC (weights) | BIC (weights) | R2 | R2 (adj.) | RMSE  
---------------------------------------------------------------------------  
fit1\_lm | lm | 187.0 (0.003) | 200.5 (>.999) | 0.788 | 0.782 | 0.541  
fit2\_lm | lm | 247.9 (<.001) | 261.4 (<.001) | 0.631 | 0.620 | 0.713  
fit3\_lm | lm | 181.4 (0.050) | 219.2 (<.001) | 0.829 | 0.808 | 0.486  
fit4\_lm | lm | 175.6 (0.938) | 218.8 (<.001) | 0.843 | 0.820 | 0.465  
fit5\_lm | lm | 184.8 (0.009) | 244.2 (<.001) | 0.847 | 0.813 | 0.459

## Use ols() fits instead?

* drops adjusted .

compare\_performance(fit1\_ols, fit2\_ols, fit3\_ols, fit4\_ols, fit5\_ols,   
 metrics = "common")

# Comparison of Model Performance Indices  
  
Name | Model | AIC (weights) | BIC (weights) | R2 | RMSE  
----------------------------------------------------------------  
fit1\_ols | ols | 187.0 (0.003) | 200.5 (>.999) | 0.788 | 0.541  
fit2\_ols | ols | 247.9 (<.001) | 261.4 (<.001) | 0.631 | 0.713  
fit3\_ols | ols | 181.4 (0.050) | 219.2 (<.001) | 0.829 | 0.486  
fit4\_ols | ols | 175.6 (0.938) | 218.8 (<.001) | 0.843 | 0.465  
fit5\_ols | ols | 184.8 (0.009) | 244.2 (<.001) | 0.847 | 0.459

## Error Summaries in Training Sample

Summarize mean absolute error and root mean squared error using the training sample…

train\_err <- tibble(  
 mod = c("fit1", "fit2", "fit3", "fit4", "fit5"),  
 MAE = c(performance\_mae(fit1\_lm), performance\_mae(fit2\_lm),   
 performance\_mae(fit3\_lm), performance\_mae(fit4\_lm),   
 performance\_mae(fit5\_lm)),  
 RMSE = c(performance\_rmse(fit1\_lm), performance\_rmse(fit2\_lm),  
 performance\_rmse(fit3\_lm), performance\_rmse(fit4\_lm),  
 performance\_rmse(fit5\_lm)))  
  
train\_err

# A tibble: 5 × 3  
 mod MAE RMSE  
 <chr> <dbl> <dbl>  
1 fit1 0.412 0.541  
2 fit2 0.543 0.713  
3 fit3 0.371 0.486  
4 fit4 0.344 0.465  
5 fit5 0.342 0.459

# Checking Model Assumptions

## Model fit1

model\_performance(fit1\_lm)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
---------------------------------------------------------------  
186.999 | 187.576 | 200.502 | 0.788 | 0.782 | 0.541 | 0.551

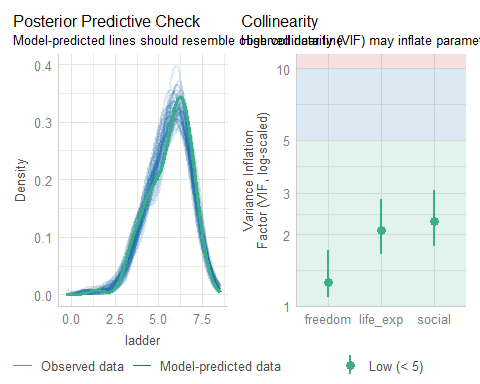
model\_parameters(fit1\_lm)

Parameter | Coefficient | SE | 95% CI | t(106) | p  
-------------------------------------------------------------------  
(Intercept) | -4.41 | 0.68 | [-5.76, -3.05] | -6.46 | < .001  
social | 3.70 | 0.60 | [ 2.51, 4.89] | 6.18 | < .001  
life exp | 0.07 | 0.01 | [ 0.05, 0.10] | 5.26 | < .001  
freedom | 2.98 | 0.48 | [ 2.04, 3.93] | 6.27 | < .001

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

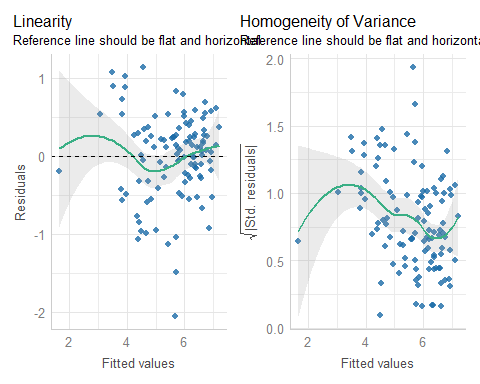
## fit1 plots A and B

check\_model(fit1\_lm, check = c("pp\_check", "vif"))



## fit1 plots C and D

check\_model(fit1\_lm, check = c("linearity", "homogeneity"))



## fit1 additional summaries for B and D

check\_collinearity(fit1\_lm)

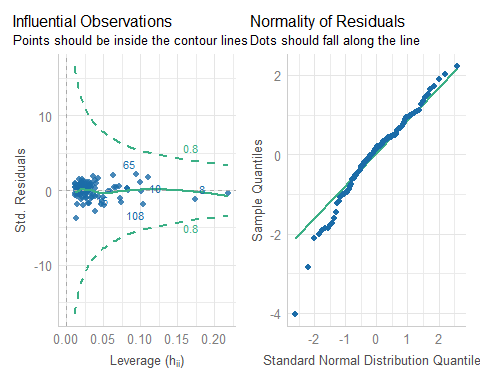
# Check for Multicollinearity  
  
Low Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
 social 2.28 [1.78, 3.09] 1.51 0.44 [0.32, 0.56]  
 life\_exp 2.09 [1.65, 2.82] 1.45 0.48 [0.35, 0.61]  
 freedom 1.26 [1.09, 1.72] 1.12 0.80 [0.58, 0.92]

check\_heteroscedasticity(fit1\_lm)

Warning: Heteroscedasticity (non-constant error variance) detected (p = 0.006).

## fit1 plots E and F

check\_model(fit1\_lm, detrend = FALSE, check = c("qq", "outliers"))



## fit1 additional summaries for E and F

check\_outliers(fit1\_lm)

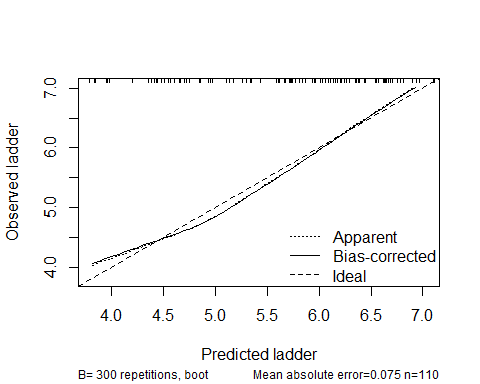
OK: No outliers detected.  
- Based on the following method and threshold: cook (0.845).  
- For variable: (Whole model)

check\_normality(fit1\_lm)

Warning: Non-normality of residuals detected (p = 0.009).

## Calibration Plot for fit1

set.seed(43203); plot(calibrate(fit1\_ols, method = "boot", B = 300))



n=110 Mean absolute error=0.075 Mean squared error=0.00884  
0.9 Quantile of absolute error=0.146

## Calibration Plot Evaluation

* Ideal line shows predicted = observed
* Apparent line shows loess smooth over the observed data
* Bias-Corrected incorporates 300 bootstrapped replications to account for some potential **over-fitting** in estimating the calibration curve

*Overfitting* occurs when the model performs well on training data but poorly on new data, usually because the model is too complex for the data set it is trained on.

## Calibration Plot Summary Statistics

Calling the calibration plot also summarizes:

* mean absolute error,
* mean squared error and
* 90th percentile of absolute error,

where error here refers to the difference between the predicted values and the corresponding bias-corrected calibrated values.

## Model fit2

model\_performance(fit2\_lm)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
---------------------------------------------------------------  
247.883 | 248.460 | 261.385 | 0.631 | 0.620 | 0.713 | 0.727

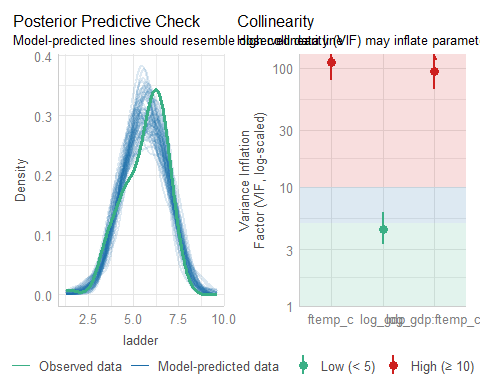
model\_parameters(fit2\_lm)

Parameter | Coefficient | SE | 95% CI | t(106) | p  
--------------------------------------------------------------------------------  
(Intercept) | -5.28 | 1.25 | [-7.76, -2.81] | -4.23 | < .001  
log gdp | 1.11 | 0.12 | [ 0.87, 1.35] | 9.10 | < .001  
ftemp c [warm] | 4.45 | 1.47 | [ 1.53, 7.36] | 3.03 | 0.003   
log gdp × ftemp c [warm] | -0.42 | 0.15 | [-0.72, -0.13] | -2.84 | 0.005

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

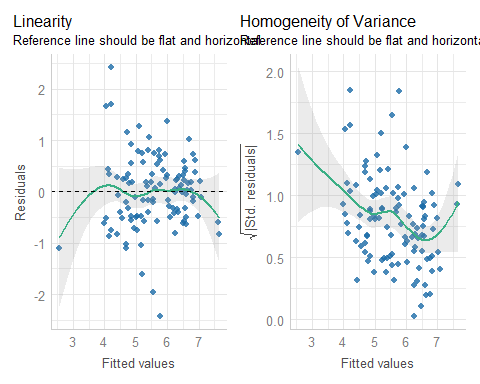
## fit2 plots A and B

check\_model(fit2\_lm, check = c("pp\_check", "vif"))



## fit2 plots C and D

check\_model(fit2\_lm, check = c("linearity", "homogeneity"))



## fit2 additional summaries for B and D

* Remember we have an interaction here, so …

check\_collinearity(fit2\_lm)

Model has interaction terms. VIFs might be inflated.  
 Try to center the variables used for the interaction, or check  
 multicollinearity among predictors of a model without interaction terms.

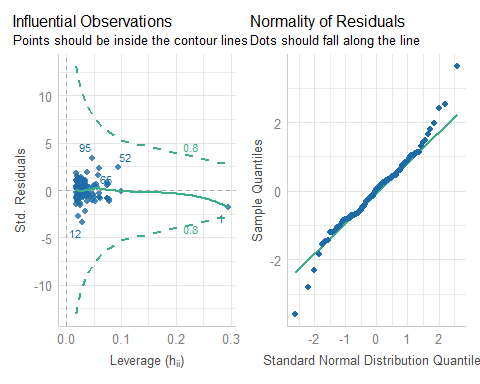
# Check for Multicollinearity  
  
Low Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
 log\_gdp 4.44 [ 3.30, 6.14] 2.11 0.23 [0.16, 0.30]  
  
High Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
 ftemp\_c 112.37 [79.20, 159.60] 10.60 8.90e-03 [0.01, 0.01]  
 log\_gdp:ftemp\_c 93.94 [66.24, 133.41] 9.69 0.01 [0.01, 0.02]

check\_heteroscedasticity(fit2\_lm)

Warning: Heteroscedasticity (non-constant error variance) detected (p < .001).

## fit2 plots E and F

check\_model(fit2\_lm, detrend = FALSE, check = c("qq", "outliers"))



## fit2 additional summaries for E and F

check\_outliers(fit2\_lm)

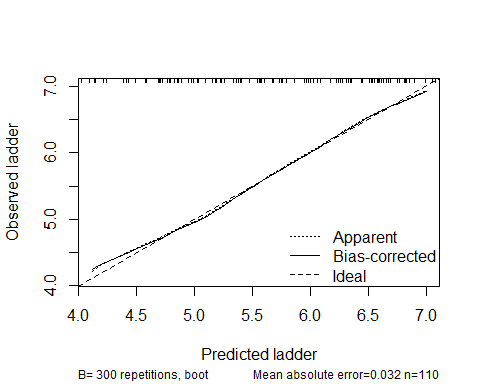
OK: No outliers detected.  
- Based on the following method and threshold: cook (0.7).  
- For variable: (Whole model)

check\_normality(fit2\_lm)

OK: residuals appear as normally distributed (p = 0.057).

## Calibration Plot for fit2

set.seed(43204); plot(calibrate(fit2\_ols, method = "boot", B = 300))



n=110 Mean absolute error=0.032 Mean squared error=0.00201  
0.9 Quantile of absolute error=0.058

## Model fit3 performance

model\_performance(fit3\_lm)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
---------------------------------------------------------------  
181.437 | 185.858 | 219.244 | 0.829 | 0.808 | 0.486 | 0.518

## Model fit3 parameters

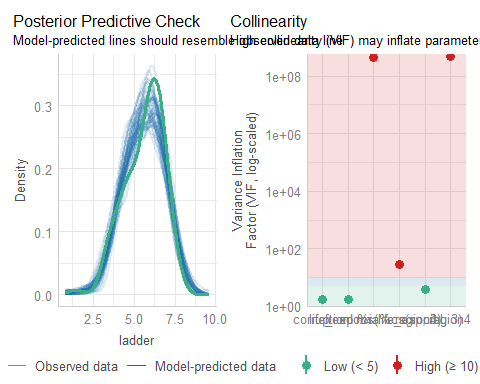
model\_parameters(fit3\_lm)

Parameter | Coefficient | SE  
-------------------------------------------------------------------  
(Intercept) | -3.61 | 3.01  
rcs(social [ degree] | 3.41 | 0.84  
rcs(social [ degree] | 0.38 | 0.94  
rcs(life exp [ degree] | 0.08 | 0.05  
rcs(life exp [ degree] | -0.01 | 0.05  
region4 [Europe] | -0.53 | 3.98  
region4 [Africa] | 1.94 | 3.11  
region4 [Other] | 1.61 | 3.90  
life exp %ia% region4life exp \* region4=Europe | 0.01 | 0.06  
life exp %ia% region4life exp \* region4=Africa | -0.03 | 0.05  
life exp %ia% region4life exp \* region4=Other | -0.02 | 0.06  
freedom | 2.39 | 0.53  
corruption | -1.14 | 0.38  
  
Parameter | 95% CI | t(97) | p  
--------------------------------------------------------------------------------  
(Intercept) | [-9.57, 2.36] | -1.20 | 0.233   
rcs(social [ degree] | [ 1.74, 5.08] | 4.04 | < .001  
rcs(social [ degree] | [-1.48, 2.24] | 0.40 | 0.688   
rcs(life exp [ degree] | [-0.02, 0.18] | 1.54 | 0.127   
rcs(life exp [ degree] | [-0.10, 0.08] | -0.31 | 0.761   
region4 [Europe] | [-8.43, 7.37] | -0.13 | 0.894   
region4 [Africa] | [-4.23, 8.11] | 0.62 | 0.534   
region4 [Other] | [-6.14, 9.35] | 0.41 | 0.682   
life exp %ia% region4life exp \* region4=Europe | [-0.10, 0.13] | 0.18 | 0.855   
life exp %ia% region4life exp \* region4=Africa | [-0.13, 0.07] | -0.55 | 0.583   
life exp %ia% region4life exp \* region4=Other | [-0.13, 0.10] | -0.29 | 0.774   
freedom | [ 1.35, 3.44] | 4.55 | < .001  
corruption | [-1.89, -0.38] | -2.97 | 0.004

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

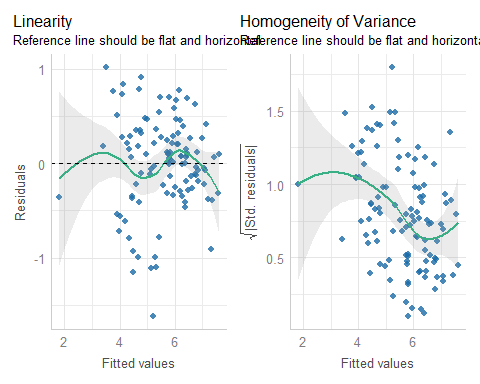
## fit3 plots A and B

check\_model(fit3\_lm, check = c("pp\_check", "vif"))



## fit3 plots C and D

check\_model(fit3\_lm, check = c("linearity", "homogeneity"))



## fit3 additional summaries for B and D

check\_collinearity(fit3\_lm)

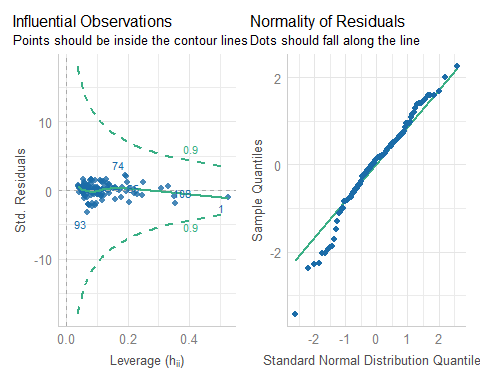
# Check for Multicollinearity  
  
Low Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance  
 rcs(social, 3) 3.71 [ 2.85, 4.97] 1.93 0.27  
 freedom 1.74 [ 1.43, 2.27] 1.32 0.57  
 corruption 1.74 [ 1.43, 2.26] 1.32 0.58  
 Tolerance 95% CI  
 [0.20, 0.35]  
 [0.44, 0.70]  
 [0.44, 0.70]  
  
High Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance  
 rcs(life\_exp, 3) 29.56 [ 21.58, 40.62] 5.44 0.03  
 region4 4.91e+08 [3.56e+08, 6.77e+08] 22156.75 2.04e-09  
 life\_exp %ia% region4 4.61e+08 [3.34e+08, 6.36e+08] 21466.22 2.17e-09  
 Tolerance 95% CI  
 [0.02, 0.05]  
 [0.00, 0.00]  
 [0.00, 0.00]

check\_heteroscedasticity(fit3\_lm)

Warning: Heteroscedasticity (non-constant error variance) detected (p < .001).

## fit3 plots E and F

check\_model(fit3\_lm, detrend = FALSE, check = c("qq", "outliers"))



## fit3 additional summaries for E and F

check\_outliers(fit3\_lm)

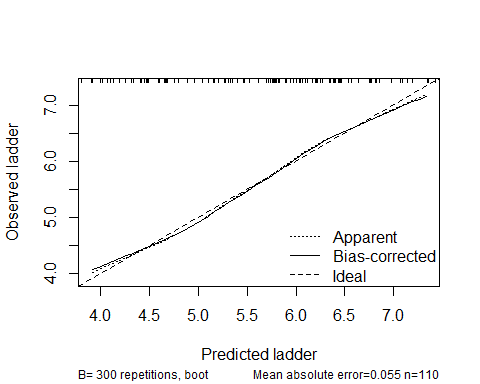
OK: No outliers detected.  
- Based on the following method and threshold: cook (0.9).  
- For variable: (Whole model)

check\_normality(fit3\_lm)

Warning: Non-normality of residuals detected (p = 0.037).

## Calibration Plot for fit3

set.seed(43205); plot(calibrate(fit3\_ols, method = "boot", B = 300))



n=110 Mean absolute error=0.055 Mean squared error=0.00435  
0.9 Quantile of absolute error=0.101

## Model fit4 performance

model\_performance(fit4\_lm)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
---------------------------------------------------------------  
175.565 | 181.415 | 218.773 | 0.843 | 0.820 | 0.465 | 0.500

## Model fit4 parameters

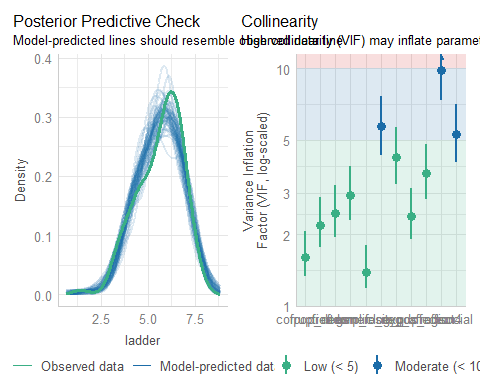
model\_parameters(fit4\_lm)

Parameter | Coefficient | SE | 95% CI | t(95) | p  
-----------------------------------------------------------------------  
(Intercept) | -3.23 | 1.36 | [-5.92, -0.53] | -2.37 | 0.020   
log gdp | 0.18 | 0.08 | [ 0.02, 0.35] | 2.24 | 0.027   
social | 3.26 | 0.83 | [ 1.62, 4.91] | 3.94 | < .001  
life exp | 0.04 | 0.02 | [ 0.00, 0.08] | 2.11 | 0.037   
freedom | 1.78 | 0.60 | [ 0.58, 2.98] | 2.95 | 0.004   
generosity | -0.09 | 0.34 | [-0.76, 0.58] | -0.27 | 0.786   
corruption | -0.93 | 0.35 | [-1.64, -0.23] | -2.63 | 0.010   
pos affect | 0.81 | 0.82 | [-0.81, 2.43] | 0.99 | 0.323   
neg affect | 0.37 | 0.82 | [-1.26, 2.01] | 0.45 | 0.653   
ftemp c [warm] | 0.22 | 0.16 | [-0.11, 0.54] | 1.33 | 0.188   
fpop dens [med] | -0.09 | 0.14 | [-0.37, 0.20] | -0.61 | 0.540   
fpop dens [high] | 2.31e-03 | 0.15 | [-0.30, 0.30] | 0.02 | 0.988   
region4 [Europe] | 0.33 | 0.17 | [-0.01, 0.68] | 1.92 | 0.057   
region4 [Africa] | 0.15 | 0.18 | [-0.22, 0.51] | 0.81 | 0.420   
region4 [Other] | 0.41 | 0.20 | [ 0.01, 0.81] | 2.04 | 0.044

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

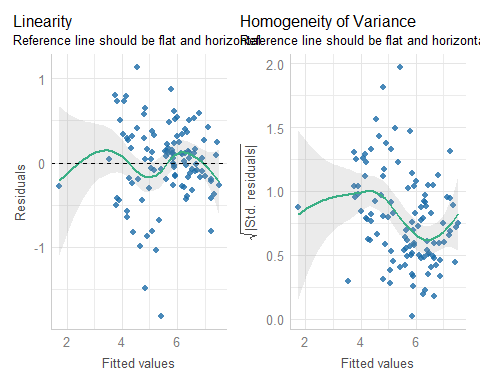
## fit4 plots A and B

check\_model(fit4\_lm, check = c("pp\_check", "vif"))



## fit4 plots C and D

check\_model(fit4\_lm, check = c("linearity", "homogeneity"))



## fit4 additional summaries for B and D

check\_collinearity(fit4\_lm)

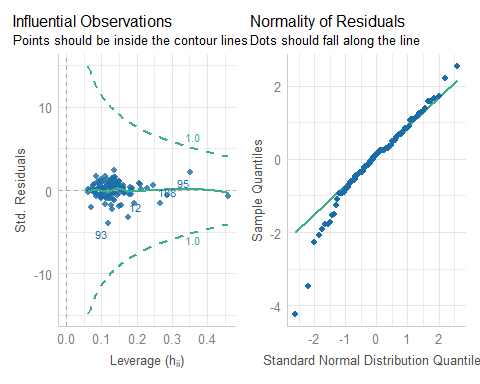
# Check for Multicollinearity  
  
Low Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
 log\_gdp 4.25 [3.26, 5.68] 2.06 0.24 [0.18, 0.31]  
 freedom 2.46 [1.95, 3.22] 1.57 0.41 [0.31, 0.51]  
 generosity 1.39 [1.19, 1.81] 1.18 0.72 [0.55, 0.84]  
 corruption 1.60 [1.34, 2.07] 1.27 0.62 [0.48, 0.75]  
 pos\_affect 3.63 [2.81, 4.83] 1.91 0.28 [0.21, 0.36]  
 neg\_affect 2.39 [1.91, 3.14] 1.55 0.42 [0.32, 0.52]  
 ftemp\_c 2.93 [2.30, 3.87] 1.71 0.34 [0.26, 0.44]  
 fpop\_dens 2.20 [1.77, 2.87] 1.48 0.45 [0.35, 0.57]  
  
Moderate Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
 social 5.30 [4.03, 7.11] 2.30 0.19 [0.14, 0.25]  
 life\_exp 5.72 [4.33, 7.69] 2.39 0.17 [0.13, 0.23]  
 region4 9.90 [7.38, 13.41] 3.15 0.10 [0.07, 0.14]

check\_heteroscedasticity(fit4\_lm)

Warning: Heteroscedasticity (non-constant error variance) detected (p = 0.002).

## fit4 plots E and F

check\_model(fit4\_lm, detrend = FALSE, check = c("qq", "outliers"))



## fit4 additional summaries for E and F

check\_outliers(fit4\_lm)

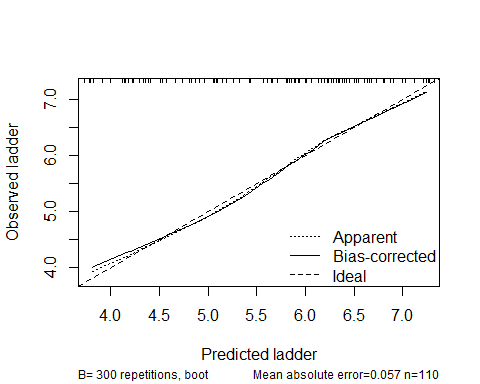
OK: No outliers detected.  
- Based on the following method and threshold: cook (0.933).  
- For variable: (Whole model)

check\_normality(fit4\_lm)

Warning: Non-normality of residuals detected (p = 0.002).

## Calibration Plot for fit4

set.seed(43206); plot(calibrate(fit4\_ols, method = "boot", B = 300))



n=110 Mean absolute error=0.057 Mean squared error=0.00508  
0.9 Quantile of absolute error=0.103

## Model fit5 performance

model\_performance(fit5\_lm)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
---------------------------------------------------------------  
184.770 | 196.403 | 244.181 | 0.847 | 0.813 | 0.459 | 0.510

## Model fit5 parameters

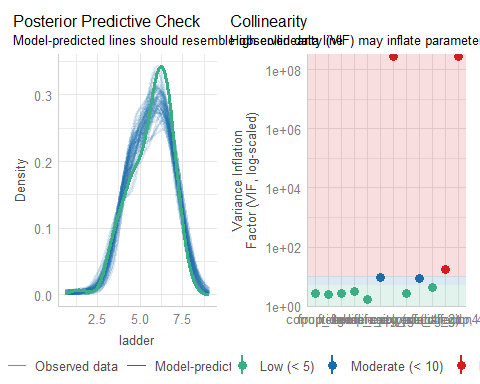
model\_parameters(fit5\_lm)

Parameter | Coefficient | SE | 95% CI | t(89) | p  
----------------------------------------------------------------------------------  
(Intercept) | -0.40 | 2.36 | [-5.10, 4.30] | -0.17 | 0.865   
rcs(log gdp [ degree] | 0.01 | 0.21 | [-0.41, 0.44] | 0.07 | 0.944   
rcs(log gdp [ degree] | 0.31 | 0.39 | [-0.47, 1.09] | 0.79 | 0.434   
rcs(log gdp [ degree] | -1.15 | 2.18 | [-5.49, 3.18] | -0.53 | 0.598   
social [1st degree] | 4.50 | 1.29 | [ 1.93, 7.07] | 3.48 | < .001  
social [2nd degree] | 0.35 | 0.64 | [-0.94, 1.63] | 0.54 | 0.593   
life exp | 0.05 | 0.03 | [ 0.00, 0.11] | 2.03 | 0.046   
region4 [Europe] | 2.33 | 3.75 | [-5.12, 9.78] | 0.62 | 0.536   
region4 [Africa] | 1.16 | 2.40 | [-3.61, 5.92] | 0.48 | 0.631   
region4 [Other] | 3.50 | 3.82 | [-4.08, 11.08] | 0.92 | 0.362   
freedom | 1.87 | 0.65 | [ 0.58, 3.16] | 2.87 | 0.005   
generosity | -0.13 | 0.38 | [-0.88, 0.61] | -0.35 | 0.726   
corruption | -0.81 | 0.46 | [-1.73, 0.11] | -1.75 | 0.083   
pos affect | 1.02 | 0.91 | [-0.79, 2.83] | 1.12 | 0.264   
neg affect | 0.62 | 0.89 | [-1.15, 2.39] | 0.69 | 0.490   
ftemp c [warm] | 0.21 | 0.17 | [-0.13, 0.55] | 1.22 | 0.226   
fpop dens [med] | -0.05 | 0.15 | [-0.35, 0.25] | -0.34 | 0.738   
fpop dens [high] | 0.02 | 0.16 | [-0.29, 0.34] | 0.16 | 0.877   
life exp × region4 [Europe] | -0.03 | 0.05 | [-0.14, 0.08] | -0.56 | 0.576   
life exp × region4 [Africa] | -0.02 | 0.04 | [-0.09, 0.06] | -0.41 | 0.680   
life exp × region4 [Other] | -0.05 | 0.06 | [-0.16, 0.07] | -0.82 | 0.415

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

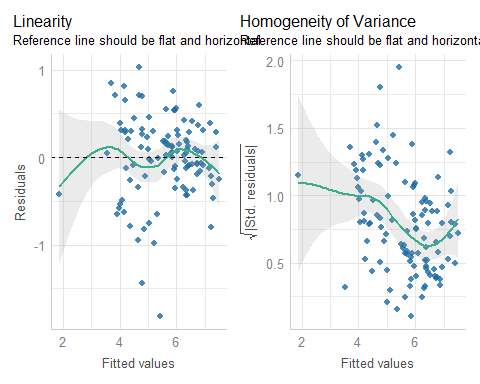
## fit5 plots A and B

check\_model(fit5\_lm, check = c("pp\_check", "vif"))



## fit5 plots C and D

check\_model(fit5\_lm, check = c("linearity", "homogeneity"))



## fit5 additional summaries for B and D

check\_collinearity(fit5\_lm)

Model has interaction terms. VIFs might be inflated.  
 Try to center the variables used for the interaction, or check  
 multicollinearity among predictors of a model without interaction terms.

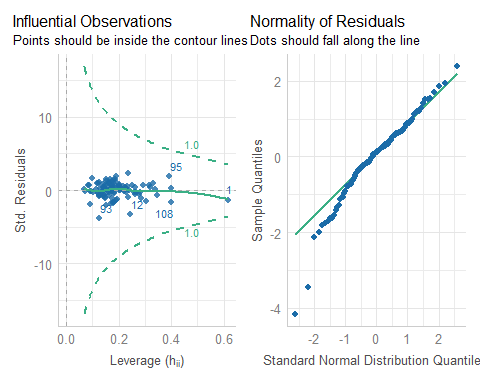
# Check for Multicollinearity  
  
Low Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
 freedom 2.74 [ 2.19, 3.54] 1.66 0.37 [0.28, 0.46]  
 generosity 1.65 [ 1.38, 2.09] 1.28 0.61 [0.48, 0.72]  
 corruption 2.63 [ 2.11, 3.40] 1.62 0.38 [0.29, 0.47]  
 pos\_affect 4.36 [ 3.39, 5.72] 2.09 0.23 [0.17, 0.29]  
 neg\_affect 2.69 [ 2.16, 3.48] 1.64 0.37 [0.29, 0.46]  
 ftemp\_c 3.15 [ 2.49, 4.09] 1.77 0.32 [0.24, 0.40]  
 fpop\_dens 2.48 [ 1.99, 3.19] 1.57 0.40 [0.31, 0.50]  
  
Moderate Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance  
 poly(social, 2) 9.01 [ 6.85, 11.96] 3.00 0.11  
 life\_exp 9.27 [ 7.05, 12.31] 3.05 0.11  
 Tolerance 95% CI  
 [0.08, 0.15]  
 [0.08, 0.14]  
  
High Correlation  
  
 Term VIF VIF 95% CI Increased SE Tolerance  
 rcs(log\_gdp, 4) 17.45 [ 13.14, 23.30] 4.18 0.06  
 region4 2.80e+08 [2.08e+08, 3.76e+08] 16725.31 3.57e-09  
 life\_exp:region4 2.67e+08 [1.99e+08, 3.58e+08] 16336.13 3.75e-09  
 Tolerance 95% CI  
 [0.04, 0.08]  
 [0.00, 0.00]  
 [0.00, 0.00]

check\_heteroscedasticity(fit5\_lm)

Warning: Heteroscedasticity (non-constant error variance) detected (p < .001).

## fit5 plots E and F

check\_model(fit5\_lm, detrend = FALSE, check = c("qq", "outliers"))



## fit5 additional summaries for E and F

check\_outliers(fit5\_lm)

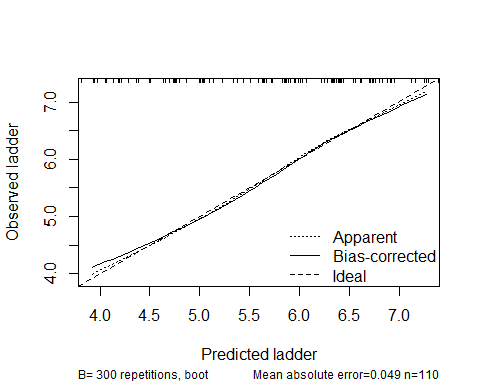
OK: No outliers detected.  
- Based on the following method and threshold: cook (0.933).  
- For variable: (Whole model)

check\_normality(fit5\_lm)

Warning: Non-normality of residuals detected (p = 0.004).

## Calibration Plot for fit5

set.seed(43207); plot(calibrate(fit5\_ols, method = "boot", B = 300))



n=110 Mean absolute error=0.049 Mean squared error=0.00434  
0.9 Quantile of absolute error=0.122

# Making Predictions

## Status of the USA

* USA is in the training sample

usadat <- happy\_si\_train |> filter(iso3 == "USA") |>  
 select(iso3, country, ladder, social, life\_exp, freedom, everything())  
  
usadat

# A tibble: 1 × 19  
 iso3 country ladder social life\_exp freedom log\_gdp generosity corruption  
 <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 USA United Sta… 6.52 0.861 65.6 0.721 11.1 0.185 0.722  
# ℹ 10 more variables: pos\_affect <dbl>, neg\_affect <dbl>, region <fct>,  
# temp\_c <dbl>, pop\_dens <dbl>, region4 <fct>, ftemp\_c <fct>,  
# fpop\_dens <fct>, .row\_id <int>, ladder2 <dbl>

## USA fit1 Predictions (90% PI)

* Using the lm() fit, and a prediction interval…

predict(fit1\_lm, newdata = usadat, interval = "confidence", level = 0.90)

fit lwr upr  
1 5.69303 5.559333 5.826727

* Using the ols() fit:

predict(fit1\_ols, newdata = usadat,   
 conf.int = 0.90, conf.type = "individual") |>  
 as\_vector()

linear.predictors.1 lower.1 upper.1   
 5.693030 4.768871 6.617189

## USA fit1 Predictions (90% CI)

* Using the lm() fit, and a confidence interval for a mean…

predict(fit1\_lm, newdata = usadat, interval = "confidence", level = 0.90)

fit lwr upr  
1 5.69303 5.559333 5.826727

* Using the ols() fit:

predict(fit1\_ols, newdata = usadat,   
 conf.int = 0.90, conf.type = "mean") |>  
 as\_vector()

linear.predictors.1 lower.1 upper.1   
 5.693030 5.559333 5.826727

## Predictions for USA by each fit

* Recall actual USA ladder is 6.52

predict(fit1\_lm, newdata = usadat, interval = "confidence", level = 0.90)

fit lwr upr  
1 5.69303 5.559333 5.826727

predict(fit2\_lm, newdata = usadat, interval = "confidence", level = 0.90)

fit lwr upr  
1 6.996539 6.757462 7.235617

predict(fit3\_lm, newdata = usadat, interval = "confidence", level = 0.90)

fit lwr upr  
1 6.052758 5.75337 6.352145

predict(fit4\_lm, newdata = usadat, interval = "confidence", level = 0.90)

fit lwr upr  
1 6.180744 5.800826 6.560663

predict(fit5\_lm, newdata = usadat, interval = "confidence", level = 0.90)

fit lwr upr  
1 6.324496 5.844953 6.804038

## Using augment() after a fit

fit1\_lm\_aug <- augment(fit1\_lm)  
  
names(fit1\_lm\_aug)

[1] "ladder" "social" "life\_exp" "freedom" ".fitted"   
 [6] ".resid" ".hat" ".sigma" ".cooksd" ".std.resid"

* augment stores .fitted: predicted values, .resid: residuals and
* .hat = leverage values (diagonal of the hat matrix)
* .sigma = estimated sigma when this observation is dropped from model
* .cooksd = Cook’s distance
* .std.resid = standardized residuals
* Details at the broom site on [augment for a linear model](https://broom.tidymodels.org/reference/augment.lm.html).
* augment() doesn’t work with ols() fits.

## Using augment() in our training sample

fit1\_aug <- augment(fit1\_lm) |> mutate(mod = "fit1") |> relocate(mod)  
fit2\_aug <- augment(fit2\_lm) |> mutate(mod = "fit2") |> relocate(mod)  
fit3\_aug <- augment(fit3\_lm) |> mutate(mod = "fit3") |> relocate(mod)  
fit4\_aug <- augment(fit4\_lm) |> mutate(mod = "fit4") |> relocate(mod)  
fit5\_aug <- augment(fit5\_lm) |> mutate(mod = "fit5") |> relocate(mod)

Sample results: first two rows of fit4\_aug…

fit4\_aug |> head(2)

# A tibble: 2 × 19  
 mod ladder log\_gdp social life\_exp freedom generosity corruption pos\_affect  
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 fit4 1.45 7.08 0.368 55.2 0.228 -0.179 0.738 0.261  
2 fit4 6.39 9.99 0.892 67.3 0.832 -0.129 0.846 0.720  
# ℹ 10 more variables: neg\_affect <dbl>, ftemp\_c <fct>, fpop\_dens <fct>,  
# region4 <fct>, .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>,  
# .cooksd <dbl>, .std.resid <dbl>

# Using our Test Sample

## Predicting fit1 into a Test Sample

test\_1 <- augment(fit1\_lm, newdata = happy\_si\_test)  
  
test\_1\_res <- test\_1 |>   
 summarise(MAPE = mean(abs(.resid)),  
 maxAPE = max(abs(.resid)),  
 RMSPE = sqrt(mean(.resid^2)),  
 rsqr = cor(ladder, .fitted)^2)   
  
test\_1\_res |> gt() |> tab\_options(table.font.size = 24) |>  
 fmt\_number(decimals = 3) |> opt\_stylize(style = 5, color = "cyan")

| MAPE | maxAPE | RMSPE | rsqr |
| --- | --- | --- | --- |
| 0.388 | 1.122 | 0.509 | 0.744 |

## Predicting 5 models in Test Sample

test\_1 <- augment(fit1\_lm, newdata = happy\_si\_test) |> mutate(mod = "fit1")  
test\_2 <- augment(fit2\_lm, newdata = happy\_si\_test) |> mutate(mod = "fit2")  
test\_3 <- augment(fit3\_lm, newdata = happy\_si\_test) |> mutate(mod = "fit3")  
test\_4 <- augment(fit4\_lm, newdata = happy\_si\_test) |> mutate(mod = "fit4")  
test\_5 <- augment(fit5\_lm, newdata = happy\_si\_test) |> mutate(mod = "fit5")  
  
test\_res <- bind\_rows(test\_1, test\_2, test\_3, test\_4, test\_5) |>   
 relocate(mod, .fitted, ladder, .resid, everything())  
  
test\_res |> head(3)

# A tibble: 3 × 21  
 mod .fitted ladder .resid iso3 country log\_gdp social life\_exp freedom  
 <chr> <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
1 fit1 5.77 5.44 -0.330 ALB Albania 9.69 0.691 69.2 0.872  
2 fit1 6.27 6.01 -0.258 BIH Bosnia and … 9.76 0.879 67.4 0.847  
3 fit1 3.97 4.46 0.492 BFA Burkina Faso 7.69 0.580 56.4 0.715  
# ℹ 11 more variables: generosity <dbl>, corruption <dbl>, pos\_affect <dbl>,  
# neg\_affect <dbl>, region <fct>, temp\_c <dbl>, pop\_dens <dbl>,  
# region4 <fct>, ftemp\_c <fct>, fpop\_dens <fct>, .row\_id <int>

## Error Summaries in Test Sample

test\_summ <- test\_res |>   
 group\_by(mod) |>  
 summarise(MAPE = mean(abs(.resid)),  
 maxAPE = max(abs(.resid)),  
 RMSPE = sqrt(mean(.resid^2)),  
 rsqr = cor(ladder, .fitted)^2)   
  
test\_summ |> gt() |> tab\_options(table.font.size = 24) |>  
 fmt\_number(decimals = 3) |> opt\_stylize(style = 5, color = "cyan")

| mod | MAPE | maxAPE | RMSPE | rsqr |
| --- | --- | --- | --- | --- |
| fit1 | 0.388 | 1.122 | 0.509 | 0.744 |
| fit2 | 0.812 | 3.518 | 1.235 | 0.142 |
| fit3 | 0.388 | 0.896 | 0.458 | 0.792 |
| fit4 | 0.388 | 1.003 | 0.475 | 0.770 |
| fit5 | 0.370 | 0.985 | 0.469 | 0.777 |

# Validated and MSE for our Five Models

## Validated Summaries for fit1

set.seed(43208)  
validate(fit1\_ols, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.7878 0.7913 0.7790 0.0123 0.7755 300  
MSE 0.2926 0.2810 0.3047 -0.0237 0.3164 300  
g 1.1611 1.1545 1.1574 -0.0029 1.1640 300  
Intercept 0.0000 0.0000 0.0379 -0.0379 0.0379 300  
Slope 1.0000 1.0000 0.9932 0.0068 0.9932 300

## Validated Summaries for fit2

set.seed(43209)  
validate(fit2\_ols, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.6309 0.6369 0.6136 0.0233 0.6077 300  
MSE 0.5090 0.4900 0.5329 -0.0429 0.5519 300  
g 1.0685 1.0600 1.0590 0.0009 1.0676 300  
Intercept 0.0000 0.0000 0.0091 -0.0091 0.0091 300  
Slope 1.0000 1.0000 0.9979 0.0021 0.9979 300

## Validated Summaries for fit3

set.seed(43210)  
validate(fit3\_ols, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.8287 0.8458 0.8025 0.0433 0.7855 300  
MSE 0.2362 0.2101 0.2723 -0.0622 0.2985 300  
g 1.2149 1.2202 1.1973 0.0230 1.1919 300  
Intercept 0.0000 0.0000 0.1074 -0.1074 0.1074 300  
Slope 1.0000 1.0000 0.9801 0.0199 0.9801 300

## Validated Summaries for fit4

set.seed(43211)  
validate(fit4\_ols, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.8434 0.8635 0.8142 0.0493 0.7941 300  
MSE 0.2159 0.1816 0.2562 -0.0746 0.2905 300  
g 1.2252 1.2194 1.2063 0.0131 1.2121 300  
Intercept 0.0000 0.0000 0.0729 -0.0729 0.0729 300  
Slope 1.0000 1.0000 0.9856 0.0144 0.9856 300

## Validated Summaries for fit5

set.seed(43212)  
validate(fit5\_ols, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.8473 0.8756 0.7963 0.0793 0.7680 300  
MSE 0.2105 0.1695 0.2809 -0.1114 0.3219 300  
g 1.2330 1.2461 1.2000 0.0461 1.1869 300  
Intercept 0.0000 0.0000 0.2383 -0.2383 0.2383 300  
Slope 1.0000 1.0000 0.9575 0.0425 0.9575 300

# Other Means of Validation

## Cross-Validation via Holdout Sample

set.seed(43213)  
performance\_cv(fit1\_lm, method = "holdout", metrics = "all", prop = 0.3)

# Cross-validation performance (30% holdout method)  
  
MSE | RMSE | R2  
-----------------  
0.35 | 0.59 | 0.6

performance\_cv(fit2\_lm, method = "holdout", metrics = "all", prop = 0.3)

# Cross-validation performance (30% holdout method)  
  
MSE | RMSE | R2  
------------------  
0.51 | 0.71 | 0.53

performance\_cv(fit3\_lm, method = "holdout", metrics = "all", prop = 0.3)

# Cross-validation performance (30% holdout method)  
  
MSE | RMSE | R2  
------------------  
0.81 | 0.9 | 0.56

## Cross-Validation continues

set.seed(43213)  
performance\_cv(fit4\_lm, method = "holdout", metrics = "all", prop = 0.3)

# Cross-validation performance (30% holdout method)  
  
MSE | RMSE | R2  
------------------  
0.29 | 0.54 | 0.67

performance\_cv(fit5\_lm, method = "holdout", metrics = "all", prop = 0.3)

# Cross-validation performance (30% holdout method)  
  
MSE | RMSE | R2  
------------------  
0.23 | 0.48 | 0.79

## 5-fold Cross-Validation

set.seed(43214)  
performance\_cv(fit1\_lm, method = "k\_fold", metrics = "all", k = 5)

# Cross-validation performance (5-fold method)  
  
MSE | RMSE | R2  
------------------  
0.31 | 0.56 | 0.77

performance\_cv(fit2\_lm, method = "k\_fold", metrics = "all", k = 5)

# Cross-validation performance (5-fold method)  
  
MSE | RMSE | R2  
------------------  
0.53 | 0.73 | 0.61

performance\_cv(fit3\_lm, method = "k\_fold", metrics = "all", k = 5)

# Cross-validation performance (5-fold method)  
  
MSE | RMSE | R2  
------------------  
0.31 | 0.56 | 0.78

## 5-fold Cross-Validation continues

set.seed(43214)  
performance\_cv(fit4\_lm, method = "k\_fold", metrics = "all", k = 5)

# Cross-validation performance (5-fold method)  
  
MSE | RMSE | R2  
------------------  
0.29 | 0.53 | 0.79

performance\_cv(fit5\_lm, method = "k\_fold", metrics = "all", k = 5)

# Cross-validation performance (5-fold method)  
  
MSE | RMSE | R2  
-----------------  
0.4 | 0.63 | 0.71

## Accuracy and cross-validation

set.seed(43215)  
performance\_accuracy(fit1\_lm, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 86.28% [81.49%, 91.46%]  
Method: Correlation between observed and predicted

performance\_accuracy(fit2\_lm, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 77.87% [69.51%, 87.24%]  
Method: Correlation between observed and predicted

performance\_accuracy(fit3\_lm, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 87.71% [83.66%, 92.38%]  
Method: Correlation between observed and predicted

## Accuracy and c-v, continued

set.seed(43215)  
performance\_accuracy(fit4\_lm, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 85.39% [76.62%, 94.16%]  
Method: Correlation between observed and predicted

performance\_accuracy(fit5\_lm, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 87.23% [85.81%, 87.94%]  
Method: Correlation between observed and predicted

# Multiple Imputation with mice

## Our five models

| Model | ladder is predicted using… | df |
| --- | --- | --- |
| fit1 | social + life\_exp + freedom | 3 |
| fit2 | log\_gdp \* ftemp\_c | 3 |
| fit3 | rcs(social,3) + rcs(life\_exp, 3) + region4 + life\_exp %ia% region4 + freedom + corruption | 12 |
| fit4 | log\_gdp + social + life\_exp + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens + region4 | 14 |
| fit5 | rcs(log\_gdp,4) + poly(social,2) + life\_exp \* region4 + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens | 20 |

## Build 15 imputations, estimate fit1

Why are we using 15 imputations?

prop\_miss\_case(happy)

[1] 0.1304348

Let’s run fit1…

fit1\_imp\_ests <-   
 mice(happy, m = 15, seed = 43216, print = FALSE) |>  
 with(lm(ladder ~ social + life\_exp + freedom)) |>  
 pool()

Warning: Number of logged events: 654

## What’s in fit1\_imp\_ests?

fit1\_imp\_ests

Class: mipo m = 15   
 term m estimate ubar b t dfcom  
1 (Intercept) 15 -4.03834630 0.3450289625 2.793143e-02 0.3748224913 134  
2 social 15 3.62292244 0.2821454138 2.500557e-02 0.3088180231 134  
3 life\_exp 15 0.06699189 0.0001399771 2.081341e-05 0.0001621781 134  
4 freedom 15 3.06094350 0.1906831411 1.836780e-03 0.1926423729 134  
 df riv lambda fmi  
1 115.22729 0.08635081 0.07948704 0.09505896  
2 113.35269 0.09453497 0.08636999 0.10207448  
3 98.88326 0.15860432 0.13689256 0.15383563  
4 130.57478 0.01027480 0.01017031 0.02499091

## Across 15 imputations: fit1 estimates

glance(fit1\_imp\_ests)

nimp nobs r.squared adj.r.squared  
1 15 138 0.7749567 0.7699182

model\_parameters(fit1\_imp\_ests, ci = 0.90)

Warning: Number of logged events: 654  
Warning: Number of logged events: 654  
Warning: Number of logged events: 654  
Warning: Number of logged events: 654

# Fixed Effects  
  
Parameter | Coefficient | SE | 90% CI | t | df | p  
---------------------------------------------------------------------------  
(Intercept) | -4.04 | 0.61 | [-5.05, -3.02] | -6.60 | 115.23 | < .001  
social | 3.62 | 0.56 | [ 2.70, 4.54] | 6.52 | 113.35 | < .001  
life exp | 0.07 | 0.01 | [ 0.05, 0.09] | 5.26 | 98.88 | < .001  
freedom | 3.06 | 0.44 | [ 2.33, 3.79] | 6.97 | 130.57 | < .001

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

## Build 15 imputations, estimate fit2

fit2\_imp\_ests <-   
 mice(happy, m = 15, seed = 43217, print = FALSE) |>  
 with(lm(ladder ~ log\_gdp \* ftemp\_c)) |>  
 pool()

Warning: Number of logged events: 650

## Across 15 imputations: fit2 estimates

glance(fit2\_imp\_ests)

nimp nobs r.squared adj.r.squared  
1 15 138 0.5689855 0.5593

model\_parameters(fit2\_imp\_ests, ci = 0.90)

Warning: Number of logged events: 650  
Warning: Number of logged events: 650  
Warning: Number of logged events: 650  
Warning: Number of logged events: 650

# Fixed Effects  
  
Parameter | Coefficient | SE | 90% CI | t | df | p  
---------------------------------------------------------------------------------------  
(Intercept) | -2.94 | 1.54 | [-5.54, -0.33] | -1.90 | 37.87 | 0.065   
log gdp | 0.89 | 0.15 | [ 0.63, 1.14] | 5.91 | 39.51 | < .001  
ftemp c [warm] | 2.07 | 1.56 | [-0.53, 4.67] | 1.33 | 76.77 | 0.189   
log gdp × ftemp c [warm] | -0.20 | 0.16 | [-0.46, 0.06] | -1.29 | 83.50 | 0.202

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

## Build 15 imputations, estimate fit3

fit3\_imp\_ests <-   
 mice(happy, m = 15, seed = 43218, print = FALSE) |>  
 with(lm(ladder ~ rcs(social,3) + rcs(life\_exp, 3) + region4 +   
 life\_exp %ia% region4 + freedom + corruption)) |>  
 pool()

Warning: Number of logged events: 642

## Across 15 imputations: fit2 estimates

glance(fit3\_imp\_ests)

nimp nobs r.squared adj.r.squared  
1 15 138 0.8130362 0.7950858

model\_parameters(fit3\_imp\_ests, ci = 0.90)

Warning: Number of logged events: 642  
Warning: Number of logged events: 642  
Warning: Number of logged events: 642  
Warning: Number of logged events: 642

# Fixed Effects  
  
Parameter | Coefficient | SE  
-------------------------------------------------------------------  
(Intercept) | -1.68 | 3.70  
rcs(social [ degree] | 3.23 | 0.81  
rcs(social [ degree] | 0.30 | 0.89  
rcs(life exp [ degree] | 0.04 | 0.06  
rcs(life exp [ degree] | 0.02 | 0.05  
region4 [Europe] | 0.72 | 4.01  
region4 [Africa] | 1.20 | 3.56  
region4 [Other] | 3.03 | 3.24  
life exp %ia% region4life exp \* region4=Europe | -6.97e-03 | 0.06  
life exp %ia% region4life exp \* region4=Africa | -0.02 | 0.06  
life exp %ia% region4life exp \* region4=Other | -0.04 | 0.05  
freedom | 2.63 | 0.50  
corruption | -0.81 | 0.38  
  
Parameter | 90% CI | t  
-----------------------------------------------------------------------  
(Intercept) | [-8.00, 4.65] | -0.45  
rcs(social [ degree] | [ 1.88, 4.58] | 3.98  
rcs(social [ degree] | [-1.17, 1.78] | 0.34  
rcs(life exp [ degree] | [-0.06, 0.15] | 0.70  
rcs(life exp [ degree] | [-0.07, 0.10] | 0.32  
region4 [Europe] | [-6.08, 7.52] | 0.18  
region4 [Africa] | [-4.81, 7.22] | 0.34  
region4 [Other] | [-2.34, 8.41] | 0.93  
life exp %ia% region4life exp \* region4=Europe | [-0.10, 0.09] | -0.12  
life exp %ia% region4life exp \* region4=Africa | [-0.12, 0.08] | -0.32  
life exp %ia% region4life exp \* region4=Other | [-0.12, 0.04] | -0.78  
freedom | [ 1.81, 3.45] | 5.31  
corruption | [-1.45, -0.16] | -2.10  
  
Parameter | df | p  
----------------------------------------------------------------  
(Intercept) | 25.42 | 0.655   
rcs(social [ degree] | 110.60 | < .001  
rcs(social [ degree] | 118.93 | 0.733   
rcs(life exp [ degree] | 25.63 | 0.490   
rcs(life exp [ degree] | 29.16 | 0.750   
region4 [Europe] | 31.49 | 0.859   
region4 [Africa] | 36.85 | 0.737   
region4 [Other] | 119.36 | 0.352   
life exp %ia% region4life exp \* region4=Europe | 32.04 | 0.905   
life exp %ia% region4life exp \* region4=Africa | 37.94 | 0.754   
life exp %ia% region4life exp \* region4=Other | 119.58 | 0.437   
freedom | 98.68 | < .001  
corruption | 49.20 | 0.041

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

## Build 15 imputations, estimate fit4

fit4\_imp\_ests <-   
 mice(happy, m = 15, seed = 43219, print = FALSE) |>  
 with(lm(ladder ~ log\_gdp + social + life\_exp + freedom +   
 generosity + corruption + pos\_affect + neg\_affect +   
 ftemp\_c + fpop\_dens + region4)) |>  
 pool()

Warning: Number of logged events: 645

## Across 15 imputations: fit4 estimates

glance(fit4\_imp\_ests)

nimp nobs r.squared adj.r.squared  
1 15 138 0.8296207 0.8102271

model\_parameters(fit4\_imp\_ests, ci = 0.90)

Warning: Number of logged events: 645  
Warning: Number of logged events: 645  
Warning: Number of logged events: 645  
Warning: Number of logged events: 645

# Fixed Effects  
  
Parameter | Coefficient | SE | 90% CI | t | df | p  
--------------------------------------------------------------------------------  
(Intercept) | -2.33 | 1.32 | [-4.53, -0.12] | -1.77 | 55.15 | 0.083   
log gdp | 0.20 | 0.10 | [ 0.02, 0.37] | 1.94 | 29.01 | 0.062   
social | 2.69 | 0.70 | [ 1.53, 3.86] | 3.85 | 87.75 | < .001  
life exp | 0.03 | 0.02 | [-0.01, 0.07] | 1.29 | 29.31 | 0.206   
freedom | 1.83 | 0.52 | [ 0.98, 2.69] | 3.55 | 118.53 | < .001  
generosity | 0.04 | 0.31 | [-0.47, 0.56] | 0.14 | 90.81 | 0.887   
corruption | -0.73 | 0.32 | [-1.27, -0.20] | -2.31 | 70.52 | 0.024   
pos affect | 1.46 | 0.66 | [ 0.37, 2.56] | 2.21 | 118.49 | 0.029   
neg affect | 0.05 | 0.68 | [-1.08, 1.19] | 0.08 | 119.78 | 0.940   
ftemp c [warm] | 0.02 | 0.14 | [-0.22, 0.26] | 0.16 | 111.20 | 0.876   
fpop dens [med] | -0.07 | 0.12 | [-0.27, 0.14] | -0.54 | 112.06 | 0.592   
fpop dens [high] | 0.02 | 0.13 | [-0.19, 0.23] | 0.16 | 108.15 | 0.872   
region4 [Europe] | 0.28 | 0.15 | [ 0.03, 0.52] | 1.88 | 111.61 | 0.063   
region4 [Africa] | 0.01 | 0.17 | [-0.27, 0.30] | 0.08 | 102.46 | 0.940   
region4 [Other] | 0.36 | 0.18 | [ 0.07, 0.65] | 2.06 | 113.43 | 0.042

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

## Build 15 imputations, estimate fit5

fit5\_imp\_ests <-   
 mice(happy, m = 15, seed = 43220, print = FALSE) |>  
 with(lm(ladder ~ rcs(log\_gdp,4) + poly(social,2) + life\_exp \* region4 +   
 freedom + generosity + corruption + pos\_affect +   
 neg\_affect + ftemp\_c + fpop\_dens)) |>  
 pool()

Warning: Number of logged events: 647

## Across 15 imputations: fit5 estimates

glance(fit5\_imp\_ests)

nimp nobs r.squared adj.r.squared  
1 15 138 0.8370076 0.8091443

model\_parameters(fit5\_imp\_ests, ci = 0.90)

Warning: Number of logged events: 647  
Warning: Number of logged events: 647  
Warning: Number of logged events: 647  
Warning: Number of logged events: 647

# Fixed Effects  
  
Parameter | Coefficient | SE | 90% CI | t  
-------------------------------------------------------------------------  
(Intercept) | 0.23 | 2.30 | [-3.58, 4.05] | 0.10  
rcs(log gdp [ degree] | 0.05 | 0.21 | [-0.31, 0.40] | 0.22  
rcs(log gdp [ degree] | 0.31 | 0.39 | [-0.34, 0.96] | 0.80  
rcs(log gdp [ degree] | -1.16 | 2.12 | [-4.68, 2.36] | -0.55  
social [1st degree] | 3.70 | 1.08 | [ 1.92, 5.49] | 3.44  
social [2nd degree] | 0.13 | 0.60 | [-0.86, 1.12] | 0.22  
life exp | 0.04 | 0.03 | [-0.01, 0.08] | 1.46  
region4 [Europe] | 2.70 | 3.58 | [-3.30, 8.69] | 0.75  
region4 [Africa] | 1.02 | 2.28 | [-2.77, 4.82] | 0.45  
region4 [Other] | 2.80 | 3.19 | [-2.50, 8.10] | 0.88  
freedom | 1.87 | 0.54 | [ 0.96, 2.77] | 3.43  
generosity | 0.05 | 0.35 | [-0.54, 0.63] | 0.13  
corruption | -0.68 | 0.36 | [-1.28, -0.09] | -1.91  
pos affect | 1.57 | 0.76 | [ 0.31, 2.84] | 2.07  
neg affect | 0.12 | 0.73 | [-1.09, 1.33] | 0.17  
ftemp c [warm] | 0.02 | 0.15 | [-0.22, 0.26] | 0.12  
fpop dens [med] | -0.03 | 0.12 | [-0.24, 0.17] | -0.28  
fpop dens [high] | -5.19e-03 | 0.13 | [-0.23, 0.22] | -0.04  
life exp × region4 [Europe] | -0.04 | 0.05 | [-0.12, 0.05] | -0.70  
life exp × region4 [Africa] | -0.02 | 0.04 | [-0.08, 0.05] | -0.44  
life exp × region4 [Other] | -0.04 | 0.05 | [-0.12, 0.04] | -0.77  
  
Parameter | df | p  
---------------------------------------------  
(Intercept) | 97.01 | 0.920   
rcs(log gdp [ degree] | 78.90 | 0.827   
rcs(log gdp [ degree] | 98.22 | 0.427   
rcs(log gdp [ degree] | 96.32 | 0.586   
social [1st degree] | 105.88 | < .001  
social [2nd degree] | 112.79 | 0.826   
life exp | 50.45 | 0.149   
region4 [Europe] | 50.72 | 0.455   
region4 [Africa] | 95.45 | 0.655   
region4 [Other] | 108.19 | 0.382   
freedom | 107.33 | < .001  
generosity | 70.68 | 0.895   
corruption | 90.47 | 0.059   
pos affect | 89.36 | 0.042   
neg affect | 108.52 | 0.866   
ftemp c [warm] | 111.36 | 0.902   
fpop dens [med] | 111.71 | 0.780   
fpop dens [high] | 102.87 | 0.969   
life exp × region4 [Europe] | 50.63 | 0.490   
life exp × region4 [Africa] | 96.56 | 0.659   
life exp × region4 [Other] | 108.55 | 0.444

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

# Multiple Imputation with aregImpute()

## Our five models

| Model | ladder is predicted using… | df |
| --- | --- | --- |
| fit1 | social + life\_exp + freedom | 3 |
| fit2 | log\_gdp \* ftemp\_c | 3 |
| fit3 | rcs(social,3) + rcs(life\_exp, 3) + region4 + life\_exp %ia% region4 + freedom + corruption | 12 |
| fit4 | log\_gdp + social + life\_exp + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens + region4 | 14 |
| fit5 | rcs(log\_gdp,4) + poly(social,2) + life\_exp \* region4 + freedom + generosity + corruption + pos\_affect + neg\_affect + ftemp\_c + fpop\_dens | 20 |

## Use aregImpute() for fit1

set.seed(43221)  
dd <- datadist(happy)  
options(datadist = "dd")  
  
fit1\_imps15 <- aregImpute(~ ladder + social + life\_exp + freedom,  
 nk = c(0, 3), tlinear = FALSE, data = happy,   
 B = 10, n.impute = 15, pr = FALSE)

## Imputation Results for fit1

fit1\_imps15

Multiple Imputation using Bootstrap and PMM  
  
aregImpute(formula = ~ladder + social + life\_exp + freedom, data = happy,   
 n.impute = 15, nk = c(0, 3), tlinear = FALSE, pr = FALSE,   
 B = 10)  
  
n: 138 p: 4 Imputations: 15 nk: 0   
  
Number of NAs:  
 ladder social life\_exp freedom   
 0 0 3 2   
  
 type d.f.  
ladder s 1  
social s 1  
life\_exp s 1  
freedom s 1  
  
R-squares for Predicting Non-Missing Values for Each Variable  
Using Last Imputations of Predictors  
life\_exp freedom   
 0.545 0.506   
  
Resampling results for determining the complexity of imputation models  
  
Variable being imputed: life\_exp   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.591 0.586  
10-fold cross-validated R^2 0.623 0.584  
Bootstrap bias-corrected mean |error| 2.754 2.812  
10-fold cross-validated mean |error| 65.141 3.067  
Bootstrap bias-corrected median |error| 1.861 2.307  
10-fold cross-validated median |error| 65.931 2.709  
  
Variable being imputed: freedom   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.4064 0.3432  
10-fold cross-validated R^2 0.3531 0.4107  
Bootstrap bias-corrected mean |error| 0.0718 0.0807  
10-fold cross-validated mean |error| 0.9239 0.0794  
Bootstrap bias-corrected median |error| 0.0486 0.0524  
10-fold cross-validated median |error| 0.6886 0.0601

## Fit fit1 with fit.mult.impute()

fit1\_imp <-   
 fit.mult.impute(ladder ~ social + life\_exp + freedom,  
 fitter = ols, xtrans = fit1\_imps15, data = happy,  
 fitargs=list(x = TRUE, y = TRUE))

Wald Statistic Information  
  
Variance Inflation Factors Due to Imputation:  
  
Intercept social life\_exp freedom   
 1.00 1.01 1.01 1.01   
  
Fraction of Missing Information:  
  
Intercept social life\_exp freedom   
 0.00 0.01 0.01 0.01   
  
d.f. for t-distribution for Tests of Single Coefficients:  
  
Intercept social life\_exp freedom   
 709901.8 209644.8 360326.2 232384.8   
  
The following fit components were averaged over the 15 model fits:  
  
 fitted.values stats linear.predictors

## What’s in fit1\_imp?

fit1\_imp

Linear Regression Model  
  
fit.mult.impute(formula = ladder ~ social + life\_exp + freedom,   
 fitter = ols, xtrans = fit1\_imps15, data = happy, fitargs = list(x = TRUE,   
 y = TRUE))  
  
 Model Likelihood Discrimination   
 Ratio Test Indexes   
Obs 138 LR chi2 210.20 R2 0.782   
sigma0.5380 d.f. 3 R2 adj 0.777   
d.f. 134 Pr(> chi2) 0.0000 g 1.122   
  
Residuals  
  
 Min 1Q Median 3Q Max   
-2.05939 -0.27086 0.07173 0.31763 1.10781   
  
 Coef S.E. t Pr(>|t|)  
Intercept -4.2124 0.5841 -7.21 <0.0001   
social 3.5140 0.5264 6.68 <0.0001   
life\_exp 0.0713 0.0119 5.99 <0.0001   
freedom 3.0353 0.4319 7.03 <0.0001

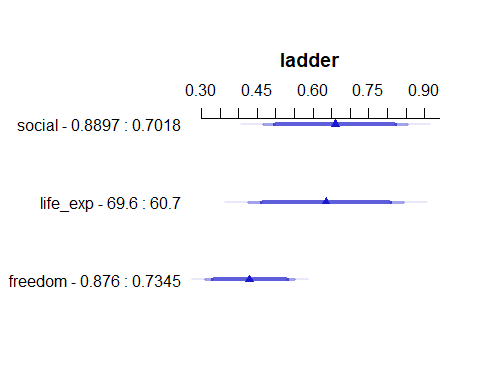
## Summary of fit1 After Imputation

summary(fit1\_imp)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.95 Upper 0.95  
 social 0.70185 0.88966 0.18782 0.65999 0.098863 0.46445 0.85552   
 life\_exp 60.70000 69.60000 8.90000 0.63494 0.105970 0.42535 0.84452   
 freedom 0.73445 0.87598 0.14153 0.42957 0.061120 0.30869 0.55046

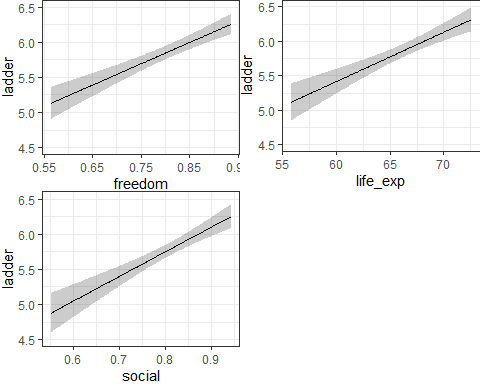
## fit1 Effects Plot after imputation

plot(summary(fit1\_imp))



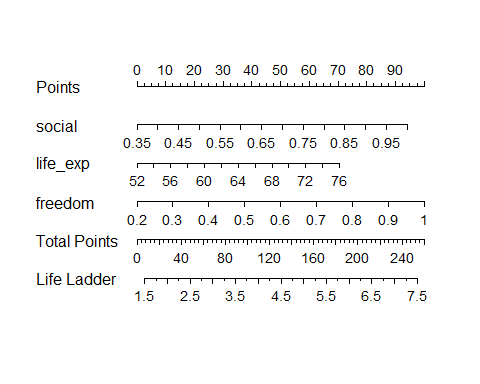
## Prediction Plot: fit1 post-imputation

ggplot(Predict(fit1\_imp))



## Nomogram of fit1 after imputation

plot(nomogram(fit1\_imp), lplabel = "Life Ladder")



## fit1 Bootstrap Validation after Imputation

set.seed(43222)  
validate(fit1\_imp, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.7814 0.7821 0.7744 0.0078 0.7736 300  
MSE 0.2818 0.2744 0.2908 -0.0164 0.2982 300  
g 1.1202 1.1109 1.1181 -0.0072 1.1274 300  
Intercept 0.0000 0.0000 0.0212 -0.0212 0.0212 300  
Slope 1.0000 1.0000 0.9960 0.0040 0.9960 300

## Use aregImpute() for fit2

set.seed(4324324)  
dd <- datadist(happy)  
options(datadist = "dd")  
  
fit2\_imps15 <- aregImpute(~ ladder + log\_gdp + ftemp\_c,  
 nk = c(0, 3), tlinear = FALSE, data = happy,   
 B = 10, n.impute = 15, pr = FALSE)

## Imputation Results for fit2

fit2\_imps15

Multiple Imputation using Bootstrap and PMM  
  
aregImpute(formula = ~ladder + log\_gdp + ftemp\_c, data = happy,   
 n.impute = 15, nk = c(0, 3), tlinear = FALSE, pr = FALSE,   
 B = 10)  
  
n: 138 p: 3 Imputations: 15 nk: 0   
  
Number of NAs:  
 ladder log\_gdp ftemp\_c   
 0 9 0   
  
 type d.f.  
ladder s 1  
log\_gdp s 1  
ftemp\_c c 1  
  
R-squares for Predicting Non-Missing Values for Each Variable  
Using Last Imputations of Predictors  
log\_gdp   
 0.68   
  
Resampling results for determining the complexity of imputation models  
  
Variable being imputed: log\_gdp   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.633 0.668  
10-fold cross-validated R^2 0.664 0.670  
Bootstrap bias-corrected mean |error| 0.505 0.484  
10-fold cross-validated mean |error| 9.529 0.490  
Bootstrap bias-corrected median |error| 0.375 0.373  
10-fold cross-validated median |error| 9.504 0.396

## Fit fit2 with fit.mult.impute()

fit2\_imp <-   
 fit.mult.impute(ladder ~ log\_gdp \* ftemp\_c,  
 fitter = ols, xtrans = fit2\_imps15, data = happy,  
 fitargs=list(x = TRUE, y = TRUE))

Wald Statistic Information  
  
Variance Inflation Factors Due to Imputation:  
  
 Intercept log\_gdp ftemp\_c=warm   
 1.19 1.21 1.16   
log\_gdp \* ftemp\_c=warm   
 1.16   
  
Fraction of Missing Information:  
  
 Intercept log\_gdp ftemp\_c=warm   
 0.16 0.17 0.14   
log\_gdp \* ftemp\_c=warm   
 0.14   
  
d.f. for t-distribution for Tests of Single Coefficients:  
  
 Intercept log\_gdp ftemp\_c=warm   
 538.10 477.42 757.89   
log\_gdp \* ftemp\_c=warm   
 741.28   
  
The following fit components were averaged over the 15 model fits:  
  
 fitted.values stats linear.predictors

## What’s in fit2\_imp?

fit2\_imp

Linear Regression Model  
  
fit.mult.impute(formula = ladder ~ log\_gdp \* ftemp\_c, fitter = ols,   
 xtrans = fit2\_imps15, data = happy, fitargs = list(x = TRUE,   
 y = TRUE))  
  
 Model Likelihood Discrimination   
 Ratio Test Indexes   
Obs 138 LR chi2 122.24 R2 0.586   
sigma0.7406 d.f. 3 R2 adj 0.577   
d.f. 134 Pr(> chi2) 0.0000 g 1.002   
  
Residuals  
  
 Min 1Q Median 3Q Max   
-4.16960 -0.42446 0.08669 0.39210 1.74014   
  
 Coef S.E. t Pr(>|t|)  
Intercept -3.3796 1.3510 -2.50 0.0136   
log\_gdp 0.9283 0.1331 6.97 <0.0001   
ftemp\_c=warm 2.0772 1.5487 1.34 0.1821   
log\_gdp \* ftemp\_c=warm -0.1945 0.1580 -1.23 0.2205

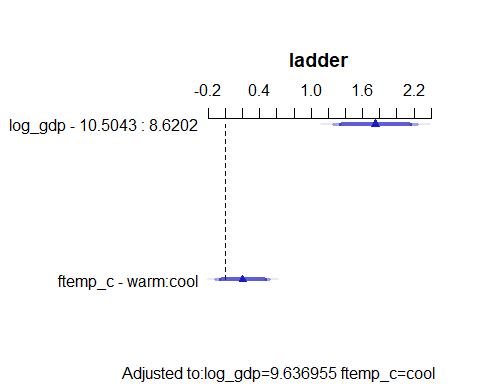
## Summary of fit2 After Imputation

summary(fit2\_imp)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.95 Upper 0.95  
 log\_gdp 8.6202 10.504 1.884 1.74890 0.25077 1.25290 2.24490   
 ftemp\_c - warm:cool 1.0000 2.000 NA 0.20269 0.16054 -0.11482 0.52021   
  
Adjusted to: log\_gdp=9.636955 ftemp\_c=cool

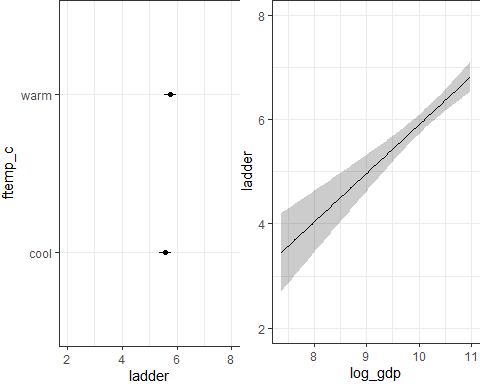
## fit2 Effects Plot after imputation

plot(summary(fit2\_imp))



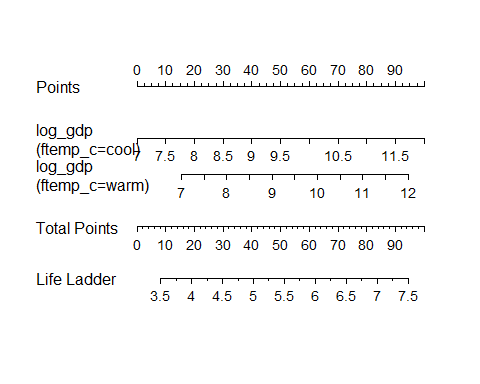
## Prediction Plot: fit2 post-imputation

ggplot(Predict(fit2\_imp), layout = c(1,2))



## Nomogram of fit2 after imputation

plot(nomogram(fit2\_imp), lplabel = "Life Ladder")



## fit2 Bootstrap Validation after Imputation

set.seed(43223)  
validate(fit2\_imp, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.5362 0.5518 0.5213 0.0305 0.5056 300  
MSE 0.5979 0.5788 0.6171 -0.0383 0.6361 300  
g 0.9869 0.9626 0.9516 0.0110 0.9759 300  
Intercept 0.0000 0.0000 0.1362 -0.1362 0.1362 300  
Slope 1.0000 1.0000 0.9770 0.0230 0.9770 300

## Use aregImpute() for fit3

set.seed(43224)  
dd <- datadist(happy)  
options(datadist = "dd")  
  
fit3\_imps15 <- aregImpute(~ ladder + social + life\_exp + region4 +   
 freedom + corruption,  
 nk = c(0, 3), tlinear = FALSE, data = happy,   
 B = 10, n.impute = 15, pr = FALSE)

## Imputation Results for fit3

fit3\_imps15

Multiple Imputation using Bootstrap and PMM  
  
aregImpute(formula = ~ladder + social + life\_exp + region4 +   
 freedom + corruption, data = happy, n.impute = 15, nk = c(0,   
 3), tlinear = FALSE, pr = FALSE, B = 10)  
  
n: 138 p: 6 Imputations: 15 nk: 0   
  
Number of NAs:  
 ladder social life\_exp region4 freedom corruption   
 0 0 3 0 2 7   
  
 type d.f.  
ladder s 1  
social s 1  
life\_exp s 1  
region4 c 3  
freedom s 1  
corruption s 1  
  
R-squares for Predicting Non-Missing Values for Each Variable  
Using Last Imputations of Predictors  
 life\_exp freedom corruption   
 0.751 0.499 0.287   
  
Resampling results for determining the complexity of imputation models  
  
Variable being imputed: life\_exp   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.712 0.703  
10-fold cross-validated R^2 0.700 0.641  
Bootstrap bias-corrected mean |error| 2.375 2.408  
10-fold cross-validated mean |error| 65.317 2.553  
Bootstrap bias-corrected median |error| 1.774 1.934  
10-fold cross-validated median |error| 66.014 2.216  
  
Variable being imputed: freedom   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.3733 0.3894  
10-fold cross-validated R^2 0.4813 0.3890  
Bootstrap bias-corrected mean |error| 0.0739 0.0724  
10-fold cross-validated mean |error| 0.8827 0.0814  
Bootstrap bias-corrected median |error| 0.0481 0.0493  
10-fold cross-validated median |error| 0.7244 0.0573  
  
Variable being imputed: corruption   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.2049 0.3861  
10-fold cross-validated R^2 0.2257 0.3963  
Bootstrap bias-corrected mean |error| 0.1226 0.1239  
10-fold cross-validated mean |error| 0.9829 0.1213  
Bootstrap bias-corrected median |error| 0.0996 0.0875  
10-fold cross-validated median |error| 0.8601 0.1012

## Fit fit3 with fit.mult.impute()

fit3\_imp <-   
 fit.mult.impute(ladder ~ rcs(social,3) + rcs(life\_exp, 3) + region4 +   
 life\_exp %ia% region4 + freedom + corruption,  
 fitter = ols, xtrans = fit3\_imps15, data = happy,  
 fitargs=list(x = TRUE, y = TRUE))

Wald Statistic Information  
  
Variance Inflation Factors Due to Imputation:  
  
 Intercept social social'   
 1.02 1.01 1.01   
 life\_exp life\_exp' region4=Europe   
 1.02 1.03 1.10   
 region4=Africa region4=Other life\_exp \* region4=Europe   
 1.01 1.01 1.10   
life\_exp \* region4=Africa life\_exp \* region4=Other freedom   
 1.01 1.01 1.02   
 corruption   
 1.18   
  
Fraction of Missing Information:  
  
 Intercept social social'   
 0.02 0.01 0.01   
 life\_exp life\_exp' region4=Europe   
 0.02 0.03 0.09   
 region4=Africa region4=Other life\_exp \* region4=Europe   
 0.01 0.01 0.09   
life\_exp \* region4=Africa life\_exp \* region4=Other freedom   
 0.01 0.01 0.02   
 corruption   
 0.16   
  
d.f. for t-distribution for Tests of Single Coefficients:  
  
 Intercept social social'   
 42731.11 246291.29 73998.01   
 life\_exp life\_exp' region4=Europe   
 60167.56 17556.72 1610.87   
 region4=Africa region4=Other life\_exp \* region4=Europe   
 74247.67 154549.04 1771.43   
life\_exp \* region4=Africa life\_exp \* region4=Other freedom   
 66464.54 156907.00 54610.15   
 corruption   
 580.49   
  
The following fit components were averaged over the 15 model fits:  
  
 fitted.values stats linear.predictors

## What’s in fit3\_imp?

fit3\_imp

Linear Regression Model  
  
fit.mult.impute(formula = ladder ~ rcs(social, 3) + rcs(life\_exp,   
 3) + region4 + life\_exp %ia% region4 + freedom + corruption,   
 fitter = ols, xtrans = fit3\_imps15, data = happy, fitargs = list(x = TRUE,   
 y = TRUE))  
  
 Model Likelihood Discrimination   
 Ratio Test Indexes   
Obs 138 LR chi2 239.24 R2 0.823   
sigma0.5014 d.f. 12 R2 adj 0.806   
d.f. 125 Pr(> chi2) 0.0000 g 1.173   
  
Residuals  
  
 Min 1Q Median 3Q Max   
-1.71738 -0.29104 0.05693 0.29776 1.24805   
  
 Coef S.E. t Pr(>|t|)  
Intercept -4.0701 2.7546 -1.48 0.1420   
social 3.2042 0.7730 4.15 <0.0001   
social' 0.2321 0.8609 0.27 0.7880   
life\_exp 0.0897 0.0478 1.88 0.0626   
life\_exp' -0.0192 0.0401 -0.48 0.6322   
region4=Europe -1.0146 3.4400 -0.29 0.7685   
region4=Africa 3.4253 2.8813 1.19 0.2368   
region4=Other 2.7787 3.1529 0.88 0.3798   
life\_exp \* region4=Europe 0.0179 0.0496 0.36 0.7188   
life\_exp \* region4=Africa -0.0536 0.0468 -1.14 0.2544   
life\_exp \* region4=Other -0.0338 0.0468 -0.72 0.4709   
freedom 2.3464 0.4627 5.07 <0.0001   
corruption -0.9870 0.3283 -3.01 0.0032

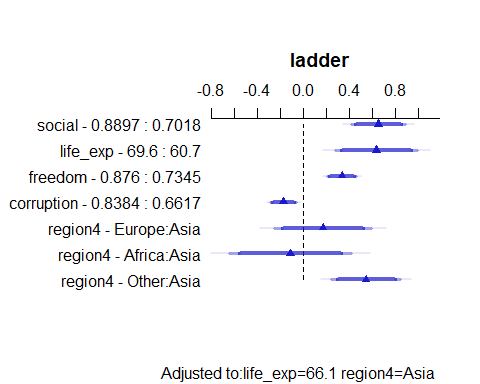
## Summary of fit3 After Imputation

summary(fit3\_imp)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E. Lower 0.95  
 social 0.70185 0.88966 0.18782 0.64939 0.120740 0.41044   
 life\_exp 60.70000 69.60000 8.90000 0.63270 0.182930 0.27066   
 freedom 0.73445 0.87598 0.14153 0.33208 0.065484 0.20248   
 corruption 0.66168 0.83839 0.17671 -0.17441 0.058018 -0.28923   
 region4 - Europe:Asia 1.00000 2.00000 NA 0.16836 0.214630 -0.25641   
 region4 - Africa:Asia 1.00000 3.00000 NA -0.11589 0.269780 -0.64981   
 region4 - Other:Asia 1.00000 4.00000 NA 0.54146 0.152870 0.23892   
 Upper 0.95  
 0.888340   
 0.994730   
 0.461680   
 -0.059582   
 0.593130   
 0.418040   
 0.844000   
  
Adjusted to: life\_exp=66.1 region4=Asia

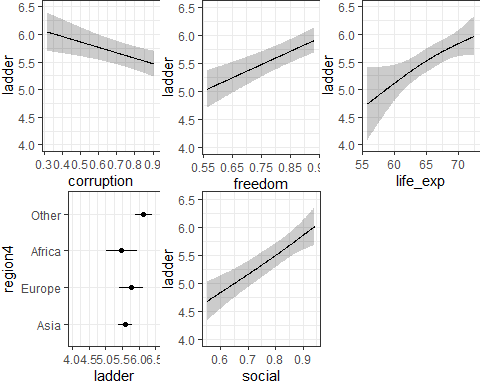
## fit3 Effects Plot after imputation

plot(summary(fit3\_imp))



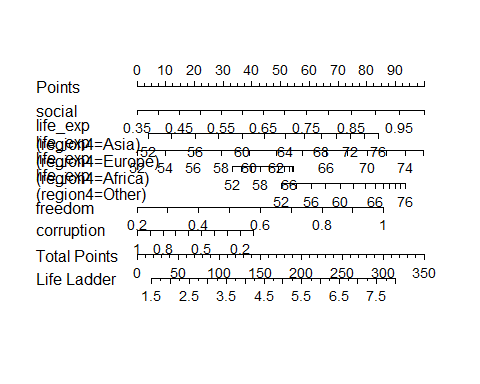
## Prediction Plot: fit3 post-imputation

ggplot(Predict(fit3\_imp))



## Nomogram of fit3 after imputation

plot(nomogram(fit3\_imp), lplabel = "Life Ladder")



## fit3 Bootstrap Validation after Imputation

set.seed(43225)  
validate(fit3\_imp, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.8173 0.8293 0.7951 0.0342 0.7832 300  
MSE 0.2355 0.2128 0.2641 -0.0512 0.2867 300  
g 1.1699 1.1569 1.1544 0.0025 1.1674 300  
Intercept 0.0000 0.0000 0.0858 -0.0858 0.0858 300  
Slope 1.0000 1.0000 0.9840 0.0160 0.9840 300

## Use aregImpute() for fit4

set.seed(43226)  
dd <- datadist(happy)  
options(datadist = "dd")  
  
fit4\_imps15 <- aregImpute(~ ladder + log\_gdp + social + life\_exp +   
 freedom + generosity + corruption +   
 pos\_affect + neg\_affect + ftemp\_c +   
 fpop\_dens + region4,  
 nk = c(0, 3), tlinear = FALSE, data = happy,   
 B = 10, n.impute = 15, pr = FALSE)

## Imputation Results for fit4

fit4\_imps15

Multiple Imputation using Bootstrap and PMM  
  
aregImpute(formula = ~ladder + log\_gdp + social + life\_exp +   
 freedom + generosity + corruption + pos\_affect + neg\_affect +   
 ftemp\_c + fpop\_dens + region4, data = happy, n.impute = 15,   
 nk = c(0, 3), tlinear = FALSE, pr = FALSE, B = 10)  
  
n: 138 p: 12 Imputations: 15 nk: 0   
  
Number of NAs:  
 ladder log\_gdp social life\_exp freedom generosity corruption   
 0 9 0 3 2 9 7   
pos\_affect neg\_affect ftemp\_c fpop\_dens region4   
 0 0 0 0 0   
  
 type d.f.  
ladder s 1  
log\_gdp s 1  
social s 1  
life\_exp s 1  
freedom s 1  
generosity s 1  
corruption s 1  
pos\_affect s 1  
neg\_affect s 1  
ftemp\_c c 1  
fpop\_dens c 2  
region4 c 3  
  
R-squares for Predicting Non-Missing Values for Each Variable  
Using Last Imputations of Predictors  
 log\_gdp life\_exp freedom generosity corruption   
 0.792 0.834 0.582 0.211 0.398   
  
Resampling results for determining the complexity of imputation models  
  
Variable being imputed: log\_gdp   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.783 0.747  
10-fold cross-validated R^2 0.792 0.810  
Bootstrap bias-corrected mean |error| 0.395 0.443  
10-fold cross-validated mean |error| 9.532 0.399  
Bootstrap bias-corrected median |error| 0.242 0.307  
10-fold cross-validated median |error| 9.523 0.309  
  
Variable being imputed: life\_exp   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.777 0.741  
10-fold cross-validated R^2 0.753 0.693  
Bootstrap bias-corrected mean |error| 2.026 2.338  
10-fold cross-validated mean |error| 64.949 2.407  
Bootstrap bias-corrected median |error| 1.486 1.952  
10-fold cross-validated median |error| 65.335 1.968  
  
Variable being imputed: freedom   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.5052 0.4976  
10-fold cross-validated R^2 0.4572 0.4699  
Bootstrap bias-corrected mean |error| 0.0623 0.0692  
10-fold cross-validated mean |error| 0.9377 0.0761  
Bootstrap bias-corrected median |error| 0.0424 0.0489  
10-fold cross-validated median |error| 0.7463 0.0548  
  
Variable being imputed: generosity   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.0727 0.138  
10-fold cross-validated R^2 0.1069 0.139  
Bootstrap bias-corrected mean |error| 0.1246 0.134  
10-fold cross-validated mean |error| 0.7599 0.164  
Bootstrap bias-corrected median |error| 0.0879 0.115  
10-fold cross-validated median |error| 0.5987 0.127  
  
Variable being imputed: corruption   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.2323 0.3900  
10-fold cross-validated R^2 0.2976 0.3612  
Bootstrap bias-corrected mean |error| 0.1254 0.1145  
10-fold cross-validated mean |error| 0.9722 0.1150  
Bootstrap bias-corrected median |error| 0.0894 0.0808  
10-fold cross-validated median |error| 0.8304 0.0854

## Fit fit4 with fit.mult.impute()

fit4\_imp <-   
 fit.mult.impute(ladder ~ log\_gdp + social + life\_exp + freedom +   
 generosity + corruption + pos\_affect + neg\_affect +   
 ftemp\_c + fpop\_dens + region4,  
 fitter = ols, xtrans = fit4\_imps15, data = happy,  
 fitargs=list(x = TRUE, y = TRUE))

Wald Statistic Information  
  
Variance Inflation Factors Due to Imputation:  
  
 Intercept log\_gdp social life\_exp freedom   
 1.12 1.16 1.03 1.13 1.03   
 generosity corruption pos\_affect neg\_affect ftemp\_c=warm   
 1.09 1.11 1.01 1.01 1.04   
 fpop\_dens=med fpop\_dens=high region4=Europe region4=Africa region4=Other   
 1.02 1.03 1.02 1.03 1.03   
  
Fraction of Missing Information:  
  
 Intercept log\_gdp social life\_exp freedom   
 0.11 0.14 0.03 0.12 0.03   
 generosity corruption pos\_affect neg\_affect ftemp\_c=warm   
 0.09 0.10 0.01 0.01 0.04   
 fpop\_dens=med fpop\_dens=high region4=Europe region4=Africa region4=Other   
 0.02 0.03 0.02 0.03 0.02   
  
d.f. for t-distribution for Tests of Single Coefficients:  
  
 Intercept log\_gdp social life\_exp freedom   
 1159.65 711.36 13336.18 1010.21 17500.12   
 generosity corruption pos\_affect neg\_affect ftemp\_c=warm   
 1925.02 1328.48 245805.13 395269.64 8436.44   
 fpop\_dens=med fpop\_dens=high region4=Europe region4=Africa region4=Other   
 25692.44 21858.64 29605.46 19998.02 23429.31   
  
The following fit components were averaged over the 15 model fits:  
  
 fitted.values stats linear.predictors

## What’s in fit4\_imp?

fit4\_imp

Linear Regression Model  
  
fit.mult.impute(formula = ladder ~ log\_gdp + social + life\_exp +   
 freedom + generosity + corruption + pos\_affect + neg\_affect +   
 ftemp\_c + fpop\_dens + region4, fitter = ols, xtrans = fit4\_imps15,   
 data = happy, fitargs = list(x = TRUE, y = TRUE))  
  
 Model Likelihood Discrimination   
 Ratio Test Indexes   
Obs 138 LR chi2 250.26 R2 0.837   
sigma0.4857 d.f. 14 R2 adj 0.818   
d.f. 123 Pr(> chi2) 0.0000 g 1.184   
  
Residuals  
  
 Min 1Q Median 3Q Max   
-1.85283 -0.25165 0.01009 0.29096 1.17983   
  
 Coef S.E. t Pr(>|t|)  
Intercept -2.4431 1.1891 -2.05 0.0420   
log\_gdp 0.2124 0.0881 2.41 0.0174   
social 2.5785 0.6551 3.94 0.0001   
life\_exp 0.0300 0.0192 1.57 0.1200   
freedom 1.8793 0.5078 3.70 0.0003   
generosity 0.0143 0.3032 0.05 0.9626   
corruption -0.8034 0.2963 -2.71 0.0076   
pos\_affect 1.4337 0.6444 2.22 0.0279   
neg\_affect 0.1661 0.6686 0.25 0.8042   
ftemp\_c=warm 0.0174 0.1394 0.12 0.9009   
fpop\_dens=med -0.0607 0.1177 -0.52 0.6071   
fpop\_dens=high 0.0008 0.1241 0.01 0.9946   
region4=Europe 0.2899 0.1405 2.06 0.0412   
region4=Africa 0.0539 0.1621 0.33 0.7402   
region4=Other 0.3885 0.1707 2.28 0.0245

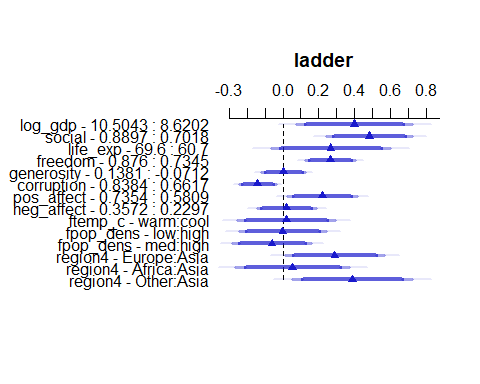
## Summary of fit4 After Imputation

summary(fit4\_imp)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E.   
 log\_gdp 8.620200 10.50400 1.88400 0.40021000 0.165950  
 social 0.701850 0.88966 0.18782 0.48430000 0.123030  
 life\_exp 60.700000 69.60000 8.90000 0.26703000 0.170540  
 freedom 0.734450 0.87598 0.14153 0.26597000 0.071870  
 generosity -0.071157 0.13805 0.20921 0.00298280 0.063442  
 corruption 0.661680 0.83839 0.17671 -0.14197000 0.052352  
 pos\_affect 0.580940 0.73544 0.15449 0.22150000 0.099559  
 neg\_affect 0.229710 0.35724 0.12753 0.02118500 0.085266  
 ftemp\_c - warm:cool 1.000000 2.00000 NA 0.01740500 0.139430  
 fpop\_dens - low:high 3.000000 1.00000 NA -0.00083372 0.124080  
 fpop\_dens - med:high 3.000000 2.00000 NA -0.06153400 0.111900  
 region4 - Europe:Asia 1.000000 2.00000 NA 0.28988000 0.140530  
 region4 - Africa:Asia 1.000000 3.00000 NA 0.05387600 0.162090  
 region4 - Other:Asia 1.000000 4.00000 NA 0.38854000 0.170670  
 Lower 0.95 Upper 0.95  
 0.071725 0.728690   
 0.240760 0.727830   
 -0.070552 0.604610   
 0.123710 0.408230   
 -0.122600 0.128560   
 -0.245600 -0.038345   
 0.024427 0.418570   
 -0.147590 0.189960   
 -0.258590 0.293400   
 -0.246440 0.244770   
 -0.283040 0.159970   
 0.011710 0.568050   
 -0.266970 0.374720   
 0.050711 0.726360

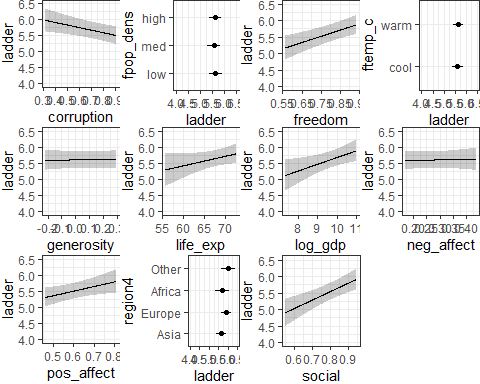
## fit4 Effects Plot after imputation

plot(summary(fit4\_imp))



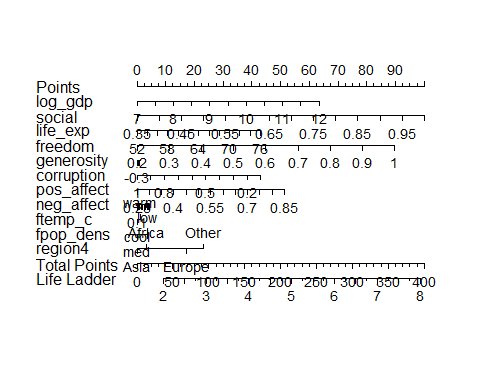
## Prediction Plot: fit4 post-imputation

ggplot(Predict(fit4\_imp))



## Nomogram of fit4 after imputation

plot(nomogram(fit4\_imp), lplabel = "Life Ladder")



## fit4 Bootstrap Validation after Imputation

set.seed(43227)  
validate(fit4\_imp, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.8322 0.8491 0.8085 0.0406 0.7916 300  
MSE 0.2163 0.1920 0.2469 -0.0549 0.2712 300  
g 1.1779 1.1850 1.1679 0.0171 1.1608 300  
Intercept 0.0000 0.0000 0.0731 -0.0731 0.0731 300  
Slope 1.0000 1.0000 0.9858 0.0142 0.9858 300

## Use aregImpute() for fit5

set.seed(43228)  
dd <- datadist(happy)  
options(datadist = "dd")  
  
fit5\_imps15 <- aregImpute(~ ladder + log\_gdp + social + life\_exp +   
 freedom + generosity + corruption +   
 pos\_affect + neg\_affect + ftemp\_c +   
 fpop\_dens + region4,  
 nk = c(0, 3), tlinear = FALSE, data = happy,   
 B = 10, n.impute = 15, pr = FALSE)

## Imputation Results for fit5

fit5\_imps15

Multiple Imputation using Bootstrap and PMM  
  
aregImpute(formula = ~ladder + log\_gdp + social + life\_exp +   
 freedom + generosity + corruption + pos\_affect + neg\_affect +   
 ftemp\_c + fpop\_dens + region4, data = happy, n.impute = 15,   
 nk = c(0, 3), tlinear = FALSE, pr = FALSE, B = 10)  
  
n: 138 p: 12 Imputations: 15 nk: 0   
  
Number of NAs:  
 ladder log\_gdp social life\_exp freedom generosity corruption   
 0 9 0 3 2 9 7   
pos\_affect neg\_affect ftemp\_c fpop\_dens region4   
 0 0 0 0 0   
  
 type d.f.  
ladder s 1  
log\_gdp s 1  
social s 1  
life\_exp s 1  
freedom s 1  
generosity s 1  
corruption s 1  
pos\_affect s 1  
neg\_affect s 1  
ftemp\_c c 1  
fpop\_dens c 2  
region4 c 3  
  
R-squares for Predicting Non-Missing Values for Each Variable  
Using Last Imputations of Predictors  
 log\_gdp life\_exp freedom generosity corruption   
 0.849 0.868 0.570 0.223 0.354   
  
Resampling results for determining the complexity of imputation models  
  
Variable being imputed: log\_gdp   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.798 0.790  
10-fold cross-validated R^2 0.774 0.722  
Bootstrap bias-corrected mean |error| 0.390 0.424  
10-fold cross-validated mean |error| 9.550 0.432  
Bootstrap bias-corrected median |error| 0.263 0.348  
10-fold cross-validated median |error| 9.562 0.350  
  
Variable being imputed: life\_exp   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.801 0.769  
10-fold cross-validated R^2 0.807 0.763  
Bootstrap bias-corrected mean |error| 1.970 2.175  
10-fold cross-validated mean |error| 65.131 2.227  
Bootstrap bias-corrected median |error| 1.556 1.741  
10-fold cross-validated median |error| 65.938 1.905  
  
Variable being imputed: freedom   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.4834 0.4823  
10-fold cross-validated R^2 0.4511 0.5045  
Bootstrap bias-corrected mean |error| 0.0657 0.0706  
10-fold cross-validated mean |error| 0.9490 0.0772  
Bootstrap bias-corrected median |error| 0.0411 0.0439  
10-fold cross-validated median |error| 0.7457 0.0469  
  
Variable being imputed: generosity   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.0917 0.207  
10-fold cross-validated R^2 0.1740 0.157  
Bootstrap bias-corrected mean |error| 0.1272 0.132  
10-fold cross-validated mean |error| 0.7970 0.145  
Bootstrap bias-corrected median |error| 0.0942 0.103  
10-fold cross-validated median |error| 0.6369 0.109  
  
Variable being imputed: corruption   
 nk=0 nk=3  
Bootstrap bias-corrected R^2 0.2509 0.3711  
10-fold cross-validated R^2 0.2966 0.3808  
Bootstrap bias-corrected mean |error| 0.1251 0.1130  
10-fold cross-validated mean |error| 0.9773 0.1209  
Bootstrap bias-corrected median |error| 0.0857 0.0792  
10-fold cross-validated median |error| 0.8539 0.0863

## Fit fit5 with fit.mult.impute()

fit5\_imp <-   
 fit.mult.impute(ladder ~ rcs(log\_gdp,4) + pol(social,2) +   
 life\_exp \* region4 + freedom + generosity +   
 corruption + pos\_affect + neg\_affect +   
 ftemp\_c + fpop\_dens,  
 fitter = ols, xtrans = fit4\_imps15, data = happy,  
 fitargs=list(x = TRUE, y = TRUE))

Wald Statistic Information  
  
Variance Inflation Factors Due to Imputation:  
  
 Intercept log\_gdp log\_gdp'   
 1.04 1.11 1.06   
 log\_gdp'' social social^2   
 1.04 1.03 1.03   
 life\_exp region4=Europe region4=Africa   
 1.07 1.23 1.03   
 region4=Other freedom generosity   
 1.01 1.04 1.10   
 corruption pos\_affect neg\_affect   
 1.12 1.03 1.02   
 ftemp\_c=warm fpop\_dens=med fpop\_dens=high   
 1.04 1.02 1.03   
life\_exp \* region4=Europe life\_exp \* region4=Africa life\_exp \* region4=Other   
 1.23 1.03 1.01   
  
Fraction of Missing Information:  
  
 Intercept log\_gdp log\_gdp'   
 0.03 0.10 0.06   
 log\_gdp'' social social^2   
 0.04 0.03 0.03   
 life\_exp region4=Europe region4=Africa   
 0.06 0.19 0.03   
 region4=Other freedom generosity   
 0.01 0.03 0.09   
 corruption pos\_affect neg\_affect   
 0.11 0.03 0.02   
 ftemp\_c=warm fpop\_dens=med fpop\_dens=high   
 0.03 0.02 0.03   
life\_exp \* region4=Europe life\_exp \* region4=Africa life\_exp \* region4=Other   
 0.19 0.03 0.01   
  
d.f. for t-distribution for Tests of Single Coefficients:  
  
 Intercept log\_gdp log\_gdp'   
 11807.23 1422.57 3787.39   
 log\_gdp'' social social^2   
 7816.01 13325.08 17704.39   
 life\_exp region4=Europe region4=Africa   
 3552.79 387.87 16117.76   
 region4=Other freedom generosity   
 77802.29 11724.67 1758.33   
 corruption pos\_affect neg\_affect   
 1196.48 12412.41 27827.19   
 ftemp\_c=warm fpop\_dens=med fpop\_dens=high   
 11651.05 27221.01 14528.93   
life\_exp \* region4=Europe life\_exp \* region4=Africa life\_exp \* region4=Other   
 400.79 15205.48 77082.31   
  
The following fit components were averaged over the 15 model fits:  
  
 fitted.values stats linear.predictors

## What’s in fit5\_imp?

fit5\_imp

Linear Regression Model  
  
fit.mult.impute(formula = ladder ~ rcs(log\_gdp, 4) + pol(social,   
 2) + life\_exp \* region4 + freedom + generosity + corruption +   
 pos\_affect + neg\_affect + ftemp\_c + fpop\_dens, fitter = ols,   
 xtrans = fit4\_imps15, data = happy, fitargs = list(x = TRUE,   
 y = TRUE))  
  
 Model Likelihood Discrimination   
 Ratio Test Indexes   
Obs 138 LR chi2 253.72 R2 0.841   
sigma0.4918 d.f. 20 R2 adj 0.814   
d.f. 117 Pr(> chi2) 0.0000 g 1.189   
  
Residuals  
  
 Min 1Q Median 3Q Max   
-1.80571 -0.23829 0.01502 0.27750 0.96708   
  
 Coef S.E. t Pr(>|t|)  
Intercept -1.8855 2.2527 -0.84 0.4043   
log\_gdp 0.0640 0.2054 0.31 0.7560   
log\_gdp' 0.3084 0.3616 0.85 0.3955   
log\_gdp'' -1.4992 2.0946 -0.72 0.4756   
social 2.1871 3.5398 0.62 0.5379   
social^2 0.1411 2.4927 0.06 0.9549   
life\_exp 0.0416 0.0246 1.69 0.0934   
region4=Europe 2.1968 3.4900 0.63 0.5303   
region4=Africa 1.4019 2.2478 0.62 0.5341   
region4=Other 3.1003 3.1470 0.99 0.3266   
freedom 1.8802 0.5335 3.52 0.0006   
generosity 0.0473 0.3328 0.14 0.8872   
corruption -0.8132 0.3700 -2.20 0.0299   
pos\_affect 1.5288 0.7195 2.12 0.0357   
neg\_affect 0.1884 0.7130 0.26 0.7921   
ftemp\_c=warm 0.0110 0.1452 0.08 0.9395   
fpop\_dens=med -0.0291 0.1223 -0.24 0.8125   
fpop\_dens=high -0.0047 0.1303 -0.04 0.9716   
life\_exp \* region4=Europe -0.0287 0.0504 -0.57 0.5697   
life\_exp \* region4=Africa -0.0221 0.0365 -0.60 0.5468   
life\_exp \* region4=Other -0.0407 0.0468 -0.87 0.3865

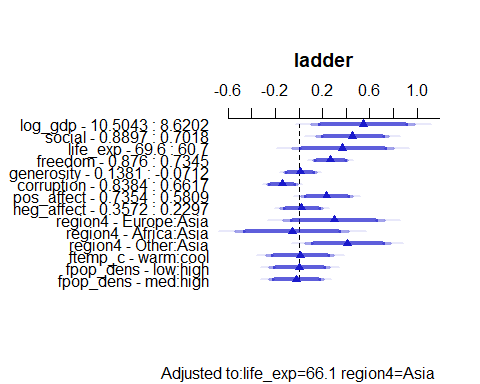
## Summary of fit5 After Imputation

summary(fit5\_imp)

Effects Response : ladder   
  
 Factor Low High Diff. Effect S.E.   
 log\_gdp 8.620200 10.50400 1.88400 0.5436500 0.223720  
 social 0.701850 0.88966 0.18782 0.4529600 0.157220  
 life\_exp 60.700000 69.60000 8.90000 0.3699400 0.218700  
 freedom 0.734450 0.87598 0.14153 0.2661000 0.075504  
 generosity -0.071157 0.13805 0.20921 0.0098983 0.069622  
 corruption 0.661680 0.83839 0.17671 -0.1436900 0.065373  
 pos\_affect 0.580940 0.73544 0.15449 0.2361900 0.111160  
 neg\_affect 0.229710 0.35724 0.12753 0.0240240 0.090926  
 region4 - Europe:Asia 1.000000 2.00000 NA 0.2976000 0.219740  
 region4 - Africa:Asia 1.000000 3.00000 NA -0.0570550 0.244500  
 region4 - Other:Asia 1.000000 4.00000 NA 0.4117700 0.183540  
 ftemp\_c - warm:cool 1.000000 2.00000 NA 0.0110450 0.145230  
 fpop\_dens - low:high 3.000000 1.00000 NA 0.0046554 0.130340  
 fpop\_dens - med:high 3.000000 2.00000 NA -0.0244240 0.117570  
 Lower 0.95 Upper 0.95  
 0.100580 0.986710   
 0.141590 0.764320   
 -0.063184 0.803050   
 0.116570 0.415630   
 -0.127980 0.147780   
 -0.273160 -0.014225   
 0.016054 0.456330   
 -0.156050 0.204100   
 -0.137590 0.732790   
 -0.541270 0.427160   
 0.048282 0.775260   
 -0.276570 0.298660   
 -0.253470 0.262780   
 -0.257260 0.208410   
  
Adjusted to: life\_exp=66.1 region4=Asia

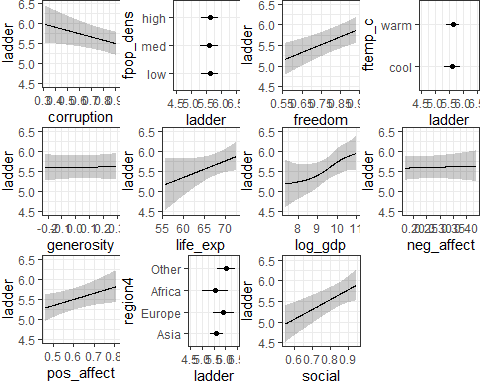
## fit5 Effects Plot after imputation

plot(summary(fit5\_imp))



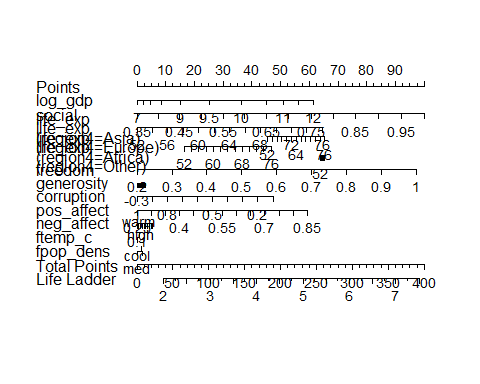
## Prediction Plot: fit5 post-imputation

ggplot(Predict(fit5\_imp))



## Nomogram of fit5 after imputation

plot(nomogram(fit5\_imp), lplabel = "Life Ladder")



## fit5 Bootstrap Validation after Imputation

set.seed(43229)  
validate(fit5\_imp, method = "boot", B = 300)

index.orig training test optimism index.corrected n  
R-square 0.8371 0.8613 0.8006 0.0607 0.7764 300  
MSE 0.2100 0.1734 0.2571 -0.0837 0.2937 300  
g 1.1819 1.1826 1.1656 0.0170 1.1648 300  
Intercept 0.0000 0.0000 0.1305 -0.1305 0.1305 300  
Slope 1.0000 1.0000 0.9753 0.0247 0.9753 300

# A Big Comparison, across all five models

## Summaries in our training sample

Training sample: *n* = 110 countries after single imputation.

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| model df | 3 | 3 | 12 | 14 | 20 |
| raw R-sq | 0.788 | 0.631 | 0.829 | 0.843 | **0.847** |
| adj. R-sq | 0.782 | 0.620 | 0.808 | **0.820** | 0.813 |
| RMSE | 0.541 | 0.713 | 0.486 | 0.465 | **0.459** |
| MAE | 0.412 | 0.543 | 0.371 | 0.344 | **0.342** |
| AIC | 187.0 | 247.9 | 181.4 | **175.6** | 184.8 |
| BIC | 200.5 | 261.4 | 219.2 | 218.8 | **196.4** |

## Bootstrapped Calibration Summaries

Training sample: *n* = 110 countries after single imputation.

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| Mean absolute error | 0.075 | **0.032** | 0.055 | 0.057 | 0.049 |
| Mean squared error | 0.00884 | **0.00201** | 0.00435 | 0.00508 | 0.00434 |
| 90th quantile abs error | 0.146 | **0.058** | 0.101 | 0.103 | 0.122 |

* Model fit2 looks like it’s the best calibrated of these.
* All of the models have at least some issues with regression assumptions.

## Comparing R-square estimates

These aren’t cross-validated or bootstrap validated. These are just the raw values.

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| Single imp | 0.788 | 0.631 | 0.829 | 0.843 | **0.847** |
| MI with mice | 0.775 | 0.569 | 0.813 | 0.830 | **0.837** |
| MI with areg | 0.782 | 0.586 | 0.823 | 0.837 | **0.841** |

## Bootstrap-Validated Summaries

* After single imputation:

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| R-sq | 0.776 | 0.608 | 0.786 | **0.794** | 0.768 |
| MSE | 0.316 | 0.552 | 0.299 | **0.291** | 0.322 |

* After multiple imputation with aregImpute():

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| R-sq | 0.774 | 0.506 | 0.783 | **0.792** | 0.776 |
| MSE | 0.298 | 0.636 | 0.287 | **0.271** | 0.294 |

## Test Sample Error Summaries

Test sample: *n* = 28 countries after single imputation.

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| MAPE | 0.388 | 0.812 | 0.388 | 0.388 | **0.370** |
| max APE | 1.122 | 3.518 | **0.896** | 1.003 | 0.985 |
| RMSPE | 0.509 | 1.235 | **0.458** | 0.475 | 0.469 |
| R-sq (val.) | 0.744 | 0.142 | **0.792** | 0.770 | 0.777 |

## Other Examples to Consult

* The support1000 example in our Shared Google Drive
* The [Project A demonstration project](https://thomaselove.github.io/432-2025/432_projectA_demo.html)
* [432 Course Notes](https://thomaselove.github.io/432-notes): examples in Chapters 12-15 are especially relevant.
* See <https://hbiostat.org/rmsc/> for Frank Harrell’s Regression Modeling Strategies text

## Session Information

xfun::session\_info()

R version 4.4.2 (2024-10-31 ucrt)  
Platform: x86\_64-w64-mingw32/x64  
Running under: Windows 11 x64 (build 22631)  
  
Locale:  
 LC\_COLLATE=English\_United States.utf8   
 LC\_CTYPE=English\_United States.utf8   
 LC\_MONETARY=English\_United States.utf8  
 LC\_NUMERIC=C   
 LC\_TIME=English\_United States.utf8   
  
Package version:  
 abind\_1.4-8 askpass\_1.2.1 backports\_1.5.0   
 base64enc\_0.1-3 bayestestR\_0.15.1 bigD\_0.3.0   
 bit\_4.5.0.1 bit64\_4.6.0.1 bitops\_1.0.9   
 blob\_1.2.4 boot\_1.3-31 broom\_1.0.7   
 bslib\_0.9.0 cachem\_1.1.0 callr\_3.7.6   
 car\_3.1-3 carData\_3.0-5 caret\_7.0-1   
 cellranger\_1.1.0 checkmate\_2.3.2 chk\_0.10.0   
 class\_7.3-22 cli\_3.6.3 clipr\_0.8.0   
 clock\_0.7.2 cluster\_2.1.6 cobalt\_4.5.5   
 coda\_0.19-4.1 codetools\_0.2-20 colorspace\_2.1-1   
 commonmark\_1.9.2 compiler\_4.4.2 conflicted\_1.2.0   
 correlation\_0.8.6 cowplot\_1.1.3 cpp11\_0.5.1   
 crayon\_1.5.3 curl\_6.2.0 cutpointr\_1.2.0   
 data.table\_1.16.4 datasets\_4.4.2 datawizard\_1.0.0   
 DBI\_1.2.3 dbplyr\_2.5.0 Deriv\_4.1.6   
 diagram\_1.6.5 digest\_0.6.37 doBy\_4.6.25   
 dplyr\_1.1.4 dtplyr\_1.3.1 e1071\_1.7.16   
 easystats\_0.7.3 effectsize\_1.0.0 emmeans\_1.10.7   
 estimability\_1.5.1 evaluate\_1.0.3 fansi\_1.0.6   
 farver\_2.1.2 fastmap\_1.2.0 fontawesome\_0.5.3   
 forcats\_1.0.0 foreach\_1.5.2 foreign\_0.8-88   
 Formula\_1.2-5 fs\_1.6.5 future\_1.34.0   
 future.apply\_1.11.3 gargle\_1.5.2 generics\_0.1.3   
 ggformula\_0.12.0 ggplot2\_3.5.1 ggrepel\_0.9.6   
 ggridges\_0.5.6 glmnet\_4.1-8 globals\_0.16.3   
 glue\_1.8.0 goftest\_1.2-3 googledrive\_2.1.1   
 googlesheets4\_1.1.1 gower\_1.0.2 graphics\_4.4.2   
 grDevices\_4.4.2 grid\_4.4.2 gridExtra\_2.3   
 gt\_0.11.1 gtable\_0.3.6 hardhat\_1.4.1   
 haven\_2.5.4 highr\_0.11 Hmisc\_5.2-2   
 hms\_1.1.3 htmlTable\_2.4.3 htmltools\_0.5.8.1   
 htmlwidgets\_1.6.4 httpuv\_1.6.15 httr\_1.4.7   
 ids\_1.0.1 insight\_1.0.1 ipred\_0.9-15   
 isoband\_0.2.7 iterators\_1.0.14 janitor\_2.2.1   
 jomo\_2.7-6 jquerylib\_0.1.4 jsonlite\_1.8.9   
 juicyjuice\_0.1.0 KernSmooth\_2.23.24 knitr\_1.49   
 labeling\_0.4.3 labelled\_2.14.0 later\_1.4.1   
 lattice\_0.22-6 lava\_1.8.1 lifecycle\_1.0.4   
 listenv\_0.9.1 lme4\_1.1-36 lubridate\_1.9.4   
 magrittr\_2.0.3 markdown\_1.13 MASS\_7.3-64   
 Matrix\_1.7-1 MatrixModels\_0.5-3 memoise\_2.0.1   
 methods\_4.4.2 mgcv\_1.9-1 mice\_3.17.0   
 microbenchmark\_1.5.0 mime\_0.12 minqa\_1.2.8   
 mitml\_0.4-5 modelbased\_0.8.9 ModelMetrics\_1.2.2.2  
 modelr\_0.1.11 mosaic\_1.9.1 mosaicCore\_0.9.4.0   
 mosaicData\_0.20.4 multcomp\_1.4-28 munsell\_0.5.1   
 mvtnorm\_1.3-3 naniar\_1.1.0 nlme\_3.1-166   
 nloptr\_2.1.1 nnet\_7.3-20 norm\_1.0.11.1   
 nortest\_1.0-4 numDeriv\_2016.8.1.1 olsrr\_0.6.1   
 openssl\_2.3.2 ordinal\_2023.12.4.1 pan\_1.9   
 parallel\_4.4.2 parallelly\_1.42.0 parameters\_0.24.1   
 patchwork\_1.3.0 pbkrtest\_0.5.3 performance\_0.13.0   
 pillar\_1.10.1 pkgconfig\_2.0.3 plyr\_1.8.9   
 polspline\_1.1.25 prettyunits\_1.2.0 pROC\_1.18.5   
 processx\_3.8.5 prodlim\_2024.06.25 progress\_1.2.3   
 progressr\_0.15.1 promises\_1.3.2 proxy\_0.4.27   
 ps\_1.8.1 purrr\_1.0.2 quantreg\_6.00   
 R6\_2.5.1 ragg\_1.3.3 rappdirs\_0.3.3   
 rbibutils\_2.3 RColorBrewer\_1.1.3 Rcpp\_1.0.14   
 RcppEigen\_0.3.4.0.2 Rdpack\_2.6.2 reactable\_0.4.4   
 reactR\_0.6.1 readr\_2.1.5 readxl\_1.4.3   
 recipes\_1.1.0 reformulas\_0.4.0 rematch\_2.0.0   
 rematch2\_2.1.2 report\_0.6.0 reprex\_2.1.1   
 reshape2\_1.4.4 rlang\_1.1.5 rmarkdown\_2.29   
 rms\_7.0-0 rpart\_4.1.24 rstudioapi\_0.17.1   
 rvest\_1.0.4 sandwich\_3.1-1 sass\_0.4.9   
 scales\_1.3.0 see\_0.10.0 selectr\_0.4.2   
 shape\_1.4.6.1 shiny\_1.10.0 snakecase\_0.11.1   
 sourcetools\_0.1.7.1 SparseM\_1.84-2 sparsevctrs\_0.2.0   
 splines\_4.4.2 SQUAREM\_2021.1 stats\_4.4.2   
 stats4\_4.4.2 stringi\_1.8.4 stringr\_1.5.1   
 survival\_3.8-3 sys\_3.4.3 systemfonts\_1.2.1   
 textshaping\_1.0.0 TH.data\_1.1-3 tibble\_3.2.1   
 tidyr\_1.3.1 tidyselect\_1.2.1 tidyverse\_2.0.0   
 timechange\_0.3.0 timeDate\_4041.110 tinytex\_0.54   
 tools\_4.4.2 tzdb\_0.4.0 ucminf\_1.2.2   
 UpSetR\_1.4.0 utf8\_1.2.4 utils\_4.4.2   
 uuid\_1.2.1 V8\_6.0.1 vctrs\_0.6.5   
 viridis\_0.6.5 viridisLite\_0.4.2 visdat\_0.6.0   
 vroom\_1.6.5 withr\_3.0.2 xfun\_0.50   
 xml2\_1.3.6 xplorerr\_0.2.0 xtable\_1.8-4   
 yaml\_2.3.10 zoo\_1.8-12

1. We’re doing this for teaching purposes, not because it’s generally a good idea. [↑](#footnote-ref-32)
2. This table was hand-crafted by Dr. Love. [↑](#footnote-ref-77)