Supervised Learning

Linear regression

June 22th, 2023

Simple linear regression

We assume a **linear relationship** for $Y = f(X)$:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad \text{for } i = 1, 2, \ldots, n$$

- Y_i is the ith value for the **response** variable
- X_i is the ith value for the **predictor** variable
- β_0 is an *unknown*, constant **intercept**: average value for Y if $X = 0$
- β_1 is an *unknown*, constant **slope**: increase in average value for Y for each one-unit increase in X
- ϵ_i is the **random** noise: assume **independent, identically distributed** (*iid*) from Normal distribution

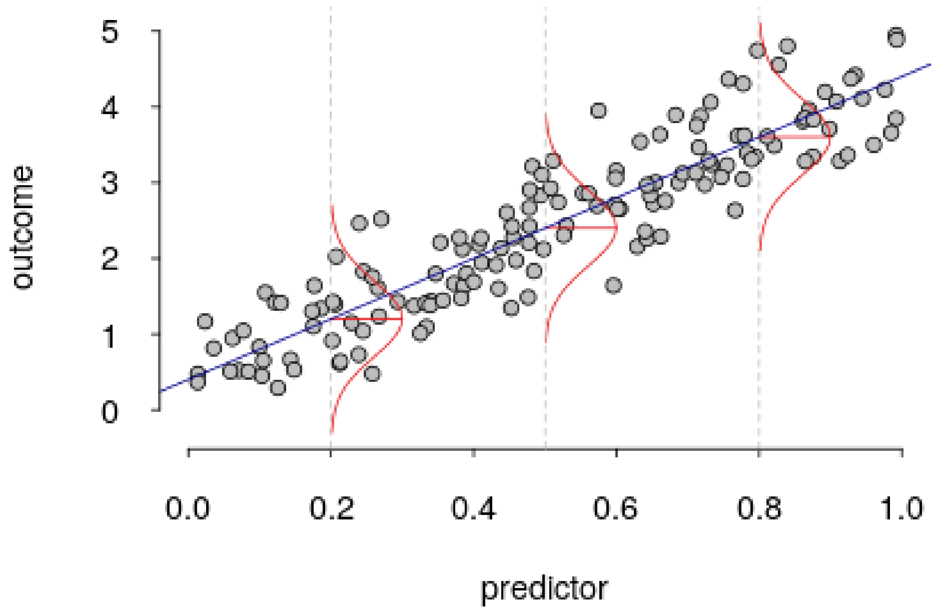
$$\epsilon_i \overset{iid}{\sim} N(0, \sigma^2) \quad \text{with constant variance } \sigma^2$$

Simple linear regression estimation

We are estimating the **conditional expection (mean)** for Y:

$$\mathbb{E}[Y_i|X_i] = \beta_0 + \beta_1 X_i$$

- average value for Y given the value for X
- averaging out the error ϵ (disappears because ϵ has mean 0)



How do we estimate the **best fitting** line?

Ordinary least squares (OLS) - by minimizing the **residual sum of squares (RSS)**

$$RSS\left(\beta_0, \beta_1\right) = \sum_{i=1}^{n} \left[Y_i - \left(\beta_0 + \beta_1 X_i\right)\right]^2 = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_i\right)^2$$

Remember MSE? $\frac{1}{n} \sum_i^n (Y_i - \hat{f}\left(X_i\right))^2$

RSS is similar: not a mean (no $\frac{1}{n}$), but it is the sum of the squared differences

$f(X)$ in this case is the model specified before: $\beta_0 - \beta_1 X_i$

Minimized at

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sum_{i=1}^n \left(X_i - \bar{X}\right)^2} \quad \text{and} \quad \widehat{\beta}_0 = \bar{Y} - \widehat{\beta}_1 \bar{X}$$

where $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$ and $\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$

Connection to covariance and correlation

Covariance describes the **joint variability of two variables**

$$\mathrm{Cov}(X,Y) = \sigma_{X,Y} = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

We compute the **sample covariance** (use $n-1$ since we are using the means and want **unbiased estimates**)

$$\hat{\sigma}_{X,Y} = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)$$

Correlation is a *normalized* form of covariance, ranges from -1 to 1

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \cdot \sigma_Y}$$

Sample correlation uses the sample covariance and standard deviations, e.g. $s_X^2 = \frac{1}{n-1}\sum_i (X_i - \bar{X})^2$

$$r_{X,Y} = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2 \sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2}}$$

Connection to covariance and correlation

So we have the following:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sum_{i=1}^n \left(X_i - \bar{X}\right)^2} \quad \text{compared to} \quad r_{X,Y} = \frac{\sum_{i=1}^n \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^n \left(X_i - \bar{X}\right)^2 \sum_{i=1}^n \left(Y_i - \bar{Y}\right)^2}}$$

$\Rightarrow$ Can rewrite $\hat{\beta}_1$ as:

$$\widehat{\beta}_1 = r_{X,Y} \cdot \frac{s_Y}{s_X}$$

$\Rightarrow$ Can rewrite $r_{X,Y}$ as:

$$r_{X,Y} = \widehat{\beta}_1 \cdot \frac{s_X}{s_Y}$$

Can think of $\widehat{\beta}_1$ weighting the ratio of variance between X and Y…

Gapminder data

Health and income outcomes for 184 countries from 1960 to 2016 from the famous Gapminder project

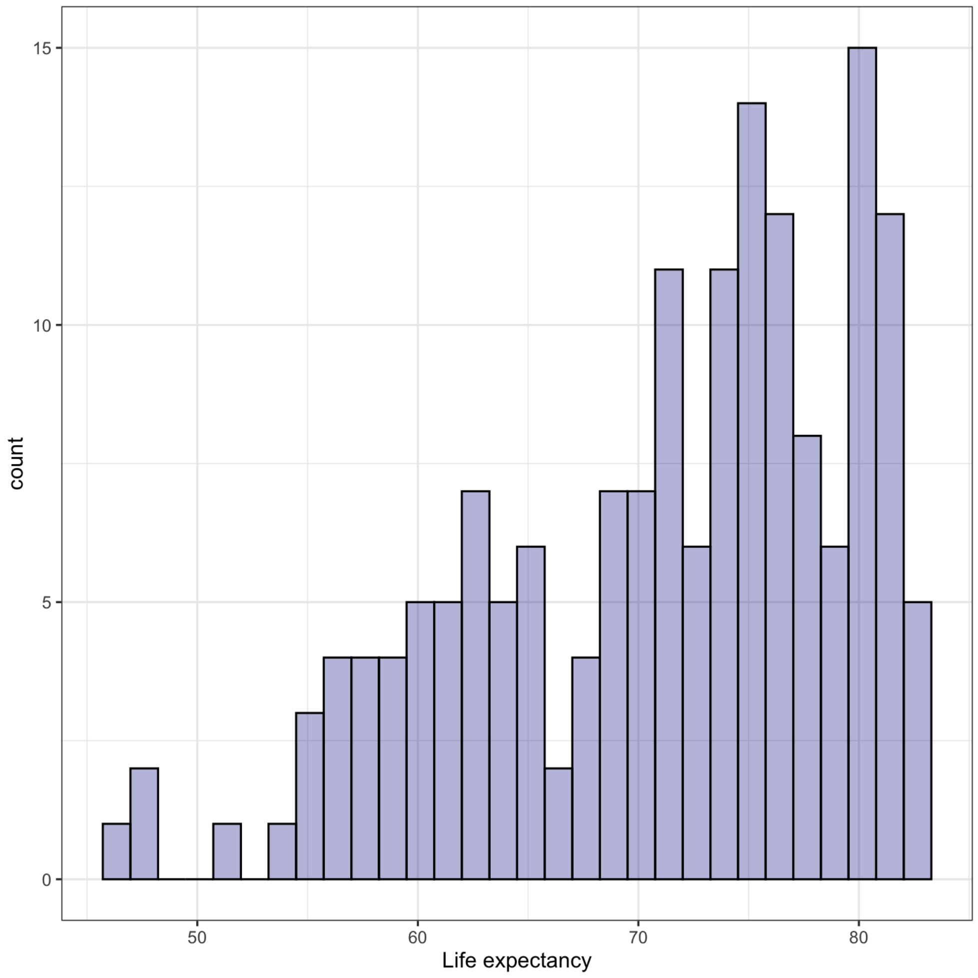
```
library(tidyverse)
library(dslabs)
gapminder <- as_tibble(gapminder)
clean_gapminder <- gapminder %>%
  filter(year == 2011, !is.na(gdp)) %>%
  mutate(log_gdp = log(gdp))
clean_gapminder
```

```
## # A tibble: 168 × 10
##    country   year infan…¹ life_…² ferti…³ popul…⁴     gdp conti…⁵ region log_gdp
##    <fct>    <int>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <fct>   <fct>    <dbl>
##  1 Albania   2011    14.3    77.4    1.75  2.89e6 6.32e 9 Europe  South…    22.6
##  2 Algeria   2011    22.8    76.1    2.83  3.67e7 8.11e10 Africa  North…    25.1
##  3 Angola    2011   107.     58.1    6.1   2.19e7 2.70e10 Africa  Middl…    24.0
##  4 Antigua…  2011     7.2    75.9    2.12  8.82e4 8.02e 8 Americ… Carib…    20.5
##  5 Argenti…  2011    12.7    76      2.2   4.17e7 4.73e11 Americ… South…    26.9
##  6 Armenia   2011    15.3    73.5    1.5   2.97e6 4.29e 9 Asia    Weste…    22.2
##  7 Austral…  2011     3.8    82.2    1.88  2.25e7 5.73e11 Oceania Austr…    27.1
##  8 Austria   2011     3.4    80.7    1.44  8.42e6 2.31e11 Europe  Weste…    26.2
##  9 Azerbai…  2011    32.5    70.8    1.96  9.23e6 2.14e10 Asia    Weste…    23.8
## 10 Bahamas   2011    11.1    72.6    1.9   3.67e5 6.76e 9 Americ… Carib…    22.6
```

Modeling life expectancy

Interested in modeling a country's **life expectancy**

```
clean_gapminder %>%
  ggplot(aes(x = life_expectancy)) +
  geom_histogram(color = "black",
                 fill = "darkblue",
                 alpha = 0.3) +
  theme_bw() +
  labs(x = "Life expectancy")
```

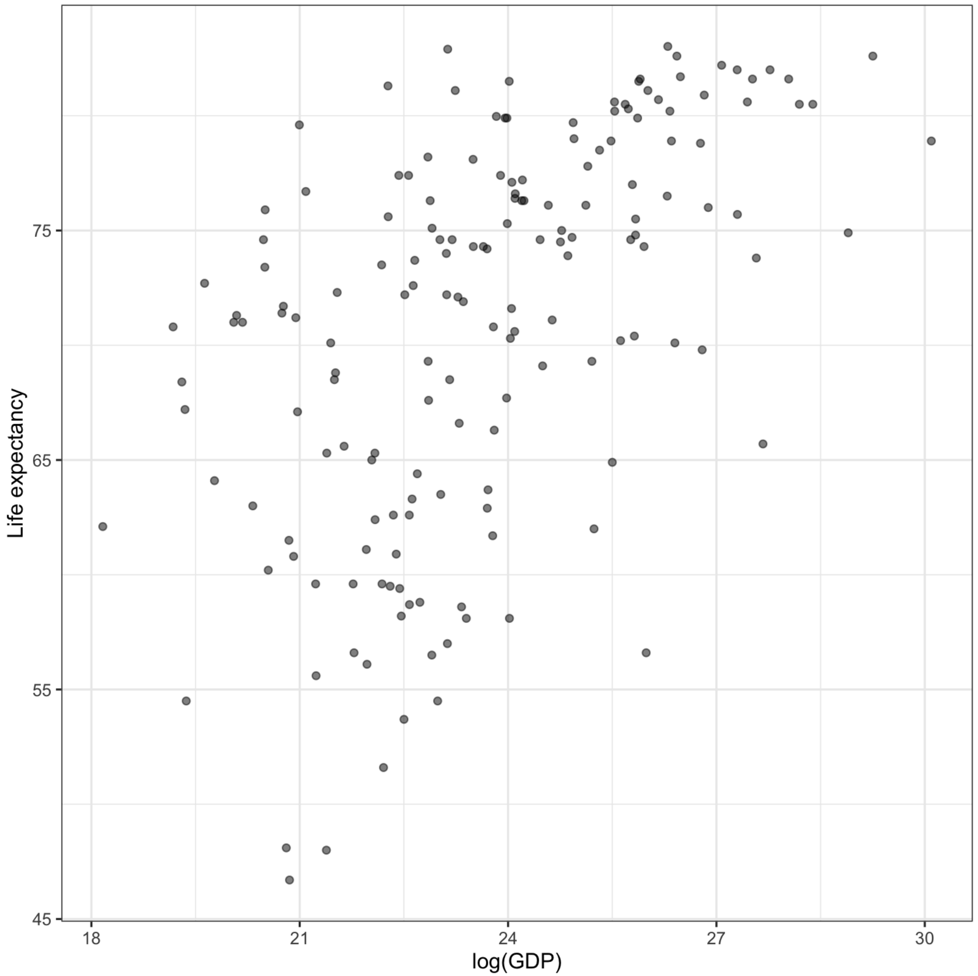


Relationship between life expectancy and log(GDP)

```
gdp_plot <- clean_gapminder %>%
  ggplot(aes(x = log_gdp,
             y = life_expectancy)) +
  geom_point(alpha = 0.5) +
  theme_bw() +
  labs(x = "log(GDP)",
       y = "Life expectancy")
gdp_plot
```

We fit linear regression models using `lm()`, formula is input as: `response ~ predictor`

```
init_lm <- lm(life_expectancy ~ log_gdp,
              data = clean_gapminder)
```



View the model `summary()`

```
summary(init_lm)
```

```
##
## Call:
## lm(formula = life_expectancy ~ log_gdp, data = clean_gapminder)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.901  -4.781   1.879   5.335  13.962
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   24.174      5.758   4.198 4.38e-05 ***
## log_gdp        1.975      0.242   8.161 7.87e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.216 on 166 degrees of freedom
## Multiple R-squared:  0.2864,	Adjusted R-squared:  0.2821
## F-statistic: 66.61 on 1 and 166 DF,  p-value: 7.865e-14
```

Inference with OLS

Reports the intercept and coefficient estimates: $\hat{\beta}_0 \approx 24.174 \quad , \quad \hat{\beta}_1 \approx 1.975$

Estimates of uncertainty for βs via **standard errors**: $\widehat{SE}(\hat{\beta}_0) \approx 5.758 \quad , \quad \widehat{SE}(\hat{\beta}_1) \approx 0.242$

t-statistics are coefficients `Estimates` / `Std. Error`, i.e., number of standard deviations from 0

- *p-values* (i.e., `Pr(>|t|)`): estimated probability observing value as extreme as `|t value|` **given the null hypothesis** $\beta = 0$
- p-value $<$ conventional threshold of $\alpha = 0.05$, **sufficient evidence to reject the null hypothesis that the coefficient is zero**,
- Typically `|t values|` > 2 indicate **significant** relationship at $\alpha = 0.05$
- i.e., there is a **significant** association between `life_expectancy` and `log_gdp`

Be careful!

Caveats to keep in mind regarding p-values:

If the true value of a coefficient $\beta = 0$, then the p-value is sampled from a Uniform(0,1) distribution

- i.e., it is just as likely to have value 0.45 as 0.16 or 0.84 or 0.9999 or 0.00001...

$\Rightarrow$ Hence why we typically only reject for low α values like 0.05

- Controlling the Type 1 error rate at $\alpha = 0.05$, i.e., the probability of a **false positive** mistake
- 5% chance that you'll conclude there's a significant association between x and y **even when there is none**

Remember what a standard error is? $SE = \frac{\sigma}{\sqrt{n}}$

- $\Rightarrow$ As n gets large **standard error goes to zero**, and *all* predictors are eventually deemed significant
- While the p-values might be informative, we will explore other approaches to determine which subset of predictors to include (e.g., holdout performance)

Back to the model summary: `Multiple R-squared`

Back to the connection between the coefficient and correlation:

$$r_{X,Y} = \widehat{\beta}_1 \cdot \frac{s_X}{s_Y} \quad \Rightarrow \quad r^2_{X,Y} = \widehat{\beta}_1^2 \cdot \frac{s^2_X}{s^2_Y}$$

Compute the correlation with `cor()`:

```
with(clean_gapminder, cor(log_gdp, life_expectancy))
```

```
## [1] 0.5351189
```

The squared `cor` matches the reported `Multiple R-squared`

```
with(clean_gapminder, cor(log_gdp, life_expectancy))^2
```

```
## [1] 0.2863522
```

Back to the model summary: `Multiple R-squared`

Back to the connection between the coefficient and correlation:

$$r_{X,Y} = \widehat{\beta}_1 \cdot \frac{s_X}{s_Y} \quad \Rightarrow \quad r^2_{X,Y} = \widehat{\beta}_1^2 \cdot \frac{s^2_X}{s^2_Y}$$

r^2 (or also R^2) estimates the **proportion of the variance** of Y explained by X

- More generally: variance of model predictions / variance of Y

```
var(predict(init_lm)) / var(clean_gapminder$life_expectancy)
```

```
## [1] 0.2863522
```

Generating predictions

We can use the `predict()` function to either get the fitted values of the regression:

```
train_preds <- predict(init_lm)
head(train_preds)
```

```
##        1        2        3        4        5        6
## 68.74401 73.78465 71.61243 64.66585 77.26605 67.97876
```

Which is equivalent to using:

```
head(init_lm$fitted.values)
```

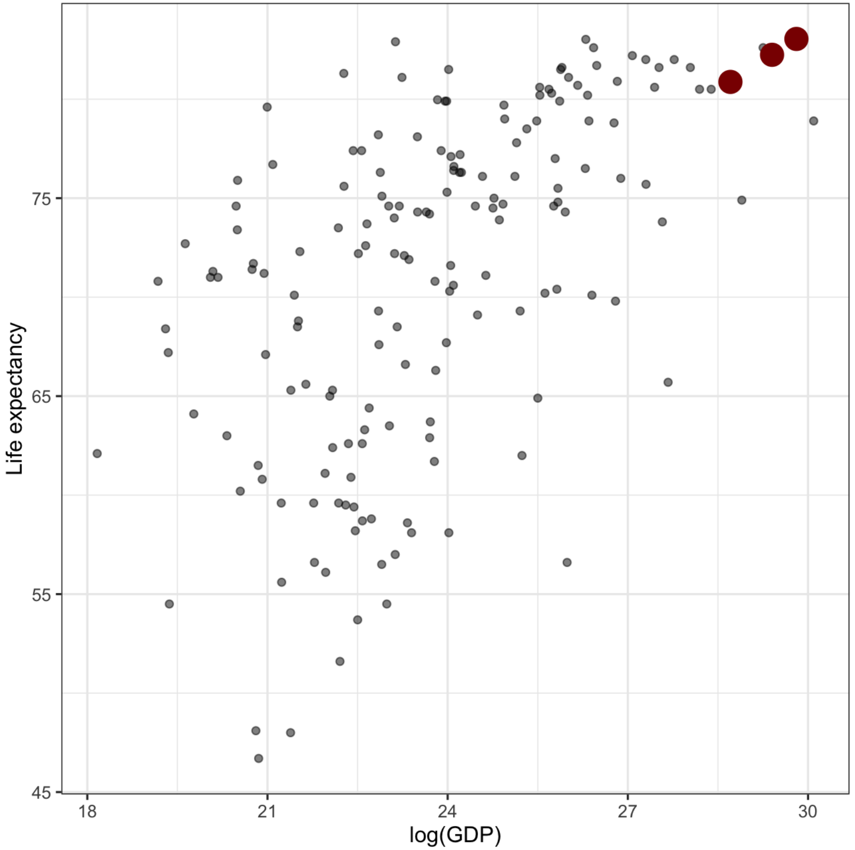
```
##        1        2        3        4        5        6
## 68.74401 73.78465 71.61243 64.66585 77.26605 67.97876
```

Predictions for new data

Or we can provide it `newdata` which **must contain the explanatory variables**:

```
us_data <- clean_gapminder %>%
  filter(country == "United States")

new_us_data <- us_data %>%
  dplyr::select(country, gdp) %>%
  slice(rep(1, 3)) %>%
  mutate(adj_factor = c(0.25, 0.5, 0.75),
         log_gdp = log(gdp * adj_factor))
new_us_data$pred_life_exp <-
  predict(init_lm, newdata = new_us_data)
gdp_plot +
  geom_point(data = new_us_data,
             aes(x = log_gdp,
                 y = pred_life_exp),
             color = "darkred", size = 5)
```

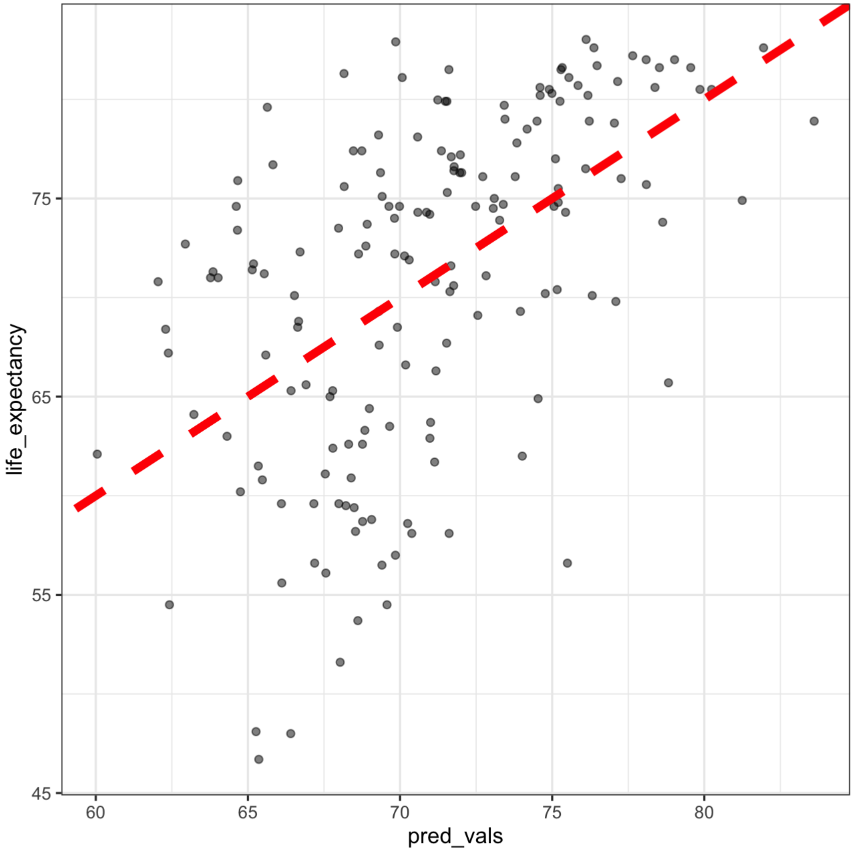


Plot observed values against predictions

Useful diagnostic (for **any type of model**, not just linear regression!)

```
clean_gapminder %>%
  mutate(pred_vals = predict(init_lm)) %>%
  ggplot(aes(x = pred_vals,
             y = life_expectancy)) +
  geom_point(alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0,
              linetype = "dashed",
              color = "red",
              size = 2) +
  theme_bw()
```

- "Perfect" model will follow **diagonal**

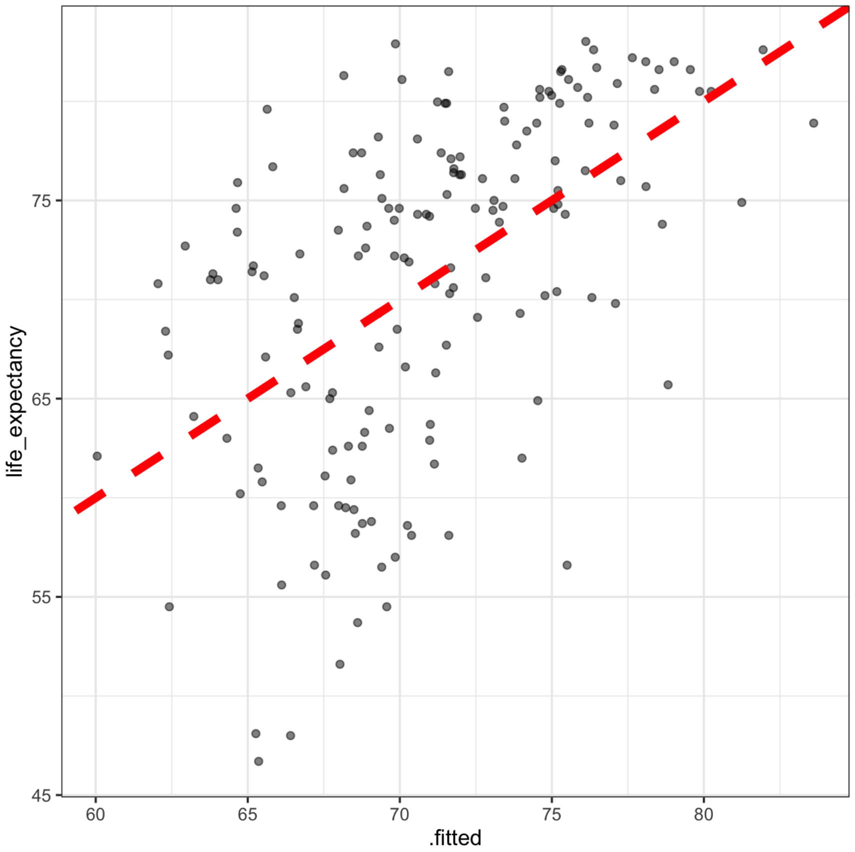


Plot observed values against predictions

Can augment the data with model output using the `broom` package

```
clean_gapminder <-
  broom::augment(init_lm, clean_gapminder)
clean_gapminder %>%
  ggplot(aes(x = .fitted,
             y = life_expectancy)) +
  geom_point(alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0,
              linetype = "dashed",
              color = "red",
              size = 2) +
  theme_bw()
```

- Adds various columns from model fit we can use in plotting for model diagnostics

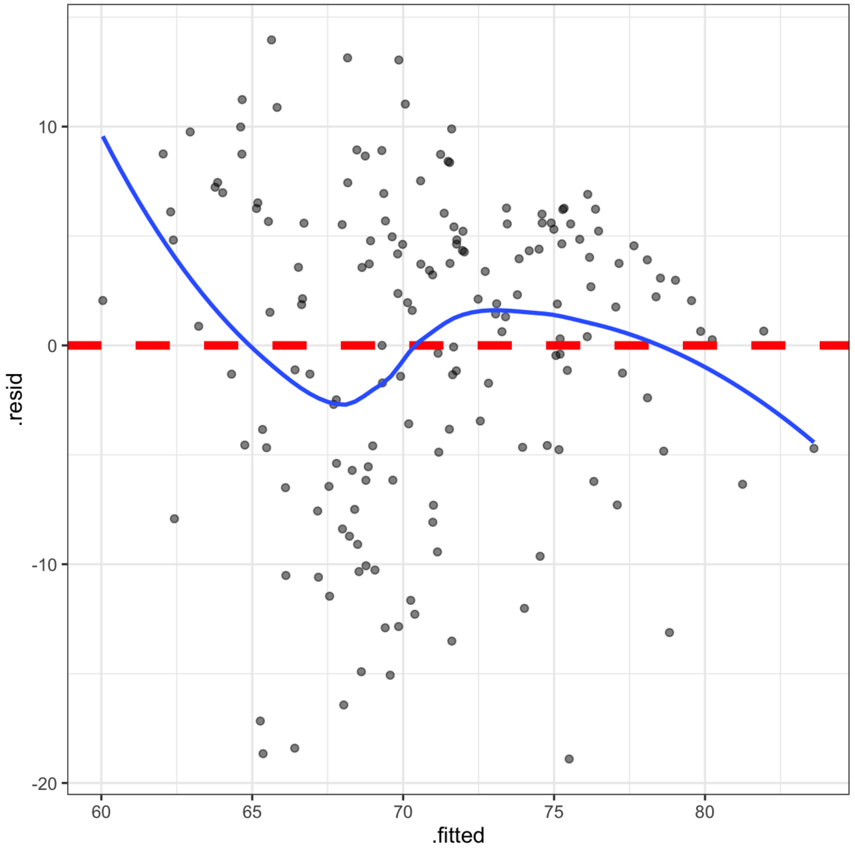


Plot residuals against predicted values

- Residuals = observed - predicted
- Conditional on the predicted values, the **residuals should have a mean of zero**

```
clean_gapminder %>%
  ggplot(aes(x = .fitted,
             y = .resid)) +
  geom_point(alpha = 0.5) +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "red",
             size = 2) +
  # To plot the residual mean
  geom_smooth(se = FALSE) +
  theme_bw()
```

- Residuals **should NOT display any pattern**



Multiple regression

We can include as many variables as we want (assuming $n > p$!)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

OLS estimates in matrix notation ($\boldsymbol{X}$ is a $n \times p$ matrix):

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}$$

Can just add more variables to the formula in `R`

```
multiple_lm <- lm(life_expectancy ~ log_gdp + fertility,
                  data = clean_gapminder)
```

- Use the `Adjusted R-squared` when including multiple variables $= 1 - \frac{(1-R^2)(n-1)}{(n-p-1)}$
  - Adjusts for the number of variables in the model p
  - Adding more variables **will always increase** `Multiple R-squared`

What about the Normal distribution assumption???

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

- ϵ_i is the **random** noise: assume **independent, identically distributed** (*iid*) from Normal distribution

$$\epsilon_i \overset{iid}{\sim} N(0, \sigma^2) \quad \text{with constant variance } \sigma^2$$

OLS doesn't care about this assumption, it's just estimating coefficients!

In order to perform inference, **we need to impose additional assumptions**

By assuming $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, what we really mean is:

$$Y \overset{iid}{\sim} N(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p, \sigma^2)$$

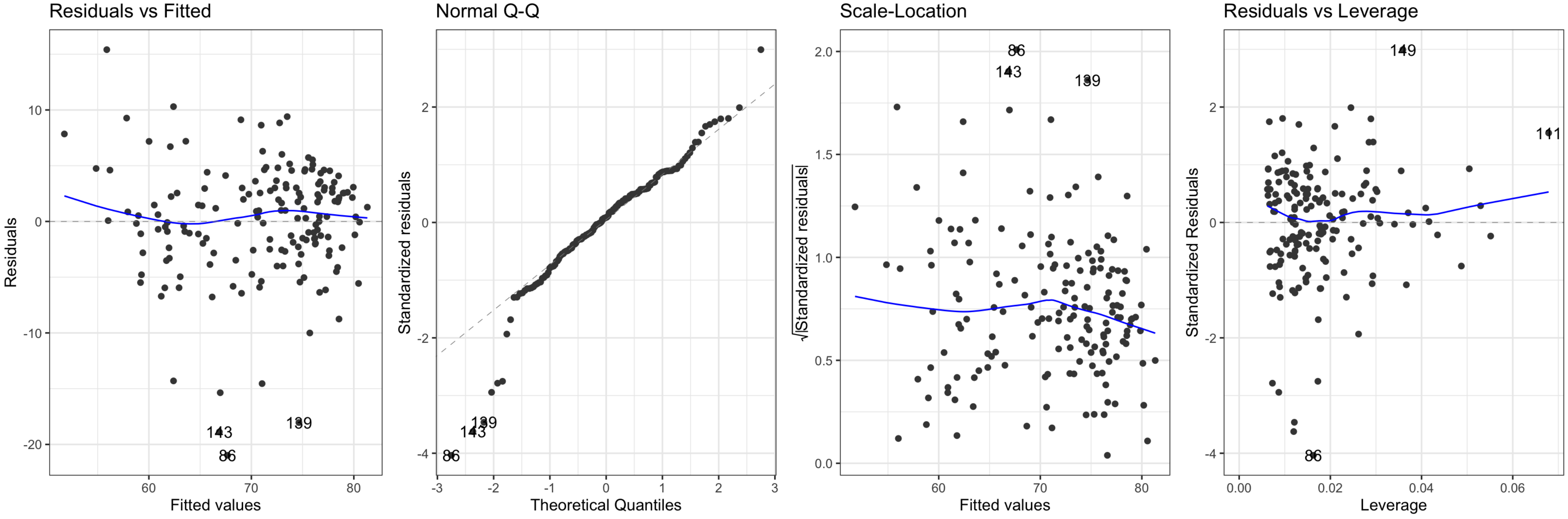
So we're estimating the mean μ of this conditional distribution, but what about σ^2?

Unbiased estimate $\hat{\sigma}^2 = \frac{RSS}{n-(p+1)}$, its square root is the `Residual standard error`

- **Degrees of freedom**: $n - (p+1)$, data supplies us with n "degrees of freedom" and we used up $p+1$

Check the assumptions about normality with `ggfortify`

```r
library(ggfortify)
autoplot(multiple_lm, ncol = 4) + theme_bw()
```



- Standardized residuals = residuals `/ sd(residuals)` (see also `.std.resid` from `augment`)