Regression

Estimating models and predicting values with R

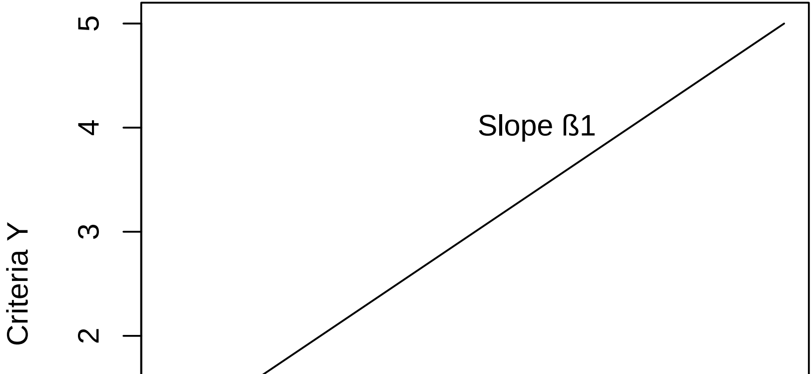
Goals

- An introduction to regression analyses
- An introduction to fitting regression models in R

Regression

$$y_i = \beta_0 + \beta_1 X_i + e_i$$

y = Criteria variable
i = Subject number (measurement number)
β_0 = Intercept of y
β_1 = Weight of predictor X
X = Predictor variable
e = Error term

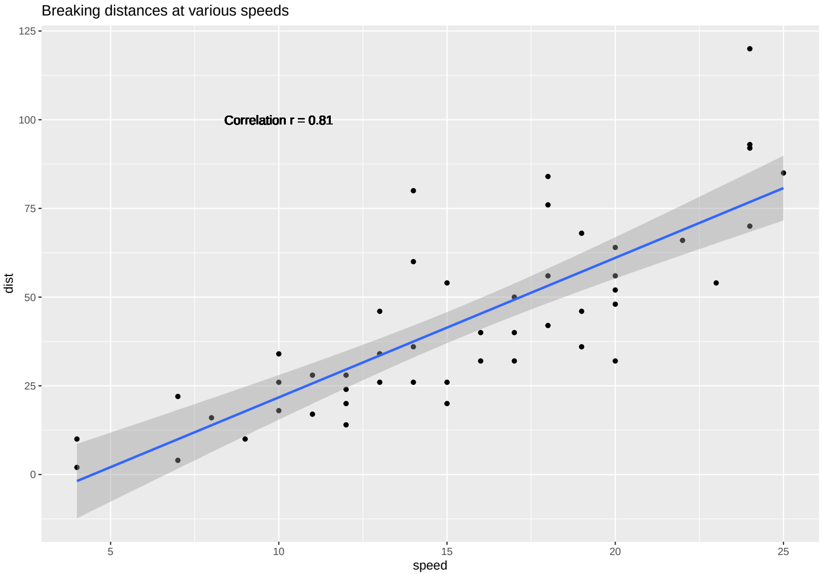


`lm()` function

- The `lm()` function fits a regression model.
- `lm(formula, data)`
- ***Formulas*** are a basic data type that is applied in many R functions.
- Basic structur: ***dependent variable*** ~ ***explanatory variables*** (e.g. ***y*** ~ ***x1*** + ***x2***)
- `data` takes a dataframe

```
lm(dist ~ speed, data = cars)
```

Example



Breaking distances at various speeds
Correlation r = 0.81
dist
speed
125
100
75
50
25
0
5
10
15
20
25

```r
fit <- lm(dist ~ speed, data = cars)
fit
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##     -17.579        3.932
```

$$dist_i = -17.579 + 3.932 * speed_i$$

Summary

```r
summary(fit)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.643
```

Task

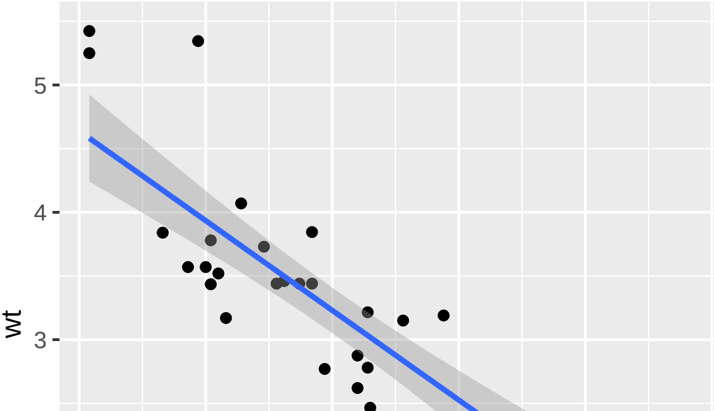
- Take the `mtcars` dataset
- Calculate the correlation of mileage `mpg` and car weight `wt`
- Plot a scatterplot with mileage on x and weight on y
- Add a regression line with (`geom_smooth(method = "lm")`)
- Regress mileage `mpg` on car weight `wt` (that is, predic mileage by means of weight)

Task

```
cor(mtcars$mpg, mtcars$wt)
```

```
## [1] -0.8676594
```

```
ggplot(mtcars, aes(x = mpg, y = wt)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Multiple predictors

- A `formula` can take multiple predictors: `y ~ x1 + x2`
- An Interaction describes a relationi between two (or more) predictors where influce of one predictor changes with the value of the other predictor (e.g. weight and smoking influence blood preassure. And the influence of smoking is even highe for those who are overweight)
- An interaction is modelled with a `:` sign: $y \sim x1 + x2 + x1{:}x2$

Task

- Take the `mtcars` dataset
- Predict mileage `mpg` by weight `wt` and number of cylinders `cyl`
- Take the interaction into account
- Discuss with your seatmate how to interpret the estimates

```r
fit <- lm(mpg ~ wt + cyl + wt:cyl, data = mtcars)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ wt + cyl + wt:cyl, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2288 -1.3495 -0.5042  1.4647  5.2344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.3068     6.1275   8.863 1.29e-09 ***
## wt           -8.6556     2.3201  -3.731 0.000861 ***
## cyl          -3.8032     1.0050  -3.784 0.000747 ***
## wt:cyl        0.8084     0.3273   2.470 0.019882 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
```

Categorical data

- Predictors can be categorical variables
- They have to be `factors`for `formulas` to recognice them as `categorical`
- Remember: you can create a factor with the `factor()` function

Task

- Take the `mtcars` dataset
- Predict `mpg` by `wt` and the transmission type `am` and their interaction
- By deafult `am` is not a factor. Please create a factor for `am` first
- Discuss with you seatmate how to interpret the estimates

```r
mtcars$am_factor <- factor(mtcars$am, labels = c("Automati
fit <- lm(mpg ~ wt + am_factor + wt:am_factor, data = mtca
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ wt + am_factor + wt:am_factor, data =
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6004 -1.5446 -0.5325  0.9012  6.0909
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          31.4161     3.0201  10.402 4.00e-11
## wt                   -3.7859     0.7856  -4.819 4.55e-05
## am_factorManual      14.8784     4.2640   3.489  0.00162
## wt:am_factorManual   -5.2984     1.4447  -3.667  0.00102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.
```

Changing the contrast

- Contrasts describe how to compare the influnce of two (or more) factor levels.
- Two common ways are:
  - Treatment: One factor level is the baseline
  - Helmert: The average of the two factor levels are the baseline
- The `constrasts()` function gets and sets contrasts for factors
- Treatment contrasts for two factor levels:
  `contrasts(mtcars$am_factor) <- contr.treatment(2)`
- Helmert contrasts for two factor levels:
  `contrasts(mtcars$am_factor) <- contr.helmert(2)`

Task

- Take the `mtcars` dataset
- Predict `mpg` by `wt` and the transmission type `am` and their interaction
- By deafult `am` is not a factor. Please create a factor for `am` first
- Set the contrasts of the factor to *Helmert* and calculate the model.
- Set the contrasts of the factor to *Treatment* and calculate the model
- Compare the results of the two models and discuss them with you seatmate

```r
mtcars$am_factor <- factor(mtcars$am, labels = c("Automati
contrasts(mtcars$am_factor) <- contr.helmert(2)
fit1 <- lm(mpg ~ wt + am_factor + wt:am_factor, data = mtca
summary(fit1)
```

```
##
## Call:
## lm(formula = mpg ~ wt + am_factor + wt:am_factor, data =
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6004 -1.5446 -0.5325  0.9012  6.0909
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     38.8553     2.1320  18.225  < 2e-16 ***
## wt              -6.4351     0.7223  -8.909 1.16e-09 ***
## am_factor1       7.4392     2.1320   3.489  0.00162 **
## wt:am_factor1   -2.6492     0.7223  -3.667  0.00102 **
## ---
```

Multilevel regression formula

$y_{ij} = \beta_{0j} + \beta_{1j} X_{ij} + e_{ij}$

Level 2:
$\beta_{0j} = \gamma_{01} W_j + \upsilon_{0j}$
$\beta_{1j} = \gamma_{10} + \upsilon_{1j}$

y = Criteria variable
i = Subject number (measurement number)
β_0 = Intercept of y
β_1 = Weight of predictor X
X = Predictor variable
e = Error term
j = Level 2 group number
γ_{00} = Intercept
W = Level 2 predictor (grouping variable)
γ_{01} = Weight for W
υ_{0j} = Error term for intercept γ_{10} = Weight of the predictor
υ_{0j} = Error term for slope