# Week 9: Scaling, Contrasts, Interactions

## Univariate Statistics and Methodology using R

Department of Psychology
The University of Edinburgh

# Part 1

## Scaling

# Learning to Read

| age | hrs_wk | method | R_AGE |
|---|---|---|---|
| 10.115 | 4.971 | phonics | 14.272 |
| 9.940 | 4.677 | phonics | 13.692 |
| 6.060 | 4.619 | phonics | 10.353 |
| 9.269 | 4.894 | phonics | 12.744 |
| 10.991 | 5.035 | phonics | 15.353 |
| 6.535 | 5.272 | word | 5.798 |
| 8.150 | 6.871 | word | 8.691 |
| 7.941 | 4.053 | word | 6.988 |
| 8.233 | 5.474 | word | 8.713 |
| 6.219 | 4.038 | word | 5.908 |

# Learning to Read

```
## ...
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.423      2.472   -0.98    0.332
## age            0.938      0.206    4.55 0.000038 ***
## hrs_wk         0.964      0.418    2.31    0.025 *
## ...
```

# Learning to Read

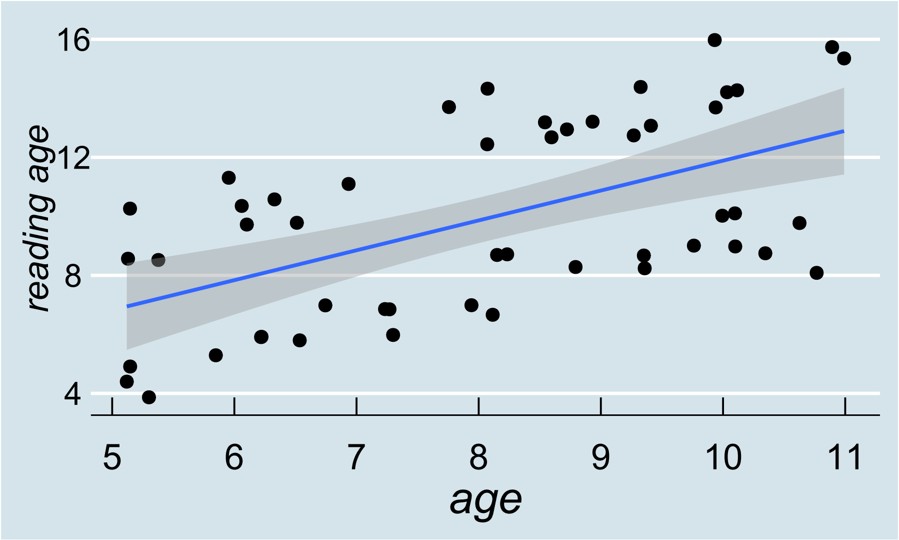
```
## ...
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.423      2.472   -0.98    0.332
## age            0.938      0.206    4.55 0.000038 ***
## hrs_wk         0.964      0.418    2.31    0.025 *
## ...
```

- as we noted last week, the *intercept* for this model is nonsensical
  - "children aged zero who read for zero hours a week have a predicted reading age of -2.423"
- perhaps there's something we can do about this?

# One-Predictor Model

- let's start with a model with a *single* predictor of age[1]

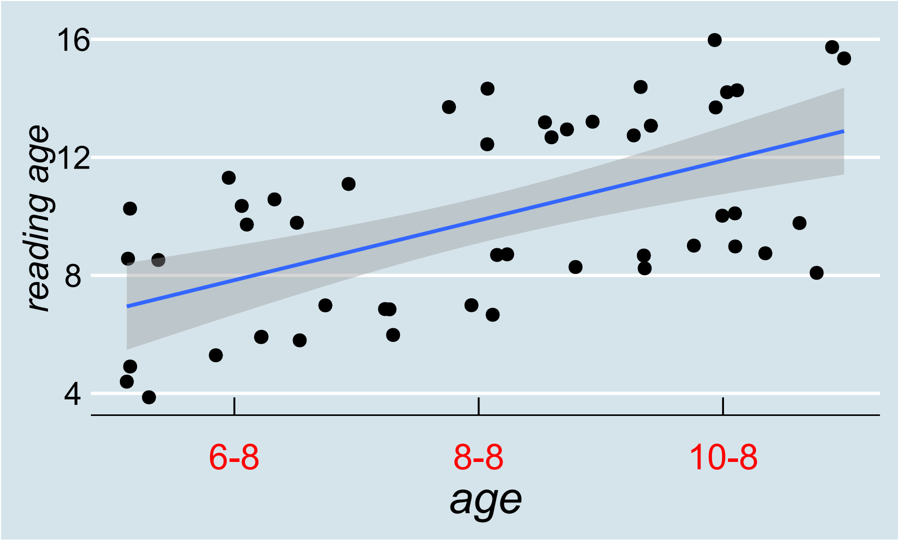
```
# model
mod2 <- lm(R_AGE ~ age,data=reading)

# figure
p <- reading %>% ggplot(aes(x=age,y=R_AGE)) +
  xlab("age") + ylab("reading age") +
  geom_point(size=3) +
  geom_smooth(method="lm")
p
```




[1] we know this model doesn't meet assumptions, but it will work for an illustration

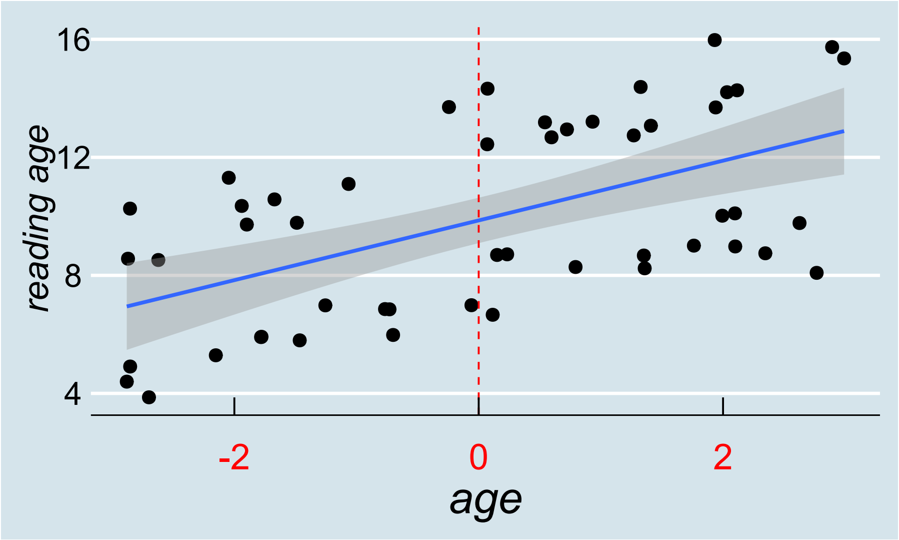
# Changing the Intercept

- actually it's fairly easy to move the intercept
- we can just pick a "useful-looking" value
- for example, we might want the intercept to tell us about students at age 8
  - this is a decision; no magic about it



# Changing the Intercept

- actually it's fairly easy to move the intercept
- we can just pick a "useful-looking" value
- for example, we might want the intercept to tell us about students at age 8
    - this is a decision; no magic about it



# A Model With a New Intercept

## original model

```
## ...
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.764      1.753    1.01     0.32
## age            1.012      0.212    4.76 0.000018 ***
## ...
```

## new model

```
mod2b <- lm(R_AGE ~ I(age-8), data=reading)
summary(mod2b)
```

```
## ...
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.862      0.383   25.77  < 2e-16 ***
## I(age - 8)     1.012      0.212    4.76 0.000018 ***
## ...
```

# Fit Remains Unchanged

## original model

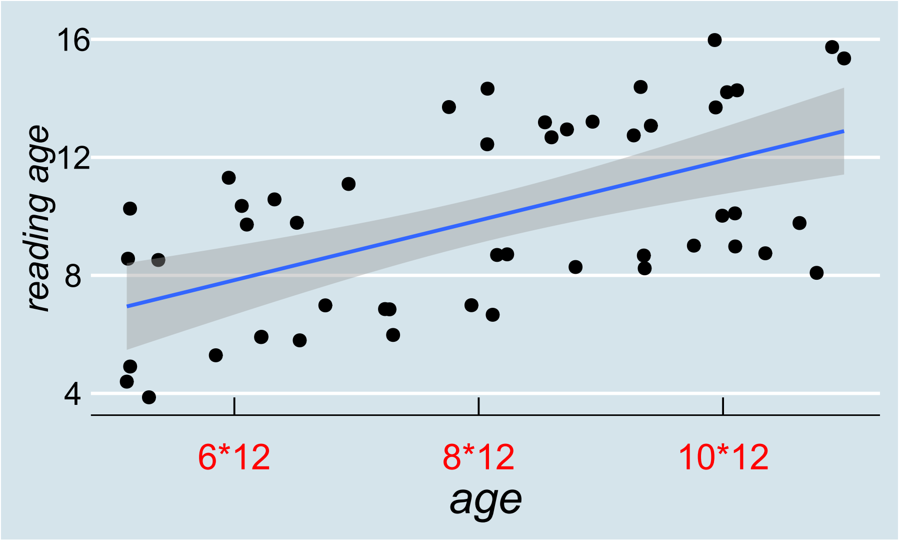
```
## ...
## Multiple R-squared:  0.321,    Adjusted R-squared:  0.307
## F-statistic: 22.7 on 1 and 48 DF,  p-value: 0.0000179
```

## new model

```
## ...
## Multiple R-squared:  0.321,    Adjusted R-squared:  0.307
## F-statistic: 22.7 on 1 and 48 DF,  p-value: 0.0000179
```

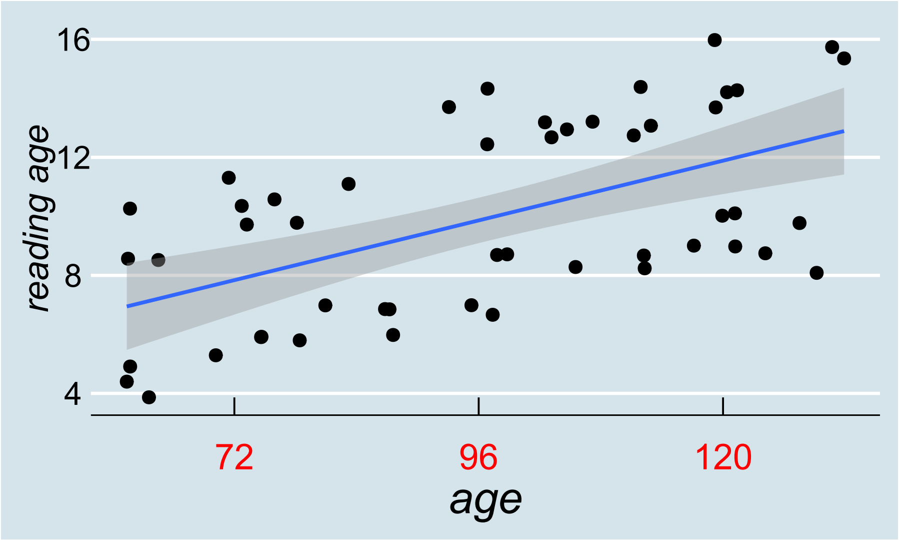
# A Model with a New Slope

- it's also easy to linearly scale the slope
- we can just pick a "useful" scale
- for example, we might want to examine the effect per month of age
    - this is a decision; no magic about it



# A Model with a New Slope

- it's also easy to linearly scale the slope
- we can just pick a "useful" scale
- for example, we might want to examine the effect per month of age
    - this is a decision; no magic about it



# A Model With a New Slope

## original model

```
## ...
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.764      1.753    1.01     0.32
## age            1.012      0.212    4.76 0.000018 ***
## ...
```

## new model

```
mod2c <- lm(R_AGE ~ I(age*12), data=reading)
summary(mod2c)
```

```
## ...
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7638     1.7534    1.01     0.32
## I(age * 12)   0.0844     0.0177    4.76 0.000018 ***
## ...
```

# Fit Remains Unchanged

## original model

```
## ...
## Multiple R-squared:  0.321,	Adjusted R-squared:  0.307
## F-statistic: 22.7 on 1 and 48 DF,  p-value: 0.0000179
```

## new model

```
## ...
## Multiple R-squared:  0.321,	Adjusted R-squared:  0.307
## F-statistic: 22.7 on 1 and 48 DF,  p-value: 0.0000179
```

# We Can Get Fancy About This

```
mod.mb <- lm(R_AGE ~ I((age-8)*12) + I(hrs_wk-mean(hrs_wk)), data=reading)
summary(mod.mb)
```

```
##
## Call:
## lm(formula = R_AGE ~ I((age - 8) * 12) + I(hrs_wk - mean(hrs_wk)),
##     data = reading)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.385 -2.251  0.326  2.395  3.201
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                9.8659     0.3665   26.92  < 2e-16 ***
## I((age - 8) * 12)          0.0782     0.0172    4.55 0.000038 ***
## I(hrs_wk - mean(hrs_wk))   0.9636     0.4176    2.31    0.025 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.59 on 47 degrees of freedom
## Multiple R-squared:  0.39,    Adjusted R-squared:  0.364
## F-statistic:   15 on 2 and 47 DF,  p-value: 0.00000896
```

# Often easier to scale *then* fit.

```
reading <- reading %>%
  mutate(
    agemonthC = (age - 8)*12,
    hrs_wkC = hrs_wk - mean(hrs_wk)
  )
mod.mb2 <- lm(R_AGE ~ agemonthC + hrs_wkC, data=reading)
summary(mod.mb2)
```

```
## Residuals:
##    Min     1Q Median     3Q    Max 
## -4.385 -2.251  0.326  2.395  3.201 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   9.8659     0.3665   26.92  < 2e-16 ***
## agemonthC     0.0782     0.0172    4.55 0.000038 ***
## hrs_wkC       0.9636     0.4176    2.31    0.025 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.59 on 47 degrees of freedom
## Multiple R-squared:  0.39,	Adjusted R-squared:  0.364 
## F-statistic:   15 on 2 and 47 DF,  p-value: 0.00000896
```

# Which Has a Bigger Effect?

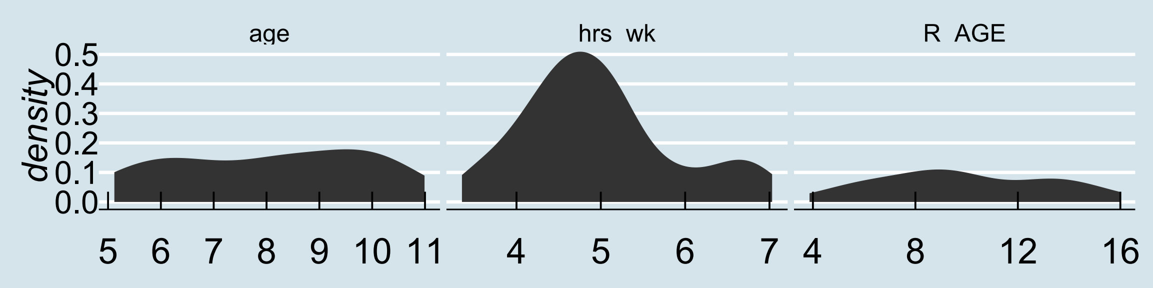
- in our two-predictor model, is age more important than practise? Or vice-versa?
- hard to tell because the predictors are in different *units*

```
## ...
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.423      2.472   -0.98    0.332
## age             0.938      0.206    4.55 0.000038 ***
## hrs_wk          0.964      0.418    2.31    0.025 *
## ...
```

- how do we compare effects of a year of age to those of an hour per week of practise?

# Standardisation

- *if* the predictors and outcome are very roughly normally distributed...




- we can calculate $z$-scores by subtracting the mean and dividing by the standard deviation

$$z_i = \frac{x_i - \bar{x}}{\sigma_x}$$

# Standardisation

- in R, the `scale()` function calculates $z$-scores
- in R, you don't need to create new columns!
  - also don't need to use `I()` because no ambiguity (though you can use it if you want)

```
mod.ms <- lm(scale(R_AGE) ~ scale(age) + scale(hrs_wk), data=reading)
```

# Standardisation

- in R, the `scale()` function calculates $z$-scores
- in R, you don't need to create new columns!
  - also don't need to use `I()` because no ambiguity (though you can use it if you want)

```
mod.ms <- lm(scale(R_AGE) ~ scale(age) + scale(hrs_wk), data=reading)
```

- the variables are now *all* in terms of standard deviations from the mean
- at the *intercept*, `age` is the mean of age and `hrs_wk` is the mean of hrs_wk
- *slopes*: "how many standard deviations does `R_AGE` change for a one standard deviation change in the predictor?"

# Standardisation

```r
summary(mod.ms)
```

```
## ...
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.41e-16   1.13e-01    0.00    1.000
## scale(age)      5.25e-01   1.15e-01    4.55 0.000038 ***
## scale(hrs_wk)   2.66e-01   1.15e-01    2.31    0.025 *
## ...
```

- `R_AGE` changes 0.52 sds for a 1-sd change in `age`, and 0.27 sds for a 1-sd change in `hrs_wk`
- reasonable conclusion might be that age has a greater effect on reading age than does practice

# Standardisation

```
summary(mod.ms)
```

```
## ...
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.41e-16   1.13e-01    0.00    1.000
## scale(age)     5.25e-01   1.15e-01    4.55 0.000038 ***
## scale(hrs_wk)  2.66e-01   1.15e-01    2.31    0.025 *
## ...
```

- `R_AGE` changes 0.52 sds for a 1-sd change in `age`, and 0.27 sds for a 1-sd change in `hrs_wk`
- reasonable conclusion might be that age has a greater effect on reading age than does practice
- model fit doesn't change with standardisation

```
## ...
## Multiple R-squared:  0.39,    Adjusted R-squared:  0.364
## F-statistic:   15 on 2 and 47 DF,  p-value: 0.00000896
```

# Standardisation pre-fit

- Using `scale()` inside the `lm()` is just the same as adjusting the variable prior to fitting the model

```
reading <- reading %>%
  mutate(
    zR_AGE = (R_AGE - mean(R_AGE)) / sd(R_AGE),
    zage = (age - mean(age))/sd(age),
    zhrs_wk = scale(hrs_wk)
  )
mod.ms2 <- lm(zR_AGE ~ zage + zhrs_wk, data=reading)
summary(mod.ms2)
```

```
## ...
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.41e-16   1.13e-01    0.00    1.000
## zage         5.25e-01   1.15e-01    4.55 0.000038 ***
## zhrs_wk      2.66e-01   1.15e-01    2.31    0.025 *
## ...
```

# Standardisation Post-Hoc

- we can convert "raw" model coefficients $b$ to standardised coefficients $\beta$ without re-running the regression]
- for predictor $x$ of outcome $y$:

$$\beta_x = b_x \cdot \frac{\sigma_x}{\sigma_y}$$

- or there are functions to do it for you

```r
library(lsr)
standardCoefs(mod.m)
```

```
##              b   beta
## age    0.9378 0.5250
## hrs_wk 0.9636 0.2662
```

```r
library(lm.beta)
summary(lm.beta(mod.m))
```

```
## Coefficients:
##             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)   -2.423           NA      2.472   -0.98    0.332
## age            0.938        0.525      0.206    4.55 0.000038 ***
## hrs_wk         0.964        0.266      0.418    2.31    0.025 *
```

End of Part 1

# Part 2

## Categorical Predictors

# Playmobil vs. SuperZings

- some important pretesting went into these lectures
- every individual figure rated for "usefulness" in explaining stats
- how do we decide which to use?

# Playmobil vs. SuperZings

| type | UTILITY |
|---|---|
| playmo | 8.2 |
| zing | 0.6 |
| playmo | 8.0 |
| zing | 1.0 |
| playmo | 7.7 |
| zing | 3.0 |
| playmo | 7.7 |
| zing | 5.4 |
| playmo | 3.6 |
| zing | 0.9 |

- some important pretesting went into these lectures
- every individual figure rated for "usefulness" in explaining stats
- how do we decide which to use?

# Playmobil vs. SuperZings

| type | UTILITY |
|---|---|
| playmo | 8.2 |
| zing | 0.6 |
| playmo | 8.0 |
| zing | 1.0 |
| playmo | 7.7 |
| zing | 3.0 |
| playmo | 7.7 |
| zing | 5.4 |
| playmo | 3.6 |
| zing | 0.9 |

- we already know one way to answer this

```
t.test(UTILITY~type, data=toys,
       var.equal=TRUE)
```

```
##
##      Two Sample t-test
##
## data:  UTILITY by type
## t = 3.9, df = 8, p-value = 0.005
## alternative hypothesis: true difference in means between group playmo and gro
## 95 percent confidence interval:
##  1.964 7.756
## sample estimates:
## mean in group playmo   mean in group zing
##                 7.04                 2.18
```

# The Only Equation You'll Ever Need

- which toys are the most useful?

$$\text{outcome}_i = (\text{model})_i + \text{error}_i$$

$$\text{utility}_i = (\text{some function of type})_i + \epsilon_i$$

- we need to represent `type` as a number
- the simplest way of doing this is to use 0 or 1

# Quantifying a Nominal Predictor

```
toys <- toys %>%
  mutate(is_playmo =
           ifelse(type=="playmo",1,0))
toys
```

```
## # A tibble: 10 × 3
##    type   UTILITY is_playmo
##    <fct>    <dbl>     <dbl>
##  1 playmo     8.2         1
##  2 zing       0.6         0
##  3 playmo     8           1
##  4 zing       1           0
##  5 playmo     7.7         1
##  6 zing       3           0
##  7 playmo     7.7         1
##  8 zing       5.4         0
##  9 playmo     3.6         1
## 10 zing       0.9         0
```

# Quantifying a Nominal Predictor

```
toys <- toys %>%
  mutate(is_playmo =
           ifelse(type=="playmo",1,0))
toys
```

```
## # A tibble: 10 × 3
##    type   UTILITY is_playmo
##    <fct>    <dbl>     <dbl>
##  1 playmo     8.2         1
##  2 zing       0.6         0
##  3 playmo     8           1
##  4 zing       1           0
##  5 playmo     7.7         1
##  6 zing       3           0
##  7 playmo     7.7         1
##  8 zing       5.4         0
##  9 playmo     3.6         1
## 10 zing       0.9         0
```

- this maps to a linear model

$$\text{utility}_i = b_0 + b_1 \cdot \text{is\_playmo} + \epsilon_i$$

- $\overline{\text{utility}}$ for SuperZings is intercept
- "change due to playmo-ness" is slope

# Linear Model Using is_playmo

```
mod1 <- lm(UTILITY~is_playmo,data=toys)
summary(mod1)
```

```
##
## Call:
## lm(formula = UTILITY ~ is_playmo, data = toys)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.440 -1.255  0.660  0.925  3.220
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.180      0.888    2.46   0.0396 *
## is_playmo      4.860      1.256    3.87   0.0047 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.99 on 8 degrees of freedom
## Multiple R-squared:  0.652,    Adjusted R-squared:  0.608
## F-statistic:   15 on 1 and 8 DF,  p-value: 0.00474
```

# Let R Do the Work

```
contrasts(toys$type)
```

```
##        zing
## playmo    0
## zing      1
```

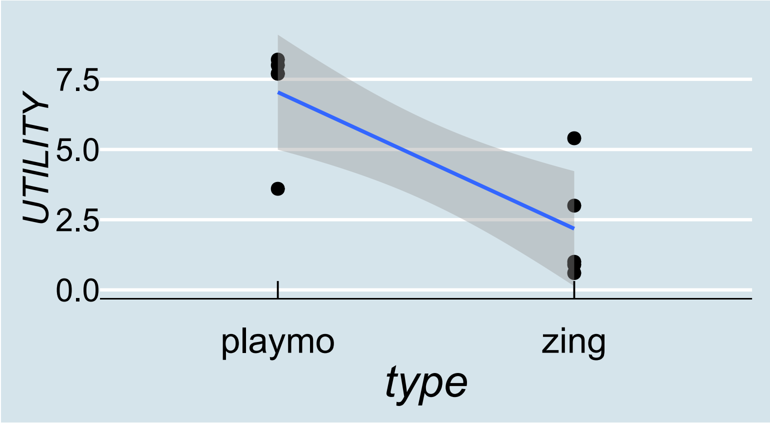
- already built-in to factors
- NB the first value will be the default intercept (because $b_n = 0$ for that value)
  - can change this using the `relevel()` function (or tidyverse `fct_relevel()`)
- as long as we have a *factor*, can just use lm() with that column

# Linear Model Using `type`

```
mod2 <- lm(UTILITY~type, data=toys)
summary(mod2)
```

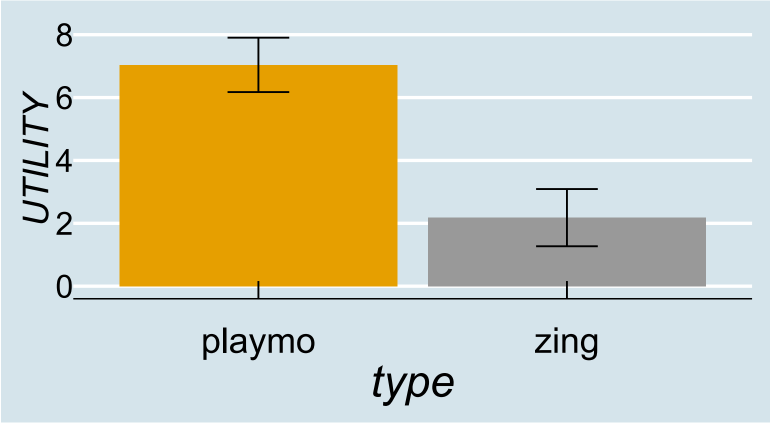
```
##
## Call:
## lm(formula = UTILITY ~ type, data = toys)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.440 -1.255  0.660  0.925  3.220
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.040      0.888    7.93 0.000047 ***
## typezing      -4.860      1.256   -3.87   0.0047 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.99 on 8 degrees of freedom
## Multiple R-squared:  0.652,    Adjusted R-squared:  0.608
## F-statistic:   15 on 1 and 8 DF,  p-value: 0.00474
```

# Graphically



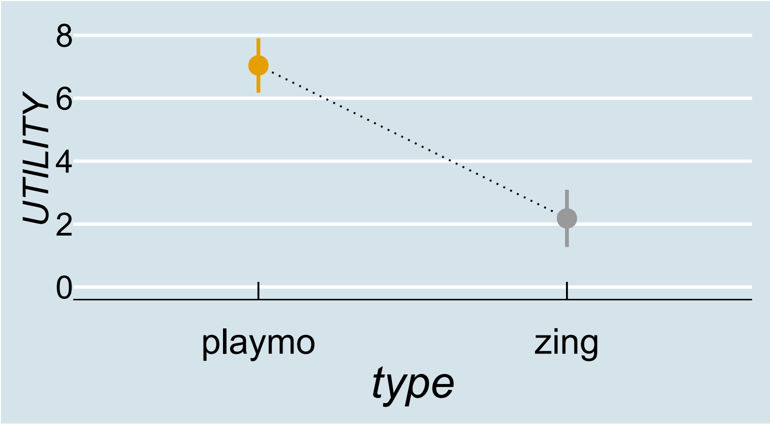

- shows "what the model is doing", but isn't a very good presentation
- the line suggests you can make predictions for types between *playmo* and *zing*

# Graphically




- error bars represent one standard error of the mean

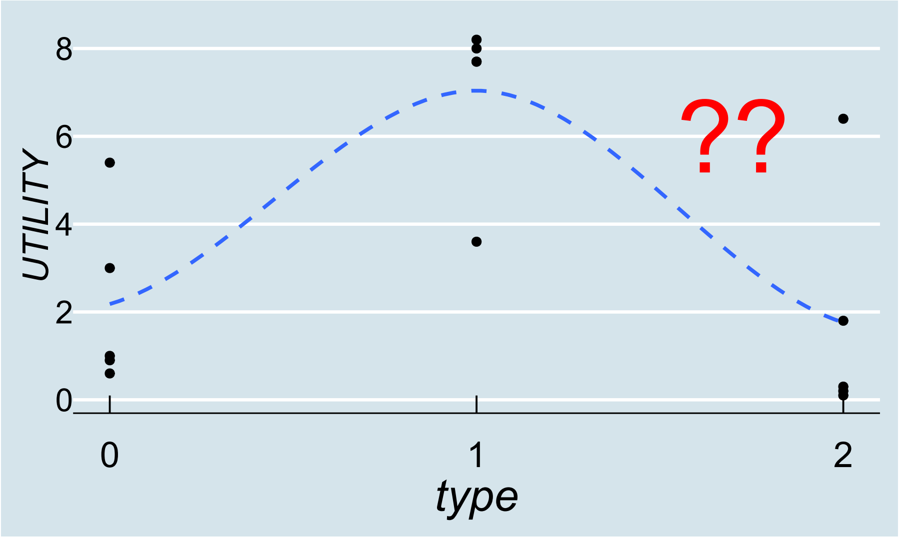
# Graphically




- error bars represent one standard error of the mean

# What About Lego Figures?

- we now have three groups
- can't label them `c(0, 1, 2)` because that would express a linear relationship



# Independent Effects

- "change due to lego-ness" is *independent* of change due to anything else
- solution: add another predictor

```
toys <- toys %>% mutate(
  is_playmo = ifelse(type=="playmo",1,0),
  is_lego   = ifelse(type=="lego",1,0)
)
head(toys)
```

```
## # A tibble: 6 × 4
##   type   UTILITY is_playmo is_lego
##   <fct>    <dbl>     <dbl>   <dbl>
## 1 zing       0.6         0       0
## 2 playmo     8.2         1       0
## 3 lego       0.3         0       1
## 4 zing       1           0       0
## 5 playmo     8           1       0
## 6 lego       1.8         0       1
```

# Three-Level Predictor: Two Coefficients

| type | UTILITY | is_playmo | is_lego |
|---|---|---|---|
| zing | 0.6 | 0 | 0 |
| playmo | 8.2 | 1 | 0 |
| lego | 0.3 | 0 | 1 |
| zing | 1.0 | 0 | 0 |
| playmo | 8.0 | 1 | 0 |
| lego | 1.8 | 0 | 1 |

$$\text{UTILITY}_i = b_0 + b_1 \cdot \text{is\_playmo}_i + b_2 \cdot \text{is\_lego}_i + \epsilon_i$$

$$\text{UTILITY}_i = b_0 + b_1 \cdot 0 + b_2 \cdot 0 + \epsilon_i$$

"utility of a zing"

# Three-Level Predictor: Two Coefficients

| type | UTILITY | is_playmo | is_lego |
|---|---|---|---|
| zing | 0.6 | 0 | 0 |
| playmo | 8.2 | 1 | 0 |
| lego | 0.3 | 0 | 1 |
| zing | 1.0 | 0 | 0 |
| playmo | 8.0 | 1 | 0 |
| lego | 1.8 | 0 | 1 |

$$\text{UTILITY}_i = b_0 + b_1 \cdot \text{is\_playmo}_i + b_2 \cdot \text{is\_lego}_i + \epsilon_i$$

$$\text{UTILITY}_i = b_0 + b_1 \cdot 1 + b_2 \cdot 0 + \epsilon_i$$

"change in utility from a zing due to being a playmo"

# Three-Level Predictor: Two Coefficients

| type | UTILITY | is_playmo | is_lego |
|---|---|---|---|
| zing | 0.6 | 0 | 0 |
| playmo | 8.2 | 1 | 0 |
| lego | 0.3 | 0 | 1 |
| zing | 1.0 | 0 | 0 |
| playmo | 8.0 | 1 | 0 |
| lego | 1.8 | 0 | 1 |

$$\text{UTILITY}_i = b_0 + b_1 \cdot \text{is\_playmo}_i + b_2 \cdot \text{is\_lego}_i + \epsilon_i$$

$$\text{UTILITY}_i = b_0 + b_1 \cdot 0 + b_2 \cdot 1 + \epsilon_i$$

"change in utility from a zing due to being a lego"

# R Has Our Backs

- this is the *default* contrast coding in R
- known as **treatment** (or **dummy**) contrasts

```
contrasts(toys$type)
```

```
##        playmo lego
## zing        0    0
## playmo      1    0
## lego        0    1
```

# R Has Our Backs

- this is the *default* contrast coding in R
- known as **treatment** (or **dummy**) contrasts

```
contrasts(toys$type)
```

```
##        playmo lego
## zing        0    0
## playmo      1    0
## lego        0    1
```

## a subtle difference

```
# core R: alphabetical
contrasts(factor(toys$type))
contrasts(as.factor(toys$type))
```

```
##        playmo zing
## lego        0    0
## playmo      1    0
## zing        0    1
```

```
# tidyverse: order of mention
contrasts(as_factor(toys$type))
```

```
##        playmo lego
## zing        0    0
## playmo      1    0
## lego        0    1
```

# A Linear Model

```
mod <- lm(UTILITY ~ type, data=toys)
summary(mod)
```

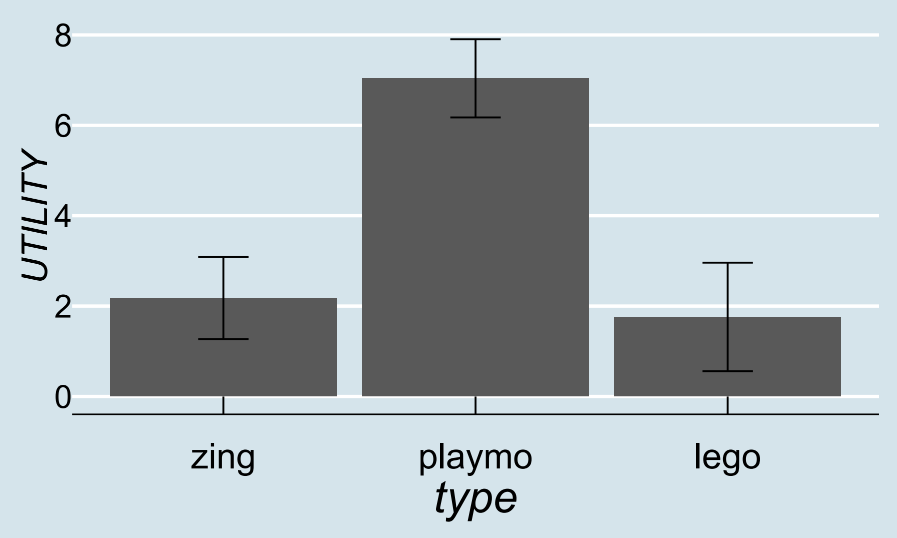
```
##
## Call:
## lm(formula = UTILITY ~ type, data = toys)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3.44  -1.51   0.04   0.89   4.64
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.18       1.00    2.17    0.051 .
## typeplaymo      4.86       1.42    3.43    0.005 **
## typelego       -0.42       1.42   -0.30    0.772
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.24 on 12 degrees of freedom
## Multiple R-squared:  0.588,    Adjusted R-squared:  0.519
## F-statistic: 8.56 on 2 and 12 DF,  p-value: 0.0049
```

# Group Means, Graphically

```
gd <- toys %>% group_by(type) %>%
  summarise(mean_se(UTILITY))
gd
```

```
## # A tibble: 3 × 4
##   type         y  ymin  ymax
##   <fct>    <dbl> <dbl> <dbl>
## 1 zing      2.18 1.27   3.09
## 2 playmo    7.04 6.17   7.91
## 3 lego      1.76 0.559  2.96
```

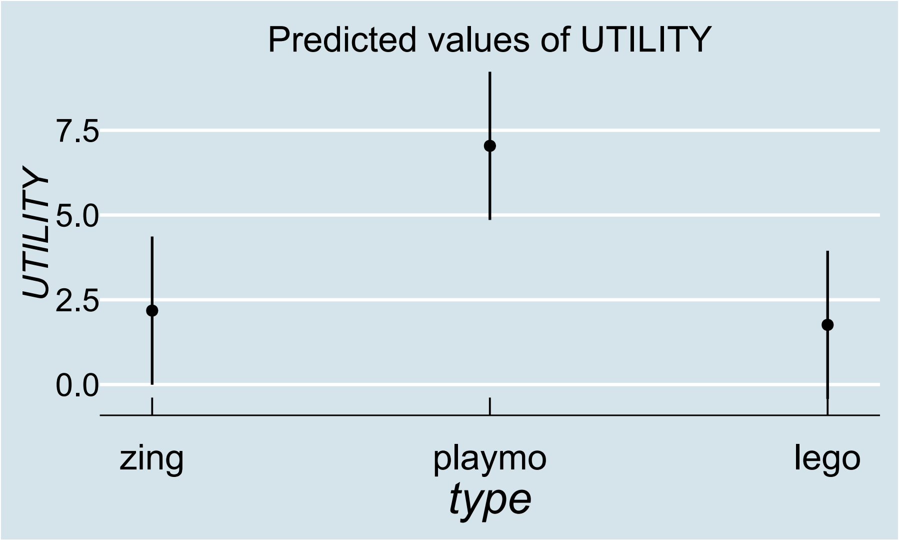
```
gd %>% ggplot(aes(x=type,y=y,
                  ymin=ymin,ymax=ymax)) +
  geom_bar(stat="identity") +
  geom_errorbar(width=.2) +
  ylab("UTILITY")
```



# Model estimates, Graphically

```
mod <- lm(UTILITY ~ type, data=toys)

library(sjPlot)
plot_model(mod, type = "eff")
```

```
## $type
```



# Contrast Coding

- there may be a different contrast coding which better suits our research interests
- for a predictor with $g$ levels (or "groups") there are $g - 1$ possible contrasts
- these can be anything you like (values don't have to be zero or one): there are a few built in to R
- usefulness depends on your research question
- contrasts act like "tests of differences of interest" once your model has been fit
- model fit is not affected by the choice of contrasts[1]

[1] for type 1 sums of squares

End of Part 2

# Part 3

## Interactions

# Back to Reading

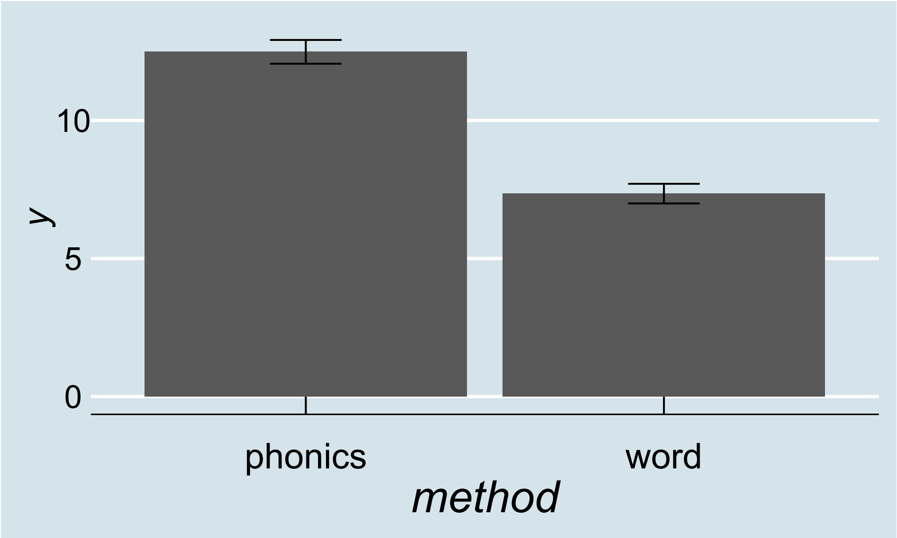
| age | hrs_wk | method | R_AGE |
|---|---|---|---|
| 10.115 | 4.971 | phonics | 14.272 |
| 9.940 | 4.677 | phonics | 13.692 |
| 6.060 | 4.619 | phonics | 10.353 |
| 9.269 | 4.894 | phonics | 12.744 |
| 10.991 | 5.035 | phonics | 15.353 |
| 6.535 | 5.272 | word | 5.798 |
| 8.150 | 6.871 | word | 8.691 |
| 7.941 | 4.053 | word | 6.988 |
| 8.233 | 5.474 | word | 8.713 |
| 6.219 | 4.038 | word | 5.908 |

# We Know Enough to Fit a Linear Model

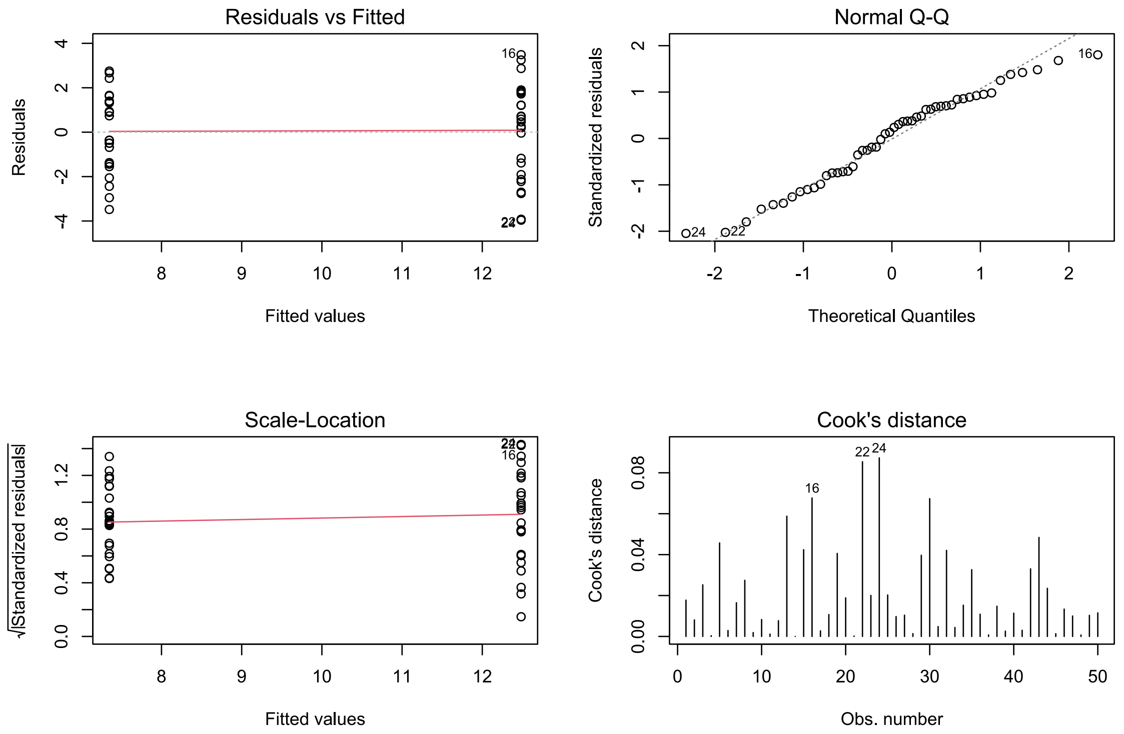
```
mod3 <- lm(R_AGE~method,data=reading)
summary(mod3)
```

```
##
## Call:
## lm(formula = R_AGE ~ method, data = reading)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3.96  -1.44   0.36   1.39   3.49
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   12.485      0.395   31.59  < 2e-16 ***
## methodword    -5.135      0.559   -9.19  3.8e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.98 on 48 degrees of freedom
## Multiple R-squared:  0.637,    Adjusted R-squared:  0.63
## F-statistic: 84.4 on 1 and 48 DF,  p-value: 3.76e-12
```

# We Know Enough to Draw a Graph




- we also know enough to run model diagnostics


Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Cook's distance
Cook's distance
Obs. number

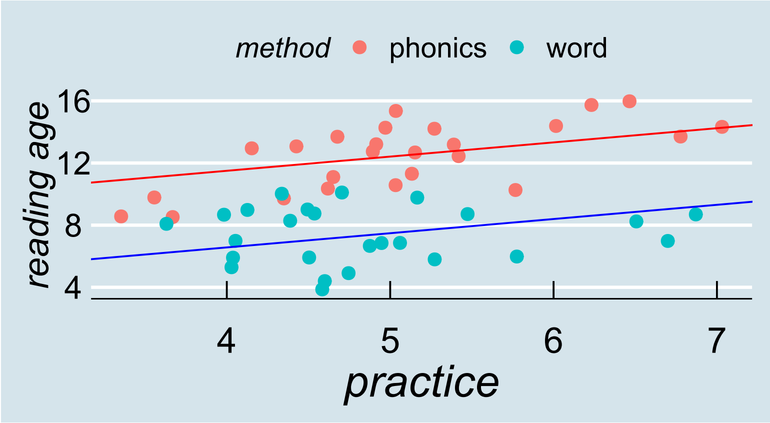
# Adding Predictors

- we also know that `hrs_wk` affects reading age
- we can build a multiple regression, and inspect the coefficients

```
mod.m2 <- lm(R_AGE ~ hrs_wk + method,data=reading)
summary(mod.m2)
```

```
## ...
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.843      1.524    5.15  5.1e-06 ***
## hrs_wk         0.914      0.291    3.14   0.0029 **
## methodword    -4.932      0.518   -9.53  1.5e-12 ***
## ...
```

# Graphically



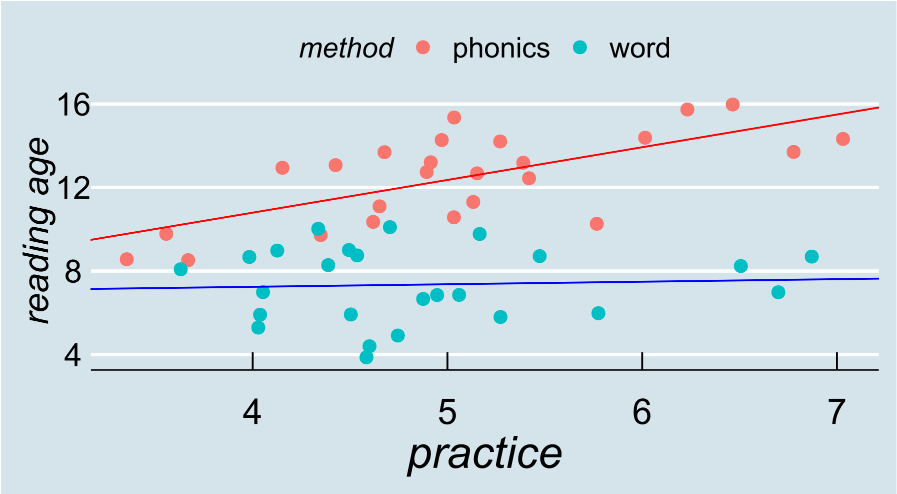

- note that according to this model the lines are parallel
- an hour of practice has *exactly the same* effect, however you're taught

# Different Effects for Different Methods

```
mod.m3 <- lm(R_AGE ~ hrs_wk + method + hrs_wk:method,data=reading)
summary(mod.m3)
```

```
##
## Call:
## lm(formula = R_AGE ~ hrs_wk + method + hrs_wk:method, data = reading)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.449 -1.377 -0.092  1.428  2.936
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.528      1.916    2.36  0.02238 *
## hrs_wk                1.567      0.371    4.22  0.00011 ***
## methodword            2.228      2.780    0.80  0.42697
## hrs_wk:methodword    -1.445      0.552   -2.62  0.01199 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.71 on 46 degrees of freedom
## Multiple R-squared:  0.739,    Adjusted R-squared:  0.722
## F-statistic: 43.4 on 3 and 46 DF,  p-value: 1.81e-13
```

# Different Effects for Different Methods



# Interaction is Just Multiplication

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{1i} x_{2i}$$

$$\widehat{\text{R\_AGE}} = b_0 + b_1 \cdot \text{hrs\_wk} + b_2 \cdot \text{word} + b_3 \cdot \text{hrs\_wk} \cdot \text{word}$$

- when $\text{word} = 0$:

$$\widehat{\text{R\_AGE}} = b_0 + b_1 \cdot \text{hrs\_wk} + b_2 \cdot 0 + b_3 \cdot \text{hrs\_wk} \cdot 0$$

- when $\text{word} = 1$:

$$\widehat{\text{R\_AGE}} = b_0 + b_1 \cdot \text{hrs\_wk} + b_2 \cdot 1 + b_3 \cdot \text{hrs\_wk} \cdot 1$$

# Interaction is Just Multiplication

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{1i} x_{2i}$$

$$\widehat{\text{R\_AGE}} = 4.53 + 1.57 \cdot \text{hrs\_wk} + 2.23 \cdot \text{word} + -1.44 \cdot \text{hrs\_wk} \cdot \text{word}$$

- when $\text{word} = 0$:

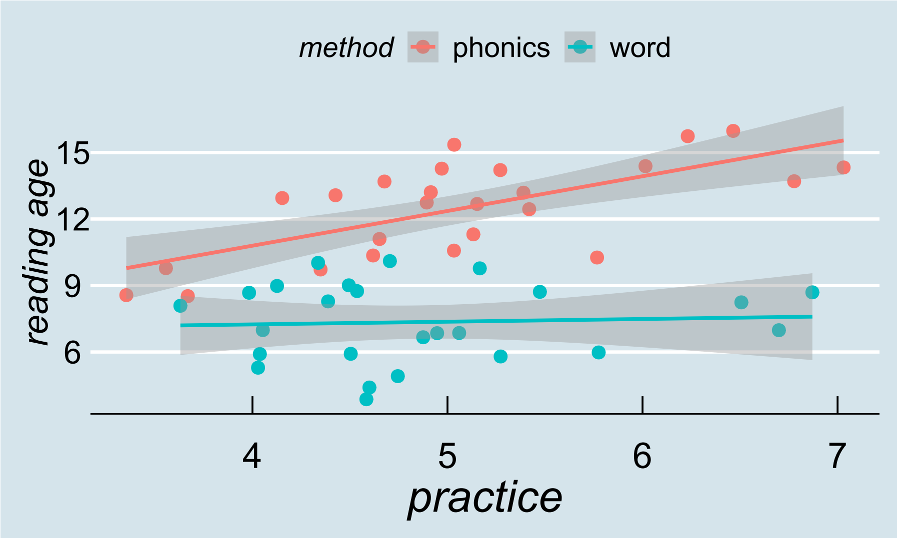
$$\widehat{\text{R\_AGE}} = 4.53 + 1.57 \cdot \text{hrs\_wk} + 2.23 \cdot 0 + -1.44 \cdot \text{hrs\_wk} \cdot 0$$

- when $\text{word} = 1$:

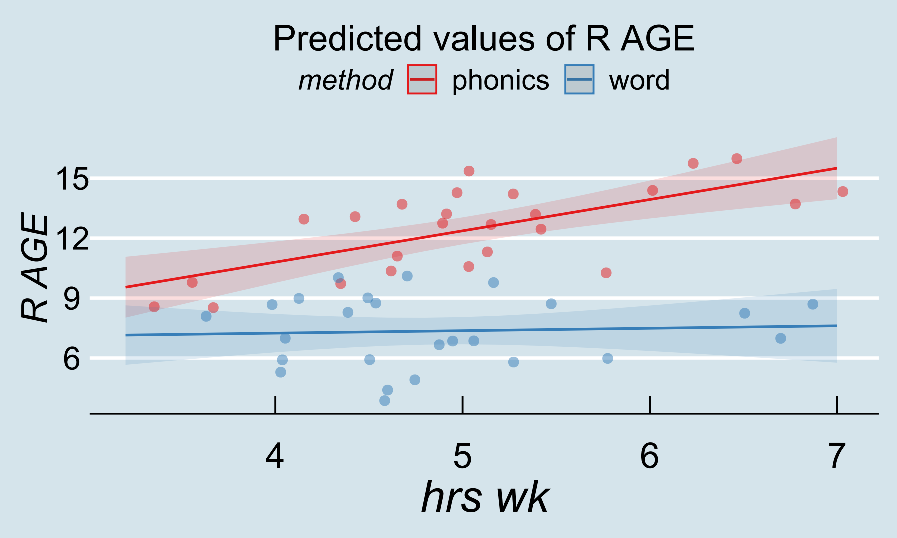
$$\widehat{\text{R\_AGE}} = 4.53 + 1.57 \cdot \text{hrs\_wk} + 2.23 \cdot 1 + -1.44 \cdot \text{hrs\_wk} \cdot 1$$

# Nice Graphs

```
reading %>% ggplot(
  aes(x=hrs_wk,y=R_AGE,colour=method)) +
  xlab("practice") +
  ylab("reading age") +
  geom_point(size=3) +
  geom_smooth(method="lm")
```




```
library(sjPlot)
plot_model(mod.m3, type = "int",
           show.data = TRUE)
```



# Interaction Really *Is* Just Multiplication

- in our dataset it's also possible that age and practice interact

"effect of practice is not constant across ages"

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{1i} x_{2i}$$

```
mod.m4 <- lm(R_AGE ~ age + hrs_wk + age:hrs_wk, data=reading)
```

# Interaction Really *Is* Just Multiplication

- in our dataset it's also possible that age and practice interact

"effect of practice is not constant across ages"

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{1i} x_{2i}$$

```
mod.m4 <- lm(R_AGE ~ age + hrs_wk + age:hrs_wk, data=reading)
```

`a + b + a:b` can also be written `a * b`

```
mod.m4 <- lm(R_AGE ~ age * hrs_wk, data=reading)
```

# Interaction of Age and Practice

```
summary(mod.m4)
```

```
##
## Call:
## lm(formula = R_AGE ~ age * hrs_wk, data = reading)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.445 -1.967 -0.336  2.302  3.925
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   16.596      9.714    1.71    0.094 .
## age           -1.449      1.198   -1.21    0.233
## hrs_wk        -3.079      2.041   -1.51    0.138
## age:hrs_wk     0.504      0.249    2.02    0.049 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.51 on 46 degrees of freedom
## Multiple R-squared:  0.44,    Adjusted R-squared:  0.403
## F-statistic:   12 on 3 and 46 DF,  p-value: 0.00000607
```

# Interaction Effect

```
## ...
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    16.596      9.714    1.71    0.094 .
## age            -1.449      1.198   -1.21    0.233
## hrs_wk         -3.079      2.041   -1.51    0.138
## age:hrs_wk      0.504      0.249    2.02    0.049 *
## ...
```

$$\widehat{\text{R\_AGE}}_i = b_0 + b_1 \cdot \text{age}_i + b_2 \cdot \text{hrs\_wk}_i + b_3 \cdot \text{age}_i \cdot \text{hrs\_wk}_i$$

**age 7; hrs_wk 5**

$$16.6 + -1.45 \cdot 7 + -3.08 \cdot 5 + 0.5 \cdot 7 \cdot 5$$

$$= 8.55$$

**age 12; hrs_wk 6**

$$16.6 + -1.45 \cdot 12 + -3.08 \cdot 6 + 0.5 \cdot 12 \cdot 6$$

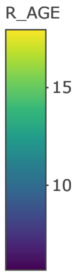
$$= 16.72$$

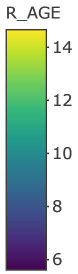
# Significance

```
## ...
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   16.596      9.714    1.71    0.094 .
## age           -1.449      1.198   -1.21    0.233
## hrs_wk        -3.079      2.041   -1.51    0.138
## age:hrs_wk     0.504      0.249    2.02    0.049 *
## ...
```

- note that not all of the effects are significant
- the model's best guess at the data ( $\widehat{\mathrm{R\_AGE}}$ ) is expressed by the coefficients
- but we're not confident that the highlighted effects would reliably differ from zero less than 5% of the time if we repeatedly sampled from the same population
- so the *predictions* of the model are as above (and below) but our *conclusion* is only that practise is more beneficial the older a child is

# Graphical Model



End

# Acknowledgements


- icons by Diego Lavecchia from the Noun Project