Lineární a logistická regrese

Mojmír Dočekal

2023-05-02

Contents

Chapter 1

Dvě témata

- lineární regrese
  - mimo-lingvistiku
  - lingvistické užití
- logistická regrese (mimolingvistický příklad)

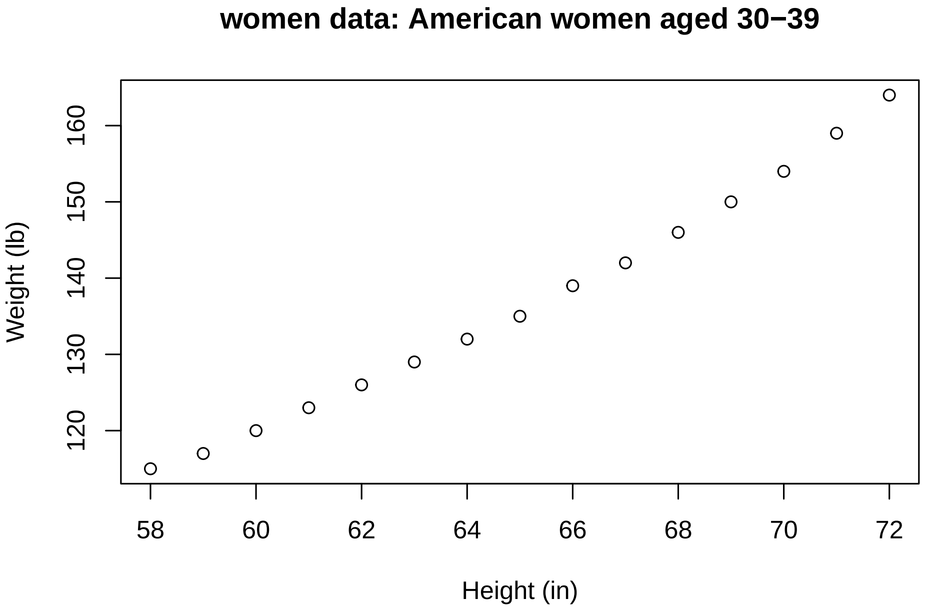
Chapter 2

Linearní regrese

2.1 Historie

- Laplace, Gauss, …
- metoda nejmenších čtverců
- Gaus a astronomie: Ceres

```r
require(graphics)

plot(women, xlab = "Height (in)", ylab = "Weight (lb)",
     main = "women data: American women aged 30-39")
```



```
data("women")

head(women)
```

```
##   height weight
## 1     58    115
## 2     59    117
## 3     60    120
## 4     61    123
## 5     62    126
## 6     63    129
```

```
help("women")
```

Average Heights and Weights for American Women Description This data set gives the average heights and weights for American women aged 30–39.

Usage women Format A data frame with 15 observations on 2 variables.

[,1] height numeric Height (in) [,2] weight numeric Weight (lbs) Details The data set appears to have been taken from the American Society of Actuaries Build and Blood Pressure Study for some (unknown to us) earlier year.

The World Almanac notes: “The figures represent weights in ordinary indoor clothing and shoes, and heights with shoes”.

Source The World Almanac and Book of Facts, 1975.

References McNeil, D. R. (1977) Interactive Data Analysis. Wiley.

Examples require(graphics) plot(women, xlab = "Height (in)", ylab = "Weight (lb)", main = "women data: American women aged 30-39") [Package datasets version 4.0.5 Index]

- a nyní model interpretující intuitivně jasný vztah mezi výškou a váhou
- výška je *explanatory* variable (vysvětlující proměnná)
- váha je *dependent* variable (závislá proměnná)
  - jiná terminologie: explanatory – independent variables
- model:

```
lm <- lm(women$weight ~ women$height)

lm
```

```
##
## Call:
## lm(formula = women$weight ~ women$height)
##
## Coefficients:
##  (Intercept)  women$height
##       -87.52          3.45
```

- y-intercept (-87.52)
  - žena s nulovou výškou by vážila -87.52 liber
- regression-coefficient (3.45)
  - numericky vyjádřený vztah mezi explanatory a dependent variable
  - nárustek závislé proměnné, vzroste-li explanatory proměnná a 1 jednotku

```
summary(lm)
```

```
##
## Call:
## lm(formula = women$weight ~ women$height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7333 -1.1333 -0.3833  0.7417  3.1167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -87.51667    5.93694  -14.74 1.71e-09 ***
## women$height   3.45000    0.09114   37.85 1.09e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.525 on 13 degrees of freedom
## Multiple R-squared:  0.991,  Adjusted R-squared:  0.9903
## F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

- každý koeficient má:

1. sign (znaménko): pozitivní nebo negativní

2. velikost (síla efektu)

3. signifikance: pravděpodobnost nulové hypotéze vůči danému vzorku

- při dostatečném množství pozorování se díky central limit theoeremu neoctneme příliš daleko (standard error) od skutečného mean celé populace

- R^2 měří úspěšnost modelu

  - v daném, velmi omezeném vzorku, vysvětluje model 99% – nerealistické v normálním vzorku

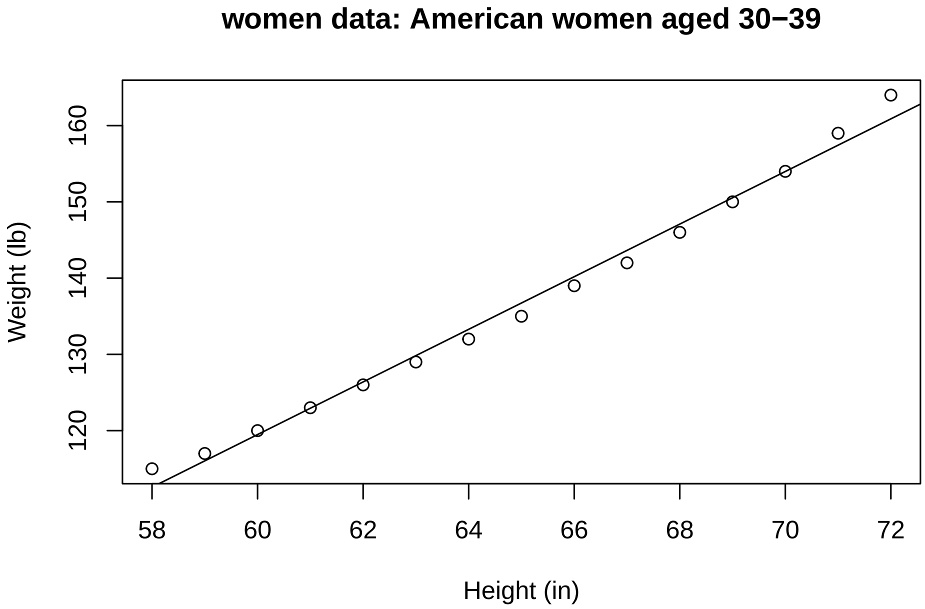
    - samozřejmě mnohem víc faktorů: věk, pohlaví, …

- koeficienty lineární regrese zároveň určují proložení přímky

- lineární regrese

- a také residuals pro každé pozorování

```
plot(women, xlab = "Height (in)", ylab = "Weight (lb)",
     main = "women data: American women aged 30-39")

abline(lm(women$weight ~ women$height))
```



2.3 Multiple regression analysis (vícenásobná lineární regrese)

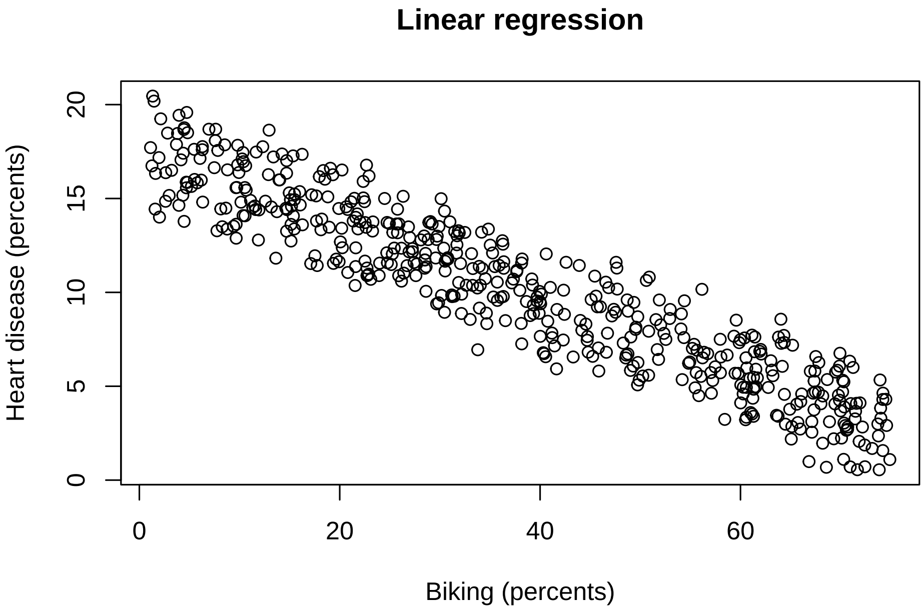
- malinko realističtější příklad
- data z link

```
setwd("heart_biking_smoking")

heart.data <- read.csv2("heart.data.csv", header = TRUE, sep = ",")

heart.data$X <- NULL

heart.data$biking <- as.numeric(heart.data$biking)
heart.data$smoking <- as.numeric(heart.data$smoking)
heart.data$heart.disease <- as.numeric(heart.data$heart.disease)

head(heart.data)
```

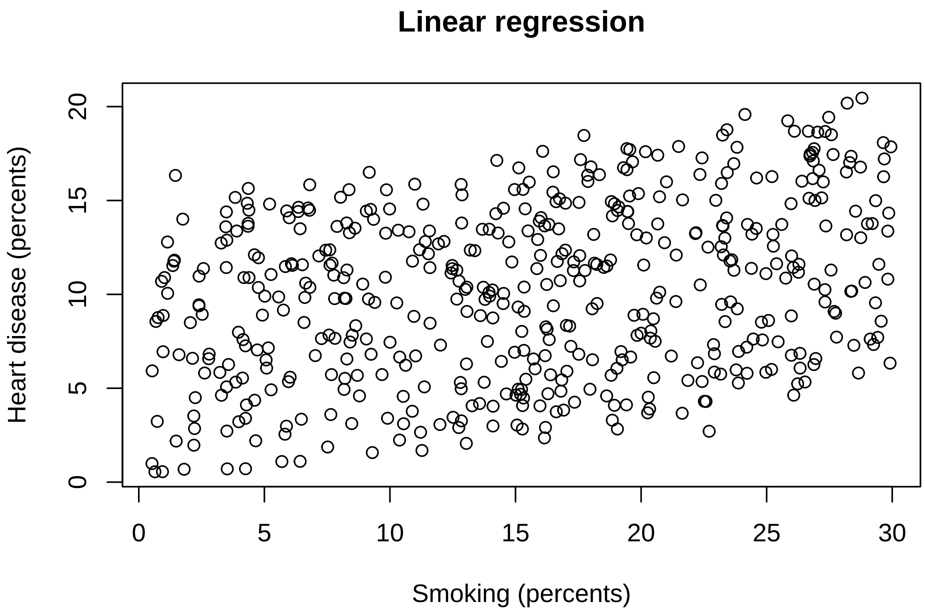
```
##      biking   smoking heart.disease
## 1 30.801246 10.896608     11.769423
## 2 65.129215  2.219563      2.854081
## 3  1.959665 17.588331     17.177803
## 4 44.800196  2.802559      6.816647
## 5 69.428454 15.974505      4.062224
## 6 54.403626 29.333176      9.550046
```

- statistika měst a procentuálního poměru cyklistů, kuřáká a osob se srdeční chorobou
- dvě explanatory proměnné
- jedna závislá proměnná
- vícenásobná lineární regrese
- korelační grafy:

```
plot(heart.data$heart.disease ~ heart.data$biking, xlab = "Biking (percents)", ylab = "
     main = "Linear regression")
```



```
plot(heart.data$heart.disease ~ heart.data$smoking, xlab = "Smoking (percents)", ylab =
     main = "Linear regression")
```



- model podobně konstruovaný, ale s dvěma explanatory proměnnými

```
heart.disease.lm<-lm(heart.disease ~ biking + smoking, data = heart.data)

heart.disease.lm
```

```
##
## Call:
## lm(formula = heart.disease ~ biking + smoking, data = heart.data)
##
## Coefficients:
## (Intercept)       biking      smoking
##     14.9847      -0.2001       0.1783
```

- podrobnější info:

```
summary(heart.disease.lm)
```

```
##
## Call:
## lm(formula = heart.disease ~ biking + smoking, data = heart.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1789 -0.4463  0.0362  0.4422  1.9331
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.984658   0.080137  186.99   <2e-16 ***
## biking      -0.200133   0.001366 -146.53   <2e-16 ***
## smoking      0.178334   0.003539   50.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.654 on 495 degrees of freedom
## Multiple R-squared:  0.9796, Adjusted R-squared:  0.9795
## F-statistic: 1.19e+04 on 2 and 495 DF,  p-value: < 2.2e-16
```

```
# library(ggplot2)

#abline(lm(heart.disease ~ biking, data = heart.data))
```

- jednodušší model:

```
heart.disease.lm2<-lm(heart.disease ~ biking, data = heart.data)

heart.disease.lm2
```

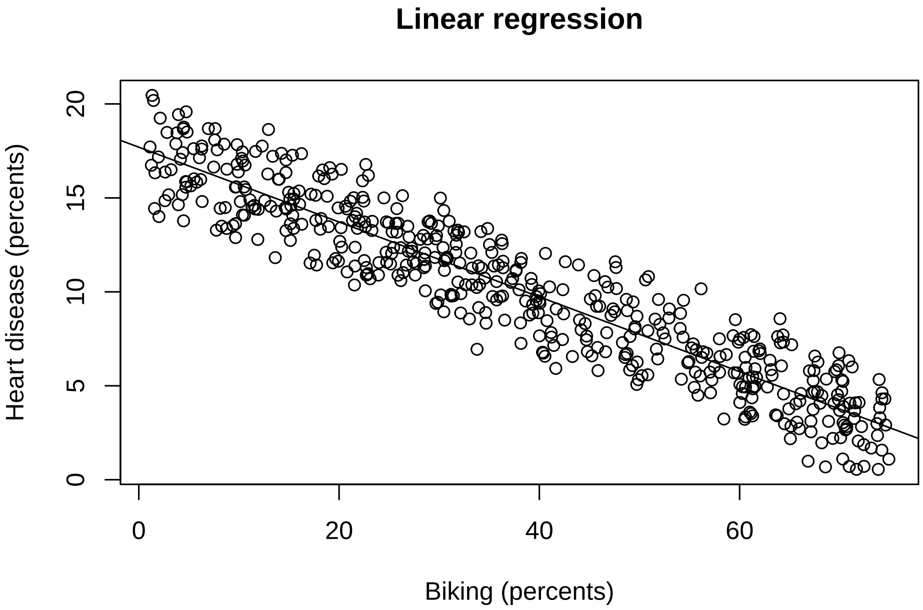
```
##
## Call:
## lm(formula = heart.disease ~ biking, data = heart.data)
##
## Coefficients:
## (Intercept)       biking
##     17.6979      -0.1991
```

```
summary(heart.disease.lm2)
```

```
##
## Call:
## lm(formula = heart.disease ~ biking, data = heart.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.028 -1.206 -0.004  1.151  3.643
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.697884   0.146780  120.57   <2e-16 ***
## biking      -0.199091   0.003378  -58.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.618 on 496 degrees of freedom
## Multiple R-squared:  0.8751, Adjusted R-squared:  0.8748
## F-statistic:  3474 on 1 and 496 DF,  p-value: < 2.2e-16
```

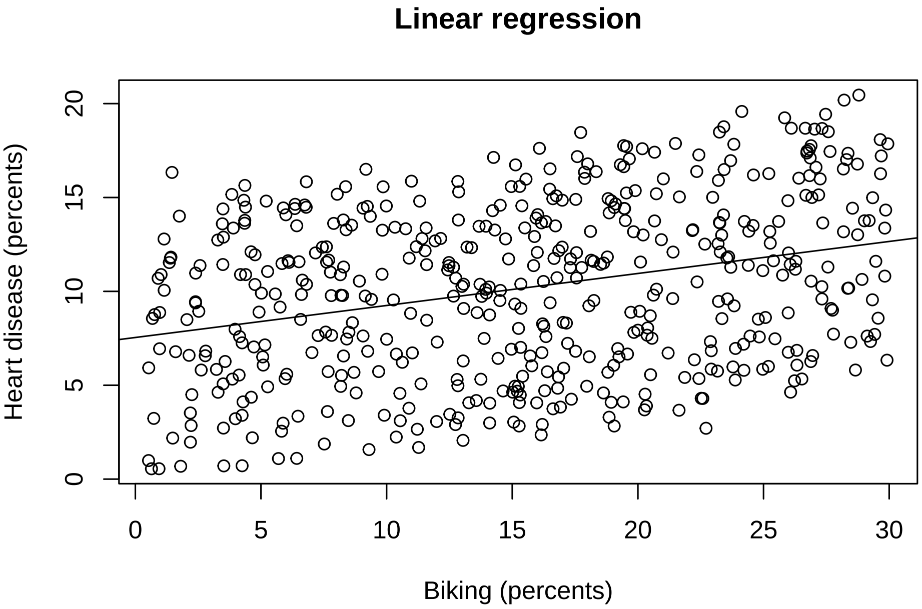
- horší R^2
- graf jednodušího modelu 3:

```
plot(heart.data$heart.disease ~ heart.data$biking, xlab = "Biking (percents)", ylab = "Heart dise
     main = "Linear regression")
abline(lm(heart.data$heart.disease ~ heart.data$biking))
```



- graf jednoduššího modelu 2:

```
plot(heart.data$heart.disease ~ heart.data$smoking, xlab = "Biking (percents)", ylab = "Heart dis
     main = "Linear regression")
abline(lm(heart.data$heart.disease ~ heart.data$smoking))
```



```
heart.disease.lm3<-lm(heart.disease ~ smoking, data = heart.data)

heart.disease.lm3
```

```
##
## Call:
## lm(formula = heart.disease ~ smoking, data = heart.data)
##
## Coefficients:
## (Intercept)      smoking
##      7.5431       0.1705
```

```
summary(heart.disease.lm3)
```

```
##
## Call:
## lm(formula = heart.disease ~ smoking, data = heart.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7065 -3.7069  0.5007  3.6597  8.5434
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.54311    0.41251  18.286  < 2e-16 ***
## smoking      0.17048    0.02355   7.239 1.73e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.352 on 496 degrees of freedom
## Multiple R-squared:  0.09556,    Adjusted R-squared:  0.09374
## F-statistic: 52.41 on 1 and 496 DF,  p-value: 1.729e-12
```

- srovnání modelů pomocí ANOVA

```
anova(heart.disease.lm2, heart.disease.lm)
```

```
## Analysis of Variance Table
##
## Model 1: heart.disease ~ biking
## Model 2: heart.disease ~ biking + smoking
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    496 1297.74
## 2    495  211.74  1      1086 2538.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- komplexnější (Df) model je mnohem lepší
- srovnání dvou jednodušších modelů

```
anova(heart.disease.lm3, heart.disease.lm2)
```

```
## Analysis of Variance Table
##
## Model 1: heart.disease ~ smoking
## Model 2: heart.disease ~ biking
##   Res.Df    RSS Df Sum of Sq F Pr(>F)
## 1    496 9395.6
## 2    496 1297.7  0    8097.8
```

- model užívající biking jako explanatory proměnnou je mnohem lepší
- zpět ke komplexnímu modelu

```
summary(heart.disease.lm)
```

```
##
## Call:
## lm(formula = heart.disease ~ biking + smoking, data = heart.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1789 -0.4463  0.0362  0.4422  1.9331
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.984658   0.080137  186.99   <2e-16 ***
## biking       -0.200133   0.001366 -146.53   <2e-16 ***
## smoking       0.178334   0.003539   50.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.654 on 495 degrees of freedom
## Multiple R-squared:  0.9796, Adjusted R-squared:  0.9795
## F-statistic: 1.19e+04 on 2 and 495 DF,  p-value: < 2.2e-16
```

```
# library(ggplot2)

#abline(lm(heart.disease ~ biking, data = heart.data))
```

- jednotlivé koeficienty jsou propočítány vzhledem k tomu, kdy jsou ostatní faktory fixovány
- např. jak moc zlepšuje cyklistika srdeční choroby, pokud je kouření fixováno na stejné hodnotě

2.4 Lingvistický příklad

- lineární model z Hančiny diplomky

2.4.1 Comparatively and superlatively modified numerals (theories, contrasts)

2.4.1.1 Background

- the accepted contrasts between comparative modifiers and superlative modifiers (see **?**, **?**,**?**, **?**, **?** a.o.):

1. comparative modifiers don't but superlative modifiers do give raise to obligatory ignorance implicatures (*?I have at most three bikes* vs. *I have less than four bikes*);

2. comparative modifiers can scope over or under existential modals (EM) but superlative modifiers have to outscope them (*The cup of Darjeeling tea can contain less than 50mg of caffeine*: $\Diamond > 50$ ok vs. *The cup of Darjeeling tea can contain at most 50mg of caffeine*: $*\Diamond > 50$)

- A *no more than Num* construction like in English (1a) from **?** is then claimed to:

1. to allow both scopes w.r.t. existential modals (**?**): (1a)) – $\Diamond > 20/20 > \Diamond$;

2. to have scalar bounding inference (50 for (1b));

- *no more than Num* construction is then claimed to be:

1. subtype of differential quantifiers in the class of comparative modifiers (as *slightly less* in (1c));

2. supports come from the comparative morphology of *no more than Num*;

2.4.1.2 Our claims

- we bring **new experimental evidence** (from Czech) against such claims, showing that (unlike in English):

1. *no more than Num* prefers to be interpreted with wider scope than existential modals;
2. *no more than Num* can behave unlike other differential comparative modifiers;
3. our experiment shows:

- there are two kinds of differential quantifiers: comparative ((1c) and English (1a)) and superlative (Czech *no more than Num* as in (2a)/(2c));

2.4.2 Experiment

- two experiments to target two research questions:

1. does Czech *no more than* behave more like a comparative modifier or a superlative modifier (in the modal environment)?
2. does Czech *no more* behave like other differential quantifiers?

2.4.2.1 Design of both experiments

- Czech native speakers
- Likert scale 1-5
- the appropriateness of one of the conditions in a context
- further, only exp 2: it included all the conditions of exp 1
- truth-value judgment task where a context described a situation strongly preferring the wide scope of the existential modal over the degree quantifiers
- 16 items and 16 fillers
- 98 subjects (all of them passed fillers uncontroversial TVJT)
- implemented on L-Rex

experiment followed an observation (**?**,**?** a.o.):

- comparative modifiers allow both wide and narrow scope w.r.t. an existential modal reading but superlative modifiers have to out-scope the existential modals (split-scope);
- four conditions:

1. comparative modifiers: FEWER, (2a);
2. superlative modifiers: AT-MOST, (2b);
3. *no more* modifier: NO-MORE, (2c);

4. differential *slightly less* modifier, (2d) – SLIGHTLY-LESS;

2.4.2.2 Assumptions

- the conditions FEWER and AT-MOST tested the acceptability of modified numerals without differential;
- the conditions SLIGHTLY-LESS, NO-MORE tested the presence of a differential (zero degree differential in the case of NO-MORE);
- all conditions tested possible narrow scope w.r.t. the existential modal;
- **?** predicts NO-MORE to be the comparative modifiers as SLIGHTLY-LESS;
- the design was 2x2 factorial: comparative modifiers or superlative modifiers (CLASSA,CLASSB) x absence/presence of a differential (DIFFYES,DIFFNO);
- an example item from the experiment is in (2).

2.4.2.3 Predictions

- in the existential modal environment manipulated in a way strongly prefering wide scope of the modal
- *N(o)M(ore than)C(onstruction)*
- *C(omparative)M(odifier)*
- *S(uperlative)M(odifier)*

		FEWER	AT-MOST	NO-MORE	SLIGHTLY-LESS
Predictions	NMC as CM	✓		✓	✓
	NMC as SM	✓			✓

2.4.2.4 Results

- the data was analyzed in a mixed-effects linear model with subject and item intercept+slope random effects (R package LMERTEST);
- the dependent variable: the subject's response;
- several models, the best: conditions as independent variables, plus their interaction;
- the less fitting models included models with main effects only and models where *no more* was treated as a comparative modifiers

1. we found a negative main effect of CLASSB (superlative modifiers) (t-value: -11.004, $p < 0.001$) and a positive effect of the absence of a differential (t-value: 3.946 $p < 0.001$);

2. the model also reports a negative interaction of CLASSB (superlative modifiers) by DIFFNO (t-value: -3.129, $p = 0.002$);

3. Tukey's pairwise comparison of the conditions reveals that only AT-MOST and NO-MORE were statistically non-significantly different (t-value: -0.478, $p = 0.964$);
    - all other pairs of conditions differed significantly;

- the boxplot representing means and SEs in Fig. 2.1
- the experiment thus confirms:

1. the scope behavior of Czech *no more* construction follows the pattern of superlative modifiers, not the comparative modifiers, since subjects accepted to the same extent NO-MORE as AT-LEAST;
2. the significant difference between NO-MORE and SLIGHTY-LESS which can be explained by classifying *no more* as an superlative modifiers differential quantifier and *slightly less* as a comparative modifiers;

The surprising result of this exp is the overall low acceptability of all conditions:

- even the most default comparative modifiers without a differential (cond FEWER) had μ=2.51 (SD: 1.61, SE: 0.04);
    - we hypothesize that this results from the priming effect of the most frequent everyday contexts, which strongly prefer the $max_d > \Diamond$ reading, just the opposite against the contexts described in our exp.

```
knitr::include_graphics("plot_zoom.png")
```

2.4.2.5 Analysis

- the scope behaviour of Czech NMC is of a superlative modifiers profile
- we follow original (**?**) suggestion to analyze German/Dutch *nicht mehr/niet meer* as a negative differential expressing that there is no positive difference in degree between the arguments of the comparative *more*:

$[[\text{nicht mehr }]]\alpha = \lambda P.\neg\exists d'[max_d(P(d)) = \alpha + d']$

- since the negative differential analysis is equivalent to the superlative modifiers at-issue semantics of *at most*:

$\lambda P.max_d(P(d)) \leq \alpha$ (after (**?**))

- applied to Czech experimental data correctly derives the similar scope behaviour of NMC and superlative modifiers;
- the wide scope of the NMC/superlative modifiers and then is:

$max_d(\Diamond\text{contain}(\text{ChocBar}, d)) \leq 65g$

- incompatible with Alex's continuation and predicts low acceptability of NO-MORE and AT-MOST in the experiment;

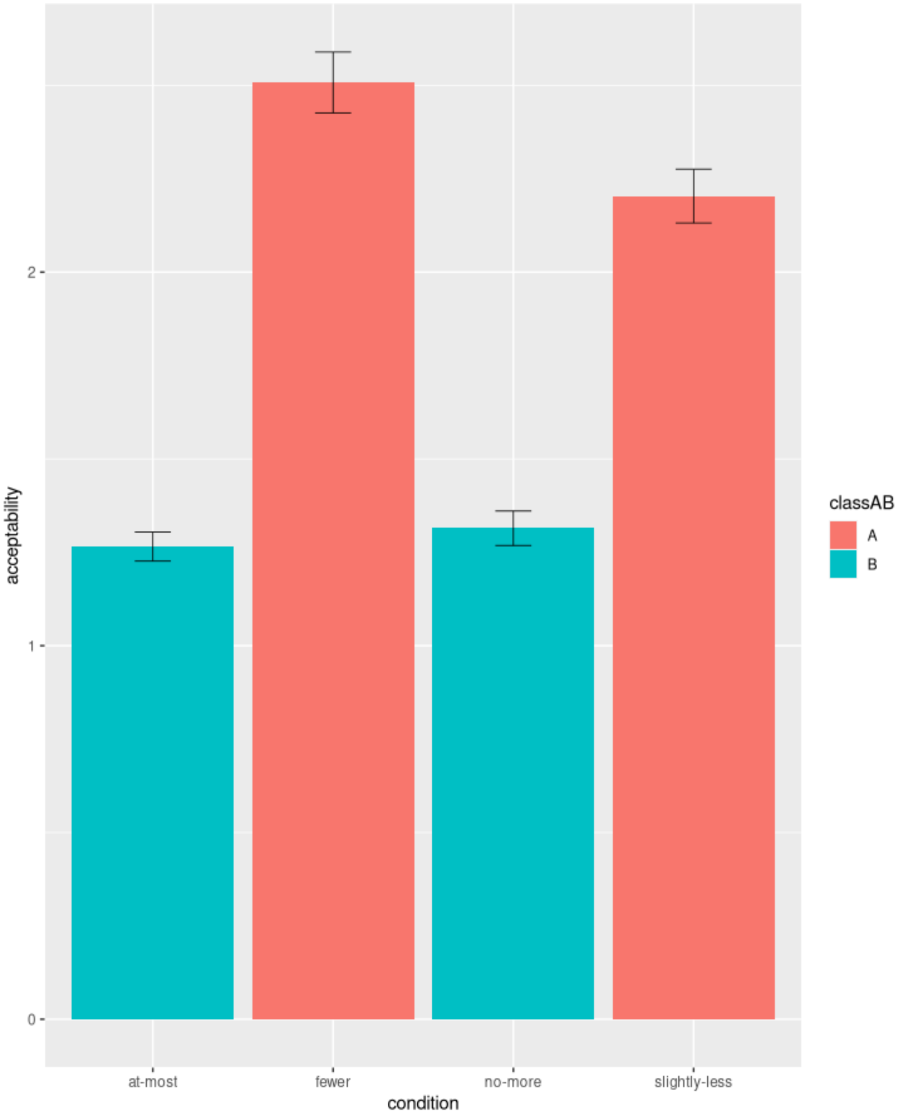


Figure 2.1: Boxplot of responses

- weak surface scope ($\Diamond[max_d(\text{contain}(\text{ChocBar}, d)) \leq 65g]$) which allows 'more than' reading is allowed only for comparative modifiers and explains the higher acceptability of FEWER and SLIGHTLY-LESS (whatever the reasons for obligatory wide scope of superlative modifiers over existential modals are, see (**?**));

- the scope behaviour of Czech NMC then shows that semantically NMC behaves as superlative modifiers, despite its comparative morphology;

- Secondly, the experiment brings support for the comparative modifiers vs. superlative modifiers theory presented by (**?**) where the distinction boils down to the type of ordering relation (strict vs. non-strict)

- NMC can, at least in languages like Czech, be interpreted as ¬ (strict) resulting in ordering entailments of non-strict ordering. Regular differential quantifiers (SLIGHTLY-LESS) remain strictly ordered, thus comparative modifiers;

- finally, cross-linguistically we found three types of NMC-languages:

1. NMC as comparative modifiers, English type of NMC (bounding inferences and both scopes w.r.t. existential modals);
2. NMC as superlative modifiers, Czech type of NMC (only $max_d > \Diamond$, lack of bounding inferences: (**?**));
3. languages where NMC depending on its realization behaves as comparative modifiers or as superlative modifiers (Hungarian according to Balázs Surányi (p.c.));

- the variation is related to the morpho-syntactic status, a constituent negation in NMC (Czech) behaves as superlative modifiers; a negative quantifier (English) in NMC leads to comparative modifiers;
- our experiment clearly shows that treating uniformly all NMC as comparative modifiers is cross-linguistically untenable and the distinction between comparative modifiers and superlative modifiers isn't purely morphological:
- Czech NMC contains both comparative marker and comparative standard marker but unlike regular comparative modifiers differentials, Czech NMC acts as a superlative numeral modifier.

2.4.3 Reports

```r
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
```

```
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#setwd("breakout_rooms_report")
items <- read.csv2("clean_items.csv", encoding = 'UTF-8', header = TRUE)

items <- items %>%
     mutate(condition=replace(condition, condition == "item-méně_než", "fewer")) %>%
     mutate(condition=replace(condition, condition == "item-nanejvýš", "at-most")) %>%
     mutate(condition=replace(condition, condition == "item-ne_víc_než", "no-more")) %
     mutate(condition=replace(condition, condition == "item-trochu_méně", "slightly-les
     as.data.frame()

ddply(items, .(condition), summarise, Means = mean(rating1, na.rm=TRUE))
```

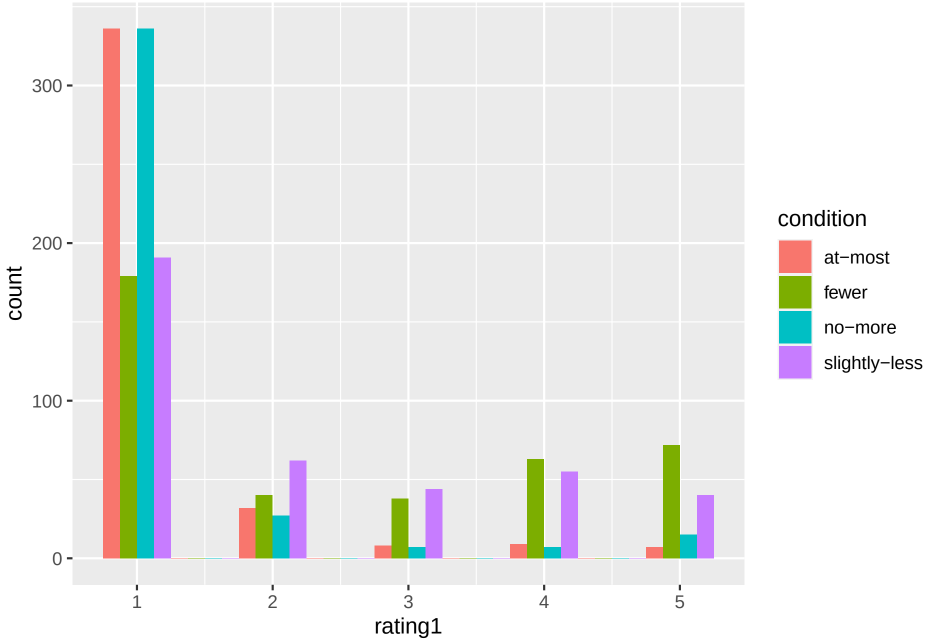
```
##       condition    Means
## 1       at-most 1.262755
## 2         fewer 2.512755
## 3       no-more 1.311224
## 4 slightly-less 2.211735
```

```
ddply(items, .(condition), summarise, Medians = median(rating1,na.rm=TRUE))
```

```
##       condition Medians
## 1       at-most       1
## 2         fewer       2
## 3       no-more       1
## 4 slightly-less       2
```

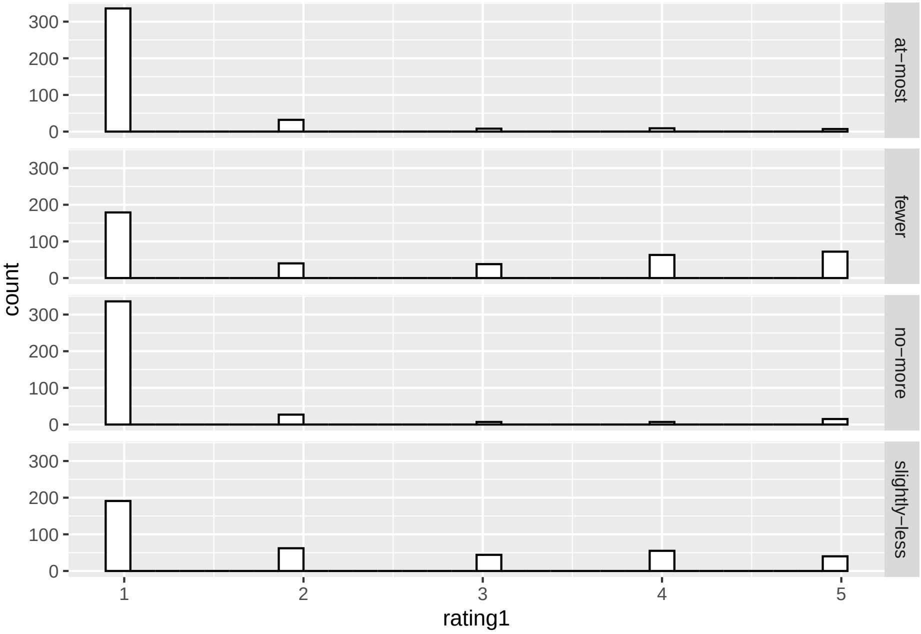
- graphs

```
library(ggplot2)

ggplot(items, aes(x = rating1, fill = condition)) +
geom_histogram(position = "dodge", alpha = 1, binwidth = 0.5)
```



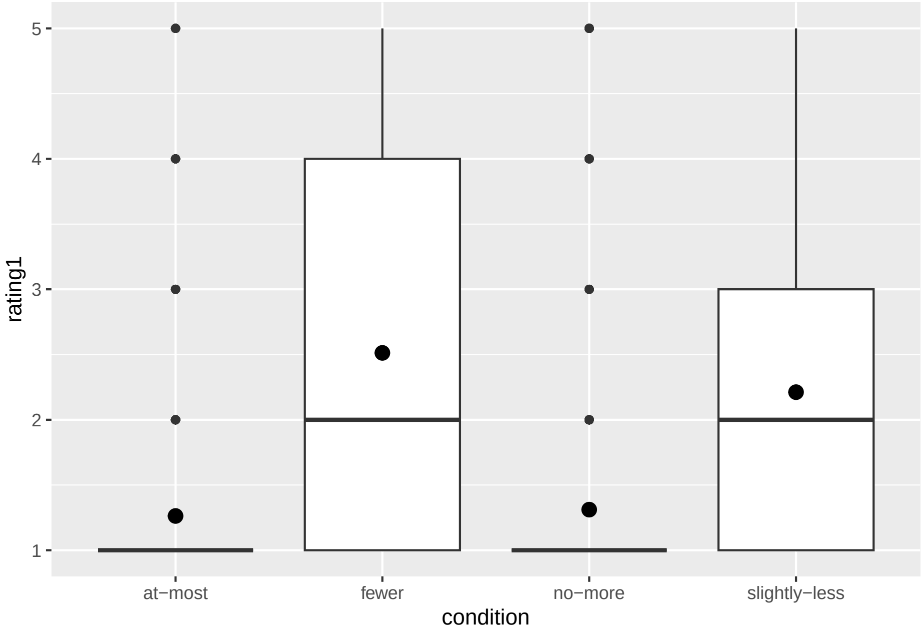
```r
ggplot(items, aes(x = rating1)) +
geom_histogram(fill = "white", colour = "black") +
facet_grid(condition ~ .)
```

```
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
```



```
data.to.plot <- items
graph_to_plot <- ggplot(data.to.plot, aes(condition,rating1))
graph_to_plot + geom_boxplot() + stat_summary(fun.y=mean, geom="point", size=3)
```

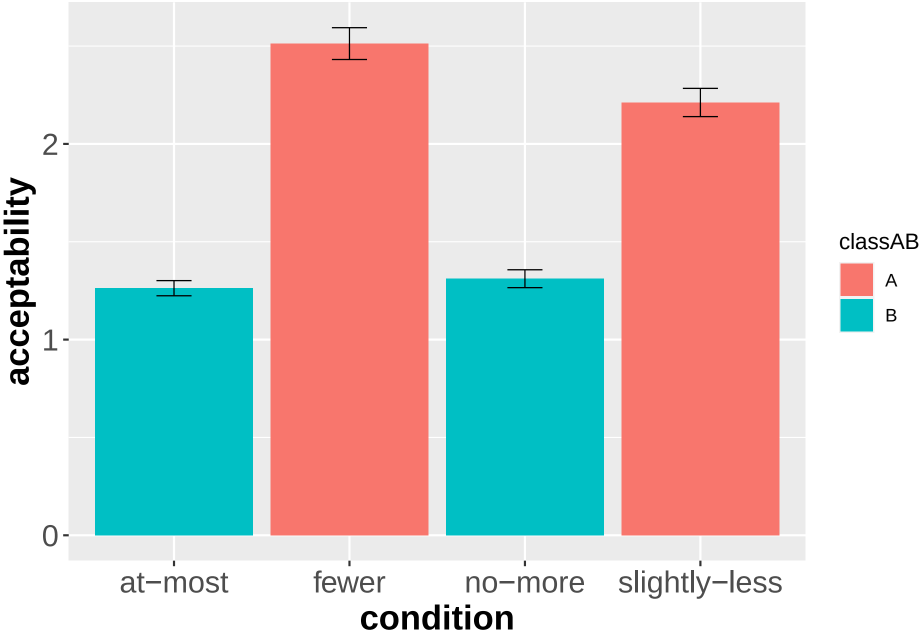
```
## Warning: The `fun.y` argument of
## `stat_summary()` is deprecated as of
## ggplot2 3.3.0.
## i Please use the `fun` argument
##   instead.
## This warning is displayed once every
## 8 hours.
## Call
## `lifecycle::last_lifecycle_warnings()`
## to see where this warning was
## generated.
```



```r
items$classAB <- "A"
items$classAB[items$condition == "at-most"] <- "B"
items$classAB[items$condition == "no-more"] <- "B"

p <- ggplot(items, aes(condition, rating1, fill = classAB)) +
stat_summary(geom = "bar", fun.y = mean, position = "dodge") +
stat_summary(geom = "errorbar", fun.data = mean_se, size=.3,
width=.2,
position=position_dodge(.9))
```

```
## Warning: Using `size` aesthetic for lines was
## deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every
## 8 hours.
## Call
## `lifecycle::last_lifecycle_warnings()`
## to see where this warning was
## generated.
```

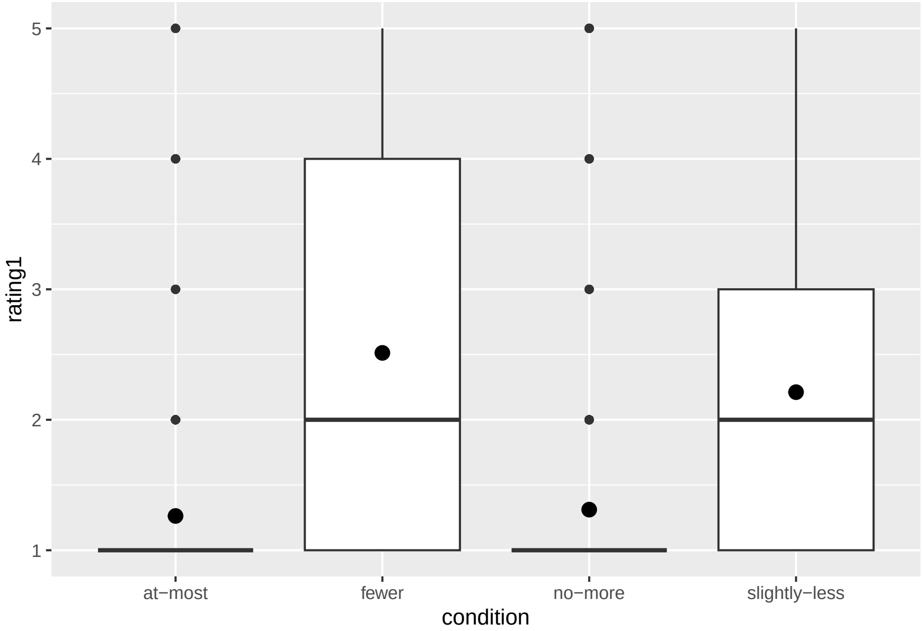
```r
p + labs(y = "acceptability") +
theme(axis.text=element_text(size=15),
axis.title=element_text(size=17,face="bold"))
```



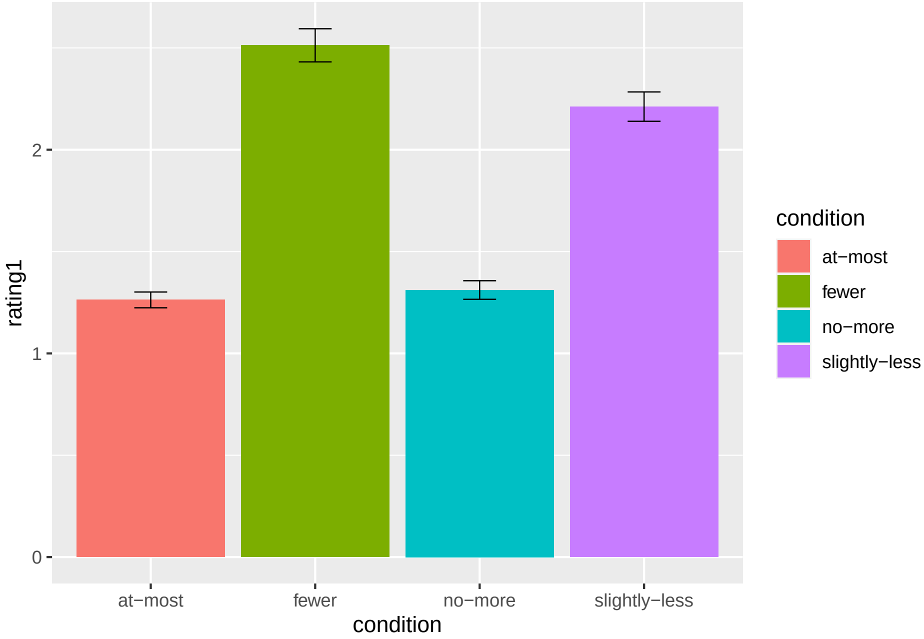
2.4.4 Modely

- modely

```r
library(ggplot2)
data.to.plot <- items

graph_to_plot <- ggplot(data.to.plot, aes(condition,rating1))
graph_to_plot + geom_boxplot() + stat_summary(fun.y=mean, geom="point", size=3)
```



```
ggplot(items, aes(condition, rating1, fill = condition)) +
  stat_summary(geom = "bar", fun.y = mean, position = "dodge") +
  stat_summary(geom = "errorbar", fun.data = mean_se, size=.3,
               width=.2,
               position=position_dodge(.9))
```



```
library(dplyr)


N    = length(items$rating1)
N
```

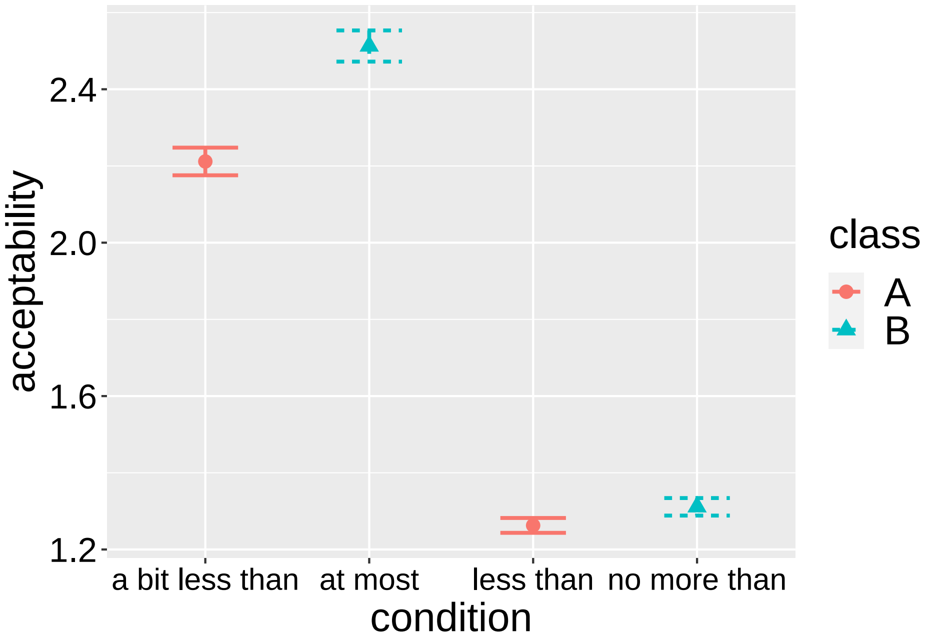
```
## [1] 1568
```

```
items.summary <- items %>%
  group_by(condition) %>%
  summarise(
    sd = sd(rating1),
    se   = sd / sqrt(N),
    acceptability = mean(rating1)
  )
items.summary
```

```
## # A tibble: 4 x 4
##   condition        sd     se acceptability
##   <chr>         <dbl>  <dbl>         <dbl>
## 1 at-most       0.767 0.0194          1.26
## 2 fewer         1.61  0.0407          2.51
## 3 no-more       0.905 0.0228          1.31
## 4 slightly-less 1.43  0.0361          2.21
```

```
items.summary[1,1] <- "less than"
items.summary[2,1] <- "at most"
items.summary[3,1] <- "no more than"
items.summary[4,1] <- "a bit less than"

items.summary$class <- NA
items.summary$class[1] <- "A"
items.summary$class[2] <- "B"
items.summary$class[3] <- "B"
items.summary$class[4] <- "A"


dodge <- position_dodge(width=0.3)
qplot(condition, acceptability, colour=class, pch=class, lty=class, data=items.summary
```

```
## Warning: `qplot()` was deprecated in ggplot2
## 3.4.0.
## This warning is displayed once every
## 8 hours.
## Call
## `lifecycle::last_lifecycle_warnings()`
## to see where this warning was
## generated.
```



- models

```
# linear model

library(lmerTest)
```

```
## Loading required package: lme4

## Loading required package: Matrix

##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step
```

```
items$condition <- as.factor(items$condition)

items$condition <- relevel(items$condition, ref="at-most")

m1 <- lmer(as.numeric(rating1) ~ condition  + (1|participant) + (1|item), data=items)
```

```
summary(m1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: as.numeric(rating1) ~ condition + (1 | participant) + (1 | item)
##    Data: items
##
## REML criterion at convergence: 4936.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1683 -0.6481 -0.2050  0.4471  3.6096
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 0.1414   0.3761
##  item        (Intercept) 0.1375   0.3708
##  Residual                1.2402   1.1137
## Number of obs: 1568, groups:  participant, 98; item, 16
##
## Fixed effects:
##                          Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)            1.271e+00  1.149e-01 2.942e+01  11.063  5.3e-12 ***
## conditionfewer         1.242e+00  7.959e-02 1.452e+03  15.601  < 2e-16 ***
## conditionno-more       3.996e-02  7.959e-02 1.452e+03   0.502    0.616
## conditionslightly-less 9.322e-01  7.961e-02 1.452e+03  11.710  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cndtnf cndtnn-
## conditinfwr -0.346
## conditnn-mr -0.346  0.500
## cndtnslght- -0.346  0.500  0.500
```

```
library(emmeans)

emmeans(m1, list(pairwise ~ condition), adjust = "tukey")
```

```
## Cannot use mode = "kenward-roger" because *pbkrtest* package is not installed
```

```
## $`emmeans of condition`
##  condition     emmean    SE   df lower.CL upper.CL
##  at-most         1.27 0.115 29.4     1.04     1.51
##  fewer           2.51 0.115 29.4     2.28     2.75
##  no-more         1.31 0.115 29.4     1.08     1.55
##  slightly-less   2.20 0.115 29.4     1.97     2.44
##
## Degrees-of-freedom method: satterthwaite
## Results are given on the as.numeric (not the response) scale.
## Confidence level used: 0.95
##
## $`pairwise differences of condition`
##  1                           estimate     SE   df t.ratio p.value
##  (at-most) - fewer             -1.242 0.0796 1452 -15.601  <.0001
##  (at-most) - (no-more)         -0.040 0.0796 1452  -0.502  0.9586
##  (at-most) - (slightly-less)   -0.932 0.0796 1452 -11.710  <.0001
##  fewer - (no-more)              1.202 0.0796 1452  15.095  <.0001
##  fewer - (slightly-less)        0.309 0.0796 1452   3.888  0.0006
##  (no-more) - (slightly-less)   -0.892 0.0796 1452 -11.211  <.0001
##
## Note: contrasts are still on the as.numeric scale
## Degrees-of-freedom method: satterthwaite
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```r
# m2: class A vs. class B traditionally
# going from morphology: adding column with class A/class B status
# condition "at-most" the only class B

items$classAB <- "A"
items$classAB[items$condition == "at-most"] <- "B"
items$classAB <- as.factor(items$classAB)

m2 <- lmer(as.numeric(rating1) ~ classAB + (1|participant) + (1|item), data=items)

summary(m2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: as.numeric(rating1) ~ classAB + (1 | participant) + (1 | item)
##    Data: items
##
## REML criterion at convergence: 5157.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8526 -0.6835 -0.3082  0.4417  3.3690
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 0.1285   0.3584
##  item        (Intercept) 0.1366   0.3696
##  Residual                1.4480   1.2033
## Number of obs: 1568, groups:  participant, 98; item, 16
##
## Fixed effects:
##                Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    2.00910    0.10526   20.39488   19.09 1.76e-14 ***
## classABB      -0.73794    0.07022 1454.35268  -10.51  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## classABB -0.167
```

```
anova(m1,m2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: items
## Models:
## m2: as.numeric(rating1) ~ classAB + (1 | participant) + (1 | item)
## m1: as.numeric(rating1) ~ condition + (1 | participant) + (1 | item)
##    npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## m2    5 5161.3 5188.1 -2575.7   5151.3
## m1    7 4937.8 4975.3 -2461.9   4923.8 227.5  2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# m1 fares better

# adding m3 with "no more" reclassified as class B

items$classAB[items$condition == "no-more"] <- "B"

m3 <- lmer(as.numeric(rating1) ~ classAB + (1|participant) + (1|item), data=items)

summary(m3)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: as.numeric(rating1) ~ classAB + (1 | participant) + (1 | item)
##    Data: items
##
## REML criterion at convergence: 4945.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0147 -0.6657 -0.2035  0.4376  3.5794
```

```
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 0.1407   0.3751
##  item        (Intercept) 0.1367   0.3698
##  Residual                1.2517   1.1188
## Number of obs: 1568, groups:  participant, 98; item, 16
##
## Fixed effects:
##                 Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)     2.35808    0.10761   22.83181   21.91   <2e-16 ***
## classABB       -1.06693    0.05655 1454.41148  -18.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## classABB -0.263
```

```
anova(m1,m2,m3)
```

```
## refitting model(s) with ML (instead of REML)

## Data: items
## Models:
## m2: as.numeric(rating1) ~ classAB + (1 | participant) + (1 | item)
## m3: as.numeric(rating1) ~ classAB + (1 | participant) + (1 | item)
## m1: as.numeric(rating1) ~ condition + (1 | participant) + (1 | item)
##    npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
## m2    5 5161.3 5188.1 -2575.7   5151.3
## m3    5 4949.1 4975.9 -2469.6   4939.1 212.180  0
## m1    7 4937.8 4975.3 -2461.9   4923.8  15.316  2  0.0004723 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# m1 still the best

# m4: interactions?
# class A vs. class B as the main effect
# presence of the differential phrase ("ne" or "trochu") as an interaction effect

items$Diff <- "NoDiff"
items$Diff[items$condition == "no-more"|items$condition == "slightly-less"] <- "Diff"
items$Diff <- as.factor(items$Diff)

m4 <- lmer(as.numeric(rating1) ~ classAB * Diff + (1|participant) + (1|item), data=items)
```

```
summary(m4)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: as.numeric(rating1) ~ classAB * Diff + (1 | participant) + (1 |
##     item)
##    Data: items
##
## REML criterion at convergence: 4936.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1683 -0.6481 -0.2050  0.4471  3.6096
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 0.1414   0.3761
##  item        (Intercept) 0.1375   0.3708
##  Residual                1.2402   1.1137
## Number of obs: 1568, groups:  participant, 98; item, 16
##
## Fixed effects:
##                       Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)             2.20337    0.11491   29.42257  19.175  < 2e-16 ***
## classABB               -0.89223    0.07959 1452.25414 -11.211  < 2e-16 ***
## DiffNoDiff              0.30942    0.07959 1452.25414   3.888 0.000106 ***
## classABB:DiffNoDiff   -0.34938    0.11252 1452.10172  -3.105 0.001939 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) clsABB DffNDf
## classABB    -0.346
## DiffNoDiff  -0.346  0.500
## clssABB:DND  0.245 -0.707 -0.707
```

```
# m5: "no more" as classA

anova(m1,m2,m3,m4)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: items
## Models:
## m2: as.numeric(rating1) ~ classAB + (1 | participant) + (1 | item)
## m3: as.numeric(rating1) ~ classAB + (1 | participant) + (1 | item)
## m1: as.numeric(rating1) ~ condition + (1 | participant) + (1 | item)
## m4: as.numeric(rating1) ~ classAB * Diff + (1 | participant) + (1 | item)
##    npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
## m2    5 5161.3 5188.1 -2575.7   5151.3
## m3    5 4949.1 4975.9 -2469.6   4939.1 212.180  0
## m1    7 4937.8 4975.3 -2461.9   4923.8  15.316  2  0.0004723 ***
## m4    7 4937.8 4975.3 -2461.9   4923.8   0.000  0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
items$classAB[items$condition == "no-more"] <- "A"

m5 <- lmer(as.numeric(rating1) ~ classAB * Diff + (1|participant) + (1|item), data=items)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```
summary(m5)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: as.numeric(rating1) ~ classAB * Diff + (1 | participant) + (1 |
##     item)
##    Data: items
##
## REML criterion at convergence: 5054.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0655 -0.6653 -0.2473  0.4431  3.4743
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 0.1348   0.3672
##  item        (Intercept) 0.1387   0.3724
##  Residual                1.3465   1.1604
## Number of obs: 1568, groups:  participant, 98; item, 16
##
## Fixed effects:
##                 Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)      1.75728    0.10845   22.67242   16.20 5.88e-14 ***
## classABB        -1.24116    0.08292 1453.27185  -14.97  < 2e-16 ***
## DiffNoDiff       0.75525    0.07183 1453.38009   10.52  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) clsABB
## classABB    0.000
## DiffNoDiff -0.221 -0.577
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```
summary(m4)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: as.numeric(rating1) ~ classAB * Diff + (1 | participant) + (1 |
##     item)
##    Data: items
##
## REML criterion at convergence: 4936.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1683 -0.6481 -0.2050  0.4471  3.6096
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 0.1414   0.3761
##  item        (Intercept) 0.1375   0.3708
##  Residual                1.2402   1.1137
## Number of obs: 1568, groups:  participant, 98; item, 16
##
## Fixed effects:
##                        Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)             2.20337    0.11491   29.42257  19.175  < 2e-16 ***
## classABB               -0.89223    0.07959 1452.25414 -11.211  < 2e-16 ***
## DiffNoDiff              0.30942    0.07959 1452.25414   3.888 0.000106 ***
## classABB:DiffNoDiff    -0.34938    0.11252 1452.10172  -3.105 0.001939 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) clsABB DffNDf
## classABB    -0.346
## DiffNoDiff  -0.346  0.500
## clssABB:DND  0.245 -0.707 -0.707
```

```
anova(m4,m5)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: items
## Models:
## m5: as.numeric(rating1) ~ classAB * Diff + (1 | participant) + (1 | item)
## m4: as.numeric(rating1) ~ classAB * Diff + (1 | participant) + (1 | item)
##    npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m5    6 5056.6 5088.8 -2522.3   5044.6
## m4    7 4937.8 4975.3 -2461.9   4923.8 120.81  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# m1 wins but m4 reports intersting interaction
```

Chapter 3

Logistická regrese

- podle The Art of Statistics GitHub repo

3.1 Klasická lineární regrese

Data from 1991-1995 are contained in 05-1-galton-x.csv, Although the book says the data is from HistData: Data Sets from the History of Statistics and Data Visualization, 2018, I have actually used this version of Galton's Height Data

```
galton<-read.csv("05-1-galton-x.csv",header=TRUE) # read csv file into dataframe galton
attach(galton) #uncomment if/while necessary

summary(galton)
```

```
##     Family              Father          Mother          Gender         
##  Length:898         Min.   :62.00   Min.   :58.00   Length:898        
##  Class :character   1st Qu.:68.00   1st Qu.:63.00   Class :character  
##  Mode  :character   Median :69.00   Median :64.00   Mode  :character  
##                     Mean   :69.23   Mean   :64.08                     
##                     3rd Qu.:71.00   3rd Qu.:65.50                     
##                     Max.   :78.50   Max.   :70.50                     
##      Height           Kids       
##  Min.   :56.00   Min.   : 1.000  
##  1st Qu.:64.00   1st Qu.: 4.000  
##  Median :66.50   Median : 6.000  
##  Mean   :66.76   Mean   : 6.136  
##  3rd Qu.:69.70   3rd Qu.: 8.000  
##  Max.   :79.00   Max.   :15.000  
```

```
# summary statistics
# need means for unique fathers and mothers - identify first mention of each family
Unique.Fathers=numeric()
Unique.Mothers=numeric()
nunique=1 # number of unique families
Unique.Fathers[1] = Father[1]
Unique.Mothers[1] = Mother[1]
 for(i in 2:length(Family))
{
    if(Family[i] != Family[i-1]){
      nunique=nunique+1
    Unique.Fathers[nunique]=Father[i]
    Unique.Mothers[nunique]=Mother[i]
    }
  }

length(Unique.Fathers)
```

```
## [1] 197
```

```
summary(Unique.Fathers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   62.00   68.00   69.50   69.35   71.00   78.50
```

```
sd(Unique.Fathers)
```

```
## [1] 2.622034
```

```
length(Unique.Mothers)
```

```
## [1] 197
```

```
summary(Unique.Mothers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   58.00   62.70   64.00   63.98   65.50   70.50
```

```
sd(Unique.Mothers)
```

```
## [1] 2.355607
```

```
Son = Height[Gender=="M"]
length(Son)
```

```
## [1] 465
```

```
summary(Son)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.00   67.50   69.20   69.23   71.00   79.00
```

```r
sd(Son)
```

```
## [1] 2.631594
```

```r
Daughter = Height[Gender=="F"]
length(Daughter)
```

```
## [1] 433
```

```r
summary(Daughter)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   56.00   62.50   64.00   64.11   65.50   70.50
```

```r
sd(Daughter)
```

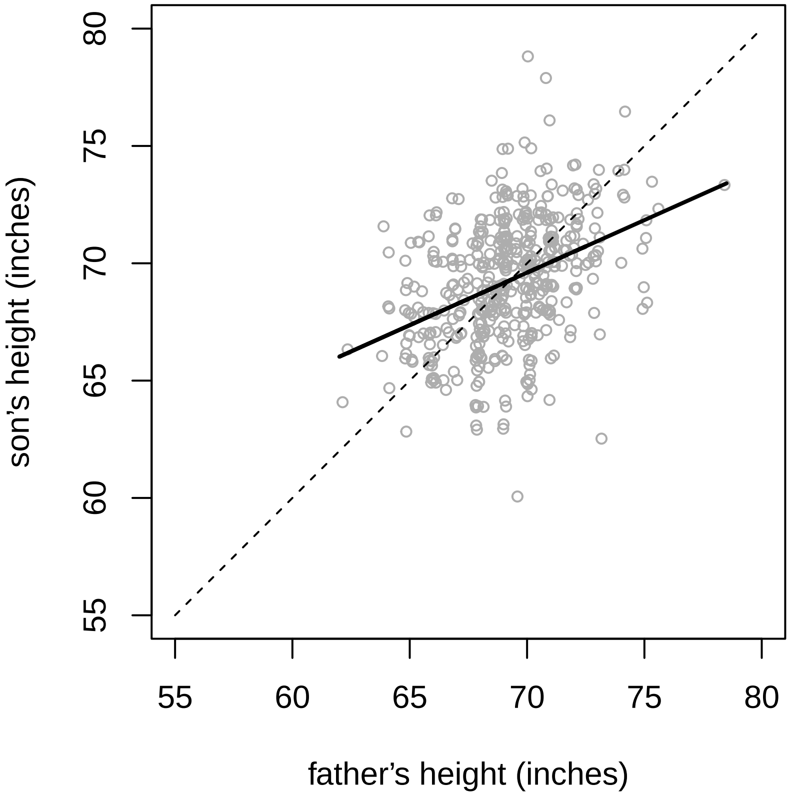
```
## [1] 2.37032
```

3.1.1 Figure 5.1 (page 124) Linear regression of sons' on fathers' heights

```r
# Heights of fathers of sons
FatherS = Father[Gender=="M"]

fit <- lm(Son ~ FatherS) # linear regression data in fit
Predicted <- predict(fit)   # Get the predicted values
summary(fit)
```

```
##
## Call:
## lm(formula = Son ~ FatherS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3774 -1.4968  0.0181  1.6375  9.3987
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.25891    3.38663   11.30   <2e-16 ***
## FatherS      0.44775    0.04894    9.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 463 degrees of freedom
## Multiple R-squared:  0.1531, Adjusted R-squared:  0.1513
## F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

```
FatherS.j <- jitter(FatherS, factor=5)
Son.j <- jitter(Son, factor=5)

xlims=ylims=c(55,80)
par(mfrow=c(1,1), mar=c(4,4,2,0), pty="s")  # square plot

plot(FatherS.j, Son.j, xlim=xlims,ylim=ylims,cex=0.7,
     xlab="father's height (inches)",ylab="son's height (inches)" , col="gray68")
lines(c(xlims[1],xlims[2]),c(xlims[1],xlims[2]),lty=2 )
lines(Predicted~FatherS,lwd=2)
```



3.1.2 Now in ggplot

```
library(ggplot2)
# create new data frame with exact and jittered, and predcted values

Males = cbind.data.frame(FatherS,FatherS.j,Son,Son.j,Predicted)

p <- ggplot(Males, aes(x=FatherS, y=Son)) # initial plot object
```

```
p <- p + geom_point(x=FatherS.j,y=Son.j,shape= 1) # defines scatter type plot
p <- p + labs(x="Father's height (inches)", y= "Son's height (inches)") # adds x and y axis label
p <- p + theme(legend.position="none")#, legend.box = "horizontal") # removes the legend
p <- p + expand_limits(x = c(55,80),y = c(55,80)) # expand the axis limits
p <- p + geom_line(aes(FatherS,Predicted),size=1.5) # add previously fitted linear regression lin

p <- p + geom_abline(slope=1, linetype="dashed") # line to represent equality between son and fat


# select single data points by CSV datarow numbers
pointA=c(137)
pointB=c(28)

# plot residual line and end points for selectedpointA
p <- p + geom_point(aes(x=FatherS.j[pointA], y = Predicted[pointA]), shape = 1)
p <- p + geom_point(aes(x=FatherS.j[pointA], y = Son.j[pointA]), shape = 1)
p <- p + geom_segment(linetype="dashed", size=1, colour="purple",aes(x=FatherS.j[pointA],y=Son.j[

# plot residual line and end points for pointB
p <- p + geom_point(aes(x=FatherS.j[pointB], y = Predicted[pointB]), shape = 1)
p <- p + geom_point(aes(x=FatherS.j[pointB], y = Son.j[pointB]), shape = 1)
p <- p + geom_segment(linetype="dashed", size=1, colour="purple",aes(x=FatherS.j[pointB],y=Son.j[

p #displays the result
```

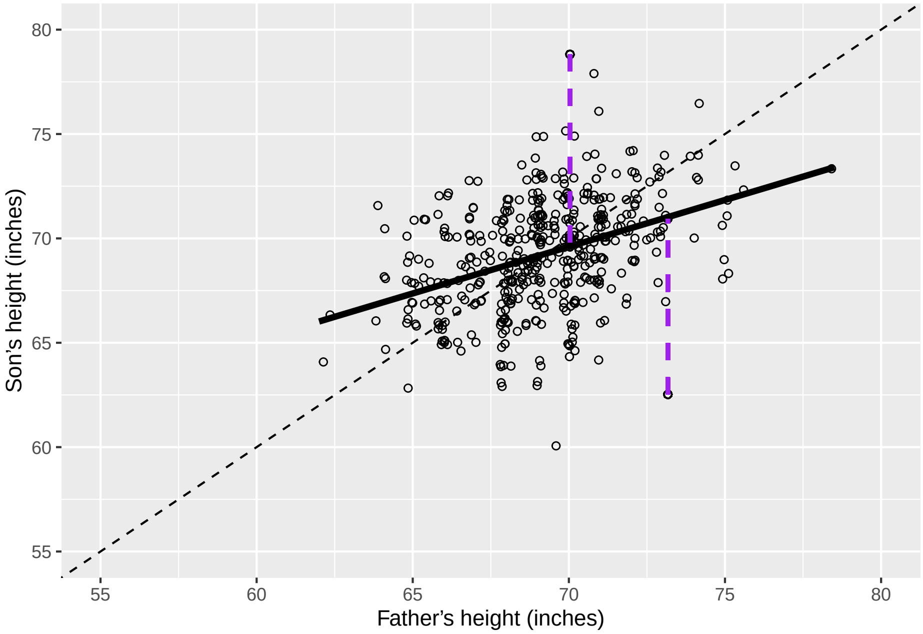


Figure 5.1 Scatter of heights of 465 fathers and sons from Galton's data (many fathers are repeated since they have multiple sons). A jitter has been added to separate the points, and the diagonal dashed line represents exact equality between son and father's heights. The solid line is the standard 'best-fit' line. Each point gives rise to a 'residual' (dashed line), which is the size of the error were we to use the line to predict a son's height from his father's.

3.2 Logistická regrese

Data from 1991-1995 are contained in 02-5-child-heart-surgery-1991-x.csv, and taken from D. J. Spiegelhalter et al., Commissioned Analysis of Surgical Performance Using Routine Data: Lessons from the Bristol Inquiry.

```
library(ggplot2)

child.1991 <- read.csv("02-5-child-heart-surgery-1991-x.csv") # read data into datafra

attach(child.1991)
# leave first row (Bristol) out of the fit
fit=glm(Survivors/Operations ~ Operations, weight=Operations, family="binomial",data=c
summary(fit)
```

```
##
## Call:
## glm(formula = Survivors/Operations ~ Operations, family = "binomial",
##     data = child.1991[-1, ], weights = Operations)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.7348545  0.1410843  12.297   <2e-16 ***
## Operations  0.0009615  0.0003807   2.526   0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 18.623  on 10  degrees of freedom
## Residual deviance: 12.169  on  9  degrees of freedom
## AIC: 72.622
##
## Number of Fisher Scoring iterations: 4
```

```
predictions=100*predict(fit, data.frame(Operations=0:700), type="response")  # predict

pred.frame = data.frame(Extremes=0:700,predictions ) # data frame for predictions
```

```
p <- ggplot()
p <- p + geom_point(child.1991, mapping=aes(x=Operations, y=100*Survivors/Operations, col=Hospita
p <- p + expand_limits(x = c(0,700),y=c(70,100))
p <- p +  labs(x="Number of operations", y = "% 30-day survival", title="(a) Survival in under-1s
p <- p + geom_line(dat=pred.frame, aes(x=Extremes,y=predictions), size=1) # add previously fitte

p
```

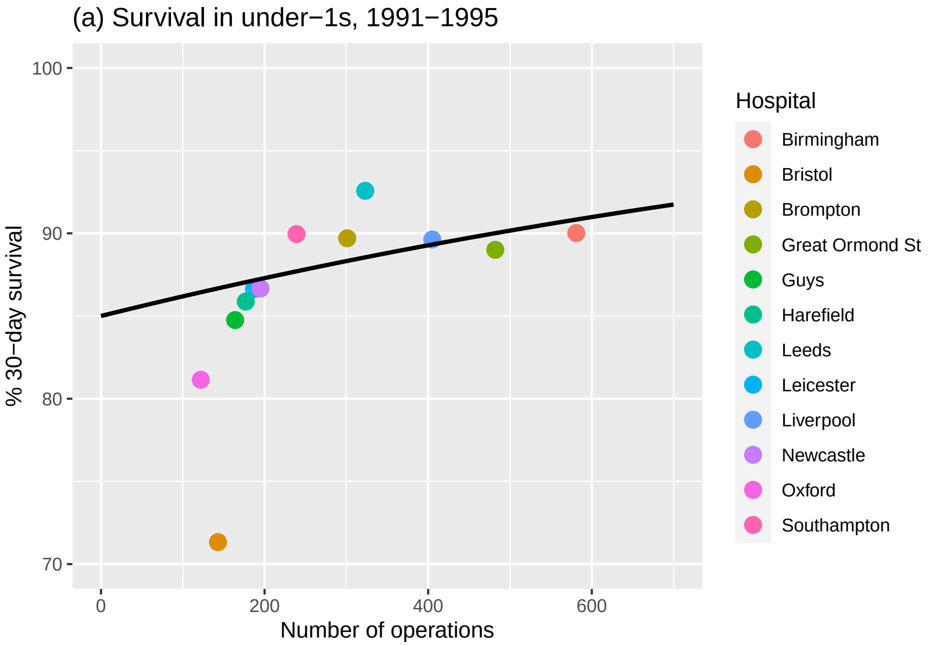
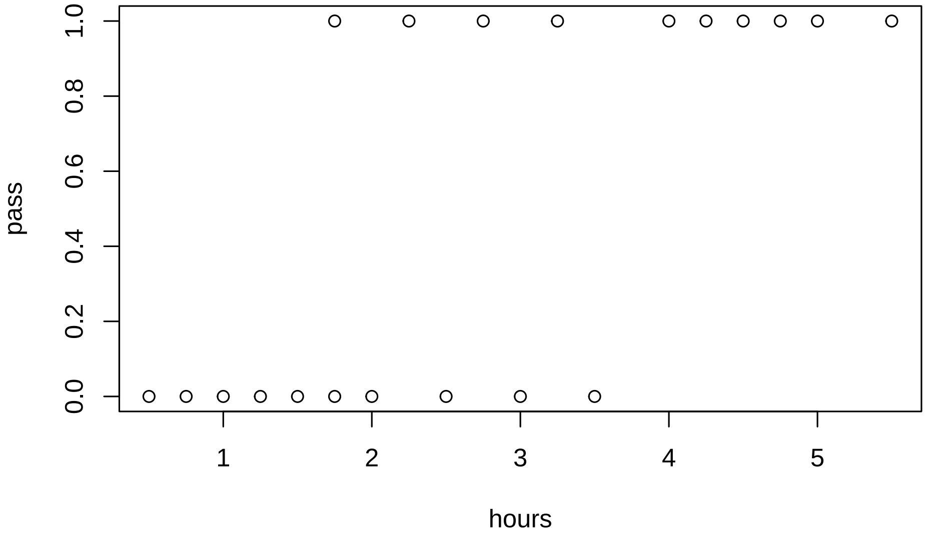


Figure 5.2 Fitted logistic regression model for child heart surgery data for under-1s in UK hospitals between 1991 and 1995. Hospitals treating more patients have better survival. The line is part of a curve that will never reach 100%, and is fitted ignoring the outlying data-point representing Bristol.

3.3 Jiný nelingvistický příklad

- z Wikipedie

```
hours <- c(0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 1.75, 2.00, 2.25, 2.50, 2.75, 3.00, 3.25, 3.50, 4.

pass <- c(0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1)

plot(hours, pass)
```



```
df <- data.frame(hours, pass)

logitm <- glm(pass ~ hours ,data = df, family = "binomial")

logitm
```

```
##
## Call:  glm(formula = pass ~ hours, family = "binomial", data = df)
##
## Coefficients:
## (Intercept)        hours
##      -4.078        1.505
##
## Degrees of Freedom: 19 Total (i.e. Null);  18 Residual
## Null Deviance:       27.73
## Residual Deviance: 16.06     AIC: 20.06
```

```
summary(logitm)
```

```
##
## Call:
## glm(formula = pass ~ hours, family = "binomial", data = df)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.0777     1.7610  -2.316   0.0206 *
## hours         1.5046     0.6287   2.393   0.0167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.726  on 19  degrees of freedom
## Residual deviance: 16.060  on 18  degrees of freedom
## AIC: 20.06
##
## Number of Fisher Scoring iterations: 5
```

```
exp(1.5046)
```

```
## [1] 4.502352
```

3.4 Lingvistický příklad

- Neg-raising experiment

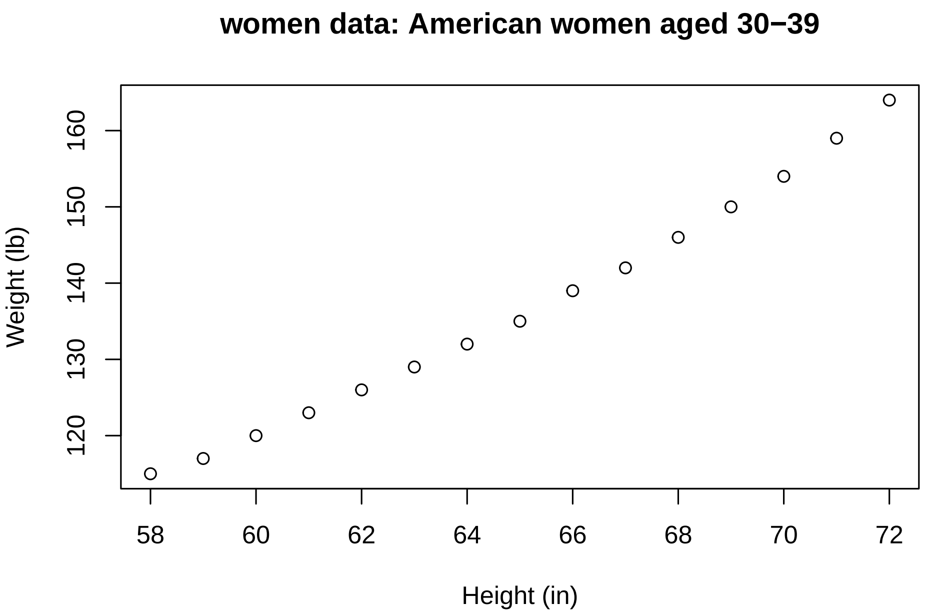
Chapter 4

Shrnutí a domácí úkol (do prezentace)

- lineární regrese

  - sonický šroubovák humanitních věd
  - jednoduché použítí je naprosto intuitivní a jednoduché:

```r
require(graphics)

plot(women, xlab = "Height (in)", ylab = "Weight (lb)",
     main = "women data: American women aged 30-39")
```



```
data("women")

head(women)
```

```
##   height weight
## 1     58    115
## 2     59    117
## 3     60    120
## 4     61    123
## 5     62    126
## 6     63    129
```

```
help("women")
```

```
lm <- lm(women$weight ~ women$height)

lm
```

```
##
## Call:
## lm(formula = women$weight ~ women$height)
##
## Coefficients:
##  (Intercept)  women$height
##       -87.52          3.45
```

- y-intercept (-87.52)
  - žena s nulovou výškou by vážila -87.52 liber

- regression-coefficient (3.45)
  - numericky vyjádřený vztah mezi explanatory a dependent variable
  - nárustek závislé proměnné, vzroste-li explanatory proměnná a 1 jednotku

```
summary(lm)
```

```
##
## Call:
## lm(formula = women$weight ~ women$height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7333 -1.1333 -0.3833  0.7417  3.1167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -87.51667    5.93694  -14.74 1.71e-09 ***
## women$height   3.45000    0.09114   37.85 1.09e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.525 on 13 degrees of freedom
## Multiple R-squared:  0.991,  Adjusted R-squared:  0.9903
## F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

- každý koeficient má:

1. sign (znaménko): pozitivní nebo negativní
2. velikost (síla efektu)
3. signifikance: pravděpodobnost nulové hypotéze vůči danému vzorku

4.1 Lingvistické použití

- v experimentu je závislá proměnná odpověď
- vysvětlující proměnná (manipulovaná) podmínka
- vždy stanovíme jednu podmínku jako referenční
- od ní pak model poměřuje "zlepšování"/"zhoršování" jinými podmínkami
- základ formule: as.numeric(rating1) ~ condition + (1|participant) + (1|item), data=items
- závislá ~ vysvětlující_podmínka + (random effect1) + ...

```
library(plyr)
library(dplyr)

items <- read.csv2("clean_items.csv", encoding = 'UTF-8', header = TRUE)
items <- items %>%
     mutate(condition=replace(condition, condition == "item-méně_než", "fewer")) %>%
     mutate(condition=replace(condition, condition == "item-nanejvýš", "at-most")) %>%
     mutate(condition=replace(condition, condition == "item-ne_víc_než", "no-more")) %
     mutate(condition=replace(condition, condition == "item-trochu_méně", "slightly-les
     as.data.frame()

ddply(items, .(condition), summarise, Means = mean(rating1, na.rm=TRUE))
```

```
##       condition    Means
## 1       at-most 1.262755
## 2         fewer 2.512755
## 3       no-more 1.311224
## 4 slightly-less 2.211735
```

```
ddply(items, .(condition), summarise, Medians = median(rating1,na.rm=TRUE))
```

```
##       condition Medians
## 1       at-most       1
## 2         fewer       2
## 3       no-more       1
## 4 slightly-less       2
```

```
library(lmerTest)

items$condition <- as.factor(items$condition)

items$condition <- relevel(items$condition, ref="at-most")

m1 <- lmer(as.numeric(rating1) ~ condition  + (1|participant) + (1|item), data=items)

summary(m1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: as.numeric(rating1) ~ condition + (1 | participant) + (1 | item)
##    Data: items
##
## REML criterion at convergence: 4936.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1683 -0.6481 -0.2050  0.4471  3.6096
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  participant (Intercept) 0.1414   0.3761
##  item        (Intercept) 0.1375   0.3708
##  Residual                1.2402   1.1137
## Number of obs: 1568, groups:  participant, 98; item, 16
##
## Fixed effects:
##                         Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)            1.271e+00  1.149e-01 2.942e+01  11.063  5.3e-12 ***
## conditionfewer         1.242e+00  7.959e-02 1.452e+03  15.601  < 2e-16 ***
## conditionno-more       3.996e-02  7.959e-02 1.452e+03   0.502    0.616
## conditionslightly-less 9.322e-01  7.961e-02 1.452e+03  11.710  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cndtnf cndtnn-
## conditinfwr -0.346
## conditnn-mr -0.346  0.500
## cndtnslght- -0.346  0.500  0.500
```

4.2 Domácí úkol do prezentace

- lineární model dvou podmínek
- měl by vyjít analogicky vůči zdrojovému textu
- random effects: není nutné, ale lze za ně získat body navíc

4.3 Dodatky

Data from 1991-1995 are contained in 05-1-galton-x.csv, Although the book says the data is from HistData: Data Sets from the History of Statistics and Data Visualization, 2018, I have actually used this version of Galton's Height Data

```r
galton<-read.csv("05-1-galton-x.csv",header=TRUE) # read csv file into dataframe galton
attach(galton) #uncomment if/while necessary
```

```
## The following objects are masked from galton (pos = 4):
##
##     Family, Father, Gender, Height, Kids, Mother
```

```r
summary(galton)
```

```
##     Family              Father          Mother          Gender
##  Length:898         Min.   :62.00   Min.   :58.00   Length:898
##  Class :character   1st Qu.:68.00   1st Qu.:63.00   Class :character
##  Mode  :character   Median :69.00   Median :64.00   Mode  :character
##                        Mean   :69.23   Mean   :64.08
##                        3rd Qu.:71.00   3rd Qu.:65.50
##                        Max.   :78.50   Max.   :70.50
##      Height            Kids
##  Min.   :56.00   Min.   : 1.000
##  1st Qu.:64.00   1st Qu.: 4.000
##  Median :66.50   Median : 6.000
##  Mean   :66.76   Mean   : 6.136
##  3rd Qu.:69.70   3rd Qu.: 8.000
##  Max.   :79.00   Max.   :15.000
```

```
# summary statistics
# need means for unique fathers and mothers - identify first mention of each family
Unique.Fathers=numeric()
Unique.Mothers=numeric()
nunique=1 # number of unique families
Unique.Fathers[1] = Father[1]
Unique.Mothers[1] = Mother[1]
 for(i in 2:length(Family))
{
    if(Family[i] != Family[i-1]){
      nunique=nunique+1
    Unique.Fathers[nunique]=Father[i]
    Unique.Mothers[nunique]=Mother[i]
    }
  }

length(Unique.Fathers)
```

```
## [1] 197
```

```
summary(Unique.Fathers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   62.00   68.00   69.50   69.35   71.00   78.50
```

```
sd(Unique.Fathers)
```

```
## [1] 2.622034
```

```
length(Unique.Mothers)
```

```
## [1] 197
```

```
summary(Unique.Mothers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   58.00   62.70   64.00   63.98   65.50   70.50
```

```r
sd(Unique.Mothers)
```

```
## [1] 2.355607
```

```r
Son = Height[Gender=="M"]
length(Son)
```

```
## [1] 465
```

```r
summary(Son)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.00   67.50   69.20   69.23   71.00   79.00
```

```r
sd(Son)
```

```
## [1] 2.631594
```

```r
Daughter = Height[Gender=="F"]
length(Daughter)
```

```
## [1] 433
```

```r
summary(Daughter)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   56.00   62.50   64.00   64.11   65.50   70.50
```

```r
sd(Daughter)
```

```
## [1] 2.37032
```

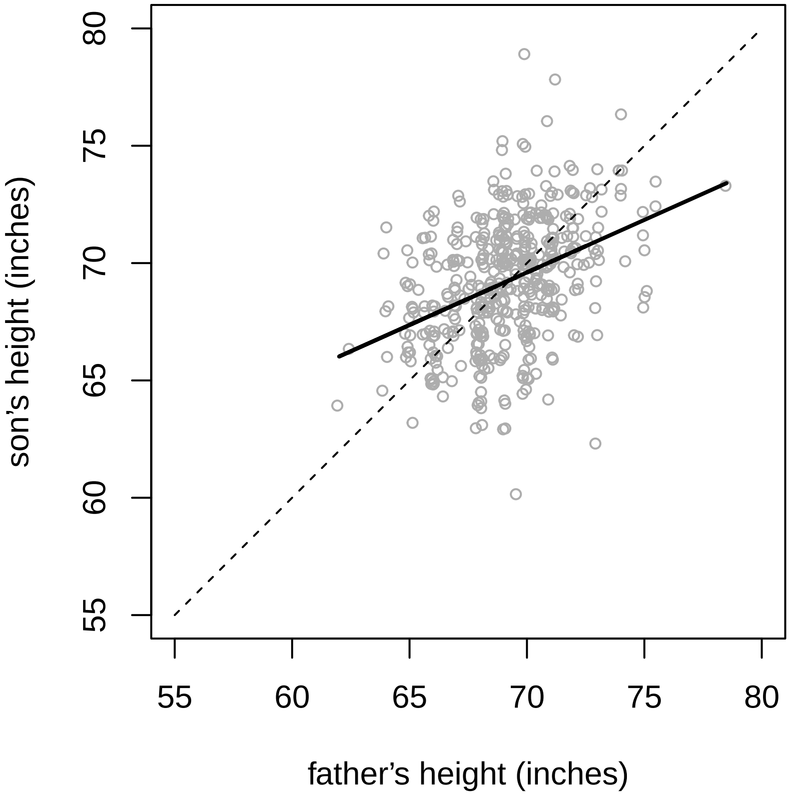
4.3.1 Figure 5.1 (page 124) Linear regression of sons' on fathers' heights

```r
# Heights of fathers of sons
FatherS = Father[Gender=="M"]

fit <- lm(Son ~ FatherS) # linear regression data in fit
Predicted <- predict(fit)   # Get the predicted values
summary(fit)
```

```
##
## Call:
## lm(formula = Son ~ FatherS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3774 -1.4968  0.0181  1.6375  9.3987
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 38.25891    3.38663   11.30   <2e-16 ***
## FatherS      0.44775    0.04894    9.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.424 on 463 degrees of freedom
## Multiple R-squared:  0.1531, Adjusted R-squared:  0.1513 
## F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

```
FatherS.j <- jitter(FatherS, factor=5)
Son.j <- jitter(Son, factor=5)

xlims=ylims=c(55,80)
par(mfrow=c(1,1), mar=c(4,4,2,0), pty="s")  # square plot

plot(FatherS.j, Son.j, xlim=xlims,ylim=ylims,cex=0.7,
     xlab="father's height (inches)",ylab="son's height (inches)" , col="gray68")
lines(c(xlims[1],xlims[2]),c(xlims[1],xlims[2]),lty=2 )
lines(Predicted~FatherS,lwd=2)
```



4.3.2 Now in ggplot

```
library(ggplot2)
# create new data frame with exact and jittered, and predcted values

Males = cbind.data.frame(FatherS,FatherS.j,Son,Son.j,Predicted)

p <- ggplot(Males, aes(x=FatherS, y=Son)) # initial plot object
p <- p + geom_point(x=FatherS.j,y=Son.j,shape= 1) # defines scatter type plot
p <- p + labs(x="Father's height (inches)", y= "Son's height (inches)") # adds x and y axis label
p <- p + theme(legend.position="none")#, legend.box = "horizontal") # removes the legend
p <- p + expand_limits(x = c(55,80),y = c(55,80)) # expand the axis limits
p <- p + geom_line(aes(FatherS,Predicted),size=1.5) # add previously fitted linear regression lin

p <- p + geom_abline(slope=1, linetype="dashed") # line to represent equality between son and fat


# select single data points by CSV datarow numbers
pointA=c(137)
```

```
pointB=c(28)

# plot residual line and end points for selectedpointA
p <- p + geom_point(aes(x=FatherS.j[pointA], y = Predicted[pointA]), shape = 1)
p <- p + geom_point(aes(x=FatherS.j[pointA], y = Son.j[pointA]), shape = 1)
p <- p + geom_segment(linetype="dashed", size=1, colour="purple",aes(x=FatherS.j[pointA

# plot residual line and end points for pointB
p <- p + geom_point(aes(x=FatherS.j[pointB], y = Predicted[pointB]), shape = 1)
p <- p + geom_point(aes(x=FatherS.j[pointB], y = Son.j[pointB]), shape = 1)
p <- p + geom_segment(linetype="dashed", size=1, colour="purple",aes(x=FatherS.j[pointB

p #displays the result
```

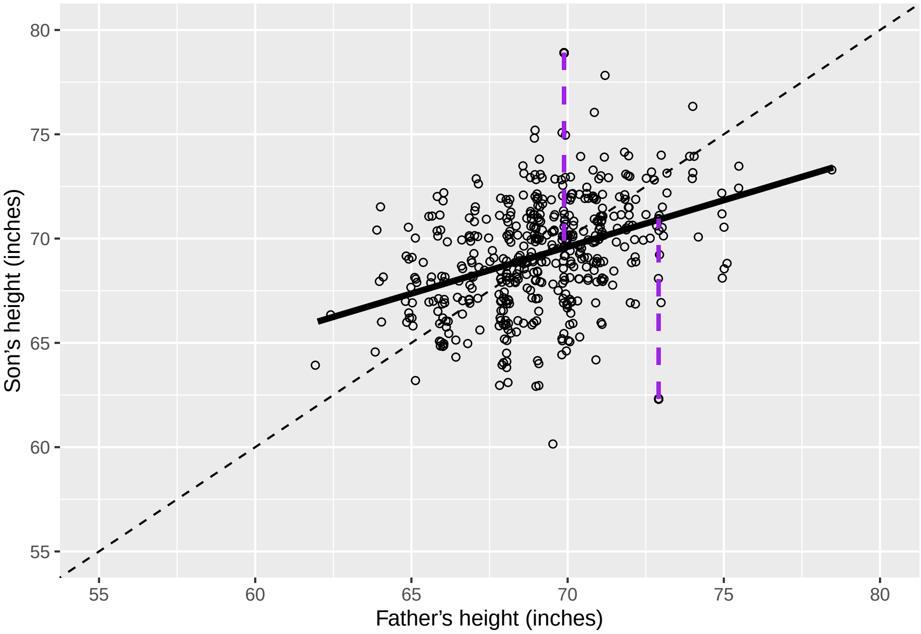


Figure 5.1 Scatter of heights of 465 fathers and sons from Galton's data (many fathers are repeated since they have multiple sons). A jitter has been added to separate the points, and the diagonal dashed line represents exact equality between son and father's heights. The solid line is the standard 'best-fit' line. Each point gives rise to a 'residual' (dashed line), which is the size of the error were we to use the line to predict a son's height from his father's.

Data from 1991-1995 are contained in 02-5-child-heart-surgery-1991-x.csv, and taken from D. J. Spiegelhalter et al., Commissioned Analysis of Surgical Performance Using Routine Data: Lessons from the Bristol Inquiry.

```
library(ggplot2)

child.1991 <- read.csv("02-5-child-heart-surgery-1991-x.csv") # read data into dataframe

attach(child.1991)
```

```
## The following objects are masked from child.1991 (pos = 4):
##
##     Deaths, Hospital, Operations, PercentageDying, Survivors,
##     ThirtyDaySurvival
```

```
# leave first row (Bristol) out of the fit
fit=glm(Survivors/Operations ~ Operations, weight=Operations, family="binomial",data=child.1991[-
summary(fit)
```

```
##
## Call:
## glm(formula = Survivors/Operations ~ Operations, family = "binomial",
##     data = child.1991[-1, ], weights = Operations)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.7348545  0.1410843  12.297   <2e-16 ***
## Operations  0.0009615  0.0003807   2.526   0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 18.623  on 10  degrees of freedom
## Residual deviance: 12.169  on  9  degrees of freedom
## AIC: 72.622
##
## Number of Fisher Scoring iterations: 4
```

```
predictions=100*predict(fit, data.frame(Operations=0:700), type="response")  # predictions for ex

pred.frame = data.frame(Extremes=0:700,predictions ) # data frame for predictions

p <- ggplot()
p <- p + geom_point(child.1991, mapping=aes(x=Operations, y=100*Survivors/Operations, col=Hospita
p <- p + expand_limits(x = c(0,700),y=c(70,100))
p <- p +  labs(x="Number of operations", y = "% 30-day survival", title="(a) Survival in under-1s
p <- p + geom_line(dat=pred.frame, aes(x=Extremes,y=predictions), size=1) # add previously fitted

p
```

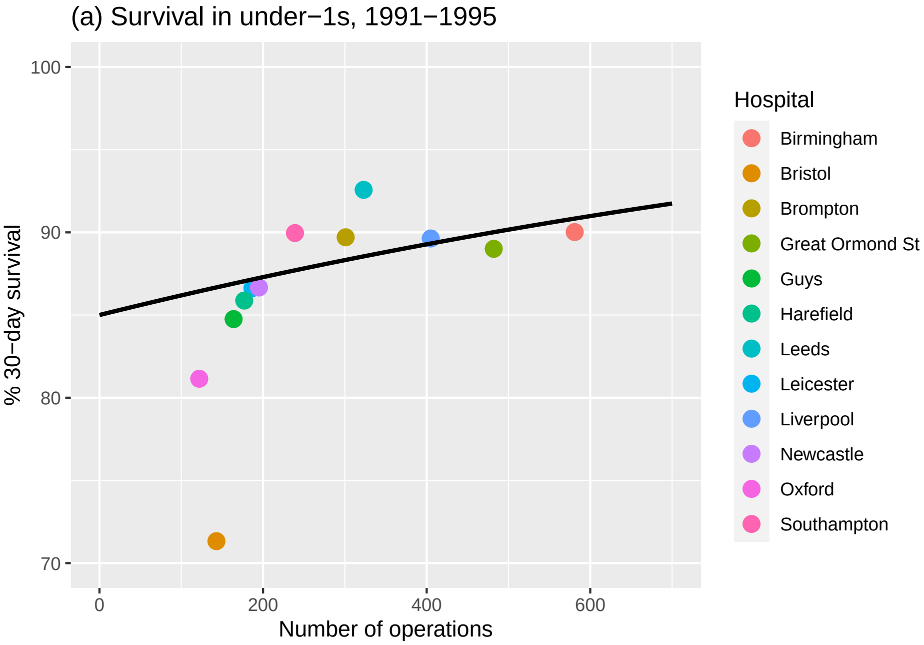


Figure 5.2 Fitted logistic regression model for child heart surgery data for under-1s in UK hospitals between 1991 and 1995. Hospitals treating more patients have better survival. The line is part of a curve that will never reach 100%, and is fitted ignoring the outlying data-point representing Bristol.

Bibliography