GLMs: binomial family

Mandy Vogel

University Leipzig

August 31, 2015

Overview

Ancova with a Binary Response Variable

Multiple Numeric Regressors

Summarizing the Fit of a Linear Model

Table of Contents I

Parasite Infection Example

- the binary response variable is parasite infection (infected or not)
- the explanatory variables are weight and age (continuous)
- and sex (categorical)
- we want to investigate if there is a different effect of age for each of the sexes on the outcome variable

Input/Output

```
> load("infection.rdata")
> summary(infection)
         infected          age                sex
 infected    :338   Min.   :  2.00   female:243
 not infected:162   1st Qu.: 46.00   male  :257
                    Median : 84.50
                    Mean   : 93.69
                    3rd Qu.:139.25
                    Max.   :200.00
```

Parasite Infection Example

Input/Output

```
> m.inf <- glm(infected~age*sex,family=binomial,
+                                 data=infection)
> summary(m.inf)
Call:
glm(formula = infected ~ age * sex, family = binomial,
                                       data = infection)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0411  -0.7307  -0.4363   0.6632   2.3215
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.000513   0.413639  -7.254 4.05e-13 ***
age          0.015657   0.003176   4.929 8.25e-07 ***
sex          0.116664   0.553956   0.211   0.8332
age:sex      0.011050   0.004612   2.396   0.0166 *

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 629.85  on 499  degrees of freedom
Residual deviance: 477.61  on 496  degrees of freedom
AIC: 485.61
```

Parasite Infection Example

- so for male at a age of 0 there is a probability of

Input/Output

```
> invlogit(coef(m.inf)[1])
(Intercept)
 0.04740269
```

- for females the probability at age 0 is

Input/Output

```
> invlogit(coef(m.inf)[1]+coef(m.inf)[3])
(Intercept)
 0.05295775
```

Compare Slopes

- so what about the slope?
- for males the underlying model is the following

$$\text{Pr(infection)} = \text{logit}^{-1}(-3.000513 + 0.015657 \cdot \text{age})$$

- for females the slope is almost twice as high

$$\text{Pr(infection)} = \text{logit}^{-1}(-2.883849 + 0.02670685 \cdot \text{age})$$

Compare Slopes

- looking at the odds ratios (which seem to be rather small)
- for males and females:

Input/Output

```
> exp(coef(m.inf)[2]) ## males
    age
1.01578
> exp(coef(m.inf)[2] + coef(m.inf)[4]) ## females
     age
1.027067
```

- these are the odds ratios for +1 time unit

Compare Slopes

- if time unit is days you get the odds ratio for +1 month by

Input/Output

```
> exp(30 * coef(m.inf)[2])
     age
1.599512
> exp(30 * (coef(m.inf)[2] + coef(m.inf)[4]))
     age
2.228225
```

- so keep in mind the scale you are measuring on

Compare Slopes

- we can also compare them by looking at the age where the probability to be infected is 50%
- this is the case when

$$-3.000513 + 0.015657 \cdot \text{age} = 0$$

respectively

$$-2.883849 + 0.02670685 \cdot \text{age} = 0$$

you can do it by hand or use R

Compare Slopes

- `solve()` solves systems of linear equations in the form A*x=b, where A is the matrix of cœfficients and b are the (negative) intercepts, here we have the special case with just one equation

Input/Output

```
> ## male
> solve(0.015657,3.000513)
[1] 191.6404
> ## female
> solve(0.02670685,2.883849)
[1] 107.9816
```

Compare Effects

- you can also use the `allEffects()` function (part of the `effects` package), which give you the probabilities for being infected on several ages for both sexes

Input/Output

```
> allEffects(m.inf)
 model: infected ~ age * sex

 age*sex effect
     sex
age             0          1
  2   0.04883687 0.05570148
  24  0.06756215 0.09596497
  46  0.09276694 0.16038932
  68  0.12610300 0.25582483
  90  0.16918450 0.38219715
  112 0.22322468 0.52680374
  134 0.28853152 0.66704908
  156 0.36399154 0.78286130
  178 0.44679328 0.86645480
  200 0.53265591 0.92110968
```

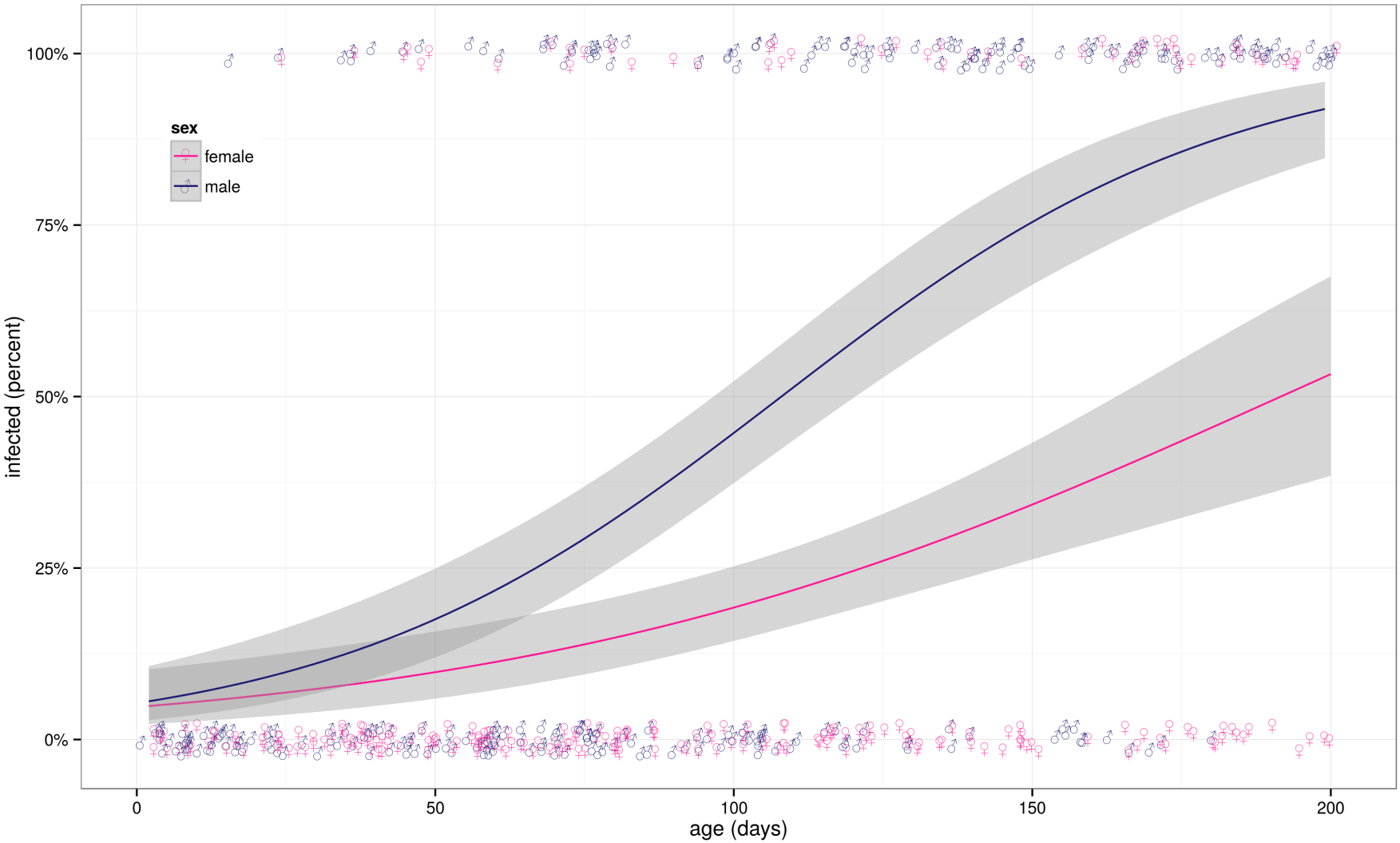
Compare Effects

- choose values of age

Input/Output

```
> allEffects(m.inf,
+            xlevels = list(age = seq(0,200,by = 50)))
 model: infected ~ age * sex

 age*sex effect
     sex
age         female       male
  0   0.04740269 0.05295775
  50  0.09817379 0.17530204
  100 0.19234385 0.44690980
  150 0.34253427 0.75439251
  200 0.53265591 0.92110968
```

Parasite Infection graph



Exercise

Try to reproduce the plot! Hints:

1. set up a ggplot object, think about the æsthetics (`aes()`). Which quality of the graph you wanna set to which variable?
2. begin with the lines (`geom_smooth()`)
3. add the points (`geom_jitter()`; do not think about the symbols in the first place; try to adjust the width and height appropriately)
4. change the colour of the lines and points (`scale_colour_manual()`); I used midnightblue for male and deeppink for female
5. change the symbols (`scale_shape_manual()`); use

```
values = c("male" = "\u2642","female" = "\u2640")
```

   as values
6. set the axes titles
7. change to text of the y axis to percentage
8. etc

Exercise

Try to reproduce the plot! Hints:

1. read the data melanoma.dat, you find a codebook under melanomacodebook.xlsx
2. start by looking at the number of cases by each variable separately, ignoring age and sex (`table()`)
3. recode sex, skin complexion, hair colour, eye colour, freckles into factors; here is a example for sex

   ```
   > mel$sex <- factor(mel$sex,labels=c("M","F"))
   ```

4. build up a model to see the effect of skin colour, look at hair eyes and freckles in the same way.
5. now look at the effect of freckles, but control for age and sex
6. try to find an appropriate way to visualize the last model

Table of Contents I

Adjusting

- what dœs it mean, this adjusting for the other variables?
- trying to describe a partial effect on an explanatory variable
- this information can be extracted from examination of the residuals
- so first: what are residuals?

Residuals & Errors

- they are closely related but not the same
- error is the difference between an observed and a true value
- residual is the difference an observed value and an estimated (or fitted) value

Extracting Residuals

- in R the function `resid()` is used to extract the residual from a fitted model
- as a example I use a data frame from the Scottish Hill Runners Association

Input/Output

```
> m1 <- lm(time ~ climb + distance, data = sc.race)
> m1

Call:
lm(formula = time ~ climb + distance, data = sc.race)

Coefficients:
(Intercept)        climb     distance
    -13.109       11.780        6.351
```

Extracting Residuals

Input/Output

```
> resid(m1)
           1            2            3            4            5
 5.654075838 -6.097514205 -1.949301243  1.652279007 -11.594100819
           6            7            8            9           10
 1.759046334 27.762266559  1.948712067  1.679667440  7.095732032
          11           12           13           14           15
 3.213520827  0.843507719 -8.141818080 13.242751997 -6.210948044
          16           17           18           19           20
-13.491402248 -3.301117259  6.052638082 -9.830881494  3.680216374
          21           22           23           24           25
 6.471644737  1.110007329 -3.263474275  7.734520249  5.667604806
          26           27           28           29           30
 -9.635536908  0.008712067 -1.267352525  1.712107934 -3.586292630
          31           32           33           34           35
-16.653588589  0.314738687  7.784661945  2.616138471 -12.981222181
```

Residuals and Adjusting

- suppose you have one outcome y and two explanatory variables $x1$ and $x2$
- if you regress y on a variable $x2$ (1)
- and $x1$ on $x2$ (2)
- and then the residuals from (1) on the residuals from (2)
- this last fit is identical to the partial effect of $x1$

Residuals and Adjusting

To get the partial effect of climb in model `m1`

- if you regress $time$ on a variable $climb$ (1)

Input

```
> m.climb <- lm(time ~ distance, data = sc.race)
```

- and $climb$ on $distance2$ (2)

Input

```
> m.dist <- lm(climb ~ distance, data = sc.race)
```

- and then the residuals from (1) on the residuals from (2)

Input

```
> m.res <- lm(resid(m.climb) ~ resid(m.dist))
```

Residuals and Adjusting

Input

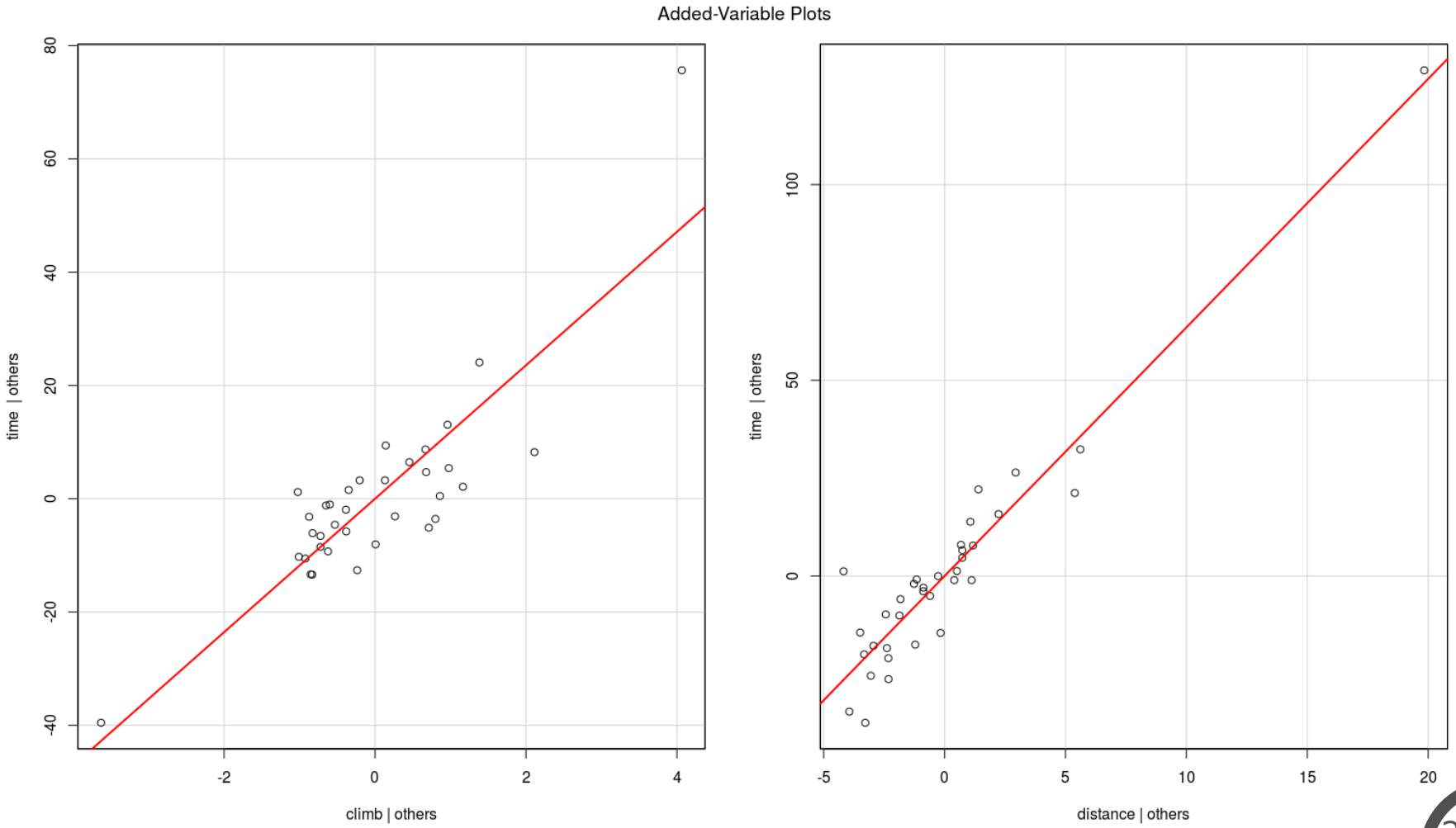
```
> m.res

Call:
lm(formula = resid(m.climb) ~ resid(m.dist))

Coefficients:
  (Intercept)  resid(m.dist)
   -7.465e-16      1.178e+01

> m1

Call:
lm(formula = time ~ climb + distance, data = sc.race)

Coefficients:
(Intercept)        climb     distance
    -13.109       11.780        6.351
```

Residuals and Adjusting

- this concept also holds for partial correlation
- the `avPlots()` command (`car` package provides a feasible way to plot a partial effect

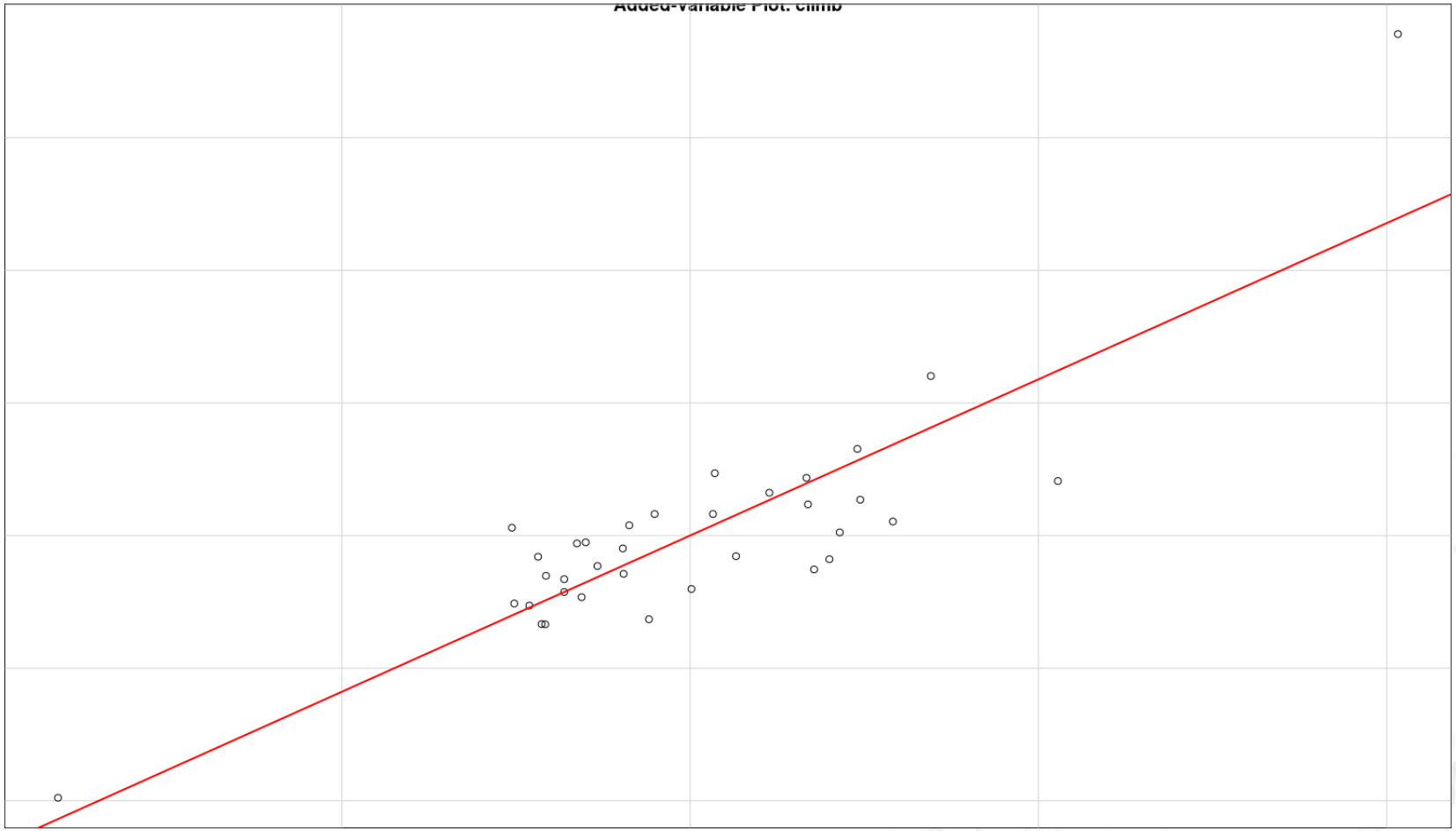
avPlots()

```
> require(car)
> avPlots(m1)
```



avPlots()

```
avPlot(m1,"climb")
```



avPlots()

- avPlots also returns the aforementioned residuals

Input/Output

```
> head(cbind(resid(m.dist),
+            avPlot(m1,"climb"),
+            resid(m.climb)))
                    climb        time
1 -0.2037877 -0.2037877   3.253430   3.253430
2  0.9769679  0.9769679   5.411298   5.411298
3 -0.6230321 -0.6230321  -9.288702  -9.288702
4 -1.0098511 -1.0098511 -10.243901 -10.243901
5  1.1645426  1.1645426   2.124366   2.124366
6  0.9605426  0.9605426  13.074366  13.074366
```

Table of Contents I

Exercise

- get the mean and the standard deviation of the three numeric variables in the `sc.race` data frame
- use the `pairs()` command and the `cor()` command to get a scatterplot matrix and the respective correlation matrix (Hint: these commands only work on numeric columns, so you have to get rid off the non-numeric ones. Remember indexing with negative integers)

Exercise

- get the mean and the standard deviation of the three numeric variables in the `sc.race` data frame

Input/Output

```
> describe(sc.race[,-1])
         vars  n  mean    sd median trimmed   mad   min    max  range skew
distance    1 35  7.53  5.52   6.00    6.55  2.22  2.00  28.00  26.00 1.99
climb       2 35  1.82  1.62   1.00    1.54  0.74  0.30   7.50   7.20 1.68
time        3 35 56.09 50.39  36.37   46.46 20.95 15.95 204.62 188.67 1.77
         kurtosis   se
distance     3.83 0.93
climb        2.66 0.27
time         2.07 8.52
```

Solution

- get the correlation matrix

Input/Output

```
> M <- cor(sc.race[,-1])
> M
          distance     climb      time
distance 1.0000000 0.6523461 0.9430944
climb    0.6523461 1.0000000 0.8326535
time     0.9430944 0.8326535 1.0000000
```

Solution

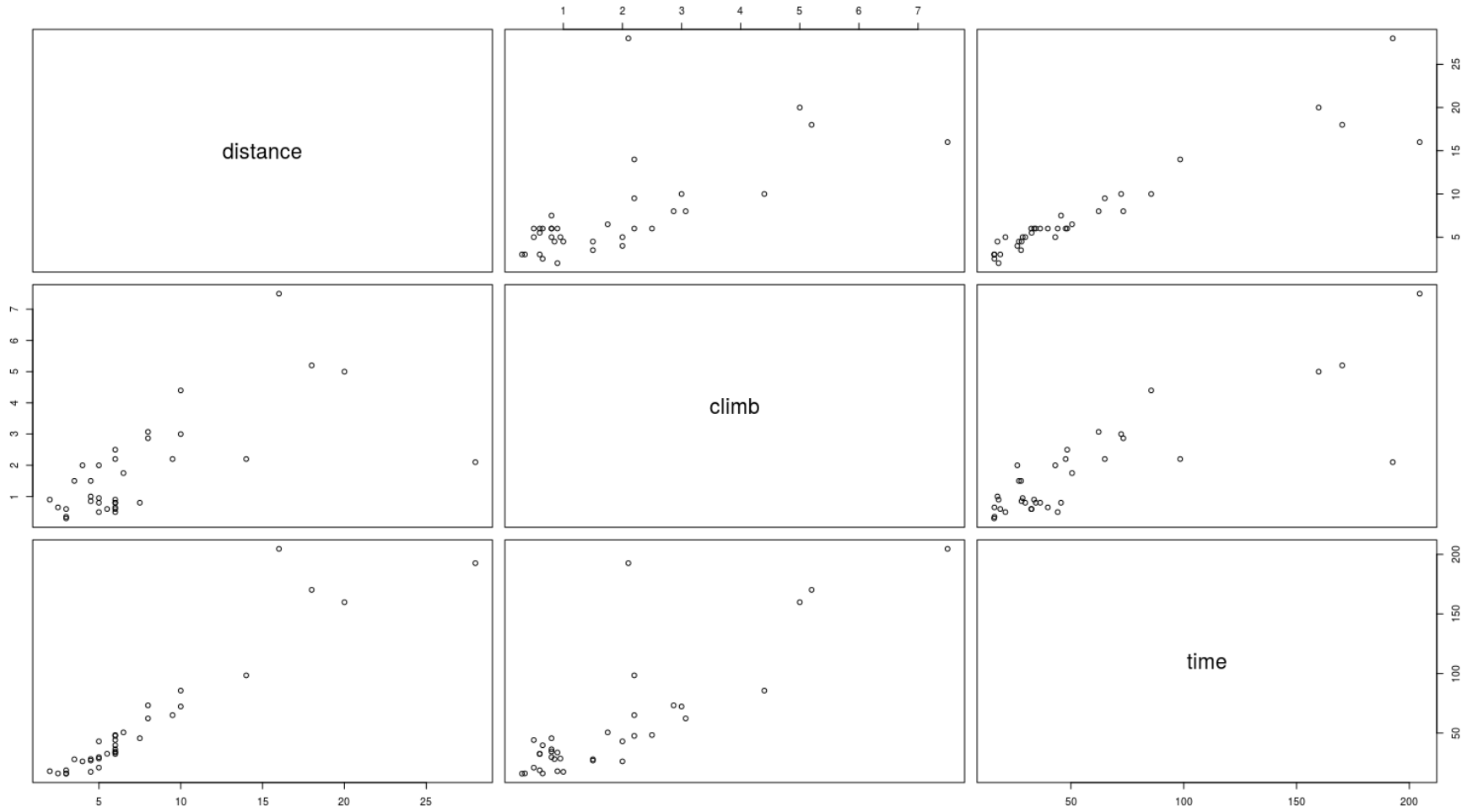
- get the scatterplot matrix

Input/Output

```
> pairs(sc.race[,-1])
```

Solution

- get the scatterplot matrix



Solution

- get the scatterplot matrix

Input/Output

```
> pairs(sc.race[,-1],panel = panel.smooth)
```

Solution

- get the scatterplot matrix