Regression

- Use regression when one variable is an outcome (*response*, y).
- See if/how response depends on other variable(s), *explanatory*, $x_1, x_2, \ldots$.
- Can have *one* or *more than one* explanatory variable, but always one response.
- Assumes a *straight-line* relationship between response and explanatory.
- Ask:
  - *is there* a relationship between y and x's, and if so, which ones?
  - what does the relationship look like?

A regression with one x

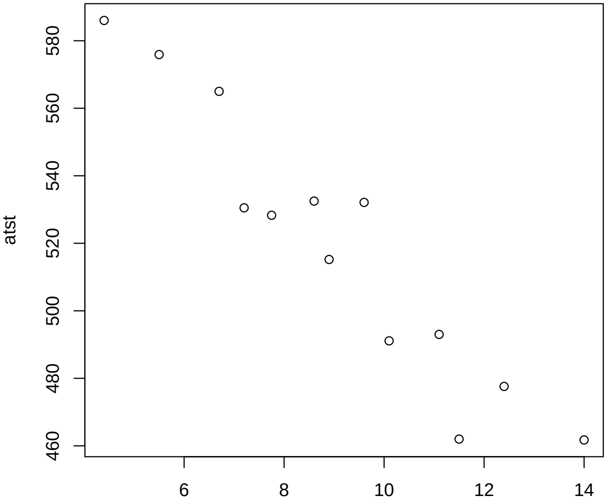
13 children, measure average total sleep time (ATST, mins) and age (years) for each. See if ATST depends on age. Data in `sleep.txt`, ATST then age. Read in data:

```
> sleep=read.table("sleep.txt",header=T)
> attach(sleep)
```

`attach` makes columns of data frame available for use.

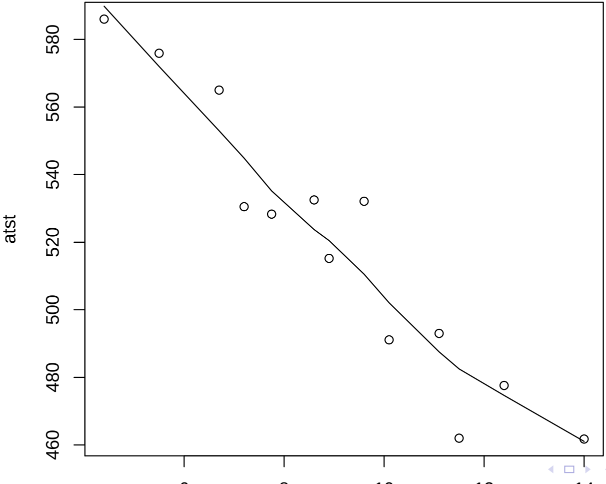
The scatterplot

```
> plot(age,atst)
```



The scatterplot, improved

```
> plot(age,atst)
> lines(lowess(age,atst))
```



The regression

Scatterplot shows no obvious curve, and a pretty clear downward trend. So we can run the regression:

```
> sleep.1=lm(atst~age)
> summary(sleep.1)
Call:
lm(formula = atst ~ age)

Residuals:
    Min      1Q  Median      3Q     Max
-23.011  -9.365   2.372   6.770  20.411

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  646.483     12.918   50.05 2.49e-14 ***
age          -14.041      1.368  -10.26 5.70e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Conclusions

- The relationship appears to be a straight line, with a downward trend.
- F-tests for model as a whole and t-test for slope (same) both confirm this.
- Slope is -14, so a 1-year increase in age goes with a 14-minute decrease in ATST on average.

CI for mean response and prediction intervals

Once useful regression exists, use it for prediction:

- To get a single number for prediction at a given x, substitute into regression equation, eg. age 10: predicted ATST is $646.48 - 14.04(10) = 506$ minutes.
- To express uncertainty of this prediction:
  - *CI for mean response* expresses uncertainty about mean ATST for all children aged 10, based on data.
  - *Prediction interval* expresses uncertainty about predicted ATST for a new child aged 10 whose ATST not known. More uncertain.
- Also do above for a child aged 3.

Intervals

- Make new data frame with these values for age
- Feed into predict

```
> ages.new=data.frame(age=c(10,3))
> ages.new

  age
1  10
2   3

> pc=predict(sleep.1,ages.new,interval="c")
> pp=predict(sleep.1,ages.new,interval="p")
> cbind(ages.new,pc,pp)

  age      fit      lwr      upr      fit      lwr      upr
1  10 506.0729 497.5574 514.5883 506.0729 475.8982 536.2475
2   3 604.3602 584.4305 624.2899 604.3602 569.2149 639.5055
```

Comments

- Age 10 closer to centre of data, so intervals are both narrower than those for age 3.
- Age 3 assumes that straight line continues to hold (don't have any data to support that)
- Prediction intervals bigger than CI for mean (additional uncertainty).

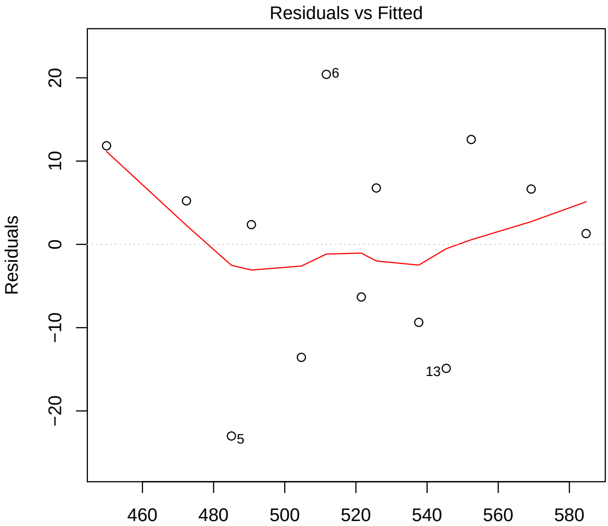
Diagnostics

How do we tell whether a straight-line regression is appropriate?

- Before: check scatterplot for straight trend.
- After: plot *residuals* (observed minus predicted response) against predicted values. Aim: a plot with no pattern.

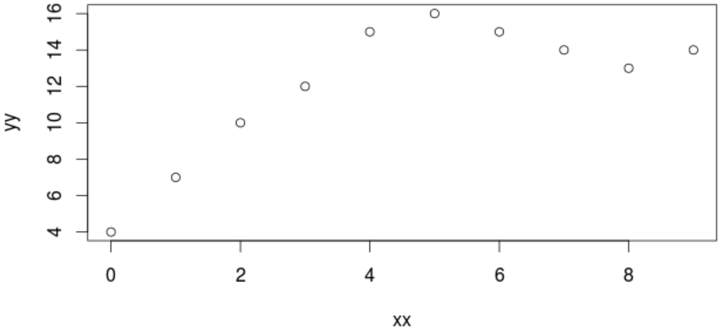
Output

> *plot(sleep.1)*



An inappropriate regression

Scatterplot of different data:



Trend goes up, then levels off, but a line would keep going up.

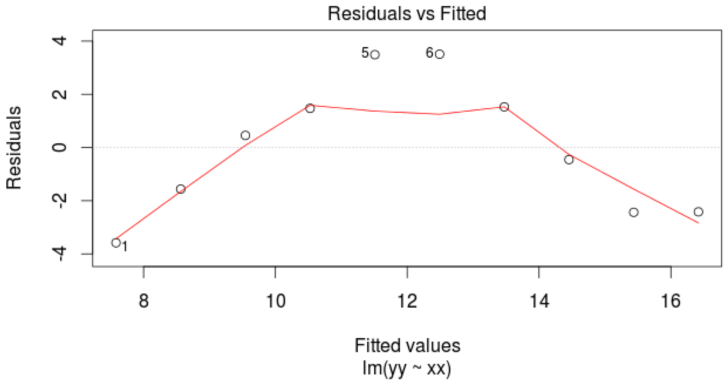
Regression line

Try fitting a regression line anyway:

```
> curvy=read.table("curvy.txt",header=T)
> attach(curvy)
> plot(xx,yy)
> curvy.1=lm(yy~xx)
> summary(curvy.1)
```

```
Call:
lm(formula = yy ~ xx)

Residuals:
   Min     1Q Median     3Q    Max
-3.582 -2.204  0.000  1.514  3.509

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.5818     1.5616   4.855  0.00126 **
xx            0.9818     0.2925   3.356  0.00998 **
---
```

Residual plot

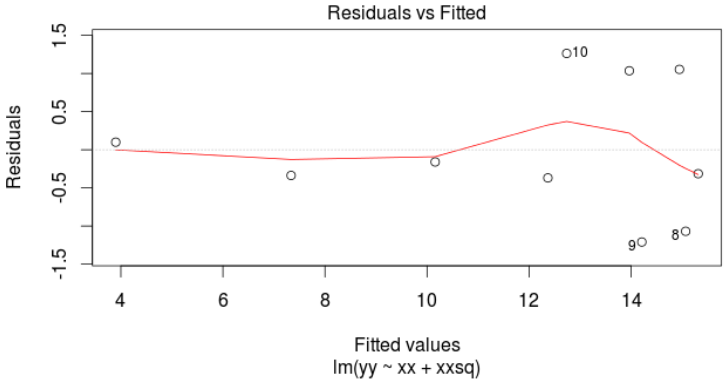


Residual plot has *curve*: middle residuals positive, high and low ones negative. Bad.

Fixing it up

```
> xxsq=xx^2
> curvy.2=lm(yy~xx+xxsq)
> summary(curvy.2)
Call:
lm(formula = yy ~ xx + xxsq)

Residuals:
    Min       1Q   Median       3Q      Max 
-1.2091 -0.3602 -0.2364  0.8023  1.2636 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.90000    0.77312   5.045 0.001489 ** 
xx           3.74318    0.40006   9.357 3.31e-05 ***
xxsq        -0.30682    0.04279  -7.170 0.000182 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
```

The residual plot now



No problems any more.

Multiple regression

- What if more than one x? Extra issues:
  - Now one intercept and a slope for each x: how to interpret?
  - Which x-variables actually help to predict y? Different interpretations of "global" F-test and individual t-tests.
- In `lm` line, add extra xs after `~`.
- Interpretation not so easy (and other problems that can occur).

Multiple regression example

Study of women and visits to health professionals, and how the number of visits might be related to other variables:

- timedrs: number of visits to health professionals (over course of study)
- phyheal: number of physical health problems
- menheal: number of mental health problems
- stress: result of questionnaire about number and type of life changes

`timedrs` response, others explanatory.

The code

```
visits=read.table("regressx.txt",header=T)
head(visits)
attach(visits)
visits.1=lm(timedrs~phyheal+menheal+stress)
summary(visits.1)
```

Output part 1

```
Call:
lm(formula = timedrs ~ phyheal + menheal + stress)

Residuals:
    Min      1Q  Median      3Q     Max
-14.792  -4.353  -1.815   0.902  65.886

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.704848   1.124195  -3.296 0.001058 **
phyheal      1.786948   0.221074   8.083  5.6e-15 ***
menheal     -0.009666   0.129029  -0.075 0.940318
stress       0.013615   0.003612   3.769 0.000185 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.708 on 461 degrees of freedom
Multiple R-squared: 0.2188, Adjusted R-squared: 0.2137
F-statistic: 43.03 on 3 and 461 DF,  p-value: < 2.2e-16
```

The slopes

Model as a whole strongly significant even though R-sq not very big (lots of data). At least one of the x's predicts `timedrs`. (repeat output)

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.704848   1.124195  -3.296 0.001058 **
phyheal      1.786948   0.221074   8.083  5.6e-15 ***
menheal     -0.009666   0.129029  -0.075 0.940318
stress       0.013615   0.003612   3.769 0.000185 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The physical health and stress variables definitely help to predict the number of visits, but *with those in the model* we don't need `menheal`.
However, look at prediction of `timedrs` from `menheal` by itself:

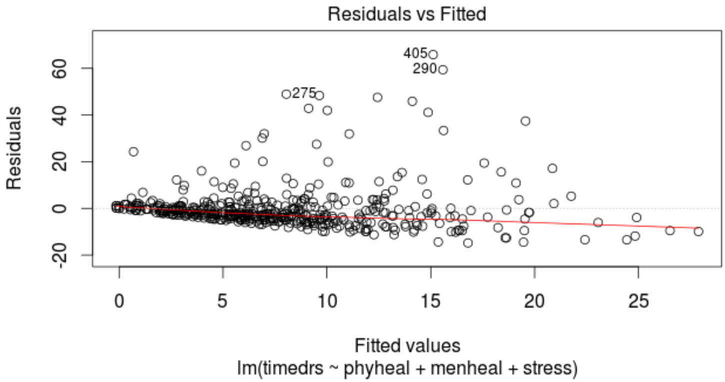
Just menheal

```
> visits.2=lm(timedrs~menheal)
> summary(visits.2)

Call:
lm(formula = timedrs ~ menheal)
<...>
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.8159     0.8702   4.385 1.44e-05 ***
menheal       0.6672     0.1173   5.688 2.28e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.6 on 463 degrees of freedom
Multiple R-squared: 0.06532, Adjusted R-squared: 0.0633
F-statistic: 32.35 on 1 and 463 DF,  p-value: 2.279e-08
```

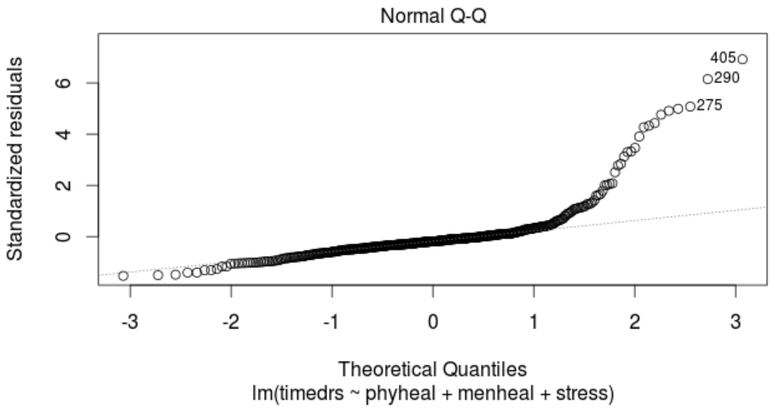
`menheal` by itself *does* significantly help to predict `timedrs`. But the R-sq is much less (6.5% vs. 22%) so the other two variables do a better job of prediction.

Residual plot

Go back to regression of `timedrs` on all x's: predicts significantly, but is it appropriate? Look at plot of residuals vs. predicted values.



Normal QQ plot of residuals



Residuals are not normal

- No pattern
- but some very positive residuals (compared to how negative).
- Distribution of residuals is *skewed*, not normal as it should be.

Fixing the problems

- Sometimes residuals are *very* positive: observed a *lot* larger than predicted.
- Try *transforming* response: use log or square root of response. (Note that response is *count*, often skewed to right.)
- Try regression again. Define transformed `timedrs` in data step, and use transformed variable as response. Check residual plot to see that it is OK now:

```
lgtime=log(timedrs+1)
visits.3=lm(lgtime~phyheal+menheal+stress)
plot(visits.3)
```

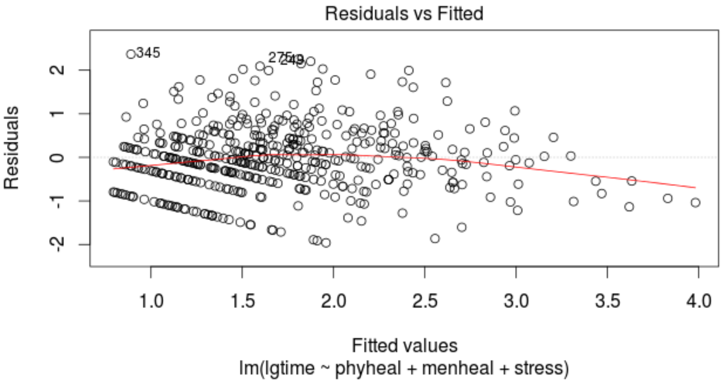
Output

```
> summary(visits.3)

Call:
lm(formula = lgtime ~ phyheal + menheal + stress)

Residuals:
     Min       1Q   Median       3Q      Max
-1.95865 -0.44076 -0.02331  0.42304  2.36797

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.3903862  0.0882908   4.422 1.22e-05 ***
phyheal     0.2019361  0.0173624  11.631  < 2e-16 ***
menheal     0.0071442  0.0101335   0.705    0.481
stress      0.0013158  0.0002837   4.638 4.58e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7625 on 461 degrees of freedom
Multiple R-squared: 0.3682, Adjusted R-squared: 0.3641
F-statistic: 89.56 on 3 and 461 DF,  p-value: < 2.2e-16
```

Comments

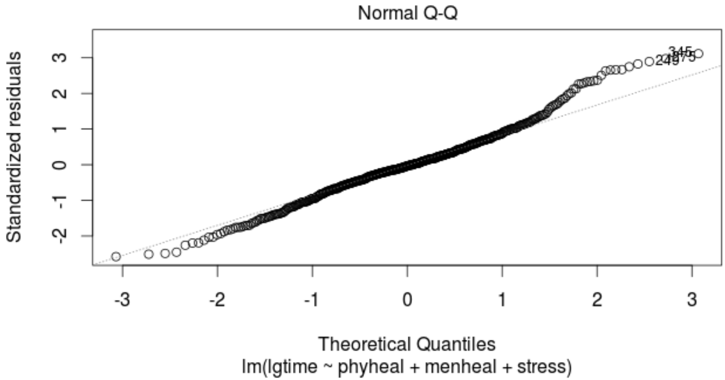
- Model as a whole strongly significant again
- R-sq higher than before (37% vs. 22%) suggesting things more linear now
- Same conclusion re `menheal`: can take out of regression.
- Should look at residual plot (next page).

The residual plot



Much better. Residuals range from 2 to -2, and look symmetric in shape. Should be trustworthy now.

Normal QQ plot of residuals



Box-Cox transformations

- Taking log of `timedrs` and having it work: lucky guess. How to find good transformation?
- Idea: Box-Cox: *estimate* the kind of transformation that would work: take power of response ($1 =$ no change, $0.5 =$ square root, $0 = log$).
- `boxcox` in package `MASS`.

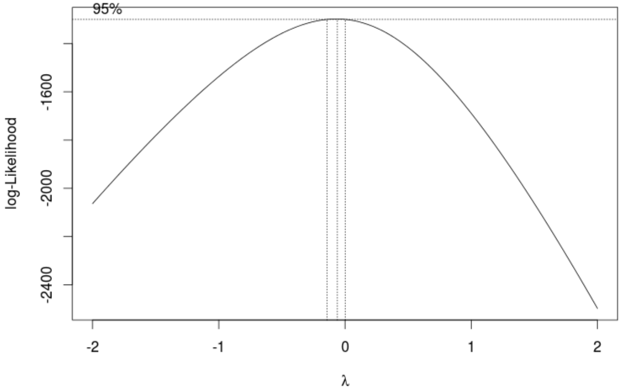
boxcox

- Some of `timedrs` values are 0, but Box-Cox expects all $+$. Define new variable `tp` in data step, then call `boxcox` with that as response.

  ```
  tp=timedrs+1
  library(MASS)
  boxcox(tp~phyheal+menheal+stress)
  ```

- `tp` only necessary here because of zeros in `timedrs`; normally omit and use original response in `boxcox`.
- Output from `boxcox` is plot, as on next page.

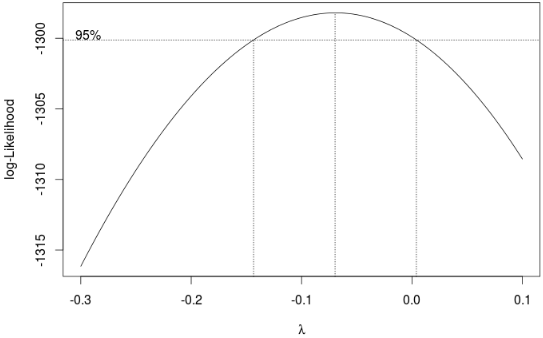
Try 1



- Best: λ just less than zero.
- Hard to see scale.
- Focus on λ in $(-0.3, 0.1)$.

Try 2

```
boxcox(tp~phyheal+menheal+stress,lambda=seq(-0.3,0.1,0.01))
```



- Best: λ just about -0.07.
- CI for λ about $(-0.14, 0.01)$.
- Only round number: $\lambda = 0$, log transformation.

Testing more than one x at once

The t-tests test only whether one variable could be taken out of the regression you're looking at. To test significance of more than one variable at once, fit model with and without variables and use anova to compare fit of models:

```
> visits.5=lm(lgtime~phyheal+menheal+stress)
> visits.6=lm(lgtime~stress)
> anova(visits.6,visits.5)
Analysis of Variance Table

Model 1: lgtime ~ stress
Model 2: lgtime ~ phyheal + menheal + stress
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    463 371.47
2    461 268.01  2    103.46 88.984 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Results of tests

- Test says "taking both variables out makes the fit worse, so don't do it".
- There is significant difference between the two models, meaning the model with *more* x's fits better. Taking out those x's is a mistake. Or putting them in is a good idea.

The punting data

Data set punting.dat contains 4 variables for 13 right-footed football kickers (punters): left leg and right leg strength (lbs), distance punted (ft), another variable called "fred". Predict punting distance from other variables:

```
punting=read.table("punting.txt",header=T)
attach(punting)
punting.1=lm(punt~left+right+fred)
summary(punting.1)
```

Regression output (edited)

```
> punting.1=lm(punt~left+right+fred)
> summary(punting.1)

Call:
lm(formula = punt ~ left + right + fred)
<...>
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.6855    29.1172  -0.161    0.876
left          0.2679     2.1111   0.127    0.902
right         1.0524     2.1477   0.490    0.636
fred         -0.2672     4.2266  -0.063    0.951

Residual standard error: 14.68 on 9 degrees of freedom
Multiple R-squared: 0.7781, Adjusted R-squared: 0.7042
F-statistic: 10.52 on 3 and 9 DF,  p-value: 0.00267
```

Comments

- Overall regression strongly significant, R-sq high.
- None of the x's significant! Why?
- t-tests only say that you could take any one of the x's out without damaging the fit; doesn't matter which one.
- Explanation: look at *correlations*.

The correlations

```
> cor(punting)
          left     right      punt      fred
left  1.0000000 0.8957224 0.8117368 0.9722632
right 0.8957224 1.0000000 0.8805469 0.9728784
punt  0.8117368 0.8805469 1.0000000 0.8679507
fred  0.9722632 0.9728784 0.8679507 1.0000000
```

All correlations are high: x's with `punt` (good) and with each other (bad, at least confusing).

What to do? Probably do just as well to pick one variable, say `right` since kickers are right-footed.

Just right

```
> punting.2=lm(punt~right)
> anova(punting.2,punting.1)
Analysis of Variance Table

Model 1: punt ~ right
Model 2: punt ~ left + right + fred
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     11 1962.5
2      9 1938.2  2    24.263 0.0563 0.9456
```

No significant loss by dropping other two variables. Compare R-squareds for the two models:

```
> summary(punting.1)$r.squared
[1] 0.7781401
> summary(punting.2)$r.squared
[1] 0.7753629
```

Basically no difference.

Regression results

```
> summary(punting.2)

Call:
lm(formula = punt ~ right)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.7576 -11.0611   0.3656   7.8890  19.0423 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.6930    25.2649  -0.146    0.886    
right         1.0427     0.1692   6.162 7.09e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.36 on 11 degrees of freedom
Multiple R-squared: 0.7754, Adjusted R-squared: 0.7549 
F-statistic: 37.97 on 1 and 11 DF,  p-value: 7.088e-05
```