

## Series 3 - solutions

### The fruitflies dataset

This dataset contains observations on five groups of male fruitflies – 25 fruitflies in each group – from an experiment designed to test if increased reproduction reduces longevity for male fruitflies. The five groups are: males forced to live alone, males assigned to live with one or eight interested females, and males assigned to live with one or eight non-receptive females.

```
#detach(data)

url <- "https://ww2.amstat.org/publications/jse/datasets/fruitfly.dat.txt"
data <- read.table(url)
data <- data[,c(-1,-6)] # remove id and sleep
names(data) <- c("partners", "type", "longevity", "thorax")
attach(data)

head(data)

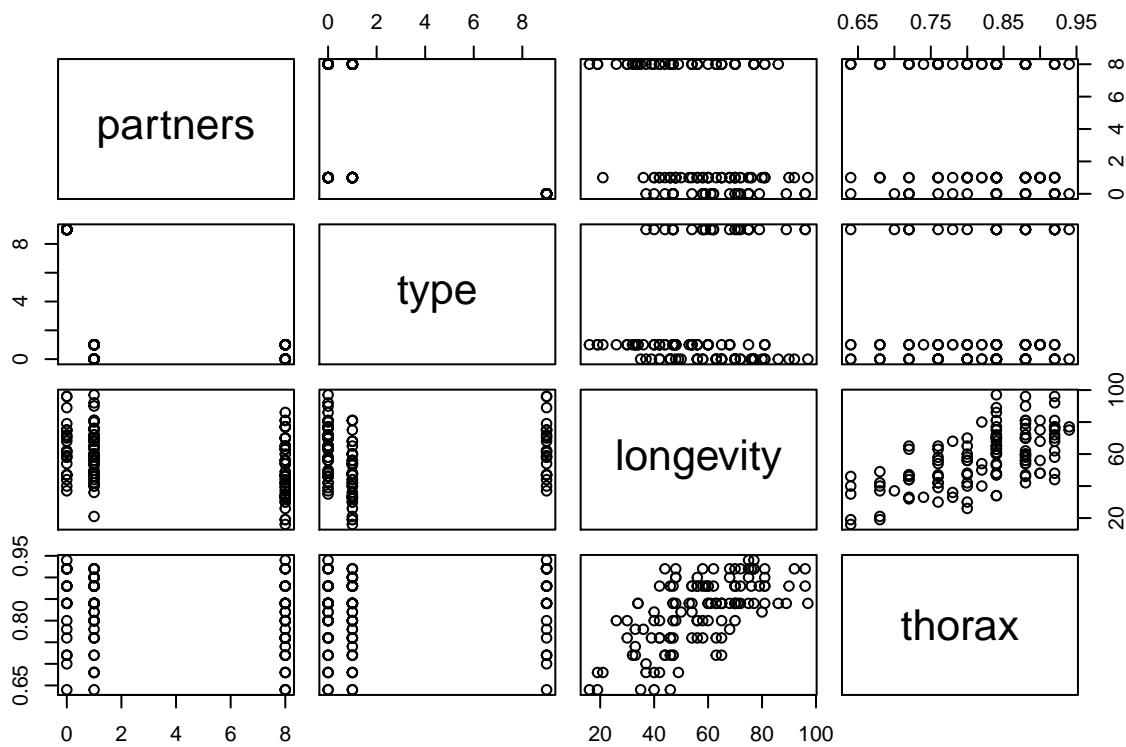
##   partners type longevity thorax
## 1       8    0      35    0.64
## 2       8    0      37    0.68
## 3       8    0      49    0.68
## 4       8    0      46    0.72
## 5       8    0      63    0.72
## 6       8    0      39    0.76

summary(data)

##      partners        type     longevity      thorax
##  Min.   :0.0   Min.   :0.0   Min.   :16.00   Min.   :0.640
##  1st Qu.:1.0  1st Qu.:0.0  1st Qu.:46.00  1st Qu.:0.760
##  Median :1.0  Median :1.0  Median :58.00  Median :0.840
##  Mean   :3.6  Mean   :2.2  Mean   :57.44  Mean   :0.821
##  3rd Qu.:8.0  3rd Qu.:1.0  3rd Qu.:70.00  3rd Qu.:0.880
##  Max.   :8.0  Max.   :9.0  Max.   :97.00  Max.   :0.940

dim(data)

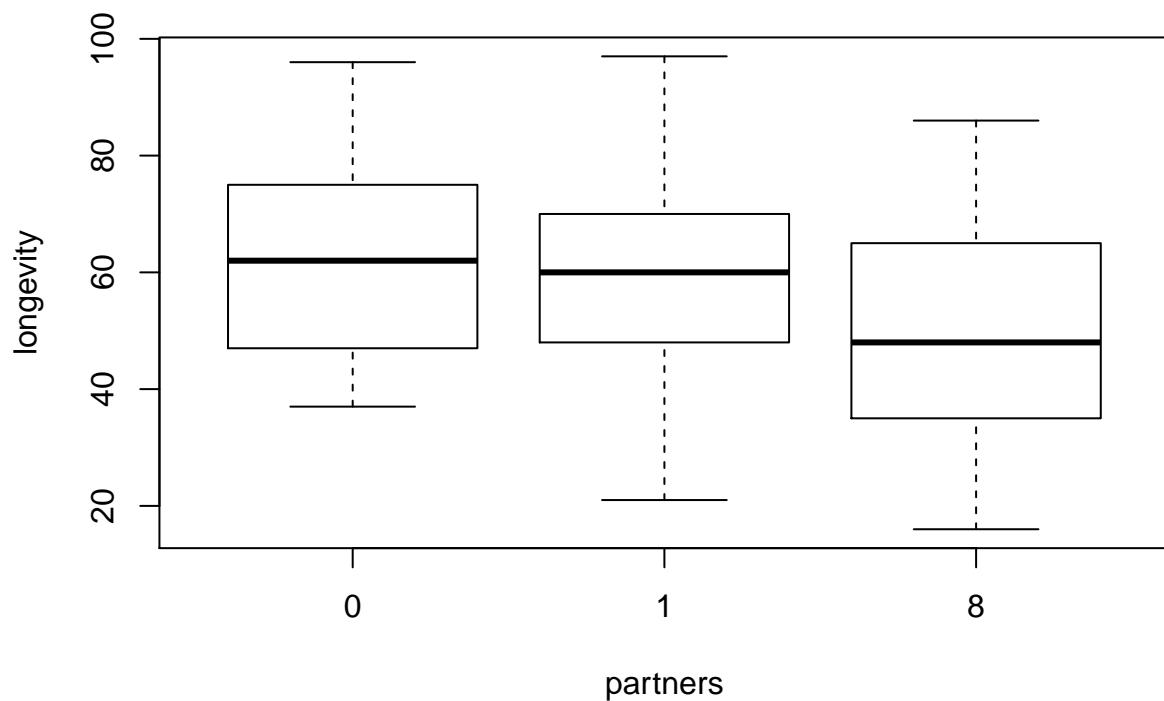
## [1] 125   4
# let's get a visual understanding of this data
pairs(data)
```



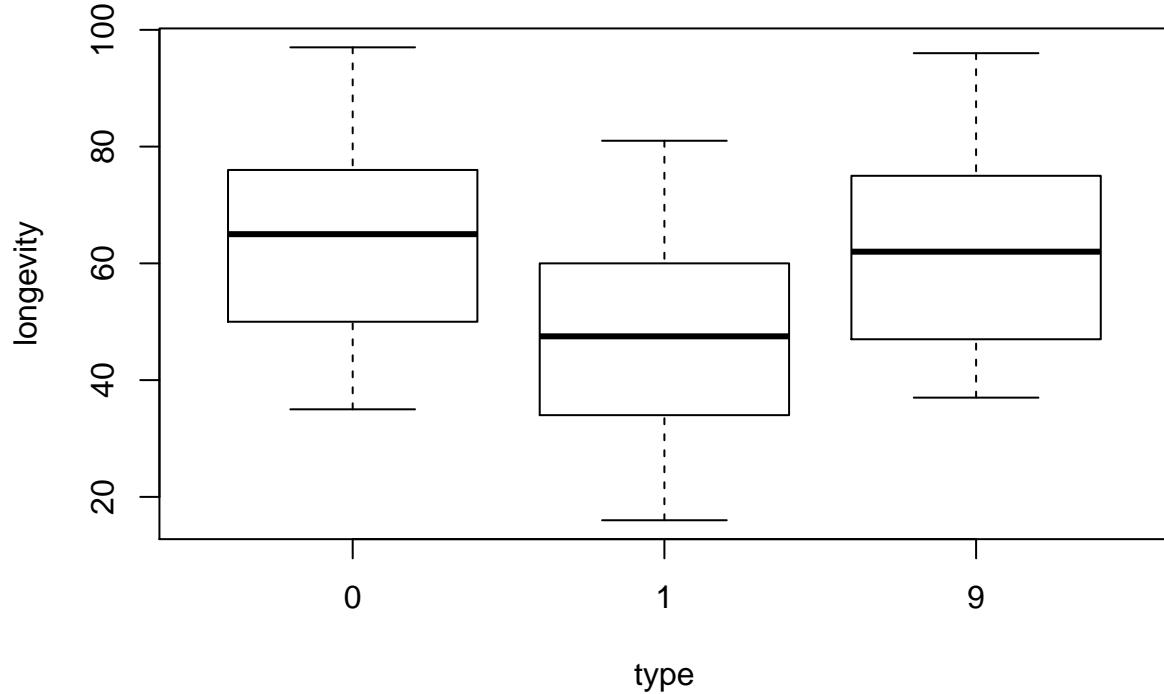
```
cor(data)
```

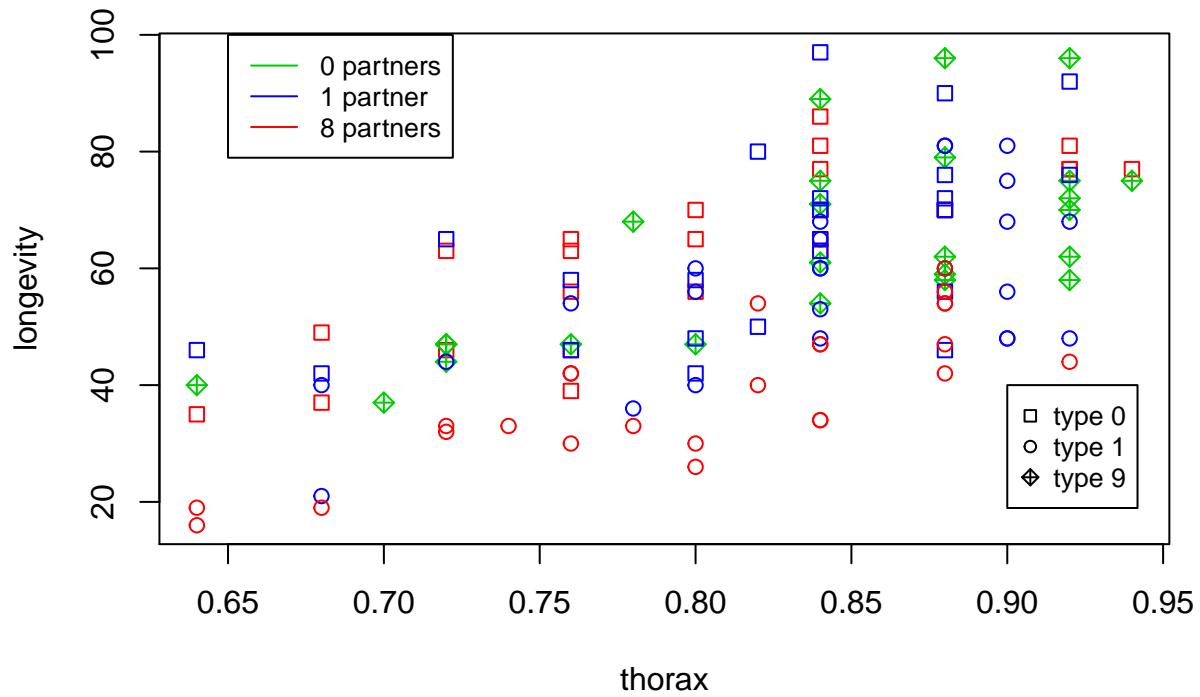
```
##           partners      type  longevity      thorax
## partners  1.0000000 -0.49420708 -0.3030521 -0.19332920
## type     -0.4942071  1.00000000  0.1189528  0.09906777
## longevity -0.3030521  0.11895277  1.0000000  0.63648353
## thorax   -0.1933292  0.09906777  0.6364835  1.00000000
```

```
boxplot(longevity~partners)
```



```
boxplot(longevity~type)
```

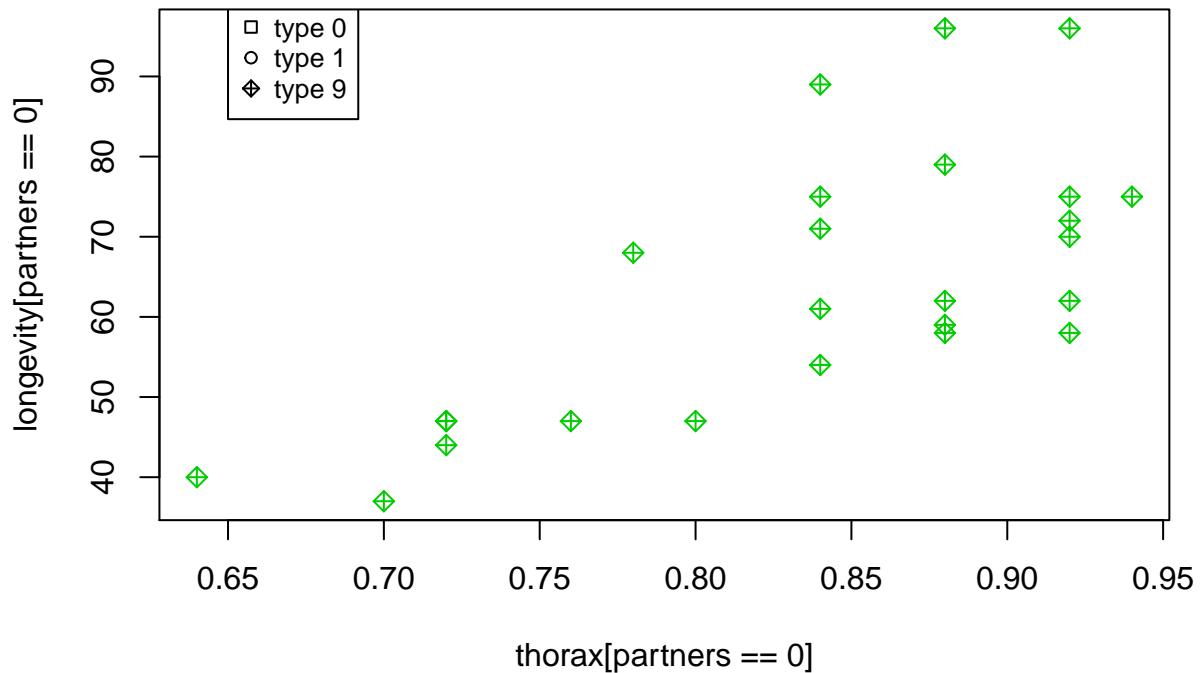




Let's separate the points based on the number of partners available.

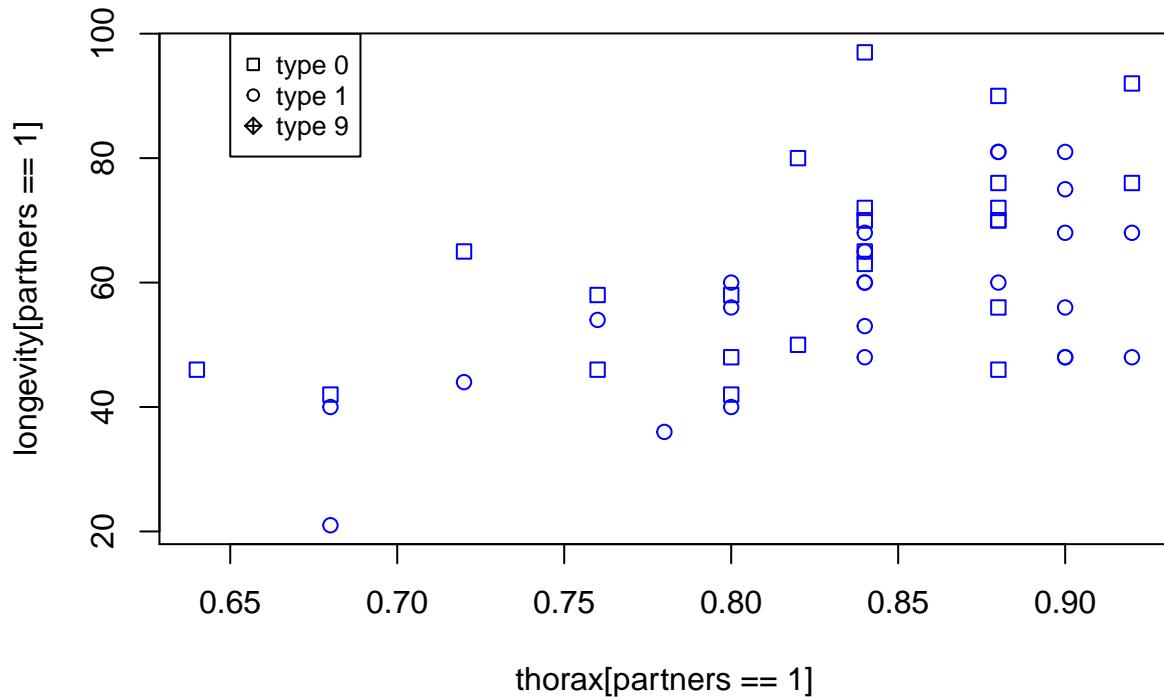
```
plot(thorax[partners==0], longevity[partners==0], col=3, pch=type[partners==0], main = "0 partners")
legend(0.65, 100,
       legend=c("type 0", "type 1", "type 9"),
       col=1,
       cex=.8,
       pch=c(0,1,9))
```

## 0 partners



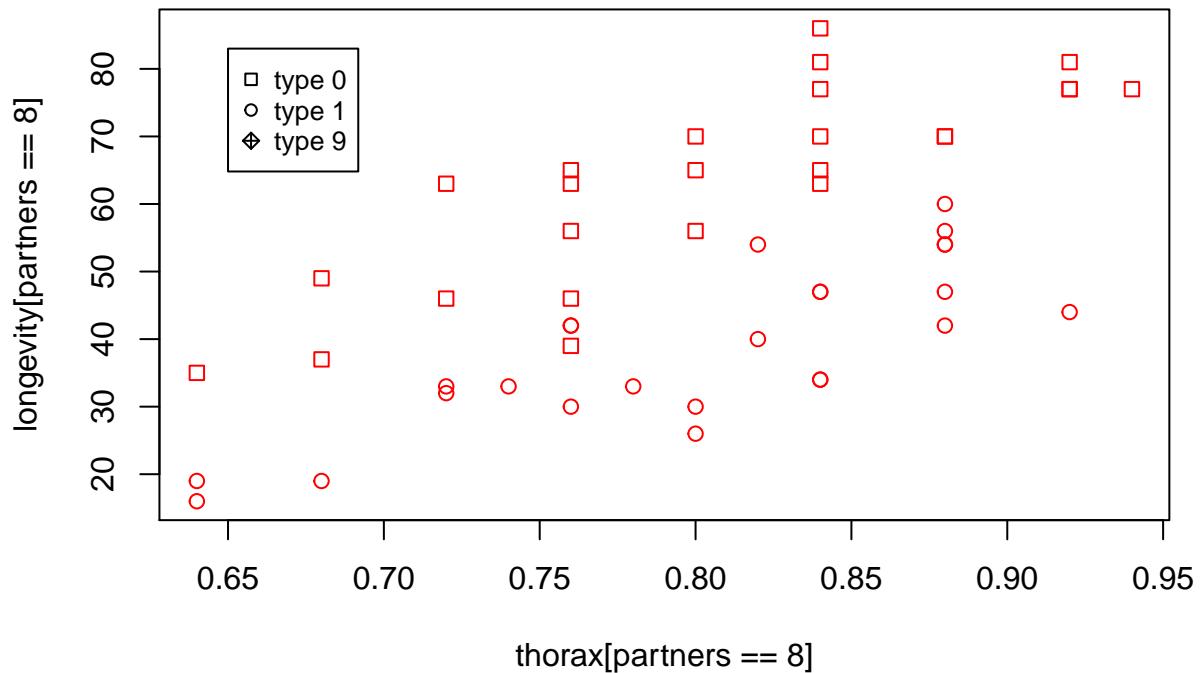
```
plot(thorax[partners==1], longevity[partners==1], col=4, pch=type[partners==1], main = "1 partner")
legend(0.65, 100,
       legend=c("type 0", "type 1", "type 9"),
       col=1,
       cex=.8,
       pch=c(0,1,9))
```

### 1 partner



```
plot(thorax[partners==8], longevity[partners==8], col=2, pch=type[partners==8], main = "8 partners")
legend(0.65, 83,
       legend=c("type 0", "type 1", "type 9"),
       col=1,
       cex=.8,
       pch=c(0,1,9))
```

## 8 partners

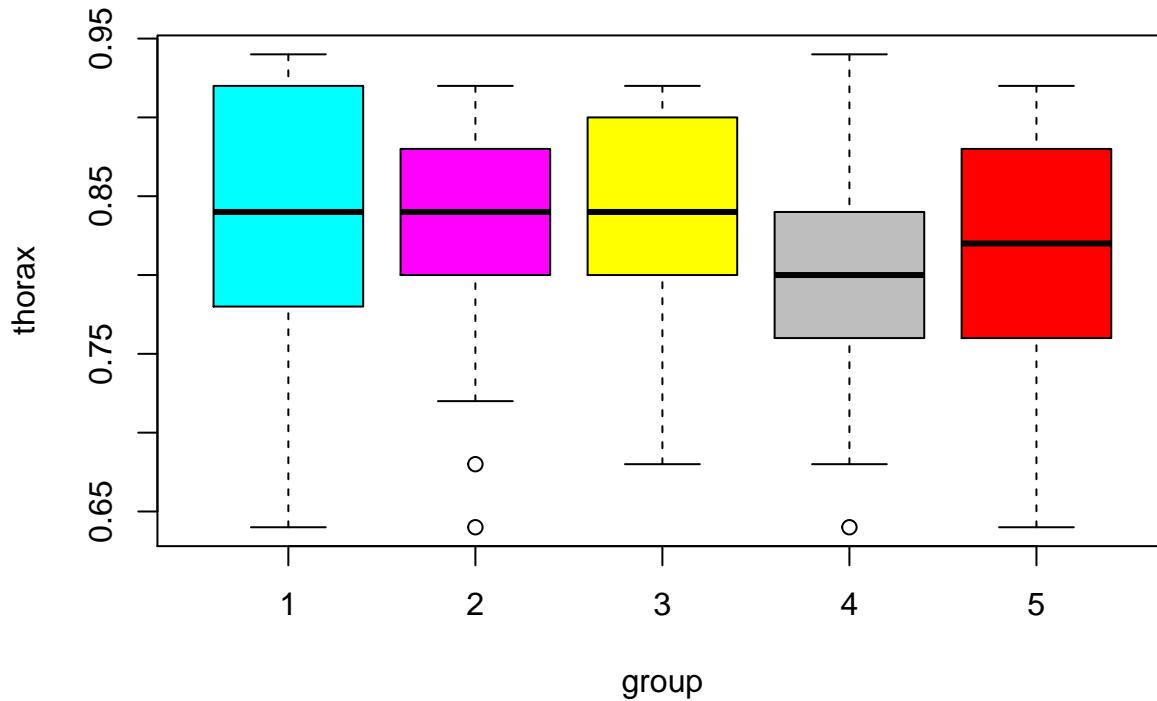


Looking at the second and third plot it emerges an interaction between the number of partners and the type of female (interested , ... ) on longevity. We now want to encode these 5 different study groups with dummy variables.

```
group1 <- (partners==0) * 1
group2 <- (partners==1 & type==0) *2
group3 <- (partners==1 & type==1) *3
group4 <- (partners==8 & type==0) *4
group5 <- (partners==8 & type==1) *5
group <- group1 + group2 + group3 + group4 + group5
```

Let's look at the thorax length among these 5 different groups.

```
boxplot(thorax~group, col=c(5,6,7,8,2))
```



Is there a statistically significant difference in the thorax length among the groups? Let's use an ANOVA to test it.

```
fit1 <- lm(thorax~1)
fit2 <- lm(thorax~group)
anova(fit1, fit2)

## Analysis of Variance Table
##
## Model 1: thorax ~ 1
## Model 2: thorax ~ group
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1     124 0.74388
## 2     123 0.72272  1  0.02116 3.6012 0.06008 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By looking at the above table we can conclude that there's no statistically significant difference in terms of thorax length between the 5 groups. This was to be expected since the assignments to the groups were random, hence the distribution of thorax should be similar among the different groups.

But can we omit thorax from the model then? Probably not, because thorax length could be a fundamental indicator of the health of the animal, which is in the end positively correlated with longevity. But let's test it!

```
model1 <- lm(longevity~factor(group))
model2 <- lm(longevity~factor(group)+thorax)
anova(model1,model2)

## Analysis of Variance Table
```

```

## 
## Model 1: longevity ~ factor(group)
## Model 2: longevity ~ factor(group) + thorax
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1     120 26314
## 2     119 13145  1     13169 119.22 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Let's test now the effect of the presence of thorax over a specific test group.

```

model1 <- lm(longevity[partners==1] ~ factor(group[partners==1]))
model2 <- lm(longevity[partners==1] ~ factor(group[partners==1]) + thorax[partners==1])
anova(model1, model2)

```

```

## Analysis of Variance Table

## 
## Model 1: longevity[partners == 1] ~ factor(group[partners == 1])
## Model 2: longevity[partners == 1] ~ factor(group[partners == 1]) + thorax[partners ==
##           1]
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1     48 11228.6
## 2     47  6962.9  1     4265.7 28.793 2.417e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Again, the presence of the variable thorax drastically reduces the RSS. Let's look at the difference in terms of coefficients.

```

summary(model1)

## 
## Call:
## lm(formula = longevity[partners == 1] ~ factor(group[partners ==
##           1]))
## 
## Residuals:
##   Min     1Q Median     3Q    Max 
## -35.76  -8.79   0.20  10.46  32.20 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 64.800     3.059   21.184 <2e-16 ***
## factor(group[partners == 1])3 -8.040     4.326  -1.859   0.0692 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 15.29 on 48 degrees of freedom
## Multiple R-squared:  0.06713,   Adjusted R-squared:  0.0477 
## F-statistic: 3.454 on 1 and 48 DF,  p-value: 0.06923

```

```

summary(model2)

## 
## Call:
## lm(formula = longevity[partners == 1] ~ factor(group[partners ==
##           1]) + thorax[partners == 1])

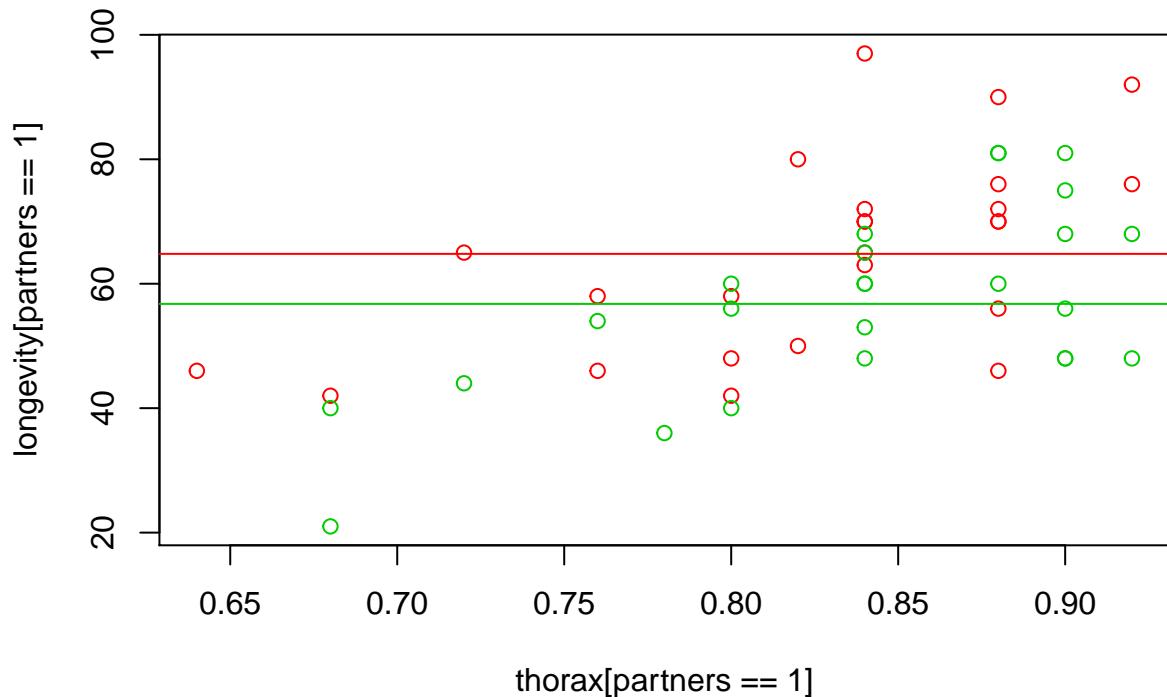
```

```

## 
## Residuals:
##   Min     1Q Median     3Q    Max 
## -26.103 -9.123  1.092  7.273 30.267 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 -46.038    20.799  -2.214  0.03175 *  
## factor(group[partners == 1])3   -9.651     3.456  -2.793  0.00753 ** 
## thorax[partners == 1]            134.252    25.019   5.366 2.42e-06 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 12.17 on 47 degrees of freedom 
## Multiple R-squared:  0.4215, Adjusted R-squared:  0.3969 
## F-statistic: 17.12 on 2 and 47 DF,  p-value: 2.593e-06 

plot(thorax[partners==1], longevity[partners==1], col=group[partners==1])
abline(a=model1$coefficients[1], b=0, col=2)
abline(a=model1$coefficients[1]+model1$coefficients[2], b=0, col=3)

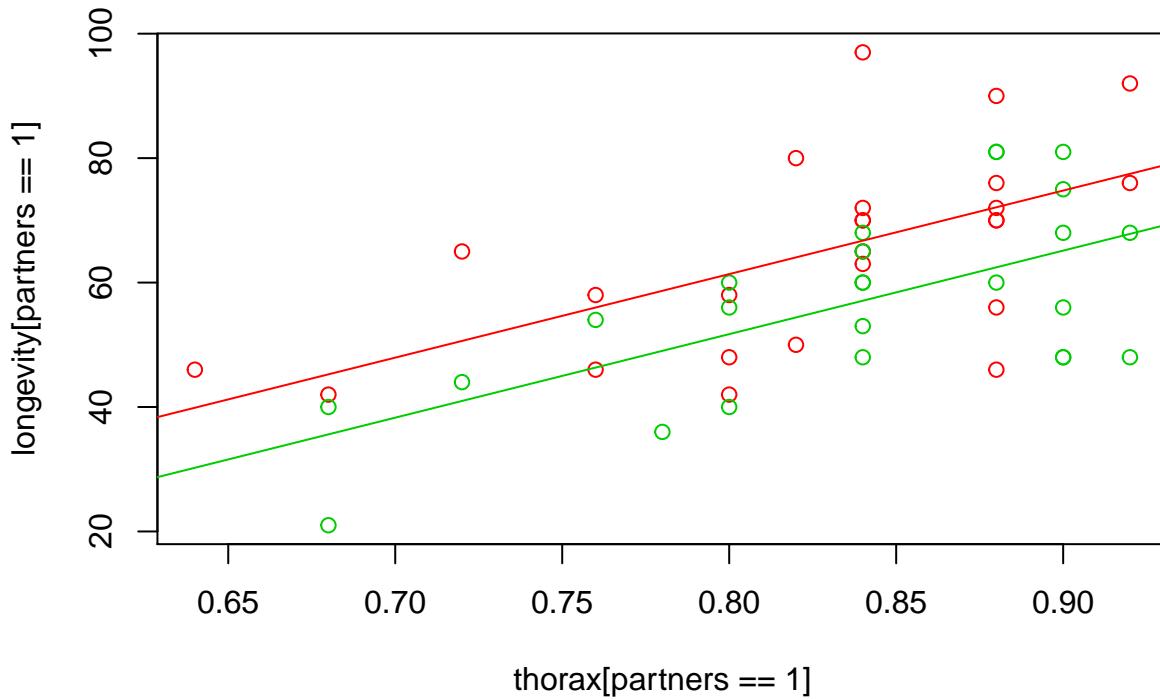
```



```

plot(thorax[partners==1], longevity[partners==1], col=group[partners==1])
abline(a=model2$coefficients[1], b=model2$coefficients[3], col=2)
abline(a=model2$coefficients[1]+model2$coefficients[2], b=model2$coefficients[3], col=3)

```



Now we want to test for interaction between type of female and number of females.

```
wrong.model <- lm(longevity ~ thorax + as.factor(type) * as.factor(partners))
summary(wrong.model)
```

```
##
## Call:
## lm(formula = longevity ~ thorax + as.factor(type) * as.factor(partners))
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -26.189  -6.599  -0.989   6.408  30.244 
##
## Coefficients: (4 not defined because of singularities)
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 -46.055   10.239  -4.498 1.61e-05 ***
## thorax                      135.819   12.439  10.919 < 2e-16 ***
## as.factor(type)1              -23.879   2.973  -8.031 7.83e-13 ***
## as.factor(type)9                -3.929   2.997  -1.311 0.192347  
## as.factor(partners)1            -1.276   2.983  -0.428 0.669517  
## as.factor(partners)8                  NA        NA        NA        NA  
## as.factor(type)1:as.factor(partners)1  14.210   4.210   3.375 0.000996 ***
## as.factor(type)9:as.factor(partners)1  NA        NA        NA        NA  
## as.factor(type)1:as.factor(partners)8  NA        NA        NA        NA  
## as.factor(type)9:as.factor(partners)8  NA        NA        NA        NA  
## ---                                 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

## 
## Residual standard error: 10.51 on 119 degrees of freedom
## Multiple R-squared:  0.6564, Adjusted R-squared:  0.6419
## F-statistic: 45.46 on 5 and 119 DF,  p-value: < 2.2e-16

Note: the above model doesn't make sense since we should only account for 5 of the possible combination of dummy variables. Now let's create a better one.

better.model <- lm(longevity~thorax+as.factor(group2)+as.factor(group3)+as.factor(group4)+as.factor(group5))
summary(better.model)

## 
## Call:
## lm(formula = longevity ~ thorax + as.factor(group2) + as.factor(group3) +
##     as.factor(group4) + as.factor(group5))
## 
## Residuals:
##      Min      1Q  Median      3Q      Max 
## -26.189   -6.599  -0.989   6.408  30.244 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -49.984    10.609  -4.711 6.73e-06 ***
## thorax       135.819    12.439  10.919 < 2e-16 ***
## as.factor(group2) 2.653     2.975   0.891   0.3745  
## as.factor(group3) -7.017     2.973  -2.361   0.0199 *  
## as.factor(group4)  3.929     2.997   1.311   0.1923  
## as.factor(group5) -19.951    3.006  -6.636  1.00e-09 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 10.51 on 119 degrees of freedom
## Multiple R-squared:  0.6564, Adjusted R-squared:  0.6419
## F-statistic: 45.46 on 5 and 119 DF,  p-value: < 2.2e-16

```

Is the interaction between type and partners statistically significant? Let's test it with an ANOVA.

```

group1 <- (partners==0) * 1
group2 <- (partners==1 & type==0) *1
group3 <- (partners==1 & type==1) *1
group4 <- (partners==8 & type==0) *1
group5 <- (partners==8 & type==1) *1
reduced.model <- lm(longevity~thorax+(I(group2+group3))+(I(group2+group4))+(I(group5-group2)))
summary(reduced.model)

## 
## Call:
## lm(formula = longevity ~ thorax + (I(group2 + group3)) + (I(group2 +
##     group4)) + (I(group5 - group2)))
## 
## Residuals:
##      Min      1Q  Median      3Q      Max 
## -29.8501  -6.7025  -0.5518   6.6970  26.6700 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -49.984    10.609  -4.711 6.73e-06 ***
## thorax       135.819    12.439  10.919 < 2e-16 ***
## I(group2 + group3) 2.653     2.975   0.891   0.3745  
## I(group2 + group4) -7.017     2.973  -2.361   0.0199 *  
## I(group5 - group2)  3.929     2.997   1.311   0.1923  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

## (Intercept)      -51.8097    11.0449  -4.691 7.27e-06 ***
## thorax          138.0021   12.9490   10.657 < 2e-16 ***
## I(group2 + group3) -10.5636   2.8988  -3.644 0.000398 ***
## I(group2 + group4)   0.4525   2.9334   0.154 0.877674
## I(group5 - group2) -16.3291   2.9273  -5.578 1.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.96 on 120 degrees of freedom
## Multiple R-squared:  0.6235, Adjusted R-squared:  0.6109
## F-statistic: 49.68 on 4 and 120 DF,  p-value: < 2.2e-16
anova(reduced.model, better.model)

```

```

## Analysis of Variance Table
##
## Model 1: longevity ~ thorax + (I(group2 + group3)) + (I(group2 + group4)) +
##           (I(group5 - group2))
## Model 2: longevity ~ thorax + as.factor(group2) + as.factor(group3) +
##           as.factor(group4) + as.factor(group5)
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1     120 14403
## 2     119 13145  1    1258.5 11.394 0.0009957 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

From the anova we can conclude there's statistically significant interaction between the variables.

```

group <- as.factor(group)
full.model <- lm(longevity~thorax+group+thorax*group)
summary(full.model)

```

```

##
## Call:
## lm(formula = longevity ~ thorax + group + thorax * group)
##
## Residuals:
##       Min     1Q     Median      3Q     Max 
## -25.9509  -6.5324  -0.7693   6.3792  30.3071 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -50.2420   21.7221  -2.313   0.0225 *  
## thorax       136.1268   25.8576   5.264 6.61e-07 *** 
## group2        6.5172   33.7479   0.193   0.8472    
## group3       -7.7501   33.8457  -0.229   0.8193    
## group4       -5.4574   30.6537  -0.178   0.8590    
## group5      -11.0380   31.1731  -0.354   0.7239    
## thorax:group2 -4.6771   40.5042  -0.115   0.9083    
## thorax:group3  0.8743   40.2786   0.022   0.9827    
## thorax:group4 11.6629   37.1806   0.314   0.7543    
## thorax:group5 -11.1268   37.9816  -0.293   0.7701    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.67 on 115 degrees of freedom

```

```

## Multiple R-squared:  0.6575, Adjusted R-squared:  0.6307
## F-statistic: 24.53 on 9 and 115 DF,  p-value: < 2.2e-16
anova(better.model, full.model)

## Analysis of Variance Table
##
## Model 1: longevity ~ thorax + as.factor(group2) + as.factor(group3) +
##           as.factor(group4) + as.factor(group5)
## Model 2: longevity ~ thorax + group + thorax * group
##   Res.Df   RSS Df Sum of Sq    F Pr(>F)
## 1     119 13145
## 2     115 13102  4    42.523 0.0933 0.9844

```

## The life expectancy dataset

```

url <- "https://raw.githubusercontent.com/jawj/coffeestats/master/lifeexp.dat"
data <- read.table(url, sep="\t", header=T, row.names=1)
data <- data[,c("LifeExp", "People.per.TV", "People.per.Dr")]

```

```

detach(data)
attach(data)
head(data)

```

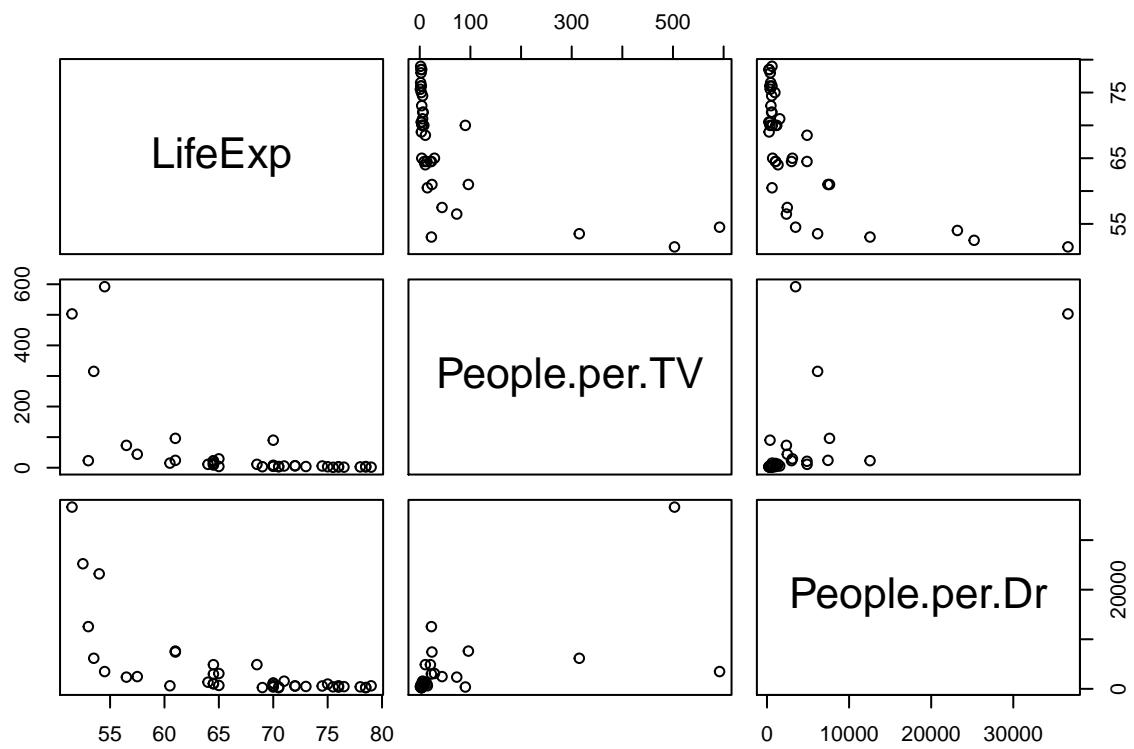
	LifeExp	People.per.TV	People.per.Dr
## Argentina	70.5	4.0	370
## Bangladesh	53.5	315.0	6166
## Brazil	65.0	4.0	684
## Canada	76.5	1.7	449
## China	70.0	8.0	643
## Colombia	71.0	5.6	1551

```
dim(data)
```

```
## [1] 40 3
```

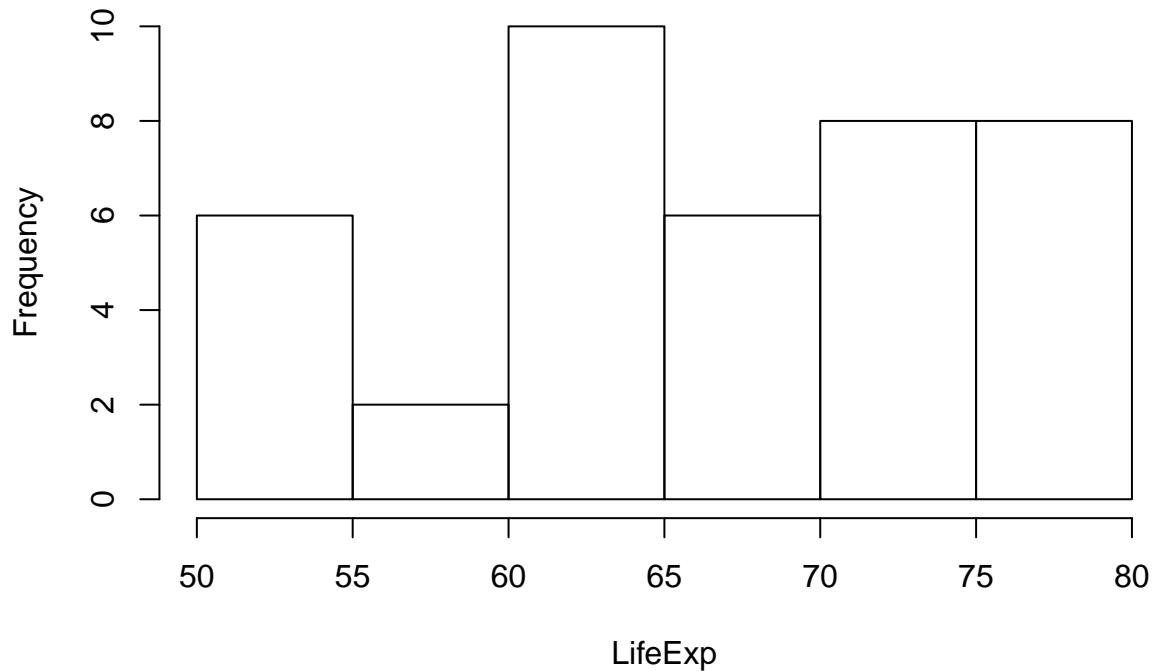
Let's have a look at the data!

```
pairs(data)
```



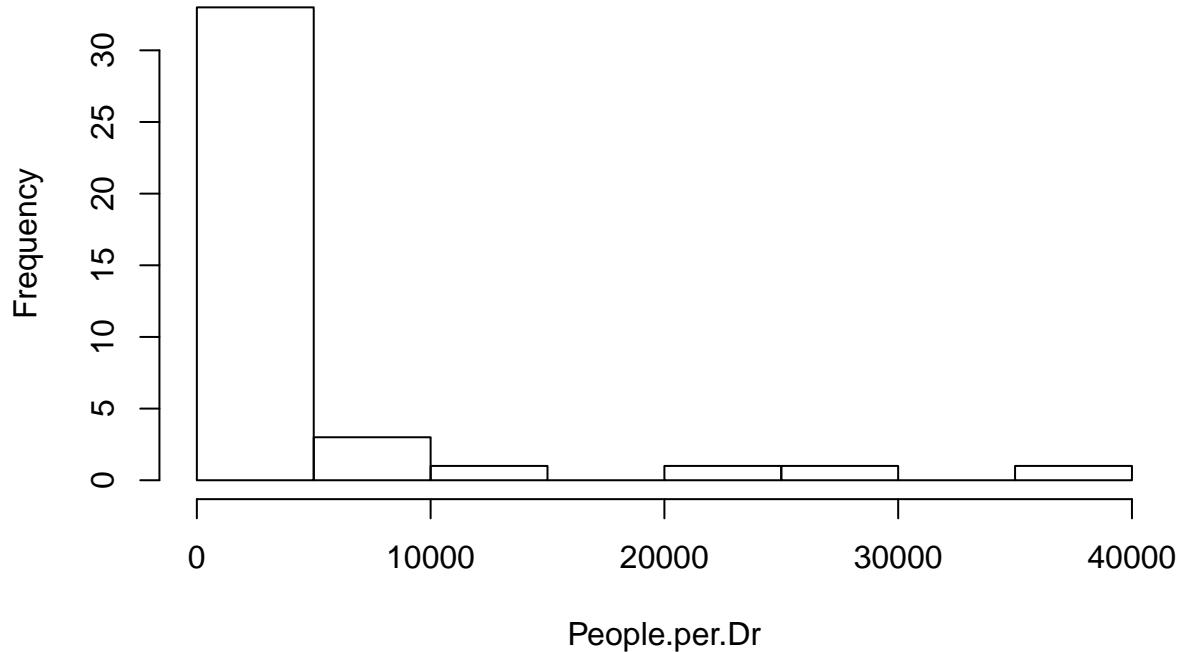
```
hist(LifeExp)
```

### Histogram of LifeExp



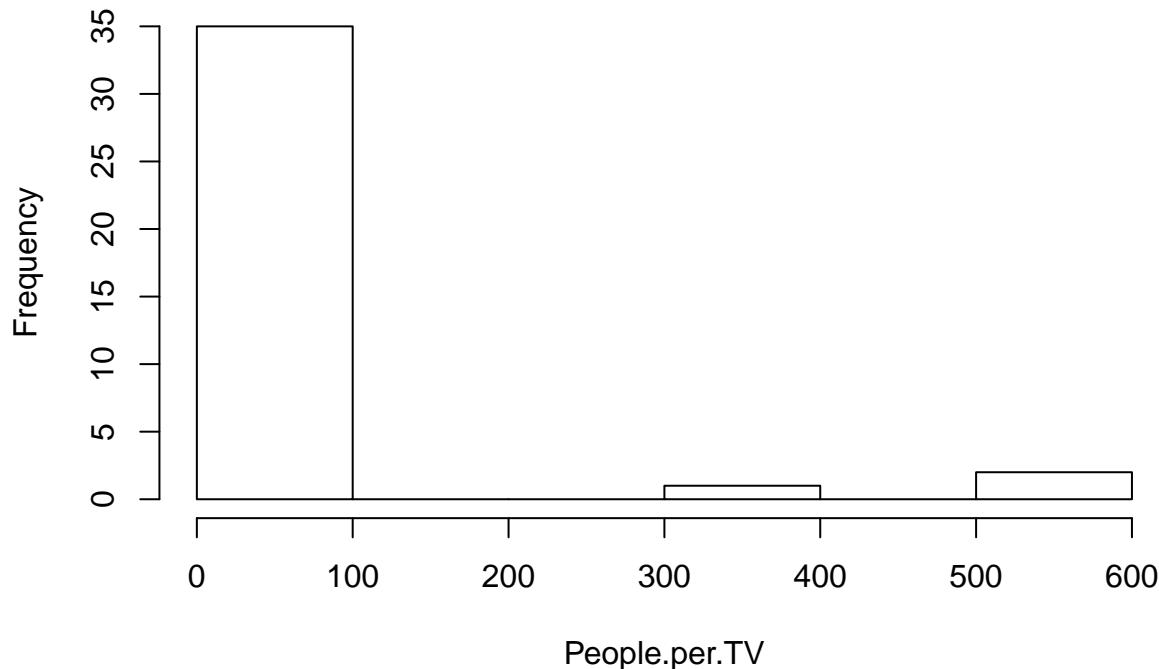
```
hist(People.per.Dr)
```

### Histogram of People.per.Dr



```
hist(People.per.TV)
```

## Histogram of People.per.TV



```
# States with highest life expectancy  
data[order(LifeExp, decreasing = TRUE),]
```

```
##          LifeExp People.per.TV People.per.Dr  
## Japan      79.0      1.8        609  
## Italy      78.5      3.8        233  
## Spain      78.5      2.6        275  
## France     78.0      2.6        403  
## Canada     76.5      1.7        449  
## Germany    76.0      2.6        346  
## UK         76.0      3.0        611  
## USA         75.5      1.3        404  
## Taiwan      75.0      3.2        965  
## Venezuela   74.5      5.6        576  
## Poland      73.0      3.9        480  
## Mexico      72.0      6.6        600  
## Romania     72.0      6.0        559  
## Colombia    71.0      5.6       1551  
## Argentina   70.5      4.0        370  
## Ukraine     70.5      3.0        226  
## China        70.0      8.0        643  
## Korea.North 70.0     90.0        370  
## Korea.South 70.0      4.9       1066  
## Turkey       70.0      5.0       1189  
## Russia       69.0      3.2        259  
## Thailand     68.5     11.0       4883
```

```

## Brazil      65.0      4.0      684
## Vietnam    65.0     29.0     3096
## Iran       64.5     23.0     2992
## Morocco    64.5     21.0     4873
## Peru        64.5     14.0     1016
## Philippines 64.5      8.8     1062
## South.Africa 64.0     11.0     1340
## Indonesia   61.0     24.0     7427
## Kenya       61.0     96.0     7615
## Egypt       60.5     15.0      616
## India        57.5     44.0     2471
## Pakistan    56.5     73.0     2364
## Burma       54.5     592.0    3485
## Zaire       54.0      NA      23193
## Bangladesh  53.5     315.0    6166
## Sudan        53.0     23.0     12550
## Tanzania    52.5      NA      25229
## Ethiopia    51.5     503.0    36660

```

*# States with highest PeopleTV*

```
data[order(People.per.TV, decreasing = TRUE),]
```

	LifeExp	People.per.TV	People.per.Dr
## Burma	54.5	592.0	3485
## Ethiopia	51.5	503.0	36660
## Bangladesh	53.5	315.0	6166
## Kenya	61.0	96.0	7615
## Korea.North	70.0	90.0	370
## Pakistan	56.5	73.0	2364
## India	57.5	44.0	2471
## Vietnam	65.0	29.0	3096
## Indonesia	61.0	24.0	7427
## Iran	64.5	23.0	2992
## Sudan	53.0	23.0	12550
## Morocco	64.5	21.0	4873
## Egypt	60.5	15.0	616
## Peru	64.5	14.0	1016
## South.Africa	64.0	11.0	1340
## Thailand	68.5	11.0	4883
## Philippines	64.5	8.8	1062
## China	70.0	8.0	643
## Mexico	72.0	6.6	600
## Romania	72.0	6.0	559
## Colombia	71.0	5.6	1551
## Venezuela	74.5	5.6	576
## Turkey	70.0	5.0	1189
## Korea.South	70.0	4.9	1066
## Argentina	70.5	4.0	370
## Brazil	65.0	4.0	684
## Poland	73.0	3.9	480
## Italy	78.5	3.8	233
## Russia	69.0	3.2	259
## Taiwan	75.0	3.2	965
## Ukraine	70.5	3.0	226
## UK	76.0	3.0	611

```

## France      78.0      2.6      403
## Germany    76.0      2.6      346
## Spain       78.5      2.6      275
## Japan       79.0      1.8      609
## Canada     76.5      1.7      449
## USA         75.5      1.3      404
## Tanzania   52.5      NA      25229
## Zaire       54.0      NA      23193

# States with highest PeoplexDr
data[order(People.per.Dr, decreasing = TRUE),]

##          LifeExp People.per.TV People.per.Dr
## Ethiopia      51.5      503.0      36660
## Tanzania     52.5        NA      25229
## Zaire         54.0        NA      23193
## Sudan         53.0      23.0      12550
## Kenya          61.0      96.0      7615
## Indonesia     61.0      24.0      7427
## Bangladesh    53.5      315.0      6166
## Thailand       68.5      11.0      4883
## Morocco        64.5      21.0      4873
## Burma          54.5      592.0      3485
## Vietnam        65.0      29.0      3096
## Iran           64.5      23.0      2992
## India          57.5      44.0      2471
## Pakistan       56.5      73.0      2364
## Colombia       71.0      5.6      1551
## South.Africa   64.0      11.0      1340
## Turkey         70.0      5.0      1189
## Korea.South    70.0      4.9      1066
## Philippines    64.5      8.8      1062
## Peru            64.5      14.0      1016
## Taiwan          75.0      3.2      965
## Brazil          65.0      4.0      684
## China           70.0      8.0      643
## Egypt           60.5      15.0      616
## UK              76.0      3.0      611
## Japan           79.0      1.8      609
## Mexico          72.0      6.6      600
## Venezuela       74.5      5.6      576
## Romania         72.0      6.0      559
## Poland          73.0      3.9      480
## Canada          76.5      1.7      449
## USA             75.5      1.3      404
## France          78.0      2.6      403
## Argentina       70.5      4.0      370
## Korea.North     70.0      90.0      370
## Germany         76.0      2.6      346
## Spain           78.5      2.6      275
## Russia          69.0      3.2      259
## Italy            78.5      3.8      233
## Ukraine         70.5      3.0      226

```

Now we'll get rid of the missing values by simply deleting the corresponsing entries in the dataframe.

```

data <- na.omit(data)
dim(data)

## [1] 38 3

Let's fit a linear model on the logged transformed variables.

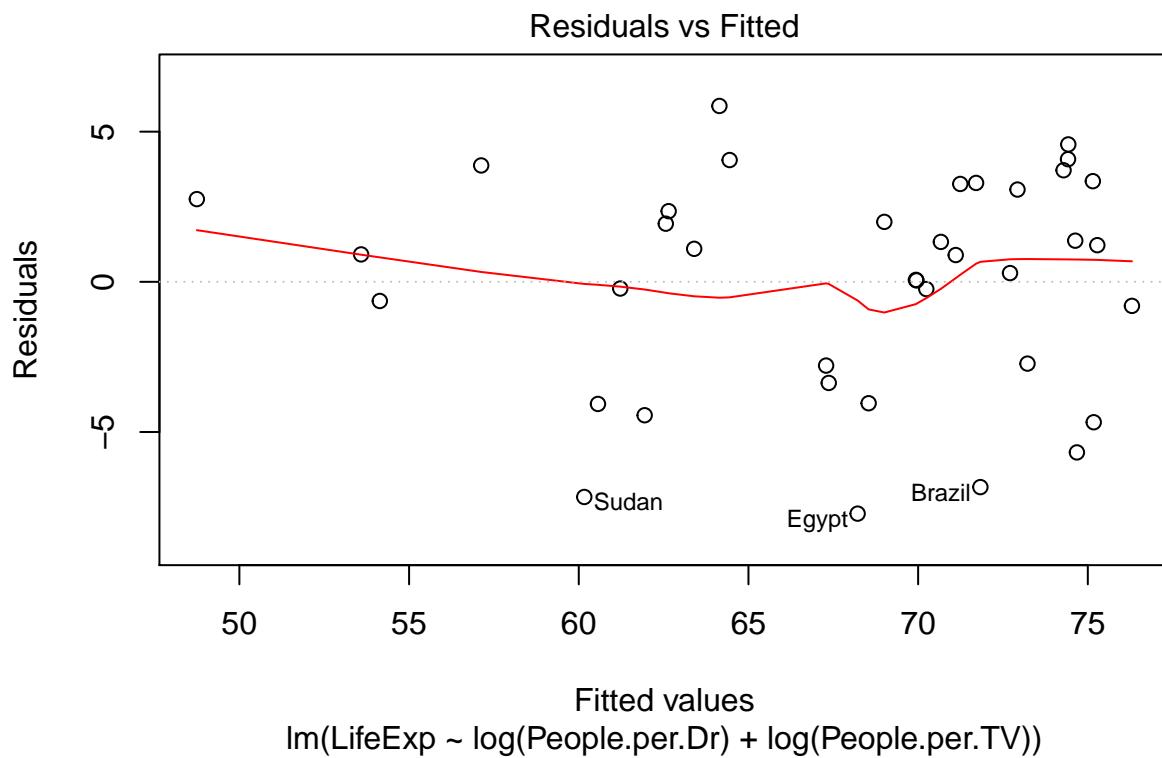
model <- lm(LifeExp~log(People.per.Dr)+log(People.per.TV), data=data)
summary(model)

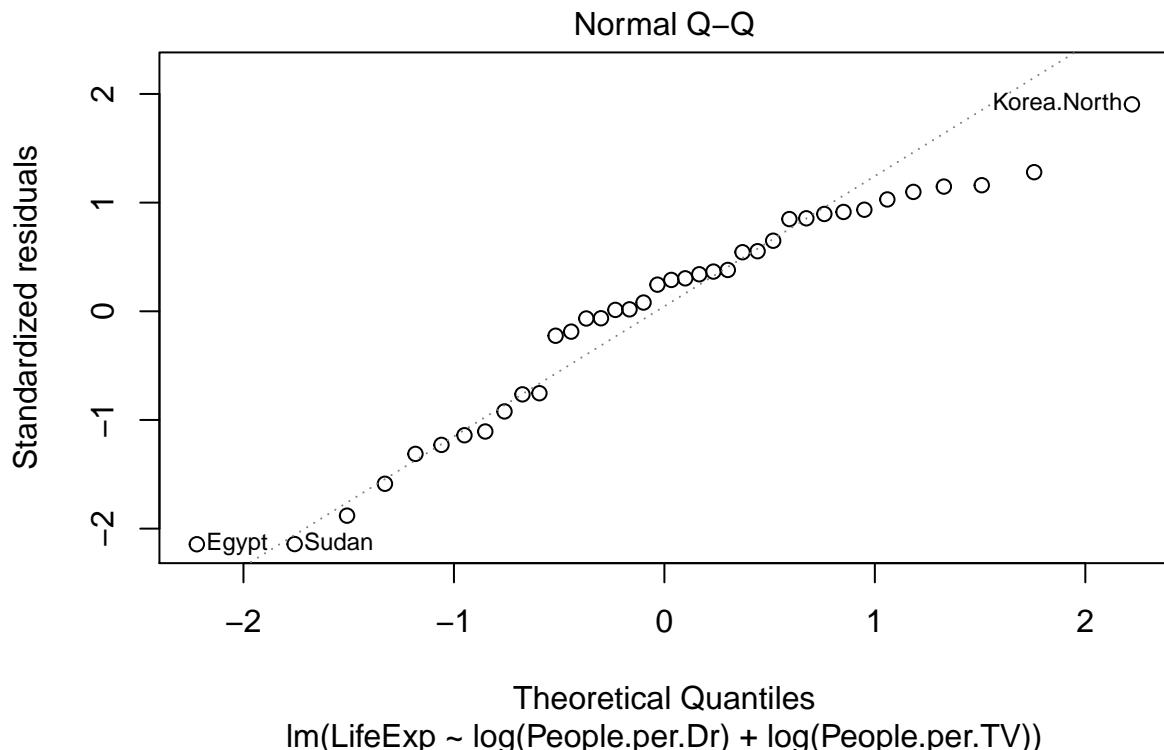
##
## Call:
## lm(formula = LifeExp ~ log(People.per.Dr) + log(People.per.TV),
##      data = data)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -7.7173 -2.7718  0.9026  2.9923  5.8553
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 90.6222   4.3557  20.806 < 2e-16 ***
## log(People.per.Dr) -2.2589   0.7474 -3.022  0.00467 **
## log(People.per.TV) -2.9156   0.5907 -4.936 1.95e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.704 on 35 degrees of freedom
## Multiple R-squared:  0.7868, Adjusted R-squared:  0.7747
## F-statistic:  64.6 on 2 and 35 DF,  p-value: 1.788e-12

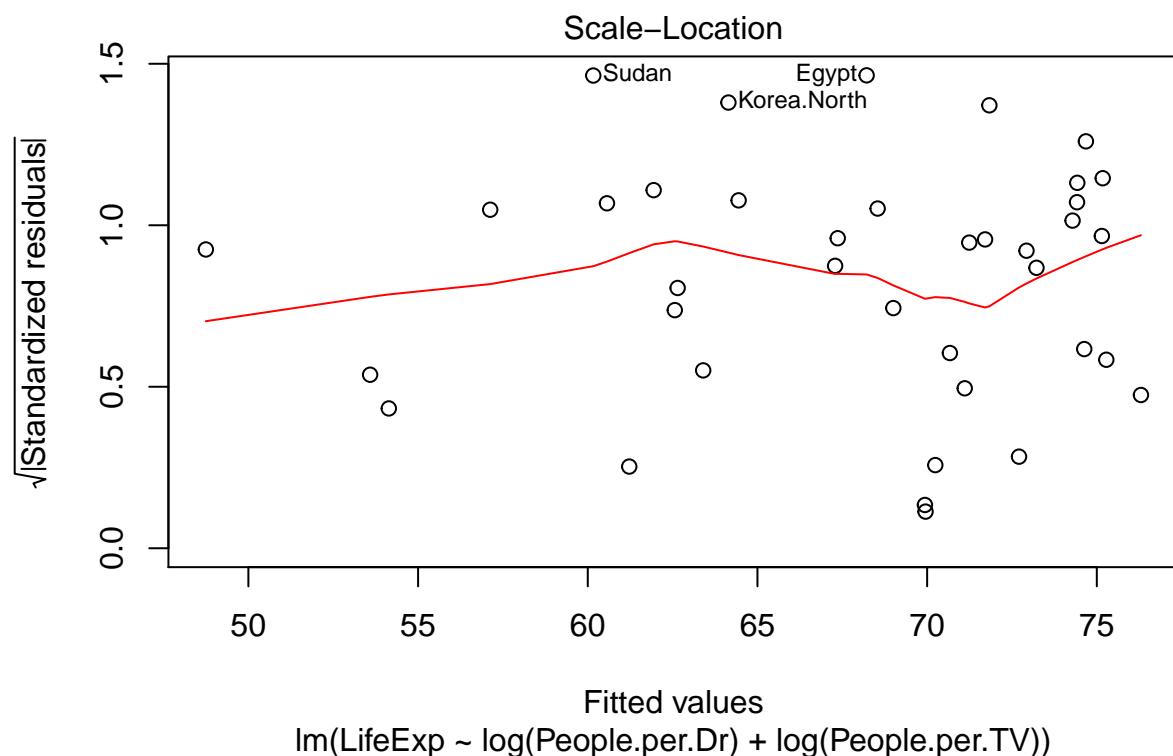
```

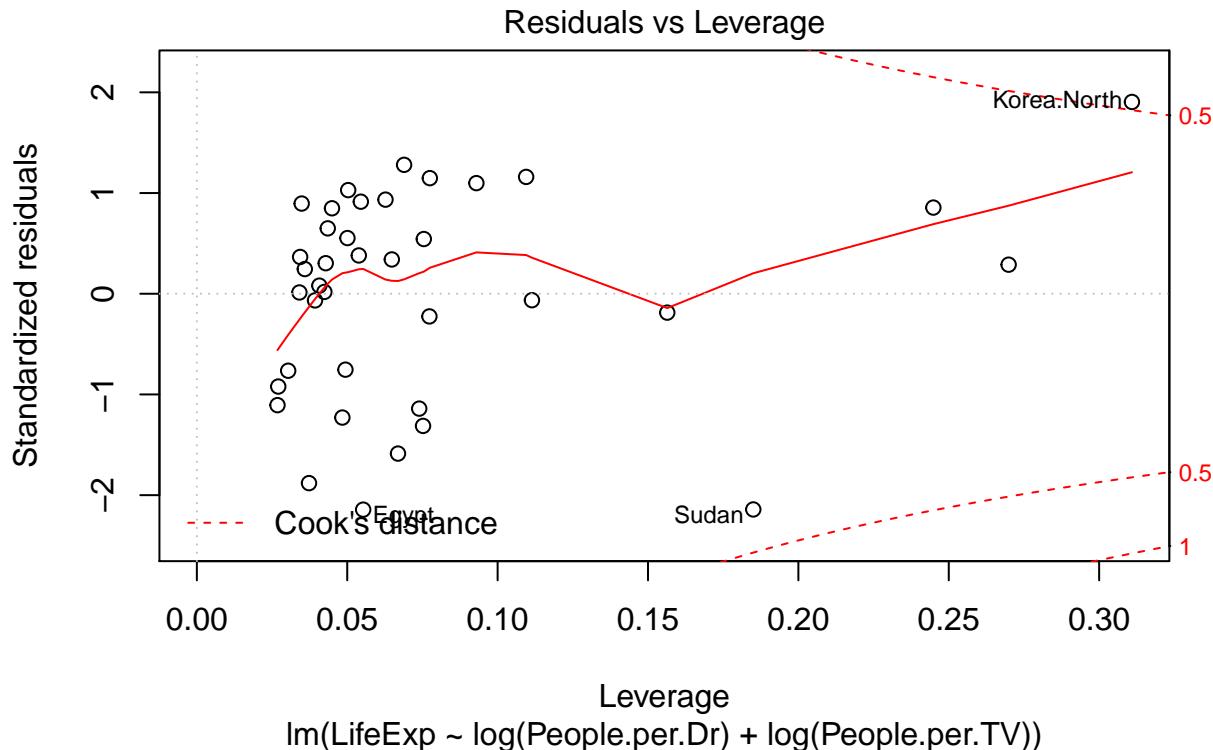
Beware: the coefficients refer to the log-transformed variables, hence the right interpretation , for instance of the second coefficient, would be: by increasing the number of people per Dr. by a factor of (e), while keeping the other variable fixed, the life expectancy would, on average, decrease by -2.25\*\*.

```
plot(model)
```









Can we conclude that more TVs imply a higher life expectancy? No, because we're not analysing the data with a causal model. However, we can use the estimated coefficient to predict the LifeExp for a new point.

Looking at the Cook distance we can clearly pinpoint at least two outliers in the dataset: 17 and 30.

```
data[c(17,30),]
```

```
##           LifeExp People.per.TV People.per.Dr
## Korea.North      70         90        370
## Sudan          53         23     12550
```

Let's remove the two outliers and refit the model.

```
data.no.out <- data[c(-17,-30),]
dim(data.no.out)
```

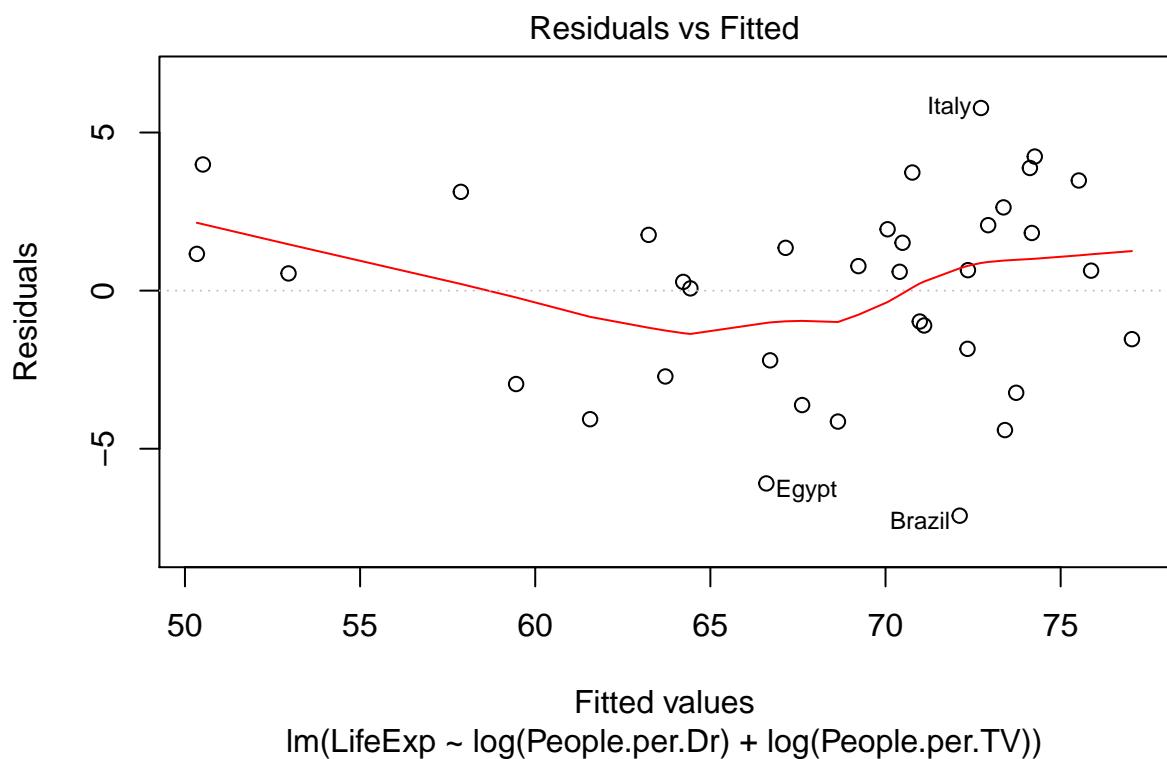
```
## [1] 36 3
model <- lm(LifeExp ~ log(People.per.Dr) + log(People.per.TV), data=data.no.out)
summary(model)
```

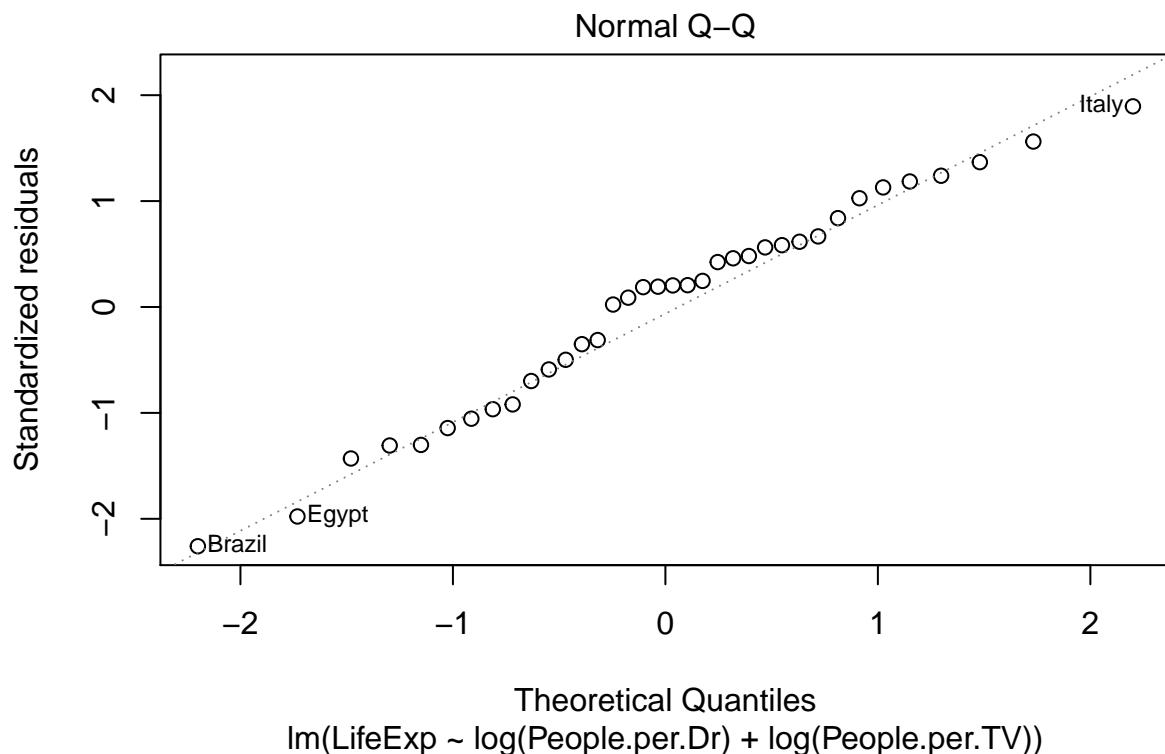
```
##
## Call:
## lm(formula = LifeExp ~ log(People.per.Dr) + log(People.per.TV),
##     data = data.no.out)
##
## Residuals:
##    Min     1Q   Median     3Q    Max
## -7.1175 -2.3328  0.6134  1.9728  5.7746
##
```

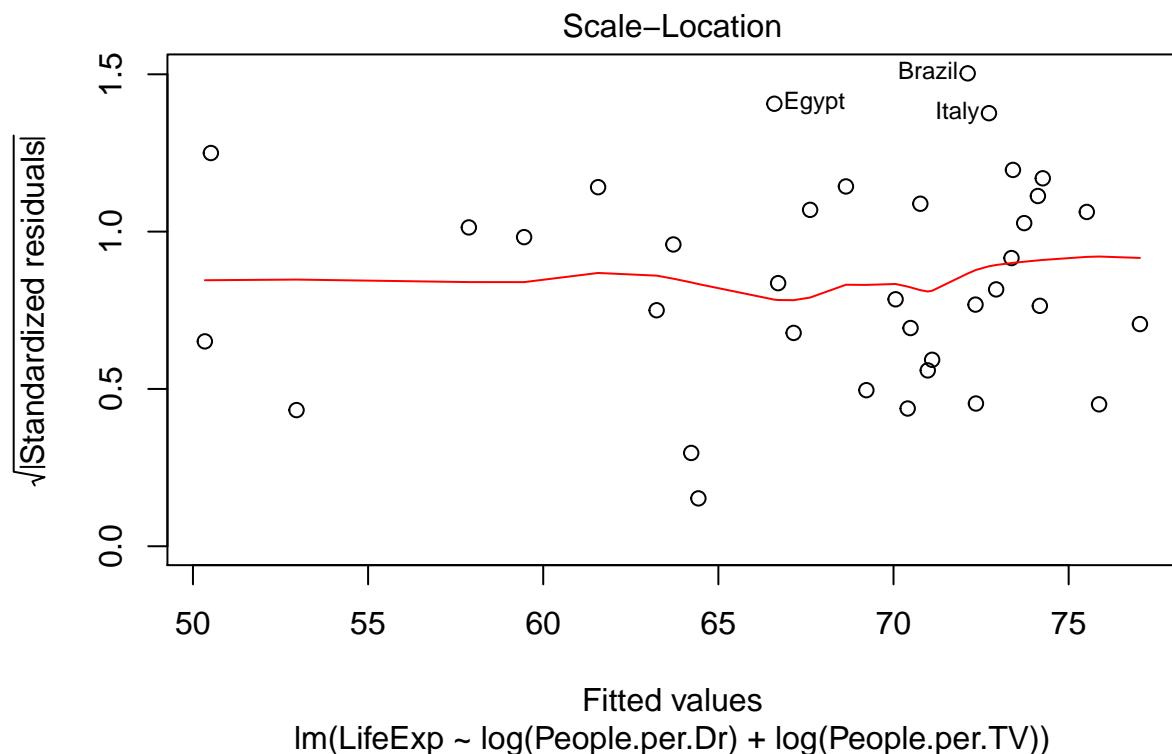
```

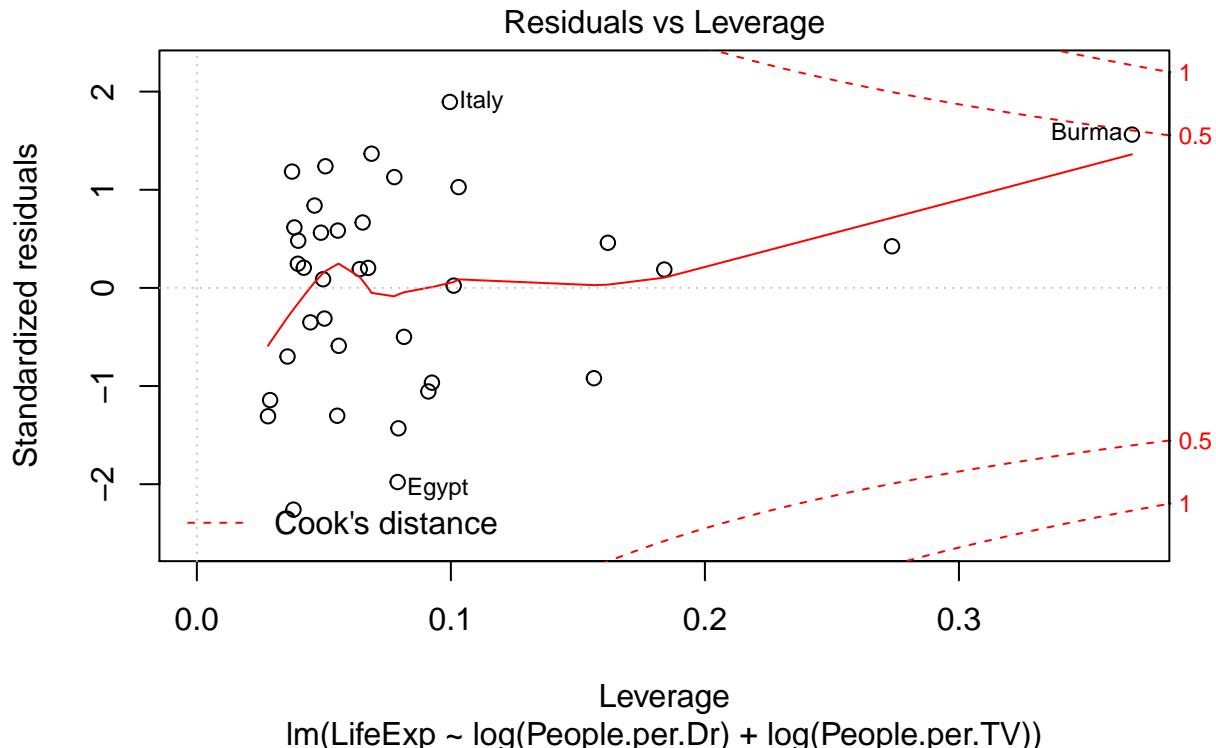
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           80.3245     4.7221 17.010 < 2e-16 ***
## log(People.per.Dr) -0.3642     0.8327 -0.437   0.665
## log(People.per.TV) -4.2050     0.6348 -6.624 1.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.212 on 33 degrees of freedom
## Multiple R-squared:  0.8319, Adjusted R-squared:  0.8217
## F-statistic: 81.63 on 2 and 33 DF,  p-value: 1.675e-13
plot(model)

```









Notice that the R-squared has significantly improved, while the coefficient estimates have changed markedly. Now let's use the summary data to compute some confidence intervals.

```
new.point <- data.frame(People.per.Dr=3000,People.per.TV=50)
#95% confidence interval for the
predict(model, new.point, interval = "confidence")

##          fit      lwr      upr
## 1 60.95844 59.37872 62.53816

predict(model, new.point, interval = "prediction")

##          fit      lwr      upr
## 1 60.95844 54.23489 67.68199
```

Where did these 2 interval come out from?

```
n <- dim(data.no.out)[1]
p <- dim(data.no.out)[2]
n

## [1] 36
p

## [1] 3

beta.hat <- model$coefficients
x0 <- matrix(c(1,log(3000),log(50)))
point.estimate <- t(x0) %*% beta.hat
point.estimate
```

```

##           [,1]
## [1,] 60.95844

X <- as.matrix(cbind(1,data.no.out[,2:3]))
se.hat <- summary(model)$sigma
xtx.inv <- solve(t(X) * X)
confidence.average <- sqrt(t(x0) * xtx.inv * x0) * se.hat * qt(0.975, df=n-p)
c(point.estmate - confidence.average, point.estmate + confidence.average)

## [1] 59.76292 62.15395

confidence.actual <- sqrt(1 + t(x0) * xtx.inv * x0) * se.hat * qt(0.975, df=n-p)
c(point.estmate - confidence.actual, point.estmate + confidence.actual)

## [1] 54.31465 67.60223

```

## The bias variance tradeoff

We're going to show the bias-variance tradeoff on a made up dataset.

```

# Creating the dataset
nsim <- 1000
n <- 100

# A non linear function
f <- function(x){
  .3*x - 0.2*x^2 + 0.1*x^3 + sin(2*x)
}

generate_dataset <- function(x, sigma, n){
  y <- f(x) + rnorm(n, mean=0, sd = sigma)
  return(y)
}

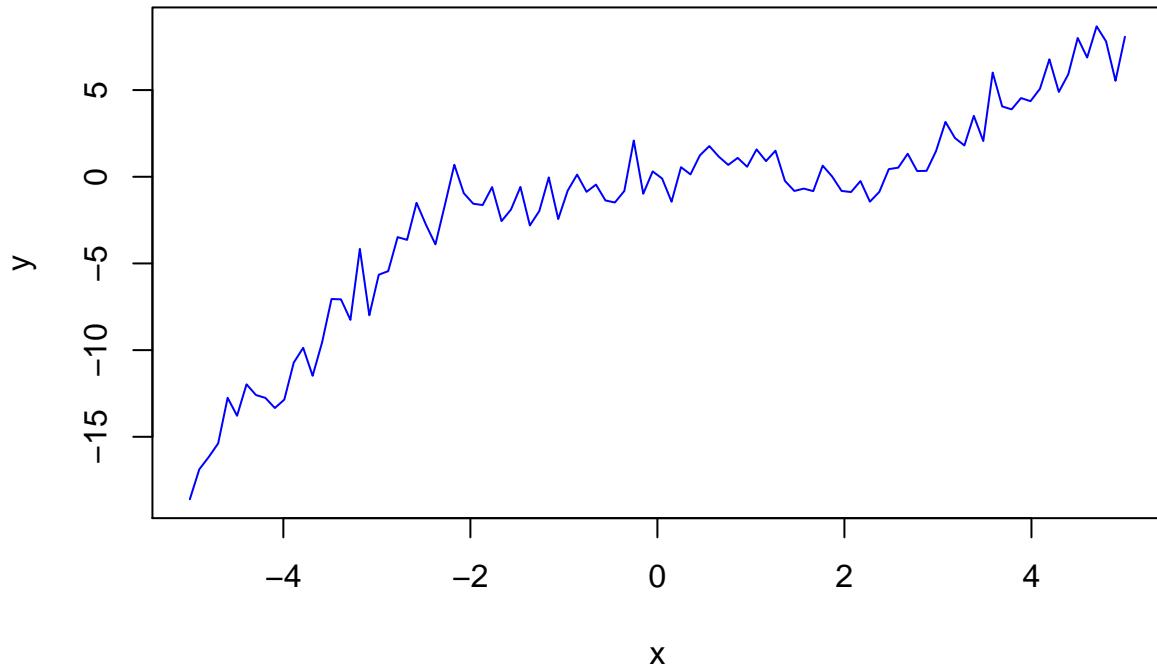
```

Let's have a look at a generated dataset:

```

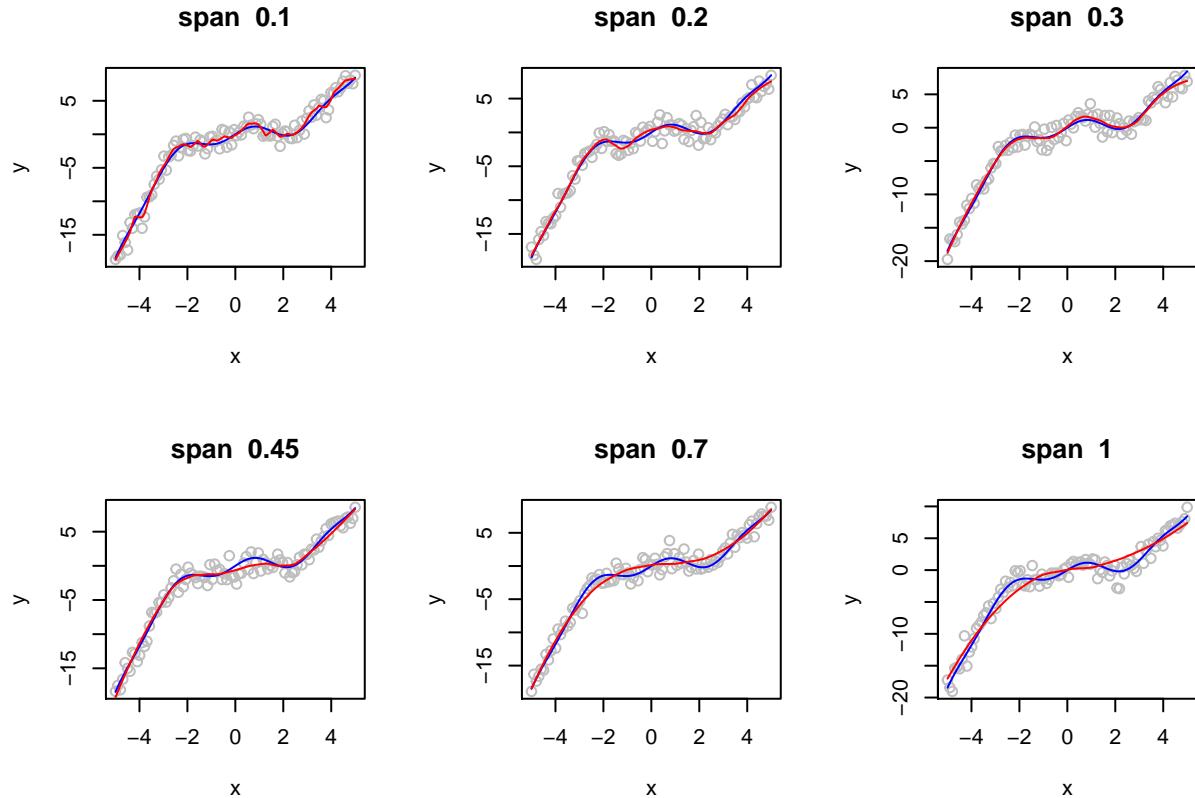
x <- seq(-5, 5, length = 100)
y <- generate_dataset(x, 1,100)
plot(x,y, type = "l", col="blue")

```



We're going to use local regression, and we're going to play a bit with its smoothing hyperparameter alpha to study its correlation with bias and variance.

```
alpha.seq <- c(0.1,0.2,0.3,0.45,0.7,1) # small values -> little smoothing
par(mfrow=c(2,3))
for(j in 1:6){
  span <- alpha.seq[j]
  y <- generate_dataset(x, 1, n)
  plot(x,y, main=paste("span ",span), col="gray")
  fit <- loess(y~x,span=span)
  preds <- predict(fit, x)
  lines(x, f(x), col="blue")
  lines(x, preds, col="red")
}
```



Let's play a bit with sigma and n to get a sense of the influence of both the noise level and the dataset dimension on the fit.

```

ns <- c(20,100,500,1000)
sigmas <-c(0.2,1,2)
alpha.seq <- c(0.1,0.2,0.3,0.45,0.7,1) # small values -> little smoothing
titles = c()
for(n.i in ns){
  for(sigma.i in sigmas){
    par(mfrow=c(2,3))
    for(j in 1:6){
      span <- alpha.seq[j]
      x <- seq(-5, 5, length = n.i)
      y <- generate_dataset(x, sigma.i, n.i)
      plot(x,y, main=paste("SIGMA ",sigma.i," N ",n.i," span ",span), col="gray")
      fit <- loess(y~x,span=span)
      preds <- predict(fit, x)
      lines(x, f(x), col="blue")
      lines(x, preds, col="red")
    }
  }
}

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

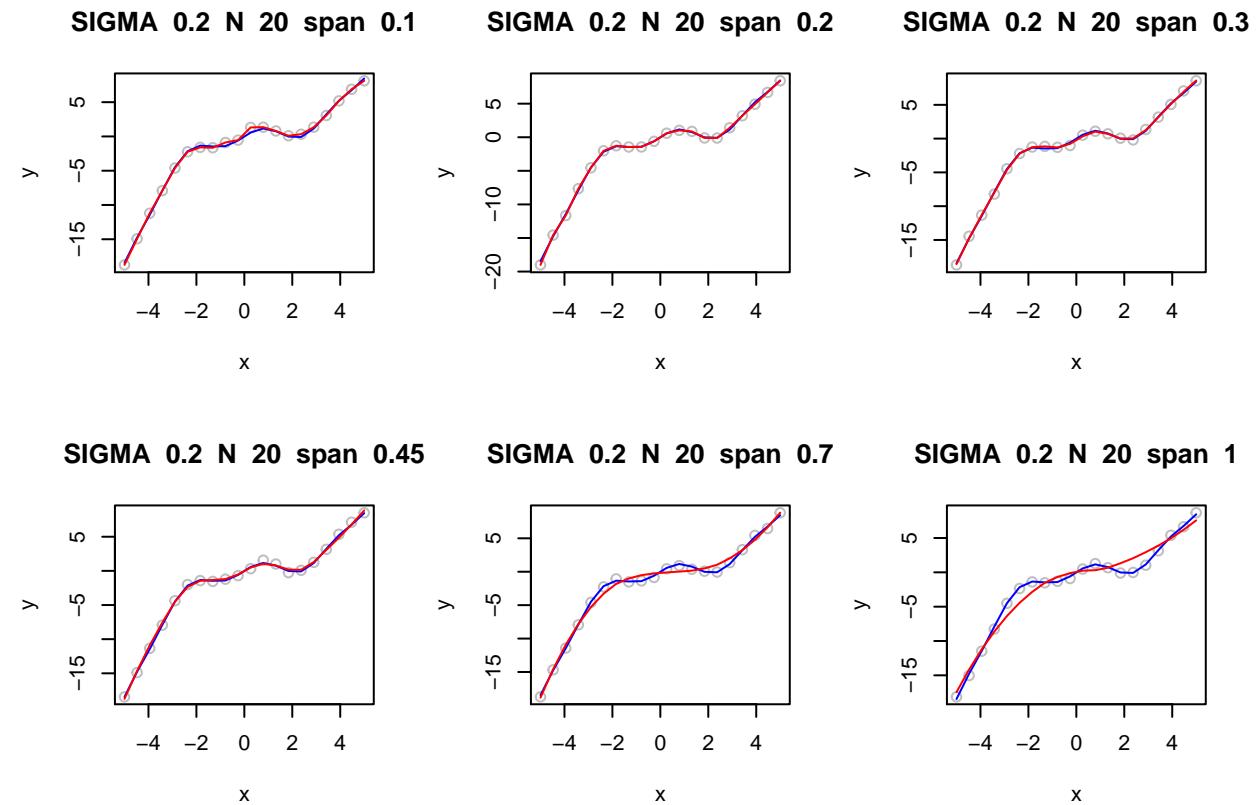
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -5.05

```

```

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.57632
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.33214
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 20
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 20
## Warning in sqrt(sum.squares/one.delta): NaNs produced

```



```

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -5.05
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.57632
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =

```

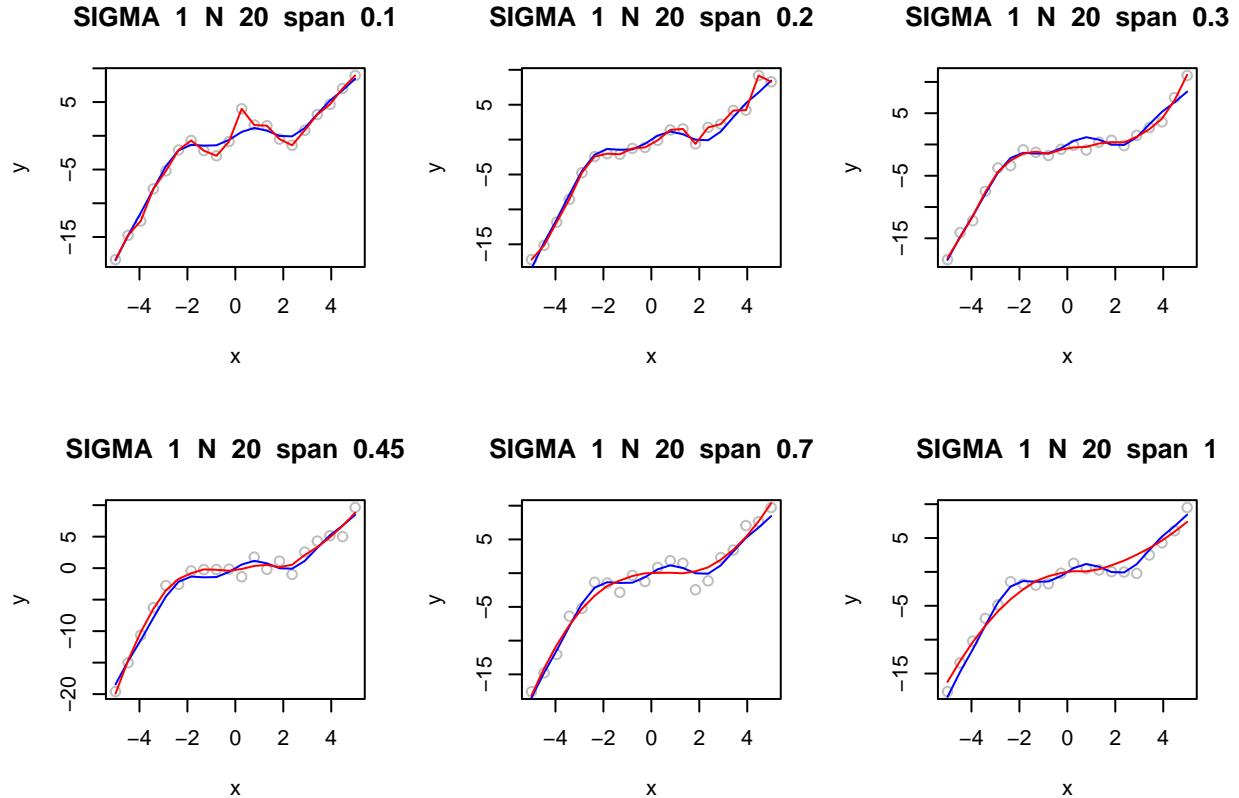
```

## parametric, : There are other near singularities as well. 0.33214
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 20

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 20

## Warning in sqrt(sum.squares/one.delta): NaNs produced

```



```

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -5.05

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.57632

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

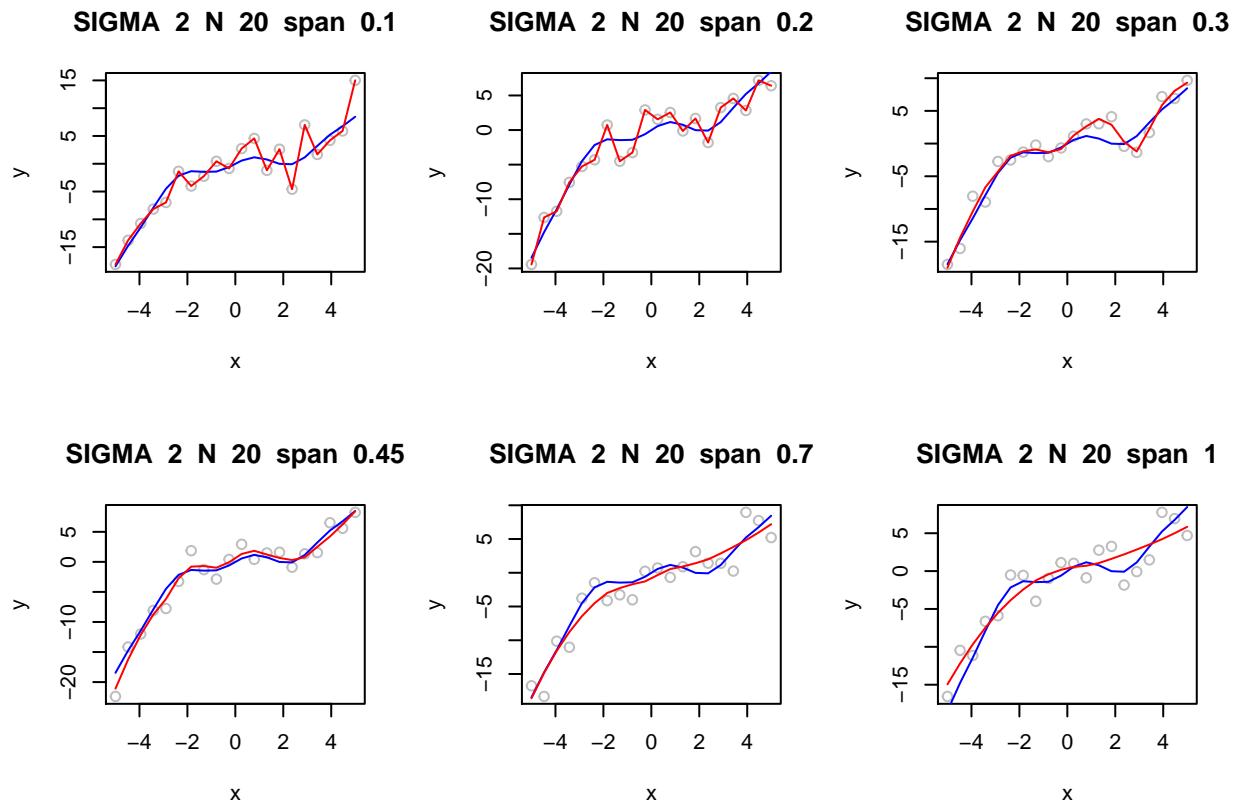
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.33214

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 20

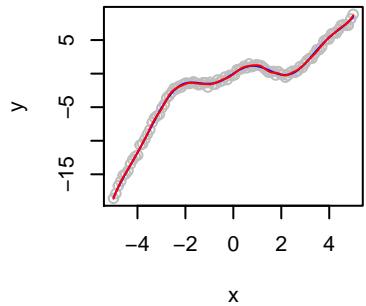
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 20

```

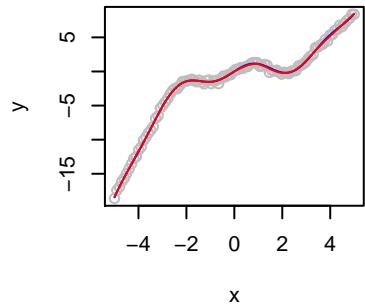
```
## Warning in sqrt(sum.squares/one.delta): NaNs produced
```



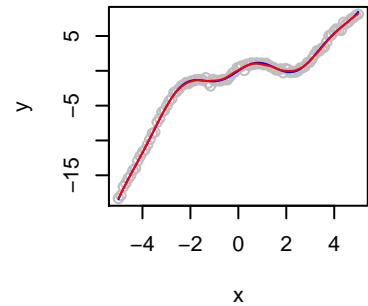
**SIGMA 0.2 N 100 span 0.1**



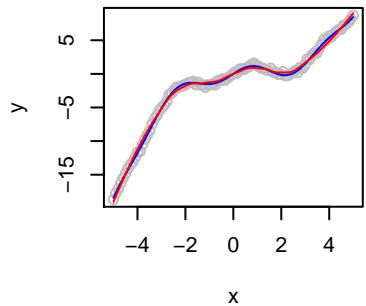
**SIGMA 0.2 N 100 span 0.2**



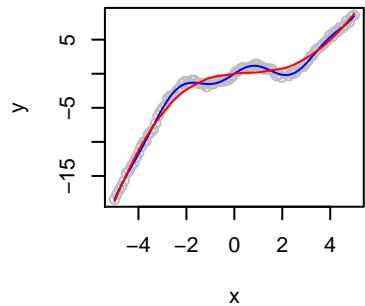
**SIGMA 0.2 N 100 span 0.3**



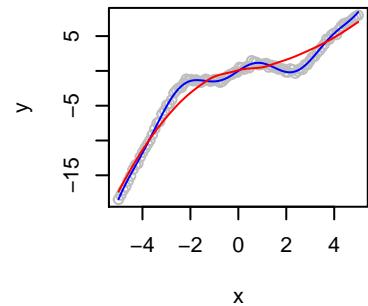
**SIGMA 0.2 N 100 span 0.45**



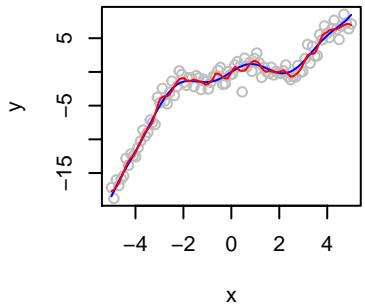
**SIGMA 0.2 N 100 span 0.7**



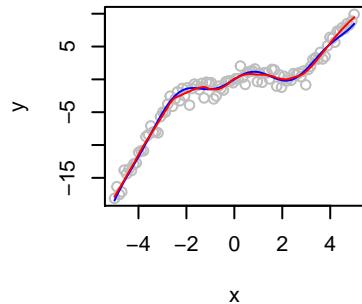
**SIGMA 0.2 N 100 span 1**



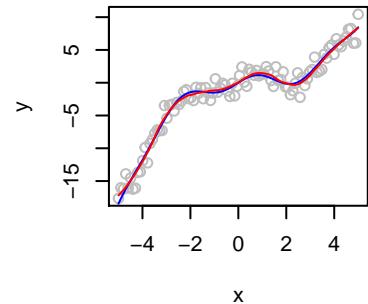
**SIGMA 1 N 100 span 0.1**



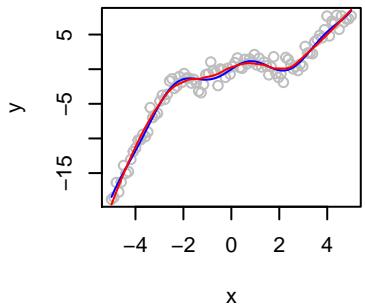
**SIGMA 1 N 100 span 0.2**



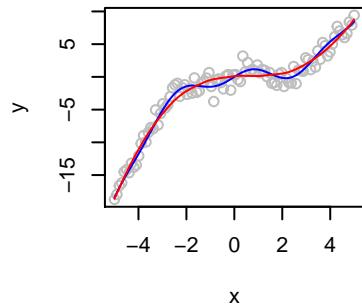
**SIGMA 1 N 100 span 0.3**



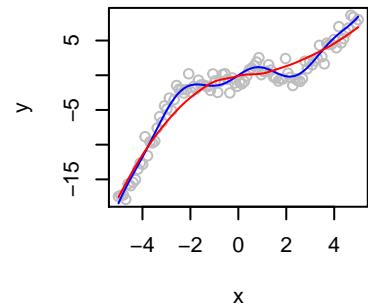
**SIGMA 1 N 100 span 0.45**



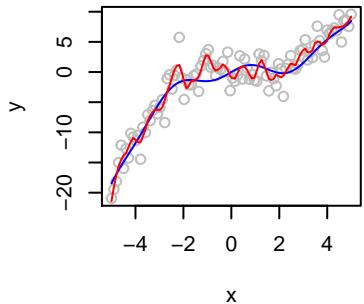
**SIGMA 1 N 100 span 0.7**



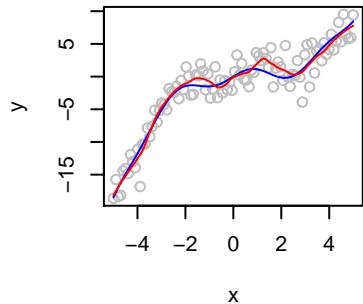
**SIGMA 1 N 100 span 1**



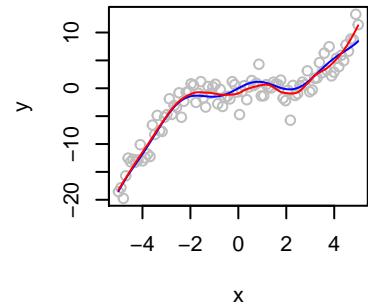
**SIGMA 2 N 100 span 0.1**



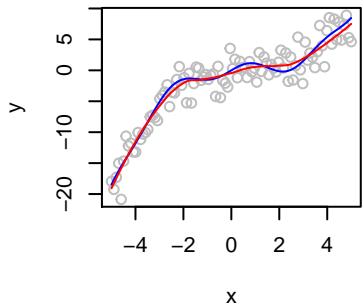
**SIGMA 2 N 100 span 0.2**



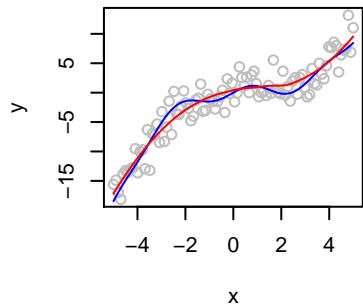
**SIGMA 2 N 100 span 0.3**



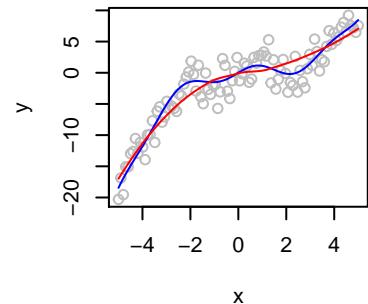
**SIGMA 2 N 100 span 0.45**



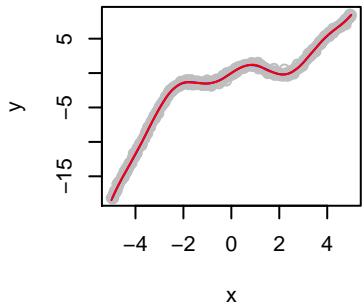
**SIGMA 2 N 100 span 0.7**



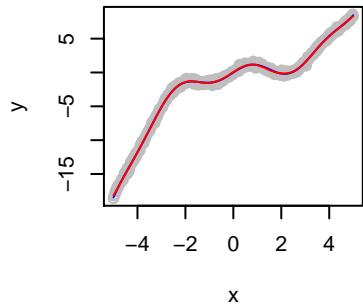
**SIGMA 2 N 100 span 1**



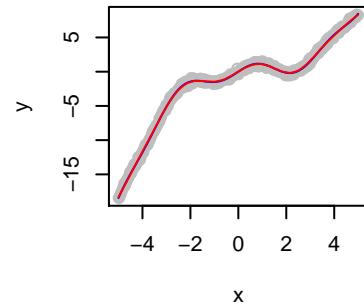
**SIGMA 0.2 N 500 span 0.1**



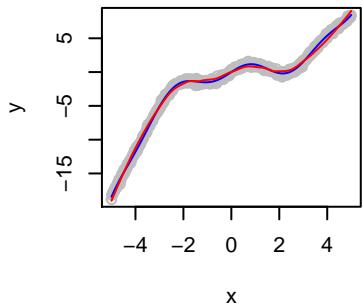
**SIGMA 0.2 N 500 span 0.2**



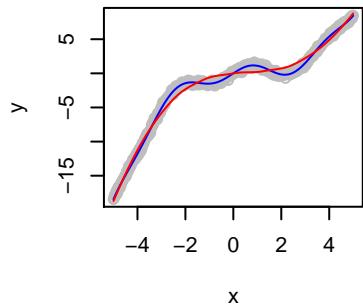
**SIGMA 0.2 N 500 span 0.3**



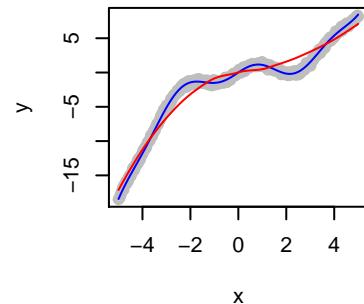
**SIGMA 0.2 N 500 span 0.45**



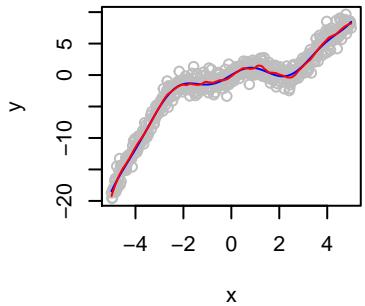
**SIGMA 0.2 N 500 span 0.7**



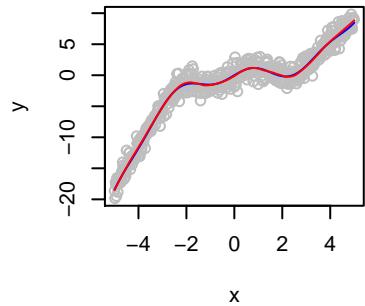
**SIGMA 0.2 N 500 span 1**



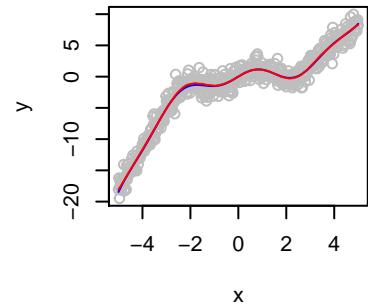
**SIGMA 1 N 500 span 0.1**



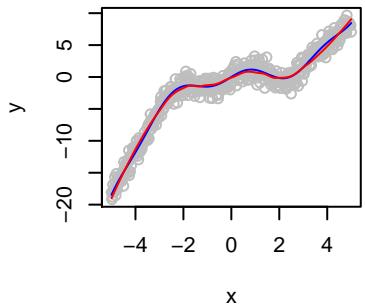
**SIGMA 1 N 500 span 0.2**



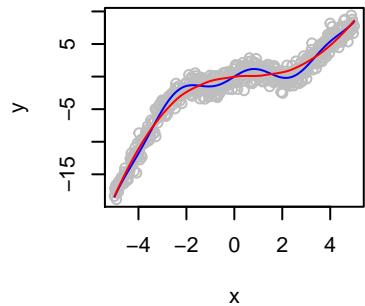
**SIGMA 1 N 500 span 0.3**



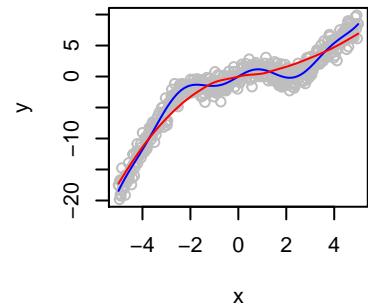
**SIGMA 1 N 500 span 0.45**

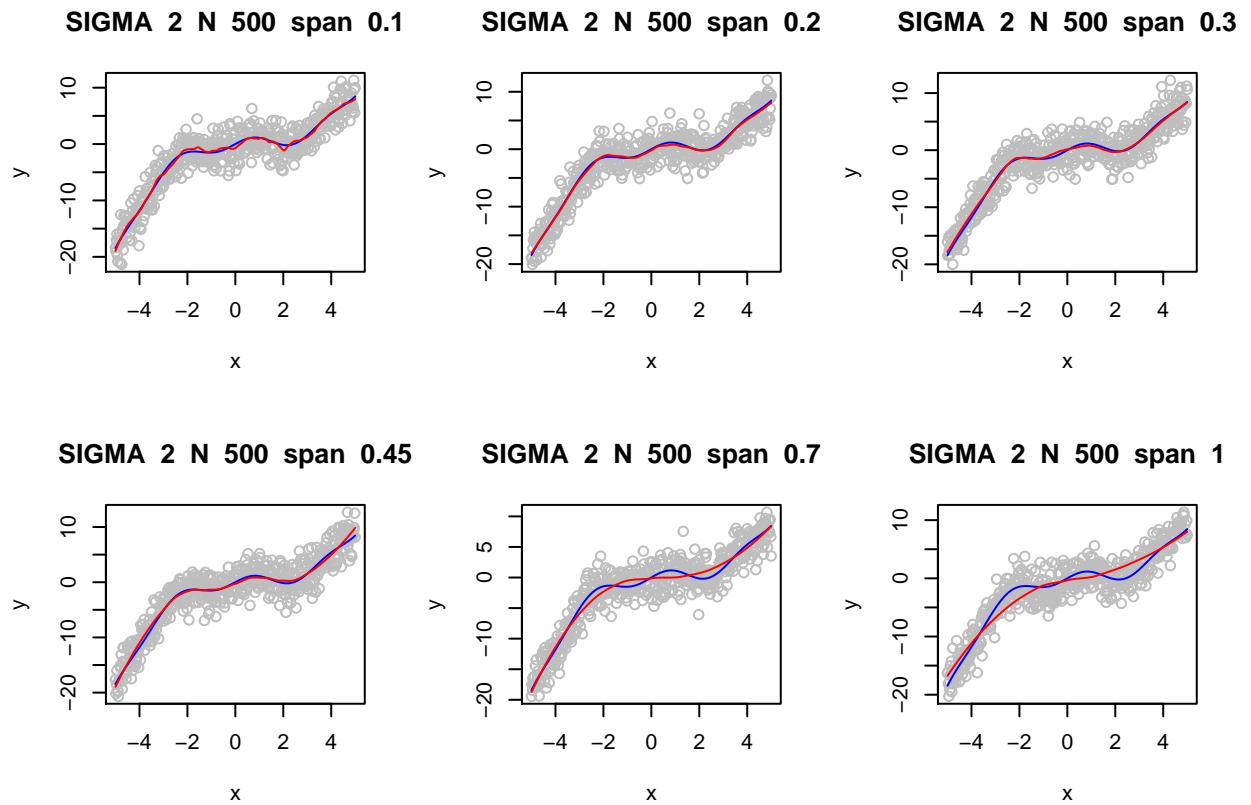


**SIGMA 1 N 500 span 0.7**

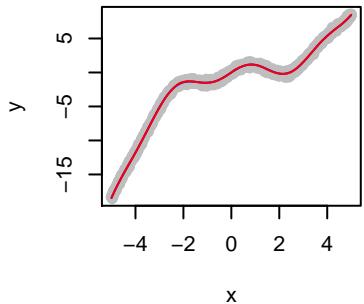


**SIGMA 1 N 500 span 1**

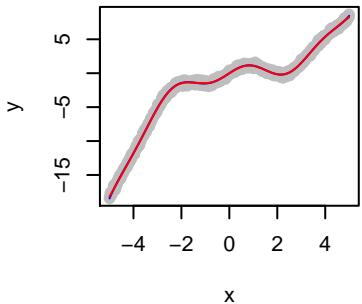




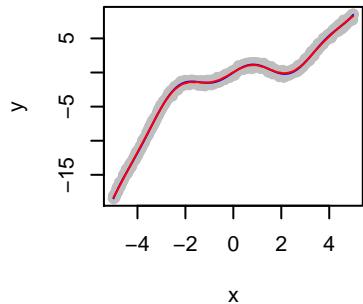
**SIGMA 0.2 N 1000 span 0.1**



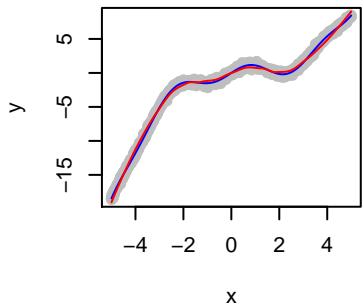
**SIGMA 0.2 N 1000 span 0.2**



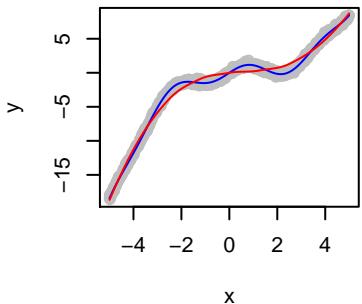
**SIGMA 0.2 N 1000 span 0.3**



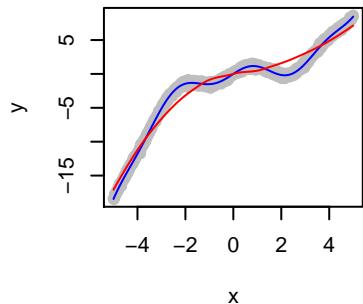
**SIGMA 0.2 N 1000 span 0.4**



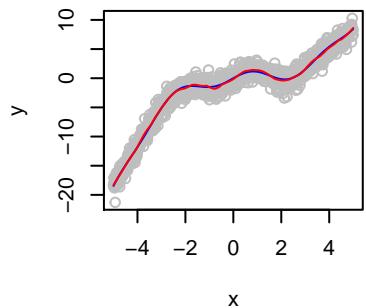
**SIGMA 0.2 N 1000 span 0.7**



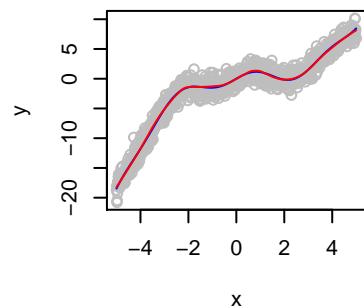
**SIGMA 0.2 N 1000 span 1**



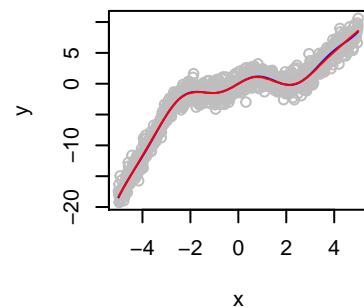
**SIGMA 1 N 1000 span 0.1**



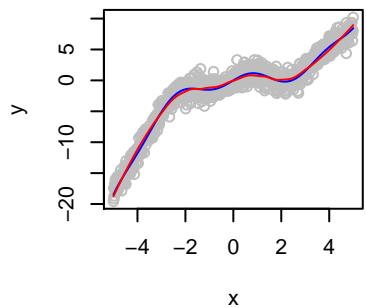
**SIGMA 1 N 1000 span 0.2**



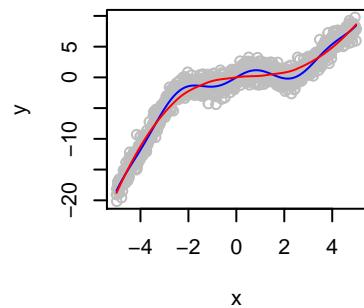
**SIGMA 1 N 1000 span 0.3**



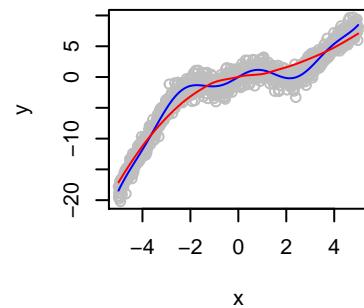
**SIGMA 1 N 1000 span 0.45**

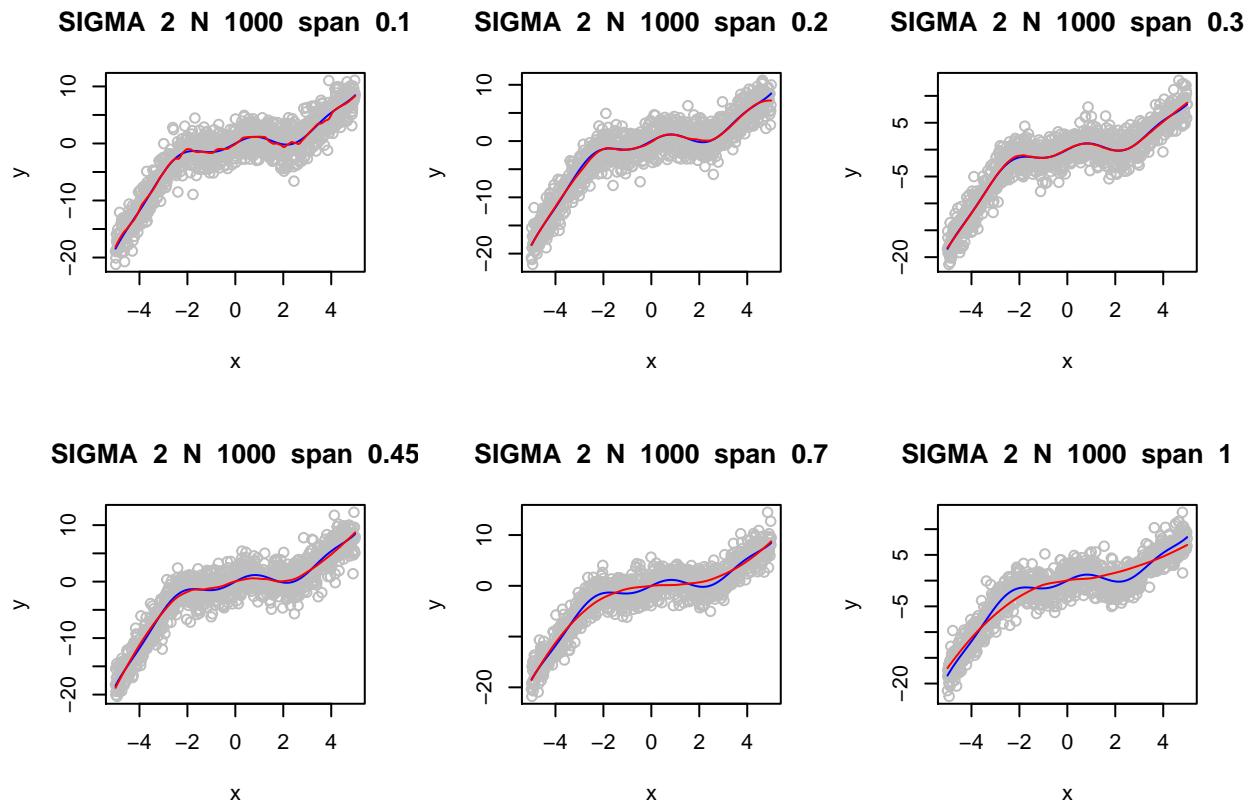


**SIGMA 1 N 1000 span 0.7**



**SIGMA 1 N 1000 span 1**

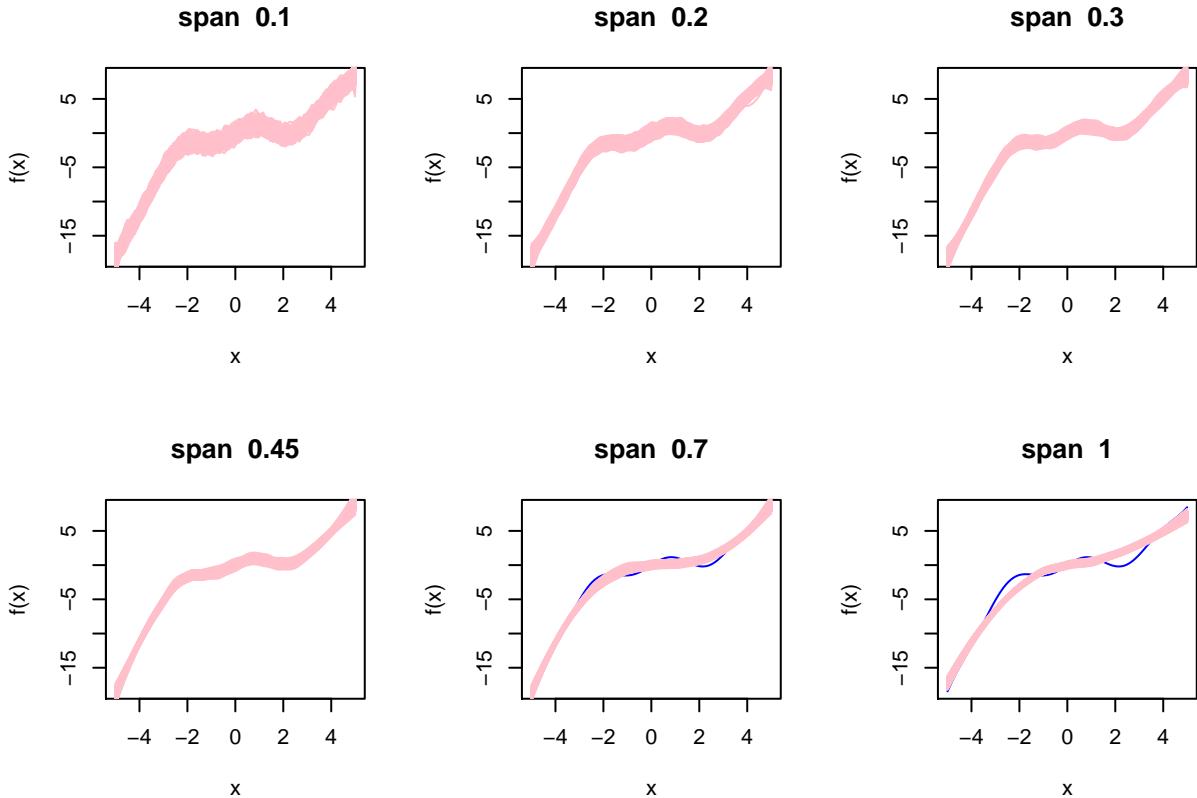




Notice how the high-variance curves get smoothed as the dataset size increases, while the biased curves remain biased.

Now let's generate a lot more datasets and let's build a model for each of them to analyse its bias and variance.

```
# again back to n=100
n<- 100
x <- seq(-5, 5, length = n)
par(mfrow=c(2,3))
for(j in 1:6){
  span <- alpha.seq[j]
  plot(x,f(x), type="l", main=paste("span ",span), col="blue")
  for(i in 1:nsim){
    yi <- generate_dataset(x, 1, n)
    fit <- loess(yi~x,span=span)
    preds <- predict(fit, x)
    lines(x, preds, col="pink")
  }
}
```



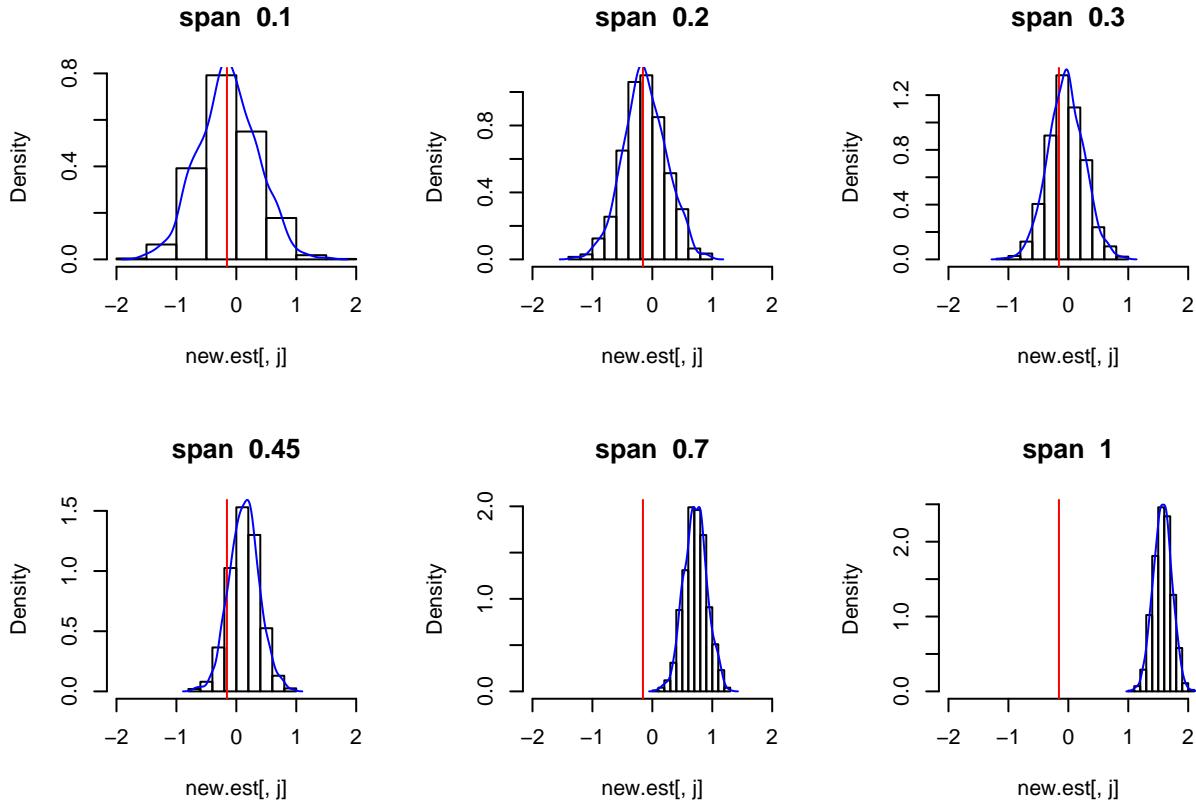
Inspecting the above plots we get a sense of what is called the *bias-variance tradeoff*, which we're now going to illustrate by numbers using an estimate of the MSE.

```

new.x <- 2
new.est <- matrix(nrow=nsim, ncol=6)
new.y <- rep(NA, nsim)
for(i in 1:nsim){
  yi <- generate_dataset(x, 1, n)
  yi.new <- generate_dataset(new.x, 1, 1)
  new.y[i] <-yi.new
  for(j in 1:6){
    span <- alpha.seq[j]
    fit <- loess(yi~x,span=span)
    new.point.pred <- predict(fit, new.x)
    new.est[i,j] <- new.point.pred
  }
}

par(mfrow=c(2,3))
for(j in 1:6){
  hist(new.est[,j], main =paste("span ",alpha.seq[j]), freq = F, xlim = c(-2,2))
  abline(v=f(new.x), col="red")
  lines(density(new.est[,j]), col="blue")
}

```



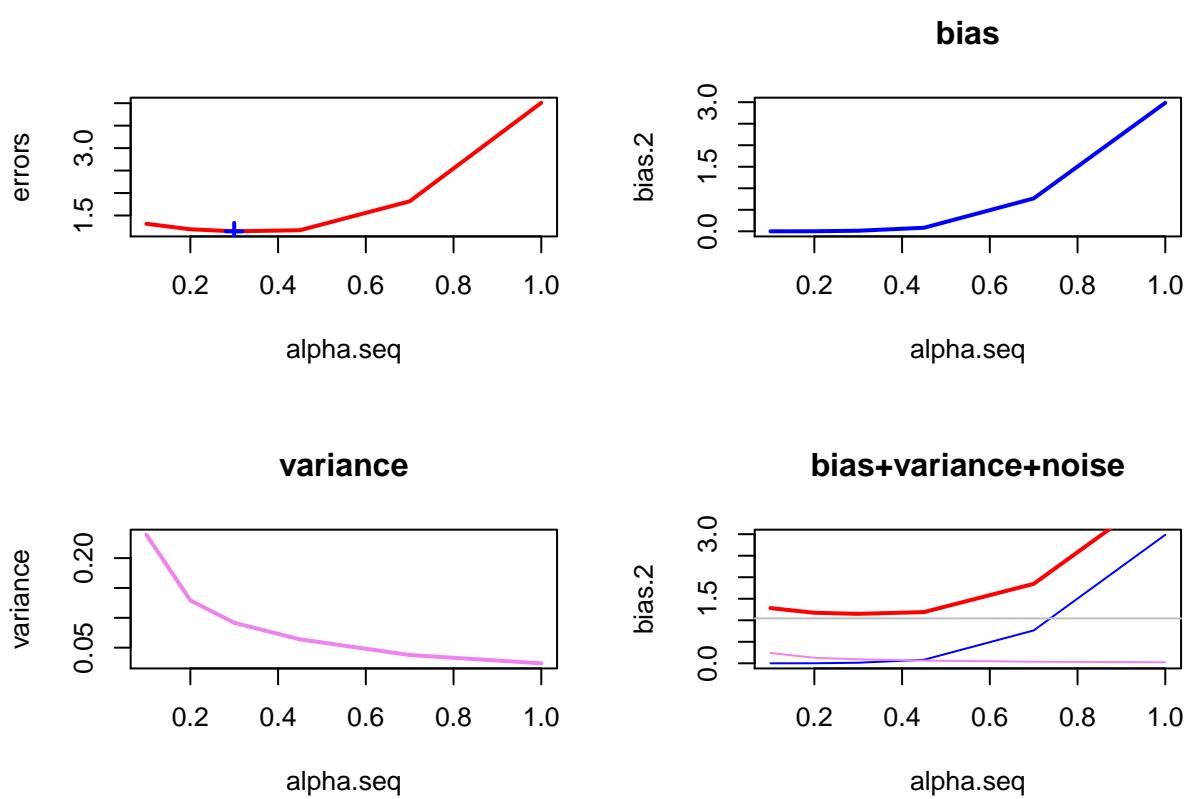
Now let's look at an approximation of the MSE for the different values of alpha, also showing bias, variance and noise level.

```

errors <- colSums((new.y - new.est)**2)/nsim
bias.2 <- (f(new.x) - colMeans(new.est))**2
variance <- apply(new.est, 2, var)
noise.level <- var(new.y)
par(mfrow=c(2,2))
plot(alpha.seq, errors, type="l", lwd=2, col="red")
min <- which.min(errors)
points(alpha.seq[min], errors[min], col="blue", pch=3, lwd=2)
plot(alpha.seq, bias.2, type="l", lwd=2, col="blue", main="bias")
plot(alpha.seq, variance, type="l", lwd=2, col="violet", main="variance")

plot(alpha.seq, bias.2, type="l", lwd=1, col="blue", main="bias+variance+noise")
lines(alpha.seq, variance, lwd=1, col="violet")
lines(alpha.seq, variance+bias.2+noise.level, lwd=2, col="red")
abline(h=noise.level, col="gray")

```



The results above confirm our intuitions, and the bias-variance tradeoff.