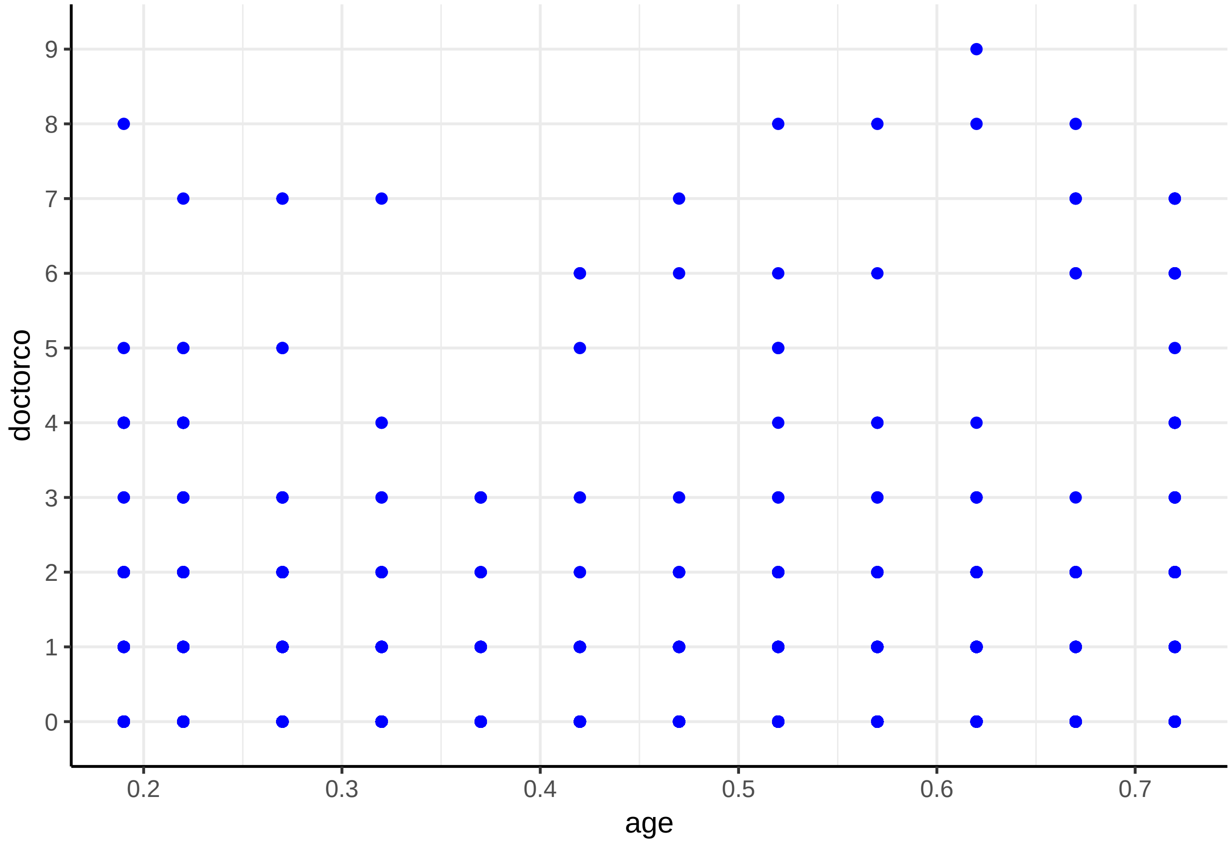
Homework 06

Subhankar Ghosh

Question 1

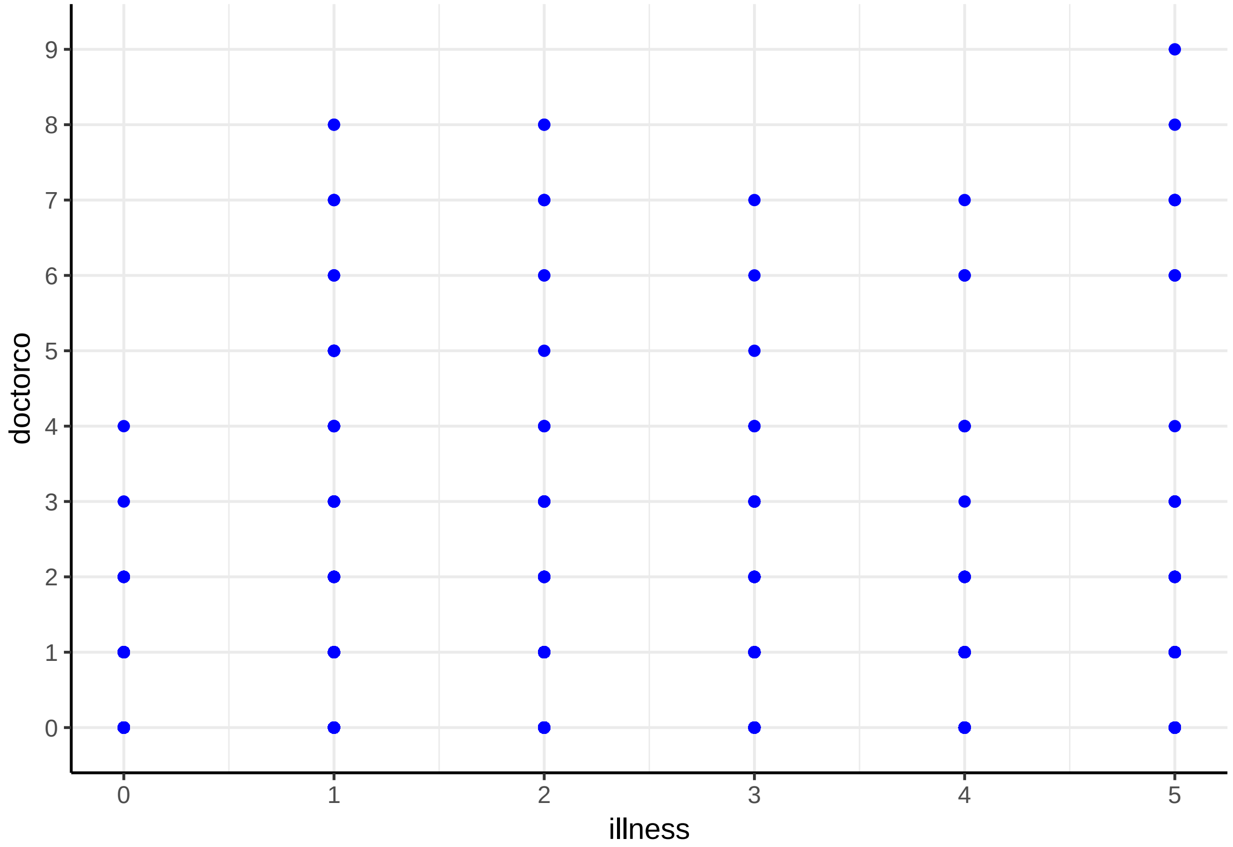
a

```
ggplot(data = dvisits, aes(x = age, y = doctorco)) +
  geom_point(color = "blue") +
  scale_y_discrete(name = "doctorco", limits = seq(0,9,by=1)) +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line())
```



We can notice a pattern that with high age the doctorco value gets high, hinting that older people go to doctors more often.

```
ggplot(data = dvisits, aes(x = illness, y = doctorco)) +
  geom_point(color = "blue") +
  scale_y_discrete(name = "doctorco", limits = seq(0,9,by=1)) +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line())
```



The plot shows the distribution of doctorco variable with respect to illness is uniform. We do not notice any strong particular pattern.

But it is quite obvious that illness level of 5 has high doctorco value.

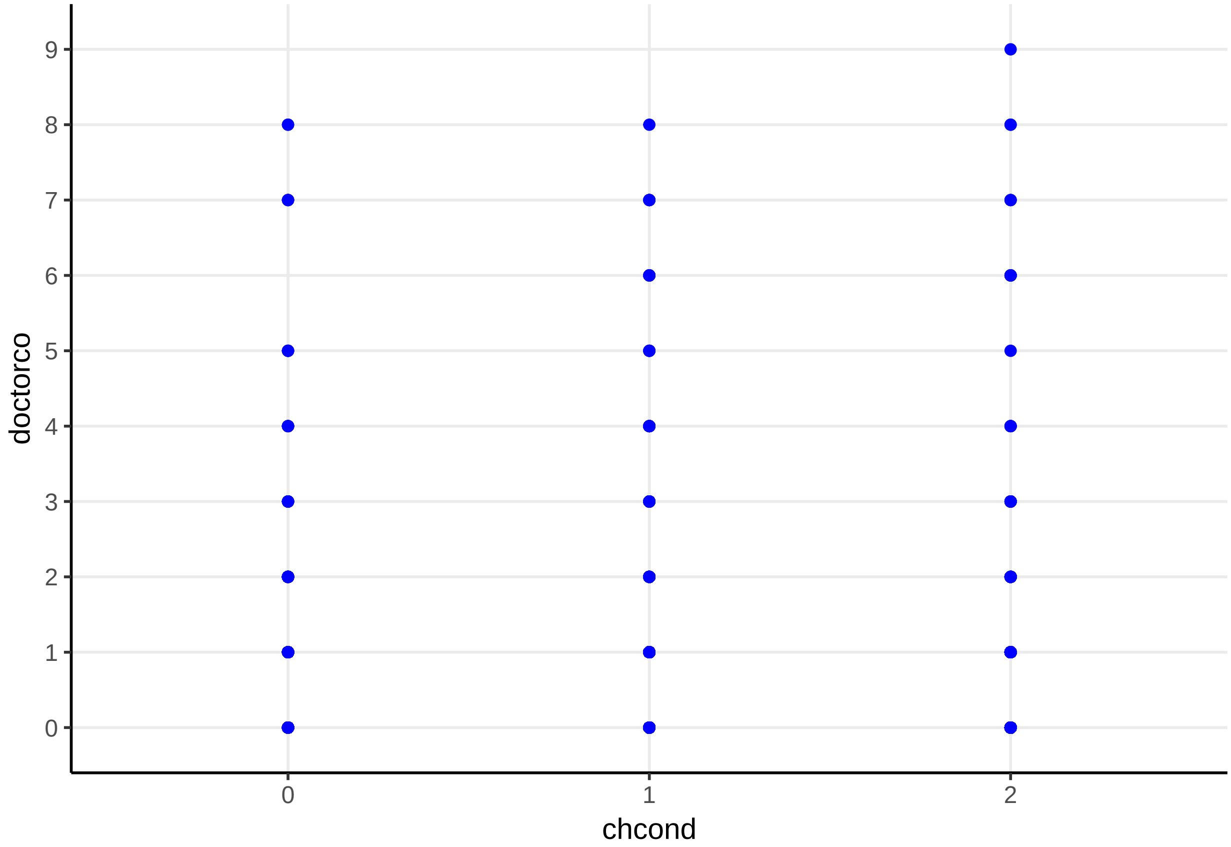
b

chcond1 = 1 if chronic condition(s) but not limited in activity, 0 chronic condition(s) and limited in activity | no chronic condition

chcond2 = 1 if chronic condition(s) and limited in activity, 0 otherwise

New variable chcond = 0 = No chronic condition 1 = chronic condition(s) but not limited in activity 2 = chronic condition(s) and limited in activity

```
dvisits2 = dvisits
dvisits2$chcond = ifelse(((dvisits2$chcond1 == 0) & (dvisits2$chcond2 == 0)), 0,
                         ifelse(dvisits2$chcond1 == 1, 1, 2))
dvisits2$chcond = as.factor(dvisits2$chcond)

ggplot(data = dvisits2, aes(x = chcond, y = doctorco)) +
  geom_point(color = "blue") +
  scale_y_discrete(name = "doctorco", limits = seq(0,9,by=1)) +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line())
```



Here the distribution is quite uniformly spread out.

c

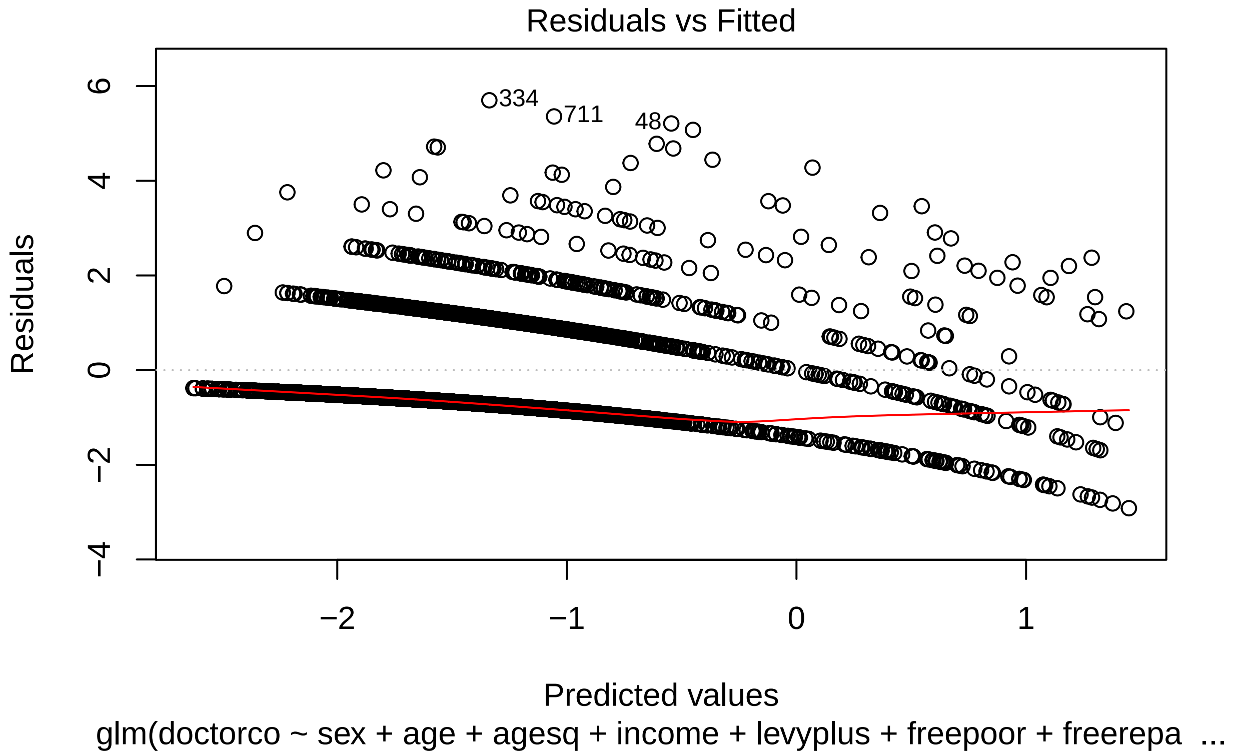
```
poisson_mdl = glm(doctorco ~ sex + age + agesq + income + levyplus + freepoor +
                    freerepa + illness + actdays + hscore + chcond,
                  family = "poisson",
                  data = dvisits2)
poisson_mdl
```

```
##
## Call:  glm(formula = doctorco ~ sex + age + agesq + income + levyplus +
##     freepoor + freerepa + illness + actdays + hscore + chcond,
##     family = "poisson", data = dvisits2)
##
## Coefficients:
## (Intercept)          sex          age        agesq       income
##    -2.22385      0.15688      1.05630     -0.84870     -0.20532
##    levyplus     freepoor     freerepa      illness      actdays
##     0.12319     -0.44006      0.07980      0.18695      0.12685
##      hscore      chcond1      chcond2
##     0.03008      0.11409      0.14116
##
## Degrees of Freedom: 5189 Total (i.e. Null);  5177 Residual
## Null Deviance:       5635
## Residual Deviance: 4380  AIC: 6737
```

Residual deviance of the model is 4380 which is very high so we can conclude that this model does not fit the data very well. It is not really very different from the null deviance of 5635

d

```
plot(poisson_mdl, which = 1)
```



Why the curves?

This is the appearance you expect of such a plot when the dependent variable is discrete. Each curvilinear trace of points on the plot corresponds to a fixed value of the dependent variable "doctorco". The plot of $k???\hat{y}$ versus $\hat{y}$ is obviously a line with slope ???1. In Poisson regression, the x-axis is shown on a log scale: it is $log(\hat{y})$. The curves now bend down exponentially. As kk varies, these curves rise by integral amounts. Exponentiating them gives a set of quasi-parallel curves.

e

```
model.aic = step(poisson_mdl, direction = "backward", trace = 1)
```

```
## Start:  AIC=6737.08
## doctorco ~ sex + age + agesq + income + levyplus + freepoor + 
##     freerepa + illness + actdays + hscore + chcond
## 
##            Df Deviance    AIC
## - agesq     1   4380.1 6735.7
## - freerepa  1   4380.3 6735.8
## - age       1   4380.6 6736.2
## - chcond    2   4383.2 6736.7
## <none>          4379.5 6737.1
## - levyplus  1   4382.5 6738.1
## - income    1   4385.0 6740.5
## - freepoor  1   4386.2 6741.8
## - sex       1   4387.4 6743.0
## - hscore    1   4388.1 6743.7
## - illness   1   4481.8 6837.4
## - actdays   1   4917.1 7272.7
##
## Step:  AIC=6735.7
## doctorco ~ sex + age + income + levyplus + freepoor + freerepa +
##     illness + actdays + hscore + chcond
##
##            Df Deviance    AIC
## - freerepa  1   4381.0 6734.5
## <none>          4380.1 6735.7
## - chcond    2   4384.2 6735.8
## - age       1   4383.0 6736.5
## - levyplus  1   4383.3 6736.9
## - income    1   4385.0 6738.6
## - freepoor  1   4386.8 6740.4
## - sex       1   4388.0 6741.5
## - hscore    1   4389.1 6742.7
## - illness   1   4481.9 6835.4
## - actdays   1   4917.1 7270.7
##
## Step:  AIC=6734.53
## doctorco ~ sex + age + income + levyplus + freepoor + illness +
##     actdays + hscore + chcond
##
##            Df Deviance    AIC
## <none>          4381.0 6734.5
## - levyplus  1   4383.4 6735.0
## - chcond    2   4385.5 6735.0
## - income    1   4386.7 6738.2
## - age       1   4387.1 6738.7
## - freepoor  1   4389.1 6740.6
## - sex       1   4389.5 6741.0
## - hscore    1   4390.2 6741.8
## - illness   1   4482.7 6834.2
## - actdays   1   4917.6 7269.2
```

```
model.aic
```

```
##
## Call:  glm(formula = doctorco ~ sex + age + income + levyplus + freepoor +
##     illness + actdays + hscore + chcond, family = "poisson",
##     data = dvisits2)
##
## Coefficients:
## (Intercept)          sex          age       income     levyplus
##    -2.08906      0.16200      0.35513     -0.19981      0.08369
##    freepoor      illness      actdays       hscore      chcond1
##    -0.46960      0.18610      0.12661      0.03112      0.12110
##     chcond2
##     0.15889
##
## Degrees of Freedom: 5189 Total (i.e. Null);  5179 Residual
## Null Deviance:       5635
## Residual Deviance: 4381  AIC: 6735
```

Going by the coefficients we can say that **females**, **old**, **low-income**, **more ill** people tend to go to the doctor more often.

f

```r
lambda = predict(poisson_mdl, newdata = dvisits2[nrow(dvisits2),], type = "response")
prob = rep(0, 10)
prob[1] = exp(-lambda)
for(i in 1:9)
{
  prob[i + 1] = exp(-lambda)*(lambda ^ i)/prod(1:i)
}
round(prob, 3)
```

```
##  [1] 0.858 0.132 0.010 0.001 0.000 0.000 0.000 0.000 0.000 0.000
```

The last person has a probability of 0.858 of visiting the doctor 0 times.

Probability of 0.132 of visiting once.

Probability of 0.010 of visiting twice.

g

```r
predictDoctorVisits = function(lambda)
{
  prob = rep(0, 10)
  prob[1] = exp(-lambda)
  for(i in 1:9)
  {
    prob[i + 1] = exp(-lambda)*(lambda ^ i)/prod(1:i)
  }
  round(prob, 3)
  which(prob == max(prob))-1
}
```

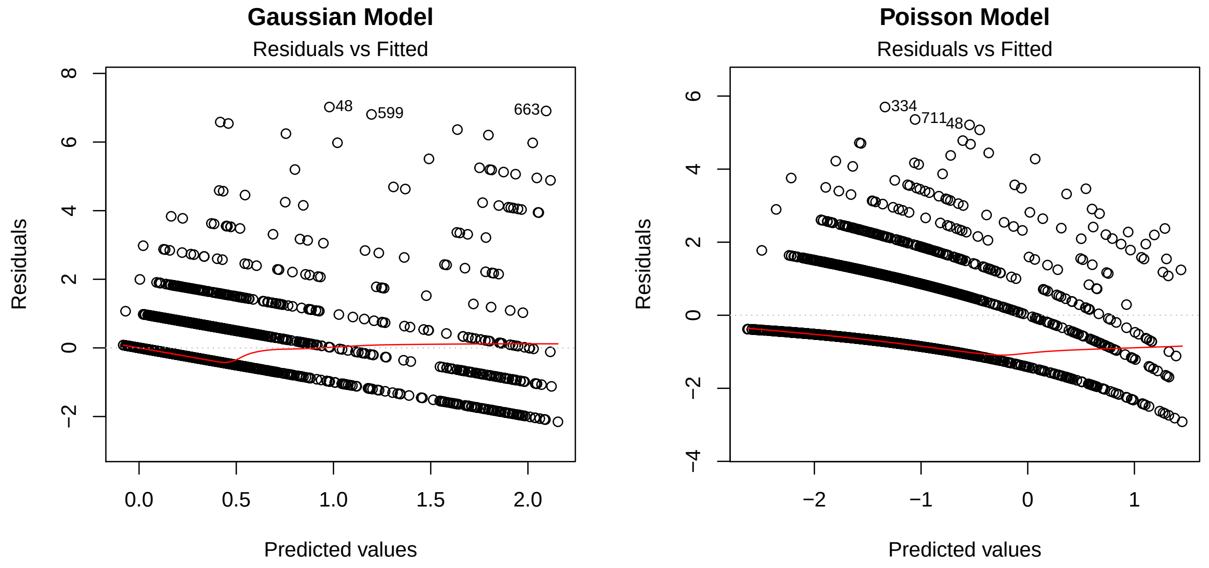
```r
preds = sapply(poisson_mdl$fitted.values, predictDoctorVisits)
pred_freq = as.data.frame(table(preds))
orig_freq = as.data.frame(table(dvisits2$doctorco))
orig_freq$PredictedFreq = c(pred_freq$Freq, rep(0, 5))
colnames(orig_freq) = c("doctorco", "OriginalFrequency", "PredictedFreq")
orig_freq
```

```
##    doctorco OriginalFrequency PredictedFreq
## 1         0              4141          4992
## 2         1               782           121
## 3          2                174              48
## 4          3                 30              26
## 5          4                 24               3
## 6          5                  9               0
## 7          6                 12               0
## 8          7                 12               0
## 9          8                  5               0
## 10         9                  1               0
```

From the table above the fit is not good at all for “doctorco” values above 4. Even for doctorco values 1 and 2 the fit is not quite compelling. So I would say that this is not a very good fit.

h

```
gaussian_mdl = glm(formula = doctorco ~ sex + age + income + levyplus + freepoor +
                     illness + actdays + hscore + chcond, family = "gaussian",
                   data = dvisits2)

par(mfrow = c(1,2))
plot(gaussian_mdl, which = 1, main = "Gaussian Model")
plot(poisson_mdl, which = 1, main = "Poisson Model")
```



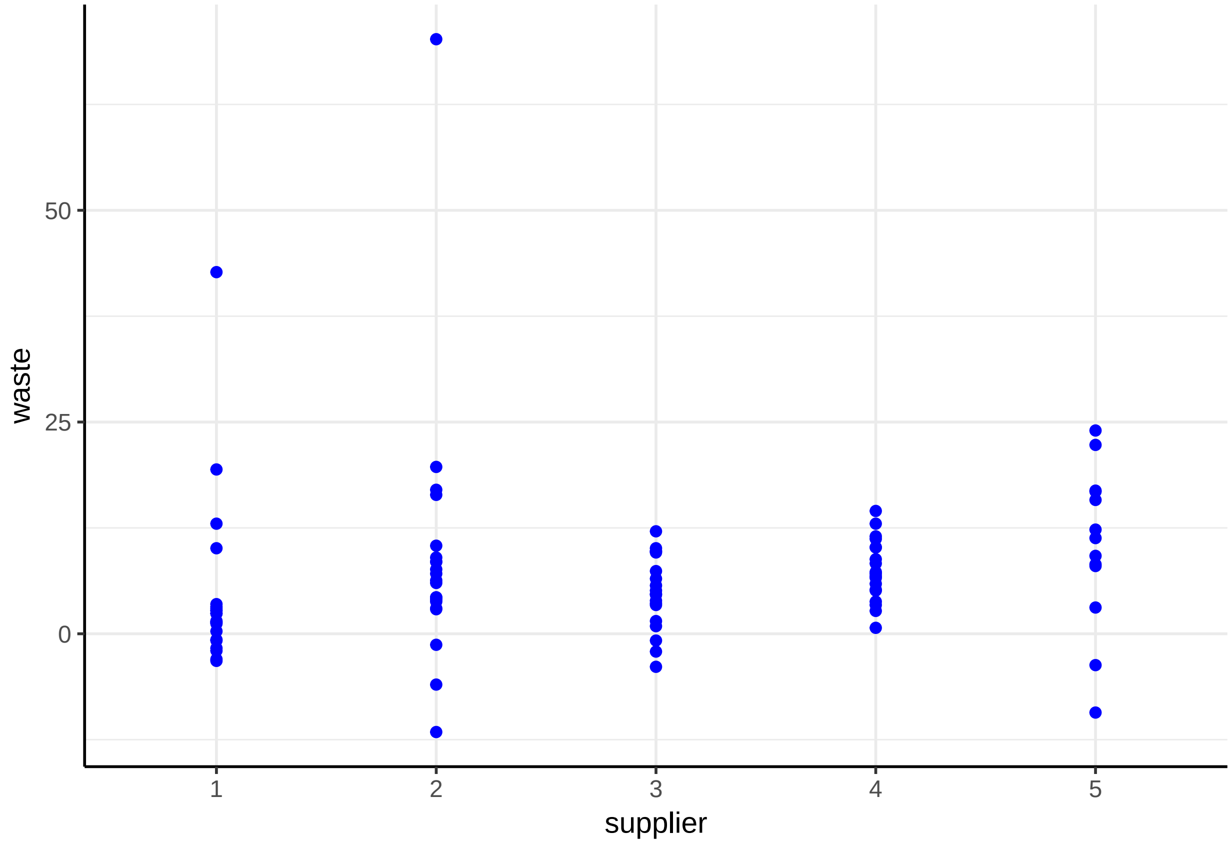
Even in Gaussian the predicted values do not go beyond 2 so we can say that even this is nor a good fit for the given data. But comparatively Gaussian family fits the data better than the Poisson model, which can be seen clearly from the distance between the residual lines. THe residuals are close to the 0 mark which is not the case in Poisson model.

2

a

Plot the data

```r
ggplot(data = denim, aes(x = supplier, y = waste)) +
  geom_point(color = "blue") +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line())
```



Supplier 1 is having waste close to 0 since mostly the data is accumulated near 0. Suppliers 2 and 5 have high variance in the waste plot. Supplier 3 and 4 waste ranges are almost similar although the mean of 3 is lower than that of 4.

It looks like the waste is having its own distribution in each supplier. There is a possibility of random effect.

b

Linear fixed effect model

```r
denim.fixed = lm(waste ~ supplier, denim)
summary(denim.fixed)
```

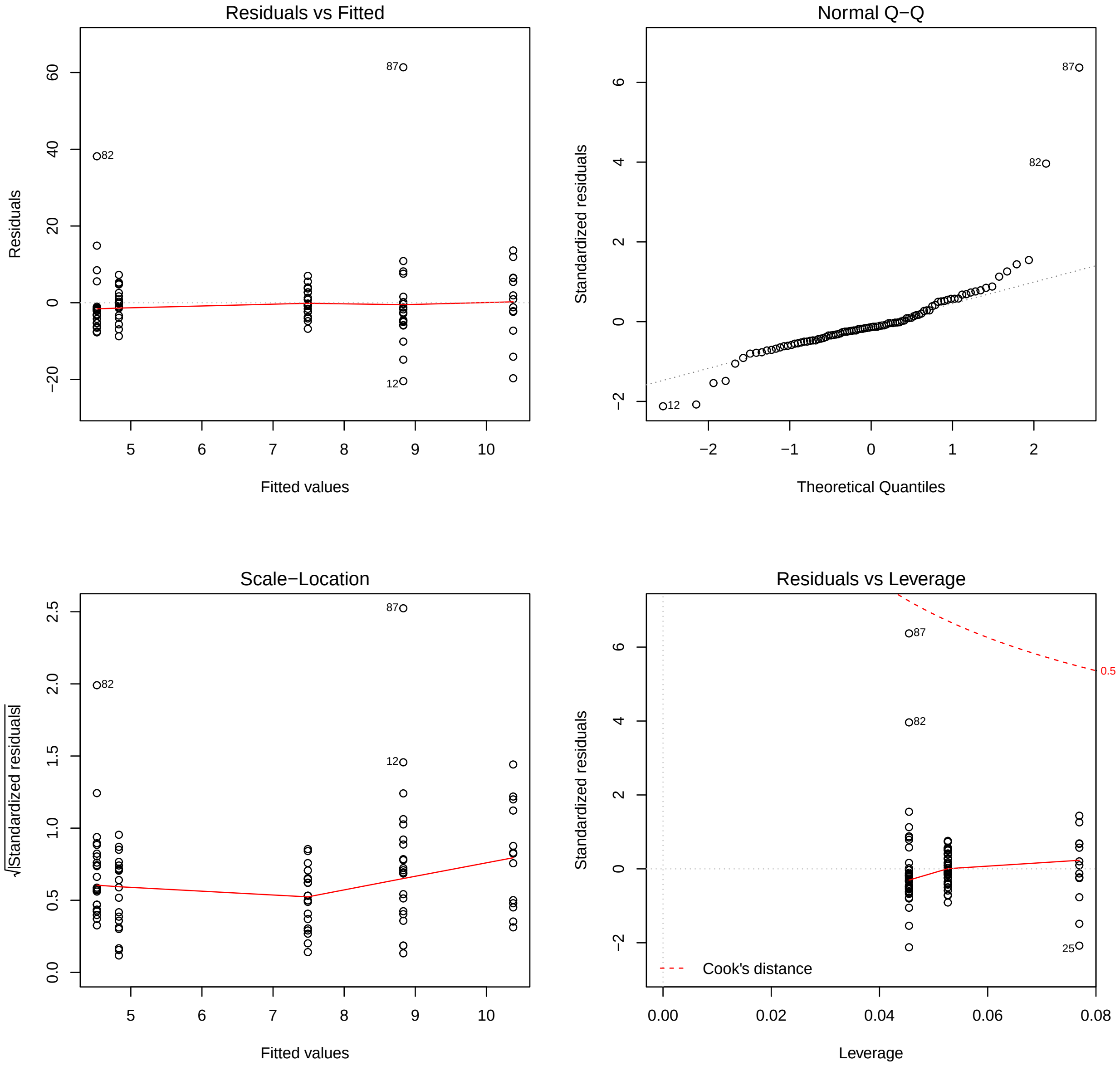
```
##
## Call:
## lm(formula = waste ~ supplier, data = denim)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.432  -4.377  -1.323   2.639  61.368
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.5227     2.1021   2.152   0.0341 *
## supplier2     4.3091     2.9728   1.450   0.1507
## supplier3     0.3089     3.0879   0.100   0.9206
## supplier4     2.9667     3.0879   0.961   0.3392
## supplier5     5.8542     3.4491   1.697   0.0931 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.86 on 90 degrees of freedom
## Multiple R-squared:  0.04901,    Adjusted R-squared:  0.006747
## F-statistic:  1.16 on 4 and 90 DF,  p-value: 0.334
```

The operator is not significant at all in a fixed effect model. The $p-values$ show that non of the Suppliers were significant. The R^2 value is also extremely low. So we can conclude that the fit was a bad fit and the operator is not significant.

c

```r
par(mfrow = c(2,2))
plot(denim.fixed)
```



- The Residual-FittedValue look fine with the constant variance.
- The QQ plot shows us that the observations errors are not normally distributed
- There are no outliers.

d

```
denim.random = lmer(waste ~ (1|supplier), denim)
summary(denim.random)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: waste ~ (1 | supplier)
##    Data: denim
##
```

```
## REML criterion at convergence: 702.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9095 -0.4363 -0.1669  0.3142  6.3817
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  supplier (Intercept)  0.6711  0.8192
##  Residual             97.3350  9.8658
## Number of obs: 95, groups:  supplier, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    6.997      1.078    6.49
```

$\hat{\sigma_\alpha}^2 = 0.8192$ and $\hat{\sigma}^2 = 9.8658$

e

Using parametric bootstrapping

```
nullmod <- lm(waste ~ 1, data=denim)
llrts <- as.numeric(2 * (logLik(denim.random) - logLik(nullmod)))

lrstats <- numeric(10000)
for(i in 1:10000)
{
  y <- unlist(simulate(nullmod))
  nullsim <- lm(y ~ 1)
  altsim <- lmer(waste ~ (1|supplier), data=denim, REML=FALSE)
  lrstats[i] <- as.numeric(2 * (logLik(altsim) - logLik(nullsim)))
}

pval <- mean(lrstats >= llrts)
pval
```

```
## [1] 0.4239
```

```
se.pval <- sqrt(pval*(1-pval)/10000)
se.pval
```

```
## [1] 0.004941749
```

The $p - value$ is way below the significance level of 0.05 so we cannot say that it is not significant.

f

Confidence Interval for the random effect standard deviation.

```
confint(denim.random)
```

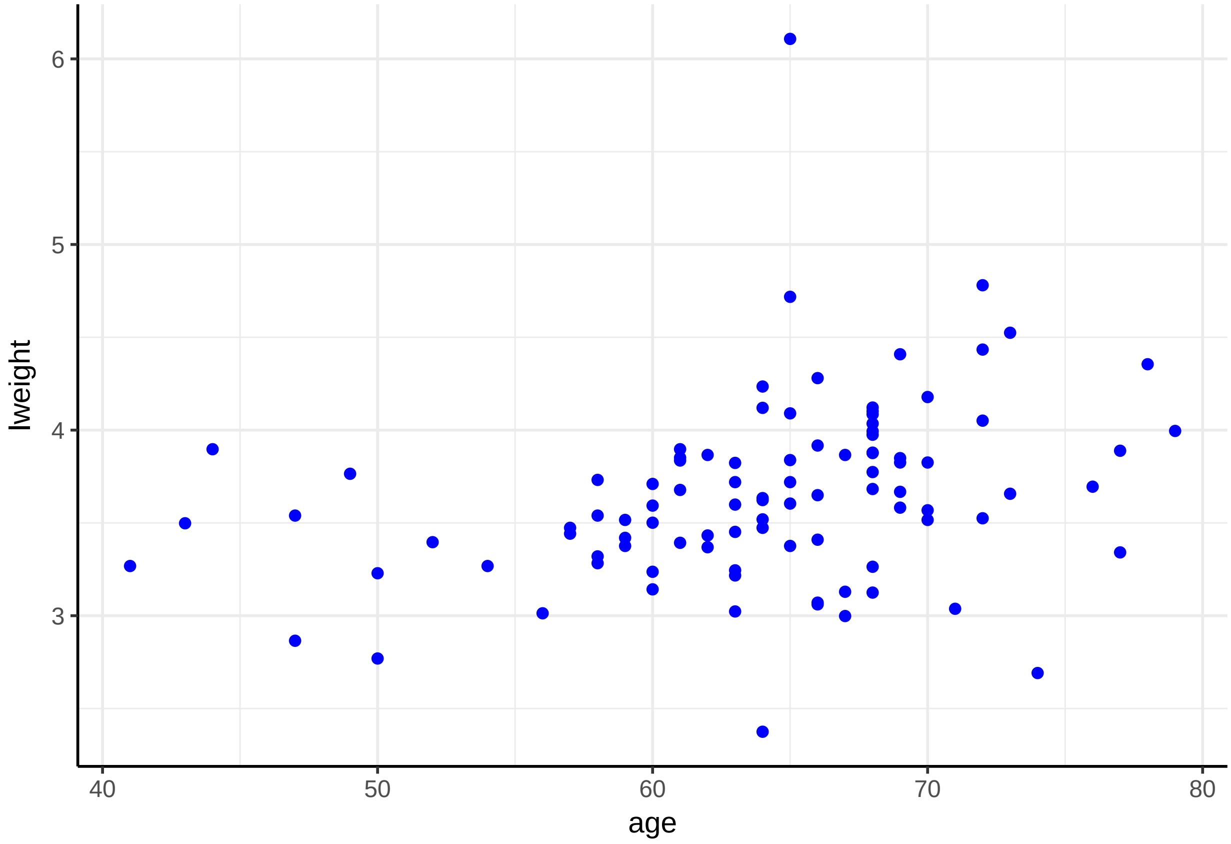
```
## Computing profile confidence intervals ...
```

```
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep
## Warning in FUN(X[[i]], ...): non-monotonic profile for .sig01
## Warning in if (parm == "theta_") {: the condition has length > 1 and only
## the first element will be used
## Warning in if (parm == "beta_") {: the condition has length > 1 and only
## the first element will be used
## Warning in confint.thpr(pp, level = level, zeta = zeta): bad spline fit
## for .sig01: falling back to linear interpolation
##                2.5 %    97.5 %
## .sig01      0.000000  4.206374
## .sigma      8.591567 11.424996
## (Intercept) 4.977790  8.975895
```

3

a

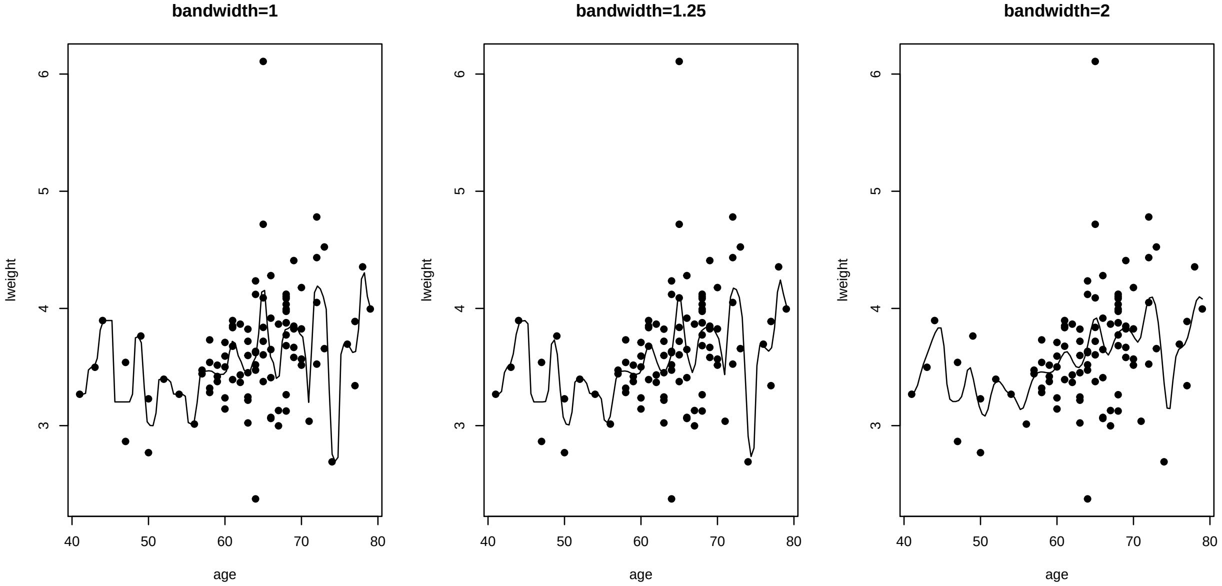
```
ggplot(data = prostate, aes(x = age, y = lweight)) +
  geom_point(color = "blue") +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line())
```



The data looks like a U-shaped curve but we can also see the presence of some outliers. There is also a funneling effect towards the high values of age.

b

```
par(mfrow = c(1,3))

plot(lweight ~ age,
     prostate,main="bandwidth=1", pch=19)
lines(ksmooth(prostate$age,prostate$lweight,"normal",1))

plot(lweight ~ age,
     prostate,main="bandwidth=1.25", pch=19)
lines(ksmooth(prostate$age,prostate$lweight,"normal",1.25))

plot(lweight ~ age,
     prostate,main="bandwidth=2", pch=19)
lines(ksmooth(prostate$age,prostate$lweight,"normal",2))
```

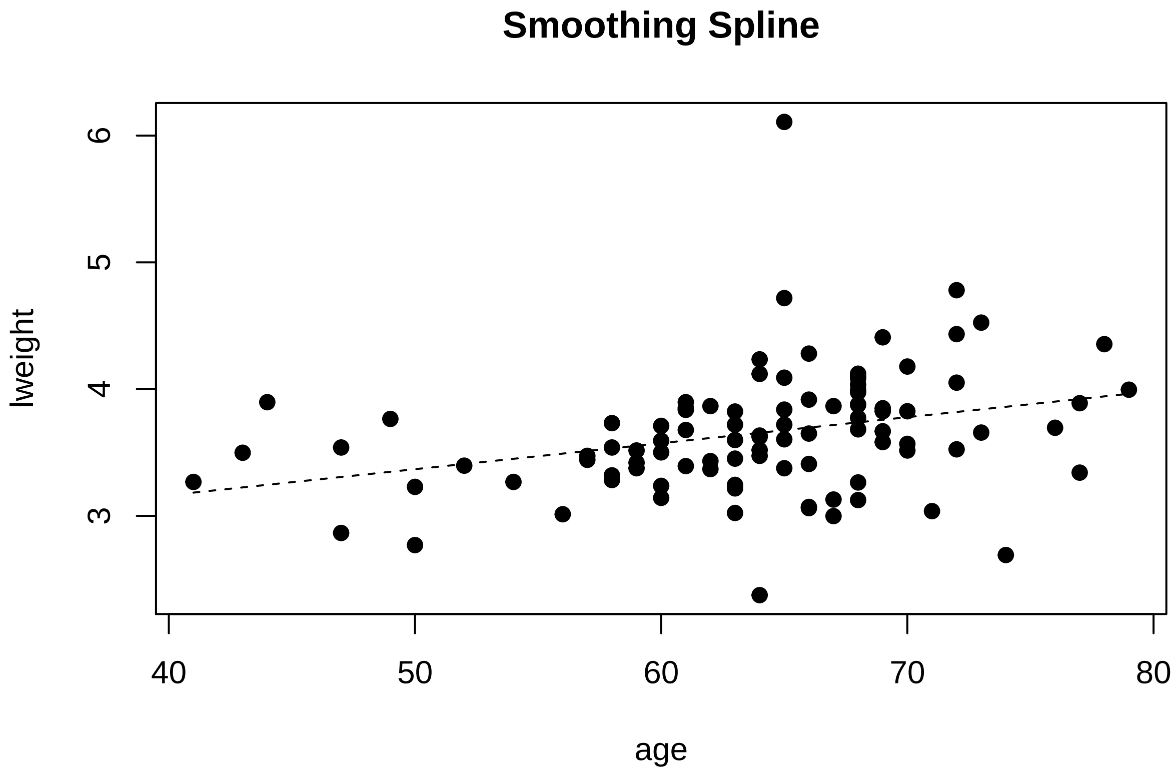


Plot 3(bandwidth = 2) looks the best. I think the fit was using "normal" kernel and bandwidth=2.

The outliers pull the curve towards itself. As we can see at around age~75 the curve dips due to an outlier and similarly the outlier having highest lweight at age around 65, the curve tends upward inspite of many points lying below it.

c

```
par(mfrow = c(1,1))

smspl = smooth.spline(prostate$age,prostate$lweight)
plot(lweight ~ age,
     prostate,main="Smoothing Spline", pch=19)
lines(smspl, lty=2)
```



What type of curve has been fit to the data?

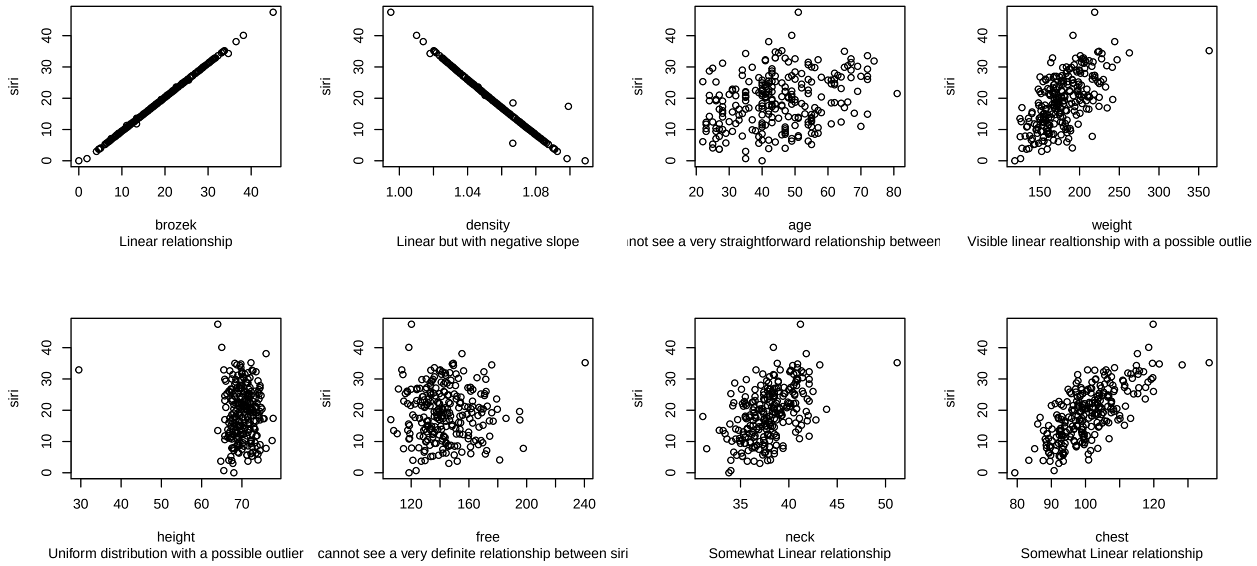
```
smspl
```

```
## Call:
## smooth.spline(x = prostate$age, y = prostate$lweight)
##
## Smoothing Parameter  spar= 1.499836  lambda= 4341.236 (28 iterations)
## Equivalent Degrees of Freedom (Df): 2.000035
## Penalized Criterion (RSS): 7.547608
## GCV: 0.2304165
```

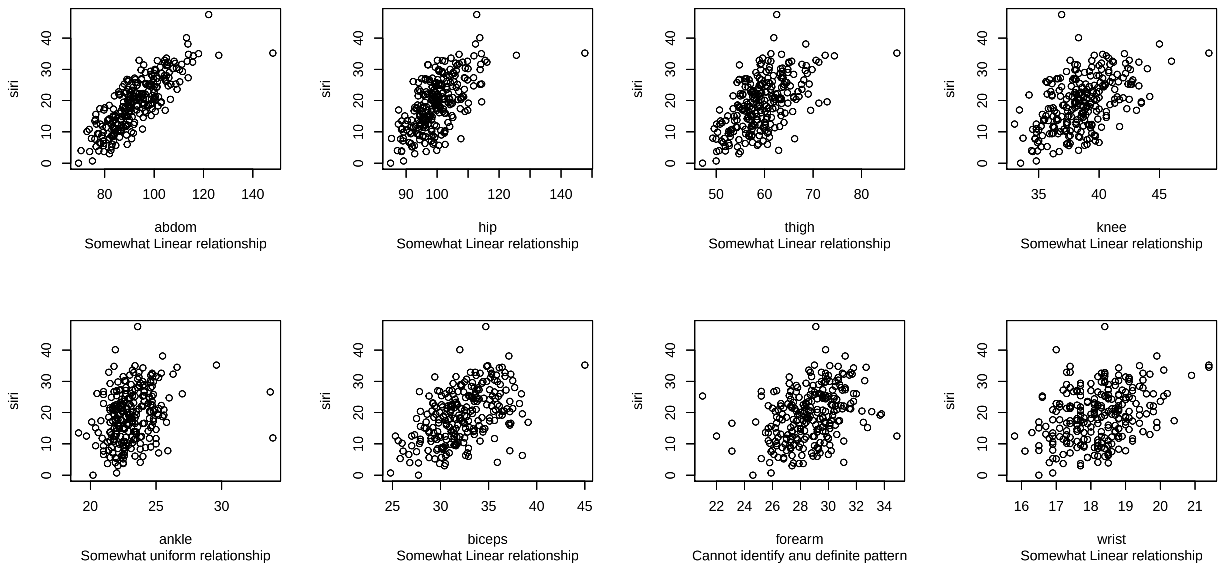
4

a

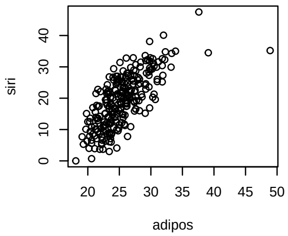
```r
par(mfrow = c(2,4))
plot(siri~brozek, data = fat,
sub = "Linear relationship")

plot(siri~density, data = fat,
sub = "Linear but with negative slope")

plot(siri~age, data = fat,
sub = "We cannot see a very straightforward relationship between siri and age")

plot(siri~weight, data = fat,
sub = "Visible linear realtionship with a possible outlier")

plot(siri~height, data = fat,
sub = "Uniform distribution with a possible outlier")

plot(siri~free, data = fat,
sub = "We cannot see a very definite relationship between siri and age")

plot(siri~neck, data = fat,
sub = "Somewhat Linear relationship")

plot(siri~chest, data = fat,
sub = "Somewhat Linear relationship")
```



```r
plot(siri~abdom, data = fat,
sub = "Somewhat Linear relationship")

plot(siri~hip, data = fat,
sub = "Somewhat Linear relationship")

plot(siri~thigh, data = fat,
sub = "Somewhat Linear relationship")

plot(siri~knee, data = fat,
sub = "Somewhat Linear relationship")

plot(siri~ankle, data = fat,
sub = "Somewhat uniform relationship")

plot(siri~biceps, data = fat,
sub = "Somewhat Linear relationship")

plot(siri~forearm, data = fat,
sub = "Cannot identify anu definite pattern")

plot(siri~wrist, data = fat,
sub = "Somewhat Linear relationship")
```



```
plot(siri~adipos, data = fat,
sub = "Somewhat Linear relationship")
```



Somewhat Linear relationship

Overall most of the predictors were linearly related to the response variable Siri

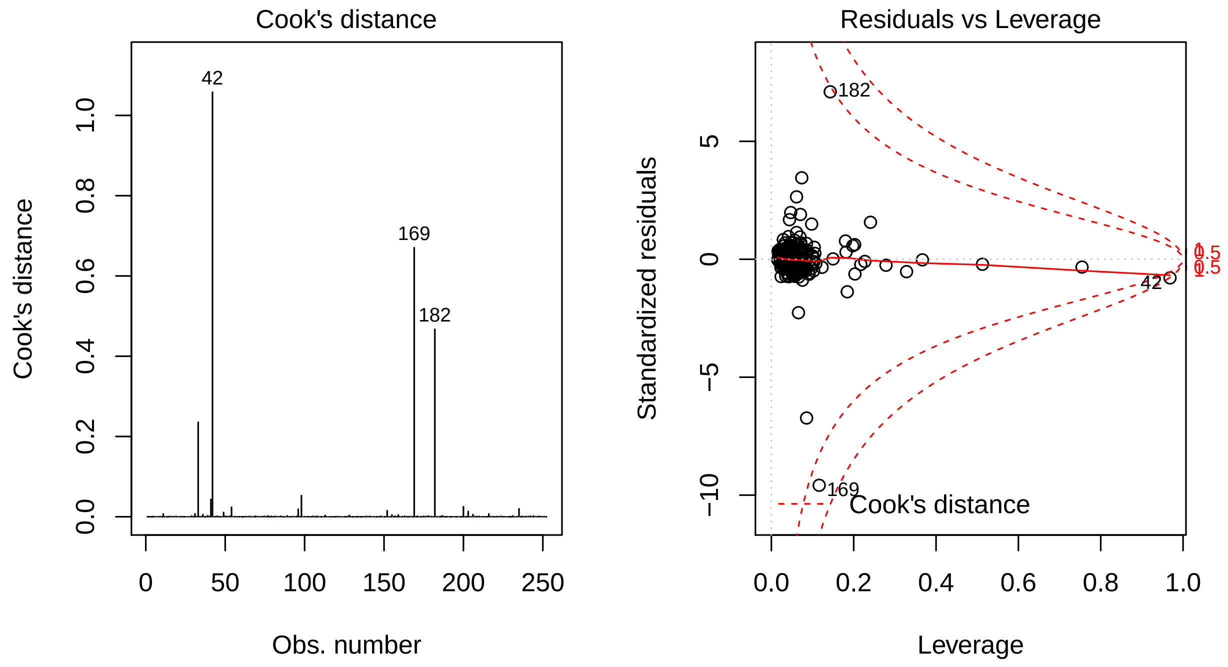
b

```
lm.mdl = lm(siri~., data=fat)
summary(lm.mdl)
```

```
## 
## Call:
## lm(formula = siri ~ ., data = fat)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.69918 -0.04597 -0.00371  0.04846  1.23949 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -3.5728847  4.6889514  -0.762  0.44684    
## brozek       1.0857047  0.0136068  79.792  < 2e-16 ***
## density      1.7409797  4.2015371   0.414  0.67898    
## age          0.0008217  0.0014295   0.575  0.56597    
## weight      -0.0031903  0.0040431  -0.789  0.43087    
## height       0.0011135  0.0049123   0.227  0.82087    
## adipos       0.0079147  0.0138286   0.572  0.56764    
## free         0.0017816  0.0049430   0.360  0.71885    
## neck         0.0002322  0.0104224   0.022  0.98225    
## chest        0.0003290  0.0047575   0.069  0.94492    
## abdom       -0.0006778  0.0048890  -0.139  0.88986    
## hip          0.0057731  0.0065127   0.886  0.37630    
## thigh       -0.0149058  0.0066039  -2.257  0.02492 *  
## knee         0.0288436  0.0108816   2.651  0.00858 ** 
## ankle       -0.0029944  0.0099029  -0.302  0.76263    
## biceps       0.0164437  0.0076484   2.150  0.03258 *  
## forearm     -0.0129444  0.0089614  -1.444  0.14995
## wrist       -0.0302412  0.0241338  -1.253  0.21143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1886 on 234 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 2.906e+04 on 17 and 234 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(1,2))
plot(lm.mdl, which=4)
plot(lm.mdl, which=5)
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



The following influential points were identified from the above plot

```
knitr::kable(fat[42,])
```

	brozek	siri	density	age	weight	height	adipos	free	neck	chest	abdom	hip	thigh	knee	ankl
42	31.7	32.9	1.025	44	205	29.5	29.9	140.1	36.6	106	104.3	115.5	70.6	42.5	23.

Height = 29.5 is the value that has affected this data point to make it an outlier.

```
lm.mdlmod = lm(siri~., data=fat[-42,])
summary(lm.mdlmod)
```

```
##
```

```
## Call:
## lm(formula = siri ~ ., data = fat[-42, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.68399 -0.04725 -0.00386  0.04450  1.24777
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.0691084  5.0650884  -0.409  0.68328
## brozek       1.0882402  0.0139919  77.777  < 2e-16 ***
## density      1.6058748  4.2084171   0.382  0.70312
## age          0.0007490  0.0014336   0.522  0.60185
## weight      -0.0006910  0.0051391  -0.134  0.89316
## height      -0.0191409  0.0261401  -0.732  0.46476
## adipos      -0.0181909  0.0358681  -0.507  0.61252
## free         0.0033720  0.0053420   0.631  0.52850
## neck        -0.0005412  0.0104768  -0.052  0.95885
## chest        0.0004452  0.0047636   0.093  0.92562
## abdom       -0.0005987  0.0048940  -0.122  0.90274
## hip          0.0064669  0.0065771   0.983  0.32650
## thigh       -0.0158557  0.0067180  -2.360  0.01909 *
## knee         0.0290321  0.0108930   2.665  0.00823 **
## ankle       -0.0025512  0.0099268  -0.257  0.79740
## biceps       0.0160203  0.0076734   2.088  0.03790 *
## forearm     -0.0125699  0.0089812  -1.400  0.16297
## wrist       -0.0308242  0.0241646  -1.276  0.20337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1887 on 233 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 2.87e+04 on 17 and 233 DF,  p-value: < 2.2e-16
```

Most significant = brozek

$R^2 = 0.9995$

c

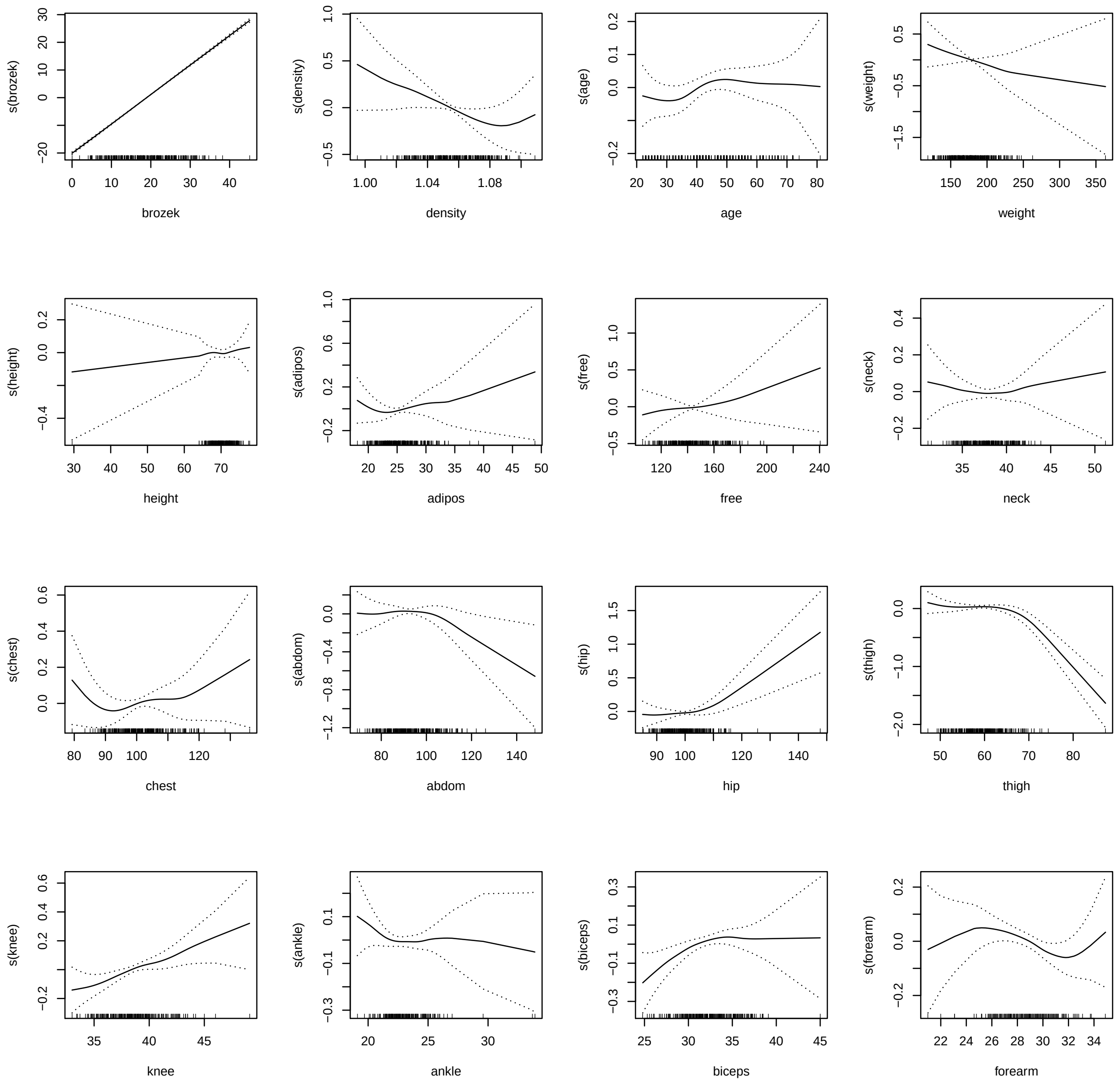
```
gam.mdl = gam(siri ~ s(brozek) + s(density) + s(age) + s(weight) + s(height) + s(adipos) +
                s(free) + s(neck) + s(chest) + s(abdom) + s(hip) + s(thigh) + s(knee) +
                s(ankle) + s(biceps) + s(forearm) + s(wrist),
              data = fat)
summary(gam.mdl)
```

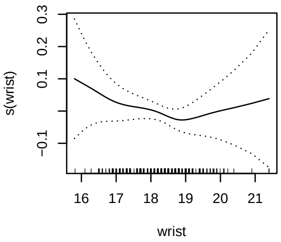
```
##
## Call: gam(formula = siri ~ s(brozek) + s(density) + s(age) + s(weight) +
##     s(height) + s(adipos) + s(free) + s(neck) + s(chest) + s(abdom) +
##     s(hip) + s(thigh) + s(knee) + s(ankle) + s(biceps) + s(forearm) +
##     s(wrist), data = fat)
## Deviance Residuals:
##        Min         1Q     Median         3Q        Max
## -1.1029207 -0.0492398  0.0009844  0.0459799  0.5446793
```

```
##
## (Dispersion Parameter for gaussian family taken to be 0.0264)
##
##     Null Deviance: 17578.99 on 251 degrees of freedom
## Residual Deviance: 4.8259 on 183.0003 degrees of freedom
## AIC: -141.6248
##
## Number of Local Scoring Iterations: 12
##
## Anova for Parametric Effects
##             Df  Sum Sq Mean Sq    F value    Pr(>F)
## s(brozek)    1 17565.7 17565.7 6.6610e+05 < 2.2e-16 ***
## s(density)   1     0.1     0.1 2.8873e+00  0.090982 .
## s(age)       1     0.1     0.1 4.6737e+00  0.031927 *
## s(weight)    1     0.1     0.1 3.8217e+00  0.052115 .
## s(height)    1     0.0     0.0 9.0300e-02  0.764131
## s(adipos)    1     0.0     0.0 3.3000e-03  0.954452
## s(free)      1     0.0     0.0 1.1330e-01  0.736849
## s(neck)      1     0.0     0.0 1.1533e+00  0.284266
## s(chest)     1     0.0     0.0 3.0060e-01  0.584162
## s(abdom)     1     0.0     0.0 1.5920e-01  0.690361
## s(hip)       1     0.1     0.1 3.3997e+00  0.066823 .
## s(thigh)     1     0.1     0.1 2.6641e+00  0.104355
## s(knee)      1     0.2     0.2 7.1584e+00  0.008138 **
## s(ankle)     1     0.0     0.0 7.0230e-01  0.403099
## s(biceps)    1     0.0     0.0 1.8412e+00  0.176477
## s(forearm)   1     0.1     0.1 3.9664e+00  0.047904 *
## s(wrist)     1     0.0     0.0 6.6180e-01  0.416973
## Residuals  183     4.8     0.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##             Npar Df  Npar F     Pr(F)
## (Intercept)
## s(brozek)         3  1.8129  0.146371
## s(density)        3  4.4620  0.004737 **
## s(age)            3  1.3616  0.256012
## s(weight)         3  1.2335  0.298944
## s(height)         3  0.2772  0.841836
## s(adipos)         3  2.6382  0.051033 .
## s(free)           3  1.4805  0.221341
## s(neck)           3  0.5660  0.638072
## s(chest)          3  3.3870  0.019295 *
## s(abdom)          3  3.6704  0.013331 *
## s(hip)            3  5.5165  0.001196 **
## s(thigh)          3 21.0078 9.691e-12 ***
## s(knee)           3  0.6066  0.611533
## s(ankle)          3  0.9240  0.430356
## s(biceps)         3  1.9551  0.122342
## s(forearm)        3  1.9420  0.124391
## s(wrist)          3  1.1663  0.323981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To identify the influential points it is very important to look at the plots of transformation vs variable. This plot shows us the transformed variable on the y axis and the variable asis on the x-axis. Using these plots we can identify the outliers since we will notice a change in pattern on the curve in case of outliers.

```r
par(mfrow = c(2,4))
plot(gam.mdl, se = TRUE)
```





Based on the plots we can identify the following criterias and the corresponding observations satisfying these criterias. It can be seen quite easily that at the following regions the curves show some abnormalities/ abnormal turns.

```
fat[fat$hip > 130, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 39   33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2 148.1
##      hip thigh knee ankle biceps forearm wrist
## 39 147.7  87.3 49.1  29.6     45      29  21.4
```

```
fat[fat$thigh > 80, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 39   33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2 148.1
##      hip thigh knee ankle biceps forearm wrist
## 39 147.7  87.3 49.1  29.6     45      29  21.4
```

```
fat[fat$ankle > 30, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 31   12.3 11.9  1.0716  32    182  73.75   23.6 159.7 38.7 100.5  88.7
## 86   25.8 26.6  1.0386  67    167  67.50   26.0 123.9 36.5  98.9  89.7
##     hip thigh knee ankle biceps forearm wrist
## 31 99.8  57.5 38.7  33.9   32.5    27.7  18.4
## 86 96.2  54.7 37.8  33.7   32.4    27.7  18.2
```

```
fat[fat$biceps > 40, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 39   33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2 148.1
##      hip thigh knee ankle biceps forearm wrist
## 39 147.7  87.3 49.1  29.6     45      29  21.4
```

```
fat[fat$weight > 300, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 39   33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2 148.1
##      hip thigh knee ankle biceps forearm wrist
## 39 147.7  87.3 49.1  29.6     45      29  21.4
```

```
fat[fat$height < 60, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 42   31.7 32.9   1.025  44    205   29.5    29.9 140.1 36.6   106 104.3
##      hip thigh knee ankle biceps forearm wrist
## 42 115.5  70.6 42.5  23.7   33.6    28.7  17.4
```

```
fat[fat$adipos > 40, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 39   33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2 148.1
##      hip thigh knee ankle biceps forearm wrist
## 39 147.7  87.3 49.1  29.6     45      29  21.4
```

```
fat[fat$neck > 45, ]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
## 39   33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2 148.1
##      hip thigh knee ankle biceps forearm wrist
## 39 147.7  87.3 49.1  29.6     45      29  21.4
```

The observations we can conclude as influential are : **39, 42, 86, 31**.

It can be seen quite easily that at the following regions where the values of these variables reach such extremes the curves show some abnormalities/abnormal turns.

e

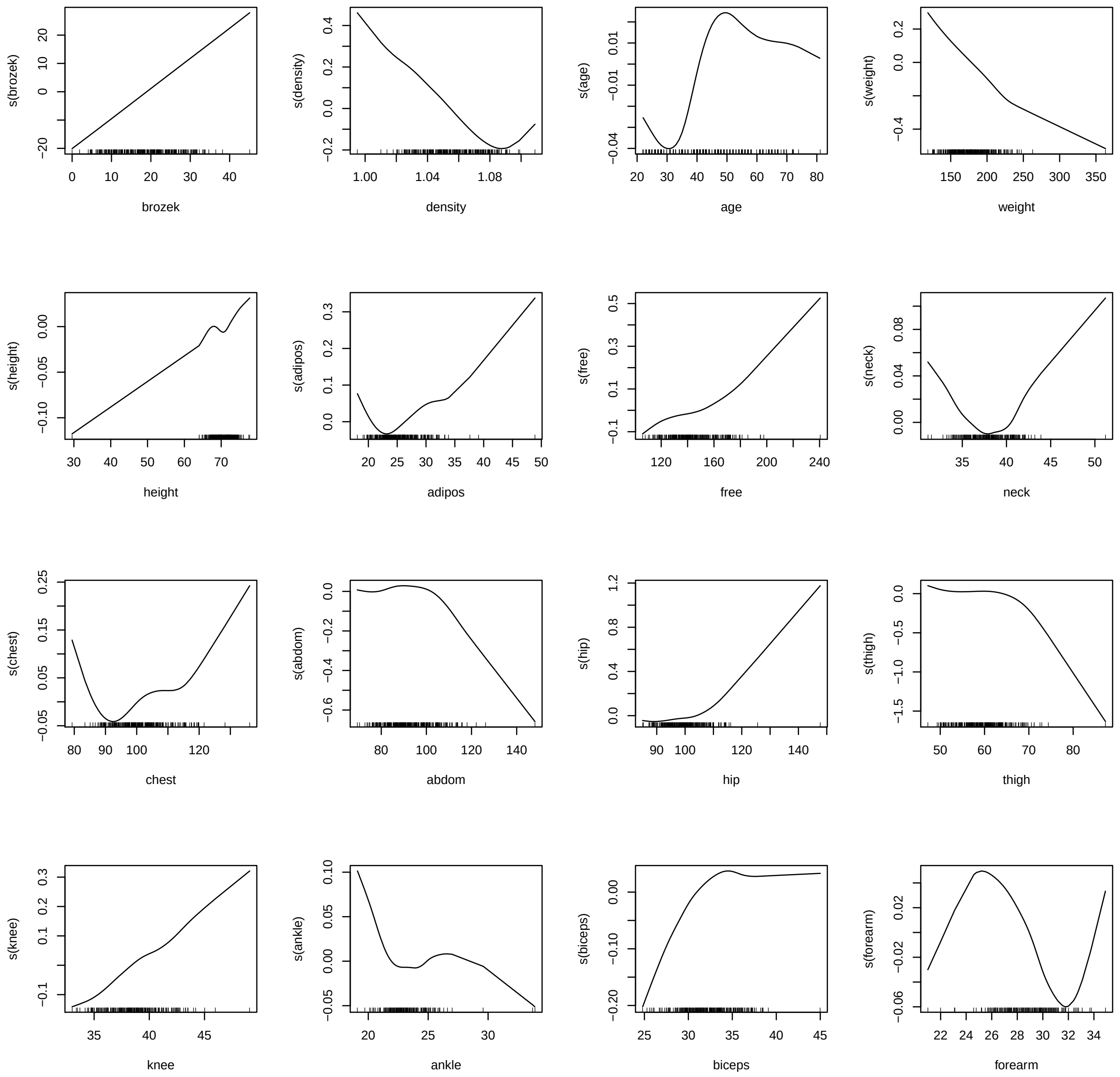
The R^2 for this additive model can be found as follows

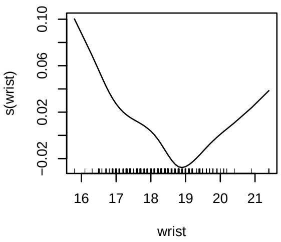
```
nullmdl = lm(siri ~ 1, data = fat)
TSS = sum(nullmdl$residuals^2)
RSS = sum(gam.mdl$residuals^2)
r_square = 1-(RSS/TSS)
r_square
```

```
## [1] 0.9997255
```

The R^2 of the additive model is 0.9997 compared to the R^2 of the linear model as 0.9995. The difference as we can see is not very high infact it is as low as 0.0002. But the complexity we have included in the model just to achieve this much improvement in R^2 is not worth it. So I would disagree that the additive model is a better model.

```
par(mfrow = c(2,4))
plot(gam.mdl)
```


s(brozek)
−20
0
10
20
0
10
20
30
40
brozek
s(density)
−0.2
0.0
0.2
0.4
1.00
1.04
1.08
density
s(age)
−0.04
−0.01
0.01
20
30
40
50
60
70
80
age
s(weight)
−0.4
0.0
0.2
150
200
250
300
350
weight
s(height)
−0.10
−0.05
0.00
30
40
50
60
70
height
s(adipos)
0.0
0.1
0.2
0.3
20
25
30
35
40
45
50
adipos
s(free)
−0.1
0.1
0.3
0.5
120
160
200
240
free
s(neck)
0.00
0.04
0.08
35
40
45
50
neck
s(chest)
−0.05
0.05
0.15
0.25
80
90
100
120
chest
s(abdom)
−0.6
−0.4
−0.2
0.0
80
100
120
140
abdom
s(hip)
0.0
0.4
0.8
1.2
90
100
120
140
hip
s(thigh)
−1.5
−1.0
−0.5
0.0
50
60
70
80
thigh
s(knee)
−0.1
0.1
0.2
0.3
35
40
45
knee
s(ankle)
−0.05
0.00
0.05
0.10
20
25
30
ankle
s(biceps)
−0.20
−0.10
0.00
25
30
35
40
45
biceps
s(forearm)
−0.06
−0.02
0.02
22
24
26
28
30
32
34
forearm

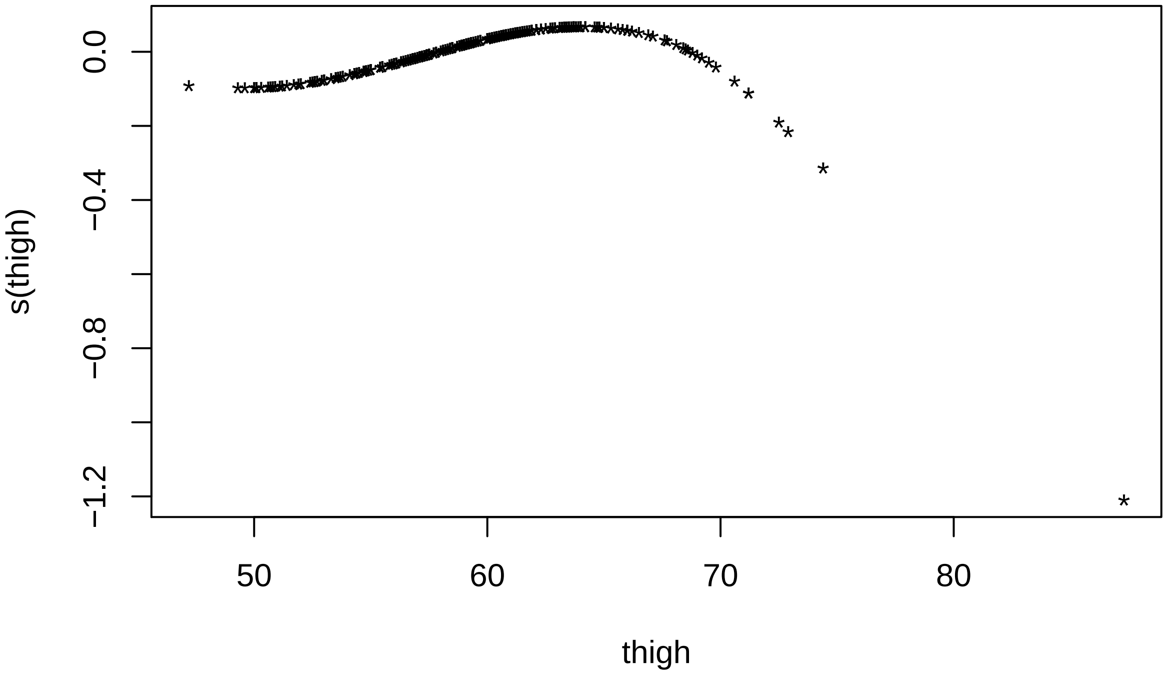


s(thigh) is the predictor that makes the most nonlinear contribution towards the resonse.

This predictor can be made a linear predictor. Mainly this non-linearity is due to the presence of high influencial points in the data.

f

```r
plot(fat$thigh ,gam.mdl$smooth[, "s(thigh)"], xlab = "thigh", ylab = "s(thigh)", pch = "*")
```



The non linearity looks like a threshold kind of a curve, wherein the curve drops drastically after a certain point and before that it is a smooth curve. The plot looks like increasing at first till around 65 but after that it takes a sudden change and dips drastically. The change in direction in such a short range cannot be captured by a linear relation thus this non linear transformation is necessary.