653HW1

Zixuan Yu

2/7/2022

Due on 02/11/2022

I. Interpreting Simple and Multiple Linear Regression Coefficients

- 1. Using only the data from the first measurement time for each child, plot weight against age as if for an international nutrition journal. Label the axes clearly and make sure that all observations can be seen. Jitter the data or use different levels of transparency as necessary. Use different colors for the plotting symbols for boys and girls. Add a smooth curve (e.g. natural spline with ~3 degrees of freedom or loess with span =0.5 or kernel smoother with bandwidth 20 months) to the plot to emphasize the relationship of the observed mean weight at each age without making a stronger parametric assumption (e.g. linearity). Familiarize yourself with how each of these smoothers works. Now make the curves separately for boys and girls.

```
library(RColorBrewer)
library(ggplot2)
library(splines)
library(dplyr)
```

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

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

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

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.1.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

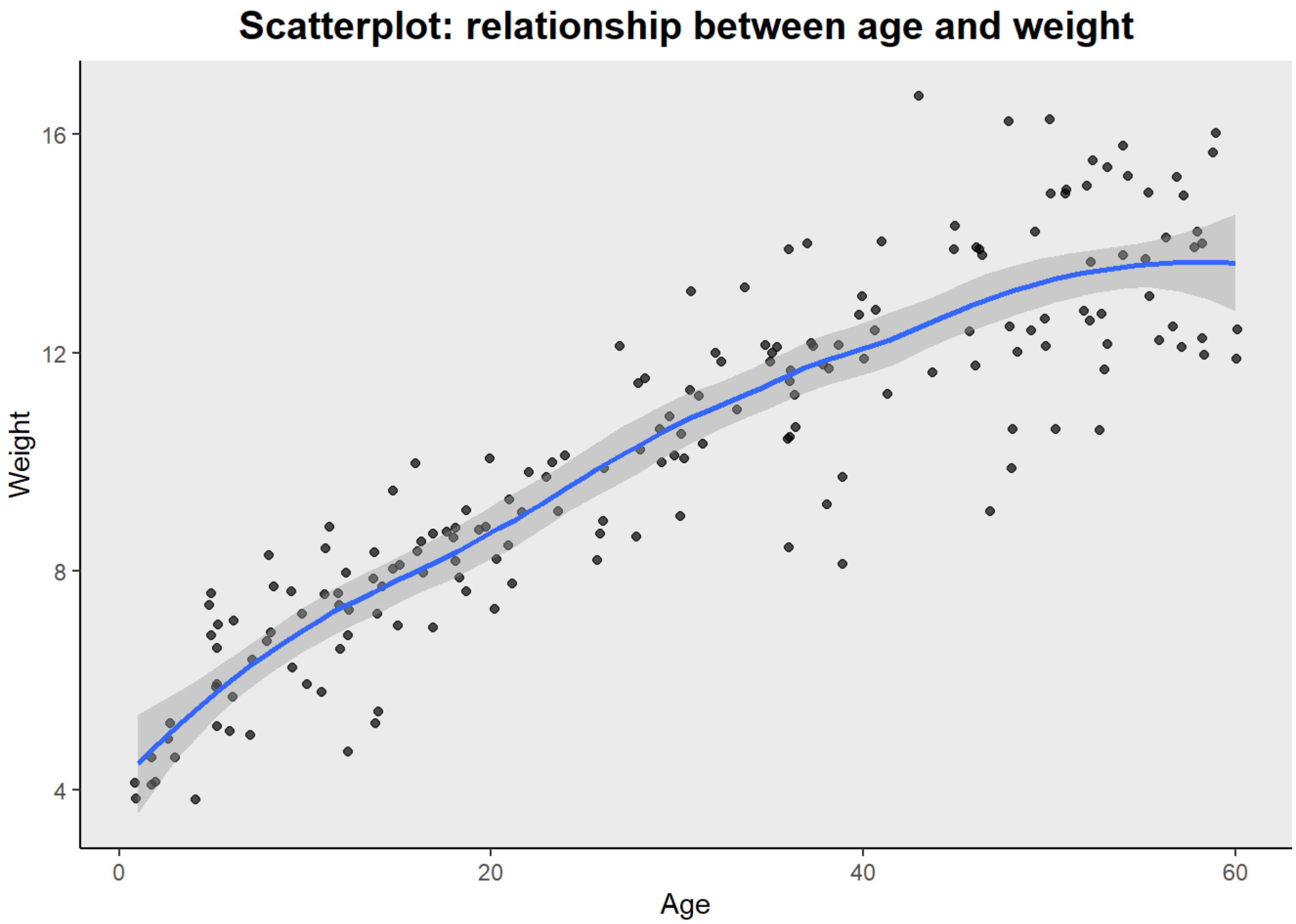
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
load("C:/Users/19092/OneDrive - Johns Hopkins/Term3/Biostat653/Data/NepalAnthroZip/nepal.anthro.rdata")
d1 <- nepal.anthro[nepal.anthro$num==1, c("sex", "wt", "ht", "age")]  # select first observation for each child and desired variables
d <- d1[complete.cases(d1),] # drop cases without one or more of these variables
d <- d[order(d$age),] # reorder the dataframe to increasing age for later plotting
```

Scatter Plot for the relationship between age and sex.

```
d %>% ggplot(aes(x=age, y=wt)) +
    geom_jitter(alpha = 0.7) +
    geom_smooth(method = "loess", span = 0.5) +
    labs(title = "Scatterplot: relationship between age and weight", x = "Age", y = "Weight") +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 15, face = "bold", hjust = 0.5))
```

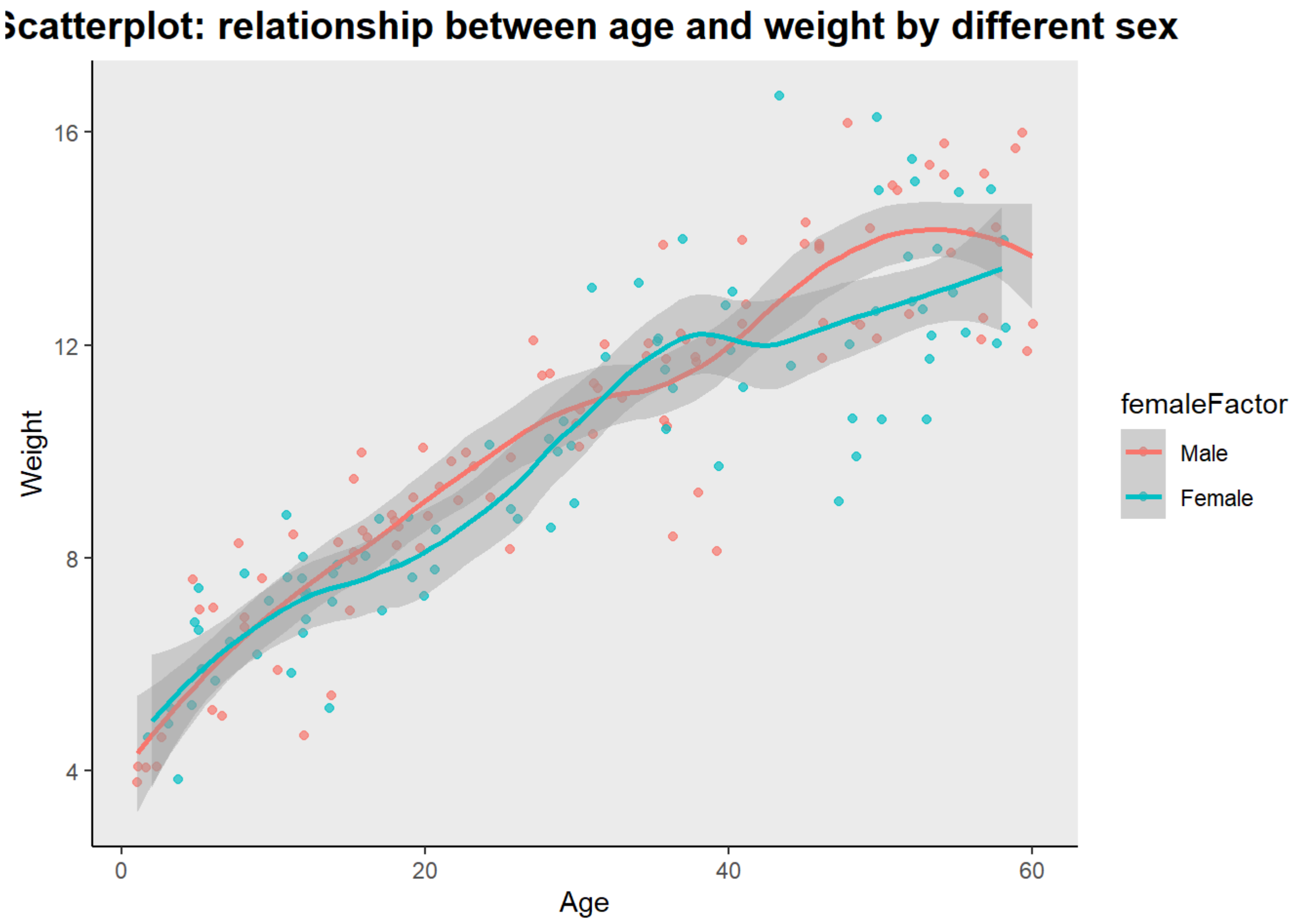
```
## `geom_smooth()` using formula 'y ~ x'
```



Scatter Plot for the relationship between age and weight, make the curves separately for boys and girls.

```
# Sex is coded as 2=Female, 1=Male
# recode sex with Female=1, Male=0
d <- d %>% mutate(femaleFactor= case_when(sex==2 ~ 1,
                                     sex==1 ~ 0))
d$femaleFactor=factor(d$femaleFactor,levels = c(0,1), labels = c("Male","Female"))
d %>% ggplot(aes(x=age, y=wt, color= femaleFactor)) +
    geom_jitter(alpha = 0.7) +
    geom_smooth(method = "loess", span = 0.5) +
    labs(title = "Scatterplot: relationship between age and weight by different sex", x = "Age",
y = "Weight") +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 15, face = "bold", hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



We could see that as age increases, the mean weight also increases. The increase trend and speed is quite similar across different sex.

- 2. Fit the simple linear regression of weight on age. In a few sentences, as if for a public health audience, interpret the: intercept, slope, and residual standard deviation in anthropometric terms. Include the estimates and confidence intervals in your sentences to be quantitative but use no statistical jargon (e.g. “intercept”, “slope”). For example, use “difference in average weight among children one year older” rather than “slope”.

```
lmfit <- lm(wt ~ age, d)
summary(lmfit); confint(lmfit)
```

```
##
## Call:
## lm(formula = wt ~ age, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7237 -0.8276  0.1854  0.9183  4.5043
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.444528   0.204316   26.65   <2e-16 ***
## age         0.157003   0.005845   26.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.401 on 183 degrees of freedom
## Multiple R-squared:  0.7977, Adjusted R-squared:  0.7966
## F-statistic: 721.4 on 1 and 183 DF,  p-value: < 2.2e-16
```

```
##                 2.5 %    97.5 %
## (Intercept) 5.0414110 5.8476459
## age         0.1454701 0.1685361
```

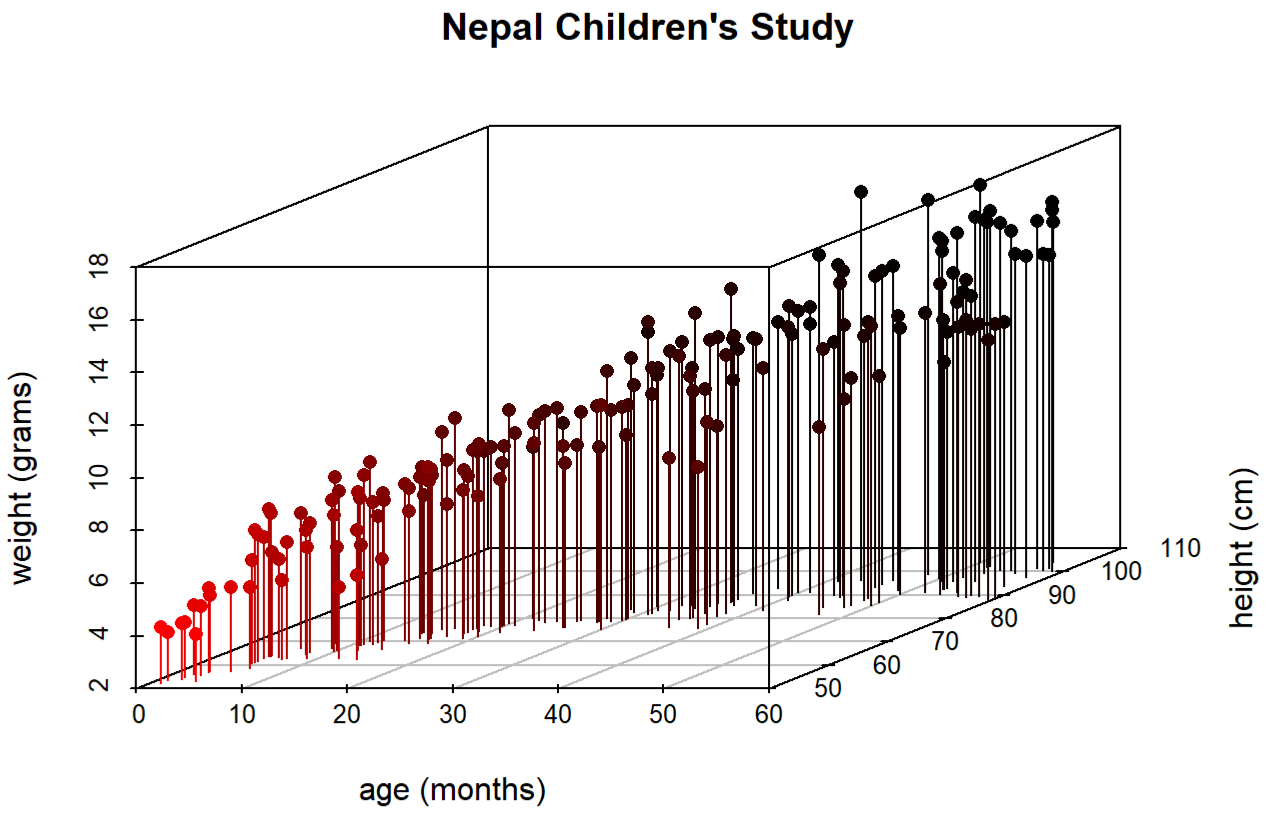
Overall, we can see that as age increases, the mean weight also increases. According to this simple linear regression, the average weight is 5.445kg(95%CI: 5.041, 5.848) for children at birth. Average weight will increase 0.157kg (95%CI:0.145,0.169) per month increase in age.

- 3.Now display the three variables age, weight, and height so that you can better understand their joint distribution.

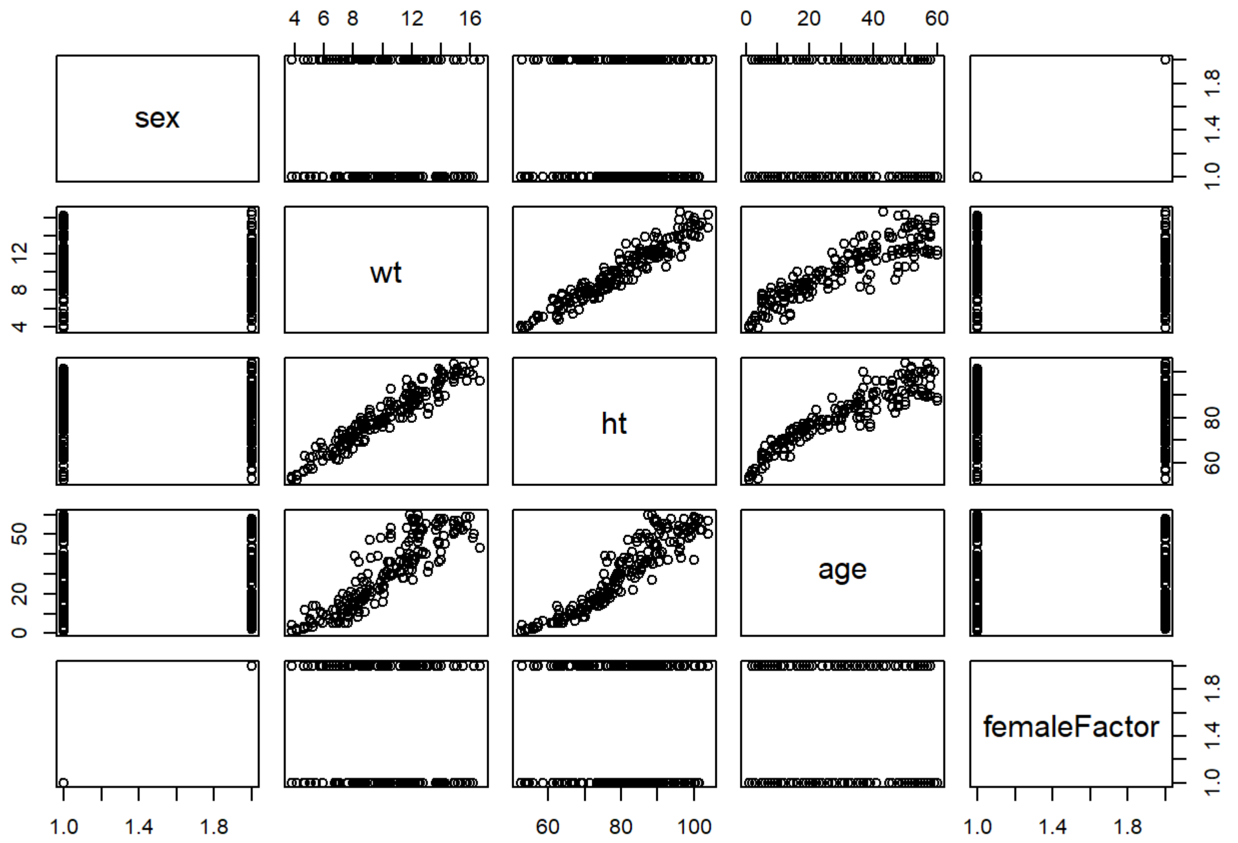
```
## install.packages("scatterplot3d")
## install.packages("rgl")

library(rgl)
```

```
## Warning: package 'rgl' was built under R version 4.1.2
```

```
library(scatterplot3d)

plot3d(d$age,d$ht,d$wt)
scatterplot3d(d$age,d$ht,d$wt,pch=16,type="h",highlight.3d=TRUE,xlab="age (months)",ylab="height (cm)",zlab="weight (grams)",main="Nepal Children's Study")
```


Nepal Children's Study
weight (grams)
2
4
6
8
10
12
14
16
18
0
10
20
30
40
50
60
age (months)
50
60
70
80
90
100
110
height (cm)

```
pairs(d)
```



- 4.Conduct a multiple linear regression of weight on age and height. In a few sentences, as if for a public health audience, interpret the intercept, age coefficient, and residual standard deviation in anthropometric terms. Include the estimates and confidence intervals in your sentences to be quantitative but use no statistical jargon (e.g. “intercept”, “slope”).

```
mlr.model <- lm(wt~age+ht,data=d)
summary(mlr.model); confint(mlr.model)
```

```
##
## Call:
## lm(formula = wt ~ age + ht, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48498 -0.53548  0.01508  0.51986  2.77917
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.297442   0.865929  -9.582   <2e-16 ***
## age          0.005368   0.010169   0.528    0.598
## ht           0.228086   0.014205  16.057   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9035 on 182 degrees of freedom
## Multiple R-squared:  0.9163, Adjusted R-squared:  0.9154
## F-statistic: 995.8 on 2 and 182 DF,  p-value: < 2.2e-16
```

```
##                   2.5 %      97.5 %
## (Intercept) -10.00599239 -6.58889209
## age          -0.01469529  0.02543175
## ht            0.20005782  0.25611318
```

Overall, we can see that as age and height increases, the mean weight also increases. According to this multiple linear regression, for children at birth and whose birth weight is 0, the average weight is -8.297 kg (95%CI:-10.006, -6.589).
For children with the height, average weight will increase 0.005 (-0.015,0.025) per month increase in age. However, after controlling for height, the age coefficient is much smaller when compared to what we get in simple linear regression. The 95% CI of age coefficient overlap 0, indication a non-significant linear relationship between weight and age for children with the same weight.
For children with the age, average weight will increase 0.228 (95%CI: 0.200,0.256) per centimeters increase height.

6.Draw a directed acyclic graph (DAG) showing the likely causal relationships of aging on height and weight.

```
#install.packages("dagitty")
#install.packages("ggdag")
library(dagitty)
```

```
## Warning: package 'dagitty' was built under R version 4.1.2
```

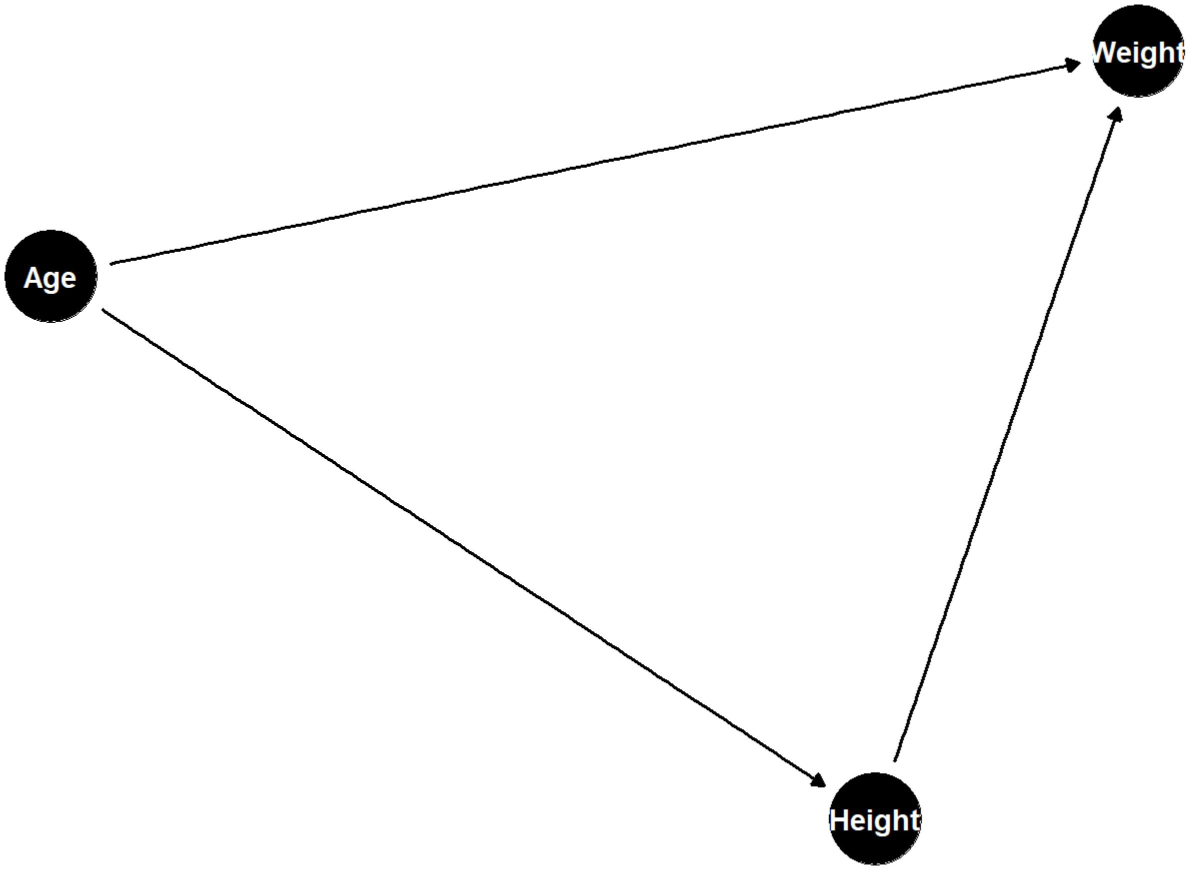
```
library(ggdag)
```

```
## Warning: package 'ggdag' was built under R version 4.1.2
```

```
##
## Attaching package: 'ggdag'
```

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

```
dag <- dagify (Weight ~ Height, Weight~ Age, Height~ Age)
ggdag(dag) + theme_dag()
```



II. Modeling Non-linear Relationships with MLR

1. Linear splines:

a. create three new variables:

age_c = age - 6
age_sp6 = (age – 6)+ = age – 6 if age > 6, 0 if not
age_sp12 = (age - 12)+ = age – 12 if age > 12, 0 if not

```
d_spline <- d %>% mutate(age_c = age - 6,
                         age_sp6 =ifelse(age-6>0, age-6,0),
                         age_sp12=ifelse(age-12>0, age-12,0))
```

b. Regress weight on age_c, age_sp6 and age_sp12

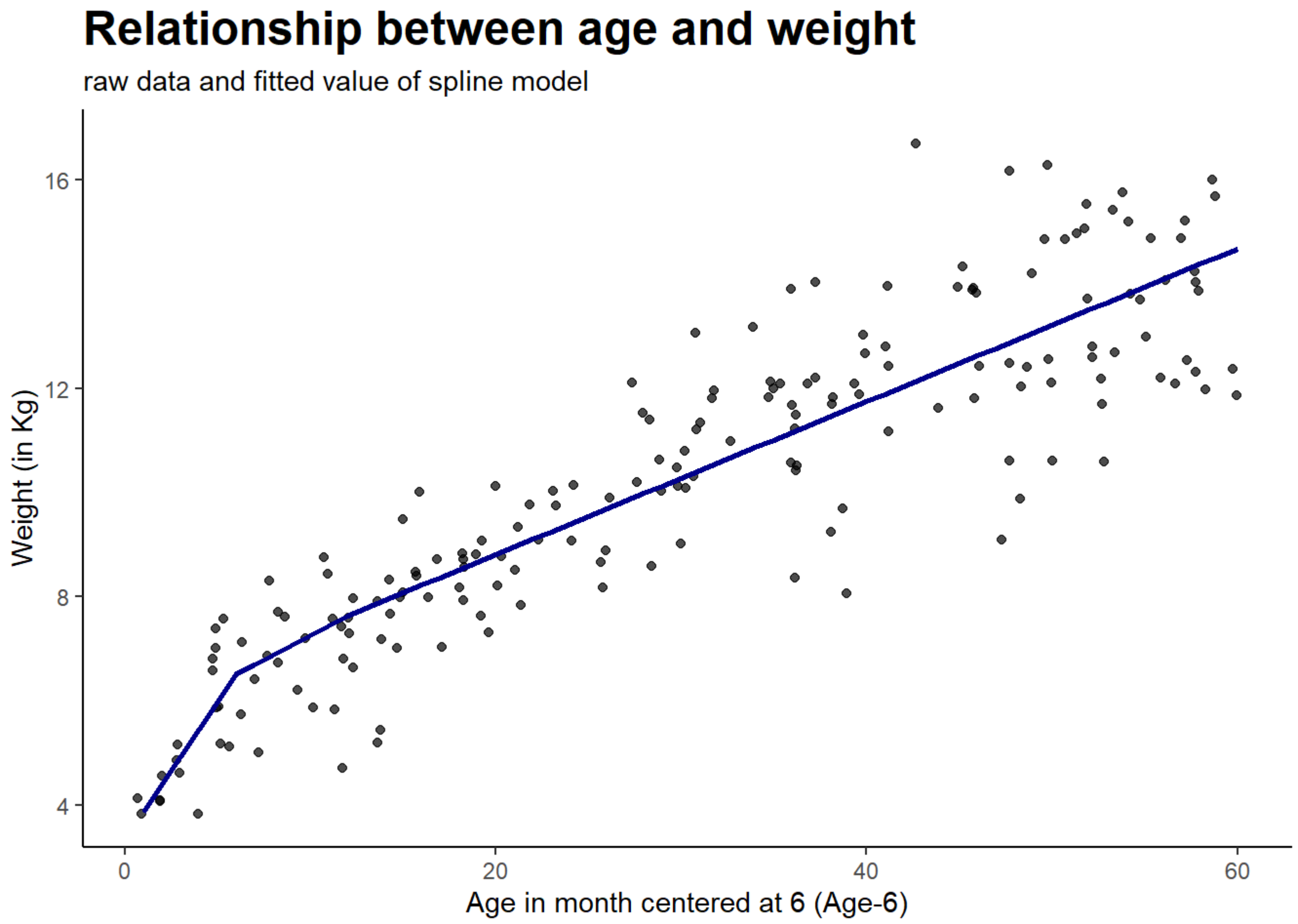
```
sp.model<-lm(data=d_spline, wt~age_c+age_sp6+age_sp12)
summary(sp.model); confint(sp.model)
```

```
##
## Call:
## lm(formula = wt ~ age_c + age_sp6 + age_sp12, data = d_spline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6698 -0.7883  0.0106  0.8976  4.5171
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5171     0.4022  16.205  < 2e-16 ***
## age_c         0.5285     0.1678   3.150  0.00191 **
## age_sp6      -0.3423     0.2265  -1.511  0.13250
## age_sp12     -0.0394     0.0817  -0.482  0.63025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.368 on 181 degrees of freedom
## Multiple R-squared:  0.809,  Adjusted R-squared:  0.8058
## F-statistic: 255.5 on 3 and 181 DF,  p-value: < 2.2e-16
```

```
##                  2.5 %    97.5 %
## (Intercept)  5.7235681 7.3106273
## age_c        0.1973905 0.8595585
## age_sp6     -0.7893468 0.1046705
## age_sp12    -0.2006070 0.1218142
```

c.Plot the raw weight against age data; add the fitted values from this regression.

```
ggplot(d_spline,aes(x=age, y=wt)) +
  geom_jitter(alpha = 0.7) + theme_bw() +
  geom_line(aes(x= age, y = sp.model$fitted.values),color = 'darkblue', size = 0.9) +
  #scale_y_continuous(breaks=seq(8,18,2),limits=c(8,18)) +
  #scale_x_continuous(breaks=seq(0,60,6),limits=c(0,60)) +
  labs(title = "Relationship between age and weight", subtitle = "raw data and fitted value of s
pline model", y = "Weight (in Kg)", x = "Age in month centered at 6 (Age-6)") +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 18, face = "bold"))
```



d.Interpret the meaning of the coefficients for the three terms: age_c, age_sp6 and age_sp12 as if for a growth journal.

age_c: For children younger than 6 months of age, their average weight will increase 0.53kg per month increase in age.

age_sp6: This coefficient is the is the difference in the change in weight per month of age between 6 months to 12 months of age*versus* before 6 months of age. For children between 6 months of age 12 months of age, their average weight will increase 0.5285-0.3423= 0.19kg per month increase in age.
age_sp12: This coefficient is the is the difference in the change in weight per month of age after 12 months of age *versus* before 6 months to 12 months of age. For children older than 12 months of age, their average weight will increase 0.5285-0.3423-0.0394 = 0.15kg per month increase in age.

e. Comment in a few sentences on the evidence from this analysis for or against a linear growth curve
Children’s weight increase rate are much faster within 6 months of age, then dropped significantly after 6 months (from 0.53kg per month to 0.19kg per month). For children older than 12 months of age, the average weight increase drop further to 0.15kg per month. Both from the model coefficient and from the plot we could see that the broken arrow relationship between average weight and age clearly violate the linearity assumption. There are evidences against a linear growth curve.

2. Cubic regression splines:
a. create three new variables: age2 = $(age– 6)^2$
age3= $(age-6)^3$
age_csp1 = $[(age-6)+]$

```
d_spline <- d_spline %>% mutate(age_c = (age-6),
                    age2 = (age-6)^2,
                     age3 = (age-6)^3,
                     age_csp1 = ifelse(age-6>0, (age-6)^3, 0))
```

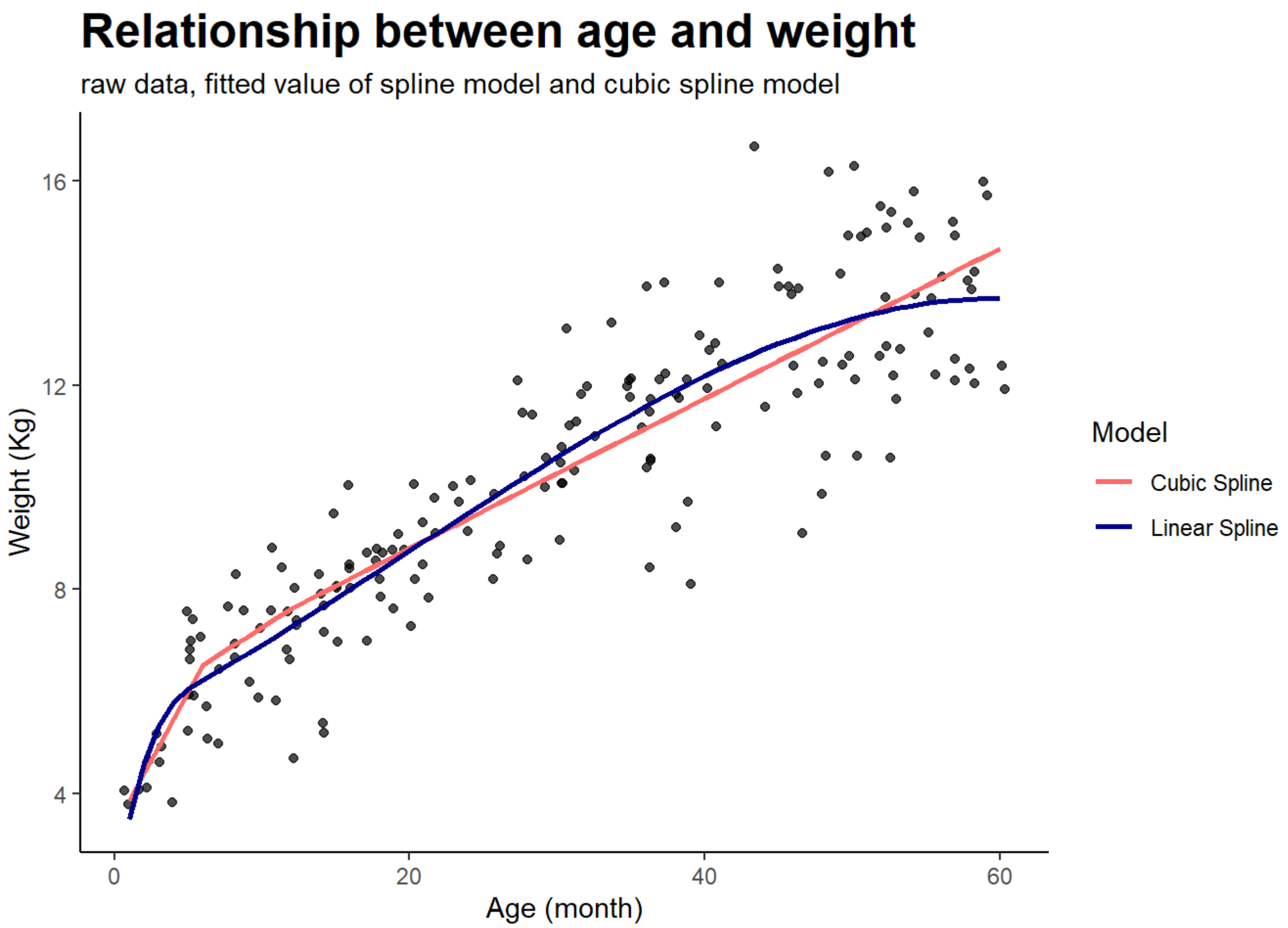
b. Regress weight on age_c, age2, age3 and age_csp1.

```
model_cs = lm(wt~ age_c + age2 + age3 + age_csp1,d_spline)
summary(model_cs)
```

```
##
## Call:
## lm(formula = wt ~ age_c + age2 + age3 + age_csp1, data = d_spline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9372 -0.8113  0.0994  0.7262  4.1256
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.233515   0.258945  24.073  < 2e-16 ***
## age_c        0.165051   0.047202   3.497 0.000593 ***
## age2         0.001623   0.002234   0.726 0.468502
## age3         0.015562   0.008252   1.886 0.060912 .
## age_csp1    -0.015601   0.008265  -1.888 0.060695 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.332 on 180 degrees of freedom
## Multiple R-squared:   0.82,  Adjusted R-squared:  0.8159
## F-statistic: 204.9 on 4 and 180 DF,  p-value: < 2.2e-16
```

c. Plot the weight data with the fitted values from this “cubic regression spline”, added along with the fitted values from the linear spline.

```
ggplot(d_spline,aes(x=age, y=wt)) +
  geom_jitter(alpha = 0.7) + theme_bw() +
  geom_line(aes(x= age, y = sp.model$fitted.values, color='sp'), size = 0.9) +
  geom_line(aes(x= age, y = model_cs$fitted.value, color = 'cs'), size = 0.9) +
  scale_colour_manual(name = 'Model', values = c('sp'='Indianred1','cs'='darkblue'),
labels = c('Cubic Spline', 'Linear Spline')) +
  labs(title = "Relationship between age and weight", subtitle = "raw data, fitted value of spli
ne model and cubic spline model", y = "Weight (Kg)", x = "Age (month)") +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 18, face = "bold"))
```



d. Contrast your estimated curves using linear and cubic splines.

The predication values of the cubic spline model and linear spline models are quite similar. Linear spline assume that the slope is constant within each range and thus may cause inaccuracy in prediction. Cubic splines are more smoother than the linear splines and fit the data better especially in the far right when age is large and variance are high.

The predicted growth rate are higher in the cubic spline approximately from 20-50 months of age. Then the predicted growth rate are higher in linear spline models.

3. Natural cubic splines

a. Read about natural splines, ns(x,df), to learn how they differ from regression splines. Natural splines are linear regressions.

b. Regress weight on the natural spline ns(age,df=3).

```
model_ns <- lm(wt ~ ns(age, df = 3), data = d)
summary(model_ns)
```

```
##
## Call:
## lm(formula = wt ~ ns(age, df = 3), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9308 -0.8202  0.1021  0.7982  4.1894
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.8919     0.3471   14.09   <2e-16 ***
## ns(age, df = 3)1   6.4816     0.4253   15.24   <2e-16 ***
## ns(age, df = 3)2  11.8892     0.8737   13.61   <2e-16 ***
## ns(age, df = 3)3   6.8342     0.3321   20.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.344 on 181 degrees of freedom
## Multiple R-squared:  0.8158, Adjusted R-squared:  0.8128
## F-statistic: 267.3 on 3 and 181 DF,  p-value: < 2.2e-16
```

c. Obtain the design matrix (call it X) for this linear regression. (Use the R command model.matrix). Calculate the “hat” matrix $H = X(X'X)^{-1}X$ that takes its name because the vector of predicted values in the regression (“Y-hat”) is given by the matrix product H and Y where Y is the vector of observed responses. That is, the jth predicted value is a linear combination of all the responses Y with weights given by jth row of H.

```
x = model.matrix(model_ns)
head(x)
```

```
##     (Intercept) ns(age, df = 3)1 ns(age, df = 3)2 ns(age, df = 3)3
## 446           1       0.00000000       0.00000000       0.00000000
## 671           1       0.00000000       0.00000000       0.00000000
## 241           1      -0.01340447       0.04094975      -0.02751999
## 261           1      -0.01340447       0.04094975      -0.02751999
## 301           1      -0.01340447       0.04094975      -0.02751999
## 196           1      -0.02658949       0.08169303      -0.05490123
```

```
H = x %*% solve(t(x) %*% x) %*% t(x)
```

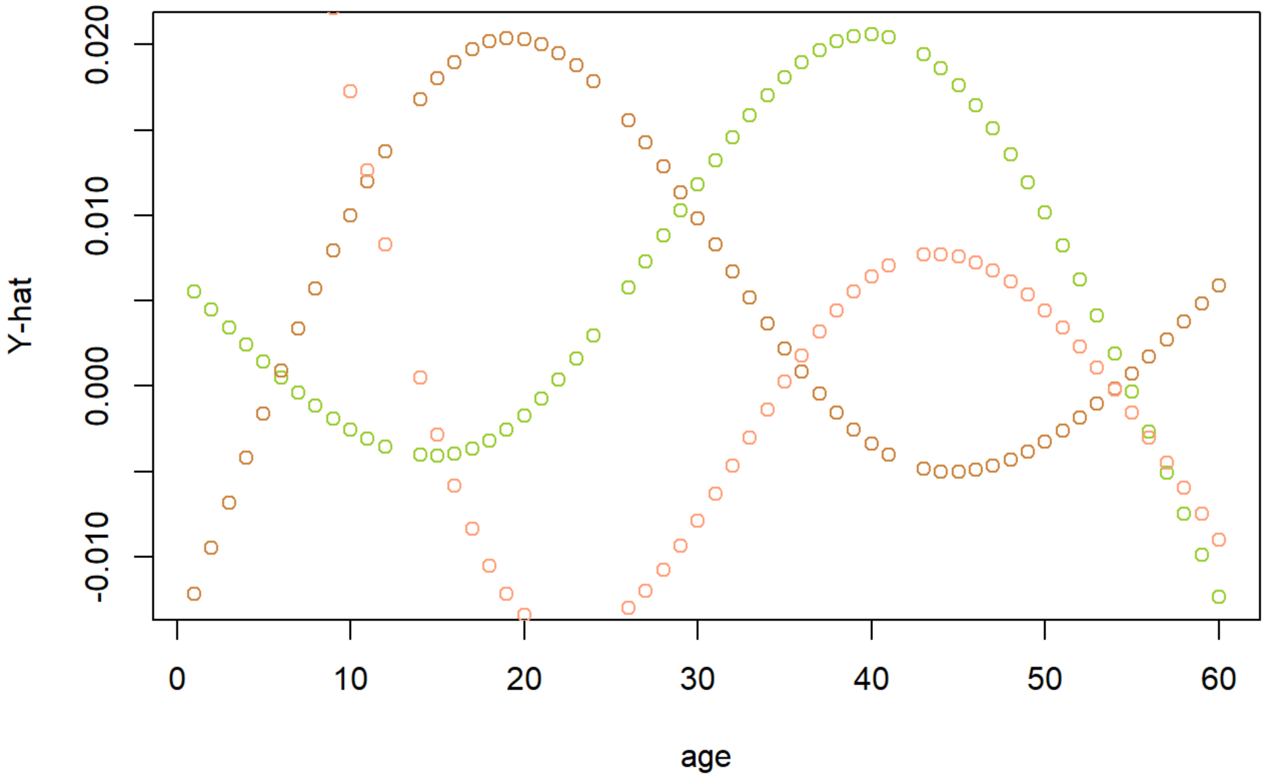
Choose three children from the data with different ages.

```
j <- sample(c(1:nrow(d)),3)
j
```

```
## [1] 123  63   1
```

- **On a single graph, plot each child’s row of H against age.**

```
library(tidyr)
hplot = cbind(d$age, H[j[1], ], H[j[2], ], H[j[3], ])
colnames(hplot)= c("age",paste("child",j[1], sep ="_"), paste("child",j[2], sep ="_"), paste("ch
ild",j[3], sep ="_"))
hplot = as.data.frame(hplot)

plot(hplot[,1], hplot[,2], col = "yellowgreen",xlab="age", ylab ="Y-hat" )
points(hplot[,1], hplot[,3], col= "tan3")
points(hplot[,1], hplot[,4], col= "lightsalmon")
```

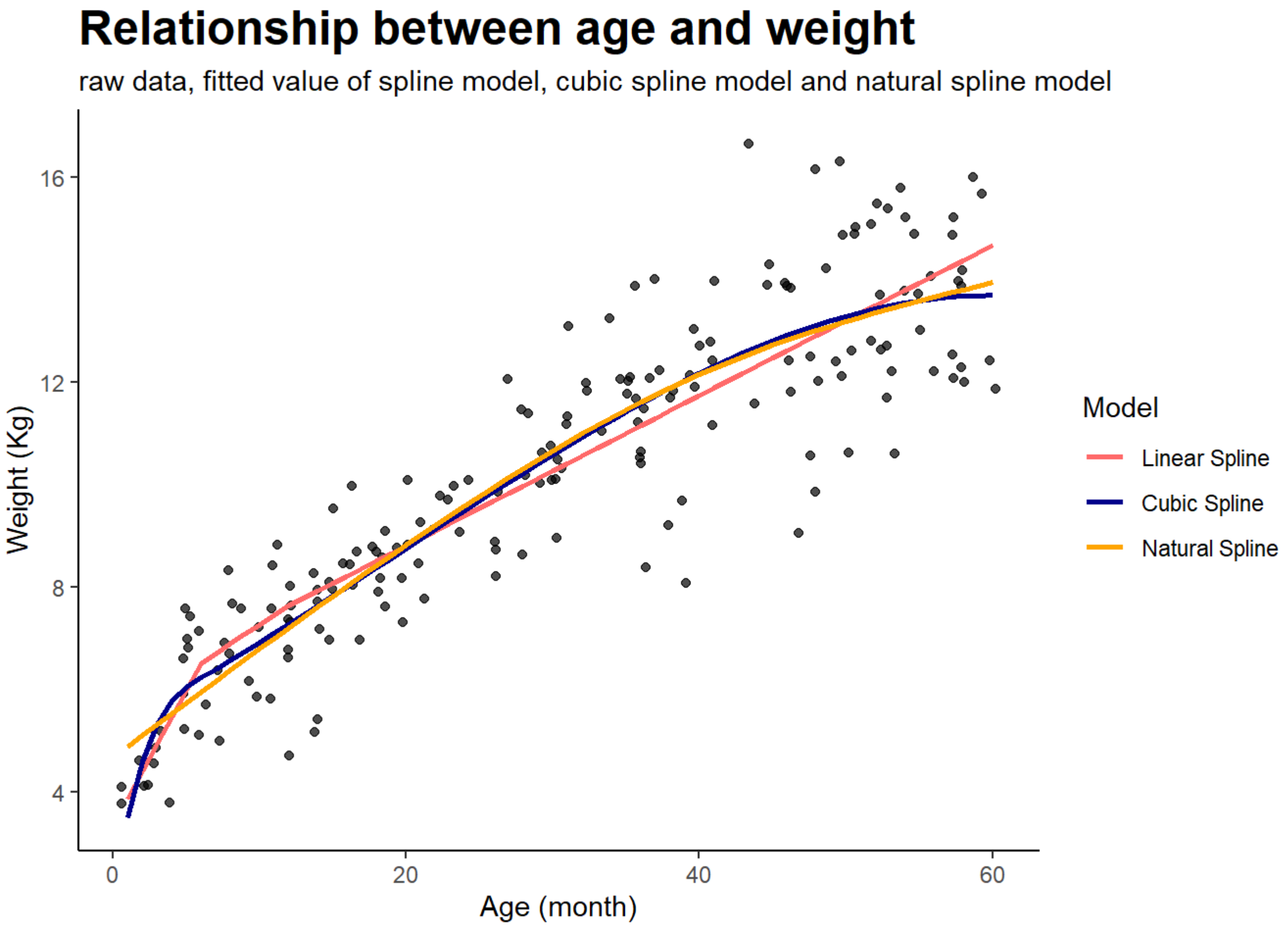


Comment on patterns you observe; i.e. what values of Y are most informative for each child’s predicted value?

The y axis the the Y-hat of the child’s age. After comparing the most likely y-hat which is the peak of each curve and the relative x (age) with the true age of children, we can conclude that when y hat is maximized, we can get a quite accurate estimate of children’s age.

d. Plot the weight data as above in 2c. Add the fitted values from this “natural cubic spline” along with the fitted values from the linear spline and cubic regression spline. Contrast your estimated curves.

```
ggplot(d_spline,aes(x=age, y=wt)) +
  geom_jitter(alpha = 0.7) + theme_bw() +
  geom_line(aes(x= age, y = sp.model$fitted.values, color='sp'), size = 0.9) +
  geom_line(aes(x= age, y = model_cs$fitted.value, color = 'cs'), size = 0.9) +
  geom_line(aes(x= age, y = model_ns$fitted.value, color = 'ns'), size = 0.9) +
  scale_colour_manual(name = 'Model', values = c('sp'='Indianred1','cs'='darkblue', 'ns'= 'orang
e'),
labels = c( 'Linear Spline','Cubic Spline', 'Natural Spline')) +
  labs(title = "Relationship between age and weight", subtitle = "raw data, fitted value of spli
ne model, cubic spline model and natural spline model", y = "Weight (Kg)", x = "Age (month)") +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 18, face = "bold"))
```



Linear spline seems to be the most ‘unique’ model among these three models for in many age ranges, it has different predicted weight when comparing to cubic splines and natural spline. The cubic spline and natural spline model agree well with each other, except for the far left and the far right where we only have limited number of observations.

Overall, natural spline are more sensitive than the other two models in this case. The weight predicted by natural spline model grow rapidly within 6 months of birth as what we would expect in the reality and fits the data well.

### III. Selecting Among Competing Models Based Upon Cross-validated Prediction Error

For each of the models above, we used 3 or 4 degrees of freedom for age, thereby allowing the relationship of average weight to be a non-linear function of age. The question is how many degrees of freedom are optimal to predict weight given any one of these methods? In this question, we use cross-validation to choose the degrees of freedom, df=1,…,8 within the natural spline family.

1. Randomly split the observations into 10 categories

2. For each df value, obtain the total cross-validated prediction error by regressing weight on ns(age, df), df=1,..,8, leaving out 1/10th of the observations and summing the squared prediction errors for the left out values across the 10 “leave-out” iterations.

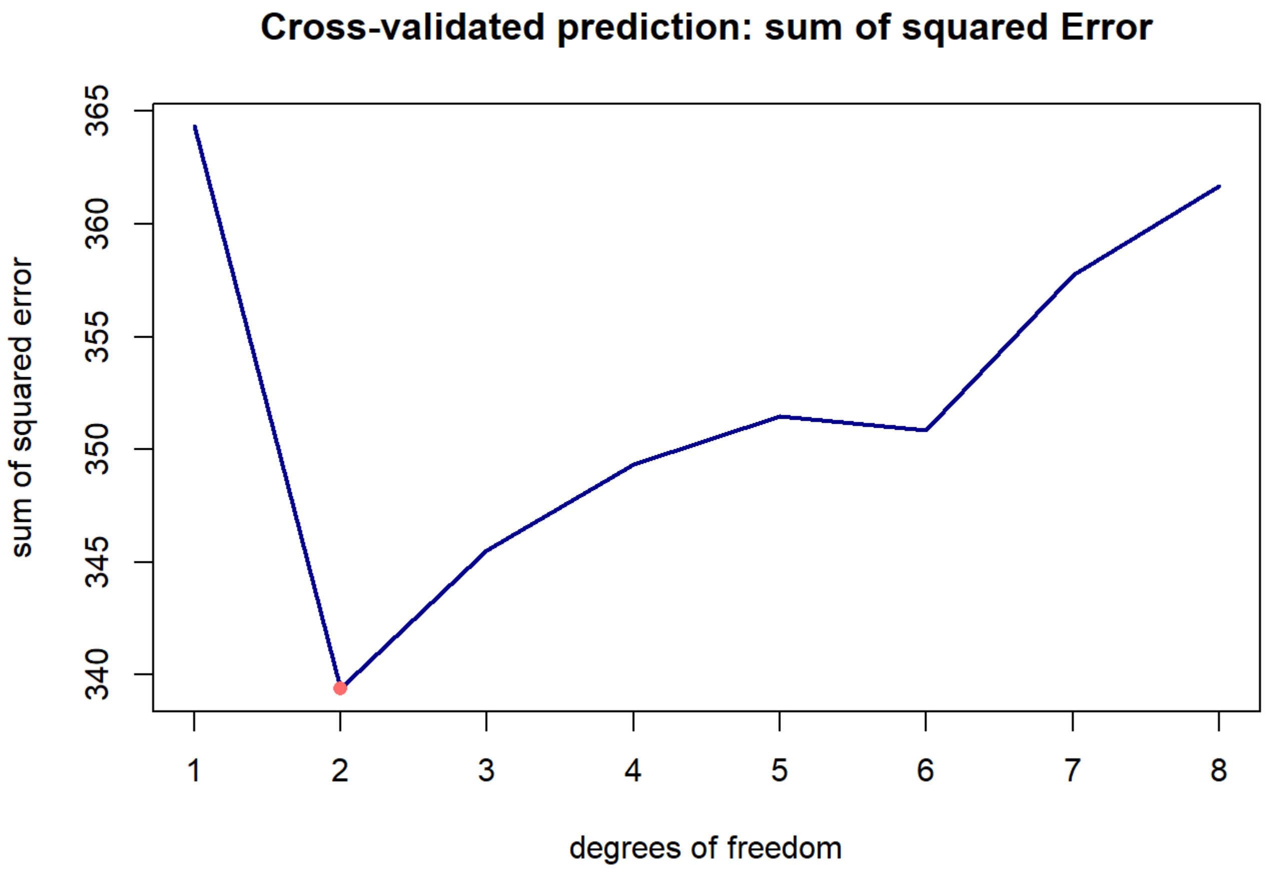
3. Plot the total cross-validated prediction error against the degrees of freedom to see which of the df values results in the best predictions of data, not also used to fit the model.

```
# randomly split the observations into 10 groups
rows <- 1:nrow(d)

shuffled.rows <- sample(rows, replace = FALSE) ## shuffle the rows

## Divide the data in 10 folds
folds <- cut(rows, breaks = 10, labels=FALSE)

sum_error=rep(NA,8)

#df=1:8

for (i in 1:8) {
    pred.data = NULL
        for (j in 1:10){
        test.rows=shuffled.rows[which(folds==j)]
        train.rows=shuffled.rows[which(folds!=j)]

        test.data=d[test.rows,]
        train.data=d[train.rows,]

        # Fit ns model with parameter: i
        fitns <- lm(wt ~ ns(age, i), data = train.data)
        test.data$predicted.ns <- predict(fitns, newdata = data.frame(age=test.data[,"age"]))

        # Stack and store new predicted data
        pred.data = rbind(pred.data,test.data)
        }
 sum_error[i]=sum((pred.data$predicted.ns-pred.data$wt)^2)
}

sum_error
```

```
## [1] 364.3370 339.3772 345.5361 349.3217 351.4608 350.8666 357.7346 361.6804
```

PLOT SSE

```
plot(sum_error, type = "l", main = "Cross-validated prediction: sum of squared Error",
     xlab = "degrees of freedom", ylab = "sum of squared error", col = 'darkblue', lwd =2)
points(x = which.min(sum_error), y = min(sum_error),col = 'Indianred1', pch = 16)
```



When we define degree of freedom as 2, we would get the lease sum of squared error, indicating the most accurate prediction.

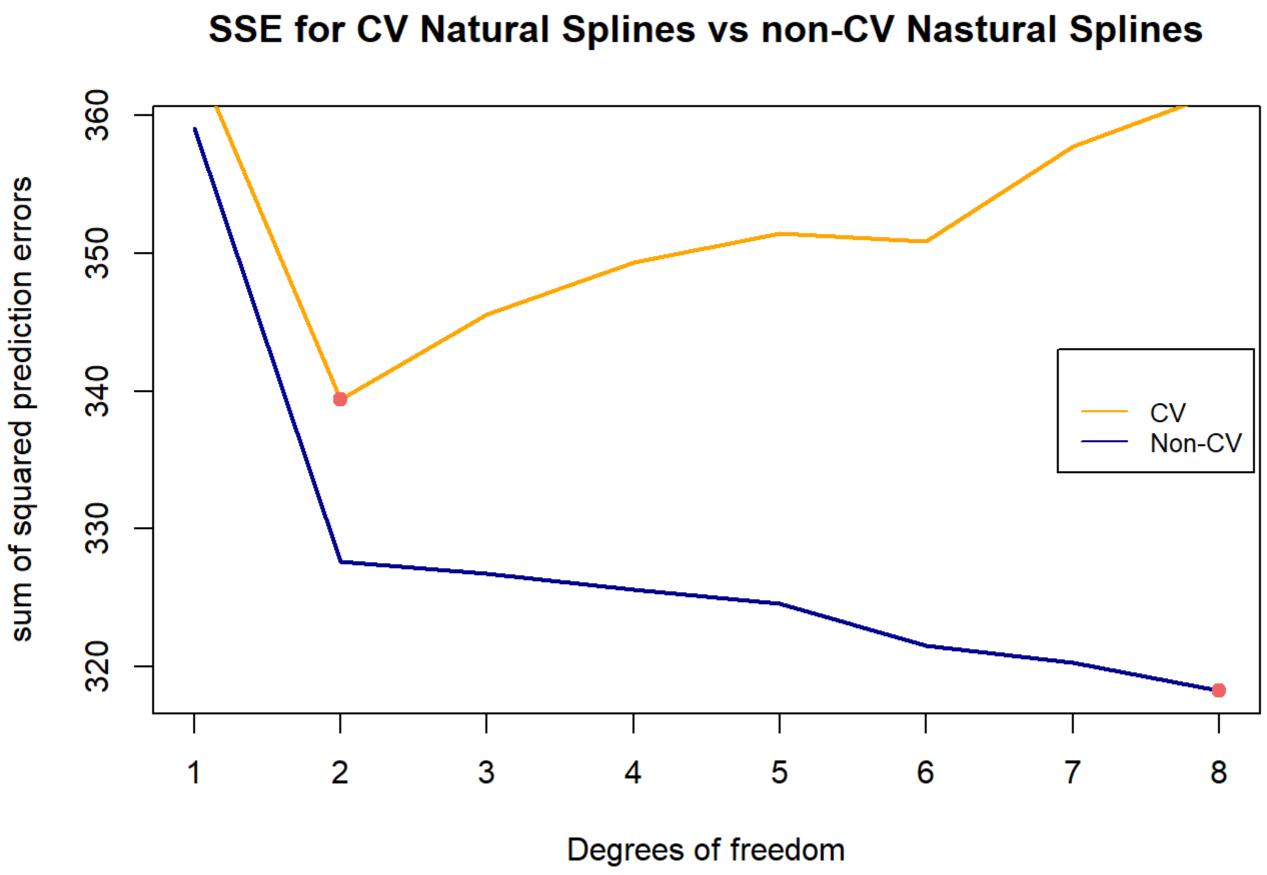
4. Compare the cross-validated prediction error to the non-CV prediction error for each df where the latter uses the same data to fit the model as assess its prediction error.

```
d_noncv= d
sum_error_noncv=rep(NA,8)

## test through df = 1,2,...8
for (i in 1:8) {
    # Fit ns model with parameter: i
    fitns <- lm(wt ~ ns(age, i), data = d_noncv)
    d_noncv$predicted.ns = predict(fitns, newdata = data.frame(age=d_noncv[,"age"]))
    sum_error_noncv[i]=sum((d_noncv$wt - d_noncv$predicted.ns)^2)
        }
sum_error_noncv; which.min(sum_error);min(sum_error)
```

```
## [1] 359.0126 327.6287 326.7787 325.6065 324.6211 321.5112 320.3036 318.2611
## [1] 2
```

```
## [1] 339.3772
```

```
plot(sum_error_noncv, type = "l", main = "SSE for CV Natural Splines vs non-CV Nastural Splines"
, xlab = "Degrees of freedom", ylab = "sum of squared prediction errors", col = "darkblue", lwd
 =2)
points(x = which.min(sum_error_noncv), y = min(sum_error_noncv),col = 'Indianred2', pch = 19)
lines(sum_error, lwd = 2, col = "Orange")
points(x = which.min(sum_error), y = min(sum_error),col = 'Indianred2', pch = 19)
legend(6.9,343, legend=c("CV", "Non-CV"), col=c("Orange", "darkblue"),title="", lty=1, cex=0.8)
```



```
which.min(sum_error_noncv);min(sum_error_noncv)
```

```
## [1] 8
```

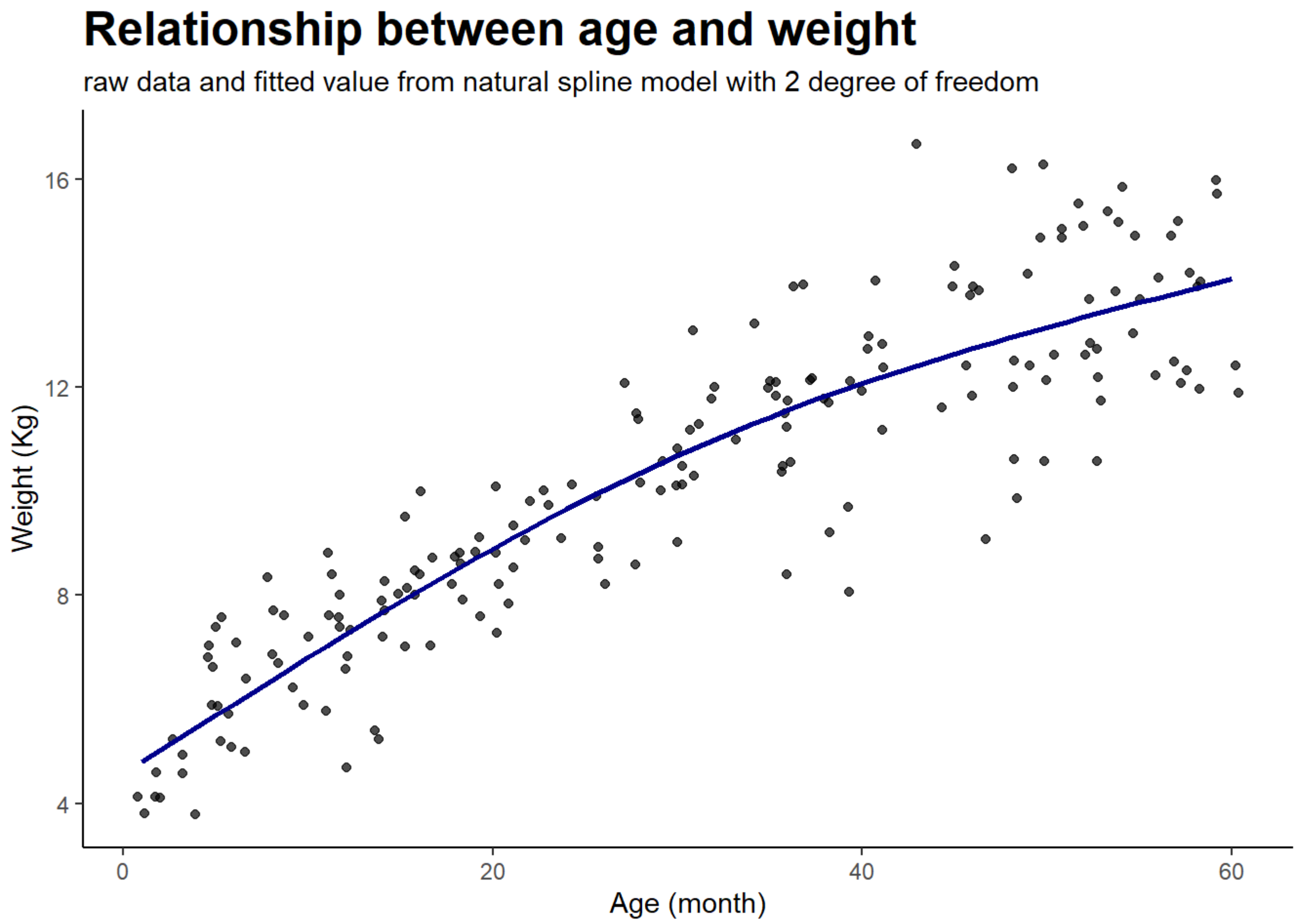
```
## [1] 318.2611
```

As we can see from the plot, if we do not use cross validation, the sum of squared errors will drop as we add to degrees of freedom (in the range of 1-8), and the natural spline model will achieve minimum SSE at df=8 (SSE = 318.3). However, if we use cross validation, the natural spline model will achieve minimum SSE at df=2 (SSE =

340.4).From my perspective, the cross validation will provide a more accurate evaluation to model. The low SSE of non-cross validated model with df=8 may fall into the pitfall of overfitting of this data set and can not be generalized to other data set.

5. Fit this optimal model to all of the data; plot weight data against age, and add this optimal curve to the display.

```
d_nscv2 = d
model_nscv2 <- lm(wt ~ ns(age, 2), data = d_nscv2)
d_nscv2$predicted.ns2 <- predict(model_nscv2)
ggplot(d_nscv2,aes(x=age, y=wt)) +
  geom_jitter(alpha = 0.7) + theme_bw() +
  geom_line(aes(x= age, y = predicted.ns2), col = "darkblue", size = 0.9) +
  labs(title = "Relationship between age and weight", subtitle = "raw data and fitted value from
natural spline model with 2 degree of freedom", y = "Weight (Kg)", x = "Age (month)") +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 18, face = "bold"))
```



6. In a paragraph or two, summarize your findings as if for a public health journal. Explain the method you used and the results you found.

After exploratory data analysis and visualization, we conclude that natural spline model would provide the most accurate prediction of weight using age. In order to find the most suitable degree of freedom of the natural spline model, we adopt ten-fold cross validation methods. First, we choose a degree of freedom. Second, we split the observations into 10 categories, use 1/10 as test data and 9/10 as training data. We train the model using the 9/10 training data (fit to natural spline model) and then predict the weight of the 1/10 observations. By calculating the squared difference true weight and the predicted weight and sum them up, we get the squared sum of errors of this 1/10 observations. Then we do the same thing for 9 times, each time using a different 1/10 as testing data. Finally, we sum up all the suqared errors and get the cross validated SSE of the model.

We test the through degree of freedom from 1 to 8, and find that the natural spline term will have the least SSE when degree of freedom is equal to 2. The natural spline model accurately captures the growth rate change as children grow.

For non-cross validated natural spline model, SSE degree as degree of freedom increases (in range 1-8). However, this could cause overfitting of this data set. Since cross validation methods would provide more powerful evaluation of models, we choose the natural spline model with 2 degrees of freedom.